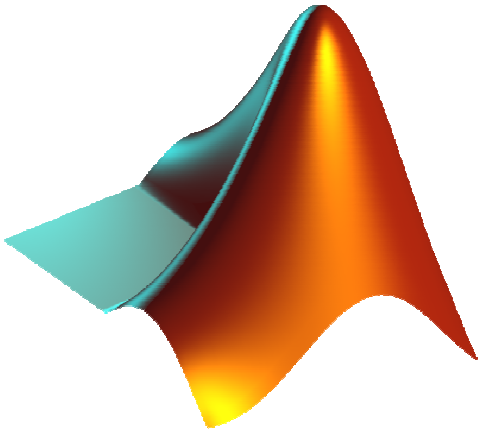
Rapid Prototyping a Two Channel Autopilot for a Generic Aircraft

YOGANANDA JEPPU

Head R&D Systems

Moog India Technology Center



The Team

Atit Mishra
Basavaraj M
Chethan CU
Chinmayi J
Manjunatha L Rao
Surya Karthik
Vanishree
Yogananda

at flights 4 fantasy, forum mall ...

Agenda

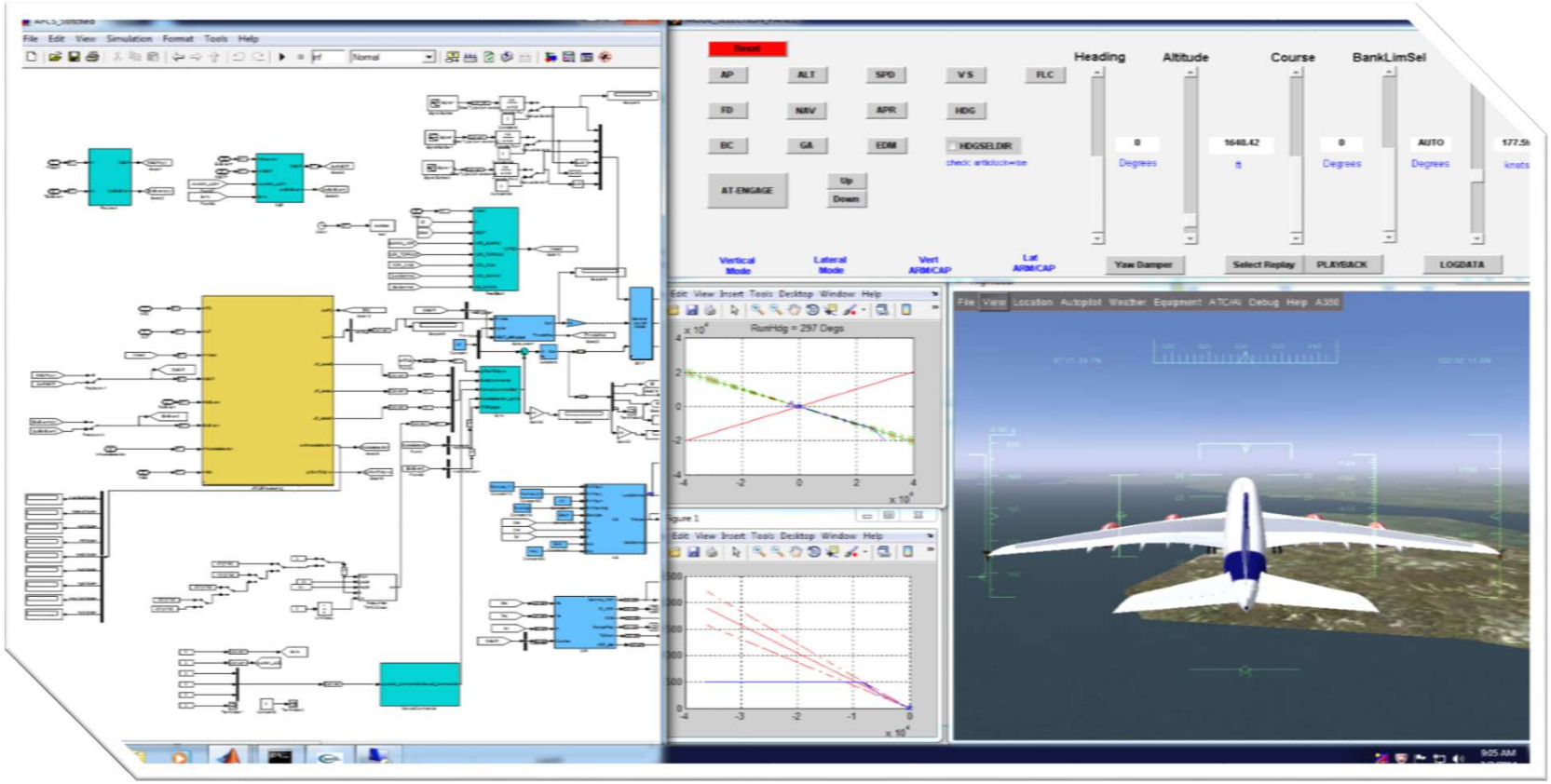
- What are autopilots
- Design Process
- Aircraft Model, Trims and Envelope
- Control Design
- Mode Transition Logic
- Code and Test
- 2 Channel Operation
- Demo

I am not here to praise Matlab but its users who have done so much with it …

Take away

- How a small team has been able to design a full fledged two channel autopilot with all the modes in a time frame of 8 months
- How Matlab and Simulink and the user community has helped us achieve this task
- Some things we have given back to the community

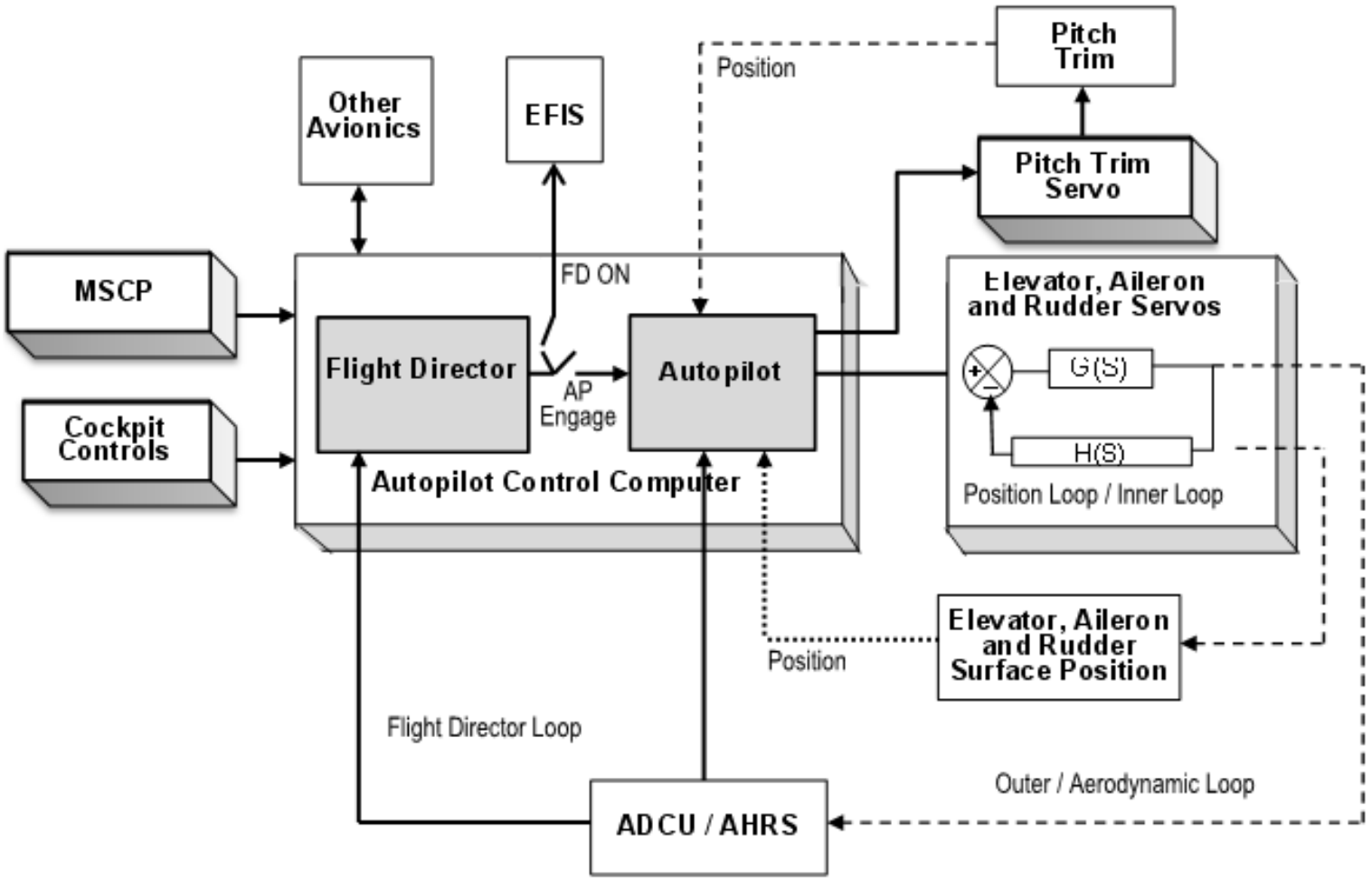
Autopilots


Heading
Altitude
Course
BankLimSel
AP
ALT
SPD
VS
FLC
FD
NAV
APR
HDG
BC
GA
EDM
HDGSELDIR
AT-ENGAGE
Up
Down
Yaw Damper
Select Replay
PLAYBACK
LOGDATA

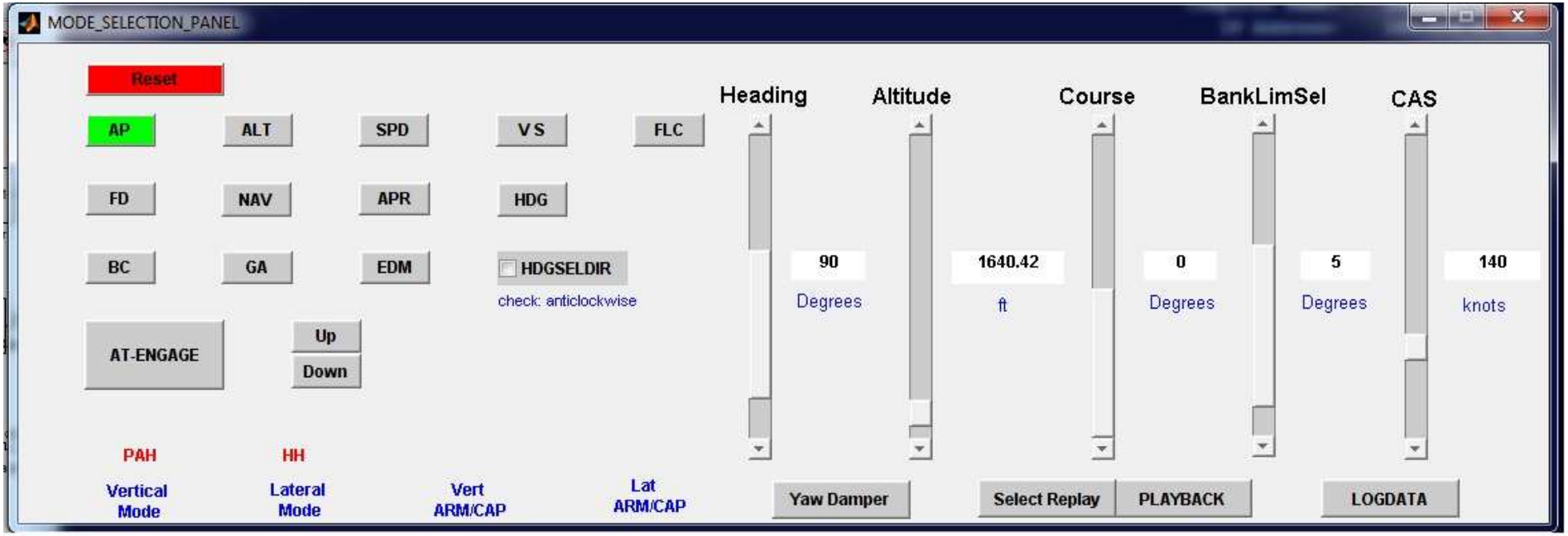
Autopilot

- Autopilot system is a mechanical, electrical or hydraulic system used to guide an airplane with minimal or no assistance from the pilot. It also reduces the fuel consumption and increases flight safety.
- Flight Director is a navigational aid that is overlaid on the attitude indicator that shows the pilot of an aircraft the attitude required to follow a certain trajectory.

Autopilot overview



MSCP



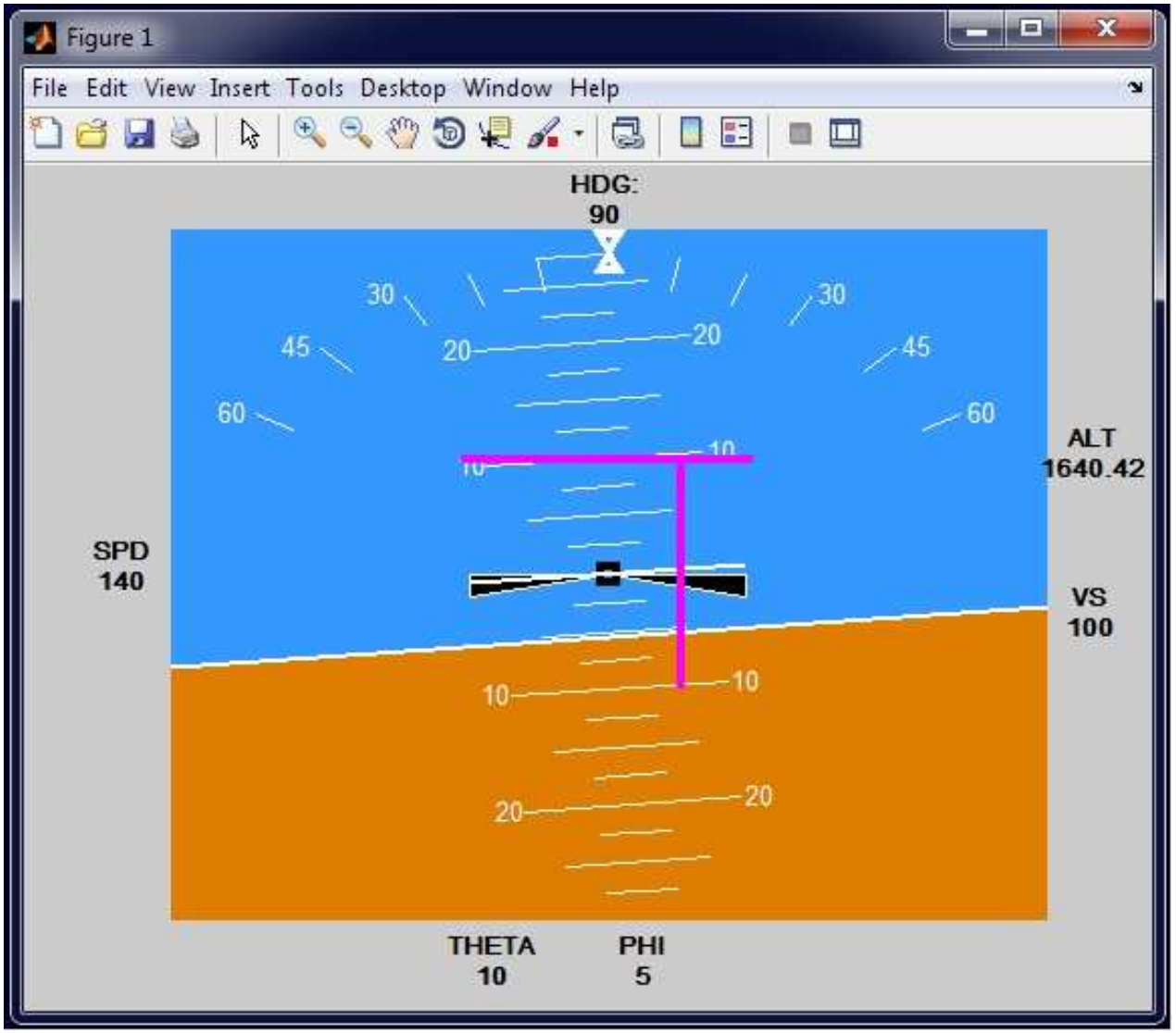
Mode Select Control Panel

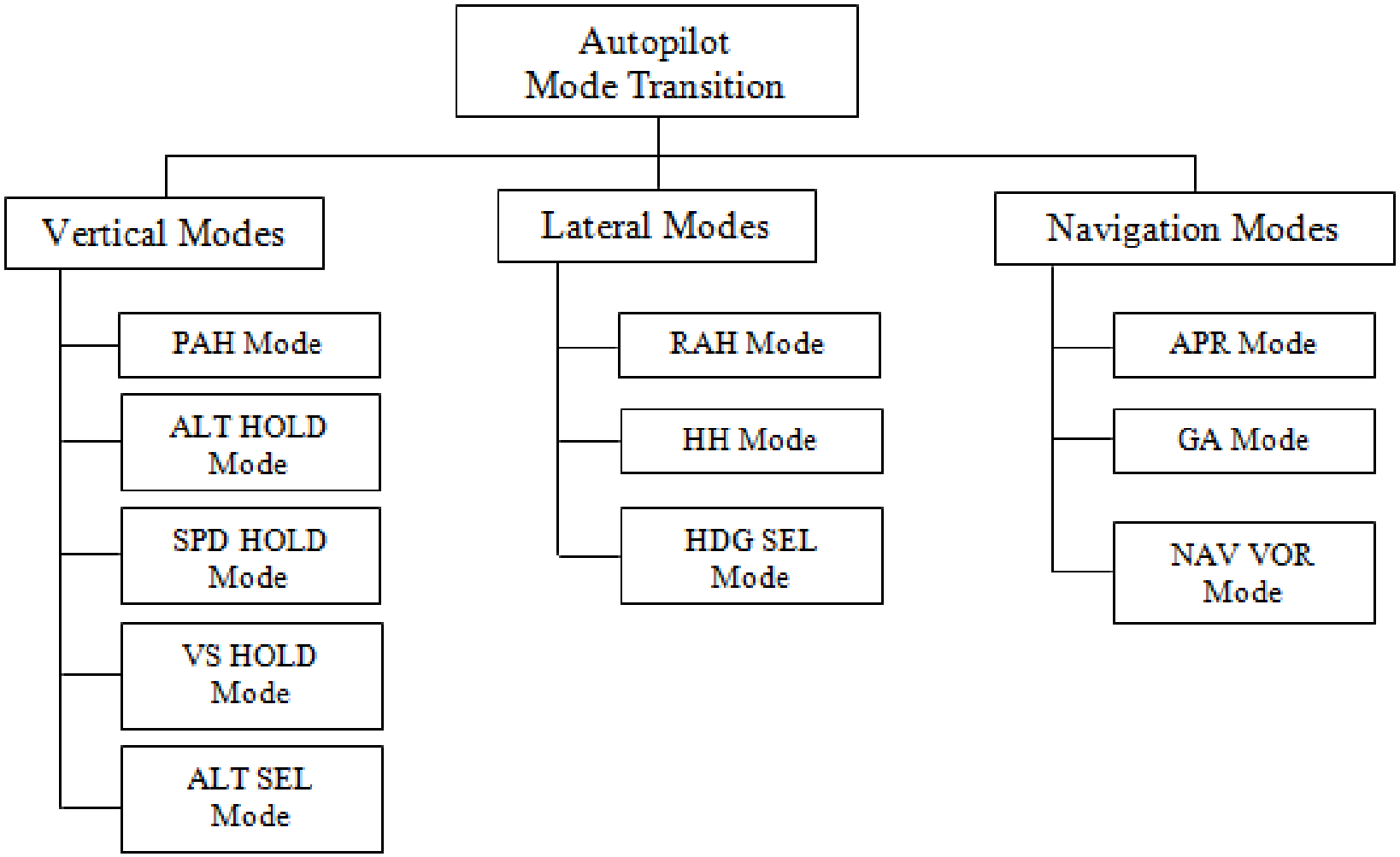
This is developed using Matlab UI and has a real time interface with the Simulink simulation. It has a record and playback capability for system tests

PFD

Primary Flight Display

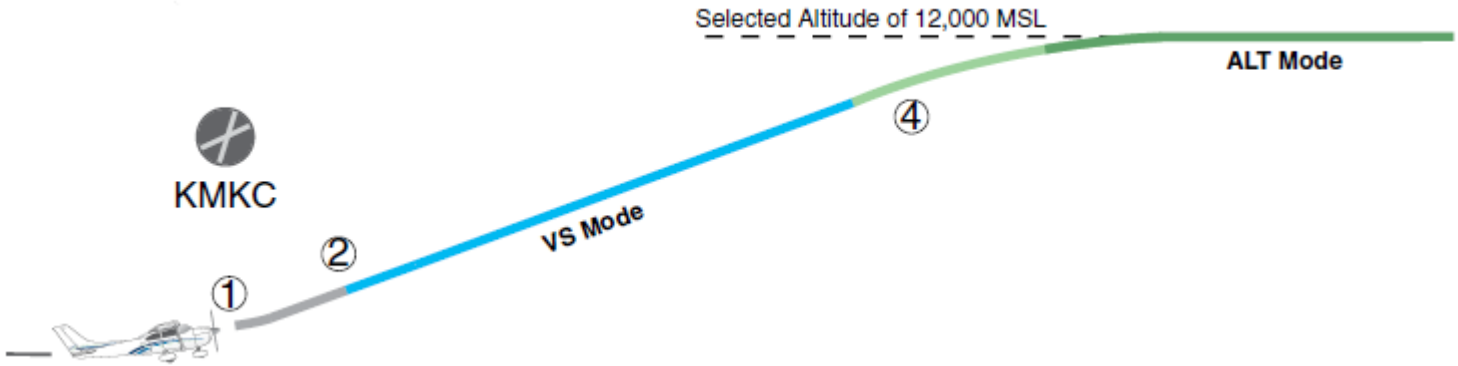
This is developed using the simple plot program and runs in real time displaying the Simulink data




Autopilot
Mode Transition
Vertical Modes
PAH Mode
ALT HOLD
Mode
SPD HOLD
Mode
VS HOLD
Mode
ALT SEL
Mode
Lateral Modes
RAH Mode
HH Mode
HDG SEL
Mode
Navigation Modes
APR Mode
GA Mode
NAV VOR
Mode

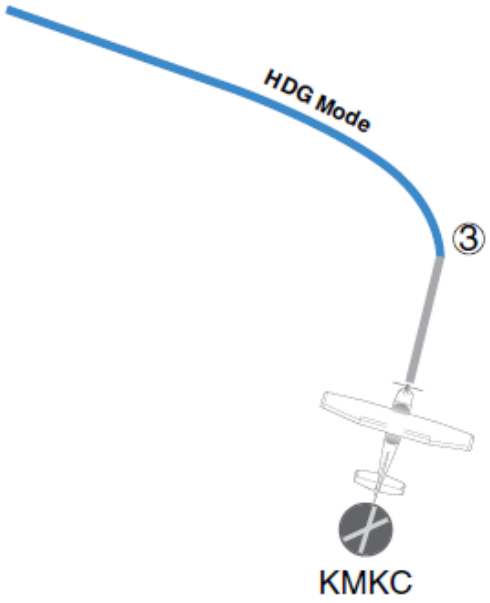
Vertical Modes

- PAH mode – This is the basic autopilot mode in vertical axis. This mode holds the current pitch angle.
- Altitude Hold mode – This mode holds the aircraft at the current altitude reference.
- Speed Hold mode – This mode maintains the present airspeed.
- Vertical Speed mode – This mode is used to automatically maintain the aircraft at a selected vertical speed (climb rate) reference.
- Altitude Select mode – This mode captures the Selected Altitude. The 3 phases are, Arming, Capture and Hold.



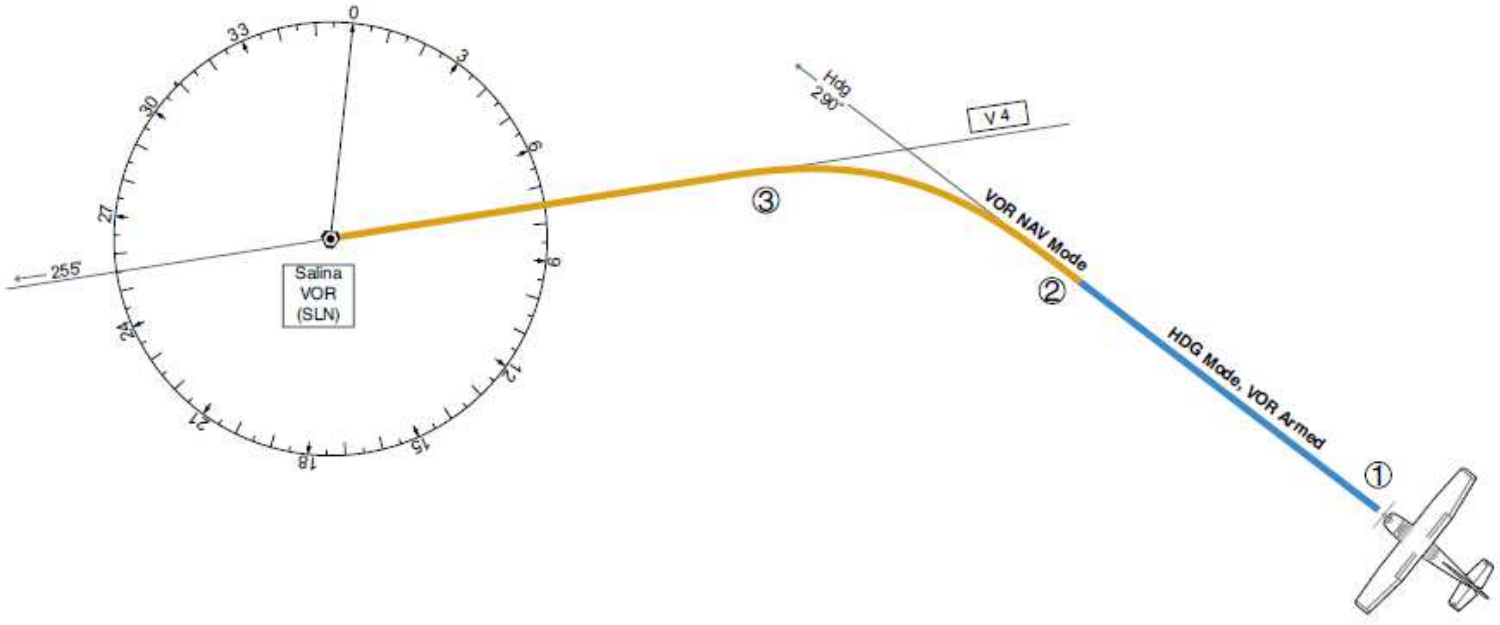
Lateral Modes

- RAH mode – This is the basic autopilot mode in lateral axis. This mode holds the current roll angle of the aircraft.
- Heading Hold mode – This mode is used to hold the heading of the aircraft.
- Heading Select mode – This mode is used to turn towards the Selected Heading.



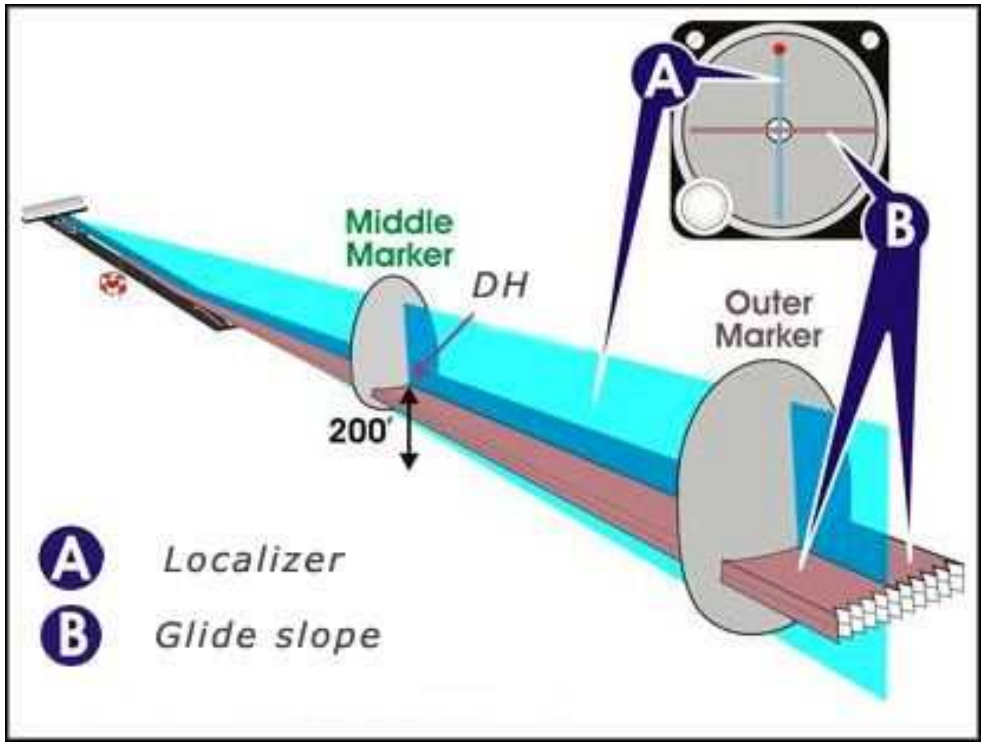
Navigational Modes

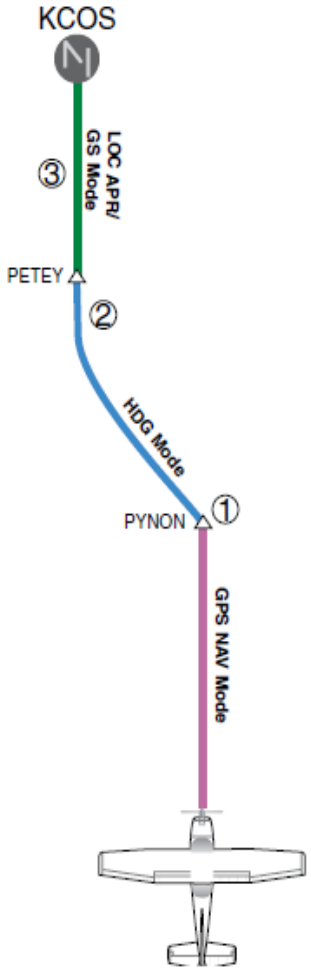
- Go-Around mode – The Go-Around mode is used transition from an approach mode to a climb mode when a missed approach occurs.
- Navigation VOR - This is a type of short-range radio navigation system for aircraft which defines the radials in space for tracking.



Approach Modes

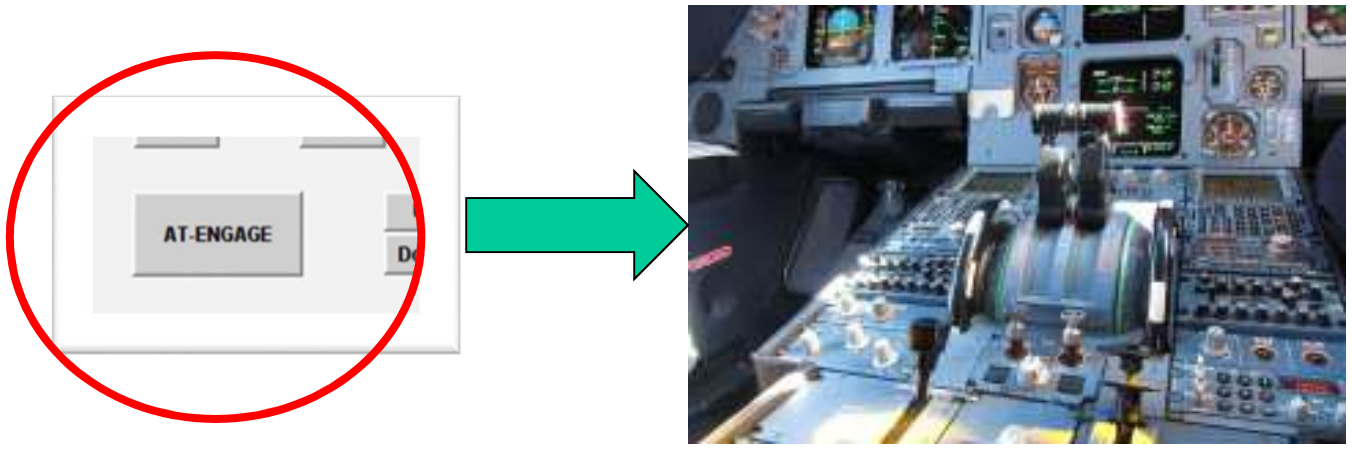
- Approach mode – This mode is used to approach the runway. Glide-slope and Localizer modes assist in vertical and lateral guidance.



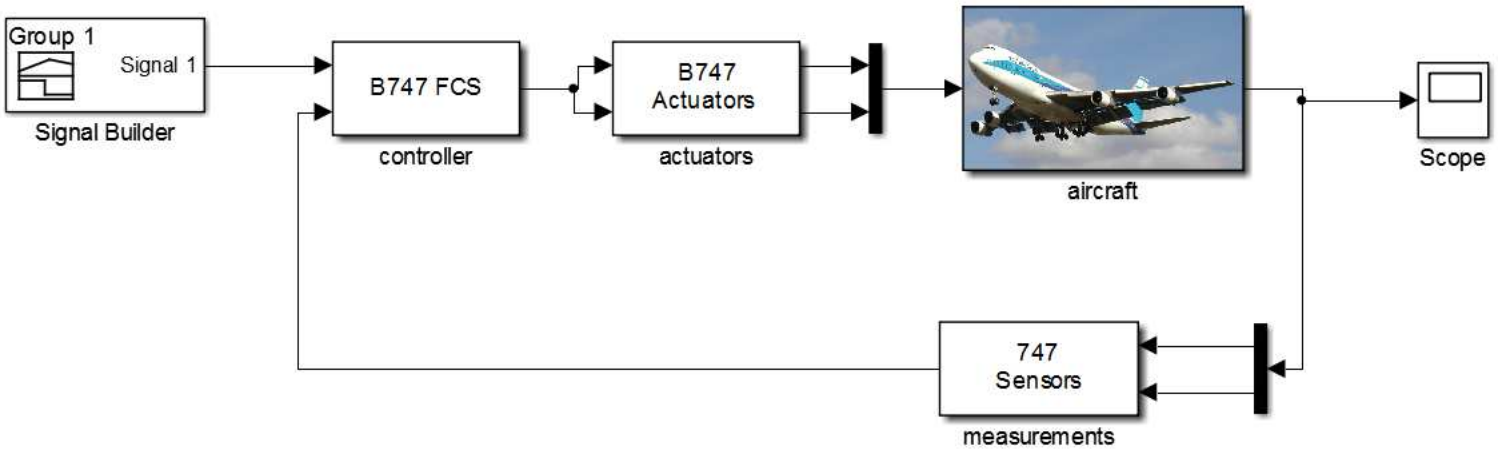


Autothrottle and Yaw Damper

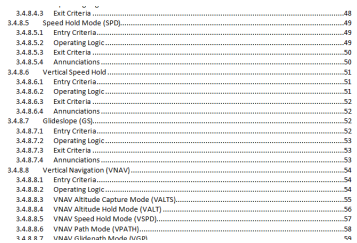
- An autothrottle mode is available to ensure that the engine maintains the speed or vertical speed reference.
- The yaw damper mode is also available to damp out the Dutch roll oscillation in yaw and maintain a low value of sideslip angle.
- In the yaw damper mode the commands are given to the rudder.
- These modes can be engaged with the main autopilot being off



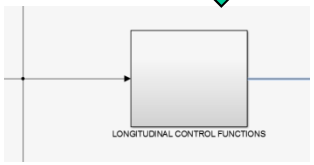
Design Process



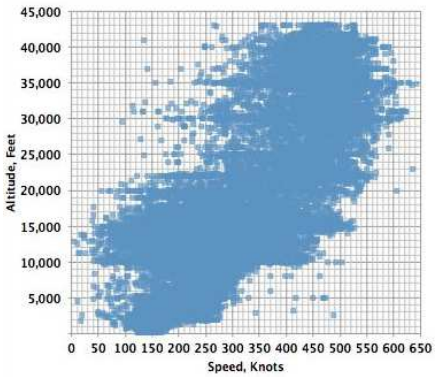
Requirements



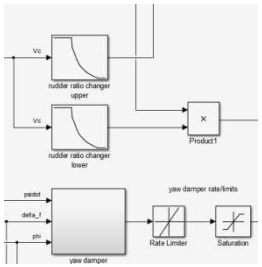
Linear Models

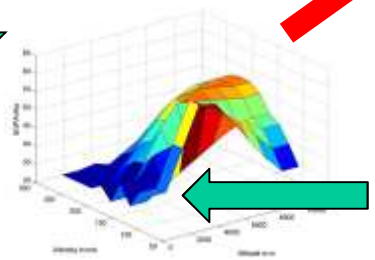


Linear Control

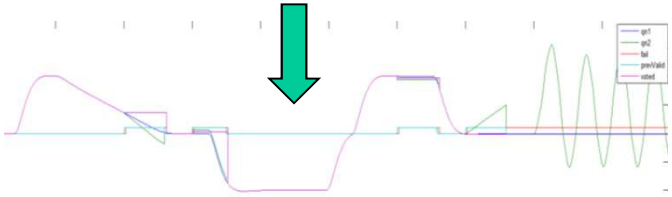


Flight Envelope





Non Linear Shaping

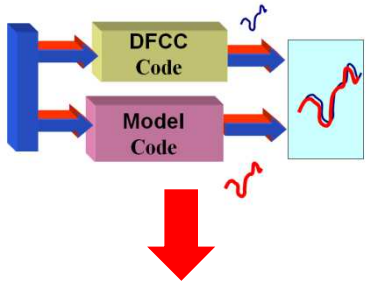


Simulations

Code

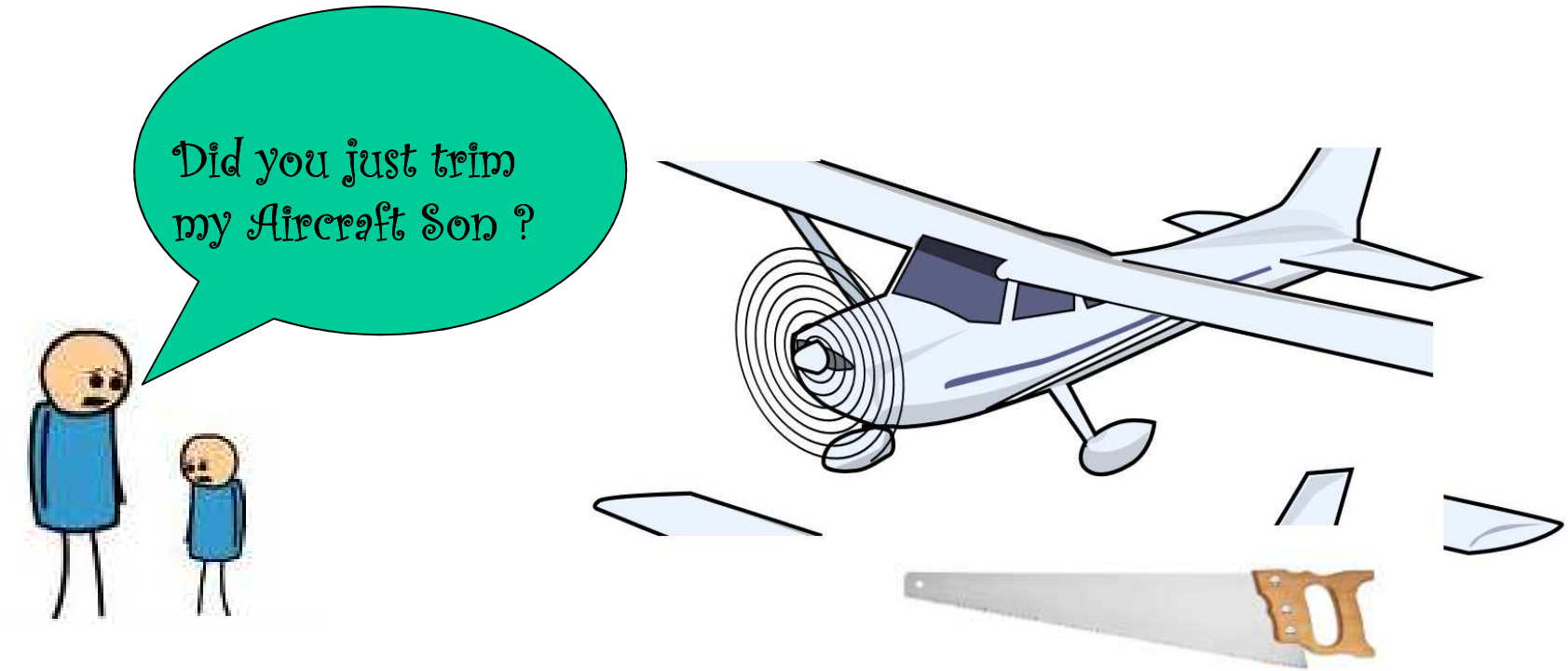
```
#include "APUTLRateLimit.h"
#include "APUTLSysDefs.h"

//-----------------------------------------------------------------------
// Local Defines
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
// Global Variables
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
// Global Functions
//-----------------------------------------------------------------------
// ***********************************************************************
//
// FUNCTION NAME: RateLimit
//
// Notes:
//
// ***********************************************************************
TFloat32 RateLimit(TRateLimit *thisPtr, TFloat32 input,
                   TFloat32 posSr, TFloat32 negSr,
                   TBool init)
{
  TFlo    owerLim = 0.0f;
  TF      upperLim = 0.0f;

  nitialization
  f (init == TRUE)
```

Test



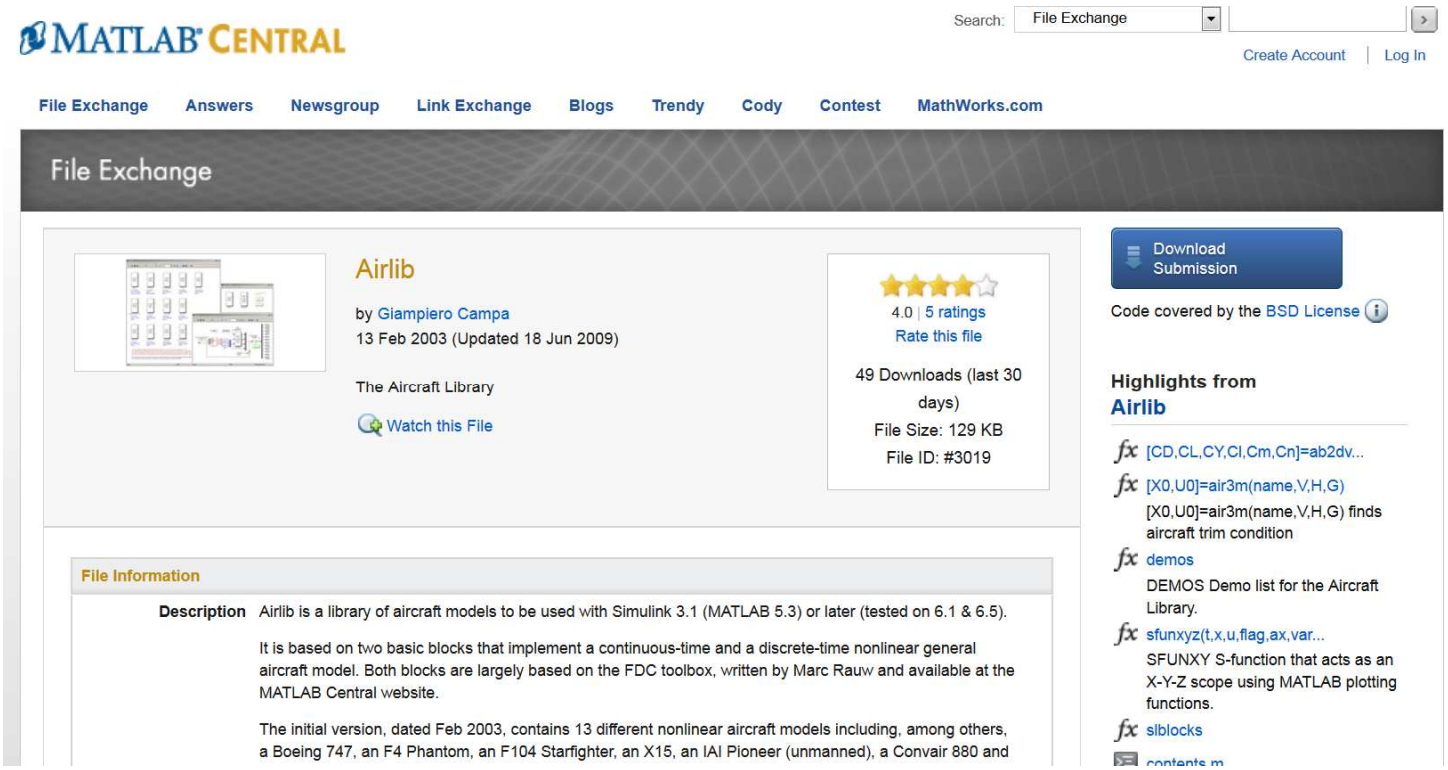
Deploy and Test

Aircraft Model, Trim, Envelope



Aircraft Model

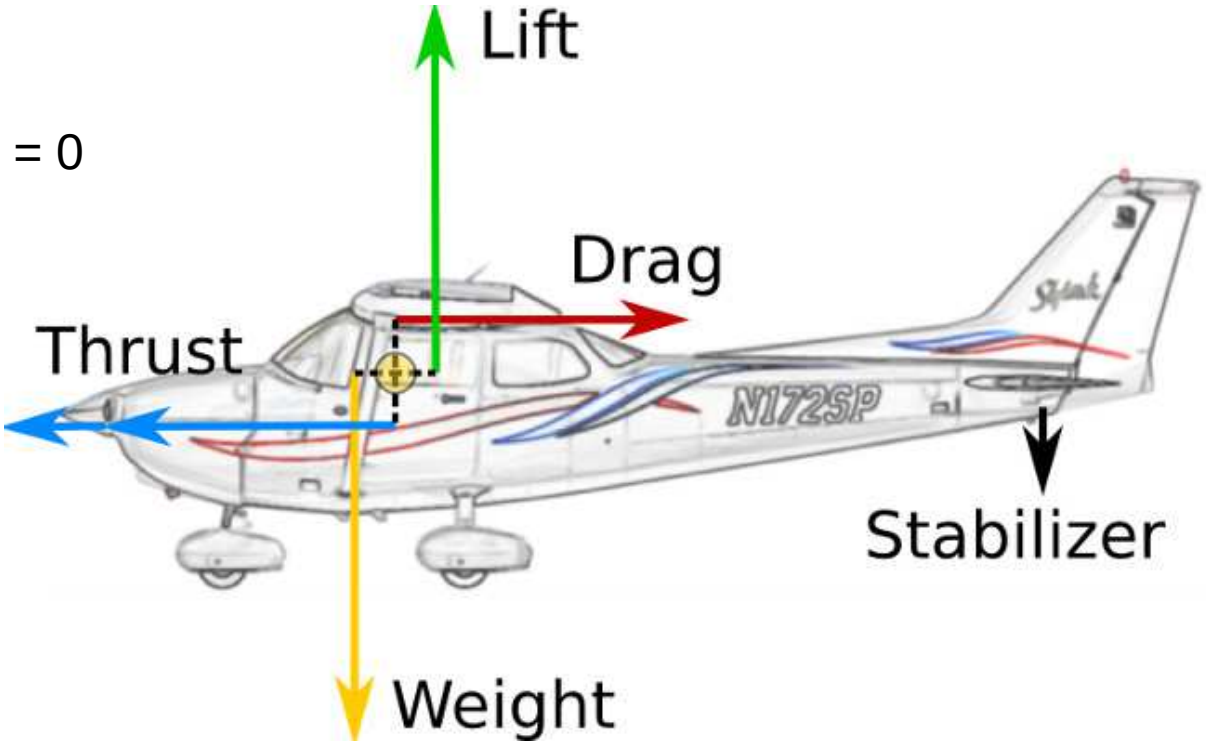
- Airlib – by Giampiero Campa



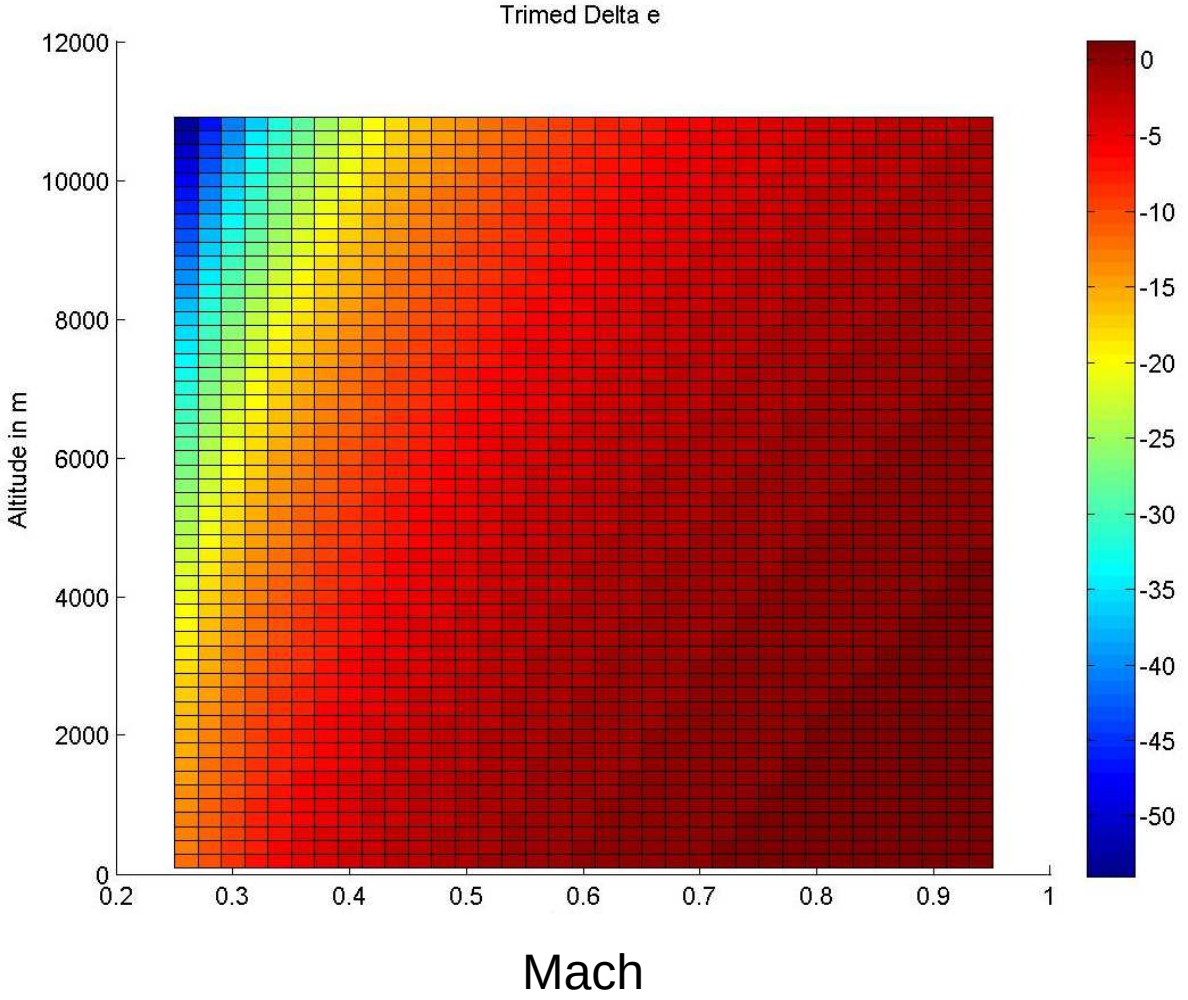
Aircraft Trim

Equilibrium point is computed optimally for a specific velocity(Mach) and Altitude

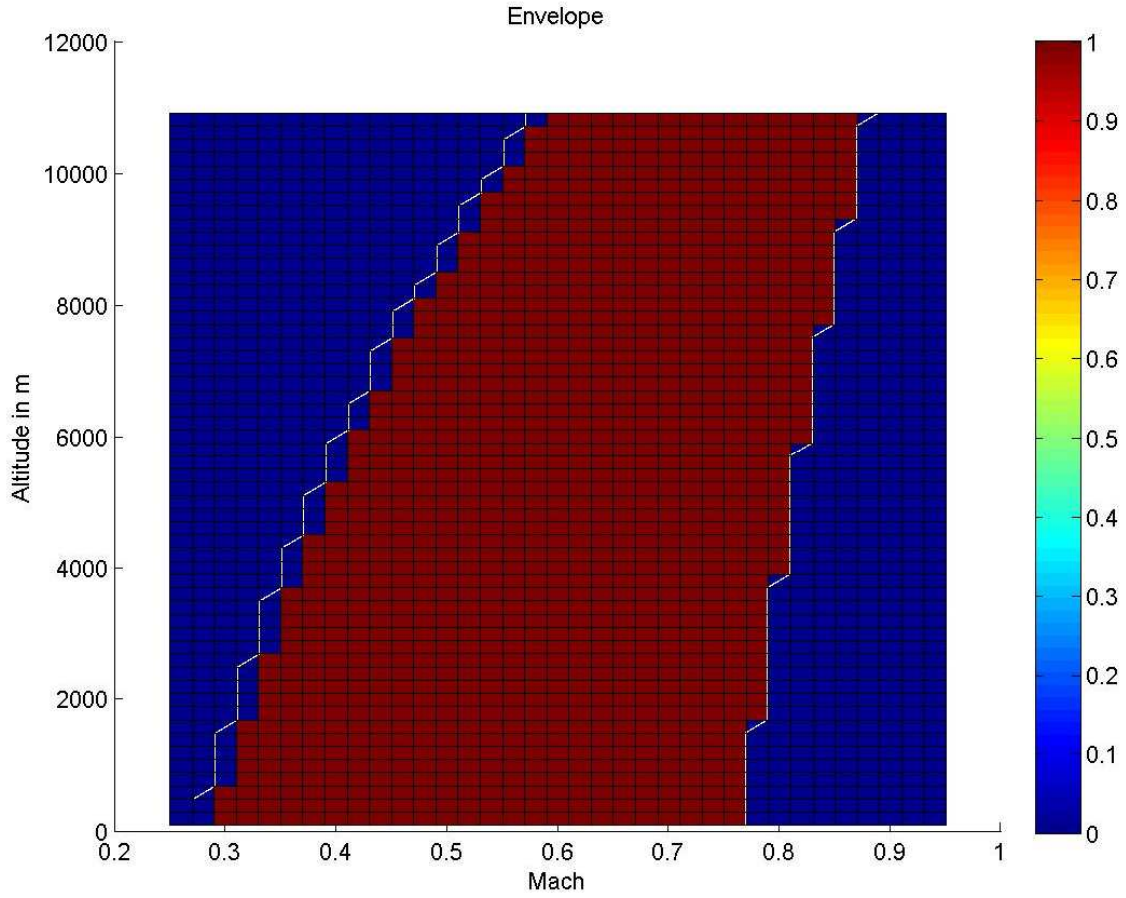
Lift = Weight
Thrust = Drag
All rates and Accelerations = 0



Flight Envelope

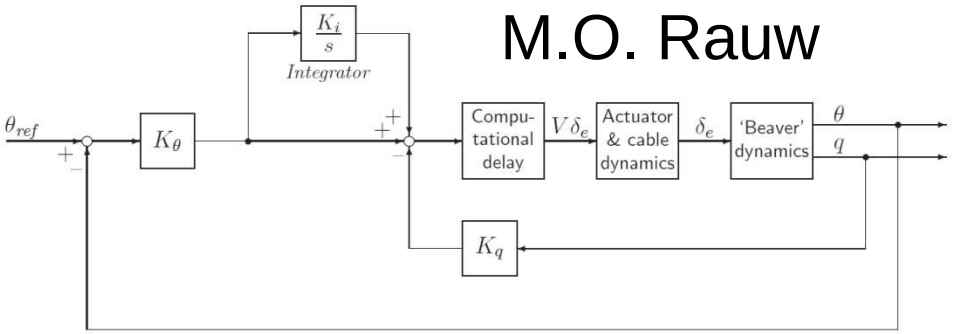


Flight Envelope



The FDC Toolbox

M.O. Rauw



The Flight Dynamics & Control Toolbox

Home
News
Info
- The author
- History
- Objectives
- Dials & Gauges
- FAQ
- Known problems
- Thanks!

Downloads
- Licenses
- *FDC toolbox*
- FDC manual
- DUBSI blockset

FDC toolbox downloads

Currently there are two versions of the Flight Dynamics and Control toolbox available for dowload: FDC 1.3.3, which was designed for Matlab 5.1 / Simulink 2.1 ("Release 9") or newer, and FDC 1.2.1, which is compatible with Matlab 4.2 / Simulink 1.3. The toolbox is open source software; see the License Agreement for details.

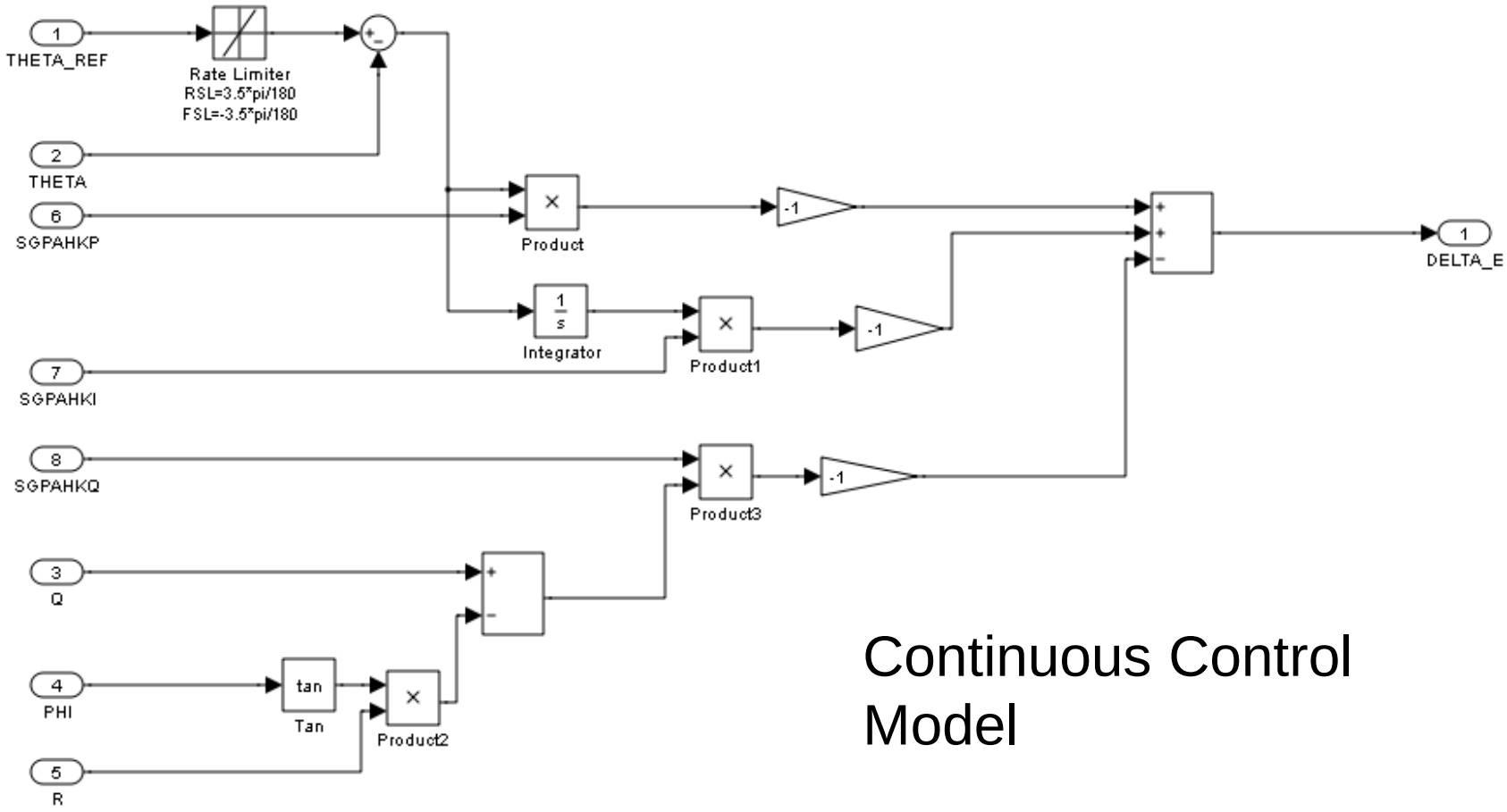
In addition to these current FDC versions, there is also a 'beta release' of the upcoming FDC 1.4, which can be obtained from the Dutchroll CVS tree at SourceForge, or as a zipped archive. FDC 1.4 beta is compatible with Matlab R11 or newer; it is subject to the terms of the Open Software License. Although the development of this version is not entirely completed and its user-manual is still under construction, it is deemed superiour to FDC 1.3.

Before using the software, it is recommended to take a look at the frequently asked questions and known problems pages too.

Preview of the next version: FDC 1.4 beta 3

FDC 1.4 beta 3 is the second 'official' beta of the next FDC release. Compared to the previous two beta's, several helptexts

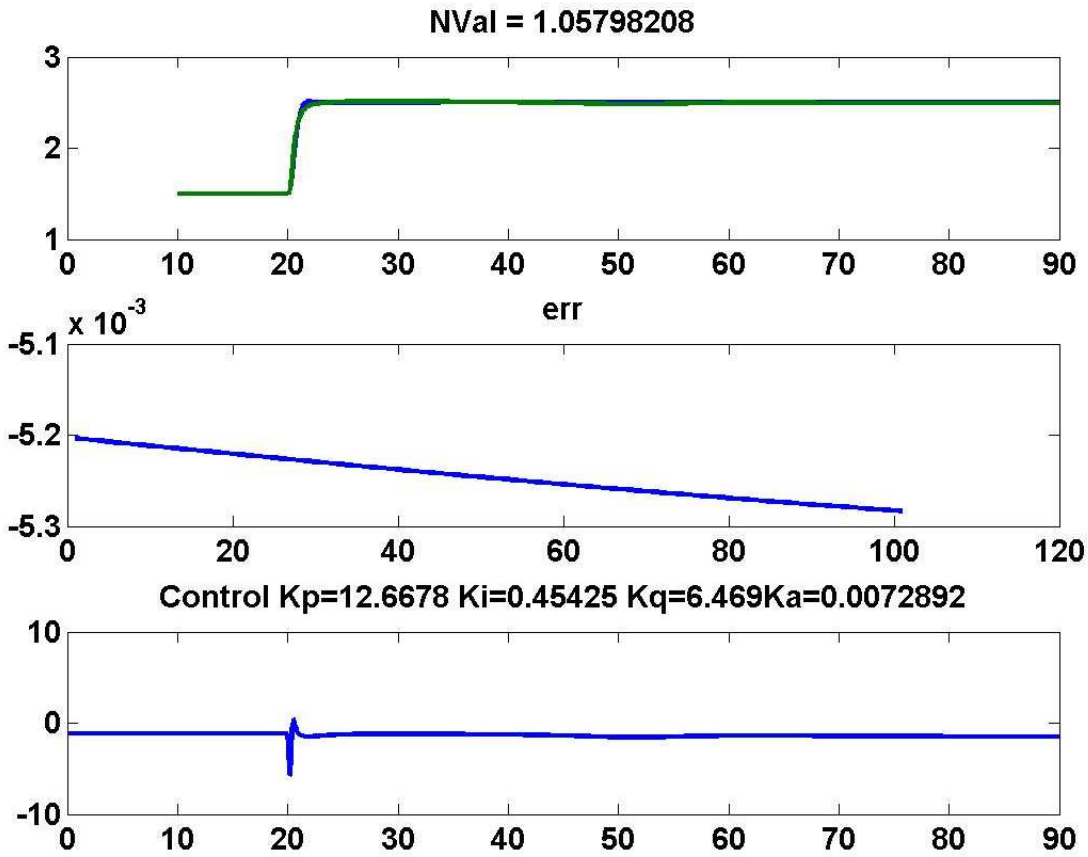
Pitch Attitude Hold



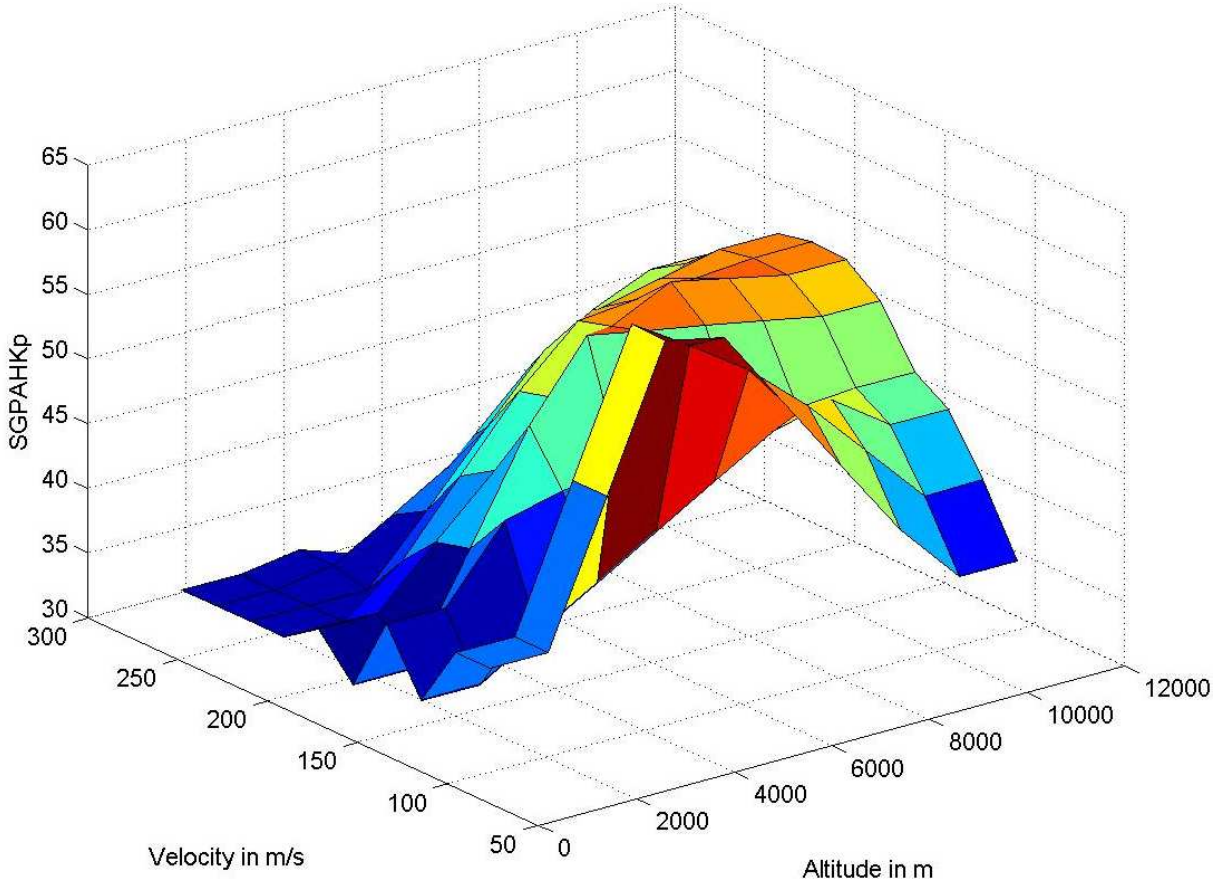
Continuous Control Model

Rapid Prototyping

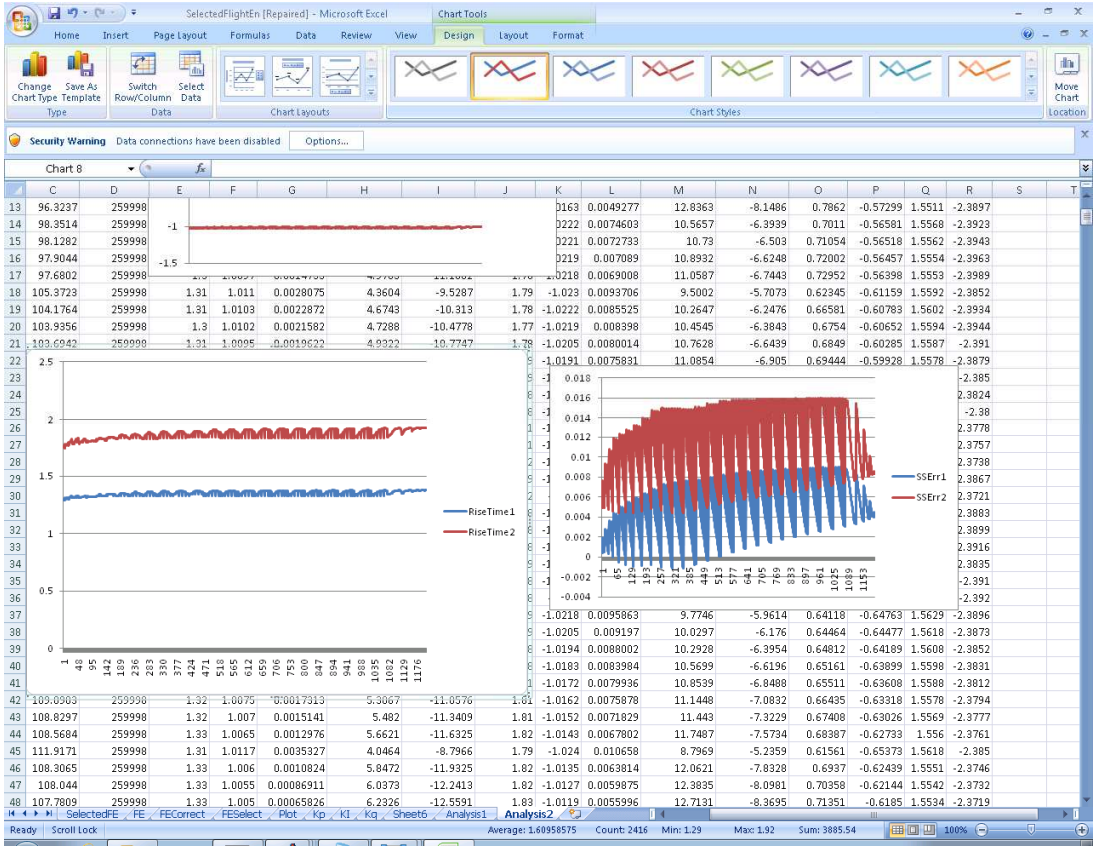
fminsearch()



Scheduled Gains

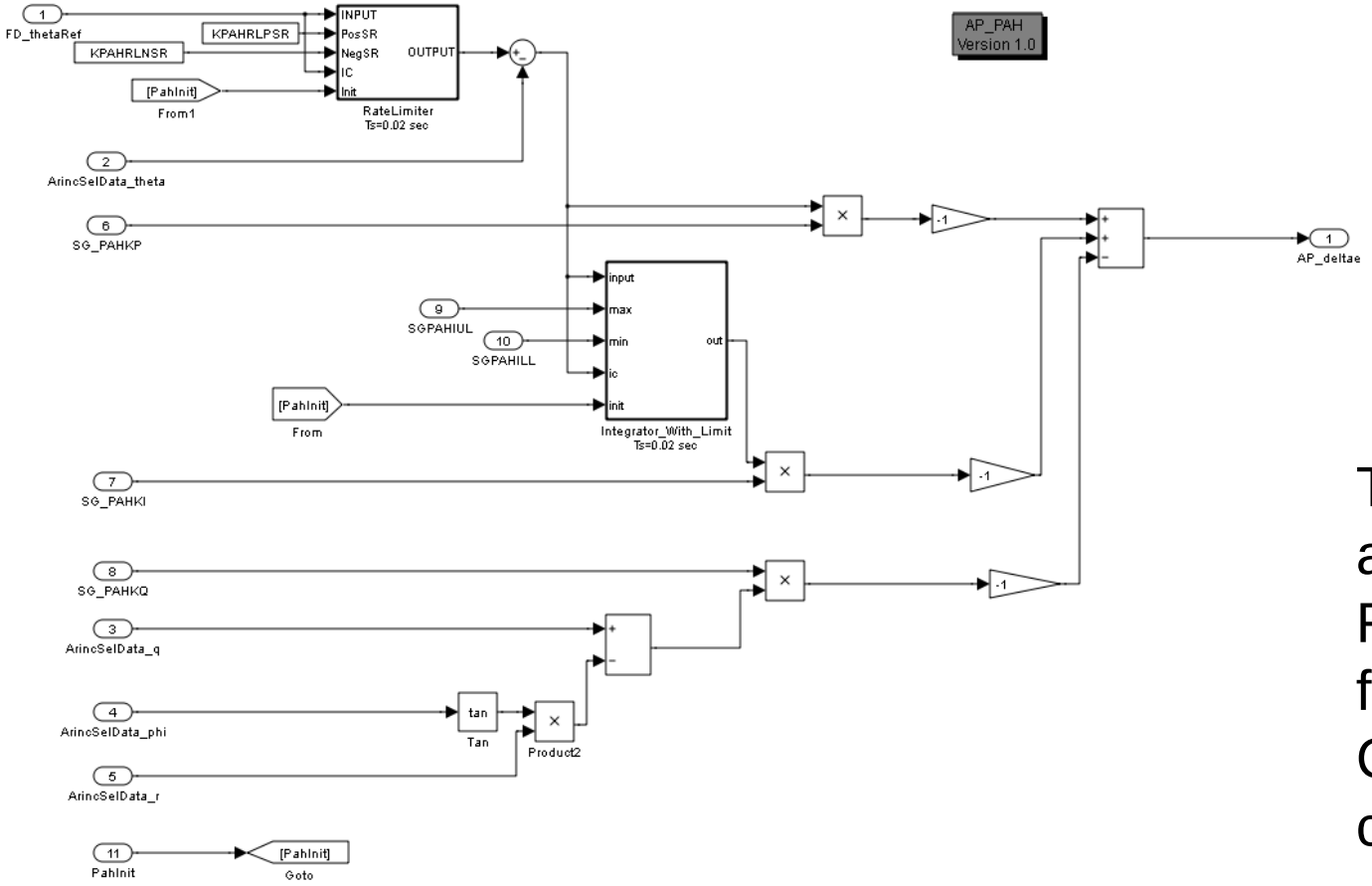


Validation of Schedule Gains



Scheduled gains validated against 1200 models in the flight envelope

Digital PAH



These models are the SDD Requirements from which the C code is hand coded.

Mode Transition Logic

- Mode Transition table indicates the possible transitions from any operational mode to another mode due to an event.
- Event can be a software trigger or button press from the pilot.
- Condition table indicates the conditions that must be TRUE for the respective transition to occur.
- Conditions are tabulated in a separate table.

States	Sl. No.	Modes	01	02	03	04	05	06	07	08	09	1
			AP	FD	SPD	VS	ALT	ALTS	ALTCAP	ALTCPDN	GA	A
Vertical	01	DIS(Vertical)	02	02	00	00	00	00	00	00	00	0
	02	PAH	010202	010202	03	04	05	00	07	00	0601	0
	03	SPD HOLD	010303	010303	02	04	05	00	07	00	0601	0
	04	VS	010404	010404	03	02	05	00	07	00	0601	0
	05	ALT HOLD	010505	010505	03	04	02	00	00	00	0601	0
	06	GA(Vertical)	02	01	00	00	00	00	00	00	00	0
	07	ALTS CAP	010707	010707	00	00	05	00	00	05	0601	0
	08	APR GS(CAP & ACT)	010808	010808	00	00	00	00	00	00	0601	0
	09	EDM(Vertical)	02	01	00	00	00	00	00	00	00	0
Lateral	01	DIS(Lateral)	0203	0203	00	00	00	00	00	00	00	0
	02	RAH	010202	010202	00	00	00	00	00	00	0501	0
	03	HH	010303	010303	00	00	00	00	00	00	0501	0
	04	HDG SEL	010404	010404	00	00	00	00	00	00	0501	0
	05	GA(Lateral)	0203	01	00	00	00	00	00	00	00	0
	06	APR LOC(CAP & ACT)	010606	010606	00	00	00	00	00	00	0501	0
	07	NAV VOR	010707	010707	00	00	00	00	00	00	0501	0
	08	EDM(Lateral)	0203	01	00	00	00	00	00	00	00	0
AP	01	AP ON	02	00	00	00	00	00	00	00	02	0
	02	AP OFF	0103	00	00	00	00	00	00	00	00	0
	03	AP SYNC	02	00	00	00	00	00	00	00	00	0

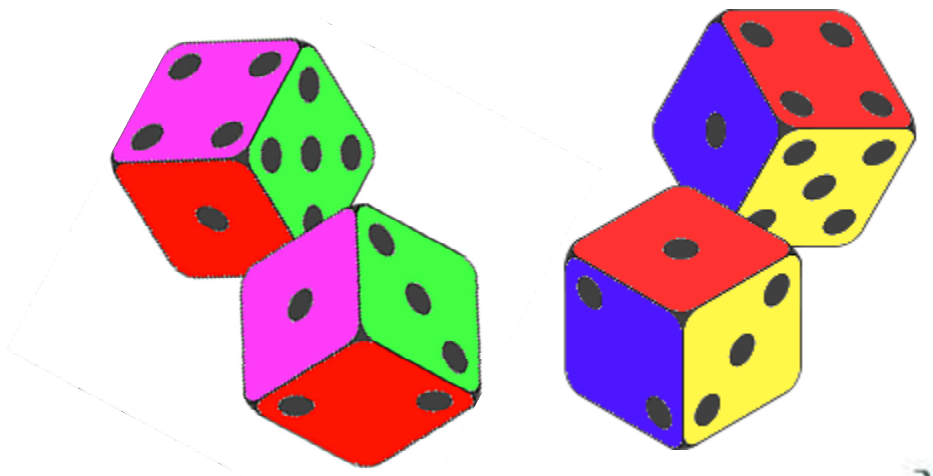
Validated with Assertions

- Starting from a safe state the autopilot modes have to transition to a safe state.
- The correctness of the safe state is verified by reviews
- Violation of any assertion for all combination of states, triggers and conditions requires a redesign of that transition
- This amounts to 36,391,680 executions during the validation process. 2 days of weekend runs

```
% ------------ ASSERTION 4 ------------
% When ALT SEL is captured, then
% vertical mode need to be ALT CAP.
if(ALTS(i5) ==3)
    if(Vertical(i1)~=7)
        assert = 1;
    end
end  % END ASSERTION 4

% ------------ ASSERTION 5 ------------
% During ALT HOLD mode ALT
% SEL is turned OFF.
if(Vertical(i1) == 5)
    if(ALTS(i5) ~=1)
        assert = 1;
    end
end % END ASSERTION 5

% ------------ ASSERTION 6 ------------
% Deals with ALT SEL Capture scenario.
if(Vertical(i1) == 7)
    if(ALTS(i5) ~=3)
        assert = 1;
    end
end % END ASSERTION 6
```

Code and Random Test

```
पूर्णांक मुख्य()
{
        जानवर ज,*ज1;
        शेर श;
        चीता च;

        ज.लिखो();
        श.लिखो();
        ज1->लिखो();
        ज1=&च;
        ज1->लिखो();

        वापस();
}
```

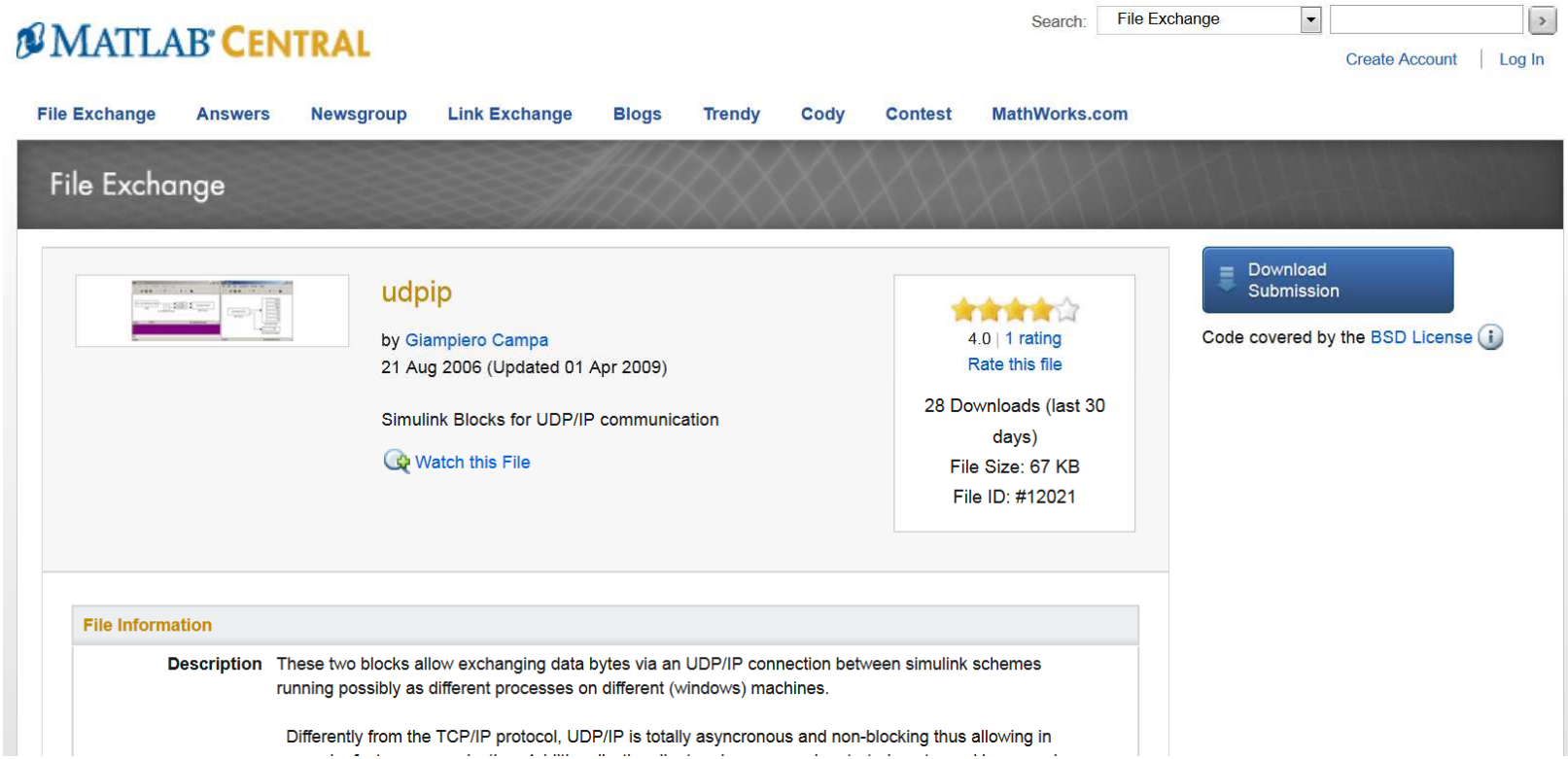


C Code and Test

- Manual coding with the utility functions for the various blocks
- Build each block using the utility functions
- Test each block using the mex function
- Random sinusoidal waveforms for float signals and random toggles for Boolean signals used to test the C code with the Simulink models
- Overnight runs cleared the code

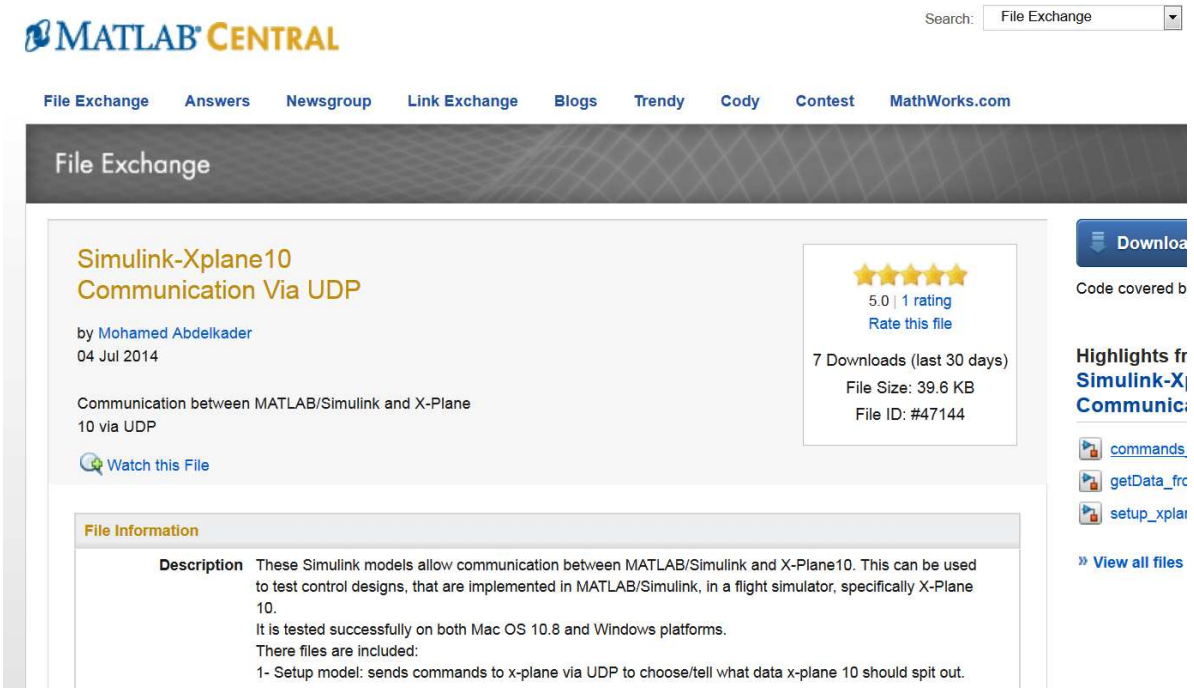
UDP

- UDPIP – by Giampiero Campa



UDP – X-planes

- Simulink – X-plane 10 UDP – by Mohamed Abdelkader



Model in Loop Tests

- We have tried something new with the C code compiled as an exe and communicating with the Simulink model using UDP
- In a two channel operation we have two exe files communicating with each other and the expected result generated from the two channel stitched model

Error plots

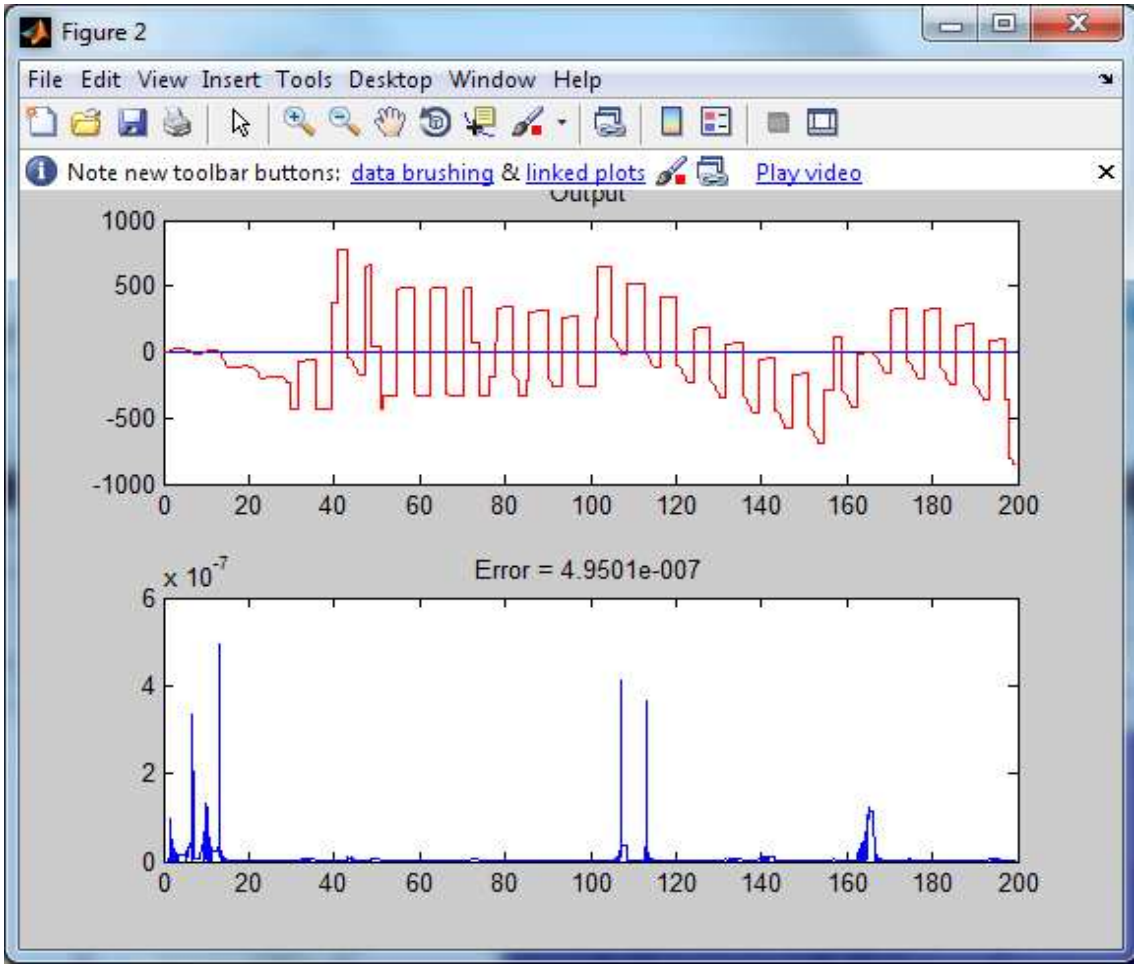
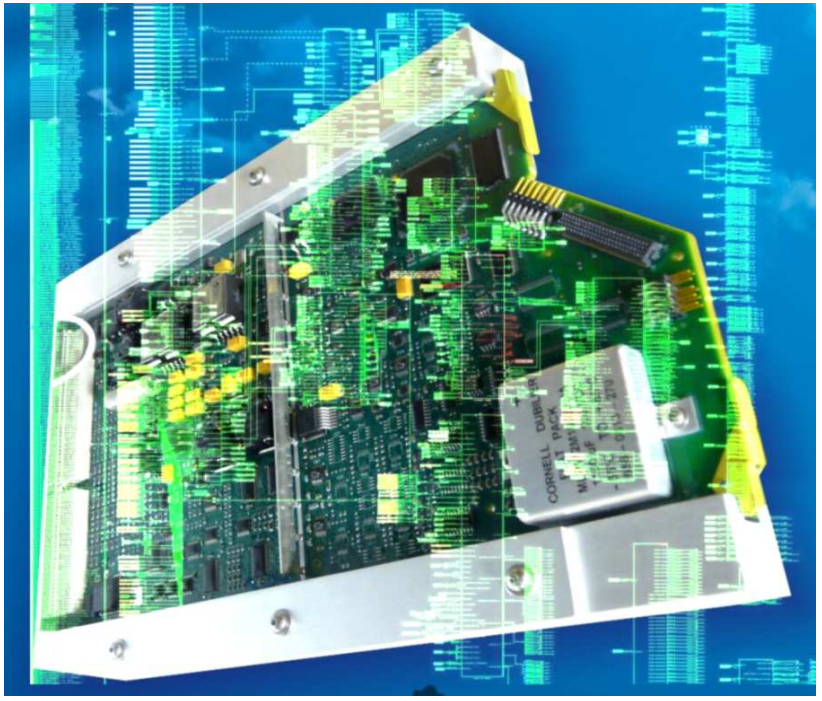


Figure 2
File Edit View Insert Tools Desktop Window Help
Note new toolbar buttons: data brushing & linked plots
Play video
Output
1000
500
0
-500
-1000
0 20 40 60 80 100 120 140 160 180 200
x 10^-7
Error = 4.9501e-007
6
4
2
0
0 20 40 60 80 100 120 140 160 180 200

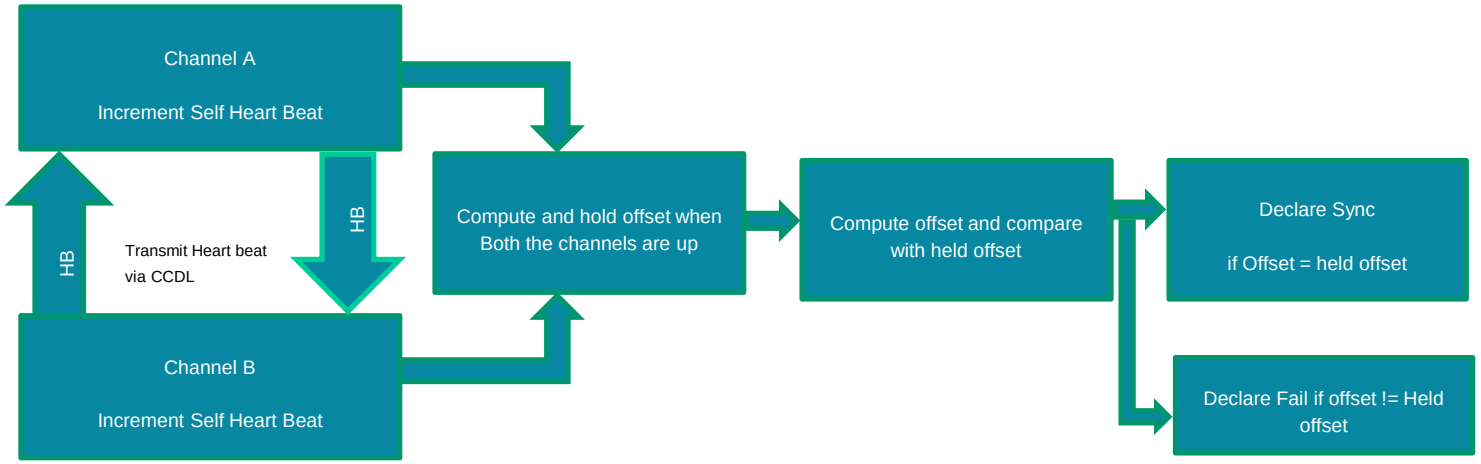
Errors

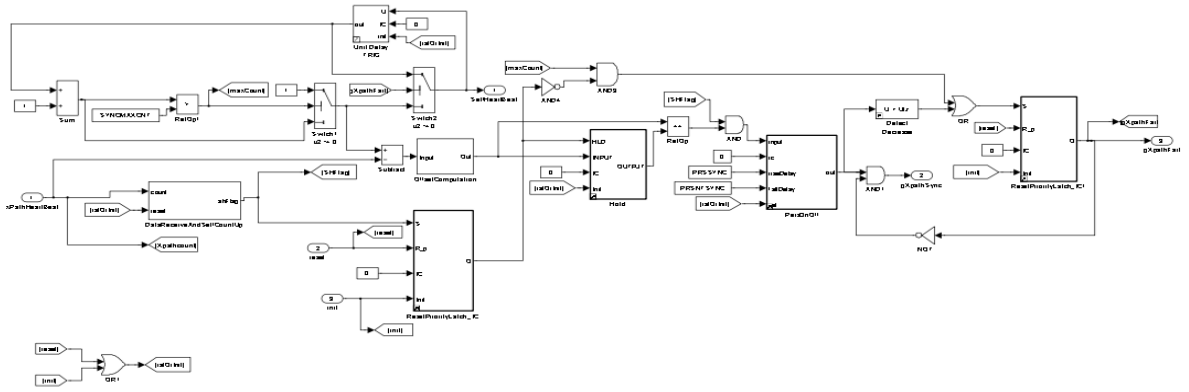
- Moog does hand coding – therefore there are going to be errors. ***Testing finds them!***
- Lookup table data mismatch in the 4^{th} decimal place
- Logic errors between the model and code
- Input not connected properly in the model
- Initialization errors – first frame
- Errors in mode transition
- One frame delays due to call order errors

Two Channel Operation

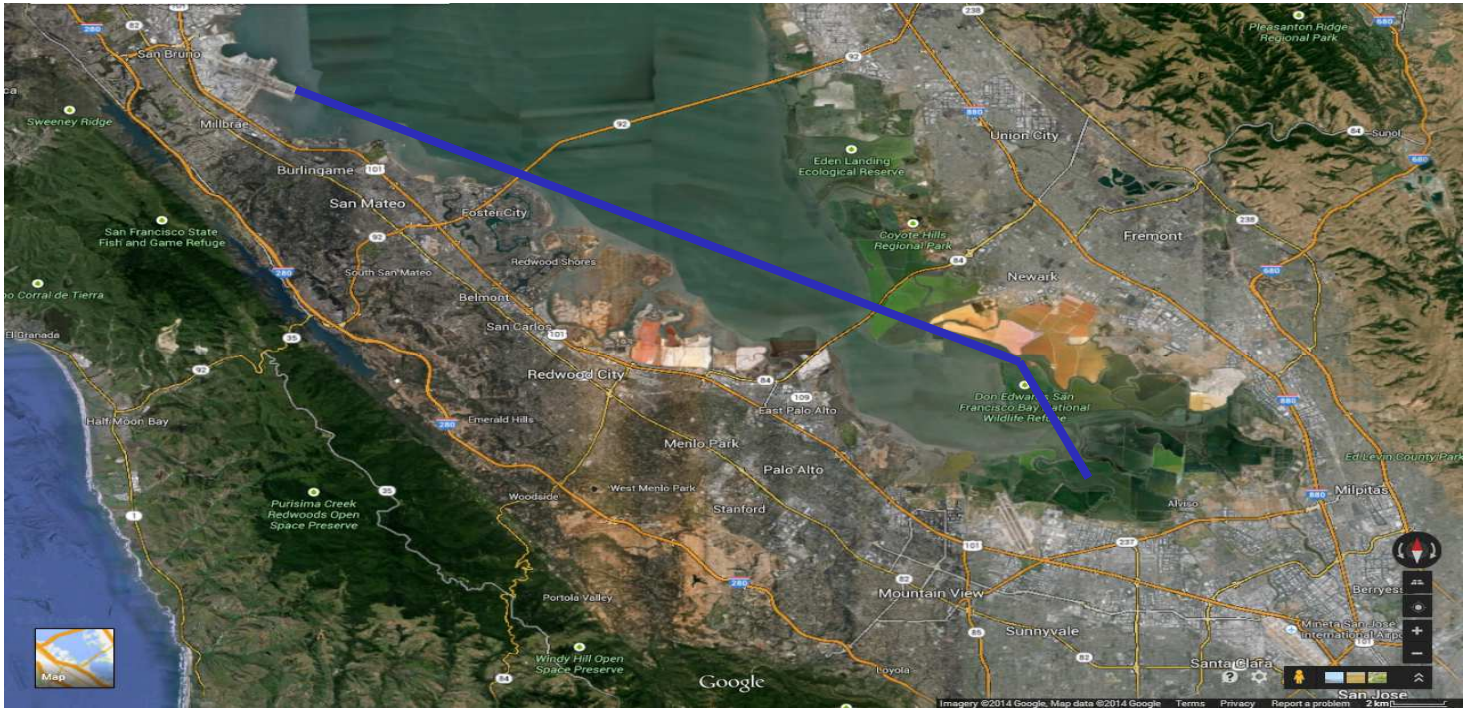


Cross Channel Sync Logic



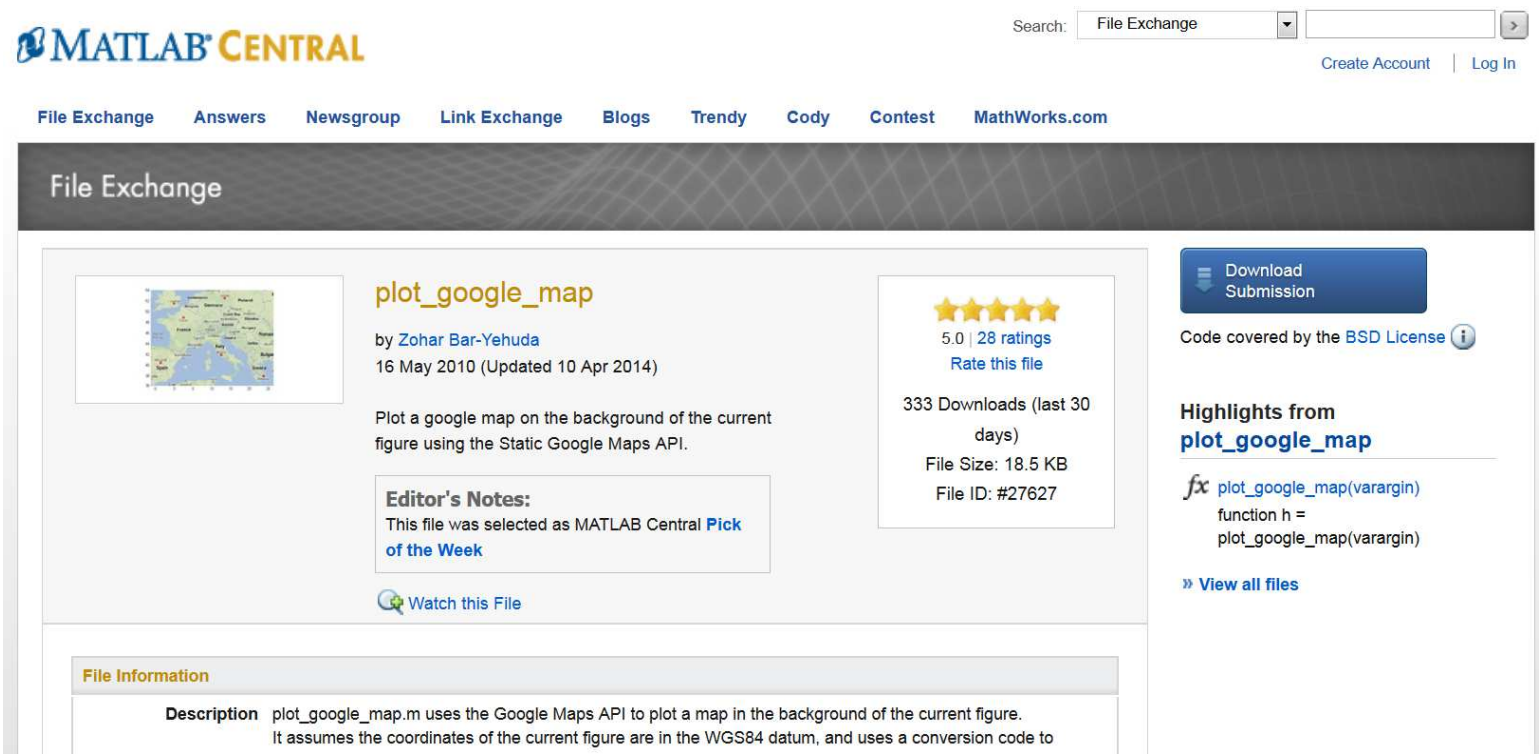


Demo


San Bruno
Sweeney Ridge
Millbrae
Burlingame
San Mateo
Foster City
San Francisco State Fish and Game Refuge
Redwood Shores
South San Mateo
Belmont
San Carlos
Redwood City
Emerald Hills
Half Moon Bay
Menlo Park
Palo Alto
East Palo Alto
Woodside
West Menlo Park
Stanford
Purisima Creek Redwoods Open Space Preserve
Portola Valley
Windy Hill Open Space Preserve
Mountain View
Sunnyvale
Santa Clara
San Jose
Milpitas
Alviso
Union City
Eden Landing Ecological Reserve
Coyote Hills Regional Park
Fremont
Newark
Pleasanton Ridge Regional Park
Sunol
Google
Map
Imagery ©2014 Google, Map data ©2014 Google
Terms
Privacy
Report a problem

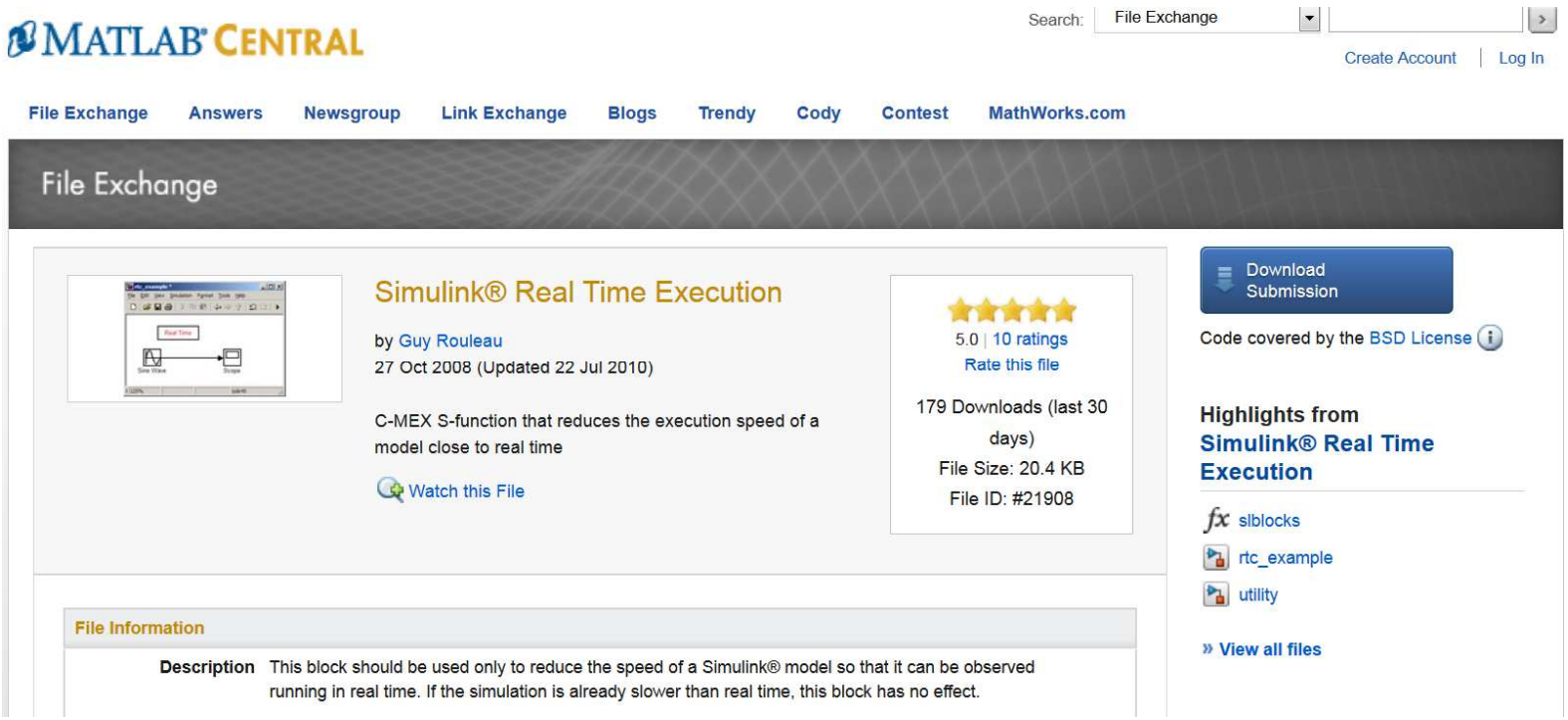
Google Maps

- plot_google_map - by Zohar Bar-Yehuda



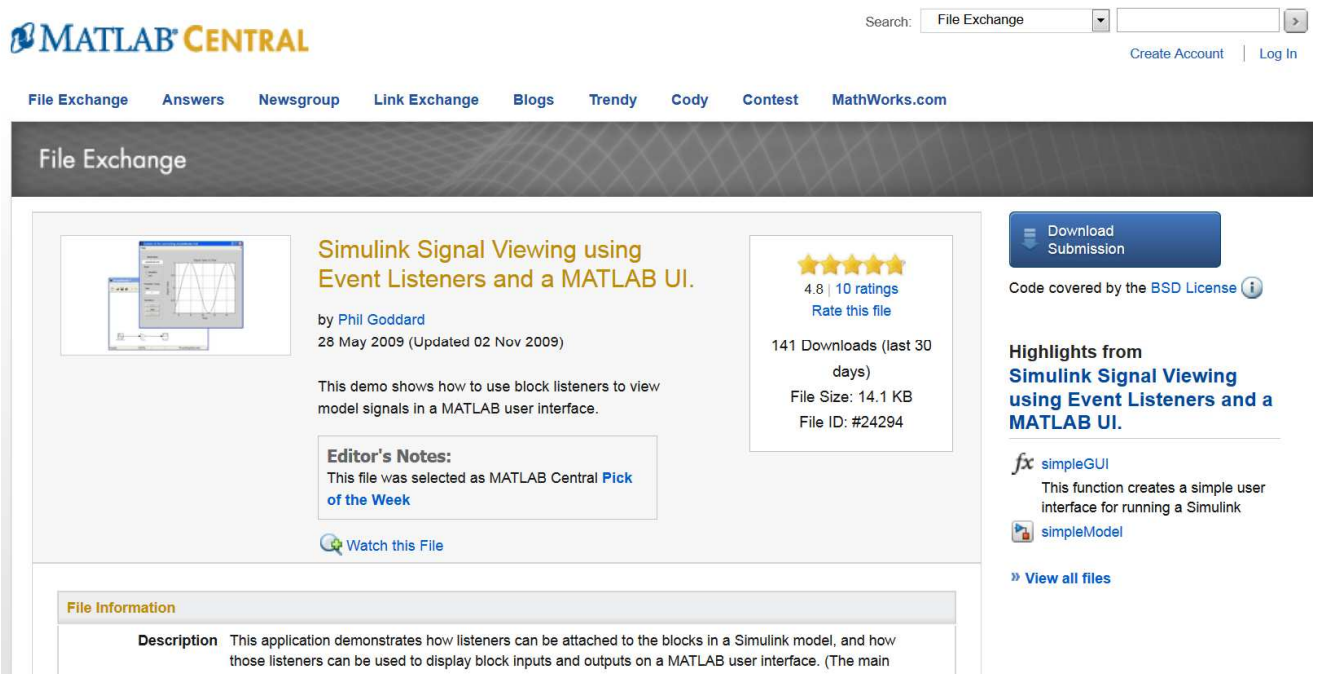
Real time simulation

- Simulink® Real Time Execution - by Guy Rouleau



Matlab UI Interface

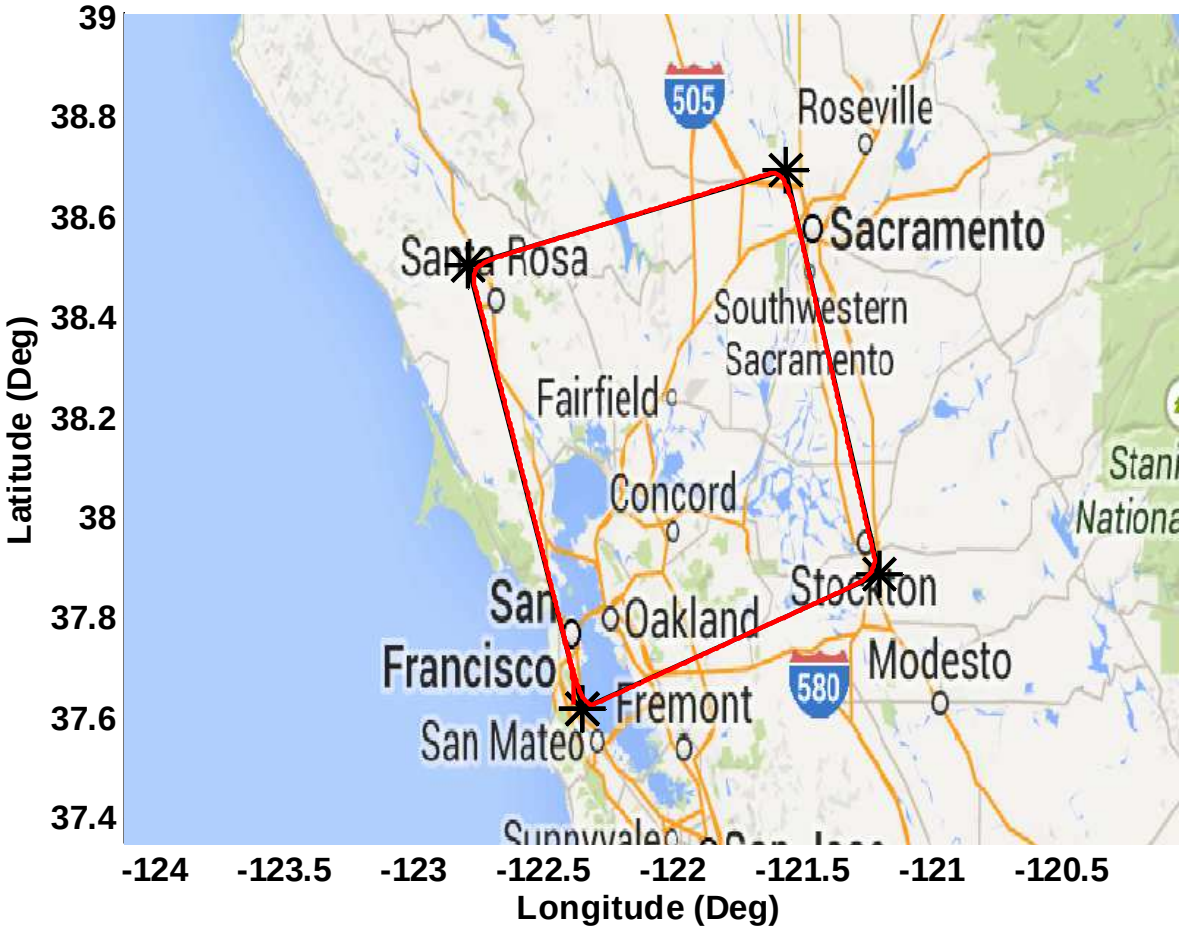
- Simulink Signal Viewing using Event Listeners and a MATLAB UI - by Phil Goddard



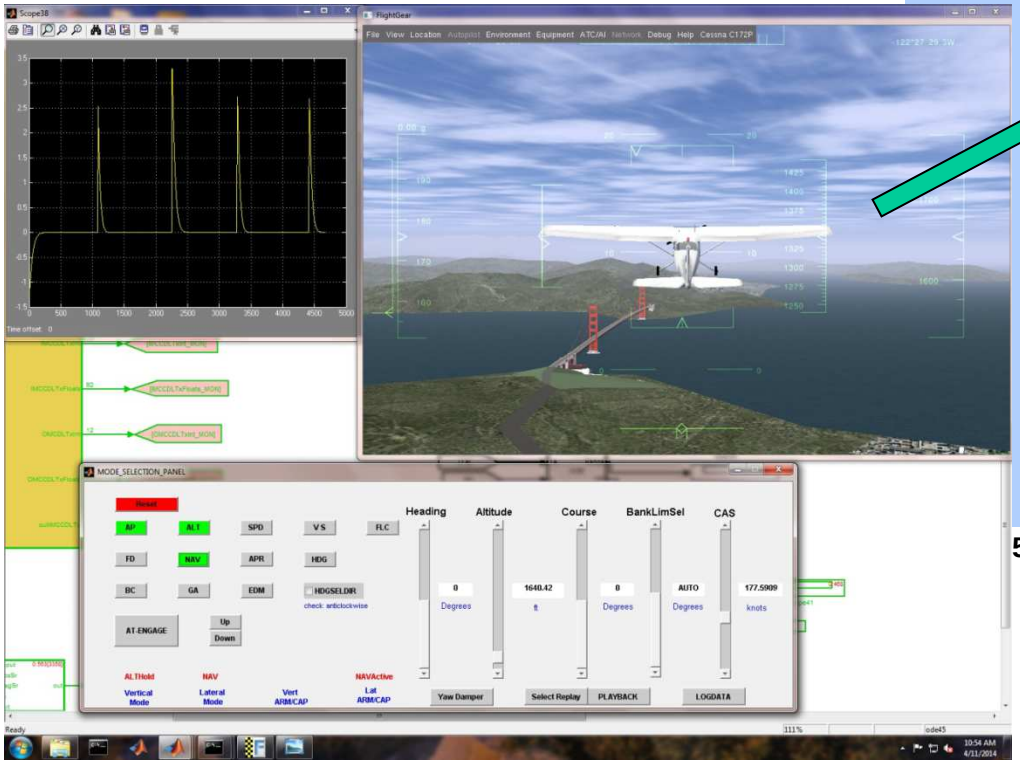
GPS Navigation

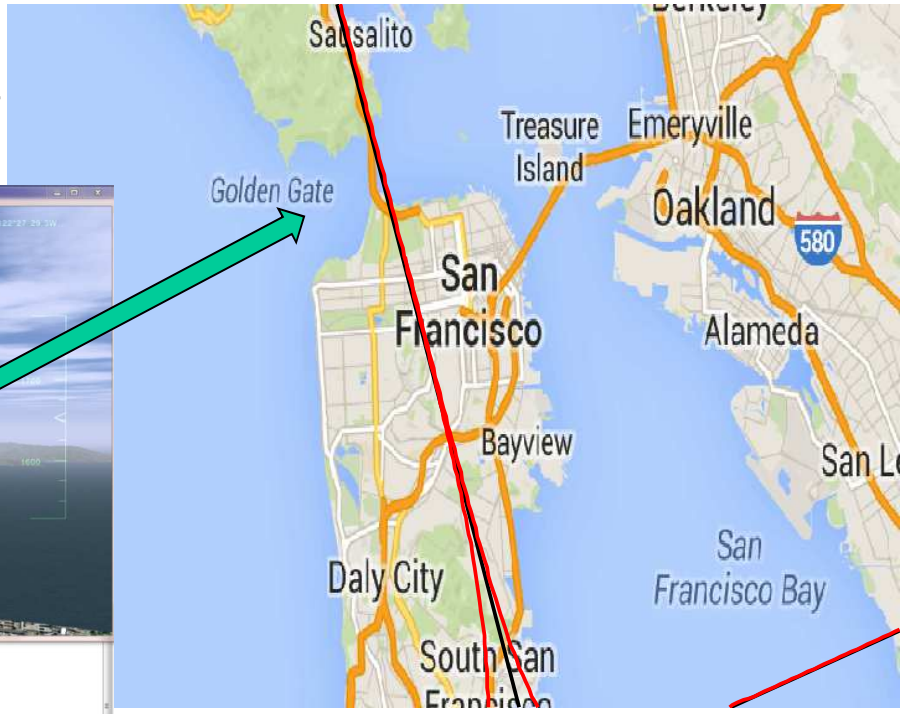
Black is desired track. Each leg is 100 Km

Red is aircraft trajectory



GPS Navigation





Matlab Toolboxes …

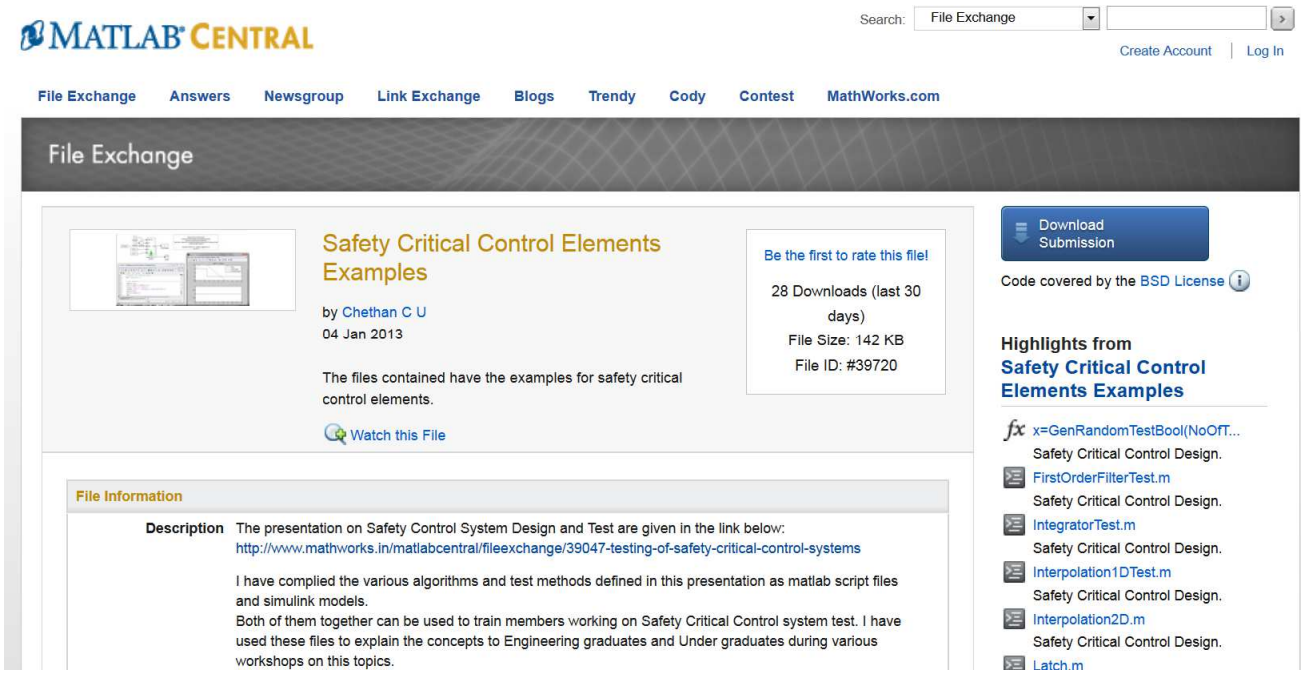
- We used these toolboxes for our design
  - **Matlab and Simulink**
  - **Control System Toolbox**
  - **Aerospace Blockset**
  - **Real Time Coder**

Our Contributions

We have tried to give back to the Matlab community. It has made a moderate impact going by the download rate and comments …

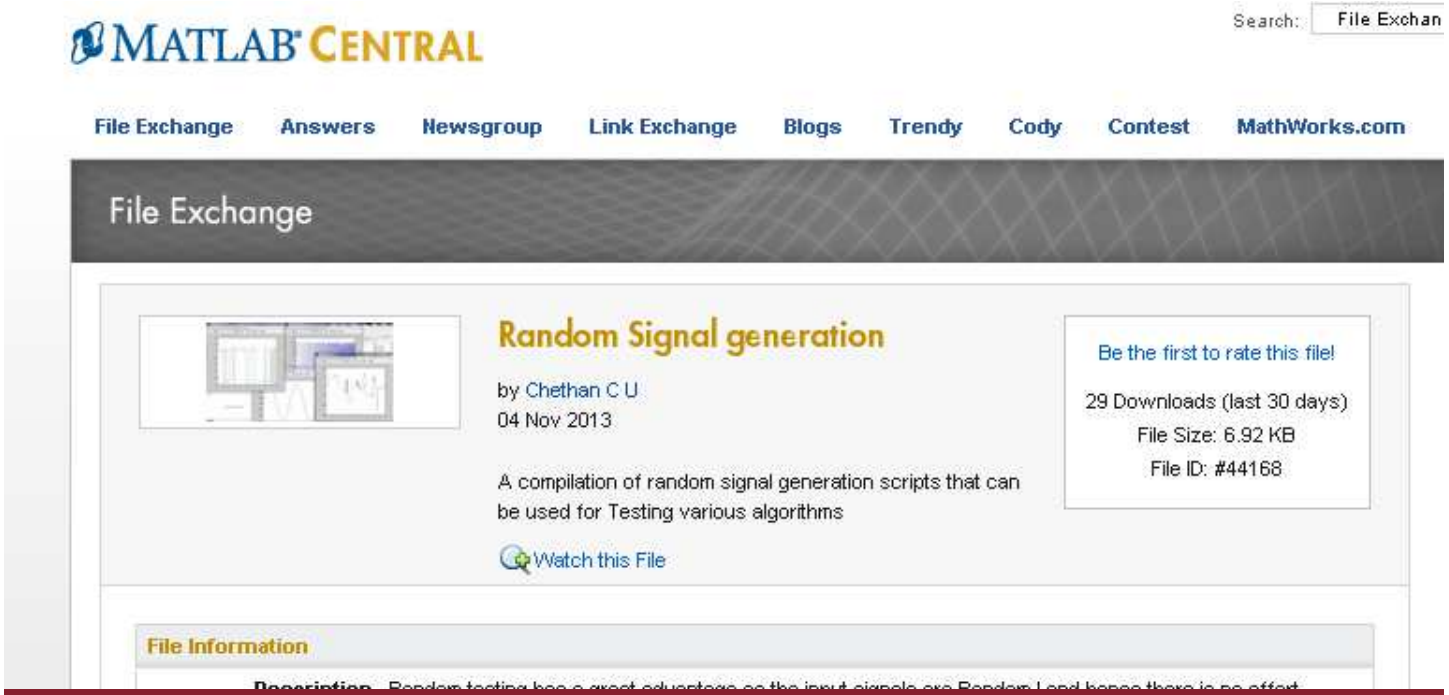
Safety Critical Blocks

- Safety Critical Control Elements Examples - by Chethan C U



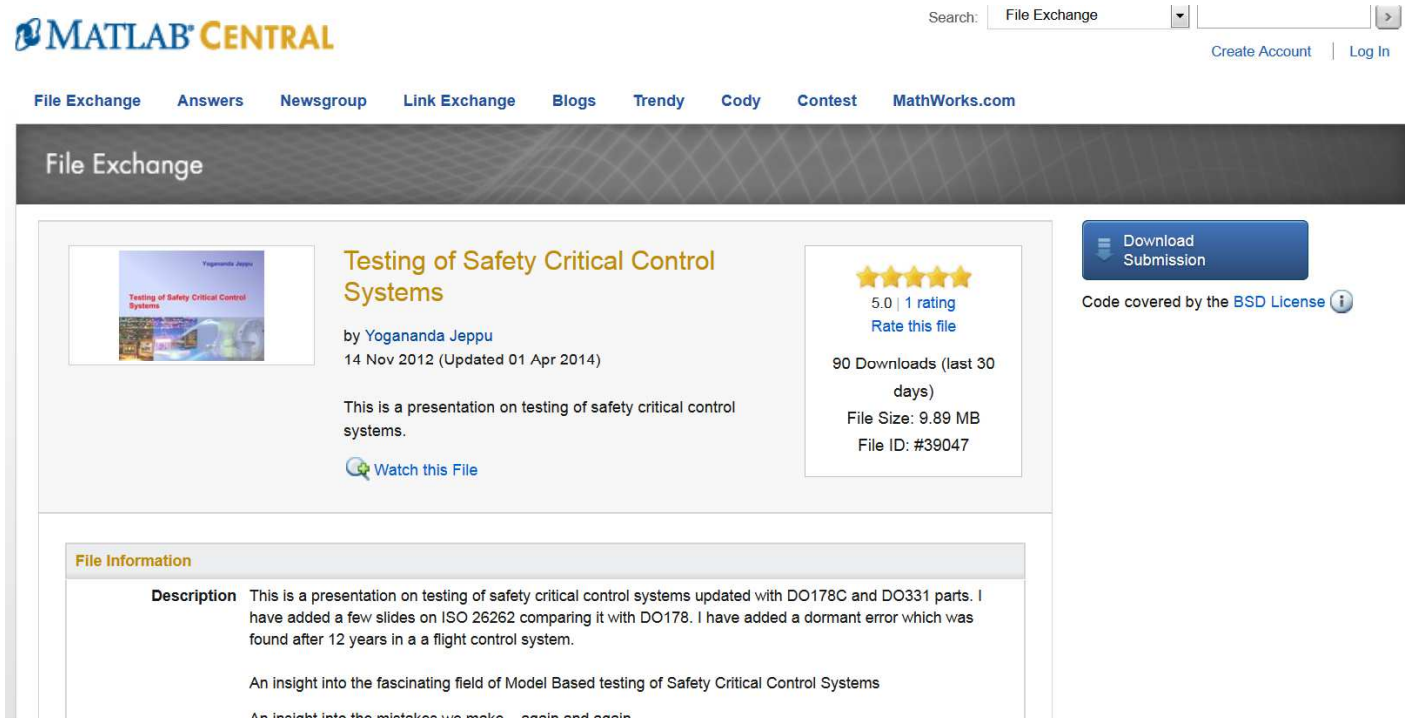
Random Signal Generation

- Random Signal Generation-
  by Chethan C U



Testing Safety Critical Systems

- Testing of Safety Critical Control Systems - by Yogananda Jeppu



Final Words

- *The idea is to try to give all the information to help others to judge the value of your contribution; not just the information that leads to judgment in one particular direction or another. - Richard P. Feynman*

- Finally - Thank you Mathworks for making this possible

- jyogananda@moog.com

Video – 3 Mins. Questions?