
```
% Jacob Anderson and Justin Francis
% MEEN 5210, State Space Lab
% Dr. Abbot, U of U
% DC Motor Lab
```

part 4

```
clc;
clear; close all;

load('mathModel.mat');
load('robust.mat');

%finding A_tilde
syms k1 k2 ka
K_lqr = [k1 k2 ka];
B_lqr = [-B; 0];
A_lqr = double(A_rt + B_lqr*K_lqr);

%form Q and R
%diag
q11 = 1;
q22 = 1e7;
q33 = 1e9;
%off diag
q12 = 0;
q13 = 0;
q23 = 0;

Q = [[q11 q12 q13];[q12 q22 q23];[q13 q23 q33]]; %cost applied to
states

R = 1e4; %cost applied to inputs

[K S CLP] = lqr(A_lqr, B_lqr, Q, R);

K_lqr = K(1:2);
ka = K(end);

% model
lqrSim = sim('part4model.slx');

Ts = findSettlingTime(lqrSim.tout, lqrSim.pos)

RMSE = findRMSE(lqrSim.tout, lqrSim.pos, 1)

poles_cl = eig(A_lqr-B_lqr*K)

%plot
```

```

figure();
hold on;
plot(lqrSim.tout, lqrSim.pos, 'DisplayName', 'System Output');
plot(Ts, lqrSim.pos(find(lqrSim.tout ==
    Ts)), 'ro', 'DisplayName', 'Settling Time = 0.320 [s]');
title('LQR Method on Robust Tracking Controller');
xlabel('Time, t[s]');
ylabel('Position, \theta[rad]');
grid();
legend('Location', 'southeast');

%save data for next part
save('lqrPoles.mat', 'CLP', 'K_lqr', 'ka', 'encGain');

% commentary
comments = ['The improvement was a %.3f percent decrease in settling
    time' ...
    ' when compared to the robust tracking controller in part 3.'];
fprintf(comments, 80.618);

```

Ts =

0.3200

RMSE =

0.0047

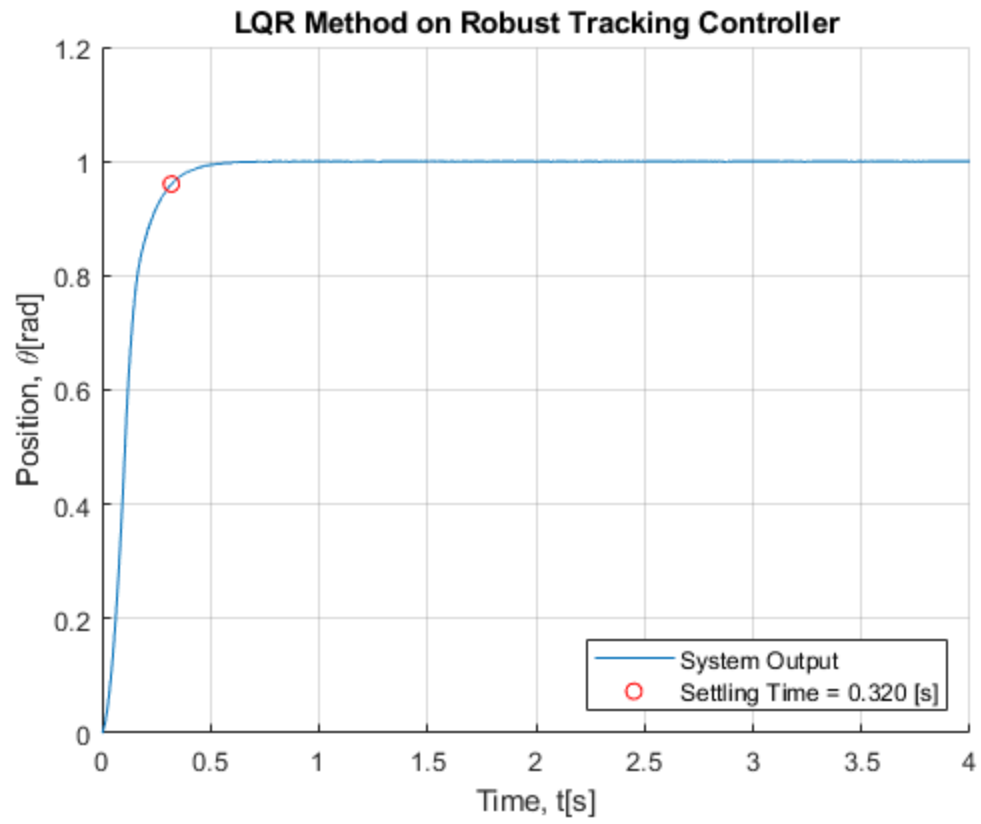
poles_cl =

-40.0859 +40.6657i

-40.0859 -40.6657i

-9.9956 + 0.0000i

The improvement was a 80.618 percent decrease in settling time when compared to the robust tracking controller in part 3.



Published with MATLAB® R2019b