Table 1: Parameters of the quadrotor

Name	Symbol	Units	Value
Body inertia x axis	I_{xx}	[kg·m^2]	0.0117
Body inertia y axis	I_{yy}	[kg·m^2]	0.0117
Body inertia z axis	I_{zz}	[kg·m^2]	0.00234
Rotor inertia	J_R	[kg·m^2]	0.00005
Distance propeller axis to CoG	L	[m]	0.18
Total mass	m	[kg]	0.478
Thrust factor	b	[N/(rpm)^2]	6.11×10^{-8}
Drag factor	d	[N·m/(rpm)^2]	1.5×10^{-9}

1 Quadrotor parameters

In Table 1, the parameters values for the quadrotor that is part of the MARHES Lab are summarized. These values will be used in SIMULINK to implement the model of the quadrotor and its controllers for simulation.

2 System Control

As result of the previous section, the differential equations that describe the model of the Quad-Rotor were obtained. However, these equations that formulate the dynamics of the system are non-linear. Since the goal is to design linear controller for the Quad-Rotor, these equation system must be linearized around an operating point. Therefore, a general linearization is done with the purpose of describing the model as a linear state-space model.

2.1 Linear Model of the Quadrotor

The general linearization applied here is calculated by using Jacobians of the nonlinear state equations with respect to the states and the inputs around operating conditions, in this case the hovering condition. Some similar approaches are developed in [1, 2, 3, 4].

One possible simplification to the quadrotor model is related with the angular accelerations that are referred to the angles of the quadrotor measured in the body frame [1, 2]; these are $[\dot{p}\ \dot{q}\ \dot{r}]$. Indeed, these are not equal to the acceleration of the Euler angles which determines the attitude in the earth frame. Furthermore, the transfer matrix ${}^{B}T_{E}$ defines the relation between the angular velocities in the earth frame and those ones in the body-fixed frame. When the hovering condition is considered, it is close to the identity matrix. Consequently, the acceleration equations have been referred directly to the Euler angle accelerations. This means that one can write $[\dot{\phi}\ \dot{\theta}\ \dot{\psi}] \approx [p\ q\ r]$, so the differential equations are now:

$$
\begin{aligned}
\ddot{x} &= (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)\,\frac{1}{m}U_1, \\
\ddot{y} &= (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)\,\frac{1}{m}U_1, \\
\ddot{z} &= -g + (\cos\phi \cos\theta)\,\frac{1}{m}U_1, \\
\ddot{\phi} &= \left(\frac{I_{yy} - I_{zz}}{I_{xx}}\right)\dot{\theta}\dot{\psi} - \frac{J_R}{I_{xx}}\dot{\theta}\Omega + \frac{L}{I_{xx}}U_2, \\
\ddot{\theta} &= \left(\frac{I_{zz} - I_{xx}}{I_{yy}}\right)\dot{\phi}\dot{\psi} - \frac{J_R}{I_{xx}}\dot{\phi}\Omega + \frac{L}{I_{xx}}U_3, \\
\ddot{\psi} &= \left(\frac{I_{xx} - I_{yy}}{I_{zz}}\right)\dot{\phi}\dot{\theta} + \frac{1}{I_{zz}}U_4,
\end{aligned}
$$

Before finding the linear model, it is required to rewrite the model differential equations in state space form. The state vector, $\mathbf{X}$, is chosen as

$$
\begin{aligned}
\mathbf{X} &= \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} & \phi & \dot{\phi} & \theta & \dot{\phi} & \psi & \dot{\psi} \end{bmatrix}^T, \\
&= [x_1\ \ x_2\ \ x_3\ \ x_4\ \ x_5\ \ x_6\ \ x_7\ \ x_8\ \ x_9\ \ x_{10}\ \ x_{11}\ \ x_{12}]^T
\end{aligned} \tag{1}
$$

then, using the previous differential equations, the state representation $\dot{\mathbf{X}} =$

$f\left(\mathbf{X}, \Omega_1, \Omega_2, \Omega_3, \Omega_4\right)$, is:

$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \left(\cos x_7 \sin x_9 \cos x_{11} + \sin x_7 \sin x_{11}\right) \frac{1}{m} U_1, \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= \left(\cos x_7 \sin x_9 \sin x_{11} - \sin x_7 \cos x_{11}\right) \frac{1}{m} U_1, \\
\dot{x}_5 &= x_6 \\
\dot{x}_6 &= -g + \left(\cos x_7 \cos x_9\right) \frac{1}{m} U_1, \\
\dot{x}_7 &= x_8 \\
\dot{x}_8 &= \left(\frac{I_{yy} - I_{zz}}{I_{xx}}\right) x_{10} x_{12} - \frac{J_R}{I_{xx}} x_{10} \Omega + \frac{L}{I_{xx}} U_2, \\
\dot{x}_9 &= x_{10} \\
\dot{x}_{10} &= \left(\frac{I_{zz} - I_{xx}}{I_{yy}}\right) x_8 x_{12} - \frac{J_R}{I_{xx}} x_8 \Omega + \frac{L}{I_{xx}} U_3, \\
\dot{x}_{11} &= x_{12} \\
\dot{x}_{12} &= \left(\frac{I_{xx} - I_{yy}}{I_{zz}}\right) x_8 x_{10} + \frac{1}{I_{zz}} U_4,
\end{aligned}
$$

where $\Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4$, and

$$
\begin{aligned}
U_1 &= b\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right), \\
U_2 &= b\left(\Omega_2^2 - \Omega_4^2\right), \\
U_3 &= b\left(\Omega_3^2 - \Omega_1^2\right), \\
U_4 &= d\left(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right).
\end{aligned}
$$

At the hovering condition, all the propellers have the same speed to counterbalance the acceleration due to gravity and the orientation of the Quad-Rotor is zero for all its angles. Consequently, $\Omega_1 = \Omega_2 = \Omega_3 = \Omega_4$ and its value can be calculated by

$$
\Omega_1 = \Omega_2 = \Omega_3 = \Omega_4 = \sqrt{\frac{mg}{4b}} = 4380 \text{ rpm}.
$$

Linearizing around the hovering point [1, 2], it is possible to obtain a linear state-space model for the quadrotor

$$\dot{\mathbf{X}} = \mathbf{AX} + \mathbf{BU}$$
$$\mathbf{Y} = \mathbf{CX}$$

with $\mathbf{X}$ defined in (1), $\mathbf{U} = [\Omega_1 \ \ \Omega_2 \ \ \Omega_3 \ \ \Omega_4]^T$, $\mathbf{C} = \mathbf{I}_{12\times 12}$,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9.8089 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -9.8089 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \text{ and}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.001119 & 0.001119 & 0.001119 & 0.001119 \\ 0 & 0 & 0 & 0 \\ 0 & 0.008234 & 0 & -0.008234 \\ 0 & 0 & 0 & 0 \\ -0.008234 & 0 & 0.008234 & 0 \\ 0 & 0 & 0 & 0 \\ -0.005615 & 0.005615 & -0.005615 & 0.005615 \end{bmatrix}.$$

2.2 LQR for Attitude Stabilization

Once a linear model of the Quad-Rotor is obtained, the first goal is the stabilization of the attitude (stabilization of the Euler angles). In order to accomplish this goal, an Linear Quadratic Regulator [5], LQR, will be designed in this section.

An LQR (Linear Quadratic Regulator) is a widely used optimal MIMO control technique. Moreover, its formulation considers that given the system dynamics

$$\dot{\mathbf{X}} = \mathbf{AX} + \mathbf{BU}$$

and a performance objective of the form

$$V = \int_0^\infty \left(\mathbf{X}^T\mathbf{QX} + \mathbf{U}^T\mathbf{RU}\right) dt$$

the state feedback control law that minimizes the performance measure V is given by

$$\mathbf{U} = -\mathbf{KX}$$

where $\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\bar{\mathbf{P}}$ and $\bar{\mathbf{P}}$ is the solution to the Algebraic Riccati Equation

$$\mathbf{0} = \mathbf{A}^T\bar{\mathbf{P}} + \bar{\mathbf{P}}\mathbf{A} + \mathbf{Q} - \bar{\mathbf{P}}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\bar{\mathbf{P}}$$

Furthermore, $\mathbf{Q}$ and $\mathbf{R}$ must be symmetric positive definite matrices, the pair $(\mathbf{A}, \mathbf{B})$ must be controllable and the pair $\left(\mathbf{A}, \sqrt{\mathbf{Q}}\right)$ must be observable in order that a solution to this steady-state LQR problem exists. The value of the weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ are

$$\mathbf{Q} = 200\mathbf{I}_{12\times 12} \text{ and } \mathbf{R} = 0.1\mathbf{I}_{4\times 4}.$$

These values are the ones that give better results after many tries using the simulation. The corresponding K obtained in MATLAB is

$$\mathbf{K} = \left[\begin{matrix} -31.6228 & -67.8748 & 0 & 0 & 22.3607 & 102.4274 \\ 0 & 0 & -31.6228 & -67.8748 & 22.3607 & 102.4274 \\ 31.6228 & 67.8748 & 0 & 0 & 22.3607 & 102.4274 \\ 0 & 0 & 31.6228 & 67.8748 & 22.3607 & 102.4274 \end{matrix}\right.$$
$$\left.\begin{matrix} 0 & 0 & -559.4168 & -262.5640 & -22.3607 & -49.9115 \\ 559.4168 & 262.5640 & 0 & 0 & 22.3607 & 49.9115 \\ 0 & 0 & 559.4168 & 262.5640 & -22.3607 & -49.9115 \\ -559.4168 & -262.5640 & 0 & 0 & 22.3607 & 49.9115 \end{matrix}\right]$$

2.3 LQR for Attitude Stabilization and Height Control

In this case, it is desired to stabilized the attitude plus maintain the quadrotor in a required altitude, z-position = a constant position z_o. For that reason, the desired state is

$$\tilde{\xi} = [0 \;\; 0 \;\; 0 \;\; 0 \;\; z_o \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]^T$$

while the system state ξ is given by

$$\dot{\xi} = \mathbf{A}\xi + \mathbf{B}\mathbf{U}$$

The ideal state is no more the zero state, so this is a tracking problem. Checking the value of $\omega = \mathbf{A}\tilde{\xi} - \dot{\tilde{\xi}}$, this value is going to be zero. As a result, the tracking problem can be reduced to a regulator problem [5]. In fact, it is the same regulator problem solved in the previous section just that know the optimal feedback control law is simply

$$\mathbf{U} = -\mathbf{K}\mathbf{X} = -\mathbf{K}\left(\xi - \tilde{\xi}\right)$$

2.4 LQR for tracking (X and Y)

Now, it is desired that the quadrotor follows a trajectory. This problem is known as the tracking problem. In particular, the desired state is given by

$$\tilde{\xi} = [v_x t + x_o \;\; v_x \;\; v_y t + y_o \;\; v_y \;\; z_o \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]^T$$

where v_x, v_y, x_o, y_o and z_o are constants.

For this case, the value of $\omega = \mathbf{A}\tilde{\xi} - \dot{\tilde{\xi}} = 0$. Therefore, this problem can be reduced to a regulator with the same $\mathbf{K}$ found in the previous section [5]. Moreover, we will obtain a similar result if the tracking is extended also to the z axis.

3 Simulation Results

The first step is to simulate the model of the non-linear system without any controller. The quadrotor system is implemented in SIMULINK. The main scheme of the simulation block diagram is shown in Figure 1.

The open-loop behavior is obtained as a response to an initial roll or pitch angular speed condition. As it is clear from Figure 2, the system is highly unstable due to the gyroscopic effect that changes the orientation of the propeller axis of rotation [1].

Afterwards, the value of the matrix gain $\mathbf{K}$ for attitude stabilization is obtained in MATLAB using the `lqr` command. Besides, Figure 3 shows the LQR controller together with the non-linear model. The value of the velocities of the motors to reach the hover condition (4380 rpm) is added to the output of the controller since the linearized model was obtained considering the hover condition. The attitude stabilization of the system is tested giving some initial values to the pitch, roll and yaw angles inside of the non-linear model while the reference $\mathbf{X_o}$ is a zero vector. The results for initial values of $\pi/6$ for roll, $-\pi/6$ for pitch and $\pi/6$ for yaw are displayed in Figure 4.

Our next goal is to add the height controller to the attitude stabilization. As it was indicated in the previous section, this problem can be reduced to a regulator problem with a matrix gain equal to the one obtained for attitude stabilization. Therefore, the task is to reach some reference in the z axis and stayed there. In this case, $\mathbf{X_o}$ has a value just for the z axis while we give an initial condition to the x and y axes. Figure 5 illustrates the response obtained for a z-reference value of 2 m and a initial condition of 0.5 m to the x and y axes.

Finally, the tracking in X and Y is tested. Some minor changes are developed in the main block diagram in order to simulate the tracking response. In fact, the configuration for this test can be seen in Figure 6.

Three different tests are developed using the configuration presented in Figure 6. Firstly, a fixed point $[x_o\ y_o\ z_o] = [5\ 5\ 5]$ is given as a reference and the quadrotor must reach this point and stay there, see Figure 7. Next, a ramp with a desired slope is given for each one of the axes x, y and z. The slope is the same for the three axis, see Figure 8. Third, a ramp with different slopes are given for the X and Y axes, while a desired fixed value is given for the Z axis, see Figure 9.

References

[1] S. Bouabdallah and R. Siegwart, "Design and control of quadrotors with application to autonomous flying," Thèse sciences, Faculté des sciences et techniques de l'ingénieur STI, Section de microtechnique, Institut d'ingénierie des systèmes I2S (Laboratoire de systèmes autonomes 1 LSA1). Dir.: Roland Siegwart, Ecole polytechnique fédérale de Lausanne, EPFL, Lausanne, 2007.

[2] T. Bresciani, "Modelling , identification and control of a quadrotor helicopter," Master Thesis, Department of Automatic Control, Lund University, Sweden, October 2008.

[3] A. Friis, “Autonomous control of miniature quadrotor following fast trajectories,” Control Engineering Master’s Thesis, Aalborg University, Denmark, June 2010.

[4] C. Mary, L. Cristiana, and S. Konge, “Modelling and control of autonomous quad-rotor,” 2^{nd} Semester Project of the Intelligent Systems Master Program, Aalborg University, Denmark, June 2010.

[5] P. Dorato, C. Abdallah, and V. Cerone, *Linear Quadratic Control: An Introduction.* Melbourne, FL, USA: Krieger Publishing Co., Inc., 2000.

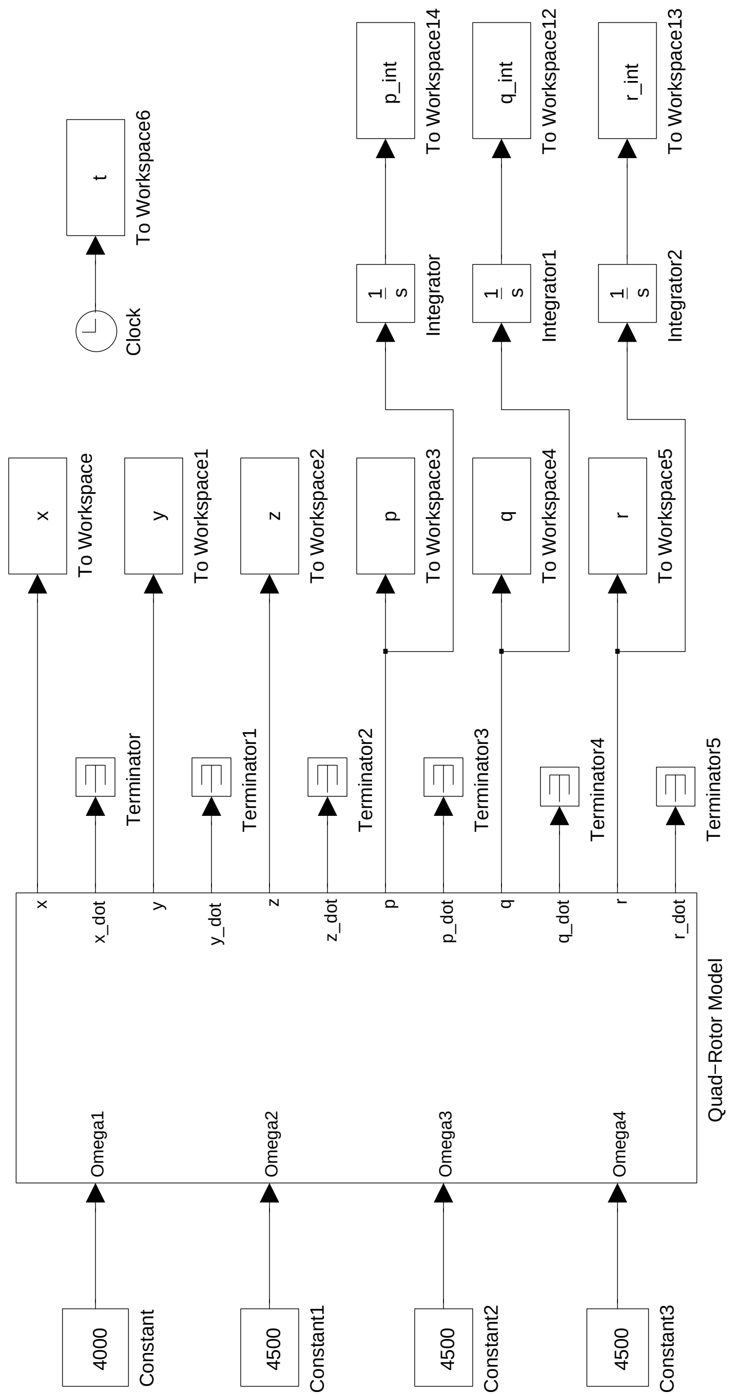


Figure 1: Block diagram of the quadrotor model implementation in SIMULINK.

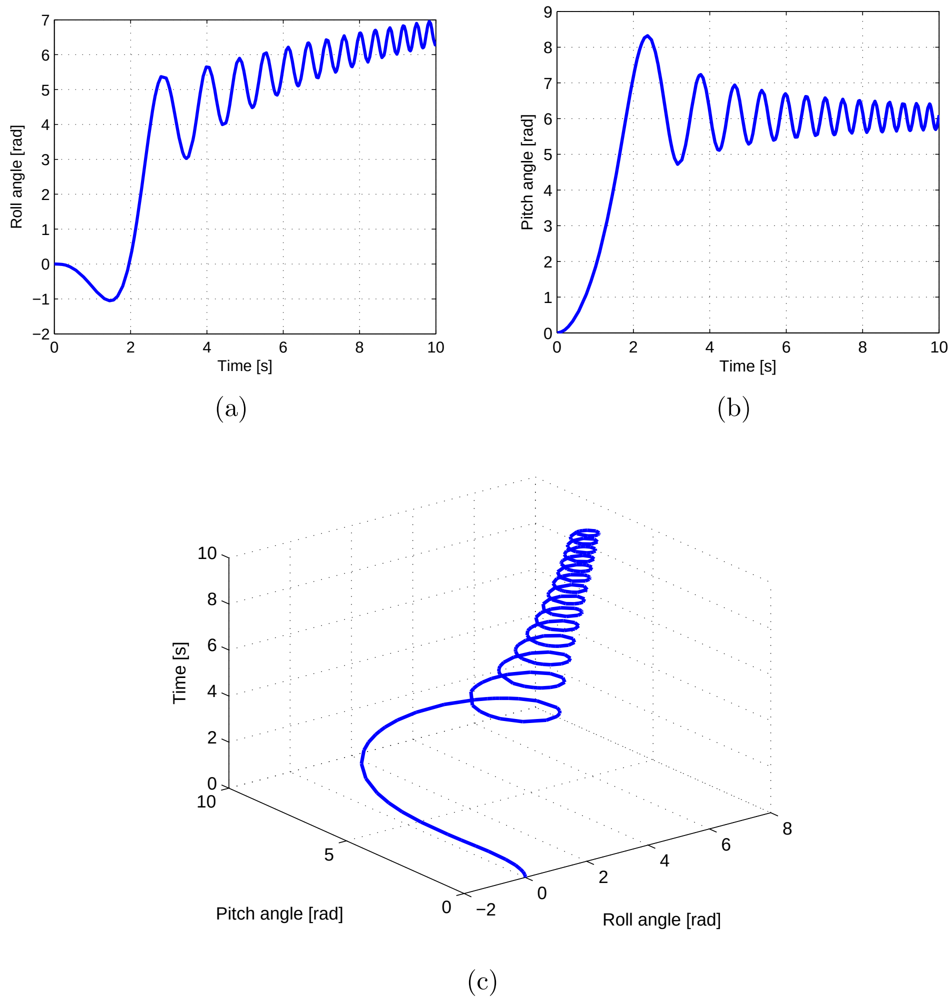


Figure 2: Response of the non-linear model for the roll, pitch angles. These angles are respect to the body-frame. Figure (c) shows the interdependence between the angles.

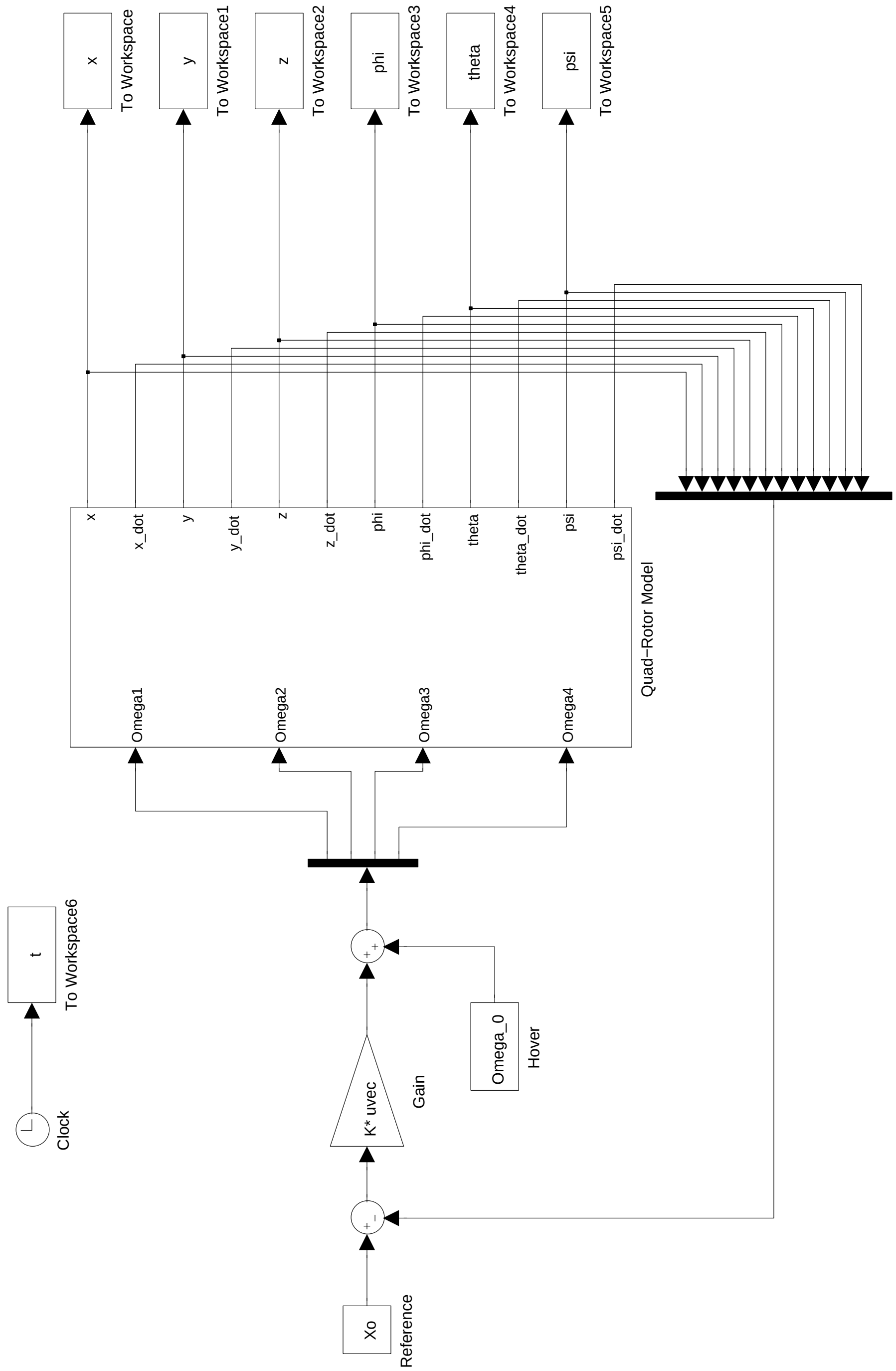
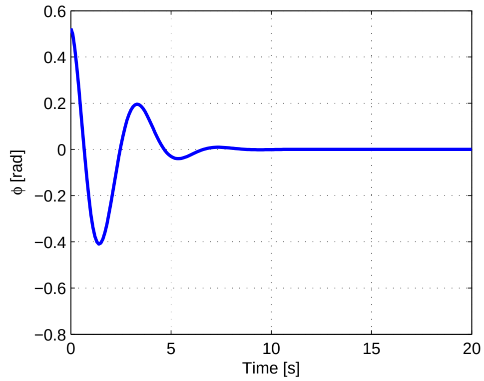
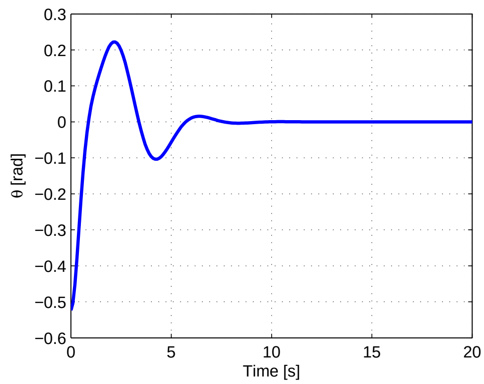


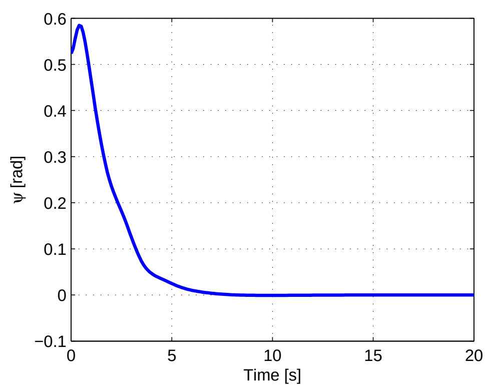
Figure 3: LQR controller block diagram.



(a)

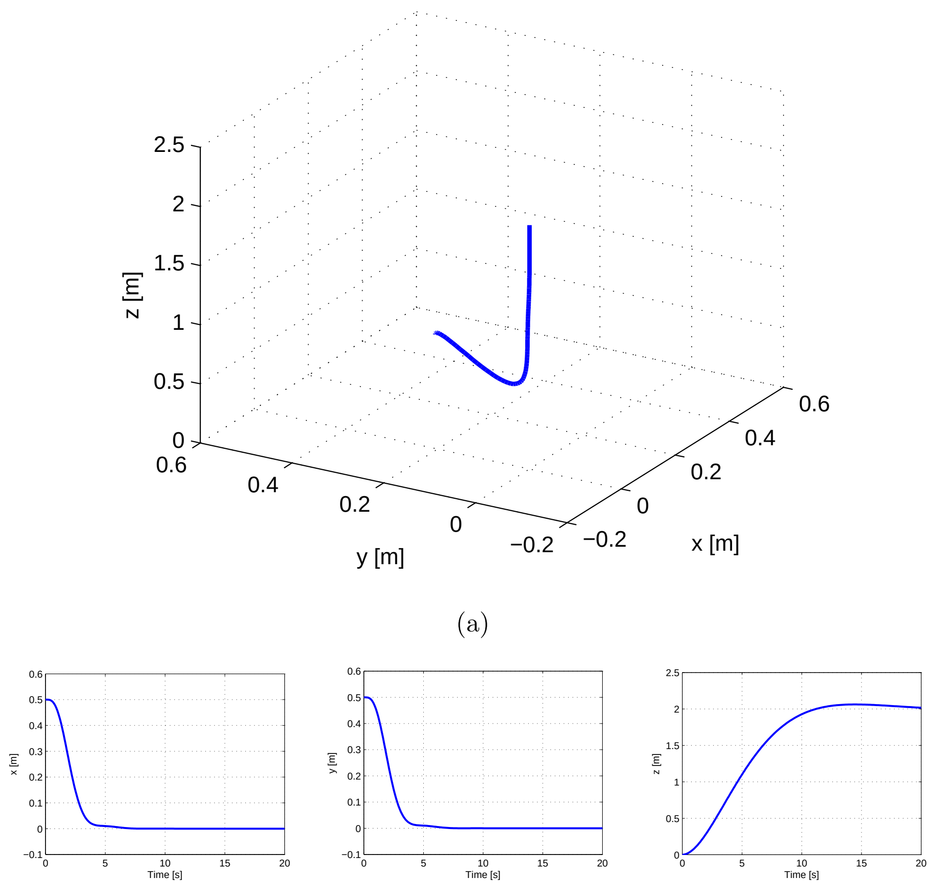


(b)



(c)

Figure 4: Roll, pitch and yaw responses of the LQR regulator.



(a)

(b) (c) (d)

Figure 5: Simulation results for the height controller.

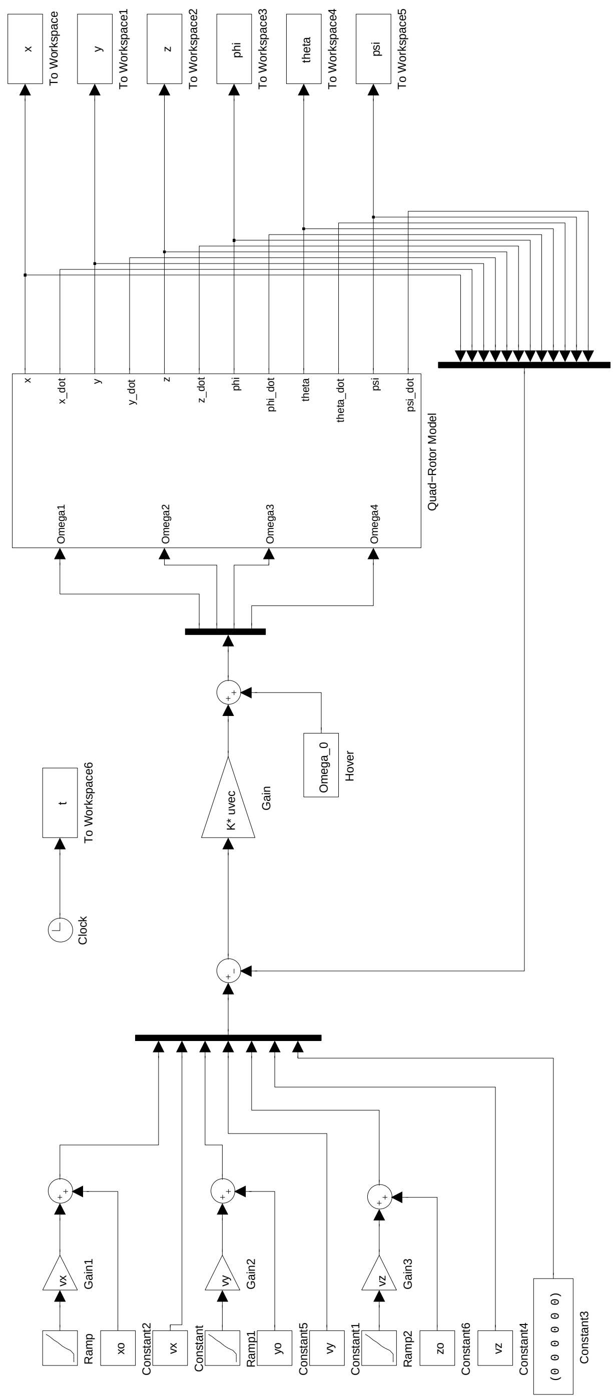
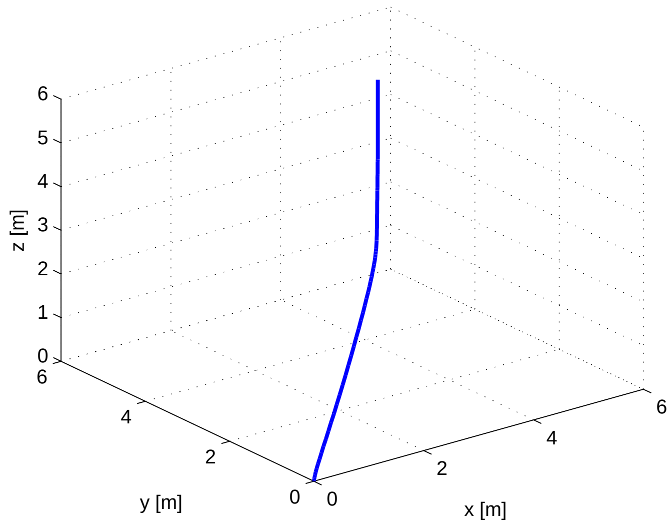
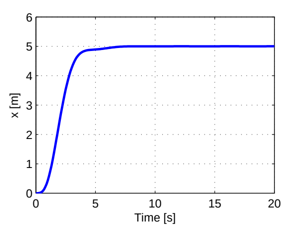


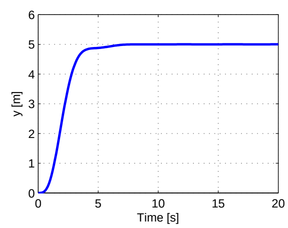
Figure 6: Block diagram configuration for simulating the tracking problem.



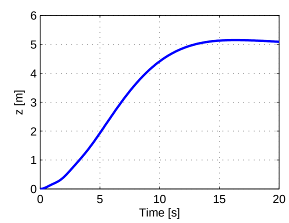
(a)



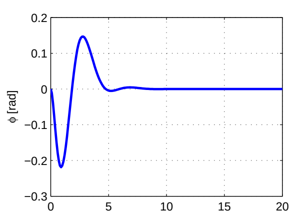
(b)



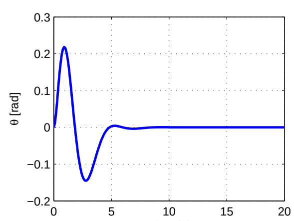
(c)



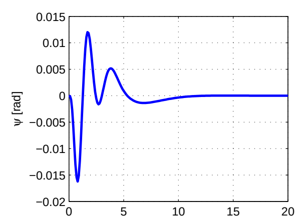
(d)



(e)

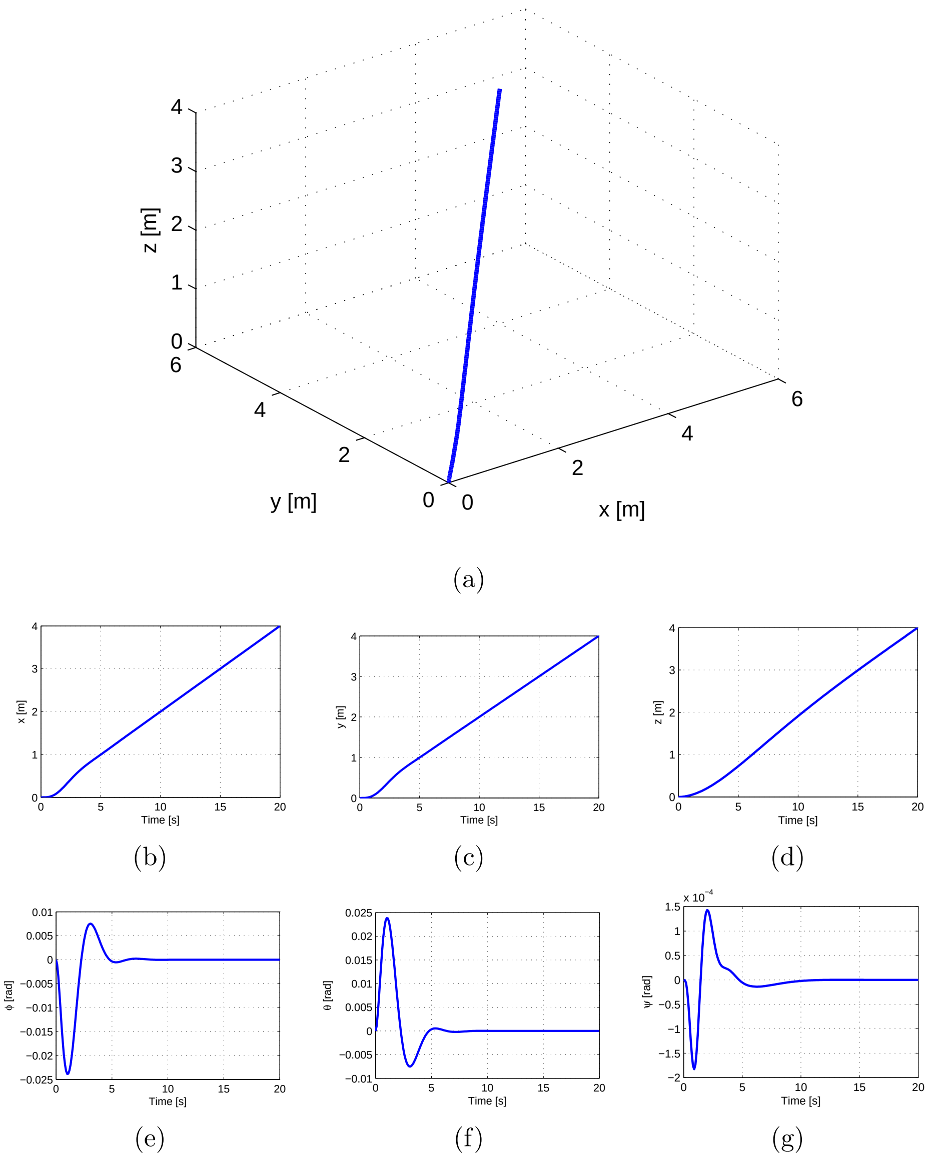


(f)



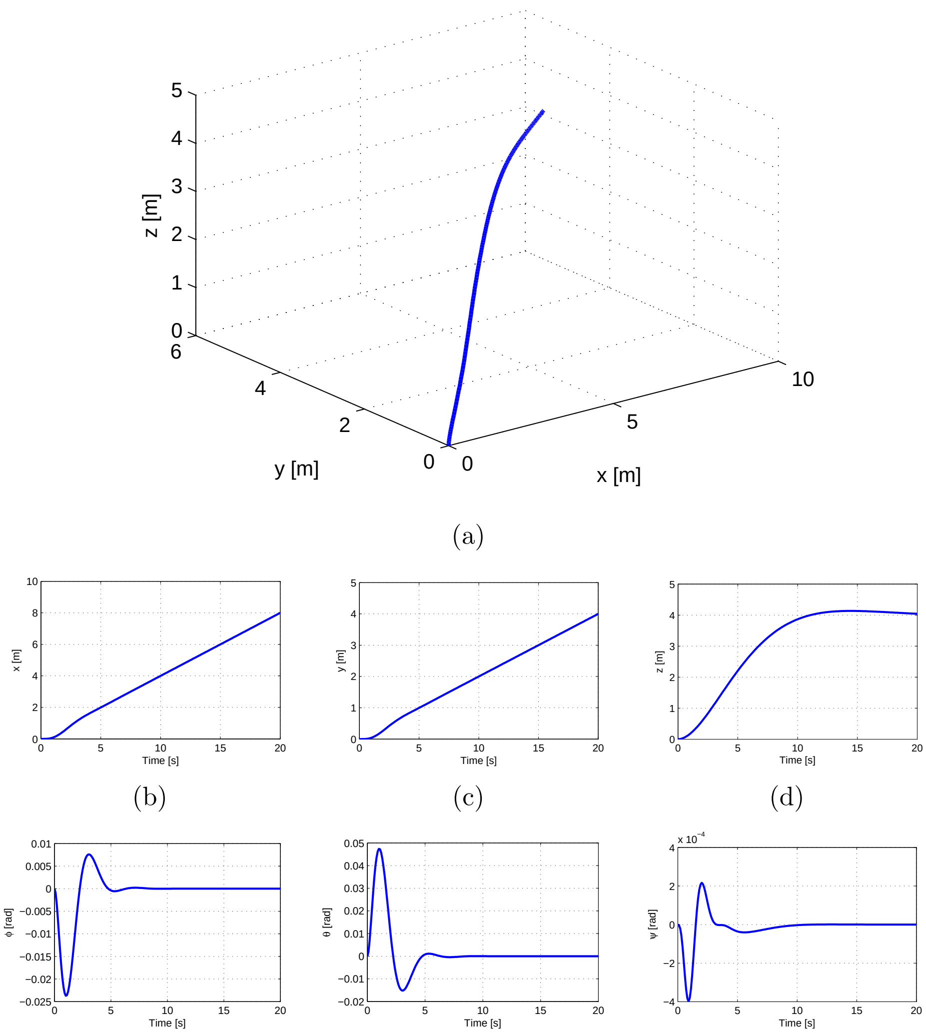
(g)

Figure 7: Test 1, a fixed point $[x_o\ y_o\ z_o] = [5\ 5\ 5]$ is given as a reference.



(a) (b) (c) (d) (e) (f) (g)

Figure 8: Test 2, a ramp with a desired slope is given for each one of the axes.



(a) (b) (c) (d) (e) (f) (g)

Figure 9: Test 3, a ramp with different slopes are given for the x and y axes, while a desired fixed value is given for the z axis.