Table of Contents

Clean

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

Problem 1

```
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];

Ahat = [A, B; zeros(2,6)];
dt = 0.05;

expMat = expm(Ahat .* dt);
F = expMat(1:4, 1:4);
G = expMat(1:4,5:6);
H = C;
M = [0, 0; 0, 0];

omega_sample = 2*pi / dt;

[vec, val] = eig(A);
[vec_2, val_2] = eig(F);

Ob = obsv(F,H);
Gram = Ob'*Ob;
```

Problem 1d

```
load('hw3problem1data.mat'); %Udata and Ydata

% Construct the Y matrix
Y = [];
O = [];
control_mat = zeros(4,1);
for n = 1:length(Ydata(:,1))
    % Control Portion
    % for j = 0:n-1
    %     if j == 0
    %         mat = H*G * Udata(n-j, :)';
```

```
    %     else
    %         mat = mat + H*(F^(j))*G * Udata(n-j, :)';
    %     end
    % end

    % New implementation
    control_mat = F*control_mat + G * Udata(n, :)';

    % Left hand side
    y_mat = Ydata(n,:)' - H*control_mat;

    % Y matrix
    Y = [Y; y_mat];

    % Observability matrix
    O = [O; H*(F^n)];
end

% Solve for the x(0)
x_0 = inv(O' * O) * O' * Y;

% Create the state space
state_space = ss(F,G,H,M,dt);


% Simulate pertubations
times = linspace(dt, dt*length(Ydata(:,1)), length(Ydata(:,1)));
u = Udata(2:end,:);
simx0 = F*x_0 + G*Udata(1,:)';

[yout, tout, xout] = lsim(state_space, u, times, simx0);

% Calculate difference between actual and predicted
y_resid = Ydata - yout;
y_resid = -1*y_resid;
```

Problem 1e

```
Hnew = [1,0,0,0;1,0,0,0;1,0,0,0];
Hnew_2 = [1,0,0,0];

O_new_2 = obsv(F, Hnew_2);
O_new = obsv(F, Hnew);
gram_new = O_new' * O_new;
```

Problem 2

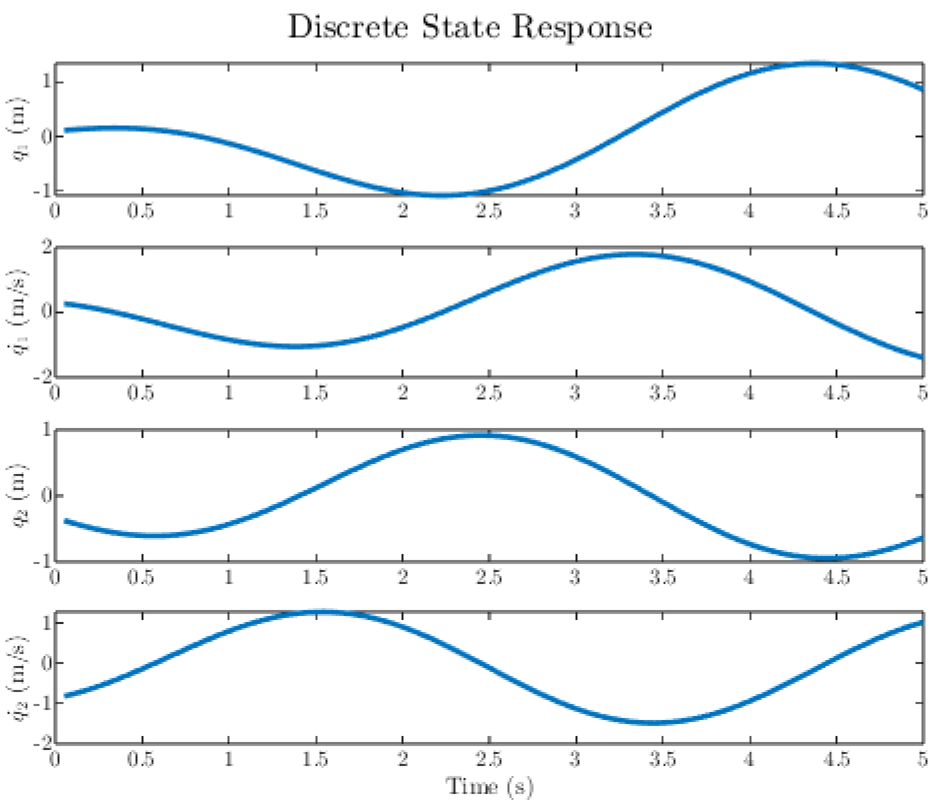
```
z_0 = [100; 20];
z_1 = [43.6658; 39.2815];
z_2 = [40.5785; 40.3382];
z_3 = [40.4093; 40.3961];
z_4 = [40.4000; 40.3993];
z_5 = [40.3995; 40.3995];
```

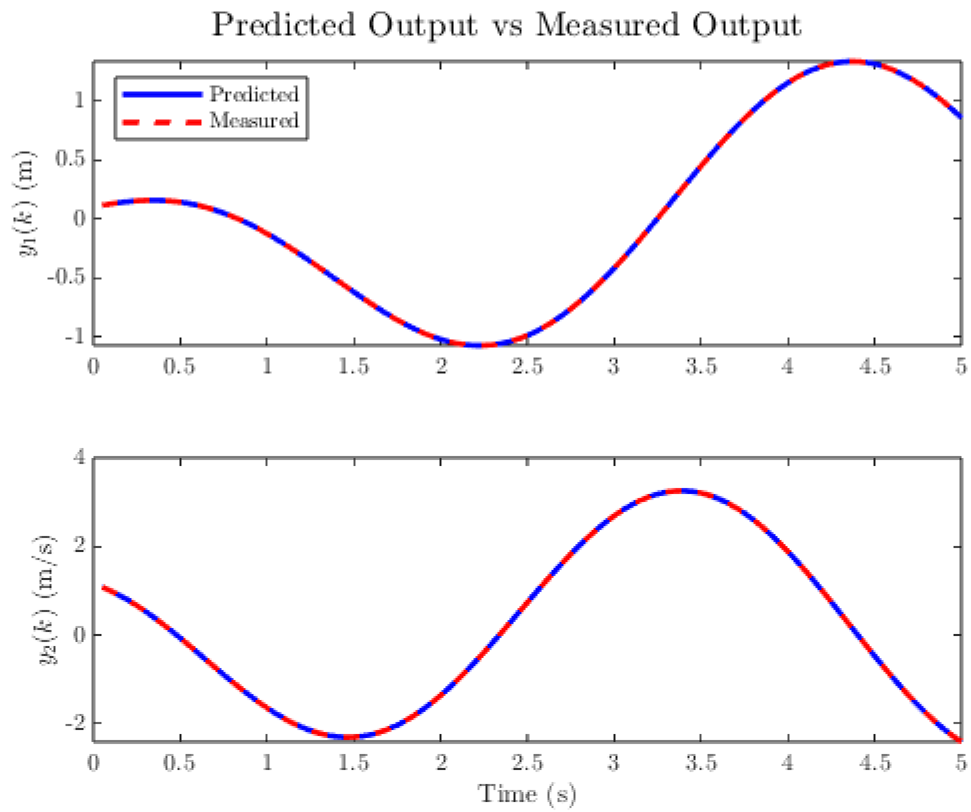
```
Y = [z_1 - z_0;
    z_2 - z_1;
    z_3 - z_2;
    z_4 - z_3;
    z_5 - z_4];

H = [(z_0(1) - z_0(2)) .* eye(2);
    (z_1(1) - z_1(2)) .* eye(2);
    (z_2(1) - z_2(2)) .* eye(2);
    (z_3(1) - z_3(2)) .* eye(2);
    (z_4(1) - z_4(2)) .* eye(2)];


x = inv(H' * H) * H' * Y;
```

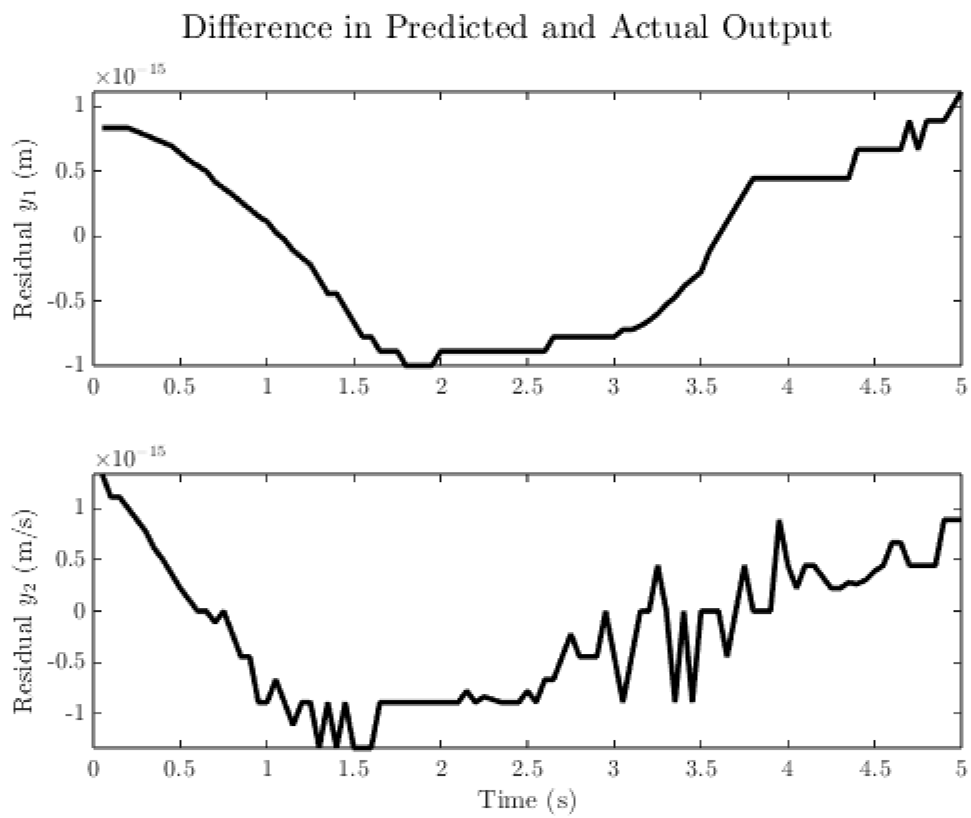
Plotting

```
set(groot, 'DefaultTextInterpreter', 'latex');
set(groot, 'DefaultAxesTickLabelInterpreter', 'latex');
set(groot, 'DefaultLegendInterpreter', 'latex');

figure();
sgtitle("Discrete State Response")
subplot(4,1,1)
plot(tout, xout(:,1), 'linewidth', 2);
ylabel('$q_{1}$ (m)')

subplot(4,1,2)
plot(tout, xout(:,2), 'linewidth', 2);
ylabel('$\dot{q}_{1}$ (m/s)')

subplot(4,1,3)
plot(tout, xout(:,3), 'linewidth', 2);
ylabel('$q_{2}$ (m)')

subplot(4,1,4)
plot(tout, xout(:,4), 'linewidth', 2);
xlabel('Time (s)')
ylabel('$\dot{q}_{2}$ (m/s)')


%%%% Predicted values
figure();
sgtitle('Predicted Output vs Measured Output')
subplot(2,1,1)
plot(tout, yout(:,1), 'LineWidth', 2, 'color', 'b')
hold on
plot(tout, Ydata(:,1), 'LineWidth', 2, 'color', 'r', 'linestyle', '--')

ylabel('$y_{1}(k)$ (m)')
legend('Predicted', 'Measured', 'location', 'nw')
```

```
subplot(2,1,2)
plot(tout, yout(:,2), 'LineWidth', 2, 'color', 'b')
hold on
plot(tout, Ydata(:,2), 'LineWidth', 2, 'color', 'r', 'linestyle', '--')

xlabel('Time (s)')
ylabel('$y_{2}(k)$ (m/s)')


%%%% Residuals
figure();
sgtitle('Difference in Predicted and Actual Output')
subplot(2,1,1)
plot(tout, y_resid(:,1), 'LineWidth', 2, 'color', 'k')

ylabel('Residual $y_{1}$ (m)')

subplot(2,1,2)
plot(tout, y_resid(:,2), 'LineWidth', 2, 'color', 'k')

xlabel('Time (s)')
ylabel('Residual $y_{2}$ (m/s)')
```




Predicted Output vs Measured Output
Predicted
Measured
$y_1(k)$ (m)
$y_2(k)$ (m/s)
Time (s)


Difference in Predicted and Actual Output
$\times 10^{-15}$
Residual y_1 (m)
Residual y_2 (m/s)
Time (s)

Published with MATLAB® R2023b