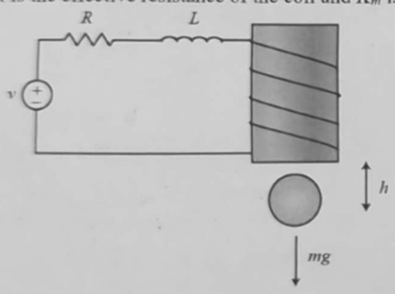
d. **Programming in scripts.** Instead of executing the commands line by line in the command window, you can compose an entire instruction set into a script file (.m file) to save, edit and run. Create a new script by clicking on 'New Script'. The commands `clc` will clear the command window and `clear all` will clear all variables in the workspace.

e. **Graphical plotting.** MATLAB's plot command creates linear x-y plots. If x and y are vectors of the same length, the command

```
plot(x,y)
```

plots the values in **y** against the values in **x**.

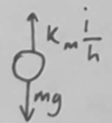
3. Let's use MATLAB to design controllers and observers using state-space methods. Consider the classical problem of magnetic suspension. Here as shown in the figure below, a voltage **[INPUT]** is created which produces the current through the coils induces a magnetic force which can balance the force of gravity and cause the ball (made of ferromagnetic material) to be suspended in midair. You may assume that the magnetic force exerted by the coil on the sphere is $f_m = K_m \frac{i^2}{h}$, where i is the current flowing through the electrical circuit (and in the coil). In addition, h is the vertical position of the ball **[OUTPUT]** as measured from the coil's surface to the centre of the sphere, v is the applied voltage, m is the mass of the ball, g is gravity, L is the effective inductance of the coil, R is the effective resistance of the coil and K_m is the magnetic force constant.



m	0.05 kg
K_m	0.0001
L	0.01 H
R	1 Ω
g	9.81 ms^{-2}

4. **[Modelling]** Construct the Free Body Diagram for the sphere and the forces acting on the sphere. This will allow you to derive the mechanical EOM of the system.

$$m\ddot{h} = mg - f_m = mg - K_m \frac{i}{h}$$



5. **[Modelling]** Use Kirchhoff's Voltage law to derive the mathematical model for the electrical component of the system.

$$v = L\frac{di}{dt} + iR$$

6. **[Modelling]** As the mechanical system is non-linear, we need to linearize it. We would like to operate the system about the desired equilibrium position of h=**0.01m**.

$$mg = K_m \frac{i^2}{h}$$

$$i = \sqrt{\frac{mgh}{K_m}}$$

$$= \sqrt{\frac{0.05 \times 9.81 \times 0.01}{0.0001}}$$

$$= 7A \qquad \bar{i} = 7A, \ \bar{h} = 0.01 m$$

$$f_m(i,h) = K_m \frac{i^2}{h}$$

$$\approx f_m(\bar{i},\bar{h}) + \frac{\partial f_m(i,h)}{\partial i}\Big|_{\substack{i=\bar{i}\\ h=\bar{h}}}(i-\bar{i}) + \frac{\partial f_m(i,h)}{\partial h}\Big|_{\substack{i=\bar{i}\\ h=\bar{h}}}(h-\bar{h})$$

$$= K_m \frac{\bar{i}^2}{\bar{h}} + \frac{2K_m i}{h}\Big|_{\substack{i=\bar{i}\\ h=\bar{h}}}\tilde{i} - K_m \frac{i^2}{h^2}\Big|_{\substack{i=\bar{i}\\ h=\bar{h}}}\tilde{h}$$

$$m\ddot{\tilde{h}} = mg - K_m\frac{\bar{i}^2}{\bar{h}} - \frac{2K_m i}{h}\Big|_{\substack{i=7\\ h=0.01}}\tilde{i} + K_m\frac{i^2}{h^2}\Big|_{\substack{i=7\\ h=0.01}}\tilde{h}$$

$$= -0.14\tilde{i} + 49\tilde{h}$$

$$\ddot{\tilde{h}} = -2.8\tilde{i} + 980\tilde{h}$$

$$\frac{d(\bar{i}+\tilde{i})}{dt} = -100(\bar{i}+\tilde{i}) + 100(\bar{v}+\tilde{v})$$

$$d\tilde{i} = -100\bar{i} - 100\tilde{i} + 100\bar{v} + 100\tilde{v}$$

$$= -100\tilde{i} + 100\tilde{v}$$

7. **[Modelling]** Let's try to derive the state-space representation of the linearized electro-mechanical system.
   a. How many states are there in this system? Identify all of them?

3 States

$x = \begin{bmatrix} \tilde{h} \\ \dot{\tilde{h}} \\ \tilde{i} \end{bmatrix}$

   b. Define **x** as your state vector that contain all states. Can you express the system in the form: $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$, $y = \mathbf{C}\mathbf{x}$, where u and y are the input and output respectively. What are the numerical values for the matrices **A**, **B** and **C**?

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 980 & 0 & -2.8 \\ 0 & 0 & -100 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 100 \end{bmatrix} u$$

$$y = [1\ 0\ 0]x$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 980 & 0 & -2.8 \\ 0 & 0 & -100 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 100 \end{bmatrix} \quad C = [1\ 0\ 0]$$

   c. Construct a state-space model of the system in MATLAB. What is the full command you used to enter the system in MATLAB?

```
sys1 = ss(A,B,C,0)
```

8. **[Stability]** First thing we want to analyze is the stability of the open-loop system (without control input).
   a. What do you need to compute? Use MATLAB to assist you in this.

Eigenvalues of A

31.31, -31.31, -100

   b. Where are the poles of the system?

31.31, -31.31, -100

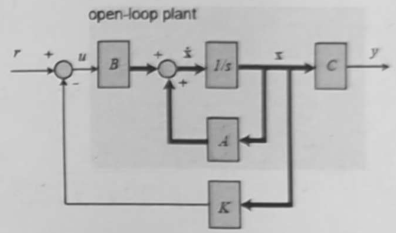
   c. Is the system stable? Why?

No.

   d. Use MATLAB to simulate the dynamic response of the system to a non-zero initial condition. Describe the response. What MATLAB commands did you use? Append the time response of the system to the back of this document for submission.

```
initial(sys, [1 1 1])
```

9. **[Controller Design]** We wouldlike to design a state feedback controller to stabilize and control the system.



   a. First we need to check if we could actually do this. We need to check complete state controllability of the system. Use MATLAB to determine the controllability matrix and the rank of the matrix.

   b. Is the system completely state controllable? Why?

Yes. Rank is 3

   c. Let's design a state feedback controller such that the 2 dominant poles are at $-10 \pm j10$ and the third pole at -50. Use MATLAB to help assist you in computing the required gain matrix **K**. What is **K** and what MATLAB commands did you use?

~~K=place(A,B,[a b c])~~

~~= -775.7143 -206.429 0.8000~~

~~sys_cl = ss(A-B*K,[0;0;0],C,0)~~

~~initial(sys_cl,[0.01 0.01 0.01])~~

```
p1 = -10+10j;
p2 = -10-10j;
p3 = -50;
K=place(A,B,[p1 p2 p3])
```

k = -280.7143 -7.7857 -0.3000

d. Now again, simulate the closed-loop system (using the **K** computed above) to a non-zero initial condition (all initial states are at 0.01). What MATLAB commands did you use? Append the time response of the **all states** of the system to the back of this document for submission. Describe the responses of all states of the system.

```
sys_cl = ss (A-B*K , [0;0;0] , C , 0)
initial = (sys_cl , [0.01 0.01 , 0.01])
```

e. You are not happy with the controller performance. Could you reduce the settling time further? What are the new **K** gains for this improved controller? What MATLAB commands did you use? Append the time response of all states of the system to the back of this document for submission.

```
p1 = -20+20i;
p2 = -20 - 20i;
p3 = -100;
K = place (A, B, [p1 p2 , p3])
sys_cl = ss (A-B*K, [0; 0;0], C, 0)
initial = (sys_cl , [0.01 , 0.01 0.01])
```

10. [**Servo Design**] We would now like to implement a controller so it can track a desired reference signal with zero steady state error. This will require a servo controller so that an integral action can be used to ensure that there is no steady state error.

a. What is the system type of linearized modelled system in 7(b)?

Type 0

b. Does the system contain any inbuilt integrators?

~~yes~~ No

c. What is the control law for this system so that it would be able to track a reference step input with zero steady state error?

$$u = -kx - k_I \xi$$
$$\xi = r - y = r - Cx$$

d. Now we need to check if we could actually do this. We need to check the complete state controllability of the servo control system. Use MATLAB to determine the controllability matrix for the servo control system and the rank of the matrix.

~~Ank~~

```
C. = [0    0      -280     28000 ;
      0    -280   28000    -3079000 ;
      100  -10000 10000000 -1000000000 ;
      0    0      0        280]
```

e. Is the system completely state controllable and a servo controller can be designed? Why?

Yes. C_o is ~~full~~ rank 4

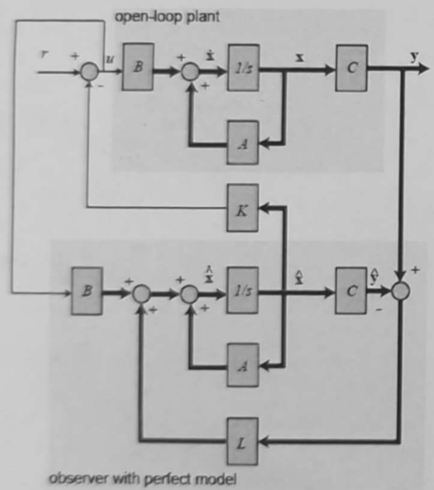
f. Let's design a servo controller such that the 2 dominant poles are at $-10 \pm j10$ and the third and fourth poles at -50 and -40. Use MATLAB to help assist you in computing the required gain matrix **Ks**. What is **Ks** and what MATLAB commands did you use?

```
p1 = -10 + 10i
p2 = -10 - 10i
p3 = -50
p4 = -40
Ks = acker (A_h, B_h, [-10+10j, -10-10j, -50 -40])
```

g. Write down the expression for the servo error dynamics and use MATLAB to simulate the system to a non-zero initial condition (all initial states are at 0.01) for the servo error system. Show, using MATLAB graphical plots, that <u>all</u> states of this system (error for individual states) with the computed gain from above will tend to zero as t goes to infinity. Append the time response of the system to the back of this document for submission.

h. Now use MATLAB to show that in the servo control configuration with the gains computed above, a unit step reference input $r(t)$ will result with the system output following the reference input with zero steady state error. Show all states of the system. What MATLAB commands did you use? Append the time response of the system to the back of this document for submission.

11. **[Observer Design]** When we can't measure all the states, we can build an observer to estimate them. The observer, as we discovered is basically a copy of the original plant and uses the actual output to correct the estimate.



a. Again we need to check if we could actually do this. We need to check complete observability of the system. Use MATLAB to determine the observability matrix and the rank of the matrix.

b. Is the system completely observable? Why?

Yes. Observability matrix is rank 3

c. Let's design an observer such that the observer dynamics are much faster than the system itself. Let's place them at -100, -101 and -102. Use MATLAB to help assist you in computing the required gain matrix $\mathbf{K_e}$. What is $\mathbf{K_e}$ and what MATLAB commands did you use?

```
p1 = -100;
p2 = -101;
p3 = -102;
ke = place(A', C', [p1 p2 p3])'
```

$K_e = 1.0e+04 * \begin{matrix} 0.0202 \\ 1.1282 \\ 0 \end{matrix}$

d. Write down the expression for the observer error dynamics and use MATLAB to simulate the system to a non-zero initial condition (all initial states are at 0.01) for the observer error system. Show, using MATLAB graphical plots, that all states of this system (error for individual states) with an observer gain will tend to zero as t goes to infinity. What MATLAB commands did you use? Append the time response of the system to the back of this document for submission.

```
sys_ob = ss(A-ke*C, B, eye(3), 0)
initial(sys_ob, [0.01 0.01 0.01])
```

12. **[Observed State Feedback Control Design]** Let's now use the observer estimation for state feedback control. Here we will use observer gains computed in 11(c) and feedback gains in 9(c).

a. What is the augmented state-space model that describes both the dynamics of the observed-state feedback control system and the observer error dynamics? Write the expression down. What is the dimension of this augmented state vector?

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A-BK & BK \\ 0 & A-K_eC \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix}$$

dimension is 6

b. Can you check if the observer-based controller is stable?

c. With the expression in 12(a), simulate the augmented system to a non-zero initial condition (all initial states are 0.01). Show that all states and errors approach zero as t goes to infinity. What MATLAB commands did you use? Append the time response of the system to the back of this document for submission.

13. **[Minimum-Order Observed State Feedback Control Design]** We do possess a high accuracy laser sensor that allows us to measure accurately the position of the spherical ball **only**. We will continue to use the feedback control gains computed in 9(c).

a. Construct a minimum-order observer such that it estimates the remaining states. What is the error equation for the minimum-order observer?

$$\dot{e} = (A_{bb} - K_e A_{ab}) e$$

$$A_{aa} = 0 \quad A_{ab} = [1\ 0], \quad A_{ba} = \begin{bmatrix} 980 \\ 0 \end{bmatrix}, \quad A_{bb} = \begin{bmatrix} 0 & -2.8 \\ 0 & -100 \end{bmatrix}, \quad B_a = 0 \quad B_b = \begin{bmatrix} 0 \\ 100 \end{bmatrix}$$

b. Before proceeding with the design, can you verify that a minimum-order observer can be constructed?

rank is 2, system can be constructed

c. Let's design a minimum-order observer such that the observer dynamics are much faster than the system itself. Let's place them at -100 and -101. Use MATLAB to help assist you in computing the required gain matrix K_e. What is K_e and what MATLAB commands did you use?

```
Kem = place(Abb', Aab', [p1 p2])'
```

$$K_{em} = \begin{matrix} 101 \\ 0 \end{matrix}$$

d. Write down the augmented state-space equation that describes the state dynamics of the system under minimum-order observer state feedback and the error dynamics of the minimum-order observer. What is the dimension of the augmented state space vector?

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A - BK & BK_b \\ 0 & A_{bb} - K_e A_{ab} \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix}$$

dimension is 5

e. Can you check if the minimum-order observer-based controller is stable?

f. With the expression in 13(d), simulate the augmented system to a non-zero initial conditions (all initial values are 0.01). Show that all states and errors approach zero as t goes to infinity. What MATLAB commands did you use? Append the time response of the system to the back of this document for submission.

```
sys_mob_cl = ss([A-B*K B*Kb; [000;000] Abb-Kem*Aab]; [0;0;0;0;0], eye(5), 0)
initial(sys_mob_cl, [0.01, 0.01 0.01 0.01 0.01])
```

14. **[Linear Quadratic Regulator Design]** As a comparison, we would like to try to design an optimal controller for the magnetic suspension system.

a. Write down the performance index for the LQR problem.

$$J = \int_0^\infty (x^* Q x + u^* R u)\, dt$$

b. If you would like to place equal emphasis on position error of the sphere (only) and control energy, what would be a possible value for your **Q** and **R** matrix to be?

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad R = 1$$

c. Use MATLAB to compute the optimal controller gain. What is **K** and what MATLAB commands did you use?

Q = [1 0 0; 0 0 0; 0 0 0]
R = 1
K1_lqr = lqr(A, B, Q, R)

K
K1_lqr = -914.1363 -29.3607 0.6261

d. If you would like to place much more emphasis (say at least 10^5 times) on position error of the sphere (only) than control energy, what would be a possible value for your **Q** and **R** matrix to be?

$$Q = \begin{bmatrix} 10^5 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad R = 1$$

e. Use MATLAB to compute the optimal controller gain. What is **K** and what MATLAB commands did you use?

Q = [10^5 0 0 ; 0 0 0; 0 0 0]
K2_lqr = lqr(A, B, Q, R)

K
K2_lqr = 1.0e+03 * -1.0580 -0.0323 0.0007

f. Now again, simulate the closed-loop system (using both of the optimal **K** computed individually above) to a non-zero initial conditions (all initial states are 0.01). What MATLAB commands did you use? Append the time response of the systems to the back of this document for submission. Can you comment on the difference in the responses?

sys1 = ss(A - B*K1_lqr, [0; 0; 0], C, 0)
initial(sys1, [0.01 0.01 0.01])
sys2 = ss(A - B*K2_lqr [0; 0; 0], C, 0)
initial(sys2, [0.01 0.01 0.01])

Aplitude of graph with K2_lqr has a shorter time to 0 than Amplitude of graph with K1_lqr
this is due to a greater ephasis of position error of sphere.