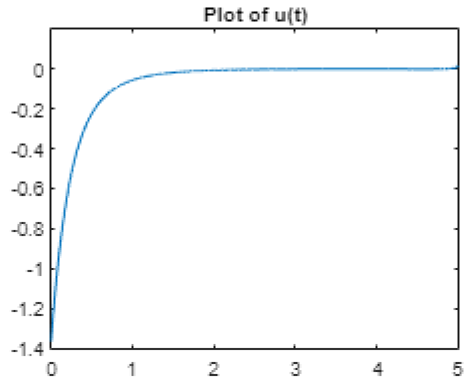
24-677 Modern Control Theory
Project 3

Exercise 1:

1. **Finite CT LQR**

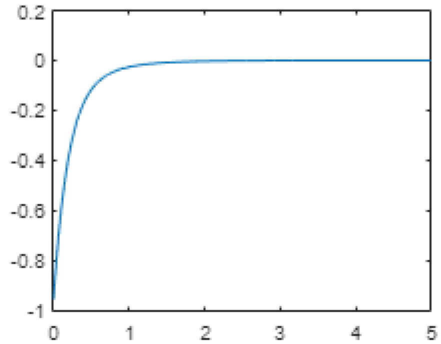


Graph of Finite CT LQR u(t)

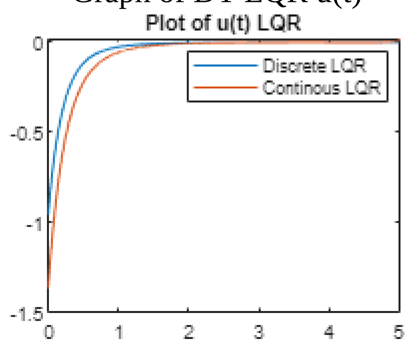
```
%Part 1
%System Dynamics
A = [0 1; -10 -7];
B = [0; 1];
%Control Setup
S = eye(2) * 20;
Q = diag([5 1]);
R = 0.25;
time_span_reverse = linspace(5, 0, 5000);
time = linspace(0, 5, 5000);
u_arr = [];
x_0 = [1;1];
%-----------------------------------------------%
%Solve LQR Problem
[t, P] = ode45(@(t, P) ricatti(t, P, A, B, Q, R), time_span_reverse, reshape(S, [4, 1]));
P = flip(P);
for i = 1:length(P)
    K(i,:) = -inv(R)*B'*reshape(P(i,:),[2,2]);
end
[t_new,x] = ode45(@(t_new,x) xcalc(x, A, B, K), time, x_0);
x = transpose(x);
for ii = 1:length(K)
    u_arr(ii) = -inv(R) * transpose(B) * [P(ii,1), P(ii,2); P(ii,3), P(ii,4)] * x(:,ii);
end
plot(t_new, u_arr)
ylim([-1.4 0.2])
title("Plot of u(t)")

function dpdt = ricatti(t, P, A, B, Q, R)
    P = reshape(P, size(A));
    dpdt = P*B*R^-1*transpose(B)*P - Q-P*A-transpose(A)*P;
    dpdt = dpdt(:);
end

function dxdt = xcalc(x, A, B, K)
    dxdt = (A-B*K(1,:))*x;
end
```

Code for Finite Continuous u(t)

2. DT LQR



Graph of DT LQR u(t)



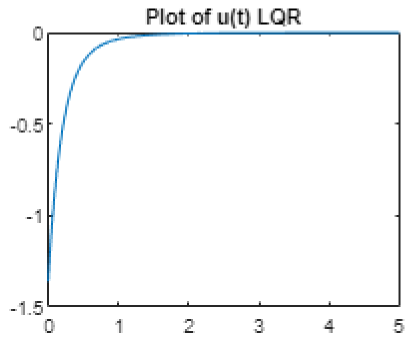
Graph of Comparison between Discrete and Finite Continuous LQR u(t)

The discrete LQR method and finite continuous LQR method both have very similar plots of the u(t) with only slight variation of the former starting at around -0.975 and the latter around -1.4. The continuous method has a stronger of higher input than the discrete method.

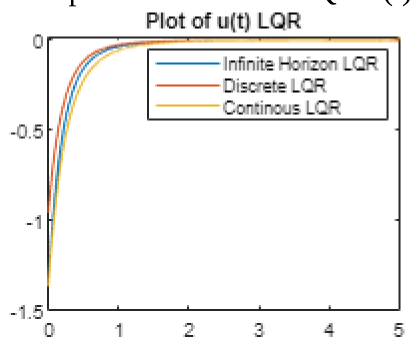
```
%Part 2
%System Dynamics
A = [0 1; -10 -7];
B = [0; 1];
%Control Setup
S = eye(2) * 20;
Q = diag([5 1]);
R = 0.25;
%Solve LQR Problem
step = 0.005;
N = 1000;
[M, A_hat] = eig(A);
A_new = M * [exp(step * A_hat(1,1)), 0; 0, exp(step * A_hat(2,2))] * inv(M);
B_new = inv(A) * (A_new - eye(2)) * B;
K_arr = [];
for ii = N:-1:1 %Discrete Time Approach
    K(ii,:) = inv(R + transpose(B_new) * S * B) * transpose(B_new) * S * transpose(A_new);
    S = transpose(A_new - B_new * K(ii,:)) * S * (A_new - B_new * K(ii,:)) + Q + transpose(K(ii,:)) * R * K(ii,:);
end
x_arr = [];
x_arr(:,1) = [1;1];
u_arr = [];
%Simulate Performance
for i = 1:N
    u_arr(i) = -K(i,:) * x_arr(:,i);
    x_arr(:, i + 1) = A_new * x_arr(:,i) + B_new * u_arr(i);
end
time = [0:step:5 - step];
plot(time, transpose(u_arr))
```

Code for DT LQR

3. Infinite Horizon LQR



Graph of Infinite CT LQR u(t)



Graph of Comparison between Infinite Horizon, Discrete and Continuous LQR u(t)

Here we can see that the infinite CT and the finite CT curve look almost identical. With that said, we can assume that this means that they both put a stronger input signal into the system compared to the DT curve initially.

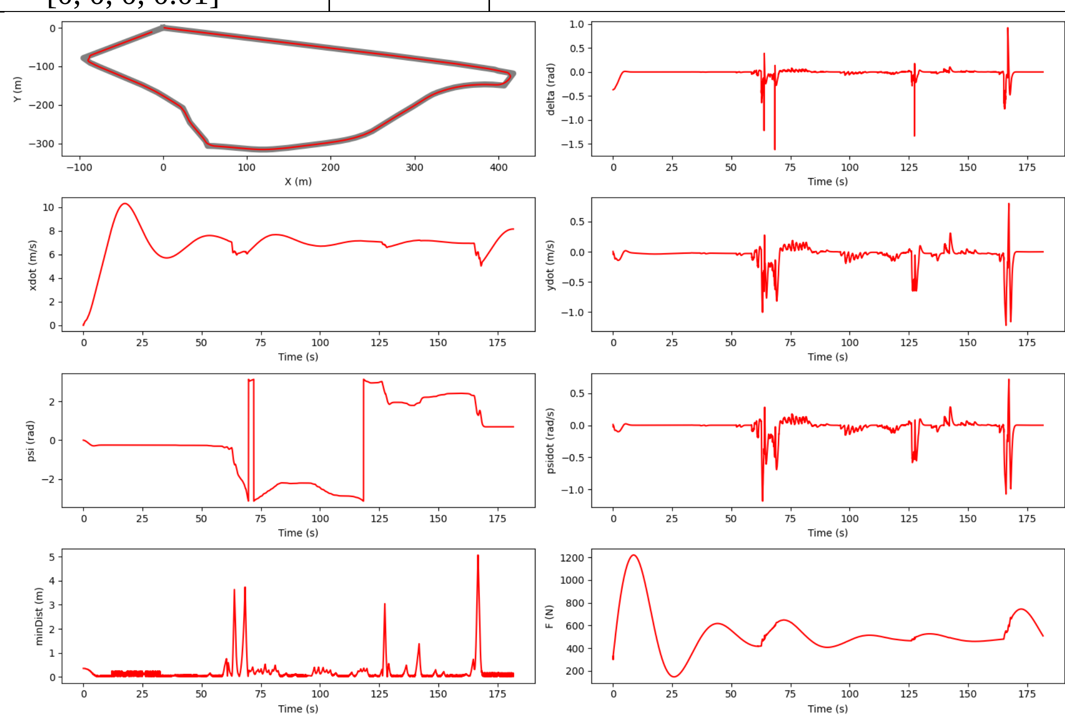
```
%Part 3 Infinite Horizon
%System Dynamics
A = [0 1; -10 -7];
B = [0; 1];
%Control Setup
S = eye(2) * 20;
Q = diag([5 1]);
R = 0.25;
%Solve LQR Problem
K = -lqr(A, B, Q, R);
time_arr = 0:0.01:5;
size_time = size(time_arr);
u_arr_inf = zeros(size_time);
x_arr = [1; 1];
x = lsim(A + B * K, B, [1, 0; 0, 1], 0, u_arr_inf, time_arr, x_arr);
figure(3)
plot(time_arr, (K*x')');
hold on;
plot(time,u)
hold on
plot(t_new, u_arr')
legend('Infinite Horizon LQR','Discrete LQR', 'Continous LQR')
title("Plot of u(t) LQR")
xlim([0 5])
hold off
```

Code for Infinite Horizon LQR

Exercise 2:

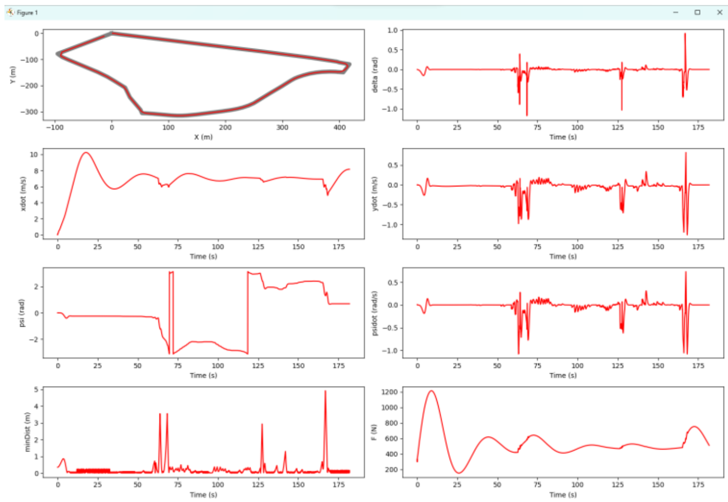
1. **Tuning Q and R**

Q Values	R Values	Results
[1, 0, 0, 0] [0, 1, 0, 0] [0, 0, 1, 0] [0, 0, 0, 1]	20	The car crashes at the first corner and tips over so the weight of the psi needs to change so it can turn with less punishement.0.
[1, 0, 0, 0] [0, 1, 0, 0] [0, 0, 0.1, 0] [0, 0, 0, 1]	20	The car made it around the first corner but it fell after it made the turn and tried to adjust. For this reason, increase R to make it more stable. We also want e2dot to have less of a penalty to change.
[1, 0, 0, 0] [0, 1, 0, 0] [0, 0, 0.1, 0] [0, 0, 0, 0.01]	75	The car seems to be more stable as it travels along the track but it won't follow the center of the track properly. I want to increase the penalty of e1.
[10, 0, 0, 0] [0, 1, 0, 0] [0, 0, 0.1, 0] [0, 0, 0, 0.01]	75	The car moves very well but it's a little weird in changing to the right reference on the track so I want to decrease the e1dot penalty.
[10, 0, 0, 0] [0, 0.1, 0, 0] [0, 0, 0.1, 0] [0, 0, 0, 0.01]	75	This result seems to match the expected result as it reaches the end within 181 seconds with a very small delta deviation of 0.006.



2. Tuning N

N Value	Results
100	Car follows track, very slow to follow though as it took 210 sec, will decrease amount of N runs to try for faster time
75	Once again, limited deviation from track and achieves 100% as time is 198 sec but I want to have a better time by decreasing N again
30	Car completes course with minimal deviation and in about 180 sec with 100%



Exercise 3:

1. LQR

```
Evaluating...
Score for completing the loop: 22.5/22.5
Score for average distance: 22.5/22.5
Score for maximum distance: 22.5/22.5
Score for average delta fluctuation: 22.5/22.5
Your time is  181.66400000000002
Your total score is : 100.0/100.0
total steps:  181664
maxMinDist:  5.071235585988137
avgMinDist:  0.213716988927932
deltaDev: 0.005916072498847916
```

2. MPC

```
Evaluating...
Score for completing the loop: 22.5/22.5
Score for average distance: 22.5/22.5
Score for maximum distance: 22.5/22.5
Score for average delta fluctuation: 22.5/22.5
Your time is  181.856
Your total score is : 100.0/100.0
total steps:  181856
maxMinDist:  4.915523442875601
avgMinDist:  0.21213768891522528
deltaDev: 0.005038468407939659
INFO: 'main' controller exited successfully.
```