Flight Control Systems:

Qball Quadrotor Helicopter

Beikrit Samia
Falaschini Clara
Abdolhosseini Mahyar
Capotescu Florin

Introduction

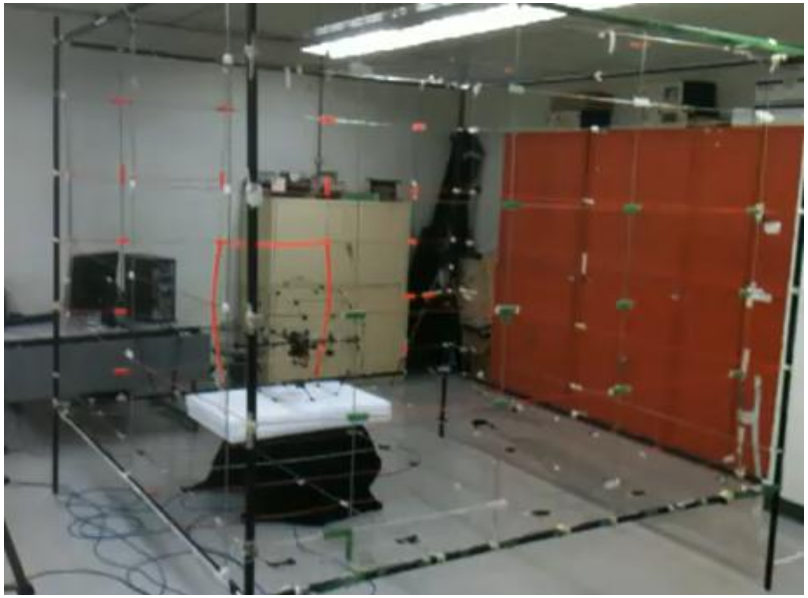
- Flight Control Systems Project :
  - Objectives for the Qball quadrotor helicopter

  - 1) Develop non linear and linearized matematical models
  - 2) Design an autopilot
  - 3) Implement, test and analyze our controller

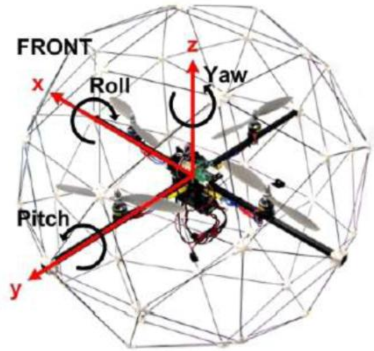
- Our project :
  - Develop models on Simulink
  - Create a PID Controller to control the altitude
  - Create a LQR Controller to control the altitude

Table of Contents

- Qball Systems
- PID Controller
- LQR Controller
- Other development



Qball System



- Quadrotor system :
  - unmanned helicopter with 4 horizontal fixed rotors designed in a square, symmetric configuration
  - According to dynamics equations :

$$\ddot{x} = (sin\psi sin\phi + cos\psi sin\theta cos\phi)\frac{U_1}{M} \qquad \ddot{\phi} = \frac{U_2}{J_{xx}}$$

$$\ddot{y} = (sin\psi sin\theta cos\phi + cos\psi sin\phi)\frac{U_1}{M} \qquad \ddot{\theta} = \frac{U_3}{J_{yy}}$$

$$\ddot{z} = -g + (cos\theta cos\phi)\frac{U_1}{M} \qquad \ddot{\psi} = \frac{U_4}{J_{zz}}$$

  - We used these equations to create our simulink model

Qball System

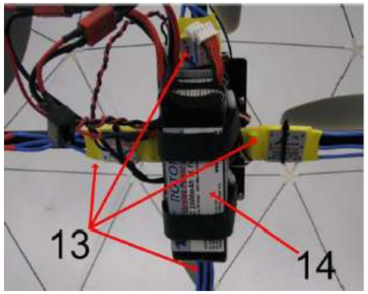
- We simplified it to control only the altitude

$$\ddot{z} = -g + (cos\theta cos\phi)\frac{U_1}{M}$$

- So the model is only reduced to control Z

Qball Systems

- To make test on real Qball system, some installation was required :
  - Battery
  - Cameras
  - QuaRC/Simulink Model



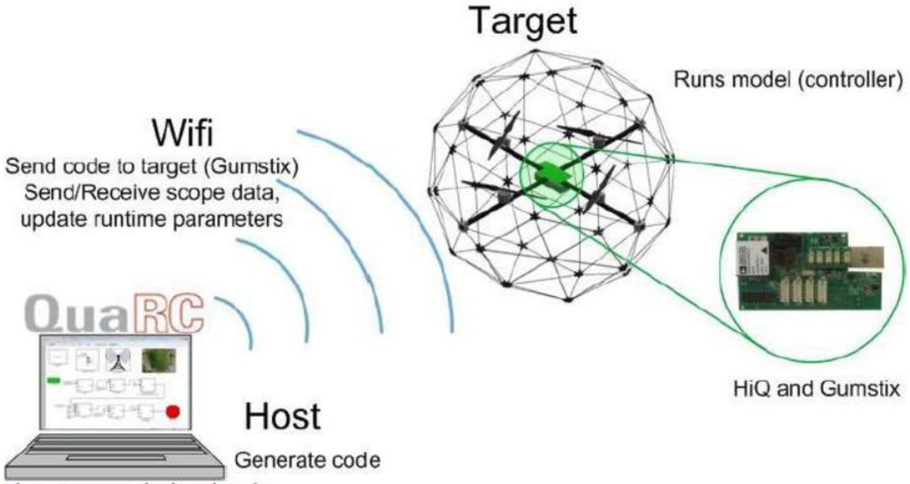


Figure 2: Communication Hierarchy

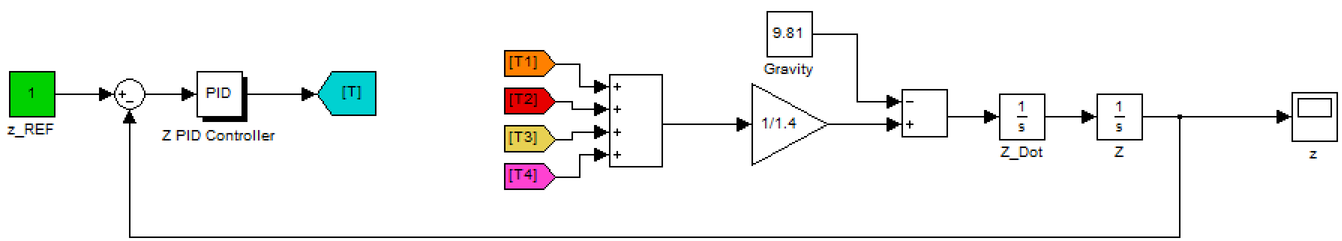
PID Controller

- In this section some assumptions are made to make the equations of motion of the plant (QBall) 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

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



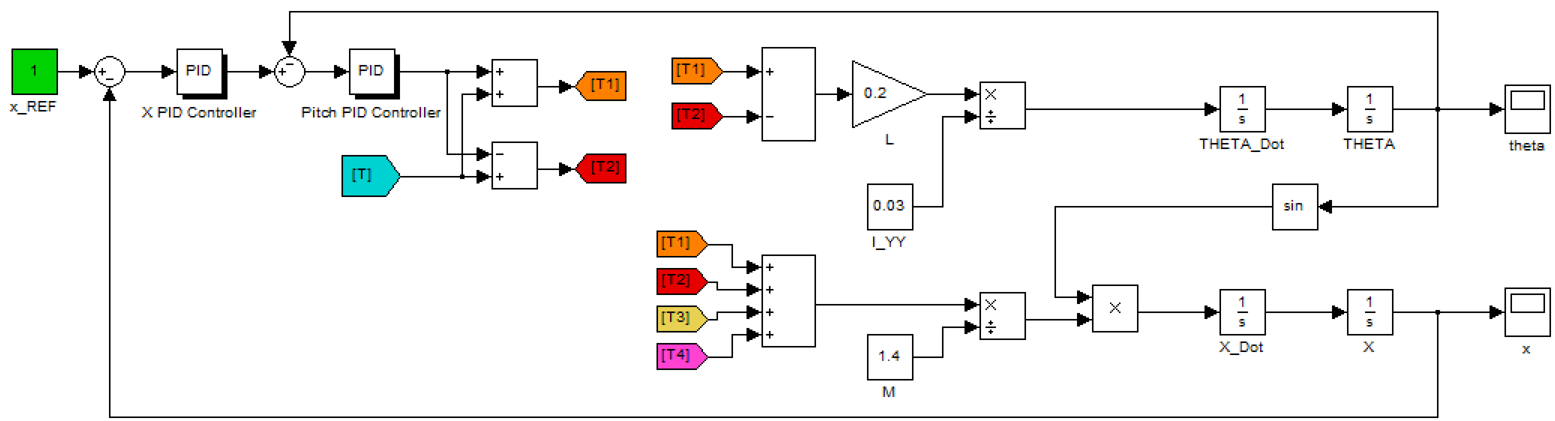
It should be notified that this modeling is valid as long as the Yaw Angle is automatically controlled to be zero.

PID Controller

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

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

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

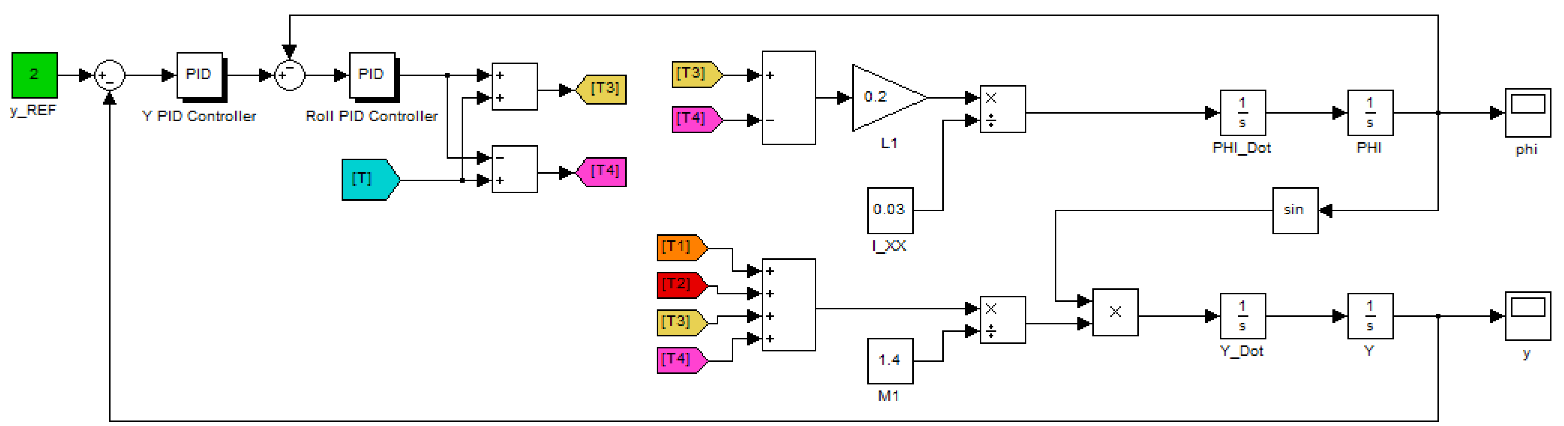


[(T1−ΔT) + (T2+ΔT) + T3 + T4] = [T1 + T2 + T3 + T4]

PID Controller

- Side Motion along the Y Axis Coupled with Rolling Motion

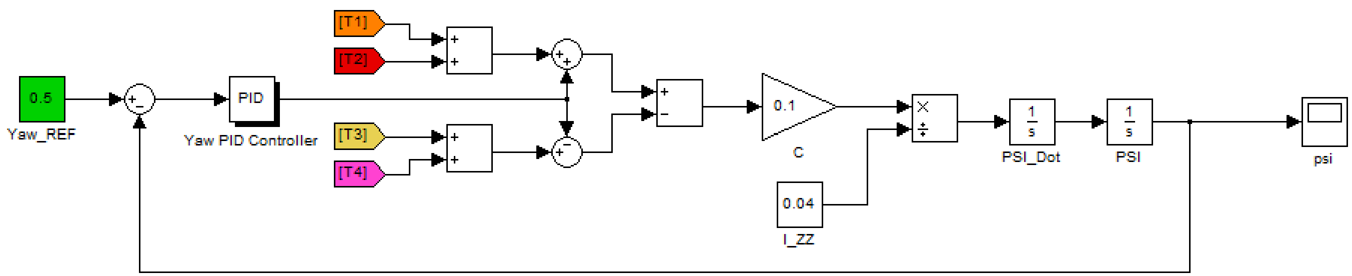
$$M\ddot{Y} = (T1 + T2 + T3 + T4)\varphi$$
$$J_{xx}\ddot{\varphi} = (T3 - T4)L$$



PID Controller

- **Pure Yawing Motion**

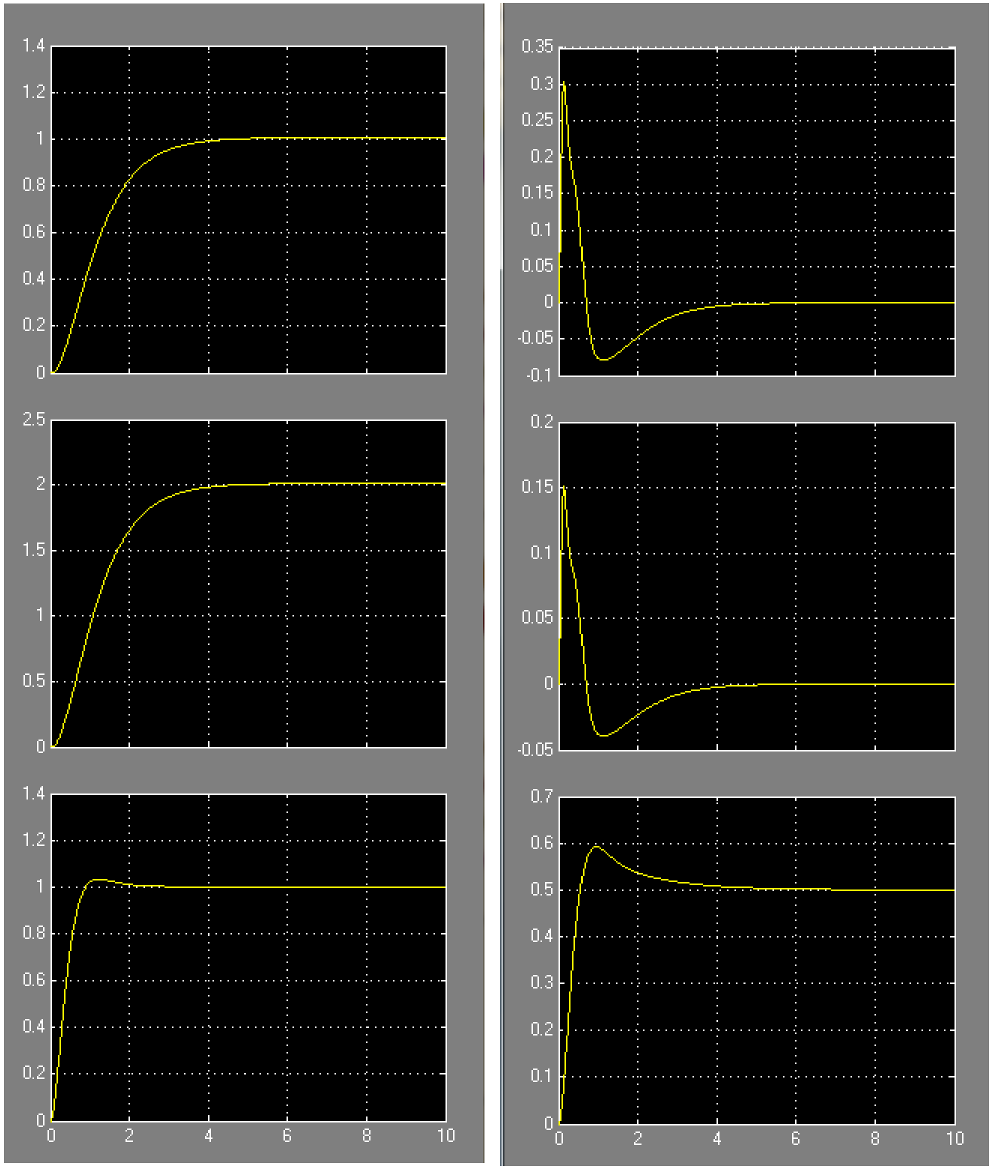
$$J_{zz}\ddot{\psi} = (T1 + T2 - T3 - T4)C$$



✓ 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 Controller Tuning

- Ziegler/Nichols tuning method
  - *The main idea is increasing the proportional gain Kp until it reaches the ultimate gain Ku at which the output control loop begins to oscillate with constant amplitude. Then, Ku and the oscillation period Tu are used to tune the other gains : integration gain and derivative.*

Ziegler–Nichols tuning method	K_p	K_i	K_d
PID controller	$0.6K_u$	$2K_p/T_u$	$K_pT_u/8$

PID Controller

- Gains value obtained for Qball system :

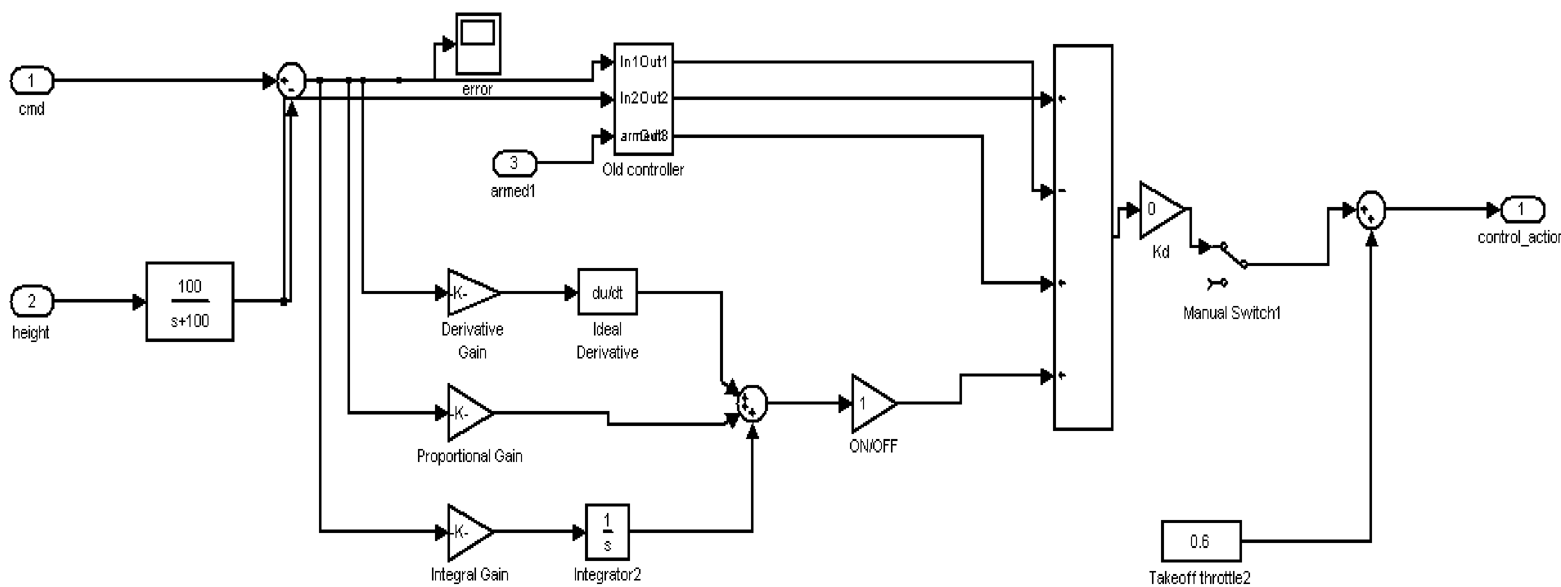
$$K_p = 0{,}0068$$

$$K_i = 0{,}003$$

$$K_d = 0{,}0063$$

PID Controller

- PID controller implemented in the real system



LQR
(Linear Quadratic Regulator)

- Optimal Control, is an area within the theory of control that deals with control of dynamic systems in a way that one specific, designer-defined function is minimized.
- Specifically speaking, the case in which the dynamics of the system is governed by a set of linear differential equations of motion and the cost function is described by a quadratic function, is called Linear Quadratic problem (LQ Problem).
- . This specific, designer-defined function is also known as "Cost Function".

$$J = \frac{1}{2}\int_0^{\infty} (x^T Q x + u^T R u)\, dt$$

LQR State Feedback Design

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

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

- Assume that all the states are available for measurement:

- Now, one can design a State Variable Feedback Control as:

$$u = -Kx + v$$

- By substituting equation 2 in equation 1 the state space representation of the closed loop system becomes:

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

LQR

- For such complex systems the Achermann's formula inconvenient for determination of all closed loops of the system:

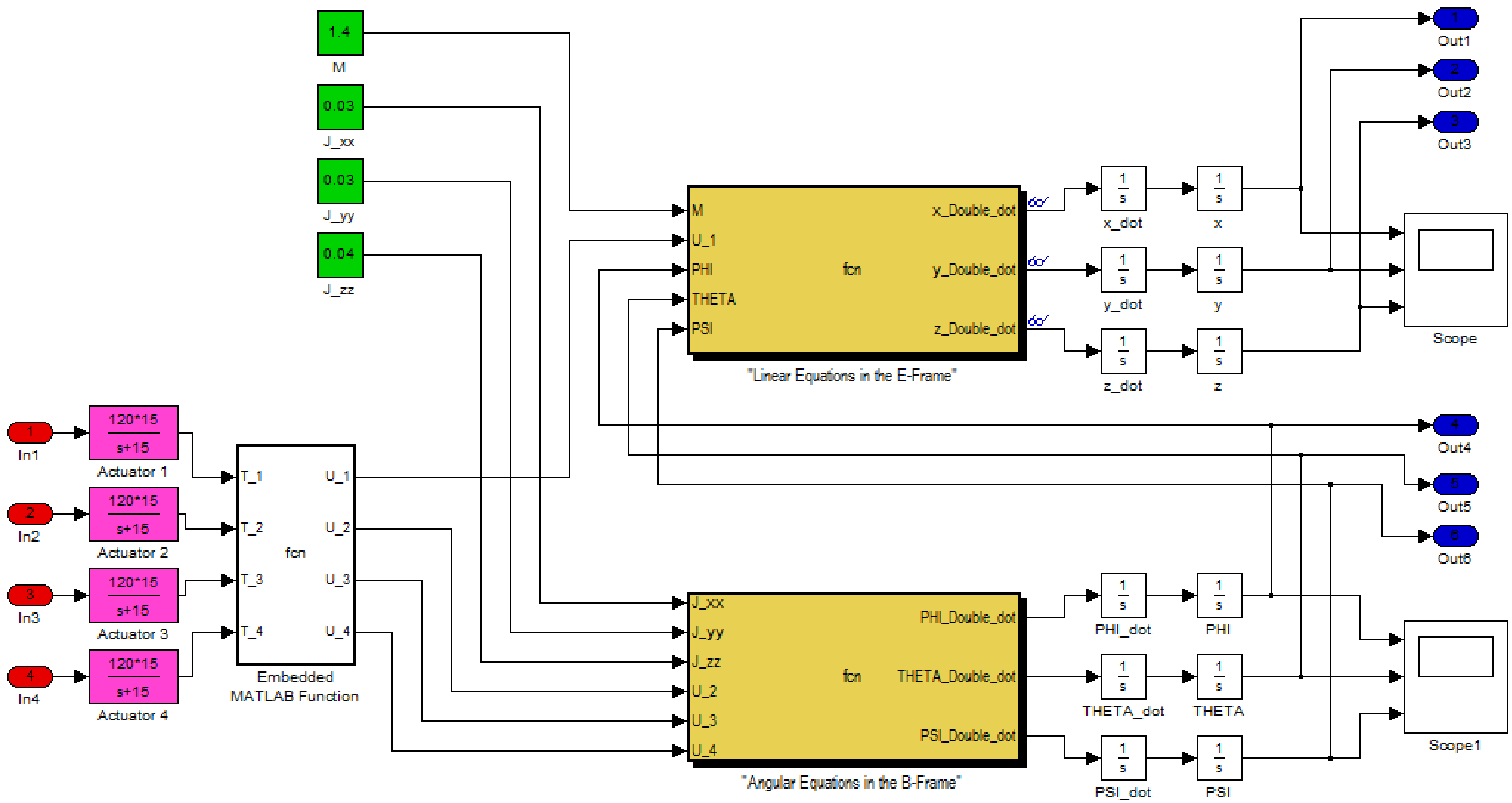
$$J = \frac{1}{2}\int_0^\infty (x^T Q x + u^T R u)\, dt$$

- As this equation suggests there are two design parameters Q and R that should be decided on prior to design.
- These Q and R are Weighting Factors and they have significance.
- For the time being, let's assume that the input v is equal to zero and our only concern is stability of the system rather than following a specific reference input.

LQR

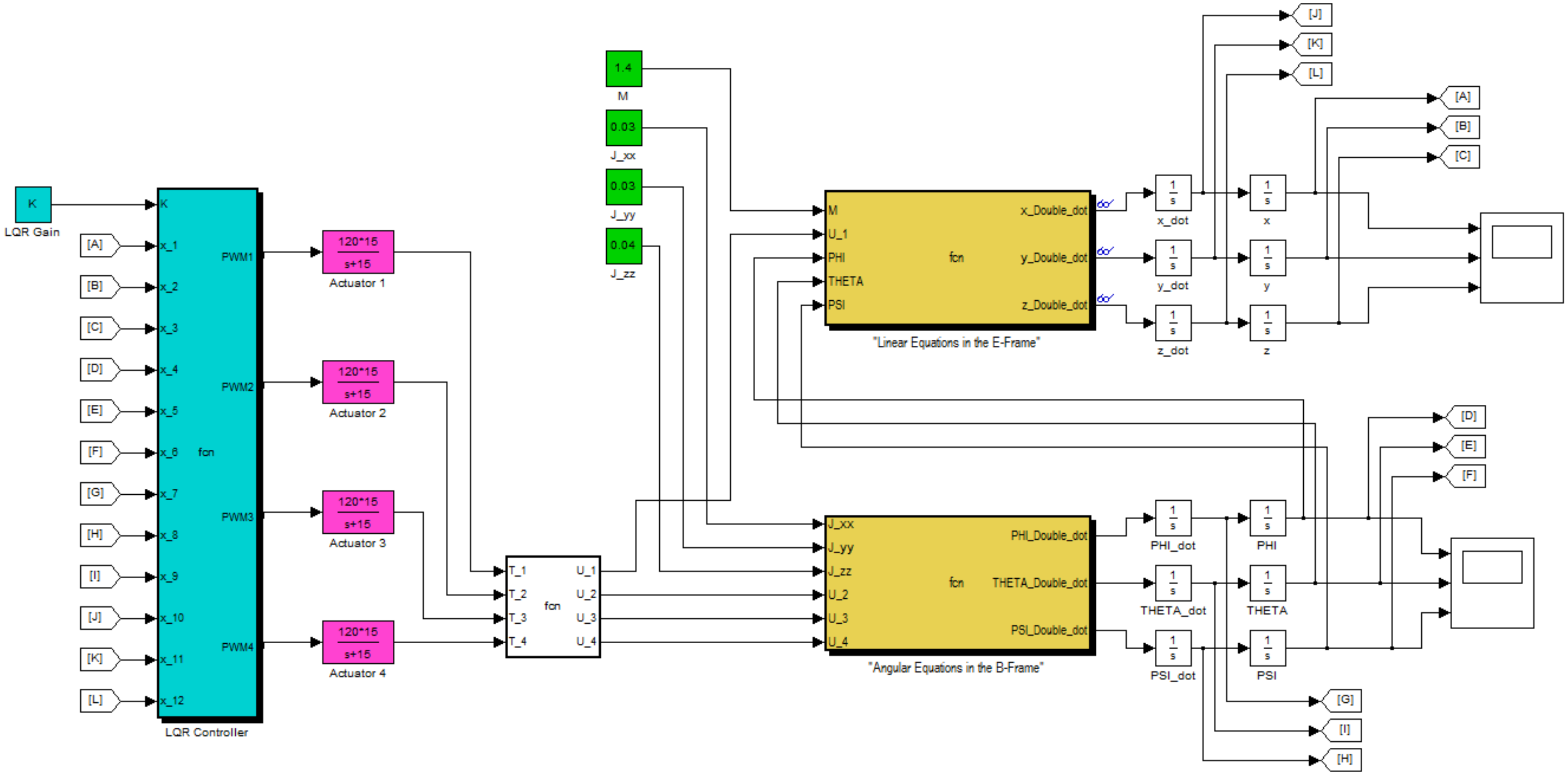
- It is worthy of attention that the LQR design procedure for solving this optimization problem is guaranteed to produce a feedback that stabilizes the system as long as the studied system is reachable or observable.
- As it was mentioned earlier, one of the drawbacks of LQR controller is its limited applicability to just linear systems.
- Later on, this trimmed point will be feed into the command “linmod” as one of its arguments.
- A 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.

Design Process for the Non-tracker Problem



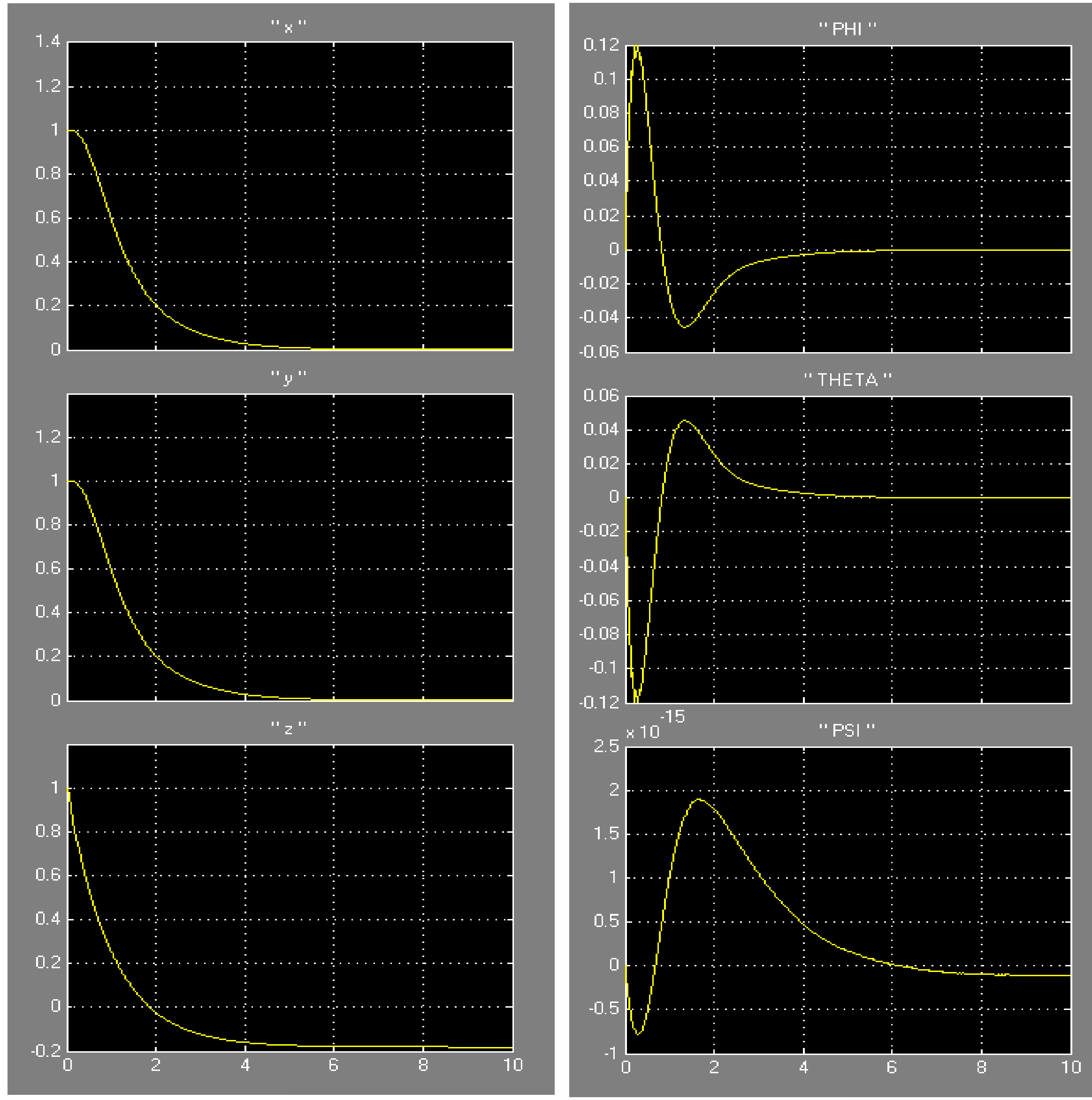
LQR

- If we put together the developed controller and the plant, hereafter is the control system;



LQR

Having chosen the values of these weighting matrices to be Q = eye(12) and R = eye(4), bellow you can find the time response of the system.



Time Response of the System

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:

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

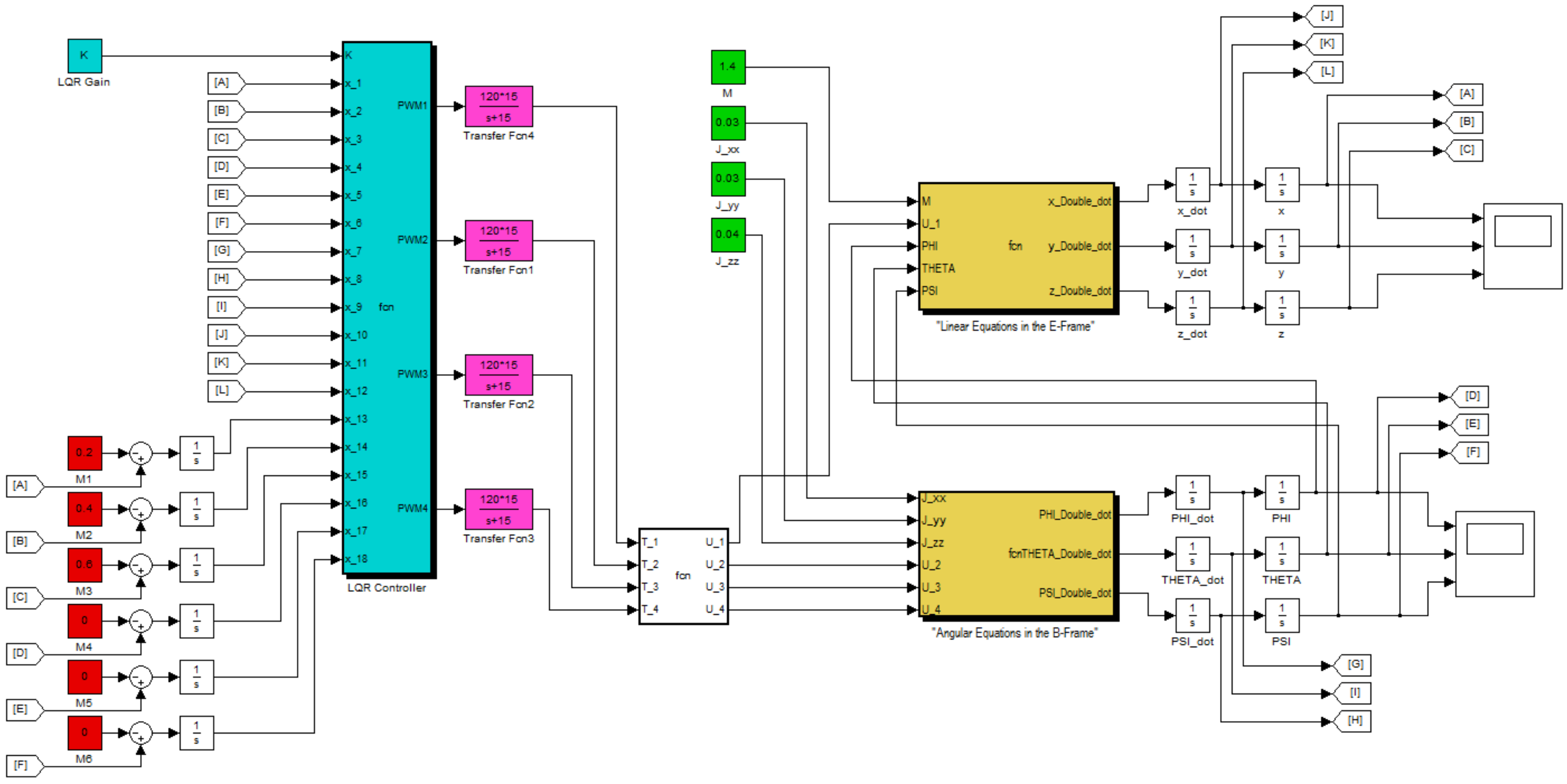
Or in a more compact form:

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

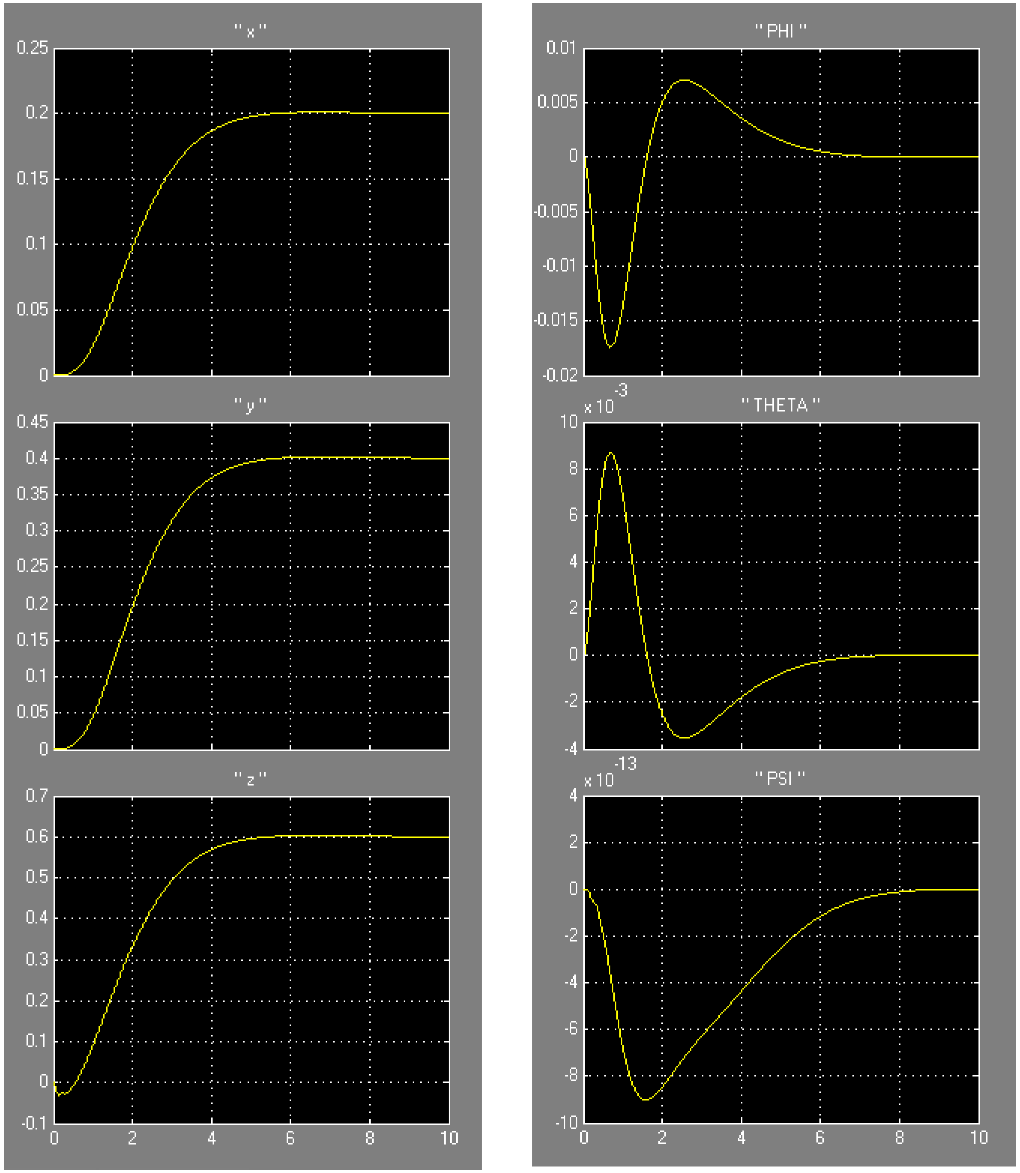
Again, once the state space representation of the control system is obtained, design of LQR Controller is almost straight forward.

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), bellow you can find the time response of the system for the following reference inputs.



Time Response of the System

Other interesting approaches

- Altitude control and Trajectory Tracking :
  - Basic commands added to Quanser controller to track a square
- Trajectory Tracking and Heading control:
  - Normally an aircraft will change its yaw angle when tracking a trajectory.
  - This approach needs an implementation of a coupled nonlinear controller for both yaw motion and altitude control.
- Sliding Mode Control (Robust control):
  - Tentative of control the altitude (works in Simulink)

Conclusion

- We developed 2 controllers : PID and LQR
- First by Simulink and then we test it in real
- We met some problems :
  - Differences between the simulink and real system
  - Batteries problems

- Really Good Experience to work on real system and deal with all what could happened