24352: Homework #4

Daniel Deng

March 5, 2021

Question 1

State space representation from differential equations

a)

$2\ddot{x} + 4\dot{x} + 4x = 3u$, where the state vector is $\begin{bmatrix} x \\ \dot{x} \end{bmatrix}$. The out put is x.

solution

$\ddot{x} + 2\dot{x} + 2x = 1.5u \Rightarrow \begin{cases} \dot{x} = 0x + \dot{x} \\ \ddot{x} = -2x - 2\dot{x} + 1.5u \end{cases}$, thus

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -2 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1.5 \end{bmatrix} u \tag{1}$$

$$\dot{\mathbf{y}} = \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} u \tag{2}$$

We get:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -2 & -2 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \mathbf{D} = \begin{bmatrix} 0 \end{bmatrix} \tag{3}$$

b)

$2\ddot{x} + 4\dot{x} + 4x = 3u$, where the state vector is $\begin{bmatrix} x + \dot{x} \\ \dot{x} \end{bmatrix}$. The out put is x.

solution

$\begin{cases} \dot{x} + \ddot{x} = \dot{x} + (-2x - 2\dot{x} + 1.5u) = -2(x + \dot{x}) + \dot{x} + 1.5u \\ \ddot{x} = -2x - 2\dot{x} + 1.5u = -2(x + \dot{x}) + 0\dot{x} + 1.5u \\ x = (x + \dot{x}) + (-1)\dot{x} \end{cases}$, thus

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} + \ddot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ -2 & 0 \end{bmatrix} \begin{bmatrix} x + \dot{x} \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix} u \tag{4}$$

$$\dot{\mathbf{y}} = \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} x + \dot{x} \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} u \tag{5}$$

We get:

$$\mathbf{A} = \begin{bmatrix} -2 & 1 \\ -2 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & -1 \end{bmatrix}, \mathbf{D} = \begin{bmatrix} 0 \end{bmatrix} \tag{6}$$

c)

Using the A,B,C,D matrices found in parts (a) and (b), use Matlab's step command to simulate the step responses to both systems. How do these step responses compare and does that comparison make sense?

solution

Matlab code:

```
t = 0:0.01:10;

A = [0, 1; -2 -2];
B = [0; 1.5];
C = [1 0];
D = [0];
sys1 = ss(A,B,C,D);

A = [-2, 1; -2 0];
B = [1.5; 1.5];
C = [1 -1];
D = [0];
sys2 = ss(A,B,C,D);

[y1,t] = step(sys1,t);
[y2,t] = step(sys2,t);
plot(t,y1,'r',t,y2,'b--','linewidth',2)
grid minor
title('Response to step')
xlabel('Time (s)')
ylabel('Amplitude')
legend('y1', 'y2')
```

Result:
We can tell from Fig.1 that the step responses are exactly the same. This make sense because we are representing the same system, and the output is the same for the two representations.

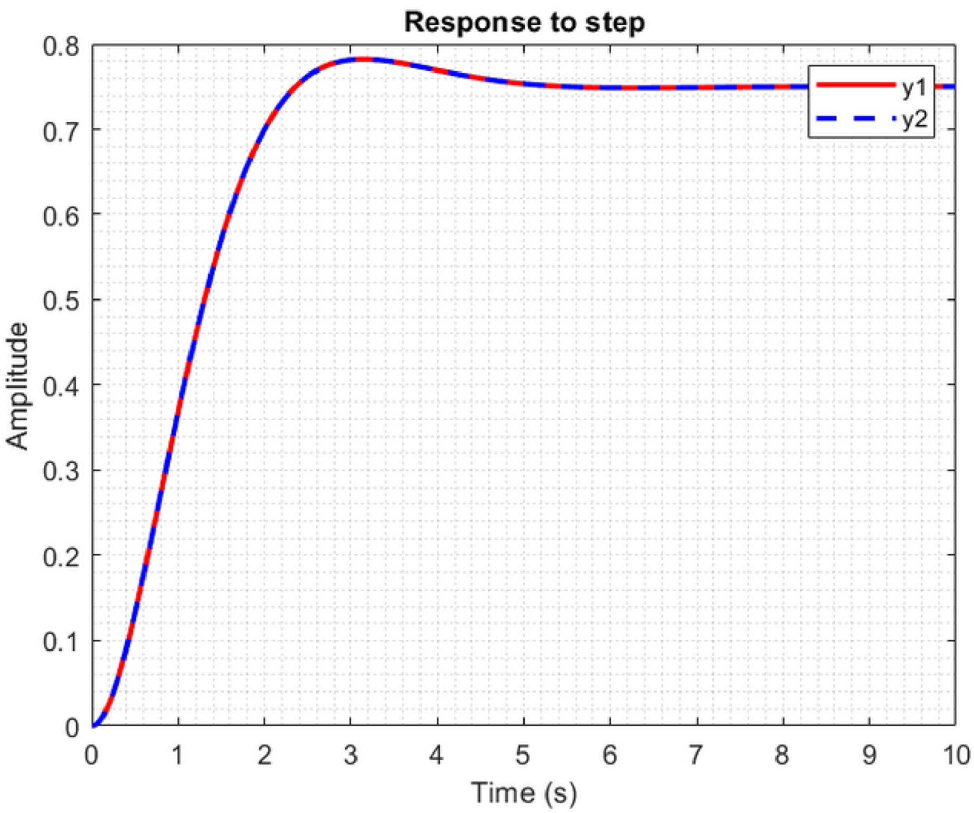


Figure 1: Step Response

Problem 2

Let $\Sigma = \{0, 1\}$. Construct a DFA A that recognizes the language that consists of all binary numbers that can be divided by 5.

Let the state q_k indicate the remainder of k divided by 5. For example, the remainder of 2 would correlate to state q_2 because $7 \mod 5 = 2$.

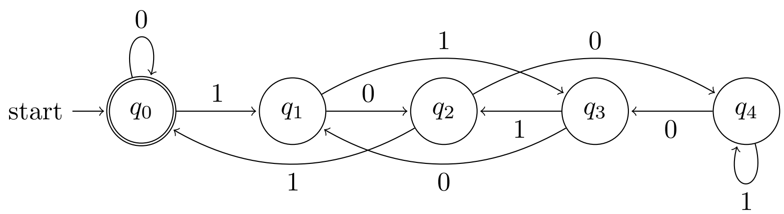


Figure 2: DFA, A, this is really beautiful, ya know?

Justification

Take a given binary number, x. Since there are only two inputs to our state machine, x can either become $x0$ or $x1$. When a 0 comes into the state machine, it is the same as taking the binary number and multiplying it by two. When a 1 comes into the machine, it is the same as multiplying by two and adding one.
Using this knowledge, we can construct a transition table that tell us where to go:

	$x \mod 5 = 0$	$x \mod 5 = 1$	$x \mod 5 = 2$	$x \mod 5 = 3$	$x \mod 5 = 4$
$x0$	0	2	4	1	3
$x1$	1	3	0	2	4

Therefore on state q_0 or ($x \mod 5 = 0$), a transition line should go to state q_0 for the input 0 and a line should go to state q_1 for input 1. Continuing this gives us the Figure 2.

Problem 3

Write part of **Quick-Sort**($list, start, end$)

```
function Quick-Sort(list, start, end)
    if start ≥ end then
        return
    end if
    mid ← Partition(list, start, end)
    Quick-Sort(list, start, mid − 1)
    Quick-Sort(list, mid + 1, end)
end function
```

Algorithm 1: Start of QuickSort

Problem 4

question

solution

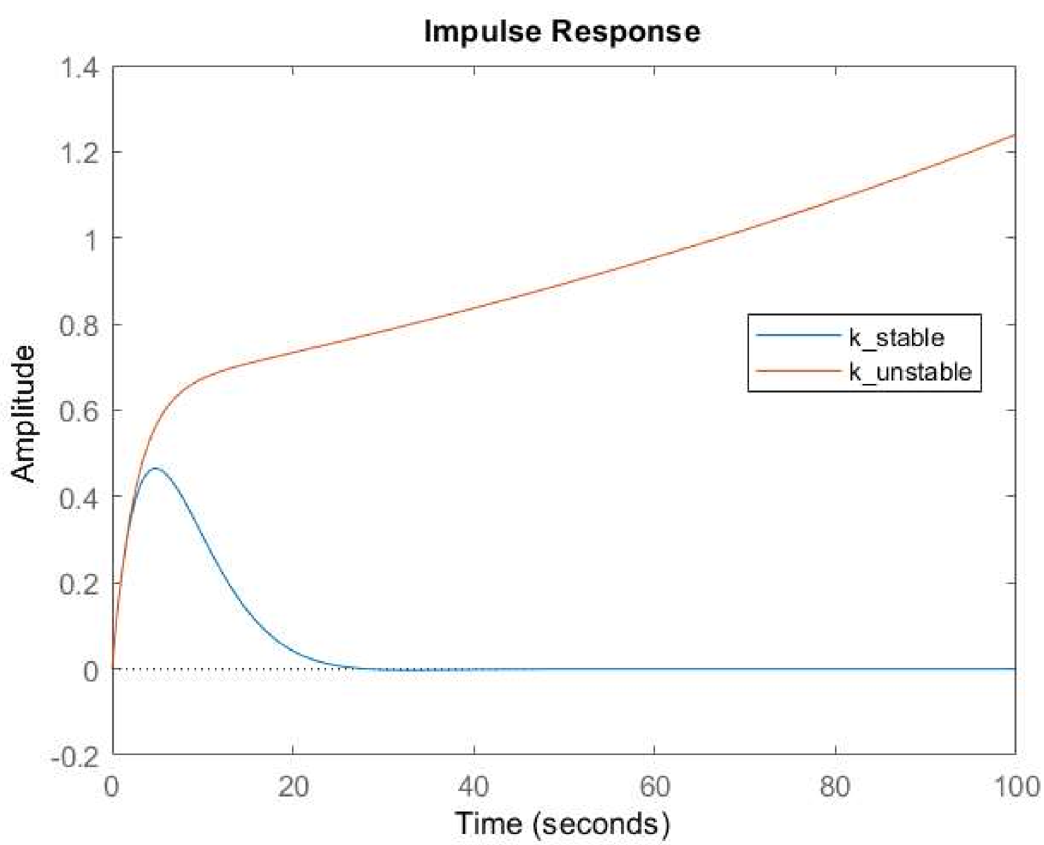
partA

Homework 3

Question 1

```
clc,clear;

k_stable = 10; % N * m^-1
k_unstable = 9.8;
g = 9.81;
m = 1;   % kg
c = 1.5; % N * s * m^-1
l = 2; % m
l1 = 1; % m
t = 0:0.1:100; % 10 seconds of time
figure;
hold on
for k = [k_stable k_unstable]
    G = tf(1, [m * l^2 c * l1^2 (k*l1^2 - m*g)]);
    impulse(G, t)
end
legend("k_{stable}", "k_{unstable}", 'Location', "best")
```



Question 2