Manish Kewalramani

Control Systems Design
14:332:417
Fall 2019

Professor Zoran Gajic

# Project 1

## Part 1: Introduction

Consider a single-link robotic manipulator with a flexible joint whose nonlinear mathematical model is given:

$$I\ddot{\theta}_1 + mgl\sin\theta_1 + k(\theta_1 - \theta_2) = 0$$

$$J\ddot{\theta}_2 - k(\theta_1 - \theta_2) = u$$

where 1; 2 are angular positions, I; J are moments of inertia, m and l are, respectively, the link's mass and length, and k is the link's spring constant. It is desired to control the robot arm to hold any reference (nominal-, operating-, trim-, steady-state- set-point) angle in the entire range [0, $2\pi$]. We will design a full-state feedback control law for this system using linearization about a set point. As a controller we will use the "pole placement" (eigenvalue assignment) controller (regulator). First we will use the Simulink state space block and simulate dynamics of the linearized system. Then we will build the Simulink block diagram of the considered nonlinear robot arm model. We will design a state space controller that keeps at steady state $\Theta_{lss} = \pi/3$.

## Part 2: Theory and Design

Using the aforementioned model, we can create a change of variables as:

$$x_1 = \theta_1,\ x_2 = \dot{\theta}_1,\ x_3 = \theta_2,\ x_4 = \dot{\theta}_2$$

Then, we can create our state space model as:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{mgl}{I}\sin x_1 - \frac{k}{I}(x_1 - x_3)$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = \frac{k}{J}(x_1 - x_3) + \frac{1}{J}u$$

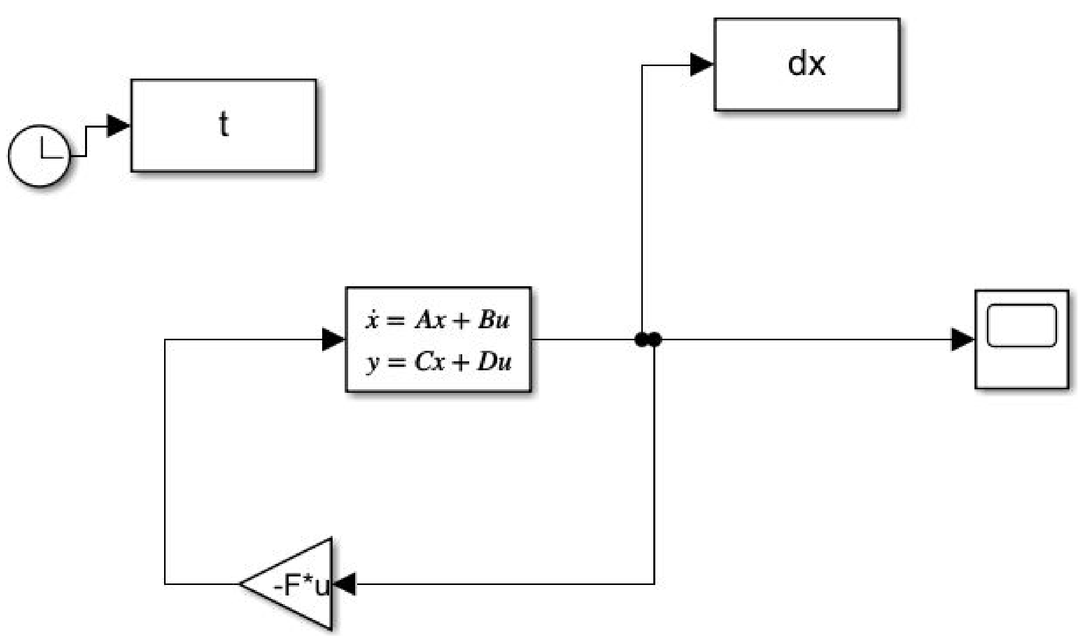
We can then find the corresponding Matrices A and B to be:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k+mgl\cos x_{1n}}{I} & 0 & \frac{k}{I} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k}{J} & 0 & -\frac{k}{J} & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J} \end{bmatrix}$$

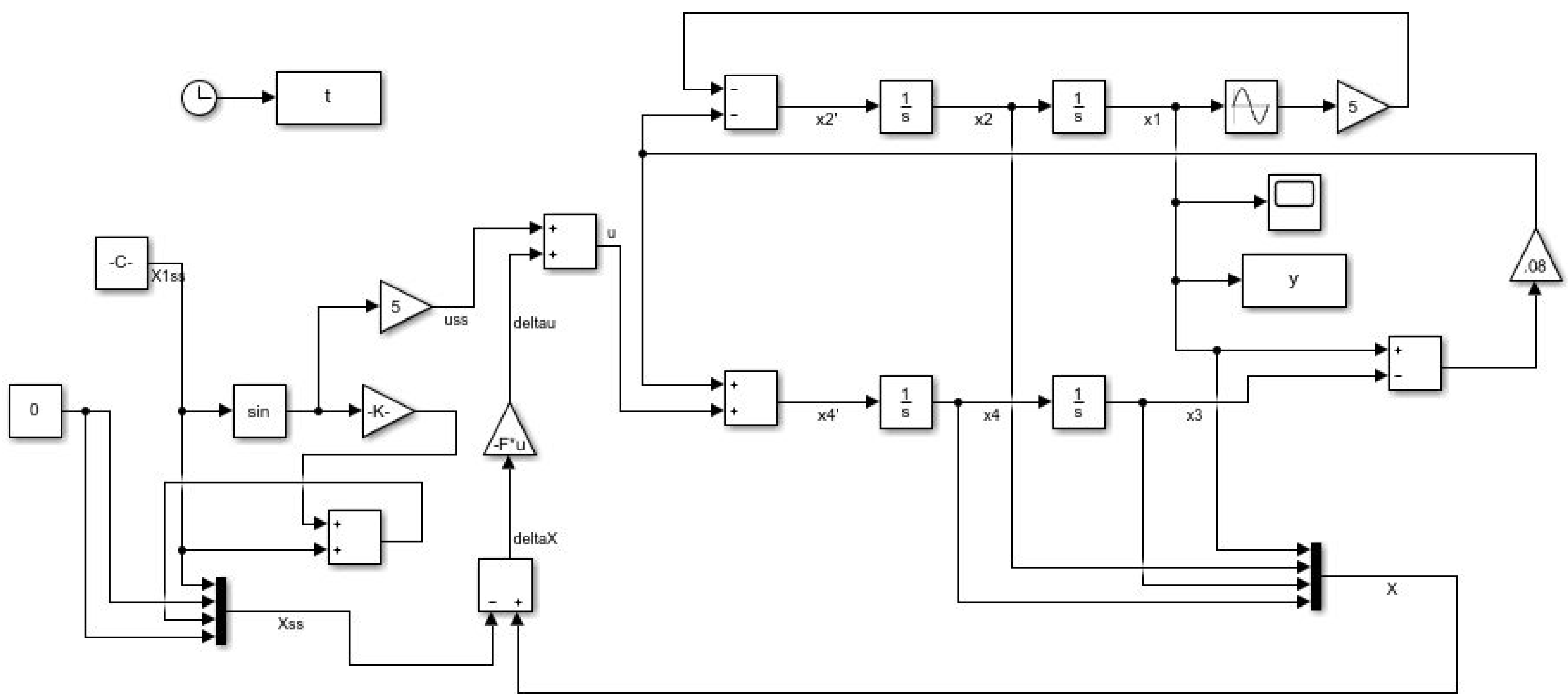
Our model uses the system parameters: mgl = 5; I = J = 1; k = 0:08. Inputting these parameters into A and B gives:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -2.58 & 0 & .08 & 0 \\ 0 & 0 & 0 & 0 \\ .08 & 0 & -.08 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

The output variable in this case, y, is equal to the first angle, so we have y = x1. In our state model the matrix C is identity and the matrix D is a zero matrix. The feedback gain F is found via the matlab function “place”, using A, B, and the desired eigenvalues as parameters. The simulink block diagram for the linearized system looks as follows:



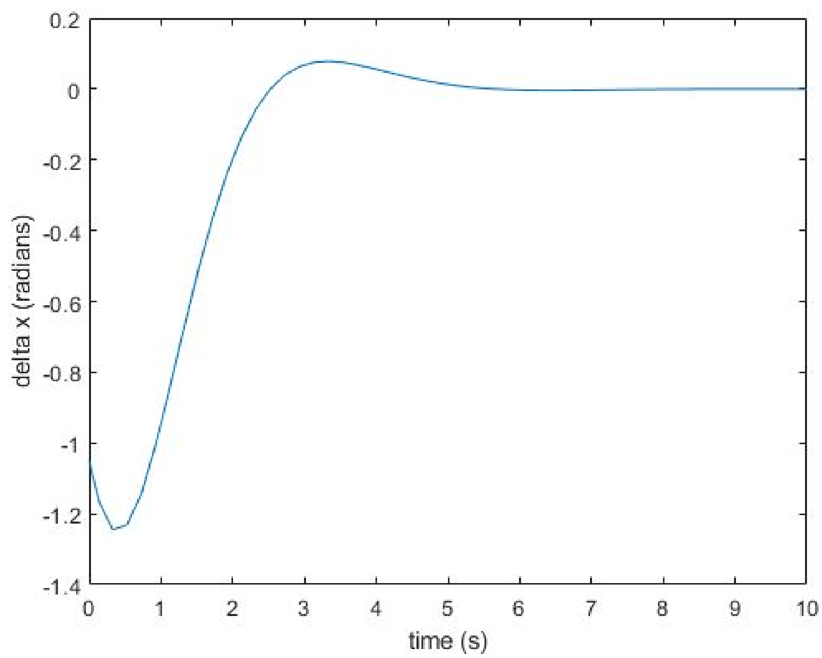

The Simulink block diagram for the considered nonlinear robot arm model looks as follows:



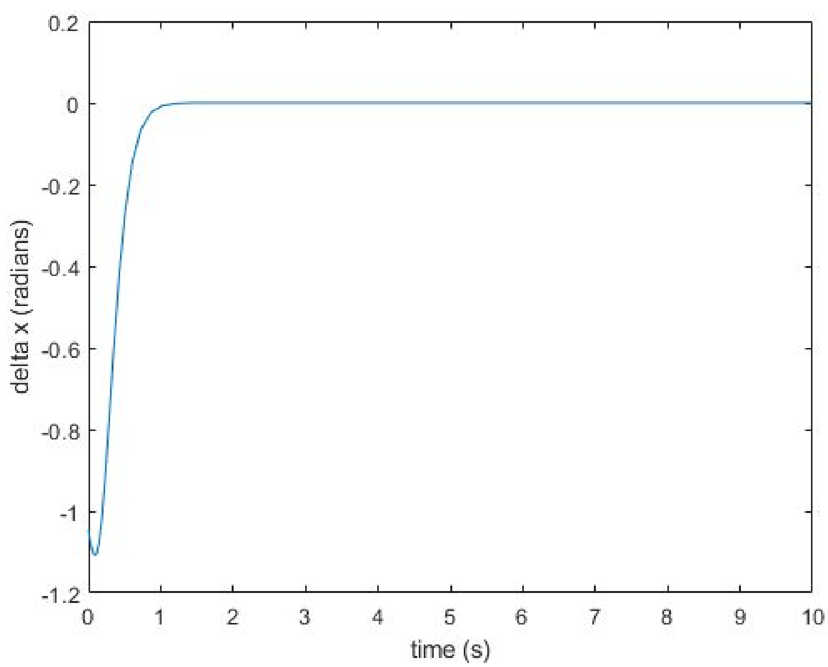
t
-C-
X1ss
0
sin
5
uss
-K-
deltau
u
-F*u
deltaX
Xss
x2'
1/s
x2
x1
5
y
.08
x4'
x4
x3
X

## Part 3: Results

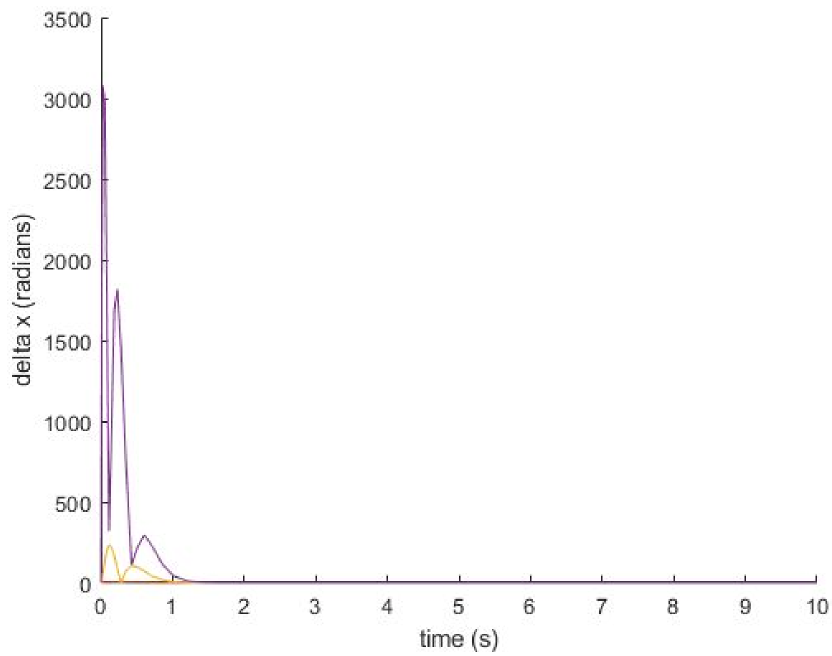
The initial condition of Δx was set to be -π/3 radians, so that the system initial conditions will be close to x = 0. The first set of eigenvalues used was -1+i, -1-i, -2+i, -2-i. The plot looked like:




We see in this case that the overshoot, the peak of the graph, is significantly above 0, at around .07 radians, the rise time is over 2 seconds and the settling time is over 5 seconds. If we increase the eigenvalues to -10+i, -10-i, -11+i, -11-i, we see the new graph looks like:

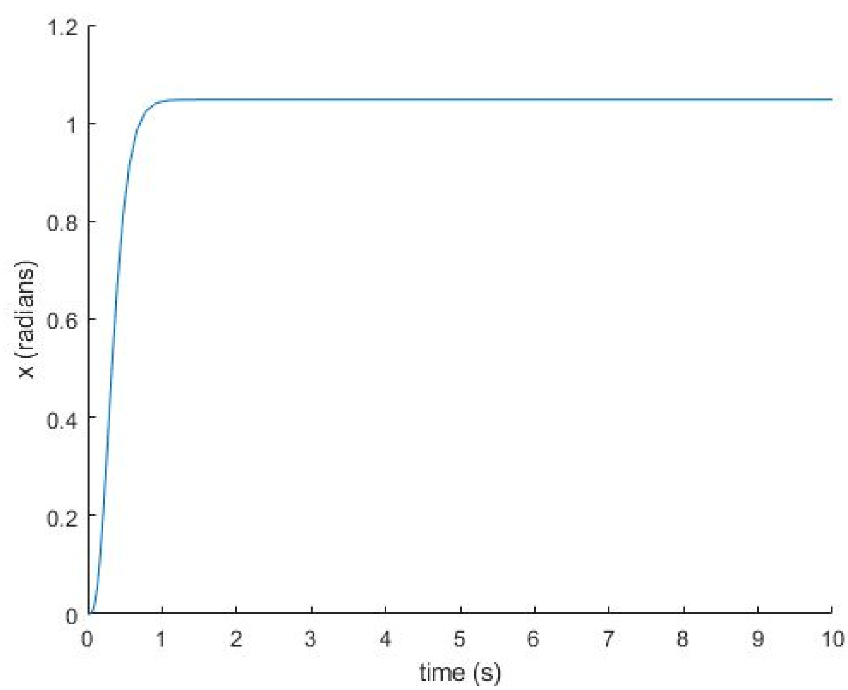


In this case, the larger eigenvalues result in almost no overshoot, a rise time of around half a second and a settling time below one second. If we look at the magnitudes of all of the signals in the closed loop system, x1 x2 x3 x4, we see the following plot:

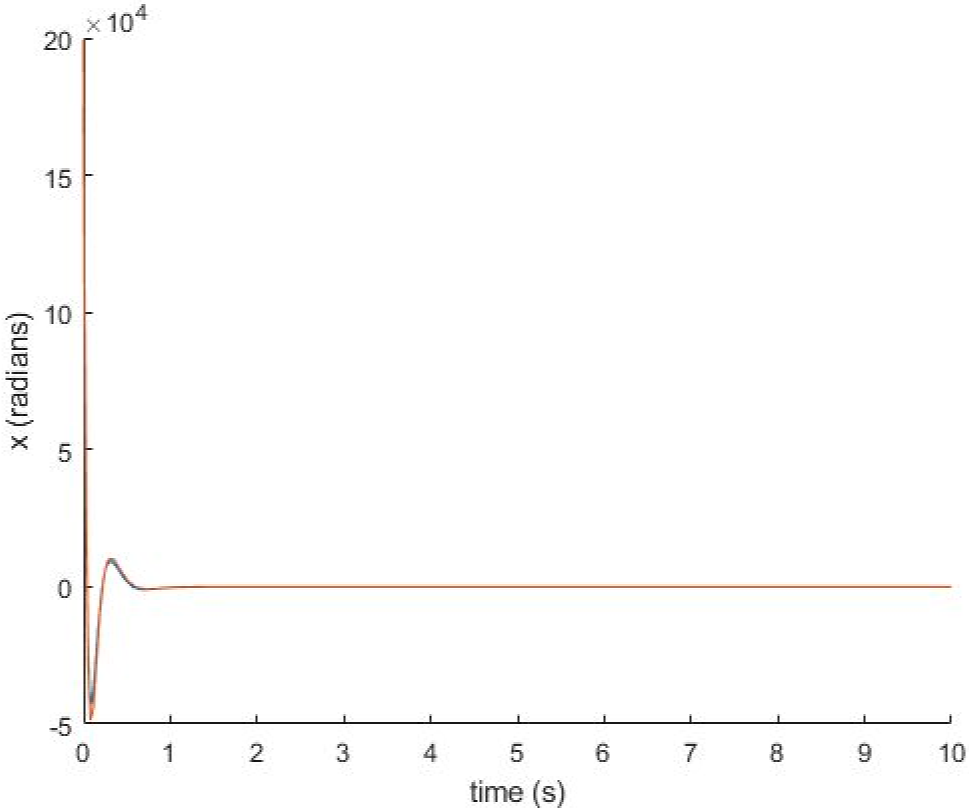



It shows that x4 > x3 > x2 > x1, with x4 being much much greater than x1.

For the nonlinear robot arm model, using the larger of the eigenvalues above, we have a plot of Θ vs t that looks like:



We see that the angle quickly approaches the steady state value of $\pi/3$ and there is very small overshoot.
The plot of the linearized input signal and the actual input signal looks like this:




With the actual input signal in blue and the linearized signal in orange. They are very close to each other and they both approach zero as the output approaches the steady state.

## Part 4: Summary

In this project we simulated a robotic arm controller in two ways: first with a linearization at steady state points and second with a nonlinear model. In both models, our system quickly approached stability, reaching the desired output steady state value of $\pi/3$. When looking at the control signal, u, in both models and comparing them we see that they are very close to each other, implying that the linearization technique allows for a very good approximation of the nonlinear model.

## Part 5: References


Elfandi, Mah. (2007). Modern control systems engineering, Z. Gajic and M. Lelic, Prentice-Hall, Europe, 1996, ISBN 0-13-134116-2. International Journal of Robust and Nonlinear Control. 17. 675 - 676. 10.1002/rnc.1081. Chapter 1, page 29

## Part 6: Appendix

The code for the matlab file is:

```
x1ss=pi/3;
uss=5*pi/3;
A=[0 1 0 0;
    -2.58 0 .08 0;
    0 0 0 1;
    .08 0 -.08 0];
B=[0;
    0;
    0;
    1];
lamda_desired=[ -10+i -10-i -11+i -11-i ];
F=place(A,B,lamda_desired);

hold on
%plot(out.t,out.du);
%plot(out.t,out.dulin);
plot(out.t,abs(out.dx(:,1)));
plot(out.t,abs(out.dx(:,2)));
plot(out.t,abs(out.dx(:,3)));
plot(out.t,abs(out.dx(:,4)));
xlabel('time (s)')
ylabel('x (radians)')
out.u
```