Milestone 4 GA Report

Landing on an unknown planet

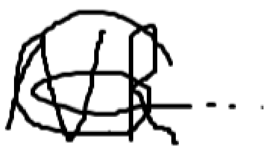
Prepared by:
Ntokozo Radebe

Prepared for:
EEE4119F
Department of Electrical Engineering
University of Cape Town

May 10, 2024

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.



May 10, 2024

Ntokozo Radebe

Date

Contents

Chapter 1

Introduction

1.1 Problem Description

In the wake of Earth's catastrophic decline, humanity's survival hinges on the discovery and colonization of habitable exoplanets. The recent identification of Omicron Persei 9, a planet capable of sustaining life, presents a beacon of hope amidst the bleakness of interstellar desolation. However, the successful colonization of this newfound world demands the development of advanced mechatronic systems. As the planets' last surviving engineer, one is tasked with designing a controller for a planetary lander, such that it safely touches down on the planet, whose exact gravity is unknown.

1.1.1 Problem parameters

The lander is characterized by physical parameters such as mass, dimensions, and thrust capabilities. Omicron Persei 9's gravitational acceleration,g, remains unknown, the mass moment of inertia, Izz the lander, as well as the centre of mass offset dL are also unknown.

1.2 Goals

Given the lander description, this report aims to outline the engineering strategy followed in system identification and thereafter control system design to achieve the following milestones:

1. An accurate mathematical model of the rocket must be developed to determine how the rocket will move through space.
2. Determine g, Izz and dL using this mathematical model though in the course course such as Euler-Lagrange Mechanics[1].
3. Use the equations of motion developed from the model to create a new equation based on the state space matrices and force matrix that will indicate the closed loop behaviour of the system.
4. Complete the controller for the system in Simulink and test against different modes it to ensure that it meets all the requirements and specifications.
5. Compile a comprehensive project report detailing modeling process, control scheme used, results, and insights gained throughout the project phase.
6. Hopefully apply for an internship at SpaceX to work on Falcon 9.

[1] It must be highlighted that, all calculations, derivations and designs were all produced in MATLAB, and relevant codes are attached

Chapter 2

Modelling

2.1 Lander Modelling

To model the Lander equation of motion, a series of steps and equations were applied, namely identifying the generalized coordinates, determining the position vector and velocity vectors, determining the kinetic and potential energy, and lastly using the manipulator equation to extract key parameters such as gravitational acceleration of the planet, second moment of inertia, and the offset COM length.

Using Euler-Lagrange Mechanics, the equations of motion were determined in this order:

Generalized coordinates vector:

$$q = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{2.1}$$

Lander Position and Velocity vector:

$$r = \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.2}$$

Using the Jacobian

$$\dot{r} = \frac{d}{dt}(r) = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \tag{2.3}$$

Kinetic energy:

$$T_{total} = T_{translational} + T_{rotational} = \frac{1}{2} m\dot{r}^T \dot{r} + \frac{1}{2} I_{zz}\dot{\theta} \tag{2.4}$$

Potential energy:

$$V_g = m \begin{bmatrix} 0 & g \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = mgy \tag{2.5}$$

Using the manipulator equation:

$$M(q)\ddot{q}+C(q,\dot{q})+G(q)=Q \tag{2.6}$$

The elements of the Mass matrix:

$$M_{i,j}(q)=\frac{\partial T_{total}}{\partial \dot{q}_i \partial \dot{q}_j} \Longrightarrow M(q)=\begin{bmatrix} 4000 & 0 & 0 \\ 0 & 4000 & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{2.7}$$

Centrifugal matrix:

$$C(q,\dot{q})=\dot{M}\dot{q}-\frac{\partial T}{\partial q}=\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \tag{2.8}$$

Gravity matrix:

$$G(q)=\frac{\partial V}{\partial q}=\begin{bmatrix} 0 & gm & 0 \end{bmatrix} \tag{2.9}$$

Generalized forces:

$$Q_i=\sum_{j=1}^{m} F_j . \frac{\partial r_j}{\partial q_i} \tag{2.10}$$

Generalised forces are determined in the inertial frame, therefore transformation of the forces from body to inertial frame is required. Position of F_1 and F_2 in body frame:

$$r_{F_1}^B=\begin{bmatrix} -\frac{L_1}{2} & 0 \end{bmatrix}^T \tag{2.11}$$

$$r_{F_2}^B=\begin{bmatrix} \frac{L_1}{2} & 0 \end{bmatrix}^T \tag{2.12}$$

Position of F_1 and F_2 in inertial frame:

$$r_{F_1}^I=\begin{bmatrix} x & y \end{bmatrix}^T+\begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix}\begin{bmatrix} -\frac{L_1}{2} & 0 \end{bmatrix}^T=\begin{bmatrix} x-\frac{L_1 cos\theta}{2} & y+\frac{L_1 sin\theta}{2} \end{bmatrix}^T \tag{2.13}$$

$$r_{F_2}^I=\begin{bmatrix} x & y \end{bmatrix}^T+\begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix}\begin{bmatrix} \frac{L_1}{2} & 0 \end{bmatrix}^T=\begin{bmatrix} x+\frac{L_1 cos\theta}{2} & y-\frac{L_1 sin\theta}{2} \end{bmatrix}^T \tag{2.14}$$

Thruster forces in inertial frame after the rotational matrix is applied:

$$F_1^I = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ F_1 \end{bmatrix} = \begin{bmatrix} -F_1 sin\theta \\ F_1 cos\theta \end{bmatrix} \tag{2.15}$$

$$F_2^I = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ F_2 \end{bmatrix} = \begin{bmatrix} -F_2 sin\theta \\ F_2 cos\theta \end{bmatrix} \tag{2.16}$$

From equation (1.10) generalized forces are determined and result to:

$$Q = \begin{bmatrix} Q_x \\ Q_y \\ Q_\theta \end{bmatrix} = \begin{bmatrix} -F_1 sin\theta - F_2 sin\theta \\ F_1 cos\theta + F_2 cos\theta \\ \frac{L_1}{2} F_2 - \frac{L_1}{2} F_1 \end{bmatrix} \tag{2.17}$$

Finally, from equation (1.6), equations of motion are determined to be:

$$\ddot{q} = \frac{Q - C(q, \ddot{q}) - G(q)}{M(q)}$$

$$\therefore \ddot{q} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} -\frac{sin\theta(F_1+F_2)}{m} \\ \frac{F_1 cos\theta}{m} + \frac{F_2 cos\theta}{m} - g \\ \frac{L_1 F_1}{2I_{zz}} - \frac{L_1 F_2}{2I_{zz}} \end{bmatrix} \tag{2.18}$$

Gravity, g

To determine acceleration due to gravity, both F_1 and F_2 are set to zero. This models the lander as being subjected to free fall, thus the only force acting on it is the gravitational force. As a result, the linear acceleration of the lander as it falls is the acceleration due to gravity of the unknown planet. This is governed by Newton 2nd law, and Figure 2.1 shows the FBD of the lander. From equation (1.18):

$$\ddot{y} = \frac{F_1 cos\theta}{m} + \frac{F_2 cos\theta}{m} - g \Longrightarrow \ddot{y} = -g$$

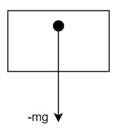


Figure 2.1: Free body diagram modelling free fall

After simulating mode 1, Figure 2.2 is obtained. A mean function is used to determine the average value of the gravitational acceleration:

$$g = 9.86 \frac{m}{s^2}$$

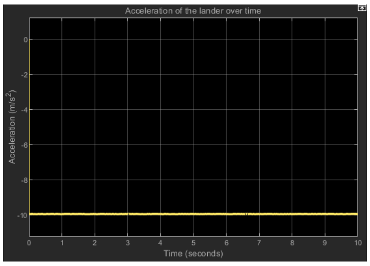


Figure 2.2: Acceleration of the Lander in free fall

Second Moment of Inertia, I$_{zz}$

To determine I_{zz}, set $F_1 \neq F_2 \neq 0$, I_{zz} is dependant on the angular acceleration as seen in equation (1.18), $[\ddot{\theta} = \frac{L_1 F_1}{2I_{zz}} - \frac{L_1 F_2}{2I_{zz}}]$. As stated in the milestone description and F_2 must be greater than F_1 for counter-clockwise rotation, therefore this cause the lander to rotate about the z-axis with an angle θ. This is governed by the torque equation:

$$\tau = I_{zz} \times \alpha, \alpha = \ddot{\theta} \Longrightarrow \frac{L_1}{2} \times F_1 - \frac{L_1}{2} \times F_2 = I_{zz} \times \ddot{\theta}$$

Since I_{zz} must be a positive value, the absolute function is applied on both sides of the equations. From the simulation, as seen in Figure 2.3 the average angular acceleration value is determined to be $\alpha = 0.0072\frac{rad}{s^2}$, when substituted in the I_{zz} equation:

$$I_{zz} = 56255.625 kg.mm^2$$

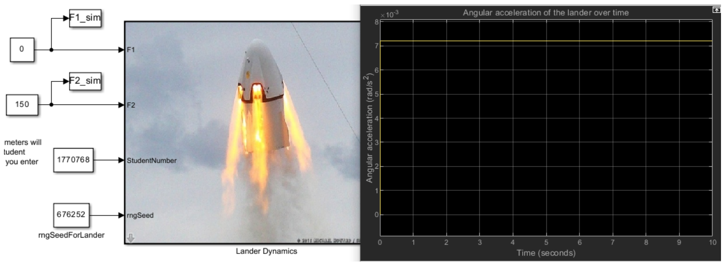


Figure 2.3: Angular acceleration of the Lander

Offset COM length, dL

After determining I_{zz} from equation (1.18), express I_{zz} using the parallel axis theorem to determine dL:

$$I_{zz} = \frac{1}{12} \times m \times (L_1^2 + L_2^2) + m \times dL^2$$

$$dL = 2.394m$$

Chapter 3

Control Scheme

From the equation of motion determined in the chapter 2, the non-linear state space model is obtained to be:

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ -\frac{sin\theta(F_1+F_2}{m} \\ \dot{y} \\ \frac{F_1cos\theta-gm-F_2cos\theta}{m} \\ \dot{\theta} \\ -\frac{L_1(F_1-F_2}{2I_{zz}} \end{bmatrix}$$

Linearizing the model is useful to determined the state matrices. These state matrices are crucial in determining the control of the lander and its thrust forces. The lander is a non-linear system that needs to be converted into state space before the LQR algorithm can be implemented.

In summary the control scheme utilized in this section is to linearized the system about an operating point, with small Δ in the states and control action, then perform a traditional state-feedback linear control system. Using a state-feedback controller is simple and less computationally intensive over implementing a PID controller. To implement the LQR, the following steps are followed:

Operating Point Selection

An operating point (also called an equilibrium point) is chosen around which the system will be linearized. This operating point represents the state of the system where the linearization is performed. Setting equilibrium state values, the operating point is set to $x = \dot{x} = y = \dot{y} = \theta = \dot{\theta} = 0$.

Linearization Process

To linearized the system, the equilibrium point must be determined as done previously, such that the non-linear system is stable about the given point [1], hence the condition to be satisfied is:

$$\dot{x} = f(x^*, u^*) = 0 \text{ where } \dot{x} \text{ represents the states}$$

From the non-linear state space model equation, using the condition $F_1 \neq 0$ & $F_2 \neq 0$ results in:

$$\ddot{x} = -\frac{sin\theta(F_1 + F_2)}{m} = 0 \Longrightarrow sin\theta(F_1 + F_2) = 0 \tag{3.1}$$

$$\ddot{y} = \frac{F_1 cos\theta}{m} + \frac{F_2 cos\theta}{m} - g = 0 \Longrightarrow F_1 + F_2 = \frac{mg}{cos\theta} \tag{3.2}$$

$$\ddot{\theta} = \frac{L_1(F_1 - F_2)}{2I_{zz}} = 0 \Longrightarrow F_1 = F_2 \tag{3.3}$$

To now satisfy the condition:

From equation (1.1), $\theta = 0$ rad

Using $\theta = 0$ rad, from equation (1.2), $F_1 + F_2 = mg$

Using equation (1.2) and (1.3), $F_1 = F_2 = \dfrac{mg}{2}$ & $\theta = 0$ rad

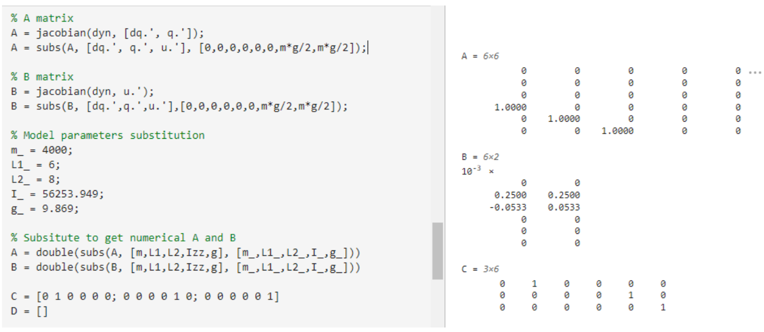
Obtaining Linearized Equation

$$u = \begin{bmatrix} F \\ \alpha \end{bmatrix}$$

The linearized equation is determined to be:

$$\Delta\dot{X} = A\Delta x + B\Delta u, \text{where } A = \frac{\partial F}{\partial x} \;\&\; B = \frac{\partial F}{\partial u}$$

Using the Jacobian matrix, the nonlinear differential equations are linearized to obtain a set of linear ordinary differential equations (ODEs) or state-space equations. These linearized equations describe the small-signal behavior of the system around the chosen operating point



```
% A matrix
A = jacobian(dyn, [dq.', q.']);
A = subs(A, [dq.', q.', u.'], [0,0,0,0,0,0,m*g/2,m*g/2]);

% B matrix
B = jacobian(dyn, u.');
B = subs(B, [dq.',q.',u.'],[0,0,0,0,0,0,m*g/2,m*g/2]);

% Model parameters substitution
m_ = 4000;
L1_ = 6;
L2_ = 8;
I_ = 56253.949;
g_ = 9.869;

% Subsitute to get numerical A and B
A = double(subs(A, [m,L1,L2,Izz,g], [m_,L1_,L2_,I_,g_]))
B = double(subs(B, [m,L1,L2,Izz,g], [m_,L1_,L2_,I_,g_]))

C = [0 1 0 0 0 0; 0 0 0 0 1 0; 0 0 0 0 0 1]
D = []
```

```
A = 6×6
        0        0        0        0        0 ...
        0        0        0        0        0
        0        0        0        0        0
   1.0000        0        0        0        0
        0   1.0000        0        0        0
        0        0   1.0000        0        0

B = 6×2
10^-3 ×
        0        0
   0.2500   0.2500
  -0.0533   0.0533
        0        0
        0        0
        0        0

C = 3×6
     0     1     0     0     0     0
     0     0     0     0     1     0
     0     0     0     0     0     1
```

Figure 3.1: Local linearization method around the operating point from Matlab

Control Design

The choice to use a Linear Quadratic Regulator (LQR) over PID or PD is because the LQR can control all the states of the system with a single controller. In matlab, Figure 3.2 shows a code snippet used to determine optimal gain K. With the A and B matrix coefficients that represent the linearized system dynamics obtained and the value of R being set to $\frac{1}{4}eye(6)$, the Q values can now be tweaked to find optimum Q values for the controller. Ultimately finding ideal Q values was through a method of trial and error by starting with q values of 1, the system was then tested.

```
Q = 750*eye(6);
Q(5,5) = 375;
R = 1/4*eye(2);

% calculate LQR gain

K = lqr(A, B, Q, R);

%% Closed Loop System
Ac = [(A-B*K)];
Bc = [B];
Cc = [C];
Dc = [D];
sys_cl = ss(Ac, B, C, D, 'statename', states,...
                        'inputname', inputs,...
                        'outputname', outputs);
```

Figure 3.2: Matlab code snippet for determining optimal gain K

Figure 3.3 shows the implementation of the LQR along with state-feedback and proportional gain.

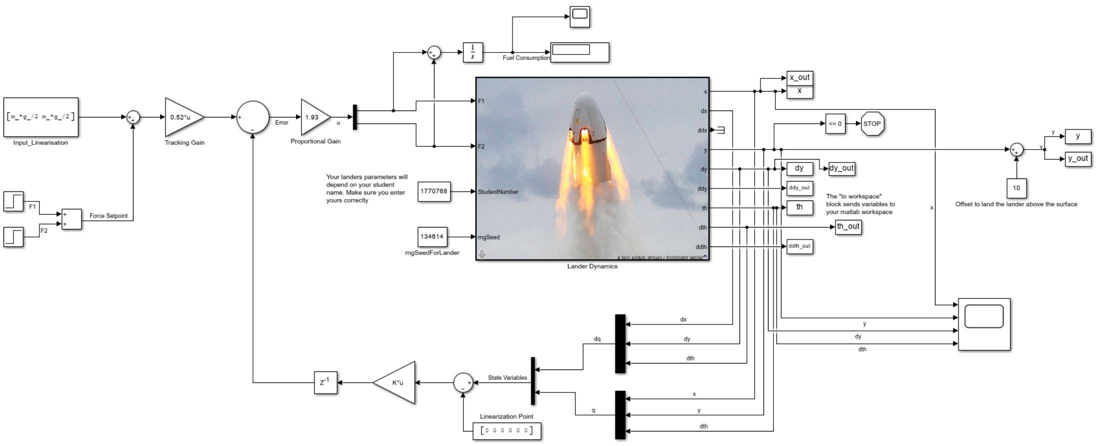
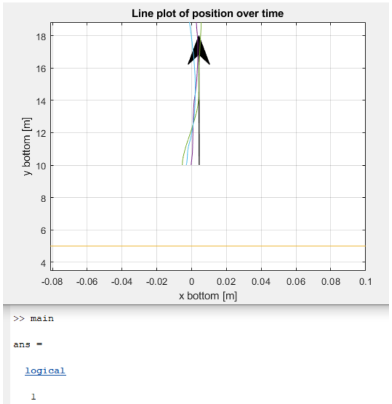


Figure 3.3: Simulink LQR controller implementation

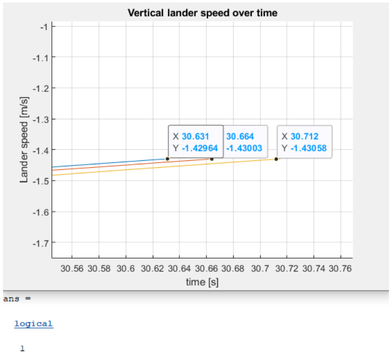
Chapter 4

Results and Discussion

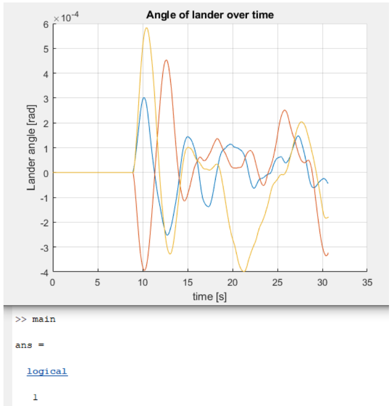
4.0.1 Mode 1



(a) Line plot



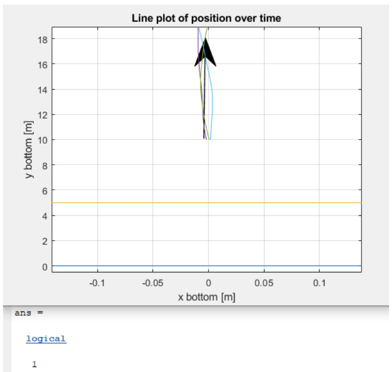
(b) Speed plot



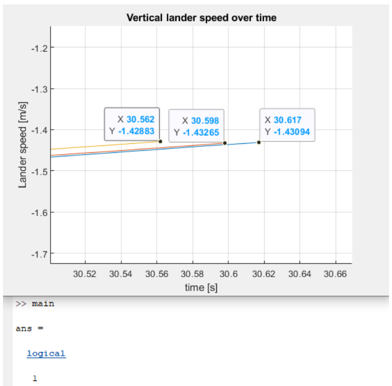
(c) Angle plot

Figure 4.1: Controller performance for mode 1

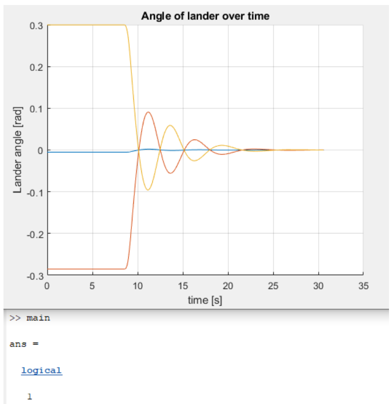
4.0.2 Mode 2



(a) Line plot



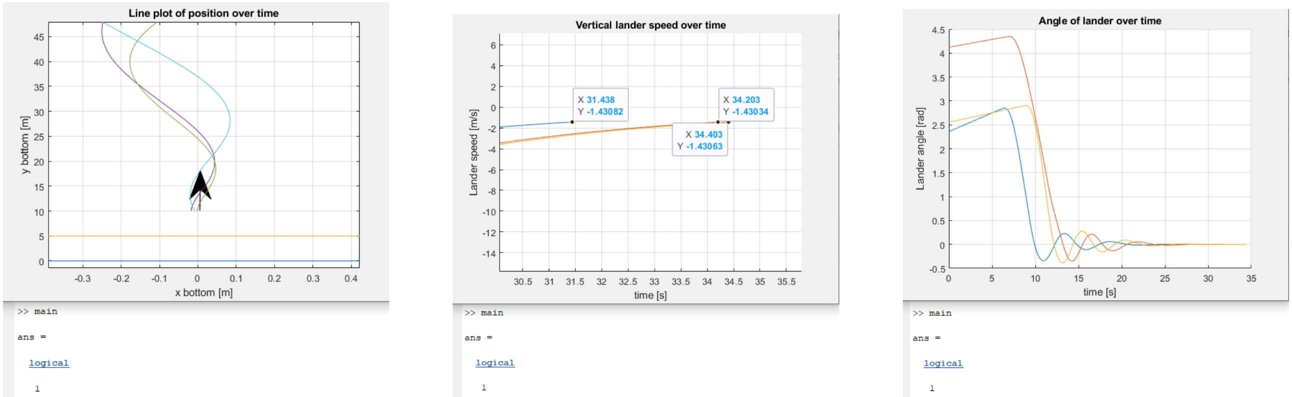
(b) Speed plot



(c) Angle plot

Figure 4.2: Controller performance for mode 2

4.0.3 Mode 3



(a) Line plot (b) Speed plot (c) Angle plot

Figure 4.3: Controller performance for mode 3

Table 4.1 provides a breakdown of the criteria used to test the quality of the controller for each mode.

Table 4.1: Summary of results in different modes

Criteria	**Mode 1**	**Mode 2**	**Mode 3**
Landing height $y < 0.5m$	10 m	10 m	10 m
Vertical velocity $v_y < 0.2m/s$	-1.43 m/s	-1.43 m/s	-1.43 m/s
Landing angle $\theta < 10°$	$6e-04°$	0°	0°
Horizontal position $x = 0m$	0.001 m	- 0.01 m	0.01 m
Minimum fuel consumed	1.58e06 Ns	1.503e06 Ns	4.11e05 Ns
Minimum time take	30.7 s	30.5 s	31.4 s

As seen in the figures and table above, the controller does perform fairly well. It lands safely at 10 m, however, the landing speed is above 0.2 m/s. It has a worst cost estimate of 1.58e06 Ns 1.503e06 Ns 4.11e05 Ns for Mode 1,2 and 3 respectively. The horizontal distance x is approximately zero for each mode, meaning it lands on the bullseye. The best minimum time it takes to land is 30.5 s. In summary, the lander can safely take passengers to any unknown planet, space travel is indeed possible.

Chapter 5

Conclusion

This project has successfully addressed the critical task of landing a 4000kg lander on an unknown planet through implementation of an effective controller. By utilizing concepts learned through the course and the power of MATLAB and Simulink, the lander was accurately modeled, and an LQR controller designed to plan the trajectory of the lander to land safely on the ground. The linearization techniques applied on the equilibrium point allowed good tracking of the landing point, $x = 0m$. A lot of trial and error was made to obtain the optimum gain in which the controller produces the desired output. This project displayed the criticalness of control fundamentals and highlighted the power of simulation tools such as MATLAB. In conclusion, the lander safely landed, and humanity gets to live to see another day in a different world.

5.1 Recommendations

A different controller such as a PID can be implemented to test its performance compared to the LQR controller. However, due to the PID being computationally intense, it requires higher expertise in the field of control engineering. However, with the fundamental control theory, it is achievable.

Bibliography

[1] A. P. P. M. C. Town, “Lecture 10: Linearization (dynamic),” Electrical Engineering Department, University of Cape Town, 11 April 2021, [Accessed 05-05-2024].