Bus Suspension

Lakhan Shiva Kamireddy and Christopher Rosemann

University of Colorado, Boulder CO 80302, USA
{lakhan.kamireddy,christopher.rosemann}@colorado.edu

Abstract. Linear Control Systems have linear dyamics. Such systems have been extensively analysed and studied. In this project we analyse one such system known as bus suspension, that has linear system dynamics. We use a range of analysis and design techniques learnt in the Linear Control Systems class, ASEN 5014. We model the state space of the system, establish control system objectives, perform stability analysis, design a state feedback controller and an observer by pole placement technique and optmize an infinite-horizon cost function to solve for optimal closed loop poles of the system. We use a numerical computing tool known as Matlab to model and analyse the system.

1 Introduction

1.1 Bus Suspension

Modern vehicles have seat suspension in addition to wheel suspension. We are considering a bus driver's seat. Oscillations caused by bumps and potholes on roads have a significant impact on the driver's comfort and must be reduced by the suspension of the seat. Given the long hours and poor road conditions, in this project we consider to design an active seat suspension system would improve the driver's comfort.

2 Physical System

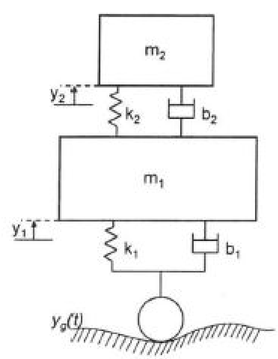

Fig. 1. Seat suspension model

3 State Space Model

The state space model for this system is
$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u}$
$\boldsymbol{y} = C\boldsymbol{x} + D\boldsymbol{u}$

3.1 Description

The state vector model is given by $\boldsymbol{x} = [f_{k2}\ f_{k1}\ v_{m2}\ v_{m1}]^T$. f_{ki} is equal to the force through the respective springs, and v_{mi} is the inertial velocity of each mass. The measured outputs in this system are the acceleration of the driver's seat a_d (measured by accelerometer in the seat), and the acceleration of the wheel body a_b (measured by the accelerometer on the bus chassis).

3.2 Control System Objectives

The following are the control system objectives that we have chosen:

- STABILITY Improve stability, speed of response, steady-state error, reduce Oscillations
- RESPONSE Improved transient response of the system in terms of overshoot, settling time and rise time
- CONTROLLER Design a controller by carefully selecting the poles of the system.
- OBSERVER Design an observer for estimating the state of the system by placing its poles at desired location, such that there is a decay in the state observation error to zero.
- LQR Develop an optimisation based controller (LQR) that minimizes infinite-horizon cost function and solves for the corresponding optimal state feedback law.

3.3 System properties

Poles of the system:
-4.0866 + 7.6030i
-4.0866 - 7.6030i
-1.4734 + 2.7698i
-1.4734 - 2.7698i

Controllability Investigate the possibility of steering the state x(t) to a desired point in R^n by manipulating u(t). The following is the controllability matrix, C. Controllability matrix, C = $[B\,AB\,A^2B\,...A^{n-1}B]$. Rank of the controllability matrix, C = 4 = Rank(A). Hence the system is controllable.

$$\begin{bmatrix} 0 & 0 & 74.434 & -33000 & -607.7 & 256960 & -580.14 & 720920 \\ 0 & -100000 & -10 & 300000 & 111.2 & 64000 & -152.2 & -3241680 \\ 0.0067 & 0 & -0.0541 & 24 & -0.0543 & 33.12 & 4.473 & -2237 \\ -1\times10^{-4} & 3 & 0.0011 & 0.64 & -0.0015 & -32.42 & -0.074 & 124.41 \end{bmatrix}$$

Observability A linear system is observable if it is possible to determine the state of the system x(t) through measurements of y(t) and u(t). The following is the observability matrix, O. Observability matrix, O = $[C\,CA\,CA^2\,...CA^{n-1}]^T$. Rank of the observability matrix, C = 4 = Rank(A). Hence the system is observable.

$$\begin{bmatrix} -0.0067 & 0 & -8 & 8 \\ 1\times10^{-4} & -1\times10^{-4} & 0.12 & -3.12 \\ 0.0541 & -8\times10^{-4} & -8.374 & -15.63 \\ -0.0011 & 3.12\times10^{-4} & -0.234 & -4.06 \\ 0.0543 & 0.0016 & 660.584 & -693.705 \\ 0.0015 & 4.056\times10^{-5} & -10.41 & 42.82 \\ -4.47 & 0.0694 & -4771 & 7008.4 \\ 0.0737 & -0.0043 & 105.126 & -229.537 \end{bmatrix}$$

Stability The system is stable since the poles have a negative real part, i.e. $\mathrm{Re}(\lambda) < 0$. The system is Bounded Input Bounded Output (BIBO) stable.

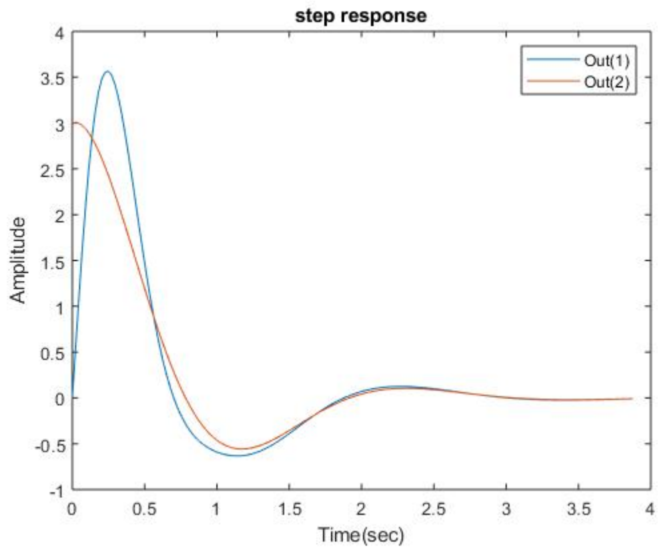

Fig. 2. Initial step response of the system

Stabillizability Since the system is fully controllable, it is therefore stabilizable.

Detectability Since the system is fully observable, it is therefore detectable.

3.4 State feedback controller

The objective is to design a state feedback controller to place poles in desired locations. We simulate the closed loop system under step response. We also discuss whether the closed loop response corresponds with what is expected considering the desired closed loop pole locations.

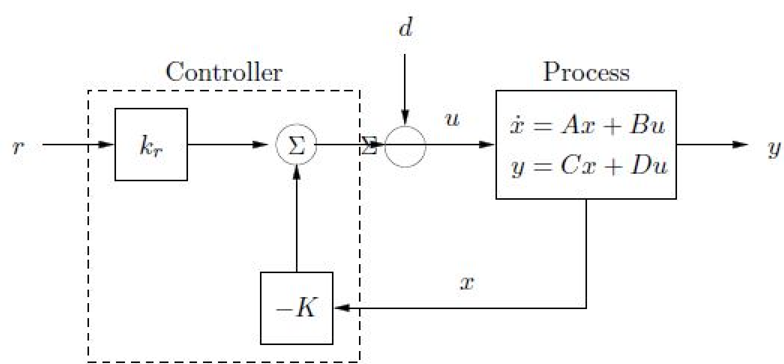

Fig. 3. Block diagram of state feedback control

We take kr as F in the equations. Closed-loop equations: 1) Plant equations

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u}$$

$$\boldsymbol{y} = C\boldsymbol{x} + D\boldsymbol{u}$$

2) Feedback control

$$\boldsymbol{u} = -K\boldsymbol{x} + F\boldsymbol{r}$$

Determining the gain matrix K is the objective of the full-state feedback design procedure. Approach followed: Use time-domain specifications to locate dominant poles - roots of $s^2 + 2\epsilon\omega_n + {\omega_n}^2 = 0$. We picked the values for ω_n and ϵ, we chose $\omega_n = 27.89$ rad/s and $\epsilon = 1$ (Critical damping). $s^2 + 100s + 2500 = 0$. We get two poles which are dominant. The other poles are selected to have faster response than these. We have the desired closed-loop poles as:
p = [-27 + 7i -27 - 7i -27.5 + 7i -27.5 - 7i]

Fig. 3 illustrates the block diagram of the closed loop system with state feedback.

Since the poles are moved towards left on the negative x-axis, we see that the response is faster. We have less settling time. We use F, the input scaling as shown in the equations above to reduce the steady state error.

$$F = -(C(A - BK)^{-1}B)^{-1}$$

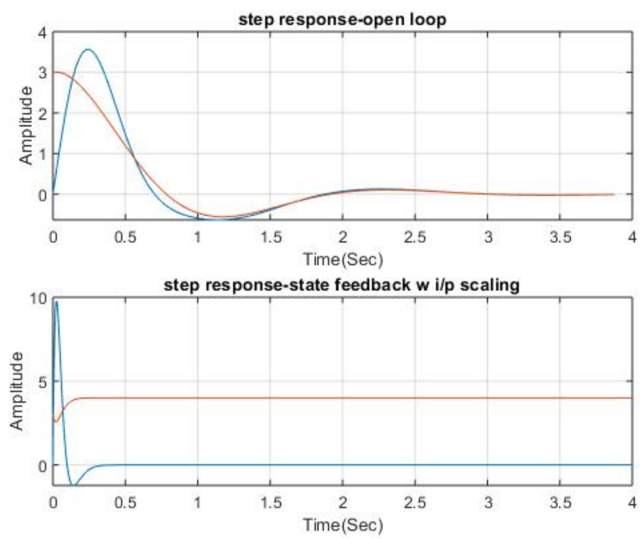

Fig. 4. Step response after controller design

In Fig. 4 we see that the closed loop step response's settling time has reduced significantly in comparison to the response of open loop system, which improves the driver seat's comfort by damping the oscillations in a shorter time. The percentage reduction in settling time = 92%. This reduction in settling time was desired.

In Fig. 5 we see that the steady state error has reduced after scaling input by F.

3.5 Observer

The objective is to design an observer to reconstruct the state. We discuss how we determined the desired observer poles. We simulate the closed loop system consisting of observer and state variable feedback. We also verify that the state observation error goes to zero at the desired rate. We compare the closed loop response with that obtained in previous subsection.

$\hat{x}$ is an estimation of the state. We design an observer that takes the inputs and outputs of the system we are observing and produces an estimate of the system's state.

To design an observer 1) We use a model of the system (A, B, C, D) to predict the system output 2) We use the difference between the predicted and measured output to drive the change in the estimate. We call the difference as Innovation.

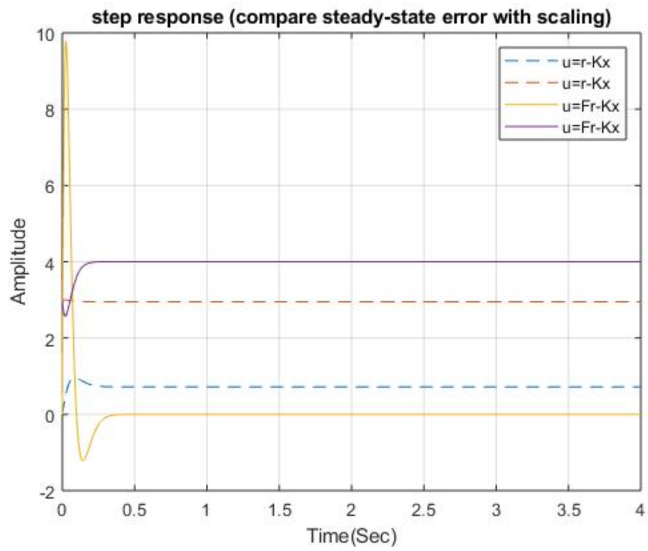

Fig. 5. Steady state error comparison w.r.t. input scaling

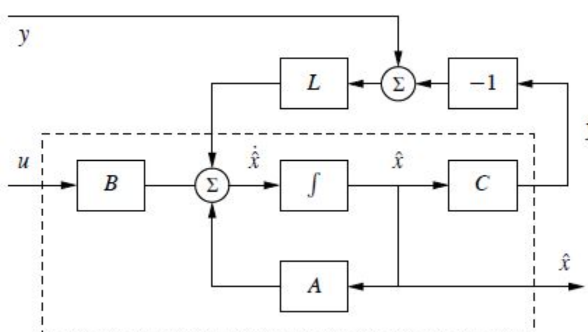

Fig. 6. Observer block diagram

$$\boldsymbol{u} = -K\boldsymbol{x} + F\boldsymbol{r}$$

$$\dot{\hat{\boldsymbol{x}}} = A\hat{\boldsymbol{x}} + B\boldsymbol{u} + L(\boldsymbol{y} - \hat{\boldsymbol{y}})$$

We choose L to provide $\hat{\boldsymbol{x}}$.

$$\dot{\hat{\boldsymbol{x}}} = A\hat{\boldsymbol{x}} + B\boldsymbol{u} + L(\boldsymbol{y} - C\hat{\boldsymbol{x}})$$

$$\dot{\boldsymbol{e}} = (A - LC)\boldsymbol{e}$$

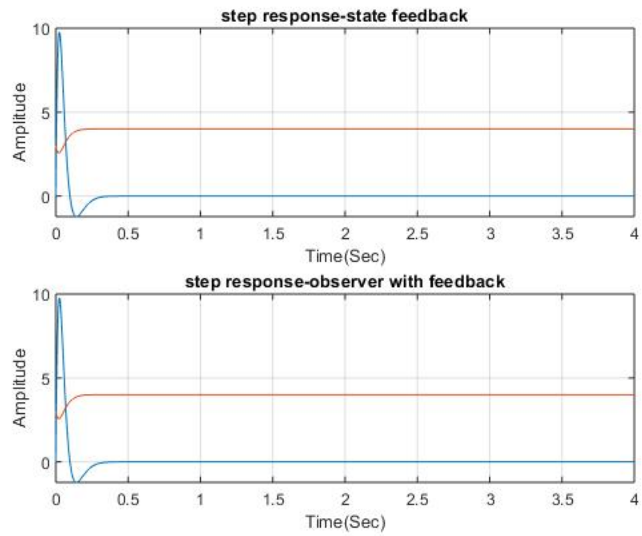

Fig. 7. Step response after observer design

The objective is to design an observer with desired poles is equivalent to finding a matrix L so that A - LC has desired eigenvalues. According to the separation theorem, observer design and controller pole placement are independent of each other. Since here the observer states are used for state feedback, the slowest eigenvalues of A - LC should be faster than the eigenvalue of the state feedback system A - BK. Feedback is implemented by assuming that the observer state estimate is the actual state.
Desired observer pole locations:
q = [-120.0 + 119.4i -20 - 119.4i -119.5 + 118.8i -119.5 - 118.8i]

Fig. 6. illustrates the observer block diagram. Fig. 7. compares the closed loop system's step response with that of the step reponse obtained in the previous section.

In Fig. 8. we see that the state observation error is decaying to zero. We can see it from the plot of the state observation error below. We chose the observer gain such that all eigenvalues of the matrix A - LC are in the left half plane. In Fig. 9. we illustrate that we can adjust the rate at which the observer error goes to zero by adjusting the values of the observer gain to adjust the locations of these eigenvalues.

3.6 Optimising cost function

The objective is to develop an infinite-horizon cost function, and solve for the corresponding optimal state feedback law. We also need to find where the optmial closed loop poles need to be located. We implement this (with an observer for the state) in simulation, and compare the response to that obtained in previous subsection.

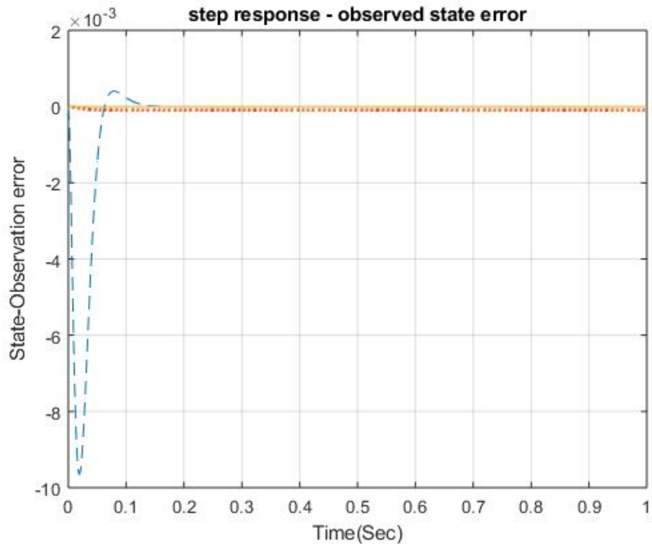

Fig. 8. Step response of decaying observed state error

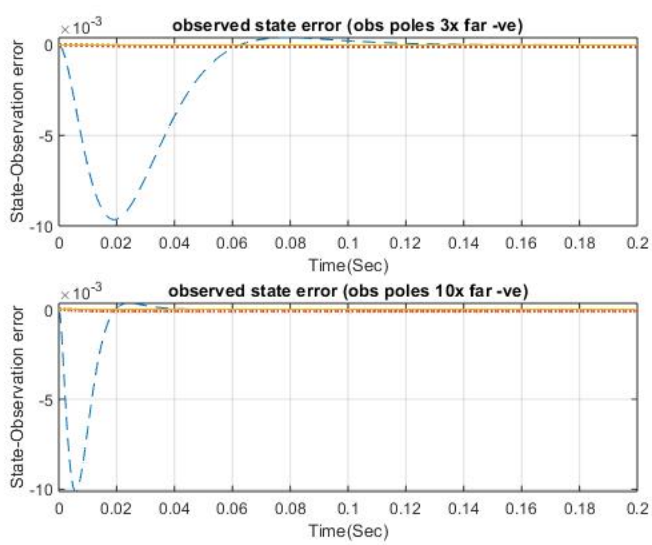

Fig. 9. Comparing observed state error decay by moving observer poles

The infinite-horizon cost function is given by

$$J(u) = \frac{1}{2}\int_{t_0}^{t_f} (x^T Q x + u^T R u) dt + \frac{1}{2} x^T(t_f) S x(t_f)$$

The problem is to find u such that for any time t $\in [t_0, t_f]$ the cost function is minimized. In other words, solve for $u^*(t)$ = arg min J(u) with the constraints being the dynamics of the system. We used matlab's lqr command to solve for the controller gain.

Optimal pole locations obtained by this technique:
-15.5301 + 1.5703i
-15.5301 - 1.5703i
-2.4898 + 1.8218i
-2.4898 - 1.8218i

In Fig. 10. we compare the step response of the closed loop system whose poles are placed by LQR controller design with that of the step response obtained in previous section. We see an improvement in transient characteristics of the system.

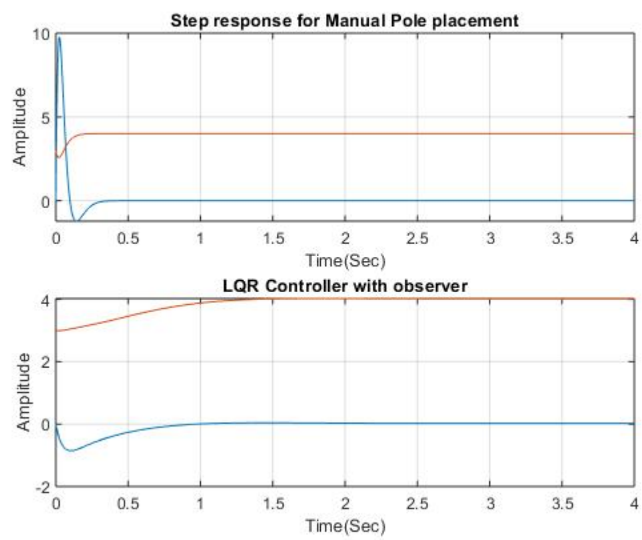

Fig. 10. Comparing step response for lqr control and manual pole placement

4 Conclusion

Initially we have the open loop response of the active seat suspension system. The characteristics of the system were found to be controllable and observable. We designed a closed loop state feedback controller for the system by manually placing the poles at desired locations. We thereby imporoved some characteristics of the system. We achieved a reduction of about 92% in settling time and also a reduction in steady state error. We then designed an observer by placing its poles at desired locations. We found the state observation error to exponentially decay and were able to adjust the rate of decay by moving its poles on the x-axis. We then developed an infinite-horizon cost function which we formulated as an optmization problem and solved for optimal controller poles. We found this to give us better system characteristics. In addition to decrease in settling time and steady state error, we were also able to achieve a reduction in peak overshoot. This improvement in step response characteristics directly translates into improving the comfort of the driver's seat. Thus, we achieved the goals set forth for our project.

5 Appendix A - Matlab Code

```
clear all;close all;clc;

A = [0 0 11000 -11000; 0 0 0 100000; -0.0066667 0 -8 8; 0.0001 -0.0001
    0.12 -3.12];
B = [0 0; 0 -100000; 0.0066667 0; -0.0001 3];
C = [-0.0066667 0 -8 8; 0.0001 -0.0001 0.12 -3.12];
D = [0.0066667 0; -0.0001 3];
G = ss(A,B,C,D);
s = stepinfo(G);
[y,t,x] = step(G);
plot(t,y(:,:,2));
title('step response');
xlabel('Time(sec)');
ylabel('Amplitude');
legend('Out(1)','Out(2)');

%Controllability Matrix
Qc = ctrb(G);
rankQc = rank(Qc);
if(rankQc == rank(A))
    disp('System is controllable');
else
    disp('System is not controllable');
end

%Observability Matrix
Qo = obsv(G);
rankQo = rank(Qo);
if(rankQo == rank(A))
    disp('System is observable');
else
    disp('System is not observable');
end

%Pole Placement
% Place command computes the gain matrix K
% state feedback u = -Kx
% eig vals of A-BK
p = [-27 + 7i -27 - 7i -27.5 + 7i -27.5 - 7i];

K = place(A,B,p);

AminusBk = A - B*K;
%To reduce the steady state error
F = -inv(C*(inv(AminusBk))*B);
Bf = B*F;
G1 = ss(AminusBk, B, C, D);
G2 = ss(AminusBk, Bf, C, D);
s1 = stepinfo(G1);
[y1, t1, x1] = step(G1,4);
[y2, t2, x2] = step(G2,4);
figure(2); clf;
subplot(2,1,1); plot(t,y(:,:,2)); hold on; grid on; ylabel('Amplitude')
    ;
xlabel('Time(Sec)');
```

```
title('step response-open loop');
subplot(2,1,2); plot(t2,y2(:,:,2)); grid on; ylabel('Amplitude');
xlabel('Time(Sec)');
title('step response-state feedback w i/p scaling');

%Compare steady-state error before scaling and after
figure(3); clf;
plot(t1,y1(:,:,2),'--'); hold on; plot(t2,y2(:,:,2)); grid on; hold off; ylabel('Amplitude');
legend('u=r-Kx', 'u=r-Kx','u=Fr-Kx','u=Fr-Kx');
xlabel('Time(Sec)');
title('step response (compare steady-state error with scaling)');

%Observer Design
%Use place to calculate L
%Generally estimator dynamics is faster than the observer dynamics
q = 3*[-27 + 7i -27 - 7i -27.5 + 7i -27.5 - 7i];
L = place(A',C',q).';

AminusLC = A - L*C;
Bk = B*K;
z = zeros(4);
A1 = [AminusBk Bk; z AminusLC];
B1 = [0 0; 39335 2624000; -13.93 -205.26; -0.9711 -75.6423; 0 0; 0 0; 0 0; 0 0];
C1 = [-0.0066667 0 -8 8 0 0 0 0; 0.0001 -0.0001 0.12 -3.12 0 0 0 0];
D1 = D;
eigA1 = eig(A1);
G3 = ss(A1, B1, C1, D1);
[y3, t3, x3] = step(G3,4);
figure(4); clf;
subplot(2,1,1); plot(t2,y2(:,:,2)); hold on; grid on; ylabel('Amplitude');
xlabel('Time(Sec)');
title('step response-state feedback');
subplot(2,1,2); plot(t3,y3(:,:,2)); grid on; ylabel('Amplitude');
xlabel('Time(Sec)');
title('step response-observer with feedback');

%Observer with feedback step response - observe state error
A2 = AminusLC;
B2 = [1; 0; 0; 0];
C2 = eye(4);
D2 = [0 0 0 0]';
G4 = ss(A2, B2, C2, D2);
[y4, t4, x4] = step(G4,1);
figure(5); clf;
plot(t4,y4(:,2),'--'); hold on;
plot(t4,y4(:,3),':','LineWidth',2); hold on;
plot(t4,y4(:,4)); hold off;
grid on; xlabel('Time(Sec)');
ylabel('State-Observation error');
title('step response - observed state error');

% Comparison of rate at which state observation error goes to zero
q_new =10*[-27 + 7i -27 - 7i -27.5 + 7i -27.5 - 7i];
L_new = place(A',C',q_new).';
```

```
AminusLC_new = A - L_new*C;
A3 = AminusLC_new;
B3 = [1; 0; 0; 0];
C3 = eye(4);
D3 = [0 0 0 0]';
G5 = ss(A3, B3, C3, D3);
[y5, t5, x5] = step(G5,0.2);
[y4, t4, x4] = step(G4,0.2);
figure(6); clf;
subplot(2,1,1);
plot(t4,y4(:,2),'--'); hold on;
plot(t4,y4(:,3),':','LineWidth',2); hold on;
plot(t4,y4(:,4)); hold off;
grid on; xlabel('Time(Sec)');
ylabel('State-Observation error');
title('observed state error (obs poles 3x far -ve)');
subplot(2,1,2);
plot(t5,y5(:,2),'--'); hold on;
plot(t5,y5(:,3),':','LineWidth',2); hold on;
plot(t5,y5(:,4)); hold off;
grid on; xlabel('Time(Sec)');
ylabel('State-Observation error');
title('observed state error (obs poles 10x far -ve)');

%LQR Controller design
R=diag([1, 1]);
Q=C'*C;
[K_lqr]=lqr(A,B,Q,R);
AminusBk_lqr = A - B*K_lqr;
F_new = -inv(C*(inv(AminusBk_lqr))*B);
Bf_new=B*F_new;
G6 = ss(AminusBk_lqr, Bf_new, C, D);
eig_A3 = eig(G6);
p_lqr = [-15.53 + 1.57i -15.53 - 1.57i -2.49 + 1.82i -2.49 - 1.82i];
q_new = 5*p_lqr;
L_new = place(A',C',q_new).';
AminusLC_new = A - L_new*C;
Bk_new = B*K_lqr;
z = zeros(4);
A4 = [AminusBk_lqr Bk_new; z AminusLC_new];
B4 = [0 0; -72714 105410; -1 -3.4139*exp(-5); 2.1964 -3.1622; 0 0; 0 0; 0 0; 0 0];
C4 = [-0.0066667 0 -8 8 0 0 0 0; 0.0001 -0.0001 0.12 -3.12 0 0 0 0];
D4 = D;
G7 = ss(A4, B4, C4, D4);
[y7, t7, x7] = step(G7,4);
figure(7); clf;
subplot(2,1,1); plot(t3,y3(:,:,2)); hold on; grid on; ylabel('Amplitude');
xlabel('Time(Sec)');
title('Step response for Manual Pole placement');
subplot(2,1,2); plot(t7,y7(:,:,2)); grid on; ylabel('Amplitude');
xlabel('Time(Sec)');
title('LQR Controller with observer');
```

References

1. Mathworks Inc., *Plotting System Responses*, 2018, Accessed from https://www.mathworks.com/help/control/examples/plotting-system-responses.html

2. S. Humbert, *Lecture Notes in Linear Control Systems*, Fall 2018, University of Colorado, Boulder.

3. W. Brogan, *Modern Control Theory* Prentice-Hall, 3rd Edition, 1991.

4. T. Kailath, *Linear Systems* Prentice-Hall, 1980.

5. J. P. How, E. Frazzoli, *Lecture Notes in Feedback Control Systems*, Fall 2010, MIT OCW.