Assignment 3

Lorenzo Fici, Hannes Gustafsson, Davide Nascivera

September 2024

1 Introduction

In this task, we will implement LQR controller for the reference tracking. The LQR controller is designed based on the discrete-time linearized dynamics

$$x_{t+1} = Ax_t + Bu_t \tag{4}$$

where $x_t \in \mathbb{R}^{12}$ and $u_t \in \mathbb{R}^6$, being system state and control input respectively. The matrices A and B are provided in the Astrobee class, method `c2d`. The objective is to design an LQR controller that achieves the following criteria:

1. It moves the astrobee from the initial state $x_0 = 0$ to the reference state

$$p_{\text{ref}} = \begin{bmatrix} 1 \\ 0.5 \\ 0.1 \end{bmatrix}, \quad v_{\text{ref}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \theta_{\text{ref}} = \begin{bmatrix} 0.087 \\ 0.077 \\ 0.067 \end{bmatrix}, \quad \omega_{\text{ref}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix};$$

2. The first three inputs (the forces) may not exceed 0.85 N;
3. The last three inputs (the torques) may not exceed 0.04 Nm;
4. From 12 seconds until the simulation ends (20 seconds):
   - the distance to the reference must be less than 0.06 m;
   - the speed of the astrobee must be less than 0.03 m/s;
   - the Euler angles must differ from their targets by less than 10^{-7} radians;
   - the position overshoot must be less than 2 cm for a safe docking maneuver.

Tracking a reference state x_{ref} with LQR is done by finding a corresponding equilibrium input u_{ref} such that

$$x_{\text{ref}} = Ax_{\text{ref}} + Bu_{\text{ref}}. \tag{5}$$

With the incremented state and control variables ($\Delta x_t = x_t - x_{\text{ref}}$, $\Delta u_t = u_t - u_{\text{ref}}$), the linear state space model of the system in (4) becomes

$$\Delta x_{t+1} = A_d \Delta x_t + B_d \Delta u_t. \tag{6}$$

With the state space model in (6), we can now calculate the optimal gain matrix L, provided by an LQR controller, such that the state feedback controller $u_t = -L\Delta x_t$ minimizes the infinite horizon quadratic cost function given by

$$J = \sum_{t=0}^{\infty} \Delta x_t^T Q \Delta x_t + \Delta u_t^T R \Delta u_t$$

where $Q \succeq 0$ and $R \succ 0$ are symmetric and positive (semi-)definite matrices of appropriate dimensions. These matrices will be used to calculate your LQR feedback gain L. With stabilizing state feedback gain L, we can now find a u_{ref} in (5) of the form

$$u_{\text{ref}} = -L x_{\text{ref}}.$$

The optimal control to be applied to the original system can be calculated as

$$u_t = -L\Delta x_t + u_{\text{ref}} = -L x_t.$$

2 Questions

2.1 Q1

For an LQR design, where we use the Q matrix, to ensure a solution with bounded cost we must have the system (A,B) stabilizable and (A, $Q^{\frac{1}{2}}$) detectable. In our case the system has all eigenvalues in 1 and his eigenvectors are not linearly independent, so it's unstable. Therefore, we need a controller to make the system stable. To verify that the system is stabilizable we used the PBH test and we found out that we don't have perpedicunal left-eigenvectors to the B matrix.

2.2 Q2

Here, we are tasked with investigating how the tuning parameters of the LQR controller, specifically the matrices Q and R, affect the performance of the controller. We initialize the parameters as $Q = I_{12}$ and $R = I_6$, meaning that all states and inputs are penalized equally.

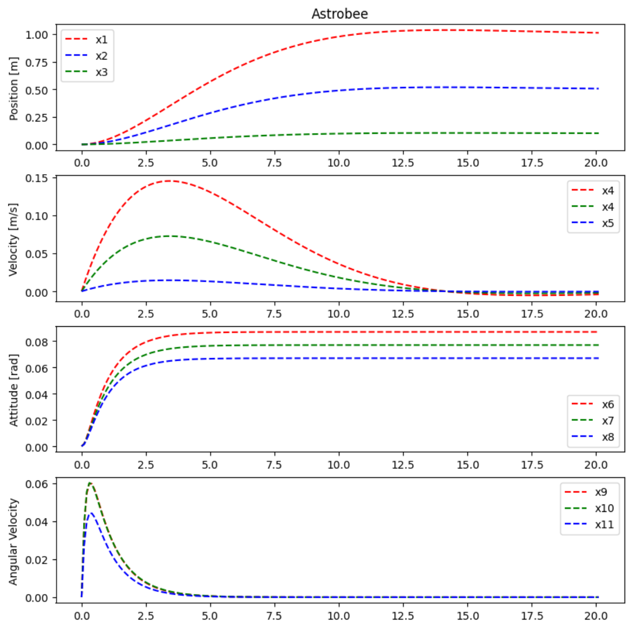

Figure 1: Intial controller performance

2.2.1 Experiment 1

Here, we multiply R by 10. This makes the controller put much more weight on the control effort. As a result, we observe the controller become more conservative in applying control inputs, compared to the baseline, resulting in the Astrobee taking longer to converge to the reference point.

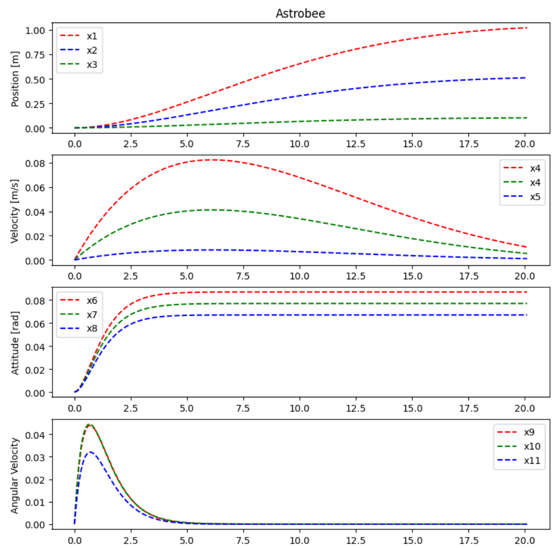

Figure 2: Controller performance with $R \times 10$.

2.2.2 Experiment 2

Here, we add 100 to the velocity components of the diagonal of Q, $Q[3:6]$ and $Q[9:]$. By adding 100 to these diagonal elements, we are making the controller place more emphasis on minimizing velocity errors, meaning the controller will react more aggressively to deviations in velocity.

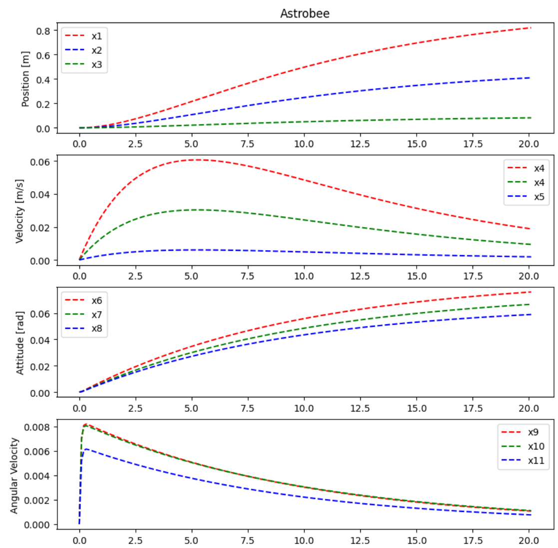

Figure 3: Controller performance with velocity components of $Q + 100$ and Rx10.

As expected, we observe faster adjustments to bring the velocity back to the reference state compared to the baseline.

2.2.3 Experiment 3

Here, we add 100 to the position and attitude-related components of Q, specifically $Q[0:3]$ and $Q[6:9]$. The controller is expected to place more emphasis on minimizing position and attitude errors, so it will react more aggressively to deviations in these states.

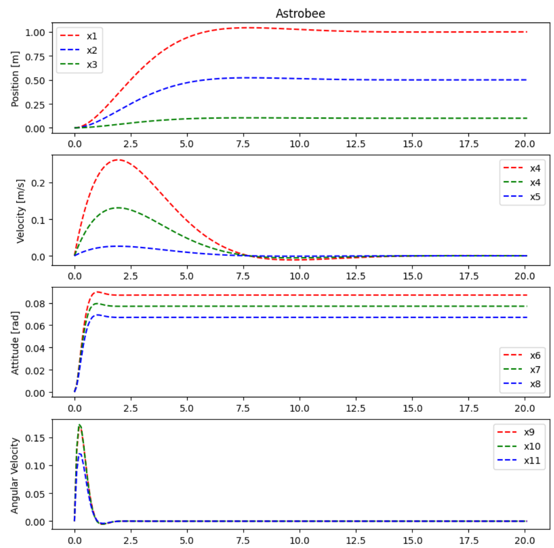

Figure 4: Controller performance with position and attitude components of $Q + 100$ and Rx10

As expected, we observe faster adjustments to bring the position and attitude back to the reference state compared to the baseline.

2.2.4 Experiment 4

Here, we add 100 to all elements of Q, meaning that position, velocity, and attitude states are all penalized much more heavily compared to the baseline. The controller will aggressively penalize all state deviations, so it will attempt to maintain all states (position, velocity, attitude) very close to their desired values.

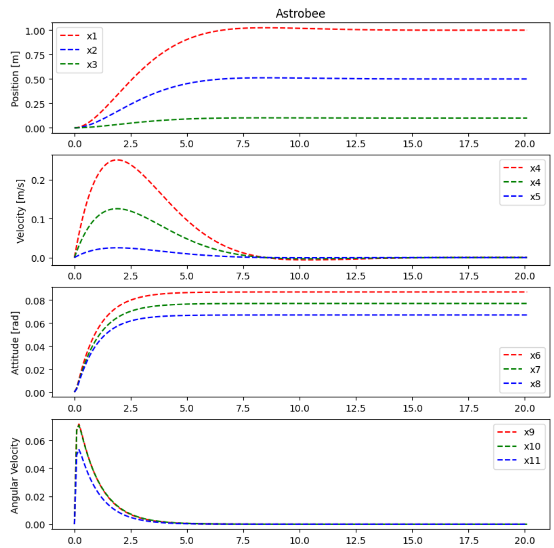

Figure 5: Controller performance with all elements of $Q + 100$ and Rx10

As expected, we observe the system showing much faster response to state deviation across all variables. However, we also observe a greater control effort than the baseline, as the controller uses more input to keep all state variables close to their targets.

2.3 Q3

The goal is to tune the LQR to meet these performance and constraint specifications while minimizing the control effort.

The tuning process was carried out in three main steps:

1. **Defining performance outputs:** To limit the degrees of freedom, we selected a matrix M that allowed us to focus specifically on the position, angle, and velocity specifications, reducing the complexity of the system.

2. **Normalizing weights using Bryson's rule:** This ensures that each state variable is properly scaled, avoiding dominance by any particular state and ensuring balanced performance.

3. **Tuning the closed-loop bandwidth:** The bandwidth of the LQR controller was tuned by selecting an appropriate $R = \rho \cdot \bar{R}$. We chose $\rho = 10$ for this system.

Specification	Description
Reference State	$p^{\text{ref}} = [1, 0.5, 0.1]^T$, $v^{\text{ref}} = [0, 0, 0]^T$, $\theta^{\text{ref}} = [0.087, 0.077, 0.067]^T$, $\omega^{\text{ref}} = [0, 0, 0]^T$
Control Force Limits	Forces must not exceed 0.85 N
Control Torque Limits	Torques must not exceed 0.04 Nm
Performance from 12 to 20 seconds	Distance error < 0.06 m, Speed < 0.03 m/s, Euler angle error $< 10^{-7}$ radians, Position overshoot < 2 cm for safe docking

Table 1: System Specifications and Control Constraints

4. **Fine-tuning performance weights:** In particular, the first adjustment was made to $Q[6:9]$. The low initial specifications created excessively high corresponding coefficients, necessitating careful refinement.

The final coefficients obtained through this tuning process are as follows:

$$Q_{\text{diag}} = \Bigg[\left(\frac{1}{0.06}\right)^2, \left(\frac{1}{0.06}\right)^2, \left(\frac{1}{0.06}\right)^2, \left(\frac{1}{0.03}\right)^2, \left(\frac{1}{0.03}\right)^2, \left(\frac{1}{0.03}\right)^2, \left(\frac{10^{-5}}{10^{-7}}\right)^2, \left(\frac{10^{-5}}{10^{-7}}\right)^2, \left(\frac{10^{-5}}{10^{-7}}\right)^2 \Bigg] \tag{1}$$

$$R_{\text{diag}} = \Bigg[\left(\frac{1.85}{0.85}\right)^2, \left(\frac{1.85}{0.85}\right)^2, \left(\frac{1.85}{0.85}\right)^2, \left(\frac{1}{0.04}\right)^2, \left(\frac{1}{0.04}\right)^2, \left(\frac{1}{0.04}\right)^2 \Bigg] \tag{2}$$

Results

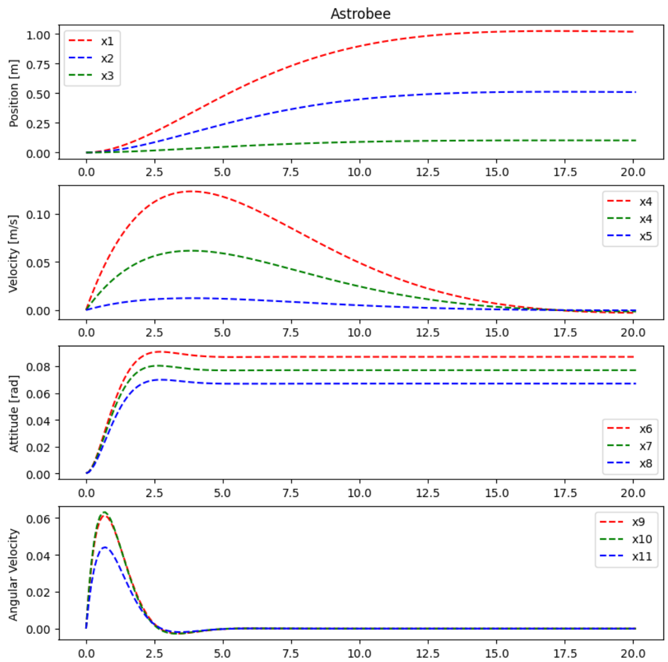

Figure 6: Tuned controller that respect the performance

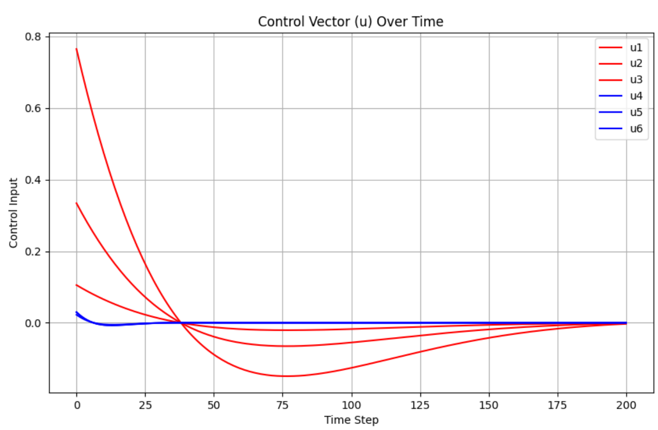

Figure 7: Input applied by the tuned controller

Parameter	Value
Max Distance to Reference	0.007705062633929502
Max Speed	0.004856610561633942
Max Forces	
x	3.0887743286831872
y	1.3494900401721106
z	0.42623838553939913
Max Torques	
x	0.10288780661987591
y	0.09593075906597595
z	0.08030073234420756
Max Euler Angle Deviations	
Roll	$3.6437471818973055 \times 10^{-7}$
Pitch	$4.4878582750040863 \times 10^{-7}$
Yaw	$1.8371656894389243 \times 10^{-7}$

2.4 Q4

Q_n is the process noise covariance matrix, i.e. how much we believe the system is affected by disturbances. A small value assumes the system dynamics are very accurate with small disturbances. The state estimate will rely more heavily on the model, while a larger value will give more weight to the measured data.

R_n is the measurement noise covariance matrix, i.e. how much we trust the accuracy of our sensors. A small value assumes that the measurements are very accurate and will rely heavily on sensor data to update the state estimate, while a larger value will trust the model more than the sensor data.

When both Q_n and R_n are equal, their magnitudes have impacts on the Kalman filter's performance. If both values are small, the Kalman filter assumes that both the system model and sensor measurements are highly accurate leading to faster response times. If both are large, a lot of uncertainty is assumed leading larger response times.

2.5 Q5

To further analyze this question, it's important to also consider the system inputs. As mentioned earlier, setting a small R (four orders of magnitude smaller) results in a controller that prioritizes the measurements more heavily. Consequently, the controller becomes less smooth, leading to sharper, smaller peaks in the control response.

Similarly, when smaller values are used in the matrix Q, the input controls become smoother, as the system places less emphasis on aggressively correcting

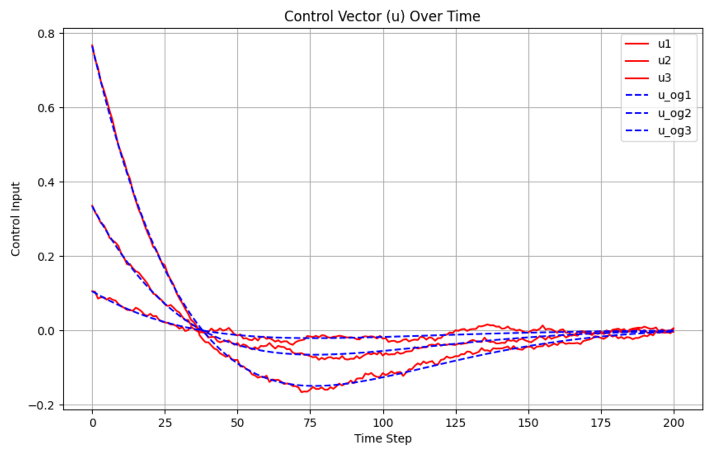

Figure 8: R Small Image

the state deviations. This trade-off between smoothness and responsiveness is a key aspect of tuning the controller.

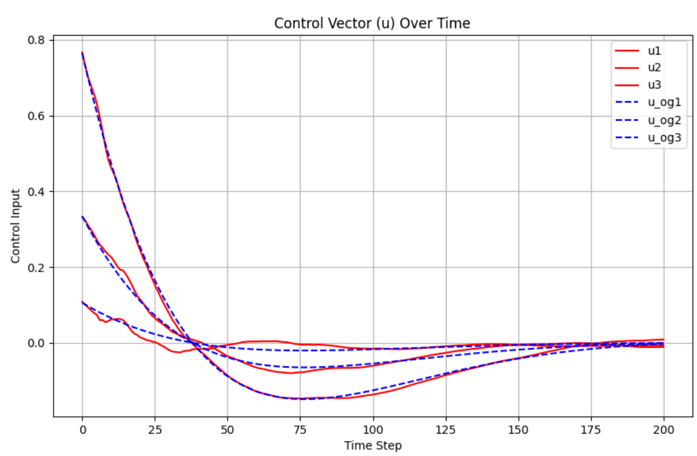

Figure 9: Q Small Image

We also want to show how the states are perturbed focusing on the position and velocity. As shown in Figure 10 where we have a small Q matrix (10^{-4}) compared to R matrix and we can see the perturbation of the positions and especially we see that the velocity along z-axis (blue line) is not really similar at the curve in Figure 6. Instead in Figure 11 we have a small R matrix (10^{-4})

compared to Q and we can see that even if the states are perturbed, the velocity are more likely to be similar at the original ones.

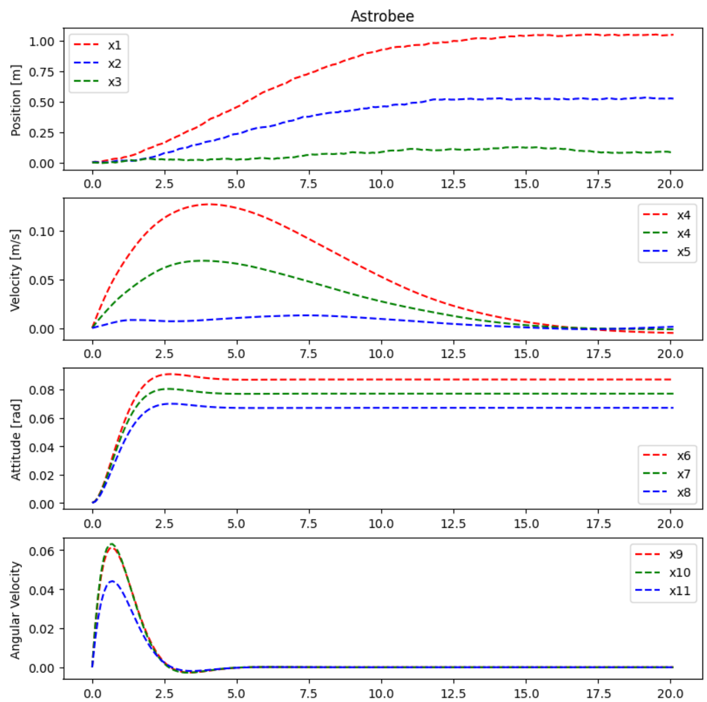

Figure 10: Perturbed states with Q Small image

3 Conclusion

In this assignment, we explored the implementation and tuning of an LQR controller for the Astrobee system. The primary goal was to design a controller that effectively tracks a specified reference while respecting performance and constraint requirements. We analyzed the effect of different LQR tuning parameters on the controller's performance. Through a series of experiments, we observed how modifications to the cost matrices Q and R influenced the system's response. The final tuned controller met the performance specifications and constraints effectively.

We then explored the implementation of the Kalman filter, understanding the importance of the process noise covariance matrix and the measurment noise

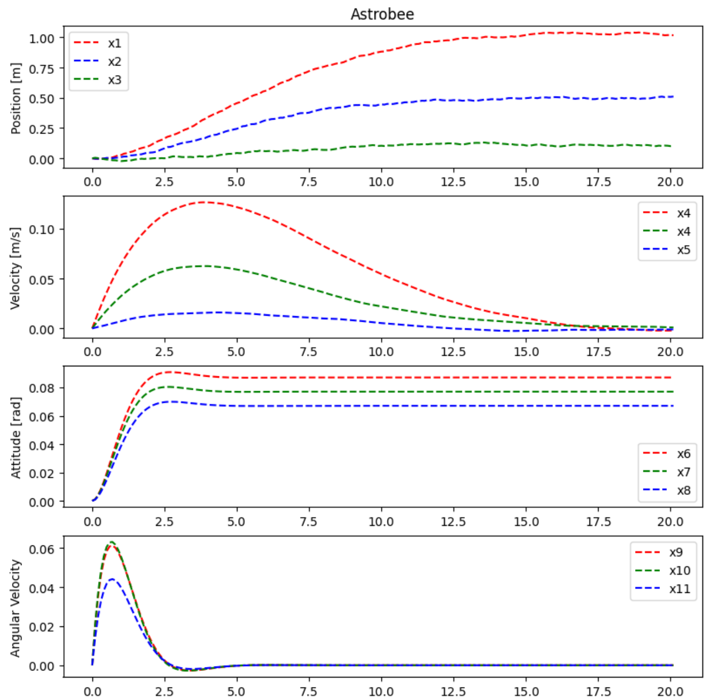

Figure 11: Perturbed states with R Small image

covariance matrix. Furthermore, our analysis showed how varying Q and R affects control smoothness and responsiveness. Small R values lead to more aggressive control inputs, while small Q values result in smoother control actions but potentially less precise state tracking.
In conclusion, this study highlights the critical role of tuning parameters in optimizing LQR controllers and tuning kalman filter for complex systems.