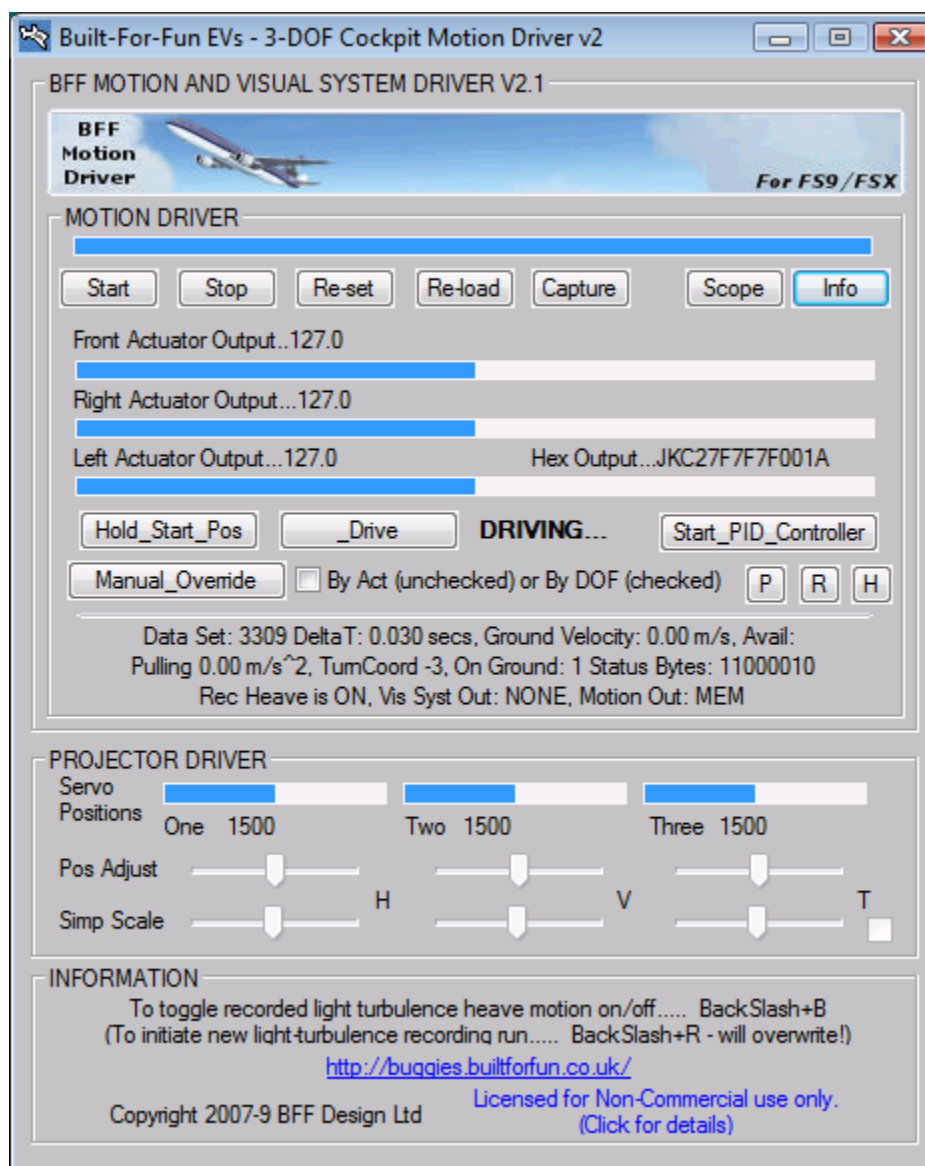


# BFF 3 Degrees of Freedom Motion Platform Driver



## User Manual

(For Driver, System Setup & PID Servo Controller Applications,  
Man. Rev 2.41)

For FS9/X (v2) & X-Plane 8.68/9.0/9.2 (v2) Driver Versions

Ian Hopper (BFF Design Ltd)

© 2007-2009 BFF Design Ltd, All rights reserved.

## Table of Contents

Recent Revisions List.....	4
BFF Motion Driver – User's Guide.....	5
1. What is it & what does it do?.....	5
2. How the platform driver works.....	5
2.1 Basis of the Motion Cue Calculation.....	6
2.1.1 Pitch.....	6
2.1.2 Roll .....	6
2.1.3 Heave.....	7
2.1.4 Coupling.....	7
2.2 Basis of the Articulated Projector Cue Calculation.....	7
3. Driver Installation & Operation.....	8
3.1 Installation.....	8
3.2 Platform Drive Data Output Formats.....	9
3.3 Motion Driver Operation.....	11
3.3.1 WINDOW BUTTONS.....	12
3.4 Articulated Projector Driver Operation.....	15
4. Driver Configuration and Setup.....	16
4.1 Motion Driver Settings (see section 4.3 for “coupled” setup where it differs from below).....	17
4.1.1 COM Port Settings.....	17
4.1.2 Driver Start Positions.....	17
4.1.3 Initial Transmit Delay.....	17
4.1.4 Pitch Cue Settings.....	18
4.1.5 Roll Cue Settings.....	20
4.1.6 Heave Cue Settings.....	22
4.2 Save Settings & Exit.....	24
4.3 Setup-3pt.exe For 3 Point Support Type Platforms.....	25
4.4 Articulated Projector Drive Setup.....	26
5. BFF PID Servo Controller.....	29
5.1 V2 Servo Controller Operation.....	29
5.2 Visual Display.....	30
5.3 Controller Settings.....	31
The Proportional Term.....	32
The Integral Term.....	32
The Derivative Term.....	32
Minimum Speed Demand.....	33
Setup Procedures.....	33
5.4 Servo Controller Data Output for Analysis and Inspection.....	33
5.5 PICAXE Chip Flash Programs for DIY servo hardware.....	34
5.6 40SPU-1 Signal Processor Card.....	34
LICENSE.....	35
Appendix 1 – Optional Data Output for Setup/Debug.....	36
Output Variables.....	36
Appendix 2 – Manually-set Config File Parameters.....	38
A2.1 Lateral Acceleration Scaling.....	38

A2.2 Vertical Acceleration Scaling.....	38
A2.3 Data Trace Enable.....	39
A2.4 Motion Platform Type.....	39
A2.5 Disable Booleans Output.....	39
A2.6 LAN Settings – Port and IP Address.....	40
A2.7 Software Drive Direction Reverse.....	40
A2.8 Runway Operations Heave Effects.....	41
A2.9 Articulated Projector Driver Configuration File.....	41
Appendix 3 – Data Output Settings for X-Plane.....	42
Appendix 4 – How to use the Shared Memory (MEM) output option.....	45
Appendix 5 – Key Flight Motion Events – Configuring the motion driver.....	46
Appendix 6 – Visual Motion Driver – Configuration Settings – Roll Example.....	48
Appendix 7 – Integration with Galil Optima Series, DMC-18x2 and DMC-21x2/3 industrial motion controllers.....	51

## Recent Revisions List

- July 2009 - Amended section 5 – Derivative term to explain PID Servo Controllers built-in filtering settings. Also clarified the PID2.cfg file line numbers where referenced in the text.
- June 2009 - Adjusted section 2 text for 40SPU-1 hardware.  
Adjusted section 3.1 text and other minor text changes.  
Added section 5.6 for 40SPU-1 card.  
Added text to 5.1 to recommend use of motion driver Hold\_Start\_Position and Drive buttons to control platform park/drive.  
Added text to section A2.8 – runway heave effects.
- April 2009 - Updated for driver version 2.1 and PID Servo Controller version 2.1.  
Updated Section 3.2 to add GAL data output format.  
Updated Section 3.3 to add oscilloscope and in-program cue settings features.  
Updated 5.2 to add oscilloscope feature to PID Servo Controller.  
Updated Appendix 1 – added trace=13 for GAL output mode.  
Added Appendix 7 – integration with Galil Position Tracking industrial controllers.
- March 2009 - Removed blank pages in section 3.3
- Feb 2009 – Updated page 33 – The Derivative Term  
Updated Section 5.5 – Updated options for SPU designs and flash programs.  
Updated Section 5.2 – Added Speed Controller Monitoring information.

# BFF Motion Driver – User's Guide

## 1. What is it & what does it do?

The BFF Motion Driver generates real-time digital **position demand** signals to drive the movements of a flight simulator motion platform or movement cockpit. V2 of the Motion Driver also provides separate motion cues for articulated projector displays (as described on the Built For Fun EV's web site).

These motion cues are derived from flight data extracted in real-time from Microsoft Flight Sims (FS9 or FSX) and Laminar Research X-Plane 8.68 / 9.2 flight simulators. Each Sim has a separate driver application.

The motion drive works for platforms with three uncoupled movement directions (Degrees of Freedom) – specifically for pitch, roll and heave movements, and also for platforms with 3 point support drives in which the three drive actuators' combined movements generate the DOF movements.

In its default configuration the motion driver assumes that movement in one degree of freedom does not affect the movement in either of the other two – ie it provides **uncoupled** outputs for each DOF. This can be changed to **coupled** 3-point support outputs by setting the Rig\_Type parameter in the configuration file to "1". Please note you do not have to use all three outputs – for example the driver can drive 2 DOF platforms by using only the uncoupled pitch and roll cues, or using actuators 2 & 3 in the coupled mode.

**NOTE** For 3-point support platforms Actuator 1 is the FORWARD actuator, Actuator 2 is the REAR RIGHT actuator and the Actuator 3 is the REAR LEFT actuator – please bear this in mind when reading the manual. This naming has been changed from previous versions to simplify the set up for 3-point type platforms.

Note also that for 3 point type rigs with two forward actuators and a single rear actuator simply reverse the pitch and roll cues in the configuration file using the Setup program.

## 2. How the platform driver works.

The V2 motion driver can run on the same PC as FS9/FSX **OR** on a networked PC. To run on a LAN PC the program BFF\_Motion\_Bridge must be running on the flight sim PC – see Section 3. The X-plane driver can run on the same machine as the X-Plane process or can be LAN networked.

The driver extracts aircraft movement flight data from FS9/FSX during your simulated flight. The extracted data includes information on aircraft instantaneous acceleration, velocity and orientation and this is used in the driver's cue algorithms to calculate appropriate force and movement cues for the motion platform to simulate the **EFFECTS** of the aircraft motion. The cues are output in the form of position demand values via one of the PC's serial (COM) ports or by other supported output formats. These can be read by external microcontroller based control units or other control devices to drive the platform motors. There is also the facility to write the cue data to Shared Memory where it can be read directly by other applications on the PC – this is how the BFF PID Servo Controller software reads the data as part of its motor speed and direction demand calculations.

It is probably helpful to state for the benefit of newbies to the subject that the **position** demand data output from the driver can not be used to drive the platform motion directly – it needs to be converted into **speed** demand signals and then amplified to drive the movement actuators on the platform. This usually involves an external position feedback control unit which outputs analogue (usually variable voltage) speed demand signals. In the case of an electrically actuated platform these speed demand

signals are then used to drive high current motor speed controllers which do the hard work of turning the electric motors on the platform.

The BFF 40SPU-1 Signal Processor Unit and control system is one DIY design for such a system. Details can be found on the Built For Fun EV's web site at <http://buggies.builtforfun.co.uk/> . There are several others that are possible but the 40SPU-1 Signal Processor Unit has been designed to interface directly with the BFF PID Servo Controller and the BFF Motion Driver - more of this later....

Versions 1.5 and later of the BFF Motion Driver support on-PC calculation of **speed** demands by the BFF PID Servo Controller. This software can be used with a much simpler external hardware set up to drive a motion platform.

The details of the data output format from the driver are described in the pop-up tool tips on the System\_Setup program and are described in detail in the Installation section 3 below.

## **2.1 Basis of the Motion Cue Calculation**

The motion driver primarily provides “force” cues which simulate the forcing effects on the pilot of the aircraft motion. These cues are probably most effective when used in an enclosed style motion cockpit – ie one in which the pilot can not see the actual movement of the platform and is unaware of his/her actual pitch and roll orientation. However the driver can also add-in pitch and roll angle elements to cue pitch and roll-rate effects and these facilities can also be used to emphasize the actual pitch and roll orientation of the aircraft if the user prefers this – eg for use in an open-cockpit “gaming” style simulation.

How are the cues calculated?

### **2.1.1 Pitch**

The pitch angle of the platform is driven by a combination of fore/aft acceleration force cueing and an element of pitch-rate effect. The “gravity alignment” element of the pitch angle is calculated from the fore/aft acceleration of the aircraft compensated for actual pitch orientation. This is used rather than the “simple” forward acceleration of the aircraft as it allows the fore/aft forces actually felt by the pilot to be determined. This gravity alignment element can be washed out or sustained for the duration of the accelerating event by adjusting the filter time constant. The cue is also smoothed using a low-pass filter which also gives control over the rate at which the force cues come on.

A user specified proportion of actual pitch angle is then superimposed to cue pitch rate effects and this rate element is also washed-out using a high-pass filter, the time constant for which can be set to vary the washout rate. The washout is used to limit the extent of false force cues which might arise from holding a pitch angle in the platform.

### **2.1.2 Roll**

The roll cue is derived using the same overall methods as the Pitch cue, however the parameter saturation levels and filter time constants are usually set differently to obtain different behaviour.

As with pitch there are two elements: a low-pass filter smoothed force cue based on the actual “felt” lateral forces on the pilot with a roll-rate element superimposed to cue roll-rate effects and match the sensation of rolling to the strong visual roll cues from the cockpit viewing screen. As before the force element can be washed out or sustained for the duration of the acceleration event; with lateral accelerations especially, these are likely to be of short duration anyway. The longest duration sideways-forcing events are likely to be taxiing turns and even these will last a few seconds only. Note that normal banked turns in an aircraft do not produce large sideways forces on the pilot.

To simulate the sensation of rolling the second direct roll angle based element is superimposed on the cue. This element is also washed-out out to mid to prevent sustained false force cues being generated. A high-pass filter is used for this and the washout rate can be controlled by varying the filter's time constant.

### **2.1.3 Heave**

The heave cue derivation is a bit more complicated than that for the pitch and roll movements. The complication arises from the natural limitations imposed by using a fixed stroke length motion platform to simulate the actual vertical accelerations. The sustained accelerations have to be cued using short duration impulse type movements which catch the onset of offset of the real acceleration event. To do this the local vertical acceleration of the aircraft is smoothed, scaled (if required), capped and washed-out to produce the position demand signal for the platform. The scaling is exponential (with a user specified variable exponent) and allows control event related accelerations to be emphasized over the normal continuous low level movements which arise from flying in light turbulence and variable air density conditions. These low level movements are left intact and are quite effective in giving the impression of continual in-flight movement.

The driver also provides the facility to toggle on/off pre-recorded light turbulence heave motion which can be superimposed on the live heave cue in circumstances where the simulator reports unrealistic flat-pan flying conditions. The same principal is used to generate more realistic runway movements during takeoff and landing on what might otherwise be reported by the sim as perfectly smooth runways. Both these add-in effects can be scaled or disabled completely by the user to suit their preference.

### **2.1.4 Coupling**

Two platform modes of output are available and these are set by the Rig\_Type parameter in the .bff configuration file. The default setting is Rig\_Type=0 and is used for rigs in which each movement DOF is driven independently – there is no cross-coupling between the outputs with this setting. Setting Rig\_Type=1 can be used for platforms driven by a symmetrical “triangle” of actuators with one at the front and two at the rear. In this mode pitch, roll and heave movement of the platform is obtained from the combined movements of the actuators, for example platform roll is obtained by raising the right side actuator and lowering the left side actuator. This signal mixing is done in the driver to produce coordinated drive signals for each actuator.

For 3 point type rigs with two forward actuators and a single rear actuator simply reverse the pitch and roll cues with the Setup program.

The configuration settings governing range of movement etc will be different for these two types of rig so a config file set up for one is not likely to work properly for the other..

Please note that the Rig\_Type parameter must be set manually in the configuration file – it can not be set through the Setup program. There are a number of other manually set parameters in the configuration file and these are described in Appendix 2.

NOTE For 3-point support platform mode the Actuator 1 is the FORWARD actuator, Actuator 2 is the REAR RIGHT actuator and the Actuator 3 is the REAR LEFT actuator – please bear this in mind when reading the manual. This naming has been changed from previous versions to simplify the set up for 3-point type platforms.

NOTE ALSO that there are now two separate Setup programs – one for uncoupled (independent) rig set ups and one for coupled (3 point support) type rigs – see section 4.

## **2.2 Basis of the Articulated Projector Cue Calculation**

The movement cues for the articulated projector drive are generated from the actual aircraft in-flight velocities. Projector horizontal movement is determined from aircraft lateral and yaw velocities, vertical movement from aircraft heave and pitch velocities and tilt movement from aircraft roll velocity. The cue processing is configurable to vary movement range, speeds, washouts etc through the settings in the vis.cfg configuration file associated with the main driver configuration file.

Section 4.4 gives more detail and Appendix 6 shows an example.

## **3. Driver Installation & Operation**

### **3.1 Installation**

For MS sims you will need an up-to-date version of FSUIPC installed for the FS version you are using. FSUIPC3 for FS9 and FSUIPC4 for FSX, note the driver may well work on some versions of Combat Flight Simulator but I haven't tested this. Please read carefully Pete Dowson's installation procedures especially those for FSX – they can be found on his forum which is accessible from the FSUIPC web page -

<http://www.schiratti.com/dowson.html>

Your copy of FSUIPC does not need to be registered to run the BFF Motion Driver.

For MSFS versions -

The motion driver package should be unzipped to a dedicated directory on the same machine as FS is running or on a LAN PC. If you are operating the driver over a network the BFF Motion Bridge software must be running on the sim PC – the .exe is included in a separate zip file in the main zip archive and this Bridge zip file must be copied to the sim PC and its files extracted to a dedicated directory. The Bridge.cfg configuration file on the sim PC must be edited to set the IP addresses of the sim and networked PC's and the PortS and IPAddress settings must be configured in the .bff driver configuration file on the LAN PC.

Although the motion driver itself takes up little computing resource the PID Servo Controller software is more CPU intensive and may affect frame rates on less powerful machines running FSX. If you intend to use the BFF PID Servo Controller with the BFF Single Chip SPU hardware you may wish to consider using a separate PC to drive the motion platform if you do not have a powerful PC.

For X-Plane versions -

For X-Plane versions there are no third party utilities required. The driver receives data directly from X-Plane using UDP protocols. It is necessary however to set up the data outputs in X-Plane and this is described in Appendix 3. The driver package should be installed on the machine on which you want it to run.

OPERATION – described in more detail in Section 3.3

The main working programs are the setup programs System\_Setup.exe (there are two versions of this – one each for uncoupled and 3-pt support type rigs) and the motion driver itself BFF\_Motion\_Driver.exe. These should be run from their installation directories. If you wish, desktop shortcuts can be set up to point to these .exe's, however do not remove them from the other files in the package. They will look for the other working files in the same directory and will not work properly if separated from them.

The operation of the BFF PID Servo Controller is described in section 5.

The motion driver can be configured using the appropriate System\_Setup program. This will read and save the settings to a .bff configuration file in the driver directory. You can specify the name of the configuration file through the open/save dialogue and can save as many different config files as you wish so as to setup different configurations for different aircraft or flying styles.

V2.1 and later of the driver also allow in-program setting of the main motion cue parameters – see section 3.3.1 below.

To run the motion driver on the sim PC first start FS in windowed mode, then run the motion driver executable. Follow the screen prompts to select your config file. The driver will suspend output until a flight is active and will further suspend output when the flight is paused or otherwise interrupted. In



either case the “Hold Start Pos” and “Drive” buttons are used to hold output at the specified start position or to commence live output under the pilot’s control.

If you are using the BFF PID Servo Controller software and hardware then it is best to set the PID Servo Controller to “Engage\_Drive” first and then use the BFF Motion Driver Hold and Drive buttons to control the platform. This will give smoother “start drive” and “park” behaviour.

To run the MSFS driver on a LAN PC proceed as before and you will be prompted to confirm you are running on a LAN PC when the driver fails to detect the flight sim on the PC. Once you have confirmed the driver will open a UDP port and listen for flight data exported from the main sim PC by the BFF Motion Bridge program which you should run on the main PC. If no data is available the driver will suspend it’s activity and will indicate “waiting for data...”.

To run the X-Plane version you can start it before X-Plane and it will wait for data, or start X-Plane first and then the driver after – this is cleaner and avoids the driver locking up as it waits for UDP data packages which never come. You may have to hit the “Windows” keyboard key to bring up the task bar if you run a full-window setting for X-Plane.

### 3.2 Platform Drive Data Output Formats

In v2.1 and later there are four data output formats available – binary, hexadecimal, shared memory and the Galil DMC Controller format. See the System\_Setup section to see how to set these in the configuration file.

For the BIN, HEX2 and MEM output formats each position demand value is output as a uni-polar value in the range 0 to 255. When no working sub-range is specified this gives a value of 127 as the zero position, 0 is max negative cue and 255 is max positive cue. A working sub-range can be specified – see the System Setup section.

For the GAL Galil DMC controller output format the position demand range is not limited and is set to correspond to the feedback encoder count range used in the control system.

For binary (BIN) mode the data is transmitted in binary 8 bit bytes. For hexadecimal (HEX2) each number is transmitted as two characters giving the 2 character hex value of the number eg FF for Hex 0xFF. The shared memory (MEM) output option uses the same data format as the HEX2 option but rather than write to a COM port it writes to a Windows Shared Memory area where it can be read directly by other applications. In GAL mode the driver communicates directly with the Galil Optima Series controller – see Appendix 7 for details. A fifth option is available (NONE) in which the output COM port is not opened and no data is sent – this is useful if you just want to try the program and do not want it to interact with your COM ports.

For details of how to use the Shared Memory (MEM) output option see Appendix 4. When using v1.5 and later of the driver with the BFF PID Servo Controller the output mode must be set to MEM.

BIN output format is - “AB” byte1 byte2 byte3 byte4 CR

“AB” - start of data identifier for the receiving (PICAXE) micro controller chip

byte1 - code for SPU (ignore if you don't use the BFF SPU)

byte2 - 8 bit binary number giving pitch demand in 0-255 scale

byte3 - 8 bit binary number giving roll demand in 0-255 scale

byte4 - 8 bit binary number giving heave demand in 0-255 scale

0x0D - single byte Carriage Return data terminator

(byte1 is a code set to 1 to tell the older BFF 4 Chip SPU to process the position data normally, otherwise code = 2 pitch to slow, = 3 roll to slow, = 4 heave to slow, 5 = all DOF's to slow. Additional values of 6, 7 & 8 are used by the system setup program to set the SPU 18X chips.)

HEX2 (and MEM) output format is - "JK" JJ XX YY ZZ 0x0D

"JK" - Start of data identifier for the receiving micro controller chip

JJ - 1 byte - 8 bits, each indicates on/off for status indicators - see below

XX - hex value of pitch demand in 0-255 (0x00 to 0xFF) scale

YY - hex value of roll demand in 0-255 (0x00 to 0xFF) scale

ZZ - hex value of heave demand in 0-255 (0x00 to 0xFF) scale

RR - hex value for current forward ground speed in m/s, capped at 255

SS - hex value for engine 1 rpm, 0 -100 percent

1 byte - "CR" data terminator

The JJ bits are -

b7 (msb) flaps

b6 landing gear

b5 brake indicator

b4 spoiler (not currently available in X-Plane version)

b3 overspeed (not currently available in X-Plane version)

b2 stall

b1 on ground

b0 (lsb) not used

The serial output format is 8 bit, 1 stop bit, no parity to the COM port stated at the baud rate stated in the configuration file.

Note the BIN output format is the more efficient and uses approximately half the data length of the HEX2 format to carry the same information. This might be significant if you want to use low transmission baud rates.

The driver works at baud rates up to 115200. For the MEM output the data is written to the shared memory area at the refresh rate of the Driver.

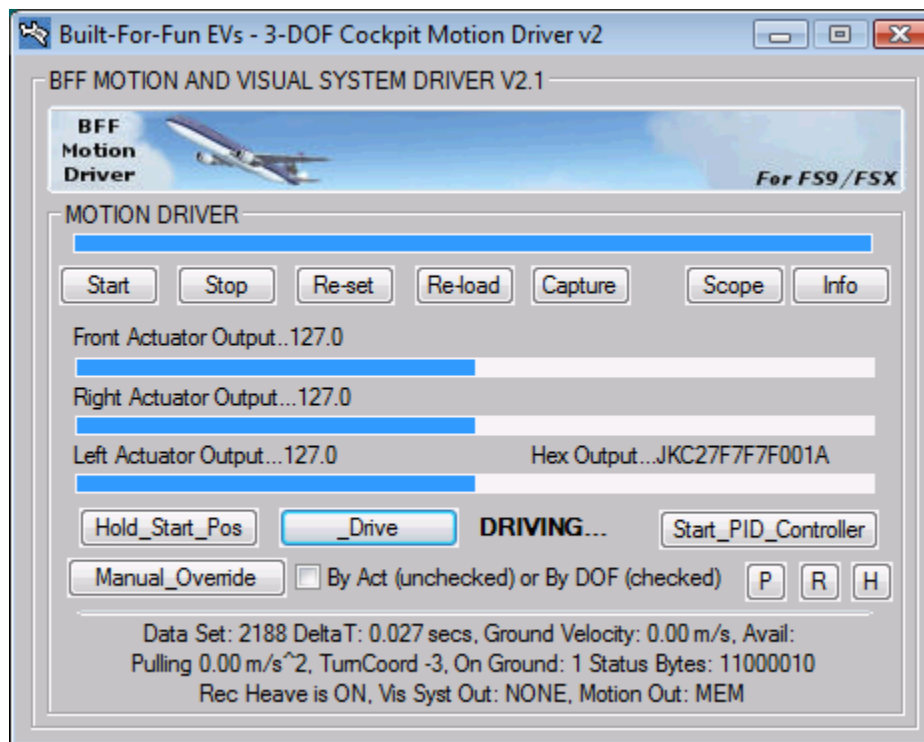
For GAL output format the driver sends its position demands directly to the Galil Optima Series controller using the Galil DMCwin32.dll Windows Toolkit API. Communications will be by either Ethernet or Serial comms to match the installation of the DMC controller. See Appendix 7 for details.

NOTE 1, there is an optional parameter "itrace" that can be set in the .bff config file to instruct the driver to store in-flight cue data and then dump it to a text file when the driver is closed. This provides a record of the data and the calculation steps for inspection and/or charting purposes and is useful when setting the driver up for a particular motion platform. Details can be found in Appendix 1 at the end of the manual.

NOTE 2, there is a further manually-set parameter in the configuration file - Use\_Boot. If Use\_Boot=0 the JJ bits and final SS engine RPM byte are disabled. The b1 (on-ground) bit of the JJ byte will alone be active – the rest will always be zero, and the SS byte will also always return zero. This setting can be used to slightly ease the processing load if these outputs are not required. When Use\_Boot=1 (default setting) all the output data is active in the HEX2 mode. This parameter has no effect in the X-Plane version.

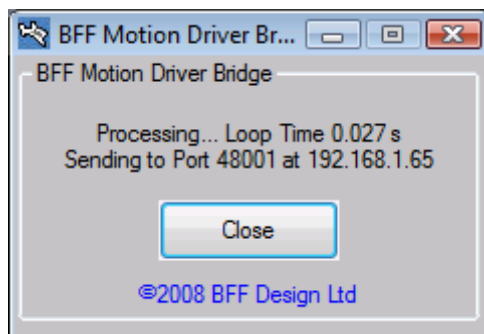
### 3.3 Motion Driver Operation

For operation of the Visual System Driver please see section 3.4.



To start the driver click Start; for unregistered versions you will be given an option to register or continue in Demo mode. Registered versions are tied to the PC from which they were registered – if you wish to run the software on a 2<sup>nd</sup> PC go through the registration process again on the 2<sup>nd</sup> PC. If you continue in Demo mode the software will run as normal but will close after about 10 minutes processing time.

Continuation will bring up an open file dialogue from which you can select the particular configuration file you want to use. Once the settings from the file are loaded the driver performs some checks to find FS9 or FSX, opens a connection to FSUIPC, opens and configures the data export COM port and initializes the cue calculations. If a flight is active the driver will commence the cue calculations, if the simulator is paused or a menu interaction is active the driver will suspend its calculations until the flight resumes. The driver starts with the “Hold Start Pos” button active – ie its output will hold the platform at the specified start position. When you are ready click “Drive” and the driver will resume live position demand output.



If you are using the MSFS v1.7 driver (and later) on a LAN networked PC separate from your main sim PC then you will be prompted to confirm this when the driver fails to detect FS9/FSX on the PC on which it is running. On confirmation the driver will open a UDP socket to listen for flight data exported by the BFF Motion Bridge program which you should run on the main flight sim PC.

There are some configuration settings that need to be made to get the BFF Motion Bridge running. Firstly, the IP Addresses of both PC's need to be set in the Bridge.cfg file on the main sim PC. The .cfg file is included with the .exe in the Bridge .zip archive and should be left in the same directory as the .exe. Secondly, the PortS and IPAddress parameters must be set in the motion driver's .bff configuration

file on the motion driver's PC. Of course the settings in these two files must match where appropriate – particularly the port numbers and the motion driver's PC IP address. - Tip, leave the port numbers at their default settings and just change the IP addresses to suit your network.

For an active flight the drive buttons will be suspended until the number of calculation cycles specified in the configuration file have elapsed – this is to allow the washout and smoothing filters in the algorithms to stabilize. Thereafter the “Drive” button goes active. The cue outputs are displayed both digitally and in the analogue progress bars – see above.

### 3.3.1 WINDOW BUTTONS

**Drive** - resume active platform drive.

**Hold\_Start\_Position** - return the platform to its “start” or parked position. In this mode a constant stream of position demands are output set to the hold position of the platform actuators.

**Stop** - stop and close the driver (or hit Tab).

**Re-Set** - re-start with the current configuration file.

**Re-Load** - re-load a new configuration file.

**Info** – show/unshow the full driver window with information.

**Start\_PID\_Controller** - start the BFF PID Servo Controller application – this will require the data output mode to have been set to MEM, otherwise the Servo Controller will not start. See Section 5 for PID Servo Controller operation.

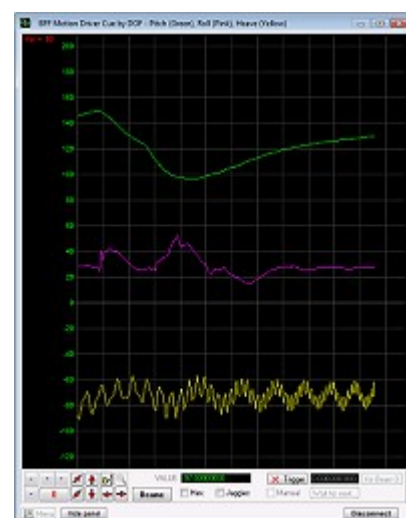
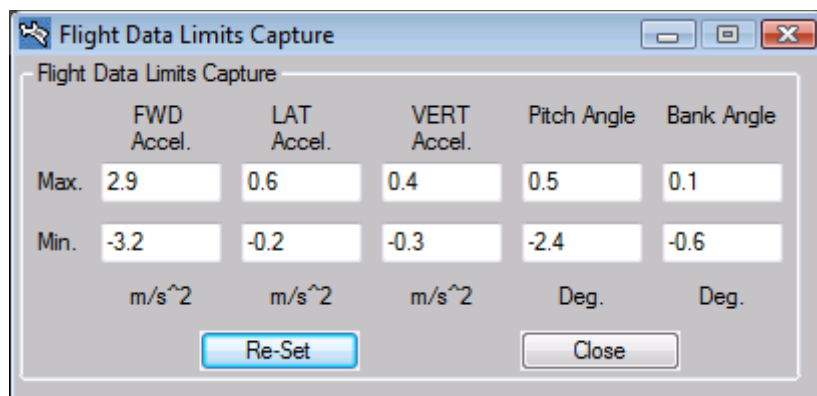
**Capture** - capture the maximum and minimum flight accelerations and pitch and roll angles.

The Flight Data Limits Capture window allows you to capture the recent maximum and minimum values of key flight parameters. These include forward, lateral and vertical accelerations and pitch and roll angles and good knowledge of these is needed to set the configuration settings for the driver.

Use the capture button to capture the accelerations associated with key flight events such as takeoff accelerations, braking decelerations, vertical lift-off accelerations, turbulence induced accelerations etc etc. With this information you will be able to better judge the acceleration cap levels and scaling to the platform movement range.

**Scope** – activate the built-in virtual oscilloscope to display real-time traces of the motion cues. The traces are of the raw motion cues before any mixing or management processing is done by the driver. The output will be active so long as the flight sim is not paused and will allow the active motion cues to be displayed even with the driver on **Hold**.

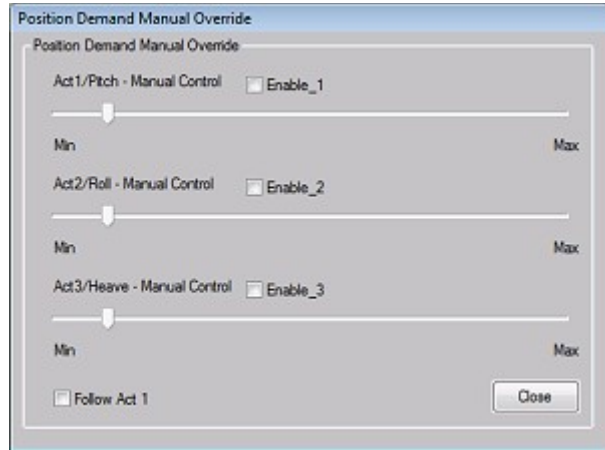
The oscilloscope can be used with the in-program cue setting windows (see below) to assess adjustments to the motion cue settings whilst the driver is running and so allow the effects of changes to be assessed rapidly. **IT IS NOT RECOMMENDED THAT THIS IS DONE WITH THE PLATFORM DRIVE ACTIVE!**



The virtual oscilloscope is implemented using Michael Bernstein's excellent Software Oscilloscope DLL – see his web site at <http://www.oscilloscope-lib.com/> for more details. The traces can be manipulated on the scope using the buttons provided as is normal for any oscilloscope.

**Manual Override** - manually override the position demand outputs from the driver.

This will open the Override window which has three sliders – one for each output channel/DOF of the driver. Moving a slider will adjust the value of the position demand output for that Actuator or DOF within the **full 0-255 range** if the “Enable” check box has been checked for the Actuator. If the “Follow Act 1” box is checked than all three sliders will follow the position of the Actuator 1 slider.



If the “By DOF” checkbox is checked on the main UI the output will be by DOF and not by individual actuator. For coupled drive platforms this will result in all the actuators moving as required for the selected DOF.

**WARNING** – Because this feature is intended as an aid to establishing the working range of a platform the sliders operate within the FULL 0-255 position output range. **Use with great care** – if your platform drive is active the platform may move suddenly in response to movements of the sliders, if you demand a position out with the physical working range of your platform the platform will drive onto its end stops.

The sliders can be moved with the mouse or more sensitively with the left/right keyboard arrows.

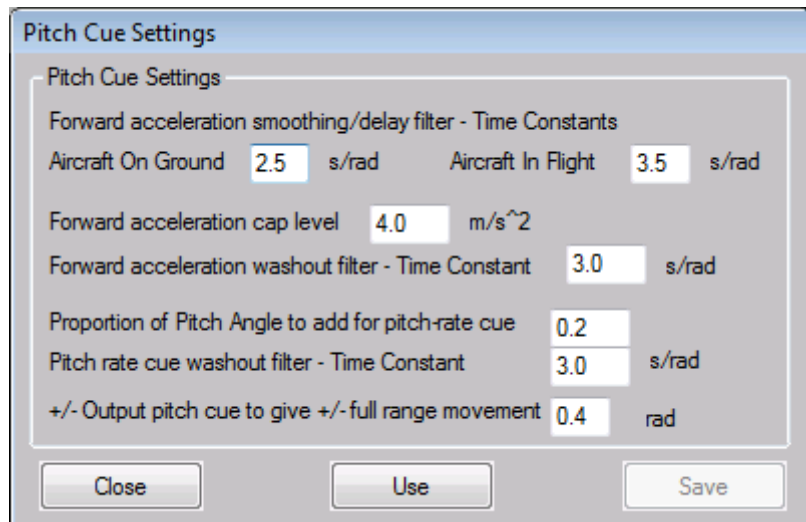
**In-Program Cue Adjustment Settings** – there are three small buttons provided - **P**, **R** and **H** to open the settings windows for the Pitch, Roll and Heave degrees of freedom. For details of the individual settings see section 4.1.

For illustration the Pitch Settings window is shown right -

Opening either of the three settings windows will send the driver output back to the “hold” position to disable live output of the cues whilst the settings are being adjusted.

To adjust a setting make the change in the appropriate text box and click **Use**. This will make the adjusted settings active in the driver but will not save them to file.

To save the new settings to the active configuration file click **Save**.



**IMPORTANT** – Take great care when making changes to the cue settings as they can directly impact on the motion range. It is recommended that the adjusted cue output is viewed in the Virtual Oscilloscope and its stability assessed before the driver is put back in **Drive** mode and the platform made active.

There are some HotKeys programmed -

TAB – will close the driver from the keyboard, useful if you want to quickly cut the output with a single key-stroke.

\+B (Backslash + B) – will toggle the pre-recorded light turbulence heave motion on or off – the MSFS driver starts with this ON. The effect can be scaled or disabled completely by adjusting the value of the Turb\_Scale parameter in your .bff configuration file (set =0 to disable).

\+R (Backslash + R) – will initiate a new heave motion recording run. This will last two minutes and will end by closing the driver and writing the newly recorded data to the BFF-Turb.txt file in the driver's working directory. This will overwrite the existing file, so back it up if you want to keep the original data supplied in the driver zip package. To use this facility set up a flight in your aircraft of choice and fly straight and level – best to use the auto-pilot. Set the weather conditions to remove all weather effects except an additional turbulence level – you chose what level of turbulence in the User Defined Weather settings. The aim is to get a 2 minute run with turbulence induced heave movement alone which can later be superimposed on the live heave motion cue.

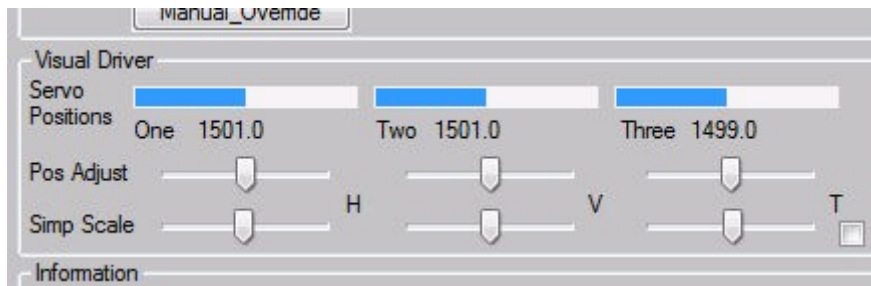
The motion driver can also generate additional runway heave effects to add more movement during take off and landings to make up for the limited motion reported by FS9/FSX during runway operations. The runway effects are generated from a combination of two sinusoidal waves – a slower frequency heave wave which increases in amplitude and frequency proportional to speed to simulate slower aircraft movements and a higher frequency wave to simulate wheel rotation/imbalance. This faster wave increases in frequency proportional to wheel rotational speed. The base amplitudes and frequencies can be set in the .bff configuration file – see Appendix 2 for details.

Both effects can be scaled or disabled using the appropriate settings in the Driver\_Setup program.

Links to the license details and to the <http://buggies.builtforfun.co.uk> web site are provided on the interface – pay me a visit. You can contact me by email at [lan@builtforfun.co.uk](mailto:lan@builtforfun.co.uk)

To show/hide the above information click the **Info** button.

### 3.4 Articulated Projector Driver Operation



The Projector Driver area of the window will be shown if the visual system output is mode is not set to NONE in the vis.cfg configuration file.

The Articulated Projector Driver element of the software exports three servo position demands through a serial COM port specified in your vis.cfg configuration file. The servos are part of a projector drive mechanism such as that described on the Built For Fun EV's and should be driven by a Pololu servo controller or other servo control hardware.

The Articulated Projector driver output is displayed in the "Visual Driver" section of the driver window. Three progress bars display the current position demands sent to the three projector mechanism servos. Fine position adjustment sliders and simple output scaling sliders are provided for each of the horizontal, vertical and tilt movement DOF's of the visual system driver and allow some fine tuning to be made whilst the drive is active. The small check box on the lower-right corner of the area disables the visual driver output when checked.

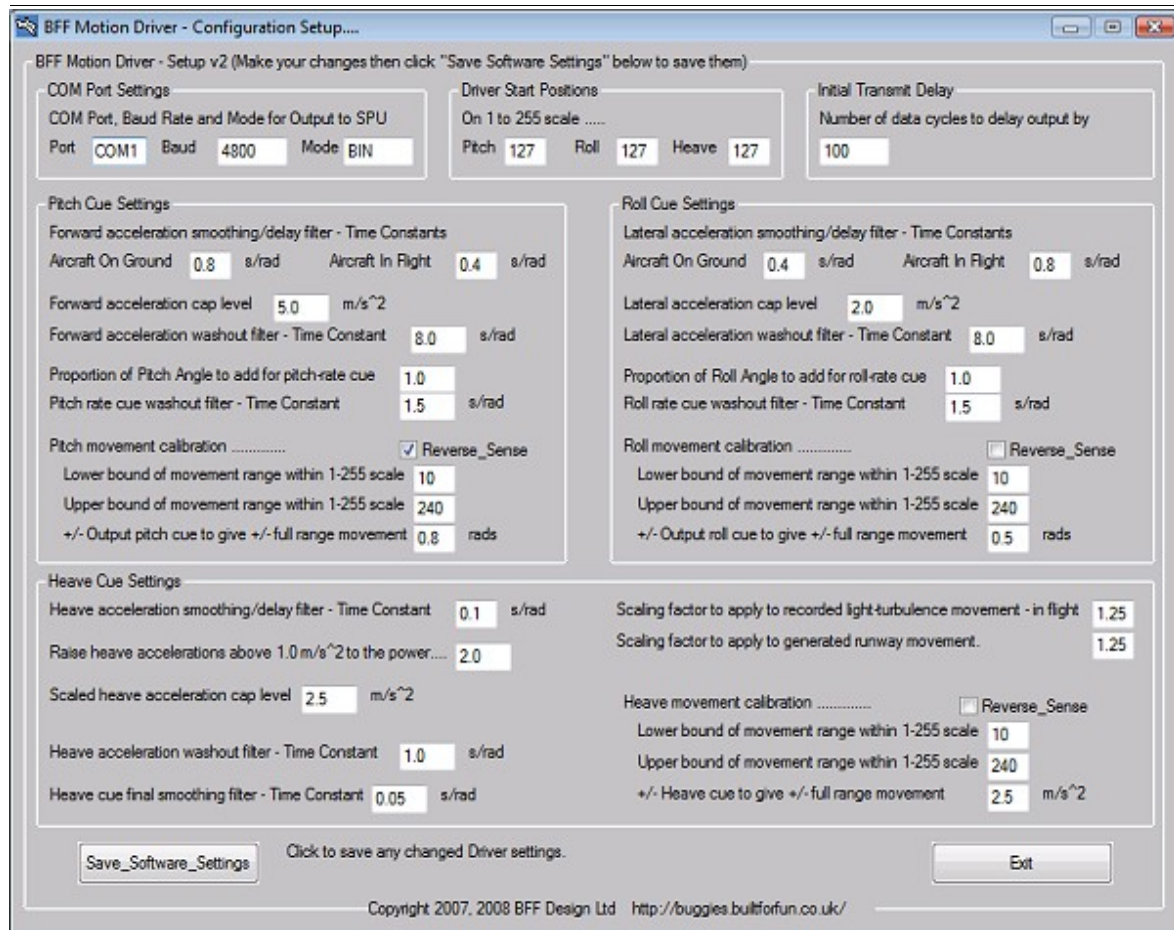
The detailed settings for the visual driver are held in the vis.cfg file specified by the vis\_fname parameter in the current .bff config file. The vis.cfg file contains detailed cue processing parameters which affect the magnitudes, directions, smoothness, washout times, mixing proportions etc for the projector motion cues – if you wish to adjust these to experiment with the cues then keep a copy of the original settings and take care – read section 4.4 carefully.

A setting you will need to specify is the serial COM port to be used for the cue output to the servo controller board.

If you do not wish to use the Visual System Driver then the output mode in the vis.cfg file should be set to NONE.

## 4. Driver Configuration and Setup

Two setup programs which run separately from the Motion Driver are available and these allow you to configure the setup parameters for the driver. The setup programs do this by reading from and writing to a \*.bff configuration file which is subsequently read by the motion driver at start-up. Default configuration files are included in the package which should be modified to suit your set up.



### Driver Configuration Interface

There are **two** different set up programs – one to deal with the driver settings for **uncoupled** (independent) DOF rigs and one to deal with **coupled** (3 point support) type rigs. The two programs are **Setup-3pt.exe** and **Setup-ind.exe**. This change has been made to simplify the set up for the 3 point type rigs. **Section 4.3 describes the Setup-3pt.exe parameters where they differ from the Setup-ind.exe items.**

To run the uncoupled rig type setup program double click on the **Setup-ind.exe** file. A file open dialog will be displayed from which you can chose the configuration file you wish to inspect or modify.

If you cancel the dialog box without opening a file the driver setup program will close. Once the configuration data has been read and loaded into the program you can go on and edit the individual values.

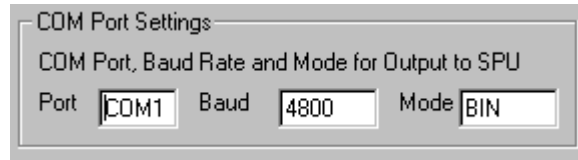
NOTE There are a number of parameters in the .bff configuration file that are not set through the setup programs. See Appendix 2 for details.



## 4.1 Motion Driver Settings (see section 4.3 for “coupled” setup where it differs from below)

Starting at the top left of the setup program window.....

### 4.1.1 COM Port Settings



COM Port Settings  
COM Port, Baud Rate and Mode for Output to SPU  
Port  Baud  Mode

Port number, baud rate and data output Mode. The output is always 8 bit, no parity, 1 stop bit. The output Mode is as described in the Installation section above and can be BIN, HEX2, MEM, GAL or NONE. If you input something else the mode will default to BIN. An output test can be carried out by connecting the output COM port to a second port on the same PC using a Null Modem cable and then opening a HyperTerminal connection to the second COM port. The connection will need to be configured correctly to suit the driver output format. If data is being output it should appear as a data trace in the HyperTerminal window. Bauds up to 115200 are possible.

For use with the BFF PID Servo Controller the Mode should be set to MEM and the Port set to the COM port to be used by the Servo Controller to communicate with the single chip SPU.

The download package includes a simple test script to test the Shared Memory (MEM) option. For more details see Appendix 4.

### 4.1.2 Driver Start Positions

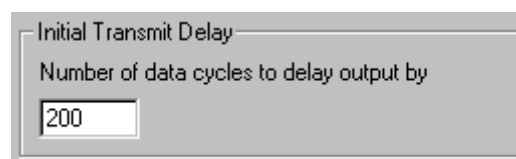


Driver Start Positions  
On 1 to 255 scale .....  
Pitch  Roll  Heave

Set these to define the positions on the 1-255 scales where the driver should first position its output. This is where the platform will first be sent until the delay period is over and the driver commences live output based on the flight situation.

The 1 to 255 scale can be interpreted as representing the max possible movement range described by the full scale output of the position feedback potentiometers on the platform. (The 18X driver chips on the BFF SPU for example 'see' the 0 to 5V output range of a pot as a 0-255 scale.) However the platform may not be physically capable of using this full range of output of the potentiometers and may reach its physical travel limits before the pots reach their end of rotation. It depends on how the pot rotation is geared to the rig movement. Accordingly the actual travel range of each DOF within the possible 1-255 scale can be set separately (see 'movement calibrations' below). This 'Start Position' setting indicates to the Driver which position to send the rig to when the Driver starts up. It should logically be set within the actual working range of the rig as defined for the particular DOF otherwise the movement may drive onto the end stops when the Driver first starts up. It makes some sense to coordinate the driver start position with the SPU start position. The overall approach provides the means to make a significant degree of adjustment to the platform's position setup without having to electrically calibrate the potentiometers to the physical movement.

### 4.1.3 Initial Transmit Delay



Initial Transmit Delay  
Number of data cycles to delay output by

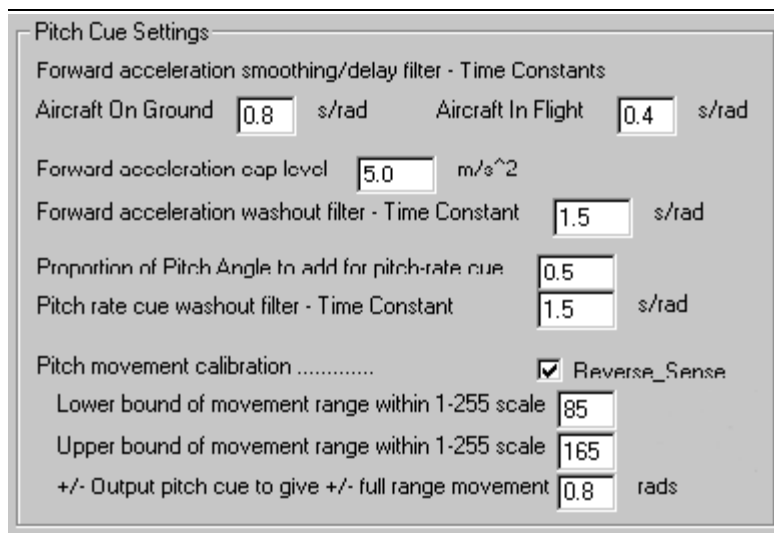
The motion cue algorithms can take

a finite time to stabilize after starting up. This is principally due to the software filtering used to induce wash-outs or noise filtering in the cues. To protect the platform during this stabilization period the data output can be suspended until a set number of calculation cycles has completed. The default is 200.

If you make changes to the time constants used in the algorithms it is wise to make a few test runs without the motion platform powered up and to watch the stability of the cues in the progress bars on the driver interface. If an output appears to jump to an extreme position following start-up and then slowly return to mid it may be appropriate to increase the number of delay cycles.

#### 4.1.4 Pitch Cue Settings

The pitch movement of the platform is used primarily by the driver to generate force cueing related to the aircraft's combined fore/aft acceleration and pitch angle. It can also cue pitch angle directly; an element of the pitch angle can be added-in to cue pitch rate, this pitch-rate element is washed-out out to ensure that sustained false force cues are minimised. The acceleration related fore/aft force cue can also be washed out ie the force effect duration of runway acceleration or in-flight decelerations can be adjusted. These do not usually last for more than a minute or so at most and my preference for the driver was to sustain the fore/aft forcing effects for the duration of the flight activity causing the effect, however reducing the washout filter time constant will introduce a speedier washout if required.



Pitch Cue Settings	
Forward acceleration smoothing/delay filter - Time Constants	
Aircraft On Ground	0.8 s/rad
Aircraft In Flight	0.4 s/rad
Forward acceleration cap level	5.0 m/s <sup>2</sup>
Forward acceleration washout filter - Time Constant	1.5 s/rad
Proportion of Pitch Angle to add for pitch-rate cue	0.5
Pitch rate cue washout filter - Time Constant	1.5 s/rad
Pitch movement calibration .....	<input checked="" type="checkbox"/> Reverse Sense
Lower bound of movement range within 1-255 scale	85
Upper bound of movement range within 1-255 scale	165
+/- Output pitch cue to give +/- full range movement	0.8 rads

Starting at the bottom of the input area...

"Pitch movement calibration". See the notes on the "Driver Start Positions" above – the lower and upper limits of the actual platform movement within the full 0-255 possible output range are set here.

" +/- Output pitch cue to give +/- full range movement". The above range settings will represent a particular +/- angular movement of the pitch DOF. If you want the output "gravity alignment" force cues to give an accurate simulation of the actual fore/aft acceleration forces of the flight then set this equal to the actual +/- angular range of the platform's pitch movement. The units are radians, 0.8 radians is about 45°, so the default 0.8 setting instructs the driver to fit a +/- 45° pitch angle demand to the actual pitch movement range of the rig. If the physical movement range is also +/- 45° (the 80-160 range is +/- 45° on my rig) then there will be a 1 to 1 mapping between the "felt" acceleration forces in the motion cockpit and the "felt" inertial acceleration forces of the actual flight. Off course these are subject to upper limits – as stated next. The calibration can be used to either exaggerate or reduce the actual felt force effects in comparison to the "real" forces of the flight.

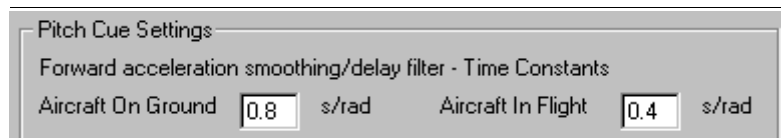
"Forward acceleration cap level". It isn't possible to simulate the full range of fore/aft acceleration forces – especially those experienced in high-g jets. A pitch angle of 45° on the rig will produce a felt fore/aft force equivalent to an acceleration of 0.707g (sin 45°) ie ~7.1 m/s<sup>2</sup>. You can set here an upper limit acceleration to be used for the cue calculation. Logically you can set this = sin(+/- movement angle) \* 9.81 where you have set up a 1 to 1 calibration and you want to use the full movement range. The 5.0 m/s<sup>2</sup> value shown limits the actual accelerations to about 0.5g and so

limits the actual pitch movement to +/- 30° (0.52 rads) even though the available working range is +/- 45°.

**In general a forward acceleration of x g's will produce a force cue angle of roughly x radians. So to fit, say, the force cue from a 0.8g acceleration to the full motion range of the platform set the “+/- output pitch cue to give +/- full range movement” to 0.8 rads and set the acceleration cap level to 8.0 m/s<sup>2</sup> (0.8g)**

**Use the Capture button to capture the actual maximum and minimum accelerations experienced in the flight during important flight events (see Appendix 5).**

“Forward acceleration washout filter – Time Constant” . If desired the forward acceleration force element of the cue can be washed out to mid. The washout rate can be slowed by increasing the time constant.



#### Pitch cue low-pass filter Time Constants.

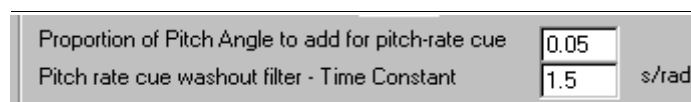
A low-pass software filter is used to cleanup and smooth the fore/aft acceleration value for the cue. The “time constant” used in the filter sets how much of the higher frequency components (noise) is removed. Shorter time constants cause more of the higher frequencies to be passed through untouched. A further effect of the low-pass filter is to slow the response of the filtered output, and longer time constants cause longer delays as only the lower frequency changes make it through the filter. These behaviours are very useful for motion cueing – they can be used both to remove noise from an acceleration trace and to delay the effects of acceleration changes.

For the pitch cue it is actually the delaying behaviour that is mostly used. Ground manoeuvring braking acceleration changes can be quite sharp and allowing the pitch angle position to change as rapidly as the actual acceleration rate is likely to generate a felt pitch-rate cue in the motion cockpit that is false. In flight the rate of change of fore/aft accelerations tends to be slower and less delay is required in the pitch cue to avoid false pitch rate cues getting through. For this reason two time constants can be set, one each for taxiing and in-flight operations.

The best way to set these is by trial and error – test to discover what feels right for you and your platform.

It is worth making the general point again here that the fore/aft force cue is not the “simple” acceleration based solely on the apparent change in the aircraft's forward velocity. If you throttle the engines back and point the aircraft vertically nose down it will accelerate alarmingly, but as both the aircraft and the pilot are effectively in free-fall the pilot will not “feel” strong forcing effects associated with this simple acceleration. The apparent inertia force due to simple velocity change must be compensated for the actual pitch angle orientation of the aircraft to determine the true “felt” fore/aft force acting on the pilot and hence an appropriate force cue for the motion platform. The net result of this is that for a lot of the time the force cue appears to be more closely related to the magnitude of the thrust being produced by the engines – rather than the actual pitch angle of the aircraft. As the speed and hence drag increases the accelerating effect of the engine thrust reduces and there is less of a relationship but, be warned, the behaviour of the force element of the pitch motion cue is not always what you might expect.

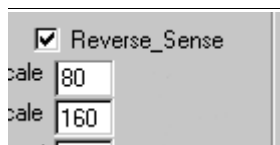
#### Pitch Rate Effects



Two parameters control these. Firstly the proportion of the actual pitch angle to pass through to the cue can be set. Inducing a feeling of pitching in the correct direction does not necessarily require that

the platform pitch angle matches exactly the full pitch angle of the aircraft. It may be sufficient only that it pitches at a noticeable rate and in the correct direction. To an extent the strength of this effect depends on whether or not you have an enclosed platform – open cockpits may require an enhanced pitch angle response to satisfy the expectations of the user. If the proportion set here is too large there is a risk that the cockpit pitch angle will be sustained long enough to generate a false fore/aft force cue. The effects are related to the second parameter, the high-pass filter time constant.

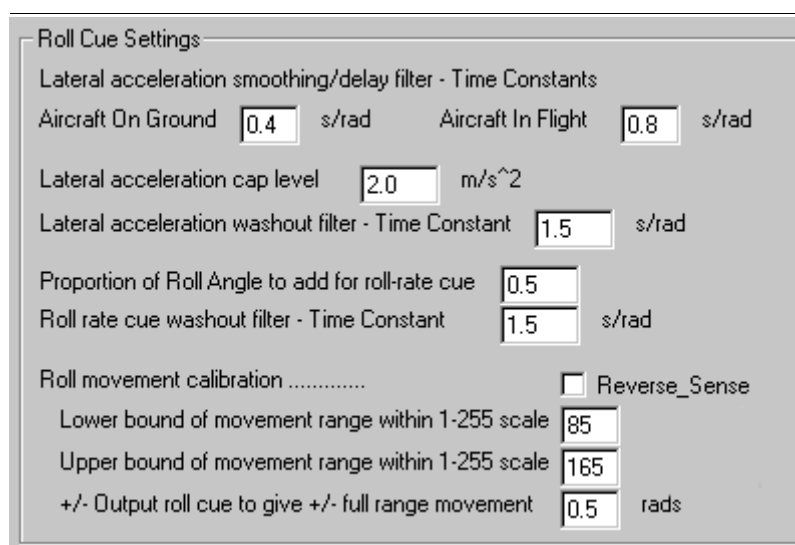
A high-pass filter allows the higher frequency elements of a trace to pass untouched but filters out the lower frequency changes in the values. This has the effect of always returning the trace to zero when there are no high frequency movements taking place – ie of washing-out the movement. The longer the time constant the slower the washout speed. The setting above of 1.5 s/rad (s/1) brings the pitch angle back to zero in 2 or 3 seconds or so – slow enough for the pilot not to notice it. Again you can experiment with these settings to generate different effects. The 0.05 setting shown for Pitch Angle Proportion illustrates how the pitch-rate effect can be disabled by setting the value low if you wish only to use the acceleration based force cueing element.



Lastly for the pitch cue setup is the Reverse\_Sense checkbox. This provides the means to reverse the direction of the motion cues. Mainly of use when matching what should be +ve movement cues with the actual +ve direction of movement of the platform. For example a positive forward acceleration should produce a backwards pitch of the platform. On my motion cockpit positive drive to the pitch DOF actually causes it to pitch nose down – hence the reverse box is checked. Note that the “reversal” is carried out before the cue is translated into the final 0-255 output range, so the nominal zero or midpoint position is still 127.

#### 4.1.5 Roll Cue Settings

The roll movement of the platform is used by the driver to force-cue lateral accelerations and to add an element of roll-rate cueing to the simulation. As with the fore/aft force effects of the pitch degree of freedom the lateral acceleration force effects producing platform roll can be washed out. Again, their actual duration is fairly short, the longest lasting lateral forcing event is probably taxiing ground turns and even these will reduce to zero once the turn is completed. In-flight turns do produce mild lateral acceleration effects, as can turbulent flight conditions, but they tend to oscillate about mid and/or are of short duration and don't need much wash out treatment (at least I don't think so) but you can set this as required. I have noticed however with my cockpit that aircraft roll produces a strong visual cue and my mind seems to expect to feel the aircraft rolling. To this end a washed-out roll angle cue is superimposed on the lateral force cue which produces a body roll that coincides with the screen



visual cues of roll and direction, but is not sustained long enough to develop marked false sideways

force cues. The washout does allow the platform to react to rolling which can extend over the full 360°.

When full 360° rolls are made in the flight however some adjustment has to be made to the roll cue output – because the cockpit is unlikely to be able to do full rolls. The driver is set to roll in reverse when the in-flight roll angle is greater than +/-90° - this causes the cockpit to roll back at reasonable speed to the opposite position extreme to pick-up the actual roll at the end of its travel. Without this setting the cockpit would do a “snap” position re-set to the opposite position extreme when the in-flight roll angle passes through the 180° position.

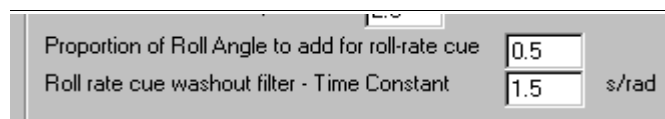
So the Roll cue settings are similar to those for the Pitch motion cue but with the addition of two parameters which control the additional cueing effect for roll-rate.

The “Lateral acceleration cap level” is set a little more stringently than the equivalent forward acceleration cap. Although an element of sideways forcing resulting from taxiing turns is effective, larger actual roll angles feel unnatural in the cockpit. So the lateral force is cued up to a modest limit as defined by the cap level – set at 2.0 m/s<sup>2</sup> (0.2g) in the above example. This only limits that element of the movement cue derived from accelerations – the roll-rate element of the cue may still cause roll movement of the cockpit over the full range available.

“Lateral acceleration washout filter – Time Constant” . If desired the lateral acceleration force element of the cue can be washed out to mid. The speed of the washout can be reduced by increasing the time constant.

Smoothing/Delay filter time constants. As before separate values can be specified for taxiing and in-flight stages. Lateral accelerations due to taxiing turns can come on and then diminish rapidly as the aircraft moves into and out of a curved trajectory. Allowing the platform roll angle to follow closely these rapid changes will induce unnatural roll rate cues that do not match the visual cues of the movement from the cockpit screen. The On Ground time constant is set to achieve a balance between fast enough response for the roll movement to apparently match the onset/offset of the turn but not too fast to roll the pilot unnaturally when the aircraft is actually level on the ground. The in-flight time constant is set for a longer response time mainly to counteract the effects of rapid oscillation of lateral acceleration forces during flight in turbulent conditions. There is plenty of scope for experimentation here, especially with the in-flight effects. There is probably an opportunity to sensitively cue the effects of uncoordinated turns.

### Roll rate effects



Two parameters control these. Firstly the proportion of the actual roll angle to pass through to the cue can be set. Inducing a feeling of rolling in the correct direction does not require that the platform roll angle matches exactly the full roll angle of

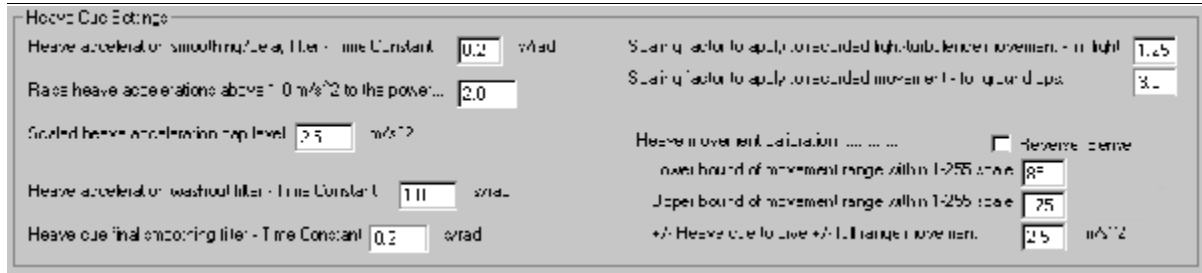
the aircraft. It seems to be sufficient only that it rolls at a noticeable rate and in the correct direction. I have used 50% of actual roll angle. There is an issue as to the rig response here in that if the proportion is too low the drive signal may not be strong enough to induce the roll movement fast enough and a delay might become noticeable. If the proportion is too large there is a risk that the cockpit roll angle will be sustained long enough to generate a false sideways force cue. The effects are related to the second parameter, the high-pass filter time constant.

A high-pass filter allows the higher frequency elements of a trace to pass untouched but filters out the lower frequency changes in the values. This has the effect of always returning the trace to zero when there are no high frequency movements taking place – ie of washing-out the movement. The longer the time constant the slower the washout speed. The setting above of 1.5 s/rad (s/1) brings the roll angle back to zero in 2 or 3 seconds or so – slow enough for the pilot not to notice it. Again you can experiment with these settings to generate different effects.

Remember to check out your settings for ground manoeuvres as well as in-flight conditions – it is irritating to come through a believable flight only for the rig to start behaving strangely as soon as you

touch-down on the runway.

#### 4.1.6 Heave Cue Settings

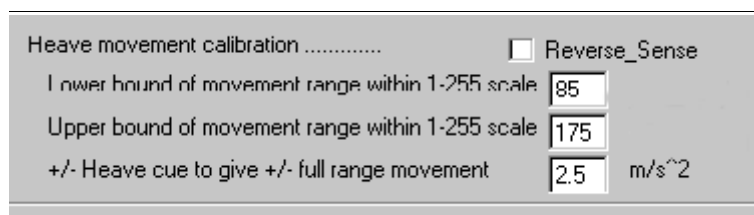


Easily the most complicated of the three motion cues to implement. Unlike the pitch and roll degrees of freedom the heave movement is not capable of force cueing sustained accelerations; the heave stroke length is simply too short. The best that can be done is to cue the onset and offset of vertical acceleration events.

A second issue is the magnitude of the vertical force effect that can be generated by a short stroke motion platform. It is only the accelerating portion of the stroke movement that generates a force effect – running the cockpit from top to bottom at constant speed does not generate any accelerating forces on the pilot other than those arising from the initial acceleration and the final deceleration of the movement. The acceleration magnitudes that can be generated before the rig reaches its mid-stroke and has to start decelerating are limited.

So, sustained accelerations are out and are replaced with what can better be described as impulse movements, however much of the other vertical movement during a flight is due to the effects of flying through inconsistent air density and light turbulence which cause the aircraft to “bob” about – these lower amplitude accelerations can be dealt with fairly well by a limited stroke simulator although even with these there will be a considerable amount of scaling down involved.

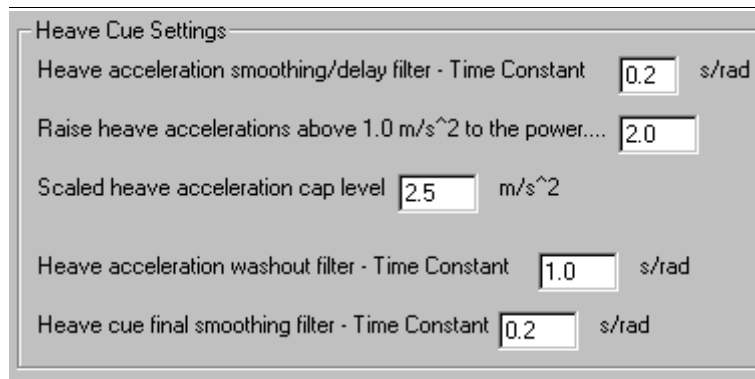
#### Heave Movement Calibration



The same in principal as described above for the Pitch movement - upper and lower limit settings for the movement range and a calibration figure which tells the driver what magnitude of cue output to fit to the working range. Whereas the units of this parameter in the pitch and roll DOF's are angular here it has linear acceleration units – m/s^2. Why this is is described below. Logically the full range movement cue should be set equal to the “acceleration cap level” described below. The “Reverse” checkbox performs the same functions as before.

See Section 4.3 for more information about these settings with a 3-point support type platform.

#### Heave Cue Settings



The aircraft's "local" vertical component of acceleration is used to derive the heave motion cue. This is also the acceleration component used to determine the more familiar "G-Force".

The actual acceleration has to go through a fair bit of processing however to yield a cue that can be used in the motion platform. The basic approach implemented in the BFF Motion Driver is to use a capped and washed-out acceleration profile as the position demand for the heave movement. Whereas the peak "felt" forces in the aircraft coincide with the peaks and troughs of the acceleration trace, in the motion cockpit the peak "felt" forces coincide with the onset of the acceleration event. Why, because what was an acceleration profile in the aircraft is being used as a displacement profile in the motion platform. This is a fairly simple and effective approach for a DIY motion driver and for DIY platform builds in which the fidelity of the heave response to the position demand trace probably doesn't warrant more sophisticated heave cue algorithms.

Firstly the acceleration is low-pass filtered to take the sharp edges off the trace. The 0.2 s/rad "time constant" shown can be altered to filter out more or less of the higher frequencies. Setting it to a low value eg 0.05 or less will leave most of the raw acceleration trace untouched.

"Raise heave accelerations above 1.0 to the power ..." This parameter allows you to exaggerate the effects of distinct acceleration events to raise them above the continuous low level flight movements. The events may typically arise from aircraft control actions to which you may wish the platform to respond clearly. It is particularly useful for simulation of lower speed light aircraft in which the normal control action accelerations may not be much bigger than the normal turbulence induced heave motions. It works by raising acceleration values above 1.0 m/s<sup>2</sup> to the power stated – ie 2 will square the acceleration value, 3 will cube it etc. Values on or below 1.0 m/s<sup>2</sup> are left unaltered – in my experience most light turbulence effects are lower than this limit.

In the case of high speed single seater jets the "normal" manoeuvring accelerations will probably be much higher anyway and don't need any special treatment to separate them from the low level stuff. In these cases the value should be set at 1.0.

"Scaled heave acceleration cap level" The acceleration trace is high-pass filtered to induce a washout, and to allow this to work it is necessary to cap the acceleration magnitude. It is important to realise that we are working right down at the bottom level of the actual heave acceleration range of the aircraft – due to the physical limitations of the motion platform. In a sustained banked turn for example the actual acceleration levels can continue to climb well beyond the limit of the rig's initial heave response. To allow the washout filter to work the accelerations it "sees" must be forced to level-off well before the actual accelerations do; the rig can then slowly return to mid position even though the actual accelerations may still be climbing or varying at a high level. The "cap level" sets this saturation limit. Note logically it should match the "+/- full range movement value" used above to calibrate the heave movement. The 2.5 m/s<sup>2</sup> (0.25g) setting illustrated is one I've found works for my rig over a range of flying conditions.

**Use the Capture button to capture the actual maximum and minimum accelerations experienced in the flight during important flight events (see Appendix 5). For the heave DOF you may be most interested in accelerations associated with take-off, touch-down, moderate in-flight pitch adjustments and moderate turbulence effects. You can then set the acceleration cap level appropriately to "feel" these flight key events.**

Heave acceleration washout filter - Time Constant	<input type="text" value="1.0"/>	s/rad
Heave cue final smoothing filter - Time Constant	<input type="text" value="0.2"/>	s/rad

The “Heave acceleration washout filter Time Constant” controls the main heave cue washout. In my experience a fairly fast washout can be tolerated because only the higher rate heave movements of the rig are noticeable

by the pilot. Increasing the time constant of this high-pass filter will slow the speed of washout.

Finally another low-pass filter is run on the cue to remove any jagged edges. Reducing the “Heave cue final smoothing filter” time constant will reduce the effect of this filter.

Scaling factor to apply to recorded light-turbulence movement - in flight	<input type="text" value="1.25"/>
Scaling factor to apply to recorded movement - for ground ops.	<input type="text" value="3.0"/>

There are two remaining settings which control two special heave effects.

The Motion Driver provides the means for you to chose to superimpose a recorded “light turbulence” heave movement onto the live movement in circumstances when the flight simulator program is reporting unrealistic flat-pan flying conditions. My FS9 seems to do this quite a lot even under “real” weather flying conditions. The scaling factor to be applied to this recorded data can be set here so that you can adjust the resulting amplitude of the actual movement of the platform.

A similar factor can be applied to the generated runway induced movement which is superimposed on the actual heave movement during runway operations. See Appendix 2 for the configuration settings needed in the .bff config file to define the runway effects.

Either or both of these “manufactured” effects can be disabled by setting their scaling factor to zero.

It is quite likely that you may want to disable these for X-Plane use. Why? Because X-Plane will usually report much more flight movement than FS9/FSX, including heave movement when the aircraft is on the ground, and there may be much less need to use “manufactured” movement. This does depend on your realism settings however.

## 4.2 Save Settings & Exit

Changes made to the Motion Driver settings can be saved to the original or a new configuration file using the “Save\_Software\_Settings” button. Note that no changes made to the data fields above the save and exit buttons are saved until the Save button is clicked. If you click Exit without saving first, your changes will be lost.

The configuration file also holds the settings made in the area below the save and exit buttons. These apply specifically to the older BFF 4-Chip Signal Processor Unit (SPU), particularly to the 18X PICAXE microcontroller chips which control the positions of each of the 3 DOF's on the platform. These settings will be saved along with the driver software settings when the save button is clicked. Additionally, the settings for an individual chip will be saved to the current configuration file when the relevant “Set\_18X” button is used to set the SPU – more below.

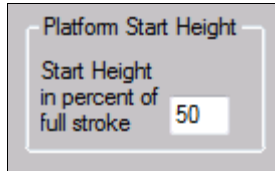
The file save arrangements allow you to keep more than one configuration file, each with different settings. These can be loaded into the Motion Driver on startup allowing you to change the response of the motion platform to suit different aircraft or flying styles. I should say however that I find a single setup seems to do me for most occasions – but then again I don't do much combat flying for example and am happier flying about looking at the scenery!



### 4.3 Setup-3pt.exe For 3 Point Support Type Platforms

To simplify the setup for motion platforms that use a triangle of driving actuators I've introduced a separate set up application. This uses slightly different data items and should make the settings for these rigs easier to understand and implement and should be used with the MSFS driver versions v1.6 and later and X-Plane driver versions 1.3 and later.

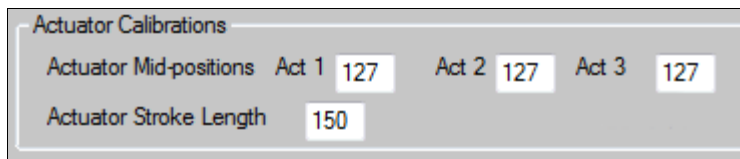
Most of the cue generation settings are the same as those used in the uncoupled rig type setup and you should refer to section 4.1 above for these. The following items are particular to the 3-pt setup.



#### Platform Height

The position (height) to which you want the platform to be sent when the driver first starts-up. This will also be the position that the PID Servo Controller will send the platform when it is instructed to "Hold" position. The height is given as a percentage of the full stroke length of the actuator as defined below.

#### Actuator Calibrations



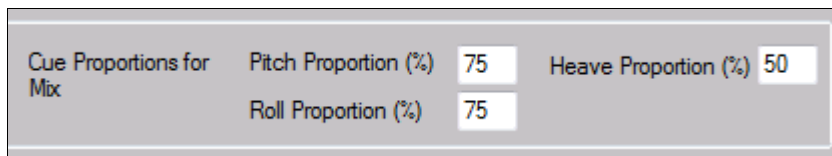
#### Actuator Mid-positions

The feedback pot centering on each actuator may not be the same and this setting allows you to define for each actuator where its

mid-position is on a scale of 0-255.

Actuator Stroke Length – It is assumed that each actuator is the same and has the same stroke length. This is specified with reference to the same 0-255 scale as the mid-position settings. Example, if your actuator strokes from, say 50 to 200 on the 0-255 scale then its Stroke Length is  $200 - 50 = 150$  and the value is assumed for all the actuators.

The stroke length and mid-positions are used to calculate the upper and lower movement limits for each actuator and all the motion cues are fitted within these working bands.



#### Cue Proportions for Mixing

The movement of each actuator is made up from contributions from the cues for each DOF. These

settings allow you to specify how much of the actuator full stroke length can be used to generate each of the DOF cues. Logically this might be set to 33-33-33% to indicate each cue can use 1/3<sup>rd</sup> of the available stroke – with the reasoning that at no point will the actuators max-out. However because the 3 DOFs rarely all go to their extremes at the same time this would result in a substantial underutilisation of the available actuator stroke for most of the operating time of the platform. It is better therefore to "over-specify" this to some degree to ensure that most of the stroke is used most of the time to generate good pitch, roll and heave movements.

In the above image the splits are 75-75-50%. You will need to trial this to determine a mix that works for you.

Note that the cues will always be capped at the limits of movement of the actuators should the individual contributions add to more than 100% of the available stroke length.

Note also that Act 1 is the single front actuator and Acts 2 & 3 are the paired rear actuators in the 3-point set up. If you wish to use a reverse arrangement with one actuator at the rear and two at the front simply reverse the Pitch and Roll cue senses in the setup program.

## 4.4 Articulated Projector Drive Setup

The detailed settings for the visual driver are held in the vis.cfg file specified by the vis\_frame parameter in the current .bff config file. The vis.cfg file contains detailed parameters which control the cue algorithms and parameters to set the data output mode and COM port settings.

The contents of the vis.cfg file is typically (values may be different)-

```
-----  
----Lateral velocity settings - noise filter TC, washout filter TC, working/fit range, final scaling factor, reverse sense (1 or -1)  
0.2  
0.2  
0.8  
0.1  
-1  
----Yaw  
0.20  
0.0  
1.0  
0.6  
1  
----Heave  
0.15  
0.2  
5.0  
0.6  
-1  
----Bank  
0.3  
0.0  
0.1  
0.45  
1  
----Fwd (not presently used in the algorithms)  
0.15  
0.25  
5.0  
1.0  
1  
----Pitch  
0.15  
0.15  
0.2  
2.0  
-1  
----Servo Mid positions and amplitudes (movement range = 2 x amp)  
1500  
100
```

```

1500
100
1500
100
----COM Port and Baud
COM5
28800
----Rig type (1 = independent DOFs, 2 = partial Stuart drive), output mode (Pololu, Picaxe or NONE) and tilt horz adjustment
factor
2
Pololu
-0.15
----GUI Slider Settings - position adjustment and simple scaling – written by software
3
-1
-1
99
113
99
----End of File

```

-----

The upper section contains the settings for the cue calculations, below are the servo settings, COM and output mode settings. The last section is for the GUI slider positions stored by the driver when last closed.

The most likely settings you will want to change are the COM port and servo midpositions and working ranges. The COM port specified should be different from that used for the motion driver main platform drive. 28800 baud is the value I've used with the Pololu servo controllers, this can be altered to suit other hardware.

The servo data is in the order servo 1, 2 then 3, default mid is 1500 for each and the working amplitude is 100 to suit the projector articulation drive mechanism described on the Built For Fun website. If you are using different servos or a different setup then you may need to adjust these values.

The rig type settings allow you to specify either an output mode to suit independent servo drives for each DOF of the articulation mechanism (=1) or for a coupled, partial Stuart type mechanism (=2) as I've used on the web site. Two data output modes are available "Pololu" to write the output data in "Pololu Mode" to drive their servo controller or "Picaxe" to write the servo positions in a format readable by PICAXE (and other) microcontrollers.

The "Picaxe" output mode is:

"JK" byte1 byte2 byte3 byte4 byte5 byte6

where JK is a two ascii character start-of-data indicator.

byte1 & byte 2 are the LSB and MSB of a word variable holding the servo 1 position

byte3 & byte 4 are the LSB and MSB of a word variable holding the servo 2 position

byte5 & byte 6 are the LSB and MSB of a word variable holding the servo 3 position

The "tilt horizontal adjustment factor" is for use with partial Stuart type mechanisms in which the projector lens lies below the natural center of rotation of the projector support plate. This factor can be set to compensate for the horizontal movement that occurs at the lens when the platform tilts by

moving the platform horizontally in the opposite direction as it tilts.

## CUE SETTINGS

Unless you specifically wish to experiment with the projector motion I suggest that the cue calculation parameters are left at their default values. Simple scaling can be applied to the output motion using the appropriate sliders on the Motion Driver window.

The motion cues are generated from the actual aircraft in-flight velocities, projector horizontal movement is determined from aircraft lateral and yaw velocities, vertical movement from aircraft heave and pitch velocities and tilt movement from aircraft roll velocity.

The overall approach taken to develop the motion cue from each velocity component is:

- Read raw velocity from sim
- Apply noise/smoothing filter (not much smoothing is required but gives opportunity to add a delay)
- Apply washout filter (to quickly return cue to zero after initially following the velocity trace - an important setting as it affects the sharpness of the cue recovery to zero for linear motions)
- Apply cap (can be set above or below actual velocity levels)
- Invert signal if specified (to get sense right and fit to rig mechanics)
- Calculate corresponding servo movement (using the cap value and specified servo movement amplitude)
- If required mix with the second velocity term and determine final servo signal

The parameters for each velocity term are therefore:

- noise filter TC
- washout filter TC
- working/fit range
- final scaling factor
- reverse sense (1 or -1)

A worked example is shown in Appendix 6. The "itrace" parameter in the .bff configuration file can be set to export the data used in the calculations for a particular velocity component, this data can then be plotted and used to determine how the processing might be adjusted to obtain the final motion profile. Details of the itrace settings are also shown in Appendix 6.

This approach provides great flexibility in the cue processing but is complicated and needs to be undertaken thoughtfully.

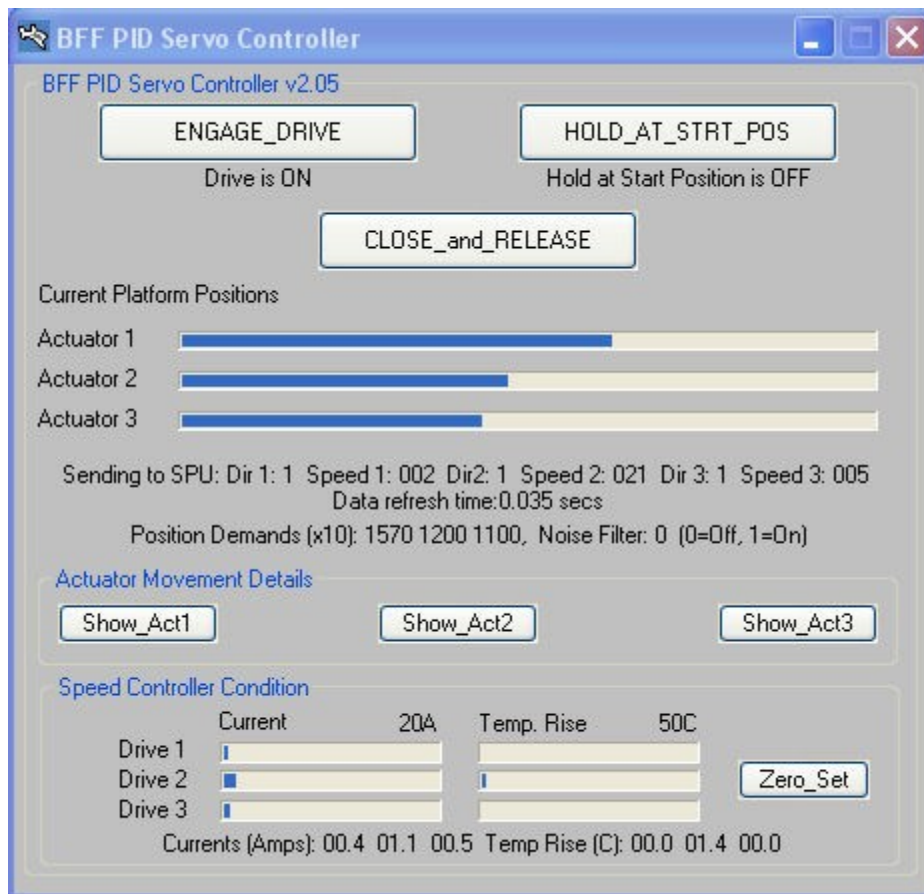
## 5. BFF PID Servo Controller

### 5.1 V2 Servo Controller Operation

A substantial simplification to the external signal processing hardware was made possible by bringing the speed demand control calculations off the external hardware and on to the PC. These calculations are now done by the BFF PID Servo Controller application which runs alongside the BFF Motion Driver. It uses industry standard Proportional-Integral-Derivative control algorithms to determine platform speed which is then sent out to the new single-chip SPU to be passed on to the motor speed controllers which drive the platform motors. To do this it reads position demand data provided by the Motion Driver from a shared memory area and real-time position feedback data sent by the single-chip SPU through a COM port.

The PID Servo Controller is started from the v1.5 and later MSFS Motion Driver using the “Start\_PID\_Controller” button.

V2.05 of the Servo Controller adds a speed controller condition monitoring facility. This allows the real time current draw and temperature rise of the motion platform MD03 speed controllers to be monitored.



The Servo Driver should be started once a flight is started and the Motion Driver has been able to write the initial data to the Shared Memory area. Once started the servo controller will check that it is receiving valid position feedback data from the SPU (make sure the SPU is connected and powered-up) and then will commence exporting speed demand data to the SPU. It starts in a “Hold\_At\_Start\_Position” mode until the user clicks the “Engage\_Drive” button at which point it will send the live speed demand to the platform. The “Close\_and\_Release” button closes the Servo Controller and ceases sending the speed demand – this will be detected by the single-chip SPU

which in turn shuts down the drive to the platform motors.

For smoothest “start” and “stop” control of the platform drive it is best to set the PID Servo Controller to ENGAGE\_DRIVE before the BFF Motion Driver is set to Drive – ie whilst the motion driver is demanding parked position. Then use the Motion Driver Hold\_Start\_Position and Drive buttons to smoothly park and un-park the platform.

Both the PID Servo Controller and the single-chip SPU have been designed to work in a fully integrated way to help protect the cockpit from loss of control. These design features are:

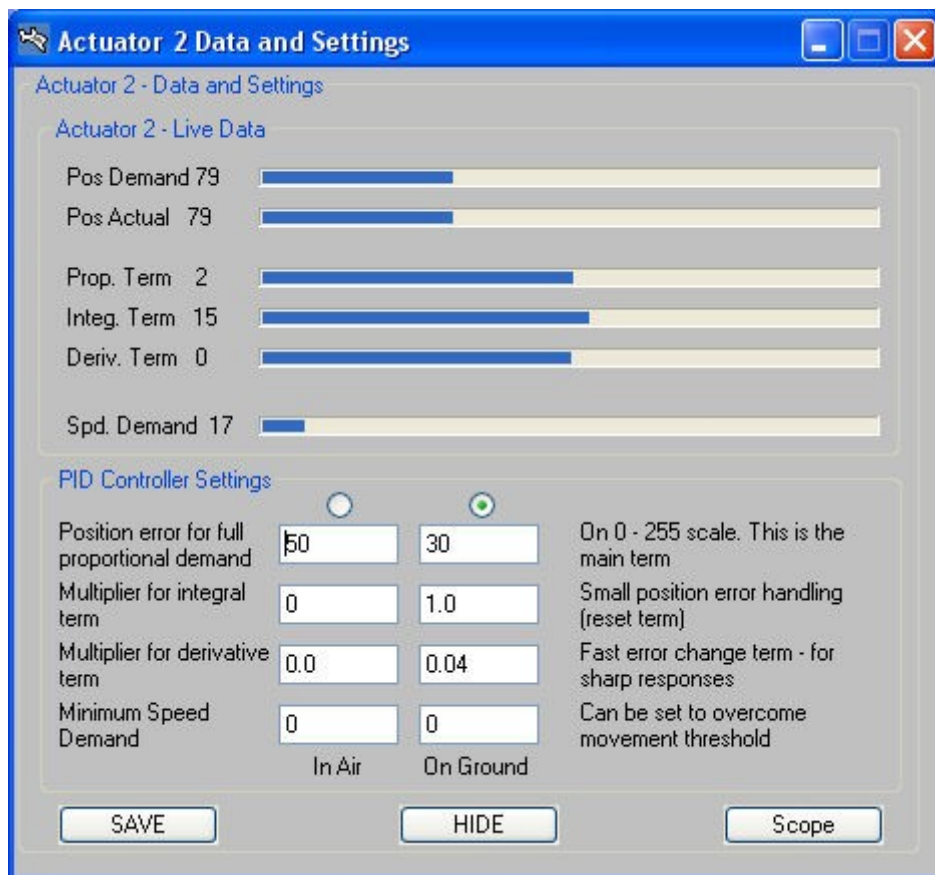
- The PID Servo Controller runs as a separate process from the BFF Motion Driver and will continue to issue speed commands to the SPU if the flight sim or Motion Driver hang.
- The Servo Controller monitors the comms with the SPU and issues a warning message if there is a loss of position feedback data from the SPU.
- The SPU monitors the Comms with the Servo Controller and will close down the platform drive if the flow of data from the Servo Controller is interrupted.

## 5.2 Visual Display

The PID Servo Control provides a range of textual and graphical information. The main window (see above) shows three progress bars which reflect the current position feedback from the platform position potentiometers. This is on the full possible 0-5V scale of the output from the feedback pots – your rig may only operate within a restricted band within this possible range. Textual data is also provided on the current speed demand and directions, refresh time, noise filter status etc.

Three buttons at the bottom of the window can be used to display secondary windows with more detail of the output and settings for each of the three output drives - “Show\_Act1” , “Show\_Act2” and “Show\_Act3”.

Using Actuator 2 as the illustration .....



The upper “Live Data” section shows the current actual and demanded position data, the PID terms which make up the speed demand and the final speed demand output.

The lower “PID Controller Settings” area shows the current settings for the drive and includes a **SAVE** button to save and activate any changes you make to the settings. **THIS MUST BE USED WITH CAUTION!** The behaviour of the speed demand output (and hence of your platform motion) is very sensitive to changes in these settings. When you make changes make sure they are small and incremental and have an emergency stop button near to hand to kill the drive to the platform should you changes have an unexpected result.

Two sets of PID settings are used for each Actuator, one set for on-ground operations and one for in-air operations. The radio buttons in the PID Controller Settings area indicate which set is active at any time.

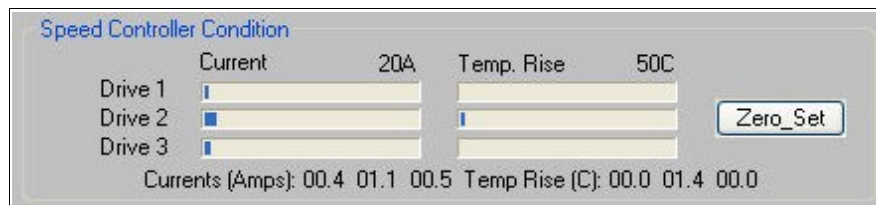
The speed term bars use a scale of 0 to 255, the speed related outputs use a scale of -255 to 255 with 0 as the mid position. The final speed output bar uses a 0-255 scale with 0 as zero speed and 255 as full speed output.

The **Scope** button activates the built-in Virtual Oscilloscope and traces the position demand, actual position and speed demand for the current actuator. This can be used to visually inspect the effects of changes made to the PID settings.

The virtual oscilloscope is implemented using Michael Bernstein's excellent Software Oscilloscope DLL – see his web site at <http://www.oscilloscope-lib.com/> for more details. The traces can be manipulated on the scope using the buttons provided as is normal for any oscilloscope.

#### Speed Controller Condition Monitoring -

The lower section of the Servo Controller main window shows real-time reporting of your MD03 speed controller current draw and temperature rise. This information will allow you to keep a check on the speed controller workload. The rated current of the MD03 controller is 20 Amps. The temperature rise progress bars have a full scale reading of 50 Centigrade.



Speed controller monitoring is only enabled if you are using the Enhanced Speed SPU and the correct PICAXE chip flash program, and are running V2.05 or later of the Servo Controller.

### 5.3 Controller Settings

There is a good article about PID control [here](#) should you want to read up on it in more detail. In short the final speed demand for each drive is made up from three elements -

- the Proportional term is the main element and sets the speed proportional to the size of the position error (the difference between the desired and actual actuator positions).
- the Integral term adjusts the speed based on the accumulating position error and is useful for ensuring the pitch and roll angles return fully to their upright positions.
- the Derivative term adjusts speed on the basis of how fast the error is changing and can act as a brake on the system to reduce overshoot or to induce a sharper response in reaction to sharp acceleration changes. This term is the trickiest to work with as it is most affected by noise in the feedback signals which can make the response of the platform very jumpy.

A fourth fixed element is added here – the “Minimum Speed Demand” which allows a small fixed speed to be added to overcome any friction related movement threshold that may exist in the platform

mechanics.

For motion platforms which have independent (uncoupled) DOF's it is possible to have different settings for the three outputs which reflect the requirements of the different motions. However for use with coupled systems, eg 3-pint support type rigs it may be necessary to set each drive with the same settings.

In V2 of the PID Servo Controller it is possible to define two different sets of PID parameters for each channel for on-ground and in-air operations. The servo software switches between the sets automatically at the point of takeoff or on touchdown. This allows a sharper servo response to be set for on-ground operations where the aircraft accelerations (eg due to touch down impacts, sharp braking etc) are likely to be sharper. The in-air response can be left fairly soft to give a smoother more cushioned feel to the motion.

### ***The Proportional Term***

This is the main one. This is the position error within the 0-255 movement range that gives full speed demand. If the error is below this value the speed demand is scaled down if it is above this value the speed demand holds at maximum. The smaller the value is the faster the movement will respond to changes in the demanded position. However the smaller values will also mean it has more trouble slowing to a stop once the rig has moved to the demanded position. So small value for fast response with likely position overshoot, and large value for slow response with no overshoot.

Values between 20 and 100 as a guide only.

### ***The Integral Term***

Also known as the re-set term, this calculates a contribution to the speed demand based on how long a position error is sustained. It is mainly of use for making sure that the position comes back to that demanded when the proportional term has lost its effect on small position changes.

Guide values would be from 0 to 4 or 5.

### ***The Derivative Term***

The trickiest term to get right. This speed demand component is calculated from the rate at which the error is changing. If the rate is high it means that the actual and demanded positions are either moving apart quickly or coming together quickly. If they are moving apart quickly the rig isn't moving fast enough and the term gives a boost to the speed, if they are coming together quickly the rig needs to slow down faster and the term acts as a brake on the motion. The practical effect can be as a brake and to help to reduce the extent of any overshoot there may be if the proportional term is set to respond strongly to position errors. Its use is therefore likely to be able to allow faster response. Proportional terms to be used by reducing the extent of the associated overshoot.

It can also be used in conjunction with softer proportional settings to add a sharper reaction to short term acceleration spikes such as touch-down bumps which a soft proportional response would miss.

However..... the derivative term can be badly affected by noise on the position feedback signals. And this can lead to very twitchy response when the Derivative term Multiplier is set too high. For it to be effective the feedback signals need to be clean and noise filtering might help.

To this end 4.7µF capacitors or other hardware filtering might be fitted to the feedback inputs to the SPU PICAXE chip. In later versions of the software an internal noise filter is available which can be switched **ON** or **OFF** by setting line 25 in the PID2.cfg file to **1** or **0** respectively, and by setting line 26 to the value of the filter's time constant. This should be set to a value of **0.1 or lower** – higher values can introduce too much lag into the servo loop and it can become unstable. I prefer not to use the internal filter but to use the best possible position feedback devices to ensure clean position signals.

In V2 of the PID Servo Controller support has been added for the use of PWM output position feedback encoders with the BFF single chip SPU hardware. These produce a much cleaner position



feedback signal and allow greater use to be made of the Derivative term in the PID set up. See the web site for more information as the Signal Process Unit (SPU) must run at 16MHz to achieve the required signal resolution. See also Section 5.5 below.

Guide values for the Derivative term multiplier – from 0.01 to 0.1

### ***Minimum Speed Demand***

This is an additional term that I have added which might be of use. It allows a small fixed speed demand to be added in to the total to help overcome any friction based movement threshold in the system. Setting this just below the speed demand that actually initiates movement may help improve the speed of response.

Guide values, 0 to 20 in the 0-255 full speed scale.

### ***Setup Procedures***

Unfortunately there is no set of settings that will be correct for every motion platform. Many of the above variables need to be set experimentally. Start by setting the Integral and Derivative terms to zero and work with the Proportional term first. This should be set at the higher end of the guide scale and then reduced gradually until the rig motion in response to step position inputs starts to overshoot or oscillate.

Note that V2 of the motion driver has a Manual Override feature in which the position demand outputs can be manipulated manually using sliders in the driver window.

If you wish to make use of the Minimum Speed Demand setting, set this next. With the Servo Controller holding mid position simply increase the setting from zero until the movement starts to oscillate slightly – do it in small steps. A setting which holds the motion steady but just “ready” to move is what you are looking for. If this is set too high you will feel a “clunk” each time the platform drive changes direction.

The Integral term can also be set with the controller holding position. Increase the values in small steps until you start to see small oscillations occurring. Watch the progress bars – you will see the integral term building until a small movement adjustment takes place and the term reduces again.

The Derivative term will be the hardest to get right. If you feel it is not needed then don't use it – I have it set to zero on the Pitch and Roll DOF's on my rig – the speed of response in these DOF's doesn't need to be high. Again increase the setting in small steps from zero and watch the response of the rig to step position demands before going on to see if you can then further reduce the Proportional term error setting.

**ALWAYS TAKE CARE – ADJUSTING THE CONTROL SETTINGS CAN CAUSE RAPID MOVEMENT OF THE PLATFORM. ENSURE SAFETY PRECAUTIONS ARE TAKEN AND ENSURE THE PLATFORM TRAVEL LIMIT SWITCHES ARE FITTED AND FULLY OPERATIONAL.**

## **5.4 Servo Controller Data Output for Analysis and Inspection**

The best way to understand the response of your platform is to plot the motion data. If the line 27 of the PID.cfg file is changed from 0 to 1, 2 or 3 then the Servo Controller will store the control data for the specified drive (1, 2 or 3) and dump it to the file PID\_Output.txt when the Controller is closed. The file will be in the same directory as the controller program.

**DO NOT LEAVE THIS SETTING ON DURING NORMAL USE!**

The data is written as a space delimited text file that can be opened with a spreadsheet program and charted as required.

The data output includes -

- Position feedback
- Demanded position

- Position error
- Proportional, Integral and Derivative speed demand terms
- Total speed demand
- Speed Controller Current (see 5.5 below)
- Speed Controller Temperature Rise (see 5.5 below)

and is reported for each processing loop, so the file will build in size quickly.

Plotting position feedback and demanded position is a good place to start as it will show quickly how well your platform is following the demanded motion output by the Motion Driver.

## 5.5 PICAXE Chip Flash Programs for DIY servo hardware

For the PID Servo Controller to work the PICAXE 28X1 chip on the single chip Signal Processor Unit must be flashed with the correct .bas program. At the time of writing 28X1 chips with A.3 firmware are being shipped by PICAXE and there are currently a number of alternative Signal Processor Unit (SPU) design and flash programs that can be used with the A.3 firmware chip -

- Basic SPU with 8MHz internal resonator
- Basic SPU with 16MHz external resonator
- Basic SPU with 16MHz external resonator with US Digital MA3 12bit PWM output position encoder.
- Enhanced Comms Speed SPU (115200 Baud) with MA3 Encoder Feedback – Requires V2.04 or later of the PID Servo Controller Software.
- Enhanced Comms Speed SPU with MA3 Encoder Feedback + MD03 Speed Controller Condition Monitoring – Requires V2.05 or later of the PID Servo Controller Software.

### IMPORTANT

V2.04 and later of the Servo Controller can operate at different baud rates. The 40SPU-1 signal processor card and the DIY Enhanced Speed SPU design both require that the Servo Controller is set to run at 115200 Baud. This is done by setting line 28 of the PID2.cfg file to “115200” (without the inverted commas). If your PID2.cfg file does not have a line 28 then add one.

## 5.6 40SPU-1 Signal Processor Card

As of June 09 a ready made signal processor card is available which operates with **v2.5 or later** of the BFF PID Servo Controller as part of a DIY servo drive control system.

For details of the 40SPU-1 hardware and its required PID2.cfg settings see the 40SPI-1 data sheet available on the web site.

In summary the 40SPU-1 requires an 8 bit binary entry in line 29 of the PID2.cfg file to control active channels, read/write enable/disable, encoder direction reverse etc and a 115200 baud rate. Please read the separate 40SPU-1 data sheet.

Check the web site for the latest information on available chip flash programs and SPU designs.

Ian Hopper

BFF Design Ltd.

Last Updated July 2009

## **LICENSE**

This SOFTWARE PRODUCT is licensed as separate components.

Its component parts may be separated. The script component (BFF Motion Driver) is licensed under the terms of the EULA below. The script's interpreter component is licensed under the terms of the GNU General Public License (<http://www.gnu.org/copyleft/gpl.html>).

Credit given to: AutoHotKey ©2003-2007 Chris Mallett, portions ©Autolt

Team (<http://www.autohotkey.com/> and <http://www.autoitscript.com/>). Credit also to

Laszlo Hars <http://www.Hars.US> and Icarus on the AHK forum for the software

registration components – see topic <http://www.autohotkey.com/forum/viewtopic.php?t=5763>.

### EULA

BFF Motion Driver has been provided WITHOUT ANY WARRANTY without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. You assume all responsibility for the consequences of your use of BFF Motion Driver.

BFF Motion Driver is licensed for your personal NON-COMMERCIAL use only. I am happy to negotiate a reasonable charge with you if you wish to use BFF Motion Driver for the purposes of commercial development including but not limited to research & development, design feasibility study, prototype development, production and/or inclusion within or packaging with domestic or commercial products.

You may not further distribute BFF Motion Driver. You may not reverse engineer or modify BFF Motion Driver.

Removal or modification of the License file or of the BFF\_Registration.ini file will disable the software.

BFF Motion Driver is copyright ©, 2007, 2008, 2009 BFF Design Ltd. All Rights Reserved.

## Appendix 1 – Optional Data Output for Setup/Debug

The optional parameter “itrace” in the .bff config file enables calculation data output to the file BFF-output.txt in the working directory. If this parameter is non-zero the program will store internal calculation data during the flight and dump it to file when the driver is closed. NOTE the program will append the new data to any existing BFF-output.txt file – if you want the file to hold only the current set delete or rename it first.

The “itrace” settings are -

- = 0 , for no data output – use for normal operation.
- = 1 , for output of base acceleration data and final cues
- = 2 , for output of heave calculation data
- = 3 , for output of pitch calculation data
- = 4, for output of roll calculation data
- =13, for GAL output mode only

The data is written to the text file in space delimited format, one row per calculation time step. It can be opened with a spreadsheet program and charted for analysis purposes. The non-zero settings should not be used for normal flying. The number of internal variables for the process will increase to a level that affects the program speed and the simulator frame rates. Use only for short test flights.

Additional values relating to the Articulated Projector Drive are described in Appendix 6.

### Output Variables

Data item by spreadsheet column letter, for itrace set.....

itrace=1

- A) Pitch cue,
- B) Roll cue,
- C) Heave cue,
- D) Fore/aft component of accel acting on pilot,
- E) Lateral component of accel acting on pilot,
- F) Vertical component of accel acting on pilot (minus 1g)

*The acceleration components include the contribution made by any component of gravitational acceleration acting in the referenced direction. I.e the accelerations account for the force effects of the pitch and roll angle of the aircraft as they are felt by the pilot. Accelerations in m/s<sup>2</sup>.*

itrace=2

- A) Vertical component of accel acting on pilot (minus 1g)
- B) Same accel after 1<sup>st</sup> low pass filter,
- C) ..... with exponential scaling applied,
- D) .... with cap level applied,
- E) ..... capped accel after high-pass wash-out filter,
- F) .... with any recorded heave accels added in,
- G) Final processed accel used for heave position cue – after final smoothing.

*Charting the data columns together will show the effects of each stage of the processing. Note that if an exponential scaling factor is used (see setup program) the data in the 3<sup>rd</sup> column may be of a much higher magnitude than the others. All values in m/s<sup>2</sup>*

itrace= 3

- A) Fore/aft component of accel acting on pilot (m/s<sup>2</sup>),
- B) .... after low-pass smoothing filter (m/s<sup>2</sup>)
- C) .... after cap level and washout applied and expressed in g's
- D) .... pitch angle required to simulate true force effect of the acceleration (radians)
- E) Additional washed-out pitch angle contribution (radians)
- F) Final combined pitch angle position used for the pitch cue (radians).

*Note that C & D will be very similar for the smaller accelerations ( $D = \arcsin C$ , and C is in radians) – they will diverge as the angles increase.*

itrace= 4

- A) Lateral component of accel acting on pilot (m/s<sup>2</sup>),
- B) .... accel after any scaling has been applied (see note below).
- C) .... after low-pass smoothing filter (m/s<sup>2</sup>)
- D) .... after cap level and washout applied and expressed in g's
- E) .... roll angle required to simulate true force effect of the acceleration (radians)
- F) Additional washed-out roll angle contribution (radians)
- G) Final combined roll angle position used for the roll cue (radians).

*Note that C & D will be very similar for the smaller accelerations ( $D = \arcsin C$ , and C is in radians) – they will diverge as the angles increase. B will normally be the same as A – there are two scaling factors in the .bff config file that can be amended from their default values of 1 to apply a scaling factor to the lateral acceleration when the aircraft is on the ground or in the air. These do not appear in the Driver Setup program but can be adjusted if required directly in the configuration file.*

itrace=13

- A) Channel 1 Demanded position
- B) Channel 2
- C) Channel 3
- D) Channel 1 Actual position
- E) Channel 2
- F) Channel 3

*This setting is active for the Mode=GAL output for Galil Optima series controllers only.*

## Appendix 2 – Manually-set Config File Parameters

### Parameters

Appendix 2 – Manually-set Config File Parameters.....	38
A2.1 Lateral Acceleration Scaling.....	38
A2.2 Vertical Acceleration Scaling.....	38
A2.3 Data Trace Enable.....	39
A2.4 Motion Platform Type.....	39
A2.5 Disable Booleans Output.....	39
A2.6 LAN Settings – Port and IP Address.....	40
A2.7 Software Drive Direction Reverse.....	40
A2.8 Runway Operations Heave Effects.....	41
A2.9 Articulated Projector Driver Configuration File.....	41

Most of the parameters in the .bff configuration file can be set through the driver setup program. There are a handful of parameters in the config file however that need to be set manually if you want to change them.

They are -

#### A2.1 Lateral Acceleration Scaling

---

```
; Scaling factor to be applied to raw lateral accelerations
; When aircraft is on ground
LaScaleg=1.0
;When aircraft is in air
LaScalea=1.0
```

---

These provide scaling factors that can be applied to the raw lateral accelerations before any of the cue calculations are done. Separate values are possible for on-ground and in-flight conditions. Default value is 1 for both – ie no scaling applied. To implement lateral acceleration true-direction impulse cues rather than gravity alignment force cues then give both parameters a -ve value (to reverse the cue direction) and set a fast acceleration cue washout.

#### A2.2 Vertical Acceleration Scaling

---

```
; Scaling factor to be applied to raw vertical accelerations before any calcs are done - these do not
appear in the setup program but are active
; When aircraft is on ground
VaScaleg=1.0
;When aircraft is in air
VaScalea=1.0
```

---

As above but this time for vertical or heave accelerations. This allows separate scaling of unrealistically high touch-down accelerations for example.

### **A2.3 Data Trace Enable**

---

;Trace Output, =0 no output, =1 raw accels & final cues, =2 heave cue calc outputs, =3 pitch cue calc outputs, =4 roll cue calc outputs, =5 to 12 projector driver data tracing

;Use for setup purposes only - do not leave on - the processing will eventually grind to a halt as it tries to store all the data.

itrace=0

---

See Appendix 1 for details

### **A2.4 Motion Platform Type**

---

;Platform Type, =0 for independent (uncoupled) DOFs, =1 for 3 point support & 3 DOF's, =2 for ?

Rig\_Type=0

---

Parameter to set the motion platform type. This affects the motion cue output. For =0 the cues are output directly as uncoupled position demand signals which should be used to drive actuators dedicated to their single DOF – ie there is no coupling between the outputs.

For =1 the cues are “mixed” before output and are used to drive motion platforms with 3 point support systems. In this mode the cues are in the order Front, Rear-Right, Rear-Left rather than the straight Pitch, Roll & Heave of the uncoupled setting. Pitch cues are delivered by raising the front actuator and lowering the rear actuators by the same amount, Roll cues are delivered by leaving the front actuator unaltered, raising the Rear-Right and lowering the Rear-Left actuators and Heave cues are delivered by moving all actuators up or down by the same amount. The cues are combined in the driver and then split to provide position demands for each of the three actuators.

### **A2.5 Disable Booleans Output**

---

;Flag to indicate whether Booleans are needed, =0 No, =1 Yes, setting to 0 will reduce the processing load slightly

Use\_Bool=1

---

The HEX2 data output mode includes a number of pieces of information additional to the main motion cues. These include a single byte with status indicators for a number of aircraft systems (gear position, flap position, overspeed etc) and a byte for engine RPM. Obtaining these places a slightly higher processing load on the driver and if they are not required setting Use\_Bool=0 will disable them.

Does not apply to BIN output mode (in BIN these items are automatically disabled).

See section 3.1 for more details.

## A2.6 LAN Settings – Port and IP Address

---

;For X-Plane and LAN MSFS versions, If X-Plane and driver are on same machine set to - local machine - ie 127.0.0.1

;If the Driver is running on a separate LAN machine set this to the LAN IP address of the Driver's machine eg "192.168.1.x"

IPAddress=127.0.0.1

---

This and the next parameter are for X-Plane and LAN networked versions of the motion driver. This uses UDP data transfer to obtain the required flight data from the flight sim. IPAddress is the IP address of the machine on which the Motion Driver is running. If it is running on the same machine as the flight sim then this should be set to 127.0.0.1, If it is running over a LAN it should be set to the LAN IP Address of the machine on which the driver is running, eg 192.168.0.2.

For the X-Plane version this should also be the address set in the X-Plane "Inet 2" set up page for the data receiving machine.

For the MSFS version the IP address should also be set in the Bridge.cfg file on the sim PC.

---

;For X-Plane and LAN MSFS version version, Receiving Port on Driver's machine.

PortS=48001

;If using a networked machine you may have to open port PortS on your firewall to enable it to receive data.

---

The receiving port number on the machine on which the Driver is running. Note than running over a network may require your firewall setting to be adjusted to allow coms between the machines concerned and to open port PortS on the receiving machine.

---

## A2.7 Software Drive Direction Reverse

---

;For 3-pt rigs with non-reversible feedback encoders - software drive direction reverse on individual channels. No effect when Rig\_Type=0

Act1\_Rev=1

Act2\_Rev=0

Act3\_Rev=0

---

For use with 3-point support type platforms in which non-polarity-reversible feedback encoders might be used for position feedback. With these encoders it is not possible to electrically reverse the positive drive direction such as is often needed if "handed" actuators are used – ie where a positive drive signal causes one actuator to move in the opposite direction from another. These parameters allow the driver to reverse the position direction for an actuator AFTER the coupled output has been derived from the combined pitch, roll and heave demands.

No effect on Rig\_Type=0 uncoupled platforms – use the individual DOF sense reversal parameters



for these.

NOTE if you are using the 40SPU-1 signal processor unit card then this encoder direction reverse can be made more easily in the PID2.cfg file – see the 40SPU-1 data sheet for full details.

## **A2.8 Runway Operations Heave Effects**

---

;Runway heave effects

;Slow acceleration sine wave amplitude at takeoff speed  $m/s^2$

RW\_Amp\_3=3.0

;Slow sine wave frequency at takeoff speed - in cycles per second (Hz)

RW\_CPS\_3=0.9

;Fast acceleration sine wave max amplitude  $m/s^2$ , set a very small value to effectively disable if platform does not have good high freq response

RW\_Amp\_1=0.5

;Wheel Diameter for Fast Wave Frequency - m, set at unrealistically large dia to simulate centerline bumps

RW\_Dia=0.2

;Reference take-off speed for calcs - knots

RW\_Tos=100.0

---

Set to generate additional runway heave motion. The slow wave motion increases in frequency and amplitude to the specified values at speed RW\_Tos at which it is held constant until takeoff.

The fast wave is at the wheel rotational speed and its frequency is calculated from the specified wheel diameter. This can also be used to generate slightly lower frequency “bumps” if the wheel diameter is set large (and hence rotates slower).

The fast wave effects will only be effective if your platform has good higher frequency response – I tend to use the slow wave but disable the fast wheel rotation effects.

NOTE the runway heave scaling parameter Turb\_Scaleg in the main heave section of the configuration file will be used to apply an overall scaling to the runway effects set above. Set Turb\_Scaleg=1 for no scaling. Set Turb\_Scale=0 to disable the runway effects completely. Set Turb\_Scle=2 to scale effects by a factor of 2 etc.

## **A2.9 Articulated Projector Driver Configuration File**

---

;The name of the visual system motion driver config file to use with this set up - must be in the same directory

vis\_fname=vis.cfg

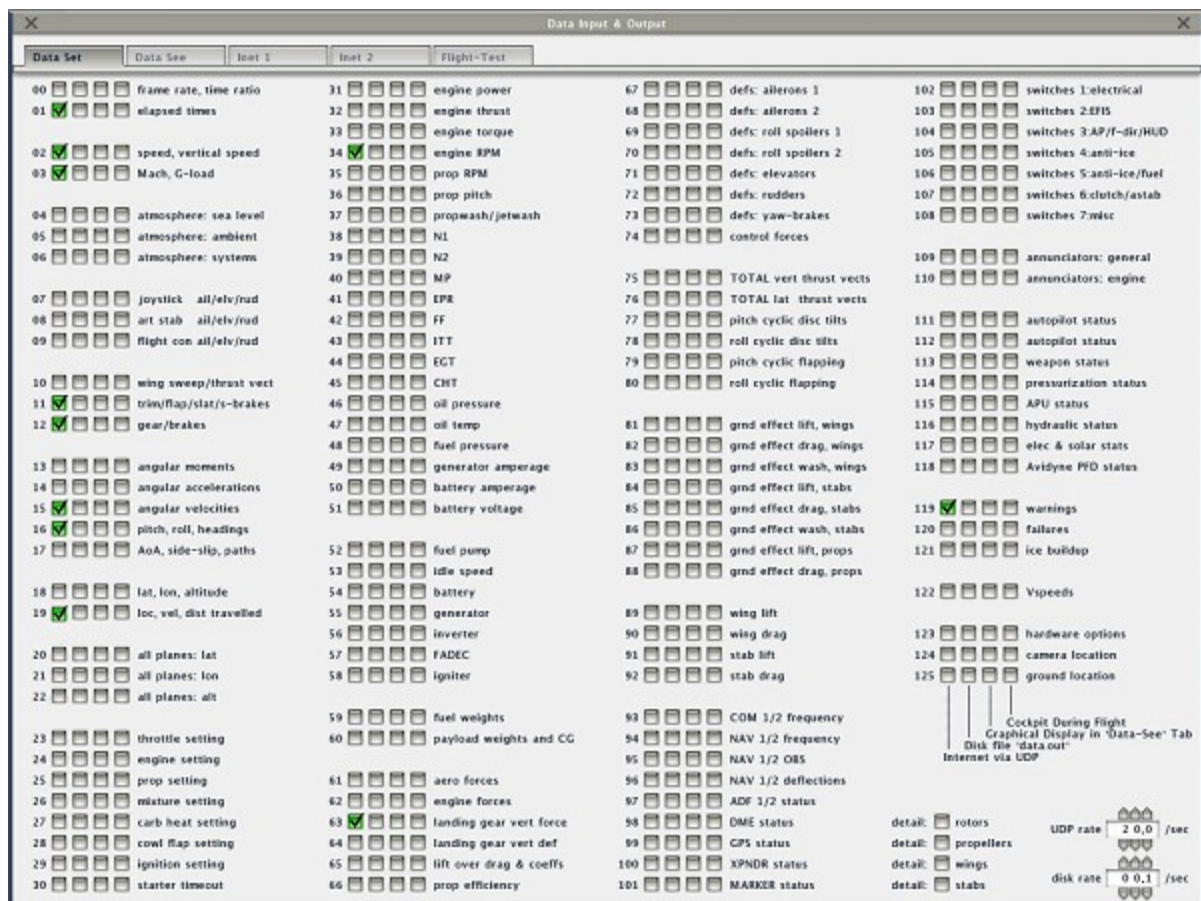
---

Set to associate different articulated projector drive settings with this .bff set up.

## Appendix 3 – Data Output Settings for X-Plane

The correct data output must be set in X-Plane for the Driver to work. To do this go to Settings – Data Input & Output in the X-Plane menu. Select the Data Set tab and check the first box for the following data items – For X-Plane 8.6 (NB the V9.0 settings are different – see next page.)

- 01 “elapsed times”
- 02 “speed, vertical speed”
- 03 “Mach, G-Load”
- 11 “trim/flap etc”
- 12 “gear/brakes”
- 15 “angular velocities”
- 16 “pitch, roll etc”
- 19 “loc, vel, dist traveled”
- 34 “engine RPM”
- 63 “landing gear force”
- 119 “warnings”



The data output rate must also be set. In the bottom right hand corner set “UDP rate” to 20 /sec. The driver will cycle slightly faster than this and so at this speed will always clear the UDP data before it gets a chance to queue. The X-Plane output rate can be slowed if desired but don't run it much faster

than this or the motion cues will begin to lag the flight.

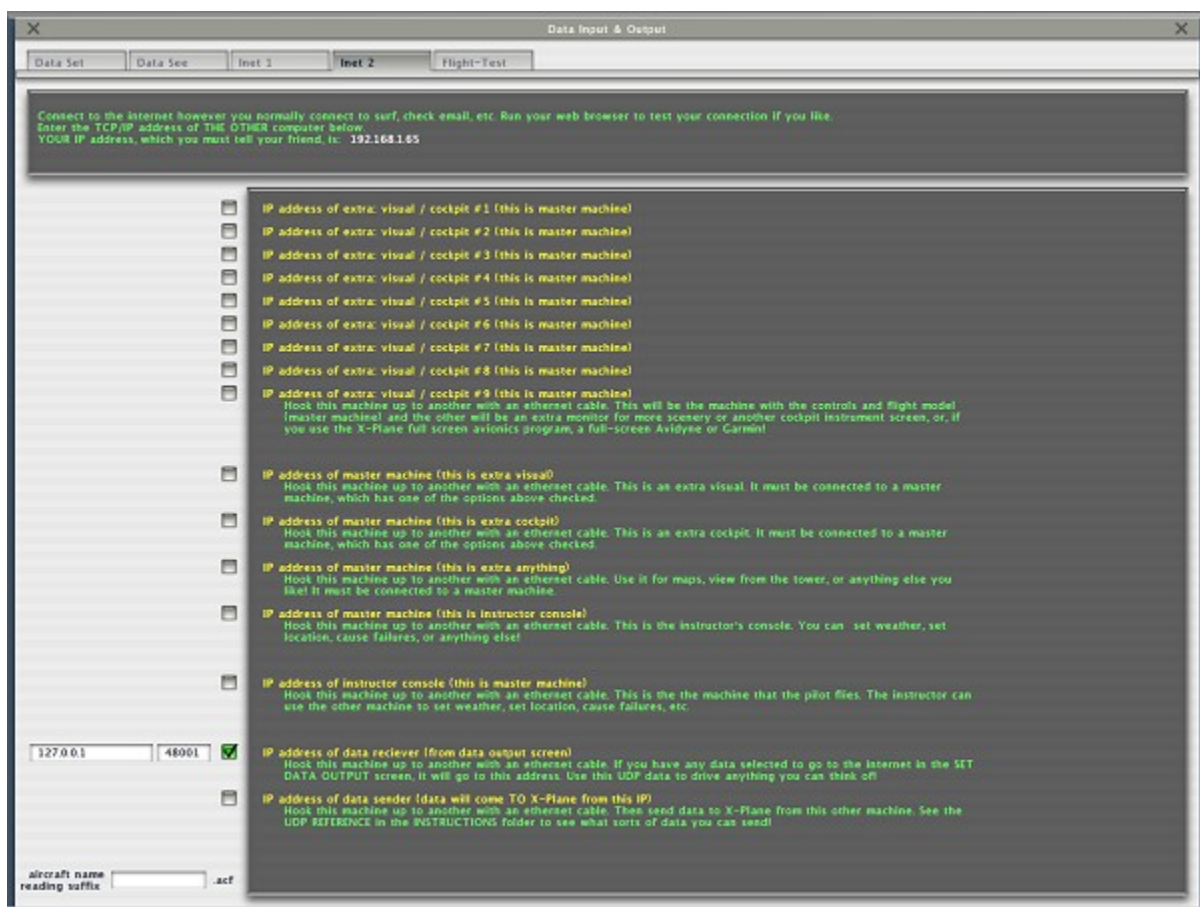
You can set more data items than this for output if you wish (eg to drive gauges etc), but setting too few or the wrong ones will produce an error message from the Driver.

The remaining setting is to specify the IP address and receiving Port of the machine on which the Driver is running.

Go to the Inet 2 tab.

Set the IP address and Port of the “receiver” machine. Use 127.0.0.1 if the Driver runs on the same machine as X-Plane. Use the LAN IP address of the Driver's machine if it is to run over a network. Set the Port number – I use 48001.

These two settings must be the same as those for the IPAddress and PortS parameters in the .bff configuration file you will be using for the Driver. Note that with networked systems your firewall settings may need to be adjusted to allow the machines to communicate or to open the specified receiver port.



### X-Plane v9.0/9.2 Settings

The above settings are for X-Plane v 8.6, for X-Plane v9.0/9.2 the data items to select are different. The data item numbers for v9.0/9.2 are -

- 01 "elapsed times"
- 03 "speed, vertical speed"
- 04 "Mach, G-Load"

- 13 “trim/flap etc”
- 14 “gear/brakes”
- 17 “angular velocities”
- 18 “pitch, roll etc”
- 21 “loc, vel, dist traveled”
- 37 “engine RPM”
- 66 “landing gear force”
- 124 “warnings”

In addition the IP Address and Port settings made previously in the Inet2 tab are now made in the Inet3 tab under “IP for Data Output” section.

It is likely that future upgrades of X-Plane will change this again.

## Appendix 4 – How to use the Shared Memory (MEM) output option.

Setting the Mode parameter in the configuration file to MEM sets the data output from the driver to Shared Memory output. In this mode the data is not sent to a COM port but is instead written directly to a shared memory area created by the Driver process. This memory area can then be accessed by other applications running on the same PC which can read the cue data in real time.

This will mainly be of interest to other application programmers and may enable other applications such as a PC based position feed back controller to convert the motion cues into speed demand signals in the PC for sending to external motor speed controllers.

The AHK script "Shared Mem Test Script.ahk" is included in the download package and shows how to access the shared memory from AHK scripted applications.

If you are using other programming languages then please refer to \_

<http://msdn2.microsoft.com/en-us/library/aa366551.aspx>

for implementation information. With reference to this web page you should look at the "second process" programming – ie that for the reading process.

The file mapping object name is – "BFF\_DRIVER\_SHD\_MEM", and its max size is 50 bytes, although only 15 are used at present. The data format is the same as the HEX2 output mode, and is:

HEX2 (and MEM) output format is - "JK" JJ XX YY ZZ 0x0D

"JK" - Start of data identifier

JJ - 1 byte - 8 bits, each indicates on/off for status indicators - see below

XX - hex value of pitch demand in 0-255 (0x00 to 0xFF) scale

YY - hex value of roll demand in 0-255 (0x00 to 0xFF) scale

ZZ - hex value of heave demand in 0-255 (0x00 to 0xFF) scale

RR - hex value for current forward ground speed in m/s, capped at 255

SS - hex value for engine 1 rpm, 0 -100 percent

1 byte - "CR" data terminator

The JJ bits are -

b7 (msb) flaps

b6 landing gear

b5 brake indicator

b4 spoiler (not currently available in X-Plane version)

b3 overspeed (not currently available in X-Plane version)

b2 stall

b1 on ground

b0 (lsb) not used

Note the refresh rate of the data in the shared memory is approx 20 sets/sec, but this is machine and load dependent.

## Appendix 5 – Key Flight Motion Events – Configuring the motion driver.

Setting the configuration parameters for the motion driver can take a fair bit of trial and error. You will get there quicker if you have good knowledge of the actual flight accelerations associated with the main motion events of a flight.

With knowledge of the actual acceleration magnitudes you can consider where you wish to set cap levels and how to fit force cue angles to the actual working range of your platform. See section 4 for details of the configuration settings.

To assist in your set up activities I've added a short list of the flight events you may wish to consider. Some of these produce relatively small acceleration changes and the set up task may be one of finding a good balance between having sufficient sensitivity in the force cueing to be able to feel the events in the platform and having enough scaling not to lose lots of the higher magnitude acceleration changes.

The more moment range your platform has the easier this balancing act is.

My suggested list of key motion events is:

Events affecting pitch motion of the platform:

1. Throttle-up on take-off.
2. On-runway braking – when landing or general manoeuvring.
3. Significant throttle adjustments in flight – eg reducing speed on approach.
4. Flaps operations – decelerations when flaps are extended and accelerations when they are withdrawn (air braking and spoiler operations may be included here).
5. Pitch attitude changes at take-off rotation.
6. Pitch attitude changes on touchdown.
7. Pitch attitude changes during normal air manoeuvres.

Items 1 to 4 affect the fore/aft force cue sensitivity and items 5 to 7 affect the pitch angle following sensitivity.

Events affecting the roll motion of the platform:

1. On-ground cornering – taxiing.
2. Lateral force impulses during flight turbulence.
3. Lateral force impulses during in-flight rudder adjustments.
4. Bank attitude changes on touchdown.
5. Bank attitude changes during normal air manoeuvres.

Items 1 to 3 relate to lateral force cue sensitivity and items 4 & 5 to bank angle following sensitivity.

And events affecting the heave motion of the platform:

1. Lift-off accelerations.
2. Touch-down bumps.

3. Short term in-flight pitch adjustments.
4. Turbulence.
5. Off-runway excursions!
6. Runway roughness (difficult unless you have good high frequency response).

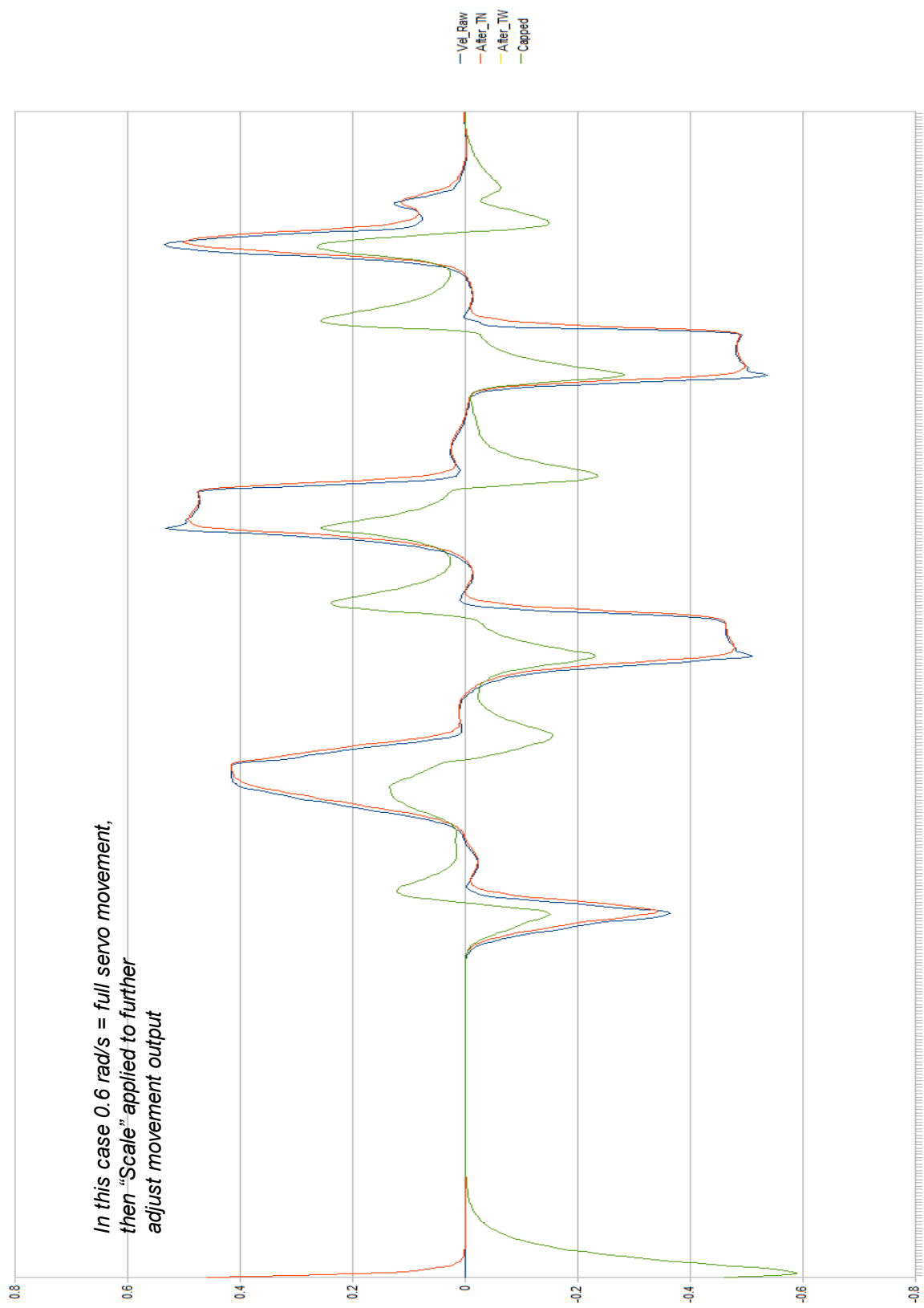
Note that I do not suggest trying to simulate sustained g-forces with the heave DOF- the ability of any motion platform to generate sustained linear accelerations is very limited

For Pitch and Roll DOF's consider the acceleration/force cue events separately from the angle "following" events as they have separate set up parameters in the configuration file.

Versions 2.1 and later of the motion driver include a Virtual Oscilloscope facility which allows the cue output from the driver to be traced in real-time and in-program settings adjustment. These two facilities allow you to make changes to the cue settings whilst the driver is active and to visually inspect the effects of those changes in the motion cues. This can be done without the motion platform being active and allows a good deal of experimentation with the settings before proceeding to test them on the live platform.

See sections 3.3.1 for more information.

## Appendix 6 – Visual Motion Driver – Configuration Settings – Roll Example.



*In this case 0.6 rad/s = full servo movement,  
then "Scale" applied to further  
adjust movement output*



With reference to the above trace:

The trace shows processing for the Roll velocity contribution to the projector tilt drive.

The graph traces are:

Vel-Raw (Blue) is the incoming raw bank velocity.

After\_TN (red) is the velocity following noise filtering (not much difference).

After\_TW (yellow) washed-out, and Capped (green) coincide because the cap level of 0.6 rad/s is above the actual values and no capping takes place.

The horizontal axis is time.

Settings in vis.cfg file for this Roll velocity example were -

TN = 0.15 rad/s

TW = 0.5 rad/s

Vel Cap = 0.6rad/s (m/s for linear DOFs)

Scale = 0.6

Invert = 1

So the green trace shows the processed roll velocity used to generate the tilt servo demand. The Vel Cap value is used to convert to servo units such that an output vel of 0.6 rad/s is fitted to the full servo range amplitude given in the config file.

Also in this example a scale factor of 0.6 is applied to the servo demand. The example is for a partial Stuart type drive in which all the servos must have the same working range and so a servo's working amplitude can't be adjusted in the vis.cfg file to suit the movement required for that individual DOF. Instead once the "shape" of the output trace is made satisfactory by adjusting the time constants and cap level (if needed for capping large peaks) the trace can be sized to give the appropriate servo movement using the Scale factor. However for independent DOF drives Scale could be left at 1.0 and the servo amplitude used to scale the output movements for that DOF.

The Scale factor is also used to weight the contributions from two aircraft motion DOF's where they are mixed to produce a single visual drive DOF – eg vertical velocity and pitch velocity when mixed to give projector vertical motion.

These settings are made for each aircraft DOF in the Config file.

The washout filter time constant has a significant effect. Longer TW time constants will slow the washout and consequently also increase the output velocity magnitudes as the washed-out trace approaches the shape of the raw vel trace. Shorter TW time constants will return the trace to zero more sharply and consequently also reduce it's magnitude. It is likely then that a change in TW may also need a change in Scale to keep the final movement range the same when adjustments are made to slow or speed the projector movements.

The output data can be dumped to file by setting the itrace variable in the main motion driver config file (not in the smaller vis.cfg file).

To dump data to the BFF-output.txt file in the driver's working directory set the "itrace" variable in the driver's config file. Don't leave this on during normal use of the driver. When non-zero the driver will store data during the flight (keep it short) and then dump the data to file when closed.

The itrace settings are:

= 5 to dump all the raw velocities – useful for scoping the full range of flight velocities during different manoeuvres.

= 6 to dump the lateral velocity processing

= 7 for yaw

= 8 for vertical

- = 9 for pitch
- = 10 for roll
- = 11 for fwd (not used in processing currently)

Data items for these are -

- raw vel
- post noise filtering
- post washout
- post capping
- post inversion
- servo movement
- final servo demand,

- = 12 to dump all the processed velocity outputs in a single output

The data is appended to the BFF-output.txt file if it already exists. If you want a clean file each run then delete the old one first. The file can be opened in a spreadsheet as a space delimited file and the data plotted therein.

## Appendix 7 – Integration with Galil Optima Series, DMC-18x2 and DMC-21x2/3 industrial motion controllers.

Version 2.1 of the BFF Motion Driver introduces support for [Galil Optima Series, DMC-18x2, and DMC-21x2/3 motion controllers](#). These are high specification industrial motion controllers and can be used for controlling multiple channel servo systems.

The control scheme adopted utilises the Position Tracking (PT) motion mode of the above Galil controllers. In this mode the DMC controller is programmed to follow on-the-fly position demands sent from the Motion Driver to the controller through the DMCwin32 Galil Windows API. In this motion mode the DMC controller handles all the closed loop servo processing and the scheme can result in high fidelity motion response to the motion cues exported by the motion driver. It will be of particular interest to commercial platform builders.

To implement the scheme a DMC code control program must be written for the controller. This program is held in a text file and is loaded into the controller by the motion driver and executed by the driver at run time under operator instruction. The DMC program will include the core commands which put the controller into PT mode and which instruct the controller to read the position demands sent from the driver and use them as its position targets. The program will also include appropriate startup encoder calibration, motion parameter settings such as maximum speeds and accelerations and PID gain settings and platform parking and error handling functions.

An **example** of the DMC program is shown below. Please consult the Galil controller and DMC programming manuals for full details of the controller programming facilities. You will need to write the program to suit you particular drive system and performance requirements. Comments in **blue** are explanatory and would not be included in the program.

Contents of BFFP1.dmc text file -

#BFFP1	Program start label – must be as shown
KP5,5,5	PID gains – values illustrative
KI0.005,0.005,0.005	
KD0.8,0.8,0.8	
DP-1080,-1080,-1080	Set absolute position – known when physically parked – values illustrative
FL1000,1000,1000	Set travel limits – values illustrative
BL-1000,-1000,-1000	
SP200,200,200	Set slow speed – all values illustrative
AC67107840,67107840,67107840	Set acceleration (high for PT mode)
DC67107840,67107840,67107840	Set deceleration
SH	
PA0,0,0	Send to start/hold position
BG	Begin motion
pos1=0	Set initial PT mode position demands
pos2=0	pos1, pos2 and pos3 are the internal variables that hold the current position demands
pos3=0	these are updated by the motion driver in real-time
park=0	Set park instruction variable to 0
WT10000	Wait for the platform to settle at hold position
SP30000,30000,30000	Set PT mode working max speed
PT1,1,1	Initiate PT mode in each channel
#LOOP	Start of PT mode internal loop

PApos1,pos2,pos3	Set pos1, pos2 and pos3 as the current position targets (updated in the background) by the motion driver
IF(park=1)	Go to #PRK subroutine if park variables goes to 1
JP#PRK	Jump to #PRK subroutine
ENDIF	
WT10	Wait 10ms
JP#LOOP	Go back to start of PT loop
#PRK	#PRK Park subroutine
ST	
SP200,200,200	Set slow speed
PA-1080,-1080,-1080	Go to park position
BG	
AM	
WT4000	
ST	
MO	shut down motors
EN	End

Use of **#BFFP1** and **#PRK** program labels is mandatory, as is the use of the variables **pos1**, **pos2**, **pos3** and **park** to hold the position demands and parking instruction.

Communication with the motion driver requires only that the DMC controller has been “registered” on the host PC using the Galil software provided for the controller. The communications mechanism is either TCP/IP or UDP or by RS232 Serial comms and is specified using the Galil software provided for use with the controller. Once so registered the motion driver can communicate with the controller regardless of the connection mechanism implemented.

## Operation

The DMC control program must be saved in a “BFFP1.dmc” file in the motion driver's working directory.

The data output mode of the motion driver should be set to Mode=GAL in the .bff configuration file. The working range of the actuators and start/hold positions should also be set in the .bff configuration file in encoder quadrature counts AS USED BY THE DMC CONTROLLER.

Ensure that a copy of the DMCwin32.dll is installed on the host PC. This should automatically be the case if the Galil controller software has previously been installed and the controller has been registered on the PC.

On Driver startup the user will be prompted to confirm loading and execution of the #BFFP1 DMC program to the controller. If programmed appropriately the controller will go through the programmed startup calibration and slow travel to start/hold position routine and then enter Position Tracking mode. On confirmation from the operator the motion driver will then commence exporting motion cue position demands to the controller.

On driver shut-down the operator will be prompted to confirm execution of the #PRK parking subroutine in the DMC program which will take the controller out of PT mode and send it at slow speed to its park position.

NOTE Please consult the Galil documentation for your controller to establish a suitable DMC program for your system. The above program is illustrative.