FDD AND FTC DESIGN AND IMPLEMENTATION TO A QUAD-ROTOR UAV

Mahyar Abdolhosseini
Jian Nan Zheng
Hassan Al Garni

ENGR 691X: Fault Diagnosis and Fault-Tolerant Control Systems
Final Project - Fall 2010


Camera: Olympus E-P2 4/3rds sensor
Camera art filter: Diorama
Settings:
ISO 400, 17mm, 0EV, f/2.8, 1/60

Draganflyer X8

OUTLINE

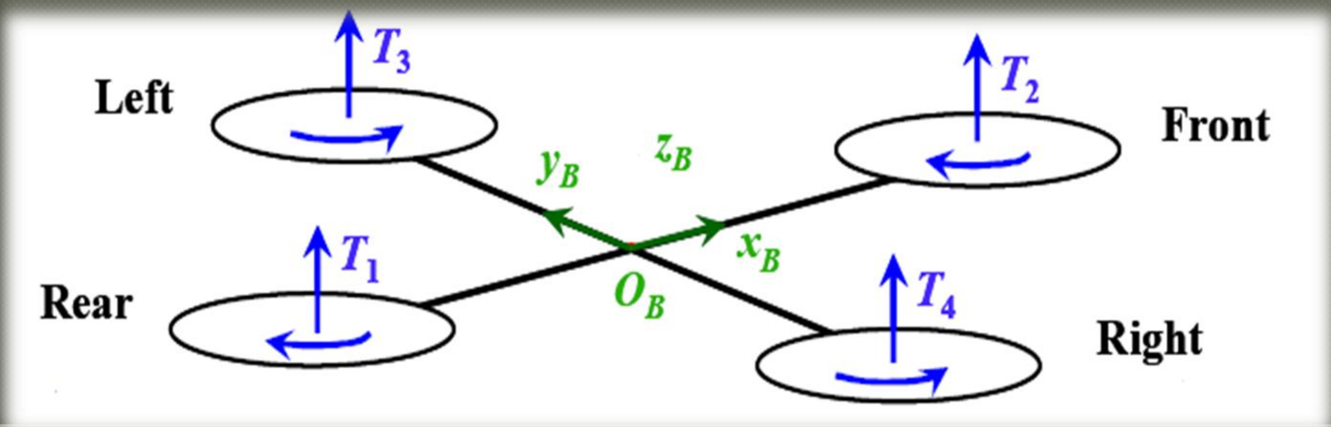
- Objectives
- Equations of motions
- LQR & PID controller concepts
- AFTCS concepts
- AFTCS implementation
- Conclusion

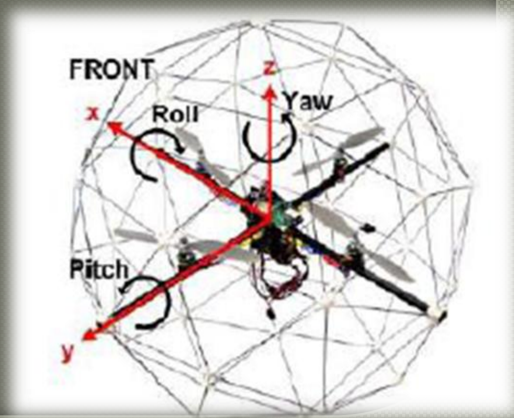
Project Objectives

- *Apply AFTCS to Q-Ball, so it can maintain a certain attitude despite of partial loss in Actuator*
- *Use two different approaches for Reconfigurable Controller part*
- *Post failure system can maintain stability and performance*
- *Hardware Implementation (Fault-free)*

Equations of Motions:

- The Quad-rotor has 6-DOF, it is equipped just with 4 propellers
- Four Basic Movements :
- ***Throttle U1*** **(**Ω1 = Ω2 = Ω3 =Ω4)
- ***Roll U2 (***Ω3↑& Ω4↓)
- ***Pitch U3 (***Ω1↑ & Ω2↓)
- ***Yaw U4 (***Ω1 Ω2↑& Ω3 Ω4↓)





Cont.

- Simplified Models:

$$\begin{bmatrix} U1 \\ U2 \\ U3 \\ U4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & L & -L \\ L & -L & 0 & 0 \\ C & C & -C & -C \end{bmatrix} \begin{bmatrix} T1 \\ T2 \\ T3 \\ T4 \end{bmatrix} \qquad Ti = K \frac{w}{s+w} ui$$

- $$\ddot{x} = (\sin\Psi \sin\Phi + cos\Psi\ sin\theta\ cos\Phi)\ \frac{U1}{M}$$
  $$\ddot{y} = (\sin\Psi \sin\theta\ cos\Phi - cos\Psi\ sin\Phi)\ \frac{U1}{M}$$
  $$\ddot{z} = -g + (\cos\theta\ \cos\Phi)\ \frac{U1}{M}$$
  $$\ddot{\Phi} = \frac{U2}{Jxx}$$
  $$\ddot{\theta} = \frac{U3}{Jyy}$$
  $$\ddot{\Psi} = \frac{U4}{Jzz}$$

(Linear Quadratic Regulator) LQR

I. **LQR – Regulation Problem**

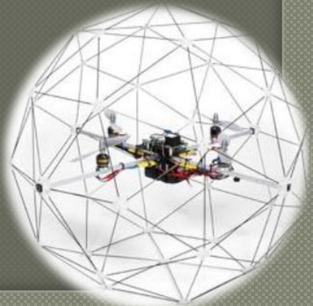
$$\dot{x} = Ax + Bu$$

$$u = -Kx + v$$

$$\dot{x} = (A - BK)x + Bv = A_{NEW}x + Bv$$

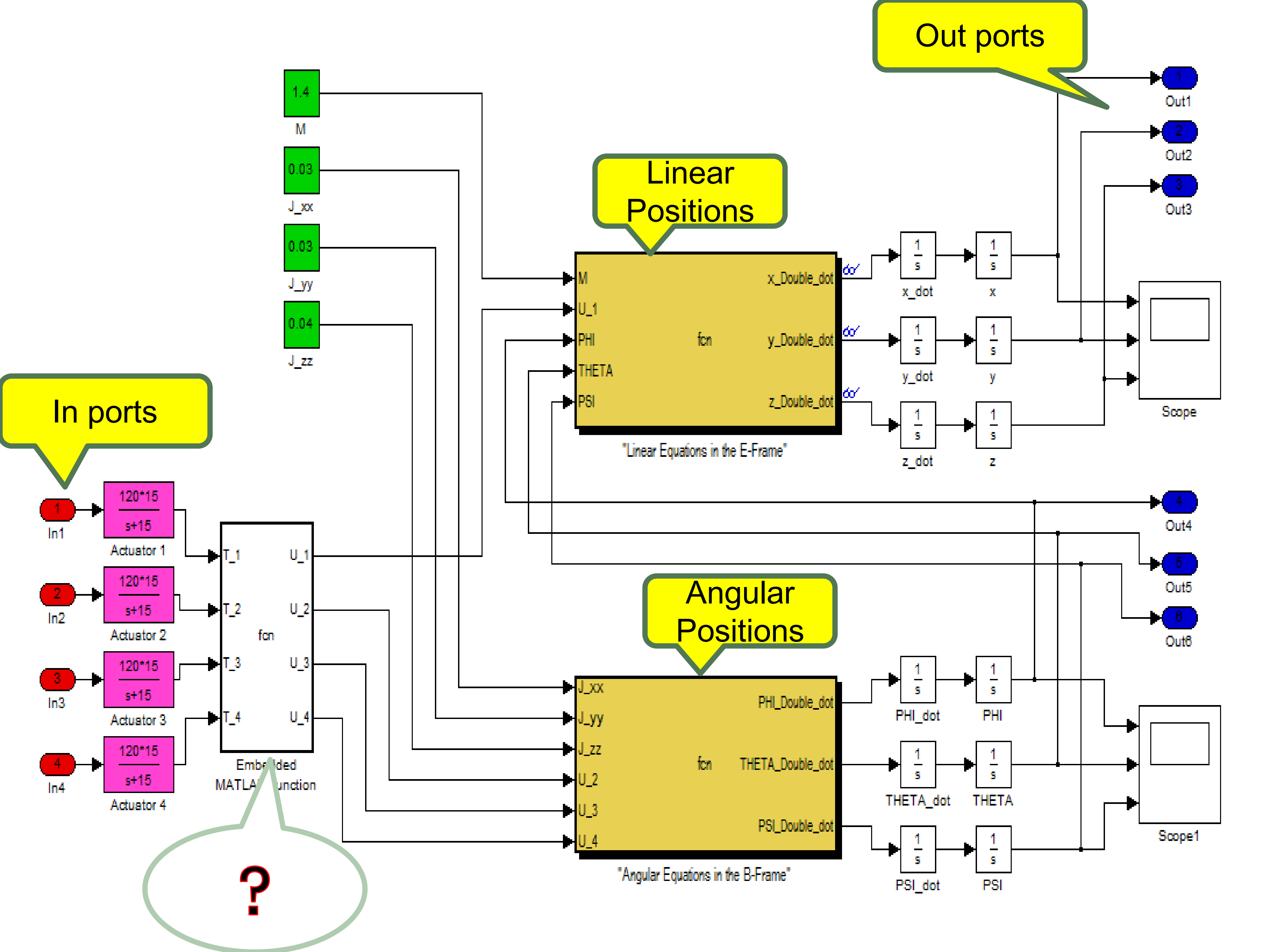
- The "cost function" is defined as a sum of the deviations of key measurements from their desired values.

- Still needs to specify the weighting ***Q*** & ***R*** factors and compare the results with the specified design goals

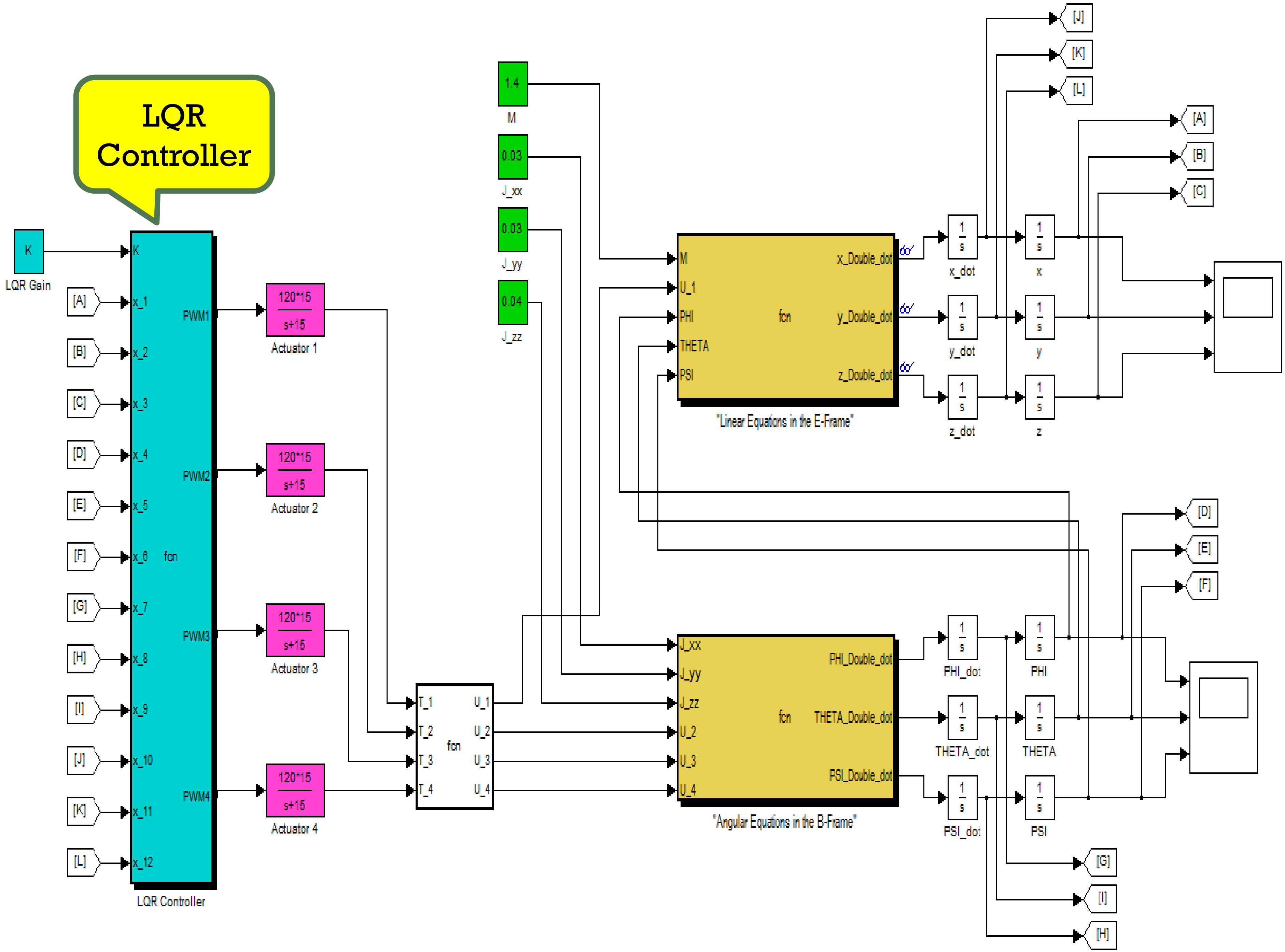


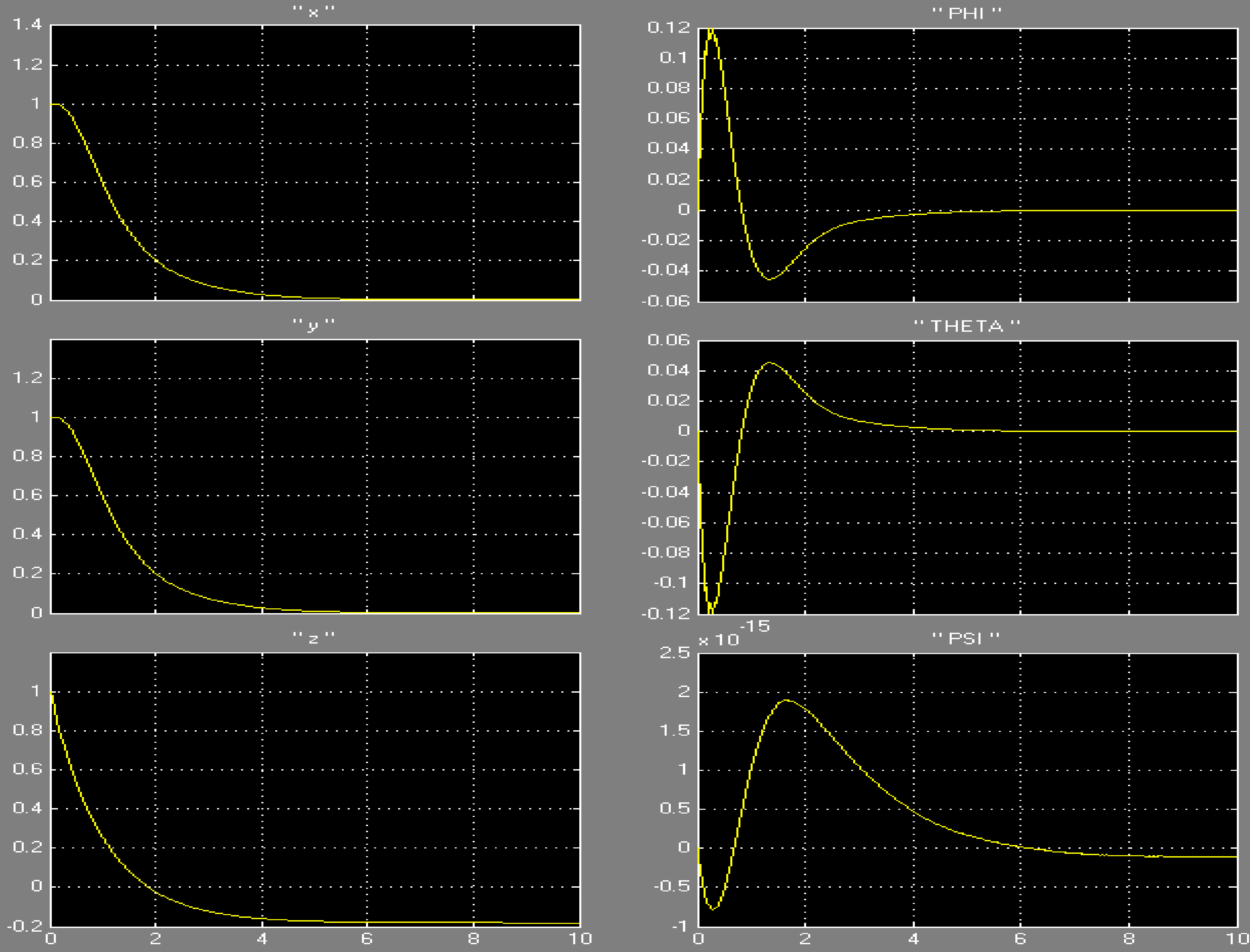
Cont.

- ***LQR - Regulator*** implies that brings all state variables to zero and stabilizes the control system
- ***K = lqr (A, B, Q, R)*** in MATLAB
- LQR controller is its limited applicability to just linear systems.
- In our project it was required to obtain linearized equations of motion of Q-Ball
- **MATLAB** routine ***"linmod"*** has been employed
- A "***Trim"*** *command used before linearization .* ***Trim Point***, also known as an equilibrium point, is a point in the parameter space of a dynamic system at which the system is in a steady state



Out ports
Linear Positions
In ports
Angular Positions
?
1.4
M
0.03
J_xx
0.03
J_yy
0.04
J_zz
In1
In2
In3
In4
120*15
s+15
Actuator 1
Actuator 2
Actuator 3
Actuator 4
T_1
T_2
T_3
T_4
U_1
U_2
U_3
U_4
fcn
PHI
THETA
PSI
x_Double_dot
y_Double_dot
z_Double_dot
"Linear Equations in the E-Frame"
J_xx
J_yy
J_zz
PHI_Double_dot
THETA_Double_dot
PSI_Double_dot
"Angular Equations in the B-Frame"
x_dot
x
y_dot
y
z_dot
z
PHI_dot
THETA_dot
PSI_dot
Scope
Scope1
Out1
Out2
Out3
Out4
Out5
Out6


LQR Controller
K
LQR Gain
[A]
[B]
[C]
[D]
[E]
[F]
[G]
[H]
[I]
[J]
[K]
[L]
K
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
x_9
x_10
x_11
x_12
fcn
PWM1
PWM2
PWM3
PWM4
LQR Controller
120*15
s+15
Actuator 1
Actuator 2
Actuator 3
Actuator 4
T_1
T_2
T_3
T_4
U_1
U_2
U_3
U_4
1.4
M
0.03
J_xx
0.03
J_yy
0.04
J_zz
M
U_1
PHI
THETA
PSI
x_Double_dot
y_Double_dot
z_Double_dot
"Linear Equations in the E-Frame"
J_xx
J_yy
J_zz
U_2
U_3
U_4
PHI_Double_dot
THETA_Double_dot
PSI_Double_dot
"Angular Equations in the B-Frame"
1
s
x_dot
y_dot
z_dot
x
y
z
PHI_dot
THETA_dot
PSI_dot
PHI
THETA
PSI


" x "
" y "
" z "
" PHI "
" THETA "
" PSI "
x 10-15
0
2
4
6
8
10

LQR – Tracking Problem

- Imagine a control system expressed in state space format as follows:

$$\dot{x} = Ax + Bu$$

- The same as for non-tracking problem/regulator, here the control signal is:

$$u = -KX$$

- Let's assume x = [x_1 x_2 x_3 … x_n], indicating n state variables.
- Also, imagine that there are reference values for x_{1d}, x_{2d}, x_{3d}, …, and x_{md} for which the controller is responsible:

$$X = [x_1, x_2, x_3, \ldots, x_n, z_{1d}, z_{2d}, z_{3d}, \ldots, z_{md}]$$

$$z_{id} = \int x_i - x_{id} \, dt$$

LQR

- With this definition the new representation of the system becomes:

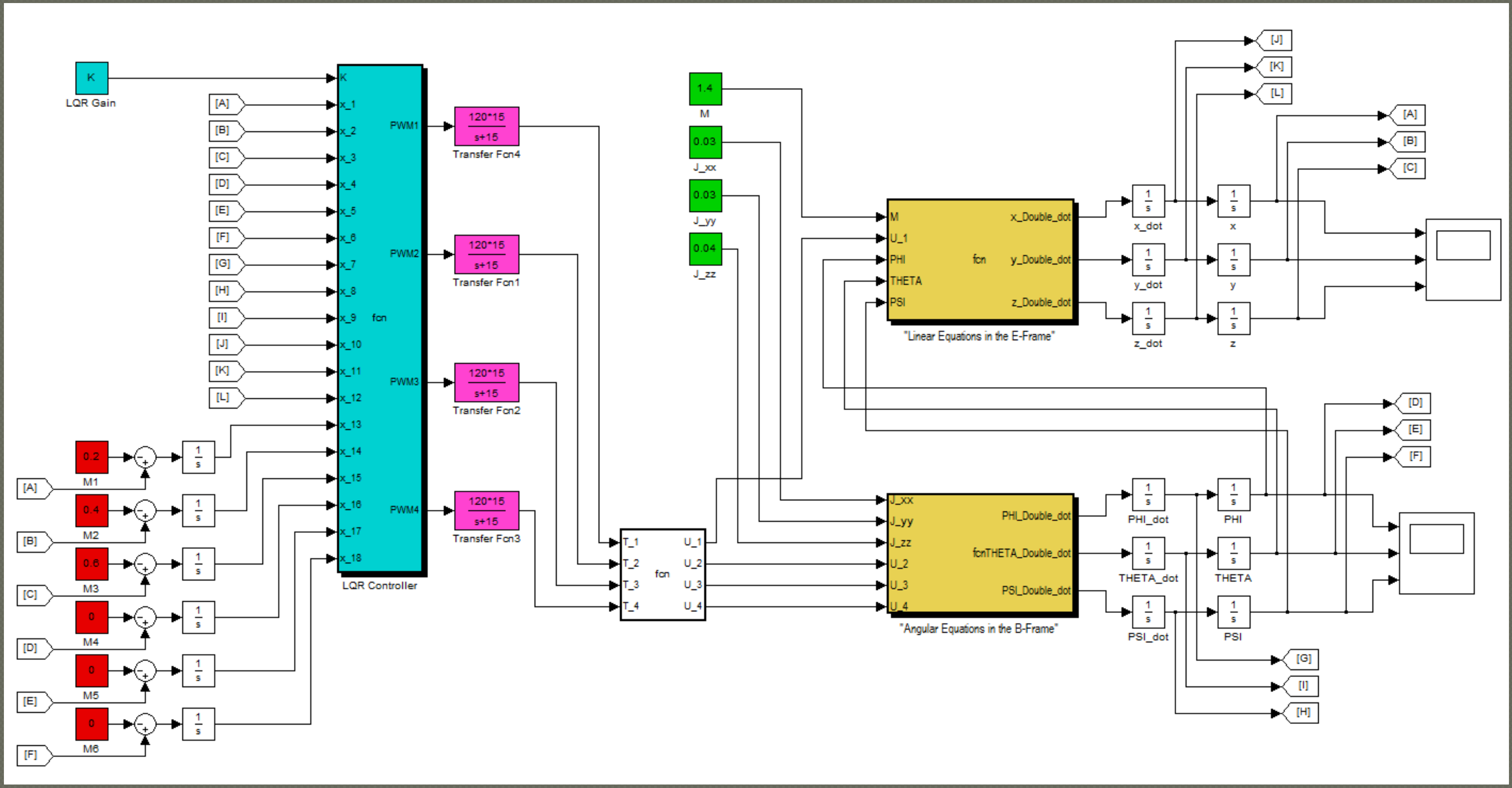
$$\dot{X} = \begin{bmatrix} [A] & [0] \\ \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}_{m*m} & [0] \end{bmatrix} X + \begin{bmatrix} [B] \\ [\,] \end{bmatrix} u + \begin{bmatrix} [0]_{n*m} \\ \begin{bmatrix} -1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -1 \end{bmatrix}_{m*m} \end{bmatrix} \begin{bmatrix} x_{d1} \\ x_{d2} \\ x_{d3} \\ \vdots \\ x_{dm} \end{bmatrix}$$

Or in a more compact form:

$$\dot{X} = \bar{A}X + \bar{B}u + B_d P_d$$

- Again, once the state space representation of the control system is obtained, design of LQR Controller is almost straight forward. *K = lqr (A_bar, B_bar, Q, R).*

Application of Design Methodology to QBall



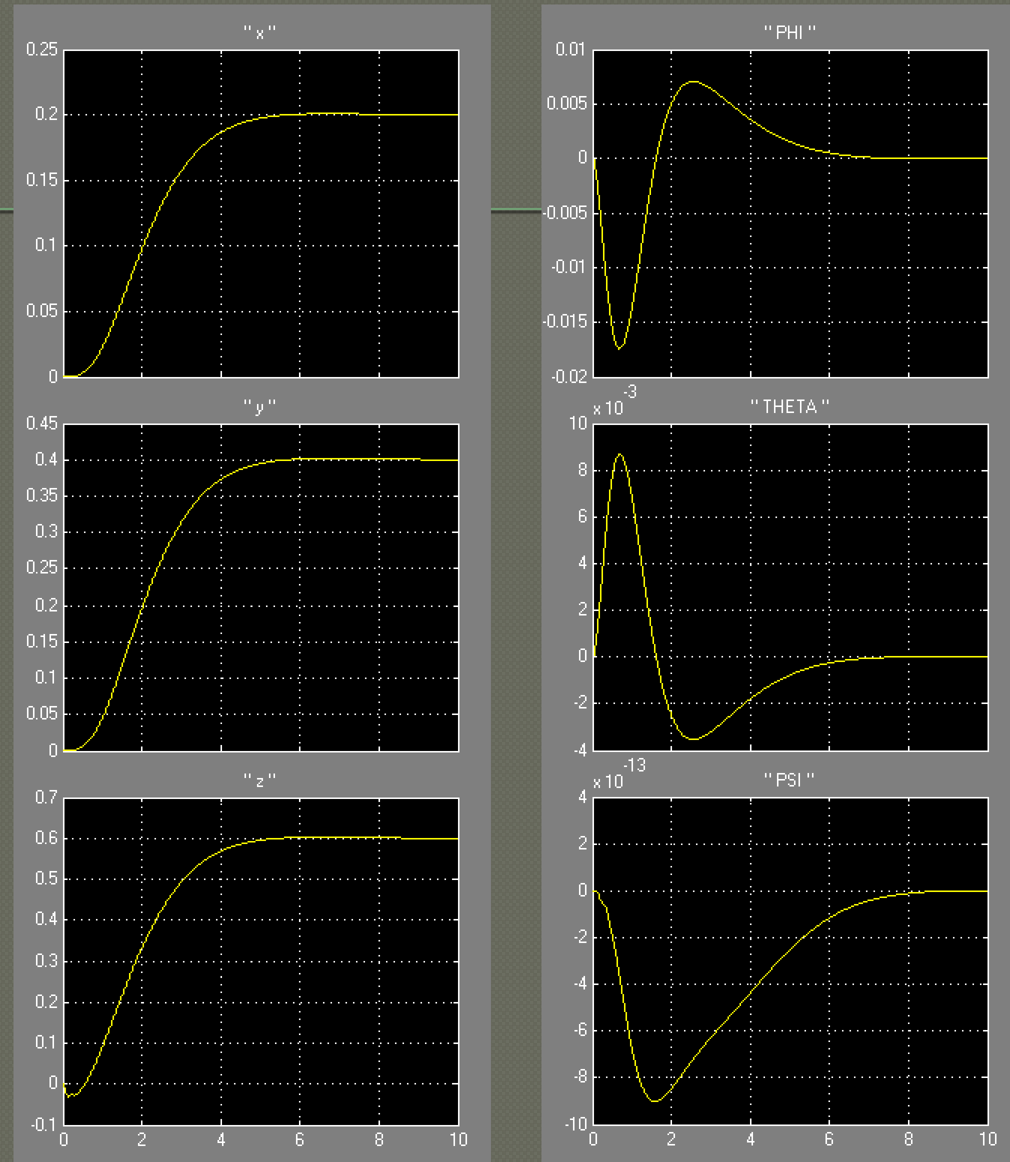
- Having chosen the values of these weighting matrices to be

  Q = eye(16)

  And

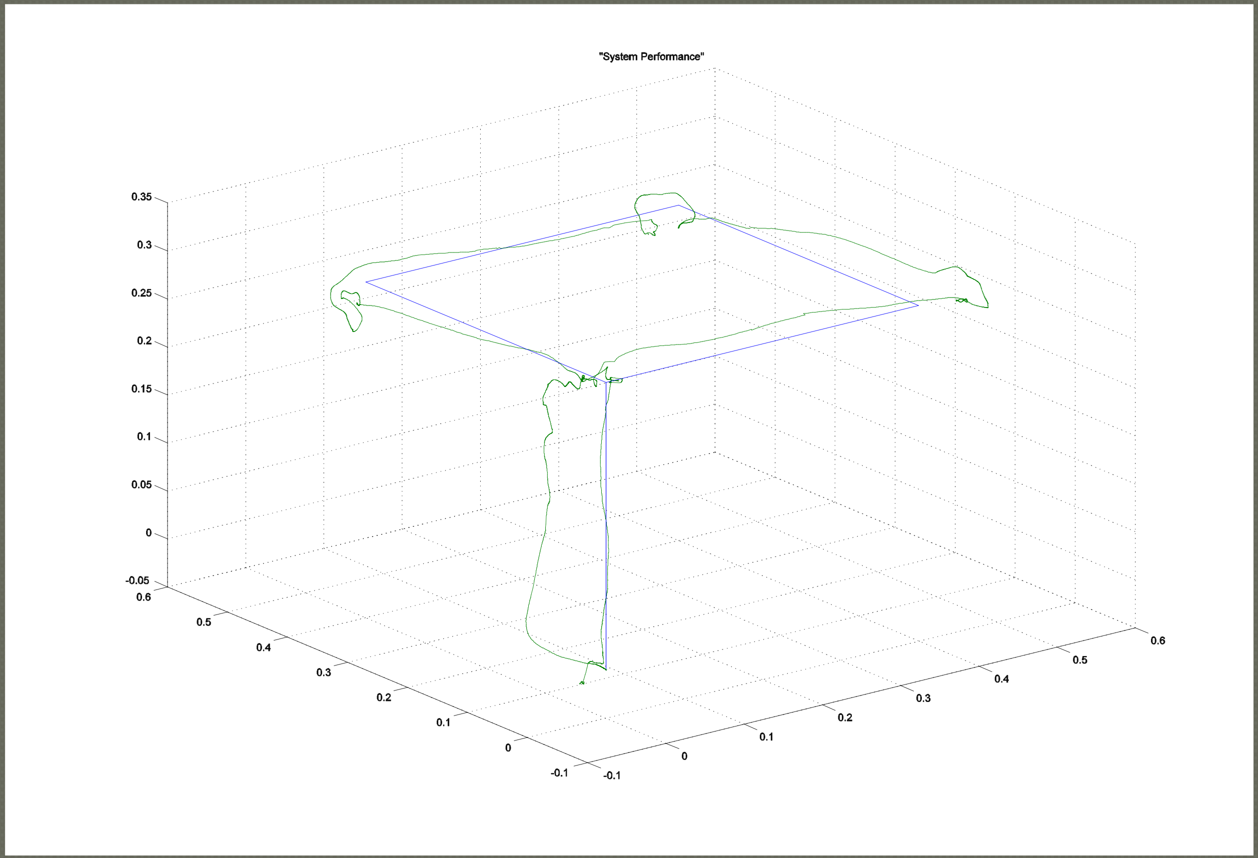
  R = eye(4)

- Next is the time response of the system for the following reference inputs.



Time Response of the System

LQR - Real System Results (Fault-Free)



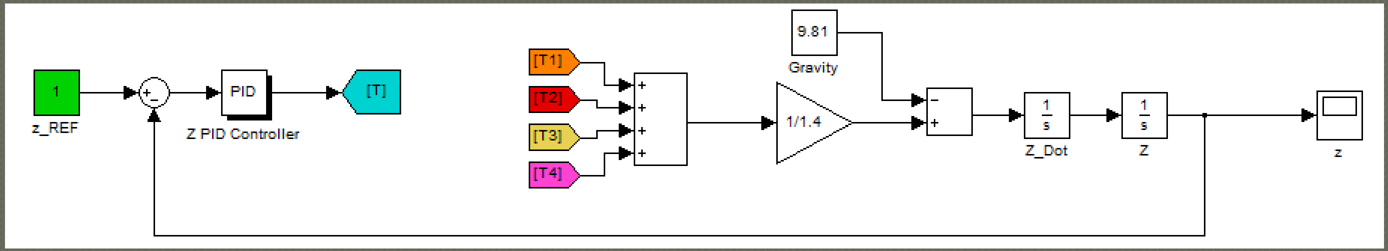
PID Controller

- In this section some assumptions are made to make the equations of motion of the plant (Q-Ball) simpler.
- This simplification let us neglect some cross-couplings effects among the equations of motion describing dynamics of the system.
- This way, the motion of the system is broken down into four independent channels:
- Vertical Motion along the Z Axis Forwards and
- Backwards Motion along the X Axis Coupled with Pitching Motion
- Side Motion along the Y Axis Coupled with Rolling Motion And
- Pure Yawing Motion

PID Controller : Decoupled, Simplified Equation of Motion

- Vertical Motion along the Z Axis

$$M\ddot{Z} = (T1 + T2 + T3 + T4) - Mg$$



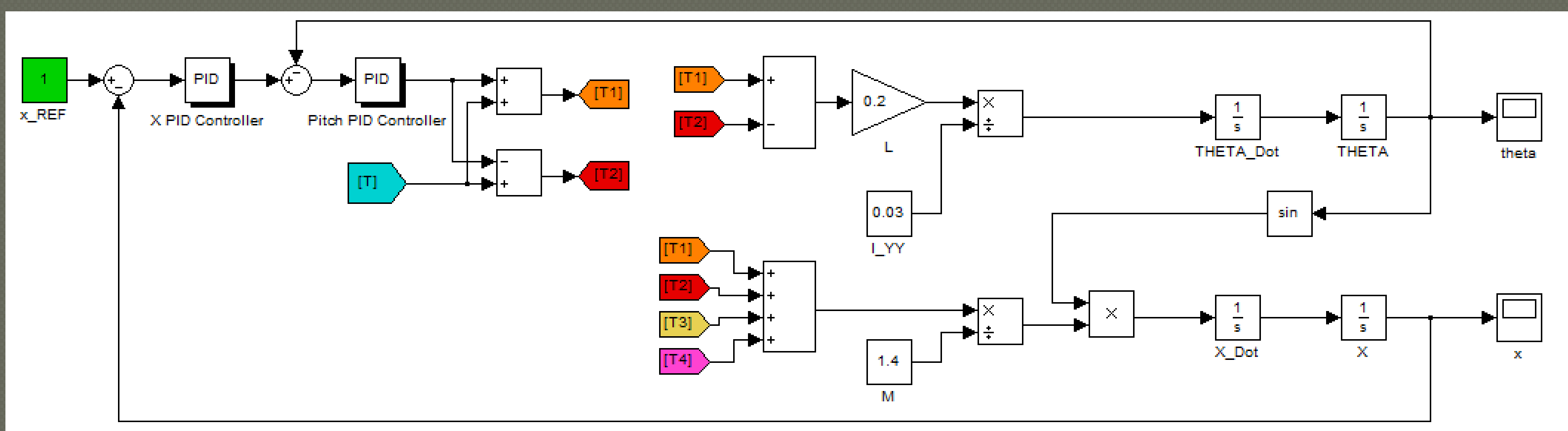
It should be notified that this modeling is valid as long as the Yaw Angle is automatically controlled to be zero.

PID Controller

- Forwards and Backwards Motion along the X Axis Coupled with Pitching Motion

$$M\ddot{X} = (T1 + T2 + T3 + T4)\sin\theta$$

$$J_{yy}\ddot{\theta} = (T1 - T2)L$$



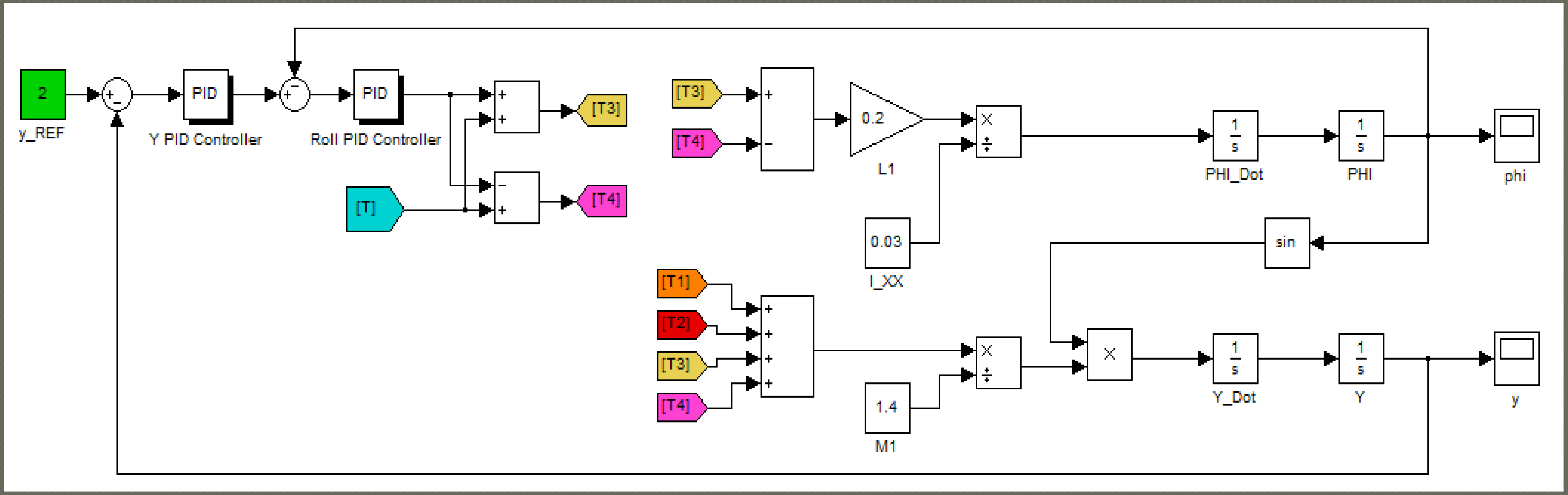
[(T1-ΔT) + (T2+ΔT) + T3 + T4] = [T1 + T2 + T3 + T4]

PID Controller

- **Side Motion along the Y Axis Coupled with Rolling Motion**

$$M\ddot{Y} = (T1 + T2 + T3 + T4)\varphi$$

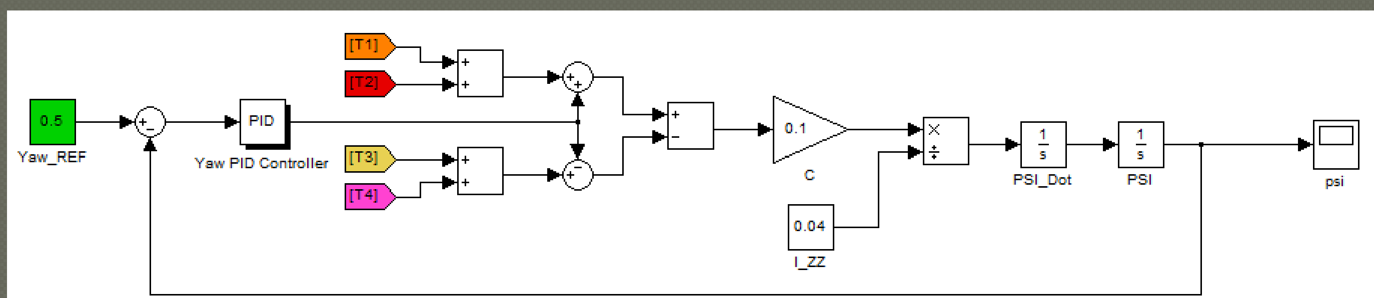
$$J_{xx}\ddot{\varphi} = (T3 - T4)L$$



PID Controller

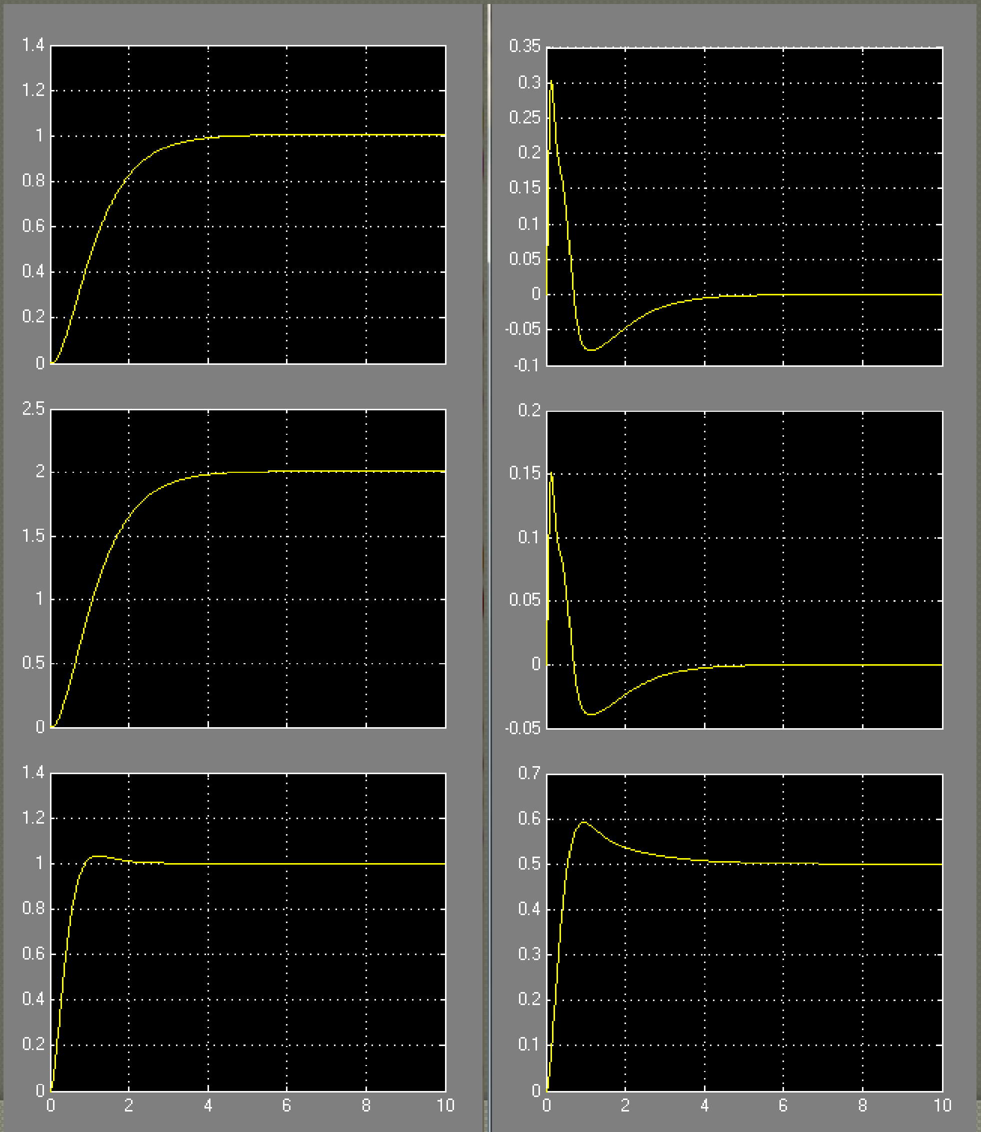
- **Pure Yawing Motion**

$$J_{zz}\ddot{\psi} = (T1 + T2 - T3 - T4)C$$

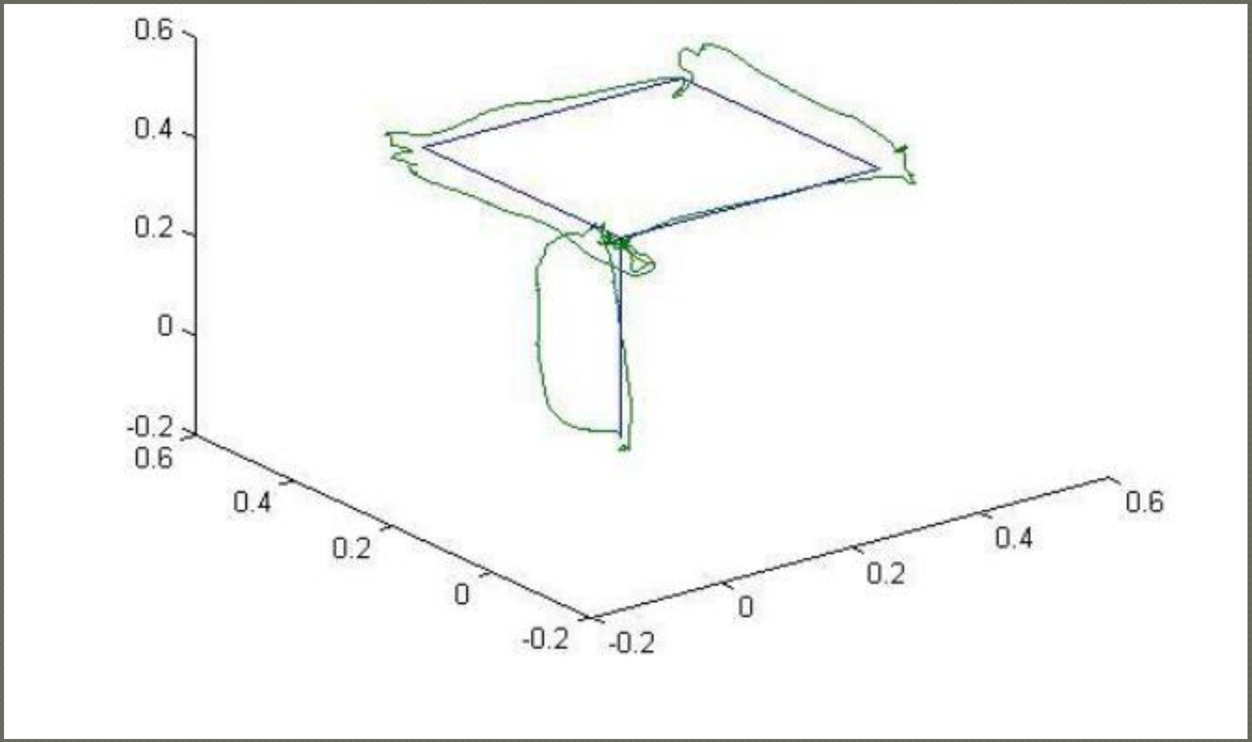


- ***A Remark on PID Tuning:*** Tuning of the inner loop PID Controller prior to tuning of the outer loop PID Controller is required for the sake of fine and effective tuning.

PID Controller :Results



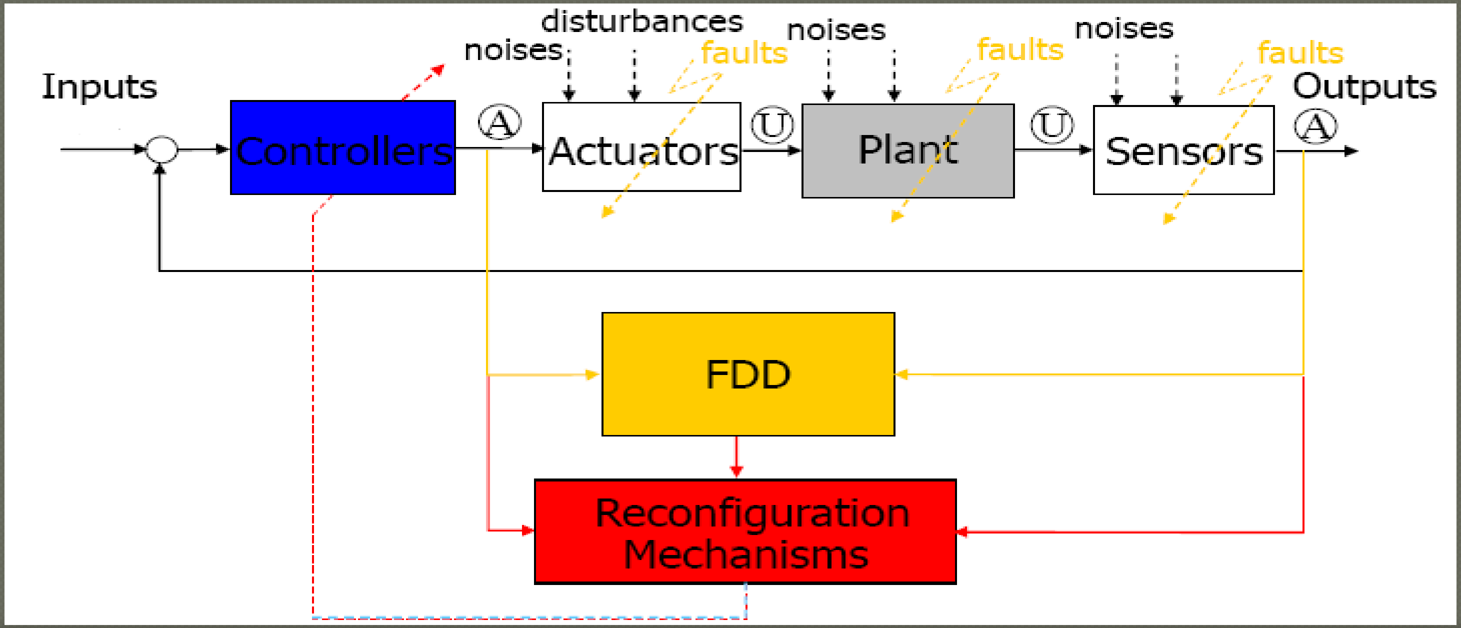
PID- Real System Results (Fault-Free)

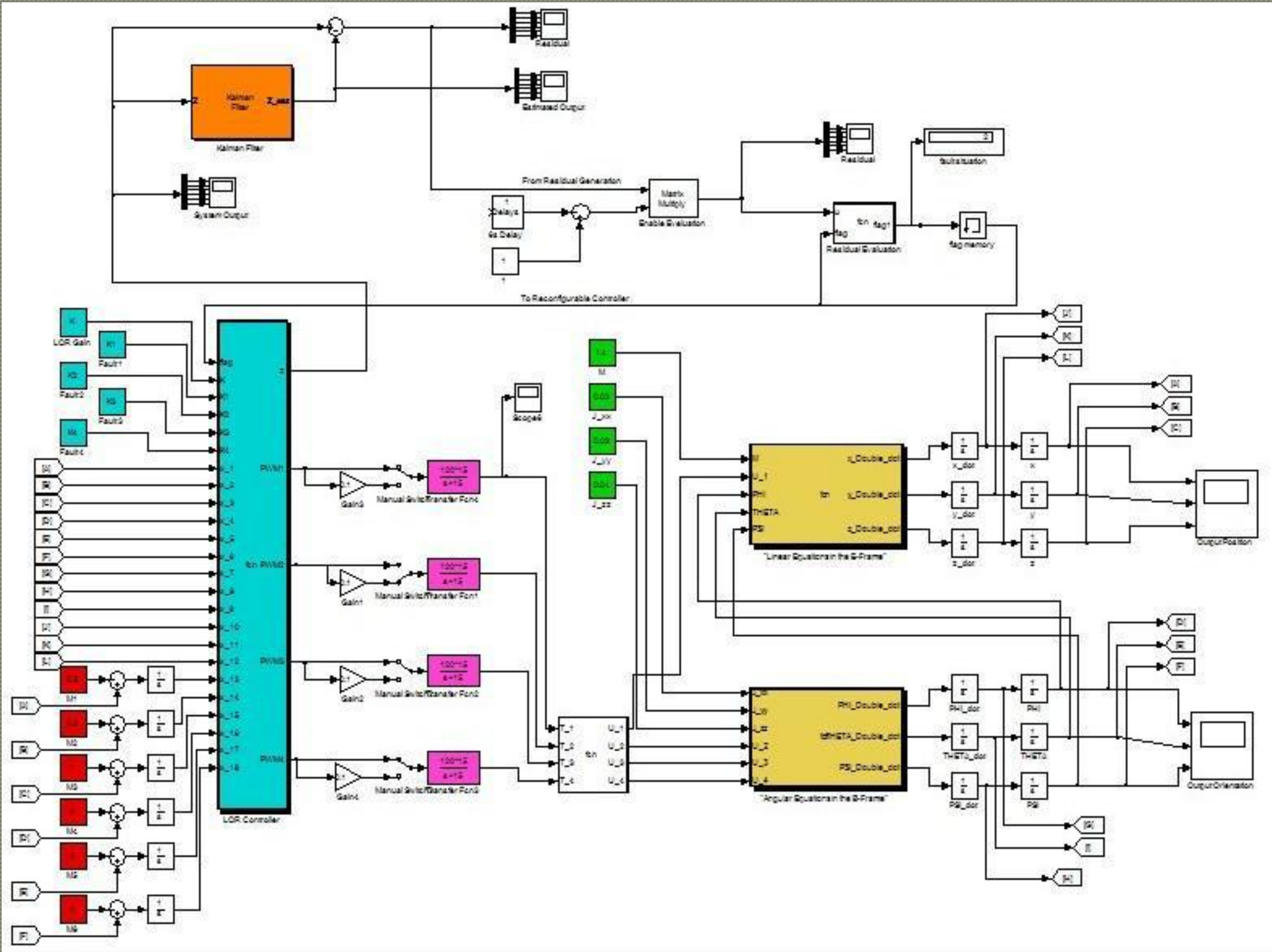


AFTCS Concepts

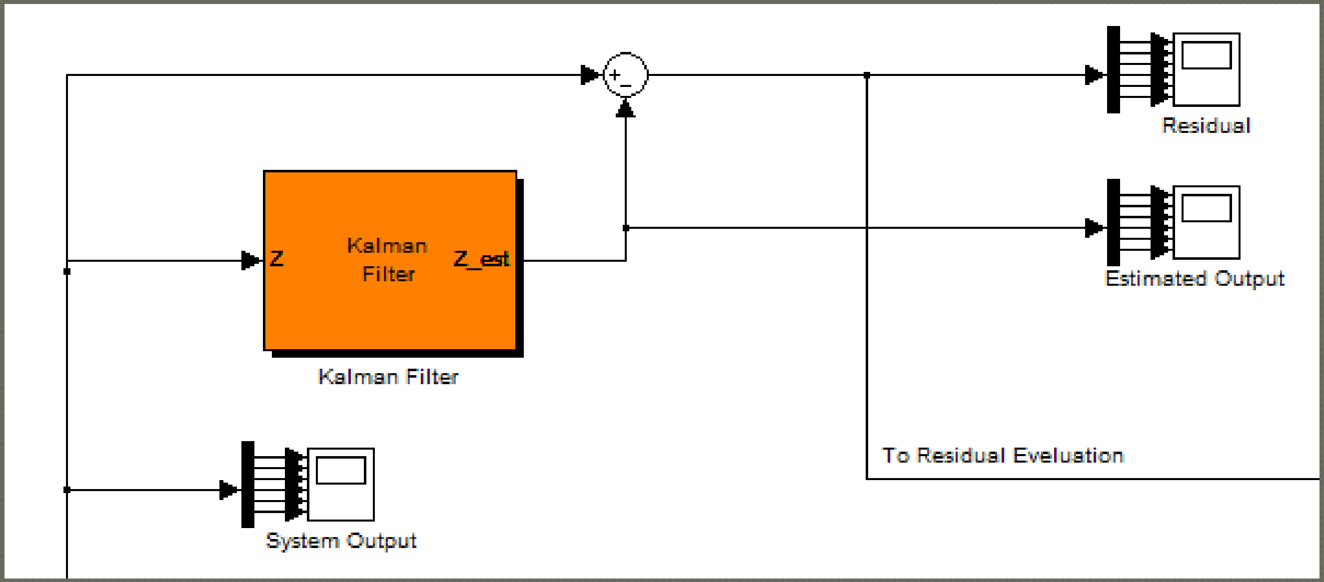
- ***AFTCS*** is combined by three parts:

1. FDD part
2. Reconfiguration Mechanism
3. Reconfigurable Controller.



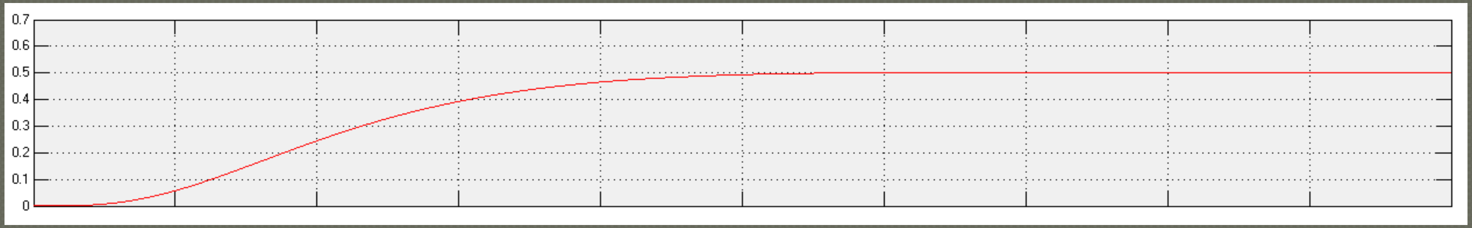

Residual
Estimated Output
Kalman Filter
System Output
From Residual Generation
Matrix Multiply
Enable Evaluation
Residual Evaluation
flag memory
To Reconfigurable Controller
LQR Gain
LQR Controller
Scope6
"Linear Equations in the E-Frame"
"Angular Equations in the B-Frame"
Output Position
Output Orientation

LQR-FDD

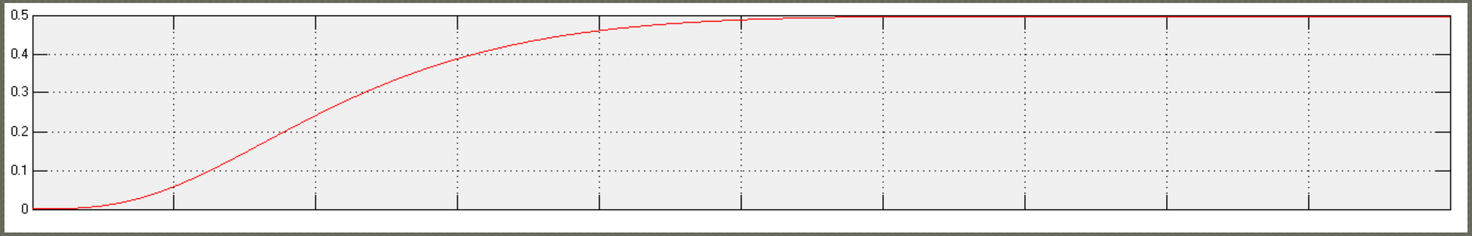


- **General Frame of Residual Generation**

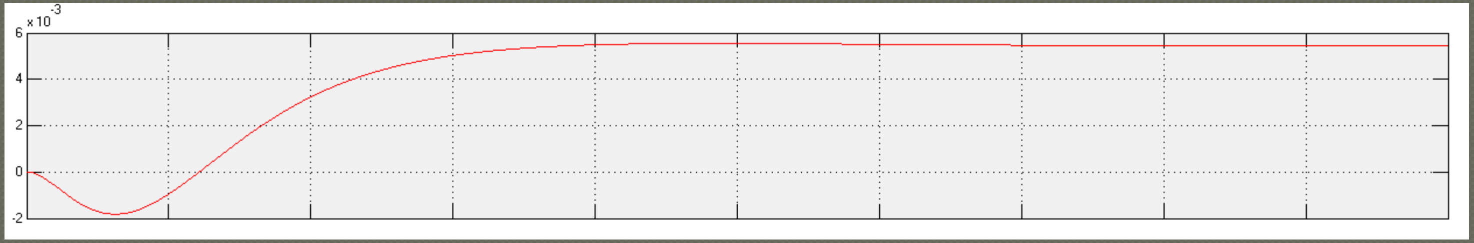
Residual Generation



System output

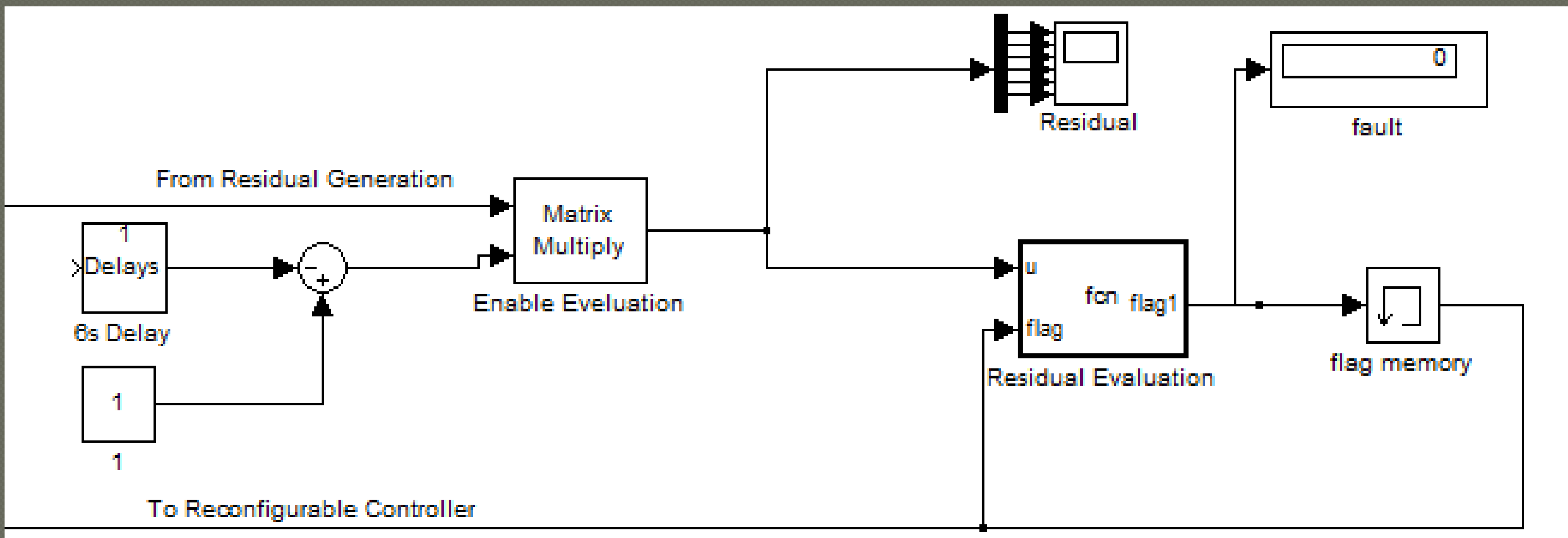


Estimated output

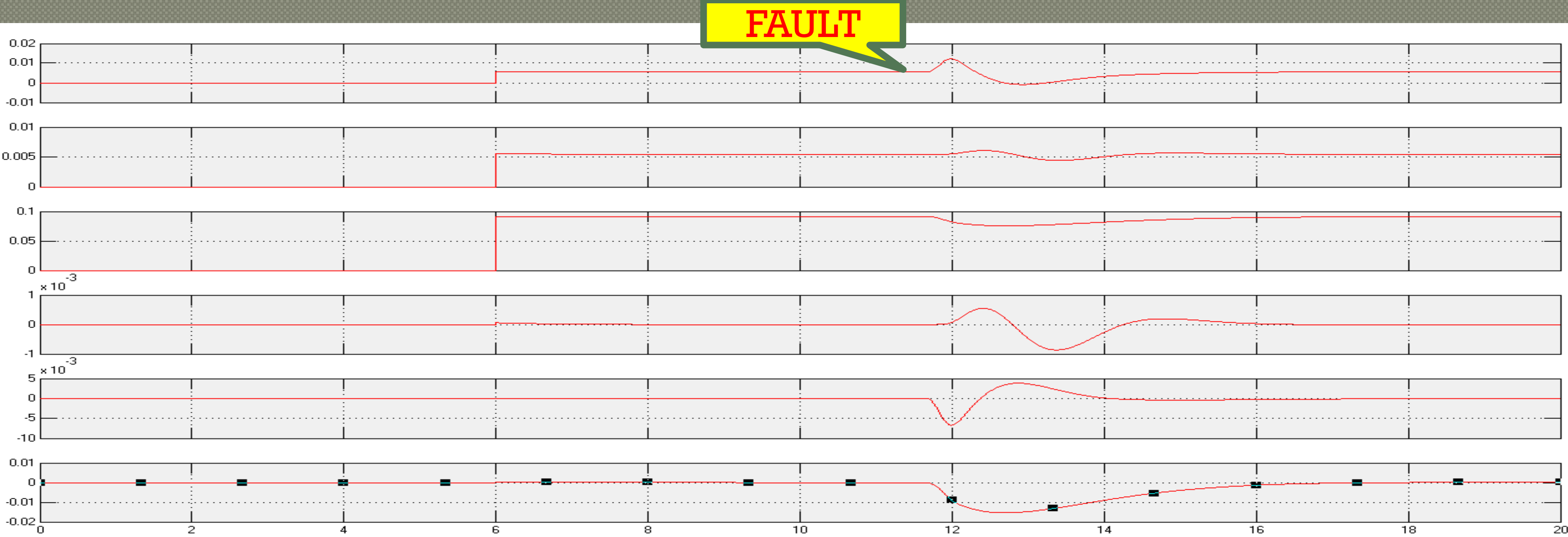


Residual (Fault-Free)

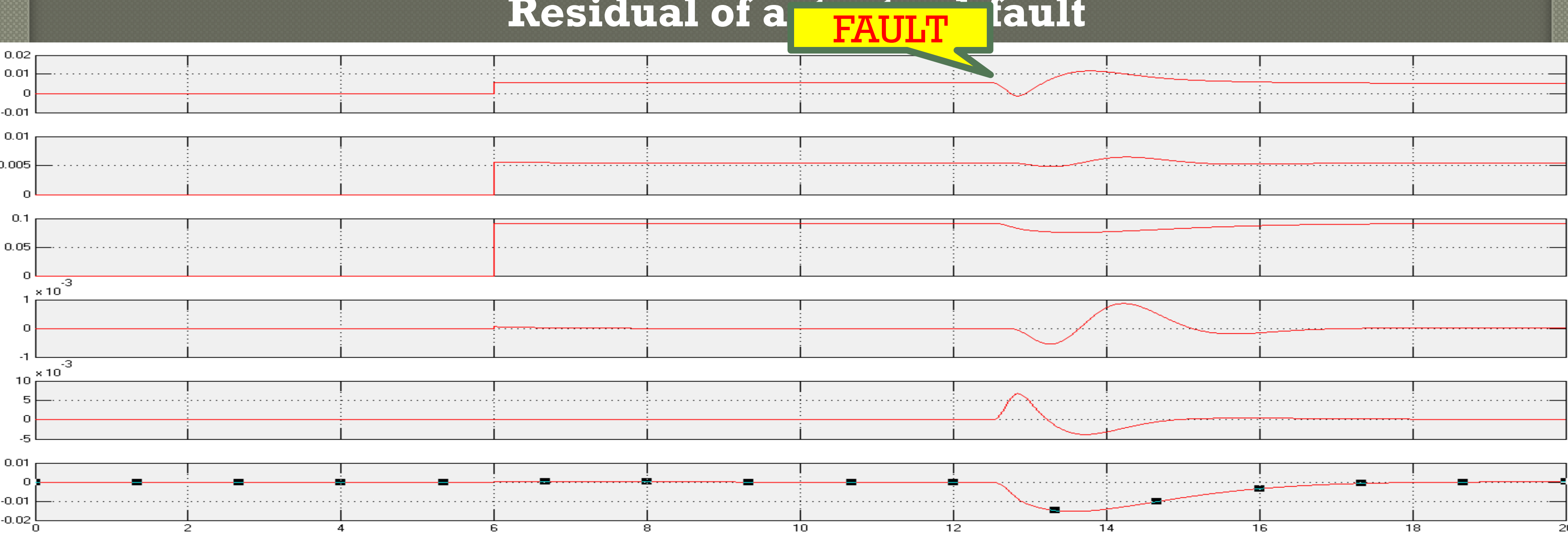
Residual Evaluation Part



Residual Evaluation and Reconfigurable Mechanism

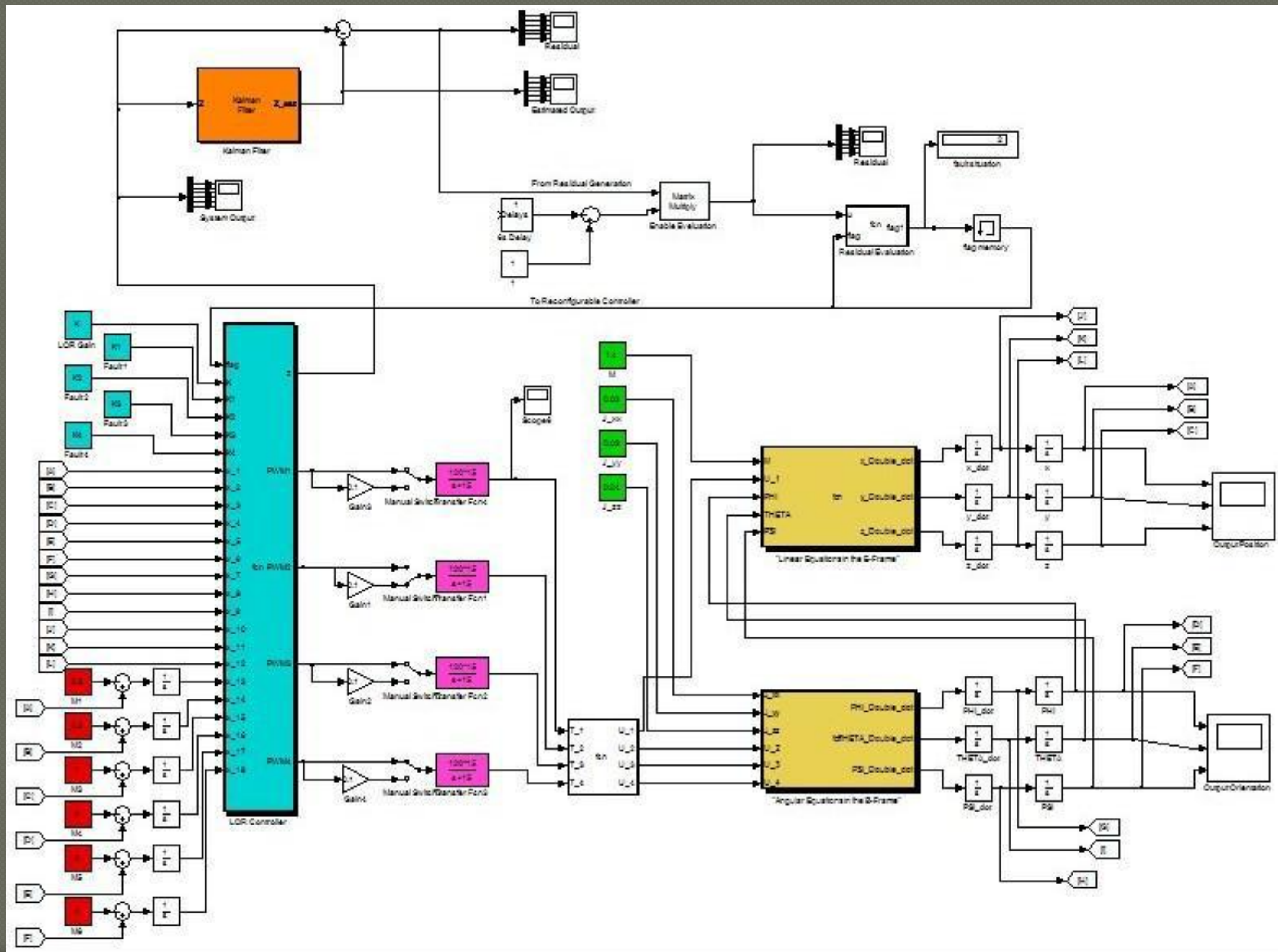


Residual of a[...] fault



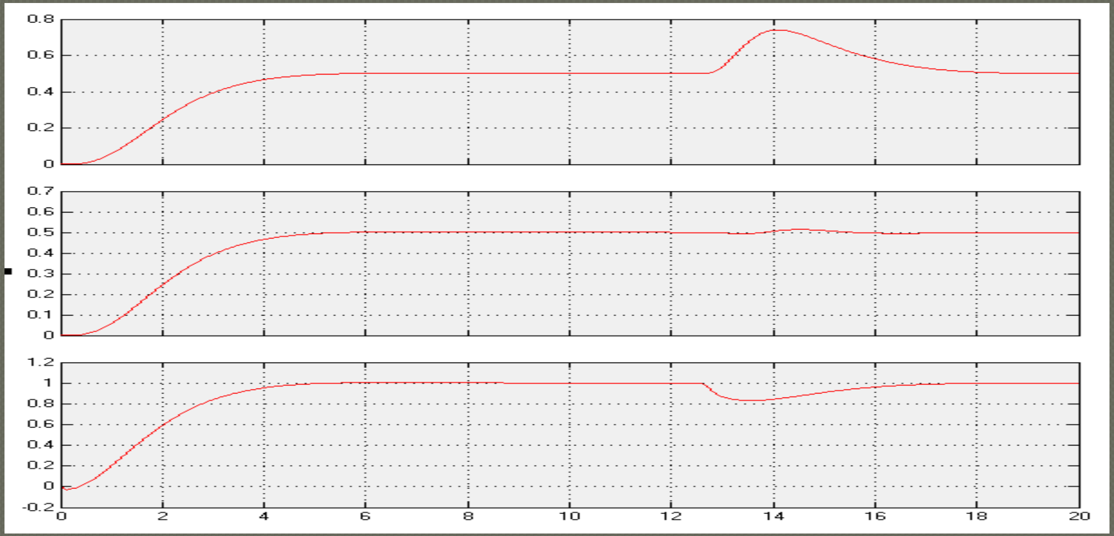
Residual of actuator 2 fault

LQR-Reconfigurable Controller

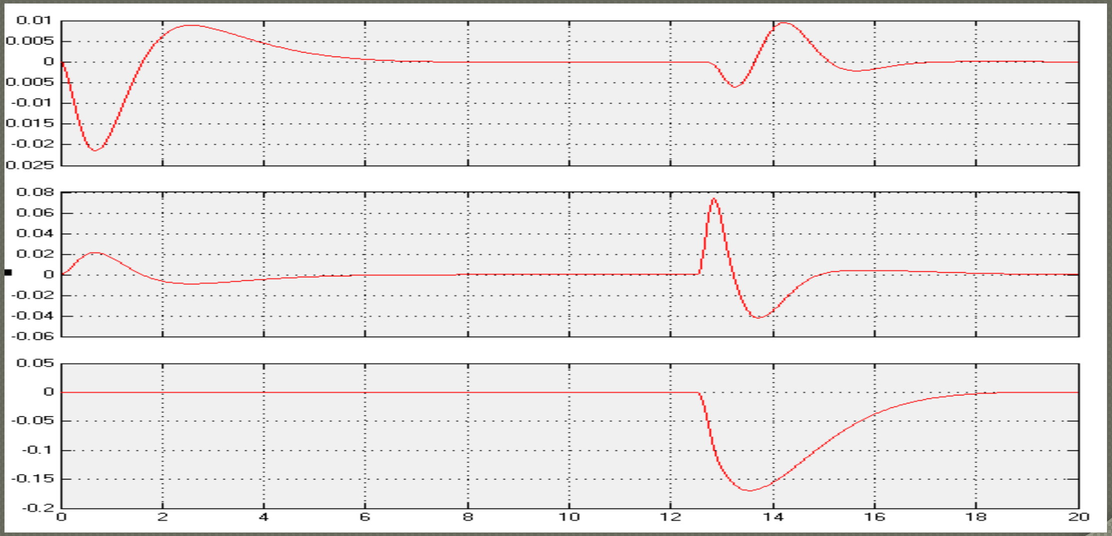


LQR AFTCS results

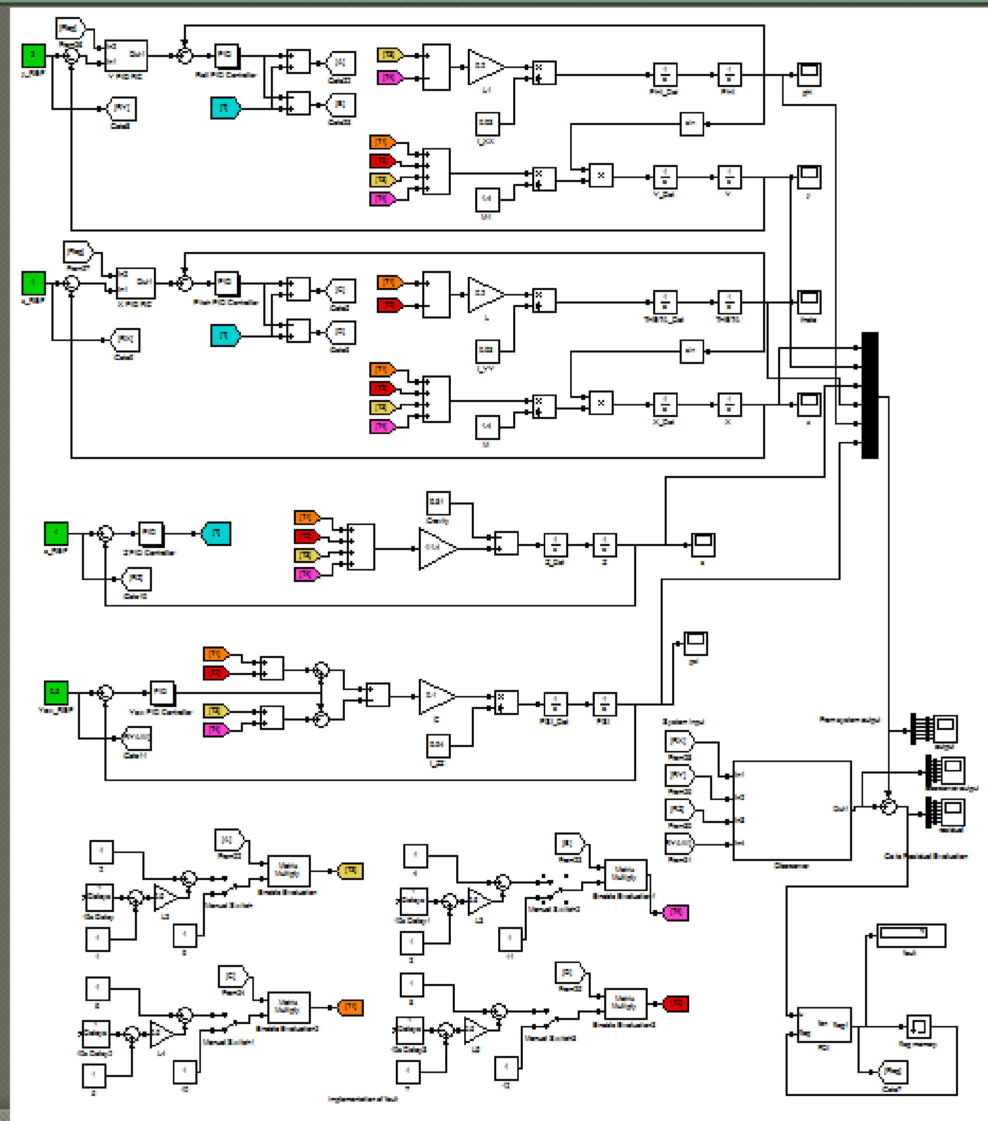
Output of actuator 2 fault (x, y, z)



Output of actuator 2 fault (φ, θ, Ψ)

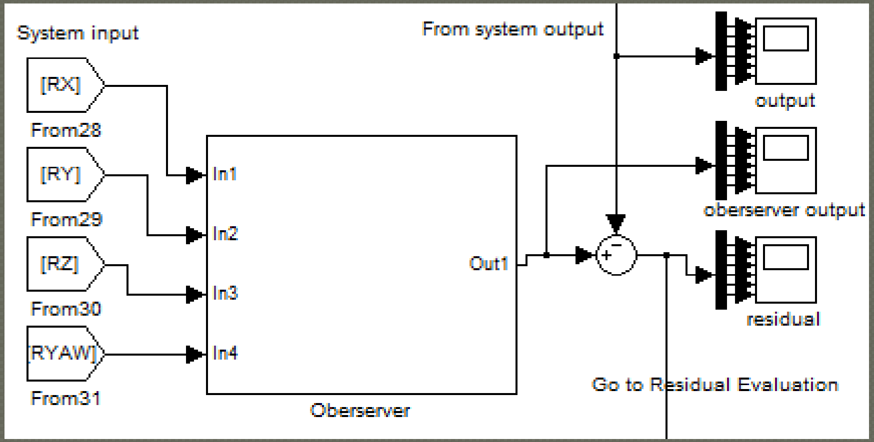


PID AFTCS simulink block

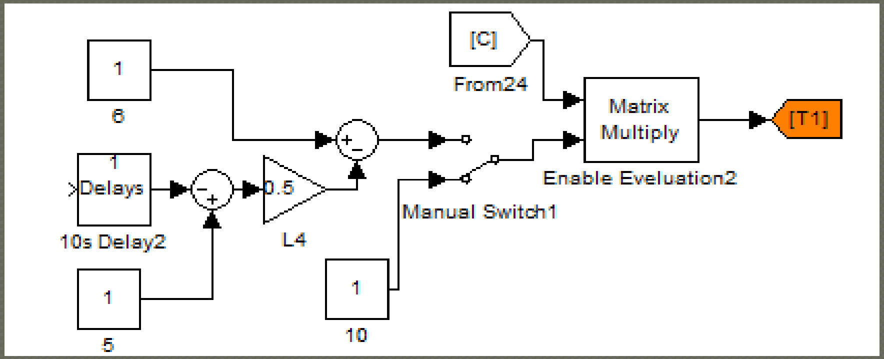


PID - FDD

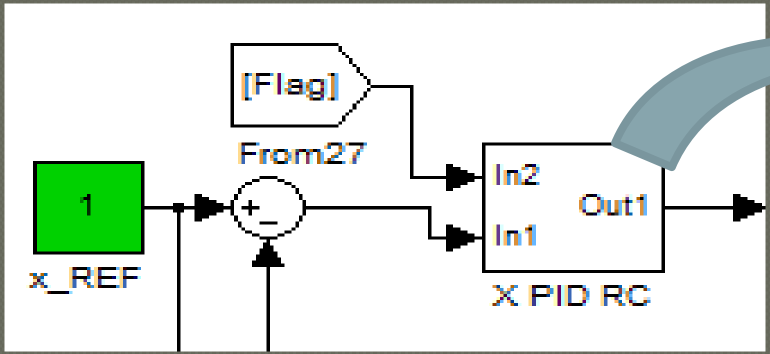
General frame of Analytical Redundancy

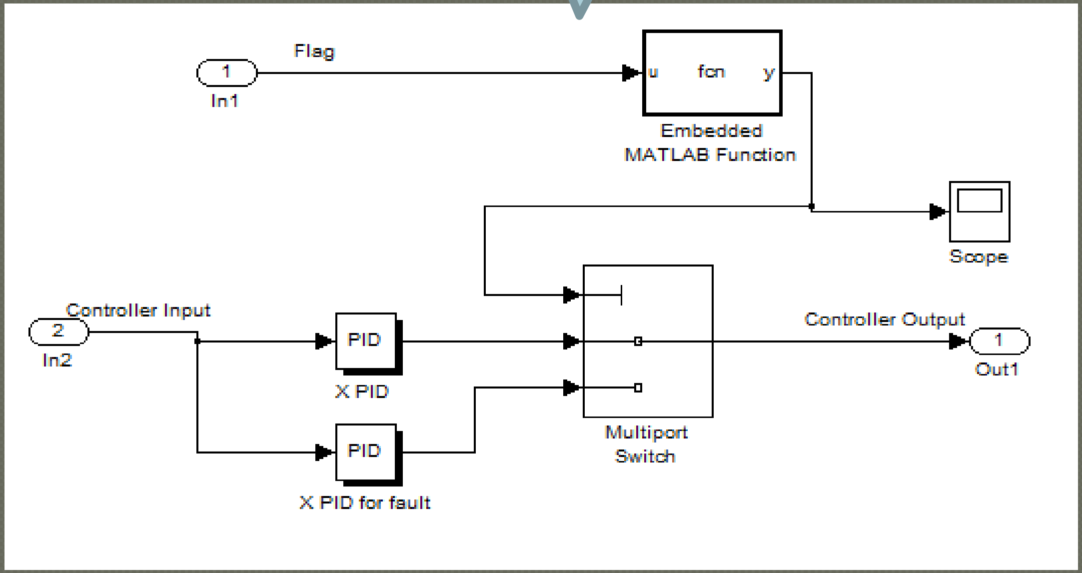


Fault Implementation



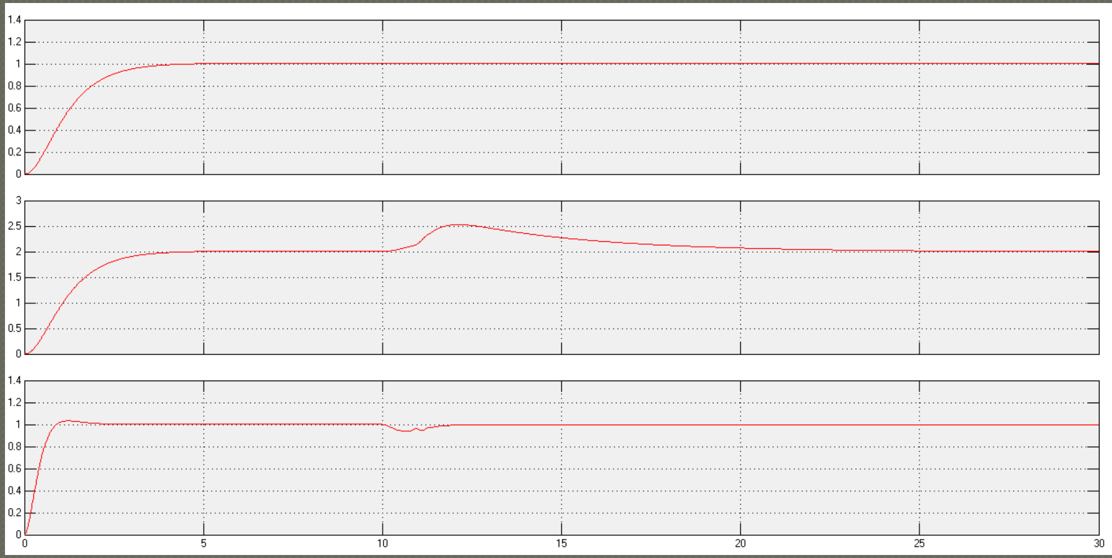
PID-Reconfigurable Controller



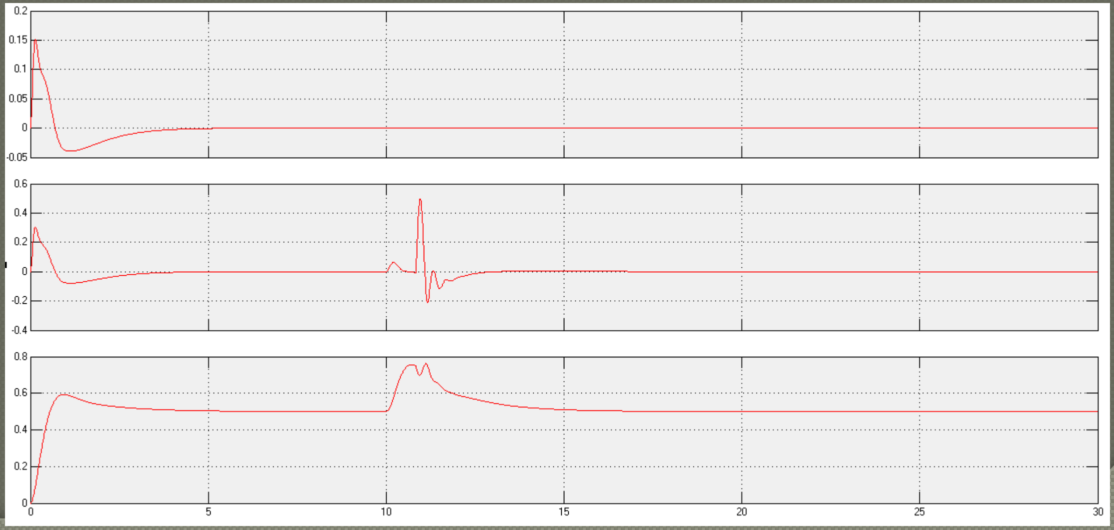


PID AFTCS results

Output of actuator 4 fault (x, y, z)



Output of actuator 4 fault (φ, θ, Ψ)



Conclusion

- Two AFTCS have been designed for the Q-Ball to rectify performance of the system and maintain stability in the presence of actuator partial loss.
- Only actuator faults have been considered, however the system could be extended to deal with sensor and system component faults.
- Implementation on the Real system.

Thank you

Questions !