Nonlinear Control of a Two-link Planar Manipulator

Jin Hyun (Addy) Park

May 17 2024

1 Introduction

The dynamics of robot manipulators are nonlinear. In particular, manipulators are characterized by trigonometric nonlinearities. Also, additional nonlinearities arise from Coriolis and centrifugal forces, making the control of the manipulator challenging. In this final project, I will take three steps to designing an optimal controller for a two-link manipulator in two dimensions. First, I will use feedback linearization to linearize the system. Then, on this linearized system, I will implement an LQR controller. Then, I will use the control policy given by the LQR controller as an initial guess for optimization using Pontryagin's maximum principle. While the LQR is "optimal" with respect to the transformed control input defined by feedback linearization, it is not necessarily optimal with respect to the actual control input $\mathbf{u}$.

2 Dynamics of a Two-link Planar Manipulator

First, we need to determine the dynamics of the two-link manipulator in two dimensions. The problem setup is as follows:

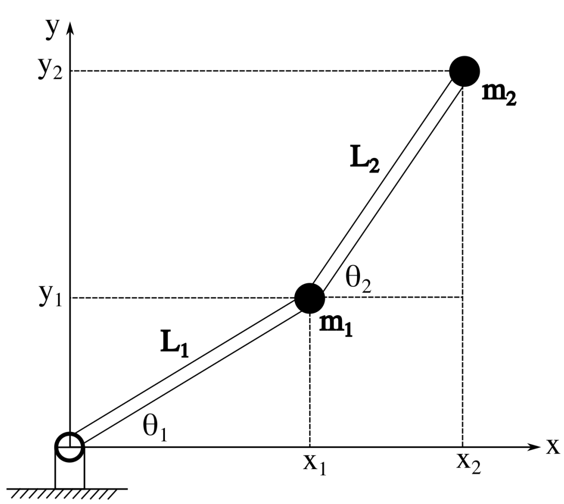

Figure 1: Diagram of two-link manipulator in 2D

Let us define three terms:

$$M(\boldsymbol{\theta}) = \begin{bmatrix} (m_1+m_2)L_1^2 & m_2L_1L_2(\theta_1-\theta_2) \\ m_2L_1L_2cos(\theta_1-\theta_2) & m_2L_2^2 \end{bmatrix}$$

$$C(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) = \begin{bmatrix} m_2L_1L_2\dot{\theta_2}^2 sin(\theta_1-\theta_2) \\ -m_2L_1L_2\dot{\theta_1}^2 sin(\theta_1-\theta_2) \end{bmatrix}$$

$$G(\boldsymbol{\theta}) = \begin{bmatrix} (m_1+m_2)gL_1cos(\theta_1) \\ m_2gL_2cos(\theta_2) \end{bmatrix}$$

where $\boldsymbol{\theta} = \begin{bmatrix}\theta_1 & \theta_2\end{bmatrix}^T$ and $\dot{\boldsymbol{\theta}} = \begin{bmatrix}\dot{\theta}_1 & \dot{\theta}_2\end{bmatrix}^T$.
$M(\boldsymbol{\theta})$ is called the mass matrix, $C(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ is called the Coriolis term, and $G(\boldsymbol{\theta})$ is the gravity term. Using these three terms, the dynamics of the manipulator can be written compactly as the following:

$$\ddot{\boldsymbol{\theta}} = \begin{bmatrix}\ddot{\theta}_1 \\ \ddot{\theta}_2\end{bmatrix} = -M^{-1}(\boldsymbol{\theta}) \left[C(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + G(\boldsymbol{\theta}) - \mathbf{u}\right]$$

where $\mathbf{u} = \begin{bmatrix}\tau_1 & \tau_2\end{bmatrix}^T$. τ_1 and τ_2 are the applied torques on each of the two joints.

3 Feedback Linearization

Feedback linearization can be used to linearize the system. Let us define $\mathbf{v} = \ddot{\boldsymbol{\theta}}$. This allows us to obtain the following *linear* state space system:

$$\dot{\mathbf{x}} = \begin{bmatrix}\dot{\theta}_1 \\ \ddot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_2\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0\end{bmatrix} \begin{bmatrix}\theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2\end{bmatrix} + \begin{bmatrix}0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1\end{bmatrix} \begin{bmatrix}v_1 \\ v_2\end{bmatrix} \tag{1}$$

$$\mathbf{y} = \begin{bmatrix}1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1\end{bmatrix} \begin{bmatrix}\theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2\end{bmatrix} \tag{2}$$

With the system linearized, we can design a control policy $\mathbf{v}$ to control $\mathbf{x} = \begin{bmatrix}\theta_1 & \dot{\theta}_1 & \theta_2 & \dot{\theta}_2\end{bmatrix}^T$. Once we have found $\mathbf{v}$, we can invert the mapping to get the real control input $\mathbf{u}$.

4 Linear Quadratic Regulator with Reference Input

To control $\mathbf{x}$ for the linearized system defined by equations 1 and 2 using the control input $\mathbf{v}$, I chose to use LQR with a reference input. But first, consider the system without a reference (a regulator). Assuming that the full state $\mathbf{x}$ is observed (which is valid for a manipulator since the motors have encoders), the block diagram for LQR looks like the following:

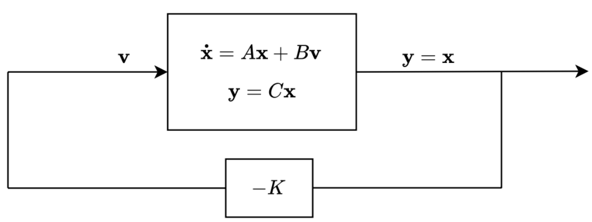

Figure 2: Block diagram of LQR on linearized system without reference input

where matrices A,B, and C are the system matrices of the linear system defined by equations 1 and 2. K is computed using the MATLAB command `lqr()`. The penalty weights used for the state and control effort were `Q=diag([10 1 10 1])` and `R=diag([2 2])`. Note that R is the penalty weighting on the control input $\mathbf{v}$, not $\mathbf{u}$. Therefore, the LQR solution is optimal with respect to $\mathbf{v}$ but not necessarily optimal with respect to the actual control input $\mathbf{u}$. Because this controller is a regulator, it seeks to steer the system to the origin. To steer the system towards a reference state $\mathbf{r}$ instead of the origin, we can write $\mathbf{v} = -K(\mathbf{x} - \mathbf{r})$ (this effectively "shifts" the tracking point from the the origin to $\mathbf{r}$). The gain K is still the same as the system is linear (so it doesn't matter if our operating point is the origin or if it's around some reference state $\mathbf{r}$) and we are not changing the cost function. Also, because the system has two poles at the origin, the steady-state error is guaranteed to converge to zero. The block diagram with the reference input included is given in Figure 3 on the next page.

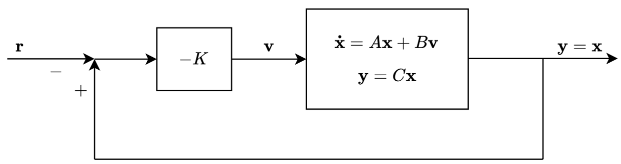

Figure 3: Block diagram of LQR on linearized system with reference input

To simulate the step response (i.e. response to unit reference), I defined a new linear system with system matrices $\tilde{A} = A - BK$, $\tilde{B} = BK$, and $\tilde{C} = C$ with $\mathbf{r}$ as the input and $\mathbf{y}$ as the output and used the `step()` command. Here is the resulting step response with a reference input of $\mathbf{r} = [\pi, 0, \pi/2, 0]$:

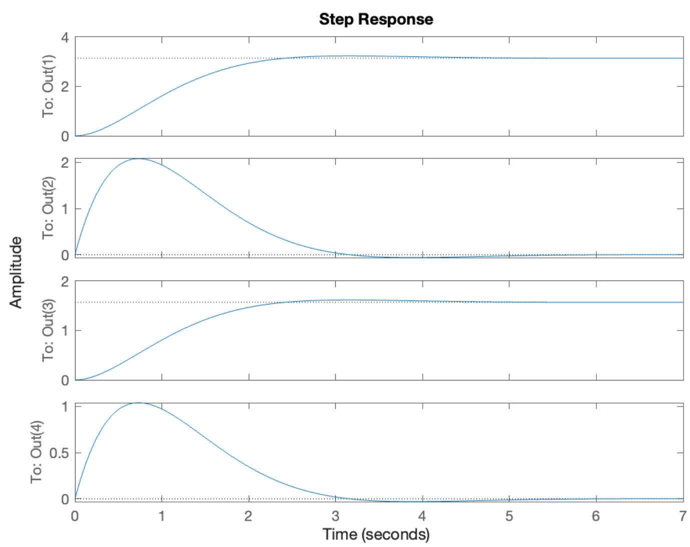

Figure 4: LQR step response of linearized system with reference input

As expected, the steady-state error is zero. The control law that we solved for is a control law for $\mathbf{v}$, not a control law for the the actual physical control input $\mathbf{u}$. To get the corresponding control law for $\mathbf{u}$, invert the mapping we used to linearize the system in Section 3. Recall that the mapping used to linearize the nonlinear system was the following:

$$\mathbf{v} = M^{-1}(\boldsymbol{\theta}) \left[C(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}} + G(\boldsymbol{\theta})) \right] + \mathbf{u}$$

To invert the mapping, solve for $\mathbf{u}$:

$$\mathbf{u} = -C(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}} - G(\boldsymbol{\theta}) + M(\boldsymbol{\theta})\mathbf{v}$$

where $\mathbf{v} = -K(\mathbf{x} - \mathbf{r})$. Now we have $\mathbf{u}$ as a function of $\mathbf{x}$ only, which is the control law. Here is a video of the controller successfully tracking a reference input $\mathbf{r} = [\pi, 0, \pi/2, 0]$: `https://youtu.be/KalPLfjl1b8`. A plot of θ_1 and θ_2 plotted over time is shown in Figure 5 is shown on the next page. As expected, the plots in Figures 4 and 5 match.

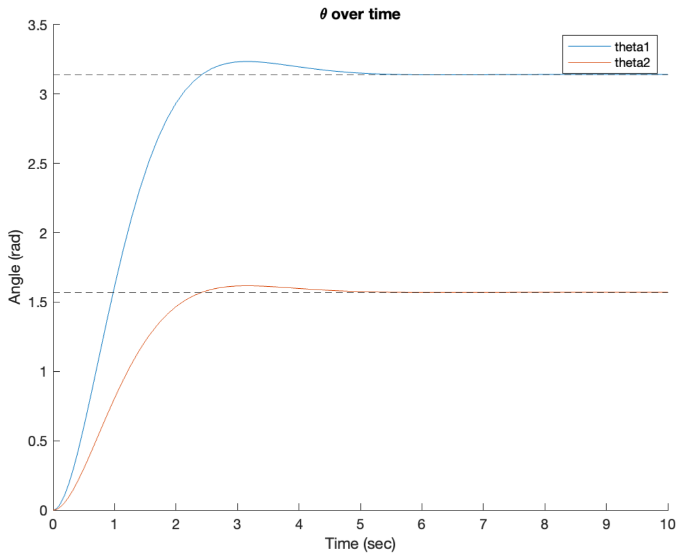

Figure 5: Plot of θ_1 and θ_2 over time

5 Optimization using Pontryagin's Maximum Principle

While the system was successfully controlled using feedback linearizaton and LQR,