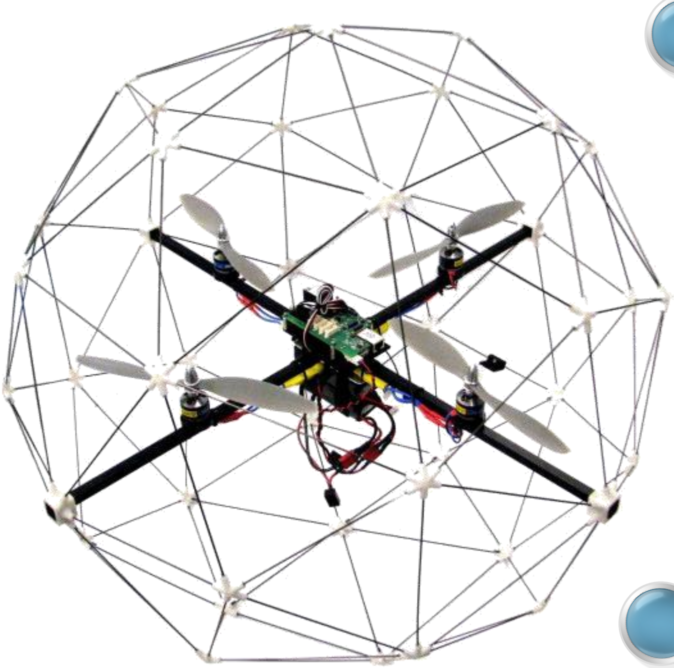
MECH 6091 – Flight Control System

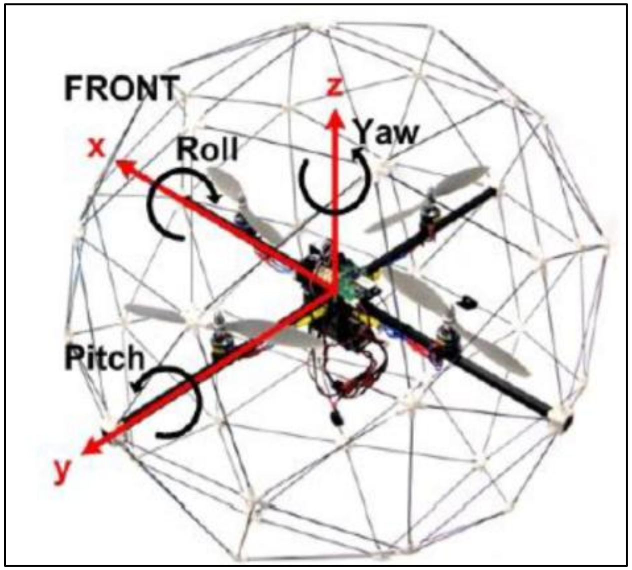
Javier Ortiz
Tarek Gouda

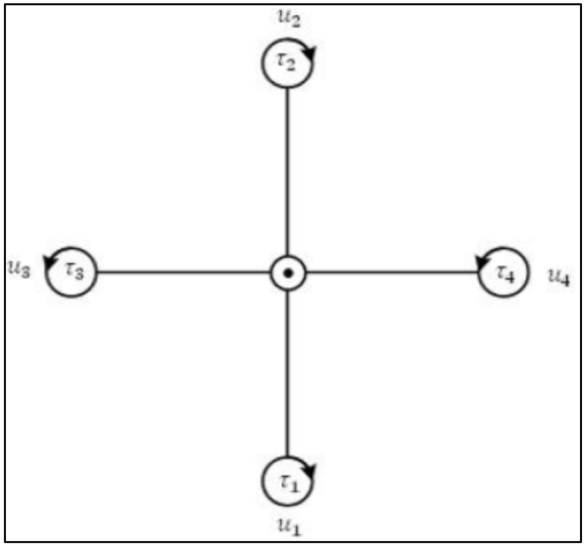
UNIVERSITÉ Concordia UNIVERSITY

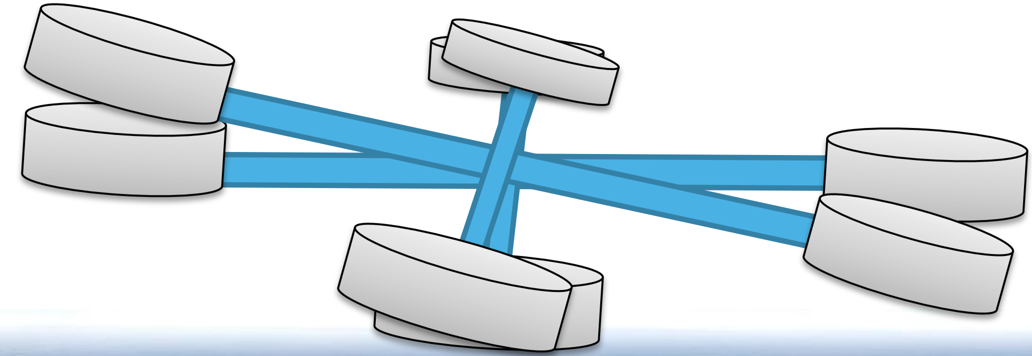
Contents



Introduction





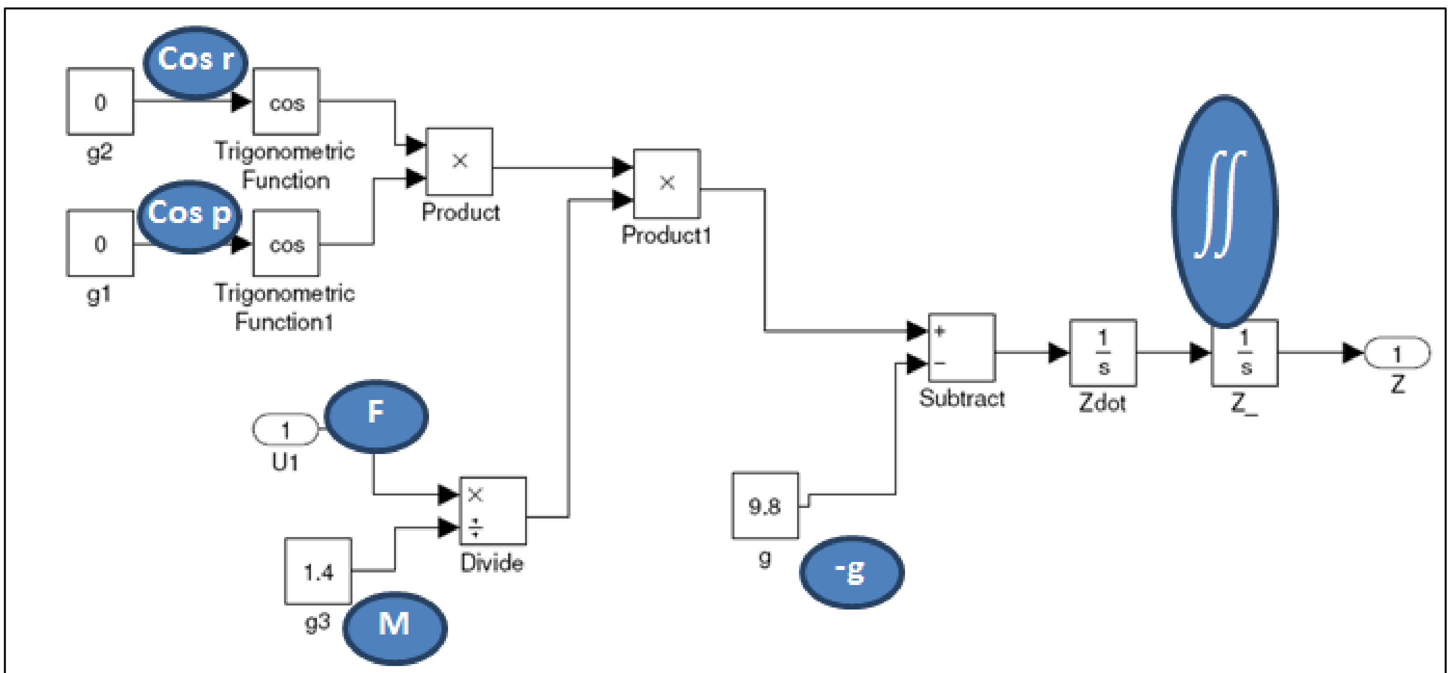


Building the Height Model

$$M\ddot{Z} = 4F\cos(r)\cos(p) - Mg$$

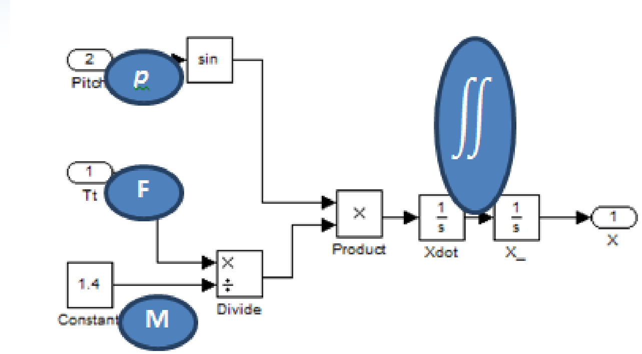
$$\ddot{Z} = \frac{4F}{M}\cos(r)\cos(p) - g$$

$$Z = \iint \frac{4F}{M}\cos(r)\cos(p) - g$$



Building X and Pitch Model

$$M\ddot{X} = 4F\sin(p)$$

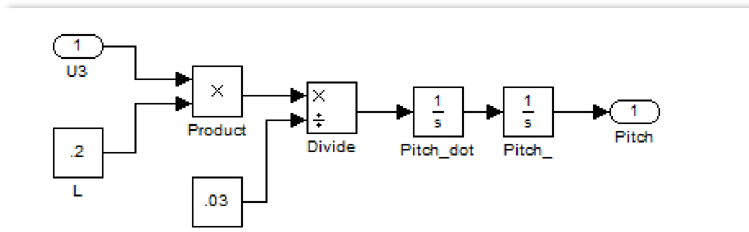


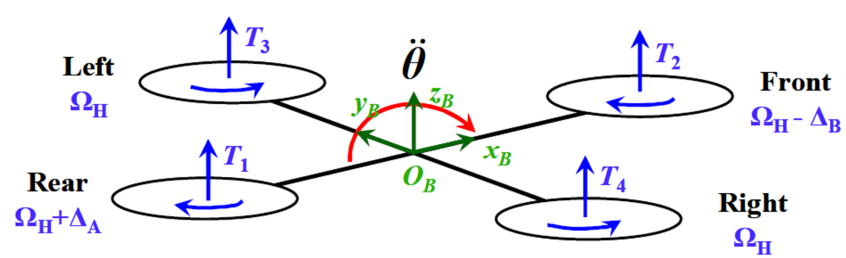
•The motion along X is caused by changing pitch angle.

•The command to change Pitch is increasing or decreasing the rear propeller speed and decreasing or increasing the front propeller.

$$J\ddot{\theta} = \Delta F L$$

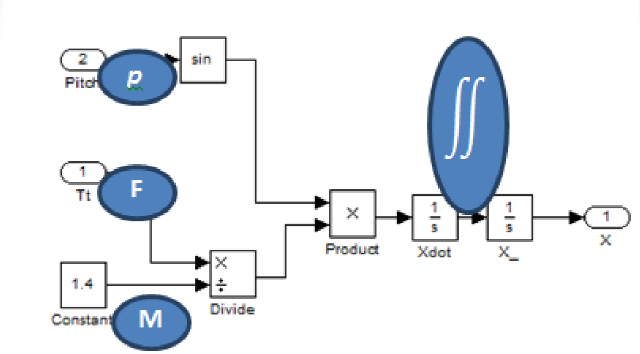
$$J = J_{roll} = J_{pitch}$$





Building Y and Roll Model

$$M\ddot{Y} = -4F\sin(r)$$

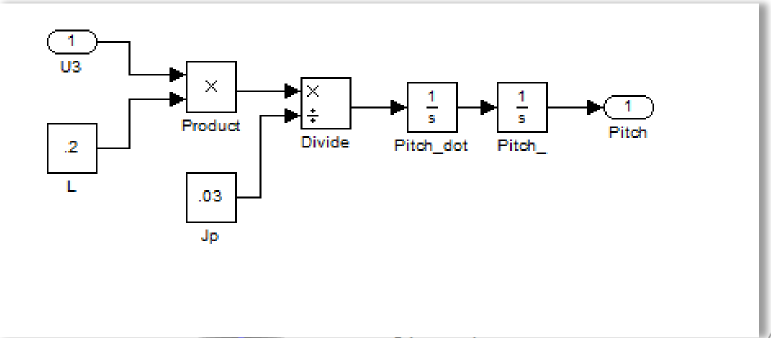


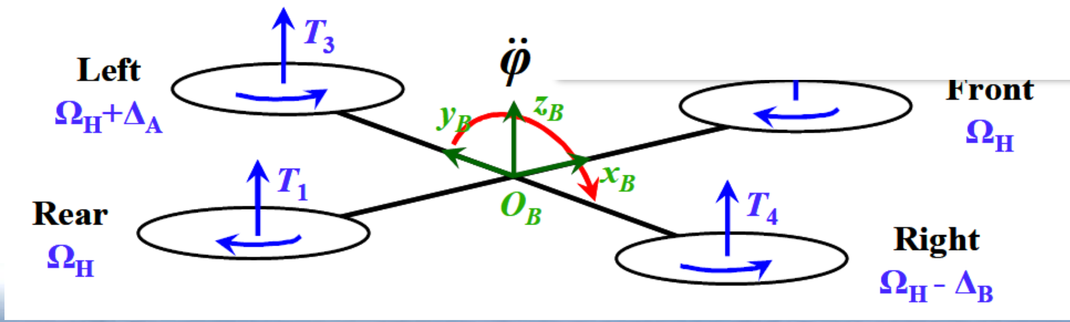
•The motion along is caused by changing Roll angle.

•The command to change Roll is increasing or decreasing the left propeller speed and decreasing or increasing the right propeller.

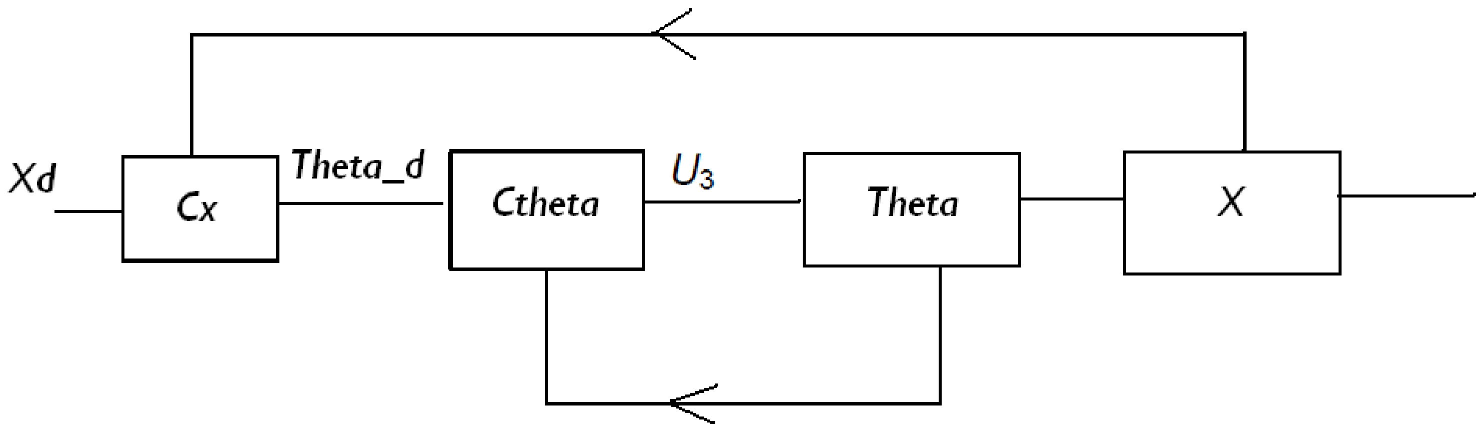
$$J\ddot{\theta} = \Delta F\, L$$

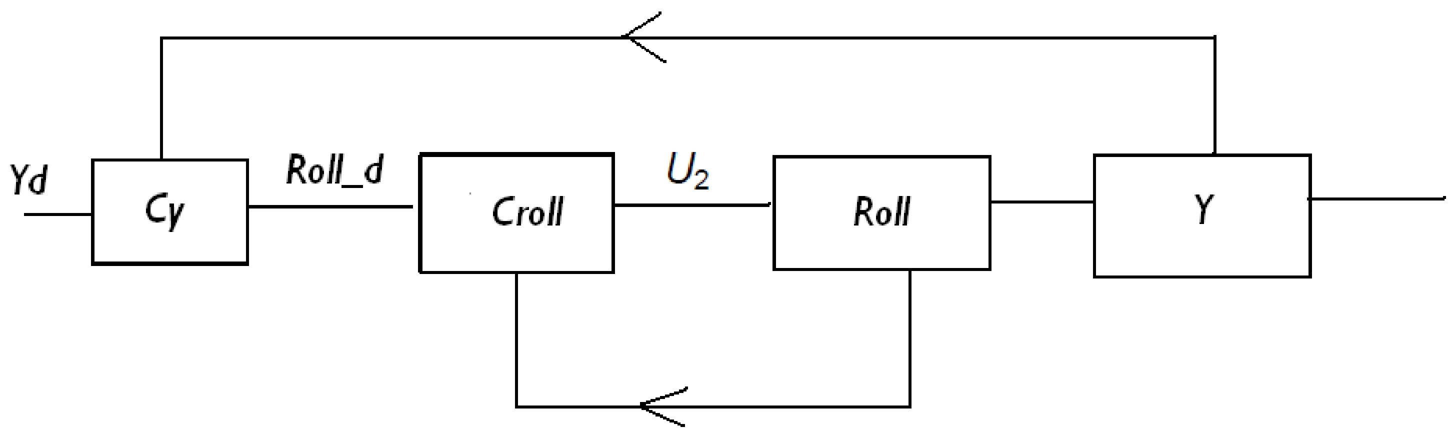
$$J = J_{roll} = J_{pitch}$$



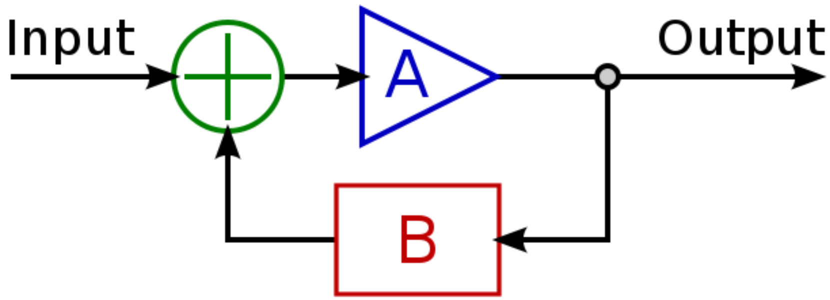


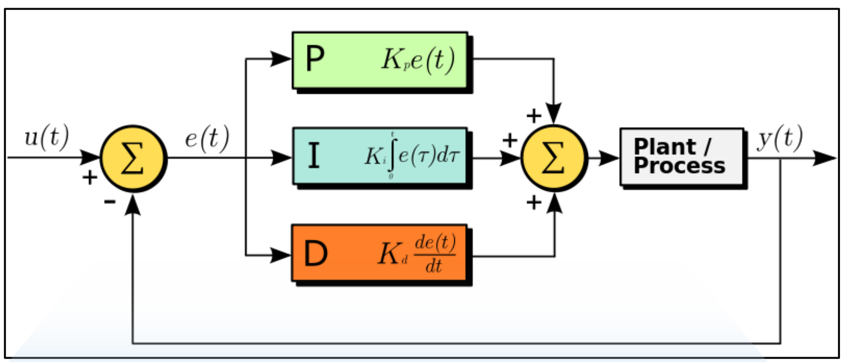
X & PICH Y & ROLL



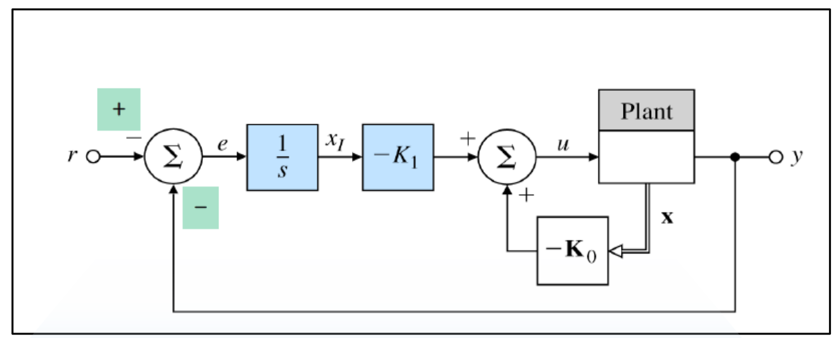


Control Mechanisms



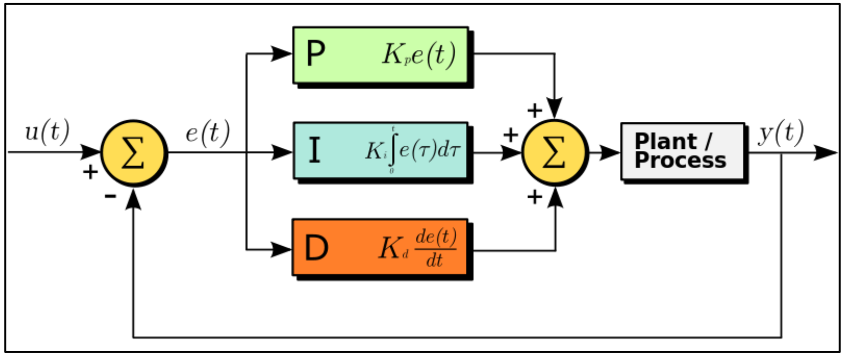


PID



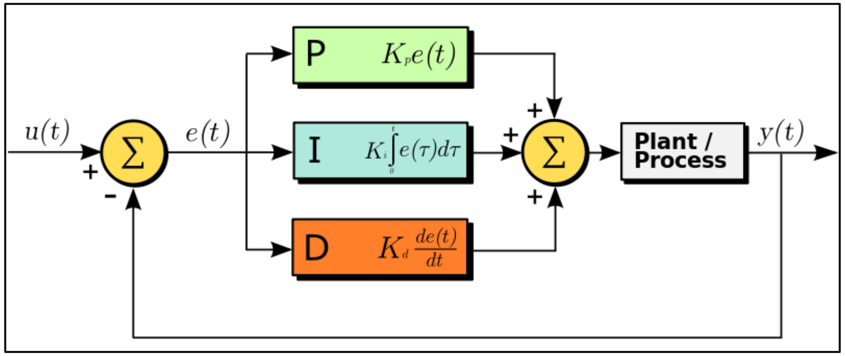
LQR

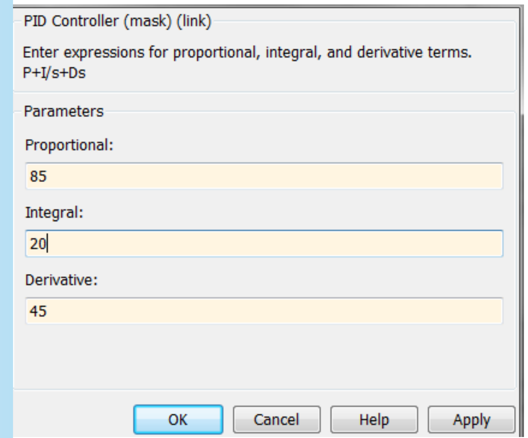
PID



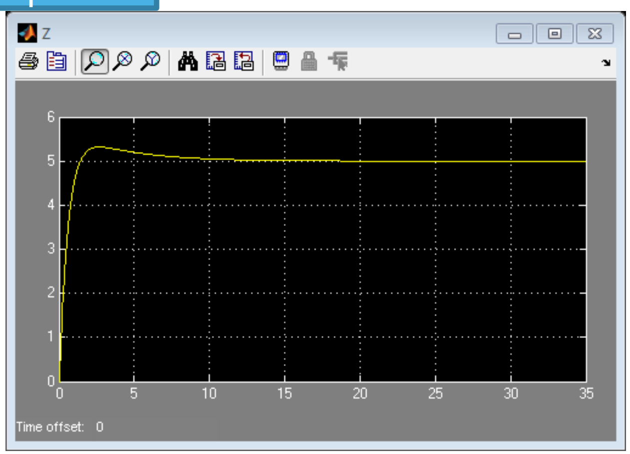
Term	Math Function	Effect on Control System
P **Proportional**	KP x Verror	Typically the main drive in a control loop, KP reduces a large part of the overall error.
I **Integral**	KI x $\int$ Verror dt	Reduces the final error in a system. Summing even a small error over time produces a drive signal large enough to move the system toward a smaller error.
D **Derivative**	KD x dVerror / dt	Counteracts the KP and KI terms when the output changes quickly. This helps reduce overshoot and ringing. It has no effect on final error.

PID – Height Model

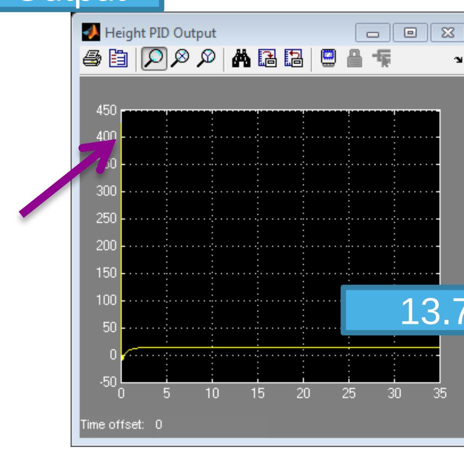




Z Response



PID Output

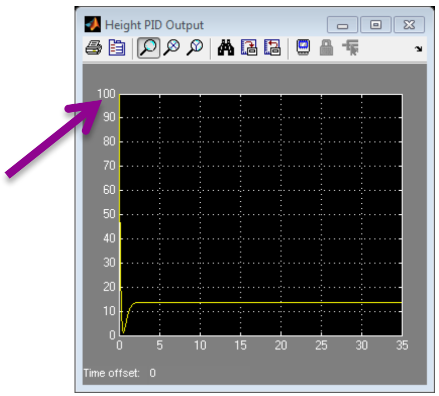


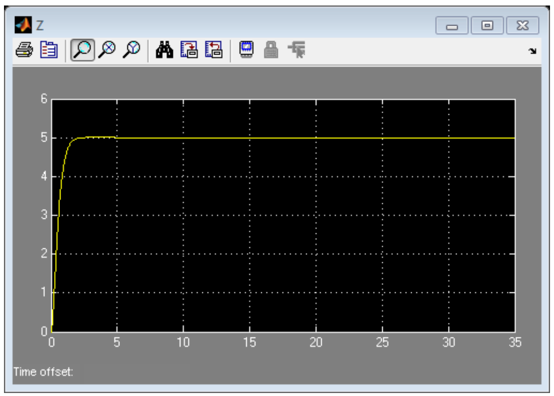
13.7N

Trial

PID – Height Model

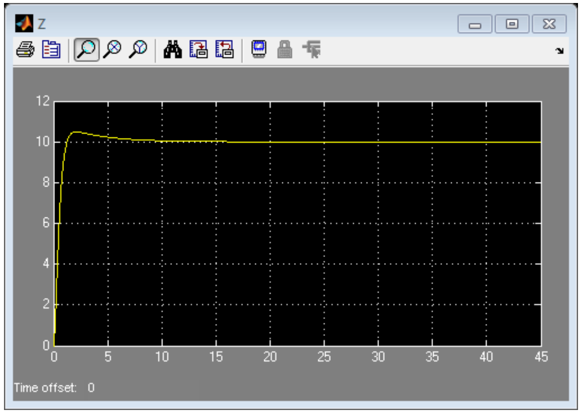
Desired Z=5		
Term	**1st Trial**	**Final Trial**
P	85	20
I	20	5
D	45	10

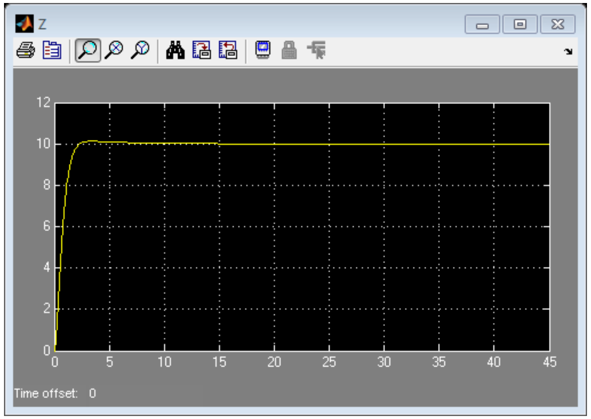




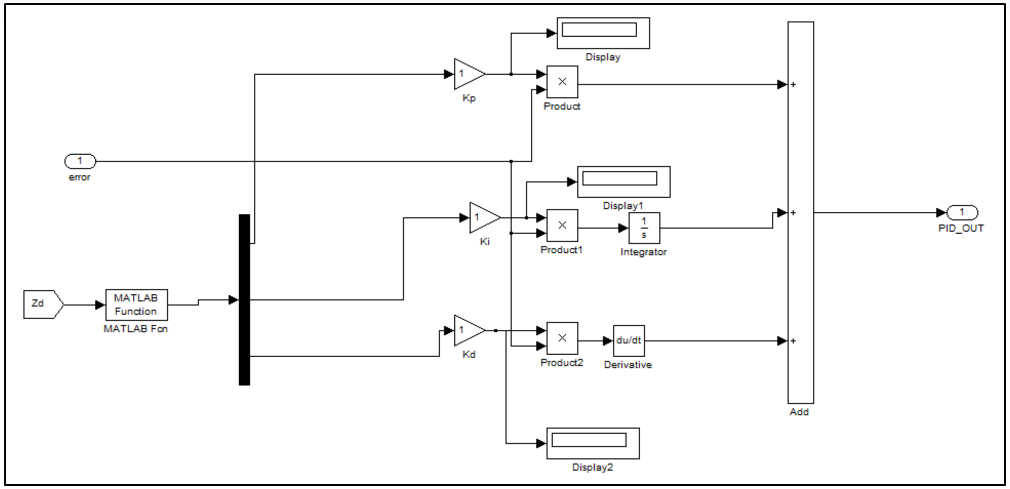
PID Height Model

Desired Z=10			
Term	**1st Trial**	**Final Trial**	**New Gain**
P	85	20	10
I	20	5	2
D	45	10	7



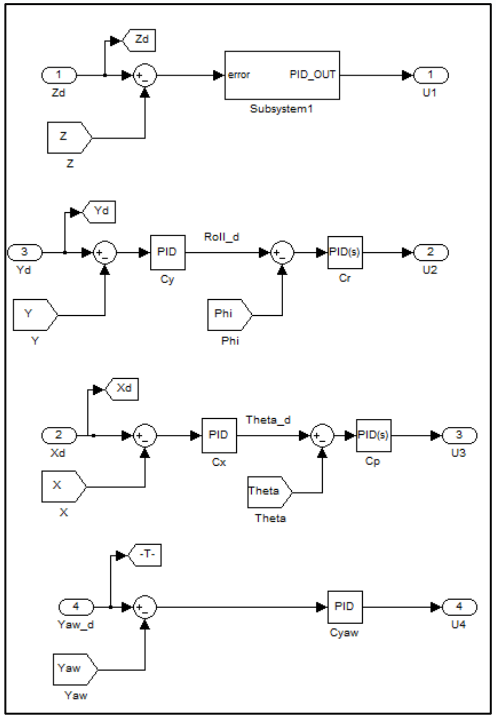


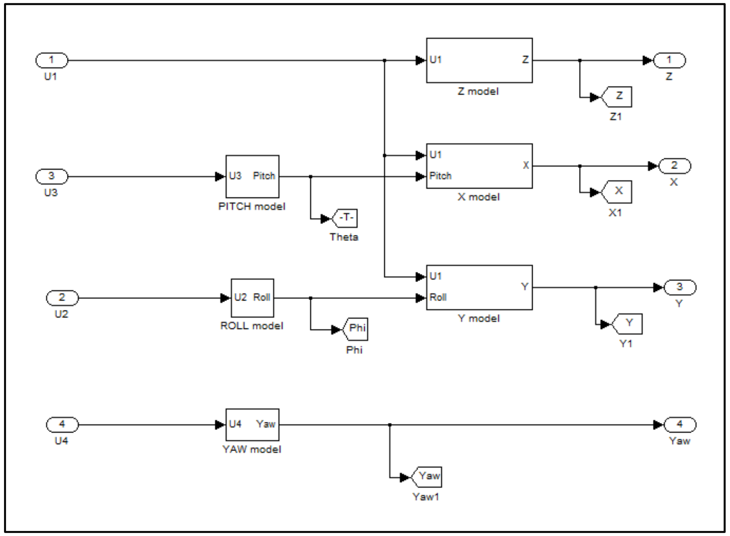
PID – Gain Scheduling




```
function out = GAIN(Zd)


if Zd==1
   Kp=85
   Ki=20
   Kd=45

elseif (Zd>=2 & Zd<=5)
   Kp=20
   Ki=5
   Kd=10

elseif (Zd>=6 & Zd<=10)
       Kp=10
       Ki=2
       Kd=7
    end
out(1)=Kp
out(2)=Ki
out(3)=Kd
```

PID Other Elements





Parameter	Proportional	Integral	Derivative
X position	0.5	0	0.5
Y position	0.5	0	0.5
Pitch angle	10	0	2
Roll angle	10	0	2
Yaw angle	0.2	0	0.1

LQR

- A system can be defined in state space form as:

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

$$Y = Cx$$

- The feedback control law is to determine the gain K to stabilize and improve the performance of the system with the state-variable feedback

$$U = -Kx$$

- The new controlled dynamics of the system becomes

$$\dot{X} = (A - BK)x$$

LQR

- The selection of the feedback gains K is made by LQR (Linear Quadratic Regulator). This method is based in the minimized of the cost function J

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt$$

- The Matlab Function ‘K=lqr(A,B,Q,R)’ is used to find the values for K.

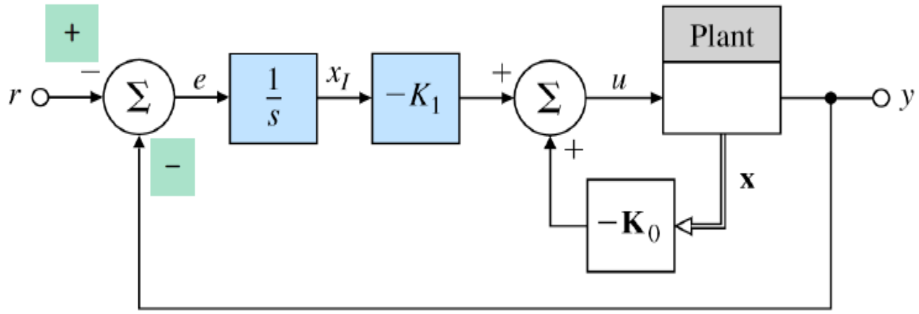
- Where Q is a positive-define matrix and R is positive-define matrix, both are symmetric.

- LQR methodology attempts to balance between a faster response and a low control effort.

LQR

TRACKING A REFERENCE INPUT

- For tracking a reference input we implement a LQR + INTEGRAL FORWARD CONSTANT (Ki).



LQR

LQR MATLAB

- Controllable and Observable.
- Matlab function ‘*ctrb’* and *‘obsv’*

MODEL	CTR_RANK	OBSV_RANK
Z	4	4
X & Y	3	3
Θ & ϕ	3	3
Yaw	2	2

LQR

Height Model

$$\begin{bmatrix} \dot{Z} \\ \ddot{Z} \\ \dot{v} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{4K}{M} & 0 \\ 0 & 0 & -w & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ w \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -g \\ 0 \\ 0 \end{bmatrix}$$

- The last row is a forth state to facilitate the use of an integrator for tracking input controller. The values for K, M and w is taken from Lab manual (K= 120N, M=1.4Kg & w=15 rad/sec)

$$\begin{bmatrix} \dot{Z} \\ \ddot{Z} \\ \dot{v} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 342.8571 & 0 \\ 0 & 0 & -15 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 15 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -9.8 \\ 0 \\ 0 \end{bmatrix}$$

LQR

$$Az = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 342.8571 & 0 \\ 0 & 0 & -15 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad Bz = \begin{bmatrix} 0 \\ 0 \\ 15 \\ 0 \end{bmatrix} \quad Cz = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Dz = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

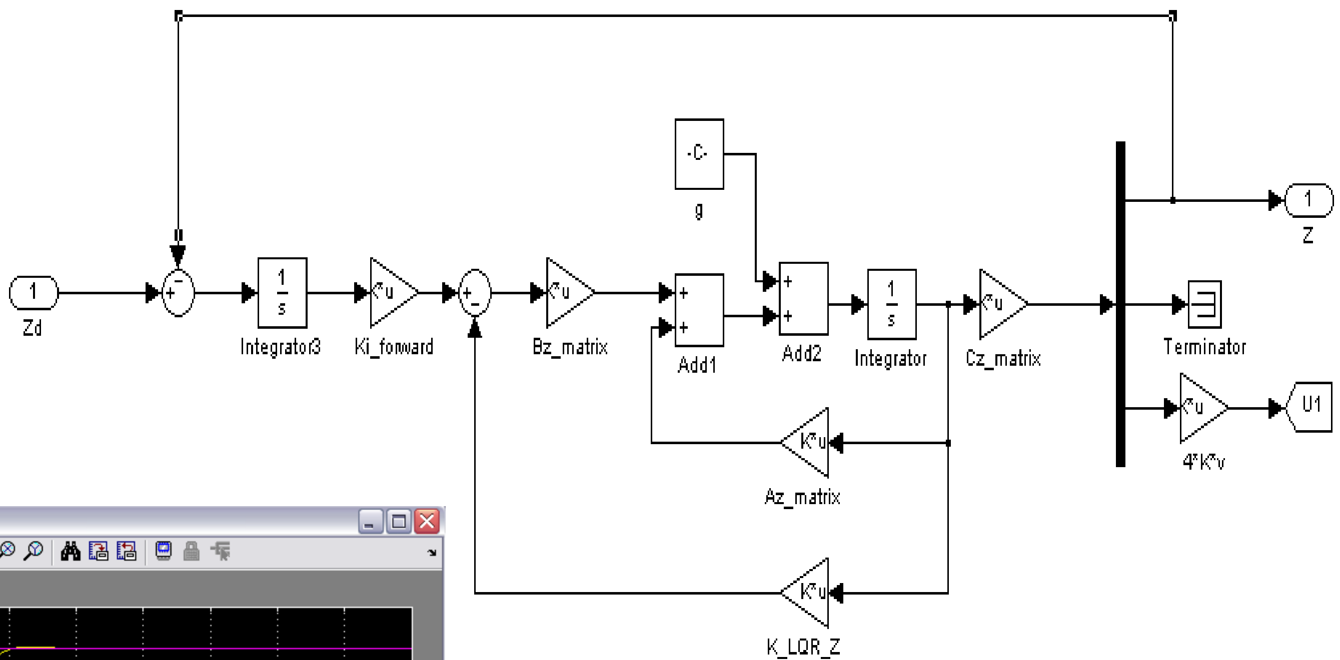
- We use the LQR matlab function to calculate K. We selected Q as a diagonal matrix of 1 (4x4) and R is 1.

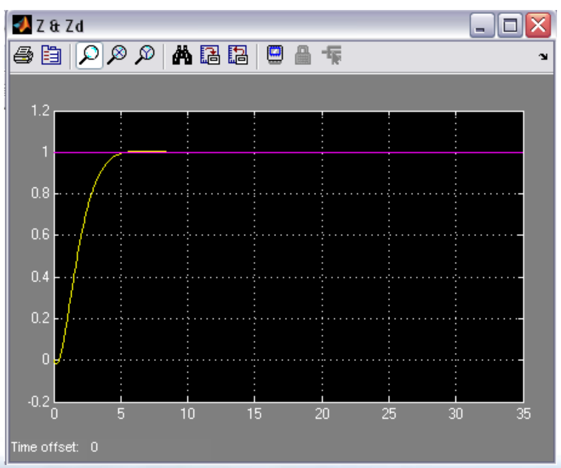
- K=lqr(Az,Bz,Q,1) K =[1.7522 1.0351 6.0227 1]

- The first 3 terms are the LQR feedback controller and the last term is the Ki (forward integral constant).

LQR

- HEIGHT SIMULINK MODEL





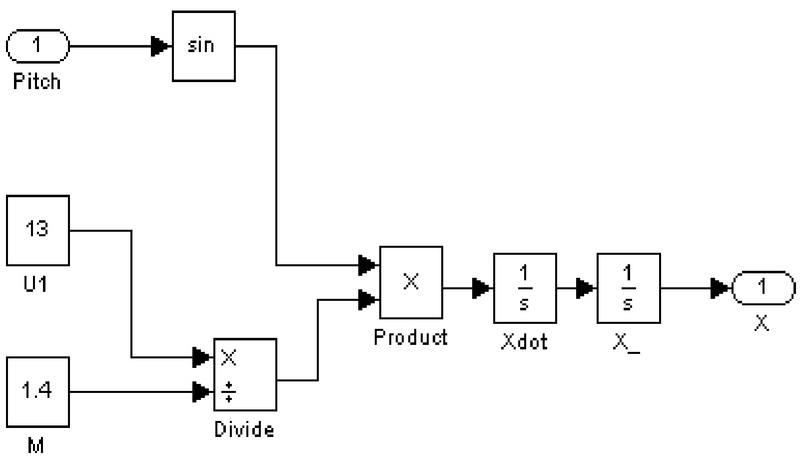
LQR

X & Y POSITION

$$M\ddot{X} = 4Fsin(p)$$

$$M\ddot{Y} = 4Fsin(r)$$

- These equations were implemented on simulink. We use the '*linmod'* matlab function to get the matrix A,B,C and D for X & Y.

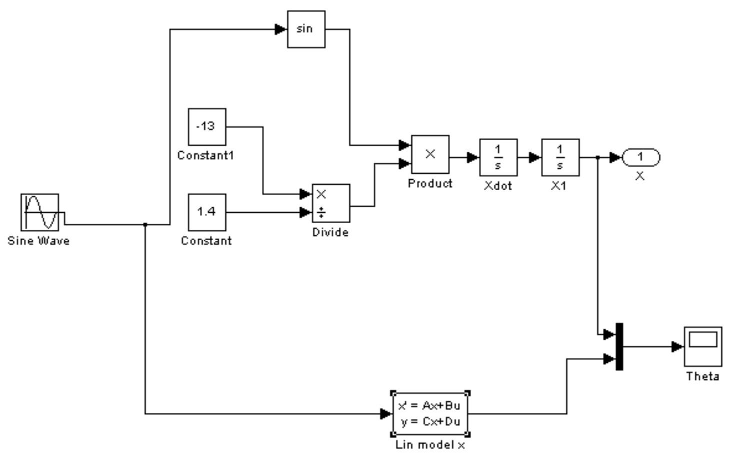


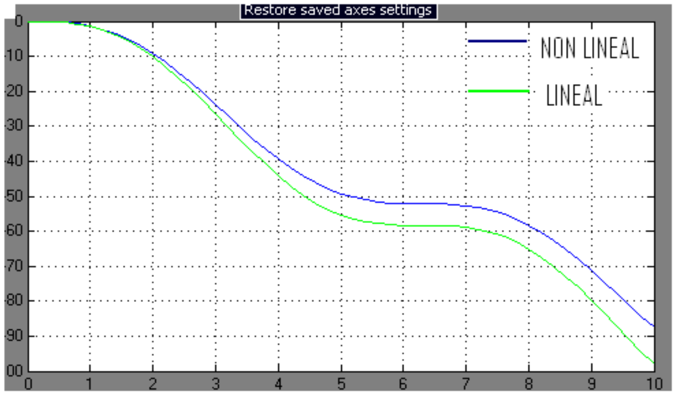
LQR

- [Axlin,Bxlin,Cxlin,Dxlin]=linmod('NONLIN_X1')

$$Axlin = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad Bxlin = \begin{bmatrix} 0 \\ 9.2857 \end{bmatrix} \quad Cxlin = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad Dxlin = 0$$

- The lineal and non lineal were compared.

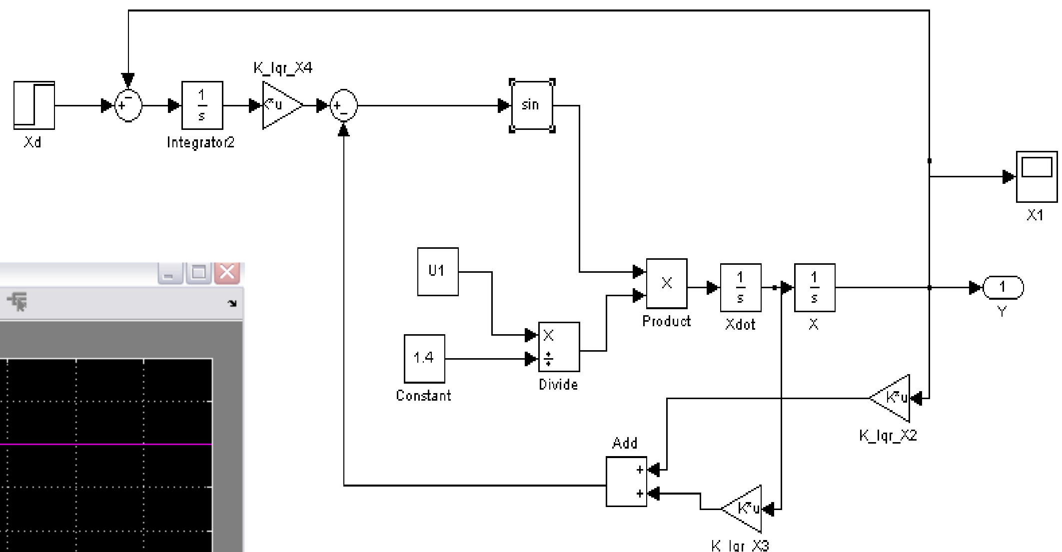


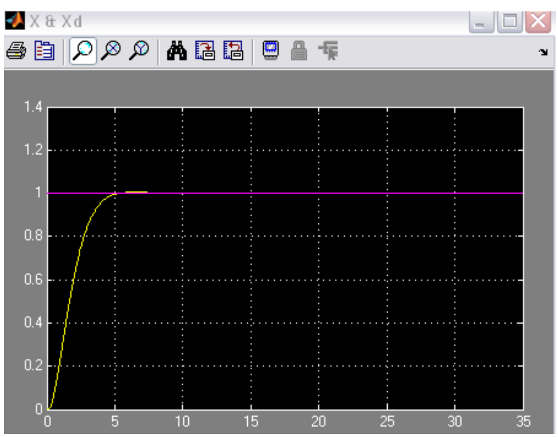


LQR

- The K gain matrix is calculated with *'lqr'* matlab function and the Q matrix is a diagonal matrix of 1 (3X3) and R is 1.

- K=lqr(Axlin1,Bxlin,Q,1) K =[1.8336 1.1811 1.0000]





LQR

ROLL & PITCH MODEL

$$\begin{bmatrix}\dot{\theta}\\ \ddot{\theta}\\ \dot{v}\\ \dot{s}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & \frac{KL}{J} & 0\\ 0 & 0 & -w & 0\\ 1 & 0 & 0 & 0\end{bmatrix} + \begin{bmatrix}0\\ 0\\ w\\ 0\end{bmatrix}$$

- The values for K, L , w & J are taken form lab manual (K=120N, L=0.2m, J=.03Kgm^2 & w=15 rad/sec)

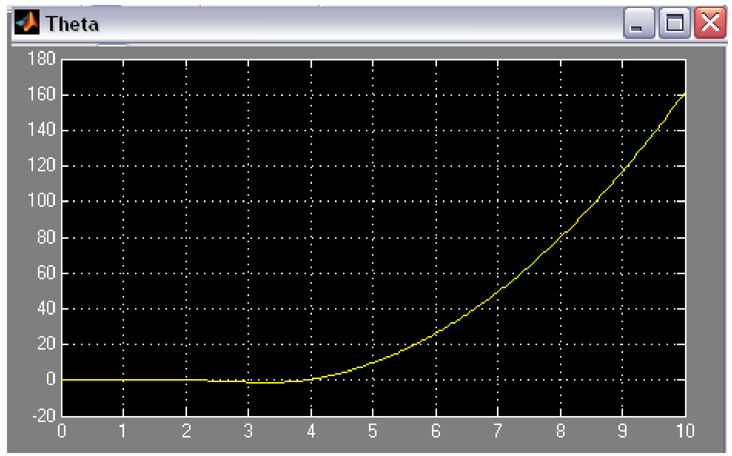
$$\begin{bmatrix}\dot{\theta}\\ \ddot{\theta}\\ \dot{v}\\ \dot{s}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 800 & 0\\ 0 & 0 & -15 & 0\\ 1 & 0 & 0 & 0\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 15\\ 0\end{bmatrix}$$

- The A,B,C and D matrix are:

$$Ap = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 800 & 0\\ 0 & 0 & -15 & 0\\ 1 & 0 & 0 & 0\end{bmatrix} \quad Bp = \begin{bmatrix}0\\ 0\\ 15\\ 0\end{bmatrix} \quad Cp = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \quad Dp = \begin{bmatrix}0\\ 0\\ 0\\ 0\end{bmatrix}$$

LQR

- The first design was with an identity matrix 4x4 and R=1.

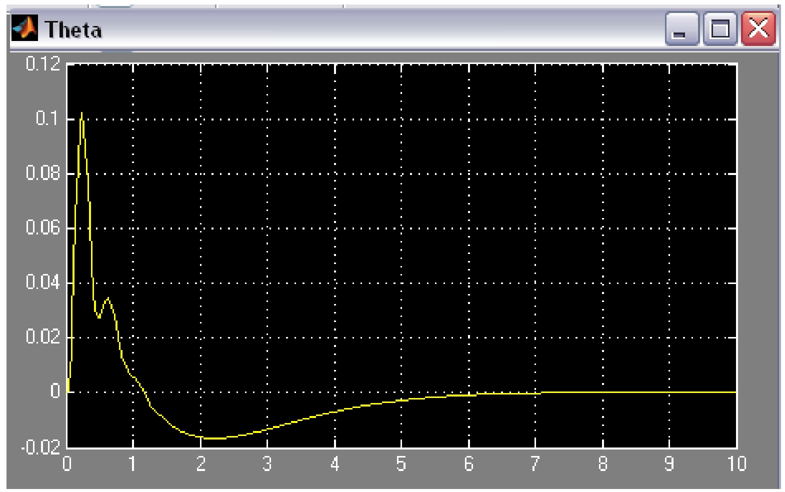


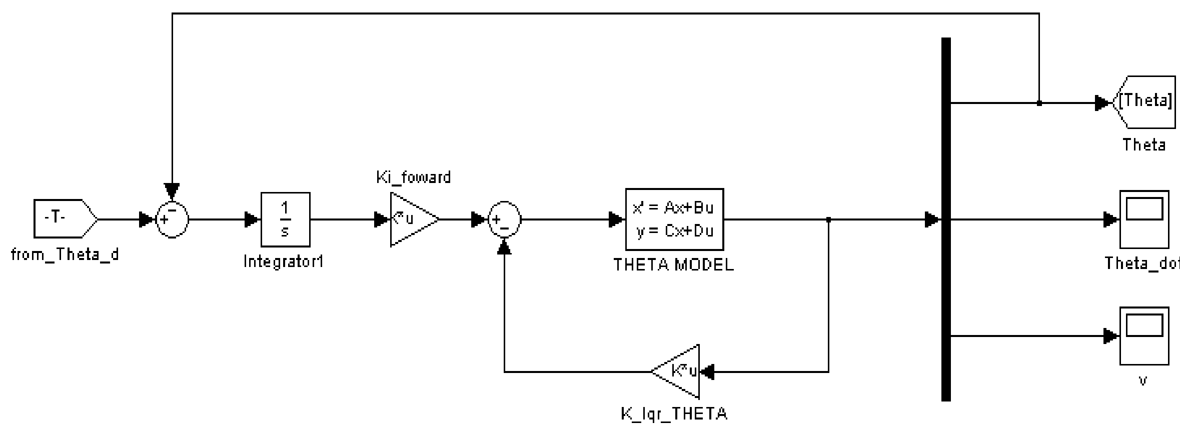
- The response in pitch is not stable. We need to play with matrix Q and R to improve the response. The final values for Q and R :

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & .05 & 0 & 0 \\ 0 & 0 & .05 & 0 \\ 0 & 0 & 0 & 20000 \end{bmatrix} \qquad R = .05$$

LQR

- The final response on pith and simulink model :





LQR

YAW

$$\begin{bmatrix} \dot{\varphi} \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{Ky}{Jyaw} \end{bmatrix}$$

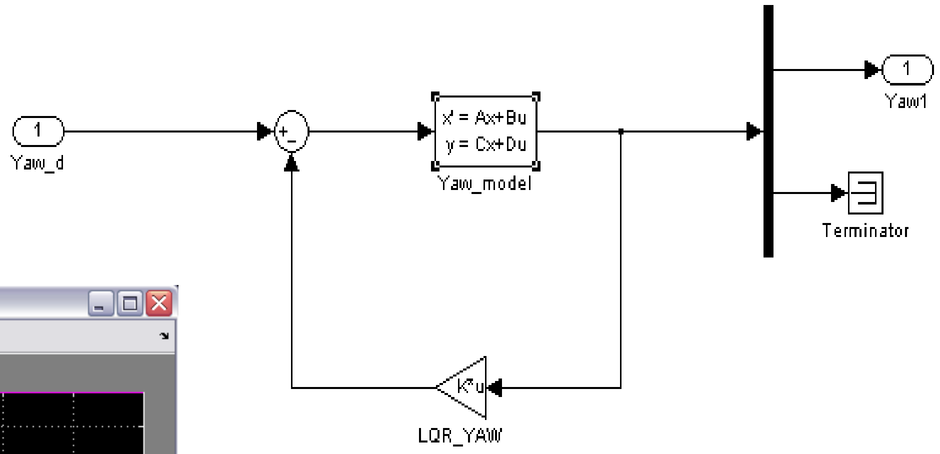
- Ky and Jyaw is taken from Lab manual (Ky=4Nm & Jyaw=0,4 Kg m^2)

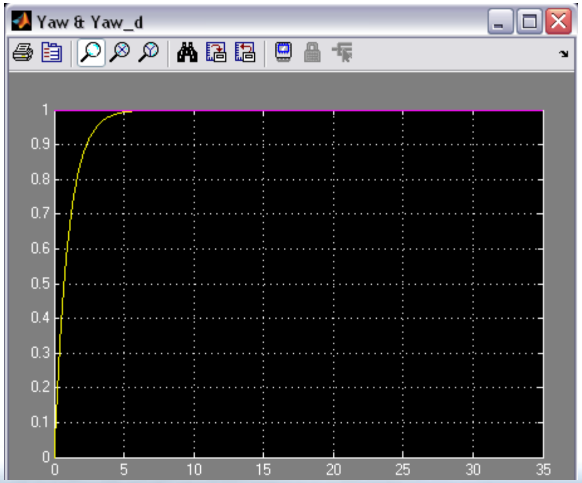
$$\begin{bmatrix} \dot{\varphi} \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 100 \end{bmatrix}$$

$$Ayaw = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad Byaw = \begin{bmatrix} 0 \\ 100 \end{bmatrix} \quad Cyaw = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad Dyaw = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

LQR

- The K gain matrix is calculated with *'lqr'* matlab function and the Q matrix is a diagonal matrix of 1 (2X2) and R is 1.
- K=lqr(Ayaw,Byaw,Q,1)   K =[1.0000 1.0100]

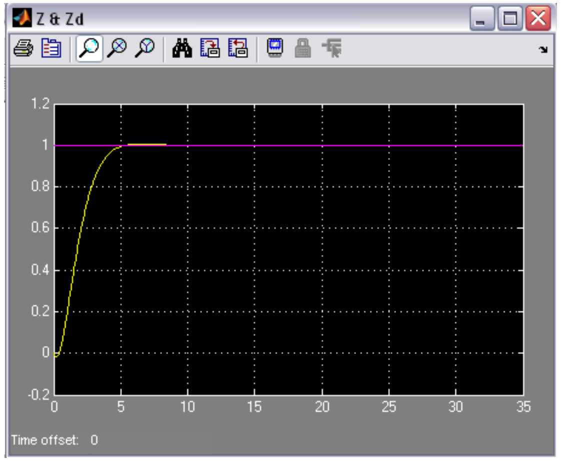


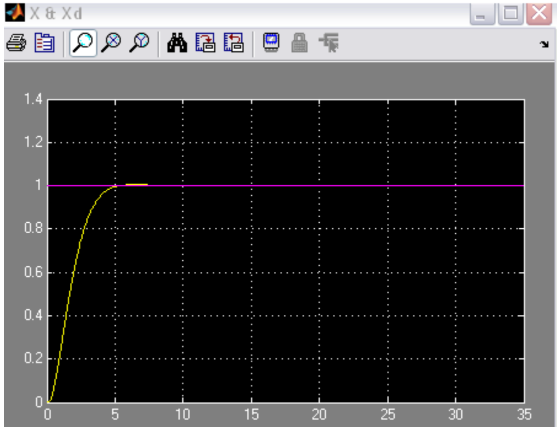


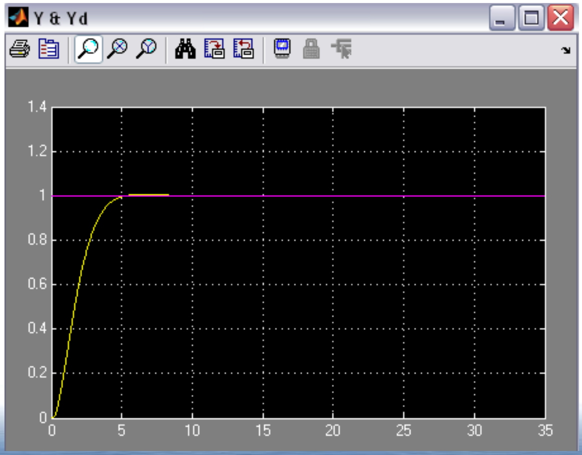
LQR

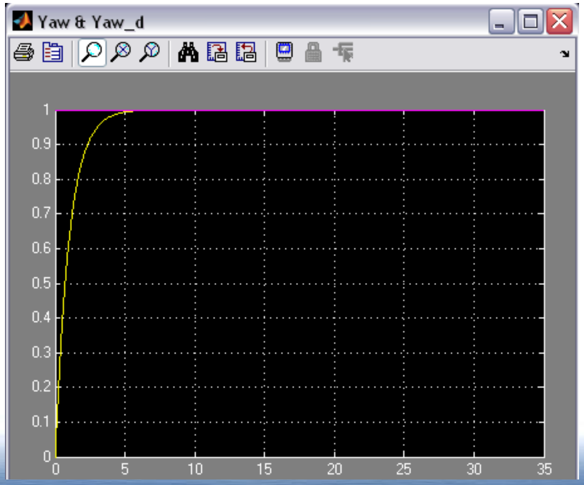
LQR OUTPUTS RESULTS

- For Zd= 1   Xd=1   Yd=1   Yawd=1





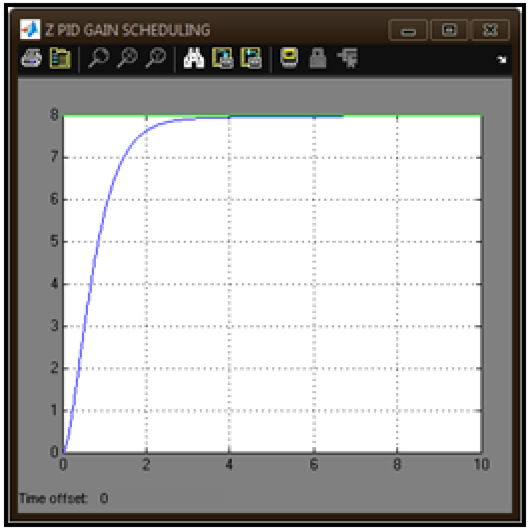


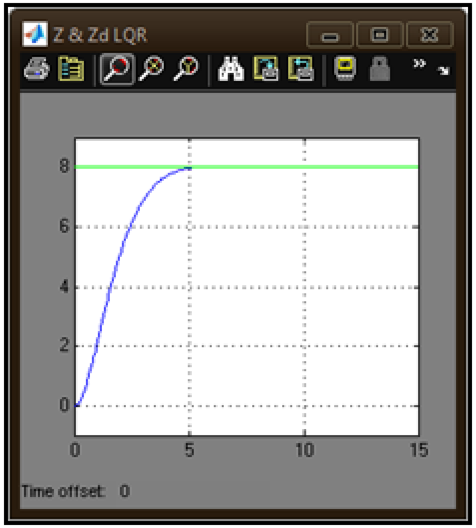


Compare Control Methods

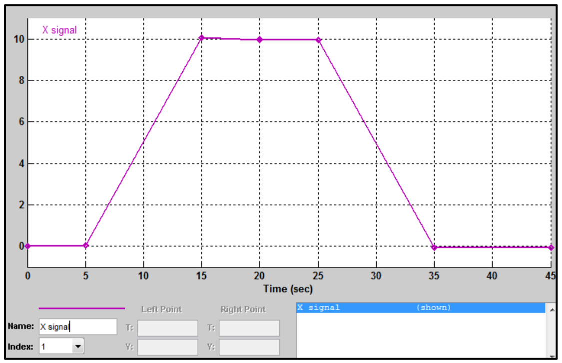
PID
Start with Guess
Range Limit
Controllability not known

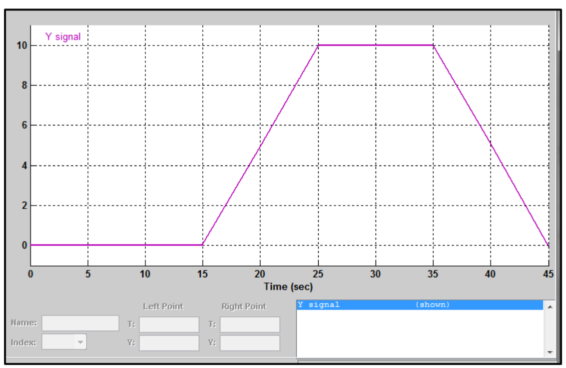
LQR
Modeled to System Dynamics
No Range Limit
Controllability Known

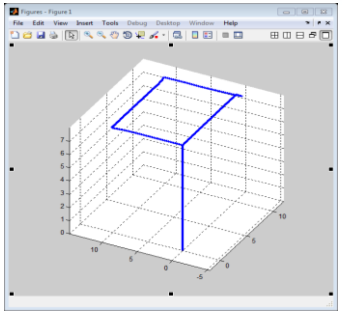




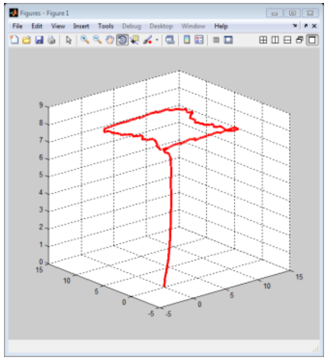
Trajectory Tracking



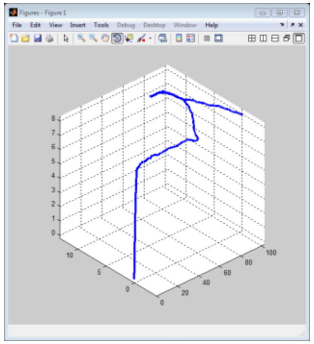




No Noise

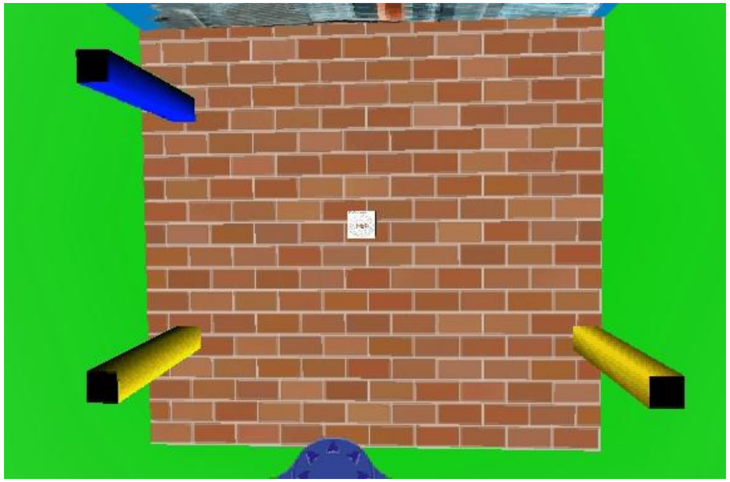


0.3 Noise



1 Noise

Trajectory Tracking

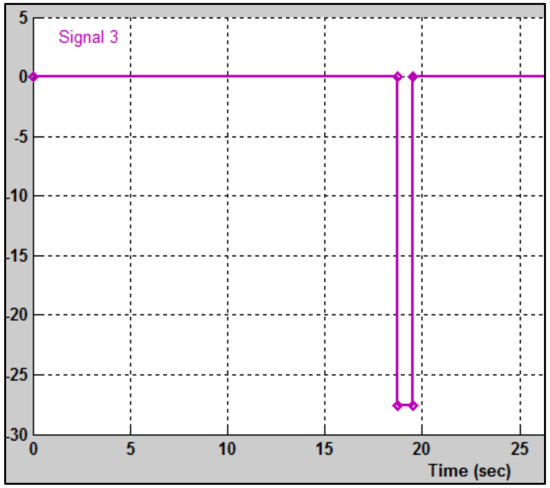


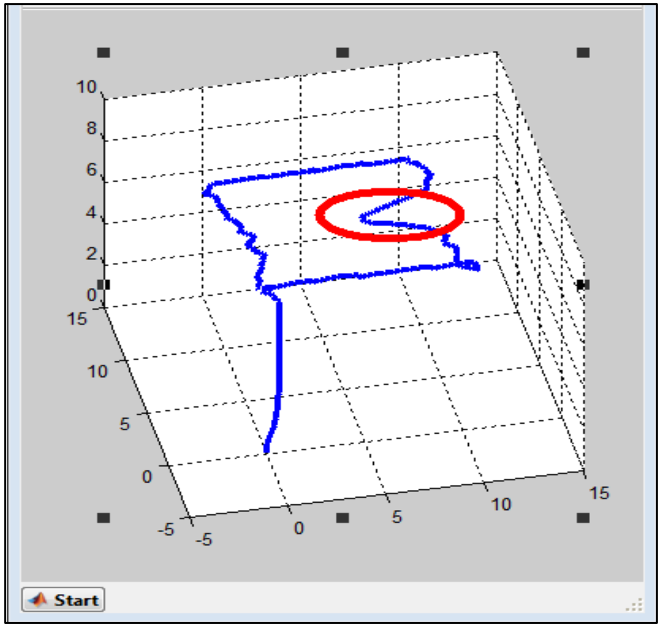
Top View

Front View

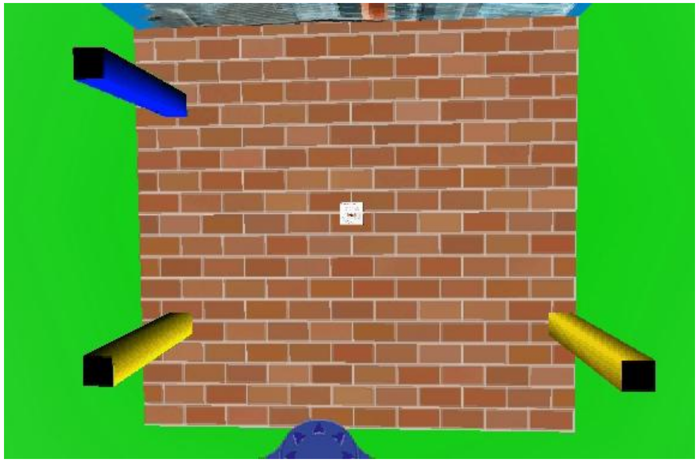
Disturbance

Negative Pulse

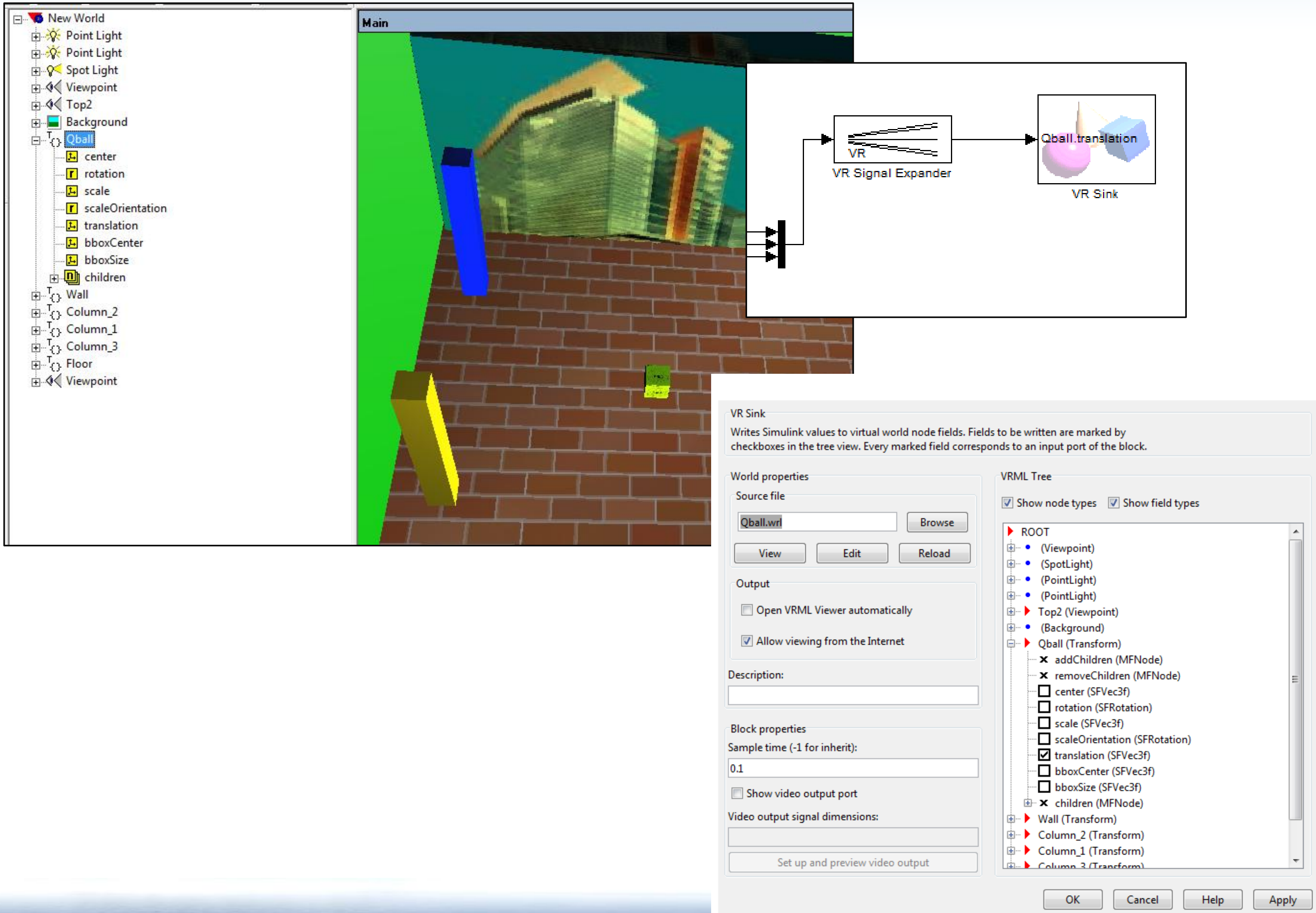




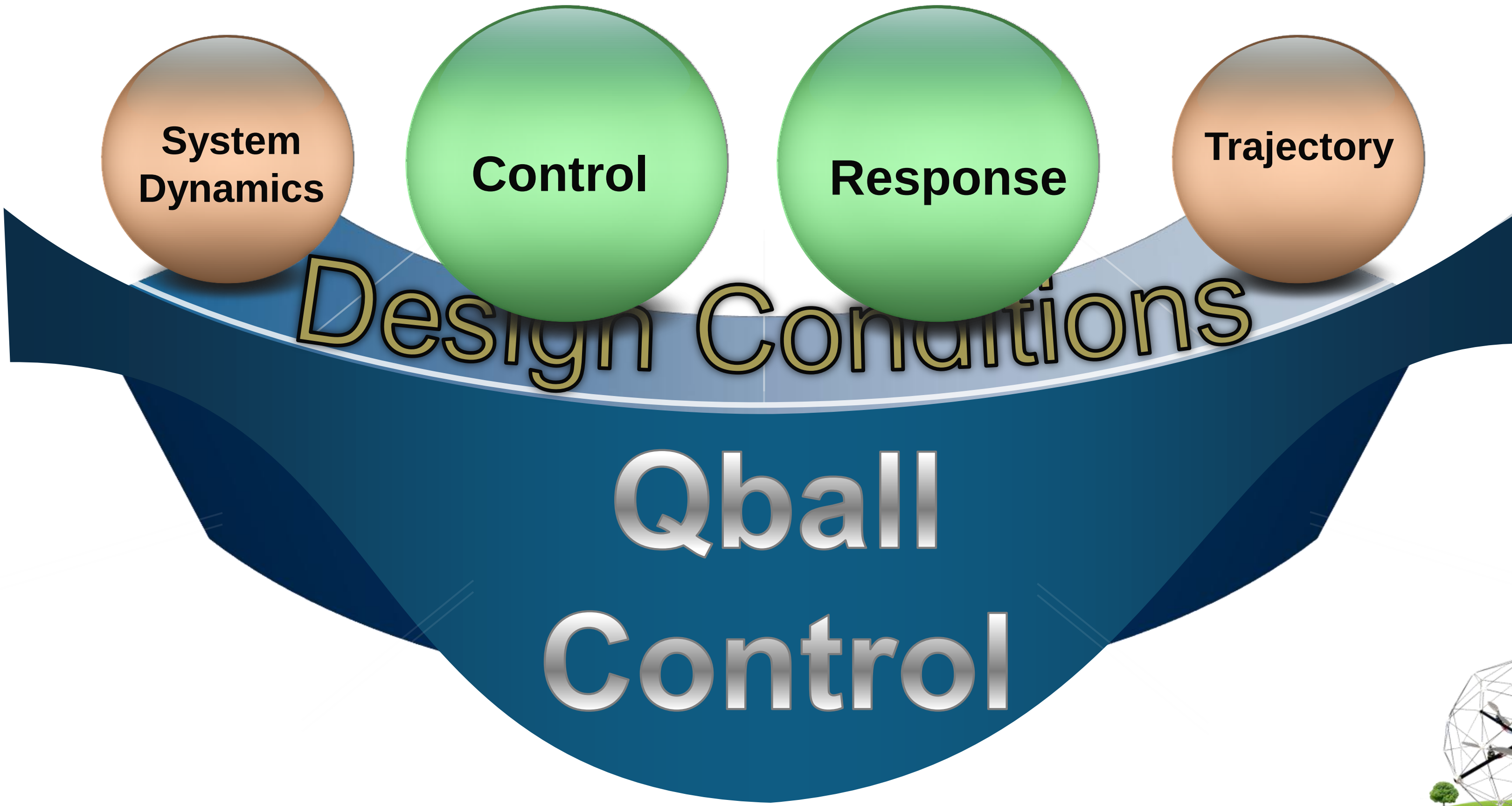
Disturbance



3D Build


New World
Point Light
Point Light
Spot Light
Viewpoint
Top2
Background
Qball
center
rotation
scale
scaleOrientation
translation
bboxCenter
bboxSize
children
Wall
Column_2
Column_1
Column_3
Floor
Viewpoint
Main
VR
VR Signal Expander
Qball.translation
VR Sink
VR Sink
Writes Simulink values to virtual world node fields. Fields to be written are marked by checkboxes in the tree view. Every marked field corresponds to an input port of the block.
World properties
Source file
Qball.wrl
Browse
View
Edit
Reload
Output
Open VRML Viewer automatically
Allow viewing from the Internet
Description:
Block properties
Sample time (-1 for inherit):
0.1
Show video output port
Video output signal dimensions:
Set up and preview video output
VRML Tree
Show node types
Show field types
ROOT
(Viewpoint)
(SpotLight)
(PointLight)
(PointLight)
Top2 (Viewpoint)
(Background)
Qball (Transform)
addChildren (MFNode)
removeChildren (MFNode)
center (SFVec3f)
rotation (SFRotation)
scale (SFVec3f)
scaleOrientation (SFRotation)
translation (SFVec3f)
bboxCenter (SFVec3f)
bboxSize (SFVec3f)
children (MFNode)
Wall (Transform)
Column_2 (Transform)
Column_1 (Transform)
OK
Cancel
Help
Apply

Conclusion


System Dynamics
Control
Response
Trajectory
Design Conditions
Qball Control

Questions?