UNIVERSITÉ Concordia UNIVERSITY

FDD AND AFTCS OF QUAD-ROTOR UAV

Professor: Dr. Youmin Zhang

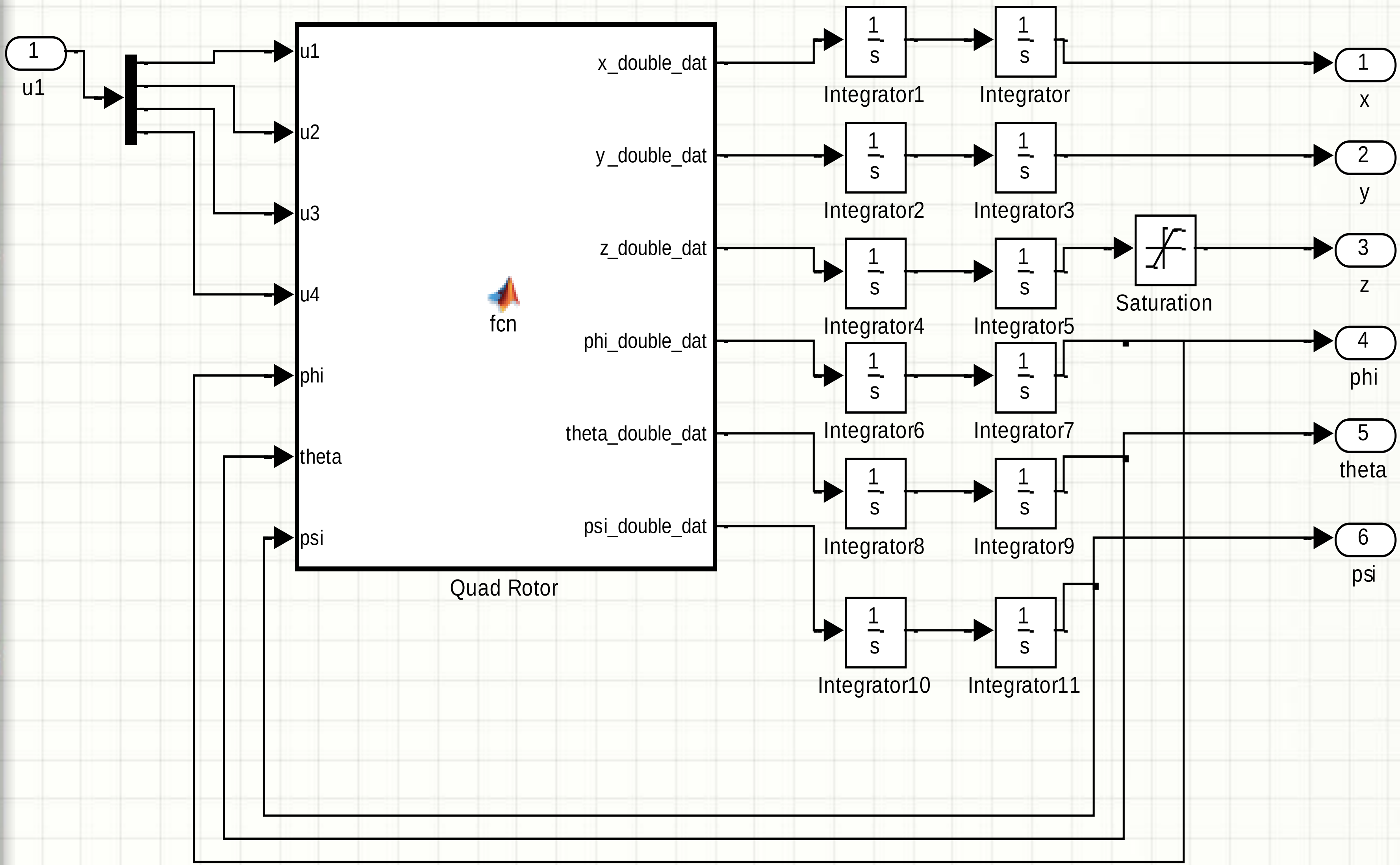
Preeti Dhanapalan, Saba Kiakojoori, Azita Malek

Fall 2011

Contents

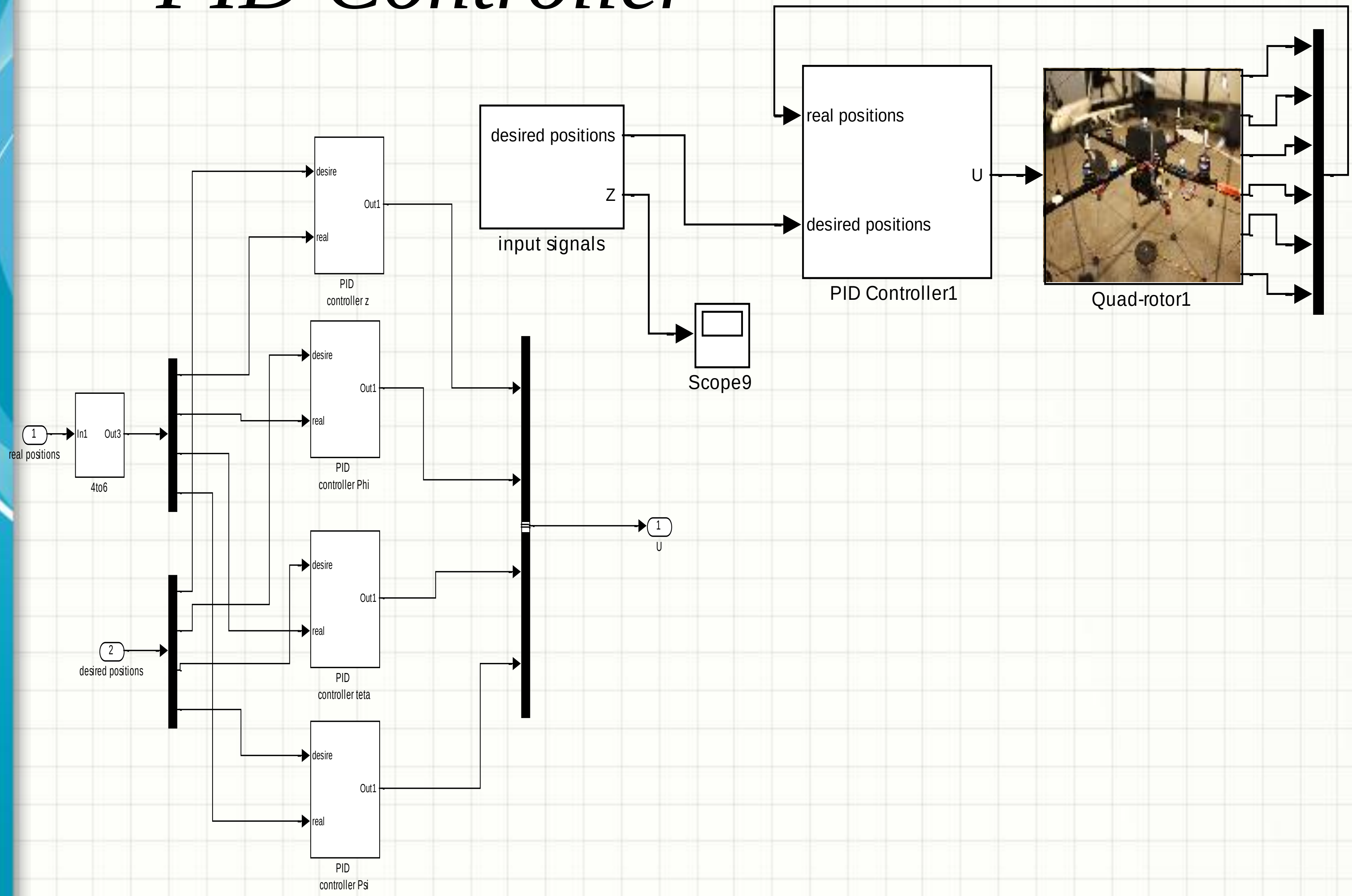
Quad-rotor Modeling

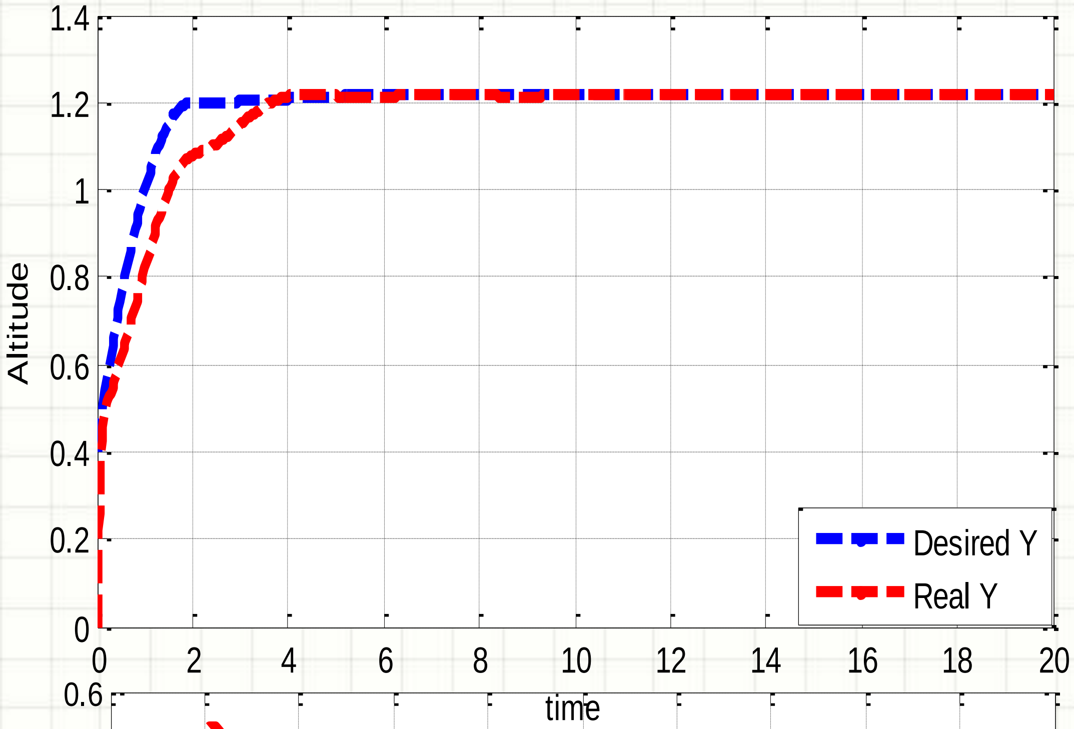
- $\ddot{X} = u_1(cos\emptyset sin\theta cos\psi + sin\emptyset sin\psi)$
- $\ddot{Y} = u_1(cos\emptyset sin\theta sin\psi - sin\emptyset cos\psi)$
- $\ddot{Z} = u_1(cos\emptyset cos\theta) - g + g_r(z)$
- $\ddot{\emptyset} = u_2 l$
- $\ddot{\theta} = u_3 l$
- $\ddot{\psi} = u_4$

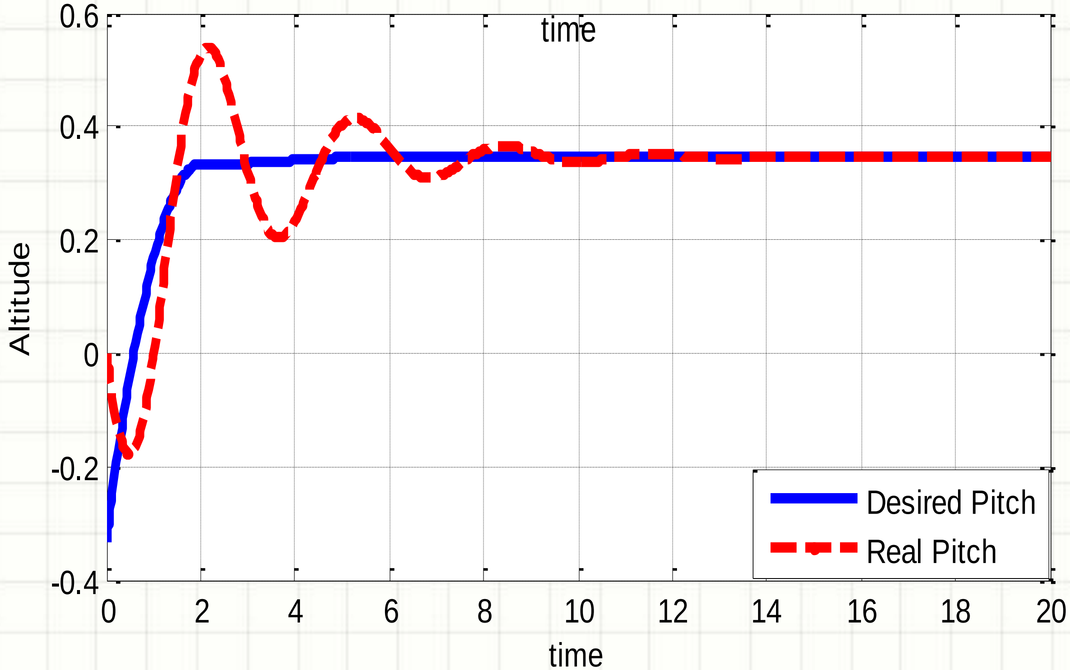


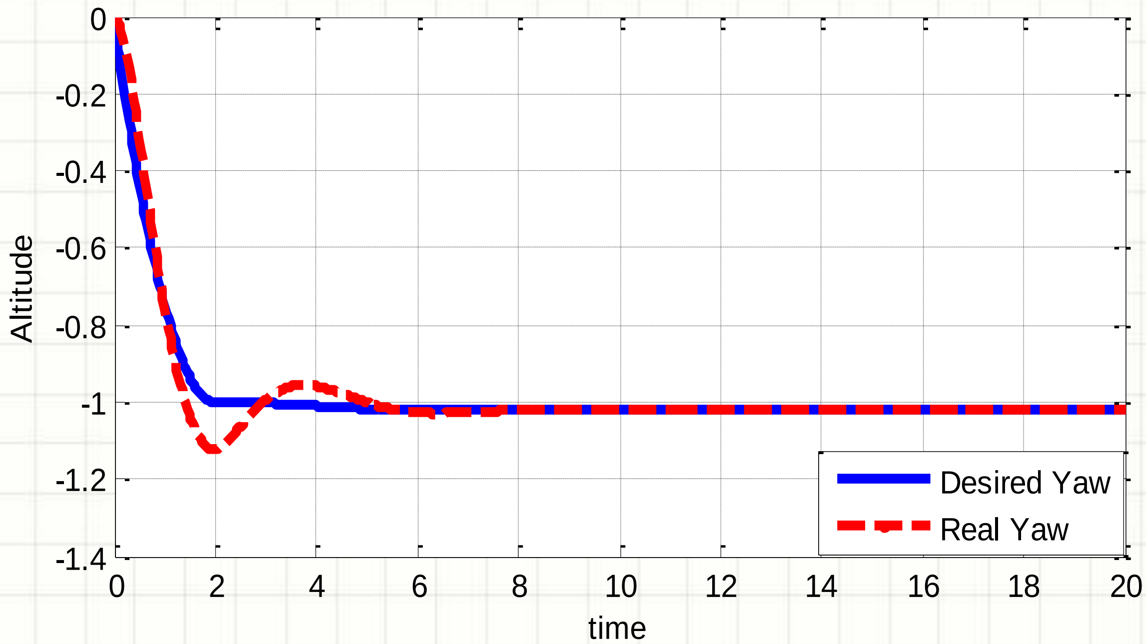
Model of Quad-rotor in Simulink

PID Controller

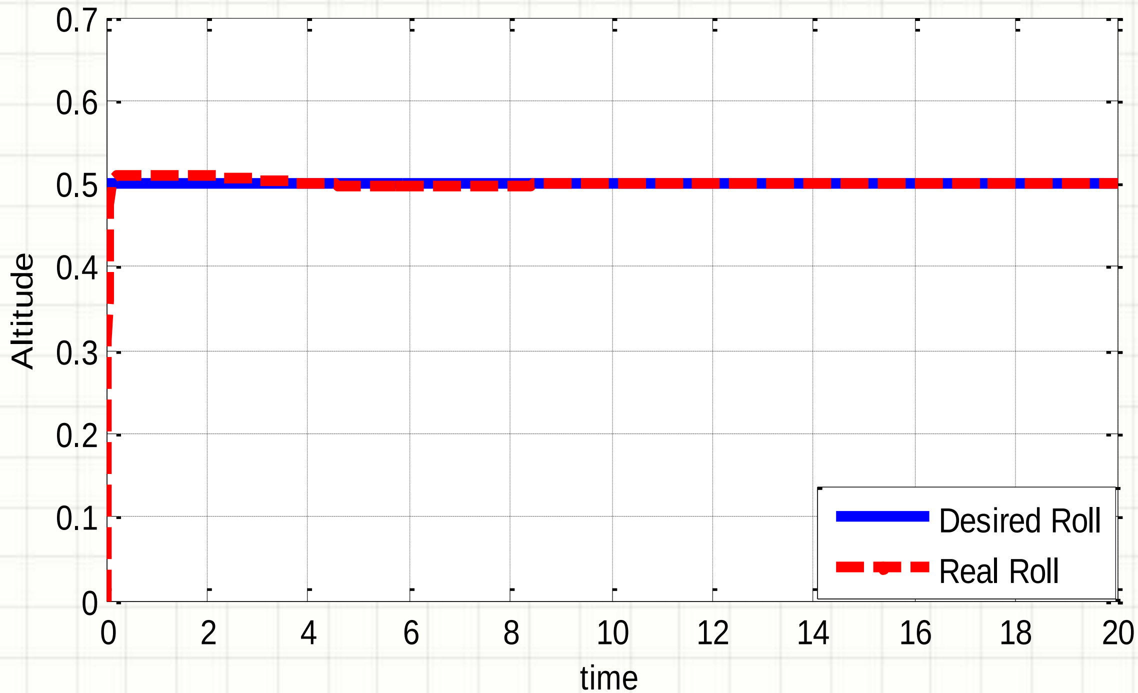



1.4
1.2
1
0.8
0.6
0.4
0.2
0
Altitude
0 2 4 6 8 10 12 14 16 18 20
time
Desired Y
Real Y

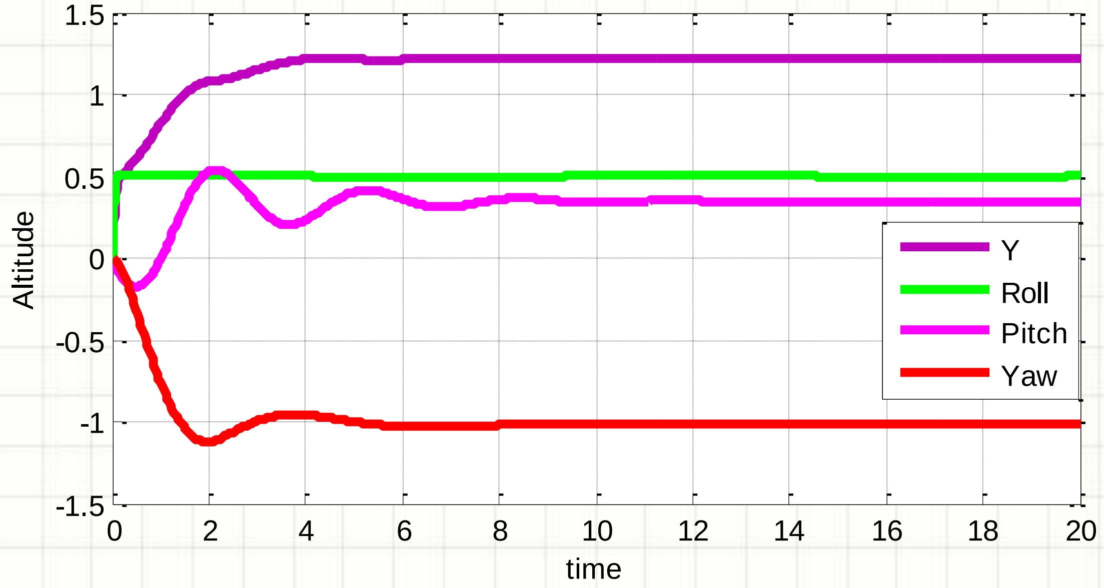

0.6
0.4
0.2
0
-0.2
-0.4
Altitude
0 2 4 6 8 10 12 14 16 18 20
time
Desired Pitch
Real Pitch



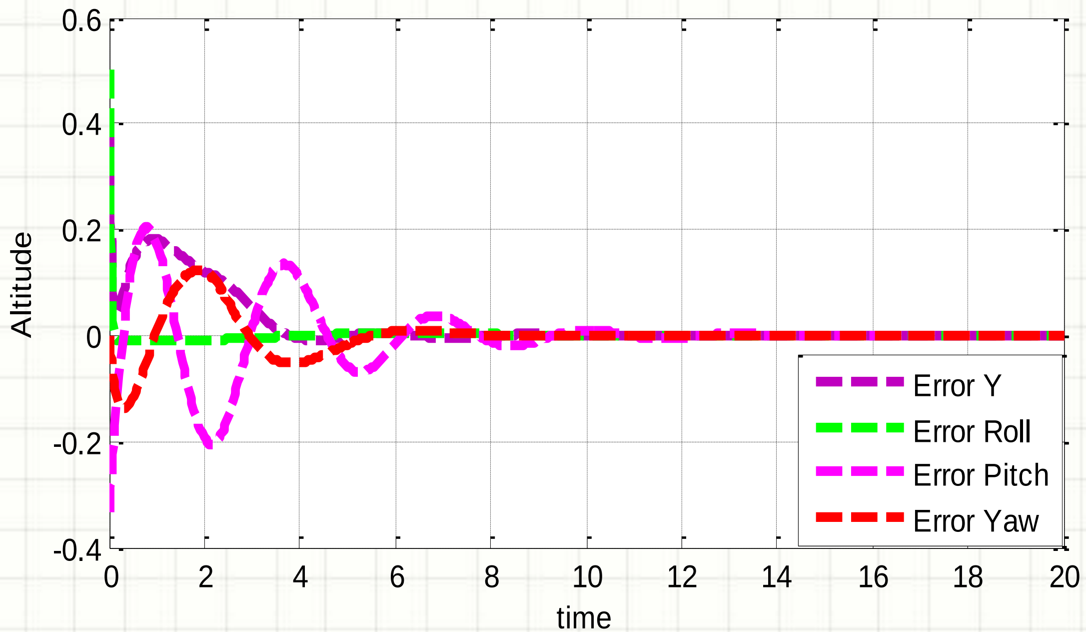
Altitude
time
Desired Yaw
Real Yaw



Altitude
time
Desired Roll
Real Roll

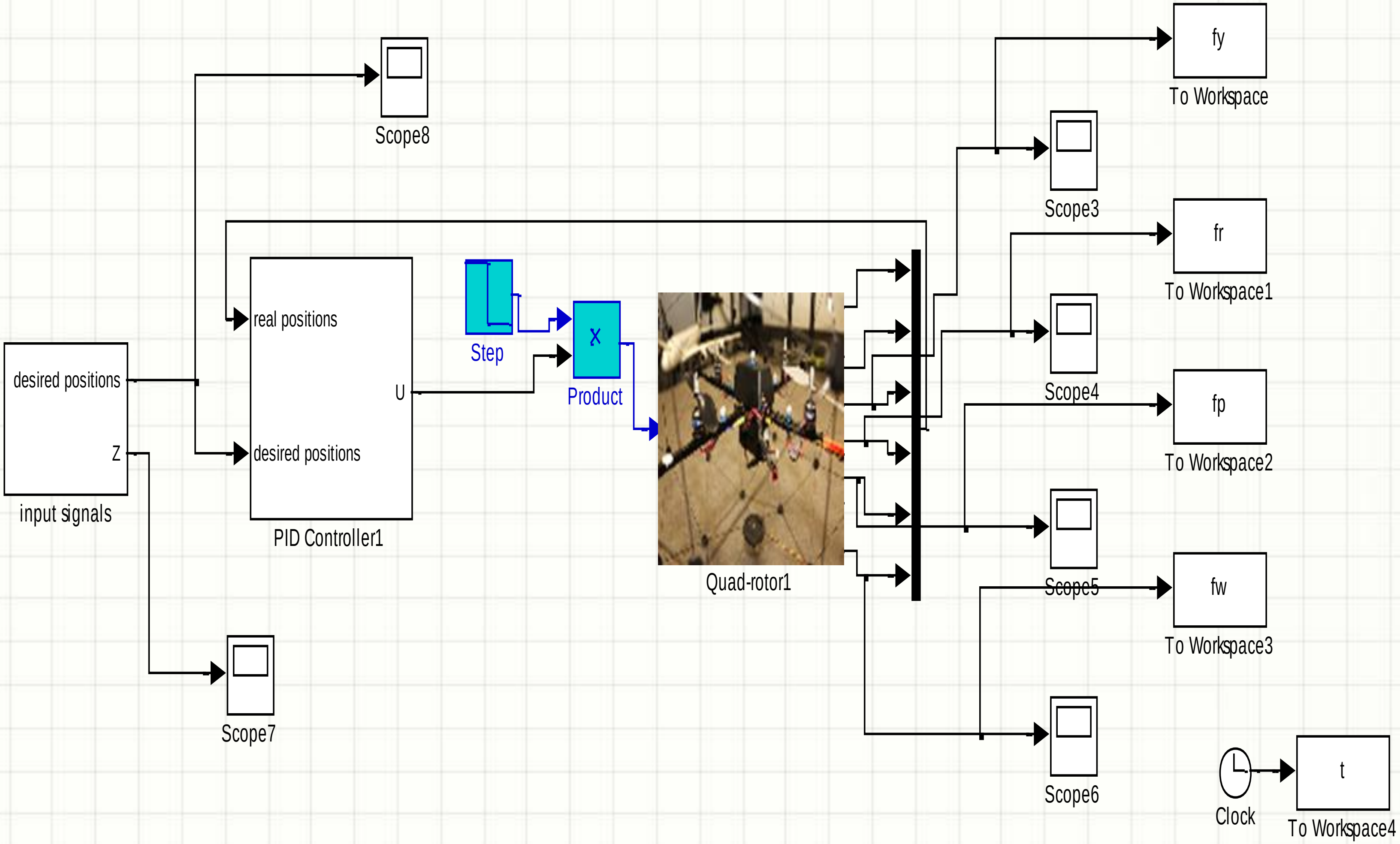


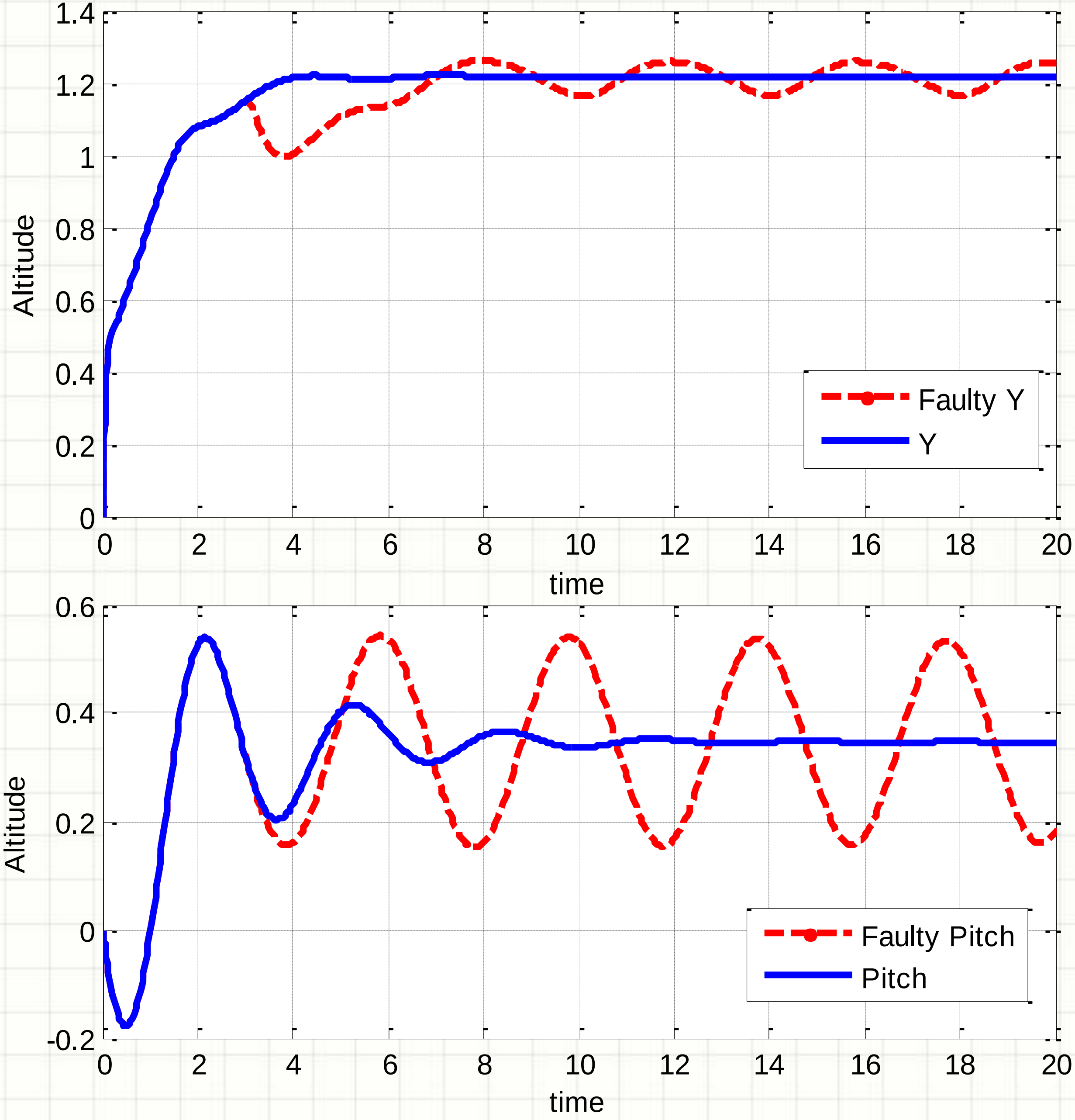
1.5
1
0.5
0
-0.5
-1
-1.5
Altitude
0
2
4
6
8
10
12
14
16
18
20
time
Y
Roll
Pitch
Yaw



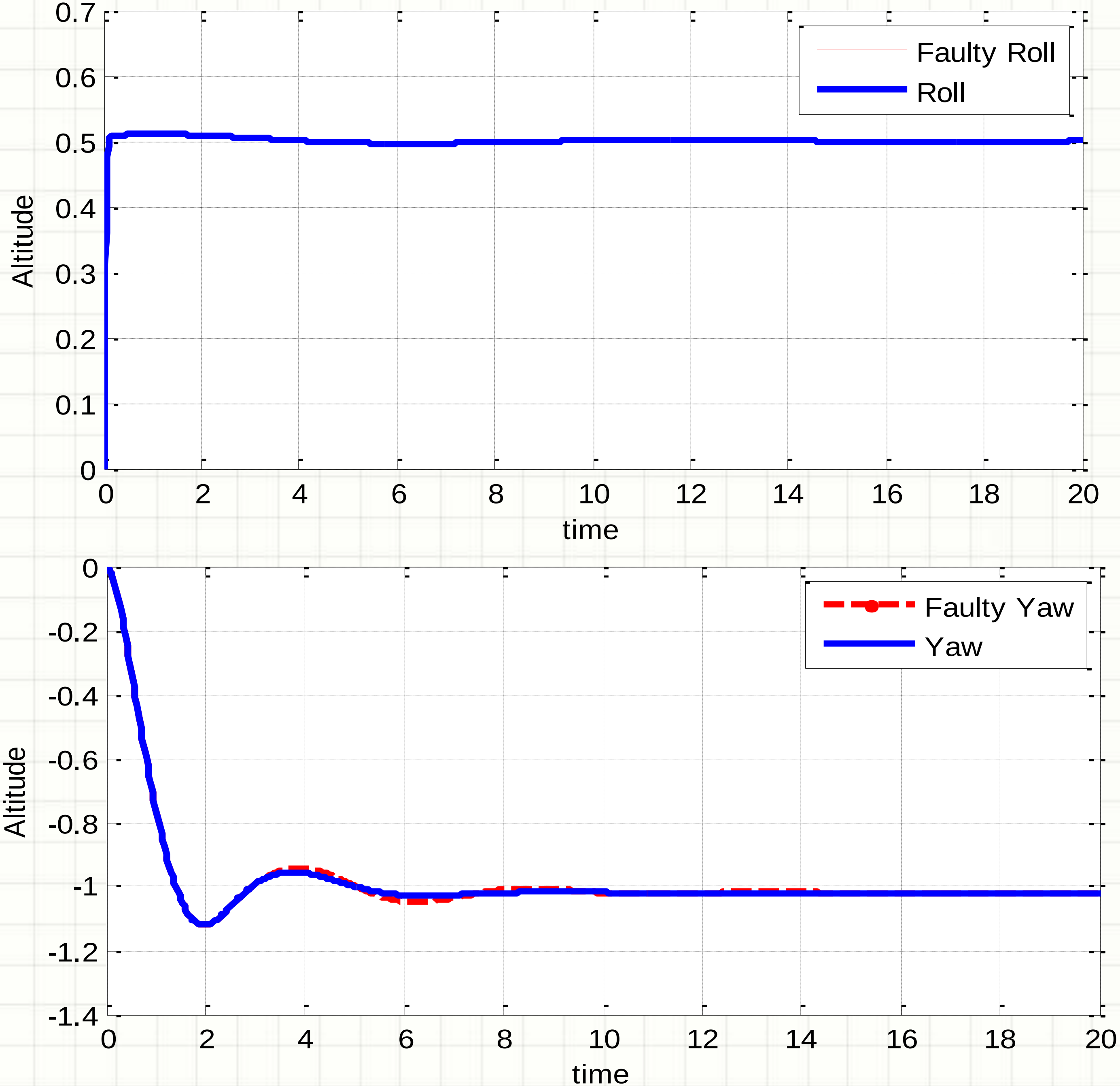
0.6
0.4
0.2
0
-0.2
-0.4
Altitude
0
2
4
6
8
10
12
14
16
18
20
time
Error Y
Error Roll
Error Pitch
Error Yaw

Fault Detection





Altitude
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
2
4
6
8
10
12
14
16
18
20
time
Faulty Y
Y
Altitude
0.6
0.4
0.2
0
-0.2
time
Faulty Pitch
Pitch


Faulty Roll
Roll
Altitude
time
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
2
4
6
8
10
12
14
16
18
20
Faulty Yaw
Yaw
Altitude
time
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
-1.4

Linear Model of the System

$$A=\begin{bmatrix}0&1&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&1\\0&0&0&0&0&0&0&0\end{bmatrix}$$

$$B=\begin{bmatrix}0&0&0&0\\g_1&0&0&0\\0&0&0&0\\0&g_2&0&0\\0&0&0&0\\0&0&g_3&0\\0&0&0&0\\0&0&0&g_4\end{bmatrix}$$

$$C=\begin{bmatrix}1&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&0&0&1&0\end{bmatrix}$$

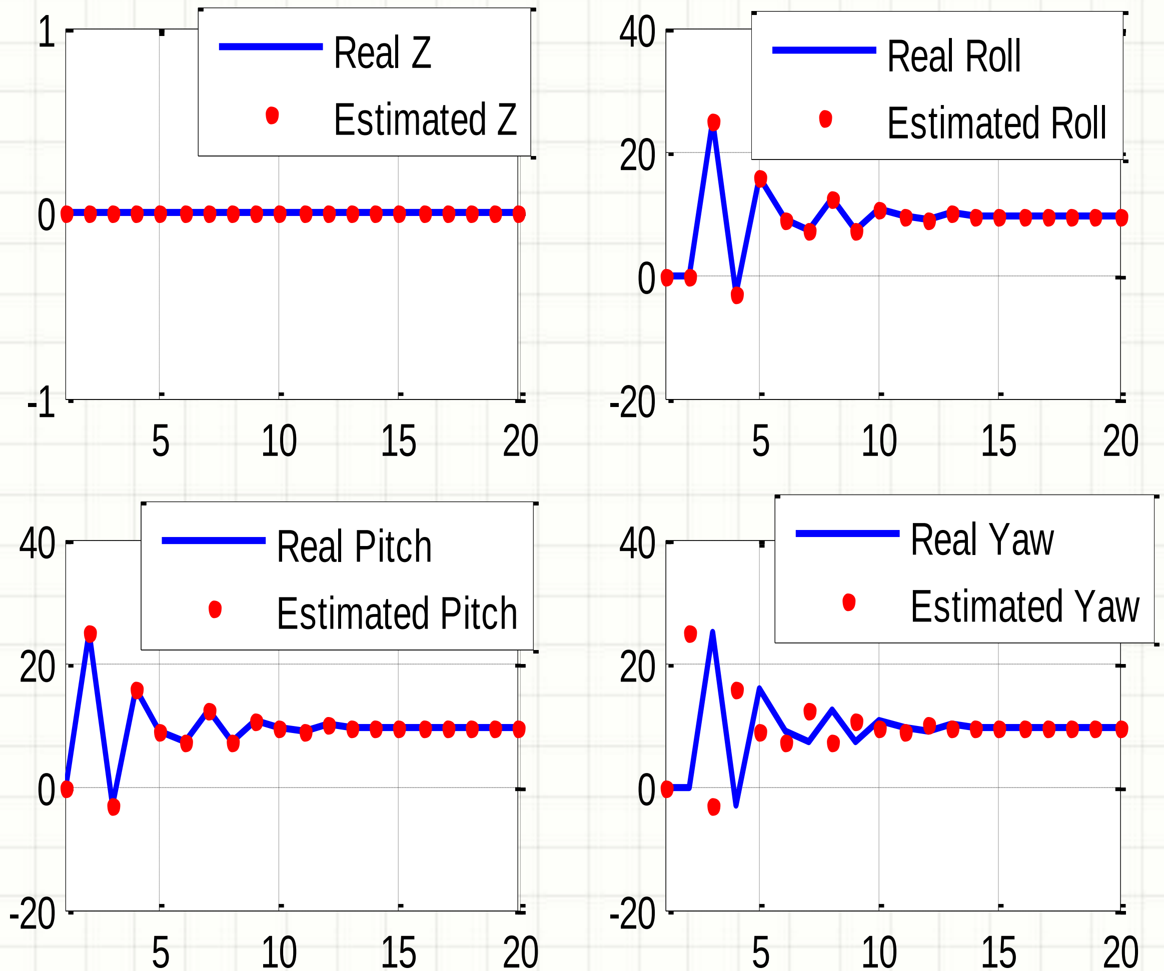
$$g_1=\frac{-2}{m}\times b_1\cos(\frac{pi}{18})\times\sin(\frac{pi}{18})$$

$$g_2=\frac{l}{J_1}\times b_2$$

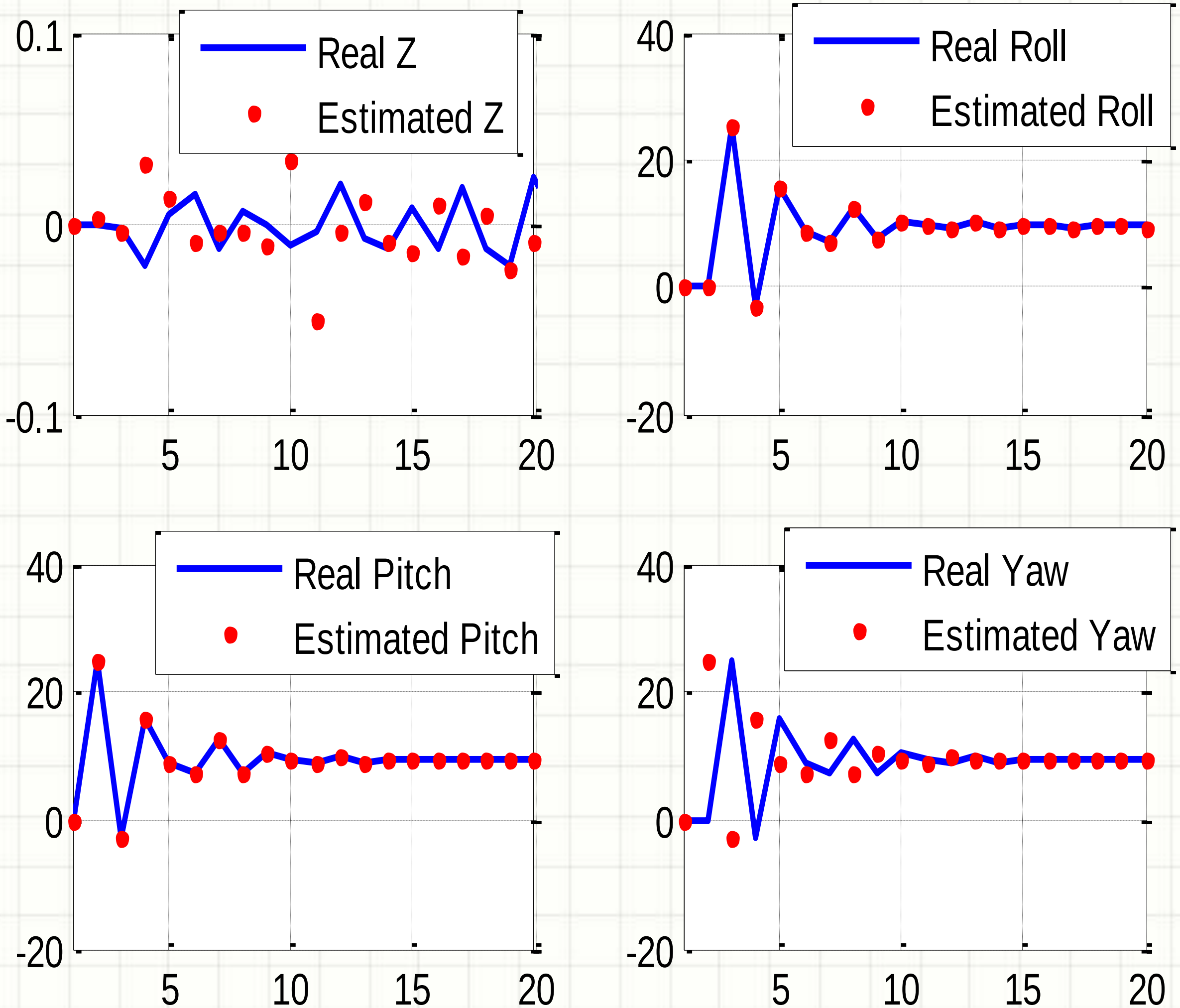
$$g_3=\frac{b_3 l}{J_2}$$

$$g_4=\frac{b_4 l}{J_3}$$

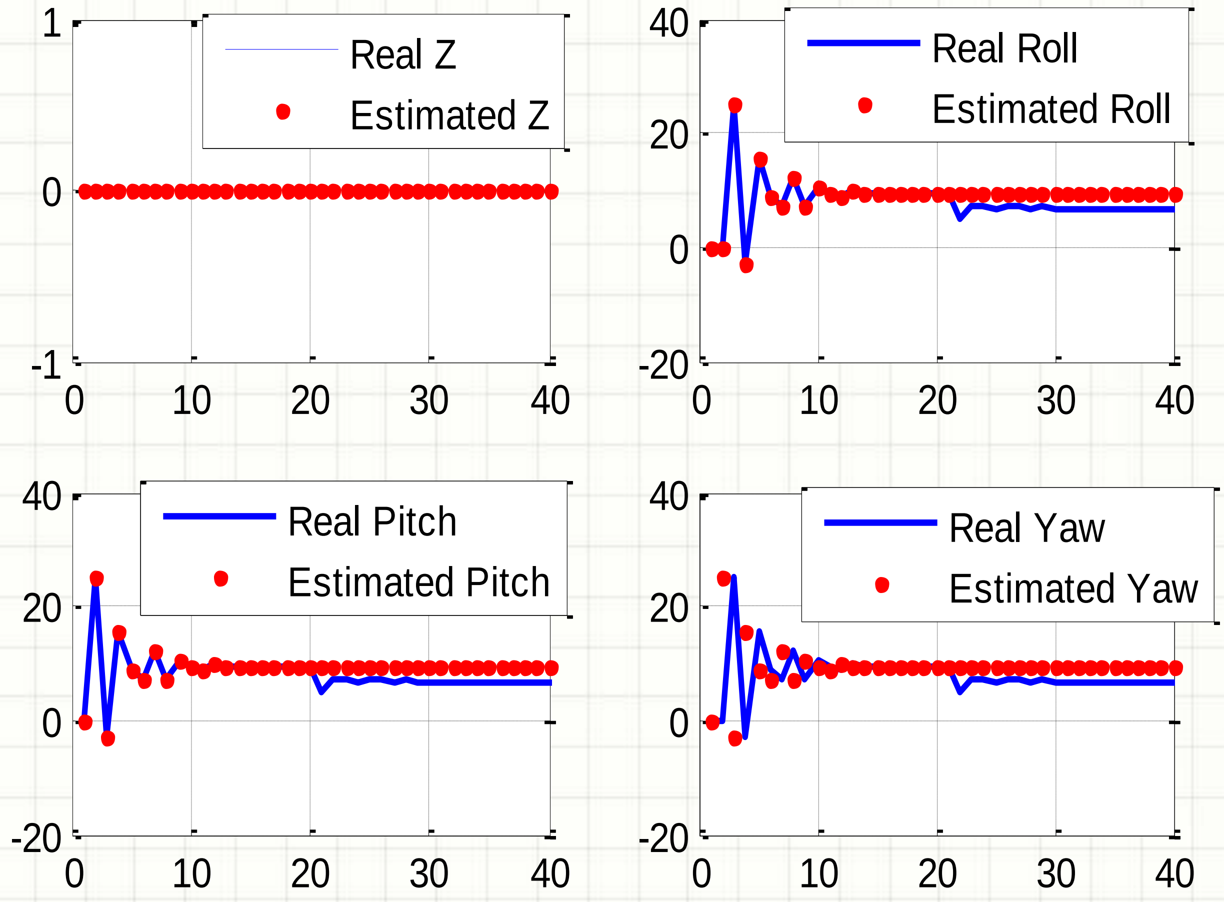
No fault, No noise

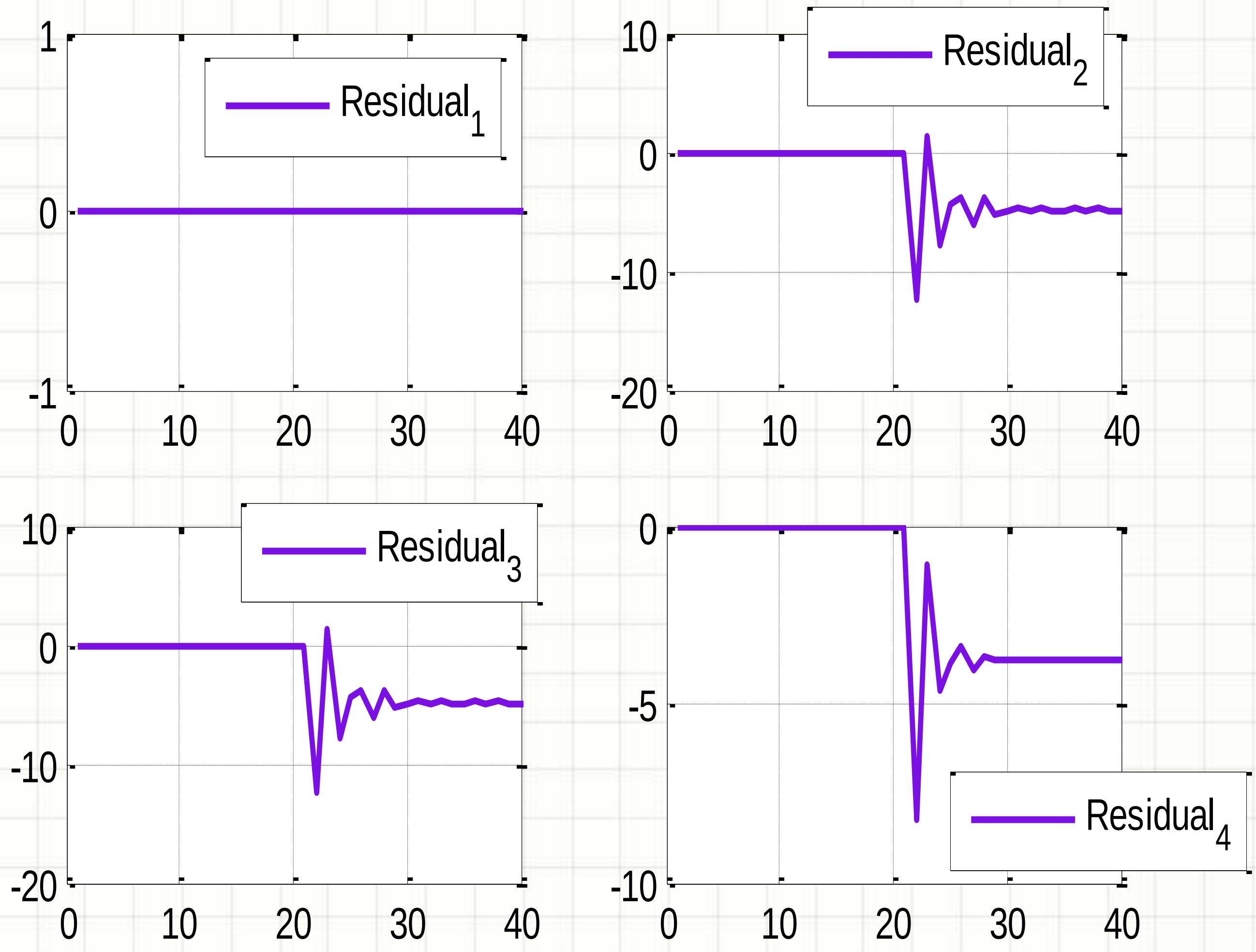


No fault, With noise

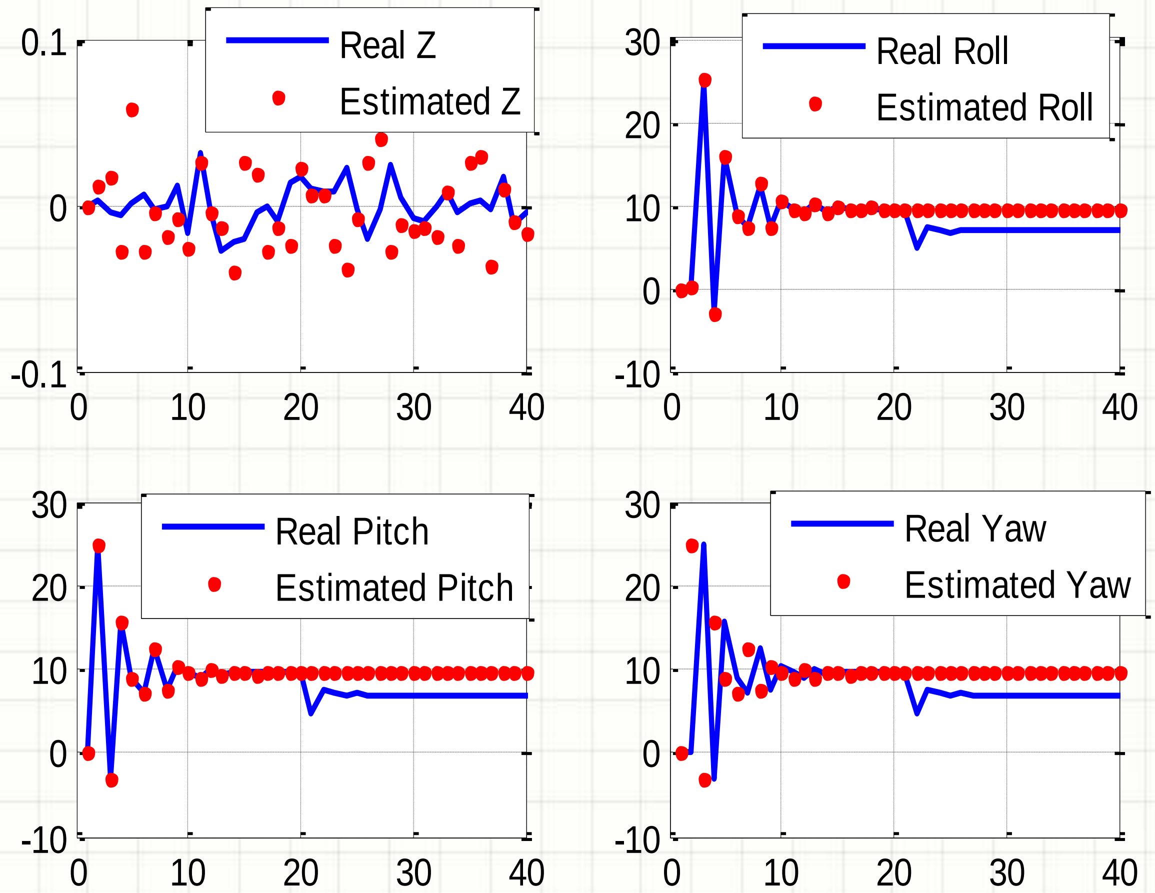


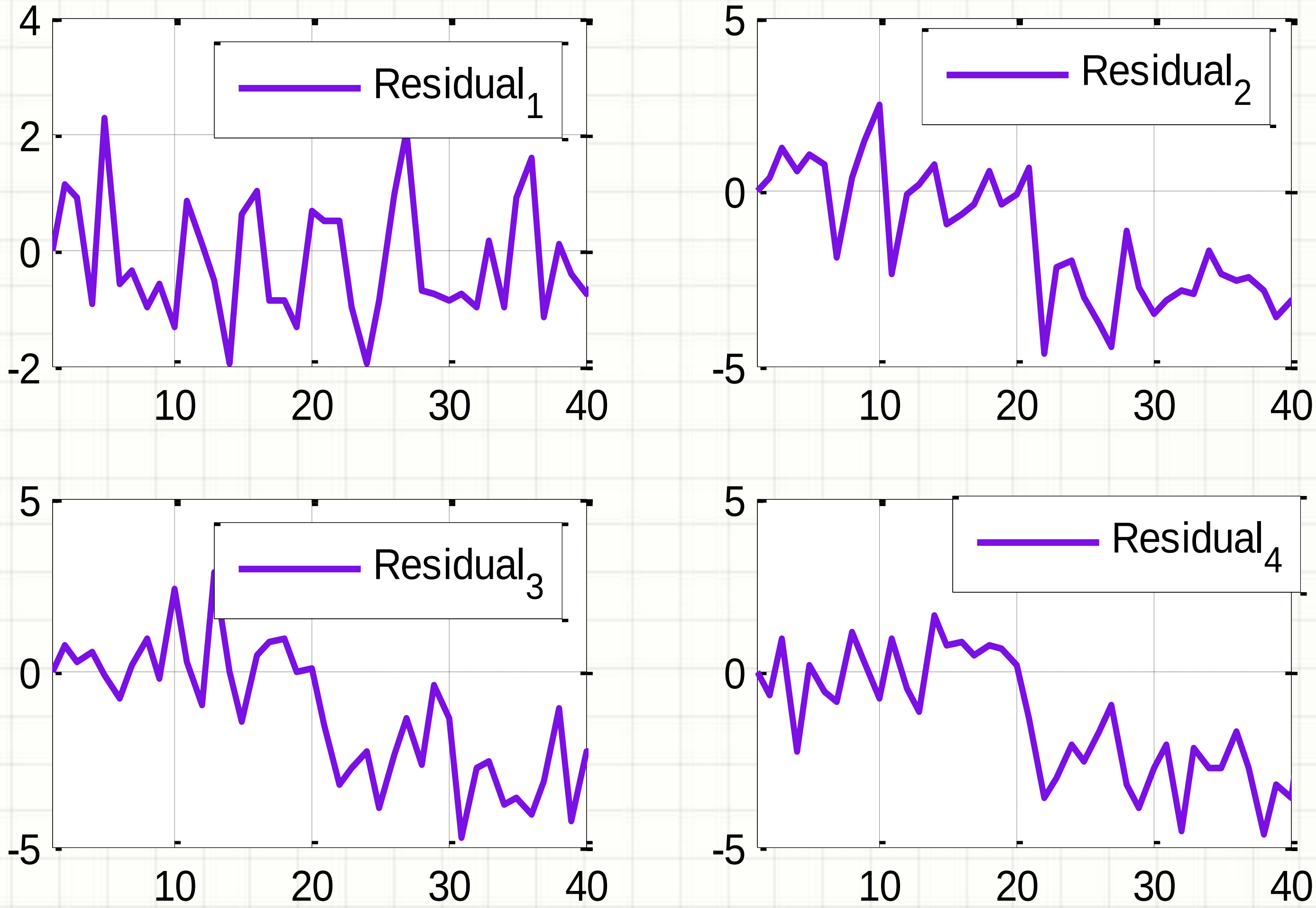
Faulty System , Without noise




Residual1
Residual2
Residual3
Residual4
1
0
-1
10
0
-10
-20
-5
0
10
20
30
40

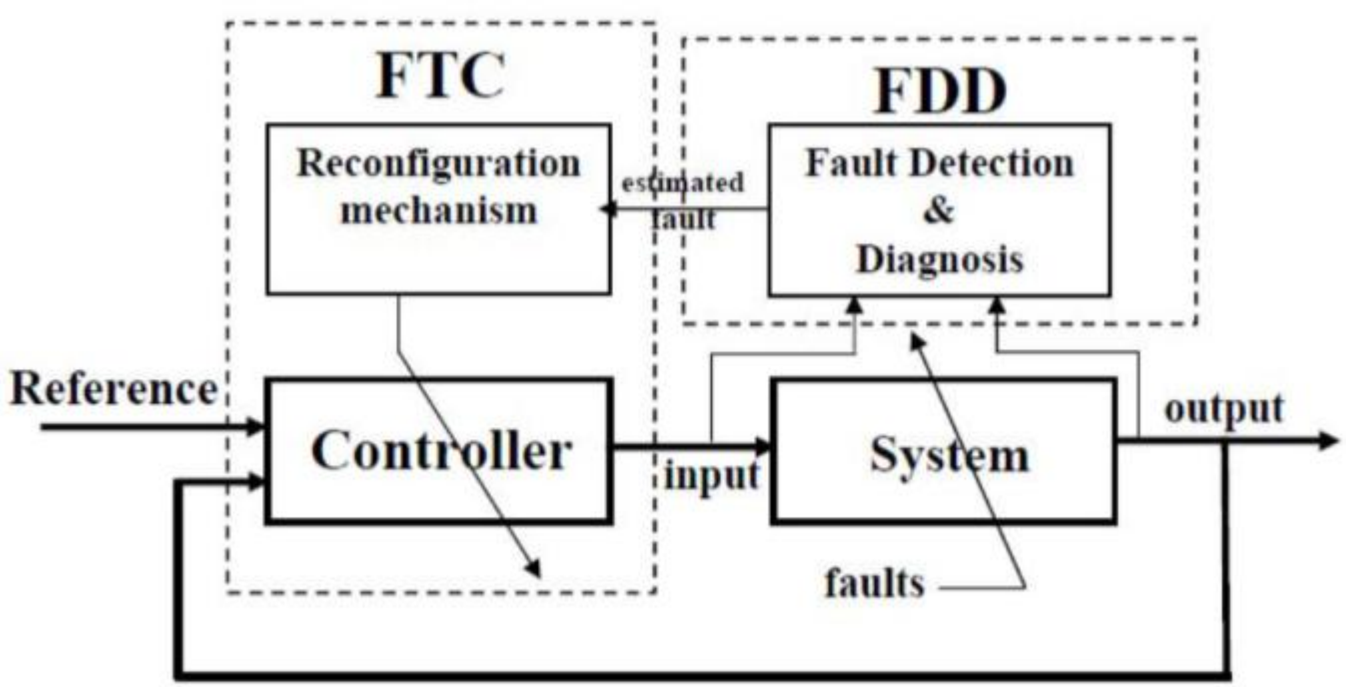
Faulty System With noise





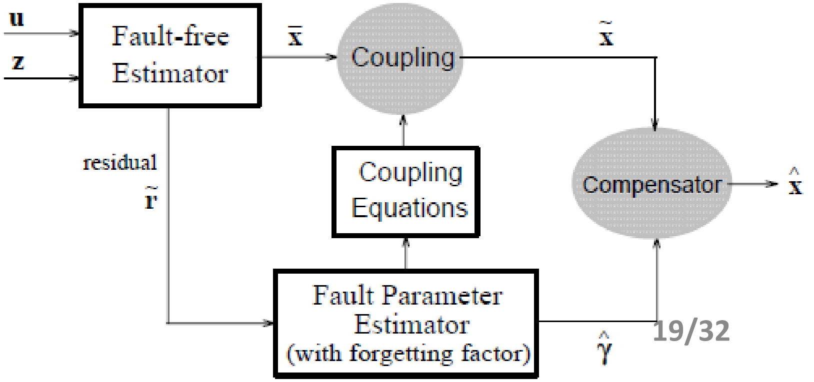
Residual1
4
2
0
-2
10
20
30
40
Residual2
5
0
-5
10
20
30
40
Residual3
5
0
-5
10
20
30
40
Residual4
5
0
-5
10
20
30
40

AFTCS



TSKF

Estimate the *state* and *bias* (*fault parameter*) simultaneously

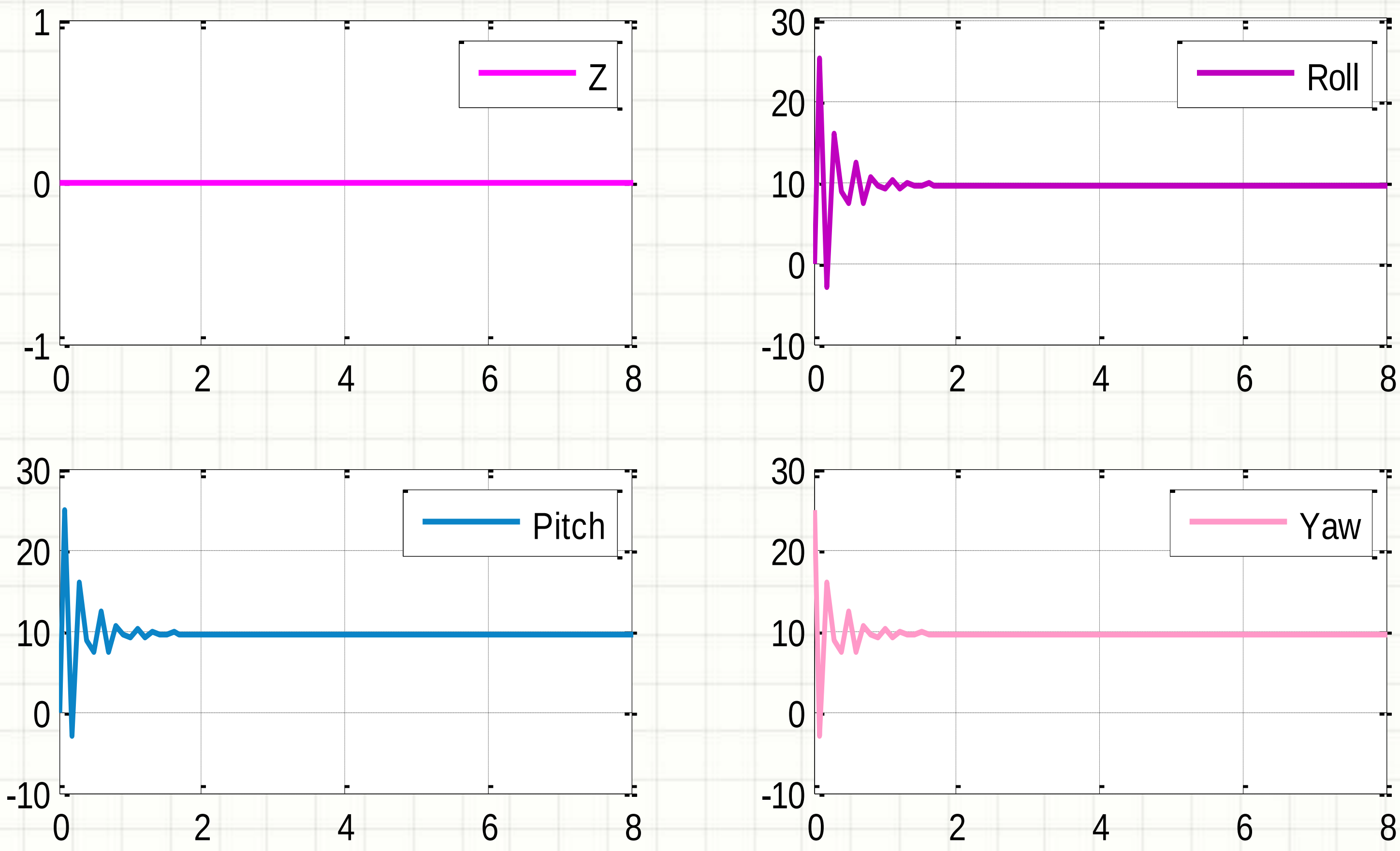


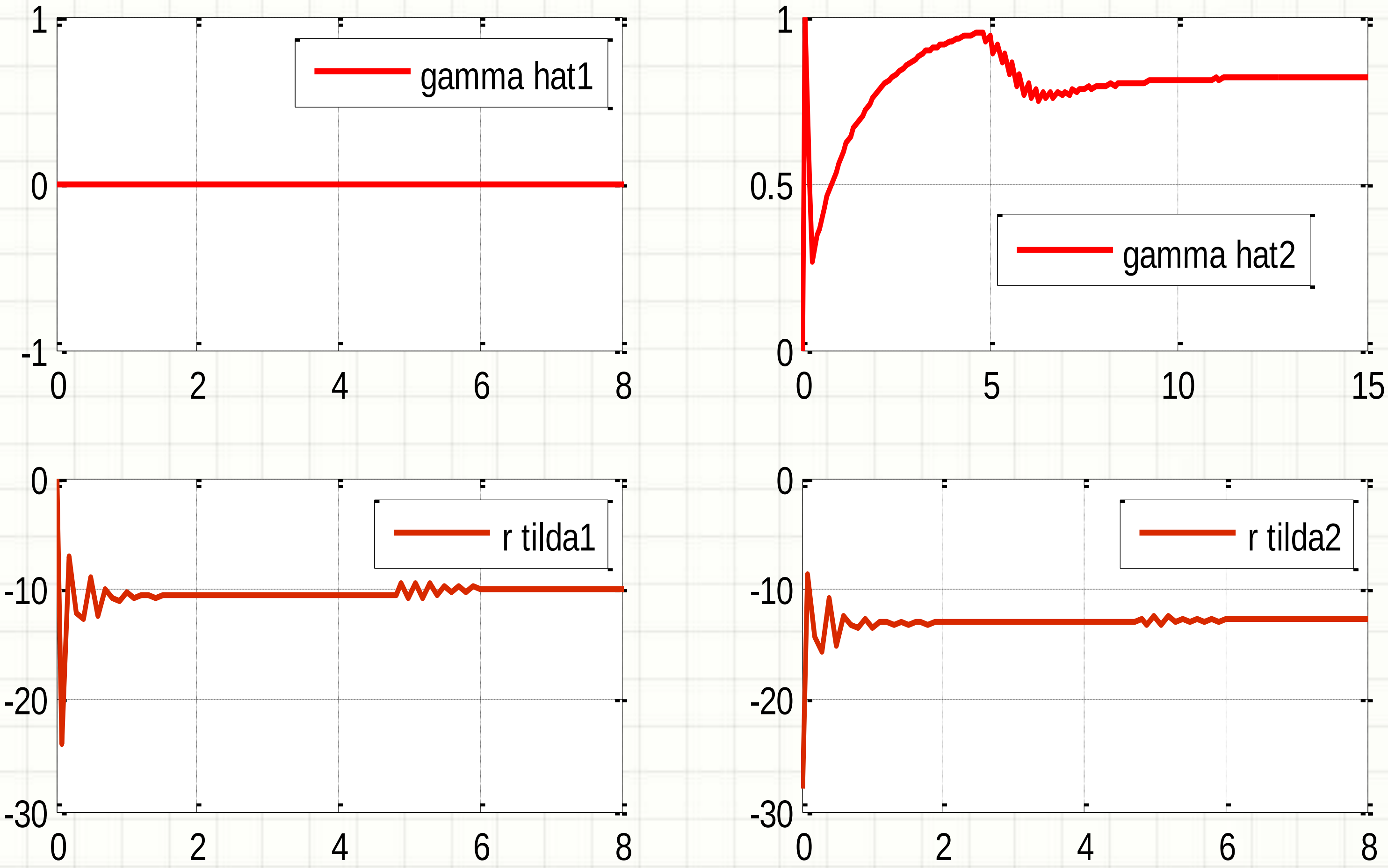
LQR

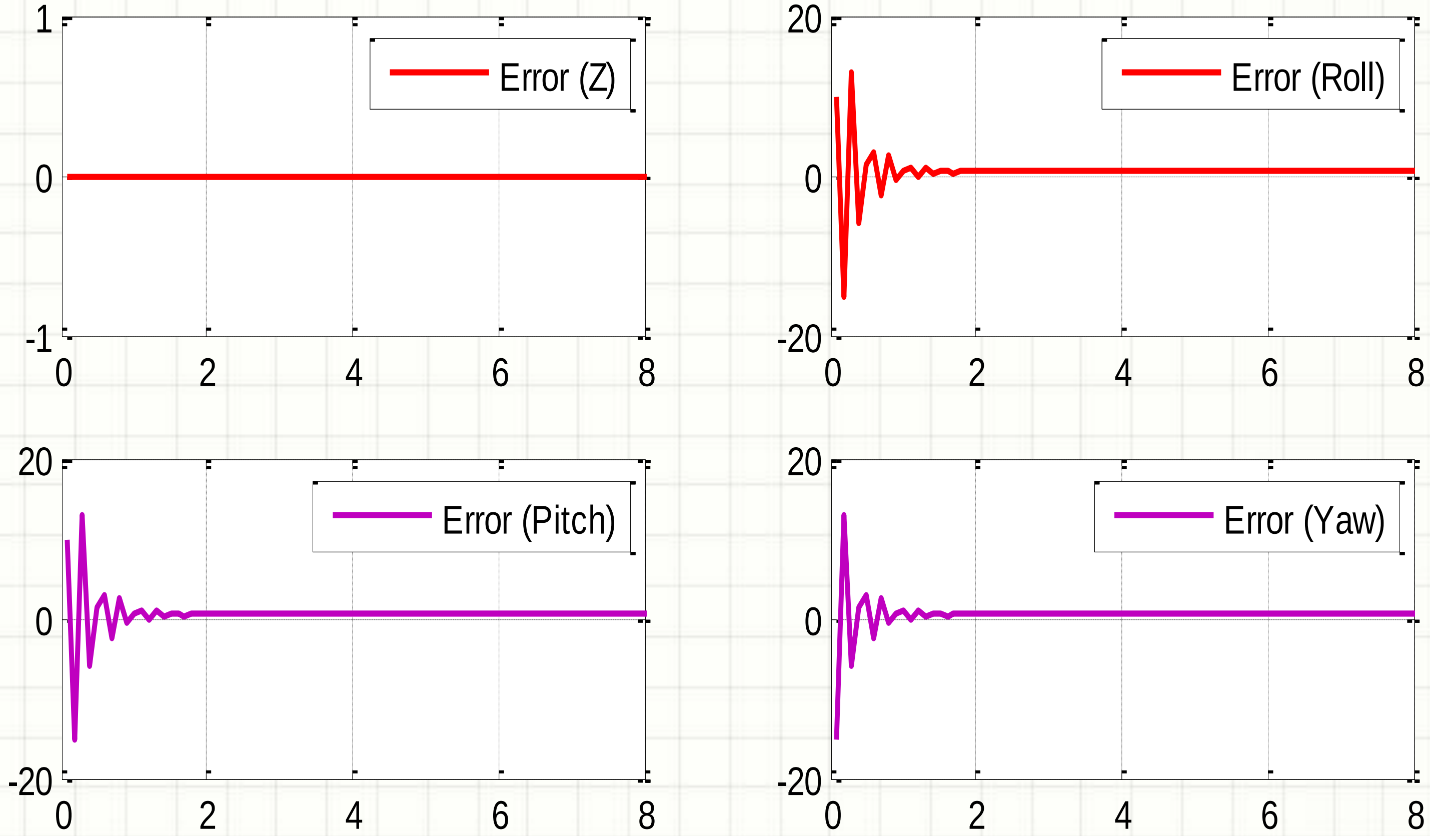
$$J = \int_{o}^{\infty} [X^T(t)QX(t) + u^T(t)Ru(t)]dt$$

$$[K, S, E] = LQR(A, B, Q, R, N)$$

Requirements: Information on fault time and post fault model


Z
Roll
Pitch
Yaw
1
0
-1
30
20
10
0
-10
0
2
4
6
8


gamma hat1
1
0
-1
0
2
4
6
8
gamma hat2
1
0.5
0
0
5
10
15
r tilda1
0
-10
-20
-30
0
2
4
6
8
r tilda2
0
-10
-20
-30
0
2
4
6
8

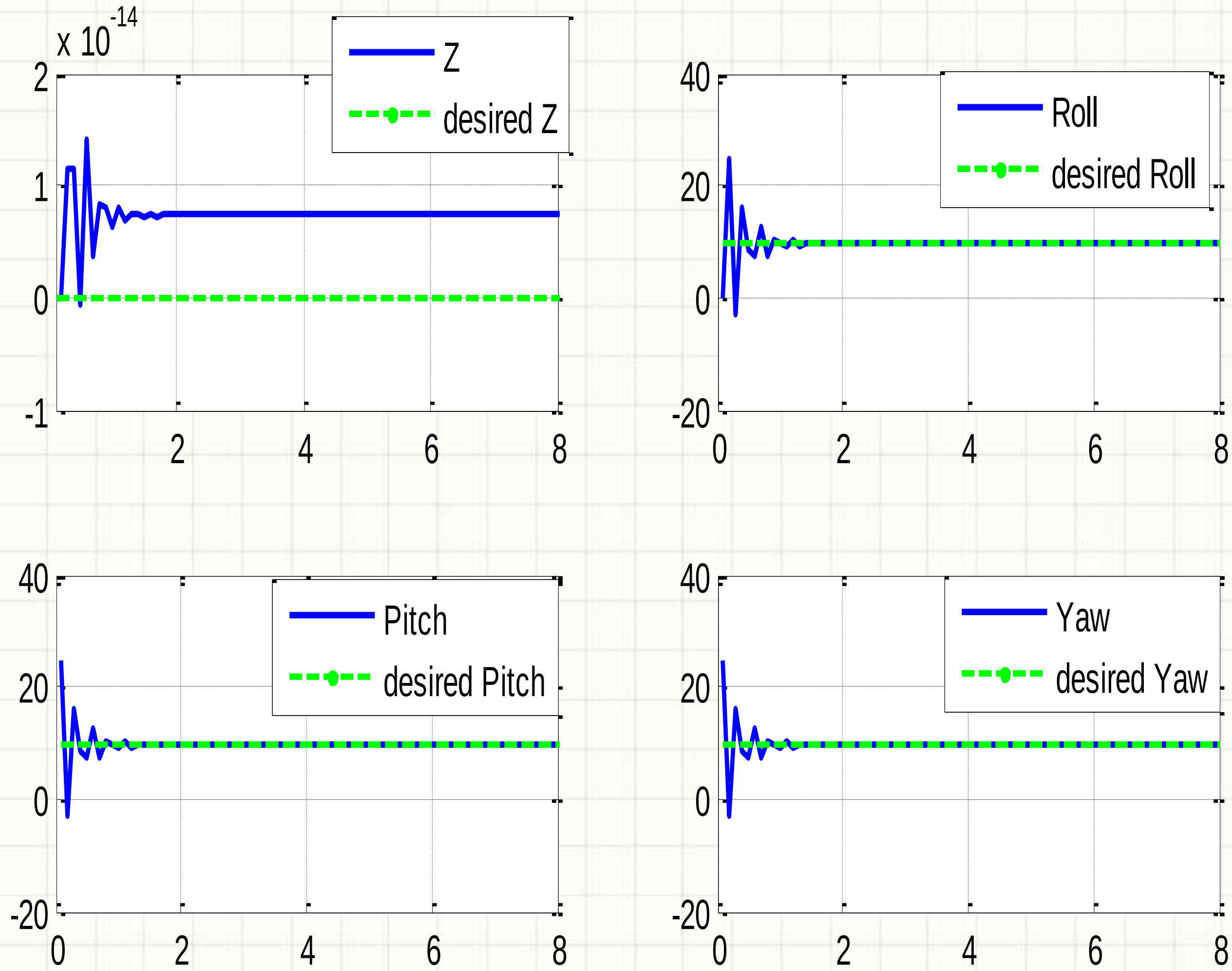

Error (Z)
Error (Roll)
Error (Pitch)
Error (Yaw)
1
0
-1
20
0
-20
0 2 4 6 8

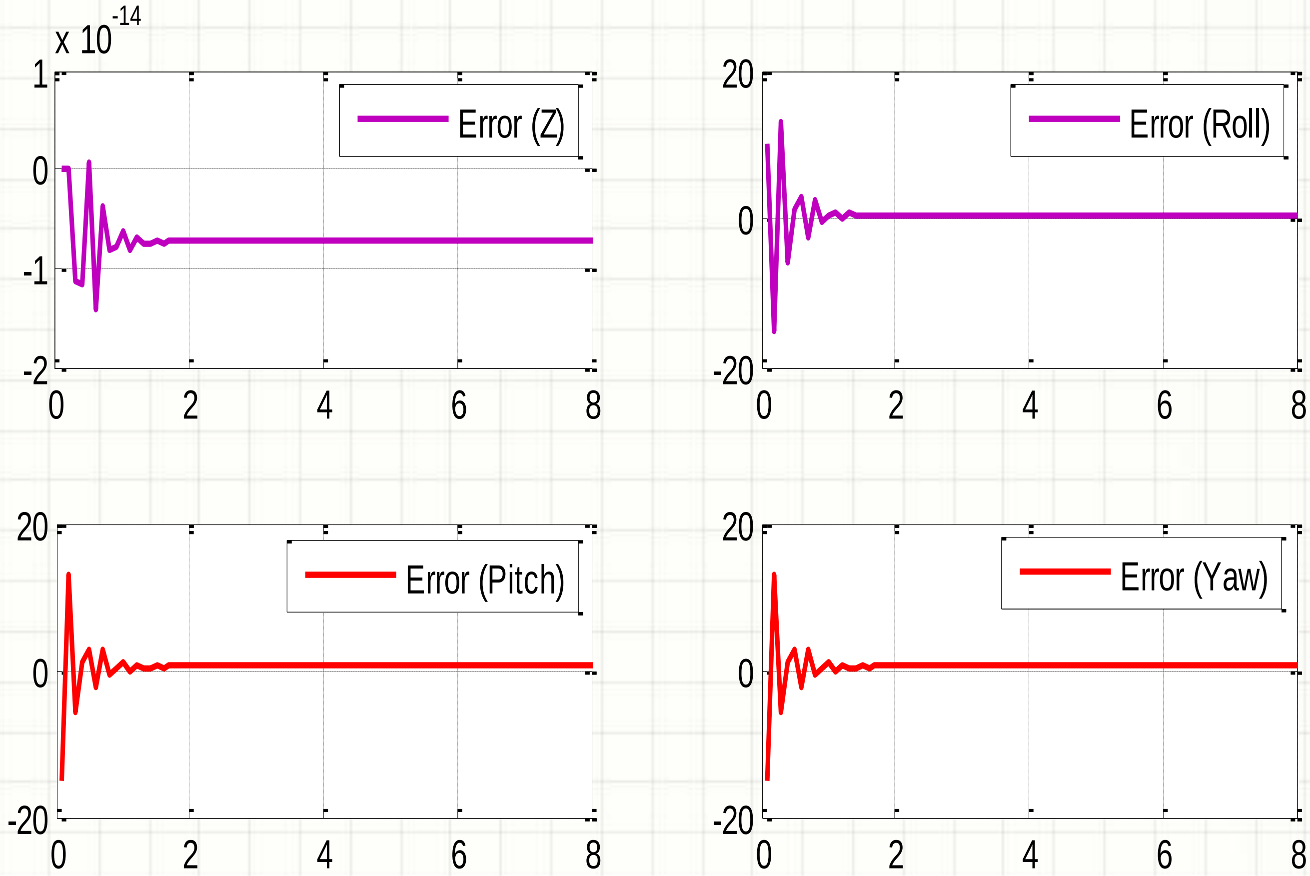
EA

$$\lambda_i^f = \lambda_i$$

V_i^f *as close as possible to* V_i

$$\lambda\left(A_f + B_f K_{feedback}\right) = \lambda(A + BK_{normal})$$


x 10^-14
Z
desired Z
Roll
desired Roll
Pitch
desired Pitch
Yaw
desired Yaw


x 10^-14
1
0
-1
-2
0
2
4
6
8
Error (Z)
20
0
-20
0
2
4
6
8
Error (Roll)
20
0
-20
0
2
4
6
8
Error (Pitch)
20
0
-20
0
2
4
6
8
Error (Yaw)

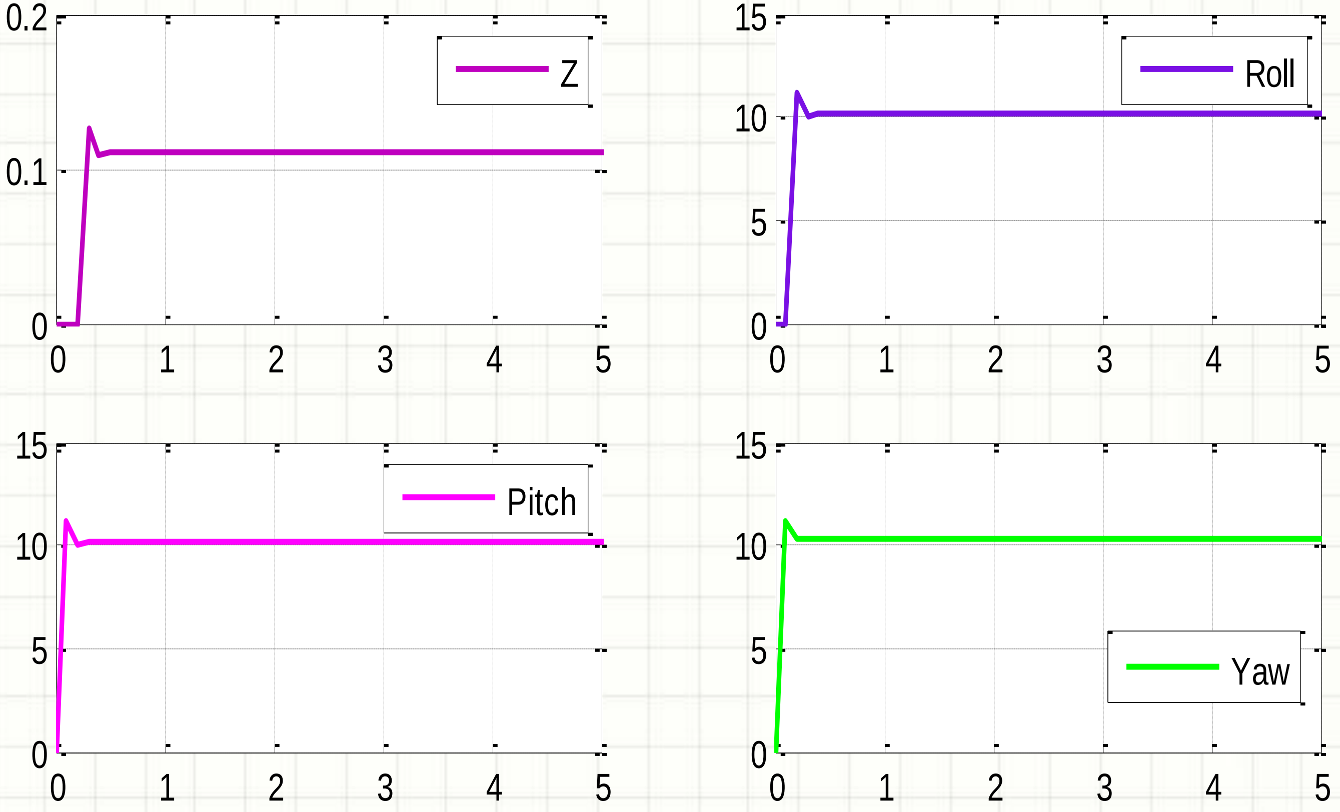
Pseudo-Inverse Method

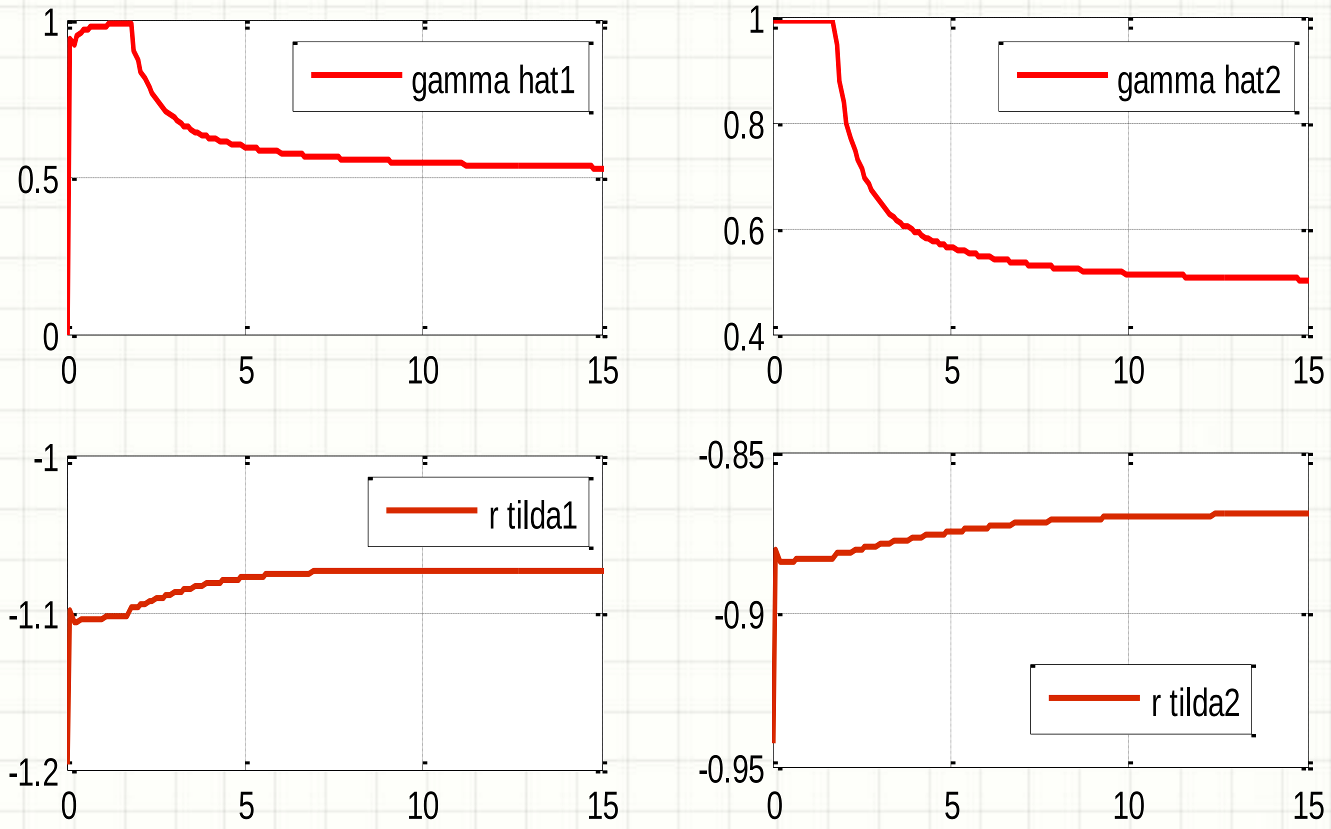
$$A_f + B_f K_f = A + BK$$

$$K_{feedback} = B_f^{\dagger}(A - A_f + BK)$$

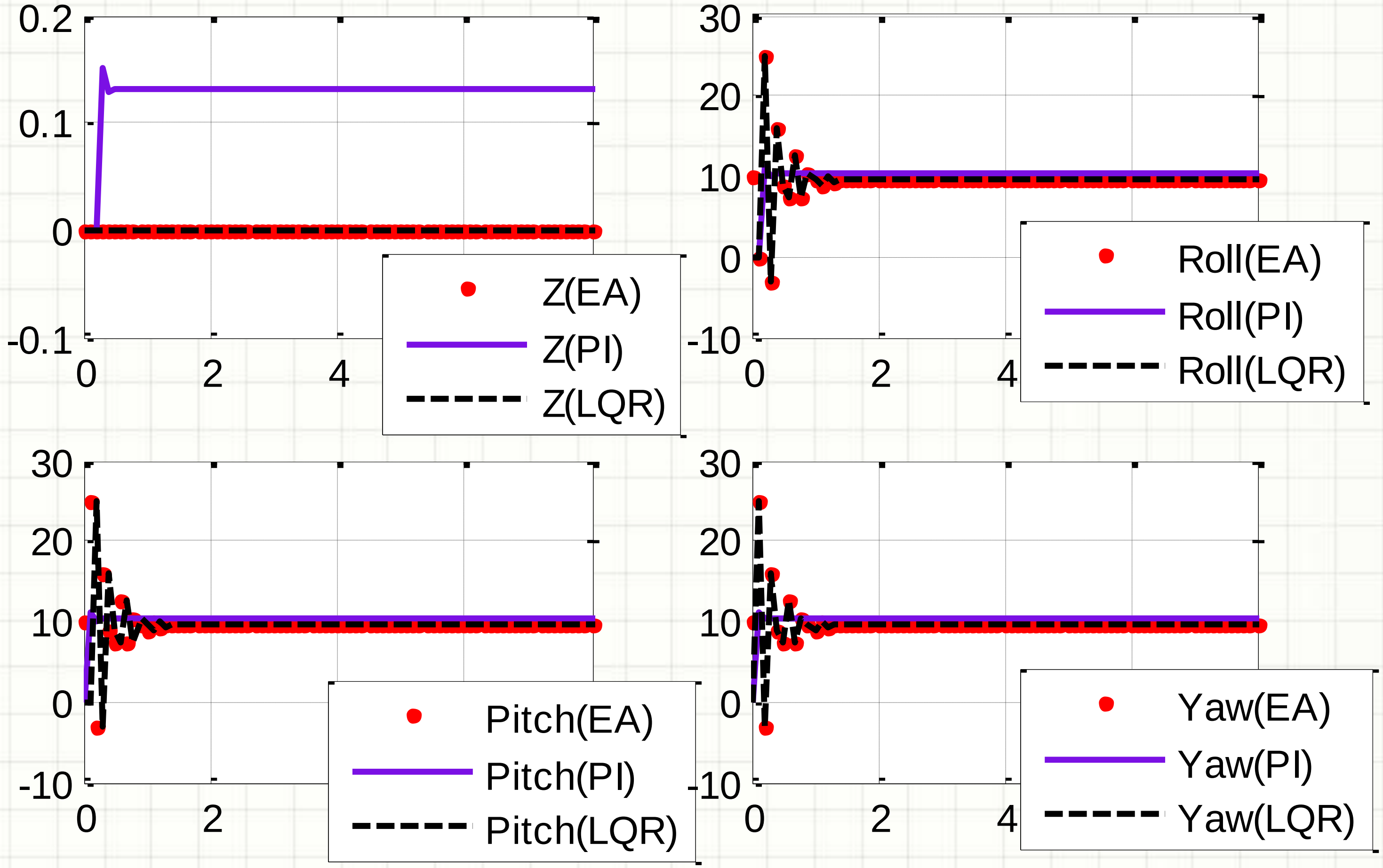
$$B_f^{\dagger} = ({B_f}^T B_f)^{-1} {B_f}^T$$

The controller is redesigned so that the closed-loop system matrix of the reconfigured system will be made as close as possible to that of the nominal system


Z
Roll
Pitch
Yaw
0.2
0.1
0
15
10
5
0
1
2
3
4
5


1
0.5
0
0
5
10
15
gamma hat1
1
0.8
0.6
0.4
0
5
10
15
gamma hat2
-1
-1.1
-1.2
0
5
10
15
r tilda1
-0.85
-0.9
-0.95
0
5
10
15
r tilda2

Comparison



0.2
0.1
0
-0.1
0
2
4
Z(EA)
Z(PI)
Z(LQR)
30
20
10
0
-10
0
2
4
Roll(EA)
Roll(PI)
Roll(LQR)
30
20
10
0
-10
0
2
Pitch(EA)
Pitch(PI)
Pitch(LQR)
30
20
10
0
-10
0
2
4
Yaw(EA)
Yaw(PI)
Yaw(LQR)

Conclusion

- PID controller can control the nonlinear model of the system
- Kalman filter can detect actuator fault
- All three methods of AFTCS can control the system
- PIM has error in comparison to the other methods

Suggestion for future work

- ATSEKF
- MPIM

Thank you