AEROSOFT

Simplot

Author: D J Allerton
Version: 1.2
Date: 20th August 2017

1 Introduction

Simplot enables flight simulation models to be developed and tested off-line. The main flight dynamics modules (e.g. aerodynamics, engines dynamics, undercarriage dynamics, flight control systems and equations of motion) used in *Simplot* are identical to the modules used in the real-time flight simulator. Additional modules enable specific inputs to be applied to the flight model and data from a simulation to be recorded and displayed. The output from *Simplot* is a GNUPlot script, which can be opened in GNUplot to generate a .png file which can be imported into other packages, for example Microsoft Word.

2 The *Simplot* Language

A *Simplot* program can record simulator data, set simulator conditions, define inputs and initialise the state of the simulator. A *Simplot* program is written as a standard text file which can be edited using text editors such as Programmers Notepad (http://www.pnotepad.org/). *Simplot* programs are case insensitive and spaces and blanks lines are ignored. A *Simplot* program contains lines of commands, where each command includes optional arguments and where appropriate, arguments can be defined in specific units. Any characters after a semicolon are treated as a comment and ignored for the remainder of the line.

3 Basic Commands

Seven basic commands are provided to control the off-line simulation:

Command	Description
`set`	*Set a flight condition*
`plot`	*Plot a variable*
`engage`	*Engage an autopilot mode*
`disengage`	*Disengage an autopilot mode*
`time`	*Set the recording time*
`autotrim`	*Set the simulator inputs for steady-state conditions*
`input`	*Apply an input*

4 Set Command

The *set* command defines initial flight conditions including control inputs. The command has one argument which can be defined in specific units. The arguments are as follows:

Argument	Description	Range
`turbulence`	*Turbulence conditions*	0-100%
`wind_speed`	*Wind speed*	0-80 kt
`wind_direction`	*Wind direction*	0-360°
`QNH`	*QNH pressure setting*	950-1050 mbar
`mag_var`	*Magnetic variation*	0-360°
`OAT`	*Outside air temperature*	-50-50°C
`rudder`	*Rudder input*	± rudder range
`aileron`	*Aileron input*	± aileron range
`elevator`	*Elevator input*	± elevator range
`engine_lever`	*Engine lever input*	0-1
`altitude`	*Aircraft altitude*	0-50000 ft
`heading`	*Aircraft heading*	0-360°
`TAS`	*Aircraft true airspeed*	0-500 kt
`IAS`	*Aircraft indicated airspeed*	0-500 kt
`rate_of_climb`	*Aircraft rate of climb*	±5000 ft/min
`cg_position`	*Centre of gravity position (%)*	-100-100%
`flaps`	*Flap position*	0-1 (0-up, 1=fully down)
`gear`	*Gear position*	0-1 (0-up, 1=fully down)
`spoiler`	*Spoiler (air brake) position*	0-1 (0-down, 1=fully up)
`parkbrake`	*Park brake*	0-1 (0-off, 1=on)
`leftbrake`	*Left brake*	0-1 (0-off, 1=on)
`rightbrake`	*Right Brake*	0-1 (0-off, 1=on)
`AP_ALT`[1]	*Autopilot altitude*	units default to ft
`AP_HDG`[1]	*Autopilot heading*	units default to degrees
`AP_SPD`[1]	*Autopilot speed*	units default to kt
`AP_VSPD`[1]	*Autopilot vertical speed*	units default to ft/min
`Kp`	*PID control term*	n/a
`Ki`	*PID control term*	n/a
`Kd`	*PID control term*	n/a

[1] Autopilot units have no units but are implicit from their respective FCU panel units.

4.1 Examples of the set command

```
set altitude 5000 ft
set altitude 2000 m
set aileron 10 deg
set AP_HDG 035
set gear 0.667
```

Note: there is no consistency check on units, e.g. it is possible to set altitude to kg.

5 Plot Command

The *plot* command is used to add flight data to be recorded during the simulation. Each variable plotted can be defined in terms of the units and range. The arguments are as follows:

Argument	Description
`rudder`	*Rudder position*
`aileron`	*Aileron position*
`elevator`	*Elevator position*
`TAS`	*Aircraft true airspeed*
`IAS`	*Aircraft indicated airspeed*
`altitude`	*Aircraft altitude*
`latitude`	*Aircraft latitude*
`longitude`	*Aircraft longitude*
`beta_rate`	*Rate of change of angle of sideslip*
`beta`	*angle of sideslip*
`alpha_rate`	*Rate of change of angle of attack*
`alpha`	*angle of attack*
`pitch_rate`	*Rate of change of pitch*
`roll_rate`	*Rate of change of roll*
`yaw_rate`	*Rate of change of yaw*
`pitch_accn`	*Pitch acceleration*
`roll_accn`	*roll acceleration*
`yaw_accn`	*roll acceleration*
`pitch`	*Pitch angle*
`roll`	*Roll angle*
`yaw`	*Yaw angle*
`lift`	*Lift force*
`thrust`	*Thrust force*
`drag`	*Drag force*
`sideforce`	*Side force*
`engine_lever`	*Engine lever position*
`gamma`	*Flight path angle*
`vertical_speed`	*Rate of climb*
`spoiler`	*Spoiler (air brake)*
`parkbrake`	*Park brake*
`leftbrake`	*Left brake*
`rightbrake`	*Right Brake*

5.1 Examples of the plot command

```
plot pitch degs -10 20
plot TAS kts 100 250
plot altitude ft 2000 4000
plot elevator degs -30 30
```

6 Engage Command

The *engage* command is used to engage autopilot functions. The arguments are as follows:

Argument	Description
`ALT_HOLD`	*Altitude hold*
`HDG_HOLD`	*Heading hold*
`SPD_HOLD`	*Speed hold*
`VSPD_HOLD`	*Vertical speed hold*

6.1 Examples of the engage command

```
engage spd_hold
engage hdg_hold
```

7 Disengage Command

The *disengage* command is used to disengage autopilot functions. The arguments are as follows:

Argument	Description
`ALT_HOLD`	*Altitude hold*
`HDG_HOLD`	*Heading hold*
`SPD_HOLD`	*Speed hold*
`VSPD_HOLD`	*Vertical speed hold*

7.1 Examples of the disengage command

```
disengage spd_hold
disengage hdg_hold
```

8 Time Command

The *time* command is used to specific the length of time of the simulation. It also corresponds to the time axis for all recorded data. The argument is the time in seconds.

9 Autotrim Command

The *autotrim* command is used to set the aircraft inputs for non-accelerating trimmed flight. The initial conditions should be defined using the *set* commands. By default, the aircraft is not is a trimmed state at the start of the simulation. The *autotrim* command has no arguments.

10 Input Command

The *input* command is used to define elevator, aileron and rudder inputs applied to the simulator. The input can be a step, pulse, doublet, ramp or sinusoidal input. If a constant input is defined (e.g. set elevator -5 deg), the corresponding input should be defined as *off*. By default, no input function is applied. The arguments are as follows:

Argument	Description
`off`	*No inputs are applied*
`elevator`	*elevator input*
`aileron`	*aileron input*
`rudder`	*rudder input*

Each input waveform can be defined as follows:

Argument	Description
`step`	*A step input*
`pulse`	*A pulse input*
`doublet`	*A positive going pulse followed by a negative going pulse*
`ramp`	*A ramp*
`sine`	*A sinusoidal input*

Each waveform is defined by three parameters: the delay before it is applied in seconds, the period of the waveform in seconds and the amplitude of the waveform.

10.1 Examples of the input command

```
input elevator pulse 0.5 10 degs 5
input rudder sine 3 0.2 rad 0.3
```

The first *input* command is a pulse, starting after 0.5s, for a period of 10s with an amplitude of 5°. The second input command is a sinusoidal input after 3s, for a period of 0.2 and an amplitude of 0.3 radians.

11 Units

For the most part, the flight simulator operates in SI units. However, units can be specified in a *Simplot* program that are appropriate to an application. For example, in plotting flight data, the user may optionally specify the recording units, e.g.

```
plot TAS mph 20 30
plot IAS m/s 5 25
```

If the units are omitted, the default simulator units are used, in this case m/s. The units recognised by *Simplot* are summarised in the following table:

Variable	Units	Notes
`time`	secs, mins, hours	
`angle`	degs, rads	*degrees, radians*
`angular velocity`	deg/s, rad/s, RPM	*degrees per second, radians per second, revs per minute*
`angular acceleration`	deg/s/s, rad/s/s	
`length`	m, Ft, Km, nm	*metres, feet, kilometres, nautical miles*
`velocity`	m/s, fpm, Kts, Km/hr, mph	*meters per second, feet per minute, Knots, Kilometres per hour, miles per hour*
`acceleration`	m/s/s, ft/s/s	
`force`	N, lbf	*Newtons, pounds force*
`pressure`	mb, InHg	*millibars, inches of mercury*
`temperature`	deg_C	*degrees centigrade*
`flow rate`	kg/hr, lbs/hr	*Kilograms per hour, pounds per hour*
`mass`	Kg, lb	*Kilograms, pounds*

12 Running a *Simplot* Program

Create a text file to define the simulator initial conditions, the inputs and the variables to be plotted. Make sure a copy of the file **`simplot.exe`** is in your current directory. Using either a MinGw terminal (**`C:\MinGW\msys\1.0\msys.bat`**) or a Windows command prompt (Windows\run\cmd.exe), type:

```
./simplot filename
```

where 'filename' is the name of the text file to be executed, for example **`phugoid.txt`**. (Note that ./ is not required for the Windows command prompt).

Simplot will generate a script file **`filename.plt`** that can be loaded into GNUplot and a data file **`filename.dat`** containing the recorded data. Note that the .plt script file is a text file and, if appropriate, can be modified in order to plot data in a different format. Similarly, the .dat data file can be used with alternative plotting packages. The .plt script file enables GNUplot to generate a .png graphics file, which can be imported into standard packages such as Microsoft Word.

13 An Example of a *Simplot* Program

The following program positions the aircraft at 3000 ft, with an airspeed of 180 kt. The flaps are set at 20° (by default the gear is up). The simulation will record pitch in degrees in the range -10° to +20°, true airspeed in Kt in the range 100 Kt to 250 Kt, altitude in feet in the range 2000 ft to 4000 ft and elevator in degrees in the range -30° to +30°. The simulation runs for 120 seconds. Prior to starting, the simulation is auto-trimmed so that the engine lever input and elevator inputs will be set at trim values (by default, the aileron and rudder are zero). An elevator pulse of -10° is applied after 5s for 5s.

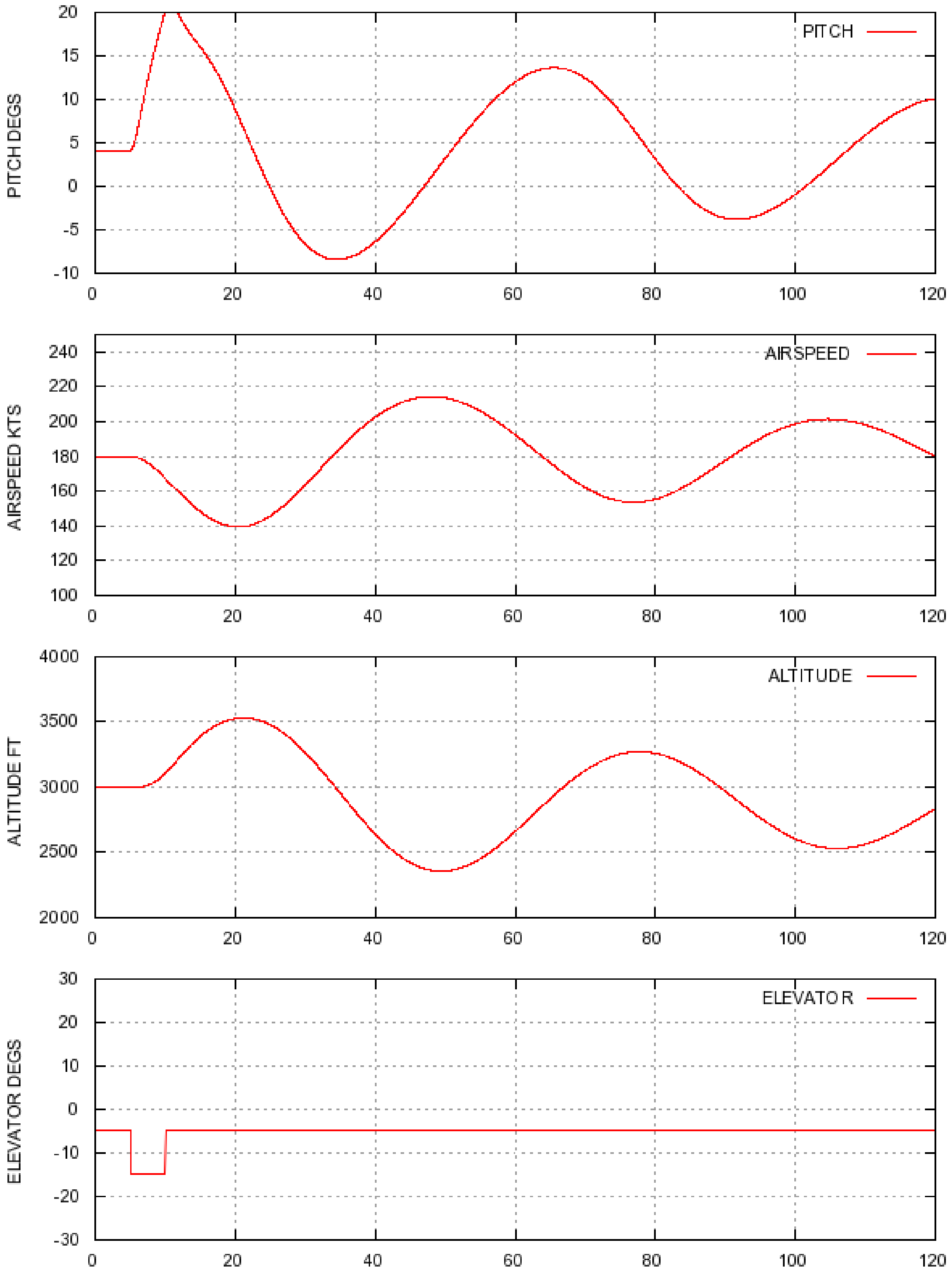
```
set altitude 3000 ft
set TAS 180 kts
set flaps 20
plot pitch degs -10 20
plot TAS kts 100 250
plot altitude ft 2000 4000
plot elevator degs -30 30
time 120 secs
autotrim
input elevator pulse 5 5 -10
```

The GNUplot script generated by *Simplot* is shown below:

```
set terminal png truecolor font arial 8 size 600,800
set output "ft4.png"
set size 1,1
set origin 0,0
set lmargin 10
set multiplot
set grid
set format y "%5g"
set size 1.0, 0.250000
set origin 0, 0.750000
set ylabel "PITCH DEGS"
set xr[0.0:120.000000]
set yr[-10.000000:20.000000]
plot 'ft4.dat' using 1:2 title 'PITCH' with lines
```

```
set origin 0, 0.500000
set ylabel "TAS KTS"
set xr[0.0:120.000000]
set yr[100.000000:250.000000]
plot 'ft4.dat' using 1:3 title 'TAS' with lines
set origin 0, 0.250000
set ylabel "ALTITUDE FT"
set xr[0.0:120.000000]
set yr[2000.000000:4000.000000]
plot 'ft4.dat' using 1:4 title 'ALTITUDE' with lines
set origin 0, 0.000000
set ylabel "ELEVATOR DEGS"
set xr[0.0:120.000000]
set yr[-30.000000:30.000000]
plot 'ft4.dat' using 1:5 title 'ELEVATOR' with lines
unset multiplot
reset
```

Finally, the output produced by GNUplot is the following .png file:



14 The *Simplot* Package

Simplot comprises the following source files written in C:

```
-rw-r--r-- 1 dave Administrators  9712 Aug 15 12:25 aero.c
-rw-r--r-- 1 dave Administrators  1300 Aug 14 18:40 aerolink.c
-rw-r--r-- 1 dave Administrators  3460 Jan 17  2017 clocks.c
-rw-r--r-- 1 dave Administrators 11477 Aug 13 10:35 engines.c
-rw-r--r-- 1 dave Administrators  3198 Aug 10 15:22 englink.c
-rw-r--r-- 1 dave Administrators 17398 Aug  8 08:34 fcs.c
-rw-r--r-- 1 dave Administrators 12182 Apr  3 17:37 gear.c
-rw-r--r-- 1 dave Administrators  7941 Aug 10 14:52 iolib.c
-rw-r--r-- 1 dave Administrators  2631 Mar 30  2016 maths.c
-rw-r--r-- 1 dave Administrators 18740 Aug 15 08:53 model.c
-rw-r--r-- 1 dave Administrators 29283 Aug 14 21:43 simplot.c
-rw-r--r-- 1 dave Administrators 17868 Aug 14 19:27 simulate.c
-rw-r--r-- 1 dave Administrators  4117 Aug 14 09:12 stab.c
-rw-r--r-- 1 dave Administrators  2053 Aug 13 10:21 systems.c
-rw-r--r-- 1 dave Administrators 15182 May  9 14:37 weather.c
```

The files aero.c, engines.c, fcs.c, gear.c and model.c are identical to the real-time modules used in the flight simulator.

The files clocks.c, maths.c and weather.c are identical to the real-time library modules used in the flight simulator

The files aerolink.c, englink.c, iolib.c and systems.c are similar to the real-time flight simulator modules but are simplified to contain only the code necessary to run *Simplot*.

The files simplot.c, simulate.c and stab.c are specific to *Simplot* and perform the following functions:

Module	Function
simplot.c	The main program – reads the *Simplot* file, parsing the code, initialising the variables defined in the *Simplot* file and generating the output files (.dat and .plt) for GNUplot.
simulate.c	Responsible for auto-trimming and setting up the environment for the main simulator modules. This module manages the scheduling of the simulator modules aero.c, engines.c, fcs.c, gear.c and model.c and provides overall initialisation and management of the simulation.
stab.c	Computes and displays the non-dimensional aerodynamic derivatives prior to the simulation.