**ME 5554 / AOE5754 / ECE5754**

**Applied Linear Systems**

**HW2**

Luan Cong Doan

[luandoan@vt.edu](mailto:luandoan@vt.edu)

**1. Place the following nonlinear ODE into first-order form:**

Define:

So:

The resulting nonlinear state equation plus output equation then are:

Assume that at initial condition t = to:

We have:

with:

.

⇔

⇔

It follows directly that deviation variables are specified by:

With:

Partial differentiation yields:

Evaluating at the nominal trajectory gives:

**2. Given the following State Equation and initial condition:**

with

**2a.** Use the Matlab function EXPM() to compute a numerical representation for the Matrix Exponential.

A = [0 -5; 1 -2]; % state space matrix

t = linspace(0,5,10); % time vector

phi = zeros(2,2,10); % create temperate matrix

for i = 1:10

phi(:,:,i) = expm(A\*t(i));

end

Result:

phi(:,:,1) = 1 0

0 1

phi(:,:,2) = 0.5117 -1.2855

0.2571 -0.0025

phi(:,:,3) = -0.0687 -0.6545

0.1309 -0.3305

phi(:,:,4) = -0.2034 0.0900

-0.0180 -0.1674

phi(:,:,5) = -0.0809 0.2613

-0.0523 0.0236

phi(:,:,6) = 0.0258 0.1034

-0.0207 0.0671

phi(:,:,7) = 0.0398 -0.0334

0.0067 0.0264

phi(:,:,8) = 0.0118 -0.0510

0.0102 -0.0086

phi(:,:,9) = -0.0071 -0.0150

0.0030 -0.0131

phi(:,:,10) = -0.0075 0.0092

-0.0018 -0.0038

**2b.** Use the Symbolic Toolbox with the EXPM() function in Matlab to generate a symbolic representation for the Matrix Exponential:

syms a b c d t;

A = [a b;c d];

PhiB = expm(A\*t);

**2c.** Numerically evaluate the symbolic representation from 2b using the same time vector from 2a:

A(1,1) = 0; A(1,2) = -5;

A(2,1) = 1; A(2,2) = -2; % numerical elements in A

PB = expm(A\*t); % re-calculate

T = linspace(0,5,10); % time vector

p11 = PB(1,1);

p12 = PB(1,2);

p21 = PB(2,1);

p22 = PB(2,2);

for i = 1:10

PB1(i) = double(subs(p11,{t},T(i)));

PB2(i) = double(subs(p12,{t},T(i)));

PB3(i) = double(subs(p21,{t},T(i)));

PB4(i) = double(subs(p22,{t},T(i)));

end

Result:

>> PB11

PB11 =

1.0000 0.5117 -0.0687 -0.2034 -0.0809 0.0258 0.0398 0.0118 -0.0071 -0.0075

>> PB12

PB12 =

0 -1.2855 -0.6545 0.0900 0.2613 0.1034 -0.0334 -0.0510 -0.0150 0.0092

>> PB21

PB21 =

0 0.2571 0.1309 -0.0180 -0.0523 -0.0207 0.0067 0.0102 0.0030 -0.0018

>> PB22

PB22 =

1.0000 -0.0025 -0.3305 -0.1674 0.0236 0.0671 0.0264 -0.0086 -0.0131 -0.0038

**2d.** Generate an analytic representation for the Matrix Exponential using Laplace Transforms:

We have:

Taking Laplace transform for an analytic representation:

So:

Final result:

Test with Matlab: A = [0 -5; 1 -2];

syms s;

R = s\*eye(2) - A;

RI = inv(R);

PD = ilaplace(RI);

Result:

>> PD

PD =

[ exp(-t)\*(cos(2\*t) + sin(2\*t)/2), -(5\*sin(2\*t)\*exp(-t))/2]

[ (sin(2\*t)\*exp(-t))/2, exp(-t)\*(cos(2\*t) - sin(2\*t)/2)]

**2e.** Numerically evaluate the symbolic representation from 2d using the same time vector from 2a:

T = linspace(0,5,10); % time vector

for i = 1:10

PD11(i) = double(subs(PD(1,1),{t},T(i)));

PD12(i) = double(subs(PD(1,2),{t},T(i)));

PD21(i) = double(subs(PD(2,1),{t},T(i)));

PD22(i) = double(subs(PD(2,2),{t},T(i)));

end

Results:

>> PD11

PD11 =

1.0000 0.5117 -0.0687 -0.2034 -0.0809 0.0258 0.0398 0.0118 -0.0071 -0.0075

>> PD12

PD12 =

0 -1.2855 -0.6545 0.0900 0.2613 0.1034 -0.0334 -0.0510 -0.0150 0.0092

>> PD21

PD21 =

0 0.2571 0.1309 -0.0180 -0.0523 -0.0207 0.0067 0.0102 0.0030 -0.0018

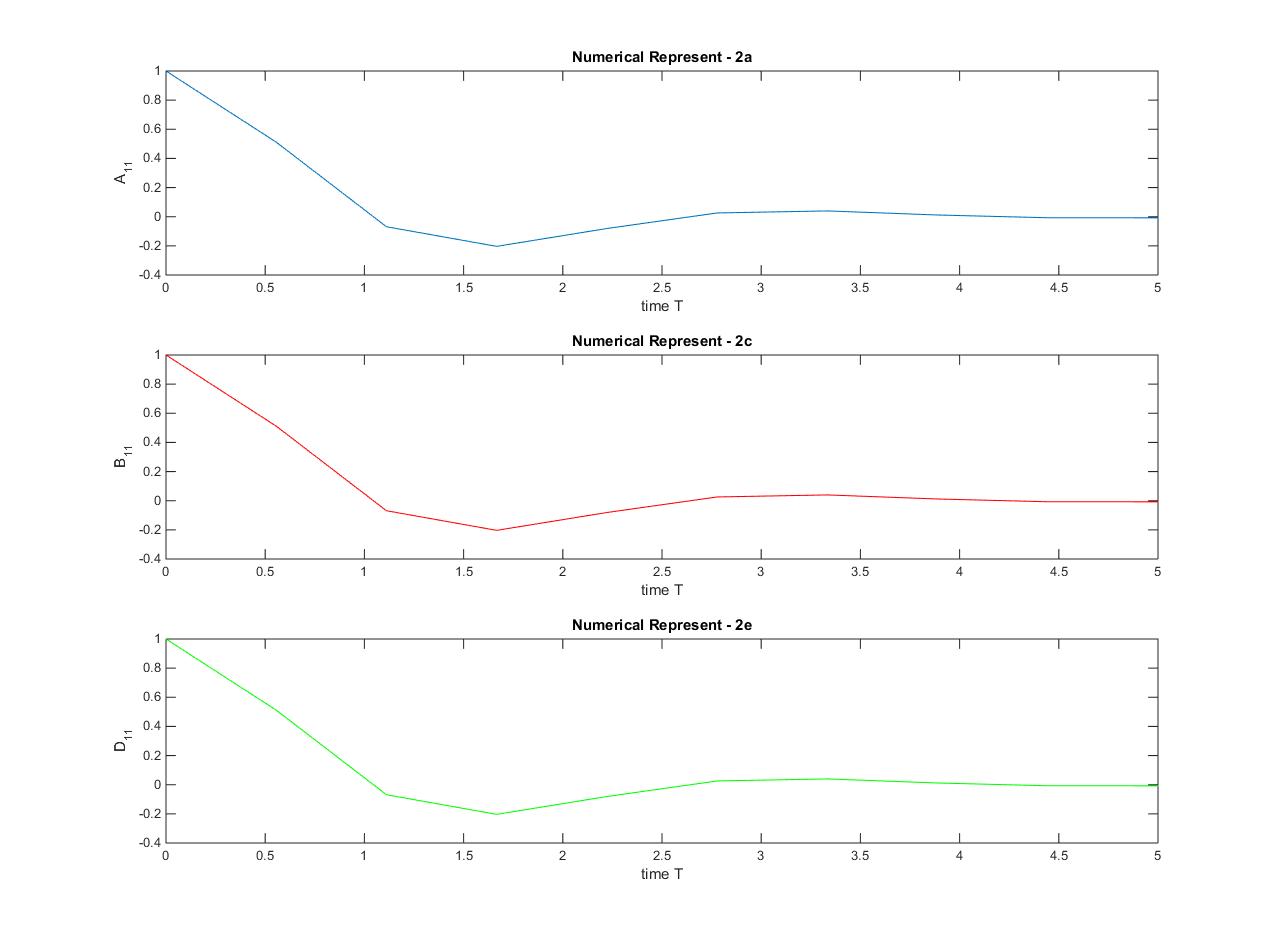
>> PD22

PD22 =

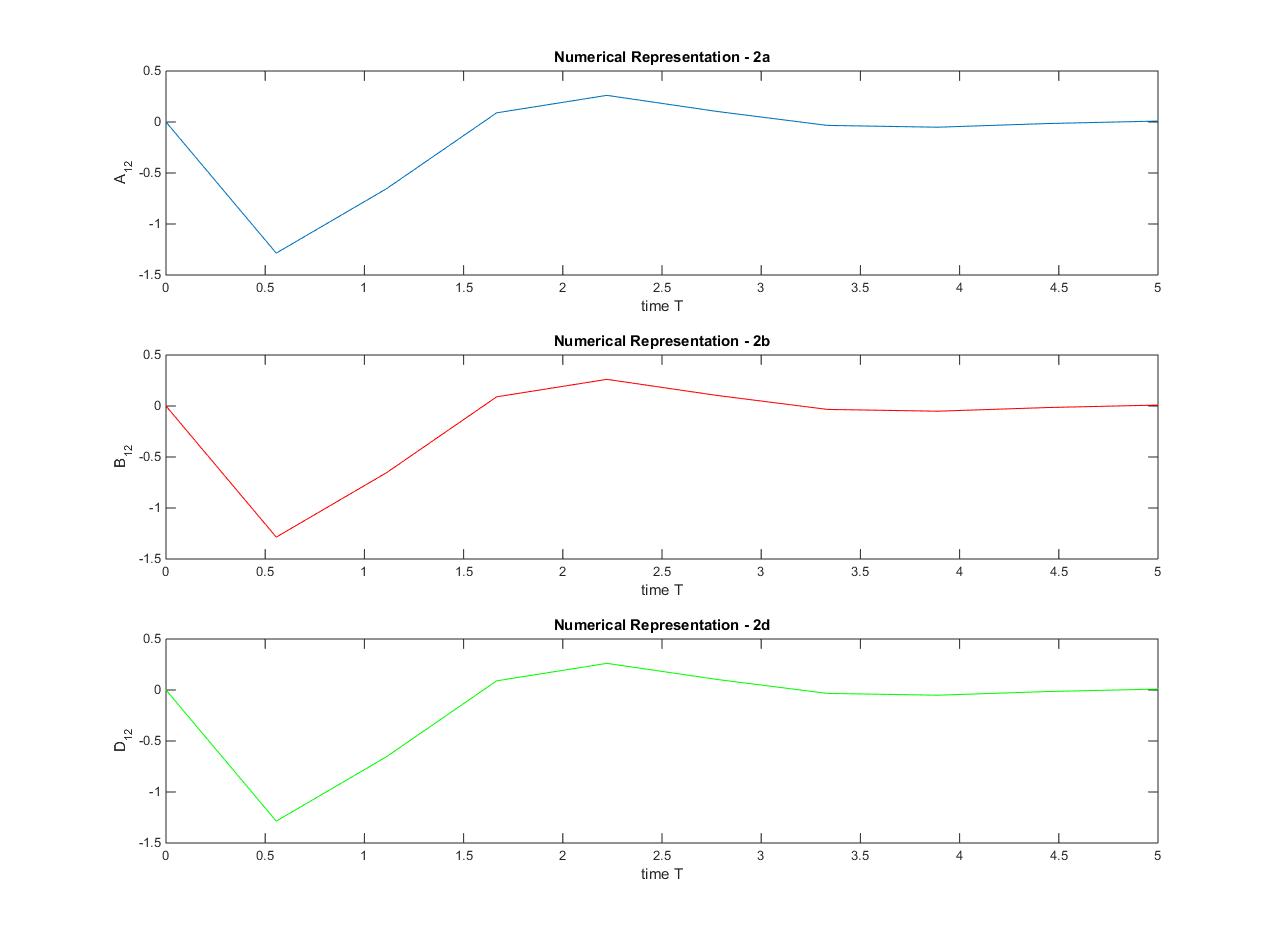
1.0000 -0.0025 -0.3305 -0.1674 0.0236 0.0671 0.0264 -0.0086 -0.0131 -0.0038

**2f.** Use Matlab to generate a plot of all elements in the Matrix Exponential:

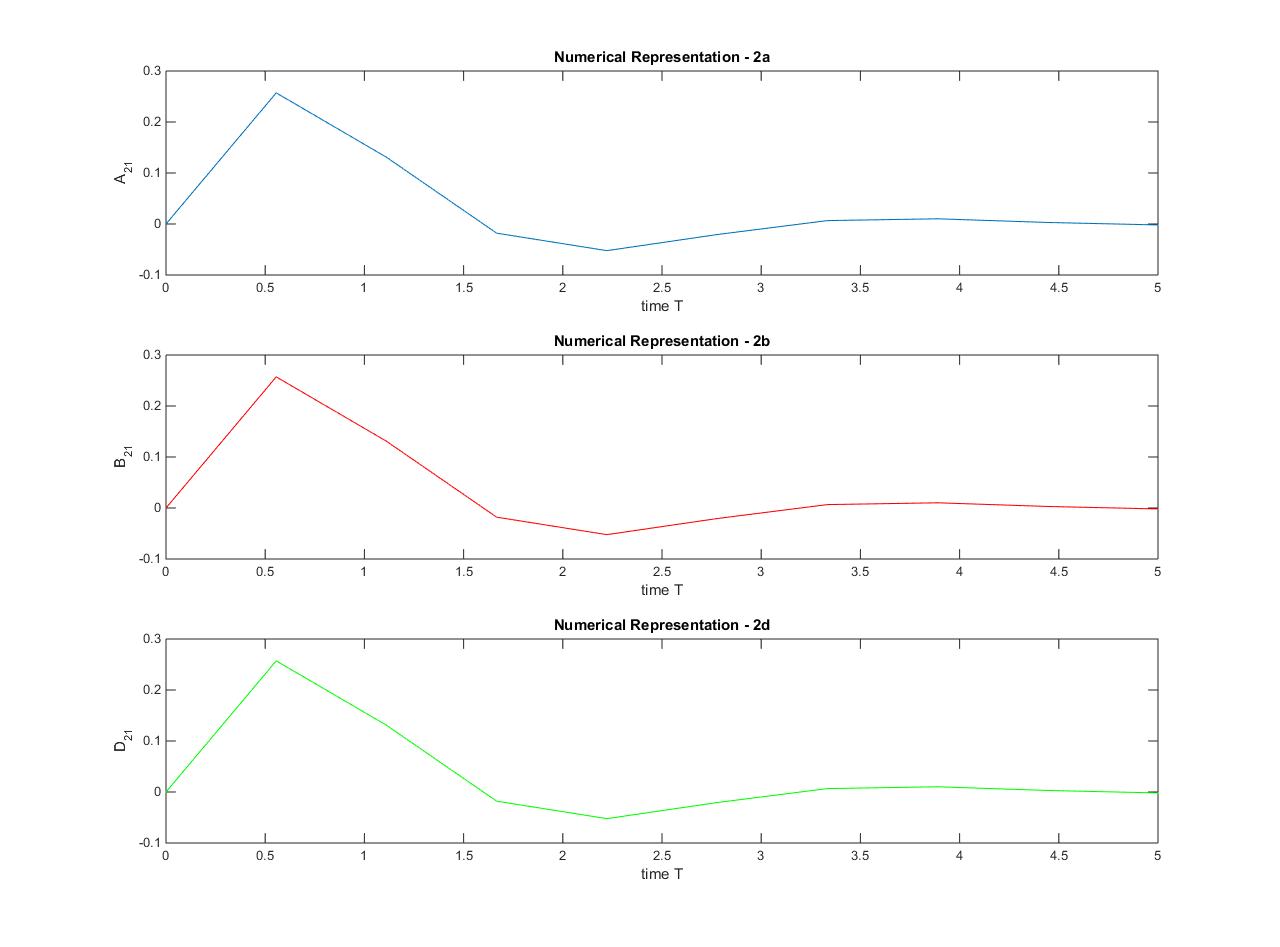
First element: first row + first column



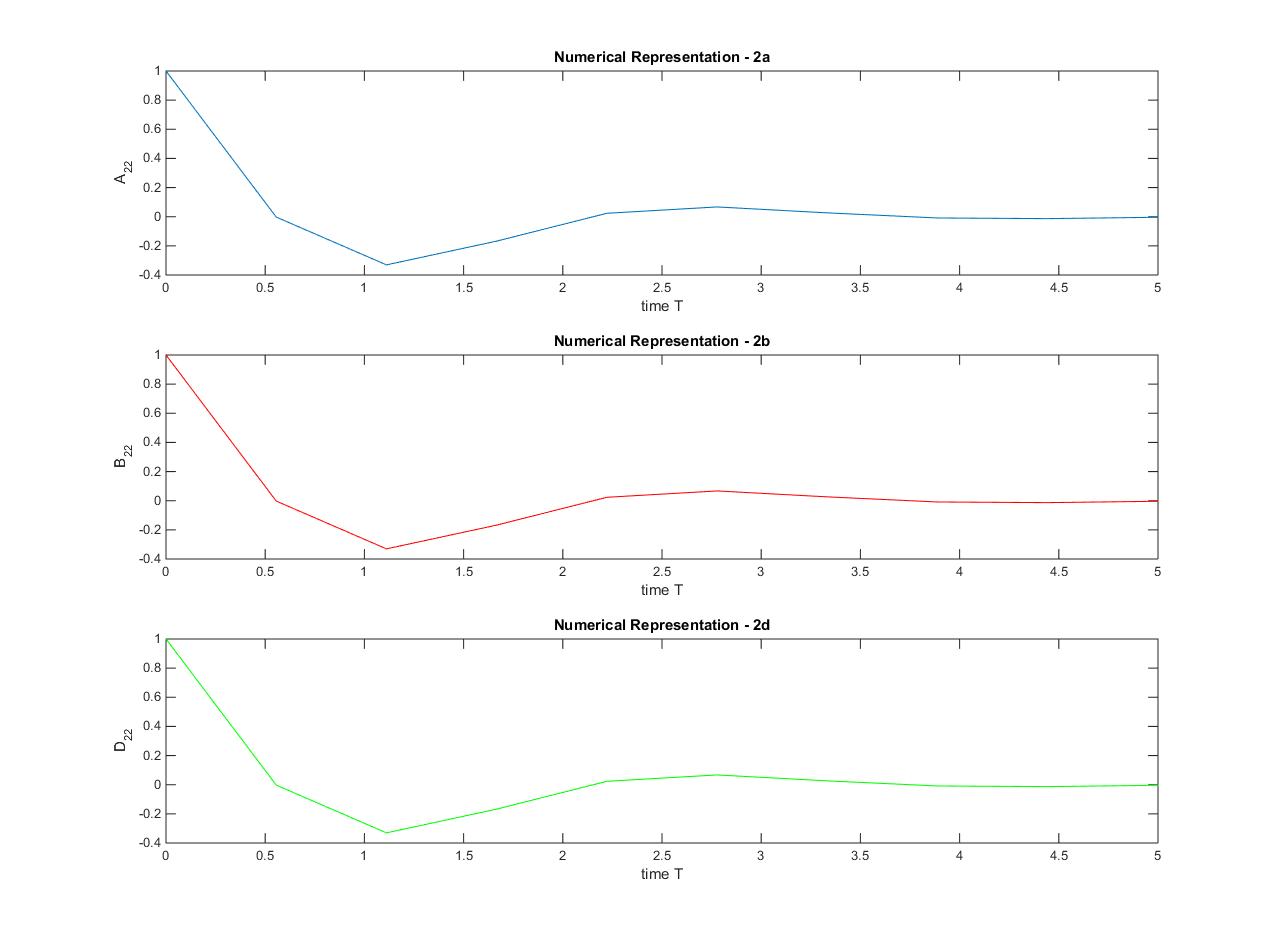
Second element: first row + second column:



Third element: second row + first column:



Forth element: second row + second column:



**3. Use any of the result from problem 2 to compute and plot the state trajectory for the Initial Value response of this dynamic system:**

- Use the result from problem 2b with assume that input u(t) = 0. We have:

A = [0 -5;1 -2];

syms t;

x0 = [1; -1];

PB = expm(A\*t); % expm(A\*t)

T = linspace(0,5,10); % time vector

x\_t = PB\*x0;

for i = 1:10

x\_t11(i) = double(subs(x\_t(1,1),{t},T(i)));

x\_t21(i) = double(subs(x\_t(2,1),{t},T(i)));

end

plot(T,x\_t11,T,x\_t21,'r');

xlabel('time T');

ylabel('X');

title('State Trajectory');

legend('x\_1','x\_2');

Result:

