Sujay Nayak Anil
ECE Department
NYU Tandon School of Engineering
N10462423 (san440)

INTRODUCTION

The following report offers a detailed review of the work submitted for the final coursework of the class *"Optimal and Learning Control for Robotics"*. The work presented for the coursework is based on the implementation of an iLQR regulator on a two planar arm to execute a desired trajectory. The objective of the project was to control a two planar arm in order to block an incoming punch. Along with the iLQR regulator, a version of MPC was used to compensate for the noise in the system modelling.

Part A. System model

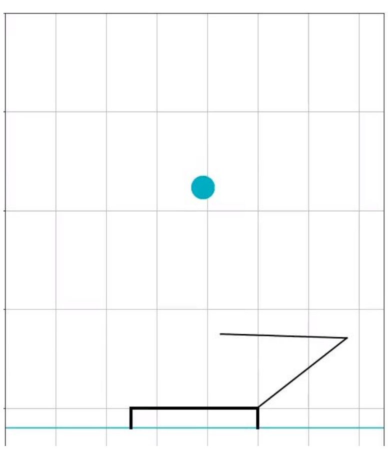


Figure. 1 System Model

The state space of the model is given by :

$$x = \begin{pmatrix} q_1 \\ q_2 \\ \dot{q}_1 \\ \dot{q}_2 \end{pmatrix}$$

Since the system is continuous, the state dynamic is given by:

$$x(i+1) = x(i) + x(i)' * dt$$

where x_dot is given by :

$$x\_dot = \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \ddot{q}_1 \\ \ddot{q}_2 \end{pmatrix}$$

Here q_1' and q_2' are known from the previous states

And q_1'' and q_2'' are determined by the applied input control and the system dynamics :

$$B(q) * q'' = \tau - C(q, q') - K_v * q'$$

where B is given by :

$$B = \begin{pmatrix} \cos(x_2) + \frac{833}{500} & \frac{\cos(x_2)}{2} + \frac{333}{1000} \\ \frac{\cos(x_2)}{2} + \frac{333}{1000} & \frac{333}{1000} \end{pmatrix}$$

The inertial matrix is based on certain predefined parameters.

Email : *san440@nyu.edu*
Uploaded on May 19th 2020

For the following implementation, the parameters chosen are :
$M\_l_1 = M\_l_2 = 4$ kg
$I\_l_1 = I\_l_2 = 0.05$ Nm
$Len\_l_1 = Len\_l_2 = 0.25$ m
$Len\_a_1 = Len\_a_2 = 0.5$ m

Also , $K\_v_1 = K\_v_2 = 1$ Ns/m is chosen to account for friction.

Part B. Process model

At the beginning, A random trajectory is generated for a punch and the system is initialized at an arbitrary point, x_0 . Based on the trajectory of a punch, the point of interception and time to intercept is predicted. The horizon length is calculated based on the time to intercept. The inverse kinematics of the two planar arm is determined based on the point of interception.

Here, the system dynamics is non-linear in nature. Hence, iLQR regulator was chosen to control the system. In iLQR regulator, the system dynamics is linearized w.r.t to a certain guess of the state and the corresponding control. The guess of the states is computed prior to the optimization based on an arbitrary guess of the control inputs. The system dynamics will be linearized to a form of :

$$x(i+1) = A\,x(i) + B\,u(i) + C(i)$$

Since the above equation is of a different notation compared to the regular riccati equations, an altered form of the optimization needs to be used.

K(i) = - inv(R+ B'*S(i+1)*B)*B'*S(i+1)*A

k(i) = - inv(R+ B'*S(i+1)*B)*B'*
(S(i+1)*C + M)

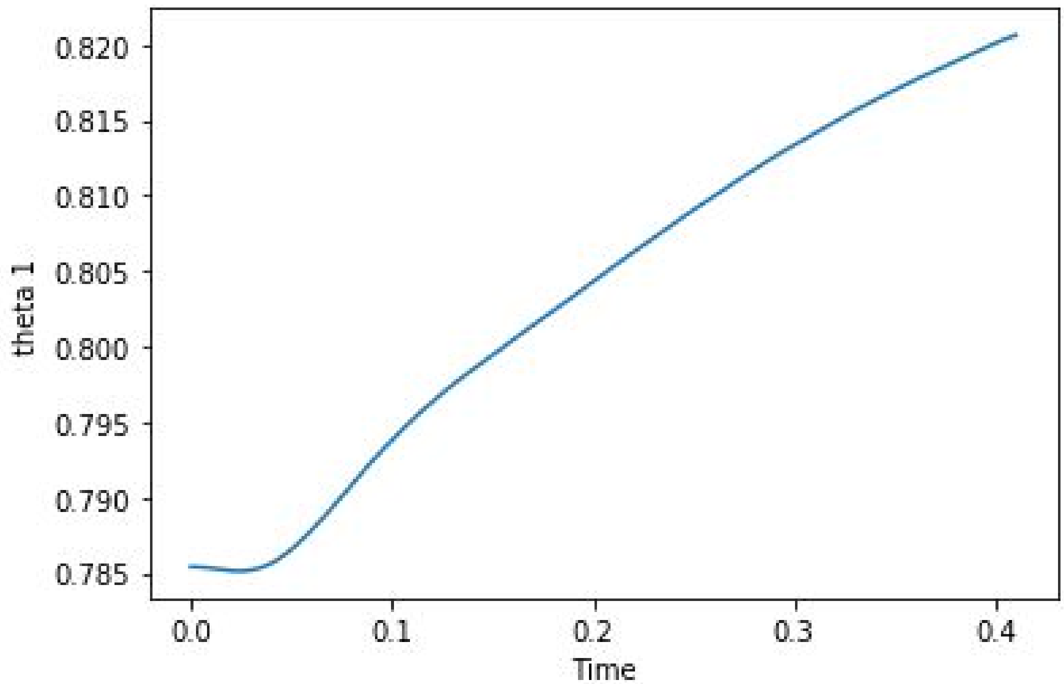
where S and M are quadratic cost coefficients.

In the linearized system dynamics,
$A = I + f_x*dt$
$B = f_u*dt$
$C = (f(x,u) - f_x*x - f_u*u)*dt$
where x and u are based on the initial guess

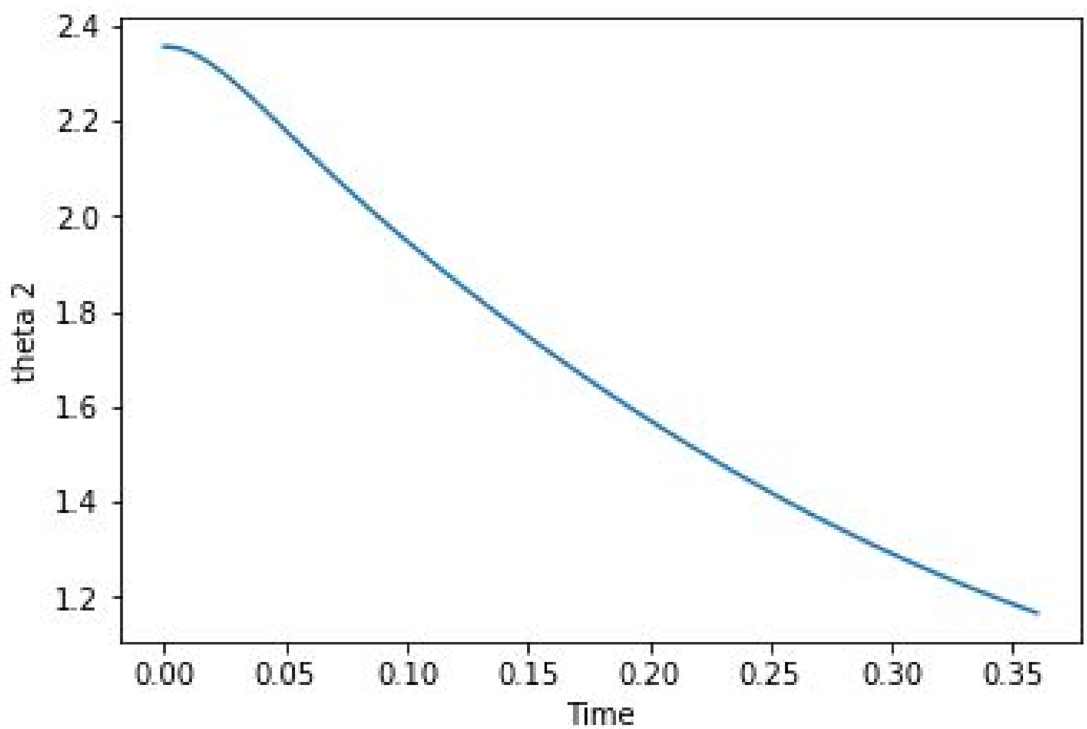
Once the optimization is made for one iteration, the new guess of the state and control is updated based on the current trajectory. The iLQR is run for several iterations in a similar fashion. The updation of the guesses ensures the convergence of the controller towards the optimal policy.

In the following implementation, for each step, the iLQR controller is called upon to update feedback and feedforward gains based on the current state of presence. By doing so, a model predictive control was applied to compensate for any deviations from the optimal trajectory.

Plot for *theta 1* :

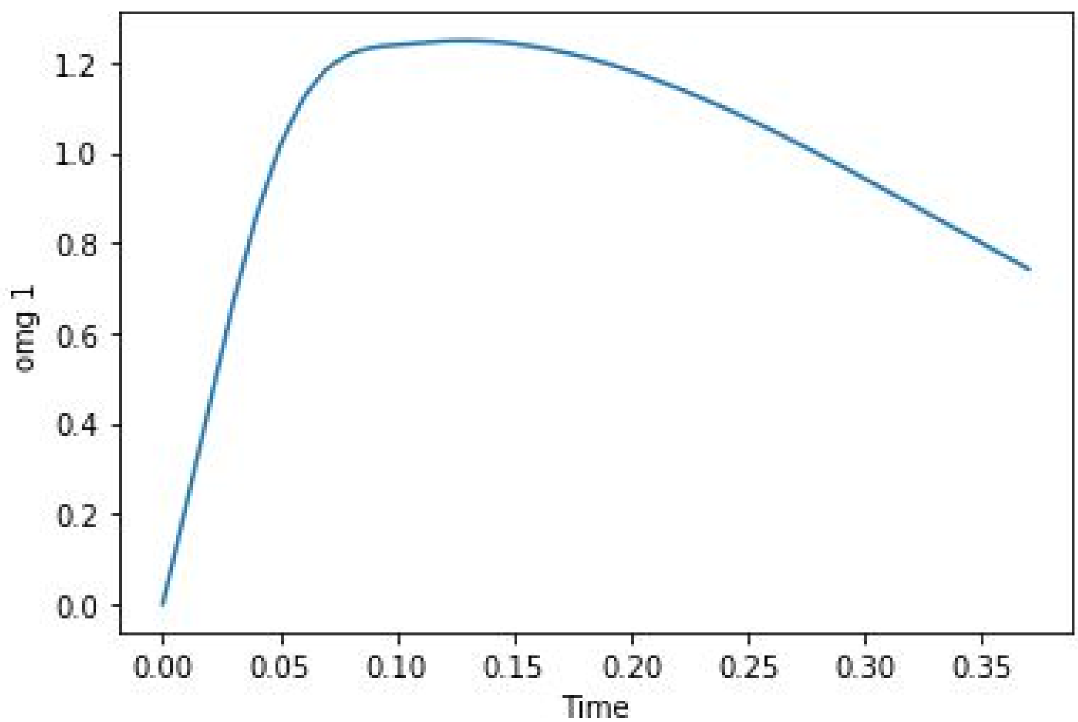


Plot for *theta 2*:

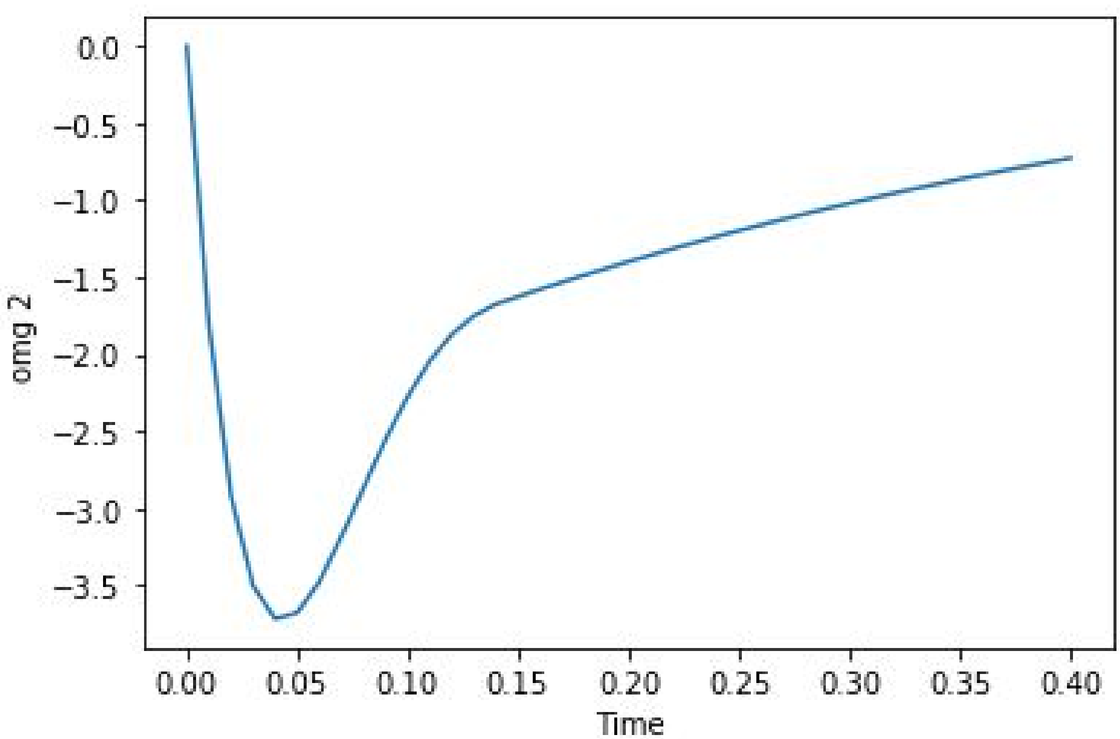


Email : *san440@nyu.edu*

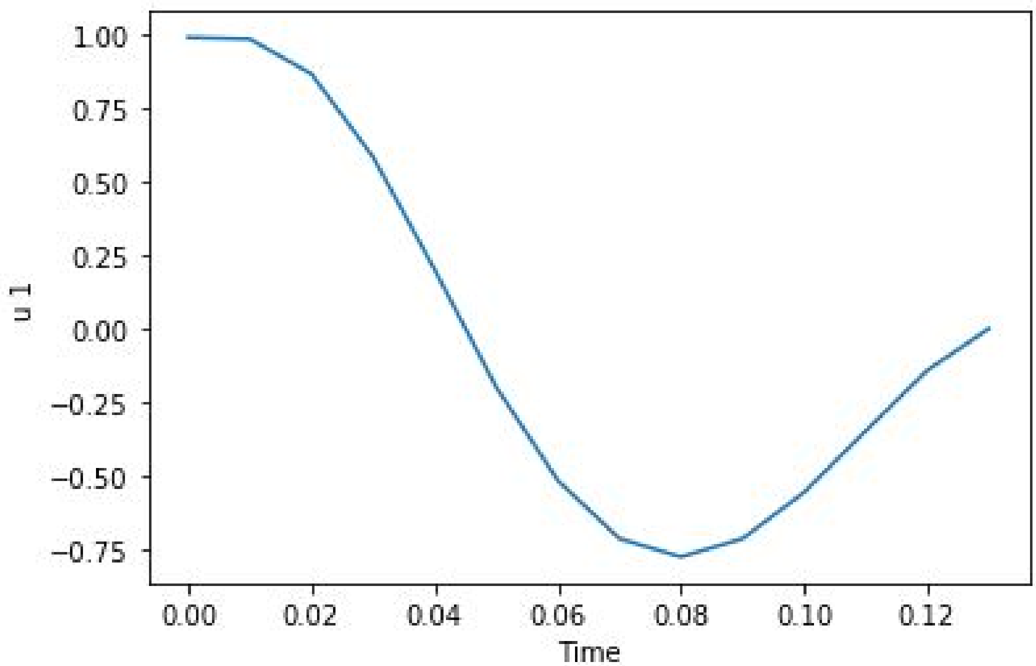
Plot for *omg 1*:



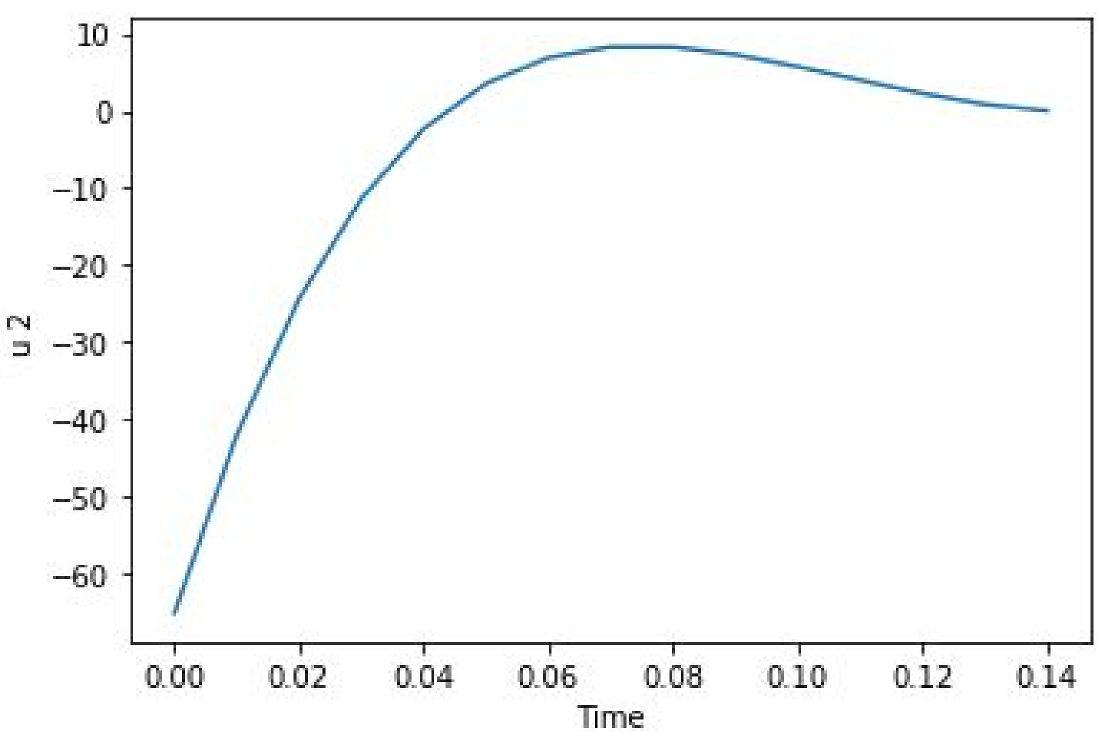
Plot for *omg 2*:



Plot for *u1*:



Plot for *u2*:



Email : *san440@nyu.edu*

Conclusion

Reviewing the results of the controller to block an incoming punch, the controller was successfully in generating the right sequence of the controls to block the incoming punch. Though occasionally, we could also observe a few punches (~1 in 10) not being defended. In such cases, the two planar arm would fail to intercept the punches and miss the rendezvous event at a short margin. Also, when the velocity of the punches are too high (7.5 to 10 ms^{-1}), the controller would also fail to intercept due to the shorter horizon lengths.

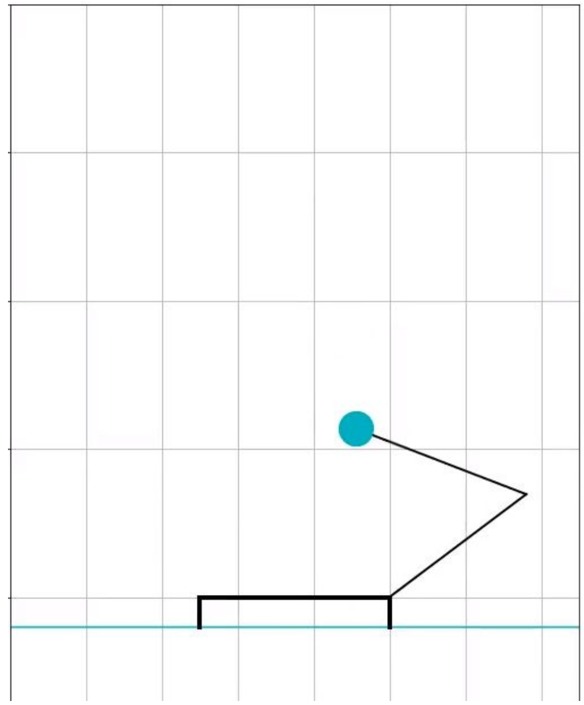


Figure. 2 A punch intercepted

Though iLQR with MPC seems to be an attractive solution for nonlinear systems, it is significantly expensive to compute for higher horizon length. In the following work presented, due to shorter horizon length (~ 8 to 15), the computational was much feasible with lower tendency for policy lag.

Email : *san440@nyu.edu*