CS 6756 HW #1 Writeup

Tyler King

September 22, 2023

1 2-link Dynamics

To sanity check the accuracy of our model, we ran numerous tests with various initial conditions and certified that our dynamics followed:

$$\theta(t+dt) = \theta(t) + dt \cdot \dot{\theta}(t) + \frac{1}{2}dt^2 \cdot \ddot{\theta}(t)$$
$$\dot{\theta}(t+dt) = \dot{\theta}(t) + dt \cdot \ddot{\theta}(t)$$

where we used the fake dynamics method $\tau_i = ml^2\ddot{\theta}_i$ to obtain our angular acceleration for each joint.

We make a few key assumptions about our model

1. The entirety of the mass of our manipulator falls at the end of the link, which allows us to utilize the fake dynamics method due to the moment of inertia of such an object.
2. The manipulator lies flat on a table (and additionally is frictionless), so we can assume there are no external forces at play (such as gravity or friction) besides the applied torques.
3. The torque applied to each joint is independent (i.e. $\ddot{\theta}_1$ is applied independently to θ_1 and similarly with θ_2).
4. The model can be approximated linearly (this would be important for LQR).

2 Linear Quadratic Regulator

We include 3 images with various initializations to showcase the performance of our LQR implementation.

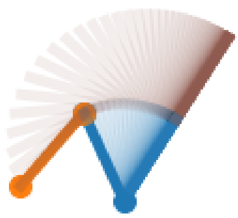

Figure 1: Trace of LQR dynamics for arm starting at $x = \begin{bmatrix} 2 & 2 & 0 & 0 \end{bmatrix}$ with $\theta_{ref} = \begin{bmatrix} 1 & 0 \end{bmatrix}$.

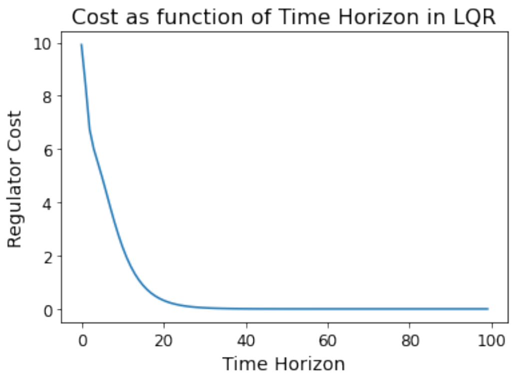

Figure 2: Cost plot over time horizon of LQR dynamics for arm starting at $x = \begin{bmatrix} 2 & 2 & 0 & 0 \end{bmatrix}$ with $\theta_{ref} = \begin{bmatrix} 1 & 0 \end{bmatrix}$.

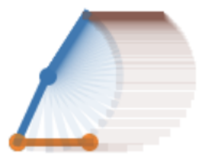

Figure 3: Trace of LQR dynamics for arm starting at $x = \begin{bmatrix} -2 & 2 & 1 & -1 \end{bmatrix}$ with $\theta_{ref} = \begin{bmatrix} 1 & -1 \end{bmatrix}$.

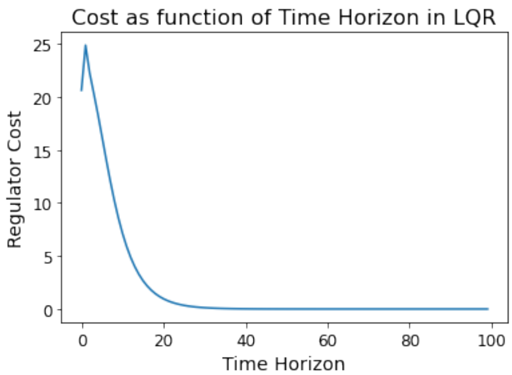

Figure 4: Cost plot over time horizon of LQR dynamics for arm starting at $x = \begin{bmatrix} -2 & 2 & 1 & -1 \end{bmatrix}$ with $\theta_{ref} = \begin{bmatrix} 1 & -1 \end{bmatrix}$.

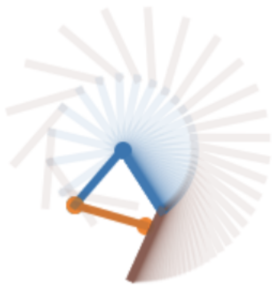

Figure 5: Trace of LQR dynamics for arm starting at $x = \begin{bmatrix} 4 & 2 & 0 & 10 \end{bmatrix}$ with $\theta_{ref} = \begin{bmatrix} -1 & -1 \end{bmatrix}$.

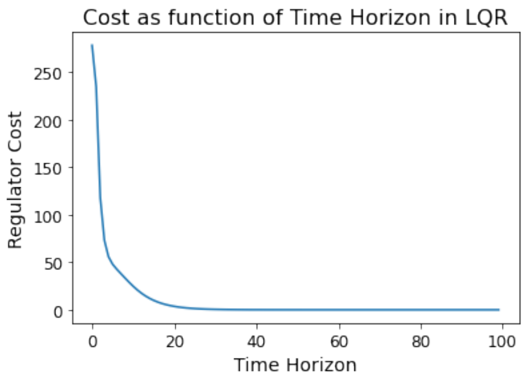

Figure 6: Cost plot over time horizon of LQR dynamics for arm starting at $x = \begin{bmatrix} 4 & 2 & 0 & 10 \end{bmatrix}$ with $\theta_{ref} = \begin{bmatrix} -1 & -1 \end{bmatrix}$.

Across all these figures, we note that the cost curve typically appears to be similar to a plot of exponential decay. Typically, there is a linear portion at the start before it seems to smoothly decay to 0 cost.

2.1 Control cost to 0

Despite removing the control cost penalty, we note that our 2-link manipulator still effectively converged to some target state. When looking at the cost function per time horizon when there's no cost penalty, we notice a large spike initially, which we suspect is due to no u penalty causing an explosion in $\dot{\theta}$, which initially drives up cost before slowly converging to the final target state.

This also implies that our torques do not go to infinity, which we feel is logical since an infinite torque would cause the model to severely overshoot after one time step, leading to a potentially larger cost.

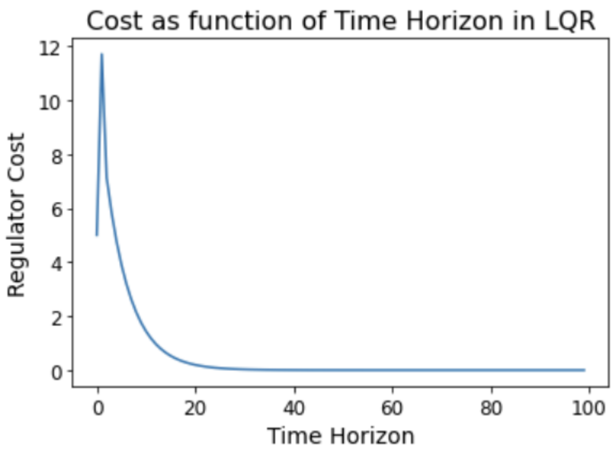

Figure 7: Plot of cost during each time horizon with no control cost penalty.

Figure 8: Trace of LQR dynamics given the same configuration as figure 1 and no control cost penalty.

3 Iterative Linear Quadratic Regulator

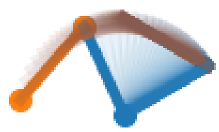

Figure 9: Trace of iLQR dynamics for arm starting at $x = \begin{bmatrix} 2 & 2 & 0 & 0 \end{bmatrix}$ with $\theta_{ref} = \begin{bmatrix} 0 & 0.5 \end{bmatrix}$. Note that our trajectory was initialized randomly using Numpy and 10 iterations of iLQR were applied.

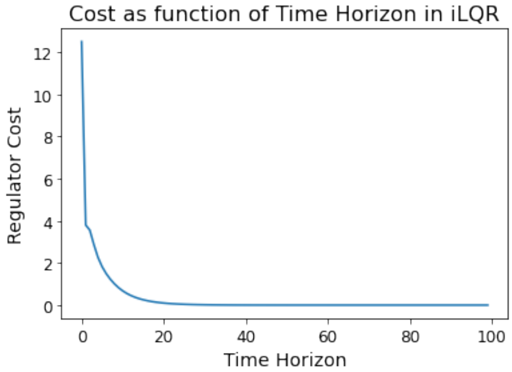

Figure 10: Cost plot over time horizon of iLQR dynamics for arm starting at $x = \begin{bmatrix} 2 & 2 & 0 & 0 \end{bmatrix}$ with $\theta_{ref} = \begin{bmatrix} 0 & 0.5 \end{bmatrix}$.

To enforce the PSD nature of our Q_t matrices, we first diagonalize it into the form $X\Lambda X'$ where X is the orthonormal matrix of eigenvectors. Since our Q_t is symmetric, we can guarantee all eigenvalues are real (and thus when computing eigenvalues/eigenvectors any complex component can be ignored), and finally to enforce PSD, we set negative eigenvalues $\lambda_i \in \Lambda$ to be 0.

To select lambda for Levenberg-Marquedt, we first check the conditional number of our matrix (as defined by $\|Q_t\| \cdot \|Q_t\|^{-1}$), and if it is above some bound (which we arbitrarily select to be 1000), then we continuously increment $Q_t := Q_t + \lambda I$ until the conditional number of Q_t is below 100, where we define $\lambda = 1e{-2}$ (again somewhat arbitrarily defined; these values seemed to work well for our implementation).

4 Extra Credit

For the extra credit, we implemented an $N = 7$ link manipulator, assuming that our target state is of the form $\begin{bmatrix} \theta_1 & \theta_2 & \dots & \theta_7 & 0 & 0 & \dots & 0 \end{bmatrix}$. We then regularize around that state, and show that we have a working LQR and iLQR for this particular manipulator.