RBE502: Robot Control

Project Phase 2

Student: Dhruv, Agrawal

Problem Statement

Trajectory Following with PD/PID Controller. And Trajectory Following with LQR Controller or MPC

In this phase, you will integrate the trajectory generator and controller to simulate the quadrotor flying in the space tracking a trajectory. Combine with PD controller you designed in previous step. In addition you are expected to implement a LQR or MPC controller for the quadrotor, combine with trajectory generation system designed at previous step, see if the results are better compared to using PID/PD controller.

You are given 2 trajectory generators:

1. circle.m

A helix in the xy plane of radius 1 m centered about the point (0.5, 0, 0) starting at the point (1, 0, 0). The z coordinate should start at 0 and end at 0.5 . The helix should go counter-clockwise.

2. diamond.m

A "diamond helix" with corners at (0, 0, 0), (0, $\sqrt{2}$, $\sqrt{2}$), (0, 0, 2$\sqrt{2}$), and (0, $-\sqrt{2}$, $\sqrt{2}$) when projected into the yz plane, and an x coordinate starting at 0 and ending at 1 . The quadrotor should start at (0, 0, 0) and end at (1, 0, 0).

You can switch between this 2 different trajectory generators by modifying the trajhandle in the test_trajectory.m file

Submission

The students are expected to submit the following files:

1. lqr_controller.m or mpc_controller.m;

2. final report in pdf form, reporting the tracking results of the quadrotor

LQR Implementation

The state vector of the quadrotor is taken as:

$$X = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \phi \\ \theta \\ \psi \\ p \\ q \\ r \end{bmatrix}$$

where **p** is the angular velocity around x,**q** is the angular velocity around y and **r** is the angular velocity around z.

The rate of change of state vector is then calculated as:

$$X = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}$$

where $\dot{p}$, $\dot{q}$ and $\dot{r}$ are the angular accelerations around x,y and z.
For calculating $\dot{X}$,
we have $\dot{x}$, $\dot{y}$ and $\dot{z}$ in the state vector, so we can directly take it from there.
$\ddot{x}$, $\ddot{y}$, $\ddot{z}$ can be calculated as:

$$\ddot{x} = g(\theta \cos(\psi) + \phi \sin(\psi))$$

$$\ddot{y} = g(\theta \sin(\psi) - \phi \cos(\psi))$$

$$\ddot{z} = \frac{u_1}{m} - g$$

$\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$ are calculated using the equation:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & -\cos(\phi)\sin(\theta) \\ 0 & 1 & \sin(\phi) \\ \sin(\theta) & 0 & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

where p,q and r and the angle values can be taken from the state vector X. The values of $\dot{p}$,$\dot{q}$ and $\dot{r}$ can be found from the equation:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = I^{-1} \begin{bmatrix} 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ \gamma & -\gamma & \gamma & -\gamma \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix}$$

Now we can write the state space equation as:

$$\dot{X} = f(X, U)$$

We can use Taylor's approximation to Linenarize the above equation to get:

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

where A is the Jacobian obtained by partially differentiating f(X,U) with X and B is the Jacobian obtained by partially differentiating f(X,U) with U.

The Q and R matrices have been calculated by trial and error method where

$$Q = diag([20, 20, 20, 1, 1, 1, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01])$$

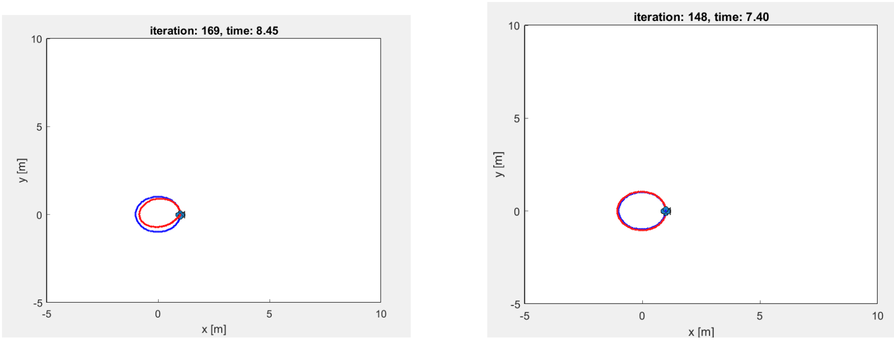
$$R = diag([0.001, 0.1, 0.1, 0.1])$$

which penalizes the cost function more for position and velocity errors in the state and penalizes the control of x,y and z moments.

The results with respect to PD control and LQR control are given in the next subsection.

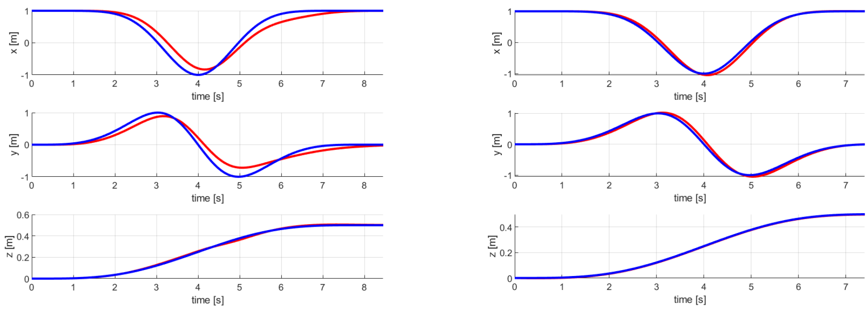
Results

1. **Circle:** The results for the circular trajectory comparing both PD control and LQR control are given below:



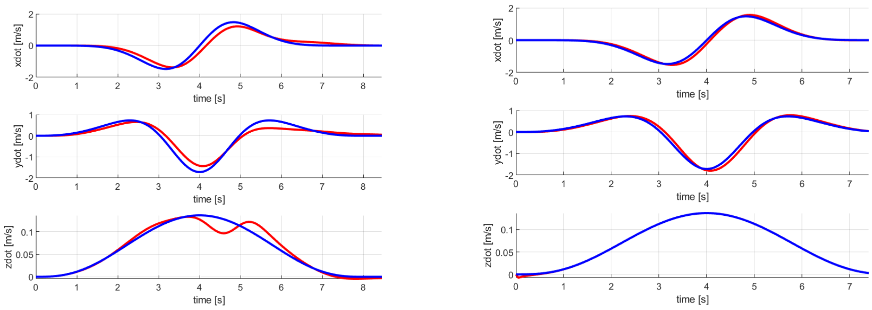
(a) PD Control (b) LQR Control

Figure 1: Circular Trajectory



(a) PD Control (b) LQR Control

Figure 2: Position Tracking

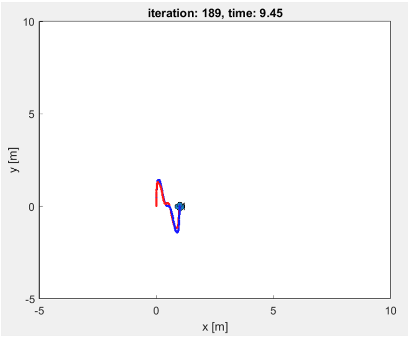


(a) PD Control (b) LQR Control

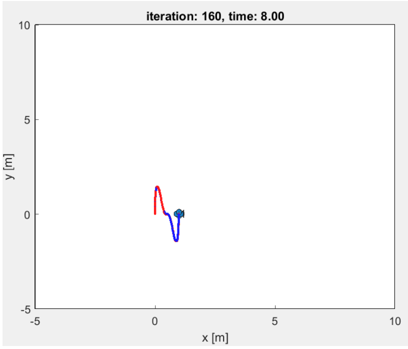
Figure 3: Velocity Tracking

As seen from the above comparison, the LQR tracks the circular trajectory in a better way than the PD control method.
The desired and current position and velocity curves are much closer in LQR graphs than in PD controller graph.

2. **Diamond:** The results for the diamond trajectory comparing both PD control and LQR control are given below:

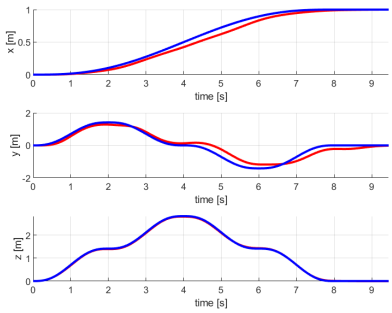


(a) PD Control

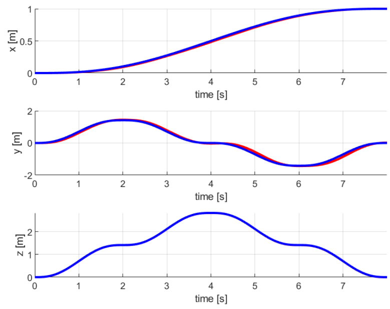


(b) LQR Control

Figure 4: Diamond Trajectory

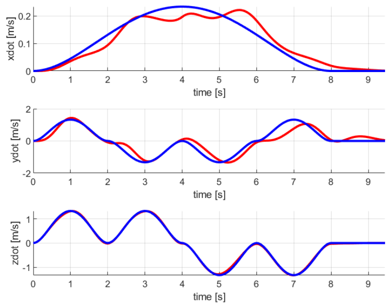


(a) PD Control

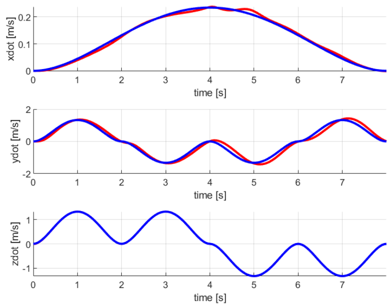


(b) LQR Control

Figure 5: Position Tracking



(a) PD Control



(b) LQR Control

Figure 6: Velocity Tracking

Similar to the circular trajectory case, the LQR tracks the diamond trajectory in a better way than the PD control method.
The desired and current position and velocity curves are much closer in LQR graphs than in PD controller graph. From the above comparisons, we can see that LQR minimizes the cost function based on the Q and R values and provides an optimal control solution that tracks the desired trajectory closely by taking advantage of the full state feedback control.
However, LQR has a few drawbacks too.
LQR requires solving the Riccati Equation which is mathematically complex and comaputationally expensive to solve especially for high dimensional systems.
LQR assumes a perfectly known system model. If the system model is inaccurate or subject to significant uncertainties, the performance of the LQR controller may degrade or become suboptimal.