Report:

Part 1:

The Discretized equations of the system dynamics are as follows.

$$x_{n+1} = x_n + v_x dt$$
$$v_{x(n+1)} = v_{x(n)} + \frac{(-(u_1 + u_2)\, sin\,\theta\,)}{m} dt$$
$$y_{n+1} = y_n + v_y dt$$
$$v_{y(n+1)} = v_{y(n)} + \frac{((u_1 + u_2)\, cos\,\theta\, - mg)}{m} dt$$
$$\theta_{n+1} = \theta_n + \omega dt$$
$$\omega_{n+1} = \omega_n + \left(\frac{r\,(u_1 - u_2)}{I}\right) dt$$

For the robot with the above system dynamics to stay at rest forever, the following control is found.

$$[u_1^\star,\, u_2^\star] = \left[\frac{mg}{2},\, \frac{mg}{2}\right]$$

By analyzing the system equations, it cannot be moved in x-direction, with $\theta = 0$ as sin(0) is 0. Intuitively, if the orientation is zero, the control only acts in y-direction. Therefore, the robot cannot moved in x-direction from rest.

Similarly, with $\theta = pi/2$ the robot cannot remain at rest. There is no force to balance the control in X-direction and keep the robot at rest.

Part 2:

On linearizing the dynamics, following equations are obtained. Since the system should remain at rest, the A, B values of the linearized dynamics are constant.

$$z_{n+1} - z_{*} = A\left(z_n - z^*\right) + B\left(u_n - u^*\right)$$

$$A_n = \begin{pmatrix} 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 \\ 0 & 0 & 1 & 0 & 0 & dt \\ 0 & 0 & -\frac{(u_1+u_2)cos\theta}{m} dt & 1 & 0 & 0 \\ 0 & 0 & -\frac{(u_1+u_2)sin\theta}{m} dt & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$B_n = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\frac{sin\theta}{m} dt & -\frac{sin\theta}{m} dt \\ \frac{cos\theta}{m} dt & \frac{cos\theta}{m} dt \\ \frac{r}{I} & -\frac{r}{I} \end{pmatrix}$$

The following control law and cost function is used.

$$u = u^* + K\left(z_n - z^*\right)$$

$$\min \frac{1}{2}(x_N - \bar{x}_N)^T Q_N (x_N - \bar{x}_N) + \sum_{n=0}^{N-1} \frac{1}{2}(x_n - \bar{x}_n)^T Q_n (x_n - \bar{x}_n) + \frac{1}{2} u_n R_n u_n$$

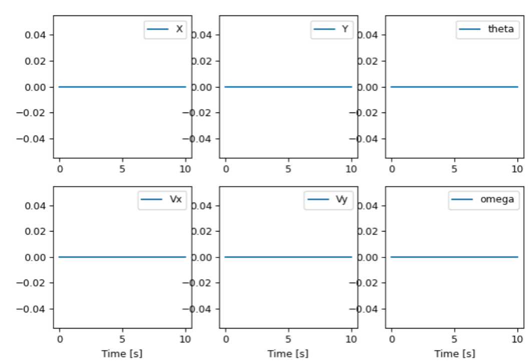
where,

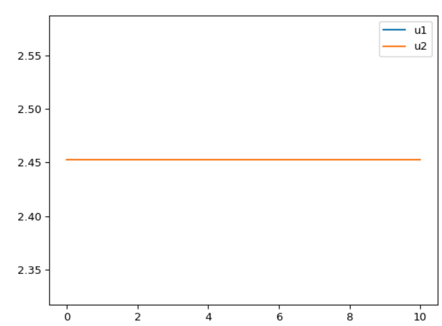
$[u_1^\star,\, u_2^\star] = \left[\frac{mg}{2},\, \frac{mg}{2}\right]$ and $z^* = [0, 0, 0, 0, 0, 0]$.

Q, R are weights of cost for the states and control.

Output:

Plots of states and control. It can be seen that the states and control are constant.





Part 3:

Since the states of the system are changing at every time step according to the trajectory, the dynamics change along with the linearization of dynamics. It is as follows.

$$z_{n+1} - z_{*} = A\left(z_n - z^*\right) + B\left(u_n - u^*\right)$$

$$A_n = \begin{pmatrix} 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 \\ 0 & 0 & 1 & 0 & 0 & dt \\ 0 & 0 & -\frac{(u_1+u_2)cos\theta}{m}dt & 1 & 0 & 0 \\ 0 & 0 & -\frac{(u_1+u_2)sin\theta}{m}dt & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} B_n = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\frac{sin\theta}{m}dt & -\frac{sin\theta}{m}dt \\ \frac{cos\theta}{m}dt & \frac{cos\theta}{m}dt \\ \frac{r}{I} & -\frac{r}{I} \end{pmatrix}$$

where,

$$z^* = [cos\omega t,\ -bsin\omega t,\ sin\omega t,\ bcos\omega t,\ 0,\ \omega]$$

$$u^* = \left[-0.5mg + mb^2 sin\omega t,\ -0.5mg + mb^2 sin\omega\, t\right]$$

Backward Ricatti equations are solved, and the following control is used,

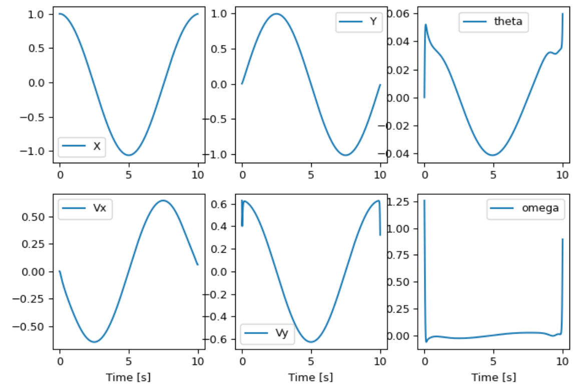
$$u = u^* + K\left(z_n - z^*\right) + k$$

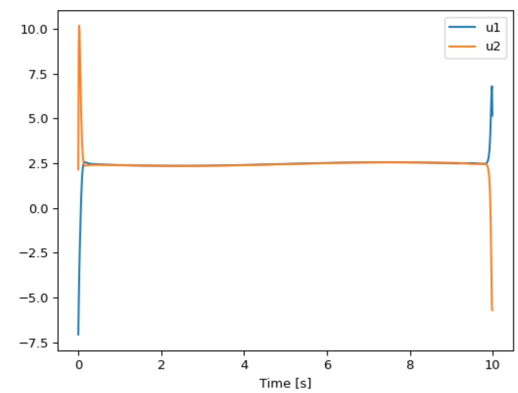
Where K_gains and k feedforward are obtained from the ricatti equations.

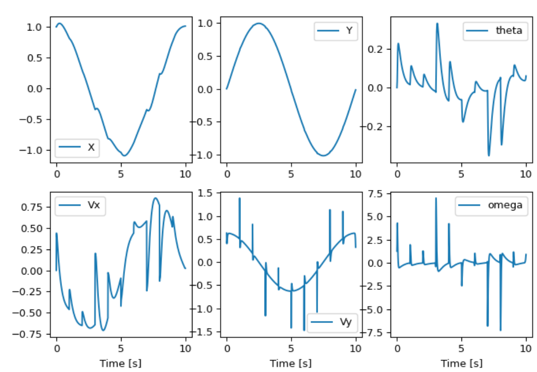
$$K_n = -(R_n + B_n^T P_{n+1} B_n)^{-1} B_n^T P_{n+1} A_n$$
$$P_n = Q_n + A_n^T P_{n+1} A_n + A_n^T P_{n+1} B_n K_n$$
$$k_n = -(R_n + B_n^T P_{n+1} B_n)^{-1} B_n^T p_{n+1}$$
$$p_n = q_n + A_n^T p_{n+1} + A_n^T P_{n+1} B_n k_n$$

Output:

According to the plots, it can be verified that the current LQR control can indeed track a circle with and without disturbances. The orientation is almost zero throughout the trajectory. i.e between -0.04 to 0.06.

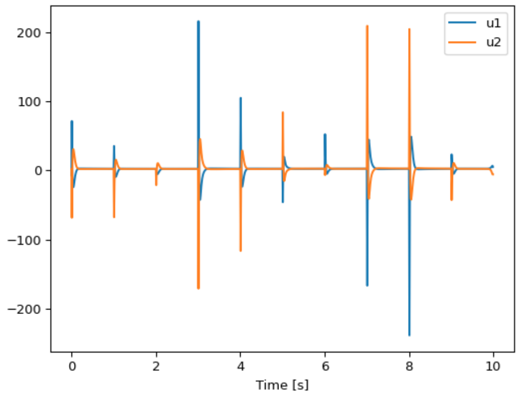






One of the issues with this approach is that there is a need-to-know x_star i.e trajectory. In case the robot deviates from the path, there is no way to improve the tracking.

The benefit is that it can be considered as a simple LQ tracking problem on linearizing the dynamics and it can be solved in one iteration.



Part 4:

Cost function:

The following cost function was used, x_goal was considered as x = 3, y = 3 and $\theta = pi/2$ at time t = 5 seconds while the other goal states at the rest of the time steps were zero. U_goal was cosidered as [mg/2, mg/2] throughout the time steps.

$$Cost = \frac{1}{2}(z_N - z_{goal})^T Q_N (z_N - z_{goal}) +$$
$$\frac{1}{2}\sum_{n=0}^{t-1}\left((z_n - z_{goal})^T Q_n (z_n - z_{goal}) + (z_N - z_{goal})^T S_n (u_n - u_{goal})\right)$$

Quadratic approximation:

The following matrices were obtained by quadratic approximation at x_goal and u_goal.

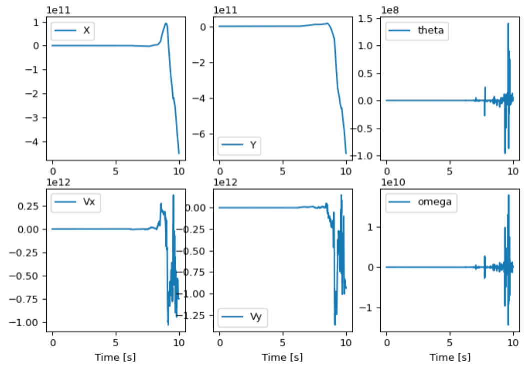
$$Q = \frac{\partial^2}{\partial x^2}(f(x,u)),\ R = \frac{\partial^2}{\partial u^2}(f(x,u)),\ S = \frac{\partial^2}{\partial x.\partial u}(f(x,u))$$

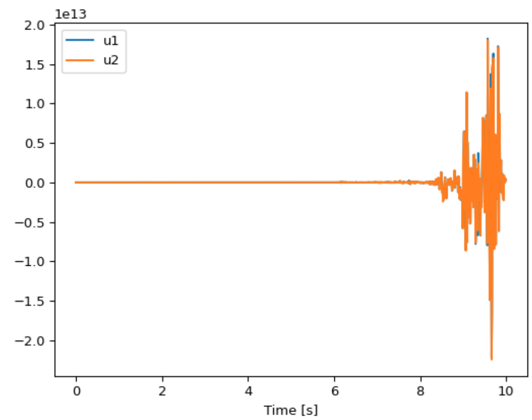
Initially, x_star and u_star were found using a dummy controller. Riccati equations were solved using them. LQ problem was used to obtain a new set of x_star and u_star. The whole process was repeated until the cost was reduced. The following control was used to solve the LQ problem at each iteration.

$$u = u^* + K(z_n - z^*) + \alpha k$$

Results:

The following results were obtained. It can be seen that for the first five seconds the control tries to keep the robot towards the center.





The advantage of this approach is that the path can be contantly optimized inspite of not kmowing the exact trajectory. One of the disadvantage is to be able to find the right cost function for the task.