Optimal Control of Inverted Pendulum on a Hovering Plate in 2D

Vismaya Walawalkar

I. Objective

2D Simulation of optimal control to maintain an inverted pendulum at pi (upright position) on a plane having free motion (X and Y axis).

II. Motivation

The motion of a person inside a moving vehicle is a commonly explored phenomenon. Here, the motion of the person inside an aerial vehicle like a quadrotor is studied. The analysis presented uses the classic inverted pendulum as an undamped version of a passenger inside such a vehicle. The model of the vehicle used is simple rod of uniformly distributed mass in free space. The analysis uses iLQR algorithm which tries to stabilize the system and maintain the pendulum at π angle with y axis, to determine the trend of the passenger motion inside this vehicle. The main assumption for this system is that it does not rotate in the X-Y plane i.e the rotation of the vehicle around Z axis (yaw) is constant.

The reason for this assumption is that I was unable to efficiently code a system with 8 states (x, y, yaw angle of hovering plate, pendulum angle and corresponding velocities) in Mathematica.I have however include the code with these 8 states in the zip file.

The project submitted has 6 states (x, y, pendulum angle and corresponding velocities).

III. Model

The world frame (W) translated to points (X, Y) to obtain vehicle frame (Q).

The configuration variables used are as follows: x : x coordinate of the system. Located at the centre of the rod representing hovering plate. This is also the point where the pendulum is attached to the hover plate.

y : y coordinate of the system. Located at the centre of the rod representing hovering plate. This is also the point where the pendulum is attached to the hover plate.

θ_p : The angle the pendulum makes with the Y axis of Q.

The mechanical parameters to the system are the masses of pendulum and vehicle as $m_p and m_q$ respectively and the length of the pendulum and width of the system given by L_p, L_q respectively.

The inputs to the system are in terms of acceleration applied (as thrusts / forces) at the 2 ends of the rod. Let these be denoted as $[u1, u2]$.

The simplified system dynamics considered are:

$$\ddot{x} = u2$$

$$\ddot{y} = u1$$

$$\ddot{\theta}_p = (g + u1)Sin(\theta_p) - (u2)Cos(\theta_p)$$

IV. Algorithm

Strategy: iLQR

Cost functions used: 1) The system cost is considered with

$$Q = DiagonalMatrix[200, 0, 200, 0, 100.0]$$

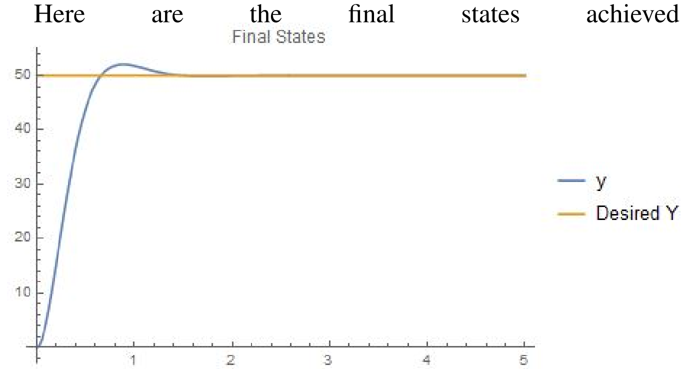
$$R = DiagonalMatrix[0.3, 0.3]$$

The cost matrices Q and R for Riccati equations are identity matrices of appropriate dimensions.

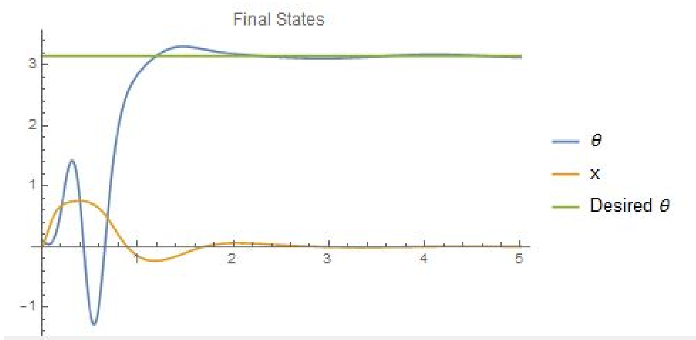
V. Results

The desired position is that the vehicle should begin at ground level (= 0) and hover at 50mtrs above ground level. The pendulum initially at position zero should swing up and settle in inverted position (= π). There is no desired fixed position on x as such a constraint may often lead to a singularity. In other words, motion in x direction is necessary to maintain the inverted pendulum while the swing up can be attributed to the acceleration in Y direction.

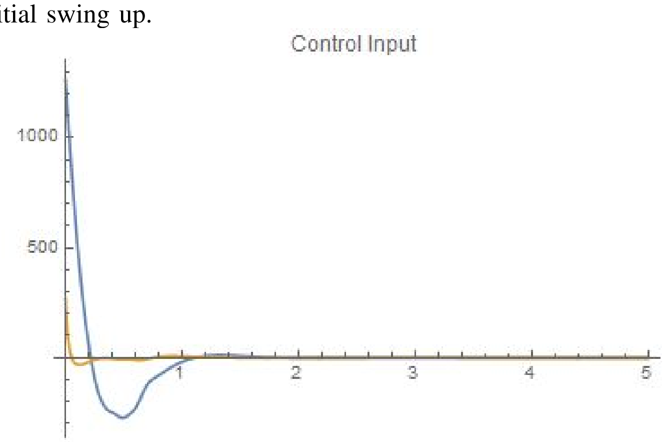
Here are the final states achieved



Y stabilizes to the desired position in a settling time of a



θ stabilizes to the desired position in a settling time of about 2.5 seconds The initial graph uptil time 1.2 seconds is the initial swing up.



The initial inputs required by the system can be seen to be very high and often not achievable, here the discussion on input saturation can be introduced (treatable by applying SAC with energy swing up as initial input and using LQR near equilibrium)

VI. Limitations

1. The system is simplified in terms of it's dynamics to enable evaluation in Mathematica as it has a large state space of 6-8 states.

2. The input saturation is not considered here. It can be observed that initial u1 value is around $1000 m/s^2$.

3. Due to limited experience and time constraints on completion of this project, implementation of SAC for this system could not be achieved.

4. The code takes 15 minutes to run thus there is scope to write this program in a much more efficient way.