ASEN 5044 HW 3 Script

Table of Contents

By: Ian Faber

Housekeeping

```
clc; clear; close all;
```

Problem 1a

Setup

```
dt = 0.05; % sec

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

B = [
        0  0
        -1 0
        0  0
        1  1
    ];

C = [
        1 0 0 0
        0 1 0 -1
    ];

D = zeros(2,2);

Ahat = [
            A B
            zeros(2,6)
       ];

% Calculate DT system
```

```
matExp = expm(Ahat*dt);

F = matExp(1:4, 1:4)
G = matExp(1:4, 5:6)
H = C
M = D
```

```
F =

    0.9975    0.0500    0.0012    0.0000
   -0.0999    0.9975    0.0499    0.0012
    0.0012    0.0000    0.9975    0.0500
    0.0499    0.0012   -0.0999    0.9975


G =

   -0.0012    0.0000
   -0.0499    0.0000
    0.0012    0.0012
    0.0499    0.0500


H =

     1     0     0     0
     0     1     0    -1


M =

     0     0
     0     0
```

Problem 1b

Compute observability matrix

```
O = [
        H
        H*F
        H*F^2
        H*F^3
    ]

obsRank = rank(O)
```

```
O =

    1.0000         0         0         0
         0    1.0000         0   -1.0000
```

```
    0.9975    0.0500    0.0012    0.0000
   -0.1498    0.9963    0.1498   -0.9963
    0.9900    0.0997    0.0050    0.0002
   -0.2985    0.9850    0.2985   -0.9850
    0.9776    0.1489    0.0112    0.0006
   -0.4450    0.9664    0.4450   -0.9664


obsRank =

     4
```

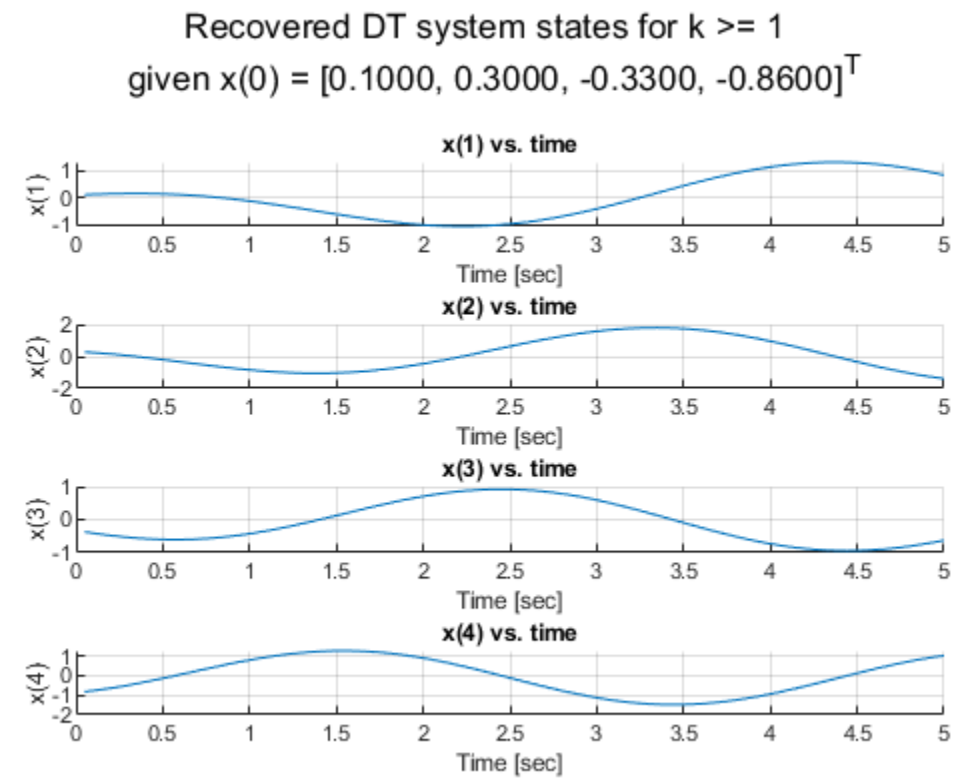
Problem 1d

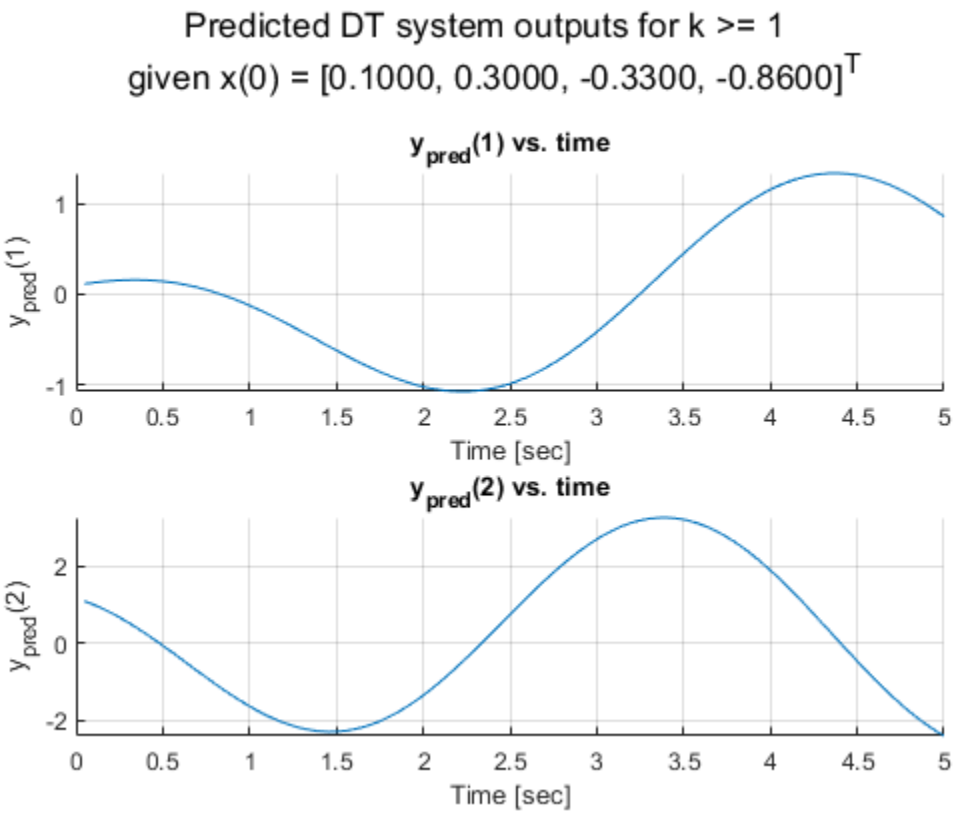
Setup

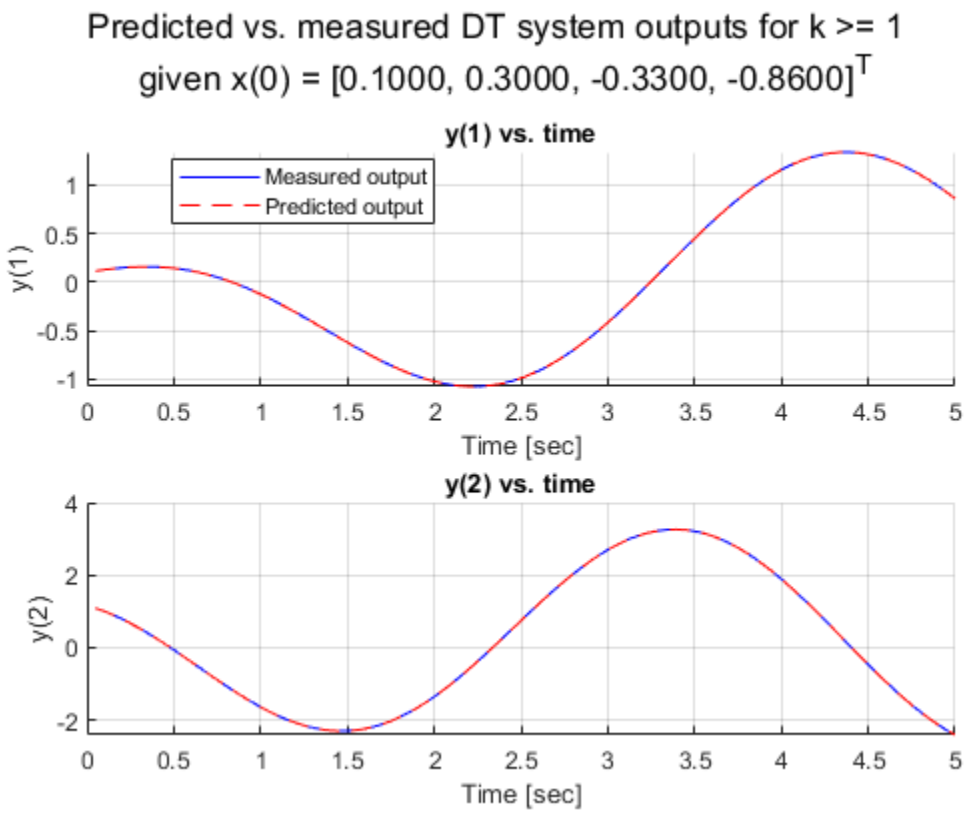
```
load("hw3problem1data.mat") % Ydata from k=1, Udata from k=0
t = dt:dt:5; % Start at k = 1

% Format and create vectors as needed
Y = [];
U = [];
E = [];
bigG = [];
bigM = [];
meas = 1:size(Ydata,1); % Which samples to analyze for x(0)
for k = meas
    U = [U; Udata(k+1,1); Udata(k+1,2)]; % Udata is from k = 0, don't want it
    Y = [Y; Ydata(k,1); Ydata(k,2)];
    E = [E; H*(F^k)];

    block = []; % Reset helper variable for building bigG
    for kk = k:-1:1
        if kk == 1
            block = [block, H*G];
        else
            block = [block, H*(F^(kk-1))*G];
        end
    end
    if kk < size(Ydata, 1) % Fill out the rest of this block of bigG with 0's
        block = [block, zeros(size(block,1), length(meas)*size(Udata,2) -
size(block, 2))]; % Final block should be k*m columns
    end
    bigG = [bigG; block];

    bigM = blkdiag(bigM, M);
end

U1 = [Udata(1,:)'; U(1:end-size(Udata,2))];
U2 = U;

% Calculate x(0)
```

```
x0 = ((E'*E)^-1)*E'*(Y-bigG*U1-bigM*U2)

% Define input function
u = @(t) [sin(t); 0.1*cos(t)];

% Get states and predicted state output
x = [];
yPred = [];
xLast = x0;
for kk = t % t starts at k = 1!
    xNew = F*xLast + G*u(kk-dt); % x(k+1) = Fx(k) + Gu(k)
    yNew = H*xNew + M*u(kk); % y(k+1) = Hx(k+1) + Gu(k+1)
    x = [x, xNew];
    yPred = [yPred, yNew];
    xLast = xNew;
end

% Plot recovered states
titleText = sprintf("Recovered DT system states for k >= 1 \n given x(0) =
[%.4f, %.4f, %.4f, %.4f]^T", x0);
figure(1)
ax = zeros(4,1);
sgtitle(titleText)
ax(1) = subplot(4,1,1);
    hold on; grid on;
    title("x(1) vs. time")
    plot(t, x(1,:))
    xlabel("Time [sec]")
    ylabel("x(1)")
ax(2) = subplot(4,1,2);
    hold on; grid on;
    title("x(2) vs. time")
    plot(t, x(2,:))
    xlabel("Time [sec]")
    ylabel("x(2)")
ax(3) = subplot(4,1,3);
    hold on; grid on;
    title("x(3) vs. time")
    plot(t, x(3,:))
    xlabel("Time [sec]")
    ylabel("x(3)")
ax(4) = subplot(4,1,4);
    hold on; grid on;
    title("x(4) vs. time")
    plot(t, x(4,:))
    xlabel("Time [sec]")
    ylabel("x(4)")

% Plot predicted outputs
titleText = sprintf("Predicted DT system outputs for k >= 1 \n given x(0) =
[%.4f, %.4f, %.4f, %.4f]^T", x0);
figure(2)
ax = zeros(4,1);
sgtitle(titleText)
```

```
ax(1) = subplot(2,1,1);
    hold on; grid on;
    title("y_{pred}(1) vs. time")
    plot(t, yPred(1,:))
    xlabel("Time [sec]")
    ylabel("y_{pred}(1)")
ax(2) = subplot(2,1,2);
    hold on; grid on;
    title("y_{pred}(2) vs. time")
    plot(t, yPred(2,:))
    xlabel("Time [sec]")
    ylabel("y_{pred}(2)")

% Plot comparison of measured and predicted outputs
Ydata = Ydata';
titleText = sprintf("Predicted vs. measured DT system outputs for k >= 1 \n
given x(0) = [%.4f, %.4f, %.4f, %.4f]^T", x0);
figure(3)
ax = zeros(4,1);
sgtitle(titleText)
ax(1) = subplot(2,1,1);
    hold on; grid on;
    title("y(1) vs. time")
    measY = plot(t, Ydata(1,:), 'b-');
    predY = plot(t, yPred(1,:), 'r--');
    xlabel("Time [sec]")
    ylabel("y(1)")
    legend([measY, predY], ["Measured output", "Predicted output"],
'location', 'best')
ax(2) = subplot(2,1,2);
    hold on; grid on;
    title("y(2) vs. time")
    plot(t, Ydata(2,:), 'b-')
    plot(t, yPred(2,:), 'r--')
    xlabel("Time [sec]")
    ylabel("y(2)")


x0 =

    0.1000
    0.3000
   -0.3300
   -0.8600
```


Recovered DT system states for k >= 1
given x(0) = [0.1000, 0.3000, -0.3300, -0.8600]T
x(1) vs. time
x(1)
Time [sec]
x(2) vs. time
x(2)
Time [sec]
x(3) vs. time
x(3)
Time [sec]
x(4) vs. time
x(4)
Time [sec]


Predicted DT system outputs for k >= 1
given x(0) = [0.1000, 0.3000, -0.3300, -0.8600]T
ypred(1) vs. time
ypred(1)
Time [sec]
ypred(2) vs. time
ypred(2)
Time [sec]



Problem 1e

New H

```
H2 = [1 0 0 0; 1 0 0 0; 1 0 0 0];

% Compute observability matrix
O2 = [
        H2
        H2*F
        H2*F^2
        H2*F^3
    ];

obsRank2 = rank(O2)
```

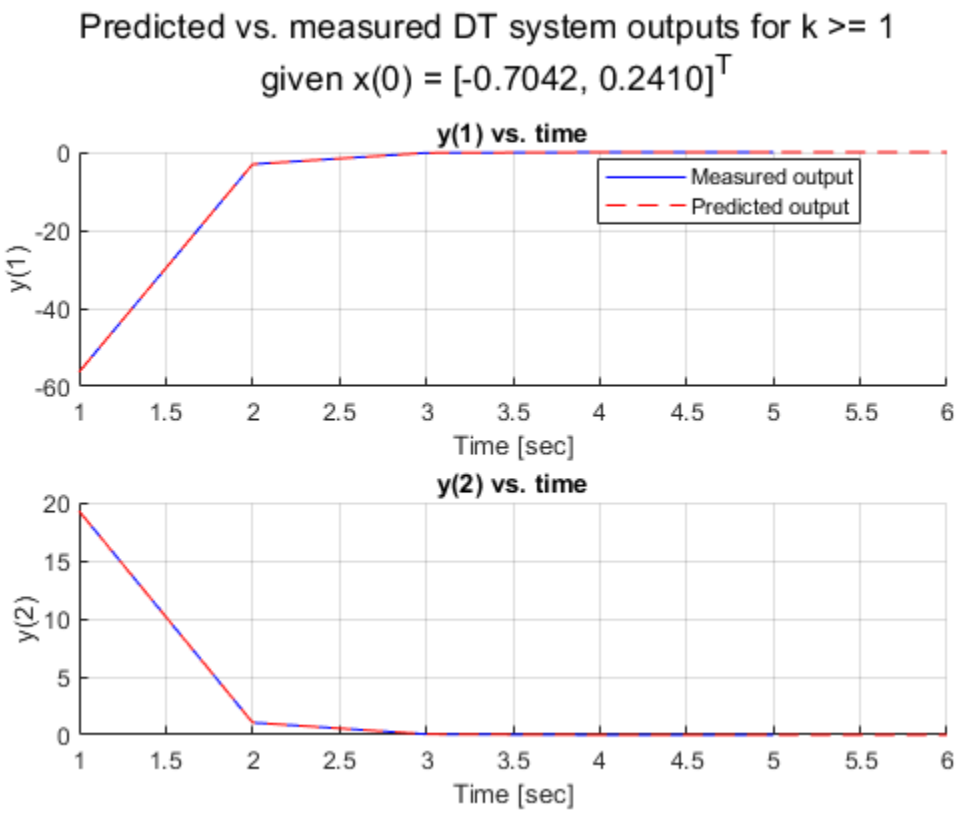
```
obsRank2 =

     4
```

Problem 2c

```
z1 = [100; 43.6658; 40.5785; 40.4093; 40.4; 40.3995];
z2 = [20; 39.2815; 40.3382; 40.3961; 40.3993; 40.3995];
```

```
F2 = eye(2);

Hfunc = @(k) [z1(k) - z2(k) 0; 0 z1(k) - z2(k)];

E2 = [];
Y2 = [];
for k = 1:length(z1)-1
    Y2 = [Y2; z1(k+1) - z1(k); z2(k+1) - z2(k)];
    E2 = [E2; Hfunc(k)];
end

x02 = ((E2'*E2)^-1)*E2'*Y2

t2 = 1:length(z1);

x = [];
y = [];
for k = t2
    x = [x, F2^k*x02];
    y = [y; Hfunc(k)*x(:,k)];
end

titleText = sprintf("Predicted vs. measured DT system outputs for k >= 1 \n
given x(0) = [%.4f, %.4f]^T", x02);
figure(4)
ax = zeros(2,1);
sgtitle(titleText)
ax(1) = subplot(2,1,1);
    hold on; grid on;
    title("y(1) vs. time")
    measY = plot(1:length(Y2)/2, Y2(1:2:end), 'b-');
    predY = plot(t2, y(1:2:end), 'r--');
    xlabel("Time [sec]")
    ylabel("y(1)")
    legend([measY, predY], ["Measured output", "Predicted output"],
'location', 'best')
ax(2) = subplot(2,1,2);
    hold on; grid on;
    title("y(2) vs. time")
    plot(1:length(Y2)/2, Y2(2:2:end), 'b-')
    plot(t2, y(2:2:end), 'r--')
    xlabel("Time [sec]")
    ylabel("y(2)")
```

```
x02 =

   -0.7042
    0.2410
```



Published with MATLAB® R2023b