

Problem 2 Matlab Code:

```
syms z1 z2 eps1 eps2 t t0 s
f = [2*z1-z1*z2; 2*z1^2 - z2];
x = [z1; z2];
A = jacobian(f,x); %symbolic A matrix (needs evaluated at equilibrium)

%For equilibrium point z1 = 0 and z2 = 0
A1 = double(subs(A,{z1,z2},{0,0})); %substitute in equilibrium values
%calculate state transition matrix using inverse laplace
st_tr_1 = simplify(subs(ilaplace(inv(s*eye(2) - A1), s,t),t,t-t0))
%calculate state transition matrix using Cayley-Hamilton Theorem
eig_A1 = eig(A1);
Vantermonte = [1 eig_A1(1); 1 eig_A1(2)];
b = [exp(eig_A1(1)*t);exp(eig_A1(2)*t)];
alpha = Vantermonte\b;
st_tr_1_Cayley = simplify(subs(alpha(1)*eye(2) + alpha(2)*A1,t,t-t0));
%calculate state transition matrix using eigenvalue eigenvectors
[V,D] = eig(A1);
st_tr_1_eig = simplify(V*expm(D*(t-t0))*inv(V));
%calculate state transition matrix using matlabs expm
st_tr_1_matlab = simplify(expm(A1*(t-t0)));

%For equilibrium point z1 = 1 and z2 = 2
A2 = double(subs(A,{z1,z2},{1,2}));
%calculate state transition matrix usgin inverse laplace
st_tr_2 = simplify(subs(ilaplace(inv(s*eye(2) - A2), s,t),t,t-t0))
%calculate state transition matrix using Cayley-Hamilton Theorem
eig_A2 = eig(A2);
Vantermonte = [1 eig_A2(1); 1 eig_A2(2)];
b = [exp(eig_A2(1)*t);exp(eig_A2(2)*t)];
alpha = Vantermonte\b;
st_tr_2_Cayley = simplify(subs(alpha(1)*eye(2) + alpha(2)*A2,t,t-t0));
%calculate state transition matrix using eigenvalue eigenvectors
[V,D] = eig(A2);
st_tr_2_eig = simplify(V*expm(D*(t-t0))*inv(V));
%calculate state transition matrix using matlabs expm
st_tr_2_matlab = simplify(expm(A2*(t-t0)));

%For equilibrium point z1 = -1 and z2 = 2
A3 = double(subs(A,{z1,z2},{-1,2}));
%calculate state transition matrix using inverse laplace
st_tr_3 = simplify(subs(ilaplace(inv(s*eye(2) - A3), s,t),t,t-t0))
%calculate state transition matrix using Cayley-Hamilton Theorem
eig_A3 = eig(A3);
Vantermonte = [1 eig_A3(1); 1 eig_A3(2)];
b = [exp(eig_A3(1)*t);exp(eig_A3(2)*t)];
alpha = Vantermonte\b;
st_tr_3_Cayley = simplify(subs(alpha(1)*eye(2) + alpha(2)*A3,t,t-t0));
%calculate state transition matrix using eigenvalue eigenvectors
[V,D] = eig(A3);
st_tr_3_eig = simplify(V*expm(D*(t-t0))*inv(V));
%calculate state transition matrix using matlabs expm
st_tr_3_matlab = simplify(expm(A3*(t-t0)));
```

Problem 5 Matlab Code:

% Create State Space System

```
A = [-0.1, 2; 0, -1];
```

```
B = [10; 0.1];
```

```
C = [0.1, -1];
```

```
D = 0;
```

```
sys = ss(A,B,C,D);
```

% Part (a) Calculate state variables and output for a unit step input

```
[y, t, x] = step(sys);
```

% Part (b) Plot and label the two state variables and output over time

```
figure(1)
```

```
subplot(3,1,1)
```

```
plot(t,y);
```

```
ylabel('y')
```

```
title('System Unit Step Response')
```

```
subplot(3,1,2)
```

```
plot(t,x(:,1))
```

```
ylabel('x_1')
```

```
subplot(3,1,3)
```

```
plot(t,x(:,2))
```

```
ylabel('x_2')
```

```
xlabel('Time (sec)')
```

% Part (c) Find similarity transform matrix T

%similarity transform

```
T = [0.1, 0; 0, 200];
```

```
Tinv = inv(T);
```

% Part (e) Plot states and output of equivalent system for unit step input

%put together state space system

```
Abar = T*A*Tinv;
```

```
Bbar = T*B;
```

```
Cbar = C*Tinv;
```

```
Dbar = 0;
```

```
sysbar = ss(Abar, Bbar, Cbar, Dbar);
```

%compute state variables and output for a step input

```
[ybar, tbar, xbar] = step(sysbar);
```

%plot and label the two state variables and output over time

```
figure(2)
```

```
subplot(3,1,1)
```

```
plot(tbar,ybar);
```

```
ylabel('y response')
```

```
title('Equivalent System Unit Step Response')
```

```
subplot(3,1,2)
```

```
plot(tbar,xbar(:,1))
```

```
f1 = ylabel('$\bar{x}_1$');
```

```
set(f1, 'interpreter', 'Latex', 'FontSize', 18)
```

```
subplot(3,1,3)
```

```
plot(tbar,xbar(:,2))
```

```
f = ylabel('$\bar{x}_2$');
```

```
set(f, 'interpreter', 'Latex', 'FontSize', 18)
```

```
xlabel('Time (sec)')
```



