AEROSOFT

Large Flight Simulator Operator Manual

Author: D J Allerton
Version: 1.4
Date 21st May 2017

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Purpose of this Manual

This manual describes the operation of the large flight simulator (LFS), based in Building 83 at Cranfield University, from the perspective of an academic member of the teaching staff, a laboratory demonstrator or a student. It includes brief descriptions of the flight simulator controls and settings to enable the flight simulator to be operated. It provides essential information to operate the flight simulator safely and effectively.

This information includes:

- The procedures to switch the flight simulator on and off
- Descriptions of the major systems in the flight simulator
- Descriptions of the aircraft systems to operate the flight simulator
- Operation of the instructor station
- Guidelines for preventative maintenance and fault finding.

1.2 Overview

In general, this manual is intended as a reference manual to enable an academic member of the teaching staff to operate the flight simulator.

Section 1 outlines purpose of this manual, the aims of an engineering flight simulator and the contents of this manual.

Section 2 covers the computer systems in the flight simulator, the overall structure of the flight simulator, including the network protocol and interconnections (cabling) and the functions of the specific systems.

Section 3 provides an overview of the cockpit systems to enable the simulator to be operated. The aircraft systems simulated in the flight simulator, including the aircraft flight instruments and the flight controls are explained.

Section 4 describes the operational aspects of the LFS to enable a university engineering department to operate the flight simulator safely, to start up the flight simulator and to shut it down.

Section 5 is a review of the software used in the flight simulator. It outlines the software used in the different computer systems and the organisation of code and data files.

Section 6 describes the instructor station including use of the mouse and menus, describing the method of execution and the effect of each command.

Section 7 describes the menu commands provided by the instructor station, in detail. Each command is described on a separate page in a standard format, explaining the method of selection and the effect of each command. An explanatory example is provided for each command, including a list of related commands.

Section 8 covers the interfacing of Matlab to the flight simulator, from the perspective of both a Matlab user and also a developer needing to modify the interface.

Section 9 outlines the maintenance aspects for both regular preventative maintenance and also calibration and adjustment.

Section 10 is a summary of the diagnostic software provided to ensure the flight simulator is operating correctly or to detect faults or failure.

1.3 Aims of the Large Flight Simulator

The large flight simulator (LFS) is an engineering flight simulator. It is not intended to be used as a flight training device and is not qualified for any aspect of flight training courses. Its primary purpose is to demonstrate aerodynamics, flight mechanics, avionics, flight dynamics, flight control system design, aircraft stability and handling and real-time software design to postgraduate students. The simulator also provides a platform to develop and validate prototype designs of aircraft systems. A number of aircraft types are simulated and piloted flight tests can be undertaken to capture and analyse flight data.

The flight simulator is similar to engineering flight simulators used in the aerospace industry, enabling designers to develop and test prototype designs before they are flown in an aircraft. The LFS is designed on a modular basis, to enable hardware and software modules to be interchanged or modified and enables students to develop and test their software in a real-time pilot-in-the-loop environment.

2 SYSTEM ARCHITECTURE

2.1 Systems

The LFS comprises:

1. The flight deck of the LFS is based on a two-seat Boeing 747-100 cockpit procedures trainer, originally manufactured by Burtek and operated by British Airways. In addition to two centre columns, there are two Airbus side-sticks. The centre pedestal includes, flap selection, trim wheel, engine levers, fuel selection, speed brake, rudder trim and a Radio Management Panel (RMP). There is a Flight Control unit (FCU) below the glare-shield. The overhead panel includes switches for engine start and handles for fire suppression. A gear selection lever is forward of the centre pedestal. Note that the FCU and RMP are based on Airbus units rather than Boeing units.

2. The computer systems: ten computers provide a flight dynamic model, an engine model, aircraft displays, navigation and avionics, an instructor station, interfacing and a real-time visual display. The computers are inter-connected by a dedicated Ethernet network.

3. Aircraft instrument displays: real-time computer graphics are used to display aircraft flight instruments including navigation and engine instruments for EFIS and EICAS displays. The displays are updated by the flight model computer, the engine model computer and the navigation computers.

4. A sound system to generate typical aircraft sounds including jet engines, warnings, background noise and idents.

5. An instructor station: this PC enables the instructor to monitor and control a session by means of a mouse and interactive menus. The instructor station is also used for flight data recording.

6. A real-time Image Generation (IG) system which produces imagery for the three projectors. The image generation system can also overlay HUD formats.

7. Three projectors which display images on a wide curved mirror with a field of view 180° by 40°, typically to show airfield scenes.

8. A dedicated PC to run Matlab software which is straightforward to interface to the flight simulator.

9. Two dedicated I/O systems interfaced to the cockpit control, levers, switches and selectors, to provide analogue inputs, digital inputs and digital outputs.

2.2 Organisation

The organisation of the computer systems is shown in Figure 2.2.

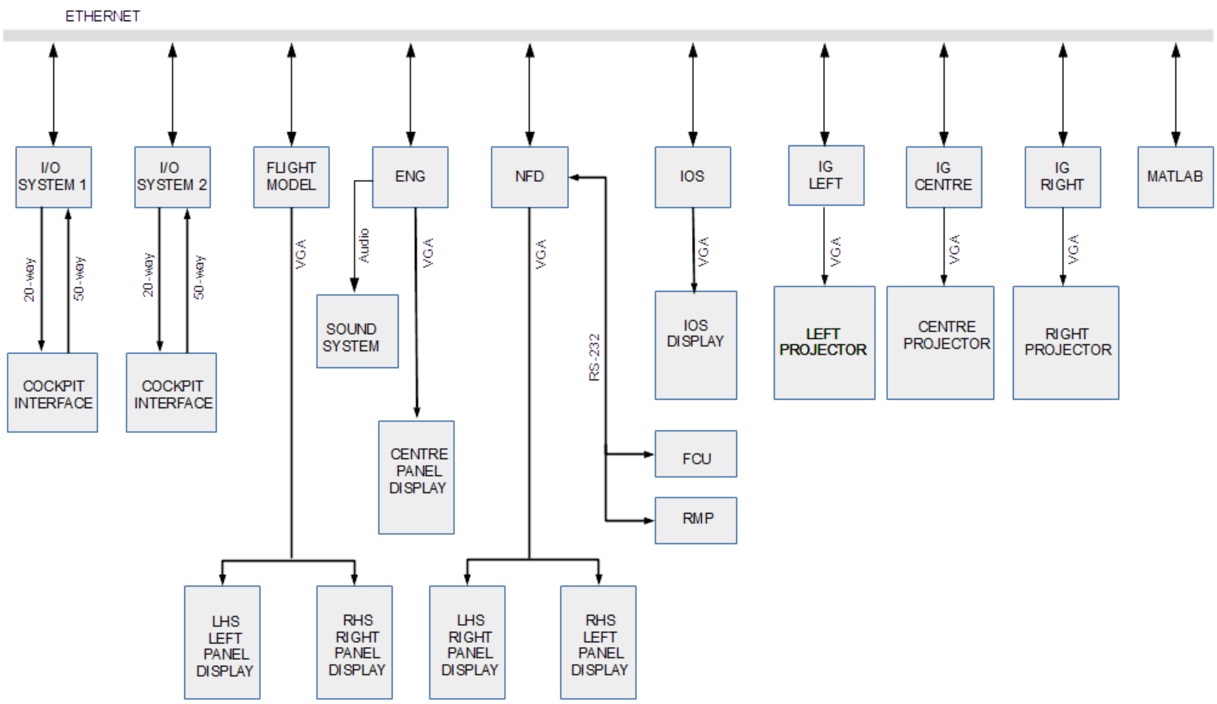


Figure 2.2 Organisation of the Flight Simulator

2.3 Network Protocol

The ten computers are connected via a 16-port Ethernet switch to a dedicated local Ethernet network. A protocol based on broadcast UDPs is used for communication between computers which are numbered as follows:

1. I/O system 1
2. I/O system 2
3. PFD computer
4. ENG computer
5. NFD computer
6. IOS computer
7. IG computer (left channel)
8. IG computer (centre channel)
9. IG Computer (right channel)
10. Matlab computer

The numbers 1-10 are referred to as node numbers and identify the individual computers.

If the Matlab computer is switched on and is enabled, it also executes the protocol described below. Otherwise it waits to detect a packet from the IOS computer (6) that it has been enabled. The following protocol outlines the packet transfers (assuming the Matlab computer is enabled).

At the start of every 20ms (50Hz) frame, the transfers are as follows:

1. I/O system 1 (node 1) broadcasts a packet when 20 ms has elapsed since its previous broadcast and waits for packets from nodes 2, 3, 4, 5, 6 and 10. It then captures analogue and digital inputs and generates digital outputs.
2. When the packet from node 1 is detected by the I/O system 2 (node 2), it broadcasts a packet and waits for packets from nodes 3, 4, 5, 6 and 10. It then captures analogue and digital inputs and generates digital outputs.
3. When the packets from nodes 1 and 2 are detected by the flight model computer (node 3), it broadcasts a packet and waits for packets from nodes 4, 5, 6 and 10. It then computes the flight model software and updates the primary flight display (PFD).
4. When the packet from nodes 1, 2 and 3 are detected by the engine model computer (node 4), it broadcasts a packet and waits for packets from nodes 5, 6 and 10. It then computes the engine model and the sound generation software and updates the EICAS display.
5. When the packets from nodes 1, 2, 3 and 4 are detected by the navigation and avionics computer (node 5), it broadcasts a packet and waits for packets from nodes 6 and 10. It then computes the navigation and avionics software and updates the navigation flight display (NFD).
6. When the packets from nodes 1, 2, 3, 4 and 5 are detected by the instructor station (IOS) computer (node 6), it broadcasts a packet and waits for a packet from node 10. It then responds to user inputs at the IOS (if any), computes the IOS software and updates the IOS display.
7. The three IG computers are passive and detect incoming UDP packets from nodes 1, 2, 3, 4, 5, 6 and 10 before executing the IG software, generating images for the projectors.
8. When the packets from nodes 1, 2, 3, 4, 5, and 6 are detected by the Matlab computer, it broadcasts a packet. It then computes the Matlab software.

The transfers are illustrated in Figure 2.3 (not to scale) with the Matlab transfers omitted.

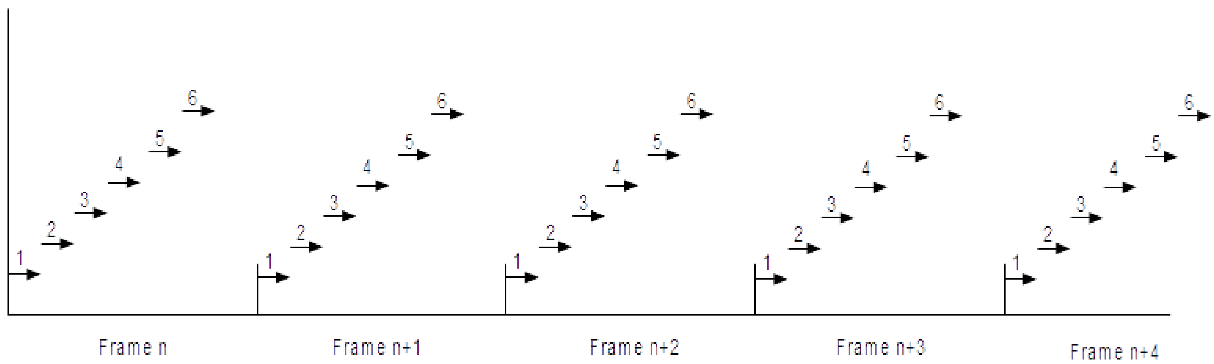


Figure 2.3 UDP Transfers

The three cockpit displays (1024x768 resolution) are driven by VGA outputs from the flight model computer, the engine model computer and the navigation and avionics computer, respectively. The IOS display (1024x768 resolution) includes a mouse and is used to control and monitor simulator sessions. The VGA outputs of the three IG computers drive the three projectors.

For the two I/O systems, the cockpit signals are connected to a breakout card which is connected to the I/O-system via a 50-way ribbon cable for analogue and digital inputs and a 20-way ribbon cable for digital outputs, as shown in Figure 2.4. The connections for the two connectors are defined in Section 10.3. The two I/O systems are headless, meaning that they execute the simulator software automatically on start-up and require no user interaction and do not display any information.

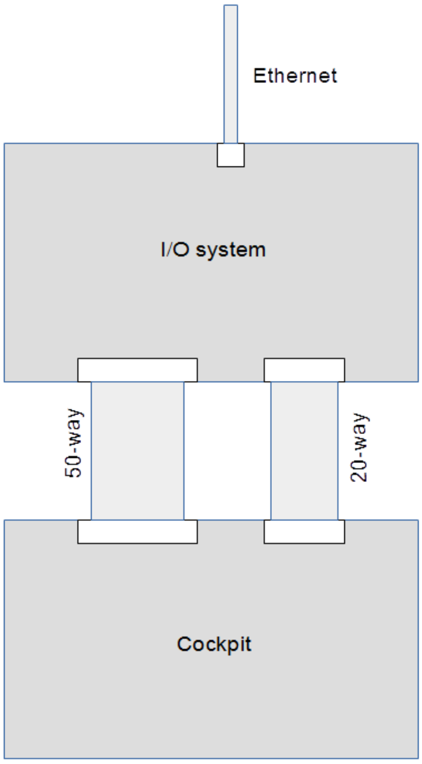


Figure 2.4 I/O System Connections

2.4 System Functions

The I/O system, shown in Figure 2.4, samples analogue and digital data from the cockpit controls, knobs, switches and buttons at 50 Hz. The analogue data is sampled with a resolution of 12 bits. In addition, the I/O system generates 16-bit digital outputs for the cockpit motors, solenoids and lamps. The I/O system is based on a dedicated Raspberry Pi Model 3 running the Debian Raspian Linux operating system and contains a custom PCB for data acquisition using I2C devices. The I/O system includes a green LED indicator for power and produces a pattern on 8 LEDs to show that the flight simulator is running. The analogue and digital channels are defined in Section 10.3.

The flight model computer solves the equations of motion and computes the aerodynamic model and the undercarriage model at 50Hz. The models are generally complex non-linear models of aircraft dynamics. In addition, the flight model computer renders the left hand EFIS display at a resolution of 1024x768 at 50 Hz.

The engine model computer solves the equations for a Pratt and Whitney JT9D engine to compute the thrust, fuel flow and exhaust gas temperate (EGT) for each engine. The engine model computer renders the centre EICAS display at a resolution of 1024x768 at 50 Hz. In addition, the sound card is programmed to provide a range of aircraft sounds and is connected to an audio amplifier and speaker system.

The navigation and avionics computer solves the navigation equations for a wide range of aircraft sensors and reads a database of navigation beacons (ADF, VOR, DME and ILS) and runways at the start of the simulation. This computer also updates the right hand display at 50 Hz. In addition, the navigation and avionics computer is interfaced to the Flight Control Unit (FCU) and the Radio Management Panel (RMP) via RS-232C serial lines.

The IOS computer enables the instructor to monitor a simulator session, using a map display, approach display or raw data display. The IOS computer provides a graphical user interface (GUI) to control the simulation, which is described in Section 7. In addition, the IOS provides a data recording facility to record, store and replay flight data in real-time. The data recording can also be viewed off-line.

The three image generators (IGs) provide a pilot's-eye view of an outside world at 50Hz. A visual database is loaded at the start of the simulation to provide realistic scenery containing fields, hills, airfields, sea and aircraft. In addition, a 2D HUD can be overlaid on the 3D scene and the instructor can set environmental effects such as visibility, cloud base and time of day.

The Matlab computer enables a user to develop control algorithms in Matlab which can be interfaced to the flight simulator for testing. Wrappers are provided to simplify the interface between the simulator and the Matlab PC. A set of templates are provided for users as simple examples to acquire aircraft data and to set aircraft flight controls. An important requirement is that the Matlab implementation must not exceed the 20ms frame time of the simulator; if it does, these delays will propagate to the other PCs, reducing the real-time performance (50 Hz update rate) of the flight simulator.

3 COCKPIT SYSTEMS

The flight deck, shown in Figure 3.1, is based on a Boeing 747-100 cockpit procedures trainer (originally operated by British Airways) but with Boeing 747-400 flight displays. Pilot notes for the Boeing 747, e.g. http://www.meriweather.com/flightdeck/747/deck-747.html explain the operation and functions of the cockpit systems in detail.

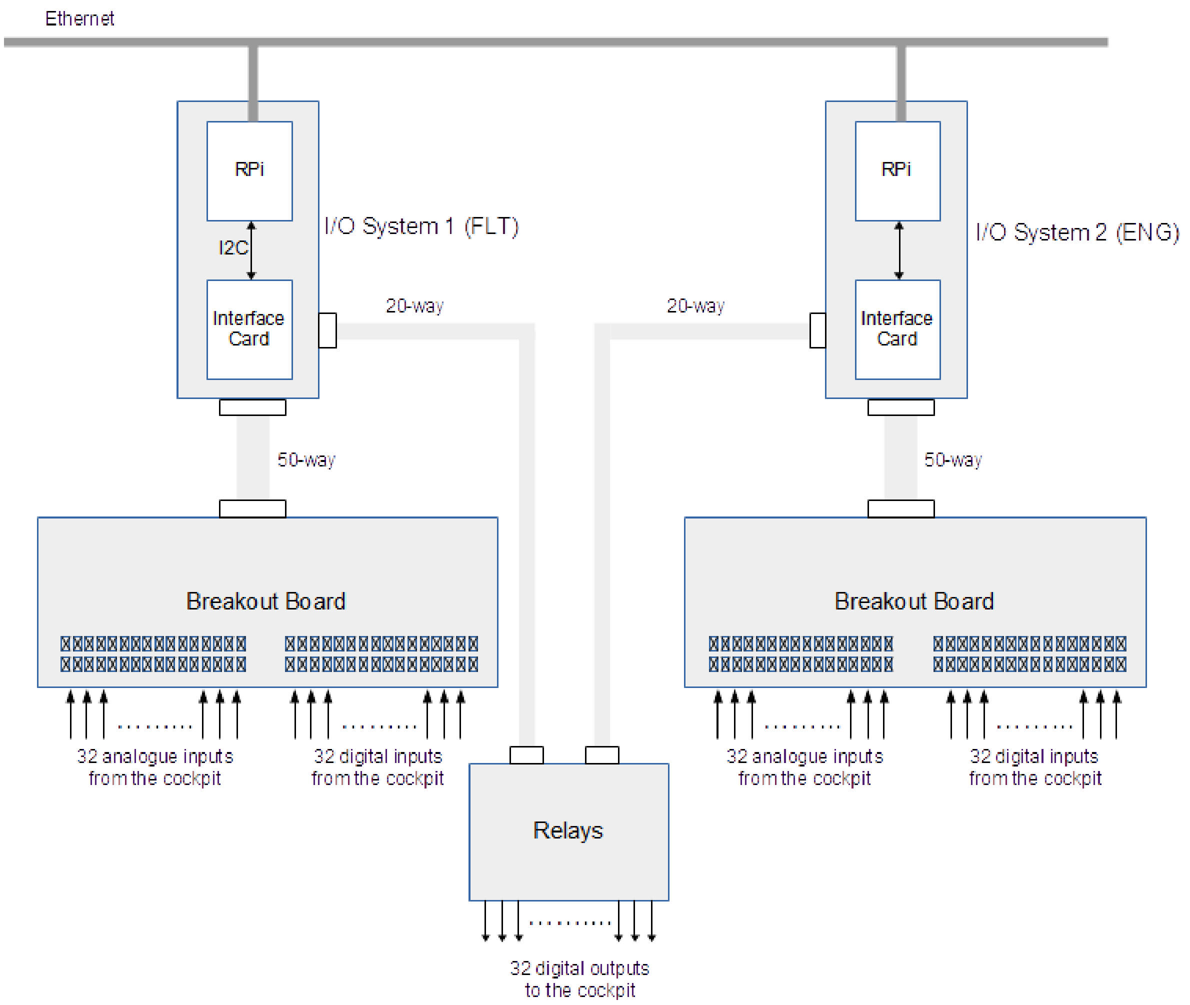
In addition, Airbus side-sticks are located at each side of the flight deck. The side-sticks include a PTT switch and a trigger switch to disengage the flight control system.

Figure 3.1 Flight Deck

The two control columns are mechanically connected, with passive control loading provided by a spring-damper mechanism. The left hand column has electrical trim switches to trim the elevator, in addition to the stab trim switches on the centre pedestal. The rudder pedals are also mechanically connected and include toe brakes. The rudder pedals are not adjustable. A nose wheel steering tiller is provided to the side of each pilot seat. The overhead panel provides engine start (both ground and air) and fire suppression pull handles. Fuel selection for four engines is located in the centre pedestal. Additional controls are provided for flap selection, gear selection, park brake and rudder trim. An Airbus A320 flight control unit (FCU) is located below the glare shield and a single Airbus A320 Radio Management Panel (RMP) is located to the left hand side of the centre pedestal. Currently, the TCAS/XPDR transponder unit and audio control panels are non-operational.

Note that the elevator trim buttons and the stab trim levers drive the trim motor, whereas the rudder and aileron trims are passive inputs.
The right hand side EFIS displays (PFD and NFD) are replicated video signals of the left hand displays and consequently, the right hand settings of the FCU are not used.

The knobs, levers and switches are connected to two I/O systems, which acquire inputs for the simulator computers and produce outputs for motors, solenoids and lamps. Cables from the cockpit systems are connected to the breakout cards shown in Figure 3.2. This card provides calibration of analogue signals, input protection and interfacing to the cockpit controls, switches and levers. The analogue input ranges are based on nominal 10K linear potentiometers. Of the 32 analogue input channels, 16 are provided with adjustment for gain and offset and the other 16 analogue inputs are used for input within the range of 0-5V. The I/O systems also generate digital outputs for motors, solenoids and lamps in the cockpit. As the I/O system has a low current output rating, these output signals drive relays to provide the higher current needed for these cockpit items.



<u>Figure 3.2 I/O Systems</u>

The small panel of three buttons in the overhead panel is shown in Figure 3.3. The right red button is a HOLD button, which is used to suspend the simulation. When pressed, the flight model, engine model, navigation computations and sound system are suspended, although the IOS still responds to user commands. The middle yellow button is a momentary RESET button and will reset the simulator to the last restored state (if any). The left green button is a FREEZE button. When this button is pressed, the simulator will update but the aircraft position does not change, for example to set up trim conditions. These switches illuminate when pressed. Note that the centre yellow button is also pressed to start the simulation.

Figure 3.3 Hold, Reset and Freeze Buttons

The two seats have levers to move the seats fore and aft (along tracks) and up and down. It is important that the seat is positioned correctly in terms of pilot eye height and access to the rudder and toe brake pedals at the start of a simulator session. The use of the seat harnesses is optional and is left to the discretion of the supervisor.

The FMS CDU, TCAS/XPDR panel and audio panel are currently non-operational.

4 OPERATION

4.1 Installation

The simulator is installed so that the projection screen and projectors are correctly aligned. On no account should these items be adjusted.

There are numerous cables and connectors and care is needed working close to these to ensure that cables are not damaged or connectors removed.

4.2 Operating Environment

The LFS incorporates ten computer systems and should be operated in accordance with standard office practice for computer systems, in terms of temperature variation, humidity, electrical supply and cleanliness.

Careful examination of the flight simulator prior to switching on and observation of the electrical and environmental requirements are essential.

The flight simulator operates in a standard office environment from a single 32 amp supply and several 13 amp mains sockets. The immediate area must have an adequate stable power source free from power surges and voltage 'spikes'. The electrical systems must be 'earthed' at all times.

Electrical Supply	240V 50Hz AC
Temperature	+10° C to +35° C
Humidity	to 90% relative humidity (over the whole temperature range, non-condensing)

4.3 Safety

The 1974 Health and Safety at Work Act requires the appointment of a site safety officer, who is legally responsible for all aspects of safety. This responsibility also extends to managers and supervisors and requires the correct operation of the flight simulator under the statutory provisions of the Act as well as the Operator's written Statement of Safety Policy, which must include the specific points given below, together with any additional requirements.

- All users must be trained to observe all safety requirements before entering the flight simulator.
- On detection of any fumes, smoke or hazardous conditions, all personnel must exit the flight simulator immediately and report the situation. Electrical power should be turned off, provided this does not affect the safe evacuation.
- If an electrical shock is experienced, the mains supply must be turned off immediately.
- Fire extinguishers, which are suitable for use on electric fires, must be placed adjacent to the flight simulator and all personnel must be instructed in their use.
- Smoking is prohibited inside the flight simulation room.

- No liquids or hazardous materials may be taken into the flight simulator room.
- No modification, alteration or maintenance of the flight simulator may be made without the prior approval of the site safety officer. All work must be carried out by approved and qualified personnel.
- Periodic maintenance schedules must be established to test the electrical safety of the flight simulator.
- No food or drink may be taken into the flight simulator room.

Only authorised staff are permitted to access the platform housing the projectors, remove equipment, access underfloor wiring or pass beyond the bench where the computers are mounted.

4.4 Start-Up Procedure

The flight simulator power should be switched on in the following sequence:

Switch on the light switch by the simulator door as shown in Figure 4.1 and the enclosure light switch at the back of the simulator, if required, as shown in Figure 4.2.

Figure 4.1 Entrance Light

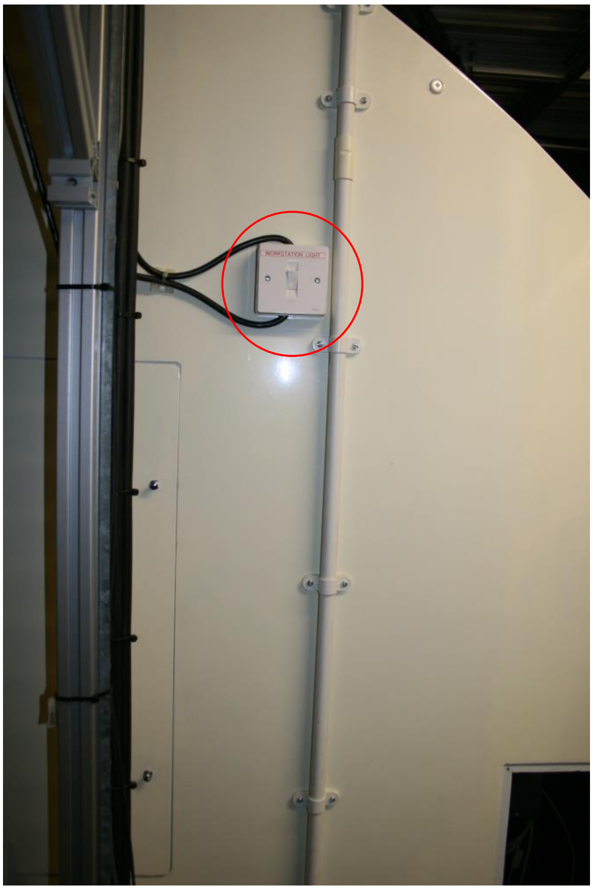


Figure 4.2 Enclosure Light

Ensure that the mains switch on both I/O systems is turned on, as shown in Figure 4.3.

This precaution is essential because these I/O systems will switch on in a state that ensures all digital outputs, particularly motor drives, are switched off.

Figure 4.3 I/O Systems Mains Switch

Switch on the power supply below the bench as shown in Figure 4.4.

Figure 4.4 Power Supply Unit

CAUTION: If any faults are evident, immediately switch off the mains supply.

The computer systems in the LFS are switched on by pressing the power switch on the front panel of each PC, waiting until the display for each PC shows the standard Windows10 desktop.

To simplify operation of the flight simulator, an ATEN KVM switch is provided to switch the instructor station monitor to any of five computers. The mouse/keyboard USB cable and the VGA cable of each PC is connected to the KVM switch as shown in Figure 4.5.

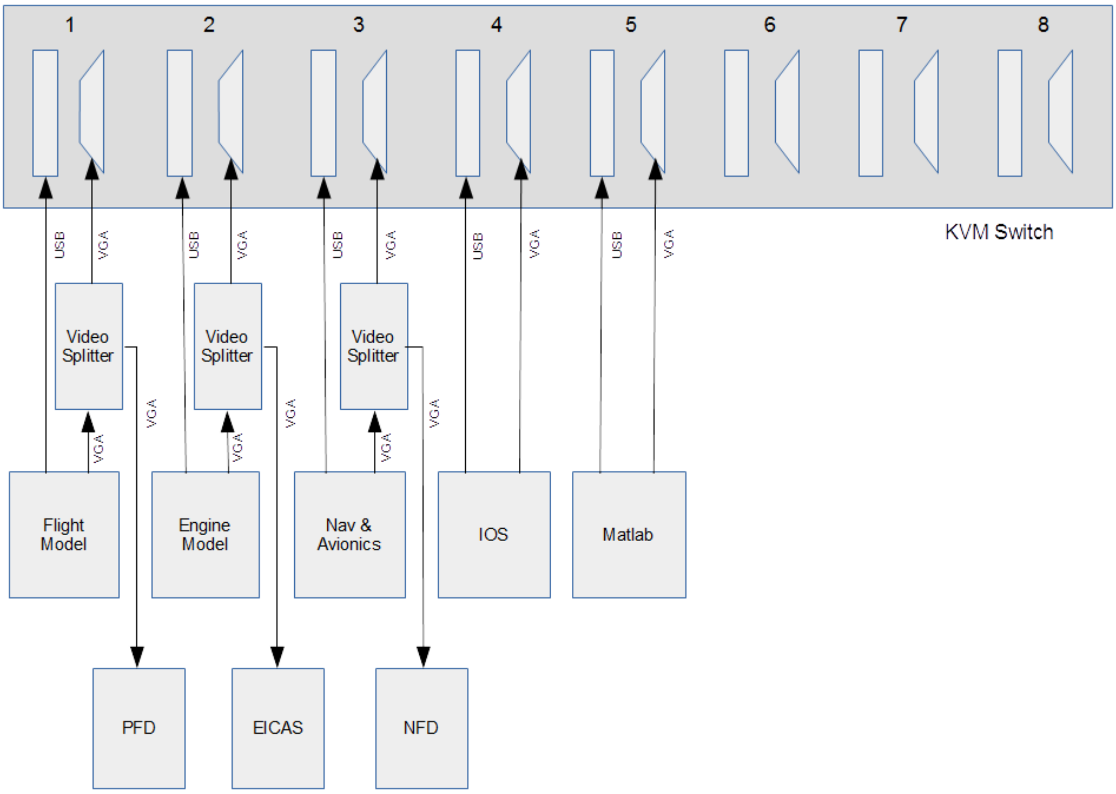


Figure 4.5 KVM Switch Connections

Initially, the KVM switch will prompt for a user name and password. Press enter for each prompt and then select the appropriate channel 1-5 to connect the keyboard, mouse and monitor to the any of the five PCs. The respective channel numbers are indicated at the top of Figure 4.5. Channels can be switched at any time by pressing one of the front panel buttons of the ATEN KVM switch shown in Figure 4.6. The selected channel is illuminated on the front panel of the KVM switch. Alternatively, hold down the Num Lock key and then press and release the minus key on the numeric keypad. When prompted for a channel number, enter the number using the alphanumeric key pad (not the numeric keypad) followed by the ENTER key. Currently, channels 6, 7 and 8 are unused.

Figure 4.6 KVM Switch

The KVM channel numbers are as follows:

1. PFD
2. ENG
3. NFD
4. IOS
5. Matlab

In addition, the three IG systems are located in the Simulator Laboratory. Three monitors are provided to display the respective IG channels. A mouse/eyboard switch is provided to connect the mouse and keyboard to any of the three IG PCs.

The two I/O systems are 'headless' – that is to say they run autonomously. They should not need to be switched on or off and they will detect the necessary conditions to start the simulation. They will also restart autonomously, following termination of a simulator session. The LED indicator, shown in Figure 4.8, will show an alternating pattern (changing twice per second) when the I/O system is waiting to start and a faster rotating pattern, to indicate that the simulator is running.

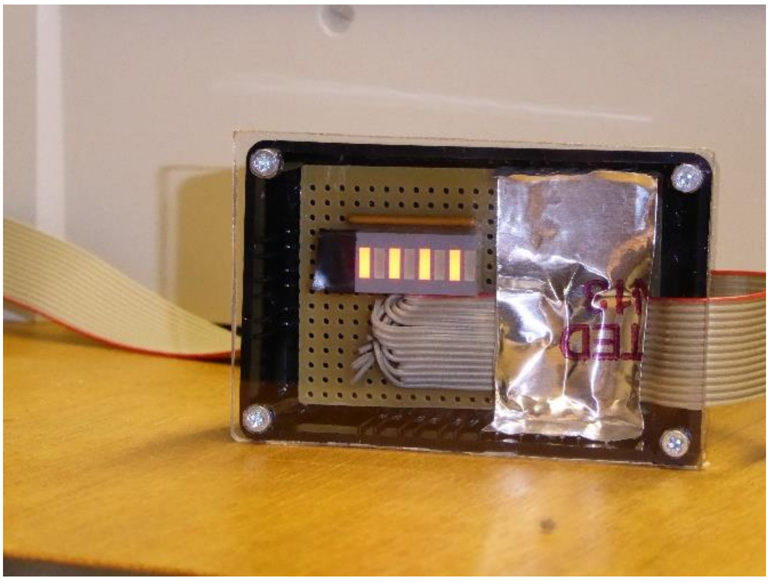


Figure 4.8 LED Indicator

The five PCs can be started in any order. Each PC has a MinGw prompt on the desktop which is used to start the simulation. For example, to run the Boeing 747 simulation with the Bristol visual database, click on the MinGw icon for each PC to open a terminal and enter the following sequence of commands:

Flight model PC (3)
```
sim>cd /c/sim/pfd/b747
sim>./b747
```

Engine model PC (4)
```
sim>cd /c/sim/eng/b747
sim>./b747
```

Navigation and avionics PC (5)
```
sim>cd /c/sim/nfd/b747
sim>./b747
```

IOS PC (6)
```
sim>cd /c/user
sim>ios
```

Left IG (Simulator Laboratory)
```
sim>cd /c/ig/visual
sim>./cgi Bristol
```

Centre IG (Simulator Laboratory)
```
sim>cd /c/ig/visual
sim>./cgi Bristol
```

Right IG (Simulator Laboratory)
```
sim>cd /c/ig/visual
sim>./cgi Bristol
```


If the MATLAB PC is to be used, it should be switched on and Matlab started at his time. The execution of a Matlab program is described in Section 8.

In the example above, the argument *b747* denotes the Boeing 747-100 flight model. Alternatively, the Airbus A340 flight model can be loaded by changing the file name to *a340*. Initially, the navigation database is loaded, which takes a few seconds. The directory *user* (shown for the IOS PC above) denotes any user directory. It is recommended that each user operates from their own directory as saved files, printed files and other user-specific files are, by default, stored in the current operating directory.

Once the five PCs are started, the simulation is started by pressing the yellow RESET button on the overhead panel. The five flat screen LCD panels should display aircraft instruments and the IOS should respond to commands.

If the simulator is in the HOLD state, remember to release the HOLD button in the cockpit for the simulator to run.

A simulator exercise should only be terminated by executing an exit command at the instructor station. If a Windows ctrl-C sequence is entered on any PC, the remaining PCs and the I/O system will be left in an undetermined state and will need to be reset to the initial conditions.

4.5 Projectors

The final step is to turn on the projectors using the projector control unit shown in Figure 4.5.

Figure 4.5 Projector Control Unit

The projectors can be turned on at any time during the start-up procedure although the recommended practice is to turn on the projectors once the simulator computers are running. Turn on the switch on the back of the projector control unit (PCU). There is a delay of approximately one minute while the DDIU performs power-up self-test and system checks. The silver buttons are used to select projector functions. The thumb wheels are used to highlight menu options. The red buttons are used to either confirm or cancel a selection.

Press any silver key. The PCU asks if the DDIU is ready and displays a blank menu. Press the UTILS button. Use the left thumb wheel to highlight *Select Projector Control* and the press the RED button to confirm the selection. Use the left thumb wheel to select *Start All Projectors*. Press the RED button to confirm the selection. When the projectors are ready, the projector control unit will display *Start All Projectors*. Press the BEAMS button, select *Global beams* using the thumbwheel and press the button marked RGB to display the images.

Note:

1. There is no need to shut down the projectors if the simulator is stopped and restarted.
2. The projectors take at least 30 minutes to reach stable operation, and during this time the projectors will not display optimal image quality, particularly in terms of colour convergence and geometric accuracy. This is normal for CRT projectors.

4.6 Shut-Down Procedure

No damage can occur to the flight simulator during shut down. The effect of simply turning the power off is the same as a power failure for a PC. However, it is strongly recommended to shut down the operating systems cleanly, particularly to avoid possible data corruption to the file store. In particular, the projectors should be shut down to maximise the working life of the projector bulbs.

Prior to shutting down, it is recommended that the simulator is left in the HOLD mode, with the throttles in the idle position, the park brake on, engine switches off and flaps up.

The recommended shut down procedure is as follows:

1. Execute the EXIT command in the instructor station – all PC systems should revert to the Windows10 desktop.
2. Using the project control unit, use the left thumb wheel to highlight *All to Standby* and press the RED button to confirm the selection, shutting down the projectors.
3. For each PC, click on the Windows icon in the bottom left of the screen and shut down the PC.
4. Switch off the power supply unit below the bench.
5. Switch off the two mains switches at the back of the simulator.
6. Switch off all lights.
7. Exit the simulator, ensuring that the simulator door is closed and locked.

Finally it is recommended that the projectors are switched off using the push button switches on the back of the three projectors, in order to avoid the projectors rebooting after a power cut.

5 SOFTWARE ORGANISATION

The software for the flight simulator is written in standard ANSI C using the gcc compiler, to ensure a high degree of compatibility between Windows and Linux versions. Currently, The eight PCs run the Windows10 operating system and the I/O systems run under the Debian Raspian Linux operating system.

The flight simulator is connected to a local Ethernet network and must only be connected to external networks (e.g. the campus network) via a dedicated gateway. The protocol used for real-time message passing depends on access to a dedicated network where it is guaranteed that the only network traffic is generated by the simulator computers.

Each PC and the I/O system has a fixed static IP address, which is used by the simulator software. These addresses are allocated statically and should not be changed:

Computer	Computer Function	IP Address
1	I/O system 1	192.168.1.1
2	I/O system 2	192.168.1.2
3	Flight model	192.168.1.3
4	Engine model	192.168.1.4
5	Navigation and Avionics	192.168.1.5
6	Instructor station	192.168.1.6
7	IG left	192.168.1.7
8	IG centre	192.168.1.8
9	IG right	192.168.1.9
10	Matlab	192.168.1.10

Note that, for each PC, the firewalls are disabled and only Internet Protocol Version 4 (TCP/IPv4) is enabled.

5.1 Header Files

All the simulator computers use a common set of C header files which define the constants, structures, variables and functions used throughout the simulator. It is very important that all versions of these header files are identical on all computers at all times. For the I/O systems, these files are stored in the directory /usr/include/SIM. For the PCs, these files are stored in the directory /c/mingw/include/SIM. The system-wide header files include:

```
-rw-r--r-- 1 Dave Administrators   5473 Nov 21 22:00 aerodefn.h
-rw-r--r-- 1 Dave Administrators    636 Apr  8  2013 clocks.h
-rw-r--r-- 1 Dave Administrators   3052 Nov 21 21:26 engdefn.h
-rw-r--r-- 1 Dave Administrators   1635 Jun  3  2016 glib.h
-rw-r--r-- 1 Dave Administrators   3107 Apr 12  2013 iodefn.h
-rw-r--r-- 1 Dave Administrators   7461 May 29  2016 iosdefn.h
-rw-r--r-- 1 Dave Administrators    702 Apr 11  2011 maths.h
-rw-r--r-- 1 Dave Administrators   4603 Mar 29  2016 navdefn.h
-rw-r--r-- 1 Dave Administrators   2542 May 11  2016 navlib.h
-rw-r--r-- 1 Dave Administrators    121 Oct 16  2015 pnglib.h
-rw-r--r-- 1 Dave Administrators   1832 Aug  5  2011 protodefn.h
-rw-r--r-- 1 Dave Administrators    965 Apr 12  2013 soundlib.h
-rw-r--r-- 1 Dave Administrators    784 Apr  8  2013 target.h
```



```
-rw-r--r-- 1 Dave Administrators    474 Apr 11  2011 texture.h
-rw-r--r-- 1 Dave Administrators   1476 Apr  3  2016 textureid.h
-rw-r--r-- 1 Dave Administrators    317 Apr 11  2011 udplib.h
-rw-r--r-- 1 Dave Administrators   1108 Oct 19  2013 weather.h
```

The flight model computer, the engine model computer, the navigation computer and the IOS computer use OpenGL libraries and access the header files gl.h and glu.h, which are stored in the directory /c/mingw/include/GL and the header file and glfw3.h which is stored in the directory /c/mingw/include/GLFW. The sound system in the engine computer uses the OpenAL libraries and the header file alut.h which is stored in the directory /c/mingw/include/AL. The IG systems use OpenSceneGraph libraries and the header files are stored in /c/mingw/include/osg.

Each system is provided with 'make' files which can be run from a MinGw or Linux terminal, using *make clean* and then *make* commands, to build the software without manual intervention.

5.2 Common Libraries

A set of common libraries are stored and compiled (in the case of the FLT PC) in the directory /c/sim/pfd/libs/:

```
-rw-r--r-- 1   407 Oct 15  2013 Makefile     a 'make' file to recompile the libraries
-rw-r--r-- 1  1906 Aug  5  2011 pnglib.c     png file generation
-rw-r--r-- 1  2912 Oct 15  2013 clocks.c     clock and time management
-rw-r--r-- 1 20894 Oct 15  2013 glib.c       2D graphics functions
-rw-r--r-- 1  1181 Aug  5  2011 maths.c      additional maths functions
-rw-r--r-- 1  5501 Oct 15  2013 target.c     management of targets
-rw-r--r-- 1  6036 Aug  5  2011 texture.c    display textures
-rw-r--r-- 1  4647 Oct  4  2013 udplib.c     UDP transfers
-rw-r--r-- 1 11644 Oct 19  2013 weather.c    weather model
```

5.3 I/O Systems

The I/O systems contain a Model 3 Raspberry Pi with a custom PCB providing I2C input and output. To modify or develop software for the I/O systems, it is necessary to logon to the Raspian system. By default, the I/O system runs as a Linux terminal. However, it is possible to connect an HDMI display and a mouse and keyboard to the Raspberry Pi system and use it with a standard Linux desktop. Alternatively, the I/O system can be connected via an SSH protocol using the *putty* program (or similar) from another PC on the network.

For the RPi, the main directory is /home/pi with two subdirectories sim/rpi1/calibrate and sim/rpi1/iosystem (the second RPi system uses the subdirectory rpi2).

The iosystem directory contains the following source files:

```
-rw-r--r-- 1   426 Oct 10  2014 Makefile     make file to build the system
-rw-r--r-- 1 22862 Aug 27 17:36 iolib.c      library to access the I/O components
-rw-r--r-- 1  2228 Aug 27 16:46 iolib.h      header file for the I/O components
-rw-r--r-- 1  8003 Aug 27 16:34 iolink.c     library for network connection
-rw-r--r-- 1   779 Oct 10  2014 iolink.h     header file for network connection
-rw-r--r-- 1  5091 Aug 27 18:00 iosystem.c   main program for input and output
-rw-r--r-- 1  4449 Oct 10  2014 udplib.c     library for UDP transfers
```


5.4 Flight Model PC

The various flight models are stored in sub-directories of the flight model computer in /c/sim/pfd/ where the directory name corresponds to the aircraft type.

For example, the Boeing 774-100 model is stored in /c/sim/pfd/b747 and comprises the following files:

```
-rw-r--r-- 1   879 Dec 13 15:20 Makefile     make file to build the software
-rw-r--r-- 1  9638 Jan 17 16:27 aero.c       aerodynamic model
-rw-r--r-- 1  2455 Dec 13 16:31 aero.h       header for aero.c
-rw-r--r-- 1 28950 Feb  5 09:36 aerolink.c   network transfers
-rw-r--r-- 1  1290 Feb  5 09:33 aerolink.h   header for aerolink.c
-rw-r--r-- 1 11299 Mar 30  2016 ai.c         attitude indicator display
-rw-r--r-- 1   480 Apr  8  2013 ai.h         header file for a1.c
-rw-r--r-- 1 11847 Jan  6 16:03 alt.c        altimeter display
-rw-r--r-- 1   403 Apr  8  2013 alt.h               header for alt.h
-rw-r--r-- 1  7454 Mar 30  2016 asi.c               airspeed display
-rw-r--r-- 1   367 Apr  8  2013 asi.h               header for asi.c
-rw-r--r-- 1  5651 Feb  5 09:33 b747.c              main program
-rw-r--r-- 1  6417 Mar 30  2016 compass.c           compass display
-rw-r--r-- 1   171 Aug  5  2011 compass.h           header for compass display
-rw-r--r-- 1  1901 Jan 17 14:36 diagnostics.c       graphics diagnostics display
-rw-r--r-- 1   170 Mar 29  2016 diagnostics.h       header for diagnostics.c
-rw-r--r-- 1 16749 Jan 17 16:44 fcs.c               flight control systems
-rw-r--r-- 1  1676 Dec 13 16:38 fcs.h               header for fcs.c
-rw-r--r-- 1 11984 Dec 13 12:14 gear.c              undercarriage model
-rw-r--r-- 1   244 Dec 13 12:12 gear.h              header for gear.c
-rw-r--r-- 1  9182 Feb  3 10:41 iolib.c             input/output functions
-rw-r--r-- 1  2039 Feb  3 12:19 iolib.h             header for iolib.c
-rw-r--r-- 1 18599 Jan 17 16:49 model.c             equations of motion
-rw-r--r-- 1  2565 Dec 13 12:08 model.h             header for model.c
-rw-r--r-- 1  9028 Feb  5 09:35 pfd.c               main display
-rw-r--r-- 1  1176 Aug  5  2011 pfd.h               header for pfd.c
-rw-r--r-- 1  7408 Jan 17 14:39 systems.c           aircraft sub-systems
-rw-r--r-- 1   856 Dec 15 15:59 systems.h           header for systems.c
-rw-r--r-- 1  4308 Apr  1  2016 vsi.c               vertical speed indicator
-rw-r--r-- 1   129 Aug  5  2011 vsi.h               header for vsi.c
```

5.5 Engine Model

The various flight models are stored in sub-directories of the navigation computer in /c/sim/eng/ where the directory name corresponds to the aircraft type.

For example, the Boeing 747-100 engine model is stored in /c/sim/eng/b747 and comprises the following files:

```
-rw-r--r-- 1   863 Jan 17 12:59 Makefile            make file to build the software
-rw-r--r-- 1  5153 Feb  3 12:24 b747.c              main program
-rw-r--r-- 1  1901 Jan 17 15:02 diagnostics.c       graphics diagnostics display
-rw-r--r-- 1   170 Dec 13 23:17 diagnostics.h       header for diagnostics.c
-rw-r--r-- 1  7202 Feb  5 09:39 efd.c               main display
-rw-r--r-- 1   506 Dec 14 09:12 efd.h               header for efd.c
-rw-r--r-- 1  8877 Jan 17 15:04 eicas.c             engine displays
-rw-r--r-- 1   377 Dec 13 23:31 eicas.h             header for eicas.c
-rw-r--r-- 1 10411 Jan 17 15:04 engines.c           engine model
-rw-r--r-- 1  1291 Dec 15 09:34 engines.h           header for engines.c
-rw-r--r-- 1 17944 Feb 12 11:53 englink.c           network transfers
-rw-r--r-- 1  1175 Feb  5 09:37 englink.h           header for englink.c
-rw-r--r-- 1  7556 Feb  3 10:48 iolib.c             input/output functions
-rw-r--r-- 1  1985 Dec 14 09:08 iolib.h             header for iolib.c
-rw-r--r-- 1  4045 Jan 17 15:06 sounds.c            sound generation
-rw-r--r-- 1   101 Dec 15 15:30 sounds.h            header for sounds.c
-rw-r--r-- 1  3696 Jan 17 15:07 systems.c           aircraft sub-systems
-rw-r--r-- 1   471 Dec 14 10:43 systems.h           header for systems.c
```


5.6 Navigation and Avionics PC

The various flight models are stored in sub-directories of the navigation computer in /c/sim/nfd/ where the directory name corresponds to the aircraft type.

For example, the Boeing 747-100 flight model is stored in /c/sim/nfd/b747 and comprises the following files:

```
-rw-r--r-- 1    874 Mar 28  2013 Makefile          make file to build the system
drwxr-xr-x 2      0 Jul 21 11:43 Textures          directory of texture files
-rwxr-xr-x 1  32768 Mar 28  2013 alut.dll          Open-AL library
-rw-r--r-- 1   3273 Aug 21 14:34 b747.c            main program
-rw-r--r-- 1  22336 Apr  9  2013 compass.c         compass display
-rw-r--r-- 1   1160 Apr  9  2013 compass.h         header for compass.c
-rw-r--r-- 1   7376 Aug  5  2011 demo.c            display demonstration program
-rw-r--r-- 1   1901 Jan 17 15:14 diagnostics.c     graphics diagnostics display
-rw-r--r-- 1    170 Jan  9 13:07 diagnostics.h     header for diagnostics.c
-rw-r--r-- 1  20604 Apr  9  2013 fcu.c             flight control unit management
-rw-r--r-- 1   1207 Apr  9  2013 fcu.h             header for FCU.C
-rw-r--r-- 1    675 Feb  3 10:52 iolib.c           input/output functions
-rw-r--r-- 1    623 Feb  3 12:28 iolib.h           header for iolib.c
-rw-r--r-- 1  18952 Apr  9  2013 nav.c             navigation algorithms
-rw-r--r-- 1   1766 Apr  9  2013 nav.h             header for nav.c
-rw-r--r-- 1  10386 Jan  9 18:17 navinfo.c         display information
-rw-r--r-- 1    447 Jan  9 13:09 navinfo.h         header for navinfo.c
-rw-r--r-- 1  16444 Aug 25 09:46 navlink.c         network transfers
-rw-r--r-- 1    945 Apr  9  2013 navlink.h         header for navlink.c
-rw-r--r-- 1   6751 Apr  9  2013 nfd.c             displays main program
-rw-r--r-- 1   1237 Apr  9  2013 nfd.h             header for nfd.c
-rw-r--r-- 1  20746 Apr  9  2013 radio.c           radio management
-rw-r--r-- 1    402 Apr 11  2011 radio.h           header for radio.c
-rw-r--r-- 1    925 Apr  9  2013 systems.c         aircraft sub-systems
-rw-r--r-- 1    587 Apr  9  2013 systems.h         header for systems.c
-rw-r--r-- 1    255 Apr 11  2011 textureid.h       header for texture files
```

5.7 Instructor Station PC

The IOS software is stored in the /c/sim/ios/ directory. There are three sub-directories: *ios* for the IOS source files, *libs* for the libraries used by the IOS and *files* for data files.

5.7.1 Data Files

A number of data files are provided which users are able to modify. However, care is needed in altering these files and it is advisable that the simulator manager is responsible for any changes and that back-up copies of these files are retained.

The following data files are stored in the directory /c/sim/ios/files

```
-rw-r--r-- 1   3046790 Oct 15 09:17 Airports.txt    airfields/runways database
-rw-r--r-- 1    912130 Oct 15 09:17 Navaids.txt     navaids database
-rw-r--r-- 1      5619 Jul 28 12:10 menu.dat        the IOS menu table
-rw-r--r-- 1        49 Apr  9  2012 models.lst      list of flight models
-rw-r--r-- 1        80 Mar 30  2012 targets.lst     list of aircraft targets
-rw-r--r—1          41 Apr  5  2012 visual.lst      list of visual system databases
```

By default, the simulator selects the UK navaids and airfields from the respective databases. The list of flight models corresponds to the currently compiled flight models: B747 and A340. The list of target aircraft corresponds to the OpenFlight models used in the IGs, currently A10, B737, B747, F117, Dash8, Spitfire, F16, Commanche and Grob. The list of visual systems corresponds to the OpenFlight databases, currently Bristol, Hong_Kong and Monterey.

5.7.2 IOS Menu

The menu structure is defined in the file /c/sim/ios/files/menu.dat. In addition, the numbering of IOS commands is defined in the header file iosdefn.h. If any changes are made to these files, it is essential that the correct numbering is maintained.

The file menu.dat is accessed when the instructor station is started. It is possible for a user to edit this file to change default settings, such as the initial cloud base or magnetic variation. However, it must be stressed that inadvertent editing of this file can lead to undesired effects in the operation of the instructor station and that care should be taken to follow the guidelines set out below.

The current version of the file menu.dat is as follows:

```
; Instructor Station Menu Version 5.1 08 Oct 2015

Master,Reposition,Map,Weather,Settings,Failures,Autopilot,Flight Plan,Target,Approach,Data
Recording,Flight Data,Camera

Exit\q               31 <OK to EXIT?>
Display\o            36 <Map Approach Flightdata Rawdata>
Mode\o              124 <Run Hold Freeze>
Print\f              32 <File>
Model\l              33
Visual\l             41
Matlab\o             38 <Off On>

Restore\l            34
Save\f               35 <File>
Position\c           61
Altitude\n          121 <Aircraft Altitude (ft)> 0 50000 3000
Heading\n           122 <Aircraft Heading> 1 360 360
Airspeed\n          123 <Aircraft Airspeed> 0 500 150

Centre\c             62
Compass\c            63
Find\x               64
Track\c              65
Scale\n              66 <Map Scale> 5 5000 250
Reset\q              67 <OK to reset map?>
Zoom/Pan\x           68

Turbulence\n         91 <Turbulence> 0 1 0
Wind Speed\n         92 <Wind Speed (kt)> 0 80 0
Wind Dir\n           93 <Wind Direction> 1 360 360
QNH\n                94 <Area QNH> 995 1035 1013
Magnetic Var.\n      95 <Magnetic Variation> -20 20 -5
Cloud Base\n         96 <Cloud Base (ft)> 0 30000 30000
Visibility\n         97 <Visibility (m)> 0 150000 50000
OAT\n                98 <OAT> -20 30 15
Time of Day\n        99 <Time> 0 24 12
Vis rate\n          101 <Vis rate (m/s)> -10000 10000 0

RMI Card\o          125 <Moving Fixed>
Morse\o             126 <Off ILS1 VOR1 VOR2 ADF1 ADF2>
Engines\o           127 <Multiple Single>
CG Position\n       128 <CG Position (%)> 0 100 25
Flight Controls\o   132 <Column SideStick>
HUD\o               191 <Off On>

Flaps\o             151 <OK Failed>
Gear\o              152 <OK Failed>
NAV1 Loc\o          153 <OK Failed>
NAV1 G/S\o          154 <OK Failed>
```



```
NAV2 Loc\o          155 <OK Failed>
NAV2 G/S\o          156 <OK Failed>
RMI 1\o             157 <OK Failed>
RMI 2\o             158 <OK Failed>
DME\o               159 <OK Failed>
Engine 1\o          160 <OK Failed>
Engine 2\o          161 <OK Failed>
Engine 3\o          162 <OK Failed>
Engine 4\o          163 <OK Failed>
ASI\o               164 <OK Failed>
AI\o                165 <OK Failed>
VSI\o               166 <OK Failed>
Altimeter\o         167 <OK Failed>
Turn\o              168 <OK Failed>
HSI Card\o          169 <OK Failed>
RMI Card\o          170 <OK Failed>
Engine 1 Fire\q     171 <Engine 1 Fire>
Engine 2 Fire\q     172 <Engine 2 Fire>
Engine 3 Fire\q     173 <Engine 3 Fire>
Engine 4 Fire\q     174 <Engine 4 Fire>

A/P ALT\n           133 <Altitude (ft)> 100 40000 3000
A/P HDG\n           134 <Heading> 1 360 360
A/P SPD\n           135 <Speed (kt)> 80 400 150
A/P VSPD\n          136 <Vertical Speed (fpm)> -1000 1000 0
Autoland\o          141 <DisEngage Engage>
A/P ALT Select\o    137 <DisEngage Engage>
A/P HDG Select\o    138 <DisEngage Engage>
A/P SPD Select\o    139 <DisEngage Engage>
A/P VSPD Select\o   140 <DisEngage Engage>

Load Flight Plan\l   39
Flight Plan\o        40 <Disengage Engage>
Script\l             37

Target File\l       181
Target Off\q        182 <Target Off?>
Position\c          183
Distance\n          184 <Distance (m)> -32000 32000 100
Airspeed\n          185 <Airspeed (kt)> 0 800 150
Heading\n           186 <Heading> 1 360 360
Turn Rate\n         187 <Turn Rate (deg/sec)> -30 30 0
Altitude\n          188 <Altitude (ft)> 0 40000 1000
Climb Rate\n        189 <Climb Rate (ft/min)> -5000 5000 0
Conflict\q          192 <Conflict?>

Approach Range\n    214 <Range (nm)> 1 20 10
Reset\q             215 <OK to reset approach?>
G/S angle\n         142 <Glide Slope> 0 20 15>

Record\o            241 <Stop Start Continue>
Save\q              242 <OK to save?>
Next Page\x         243
Previous Page\x     244
Next Mark\x         245
Previous Mark\x     246
Mark\x              247
Goto\n              248 <Secs> 0 300 0
Time\n              249 <Secs> 2 1000 10
Kp\n                250 <rad> -100 100 0
Ki\n                251 <rad> -100 100 0
Kd\n                252 <rad> -100 100 0

Rudder\d            271 <deg rad> -30 30 10
Aileron\d           272 <deg rad> -30 30 10
Elevator\d          273 <deg rad> -20 20 5
Airspeed\d          274 <kt m/s> 0 200 20
Altitude\d          275 <ft m> 0 30000 5000
Beta rate\d         276 <deg/s rad/s> -20 20 10
Beta\d              277 <deg rad> -20 20 10
Alpha rate\d        278 <deg/s rad/s> -20 20 10
Alpha\d             279 <deg rad> -20 20 10
Yaw rate\d          280 <deg/s rad/s> -20 20 10
Roll rate\d         281 <deg/s rad/s> -20 20 10
Pitch rate\d        282 <deg/s rad/s> -20 20 10
Yaw\d               283 <deg rad> -180 180 10
```



```
Roll\d              284 <deg rad> -50 50 10
Pitch\d             285 <deg rad> -20 20 10
G\d                 286 <G> -5 5 1
Loc Error\d         287 <deg rad> -5 5 1
G/S Error\d         288 <deg rad> -2 2 0.5
Rate of Climb\d     289 <fpm m/s> -1000 1000 100
Flight Data #1\d    290 <> 14.0 34.0 2.0
Flight Data #2\d    291 <> 14.0 34.0 2.0
Flight Data #3\d    292 <> 14.0 34.0 2.0

Record\o            301 <Stop Start Continue>
Replay\o            302 <Stop Start Continue>
Position\o          303 <Cockpit Rear Side Wing Tower>
Time\n              304 <Secs> 20 1000 60
Load file\l         305
```

This file is accessed when the IOS starts and provides the menus for the IOS display. The first line is treated as a comment and is ignored. Blank lines are also ignored. The list of commands *Master, Reposition, Map, Weather, Settings, Failures, Autopilot, Flight Plan, Target, Approach, Data Recording, Flight Data and Camera* define the top level commands.

Each command is defined by a unique number that matches the corresponding value in iosdefn.h. For example, *compass* is 63 and *visibility* is 97 and has a unique name that is displayed when the command is selected. The commands also have a qualifier *q, o, f, l, c, n, x* or *d* following a backslash, as defined in the following table:

qualifier	Abbrev.	Meaning
q	question	Requires a confirmation
o	option	Select an option
f	file name	Enter a file name
l	file list	Select a file from a list
c	cursor	Position the cursor
n	numeric	Enter a number
x	execute	Execute a command
d	data	Set a flight test data mode

A prompt, specific to each option type follows the command number inside angular brackets, in turn followed by default values, where the first value in the list is the default value.

qualifier	Meaning
q	Prompt requiring a response (accept or decline)
o	List of options (first option is the default)
f	Prompt
l	Not applicable
c	Not applicable
n	Default value, minimum value, maximum value
x	Not applicable
d	Minimum value, maximum value, increment

For example, `Flight Controls\o 132 <Column SideStick>` will prompt the user with 'Flight Controls', where the default is 'column' and the user can select either 'column' or 'sidestick'. Similarly, `Wind Speed\n 92 <Wind Speed (kt)> 0 80 0` will prompt the user with 'Wind Speed', where the minimum value is 0, the maximum value is 80 and the default value is 0. In addition, values are given in appropriate units and are converted to SI units by the IOS.

This file can be edited with a standard text editor. For example, to change the default magnetic heading to 11° E, alter the value from –5 to 11 on the line defining the values for 'Magnetic Variation' in this file.

5.7.3 Navaids Database

The navaids database is downloaded from the Navigraph website www.navigraph.com (X-Plane GNS430 version). The file /c/sim/ios/files/Navaids.txt contains the worldwide list of navaids (ADF, VOR, DME and ILS) used in the simulator. Note that the same file is also stored in the directory /c/sim/nfd/files/ in the navigation computer and both files must be identical. There is no requirement to modify this file and the database can be overwritten with a new revision.

A fragment of the beacons database is given below:

```
CIT,CRANFIELD,850.000,0,0,195,52.130156,-0.556881,0,EG,0
CIV,CHIEVRES,113.200,1,0,195,50.573972,3.832872,221,EB,0
CJ,JIANQIAO,324.000,0,0,195,30.305000,120.166667,0,ZS,0
CJ,CHUCHON,360.000,0,0,195,37.886667,127.725000,245,RK,0
CJ,CALDE SPRINGFIELD,240.000,0,0,195,39.794603,-89.592742,584,K5,0
CJN,CERREJON,113.400,1,1,195,11.230525,-72.492511,297,SK,0
CJN,CERREJON,415.000,0,0,195,11.235278,-72.508889,272,SK,0
CJN,CASTEJON,115.600,1,1,195,40.371972,-2.544611,3543,LE,0
```

The fields of this file are critical. The ident of the navigation aid is given followed by the name. The other fields include the frequency of the aid, the latitude and longitude in degrees, the beacon range and elevation. The frequency of each navaid is given in MHz for ILS and VOR beacons and KHz for ADF beacons. TACAN beacons are represented as DMEs.

5.7.4 Runways Database

The runways database is also downloaded from the Navigraph website (www.navigraph.com). The file /c/sim/files/Airports.txt contains the worldwide list of runways used in the simulator and the same file is also stored in the directory /c/sim/nfd/files/ in the navigation computer. There is no requirement to modify this file and the database can be overwritten with a new revision.

A fragment of the runways database is given below:

```
A,EGTB,BOOKER,51.611667,-0.808333,520,0,0,2300,0
R,06,64,2395,75,0,0.000,0,51.610228,-0.812881,513,3.00,40,1,0
R,24,244,2395,75,0,0.000,0,51.613264,-0.803539,513,3.00,40,1,0
R,35,348,2280,98,0,0.000,0,51.607136,-0.806622,482,3.00,40,1,0

A,EGTC,CRANFIELD,52.072222,-0.616667,358,6000,0,5900,0
R,03,33,5902,151,0,0.000,0,52.066797,-0.622022,358,3.00,50,1,0
R,21,213,5902,151,1,108.900,213,52.078978,-0.609797,358,3.00,54,1,0
```

The first two lines of the file provide a record of the version number and cycle date of the file and are ignored. The fields of this file are critical. The first field is 'A' for an Airfield or 'R' for a Runway. The fields include the ICAO ident, the name of the runway and the latitude, longitude and elevation. The runway data also includes the runway QDM, the latitude and longitude at the threshold, the length and width of the runway and the ILS frequency, for runways equipped with ILS. For airfields with several runways, only the major (the longest and/or ILS equipped) runway is selected.

5.7.5 Database Limitations

The databases contain worldwide navaids and runways. However, latitude and longitude ranges are defined in navlib.c to restrict the navigation region to an area covered by approximately 12° of latitude and 12° of longitude. The default values are 49°N to 61°N and 8°W to 2°E for the UK. However, it is straightforward to modify these values for other regions of the world. Currently, the maximum number of navaids and runways is set to 1000 in navlib.h. The databases provided by Navigraph are downloaded using the 'manual' option and are currently based on the X-Plane GNS430 format. Other databases and formats, based on the Arinc 424 standard, could be used.

5.8 Image Generation PC

The LFS includes a real-time image generation system for rendering an external scene. This system is composed of three image generators (IGs), one for each screen projection channel. The IG computers read the UDP packets broadcast by the other systems and render the view based on the data in these packets. Within the IG software, scene management is provided by OpenSceneGraph (OSG) 3.4 (http://www.openscenegraph.com/). OSG is an open source 3D graphics toolkit and enables the IGs to display visual databases of terrain, aircraft visual models, time of day, visibility and 2D overlays, such as Head-Up Displays.

5.8.1 OpenSceneGraph Libraries

OSG is a library used by the IG software. Precompiled libraries are located in the following directory on each of the IG computers:

```
/c/Program Files (x86)/OpenSceneGraph
```

This directory contains the libraries that the IG software links to (/bin), example OSG programs (/share), software development header files (/include) and libraries (/lib).

5.8.2 Using the Image Generators

The installation of the visualisation software is replicated on each of the IG computers. The visualisation program is called **cgi** and is found in the following directory:

```
/c/IG/visual
```

The IG software is executed via the MinGW prompt (opened by double-clicking on the MinGW icon found on the desktop of each IG computer). Once opened, the following commands will start the visualisation software, where *sim>* is the MinGw prompt.

```
sim> cd /c/IG/visual
sim> ./cgi bristol
```


The **cgi** program takes one string argument, defining the visual database to be used as the outside world view. Databases are currently available for Bristol, Hong Kong and Monterey airports with their respective cgi arguments *bristol, hong_kong* and *monterey*.

5.8.3 Organisation of the Flight Simulator Visual Databases

The flight simulator visual databases for the terrain and other objects are based on OpenFlight models, which are found in the following directory:

```
/c/SIM-DATA/databases
```

Each database is stored in a specific directory. The **cgi** software also uses additional models found in the **multigen** subdirectory.

5.8.4 Projector Calibration

To correctly display images produced by the IG system on the projection screens, the projectors require calibration. The images generated on the IG computers (which by default assume a flat projection surface) must be warped otherwise the projected images will be cast onto a curved screen and be distorted. The LFS projector calibration tool is a portable panel housed within the LFS cabin.

During the projector calibration process, it is helpful for the IG computers to display test patterns that clearly show the image warping effects of the current calibration configuration. A test pattern provides a feedback image from which calibration adjustments can be seen in real-time. The LFS IG systems can use the test pattern tool **pattern.exe** found in the following directory.

```
C:\IG\ScreenPattern
```

To execute to the tool on each of the IG computers, locate pattern.exe and double-click on its icon. Alternatively, in a MinGW window type the following commands.

```
sim> cd /c/IG/ScreenPattern
sim> ./pattern
```

A green and white grid will now be shown. By default, the pattern will be configured for the middle channel. The left and right patterns must be configured to show the correct pattern for the left and right projection channels. This is done by right-clicking the mouse anywhere and from the pop-up menu selecting **Channel 1** for the left image, or **Channel 3** for the right image. To further assist with the calibration process, it is useful to highlight the area of overlap between adjacent projected images. To do this, from the pop-up menu select **Toggle Overlap** followed by **Toggle Overlap Dots**. For the left channel, white vertical dashed and dotted lines will be shown on the right hand side of the screen. For the right channel, the white vertical lines will be on the left side of the screen. The left, middle and right channels should be showing images similar to those shown in Figure 5.8.4.

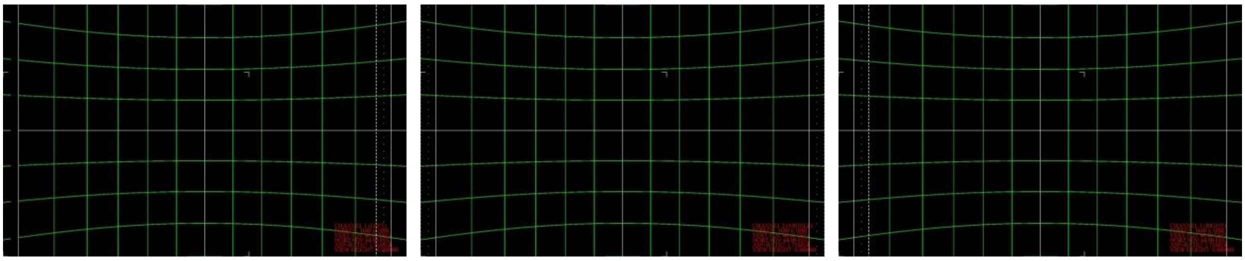


Figure 5.8.4 Projector Test Patterns

Once the projectors are calibrated, the curved grids shown in the figures will appear close to orthogonal from the pilot eye position. At this point a further calibration test can be made by displaying a *wall* pattern (a set of horizontal white lines spanning each of the channels). To show this pattern, from the pop-up select **Toggle Wall** for each of the IG computers. If the projectors are correctly calibrated, the horizontal white lines should look unwarped from the pilot eye position, similar to the effect of viewing a low straight wall from a position parallel to it.

To exit the calibration test pattern tool, right-click the mouse to show the pop-up menu and select **Quit**.

5.8.5 Compiling the OSG Libraries

OSG is preinstalled on the IG computers. In the event that OSG requires updating to a new version or reinstallation, OSG must be recompiled. The following steps describe this process. Note that these steps assume that MinGW is installed (along with the GCC compiler suite).

1. Download and install **cmake** (https://cmake.org/). This will install a program named CMAKE-GUI and a shortcut will be installed on the desktop.

2. Build the OSG libraries:
    a) From the OSG website (http://www.openscenegraph.com/), download and extract the latest OSG source files.
    b) Open the CMAKE-GUI application and drag the file **CMakeLists.txt** from the OSG root build directory to anywhere in the CMAKE-GUI window.
    c) Complete the top text entry box by browsing for the **root folder** of the extracted OSG source files. The text box below (where to build the binaries) can be set to the same directory.
    d) Press the *CONFIGURE* button and choose *MSYS* compilers.
    e) At this point, CMAKE-GUI will show various OSG build and installation options, compiler paths and potentially missing libraries. Most of the settings are correct, unless extra functionality is required from OSG. Additional libraries can be included such as *freetype*, *jpeg*, *png*. Configuration errors will be reported in the text panel at the bottom of the window. These must be fixed by correcting path errors or by setting the value to an empty string (inform the build system to ignore this option).
    f) Once errors have been corrected in CMAKE-GUI press *CONFIGURE* again.
    g) If there are no more configuration errors reported, in CMAKE-GUI press *GENERATE*. This will generate the makefiles for the particular system.

h) Open an MinGW prompt in 'Administrator Mode' (via the Start Menu, right-click on the MinGW shell icon and select *Run as Administrator*). Change directory to the OSG build root directory.
i) Type **make** (Note, this may take a significant time, typically one hour).
j) Type **make install** (Upon completion, the OSG build directory can be deleted)

3. Configure additional environment variables. In a DOS command shell, type:

```
setx OSG_FILE_PATH "C:\Program Files (x86)\OpenSceneGraph-Data" \M
setx PATH "%PATH;C:\ Program Files (x86)\OpenSceneGraph\bin" \M
```

These two commands only have to be executed once. The environment variables will remain, even after restarting the PC.

4. When building an OSG application, for example, modifying the cgi visualisation program (**`cgi.cpp`**), the OSG '**`include`**' and '**`lib`**' directories must be included in a Makefile (see the existing **`Makefile`** file in the directory **`/c/IG/visual`**).

5. Copy the OpenSceneGraph install directory **`C:\Program Files (x86)\OpenSceneGraph`** to the other IG PCs and repeat step 3 to set up the environment variables for these PCs.

6 INSTRUCTOR STATION

6.1 Introduction

The instructor station allows the instructor to monitor the simulated aircraft and to control the session. These functions are provided in the form of a monitor, a mouse and a keyboard. The monitor is used to display navigation data, aircraft tracks, approach paths and flight data. The mouse is used to select options from menus.

In many flight simulators, the instructor station is a complicated computer system, requiring several volumes of operating manuals and extensive training. The instructor station is intentionally designed for 'user friendliness' and can be understood by a user in a few hours.

Finally, one word to dispel any sense of alarm; it is not possible to cause any damage whatsoever to the flight simulator by misuse of the instructor station. In practice, users with minimal computer experience grasp the operation of the instructor station very quickly.

6.2 Initialisation

The instructor station is operated using a mouse. No keyboard operations are required.

Users are advised to work in their own directory because all saved files and printed files are written to the current directory. By working in a user-specific directory, users can generate and access their own files without interfering with the files of other users.

The initial display contains a map of the navigation region which is set to an arbitrary position as shown in Figure 6.1, with the mouse pointer positioned in the centre of the map display. The map can be readily re-positioned and these commands are described in sections 7.3.1 to 7.3.7. By clicking the left mouse key when there are no menus shown, a set of pop-up menus will appear to enable the user to make a wide range of selections. Three levels of menu are used; at the top level, the menu items group the main sets of commands together; at the next level, sub-menus are provided to select specific functions; at the lowest level, the user can enter or modify values or conditions, relating to the selected command.

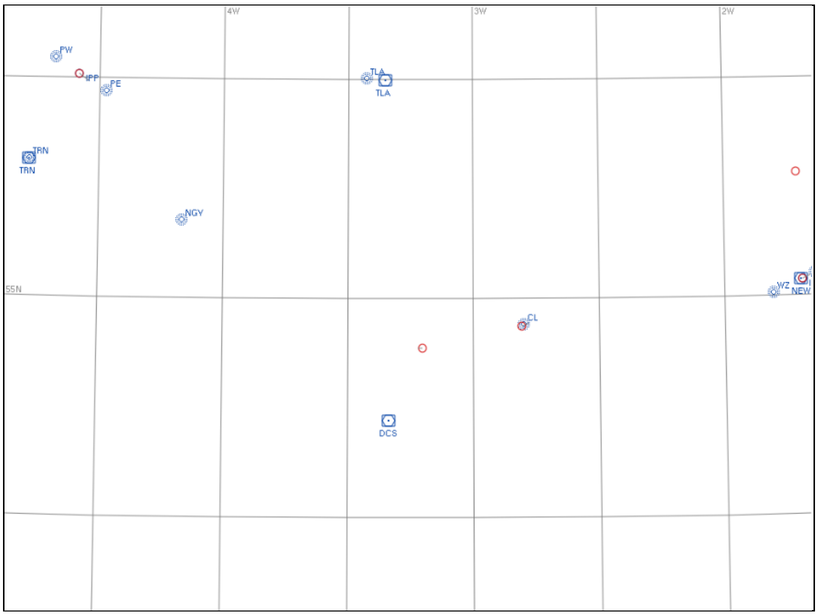


Figure 6.1

The mouse can be moved over both the map display and the menus without any effect and will simply follow the hand movement of the user. It is important that the mouse is correctly orientated and that the mouse is held in the natural axis of the hand, otherwise the mouse pointer movement and the actual mouse movement will appear to be 'disconnected'.

The mouse has two large buttons, near the mouse cable. Only the left mouse button is used for selection in the LFS instructor station. The right mouse button is not used and is ignored.

6.3 Menu Selection

Whenever no menu options are displayed, a set of menus will pop up in response to a mouse key press, as shown in Figure 6.2.

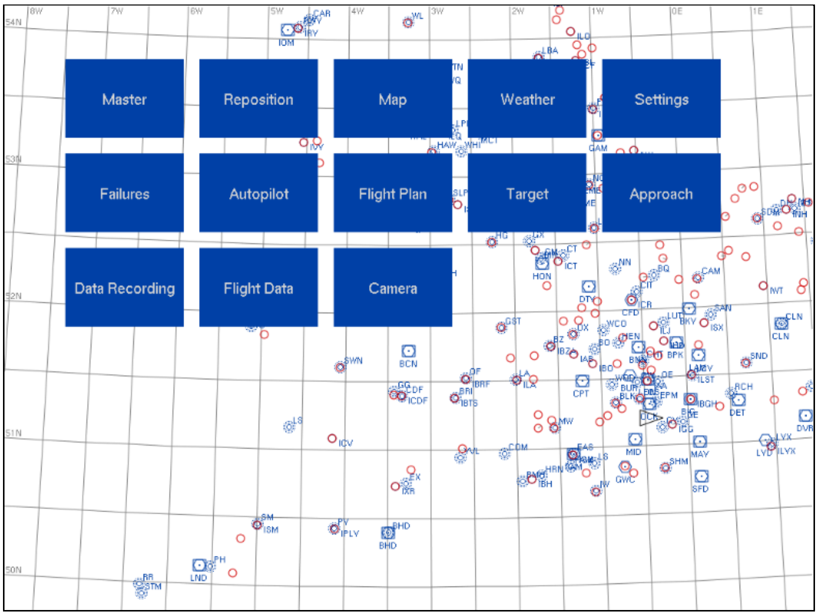


Figure 6.2

This is the main menu and shows the list of sub menus arranged in categories of simulation functions. Clicking on the sub menus shows a further set of menus relating to the selected function.

Occasionally, the wrong option may be selected if the mouse pointer has not been positioned carefully. However, all commands have a cancel option. In addition, to remove a sub-menu, simply move the mouse pointer away from the sub-menu and press the left mouse key.

Once the desired option is selected from the menu, a prompt box will appear which is specific to that command. This prompt box enables the user to select specific modes or flight conditions or to enter specific values. The details of each command are explained in detail in section 7.

The menus offer a range of commands appropriate to the management and monitoring of simulator sessions and the recording of flight data. These commands cover eight categories:

- A query, where the user can confirm or ignore an action, for example to quit the simulation.
- An option, where the user can select one of two or more options offered, for example to select the HUD mode
- Enter a file name to be used, for example to save flight data
- Obtain a list of file names, for example to select a previously restored state
- Action a command at the current cursor location
- Execute a specific command
- Enter numeric data
- Set flight data monitoring modes.

All sub menus provide options to cancel the selection or to modify values entered. Although there are over 150 commands, the grouping of the commands and consistency of the user interface provide a straightforward interface with the flight simulator.

Note that an error message is displayed for values which are out of range, as shown in Figure 6.3.

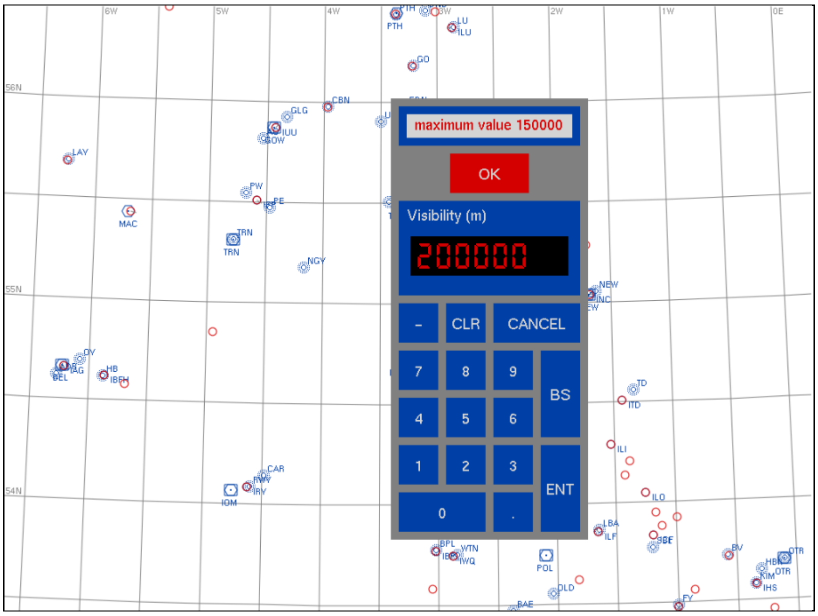


Figure 6.3

6.4 Main Menu Options

The main menu offers the range of commands to allow the instructor to monitor and to control a session in the flight simulator. The specific operations for each command of the main menu are given by the respective sub-menus.

Although there are over 100 possible options, only a few options are used regularly and it is quite possible to operate the instructor station by means of a small set of easily remembered commands. For completeness, all commands are described in detail in section 7.

The main menu commands, shown in Figure 6.2, are as follows:

6.4.1 Master

This command provides high level commands including exiting the simulation, selecting display modes, reloading a flight model or visual scene and activating Matlab

6.4.2 Reposition

It is a common requirement to reposition an aircraft to a known location or to reset flight conditions prior to an exercise. This command simplifies the operation.

6.4.3 Map

This command is the most commonly used command and allows an instructor to monitor the flight path (or track) of the aircraft and to organise the map in the same way that a chart is used. For example, it is possible to alter the scale of the map display or to re-position the centre of the map.

6.4.4 Weather

It is possible to simulate a range of weather conditions from the instructor station. In particular, for IFR operation it is possible to select a cloud base or to provide a cross wind. There are other facilities to alter the outside air temperature, the area QNH barometric pressure setting, turbulence and magnetic variation. Initially the flight simulator is set with a cloud base at 36000 feet, no wind, no turbulence, an outside air temperature of 15 degrees centigrade at sea-level and an area QNH of 1013 hPa.

6.4.5 Settings

This command is used to alter specific aircraft settings. For example, it is possible to alter the aircraft altitude, speed or heading or to alter the fuel quantity. The command also enables the side-stick to be selected rather than the centre stick or enables the Head-up Display (HUD).

6.4.6 Failures

This menu option allows the instructor to fail aircraft systems such as gear, flaps, DME, localiser, glide slope indicator and engines. For the selected aircraft sub-system, the current state of the sub-system is indicated when the sub-menu option is selected and can be set to an operational or failed (malfunction) state. The failures also include instruments. For limited panel operation in instrument flight training, the instructor can fail one or more of the aircraft instruments. Note that the state of an instrument or sub-system can be examined simply by selecting the appropriate instrument or sub-system and then clicking on the CANCEL button.

6.4.7 Autopilot

The flight simulator provides a range of autopilot functions which can be entered via the FCU. The sub-menu functions provide height, heading and speed control as well as auto-land functions. The flight control modes can be enabled or disabled from the instructor station.

6.4.8 Flight Plan

A flight plan can be loaded and activated or deactivated. In addition, a Script file can be activated from the instructor to initiate a sequence of complex events and to record aircraft data in standard spread-sheet formats.

6.4.9 Target

Dynamic aircraft targets can be generated by the visual system, for example to support formation flying or to introduce conflicting traffic. The type, speed, position and flight path of the target aircraft can be defined to specify simple aircraft dynamics.

6.4.10 Approach

On the approach mode, the aircraft track is displayed showing the glide path, the localiser situation and the airspeed. This command allows the user to define the plotting range and to reset the display.

6.4.11 Data Recording

Specific options are provided to manage the recording and display of flight data, for example, to start or stop flight data recording or to add a recording mark.

6.4.12 Flight Data

Specific flight data variables can be selected to be displayed and recorded. In addition, the range and units of each variable can be selected.

6.4.13 Camera

Simulator frames can be stored for replay by the visual system. Options are provided to start, stop and replay a recording, set the length of recording and alter the camera location (or eye point). By default, the eye point used by the visual system is the pilot eye point. The camera location can be changed to behind or to the side of the aircraft or to the airfield control tower.

7 MENU COMMANDS

In the following section, each command in the sub-menus is described in detail. This description includes the function of the command, the effect of the command, the method of selection of the command, confirmation that the command has been executed, an illustrated example of the command, applications of the commands and related commands.

7.1 Master

These commands provide overall control of the simulation, setting operating modes, taking a copy of the screen and termination of a simulator session. The menu options are shown in Figure 7.1.

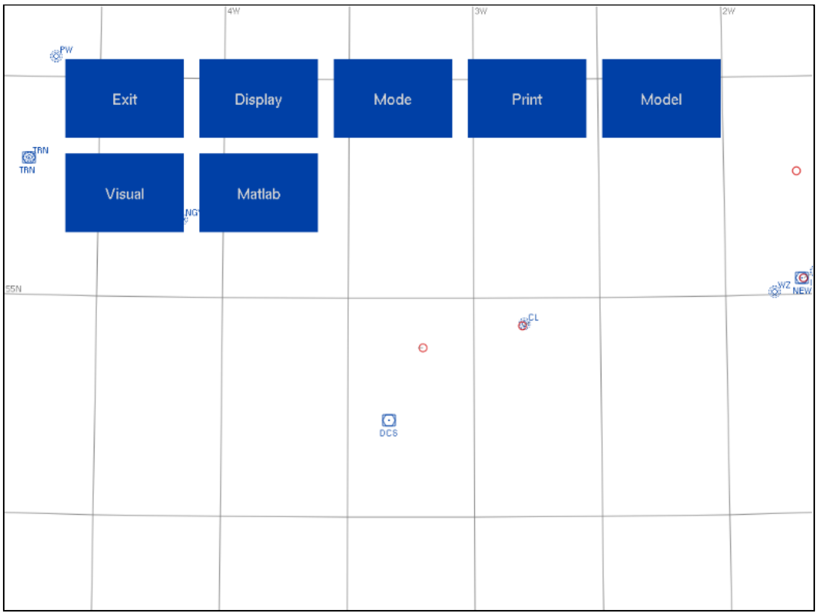


Figure 7.1

7.1.1 Master-Exit

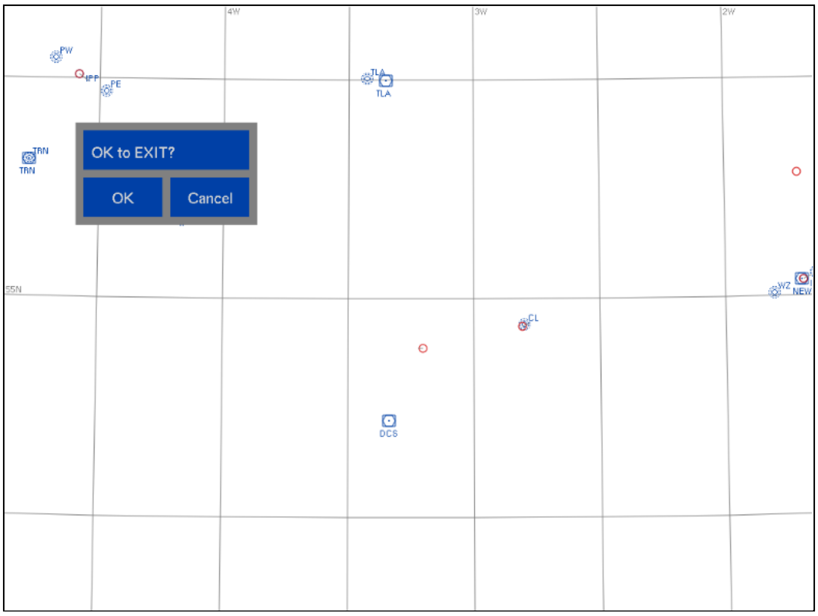


Figure 7.1.1

Function: Exit the instructor station program.

Effect: The instructor station program is terminated, returning control to MinGw. A message is sent to stop the other simulator computers. The display is cleared and any pending file operations are completed.

Selection: Select the **Master** option from the main menu. Select the **Exit** option from the sub-menu. Click the OK confirmation button to exit to MinGw.

Confirmation: The IOS and all the other simulator computers should also exit the simulator software and return to the MinGw terminal.

Example: In Figure 7.1.1, the Exit command has been selected.

Applications:

- To terminate a simulator session.

See also:

7.1.2 *Master-Display*

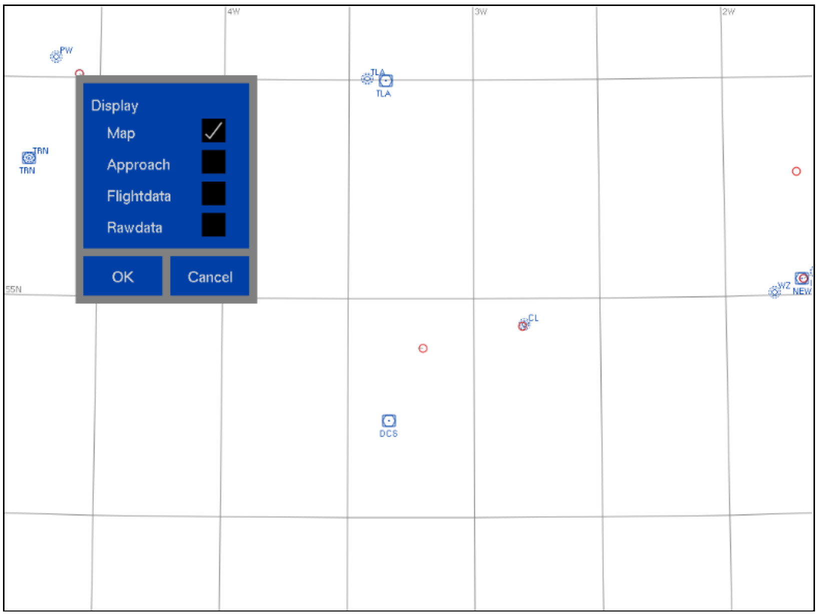


Figure 7.1.2

Function: Select the display.

Effect: The display will display a map view of navaids, runways and aircraft tracks, an approach showing glidepath, localiser and airspeed, flight data as strip graphs or raw numeric data. Examples of the four modes are shown on the following page.

Selection: Select the **Master** option from the main menu. Select the **Display** option from the sub-menu. Click on one of the four modes. Click the OK button to confirm.

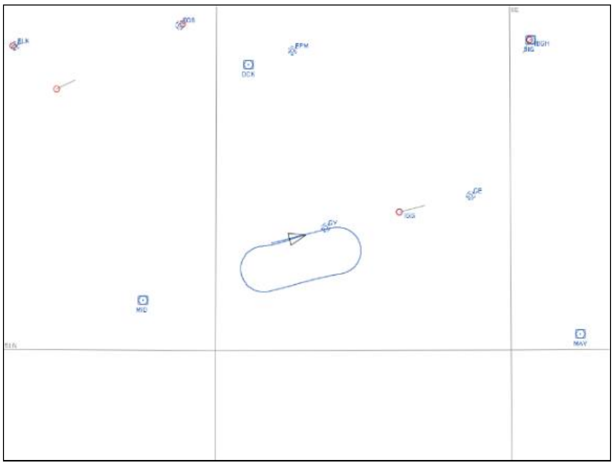
Confirmation: The display shows the selected display mode.

Example: In Figure 7.1.2, the Display command has been selected and the Map option is selected.

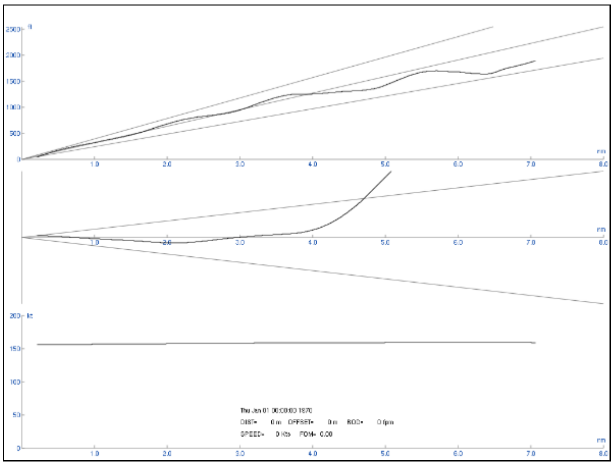
Applications:

- To illustrate an approach or check raw data values.

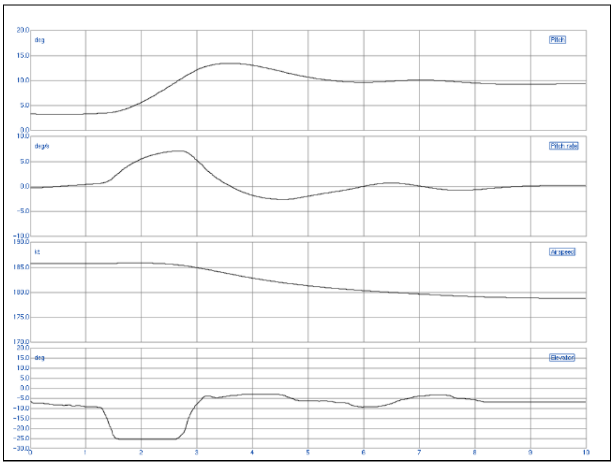
See also:



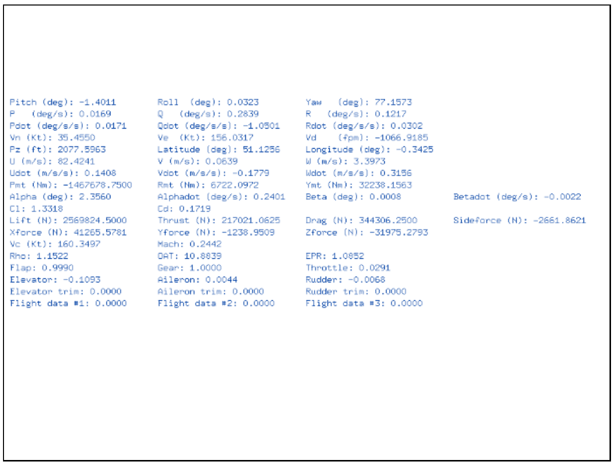
Map mode



Approach Mode



Data Recording mode



Raw Data Mode

7.1.3 Master-Mode

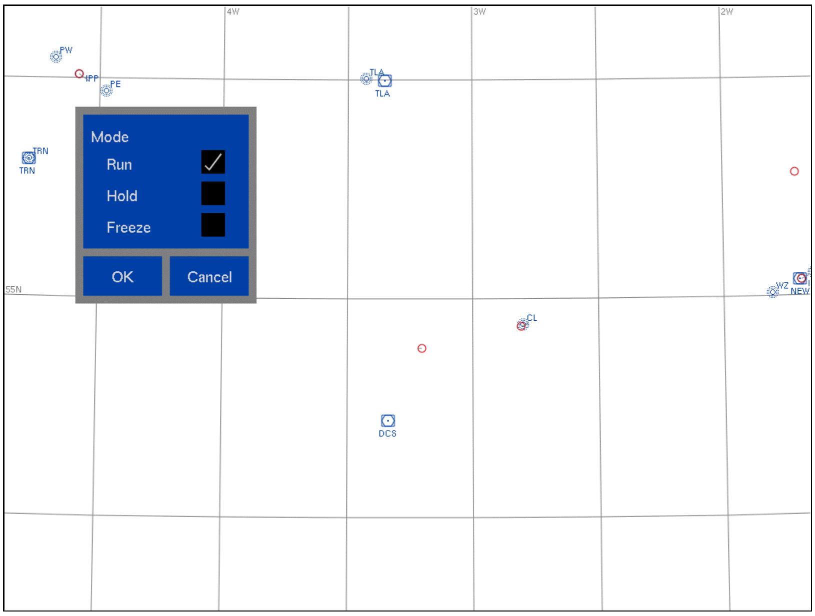


Figure 7.1.3

Function: Select or release the Run, Hold and Freeze modes from the instructor station.

Effect: The trainer is put into one of three modes:

1. HOLD - the equivalent of pressing the HOLD button – the simulator is suspended
2. FREEZE - the aircraft dynamics are correctly simulated but the aircraft position is fixed in space
3. RUN - the equivalent of releasing the HOLD button.

Note: The hold mode can be engaged by either pressing the flight simulator HOLD button or selecting the instructor station HOLD mode. To release the hold mode, both the flight simulator HOLD button and the instructor station hold mode must be released. In the hold mode, the flight model is not updated, the sound effects are switched off and the clock/timer is frozen.

Selection: Select the **Master** option from the main menu. Select the **Mode** option from the sub-menu. Select one of the three options to activate the Hold, Freeze or Running mode. The mode is maintained until another mode is selected.

Confirmation: The flight simulator is placed in the hold/freeze mode or released from the hold/freeze mode as appropriate.

Example: In Figure 7.1.3, the flight simulator has been set in the RUN mode.

Applications:

- To suspend the simulator to explain a situation
- To resume the simulation.

See also:

7.1.4 Master-Print

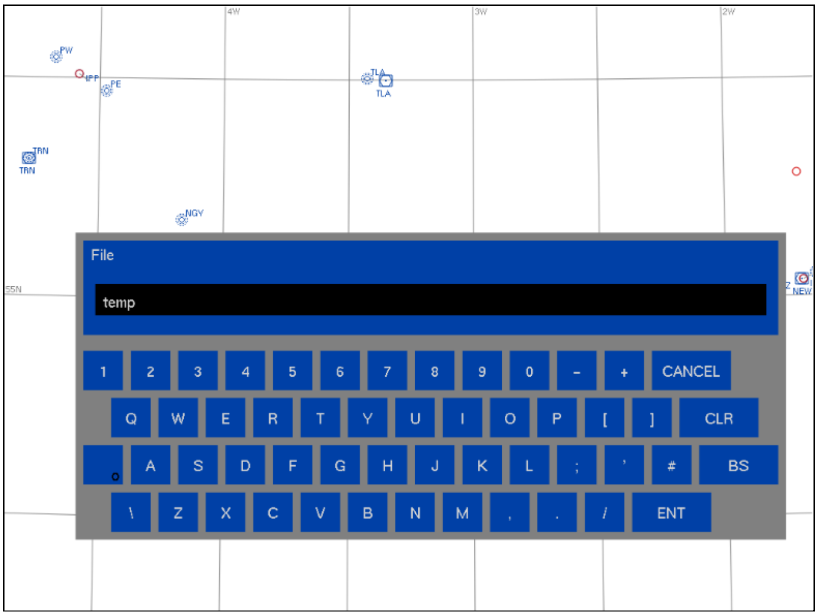


Figure 7.1.4

Function: Print the current display.

Effect: The user is prompted for the name of a file to save a copy of the screen for subsequent printing. The file is written in .PNG format, which can be imported into a wide range of software packages.

Selection: Select the **Master** option from the main menu. Select the **Print** option from the sub-menu. A file name box is displayed, allowing the name of the printed file to be entered. The .PNG extension is not required and is automatically appended to the file name. The file name can be edited using the keyboard characters, back space, CLR and shift keys.

Confirmation: The file name box is removed.

Example: In Figure 7.4, the Print command has been selected and a file name 'temp' has been entered. A file 'temp.png' will be written to the current user directory.

Applications:

- To generate a print file for a report.
- To save a copy of the map display for de-briefing.

See also:

7.1.5 Master-Model

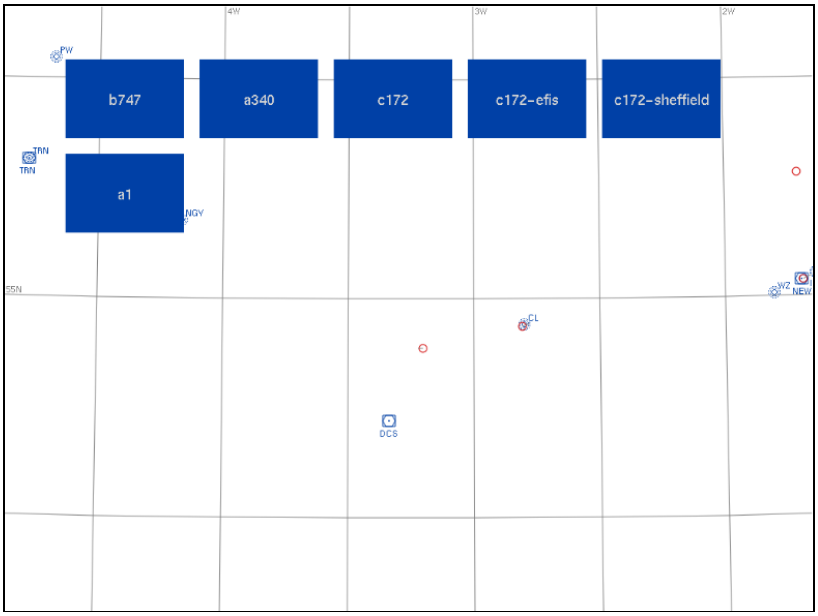


Figure 7.1.5

Function: Load a flight model.

Effect: The user can select a specific flight model to replace the existing flight model. All current flight model settings are lost and the default settings are selected for the loaded flight model.

Selection: Select the **Master** option from the main menu. Select the **Model** option from the sub-menu. A list of flight models is displayed. Move the mouse over the list of models and press the left mouse key to select the model. There is no confirmation prompt for this command.

Confirmation: A new flight model will be loaded and the instrument displays for the new flight model will be displayed.

Example: In Figure 7.1.5, the Model command has been selected and the current selection of flight models is displayed.

Applications:

- To load a new flight model.

See also: Master-Visual, Master-Reposition-Restore

7.1.6 Master-Visual

Figure 7.1.6

Function: Load a visual database for the IG systems.

Effect: The user can select a specific visual database.

Selection: Select the **Master** option from the main menu. Select the **Visual** option from the sub-menu. A list of visual databases is displayed. Move the mouse over the list of databases and press the left mouse key to select the database. There is no confirmation prompt for this command.

Confirmation: A new visual scene will be displayed.

Example: In Figure 7.1.6, the Visual command has been selected and the current selection of visual databases is displayed.

Applications:

- To load a new scene.

See also: Master-Model, Master-Reposition-Restore

7.1.7 Master-Matlab

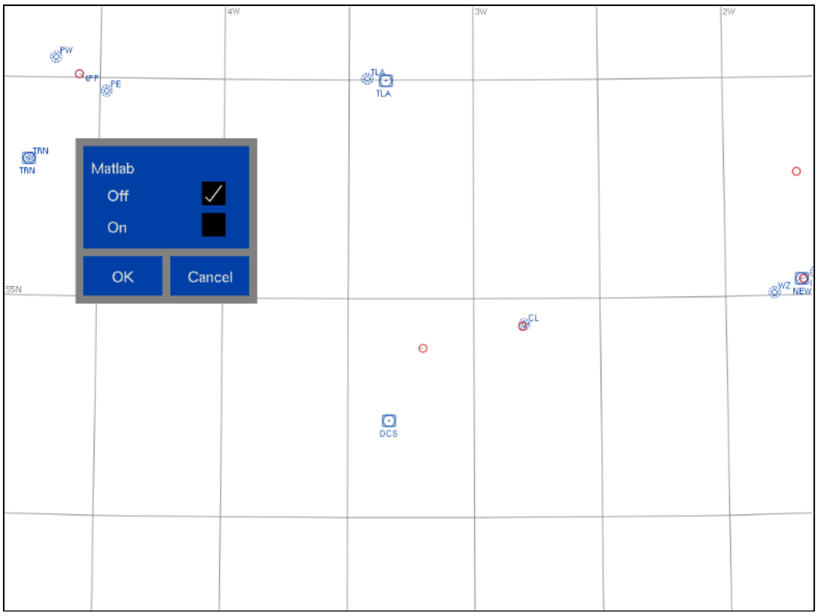


Figure 7.1.7

Function: Activate or deactivate a Matlab program.

Effect: A compiled Matlab program is activated to run during the simulation. The Matlab program responds to packets generated during each frame of simulation and sends a packet to actuate the flight controls.

Note: Matlab must be running and a Matlab program must have been loaded and started.

Selection: Select the **Master** option from the main menu. Select the **Matlab** option from the sub-menu. Click the OK confirmation button to activate or deactivate a Matlab program computer on the Matlab computer.

Confirmation: The Matlab program will start or stop operating.

Example: In Figure 7.1.7 the Matlab command program will be turned off (deactivated).

Applications:

- To test Matlab algorithms.
- To demonstrate flight control laws

See also:

7.2 Reposition

These commands are used to reposition the aircraft and set specific flight conditions such as altitude or heading. The menu options are shown in Figure 7.2. In addition, flight data can be saved and subsequently retrieved.

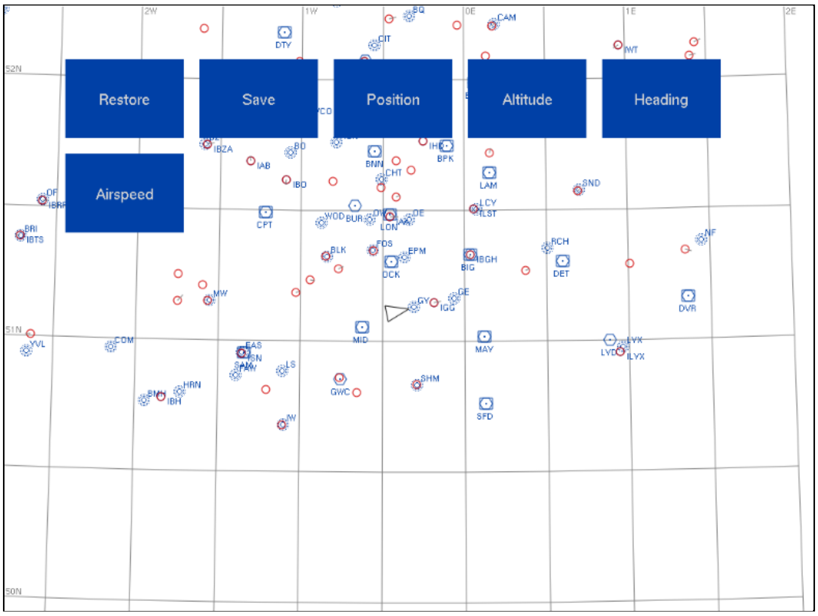


Figure 7.2

7.2.1 Reposition-Restore

Figure 7.2.1

Function: Restore the aircraft state from a previously saved file.

Effect: A file with a .sav file extension is selected from the current user directory and the navigation frequencies, altitude, heading and aircraft speed are reset to the values saved in the file.

Selection: Select the **Reposition** option from the main menu. Select the **Restore** option from the sub-menu. A list of files in the current directory with a .sav extension is displayed. Select the particular saved file by clicking on the file.

Confirmation: The aircraft is re-positioned, an aircraft symbol should appear at the appropriate position on the map display and new navigation frequencies are displayed. Note that if the flight simulator is in the 'hold' mode, some aircraft instruments may not necessarily be updated until the aircraft is released from the 'hold' mode.

Example: In Figure 7.2.1, the aircraft is restored to the position, altitude, speed, attitude and frequencies previously stored in the file gat6.sav.

Applications:
- To re-position the aircraft on an ILS approach at the correct speed and rate of descent and with the correct frequencies selected.
- To reposition the aircraft for take-off.

See also: Reposition-Save.

7.2.2 Reposition-Save

Figure 7.2.2

Function: Write the current state of the aircraft to a file for subsequent use by a 'restore' command.

Effect: The user is prompted for the name of a file to save a copy of the aircraft state. Variables including the aircraft position, speed, attitude and selected navigation frequencies are saved in the named file.

Selection: Select the **Reposition** option from the main menu. Select the **Save** option from the sub-menu. A file name box is displayed, allowing the name of the file to be saved to be entered. The .sav extension is not required and is automatically appended to the file name. The file is written to the current user directory. The file name can be edited using the back-space, CLR and shift keys. The aircraft is unaffected by execution of this command. If a file with the same name exists in the user directory, it is over-written (and no warning is given).

Confirmation: The data necessary to reset the aircraft exactly to the current state is written to the named file.

Example: In Figure 7.2.2, the current state of the aircraft has been saved in the file 'filename'.

Applications:

- To save a correctly established ILS approach to provide the initial position for an ILS approach.
- To save a take-off position in order to save time in setting up the simulator to start a navigation exercise.
- To save the state of a flight at a particular point in order to resume the exercise from that position or to illustrate some particular point during de-briefing.

See also: Reposition-Restore.

7.2.3 Reposition-Position

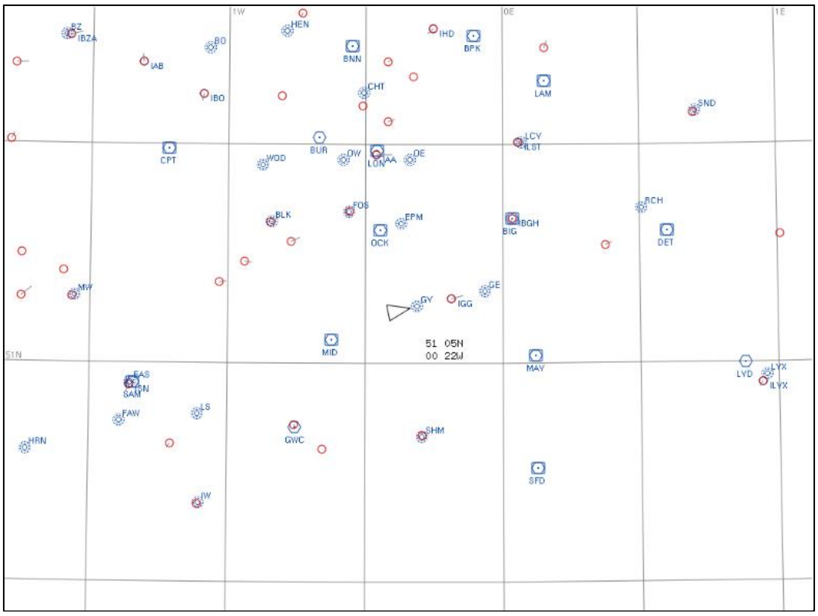


Figure 7.2.3

Function: Position the aircraft on the displayed map.

Effect: The aircraft is moved to the selected position. The navigation frequencies, altitude, heading and aircraft speed are unaltered by this command.

Selection: Select the **Reposition** option from the main menu. Select the **Position** option from the sub-menu. As the cursor is moved over the map display area, the map co-ordinates of the cursor are displayed. At the appropriate position on the map display, press the left mouse key to re-position the aircraft.

Confirmation: As soon as the aircraft is re-positioned, a new aircraft symbol should appear on the display.

Example: In Figure 6.8, the aircraft has been positioned approximately 10 NM north east of Midhurst VOR (MID), at 51° 05' N and 00° 22' W.

Applications:
- To position the aircraft overhead an NDB or VOR.
- To position the aircraft at a known position of latitude and longitude in order to commence an approach.
- To position an aircraft left (or right) of the localiser for an ILS approach.

See also:
Reposition-Altitude, Reposition-Heading, Reposition-Speed, Reposition-Restore.

7.2.4 Reposition-Altitude

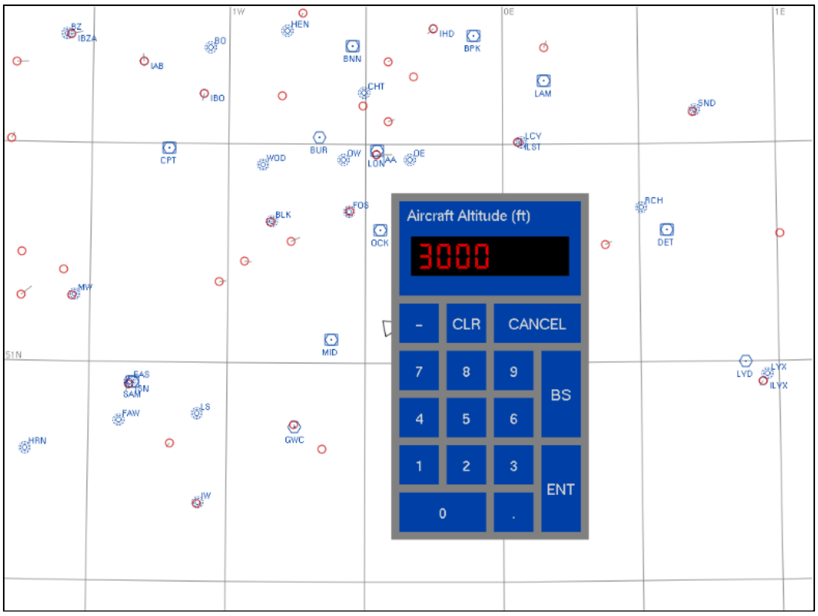


Figure 7.2.4

Function: Set the aircraft altitude in feet ASL.

Effect: The altitude of the aircraft is set to the selected value in feet ALS.

Selection: Select the **Reposition** option from the main menu. Select the **Altitude** option from the sub-menu. The current value of the aircraft altitude is displayed and can be set to a new value.

Confirmation: The new altitude is indicated on the aircraft altimeters. Note that the altitude indicated on the altimeters includes the effect of barometric pressure settings.

Example: In Figure 7.2.4, the aircraft altitude has been set to 3000 feet.

Applications:

- To illustrate the effect of altitude on engine performance
- To illustrate the effect of altitude on runway perspective by altering aircraft altitude.
- To position an aircraft above (or below) the glide slope for an ILS approach.

See also: Reposition-Heading, Reposition-Speed.

7.2.5 Reposition-Heading

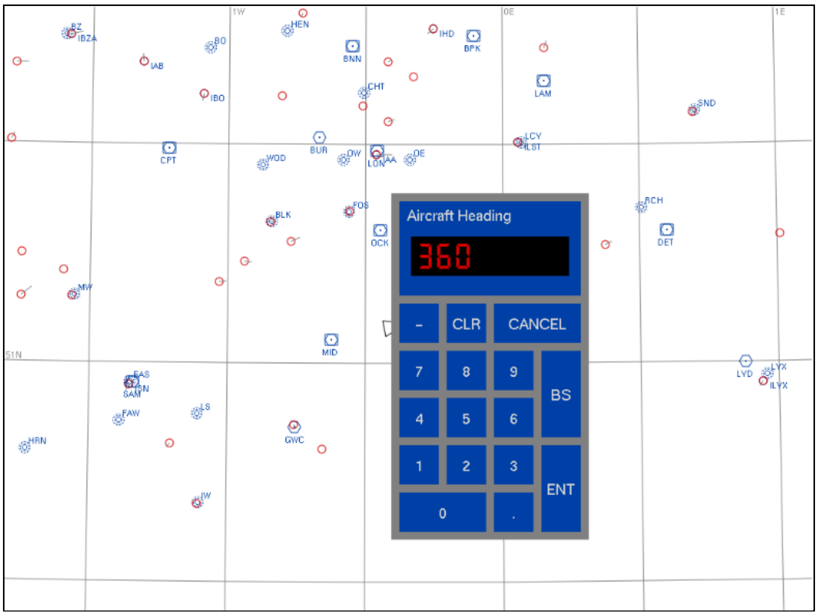


Figure 7.2.5

Function: Set the aircraft heading in degrees magnetic.

Effect: The magnetic heading of the aircraft is set to the selected value in degrees magnetic.

Selection: Select the **Reposition** option from the main menu. Select the **Heading** option from the sub-menu. The current value of the aircraft heading is displayed and can be set to a new value.

Confirmation: The new heading is indicated on the aircraft magnetic compass DI, HSI and RMI, as appropriate. The heading selected is the magnetic heading.

Example: In Figure 7.2.5, the aircraft heading has been set to 360 degrees.

Applications:
- To set the aircraft heading to a runway QDM.

See also: Reposition-Altitude, Reposition-Speed.

7.2.6 Reposition-Speed

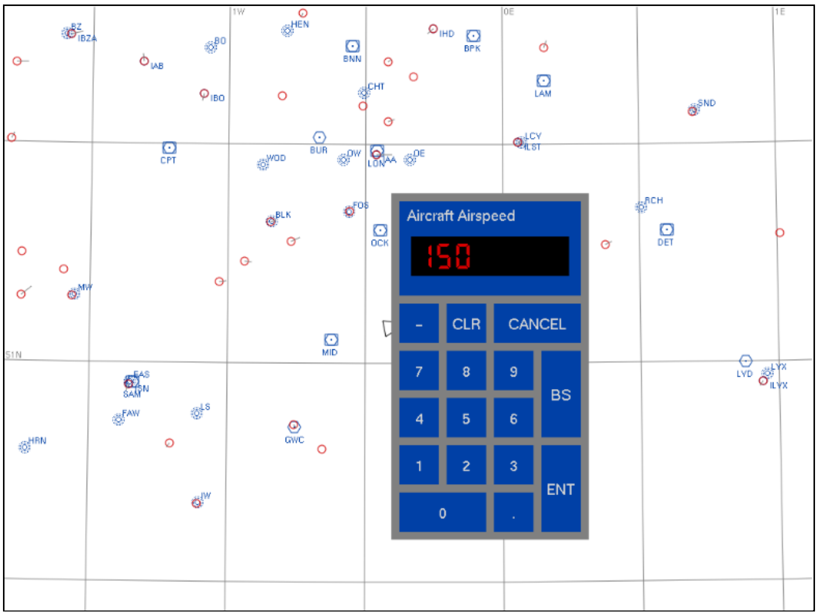


Figure 7.2.6

Function: Set the aircraft speed in Kts.

Effect: The speed of the aircraft is set to the selected value in Kts. Note that the value transferred to the flight model is the indicated airspeed, not the true airspeed.

Selection: Select the **Reposition** option from the main menu. Select the **Speed** option from the sub-menu. The current value of the aircraft airspeed is displayed and can be set to a new value.

Confirmation: A new aircraft speed is indicated on the airspeed indicator and Mach meter (if applicable).

Example: In Figure 7.2.6, the aircraft speed has been set to 150 Kts.

Applications:

- To set the aircraft speed for an approach.
- To simulate the effects of wind shear by setting the aircraft speed to a lower value than the indicated airspeed.

See also: Reposition-Altitude, Reposition-Heading.

7.3 Map

A selection of commands are provided to manage the map display, as shown in Figure 7.3.

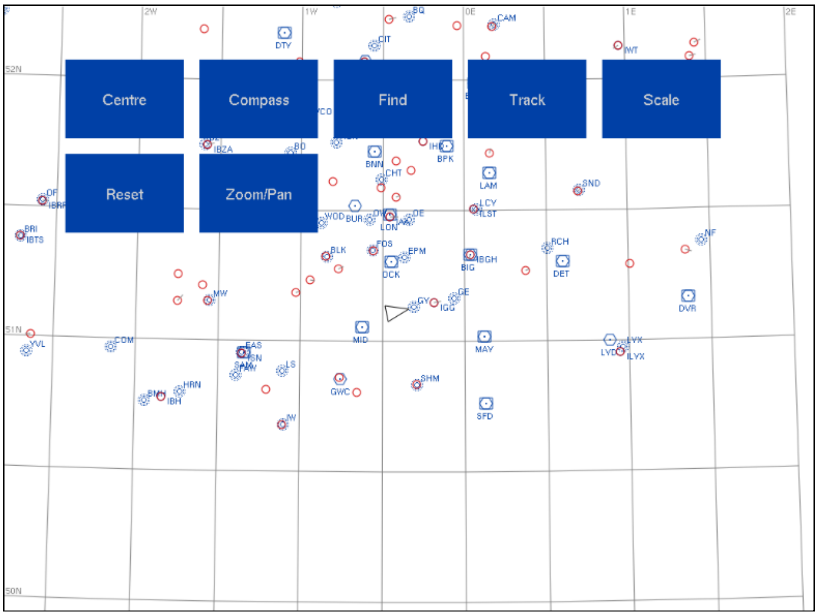


Figure 7.3

Tracks and compasses can be added to the map and the user is able to centre, zoom and pan to align the display according to the requirements of an exercise. In addition, the map can centred on the current aircraft location.

7.3.1 Map-Centre

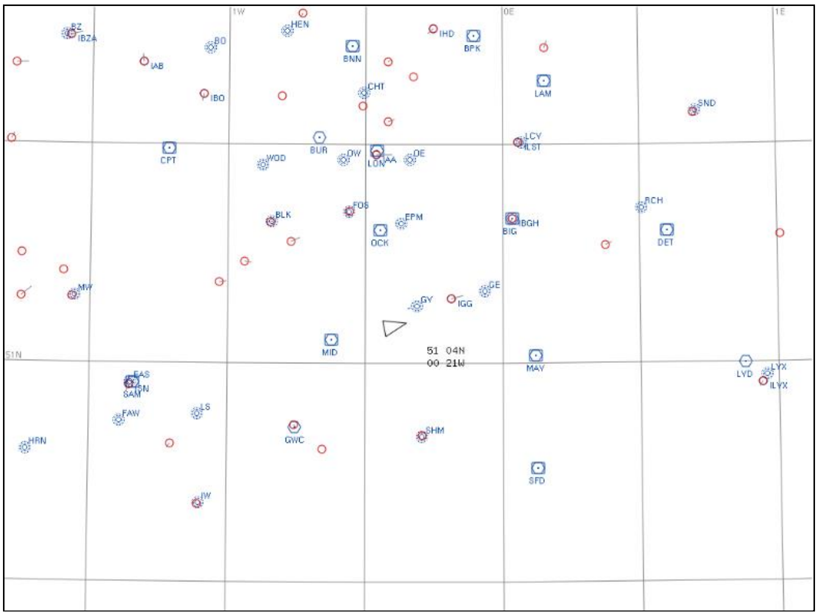


Figure 7.3.1

Function: Centres the displayed map.

Effect: As the cursor is moved, the latitude and longitude of the cursor is displayed. When the left key is pressed the map is redrawn, with the map re-centred at the selected position. The scale of the map, aircraft tracks and entered tracks are unaltered by this command. The command has no effect on aircraft operation.

Selection: Select the **Map** option from the main menu. Select the **Centre** option from the sub-menu. As the mouse pointer is moved over the map display area, the map co-ordinates of the mouse pointer are displayed. At the appropriate position on the map display, press the left mouse key to re-centre the map.

Confirmation: The map is redrawn, with the mouse pointer at the centre of the map display.

Example: In Figure 7.3.1, the map is about to be centred at the position 51 04N 00 21W.

Applications:
- To 'scroll' the map up, down, left or right without altering previous map settings.
- To centre the map on a navigation beacon or runway.
- To centre the map at a specific position of latitude and longitude in order to monitor a holding pattern.

See also: Map-Find, Map-Scale, Map-Reset.

7.3.2 Map-Compass

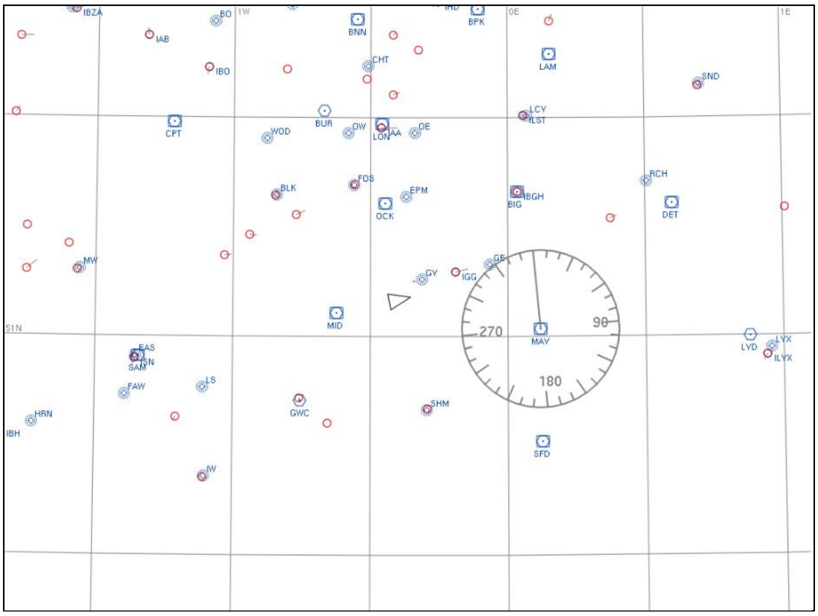


Figure 7. 3.2

Function: Displays a 'compass rose' set to the current magnetic variation.

Effect: A compass rose, aligned to magnetic north, is drawn at the selected position on the map. Up to 20 compass roses may be entered. A compass rose once entered, can only be erased by executing the **Map-Reset** command. This command has no effect on aircraft operation.

Selection: Select the **Map** option from the main menu. Select the **Compass** option from the sub-menu. As the mouse pointer is moved over the map display area, the map co-ordinates of the mouse pointer are displayed. At the appropriate position on the map display, press the left mouse key to draw a compass rose.

Confirmation: A compass rose is drawn at the tip of the mouse pointer.

Example: In Figure 7.3.2, a compass rose has been draw over the VOR at Mayfield (MAY).

Applications:

- To draw a compass rose at a navigation beacon or runway.
- To draw a compass rose at the current aircraft position.
- To draw a compass rose at a specific position of latitude and longitude.

See also: Map-Track.

7.3.3 Map-Find

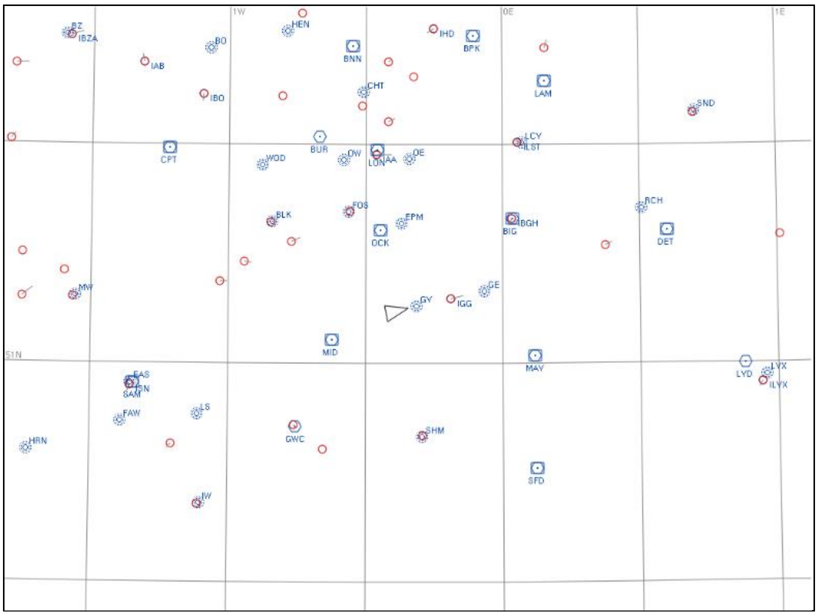


Figure 7.3.3

Function: Centres the displayed map at the current aircraft position.

Effect: The map is redrawn, with the aircraft at the centre of the map display. The scale of the map, aircraft tracks, entered tracks and 'compass roses' are unaltered by this command. The command has no effect on aircraft operation.

Selection: Select the **Map** option from the main menu. Select the **Find** option from the sub-menu. When the left mouse key is pressed, the map is redrawn with the aircraft at the centre of the map.

Confirmation: The map is redrawn, with the aircraft symbol at the centre of the map display.

Example: In Figure 7.3.3, the map will been centred on the aircraft, which is currently positioned near the NDB GY, heading 080.

Applications:
- To locate the aircraft, if it is off the map display.
- To centre the display at the current aircraft position before printing.

See also: Map-Centre, Map-Scale, Map-Reset.

7.3.4 Map-Track

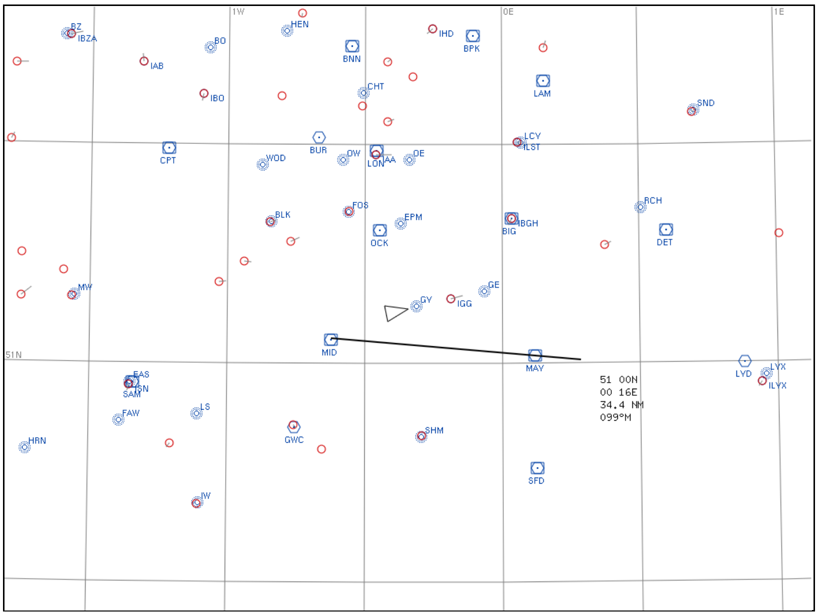


Figure 7.3.4

Function: Displays a straight line on the map display.

Effect: A straight line is added to the map display. As the line is drawn, it is 'rubber-banded' and the current line length (in NMs) and bearing (in degrees magnetic) are displayed. Up to 20 track lines may be entered. Track lines once entered can only be erased by executing the **Reset** command. This command has no effect on aircraft operation.

Selection: Select the **Map** option from the main menu. Select the **Track** option from the sub-menu. As the mouse pointer is moved over the map display area, the map co-ordinates of the mouse pointer are displayed. At the appropriate position on the map display, press the left mouse key to establish the start point of the line. As the mouse is moved, the line 'follows' the mouse pointer (rubber banding) and the track length and bearing are displayed. Press the left mouse key to establish the end point.

Confirmation: A grey track line is drawn on the map.

Example: In Figure 6.12, a track line has been drawn from Midhurst VOR (MID) through Mayfield VOR (MAY). The track length is 34.4 Nm and the bearing is 099°M.

Applications:

- To draw a track between two navigation beacons. The pilot can be asked to intercept the inbound track to one of the stations.
- To add a reporting point on the map in the form of a cross (two short track lines).

See also: Map-Compass.

7.3.5 Map-Scale

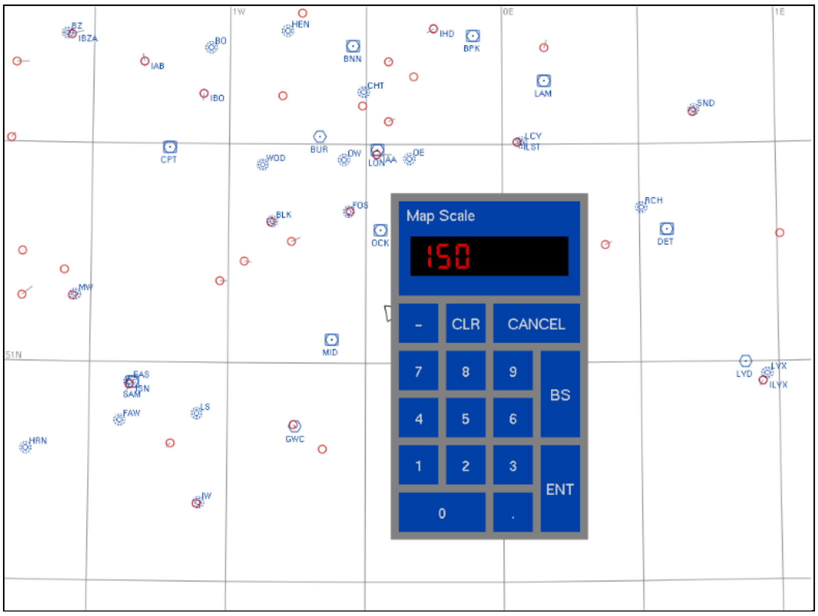


Figure 7.3.5

Function: Selects the scale of the map.

Effect: The map is redrawn to the selected scale. The centre of the map is unaltered by this command. The position of the mouse pointer has no effect on this command. The scale factor does not correspond to chart scales. Increasing the scale factor increases the map scale. Existing tracks and 'compass roses' are re-displayed. This command has no effect on aircraft operation.

Selection: Select the **Map** option from the main menu. Select the **Scale** option from the sub-menu. The current map scale is displayed and can be set to a new value. The position of the centre of the map is not affected by this command.

Confirmation: The map is redrawn at the new scale.

Example: In Figure 7.3.5, the map scale has been changed to 150.

Applications:
- To 'zoom in', in order to monitor a hold pattern so that the aircraft track 'fills' the map display.
- To change the map scale to monitor an aircraft track if the aircraft moves off the map display.

See also: Map-Find, Map-Centre, Map-Reset.

7.3.6 Map-Reset

Figure 7.3.6

Function: Reset the displayed map.

Effect: The map is redrawn at the current map scale and centre. The tracks formed by the aircraft and entered by the instructor are discarded. This command has no effect on aircraft operation. Previously entered 'compass roses' are also discarded.

Selection: Select the **Map** option from the main menu. Select the **Reset** option from the sub-menu. Click the OK confirmation button to reset the map display.

Confirmation: The map is redrawn centred at its current position and with its current scale. All tracks and compass cards are removed.

Example: In Figure 7.3.6, the user is asked to confirm resetting the map.

Applications:

- To erase the tracks from a previous exercise.

See also: Map-Find, Map-Scale, Map-Centre.

7.4 Weather

A selection of commands are provided to set the weather and environmental conditions, as shown in Figure 7.4.

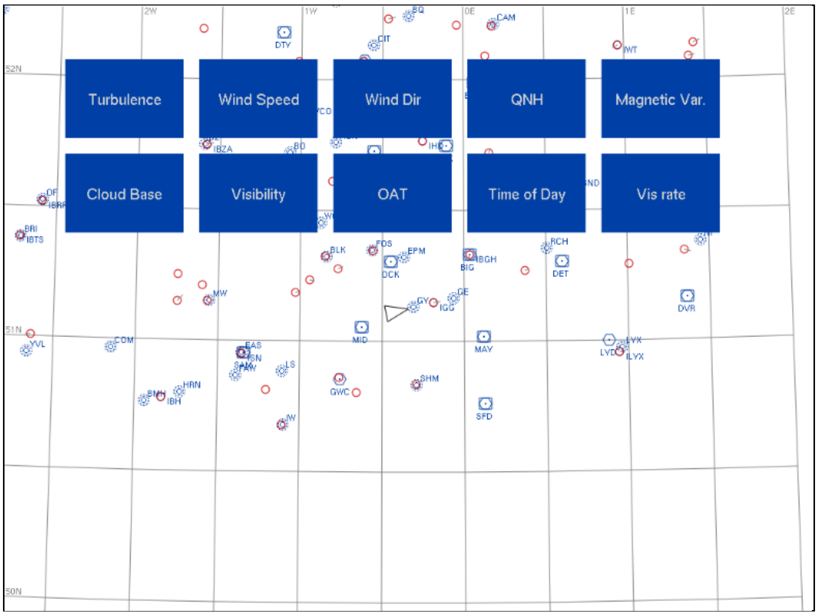


Figure 7.4

The effects of these settings impact on the aerodynamic model (turbulence, wind speed, wind direction and outside air temperature), the instrument displays (magnetic variation and QNH) and the visual scene (cloud base, visibility, visibility rate and time of day). They are provided to create a realistic external environment and can also be used to increase pilot work load.

7.4.1 Weather-Turbulence

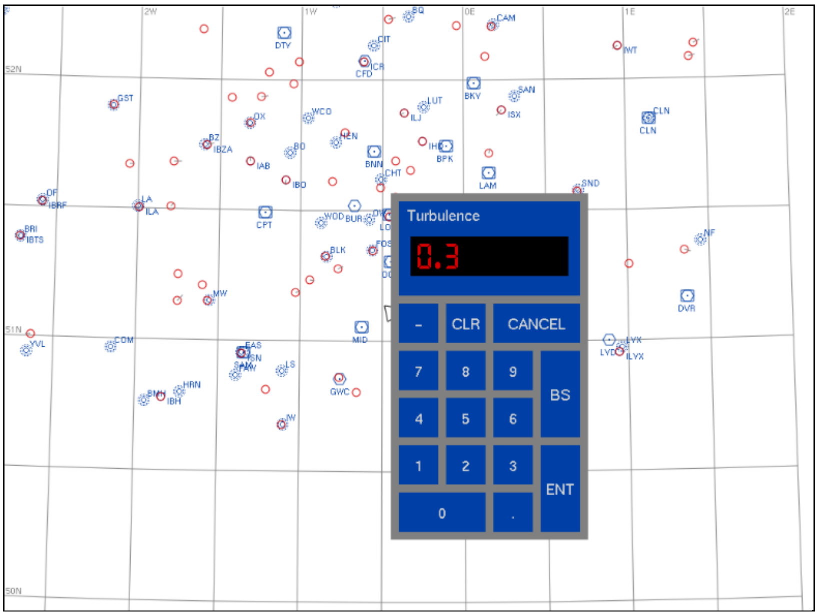


Figure 7.4.1

Function: Set a turbulence level. The turbulence model is based on the RAE model.

Effect: A turbulence effect is incorporated in the flight model. Initially, the turbulence is set to zero (no turbulence). The maximum turbulence value of 1.0 corresponds to a very high level of turbulence. Values of turbulence between 0.1 and 0.2 correspond to typical turbulence in strong gusty winds.

Selection: Select the **Weather** option from the main menu. Select the **Turbulence** option from the sub-menu. The current level of turbulence is displayed and can be set to a new value.

Confirmation: There is no visible confirmation, although the effect of a change of turbulence may become apparent in terms of the handling qualities of the simulated aircraft.

Example: In Figure 7.4.1, the turbulence has been set to 0.3.

Applications:

- To add turbulence to increase pilot work load during an IFR exercise.

See also: Weather-Wind Dir, Weather-Wind Speed.

7.4.2 Weather-Wind Speed

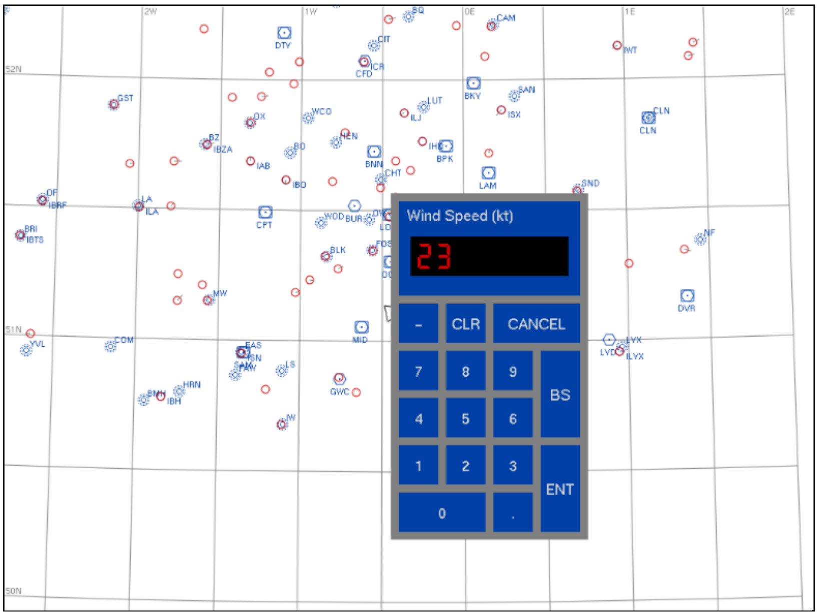


Figure 7.4.2

Function: Set the wind speed in Kts.

Effect: The wind speed used by the flight model is set to the selected value. Initially, the wind is set to 0 Kts (still air), with a wind direction of 360°.

Selection: Select the **Weather** option from the main menu. Select the **Wind Speed** option from the sub-menu. The current wind speed is displayed and can be set to a new value.

Confirmation: There is no visible confirmation, although the effect of a change of wind may become apparent according to the specific wind settings.

Example: In Figure 7.4.2, the wind speed has been set to 23 Kts.

Applications:

- To introduce a specific head-wind during a navigation exercise.

See also: Weather-Wind Dir, Wind-Turbulence.

7.4.3 Weather-Wind Dir

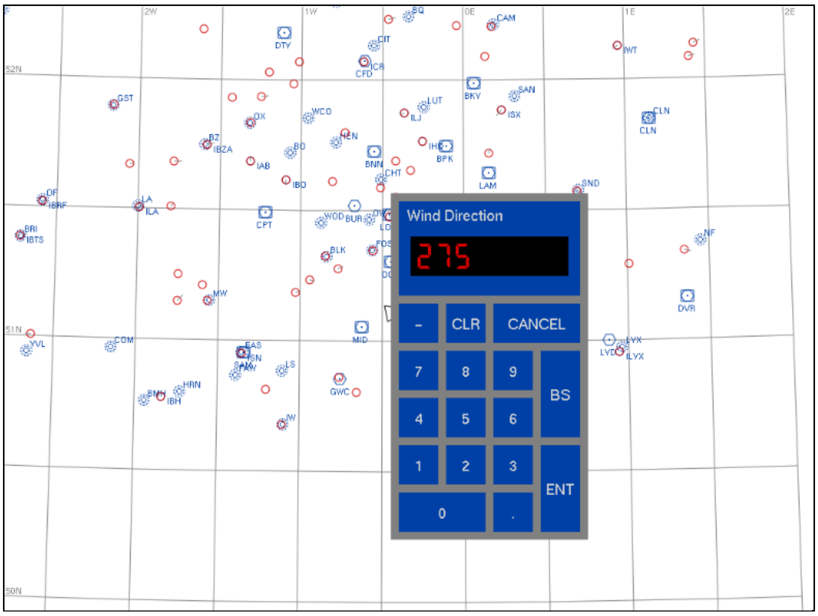


Figure 7.4.3

Function: Set the wind direction in degrees (True).

Effect: The wind direction used by the flight model is set to the selected value. Initially, the wind is set to 0 Kt. (still air), with a wind direction of 360°.

Selection: Select the **Weather** option from the main menu. Select the **Wind Dir** option from the sub-menu. The current wind direction is displayed and can be set to a new value.

Confirmation: There is no visible confirmation, although the effect of a change of wind may become apparent according to the specific wind settings.

Example: In Figure 7.4.3, the wind direction has been set to 275°T.

Applications:

- To introduce a cross-wind during an approach.

See also: Weather-Wind Speed, Weather-Turbulence.

7.4.4 Weather-QNH

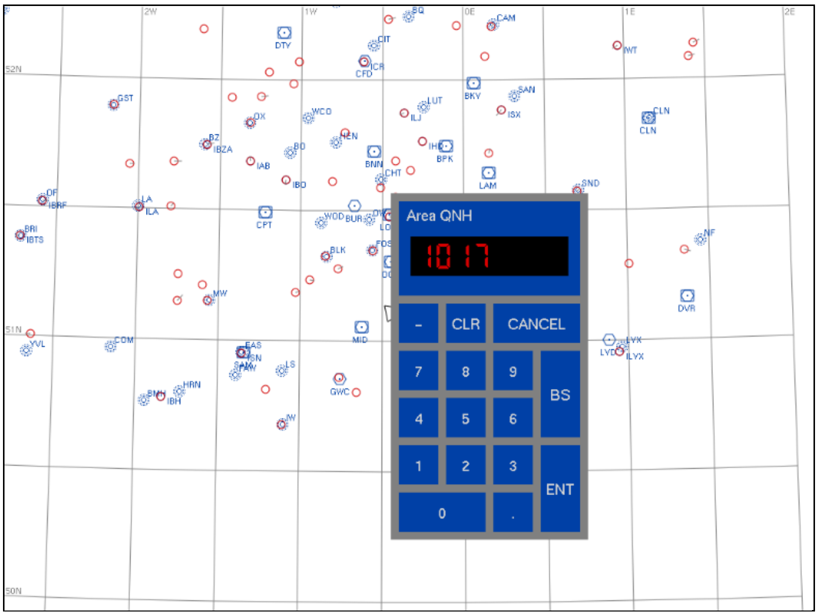


Figure 7.4.4

Function: Set the regional area QNH in HectoPascals (Millibars).

Effect: The area QNH used by the flight model and the aircraft instruments is set to the selected value. Initially, the area QNH is set to 1013 hPa.

Selection: Select the **Weather** option from the main menu. Select the **QNH** option from the sub-menu. The current QNH is displayed and can be set to a new value.

Confirmation: Flight instruments affected by pressure altitude may be altered.

Example: In Figure 7.4.4, the area QNH has been set to 1017 hPa.

Applications:
- To demonstrate the effect of an incorrectly set altimeter.
- To select a specific QNH

See also: Weather-OAT.

7.4.5 Weather-Magnetic Variation

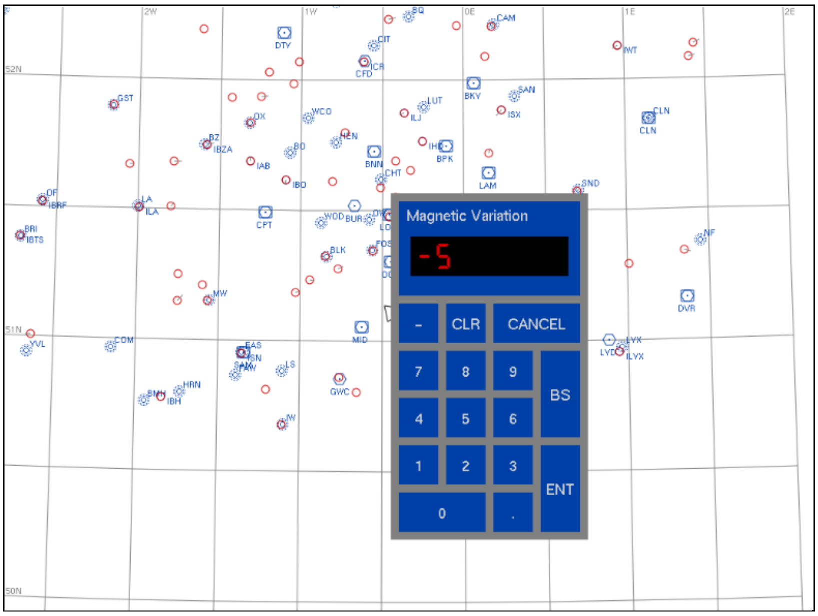


Figure 7.4.5

Function: Set the magnetic variation in degrees.

Effect: The magnetic variation used by the flight model, navigation systems and instrument displays is set to the selected value. Initially, the magnetic variation is set to a typical value for the navigation region (defined in the file /c/sim/files/menu.dat on the instructor station computer).

Selection: Select the **Weather** option from the main menu. Select the **Magnetic Variation** option from the sub-menu. The current magnetic variation is displayed and can be set to a new value. Note that westerly values are entered as negative values and easterly values are entered as positive values. For example, 5° west is -5 and 11° east is +11. There are no checks on consistency of the entered value and the value entered should be compatible with the runway QDM values given in the navigation data.

Confirmation: Flight instruments affected by magnetic variation may be altered. The compass card displayed on the instructor station map display will indicate the selected magnetic variation.

Example: In Figure 7.4.5, the magnetic variation has been set to 5° west.

Applications:

- To illustrate the difference between a magnetic track and true track.
- To select the magnetic variation for a specific region

See also: Weather-QNH

7.4.6 Weather-Cloud Base

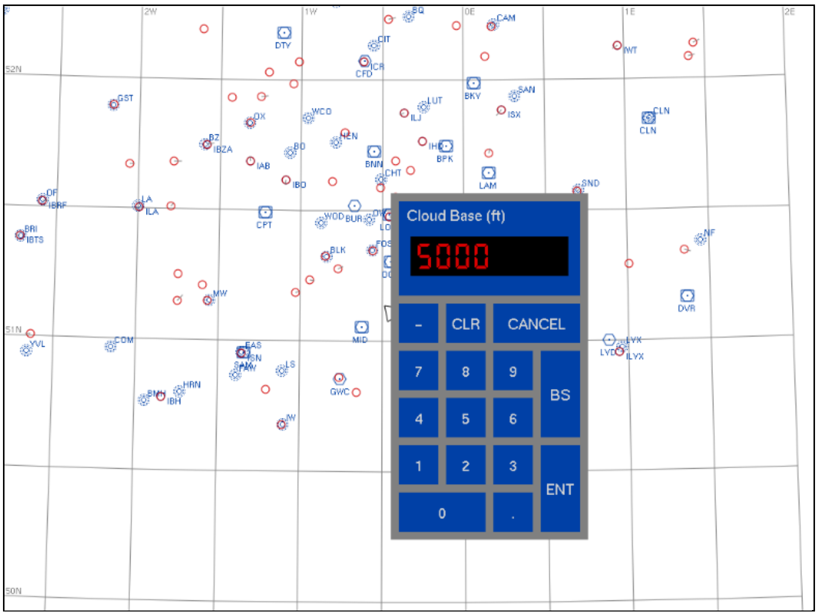


Figure 7.4.6

Function: Set the cloud base in feet above sea level.

Effect: The cloud base used by the visual system is set to a specific value. At this altitude, the aircraft is in IFR conditions. The previously selected value is displayed in the dialogue box. Initially, the cloud base is set to 36000 feet. Note that the value entered is in feet ASL.

Selection: Select the **Weather** option from the main menu. Select the **Cloud Base** option from the sub-menu. The current cloud base is displayed and can be set to a new value.

Confirmation: There is no obvious confirmation of this command other than the state of the visual system.

Example: In Figure 7.4.6, the cloud base has been set to 5000 feet ASL.

Applications:

- To select a cloud base to practise an ILS approach with a known minima.
- To enter cloud shortly after take-off.

See also:
Weather-Visibility, Weather-Wind Speed, Weather-Wind Dir.

7.4.7 Weather-Visibility

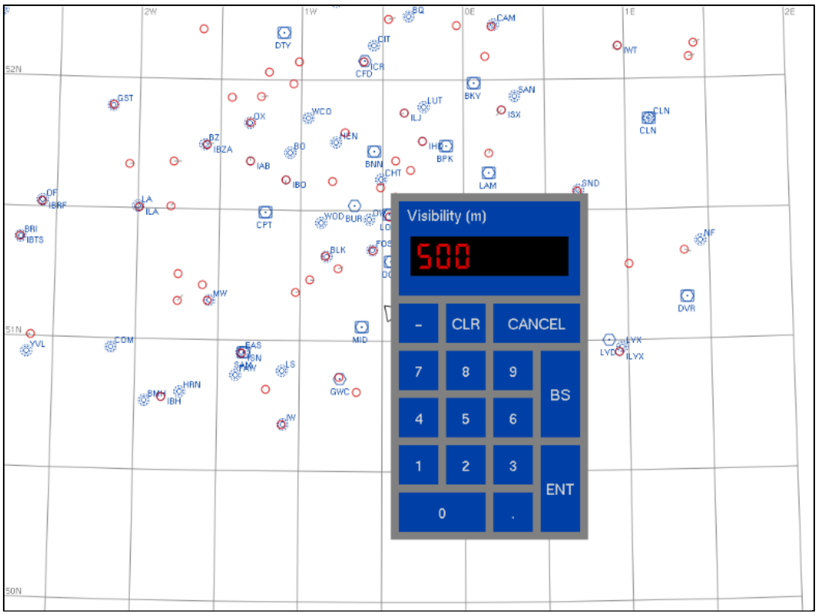


Figure 7.4.7

Function: Set the visual system visibility in metres.

Effect: The visibility is altered to the selected value in metres. The current visibility is displayed in the dialogue box. Initially, the visibility set to 50 Km (50000m).

Selection: Select the **Weather** option from the main menu. Select the **Visibility** option from the sub-menu. The current visibility is displayed and can be set to a new value.

Confirmation: The visibility shown in the visual system is set to the selected distance.

Example: In Figure 7.4.7, the visibility has been set to 500m.

Applications:

- To select a new minima.
- To select conditions for an IFR approach.

See also: Weather-Cloud base.

7.4.8 Weather-OAT

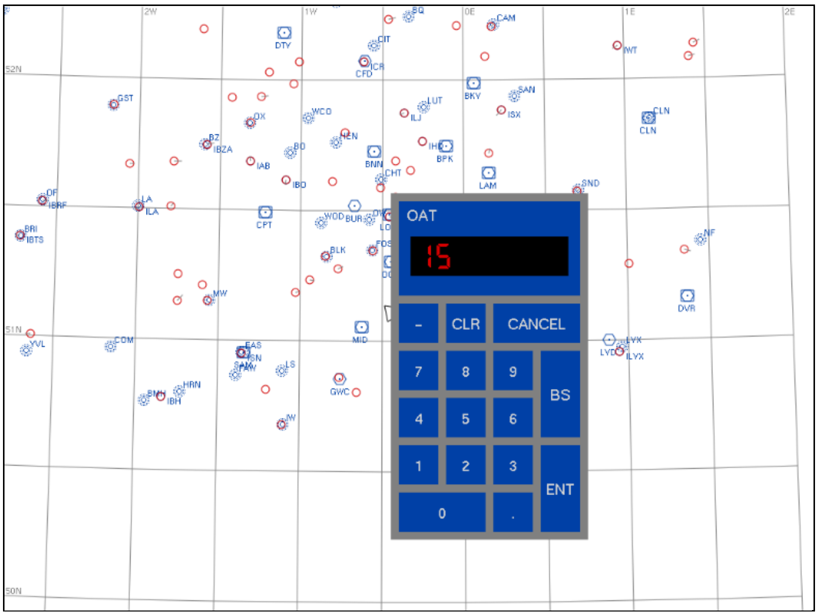


Figure 7.4.8

Function: Set the outside air temperature in degrees centigrade.

Effect: The outside air temperature is altered to the selected value in degrees centigrade. The current temperature is displayed in the dialogue box. Initially, the outside air temperature is based on an International Standard Atmosphere of 15°C at sea-level. Note that the temperature entered is the value at sea-level.

Selection: Select the **Weather** option from the main menu. Select the **OAT** option from the sub-menu. The current temperature is displayed and can be set to a new value.

Confirmation: There is no obvious confirmation of this command other than the overall effects of outside air temperature on the flight model, for example, effects on take-off distance or engine performance at altitude.

Example: In Figure 7.4.8, the outside air temperature has been set to 15°C.

Applications:

- To select a high outside air temperature to illustrate the effect on aircraft performance during take-off.

See also: Weather-Area QNH.

7.4.9 Weather-Time of Day

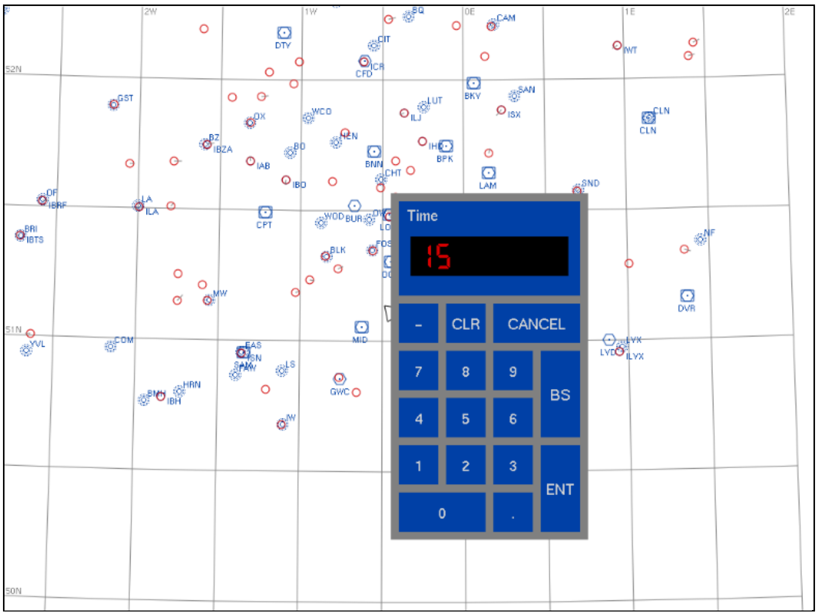


Figure 7.4.9

Function: Set the time of day.

Effect: The visual system light intensity is based on dawn at 06.00 hours and dusk at 17.00 hours with maximum brightness at 12.00. By default, the time of day is set to the IOS PC time.

Note: If the simulator is used before 6am or after 6pm, the visual system will start in a night time mode.

Selection: Select the **Weather** option from the main menu. Select the **Time of Day** option from the sub-menu. The time of day is displayed and can be set to a new value.

Confirmation: The level of brightness is set to the time of day.

Example: In Figure 7.4.9, the time of day is set to 15 (3pm).

Applications:
- To simulate dusk conditions.
- To simulate time of day effects

See also: Weather-Visibility.

7.4.10 Weather-Vis Rate

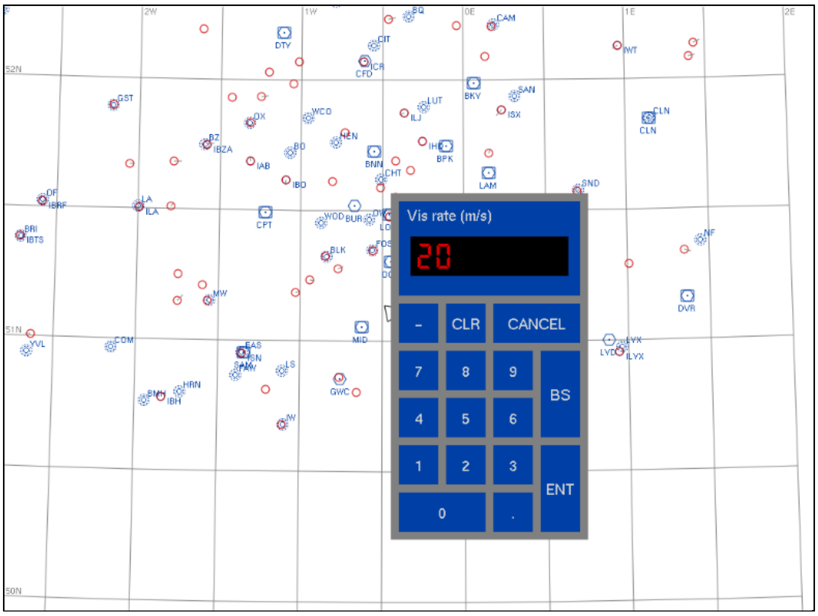


Figure 7.4.10

Function: Set the rate of change of visibility in m/s. A positive value increases visibility and a negative value reduces visibility.

Effect: The visibility in the visual systems will increase or decrease at a defined rate.

Note: To stop the visibility changing, set the Vis-Rate value to zero. The visibility does not change while the simulator is in the HOLD state.

Selection: Select the **Weather** option from the main menu. Select the **Vis Rate** option from the sub-menu. The current value of rate of change of visibility is displayed and can be set to a new value.

Confirmation: The visibility seen in the visual system will start to change.

Example: In Figure 7.4.10, the rate of change of visibility is set to 20. After one minute, the visibility will have increased by 1.2 Km.

Applications:

- To produce degrading visual conditions.
- To vary visibility during specific phases of an exercise.

See also: Weather-Visibility, Weather-Cloud Base.

7.5 Settings

A selection of commands are provided to set aircraft parameters, as shown in Figure 7.5.

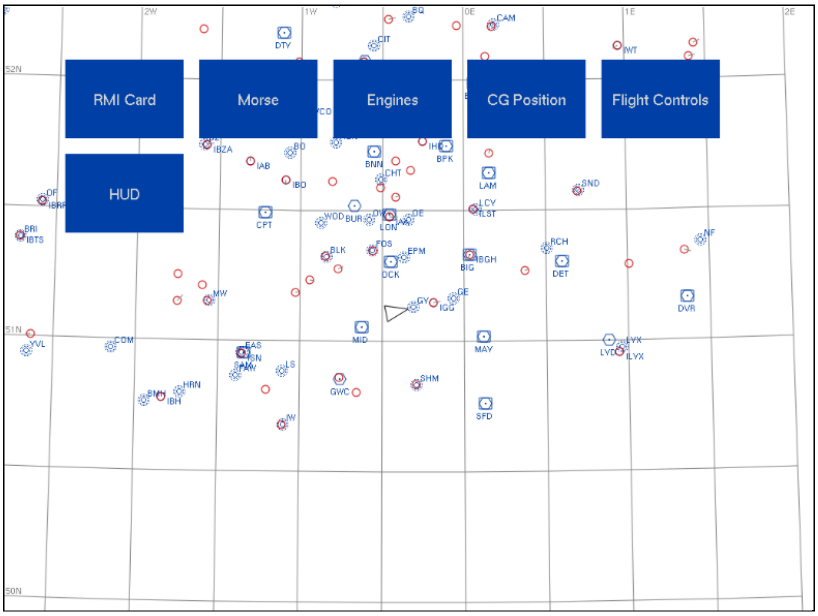


Figure 7.5

The commands characterise or set specific options which alter the basic simulator settings. For light aircraft, the RMI can be set to a fixed or rotating card instrument. Morse ident from navigation beacons can be enabled or disabled. For simulations with multiple engine levers, an option is provided to control the engine from a single lever. The centre of gravity of the aircraft can be changed. The aircraft can be flown from the centre stick or the side stick. For the visual system, a 2D HUD can be overlaid on the outside scene.

7.5.1 Settings-RMI Card

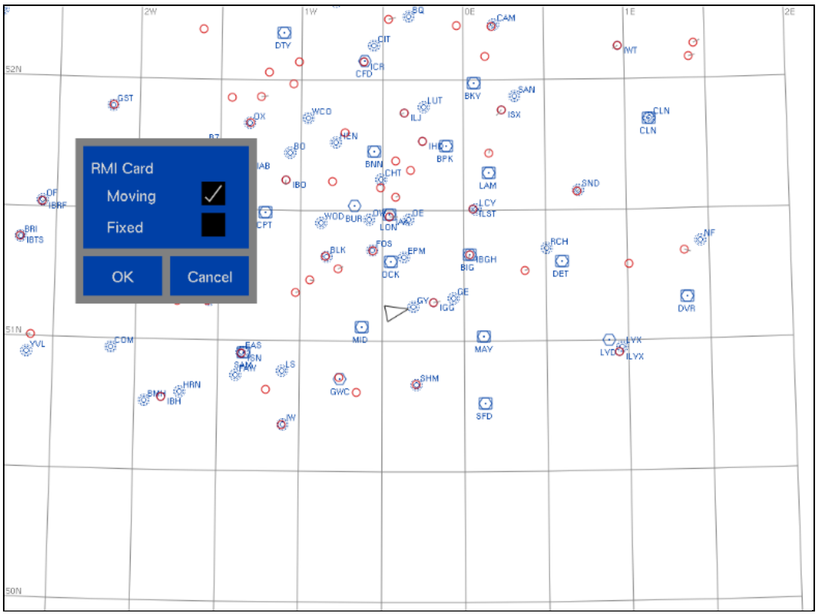


Figure 7.5.1

Function: The instructor can select the ADF (or RMI) to operate as either a fixed card or a moving card instrument (i.e. as an RBI or an RMI respectively).

Effect: The ADF/RMI is set to either a fixed card instrument or a moving card instrument. In the RBI mode, the ADF/RMI card is constantly aligned to North.

Selection: Select the **Settings** option from the main menu. Select the **RMI Card** option from the sub-menu. Select the radio button for either Fixed or Moving.

Confirmation: The RMI Card is set either to a fixed card instrument (RBI) or to a moving card instrument, automatically aligned (slaved) to magnetic north (RMI).

Example: In Figure 7.5.1, the ADF has been set to a moving card ADF (or RMI).

Applications:

- To demonstrate the relationship between an RMI or an RBI ADF instrument.
- To select an ADF instrument appropriate to a specific aircraft.

See also:

7.5.2 Settings-Morse

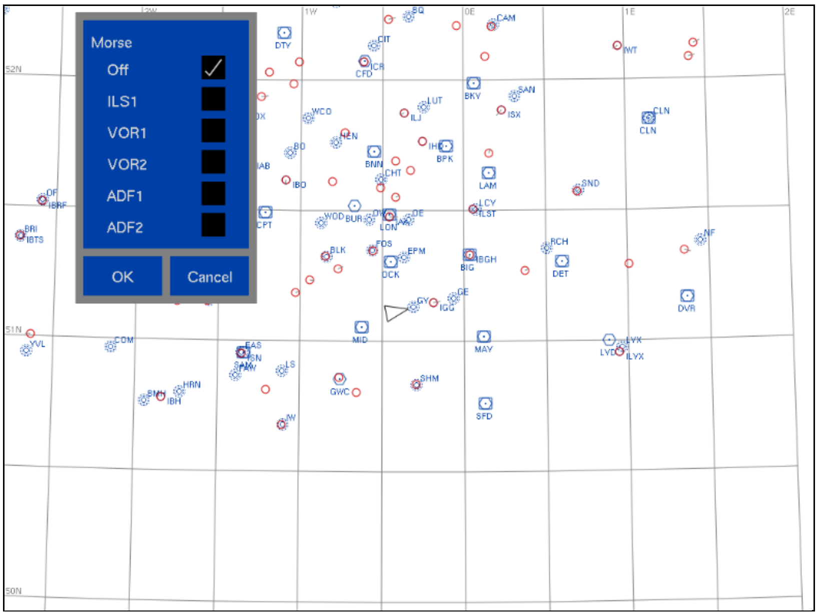


Figure 7.5.2

Function: The instructor can enable specific Morse code idents or suppress the aircraft Morse audio output. By default all Morse idents are turned off (to minimise distraction).

Effect: The Morse idents provided by the sound generation sub-system can be switched on or off.

Selection: Select the **Settings** option from the main menu. Select the **Morse** option from the sub-menu. Select the appropriate radio button to enable or disable specific Morse idents.

Confirmation: The Morse ident will be activated or suppressed.

Example: In Figure 7.5.2, the Morse channels have been turned off.

Applications: To fail the Morse in order to check that the pilot correctly tunes and identifies stations.

See also:

7.5.3 Settings-Engines

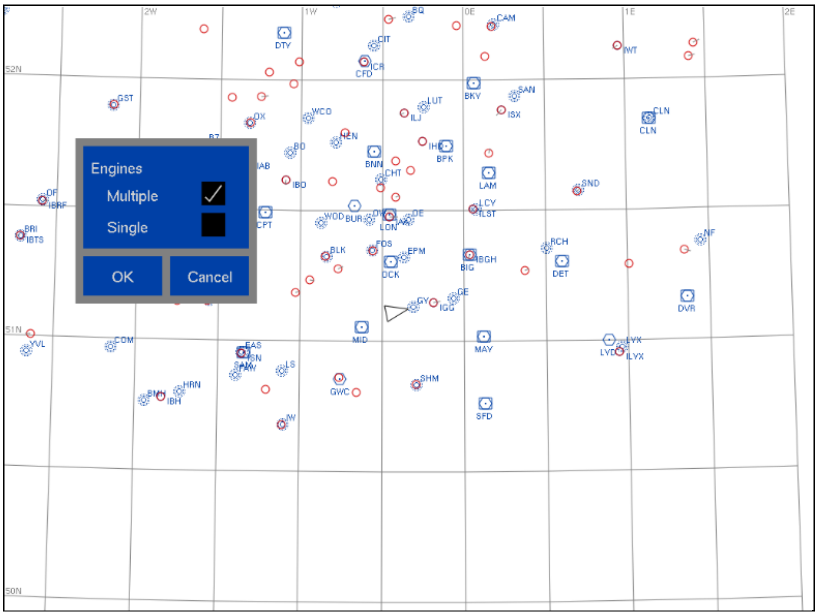


Figure 7.5.3

Function: The instructor can select a mode to allow the pilot to operate all the aircraft engines from a single lever. This command only applies to twin and multi-engine aircraft flight models.

Effect: The aircraft engines respond to the left-most lever (or set of levers for propeller pitch and mixture controls).

Selection: Select the **Settings** option from the main menu. Select the **Engines** option from the sub-menu. Select the radio button for Single or Multiple.

Confirmation: The aircraft engine model recognises the left-most lever as the control lever for all engines in the Single mode. In the Multiple mode, the levers operate normally.

Example: In Figure 7.5.3, the engine operating mode has been set to multiple mode.

Applications:

- To suppress asymmetric effects during a twin-engine aircraft exercise.

See also:

7.5.4 Settings-CG Position

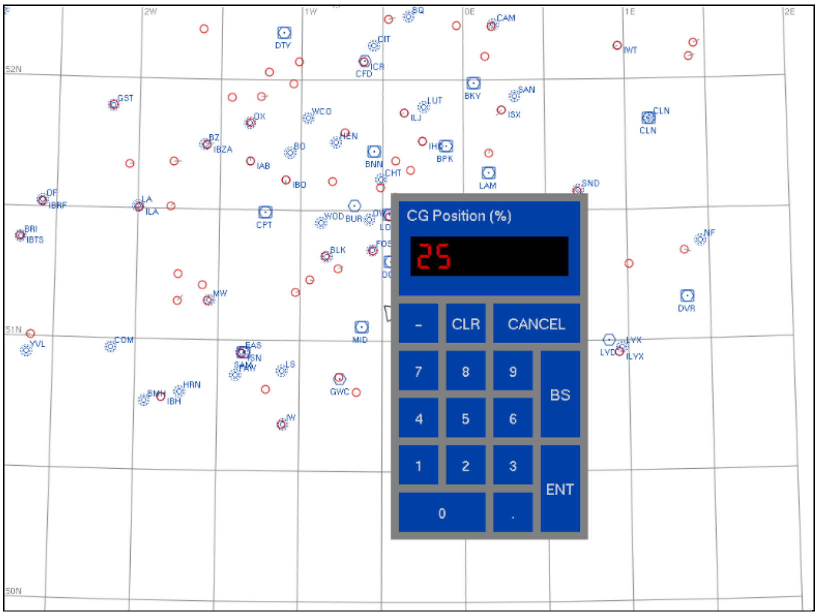


Figure 7.5.4

Function: Set the aircraft centre of gravity as a function of the percentage of wing mean chord.

Effect: The position of the aircraft centre of gravity is set to the selected value as a percentage of the wing mean chord.

Selection: Select the **Settings** option from the main menu. Select the **CG Position** option from the sub-menu. The current value of the centre of gravity is displayed and can be set to a new value.

Confirmation: There is no visible confirmation, although the effect of a change of CG position will affect the aircraft handling and stability.

Example: In Figure 7.5.4, the centre of gravity has been set to 25 per cent (the default value).

Applications:
- To simulate the effect of aircraft loading.
- To change the aircraft handling characteristics

See also: Settings-Flight Controls

7.5.5 Settings-Flight Controls

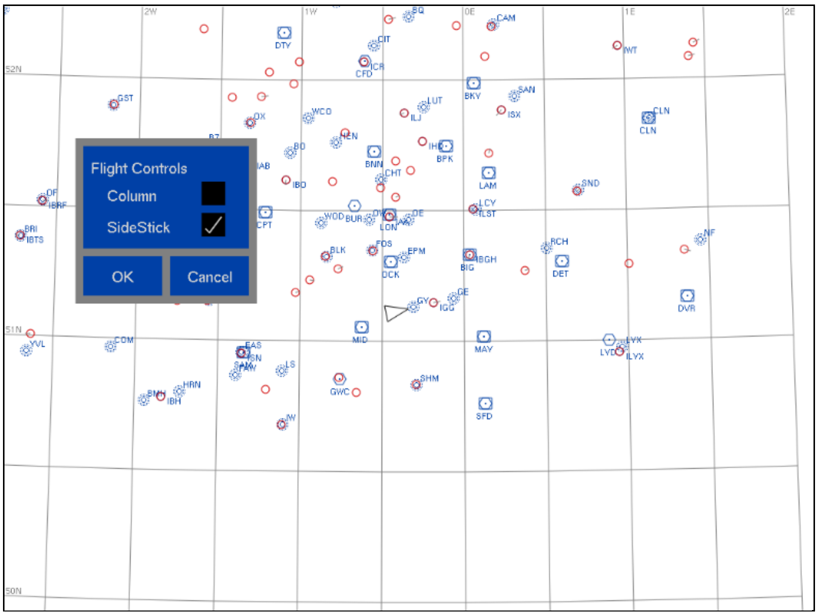


Figure 7.5.5

Function: An option is provided to use either the control column or side-stick.

Effect: Pilot inputs, including trimming functions, are taken from the side-stick or the control column. By default, pilot inputs are taken from the control column.

Selection: Select the **Settings** option from the main menu. Select the **Flight Controls** option from the sub-menu. The setting is displayed and can be set to a new value, by selection of the radio buttons.

Confirmation: Pilot control is taken from either the side-stick or control column, according to the settings.

Example: In Figure 7.5.5, pilot input is taken from the side-stick.

Applications:

- To enable the side-stick pilot control.

See also:

7.5.6 Settings-HUD

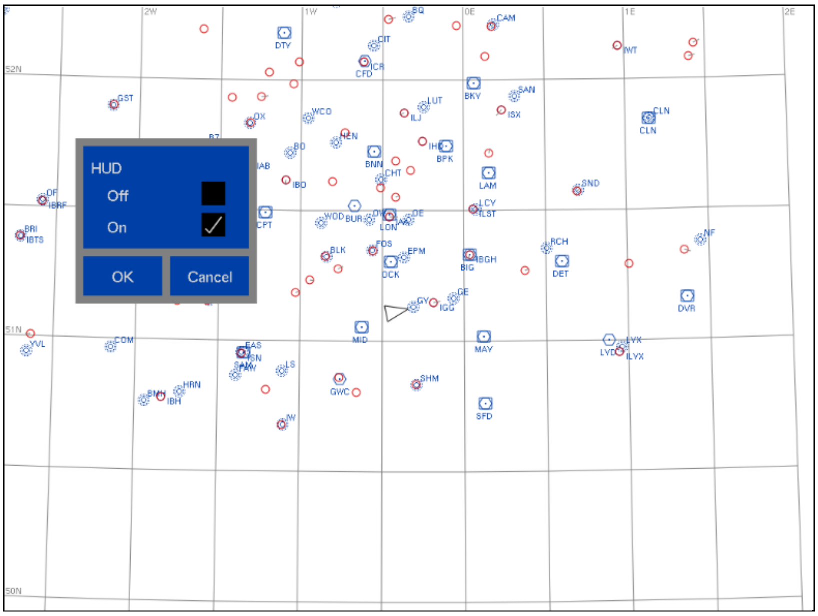


Figure 7.5.6a

Function: A head-up display (HUD) is projected on the outside scene by the centre channel IG, as shown below. The HUD is positioned in the visual display corresponding to the location of HUD mounted in front of the pilot. The HUD format is based on a set of C libraries in the IG system.

Effect: Disable or enable the HUD. The effect is apparent instantaneously in the visual system.

Selection: Select the **Settings** option from the main menu. Select the **HUD** option from the sub-menu. The setting is displayed and the HUD can be turned on or off by selection of the radio buttons.

Confirmation: The HUD is displayed (or removed).

Example: In Figure 7.5.6a the HUD is selected (turned on).

Applications:

- To simulate an aircraft with a HUD, as shown in Figure 7.5.6b.

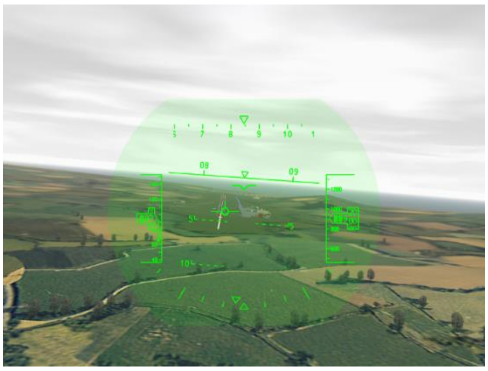


Figure 7.5.6b

See also:

7.6 Failures

This menu option allows the instructor to fail aircraft systems such as gear, flaps, DME, localiser, glide slope indicator and engines. The selected aircraft sub-system can be set to an operational or failed (malfunction) state. The failures also include instruments. Note that the state of an instrument or sub-system can be examined simply by selecting the appropriate instrument or sub-system and then clicking on the CANCEL button. The failure options are shown in Figure 7.6.

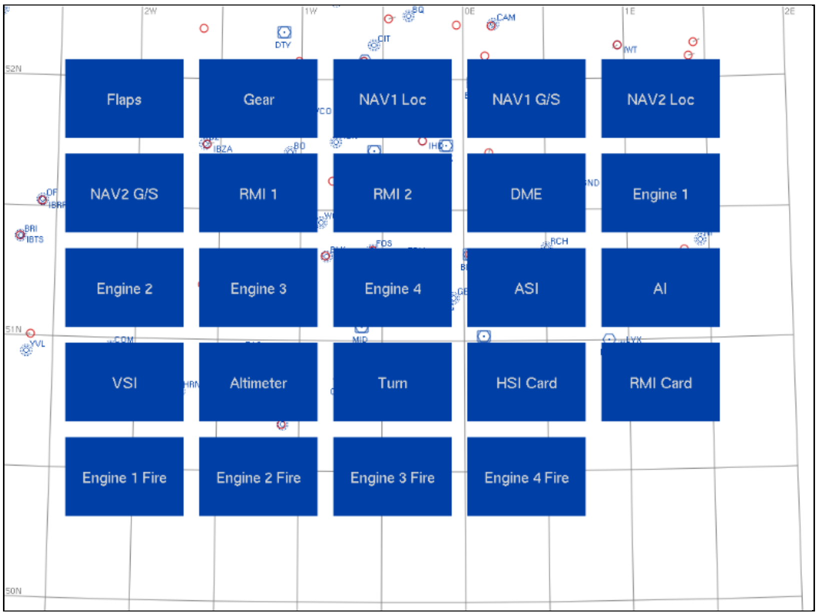


Figure 7.6

The format of the command is common for the 24 selections. For reasons of brevity, only the flaps failure is illustrated overleaf. For each selection, the user is prompted to set or fail a system and can also cancel the selection.

The sub-systems that can be failed are as follows:

Flaps	The flaps are failed in their current position and will not respond to movement of the flap selection lever. Typically, the flaps can be failed fully up or fully down.
Gear	The gear is failed in its current position and will not respond to movement of the undercarriage lever. The gear can be failed fully up, fully down or in transit. If the gear is failed in transit, the gear will not display the green 'gear down' lights.
Nav1 Loc	The NAV1 localiser is failed, causing the NAV flag to drop on the HSI or VOR instruments.
Nav1 G/S	The NAV1 glide slope is failed, causing the GS flag to drop on the HSI or VOR instruments.
Nav2 Loc	The NAV2 localiser is failed, causing the NAV flag to drop on the HSI or VOR

instruments.

Nav2 G/S — The NAV2 glide slope is failed, causing the GS flag to drop on the HSI or VOR instruments.

RMI 1 — The RMI single needle is failed. The needle will no longer point to the station selected for ADF-1 or NAV-1.

RMI 2 — The RMI double needle is failed. The needle will no longer point to the station selected for ADF-2 or NAV-2.

DME — The DME (and ground-speed indicator, if appropriate) is failed. The DME flags drop to obscure the DME read-out.

Engine 1 — Engine number 1 is failed. For a single engine aircraft, the engine is failed. For a twin engine aircraft, the port engine is failed. For a four-engine aircraft, the port outer engine is failed. The failure is equivalent to stopping the fuel flow to the engine. It may be necessary to restart a failed engine in flight by application of the engine start switch.

Engine 2 — Engine number 2 is failed. For a single engine aircraft, the command is ignored. For a twin engine aircraft, the starboard engine is failed. For a four-engine aircraft, the port inner engine is failed. The failure is equivalent to stopping the fuel flow to the engine. It may be necessary to restart a failed engine in flight by application of the engine start switch.

Engine 3 — Engine number 3 is failed. For a single engine aircraft or twin engine aircraft, the command is ignored. For a four-engine aircraft, the starboard inner engine is failed. The failure is equivalent to stopping the fuel flow to the engine. It may be necessary to restart a failed engine in flight by application of the engine start switch.

Engine 4 — Engine number 4 is failed. For a single engine aircraft or twin engine aircraft, the command is ignored. For a four-engine aircraft, the starboard outer engine is failed. The failure is equivalent to stopping the fuel flow to the engine. It may be necessary to restart a failed engine in flight by application of the engine start switch.

ASI — The airspeed indicator is failed. It maintains its current indication to give the effect of a damaged pitot tube, an iced-up pitot tube or a blocked static system.

AI — The attitude indicator is failed. It maintains its current indication in terms of pitch and bank to give the effect of failed suction.

VSI — The vertical speed indicator is failed. It maintains its current indication to give the effect of a blocked static system.

Altimeter	The altimeter is failed. It maintains its current indication to give the effect of a blocked static system.
Turn	The turn indicator is failed. It maintains its current indication to give the effect of an electrical failure.
HSI card	The HSI card fails to rotate to give the effect of a failed gyro or loss of compass slaving.
RMI card	The RMI card fails to rotate to give the effect of a failed gyro or loss of compass slaving.
Engine 1 Fire	The fire alarm for engine 1 can be activated or deactivated.
Engine 2 Fire	The fire alarm for engine 2 can be activated or deactivated.
Engine 3 Fire	The fire alarm for engine 3 can be activated or deactivated.
Engine 4 Fire	The fire alarm for engine 4 can be activated or deactivated.

7.6.1 Failures-Flaps

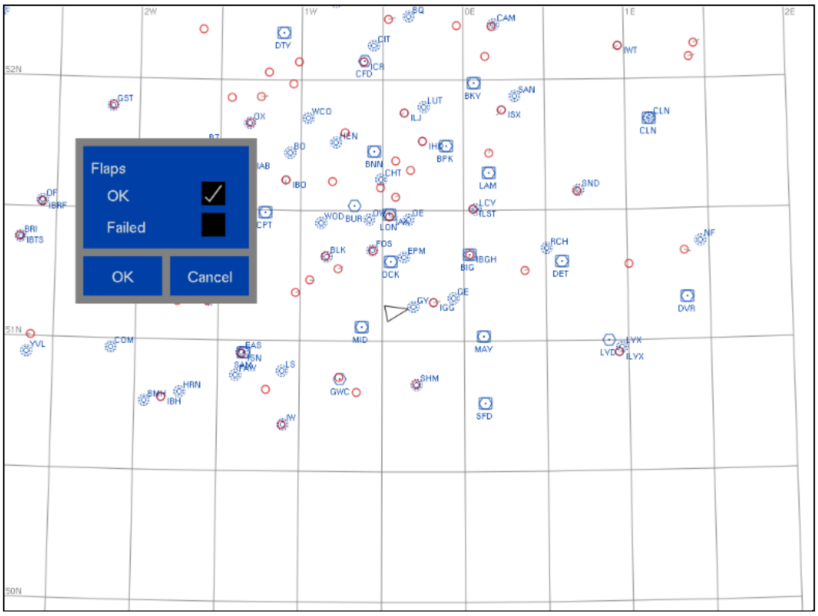


Figure 7.6.1

Function: The instructor is able to fail specific aircraft systems and instruments. Aircraft flap failure is shown above.

Effect: The selected aircraft system or instrument is set to either an operational or failed state. For example, this command allows the instructor to simulate limited panel IFR conditions or to initiate system failures.

Selection: Select the **Failures** option from the main menu. Select the appropriate system or instrument option from the sub-menu. For each sub-option, radio buttons are provided to select either OK (operational) or a failed state. In addition, an inadvertent selection can be cancelled.

Confirmation: The selected aircraft system or instrument will fail to function correctly, or alternatively, a failed sub-system will start to function normally.

Options: The actual options vary for the range of flight models. For example, the number of engines, number of VORs and retractable landing gear define the sub-menu format.

Example: In Figure 7.6.1, the aircraft flaps are set to operate correctly.

Applications:

- To introduce instrument failures to simulate limited panel conditions.
- To induce failures to increase pilot workload and to check flight crew procedures.

See also:

7.7 Autopilot

A selection of commands are provided to set autopilot modes from the instructor station. These selections are independent of any selections made by the flight crew or the flight management system. Engaging or disengaging a flight control mode at the instructor station is equivalent to setting the flight control modes set on the FCU. The autopilot modes and settings are shown in Figure 7.7.

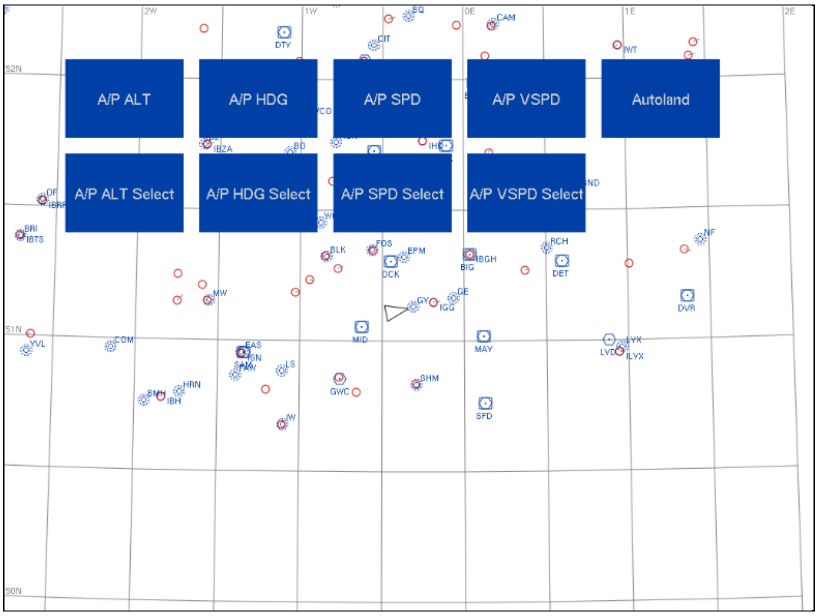


Figure 7.7

These commands are also useful to test flight control laws allowing flight control laws to be engaged from the instructor station.

7.7.1 Autopilot-A/P ALT

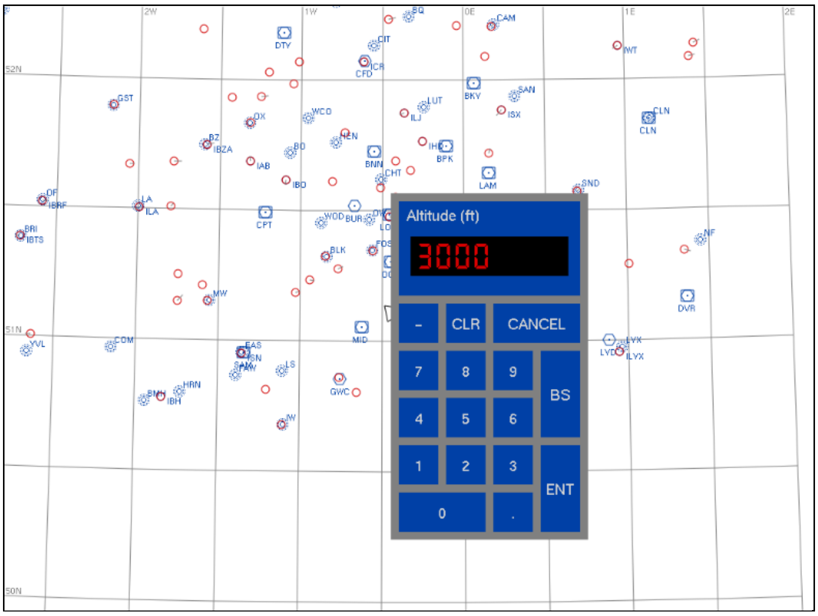


Figure 7.7.1

Function: Set the autopilot altitude to feet ASL.

Effect: For versions of the simulator where autopilot functions are supported, the altitude setting of the autopilot is set to the selected value in feet ASL.

Selection: Select the **Autopilot** option from the main menu. Select the **A/P ALT** option from the sub-menu. The current setting of the autopilot altitude is displayed and can be set to a new value.

Confirmation: The new autopilot altitude is indicated on the aircraft altimeters. Note that the altitude indicated on the altimeters includes the barometric pressure settings.

Example: In Figure 7.7.1, the autopilot altitude hold has been set to 3000 feet.

Applications:

- To select altitude hold in order to reduce the pilot workload during an ADF hold pattern.

See also: Autopilot-HDG, Autopilot-SPD, Autopilot-VSPD.

7.7.2 *Autopilot-A/P HDG*

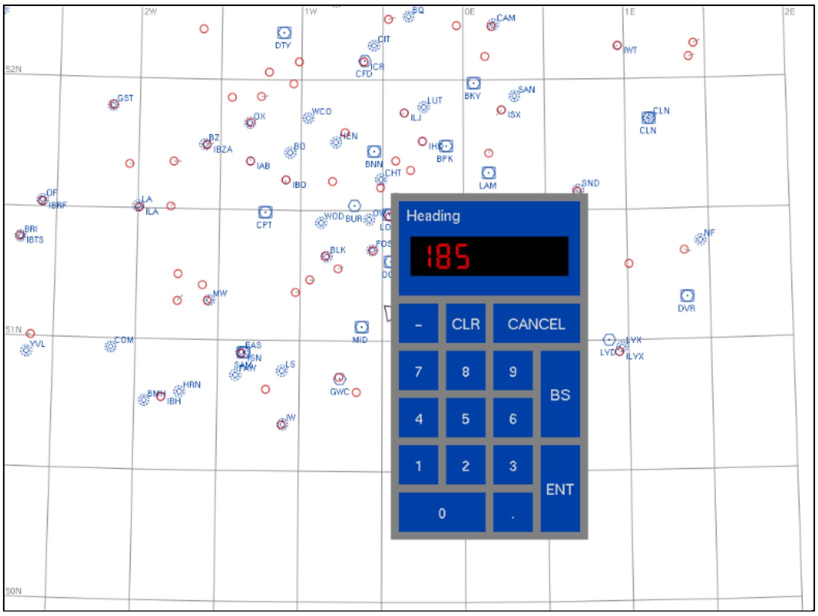


Figure 7.7.2

Function: Set the autopilot heading in degrees magnetic.

Effect: For versions of the simulator where autopilot functions are supported, the magnetic heading setting of the autopilot is set to the selected value in degrees.

Selection: Select the **Autopilot** option from the main menu. Select the **A/P HDG** option from the sub-menu. The current setting of the autopilot heading is displayed in degrees magnetic and can be set to a new value.

Confirmation: The new autopilot heading is indicated on the aircraft magnetic compass, DI, HSI and RMI. The value transferred is the magnetic heading in degrees.

Example: In Figure 7.7.2, the autopilot heading hold has been set to 175 degrees (magnetic).

Applications:

- To 'fly' the aircraft on a constant heading to illustrate an intercept to a VOR radial.
- To align the aircraft with the runway QDM.

See also: Autopilot-ALT, Autopilot-SPD, Autopilot-VSPD.

7.7.3 Autopilot-A/P SPD

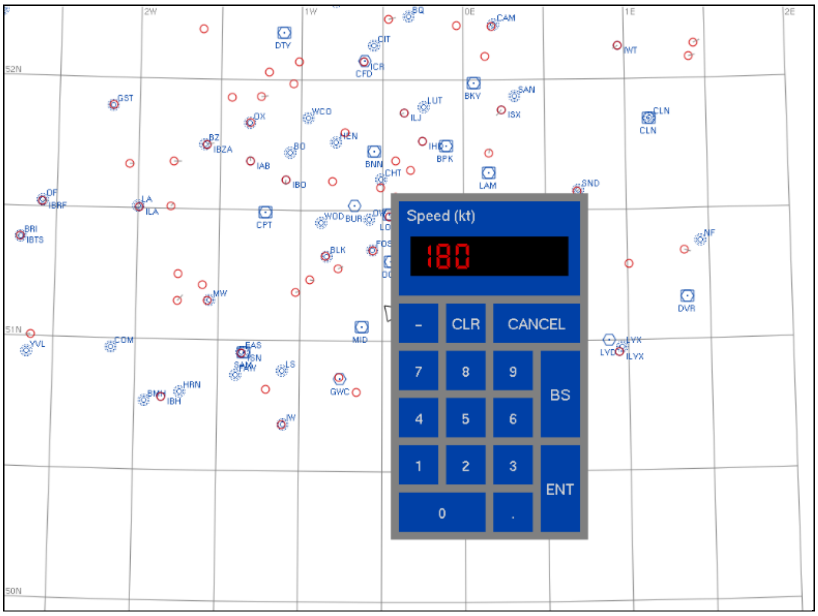


Figure 7.7.3

Function: Set the autopilot (or auto-throttle) speed in Kts.

Effect: For versions of the simulator where autopilot functions are supported, the autopilot speed setting is set to the selected value in Kts.

Selection: Select the **Autopilot** option from the main menu. Select the **A/P SPD** option from the sub-menu. The current setting of the autopilot speed is displayed in Kts and can be set to a new value.

Confirmation: A new speed is indicated on the airspeed indicator and Mach meter (if appropriate).

Example: In Figure 7.7.3 the auto-throttle speed hold has been set to 180 Kts.

Applications:

- To select a speed for an ILS approach reducing the overall workload during initial instruction in ILS operation.

See also: Autopilot-ALT, Autopilot-HDG, Autopilot-VSPD.

7.7.4 Autopilot-A/P VSPD

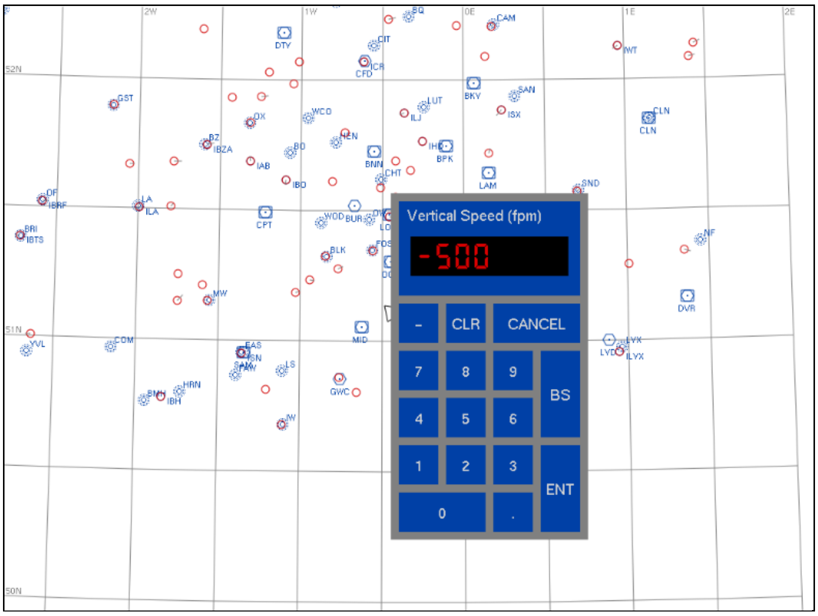


Figure 7.7.4

Function: Set the autopilot vertical speed in feet per minute.

Effect: For versions of the simulator where autopilot functions are supported, the autopilot vertical speed setting is set to the selected value in ft/min.

Selection: Select the **Autopilot** option from the main menu. Select the **A/P VSPD** option from the sub-menu. The current setting of the autopilot heading is displayed in Kts and can be set to a new value.

Note: A positive value is used for climbing flight and a negative value is used for descending flight.

Confirmation: A new vertical speed is indicated on the airspeed vertical speed indicator.

Example: In Figure 7.7.4 the autopilot vertical speed hold has been set to a descent rate of 500 fpm.

Applications:
- To select a vertical speed during a climb phase.
- To expedite a descent during an approach.

See also: Autopilot-ALT, Autopilot-HDG, Autopilot-SPD.

7.7.5 A/P Autoland

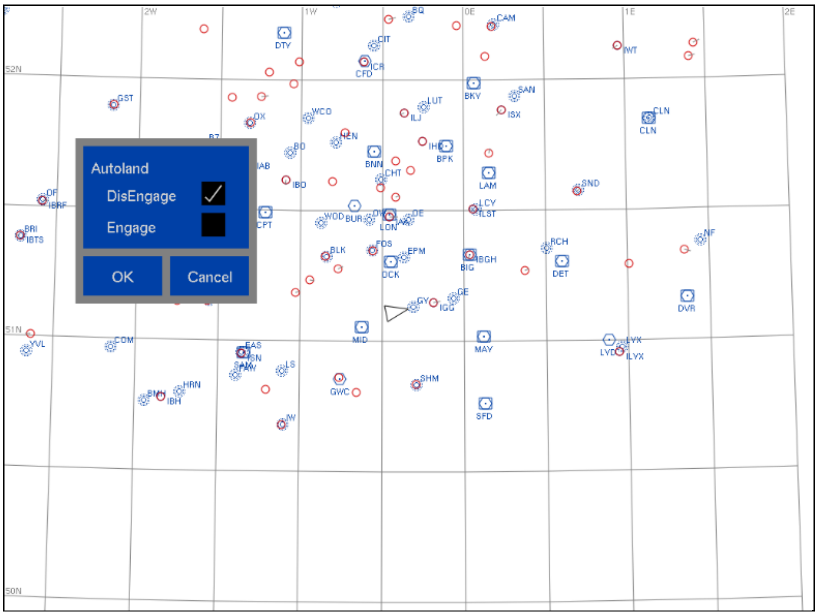


Figure 7.7.5

Function: Engage/disengage the autopilot auto-land mode.

Effect: For versions of the simulator where autopilot functions are supported, the autopilot auto-land mode is engaged or disengaged.

Selection: Select the **Autopilot** option from the main menu. Select the **A/P Autoland** option from the sub-menu. Select the appropriate radio button to engage or disengage the autopilot auto-land function.

Confirmation: The autopilot auto-land is engaged or disengaged, as appropriate.

Example: In Figure 7.7.5, the autopilot auto-land mode has been disengaged.

Applications: To replicate autopilot functions or to reduce pilot workload during training exercises.

See also: Autopilot-A/P HDG Select, Autopilot-A/P ALT Select, Autopilot-A/P SPD Select

7.7.6 A/P ALT Select

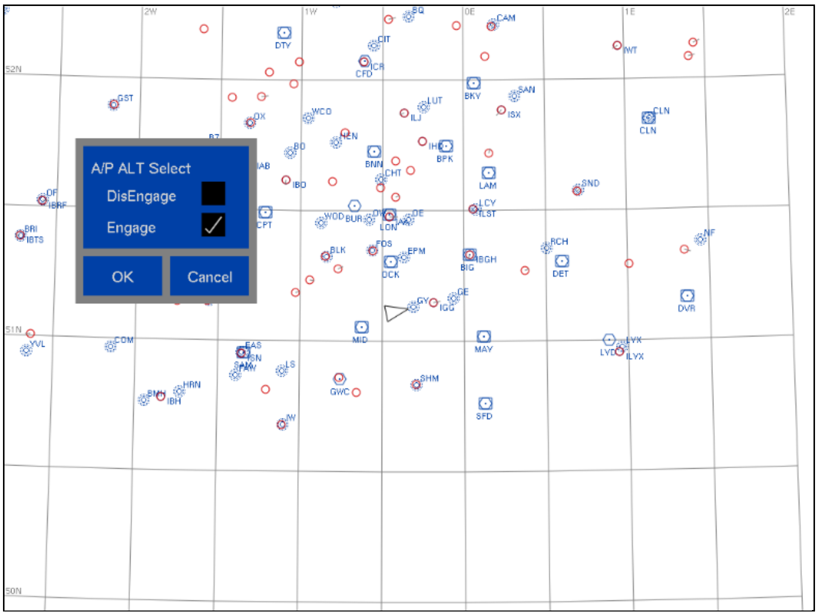


Figure 7.7.6

Function: Engage/disengage the autopilot altitude hold selection.

Effect: For versions of the simulator where autopilot functions are supported, the autopilot altitude selection is engaged or disengaged.

Selection: Select the **Autopilot** option from the main menu. Select the **A/P ALT Select** option from the sub-menu. Select the appropriate radio button to engage or disengage the autopilot altitude hold function.

Confirmation: The autopilot altitude hold is engaged or disengaged, as appropriate.

Example: In Figure 7.7.6, the autopilot altitude hold has been engaged.

Applications: To replicate autopilot functions or to reduce pilot workload during training exercises.

See also: Autopilot-A/P HDG Select, Autopilot-A/P SPD Select, Autopilot-Autoland

7.7.7 Autopilot-A/P HDG Select

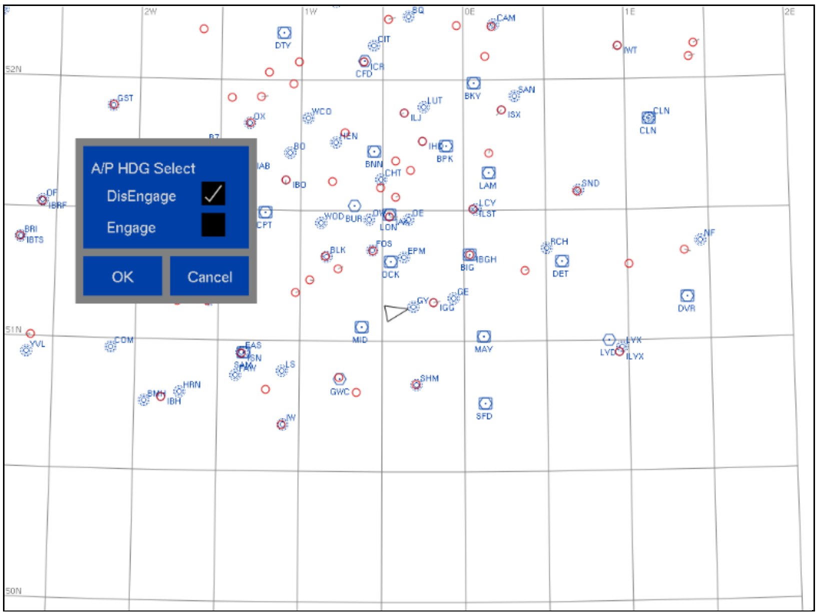


Figure 7.7.7

Function: Engage/disengage the autopilot heading hold selection.

Effect: For versions of the simulator where autopilot functions are supported, the autopilot heading selection is engaged or disengaged.

Selection: Select the **Autopilot** option from the main menu. Select the **A/P HDG Select** option from the sub-menu. Select the appropriate radio button to engage or disengage the autopilot heading hold function.

Confirmation: The autopilot heading hold is engaged or disengaged, as appropriate.

Example: In Figure 7.7.7, the autopilot heading hold has been disengaged.

Applications: To replicate autopilot functions or to reduce pilot workload during training exercises.

See also: Autopilot-A/P ALT Select, Autopilot-A/P SPD Select, Autopilot-Autoland

7.7.8 Autopilot-A/P SPD Select

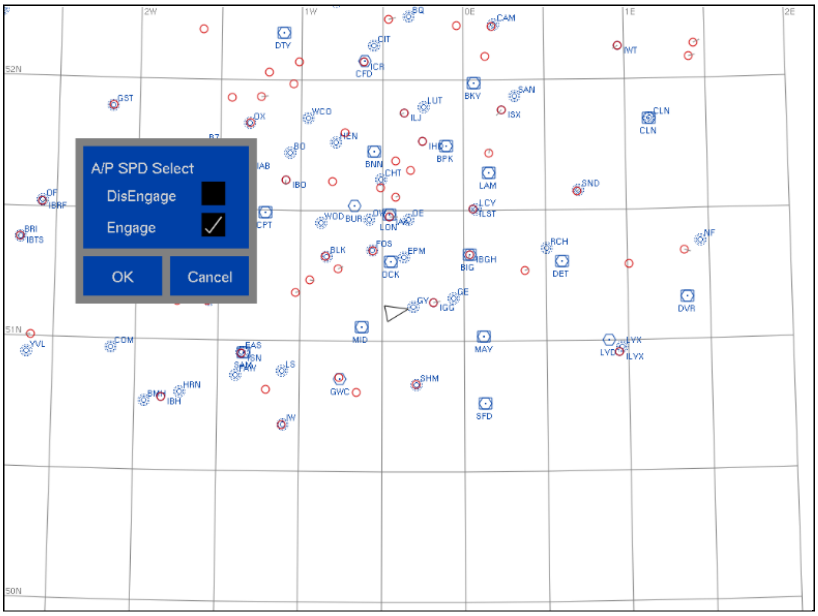


Figure 7.7.8

Function: Engage/disengage the autopilot speed hold selection.

Effect: For versions of the simulator where autopilot functions are supported, the autopilot speed hold selection is engaged or disengaged.

Selection: Select the **Autopilot** option from the main menu. Select the **A/P SPD Select** option from the sub-menu. Select the appropriate radio button to engage or disengage the autopilot speed hold function.

Confirmation: The autopilot speed hold is engaged or disengaged, as appropriate.

Example: In Figure 7.7.8, the autopilot speed hold has been engaged.

Applications: To replicate autopilot functions or to reduce pilot workload during training exercises.

See also: Autopilot-A/P ALT Select, Autopilot-A/P HDG Select, Autopilot-Autoland

7.8 Flight Plan

Three commands are provided to manage a simple flight or to activate a flight plan, as shown in Figure 7.7. A flight plan can be loaded from a file in the user directory with a .pln extension. The current flight plan can be activated or deactivated. A script can be loaded from a file in the user directory with a .scr extension. The details of writing and using script files is described in the document 'The *Script* Language'

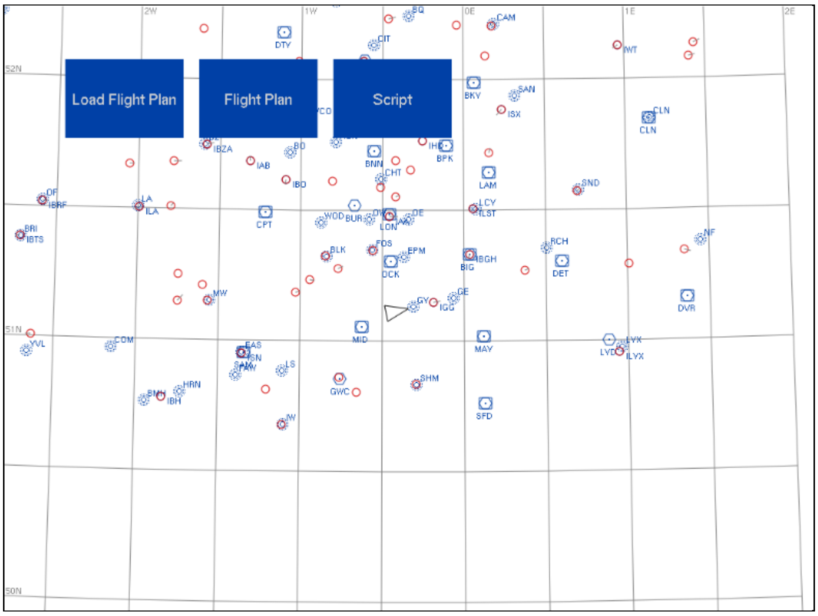


Figure 7.8

A typical flight plan file is as follows:

```
WPT1 S2614.00 E15123.67 5500 96
WPT2 S2616.00 E15125.67 5500 96
WPT3 S2617.00 E15123.67 5500 96
WPT4 S2620.00 E15121.67 5500 96
WPT5 S2622.00 E15123.67 5500 96
WPT6 S2620.00 E15125.67 5500 96
WPT7 S2617.00 E15123.67 5500 105
WPT8 S2616.00 E15121.67 5500 105
WPT1 S2614.00 E15123.67 5500 105
WPT2 S2616.00 E15125.67 5500 105
WPT3 S2617.00 E15123.67 5500 105
WPT4 S2620.00 E15121.67 6500 90
WPT5 S2622.00 E15123.67 6500 90
WPT6 S2620.00 E15125.67 6500 90
WPT7 S2617.00 E15123.67 6500 90
WPT8 S2616.00 E15121.67 6500 90
WPT1 S2614.00 E15123.67 6500 90
WPT2 S2616.00 E15125.67 6500 90
WPT3 S2617.00 E15123.67 6500 90
WPT4 S2620.00 E15121.67 6500 90
WPT5 S2622.00 E15123.67 6500 90
WPT6 S2620.00 E15125.67 6500 90
WPT7 S2617.00 E15123.67 6500 90
WPT8 S2616.00 E15121.67 3000 96
```

The first column denotes the waypoint name. The next two columns denote the waypoint latitude and longitude. The fourth column is the altitude to be flown for the flight plan segment (ft ASL). The fifth column is the airspeed to be flown for the flight plan segment (Kts).

7.8.1 Flight Plan-Load Flight Plan

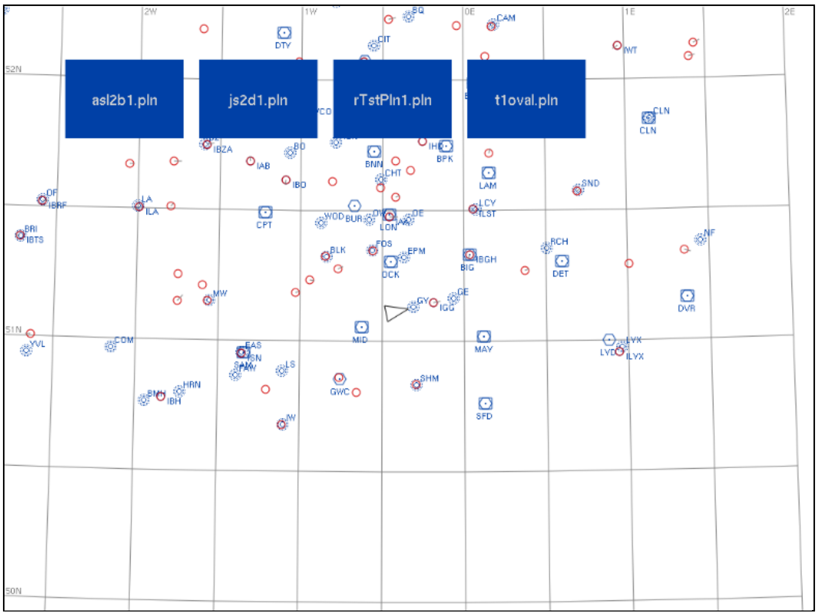


Figure 7.8.1

Function: Load a flight plan.

Effect: A flight plan file is loaded from the user directory.

Selection: Select the **Flight Plan** option from the main menu. Select the **Load Flight Plan** option from the sub-menu. A list of flight plans is displayed. Select a flight plan from the list of flight plans.

Confirmation: The flight plan is loaded. Any syntax errors are reported and if errors are encountered, the execution of the flight plan is abandoned. Note that the flight plan is not activated.

Example: In Figure 7.8.1, four flight plans are displayed.

Applications:

- To demonstrate the operation of a flight management system.
- To assess the performance of flight guidance laws.
- To provide an interface for prototype flight management systems.
- To fly complex route patterns.

See also: Flight Plan-Flight Plan

7.8.2 Flight Plan-Flight Plan

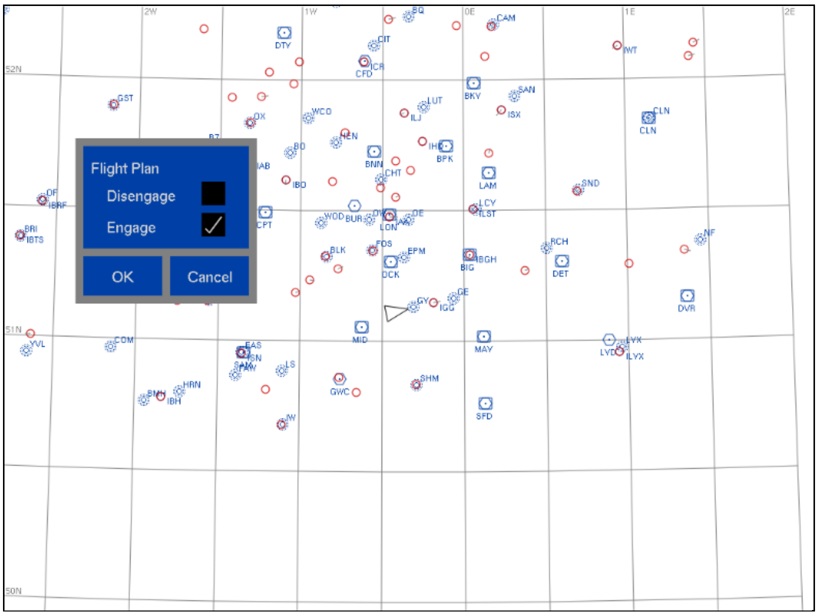


Figure 7.8.2

Function: Engage or disengage a flight plan.

Effect: The currently loaded flight plan file is engaged or disengaged. The command is ignored if no plan is currently loaded.

Selection: Select the **Flight Plan** option from the main menu. Select the **Flight Plan** option from the sub-menu. Select engage or disengage from the displayed options.

Confirmation: The flight plan is engaged or disengaged. If engaged, the flight simulator will execute the flight until it is disengaged or reaches the end of the final flight plan segment.

Example: In Figure 7.8.2, the flight plan is engaged.

Applications:

- To demonstrate the operation of a flight management system.
- To assess the performance of flight guidance laws.
- To provide an interface for prototype flight management systems.
- To fly complex route patterns.

See also: Flight Plan-Load Flight Plan

7.8.3 Flight Plan-Script

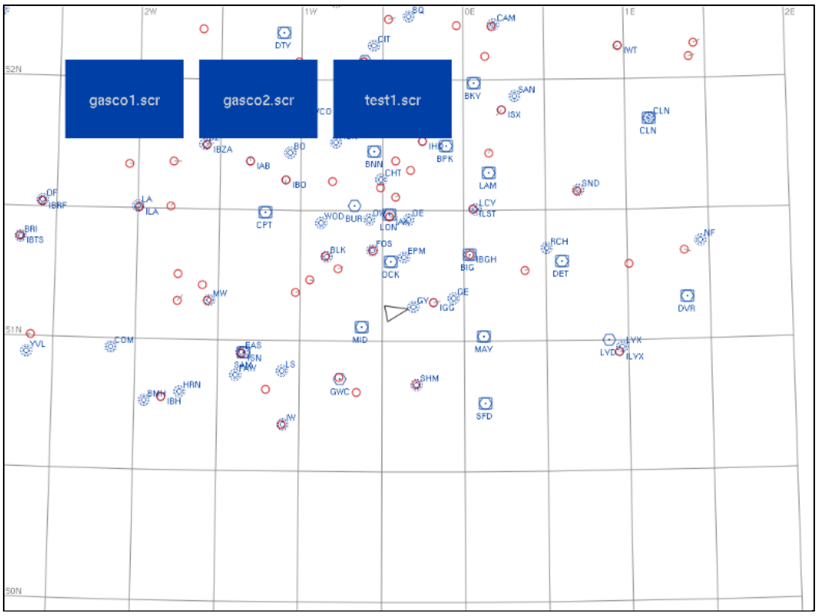


Figure 7.8.3

Function: Invoke a Script program.

Effect: A Script program is loaded and the events defined in the Script are executed.

Selection: Select the **Flight Plan** option from the main menu. Select the **Script** option from the sub-menu. A list of Script programs is displayed. Select a Script file from the list of Script program.

Confirmation: A new Script program is loaded and the events defined in the program are executed in parallel with the operation of the flight simulator. Any syntax errors are reported and if errors are encountered, the execution of the Script program is abandoned.

Example: In Figure 7.8.3, three Script programs are displayed.

Applications:

- To execute a program to initiate a predefine sequence of events.
- To record pilot performance
- To repeat a series of experiments with identical flight conditions

See also: Master-Restore

7.9 Target

A selection of commands are provided to load a target file in the visual scene and specify the target trajectory and speed. A number of detailed OpenFlight aircraft models are provided in the IG system. The target is currently modelled as a simple point mass, flying straight line or curved flight manoeuvres until modified by a new command issued at the instructor station. The target options are illustrated in Figure 7.9

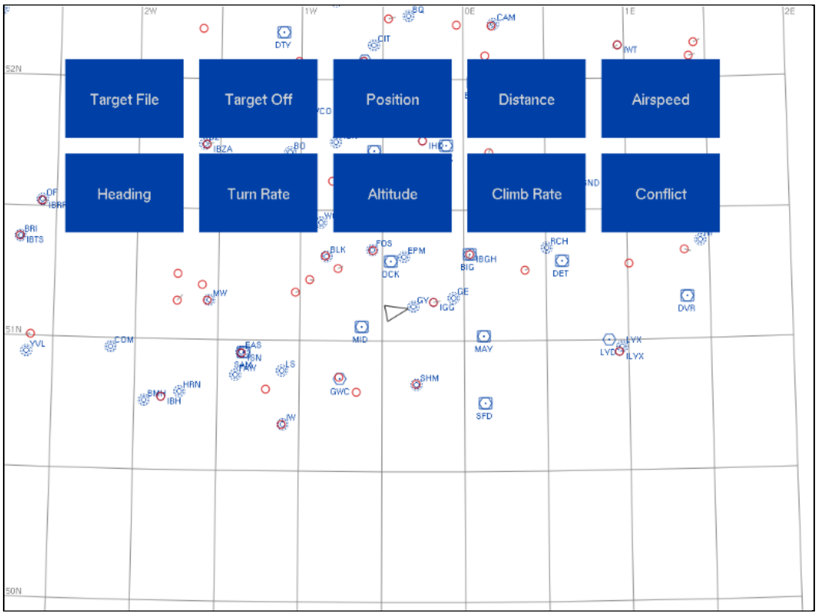


Figure 7.9

These commands are used to load a target in the scene, remove a target from the scene or to specify target parameters. Note that currently, the target dynamics are very simple. Although only one target can be active at any time, it is straightforward to implement a number of targets, subject to the performance of the IG system, and also to develop more complex dynamic models of targets.

7.9.1 Target-File

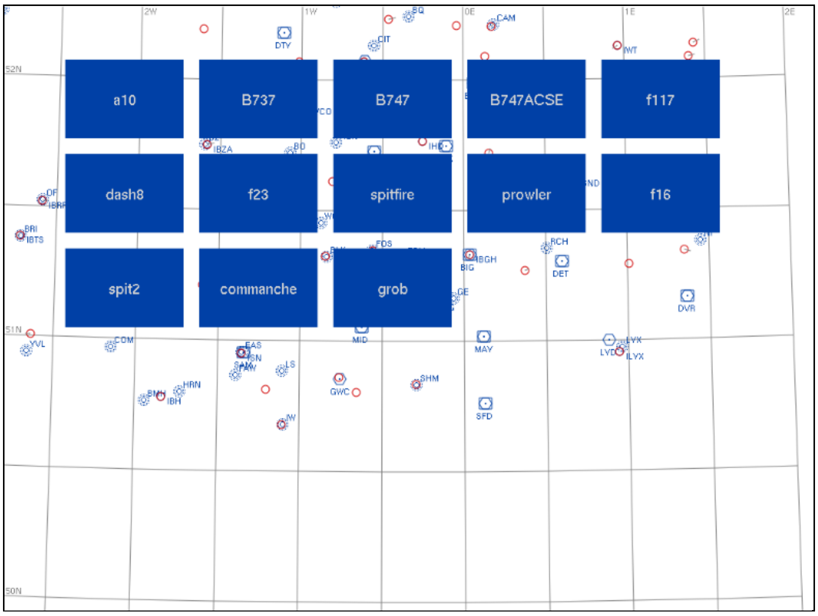


Figure 7.9.1

Function: Load a target file.

Effect: A specific target can be selected and loaded by the IG system, either to introduce a target or to replace an existing target.

Selection: Select the **Target** option from the main menu. Select the **Target File** option from the sub-menu. A list of targets is displayed. A target can be selected from the list of targets and replaces the current target (if any).

Confirmation: A new target is loaded and may appear in the visual system, depending on the aircraft position and attitude.

Note: In order to view the aircraft in the camera mode (see section 7.13), a target file must be loaded.

Example: In Figure 7.9.1, the Target File command has been selected and a list of target files is displayed.

Applications:
- To load a new target.
- To introduce conflicting traffic
- To provide a target for basic formation flying exercises

See also: Target-Off, Target-Position, Target-Distance, Target-Speed, Target-Heading, Target-Altitude, Target-Climb Rate, Target-Conflict.

7.9.2 Target-OFF

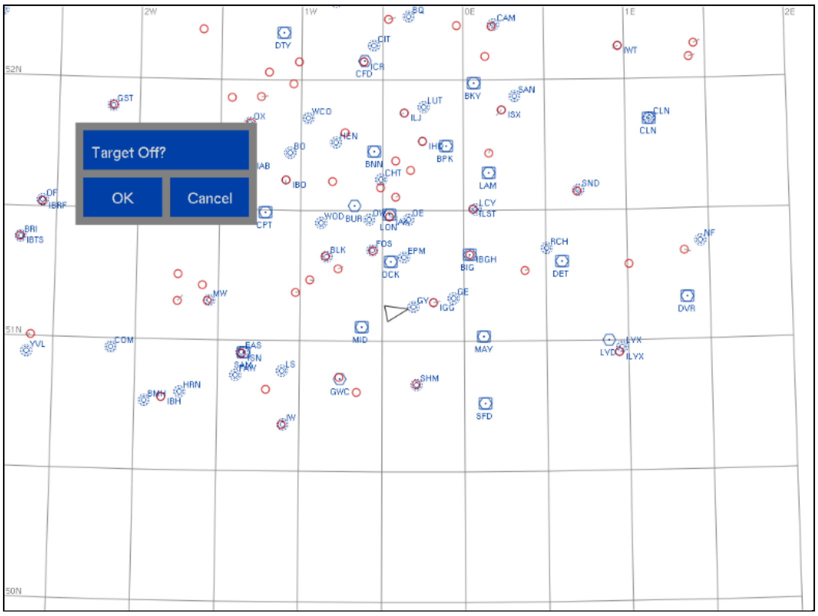


Figure 7.9.2

Function: The currently selected target can be turned off.

Effect: The visual system no longer displays the previously selected target. The target file is unloaded by the visual system.

Selection: Select the **Target** option from the main menu. Select the **Target OFF** option from the sub-menu. Click the OK confirmation button to stop displaying the target.

Confirmation: The target is no longer displayed.

Example: In Figure 7.9.2, the target options are displayed.

Note: If the target is removed, no model will be visible in the camera mode (see section 7.13).

Applications:

- To remove conflicting air traffic.

See also: Target-File, Target-Position, Target-Distance, Target-Speed, Target-Heading, Target-Altitude, Target-Climb Rate, Target-Conflict.

7.9.3 Target-Position

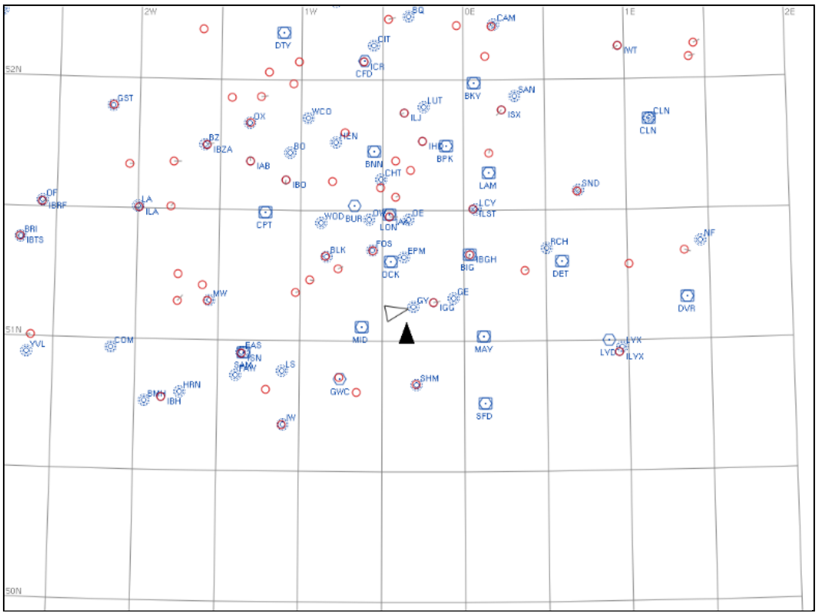


Figure 7.9.3

Function: The target is positioned on the displayed map.

Effect: The target is moved to the selected position.

Selection: Select the **Target** option from the main menu. Select the **Position** option from the sub-menu. As the mouse pointer is moved over the map display area, the map co-ordinates of the mouse pointer are displayed. At the appropriate position on the map display, press the left mouse key to re-position the target. The target is shown on the map as a solid black triangle.

Confirmation: The target may appear in the visual system, depending on the aircraft position and attitude.

Example: In Figure 7.9.3, the target has been positioned approximately at 51° 06' N, 00° 22' W.

Applications:

- To position the aircraft at a known position of latitude and longitude.
- To provide conflicting traffic at a specific location.

See also: Target-File, Target-OFF, Target-Distance, Target-Speed, Target-Heading, Target-Altitude, Target-Climb Rate.

7.9.4 Target-Distance

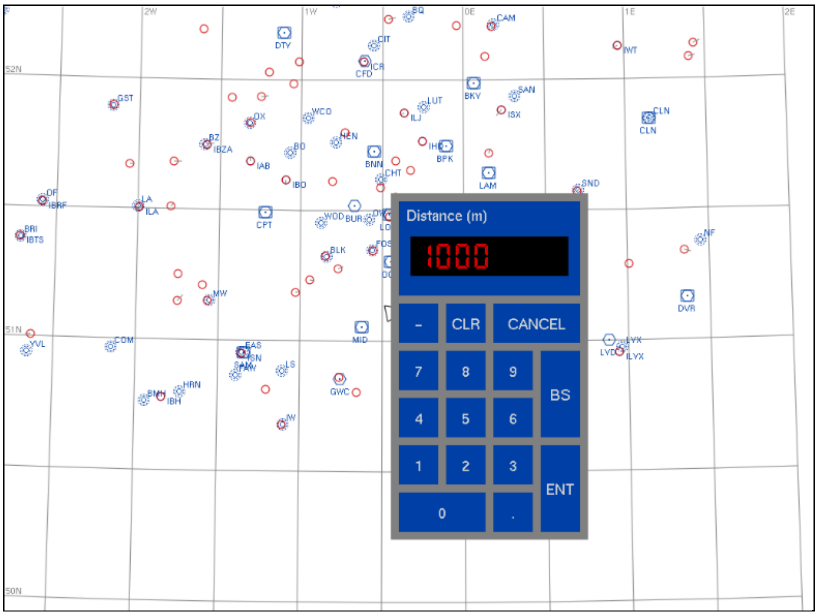


Figure 7.9.4

Function: The target is positioned at a specific distance directly in front of the aircraft.

Effect: The currently selected target should appear in the visual system.

Selection: Select the **Target** option from the main menu. Select the **Distance** option from the sub-menu. The current target distance is displayed in metres and can be set to a new value.

Confirmation: The target should appear at the specified distance.

Example: In Figure 7.9.4, the target distance has been set to 1000 m.

Applications:

- To introduce conflicting traffic.

See also: Target-File, Target-OFF, Target-Position, Target-Speed, Target-Heading, Target-Altitude, Target-Climb Rate.

7.9.5 Target-Speed

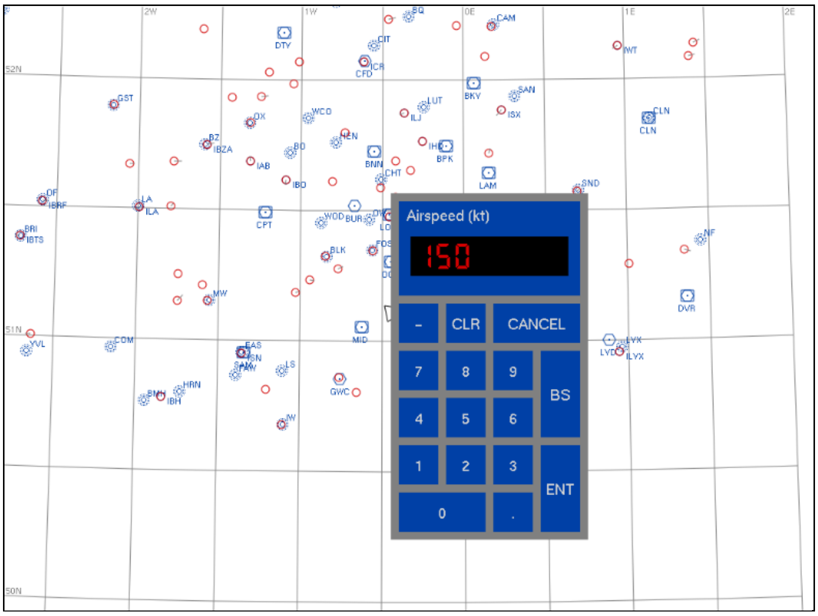


Figure 7.9.5

Function: The speed of the target is set to a specific value.

Effect: The airspeed of the currently selected target should change to the specified value.

Selection: Select the **Target** option from the main menu. Select the **Speed** option from the sub-menu. The current target airspeed is displayed in Kts and can be set to a new value.

Confirmation: The airspeed of the target should change to the specified value.

Example: In Figure 7.9.5, the target speed has been set to 150 Kts.

Applications:

- To select the closure rate of conflicting traffic.

See also: Target-File, Target-OFF, Target-Position, Target-Distance, Target-Heading, Target-Altitude, Target-Climb Rate.

7.9.6 Target-Heading

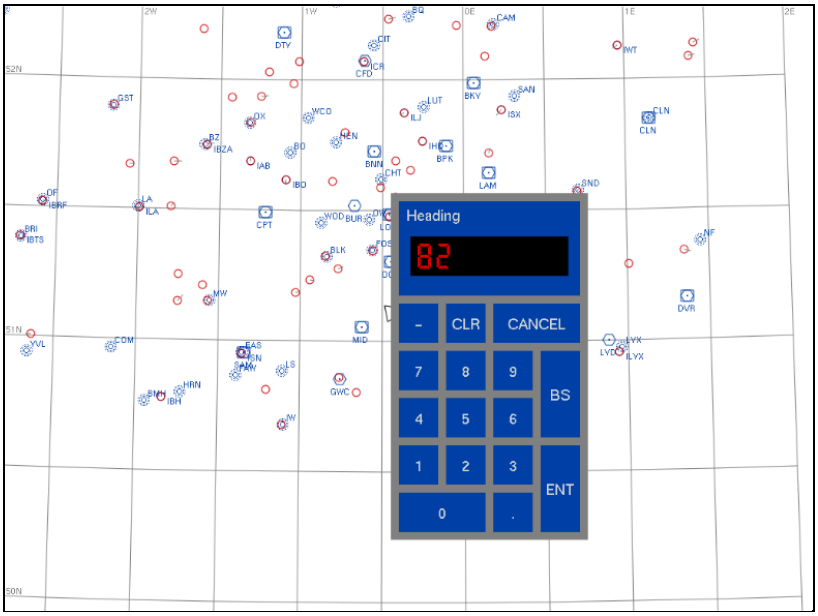


Figure 7.9.6

Function: The magnetic heading of the target is set to a specific value.

Effect: The heading of the currently selected target should change to the specified value.

Selection: Select the **Target** option from the main menu. Select the **Heading** option from the sub-menu. The magnetic heading of the current target is displayed in degrees and can be set to a new value.

Confirmation: The heading of the target should change to the specified value.

Example: In Figure 7.9.6, the heading of the target has been set to 082° M.

Applications:

- To introduce conflicting traffic.

See also: Target-File, Target-OFF, Target-Position, Target-Distance, Target-Speed, Target-Altitude, Target-Climb Rate.

7.9.7 *Target-Turn Rate*

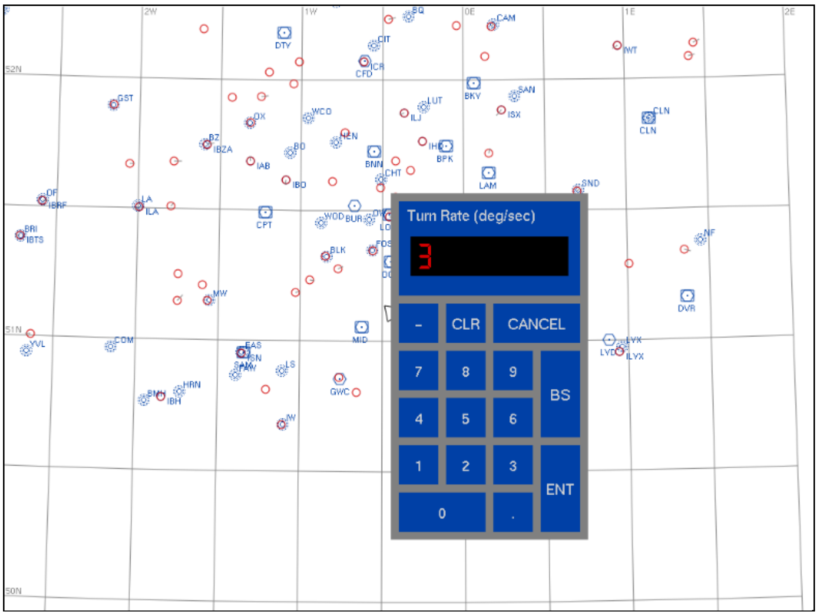


Figure 7.9.7

Function: The rate of turn of the target is set to a specific value.

Effect: The rate of turn of the currently selected target should change to the specified value.

Selection: Select the **Target** option from the main menu. Select the **Turn Rate** option from the sub-menu. The rate of turn of the current target is displayed in degrees per second and can be set to a new value. A positive value selects a rate of turn to the right and a negative value selects a rate of turn to the left.

Confirmation: The rate of turn of the target should change to the specified value.

Example: In Figure 7.9.7, the rate of turn of the target has been set to 3° per second (rate 1 turn).

Applications:

- To introduce conflicting traffic.

See also: Target-File, Target-OFF, Target-Position, Target-Distance, Target-Speed, Target-Altitude, Target-Climb Rate.

7.9.8 Target-Altitude

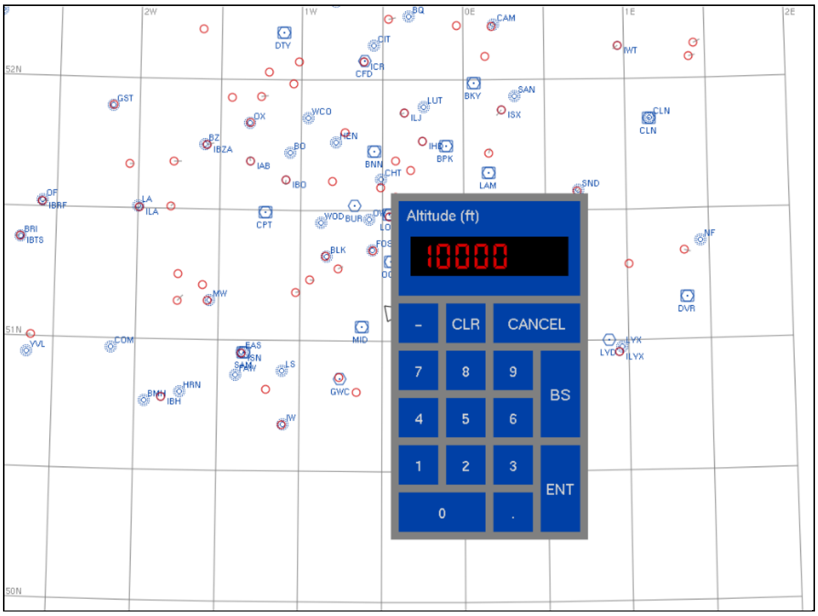


Figure 7.9.8

Function: The altitude of the target is set to a specific value.

Effect: The altitude of the currently selected target should change to the specified value.

Selection: Select the **Target** option from the main menu. Select the **Altitude** option from the sub-menu. The altitude of the current target is displayed in feet (ASL) and can be set to a new value.

Confirmation: The altitude of the target should change to the specified value.

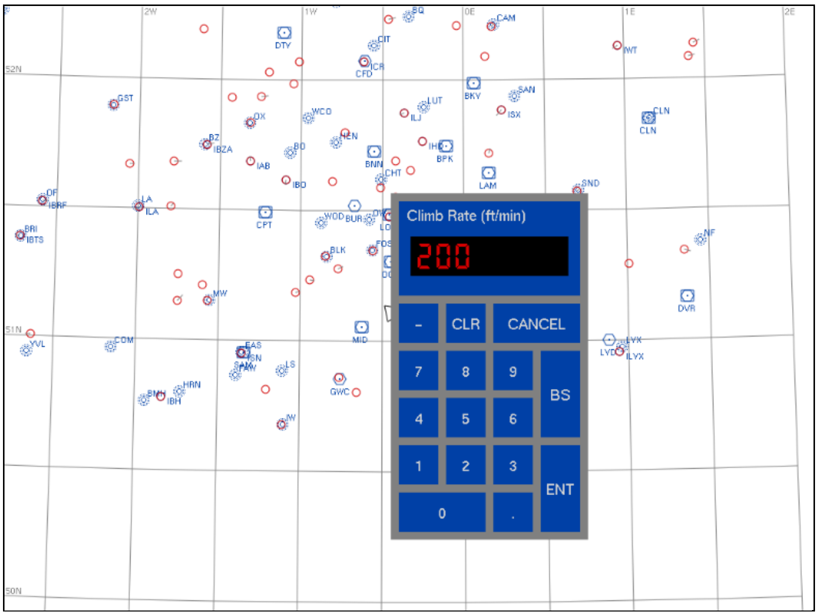
Example: In Figure 7.9.8, the altitude of the target has been set to 10000 feet.

Applications:

- To introduce conflicting traffic.

See also: Target-File, Target-OFF, Target-Position, Target-Distance, Target-Speed, Target-Turn Rate, Target-Climb Rate.

7.9.9 Target-Climb Rate



<u>Figure 7.9.9</u>

Function: The rate of climb of the target is set to a specific value.

Effect: The rate of climb of the currently selected target should change to the specified value.

Selection: Select the **Target** option from the main menu. Select the **Climb Rate** option from the sub-menu. The rate of climb of the current target is displayed in feet per minute and can be set to a new value. A positive value selects a rate of climb and a negative value selects a rate of descent.

Confirmation: The rate of climb of the target should change to the specified value.

Note: The target will continue to climb or descend unless the target is disabled or a climb rate of zero is selected.

Example: In Figure 7.9.9, the rate of climb of the target has been set to 200 feet per minute.

Applications:

- To introduce conflicting traffic.

See also: Target-File, Target-OFF, Target-Position, Target-Distance, Target-Speed, Target-Altitude, Target-Turn Rate.

7.9.10 Target-Conflict

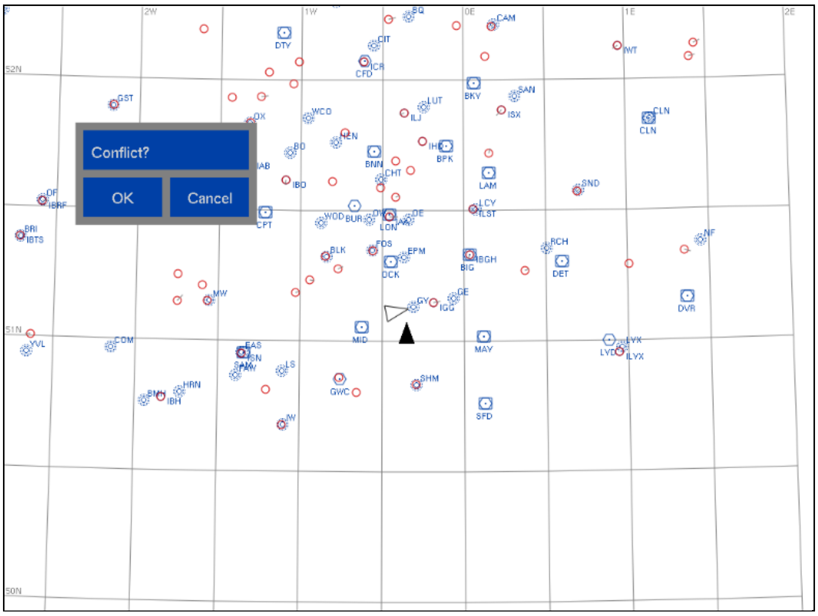


Figure 7.9.10

Function: Initiate a conflict between the aircraft and the target.

Effect: The target aircraft will attempt to manoeuvre to cause a traffic conflict.

Selection: Select the **Target** option from the main menu. Select the **Conflict** option from the sub-menu. To activate a conflict, click on the OK button.

Confirmation: The target aircraft should manoeuvre so that it poses a major conflict threat.

Example: In Figure 7.9.10, conflict option is displayed.

Applications:

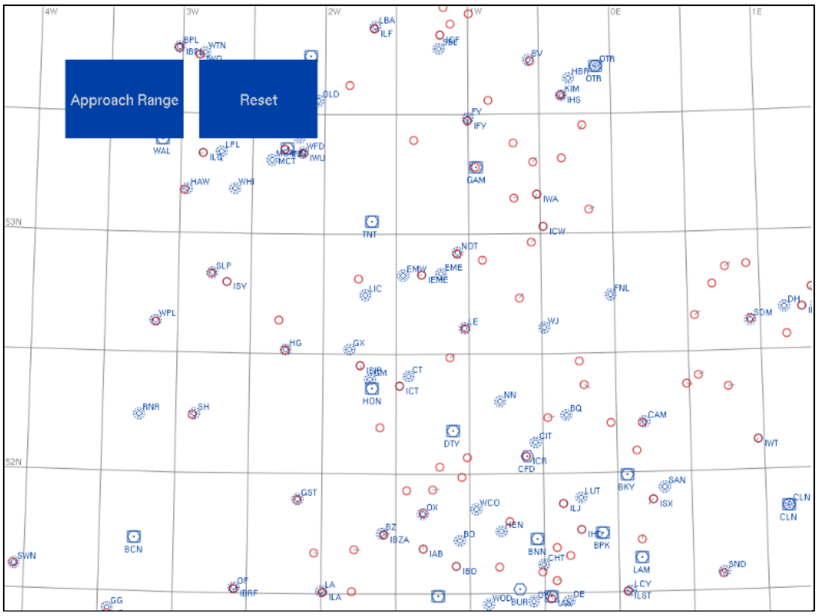
- To introduce conflicting traffic.

See also: Target-File, Target-OFF, Target-Position, Target-Distance, Target-Speed, Target-Altitude, Target-Turn Rate.

7.10 Approach

If the approach mode is displayed, the range displayed can be changed. In addition, previous aircraft tracks can be removed, resetting the display.

Figure 7.10



Normally, the display shows a range of 10 NM and a glideslope of 3°. If another approach is initiated, the previous tracks can be removed (or retained for comparison).

7.10.1 Approach-Range

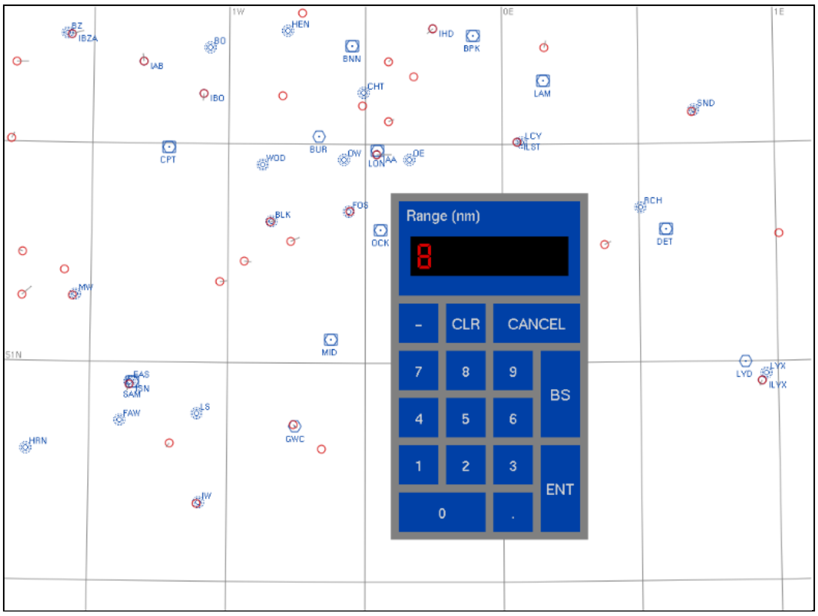


Figure 7.10.1

Function: Selects the range for the approach mode display in nautical miles.

Effect: The range of the approach mode is changed to the selected value and the approach mode display is redrawn.

Selection: Select the **Mode** option from the main menu. Select the **Approach Range** option from the sub-menu. The current range is displayed in nautical miles and can be set to a new value.

Confirmation: The approach mode is redrawn, with the selected range.

Example: In Figure 7.10.1, the approach mode display range has been set to 5 NM.

Applications:

- To monitor an ILS approach from a specific distance.

See also: Approach-Reset, Approach-G/S angle, Master-Map

7.10.2 Approach-Reset

Figure 7.10.2

Function: Reset the tracks displayed on the approach display.

Effect: The approach chart is redrawn and the tracks formed by the aircraft are discarded. This command has no effect on aircraft operation.

Selection: Select the **Approach** option from the main menu. Select the **Reset** option from the sub-menu. Click the OK confirmation button to reset the map display.

Confirmation: The approach display is redrawn and all tracks are removed.

Example: In Figure 7.10.2, the user is invited to reset the approach display.

Applications:

- To erase the tracks from a previous exercise.

See also: Approach-Range.

7.11 Data Recording

Commands are provided to manage the recording of flight data during simulation exercises. The facility is similar to flight data recording on flight test aircraft, where the primary flight data is recorded for subsequent analysis. The sub options are shown in Figure 7.11.

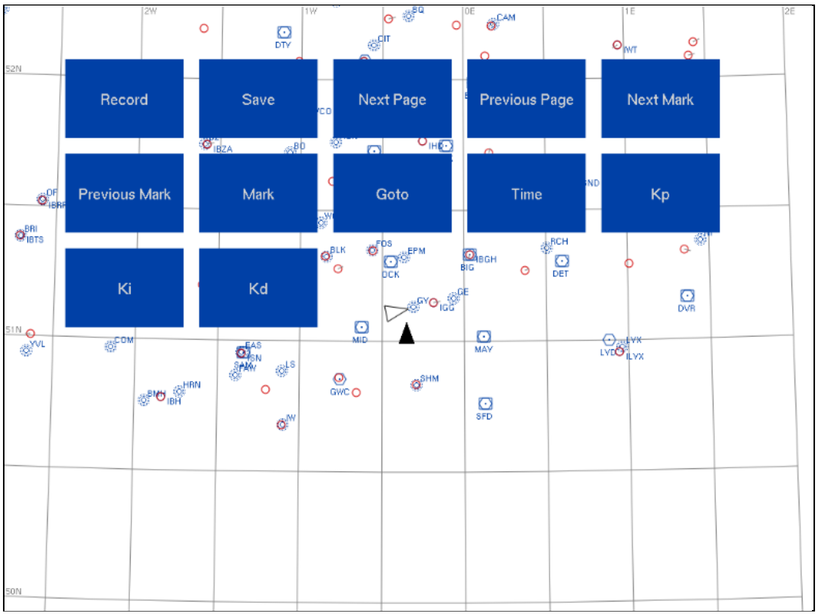


Figure 7.11

As the flight simulator is running, packets of data are broadcast by the simulator computers. In the data recording mode, these packets are stored and can be saved and retrieved for analysis. The data can be displayed as strip charts and analysed both on-line and off-line.

7.11.1 Data Recording-Record

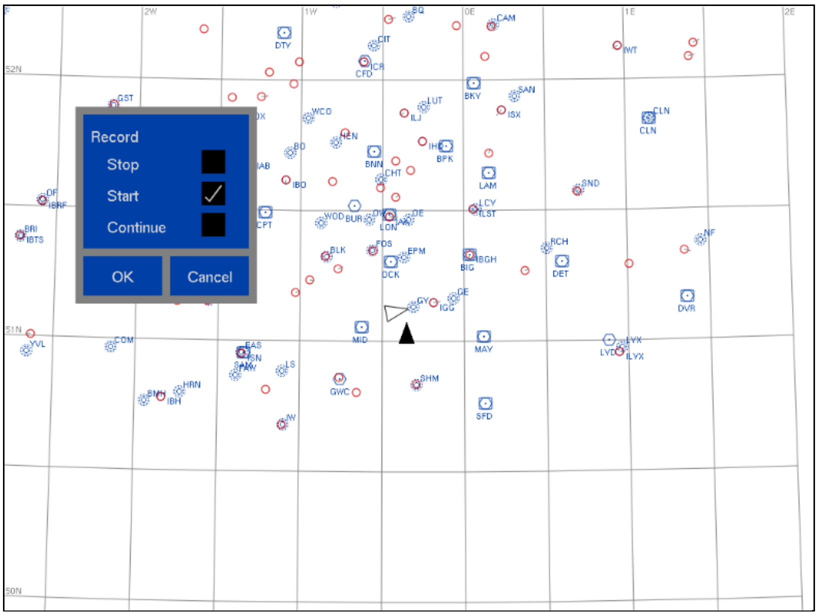


Figure 7.11.1

Function: Start, stop or continue the recording of flight data.

Effect: The organisation of the data packets is defined in the system-wide header file iosdefn.h and comprises the packets produced by the flight model computer, the navigation computer and the I/O computer. The flight data is written to a large buffer in the RAM area of the instructor station computer. It is advisable to turn off data recording when it is not needed. No data is recorded while the flight simulator is in the HELD mode.

Selection: Select the **Data Recording** option from the main menu. Select the **Record** option from the sub-menu. Select the appropriate radio button to stop, start or continue data recording.

Confirmation: The flight data will be recorded and visible if the flight data display mode is selected.

Note: Data recording needs to be turned on for data to be recorded.

Example: In Figure 7.11.1, the flight data recording hold has been started.

Applications: To record flight data for demonstrations, to validate a design, to acquire flight test data for a report.

See also: Data Recording-Save, Data Recording-Time

7.11.2 Data Recording-Save

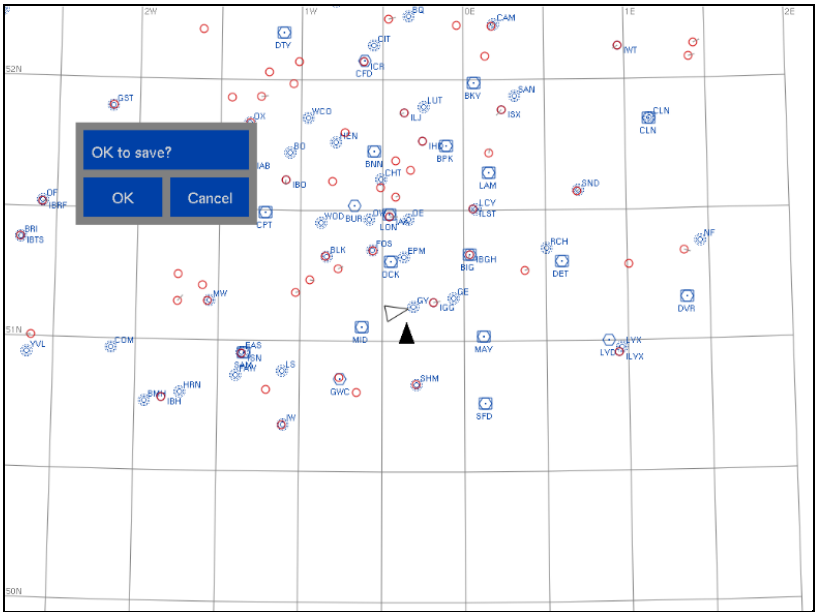


Figure 7.11.2

Function: Save the recording data to a file.

Effect: If the user confirms the save operation, the data is written to a file, typically as 1024-byte blocks of binary data which are a direct copy of the packet structure defined in the header file iosdefn.h. The file name is derived from the current time and date and the extension '.dat' is appended to the file name, for example a file created at 17:05 on the 30th of August will be given the name 30081705.dat.

Selection: Select the **Data Recording** option from the main menu. Select the **Save** option from the sub-menu. Click the OK confirmation button to save the flight data to a file.

Confirmation: There may be a small delay as the data is written to a file.

Example: In Figure 7.11.2, the user is invited to save the flight data.

Applications:

- To save data from an exercise, for later analysis.

See also: Data Recording-Record, Data Recording-Time.

7.11.3 Data Recording-Next Page

Function: Display the next page of flight data.

Effect: When the Data Recording option Next Page is selected, the next page of data is displayed, where the size of a page is determined by the time scale. The command is ignored if the displayed page is the last page of flight data.

Selection: Select the **Data Recording** option from the main menu. Select the **Next Page** option from the sub-menu.

Confirmation: The next page of flight data will be displayed.

Example:

Applications: To access different sections of flight data recording.

See also: Data Recording-Previous Page, Data Recording-Next Mark, Data Recording-Previous Mark, Data Recording-Time

7.11.4 Data Recording-Previous Page

Function: Display the previous page of flight data.

Effect: When the Data Recording option Previous Page is selected, the previous page of data is displayed, where the size of a page is determined by the time scale. The command is ignored if the displayed page is the first page of flight data.

Selection: Select the **Data Recording** option from the main menu. Select the **Previous Page** option from the sub-menu.

Confirmation: The previous page of flight data will be displayed.

Example:

Applications: To access different sections of flight data recording.

See also: Data Recording-Next Page, Data Recording-Next Mark, Data Recording-Previous Mark, Data Recording-Time

7.11.5 Data Recording-Next Mark

Function: Display the page of flight data indicated by the next data mark.

Effect: When the Data Recording option Next Mark is selected, the data page displayed starts from the next mark, where the size of a page is determined by the time scale. The command is ignored if the displayed page corresponds to the last marked page of flight data.

Selection: Select the **Data Recording** option from the main menu. Select the **Next Mark** option from the sub-menu.

Confirmation: The page of flight data starting from the next mark will be displayed.

Example:

Applications: To access different sections of flight data recording.

See also: Data Recording-Next Page, Data Recording-Previous Page, Data Recording-Previous Mark, Data Recording-Time

7.11.6 Data Recording-Previous Mark

Function: Display the page of flight data indicated by the previous data mark.

Effect: When the Data Recording option Previous Mark is selected, the data page displayed starts from the previous mark, where the size of a page is determined by the time scale. The command is ignored if the displayed page corresponds to the first marked page of flight data.

Selection: Select the **Data Recording** option from the main menu. Select the **Previous Mark** option from the sub-menu.

Confirmation: The page of flight data starting from the previous mark will be displayed.

Example:

Applications: To access different sections of flight data recording.

See also: Data Recording-Next Page, Data Recording-Previous Page, Data Recording-Next Mark, Data Recording-Time

7.11.7 Data Recording-Mark

Function: Enter a mark at the current point in the page of displayed flight data.

Effect: A visible mark is added to the time axis of the displayed data. Normally, a mark is added when the simulator is in the HOLD state, prior to commencing an exercise. However, a mark can also be added while data is being recording, for example to note some particular event.

Selection: Select the **Data Recording** option from the main menu. Select the **Mark** option from the sub-menu.

Confirmation: The current page of flight data will indicate a mark at the frame time when the command is activated.

Example:

Applications: To note events during flight data recording, to indicate the start point of a recording.

See also: Data Recording-Next Page, Data Recording-Previous Page, Data Recording-Next Mark, Data Recording-Previous Mark

7.11.8 Data Recording-Goto

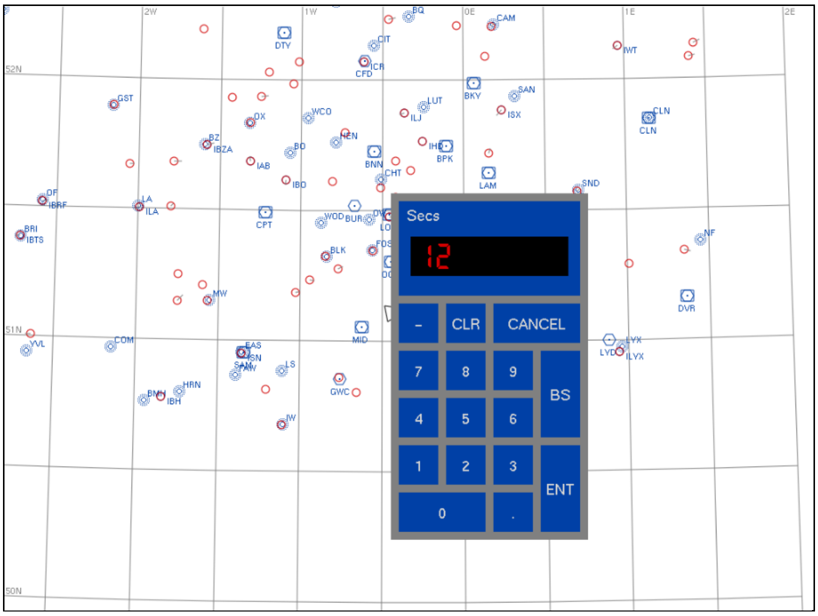


Figure 7.11.8

Function: The current page of flight data is set to start from a specific time.

Effect: The current displayed page of flight data is reset to display data starting from the entered time.

Selection: Select the **Data Recording** option from the main menu. Select the **Goto** option from the sub-menu. The current time of the time axis origin is displayed and can be set to a new value.

Confirmation: The displayed page of flight is reset to start from the new time entered.

Example: In Figure 7.11.8, the flight data origin is set to 12 seconds.

Applications: To access different sections of flight data recording, to display flight data from a specific point in the data recording.

See also: Data Recording-Time.

7.11.9 Data Recording-Time

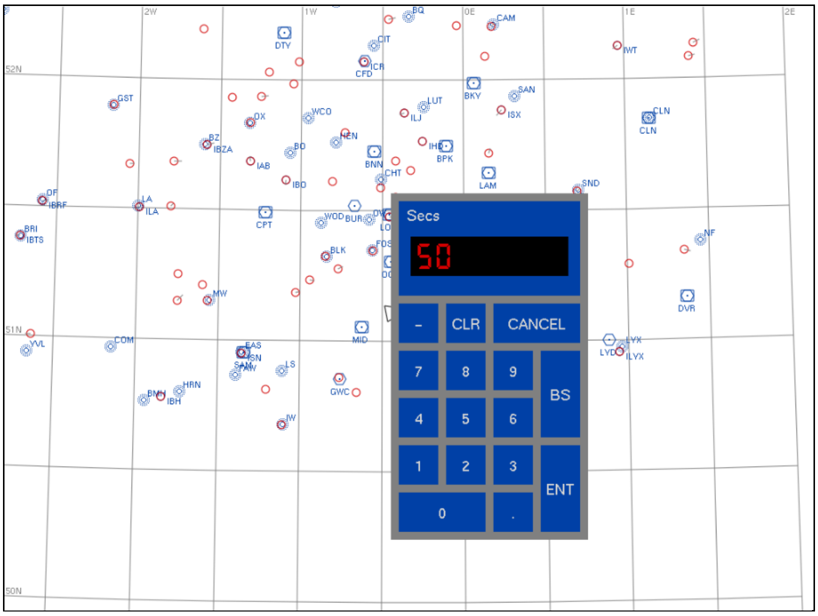


Figure 7.11.9

Function: The time axis of the flight data display is set to a specific value. By default, a displayed page is 10 seconds.

Effect: The current displayed page of flight data is redisplayed with a new time axis.

Selection: Select the **Data Recording** option from the main menu. Select the **TIME** option from the sub-menu. The current range of the time axis origin is displayed and can be set to a new value.

Confirmation: The displayed page of flight is reset to display flight data over the entered time range.

Example: In Figure 7.11.9, the flight data time axis is reset to 50 seconds.

Applications: To expand or compress displayed data during flight data recording.

See also: Data Recording-Goto.

7.11.10 Data Recording-Kp

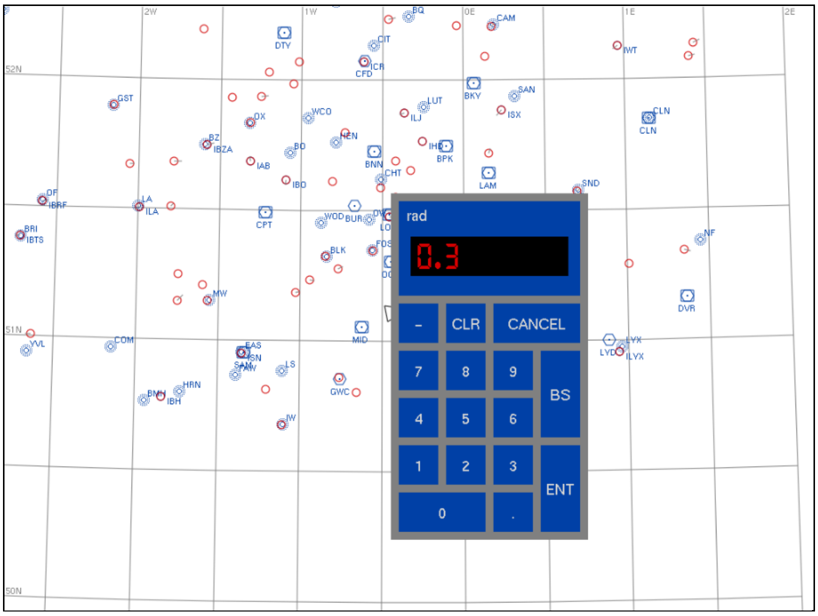


Figure 7.11.10

Function: Three variables K_P, K_I and K_D are provided (which correspond to the coefficients of a PID controller). The purpose is to include these variables in a user program (which can be accessed via the IOS packet) and to be able to modify these values at the IOS without needing to recompile the software.

Effect: The current value of the variable is displayed and is set to a new value.

Selection: Select the **Data Recording** option from the main menu. Select the **Kp, Ki or Kd** option from the sub-menu.

Confirmation: The new value of the specific variable will change. The effect may be seen in the user software.

Example: In Figure 7.11.10, the value of Kp is set to 0.3.

Applications: To provide three coefficients for a user program which can be modified from the IOS.

See also:

7.12 Flight Data

The variables displayed on the flight data display as strip charts can be displayed. Typically, up to five variables can be displayed at any time, where the data recording and display is selected in the Data Recording options, described in section 7.11.

The flight data variables are shown in Figure 7.12. A variable can be selected to be displayed with the appropriate units.

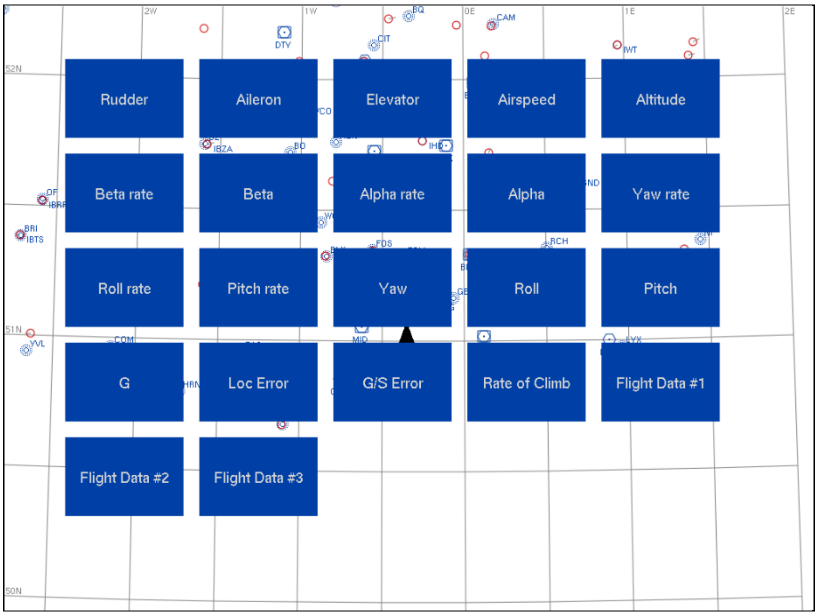


Figure 7.12

The following variables are displayed:

Name	IOS Packet Variable	Details
Rudder	Aeropkt.Rudder	Rudder position
Aileron	AeroPkt.Aileron	Aileron position
Elevator	Aeropkt.Elevator	Elevator position
Airspeed	Aeropkt.Vc	True airspeed
Altitude	-Aeropkt.Pz	Altitude (+ve up)
Beta Rate	Aeropkt.BetaDot	Rate of change of angle of sideslip
Beta	Aeropkt.Beta	Angle of sideslip
Alpha Rate	AeroPkt.AlphaDot	Rate of change of angle of attack
Alpha	AeroPkt.Alpha	Angle of attack
Yaw Rate	Aeropkt.R	Yaw rate
Roll Rate	Aeropkt.P	Roll rate
Pitch Rate	Aeropkt.Q	Pitch rate
Yaw	Aeropkt.Yaw	Yaw
Roll	Aeropkt.Roll	Roll
Pitch	Aeropkt.Pitch	Pitch
G	AeroPkt.Lift / (AeroPkt.Mass * 9.81)	G load
Loc Error	NavPkt.ILS1.LocaliserError	Localiser error
G/S Error	Pkt.NavPkt.ILS1.GlideSlopeError	Glide slope error
Rate of Climb	-Pkt.AeroPkt.Vd	Rate of climb (+ve up)
Flight Data #1	AeroPkt.FlightData[0]	User defined variable
Flight Data #2	AeroPkt.FlightData[1]	User defined variable
Flight Data #3	AeroPkt.FlightData[2]	User defined variable

Although a fixed range of variables is presented, it is straightforward to modify the IOS software to display other variables.

Note the three user defined variables that can be set in a user program and displayed.

The flight data variable *pitch rate* is shown in Figure 7.12.1 to illustrate the selection and setting of flight data variables for recording.

7.12.1 Flight Data-Pitch Rate

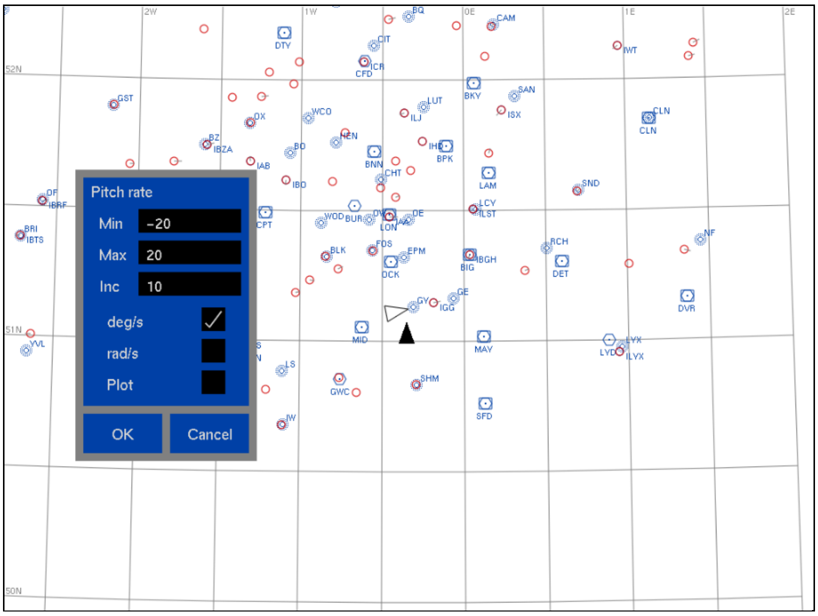


Figure 7.12.1

Function: A flight data variable can be selected for plotting on a strip chart. The range, spacing and units can also be selected for each variable.

Effect: In the flight data display mode, the selected variable can be displayed or removed and its range and units can be modified.

Selection: Select the **Flight Data** option from the main menu. Select the **Pitch Rate** option from the sub-menu.

Confirmation: The current display status, range and units associated with the flight data variable are displayed and can be modified.

Example: In Figure 7.12.1, the flight data variable Pitch Rate is not plotted, its current range is ±20°, plotted at 10° intervals and the units are degrees per second.

Applications: To customise the plotting of flight data.

See also:

In Figure 7.12.2 the maximum value of Pitch Rate is set to 20°.

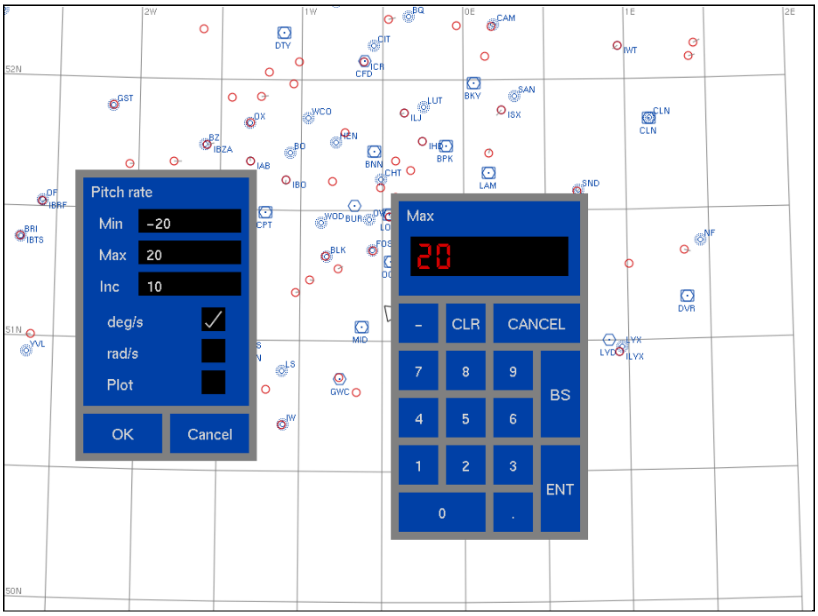


Figure 7.12.2

In Figure 7.12.3 the variable pitch Rate is selected for plotting (in degrees per second).

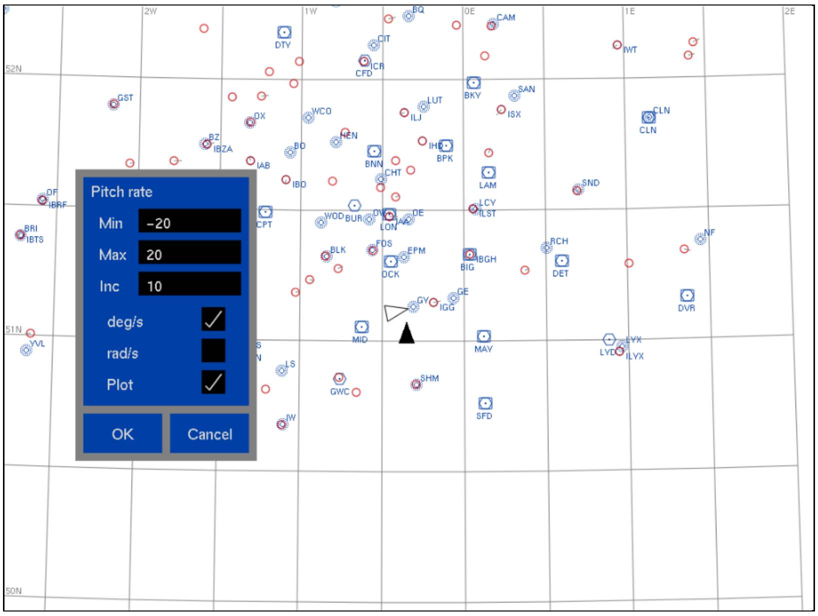


Figure 7.12.3

7.13 Camera

For most applications, the camera position, rendered by the IGs and projected is the pilot eye-point. However the camera position can be set to other locations and recorded data can be replayed. The five options relating to control of the camera and recording are shown in Figure 7.13

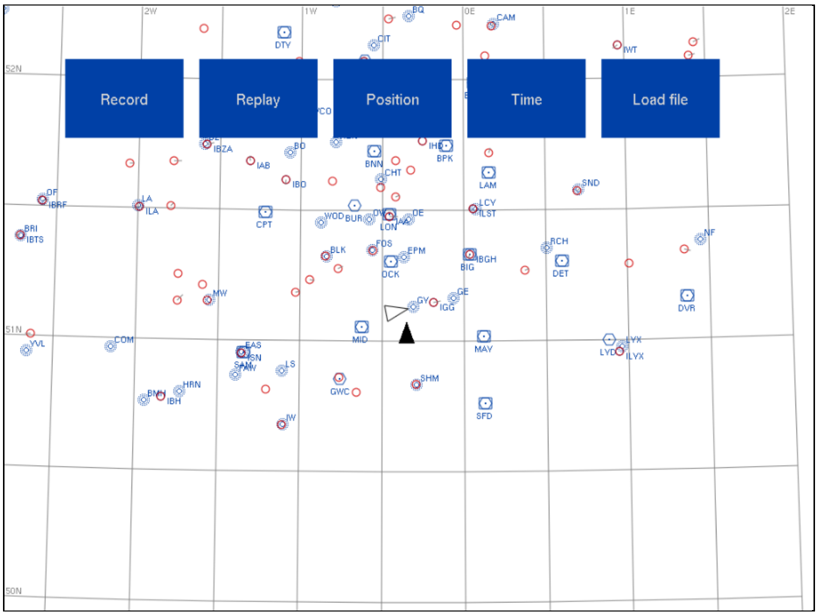


Figure 7.13

7.13.1 Camera-Record

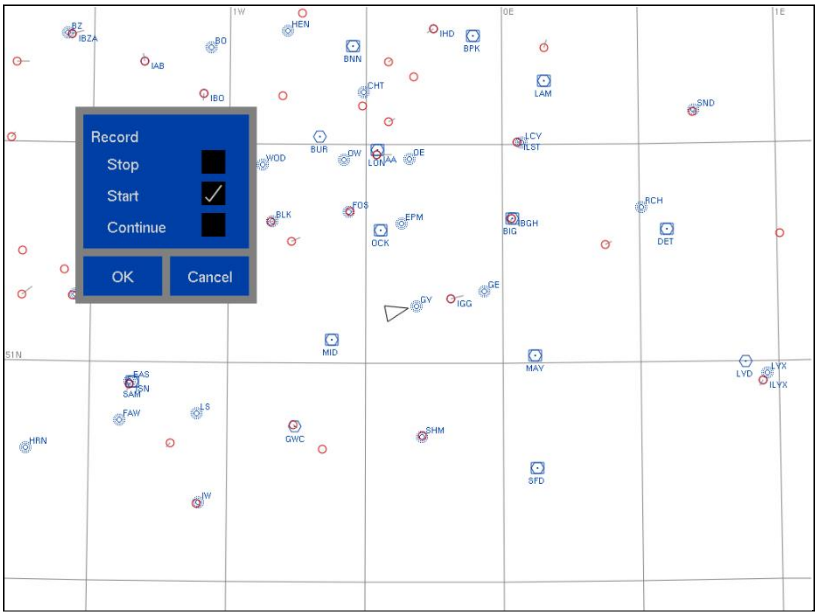


Figure 7.13.1

Function: Start, stop or continue the recording of flight data for subsequent replay.

Effect: The simulator data packets can be recorded and the recording can be started (overwriting previously recorded data), stopped and resumed.

Selection: Select the **Camera** option from the main menu. Select the **Record** option from the sub-menu. Select the appropriate radio button to stop, start or continue recording.

Confirmation: The flight data will be recorded in the background.

Example: In Figure 7.13.1, the flight data recording has been started.

Applications: To record flight data for demonstrations, to replay flight manoeuvres.

See also: Camera-Replay, Camera-Position, Camera-Load File, Camera-Time.

7.13.2 Camera-Replay

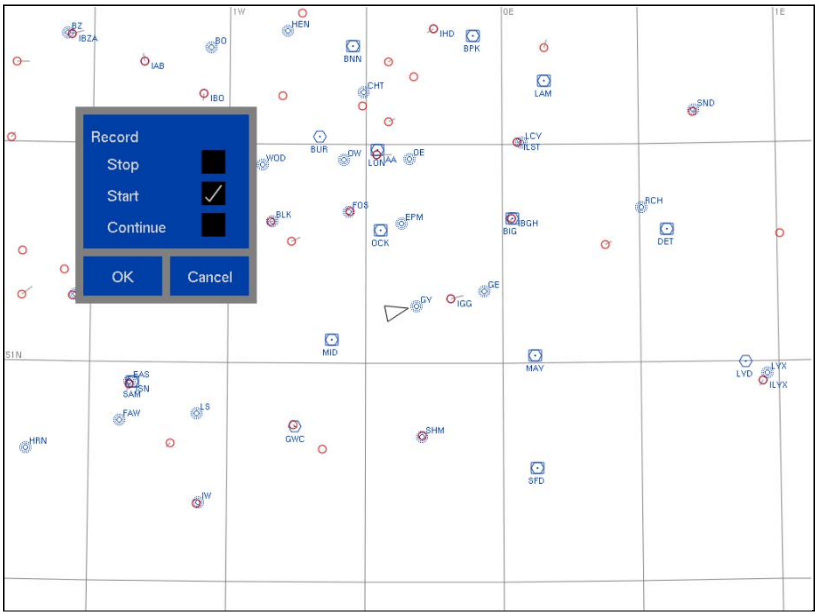


Figure 7.13.2

Function: Start, stop or continue the replay of flight data.

Effect: Recorded data packets are replayed. The replay can be started, stopped or resumed.

Selection: Select the **Camera** option from the main menu. Select the **Replay** option from the sub-menu. Select the appropriate radio button to stop, start or continue replaying.

Confirmation: If flight data has been recorded, the current recording will be replayed.

Note: Recording needs to be turned on for data to be recorded.

Example: In Figure 7.13.2, the flight data replay has been started.

Applications: To replay flight data for demonstrations, to replay flight manoeuvres.

See also: Camera-Record, Camera-Position, Camera-Load File, Camera-Time.

7.13.3 Camera-Position

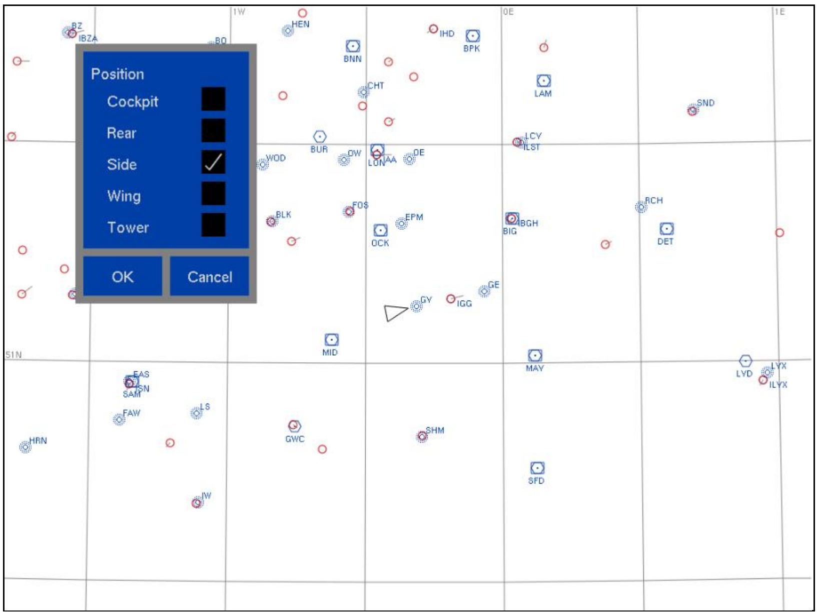


Figure 7.13.3

Function: Select the camera position.

Effect: The camera position can be set to one of five positions during flight or during replay.

Selection: Select the **Camera** option from the main menu. Select the **Position** option from the sub-menu. Select the appropriate radio button to select the camera position.

Confirmation: The camera position used by the IG system will change to the new setting.

Note: A target file needs to be loaded to visualise an aircraft in the IG system.

Example: In Figure 7.13.3, the camera location is set to the side view.

Applications: To replay flight data for demonstrations, to replay flight manoeuvres.

See also: Camera-Record, Camera-Replay, Camera-Load File.

7.13.4 Camera-Time

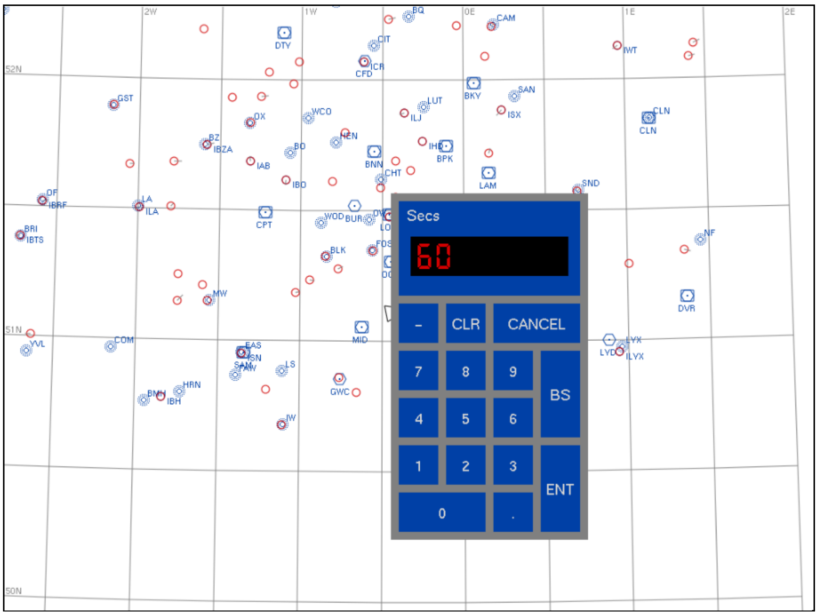


Figure 7.13.4

Function: The time for recording flight data is set to a specific value.

Effect: The amount of data recorded is set to the entered value.

Selection: Select the **Camera** option from the main menu. Select the **Time** option from the sub-menu. The current recording time is displayed and can be set to a new value. By default, 60 seconds of flight data is recorded.

Confirmation: The recording time will change to the new value.

Example: In Figure 7.13.4, the recording time is reset to 60 seconds.

Applications: To change the amount of flight data recorded for a demonstration.

See also: Camera-Record, Camera-Replay, Camera-Load File.

7.13.5 Camera-Load File

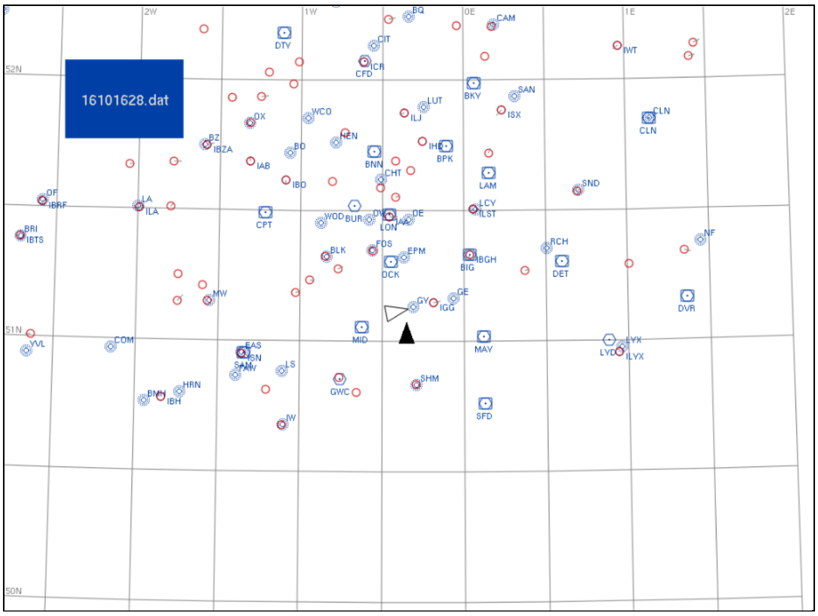


Figure 7.13.5

Function: Load a flight data file.

Effect: The user can select a specific flight data file for replay.

Selection: Select the **Camera** option from the main menu. Select the **Load File** option from the sub-menu. A list of recorded flight data files is displayed. Move the mouse over the list of files and press the left mouse key to select the file. There is no confirmation of this command.

Confirmation: A new flight data file will be loaded and can be viewed using the Camera Replay option.

Example: In Figure 7.13.5, the Load File option has been selected and the current selection of recorded flight data files (16101627.dat) is displayed.

Applications:

- To load a recording for a demonstration.

See also: Camera-Record, Camera-Replay, Camera-Position.

8 MATLAB INTERFACE

8.1 Introduction

The LFS includes an integrated interface to Mathworks MATLAB. The interface enables students to design and develop prototype aircraft flight control code, written as M-FILEs and executed in MATLAB, where the output of the flight control code is transmitted to the flight simulator to overriding manual or automatic flight control inputs (elevator, aileron, rudder and throttle). This *external interface* between the flight simulator and MATLAB provides a convenient method of rapid prototyping flight control code in the MATLAB environment without the need to access, develop or recompile the flight simulator software. Additionally, the interface provides an accessible way for students to learn and test automated flight control concepts in an environment with which they may already be familiar.

This section comprises two parts, which are intended for two different groups of users:

1. Users developing and testing flight control code using MATLAB (e.g. students, educators, researchers), and
2. Developers needing to understand, modify or extend the interface code that connects the flight simulator to MATLAB (e.g. simulator support staff).

8.2 User Guide

Use this guide if you intend to use MATLAB scripts to develop flight control laws and test the output using the flight simulator.

8.2.1 Overview of the Interface and Components

Connecting MATLAB to the flight simulator is achieved by use of a custom *MEX* file that passively listens for flight simulator UDP packets and returns a UDP packet to the flight simulator at the appropriate time. A precompiled MEX file is provided which requires the 64-bit version of MATLAB running on Microsoft Windows (7, 8, 10) and is connected to the dedicated flight simulator LAN - see Section 2.3. No additional MATLAB toolboxes are required.

Required files:
fslink.mexw64 Precompiled MEX file.
simctrl-template.m Template script.
simctrl-constants.m Script containing constants used within **simctrl-template.m**

These files should be copied from the flight simulator external interface resources directory into a working directory. The working directory should be added to the MATLAB path. There is no need to alter or recompile the fslink.mexw64 file. The copy of **simctrl-template.m** script file in the working directory should be renamed, for example **simctrl-speedcontrol.m**.

8.2.2 MATLAB Interface M-File Script

A template M-FILE script is presented as an example to assist with familiarisation of the Flight Simulator – MATLAB interface. This template should be copied and renamed, rather than being overwritten. This walk-through of the script describes the basic structure of the M-FILE and the necessary MEX file calls required to establish a link to the flight simulator.

At the start of the file, there is a call to another M-FILE script. The purpose of this script is to define constants that are used throughout this template script. The variable **done** is used as a condition to control the main loop. It is set to 1 when MATLAB mode is disabled in the IOS, causing the loop to exit cleanly. The variable **matlabMode** is read from the flight simulator data every frame and indicates that the IOS has activated the flight simulator MATLAB interface. The variable **matlabRunning** is set to 1 if **matlabMode** is ever true. If **matlabMode** subsequently becomes 0 because the IOS has disabled the MATLAB interface, **matlabRunning** remains 1. The Boolean state of **matlabMode = 0** and **matlabRunning = 1** provides the necessary condition to determine that MATLAB mode has been disabled at the IOS.

```
% Define the constants used in the MEX file interface
simctrl-constants;

% Main loop control
done = 0;

% Gets set to 1 when the IOS activates MATLAB mode
matlabMode = 0;

% Set to 1 when matlabMode 1 detected. Used to detect IOS has deactivated MATLAB mode
matlabRunning = 0;
```

The following line is the first call to the MEX file interface. Constants are used to inform the MEX file which operation to execute and to pass a parameter specific to that operation.

```
% Open the UDP port for the flight simulator connection
fslink(fslink_open,fslink_defaultport);
```

Here, the script enters the main loop of waiting for the data from the current frame of the flight simulator, executing any control calculation, and returning data back to the flight simulator in the form of control inputs.

```
%
% Main Loop - It is advised to not execute a Ctrl-C at the MATLAB prompt.
% The script will exit in an orderly way via the IOS. Only in the event that
% the flight simulator has unexpectedly stopped, should Ctrl-C be used.
%
while done == 0
```

Several calls to the MEX file interface are made before any script code is used in the computation of a control law.

```
% Wait for the current flight simulation iteration packets to arrive.
fslink(fslink_recv);

% Extract the received data from the MEX file into a Matlab array - DIN (Data IN)
DIN = fslink(fslink_dataget);
matlabMode = DIN(fslink_Active); % Flight simulator MATLAB mode state.
```

If the data received from the IOS indicates that MATLAB mode is enabled, the script executes the block containing the flight control law code, otherwise it waits for MATLAB mode to be enabled.

```
% Only compute and send data to the flight simulator if mode is active
if matlabMode == 1

    % Flag that MATLAB mode on IOS has been set. Once set, it remains 1
  % If the IOS switches MATLAB mode off, matlabMode = 0 and matlabRunning = 1
  % signals that the script exit.
  matlabRunning = 1;

  %
  % CUSTOM FLIGHT CONTROL CODE STARTS HERE
  %
```

This code marks the beginning of the region in the script where user code can be written.

The output from a prototype control law needs to be formed as a small packet to be sent to the flight simulator. The MEX file interface requires a four element array containing values for aileron, elevator, rudder and throttle position, in that order. If the control law requires that any of the flight control inputs are unaltered, then those values can be returned using the data in the DIN array. See the examples in sections 8.2.4 and 8.2.5.

```
% Finally, prepare the control data array for the MEX file - DOUT (Data OUT)
% This should be the last line in the custom code

DOUT = [0.0, 0.0, 0.0, 1.0];

%
% CUSTOM FLIGHT CONTROL CODE ENDS HERE
%
```

Flight controls are now passed back to the MEX file to be sent to the flight simulator. The values returned to the flight simulator will override the usual flight control inputs used by the flight model in the next frame of the flight simulator.

```
% Pass the control data to the MEX file
fslink(fslink_dataset,DOUT);

% Send the control data to the flight simulator
fslink(fslink_send);
```

As soon as the script detects that MATLAB mode has been disabled from the IOS, the main loop of the script exits. The remaining lines in the script close the network connection and clear the memory used by the MEX file.

```
% Close UDP connection
fslink(fslink_close);

% Clean up the memory used by the MEX file
clear fslink;
```

8.2.3 Executing a Script

In order to execute a flight control script (assuming no M-FILE control script bugs are present), the following sequence should be followed to ensure that the flight simulator is able to receive data from MATLAB:

1. Start the flight simulator as described in **Section 4.4 Start Up Procedure**
2. Restore the flight simulator to a position and state suitable for testing the control script. The restore operation is found in **Section 7.2.1 Reposition-Restore**. Place the simulator in the HOLD state.
3. Start the MATLAB script, either via typing the script name at the MATLAB command prompt, or by clicking the Run button. If the script has started and initialised the MEX file component correctly, the following output will be seen in MATLAB. These lines show that the interface to the flight simulator has been established and that the script is now waiting for the flight simulator to been switched into MATLAB MODE. If your output does not show this, then the network connection to the flight simulator has probably failed to have been established. In this case, follow the procedure outlined in **Section 8.2.6, Restarting the Flight Simulator - MATLAB Interface Component.**

```
Calling Function MEX_UDP_START
Port : 54321
UDP_Start : Matlab UDP connection ready for datagrams 54321
rv = 1
```

4. At the IOS, activate the flight simulator MATLAB mode as described in **Section 7.1.7 Master-Matlab.** Selecting the ON option (followed by OK) notifies the simulator that the MATLAB PC should be included in the protocol, receiving data packets from MATLAB to provide the flight control inputs.
5. The flight simulator will still be in the HELD state and the MATLAB script will be executing, but receiving unchanging data from the simulator. In turn, the MATLAB script will now be executing the custom control code and sending data back to the flight simulator. Remove the flight simulator from the HELD state. The flight control script will now be providing the flight control inputs for the simulator.
6. If the simulation is complete (for example, modifications to the script are required) **DO NOT type Ctrl-C** at the MATLAB command prompt, as this will violate the protocol with the flight simulator and cause the other simulator nodes to 'hang'. Rather, switch the flight simulator to the HELD state, and then deselect the flight simulator MATLAB mode as described in **Section 7.1.7 Master-Matlab.** This will take the simulator out of MATLAB mode cleanly and also inform the MATLAB script to stop and disconnect cleanly.
7. The simulator should still be operating, but in the HELD state. Modifications can now be made to the flight control script, or a new script developed.
8. To test the modified script, restart this process from step (2) above.

8.2.4 Example 1. Echoing the Flight Controls

A portion of an M-FILE script is presented (**simctrl-echo.m**) which receives data from the flight simulator and returns the same flight control inputs back to the flight simulator. This simple script demonstrates an important concept - reading and writing data to and from the flight simulator.

As described in the template script, a MATLAB array, DIN, is created and populated with the data received from the flight simulator. Variables are then created which store a local copy of the current flight controls by accessing the DIN array, with the specific indices defined in **simctrlconstants.m**.

```
% Extract the received data from the MEX file into a Matlab array - DIN (Data IN)
DIN = fslink(fslink_dataget);

% Access simulator data
de = DIN(fslink_Elevator);
da = DIN(fslink_Aileron);
dr = DIN(fslink_Rudder);
dt = IN(fslink_Throttle);
```

Because this example is only intended to demonstrate how to access simulator data, the retrieved data is returned back to the flight simulator by filling the array DOUT with the values that were previously read. If this script is executed as described in **Section 8.2.3 Executing a Script**, the flight simulator should operate under manual flight control for elevator, aileron, rudder and throttle inputs.

```
DOUT = [da, de, dr, dt];
```

8.2.5 Example 2. Altitude Hold Control Law

A portion of an M-FILE script is presented (**simctrl-altitudehold.m**) which implements a basic altitude hold control law. When executing this code, it is advisable to position the flight simulator model at a different altitude (e.g. ±500 feet) from the reference altitude defined in the altitude control law M-FILE script.

Near the start of the script, global variables defining a reference altitude and elevator limits are declared and initialised;

```
Href   = -2500.0 * 0.3048;   % 2500 ft -> m
de     = 0.0;
demin  = -0.4;
demax  = 0.4;
```

As in the previous example 8.2.4, data needed for the altitude hold flight control law is extracted from the array DIN.


```
U     = DIN(fslink_U);  % U
Udot  = DIN(fslink_Udot);  % Udot
H     = DIN(fslink_Altitude);  % Altitude
pitch = DIN(fslink_Pitch);  % Pitch
alpha = DIN(fslink_Alpha);  % Alpha
q     = DIN(fslink_Q);  % Q
Vd    = DIN(fslink_Vd);  % Vd
if ( U < 0.1 )
    U = 0.1;
End
```

The script lines that follow implement an altitude hold control law. At the end of this block of code, the elevator command (in the range -1.0 to 1.0) is includes in the array DOUT, to be transmitted to the flight simulator. Note that this control law only acts as an elevator command and sets the aileron and rudder commands to zero. These zeros values represent a neutral flight control input, in other words the flight control column is centered. Also note that the throttle input is set to echo the current throttle setting.

```
VSref = -0.08333*(Href - H);
               if VSref > 5.08
    VSref = 5.08;
elseif VSref < -5.08
    VSref = -5.08;
end

% Vertical speed controller
FPAngC = VSref / U;

% Flight path angle controller
pitchC = FPAngC + alpha;

% Pitch angle controller (converts pitch angle error into pitch rate demand)
qC = 0.2*(pitchC - pitch); % Pitch rate commanded

% Pitch rate controller (converts pitch rate error into elevator command)
dedot = -2.0*(qC - q); % Elevator change from pitch rate error
de = de + 0.02*dedot; % Forward Euler integration

% Apply the elevator input saturation limits
% Limit: demin <= de <= demax
               if de > demax
    de = demax;
elseif de < demin
    de = demin;
end

% Finally, prepare the control data array for the MEX file - DOUT (Data OUT)
               % This should be the last line in the custom code

DOUT = [0.0, de, 0.0, DIN(fslink_Throttle)];
```

8.2.6 Restarting the Flight Simulator - MATLAB Interface Component

If during step 3, executing a Script, MATLAB fails to display the shown output, this situation indicates a network connectivity problem. In this case, check the LAN configuration. Additional steps that may resolve connectivity problems include:

1. Manually cleaning up the network connection held by the MEX component by typing the following commands at the MATLAB command prompt:

```
fslink(fslink_close);
```


2. Removing the MEX external interface component from MATLAB's memory by typing the following commands:

```
clear fslink;
```

The fslink MEX component is automatically reloaded when the flight control script is next executed.

8.3 Developer Guide

Use this guide if you intend to further develop or understand the implementation of the MEX interface between the flight simulator and MATLAB.

8.3.1 Technical Description of the Simulator – MATLAB Interface

This interface has been developed using MATLAB MEX files. MEX files extend MATLAB by providing access to functions developed in C/C++ or FORTRAN to act as if they are built-in MATLAB functions. As outlined in **Section 2, System Architecture** the flight simulator subsystems are distributed over a dedicated local network, which includes the MATLAB PC. The MATLAB component has been network enabled by developing a custom MEX function in C to support UDP networking primitives, including the necessary connection control and synchronisation required by the flight simulator networking protocol. This approach simplifies the M-FILE scripts and also avoids the need to use additional MATLAB toolboxes.

8.3.2 MEX File Interface Implementation

Developers interested in extending the functionality found within the **fslink.mexw64** MEX file should read this section. A source code *walk-through* follows, to describe the functionality provided in the MEX file. The interface implementation is found in the C file **fslink.c**, along with the associated header file **fslink.h**.

The entry point to a MEX file function is via the **mexFunction()** function. MEX files usually act as a single function within MATLAB, called in a script by the name of the MEX file without the extension. Refer to the MATLAB documentation for detailed description of the mexFunction() parameters. These parameters provide a means of setting the number of inputs and outputs for a function. The input and output parameters ***plhs** and ***prhs** are MATLAB mxArrays in which function inputs and outputs are populated. MEX files can only pass data to and from a MATLAB script using these arrays.

```
void mexFunction (int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[])
```

Within the mexFunction, a *switch yard* provides a multi-function capability in comparison with the usual single function MEX file. The first element of the input array, **prhs**, contains an identifier defining the sub-function to call. These identifiers are C macros, defined at the top of the file, corresponding to the function identifiers defined in **simctrl-constants.m**.


```
#define MEX_UDP_START 1
#define MEX_UDP_STOP 2
#define MEX_UDP_SEND 3
#define MEX_UDP_RECV 4
#define MEX_UDP_DATASET 5
#define MEX_UDP_DATAGET 6
```

The switch yard, shown below, calls the sub-function corresponding to the **prhs[0]** element. These sub-functions are regular C functions. In this case, these sub-functions deal with the UDP networking protocol used to connect to the flight simulator. Each sub-function can take different parameters, but it is important to remember which parameters map to each element in the input array **prhs[]**.

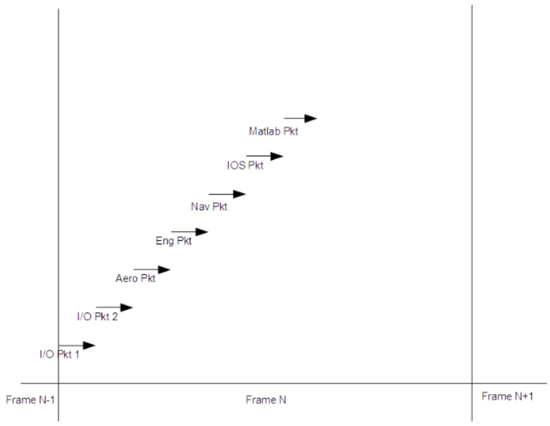
```
functionId = (int)mxGetScalar(prhs[0]);
switch(functionId) {
        case MEX_UDP_START:
                mexPrintf("Calling Function MEX_UDP_START\n");
                port = (unsigned short)mxGetScalar(prhs[1]);
                mexPrintf("Port : %d\n", (int)port);
                rv = UDP_Start(port);
                mexPrintf("rv = %d\n", rv);
                break;
        case MEX_UDP_STOP:
                mexPrintf("Calling Function MEX_UDP_STOP\n");
                UDP_Stop();
                break;
        case MEX_UDP_SEND:
                rv = UDP_Send();
                break;
        case MEX_UDP_RECV:
                rv = UDP_Recv();
                break;
        case MEX_UDP_DATASET:
                /* create a pointer to the real data in the input matrix  */
                inMatrix = mxGetPr(prhs[1]);
                /* get dimensions of the input matrix */
                ncols = mxGetN(prhs[1]);
                UDP_Data_Set(inMatrix);
                break;
        case MEX_UDP_DATAGET:
                /* create the output matrix */
                plhs[0] = mxCreateDoubleMatrix(1,(mwSize)MEX_OUTPUT_DATANUM,mxREAL);
                /* get a pointer to the real data in the output matrix */
                outMatrix = mxGetPr(plhs[0]);
                UDP_Data_Get(outMatrix);
                break;
}
```

To add further sub-functions, additional function identifiers can be created (C macros at the top of **fslink.c** and the equivalent constants in **simctrl-constants.m**) and their respective case blocks can be added to the switch yard.

The following functions deal with setting and acquiring data in a form that is compatible with MATLAB. The first function (UDP_Data_Set) populates the data in the return packet sent to the flight simulator and the second function (UDP_Data_Get) extracts data from the packets received from the flight simulator.

```
void UDP_Data_Set(double *data)
void UDP_Data_Get(double *data)
```


The remainder of the interface is composed of functions that handle the networking aspect of the interface. UDP_Start and UDP_Stop create and close the sockets required for receiving and transmitting UDP packets. The UDP_Send function transmits the ProtoPkt structure object that was populated via the UDP_Data_Set function. The most complex function, UDP_Recv, contains loops to ensure that, when receiving packets from the simulator, the MATLAB interface receives each of the flight simulator packets in the necessary order to maintain packet consistency across a single frame, or iteration, of the flight simulator. The first loop causes this function to block until the packet from I/O system 1 has been received. Once received, the MEX interface is now synchronised to the current simulation frame and then waits for packet from I/O system 2, followed by packets from the Aero, Eng, Nav and IOS computers. Note that this function will block until each of these packets has arrived. The following diagram (not to scale) shows the sequence of packets arriving at the MATLAB node in a single frame and also the transmission of the MATLAB packet.



In the event that the flight simulator unexpectedly stops with MATLAB mode enabled, it is likely that MATLAB will become blocked and will not respond to a Ctrl-C at the prompt. In such an event, MATLAB will require restarting.

```
int  UDP_Start(unsigned short);
void UDP_Stop(void);
int  UDP_Send(void);
int  UDP_Recv(void);
```

Developers needing to access additional variables from the flight simulator can expand the function UDP_Data_Get with the necessary assignments from the data available from the AeroPkt, NavPkt, IOSPkt and IOPkt and alter the macro MEX_OUTPUT_DATANUM to reflect the number of data elements being retrieved. This macro is used when allocating the MATLAB array used to provide the data for the calling M-FILE script.

8.3.3 Modifying and Recompiling the Mex File Interface

In the event that a change to the MEX file is required, for example, the MEX file requires new functionality, the modified C source files will need recompiling. A compiler compatible with MATLAB is required (refer to the MATLAB documentation for a list of compatible compilers). The supplied MEX file, **fslink.mexw64**, was compiled with a Windows version of the GCC (Gnu

Compiler Collection) C compiler. GCC is available for Windows through suites such as MinGW. The Matlab PC has a 64 bit version of MinGW installed.

To configure MATLAB to recognise your compiler, please refer to the MATLAB documentation. Comprehensive instructions exist for MS Visual Studio. However, there is minimal formal documentation for using GCC. A short description of this process now follows (which is only applicable to users of GCC on Windows). Note, these instructions only have to be executed once. Subsequent recompilations of the MEX file will not require these steps to be followed.

i. Locate the XML configuration file required to enable MATLAB to use the installed compiler. This file is found in the currently logged-in users profile data. This file has been configured to work with the current installation of MinGW.
E.g. C\Users\SIM8\AppData\Roaming\Mathworks\MATLAB\R2015a\mex_C.xml
Note, on the LFS, the user profile may be referred to as Cranfield and not SIM8.
ii. At the MATLAB command prompt, type:

```
mex –setup :C\Users\SIM8\AppData\Roaming\Mathworks\MATLAB\R2015a\mex_C++_mingw-w64.xml
```

Follow the instructions output at the MATLAB command prompt. You will be asked to choose which compiler MATLAB will use when building MEX files. Choose GCC. MATLAB is now configured to use GCC to compile MEX files.

To compile the flight simulator MATLAB interface MEX file (fslink.c), in MATLAB, change to the directory where the C source files are stored and type the following at the MATLAB command prompt.

```
mex '-IC:\mingw-w64\mingw64\x86_64-w64-mingw32\include' fslink.c C:\mingw-w64\mingw64\x86_64-
w64mingw32\lib\libwsock32.a
```

If your installation of GCC resides elsewhere, you will have to alter the above line to refer to the correct *include* directory and the correct Winsock library. If no errors are encountered, a new fslink.mexw64 file will have been created. This module can now be used as outlined in Section 8.2.

9 MAINTENANCE

9.1 Weekly Maintenance

Scheduled servicing of the LFS is not required as a result of the technology used in the simulator. However, to ensure reliable long service life, the following items should be completed on a daily to weekly basis dependent on the utilisation of the flight simulator.

WARNING: Before any maintenance is performed, the unit must be disconnected from the AC mains supply.

9.1.1 Mouse
The instructor station mouse should be cleaned at regular intervals to ensure correct operation. If the mouse performs erratically or intermittently, follow the instructions in the user manual for the PC.

9.1.2 Monitors
The IOS monitor and the five LCD monitors should be cleaned regularly as they can attract dust. A proprietary cleanser designed for use on computer terminals should be used. No moisture or water based cleaners should be used as the monitor could be damaged. Under no circumstances should any abrasive cleanser be used.

9.1.3 Flight Controls
If at any stage the integrity of the flight controls or engine levers is suspected, the diagnostic test program, described in Section 10, should be run to check the individual inputs for calibration.

9.2 Mechanical Adjustments

It is recommended that the mechanical controls are only serviced by authorised personnel. However, the majority of the transducers and switches are standard parts and may be replaced locally, provided that:

- Care is taken to isolate the AC mains supply.
- The approved replacement part (and value) is used.
- The polarity of the replacement part is the same as the original part.

A few transducers are provided with mechanical adjustment to accommodate the linkage mechanism or to cater for asymmetric movement about an operating point. In these cases, the replacement part must be set so that the mid-point of travel of the linkage corresponds to the mid-point of the transducer displacement.

After replacement of a transducer, the 'analogue' part of the diagnostic test should be executed to check the polarity, full range input and sensitivity. Similarly, the 'digital' part of the diagnostic test should be undertaken if any switches are replaced.

9.3 Calibration

The interface cards at the back of the bench in the simulator enclosure include a set of adjustments for 16 analogue inputs. If these channels require adjustment as the result of weekly maintenance checks or because a part is removed or replaced, the following procedure should be followed carefully to recalibrate these inputs.

The interface card is shown in Appendix A-3. There are two banks of trim potentiometers for the 16 analogue input channels P01-P16. Each channel has two potentiometers, one for a gain adjustment and one for an offset adjustment.

The trimming potentiometers are allocated as follows:

Channel	Offset	Gain
P01	VR1	VR2
P02	VR3	VR4
P03	VR5	VR6
P04	VR7	VR8
P05	VR9	VR10
P06	VR11	VR12
P07	VR13	VR14
P08	VR15	VR16
P09	VR17	VR18
P10	VR19	VR20
P11	VR21	VR22
P12	VR23	VR24
P13	VR25	VR26
P14	VR27	VR28
P15	VR29	VR30
P16	VR31	VR32

These potentiometers are relatively sensitive. For each channel, the gain should be adjusted towards full deflection. The offset is then adjusted for the end-of-travel positions. This process is repeated until full-scale deflection results from full travel of the channel transducer.

If adjustment for a channel is such that the channel cannot be 'located' by means of the calibration test software, turn the gain potentiometer fully clockwise and the offset potentiometer fully anti-clockwise and repeat the above process.

NOTE: It is advisable to ground unused analogue inputs to minimise noise.

10 DIAGNOSTICS

The LFS operates with standard PCs and commercial off-the-shelf cards. A simple set of diagnostic programs is provided to enable an operator to run confidence tests and to isolate specific problems.

10.1 Preliminary Checks

When the AC power is switched on, the PCs should display the Windows10 desktop.

- Check the power connections to the PCs;
- Check the power connection the cockpit;
- Check the power connection to the IOS terminal;
- Check the power connection to the Ethernet switch;
- Check the power connection to the KVM switch;
- Check the power connection to the projectors.

Check that the KVM switch can switch between systems. The PC desktops should show the common desktop icons and also the MinGw icon. If the desktop is not displayed or the PC does not respond to mouse or keyboard input, run diagnostic software for the PC.

To check the I/O system, check that the Raspberry Pi interface box is switched on and press the reset button. Run *putty* from a PC and select an IP address of 192.168.1.1 for RPi1 or 192.168.1.2 for RPi2. The RPi I/O system should display a Linux terminal inviting users to log on. Enter the user name *pi* and the password *raspberry*.

10.2 Software Checks

For each PC, double click the MinGw icon. A Linux-like terminal should appear. Change to the main directory /c/sim and check that the simulator files (see section 6) are visible, using the Linux *ls* and *cd* commands.

10.2.1 Networking

For each PC, type:

>*ipconfig*

The IPv4 address should be 192.168.1.3 through to 192.168.1.10 for the eight PCs. If the Ethernet IPv4 mode has been set to DHCP, reset it to the corresponding static IP address given at the start of Section 5. Also check that the firewall has been disabled.

For the RPi I/O systems, type:

>*ifconfig*

Check that the IP address is 192.168.1.1 for RPi1 and 192.168.1.2 for RPi2 and if not, reset it to this static address. The IP addresses are defined in the file /etc/dhcpcd.conf, Note that the flight simulator PCs run with the IPv4 protocol enabled, all other protocols disabled and with the Windows firewalls disabled.

To check that the Ethernet connections are functioning correctly, type:

>ping 192.168.1.n

Where *n* is the network number of each computer shown by ifconig or ipconfig. The 'ping' should show that transfers can be made to and from different computers on the network.

If these tests fail, check the Ethernet connections at the Ethernet switch and at the back of each PC.

10.2.2 Graphics Tests

Generally, the Windows10 desktop is a good test of the graphics cards and drivers.

Note that the main simulator PCs (excluding the Matlab PC) should be set to a screen resolution of 1024x768.

In each directory of the flight model PC, the engine model PC and the navigation/avionics PC, there is a program *demo.exe* which can be executed to test the graphics. This software does not use any Ethernet transfers and simply tests the real-time performance of the graphics.

Similarly, demo.exe, a version of the IOS, can be used in the /c/sim/ios/ios directory to check that the graphics, mouse and keyboard are operating correctly.

For the IG systems, run *osgviewer* with a suitable .flt file to check the graphics functionality of the three IG systems.

10.2.3 Sound System

In Windows10, it is straightforward to test the sound system. For the engine model PC, select the Control Panel, select Sound, select Sounds, then select a particular sound (e.g. Asterisk) and click on Test. If this test fails, check that power is applied to the sound amplifiers, the audio connectors are in place and that the volume is set to an appropriate level.

To test the flight simulator sounds, for the engine model PC, type:

```
>cd /c/sim/eng/sound/
>./soundtest
```


The range of sounds generated by the sound card can be tested by entering the following simple commands:

A	Airconditioning ON/OFF
C	Configuration warning ON/OFF
E	Electrical noise ON/OFF
F	Fire warning ON/OFF
G	Gear warning ON/OFF
H	Toggle HOLD ON/OFF
I	Ident <3-4 chars>
J	Jet Engine <EngineNo> <Rpm%%>
M	Middle marker ON/OFF
O	Outer marker ON/OFF
P	Piston Engine <EngineNo> <Rpm%%>
Q	Quit
R	Rumble <v> kts
S	Stall warning ON/OFF
T	Turboprop Engine <EngineNo> <Rpm%%>
U	Undercarriage <Position%%> <Speed>
W	Wind <n> kts
Z	Gear Motor
+	Increase engine RPM
-	Decrease engine RPM
<	Increase airspeed
>	Decrease airspeed
!	Toggle reverse thrust mode>sndtest2

Where the values in angle brackets are appropriate numerical values. For example, to generate the sound of jet engine 1 at 95% RPM, type: `j 1 95`

Check that the speaker system has power, is correctly configured, the volume level is set correctly and that the phono lead from the sound card is connected to the speaker system.

10.3 I/O Cards

The RPi computers are 'headless', that is to say, they will start automatically and execute software to participate in the simulator network transfers. In order to run the diagnostic tests, it is necessary to kill these processes. For each RPi, using a *putty* terminal, enter:

```
>ps –e
```

A list of process should appear. Note the process number for the *iosystem* process, for example 123, and execute the following command:

```
>sudo kill 123
```

It is essential to kill these processes before running the diagnostics.

Two test programs are provided to test each I/O system:

10.3.1 Analogue and Digital Inputs, Digital Outputs Tests

For the RPi1 computer, using a putty terminal, type:

```
>cd sim/calibrate
>./iotest
```

For the flight model PC (FLT), type:

```
>cd /c/sim/pfd/calibrate
>./iotest
```

Note that the RPi command should be executed before the PC command. The Flight Model PC should show the display in Figure 10.3.1.

To stop the test, press the ESC key at the flight model PC and then enter ctrl-C at the RPi computer. Following the instructions in section 9.3, check the channel allocation and calibration of the analogue channels and the polarity and functionality of the digital channels, given in sections 10.3.4-10.3.7. Calibrate any analogue input channels as necessary.

For the RPi2 computer, using a putty terminal, type:

```
>cd sim/calibrate
>./iotest
```

For the engine model PC (ENG), type:

```
>cd /c/sim/pfd/calibrate
>./iotest
```

For these tests, the RPi captures the inputs and the flight model PC displays the values in the respective cockpit displays. This arrangement enables one person to adjust the analogue calibration while a second person can monitor the values seen in the cockpit and advise on adjustments. A copy of the display can be obtained by pressing the ‘P’ key.

Channels P01 to P16 have gain/offset adjustment. The four digit values in white indicate the instantaneous value of the channel in the range 0 to 4095. The three digit values in yellow indicate the channel noise level. A value greater than 50 implies an unacceptable level of interference or noise in an analogue channel, which may adversely affect the fidelity of the flight simulator.

In Figure 10.3.1, channel P04 is indicating 1012. Note the mid-point mark, which corresponds to 2048. For the digital inputs, red corresponds to +5V and blue corresponds to 0V. In Figure 10.1.3, channel B2 is 0V and channel C2 is +5V.

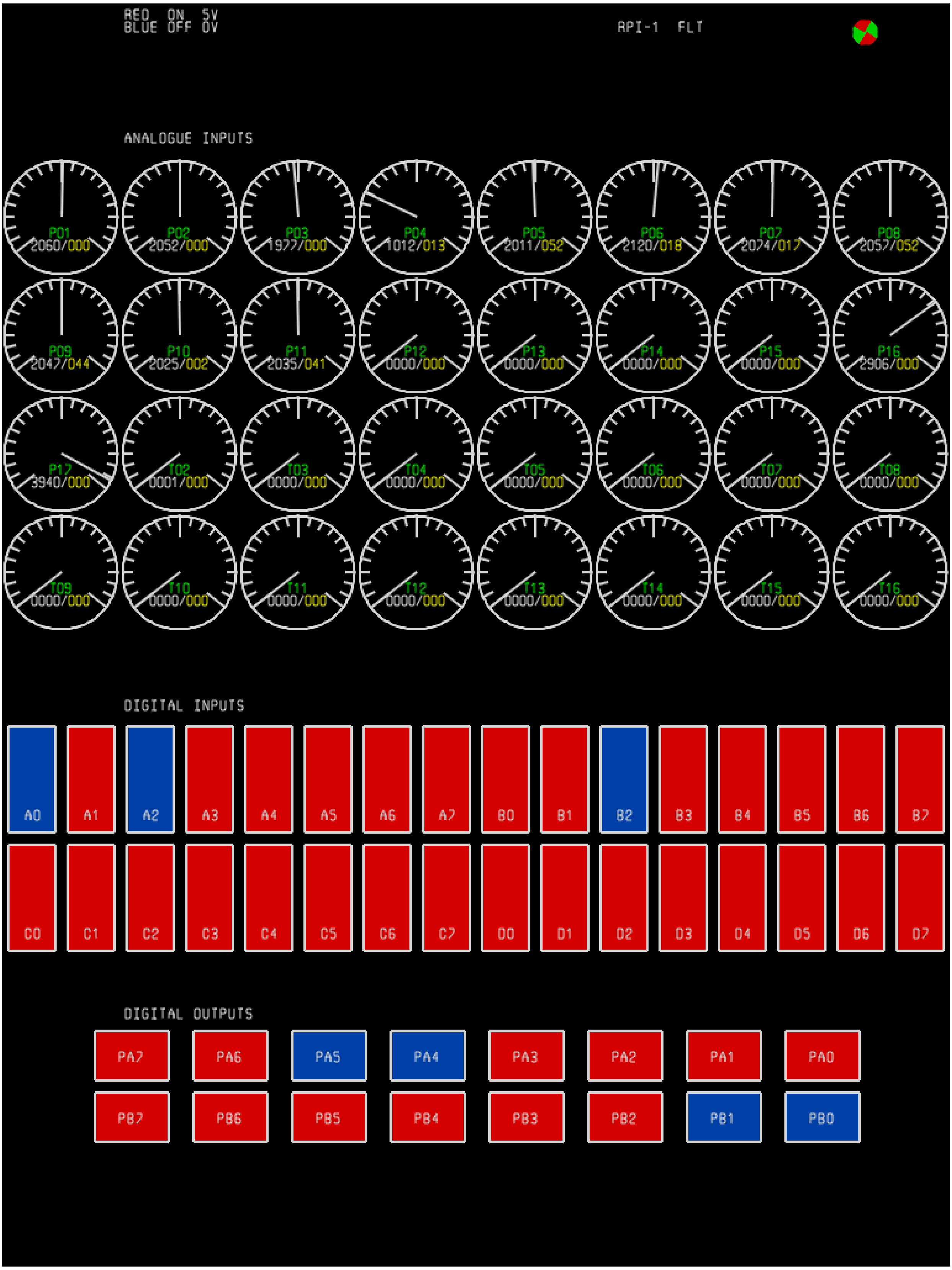


Figure 10.3.1 Diagnostics Display

At the top of the digital display, a small rotating red and green icon indicates that the analogue and digital channels are being sampled. If this icon is not rotating, no samples are being acquired; in this case, it is likely that the program has failed or there is an interconnection problem.

Check that the 50-way ribbon cable from the break-out card is connected to CON1 and the 20-way ribbon cable is connected to CON2 inside the RPi interface box.

10.3.2 *Analogue Inputs Noise Test*

This test measures the level of noise on the 32 analogue input channels. It should be run if *iotest* shows unacceptable levels of noise to confirm the noise levels.

For the RPi1 system, type:

```
>cd sim/rpi1/calibrate
>./iotest
```

For the flight model PC (FLT), type:

```
>cd /c/sim/pfd/noise
>./noisetest
```

Similarly for the RPi2 system type:

```
>cd sim/rpi2/calibrate
>./iotest
```

For the engine model PC (ENG), type:

```
>cd /c/sim/eng/noise
>./noisetest
```

The display is shown in Figure 10.3.2 where the noise is displayed for each channel, which shows noise levels up to a maximum value of 25. Values less than 3 should be observed. In this example, channels P04, P05 and P09 (corresponding to the channel numbers in section 10.3.5) show unacceptable noise levels. The vertical white line traverses the display to show the current sampling position. The display shows 700 frames (14s) and rolls around continuously. To stop the test, press the ESC key at the PC and enter ctrl-C at the RPi. A copy of the display can be obtained by pressing the ‘P’ key.

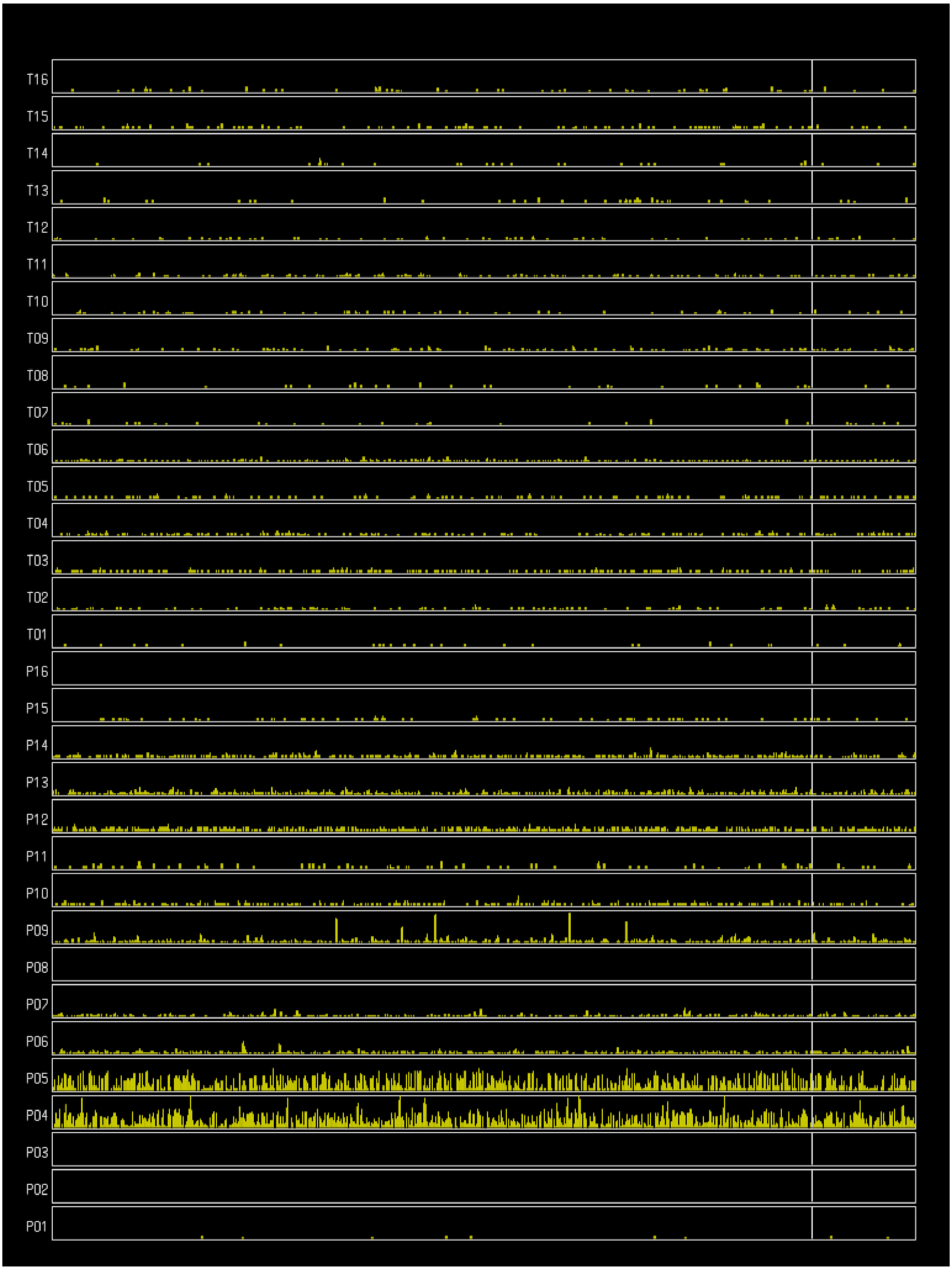


Figure 10.3.2 Channel Noise Display

10.4 Interface Connections

Two I/O systems, known as RPi1 for the FLT computer and RPi2 for the ENG computer, provide connections for analogue inputs, digital inputs and digital outputs. These values are summarised in the following tables.

10.4.1 RPi1 Analogue Inputs

Item	**Diagnostics**	**Channel**
Elevator	P01	0
Aileron	P02	1
Rudder	P03	2
Stab trim	P04	3
Rudder trim	P05	4
LHS sidestick elevator	P06	5
LHS sidestick aileron	P07	6
RHS sidestick elevator	P08	7
RHS sidestick aileron	P09	8
LHS nose wheel steering	P10	9
RHS Nose wheel steering	P11	10
LHS left brake	P12	11
LHS right brake	P13	12
RHS left brake	P14	13
RHS right brake	P15	14
Speed brake	P16	15
Flap	TO1	16
	T02	17
	T03	18
	T04	19
	T05	20
	T06	21
	T07	22
	T08	23
	T09	24
	T10	25
	T11	26
	T12	27
	T13	28
	T14	29
	T15	30
	T16	31

Note:

- All input channels are 0-5V
- Inputs P01-P16 are adjustable (gain/offset)
- Channels T02-T16 are unused

10.4.2 RPi1 Digital Inputs

Item	Diagnostics
Lower right wing	A7
Lower left wing	A6
Elevator trim nose down	A5
Elevator trim nose up	A4
Speed-brake armed	A3
Gear down	A2
Gear up	A1
Park brake	A0
	B7
	B6
	B5
	B4
	B3
Red button (HOLD)	B2
Yellow button (FREEZE)	B1
Green button (RESET)	B0
	C7
	C6
	C5
	C4
	C3
	C2
	C1
	C0
	D7
	D6
	D5
	D4
	D3
	D2
	D1
	D0

Note:

- Red = 5V, Blue = 0V
- Channels B3-B7, C0-C7 and D0-D7 are unused

10.4.3 RPi1 Digital Outputs

Item	Diagnostics
Stab trim nose down	A7
Stab trim nose up	A6
Gear lever solenoid	A5
Gear down lamp (green)	A4
Door open lamp (red)	A3
Gear transit lamp (red)	A2
Stick shaker	A1
Spoiler solenoid	A0
	B7
	B6
	B5
	B4
RHS brake release lamp (amber)	B3
LHS brake release lamp (amber)	B2
Ground idle lamp (amber)	B1
Ground band lamp (amber)	B0

Note:

- Red = 5V, Blue = 0V
- Channels B4-B7 are unused

10.4.4 RPi2 Analogue Inputs

Item	Diagnostics	Channel
Throttle 1 forwards	P01	0
Throttle 1 reverse	P02	1
Throttle 2 forwards	P03	2
Throttle 2 reverse	P04	3
Throttle 3 forwards	P05	4
Throttle 3 reverse	P06	5
Throttle 4 forwards	P07	6
Throttle 4 reverse	P08	7
	P09	8
	P10	9
	P11	10
	P12	11
	P13	12
	P14	13
	P15	14
	P16	15
	T01	16
	T02	17
	T03	18
	T04	19
	T05	20
	T06	21
	T07	22
	T08	23
	T09	24
	T10	25
	T11	26
	T12	27
	T13	28
	T14	29
	T15	30
	T16	31

Note:

- All input channels 0-5V, 12 bits 0-4095
- Inputs P01-P16 are adjustable (gain/offset)
- P01/P02, P03/P04, P04/P05 and P06/P07 are common
- Channels P09-P16 and T01-T16 are unused

10.4.5 RPi2 Digital Inputs

Item	Diagnostics
Engine 4 inflight start	A7
Engine 3 inflight start	A6
Engine 2 inflight start	A5
Engine 1 inflight start	A4
Engine 4 ground start	A3
Engine 3 ground start	A2
Engine 2 ground start	A1
Engine 1 ground start	A0
Engine 4 fire pull	B7
Engine 3 fire pull	B6
Engine 2 fire pull	B5
Engine 1 fire pull	B4
Engine 4 fuel switch	B3
Engine 3 fuel switch	B2
Engine 2 fuel switch	B1
Engine 1 fuel switch	B0
APL Nose down	C7
APL Nose up	C6
No 3 stab trim cutout	C5
No 2 stab trim cutout	C4
Throttle 4 switch	C3
Throttle 3 switch	C2
Throttle 2 switch	C1
Throttle 1 switch	C0
Auto-throttle button	D7
Auto-pilot button	D6
Flight Director button	D5
Go-around button	D4
	D3
Warning horn cut out	D2
Fire button push	D1
Autopilot disconnect	D0

Note:
Red = 5V, Blue = 0V
Channel D3 is not used

10.4.6 RPi2 Digital Outputs

Item	Diagnostics
Fire handle 4 lamp	A7
Fire handle 3 lamp	A6
Fire handle 2 lamp	A5
Fire handle 1 lamp	A4
Engine 4 reverse solenoid	A3
Engine 3 reverse solenoid	A2
Engine 2 reverse solenoid	A1
Engine 1 reverse solenoid	A0
	B7
	B6
	B5
	B4
	B3
Glare-shield fire lamp	B2
Auto-throttle forwards	B1
Auto-throttle backwards	B0

Note:
Red = 5V, Blue = 0V
Channels B3-B7 are not used

10.4.7 I/O System Interface

50-way Connector	Break-out Board	Function	Notes
X1		analogue ground	
X2	6/IC15	analogue input	
X3		analogue ground	
X4		not connected	
X5		analogue ground	
X6		not connected	
X7		analogue ground	
X8		not connected	
X9		analogue ground	
X10	CA0/IC2	analogue multiplexer S_0	
X11		-12V	
X12	CA1/IC2	analogue multiplexer S_1	
X13		-12V	
X14	CA2/IC2	analogue multiplexer S_2	
X15		-12V	
X16	CA3/IC2	analogue multiplexer S_3	
X17		-12V	
X18	CA4/IC3	analogue multiplexer S_4	
X19		-12V	
X20		not used	
X21		digital ground	
X22	~G2B/IC10	digital selector	always 0
X23		digital ground	
X24	~R/IC6-9	digital read/write dir	always 0 (digital input)
X25		digital ground	
X26	~G2A/IC10	digital selector	always 0
X27		digital ground	
X28	A/IC10	digital mutiplexer S_0	00=P101-108 01=P109-116
X29		digital ground	
X30	B/IC10	digital mutiplexer S_1	10=P117-124 11=P125-132
X31		+5V	
X32	C/IC10	not required	always 0
X33		+5V	
X34	D0/IC1	data input D_0	
X35		+5V	
X36	D1/IC1	data input D_1	
X37		+5V	
X38	D2/IC1	data input D_2	
X39		+5V	
X40	D3/IC1	data input D_3	
X41		+12V	
X42	D4/IC1	data input D_4	
X43		+12V	
X44	D5/IC1	data input D_5	
X45		+12V	
X46	D6/IC1	data input D_6	
X47		+12V	
X48	D7/IC1	data input D_7	
X49		+12V	
X50		not connected	

10.5 Raspberry Pi I/O System

The Raspberry Pi I/O system is shown in Figure 10.4. The Raspberry Pi has a dedicated I/O system based on I2C devices for analogue input, digital input and digital output.

In addition, the breakout card provides multiplexing of analogue inputs, digital inputs and digital outputs, as shown in Figure 10.4 . The 32 analogue inputs are multiplexed by a 5 bit digital output. A single channel I2C 12-bit ADC samples the analogue input. Similarly, the 32 digital inputs are multiplexed as 4 8-bit bytes by a 2-bit digital output.

The Raspberry Pi also uses the Linux real-time clock to synchronise to a 20ms frame. At the start of each frame, RPi1 I/O system broadcasts its packet which provides a timing datum for all the other computers.

Each RPi I/O system also provides up 16 digital outputs to drive cockpit lamps, solenoids and electric motors. In addition the I/O system has eight dedicated LEDs which are either displaying a rolling pattern to indicate that the simulator is running or can be used to display values generated by user software (overriding the rolling pattern).

On start-up, the RPi attempts to initialise the I2C devices. Any errors are logged. During run-time, any error detected during I2C transfers are logged and the RPi terminates the I/O program. Run-time errors leading to a failure are critical and should be reported. In normal operation, the RPi system should run without any errors. If errors are detected, the calibration program described in section 10.3.1 provides a good confidence test.

The most likely cause of problems are that power supplies are not connected or switched on or that a connector is not securely attached or is missing. Note that, if the calibration test program fails, the simulator software will not execute.

The start-up file in /boot/config.txt is configured to initialise the I2C devices and sets the I2C fast transmission speed of 400,000 bytes/s. The main confidence test for the I2C devices is to execute the following command at a terminal:

```
>sudo i2cdetect –y 1
```

This test should provide a list of I2C device showing addresses 0x20, 0x21, 0x22, 0x23, 0x24, 0x48, 0x4d and 0x60. If any addresses are missing, the associated I2C device is likely to be faulty.

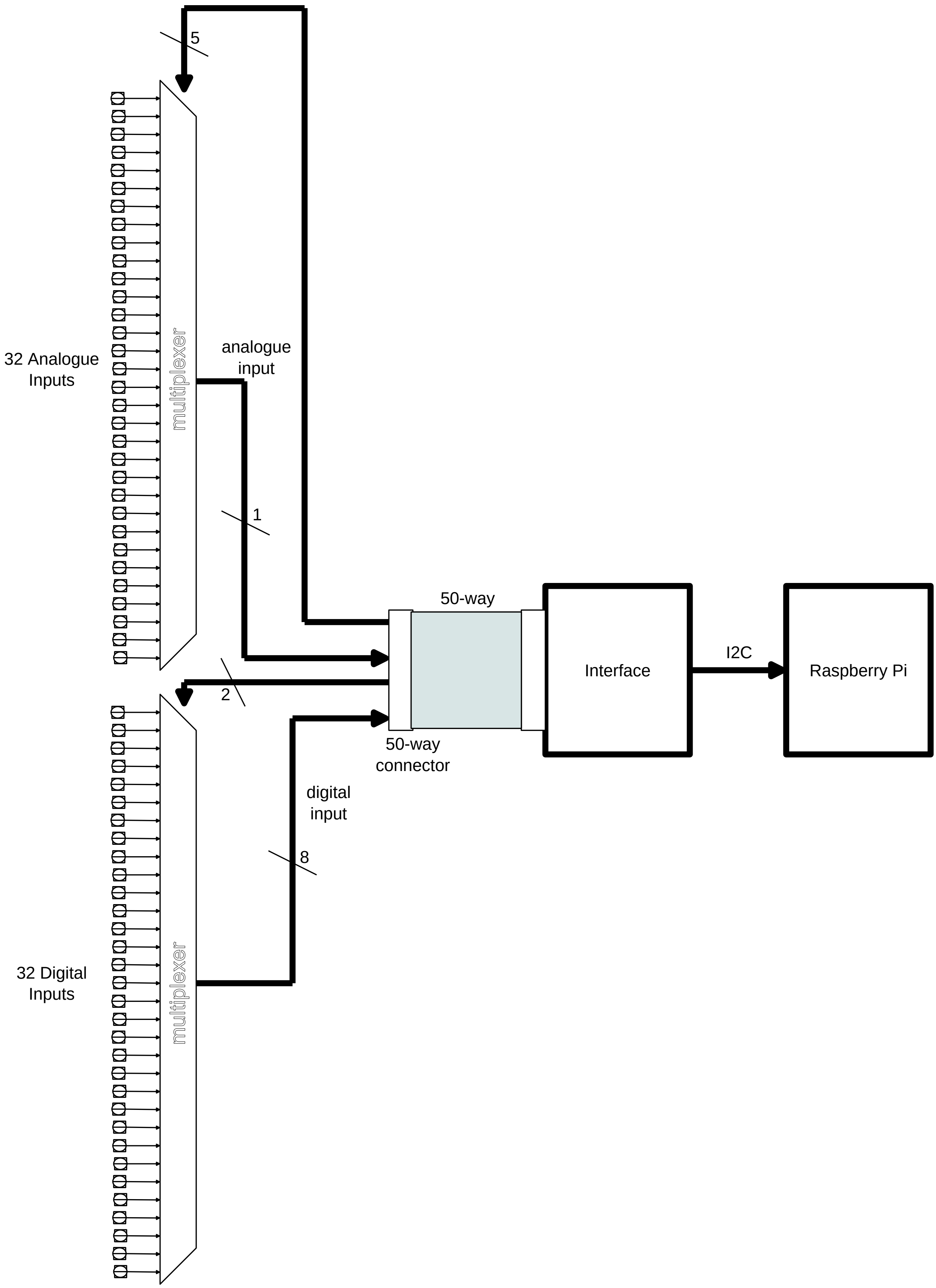


Figure 10.4 RPi I/O System Analogue and Digital Data Acquisition

10.5 Fault Finding

This section explains possible problems, their likely cause and probable remedy.

Many problems are likely to result from faulty connections or incorrect Windows settings. It is sensible to run Windows test software prior to running specific simulator tests.

The PCs do not come on when the AC switch is turned on. Check the AC mains supply is switched on. Check that the PCs, LCD panels and the IOS terminal are plugged into the internal AC supply and switched on. Check that the KVM switch is switched on and set to a specific channel.

The simulator appears to hang. Check the display on the RPi interface box. If the LEDs appear to be stuck, the packet transfer protocol has failed. See below.

The simulator stops after a specific number of packet transfers. If the simulation stops after a transfer, it is likely that the fault occurred in the next computer in the chain of transfers. With Windows, care is needed that the correct static IP addresses are selected, protocols other than IPv4 are disabled, the Firewall options are disabled and any background Windows process likely to invoke network transfers are disabled.

A particularly difficult situation to detect is the corruption of the real-time protocol. Insight can be gained by running Ethernet monitoring programs such as Ethreal on one of the IG computers or the Matlab computer, which can display packet transfers. UDP transfers should occur in the following sequence:

```
192.168.1.1
192.168.1.2
192.168.1.3
192.168.1.4
192.168.1.5
192.168.1.6
192.168.1.1
192.168.1.2
```
… and so on.

The visual system does not display a visual scene. Check the power supply to the IGs and projectors is switched on and the projectors have been started. Check the IG display on the monitor. Check the graphics connection between the three IG computers and the video splitters. Check the time of day setting as the IGs may start up in night-time conditions.

The instrument displays remain off. Check the power supply to the LCD panels. Check the graphics connection between the flight model computer and the left LCD panel, the engine computer ab the centre LCD panel and the navigation computer and the right LCD panel.

The simulator stops immediately. Check that all PCs and the RPi have started. Check the connections between the interface cards and the RPi I/O systems.

The instructor station mouse does not respond to commands. Check the mouse connections. Check that the flight model is operating. Check the Ethernet connections.

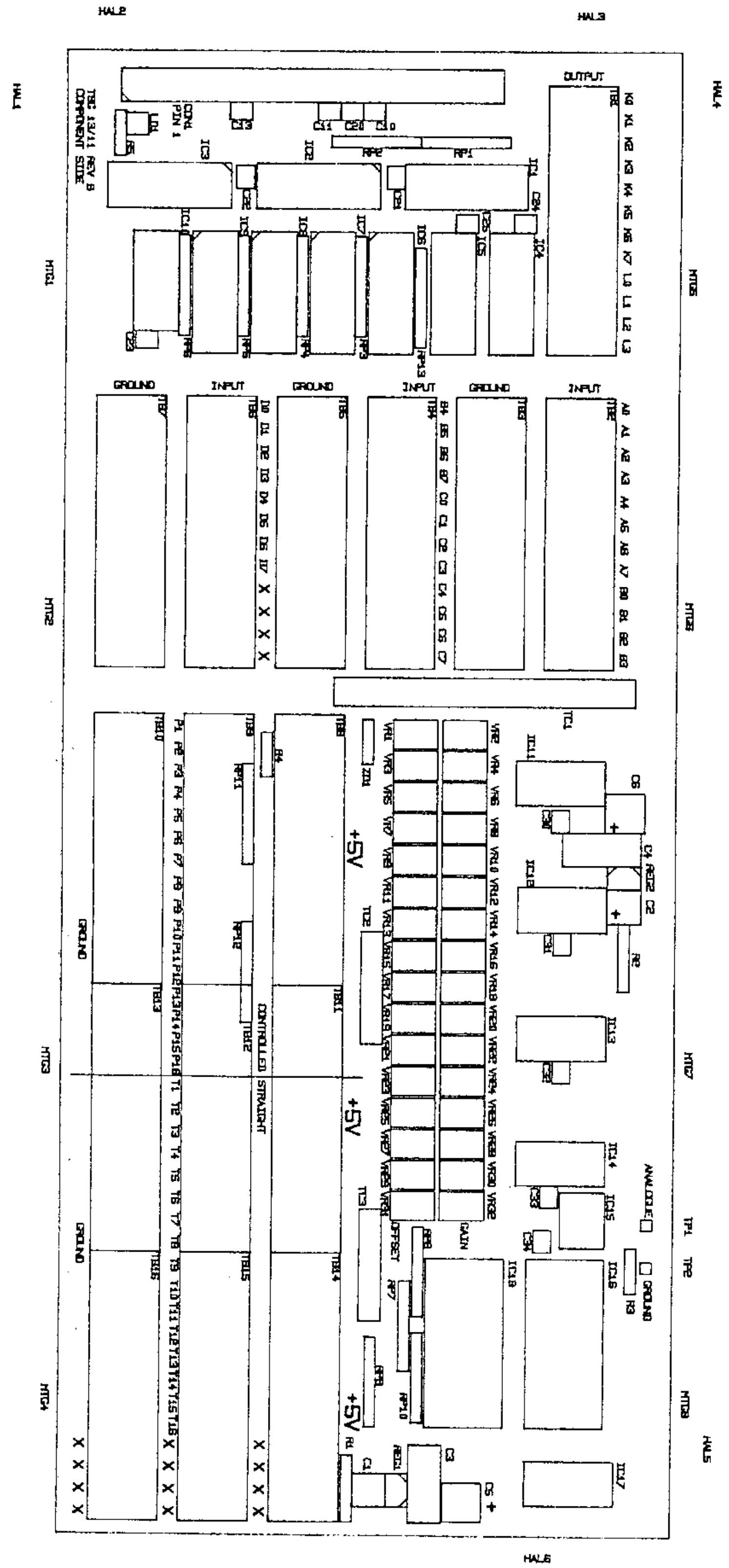
The engines do not respond to the start-up sequence. Check that the flight simulator is not set in the HOLD mode. (Check both the HOLD button and the instructor station HOLD command). Check that the instructor has not failed the engines. Check the engine switches.

The aircraft does not accelerate on the runway. Check the park brake. Check the toe brakes. Check the wind settings. Check that the IOS has restored the aircraft to a known location.

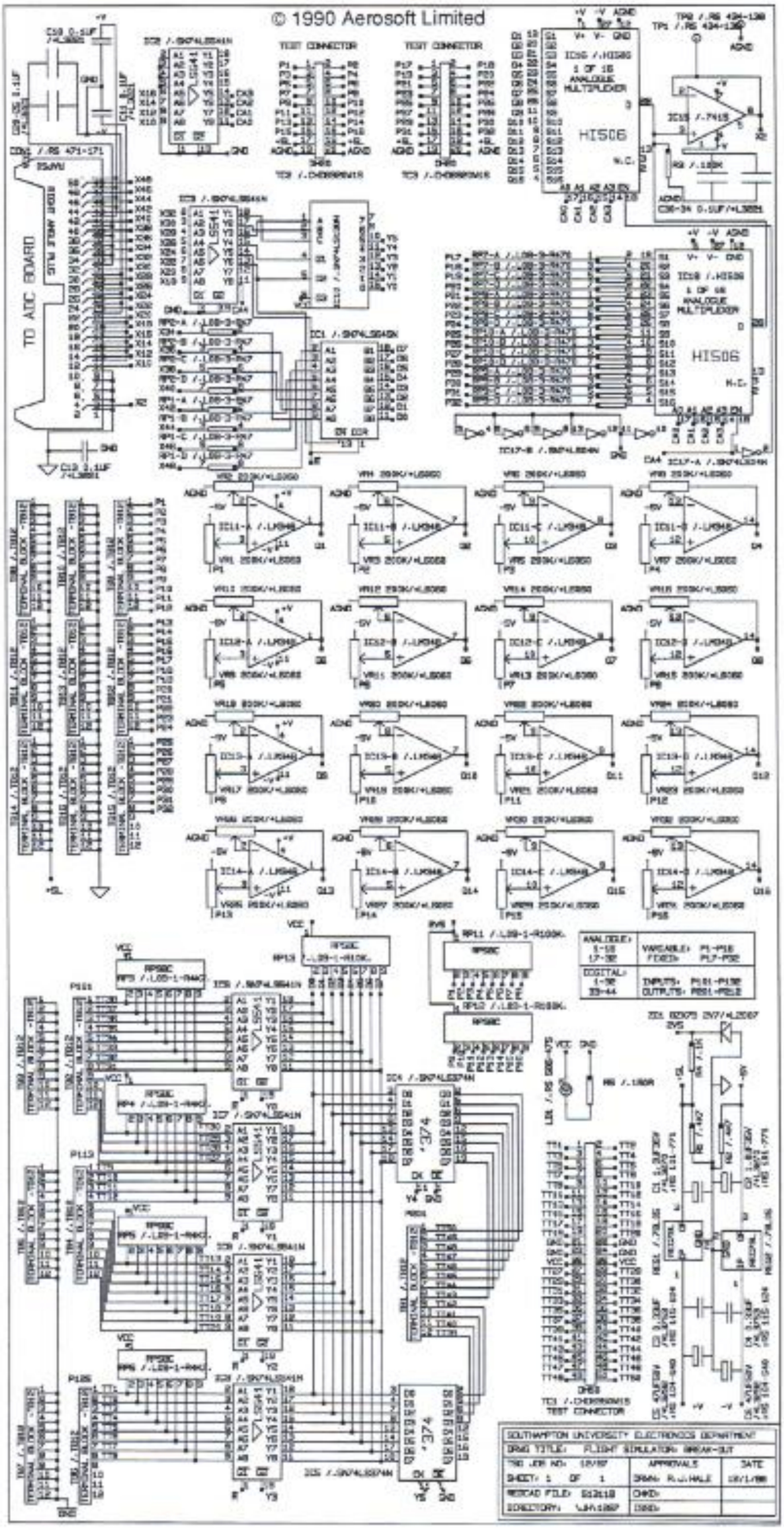
The navigation system does not respond to the selected frequency. Check the current navigation frequencies. Check the aircraft altitude for VHF facilities. Check that the instrument is not failed by the instructor.

An aircraft control input or switch does not function as expected. Run the diagnostic test for that particular control input or switch.

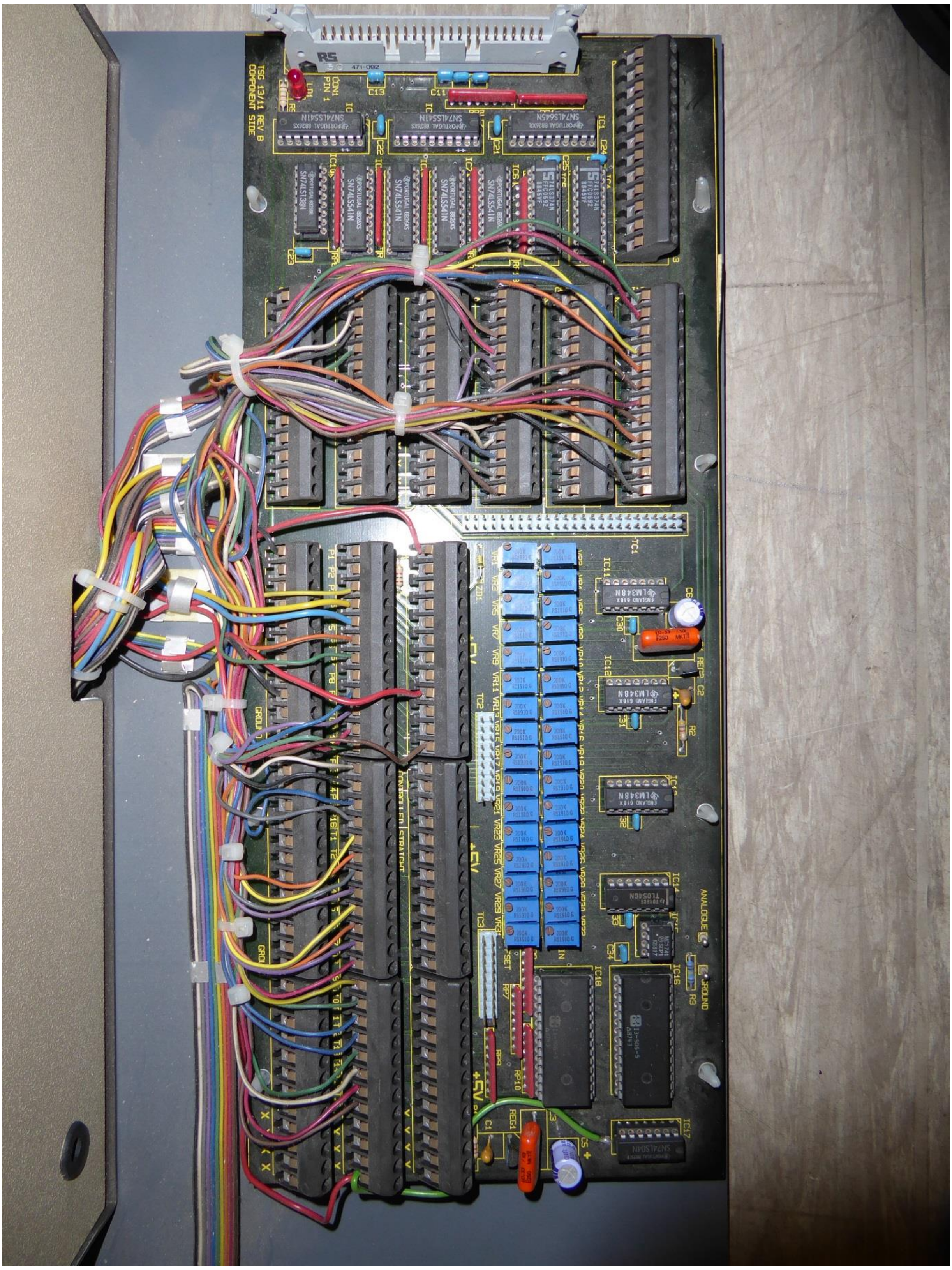
Appendix A-1 Breakout Card Layout



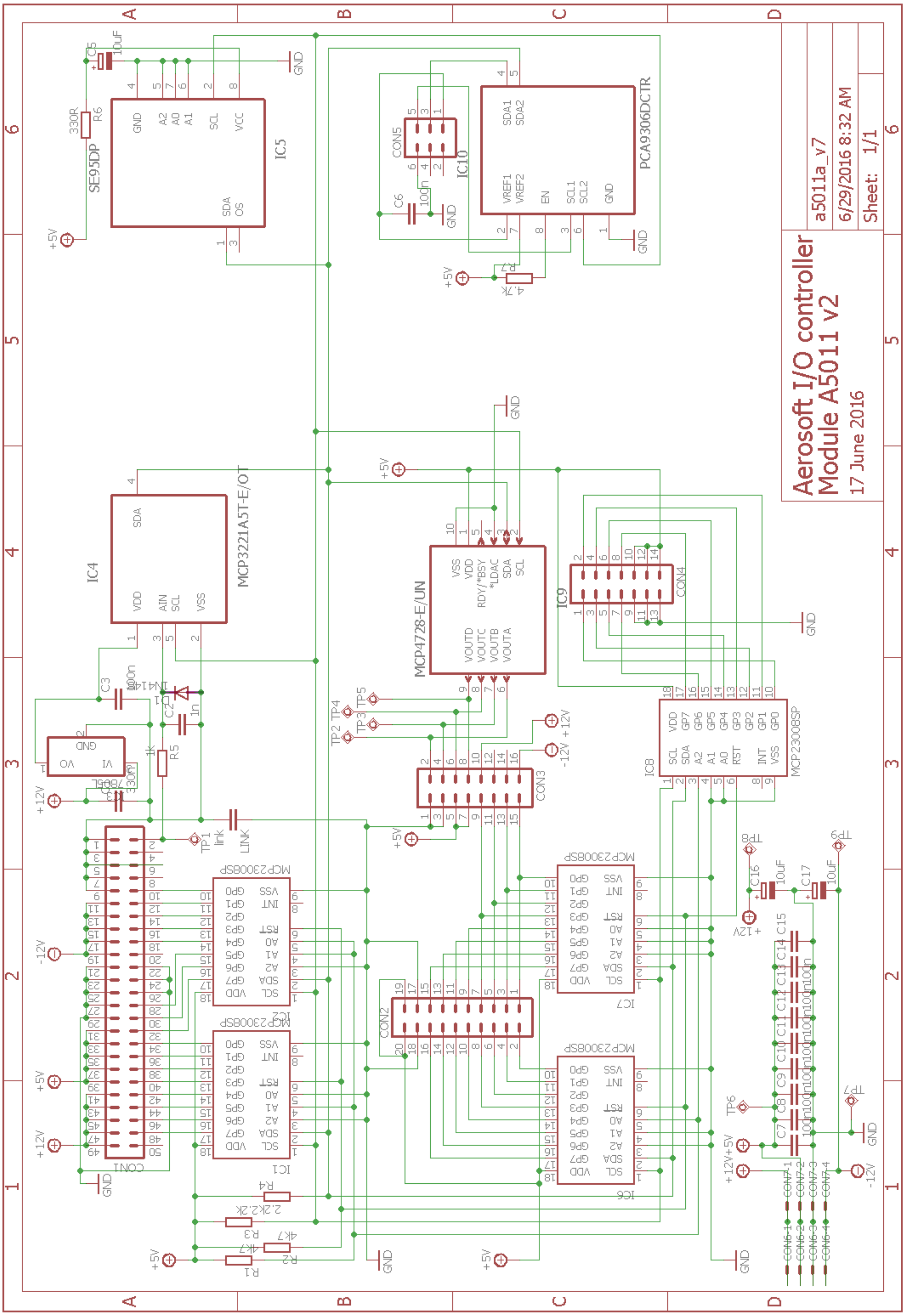
Appendix A-2 Breakout Card Schematic



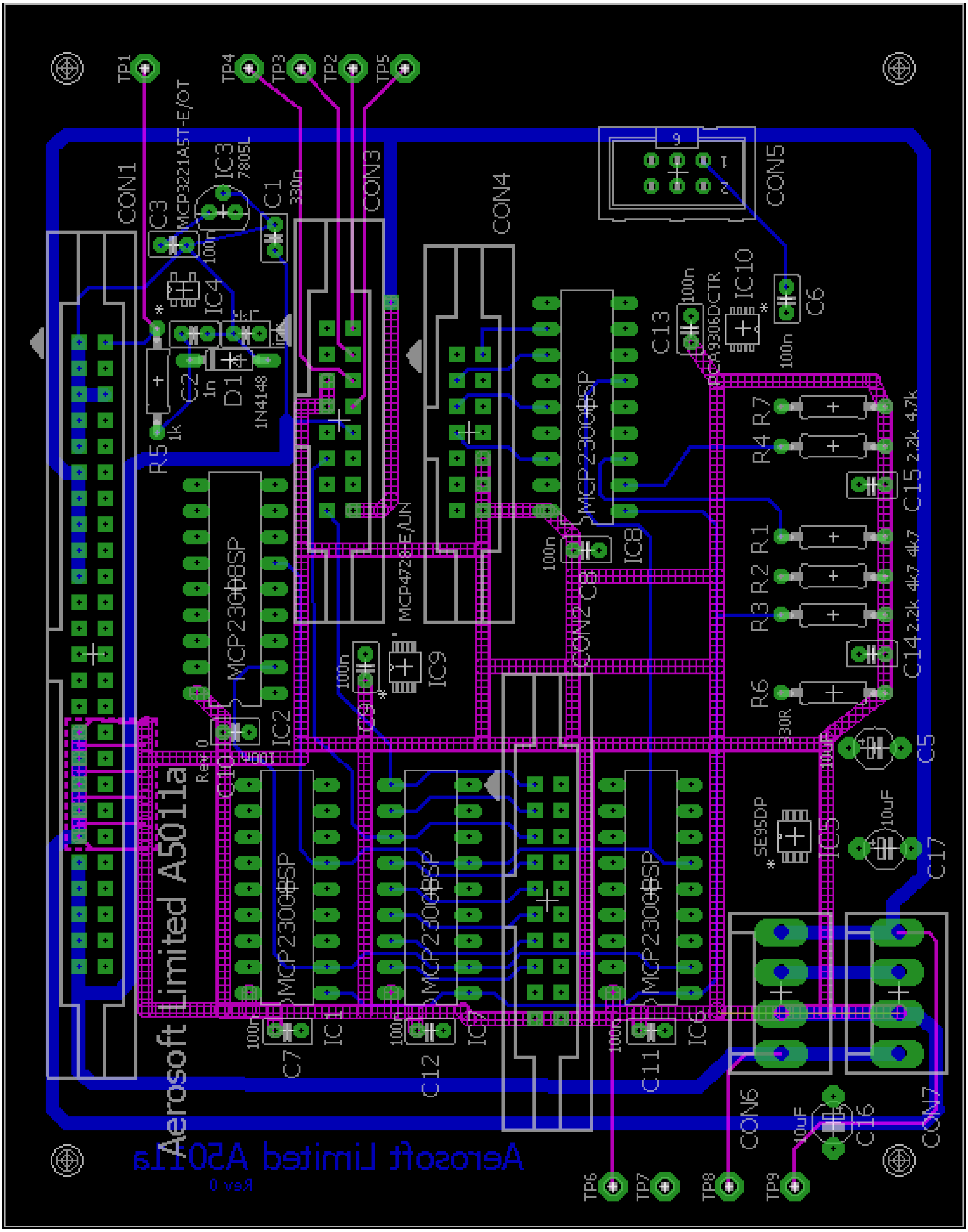
Appendix A-3 Breakout Card Photograph



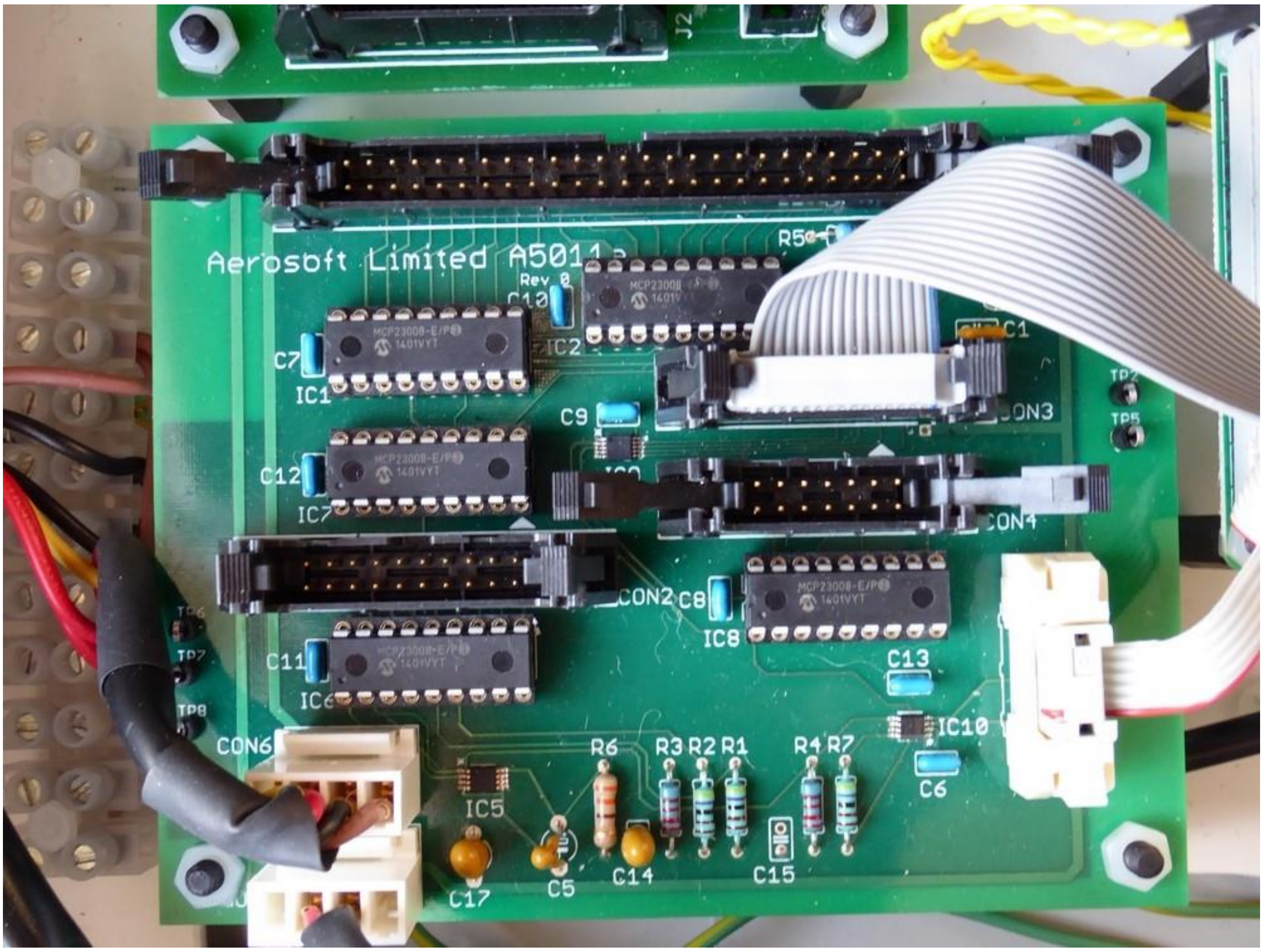
Appendix A-4 A5011a Interface Card Schematic



Appendix A-5 A5011a Interface Card PCB Layout



Appendix A-6 A5011a Interface Card Photograph



Note.
The ribbon cable, shown for CON3, is not required for the LFS I/O interface.

Appendix A-7 Interface Unit