# ECE560 Project 2 – Control Design for Two Link Planar Manipulator

Mark Keranen and Jonathon Klapatch

University of Michigan – Dearborn
College of Engineering and Computer Science

# Outline

# 1. Background / Problem Statement

The industrial robot in Figure 1a is modeled by a two-link planar manipulator shown in Figure 1b. The dynamic equations for the two-link manipulator are given by

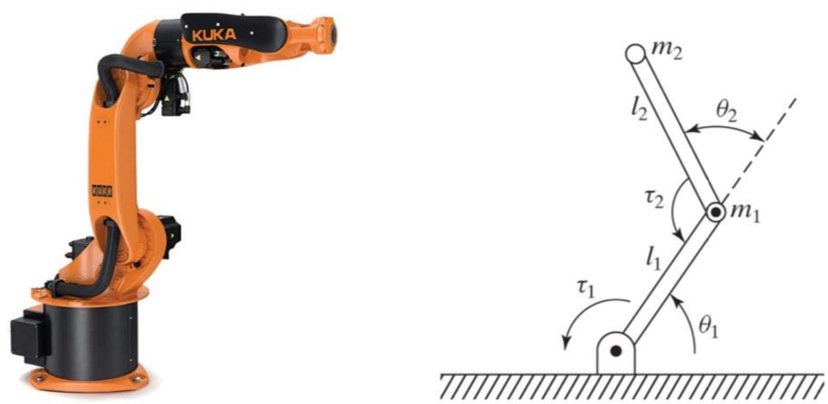
$$\tau_1 = m_2 l_2^2(\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 \cos(\theta_2)\,(2\ddot{\theta}_1 + \ddot{\theta}_2) + (m_1 + m_2) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 \sin(\theta_2)\,\dot{\theta}_2^2 - 2 m_2 l_1 l_2 \sin(\theta_2)\,\dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g \cos(\theta_1 + \theta_2) + (m_1 + m_2) l_1 g \cos(\theta_1)$$

$$\tau_2 = m_2 l_1 l_2 \cos(\theta_2)\,\ddot{\theta}_1 + m_2 l_1 l_2 \sin(\theta_2)\,\dot{\theta}_1^2 + m_2 l_2 g \cos(\theta_1 + \theta_2) + m_2 l_2^2(\ddot{\theta}_1 + \ddot{\theta}_2)$$

*Equation 1: Dynamic Equations for the Two-Link Manipulator Torques*

where $m_1$ and $m_2$ are point masses at the distal end of each link.

Let $m_1$ = 2 kg, $m_2$ = 1 kg, $l_1 = l_2$ = 1m, and g = 9.8m/s$^2$.




(a) Industrial robot (b) Simple 2 DOF model

*Figure 1. A two-link planar manipulator model for an industrial robot.*

The purpose of this project is to model and design several different control systems with which to control the motion and orientation of the manipulator. Once modeled, these control systems will be implemented to drive the dual linkage to a known desired state.

# 2. Control System Design

## Problem 1: Stability Analysis

For the first part we analyze the stability of the several equilibrium states of the system (shown below), in terms of Lyapunov, internal, and bounded input bounded output (BIBO).

State 1: $\begin{bmatrix}\theta_1\\ \dot{\theta}_1\\ \theta_2\\ \dot{\theta}_2\end{bmatrix} = \begin{bmatrix}\pi/2\\ 0\\ 0\\ 0\end{bmatrix}$ State 2: $\begin{bmatrix}\theta_1\\ \dot{\theta}_1\\ \theta_2\\ \dot{\theta}_2\end{bmatrix} = \begin{bmatrix}-\pi/2\\ 0\\ 0\\ 0\end{bmatrix}$ State 3: $\begin{bmatrix}\theta_1\\ \dot{\theta}_1\\ \theta_2\\ \dot{\theta}_2\end{bmatrix} = \begin{bmatrix}\pi/2\\ 0\\ \pi\\ 0\end{bmatrix}$ State 4: $\begin{bmatrix}\theta_1\\ \dot{\theta}_1\\ \theta_2\\ \dot{\theta}_2\end{bmatrix} = \begin{bmatrix}-\pi/2\\ 0\\ \pi\\ 0\end{bmatrix}$

## Intuitive Stability in Terms of Lyapunov

### State 1

The system at this equilibrium state is not stable. There is not a value ε > 0 that can satisfy the ε - δ requirement. Any perturbation will cause the pendulum to leave State 1 and settle near State 2.

### State 2

The system is stable at this equilibrium point, for any value of ε > 0, we can satisfy ε - δ requirement. The pendulum will always return to equilibrium state 2 even if a small perturbation is applied to the system.

### State 3

The system is not stable at this equilibrium point. There is not a value of ε > 0 that can satisfy ε - δ requirement. Any perturbation applied to the system will cause the pendulum to leave State 3 and settle near State 2.

### State 4

The system is not stable at this equilibrium point. There is not a value of ε > 0 that can satisfy ε - δ requirement. Any perturbation applied to the system will cause the pendulum to leave State 4 and settle near State 2.

## Internal and BIBO Stability

For assessing internal stability of a system, ||x(t)|| is examined as t → ∞. Here, excluding State 2, each state (State 1, 3, 4) is (marginally) stable since ||x(t)|| < M < ∞. Each state approaches Equilibrium State 2, which ends up being less than infinity. State 2 is asymptotically stable because ||x(t)|| → 0 as t → ∞.

For a function to be considered BIBO stable, its transfer function must contain negative real parts to its poles. To determine this, we find the transfer function from which the poles can be inspected as follows.

We have: $\dot{x} = Ax + Bu$ and $y = Cx + Du$

From which we can calculate the transfer function

$$G(s) = C(SI - A)^{-1}B + D$$

The poles of *G(s)* for each state determine BIBO stability at each state.

### State 1

Poles: -6.4969, 6.4969, -2.6153, 2.6153

At State 1, *G(s)* has two positive poles, therefore the system is not BIBO stable.

### State 2

Poles: 0.0000 + 6.4969i, 0.0000 - 6.4969i, 0.0000 + 2.6153i, 0.0000 - 2.6153i

At State 2, *G(s)* has only complex poles with no real positive part, therefore the system is BIBO stable.

### State 3

Poles: -4.7529 + 0.0000i, 4.7529 + 0.0000i, -0.0000 + 3.5749i, -0.0000 - 3.5749i

At State 3, *G(s)* has one positive real pole, therefore the system is not BIBO stable.

### State 4

Poles: -3.5749 + 0.0000i, 0.0000 + 4.7529i, 0.0000 - 4.7529i, 3.5749 + 0.0000i

At State 4, *G(s)* has one positive real pole, therefore the system is not BIBO stable.

## Stability Analysis using Lyapunov Indirect Method

Here we assess the stability of the system via the equation $AX + XA^T + Q = 0$ using the 'lyap' function. The 'lyap' function will only return a matrix if the Lyapunov solution X exists (called 'P' in the figures below). The Q matrix must be a symmetric positive definite matrix with the same size as A. A 4x4 identity matrix was arbitrarily selected.

### State 1

In this state, the Lyapunov solution does not exist, and is therefore unstable.

```
Error using lyap (line 69)
The solution of this Lyapunov equation does not exist or is not unique.

Error in ManipulatorLinearSystem (line 79)
    P = lyap(A2,Q);
```

### State 2

We find that there is asymptotic stability in this instance, this is when the manipulator is hanging straight down.

```
P =

   1.0e+15 *

  -0.0528   -0.0000    0.0613    0.0000
  -0.0000   -1.6366   -0.0000    3.3564
   0.0613   -0.0000   -0.2196   -0.0000
   0.0000    3.3564   -0.0000   -8.2661
```

### State 3

In this state, the Lyapunov solution exists with the following solution and is therefore stable.

```
P =

   1.0e+15 *

    0.0641   -0.0000   -0.1436    0.0000
   -0.0000   -2.6670   -0.0000   -0.7801
   -0.1436   -0.0000   -0.3668   -0.0000
    0.0000   -0.7801   -0.0000   -5.0073
```

### State 4

In this state, the Lyapunov solution does not exist, and is therefore unstable.

```
>> Manipulator
Error using lyap (line 69)
The solution of this Lyapunov equation does not exist or is not unique.
```

## Problem 2: State Feedback Compensator Design and Implementation

- **Discuss the performance of the compensators - transient response and steady state response**

The placement of the poles and whether they contain a complex component determines how quickly the compensator responds and acts to correct the perturbation applied to the system (Figure 2). If the poles are further from the origin, the transient response of the compensator more aggressively counteracts the perturbation to reach the desired steady state but if they are too far, simulation times increase beyond practicality. When poles are near the origin, the transient response is slower and less aggressive, but if they are too near the origin, can fail to counteract gravity resulting in an inability to reach the desired state. In the case of the poles with the complex component, the transient response tends to oscillate about the desired state more than the poles with only real components. Each pole placement eventually results in near zero steady state error.

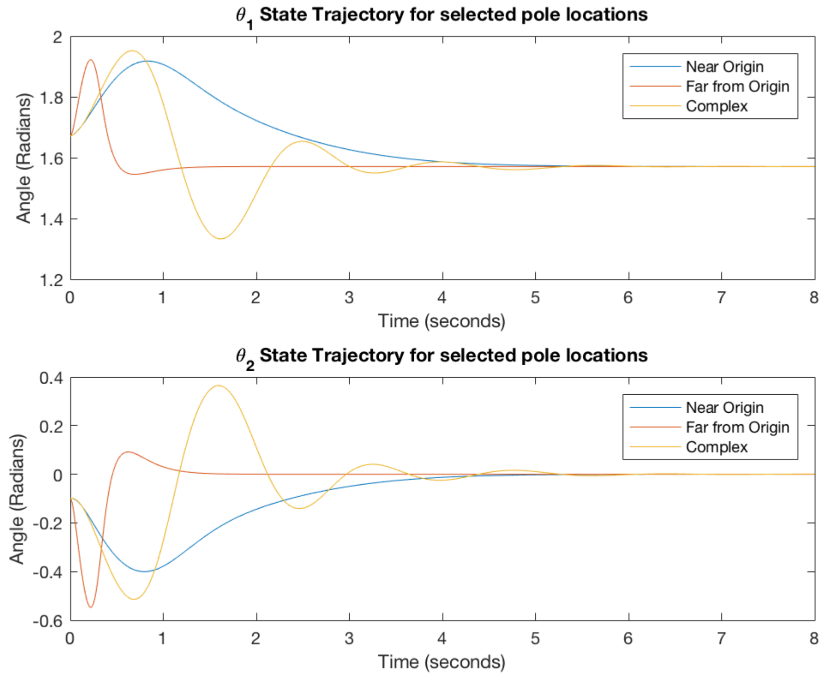



*Figure 2. State trajectories of system with different pole locations*

- Discuss the maximum input requirement

The maximum input requirement occurs at the point in which the system reverses motion and begins to move back towards the desired state (Figure 3). This maximum input requirement is largest in the case where the poles are furthest from the origin and similar for poles about the same distance from the origin whether they contain imaginary components or not.

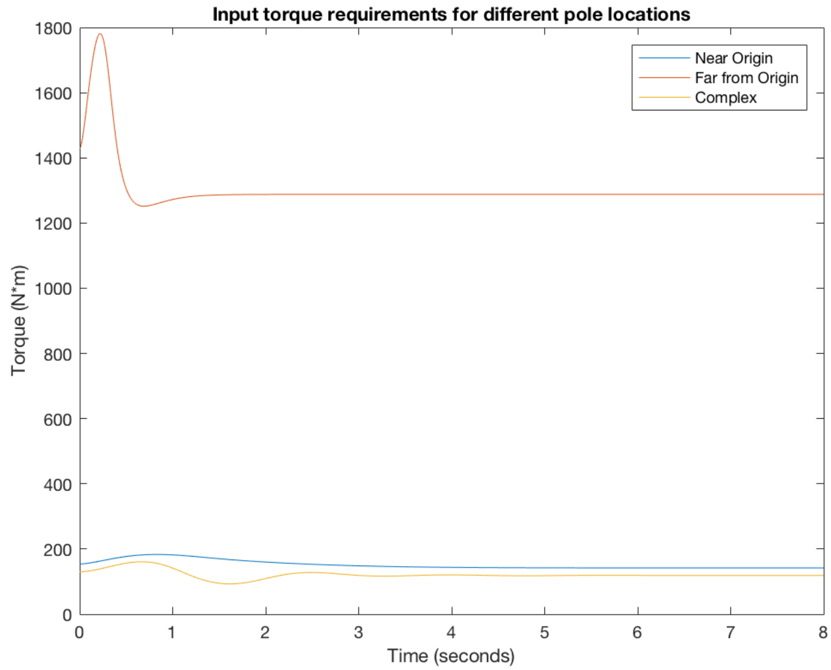



*Figure 3. Input torque requirements for different pole locations*

- **Which compensator is the best based on your own criteria? (Which compensator would you buy if you were a buyer?)**

The criteria for this system include reasonable torque values with minimal oscillation. Based on these criterion, the system with the real, all-negative poles near the origin is the best choice and the one that we would buy. The poles far from the origin require too much torque to be feasible and the complex poles allow too much oscillation.

- **As your initial angle ($\theta_1$) moves away from the linearized point, how does the performance of your best compensator change? (2 – 3 sentences)**

As Theta1 moves away from the linearized point, the performance of the compensator degrades, the system starts to behave more erratically. As you move further from the linearized point, it becomes harder for the linearized system to converge on a solution, which also impacts simulation time.

## Problem 4: Linear Quadratic Regulator (LQR) Design and Implementation, No torque on Joint 2

The Q and R matrices need to be diagonal matrices, with each diagonal element representing a "scaling" of the cost for a state. In this case, Q matrix is the 4x4 matrix shown below:

$$Q = \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The diagonal elements represent the four states, $\Theta_1$, $d\Theta_1$, $\Theta_2$, $d\Theta_2$. The larger the associated diagonal element, the more it "costs" in our cost function. So, by making the elements for $\Theta_1$ and $\Theta_2$ much larger than the element for $d\Theta_1$ and $d\Theta_2$, we are placing more value on the accuracy of our $\Theta$ outputs than the outputs of our $d\Theta$ outputs. If all the diagonal elements were the same, then the $\Theta$ outputs would be more inaccurate, possibly leading to issues in applications where positional accuracy is critical.

The Q and R matrices selected here are well-optimized because we chose to prioritize accuracy of the final position as the primary objective. Our R matrix (a 1x1 value of 0.1) specifies that the input torque $u_1$ does not "cost" as much as $\Theta$ or $d\Theta$ states do. This led to a larger input torque, but consequently also meant more accurate $\Theta$ and $d\Theta$ states as evidenced by near zero steady state error shown in Figure 4, Figure 5, and Figure 6. Each figure shows the state trajectories of the same system with different desired end positions. Note that time axis is not consistent between charts to allow detailed view of transient response.

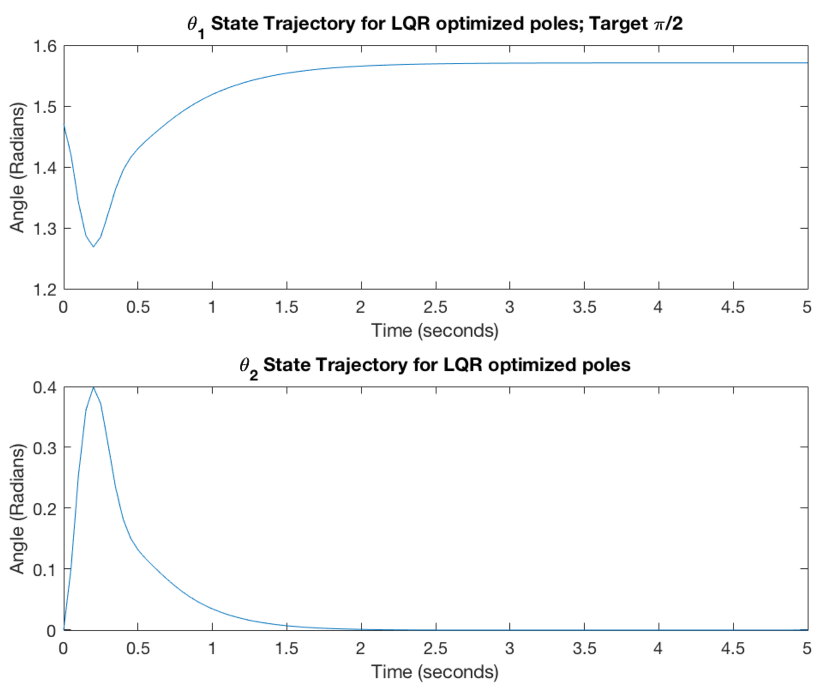



*Figure 4. State trajectories for LQR optimized poles, $\theta_1$ target - $\pi/2$*

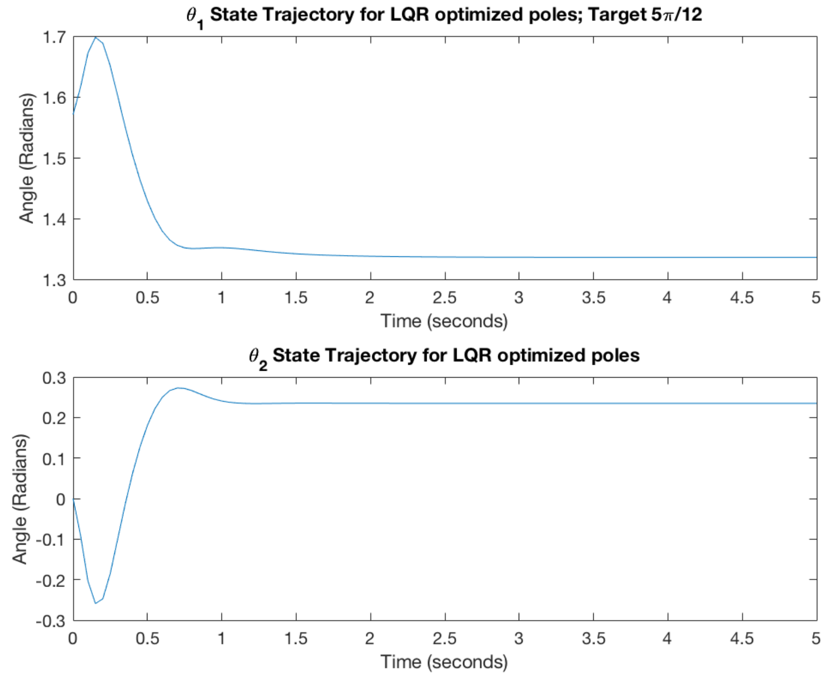



*Figure 5. State trajectories for LQR optimized poles, $\theta_1$ target - $5\pi/12$*

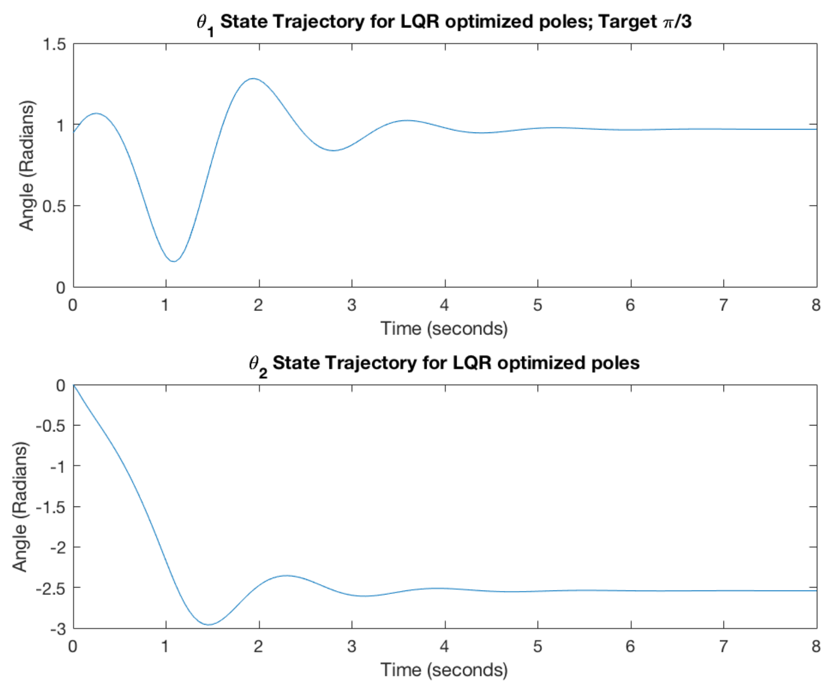



*Figure 6. State trajectories for LQR optimized poles, $\theta_1$ target - $\pi/3$*

## Problem 5: Linear Quadratic Regulator (LQR) Design and Implementation with Torque on both joints

Here, the Q and R matrices are very similar to the previous problem. They need to be diagonal matrices, with each diagonal element representing a "scaling" of the cost for a state. The difference is that now we can utilize torque in the second joint of the linkage to drive the system to a known state. In this case, Q matrix is the 4x4 matrix shown below:

$$Q = \begin{bmatrix} 50000 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 50000 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

And R is a 2x2 matrix now to incorporate the additional torque value:

$$R = \begin{bmatrix} 0.25 & 0 \\ 0 & 1 \end{bmatrix}$$

Again, in this case, we make the elements for $\Theta_1$ and $\Theta_2$ much larger than the element for $d\Theta_1$ and $d\Theta_2$, to place more value on the accuracy of our $\Theta$ outputs than the $d\Theta$ outputs in our position critical application.

The Q and R matrices selected here are well-optimized because we chose to prioritize accuracy of the final position as the primary objective (Figure 7). Our R matrix of smaller relative magnitude specifies that the input torque does not "cost" as much as $\Theta$ or $d\Theta$ states do. This led to a larger input torque, but consequently also meant more accurate $\Theta$and $d\Theta$ states. We also balance the R matrix to account for the maximum allowable relative torque values for $\Theta_1$ and $\Theta_2$ (Figure 8).

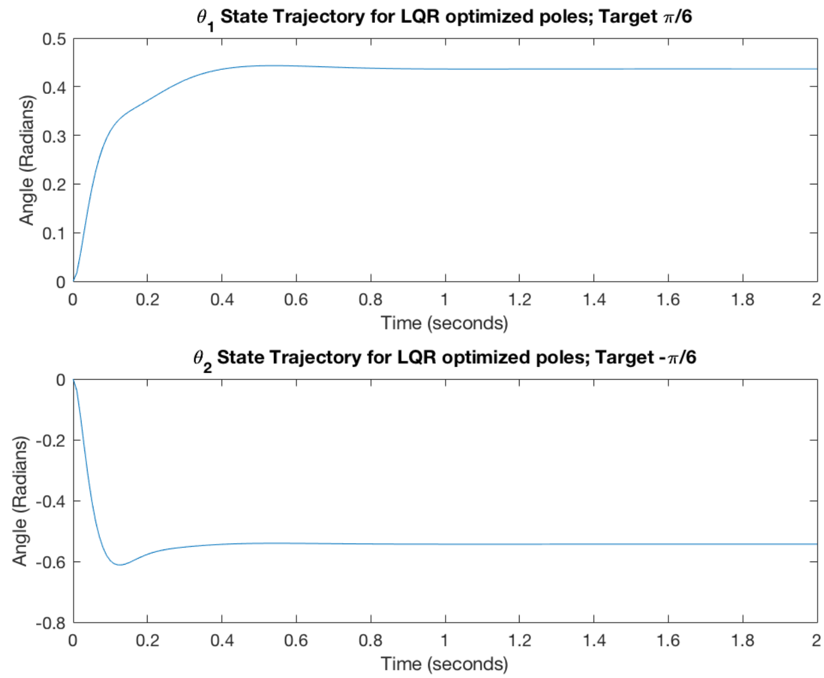



*Figure 7. State Trajectories for LQR and torque at both joints*

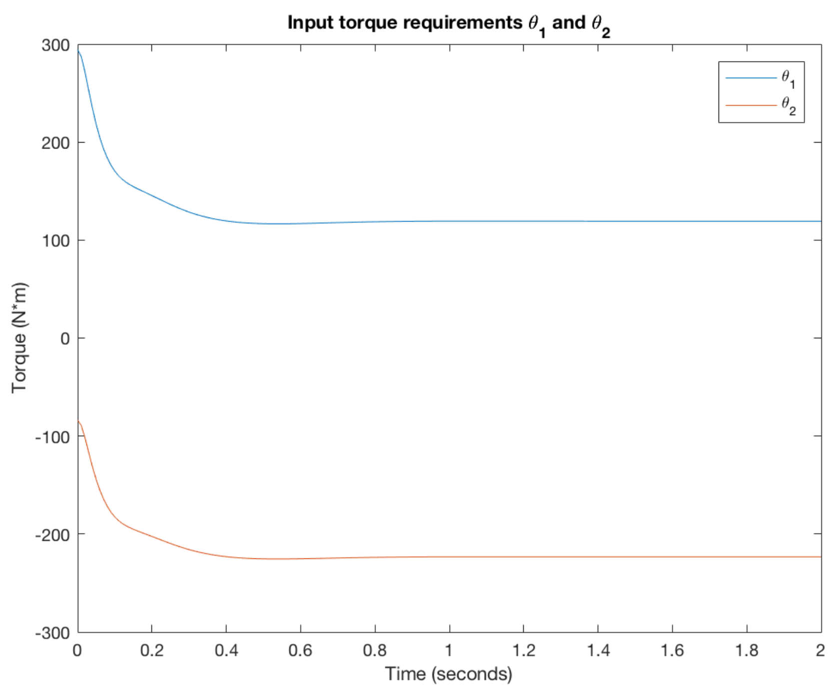



*Figure 8. Torque requirements*

# 3. Attachments

| # | Name |
|---|---|
| 1 | Problem 1 MATLAB (P1.zip) |
| 2 | Problem 2 MATLAB (P2.zip) |
| 3 | Problem 4 MATLAB (P4.zip) |
| 4 | Problem 5 MATLAB (P5.zip) |