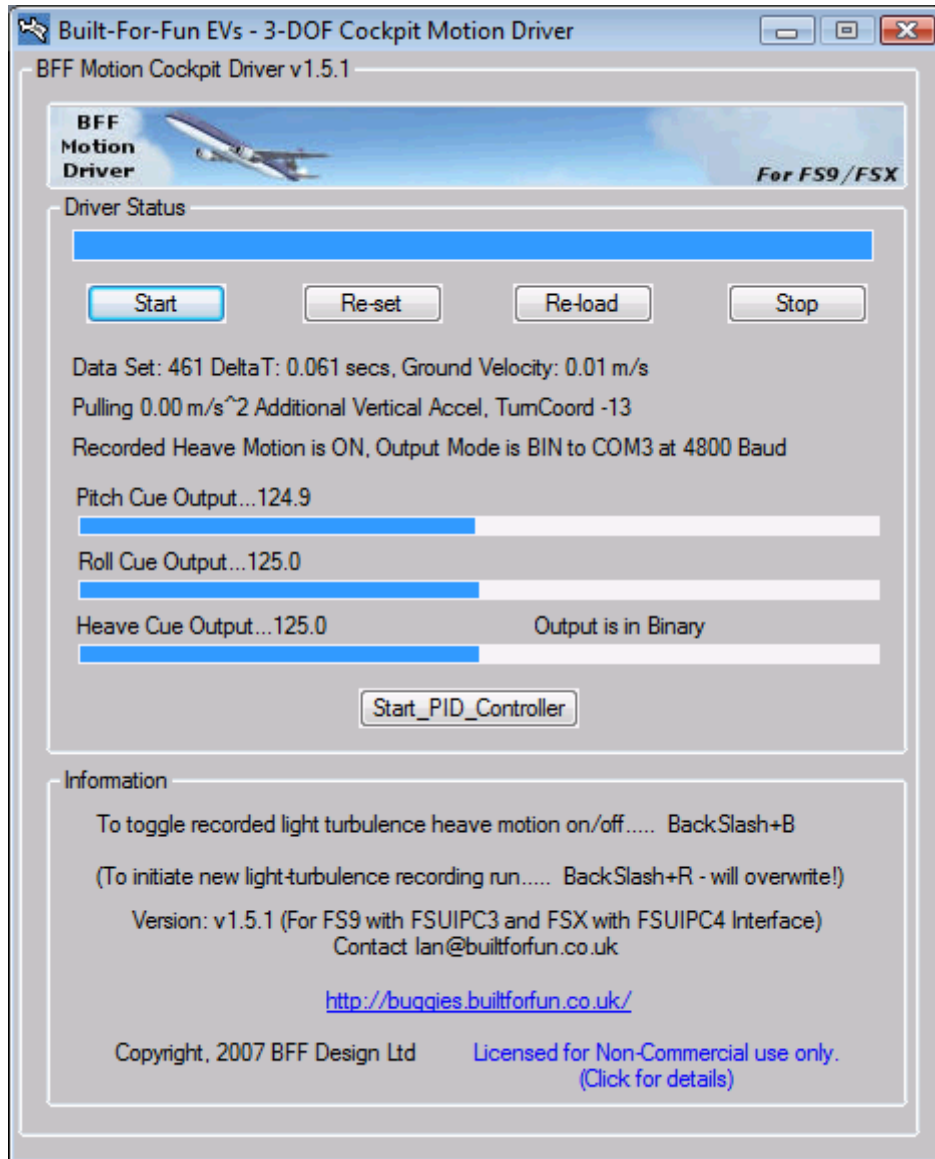


BFF 3 Degrees of Freedom Cockpit Motion Driver



User Manual

(For Driver, System Setup & PID Servo Controller Applications,
Man. Rev 1.6)

For FS9/X (v1.6) & X-Plane 8.68/9.0 beta (v1.3) Driver Versions

Ian Hopper (BFF Design Ltd)

© 2007, 2008 BFF Design Ltd, All rights reserved.

Table of Contents

BFF Motion Driver – User's Guide.....	3
1. What is it & what does it do?.....	3
2. How it works.....	3
2.1 Basis of the Motion Cue Calculation.....	4
2.1.1 Pitch.....	4
2.1.2 Roll	4
2.1.3 Heave.....	4
2.1.4 Coupling.....	5
3. Installation & Operation.....	6
3.1 Data Output Formats.....	6
4. Driver_Setup_v1.6.....	9
4.1 Motion Driver Settings.....	10
4.1.1 COM Port Settings.....	10
4.1.2 Driver Start Positions.....	10
4.1.3 Initial Transmit Delay.....	11
4.1.4 Pitch Cue Settings.....	11
4.1.5 Roll Cue Settings.....	14
4.1.6 Heave Cue Settings.....	16
4.2 Save Settings & Exit.....	18
4.3 SPU PICAXE 18X Chip Settings.....	19
4.4 Setup-3pt.exe For 3 Point Support Type Platforms.....	20
5. Motion Driver Operation.....	21
6. BFF PID Servo Controller.....	24
6.1 Visual Data Outputs.....	25
6.2 Controller Settings.....	26
The Proportional Term.....	26
The Integral Term.....	26
The Derivative Term.....	26
Minimum Speed Demand.....	27
Setup Procedures.....	27
6.3 Data Output for Analysis and Inspection.....	28
LICENSE.....	29
Appendix 1 – Optional Data Output for Setup/Debug.....	30
Output Variables.....	30
Appendix 2 – Manually-set Config File Parameters.....	32
Appendix 3 – Data Output Settings for X-Plane.....	35
Appendix 4 – How to use the Shared Memory (MEM) output option.....	38

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. These motion cues are derived (by the relevant version) from flight data extracted in real-time from MicroSoft(FS9 or FSX), Laminar Research X-Plane 8.68 / 9.0 Beta 2 and Lead Pursuit Falcon 4.0 Allied Force flight simulators. It does this 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 in which the actuators' combined movements generate the DOF movements. In its default configuration the 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.

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 that 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.

2. How it works.

The driver runs on the same PC as FS9/FSX and extracts aircraft movement flight data from FS9/FSX during your simulated flight. It does this using Pete Dowson's FSUIPC interface. 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 and can be read by microcontroller based control units. 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.

The current FS version will not run over a network via WideFS – perhaps in a future version. For X-plane however the driver can run on the same machine as the X-Plane process or LAN networked.

It is probably helpful to state for the benefit of Newbies to the subject that the **position** demand data output from the driver via the COM port 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 Signal Processor Unit and control system represents 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 but the BFF Signal Processor Unit has been designed to interface directly with the BFF Motion Driver and can be setup directly from the Driver Setup program on the PC. More of this later....

Version 1.5 of the driver now supports 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.

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. The driver can also however 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 cuing and an element of pitch-rate effect. The “gravity alignment” element of the pitch angle is calculated from the actual 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 user 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 by the user 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 stationary 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 however the second direct roll angle based element is superimposed on the cue. This element is also washed-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 the user 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 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.

3. Installation & Operation

Installation is fairly straightforward. 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. The motion driver package should be unzipped to a dedicated directory on the same machine as FS is running. It will not currently operate over a network through WideFS. The driver takes up little computing resource and running it on the same machine as FS should not significantly affect frame rates.

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.

The two working programs are the setup program System_Setup.exe and the motion driver itself BFF_Motion_Driver.exe. These should be run from the installation directory. 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 driver can be configured using the 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.

To run the motion driver first start FS in windowed mode, then run the motion driver. The driver will suspend output until a flight is active and will further suspend output when the flight is paused or otherwise interrupted. Depending on the behaviour of your control system this may hold the motion platform position at the last demanded position until the flight and hence the driver output resumes.

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.

3.1 Data Output Formats

In v1.0 there are three data output formats available – binary, hexadecimal and shared memory. See the System_Setup section to see how to select these.

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 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. A fourth 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 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 BFF 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.

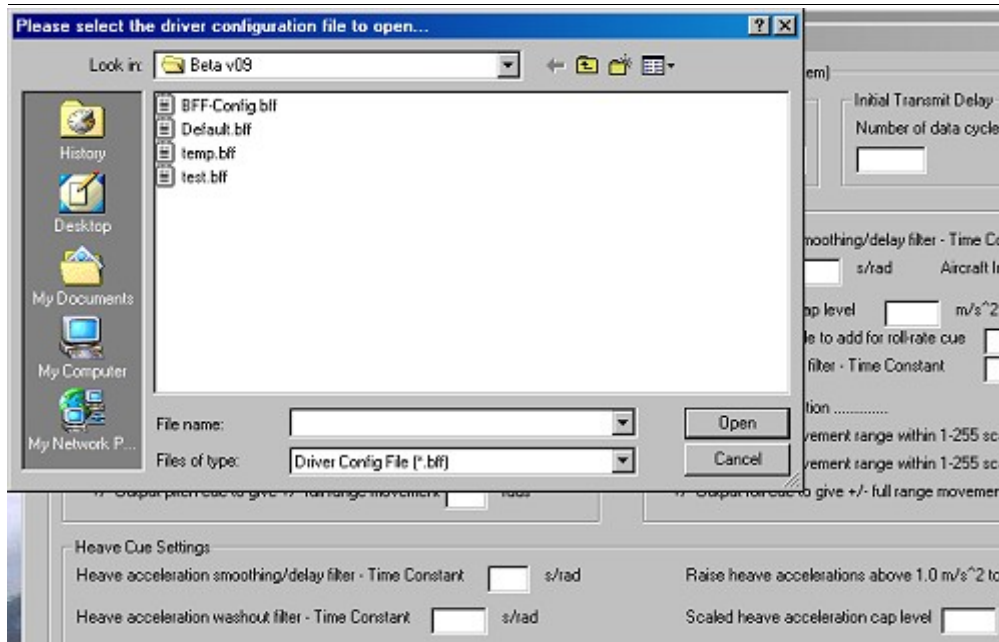
The driver works at 2400 & 4800 baud rates, and higher rates succeed in sending data that can be read by Hyper Terminal through a Null Modem cable, however you will have to test the higher rates yourself with actual external hardware. For the MEM output the data is written to the shared memory area at the refresh rate of the Driver.

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.

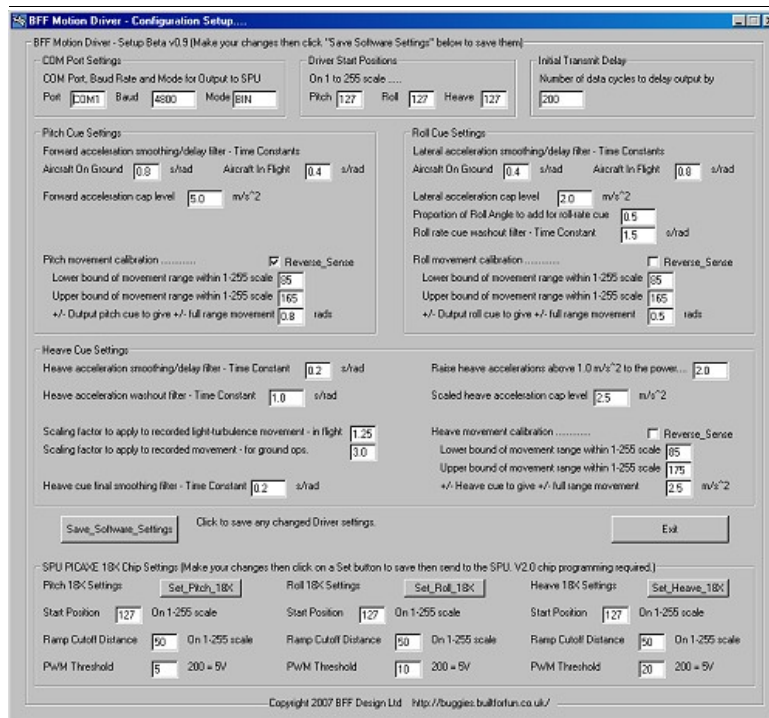
NOTE 2, there is a further manually-set parameter in the configuration file - Use_Bool. If Use_Bool=0 the JJ bits and final SS engine RPM byte are disabled. The b1 (on-ground) bit of the JJ byte will alone be active – the rest will always be zero, and the SS byte will also always return zero. This setting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

4. Driver_Setup_v1.6

The driver setup program runs separately from the Motion Driver and allows you to modify the setup parameters for the driver. The setup program does this by reading from and writing to a *.bff configuration file which can subsequently be opened by the motion driver at start-up. A default configuration file is included in the package which can be used as-is or modified to suit.



Driver Configuration Interface



As of version 1.6 of the driver set up program there are now two separate 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. This change has been made to simplify the set up for the 3 point type rigs. Section 4.4 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 Driver_Setup_v1.0 exe file. A file open dialogue will be displayed from which you can chose the configuration file you wish to inspect or modify.

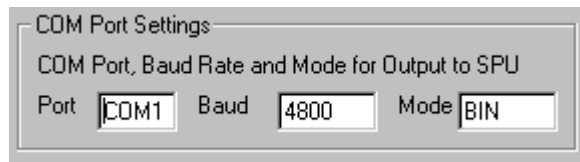
If you cancel without opening a file the driver setup program will close. Once the configuration data has been read and loaded into the program you can go on and edit the individual values.

The setup program has two main working areas; the upper area deals with the settings for the motion driver. The lower area provides setup functions for the BFF Signal Processor Unit (SPU). Both sets of data are stored in the same .bff configuration file however the SPU set up is performed “live” and requires the SPU to be powered up and connected to the configured COM port. If you are not using the BFF SPU just ignore the lower settings.

4.1 Motion Driver Settings

Starting at the top left.....

4.1.1 COM Port Settings

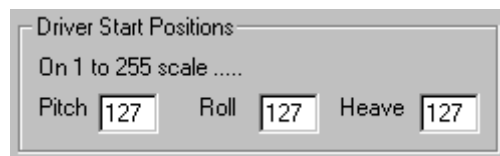


Port number, baud rate and data output Mode. The output is always 8 bit, no parity, 1 stop bit. The output Mode is as described in the Installation section above and can be BIN, HEX2, MEM or NONE. If you input something else the mode will default to BIN. The BFF SPU uses a baud rate of 4800. Beta users have tested faster baud rates and report no problems – let me know how you get on with them. 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.

For v1.5 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



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

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 cuing 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.

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.

THE "SEND_TO" BUTTONS NEXT TO THE RANGE INPUT FIELDS SHOULD BE USED WITH CARE! They will send a command to the configured COM port (or Shared Memory Area) instructing the platform to move its Pitch actuator (or FRONT actuator in 3 point systems) to the position set in the adjacent input field ie to the current upper or lower range limit. The Roll (RIGHT actuator) and Heave DOF's (LEFT actuator) will be sent to their "Start Positions" at the same time (see above); if you do not want them to move at all during this set up process disable their drives on the platform.

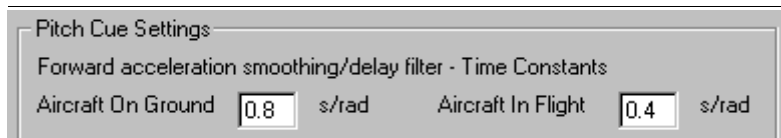
If you are using the BFF 4 -chip SPU the motion rate will be set to "slow" so that the platform will not move at full speed during these operations. If you are not using the BFF SPU, but are using a controller that does not utilize the BIN output mode movement rate (byte1) code then the output will be treated as a step demand for the set position and your rig may move rapidly to the demanded position – be careful.

The buttons can be used to determine the actual travel limits of the rig – start at a position you know to be within the working range of the movement – eg at a nominal mid setting of 127. Then increment the limit position (either up or down) sending the updated position demand each time to slowly move the rig towards its end stops. You can then decide where you want to set the upper and lower limits.

"+/- 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 $0.707g (\sin 45^\circ)$ ie $\sim 7.1 \text{ m/s}^2$. You can set here an upper limit acceleration to be used for the cue calculation. Logically you can set this = $\sin(\text{+/- movement angle}) * 9.81$ where you have set up a 1 to 1 calibration and you want to use the full movement range. The 5.0 m/s^2 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°.

"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 cuing – 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

Proportion of Pitch Angle to add for pitch-rate cue	<input type="text" value="0.05"/>	
Pitch rate cue washout filter - Time Constant	<input type="text" value="1.5"/>	s/rad

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 cuing element.

<input checked="" type="checkbox"/> Reverse_Sense	
Scale <input type="text" value="80"/>	<input type="button" value="Send_P1"/>
Scale <input type="text" value="160"/>	<input type="button" value="Send_P2"/>

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 cuing 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.

Roll Cue Settings

Lateral acceleration smoothing/delay filter - Time Constants

Aircraft On Ground s/rad Aircraft In Flight s/rad

Lateral acceleration cap level m/s²

Lateral acceleration washout filter - Time Constant s/rad

Proportion of Roll Angle to add for roll-rate cue

Roll rate cue washout filter - Time Constant s/rad

Roll movement calibration

Reverse_Sense

Lower bound of movement range within 1-255 scale

Upper bound of movement range within 1-255 scale

+/- Output roll cue to give +/- full range movement rads

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 cuing 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/s² (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

Proportion of Roll Angle to add for roll-rate cue	<input type="text" value="0.5"/>
Roll rate cue washout filter - Time Constant	<input type="text" value="1.5"/> s/rad

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

Heave Cue Settings	
Heave acceleration smoothing/delay filter - Time Constant	0.2 s/rad
Raise heave accelerations above 1.0 m/s ² to the power...	2.0
Scaled heave acceleration cap level	2.5 m/s ²
Heave acceleration washout filter - Time Constant	1.0 s/rad
Heave cue final smoothing filter - Time Constant	0.2 s/rad
Scaling factor to apply to recorded light-turbulence movement - in flight	1.25
Scaling factor to apply to recorded movement - for ground ops.	3.0
Heave movement calibration	<input type="checkbox"/> Reverse_Sense
Lower bound of movement range within 1-255 scale	85 Send_H1
Upper bound of movement range within 1-255 scale	175 Send_H2
+/- Heave cue to give +/- full range movement	2.5 m/s ²

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 cuing 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.

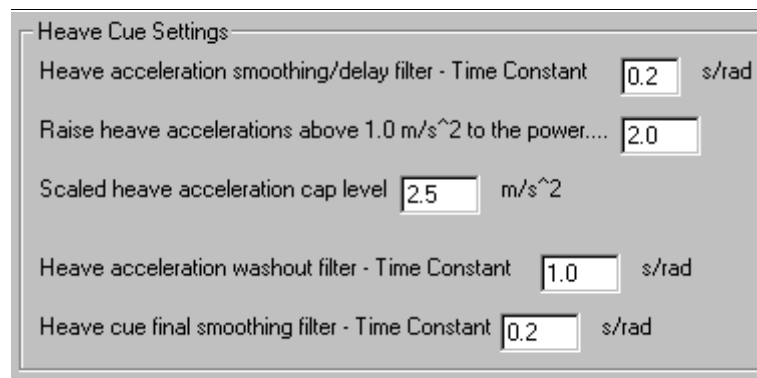
Heave Movement Calibration

Heave movement calibration	<input type="checkbox"/> Reverse_Sense
Lower bound of movement range within 1-255 scale	85 Send_H1
Upper bound of movement range within 1-255 scale	175 Send_H2
+/- Heave cue to give +/- full range movement	2.5 m/s ²

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/s². Why this is is described below. Logically the full range movement cue should be set equal to the “acceleration cap level” described below. The “Send” buttons and “Reverse” checkbox perform the same functions as before – again, BE CAREFUL using the send_to buttons – they will cause the platform to move if it is powered up and connected.

See Section 4.2 for more information about these settings with a 3-point support type platform.

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".

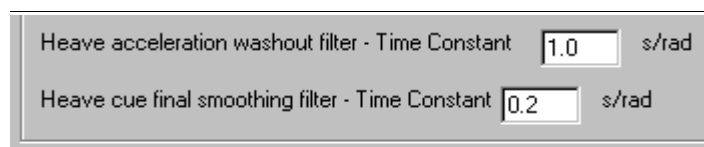
The actual acceleration has to go through a fair bit of processing however to yield a cue that can be used in the motion platform. The basic approach implemented in the BFF Motion Driver is to use a capped and washed-out acceleration profile as the position demand for the heave movement. Whereas the 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/s² to the power stated – ie 2 will square the acceleration value, 3 will cube it etc. Values on or below 1.0 m/s² 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/s² (0.25g) setting illustrated is one I've found works for my rig over a range of flying conditions.



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	1.25
Scaling factor to apply to recorded movement - for ground ops.	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 recorded movement which is superimposed on the actual heave movement during higher speed runway operations.

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 BFF 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 SPU PICAXE 18X Chip Settings

SPU PICAXE 18X Chip Settings (Make your changes then click on a Set button to save then send to the SPU. V2.0 chip programming required.)

Pitch 18X Settings	Set_Pitch_18X	Roll 18X Settings	Set_Roll_18X	Heave 18X Settings	Set_Heave_18X
Start Position	127 On 1-255 scale	Start Position	127 On 1-255 scale	Start Position	127 On 1-255 scale
Ramp Cutoff Distance	50 On 1-255 scale	Ramp Cutoff Distance	50 On 1-255 scale	Ramp Cutoff Distance	50 On 1-255 scale
PWM Threshold	5 200 = 5V	PWM Threshold	10 200 = 5V	PWM Threshold	20 200 = 5V

If you are using the 4-PICAXE chip BFF Signal Processor Unit as part of your drive and control system you can use the setup program to set the key SPU behaviours from your PC without having to re-program the micro-controller chips. You will need to have flashed the SPU chips with the v2.0 programming for this to work.

This is NOT for use with the newer single-chip SPU which runs with the BFF PID Servo Controller – the hardware control settings for this are set in the Servo Controller software.

SPU PICAXE 18X Chip Settings (Make your changes

Pitch 18X Settings Set_Pitch_18X

Start Position 127 On 1-255 scale

Ramp Cutoff Distance 50 On 1-255 scale

PWM Threshold 5 200 = 5V

The settings are similar for each 18X chip – those for the Pitch chip are described.

“Start Position” The position on the 1-255 scale representing the full voltage range output of the feedback potentiometer of the DOF to which the SPU will move the rig when it is powered up. Remember that the SPU is likely to be powered up and waiting before the Motion Driver starts sending data so this value is important. It should be set within the actual physical movement range of the rig and it makes some sense to match it to the “Start Position” used by the Motion Driver for the DOF. On startup the SPU sets the movement rate for each DOF to slow so that the initial positions are taken up slowly – as each 18X chip initializes it will move the DOF it controls slowly to its start position.

“Ramp Cutoff Distance” As the pitch angle of the cockpit approaches the position currently demanded by the motion driver the 18X chip will slow its drive speed. This “Cutoff” setting sets the distance from demanded position at which this slow-down starts. If the distance is short the rig will accelerate quickly when the demand position moves away from its current position, but it may not be able to slow to a stop fast enough once the rig reaches the demanded position - so might overshoot. If it is set long the accelerations will be more gradual and there will be time to decelerate without overshoot but the response will be slow. There is a balance to be found here - try to set this distance low to give prompt response but not so low as to cause frequent overshoots during normal operation - trial and error is needed, which is why the ability to set this from the PC is of great assistance. The distance must be set with the actual travel range of the DOF in mind – which is not necessarily the full output range of the feedback potentiometer.

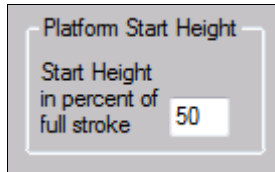
“PWM Threshold” This setting defines the minimum PWM voltage output level to be sent by the chip to its motor speed controller. The purpose is to set a minimum drive torque level from the motors to overcome any controller modulation or system mechanical friction effects which might otherwise delay movement of the rig until the speed demand reaches a high enough level to cause movement to occur. Inertia in the system (mainly noticeable in the heave DOF) will also slow the speed response of DC PM motors. Running speed of these motor types is load dependent. Setting this value too high

will cause the DOF movement to “twitch” when it should be stationary, so set the value just below that which causes a twitchy response. With this setting a rise in the speed demand voltage from the 18X chip should cause a near immediate movement response from the rig.

4.4 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.

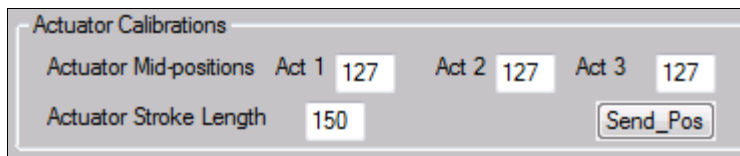
Most of the cue generation settings are the same as those used in the uncoupled rig type setup and you should refer to section 4.1 above for these. The following items are particular to the 3-pt setup.



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

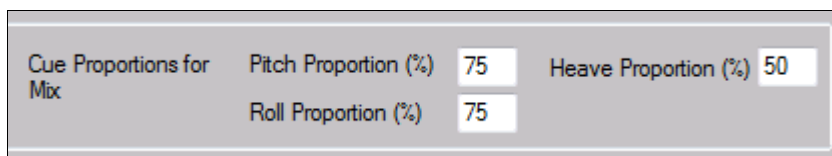
The feedback pot centering on each actuator may not be the same and this setting allows you to define for each actuator where its

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.

Send_Pos – In the 3-point setup program there is only a single Send-Pos button. This will send a position demand to the configured COM port or Shared Memory to send each actuator to the positions given in the “Mid-positions” fields. BE VERY CAREFUL WITH THIS as it will send a step demand to your platform if it is connected and active. This function can be used to determine the actuator working lengths and mid-positions prior to making your final settings.



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 1/3rd 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.

5. Motion Driver Operation

To start the BFF Motion Driver double click on the BFF_Motion_Drive.exe file or on a shortcut pointing to it. You will need to start your flight simulation program first.



BFF 3 DOF Motion Driver

Operation is fairly straight-forward.

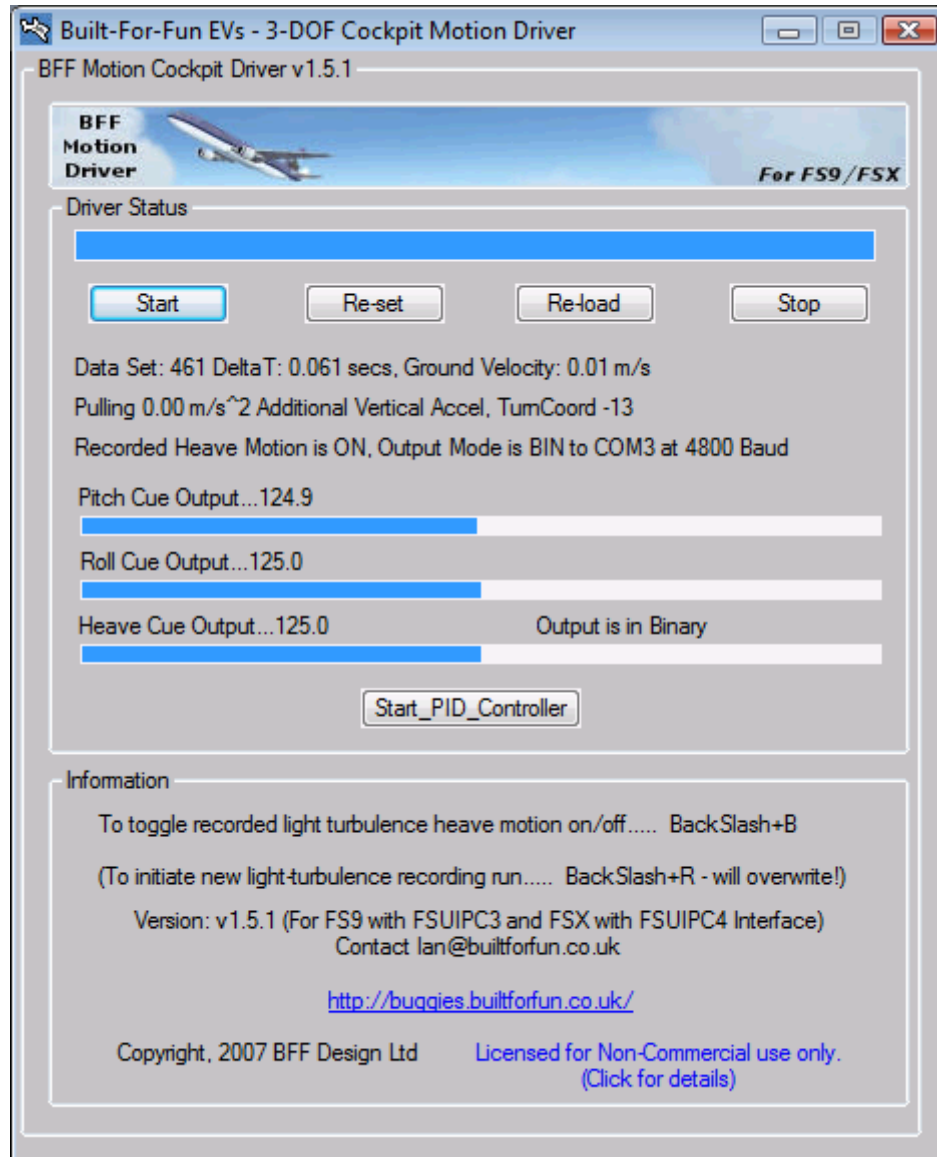
To start the driver click Start; for unregistered versions you will be given an option to register or continue in Demo mode. Registered versions are tied to the PC from which they were registered – if you wish to run the software on a 2nd PC go through the registration process again on the 2nd PC. If 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 output until the flight resumes.

For the X-Plane version an option is given to continue even if an X-Plane session is not found on the local machine. This is to allow the Driver's use on a networked machine.

For an active flight the cue output 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 live cues are sent to the COM port. The cue outputs are displayed both digitally and in the analogue progress bars – see below.



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 driver

starts with this on.

\+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 data is also used to generate additional heave motion during take-off and landing runway operations. However in this case it is played back at twice normal speed to simulate the higher frequency movements associated with on-ground running. The amplitude of the movement is scaled in proportion to ground speed and capped at a speed of 25 m/s (~ 50 knots), so the effect builds as the aircraft ground speed increases.

Both effects can be scaled or disabled using the appropriate settings in the Driver_Setup program.

To stop and close the driver hit the Stop button (or hit Tab).

To re-start with the current configuration file click the Re-Set button.

To re-load a new configuration file click the Re-Load button.

To start the BFF PID Servo Controller application click on the Start_PID_Controller button – this will require the data output mode to have been set to MEM, otherwise the Servo Controller will not start.

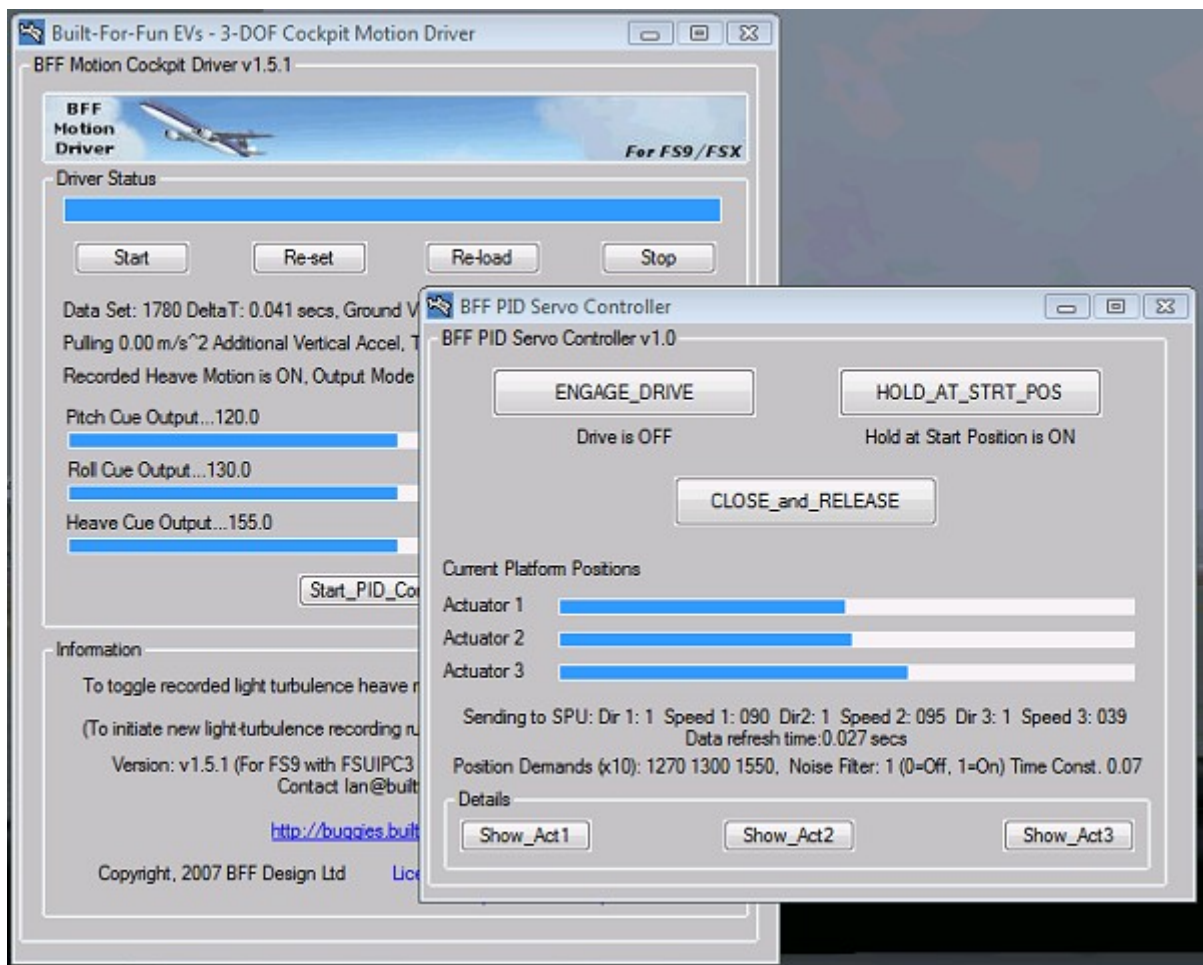
Some additional information about the motion is reported on the interface.

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

6. BFF PID Servo Controller

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 can now be 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 together with 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 MSFS Motion Driver using the “Start_PID_Controller” button.



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.

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:

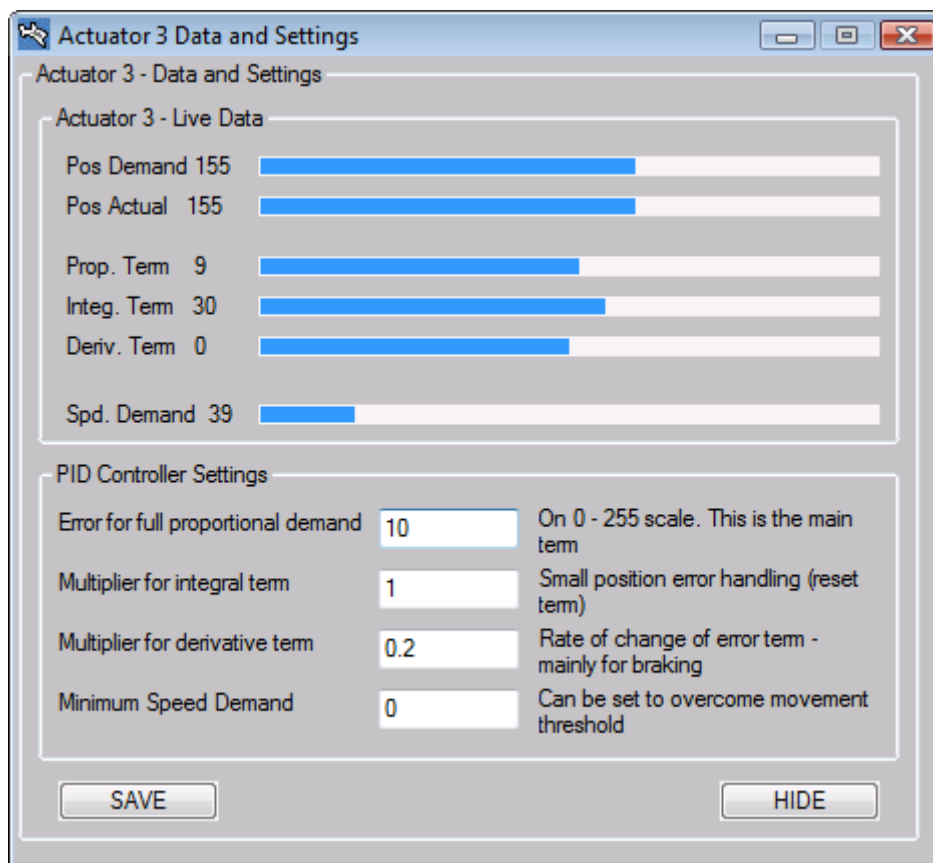
- The PID Servo Controller runs as a separate process from the BFF Motion Driver and will continue to issue speed commands to the SPU if the flight sim or Motion Driver hang.
- The Servo Controller monitors the comms with the SPU and issues a warning message if there is a loss of position feedback data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

6.1 Visual Data Outputs

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 3 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.

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.

6.2 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 mainly acts as a brake on the system to reduce overshoot. 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.

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 5 and 20 as a guide only.

The Integral Term

Also know 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 is mainly as a brake and to help to reduce the extent of any overshoot there may be. 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.

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 at either the hardware level or within the servo controller software might help.

To this end 4.7 μ F capacitors might be fitted to the feedback inputs to the SPU PICAXE chip, or the built-in noise filter in the controller can be activated. To do the latter the third last line in the PID.cfg file should be changed from 0 to 1 and, if required, the time constant for the filter adjusted by altering the second last value. The default value for the time constant is 0.07 rad/s, larger values will filter more noise but will introduce more delay the signal.

Guide values for the Derivative term multiplier – from 0 to 1 or 2.

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 the System Setup program that you use to set the configuration file settings for the motion driver can be used to send step position demands to the controller without the need for your flight simulator to be running. You can use the “Send_To” buttons to do this – see section 4.1.4.

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.

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.

6.3 Data Output for Analysis and Inspection

The best way to understand the response of your platform is to plot the motion data. If the last line 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 -

- Filtered and unfiltered position feedback
- Demanded position
- Position error
- Proportional, Integral and Derivative speed demand terms
- Total speed demand

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.

Ian Hopper
BFF Design Ltd
20th December 2007

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 - GNU GPL versions are available at <http://buggies.builtforfun.co.uk/> provided for these purposes.

Removal or modification of the License file or of the BFF_Registration.ini file will disable the software.

BFF Motion Driver is copyright ©, 2007 BFF Design Ltd.

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

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.

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. Ie 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/s².

itrace=2

- A) Vertical component of accel acting on pilot (minus 1g)
- B) Same accel after 1st 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 3rd column may be of a much higher magnitude than the others. All values in m/s²

itrace= 3

- A) Fore/aft component of accel acting on pilot (m/s²),
- B) after low-pass smoothing filter (m/s²)
- 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/s²),
- B) accel after any scaling has been applied (see note below).
- C) after low-pass smoothing filter (m/s²)
- 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.

Appendix 2 – Manually-set Config File Parameters

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 -

; 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.

; 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.

;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

;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 this one

;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.

;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.

;For X-Plane version, 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 versions of the motion driver. This uses UDP data transfer to obtain the required flight data from X-Plane. IPAddress is the IP address of the machine on which the Motion Driver is running. If it is running on the same machine as X-Plane 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. This should also be the address set in the X-Plane “Inet 2” set up page for the data receiving machine.

;For X-Plane version, Receiving Port on Driver's machine, defaults to 48001

PortS=48001

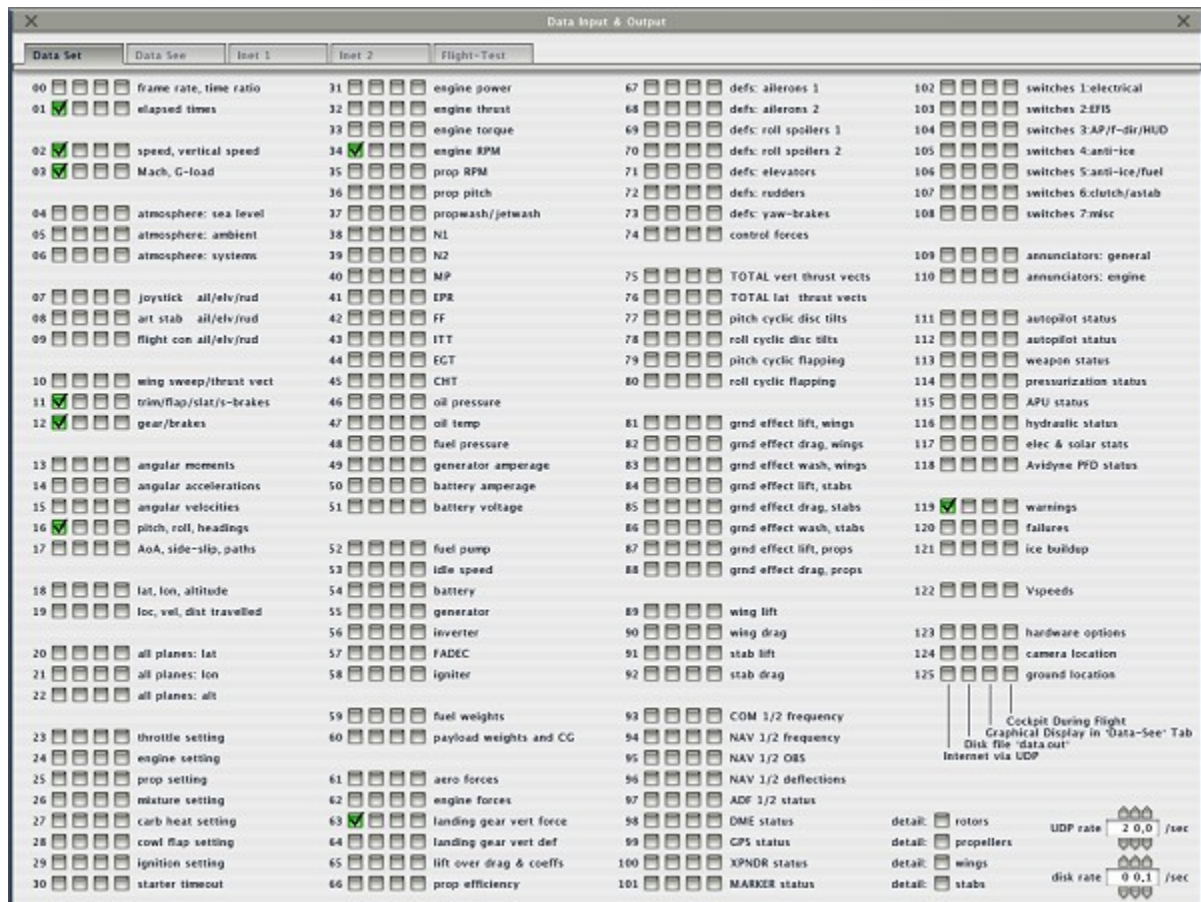
;If using a networked machine you may have to open port PortS on your firewall to enable it to receive data.

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.

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”
- 16 “pitch, roll etc”
- 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.

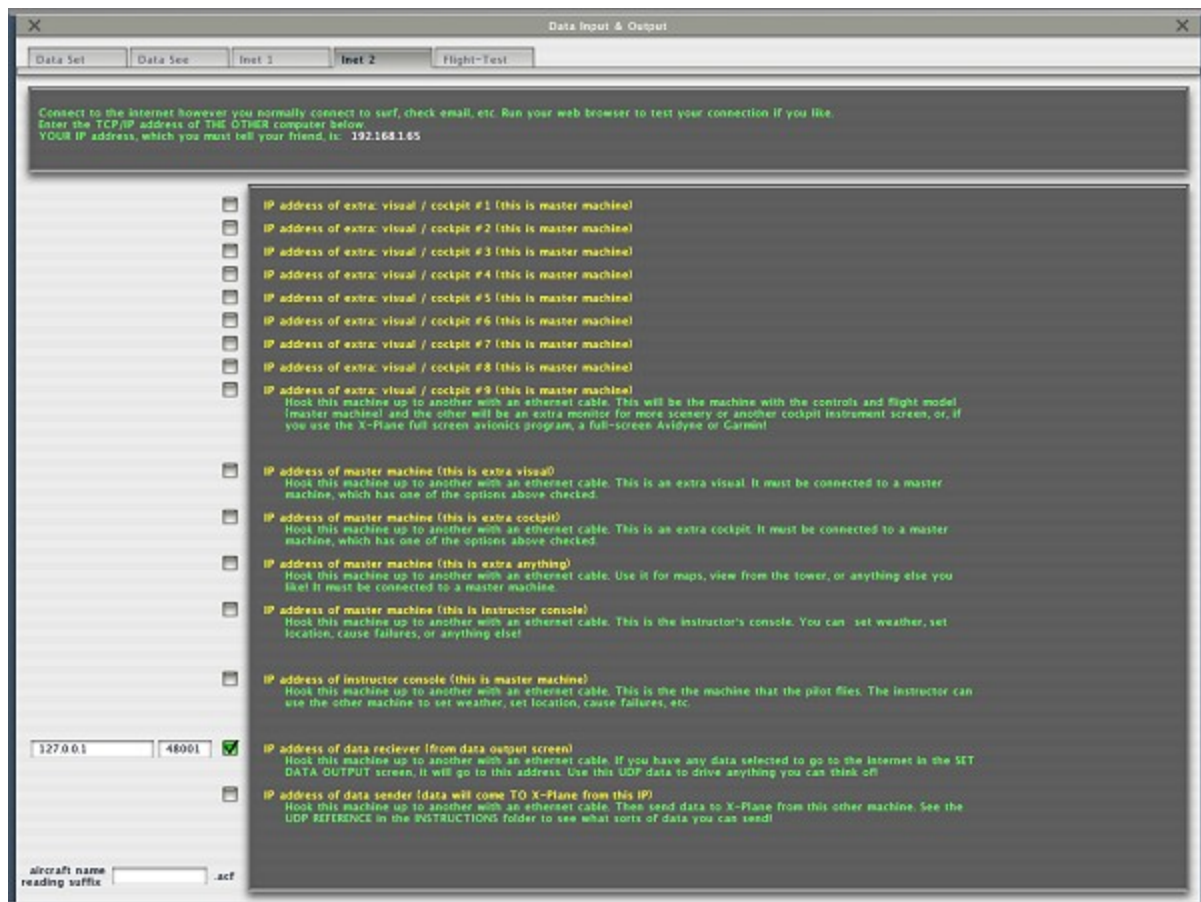
You can set more data items than this for output if you wish (eg to drive gauges etc), but setting too few or the wrong ones will produce an error message from the Driver.

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 Beta 2 Settings

The above settings are for X-Plane v 8.6, for X-Plane v9.0 Beta 2 the data items to select are different. The data item numbers for v9.0 are -

- 01 "elapsed times"
- 03 "speed, vertical speed"
- 04 "Mach, G-Load"
- 13 ""trim/flap etc"
- 14 "gear/brakes"
- 18 "pitch, roll etc"

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 9 beta 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.