MARK ROBERTS MOTION CONTROL

FLAIR V7.4

OPERATOR'S MANUAL

Flair v7.4 Operator’s Manual

Although every care has been taken to ensure that the information in this document is accurate and up to date, Mark Roberts Motion Control continuously strives to improve their products and may make changes to the hardware, firmware, and software described in this document. Mark Roberts Motion Control therefore cannot be held responsible for any error or omission in this document.

Contact information

Mark Roberts Motion Control Ltd.
Unit 3, South East Studios
Blindley Heath
Surrey
RH7 6JP
United Kingdom

Telephone: +44 (0) 1342 838000

E-mail: info@mrmoco.com (sales and general enquiries)
support@mrmoco.com (customer support)

Web: www.mrmoco.com
www.mrmocorentals.com

Contents

Chapter 1 **Introduction**

1.1. About this manual

This manual is designed to be used as a reference guide rather than something to study before you touch the machine. The Quick Guide should be enough to get you up and running, assuming that the motors are all connected and set up. The rest of the manual is divided into reference sections, which should allow you to easily get assistance on any area of running the program.

There is an extensive glossary at the back of the manual, and you must refer to this if you come across any term that you are not familiar with or not sure about. It is also possible that we use terms in a way that is unfamiliar to you. If anything you come across does not make sense, do check the definitions of any words in the Glossary to help resolve this.

If you have any suggestions for improving this manual or areas that you find hard to handle, let us know and we will take them into consideration for the next manual release.

1.1.1. Safety

Safety is a very important aspect of any system and must be kept in mind at all times especially with motion control equipment getting faster and more powerful. The level of safety at which the system is operated is very much up to the user, and he must be continuously aware of what is happening with and around his system.

Here are some safety rules, which must be obeyed to ensure safe operation:

- Always have one person ready to press the red Emergency Stop button when there is power applied to the system.
- Always watch the system, not the video monitor when you are operating. The video monitor gives a very narrow field of view, and will not tell you if someone is sitting in the way of the rig. If you want to watch the move, record it and watch it played back.
- Do not allow people to stand on or near the system unless that is specifically required for the shoot, and make sure they know when the system is going to move. Also provide them with a clear path to move away from the rig if this is required.
- Always install track buffers on the rails, and place them as close as possible to the end of the move. The same is true for lift buffers if these are fitted. This also applies to end limits, which will turn the axis off if it goes out of the intended envelope of operation for that move.
- Be aware of the back of the rig and how it moves as you rotate, the back end will move in the opposite direction of the arm, and you have to allow clearance for this.
- Turn the system off whenever you leave it, or at least disengage the motors and press the emergency stop. If you are leaving the rig alone, then the area should be locked or someone should be there to ensure no unauthorised persons get near the system.
- Always test fast moves slowly and bring them up to speed bit by bit. This is particularly true if you are running part of a move or running with pre-roll or

postroll as the size of the pre-roll or postroll increases as the square of the speed. i.e. if you double the speed the movement will quadruple.

Note

A pre-roll is when the rig has to get up to speed before the move if the move starts at speed, and postroll is when the move is stopped, or when the programmed move ends at speed, the rig has to slow down to stop moving.

- Always announce loudly when the rig is about to move if there are actors or crew in the studio as they may not always be able to see the rig.

Do not operate the system while under the influence of alcohol or drugs, even medical ones unless fully cleared by your doctor.

1.1.2. High Speed Robot Safety

With the advent of high speed industrial robots being used for Motion Control, the need for safety and care is heightened as these robots can move faster than someone can get out of the way even if they are watching the robot. Due to this, the system normally runs at a lower speed (1/6) of actual shooting speed unless the user specifically sets it to run “Fast”. This system is there for safety reasons and no attempt should be made to “work” around this or trick the system into running faster than it is set to.

It is the duty and responsibility of the operator and any system technicians to ensure safe practices are maintained at all times even at the end of a long and hard day when people’s awareness typically decreases. If at any time safe practices are not being followed, then the robot should be disengaged and this addressed immediately.

1.1.3. About the software

The best way to learn to use the software is to try it out. There is a negligible possibility of damaging the computer or software through use so if you want to know how it will work, try it out. However it is possible to damage the rig, particularly if not properly or fully set up. If you are concerned about this, keep the emergency stop nearby at all times and try all moves slowly before running them at full speed, as covered above.

This manual is intended as a user's guide to the software, it is not a specification for the software and features mentioned in this manual may or may not be available on the system you have bought. Mark Roberts Motion Control reserves the right to remove, add or alter features from those listed in this manual without prior warning.

Chapter 2 **Installation**

2.1. Installing the program

Flair is installed under Microsoft Windows®. The program is installed using an automated installation program. Flair can run with or without any motion control hardware. If it does not detect any hardware or if it has been set not to load any hardware then it will start as normal but will not try to "talk" to any motion control hardware. Following are the minimum hardware requirements for Flair 7 installation:

- A graphics card that supports OpenGL v4.3 or higher
- A Full HD (1920 x 1080) monitor ideally with an HDMI port for the best quality image, such as the AOC M2060SWDA2 19.5" Full HD Monitor.
- For the new integrated Live view in Flair 7 you will need to add a PCI-e card to your PC to grab the SDI feed from the camera. The recommended model that was successfully used in testing is the Blackmagic: Decklink Mini Recorder. You can get this card for HD video quality or 4K video quality:
  https://cvp.com/product/blackmagic_decklink-mini-recorder
  https://cvp.com/product/blackmagic_decklink_mini_recorder_4k

FlairSetup.exe is the automated installer for Flair. Multiple versions of Flair can be installed on one computer. The installer is either run from a CD or downloaded from the website, saved in a suitable location (such as C:\FlairInstall), and then run. Once the installer is started it will prompt you by asking what you would like the program to be called in the Windows Start Menu (default: Flair). It will also prompt you for which directory to install the program in on the hard disk (default: C:\Flair). Most importantly, it will ask whether you want a New Installation or are upgrading an Existing Installation. If you have not installed Flair before then choose a new installation, but if you are only upgrading the Flair version and don't want to lose your configurations then you MUST choose Upgrade. If you do not it will overwrite your existing configurations and you will lose the setup of your rig. It is strongly recommended that you make a backup of the Flair folder prior to any upgrades for total security. Once installed, you can access Flair, Flair documentation and various additional functions through the Start Menu in Windows.

2.2. Realtime Flair

For full visual effects motion control, Flair must be run in a "real-time" manner. This means that it must act and respond within known or exact time periods. Windows itself does not allow this type of operation and the motion of the motors could be stopped for some other intensive task such as downloading an image. In order to guarantee correct real-time operation with Flair, a real-time operating system must be installed that operates behind the scenes and ensures proper responsiveness. We have chosen to use "Intime®" by Tenasys. This will normally be installed on your computer at MRMC if so required. However, Flair can run perfectly well on a normal Windows machine, but MRMC cannot guarantee precise motion control features such as synchronizing to a camera when so run. It is important to understand that there are two sides to Flair, one is the Graphic User interface that responds to the User's actions with mouse and keyboard and the other is the Real-Time porting that runs at high speed communicating to the motor control boards and moving the motors in the expected fashion.

2.2.1. Flair Classic

The original version of Flair was called simply that: “Flair” but with the advent of a new UI for the Flair system, the original UI implementation of Flair has been re-named “FlairClassic”. It is identical to all intents and purposes, but since 2021, new features are only being added to FlairClassic.

2.2.2. Starting Flair

To start Flair, either click on one of the Flair shortcut icons on screen or select Flair from the Windows Start Menu. If you have made a new installation of the program, Flair will check if you have a suitable license installed. If you do not it will allow you to run the program but certain functions will be disabled. Contact MRMC if you require a new license. If you are using Flair to run some hardware, the program will bring up various messages indicating which board it is currently loading. Remember to have your hardware plugged in and switched on otherwise the program will fail to load. When completed Flair is ready to run.

2.2.3. Hot Booting

Hot booting is the action of reloading a previously run program without losing axis positions. For example, the user may have accidentally closed Flair, or a software conflict may have caused Flair to quit unexpectedly. Hot booting will re-start Flair without resetting all the axis boards and will save the axes positions. To do this, hold down the CTRL key, while clicking on the Flair icon, to start the program. Assuming that the fault was not hardware related or that the hardware has not been switched off, Flair will load with all the current axis positional data. You will need to read the last stored job from disk to get the move data back. Older versions of Flair used to save the move data, but this feature has been lost in exchange for faster program execution. You can configure Flair to automatically ask if you wish to hot boot when it determines that some part of the system is still functional.

2.2.4. Importance of Configuration Reports

Normally a Mark Roberts Motion Control system comes fully configured with the Flair software already installed for the customer. All the customer has to know about the software files is how to start the program – Flair – as in the section “Starting Flair”. However, it is recommended you also make regular configuration reports of your system and keep some copies for yourself and also regularly send copies to Mark Roberts Motion Control. Configuration reports are simply copies of your configuration files and Flair program version for backup and diagnostic purposes. These reports are vital for restoring a corrupt program, copying Flair to a replacement PC (either because of PC failure or simply for PC upgrade), and reporting bugs to Mark Roberts Motion Control. Making a report is fully automated and is accessed through the **Help → Send Report** function under Flair. From the pulldown menu, simply select Configuration so MRMC knows why you are sending it. It is recommended that you have a way to connect your machine to the internet so that these files can be sent directly from the computer. You can also save them to a storage device and send them from another computer, or to save them for your own use. Contact MRMC for how to re-implement a saved configuration.

2.2.5. Files Installed and Changed

The most important file the user should be familiar with is the Flair Initialisation file, Flair.ini. It sets up exactly what features your software has or doesn’t have, how many axes to run and so on. Chapter 2 Installation gives some of the options the user can change.

The rest of this section is for experienced computer users who wish to know more about the installation, it should not concern the average user and is supplied for information only.

File	Description
Flair.ini	Initialisation file that configures the software for your specific use. Changes such things as the number of axes etc.
NetworkDirect.ini	Configuration file for the network.
Axis.dbf	Configuration file for axes.
Config.dbf	Configuration file for miscellaneous parts of the programme.
lens.dbf	Configuration file for lenses used by system.
Carts.dbf	Configuration file for Cartesian Limits.
Serial.dbf	Serial Devices configuration file
Kine.dbf	Kinematics configuration file.
Model	Directory of model parts for the graphic rig representation.
Documentation	Directory the manual as doc file and as HTML.

Some of the files may be missing if they are not applicable to your system, and if you have multiple applications, then there may be more files.

2.3. Directory Tree Overview of Typical Installation

Additional files than the ones listed may be present, this lists the minimum files normally installed.

Flair

	JOBS	**CONFIGS**	**BTLS**
Flair.exe	(Job files)	AXIS.DBF	AXIS.BTL
Flair.ini		LENS.DBF	Quad_Axis.BTL
Winroot.rta		CARTS.DBF	RIC.BTL
NetworkDirect.ini		CONFIG.DBF	ROOT.BTL
		VTRS.DBF	RTC.BTL
		SERIAL.DBF	SERIAL.BTL
		KINE.DBF	etc

There may be some additional log files that are generated while running Flair to aid in diagnostics. These include Flairout.txt, FlairLoad.txt, FlairData.txt, Rootout.txt and Flairout.txt. The amount of diagnostic data that the log files contain depends on the Debug Level specified in the Flair Configuration File (Flair.ini).

New program versions may be added, see .

Chapter 3 **Quick Guide**

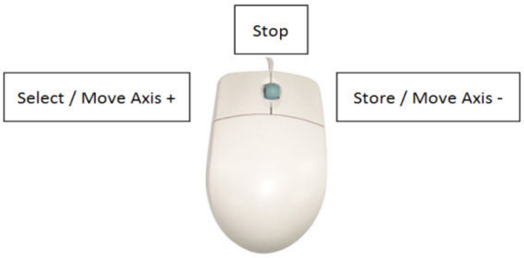
This is a very short quick start guide to Flair. The rest of the manual is more of a reference manual than a guide.

The quick guide is intended for a new user who just wants to make some simple moves. It assumes that the system has been fully set up and is ready for use. For more in-depth information, read the relevant sections of this manual. Given a system that is correctly set up, and has been left in its correct zero position, a new user should be able to start shooting moves within a very few minutes!

As you read this guide, perform each step as it is described and you will discover how easy the system is to run. Don't read through the whole section first and then try to use it, as you need to actually use the system to fully understand each step.

Several of the functions described here can be done with either the mouse or with keys on the Hand Held box.

3.1. Mouse Operation



The **left hand button** fulfils the normal mouse function of “select”. When the cursor is over an item, pressing the left hand mouse button will cause that item to be selected and if it is a push button of any kind, it will be pushed. Active or selected items often have a red or black highlight around them so you can easily tell which item is active. This is important to know as key presses on the keyboard are only directed at the active window or item; i.e. if you want to type something into a box, you have to click on it (select it) first! This button is referred to as the “select button” and if you are instructed to “select” an item, that means move the cursor above the item and click the left hand button. On many buttons and windows on the screen, if the mouse is left to “hover” over it, a “tooltip” might appear giving some information about that control and what it is for.

The **middle mouse button** has a special purpose which is to ‘stop’. Whenever it is pressed, whatever the software is doing is stopped. If it is pressed while a move is running, then the move is stopped. If you press it while the rig is already stopping, it will stop even faster. This is a safety feature to aid the user; it does not replace the emergency stop button and should not be regarded as fail-safe. This button is referred to as the “mouse stop button”. It does not work if the mouse is off the program screen or on the border of a window. In certain graphical displays, this button may be used for other purposes. Additionally, the tilde key “~” and the space bar can also function as stop buttons.

The **right hand button** is used to store data. If you want to store data into the program, then pressing the right hand button over a box will store relevant information into that box. This button will be referred to as the **mouse store button** and its exact operation is covered under relevant sections of the manual. As with the “stop” button, this function is used differently in certain graphical sections of the programme. In some cases when it is appropriate, a right click will cause the setup appropriate to that button to appear. Generally “store” works for numerical boxes, and “setup” for push buttons where it is appropriate. In many cases when you right click on a window or control, a pulldown menu will appear with other options on it. Store will often be one of the options if it is applicable. If you know that store is an option for that window, holding the shift key down and right clicking will usually bypass the pulldown and store the appropriate value directly.

Right Clicking on Displays or Controls (Advanced)

In Flair, the standard response to a right click is to store information in that numerical entry control. Sometime a context menu will show and often there will be options as well as a “Store” option. If you Shift and Right Click, then the Store option will be chosen automatically. Right clicking on other controls that do not logically have a “Store” function will often bring up a context menu for that control or area of the screen. If no context menu is applicable and you right click you may find it will bring up the Setup most appropriate to that control or area of the screen. For example, right clicking on the Sync Status will bring up the External Camera Setup which has the Sync controls in it. If you right click on the Run Control Bar, then the layout menu for that control bar will come up. Right clicking on the Target Tracking Mode select button will bring up the Kinematics setup. These are done to give you a quick way to access the setup relevant to the control.

3.1.1. Using the Mouse to Move an Axis

Often the Hand Held Box is used to move an axis. This function can also be performed using a mouse. To move an axis, click and hold the ‘+’ or ‘-’ in the row for any axis name on the main numerical screen (which will described in later sections) to move the axis in a positive or negative direction respectively. As you do this, you will see the axis position value in the **Current Value** column changing. You can also use the button in the middle of the ‘+’ and ‘-’ buttons as a joystick. To do this, press and hold the axis and move it on either side. Pressing, holding and moving it to the left moves the axis in the negative direction while pressing, holding and moving it to the right moves the axis in positive direction.

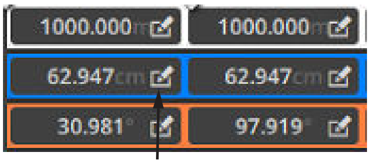
3.2. Numeric Entry

Motion Control is intimately connected with numbers, and you will sometimes have to type numbers in to let the program know different values, such as camera speed. Numerical entry is consistent throughout the program and is done as follows:

1. Select the relevant box.
2. Select the entire text and type the new value.
3. Press either of the Enter keys on the keyboard. If you make a mistake while you are typing, you can correct it with the backspace key (usually marked with an arrow pointing left and directly above the large Enter key). The value is entered once the Enter key is pressed.

In numerical entry, only certain keys are valid; these are the number keys, +, -, the decimal point, the Enter keys and the backspace key. The minus keys and the decimal point key are only valid if the value can be negative or fractional, respectively.

Alternatively, in certain circumstances, the mouse store button can be used to directly enter a numerical value. For example, when you are storing an axis position, you can place the cursor over the box you wish to store the data into, and by pressing the mouse store button, directly enter the data into that box. If a pulldown menu appears, select the **Store** option.



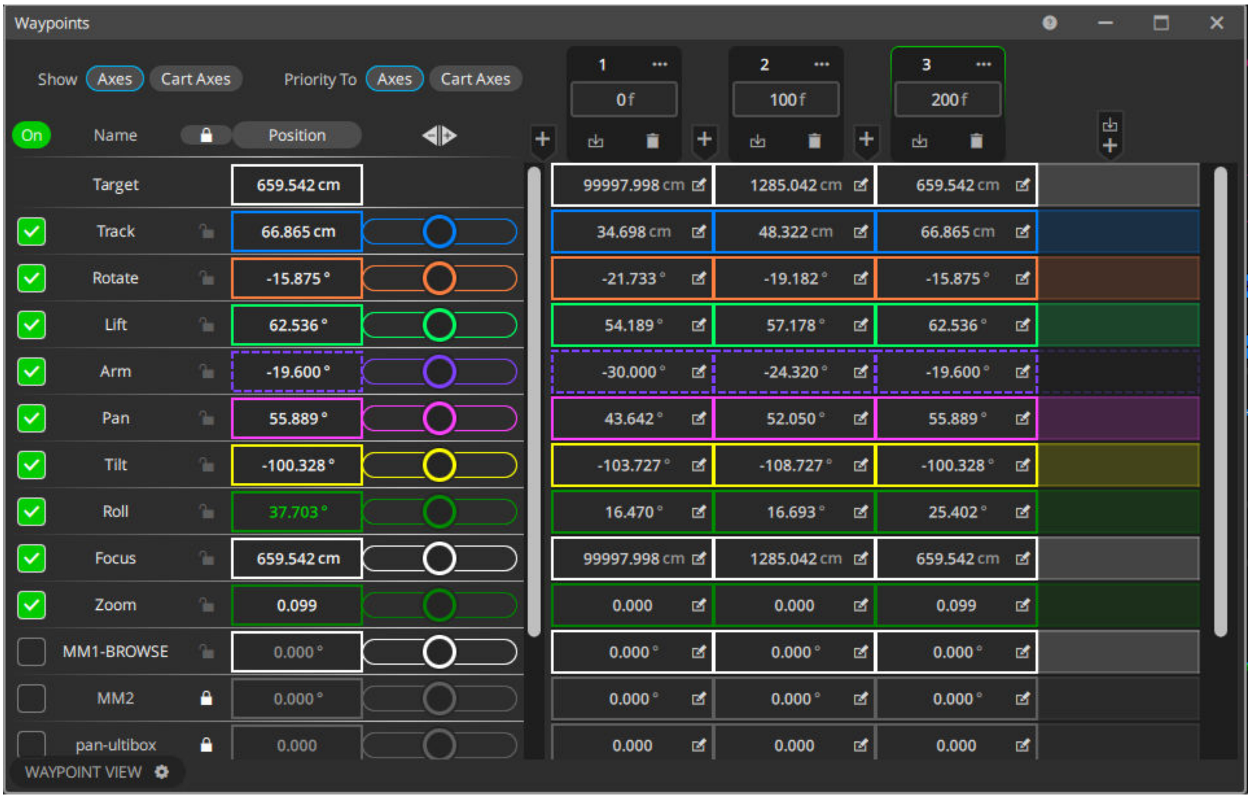
1000.000	1000.000
62.947cm	62.947cm
30.981°	97.919°

Store button

All numeric entry buttons also support a rudimentary calculator function. If you would like to add or subtract an amount from the value, type the amount, and then press the ‘+’ or ‘-‘ key. You may also multiply and divide by an amount by typing the amount and then typing the ‘*’ or ‘/’ key. If the box is for integer values only, then you will not be able to type in a value with a decimal. If the value resulting is out of the range of the box, you will get an error and the original value will remain.

3.3. Waypoint View

When the program first loads, Waypoint view appears on the bottom left of the screen. It contains columns of frame counts and motor positions for those frame counts. Each of these columns is called a **waypoint**, and when the move is executed, each motor is driven from one waypoint to the next at such a speed that it arrives at the position specified in the next waypoint at the correct time. If there are more waypoints, it will then continue on to the next one. When the program first loads, there is only one column, and the positions for this column are blank, after a few columns have been added, the display could look like this:



On	Name	Position	1 0 f	2 100 f	3 200 f
	Target	659.542 cm	99997.998 cm	1285.042 cm	659.542 cm
☑	Track	66.865 cm	34.698 cm	48.322 cm	66.865 cm
☑	Rotate	-15.875 °	-21.733 °	-19.182 °	-15.875 °
☑	Lift	62.536 °	54.189 °	57.178 °	62.536 °
☑	Arm	-19.600 °	-30.000 °	-24.320 °	-19.600 °
☑	Pan	55.889 °	43.642 °	52.050 °	55.889 °
☑	Tilt	-100.328 °	-103.727 °	-108.727 °	-100.328 °
☑	Roll	37.703 °	16.470 °	16.693 °	25.402 °
☑	Focus	659.542 cm	99997.998 cm	1285.042 cm	659.542 cm
☑	Zoom	0.099	0.000	0.000	0.099
□	MM1-BROWSE	0.000 °	0.000 °	0.000 °	0.000 °
□	MM2	0.000 °	0.000 °	0.000 °	0.000 °
□	pan-ultibox	0.000	0.000	0.000	0.000

The topmost line is **Positions** which specifies the position number, and is to let you know which position line you are on. If you have more than about five stored waypoints, then they will scroll left and right, and then this line numbering becomes vital.The **Frame count** line is the second line on the Numeric screen, and frame count numbers are entered here by selecting the entry you want to change with the mouse so that a black square appears around it, and then typing in the new value followed by the Enter key. The next row **Target** is optional and may not appear on your system; it is not necessary for simple moves and is therefore covered later in the manual. The **Position** column specifies the current position of each axis.

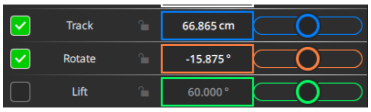
The rest of the screen is filled with axis positions in vertical columns, with the axis name at the top, and a position stored in each waypoint. If there are more than 8 axes, then the other axes can be accessed by scrolling down to inspect or modify their positions. Axis positions are modified in the same way that frame counts are; use the Store button to store the current axis position or simply select the number you want to change so that a black square appears around it and then type in the new value followed by Enter. Normally, positions are entered into these boxes by telling the computer to “store” the current axis positions using the Store button.

3.3.1. Moving the Motors

Flair supports many types of motors and the handlings for each vary slightly. However, they can be broadly categorized as **Stepper Motors** and **DC Motors**. Stepper motors are simpler and generally have only a few wires to control them. DC motors are more complex and have a device on them to tell the system their exact position even if motor power is disconnected. Stepper motors will lose position if turned off and so are normally powered up whenever the system is turned on. DC Motors are usually not powered up until the user turns them on because you can do this without fearing any loss of position etc. The data about “engaged” motors applies only to DC motors.

When the system is first powered up, no motors are engaged, the user has to engage them. The first thing to do is to turn on the power to the amplifiers by letting out the red emergency stop button and pressing the small reset button, this will usually result in a loud click as the switches in the power supply close. The state of a motor is indicated by the On column on the left on the Numeric screen. A blank against the axis name in the On column indicates that it has not been engaged - simply click in the box, and it will change to green with a tick indicating that the user has engaged the motor and wants it to be engaged (under control of the system).

In this diagram, the Track is engaged and under power and the Rotate is not engaged, and therefore is not under power.



☑	Track	66.865 cm
☑	Rotate	-15.875 °
□	Lift	60.000 °

Use the Hand Held Box, on-screen axis sliders or the mouse left and right buttons as above to move the motors. Use the + and – keys to change the speed.

Stepper Motors do NOT typically turn off when you disengage them through Flair. Contact MRMC if you have questions about this.

3.3.2. Programming a Move

Now start programming the move.

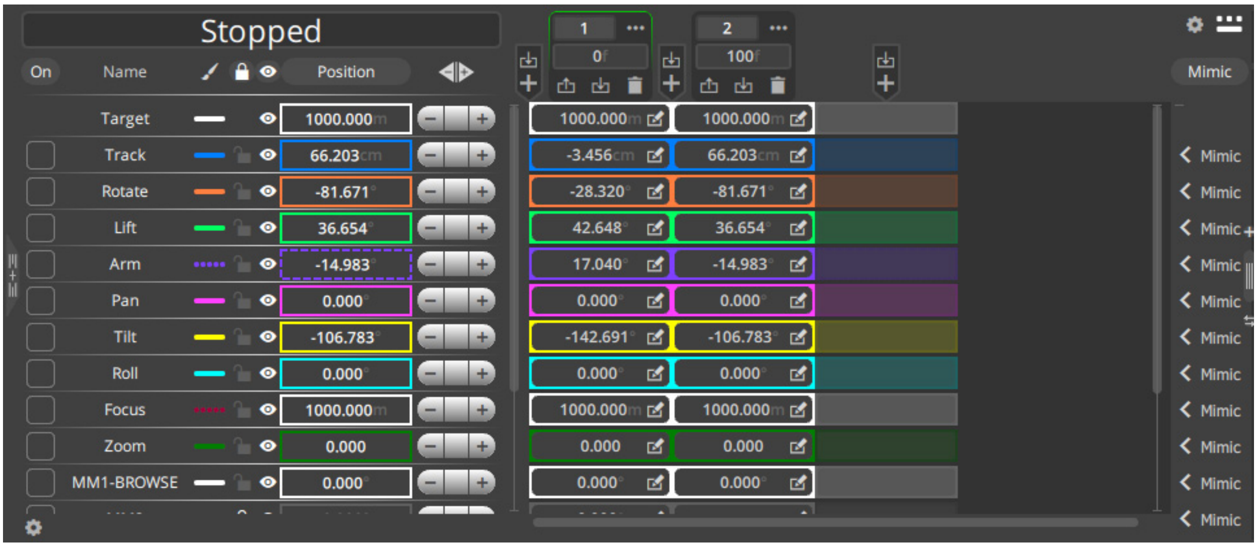
1. Move the camera to the correct position for the start of the move by using the ‘+’ and ‘-’ buttons or left or right mouse buttons for each axis that you intend to move.

2. Click on the **Store Waypoint** () button on the top of the axis positions display. This will store the current positions into the first waypoint.

3. Next, there are two options:

- Click the Add Waypoint (+) button, to add a blank waypoint. Move the axes to the desired positions and then click under the frame count row.
- Move the camera to your second position (a move needs at least two positions), click to add the current positions to a second waypoint to the display.

The new positions will be stored into the second column. If needed, change the frame count of the line to the desired frame count, and then continue adding and storing waypoints until the move is entered. Note that frame counts must always increase through the job, getting larger and larger with each waypoint. A simple two-point move could look like this.



Stopped

On	Name	Position	1 (0f)	2 (100f)	Mimic
	Target	1000.000m	1000.000m	1000.000m	
□	Track	66.203cm	-3.456cm	66.203cm	Mimic
□	Rotate	-81.671°	-28.320°	-81.671°	Mimic
□	Lift	36.654°	42.648°	36.654°	Mimic
□	Arm	-14.983°	17.040°	-14.983°	Mimic
□	Pan	0.000°	0.000°	0.000°	Mimic
□	Tilt	-106.783°	-142.691°	-106.783°	Mimic
□	Roll	0.000°	0.000°	0.000°	Mimic
□	Focus	1000.000m	1000.000m	1000.000m	Mimic
□	Zoom	0.000	0.000	0.000	Mimic
□	MM1-BROWSE	0.000°	0.000°	0.000°	Mimic

3.3.3. Running a Move

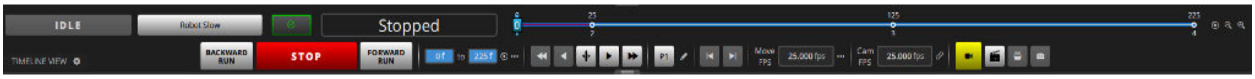
Any programmed move can be run from the beginning to the end, from the end to the beginning or from any time within the move to any other time in the move. The simplest form is to run from the beginning to the end; this is called a Forward Run. There are several ways to do this:

- Click the **Forward run** (FORWARD RUN) button on the Run control bar.
- Select **Run** > **Forward Run** on the top menu bar.
- Click the FWD RUN button on the Hand Held Box.

When one of these buttons is pressed, the move is calculated and, if the move is not too fast, the computer will prompt you with the **Goto** button which means that the current motors positions are not the same as the move start positions, and the motors must move to the start position before they can run the move. (Gotos will take the system from its current position directly to the intended start position, you must know that this is safe to do before you execute the goto)

The Timeline View displays the status of the job and holds the control buttons for running the job. In order to make the system execute a goto, you press the **GOTO** button with the mouse. The motors will then move to their respective start positions, and the message will change to **Ready to run**. In this state or any of the job running states you cannot input into the system.

At this point, you can run the move. The **GOTO** button changes to **SHOOT**. Click the **SHOOT** button, the motors will start moving and the slider will also start moving to show the frame count as the move progresses. When the move completes, the motors will come to rest, and the system will return to its normal input state.



You can also run the move from the end back to the beginning. The procedure is basically the same as the above, click the Backward Run button, click **GOTO** if you are so prompted, and then **SHOOT**.

If the computer calculates that the move is too fast for the motors to run at that speed, it will suggest a longer overall frame count for the move or a slower camera speed. You can either apply the increase in frames, or modify the move (for example, reduce the camera speed or reduce the distance to be travelled).

The various buttons present on the control toolbar allow you to inspect the move by going to a waypoint or going to a frame etc.

3.3.4. Modifying a Move

If you are not satisfied with the move, there are many ways in which the move can be modified. You can insert, delete or modify waypoints. Deleting waypoints is done by clicking the delete (🗑) button on top of the waypoint column. Waypoints can also be inserted before of after any waypoint by clicking the add a new waypoint (⊞) button where you want to add a blank waypoint. You can also insert a copied waypoint by using the copy waypoint button (⇪) and then the insert waypoint button at the desired position.

A good way to change a move is to move the system to one of the waypoints is by selecting **Goto** → **Position** from the top menu, and then typing in the number of the position you want to go to. The system will then say Ready to Goto and will move to the desired position when **GOTO** is pressed. Once at the waypoint, correct the position(s), re-store the waypoint and then re-run the move. This is the preferred method as adding and inserting waypoints can make the move too complicated and can spoil the smooth shape of the move. If an existing waypoint is wrong, move to that waypoint and make it correct, then select it and re-store the new positions.

The system has numerous advanced features that allow you to change the move in other ways and to affect the shape of the curves, but the above guide covers the basics, and should allow you to do repeat or multiple passes of the move you want.

3.3.5. Recording a Move

Once the move is to your liking, you are ready to record it. Switch the Rehearse toggle (🟨) to Record (🟥).

Film

With a film system, you have to engage the camera which is done just like the other axes; simply click the **Frame** button at the top of the numerical display, the camera will engage and zero itself. Then toggle Rehearse (🟨) to Record (🟥) on the Run Control Bar and run the move. At the beginning of the move, the camera will be started, and then the move will be shot with the camera running throughout and at the end of the move, the camera will stop. If you have electrically controlled take ups, these should be tensioned before the move is shot; press the button at the bottom of the side menu that says TUs Slack and it will change to TUs Tensioned and the take ups will turn on.

HD Video

Flair usually does not directly control HD video systems or their recorders. However you can use the input and output settings from the Flair hardware to start and stop these cameras and to synchronize to them.

The system has numerous advanced features that allow you to change the move in other ways and to affect the shape of the curves, but the above guide covers the basics, and should allow you to do repeat or multiple passes of the move you want.

Video Recording System

On a video system, a move is recorded by pressing on the button on the Run Control Bar that says VTR Off, and it will change to VTR Record.

When the move is run, you will be asked for a starting timecode for the recording. This is typed in, and then when the move is shot, the VTR will be controlled to record the move onto tape.

Chapter 4 **Flair Systems**

Flair supports a wide range of systems ranging from a single pan and tilt head running from a laptop to a 1.5 ton Cyclops running at 6 feet per second on steel tracks. The way in which Flair operates when started depends to a large degree on the initialization file Flair.ini found in the install directory. This file defines how Flair handles the required real-time aspect of motion control as well as defining the over 28 different rigs that are supported by the mathematics.

Details on the different configurable values in the configuration file are covered in detail in the Appendix.

This chapter contains additional information about Motion Control and how Flair controls motors which will be vital for Technicians and Owners, but not necessarily useful for people who just want to use Flair to shoot visual effects shots. It provides some history of motion control as provided by MRMC and this may help you to fully understand the line of products and how they are used.

4.1. Motion Control Basics

The basic premise of motion control is that motors can be controlled to perform an exact known motion. In order to do this, the controlling system must be able to direct the motion of these motors in a predictable exact fashion. Flair can work with a variety of control systems and the system configuration of Flair directs exactly how this will be achieved.

4.2. Flair 1991

The original release of Flair used a dedicated processor to control a series of linked axis boards, and with this Flair could control up to 32 axes, with up to 4 axes on each board. This style of connection is still in use today and is referred to as Link axis boards.

4.3. Real-Time Control

The dedicated processor was required to ensure that no matter what the user was doing, the positions of the motors were being updated in a known predictable fashion. This is type of timed operation is known as Real-Time control and is a stringent requirement for proper motion control. Windows itself as an operating system is not able to operate in this fashion, but Tenasys's Intime™ Operating System runs underneath Windows allowing users to see the Windows they are used to, whilst providing exact Real Time control. The use of Intime has allowed Mark Roberts Motion Control to take advantage of the abundance of powerful and economical PCs on the market today without losing the required precision control.

4.4. INtime Hardware Control

For INtime to run in a real-time fashion, it takes control of the processor and allows Windows to use it when all other real-time tasks are serviced. INtime can also take control of other devices on the computer such as serial ports and network ports. When under INtime's control, these devices can operate in a real-time fashion, but are not accessible to standard Windows.

4.5. RT10

The first system to run Flair using INtime was an industrial PC computer which allowed direct access to a control board called the Root Interface Card (Ric). The Ric together with the PC was housed in a compact grey case with all the required interconnections and was called the Real Time 10 (RT10). Many of these systems were installed around the world and they are still in use today and are fully supported by Flair up to version 7.0.

4.6. Network Direct

The original Link system of axis boards and motor control works well and is still in use today, but in order to support simpler and more readily available systems, Mark Roberts Motion Control developed the Network Direct interface which allows users to connect together axis boards using off the shelf networking equipment. Network Direct itself uses a library of UDP (User Datagram Protocol) packets to exchange the needed information between the control computer and the motor control boards known as Nodes. With Flair version 6.2 and above, there can now be as many as 32 nodes each with up to 16 axes (if the node supports that many) giving a total possibility of 256 axes.

4.7. Basic Network Direct

The basic form of Network Direct can actually operate without INtime running on the Windows PC, it uses a buffer of control information so that if Windows is unable to provide update information rapidly enough, the buffer can continue operation for a few moments until Windows can catch up. Whilst this works very well for simple systems without strict motion control requirements, it does not provide exact camera synchronization and there is necessarily a slight lag in the motor response. However, it uses the standard windows network hardware and can even work wirelessly. You may even operate axis boards on the same network as your home or business computers, however this is not recommended.

4.8. Network Direct INtime

In order to provide the easy control and connection of Network Direct with the precise real-time control of INtime, Network Direct was adapted to be able to run under INtime so that exact predictable control of motors with full motion control capabilities could be realized. Specifically, Network Direct INtime allows for full camera synchronization with Flair, providing frame accurate, repeatable motion control. When run in this fashion, the network port on the PC is passed to INtime for real-time control and is not available to the PC for regular use. In this case, the network equipment should not be mixed with a home or office network. It may operate under these conditions but heavy network traffic could cause the system to stop working reliably.

4.9. RT12 and up

The RT10 used only the original link style communication, but also provided much needed inputs, outputs and handwheel control for motion control systems. With the need for a Root Interface Card to communicate to the axis boards replaced by Network Direct, there was still a need for the other interfaces provided by the RT10. The functions are provided by the continuing line of "RT" products with the first being the RT12 and more recently the RT14. These are also connected to the control computer by network cables and are part of the Network Direct family of products. The RT12 and RT14 still support the original link control protocol and this is still the preferred communication method when working with larger motion control systems; this also allows for upgrades to Flair 7.0 for even very early Flair motion control rigs. Flair version 6.0 and above does NOT support the old transputer-style root box.

4.10. High Speed Industrial Robots

To provide for even faster camera motion, Mark Roberts Motion Control developed an interface to High Speed industrial robots, including the line of Staubli® and Kuka® robots. The Kuka robots are controlled in a similar way to the Network Direct nodes, but have a specialized protocol, the Staubli Robots are controlled over an industry standard version of Ethernet called EtherCAT®. When High speed robots are used, other required motors such as focus motors or track motors are usually run from Network Direct INtime to ensure perfect synchronization. Flair also supports running other EtherCAT capable motor controller combinations, contact MRMC for supported models.

4.11. Dual Systems

With Version 6.0, the ability to run both Network Direct and Link style boards was added. This means that you can run a link style rig controlled by a line of axis boards and also control Network Direct style Nodes to run other axes and model movers. Previously you had to add the model mover boards in line with the link axis boards and this required re-configuring the numbering of the boards and was best done well before a shoot. Now with the addition of a Dual system, you can easily add Network Direct style boards and run model movers in just a few minutes. In fact the RT12 and RT14 contain a board which runs a program called Deric as it is a Dual Ethernet Root Interface Card, and the Link connection is connected to the Rig/Head output on the back of the RT Unit. Note that running the old Link- style axis boards from the RT12 or 14 does require that the system run under INtime as the timing requirements are too precise for regular Windows Operating System.

4.11.1. Available Boards and their Use

(Node and Board are terms that are used somewhat interchangeably.)

> Note
>
> The designation **_II** refers to an improved processor. Many of the boards were released with the previous processor and these still perform fine, however the newer processor allows for some more memory and requires a different program to run on it.

Transputer Axis Board: This was the original axis card that was developed by MRMC in 1990. It is easily identifiable by the large brass coloured square chip. It ran a variety of programs, but the most common was called Axis.btl and it could run up to 4 servo axes.

DSP Quad Axis Board: This was the first generation of replacement for the out-dated Transputer processor using a line of Digital Signal Processors that were faster and had a wider range of capabilities than the transputer axis card. The motor control was improved and these boards can run 4 or 8 motors either steppers or servos and even serial lens control depending on the software loaded. The program most commonly loaded onto these boards when run on a link is Quad6Axis.btl, when run from Network Direct, it is Quad6Ether.btl. The second generation of Quad Axis cards run with the improved Delfino processor and have the designation “_II” added to differentiate, but perform much of the same functions: Quad6Axis_II.btl and Quad6Ether_II.btl. There is also a version of the program called Oct6Ether.btl that allows the card to run 4 servos and 4 steppers simultaneously.

DSP Ulti Axis: Developed for running an entire pan tilt head all by itself, this board runs up to 12 axes, and has 6 integrated servo amplifiers and can also run 3 stepper motors and 3 serially controlled lens motors. Variation on this board can also run Controller Area Network (CAN) amplifiers. The program loads onto this board is most usually Ulti6Axis.btl and Ulti6Ether.btl with the same _II designation for the Delfino (Advanced) version.

DSP Root Interface Card (RIC): Originally developed to work when attached to an industrial PC, this board provided the interface between the PC and the world of motion control providing: link connection to axis boards, serial connection to a hand held box and a host of other interfaces including timecode and triggers. In that mode it was loaded most commonly with Ric.btl. Now being used with Network Direct and providing many of the same function, the Ric has become the Deric (Dual Ethernet Ric) and is housed inside the RT12 and loads with Deric.btl or with the improved RT14 it loads Deric_II.btl.

DSP Octo: Allowing for up to 8 stepper motors, this board is most commonly used with stop frame animation systems. It runs Octo6Ether.btl.

DSP Hex: Only available in the Delfino version and thus using the _II designation, the Hex board can run up to 3 servo motors and 6 stepper motors and have other improved features as well as an on board Ethernet adapter from improved speed and reliability. Hex6Ether_II.btl.

MSA20 & 21: Generally used as a stand-alone controller for a head, the MSA21 can also be used in conjunction with an RT12 or 14 to provide handwheel style mimic inputs. In this case it would run a program called Mimic6Ether_II.btl as a Network Direct version. The earlier MSA20 can also be loaded as a link style axis board to provide the same functionality but only with the older RT10 system.

Chapter 5 **Setups**

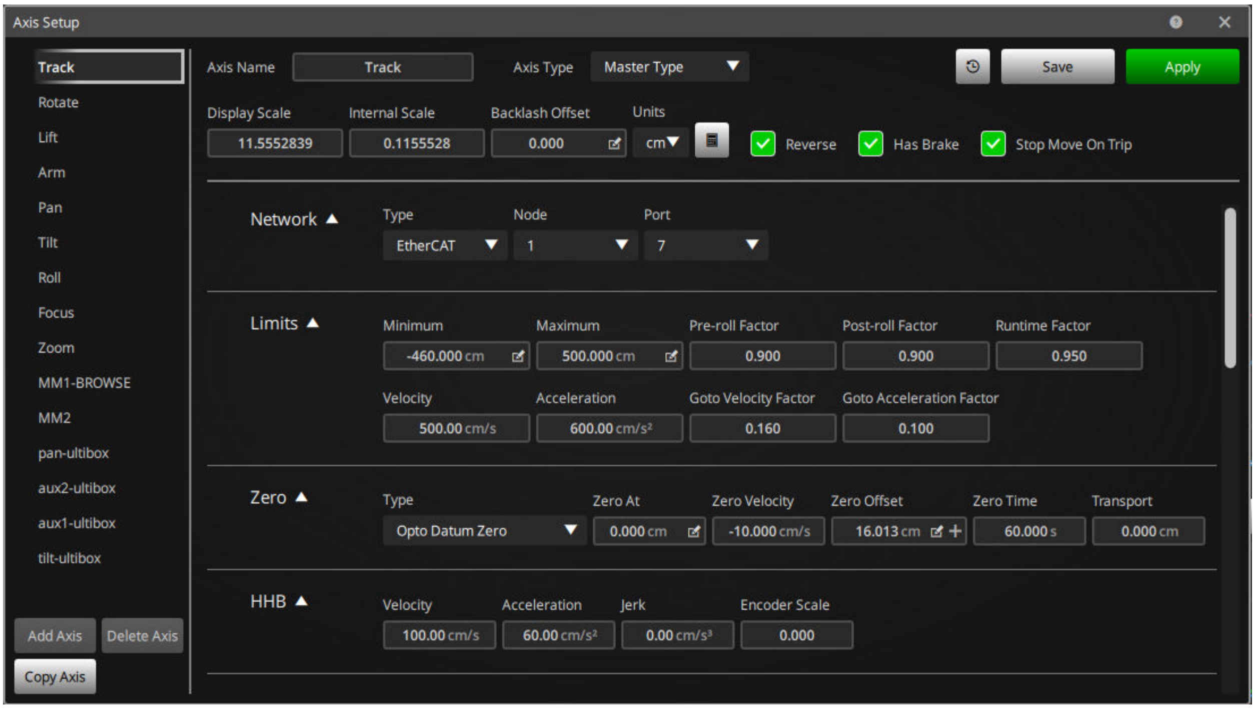
In order to fully understand the terms and settings in this chapter it is necessary to have an understanding of your system. Some settings are fairly easy and some require a profound understanding of the system to fully appreciate and adjust. If you have any uncertainty on what a particular setting should be, read this manual carefully and contact MRMC if you have any questions. It is possible to damage the system or render it less safe by incorrectly setting some of these values and they should not be adjusted without a full understanding.

5.1. Reset Settings

On several of the dialog boxes, you will find the reset (⟲) button. This button allows you to read the default setting for that setup from the disk to remove any temporary changes you may have made. Note that using the **Save** button sets the default.

5.2. Axis Setup

Each axis has its own setup which can be changed by the operator. When changes are made, they can be saved permanently to the computer hard drive by using the **Save** button on the screen, they can be applied to the axis by clicking the **Apply** button on the screen, or the changes can be ignored by pressing the Close button on the right corner of the dialog box. (Note that this does not apply to motor tuning as these changes are applied immediately for safety reasons). Since most systems have many axes, you can switch between the setups for each axis by selecting the axis from the left panel. The axes are also identified by a number, and this is displayed in the top line of the Axis Setup display along with the axis name.



Axis Name: Track
Axis Type: Master Type

Display Scale: 11.5552839
Internal Scale: 0.1155528
Backlash Offset: 0.000
Units: cm
☑ Reverse ☑ Has Brake ☑ Stop Move On Trip

Network ▲
Type: EtherCAT
Node: 1
Port: 7

Limits ▲
Minimum: -460.000 cm
Maximum: 500.000 cm
Pre-roll Factor: 0.900
Post-roll Factor: 0.900
Runtime Factor: 0.950
Velocity: 500.00 cm/s
Acceleration: 600.00 cm/s²
Goto Velocity Factor: 0.160
Goto Acceleration Factor: 0.100

Zero ▲
Type: Opto Datum Zero
Zero At: 0.000 cm
Zero Velocity: -10.000 cm/s
Zero Offset: 16.013 cm
Zero Time: 60.000 s
Transport: 0.000 cm

HHB ▲
Velocity: 100.00 cm/s
Acceleration: 60.00 cm/s²
Jerk: 0.00 cm/s³
Encoder Scale: 0.000

Copy Axis: If you wish to copy the settings of another axis into this setup, you can use this button to select an axis. When you copy an axis, you get the option to copy the basic settings with “Copy Settings” and also the option to copy everything including the Network, Board and Port settings with “Copy All”.

5.2.1. Basic settings

Axis name: This is a text area, and you can type any name for the axis. Note that changes to this field will only be permanently recorded if the Save button is pressed. It is strongly recommended not to use spaces in axis names. If you want a gap in the name, then use an underscore “_”.

Type: There are several types of axes. The two most commonly used ones are Target Tracking and Independent. This field sets up how the motor is controlled.

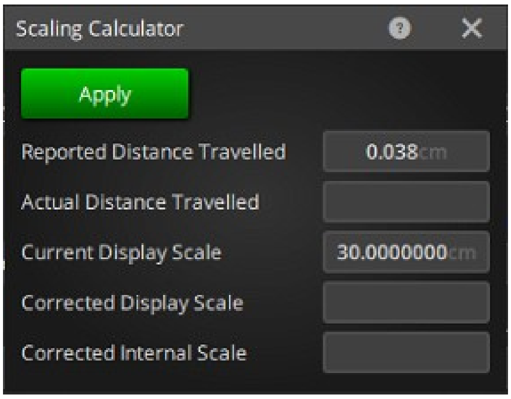
- **Independent:** An independent axis will simply move exactly as it has been programmed from point to point. Its motion is programmed in numbers and it will execute that motion regardless of other axes.
- **Algorithmic:** This mode allows a motor to be controlled by the position of another motor or encoder input through a relationship table. (See below – “Lookup Table under Control Tab”)
- **Target Tracking:** A Target Tracking axis is one whose position affects where the camera is or in which direction it is looking. When you are in the program's Target Tracking mode, the position of a target tracking axis is determined not directly from the entered numeric positions, but from the intended position of the camera and target. Typical Target tracking axes are Lift, Pan, Tilt etc.
- **Master Type:** This is a specialised axis which behaves just like an independent axis, but whose position is important to the Target Tracking geometry because it affects the camera’s position. Typical master axes are Angle and Extend.
- **Focus Type:** This is used to identify the focus axis to the system so that it can be used as a computer-controlled focus when a lens has been fully set up; see the section on setting up a lens for more details. There may be only one focus type axis.
- **Zoom Type:** This is used to identify the zoom axis to the system. A zoom's motion can be controlled in special ways so that the focal length is known or so that the apparent field size changes in a linear fashion. There may be only one zoom type axis.
- **Iris Type:** This is used to identify the iris axis to the system to change the stop on the lens in a specified manner so that it can be used to adjust for changes in exposure due to changes in the camera speed. You are also allowed only one iris type axis.
- **FDX type:** Used to be used to control the exposure by changing the shutter angle and could be used automatically to control cross fades. Not currently supported in Flair.
- **Shutter:** Used to identify an axis that has direct linkage to the shutter angle of the system. It can be programmed normally as an independent axis and also can be used to vary the exposure and thus adjust for changes in camera speed.
- **Min Auxiliary:** (Stands for Miniature Auxiliary) This axis can be used as an additional way to adjust a Target Tracking move, and this subject is covered further in the Advanced Target Tracking section in Chapter 10 Target Tracking.

- **Copy:** An axis can be made to “copy” the motion of another axis at all times. This has been used to run the focus on both lenses of a 3D system, and also to run two motors on a lens since the lens was very stiff and one motor was not enough. This is an advanced feature and you should contact MRMC if this is required.
- **Live:** Not A type of motor that is only even controlled by a user input and does not execute moves in the system.

Display Scale: The motion of a motor is monitored or controlled as a number of counts. The scaling factor is used to convert counts into usable units, such as inches, degrees, centimetres etc. The correct value can be set up by zeroing the axis directly at a known point, moving it through a known distance, then adjusting the display scaling until the distance reported on screen matches the measured distance. For example, point the camera at something, zero the pan directly, and then turn the camera right around until it is looking at the same target again. This should give a position read out of 360 degrees. If it doesn't, then adjust the scaling factor until it does. The Internal Scaling can be calculated from this. (See Display Scale Units below). It is vital that this value is set correctly for advanced system use. Consult MRMC for details or for the correct value for your system. If you scale an axis so that this value is in hundreds, you may get some jittery effect on very slow moves, and this value is best somewhere between 2 and 100. If you are getting a very high number, then choose a different measurement of motion to reduce the scaling value to a more fitting number.

Internal Scale: The internal scale of the axis is used to convert the motor position into standard scientific units of Metres and Radians. See Display Scale for more information.

Calculate (▣): This button will pop up a dialog box that allows you to correct the axis display scale if it is incorrect. Once the axis has been moved an exact known amount, pop up this box and enter the distance the axis has actually travelled (not the distance the computer says it has travelled). When the Calculate button is pressed, the new scaling will be calculated and will appear on the dialog. If the data looks correct, click the **Apply** button, and this new value will be entered into the Axis Setup, but will not take effect until that setup is accepted by the user.



Scaling Calculator

Reported Distance Travelled 0.038cm

Actual Distance Travelled

Current Display Scale 30.0000000cm

Corrected Display Scale

Corrected Internal Scale

Units: This is a pull-down list of various units types that when selected will calculate the correct Internal Scale based on the Display Scale. Once the Display Scale has been correctly worked out so that the motor position is displayed in the desired units, select the pull down menu to the right of the Display Scale Units label, and select the units you have scaled the axis in (supported units are Metres, Centimetres, Millimetres, Inches, Feet, Degrees and Radians) When one of the units is selected, then the Internal Scale is worked out based on your selection and the Internal Scale window will be disabled. Note that the Internal Scale will be automatically updated if you change the Display Scale. If you have scaled your axis in an unsupported unit, then you will have to work out the correct value to convert the axis position into Metres or Radians, contact MRMC for assistance.

Reverse: This switch can be used to reverse the direction of the motor. When an axis is moved, its position will increase or decrease; for target tracking, it is very important that axes are considered to move in the correct direction. For pan and rotate axes, positive movement is considered to be to the left looking from the back of the camera. For a roll axis, this is anti-clockwise looking from the back of the roll. For any lift axis, a positive movement is that which moves the camera higher. For a track axis, positive movement is along the track in the direction that the camera is looking when the pan and rotate are correctly zeroed. For a focus axis, forwards is moving away from infinity focus.

Has brake: This is a toggle which specifies whether the axis has a brake on it or not. This will not affect whether or not the brake is released when the axis is switched on. It is used to tell the software that the axis should not move when it is switched off. If your control system has a software-controlled e-stop (emergency stop), and if the axis is detected to be moving when the brake should be on, it will drop the e-stop. This is only available on systems that have this specific hardware. Contact MRMC if you have any question about this.

Stop Move On Trip: This is a toggle which determines if the move should be stopped if this axis trips out or loses control during a move. With large axes, it can be unsafe to have part of the rig keep running and it is recommended that this option be checked. However, a situation in which, for example, a small model mover stops working, is not serious enough to warrant stopping the shot. It is up to the user to set this flag correctly for any given shoot. An axis assigned as a Target Tracking or Master Type will raise a warning if this flag is not set.

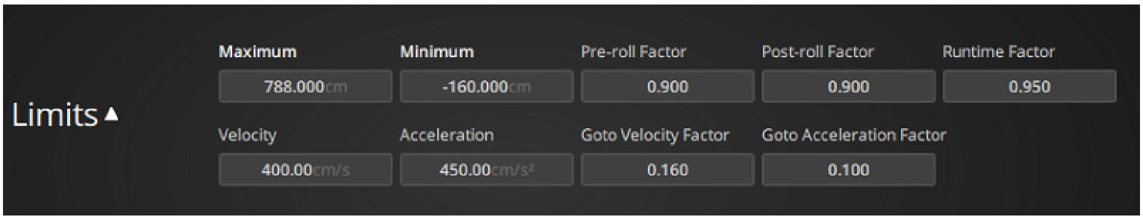
5.2.2. Network settings

Type: Flair can communicate to different types of networks, and this setting determines which network the axis is on. The options are Link, Network, Robot, and Dummy and EtherCAT. Prior to Flair v5.0, all axes were on a **Link** network of axis boards which talked to each other in a "daisy chain" configuration. With the advent of Network Direct, axis boards (Nodes) can be on an Ethernet **Network** and a central control unit talks directly to each axis board. Boards can also be controlled through other high speed communications and since these are typically on an industrial robot, this type of connection via "UDP" is called **Robot**. **Dummy** means the axis is not a real motor and is not connected to the system. **EtherCAT** is used for EtherCAT network controlled motors including the line of Bolt high speed robots. Sometimes an additional or auxiliary axis is not connected and you can assign it to dummy and errors will not keep appearing if it is tripped out or not working.

Node: This is the Node or physical axis board that the axis is assigned to. The Node is 1, and it goes up to the maximum number of boards on the system. 0 is not valid. Node numbering has changed in Flair version 6.0 and you may need to consult MRMC. If your system is Network Direct, then the Node number is the number listed down the left hand side of the list of Network Board in the Network Setup display (for more information, see Network Setup on page 263). With a Link system, it is normally the number of the board in the order that it is connected with the other boards since they are only ever daisy chained one to the next. With Dummy network, the Node and Port are not important. With EtherCAT, there is ONLY one node and all EtherCAT devices should be assigned to “Node 1”.

Port: This is the port number on the axis board to which the motor is connected. It can be 1 to 16. Read your documentation if there is any problem, but usually this will be obvious, clearly documented or already set up. Some Nodes/Boards can support four axes and some can support more. You will have to consult MRMC to find out now many axes your board supports. If you assign an axis to a port higher than the board supports, it will usually be ignored. With EtherCAT, the nodes must be numbered in the sequence they are connected on the EtherCAT connection and this should not be changed except under guidance from MRMC.

5.2.3. Limits Settings



Limits ▲

Maximum: 788.000cm

Minimum: -160.000cm

Pre-roll Factor: 0.900

Post-roll Factor: 0.900

Runtime Factor: 0.950

Velocity: 400.00cm/s

Acceleration: 450.00cm/s^2

Goto Velocity Factor: 0.160

Goto Acceleration Factor: 0.100

Velocity: This is the maximum speed the computer will allow the motor to go. It is in display units per second. If your motor is a rotational motor scaled in degrees, then the velocity is in degrees per second. Any motor will have a top speed; this is governed by the motor, its supply and the mechanics it drives. The Axis velocity should be set to the maximum speed the motor (or your nerves) can take. In general, the user will have been supplied with the approximate maximum speeds of the axes, and these can be typed in directly. More detailed set up is covered later.

It must be noted that changing this number will not magically make the motor go faster; it is only a number which is used to work out how fast to do Gotos, and is used in checking that a job will not go too fast. It does not affect the performance or behaviour of the motor in running a move in any way. If you attempt to execute a move that requires a higher axis velocity than the entered value, the computer will refuse to run the move, and will suggest a higher number of frames for the move. If you force the system to move the motor more than about 15% faster than this limit, the axis will be stopped and the move will also stop.

Acceleration: As above with axis Velocity, except that the units are display units per second per second. If an axis accelerated at its maximum acceleration from a standing start, after one second it would have reached a velocity numerically equal to its acceleration. For example, if your motor's maximum acceleration were 100 centimetres per second per second (100cm/s*s), then after a second's acceleration, it would be travelling at 100 cm/s. The value is in fact usually similar to the maximum velocity. Detailed setting up is covered later. As with the axis velocity, this value is used in Gotos and also in checking a job to see that it does not exceed the capability of the motor; it does not affect the performance of the motor in running a job. If the computer determines that the acceleration required to execute a certain move is greater than this value, an appropriate message will be produced and the move will not be run.

Maximum and Minimum (limits): The allowed range of travel of any axis can be set up so that it does not exceed certain limits of travel. There are two limits, one is the minimum limit and the other is the maximum limit. Obviously, the minimum limit is the lowest number to which the axis can go and the maximum limit is the highest limit to which it can travel. These limits are used when controlling the axis from the hand held box and also are checked at run time to see that the move does not exceed the allowed limits of travel. These limits are NOT checked when the axis is mimic controlled in Encoder Position mode which is described below. You can also disable the limits by deleting the value which is done by pressing the Delete or Backspace key (¬) in that box. These limits can also be overridden for manual control by using the Limits Override key on the Hand Held Box.

Factors: There are five different Factors. Each of these is used to limit the maximum velocity or acceleration of the system at different times. Allowed values are 0.001 to 1.0. The value is a straight multiplication factor. For example, if you set the **Goto Velocity Factor** to 0.5 then the maximum speed reached in a Goto will be limited to 1/2 of the maximum velocity.

Runtime Factor: This factor is the factor used when checking the job. The basic idea of this is to allow the user to set up an absolute maximum velocity in the axis velocity field and then to be able to say “Yes, I know it WILL go that fast, but I would rather limit it to say '90%' of that speed for normal jobs, knowing that I can change the factor to 1.0 when I need full speed”. This factor also controls the maximum acceleration allowed in running a job. Note that part-runs do not check maximum velocities and accelerations unless you select that option in the Miscellaneous Setup display.

Goto Velocity Factor: This controls the fraction of the axis velocity that is allowed when performing Gotos or Move Tos. If these are performed at maximum speed and acceleration, it can worry the client (and the owner), and usually these are best done at about 1/3 speed and acceleration.

Goto Acceleration Factor: This controls the fraction of the axis acceleration that is allowed when performing Gotos or Move Tos. If these are performed at full acceleration, it can worry the client (and the owner), and usually these are best done at about 1/3 speed and acceleration. Note that this value is also used in single stepping a job as the rig will “Goto” the next frame position automatically. This movement can sometimes be a little too fast and can cause the rig to shake. By reducing the goto acceleration factor on the most afflicted axis, this effect can be handled. With High Speed Robots, the standard acceleration is extremely high and a Goto Acceleration factor of around 0.1 is recommended.

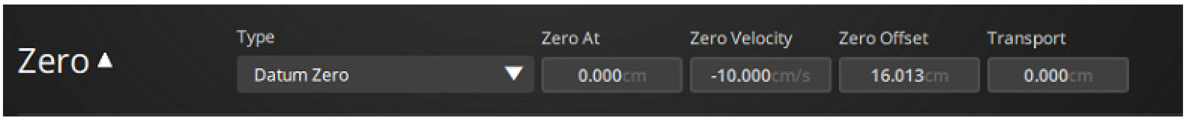
Pre-roll Factor: This and the Post-roll Factor are different from the above in that they only affect the acceleration, and not the velocity. If you have a move that starts with a velocity, either because there are no fairings or because you are part running a move, then the motors have to “pre-roll” before the actual move itself starts. This means that when the move starts, they have to be travelling at the right speed, so they will run back and then forwards in order to achieve the right speed. The velocity they end up is governed by the move, but the acceleration they use to attain the speed is controlled by this factor. If the axis pre-rolls too violently, then reduce this factor, it takes too long, then increase this factor (or set the maximum acceleration correctly). Typical value is usually between 0.5 and 0.8.

Post-roll Factor: As above, but this controls the end of the move or the end of the part run. It also controls the stopping speed when you stop the motion in the middle of a move. If you press the stop button, the system will use the maximum acceleration reduced by this factor to slow down and stop. If you press the stop button twice, then the system will use the full maximum acceleration to stop. Typical value is usually around 0.75. A high Postroll factor is recommended since this only happens after the rig has finished the section you are actually shooting and it is safer to stop quickly even if a little suddenly.

WARNING

For both the pre-roll and post-roll, if these values are set too low i.e. 0.1 or such, then the rig may do huge pre- and/or post-rolls, possibly exceeding its limits of travel and/or colliding with the set. These factors should initially be set high (at least 0.5-0.8) and only reduced when the rig is fully set up.

5.2.4. Zeroing settings



Type: Datum Zero ▼
Zero At: 0.000cm
Zero Velocity: -10.000cm/s
Zero Offset: 16.013cm
Transport: 0.000cm

Type: There are several zeroing styles which the user may need to know about. For a fuller description, seeChapter 18 Zeroing. There are two basic versions, either Direct Zero which sets the axis’s current position to zero, (so it is up to the user to position the axis correctly), or any of the other methods which seek a reference point, and then move a known distance away from it.

Zero at: Once an axis is zeroed, its position is set to 0.0. If the best position for it to zero at is not necessarily zero, then a value can be set to its position once it has completed the zeroing process. This can be useful for conditions when axes are offset, seeChapter 18 Zeroing for more details

Zero Velocity: The speed at which the axis will seek its reference point (The direction is set by the sign). This should be set low at first. It is in display scaled units just like the maximum velocity.

Zero Offset: Once a reference point has been properly located, the axis will move a given distance from this point back to its normal zero position; this is known as the Zero Offset. When the store button is clicked in this box the axis current position is added to the value currently displayed. This is useful for making small adjustments to an existing zero. Alternatively, you can set this value to 0 first and then use the store button to store the current axis position.

Transport: The “Home” or “Zero” position of a rig is usually a good place to leave it at the end of the day when it is switched off. However some rigs should be left in a different position for transport where the head is tucked in or the arm lowered. This value is the position to which you would move the rig so that it is ready to be turned off and transported. There is a feature that allows you to move the rig to the Transport positions. This value is typed in or stored using the right hand mouse button. For example, you might want to tilt the camera platform down for transport. So the normal zero position would be 0.0 looking straight forwards, and the transport position would be -90 looking straight down.

5.2.5. Hand Held Box (HHB) Settings

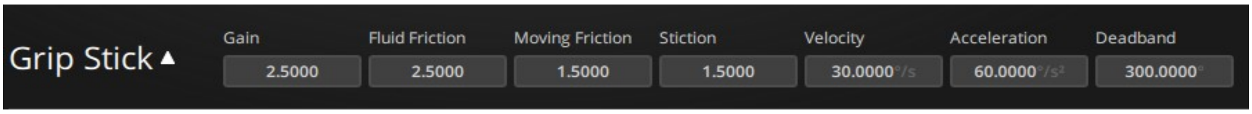
Velocity: This controls the maximum speed of the axis when being controlled by the hand held box. The hand held box should not normally drive the axis at maximum speed, about 50% is useful, however if you are doing a lot of close up work, then this value can be dropped to make exact positioning easier. It should initially be set so that the top speed on the hand held box is safe, and precise positioning work can be done at the slowest speed. This value should NEVER be higher than the axis maximum velocity and with high speed robots, it should not be more than 1/6th of the axis maximum velocity.

Acceleration: When an axis starts moving on the hand held box, it will accelerate up to speed, the rate of its acceleration being governed by the HHBox Acceleration. This value is usually about the same as the HHBox Velocity, higher for light axes and lower for large or heavy axes. Try moving the axis around with different settings, and you will soon get the feel of this. If the setting is too high, the axis will lurch, if the setting is too low, it will take a long time to get going and to stop.

Jerk: Jerk is the factor that controls the degree by which acceleration can change, and as such can be used to produce smoother hand held box motion. If this value is set to 0, then it is ignored and the normal hand held box calculations are used. If a value is entered, then the acceleration will be eased in and out much the same way that the velocity is normally eased in and out. Due to this the movement will be smoother, but it may also take longer to speed up and slow down. A value that is about three times the axis acceleration is a good starting point. If the axis takes a long time to get started and a long time to slow down, then this value is probably way too low.

Encoder Scale: An axis controlled by hand held box can move at a faster or slower rate than the encoder that controls it. The desired ratio of velocities between the controller and the controlled is entered here. That is, if you want the controlled axis to go twice as fast, then double this number. If you want it to go backwards then change the sign. These numbers must not be changed while the axis is moving. For an axis that is controlling itself, this value is automatically set to 1.0 by the software.

5.2.6. Grip Stick Settings



Grip Stick ▲

Gain	Fluid Friction	Moving Friction	Stiction	Velocity	Acceleration	Deadband
2.5000	2.5000	1.5000	1.5000	30.0000°/s	60.0000°/s²	300.0000°

This section allows you to alter the tuning values assigned to that axis for use with a Grip Stick. There are four tuning values: **Gain**, **Fluid Friction**, **Moving Friction** and **Stiction**. There are also settings for the Deadband and the maximum **Velocity** and maximum **Acceleration** to be used while under Grip Stick control.

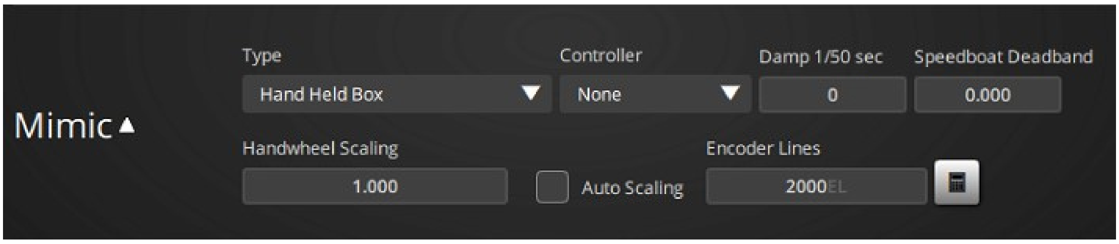
Gain: Gain amplifies the pressure applied to the Grip Stick so that full acceleration can be produced easily. A low value in this field will make the axis sluggish, while a high value will make it take off too fast and not respond to small changes in force on the Grip Stick. The Friction and Stiction values are used to slow the system down if pressure is released or reduced. The Stiction effects how the axis starts moving from stationary, the moving friction is a constant effect to slow the axis down and the fluid friction is a similar effect, but is proportional to the speed of the axis.

Acceleration and Velocity settings: Since the operation of a Grip Stick is different from the hand held box and also different from normal axis use, the Grip Stick operates with its own Acceleration and Velocity settings. During Grip Stick operation, the axis will follow the limits of velocity, acceleration and travel, but the major difference with a Grip Stick is that the axis responds to pressure on it as a demand for acceleration, whereas a normal joystick produces a demand for velocity. You will really have to try it out to see how it responds. Contact MRMC for details and costs. Grip Stick operation is set up in the Grip Stick Setup section.

Deadband: This setting determines the range of force that produces no motion at all. Since a Grip Stick is somewhat temperature sensitive, this value should be set to about 200 or more to reduce the possibility of drifting. Values of 5.0 for the gain and 2.0 for the Frictions and Stiction are good starting points. You do need to press the Apply button for your changes to be used. They will not be saved unless the Axis Setup Save button is pressed. Ensure that you are ready to press Stop or even the emergency stop button when setting up the Grip Sticks as they can produce very fast motion.

5.2.7. Mimic Settings

This section explains moving an axis by hand control which can be used to record and playback motion during a shot.



Mimic ▲

Type: Hand Held Box ▼

Controller: None ▼

Damp 1/50 sec: 0

Speedboat Deadband: 0.000

Handwheel Scaling: 1.000

□ Auto Scaling

Encoder Lines: 2000EL

Mimic Control: When an axis is mimic-controlled, it can be controlled by an encoder or it can be controlled from the hand held box (if using Grip Sticks, you should also select Hand Held Box). This pull down allows one to select from the different available options of control. The options currently available are Hand Held Box, Encoder Position, Encoder Velocity, Speedboat Deadband, Encoder Potentiometer and Tracking.These modes define the form of input that can be used to move the axis while it is learning a motion to replace.

- The **Hand Held Box** mode allows the motor to be moved exactly the same as with the normal hand held box motion.
- **Encoder Postion** directly controls the axis from another encoder, but you can run the axis into a limit or run it too fast in this mode, however when running it will exactly duplicate the controller's motion.
- **Encoder Velocity** is similar except the axis limits in terms of position and velocity are observed and this can add a slight delay to the motion, but it will stop before hitting the soft limits and will be limited as to speed and acceleration. Encoder Vel is active even if mimic is turned off.
- **Speed Boat** which gives you a speed controller for the axis, like the speed throttle on a boat. The centre position is off and forward moves the axis in one direction, backwards moves it in another.
- **Encoder Potentiometer** ensures that the mimic controller and the motor always stay in sync; whenever mimic control is turned back on, the motor will first go to the current position of its mimic controller before the user can control it again, to bring the two in sync. This is normally used with a controller that has a specific limited movement range, like a focus hand controller that cannot move more than one turn.

"Tracking" option is for advanced MRMC use only.

Controller: If an axis is controlled from an encoder, you have to inform the software which encoder is controlling it. Usually, there are several mimic controlled inputs listed, so the controlling axis will be Mimic 1, Mimic 2, Mimic 3 etc. If the axis is mimicking itself (auto-mimic), then the controlling axis is "Auto". See Chapter 12 Mimic Mode for more details. If no mimic inputs are listed, contact MRMC for getting mimic inputs set up on your system.

Damp 1/50 second: This is the smoothing window to be used with Encoder Position control. It is the number of samples at 50 Hz over which it makes the motion smooth.

Thus 0 (the default value) is no smoothing and 20 is very heavy smoothing. Note that heavy smoothing will also introduce a slight lag into the response of the system.

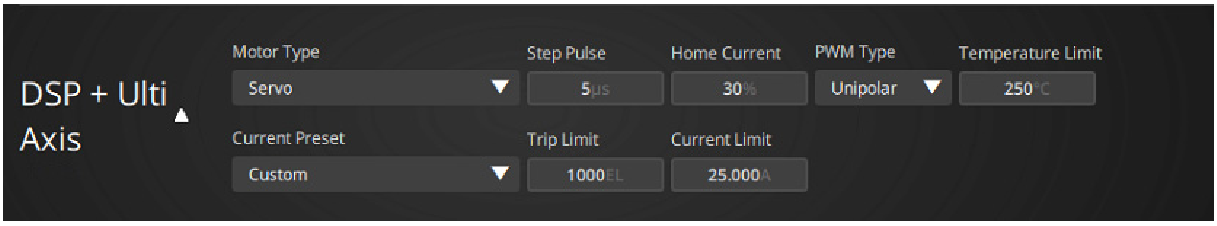
Speedboat Deadband: With the “speedboat” control, it can be very hard to find the exact position for “stopped”. For this reason, the Speedboat Deadband feature allows a wide area of motion to equate to “stopped”. This is in encoder slots, so the actual range of movement it reflects will depend on the hardware.

Handwheel Scaling: An encoder controlled mimic axis can move at a faster or slower rate than the encoder that controls it. The desired ratio of velocities between the controller and the controlled is entered here. That is, if you want the controlled axis to go twice as fast, then double this number. If you want it to go backwards then change the sign. These numbers must not be changed while the axis is moving. For an axis that is controlling itself, this value is automatically set to 1.0 by the software. When you are in Speed boat control, the scaling affects how the motion of the encoder relates to the speed of the axis. With a scaling of 1, one turn of a 1,000-line encoder will produce full speed.

Auto Scaling: You can scale an axis to follow a controlling mimic axis using this option. The dialog box allows you to move the controller to its limits and then the mimic scale will be set to allow full range of travel on the controlled axis.

Encoder Lines: If you are controlling a lens (focus, zoom or iris) from an encoder pot with only 1 turn, you can set the scaling so that one turn of the pot will give you full range of travel according to the set soft limits. Set the soft limits of the axis, and number of lines in the encoder. Enter the number of lines (such as 1000, 2000 etc.) and click the calculate (■) button and the scaling for the axis will be set appropriately.

5.2.8. DSP + Ulti Axis Tunings



DSP + Ulti Axis ▲

Motor Type: Servo ▼

Step Pulse: 5µs

Home Current: 30%

PWM Type: Unipolar ▼

Temperature Limit: 250°C

Current Preset: Custom ▼

Trip Limit: 1000EL

Current Limit: 25.000A

These settings are only valid for systems with DSP-based axis cards. This section allows the user to assign different motor types and configurations to different ports on the **same axis card**. This requires the latest version of the axis card firmware to be installed. If the axis's board has been loaded with a non-supporting software version, some of the fields may be greyed out and inaccessible.

Motor Type: This pull-down list specifies the type of motor being driven from this specific port. This is a technical section and you may need assistance from MRMC to select the correct motor type for your system. The available options are:

- **Servo:** Select this if you are using a Servo motor on this axis.

- **Stepper:** Select this if the motor being driven is a stepper motor.
- **Stepper Feedback:** Select this if the stepper has any feedback mechanism installed and Flair takes this data as an input through the motor cable.
- **DtoA:** Select this when using motors which require analogue voltage to run, like video cameras lens motors in general.
- **DtoA Feedback:** Same as D to A but with a position feedback mechanism installed.
- **Lens Iris, Lens Focus and Lens Zoom:** Select the corresponding Motor type when using a serial lens controller. (Only applicable with suitable hardware, contact MRMC for details). Currently implemented for Preston, Canon and Fuji serial lens controllers.
- **Velocity Output:** The Digital Output controls the speed of the motor.
- **CAN:** Axis boards can communicate to external drives using Controller Area Network connections. MMRC will inform you if this is required.
- **Vector**
- **PWM Servo:** PWM Servo is a type of Motor controlled through a Pulse Width Modulated amplifier built into the Hex Board. Select this if using a Servo motor with an MRMC Hex board.
- Reference Encoder:
- Servo Can Encoder
- Hydraulic
- Servo Bliss Encoder

Step Pulse (µS): Specifies the pulse width in microseconds of the step pulses generated by the axis card. Some older stepper amplifiers do not respond to very short pulses. Most modern amplifiers can handle pulses down to 1µS.

Home Current (%): When homing a lens control motor, it keeps turning the lens back until the physical limit is reached. Therefore, to eliminate the possibility of a motor damaging the lens, this control allows the user to specify the percentage of the normal running current that the motor is allowed to draw when trying to find its home position. 20-30% should be a safe margin but different lenses may require lower settings. (This feature is developmental and is subject to change.)

Important

The values of the next parameters are critical since setting them too high might allow damage to the motors.

PWM Type: Specifies the PWM type for stepper motors depending on the type used. PWM depends on the motor type and size. MRMC can advise on the setting to use for any particular motor. **Unipolar** should be selected for motors with low inductances (less than 1mH) such as focus motors and Ulti-Head pan and tilt motors, while **Bipolar** should be used on all other motors. Note: Using Bipolar incorrectly can damage the motor.

Temperature Limit (only applicable on Ulti-Axis cards)**:** Some amplifiers are equipped with temperature sensors. Flair can read the value of the sensor and make sure that the specified limit is not exceeded during the operation of the motor; otherwise it will disengage the axis. 255 is the maximum limit for the temperature sensor. This feature may be modified to read a temperature in degrees, but as of Version 6.4 it is a range from 0 to 255. Around 180 is cool, so the value should normally be set in the 200s.

Current Preset: In order to assist with setting the current limit, a number of presets are available to set the current limit to a correct value for some of the standard MRMC products. Regardless of how the value is set, it is important to monitor your motors after these values have been set to ensure that they do not get too hot when under load.

- **Custom** allows the user to set the current limit.
- **Min. Focus/Iris/Zoom** sets the limit to 0.5 Amps.
- **Milo Focus/Iris/Zoom** sets the limit to 0.8 Amps.

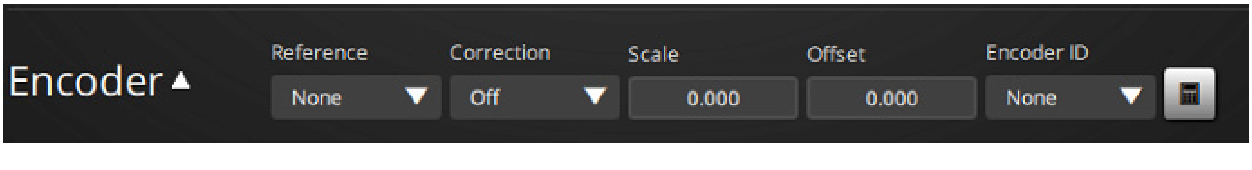
Trip Limit (Enc. Edges)**:** Specifies the number of encoder edges that the motor is allowed to lag from the controlling signal before it trips out. For a 1000 line encoder, 250 means the motor is allowed to lag behind the controlling signal by ¼ of a revolution. For mimic applications, if the mimic input varies too rapidly, the motor might not have time to cope with that and the Trip limit needs to be increased. This limit does NOT apply currently to Robot or EtherCAT type axes, or axes on the old Transputer style axis board.

Current Limit: (only applicable on Ulti-Axis cards) Specify the maximum current that the motor can draw at any time. Setting this value to high can mean that the motor overheats and burns out. (Can only be adjusted with the Custom current setting.)

5.2.9. Encoder Settings

Reference Encoder settings

A reference encoder is a secondary encoder which is used to determine the exact position of an axis when the linkage from the motor can slip. Motors can drive axes with a rubber wheel to ensure quiet operation and then might use a reference encoder to correct for any slippage.


Encoder ▲

Reference	Correction	Scale	Offset	Encoder ID
None ▼	Off ▼	0.000	0.000	None ▼

Reference: This selects which axis is being used as the reference encoder input. The encoder is treated like an actual axis even though no move can be programmed on to it and it is only used as an input. The reference axis must be scaled exactly correctly. Whenever the system is not moving and it is determined that the axis is not in the exact position as determined by the reference encoder, then the axis position is set to the corrected position. You may notice that at the end of a move, the axis position is updated shortly after it has stopped moving. The settings for the reference encoder apart from the Scale and the Board and Port are ignored. The Axis Type should be Independent, and the Motor Type should be Servo.

Correction: This sets whether the correction needs to be applied to the axis. It should be turned off until you have verified the scaling of the axis and the scaling of the absolute encoder.

Absolute Encoder Settings

Most motor encoders are incremental which means they know how much the move should be, but not exactly where they are. The position of a motor with an incremental encoder is typically set to 0 on start up. If the axis has a secondary absolute encoder, this means that even through a power down, the encoder knows exactly where it is. This can be used to set the position of an axis and avoids the need to Home or Zero an axis. Absolute encoders are not used when running the motor, they are typically only used on power up and when one wishes to verify the position of the motor.

Scale: The Scale between the absolute encoder and the motors position must be determined and this will vary depending on the gearing and the type of absolute encoder. MRMC will typically supply this value, but it can be calculated as well using the “Calculate encoder scale” feature.

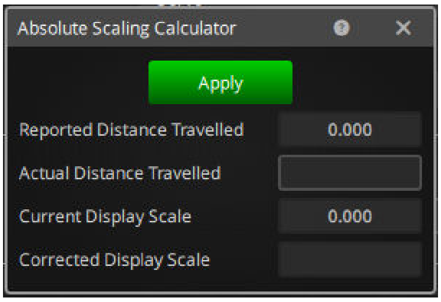
Offset: The offset defines the difference between the absolute encoders' position and the axis position. If, for example, the absolute encoder read 0 when the Tilt axis was down by 5 degrees, this could be used to correct for that.

Encoder ID: An axis board that supports auxiliary absolute encoders can support as many encoders as it supports axes though it may not have this many attached. You should consult MRMC for exact connections, but generally the Absolute encoder will have an ID matching the port of the motor it is connected to. Set this to the same as the value in **Port** as above.

Note

To use the absolute encoder to set the position of the axis, it needs to have its zeroing style set to Absolute Encoder.

Calculate (▣): The calculate button opens a dialog box that allows you to convert the reading of the absolute encoder into the matching display scale of the motor. It works in the same way as the standard Scaling dialog box except that you must set the offset for the absolute encoder at the zero position of the motor. Move the axis to its correct zero position. Then set the offset value for the absolute encoder by storing on the offset box. Then move the axis to an exact known distance from its zero and then enter that distance into the Actual distance travelled box. This will then calculate the correct scaling and adjust the entered offset value into the correct units.



Absolute Scaling Calculator

Apply

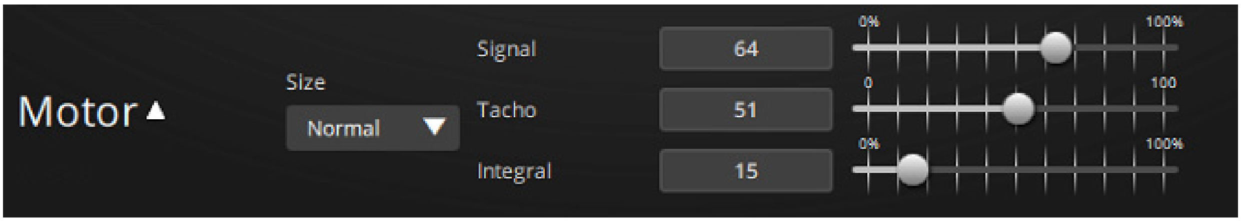
Reported Distance Travelled 0.000

Actual Distance Travelled

Current Display Scale 0.000

Corrected Display Scale

Motor Tunings



Motor ▲

Size: Normal ▼

Signal 64

Tacho 51

Integral 15

The basic concept of positional servo motor control is that a sensing device on the motor is used to determine its current position, that position is compared to its desired position, and the error is then used to make an amplifier drive the motor to the correct position. If the correcting drive is exactly proportional to the error, then the motor would be driven rapidly to the correct position, and would promptly overshoot the correct position and then go into an oscillation around the correct position. Due to this, additional factors are used to stabilise the motor. The control model we have adopted has three controls: the signal gain, tacho gain and integral gain. These three controls are varied to get the motor to hold position correctly.

Size: There are three setting for motor size. If you select **Normal**, then the tuning values are as represented. If you select **Medium**, the values used are 1/5 of the value shown and if you use **Small**, then the values used are 1/10 of the values shown. Unless definitely required, it is recommended that you always use **Normal**.

Signal Gain: This controls the response to the error. If the signal gain is low, then the error will have to become quite large before it is driven to the correct position, if the signal gain is high, then even a small error will cause the motor to try to correct its position.

Tacho Gain: The system's response to the signal gain is limited by tacho gain. If the tacho gain is too low, then the motor will oscillate (possibly violently). If the gain is too high it will be sluggish in its response, and the motor may tend to hum or buzz.

Integral Gain: Not actually intimately connected with the control of the motor like the previous two controls. The integral gain slowly corrects for offset errors. If you have a heavily loaded lift, it will tend to sag out of position until the corrective force matches the displacing force. Having an integral gain will detect this offset error, and will increase the corrective force to minimise the error.

5.2.10. Axis Text Configuration Files

Import Axis Config

The standard axes on a Milo or other system do generally not need to be changed, but when adding axes for model movers etc. often the scaling or the limits will be specific for that motion device. For example you might have a selection for different turntables with different speeds and scalings and you would have to write these down in order to set it up for a shoot. Now with the Text Config Read/Write facility, you can do this on a per axis basis, saving or reading the setting for an axis into a text file with a file name that clearly identifies it. Since you might want to read this into a specific Board and Port configuration that could be different from the one saved, when you import it, you would normally select “Ignore Pipe” so as not to change the setting on the axis you are importing to. This feature will allow you to build up a library of config files for any model mover or auxiliary system you might have. These file should be saved with a .”txt” or a “.dat” ending (extension) and are standard ascii files that can be read and edited in any text editor. This is accessed with a button at the top right hand corner of the axis setup. “Text Config..”

There is also the ability to read in a Text Configuration file for the entire system, but the difference here is that for an axis to be changed by a system config file import, the name of the axis needs to exactly match the name of the axis in the text file. If you have a “Rotator” axis and try to read in a file that contains a section for a “Turnable” it will be ignored. However, with the Axis Text Configuration import, you are reading in a specific config onto the current axis, and the name is ignored. (System Text Config files are importing under File->Config Data->Import Text Config.)

5.3. Setting up an Axis

The basic procedure for setting up an axis is as follows:

1. Set the axis board and port numbers on all axes according to the board and port to which the motors are connected, consult MRMC if you face any problems. Ensure that no axes share the same board and port, including the camera. An axis that does not move despite a reported changing position is either assigned the wrong board and port; not connected; or is being overruled by another axis in the same port on that board.

2. Ensure the motor is connected and plugged in, but leave the emergency stop pressed in.

3. Go into the Axis Setup for that axis and set the motor tuning levels to **5**, **5** and **5**.

4. Set the **Display Scaling** to a sensible value, you can use **1.0**, but it is unlikely to be **greater than 500 or less than 0.01**.

5. Set the **soft limits to +500** and **- 500**.

6. Set the **Goto factors** to **0.3** the **Runtime Factor** to **0.9** and the **other factors** to **0.75**.

7. Set the axis **Velocity** to the expected maximum speed for that axis. If you are scaling a rotational axis in degrees, then specify an estimated degrees per second value. If in doubt set this value low. Enter the same value for axis **Acceleration**.
8. Repeat for the **HHB Velocity** and **Acceleration** and **Apply** the setup.
9. (For servos) Release the emergency stop, and engage the axis that you want to set up.
10. Conditional: If the axis runs off briefly and then turns off, then it usually means that the motor leads are reversed. Handle and retry.
11. If the axis engages and the current position is now reported as black instead of white, then the axis is probably properly connected. If it is stays white even if you directly zero it, then the motor/encoder is not connected properly. If it apperas engaged when directly zeroed, and then turns disengaged again after 5 seconds, then the axis limits are not connected properly. Handle so it re-engages and stays engaged.
12. Now turn on the hand held box and set the hand held box to slow by pressing the **Slow** button and move the axis momentarily. Adjust the hand held box speed (Fast and Slow) as needed and the Hand Held Box Velocity if the speed is much too high. (Don't forget to Apply any changes you make).
13. Conditional: If the axis runs off briefly and then turns off, then it usually means that the motor leads are reversed. Handle and retry.
14. Tune the motor referring to the section Motor Tunings on page 33.
15. Note the direction the axis is moving in. If it moving the wrong way, correct it by reversing the direction in the Axis Setup, and accepting the Setup.
16. Next adjust the scaling: Move the axis to a known position and directly zero it there, then move it a known distance away (the further the better) and read what distance the computer thinks the motor has moved. Then adjust the **Display Scaling** per the following formula: x

$$\textit{correct display scaling} = \frac{\textit{old display scaling} \times \textit{actual position}}{\textit{reported position}}$$

17. When this is correct, the reported position will match the measured position. To do this, you can use the scaling calculator, which can be accessed by clicking the calculate button. See Basic settings on page 20 for details on use.
18. If you have target tracking software, enter the axis **Internal Scale** so that the units are **Metres** for translational movement and **Radians** for rotational movement. Use the **Units** pulldown to do the calculations for you. If you have scaled an axis in other units, then work out the relationship between your units and metres/radians to calculate the internal scaling appropriately.
19. Now that the axis is scaled correctly, set up the maximum speed. This is done by specifying a move on the axis in question and running it in shorter and shorter time until the axis trips out. The move should be run with computer fairings selected in the Job type display. If the computer reports that the move is too fast, then increase the axis **Velocity** in the Axis Setup or increase the axis Acceleration if it is reported to be too hard an acceleration. (This is only being

done for set up purposes, normally, these values are not changed). Remember to Apply any changes in the setups as you do this.

20. Now reduce the axis **Velocity** until the computer reports that it needs more frames for the move, and the axis will actually do the move at that speed.
21. Using the move above with 25% more frames and a similar principle, set up the maximum acceleration. Use **User Fairings** in the Job type display, and set the fairings on this axis to **50 up** and **50 down** with type 5, then run the job with shorter and shorter fairings until the axis trips out, this is the acceleration limit of your axis. Now get the computer to reject the move that the motor will not do by setting the maximum acceleration lower and lower until the computer refuses to run it. Adjust this value so that any move that has too hard an acceleration will be rejected by the computer. The maximum velocity can also be set by trying out different values in the Velocity box in HHB settings. Continue increasing this amount until the axis trips out, then reduce by about 10% for safety. HHB Speed as displayed on the Hand Held Box LCD should be at maximum.
22. Set the **Runtime Factor** to **0.9** for safety.
23. Set up the Zeroing **Type** for your axis. A detailed explanation of zeroing is covered in Chapter 18 Zeroing. Use this to determine the correct type, and also refer to this section for error handling.
24. Set the **Zero Velocity** so that the axis moves slowly towards the selected reference point. A typical value would be **between 3 and 10**. (or negative)
25. Set the **Zero Offset** to **0.0**, the **Zero At** to **0.0** and the **Zero Time** to about **20 seconds**.
26. Now Home Zero the axis. If it runs out of time, just zero it again until it reaches the reference point and stops completely.
27. Now move the axis with the hand held box to the correct zero position. (This position is up to you for an axis that is not target tracking, but for target tracking axes, this position can be critical, contact Mark Roberts Motion Control if you are unsure).
28. Once the axis is correctly positioned, go into the Axis Setup display and store the axis current position into the **Zero Offset** box using the mouse store button, and Apply the changes.
29. Now zero the axis again and verify that it returns to the correct Zero position.
30. If you want to Zero at a position that is not 0.0, enter the desired value in the **Zero At** box.
31. Set the **Zero Time** to be the amount of time it takes the axis to get from its furthest position from the reference point to the reference point when you are zeroing it.
32. Save and Apply what you have done so far.
33. Set the **Maximum** and **Minimum** limits of the axis travel: Move the axis to its greatest extent of travel, and store that position into the Max Limit box using the store button. (You may have to extend the limits first). Repeat for the minimum limit by moving the axis to its most negative extent of travel. Alternatively, set these amounts to known values, i.e. if you know the tilt can go up 30 degrees and down 30 degrees, then enter 30 in the Max Limit and -30 in the Min Limit. (If

these values are entered the wrong way round, then the axis will not move at all!)

34. Now set the **Pre-roll** and **Post-roll factors**. This is done by part running the move over just the middle few frames of a move and observing the axis starting and stopping. If it starts and stops too violently, decrease the relevant factor. The optimum Pre-roll Factor is such that a fast move does not shake once you are into the move, and the optimum Postroll Factor slows the rig down rapidly but without mechanical strain. (Note that in the Job type display, the option Start: quickly/along path should be set to quickly as should the Stop: option.) The Goto Factor and Goto Acceleration factors should be set to around 0.3 as a good starting point.

35. The Mimic settings are not relevant to general axis setup and are covered under Chapter 12 Mimic Mode.

36. If this axis has a brake on it, check the Brake option, otherwise uncheck it.

37. Set the **HHB Velocity** and **Acceleration** to values that allow you to move the axis effectively without shake and without long slow decelerations. The Jerk can also be set of this value is available to you. Start with a value of about 3 times the axis velocity and experiment until you are comfortable with it.

38. **Save** and **Apply** all the changes that you have made so far.

39. Repeat this procedure for all axes.

5.4. Cartesian Setup

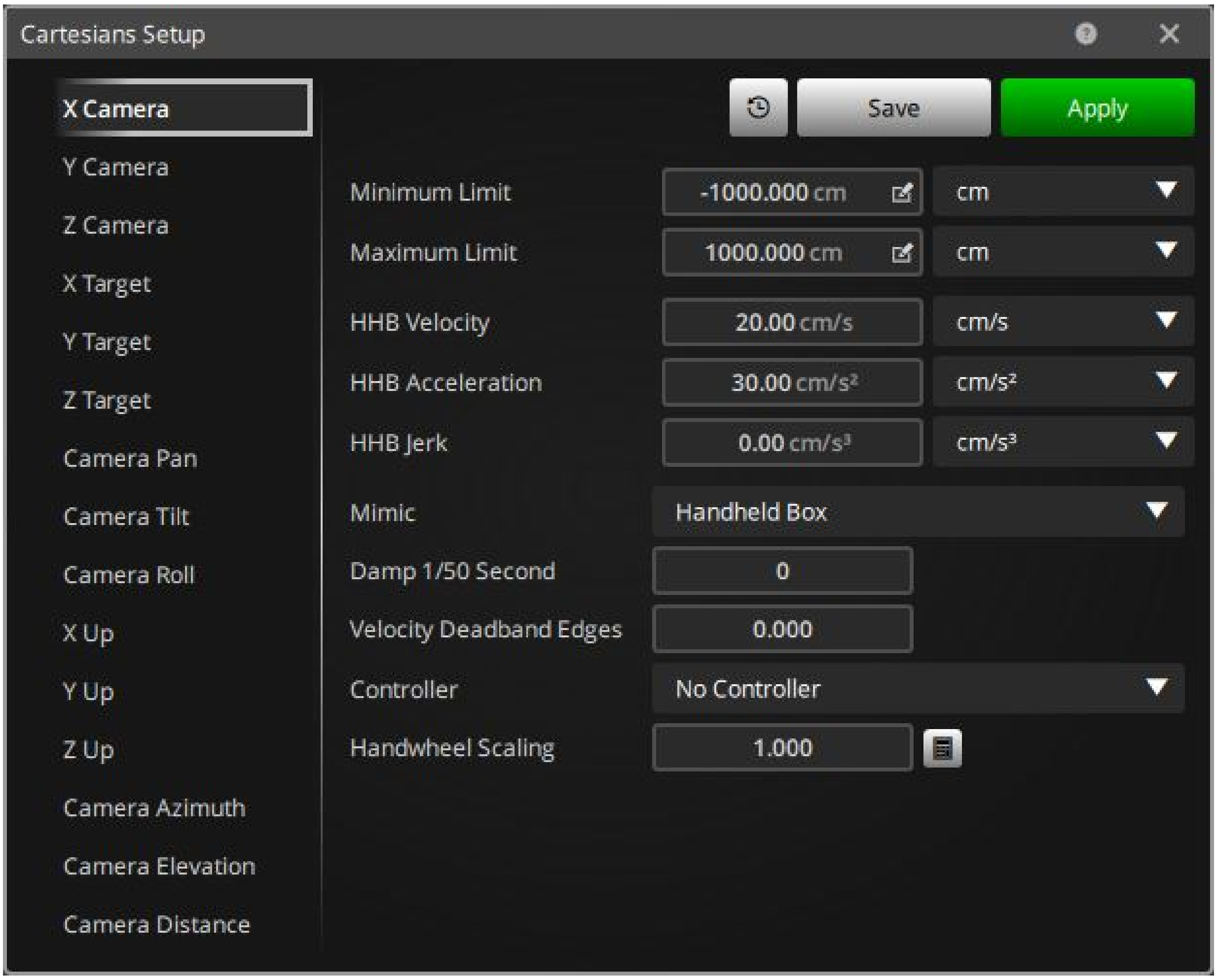
Cartesian setup allows the user to define the speed, acceleration and soft limits for each Cartesian axis which are used when in Cartesian Control mode. You can also set up the mimic controller for each Cartesian axis in almost the same way as you set up the mimic controller for a normal axis.

The units on the translational setups are in metres which is the standard units for the Cartesians as used by the Flair system. The soft limits are then in terms of metres of travel for the linear axes. When you are in the Pan, Tilt or Roll Cartesian setup, the units are in degrees. The soft limits are set in the same way as with a normal axis, you can either type in a value or store the current value using the mouse store button. It is recommended that you have the rig in Cartesian control when you are doing this.

Due to the complexity of the geometry in a rig, it is virtually impossible to have limits that will prevent the user from ever hitting some part of the set or floor with some part of the rig unless the limits are set so prohibitively as to restrict full use of the rig. For this reason, it is always the operator’s responsibility to know exactly what the rig is doing and how close it is to parts of the set and the floor or rails and particularly any people who may be near the rig.

Cartesian Limiting is made complex by the number of ways in which the rig can be configured. Whilst the limits do work, it is possible to drive beyond the limits in extreme conditions and it is also possible that limiting the motion can result is rough motion. The area is under constant review and if you run into something that does not work as expected, report it to MRMC.

The units for the speed and acceleration are currently also in metres and degrees as applicable, with speed being in metres per second (degrees per second) and the acceleration being in metres per second per second (degrees per second per second). These values are not particularly intuitive to set, and it is recommended that you start with low values like 0.3 for the speed and 0.6 for the acceleration for the translational axes and 45 for the rotational axes, and then adjust them in small increments to find the best settings for you. For more information on Jerk value, see is Hand Held Box (HHB) Settings on page 26. Normally, Jerk value should be set at about three times the HHB Acceleration.



Minimum Limit: -1000.000 cm, cm

Maximum Limit: 1000.000 cm, cm

HHB Velocity: 20.00 cm/s, cm/s

HHB Acceleration: 30.00 cm/s², cm/s²

HHB Jerk: 0.00 cm/s³, cm/s³

Mimic: Handheld Box

Damp 1/50 Second: 0

Velocity Deadband Edges: 0.000

Controller: No Controller

Handwheel Scaling: 1.000

The settings for the rotational axes such as Pan, Tilt and Roll are in degrees and these values should be set to approximately 30-45.

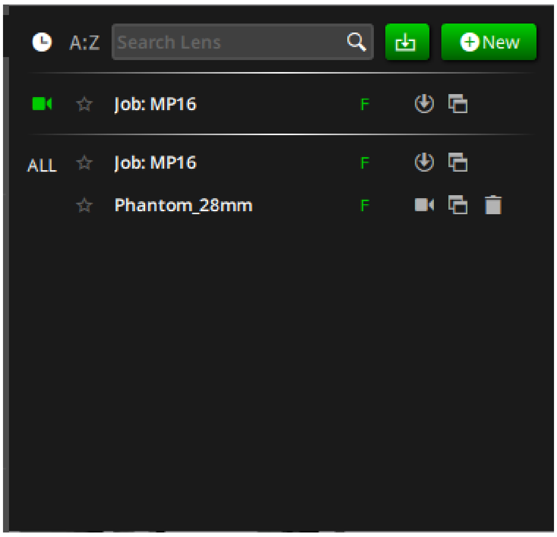
For Mimic Control, Damp 1/50s second, Speedboat Deadband, Controller, and Handwheel Scaling, see Mimic Settings on page 28.

5.5. Setting up a Lens

In order to set up a lens, in the top menu bar, select **Setups** → **Lens Setup** and select the name of the lens you wish to modify. The normal number of lenses supported is 20. This may be increased using the Lenses setting in the Flair.ini file. For more details, see .

Note

On a multiple head type system where each head can have a different lens, once you choose Lens Setup, a pull-down will appear listing the heads and you can select the lens for each head separately.



Name: The lenses are identified by a number, and the default lens currently mounted on the camera is displayed on top. If you want to set up a new lens based on a previously set up lens, clicking on the Duplicate Lens (⧉) button will create a copy of the lens.This allows you to base a new lens on an existing one and can save some time when setting up a series of lenses which often share similar offsets.

Mount on camera (■◂): This button allows you to mount another lens on the camera and make default. For the lens that is mounted and is default, the icon appears in green on the left.

Here’s an explanation of what ‘Mount on camera’ does. Different lens settings vary from lens to lens. For example, if the focusing mechanism involves driving a lens through its own ring gear, then the scaling of the lens will vary from lens to lens. Due to this, a different axis setup is needed for each lens and also different soft limits, zeroing etc. When this icon is selected for a lens, the axis setup associated with that lens is downloaded to the Focus, Zoom and Iris axes so that the lens operates correctly. This is also true when the program loads. If the default lens has this option selected, then an axis setup will automatically be downloaded to all the lens axes. The user can mount the lens (i.e. download its properties to the engine) but if the lens hasn't been configured or calibrated correctly then the focus/zoom/iris axis may not operate as intended.

Use the **Delete** button to delete the lens.

Import a new lens (⭳): This allows you to import a lens setup from a different Flair installation by specifying the path of the lens configuration file.

> Note
>
> All units except the focal length are in metres. The focal length is in millimetres.

To create a new lens configuration, click the **New** button.

5.6. Lens Setup

This dialog box allows you to view and modify the axis setup associated with the selected lens. You can store various lens configurations with different names to apply any of these lens configurations depending on the lens that you mount on the camera.

Name: This is the name that the current lens settings will be identified by.

Focal Length: Focal length of the lens.

Moves: Most lenses focus by moving the elements of the lens closer and further from the film plane or camera sensor. This is the normal way it is done and this box should be checked in almost all cases. In older camera systems, you could focus the image by holding the lens still and moving the camera body. This is mechanically more complex but allows focusing to extremely close and avoids moving the nodal point of the camera. When setting up this sort of a system, there is a different procedure and user with this system should contact MRMC for details.

When in Lens Moves mode (as opposed to Moving Camera and Fixed Lens), the three listed positions must be in increasing Focus Position, and decreasing Target distance. An error message will appear if these values are entered incorrectly and you press the Calculate button. Normally when setting up a lens, it is zeroed at Infinity and this is the first entered value (0.0) and move the lens to infinity and zero direct focus there.

You can enter a non-zero axis position value for the Infinity position, and this is used to automatically calculate the infinity offset. However unless unavoidable, it is recommended to zero the lens at infinity.

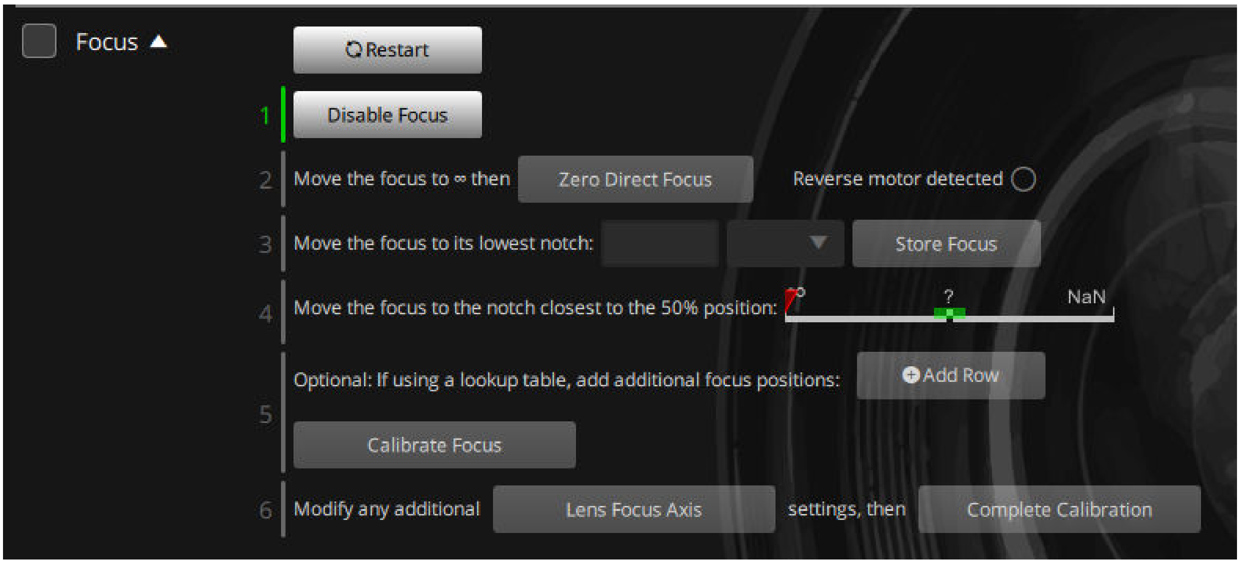
Mounted: The Mounted checkbox is read-only and informs the user that this lens is the one that has been downloaded to the engine (i.e. the one used or mounted on the rig).

Pressing the Apply button will download to the engine and set the Mounted checkbox active.

Import: Allows you to import settings from another file. These settings can be saved using the Export button on Lens Properties dialog box.

5.6.1. Focus Calibration

The Lens Setup dialog box allows settings of Focus, Zoom, IRIS and other lens settings. Clicking the Calibrate button for Focus in Lens Properties dialog box displays the Focus Calibration section of the Lens Setup dialog box. This section consists of sequence of steps that you can follow to calibrate focus axis. The target distance measurements are done in your selected measure units (such as centimetres or inches).



□ Focus ▲

Restart

1 Disable Focus

2 Move the focus to ∞ then Zero Direct Focus   Reverse motor detected ○

3 Move the focus to its lowest notch: Store Focus

4 Move the focus to the notch closest to the 50% position: 0 ? NaN

5 Optional: If using a lookup table, add additional focus positions: Add Row

Calibrate Focus

6 Modify any additional Lens Focus Axis settings, then Complete Calibration

1. The first step in setting up a lens is to got to Axis Setup for Focus axis and select an Internal Scaling, about 0.005 is a good starting point. Set the Minimum and Maximum limits very large to facilitate setting up the lens (for example +200 and -200. Press Apply in the Axis Setup dialog box.
2. In the Lens Setup dialog box, click the **Disable Focus** button. (With a stepper motor driven Focus, which you cannot back drive, you will have to do these steps using the hand held box, being very careful not to run the lens into end of travel.)
3. Move the lens with your hand to infinity then click **Zero Direct Focus**.
4. Manually move the lens to the lowest notch and enter the target distance in the next line. The reading in the left panel shows the current Focus axis position. Click **Store Focus** to store the motor position in the axis.

   Each line contains a Target distance and a matching Focus Position when the image is sharp. The Target distance is measured in the same units as you use in measuring target tracking distances (usually centimetres or inches), and the focus value is entered by using the mouse store button over that box. (This is a calculated physical measurement in metres, and depends upon the current internal scaling of the lens.)
5. Similarly, move the focus ring to the middle position and record the target distance. Clicking **Store Focus** in this line will store the motor position into the lens axis.
6. Click **Calibrate Focus** so Flair can calculate a curve for target distances and respective motor positions.
7. Clicking **Lens Focus Axis** will apply the lens' internal axis properties to the actual axis.
8. Click **Complete Calibration** to store the lens configuration in the lens.
9. Save to save to the disc and apply to apply the changes to the engine.

Lookup Focus Table

Some lenses do not calibrate well using the 3 points listed above, and you have to create a more complete table of the various focus distances and corresponding motor positions in order to more accurately map the lens. You do NOT need to enter every point, and sometimes just taking every second point works better. There is a natural inaccuracy in trying t line up to a distance not a mark and these inaccuracies can cause the curve to “wow” and so fewer points often works better. If this is the case, then checking the “Lookup Focus” button the lens setup display will enable a different screen for calibrating the focus.

The principle is the same. It is still recommended to zero the lens at infinity, and then move through the focus distances, entering the distance and storing the matching axis position. Since you will often change units from Metres to Centimetres or from Feet to Inches as you get to a closer distance, you may change the measure units while you do this, but be sure to start in the right mode.

Fixed Lenses

This calibration does not currently work for a fixed lens as the Physical Scaling can be worked out exactly from the gear ratios etc.

Curr. Int. & Curr. Disp.:

These are the current display position of the axis and the current internal position of the axis. These output only fields show that the lens is moving and display the current internal units and the current display units. This is mainly helpful if the lens is moving negative which cannot be achieved with the current lens calibration routines.

There are a series of buttons across the screen designed to make calibrating a lens easier and quicker. They mimic the usual actions in setting up a lens. These functions rely on the focus axis being set correctly as Focus Type and not Independent.

Zero Focus

The first step in setting up a lens is to zero the focus at Infinity. You can theoretically zero it and add an offset, but it is simpler to zero at infinity.

Rev Motor

If the motor moves negative as you move it away from infinity, the focus calibration will not work. This button automatically reverses the direction of the focus motor and reverses the sign of any entered values.

Set Travel

Set the maximum limit of the focus axis to the current motor position. This is very useful for setting limits and also getting the travel of the lens for setting up a mimic controller. The minimum limit should be set by the user and is usually 0.

Axis Setup

Opens the axis setup dialog box for the axis which is currently assigned to Focus Type.

Output Data

The data displayed below these buttons is for information only and cannot be directly changed. Focal Length is the focal length of the current lens in millimetres, the Lens Factor and the Scaling fields are the values to be calculated by the program when the Calculate button is used. The Lens Factor calculated should be approximately twice the focal length in metres (for example, for a 50mm lens, the Lens Factor should be approx. 0.1) but it can vary from a small negative value up to 0.5. The scaling MUST be positive and is usually somewhere between 0.05 and 0.001. These are values for standard prime lenses in the range 24mm-100mm. Lenses outside of this range or zoom lenses may give different results.

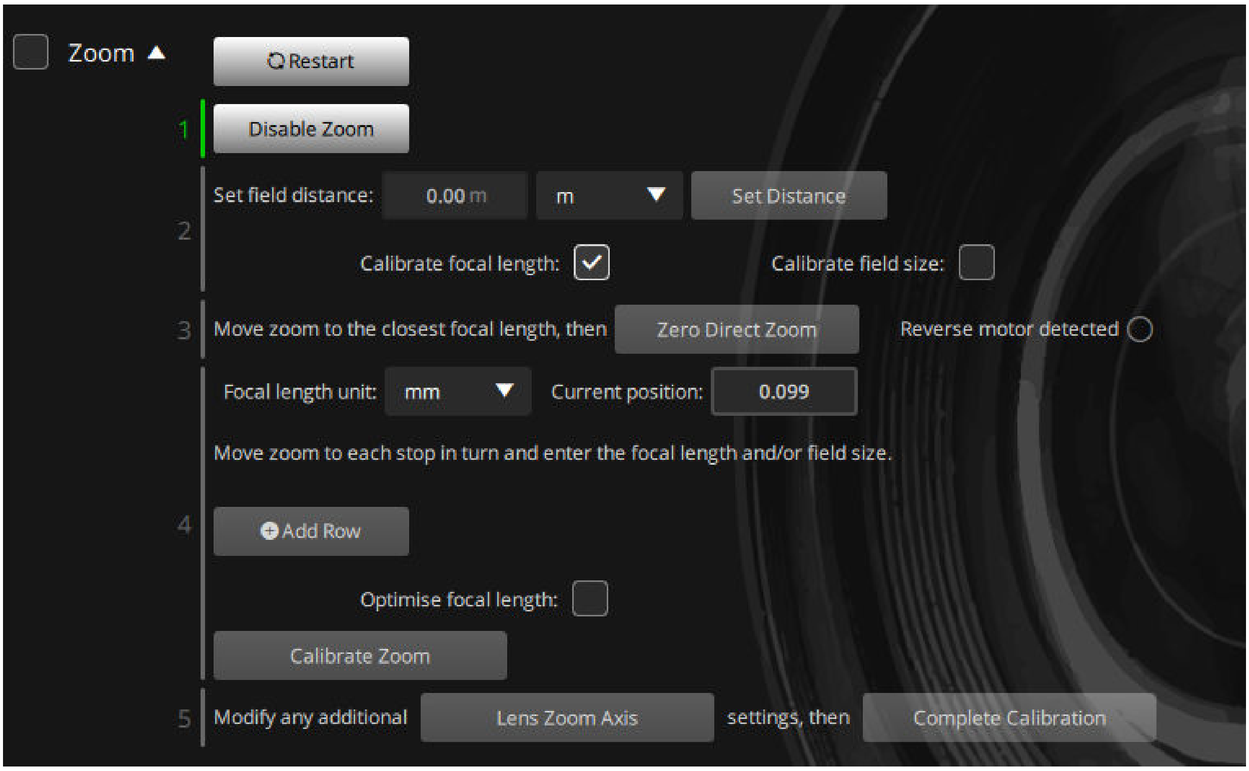
Once the Calculate button has been used, and the values produced are seen to be more or less correct (If they are wrong, they are likely to be heavily outside the limits above), the Apply button can be used to utilise the values calculated, and then test them by moving the lens around in Focus Follows Object mode, and seeing that the position produced on screen is in fact a valid object distance. It is strongly recommended that you zero the lens at Infinity as this simplifies the procedure. If you do not, the initial position for Infinity focus is used to calculate the infinity offset for this lens.

Note on calibrating lenses:

It has been found empirically that MOST lenses will calibrate very accurately using the above method. Some Anamorphic lenses and some Macro (very close focus) lenses do not follow this calibration exactly, but will still operate better than by simply using the axis in an un-calibrated fashion.

5.6.2. Zoom Linearisation or Zoom Calibration

Zoom calibration is done in a similar way as the Focus calibration.



1

2 Set field distance: 0.00 m m ▼

Calibrate focal length: ☑ Calibrate field size: □

3 Move zoom to the closest focal length, then Zero Direct Zoom Reverse motor detected ○

Focal length unit: mm ▼ Current position: 0.099

Move zoom to each stop in turn and enter the focal length and/or field size.

4

Optimise focal length: □

5 Modify any additional Lens Zoom Axis settings, then Complete Calibration

The Zoom Calibration section in the Lens Setup dialog box consists of a series of data entry boxes for the user to enter measurements into that relate to the motion of the zoom axis. The distance measurements are usually done in your selected measure units (for example, centimetres or inches).

Basic Principal

By entering the values of focus length and the size of the viewed image at different points for the zoom travel, Flair can work out for any other positions of the zoom travel, what the respective focus length would be. This means that you can plot a move on the zoom axis in terms of field size, and the zoom will be driven to change the field size in a smooth manner. Normally the field size change does not move steadily with a constant movement on the zoom lens. The feature makes it easier to do contra zooms (zoom in and track back to change perspective without apparent change in object size) and also makes it easier to match the end of one zoom to the beginning of another.

Field Distance

In order to work out a field of view or a focal length given a field size, Flair must know the distance at which field size is being measured. This distance is referred to as the Field Distance. When one is setting up a zoom lens in this manner, it is quite usual to ignore the movement of the nodal point while working out the zoom angle calculations. In this case, the field distance must be as large as is feasible. It is best also to leave the focus as close to infinity as possible when doing these measurements. For a 10x zoom or greater, it is recommended that the field distance value is not less than about 10 feet.

Field Size/Focal Lengths

A zoom lens can be calibrated, or linearised, by entering the field sizes against the motor position or the focal lengths against the motor positions. In the case of a zoom lens that has good calibration marks on its barrel, it is often easier to use the marked focal lengths. If you are doing some very critical work, it may be more accurate to use the field sizes method.

Zoom Positions, Field Size & Focal Lengths

Each line contains a zoom motor position, at each of the points, the user can enter the focal length and/or field size at a specified distance. The actual units of the field size do not matter, but the stated field distance must be in the same units as the field size. Once these figures are all entered, press the Calibrate Zoom button to fit curves to the entered points.

If the linearise calculation fails, there will normally be an error message, but the cause is usually a change of direction in one of the relevant table. If there is no error, these linearisation curves can be applied to the current lens. Once that has been done, click **Apply to Zoom Axis** button. The lens is accepted and these newly worked out curves for linearising zoom will be put into effect. If the lens Save button is used, the zoom positions, focal lengths and field sizes that you entered will be saved onto disk, and may be viewed whenever that lens is selected. It is better to use fewer points than more when linearizing a lens and 4-6 points will often work better than 10 entered at every focal length. (See Table Graph section for more data on lens curves).

Optimize: This is an option you can use to have the system work out a simple curve that closely fits the entered points. If you enter points that are close together, even minor inaccuracies can cause the system to work out a curve that is too complex making zoom motion reverse direction unexpectedly and not resulting in a smooth move. The Optimize option fits a very simple curve that might work better than multiple entered points.

Once all the above has been done, you can view and use the Job type display and set the Zoom lens from **Zoom is Independent** to **Zoom is linearised**, **Zoom is in focal length** or **Zoom is in view angle**. In the latter two, the actual zoom axis will read as focal lengths or as a view angle and can be programmed as such. The mode of each axis of this type (Focus, Zoom, and Iris) may also be changed by using the axis right click pull down menu.

Note that the actual view angle and focal length may not match with those stated on the lens. This may be due to the method you used to measure the field size. This will not adversely affect the linearisation, but if you want to correct this, you should contact Mark Roberts Motion Control. You may also select the Zoom mode by right clicking on the Zoom axis name and selecting the mode from the pulldown menu.

5.6.3. Iris Calibration

Iris calibration is done by entering up to 10 stops and motor positions. The computer will then fit a curve to the entered points and use this curve to convert motor positions to stops and vice versa. The points are entered as normal, simply decide where the iris is to zero, and zero it; then move the iris by hand through each of its marked stops, entering the stop on the right and using the mouse store button to read the iris axis position into the column on the left. The positions must not reverse direction nor can the stops, but otherwise they can be increasing or decreasing. The scaling of the iris axis is not important, but I would recommend something that gives you a number from 5 - 50 for the full travel of the iris and make sure that the internal and display scales are exactly the same. If you do not have enough stops marked on the lens to fill the entire table, then blank out any unused fields. If you have more stops on the lens than positions, make sure that you calibrate to the extremes, and leave out one or more points in the middle, as outside of the calibrated range, the results will not be entirely accurate.

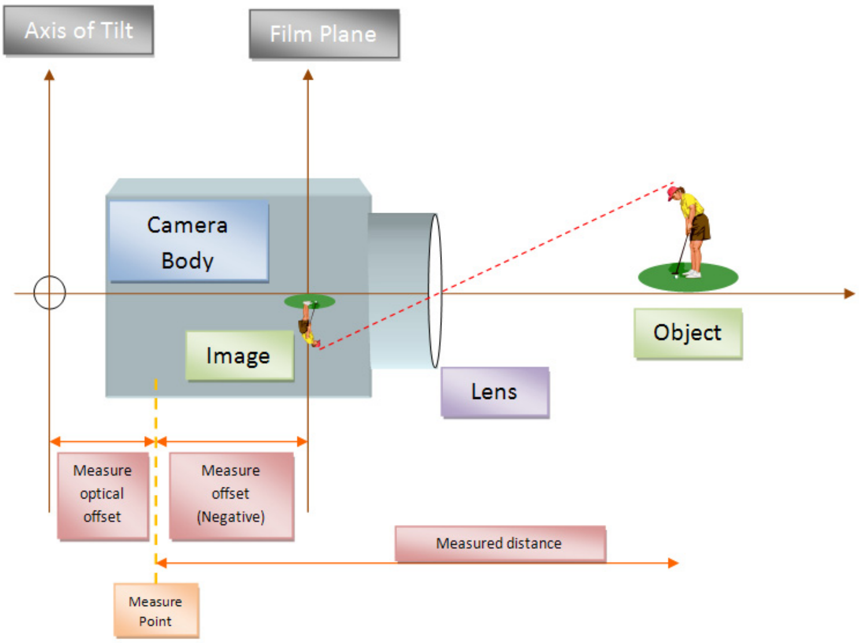
Once all the needed points are correctly entered, you can click on the calibrate button and you should get a message Iris splined OK If there are any problems, the computer should inform you of exactly what is wrong. It is also a good idea to enter the soft limits for iris at this time to make sure that you do not go outside of the calibrated range. Once all is set up and the axis speeds etc. have been verified as OK, it is a good idea to copy the setups for the iris axis into the lens iris axis setups to ensure that when this lens is next selected, the calibration data and soft limits are not lost. This is done in the same way as with a focus calibration, click the mouse store button over the Iris Axis Setup button and the setup for the iris axis will be read into the lens setup ready for use. Next, select Customised Iris to Yes and whenever that lens is selected, the data will be transferred to the iris axis to set the limits correctly.

Note

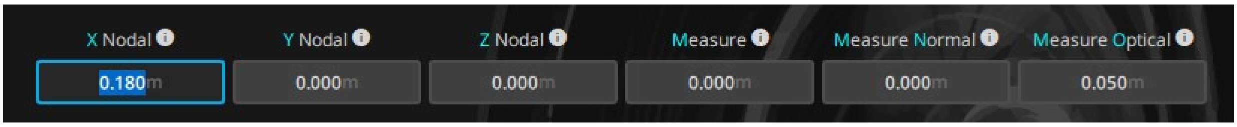
The actual values for stops should be 1.414, 2, 2.828, 4 and so on, the marked values are approximations. So if the computer works out what is 1 stop open from a 4, the result would be 2.828 NOT simply 2.80.

5.6.4. Lens Offsets

Infinity: It is simplest to zero a lens at Infinity focus and work from there. However, it is allowed to set an offset from infinity for the focus. This should only be done once the lens is fully set up as the unit of the infinity offset depends on the lens scaling and this is best worked out with the lens zeroed at infinity. However once this is done, you can move the lens to a new zero point, click the store button to set an offset from Infinity for the new zero point of this lens.



The above values all concern follow focusing. The following values are for Target Tracking.



X Nodal	Y Nodal	Z Nodal	Measure	Measure Normal	Measure Optical
0.180m	0.000m	0.000m	0.000m	0.000m	0.050m

Measuring offsets

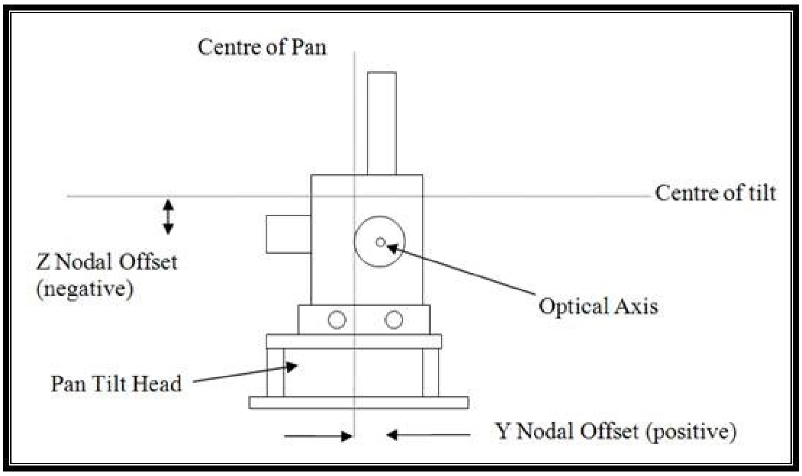
The X, Y and Z Nodal offsets are measured in a very specific way and shown here are the normal camera rigs with the camera mounted so that the last axis produces a roll of the image (mainly). The Bolt High Speed Cinebot has other ways of mounting the camera; for the measurements for the Bolt, see Chapter 10 Target Tracking. Advanced Users can also set up a snorkel as though it were a camera mounted in a different fashion and this is also covered in this chapter.

X Nodal: This is the distance in metres measured along the optical axis between the axis of tilt and the front nodal point of the lens. This can be estimated by measurement, but the exact position of the front nodal point of the lens may be unknown and often found empirically. (See under section on setting up a lens). In the case of a snorkel lens, it is the distance from the tilt axis to the centre of the 90 degree mirror. (In this manual, a snorkel is considered to have a 90 degree mirror at the end of it. A long lens which does not have a mirror is treated as a normal lens with a large X Nodal offset for example borescope/endoscope). The centre of the 90 degree mirror is the point at the end of a right angle snorkel where the optical axis changes direction and is hereafter referred to as the 'Snorkel Reflecting Point'.

For a prime lens, the X Nodal Offset can safely be set to the **Measure Offset + the Lens Factor** with a fair degree of accuracy until you are ready for setting it exactly. The exact value of this is not vital for good target tracking. But it should be accurately set up for data import/export or move scaling.

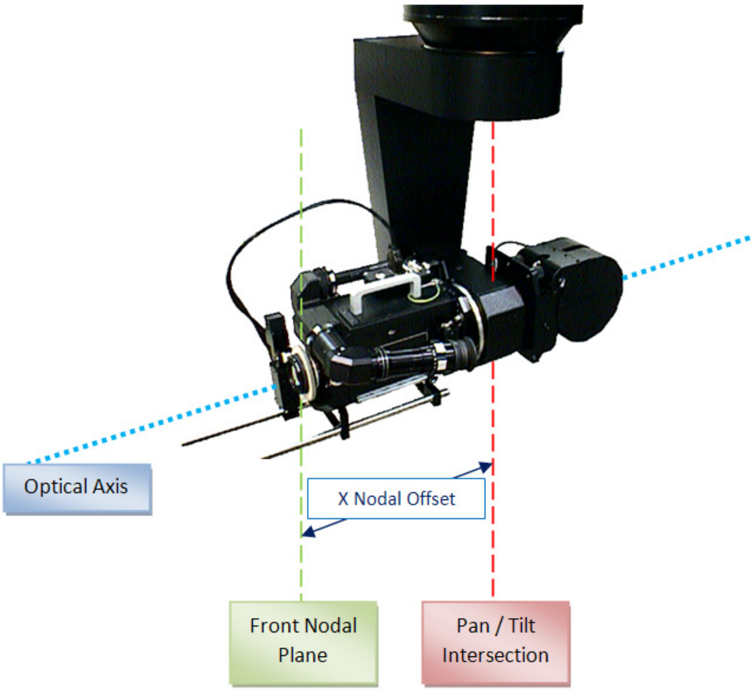
Y Nodal Offset: This is the sideways distance from the centre of the pan to the front nodal point of the lens. This value can be measured as the centre of the lens is a known position. Usually the film plane and centre of the lens are on the centre of the pan rotation, but not always. This is especially true with video cameras. This measurement is positive going to the left when viewed from the rear.

The drawing below is a view from in front of the camera of a normal geared head showing the Z and Y Nodal offsets. These are usually quite small and determining the centre of pan and the centre of tilt can be quite difficult (particularly the tilt).

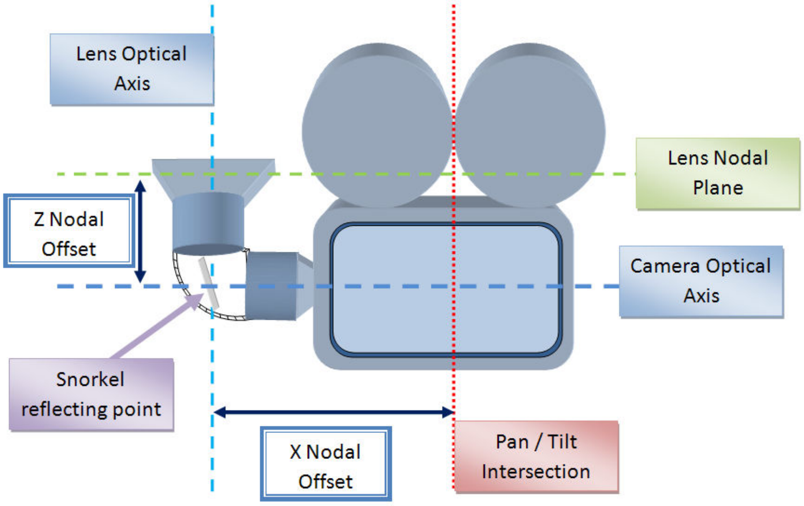


Z Nodal Offset: This is the vertical distance from the centre of the tilt axis and the front nodal point of the lens. Again this distance is usually 0.0. For a snorkel lens this is from the front nodal point of the lens to the Snorkel Reflecting Point. The Z Offset is measured positive going up.

The picture below is of a Mark Roberts Motion Control Slimline camera mounted on a Pan Tilt Roll head. The camera is mounted so that the optical axis of the camera is at the same height as the tilt pivot so there is no Z Offset. The optical axis of the camera passes through the centre of the pan so there is no Y Offset either. The nodal point of the lens is however a small distance in front of the Pan/Tilt intersection point and thus there is an X Nodal Offset of somewhere around 0.310 metres.



The diagram below shows a stylised drawing of a camera with a snorkel lens to illustrate how the X and Z nodal offsets are measured in this case. With a snorkel lens, the light is reflected on a mirror in the snorkel, and the location of this mirror is important. The centre point of this mirror is termed the Snorkel Reflecting Point.

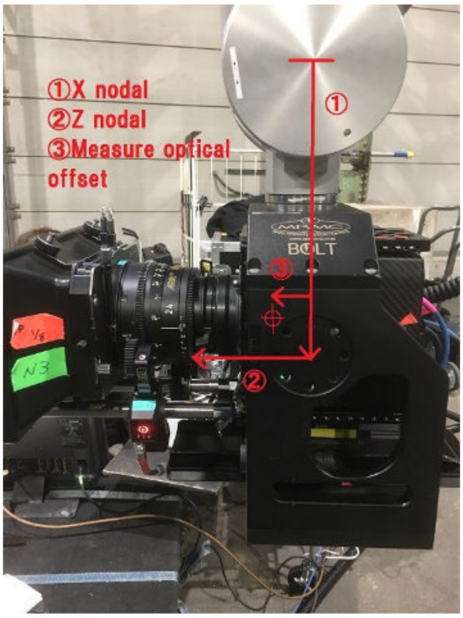


Notes on Y&Z Nodal Offsets: You can use Y&Z Nodal Offsets with a rig that has a tilting pan axis (e.g.: Cyclops, Milo, Overhead etc) but these offsets will ONLY work when you are in **Keep Horizon Level** or **Keep Horizon Relative**. If you are not in this mode, or if this mode has turned itself off, then the move will not necessarily match the programmed move. This is particularly true when you are importing XYZ data.

The X Nodal and Z Nodal Offsets each have an automatic setup routine that is invoked by pressing the right hand mouse button over the relevant entry box. These boxes each have a help button that explains how they operate and their operation is not further discussed here save to mention that you can change the offset while the system is in Cartesian Control without having to turn it off and then on again. This greatly speeds the setup time for these values. Be aware that the accuracy of the setup of your rig will have a huge effect on the accuracy of these routines.

Nodal Offsets for Cameras not “End Mounted”

The technique for measuring the Nodal offsets on a non end-mounted camera is to take the rig to 0 with all axes at 0 and measure in the standard X,Y and Z directions the offsets from the Pan Tilt Intersection to the nodal point ignoring the direction the camera is looking. So with a Top mounted camera which at zero would be looking up, the X is from the Pan Tilt Intersection forwards to the optical axis. The Z nodal offset would be the height up from the pan tilt intersection to the nodal point of the lens (usually very close to the iris ring on many lenses). See this image of a Top Mounted Camera on the end of a Bolt arm.



Global Offsets

Offset Properties (⚙): This button opens the Lens Setup dialog box where you can setup the nodal offsets and the measure offsets. Use the Lens Setup dialog box to set the values in this section.

Measure: Set this value as the distance between film plane and measure hook plane. Measure it from camera hook to the film plane.

Measure Optical: This is an offset used by target tracking to give the exact location of the target in space relative to the rig. It is the distance from the intersection of the pan and tilt axes to the measure hook measured parallel to the optical axis. For a snorkel lens, it is the distance from the measure point to the Snorkel Reflecting Point, this measurement is in the direction of view of the lens which is at 90 degrees to the normal direction.

Measure Normal: If the measure point is not close to the centre of the lens, then you tend to get a slight inaccuracy when measuring items which are very close to the lens. The handling for this is to tell the program how far the measure point is away from the optical axis. (This distance is measured normal to the optical axis). If you measure with a tape that is held parallel to the optical axis, then this value should be left as 0.0. If in doubt leave this value as 0.0.

5.7. Hand Held Box Setup

The hand held box is turned off while this setup is displayed so that buttons can be tested etc. without risk of moving the rig. Pressed buttons will turn red and the joystick values should be displayed as they change.

Since different users may prefer to use different controllers or different configurations on the controllers, you can save up to eight different setups. Each of these can have a totally different layout of controls or buttons allowing easy user to select the mode most comfortable to him.

Setup: Allows you to select between the 8 setup slots. The first is always “Default” and is the mode it starts in.

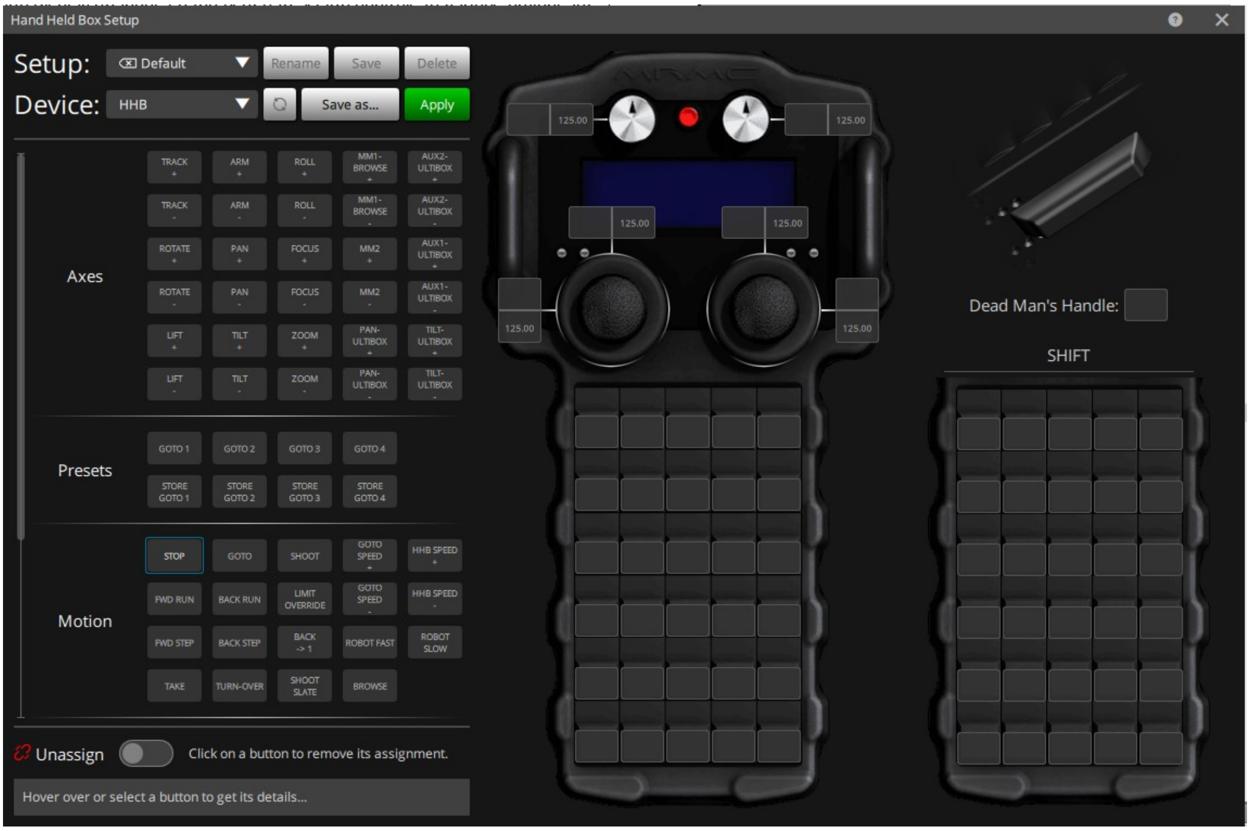
Name: The setup slots are called “Setup 1” through “Setup 7” but you can name these to identify them. Only the “Default” setup cannot be re-named.

Device: The Hand Held Box Setup display can be used to setup any controller, such as HHB, Broadcast Panel, LFP, and Mini Joystick. When any of these devices is selected, the display changes so the user can assign controls to various options for the selected controller.

Save As: Saves the current settings as the currently selected slot. Once you have named and set how you wish the Hand Held devices to operate, save these are your personal setup and you can recall it whenever you are using the system.

Delete: Allows you to delete any named setup saved by the user.

5.7.1. HHB



Hand Held Box Setup

Setup: Default

Device: HHB

Unassign — Click on a button to remove its assignment.

Hover over or select a button to get its details...

Dead Man's Handle:

When the Device is selected as HHB for the first time, the default setup appears. The buttons on the hand held box are assigned some default function. On the left are the functions that you can assign to the HHB keys, pots and joysticks on the right.

Pot 1, Pot 2

These refer to the two small knobs on either side of the Estop Button at the top of the hand held box (HHB). These can be assigned to control either the **Goto Factor** (See Miscellaneous Setup) or the **Hand Held Box Speed** which is displayed at the bottom of the Flair screen and also on the HHB display itself. For a lower value in Pot 1 will make the Gotos slower and a high value will make them faster. It will NOT change the speed of a goto while it executes, nor change the speed of a Goto that has already been calculated. (At “Ready to Goto”). You can assign Pot 2 to HHB Speed, and this will have a real time effect on the current HHB Speed. Note that when this pot is live, the other means of changing the HHB speed using keyboard keys of HHB buttons will override the current position of the pot. If the pot is moved after this, then the position of the pot will once again control the HHB Speed.

Joystick Assignments

The joysticks can be assigned to any axis that you wish simply by selecting the relevant box and then selecting the relevant axis from the left. If the joystick moves the axis in the wrong direction, simply change the sign of the axis number by entering a negative number. If an axis is assigned to a joystick, it will not then move on a button, even if one is assigned to it. Clearing out a value can be done by setting it to 0 by typing that or the “Del” key. 0 (or blank) means no axis will be moved by this joystick,

Keys

There are 2 sets of 30 keys laid out in 6 rows of 5 keys on the right of this display that imitates the handheld box. One set is the “Normal Keys” and the second set is the “Shift Keys”. The Normal Keys operate when the shift key is up and the Shift keys operate when the shift key is down. Any key can either be a motion key or a function key. A list of the available functions and axes is given on the display on the right hand side. If you want to assign a function to a key, select the key on the Handheld box drawing that you want to do a function, then select the function from the left and then select the key again on the Handheld Box on the right.

To unassign a function from a key, simply click the **Unassign** button, it would turn gree and then click the required key.

Deadman’s handle

This is one additional function key assigned to the “Deadman’s” handle which is an optional switch on the back of the MRMC HHB. This switch can be used to disable motion of the rig unless the switch is pulled. Any function can be assigned to this key, but use as a Deadman’s switch is the intended one. If the switch is pulled hard it will click again and this function is hard wired to drop the estop.

Axes

The keys can also be used to move axes. The keys can be assigned to an axis in the same manner as the joysticks are. Select the box for the key on the illustration and then select the axis. There is a difference with axes keys as opposed to the joysticks as you have to assign a positive direction key as well as a negative whereas the joysticks automatically handle both directions.

Presets

Use the Preset keys to store Gotos and then go to that position.

Motion

The keys can also be used to move axes. The keys are assigned to an axis in the same manner as the joysticks are. Click the function that you want to assign and then click the key on the right hand side that you want to assign the function to. There is a difference with motion keys as opposed to the joysticks as you have to assign a positive direction key as well as a negative whereas the joysticks automatically handle both directions.

Coordinates

These functions can be assigned to HHB keys to work in Cartesian mode.

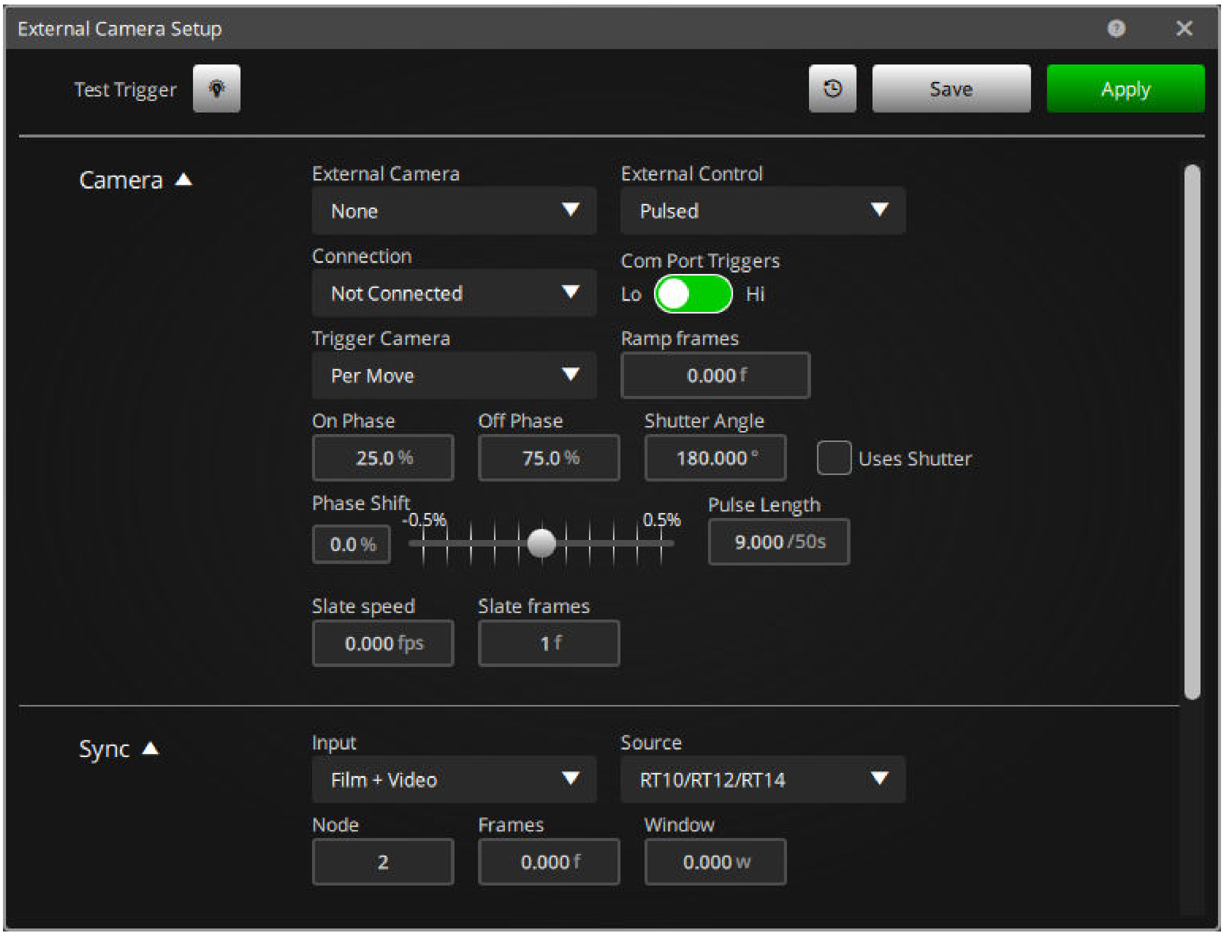
Miscellaneous

These contain some additional functions that can be assigned to the keys on the controller. One such function is the Shift key. This key is automatically assigned to the

5.8. External Camera Setup

5.8.1. Synchronisation

The setup allows the user to synchronise the system to an external video or film camera. If you are running the software as a video system, then this will be the usual operating situation for the system, and a video or synchronization signal from the camera (directly or indirectly) needs to be plugged into the back of the root box so that the system has a signal to synchronise to. When you are synchronising to an external film camera, then an input needs to be wired into the triggers input in the root box (contact Mark Roberts Motion Control for wiring assistance), and then the system can be synchronised to your external camera at any speed up to 300.0 given that your camera produces a stable regular pulse to be synchronised to.



External Camera
None

External Control
Pulsed

Connection
Not Connected

Com Port Triggers
Lo Hi

Trigger Camera
Per Move

Ramp frames
0.000 f

On Phase
25.0 %

Off Phase
75.0 %

Shutter Angle
180.000 °

☐ Uses Shutter

Phase Shift
0.0 %

Pulse Length
9.000 /50s

Slate speed
0.000 fps

Slate frames
1 f

Input
Film + Video

Source
RT10/RT12/RT14

Node
2

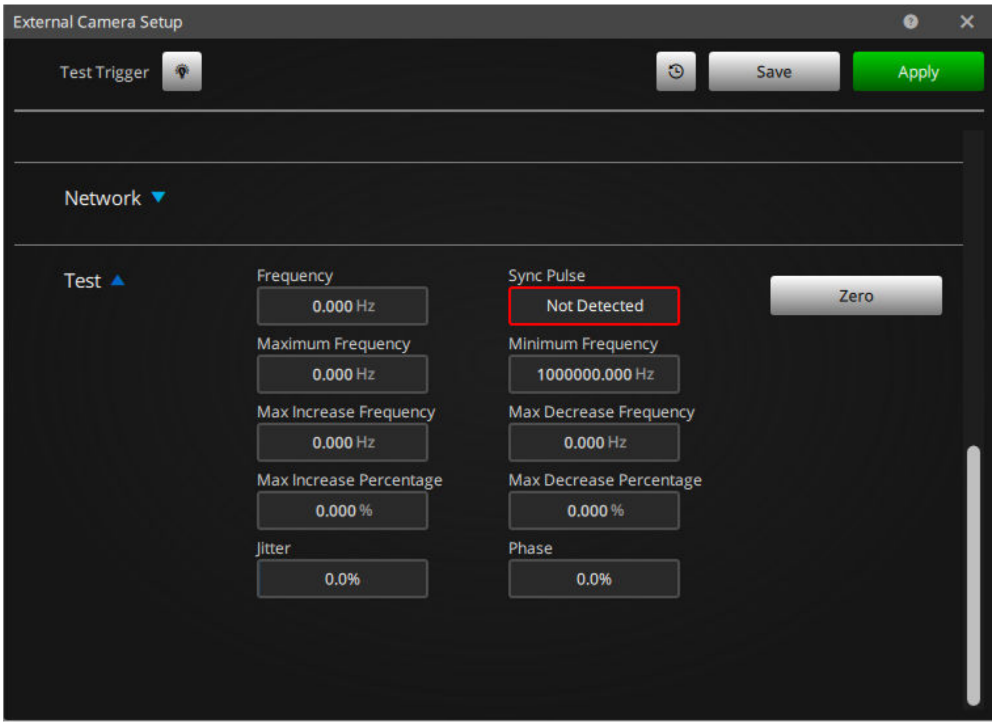
Frames
0.000 f

Window
0.000 w

Save, Apply & Exit: This dialog box supports the standard buttons for storing the setup data to disk, implementing any changes that have been made, or rejecting any changes.

Test Trigger : This button brings up a diagnostic box that allows you to study the incoming synchronisation pulses, and see if they are coming in, and what frequency they are arriving at. This can be used to determine the precise frequency of the incoming pulses, and to find out if they are coming in at all. The current incoming frequency will be displayed in pulses per second and a label will say **Sync Pulse** if pulses are coming in, and will be **No Pulse** in otherwise.

The current incoming synchronization frequency can be displayed on the run control bar and this is a quick and easy way to see if the external camera is running at the expected speed. See the section of customising the run control bar to add this control to your system.



Frequency: 0.000 Hz
Sync Pulse: Not Detected
Maximum Frequency: 0.000 Hz
Minimum Frequency: 1000000.000 Hz
Max Increase Frequency: 0.000 Hz
Max Decrease Frequency: 0.000 Hz
Max Increase Percentage: 0.000 %
Max Decrease Percentage: 0.000 %
Jitter: 0.0%
Phase: 0.0%

5.8.2. Camera

External Camera: This switch tells the system whether there is an external camera in use or not. When this switch is set to **ANY**, if the internal camera is disengaged and you try to run a move in record mode, the computer will know what you are trying to sync to an external camera. Other specific cameras may be added to the option list with later versions of the software. Other options you can select as **DSLR on S3**, **DSLR on Com** and **DSLR on Dragon**. In the latter three options, Flair can trigger a Digital Single Lens Reflex camera to take a frame via the S3 controller, via one of the control lines on a serial port or via the Dragon serial interface. These are not commonly used, contact MRMC for specific details. Mostly if you have an external camera, set this flag to **Any**.

External Control: Many cameras can be started and stopped by means of a simple switch closure or electrical pulse. This switch allows you to define the type of pulse the controlled camera expects. Some expect a single pulse to start and one to stop, others expect a switch to remain closed while they are supposed to run. If this field is set to other than **None**, then the computer will always trigger an external camera a number of frames earlier than the move, depending on the setting of **Ramp Frames** and **Sync Frames** below. This will affect the required pre-roll time for the move. The Types of trigger are **Continuous High** to hold the trigger on for as long as you want the camera to record. **Pulsed** To give it a start and a stop pulse. **One Pulse** to give it a trigger to take a single frame and stop. (Typically **One Pulse** is used when single framing with Digital Stills cameras).

Connection: This is for driving a camera directly from a serial port on the PC, this is rarely used and you should contact MRMC for more data is needed, but it uses one of the control lines in the serial port to trigger the camera, typically a DSLR. This shoud be set to “Not Connected”.

Trigger Camera: Modern digital cameras can respond to input triggers very rapidly and it is possible to trigger the camera to take every individual frame. Generally an external camera is triggered to start and the trigger is released or fired again to stop the camera. When single framing, this is also true, you give it a pulse to take a single frame and that would also be considered **Per Move** since you are giving it a single Pulse for every separate MOVE the rig does (Each move between frame positions). If you want to trigger the camera repeatedly to take individual frames as the move is running continuously (i.e. Shooting and NOT single framing) Then you can select **Per Frame**. This is most commonly done when you are trying to speed ramp with a digital Camera. It is not normally used otherwise.

Ramp Frames: You must enter here the number of frames the camera takes to get up to the speed that you are going to run the move at. This data will most likely be obtainable from the manufacturer, or by experimentation. This value is **0** for a **video system** or if you allow the motor to get up to speed before you trigger the move. In the case of a camera that is being triggered by the computer, this allows the camera to get up to speed before the move actually starts.

On Phase: The percentage through the frame that the “Per Frame” trigger is supposed to be raised at.

Off Phase: The percentage through the frame that the “Per Frame” trigger is supposed to be lowered at.

Shutter Angle: This can use used instead of the Off Phase if you wish to state the exposure in terms of a “Shutter Angle” rather than an “Off Phase”. 0.25 to 0.75 would equate to a 180 degree shutter angle. If your On Phase is later in the frame you might not be able to get as long a shutter angle as you expect and this will throw up a warning. This value is also used to convert between camera speed and exposure time.

Uses Shutter: This check box states whether your camera uses a shutter angle to convert between camera speed and exposure time. For example: a camera speed of 4 fps, gives a potential exposure time of up to 0.25 seconds. If the camera has a set shutter and Uses Shutter is selected, that exposure time is reduced but the fraction of 360 that the shutter angle is. This is seen when changing between camera speed and exposure time usually on the timelapse displays. If it is unchecked, the camera speed and exposure time are simply reciprocals of each other:

$$\mathtt{exp}osure = \frac{\mathtt{1.0}}{cameraspeed}$$

Phase Shift: This slider bar allows the user to adjust the timing relationship between the system and the incoming synchronisation pulse. This value is the amount of a frame between the start of the frame in the system, and the arrival of the sync pulse. This slider can be used to adjust the system so that when you shoot a TV screen, you can get the sync bars off screen. This value is immediately used by the system; you do not need to Apply the value. (Getting rid of the sync bars also depends on the camera shutter angle!) The units are in percentage of a frame +- 50%.

WARNING

With a slightly unstable pulse, you should not attempt to adjust the phase shift more than +/- 35 as the system may become unstable, and the move could be stopped.

Pulse Length (ticks): Where the output to control the camera is selected as a Pulse, this value sets the length of the pulse. It is measured in 50th of a second. Some cameras need a very short pulse but many need a long enough pulse for them to detect the pulse. Typical values are 10 ticks (1/5 of a second).

Sync settings

Input: This states which kind of incoming sync pulse is expected by the system. If this is **OFF** then the system will not try to sync. In **Film&Video** mode it will expect a sync input on the film sync input or video sync input (See Chapter 15 Synchronisation). **Interlaced** mode will convert a Video Trilevel signal into an edge for each field so you can sync at the interlaced speed (such as 59.952 for a 29.976 frame interlaced signal).

Source: The camera signal to which the rig needs to synchronize can come in to a number of different hardware points, ranging from the main computer to any of the axis nodes used in the system. This menu allows you to select where the signal comes in. The options are **RT10/RT12/RT14** if the signal is connected to the Control Box called an “RT10”, “RT12” or “RT14”. It can also come in to an axis node connected via a **LinkBoard** or a **NetworkBoard**. If the sync comes in on a Network Node, then you will also have to set the Type to **Network**.

Node: Defines which node the sync is connected to. Some controller boards can receive a sync input and some can’t. Contact MRMC for exact details on how your system is connected. Generally speaking sync will come in on an Ulti Board or similar and here you enter the Node number of that board whether it is on the Link Daisy Chain system or the Network System.

Frames: This is a buffer of a few frames to ensure that the system has locked sync before the move starts. It needn't be more than a couple of frames at most.

The external camera can be switched on using one of the general purpose axis triggers. The next three settings define where the trigger is wired. Contact MRMC if you need assistance in setting this up.

Window: Instability in the sync signal can cause the system to try to change speed to keep in time with the camera. This can have unwanted effects, and windowing the incoming values can help to smooth this out. 2 is a standard value to use so that the values is smoothed, but you may use up to 5 or 6 if needed.

Linked Triggers: The Nikon® series of DSLRs need to have two separate triggers to cause them to expose correctly. In this case, the first trigger is a “Fire Shutter” command and the second is a “Pull Focus” command. The “Pull Focus” trigger is left permanently on for the most part since we want to automatically maintain focus. However, the operating principle of the camera is that one must release auto-focus after taking a picture, before another picture can be taken. Therefore Flair must know that the behaviour of the “Pull Focus” trigger is **linked** to that of the “Fire Shutter” trigger.

One Shot Sync: Sometimes trying to synchronize the system to an unstable or noisy sync signal can cause instability in the motion and cause the camera to vibrate. In this case, if the speed of the camera and the speed of the system are exactly matched, then you can trigger Flair from a single pulse from the camera and then the systems should stay perfectly synced up for reasonably long periods of time. The exact drift is unknown, but for less than 1 minute it should be insignificant. If you are doing long takes, then this setting should not be used and if there is unstable sync a handling should be done to trace that down.

Unsynchronized Triggers: With Network Direct, the positions and control signals being sent to the Node can be buffered to allow for variable delays in the system. If however you are trying to grab shots with a remote DSLR system, you might not want to have any delays between when you pull the trigger and when the shutter fires. In this case, use Unsynchronised Triggers to ensure the camera responds as quickly as possible.

Network settings

Type: See the relevant section under setting up an axis. This will either be “Network” or “Link”.

Node: The Node number on the chosen network. See the section on this under axis setup for more details.

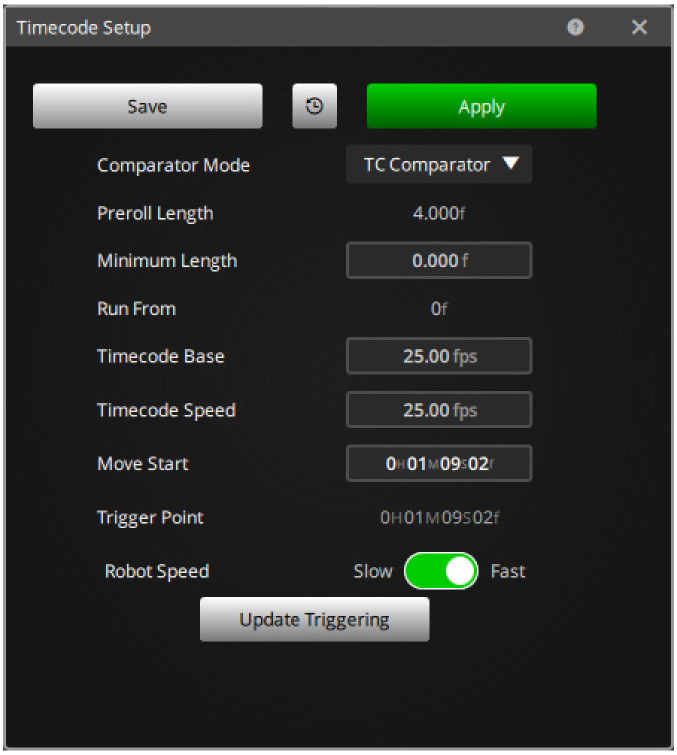
Port: As above, this assigns the port on the above assigned Node, which switches on the external camera.

> Note
>
> If you are shooting a video screen, due to the design of the camera and its shutter, the sync bars could be off screen when the screen is in one area of the shot, and back onto the screen when you are in another area of the shot. It is best to use a camera with a 180 degree shutter to minimise this effect. The reason for this is that the exposure time varies across the shot, and up and down due to the rotational motion of the shutter. This also applies to some Video or HD cameras with rolling not global shutters.

Test

See Test Trigger on page 53.

5.8.3. Timecode Setup



Comparator Mode: TC Comparator

Preroll Length: 4.000f

Minimum Length: 0.000 f

Run From: 0f

Timecode Base: 25.00 fps

Timecode Speed: 25.00 fps

Move Start: 0H01M09S02f

Trigger Point: 0H01M09S02f

Robot Speed: Slow / Fast

Available on the RT10/12/14 hardware. An OEM timecode comparator is available for use with the older style root boxes, but has limited functionality and cannot generate, only read timecode. (See Serial Devices – TCReader). More details on Timecode Setup covered in Timecode Setup on page 232.

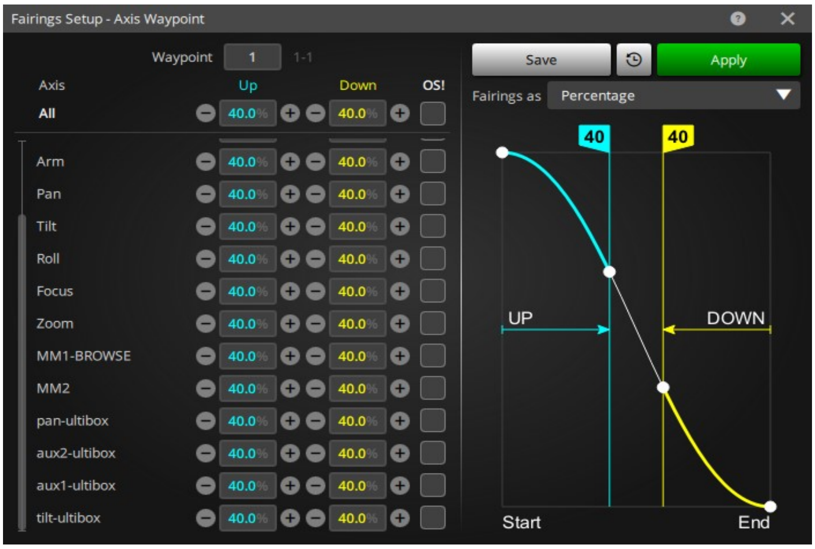
Comparator Mode: The options are "Off", "TC Comparator", "Store Trigger TC", "Store Start TC" and "Output Timecode". In Comparator mode, the timecode that is running is compared to the requested start timecode and if there is a match, the move is triggered at that point. "Store Start" means that when the "Shoot" button is pressed, the timecode at that time is stored and may be used later to trigger the move at the same time as it was manually triggered.

5.8.4. Fairings Setup

For a description of Fairings and how they work, see Chapter 7 Move Entry.

The **Setups** → **Fairings** menu allows you to select from up to 4 possible options. **Axis Default Fairing**, **Axis Waypoint Fairing**, **Cartesian Default Fairings** and **Cartesian Waypoint Fairings**. The Axis Default Fairings Setup allows you to modify the fairings for each axis for the entire move. The Axis Waypoint Fairings Setup allows you to specifically select which waypoint you wish to change the fairings on. Note that changes in the Axis Default Fairings will override any changes made earlier in the Axis Waypoint Fairings Setup. The second two options (Cartesian Default Fairings and Cartesian Waypoint Fairings) will only be available if your system supports Target Tracking. In this case you will have the same two options for changing the fairings on the XY&Z of the Camera path and the XY&Z of the target path for the whole move or for specific waypoints.

These displays are similar and are not shown individually. The main difference apart for the Axes listed is that in the default display, the Waypoint entry button is not shown; in the Waypoint display, the Save button is not present, and there is no fairing “Type” column.



Fairings Setup - Axis Waypoint

Waypoint 1 1-1

Fairings as Percentage

Axis	Up	Down	OS!
All	40.0%	40.0%	□
Arm	40.0%	40.0%	□
Pan	40.0%	40.0%	□
Tilt	40.0%	40.0%	□
Roll	40.0%	40.0%	□
Focus	40.0%	40.0%	□
Zoom	40.0%	40.0%	□
MM1-BROWSE	40.0%	40.0%	□
MM2	40.0%	40.0%	□
pan-ultibox	40.0%	40.0%	□
aux2-ultibox	40.0%	40.0%	□
aux1-ultibox	40.0%	40.0%	□
tilt-ultibox	40.0%	40.0%	□

Fairings as: This is a toggle which allows you to change between stating the fairings in **Frames** or in **Percentage** of the move.

Waypoint: This field allows you to specify which waypoint you want to change the fairings on. It is not shown in the default Fairings display, only in the **Axis Waypoint** display when this can be used to set which waypoint you are going to change the values on. If you have selected waypoint #2 then the Up value will change the way in which the motion leaves point 2 heading to point 3 and the Down value will affect the way in which the motion approaches point 2 having come from point 1. On the first waypoint, the Down value has no significance and on the last waypoint the Up value has no significance.

Up/Down: In these columns you state how many frames (how much percentage) of the move you want the fairing up or the fairing down to last for. In Percentage mode, permissible values range from 0 to 100, in frames mode, 0 to 999. Overlaps between the up and down fairings are truncated. Note: Very short up or down fairings will of course require larger accelerations.

Type: The type of a fairing affects the shape of the speed up or slow down curve. When 0 is selected, the axis will accelerate harder earlier in the fairing, and with 9 it will accelerate harder later in the fairing. The permissible values are 0 - 9, the middle value 5 is a symmetrical “S” shape.

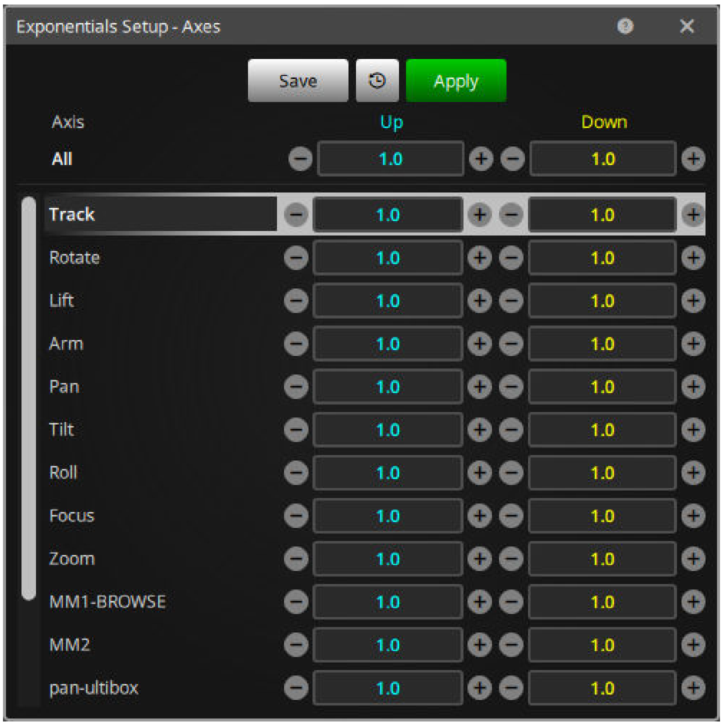
Allow Curve Reversal: This field indicates whether curve reversal handling is in operation for that axis at that waypoint. A “1” indicates that any curve reversal will be handled, a “0” indicates that no curve reversal handling will be done. See Sections later in this manual for a full description of curve reversal and its effect. This field is more easily changed through the Graphic display.

Apply: This button is pressed if you wish to apply the value in the “All Axes” box to the corresponding value for each axis.

Axes: Different fairings can be set for each axis, simply select the relevant field and type the required value in the box. The applicable box for each axis is listed next to its name. If there are more than about 8 axes, then a scroll bar will be provided to access the later axes.

5.8.5. Exponentials Setup

For a description of exponentials and how they work, see Chapter 7 Move Entry.



Axis	Up	Down
All	1.0	1.0
Track	1.0	1.0
Rotate	1.0	1.0
Lift	1.0	1.0
Arm	1.0	1.0
Pan	1.0	1.0
Tilt	1.0	1.0
Roll	1.0	1.0
Focus	1.0	1.0
Zoom	1.0	1.0
MM1-BROWSE	1.0	1.0
MM2	1.0	1.0
pan-ultibox	1.0	1.0

Up & Down: In these columns you state the relative proportions of the start and end velocities. For example, if you want the end of the move to be twice as fast as the start, you can enter 1.0 in the start, and 2.0 in the end. This is in fact identical in effect to typing 0.5 in the start and 1.0 in the end. Negative values are not accepted. Exponentials are turned off by entering numbers with a ratio of 1.0 to 1.0 in the start and end columns.

Apply: This button is pressed if you wish to apply the value in the "All Axes" box to all axes.

Axes: Different exponential ratios can be applied to each axis or as desired. Simply select the field you want, and type in the required value.

5.9. Inputs and Outputs Setup

See Chapter 16 Inputs, Outputs and DMX for information about using Trigger Inputs and Outputs.

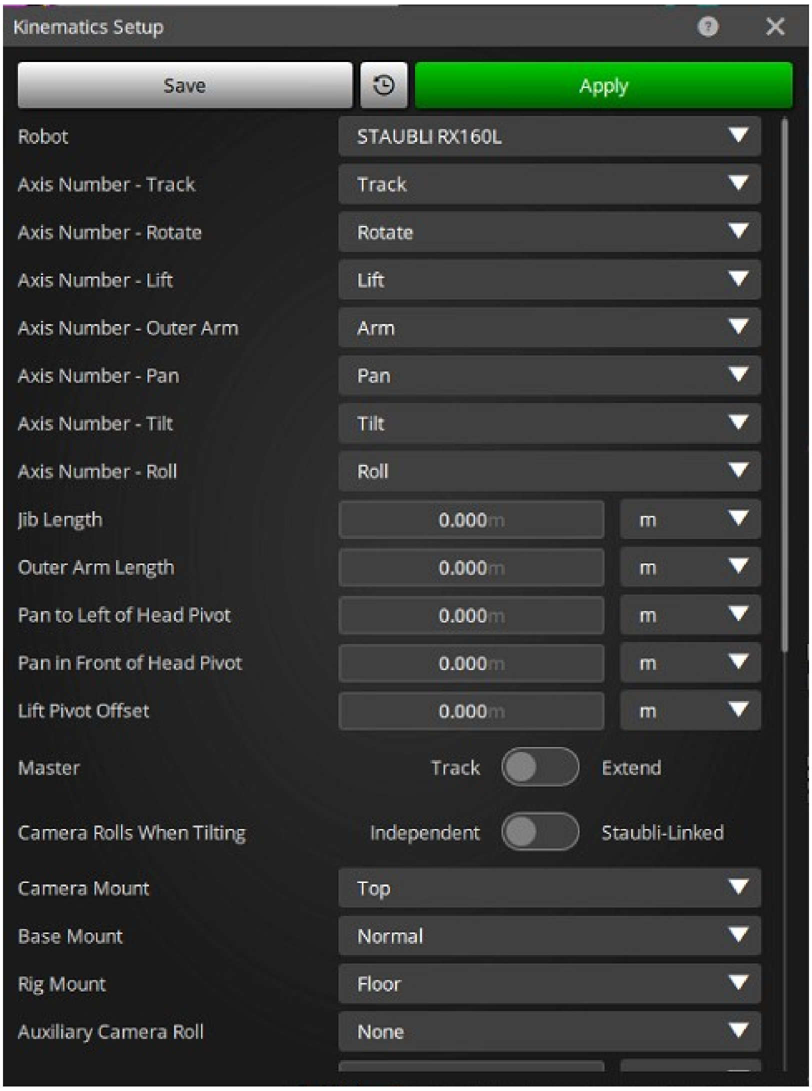
5.10. DMX Setup

See Chapter 16 Inputs, Outputs and DMX for information about using DMX Output control. This option will only be available if you have enabled it in the configuration file (Flair.ini) and have the appropriate hardware. Contact MRMC for details.

5.11. Kinematics Setup

There are currently many supported kinematics “models”. By this we mean the general configuration of the rig.

The exact layout of the Kinematics Setup display depends on the kinematics model you are using, but the principle is the same, the name of the parameter is listed on the left, and the value on the right. Many values are numerical distances which are always measured in metres - angles are measured in radians – contact MRMC for any needed parameters. If the parameter is an option, then there will be a pull down selection on the right. Shown here is part of the Cyclops kinematics setup which is used for the Cyclops rigs and the Milos.



Robot	STAUBLI RX160L	
Axis Number - Track	Track	
Axis Number - Rotate	Rotate	
Axis Number - Lift	Lift	
Axis Number - Outer Arm	Arm	
Axis Number - Pan	Pan	
Axis Number - Tilt	Tilt	
Axis Number - Roll	Roll	
Jib Length	0.000m	m
Outer Arm Length	0.000m	m
Pan to Left of Head Pivot	0.000m	m
Pan in Front of Head Pivot	0.000m	m
Lift Pivot Offset	0.000m	m
Master	Track ○ Extend	
Camera Rolls When Tilting	Independent ○ Staubli-Linked	
Camera Mount	Top	
Base Mount	Normal	
Rig Mount	Floor	
Auxiliary Camera Roll	None	

Rigs are grouped into a kinematics model category and then into a Rig Type sub category depending on the exact design. The Model is based on the required mathematical solution for its design, and the Type is the specific Rig. For example, the Titan, Cyclops and Milo are all mathematically solved the same way, but are physically very different rigs. The configuration file defines the Type and the required kinematics model is derived from this. Do not be surprised if you set up a Technocrane type rig and is comes up as “Cyclops” model. It uses the same math, but it is NOT a Cyclops. The model name is somewhat historical and can be misleading.

The supported rig models and types are:

Cyclops Model (Covers most large rigs)

Rigs: Cyclops, Titan, Milo, Animoko, Technocrane, Crane, Panther, Modula, Talos, Juno

Simple Dolly Model. (Has no rotating arm)

Rigs: SimpleDolly and Sliding Head

Hexapod Model (6 axis parallel system)

Rig: Hexapod

Bolt Model

Rigs: Robot, Staubli, Staubli_Rx60, Staubli_Rx160, Staubli_Rx160L,Staubli_Tx90, Staubli_Tx90L, Staubli_Tx90XL, Kuka_KR16_2, Kuka_KR30_3, Kuka_KR360_3. (Others may be added as their rigging is implemented) Motoman_MH50-20_II, Motoman_GP20HL

CYCLOPS is named after the Large Mark Robert's Motion Control Rig whose basic configuration is a track upon which there is a rotate, on this there is a lifting arm which can extend, at the end of the arm is a secondary shorter arm which ends in a pan and tilt head. (This configuration also works for the Milo and the configuration can be called MILO).

PANTHER is named after the Panther dolly. This one has a track, rotate and lifting jib arm and a head that is automatically kept level, and a pan and tilt head.

SIMPLE DOLLY; this is similar to the panther except that there is no means of moving the camera across the track side to side, as you would have with a rotating jib arm. This target tracking model does not do true target tracking and will not output true XYZ data, but it will follow the target! The camera can be on a jib and may be positioned off of the central column of the dolly.

ORTHOGONAL refers to a rig where the track, lift and side-to-side movements are all linear and connected orthogonally (at right angles). This is also sometimes called a crucifix rig.

HEAD is a simple pan and tilt head. The only reason it has a kinematics setup is for the import and export of data as the operation of such a unit does not really require target tracking. The setups are simply Pan, Tilt, Roll and Focus.

OFFHEAD is almost exactly the same as the panther model except that the axis of tilt is not aligned to the axis of pan. This offset is added to the kinematics setup and is otherwise the same as the Panther Setup.

OVERHEAD is used for large rigs that are similar to the Cyclops but also have a side-to-side motion of the turret. These are normally mounted overhead - hence the name.

TABLE is used for animation tables where you wish to be able to look at and modify the camera as a path of motion. Setups are self-explanatory and are not covered here in detail.

JUNO is similar to the panther in basic operation, but the pan axis can be tilted and locked in different positions. It also has the option for an extending arm, and this makes this model more complex.

ANIMOKO converts internally to JUNO model.

TALOS refers to a rig with similar kinematics to a Juno but to make without the ability to set the head at different angles, only vertically up or down.

TECHNOCRANE refers to a rig with the same shape and axes as a standard Technocrane.

MULTIHEAD Multiple heads that can be operated together to track a single or multiple targets.

HEXAPOD (or MOTIONBASE) refers to a rig that consists of a platform with 6 actuators that can move in 6 degrees of freedom. It is generally used for moving models or sets and not for moving the camera. It can have a rotator on the bottom or the top for additional functionality.

ROBOT Staubli (Bolt) is originally an industrial robot arm which was converted to a motion control arm; it was modified to be controlled from FLAIR. Other industrial robots are also supported. Kuka models and Yaskawa/Motoman

- All parameters are measured in metres where applicable or radians.
- Some parameters are common to many models such as the Axis Numbers and the Camera Height at Zero so they are described here:

Axis Numbers: The program needs to know which axis provides which motion. Select from the listed axes names on the right, which one performs the motion described on the left hand side. If you have any questions, contact Mark Roberts Motion Control as an error in this setup will prevent Target Tracking from working. If an axis does not exist on your system, then select “Not Installed”.

Auxiliary Axes: Aux XV, Aux YV, Aux ZV, Aux XT, Aux YT, Aux ZT: These are axes which can be used to adjust a Target Tracking move. Their use is fairly advanced and covered under the Target Tracking section of the manual. One simply selects which axis is being used for the function or selects “Not Installed” if you are not using the axis.

Axis number MM Pan, MM Tilt, MM Roll: If the rig is looking at model mover with up to 3 degrees of freedom, then the rig can be locked to the model mover and when the model mover is adjusted, the rig stays in the same relative position to the item on the model mover. If you do not have a model mover installed or you are not using this function, these MUST be set to “Not Installed”.

Model Mover X Position, Y Position, Z Position: These define the position for the model mover relative to the 0,0,0 of the rig. This is normally in the centre of the turret on the floor if the rig has been set up correctly. Only relevant if you are using the model mover with the rig locked to it.

Camera Height at Zero: In order to make the 3D views provide a comprehensible representation of the studio; the rig needs to know where the floor is. This value is actually of no use to the system, but it does make the numbers easier for the user and interfacing with other systems. This value is set to the height of the Nodal Point off the floor when the rig is at its zero position.

Tilt 90 Zone: This is the range of tilt position either side of tilt -90 that the computer has to use special maths handlings in order to operate correctly in Target Tracking. The basic problem is that the camera pan axis acts only as a roll when you are at tilt -90 (also +90 though that is rarer). The Tilt 90 handlings are developmental as of version 6.40 and cannot be expected to operate entirely correctly.

Lift Column Flag: See section below on lift models.

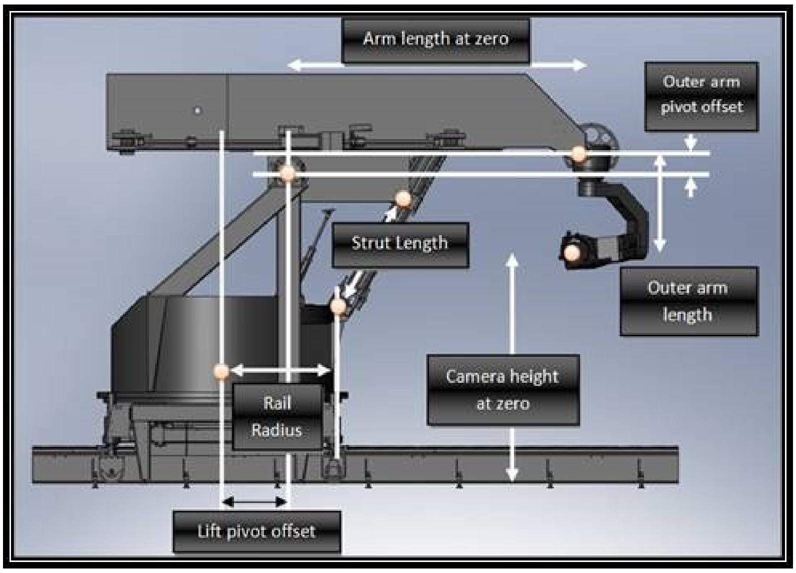
Camera Mount: Standardly with a motion control rig, the optical axis of the camera points parallel to the axis of rotation of the last axis, Roll. This is not necessarily the only way it can be done, but normally that gives the best range of travel. With some of the Robots and with some other specialized cases, other mounting options can be used. However this is most commonly set to “End Mount” and should not be changed without consultation with MRMC. Note that this setting affects the way in which the Horizon is controlled (See Roll Relative section in Jobtype). The mounting options are “End, Top, Bottom, Left and Right” The best way to visualize these is to take the rig to 0 on ALL axes and then look to see which side of the camera is facing the last axis of the rig. Normally with a system, the back END of the camera is facing the rig. With Top mount, the camera’s top surface will face the end of the rig and the camera will be looking up. This is similar to “Bottom” mount and the camera will also be looking up but the bottom of the camera will be “facing” the rig.

Head Order PTR, PRT: This is the order to the rotational axes on the head. Most Heads are PTR meaning Pan, then Tilt and then Roll. Some heads are a different order, specifically the Libra III head which has the roll axis before the tilt. In almost all cases, this should be set to PTR.

Arm Length at Zero: The length of the arm when the extend is at zero. This is the distance from the lift pivot forward to the outer arm pivot measured horizontally when the arm is horizontal. If there is no extend axis, then this value is simply the length of the arm.

As well as each rig having a different kinematics model, each one can also have a different lift mechanism, for example a Cyclops can have a motor that rotates the jib arm about a pivot to lift the arm (called an angular lift) or it can have a strut that extends from the base to lift the arm (called a strut lift). Following the description of the kinematics models there is a description of the different lift options, which are applicable to some rigs.

Cyclops Model



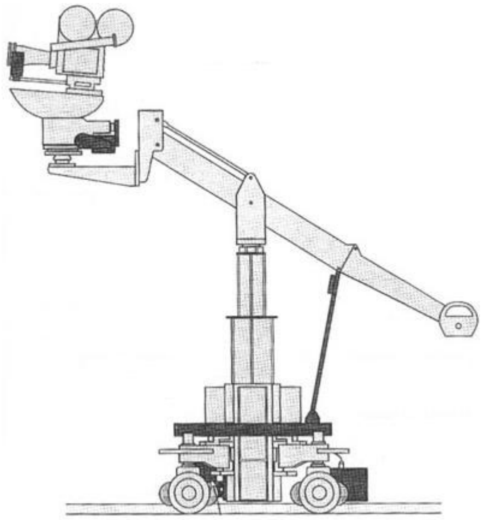
Outer Arm length: This is the length of the outer arm, also called the Angle or Pitch. The distance is from the axis about which it rotates to the tilt axis. This value may be used to assist in drawing the head in the Rig Model display.

Outer Arm Pivot Offset: The vertical height difference between the lift pivot and the out arm pivot when the arm is horizontal. It is measured up from the lift pivot to the outer arm pivot, and this value can be negative. This value is usually about 8cm i.e. 0.08 metres.

Lift Pivot Offset: The horizontal distance of the lift pivot forward from the centre of the rotate. The absolute height of the lift pivot is irrelevant as far as the kinematics is concerned as it is a constant and does not change when the axes move. (Typical value with a Milo is -0.02, i.e. 20cm behind the rotate centre)

Master - Track/Extend: When a rig has both an Extend and a Track axis, these axes can both produce the same sort of motion. In order to resolve possible ambiguities produced by this, one of the axes is set as a master axis; since it could be either, this flag can be changed to indicate accordingly. There are advantages and disadvantages to each one. Normally it is better to have the Extend as the master axis as that allows more movement of the track and rotate when target tracking and this is done by setting the Master Flag to “Extend Master”. If you have no Extend axis, then this flag has to be set to “Extend Master”.

Panther (or Modula)



For the first section of the Kinematics Setup, see the section headed 'Lift Models' further below.

Simple Dolly

This is in fact almost identical to the Panther Setup. The only differences are that there is no rotate axis, it is replaced by the addition of the jib position flag.

Jib Position NWES: This indicates the presence and position of the jib. If there is no jib, and the pan tilt head is mounted directly on the lift column or simply at a fixed height, this value should be set to NONE. Note that regardless of the jib position, the pan zero is looking up the track in the positive track direction.

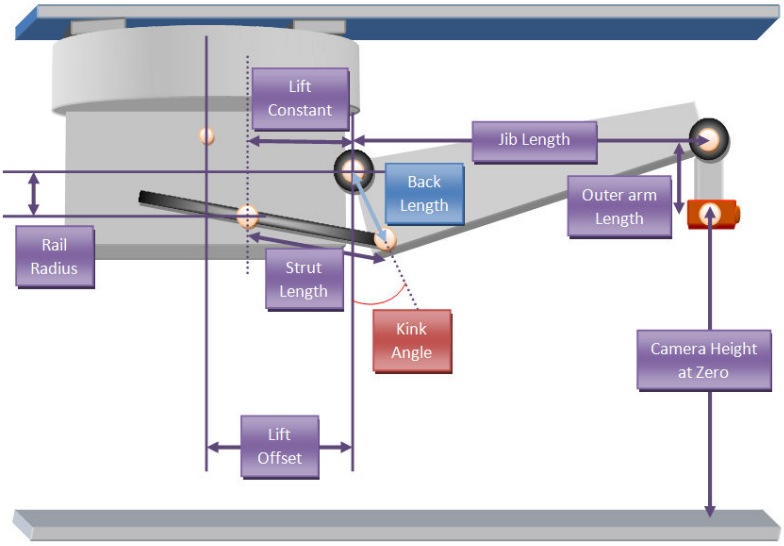
Orthogonal

Camera Side Offset at 0: This is the distance from the centre of the track sideways to the optical axis. It is relatively unimportant and can be set to 0.0. It allows you to set the side to side zero for XYZ output.

Overhead Rig

Outer Arm length: This is the length of the outer arm, also called the Angle or Pitch. The distance is from the axis about which it rotates to the tilt axis.

Lift Pivot Offset: The horizontal distance of the lift pivot forward from the centre of the rotate. The absolute height of the lift pivot is irrelevant as far as the kinematics is concerned as it is a constant and does not change when the axes move.



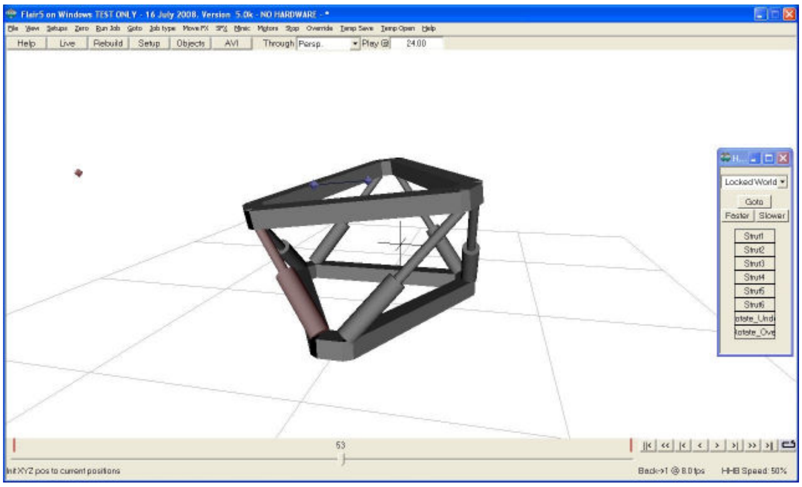
Juno (or Talos or Technocrane or Animoko)

Lift Pivot Offset: The horizontal distance of the lift pivot forward from the centre of the rotate. The absolute height of the lift pivot is irrelevant as far as the kinematics is concerned as it is a constant and does not change when the axes move.

Head Angle: This is the angle that the head has been mounted at to the end of the arm. It is normally 0, but when the head is tipped, it will change in multiples of 45 degrees.

MotionBase

Also known as a Stewart Platform, 6-DOF platform or a Hexapod.


Flair5 on Windows TEST ONLY - 16 July 2008. Version 5.0k - NO HARDWARE - *

Through Persp.

Play 24.00

Locked World

53

Init XYZ pos to current positions

Back->1 @ 8.0 fps

HHB Speed: 50%

Strut 1 through 6: These are the strut axis numbers in an anti-clockwise direction starting to the left of the X axis looking from the top. The X axis runs forward from a point on the bottom triangle and a side on the top.

Rotate Under: The axis number of the rotator that is underneath the motion base. If there is none, select “Not Installed”.

Rotate Over: The axis number of the rotate that is on top of the rotator. If none, select "Not Installed".

Strut Length at zero: The length of the struts when at zero. They must all be the same.

The legs on a hexapod are arranged with 2 legs meeting at the top and at the bottom close together, and each pair of legs spaced further apart. These gaps define the structure of the rig:

- Base Short Knuckle gap
- Base Long Knuckle gap
- Top Short Knuckle gap
- Top Long Knuckle gap

Top Height @ Zero: The height of the top of the rig when the legs are at zero.

Base Height to pivot: Height from the bottom of the base to the lower pivots.

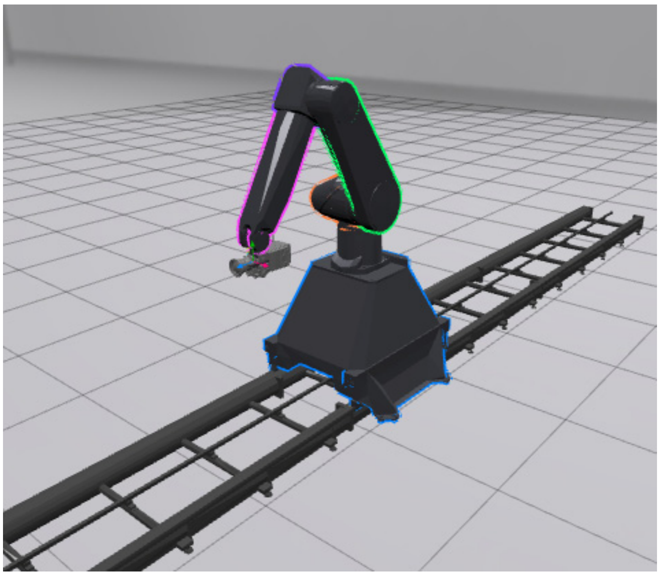
Top Height from pivot: Height from the upper pivot to the top of the rig.

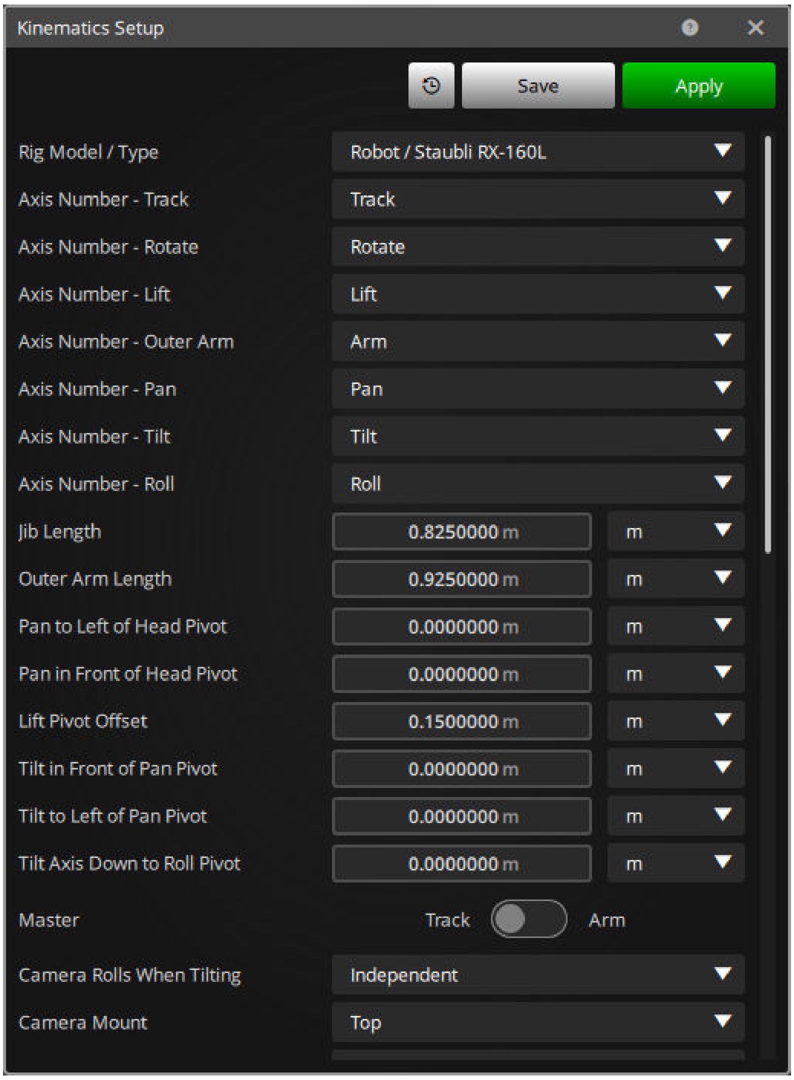
X offset from centre top: The offset from the centre of the rig to the "nodal point"

Y offset from centre top: Side offset to the "nodal point"

Z offset from centre top: Vertical offset from the top of the rig to the "nodal point"

5.11.1. Robot Kinematics Model





Rig Model / Type	Robot / Staubli RX-160L	
Axis Number - Track	Track	
Axis Number - Rotate	Rotate	
Axis Number - Lift	Lift	
Axis Number - Outer Arm	Arm	
Axis Number - Pan	Pan	
Axis Number - Tilt	Tilt	
Axis Number - Roll	Roll	
Jib Length	0.8250000 m	m
Outer Arm Length	0.9250000 m	m
Pan to Left of Head Pivot	0.0000000 m	m
Pan in Front of Head Pivot	0.0000000 m	m
Lift Pivot Offset	0.1500000 m	m
Tilt in Front of Pan Pivot	0.0000000 m	m
Tilt to Left of Pan Pivot	0.0000000 m	m
Tilt Axis Down to Roll Pivot	0.0000000 m	m
Master	Track ○ Arm	
Camera Rolls When Tilting	Independent	
Camera Mount	Top	

The Rotational axes on a robot are mostly lined up so that the pivot of one axis co-incides with the pivot of the previous one. For some specific model, this is not the case, and the distance from the Pivot of one axis to the pivot of the next needs to be set. This is best resolved in consultation with MRMC, but the following offsets are related to this.

Pan to left of Head Pivot

Pan in front of Head Pivot

Tilt in front of Pan Pivot

Tilt to left to Pan Pivot

Tilt axis down to Roll Pivot

If in doubt, set to 0.0 and get exact data from MRMC.

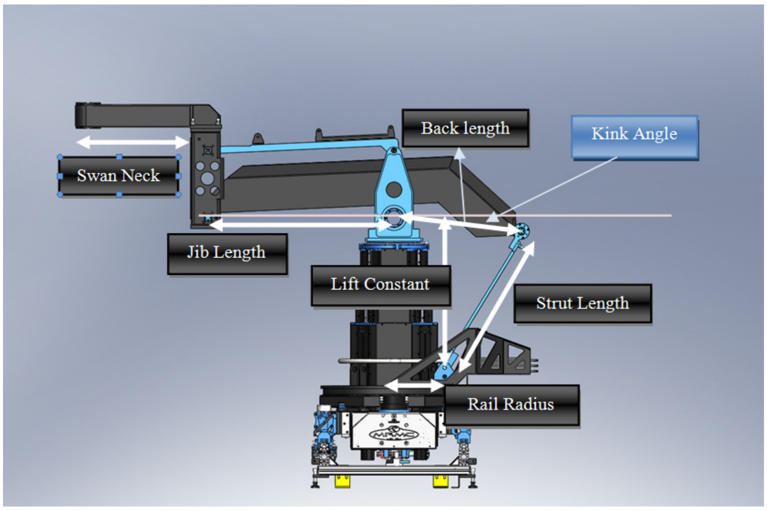
Camera Rolls when Tilting: Most robot arms run the last axis the “roll” through a special coupling that runs inside of the penultimate axis, the “tilt”. This is done for weight and stability reasons but if you are running the robot axes directly and NOT through a robot controller, it is possible, you will have to account for the extra motion that the tilt axis applies to the roll axis. That is to say that if you move the tilt, the roll also moves, so to keep the roll still, you have to compensate with exactly the opposite of the tilt to make the roll stay still.

5.11.2. Lift Models

There are nine lift models. In all cases, the lift axis raises an arm on which the camera is mounted, but in different ways. The different lift models are described here.

Panther Lift	Panther® Dollies (Similar to the lift mechanism on some early Modular rigs).
Column Lift	Vertical Lift Mechanisms.
Strut Lift	Milos and arms lifted by a strut.
Angular Lift	Cyclops or angularly driven lift mechanisms.
Juno Screw	Similar to the Milo but has an extra offset.
Fisher® Lift	Specialized lift for encoding the fisher lift using an encoder on the pivot of the arm.
Sliding Pivot	Similar to the Fisher lift but uses a tilted slide to drive the lift.
Hybrid Dolly	Chapman Leonard ® hybrid dolly.
Reverse Strut	Like the Strut lift, but operates on the back of the arm.

Panther lift: (The model shown below is of a Modular Rig)



Lift Constant: This is the height of the lift pivot above the rotate ring when the arm is level. This is the required position for the lift zero. It is measured vertically from the arm pivot to the lower swing strut pivot.

Rail Radius: The effective radius of the rotate ring at the bottom strut pivot.

Strut Length: The length of the strut from the rotate ring to the back end of the arm.

Kink Angle: The angle (in radians) between the front arm and the back arm. Measure from the pivot points. Contact MRMC if you have any difficulty setting this value.

Jib Length: The length of the jib from the central pivot on the arm to the front pivot.

Back Length: The length of the arm backwards to where it connects to the strut from the main arm pivot.

Swan Neck: The length of the swan neck from the front arm pivot to the centre of the pan's rotation. This is a horizontal distance, the vertical height is a constant and does not affect the kinematics (The vertical height is absorbed into the Camera Height at Zero).

Lift Offset: The distance from the centre of rotate forward to the lift pivot. On an actual Panther Dolly this value is 0.0.

Column lift:

Lift Constant: Not needed, should be set to 0.0.

Rail Radius: Not needed, should be set to 0.0.

Strut Length: Not needed, should be set to 0.0.

Kink Angle: Not needed, should be set to 0.0.

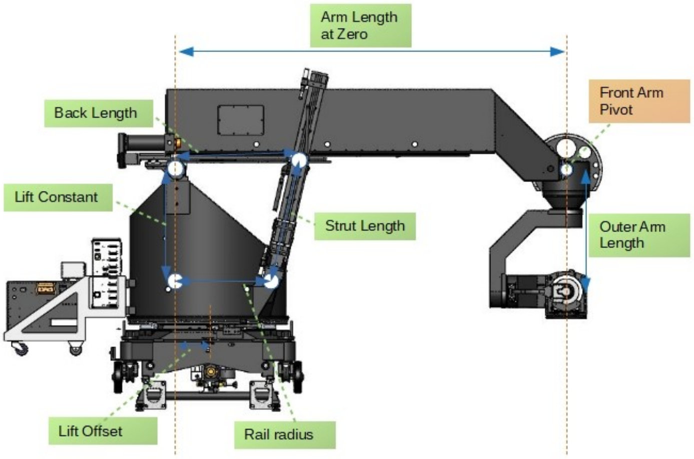
Jib Length: The length of the jib from the centre of rotate to the centre of pan measured horizontally, vertical dimensions are irrelevant.

Back Length: Not needed, should be set to 0.0.

Swan Neck: Not needed, should be set to 0.0.

Lift Offset: The distance from the centre of rotate forward to the lift pivot.

Strut Lift:



Lift Constant: This is the height of the arm pivot above the lower lift pivot.

Rail Radius: The horizontal distance from the arm pivot to the lower lift pivot.

The vertical portion of this is the Lift Constant above.

Strut Length: The length of the strut from the lower lift pivot to the upper lift pivot when the front arm is horizontal. This is the required position for the lift zero.

Kink Angle: The angle (in radians) between lines drawn from the main arm pivot to the upper lift pivot (on the strut/ball screws) at zero and from the main arm pivot to level. Contact MRMC if any difficulty setting this value. This is not shown on the diagram. It is a small angle, usually a negative number like -0.04 radians on a Milo.

Arm Length at Zero: The length of the jib from the arm pivot to the front arm pivot.

Back Length: The length from the arm pivot to where it connects to the lift strut.

Swan Neck: The length of the swan neck from the front arm pivot to the centre of the pan's rotation. This is a horizontal distance; the vertical height is a constant and does not affect the kinematics. This is only used for Panther systems; it does not affect Milos or Cyclops, and should be set to 0 in these cases.

Lift Offset: The distance from the centre of rotate forward to the lift pivot. With the Milo this value is usually about -0.20 and with the Cyclops it is about -0.55. This value is negative as the lift pivot is behind the centre of rotate.

Angular lift:

Lift Constant: Not needed, should be set to 0.0.

Rail Radius: Not needed, should be set to 0.0.

Strut Length: Not needed, should be set to 0.0.

Kink Angle: Not needed, should be set to 0.0.

Jib Length: The length of the jib from the centre of rotate to the pivot point for the head. Lift Zero would be when the arm is horizontal with the centre and front pivots at the same height.

Back Length: Not needed, should be set to 0.0.

Swan Neck: The length of the swan neck from the front arm pivot to the centre of the pan's rotation. This is a horizontal distance; the vertical height is a constant and does not affect the kinematics.

Lift Offset: The distance from the centre of rotate forward to the lift pivot.

Juno Screw: Identical to the Strut Lift except there is an offset called the Ballscrew Offset. Contact MRMC to set this value, it is not something that can easily be measured.

Fisher Lift: The Fisher Lift zeroes with the front arm horizontal and that is where the arm must be when these offsets are measured.

Lift Constant: The height of the main pivot above the front pivot when the arm is horizontal. On a Fisher 11 this is 0.0

Rail Radius: This relates the slider to the front pivot and is the vertical height that a line from the slider would pass above the front pivot. On a Fisher 11 this is 0.0102

Strut Length: The length of the strut from the front of the dolly to the middle of the lift arm. On the fisher 11 this is 55mm = 0.55 m.

Kink Angle: The angle (in radians) between lines drawn from the main arm pivot to the front head pivot and the line from the main arm pivot to the slider pivot. Contact MRMC if any difficulty setting this value. It is a small angle, 0.0759 radians on a Fisher 11.

Jib Length: The length from the centre arm pivot to the front head pivot. On a Fisher 11 this is 0.64m

Back Length: The length from the arm pivot to the pivot where it connects to the slider. 0.55 same as the Back Length on a Fisher 11.

Swan Neck: The length of the swan neck from the front arm pivot to the centre of the pan's rotation. This is a horizontal distance; the vertical height is a constant and does not affect the kinematics. In some cases this will be 0.0 if the head is mounted directly above the pivot.

Sliding Pivot: On the Fisher dolly, the back pivot of the arm runs on a slider mounted inside the dolly. This slider is angled from the horizontal by a small amount. On the Fisher® 11 it is 0.048211 radians (2.76 degrees)

Hybrid Dolly: Contact MRMC for specifics

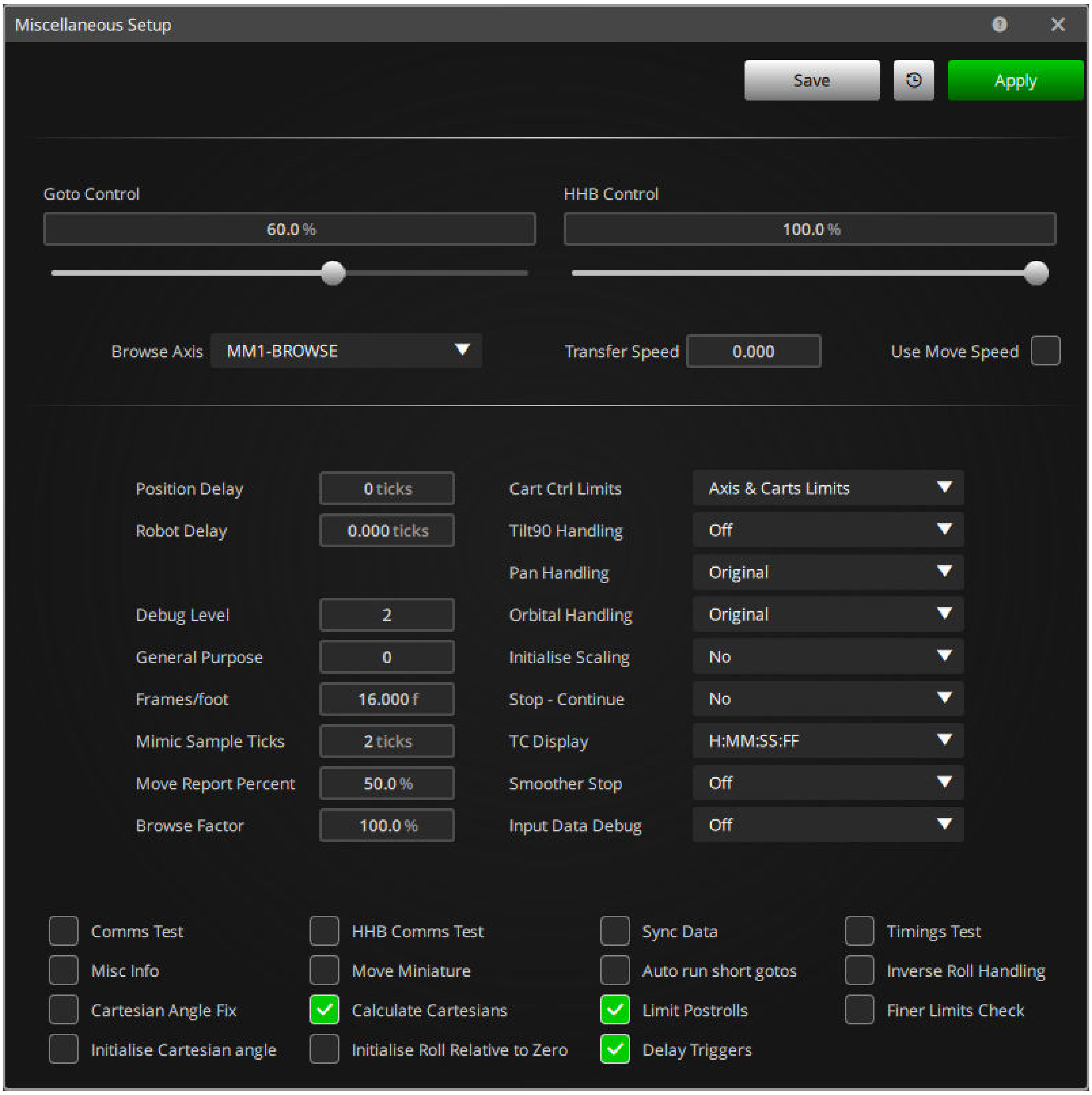
Reverse Strut: Is actually implemented by reversing the value of one of the parameters. Contact MRMC for details.

5.12. Miscellaneous Setup

The Miscellaneous Setup allow the user to change some miscellaneous control variables and certain setups. The latter should be usually used after consultation MRMC. However, it is recommended that you study this section to see if any of the options are of interest and so that you can use them if needed.

The control variables that you can change are **Goto Contro**l and **HHB Control**. Both of these comprise a slider which will go from 1 to 100 percent. They affect the speed at which Gotos are performed and the speed at which the Hand Held Box operates. One would normally set up each axis to perform Gotos and run on the Hand Held box at acceptable speeds. If, however you were working on a small model and needed to move very slowly on the hand held box, then you can reduce the overall speed of the Hand Held Box to a fraction of its normal top speed. If you moved the slider to 50, then the hand held box would only operate at half of its normal top speed. The “Slow” and “Fast” buttons will still operate on the HHB, but in this reduced range giving you more accurate control of the rig. This is true in a similar way for the Goto Control. If you are in a tight situation and you want to be extra careful with Gotos, their speed can be reduced with the slider to make Gotos execute at a fraction of their normal speed. This is particularly useful if you are working around actors who are unfamiliar with the rig and get worried if it moves very fast.

The Goto Factor can also be controlled via a small knob on the Hand Held Box. There is also control on the Hand Held box for the HHBSpeed, but this is NOT the HHB_Factor.



Goto Control 60.0 %

HHB Control 100.0 %

Browse Axis MM1-BROWSE

Transfer Speed 0.000

Use Move Speed ☐

Position Delay	0 ticks	Cart Ctrl Limits	Axis & Carts Limits
Robot Delay	0.000 ticks	Tilt90 Handling	Off
		Pan Handling	Original
Debug Level	2	Orbital Handling	Original
General Purpose	0	Initialise Scaling	No
Frames/foot	16.000 f	Stop - Continue	No
Mimic Sample Ticks	2 ticks	TC Display	H:MM:SS:FF
Move Report Percent	50.0 %	Smoother Stop	Off
Browse Factor	100.0 %	Input Data Debug	Off

☐ Comms Test
☐ Misc Info
☐ Cartesian Angle Fix
☐ Initialise Cartesian angle
☐ HHB Comms Test
☐ Move Miniature
☑ Calculate Cartesians
☐ Initialise Roll Relative to Zero
☐ Sync Data
☐ Auto run short gotos
☑ Limit Postrolls
☑ Delay Triggers
☐ Timings Test
☐ Inverse Roll Handling
☐ Finer Limits Check

Browse Axis: This selects which axis is designated for moving the rig through a move. It is not recommended to select a large axis as you do not want to move this axis when you think you are going to browse along the move. A good axis to designate would be the zoom or similar.

Transfer Speed: This sets the computation that converts frames into seconds. The default value for this is 24.0 fps and so 1 second on screen is 24 frames. Regardless of the shooting speed, film is normally transferred at 24, 25, or 30 so that the times displayed on the numeric screen are not shooting time, but screen time. If this value is set to 0, then the current film speed will be used as the transfer speed. If you check the **Use Move Speed** box, then this value is set to 0 and the Move Speed is used to convert frames into seconds.

The above four settings can be saved by clicking the **Save** button; however, the following are not so saved. Some of them may be set to new defaults through adjustments to the file “Flair.ini. For more detail.

5.12.1. Numerical Boxes

Position Delay: Adjusts the delay between calculating positions and sending them to the axes boards. The units are in ticks (1/50ths of a second) and this feature is used when exporting real-time XYZ data so that the video images can be processed before the rig actually arrives at the location. If you do not understand this, don't change this value before contacting MRMC.

Debug Level: This sets the current debug level for the software and this determines how much operational information is printed out in the Flair log file. You should not normally change this unless so directed by MRMC personnel.

General Purpose: This is for diagnostic purposes and should not be touched.

Frames/foot: This sets the speed at which the footage of film is adjusted based on the camera speed. Standard 35mm is 16 frames per foot, but 16mm is different as is vista-vision etc.

Mimic Sample Ticks: When recording the motion of an axis for playback, the position of the axis is usually stored every second system "Tick" (Normally 1/50th of a second) If you are doing a very long recording, you may increase this sample period to 4 or 6 without losing any significant accuracy but you will reduce file sizes significantly.

Cart Ctrl Limits: When manipulating the robot in Cartesian space, we can sometimes run into difficulties since the inverse mapping from Cartesian space to physical axes can, in certain situations, become a problem. These problem positions of the robot are referred to as singularities.

A Boundary Singularity is where the position in Cartesian space is beyond where the robot can be physically positioned.

An Internal Singularity is generally caused by an alignment of the robot's axes in space. For example, if 2 axes become aligned in space, rotation of one can be cancelled by counter-rotation of the other, leaving the physical axis location indeterminate. At such a singularity, small Cartesian motions may require very large joint velocities.

There are four Cart Ctrl Limit states.

- **Off:** No checking for singularities. Soft limits of the physical axes and Cartesian space limits are all ignored. If the operator takes the rig to a singularity then the axes will trip.
- **Axis Limits:** Soft limits of physical axes are respected and physical axis velocities are monitored and used to limit demanded Cartesian velocities. The operator is protected from axes tripping if he takes the rig to a singularity.
- **Axis & Cart. Limits:** Soft limits of physical axes, physical axis velocities, and soft limits of cartesian space (as specified in Cartesians Setup) are all respected.
- **Diagnostic mode:** Axis position and velocity monitoring is reported in the Rig Safety Limit dialog box.

Tilt90 Handling: "Off/Detect/Handle". For Tilt 90 Target Tracking corrections, this must be set to "Handle".

Pan Handling: "Original/Flip Tilt/Flip Pan, Pan & Tilt" When you are in target tracking, there are always two solutions to solving where the pan and tilt axes should be. If you pan, tilt and roll through 180 degrees on each axis, the camera will be in the same position looking at the same target. Due to this, a solution needs to be chosen when it is determined that the tilt has flipped across the –90 threshold. The original handling works in many cases with occasional problems on Data Import. If you find that the camera is sort of in the right place, but looking in completely the wrong direction, then try "Flip Tilt" or "Flip Pan" and see if the problem is solved. Note that this can also occur on scaling moves.

Initialize Scaling: When you are scaling a move, the start position on the scaled move is often nothing like the start position of the master move. Due to this, the motor positions are not normally initialised to the start positions of the master move. When set to "On", then the scaled axis start positions will be the closest solution that can be found to the master move start positions. Note that if you are running with the rotate beyond +- 90 degrees, then the scaled move will put the rotate back to within +- 90 degrees unless you set this toggle.

Stop-Continue: "Yes/No/Prompt". Whenever the stop button on the mouse is pressed during a move, the computer assumes that you would like to continue the move from the point that you stopped it. Based on this it calculates a run from the stop point to the end of the move. If you do not want this to happen automatically, select this switch to "No" If you would like to be prompted as to whether or not to continue the move, then select "Prompt".

TC Display: "H:MM:SS:FF/MM:SS:FF/SS:FF" When the frame count is displayed as a timecode, the format of this data can be modified. Often, a move will not last Minutes, let alone Hours, so showing only Seconds and Frames could be desirable.

Smoother Stop: Stop handling is very important for a motion control system and there is considerable logic required to make the stops as seamless as possible since they can occur at any time during the operation of the motion control rig. This switch allows a different mode of stopping to be invoked which is particularly relevant to stopping the faster robotic motion control systems.

Input Data Debug: Flair has an advanced mode which allows for incoming data to control the position of the rig. This mode is quite complex and it is sometimes needed to print on going information to the console windows to assist with detecting issues with this mode of control.

5.12.2. Toggle Buttons

Comms Test: Displays data on the back screen to show if the main axis pipeline is communicating OK or if there are errors. When first switched on, there will be a number of errors displayed, but none should occur after that. A very occasional error is not a problem, but numerous errors indicate a wiring problem, axis board problem or noise/grounding problem.

HHB Comms Test: Data is printed out to show if the communications to the Hand Held Box are in error or running OK. Normally a few errors are not a problem as this is not a vital process, but a high number of errors indicates noise or a wiring problem.

Sync Data: When synching to an external camera, data as to the incoming sync pulse can be printed onto the back screen using this toggle. This is not really needed now as the External Camera Setup has a Test display which gives information as to the incoming sync pulse etc.

Timings Test: When ON, data is printed to the back screen giving an indication if the processor is handling the needed calculations in time or if it is running into problems.

Misc. Info: Additional information about the system and its operation can be printed out as the system is running by selecting this option. It would normally only be used in conjunction with MRMC to diagnose a situation.

Move Miniature: This is a very advanced feature and is covered in the end of the Target Tracking section of the manual. This toggle basically sets whether or not a scaled miniature should simply be moved to match the master action or whether the numbers entered in its positional waypoints designate an additional motion that it should perform on top of the scaled action.

Auto Run Short Gotos: Sometimes very minor differences in the internal math make the rig want to do a tiny goto when it is extremely close to the correct start point. This can be annoying and using this switch will mean the rig will automatically execute any very short (< 1/4 second) gotos.

Inverse Roll Handling: This is a specialised control for Data input and should only be used when so instructed by MRMC.

Cartesian Angle Fix: An advanced Target Tracking feature. See Target Tracking Modes on page 174.

Calculate Cartesians: When you switch to the Numeric Cartesian screen, the current axis names switch to Cartesian names, and the current positions above show the current values of those Cartesians. However, these values are ONLY valid when you are Target Tracking Cart Control. Using this switch means that the Cartesian Values are updated all the time. It may cause issues with the graphing, and thus this remains a switch to be turned off at will.

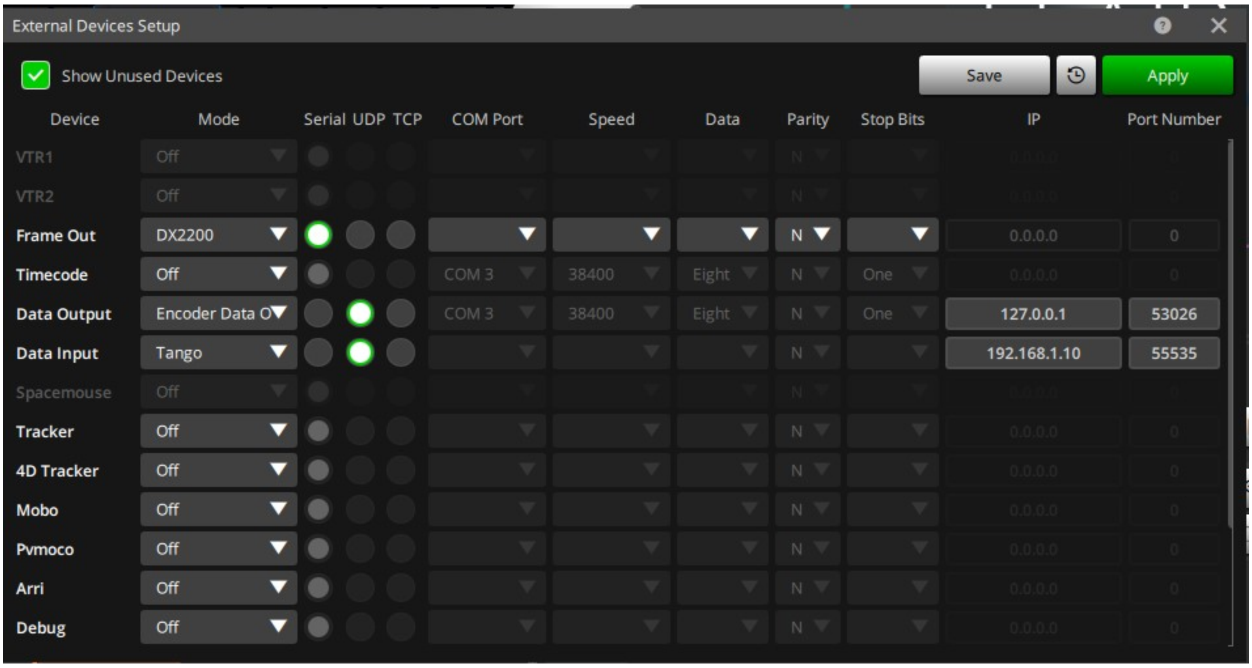
Limit Postrolls: If a post-roll is found to exceed the limit of travel of an axis, the post-roll can be re-calculated to stop faster to hopefully avoid collision with an end of travel limit. For example a move that slows down very forcefully near a limit could easily post-roll into that limit if stopped during the motion as the move deceleration could exceed the normal post deceleration. This should default to On, but can be turned off by the user.

Finer Limits Check: The standard limits checking samples the move 200 times over its duration, and this will usually detect all extremes of position and velocity. With long moves that have sudden changes in position, it can be that this testing is not fine enough to detect all extremes. In this case, you can switch to a finer test that tests the motion for every 50th of a second (System Ticks). This is a much more detailed test and is usually not needed and will slow the system down somewhat when doing graphing and preparing moves.

5.13. External Devices Setup

There are several serial devices that can be connected to Flair to add additional functionality to the system. There is one screen for all the serial devices. With the addition of Network interfaces and Network direct several devices can also be controlled via an Ethernet cable usually using UDP. Note that these devices are all communicated to via the Real Time portion of the program. When this is run under INtime, it does not have access to the standard windows devices. Only devices passed to INtime can be accessed in this case.

The standard Save, Apply, Revert and Help buttons function as expected.



☑ Show Unused Devices

Device	Mode	Serial	UDP	TCP	COM Port	Speed	Data	Parity	Stop Bits	IP	Port Number
VTR1	Off	○	○	○				N		0.0.0.0	0
VTR2	Off	○	○	○				N		0.0.0.0	0
Frame Out	DX2200	●	○	○				N		0.0.0.0	0
Timecode	Off	○	○	○	COM 3	38400	Eight	N	One	0.0.0.0	0
Data Output	Encoder Data O	○	●	○	COM 3	38400	Eight	N	One	127.0.0.1	53026
Data Input	Tango	○	●	○				N		192.168.1.10	55535
Spacemouse	Off	○	○	○				N		0.0.0.0	0
Tracker	Off	○	○	○				N		0.0.0.0	0
4D Tracker	Off	○	○	○				N		0.0.0.0	0
Mobo	Off	○	○	○				N		0.0.0.0	0
Pvmoco	Off	○	○	○				N		0.0.0.0	0
Arri	Off	○	○	○				N		0.0.0.0	0
Debug	Off	○	○	○				N		0.0.0.0	0

Available Devices

VTR 1 & 2: With a system that is configured to run the VTR, you have the option of setting the port and communications settings. Normally a VTR uses the Sony® protocol and this is 38,400 Baud, Odd Parity, 8 Data Bits and 1 Stop Bit. The Mode selection simply turns the VTR on or off. UDP/TCP not currently supported by any devices.

Frame Output: The current frame or time in seconds can be output from a serial port during the move or while browsing. This can be used to insert a frame count onto a video picture or control a record or playback medium. The first 2 options (Horita Frame and Horita Time) are for the Horita Serial Titler which inserts the data sent to it into a video signal, this allows easy matching of frame counts between playback and the move and is a very useful tool which can be used on most any motion control job. The ASCII option sends the data out as an ASCII stream and can be used in other applications one of which is slaving a Perception Video recorder to the system. (Though only at slow, less than 4FPS- speeds)

This functionality has been built into the RT12/14 hardware and you do not need the Horita device. In this case, use the “Frame Overlay Setup”. Note that the frame count is not absolutely frame accurate as it depends upon the update speed of the titler and also on the video speed. At higher than live action speeds, the video still only runs at 25/29.97 and you may find an increasing offset between the actual frame count and the number on the screen. Flicker free video tapes exacerbate this situation. However, used right this is an extremely useful tool. The Horita is only SD video, but the Microtronix DX200 can do frame overlay in HD. This device is currently only serial and is communicated to at 115,2000 Baud, 8,N,1.

Timecode: The time code reader option is for another Horita product which reads the Timecode from a standard SMPTE track and can run below speed and up to 3 times normal playback speed. When this option is enabled in the start-up config file, then a display will appear in the right hand control menu showing the currently read timecode and the desired starting timecode. For more information, see Chapter 16 Inputs, Outputs and DMX.

This option is provided by setting the TC Reader Mode Selector to “TC Comparator” If you select “Store Start TC”, then the system will record the timecode at which the move was triggered when you last ran it. You can use this mode to record the start TC, and then switch to the normal “TC Comparator” mode to play back in sync. Note that recording in sync with Timecode can be a complex subject and a fully study of it is recommended before committing to a shoot. “Store Start TC” will record the timecode at which the move is started by the user, and this can be used to sync the motion to the shot footage when the move is exported. “Output Timecode” is used to output the current time of the move via the RT12/14 hardware. This feature is not fully supported as of Version 6.2

Data Output: The options on this setting allow for a variety of different data to be output from Flair. There are a lot of different formats and uses, contact MRMC if you have specific questions. There is some data in the Appendix of this manual.

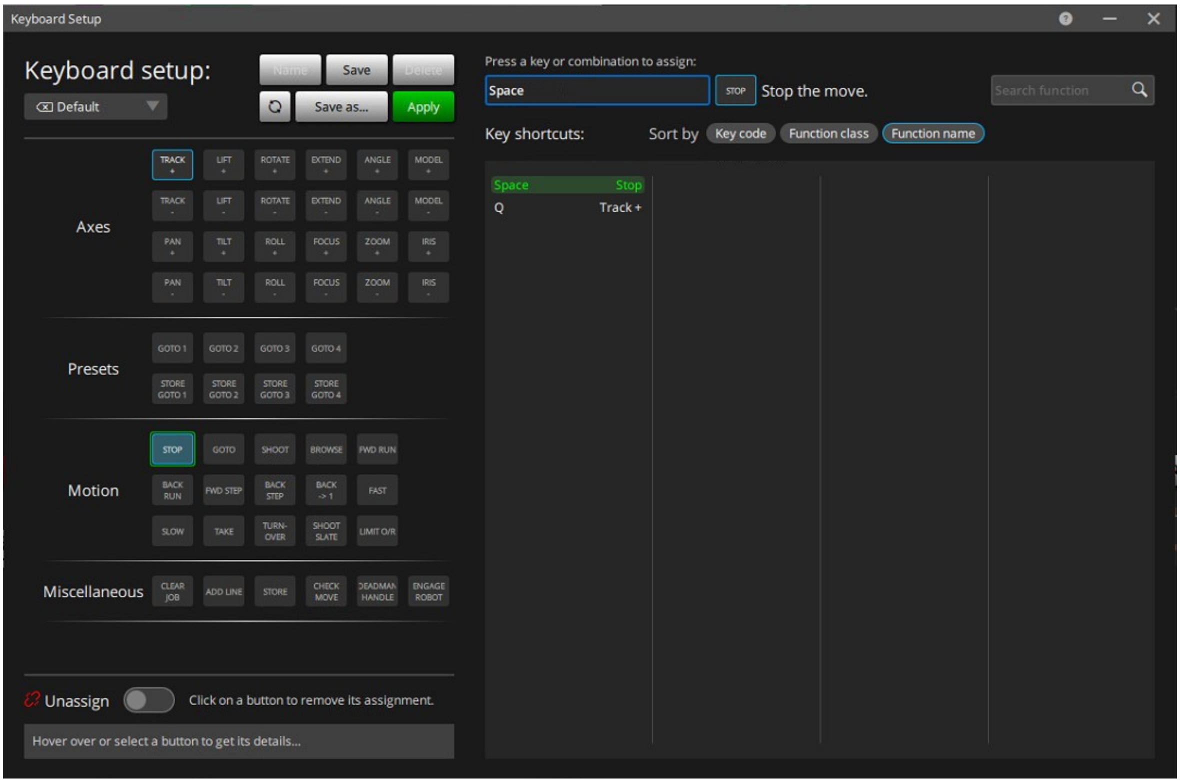
Data Input: Data input into Flair can be used to control the position of the rig from an external source. This feature should only be used in consultation with MRMC.

Other Options may be available depending on the application Flair is being used for.

Data Output 2&3: There are two other data output channels available and each can be to a different destination and can have different formats.

5.13.1. Hot Keys

There are many hot keys in Flair and some are fixed and some are configurable by the user. Generally speaking the ones involving the first 5 Functions Keys (F1 – F5) are fixed and so is Ctrl Z “Undo” Ctrl Y “Redo” and Ctrl S “Save”. To view or change the other hot keys select **Setups** → **Hot Keys** and you will get the following dialog box.



Keyboard Setup

Keyboard setup: Default

Press a key or combination to assign: Space

STOP Stop the move.

Key shortcuts: Sort by Key code / Function class / Function name

Space	Stop
Q	Track +

Unassign: Click on a button to remove its assignment.

Hover over or select a button to get its details...

The functions to be executed are listed on the left and the key combinations to trigger it are on the right. You can specify any Alphabetic key or any of the Function Keys F1 – F12 together with a modified key one or more of “Shift”, “Alt” and “Ctrl”. Note that “Alt” on its own is used by windows to bring up the main menu, so you would normally use “Alt” and “Shift or “Alt” and “Ctrl”.

The key that is typed is simply select the function to be assigned on the left and them press the key you want to assign and then press Enter. To unassign a key, simply turn the **Unassign** option **On** and then click the keyboard shortcut. As mentioned above, the first several are not editable and are there so you can see what the designations are. If two functions used the same key combination, then the first listed ONLY will be called.

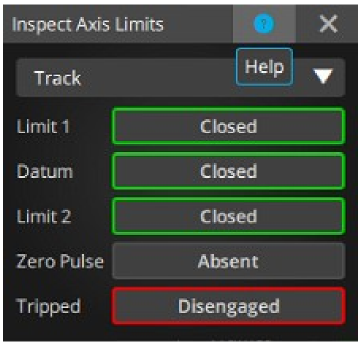
More Hot Keys will be added as the need arises – feel free to contact MRMC if you want to add a function to the Hot Key List.

Saving the Hot Keys will generate an xml file in the install directory of any keys that are changed from the default.

5.14. Limits and Safety

The **Status** menu in the menu bar provides various option that you can use for checking axis limits and safety information.

5.14.1. Limits Test



Inspect Axis Limits

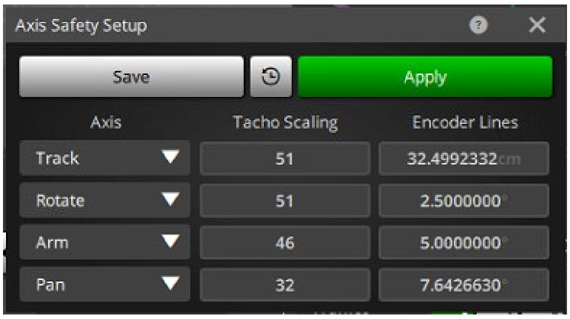
Track

Limit 1	Closed
Datum	Closed
Limit 2	Closed
Zero Pulse	Absent
Tripped	Disengaged

Each axis has three limits and can sense the state of the zero marker pulse on the encoder. It is occasionally useful to check these to see what state they are in. When the Inspect Axis Limits dialog box appears, select the axis you want to test. When this is done, a line appears stating the state of the limits for that axis. The first 3 numbers are 0 or 1 and indicate the states of End Limit 1, Datum Limit 2 and End Limit 3 respectively. The number after “ZM” indicates if a zero marker pulse has been detected, and is set to 1 when there is one and 0 otherwise. The zero marker pulse is so fast you are unlikely to see this number change; its main use is to check the sense of the zero marker pulse – It should read as a “0” 99.9% of the time. The last number after “Engaged” indicated whether the axis has been engaged by the user; 0 indicates the axis is off and 1 indicates an engaged axis.

5.14.2. Axis Safety

The Axis Safety dialog box is only applicable for users who have the Mark Roberts Motion Control Axis safety card installed on their system. This card compares the speed sense from an axis with the intended speed of the axis on a continual basis so that if the motor turns when it should not do so, then the axis is tripped out by the safety system. This feature may be retrofitted to early systems; contact MRMC for details.



Axis Safety Setup

Axis	Tacho Scaling	Encoder Lines
Track	51	32.4992332cm
Rotate	51	2.5000000°
Arm	46	5.0000000°
Pan	32	7.6426630°

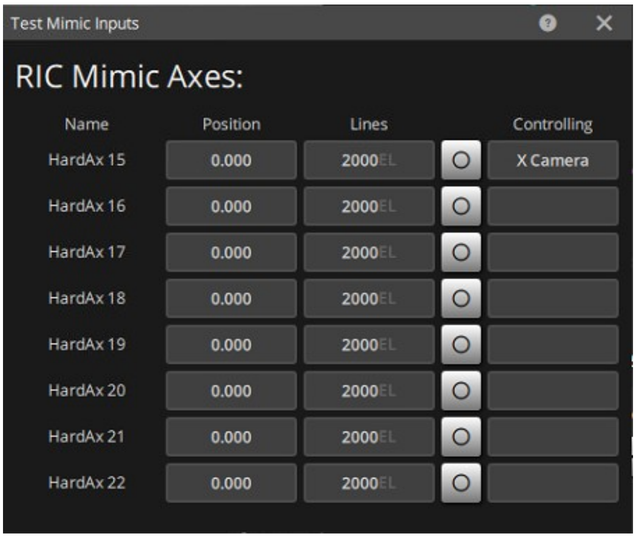
Axis: The axis to be monitored. “Not Installed” if no axis is attached to that safety port. Usually which axis has been connected will be designated by MRMC.

Tacho Scaling: This is the voltage from the tacho per 1000 rpm of the motor. If the tacho is directly connected to the motor, then this figure is just the voltage constant of the tacho, if the tacho is geared, then the voltage constant has to be appropriately scaled.

Encoder Lines: The number of lines on the encoder of the axis being monitored.

If an axis that has the tacho monitor on it is detected to be moving by the program despite having been tripped, then in versions of the hardware where the e-stop can be controlled by the software, the software will drop the e-stop. This is no way relieves the operator of his responsibility of ensuring the system is safe at all times.

5.14.3. Test Mimic Inputs



Test Mimic Inputs

RIC Mimic Axes:

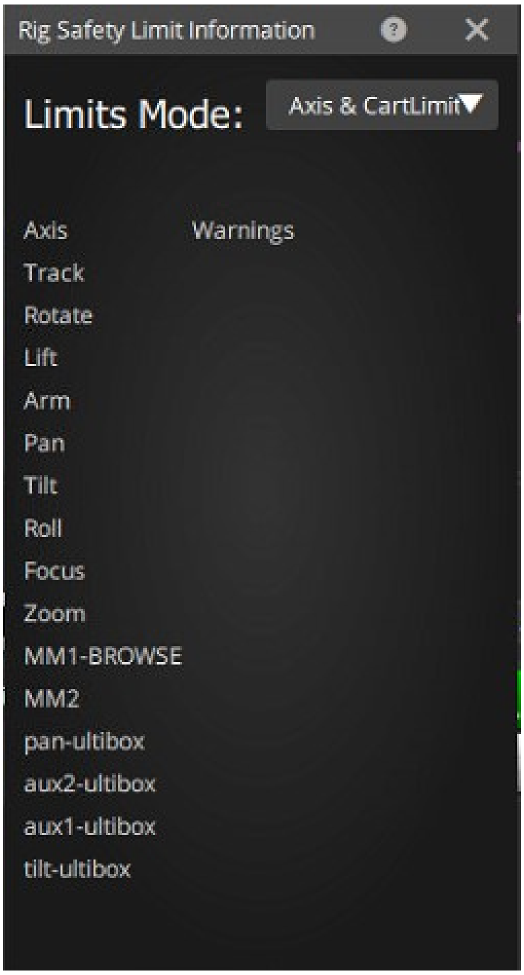
Name	Position	Lines		Controlling
HardAx 15	0.000	2000EL	○	X Camera
HardAx 16	0.000	2000EL	○	
HardAx 17	0.000	2000EL	○	
HardAx 18	0.000	2000EL	○	
HardAx 19	0.000	2000EL	○	
HardAx 20	0.000	2000EL	○	
HardAx 21	0.000	2000EL	○	
HardAx 22	0.000	2000EL	○	

This screen shows the active mimic inputs and the value that is being read from this input port. It show if an axis is assigned to this port. You can also zero the through this screen, but this is not recommended on a live axis.

5.14.4. Rig Limits Info

When the **Cart Ctrl Limits?** state in Miscellaneous Setup is set to **Diagnostic Mode**, the operator is informed when axes are on or near their physical limits, when axes would have excessive velocity or acceleration if the software was not intercepting such, or when a Cartesian control is at its limit as defined in Carts Setup.

In the example below, the tilt axis is near to its physical limit and the rotate axis has reached its physical limit.



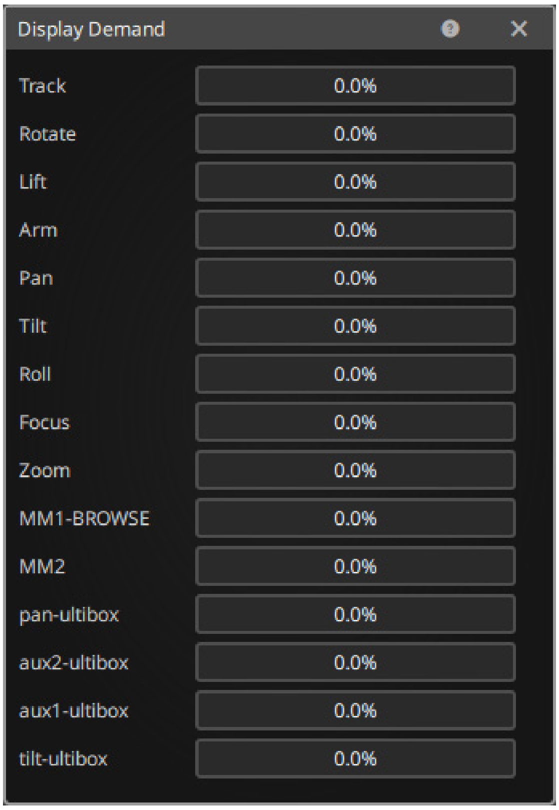
Rig Safety Limit Information

Limits Mode: Axis & CartLimit

Axis	Warnings
Track	
Rotate	
Lift	
Arm	
Pan	
Tilt	
Roll	
Focus	
Zoom	
MM1-BROWSE	
MM2	
pan-ultibox	
aux2-ultibox	
aux1-ultibox	
tilt-ultibox	

5.14.5. Test HHB Demands:

It is useful to see the demands that are incoming on the Axes and the Carts form the various methods of moving things. Sometime a demand is present when not expected or a demand is not being registers. You can use this display to see if this is occurring or even if a demand is not enabling full speed as the value should be able to go from -1 to +1. You cannot currently drag these sliders to drive motion in the axes.



Display Demand

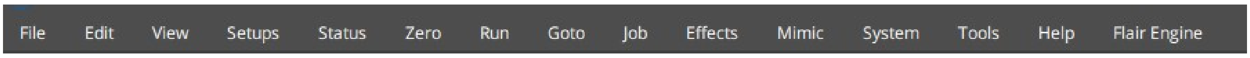
Track	0.0%
Rotate	0.0%
Lift	0.0%
Arm	0.0%
Pan	0.0%
Tilt	0.0%
Roll	0.0%
Focus	0.0%
Zoom	0.0%
MM1-BROWSE	0.0%
MM2	0.0%
pan-ultibox	0.0%
aux2-ultibox	0.0%
aux1-ultibox	0.0%
tilt-ultibox	0.0%

Chapter 6 **Main Display**

When the program first loads, you will be presented with Main Display which is composed of Top Menu Bar, Numeric Display, Side Menu Bar on the right and Control Box at the bottom. Each part of this display is described here in turn.

6.1. Top Menu Bar

The Top Menu Bar is composed of a row of buttons across the top of the screen, these buttons allow you to access the various features of the program and are always visible to the user.



6.1.1. File

New deletes all existing waypoints and any data associated with them. Before actually deleting all the data, you are prompted to confirm this action. There are several selections for clearing data.

Reset All deletes the move as above, but also re-reads the configurations of the axes and the system from the default values stored on disk.

Other options in the File menu include: “Open Job”, “Save”, “Save As”, “Job Storage”“Recent Jobs”, “Load Simple Job”, “Reload Current Jobs” and “Recent Job Files”.

Job Storage is a means of saving a job that one has run so that it can be run at a later date simply by retrieving it off disk and running it again. Further details is covered in Chapter 9 Job Storage.

Move Import, Move Export, Recent Imported Files

These will produce dialog boxes to allow the user to import or export data between Flair and CGI systems. See Chapter 17 Data Import and Export for more details.

Text Configuration Files

It is possible to have Text configuration files to automate making minor changes in the system. These files can be viewed and edited through the software or with any standard text editor (such as Notepad or WordPad) but should not edited with an advanced word processor which might add additional unsupported characters (Only ASCII is allowed). The default file extension for such files is “.asc”, but this is not compulsory.

A typical use for this could be changing the length of the arm at Zero if you have different length arms and simultaneously reducing the Rotate Maximum acceleration as this is sometime compromised with a longer arm. It is irritating to have to change multiple settings each time you switch out arms and setting up a simple text config file for the short arm and the long arm makes this much easier. You could also have different operators who like faster or slower settings on the hand held controls etc. Almost every setting on the axes, camera and kinematics can be changed through these files. You can also read multiple files in, so for example, you could read in “ShortArm.asc”, then “Peter.asc” and finally “DigitalCamera.asc” to set you rig up for a specific shot.

The basic format of the file is to have a title that indicates what type of configuration you are changing; this can be an Axis name, “Kinematics”, “Internal_Camera”, “External_Camera” or “Miscellaneous”. In the lines that follow the title, the user sets variables in much the same way that they are set in the resource file “.ini” which is described in Chapter 2 Installation. There is also an addendum in that chapter which describes all the possible variables that can be set.

Config Data Recent Config Files

Import Text Config and **View Text Config:** will bring up a file selection dialog box which will open in the “TextConfig” directory if that has been set through the “.ini” file or otherwise it should only in the default sub directory “configs”. It will display any file with the “.asc” extension by default. Selecting “View” will bring up “Notepad” on your system if it is there and will allow you to look at and edit the file. Selecting **Import** will attempt to import the file data into your system. On any significant error, the import will abort; if an invalid data string is detected, it will be ignored and the system will continue to import valid data in the file. Errors should be reported on screen and also in the Output Log file which can be viewed through the “Help” pull down menu.

> Note
>
> There is NO error checking on the data import other than correct format. You can set values in the system to illogical values that can cause anomalies, so use this feature with caution.

6.1.2. Edit

This menu gives you access to different UNDO options and settings. The Undo feature of the Flair software operates by storing the move state in memory when the user changes the numeric display or the positions in the graphic display. Beware that not every action is undoable and you are STRONGLY RECOMMENDED to save your data before trying to undo any action. Typically dialog box changes are not undoable. You have the option to disable or enable the UNDO feature, as well as specify the size of the undo buffer in memory. The default number of successive actions you can undo is 15. If you right-click in an unused part of the numeric screen or Shift right-click in a populated section, the context menu will appear. This lists the undoable actions, and you can select any of the listed action to be undone. This is basically the same as selecting undo several times.

6.1.3. View

The options in the View menu allow you to open the respective views in separate windows in addition to the ones already displayed on the main screen.

Reset view: Resets the view to default view.

Rig Control View: Displays the Edit view as a separate window. In the default view, the Edit view is on the top-left.

Timeline Control View: Provides additional options to control the move on the rig. In the default view, it is displayed on the right.

Waypoint View allows to add moves with waypoints containing positions for all the axes involved in the move. For more detail, refer to Waypoint View on page 123.

Graph View allows you to view the move graphically once it has been entered. Further information is contained in the “Graphic Move Representation” section of this manual.

Mimic controls allow you to set up mimic controls when controlling axes using a mimic device. See Chapter 12 Mimic Mode for more details.

3D Rig View shows the 3D model of the rig. It allows you to RUN and BROWSE through the move without moving the rig. This is very handy since you can have an idea of how the move will look like when it is executed in the real world. You can also look through the camera and see if the rig or track is in shot. This is described in detail in 3D View on page 137.

3D Targets View shows the target motion in 2 and 3D as opposed to a position against time. Like many CGI packages, there as Font Side Top and a perspective view of the target motion if applicable. See Chapter 10 Target Tracking for more details.

Quick Moves Adjustment allows making adjustments to an existing move.

Live View shows the target from the camera if one is connected. When the camera is not connected, this view shows the actual position of the camera and target or the camera and its orientation. Use the controls in the view to move the camera in Cartesian. See Chapter 10 Target Tracking for more details.

Axes View shows the current position for each axis along with the sliders to move the axis position.

Cart Axis View shows the current position for each cart axis along with the sliders to move the axis position.

Spin Rig View see Chapter 7 Alternate Displays

Camera Control View provides a user friendly view for controlling the camera position in Cartesian mode and individual axis mode.

Timeline View shows the current position of the system in relation to the move. Other controls help the user to run and record the move.

6.1.4. Setups

This option allows you to access the setups for the system. These are all fully described in Chapter 5 Setups on page 19.

6.1.5. Zero

Zeroing axes is the way the system knows exactly where each motor is so that it can stop it from exceeding its limits of travel, and so that target tracking can work properly. Zeroing basically involves registering the position of an axis against a known reference point. Full details are covered in the "Axis Setup" and "Zeroing" chapters of this manual.

6.1.6. Run

Allows access to options that allow you to run a job in different ways:

Browse

This puts the current move into a Browse mode allowing you to run the rig through the move at a user selectable speed. It uses the current camera speed and the current HHB speed percentage. It is recommended to try this slowly until you are confident with the camera motion. Moves with no fairings will tend to come to a fairly abrupt stop at the beginning and end of the motion.

Forward Run

Forward Run runs the job from the beginning to the end. **Back Run** runs the job from the end, back to the beginning.

Part Run

This allows you to run the job from any frame count within the move to any other frame count within the move.

Timecode Run

This is similar, but instead of entering frame counts, you define the start and end times by their relative timecode. This is only functional when you have a start time code selected in the Inputs Setup or on the start TC entry box at the bottom of the right hand screen menu. In this case, the time codes you type in will define how far into the move you want it to start and stop. These time codes have to be within the move, or the system will run as much as it can of the time code period you have entered.

Repeat Run

This allows you to enter how many times you would like the job to be run, and then runs the job forwards and backwards that many times - usually used for demonstration or test purposes. Entering **0** will cause the system to run the move continuously until stopped by the user.

Sync with Other Flairs

Api Master

The control of Flair via the API has been available for quite a while and has allowed outside companies to control Flair and write custom interfaces. There is even an application that allows you to control multiple Flairs from one program. his program has been merged into Flair so that from one Flair you can control that Flair itself and up to 4 other Flair systems. The Controlling Flair could be referred to as the Master or Host and the controlled Flairs would normally be on other machines and called Slaves or Clients.

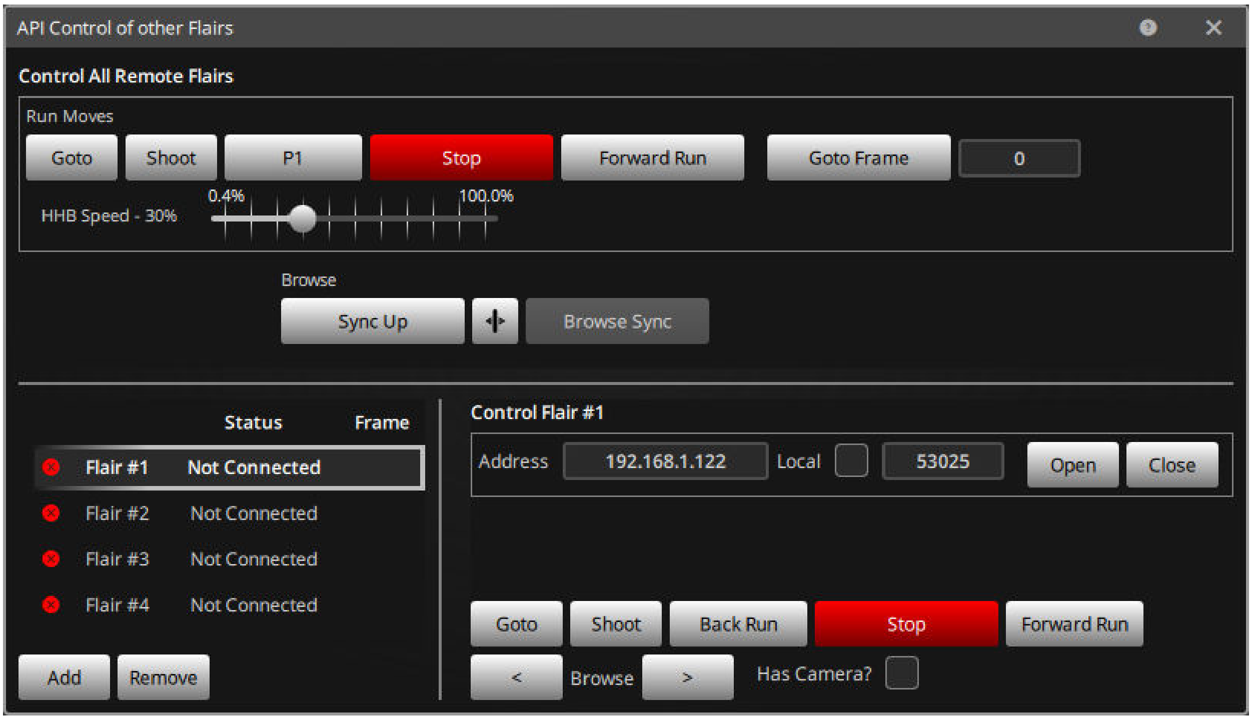
When using the API Control, you would control the motion of the Host Flair through this interface and the other Flairs can be controlled at the same time. Normally this would not be set up until all the moves are ready to be operated. Running the "Host" Flair directly through its control would not currently affect the controlled Flairs.

Connection

You can define how to connect to the controlled Flairs. You can state the IP Address of that machine and the port that the Flair on that Machine is listening to. Of course, you have to enable API Control on any of the controlled Flairs first. The Local checkbox means that the controlled Flair is on this machine which is usually used for testing since you cannot run more than one complex real time Flair on a machine at one time. This way you can add up to 8 Flairs, using the "Add" button. Use "Remove" to remove the Flair.

You can also set defaults for the connected machines in your Flair.ini file. See later for more details.

Once connected, you can use the "Run Moves" section and "Browse" to control the host Flair and the remote Flairs.



API Control of other Flairs

Control All Remote Flairs

Run Moves

Goto Frame: 0

HHB Speed - 30% (0.4% – 100.0%)

	Status	Frame
Flair #1	Not Connected	
Flair #2	Not Connected	
Flair #3	Not Connected	
Flair #4	Not Connected	

Control Flair #1

Address: 192.168.1.122 Local □ 53025

Has Camera? □

There are the standard move control buttons for each Flair for "Forward Run",:Back Run", "Goto", "Shoot" and Stop" as well as a "Forward Run All", "Goto All" and Shoot All" which would be any connected Flair as well as the Host machine, You also have the ability to make all the Flair Go To a specific frame. Obviously, each of the Flairs would need to have a valid job set up and ready to be run. Stop All will stop any ongoing motion in all the Flairs and would also take any of them out of a run state. "Back→1 All" will run all the rigs back to number one assuming that the Flair is at the end of its move.

You can Browse the remote Flair and also browse them together. The Sync mode is not fully working, but the rigs will stay pretty much in time if you browse all at the same time. The Synced Browsing is a little more complex and it will be announced when this is working.

Not all of the API features are fully operational in earlier versions of Flair, it is advised that you run only a Release build of Version 6.40 and above.

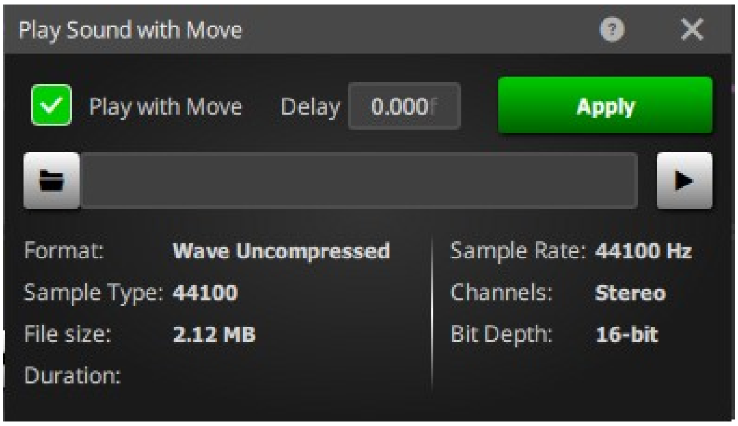
Connection Resources:

You can enter the Address, Name and Port of any of the connected machines using the format of the following lines in your Flair.ini file. The digit at the end of the resource indicates which remote machine you are entering data for.

```
*ApiServerAddress1:192.168.1.10
*ApiServerName2:SJWTOWER
*ApiServerPort3:53025
```

Play Sound File

You can select for Flair to play a windows sound file in sync with the motion of the camera when you run the move in a forward direction. Selecting this pulls up a dialog that details the file you wish to run and other data. This feature is not precisely timed to the move but will generally be within a few frames and can be used for cueing actors etc, but should not be used for lip sync purposes, use the time code trigger interface for that. Once the file has been selected, you can test play it or set it to play with the move and optionally add a delay to get the timing correct. Due to possible timing delays, Part Runs of the move may not sync exactly the same as the full run and this feature should not be relied upon for precise timing, it is intended as a helper, not an exact sync tool.



Play Sound with Move

☑ Play with Move Delay 0.000f

Format: Wave Uncompressed
Sample Type: 44100
File size: 2.12 MB
Duration:

Sample Rate: 44100 Hz
Channels: Stereo
Bit Depth: 16-bit

Open file allows you to select the file to play, Play brings it into memory ready to play. Play lets you test the file, Info will tell you how long it is and Stop obviously stops the music, but this will also occur when the move stops or is stopped. This feature only works at normal speed and will not work faster or slower if you run the camera faster or slower. Again use an off-speed playback system with timecode for that. Not all .WAV format files work with this system. Test the file by using the Info button to see if the file has been correctly interpreted.

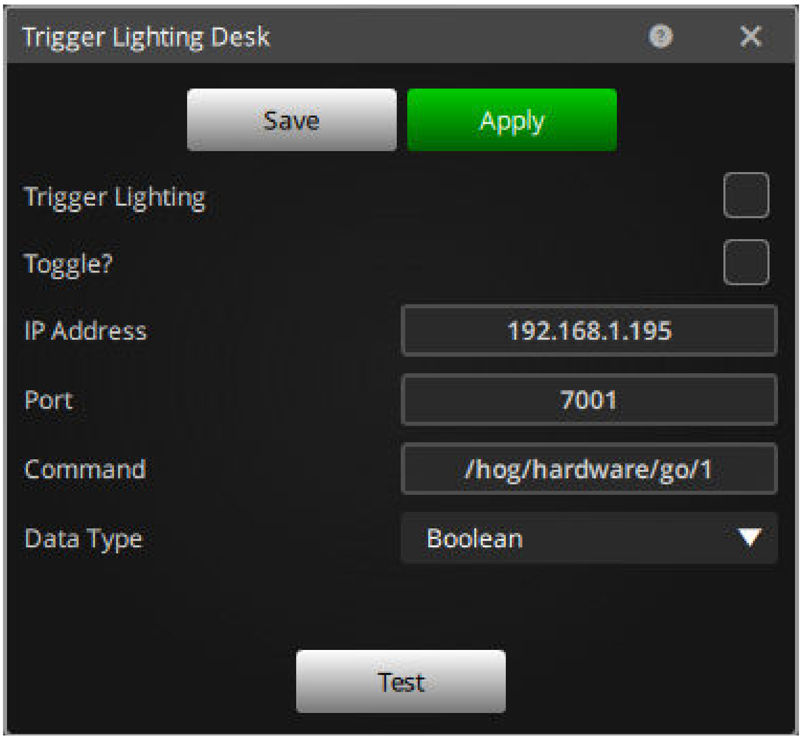
As with the displays, also listed on the pull down pane are the hot keys which can be used to access these functions from the keyboard. Usually, F2 can be used for Forward Run, F3 for Back run and F4 for Part Run.

Lighting Desk Trigger OSC control:

Most lighting Desks and in fact many electronic systems now support the “Open Sound Control” protocol which is an easy way to communicate between systems. Lighting desks for example will often respond to a command to “Play” “1” which means to execute the playlist number 1. This data is easily encoded in Flair and decoded at the lighting desk. You may need to get some technical information on the exact commands expected and supported, but this data is often available on-line. The standard basic packet consists of a string followed by a value which can be True or False, or an integral value, or a Floating Point value. Each of these can be chosen in the Lighting Desk trigger setup and when combined with the right command will cause different actions to occur in the lighting desk, the primary one of course is simply to start a program running. Some specific commands are detailed in the Help Menu and there is also a “Test” trigger to send the command to the lighting desk to see if it responds. Otherwise, the command is sent when the move starts. This feature will NOT work for part runs and you will need to ensure adjustments are made in that case.

A “Toggle” value can be used to send a “True” followed by a “False” which can be used to press and release a Key. If toggle is set for a numeric value, then 0 will be sent after the value is sent.

As covered in the Help Dialog, you will need to make sure that the Lighting Desk is on a compatible network with the real-time network of Flair which is the one usually connected to the Real Time interface unit (RT12/14/15). You will also need to ensure that the address for the board and the port on which it is listening are correct in order to make this work.



Trigger Lighting □

Toggle? □

IP Address 192.168.1.195

Port 7001

Command /hog/hardware/go/1

Data Type Boolean ▼

BlackMagic™ Hyperdeck interface

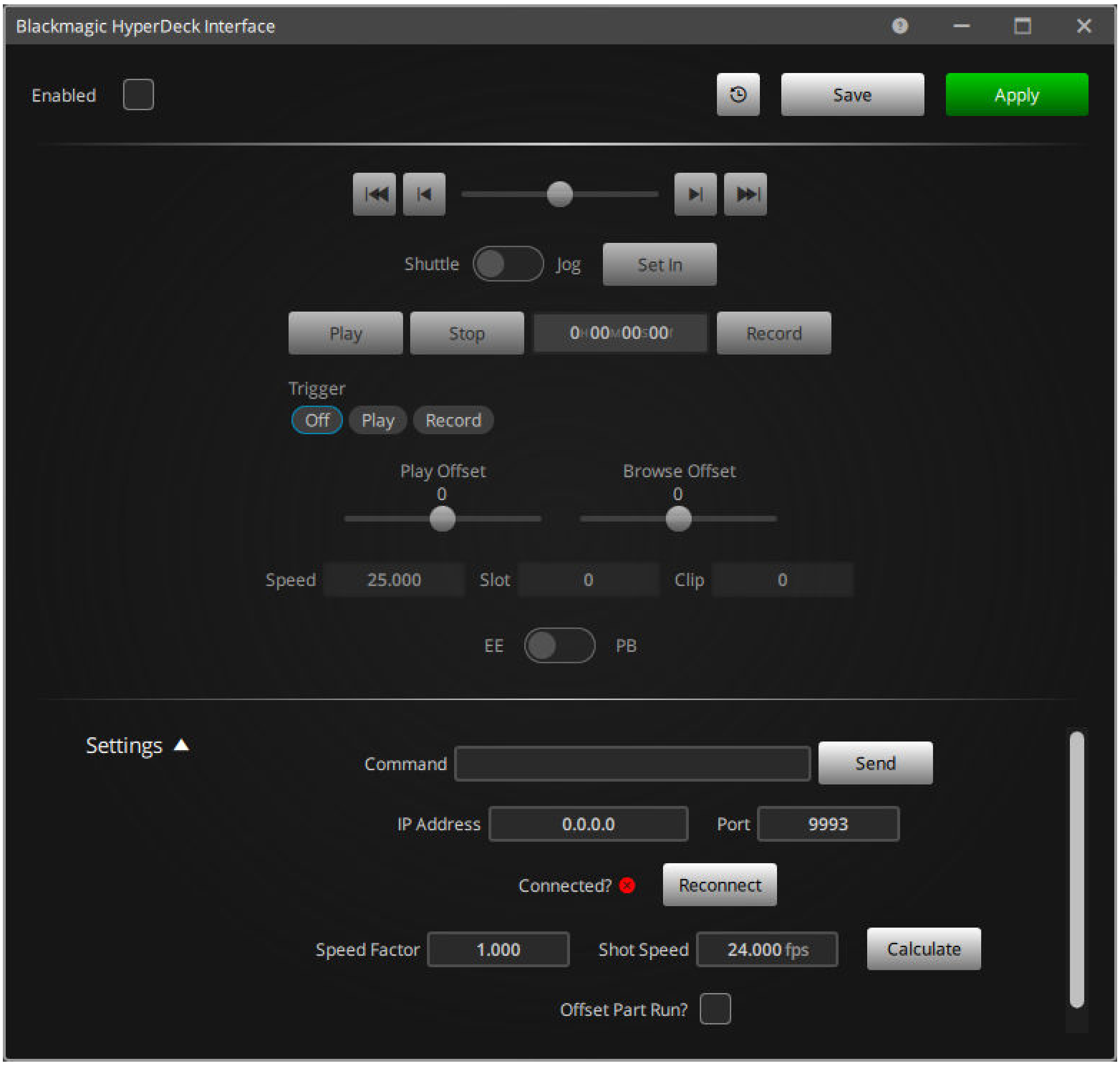
BlackMagic make an extensive line of video equipment including a line of solid-state video recorders that have an Ethernet interface. These respond to commands to play, record and shuttle video clips. An interface has been added to Flair to allow you to cause the BlackMagic to work in conjunction with Flair so that it will Record a clip and stop at the end of the move, Play it in sync with Flair and also can be made to scrub through the Clip when you browse the move in Flair. The option “Hyperdeck Control” will appear under the “Run Job” menu if you add:

```
*Hyperdeck:True
```

To your Flair.ini file. You will need to have a configured Network Adapter assigned to Flair, and this can be the same as the one used for Network Direct, of course you will need a Network Switch to add the Hyperdeck to the network if you do not already have one. It will work with an adapter assigned to Windows™ or assigned for use under Intime™. This interface has been tested with the Hyperdeck Studio and Hyperdeck Studio Mini.

This feature is accessed from the Topmenu “Run Job” pulldown “HyperDeck Control”

The Interface dialog provides the normal VTR style controls to Shuttle and switch Clips.



0H00M00S00F

Play Offset 0

Browse Offset 0

Speed 25.000 Slot 0 Clip 0

Command

IP Address 0.0.0.0 Port 9993

Connected? ☒

Speed Factor 1.000 Shot Speed 24.000 fps

|<< is Go To Previous Clip

|< is Go To Start of Current clip

The Slider Bar allows you to scrub the current clip forwards and backwards

>| is Go To End of Current Clip

>>| is Go To Next Clip

EE is a toggle between EE = Electronics to Electronics meaning show the current input and PB = Play Back which means show the current clip. This will sometimes change without updating the button and you may need to press it more than once to get the deck to change.

Play starts the deck playing.

Stop is self-explanatory.

Record will start the deck recording.

There is a Timecode display box which is output only and shows the timecode of the current clip.

Shuttle is a toggle between Shuttle and Jog which controls the speed that the slider moves the clip. Shuttle is very fast, and Jog is much slower.

Set In: This sets the start point of the Clip. Move the clip to where you want it to play back from and click this button. The current timecode of that clip will be set at the Cue Point.

Setup: Brings up a dialog to do with the connection to the BlackMagic HyperDeck.

Trigger: tells the system what to do with the deck when a move starts. If it is Off, nothing happens, if it is set to Play, then a play command is sent to the deck when the move starts. When Record is selected, then a record command is sent at the start of the move.

Play will start playing from wherever the desk is currently cued to. However, at the end of the move, the Clip will automatically be sent to the current cue point which is set using the Set In as stated above. The current move speed, and the move speed at which the clips was recorded have to match. Play also means that the clip will be shuttled in sync with the current frame count of the move if the rig is browsed. This means that as you browse the rig forwards and backwards, it will be browsed through the clip so that the footage of the clip should match the position of the camera. This is of course dependent on the clip having been shot at the move speed.

The Offset values are adjustments between the playback footage and the move. Usually, a value of 0 is pretty close, but depending on the video system you may need to make advance or retard the playback clip against live. A positive value will delay the start of the clip by 1/50s of a second and a negative value will cue into the clip later to advance the clip. The value is in Flair internal system time 1/50s of a second.

It has been found that when browsing the clip the offset is a little different. You can adjust this value while browsing and in fact this is the easiest way to do it.

Video Speed, Slot and Clip are output information from the BlackMagic to do with the currently selected Clip.

The “Setup” button will bring up the settings dialog which allows you to configure the interface to the Deck. You will have to go into the Setup on the deck to enable and configure the IP address etc. and the setting in this screen need to match the settings in the deck for it to work. The IP address will have to be within the subnet that you are using on your adapter. The “Connected” check box tells you if you are connected and if you have issues, you can “Reconnect” to establish the communication connection to the BlackMagic HyperDeck.

I know I said above that the clip MUST have been shot at the same speed as the rig’s move in order for the shot to play in sync, but you can actually adjust the speed using this dialog, but it is up to you to verify the accuracy of the calculations. If you know the speed factor difference from the clip, you can enter it into “Speed Factor”. For example, if the clip was recorded with the rig moving at ½ speed and you want to play it back with a speed factor of 2.0 so it plays twice as fast so runs for the same duration as the quicker move. The “Calculate” button is currently non-functional. The Browse offset adjusted the timing between the clip and the move when you are browsing the move, the clip browses with you. This offset lets you adjust the timing so that the clip and move together. The units are in video frames.

6.1.7. Goto

A “Goto” is when the user wishes to move the axes to a specific position directly, without worrying about the path of the move or any other factors. Using “Goto”, the user can move the system to any specified frame count or position number simply by typing in the desired frame count or position number into a small dialog box, accepting the input and then pressing the **Goto** button when the system is “Ready To Goto”. Note that the **Goto Waypoint** will drive the rig to the listed position in that waypoint number. This may not be the position of the rig at that frame of the move if advanced editing or effects are being used. **Goto Frame** should always allow you to drive the rig to the position of the rig at that specific frame count.

Goto Closest When you are using target tracking it is possible to scan through the move to locate the frame count that is closest to the current camera position. This can be very useful for getting the camera back onto the move path when it has been moved away for some reason, or the timing of the move has changed and the current frame is not necessarily the closest to the current camera position. Can also be brought up by using Ctrl + L. When chosen, it will tell you the closest frame and prepare to go to that frame count.

Goto Limits:

The **Home** and **Select to Home** options allow the user to return the entire system or selected axes to their correct home positions; this is often used at the end of the day to position the rig before shutting down.

Cartesian Positions

There are 3 menu items here and the Single Cartesian will give you a drop-down menu of all the adjustable Cartesians. The idea here is to be able to move one or more Cartesians to specific positions. This could be done by moving on the HHB in cart control, but this provides an easy way to drive to the rig to exact positions. It is an advanced feature and should only be used by users who fully understand the various Target Tracking modes as otherwise it can send the rig to an unexpected position.

The Simplest form is Goto Single Cartesian, and a dialog will come up with the Cartesian name and an entry box.

A value can be typed in (n this case the Z Cartesian of the Camera) and when Goto or Browse to is used the camera will move so that the nodal point is at the height entered, and the other cartesian values stay the same. Exactly what the camera does it somewhat affected by the cartesian mode and so it might do slightly different things than expected. For example, in Normal Target Tracking, you are not moving the Target, so as the Camera moves up, it will tilt down to maintain the same target. If you were in Pan Tilt Target Tracking, then the Tilt of the camera is a Cartesian, so it would NOT tilt down, but when stay at the same camera tilt as the arm moved up. Other more complex relationships occur between the modes.

The “Goto A Cartesian” provides the most likely Cartesians to be adjusted on a single display with a separate “Goto/Browse To” button for each.

Note that the Cam Distance is the NODAL distance from the Camera viewpoint to the Target and will not match the entered Measured Target distance, but will usually be close.

Here you could, for example, enter a value of 15 into the Tilt, and then when Goto or Browse to is pressed, the camera should go to a tilt angle of 15 degrees to the horizontal. As mentioned, these are advanced features and should be used by experienced operators as you could easily send the rig into the set or the ground if used without full understanding.

The most advanced mode lists ALL the Cartesians, and several of them will not be adjustable, but are provided to allow very advanced camera manipulation.

The main difference here is the ability to select how the Cartesian is interpreted by using the different modes to "Goto To" that cartesian position. For example you can enter an elevation and if you use "Goto Orbit" it will lift or raise the camera to get to that orientation from the Target. If you used "Goto PTR", then the camera's position is NOT defined by an orientation to the target, so elevation of 15 is interpreted as the Camera is 15 degrees up to the target, so the Camera tilts up to 15 degrees. (As I said, it is complex)

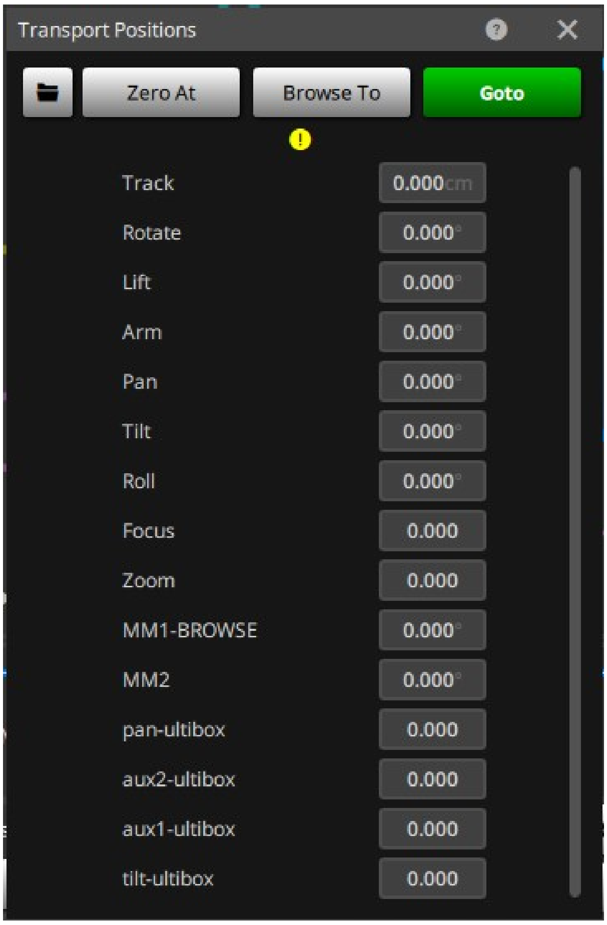
Right clicking in any of the entry boxes will store the current cartesian value into the entry box so you can see exactly where that value is before you adjust it, and even use the maths functions to move it by certain amounts. Units are Metres for the Cartesian Positions, degrees for Angles and the Camera Distance is in measure units.

Goto Information will display where each axis is trying to go to when the system is at **Ready to Goto** This is useful as it also tells you how long each axis needs. If you get an unexpected Goto or a longer than expected Goto, this can be used to find out why before sending the axes off to an unknown position. The display contains buttons to execute the Goto or set the system into Browse Goto mode. You can also access this display from the context menu by right clicking in a blank area of the Numeric display (one with no line showing).

Stored Positions: There are four banks you can choose from: "Goto 1" to "Goto 4". These are positions of the rig which the user can store and then cause the rig to go back to at some point. The primary use of these is for positions to check zeros or for a camera loading position. These positions are saved with a job and may optionally be restored when a job is read in (See Job Storage Settings on page 162) they are reset when the program is reloaded.

Preroll Extent and **Postroll Extent**: If a move or section of a move starts or stops with the rig moving, then the rig will have to run on after the move, or back up to get up to speed before the move starts. The rig cannot instantly start or stop. The distance that is required to "Preroll" or "Postroll" is dependent on a number of factors, but primarily upon speed and the acceleration of the major axes involved. It is very important to know that the rig is not going to run into someone or something while it is prerolling or postrolling, and this “Goto” is to allow you to view the numerical values of the extend of the Preroll and Postroll, and also to move the rig to those positions to check that it is not going to hit anything. Use of this function does NOT absolve the operator or his responsibility for ensuring that the rig is operated in a safe manner, and any Preroll or Postroll should be tested at gradually faster and faster speeds to ensure it is okay. Note that doubling the speed of a move will usually quadruple the Preroll/Postroll distance. Note that as of Version 6.2 this feature does not accurately drive the roll axis when in Roll Level or Roll Relative and may trip the roll axis out when it gets to the end of the Goto. It is intended that this will be fixed in an upcoming release.

Transport Positions Each axis has a user configurable transport position, and when you select this option, all axes will go to their entered transport position (unless it is left blank in the axis setup). This position will often be the “Home” or “Zero” position for the axis, but some axes might have to be moved to tuck the arm in tightly or tilt the camera platform down for the most convenient transportation position.



Track	0.000cm
Rotate	0.000°
Lift	0.000°
Arm	0.000°
Pan	0.000°
Tilt	0.000°
Roll	0.000°
Focus	0.000
Zoom	0.000
MM1-BROWSE	0.000°
MM2	0.000°
pan-ultibox	0.000
aux2-ultibox	0.000
aux1-ultibox	0.000
tilt-ultibox	0.000

6.1.8. Job Type

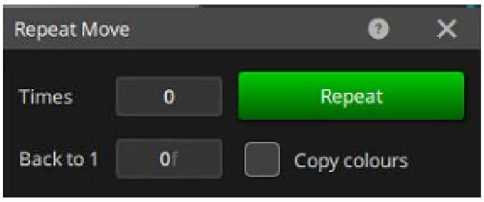
Allows the user to change the behaviour of the system in various ways ranging from the type of curves generated to how the focus axis is controlled. Refer to Job Type Configuration on page 163.

6.1.9. Effects

Contains all the move effects that Flair allows you to apply on your current move these controls are:

Flip Pan Arm: With most heads, the tilt arm is on the side of the camera, and this occasionally blocks a light or restricts the motion. The same camera position can be achieved by flipping the pan, tilt and roll through 180 degrees. This function will flip the axes but leave the move the same. It is only fully operational in target tracking moves.

Repeat: Allows you to simply repeat the same programmed move again and again. For example if you repeat a 3 waypoint move 3 times you will end up with 9 waypoints on screen. If a move goes from A to B to C, there is no time defined for moving from C back to A, the user is required to enter in the dialog box.

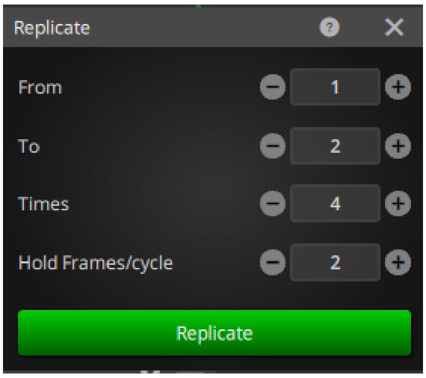


Times 0

Back to 1 0f

☐ Copy colours

Replicate: Allows you to take an existing move or part of an existing move and repeat that move several times. Occasionally, a basic move will be plotted which you then want to repeat for several times. Rather than re-enter the lines again and again, the replicate function can be used to achieve this. You simply have to state the waypoint at the start of the section, the waypoint at the end of the section to be replicated and how many times you want to replicate it. You are strongly advised to save any move to disk before you use this feature in case it does not operate as you had intended. Any disabled axes will not have their move replicated. If you have a move that runs through a series of points say A, B, C, D. If you replicate it 2 times, the resultant move will be A,B,C,D,C,B,A,B,C,D. It walks forwards and then backwards over the move. You can also state whether you want the rig to hold at the end of each cycle by entering the length of the hold in the "Hold Frames/Cycle" box. An entry of 0 means that there will not be a hold.



From 1

To 2

Times 4

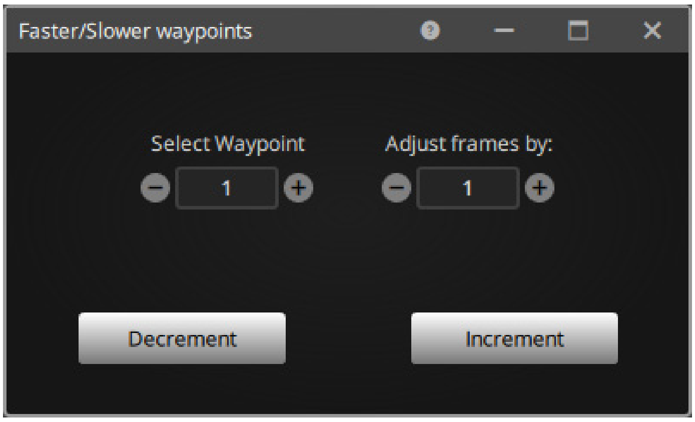
Hold Frames/cycle 2

You have the option to copy the colours of any waypoints, and this can be extremely helpful to see where the move reverse, so colour the last keyframe and then it is easy to see when the move gets to that key frame. The default mode of is to replicate the entire move and when checked, the start and end waypoints will be set to the start and end of the exiting move. If you uncheck this, then the start and end waypoints can be edited. Often, I have been asked to "Repeat the move for 40 seconds" and this requires quite some calculations, so you can also state the length of time to replicate for by leaving the "Replicate Times" blank and entering a number of seconds. The system will work out the number of replications needed to repeat the move to be at least that number of seconds and to end up back at the beginning of the move, so it may last for longer than the stated duration, but it should be at least that long.

Redistribute: A move is often planned as a series of positions the camera must go through without really defining the timings. If you have set up such a move, and want the timings of the waypoints to be evenly distributed proportional to the distances between them, this button will allocate the total length of the programmed move based on those distances with a little extra time at the start and end to allow the rig to get up to speed and allow it to ramp down. This feature is only available in target tracking.

Flip Move: Exchanges the move end for end. Triggers associated with the move are not adjusted. Mostly used if you have programmed the move from the end to the start and would rather run the move forwards than backwards. Note that the positions are changed and the move re-calculated, so depending on other settings, it may not exactly match the move run forwards.

Faster/Slower Waypoint: Sometimes you want to change the speed of a motion as it passes through a waypoint. To do this, you would want to proportionally change the number of frames before and after a Waypoint.

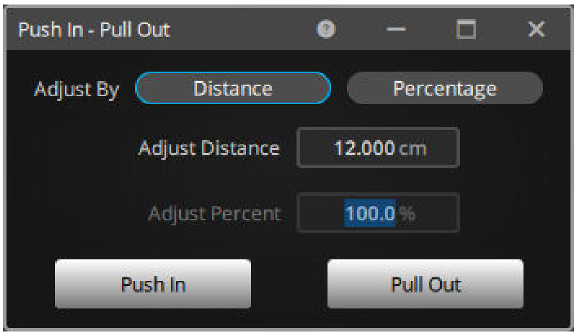

Faster/Slower waypoints

Select Waypoint: 1

Adjust frames by: 1

This dialog allows you to specify the waypoint that you want to speed adjust and the number of frames to add or remove to speed it up or slow it down. The frames are changed proportionate to the number of frames leading to this point and after this point. Adding 24 frames will make the entire move 24 frames longer, it splits the frames to be added between the sections before and after the point.

Push In Pull Out: Runs through the entered waypoints and pushes in or pulls out by a fixed distance or, if you have focus running in Focus Follows Target, then you can push in or pull out to a percentage (for example, 120% pushes in to make the object at the target distance 20% bigger). Take care as the move will be altered and large changes will make a significant change to the move. This is NOT scaling the move, it is simply pulling out a little for a looser frame or pushing in a little.

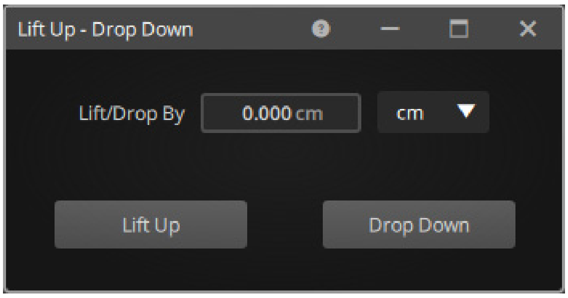

Push In - Pull Out

Adjust By: Distance

Adjust Distance: 12.000 cm

Adjust Percent: 100.0 %

Lift Up - Drop Down: Sometimes after you have programmed the entire move you find that the stand-in was shorter than the talent, or the talent has bigger shoes or bigger hair and you have to adjust every point of the move. This function booms the camera up or down vertically on every waypoint of the move by the stated measure distance.

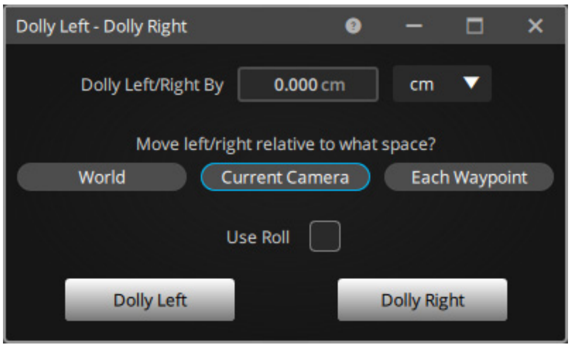


Lift Up - Drop Down

Lift/Drop By 0.000 cm cm

Dolly Left - Dolly Right: Shifts the camera to the left or right at each key frame by the stated amount. Left and Right can be relative to vertical if you do not “Use Roll”, but if you do use the roll axis, then Left and Right should be across the screen.

This function will not work well in Roll Up mode when looking steeply up or down.



Dolly Left - Dolly Right

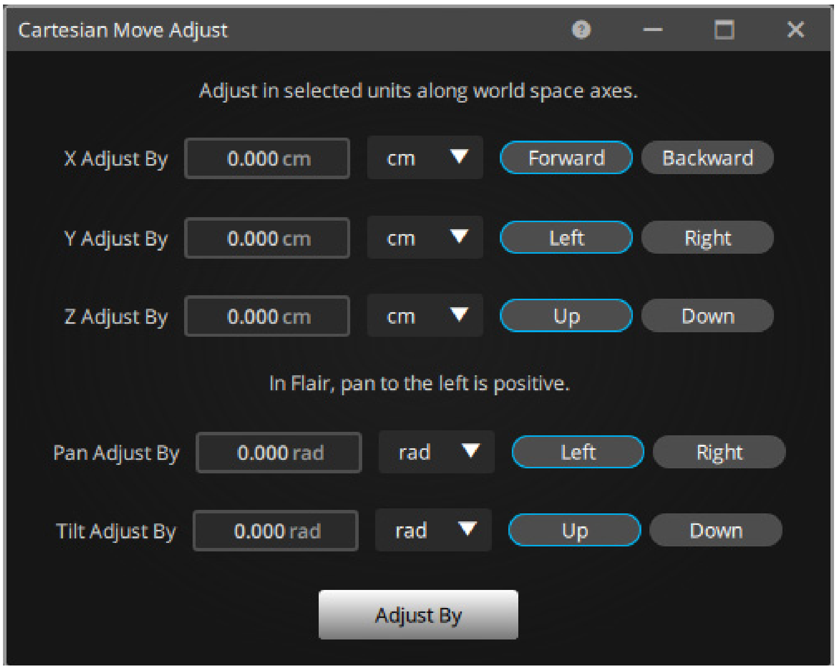
Dolly Left/Right By 0.000 cm cm

Move left/right relative to what space?

World | Current Camera | Each Waypoint

Use Roll □

Adjust Move: You can make simultaneous adjustment to several offsets using the Adjust Move feature.



Cartesian Move Adjust

Adjust in selected units along world space axes.

X Adjust By 0.000 cm cm Forward Backward

Y Adjust By 0.000 cm cm Left Right

Z Adjust By 0.000 cm cm Up Down

In Flair, pan to the left is positive.

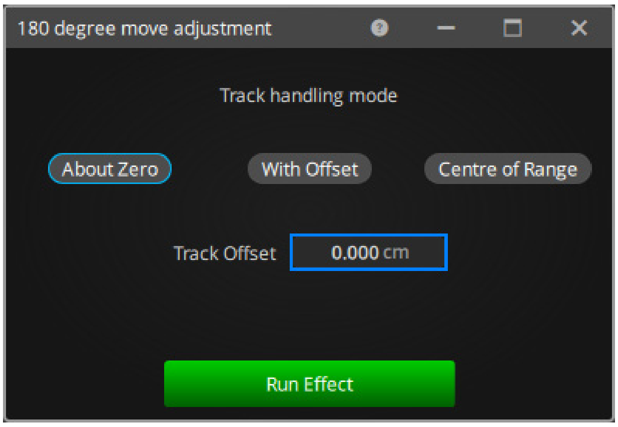
Pan Adjust By 0.000 rad rad Left Right

Tilt Adjust By 0.000 rad rad Up Down

As with the other adjustment functions, the box can be right clicked to calculate the current difference in the rigs position from the position as the current frame count. Alternatively, you can type in your adjustment directly into the numeric entry box. If you are in Camera Orbital mode, the Dialog appears slightly differently as this:

Once the required values are entered, and you have selected which direction the change is to occur, then use "Adjust By" to adjust the move.

180 the Move: Sometimes there is a need to repeat a move at the other end of the track, exactly the same move, but turned around 180 degrees.



180 degree move adjustment

Track handling mode

Track Offset: 0.000 cm

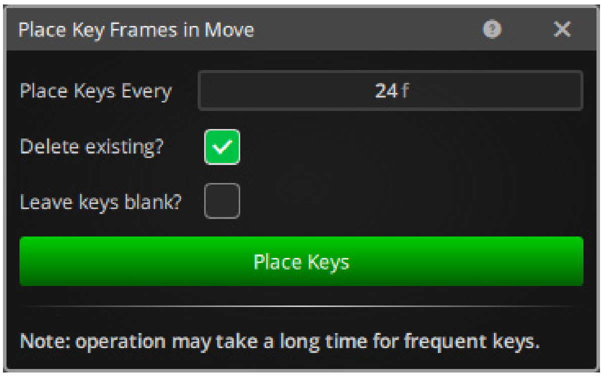
This can be done by reversing the rotate and the other rotational angles and also reversing the track motion. However, this function makes that easy to do and also give the option of reversing the track motion about 0 or reversing it within the range of the current soft limits.

Drop Keys: Editing moves in Flair sometimes requires new key frames to be placed and then edited. This is often done by browsing to a point and then inserting a waypoint and storing, but this requires you to browse or go to every waypoint. If you insert a waypoint at a specific frame count and then disable the waypoint, you can use the "Drop Keys" function to insert the rig positions at that frame count into the waypoint or waypoints. This is particularly useful if you want to "lock" a section of the move down without affecting the motion by dropping keys every 24 frames or so, then that section is fixed. Another useful application of this is adding keys to a 2 point move. If you have a long track with 2 points and you've shortened the fairings to say 15%, as soon as you add a keyframe, it stops using the fairings and makes the ramp up and down very slow. By adding key frames in the middle of the move, and at the end of the ramp up and start of the ramp down, then dropping keys there, you can keep more or less the same sort of motion as you had with the 2 point fairing motion.

Place Keys: Place keys is much like Drops keys but it will work out the frame counts for the waypoints and add them throughout the move.

If the placed keyframe coincides with an existing keyframe, you have the option of deleting the exiting waypoint of leaving it. One of the uses of this would be to convert a Mimic move into a waypoint move so that it can be smoothed out and edited.

.



Place Key Frames in Move

Place Keys Every: 24 f

Delete existing? ☑

Leave keys blank? □

Note: operation may take a long time for frequent keys.

Regenerate Targets: An advanced feature for use with normal target tracking moves. (Does not work with Pan/Tilt target tracking moves)

Regenerate Up Points:

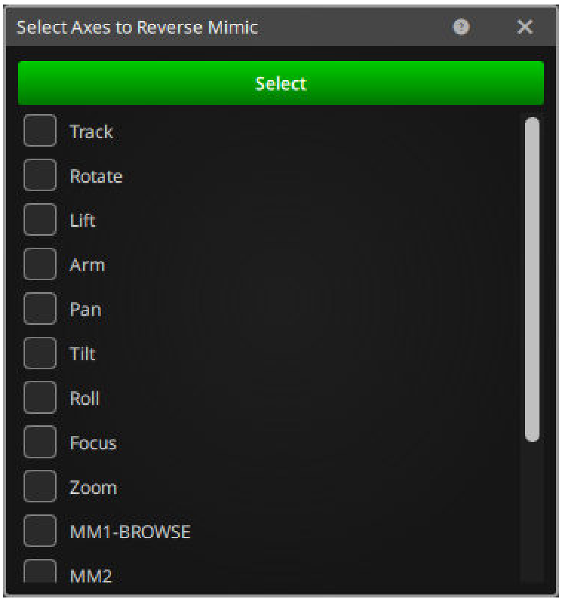
Solve Angle Positions: Advanced reverse kinematics handling which is covered in detail in Chapter 10 Target Tracking.

Euler Filter: A way of removing incorrect large changes in the camera’s orientation. This is only valid when using Target Tracking and is covered under that section of the manual.

Delete Disabled Waypoints: If a waypoint has been disabled by using the middle mouse button on its position column, then the line is effectively removed from the motion of the camera. If you have several of these, this option allows you to delete all of these to clean up the Waypoint view.

Enable All Waypoints: Allows you to enable all disabled waypoints in the Waypoint view.

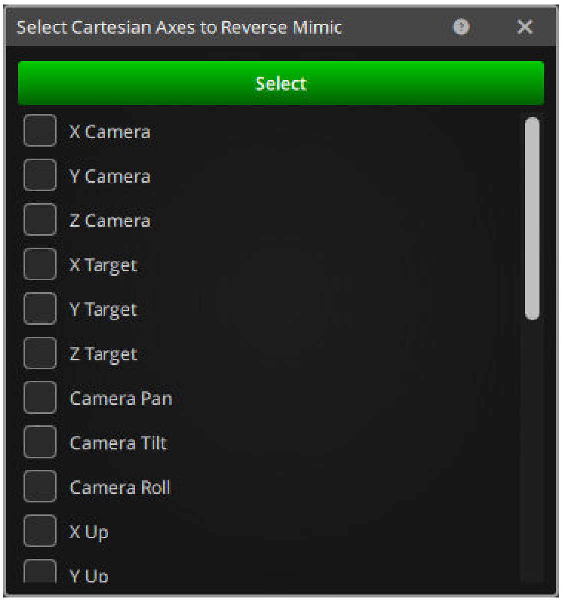
Reverse Mimic Axes: It can happen that a Mimic move learnt in one direction needs to be run the other direction. Whilst this can be done with a “Back Run” sometime that doesn’t work so well with triggers and other features and it’s easier to simply reverse the mimic move.



Select Axes to Reverse Mimic

□ Track
□ Rotate
□ Lift
□ Arm
□ Pan
□ Tilt
□ Roll
□ Focus
□ Zoom
□ MM1-BROWSE
□ MM2

Reverse Mimic Carts: As above with axes, sometimes you want to reverse the direction of a mimic learned or imported cartesian.



Select Cartesian Axes to Reverse Mimic

6.1.10. Mimic

Convert Axes: This function allows any waypoint move on any axis to be converted into a “mimic” move so that there are no waypoints. This can be useful since mimic axes can be edited graphically in ways that are not possible or too difficult with a waypoint move.

Convert Cartesian: Does the same as above but with 3D axes (for example X, Y or Z of camera).

6.1.11. System

Configuration: Allows you to set the configuration of the Flair system.

View Flair.ini: Displays the Flair.ini file used by the system when it boots up. As a standard this is Flair.ini. You may also edit the file through this display. For more detail.

Ini file settings: This lists and describes the settings in the Flair.ini file. There are various tabs at the bottom, which classify the settings based on how frequently they are used. These tabs are: Description, Always, Often, Sometimes and Seldom.

Stop: Fulfils the same function as the mouse stop button, and Is used to stop any ongoing action such as a move, a goto, a movement on the hand held box. It is also used to get the machine out of a “Ready to” run state. When the computer has been told to run a job, it will usually go into “Ready to Shoot” or “Ready to Goto”, when this happens, move entry and many other functions are inhibited. Pressing this button or the mouse stop button will return the computer to normal input mode, and data can be modified or entered.

Enable/Disable: Causes all the active axes to be enabled or disabled one after the other. When the system is first powered up, no motors are enabled for safety. Once the system is correctly setup, all the active axes can be enabled or disabled simply by pressing one of these buttons. All axes should be disabled when the system is turned off at the end of a shoot. The correct sequence is to disable all axes before pressing in the emergency stop switch. Of course, if there is an emergency, it is OK to press the emergency stop as quickly as you can!! If an axis has tripped out because of position loss or has been moved whilst disabled, then it cannot be re-enabled through the use of this button and has to be explicitly re-enabled by clicking on its name button on the numeric display.

Enable All Axes Fast: Generally speaking with a rig with many servo motors, enabling all motors is a very sudden action and can make a loud and surprising sound. This should only be used when you are running with Steppers motors or know that enabling all motors instantly does not cause adverse effects.

Override Hard Limits: Allows the user to temporarily override the hard limits on an axis. There are two options. "Hard Limits Override On" and "Hard Limits Override Off". The axis which is currently selected in the numeric display is affected and its limits can be turned off or back on. It should only be used in emergency situations such as a broken wire. The overridden function is turned off if the motor is zeroed or homed in any fashion.

6.1.12. Tools

This pull down menu provides tools to change or better use the Flair system. These include viewing and editing the system initialisation file (changes are valid ONLY when the system next loads), viewing the system configuration file as well as an "About" screen, recent changes list and a few extra system tools.

Output Monitor: A scrolling window of system messages can be put on screen using this button. In some cases, you might be asked to pull this screen up in tracing down a fault or bug.

Log files: A text file of system messages can be opened using this menu option.

- Application
- Engine

Open Application Folder: Opens the Microsoft Windows folder in which the Flair program is installed.

Calculators: Opens a sub-menu

- **Standard:** Simply brings up the Windows System calculator.
- **Timecode:** Calculations between timecode and frame counts can be quite hard and this calculator can convert to and from timecode.
- **Exposure:** There are many factors that affect exposure and most of them are available on the exposure calculator which can work out the required stop or the required speed for a shoot compensating for the shutter angle, filters and changes in camera speed.
- **Depth of Field:** Not intended to replace the A/Cs depth of field calculations, but this handy tool is based on the standard DOF formula but also compensates as the lens moves (when in follows object/target) and also compensates for the focal length and the iris stop when these are calibrated and set up. Hyperfocal is also displayed.

Import from Flair 6/Classic:

Help

This pull down menu contains a selection of items that can assist the user.

User Manual: Use this to access the entire Flair User Manual for Help topics. You can access the PDF or HTML version.

Send Report: Bug reports or feature requests can be sent directly from the Flair Computer to MRMC if the machine has internet access. This is also the preferred way to send a configuration update to MRMC so that they have a record of your exact system.

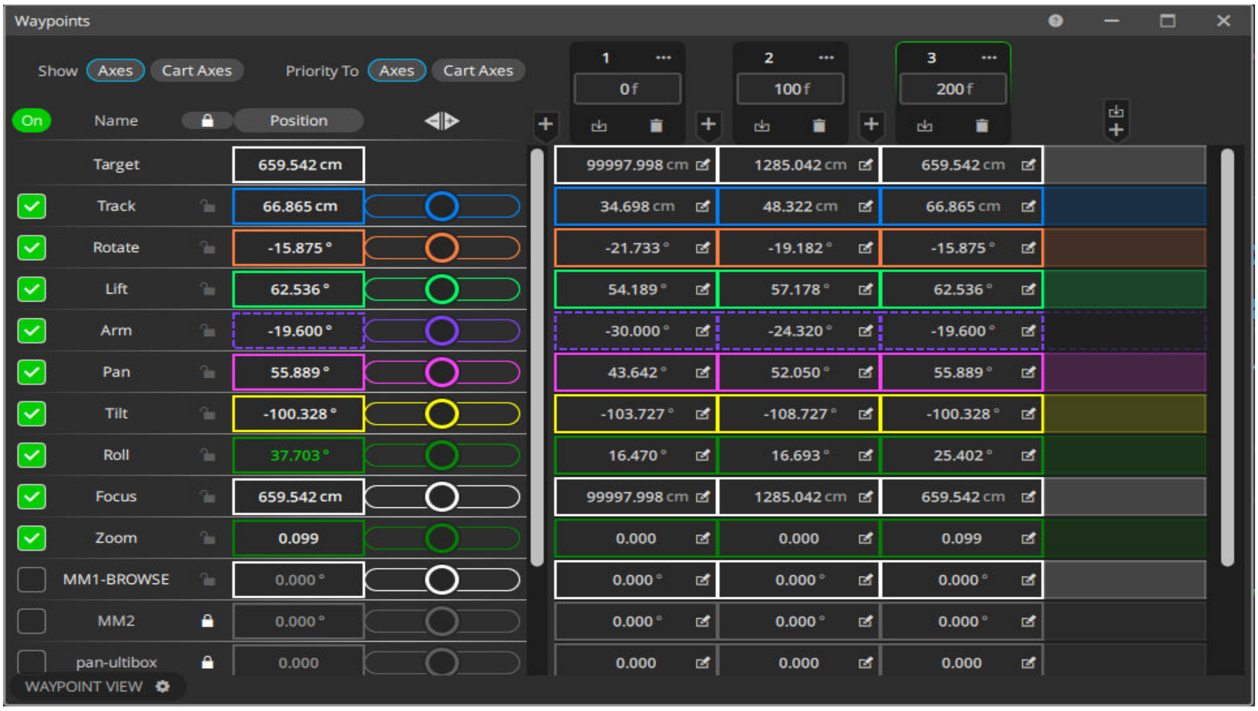
About: Displays data about the version of the software and its author as well as contact information.

License: The Flair software requires a license to read and write jobs and to import and export CGI data. If a license is needed, the system code can be obtained from this display and sent to MRMC for an update.

Changelog: Recent changes to the Flair software are recorded here to inform the user about recent improvements and bug fixes.

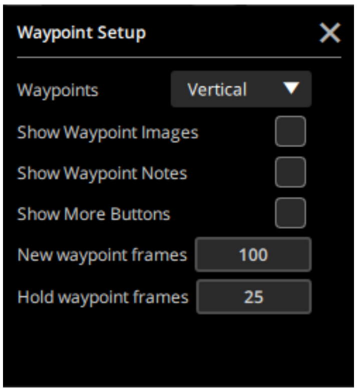
6.2. Waypoint View

The Waypoint view is composed of columns containing data about frame counts and motor positions for those frame counts. Each of these columns is called a waypoint and with each waypoint is stored a position for each axis, a frame count for that waypoint, a target distance for use with target tracking.



On	Name	Position	1 0 f	2 100 f	3 200 f
	Target	659.542 cm	99997.998 cm	1285.042 cm	659.542 cm
☑	Track	66.865 cm	34.698 cm	48.322 cm	66.865 cm
☑	Rotate	-15.875 °	-21.733 °	-19.182 °	-15.875 °
☑	Lift	62.536 °	54.189 °	57.178 °	62.536 °
☑	Arm	-19.600 °	-30.000 °	-24.320 °	-19.600 °
☑	Pan	55.889 °	43.642 °	52.050 °	55.889 °
☑	Tilt	-100.328 °	-103.727 °	-108.727 °	-100.328 °
☑	Roll	37.703 °	16.470 °	16.693 °	25.402 °
☑	Focus	659.542 cm	99997.998 cm	1285.042 cm	659.542 cm
☑	Zoom	0.099	0.000	0.000	0.099
□	MM1-BROWSE	0.000 °	0.000 °	0.000 °	0.000 °
□	MM2	0.000 °	0.000 °	0.000 °	0.000 °
□	pan-ultibox	0.000	0.000	0.000	0.000

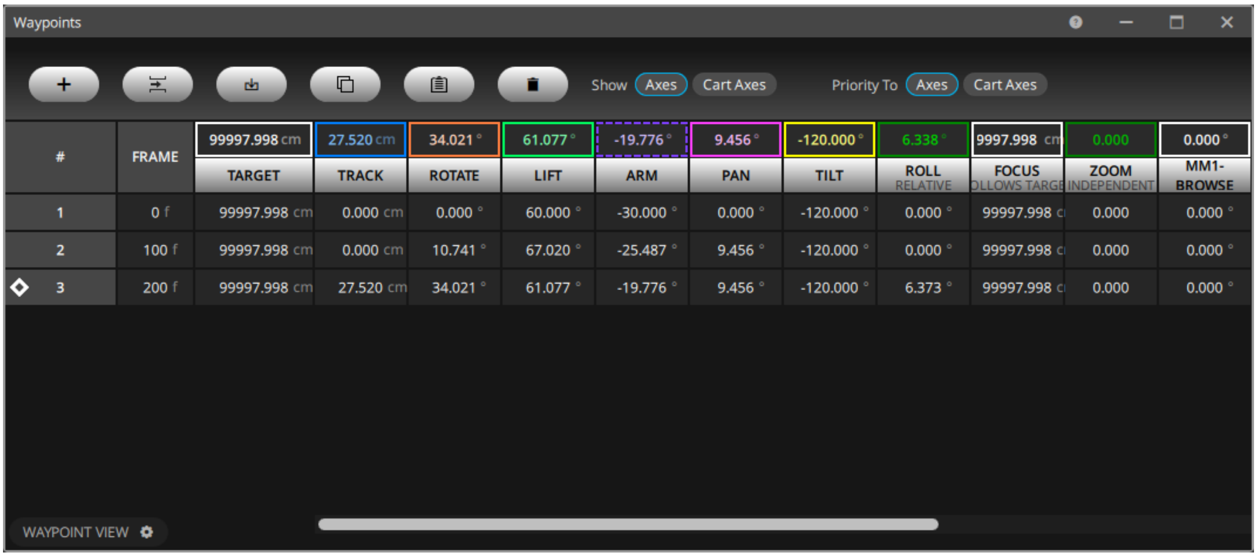
In addition to providing the default vertical view, Flair 7 also has an option to change the Waypoint view to ‘Horizontal’ by using the Waypoint View Setup (⚙) button at the bottom left corner of the Waypoint view window.



Waypoint Setup

Waypoints	Vertical
Show Waypoint Images	□
Show Waypoint Notes	□
Show More Buttons	□
New waypoint frames	100
Hold waypoint frames	25

The Horizontal view is similar to the Numeric Display in Flair 6 in that the waypoints are arranged as lines along with the position for each axis, frame count and target distance.

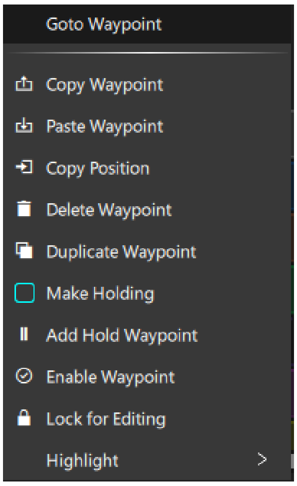


#	FRAME	99997.998 cm TARGET	27.520 cm TRACK	34.021 ° ROTATE	61.077 ° LIFT	-19.776 ° ARM	9.456 ° PAN	-120.000 ° TILT	6.338 ° ROLL RELATIVE	9997.998 cm FOCUS OLLOWS TARGE	0.000 ZOOM INDEPENDENT	0.000 ° MM1- BROWSE
1	0 f	99997.998 cm	0.000 cm	0.000 °	60.000 °	-30.000 °	0.000 °	-120.000 °	0.000 °	99997.998 c	0.000	0.000 °
2	100 f	99997.998 cm	0.000 cm	10.741 °	67.020 °	-25.487 °	9.456 °	-120.000 °	0.000 °	99997.998 c	0.000	0.000 °
◇ 3	200 f	99997.998 cm	27.520 cm	34.021 °	61.077 °	-19.776 °	9.456 °	-120.000 °	6.373 °	99997.998 c	0.000	0.000 °

6.2.1. Positions Row

The first row in the vertical Waypoint view is the position number, and is to let you know which waypoint you are on. If you have more than about 16 waypoints, then they will scroll left and right, and then this column numbering becomes vital. In addition, you can use this row to copy information from one waypoint to another. Click the ellipses next to the position box on the column which you want to be the destination column, and enter the position number of the column you want to copy and press Enter. Then all the data from the source column will be copied into the destination column. Any data in the destination column will be lost.

Clicking the ellipses next to the position box on the column presents a menu that can be used for various operations. If you wanted to add a hold at the end of the move, you would click the ellipses, select **Add a Hold Waypoint** and that would automatically give you a hold at the end of the move.



In the horizontal Waypoints view, the position column is also used for a short hand store feature. If you press the mouse store button while the cursor is over the position column, then the current positions of the axes will be stored into this line. This allows quick data storage with minimal movement of the mouse. Keyboard entry over this column can also be used to Insert or Delete lines. When you have selected a positions number, pressing “Delete” on the keyboard will delete it, and pressing “Insert” will insert a line before the one you are currently on.

You also have the capability of disabling a waypoint so that it is not figured into the move. This is done by pressing the middle mouse button over the position you wish to disable. The buttons number should then grey out. To re-enable, simply press the middle mouse button again. This feature can be activated or de-activated through a setting in the Miscellaneous Setup. It is not recommended that you leave a lot of de-activated waypoints as it may confuse editing or programming the move. If it is not good - delete it, if it is - keep it.

Any waypoint can be coloured (Highlighted) to help distinguish it from other waypoints. This is available in the “Frames Context Menu” by right clicking on a Frame Count entry, but there is a convenient short-hand, by typing the first letter of the Colour you want, the line can be changed to that colour “White”, “Pink”, “Green”, “Yellow”, “Orange”, “Cyan”, “Azure” & “Lavender”.

6.2.2. Frame Count Line

The frame count line is on the top of the Waypoint View, and frame count numbers are entered in here by selecting the number you want to change with the mouse so that a blue square appears around it, and then typing in the new value followed by the “Enter” key. Frame counts can start at a absolute non-zero value though this is not recommended, but they cannot reduce as you go down the screen, they must always increase.

There are several additional features provided for ease of use, and these are described here:

- You can store the system current frame count into a line by typing the frame count value in the waypoint. This is particularly useful when you are inserting a waypoint into a move, the best way to do this is to part run the move to the frame count that you want to modify, insert a line, adjust the positions of the motors as needed, store in the current frame count of the move as above, and then store axes positions. The store (✎) button is available for each waypoint and will store the current axis positions and frame numbers in the waypoint.In the horizontal view holding the shift key when you right click with store the current frame into this box immediately.

- Once a series of waypoints have been added, you may want to allow more or less time between two waypoints this can be done by adding the increase (say 25 frames) to the frame count at the second of these points, entering that new value, and then continuing all through the move adding 25 to each frame count. If you didn't do this, the section of the move subsequent to your change would also be changed. A shortcut is to simply select the second waypoint, type in the number of frames you want to change it by, and press the "+" key to increase that and all subsequent frame counts, or press the "-" key to reduce them. This feature is called "ripple through" as the effects of any changes you make are "rippled through" the move. The standard maths features are all available, ending any number sequence with "+", "-". "*", or "\" will perform that mathematical function on all subsequent frame entries. If you entered "2*" in the first frame count position, it would double the length of the move. Since only integral numbers are allowed in the frame column to convert a frame count from 24fps to be the same time as when run at 30fps, one could type "30*" <enter> then "24/" <enter>. Be sure to always multiply up before you divide down.
- When a move has been entered, the entire move can be scaled up or down to make it shorter or longer. There is a button labelled **Scale Move** on the Timeline Control view, and when this is pressed, a dialog box appears prompting you to enter the new total number of frames. If your move goes from 0 to 300, you can type in 600 in this box, press OK, and the whole move and any associated waypoint frame counts will be scaled up to make the whole move last 600 frames. This is also done automatically by the computer if your move is too fast and you Apply the number of frames the computer asks for in order to shoot the move (See section under running a move). Other options for scaling the move up and down are also provided for on this dialog.
- The stretch feature is an extension of scale. If you want to extend or compress only part of the move, it involves both a scale of that section and a ripple through for the rest of the move. If you press the **Stretch Move** button on the Timeline Control view, then you are prompted for the waypoint number of the beginning of the stretch, the waypoint number of the end of the stretch, and the desired frame count for the new end of the stretch. In this way the speed of the move through a series of waypoint can be changed without changing the timings of the rest of the move. Note that a new curve will be generated through all the waypoints with the new times and areas outside of the "stretched" section may also be changed slightly.

Incremental Frames + Seconds Line

The frames column by default displays absolute frames, but can be toggled to display different formats. Seconds, Incremental Frames, Incremental Seconds and Timecode. In the **Seconds** mode, the time is shown as a timecode number with the format **HH:MM:SS:FF** (Can be adjusted through the Miscellaneous Setup to not show Hours and Minutes) The seconds values are a direct translation from the frame count entered and are converted based on the transfer speed set in the Miscellaneous Setup. The **Incremental** mode displays the difference in the time between waypoints (Either frames or seconds). The **Timecode** mode show the time of each waypoint assuming that the move starts at the starting timecode shown in the Input Setup ("Move Start") This would mostly be useful for Music Videos/Promos as events can be timed against the running timecode of the song. Note that the time is the time of the final shot when transferred, it is not the time it takes to shoot it. This is an important distinction; if you want to display the shooting time which will change when you change speeds, then you can select that in the miscellaneous setups with a toggle next to the transfer speed edit box.

Display Frame Counts and Time simultaneously

Pointing on the Frame count shows the time for the waypoint. Previously in Flair, you could show the frame count OR the time for a waypoint, but this sometime meant that you would have to switch back and forth when editing and adjusting the move. You can now display both the frame count AND the time for each waypoint by selecting "Always Show Frames" from the "Frame" button context menu.

Beats and Bars

Flair works in the time base of frames. If you enter a time in minutes, seconds and frames it is stored as a frame count inside Flair. Likewise, if you switch to the "Bars And Beats" display, the bars and beats are converted using the values in the "Bars And Beats Setup" display for Beats per Bar and Beats Per Minute.If you enter frames, then that is converted to the nearest Beat.

Goto Frame Count per Waypoint

If you click the ellipses on a given waypoint frame count, there is an option to "Goto Waypoint" that frame count. Remember that it will go to the rig position that matches the stated frame and will not necessarily go to the positions listed on the waypoint.

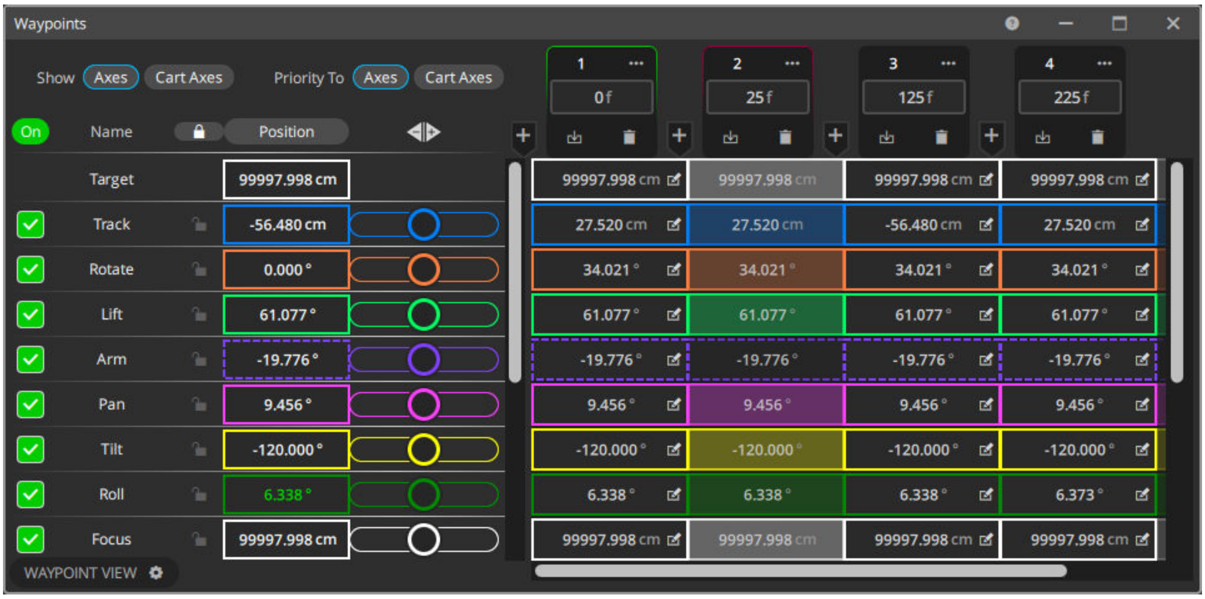
Hold Move Mode

The traditional Flair display assigns each waypoint a frame count at which it occurs. This works fine for many situations, but it is a little harder to create a hold as you have to have two identical waypoints and there is no clear indication that it is a hold. If by mistake you change one of those waypoints, but not the other, the hold disappears. The Hold Move Mode lets you state the number of frames you wish to hold at a waypoint and also the number of frames for a move to the next waypoint. This simplifies programming moves with a lot of holds in them. A blank hold value or a blank move value indicates that there is no hold time for that waypoint. A blank or zero move value with a difference to the next waypoint will cause axes to jump suddenly and should be avoided. You can switch back and forth between these modes by right clicking in any of the frame, hold or move buttons at the top of the display. Switching to Hold Move Mode will alter the programmed frame counts and double keyframe holds will not be removed when you switch. It is not compatible with Bezier curves. The default setting for Hold Move Mode is off, this can be changed in the Timeline Control view using the Hold Move switch. The default setting for Hold Move Mode is off (false), this can be changed in the .ini file.

```
*UseHolds:   False / True
```

When you switch into Hold Move mode, additional boxes appear which allow you to add number of frames that move should hold for and number of frames the move should take to move to the next waypoint.

The Screen here shows a move from waypoint 1 to 2, a hold at 2 and then a move to waypoint 3.



Name	Position	1 0 f	2 25 f	3 125 f	4 225 f
Target	99997.998 cm	99997.998 cm	99997.998 cm	99997.998 cm	99997.998 cm
☑ Track	-56.480 cm	27.520 cm	27.520 cm	-56.480 cm	27.520 cm
☑ Rotate	0.000 °	34.021 °	34.021 °	34.021 °	34.021 °
☑ Lift	61.077 °	61.077 °	61.077 °	61.077 °	61.077 °
☑ Arm	-19.776 °	-19.776 °	-19.776 °	-19.776 °	-19.776 °
☑ Pan	9.456 °	9.456 °	9.456 °	9.456 °	9.456 °
☑ Tilt	-120.000 °	-120.000 °	-120.000 °	-120.000 °	-120.000 °
☑ Roll	6.338 °	6.338 °	6.338 °	6.338 °	6.373 °
☑ Focus	99997.998 cm	99997.998 cm	99997.998 cm	99997.998 cm	99997.998 cm

Some additional options are presented for holding moves when you click the ellipses next to the position number for any waypoint.

There is an alternate way to make a hold in Flair and that is to make a waypoint “Holding”. This means that it will duplicate the positions of the previous waypoint. When this is selected, then a red background appears for the waypoint and the positions in the waypoint cannot be edited. However you can edit the prior waypoint and cause it no longer to be a hold.

Add a Hold: This inserts a line about the line you are on and copies the positions into the new line, it adds a number of frame to your current line and ripples that through the rest of the move.

Make Holding: This copies the previous waypoint on top of this waypoint and therefore makes a hold from the previous frame count to this one. This waypoint may not then be edited and the colour of the position number changes.

UnMake Holding: Undoes the above.

Target Line

The Target line contains the data for the Target distance while you are target tracking. This is also optional and will only occur if you have target tracking enabled in your configuration file. The column may need to be filled in for any target tracking waypoints, and target distance entered in the units specified. The units can be “Metres”, “Centimetres”, “Inches”, “Feet” or “Feet and Inches”. When a line of data is stored and the system is in target tracking with Focus Follows Target option selected (See Job Type section) then a target distance will be suggested to the user based on the focus axis position. For example, if the lens is set up and focused on 3 metres, a distance of 300 centimetres will be entered into the target column on the relevant line by the computer. This value can be used, or ignored and overwritten. The currently used units can be changed by adjusting the setting in the Miscellaneous Setup.

Right-clicking the mouse on the actual “Target” button itself will call up a pull-down menu with options for selecting the measurement units and Lens Setup dialog.

Axis Positions Line

The rest of the screen is filled with axis positions in horizontal lines, with the axis name on the left, and a position stored in each waypoint that is registered. If there are more than about eight axes, then the axes can be scrolled vertically so that you can get to them and study or modify their positions.

On the left of the Waypoint view is the column with buttons that allow the user to enable an axis. Checking the boxes in the **On** column on the left would enable all the axes at once. Likewise, to disable an axis, click the box again the check mark will disappear and the axis will turn off and the motor will be disabled and any brake will be engaged to prevent the axis from moving. If the axis trips out for any reason (its axis name will change colour to Pink) then click it once to disable it, then again to re-enable it. If you click the lock button for any axis, so the button assumes ‘locked’ position, the axis will be inactive. This means that although it will still hold position, it will not move either from the hand held box of from a programmed move. This state is the default for any axes not immediately displayed on the screen, and is useful for axes that are not involved in the move as they will not be involved in limits checking, curve generation etc. Clicking the ‘lock’ button again will make the axis active meaning that it will participate in the move.

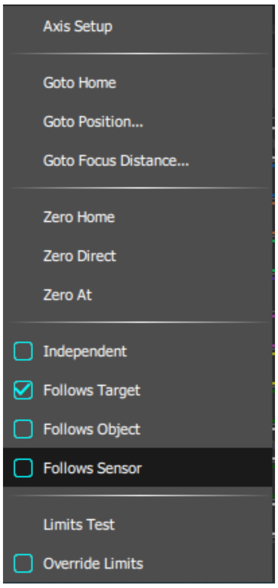
Left clicking on the Axis Name will bring up a context menu with Axis Setup option, clicking which will open the Axis Setup dialog box for that axis. The axis can be zeroed, sent to home or a known position etc. The control mode of the focus axis for example can be switched between “Independent” and “Follows Object” or “Follows Target”.

Axis positions are modified in the same way that frame counts are, simply select the number you want to change so that a black square appears around it and then type in the new value followed by “Enter”. Positions can also be stored into axis position boxes simply by positioning the cursor over the relevant box and clicking ✎n the relevant axis position box.. Positions can be deleted to leave a blank field by selecting the field and then typing the backspace key on the keyboard. You can modify a position and all subsequent positions for a given axis by selecting the first field to be changed and typing in the amount you want to change it by and ending this with the “+” or the “-” key to either add or subtract that amount from that entry and all subsequent entries, as the “ripple through” feature for frame counts does.

A motor position is usually shown in box with same colour as that of the axis. If the position shown is NOT an axis position, but some other derived value such as the Focus Target Distance or the zoom focal length, then the position will be shown in a different colour. These colours are described in Current Positions on page 130. In some control modes, the motor position is still shown in this column and in the actual entered values, this will be in black, even if the current motor position is shown in colour.

6.2.3. Context Menu

A Waypoint View context menu is available if you left click on the axis name. The Context Menu provides you some options that are available for that particular axis.



Context menu for Focus axis

6.2.4. Axis colours

On the right of the Axis Name column, there is colour column where you can set the colour for each axis. Once this is done, the axis will appear in that colour for all waypoints. Note that these colours are not the same as those in the LiveView.

6.2.5. Locked axis

The next column is axis locked and the buttons in this can be toggled to lock any axis. When an axis is locked, it can’t be moved by on screen controls or external controls such as HHB and is indicated by a ‘lock’ appearing in the column and a strike through line on the axis current position.

6.2.6. Current Positions

The area just above (or on the right, in Horizontal view) of axis name is used to display the axis current position or velocity. This display has various modes. The Current Value may be displaying the current position of the axis in display units (for the focus, this can also be a target distance – other axis such as the roll have similar alternate colours indicating a that the value show is in a different mode). **Current Velocities**

The current positions column can also be toggled to display the current velocity of the motor. If the column heading Position is clicked then it will change to show the current velocity of the axes. If the axis trips out when it is in this mode, the number continues to show the current intended velocity of the axis.

Blank Entries

If a waypoint exists, but a related box is blank, that means that no position has been stored for that axis. In fact if you simply add waypoints without storing any data, you will get entire column of blank boxes. The boxes will stay blank until some positional information is stored in them. The significance of a blank box is exactly what you would expect, it does not hold a position, and therefore does not affect the move in any way. If you have data in a box, but want to blank it out, select the box by clicking on it, and then press the backspace key on the keyboard. You may not have blank entries in the frames column.

Values Displayed

The numbers displayed in the axis positions boxes are generally in display units. When the axis is set up, it is specified what the relationship is between the motor turns, and the distance the axis moves. This means that the computer can interpret motor movement in turns or centimetre, or inches or degrees. The units that appear on the numerical screen will be in the units that the motor has been scaled for. One exception of this is the focus axis which is displayed in display units – like the other axes - when it is acting as an independent axis, but is in terms of focus target distance when the computer is controlling the focus (see Job Type Section and Target Distance data immediately above). There are other specialised axes that can display their data in other forms, but for most axes, it is simply the motor position scaled to be in workable units.

Slider Bars

If your system has more than about 8 axes, there may not be sufficient room to display them all on the screen at the same time. In order to overcome this problem, the axes are displayed as data shown through a “window” and to access the later axes, you have to scroll the data underneath the window. If you click on the arrows at the right hand end of the scroll bar, or click on the scroll bar and drag it down, any hidden axes to the right are displayed. The names change to indicate that you have scrolled across, and the positional data changes. To get back to the earlier axes, simply drag the rectangular section of the scroll bar back to the left hand side, or click on the left hand arrow as needed. The vertical scroll bar fulfils a similar function, but it is for the waypoints.

There is no practical limit on the number of waypoints that you can have in a job, however we would strongly recommend that you keep it under 30 otherwise navigating around the waypoints becomes very tedious and also too many waypoints too close together can often cause the move shape to develop unwanted curves. Should you have more than one screenful of data, the left hand vertical scroll bar can be used to access any data that is not immediately visible. Moves of up to 750 waypoints have been run, but this is unusual, and very slow in storing to disk.

6.3. Rig Control View

On the left of the 3D View is the Rig Control View with options related to the editing and manipulation of a job, The “Rig Control View” buttons are related to how the job is run and special features. Some of these buttons are multi state buttons and every time you click on it, it will cycle on to the next state, most of these buttons have only two states, but some have three or more. Other buttons with more states have a “pulldown” menu which will appear when clicked on, and the desired mode can be selected. The function of each button is described here:

HHB Off is a toggle button with three states; **HHB Off**, **HHB On** and **HHB + Grips**. It defaults to HHB On and must be explicitly turned on before you can move axes on the Hand Held Box. Its main use is to deactivate the motion functions of the Hand Held Box as a safety precaution and to allow the user to easily turn off the Grip Sticks if they are provided.

HHB Speed is a slider that you can use to control the master speed of the axis controls.

Tracking simply toggles the target tracking switch on and off. The toggle is only active while the move in being input and cannot be changed at run time. For more information, and the other states of this toggle see Chapter 10 Target Tracking.

Cart Control The options for start are Off and Robot Tracking.

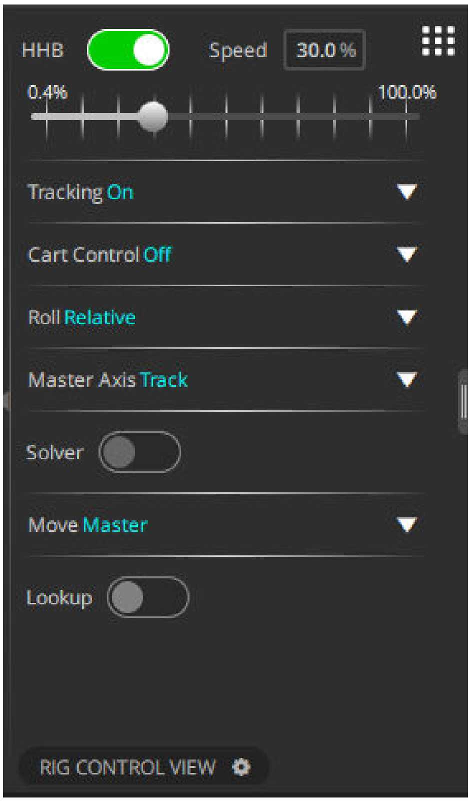
Carts options are World, View, Roll and Up.

Locked options are World, View, Roll and Up. These options and any additional ones are fully explained in the section "Cartesian Control" in Chapter 10 Target Tracking.

Roll With the "Angle" axis swinging the head over the top of the arm, if you are panned to one side, this will result in a roll motion. This roll motion may not be desirable, so the computer can compensate for this action to keep the horizon level (**Level**). This of course is fine unless you want to be able to add some roll motion to the move. In this case, the selection **Relative** allows you to program roll motion that is relative to the horizon rather than being relative to the zero position of the roll axis; so regardless of action that the roll motor is doing to compensate for the angle axis, the roll will also move clockwise and anticlockwise as you have programmed. Note that in **head Level** this function is irrelevant and should be left as roll **Moves**. In roll **Relative**, the roll axis positions are displayed in Green to indicate that they are not true axis motor positions. When you have the roll being controlled in this way, you may not zero or home any of the axes that affect its positions (i.e. Lift, Angle, Tilt, Pan), you must first take it out of roll control to do this. If the roll trips out or it turned off when in roll Level, then when it is switched back on, it will request a "Goto" to the level position. Roll **Up** is a fourth option which is only valid in normal target tracking moves. It avoids the problem of looking at the floor and ceiling when there is no horizon, it basically assigns a point that defines where "up" is and moves the roll axis to follow that as it moves.

Move When you are shooting a target tracking move, you can also scale this move up or down in size and shift it around in space. When you are doing this, you can either run the main move (Master Move) or the scaled and shifted move (Slave Move). For more information on these and the other available options see Chapter 10 Target Tracking.

The various options that control the rig in a special mode - Head Control, Roll Control, Cartesian Control etc, are selected by choosing the state you want. When this is done, either the control function will be activated, or a Goto will be required, and once the Goto is finished, the control function will be activated. If the Goto is interrupted or for some reason the control state is not activated, the label will switch back to reflect this. Only one should be selected at a time, and once its "Goto" is finished, another option can be chosen. Note that if you are in “Keep Head Level”, it makes no sense to use any of the roll controls as the roll will automatically behave normally in this case.



HHB Speed 30.0 %

0.4% 100.0%

Tracking On

Cart Control Off

Roll Relative

Master Axis Track

Solver

Move Master

Lookup

RIG CONTROL VIEW

6.4. Control View

Scale Move When a move has been entered, the entire move can be scaled up or down to make it shorter or longer. When this button is pressed, a dialog box appears prompting you to enter the new total number of frames. If your moves goes from 0 to 300, you can type in 600 in this box, press OK, and the whole move and any associated waypoint frame counts will be scaled up to make the whole move last 600 frames. There are also other ways or expressing a re-scale and there are options for making the move faster or slower by a percentage. You can also cause it to last a timecode duration. (for example, 00:01:22:10 would make it last 1 minute 22 seconds and 10 frames)

Stretch Move The stretch feature is an extension of scale. If you want to extend or compress only part of the move, it involves both a scale of that section and a ripple through for the rest of the move. When pressed you are prompted for the waypoint number of the beginning of the stretch, the waypoint number of the end of the stretch, and the desired frame count for the end of the stretch. In this way the speed of the move through a series of waypoint can be changed without changing the timings of the rest of the move.

Motion is a toggle button that has 2 states, **Stop Motion** and **Moving Step**. The state of this button is only relevant when you are stepping through a move (frame by frame animation). If you are in stop motion, then the exposure (Film or Video) occurs while the motors are stationary and the move occurs separately. If you are in moving step, then each forward step of the move is shot at the speed that it would occur at if you ran the whole move continuously. This allows animation sequences to have motion blur and avoids the strobing effect that can occur with stop motion animation. Note that Moving Step will only work if the move can be shot continuously at the programmed camera speed.

Step is a toggle button that has 2 states; **Single** and **Continuous**. The state of this toggle is only relevant when you are stepping through a move. When in Single step, every step requires you to push a button to trigger it, when you are in Continuous step, the steps will occur automatically one after the other until you stop it, or the move ends. You can also turn off Continuous stepping by pressing this button while it is shooting, the button will change back to Single Step and the current step will complete, and the system will stop in a state ready to continue.

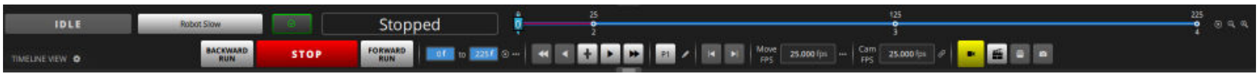
Mimic is a toggle between when you are learning a mimic pass **Learn** and replaying one already learnt **Off**. The name Mimic Off is the default state and is really mimic replay when axes are selected to mimic. There is a third option called **Learn+** which is explained Chapter 12 Mimic Mode.

Triggers Triggers **On/Off** just allows you to set if the output triggers will come on during a move or not. Very useful if your triggers are connected to fireworks and you want to check the move is reasonable without triggering anything accidentally.

Start If you have a timecode reader configured to operate with your system, then you will be able to see the current time code and the move start timecode displayed. The “Start” entry is the currently set start time of the move. It is set by using the **Move Start** box in Input Triggers display or by directly entering a number into the box. You can either right click with the mouse to set the current timecode into the box or you can type the timecode number into the box (You do not need to type the “:”s). Timecode digits cannot be directly edited, you have to type the timecode again. If you do not type enough digits, the value is preceded by zeros. (For example, if you type 510 it will be understood as 00:00:05:10).

6.5. Timeline View

The Timeline View occupies the middle section of the screen and contains a message area and a large slider bar as well as a few other boxes. The message area is used to tell you what the status of the system is. There are a number of status messages that vary from “Printing” and “Shooting” to “Going To”. When the button is labelled IDLE, you are at input, the hand held box should operate, and you can enter moves etc. In most other cases the message will be related to a move for example, “Ready to Shoot” etc. and user input will be restricted. To return to input mode, the mouse stop button or the Top Menu Bar “STOP” button can be used. The layout and button configuration of this bar can be changed by the user - see Section on “Adjust Numeric Controls”



6.5.1. Move Slider Bar

The slider bar is used to show the current position of the system in relation to the move. When a move is run, the slider bar moves across informing the user where he is in the move. When the move has completed the slider bar marker stays at the last location that the system ran to. You can also use this bar to move the system around the move, simply click on the marker, drag it to where you want to go to and release, the system will then do a “Move To” to that frame count. Clicking in the slider bar on one or other side of the index pointer will cause the system to move to a frame count 1 frame before or after depending on which side of the index pointer you click. This can be useful for inching through a move, but do not click when the system is in a run state as the results are unpredictable. Do NOT click and hold down the mouse in the slider bar as this will be interpreted as a large number of separate clicks and the system will go into a continuous stepping mode which is hard to get it out of. The number of frames that you step through the move in this way can be changed by setting the number of **Frames per Step**; see the “Special Effects” section of this manual for more detail.

At each end of the slider bar are small boxes at the beginning and end of the slider respectively. These inform the user of the start and end frame counts of the move as it is being run.

Main Buttons

The row of buttons on the Timeline view allows you to shoot the move, goto a start position or other position in the move or step through it.

GOTO/SHOOT: This button is normally labelled **IDLE**, but when the system is ready to go to a position, it says **GOTO**, and when the system is ready to run is says **SHOOT**. These three functions are all combined into one button to increase ease of use.

The Forward Run button is used to run a move from the beginning to the end. The camera direction is independent of this.

The Backward Run button is used to run the move from the end to the beginning. The camera direction is independent of this.

▶: The Goto Next Waypoint button performs a goto to the next waypoint after the marker position.

◀: The Goto Previous Waypoint button performs a goto to the previous waypoint before the marker position.

▶|: The Forward Step button is only valid when the system status is “Ready to Shoot” or “Ready to Step” and when pressed under these conditions, the system will step through the move by onestep.

|◀: Similar to Forward Step button, Backward Step button steps through the move by one step backwards.

✛: The Browse button sets the system into a mode where you can drive the rig along the path of the move slowly or fast using the hand held controller. When pressed, the system will need to move to the current frame count of the move before going into the "Browsing" state. The current frame count is shown by the position of the slider bar. Take note of this as it may not be where the rig is or where you expect it to be. This button will also allow you to Browse a goto. Goto can drive the rig rapidly in an unexpected fashion and should be executed with care. This button allows you to Browse the goto as if it were a normal move, allowing you gentle variable speed control over the goto. If you see a goto that is longer than you expected, it is always wise to Browse the goto rather than just execute it. If your move is very fast or has no fairings at either end, be sure to browse the move slowly to prevent undue stress on the rig, the rig will not preroll or postroll at the ends of the move, it will just come to a stop.

P1: The Run Back to the Start button runs the move backwards at slower speed and automatically turns the camera off. This is used for shooting repeat passes without having to turn the camera off or reduce the speed of the move. It is only valid when you are at the exact end of the move. The move will normally be run backwards at 1/3 of the camera speed, but you can set the run back speed by setting the camera speed to the runback speed you want in the Move FPS box. A message will appear telling you what the run back speed has been set to. Note that the output triggers will NOT fire during a P1. Also note that this runs the move back to the start along the move path and is different than a Goto which would not run the camera along the move path.

⏺: The Rehearse/ Record button is a toggle that sets the move to be a Rehearse shoot or a Recorded shoot. When in "Rehearse", the whole move will be executed but the camera will not run. When this is changed to "Record", the move will be shot with the camera rolling. If the system camera is off (i.e. disabled) and External Camera is ON, then the move will be shot synchronising to the External Camera. If you are running an internal camera, then this button is active during the run and can be used to start and stop the camera running during the move as long as the camera and move are both being run forwards. (Do not try this is you are running a variable camera speed move of any kind)

Move FPS: This is the move speed in frames per second. Any positive value can be entered into the box below this button up to the maximum speed of the camera and down to 0.001. The default values of this can be changed if needed by MRMC, but otherwise it will usually be set to 25.00. This button has an additional function when you are running a move with a varying camera speed – see Chapter 14 Variable Camera Speed for details.

Cam FPS: This is the speed the camera will run during the move. Most of the time, the camera speed will be set to the move speed, and if you change the move speed, then this value will change accordingly. Occasionally, you want to run the camera at a different speed to the move. This occurs occasionally, but Note that moves shot with different camera speeds, but the same move speed will NOT match. If you wish to change the speed of the move and the camera, simply enter the desired speed in the "Move FPS" entry box. A right click in this box will copy the current move speed into the camera speed box. This box defaults to a disabled state to reduce the change of inadvertent changes occurring. To enable this box, click on the box above labelled "Cam FPS"

The Roll a Slate button causes the camera to expose a certain number of frames when it is pushed. It will only work when the camera is engaged and is set up to record. For more information see the sections on Running a Move and Camera Axis Setup.

: The Turnover or Roll Camera button sets the camera rolling at the current move speed. It can be used to shoot a hold of an inexact length or it can be used to shoot a slate (a slate is a board on which is written the take number). When pressed a second time, the camera will stop. If you are at “Ready to Shoot” and you press this button, then the camera will start rolling; if while the camera is rolling, the “Shoot” key is pressed, then the move will start in sync with the already running camera. At the end of such a move, the camera will continue running until it is stopped unless there is a post hold set, then the camera will stop after that number of frames. A running camera may also be stopped with the mouse stop button.

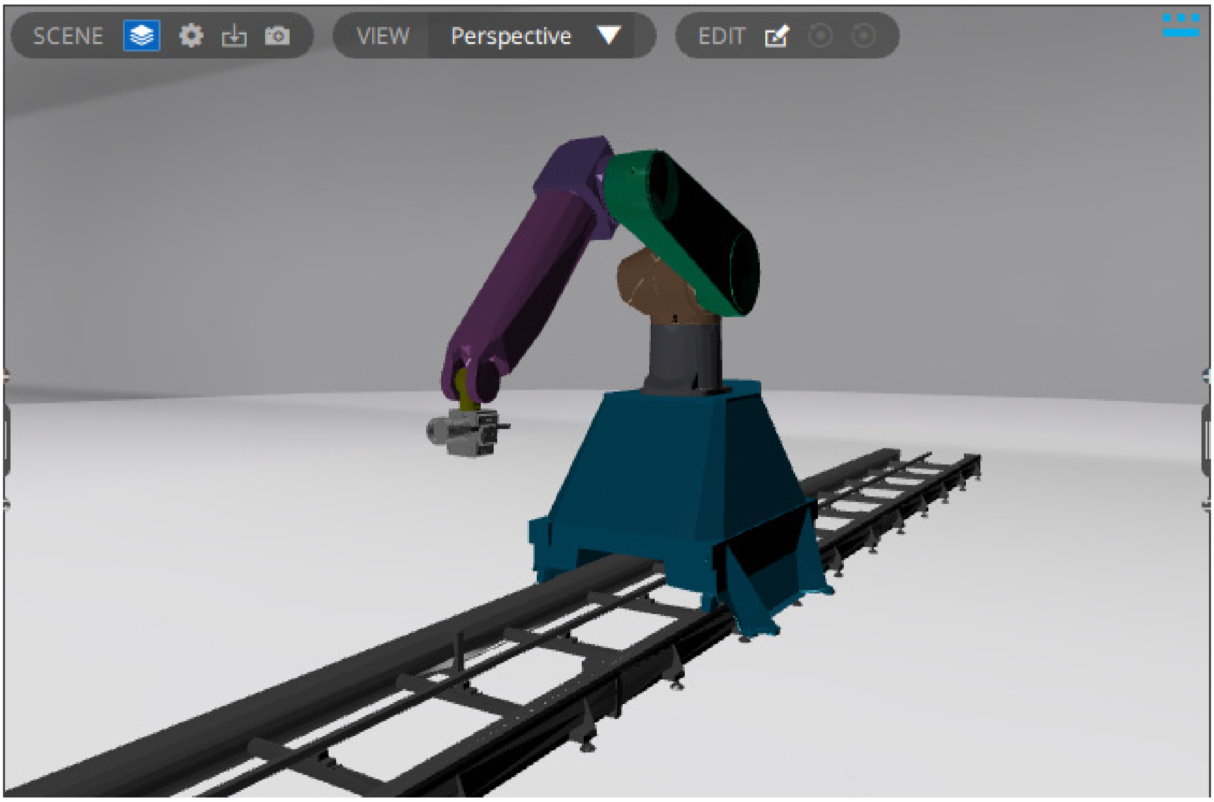
: Clicking the Take button will cause the camera to expose one frame at the current camera speed. If you are at a speed higher than a single frame speed, then the necessary ramp frames will also be exposed. This button does not work when you are in a run state such as “Ready to Goto” or “Ready to Shoot”.

STOP: Stops any on-going action or will take the system out of a run state (for example, “Ready to Shoot”) if needed.

6.6. 3D View

Since Flair can run in a stand-alone mode with no rig attached, it is invaluable to have some representation of what the rig would be doing in any given move. The 3D View provides that and much more.

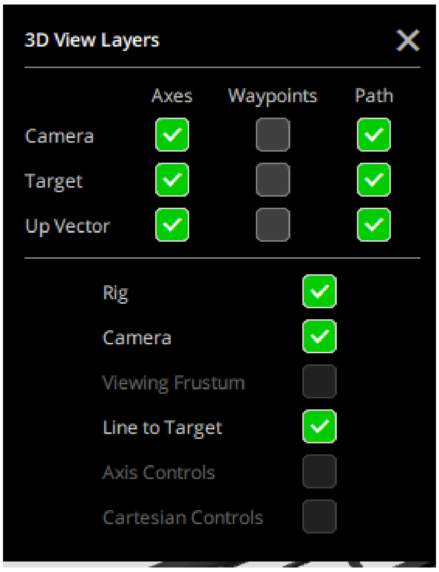
The display is a 3D view that allows you to view the rig through a Perspective or Orthogonal camera as well as being able to view through the camera mounted on the rig. Many of the rig designs that are used with Flair are supported and carefully picking your kinematics model will assist in making sure that the rig is properly displayed.Supported Models are: Bolt, Bolt Jr+, Bolt X, Milo, Talos, Modula, Animoko, UltiHead.



6.6.1. Scene options

Layers menu

Use the options to alter the 3D View layers.



3D View Layers

	Axes	Waypoints	Path
Camera	☑	□	☑
Target	☑	□	☑
Up Vector	☑	□	☑

Rig	☑
Camera	☑
Viewing Frustum	□
Line to Target	☑
Axis Controls	□
Cartesian Controls	□

Camera Axis: Shows the Cartesian movement axes of the camera.

Camera Waypoints: Shows the position of the nodal point of the camera for each waypoint in the move.

Camera Path: Shows the path that the camera is taking during the move.

Target Axes: Shows the Cartesian movement axes of the target.

Target Waypoints: Shows the position of the target for each waypoint in the move.

Target Path: Shows the path the target takes during the move.

Up Vector Axes:

Up Vector Waypoints:

Up Vector Path:

Rig: Show or hide the rig.

Camera: Show or hide the camera on the rig.

Line to Target: Adds a viewing line from the camera to the target which can assist in seeing what the camera is doing.

Rig Settings

These allow to change the camera mount on the rig model. You can change the mount to **Stirrup** or **3 Way**.

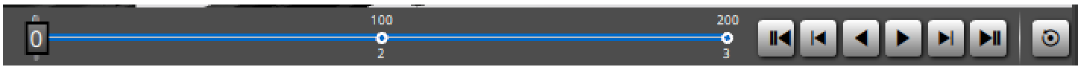
Camera Settings

Use these to set the camera mounted on the rig.If you do not have a camera yet and are just trying out Flair, then specify the camera as Best Guess, which is also the default option. Other supported camera types include: **ARRI 435**, **Mitchell**, **RED**, **Canon 5D**, **Phantom Flex** and **Phantom Flex 4K**.

6.6.2. View

The View options allows you to change the point of view of the rig model. These include: **Perspective**, **Camera**, **Top**, **Front** and **Side**.

Run Control



The run controls are just to the right of the Slider Bar and consist of 8 buttons:

The ||< and >|| keys will drive the rig to one end of the move.

|< and >| cause it to jump to the next keyframe.

< and > play the move forwards and backwards.

The middle mouse button anywhere in the screen acts as a stop button.

The Repeat Animation button causes the rig to run through the move on a loop when selected.

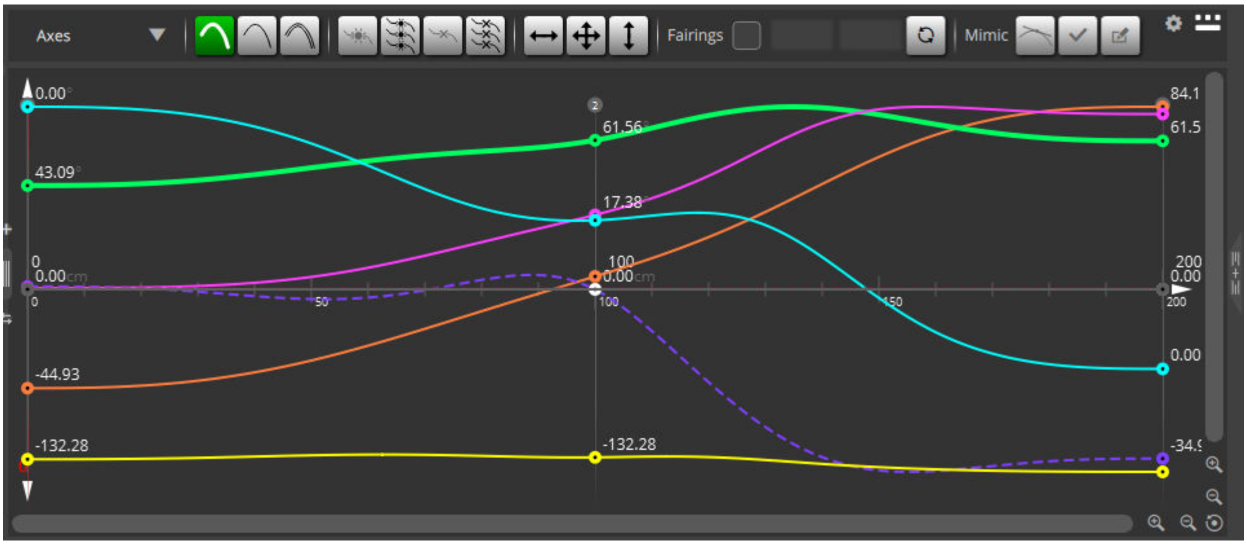
These buttons control the motion of the rig model and NOT the rig on the move. In addition you can use the mouse to move the rig in the 3D model to view from different perspective.

6.6.3. Pre-rolls and Post-rolls

Since rig pre-roll and post-roll is an important part of safely running a move, any time a pre-roll or post-roll is needed by the rig, the rig model will also show you the same pre-roll and post-roll. This can be surprising as it is never seen in a CG package, but can be very helpful to see what the rig might do when you run it. These motions are not however included in the AVI capture.

6.7. Graph View

The Graph View that appears by default on the lower right on the Flair screen displays the move on any axis as a graph of the axis position against time. The Graph View displays the variable number of graphs are displayed on the screen up to the number of axes in the system, and down to 1, along with various functions for viewing and manipulating these graphs. Like with other views, you can show, hide, or display it as a attached or unattached window.



6.7.1. Appearance

Each curve displays the information relating to one axis. The colour of the curve corresponds to the axis in the Waypoint view. Hovering the mouse over the graph shows the axis name and its position in terms of frame value and motor postion. Along the graph are shown the waypoint values for any waypoints occurring within the limits displayed by that curve. Each waypoint for that axis which is within the area displayed is shown as a small circle, and the axis path is shown as a solid line. When the cursor is within the window of a curve, its position in terms of frame counts and axis position is shown in the top left hand corner to allow accurate inspection and editing of the graph.

Hovering on a curve until it changes to a thicker solid curve and then clicking it will hide all the other curves so you can inspect one curve closely. Click anywhere on the graph will make the hidden curves reappear.

The drop-dwon list allows you to change the appearing graphs between **Axes**, **Cartesian Axes** or **Speed Profile**.

: Shows or hides position curves for various axes.

: Shows or hides velocity curves for the axes.The velocity profile for the move will only be correctly displayed once the move has been calculated with a Forward Run or a Back Run. If the velocity graph is not wholly displayed on the graph, then simply cause the move to be fully calculated, and the graph will be correctly scaled. (You can cause the move to be calculated by pressing forward run, and then “STOP”, you do not need to actually run it). You cannot edit the velocity graph directly.

: Shows or hides acceleration curves for the axes.

: Zooms the graph to extends which means that the graph will now cover all the available region that is visible.

: Zoom in the graph to horizontal extents when clicked at the bottom or zoom in the graph to vertical extents when clicked at the right.

: Zoom out on the graph vertically or horizontally depending on when clicked on the right or bottom, respectively.

: Resets the zoom level of the graph to bring all waypoints and all axes in view.

: Causes a position to be added at the mouse cursor's current position and time. In the numeric display this will be represented as a blank line with a position for only the axis whose graph you were in.

: Causes a position to be added for all axes at the mouse cursor's current position and time. In the numeric display this will be represented as a blank line with a position for only the axis whose graph you were in.

: Causes the selected axis waypoint to be deleted. On the Waypoint view, the entered value at that waypoint will be blanked out.

: Causes the selected waypoint to be deleted. On the Waypoint view, the entered value at that waypoint will be blanked out.

↔ : Allows to modify the selected waypoint’s time value for all axes.

✥ : Allows to modify the time and position values of the selected waypoint. When Edit graph is selected, the mouse can be used to edit or change the move. Simply position the mouse close to the waypoint you wish to change, press and hold the mouse left-button and drag the mouse up or down; a small square will appear which will travel up and down in a straight line (only the position changes, not the time). When you release the button, the waypoint that you chose will be changed to the new position you have selected, the move will be modified, and the graph redrawn to show the new move. You will actually be changing the numerical value of the axis position in that waypoint, and if you look at the move again in the Waypoint view, you will see that the number has changed.

If the axis movement you are looking at is a mimic recorded move, the operation of the “Edit Graph” button is somewhat different, see the later section on “Edit Range”.

↕ :Allows to modify the selected waypoint’s position value for all axes. Simply position the mouse close to the waypoint you wish to change, press and hold the mouse left button and drag the mouse up or down to drag the waypoint up and down in a straight line (only the position changes, not the time). When you release the button, the waypoint that you chose will be changed to the new position you have selected, the move will be modified, and the graph redrawn to show the new move. You will actually be changing the numerical value of the axis position in that waypoint, and if you look at the move again in the numeric display, you will see that the number has changed.

Fairings: When you check this option, the fairings position markers will be shown on the curves. Use the boxes next to the checkox to change the fairing on the selected

Tip for editing graphs

As with entering a move, the best way to achieve the shape you want is to modify the existing waypoints, not to add lots of points.

↻ **:** When pressed near a waypoint, allows curve reversal to occur at that waypoint.

↻ : Causes curve reversal phenomena to be stopped at the closest waypoint to the mouse when clicked.

The subject of curve generation and curve reversal is covered fully in Chapter 7 Move Entry. This should be studied in order to get an understanding of what this function does.

Smooth: When you have a mimic learnt move being displayed on the system, you have the option of smoothing it.

L- Select Zone: A single press allows the user to drag a box showing the range of time over which the move will be smoothed.

M- Smooth: Clicking this button causes the selected zone to be smoothed, each subsequent click smoothes the move more and more. Each smoothed graph is drawn over the top of the last graph, so you can see the effect as you smooth it. Shift Clicking in this mode will do a “Super” smooth - this means that the number of mimic sections in the selected range will be halved which has a dramatic smoothing effect and should be used cautiously.

Chapter 7 **Move Entry**

7.1. Moving the Motors

7.1.1. Engaging a motor

When the system is first powered up, no motors are engaged, the user has to engage them. The first thing to do is to turn on the power to the amplifiers by letting out the red emergency stop button and pressing the e-stop reset button if supplied. The state of a motor is indicated by the green check mark in the Enable/disable motor column next to the its axis name near the top of the numeric display, and by the colour of its "current position" which is directly above. In the Horizon view of the Numeric display, if the axis name is grey, that indicates that it has not been enabled - simply click on the axis name, and it should go white indicating that the user has enabled the motor and wants it to be engaged. If this is successful, then the current position, which was also white, will go black. The motors can also be enabled all together if the 'On' button in the Enable/Disable All column on the top is pressed. This button will turn green even if a single axis is enabled. In some robots, the robot axes can be engaged or disengaged only by using the Engage/Disengage Robot button in the Run Control bar.

7.1.2. Manual Movements

Once the motor is engaged and under power it may be moved on the hand held box, simply move the joystick, or push the button associated with that axis, and it will move. If a motor does not move then you are probably in a run state and your request is being refused. The On box turning red indicates the axis is tripping. If a motor enables, but does not engage, then the axis has hit a limit or is not properly connected - check the cables to the motor and try again.

Velocity Readouts

As well as showing axis position you can also display the velocity of an axis at any time. . The Position column can be toggled to Velocity to show the velocity of the axes during the move.

Changing manual speeds

Using the "fast" and "slow" buttons on the hand held box, you can increase and reduce the speed that a motor travels at. The behaviour of the motors when they are controlled by joysticks or the buttons on the hand held box is a function of the axis setup (See Axis Setup on page 19). If you find that the motor does not move as fast as you like even when you are at fast speed on the hand held box, then the Hand Held Box Velocity in the relevant axis Setup can be changed, but be careful not to set it too high, or you may find that fine positioning of the motor becomes impossible. The current setting is displayed on the hand held box and also in the very bottom right hand corner of the screen, where it is displayed as a percentage. With mouse jogging, you can also change the Hand Held Box speed by first clicking on one of the axis current position displays and then pressing the PC keyboard "+" and "-" keys to increase and decrease it.

The hand held box is used to move the motors around so that waypoints can be plotted for the move. The positioning and plotting of these points is key to the end result of the move.

7.2. Programming a Move

7.2.1. Basic law

One of the first laws in programming a waypoint type move is to use as few waypoints as needed to achieve the path required. If you “store a point, move a little bit, store another point” and so on, you will almost invariably end up with a poor move. The best way to enter a move is to decide on the start position and plot that, then plot any VITAL mid points and then the end point in as few points as possible. Then run the move and see what happens.

7.2.2. Positioning Waypoints

The best way to determine where to place a waypoint is to put it at an extreme of the movement. If you have to start far from an object, get closer and then move away, the obvious points to plot are the start and end, and a midpoint which is at the closest you want to be to the object. This has to be done with care and you have to take into account the likely path of the move to avoid any obstacles, but it is quickly learnt with practice.

7.2.3. Entering Waypoints

The actual entry of points is achieved by moving the motors using the hand held box to the desired position and getting the framing correct, then press the Store Waypoint (⤓) button above the column where you want to store the current axis positions in the Horizontal Waypoint View or by highlighting a waypoint and clicking (   ) in the Vertical Waypoint view.

7.2.4. Adding Waypoints

When more waypoints are wanted at the end, click the Add New Waypoint (⊞) button on top of the last waypoint, and a new line will be added to the move, then move to the correct position for that waypoint, and repeat the storing process.

7.2.5. Inserting Waypoints

If you wish to insert a waypoint in the middle of an existing move, then you click the Insert a blank waypoint button (⊞) for the waypoint after which you want to insert an additional waypoint. For example, if you have a 2-point move and you want to put a new column in between, simply click ⊞ on top of the first waypoint. If you want to insert a waypoint with current axis positions, click the Insert waypoint (⤓) button in the middle of two waypoints where you want to insert the new waypoint. If you are going to insert a waypoint in an existing move, the best way to do it is to position the motors to the correct position for the intended waypoint BEFORE you insert it, then insert it and then store the positions.

7.2.6. Duplicating Waypoints

Use the Duplicate waypoint button (▮) on top of any waypoint to insert exactly a waypoint with exactly the same axis positions. This can be used for example when you want to modify only a few axis positions in the newly inserted waypoint. This is also useful when you want to add a hold to the move

7.2.7. Copying waypoints

If you have a move that involves coming back to start position at the end of the move or a move that involves going back and forth on certain position, you might find it useful to copy waypoints. Simply click the Copy waypoint button on top of the waypoint that you want to copy and

7.2.8. Modifying a Move

Once the move is entered in its basic form, you may find that it is not exactly the move you wanted. The first thing to check is that it is correct at the waypoints. If it is not, change these until it is and only start adding waypoints when you cannot effect the change you want with the existing waypoints. Adding a waypoint is best done by finding the frame at which the departure from the intended move is greatest. (You can do this by watching the move and seeing where the slider bar is when the move leaves the desired path by the most.) Then run the move to the place where it is worst, insert a waypoint at that point of the move, then correct the positions so that the correct path is more closely followed, and store the necessary positions. Then run this and continue correcting as needed to get the path right. Again, it is usually better to use fewer rather than more points.

7.2.9. Deleting Waypoints

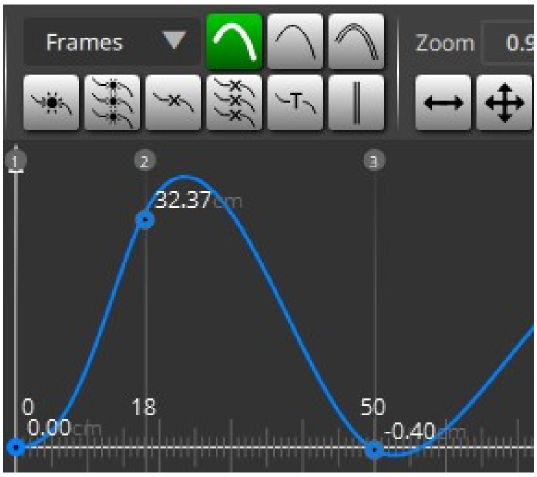
Waypoints can be deleted by simply clicking the Delete (🗑) button on top of any waypoint. A waypoint may also be disabled to see if that produces the required effect before deleting it. In this case press the middle mouse button over the waypoint position number and the number should grey out. This waypoint will not be removed from the system, but will not be factored into any move. Re-enable in the same way.

7.3. Curves and Straight Line Moves

A path can either be a straight line or a curve. If you have only 2 points, then the usual path is a straight line between them, when there are more than 2 points, the path is usually a curve. When there are numerous points, the computer will calculate a smooth path that passes exactly through all the points. You usually have to have at least 3 points to create a curve.

7.3.1. Curve Form

The actual behaviour of the computer curve fitting may not be exactly what you wanted; for example if you had a move followed by two positions exactly the same (a hold), the computer generated curve would naturally go from the first point to the second point, then overshoot it and end up at the third point. This would not be the desired effect, but it is the smoothest way to execute the move as planned. This particular occurrence can be prevented by informing the computer that 2 identical positions next to each other are a hold, and the motor is not supposed to move between them. (This is done through the Job type display, see the relevant section of this manual for more information).



Graph showing same move with “holds can move”.

Notice the “curve reversal” overshoot described below.

7.3.2. Curve Behaviour

The above example is rather trivial, but there are more subtle effects that have to be watched for. One particular effect has been termed “curve reversal”. In our use, this term means that while moving between two points, an axis moves through a position that is not between the two points. If for example you had 3 points on the tilt, each with 50 frames between the next. The first point is 10 degrees, the next is 11 degrees, and the last is 25 degrees; since the second part of the move is much faster than the first, the natural curve would cause the motor to back up from the start to less than 10 degrees in order to get up to the required speed to carry out the second part of the move. Whilst this is the most natural and smoothest curve, it may not be what the user actually intended, and if you have a camera moving along close to a floor, and then going up fast, the natural curve would cause it to hit the floor! This phenomenon can be prevented by selecting **Prevent Curve Reversal** in Fairings Setup or Graphics Display.

7.4. Times and Timings

7.4.1. Changing the length of a move

The overall length of the move is given by the final frame count minus the start frame count divided by the camera speed. The way to change the length or speed of the move is to make the move last for fewer or more frames or possibly to change the speed of the camera. (This can only be done with a film system, not a video system). If you wish to scale the whole move up or down so that the basic move and relative speeds remains unchanged, .

Adjusting the timing of part of a move

Other variations can be affected by using the “Stretch” feature which is also fully described in the above-mentioned section. The third way of adjusting the timing is to use the “ripple through” feature. This allows the user to change the time between 2 waypoints without affecting the others. Normally if you increase the frame count of a waypoint, you make the time between those waypoints more, but you automatically make the time to the next waypoint less, unless you add the same amount of frames to every subsequent waypoint. This is done automatically for you when you use “ripple through”. Simply type in the number of frames you wish to change the waypoint by and press the “+” key if you want to add them, and the “-” key if you want to subtract them. The computer will then automatically modify that waypoint and all subsequent waypoints by the amount you have entered. The effect is ‘rippled through’ all later waypoints.

7.5. Fairings

7.5.1. Introduction

When you are running a move with the user defined fairings, and it is a 2 point (straight line) move, the axis will go from rest up to a constant velocity, continue at that velocity, and then ramp down to a stop. We term the ramp up as the “Up fairing”, and the down ramp as the “Down fairing”, or the “Start” and “End” fairings. The ramps are not linear, but change the velocity of the axis in an “S” shape, this allows a smooth acceleration or deceleration.

7.5.2. When can fairings be used?

A start and end fairing can be entered for between any 2 points where the motor comes to a stop at both the start and end points. So you can have fairings for any 2 point move, and for between any 2 points in a multi-point move as long as the axis comes to a complete stop at each end of the fairing. In target tracking the fairings can also have an effect on a move. The paths of the camera target and viewpoint will be traversed as per the fairings set for each. You cannot use a 0 fairing within a move (zero fairings have no ramps, they are a constant speed move or one that starts and stops instantly), these are ONLY valid at the very end and start of the move, otherwise you would get a pre-roll or post-roll within the move which would be meaningless, and would spoil the shot. If you have a multi-point move with the axes coming to rest within the move, you can set different fairings for each waypoint both for the independent axes and for the target tracking axes. See Fairings Setup on page 58 for more details.

7.5.3. Fairing Duration

A fairing can be as long or as short as the user wants. A fairing is often defined as a percentage of the move, and when the move is run, the respective fairing will last for the stated percentage of the move. If a fairing is stated as being of 0 length, then the axis will start or end the move at a constant velocity and will have to pre-roll or post-roll to achieve this. A fairing can last for the entire move in which case it will be changing speed throughout the move. In the case of overlap, the fairings are truncated. For example, if both fairing are set at 100% of the move, then the computer will cut them back each to 50%. In a section of a move with fairings, the percentage duration is the percentage of that section of the move, not the entire move duration.

7.5.4. Fairing Shape

10 different fairing shapes are provided and these vary the way in which the axis gets up to speed during the fairing. These types are numbered from 0 to 9 inclusive. The lower the fairing type number, the earlier in the fairing the axis gets up to speed which means that it accelerates harder to start with and then eases off. A fairing of value 5 is approximately symmetrical and a higher number fairing starts accelerating more slowly and accelerates harder nearer the end of the fairing.

Type 5 has been found to be the best in most cases, but for a better idea of how these work try it out for yourself.

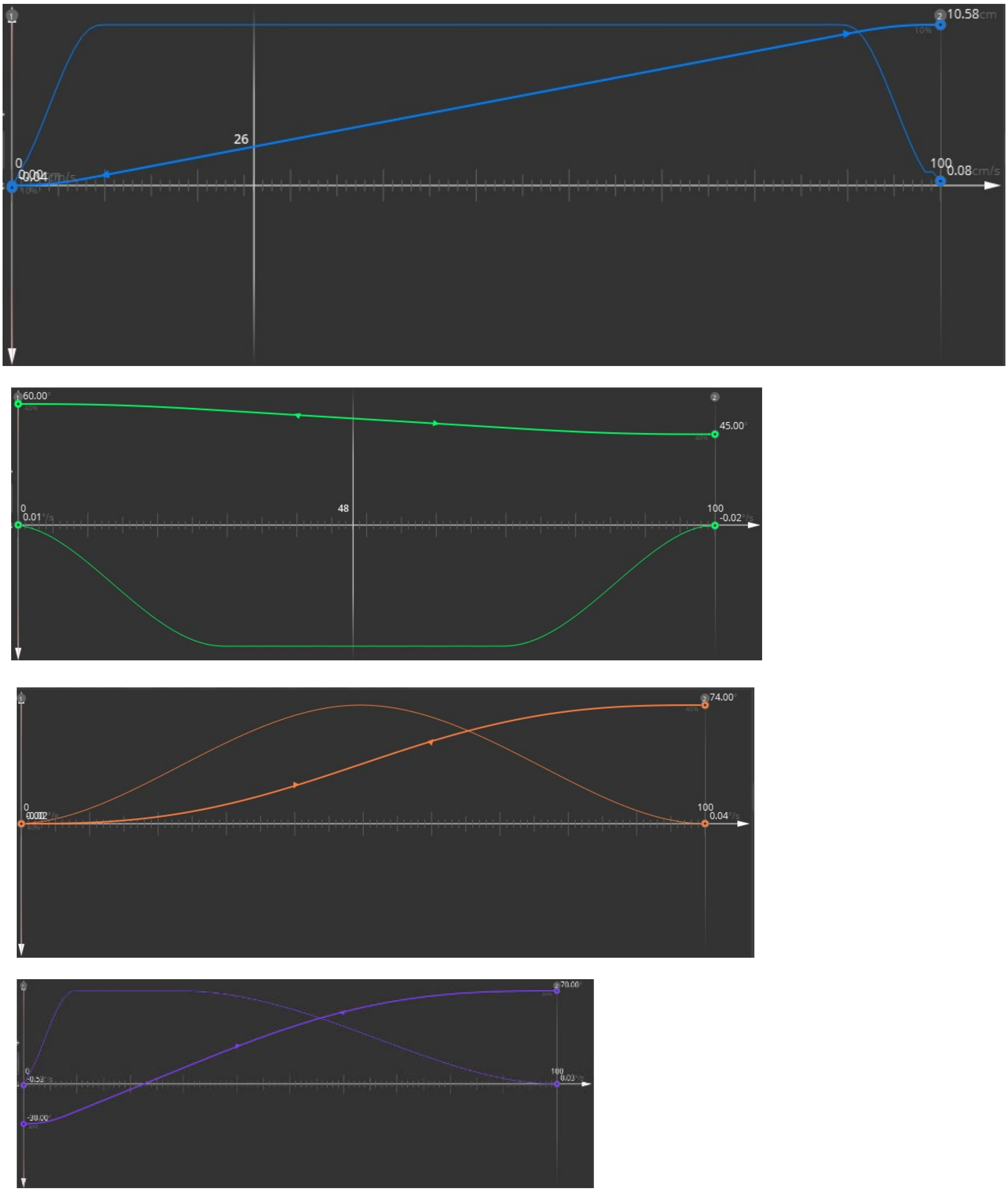


Diagram shows position and velocity graphs for the same two waypoints but with different fairings. Track – 10% fairings. Lift - 30% fairings. Rotate – 50% fairings. Extend – 10% up fairing, 70% down fairing.

7.6. Exponentials

7.6.1. Introduction

The idea of exponential moves is that you can state the ratio between the start velocity and end velocity of a move. This can be used to create unusual speed effects and also to provide a constant rate of change of field size as you pull out from an object. A normal faired move as above, has the ratio between the end and start velocities of 1 to 1, and is a special form of exponential.

7.6.2. When can exponentials be used?

An exponential ratio can be entered for between any 2 points where the motor comes to a stop at both ends. So you can have an exponential for most 2 point moves, and for between any 2 points in a multi-point move as long as the axes comes to a complete stop at each end of the exponential. Exponential moves utilise fairings to start and stop, and these can be adjusted through the fairings display. The exponentials can also be used in target tracking. The same theory as above is applied to the speed with which the target and viewpoint traverse their paths.

Note

When in target tracking, the values entered into the first axis are used to set the target tracking exponential ratios for all target tracking axes.

7.6.3. How to define an exponential

Exponentials are achieved by entering a ratio between the start and end velocities other than 1.0 to 1.0. If you enter 1.0 in both columns, the start and end velocities will be the same, and you will have a standard faired move. If you wanted to get a constant rate of change of field size on a move, you could achieve this by measuring the field size at the start, measuring the field size at the end, and entering these numbers directly into the exponential ratio for the move. See in for a more detailed description of the entry of exponential ratios.

Chapter 8 **Running a Move**

8.1. Forwards or Backwards

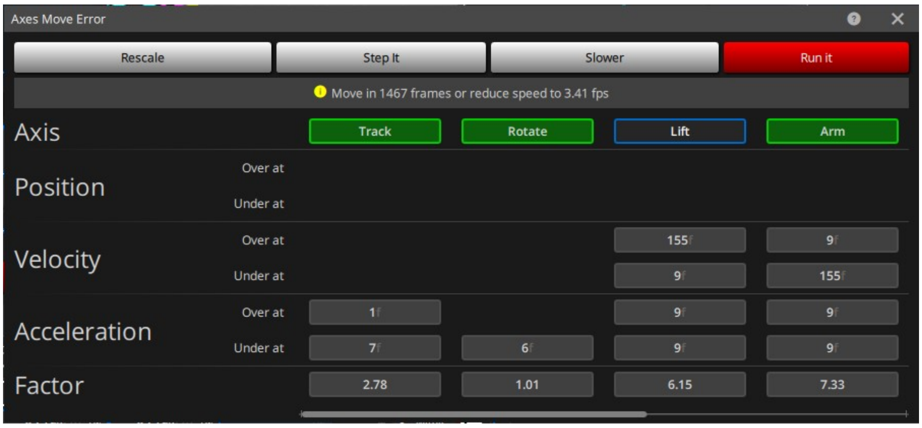
Any move once programmed can be run forwards or backwards. Press either FORWARD RUN or BACKWARD RUN on the Control Box, and the computer will check that the move is complete and there is no omitted data, calculate paths for each axis and check that each axis can run the job as entered. If everything is within allowable limits, the computer will compare the axes current positions to the move start position, and if they are different, the computer will go into a state of **Ready to Goto**. Once the **Goto** button is pressed it will then move straight to the start of the move. Once it is at the start of the move, the computer will go into the **Ready to Shoot** state, and by pressing the **Shoot** button, the move will start.

8.1.1. Job checking

Before a move is calculated, it is first checked to see if it is complete. This makes sure that there are at least two positions, that there are times entered for the waypoints as needed, and that the frame counts increase through the job, and do not reverse. Any errors at this stage will produce an error message stating the error. If you are going to shoot a target tracking job, there are other stipulations that the move must obey, and these are covered in the target tracking section of this manual. Once the move has been verified as OK to calculate, the computer makes a path for each axis, and then checks each axis individually to ensure that it does not exceed the limits of its travel, does not exceed its maximum velocity or acceleration. Often while you are programming a move, you may need to process a move so that when you go into the graphs, the curves are displayed in range, or you want to find out if the rig will run the move without actually getting it into a **Ready to Shoot** or **Ready to Goto** state. This can be achieved with the warning button in the control bar. If the move is OK and will run the warning disappears, otherwise the normal form of over limit or over speed warnings will appear.

8.1.2. Error Handling

If any axes fail these checks, a warning button appears in the control bar and clicking it displays a message detailing each axis that fails stating what test was failed and at what frame count. If any axis exceeds its travel, you will need to fix that, but if the only problem is that one or more axes are running too fast, then you will be given the option of scaling the move up to more frames so that it can be run slower or slowing the camera down if you have a film camera.



Move in 1467 frames or reduce speed to 3.41 fps

Axis		Track	Rotate	Lift	Arm
Position	Over at				
	Under at				
Velocity	Over at			155f	9f
	Under at			9f	155f
Acceleration	Over at	1f		9f	9f
	Under at	7f	6f	9f	9f
Factor		2.78	1.01	6.15	7.33

The box that appears will have five buttons on it, a **Rescale** button, which will scale up the move for you, a **Step It** button which will allow you to shoot the move only as a number of stationary exposures, a **Slower** button which will reduce the camera and move speed to a more achievable level and a **Run It** button which will ignore the limits and run the move anyway. You can even close the box to allow you to modify the move as you like using the waypoint view.

The axis name appearing on the top of the display is set in a background that shows how serious the error is. Any axis that appears, has a problem, but a light green means nearly OK to run, and as the axis is less and less able to achieve the move, its background colour turns to red which indicates that the move cannot be achieved by that axis by a factor of 2 or more.

Some of these options will not be available if any axis exceeds it limits of travel as set up in the Axis Setup display for that axis. The **Run it** button is there as an out for experienced users who know their system, there is no guarantee that the system will perform and will not trip out and/or cause damage to itself. The "slower" option is not available on a VTR system.

8.1.3. Preroll and Postroll Limits Checking

If a move requires a Preroll or a Postroll, Flair will check the movements of these to ensure that they remain within the confines of the limits of each axis. If any axis is deemed to exceed its travel during a postulated Preroll or Postroll, a message will appear on screen to that effect stating "Modify move or slow it down as Preroll/ Postroll exceeds Max/Min limit of travel." In this case the name of the offending axis will be on the title of the message box. The handling for such a situation is to slow the move or modify it so that the axis does not Preroll or Postroll so violently. An alternative is to increase the Preroll or Postroll factor in the Axis Setup for that axis.

To be done

None of the above limits checking relieves the operator of his responsibility for operating the rig in a manner that is safe. All high speed moves should be checked at slow speed and gradually speeded up rather than run at full speed immediately, and as stated above in the safety section of this manual; someone should be ready at all times that the system is under power to press a stop button or hit the red emergency stop. The Preroll and Postroll extends of the move can be viewed and checked using the **Goto** → **Preroll/Postroll Extent** on the top menu bar.

If a significant preroll or postroll is needed (> ½ second), then Flair will put up a message box alerting you to the fact that the preroll or postroll is required. If this is not expected, then you should very carefully check the motion of the axes and the fairings used to determine which axis is causing this and ensure that the preroll or postroll motion will not cause issues. Also note that the "Jobtype" settings of "Start:" and "Stop:" also will affect the motion of the preroll and postroll, and these should be fully understood.

8.1.4. Shooting

When the move starts, the axes move along their paths and the frame count changes through the move. To indicate the progress of the move, the Control Box Slider Bar moves showing the current frame count. If you are shooting the job forwards, it goes from left to right, and if you are shooting the job backwards, it goes for right to left, and the frame count you are actually at is shown just above the block.

8.1.5. Gotos

A Goto will occur whenever the motors need to get from where they are to a specific position. The difference between a move and a Goto is simply that a Goto is a move from wherever the motors are to where they need to get to. It does not follow a path, it is a straight line move very like a simple 2 point move. Gotos move in straight lines as far as each axis is concerned, and takes no account of sets or other obstacles that might be in the way. A goto will occur whenever you run a move, and any axis is not in the correct start position for that move, whenever you ask the system to move to a certain point or frame in the move, and whenever the system moves to zero. The length of the goto is determined by the slowest axis, and all axes take the same time to move during a goto. The fastest an axis will move during a goto is at its maximum velocity reduced by the Goto Factor which is set in the Axis Setups. If axes are moving too fast or too slow, the usual handling is to adjust the Goto Factor for the relevant axis.

Whenever a Goto is required, a message will appear in the top-right corner which indicates the length of the Goto. If you expect a small short goto, and this box indicates a long one, check what the rig is going to do and make sure it is safe BEFORE pressing the “Goto” button. The position that each axis is going to and the time that each axis needs can be displayed using the **Goto → Goto Information** option on the top menu bar. As mentioned above, all axes take the same time for a Goto, but this display will show which axis is taking the longest and thus setting the length of the Goto. Browsing Gotos is a safe option and a smart way to go in most cases as it allows you to move through the Goto under hand held box control and see before anything untoward happens.

8.1.6. More About Prerolls and Postrolls

A preroll occurs when an axis needs to be moving at the start of a move. Most moves start with all axes at rest, but in some cases, the axis will need to be moving right at the start of the move. In this case, the axis will move away from its start position and will then run up to speed so that it reaches the required speed right at the beginning of the move. This occurs after the **Shoot** button has been pressed, and the system status will change to “Prerolling” while this is occurring. A Postroll occurs at the end of the move or when any move is stopped if an axis is moving. It is very similar to a preroll but is done to slow the axis down, and then return it to its correct position. Axes will preroll and postroll differing amounts depending on their speed, their maximum acceleration and their preroll and postroll factors. (See Axis Setups for data on how to adjust these). Note that moves that start or stop very fast may produce tiny prerolls and postrolls. This is a by product of the preroll and postroll system and is not an error. However it may produce a variable timing issue in the triggering of the move. The best handling it to add a short hold at the beginning or the end.

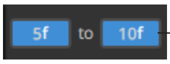
Preroll and Postroll paths

The movement of the rig during a preroll and a postroll is the combination of the motions of all the axes. Depending on how long the axes take to pre- and postroll, this can vary wildly from the actual move path. There is an option in the Job type display that allows you to either make the pre- and postrolls follow the move path as much as possible, or pre and post roll as quickly as possible. Depending on the shot and the constraints of the set, you may want to change these. See the section on Job type display for more data.

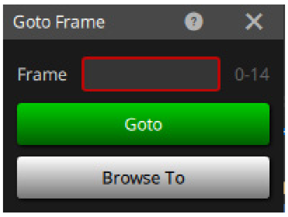
Part Runs

A part run is a shortened version of a move. In a part run, you can state the frame you want the move to start at and the end frame for it to end at. What will happen is that the section of the move that you have specified will be executed exactly as if it were part of the whole move. If you have specified a start frame where the axis is moving, then it will pre-roll in order to get up to the required speed before the move itself starts. Likewise if you have specified an end frame where the axis is moving, it will post-roll at the end of the move in order to stop the axis and return it to the position of the end frame that you have stated. A part run can be from any frame in the move to any other frame in the move. It can also be from one frame to the same frame in which case only the goto will be executed, and no move itself will be shot. Part runs can also be backwards (for example, frame 245 to frame 122). Ensure that the move is safe and will not exceed the limits of travel of the rig, and that the motors will not go too fast.

If you want t run a goto from the current frame to a specific frame, use the **Goto –> Goto Frame**.



Part run using Control box



8.1.7. Interrupting a move

If the move is shooting and the mouse stop button is pressed, or the "STOP" button on the Control Box is pressed, the move will stop, the axes will post-roll, and then will return to the end of the last frame successfully completed. If this happens part way through a frame at single framing speeds, the frame will be completed, and then the rig will stop. Once the system has stopped, it (optionally – see Miscellaneous Setup) assumes that you want to continue to shoot the rest of the move, so it recalculates the move to start where it was stopped, and goes into the "Ready to Shoot" status. The 2 takes can then be matched up and treated as a complete shot. (You may have to cut the camera ramps off when editing film). This feature is not provided for on the Stop button on the Hand Held Box. If the stop button is pressed again while the system is stopping, it will stop all the axes with maximum acceleration - this is deemed to be an emergency of some kind.

8.2. Stepping

Stepping is a means of shooting a move, section by section, so that animation can be done, or so that special effects can be created or so that a move which is too fast to occur at real time and be put together from a series of still exposures. Stepping can occur in many different forms and is a very versatile feature of this program. Stepping is done by pressing the forward step (⏭) and backward step (⏮) on the Control Box, and can only be done when the system is in the “Ready to Shoot” or “Ready to Step” mode. If you shoot with the camera at higher than its single framing speed, you will also expose ramp frames at the beginning and end of every take. Stepping should only be done at or below single framing speed - this speed depends on your camera, but for most, the make/break point is about 4 FPS.

8.2.1. Frames Per Take

The first thing to note about stepping is that you can select how many frames of film are exposed with each step. This is often just 1, and that is the default value, but in some cases, single frame animation is very slow, and a good result can be achieved with shooting doubles or even triples. The number of Frames Per Take is set in the Film Effects dialog box. Enter the number of frames you want each step to be in the “Frames Per Take” box, and “Apply”.

8.2.2. Frames Per Step

You can also vary the number of frames the system advances through the move with every step. This would usually be set to match the above figure, but not necessarily. The number of Frames Per Step is set in the Film Effects dialog box. Select **Effects–> Film Effects**, enter the number of frames you want each step to advance in the “Frames Per Step” box, and “Apply”.

8.2.3. Where does the exposure occur?

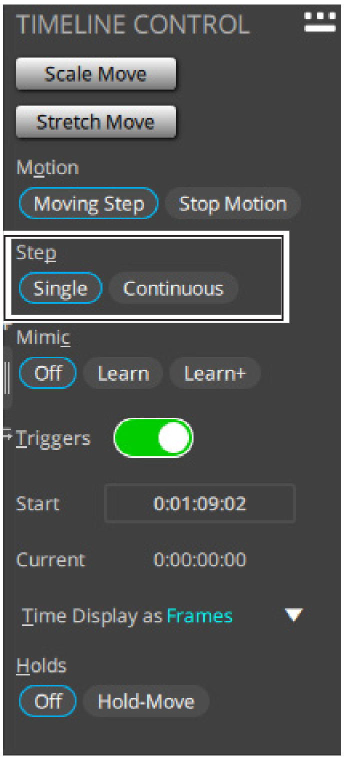
Consider a simple move with two positions lasting one frame. When this is shot, the exposure will occur with a slight blur from when the camera shutter opens (about 1/4 into the move) until it closes (about 3/4 through the move). In effect you are recording the centre area between the first position and the second position. When you are shooting steps in Stop Motion, you can adjust where the exposure occurs from right at the beginning through the move up to right at the end. This is an extension of the “Take Before Step/Take After Step” feature which exists on the older Mark Roberts Computers. See Chapter 13 Special Effects for more details.

8.2.4. Moving Step

If you are shooting a move that the rig can do at real time but you want to animate something, you are often stuck with using stop motion photography which removes motion blur and can give a stark feel to the resultant film. The handling for this is to shoot the move in Moving Step. This effectively Part Runs every single frame of the move and allows you to animate in-between frames. The rig will expose each frame at the same speed as if it were shooting continuously, and moves shot using moving step can be matched to moves shot continuously.

8.2.5. Continuous Step

If you are not actually animating between each frame, then having to press ▶| for every frame can be very tedious. In order to get round this, there is a facility whereby the computer will take frames one after another automatically until you turn this feature off. This is the Continuous Step facility and is accessed through the Side Menu Bar, there is a toggle button on this bar whose default label is "Single Step", when this is pressed, it changes to “Continuous Step”, and then if you press ▶| or |◀ when the system is ready to shoot, it will start taking steps and will continue until it reaches the end of the job. To turn off continuous stepping without stopping the move, simply toggle this button back to “Single Step”. Pressing the stop button will also set it back to “Single Step”, but may also stop the move.



Start 0:01:09:02

Current 0:00:00:00

8.2.6. FWD and BCK STP

If you are running a move forwards, you will naturally want to step through the move forwards, the reverse is true for running a move backwards; you would naturally want to step backwards through the move. However both buttons are active, and you can change direction within the move as you are stepping; the camera will continue in the same direction for film, and this has no particular use, but the move will go in whichever direction you want. For a video system, the frame can be re-recorded, so a back step when you are running forwards is taken as an attempt to correct a mistake, and when followed by a forward step, will retake any frame(s) which were mis-exposed, and then you can carry on. (Single Frame VTR control is only available with PAL systems, not NTSC)

Shooting and Stepping

Since you can step a move when it is ready to shoot, it is also true that you can shoot the move in this case. If you have an animation sequence to shoot, you can shoot any holds continuously with a part run, then go into stepping, and when the animation is completed, finish the shoot simply by pressing "Shoot" instead of one of the step keys. In fact, you can run the job forwards, letting it run till you reach the animation sequence, then press the mouse stop button, let the rig stop, go straight into single stepping without having to do anything else, and then straight back into continuous shooting when this is finished.

8.3. Browsing a Move

Browsing through a move forwards and backwards at variable speed can be very useful for checking the move out, looking for camera flares and a multitude of other reasons. This feature has been provided for with the "Browse" function. This function is accessible through the "Browse" button on the Control Box. When put into the Browse mode, the rig will effectively execute a normal run of the move starting at the current frame. If a "goto" is required, you will be so prompted, but when it is in the right place, it will go straight into "Browsing" mode. When in this mode, the position of the rig along the move is adjusted through the hand held box, pressing the assigned forward key (or joystick) will accelerate the motion of the rig up to a percentage of the move speed and the move can be browsed backwards by using the reverse motion button (or joystick). (The assigned key is selected in the

Miscellaneous Setup – “Browse Axis”) The rig can be run up to full speed by setting the Hand Held Box speed up to maximum or down to 4% of speed or even slower. Browsing to the end or beginning of the move will not drop you out of browse mode, you can simple press the other key and go the other direction. You get out of browse mode by pressing the stop key. As you browse, the current camera position (to the nearest frame) is displayed on the run control slider and on the Hand Held Box. Note that if you are browsing a very fast track move, it is usually wise to do this at a reduced camera speed to reduce the likelihood of the track tripping out. Also note that you should not browse at speed to the end or start of a move that has zero fairings as the rig will stop very suddenly and may trip out or cause damage.

8.4. Recording a Move

8.4.1. Pre- and Post-Hold

When you are shooting a move, it is often requested to have a hold at the end, or a hold at the beginning. In order to facilitate this, rather than have to note these separately, you can simply enter the length of the required pre-hold and the length of the required post-hold. When the job is shot, the pre-hold is exposed first, then the move occurs, then the post-hold occurs. These holds are only valid if the camera is in Record mode. If you run a move backwards, the post hold will be shot first, and then the pre hold. Access this feature **Effects –> Film Effects** dialog box.

8.4.2. Film System

Once the move is to your liking, you will probably want to record it. To take the camera out of Rehearse mode press the Record (●) button on the Control Box. Before the camera runs you have to engage the camera which is done just like the other axes, simply press the “Frame” button at the top of the numerical display, the camera will engage and zero itself. At the beginning of the move, the camera will be started, and then the move will be shot with the camera running throughout and at the end of the move the camera will stop.

There is an additional means of running a move such as this. If your move has fairings on it, then you can start the camera rolling before the move itself is started. When you are at “Ready to Shoot” and you press the "TURNOVER" button, the camera will start running at the current move speed. This will carry on until the stop button is pressed, or the "SHOOT" button is pressed. If you press "SHOOT", then the move will be started in sync with the camera, and will continue as normal except at the end, the move will stop, and the camera will carry on running until the stop button is pressed.

This is particularly useful in live action situations where the director will call “camera”, and then call “action” a little later, and then after the move has stopped, he will call “cut”. In this way the three actions can be separated out to conform to the way shots are most often done. If you trigger the move in such a situation from an external trigger, the move will start when the camera reaches the next frame, NOT when the trigger is received (Though at live actions speeds, this should be within 1/10th of a second).

This method will also work with an external camera. Once the camera is up to speed, you can press the shoot button. The system will them launch itself into sync with the camera and run until the end at which point, the move should stop, and the camera continue. You are strongly advised to test this before relying on it in a shoot condition.

If the camera is fully set up and ready to shoot, you can start the move with the camera in Rehearse and switch it into record during the move. When this happens, the camera will ramp up to speed and locking into phase with the motion of the axes. The reverse is also true, you can turn the camera off during the shoot. This may be done as often as you like during the actual move (though I can't imagine why you might want to!) as long as both the move and the camera are running forwards. (Do not try this is you are running a variable camera speed move of any kind.)

Chapter 9 **Job Storage**

When a job has been shot, it is quite often useful to be able to store that data onto a disk for later retrieval and re-shoot. In order to do this, a job storage facility has been incorporated into the program. When a job is stored, all the move data along with the vast majority of the system configuration at the time is stored so that when it is read back in, not only is the move retrieved, but also the accompanying setups that will allow it to be repeated exactly. It is also possible to only read in the move data if so desired, this is quicker and avoids resetting any other data that you want to remain the same – this feature is called “Simple Read”.

9.1. Automatic Store

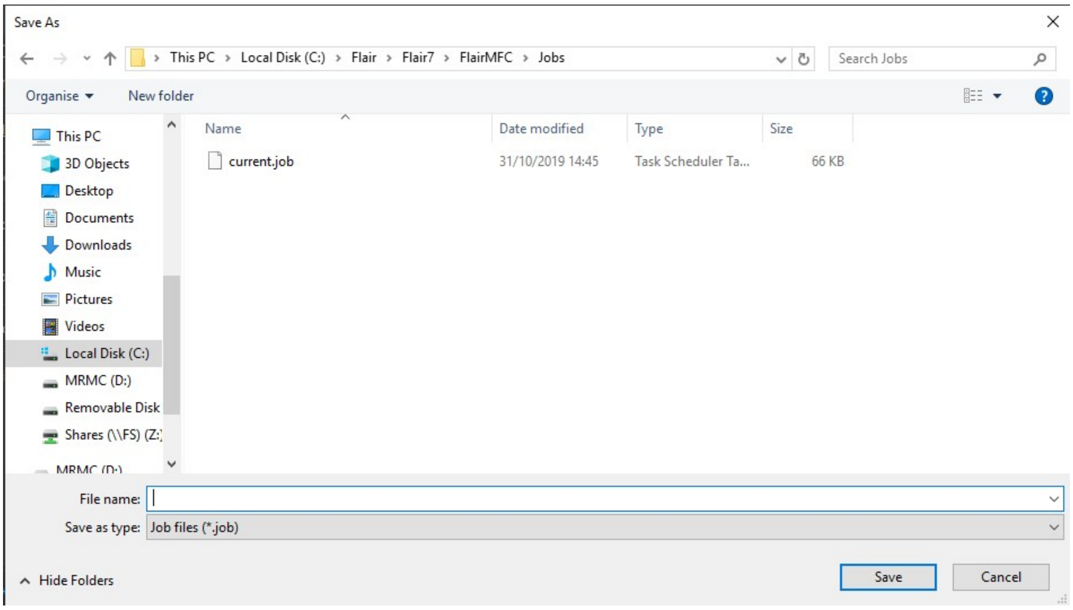
Whenever a job is calculated, it can be set to automatically store to disk and a short beep is emitted by the computer to indicate that this has happened. The job is stored under the name of “current.job” and can be easily read in. The advantage of this is that if there is a power failure or other unexpected interruption, the data in the computer is not lost and the move can easily be recovered. This is called the “Current Job”. This feature can be disabled for speed if so desired in the Miscellaneous Setup display.

9.1.1. Crash Store

If there is a system error with the axis boards and for some reason, the communications between the axis boards fail, it is possible that the program will continue operating. In this case, a dialog box is displayed with the option of saving the job, and exiting the program. Since the only solution at this time (if the fault does not self-correct) is to re-load, you can store the existing job to disk for later recovery. This job is called the “Crashed Job”. To recover it, read it in like any other saved job.

9.1.2. Job Store Dialog Box

The Job Store dialog box appears whenever you attempt to read, or write a job. The box is basically the same for both, one being an “Open” File dialog box, the other being a “Save” File dialog box. These boxes will automatically open in the “File Details” mode to make it easier to sort and select jobs.



Name	Date modified	Type	Size
current.job	31/10/2019 14:45	Task Scheduler Ta...	66 KB

File name:

Save as type: Job files (*.job)

Since this is a standard Microsoft Windows dialog box, you can also delete jobs and create new directories in the usual manner, however it is a single file selection box and only one file can be selected, moved or deleted at a time. When you save a job to disk, if you have selected “Auto Inc Files” in the Miscellaneous setups, the computer will automatically add a number to the end of the last file name, or increase the value if there is already a number there. This can make life much easier when storing multiple versions of a job, for example when you are shooting multiple takes of a move, and there is a slight variation between each.

9.1.3. Single Click Job Storage

It is a smart idea to save versions of the job as you go so that you can easily undo any changes that are not desired. It is also vital to save any mimic moves as they are learnt so that you can replay any good takes.

9.1.4. Temp Save/Temp Open

The two buttons in Job menu allow you to save a job and retrieve it with a single click without changing the current job name. The job is saved in the current job folder as “Temp.job” and this feature allows you to quickly save a job, try out a change and then go back to the previous job easily if that change does not achieve the desired result.

9.1.5. Merge Job

Merge Job is an advanced feature which allows two moves to be merged together. With a prior move saved to disk with the same configuration as the existing move, you can merge selected axes into the existing move. When you merge the move you will be presented with a selection of the axes you wish to bring in from the move you are merging. Any existing motion or positions on these axes will be totally removed and replaced with the data from the merged move. If the merging waypoints are at different frame counts, then new waypoints will be interspersed as needed to allow the merged motion to integrate with the existing move. If you have mimic motion on the axes, this will also be merged BUT the lengths of the existing and merged move MUST be the same or the motion will be scaled to fit the longer move.

Cartesian Merge: If you are in Cartesian Priority, then you may also be able to merge cart data either in positions or mimic.

Note that as with appending moves, it should ONLY be done between moves that have identical configurations. The configurations of the merged move will be ignored and this may cause the data of the merged move to be invalid. For example if you have changed the scaling on an axis, then it will possibly make the motion of the merged move different than the initial.

Possible Uses: The primary use of “Merge Job” would probably be to import an already planned motion on the master axes or the Focus and Zoom and possible Model Movers to a newly imported CGI move. In this case if the CGI move has changed, but you have already worked out the model mover and the focus, you can re-import the CGI data and then Merge the Focus, master axes and model movers into it.

9.1.6. Append Move

A move that has already been saved to disk can be “Appended” to the current move using the “Append Move” function. In this case, only the positions in the move are read in and are added at the end of the list of positions on screen. Since there is no way to easily assess how many frames need to be added from the end of the original move to the start of the appended move, the user have to locate this point and ensure that enough frames are allowed to make this transition. Typically the appended move will start at frame 0, and this will cause the frame counts to be invalid. The usual handling is to use the “ripplethrough add” function and type the needed additional frames (for example 200+) on top of the start of the appended move. Note that since only the positions are read in, the appended move must be set with the same configurations as the existing move. Appending moves with different configurations will produce undetermined results.

Job Storage Progress Bar

Writing a large job to disk can sometime take a while, especially if you have mimic data recorded with the move. If a job is of 400 waypoints or more, a box may appear when you read or write a job that lets you know that job storage is progressing. This box has a small bar that fills up to let you know it is progressing, and a cancel button if you change your mind and want to stop writing or reading the job. If this button is pressed, the current job is cleared out if you were reading a job, and if you were writing a job, then that name is deleted from the system. You are advised not to use this button unless you are sure you want to.

Exporting Jobs

Since the jobs are stored as simple files, it is very easy to copy jobs out of your computer and onto a flash drive or external media. In this way old jobs can be removed from the computer and stored elsewhere if needed, or even sent to MRMC for investigation if there is any problem with a move. In fact the easiest way for us to check out suspected faults is to send us the job on a disk or via email. In order to do this it is necessary to have some familiarity with the operating system being used on your computer. Do not attempt to edit the job file as it is not in any recognisable format.

Reading in Jobs from Different Configurations/Different Versions

If you try to read in a job for a system that is configured differently, with changes in the configuration file, Flair will attempt to merge the data from that job into the current system. If the differences are significant, then unexpected results may occur.

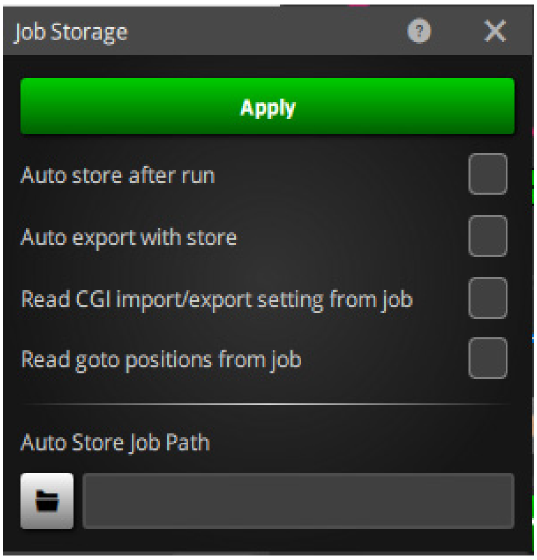
If you plan on doing this, check with MRMC if possible or try reading in the system before doing it at a critical time. It is possible the system may become unstable. Similarly if you try to read in an older job – for example from version 5.4, Flair will attempt to merge the data from that job file into the current system. There are some areas that cannot be merged and it is recommended that you test this and if you only need the move data, do a “Simple Read”.

Storing a job as an older version

You can also export a job from version 6.0 to version 5.4 and again Flair will output the data in as compatible format as possible. It is also recommended that if you read this into an earlier version of Flair that you do a “Simple Read if at all possible.

9.2. Job Storage Settings

Some details of how the data in the job storage file is handled can be affected by the Job Storage dialog box



Auto store after run: When checked the job will be automatically stored after it has been run in Record mode. The file name is incremented automatically and it is store in the “Auto Store Job Path” directory set in this dialog. If your directory is invalid then this function will fail.

Auto export with store: You can also automatically export the move as a CGI file when it is stored. The current settings for the CG Export will be used, and the file will be stored in the Auto Store Directory and will be given the same name with an appropriate file extension (.xyz or .fbx).

Read CGI import/export setting from job: The current settings on the CGI Import and Export are saved with a job and this checkbox selects whether they are restored when you read a job in. Sometime you do want this and sometimes you don’t. If you are reading a job to export the data again, then you would want to check this box. If you are exporting a series of jobs and have set up the export, you would NOT want those settings changed every time you read the next job in.

Read goto positions from job: These goto positions are stored with the job, and this setting selects whether they are restored from the job when it is read in.

Read handheld box setting from job: The settings for the HHB can also be restored from the job if so desired. By default they are not.

Auto Store Job Path: Path to where the Auto Store and Auto Export files are saved.

9.2.1. Errors

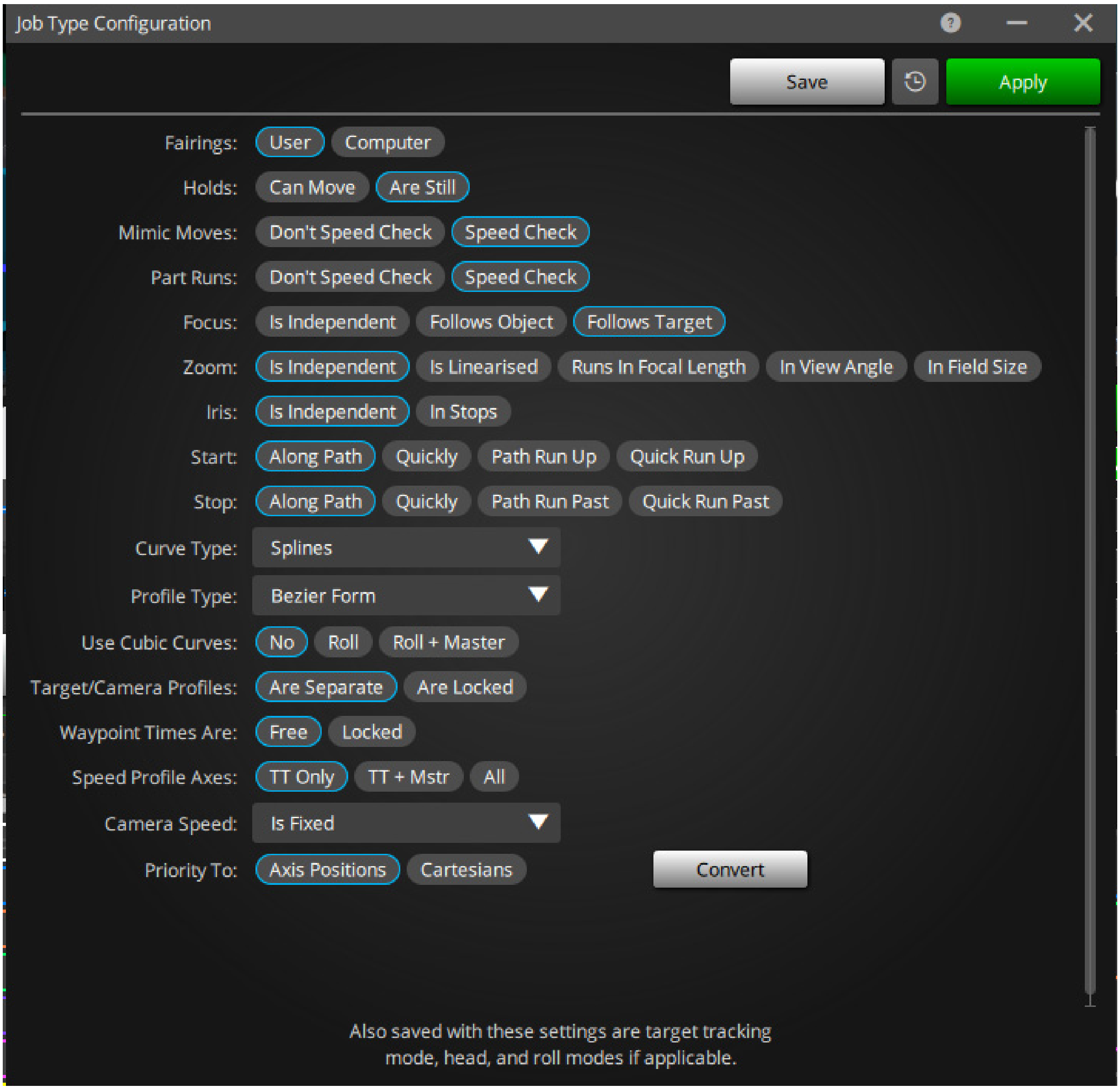
There are quite a few possible errors that can occur during job reading or writing, most of which are self-explanatory and usually boil down to an error in naming a file or some sort of typographical error. There are some system errors that can occur, and if these do occur, note them down and inform Mark Roberts Motion Control soonest for advice. If the reading or writing is aborted, you can always retry, making sure that the computer is not in a run state such as “Ready to Goto” etc. When you read or write jobs the system should be in the normal idle state when it will display the normal rest state message such as “Flair Motion Control”.

9.3. Job Type Configuration

The Job Type Configuration (**Job** → **Configuration**) display is composed of various switches which affect the way in which the system operates, and how jobs are run. Each switch has a name on the left, and on the right are the states or a pull down combination box that allows you to change the state of that switch.

9.3.1. Save, Apply & Exit

Like many of the dialog boxes, changes you make are not automatically accepted. You can check that the changes are to your liking before you **Apply**, or you can throw away the changes just by pressing **Exit**. If you want to set default values for the toggles so that every time you turn the system on they start in a certain state, simply select the desired states and use the **Save** button to store these to disk. This will ALSO save the state of some of the toggles on the side menu bar such as the target tracking state, the Roll control state, and Head Control state.



Fairings: [✓ User] Computer

Holds: Can Move [✓ Are Still]

Mimic Moves: Don't Speed Check [✓ Speed Check]

Part Runs: Don't Speed Check [✓ Speed Check]

Focus: Is Independent Follows Object [✓ Follows Target]

Zoom: [✓ Is Independent] Is Linearised Runs In Focal Length In View Angle In Field Size

Iris: [✓ Is Independent] In Stops

Start: [✓ Along Path] Quickly Path Run Up Quick Run Up

Stop: [✓ Along Path] Quickly Path Run Past Quick Run Past

Curve Type: Splines

Profile Type: Bezier Form

Use Cubic Curves: [✓ No] Roll Roll + Master

Target/Camera Profiles: [✓ Are Separate] Are Locked

Waypoint Times Are: [✓ Free] Locked

Speed Profile Axes: [✓ TT Only] TT + Mstr All

Camera Speed: Is Fixed

Priority To: [✓ Axis Positions] Cartesians

Also saved with these settings are target tracking mode, head, and roll modes if applicable.

Fairings: User/Computer

When a two-point move is being run, the user can allow the computer to generate its own fairing for the move, or the user can define his own. See Chapter 9 - Move Entry for further details about fairings. The computer-generated fairing causes the motor to speed up for half the move, and slow down for the second half. Computer-generated fairings are the equivalent of setting the user fairings to 50%.

Holds: Are Still/Can Move

When 2 identical positions are followed or preceded by a different one, it is possible to draw a curved path through all three points, and this may be what is desired. Usually, 2 identical positions indicate a hold, and this switch will cause a hold to occur between adjacent identical positions as opposed to making a curve between them.

Mimic Moves Speed Check

Specifies whether to apply speed check while the rig is being moved through a mimic controller. It is possible to exceed the stated maximum speed of an axis using mimic control by up to 20% and if the axis ran under mimic learn, it should be able to re-play under mimic play. In this case you can tell the software not to speed check and throw a warning when it finds that an axis has slightly exceeded its stated maximum speed.

Part Runs Speed Check

Specifies whether to apply speed check while a Part Run is being executed. As a standard, a Part run allows you to run a section of the move that is valid when maybe the entire move is not valid. This can happen with a scaled move or an offset move where the original move was able to be run but the modified move is not able to run in its entirety. When this switch is set on, then the section you are trying to run is checked. Use Part Runs with caution.

Focus: Is Independent/Follows Object/Follows Target /Follows Sensor

The focus axis can be driven as a normal axis running between entered points, and when this switch is in the “Independent” state, that is what happens. There are three other states in which the focus is driven to hold focus on a target which moves. In this way, target distances can be measured, and the focus will automatically maintain a sharp image of the target (or a soft one if you want!) These three modes are “Follows Object”, “Follows Target” and “Follows Sensor”. When in “Follows Object”, the distances entered in the focus column are taken as distances to an object, a curve is fitted through these, and then as the move is run, the focus is moved to maintain focus on the object as it moves. “Follows Target” is only valid when you are in Target Tracking, and the focus is driven to maintain focus on the Target as worked out from the target tracking maths. “Follows Sensor” uses the RealSense camera to determine the object distance and sets the focus axis to that distance while running a move. “Follows Object” can be used in target tracking, and would be particularly applicable if you wanted to target track one particular object, but focus on something else. When you are in “Follows Target”, the focus column holds a copy of the Target column, and this will be displayed in white. When you are in “Follows Object”, the distances in the focus column are focus distances and are displayed in yellow. The “Convert” button to the right of this entry will prompt you if you want to copy distances from the entered Target Positions into the Follow Focus distances which could be a good starting point if you are switching from “Follows Target” to “Follows Object”. These states can also be selected from the axis context menu which is brought up by clicking on the focus axis name in Numeric display.

Zoom: Is Independent/Is Linearised/Runs In Focal Length/In View Angle/In Field Size

Tracking in on an object and zooming in both produce a change in the field size at the object you are tracking in on. However, while a constant speed track in will produce a constant change of field size, this is not true of most zoom lenses. If you want to create a smooth zoom that looks like a track (ignoring perspective changes), you have to linearise the motion of the zoom. The actual zoom linearisation is done through the Lens Setup which is described in Chapter 5 Setups, however once this Setup is done, you can use this data in several ways. If you are in “Zoom is Linearised”, then the computer will drive the zoom axis to try to achieve a consistent change in field size. You can also do this and display the data as the focal length of the lens or optionally as the view angle. Note that the view angle may not be what you would normally consider to be the view angle as that depends on many factors such as the format and whether you consider view angle as horizontal, vertical or diagonal. The number displayed here is merely meant to be indicative of size. Contact MRMC for more detail. In the Focal Length and View Angle modes, you can directly enter focal lengths or view angles as numbers, and the reported current position of the zoom axis will reflect the current focal length in millimetres or the current view angle in degrees. When you are in “Focal Length” then the positions column for the zoom axis is shown in blue, if you are in “View Angle” then it is shown in green. These states may also be selected from the Axis Context Menu which is brought up by clicking on the zoom axis name.

Iris - Is Independent/ In Stops

When the system has an iris axis that is fully configured to drive the iris ring, the iris axis can be programmed in stops rather than in axis positions. This means that you can program a stop change by typing in the stop numbers (for example, 5.6 to 11). This can be useful for changing light conditions and also is used when you are using the system's Variable Camera Speed option. When the iris is in independent mode, then it acts just like a normal axis. When in the “stops” mode the current position of the iris axis will display the current stop on the lens and will be in red. These states can also be selected from the axis context menu which is brought up by clicking on the iris axis name.

Start - Quickly/ Path Run Up/ Quick Run Up/ Along Path

This toggle applies to Preroll as covered in Chapter 8 Running a Move. When the system has to Preroll in order to get up to a specific velocity to start a move or section of a move, each axis works out how long it needs for this Preroll, and the actual Preroll takes as long as the longest axis with each axis starting to move at their own time, this can result in the rig moving in an unusual way before the move actually starts. In extreme cases, this can lead to the position of the rig leaving the desired move path sufficiently to be potentially hazardous in a tight set. In this case, when “Start along path” is selected, all axes Preroll together and the preroll path will approximate the move path much more closely. It is highly advised that you experiment with these switches and their effects BEFORE you use them on set. Whilst “Start along path” might be gentler and closer to the path, it will also move further.

The “Run Up” modes also have the “Quick” and “Along Path” options, but these modes will drive the rig to a calculated pre-roll position before it goes into “Ready To Shoot”. This generally requires a little more distance for the run up, but is more comprehensible for most people as they do not understand why it starts going in the wrong direction in the normal preroll mode. Also note that the “Ready to Shoot” position is NOT the same as the start frame of the move when this form is being used and the move starts at speed. As of version 6.2 the “Run Up” modes do NOT work with Roll Level or Roll Relative and you should use the standard modes.

Stop - Quickly/ Path Run Past/ Quick Run Past/ Along Path

As above, but this relates to Postrolls. When a move is stopped either by the user or when it comes to the end of a move, the rig if still moving will go into a Postroll. As with Prerolls, the action will last as long as the greatest time needed by an axis to Postroll, and the other axes will stop in their own time during the Postroll when this toggle is in the “Stop quickly” state. When in “Stop along path” all axes will take the same time to slow down and the resultant movement will more closely follow the move path. If however while this is happening the user presses a stop button again, all the axes will immediately stop as quickly as they can at full deceleration. The “Run Past” modes also have the “Quick” and “Along Path” options, but these modes will run the rig to a calculated post-roll position at the end of the move. This generally requires a little more distance for the slow down, but is more comprehensible for most people as they do not understand why it reverses direction in the normal postroll mode. Also note that the end position is NOT the same as the end frame of the move when this form is being used and the move ends at speed. This setting also affects how the rig stops when you use the middle mouse button to stop it while it is moving on a Goto or being jogged. As of version 6.2 the “Run Past” modes do NOT work with Roll Level or Roll Relative and you should use the standard modes.

Curve Type: Splines/ Beziers/ Linears/ Cubics/ Quadratics/ Quintics/ Quik Cubics/ Bounces/ FSplines/ Quik Quintics/ Ramps

There are numerous ways of fitting a curve to a series of points. In general you should only use Splines, Beziers or Quintics. The other methods are pretty developmental and may cause unexpected or undesirable results:

When a curve is generated by the software it normally uses a curve form called a spline. (A spline originally meant a flexible strip of wood that you could bend to form a smooth curve) This form is used as it has a smooth shape and can produce movements that have a lessened incidence of causing the rig or table to shake due to rapid changes in acceleration.

Curve reversal detection and handling is currently only valid in **Splines**.

You can alternatively use **Bezier** Curves when in target tracking, these curves have the advantage that you can change the shape of the curve without changing the actual plotted points because each Bezier point (except the first and last) has 2 control points that affect the way the curve arrives at and leaves each waypoint.

Linears make straight lines between each point, but this will not work well as it makes for huge accelerations at the points.

Cubics work quite well except they tend to use higher accelerations at the waypoints. This may be desirable. If you find that splines have too “floaty” a feel, then cubics may work for you.

Quadratics do not work and will cause the system to crash.

Quintics are a fair approximation to the splines, but have a quicker calculation time with multi point moves, and in fact your software will switch you into Quintics if you enter more than about 40 points.

Qwik Cubics are experimental and work much like Quintics with the exception that you are provided with a toggle control in the graphics setup with which you can change the slope of the graph at the waypoints.

Bounces will produce almost linear moves, but will use the last few frames to abruptly change direction – set the number of frames through the Fairings Setup.

FSplines will force the system to use splines even though you have more than 40 or so points. Be warned that as you add points, the calculation time increases exponentially and you will run into the communication to the root box timing out. If this occurs, you can increase the timeout.

Qwik Quintics: Experimental (do not use).

Ramps: Experimental (do not use).

Profile Type: Bezier Form/ Cubic Form/ Fairings Form

Speed profiling is described more fully in Chapter 10 Target Tracking, however the basic principle is that you can modify the speed along a curve without changing the actual path of the curve. This toggle allows you to select 3 different ways of doing this.

Use Cubic Curve: No/Roll/Roll + Master

Cubic curves are easily understood and communicated to 3D software. If you are importing or exporting curve forms with 3D packages, you should set this flag depending on which axes you are importing/exporting. Note that this is only applicable to “curve” data, not “raw” data (a point per frame), and that cubic curves tend to accelerate/decelerate more abruptly than the normal system curves. The master axes can also be imported and optionally may also follow cubic curves.

Target & Camera Profiles: Are Separate/ Are Locked

When Bezier curves are used as per above, then the way in which the path is traversed is totally independent of the times entered on the waypoints except the first and last times. This means that you can control the speed that the camera or the target moves along its plotted path. This feature can be used to create effects that normal splines will not allow you to. Since the camera and the target are 2 totally separate paths, you can control them independently, or you can have the speed profile of the target locked to that of the camera, so that as the camera speeds up along the path so does the target.

This feature is also provided for in Speed Profile in Graphs display and can be used after the move has already been plotted. See Speeding Profiling on page 183 for more details.

Waypoint times are: Free/ Locked

When you are running target tracking move in Beziers, the timings of the waypoints are essentially ignored for the target and camera move. If you explicitly want the waypoints to be passed through at the times you have stated, then this flag should be set to “Locked” before you start programming the move. Should you have already programmed the move, then the waypoint times can be locked afterwards. See the Chapter 10 Target Tracking for more details.

Speed Profile Axes: TT Only/ TT + Mstr/All

This feature allows you some more flexibility in speed profiling. Normally when you speed profile a move, the Cartesian data is speed profiled, but not the master axes. You might find that this could lead to some problems with the master axes lagging behind or too far ahead of the rig so that the move will not run. Profiling the master axes means that the time factor that is being passed to them is the same as the one being used to profile the carts. You can also use this profiling to vary the speed for the other independent axes such as model movers etc. This mode will only really work when you are using normal curves and not Bezier curves.

Camera Speed: Is Fixed/ Varies/ ... / ...

Turns on and off the variable camera speed functions and sets the exposure compensation option. You can also select to have the move being at fixed speed, but the camera speed to vary. You of course could not shoot repeatable passes at different speeds, but this may be a desirable option.

See Chapter 14 Variable Camera Speed for a full description of this feature.

Priority to: Axis positions/ Cartesians

A target tracking move can be represented as axis positions or as Camera and Target Cartesian positions. When you start editing one of these, the 2 representations of the move do not match exactly, and this toggle allows you to select which one has priority. When you are in “Priority to Axis Positions” the

Cartesian positions are modified whenever the axis positions are changed. When you are in “Priority to Cartesian” the move is based upon the Cartesian positions which are set when the flag is switched, and can only be edited directly, not through adjusting the axis positions. See the Chapter 10 Target Tracking for more details. To the right of this button is a “Convert” button which allows you to convert a Cartesian Move back into an axis move. See Reverse Kinematics and Cartesian/Axis priority on page 180 for more details.

Note

If there are any other buttons in your job type box that are not listed here, then you have had an update or upgrade in the software beyond this revision of the manual, in that case, check in the back of the manual for any addenda or contact Mark Roberts Motion Control for additional data.

9.3.2. Notional Axis Control

Notional Axis control allows the user to control a function of the motion control without having to directly control the position of a motor. Take for example the Focus. Once calibrated, you will most likely want to know what distance the lens is focussed on and not worry about where the motor is. This also applies to other lens motors such as zoom and fstop. You can set up lens so that the zoom for example reads out in focal length and not just motor position. This also works when moves are run. The motion of a focus motor between 30 feet and 2 feet is not linear, and the user would not want to have to program points to define that, but the system can pull the notional focus distance from 30 feet to 2 feet and move the focus motor accordingly when the focus is run in “Focus Follows Object” mode.

This idea covers several areas of control, but includes the position of the head on a complex rig such as a Milo and also the position of the roll motor (and others) which may be needed to keep the horizon level or even control the angle to the horizon which may not relate directly to the position of the roll motor.

The states of these flags as originally in the Job type display and they are included here at the end of this chapter. However the method for turning on and changing these modes has mostly been added to the side control menu or for a specific axis to that axis’s context menu (left-click axis name in Numeric display).

Focus - is Independent/ Follows Object/ Follows Target/ Follows Sensor

The focus axis can be driven as a normal axis or as notional axis. Refer to Focus: Is Independent/Follows Object/Follows Target /Follows Sensor on page 165 for more detail.

Zoom - is Independent/ Linearised/ In Focal Length/In View Angle/ In Field Size

Zoom axis is a notional axis in that it allows you to change axis control without changing position. Refer to Zoom: Is Independent/Is Linearised/Runs In Focal Length/In View Angle/In Field Size on page 165 for more detail.

Iris - is Independent/ In Stops

For more detail about changing Iris axis notional options, refer to Iris - Is Independent/ In Stops on page 166.

Lift Current Position as units/degrees

Toggles the display for the lift between Degrees and Units; This may also be selected from the Axis Context Menu which is brought up by clicking on the lift axis name. This is not a true notional axis control as it does NOT affect how the axis moves, but is merely a convenience feature for users who want to know the angle of the arm.

Roll: Independent/ Level/ Relative/ Up

With the "Angle" axis swinging the head over the top of the arm, if you are panned to one side, this will result in a roll motion. This roll motion may not be desirable, so the computer can compensate for this action to keep the horizon level ("Roll Level"). This of course is fine unless you want to be able to add some roll motion to the move. In this case, the selection "Roll Relative" allows you to program roll motion that is relative to the horizon rather than being relative to the zero position of the roll axis; so regardless of action that the roll motor is doing to compensate for the angle axis, the roll will also move clockwise and anticlockwise as you have programmed. Note that in "Head Level" this function is irrelevant and should be left as "Roll Moves". In roll relative, the roll axis positions are displayed in Green to indicate that they are not true axis motor positions. When you have the roll being controlled in this way, you may not zero or home any of the axes that affect its positions (i.e. Lift, Angle, Tilt, Pan), you must first take it out of roll control to do this. If the roll trips out or it turned off when in "Roll Level", then when it is switched back on, it will request a "Goto" to the level position. "Roll Up" is a third option which is only valid in normal Target Tracking moves. It avoids the problem of looking at the floor and ceiling when there is no horizon, it basically assigns a point that defines where "up" is and moves the roll axis to follow that as it moves.

Note that with a normal motion control rig with a camera mounted so that it rolls when the rolls axis is moved, this mode should operate immediately. In specific cases where the camera is NOT mounted in this fashion, the actual horizon position is affected by multiple axes and the horizon control is only in effect when running a move or in cart control. In other words, you can be in "Roll Level" and not have a level image until the move is run or cart control is engaged. Also note that if you change modes after a line is stored or change the value of the roll relative, the rig might not go to the positions listed in the waypoint, but move to keep the nodal point where it was defined to be, but move the motors to achieve the desired horizon position.

(This covers the same topic as covered on the control menu bar.)

Chapter 10 **Target Tracking**

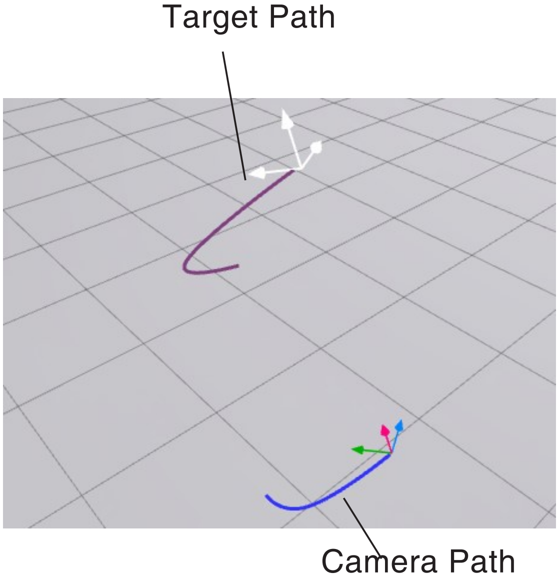
10.1. Principle of Operation

Hold your fist directly in front of your face, and move it away from you in a straight line. As you do this, you can see that your wrist, elbow and shoulder all operate independently to create the desired movement of your hand. You know the path you want your hand to follow, and the different muscles (motors) and joints (axes) are driven to achieve that path. This is the simplicity of target tracking, rather than trying to control individual axes to generate the path you want, you define the path you want, and the axes move to achieve it.

10.1.1. Viewpoint/Camera Path and Target Path

When in target tracking, the computer knows the shape and arrangement of the axes that drive the rig and it can work out exactly where the camera is and from the distance to the target, it can also work out exactly where the target is in three-dimensional space. This data is calculated for each waypoint entered, and a three-dimensional curve is fitted to the camera positions and to the target positions. These curves are referred to as the Target Path and the Camera Path. The exact point that is considered as the camera point or view point is defined relative to the Front Nodal Plane of the lens. For a more detailed description refer to Setting up a Lens on page 38.

When the move is actually executed, the computer moves the axes so that the camera follows its plotted path and always looks at the target as it moves along its path. The advantage of this is that you can make the camera lock onto a target and hold it in frame throughout the move while the camera moves along the path you want. Since the target is also defined as a path, you can lock onto a stationary target or a moving target just as easily - hence the name “Target Tracking”



10.2. Target Tracking Move Entry

Entering a move is very similar to the normal move entry, plot the waypoints you want the move to pass through, and at each waypoint, measure the distance to the target and enter this into the target box for that waypoint. If you have the focus axis in follow target mode, when you store positions, the computer will actually suggest a target distance based on the current position of the focus axis, but it is sometimes more accurate to measure the distance. Once all points and distances are entered, the move can be run just like any other move. Fairings and exponentials can be applied as in simple moves, except that the fairings and exponentials apply to the Camera and Target paths, and not to the axes themselves, and you should look at and adjust the “Cartesian Fairings” and not the axis fairings.

10.2.1. Measuring the Target Distance

Target distances are measured in the units your system has been set up for; this can be inches, feet, centimetres or metres. The measurement is taken from your selected measuring point to the target and is entered into the relevant target box for that waypoint. When you are a long way from the target, it is more accurate to measure the distance. When you get very close (a few focal lengths away), it can be easier to focus accurately on the target, and get the computer to work out how far away it is. The measure point is usually the film plane, but this can be in another location if that is easier for you; see Setting up a Lens on page 38 for more detail on this.

10.2.2. Target Tracking and Independent Axes

A **target tracking axis** is defined as an axis whose position affects either the camera position or the target position or both when in target tracking; an independent axis affects neither. In a target tracking move, independent axes can be run as normal axes, and can have fairings or exponentials of their own applied to them as you wish. A likely application of this is having a motor or motors moving a model mover which is independent from the move, and also the focus axis doesn't actually change the position of the target or viewpoint significantly, and can be run as an independent axis if so desired. In these cases, it is possible that you would want to define a waypoint that was for the model mover only, and did not affect the target tracking move or vice versa. To handle this, you can have target tracking waypoints and non target tracking waypoints. A target tracking waypoint has position for every target tracking axis entered, it may or may not also have positions entered for independent axes. A non target tracking waypoint has no target distance entered (the field is blank) and must have NO positions entered for any target tracking axes, it also may or may not have position entered for independent or master axes.

10.2.3. Target Tracking Axes and Master Axes

There is also a subtle distinction between master axes and target tracking axes. If your rig has more than 5 major axes (usually Track, Rotate, Lift, Pan and Tilt), then more than one axis can be used to produce the same result, (for example, an extending arm can move the camera forwards, just like the track in some cases). When this occurs, one or more axes will be termed “master axes” and will have been set up by MRMC at installation time. A master axis behaves just like an independent axis above, it can have fairings, and can have points on non target tracking waypoints. How this works and how it can be used to assist in move planning will only really be accomplished through experience and familiarity with target tracking and its operation. If you have specific questions, call the Software Department of MRMC.

10.2.4. Up Point: (Roll Up)

Usually, when you are running a move, you want the roll of the camera to keep the horizon level, or perhaps add a little “dutch” in as the camera moves around. This is normally achieved relatively easily with the software through the various roll control modes. You can however get into a problem where the camera looks straight up or straight down. In this case, the horizon is lost and the roll axis will often flip around and trip out. You can manually program the roll in this case, but that is pretty laborious. Another option is to use the “Roll Up” mode of roll control.

This mode is virtually identical to the third node in a 3-Node camera in Maya®, and is used to set a point which should always be “up” in frame, so if the point is below the camera, then the camera has to roll upside down to make that “up”. This mode of operation is particularly useful for situations where the floor is not parallel to the horizon, which can occur with models that are not mounted flat and other effects work. You can also use the scaling feature to “tip” a move of off level, and when in the “Roll Up” mode, this will not affect the shape/look of the move.

An Up Point is created whenever you store a target tracking waypoint, and if you are in “Roll Up” mode, then these points are splined in exactly the same way as the other target tracking paths are splined. One thing to note is that this does NOT necessarily provide a constant angle to the horizon, and you may need to add waypoint to control the roll to make it act more level when you are not looking straight up or down.

In this mode, there is a third path in the 3D Graphics display and more Cartesian graphs for the motions: X Horizon, Y Horizon and Z Horizon in the 2D Cartesian Graphic display. This mode is ALSO valid when you are in Pan/Tilt Target tracking, and the roll is then driven by this point rather than splined. This path when shown on the graphs is Green. It is only valid in normal target tracking and Pan/Tilt Target tracking.

If using the Up Point, do not enter consecutive points with 120 degrees or more of roll between them as the motion of the Up Point will probably cause the camera to roll suddenly in the middle rather than gently throughout, try adding in between points if this happens.

10.2.5. Target Tracking Roll and Roll Relative

The target tracking Roll and Roll Relative are not exactly the same. Roll Relative is totally valid outside of target tracking and when in this mode it is treated like a notionally controlled axis like the focus or zoom when in a controlled mode. Once you get into target tracking, the Roll Relative is still handled in the same way, BUT you must be careful not to get confused between this and the Roll in Pan Tilt Roll target tracking. Currently, the Roll in target tracking is NOT used to run the camera, you still use the Roll Relative values. If you import a move from Computer Graphics, the Roll Relative value is still used to determine the camera’s relative orientation to the horizon.

10.3. Target Tracking Modes

10.3.1. Pan/Tilt

Standard target tracking with the Flair software has always defined positions for the camera and also for the target, and the splining of these points has produced a camera path and a target path. A new version of target tracking has been added which treats the camera position in exactly the same way, but instead of splining the target position, the computer splines the camera orientation in terms of the direction it is looking and the roll of the camera. The two systems are both quite valid and which one you use will depend on the effect you want to create. If you are looking at a stationary target as you move the camera around, you will almost definitely want to use standard target tracking as you would if you had to pull focus on something that was intimately connected with the camera path and target path.

As an example of the two systems, imagine a move in target tracking that was simply a two-point pan from -90 to +90 with a target at 100.00 units away. A standard target tracking move would plot an almost stationery camera with a target path that goes from 100 units away, straight through the camera and out to 100 units the other side. This move will not run as the target cannot pass through the camera. The positions are valid and what the computer is trying to do is expected but may not be the result you wanted. If you enter this in Pan/Tilt target tracking, then the camera motion is exactly the same, but the orientation of the camera in space is driven by a spline from -90 to +90 degrees and so the camera will pan from +90 to -90 through the move which would be much more what the user was expecting.

In many cases, you can get the same results from standard target tracking as you can with Pan/Tilt target tracking. The cases where you might be swayed into using the Pan/Tilt form would be when you are more concerned about the orientation of the camera in space than where exactly it is looking and also where the target would be getting very close to the camera as this can cause some unexpected results in standard target tracking. The best thing is to try these various methods out and find out for yourself what the advantages or each are.

Move entry is very similar to normal target tracking moves, but the Target row is irrelevant and a waypoint is determined to be a target tracking waypoint if there is a position for any target tracking axis, in which case there must be a position for every target tracking axis. Normal target tracking is the default mode for the Flair system and if you have any problems with Pan/Tilt target tracking, try the normal mode as well.

Normally you cannot have the Pan cross the +180 or -180 boundary. In other words, you cannot cross through looking directly backwards with the camera. This is also true of Tilt. If you have a move where this is needed, you can try to use the Miscellaneous Setup fix for this “Cart Angle Fix”. Setting this to “On” should allow these large angle values to be used.

10.3.2. Easy

Easy target tracking is a simplified form of target tracking in which the major axes are all splined just like a normal non target tracking move, but the pan and tilt axes are run to follow the target path. This avoids some of the complexities that you can encounter doing target tracking, but it also negates some of the powerful features only available with full use of target tracking. For example, normal target tracking does not like the rotate axis to change from in front to behind the rig or vice versa, and will usually cause problem with speed limitations. With Easy target tracking, the rotate axis is simply splined and will do whatever the move required of it without running into this problem. However, you cannot scale this move or use other advanced target tracking functions.

10.3.3. Easy Pan/Tilt

As with Easy target tracking, the larger axes are run independently rather than being driven by the target tracking move. With this version, the cameras position is determined by the motion of the larger axes, but the orientation of the camera is driven by curve generated by the entered camera orientations at each waypoint.

10.3.4. Camera Orbital

In a similar way to Pan/Tilt target tracking, Camera Orbital target tracking uses a path and an orientation. With Camera Orbital, the Target is still plotted and controlled as a path as with normal target tracking, but the Camera’s position is defined as a distance from the Target and an orientation relative to it. This might sound a little complex, but in the simplest case where the Target is stationary, you can “orbit” around it as a fixed distance without having to define mid points to pull the camera path into a curve. With every point that is stored, the camera’s distance from the target, and the angle around it as well as the tilt up or down to the target is noted. When the move is run, these values are used to determine the camera’s position relative to the target. So with just two points you can do a perfect arc, or even a spiral. Of course the target can be moving or stationary, but the camera’s position is always worked out relative to the target. With the current version of Camera Orbital, you must make sure to note points that are less than 180 degrees apart or the software will not know which way to go – so a full 360 will need at least one mid-point. Also you can tilt up or down toward the target but you cannot currently go over the top. Previously it has been possible to move around a target in a perfect arc using Cartesian Control, but not to run that as a move. Now with Camera Orbital, you can! You cannot use Bezier Curves with this form of target tracking, however you can edit the graphs of Camera Azimuth and Camera Orientation as well as Camera Distance to effect changes in the move.

10.3.5. Camera Relative

This mode is similar to the normal mode of target tracking except that the Camera is not an independent path in space, it is regarded as relative to the Target. If you edit the move to make the target higher during the move, the camera will automatically be adjusted with the target. The use of the mode is not entirely determined, but in some cases it may provide an easier editing interface.

10.4. Limitations

There are some limitations to what the target tracking can do, and when encountered these can cause unexpected motions to occur.

10.4.1. Tilt 90

Unless you have specifically been provided with software to handle it, the tilt may not go to or through a position where the camera is looking along the axis of pan. If you can move the pan and the image rolls rather than pans, then you are in this situation. This is referred to as Tilt 90 and is a mathematical problem as the exact position of the pan is indeterminate when this occurs. The indicators of this are extremely large frame counts being requested for a move on the pan and/or tilt axes and if you go into the Graphic display, you will see the pan or tilt axis graph going right off the screen. The handling is simply to plot the move in a way that does not encounter this situation. (see note below).

10.4.2. Rotate 90

If you have a fixed length arm there is another mathematical problem that occurs when the arm is at 90 degrees to the track. In this case there is no way for the rig to control the position of the camera in a side to side manner, and in order to overcome this, the track is moved rapidly backwards and forwards over this period. This will almost undoubtedly produce an unacceptable move and can be handled by modifying the waypoints to avoid going to or through 90 degrees. This applies to +90 and well as -90. This problem can be worked around if you have an extending arm. In this case, the axis that drives the arm must be a target tracking axis – this is usually possible only with the Mark Roberts Motion Control Milo or Cyclops; and then the Track would have to be a master axis. Check the state of the “Master Track/ Extend” value in the Kinematics Setup.

> Note
>
> Even if you have plotted points that are not going to these points or through them, they can be encountered when the rig executes the move. If this occurs, just move far enough away from this zone to handle the problem, this would be typically about 5 or 10 degrees away from the 90 degree position.

Also note that if your camera path for any reason goes outside of the allowed side to side range of the rig, the rotate will simply go to 0 or 90, and there may be discontinuities in the Rotate axis’s motion. If you see the rotate either staying at 90 the whole time or dropping to 90, this usually means that you have encountered the problem above.

10.4.3. Tilt 90 Handlings

The most usual way to get around a Tilt 90 problem is to reposition the move or the subject such that you are not running the tilt across -90 degrees. You can run the tilt from positive down to -85, or from negative up to -95, it is crossing the -90 zone that causes problems. The same problem actually occurs at +90, but that is not a position that most cameras can go to. The handling for this condition with a rig that has an Angle axis can be to re-position that Angle axis so that the -90 position is not used during the move. For example, a lift from the floor to maximum height with a Milo rig will cause you to go through Tilt 90 if you are looking straight ahead. However, offset the move to the side by a couple of feet, and as the arm and Angle lift up to level, the pan axis is doing the tilt motion, and the tilt axis itself hardly moves. Sometimes with an Angle axis, you will find that the rig has a move programmed that goes through the -90 zone, but that when you shoot the move, the rig achieves the move without going through -90 and in fact goes to the physical location you specified, but the pan and tilt read different numbers (-180 different). This is not a problem unless you run the move backwards. In this case, the pan and tilt will flip around and come to the same orientation, but the other way round. If this happens, run the rig forwards and store the end position again when the rig reaches it. (See also the section Reverse Kinematics further on in this Chapter).

Tilt 90 Software handlings

There is some software implemented which does something towards handling Tilt 90 conditions. It does not work properly all the time, so it is not recommended, but it may get you out or a tricky situation. If you have set the Tilt 90 Zone in the Kinematics Setup to somewhere between 10 and 20 degrees, and have set the Tilt90 handling in the Miscellaneous Setup to “Handle”, then some form of handling for the Tilt 90 region should be implemented.

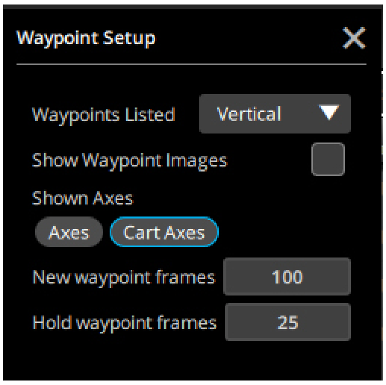
10.5. Cartesians: Numeric and Graphical

When originally written, target tracking only applied to the Camera and Target Position. Subsequent development has added these notional axes to this list: X Camera, Y Camera, Z Camera, X Target, Y Target, Z Target, Pan, Tilt, Roll, X Up, Y Up, Z Up, Lens, Intraocular, Convergence, Target, Camera Azimuth, Camera Elevation and Camera Distance. These axes are virtual or notional more often than not as there is no simple conversion from motor position to virtual position. “Cartesian” was a good name for these axes when the software was first developed, but this has become something of a misnomer, and they should more correctly be called “Virtual” or “Notional”. The first 12 relate directly to the camera position and orientation, the last 3 are somewhat different and refer to the Lens i.e. focal length of view angle and then the Stereoscopic settings for shooting 3D. The “Target” Cartesian is not used, but the Azimuth, Elevation and Distance are using in Camera Orbital target tracking.

10.5.1. Numeric Cartesians

Target tracking positional information is stored in the system in terms of Camera Cartesian positions and Target Cartesian positions. Cartesian, when referring to a point in 3D space means that its exact location is represented by three measurements along the three dimensions. Along the X, Y and Z axes or in layman's terms, height from the floor, distance forwards, and distance to the side. This sort of information may not be directly useful to the user, but it can be displayed and edited in the Waypoint view. When this display is selected, a screen very similar to the normal Waypoint view is shown except that the axes names are not “Track”, “Lift” etc. but “X Camera”, “Y Camera”, “Z Camera” and “X Target”, “Y Target” and “Z Target”, Camera Pan, Camera Tilt and Camera Roll Also X Up, Y Up and Z Up, Lens Intraocular (I/O) and Convergence (there may be other axes to the right).

Switch between Axes and Cart Axes using the cog in the Waypoint view, which opens the Waypoint Setup box.



Waypoint Setup ✕

Waypoints Listed: Vertical ▼

Show Waypoint Images □

Shown Axes

New waypoint frames: 100

Hold waypoint frames: 25

The positions listed below are the plotted XYZ positions of the camera and target for your move. These positions can be manually edited, stored using the store waypoint or axis store button and generally modified in much the same way as you can with the numeric axis positions. The Numeric Cartesians that are valid in standard target tracking are X, Y & Z Camera and X, Y & Z Target. In Pan/Tilt target tracking, they are X, Y & Z Camera and PAN and TILT of Camera. Note this is not necessarily the same as the Pan and Tilt axes and it is important to know and understand the difference. In Roll Up mode, then X Up, Y Up and Z Up are used. In Camera Orbital, the camera’s orientation to the Target is defined as Azimuth, Elevation and Distance, Azimuth and Elevation are used to distinguish it from Pan and Tilt, and these values are valid in that case.

Editing of these values is ONLY effective when you are in Cartesian Priority, otherwise new values will be worked out from the entered axis positions for each waypoint. For more details, refer to Reverse Kinematics and Cartesian/Axis priority on page 180.

Warnings

This display was originally intended to display information to the user should he so desire to see it. You may edit the data, but be aware that you will only be modifying the move when you change these positions if you have set the “Priority to” flag in the Job type dialog box to “Cartesians”. Once this is set, then the target tracking axes positions become irrelevant except that the entered position is used as a starting point to find the position required by the Cartesian data.

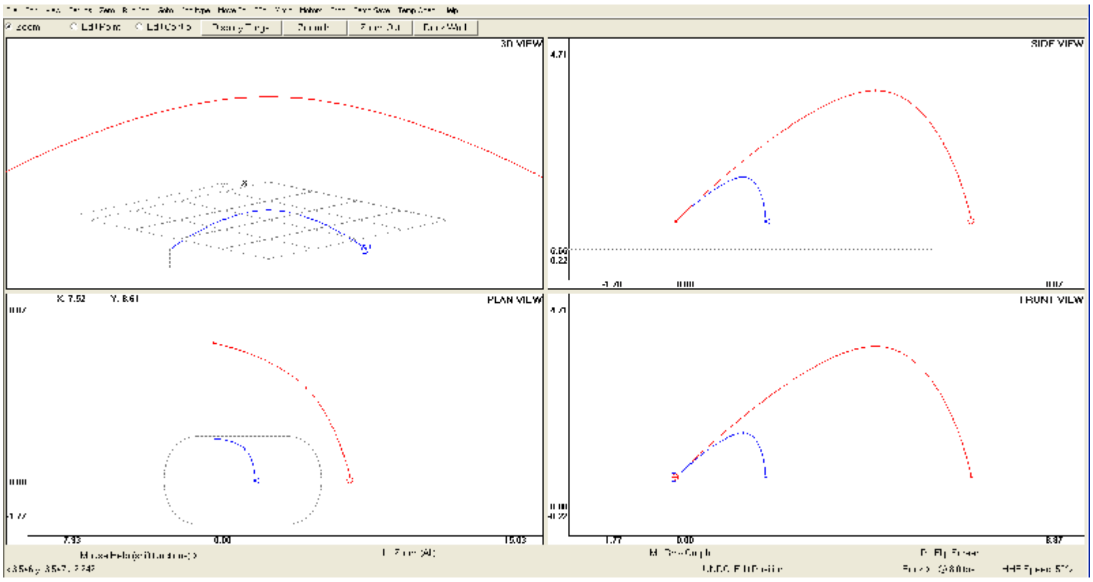
Advantages

You can offset a move by a distance in any axis by simply typing the offset you want into the top of the column that you want to offset and entering the “+” key or “-” key in exactly the same way as you can with axis positions, but in this case, it will make a more logical change. You could lift the entire move up 1.0 metre by adding 1.0 to the Z Cam and 1.0 to the Z Targ rows (And Z Up if in Roll Up mode).

> Note
>
> The Target Column and other axes columns that are visible in this display will allow you to edit the target distances and the positions for those axes. It is not recommended that you do this. The axis enable/disable feature is also not available in this display.

10.6. 3-D Graphs

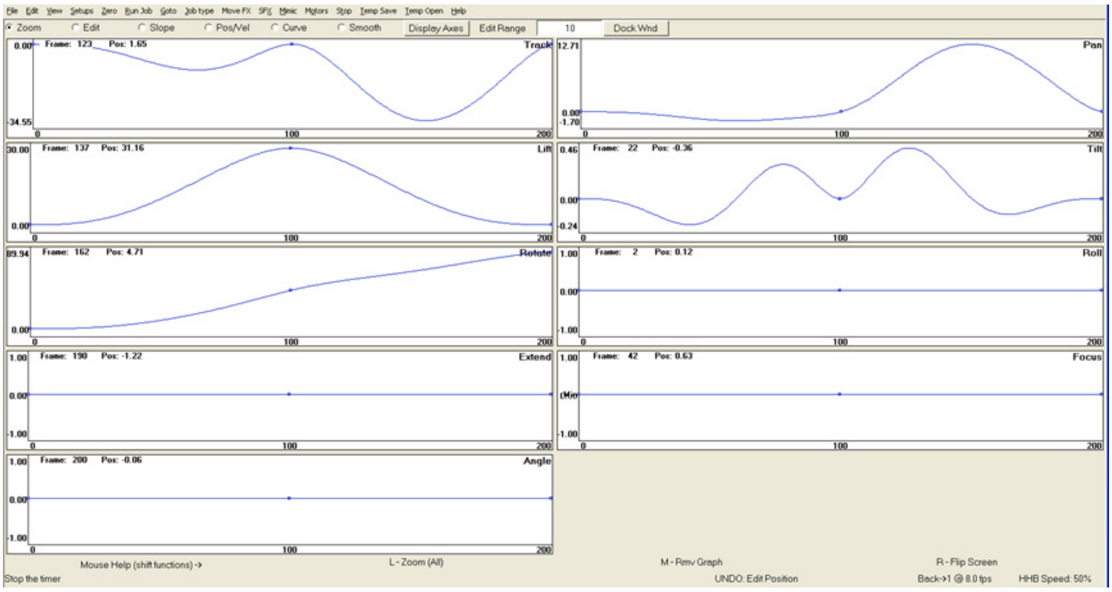
Three dimensional graphs are provided so that the user can inspect and modify the graphs to achieve the exact move wanted. There are four graphs, the first is an actual 3D representation of the move, and the others are 2D views of the move path from above (Plan View) from the side (Side View) and from the front (Front View). Each of these shows 2 of the dimensions against each other i.e. X against Y, Y against Z and X against Z. On each graph two paths are shown, the red path is the target path and the blue path is the camera path. There is a small graphic representation of the camera on the camera path and the target on the target path located at the current position in the move. If you stop the move half way through and look at these displays, the camera will be in the middle of the move, looking at the target which will be at its respective position on its path. In the 3D representation, there is a grey grid that indicates the floor level and also an arrow marked with an X that indicates positive X motion. In the 2D displays of the 3D data, there may be grey lines which are an approximate representation of the range of motion of the rig. This is for assistance when importing moves into the system to make sure that they are within the bounds of motion of the rig. In Pan/Tilt target tracking, there will only be a camera path.



You can perform various functions in the 3D Rig view. ‘Zoom’ works as usual, rolling up using the middle mouse button zooms in the 3D Rig view and rolling the middle mouse button down zooms out the view.

10.7. Cartesian Graphics

The Graphics view can be changed to show Cartesian axes (X Camera, Y Camera, Z Camera and X Target, Y Target & Z Target or Pan, Tilt & Roll, X Up Y Up and Z Up, Lens Focal length, Interocular distance and Convergence) in the same way that an axis position can be graphed against time by selecting ‘Cartesian Axes’ in the Graphics view drop-down. These graphs can be edited and modified in exactly the same way that the axis positions can be edited. See that section of the manual for details. When in Pan/Tilt target tracking, the target graphs are replaced by Pan, Tilt and Roll. There is a potential confusion here as these are NOT the same as the axes: Pan, Tilt and Roll. These are the orientation of the camera in space which is a composite of the mechanical pan, tilt and roll axes as well as the other axes, particularly the Angle (or Outer Arm) axis on a Milo or Cyclops Motion Control Rig. Added to this display with Flair Version 6.4 are also now Camera Azimuth, Camera Elevation, Camera Distance, Target Dx, Target Dy and Target Dz. See the different target tracking modes for explanation of these and how they are used.



Adding points on the Cartesian Graphic display will add a blank line with a waypoint on it for only that axis. As covered above, this is not recommended practice. Nor is deleting waypoints for one Cartesian axis.

Controls in the Cartesian Graphics are identical to those in the Axis graphics and are not described here.

10.7.1. Reverse Kinematics and Cartesian/Axis priority

The whole principle of reverse kinematics basically depends on working out from the position of the camera and target (or camera orientation in the studio) where each axis of the rig is. This subject is complicated when you have more axes than you have data to start with and is also complicated by there being more than one way to get a camera to be in one position and to look in a certain direction. Imagine a camera looking forwards - now pan through 180 degrees and tilt down through 180 degrees and you are in the same place, looking in the same direction. (Although rolled 180). Both of these are acceptable solutions to the reverse kinematics, but you cannot just jump from one to the other. There is considerable math inside of the software to keep track of such things.

Now imagine a target tracking move that is plotted with the camera at 0 and a target of 100.00 units at frame 0; at frame one there is a waypoint with a target of 100.00 and the pan and tilt are both at -180. If you run this move out of target tracking, the pan and tilt will move backwards and forwards as you run the move. If you run it in target tracking, the camera will remain still throughout the move as, as far as it is concerned, neither the camera nor the target has changed position. Now press BCK RUN and see what happens! The camera will ask for and execute a goto to the entered end positions and when the move is run will stay in that position. Again, the same camera position and orientation, but a different set of axis positions. (Assuming you are in Keep Horizon Level).

Some users have run into this. The phenomena is that the rig will do a massive goto at each end of the move and will even do a large goto if you stop it in the middle, the camera usually seems to be in the right place, but the axes are in different positions than you would expect. If you ran the move to the end and then stored the end positions to the actual motor positions at the end of the move, that would handle it. However, there is also the question of mid points and the basic problem with those is that if you stop at them, the rig will immediately do its goto. The rig is not finding the wrong solution, it is finding the closest solution to the entered positions at the closest waypoint. The handling for this is to change the positions at the waypoint t the problem.

This is done by selecting **Axis Positions** or **Cartesians** in the Priority To field in the Job Type Configuration display. When you do this, a dialog box will appear asking if you want to “Regenerate axis positions based on Cartesian data”. When you click “Yes” the computer will take a few seconds to walk through the move and store the positions of each waypoint to the correct axis positions as though you were doing a full forward run. When the process is complete, you will be informed that the move has been modified to fit the reverse kinematics. It is recommended that you save the job before you do this, it is a non-reversible process and could cause some unwanted result – if it does, send the move to MRMC for diagnosis.

In Axis Priority Mode, if any values in the move change, Flair will generate NEW Cartesian positions based on the entered axis positions. Each time a change is made the entire move is re-calculated. In Cartesian Priority, once a Cartesian move have been generated (or imported), this flag can be turned on, and then no changes in the entered axis positions should affect the Cartesian motion of the camera. They may very well affect the way in which the rig executed the motion, but the lens path should always remain the same unless directly edited in the Cartesian graphs or the numeric Cartesian display.

The potential uses of this are perhaps more extensive than it would first appear. Take for example shooting two different sets for an effects shot that have to be matched together. They could be scaled, but for the sake of argument let’s say they are the same scale. One of these sets has to be shot on a periscope due to the restrictions of the model; the other cannot be shot on the periscope due to lighting and live action considerations. Under normal circumstances, you are in trouble. However; select Cartesian priority with the periscope move fully calculated up and then switch the periscope/snorkel off and select a prime lens (same focal length!) and ensure that the nodal offsets are correct. When you run this move, the rig will move the camera to match the position and orientation produced by the periscope, but now it has a prime lens, and you will get the same actual lens motion but with a different rig configuration. Obviously, this is going to be as close to a perfect match as the rig is perfectly accurate and the lenses perfectly set up. You have to be the judge of what the limit of the shot is and how good you have to be in matching the move exactly.

10.7.2. Cartesian Priority and Importing Moves

A second use of this feature is when you are importing a move. The system will automatically switch into Cartesian Priority and then you can adjust the motion of the master axes without changing the imported data at all as long as you stay in Cartesian Priority. When you have the move worked out, you can of course perform reverse kinematics based on the new positional data that you have. You additionally have the capability of adjusting the positions of the master axes whilst in Cartesian Priority to find the fastest speed at which the rig will run without having to move to each position, adjust in Cart Control, store and re-run!

There are other tricks that can be done using Cartesian Priority – for example if you want the tilt arm to be on the other side of the pan and still have the same camera move, you can do the following:

1. Run to the beginning of the move.

2. Switch into Cartesian Priority.
3. Add (or subtract) 180 degrees from the entered pan positions.
4. Change the tilt axis positions by 180, making sure you are still within the limits of the tilt axis motion.
5. Browse the move carefully making sure that the rig runs correctly.
6. Use the “Convert” feature as above.
7. Switch the priority back to Axis Positions so you can continue editing the move. You may have to change the underlying roll axis positions if you are in Roll Relative. The reason this works is that once you are in Cartesian Priority, the camera position is maintained, no matter where you put the target tracking axis positions. However, the Pan and Tilt initial positions are solved to be the closest to the entered waypoint data. This way you can force the pan and tilt to flip whilst maintaining your framing. There is now a button to do most of this process for you under **Effects > Flip Pan Arm**.

Camera Pointing the Wrong Way

The camera might sometime point the wrong way when importing a move. This is related to their being multiple solutions to any given imported camera position and Flair finding an incorrect one. There are two simple ways to handle this:

1. Place the camera in approximate correct position to start the move when you import. In this way Flair knows the best solution for the start and then subsequence waypoints are based on that.
2. In the Miscellaneous Setup, the “Pan Handling” option allows you to tell Flair whether to flip the Pan or the Tilt axis when it comes across a situation where the camera may be pointing in the wrong direction. In this case, either “Flip Pan” or “Flip Tilt” will usually sort out the issue. The Original state of this setting is “Off” and Cartesian Control works best with that setting especially when you pass over the top of the target.

10.7.3. Editing Cartesian Priority Moves

When you are in Cartesian Priority, changing the axis positions does not change the move. This can be a little surprising, so once you import a move or switch into Cartesian Priority, you must remember this. However, you can edit the move by changing the graphs or by storing or entering data into the Numeric Carts display. The other incidence where storing axis or Cartesian positions does not change the move is if the Cartesian is a mimic playback axis, so watch for this also if a move is not being changed as expected.

10.7.4. Advanced Move Effects

The top menu bar “Effects” pulldown has a couple of advanced target tracking specific features.

Regenerate Targets If you have a move with a target that is too close to the camera, you can get undesirable motion in the orientation of the camera as it tracks the target. This can often happen with an imported move if the CGI package has put the target too close. In this case, the system must be in Cartesian priority. You can simply enter the target distance required in the “Target Column” and then regenerate targets creates new target points at the distance entered in the same line as the original one. It could also be used to bring the targets closer as needed.

10.7.5. Advanced Target Tracking Notes

Nodal Point

The exact location of the Nodal point is not that vital to target tracking working well however it is important for CG Import and Export as well as Scaling. The target tracking Mathematics tends to correct for errors in the Nodal Point.

Snorkel Target Tracking

You can only use target tracking with a snorkel when you are in “Keep Head Level” with a Milo or Cyclops kinematics model that has an outer arm. However if you look at a snorkel, you might notice that it is actually very like a normal that had been “Top Mounted”. In fact if you uncheck “Snorkel” in the set up and set the X and Z nodal offsets, you will find that you CAN target track and use the angle to gain extra reach. You will still run into issues at Til90 and when you look straight up or down in Roll Level or Roll Relative, but use of the angle makes target tracking much easier to use as you can reach much further than when in “Keep Head Level”. Some snorkels allow you to turn the end objective so that it looks to the left or right rather than straight up (When the rig is at zero). You can also set this up as a “Left Mount” or “Right Mount” and achieve otherwise quite tricky moves easily. This is an advanced feature and do not use it unless you are very comfortable with these settings.

10.8. Speed Profiling

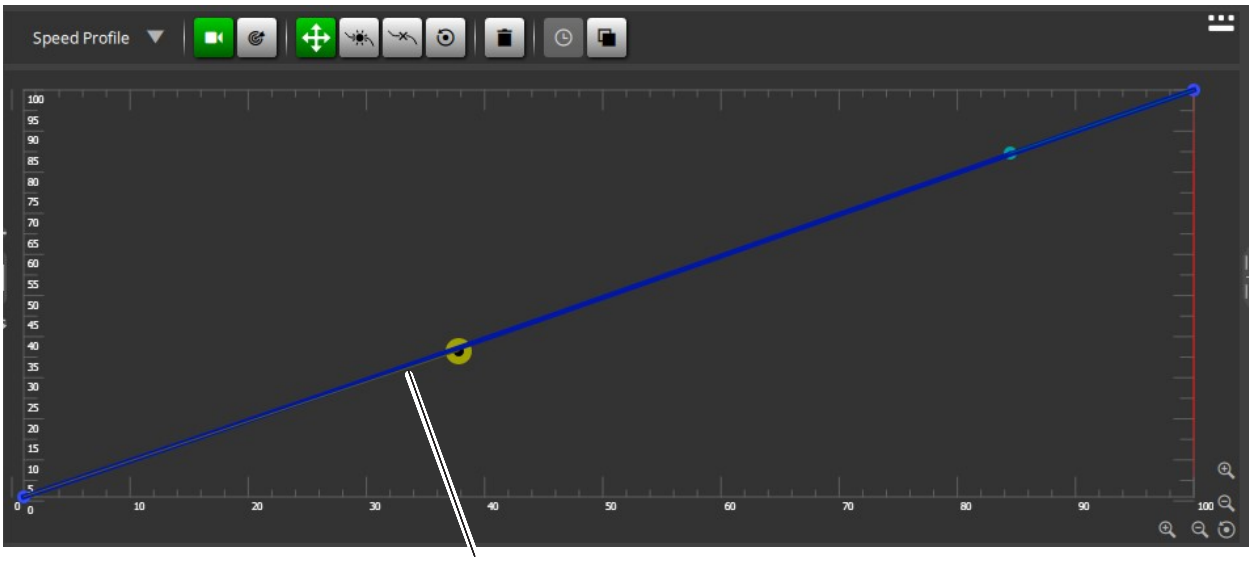
Speed profiling is a means of modifying the speed that the path is traversed without affecting the shape of the move. With the normal curve fit (splines) that is traditionally used in move generation, the shape of the path is affected, in some cases severely, by the times allotted between waypoints. Therefore if you try to change the speed by adjusting the waypoint times, you will also affect the shape of the move. Speed profiling is provided as a means to avoid this and will operate both with Bezier curves and with normal splines though it operates in different ways.

There is a potential confusion here which is to do with the shape of the curve. The move path can be created with different curves, including cubics and Bezier curves. The speed profile can also be created with cubics and Bezier curves. To make the distinction, we refer to the speed profile as being a Bezier or cubic form and the actual path as being a Bezier or cubic path.

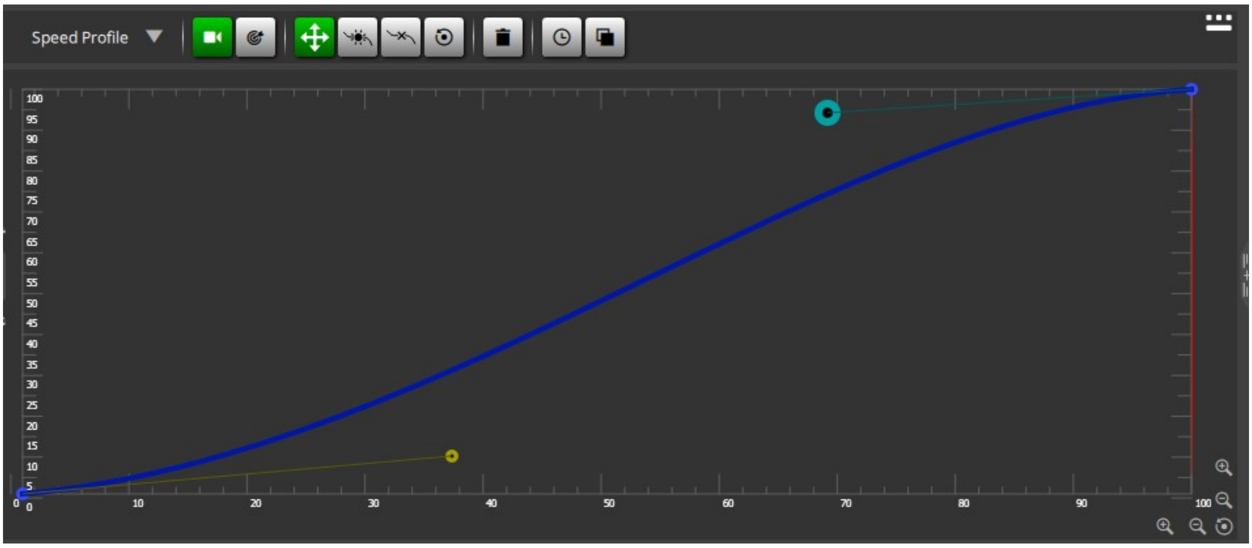
Speed profiling can be achieved by adjusting the timing of the move (achieved with speed profiling options "Bezier Form" or "Cubic Form") or by adjusting the speed along the move, achieved with the Speed Profiling Option "Fairings Form".

10.8.1. Bezier Path Speed Profiling

When you create a move path with Bezier curves, then the waypoint times are actually ignored, the only times that matter are the start and end. The move path is considered as a distance to be travelled, and the speed profile for a Bezier path is simply a graph of time along the horizontal axis against percentage of path travelled up the vertical axis (shown in frames of the move). In the default Bezier path profile, the profile starts slowly in the bottom left hand corner of the graph indicating a slow speed along the path (the profile line is almost horizontal indicating a slow change in distance over a larger change in time therefore a slow speed), then as the move progresses, the slope of the curve increases, showing increased movement along the path as the move continues, then towards the end of the move, the slope flattens off again as the speed along the path decreases to a smooth stop at the end. The gradient of the line shows the speed of travel along the move path.



Default Spline Speed Profile



Default Bezier Speed Profile

Spline Path Speed Profiling

Splines are handled differently, and it is important to understand the difference between them. With a Bezier path, you are directly controlling the speed of the camera and target along their paths. With a spline, the speed along the path has been defined by the waypoint times, but you can affect the speed at which the move progresses. The way this is done is to control the rate at which time passes during the move. Now normally you cannot change time and its rate of change, but with Spline Path Timing Profiling, there is real time along the horizontal axes, and this passes at the rate of 1 second per second, up the vertical axis is virtual time, and the rate that this passes is controlled by the speed profile curve. The default value is to have virtual time passing at the same rate as real time, so that default graph is a straight line from bottom left to top right. If you add points to this graph and edit the graph, you can affect the speed of the camera and target along their respective paths. If you cause the curve to bow upwards, starting with a greater slope, and then flattening out, the camera (or target) will start moving along its path faster at the start, and then will slow down. As with Bezier curves, the best way to learn is to play around with it before you try to use it in a shoot. It is also possible to cause the camera or target to move along its path, change direction and move backwards, and then change again. This effect is not normally used, but gives you some idea of the flexibility of the speed control.

Fairings Form Speed Profiling:

As with other speed profiling, you can use this method of speed profiling to modify the speed of the camera or target along its path. In this representation, the graph should appear more like a velocity graph, starting usually in the bottom left corner, increasing up to a constant speed along the path, and then slowing down to the bottom right corner. This is a more normal representation of a speed profile and it also works well with holds in the move which the other form of speed profiling do not. Holds are automatically plotted on the move as a double position on the graph at 0 which is how a hold should appear as it will occur with 0 speed. However in order to make full use of this feature, you MUST put the holds in the move before you start editing the speed profile or it will not add them to an existing speed profile. In the fairings profile, the vertical axis is relative speed, and no absolute speeds can be read off the graph.

Speed Checking

As you modify the speed that the move path is traversed, you will also be affecting how fast the motors are running to achieve that path. If you make the speed along the path high at some points, it is possible that the motors will not be able to achieve that speed, and therefore when you try to run the move, you will have a speed error reported. The handling is to lessen the speed along that section of the path, allow more frames, or reduce the camera speed. In any case heavily profiled moves should be checked at slow speed before running them at full speed.

Speed Profile Control Buttons

Zoom

As with the other graphs, this function allows you to define a zoom window to more closely inspect part of a graph

Flip Screen

Used to switch between viewing the one graph you are on and all the graphs.

Edit Point

Move Waypoint: This allows the user to pick up a speed profile point and adjust its position. You cannot edit the first and last points as these are fixed. When the mouse button is clicked, you will "pick up" the closest point, and then a red box will appear to show where you are moving it to, and as you move it around, the coordinates of where you are moving it to will show in the top left hand corner of the screen. When the point is placed, then the speed profile is redrawn to reflect the change you have made.

Delete Waypoint: When clicked, the closest waypoint (as long as it is not the first or the last) will be deleted.

Add Waypoint: A waypoint will be added when the right mouse button is clicked at the current cursor location.

Edit Control: When you are using Bezier & Cubic Form curve fits, you can control the shape of the speed profile without moving the points. This is handled by using what are called control points. Control points are represented by hollow circles while points in the graph are shown as solid squares. Most waypoints have 2 control points that affect the shape of the profile as it approaches and as it leaves a point. With Beziers, the distance of the control point from the point affects the "tightness" or tension of the profile at that point; with Cubics you cannot change the tension. The slope of the line from the control point to the point affects the shape of the profile as the point is approached and as it is left. Plot some simple profiles and experiment with these to see how they react. The curve can go up and down the graph, but it must always move to the right as you travel its length. If the curve doubles back on itself, then the move will make large jumps as it runs.

Move Control: Allows you to pick up the control point and move it around showing you the adjusted speed profile. Note that the curve is not allowed to go off the top or bottom of the graph and an error will be generated is you try to run such move.

Reset Point: Clicking the Middle mouse button near a point will reset the slope of that point to the default.

Reset Path: Right clicking the mouse anywhere in the graph will reset the entire profile back to the default. The default path is a gentle "S" shape for Bezier curves, and a straight line from bottom left to top right for spline curves. The default Fairings Form profile is a slope up, a flat section, and a slope down.

Time lock

Time lock is a feature whereby you define the speed profile such that the waypoint times – stored in the Numerical Display – are followed, but the rest of the move can be adjusted as desired. The button when pressed causes the existing move to be time locked, if you then modify the move, you will have to press this button again to make it time lock. This feature is not available on the Fairings Form profile as of the current release.

Speed Profile Copy

It is possible to copy the camera profile onto the target if this is desired. Simply click the “Cam → Targ” button on the top line of the display.

Errors

In Bezier form and Cubic Form, the speed profile slope can go up and can also go down, indicating that you have reversed direction, and are going back along the path, however, the slope must never become vertical or curve back on itself. These conditions are caught by the computer on the whole, but very nearly vertical lines can cause the rig to lurch.

If you have plotted a target tracking move that follows its target nicely, but when profiled, fails to follow the target, this usually means that one or more of the setups on the system is incorrect. If this happens, check the 3D displays to see if you can spot any obvious discrepancies and check the lens Setup and possibly even the Kinematics Setup. Correction of errors in the Target Tracking Setup requires careful elimination and testing and is best done with assistance from Mark Roberts Motion Control.

Path Speed

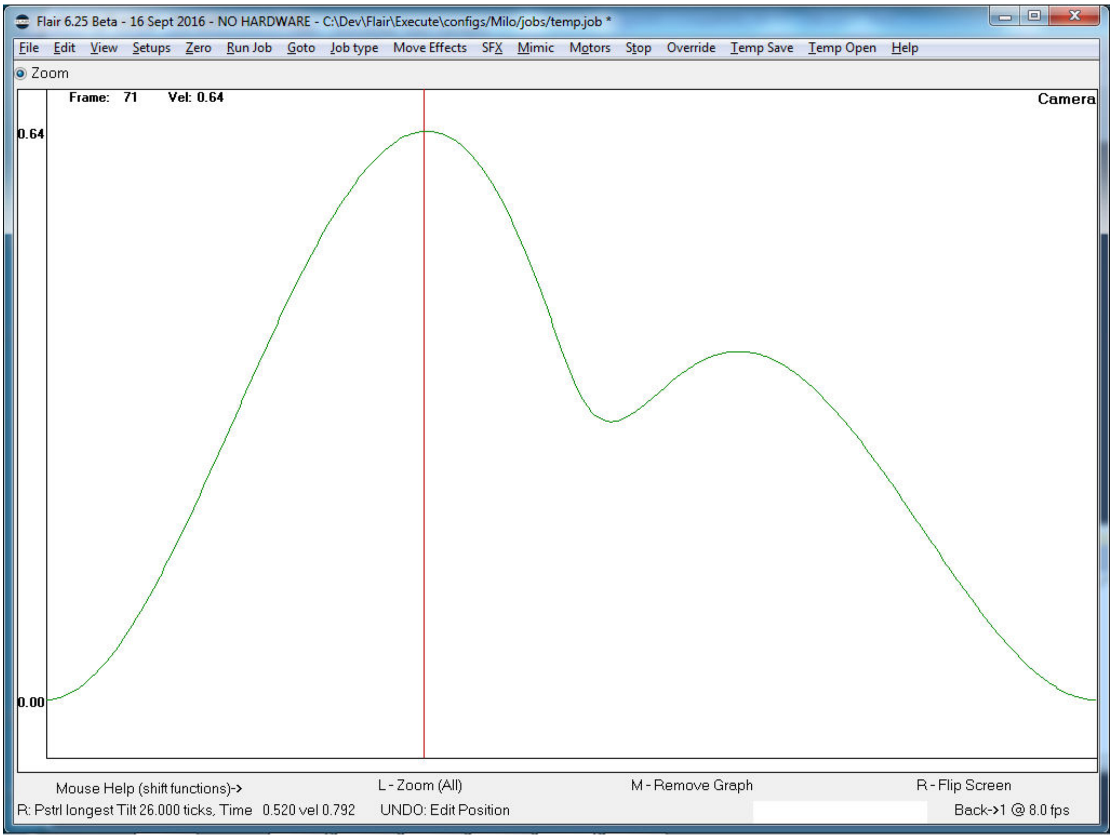
The Path Speed graph shows a representation of the speed of the camera and the target along its path against the frame count of the move. Sometimes the noise of the motors can change as the rig moves and give you a misleading idea that the camera is speeding up or slowing down. Looking at the graph will tell you how the speed is changing with time. Changing position or frame count will alter it based on the curves and times assigned to each.

The axes are time along the bottom and metres per second up the side. Only Target Tracking waypoints are editable and are shown with a larger square. Non Target Tracking waypoints as displayed as a smaller square.

Only time editing of the waypoints is currently supported. If you click and drag a waypoint you can change the time of that waypoint between the one before it and the one after it. Small motions can change the speed dramatically so gently adjustments only should be made. Changes are displayed in that graph only when you release the muse button.

It is only valid in Normal Target Tracking and Pan Tilt Target Tracking. In Normal Target Tracking you can have a camera and target path, but only the camera path is valid in Pan Tilt Target Tracking.

It does NOT work with speed profiling or Bezier curves, and a speed profiled move will not correctly display.



Flair 6.25 Beta - 16 Sept 2016 - NO HARDWARE - C:\Dev\Flair\Execute\configs/Milo/jobs/temp.job *

R: Pstrl longest Tilt 26.000 ticks, Time 0.520 vel 0.792

UNDO: Edit Position

Back->1 @ 8.0 fps

10.9. Cartesian Control

10.9.1. Principle

Cartesian Control allows the user to move the camera or target around just as though it were a point in space while always keeping the camera pointed towards the target, and can facilitate inputting a target tracking move. The camera point can be moved around to desired waypoints, these can be stored as per normal, and then the move can be executed either in or out of target tracking. The system does not have to be in target tracking for Cartesian Control to work.

Once Cartesian Control has been initialised and switched on, the Hand Held Box controls directly move the camera nodal point or the target in X, Y and Z directions. There are three types of Cartesian Control: "World", "View" and "Roll".

10.9.2. Cartesian World

In World Relative Cartesian Control, using the X button will move the camera nodal point in the X axis direction which is parallel to the track of the rig, using Y will move the camera side to side at right angles to the track. If you use the Z button, the camera will move vertically up and down. The same applies for the XYZ buttons that control the Target Point. If you move the camera nodal point, the camera will follow the target as the camera moves around, if you move the target point, then the camera will follow it.

10.9.3. Cartesian View

In Cartesian View mode, the control is similar except that the axes are relative to the current viewing direction. This means that the X direction is towards the target, the Y direction is side to side and the Z direction is up and down. For example if you were pointing directly at a target and pressed the Y Camera button and held it down, the camera would swing right round the target following it all the way round. If you use the X button, then you can zoom in and out of the target. The XYZ movements also apply to the target, but since the target is a virtual point rather than a real point like the camera, it can be tricky to control this effectively. Note that the roll position does not affect the directions you move in, so you have to look at the rig, not necessarily the monitor when using this control.

10.9.4. Cartesian Roll

In Roll mode, the concept is very similar to the Catesian View mode except that the axes are adjusted depending on the position of the roll axis and so if you move using the Y button, the image on the screen should slide side to side regardless of the orientation of the roll orientation of the camera.

In Pan Tilt Target tracking, there is no Target and the “World” “View” and “Roll” modes are not valid, you can however use the below “Locked” versions.

10.9.5. Locked World

In this mode, the camera is moved around as in the world mode, but the target point moves with the camera. This means that if you move the camera to one side, the target moves as well. The Target motion is as in the View relative mode, and the target can be panned and tilted around. The focus button can be used to drive the target away from and toward the camera.

10.9.6. Locked View

Locked View is like Locked World except that the directions that the camera moves in is dependent on the orientation of the camera. The best thing to do is to try it out and see what it does. It is a cross between "Cartesian View" and "Locked World".

10.9.7. Locked Roll

This is basically the “Locked” principle applied to the “Roll Mode”.

As with many other parts of the software, the best way to find out about this is to try it out and see what it does. Roll and Locked roll methods are best used with Horizon Relative. They will work in horizon level (but then it is the same as Cartesian View and Locked View) and it will work with horizon independent, but then the orientation of the camera changes, and you get an arcing effect.

10.9.8. Cartesian Up and Locked Up

When a camera is looking down or up at a steep angle and you pan, the camera does not simply look to the left or right, it will tend to arc in a circle to the left or right. This is the normal behaviour of a camera head where the Pan axis is vertical. However if you are looking at a steep angle and want to actually just look to the left or to the right, you can use this mode. Note that when you do this, in Roll Level or Roll Relative, the roll will tend to twist. Therefore “Roll Up” may be a better mode for this type of control.

10.9.9. Going Over the Top

If you need to program a target tracking move where the camera goes over the top (or underneath) a target, the roll control HAS to be “Roll Up”. If you use Roll Level or Roll Relative, the roll motion will surge suddenly when you are looking straight up or straight down. In “Roll Up” this will not occur. You do need to make sure that you are not also travelling through Tilt 90 when this occurs, and this is best achieved with a Milo or Cyclops by having the arc you are travelling up on NOT in line with the arm of the rig and using the Angle axis to reach out over the Target.

10.9.10. Caveat

There are a couple of situations that can cause the rig to move very suddenly when in Cartesian Control, and these should be understood and avoided.

- Never drive the View Point of the Camera into the Target point; the camera will attempt to turn through 180 degrees instantly and the rig will lurch.
- Unless you are sure you know what you are doing, do not let the rotate get more than about 75-80 degrees from straight ahead or straight back. When you the rotate approaches 90 degrees, the track will move very fast, and this will also affect the quality of the move. Extend the arm out, or use the Angle if you have one. You can also drive the target point through the view point if you is not set up correctly and you pull focus to move the target closer to the camera. This will only happen if you have not set your focus limits correctly.

10.9.11. Master Axes

When you are in any of the Cartesian control modes, the master axes are still active as usual on the hand held box, but moving them does not affect the position of the camera. The result of this is that if you move a master axis, the other axes will compensate to hold the camera nodal point and target position stationary. This is quite fun to play with, but it also has a use in that you can re-configure the rig for lighting or speed reasons without changing the framing or move path when in target tracking. Note the warnings and limitation listed below. When you do this, you may notice that there is a slight lag in the compensation, this is only 1/50 of a second, and will not occur during any move.

10.9.12. Initialisation

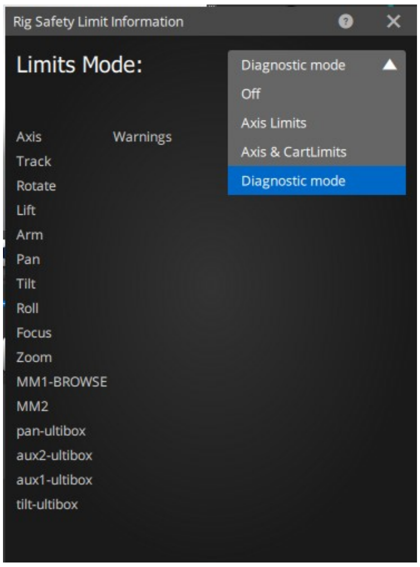
Initialisation of Cartesian Control occurs when you switch it on. Turn on Cart Control by selecting from Cartesian: “World”, "View", "Roll", "Up”; or Locked: “World”, "View", "Roll", "Up”. Select the option you want and the system may request a small “Goto”, if this is executed, you will then be in Cartesian Control. This will be reflected by a line in the bottom of the LCD on the Hand Held Box.

If you are using Cartesian Control and moving the target, then it is best to have the focus axis in one of the computer controlled modes as this makes the positioning of the target easier.

10.9.13. Cartesian Control Soft Limits (Rig Safety Limit Information)

Cartesian Control soft limits restrict the speed or motion of the camera when in Cartesian control so that axis soft limits and axis velocities are NOT exceeded when the camera is moved around. The basic mode of Cartesian Control soft limits will slow the rig down or stop its motion if that would cause an axis to travel out of range or either position or velocity.

When the Carts Limits mode is in **Axis Limits**, it will only limit the axes. If in **Axis & CartLimits**, it will also limit the motion of the Cartesian values based on the limits set in the “Cartesian Setups”. In **Diagnostics mode** it will limit based on the above and also pop up a dialog box to show you what is causing the limit.



Rig Safety Limit Information

Limits Mode: Diagnostic mode

Axis	Warnings
Track	
Rotate	
Lift	
Arm	
Pan	
Tilt	
Roll	
Focus	
Zoom	
MM1-BROWSE	
MM2	
pan-ultibox	
aux2-ultibox	
aux1-ultibox	
tilt-ultibox	

The check mark shows which axis is causing the issue, and the box checked indicates if you are on a limit, getting near a limit, exceeding the velocity of an axis or its acceleration or if the Cartesian position is being limited. Once you move out of the restricted area or move more slowly, the dialog should automatically disappear.

The Cartesian Limits (Rig Safety Limit) is very sophisticated and works well to move the rig in a safe mode. It is still the responsibility of the operator to be aware of what the rig is doing at all times. It is possible that the Rig Safety Limits will stop what should be a legal motion of the rig. Should this occur, try turning the safety off and see if that resolved the issue. Report this to MRMC and turn the safety back on again once the rig is out of the restricted area.

10.9.14. Control Assignments

The XYZ controls are achieved by the buttons that control the motion most similar to the one you are trying to achieve. Track will move the camera in X, Rotate will move it in Y, Lift will move it in Z. The target is moved in X by the Focus button, Y by the Pan button and Z by the Tilt button. Master axes are controlled as per normal but note that as you move a master axis, the other axes will move to hold the camera stationary as you are not actually moving the XYZ of the camera. The Hand Held Box should indicate which form of Cartesian Control you are in. Master axes are most commonly found on "Milo" and "Cyclops" rigs. In some kinematics models, there are options as to which axes are master axes and which are target tracking. This will change the operation of the buttons on the hand held box, so note this and get used to the response in different modes.

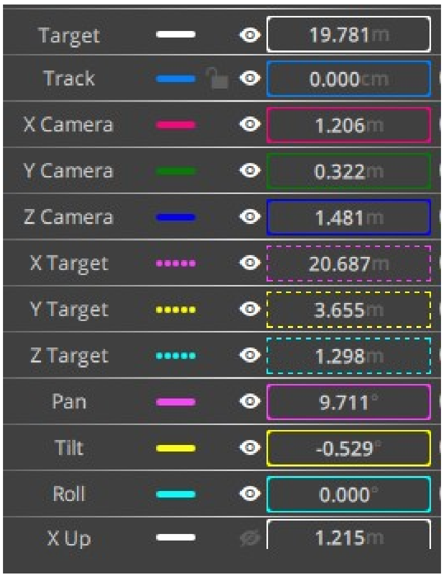
10.9.15. Control by Handwheels

If you have an axis controlled in Encoder Velocity Mode that would otherwise move a Cartesian axis, you can use the handwheel to control it. It will be active whenever you are in Cartesian Control and the Hand Held Box is on. If you are controlling Pan and Tilt in Encoder Vel mode (the most likely use) then the axis speeds of the Pan and Tilt are limited based on the Pan and Tilt Hand Held Box velocities. Note that this is different from the normal Pan and Tilt velocities when in Encoder Vel mode. This is for safety as in Cartesian Control, the rig moves around to compensate to hold the nodal point stationary as you pan and tilt.

10.9.16. Positional Readout

In Cartesian Control, some of the axis positions are replaced by the related Cartesian coordinates of the Camera and Target. If the axes names have changed to X Camera, Y Camera, Z Camera etc, then it is showing the Cartesian position, not a true axis position. These are the XYZ coordinate of the Camera (per the above axis alignment). The positions displayed for X Target and Y Target are actual studio pan and tilt angles. (for example, if the arm is rotated to 20 degrees and the camera is panned to 30 degrees, the actual studio pan angle is 50 degrees as the pan and rotate both pan the camera, but in different ways.) This is particularly helpful if you have an Angle axis and are not in keep head level.

The Cartesian positions read out over the place of the axis positions. If you wish to see the axis positions AND the Cartesian Values, clicking the cog wheel in the Waypoint view will show toggle buttons can switch between the Cartesian and Axis positions. You can also move the rig around by using the left and right mouse buttons over the labelled Cartesian.



Target	19.781m
Track	0.000cm
X Camera	1.206m
Y Camera	0.322m
Z Camera	1.481m
X Target	20.687m
Y Target	3.655m
Z Target	1.298m
Pan	9.711°
Tilt	-0.529°
Roll	0.000°
X Up	1.215m

10.9.17. Warnings and Limitations

The use of Cartesian Control is an advanced feature and only experienced users should attempt it. In this mode, you can enter Cartesian soft limits in the Cartesians Setup box as well as control Cartesian velocities and accelerations, but in the current release, the system is NOT aware of axis limits and will bypass all target tracking axis soft limits. In addition, the restrictions and limitations that apply to target tracking apply to Cartesian Control with the added warning that the rig will attempt to resolve Tilt 90 and Rotate90 situations with possibly violent movements of the pan, tilt and track axes. You must not get close to these situations when in Cartesian Control and also must not drive the camera point to or through the target point. Should any of the above situation occur, it is very likely that Cartesian Control will be switched off. Having said this, with a practised eye and some understanding of these limitations and three dimensional space, Cartesian Control can be extremely useful is setting up moves that might otherwise involve a lot of multi axis manipulation just to achieve certain framings and positioning.

10.9.18. Turning Off Cartesian Control

Cartesian control can be turned off simply by going into the “Side Menu Control Menu” and switching the Cartesian Control selection to “Off” and then moving the mouse out of the Menu Bar. Cart Control will also turn off if any axis attempts to jump violently as can happen if the above restrictions are not obeyed. It will also turn off if you attempt any form of Goto or Move running. These modes of control are mutually exclusive.

10.9.19. Table Target Tracking

There is a kinematics model that is significantly different in its approach to target tracking to warrant a fuller description than the other methods. Since there is no pan and tilt on an Animation table, the position of the target is somewhat irrelevant and does not really concern us. The use of target tracking with an animation stand would be in the ability to view and edit composite graphs of motion. When you have an animation stand moving a piece of art work around and moving the camera closer to it and further away, you could regard this as simple moving the camera around above a flat table. This way of viewing it immediately gives one the camera motion part of target tracking and opens up the full extent of target tracking features such as cart control and Graphic display and editing, as well as speed profiling.

Pan/Tilt target tracking will not work, nor will scaling, but the rest of the features of target tracking should be fully available. Being able to view the 3D motion of the camera relative to the artwork can reveal the more direct effects of moving points around in space. Also having the rotate axis modelled in the software will allow one to use it more fully. For example if you want to turn about a part of an artwork that is off centre of the table, it is almost impossible. With target tracking on, one should simply be able to rotate in Cart Control with the current image held in the centre of the and then plot the two points and because the camera does not move relative to the art work, the target tracking axes (N/W and E/W) should compensate for the master axis Rotate. (From this one can deduce that Track, N/S and E/W must be assigned as target tracking axes and that Rotate must be a master axis in the axis Setup).

10.9.20. Table Auto Focus

Auto focus is not really a target tracking concern, but since there are similarities, it is being covered in this section of the manual. Auto focus gives one the ability to move a camera closer and further from a piece of artwork but to keep the image sharply focused. Since standard target tracking can also do this in a similar way, there are similarities and differences here. You do not need to have target tracking switched on for auto focus to work.

When auto focus is engaged, the computer derives the position of the carriage and the focus motor based on the Setups and a notional camera height and a notional image height. As one uses the hand held box buttons to move the carriage and/or focus, the computer is actually moving a notional camera height and a notional image height and moving the camera and focus based on these. These notional heights above the table are displayed on the screen in red in the Track and Focus axis columns. Switching in and out of auto focus changes between the two modes. The track position and column of positional data actually reflects the notional camera height above the table in the measure units that the system is set for (usually inches or centimetres). The focus position and column of positional data reflects the height of the object (target) relative to the table. This value would normally be 0.0 and stay that way, but you can use this feature to pull focus onto thick objects placed on the table, or to pull an image in or out of focus should this be desired. When autofocus is switched off and then on again, the focus will shift back to the table top.

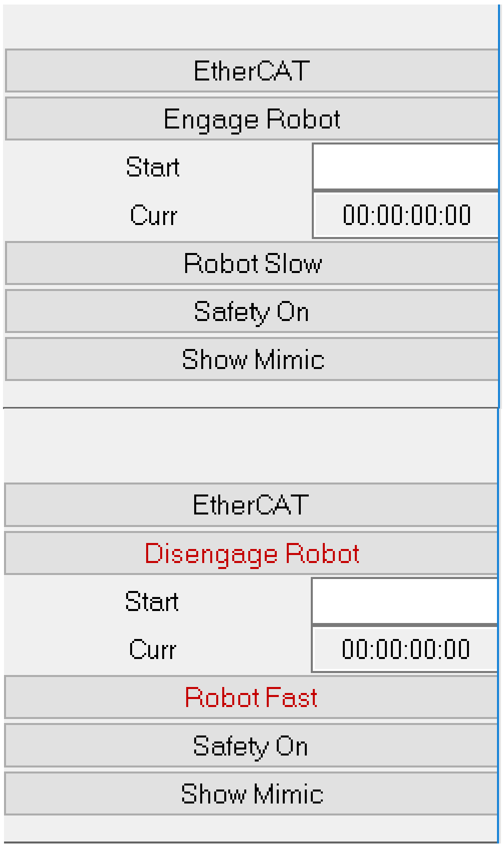
When you switch on autofocus, the computer looks at the current positions of the carriage and the lens and works out the height of the lens above the table based on that data, it then pulls focus on the table with the lens at that height. Moving the axis up and down in autofocus adjusts that height above the table. The limits on the travel are restricted by how low the carriage can go and also by how extended the bellows on the camera can be (with a moving lens). Whilst every care has been taken to make these limits safe, you should always approach the table with care and generally not at top speed. The top limit is generally restricted by the carriage as in most cases the lens should be able to go to infinity focus. If the maximum height of the carriage is restricted by the motion of the lens, then the user will have to watch this as it is not accounted for in the software.

Chapter 11 **High Speed Robots**

High speed robots have changed the face of traditional motion control and allow for extremely high speed motion and to shoot useable shots at speeds as high as 2000 Frames Per Second and still get appreciable camera motion in the slowed down footage.

The ability of these machines to move so fast has necessitated some changes and the addition of safety features in Flair.

When you have a high speed robot, it will by default run at 1/6 of the programmed speed unless the user specifically sets it into “Fast Mode”. You cannot browse a move in fast mode and any attempt to run any motor faster than 1/6 of the stated maximum axis speed including under Hand Held Box or manual control will stop that axis unless it is in fast mode”. This feature ensures that is should be impossible to run the robot fast by accident and only a deliberate change from **Robot Slow** to **Robot Fast** should allow high speed motion.



If the robot mode is changed to “Robot Fast”, it will automatically revert to “Robot Slow” if no attempt to run a move is made within 10 seconds.

When ready to shoot the move with the robot in Fast Mode, the screen will not state the usual “Ready to Shoot” but will announce in red “Ready to Shoot Fast” ensuring that the use knows the robot will move quickly.

Since the default speed is 1/6 the normal speed, external camera synchronizing and triggers are NOT valid in slow mode and should not be used or relied on.

Should you wish to use the Robot in slow modes only, then the robot should be configured for slow axis speeds by MRMC and you can engage a mode called “Collaborative Robot”. Once it has been confirmed that the axes can only run at safer speeds, “Collaborative Robot” can be engaged and the Fast Mode/Slow Mode restriction is lifted.

11.1. Robot Setup

Each Type of Robot has a slightly different setup screen.

The three types currently supported are:

- Staubli Robots communicated to over EtherCAT® known as "Bolt".
- Kuka Robots communicated to using their proprietary protocol.
- Motorman Robots communicated to over a custom network protocol. Known as "Cobra".

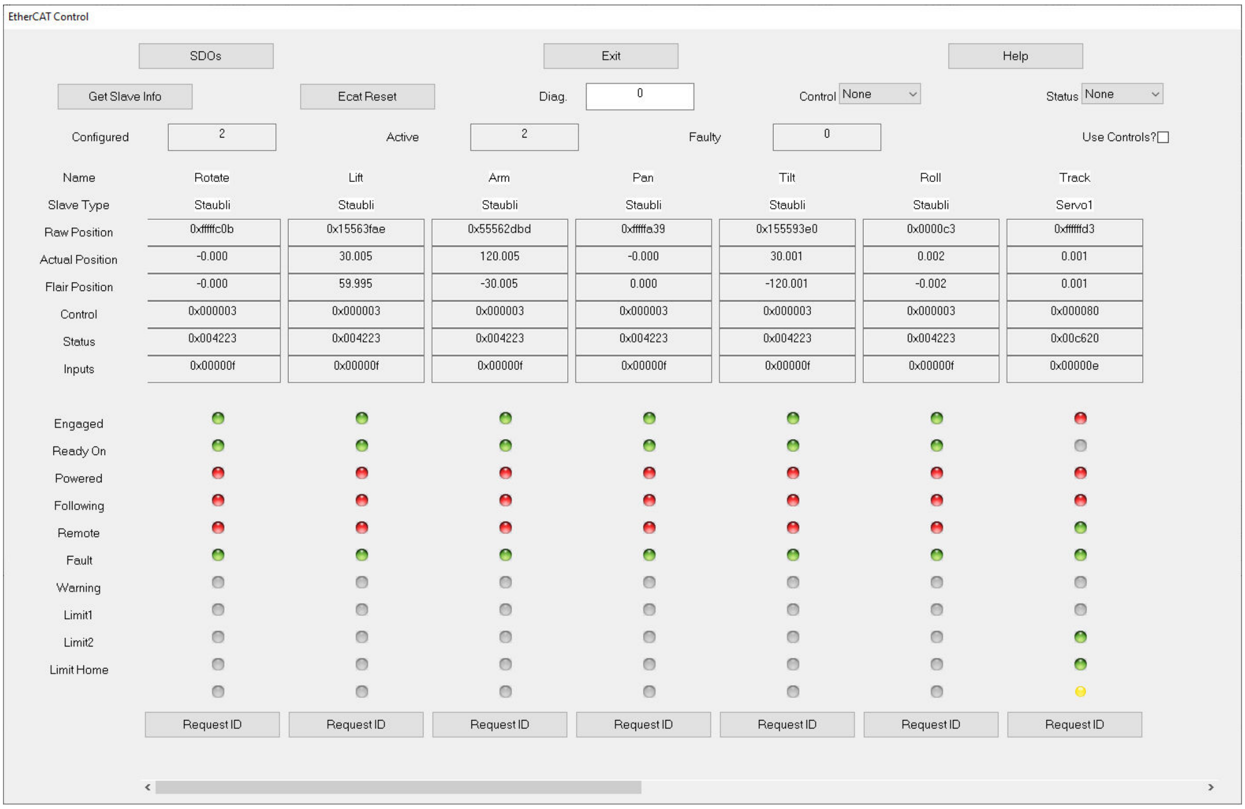
Robots operate differently in that you cannot turn on just one robot motor, but have to engage all axes at once. This is usually done with the additional "Engage Robot" button which is on the "Side Menu" and can optionally be added to the Hand Held Box and the Run time control bar. If you disengage any robot axis in the normal way by clicking on the named axis button on the top of the numeric screen, it will disengage all the robot axes. The reverse is not true, you cannot enable all the axes at once using one axis button, you need to use the "Engage Robot" button.

All robots have their own safety mechanisms and if for any reason these safety systems trip, then the entire robot will disengage and the reason has to be found and remedied. Each robot is slightly different, but they will all fault and control will be lost if communication to the robot fails, if an axis hits a limit or is run too fast and of course if an Estop is pressed and power is removed from the robot. The handling is slightly different for each robot, but the concept is basically the same. You may have to use the Robots control pendant to move it off the limit, but generally if communication is working and estops are released you can usually re-engage the robot from within Flair. See the appendix for Robot specific information.

11.1.1. EtherCAT & Staubli

The Staubli robot is controlled using an industrial standard communications protocol called "EtherCAT". This protocol is also supported by other system, and amplifiers. Due to this, you can use the EtherCAT interface to control the Staubli robot AND other EtherCAT drives, or simply control individual EtherCAT drives. In the case of using Staubli or other drives with it, you do have to ensure that the axis control and connection is correctly set in the axis setup. With other robots, it doesn't really matter what you state in the Network, Node and Port since the system knows which axes are in the robot. When using an EtherCAT drive including Staubli, you do have to ensure that the "Network" is set to "EtherCAT", the Node is set to 1, and the port number is correctly set based on the order of axes in the setup, but usually, if present the robot uses the first 6 ports. With EtherCAT, there is a different EtherCAT setup screen to the robot setup, and this can be used to see which axes are assigned to EtherCAT and to diagnose issues. If you have a Staubli robot, the Staubli setup will only show the Robot axes, but the EtherCAT setup will show all axes including the robot. At the moment only Node 1 is allowed for EtherCAT.

EtherCAT status screen



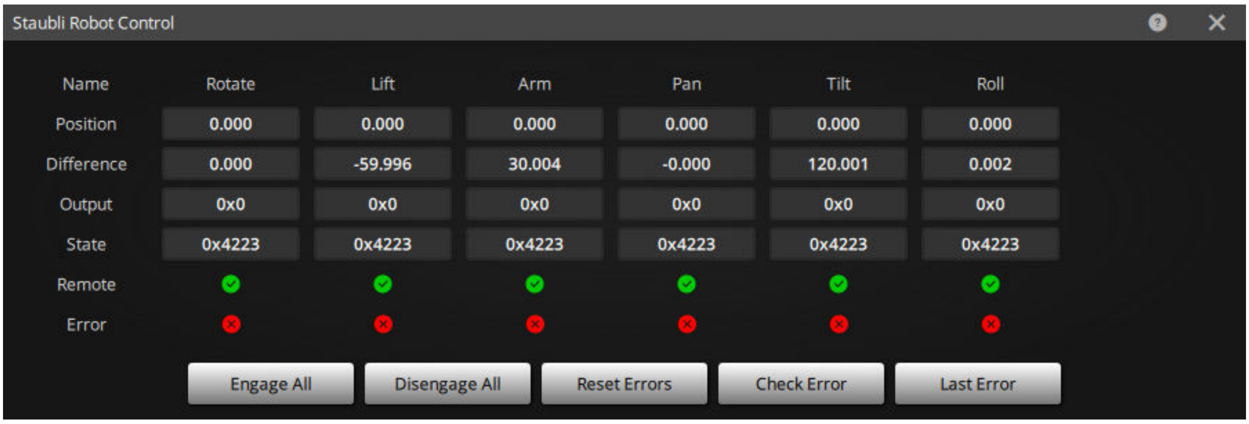
Diag. 0 Control None Status None

Configured 2 Active 2 Faulty 0 Use Controls? □

Name	Rotate	Lift	Arm	Pan	Tilt	Roll	Track
Slave Type	Staubli	Staubli	Staubli	Staubli	Staubli	Staubli	Servo1
Raw Position	0xffffc0b	0x15563fae	0x55562dbd	0xffffa39	0x155593e0	0x0000c3	0xffffd3
Actual Position	-0.000	30.005	120.005	-0.000	30.001	0.002	0.001
Flair Position	-0.000	59.995	-30.005	0.000	-120.001	-0.002	0.001
Control	0x000003	0x000003	0x000003	0x000003	0x000003	0x000003	0x000080
Status	0x004223	0x004223	0x004223	0x004223	0x004223	0x004223	0x00c620
Inputs	0x00000f	0x00000f	0x00000f	0x00000f	0x00000f	0x00000f	0x00000e
Engaged	🟢	🟢	🟢	🟢	🟢	🟢	🔴
Ready On	🟢	🟢	🟢	🟢	🟢	🟢	⚪
Powered	🔴	🔴	🔴	🔴	🔴	🔴	🔴
Following	🔴	🔴	🔴	🔴	🔴	🔴	🔴
Remote	🔴	🔴	🔴	🔴	🔴	🔴	🟢
Fault	🟢	🟢	🟢	🟢	🟢	🟢	🟢
Warning	⚪	⚪	⚪	⚪	⚪	⚪	⚪
Limit1	⚪	⚪	⚪	⚪	⚪	⚪	⚪
Limit2	⚪	⚪	⚪	⚪	⚪	⚪	🟢
Limit Home	⚪	⚪	⚪	⚪	⚪	⚪	🟢
	⚪	⚪	⚪	⚪	⚪	⚪	🟡

Clicking the EtherCAT button in Flair Classic, displays the EtherCAT Status screen. The EtherCAT Status screen cannot be viewed while the robot is ok due to timing issues. If you look at it, the robot will trip out but can be re-engaged as soon as the screen is closed.

11.1.2. Bolt (Staubli) status screen



Name	Rotate	Lift	Arm	Pan	Tilt	Roll
Position	0.000	0.000	0.000	0.000	0.000	0.000
Difference	0.000	-59.996	30.004	-0.000	120.001	0.002
Output	0x0	0x0	0x0	0x0	0x0	0x0
State	0x4223	0x4223	0x4223	0x4223	0x4223	0x4223
Remote	✓	✓	✓	✓	✓	✓
Error	✗	✗	✗	✗	✗	✗

The Staubli status screen shows a listing of each of the Staubli axes along with some status information. Generally speaking the user will not need to look at or use this screen, but information in it may be useful in the case of control issues with the robot.

Position: shows the current position of the robot axes.

Difference: Shows the difference between the commanded position and the position the robot is at. Ideally this value is 0.

Output: The control command being sent to the Robot axes. This will change when the robot is on or off, and would normally be 0x000b.

State: This is the state of the robot axes. There are specific states that the robot should or can be in. Use the "Help" button for more information.

0x4221 means OK, 4223 means On, 4227 means ready to be controlled and 0x5237 means under control. If you have any issues with the robot, the value of its state might assist MRMC with resolving your issue.

Remote: This says if the robot is in Remote mode which it does need to be to be controlled by Flair. The robot has a physical switch for this, and if the robot is not communicating or of the switch is in the wrong position, this box will be unchecked. It needs to be checked for the robot to work.

Error: If an axis has run into an error, then this box will be checked. Errors are normally cleared by turning the robot off and then on again using the "Robot Engage" button.

11.1.3. Robot Extra Axis 1 & 2

A Bolt (Staubli robot) can be mounted on the floor or it can be mounted on a track system. If you engage or disengage the Robot, you would probably want the Track to engage or disengage at the same time. If the Track motor is selected as a "Robot Extra Axis" it will turn on and off with the Robot if you use the "Engage and Disengage buttons. You can also set a secondary extra axis for example, Focus. This can be set up by default using the .ini file (See Technical Chapter on this).

The lower buttons are for future developments and the user does not need to operate them.

Chapter 12 **Mimic Mode**

12.1. Principle of operation

There are times when it is desirable to have manual control over one or more axes while the rest are controlled by a computer generated curve. This need can occur in a live action shoot where the subject is not pre-programmed and manual control of some axes can allow the talent to be kept in shot. In addition it can be required for a movement so generated to be played back to produce matt passes. Mimic Mode does just this, it allows the operator to have total manual control of as many axes as he can control throughout the move, and records any movements on these axes for replay if needed. In addition, some people feel that a computer controlled move can be too smooth and prefer to have a more “organic” feel which mimic can provide.

12.2. What controls a Mimic Axis?

An axis can be controlled when it is mimicking by several different sources:

- The hand held box
- A handwheel or encoder connected to the mimic board in the root box (if there is one)
- Grip Sticks attached to the rig
- Itself

When an axis is controlled by the hand held box, it operates exactly as normal. When it is controlled by an encoder, the motion on the encoder is used to determine the movement of the driven axis.

12.3. Types of Mimic Control

Handheld Box: This form of mimic control allows the axis to be controlled exactly as though it were being controlled on the Handheld Box during the move, or using Grip Sticks, and the motion will be learnt and recorded.

Encoder Pos: In this mode of control, the motion of the axis will EXACTLY duplicate the motion of the controller. Moving the controlling handwheel around and returning it to an exact position will return the axis to an exact position. This is a good mode of control if this type of response is needed. One of the drawbacks is that if you try to move the axis too fast; it will turn off, and the shot will be lost. Additionally, you can drive the axis into its end limits and in that case it will also turn off and the move will stop. The movement from the handwheel can be smoothed by using the smoothing window. This has the effect of sampling the encoder's movement over the stated range of time periods (in 1/50's of a second) and moving the controlled axis based on the average of these. This smoothes the motion, but also introduces a slight lag. Encoder pos mode also covers what is referred to as Auto Mimic (Self Mimic) where an axis is moved physically during the shot and its motion is learnt and can be played back. This works for smaller light axes and is most often done for Focus or Pan and Tilt. It would also be applicable if you were using Flair to encode the motion of a camera crane. (See axis controller for more details on Auto Mimic mode).

Encoder Pot: Ensures that the mimic controller and the motor always stay in Sync. This is most applicable for axes like the focus. Typically when a move is played back, the controller and the axis are not in a matching position and have to be lined up manually before mimic learn can again be turned on. With Encoder Pot control, the system moves the controlled axis into the matching position as soon as mimic learn is turned on. In this case, the system goes into a “Mimic Goto” state briefly.

Encoder Vel: The velocity of the handwheel is used to control the motion of the controlled axis, with velocity, acceleration and limits processing. Due to this, the position of the controller does not always exactly correspond to the position of the controlled axis.

The maximum speed used here is the maximum velocity of the axis, not the maximum hand held velocity. This mode is always active when the Hand Held Box is on and is very useful for easy positioning of a pan and tilt head from a set of handwheels.

Speed Boat: A speed boat control is modelled on the speed control on a motor boat where “stop” is in the middle of travel, forwards is positive motion and backwards is negative motion. The maximum speed of the axis is determined by its maximum speed in the axis Setups. There is a deadband in the encoder position that relates to no axis motion, and this deadband is user settable. Whenever this mode of control is engaged, the encoder must be at the zero position as it will be re-initialised every time it is switched on.

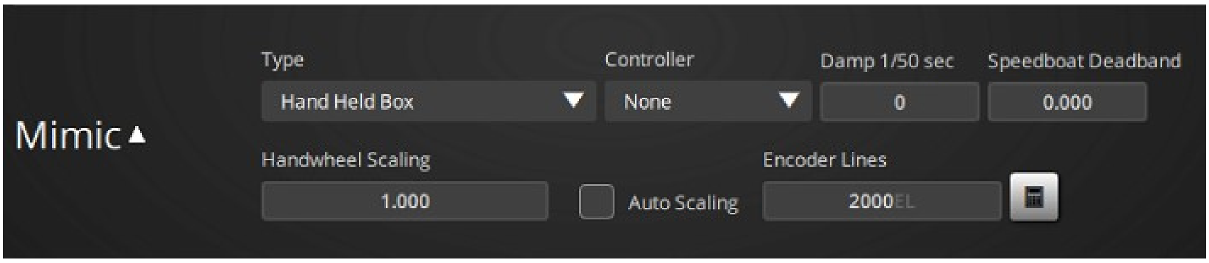
Tracking

Encoder Pos versus Encoder Vel: These two methods of control have various pros and cons which are good to fully understand. The advantage of Encoder Pos is that you get EXACT control between the handwheel and the controlled axis, there is no lag (as long as you do not use the smoothing window) and the position of the handwheel will match the position of the axis exactly as you use it. This can also be a problem as you can run the axis too fast and trip it out and also run it into a hard limit. Encoder Vel has processing to greatly reduce the possibility of this (though you can still do it if you really want to). However, due to this, there is a slight lag that you will notice on fast starts and stops.

One of the other drawbacks with Encoder Pos is that you have to be in Mimic Learn for it to work, and going into Mimic Learn requires that the handwheel be totally stationery. Encoder Vel is active whenever the Handheld Box is on, and you can do pretty much what you like with the handwheels otherwise. Encoder Vel is great for positioning the axis and getting line ups and for slow mimicking. Encoder Pos is good for that precise control of the pan and tilt which is often demanded by cameramen.

Another use for Encoder Pos is for mimicking a focus axis. The advantage being that as long as you make sure that the scale on the handwheel matches the scale on the focus whenever you switch on mimic learn, then the scales will always match. The disadvantage is that it is very easy to move the handwheel too fast and trip the focus motor out in which case you have to turn off mimic learn, re-engage the axis, make sure the scales match and then switch Mimic Learn back on — not to mention the fact that if it happens during a take, you will ruin the take. (It’s a good idea to get the focus puller used to the operation of the mimic handwheel well before you start doing takes.)

For additional detail on scaling and setting up Mimic controllers, see the Axis Setup section of this manual in Chapter 5 Setups.



Mimic ▲

Type: Hand Held Box ▼

Controller: None ▼

Damp 1/50 sec: 0

Speedboat Deadband: 0.000

Handwheel Scaling: 1.000

□ Auto Scaling

Encoder Lines: 2000EL

12.4. Mimic Add

Mimic Add is a feature that has been added to allow an existing move to be perturbed by the movement on the mimic control. One example of this would be to have a complex move on an actor, and use the pan and tilt handwheels to make only minor corrections if the actor misses his mark, but to allow the move to carry on as per normal otherwise Another possible use is to add some roughness or shake to simulate effects of motion or disturbance (for example, helicopter, earthquake etc.). When an axis is in “Mimic Add”, it will run its move as per normal, but will be displaced from this move by any movement of the controlling handwheel. Otherwise a Mimic Add axis behaves exactly like a Mimic Learn axis. Mimic Add only works with "Encoder Pos" control. You cannot Mimic Add to an axis if it is already mimicked.

12.5. Mimic Live

There can be occasions when you wish to mimic an axis, but do not need to record the data. In this case, you can use the “Mimic Live: setting whereby you will not be prompted to warn that you are going to overwrite data, and this also speeds up job storage and retrieval. This is quite commonly used with the Focus axis where the different passes all have different focus measurements, and different layers may be de-focussed in post for optimum effect.

12.5.1. Setting a Mimic Controlling axis:

Setting the mimic controller axis is done simply through the Mimic dialog box. If you are not using an encoder for controlling the axis, then this can be set to “No Controller”, otherwise select the Mimic Axis encoder that is being used.

12.5.2. How to mimic an Axis:

There is a button on the Top Menu Bar labelled “Mimic”, if this button is pressed, a pulldown menu appears which has an option “Axes Mimic”, if this is pressed, a box comes up with a list of every axis on it along with four radio buttons for each axis. The default state of these radio buttons is “Off”, the other 3 states are “On”, “Live”, “Add” and “Save”. To set up an axis to be a mimic axis, select “On” and press the “Apply” button at the top of this dialog box. When you return to the numeric display, that axis name will appear in red to show that it is a mimic axis and to warn you that it will not follow its plotted waypoints. Then set the “Mimic Off” toggle in the Control view to “Mimic Learn” click the Learn button in the Mimic menu. Once this has been done, the axis will be controlled by the controller that you have set up.

12.5.3. Learning a Move

Enter the move that you want on any other axes in the standard way, not worrying about the stored positions of any mimic controlled axes if you are using “Mimic On”. Move the mimic controlled axes to their move start positions and run the move. As the move executes, you are free to move the mimic controlled axes as you wish, and the movements so created are stored as they occur. The move is then learnt, and at the end of the move, the waypoints on the mimicked axes are modified to reflect the mimic movement.

12.5.4. Replaying a Move

Once a move has been learnt, you can replay it. The first thing to do is to turn off "Mimic Learn" or else you will destroy any recorded move, then simply run the move again. All axes will go to their start positions, and when executed, normal axes will do the usual thing while the mimic controlled axes will replay whatever motions they went through when the move was learnt. These learnt moves can be replayed as many times as wanted.

12.5.5. Saving a Move

If you have recorded a mimic move, and one of the axes is to your liking, but another one needs to be re-learnt, set the correctly learnt axis to “Save” in the Mimic Setup Display, and then its move will be saved and replayed without danger of re-recording it. In fact you could mimic every axis in this way, one at a time. A saved move can be re-learnt, simply by setting the toggle back to "On", or you can return it to a normal motion control axis by setting this toggle to "Off". When an axis has its move saved, then its name will remain red on the numeric display to remind you that it will not follow any entered waypoints.

12.5.6. Self Mimic

Self mimic (also called Auto Mimic) is a special case of mimicking when an axis learns the positions it is being pushed through and can then replay those. It requires that the axis be able to be back driven or moved in some independent way and it must be disabled from computer control. It can be done on some dollies where the track gearbox is not too stiff, or can be done on pan and tilt head if you can fit handwheels over the motors. The motor obviously has to be re-enabled to play back the move. Note that there will inevitably be some difference between the learned motion and the played back motion depending on the stiffness of your system. Only tests will indicate how much this is.

12.5.7. Mimic Storage

A learnt mimic move can be stored to disk just like any other, and can be restored as needed. It does however slow down the process quite a bit as a lot of data has to be saved. The user need do nothing extra, the recording is handled automatically by the software.

12.5.8. Stopping a Mimic Learn

If you stop a mimic learn while it is recording, you are given the option to shorten the length of the move down to the time at which you stopped the move. This will only work if no non-mimic axes were moving at the time you stopped the move. Normally if a move is stopped during a mimic learn, then that learn has been abandoned and will not be played back.

The normal use of this feature would be to program in the move desired on computer controlled axes and set the length of that move, and then add an extra line as a Rehearse hold onto the end of the move of a few seconds extra. When the move is shot, the programmed axes will execute their move and come to a stop at the end when the system goes into its hold. The mimic can be continued at this point until all the mimic action is completed, and then the stop button can be pressed. When this occurs, a box will appear which asks you if you want to truncate the move. If you reply YES, then the move will be shortened down to the length that it was shot, and you can replay it at will. If you reply NO, then the move will not be changed, but the mimic learn will be abandoned. If any non-mimic axis was moving when you selected mimic truncate, then trying to truncate the move will produce an error.

Warning: If you are doing a lot of different takes in this fashion, remember to reset the end frame count of the move after you have saved the move. If you do not the move will stop at the shortened point and any subsequent motion will NOT be recorded on the next take.

12.5.9. Recording Axes before the Move

It sometimes happens that axes need to be recorded before the principle motion of the move is triggered. This can easily be done by entering a hold, but with unpredictable motion (Animals etc) you do not always know exactly how long to make the hold. In order to handle this eventuality, there is an advanced feature called “Mimic Learn+”. When you select this option on the Side Control Menu, and the camera is rolling with the turnover button before the move, then the axes’ motion will be recorded during the preroll and once “Shoot” is pressed, during the move. If you stop the move during its execution as in the section above, then you will be asked if you want to truncate, once this is done (or if you did not stop the move), a box will come up asking if you want to insert learnt data into the move.

If you have a good take, then you should respond “Yes” to this and after turning off mimic learn promptly save the job to disk. You will notice that the move will be modified by a point added at the start with as many frames inserted into the move as you were learning before “Shoot” was pressed. Any active triggers will have their frame count increased by this inserted value, and obviously all the move frame count will be adjusted up to suit.

With Mimic Learn +, it is even more important to change the job back to the original form after each take as many things will be changed. It is recommended that you set up a base job on disc and re-read that in order to reset the frame counts and the triggers.

This is an advanced feature and you should practice it before you use it in a shoot condition. It will also work with an external camera, but it will only work if a camera is running. On occasion, you may get the insertion dialog box to appear at inappropriate times. If this happens, just press Exit and it will disappear without changing anything.

12.5.10. Mimic Edit

You can edit this data in the Graph view and pull or push around the curve to achieve the exact motion you want as well as being able to smooth out any resultant bumps.

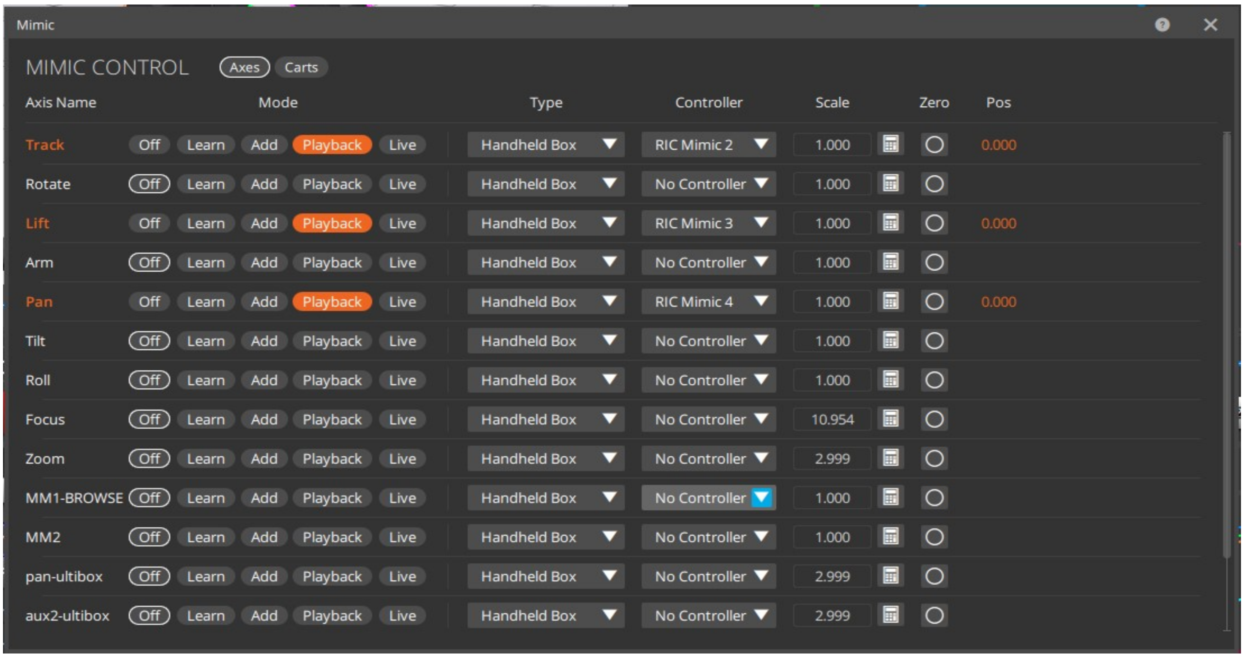
When in the Graphic display you usually edit a curve by picking up and moving one or more of the waypoints. When the axis is being controlled by this "mimic" form of data, then you can pick up any part of the curve and drag it in position, up or down. The editing effect of the mouse motion is applied to the graph over a range which is defined in frames by the "Edit Range" box at the top of the screen. The greater the edit range, the wider the effect of the editing is applied to the graph. You can edit across a large range and make a change across a large amount of the move, or a small range and pull around a few frames worth of motion. It is always a good idea to smooth the graph after you have made any changes like this to ensure a smooth junction between the modified and unmodified sections of the move.

12.5.11. “Start at” and “Make This” with mimic

“Start At” and “Make This” are useful features for offsetting programmed moves. They also work with Mimic moves and a mimicked axis can be offset so that the motion is the same, simply shifted as needed. A good application for this is replaying a Pan and Tilt in a different location for a background plate. If the move is correct, but not panned into the desired location, one simply offsets the pan axis to make the move execute in the right place. This does not change the move itself, simply where the move executes. Used incorrectly, of course this can change the camera motion and so lose the purpose behind replaying mimic move (You would not offset the Tilt or Zoom for example).

12.5.12. Mimic Display

Mimic display is designed to make setting up and running mimic moves easier and faster. When **View → Mimic Controls** is selected, a display pops up that has all the required buttons and setting for configuring and running mimic moves. In this move the normal Mimic Off/Mimic Learn button is not required and is greyed out.



Mimic

MIMIC CONTROL [✓ Axes] Carts

Axis Name	Mode	Type	Controller	Scale	Zero	Pos
Track	Off Learn Add [✓ Playback] Live	Handheld Box	RIC Mimic 2	1.000	○	0.000
Rotate	[✓ Off] Learn Add Playback Live	Handheld Box	No Controller	1.000	○	
Lift	Off Learn Add [✓ Playback] Live	Handheld Box	RIC Mimic 3	1.000	○	0.000
Arm	[✓ Off] Learn Add Playback Live	Handheld Box	No Controller	1.000	○	
Pan	Off Learn Add [✓ Playback] Live	Handheld Box	RIC Mimic 4	1.000	○	0.000
Tilt	[✓ Off] Learn Add Playback Live	Handheld Box	No Controller	1.000	○	
Roll	[✓ Off] Learn Add Playback Live	Handheld Box	No Controller	1.000	○	
Focus	[✓ Off] Learn Add Playback Live	Handheld Box	No Controller	10.954	○	
Zoom	[✓ Off] Learn Add Playback Live	Handheld Box	No Controller	2.999	○	
MM1-BROWSE	[✓ Off] Learn Add Playback Live	Handheld Box	No Controller	1.000	○	
MM2	[✓ Off] Learn Add Playback Live	Handheld Box	No Controller	1.000	○	
pan-ultibox	[✓ Off] Learn Add Playback Live	Handheld Box	No Controller	2.999	○	
aux2-ultibox	[✓ Off] Learn Add Playback Live	Handheld Box	No Controller	2.999	○	

The first five buttons on the display control the state of the axis. Mostly they are off, and as such, no colour is displayed. Any axis that is another mode is show in red so they are easily discernible. Clicking a state puts the axis directly into that state with a single click.

On the right of these state buttons are all the settings you might need for controlling a mimic axis, you can set the scaling, the mode and the controlling axis. These are the same as the values which you will find in each axis setup, but made accessible specifically got Mimic Control. Below the “Scale” Label is the current handwheel scaling for that axis. This number is taken straight from the axis setup and when it is changed, the value in the axis setup is directly changed. The “Set Scale” button pops up the same calculator that you would use in the Axis Setup when you press the “Manual Scaling” button. “Zero” allow you to zero the controller which is vital for use with “End Pot” mode. Just above the “Zero” button is a read out of the controller position which is helpful to see if you are correctly set up for mimic control.

Note that the feature to record data before the move is not provided in this display and you will have to use the normal “MimicLearn+” mode. However, this display is very useful for rapid setting up of multiple mimic axes.

An additional difference with this mode is that when you read a job back, the axes will go into the mimic mode that they were stored with. With the other system, if you were out of Mimic Learn, then any job read in is also out of Mimic Learn regardless of how it was stored.

Mimic Smooth

Data stored in the “mimic” form can be smoothed. Smoothing is selected by choosing the “Smooth” option at the top of the Graphic display. The mouse select button can be used to define the smoothing zone, the middle button will then smooth the curve every time it is clicked and will draw the new curve over the top of the old one, the right hand button is “Done” and when pressed will redraw the graph with the final smoothed curve. If you do not define a smoothing zone with the mouse select button, then pressing the middle button will smooth the entire graph. Smoothing will quickly remove sharp corners or small bumps in the move. Note that a smoothed move and the original learnt move will not be exactly the same and will not repeat, though a smoothed move will repeat subsequently. “Super Smooth” is a smoothing mode that actually removes some of the data points and so results in a much smoother curve – however overuse will rapidly flatten a graph and it should be used responsibly. One would usually do one or two “Super Smooth” only to start with and then use the normal “Smooth” to do the rest.

Mimic Restrictions

As a general rule, you may not mimic an axis if the position of another axis is dependent on it. For example, you cannot mimic the lift axis when you are in “Keep Head Level”, you also may not mimic one of the algorithmically run axes such as the focus or zoom when it is under special control. The results of doing this may work for the shot you are doing, but I cannot guarantee that it will repeat exactly between the learnt and playback moves.

If you wish to mimic the pan and tilt axes, but are not in keep head level, then the horizon will tend to tilt as you pan and tilt. In this case keep head level will work, but will tend to trip out at the start or end of the move. In order to avoid this, it is best to switch the roll into “Mimic Learn” as well and special code will ensure that it behaves as desired. You cannot at this moment mimic the horizon directly.

You can only mimic record a full forward run. You cannot do a mimic record in camera varispeed, the software will allow you, but it will not work as expected.

Focus should be in “Independent” mode when it is being mimicked and not in “Follows Object” or “Follows Target”

Target Tracking Mimic

Mimic control thus far has been applied to the control of one mechanical axis from a remote controller during a move rather than from a curve. This theory of application could of course also be applied to a theoretical axis such as the Camera Z position as well as any mechanical axis. This idea has led to the inclusion in the software of means of controlling these theoretical (or Notional) axes. The Carts Setup section allows one to define one of various forms of input as the controlling means for a Cartesian, this may be a handwheel, a joystick or even a button on the hand held box. The theory is that when the move is running, the user can alter the position of notional axis during the move and record what he has done for play back.

The uses of this become clearer when you consider the Milo or Cyclops motion control rig. If you are running the rig in target tracking and trying to frame up on an actor, the actor can easily slip in and out of frame by just missing his marks very slightly. The normal correction for this would be to use handwheels to drive the pan and tilt axes to keep the actor in shot. This is fine when the rig head is level or close to it, however if you are using the angle axis to get the maximum reach for the camera, then the actual motion of the pan and tilt axes does not correspond to the camera's pan and tilt orientation in the stage, and even the best operator has no chance of correcting for this as it will change as the Angle (Outer Arm) changes position.

If one could use the hand wheels to drive the notional camera Pan and Tilt (as in its orientation in the stage) then you would have a chance to keep the actor in frame without confusing the operator. In addition, a similar means could be adopted for moving the camera from side to side (Y) without having to make compensatory adjustments to the pan axis. From this you can also see why the PTR method of Target Tracking has been added to the system since driving a notional target that you cannot see with handwheels would still be very hard.

There is a high degree of complexity in this and it should be fully tested and experimented with before being used in a shoot situation. One thing to be aware of is that sudden Pans and Tilts about an offset nodal point can cause sudden accelerations in the Track and Lift and this may affect the move adversely. I have often reduced the X Nodal Offset of a lens to a low (and incorrect) value if I knew it was not important for later effects such as CGI Data integration or Scaling.

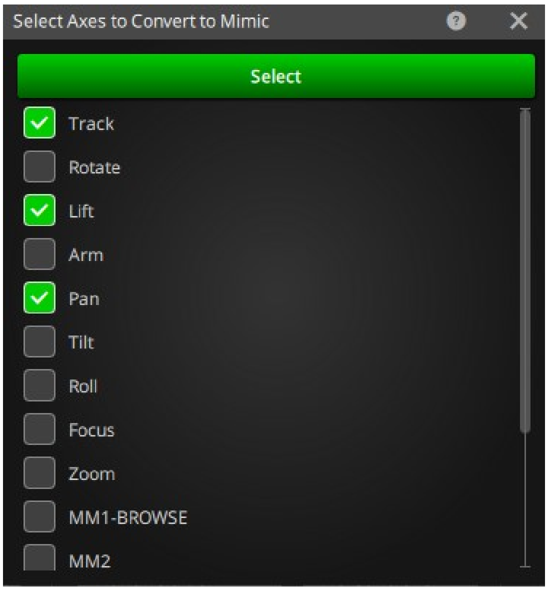
Mimic Data

In the internal workings of the system, a move is generally stored as a series of positions against time (waypoints). Whenever the computer needs to work out how to move an axis, it will calculate a curve to fit these positions, and then use the curve to work out where the axis needs to be at any given time. You don't need to know this other than to know that when you mimic record a motor's position, the positions are recorded 50 times per second and stored as a large quantity of discrete data. This is different from a normal axis move that is plotted through waypoints.

There is nothing intrinsically wrong with storing the data as a large number of discrete positions other than it is slower to save and recall the data. There is an advantage to it in that you can more easily mould the motion of the axis as there is lots of data to work with rather than a few points. The second advantage to this form of data input is that if you have to input a motion that an axis has to follow very closely (such as importing 3D data), then it is much easier to manipulate this form of "mimic" data than to have to manipulate 300 waypoints in a move.

Mimic Convert

You can import data in this "mimic" format or you can convert a normal waypoint move into this form of data. This is done through the Mimic pulldown menu selections which are titled "Axes Convert" and "Cartesian Convert" allowing you to convert axis moves to "mimic" type moves and Cartesian moves into "mimic" type moves. Pressing either of these buttons will provide the user with a list of the available axes to convert. One simply selects which axes you want to convert with the mouse so that the green checkmark appears in front of them, and then press the "Select" key. Axes that are successfully converted are switched into the "Mimic Save" mode. Any errors will produce a message, the most usual reason would be that the axis is disabled and therefore has no move calculated for it.



Select Axes to Convert to Mimic

Select

☑ Track
□ Rotate
☑ Lift
□ Arm
☑ Pan
□ Tilt
□ Roll
□ Focus
□ Zoom
□ MM1-BROWSE
□ MM2

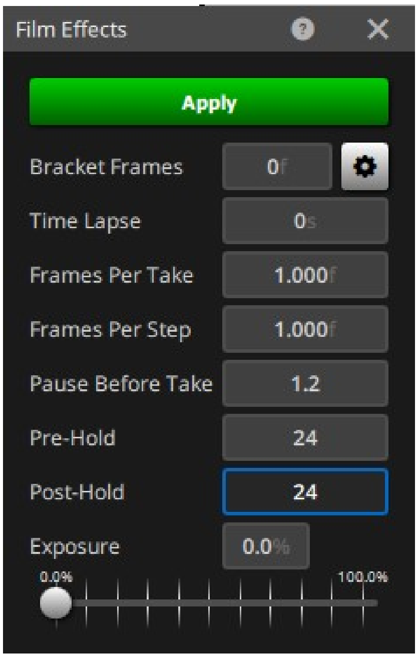
You can convert any mechanical axis motion that is planned through waypoints, you cannot convert a target tracking axis when the system is in target tracking, and you cannot convert an algorithmic axis such as focus, or zoom.

Once an axis has been converted, to change it back into a normal motion axis, switch its mimic state into "Mimic Off". Once an axis has been converted into this form of mimic data, you cannot add waypoints to the start or end of the move, the mimic data will be expanded or squeezed to run from the start of the move to the end. You can edit this data in the Graphic display and pull or push around the curve to achieve the exact motion you want as well as being able to smooth out any resultant bumps.

Chapter 13 **Special Effects**

The Special Effects setup allows the user to alter settings that mostly apply to stop-motion special effects normally used in traditional single frame motion control, but also includes scaling.

13.1. Film Effects



Bracket Frames	0f
Time Lapse	0s
Frames Per Take	1.000f
Frames Per Step	1.000f
Pause Before Take	1.2
Pre-Hold	24
Post-Hold	24
Exposure	0.0%

13.1.1. Bracket Frames

The Bracket Frames is the number of frames that will be shot at each single frame position, and the exposure on these frames can be altered to allow the user to shoot a selection of exposures for testing or other purposes. If one is not shooting a wedge, this would normally be set to 0.

13.1.2. Stepping Effects

Time Lapse

Similar to Step Delay except that each step is shot at a specific time after the move is started. Where step delay is relative to the last step, Time Lapse steps occur relative to the start of the move and are therefore more accurate. Will only work where you are running in Continuous Step mode, either Stop Motion or Moving Step. If you stop the rig while it is running a Time Lapse Shoot, the move will lose its continuity.

Frames Per Take

This value sets the number of frames that are taken with every step of the move when you single step through the move. It is the number of frames actually exposed every time a step button is pressed.

Frames Per Step

Very similar to Frames Per Take, but this value is the number of frames of the move that you advance through with every step. When Stop Motion stepping, the number of frames exposed with each step is controlled by the Frames Per Take value. When you are taking Moving Steps, the number of frames exposed is affected by this value. This value also affects how much you step through the move when you click the mouse in the move slider bar.

Pause Before Take

When stepping through a move with a portable rig that is not as stiff as a studio rig, it is sometimes necessary to let the rig settle down after each step to allow any shake to dissipate. This is not a problem when you are manually stepping through the move, but when you are continuous stepping, this can be a problem. This value introduces a delay between steps when you are continuous stepping either in Stop Motion or Moving Step. The value entered is the number of seconds of delay between each step.

Pre and Post Holds

With Pre and Post holds, you can define a hold at the end or the start of a move without having to plot extra points into the move. This has several advantages in that the hold is not part of the move, and does not have to be programmed, and it is also not scaled up and down as you adjust the length of the move to your liking. In addition, it is only executed when you actually shoot the move, so there are no unnecessary delays when you are running the move in Rehearse. Note that when you run the move backwards the posthold length is shot first, and the prehold length at the end. These holds do not work when recording onto a video system, or shooting with an external camera.

Pre-hold

You can enter the number of frames you want the camera to take before the move starts. For example if you enter 50 in this box, then whenever you record a move with a forward run, then the camera will ramp up to speed, then take 50 frames, and then the move will start. The preroll at the start of the move will then be at least as long as the prehold you have stated.

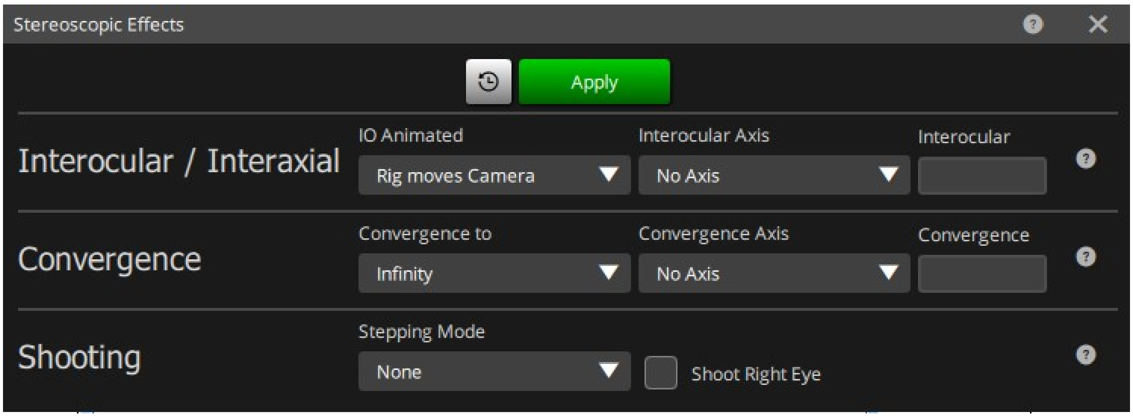
Post-hold

You can enter the number of frames that the camera will run after the move is shot. As above with Prehold. If you start the camera before the move by pressing the Turnover button when you are at “Ready to Shoot” and later trigger the move, if there is a posthold, that will be executed at the end of the move and the camera will then stop. If you do not want the camera to automatically stop in this case, set the posthold to 0.

Exposure Position

This affects the position of the exposure relative to the position at each frame. Consider a simple move lasting one frame – it has a start position and an end position. When this is shot, the exposure will occur with a slight blur from when the camera shutter opens (about 1/4 into the move) until it closes (about 3/4 through the move). In effect you are recording the centre position between the first position and the second position. When you are doing stop motion, you can vary where in this region the exposure will occur using the Exposure Position slider. For example if this slider is set to 0.5, then the exposure will take place in the middle of the frame, where the exposure would normally occur if you were shooting continuously (but without any motion blur). When you are running with Frames Per Step higher than 1, this value applies to the relative position within the whole step however many frames it is.

13.2. Stereoscopic Setup (Optional)



Stereoscopic Effects

Apply

Interocular / Interaxial

IO Animated: Rig moves Camera

Interocular Axis: No Axis

Interocular

Convergence

Convergence to: Infinity

Convergence Axis: No Axis

Convergence

Shooting

Stepping Mode: None

□ Shoot Right Eye

One of the additional special effects that you can create with Flair is shooting a left and right eye pass for stereoscopic shooting. Live action, of course, requires a stereoscopic camera, but animation and model shooting can be shot with one camera doing single frames of left and right eye or shooting the left eye and then the right eye. Mostly with puppet animation you would shoot single frames of each eye as you go along; model can be shot either way. With normal scale or miniatures, the interocular distance is often so small that you could not easily mount two cameras side by side, so moving the camera a little side to side for each frame can easily achieve the desired effect.

The Stereoscopic setup allows you to define the interocular distance and also the convergence mode of the move as well as how you want to shoot the passes.

Interocular: Defines the interocular distance in metres.

Converge to: Is a pull down menu with a list of options. Infinity, Convergence Target, Focus Distance. Converging to infinity means that the camera will simply move side to side with no convergence (toe in). You can set the desired convergence distance and the camera will converge on that and you can also have the camera converge onto the entered target tracking target or the entered focus distance.

Convergence: Allows you to set the desired convergence point if that mode is used. (Again in Metres)

Shoot Right Eye: This flag is used to determine is you are shooting the left eye or the right eye. By default the left eye is deemed to be the master and the right eye is the one that is offset from the programmed target tracking move. If the flag is left unchecked, then the left eye is shot for any pass. If you check the flag and an interocular distance is set, the right eye pass will be shot. This is valid for shooting one eye and then the other.

Stepping Mode: is valid when you single step through a move and want to shoot the left eye then the right eye. The modes are “None”, “Prompt Other Eye” and “Auto Take Other Eye” These are self-explanatory, and allow you to set the machine to prompt before it take the other eye or simply automatically do it. It would be a good idea to have checked the entire move for both eyes in a non-record mode before you did this single stepping. The rig will automatically go to the alternate eye position for each frame.

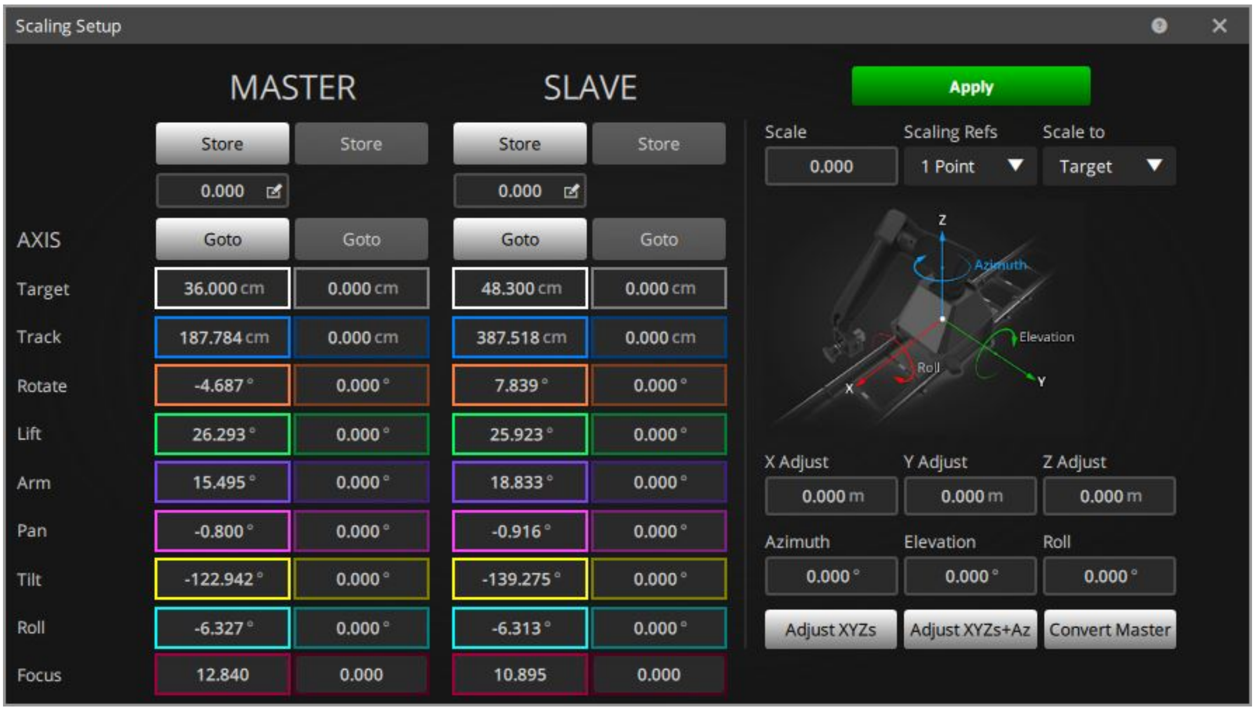
13.3. Scaling

Scaling is a sophisticated addition to the target tracking feature. It allows the user to take an existing target tracking move and scale it, up or down, to match between a model and the real item, or any two objects so that the move path is exactly the same shape, but moved and scaled, from the master move on the master item onto the slave item. The move that is entered normally is referred to as the Master Move and the scaled or adjusted move is the Slave Move. You can also keep the move the same size and use the scaling feature to move it around in space and re-orient an existing or imported move.

Scaling is one of the most advanced features of Flair and requires careful consideration and a good understanding of what you are trying to do. The mathematics behind scaling is relatively simple, so errors are often a mistake in the setup or even a misunderstanding of what exact result is expected.

13.3.1. How it is defined

There are two ways of defining the scaling for a slave move. Both methods require reference points on the master item and on the slave item to tell where they are in space. 1 point scaling only requires one point on each, and the user manually enters the scaling factor to tell the computer how much to scale the move up or down. In 2 point scaling there are two corresponding points on the Master and Slave items which serve to inform the computer of the relative sizes and positions of the master and slave items. Once these are defined, then the actual 3D Master move path is moved and scaled from the master item onto the slave item. For example you could have a full size person standing near the rig, and have a move that starts looking at his feet and lifts up and around to look at his right ear. If the user enters into the system reference points on the full size person and then enters the exact same reference items on the model, the move can be executed on the model so that the actual shape and relationship of the move is identical.



	MASTER		SLAVE	
	Store	Store	Store	Store
	0.000		0.000	
AXIS	Goto	Goto	Goto	Goto
Target	36.000 cm	0.000 cm	48.300 cm	0.000 cm
Track	187.784 cm	0.000 cm	387.518 cm	0.000 cm
Rotate	-4.687 °	0.000 °	7.839 °	0.000 °
Lift	26.293 °	0.000 °	25.923 °	0.000 °
Arm	15.495 °	0.000 °	18.833 °	0.000 °
Pan	-0.800 °	0.000 °	-0.916 °	0.000 °
Tilt	-122.942 °	0.000 °	-139.275 °	0.000 °
Roll	-6.327 °	0.000 °	-6.313 °	0.000 °
Focus	12.840	0.000	10.895	0.000

Scale	Scaling Refs	Scale to
0.000	1 Point ▼	Target ▼

X Adjust	Y Adjust	Z Adjust
0.000 m	0.000 m	0.000 m

Azimuth	Elevation	Roll
0.000 °	0.000 °	0.000 °

Adjust XYZs | Adjust XYZs+Az | Convert Master

13.3.2. Entering and Running a Scaling move

Scaling is a sophisticated feature of the program and requires considerable understanding from the user to gain the most from it. The first thing to do is to work out in your head what the desired moves are, and see if this will work in both the master and slave cases. If the master move is on a doll and finishes with the camera 2 feet above the head of the model, then the slave move on a full size man will finish with the camera about 12 feet above the head of the man (assuming 1:6 size ratio). These points all have to be thought about before the move is actually entered. In addition it is usually better to enter the larger move as the master move, and make the slave move the smaller, though this is not always possible, it has the advantage that errors are scaled down, not scaled up!

Once the move has been planned, the master and slave items must be positioned so that the move can be shot on both items without the rotate axis turning past 90 degrees on either move (see ‘scaling limitations’). Once this is done, plot the master move on the master item in the normal way entering it as a target tracking move. When the master move is worked out roughly, it is a good idea to try out the slave move to ensure that it will do it before you spend hours getting the master move exactly right, and then find out it won’t work!

In the menu bar select **Effects → Scaling**. A box with 4 columns appears: 2 columns for the master reference points and 2 columns for the slave reference points. Near the top of the screen is a box labelled “Scale” which tells you what the calculated scaling factor is. There is also a dropdown “Scaling Refs” next to it which switches between “1 point”, “2 point” and “Line Up”. When you switch this, the second column of reference points on the Master and Slave items are enabled in “2 point” scaling and disabled in “1 point” and “Line Up” scaling.

1 Point Scaling: This has the advantage that you only have to reference one point on the Master item and one on the Slave and you can manually set the scaling factor. It does not work well where the Master and Slave items are not parallel to each other, but in most cases, you would want to set them up that way. This method is easier to use in practice and should be used as a first method. The reference point that you plot is entered by pointing the rig at the chosen point, storing its positions into the column of figures on the scaling display and entering the target distance to that point in the “Target” entry box. It is the point that you are measuring to that matters, where you look at it from is relatively unimportant, but if it can be from somewhere in the move path (more or less) that is a good idea. The point on the slave item must of course be a corresponding point.

Line Up Scaling: This is an extended version of 1 Point Scaling which has been developed with the assistance and input of Jerry Nilson of Denmark. The principle is similar to 1 point scaling in that you set the size, but the orientation of the camera between the master and slave points determines the difference in orientation of the master and slave moves. This method does NOT account for tilt mismatches, and only works for rotations about the vertical axis. Tilt mismatches are ignored, but may be added with a later version. The concept is that if you grab a freeze frame on the master and store that point, then line up on the slave and store that point, it should set all the offsets etc. that you need.

2 Point Scaling: This method works well where you do not know the scale between the items or they are not located in exactly the same orientation in space. You are basically using the rig to survey the Master and Slave items so that the system knows where they are, what orientation they are and what their size relationship is. Accuracy is very important in 2 point scaling, and care must be taken. You also have to be careful that Master Item reference point 1 corresponds to Slave item reference point 1 and that Master item point 2 corresponds to Slave item point 2. As with 1 point scaling, it is not really important where the rig is, it's the point that you are looking at that is important, however putting the rig somewhere near the move path is a good idea.

You have to log the first reference point on the Master item into the first column, and enter the distance to the reference point. Then move the rig to look at the second reference point. Then enter the positions of the axes into the second reference point column, and type in the distance to the reference point in its relevant column. Now move the camera to look at the point on the slave item that corresponds to the first reference point on the master item, and enter this point as the first slave reference point. In the same way, enter the second slave reference point.

Reference points on an item must not be directly one above the other. The sideways motion of the reference points indicates the direction from which the object is to be viewed, and if you enter totally vertical reference points, then the direction of view is indeterminate and you can get very odd results, the move will be correct, but from the wrong angle. Note that with 2 reference points, you are only defining a line, and not a plane, so that the object must be in the same orientation in relation to the lines between the reference points. If you had reference points on a man's head and his toes, and also a child's head and toes, the moves would be scaled properly, but the direction from which the person was to be viewed would be indeterminate.

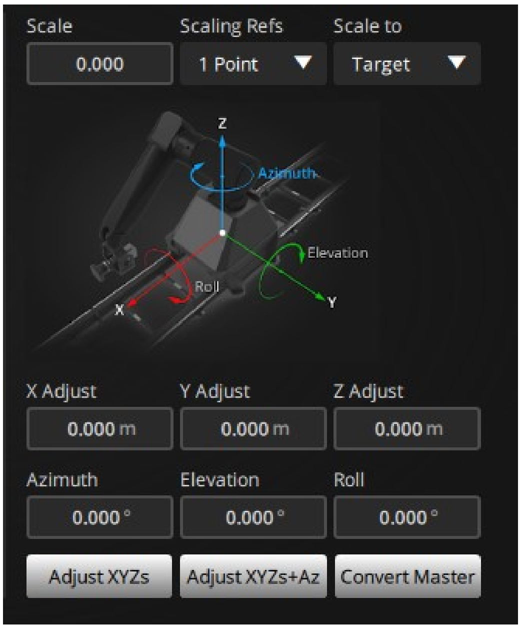
Once the reference points are all entered, they can be checked by using the "Goto" button which will cause the rig to go to the entered reference point in each case. With all the reference points entered correctly, apply the Scaling setup. In the Side Menu Control Menu, click on "Master Move" and it will change to "Slave Move". Now press FWD RUN and see if the system will allow you to run the job. If there are no errors, and you get "Ready to Goto", then go ahead and run the move. If however there is an error, it could be caused by several things, and the answer is to go into the "Scaling Effects" box again, and check the reference points. At this time, the actual size ratio between the master and slave move will have been calculated, and will be displayed in next to the label "Scale", and that may assist you in finding the fault. The most likely cause of a major error is a badly entered reference point or points. It could simply be that the move is too fast and it only needs a few more frames, or it could be that the reference points have been entered backwards on the slave as compared to the master move, and the camera wants to dig itself 6 feet under in order to get the shot!

The most common cause of a poorly matched scaling move is a badly entered reference point or a badly set up rig. If the reference points are not entered accurately, the scaling move can be badly affected. If the move is more or less right, but not up to the required standard, enter each reference point again, taking extreme care to measure the target distances accurately, a few millimetres can make an appreciable difference especially when up close.

When you are in "Scaling move: Slave" the 3D graphs will display the slave moves for both the target and the camera. This can be useful in trying to isolate any mismatch between the master and slave moves.

13.3.3. X,Y & Z Adjust

The Slave move can be moved around in 3D space by using the X,Y&Z adjust buttons. Any entry into these boxes will move the slave move around by the distance entered in metres. If you attempt to edit a scaled move in the 3D View, any movements made with the mouse will not edit one specific waypoint, but will be applied to these XY&Z offsets and will affect the entire move.



Scale: 0.000
Scaling Refs: 1 Point
Scale to: Target

X Adjust: 0.000 m
Y Adjust: 0.000 m
Z Adjust: 0.000 m

Azimuth: 0.000 °
Elevation: 0.000 °
Roll: 0.000 °

13.3.4. Azimuth, Elevation and Roll

These rotational angles are only available in “1 point” and “Line Up” scaling modes. They are 3 rotational angles that can be used to rotate the slave move around in 3D space. They are not for the faint of heart and are for seasoned professionals only; don't try this at home. Consult MRMC if you have a specific need in this regard. The Azimuth is basically a rotation of the slave move about the vertical axis based on the 0,0,0 origin (normally this would be a vertical line through the turret of the rig when it is at zero. This can be used to rotate the move to the left or right, or even 180 degrees around. The Elevation is a rotation about the horizontal side to side axis (Y) which would be similar to a tilt but this occurs after the azimuth rotation and the axis of elevation is not affected by the azimuth rotation. If you pan and then tilt the camera, the tilt axis has moved with the pan, however, the elevation rotation is not affected like this. It gets even more complex with the Roll rotation which is like a roll with the camera at zero. There are several caveats with doing this, not the least of which is the limitation in scaling of going outside of a quadrant on a rotational axis (see below), and also if you use the Elevation or Roll rotations, then the horizon will not match from one move to the other.

Scale To

“Scale To” is a feature that allows the user to define the point about which you are scaling as the camera point or the target point. When a move is only scaled up or down, one point will stay in the same place, and everything else moves about that point. This toggle allows you to state whether it is the camera that you are scaling about, or the target.

Since you have no target in PTR Target Tracking, then one should scale about the camera. Also if you are shooting a background plate and have no target to scale about, you can also use this method of scaling when you are in normal Target Tracking.

Scale to Origin scales about the origin and is a mode initiated by the software when using advanced move editing. (See “Transform” in the section on Target Tracking graphs.)

Adjust XYZs

This button is for setting the XY&Z adjust fields such that the current camera position is the same as the Slave Move start Camera position. It simply determines where the camera should be for the current frame, where it is now and sets the XYZ fields to the difference. The main use of this would be for shooting a background plate for the slave move where you are not actually trying to line up 2 items to one another. Note that the current frame is best set by doing a "Goto" to the frame at which you want to do the adjustment.

Initing axes in scaling

When you run a scaled move, the default initialisation point for any rotational axes is 0.0. This means that if you have a move where the rotate is greater than 90 degrees, the scaled version of it will be run with the rotate set to a value that is less than 90 degrees, with the other axes moved to maintain the same camera position. This may or may not be the right solution in many cases, and you do have the option to initialize the scaled move to start at the solution closest to the entered waypoints from the move. In this case check the **Init Scaling** option in Miscellaneous Setup.

Convert Master

When you are running a slave move, the positions listed in the waypoints are the positions for the master move. With this button, you can clear out the master move, and enter the slave move as its own move with a waypoint for each position. You can then adjust the slave move around with the normal move editing features. Note that when you do this, you will be editing a move and it will no longer match to the master. Save the master move to disk before converting the slave as you will lose the master move. Note that when you do this, you will be running reverse kinematics on the Slave Cartesian Positions. The Section above on Reverse Kinematics also applies. A Slave move that has been converted to a Master may not exactly match the original Scaled Master. This is because the new positions may be in a different orientation and the curve fit may not be exactly the same.

13.3.5. Note on Cartesian Priority

When you are scaling moves up or down, the motion of the master axes may have to be different between the master and the slave move. However if you simply change the master axes positions, then you will alter the master move and thus you will not get a match between the 2 moves. The handling for this is to select Cart Priority in the Job type display once you have calculated or run the master move and then you are free to adjust the master axis positions without changing the Cartesians which will still be scaled as expected. Ensure that you save the job before and after you make changes so that no work is lost. Read the section above on Reverse Kinematics for more data on this and you must verify that the rig is properly set up.

13.3.6. Tip, Warnings and Limitations

The entering and running of scaling moves is one of the more complex parts of the program, and it does require a good understanding not only of the system, but also of exactly what you are trying to achieve. Get the move worked out in your head in terms of a 3D path, and then imagine what the resultant path will look like on the slave item and see that this move is in fact achievable. Once you have a good idea of roughly what the slave path will look like, and where it will be, then you have a much better chance of getting it right. Don't forget that not only is the move scaled up, but the distance to the target is also scaled up (or down).

No system is exactly perfect, and the accuracy of any target tracking or scaling move depends on the accuracy of the Setup of the physical parameters of the rig, and also the accuracy with which the distances are measured, and points lined up. Whilst target tracking itself is reasonably forgiving, and will not show up minor errors; scaling is very unreasonable, and will tend to show up any inaccuracies in the axes scaling or zeros. In addition, if you are scaling up a move, then the inaccuracies will be scaled up too.

The normal target tracking limitations as listed above also apply to scaling, but with a couple of additional items. You cannot turn the rotate through more than 90 degrees from straight ahead during a scaling master or slave move, the rotate has to stay between +90 degrees and -90 degrees throughout the move (or it has to stay between +90 and + 270 or -90 and -270). You cannot put multiple turns on the pan or tilt axis, they have to stay between + and - 180 degrees. If you do need to use a rotate angle that is greater than 90 degrees, then you can check the Init Scaling toggle in the Miscellaneous Setup. Setting this option can also help in similar situation with the Pan and Tilt. See “Init Scaling” for further explanation.

If you find that the slave move puts the camera in more or less the right place, but points it in exactly the wrong direction, then this may be fixable through the various settings of the “Pan Handling” toggle in the Miscellaneous Setup display. This is also helpful when importing CGI moves.

13.3.7. Snorkels

When you are in target tracking, you can use a snorkel lens in Keep Head Level and it will work as intended. This however will severely reduce your range of motion and this may not be what is needed. You will find that using target tracking out of Keep Head Level will produce better results than if you are not in Target Tracking, but it will not necessarily target track and the XYZ output data will not necessarily match the movement of the lens. A snorkel does require a Z offset in order for the computer to determine which way it is looking. Due to this, you must enter a Z offset for any snorkel lens no matter how small. You may find that if you enter 0.0 it will come back as 0.001.

Moving the “Up” axis:

If your scaled move is rolled such that the horizon plane is not parallel to the master horizon plane, then the roll axis will not work properly between the Master and Slave moves unless the master move was programmed and shot in “Roll Up” mode.

The best way to gain experience and skill with the scaling feature of the Flair program is to use it and practice on simple moves until you really understand what is happening when you shoot a scaling move, then try more and more complex moves. As you go along, try making intentional errors in your input, and see what the result is so that if it happens on a set, you are more likely to know how to handle it. Experience has shown with scaling that if it is going badly wrong, then it is usually a major setup error, or an entry error. If it is going a bit wrong, but is generally right, then it is more likely a minor Setup error such as a zero, or the rails aren’t straight etc. (more likely on portable rigs)

Also to be noted is that if you drag the move slider bar to a frame count, or do a “Goto Frame”, the rig will go to a specific frame of the slave move when you are in “Slave Move” mode. However “Goto Position” will still send the rig to the listed position in the Master move.

13.3.8. Additional Scaling Options

The pull down menu in the Rig Control view has several options: Master, Slave, Reflected, Adjusted, Adjusted Target and Inverted.

Reflected: The two Master and Slave reference points designate points on a VERTICAL mirror in which the move is to be reflected. The master points locate the mirror for the Object (Master) move and the slave points locate the mirror from the Image (Slave) points. This is to handle the likely occurrence of the rails being moved from one shot to the next.

Adjusted: In this case, the Target Tracking move is run with the designated Miniature Auxiliary axes being used to adjust the motion of the Camera and Target. The axes designated as XV, YV, ZV (in the Kinematics Setup) are used to input motion in Metres that will be added to the camera and target move. If you are in PTR Target Tracking, the Pan, Tilt and Roll axes are unaffected.

Adjusted Target: In this mode, the XV, YV, ZV additions move the Camera as expected, and the XT, YT, ZT axes are used to input motion in Metres to add to the target move if in normal Target Tracking or are used to input additional motion in degrees that is to be added to the Pan, Tilt and Roll respectively when in PTR Target Tracking.

Adjusted Move & Adjusted Target

These are powerful tools for manipulating a move. They have the advantage of being completely independent of the original motion and can be keyframed on their own without affecting the original entered waypoints. Since these points are animated, you can use them to make localised adjustments to a move and are an excellent option for adjusting an imported move to make it work for the set. If the axis you are using for making these adjustments is "Min Auxiliary type and the axis is turned on and you are in "Adjusted Move" or "Adjusted Target", you can move the rig and then store in the axis position, the adjustment required to offset the move. Note that you MUST do this when at the keyframe/waypoint you are storing in and the system's current frame is that value.

Caveat: Whilst you can scale the axis in different units, these axes are best left as Metres. In this case, ensure that you type in the correct offset. If you want it to raise 4cm, type "0.04" NOT "4.0" or the rig will try to rise up by 4 metres. When using the XT, YT and ZT in Pan Tilt target tracking, the input values are interpreted in degrees.

Inverted Move: Allows the user to define an inversion plane by the height of the camera in the slave column and the Master move is inverted about this plane. This is really the same as a reflected move with a horizontal reflection plane, but is simpler. The roll can be expected to behave correctly using this mode, its motion will be negated.

Move Miniature: This setting is for working with a real time "hanging miniature" where the miniature is in the foreground and has to follow a scaled down version of the camera move to make it appear like it is not moving in respect to the background. The reference points here are used to define the location of the miniature and also where the corresponding place that the miniature is supposed to appear. One would normally program the larger Master move on the camera and define the scale of the miniature to the camera (> 1.0 if the miniature is less than full size!), and then the Min Aux axes will move the model as needed to match the camera motion, the pan and tilt of the camera is irrelevant and you have to have set up the nodal point of the lens accurately.

Slave & Miniature: A slightly more complicated version of the above, except that the Master move programmed is on the miniature, and the Slave move is for the Camera and the Min Aux axes.

In both of these latter cases, you can add an Additional motion to the miniature by putting a move into the Min Aux axes and selecting the option “Move Miniature:” to “Yes” in the miscellaneous setup. This will result in a motion on the miniature in addition to that required to make it appear to stay in one place.

Scale Zoom: Is a secret feature only to be used by the brave and foolhardy ☺ It is unlikely to show up on your menu except by prior arrangement.

Chapter 14 **Variable Camera Speed**

Some specialised camera effects can be achieved by changing the speed of the camera during the move. The feature has been implemented into the Flair software in a flexible manner which allows the user to put this effect to full use.

14.1. Variable Camera Speed Modes

There are two different modes of variable camera speed. In one form, the speed of the camera can be made to change during the move without the speed of the move changing as it is shot. This means that a move that takes 14 seconds to shoot will still take 14 seconds to shoot, but when the exposed film is replayed, it will have a different length depending on the changes in camera speed. In this mode, passes shot with different ramps will NOT match. The other mode is to have the speed of the move change with the camera. In this mode, the speed of the shoot will be affected, but the exact same number of frames will have been exposed, so the exposed film will be of the same length and exactly match as if it had been shot at constant speed.

14.2. Speed Change and Exposure Compensation

As the speed of the camera rises and falls, the exposure has to be compensated for. This can be done by dimming or raising the lights, opening or closing the iris or by opening and closing the shutter (if the camera has one). The triggering of external lights can be done through the output triggers, and the frame counts on those will match the frame counts in the camera speed programming graph, so these can easily be aligned. (In this case colour compensation may be a problem, but that is outside of the scope of this manual). Iris compensation and shutter compensation can be achieved through use of the program's automatic compensation feature.

When variable Camera Speed can be selected in the Job Type Configuration dialog box and has the following options:

Option	**Description**
is fixed	No camera speed change.
varies	Camera speed changes with the move - no exposure compensation is performed.
shutter comp.	Camera speed changes with the move, the shutter can be used to compensate for exposure changes
iris comp.	Camera speed changes with the move and the iris can be used to compensate for exposure changes
shtr+iris comp.	Camera speed changes with the move and the iris and shutter each compensate for half of the exposure change.

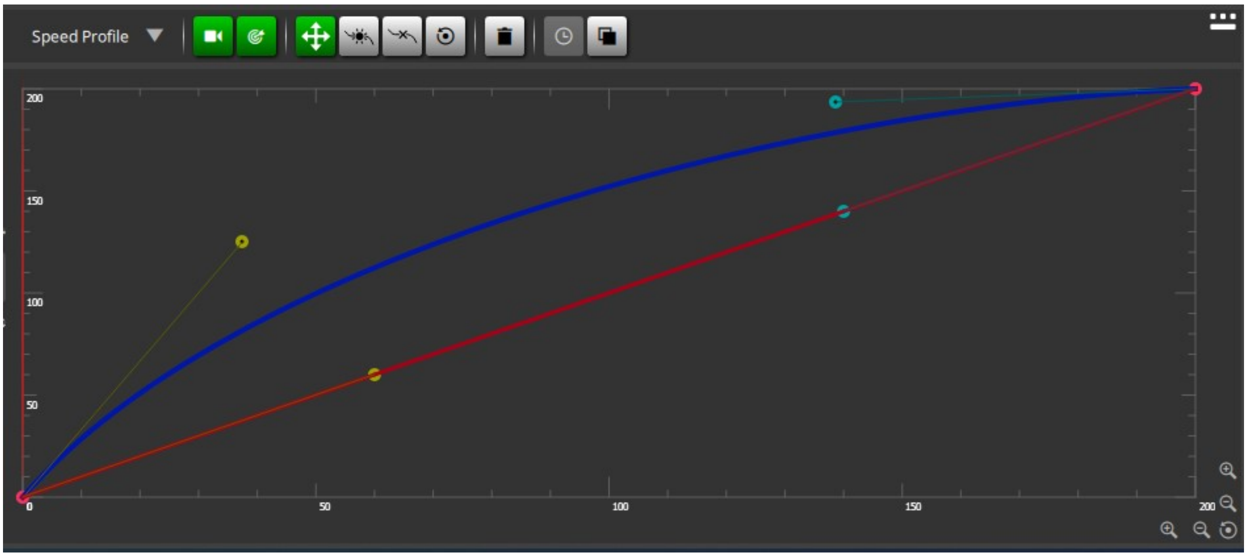
varies w/o move	Camera speed changes during the move but the move is otherwise not affected.
shtr comp w/o move	As above but the shutter compensation
iris comp w/o move	As above but the iris compensation
shtr+iris comp w/o move	As above but with the iris and shutter each making half of the required compensation
Move varies w/o Camera	Move varies but the camera speed does not vary

When one of these (apart from “is fixed”) has been selected, you can go into the camera speed editing graph and implement the desired changes.

14.3. Camera Speed Editing Graph

The speed of the camera is programmed through editing a graph of speed against frame count. Before the camera speed is edited, the move should be fully worked out and checked at the fastest speed of the camera to make sure that the move can be executed. Then set the camera speed to the speed at which you want the move to start and in the Graph view, select Speed Profile to enter the Camera Speed Editing graph.

The screen displays frame counts along the horizontal axis and camera speed up the vertical axis. The slowest speed is 0 and the highest speed depends on the highest camera speed your camera can run as set up in the “Camera Axis Setup” display. (Setting this lower can make editing the camera speed graph easier). The camera speed itself is represented by a series of dots indicating camera speed waypoints, and circles indicating beginning of ramps, these are joined by a blue line showing the camera speed at any instant. The points on the graph depict the frame count at which the camera will have completed its speed change, this allows the user to simply position a point at a frame and speed, and know that the camera will be at that speed by that frame count - ramps are applied backwards from the camera speed points.



The Control Buttons at the top of the screen provide the following features:

Zoom: The mouse scroll wheel allows the user to zoom into one area of the graph to provide more accurate editing/display of that section. Simply point your mouse at the section that you want to zoom in on and scroll the wheel up to zoom in or down to zoom out.

Display camera speed profiling curves 🟩**:** Toggle button to display or hide the camera speed profiling curve. When this button is active, the camera speed profiling curve displays in blue.

Display target speed profiling curves 🟩**:** Toggle button to display or hide the target speed profiling curve. When this button is active, the camera speed profiling curve displays in red.

Allow the moving of speed profiling curves 🟩**:** Use the left-hand mouse button to click and hold and move it. You can also use this button to allow you to alter the ramp which the camera uses to change speed. Generally these ramps are set by the maximum acceleration of the camera. Ramps can be lengthened by using the left hand button to move the ramp point (yellow circle) further to the left. If a slope is edited to be quicker than the camera's maximum acceleration, then it will drop back to that maximum rather than adopt the new incorrect ramp. The middle mouse button allows you to reset the slope at any given camera speed point back to the default which is the current maximum camera acceleration (Not yet implemented).

The waypoints can be edited in time and speed except for the first and last waypoints which have a fixed time.

Slope

The right hand button will reset the graph to a constant speed at the currently set camera speed, deleting all entered points.

Add speed profiling curve points: This button will cause a waypoint to be added at the current position of the mouse click

Delete speed profiling curve points: When this button is active, you can click a waypoint to delete it (not first or last).

Reset selected speed profiling curve point:

Reset speed profiling curve on selection: You can reset the camera or target speed profiling curve back to the default which is a constant speed at the current camera speed. Click ■ on the editing buttons in the Graph view and click the left hand mouse button on top of the graph. Note that the camera MUST be in forwards (fwd) not reverse when you do this or the graph will be off screen!.

Notes

- Velocity checking of the move is adjusted to compensate for change in camera speed and acceleration; however moves should be checked in as much as possible at the highest intended camera speed to eliminate standard over speeds caused by moving too fast. Errors then produced will mostly be due to changing the camera speed too quickly.
- If you make the camera change speed too quickly, it will attempt to execute the change you have programmed. This will most often be caused by putting two points of different speed too close together in time. In many cases you will get away with it as the camera can often change speed faster than you would think. However it is up to you to ensure that no damage is done to the camera on extreme speed changes.
- Very slow camera speeds (for example, 0) will not work if the move changes speed with the camera as the slower the move goes, the slower the camera changes speed, and everything will basically grind to a halt.

Copy camera path to target:

Frames Snap & Speed Snap

With a long move, it can be difficult to set the camera speed point at the exact position required. The “Snap” feature has been added to make this easier. This is modelled on layout programs that cause the positioned point to "snap" onto the nearest set unit of distance.

The speed and frames snap are used in exactly the same way: you either type in the exact speed and position you want and edit or add the point; or you can enter the nearest round number to snap to (for example, 10 Frames or 5 FPS), and then when added or edited, the point will only be at exact multiples of the entered values. These boxes can also be blanked out with the backspace key, and this will disable editing of that axis. (The result is undetermined when adding a point). When editing with the snap feature enabled, the point cursor will move around only to legal positions, thus clearly showing where the edited point will be.

Timings

This button produces a report on the intended move as currently programmed, stating the different frame counts at which speeds are achieved, the time taken in real seconds to get to that point, and the speed that is attained. This can be used to plan and trigger events to coincide as desired.

Programming the Compensation

Exposure will be adjusted based on the initial programmed camera speed and the initial entered position of the compensation axis on the first line of the move. That is to say that if the camera starts at 25 FPS, and the first line of the move has a shutter position of 90 degrees; then if the camera slows by 1 stop, the shutter will open to 180 degrees; and if it speeds up by 1 stop, then the shutter will close to 45 degrees. All other programmed positions of the compensating axis are totally ignored.

If the user plots a stop change that cannot be compensated for (e.g. 200 degrees of shutter or F stop of 0.2), then the move will be rejected in the same fashion as a standard move if it exceeds the limits of travel of an axis. Also if the axis has to move faster than its programmed maximum speed or acceleration to effect a speed change, then a similar error will be reported. In this case, the ramp of the camera could be extended, or the change in speed could be reduced.

Of course it is also totally possible to do the exposure compensation manually, but then the user will have to enter all positions for the compensating axis, including holds during constant speed sections of the move.

Shutter Compensation

When shutter compensation is selected, the computer will attempt to adjust the position of the shutter so as to keep the exposure on the film the same. In order for this to work, the system has to have an axis driving the shutter itself, and that axis has to be correctly scaled so that the axis read out is in degrees of shutter, with 0 being closed and 180 (or 170 etc.) being fully open. The relevant axis must also be selected as “Shutter” type in the Axis Setup dialog box (see Chapter 5 Setups for more details).

Iris Compensation

As with shutter compensation, the computer in this mode will attempt to move the Iris to compensate for the stop gains or losses. There must be an axis driving the iris itself, and it must be selected as “Iris Type” in the Axis Setup dialog box (see Chapter 5 Setups for more details). Additionally, the iris must be correctly linearised. With an iris ring, motor motion does not translate directly into F (or T) stops and thus the relationship between these has to be defined to the system in order for this to work. The exact procedure for setting this up is covered in Setting up a Lens on page 38.

Once correctly set up, the iris must also be selected as operating in “F Stops” by selecting that option in the “Job Type Configuration” dialog box. After that the procedure is as covered above.

Shutter & Iris Compensation

Sometimes too large a compensation is required to be achieved by just the iris or just the shutter; also the change in motion blur or change in depth of field can detract from the shot. In both of these cases, the compensation can be split between the iris and the shutter so that a greater range can be achieved, or the same range, but with lessened change in depth of field or motion blur.

Running the move

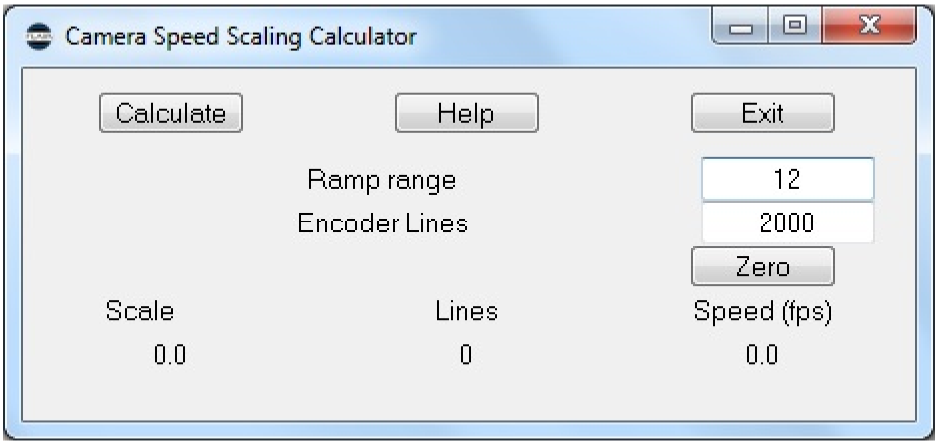
Once the move has been entered and the variable camera speed has been profiled, you can run the move as per normal. The speed checking will check the move, but it is recommended to try them out at a lower rate if you have a particularly fast move. In order to facilitate this, when you are in Variable Camera Speed mode, the FPS box on the run control bar becomes semi-inoperative, and if you click on the “Move FPS” label, it will change to “Move %” and the box below will be the % speed that you are running the move. I.E. you can try the move out at 50% and it will only get up to half of its normal full speed. You may incidentally try it out at up to 200%. If you click this back to “FPS”, the percentage automatically reverts to 100%. At the slower speeds, the iris and shutter compensation is non functional, this feature is for testing the move only.

Pre-holds and Post-holds

The feature that allows you to put pre-holds and post-holds on the move is operational with variable camera speed as is the feature that allows you to start the camera before the move. However you may not stop the camera during a vary speed move, nor start it.

Mimicking Variable Camera Speed

This is a mode in which you can control the camera speed as a mimic axis. The control is similar to the positional control of an axis, except you are changing a speed not a position with the controller. The scaling of the mimic controller is set up using the Camera Speed Scaling Calculator:



Ramp range 12
Encoder Lines 2000

Scale	Lines	Speed (fps)
0.0	0	0.0

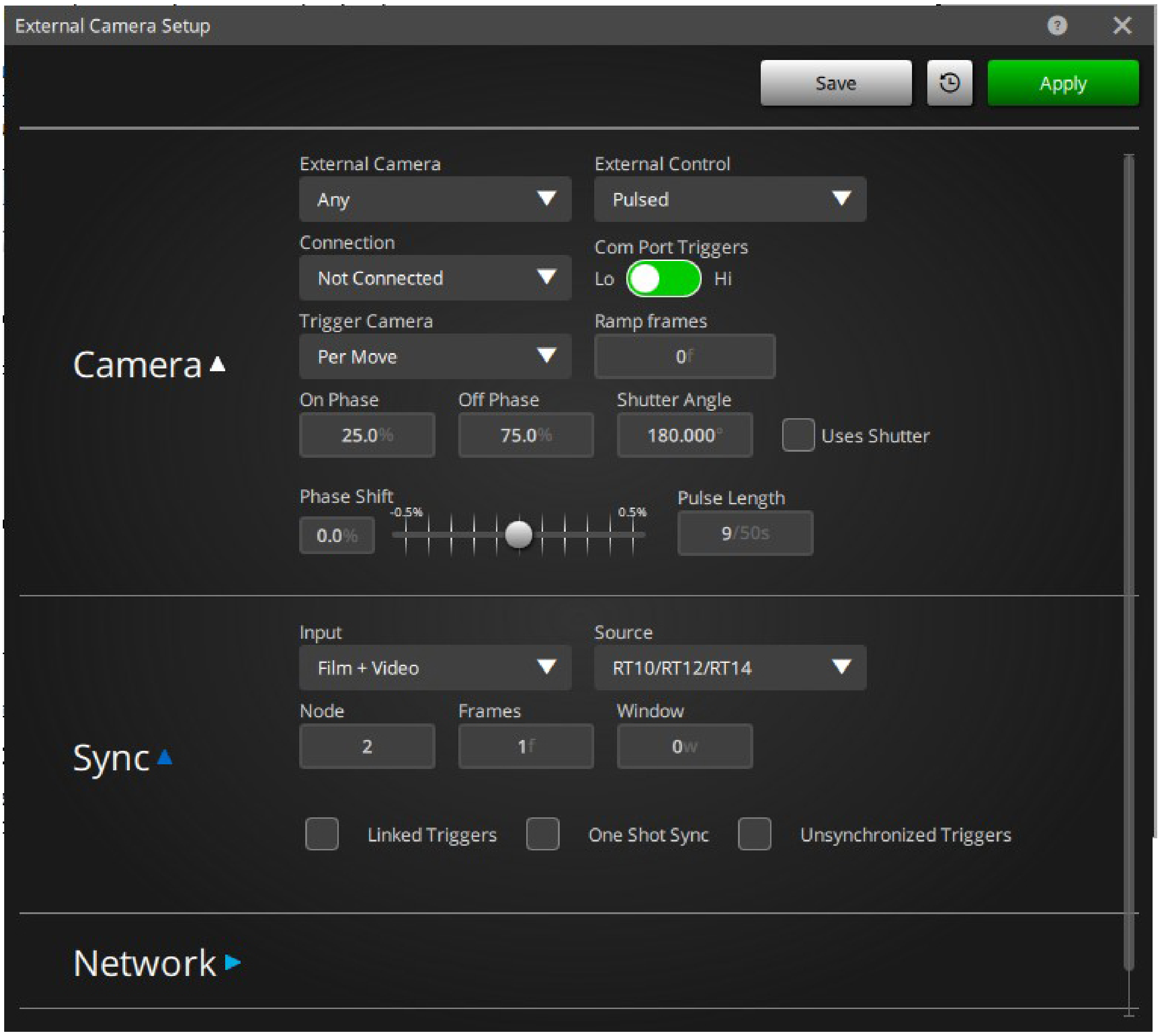
The Ramp range is the amount of speed change you want from one end of the controller's motion to the other. It is generally a good idea to make this as small as possible so that the camera speed change is as smooth as you can make it. High changes in camera speed will cause the rig to trip out and the shot to be lost. Once you have set the number of encoder lines and the desired range, pressing the “Calc” button will work out the scaling and then the display at the bottom with show you the change in camera speed as you move the controller.

The zero position of the controller is very important as every time you switch Mimic Learn on, the controller will be zeroed out and changes from its zero location will increase or decrease the camera speed and also affect the shutter and/or iris. Like a normal mimic axis, you can learn and replay speed changes if they are working as desired. Speed change cannot currently be smoothed or otherwise adjusted.

Chapter 15 **Synchronisation**

15.1. Introduction

The basic concept involved in synchronisation is to get the system to run at the same speed as an external camera so that moves can be matched even though the recording source is not fully under the control of the system. This includes video cameras as their recording rate is not under the computer's control. The External Camera Setup allows you to control this interface.



External Camera Setup

Camera ▲

External Camera: Any
External Control: Pulsed
Connection: Not Connected
Com Port Triggers: Lo ● Hi
Trigger Camera: Per Move
Ramp frames: 0f
On Phase: 25.0%
Off Phase: 75.0%
Shutter Angle: 180.000°
□ Uses Shutter
Phase Shift: 0.0% (-0.5% … 0.5%)
Pulse Length: 9/50s

Sync ▲

Input: Film + Video
Source: RT10/RT12/RT14
Node: 2
Frames: 1f
Window: 0w
□ Linked Triggers □ One Shot Sync □ Unsynchronized Triggers

Network ▶

15.1.1. Lost Synch

In earlier versions of the software, a synchronised move would stop whenever the system detected that the camera was out of phase with the move by more than a certain level, or if it was running at a speed significantly different from that of the move. Given that most modern cameras are crystal locked and the Flair system is crystal locked, the most common reason for the appearance of this could be noise on the cable connecting the camera to the Flair system. It was decided therefore that if the synch is lost, the move will continue along, but a warning box will be put onto the screen to let you know that synch has been dropped. It is then up to the user to decide whether or not to cut the shot depending on the circumstances.

15.1.2. Connections

There needs to be some electrical connection into the root box consisting of either a standard video signal being plugged into a BNC in the case of video, or some wiring into the input triggers in the case of a film camera. The technical data on the triggers should explain how to make the correct connections, if there are any problems, contact MRMC for assistance. You may also synchronise to a signal coming in to an Ulti-Head. In this case, the Sync Source is switched to be the board number of the Ulti-Head. For the standard Modula Rig, this would be Board 2 where Board 1 is running the Track, Lift and Rotate. You also would have to state if you are using a Link style Axis Board or a Network Direct style axis board.

15.2. Phase Shift

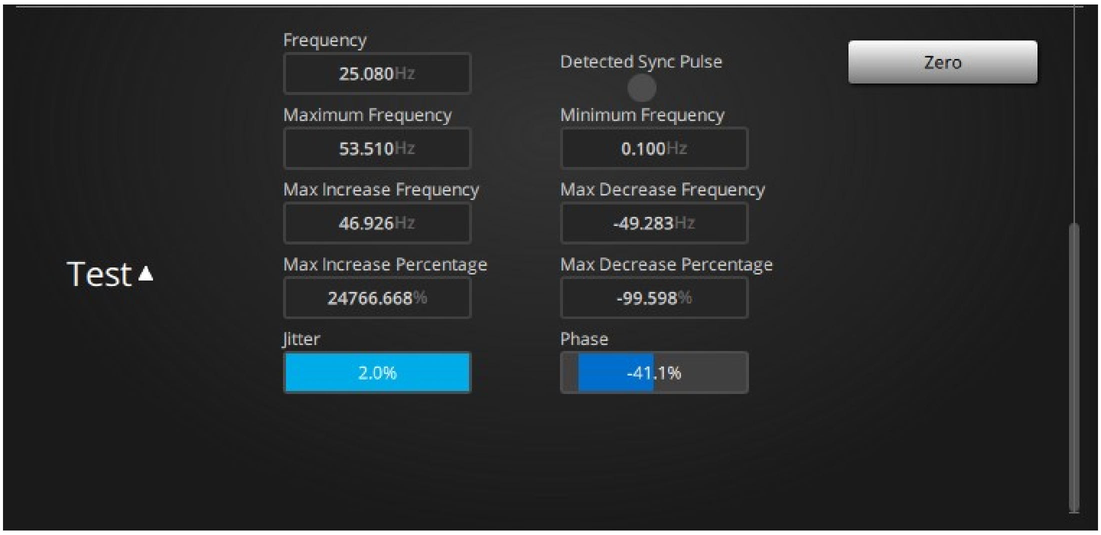
While the computer is maintaining the same speed as the incoming pulse, it also maintains the same relative time relationship to it. It's like having 2 cars driving down the motorway, if they are driving at the same speed, then the distance between them will not change. The phase shift is how far apart they are, and with this software, it can be adjusted to have one car 1/2 length up on the other, or 1/2 length behind. The practical upshot of this is no matter where in the frame the sync pulse comes, you can adjust your system to synchronise correctly to it. Also if you are shooting a TV screen, you can adjust the synchronisation phasing until the sync bars are out of the safe area. (Given that the camera has sufficient shutter angle to be able to expose a whole field).

15.3. Stability

Synchronisation will work with almost any speed up to 150 FPS as long as the incoming pulse is stable. If the incoming pulse varies by +- 5% then the sync will not be stable, and it is possible that the move will become rough as the computer tries to keep pace with an erratic incoming pulse.

15.3.1. Testing the incoming frequency

If you want to check the incoming sync pulse as to when it is actually arriving, and also at what frequency, there is a status box in the External Camera Setup display. Select **Setup** → **External Camera** to open the External Camera Setup dialog box. Select the “Test” button on this display and a new box will pop up showing the current incoming frequency in the Source Frequency box, and the label to the right of this will say “Sync Pulse” if pulses are coming in and “No Sync Pulse” otherwise. In order to assist with determining what may have caused or be causing timing problems with synchronisation, there are boxes which inform the user of the maximum detected frequency and the minimum detected frequency as well as the deviation in cycles per second and percentage. These values can be reset through use of the “Zero” button and you can see if there is significant variation in the detection of the synchronisation pulses. A percentage deviation of 5% would cause the system to drop synch.



Frequency: 25.080Hz

Maximum Frequency: 53.510Hz

Minimum Frequency: 0.100Hz

Max Increase Frequency: 46.926Hz

Max Decrease Frequency: -49.283Hz

Max Increase Percentage: 24766.668%

Max Decrease Percentage: -99.598%

Jitter: 2.0%

Phase: -41.1%

15.4. Running an External Camera

This option is labelled “External Camera” and has five options: “None”, “Any”, “DSLR on S3”, “DSLR on Com”, “DSLR on Dragon”. When “Any” is selected, then the system will attempt to run an external camera whenever a move is run. The main difference when this switch is on is that the system will not complain if you are in record but the internal camera is switched off. If the internal camera is on, it will also be run, but the external camera is assumed to be the more important. Also note that when this option is selected and the “Turnover” button is pressed, the system will run the external camera in preference to the internal camera.

15.5. External Control

One of the trigger outputs on the root unit is dedicated to running an external camera, and also you can specify an output trigger on one of the axis boards to run an external camera. The trigger can either be pulsed to start and stop the camera (”Pulsed”) or can be held continuously high during the time the camera is supposed to run (“Continuous High”) Most cameras operate with the “Continuous High” mode. In the “Pulsed” mode you can also specify how long the pulse is. Normally a single tick (1/50) of a second is enough, but some systems such as video camcorders do not detect this and need a longer pulse. Consult the camera documentation or MRMC if you have any questions. The axis board output trigger is selected through the “Board” and “Port” entries in this display, this is not normally wired to any specific output and you should contact MRMC for any assistance.

15.6. How to synchronise:

15.6.1. Before the move starts

When you press the button to shoot the move, the system waits for a few frames to allow the camera to get up to speed. This is set up in the External Camera Setup display by the box “Ramp Frames”. If your camera starts slowly and needs some time to get up to speed, then this value can be changed to allow that to happen. With a video system this value can be set to 0, and the step will be missed out. After the ramp frames value is completed, the system will start locking sync with the incoming pulses for as many frames as you have set up in the "Sync Frames" value, this is really just a buffer, and need only be a couple of frames. During this time the Status Bar will display "Locking Sync".

15.6.2. Preroll

If any pre-roll is required on the move, then this will occur immediately after the sync frames, and the axes will be moved back and then into the move with the computer keeping sync to the incoming source.

15.6.3. During the Move

As the move runs, the computer will detect every incoming pulse, and maintain the internal speed of the system to match that incoming pulse. Even if the pulse changes speed slightly throughout the move, the system will keep track with it, and will maintain the same relative phase shift to the pulse. This means that moves shot using this feature should be able to be matched frame for frame.

15.6.4. Synchronisation loss

If the timings between the system and the incoming pulse vary by too much, an error will be reported. There are two possible errors: either the frequency of the incoming pulse has varied too much in which case the system will report that synchronisation has been lost, or if the phase relationship between the system and the incoming pulse varies by more than 1/4 frame, then the system will state that Phase Lock has been lost. If this happens, then it is likely that the frequency of the sync pulse is not stable, and the computer is having trouble locking in to it. It is also possible for the sync pulse to stop completely, in this case, the computer will wait a reasonable time for it, and then will post an applicable message.

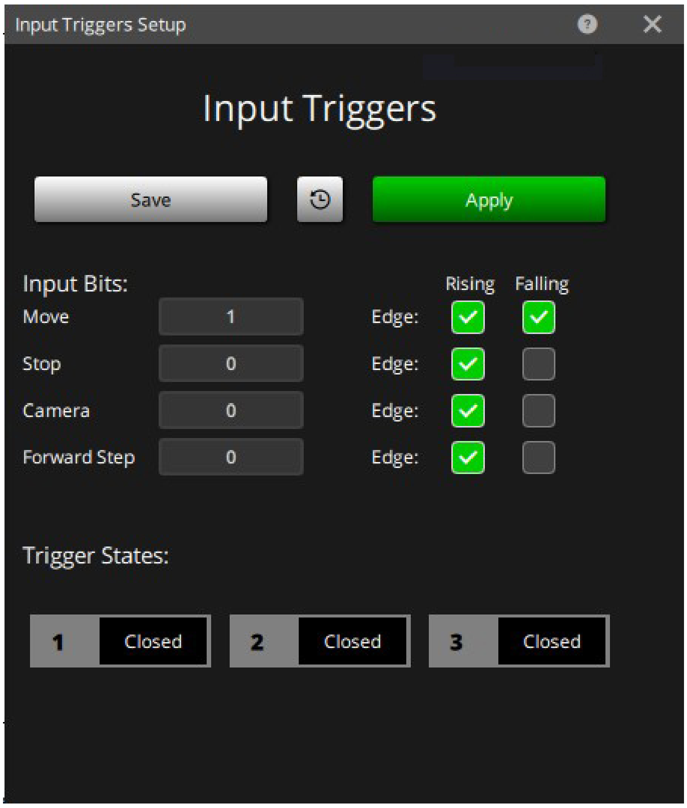
15.6.5. Adjusting Phase Shift

If you need to adjust the Phase Shift between the move and the external source, then you can do this as the move is running. The simple way is to put in a long hold, and shoot that. While it is running, adjust the Phase Shift slider in the External Camera Setup display until the correct phase shift is achieved, and then use that phase shift value for your moves. When shooting a video screen, you can either look through the gate and get the bars out of frame, or if you have a revolving mirror, look through the viewfinder and get the sync bars in frame. Once found, a phase shift value should not need to be changed, but it is always a good idea to double check before you start exposing any film. Note that the sync lock does not work when you are using the "Turnover" feature, you have to actually be shooting a move. Note that you can often see sync bars on a video assist screen which have to do with the phase relationship between the video you are shooting and your video assist – so don't use a video assist when setting this up.

Chapter 16 **Inputs, Outputs and DMX**

16.1. Input Triggers

There are three separate inputs that can be used to trigger the start of the move, the camera to start running or the move to step forwards 1 step. These can be wired up to lighting desks or to time code readers or simply to a push button if you wish to trigger one of these events from a separate source. When the button is activated, the action will start in the next 20 milliseconds, so an accurate move start is accomplished.



Input Bits:

			Rising	Falling
Move	1	Edge:	☑	☑
Stop	0	Edge:	☑	□
Camera	0	Edge:	☑	□
Forward Step	0	Edge:	☑	□

Trigger States:

1	Closed	2	Closed	3	Closed

16.1.1. Input Bits

You can select an input to be active by setting it up in the Inputs Triggers Setup dialog box. Enter the number of the input into the relevant box, and then when the function is valid, if that input is activated, then the function will be triggered. There is a row of boxes across the bottom of the Input Triggers Setup display which correspond to the three inputs, and you can test and identify your inputs by watching these boxes as they change state with the inputs. If you do not wish any of the inputs to be active, then set the Input Number to 0.

Move Input Bit: If a valid edge is detected on this input, when the system is at Ready to Shoot, then the move will be started.

Stop Input Bit: If a valid edge is detected on this input, when the system is Shooting the move will be stopped.

Camera Input bit: A valid input here will cause the camera to start rolling when you are at “Ready to Shoot” and the camera is engaged and not in Rehearse. If you want to use this trigger to start a Film Camera configured as an axis, you must check that the take ups have been tensioned, the camera is engaged, not buckled and not in Rehearse.

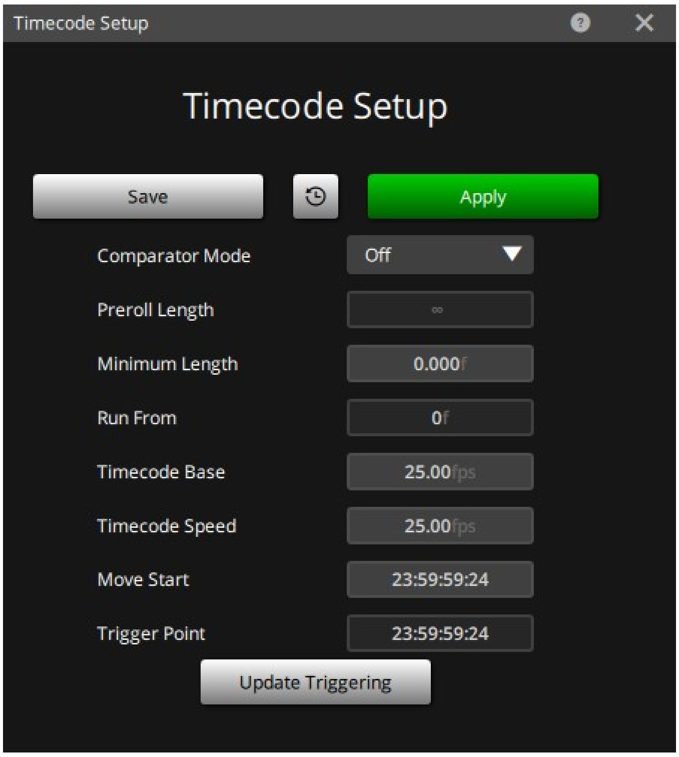
Forward Step Input Bit: Used to trigger the system to take a step forwards along the move path. This will only work when the system is at “Ready to Shoot” or “Ready to Step”.

Edge

A voltage level can rise, fall or pulse. Each input can be set to obey only a rising edge, only a falling edge or both. The button at the bottom of the screen tells you which input is in which state. “Open” indicates a low state and “Closed” indicates a high state.

Note that the inputs to the root box have logic in them to ignore switch bounces, and this can make the system miss a state change if it is preceded or followed by switch bounce. Use de-bounce circuitry for any mechanical switches or relays.

16.2. Timecode Setup



Comparator Mode: Off

Preroll Length: ∞

Minimum Length: 0.000f

Run From: 0f

Timecode Base: 25.00fps

Timecode Speed: 25.00fps

Move Start: 23:59:59:24

Trigger Point: 23:59:59:24

Save: Saves the current set up to an ASCII file in the “/configs” directory called Timecoder.ini. This file has the same format as other .ini files, and can be edited by the user. It is read only when the program starts up.

Apply: Sends the current settings to the timecode device.

Exit: Closes the display.

Comparator Mode: The options are “Off”, “TC Comparator” and “Store Start TC” In Comparator mode, the timecode that is running is compared to the requested start timecode and if there is a match, the move is triggered at that point. “Store Start” means that when the “Shoot” button is pressed, the timecode at that time is stored and may be used later to trigger the move at the same time as it was manually triggered.

16.2.1. Preroll Length

If your move requires there to be a preroll before the move actually starts, then this will occur after the input has been activated, and the actual move will not start immediately. In this case, once the move has been calculated and is ready to shoot, get the system out of a run state by pressing the Stop button and go into the Timecode Setup dialog box. There is a field that displays the amount of required preroll in frames, and this can be used to set up the correct triggering. Most moves however have very little or no preroll. If the move starts from rest the only axis that prerolls is the camera, this is usually quite short but is displayed as above if needed. This preroll is calculated in the system as a number of 1/50's of a second, but it is displayed in frames at the currently set camera speed. Therefore this number may not be a whole number of frames.

16.2.2. Run From

This is also an output field and reminds you of the frame count at which you are starting the move.

16.2.3. Timecode Calculations

A Timecode calculator allows you to calculate the correct timecode point to trigger a move. When you are recording something that has to match to a timecode (such as a music video), then several laborious calculations have to be done to make sure everything matches. If for example you do one pass starting at the beginning of the move at time code 00:01:00:00 then if you part run the move from frame 347, you have to convert 347 frames into a timecode, subtract the rig preroll times in frames from that time code and then add it to the start timecode, working in base 25 and base 60 unless you are shooting at 30 fps which needs different calculations!!

16.2.4. Timecode Base FPS & Timecode Speed FPS

Timecode is a stream of numbers representing frames, seconds, minutes and hours. There are at least four different formats for this, each one having a different number of frames in a second. The most common are 24, 25, 29.97 and 30. It is important to fully understand this when you are dealing with timecode as often the other people you are working with do not understand it, and you could be misled (for example, in the U.S. timecode for songs is almost always 30 FPS, but is sometimes referred to as 24 since the camera runs at that speed!). The Timecode Base FPS is the number of frames that make up a second and is normally one of the above speeds. The Timecode Speed FPS is the speed at which that timecode is being played back. If you make a normal 30 FPS timecode, and play it back twice as fast, it is still a 30FPS base timecode, but it is running at 60. These numbers are vital in correctly figuring out the correct trigger point for the move.

16.2.5. Move Start

The user enters the timecode at which the move should start into the “Move Start” timecode box. Once this has been done, whenever the move is calculated at the correct speed, the computer will do the calculations for you and the timecode at which the rig should start its preroll will be displayed in the “Trigger Point” timecode box. Even starting at the beginning of the move, the rig needs some time, if only to get the camera up to speed. Above on the display, the computer also tells you the intended start frame of the move you are shooting (usually 0, but can be later for part runs).

This data is only valid when you have given the computer all the data about the move. If you later change a camera speed, or switch the camera into or out of record, then the data it gives you will not be accurate and you have to calculate the move and go back into this display to get the now correct data. (Calculating the move would involve telling the computer to run it with a forward run or part run or using the “Check Move” button, and allowing it to present you with "Ready to Goto" or “Ready to Shoot”) If you leave this box on screen while you change things, the data will not be updated, you have to call up this box after all the prerequisite steps have been taken.

If you are starting the camera before the move by using the “Turnover” button, then the rig will not need the camera preroll time when you trigger it. In this case you should process the move with the camera in “Rehearse” to get the correct trigger point, and then switch it into “Record”. This is more complex, and I would not recommend it unless you have pressing reasons for doing so. It is also prone to be slightly less accurate in terms of matching the move to the timecode exactly. It could be up to 1 frame out of sync at normal timecode speeds.

If you are doing ANY form of complex timecode work with off speeds or other variations, you should always have some form of backup check to make sure that the move is working in sync with the sound as you expect, as an error here can be hard to detect and impossible to correct afterwards. The calculator here is quite complex, and even a slightly incorrect setting will cause a poor result.

16.2.6. Trigger Point

This is the calculated value of the timecode at which you should trigger the move to ensure that frame 0 will line up with the timecode entered into the Move Start box.

16.2.7. Update Triggering

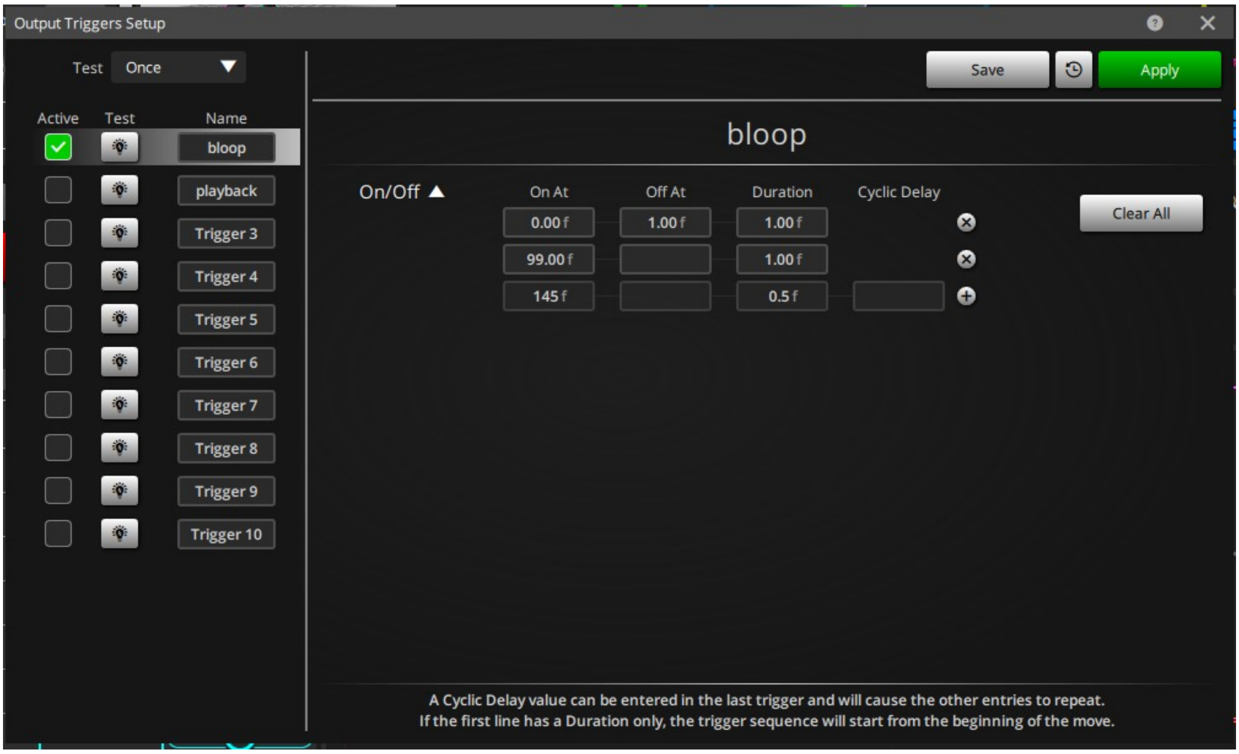
If you change the trigger point or the frame from which the move is going to be run, use this button to verify that these changes have been taken into account in the timecode and trigger calculations.

16.3. Output Triggers

16.3.1. Introduction

The user has control over 10 output triggers. These can be controlled to change state during the move at exactly repeatable times. For each Output (called a trigger), the user can change the name of the trigger, whether the trigger is active or not, and at what frame counts the trigger goes on and off. There are two columns provided with and option to add up to 15 entries in each. The user can switch any trigger on and off up to 15 times within the move. The frame count (fractional if desired) at which the trigger is to turn on is entered in the left hand column and the frame count at which the trigger is to turn off is entered in the “Off” column. Alternately if the “Off” column is left blank, you can enter the duration that the trigger is to stay on for in the “Duration” column. This is very convenient when retiming triggers as you don’t need to adjust the “Off Frame” every time you make a change.

The move can be shot forwards or backwards, or part run. Moving Step will work, but Stop Motion will not operate the triggers.



Test: Once

Active	Test	Name
☑		bloop
□		playback
□		Trigger 3
□		Trigger 4
□		Trigger 5
□		Trigger 6
□		Trigger 7
□		Trigger 8
□		Trigger 9
□		Trigger 10

bloop

On At	Off At	Duration	Cyclic Delay
0.00 f	1.00 f	1.00 f	
99.00 f		1.00 f	
145 f		0.5 f	

A Cyclic Delay value can be entered in the last trigger and will cause the other entries to repeat.
If the first line has a Duration only, the trigger sequence will start from the beginning of the move.

There is an additional way to use the triggers to produce a repeating series. Program in the repeating series you want as an on and off time, leave the next On At blank, but program into the off time at what frame count you want the series to repeat. A series of numbers: on 1.0, off 2.0 followed by a blank on, and then 3.0 in the off column would cause a trigger to turn on and off for one out of every three frames. This is called a Cyclic Trigger.

The “Insert” and “Delete” keys can be used in the triggers display to insert or delete line entries, and the subsequent entries will be shuffled up or down to suit.

Triggers may be caused to occur before the actual move by programming in a negative frame count for the trigger. When the shoot button is pressed, the system will go into a preroll state and the trigger will fire at the appropriate time before the move. When a move is run backwards, the triggers before the start will occur after the system has run back to the beginning. These triggers outside of the move will cause preroll and postroll conditions for the system, when no axis is moving and this may not be desirable whilst testing a move. In this case turn triggers off (or that relevant trigger off) and this will be remedied. If you get strange long prerolls and postrolls on a normal move, it could be that you have some trigger from a previous move that has not been switched off.

This feature’s performance is undetermined when synchronising to an external camera, or when starting the camera before the move. Note that when you are synchronising with an external camera, the timings of the system are changed to mesh in with the camera, and due to this, a trigger starting at frame 0.0 MAY be missed. In order to avoid this, you should not program any triggers that start or stop within the first few frames of a move.

You cannot combine Cyclic Triggers with Triggers before the move. Cyclic triggers only work within the actual move time, not within the prerolls or postrolls. Also note that Cyclic triggers are processor intensive and therefore it is inadvisable to enable too many Cyclic Triggers at once, as the processor may run out of time to do all the required move calculations.

Name: The name of the trigger can be changed. The triggers are normally simply labelled Trigger 1, Trigger 2 etc., but if you have a trigger wired up for a specific purpose you can change its name for example, call it "Flash Gun" or "Tape Start" etc.

Active: This toggle controls whether the trigger is active or not. When set to off, the output trigger will not activate during the move.

Test 💡: Can be used to turn the trigger on and off manually. When pressed the trigger should activate and deactivate upon release.

Frame Counts: The user can switch any trigger on and off up to 15 times within the move. The frame count (fractional if desired) at which the trigger is to turn on is entered in the “On At” box and the frame count at which the trigger is to turn off is entered in the “Off At” box. The move can be shot forwards or backwards, or part run, and the triggers will operate at the expected time. Moving step will also work, but stop motion will not operate the triggers. In order to blank out fields, select the box in the standard way and press the backspace key.

You can set up a trigger to fire before the start of the move by entering a negative frame count. If a trigger is set to fire before the move, then the preroll for the move will be extended to at least as long as the earliest trigger. The length of this preroll will be displayed in the Input Setup once the move has been calculated (see above). This can be used to provide a consistent preroll for a move regardless of whether the camera is on or off.

Cyclic Triggers:

Any trigger can be set to repeat in a cycle throughout the move. Enter the first On and Off Frames, defining the duration of the trigger. Leave the next On Frame blank, and enter trigger repeat frame into the next Off Frame box. For example to enter a trigger that flashed during exposure time, set the first On Frame to 0.25, the first Off Frame to 0.75, and set the second Off Frame to 1.0. For a trigger that flashes once every 10 frames, enter 0 and 1.0 in the first line, and enter 10 into the second Off Frame as the trigger cycle length.

Another way to add a cyclic trigger is to a enter a “Cyclic Delay” value in the last trigger. This will cause the other entries to repeat.

These triggers work forwards and backwards, but do not work outside of the normal move, that is during preroll or postroll. Note that cyclic triggers do involve considerable processing, and if many are used with a Target Tracking move, it is possible that the system will run out of time to do the move calculations. If this happens, reduce the number of cyclic triggers you are using. It is unlikely that this will occur.

16.4. Triggers On/Off

There is a switch which allows you to turn all the output triggers on or off. This button is in “Control view” and toggles between “Triggers On” and “Triggers Off” and refers only to the output triggers. This feature has been added so that physical effects which you may want to fire at specific frames do not keep turning on and off when you are rehearsing the move. The default state is “On”. Note that with a High Speed Robot, the triggers do NOT work in “Slow” mode, only in “Fast” mode.

The first trigger on the system is dedicated to operating an external camera if your system is set up for one. If the option “External Camera Setup” appears on your

Setup menu, then the first trigger is assigned to start and stop the external camera. If you are syncing to an external camera, then this trigger will operate to start the camera. The system will wait for the camera to get up to speed and then the move will be shot. This trigger will stop at the end of the move, and that should stop the external camera. More data on this trigger is available in the section on External Camera Setup in Chapter 5 - Setups.

Unlike other triggers, this trigger is operational when you are stepping. If you have set the system up to have an external camera and it is active, then you cannot really sync to it if you are single stepping in stop motion. However, this trigger will go active for a frame every time you step. This can be used to trigger an external frame store device etc. If you are stepping more than one frame per step, the trigger will go on for a longer time. Also note that if you do have the internal (camera as an axis) camera running at the same time, this trigger will activate AFTER the internal camera has taken a frame.

With an RT-14 unit, this trigger is propagated down the axis pipe to any Ulti-Board, and will cause the external camera trigger on the Ulti-Board to fire at the same time.

16.4.1. Wiring

Each of the outputs works as an independent switch closure, with both sides of the switch being presented to the user, this means that the computer can be totally isolated from the thing that is being triggered, and wiring errors will not affect the computer adversely (though they can in extreme cases damage that output). Contact Mark Roberts Motion Control if you require any assistance, if you have specifically stated that you are going to use the outputs, a technical data sheet will have been supplied to you.

Refer to Appendix at the end of this manual for related pin-out information.

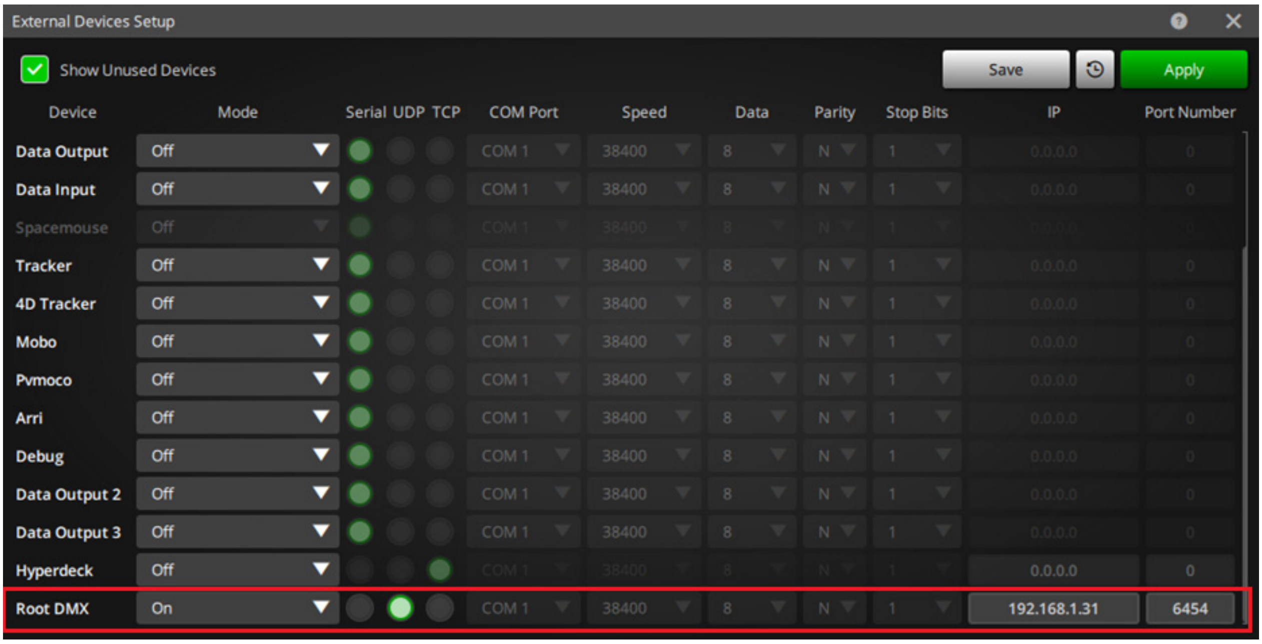
16.5. DMX Setup

Running the DMX device requires a single line to be added in the Flair.ini file.

```
*DMXControl: Root
```

The DMX can be setup in 2 ways, from External Devices Setup or directly from DMX Setups.

For setting up from External Devices Setup, go to Top Menu > Setups > External Devices. Scroll down to the bottom of the setup window and you will find Root DMX. Turning it on will enable the IP address and Port Number input field.



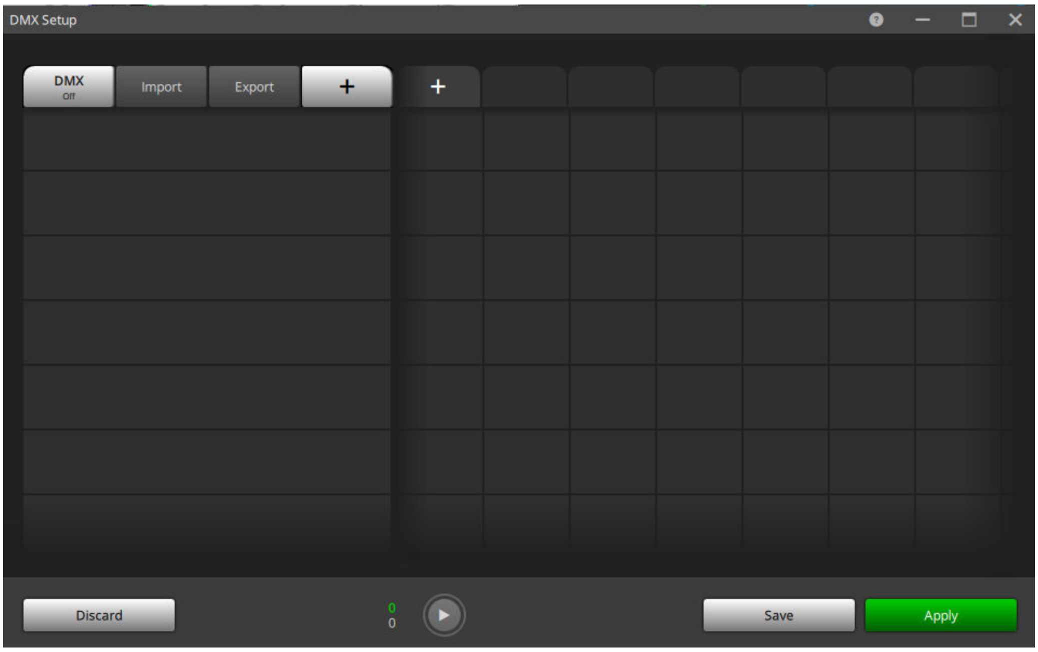
External Devices Setup

☑ Show Unused Devices

Device	Mode	Serial	UDP	TCP	COM Port	Speed	Data	Parity	Stop Bits	IP	Port Number
Data Output	Off	●	○	○	COM 1	38400	8	N	1	0.0.0.0	0
Data Input	Off	●	○	○	COM 1	38400	8	N	1	0.0.0.0	0
Spacemouse	Off	●	○	○	COM 1	38400	8	N	1	0.0.0.0	0
Tracker	Off	●	○	○	COM 1	38400	8	N	1	0.0.0.0	0
4D Tracker	Off	●	○	○	COM 1	38400	8	N	1	0.0.0.0	0
Mobo	Off	●	○	○	COM 1	38400	8	N	1	0.0.0.0	0
Pvmoco	Off	●	○	○	COM 1	38400	8	N	1	0.0.0.0	0
Arri	Off	●	○	○	COM 1	38400	8	N	1	0.0.0.0	0
Debug	Off	●	○	○	COM 1	38400	8	N	1	0.0.0.0	0
Data Output 2	Off	●	○	○	COM 1	38400	8	N	1	0.0.0.0	0
Data Output 3	Off	●	○	○	COM 1	38400	8	N	1	0.0.0.0	0
Hyperdeck	Off	○	○	●	COM 1	38400	8	N	1	0.0.0.0	0
Root DMX	On	○	●	○	COM 1	38400	8	N	1	192.168.1.31	6454

Set the IP address according to the Ethernet to DMX interface’s IP and the Port number to **6454.**

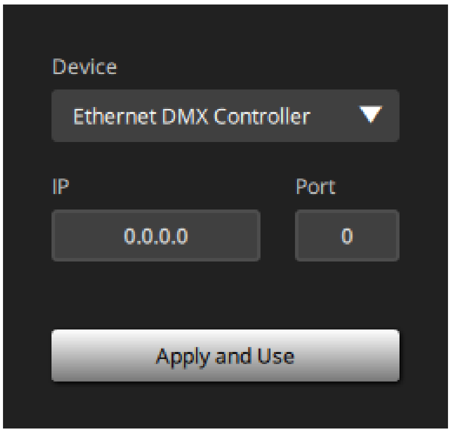
For setting up from DMX setup, go to **Setups** → **DMX**.



DMX Setup

Clicking on DMX button will open a pop up. Selecting the Ethernet DMX Controller from dropdown will show you the text fields for IP address and port.

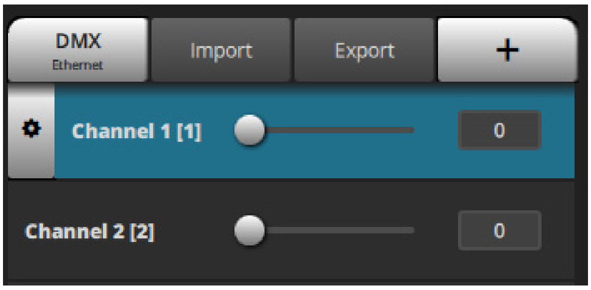
Set the IP address according to the Ethernet to DMX interface's IP and the Port number to **6454** and click **Apply and Use**.



Device: Ethernet DMX Controller

IP: 0.0.0.0

Port: 0



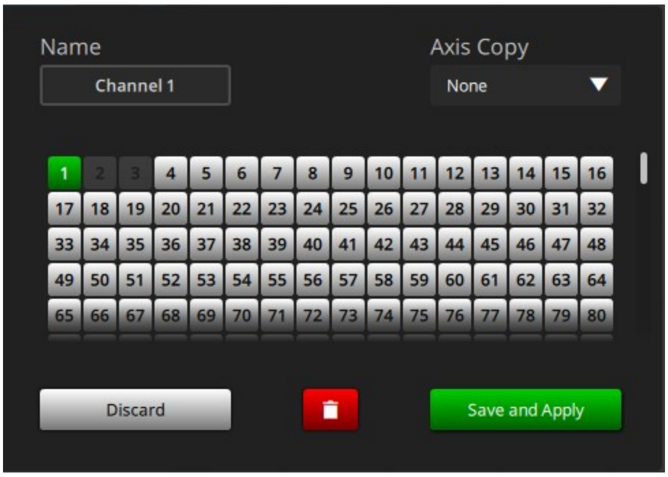
Channel 1 [1] 0

Channel 2 [2] 0

Pressing the + button, you can add channels. The DMX setup allows configuration of up to 512 DMX channels. Each DMX channel can have any number of frame positions where the value can be set.

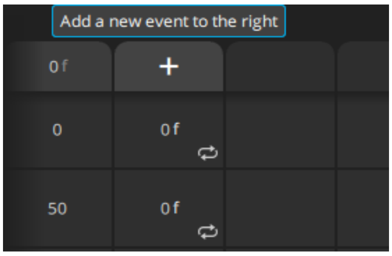
Each channel has a channel number (this is a number between 1 and 512 and is the unique address of each device in your DMX setup). In the image, "Channel 1" is the channel name and "[1]" shows the channel number. Hovering over the channel, will show the settings button on the left for the regarding channel.

In Channel settings, you can adjust the name of the channel an channel ID, delete the channel and copy to an axis positional value.



Name: Channel 1

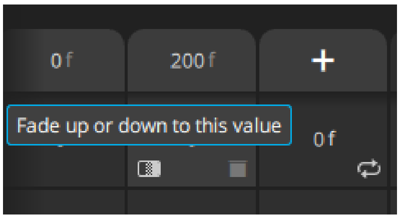
Axis Copy: None



In the right part of the setup window, a + button can be seen. Using that you can add different frame numbers along with your move to add channel value which will adjust the DMX channel.

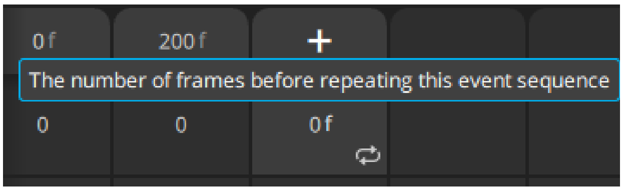
When a channel is added, a hovering over the row will appear + in the view. Clicking on that under added frame number will let you enter the value for that channel through the move. The value should be in between 0 and 255. Clicking 'Apply' will allow you to try your settings but will not save them to disk. Clicking 'Save' will save the settings to disk into the current job file.

By default, Flair changes the channel value right on the frame you determine. Flair also offers fade up and down option which creates a linear fading effect from the previous value to the next value. To enable that, hover over the left-bottom of the channel value box. By clicking on that, you can toggle on/off that option.



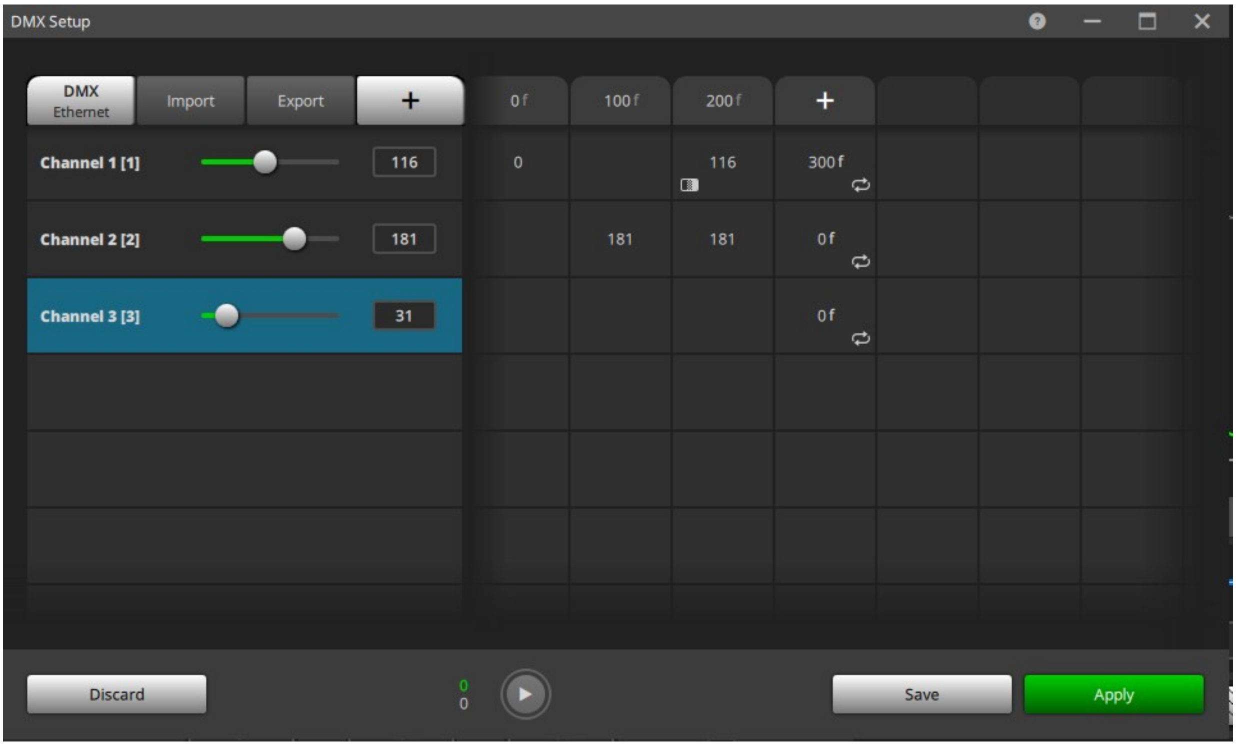
Fade up or down to this value

The sequence of value changes can be repeated by entering the number of frames between each repeat into the 'Repeat' field. This can be found at the end of the row of each channel with a little loop icon at the bottom right corner.



The number of frames before repeating this event sequence

If a channel is added when there is no underlying move in Flair, a message “Move has zero frames” will be displayed. DMX control is dependent on the timing of the underlying Flair move.



Chapter 17 **Data Import and Export**

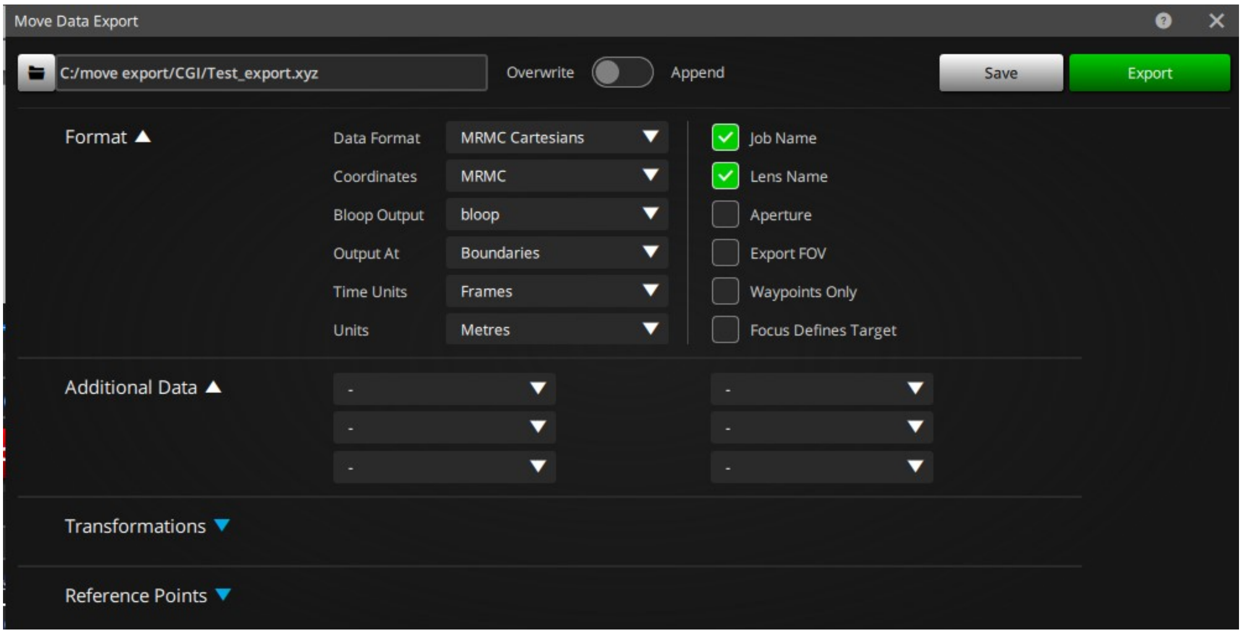
17.1. Introduction

CGI Data is imported and exported as ASCII information with lines of data defining at very least the XYZ coordinates of the Camera and the XYZ coordinates of the Target, or the orientation of the camera. Each line can optionally also contain time information, camera roll, and well as positional information on up to 8 other axes.

During both CGI Import and CGI Export, an “Action in Progress” box will appear to inform you of the progress of the import or export and allow you to interrupt it if so desired.

17.2. Move Data Export

Data can be exported from the Flair program in a variety of ways and in a variety of formats. The Move Data Export dialog box can be used to specify the data file that you want to export to, the format you want to use and several other options making the data export flexible and informative.



Move Data Export

C:/move export/CGI/Test_export.xyz

Overwrite ○ Append

Save

Export

Format ▲

Data Format: MRMC Cartesians

Coordinates: MRMC

Bloop Output: bloop

Output At: Boundaries

Time Units: Frames

Units: Metres

☑ Job Name

☑ Lens Name

□ Aperture

□ Export FOV

□ Waypoints Only

□ Focus Defines Target

Additional Data ▲

- -
- -
- -
- -
- -
- -

Transformations ▼

Reference Points ▼

17.2.1. Format

Export: This button causes the data to be exported in the selected mode to the selected destination.

Browse: The export file can either be typed straight in or you can use a standard file selection dialog box to select the file you wish to export to

File Data Format: The available formats are:

- **MRMC Cartesians:** The data will come out as an XYZ coordinate of the camera, an XYZ coordinate of the target and the roll angle, once per frame.
- **MRMC 3Node**: As with the Carts Raw data, but instead of a roll axis, the Up Point is exported. This mode is ONLY valid when in Roll Up.
- **MRMC Eulers:** Euler (pronounced “oiler”) angles are the 3 angles that define the direction in which something is looking. Pan, Tilt and Roll are a

good enough model of these angles for our purposes. In this format, the Location of the camera is output in MRMC Co-ordinates, and the Pan, Tilt and Roll angles are presented with Pan positive being to the left and Tilt positive being up and the Roll is as above.

- **MRMC Axes**: Here just the raw axis positional data is exported, for Track, Lift, Rotate etc. This has been provided for those who would model the rig themselves, or for interfacing to cranes and dollies with encoders.
- **Kuper format**: Exports data in the same form as Kuper motion control software.
- **Lightwave format**: Exports data in a Lightwave® motion file (Old version).
- **MRMC Carts Curve:** This output format has been developed in order to improve interfacing with certain CGI packages. The data is exported as curves instead of discrete (per frame) information and this format should only be used with special software that can interpret the data.

  Note that in the Kuper format, none of the later options have any effect and in the Lightwave format, only the units makes a difference.
- **FilmBox:** If you have purchased the extra licensing required to use the "FBX" format, then you will be presented with three additional export options. "FBX 1 Node", "FBX 2 Node" and "FBX 3 Node". When you use these formats, all the other format options in this display are ignored, and an FBX file is generated for the motion of the camera. The 1, 2 and 3 node allows you to create a simple camera, a camera with a target and a camera with a target and an Up point, much like the way in which Flair works in "Pan Tilt Tracking", "Target Tracking" and "Roll Up".

Coordinate System

- **MRMC:** The standard Mark Roberts coordinate system which has X forwards along the track, Y to the left and Z is up; Pan is positive to the left, Tilt is positive up and Roll is positive when the image rotates clockwise.
- **Softimage:** Softimage coordinates are different and are common to many 3D packages. In their system, Y is vertical with up being positive, X is to the side with positive to the right and Z is positive backwards along the track.
- **Kuper:** Kuper's coordinate system is similar to Softimage's but there are subtle differences.
- **Maya Y Up:** This is the default Maya coordinate system, and the MRMC coordinates are manipulated to match.
- **Maya Z Up:** This is not the default coordinate system and is not recommended though it does share Z up with the MRMC system. However, a tilt of 0.0 will point a "Maya Z up" camera at the ground.
- **Lightwave:** Lightwave® has yet another way of representing the camera and target and this mode is provided here.
- **3DMax:** Likewise for 3DMax®.
- **O/R or Append:** Allows you to select if you are going to overwrite any existing data with this output or append it to the bottom of an existing file. If "Append" is selected and the file already exists, it will be opened and the data will be appended to the end of the file, otherwise a new file will be created.

- **Job Name:** Includes the name of the job into the file as a comment if “Yes” is selected.
- **Lens Name:** Includes the name of the lens into the file as a comment if “Yes” is selected.
- **Current Aperture:** The currently set aperture is displayed here so you can include it in the file if so desired. The camera aperture is not controlled or defined by Flair, but you can select it in the Rig Model display and this data is useful for transferring data to and from CGI.
- **Aperture:** Select “Yes” to include the aperture in the exported file.

Bloop Output: It is very useful to have the data as to which frame the Bloop fired on together with the CGI Data. This toggle allows you to state which output was used for a Bloop, and then the first 5 firings of that trigger will be listed in the CGI output file as a comment.

Output At: “Boundaries” or “Frames”. This could also be described as “Centre Shutter Closed” or “Centre Shutter Open” In a standard move, the position at frame 0 and frame 1 etc. are the shutter closed positions, the data for export to CGI usually requires the shutter open position. Select “Frames” for the shutter open position and “Boundaries” for the shutter closed position. If you output a 100 frame move on boundaries, you will get 101 lines of data, in frames mode you will get 100.

Time Units: “Frames”, “Seconds” or “None”. “Frames” is a more universal format as the “Seconds” format is camera speed dependant.

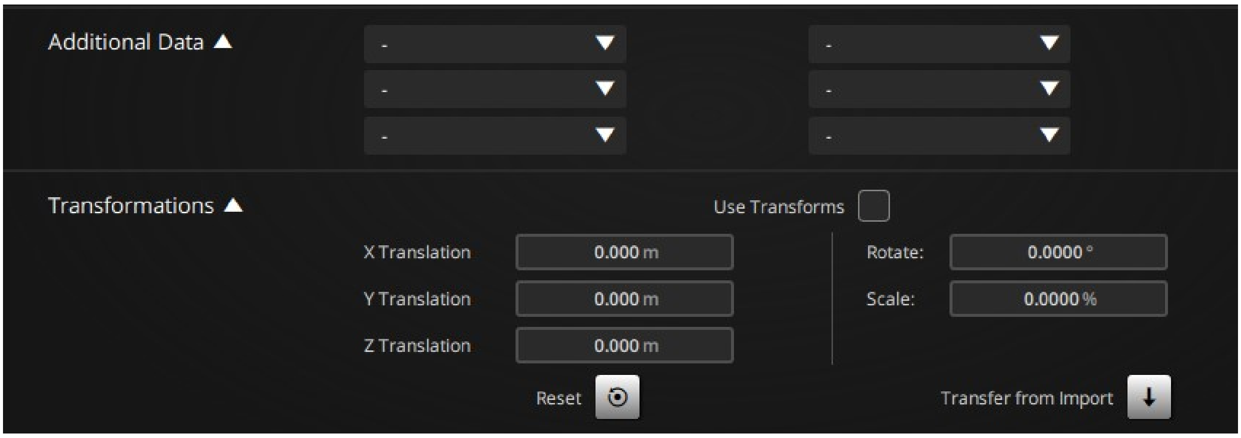
Units: The Flair system works in metres internally and is the default output mode, however inches, centimetres and feet are also supported.

Export FOV: The export of the Field of View as calculated by the system is optional. The accuracy of this data really depends on how well you have set up your lens and since the field of view is really dependant on the format and many other considerations, it should be considered as a guide rather than an absolute. Feedback and suggestions from customers on this topic would be welcome.

Waypoints Only?: If selected, then only the waypoints of the move are exported which can create a smaller file for testing rather than having to export the entire move.

17.2.2. Additional data 1-6

Sometimes the movement of other axes can be important for CGI work. In order to provide a means to put all this data into one file, the user has the option of outputting positional data of up to 6 other axes with the standard CGI data file. This could be from model movers or any other axis. A blank label indicates that the option is turned off, otherwise simply select the axis whose data you want to output and click on the “Export” button.



17.2.3. Export Transformations

When data is imported from CG very often they want to have the data back again as it is usually modified or changed slightly to make it more workable. You can use the import transformation used when data was brought in to convert the export data back into the same space as it originated.

If “Use Transforms” is checked, then the listed transformations are applied so as to undo the transforms used on import.

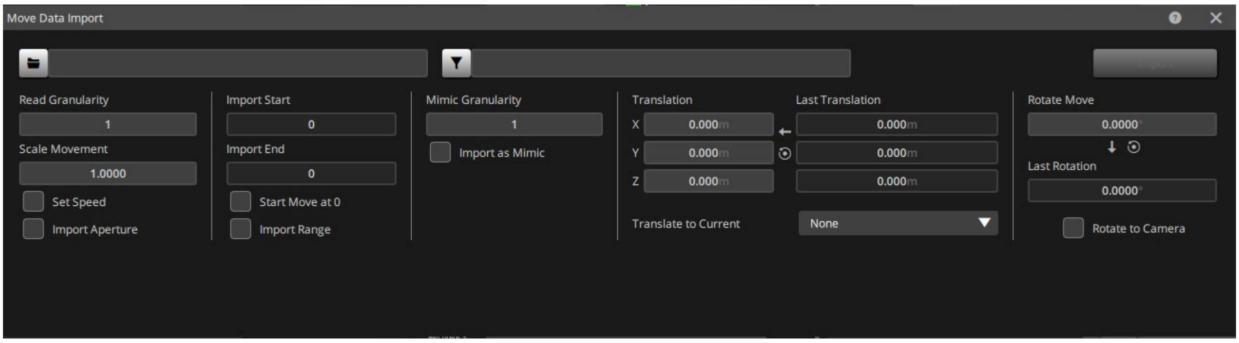
“Zero” zeros outs the values in the transforms and “Xfer” copies the transforms used on import into this display. You may also edit the transforms, but only very advanced users would do this under normal circumstances.

17.3. Move Data Import

Move Data Import is a procedure that requires some knowledge of the data you are trying to import as well as a good understanding of the limitations of the rig you are using.

File Data Formats: The file data format is determined from the header at the top of the import file. See the section above on data export for more details on data formats. More exact format data can be found in the Technical Section at the end of this Manual. The file formats that are supported are: Any Flair Export file, any Kuper export file and a Lightwave™ motion file. (.mot) (This format is now outdated)

FBX: As with CG Export, if you have purchased the licensing required to support Filmbox (.FBX) files, then these can be imported. If you try to read in a valid FBX file and you get a message about the header not being supported, then you should check the licensing. Otherwise you will be asked to select the camera and take number from the file to import. Other import functions ARE valid when importing an FBX file.



Move Data Import

Read Granularity: 1

Scale Movement: 1.0000

□ Set Speed

□ Import Aperture

Import Start: 0

Import End: 0

□ Start Move at 0

□ Import Range

Mimic Granularity: 1

□ Import as Mimic

Translation: X 0.000m, Y 0.000m, Z 0.000m

Last Translation: 0.000m, 0.000m, 0.000m

Translate to Current: None

Rotate Move: 0.0000°

Last Rotation: 0.0000°

□ Rotate to Camera

Import: Causes the data to be imported with the currently set configuration.

Test: Checks if the file is in a correct format and has a recognised header.

Browse: The import file can either be typed straight in or you can use a standard file selection dialog box to select the file you wish to import from. It is the user's responsibility to make sure that the file conforms to MRMC standards of import format.

Browse Filter: Allows the selection of a simple text file that can be manually created and will cause the input frame to be filtered depending on the frame counts listed in the Browse Filter file. This is an advanced feature and should be used only in consultation with MRMC. Normally this field would be left blank.

Read Granularity: If you do not want to import every single line of 3D data, you can select to read every second or third line or more. In this way a large move can be read in without having a waypoint for each frame. Obviously if you want it to follow absolutely exactly, you would leave this at 1. In many cases when you are importing data, I would tend to import only as much data as is needed to rough the move in unless you need it followed very closely. A large number of waypoints is hard to manage and almost impossible to edit or scale in time. This figure can be set to a high number when you are working with a move to get it imported and working out the best offsets to use, and then set back to a low number when these are figured out.

Scale Movement: The imported Cartesian data can be scaled to match units between systems. A value greater than 1.0 will scale it up. The file should state in the header what the units it is in and this should be automatically compensated for by the import routine, so additional scaling can be done here. This, of course, is ignored if the import file is axis positions instead of XYZ data.

Set Speed: Sets whether the move speed should be set to the speed in the file or not.

Import Aperture: The aperture of the camera can be included in the CGI Import file. This feature is normally not used.

Import Start & Import End: If “Import Range” is set to yes then the Import Start and Import Stop will be accessible and allow one to specify from which frame to which frame the imported file should be read and imported.

Start Move at 0?: Flair moves start at 0 by default, and often imported moves will start at some other frame count relating to the scene they came from. If you check “Start Move at 0”, the imported frame counts will be offset to start at 0 and increment from there.

Import Range:

Mimic Granularity: Just like Read Granularity, a Mimic Granularity of 1 would import every single line of mimic data. You canchange it to a valuto read

Import as Mimic: The data is imported as above, but only the first line of data and the last line of data have a waypoint assigned to them, the rest of the data is imported as a Mimic move and the imported axes are set in Mimic mode when the data has been imported. This allows all the data to be imported and followed implicitly without having to have huge numbers of waypoints. This data can also then be edited, scaled and adjusted.

X, Y & Z Translation: These boxes allow for offsets to be set for adjusting the move around as it is read in. The units are in metres. This has no effect when importing axis data instead of XYZs.

Last Translation: Is a series of three boxes that show you the values used to achieve the last translation.

Translate to Current: “None”, “Camera” or “Target”. Used to facilitate matching offsets between the CGI system and the MRMC system. If you have a move to import that starts aimed at a specific known point in space, then you can point the rig at that position and set the focus to that distance. If this flag is set to “Target”, the entire move will be offset so that the start point of the imported move matches the point you are currently targeted on, or the position of the camera.

(Alternative selection which is displayed when you are using Head kinematics)

Match Motion to Head: If you are importing a move from a motion control system into just a head for shooting background plates, the relative motion of the camera nodal point is pretty much irrelevant for distant background plates and can be ignored. Since moving a large motion control system to the top of a mountain for a background plate can be a chore, this feature will be available if your system is just configured as a head. In that case you can import the standard XYZ file and the important pan, tilt and roll data will be extracted.

Rotate Move: Allows the move to be rotated about the vertical axis as it is imported. The units are in degrees. No other rotations can be performed during import. This is also ignored if the data being imported is axis data.

Last Rotation: Shows the rotation angle used when the last move was imported if “Rotate to Camera” was selected.

Reset rotation value to current: Sets the translations and rotation values to 0. This is useful if you are resetting something or importing a different move that may have other import values.

Transfer rotation value from the last used to current: Set the translations and rotations values to the last used translation and rotation values.

Rotate to Camera: Sets whether the imported data should be rotated so that the first line of data matches the current camera pan angle.

How these are used: The Flair import system relies upon using a line up frame or position. Usually we will agree that Up is Up, and often the scale of the data is known. What is not known is how to rotate and translate the move to make it work in the real world. To handle this, the easiest thing is to place the camera in the virtual world in a position that can be duplicated in the real world and use this to define these values.

If you had to shoot a move that went around a doorway, and that move had been pre-visualised in a CG package, then put the Virtual camera a known distance from the doorway with the camera level and the cross hairs directly in the middle of the door. Export that position as a “move”. Now place the real camera in exactly the same position, making it the known distance away from the door, with the camera level and the cross hairs exactly in the middle of the door. The video tape should now show the same image as the virtual camera sees (minor differences in lens distortion not withstanding).

Now the camera is in the right place to match the virtual camera. So import the camera “move” with the translations and rotations set to 0.0 and the option “Translate to Current” set to “Camera” and “Rotate to Camera” checked. If you have got this right, then the positions shown in the numerical read out should closely match the current axis positions. If you run the move, the camera should stay almost exactly where it is with maybe only a slight “Goto”.

If you have that right, then you have discovered the rotational and translational differences between the two systems, so transfer those values into the Translations and Rotations, switch the “Translate” and “Rotate” toggles off and you are ready to import the actual move. Once done for a set up, it should work with the same values from there on out as long as nothing changes in the real world or virtual world’s space. Also read the section below on “Matching Motion Control to CGI” as this will also have a major bearing on what you are doing.

17.3.1. Add Scaling Transforms:

If you use the “Transform” feature in the 3D Graphics display, then the system uses a form of scaling to translate and rotate the move to allow you to position the move better within the range of travel of the rig. If you were to re-import the move, then these transforms would be lost. Using the “Add” button allows you to add the transforms used in the scaling mode to the transforms being used when the move is imported so that the move should be transformed to the same location as the scaled move, but without having to use scaling.

MRMC Format XYZ Information

The XYZ data is printed out in the form that we understand and use for Target Tracking. It may not match the way other 3D computers think about 3D space. See the chapter on Technical Information for more data on this. Incidentally, it seems that every CGI and Motion Control system uses a slightly different coordinate system!

17.4. Matching Motion Control to CGI

The first thing to know is that this process requires some skill and understanding. One cannot expect to import or export the data and have it work perfectly first time. On the other hand with some care and a little experience it should be relatively simple to get a good match.

17.4.1. System Setup

Any XYZ data imported into a 3D-software package from the Motion Control system is only as good as the system set up. If the scalings or the zeros on the rig are not correct, then the data will be incorrect to that degree. An angular offset of as little as 0.1° can make a difference, so it is important to make sure that the rig is set up as well as possible. Even the lens used can make a difference to the exact position of the tilt and pan axis zeroes if they are not properly aligned. Experience will tell you what is important and what is not. Even the stiffest rig will bend and flex under gravity and the XYZ coordinate output cannot be expected to be perfect. From our experience, final output accuracies of 5-10 mm are good, but this really depends on what you are shooting.

17.4.2. Image transfer

Images shot on film are generally Telecined and then imported into the CGI system via tape or disk recorder or some storage medium. The film to tape ‘Telecine’ process can be expected to change the shape of the picture somewhat. What you see on the film and what actually appears on the screen when it is displayed may well be two different things. The Telecine process can crop, scale and squeeze the image and may not give you full gate with the correct aspect ratio. The video import process can also change the aspect or pixel ratios, and this can lead to the image not matching the CGI world. A handling for this is to make up a large grid of exact known size and shoot a few feet of this for each lens. Then after the Telecine and video import, compare this image to a virtual grid which you can create on the CGI system. If these two do not match then you can identify the problem and correct it. It is also important to make sure that the centre of the grid is in the centre of the cross hairs so that you can see if there is any lateral or vertical shift. With HD cameras, the actual size of the sensor varies from camera to camera, and with different formats. You have to find out the exact size of the sensor that was used on a shot in order to get the correct size of image. Some cameras even change the size and resolution they use when you change the capture speed, and this is important to matching images.

17.4.3. Lens distortions

Since lenses vary with manufacturer, model and serial number, there is no way of modelling them all accurately in CGI. The above test is also a good way to tell how much your lens is distorting. This is usually not critical if the objects of interest stay centre screen. If they move towards the edges then lens distortion can create some image mismatch. In this case the CGI image may have to be animated to provide correction. Be particularly careful of Zoom lenses. Typical aberrations include non-linear zoom tracking (banana effect) and distortion of the image while zooming (pincushion effect). In addition to this the exact position of the eye point (nodal point) moves as you zoom (up to 1.5 feet on some lenses). There is no easy way to determine the exact field of view even if you have a rough read out of the zoom axis position. Further research is ongoing at Mark Roberts Motion Control to see how far these factors can be predicted, but for the moment, be aware that these things can cause image mismatch.

Shooting a chart while running through just the programmed zoom can assist calibration. This should be done on a chart that has clear markings and recording exactly how far the chart is from the lens. This can be used to create a virtual grid and lining it up with the exposed image will give a good idea of the actual change in focal length. Note that the important distance to record is the front nodal point to the chart. Since the front nodal point moves, this causes further problems which you will have to work with or around.

It can also be a good idea to shoot a pass with no zoom motion to help to eliminate problems in the line-up. Non-standard lenses, such as Macros and Anamorphic, raise further issues.

17.4.4. Co-ordinate zeros

The typical coordinate zero for a Mark Roberts Motion Control Rig is on the floor in the middle of the rails under the centre of the turret when the rig is at zero itself. XYZs are measured relative to this point. The typical zero on a CGI system is where the main object is placed; at the origin. This means that even if the axes are matched and scaled properly (our values are in metres) then the mismatch in the zeros may cause some confusion. The correction for this is simply translations in the three axes of the move and of the interest point. This can be done once the move has been imported, or can be done by offsets, which can be set up in the import software. The amount of translation can be worked out by measurements but could also be done by eye, correcting for the line of vision at the start and at the end of the move. If you have a specific frame where the target is exactly in the centre of frame, you can use that for the line-up, but the distance from the object is harder to match as it may be a Focal Length or Field of View adjustment that is needed.

17.4.5. Matching Coordinate Axes

Of course, everybody agrees that up is up, but there is no guarantee that what the Camera thinks of as forwards is directly aligned with the CG world. A good handling for this is to select a couple of points that are common to both the real world and the CG world. In many cases, there are walls or objects that are in both or can be set up in both or possibly even the actual product if it is a commercial. Use the Flair Camera System to survey 2 or more points, measuring the distance from the camera to them with the cross hairs lined up exactly and storing this information as a move. This will then locate the image in 3D Flair Space and you can export the data to the CG package, model the same object in CG and see that the real world and CG worlds line up. By using 2 or more points you will be able to determine the rotational and translations adjustment needed to get the 2 to match (As well as any scaling factors).

This technique has also been used in reverse with good results. Generate a move in the CG world that exactly locates points that will appear in the real world, then when the data is imported, you will know how to get them to line up through rotations and translations.

17.4.6. Additional Data

When shooting a move that will subsequently be exported into the CGI world take some measurements of specific parts of the scene relative to the rig. Also shoot a pass with an object of exact known size and position in the centre of the screen. A one-meter cube is good for most live action situations. Smaller and larger cubes should be used as per the scale of the shoot (for example, tabletop etc). The cube is lined up parallel to the rails and shot with a simple pass, preferably with no zoom. This can then be used to correct the translations to achieve the best match. Often in the heat of the moment on a shoot these ‘extra’ actions can get squeezed out, but they can save you hours or even days in post production if they are done properly.

17.5. Flair CGI Import how-to

This is a quick guide on how to import CGI data into Flair covering some tips and tricks. You do have to read the whole chapter and really understand what you are doing to make this work, there is no substitute for that or you will continue to be completely confused by what is happening. As a general rule Flair DOES import the data correctly and tries to do what is asked, it is a matter of understanding the process thoroughly. It does really help if the person generating the data has linked it to a rig model that matches yours and tested that the move is in fact the right size for your rig, but this is NOT vital.

The data needs to be either a Flair format data file in ASCII or an FBX file if you have the FBX feature licensed. Check under Help->License to see if it has been licensed.

Have some idea of the move and what you EXPECT it to do. Is it a push in, a wrap around etc. Also you need to know if the rig is going to be in the right place to do the move or if it is just a general background plate. Sometimes you are just putting the move in to see if the rig can do it and then you can work backwards from that to know WHERE to place the track to get the shot.

Using the hand held box, move the rig to roughly where you think the axes should be for the start of the move. It only needs to be somewhat in the ballpark. Move the master axes if any to a sensible value. Generally speaking if the move is large, then the Extend should be pushed out to almost maximum to give you as much reach as possible. If you have an angle (outer arm) axis, then you should have some idea of where that is going to need to be for the move.

Go into CGI Import and set the Granularity to 12 or 24. Some value so that you AREN’T reading in hundreds of points. I usually do NOT import as Mimic if I need to until the last minute.

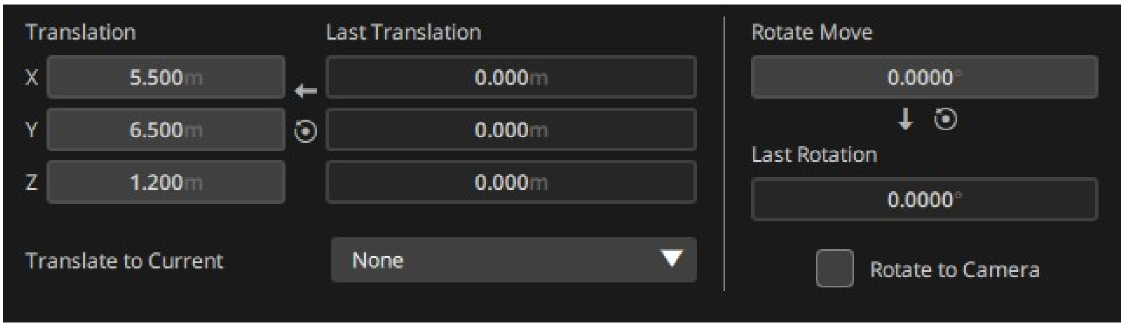
Import the file making sure that the rotation and translations are set to 0.

If the screen fills with sensible rig axis positions, then you are off to a good start, more likely you will get 90 on the rotate and possibly blank values in other axes.

Go into the Target display and look for the camera path. Ignore the target and up paths if present for now. Start with the plan (top) view.

Move the mouse along the path, or zoom out using alt and right button with drag so you get some idea of where the move is. If you place the cursor on the path, it will read out the numerical value of the camera position. If you zoom out enough you should see the graphic showing the approximate range of motion of the rig. Often the move will be very far out of the range of the rig, so you want to import with a translation to bring it closer to where the rig is. It is not certain that the CG move is set with the origin even close to where the Flair origin in. (Flair’s 0,0,0 is under the track when the base it at 0.)

You may have to translate the move in more than one direction to get it to fit. If the move is running from 5m to 8m in Y, then try entering a -6.5m translation in Y. Generally the CG height 0 (Flair Z translation, CGI Y translation) is on the floor, but sometime the rig needs to be raised on a platform and the CG height zero will still be on the floor, but Flair’s will be on the platform, so you might need to add a Z translation to account for this. Looking at the graph you can also see that it might need an X translation to bring it to where the rig is.



Translation

X 5.500m

Y 6.500m

Z 1.200m

Last Translation

0.000m

0.000m

0.000m

Translate to Current: None

Rotate Move: 0.0000°

Last Rotation: 0.0000°

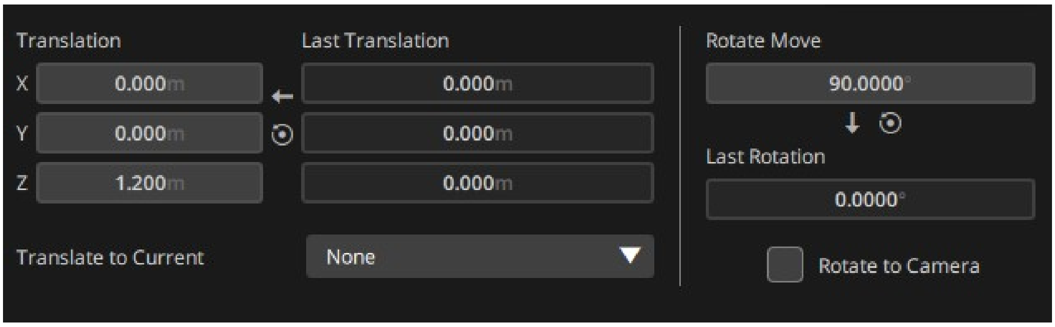
□ Rotate to Camera

You can alternatively use the “Translate to Current: Camera” to automatically translate the move so that the start of the move is where the camera is currently. This may not be exactly right, but it is a start. Sometimes I use that, then copy the values into the import translations so set the X and Y, then I manually set the Z as this is USUALLY a known about. Either 0 if the floor is correct or the height of a platform or such.

Either way should bring the move close to where the rig is. Import again and look at the graphs again. In the plan view, if the move going in the right direction. You usually have to rotate a CG move to get it in the right direction. Often just 90 or 180, sometimes other angles. It depends on the move, but look at the Plan View in the Targets Graph and see if it needs to be rotated. Import again with a guess at the rotation angle and see if it is better or worse. You DO have to look at the target to see if you are looking in the right direction. Remember that a positive rotation in Flair is Counter Clockwise.

TIP

Once you have figured this out, you CAN do it all graphically, but it is important to understand what you are doing. In the Plan View Targets display select "Transform" Then click on the graph, a small Target should appear roughly in the centre of the camera and target motion.



Translation
X 0.000m
Y 0.000m
Z 1.200m

Last Translation
0.000m
0.000m
0.000m

Translate to Current: None

Rotate Move: 90.0000°

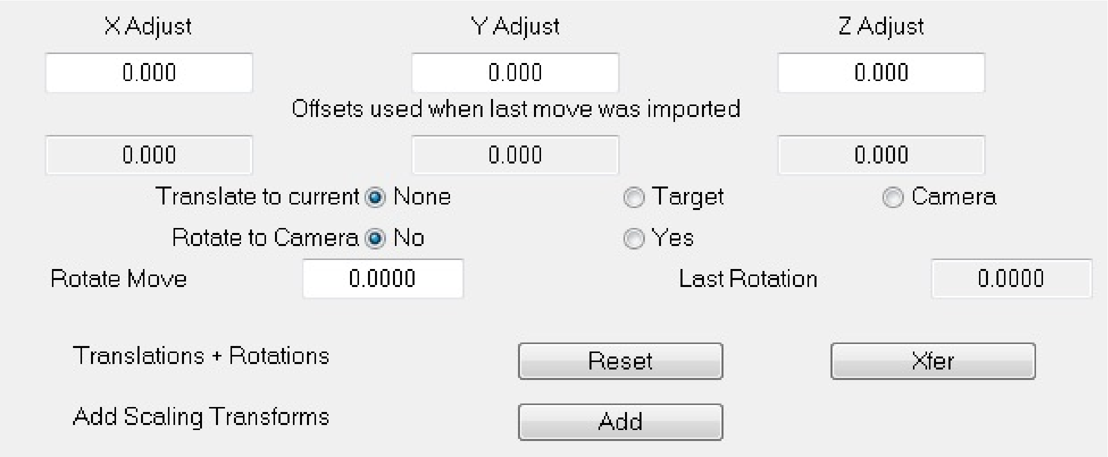
Last Rotation: 0.0000°

□ Rotate to Camera

If you click and drag this circle, you can move the entire move around in the shown axes directions, in the case of the Plan view, this would be X and Y. First zoom out so you can see where the rig's range of motion is. Now click and drag until the CAMERA's path is more or less over the track. Now click just outside of the circle and drag the mouse to rotate the move until the camera is pointing in the right direction. If this moves the path, you may have to translate it back so it is over the rig's range or operation.

So what did this ACTUALLY do in Flair terms? You imported the move and then when you did this transform, it assumed you wanted to scale the move 1 to 1, but use the translation and rotation capabilities of Flair to move the move around in space. So when you dragged it, it changed the XYZ offsets, and when you rotated it, it changed the Rotation (azimuth) value. Go into the **Effects→Scaling** and you will see this. It has set the Scale To: to "Origin" as that is basically what you are doing. If you continue from here, but then have to import again, you will have to do the whole drag and twist again. So use the convenience function in the Move Data Import dialog box to set the import translations and rotations to the values in the Scaling display. If you import again, then the move should end up exactly where you dragged it to, but without having to be in scaling.

Go back into CGI Import and under the XYZ translation and rotation values is a "Reset" and "Transfer" and under that is an "Add" button. The Add button has a tool tip which says what it does, but it basically takes the Scaling translation and Azimuth rotation and ADDS then into the CGI Import translation and Rotation. (Rotation about the vertical axis is the ONLY rotation you should be doing unless you are a very advanced user. It is called "Azimuth" in some areas of Flair to differentiate it from "Pan" or "Rotate", but it just means the same, rotating the move in space about a vertical axis)



X Adjust 0.000 Y Adjust 0.000 Z Adjust 0.000

Offsets used when last move was imported 0.000 0.000 0.000

Translate to current ● None ○ Target ○ Camera

Rotate to Camera ● No ○ Yes

Rotate Move 0.0000 Last Rotation 0.0000

As a note, you can also use the Translate to current Camera" and "Rotate to Camera" to bring the move so that it starts where the camera is now. This will set the "Offsets used when last move was imported" value, and if this works, you can transfer these offsets into the Import offsets so that you can import again with the same values, but the camera does not have to be in the exact same place. As mentioned above, it also allows you to then type in the exact correct height adjustment.

After that, the move should be transformed so that it is within the range of motion of the rig. You should be able to see if the move is the size you are expecting. If the move is a lot bigger or smaller than you expected, there may be a scale issue which you would have to resolve and then re-set the translations.

So now, can the move be run? If you try to forward run the over and it says it can do it, then go into the Rig display and just check that the rig does what you expect, make sure it is not putting the camera through the track or anything else you don't want. If it says it CAN'T do it, then also go into the rig display and drag the slider bar and see from what the rig is doing if you can work out the problem. Maybe the camera is going over the target and you are getting roll issues since the horizon is lost, and you need to use "Roll Up" Maybe you are going through Tilt 90 and need to animate the Angle/Outer Arm to better achieve the move. Maybe the move is simply too big and perhaps it needs to be scaled down. You can also get a situation where Flair adds fairings to a CG move that is built without. In this case, you can easily set Flair's fairings to 0. This will show up with large accelerations at the very start and end of the move.

So let's say we have fixed those issues or the move runs quite happily. Now you have to line it up so that it not only runs, but runs in the right place on set. Hopefully the move runs pretty easily and there is a little room to adjust the move left right up down or even in orientation and it will still move. You now have to put the rig in the right place so that the imported move runs in the real set exactly as it does in the CG set. The only way to get this perfect is to have some reference object that exists in the CG set AND the real set that you can line up on, AND that allows you to accurately determine the direction you need to look at it. (A ball is NOT a good line up object.)

If your move is simply a background plate of some trees, then there is a LOT of leeway and you can be some inches off and it works fine. If you are doing a move around a table and that needs to line up exactly, then that's what you use for your Line up.

Get the CG team to generate a static camera position or maybe a very simple move that looks at a matching point in the CG world and the real world. Let's try an example of a move around a table. Have them generate a 2 Node camera that puts the target on the corner of the table looking EXACTLY level down one edge of the table. If it's not a 2 node camera, then they need to measure how far the Nodal point of the camera is from the table and you of course have already set up and know where the lens nodal point is. You can move your camera in the real world to the exact matching point in the CG world. You can use this reference point to work out how to import the move.

Assuming you imported the move as above by just getting the move to run on the rig, you now need to lay the track in the right place to make the move around the table work. Get the table or a stand in. Get the CG team to give you the reference move and import that into Flair using the SAME offsets as you used above when getting the move to run. Go to the start of this hold/move. Now your camera SHOULD be looking exactly down the edge of the table and an exact known distance away. Place the table in that location, maybe using cart control to boom up and tilt down to verify that the table is straight, and then go back to the hold position. Measure the distance and make sure it is correct.

That is where the table needs to be relative to the track for your imported move. If the move is tight and you don't have a lot of room to adjust it, you might have to work out if that track location is right for you or if you need to work out some other solution. If you have a lot of room to adjust the move, then you can try putting the table in a better place and then try again.

Manually move the camera back to the line up/reference position, using Cart Control to get it exactly level (Cart Tilt 0.0) and looking right down one edge, Locked View Track in and out until you re exactly the right distance. Now the Rig is in the right location to match the reference shot. Clear the Import values completely and Select **Translate to current Camera** and **Rotate To Camera**. Import the line up/reference as if it were a move. Press the **Transfer** button to copy these last used import settings. (This will clear the **Translate To** and **Rotate To** flags.)

Now you are set to convert the move from CG world space into real world space. The offsets for the reference point will be exactly the same as the ones for the move as long as they did NOT change anything in the CG scene when they created the ref camera.

So Import the move again with a lower granularity and add animation to the master axes as needed.

I have imported move data numerous times and almost every time have been assured by the CG people that I need to follow their move super accurately, and I can't under sample, or maybe just every second frame or so. So I set the granularity to 2 or 4 and import the move, get it running and then the DP and director make massive changes, which takes 3 times as long since I already have 347 waypoints to work with. Usually 12 or 24 granularity is a great place to start, sometimes less, sometimes more. It is VERY rare that you need to follow the CG data exactly, and if you do, then the DP and Director shouldn't be making changes to it… Importing complex moves exactly and making it run and then editing it to change it significantly is a complete waste of time and you need to find out rapidly what the shot is and why you are importing and IF it needs to follow exactly. Modifying an imported move is MUCH slower than programming a new one from a previz video so try not to do that.

17.5.1. Other Export Data –Print Menu

A large amount of data from the computer can be printed out in order to get and keep records of moves and Setups for later reference and use. This data export function is accessed by using the "Print" button on the File pull down which then provides you with a list of the available options. The majority of the output data is printed out in simple ASCII (American Standard Code for Information Interchange).

Waypoints

When you select to print out the waypoints, only the positions as stored in the waypoints are printed out along with the associated frame counts. A heading is printed out giving the axes names, and the column titles.

Key XYZs

This data as well as the following item is provided in MRMC coordinates. The data of the waypoints is printed out converted into 3D space. Thus the data comes out as an X, Y and Z of the Viewpoint and an X, Y and Z of the Target. There are these 2 sets of coordinates for every entered waypoint of the move. This button will only work when a valid target tracking move has been entered. If you are using Mimic on a target tracking axis, the output XYZ data is NOT accurate due to the change in position occasioned by the mimic motion.

All XYZs

When this is selected, the entire move is printed out in XYZ form, with a pair of XYZs, one for the Viewpoint and one for the Target for every frame of the move. A newline character is output after every pair, and an end of file marker is placed at the end. See note above on use of Mimic with XYZ data in target tracking.

Positions

The position of each axis for every frame is printed out along with the frame number. There is no 3D coordinate data output in this form. If you are in target tracking, then the positions resolved for each axis for each frame are printed.

Velocities

The velocity of each axis at every frame is printed out for the entire move. This feature is not supported in target tracking.

Axis Configs

The axes configuration files can be printed out to be kept as a record of the move. The full data is also stored with the job, but if you need to keep this data for any reason, then these can be printed out and referred to as needed. This data should be printed out and kept as a hard copy in case of accidental loss of data in the system.

Lens Configs

The offsets and data about every lens on the system can also be printed along with any motor data that may be needed to drive the lens correctly. This data should also be printed out and kept as a hard copy in case of accidental loss of data in the system.

Kinematics

The kinematics configuration of the rig can also be printed out and kept on file as a record. As above, this data should be printed out to safeguard against accidental loss of data.

Stopping the printing

While you are printing out data, you are prevented from doing anything else as this may change the move, and thereby invalidate the data. If you start printing some data and realise that you want to stop, simply press the mouse stop button, and the printing will stop.

Chapter 18 **Zeroing**

18.1. Introduction

The process of Homing or Zeroing , both meaning the same, is used to let the computer know exactly where each axis is. The process usually involves moving each axis to a known point, either by the user or the program, and referencing from that known point. The accuracy of this depends on how exactly that known point can be sensed by the user or computer. Once zeroed, an axis can be controlled to not hit its limits of travel, and can be used to provide accurate

positional information. When the computer powers up, it assumes that all axes are at zero and before a move is shot it is vital to Zero all the axes. If the system is properly zeroed, it is possible to shoot the same shot weeks apart and still have them match even though the rig has been turned off and on many times.

Styles and variables connected with zeroing are set up for each axis in the Axis Setup display.

18.1.1. Basic principle

When an axis is zeroed by the computer, it is moved in a certain direction set by the Zero Velocity value until it reaches a sensor, once the sensor is detected, the axis slows down and stops. If the sensor is not detected within a certain time (Zero Time), then the zeroing will stop and an error will be reported. Once the computer has stopped on a sensor, it moves the motor slowly away until the sensor can no longer be detected and this point is used as a reference point to zero the axis. The axis is then moved further away from the sensor by an amount set in the Zero Offset value to its standard Zero point and then the axis position is set to the Zero At value which is usually but not always 0.0.

18.1.2. Reference Points

There are basically three different types of reference points which the computer uses. These are the **Slipping Clutch**, **Datum Switch** and **Limit Switch** (There are two variations of each of these last two types of switch). The slipping clutch method depends on the axis being driven to one extent of its travel, and then the slipping clutch allows the motor to continue moving even though the axis has stopped moving. When this has happened, the axis is at a known position, and this is taken as the reference point. The Limit switch and Datum switch are the same type of switch, but are wired in a different way. The Limit switches are wired so that the axis automatically trips out, under all circumstances except when zeroing, when the switch opens, and the Datum switch is wired so that the computer can detect when it opens, but no other action takes place.

18.1.3. Switch Accuracy

Since a magnet or roller switch is not very accurate, when one of these types of switches is used, the **zero marker pulse** (a very accurately marked position from the encoder) of the motor is used in conjunction with the switch to make zeroing more accurate. Once the switch has stopped being detected, the computer waits for the next zero marker pulse from that axis, and uses that as a reference point. Using the zero marker pulse is more accurate if there is no possibility of the motor moving in relationship to the load. With a slipping clutch of any kind, it would be impossible to use the motor zero marker pulse as this would not be in a known relationship to the travel of the motor. If the motor has no zero marker pulse or the load moves, then you can use another type of switch which is more accurate than a magnetic or mechanical switch. This is normally an **Optical Electronic switch** (Opto for short). When one of these switches is used, it is considered accurate enough by itself, and the motor zero marker pulse is not used.

18.1.4. Where is Zero?

Unless you are working with a target tracking axis, zero is wherever you want it to be. Usually Zero is about in the middle of the travel of an axis and in its normal rest position. For an animation table, the table Zero position is usually when the camera is looking directly in the middle of the table. For a target tracking axis, the Zero position is very important and must be set correctly, particularly for rotational axes. The track position is not so important and can be chosen by the user, the rotate Zero position is with the arm pointing directly down the track, the lift Zero position is usually with the arm level (This can vary, contact MRMC if you have any questions). If there is an outer arm, its Zero position is with the pan axis absolutely vertical, the pan Zero is looking directly ahead, and the tilt axis Zero is looking absolutely level. If the rig is properly zeroed, you can run the track up and down, and the viewed point on the far wall does not move.

18.2. Different Styles of Zeroing

18.2.1. Direct

In Direct Zero, the axis is considered to be at 0 wherever it is. The motor does not move and search for a sensor. This method is usually used when there is some accurate witness mark to reference off, and the axis can be moved by the user to this point, and then Zeroed. This can also be used as a quick means of setting an axis to 0 if exact positioning is not important. Note that if you have set a Zero At value, then whenever you direct zero an axis its position will be set to the "Zero At" value.

18.2.2. Limit Zero

There are several styles of Homing which involve using sensors, and this is one of these. The limit in this case is usually a magnet switch or similar which has a dual function of preventing the axis from travelling too far and for providing the reference point. Under normal conditions if this sensor is detected, the axis will turn off. When you have selected this style of zeroing and are zeroing a specific axis, this feature is temporarily overridden. With this style of zeroing, the motor zero marker pulse is used to make the zeroing more accurate.

18.2.3. Datum Zero

Similar in operation to the Limit Zero except that the switch will not under normal operation cause the axis to be turned off, it is specifically used for the purpose of zeroing. The disadvantage with the Limit style zero is that the axis will have to travel all the way to one end of its travel in order to detect the limit, and in some cases, especially with rotate axes, this can be a long way away. With the datum switch, it simply travels until it detects the datum switch which will have been positioned away from the normal area of operation of the axis, but not too far away. (If you start zeroing on the wrong side of this sensor, then you will go all the way to the end of travel without detecting it!). Once reached, the operation is the same as with the Limit Zero.

18.2.4. Limited Rotate

A rotational axis can be limited to less than a turn, can turn more than one turn but not indefinitely, or it can go round indefinitely. When the axis can go round indefinitely, then you usually use a Datum switch to zero it, and it doesn't really matter which turn it is on, as there is no difference from one to another (unless the motor to output gearing is not an even number). If the axis can turn less than one revolution, you can use limit Zero or datum Zero, whichever you like. If the axis can turn more than one turn, but not indefinitely, it is termed a “Limited Rotate” axis and it introduces its own special set of problems. Since a detection of a normal datum switch does not inform the axis in which rotation it is, and since going too far can cause the wiring to be torn out, there is often some special gearing done and a special zeroing system made. If this is the case, then it will have to be set up by Mark Roberts Motion Control to avoid undue problems, but the system is explained here if you need the data.

A special disc is set up that represents the travel of the axis, and this disk turns less than 1 turn for the entire travel of the axis. There is a vane on half the disk, and a sensor to detect the vane. From this sensor, the computer can detect if the axis is in one or other half of its travel. Sometimes there is also a datum limit on the axis which operates just as this sensor is changing state from one half of the travel to the other (The vane by itself it not accurate enough). Often this disk can also have limit switches on it to prevent the axis pulling its own wiring out.

18.2.5. Slip Zero

A slipping clutch zero is just another way of referencing an axis. It is most commonly used for applications like driving lenses as these can be hard to fit limit switches to, especially if they are hired in just for one shoot. The axis as normal will move in one direction at a speed set up in the Zero Velocity box for a specific time; when the Zero Time expires the computer will consider that the axis has travelled all the way to one end of its travel, and will use that as a reference. The axis will then move away by the Zero Offset. Once the axis has moved this amount, the position will be set to the Zero At value which is normally 0, but can be changed if convenient. In order for this system to work, the axis must have a joint that can slip somewhere in the drive mechanism, or it would simply trip out when it got to one end. The tightness of this slipping clutch is critical both to repeatable operation and to zeroing. Consult MRMC if you have any questions regarding this.

18.2.6. Opto Datum & Opto Limit

These are the same as the Datum Zero and Limit Zero respectively but these styles use a more accurate switch and do not depend on the zero marker pulse of the motor. Also can be used where the travel of the motor is very limited.

18.2.7. Opto Limited Rotate

This system is for a limited rotate axis as above where the motion is more than 360, but not unlimited.In this case the system has to know where it is and also what seg

ment of the range of motion it is in. For this, a geared down vane is used that is open for travel in one direction and closed for travel in the other direction.

This vane is used to determine the initial direction of travel, and a switch is used for the precise zeroing.

An example of this would be a Milo that does not have rotate slip rings, but can turn 270 degrees in each direction using a cable reel.

This mode does not use the zero marker pulse of the motor.

Other methods of zeroing may be listed in your system, and these can include Absolute Encoders and also axes that have no zeroing method (such as unsupported EtherCAT axes).

Contact MRMC for specific information relating to your system.

18.3. Errors

18.3.1. Time Out

If the sensor is not found within the Zero Time, the zeroing will "Time Out" and a message will appear. The handling for this is to increase the time allowed, start the axis closer to the switch, or find out why it is not detecting the switch, depending on the situation.

18.3.2. Axis Disengaged

This means that you have tried to zero a disengaged axis. Engage the axis and try again.

18.3.3. Axis Inactive

This means that you have tried to zero an inactive axis. Reactivate the axis and try again.

18.3.4. Axis Tripped

The axis was engaged, but tripped out. The most common cause for this is that the emergency stop button is pressed in. Other causes are some motor or amplifier fault preventing that axis from moving, or the axis is already on a limit (could also be caused by a wiring fault). The handling is to get the axis to power up and move normally on the Hand Held Box and try again.

18.3.5. Zero Marker Pulse too close to Limit Switch

This is an error specific to the Limit Zero and Datum Zero styles of zeroing. Since they reference off the zero marker pulse, if it occurs too close to the sensor edge, then this can cause errors. In this case, move the motor or encoder relative to the load by 1/2 turn and try again. This error should only occur when you are setting up the axis. This error means that the Zeroing MIGHT be inaccurate, it does not mean that it failed, just that you could be 1 motor turn away from the intended zero.

18.3.6. Zeroing Stopped

This occurs when an axis is stopped by the user during its zeroing process.Which zeroing style to use? Setting up the zeroing style depends on many factors, and a good understanding of the whole principle is required to make a correct decision in every case. Here are some guidelines that will help you sort out the best solution for most cases. If you have a particular application which you are having trouble with, call Mark Roberts Motion Control for more assistance.

For most axes that are not lenses, the best style to use is Datum Zero. This means that you have the accuracy of using the zero marker pulse of the motor combined with the ease of use of a datum switch which means the axis does not have to travel all the way to the end of its travel to find the limit. The datum switch has to be positioned carefully, it must not be too far away from the normal operating position of the axis, and it must be in a position so that you are VERY unlikely to try and zero the axis when it is the wrong side of the switch.

For lenses or for axes where you cannot easily fit a limit switch, you can use direct zero or slipping clutch zero. Direct zero is the easier to set up, but it requires that you set the axis to its correct position by hand. If your lens or axis is very delicate, you may not want to use slipping clutch, and therefore the best option is using a direct zero.

18.4. Setting up an axis to Zero

Getting the zeroing correctly set up is vital for easy operation of a rig and for target tracking. The accuracy of target tracking is often totally dependent on the axis being correctly scaled and having the correct Zero point.

1. Ensure the axis is correctly scaled.
2. Choose the Zeroing Style using the above information.
3. Decide which end of travel or which switch you are going to reference off.
4. Set the Zeroing Velocity to the correct direction (sign) and set its value to about 1/20 of what you expect the axis maximum speed to be. Somewhere between 2.0 and 5.0 is usually a good point to start.
5. Set the Zero At values to 0, and the Zero Offset values to 0. Set the Zero Time value to roughly how long you think it will take to reach its reference point in seconds.
6. Position the axis approximately at its zero position.
7. Zero the axis.

   Conditional: If the axis goes too fast, too slow or the wrong way, fix the velocity and repeat.

   Conditional: If the axis times out before reaching the end point, increase the Zero Time.
8. Allow the axis to reach the limit or datum switch, and observe that it slows down, reverses direction and stops. (If it is slipping clutch Zero, it will hit the end of travel, the motor will keep turning for a while and then stop).

   Conditional: If you get an error stating that the zero marker pulse is too close to the limit switch, move the limit switch a little, or turn the motor relative to the load by 1/2 turn and repeat from step 6.
9. Verify that the axis position has been set to 0. Move the axis on the hand held box to its Zero position EXACTLY.
10. Go into the Axis Setup display for that axis, and using the mouse store button, store the current position into the Zero Offset box. The axis current position as read out on the top of the screen should now be shown in this box. Apply this value.
11. Zero the axis again. If the reference point is a switch, it will move slowly off the switch and stop again, then it will move by the amount specified in the Zero Offset and stop. It should now be at the correct Zero Position. If it is not, repeat from step 5. It should move to the reference point and stop.

12. If the Zero position is correctly set up, but you want it to Zero At a different non-zero position, then adjust the Zero At value. One possible use is to Zero the Track to the middle of its travel, measure the distance from the end of travel, and put this number into the Zero At box. This would mean that the track axis position would relate to how far down the track it was. A fade shutter for example should Zero to an open position, and in that open position should read 180 degrees, this would be set up by making it Zero to the open position, and then setting the Zero At value to 180.
13. Save and Apply the changes that you have made to the Axis Setup. If you have set up a lens, check the Chapter on Lens Setups for additional information about saving this data with the relevant lens.

18.5. Re-zeroing after a System Shutdown

If the system is improperly powered down for any reason, then when the system is re-started, you are offered the option of zeroing the rig at its last known position. This can assist in bringing the axes back to their normal zero position for re-zeroing. If the lift was high in the air with the pan and roll turned through more than 180 degrees, and the power to the system was suddenly shut off, you will be faced with having to bring the arm down and move the pan and roll around to get them into a position to zero. When the system is re-powered, you should be presented with the option of setting the axes `current positions to the last known positions, and then you can do a "Move To Home" and you should be close to the normal Zero position of the rig, from which is it easy (an imperative) to re-zero.

This feature works by recording the current positions of the axes to a file every few seconds. When the program is closed down, this file is deleted. When the program is fired up and this file is found to exist, it assumes that the last time the program closed down, it was done improperly, and so offers to zero the axes at their last known positions.

> Note
>
> that this was the last saved position and can be slightly wrong if the rig was moving. Always re-zero afterwards.

Chapter 19 **Network Direct**

19.1. Introduction

Network Direct is a form of network communications to axis boards allowing for precise motion control of multiple axes.

This uses a special MRMC method of communication between Flair and the Network Boards.

An understanding of basic networking is required to setup and adjust the settings needed to connect Flair to one or more axis board or devices.

IP Address: In order to talk to the different nodes, each node is assigned a different address so that Flair can know which board it is talking to.

The addressing uses the standard IP V4 Address system with 4 numbers separated by decimal points.

For example, the first Axis board is usually at address 192.168.1.236.

The network card used for Network Direct if it is under standard windows control or under Intime control usually has a subnet mask of 255.255.255.0 which means that any board with an address 192.168.1.NNN will be able to be addressed.

When Flair knows the IP address of a Board, it can reset that board and load it with the required program and communicate to it in order to control the motors.

Each Board or Node attached MUST have a different IP address and this will usually have been set up by MRMC. The Deric (RT14) will usually have the address 192.168.1.234 and as a standard, the network board in the PC is usually assigned its own address 192.168.1.200 though this is not specifically required, it does help to keep things the same for ease of debugging.

The Network Setup display does allow you to change IP addresses of the different Nodes, but it is strongly recommended that you do NOT try doing this unless you are very sure of what you are doing as it is relatively easy to make a mistake and cause the system to not be able to find or load that board. When using Flair Classic, Flair 7 user interface has the limited capability of allowing to ping and load boards. To make any changes, in Flair Classic, select Setups → Network Setup. Without Flair Classic, Flair 7 allows the full functionality of Network Direct.

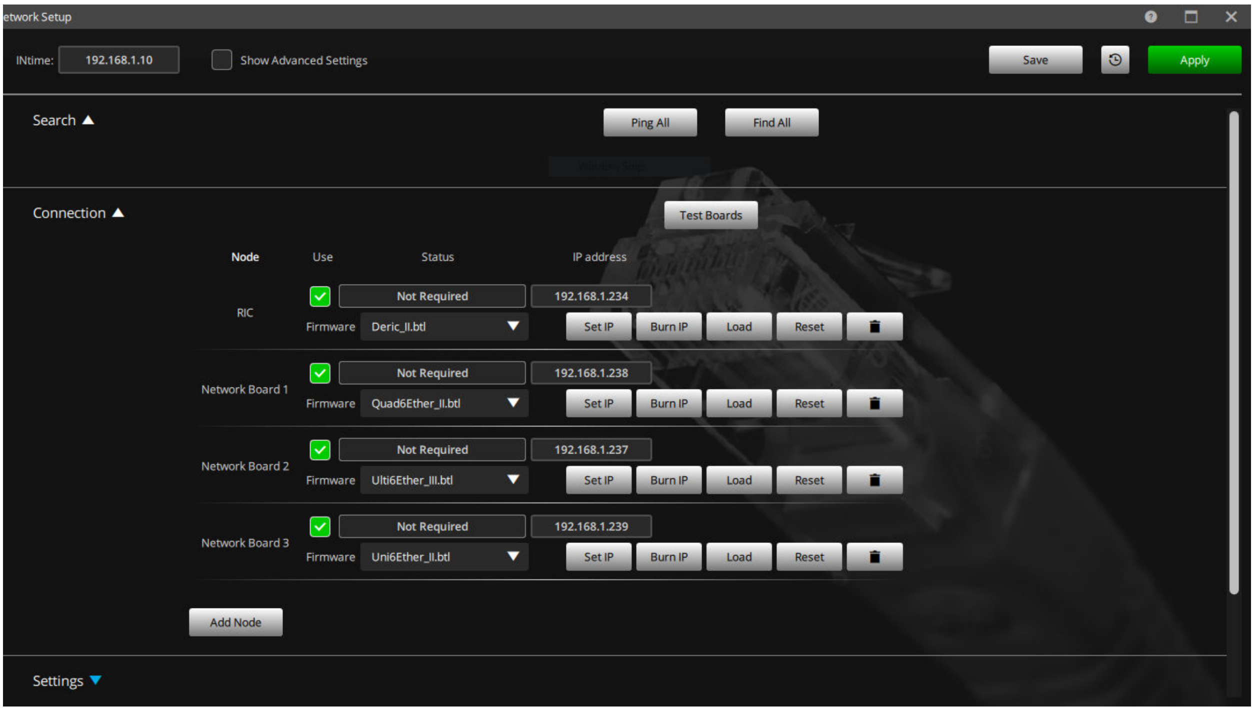
More complex installations of Network Direct can talk across local and wide areas networks and can use differing subnets, but that is outside the scope of this manual and more data on this can be obtained from Mark Roberts Motion Control.

19.2. Network Setup

Save: This saves the settings into the resource file NetworkDirect.ini and also saves IP address changes on the assigned boards. Obviously it will only save to the board if that board is connected and communicating. The NetworkDirect.ini file CAN be directly edited using a Text Editor, but this is for advanced users only.

Apply: Sets the changes made by the user in to use, changing IP Addresses etc. but NOT changing the NetworkDirect.ini file and changes made to board will be lost when the board is reset or reloaded.

Exit: Closes the dialog box.



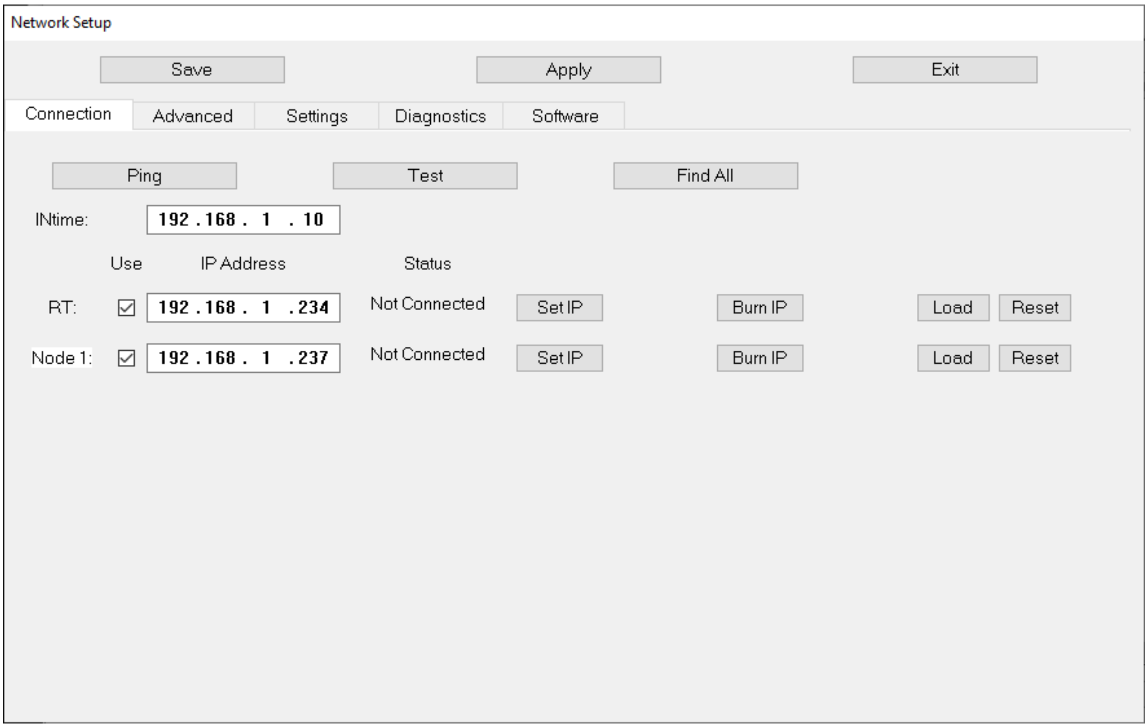
INtime: 192.168.1.10

☐ Show Advanced Settings

Node	Use	Status	IP address	Firmware
RIC	✓	Not Required	192.168.1.234	Deric_II.btl
Network Board 1	✓	Not Required	192.168.1.238	Quad6Ether_II.btl
Network Board 2	✓	Not Required	192.168.1.237	Ulti6Ether_III.btl
Network Board 3	✓	Not Required	192.168.1.239	Uni6Ether_II.btl

19.2.1. Connection Tab

The connection tab shows the IP addresses that are being used by each board and their current connection status as well as some buttons to reload the board and execute some diagnostics.

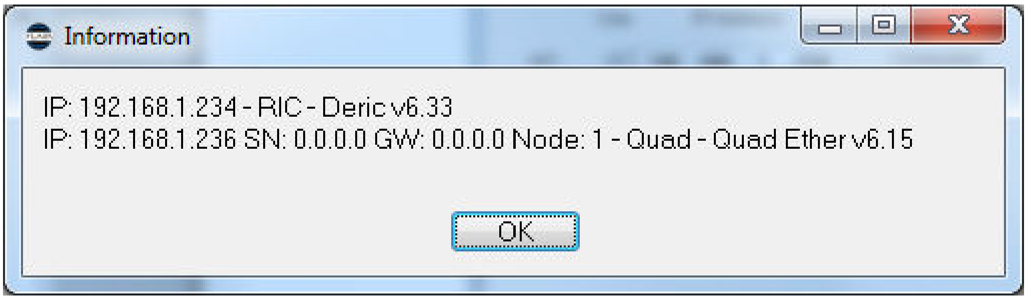


INtime: 192 . 168 . 1 . 10

	Use	IP Address	Status
RT:	☑	192 . 168 . 1 . 234	Not Connected
Node 1:	☑	192 . 168 . 1 . 237	Not Connected

Ping: This is a network standard term meaning to send a message to a network entity to see if it is alive.

In the case of Network Direct, a Ping message will be sent to each of the listed IP addresses and if a node responds, then a listing of the responses will be given usually identifying the board and or program that responded.



IP: 192.168.1.234 - RIC - Deric v6.33
IP: 192.168.1.236 SN: 0.0.0.0 GW: 0.0.0.0 Node: 1 - Quad - Quad Ether v6.15

Here we can see the addresses of the responding nodes, the first at 192.168.1.234 is the RIC board (usually referred to as an RT14) and it is running the program Deric v6.33. The attached axis node is Node 1, has the address 192.168.1.236 has no Subnet (SN) or Gateway (GW) assigned and is a Quad Board running Quat Ether v6.15. (Gateway and Subnet are not advanced configuration and are not used with standard Network Direct.)

Test: This button runs a test sending 1000 packets to each board to test the communications. Generally speaking networks tend to work 100% or fail dismally.

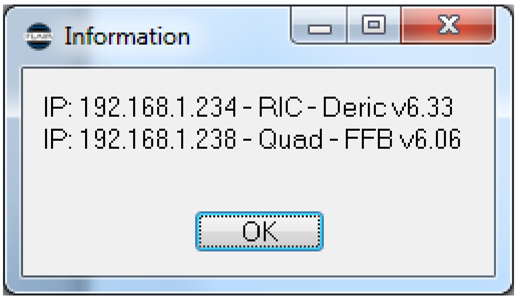
If you are connecting to the board at all it will usually be fully operational.

Of course bad cables and connections can cause errors and the three basic causes for failure should always be checked when there are issues:

- Not Plugged in
- Not Switched On
- Bad cable

Find All: Find is an advanced form of Ping which sends a broadcast ping message and should detect any board on the network if it is alive and running a Flair Program or even a Flair boot program. With Intime it will currently only detect a board with the same subnet (e.g 192.168.1.NNN) but windows should detect any board connected.

This is very useful and can detect a board that is newly plugged in if you do not know its IP address.



IP: 192.168.1.234 - RIC - Deric v6.33
IP: 192.168.1.238 - Quad - FFB v6.06

Here a board not listed has been detected with IP address 191.168.1.238 and it is a Quad Board, but it has not been loaded with a Flair Program is and running a Boot program “FFB” which stands for Flash Flair Boot, which is the program that loads from the Node’s memory when it is reset or power cycled. Seeing that the address is 192.168.1.238, you could change the address in the Network Setup, Accept, and then reload the Node with a valid program. Once you are talking to the board, you can save the IP Address to whatever value you find is useful.

Revive: This is an advanced feature and can be used to recover a board that has been corrupted and is not allowing you to connect properly.

It should only be used under guidance from MRMC and can also cause boards that are set up and work to cease working.

It will basically cause any board found to be loaded with a reset program that will set the IP address back to 192.168.1.236.

System IP Address: The IP address that the Flair computer is using is listed directly under these buttons. If you are not using INtime, it will list “Windows:” followed by the computer IP address. If the computer you are using is windows and has multiple possible network adapters, the one most closely approximating 192.168.1.236 will have been chosen.

If there are more than available network cards, a button “Change Adapter” will be on the screen and you should be able to select which adapter to use.

This is an advanced feature and it is strong recommended that you do not mix Windows Network Direct with standard Windows network traffic such as email and web traffic as it may cause timing issues.

There will also be a radio box “Static” and “Dynamic” and if you want to you can select stats and directly change the IP Address of the Flair computer.

This will cause any connected board to lose connection and they will have to be reset and reloaded.

Again this is an advanced configuration and should not need to be touched by the average user.

If you are using INtime, you will not have the “Static/Dynamic” choice and the network adapter is assigned through the INtime configuration and this should not need to be changed and should have been fully set up by MRMC on your computer.

However the address being used will be displayed in a non-editable window.

Board/Node Listing

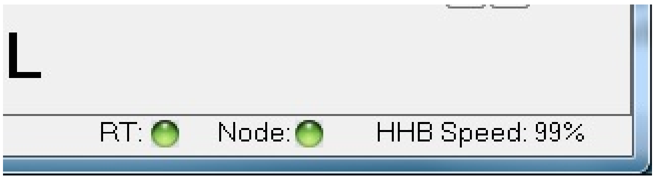
Each board or node that is supposed to be connected will be listed underneath. On the left-most will be the name of the Node, such as “RT”, “Node 1” or even “Link” when link boards are connected via the Deric. There is a checkbox called “Use” which is not currently operational and then the IP address of the Board. The Status is then listed and will be “Not Connected”, “Connected” or “Not Required”. These are mostly self-explanatory, but if the Node does not have an axis assigned to it, then it is “Not Required”.

After the IP Address is the subnet mask and Gateway which as stated above are advanced features and will mostly be blank. They may get populated if you “Ping” a node. On the right hand side are two buttons Load and Reset these are for troubleshooting and can be used to Reset a board by sending a specialised reset packet or Load the Board with the correct program.

If you “Reset” a board you can usually observe the reset light on the board flashing, and if it does not, then that is an indication that the board is not connected, has a different IP address or has crashed. Power cycling the board is another way of resetting it, and then following that with a “Find” will usually detect the address issue that is causing it to not connect.

“Load” will issue a reset and then reload with the currently assigned program for that Node. The software being loaded can be changed on the “Software” tab. If you have Link Boards in your system, these will be listed as one entry, and you can reset and reload all of them. They do not have their own IP Address but are connected to the Deric and accessed in the software via the Deric. The Deric MUST be loaded and communicating in order to reset or load the Link Axes board.

The current status of there is indicated on the Network Setup display, and also in the bottom right hand corner of the main Flair window.



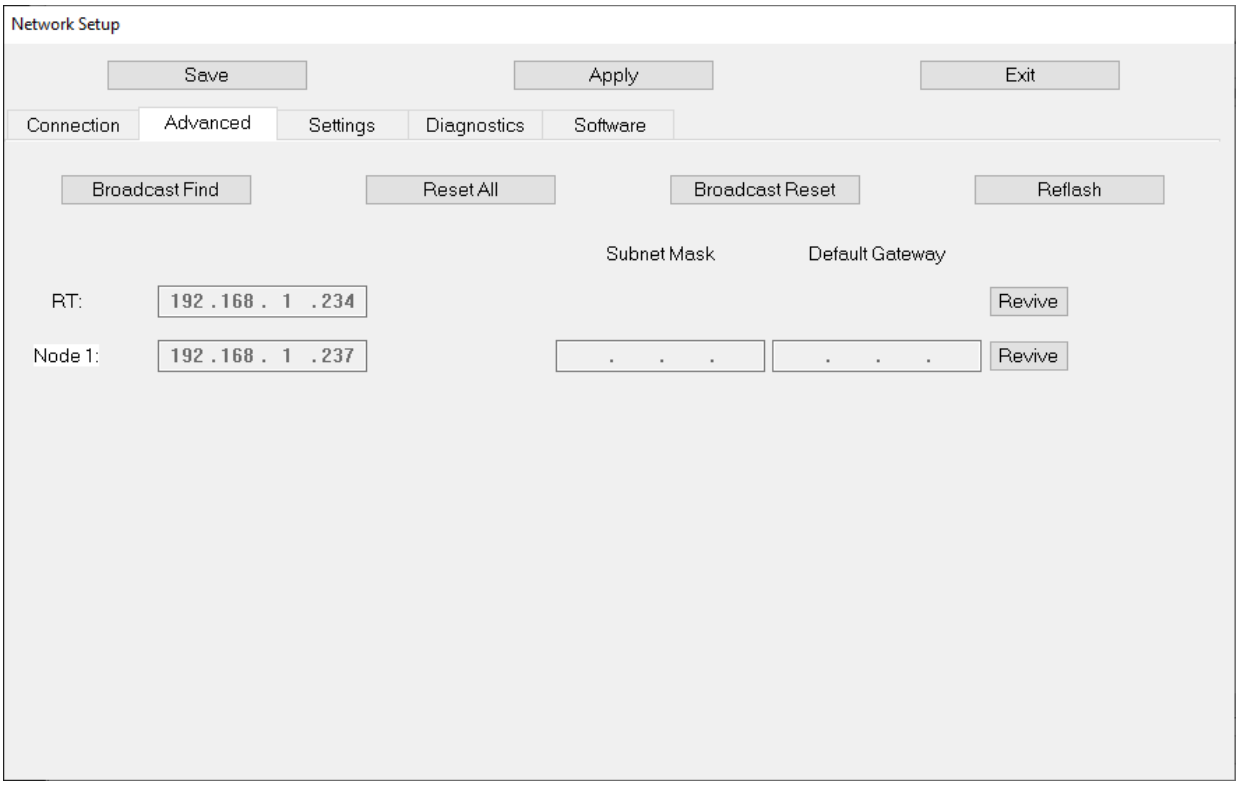
RT: ● Node: ● HHB Speed: 99%

Here you can see that the RT14 is connected and Node 1 is connected. If a node is not talking it will show as read, if it is not required, it will show as Blue.

Reloading: Any Link or Network direct node that has to be reloaded will also need to have the axes on it re-zeroed if they do not have absolute encoders.

A message will be displayed to this effect when a node is reloaded, and it is up to the operator to ensure this occurs.

19.2.2. Advanced Tab



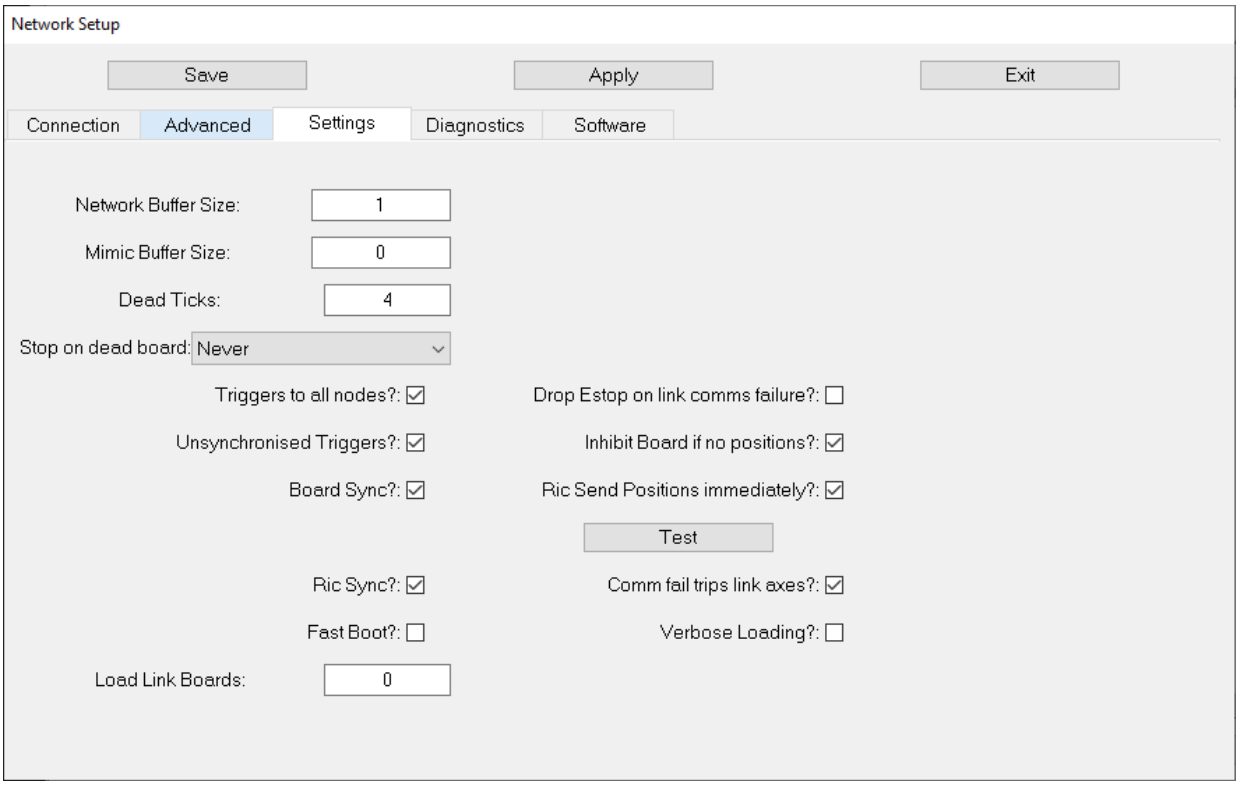
RT: 192 .168 . 1 .234

Node 1: 192 .168 . 1 .237

19.2.3. Settings Tab

There are various settings that can be adjusted by the user, and these should be well understood before making changes as they can affect the use of the system as a fully repeatable motion control system.

None of these are Saved when the “Save” button is used.



Network Buffer Size: 1

Mimic Buffer Size: 0

Dead Ticks: 4

Stop on dead board: Never

Triggers to all nodes?: ☑

Drop Estop on link comms failure?: □

Unsynchronised Triggers?: ☑

Inhibit Board if no positions?: ☑

Board Sync?: ☑

Ric Send Positions immediately?: ☑

Ric Sync?: ☑

Comm fail trips link axes?: ☑

Fast Boot?: □

Verbose Loading?: □

Load Link Boards: 0

Network Buffer Size: This is the size of the buffer on the nodes which allows for delayed packets over the network. If the buffer is 5, then the node can survive a delay of 5 Ticks before it runs out of positions, and it will continue running to that point. If it runs out of positions and requires them to maintain motion, the node will stop any motion and trip out the axes on it.

This is colloquially referred to as "Hissing". If you are having this fault with Network Direct, then you have a networking issue or your computer is not fast enough to handle the load you are using and you should get a faster computer or use INtime to guarantee that the packets arrive as expected. INtime runs with a network buffer of 1, and should never have missing packets.

Mimic Buffer Size: Mimic works on the principle of the position coming back from the axis board to control the motor. If there is an interrupt ion in the data coming BACK from the board, you can also get a jump or uneven motion in a mimicked axis, a buffer is provided in the data coming back from each node if it is used for Mimic.

2 is usually sufficient, but again for intense motion control uses, INtime is strongly recommended.

Dead Ticks: This is the number of system cycles (Ticks – usually 20 milliseconds) before a non-communicating node is considered to be dead. Once this occurs, the motors are turned off in Flair and if a move is running, it will usually stop.

Stop on Dead Board: This is a selection on what to do with a dead board if a move is running. Usually if a board that has an important axis on it stop during a move, then entire move should stop. You can select "Never" to stop, "Always" to stop or Stop ONLY if the board has an axis on it that is set in its axis set up "Stop move if axis trips" See the Axis Setup for this setting. It is normally set for any large or significant axes, something like a Focus axis, and it is optional.

Trigger to all nodes: This checkbox means that the first 4 output triggers should be forwarded to all Nodes to output on their 4 outputs. This is usually for a multiple head system and is not normally used except with that sort of a setup.

Drop Estop on link comms failure: This sets whether or not the Estop should be dropped on Link communications failure. If you do have a link communications failure, all the axes should turn off so this is optional but is an additional safety system. Only applies to systems that have Link connected axis boards.

Unsynchronized Triggers: With a large Network Buffer such as 5-10, the triggers are buffered along with the positions and this can cause unexpected delays in firing triggers such as Camera shoot outputs. If unsynchronized triggers is used, then received triggers are output immediately and not put through the same length of buffer as the incoming positions.

Inhibit Board if no positions: If a link board does not get positions, then it turns off. If a Node does not receive position from Flair, it can simply use the last received position and the motor will stay alive, or it can instantly turn the motor off.

Board Sync: Synchronization of Network Direct Boards/Nodes is a significantly complex subject. If you are NOT using a Deric, then Flair will sync itself timing wise to the first network node on the system. In this case other nodes will need to synchronize themselves to Flair and in this case Board Sync should be on. This means that all the nodes will adjust their timings very slightly to maintain a consistent timing with Flair. In the case of a Deric, then the Deric is the master timing, and Flair will adjust its timing to the Deric, ad al the Nodes should adjust to Flair. Basically if you want to be able to use a completely synced system with frame accurate repeatability and you have a Network node, this check box should be on.

Ric Send Positions Immediately: This is a historical test and should be left off. It was intended to make the system even more responsive to input controls, but was found to be unreliable.

Test: This is used to test that Board Sync is working. Pressing the test button initiates the sending of 1,000 network packets. The exact timing of these packets is evaluated and classified “Good”, “Okay” or “Bad” so that we can verify that each board is Board Syncing correctly. This analysis is sent back to Flair and reported at the end of the test. All or most of the packets should be “Good” or although a few may be “Okay”. None should be “Bad”.

Ric Sync: This is the option that we should Sync to the Ric, and in case of a Deric with Link Boards on, this should checked.

Comm fail trips link axes: If a board does not talk, the axes should tripped out.

Fast Boot: The packets being sent to the Link node via the Deric can be sent in a slow reliable fashion, or they can be sent rapidly. The Fast Boot option rarely fails, and this is recommended as it is much faster. This will allow you to select between the modes if you are having issues loading a board, particularly if it is over an extensive network system, but the default is best set in the Flair.Ini file (See Appendix on this file.)

Verbose Loading: This causes additional diagnostic information to be output on the console window or the Flair out file when loading and is useful for diagnosing loading issues.

Load Link Boards: Usually set to the value in the Ini file, this can allow you to load fewer or more Link Nodes than originally set and this is also useful for diagnosing issues with loading boards.

19.2.4. Diagnostics Tab

The Diagnostics tab shows how efficiently the network is functioning. The four counters show how many times each of the error categories have occurred.

Disconnect: This reports how many times the network has lost connection since the board was loaded.

Buffer Errors: This reports how many times the network buffer has overflowed or under-flowed since the board was loaded. Such errors mean that the network is unreliable.

Buffer -: This reports how many times the speed of Flair has been minutely slower than it should be so that Board Sync (see page 269) has needed to slow down the board by a small amount.

Buffer +: This reports how many times the speed of Flair has been minutely faster than it should be so that Board Sync (see page 269) has needed to speed up the board by a small amount.

A few Buffer+ or Buffer- errors will often occur when Flair first starts up and these are quite acceptable. An occasional occurrence of either of these errors during running is also acceptable.

Packet -: This reports how many times a position packet has failed to arrive. The board can tell this because Flair sequences every position packet.

Time: This reports the time that the board has been running in hours and minutes.

Clear: Pressing this button clears all the counters.

The box to the right which usually says **None** is for the use of MRMC staff when doing detailed diagnosis.

19.2.5. Software Tab

The Software tab lists each node (and the link nodes) in order and allows you to change the software that is loaded onto it. Generally speaking there is only one correct program for each specific node/board and this should be determined in the Flair.ini file. However when re-configuring the system you may need to try out different programs to find the right one for your node and also you can try loading older versions if you suspect there is an issue or some corruption in a later version.

“Select” brings up a file browser to locate the file and “Reload” loads the board with that file. Sufficient space is allowed for most file paths, but it is wise to keep your Flair Installs to a short path to avoid needlessly long complex names.

19.3. Network Direct Hot Boot

Hot booting is the process of restarting Flair, but not restarting a separate process or program that is still running fine. Generally speaking, this can be the Root when it is being run under INtime, or it can apply to one or more Network Direct Nodes. Of course if you hot boot a node, you can start up more quickly, and if the node has not been reloaded, generally speaking the axes on it will not have lost position and will not have to be re-zeroed. If you have multiple nodes and some do hot boot and some don't, then it is up to the user to ensure that he re-zeroes any axes that need to be re-zeroed. A very important advantage of using the Deric to run the Link axes is that Flair AND the root can be completely shut down and re-started and if the Deric Hot Boots, you will generally NOT have to re-zero any rig attached to it with Link Nodes. The option for Hot Booting should be set in the Flair.ini file and this feature will be found to be VERY useful to most operators. However it is still the operator's responsibility to ensure the rig has not lost any zeroes and is in the right position when running the move and this should be verified in critical situations such as between multiple passes.

19.4. Resolving Network Direct Issues

The most common issue with Network Direct is a board that was communicating has stopped talking and cannot be found. The primary reasons for this are listed as above, 1) Not Plugged In, 2) Not Switched On or 3) Bad cable. Check all of these before proceeding.

If this does not resolve your issue, then power cycling the Node or restarting Flair will fix most issues. If this is not solving the issue, then it is strongly recommended that you disconnect all other nodes before proceeding to ensure there is no conflict and to ensure that you do not accidentally reset a working node.

Once this is done, power cycle the Node and do a “Find”. This may take a few tries and you should wait 10-20 seconds after power cycling before doing the “Find”. If the Node does NOT show up, then it is possible that the node has been corrupted or damaged. Turn the node off again wait a minute, turn it on, give it 30 seconds and try again. If you are using INtime, it is possible that the IP address of the node has been corrupted and it is now out of reach of the network adapter. If you are an advanced user and have a Windows network adapter, you can try re-starting Flair without INtime and see if it can be located using “Find”. If you have tried the above and still not luck, then you should contact MRMC for more direct technical assistance.

If the node DOES show up on Find, then perhaps the IP address has gotten changed or you accidentally connected the wrong node. The Find message will show the IP Address of the node, and this should be the one you are trying to talk to the node with. If this is the case, then change the IP Address on the network setup and try loading. If it loads, you can THEN change the IP Address back to the correct value, save the change and reconnect the other nodes and you should be good to go.

If the Board is found, but will still not load, then maybe the reset is not working or you are trying to load the wrong file. If you press the Reset button on the “Connections” tab you should be able to see the reset light on the board flick on and off again. If this is not an Octo board (which can only be power cycled to reset) then possibly there is a more serious fault with the board and you should contact MRMC for assistance.

If the node does reset, when you reload it, the running light flickers a few times and stops, then you might be trying to load the wrong program. There are 4 types of Nodes that should respond and load from Flair:

- Ric (RT14)
- Quad (4 Axes on a card)
- Ulti (12 axes including 6 servos)
- Hex (6 Axes)
- Octo (8 Axes but same board as Quad)

Each of these boards responds with a different string when it is pinged and indicates which program should be loaded. With Network Direct Nodes, the program name is typically Board6Ether.btl. So a Quad would load with Quad6Ether.btl, and an Ulti would load with Ulti6Ether.btl. Note that there are 2 different hardware versions of the Ric, Quad and Ulti. There is an older one pre 2008 (about) and the later version which uses a better processor. If this better processor is present, the string “Delfino” should appear or the appendage _II will appear.

Either of these indicate a later processor and then a _II should be on the end of the filename before the extension such as Quad6Ether_II.btl. The Octo and Hex both only use the Delfino processor, so they would always have the _II addition to the program name. Mostly this should have been set up by MRMC and this data is provided here for the advanced user who is comfortable re-configuring his system.

Chapter 20 **Technical: Information, Assistance and Errors**

20.1. Introduction

If you have arranged telephone support for your system, then you can call Mark Roberts Motion Control during normal office hours, or leave a message on the answer machine, and we will attempt to resolve your problem as quickly as possible. If you are reporting a problem or a fault, follow the guidelines for doing so. If you simply have a technical question, check that it is not fully answered in the manual first, but if you really cannot sort it out don't hesitate to call.

20.1.1. Reporting faults or problems

If you need to get technical assistance due to a problem or fault, then it is vital that the data you report is accurate and complete. Statements like “It doesn't work” or “It went funny” are not likely to lead us to help you resolve the problem. There are many component parts to the software and the hardware and an accurate report of what happened can often help us to track down the problem quickly. The data that is needed can be summed up as

- When and how often did the fault happen?
- What part of the program were you in when you observed it?
- What sort of problem is it, a glitch or a complete crash etc?
- Description as accurately as you can give it of what happened.
- Other data that is useful is the status of the lights on the front of the root box, i.e. is the Error Light on, or are there error lights on any of the axis boards. Also as the program is running, some messages appear on at the bottom of the normal Flair Window and are printed into a file called “Flairout.txt”.
- The contents of these files can be useful in tracing down faults, and they should be copied off the system or renamed as needed as they will be overwritten when the program is next loaded.

20.2. Flair Software Wish Request and Problem Report Forms

In Flair, there is a menu which brings up forms to be used when a problem or bug is found in the software or when requesting a feature to be added to future releases. This facility is found in the Help Menu.

Remember that when reporting a problem, it is very important that we have all the information needed to repeat the problem at Mark Roberts Motion Control, or else it’s not possible to fix it. You may have to send a copy of the job you are having trouble with.

It is STRONGLY advised to always send a job, even if it is simply 2 lines of the rig position when the fault occurred as VERY often issues are specific to the set up you are currently running and the best way to ensure we can reproduce and thus fix any issue is to include a job with EVERY bug report.

20.3. Information

Every time the Flair program loads, it writes the time and date into a file called “FlairLoad.txt”. Every time it is closed down under user control it writes that time and date into the log; communication errors causing the program to shut down are likewise recorded.

If the program fails with the dreaded “Flair has encountered an error” message it will not be recorded, but you may be able to “Hot Boot”. This file is for user information about times and timing and may help to provide work hours after a shoot.

A second file “FlairOut.txt” is newly created every time the program starts which contains diagnostic information. Every time the program is started the previous file is overwritten.

“FlairData.txt” records every time a move is saved or run in camera record mode.

Job stores are saved with the appropriate filename and time stamp, move execution is also time stamped and has the move frame counts recorded as well.

20.3.1. Output & Input Data Format

Here are a few samples of output data format. The full interface is continually improved and refined and new features added, but the following samples should indicate the basic layout.

Sample outputs:

Example 1

```
# Flair Export file from Flair 3.5 Wed Apr 5 12:24:39 2000
# Job name is CARTS.JOB
DATA_TYPE CARTS_RAW MRMC_COORDS IN_METRES
POINTS 3 SPEED 25.00 TIME 0.08
Frame  XV   YV   ZV   XT   YT   ZT   Roll
 0 3.92808 0.00000 0.00039 5.73808 0.00000 0.00056 0.00000
 1 3.92808 0.00000 0.00039 5.73808 0.00000 0.00056 0.00000
 2 3.92808 0.00000 0.00039 5.73808 0.00000 0.00056 0.00000
```

Example 2

```
# Flair Export file from Flair 3.5 Wed Apr 5 12:56:41 2000
# Job name is EULERS.JOB
DATA_TYPE EULER_RAW MRMC_COORDS IN_INCHES
POINTS 3 SPEED 25.00 TIME 0.08
Frame  XV   YV   ZV   Pan   Tilt   Roll
 0 154.64876 0.00000 0.01537 0.00000 0.00542 0.00000
 1 160.71220 15.38167 14.24776 9.91823 14.06550 5.00000
 2 166.77565 30.76334 28.48014 21.13271 27.66877 10.00002
```

Specification: (Further explanations are in the Data Import/Export chapter.)Specification: (Further explanations are in the Data Import/Export chapter.)

- Any line starting with “#” is ignored and can be used as a comment. Several comment lines are generally exported with a Flair Export giving the time, date, filename of the saved file and other useful information that does not directly affect the camera path.

- Each line is terminated with a CRLF character which is 10, 13 in Dos and just 13 in other systems.
- Lines should not be more than 120 characters long and should be constrained to 1 screen width if possible. (80 Characters)
- All separate entries are white space delimited. Either Tabs or Spaces are recommended, commas also act as delimiters.
- The first valid characters in the file should be “DATA_TYPE” This is followed by a string that indicates the data designation which can be one of: “CARTS_RAW”, “CARTS_CRV”, “EULERS_RAW”, “EULERS_CRV”, “AXES_RAW”, “AXES_CRV”, “3NODE_RAW” or “3NODE_CRV”.
- The next string designates the coordinate system being used and can be one of the following: “MRMC_COORDS”, “SOFT_COORDS”, “KUPER_COORDS”, “MAYAY_COORDS” or “MAYAZ_COORDS”.
- The following string designates the units used and can be one of the following: “IN_METRES”, “IN_INCHES”, “IN_FEET”, “IN_CENTIMETRES”.
- An optional string may follow which will designate the curve type which can be one of the following. (This string is ONLY expected in the case of a curve format of data).: “SPLINES”, “BEZIERS”, “LINEARS”, “CUBICS”, “QUADRATICS”, “QUINTICS”. Most of the forms are not currently supported.
- An additional optional string may follow which will designate the form of control used on the head in the case of a Milo, Cyclops or similar rig configuration: “KEEP_HEAD_LEVEL”, “KEEP_HEAD_NINETY”, “KEEP_HEAD_CRNP”
- The second valid line contains data about the move as to number of points, camera speed and move duration at that speed. The titles are self-explanatory and the numerical entries are standard numerical entries – see sample.
- The third valid line is a header line which defines the columns. The first column must be headed either “Frames” or “Time”. This indicates is the time units are in frames or in seconds. Then follows a number of headers which must either be one of the valid axis names on the system or one of the valid Cartesian data titles which are: “XV”, “YV”, “ZV”, “XT”, “YT”, “ZT”, “PAN”, “TILT”, “ROLL”, “XH”, “YH”, “ZH”. Note that the use of the target points “XT”, “YT” etc would preclude the use of the Pan and Tilt angles. This is NOT checked for in the software. Case is ignored, but we would recommend keeping Cartesian Pan, Tilt and Roll is capitals for ease of differentiation and this may be adopted into the standard at a later date. The “H” points are the Up points for the camera to make the horizon correct and are ONLY valid with 3Node camera data.
- If you are importing Cartesian data, the strings Pan, Tilt and Roll are assumed to be Cartesian data if you are importing Eulers format data before they are interpreted as axis data.
- If you wish to leave a column entry as blank, then use a comma to delimit the data and a blank entry will be assumed. (EG “12.324 34.543 – 23.45, 12.89” will give 5 entries, the fourth of which is blank.)

20.3.2. MRMC Coordinate description

The X axis is positive forwards along the track, the Y direction is horizontal at 90 degrees to the track and is positive going to the left when viewed from behind, Z is vertical and positive is upwards. Pan is positive to the right, Tilt is positive up and a positive roll rotates the camera anti clockwise so the image turns clockwise. All 3D coordinates are relative to the camera track, and the 3 origins and orthogonal axes do not move as the rig moves around. The zero values for the coordinates vary, but in general, 0 in X and Y is in the middle of any rotate when the track is zero, and zero Z is the floor.

20.3.3. Text Config Files

You can import a text file that configures the system in various ways. The data imported in this fashion is NOT checked, so the user is responsible to make sure the data is accurate and sensible.

This feature is particularly useful for the Milo Segmented arm where the axis settings and kinematics values are slightly different and these settings can be easily changed by reading in a “ShortArm.txt” or a “LongArm.txt” etc.

This could also be used for the Reverse Guiness head when used in the Forward or Reverse mode. The Data that can be imported is categorised into Axis Data, Camera Internal Data, Camera External Data, Kinematics Data or Miscellaneous Data. Headings in the file determine what category the following data will be assigned to if it is valid. The headings are entered in the file as the Category

for example, (“Kinematics”, “Internal_Camera”, External_Camera”, “Miscellaneous”) or the Axis Name. The actual information is entered in much the same was as above, here is an example:

```
Kinematics
*Arm Length at Zero: 1.92
Rotate
*Maximum Acceleration: 30.0
*HHB Acceleration: 8.0
```

The valid titles in each category follow along with the options or a brief description. For full details refer to the relevant section in the Chapter 5 Setups of the manual.

Axes:

"Type"	**Integer or Text**	**"Independent","Algorithmic","Target Tracking", "MasterType", "FocusType","ZoomType", "IrisType", "FDXType", "Shutter", "D to A", "Min Aux"**
"Display Scale”	Float	Display Scaling Factor
“Internal Scale”	Float	Internal Scaling Factor
“Direction”	Integer or Text	“Forwards” or “Backwards”
“Brake”	Integer or Text	“No” or “Yes”
“Board”	Integer	Board to which axis is connected
“Port”	Integer	Port on board to which axis is connected
“Maximum Velocity”	Float	Maximum speed in display units
“Maximum Acceleration”	Float	Maximum acceleration in display units
“Maximum Jerk”	Float	Maximum Jerk in display units (Not Used)
“Maximum Limit”	Float	Maximum amount positive axis can travel
“Minimum Limit”	Float	Maximum amount negative axis can travel
“HHB Velocity”	Float	Top speed of motion on the Hand Held Box
“HHB Acceleration”	Float	Acceleration used with the Hand Held Box
“HHB Jerk”	Float	Jerk used on the HHB, can be 0 or about 3x the acceleration
“Mimic Controller”	Integer	Mimic Axis controller. Use Axis number.
“Mimic Scaling”	Float	Scaling from controller axis to controlled axis
“Mimic Deadband”	Float	On SpeedBoat type, range of no motion on controller in encoder slots.
“Mimic Window”	Integer	Samples or mimic data to smooth 0 – 20

“Mimic Type”	Integer or Text	"H.H. Box", "Encoder Pos", "Encoder Vel" or "Speed Boat"
“Postroll Factor”	Float	Fraction of max acc to be used on postrolls
“Preroll Factor”	Float	Fraction of max acc to be used on prerolls
“Goto Factor”	Float	Fraction of Axis Vel & Acc to be used when doing a Goto.
“Run Factor”	Float	Fraction of Axis Vel & Acc to be checked against when doing move checking.
“Zero Offset”	Float	How far to move off from the zeroing sensor to get to the correct zero position.
“Zero At”	Float	Position to be set to once zeroing is complete. Usually 0.0
“Zero Velocity”	Float	Velocity to start seeking zeroing sensor/limit
“Zero Time”	Float	Time in seconds to allow to find sensor or fail
“Zero Style”	Integer or text	"Direct", "Limit Zero", "Datum Zero", "Limited Rotate", "Slip Zero", "Opto Datum Zero", "Opto Limit Zero”
“Signal Gain”	Float	Gain to be used in motor tuning.
“Tacho Gain”	Float	Feedback dampening gain to be used.
“Integral Gain”	Float	Offset positional correction gain.
“Units”	Integer or Text	"No Units", "Metres", "Centimetres", "Millimetres","Inches", "Feet", "Degrees" or "Radians"
“Gripstick Velocity”	Float	Maximum velocity to use when under Grip Stick control
“Gripstick Acceleration”	Float	Maximum acceleration to use
“Gripstick Gain”	Float	Gain to use to amplify gripstick signal
“Gripstick Moving Friction”	Float	Friction to apply proportional to speed

“Gripstick Static Friction”	Float	Constant static friction
“Gripstick Deadband”	Float	Deadband range for gripstick
Motor Size	Integer or Text	"Normal", "Medium", "Small"

Internal_Camera

“Maximum FPS”	Float	Maximum speed of the camera
“Maximum Acceleration”	Float	Maximum acceleration of the camera, usually around 40 Frames per second per second
“Encoder Lines”	Float	Encoder lines per frame exposed
“Zero Offset”	Float	Fraction of a frame between the zero marker pulse and correct camera zero position
“Forward Offset”	Float	Fraction of a frame that the camera can turn before the shutter starts opening.
“Back Offset”	Float	Fraction of a frame that the camera can turn backwards before the shutter starts opening.
“Slate Speed”	Float	Speed at which the slate should be shot
“Slate Frames”	Integer	Number of frames to expose on each slate
“Direction”	Integer or Text	"Forwards", "Backwards"
“Cap Time”	Integer	Seconds of time required for capping shutter to close
“Signal Gain”	Float	Gain to be used in motor tuning.
“Tacho Gain”	Float	Feedback dampening gain to be used.
“Integral Gain”	Float	Offset positional correction gain.
“Shutter Angle”	Float	Shutter Angle on camera usually about 180.0
“TU On Board”	Integer	Board which controls the take ups On/Off

“TU On Port”	Integer	Port which controls the take ups On/Off
“TU Dir Board”	Integer	Board which controls the take ups Fwd/Back
“TU Dir Port”	Integer	Port which controls the take ups Fwd/Back
“Cap Board”	Integer	Board which controls the Capping Shutter On/Off
“Cap Port”	Integer	Port which controls the Capping Shutter On/Off
“Rack Board”	Integer	Board which controls the Rack View/Take
“Rack Port”	Integer	Port which controls the Rack View/Take

“Ramp Frames”	Integer	Number of frames of film required for the external camera to ramp to speed
“Sync Frames”	Integer	Number of frames at speed required for the camera to stabilize, usually only 1 needed.
“Sync Type”	Integer or text	"Off", "VTR Sync" or "Film Sync"

Miscellaneous

“HHB Factor”	Float	Percentage speed to run the Hand Held Box
“Goto factor”	Float	Fraction of the calculated speed at which to execute Gotos. Used to Slow them down!
“Transfer Speed”	Float	Speed at which time will be transferred, usually 24, 25 or 30.
“Browse Axis”	Integer	Axis number to be used for browsing the move.

Kinematics

The strings and values to be used in the kinematics section are shown on the Kinematics Setup for your system and vary from system to system.

If you are importing a Text Config file and any error occurs, the import is aborted at that time. For information of what has been imported or what the error was, look at the output file under “Help” menu “View Output File”. Errors that occur should be explained in that file, or consult MRMC.

The Rig Model displays can also save some data but as this is under development the format of these files is not yet finalized.

20.4. ERRORS

20.4.1. Error Messages

Error messages appear on the screen in a dialog box titled "Error". The likely errors that you may run into are listed here as well as any pertinent information.

No Job:

There is no job entered to be run, either there is only one waypoint, or the start and end times are the same.

No Target Distance Entered:

A Target Tracking Job has been entered, but the target distance has been omitted in one or more waypoints.

No Time for Waypoint:

The time value has been omitted from a waypoint that needed to be set.

Target Tracking Not Selected:

You have tried to do a target tracking function but this option is not checked. The most likely occurrence is when you try to look at the 3D graphics, but have forgotten to turn on Target Tracking.

Frame Count may not go negative:

Negative frame counts cannot be entered into the system, but it is possible to use one of the speed modifying functions and attempt to generate a negative frame count. This is an error and is flagged as such.

Job has a length of 0 frames:

Self-explanatory. If the move as you see it does not have 0 frames, try clicking on the numbers display under the "Displays" button on the Top Menu Bar, the screen will be re-displayed and it is possible that a change occurred but did not get reflected on the screen.

Function not yet implemented:

We at Mark Roberts are dedicated to the expansion and development of our product. Due to this, it is possible that some work may have been done on adding a feature but that feature has not yet been completed. If this happens and you try to access the feature you will get the above message. Contact Mark Roberts Motion Control for a release date for forthcoming features.

Clear Job Failed - Incorrect run state for clearing the Job:

The attempt to Clear a Job failed. Try again. The most likely cause would be that the system was in a run state of some kind such as "Ready to Goto" etc. If it was in a run state, then get it out of that state and back into the normal idle state before re-trying. (Use the mouse stop button).

No output currently open:

You have tried to output data, but have not stated where it is to go, or the attempt to direct the output to a location failed. Select the output correctly through the "Print" - "Print to ?" menu.

Incorrect run state for axis Setups:

You have tried to adjust the axis Setups when the computer is in a run state or a zeroing state. Get it out of that state or let it complete before trying to set up an axis.

Illegal Stretch Entry Range of stretch was reversed:

You have tried to stretch a section of the job but the frame counts are the wrong way round. Re-enter the correct numbers.

Invalid Time Entry:

One or more of the frame counts are invalid. The frame counts cannot go backwards, they must always increase as you go down through the job. Check these are correct and re-try.

You could also try re-displaying the screen information by selecting "Display" - "Numbers" and check to see if the data has been corrupted.

Axis cannot be reset - check limits:

The system has attempted to get an axis to engage, but the limits are tripped. The axis may be out of the allowed range of its travel, or it is possible that there is a wiring fault.

Zeroing cannot be done at runtime:

When the system is ready to run, or is running, you are not allowed to home or zero any axes. Stop the system and try again.

Axis Disabled:

This message occurs when you attempt to home or zero an axis which has been disabled. Re-enable the axis and it should work fine.

Bad Move:

There is something wrong with the job. Normally the error message would be more specific than this, is there are any problems contact MRMC for help.

Scaling to 0 frames not allowed:

If you attempt to shorten the length of a job to 0 frames, the computer will catch this as an error, possibly you did not select the correct box when you typed in the new number of frames, and the number was not registered.

20.4.2. Less common errors and system errors

Memory Allocation Error - Fatal:

There is a limit to the number of waypoints and data you can store in the system, however this number is huge and you are unlikely to run into this message if you have less than 100 waypoints! If the message does occur, contact Mark Roberts Motion Control as soon as possible and report it. Acknowledge the message and try again, if no joy you may have to shorten your job or re-boot.

File could not be opened - no apparent reason:

Hopefully you won't get this one! The computer tried to open a file, but failed and the internal error flag was not set to identify why it failed. Fix any obvious errors and try again.

File not found:

The file you tried to open was not found. The name is either incorrect or there is a more serious error. If this occurs during set up, contact Mark Roberts, otherwise it is almost definitely a typing or entry error of some kind.

Access disallowed:

The file you are trying to read or write has been protected. It is possible in many operating systems to protect files from being changed, and possibly someone has done this to one of the program files. Get the protection removed.

File already exists:

You are trying to create a file that already exists. This message is unlikely as the normal response is to question whether you want to overwrite the existing file.

Invalid Argument:

Some error in the file description or the way you have named it.

File read/write failure - End of File?:

An error occurred when trying to read/write data out to a disk, this would usually indicate a corruption of some kind of the file. It is also possible to get this message when you are first installing a new version of the program. In this case contact MRMC, there may be a simple handling for it.

Error occurred reading axis data to/from disc. Job write/retrieval aborted:

Some error has occurred when data was being read to or from the hard disk. This would also indicate some kind of data corruption. This can also occur if you are trying to read jobs created with a different version of Flair. Try again and if you have no success, contact Mark Roberts Motion Control.

Error occurred reading/writing data from/to Root. Job write/retrieval aborted:

When data is stored to disk with a job, the majority of the data is taken from the Root computer. If there is an error in obtaining this information, then this message will appear. Make sure that the system is not in a run state, and try again.

Job Check Error occurred reading data from file. Job retrieval aborted:
Job Check Error occurred - invalid job file. Job retrieval aborted:
Error occurred - invalid job file. Job retrieval aborted:

When a job is read in, it is checked to ensure that the job and the system have the same basic configuration and that the data is stored in a readable form. If this is not so, then you will get an error message. It means that you cannot read in the job, and it must have been stored from a different program or a different program configuration. The "Job Check" prefix means that this error was spotted while the program was testing that the data structure of the job was intact.

Error occurred - file corrupted, re-load program:

This is very similar to the above, but in this case it has been decided that the data read in may be suspect, and there could have been some data corruption. You are strongly advised to re-load the software. You could also try reading in the job with the Simple Read function as this will read in only the actual waypoints and not the entire job.

Could not open your selected output:

The output you selected to use could not be opened, select a new output.

General Errors

“Error Communication to the root box X, More than 5 errors are fatal – Retrying”

This means that the PC (or host computer) has tried to communicate with the Root computer and failed. The most common cause of this is the root computer has stopped working, often this is because someone has turned off the power or has disconnected a communications lead. There are many possible causes, and if the problem does not resolve easily contact MRMC with details. This message will disappear is communication is restored.

It appears that an axis board has ceased functioning

This indicates that the communications between the main transputer and the axis board has stopped working. Either a error has occurred on the axis board causing it to cease working (check the red error light on the axis boards) or there has been some interference to the communications pipe line, either the cables have been unplugged or damaged. If the fault corrects itself, the on screen message will disappear, if not you will have to re-boot. You are able to save the job you were running in this case.

Unexpected Interrupt.....: / Flair.exe has encountered a fault

This is a fault in the PC which has caused the PC program to crash. If this problem occurs frequently or repeatedly contact MRMC with details. When this happens however, the main program running in the root box will probably still be running. You can re-load the PC program without having to re-load the root program, and the whole system will come back up almost as though nothing had happened. (You may have to check that the move speed and frame counters are correctly set!)

Handling: You will have returned to the standard windows screen when this happens, open up the folder in which the Flair.exe program is and the double click on that program whilst holding down the “Ctrl” key. The move data will have been lost, but the axis positions should still be correct and one can simply read in the last saved job and continue from there.

If all else fails

Turn off the power to the root unit or Deric and to the PC, leave it a moment, and then turn everything back on.

Mark Roberts Motion Control Ltd.
Unit 3, South East Studios, Blindley Heath, Surrey RH7 6JP
United Kingdom
Telephone: +44 (0) 1342 838000
info@mrmoco.com
www.mrmoco.com