AEROSPACE ENGINEERING
DEPT. FACULTY OF
ENGINEERING CAIRO
UNIVERSITY
GIZA, 12613, EGYPT

قسم هندسة الطيران و الفضاء
كلية الهندسة
جامعة القاهرة
الجيزة – جمهورية مصر العربية

AER408

Aerospace Guidance & Control Systems

Task (5)

Longitudinal Autopilot Design

Before you start the design process:

- You will need all the transfer functions of the full Longitudinal dynamics model (4x4) available in your MATLAB workspace

- Put these lines in your code you will need those transfer functions too to be in your workspace through the design process

```
% Servo Transfer Function
servo = tf(10,[1 10]);
integrator = tf(1,[1 0]);
differentiator = tf([1 0],1);
engine_timelag = tf(0.1 , [1 0.1]);
```

Introduction:

The objective in this part of our project is to design ***"The longitudinal Autopilot"*** for a conventional fixed wing airplane. The rule of the longitudinal autopilot is to control the motion of the airplane in the longitudinal plane, shortly ***"it controls the elevator & thrust to achieve the desired command of Altitude or Climb angle"***

We will use the linearized state space model of the longitudinal dynamics (4x4) to represent the motion of the airplane in the longitudinal plane, which is a (Multi input Multi Output system MIMO), and to design our controllers we will use our previous studies about the (linear time invariant Single Input Single Output system SISO) to design our controllers like (linear PID and compensators). This method is called "Successive loop closure"

The main problem is "How to use the tools of SISO systems to treat with a MIMO system", and this is the main topic of any reference about autopilot design, i.e. the control loops layout

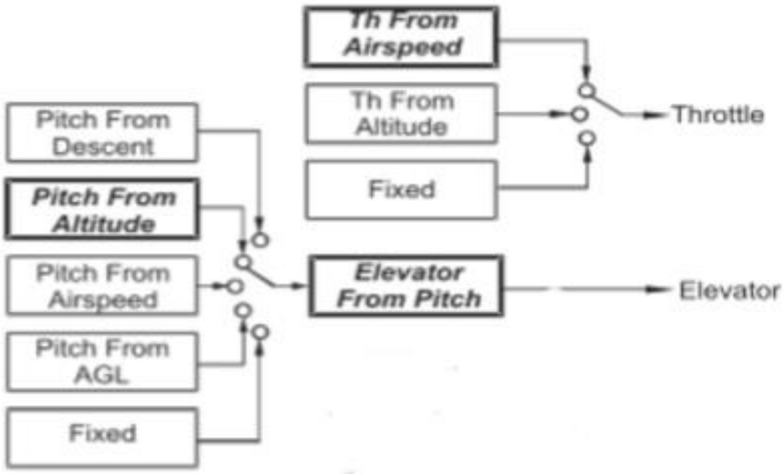
There are other ways to deal with linear MIMO systems like modern control techniques using the full state feedback and state observers. Each method has its own pros & cons and at the end of the day the designer should choose the best method to use based on his understanding of the system and other aspects like sensor's availability and so on

Remarks:

- These loops are not obligatory, you are totally free to use any control architecture you want to control your airplane with and the total response will judge your design

- The following loops are based on "Micro pilot autopilot manual" and the reference "Automatic control of Aircraft and missiles"

The loop configuration for "Level flight" condition is as follows



Throttle (δ_{th}) → control the speed (u)

And (δ_e) → control the pitch angle (Θ) → to achieve altitude (h)

Task Statement:

a) Design "Pitch controller with pitch rate feedback"

Block diagram: the block diagram designed is shown in the figure. It includes the transfer function of the pitch rate (q) w.r.t the elevator deflection (δ_e) i.e. (q/δ_e). also we can see the transfer function representing the servo dynamics which is the actuator that deflects the elevator

The "pitch rate feedback" is used here as an implementation of the "derivative gain K_d" part in a PD or PID controller

You can follow the design procedure in the accompanying document which describes in detail how to design the controller using the MATLAB control systems design toolbox

The design requirement in this loop is determined from the "Cooper-Harper Flying Qualities" table discussed in the lecture

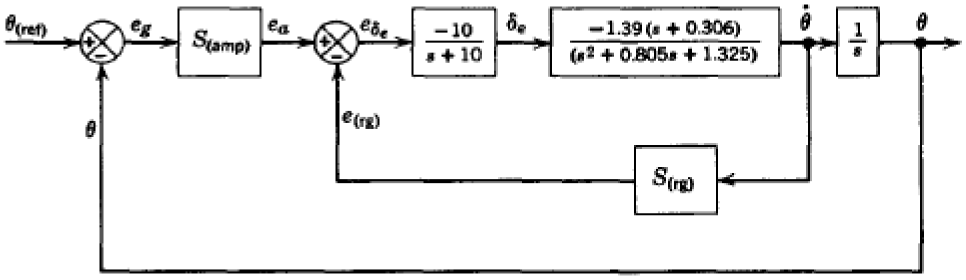
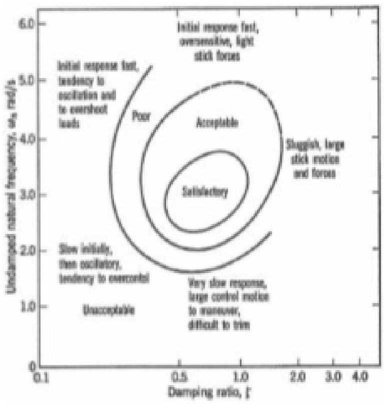


Figure 2-7 *Block diagram for the jet transport and displacement autopilot with pitch rate feedback added for damping.*

Cooper-Harper Rating and Flying Qualities

- Levels of flying qualities are correlated with data from flight dynamics parameters

Pilot state	Pilot rating	Level	Definition
	1 – 3½	1	Clearly adequate for the mission flight phase
	3½ – 6½	2	• Adequate to accomplish mission flight phase • Increase in pilot workload, or loss of effectiveness of mission, or both
	6½ – 9 (10)	3	• Aircraft can be controlled • Pilot workload excessive – mission effectiveness impaired • Category A flight phases can be terminated safely



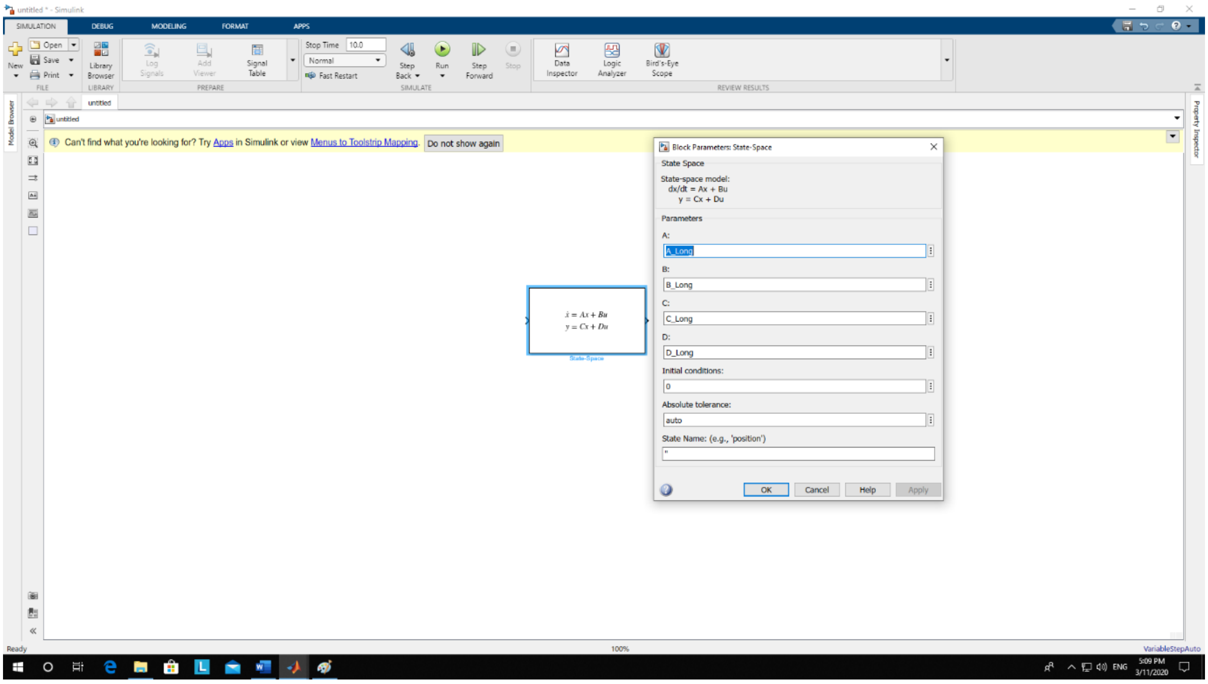
- Data is further correlated

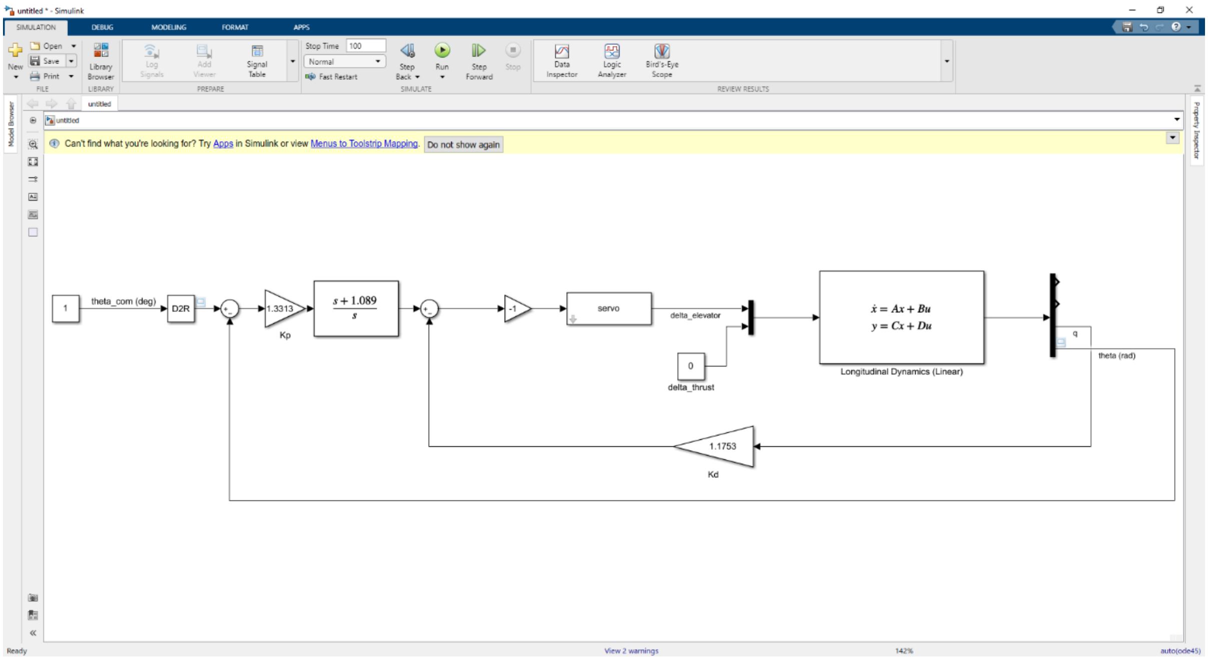
Phugoid mode	
Level 1	ζ>0.04
Level 2	ζ>0
Level 3	T>55s

Short period mode				
	Category A and C		Category B	
	ζmin	ζmax	ζmin	ζmax
Level 1	0.35	1.30	0.3	2.0
Level 2	0.25	2.00	0.2	2.0
Level 3	0.15	-	0.15	-

b) Test your Pitch controller on the full state space model

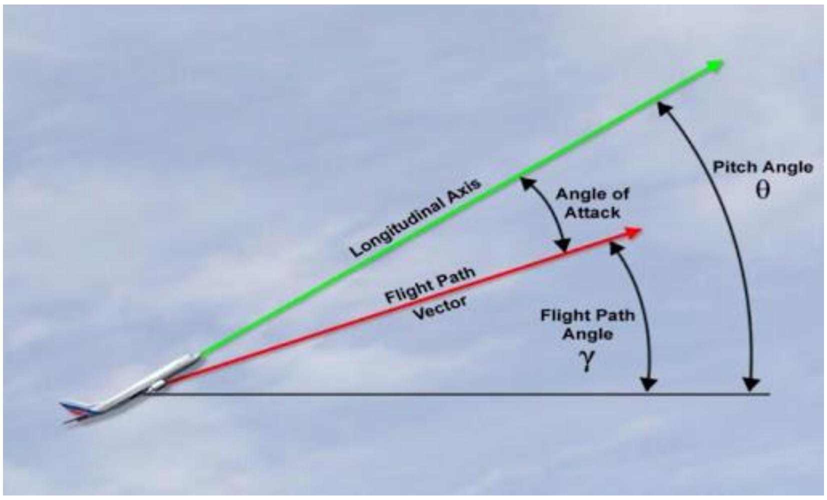
Through the design we used 1 transfer function relating one of the sates; which is pitch; to one of the inputs; which is elevator; we should test our designed controller on the MIMO system represented by the longitudinal state space model. The test is as shown in the figure





c) Necessity of velocity control (see Blakelock pages 87-89)

Using the simulink model you used to test your pitch controller, use the states of (θ & α=Uo*w) to calculate the climb angle (γ = θ – α).



- From your point of view, when you command the airplane to pitch upward (+ve θ) is the airplane supposed to climb upward or dive downward ?
  note: the altitude rate can be calculated as follows, it can be integrated to calculate altitude h

$$h^o = Vto*\sin(\gamma)$$

- How does the climb angle changes when you command a +ve pitch angle, climb upward or dive downward ?
  (please answer by showing the curves of γ & θ & θ_{com} with time)

- Does the simulation follows your predictions ?
  if not (this should be the case), Think for a reason (it is related to trim & balance of the airplane) ?

d) Design a "Velocity Controller"

Block diagram: the block diagram designed is shown in the figure. It includes the transfer function of the forward velocity (u) w.r.t thrust (δ_{thrust}) i.e. (u/ δ_{thrust}). Also we can see the transfer function representing the "throttle servo" dynamics and another block added to represent the "engine time lag" which is the time the engine take to build up thrust when the throttle stick is moved

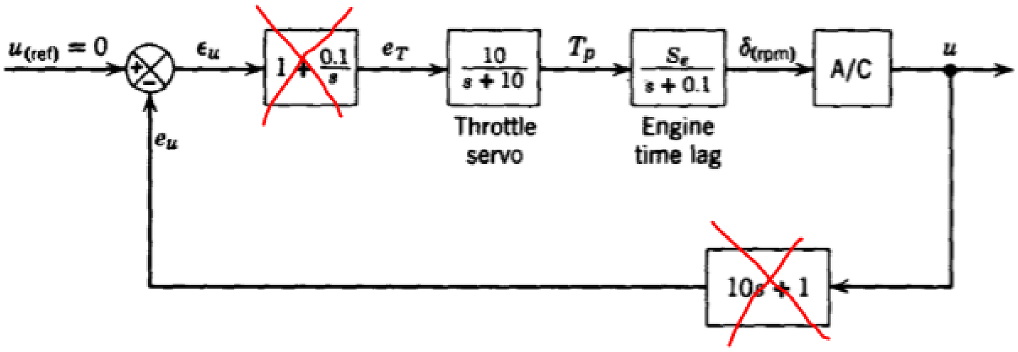


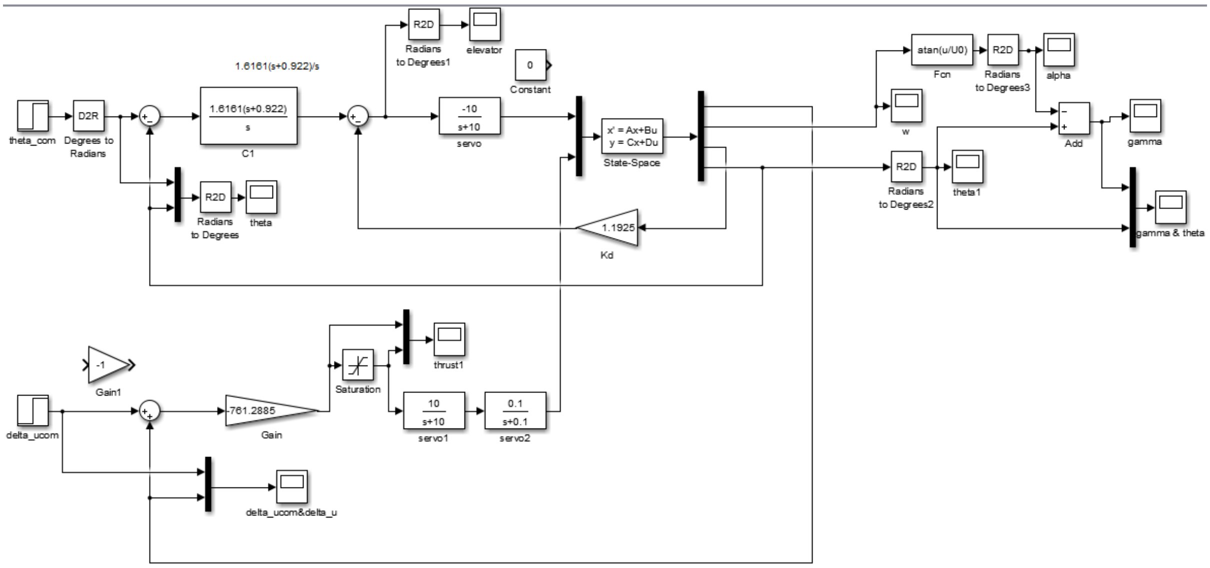
Figure 2-32 *Velocity control system using throttle.*

The rule of the velocity control loop is to *keep the forward velocity constant i.e. the trim value, this would also keep the angle of attack constant at the trim value, so any change in the pitch angle will be reflected on the climb angle !!. (+ve θ → +ve γ)*

e) Test the (Pitch & Velocity) controllers on the full state space model

Now you can add your controllers together to work simultaneously on the airplane

The pitch controller → calculates the elevator deflection. (achieves $\Delta\theta_{com}$)
The velocity controller → calculates the throttle value (Keep Δu=0)



f) Design "Altitude controller"

Now after we have controlled the pitch angle and velocity we can make an "Altitude Hold" controller, this loop uses the pitch control loop as its inner loop, i.e. the controller on this loop uses the error between the commanded altitude and the current altitude to calculate the desired pitch angle (θ_{com}) which changes the climb angle (γ) changing the altitude (h)

Block diagram: the block diagram designed is shown in the figure. It includes the closed loop transfer functions of the "Pitch control loop" the input to this transfer function is the ($\underline{\theta_{com}}$) and its output is the actual (θ)

Note: the figure treats with (θ^o) but we deal with (θ)

We need to relate (θ) to altitude (h) i.e. (h/θ) = ????

This relation can be found by different ways, amog them is to assume that (h) can be calculated by double ontegration of (az) the acceleration in the Z-axis direction

$$\ddot{h} = -a_z$$

$$h = -\frac{a_z}{s^2}$$

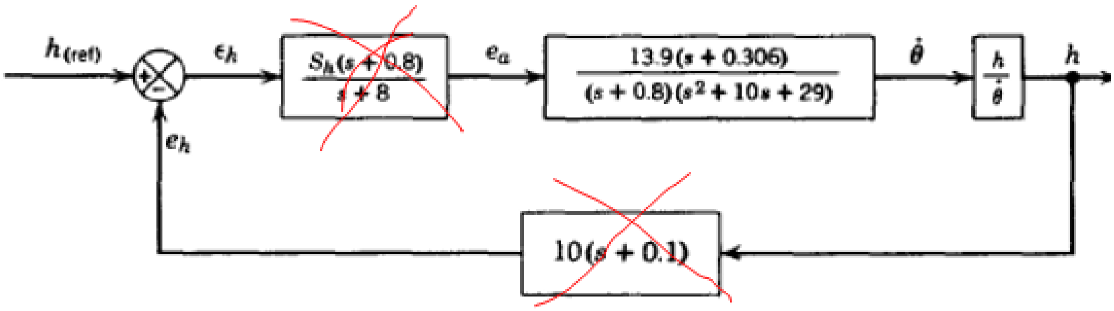


Figure 2-53 Block diagram for the outer loop of the altitude hold mode, where

$$\frac{h(s)}{\dot{\theta}(s)} = -\frac{1}{s^3}\frac{a_z(s)}{\theta(s)} = \cancel{\frac{-0.135(s-4.89)(s+4.89)}{s^2(s+0.306)}}\ \frac{\text{ft}}{\text{deg / sec}}$$

From the (RBD) Equations of motion, acceleration in the Z-axis is as follows

$$a_z = \dot{w} + pv - qu$$

After linearization

$$a_z = \dot{w} - U_o q$$

$$h = -\frac{\dot{w} - U_o q}{s^2}$$

$$\frac{h}{\dot{\theta}} = -\frac{1}{s^2} * \left(\frac{\dot{w}}{\dot{\theta}} - U_o . \frac{q}{\dot{\theta}}\right)$$

$$\frac{h}{\dot{\theta}} = -\frac{1}{s^2} * \left(\frac{w/\delta e}{\theta/\delta e} - U_o\right)$$

Finally we arrive at the transfer function relating the altitude (h) to pitch angle (θ)

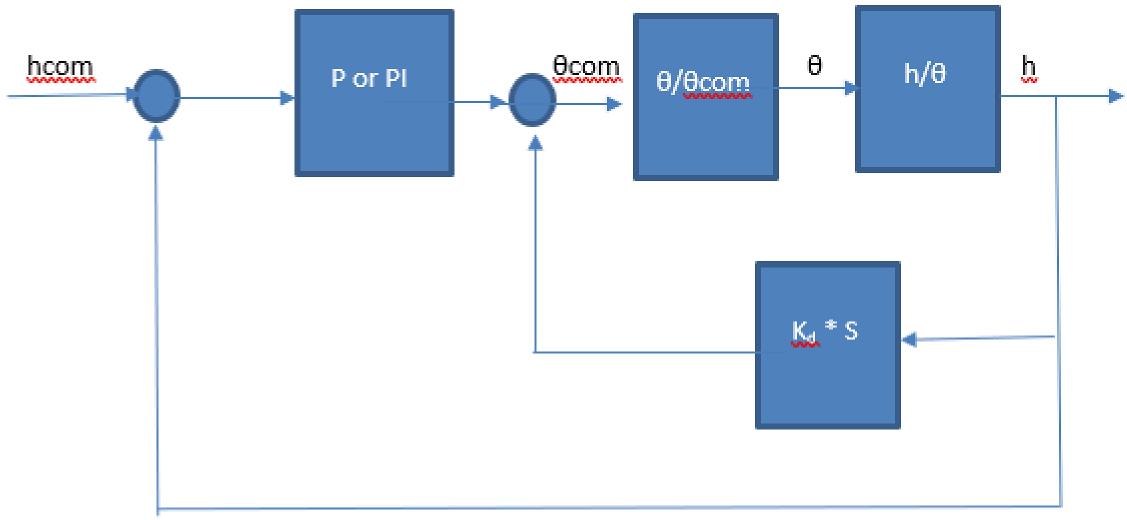
$$\frac{h}{\theta} = -\frac{1}{s} * (\frac{w/\delta e}{\theta/\delta e} - U_o)$$

Or

$$\frac{h}{\theta} = -\frac{U_o}{s} * (\frac{\alpha/\delta e}{\theta/\delta e} - 1)$$

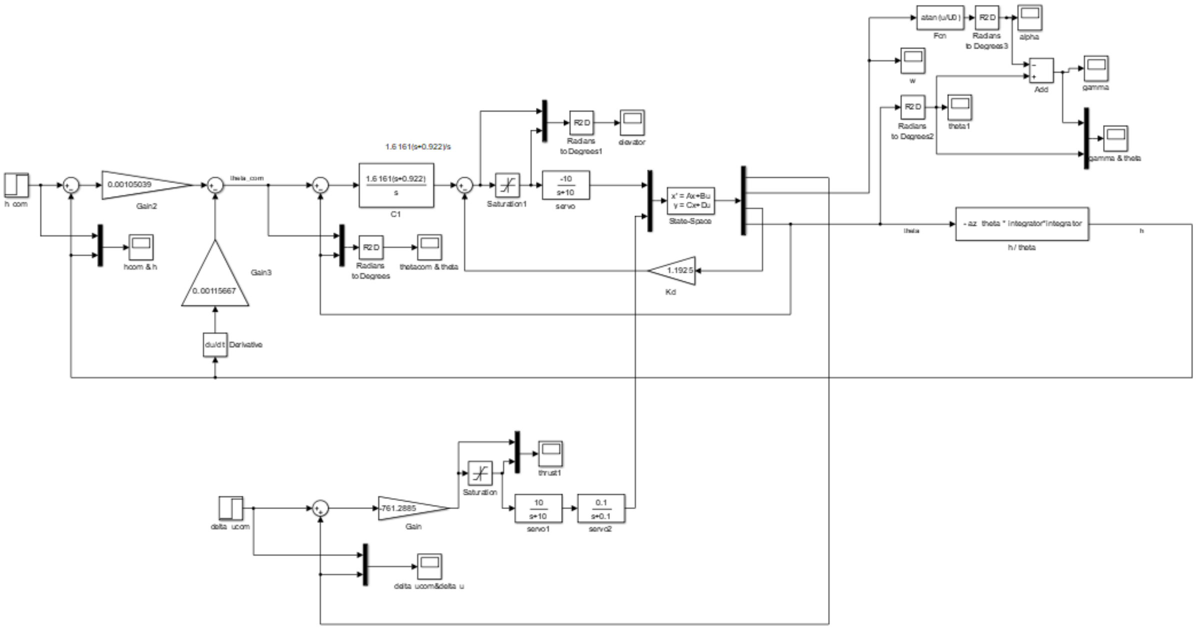
Where (w/δe) and (θ/δe) are the transfer functions obtained from the (4x4) state space model and (θ/θcom) is the closed loop of the pitch control loop, you can get it from the SISO tool easily by Expoting it to the workspace

The block diagram we will use to design our controller is as follows



g) Test the "Altitude & Velocity Controllers" together on the full state space model

You can test your controllers on the MIMO system and check its efficiency by commanding a step input in Altitude and watch your autopilot controlling the elevator and throttle limbimg your airplane up to the altitude you commanded



h com
0.00105039
Gain2
theta_com
hcom & h
0.00115667
Gain3
du/dt Derivative
1.6161(s+0.922)/s
1.6161(s+0.922)
s
C1
R2D
Radians
to Degrees
thetacom & theta
Saturation1
-10
s+10
servo
R2D
Radians
to Degrees1
elevator
x' = Ax+Bu
y = Cx+Du
State-Space
1.1925
Kd
atan(u/U0)
Fcn
R2D
Radians
to Degrees3
alpha
w
Add
gamma
R2D
Radians
to Degrees2
theta1
gamma & theta
theta
- az theta * integrator*integrator
h / theta
h
delta ucom
761.2885
Gain
Saturation
thrust1
10
s+10
servo1
0.1
s+0.1
servo2
delta ucom&delta u