Bangladesh University of Engineering and Technology

EEE 318: Control System Laboratory

Project Report

Team Members:

1606099- Sohan Salahuddin Mugdhi

1606100- A.S.M Sadman Sakib

1606101-Ruhul Amin Pranto

1606103- Laisur Rahman

1606104- Tasrif Adnan

1606119- Md. Sabit Abdal

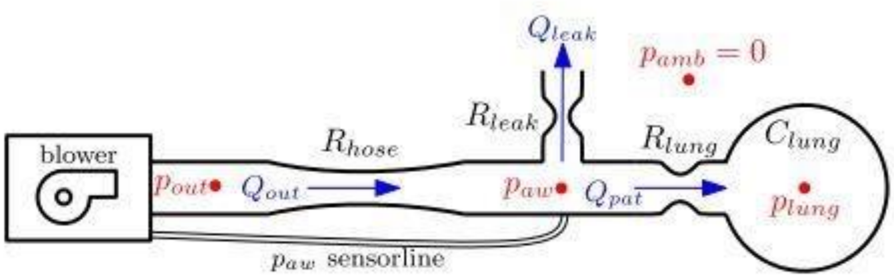
Section: B2

Level/Term: 3/2

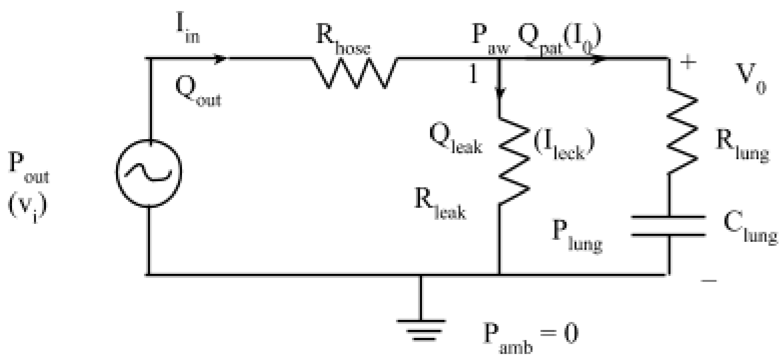
Date of Submission: 22/12/2020

Task 1:

Determine the transfer function of the respiratory system shown in Fig. 3



Equivalent Electrical Circuit



-

P ⇒ Voltage

Q ⇒ Current

C⇒ Capacitance

R ⇒ resistance

Q_{out} = Pout-PawRhose

Q_{leak} = PawRleak

Q_{pat} = Q_{out} – Q_{leak} = Paw-PlungRlung ....(i)

dPlungdt=Plung=QpatClung=Paw-PlungClungRlung ....(ii)

Applying KCL at node 1

Pout-PawRhose=PawRleak+Paw-PlungRlung

⇒ PoutRhose-PawRhose=PawRleak+PawRlung-PlungRlung

⇒ PawRhose+PawRleak+PawRlung=PoutRhose+PlungRlung

⇒ P_{aw} = PoutRhose+PlungRlung1Rhose+1Rleak+1Rlung ....(iii)

⇒ P_{aw} = 1Rlung1Rhose+1Rleak+1Rlung P_{lung+1}Rhose$_1$Rhose$_{+1Rleak+1Rlung}$ P_{out} ....(iv)

Substituting the value of P_{aw} in eqn (i) & (ii)

Q_{pat} = 1RlungPlung+1RhosePoutRlung1Rlung+1Rhose+1Rleak – 1Rlung P_{lung}

= 1Rlung-1Rlung+1Rhose+1RleakRlung1Rlung+1Rhose+1RleakP_{lung} + 1RhoseRlung1Rlung+1Rhose+1RleakP_{aw}

Q_{pat} = -1Rhose+1RleakRlung1Rlung+1Rhose+1RleakP_{lung} + 1RhoseRlung1Rlung+1Rhose+1Rleak P_{out} ....(v)

P_{lung} = 1RlungPlung+1RhosePoutClungRlung1Rlung+1Rhose+1Rleak – PlungClungRlung

= 1Rlung-1Rlung+1Rhose+1RleakClungRlung1Rlung+1Rhose+1Rleak P_{lung} + 1RhoseClungRlung1Rlung+1Rhose+1Rleak P_{out}

P_{lung} = - 1Rhose+1RleakClungRlung1Rlung+1Rhose+1Rleak P_{lung} + 1RhoseClungRlung1Rlung+1Rhose+1Rleak P_{out} ....(vi)

Here, input ⇒ P_{out} , output ⇒Paw Qpat , state ⇒ P_{lung}

From eqn (iv) to (vi) we find:

P_{lung}= A_h P_{lung} + B_h P_{out} ....(vii)

Paw Qpat = C_h P_{lung} + D_h P_{out} ....(viii)

Where,

A_h = -1Rhose+1RleakClungRlung1Rlung+1Rhose+1Rleak

B_h = 1RhoseClungRlung1Rlung+1Rhose+1Rleak

C_h = 1Rleak1Rlung+1Rhose+1Rleak -1Rhose+1RleakRlung1Rlung+1Rhose+1Rleak

D_h =1Rhose1Rhose+1Rleak+1Rlung -1RhoseRlung1Rlung+1Rhose+1Rleak

We know that,

Y(s) = Cn sI-AhBh-1+Dhu(s)

where Y(s) is the output matrix and u(s) is the input matrix

so, Y(s)u(s)=Hs=ChSI-Ah-1Bh+Dh

Using the values of parameters given in table 1 of the reference paper, H(s) was calculated using the code given below.

Rlung	5/1000
Clung	20
Rleak	60/1000
Rhose	4.5/1000
ω_n	2π30

Code:

```
clc;
clearvars;
Rlung = 5/1000;
Clung = 20;
Rleak = 60/1000;
Rhose = 4.5/1000;

Ah = -((1/Rhose)+(1/Rleak))/(Rlung*Clung*((1/Rlung)+(1/Rhose)+(1/Rleak)))
Bh = (1/Rhose)/(Rlung*Clung*((1/Rlung)+(1/Rhose)+(1/Rleak)))
Ch = [(1/Rlung)/((1/Rlung)+(1/Rhose)+(1/Rleak));...
    -((1/Rhose)+(1/Rleak))/(Rlung*((1/Rlung)+(1/Rhose)+(1/Rleak)))]
Dh = [(1/Rhose)/((1/Rlung)+(1/Rhose)+(1/Rleak));...
    (1/Rhose)/(Rlung*((1/Rlung)+(1/Rhose)+(1/Rleak)))]
```

s = tf('s');
Hs = Ch*inv((s*eye(1) - Ah))*Bh + Dh

So, the calculated H(s) is:

$$H(s) = \begin{bmatrix} \frac{0.5063s + 5.063}{s + 5.443} \\ \frac{101.3s + 1.137e-13}{s+5.443} \end{bmatrix}$$

Task 2:

Determine the overall transfer function of the closed loop control system shown in Fig. 4.

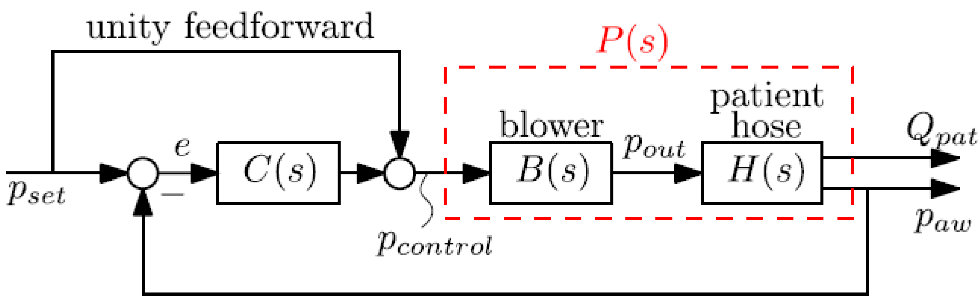
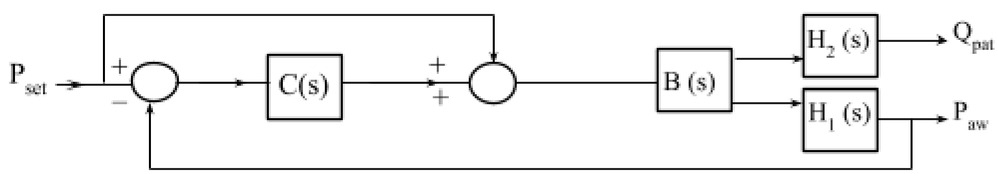


Fig. 4. Closed-loop control scheme with a linear controller $C(s)$.

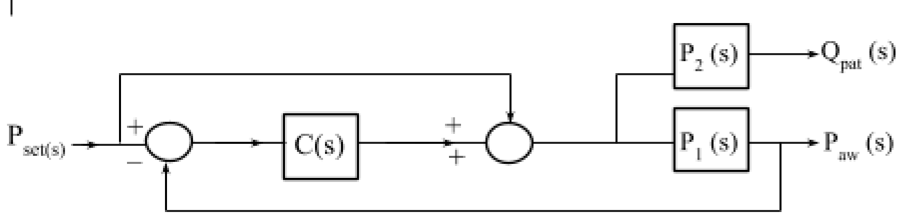
Changing the diagram to:



Let

C(s) =

BsH1s=P1(s)



BsH2s=P2(s)

So,

Now,

Paws=P1sPsets+C(s)(Psets-Paw(s))

= P1sPset(s)+P1sCsPset(s)-P1sCsPaw(s)

⇒Paw(s)+P1sCsPaw(s)=Pset(s)(P1(s)+P1(s)C(s))

⇒ Paw(s)Pset(s)=P1s+CsP1(s)1+CsP1(s)=T1(s) .... (1)

Again,

Q_{pat} (s)= P2sPsets+C(s)(Psets-Paw(s)

= P2sPsets+P2sCsPsets-P2sCsPaw(s)

= P2sPsets+P2sCsPset s-P2sCsPsetsT1(s)

= PsetsP2s+P2sCs-P2sCsT1(s)

⇒ Qpat(s)Pset(s)=P2s1+Cs(1-T1(s))

=P2s1+Cs1+CsP1s-P1s-CsP1(s)1+CsP1(s)

=P2s1+Cs-CsP1(s)1+CsP1(s)

= P2s1+CsP1s+Cs-CsP1(s)1+CsP1(s)

Qpat(s)Pset(s)=P2s+CsP2(s)1+CsP1(s)=T2(s) .... (2)

<u>By running the following code:</u>

```
clc;
clearvars;

ki = 1;
wn = 2*3.1416*30;
zeta = 1;
s = tf('s');
Hs = [(0.5063*s + 5.063)/(s + 5.443);...
    (101.3*s + 1.137e-13)/(s + 5.443)];
Bs = (wn^2)/(s^2+(2*zeta*wn*s)+wn^2);
Ps = Hs.*Bs;
Cs = ki/s;
Closed_Paw = (Ps(1)+(Cs*Ps(1)))/(1+(Cs*Ps(1)));
CLosed_Qpat = (Ps(2)+(Cs*Ps(2)))/(1+(Cs*Ps(1)));
```

We found T_1(s) to be:

```
Closed_Paw =

  1.799e04 s^9 + 1.396e07 s^8 + 4.135e09 s^7 + 5.68e11 s^6 + 3.387e13 s^5 + 5.755e14 s^4

                                              + 3.83e15 s^3 + 1.002e16 s^2 + 6.728e15 s

  -------------------------------------------------------------------------------------

  s^11 + 1147 s^10 + 5.515e05 s^9 + 1.428e08 s^8 + 2.119e10 s^7 + 1.753e12 s^6

                 + 7.044e13 s^5 + 8.957e14 s^4 + 4.759e15 s^3 + 1.052e16 s^2 + 6.728e15 s
```

And $T_2(s)$ to be:

```
Closed_Qpat =

  3.599e06 s^9 + 2.757e09 s^8 + 7.997e11 s^7 + 1.057e14 s^6 + 5.721e15 s^5 + 5.794e16 s^4

                                                + 1.869e17 s^3 + 1.346e17 s^2 + 151.1 s

  ---------------------------------------------------------------------------------------

  s^11 + 1147 s^10 + 5.515e05 s^9 + 1.428e08 s^8 + 2.119e10 s^7 + 1.753e12 s^6

                 + 7.044e13 s^5 + 8.957e14 s^4 + 4.759e15 s^3 + 1.052e16 s^2 + 6.728e15 s
```

Task 3:

Sketch the root locus of the control system shown in Fig. 4 for 0<ki< of the integral controller C(s).

Comparing equations 1 and 2 with,

Ts=KG(s)1+KG(s)H(s)P1s+CsP1(s)1+KisP1(s)=T1(s)

And,Ts=KG(s)1+KG(s)H(s)P2s+CsP2(s)1+KisP1(s)=T2(s)

We Find that, for both closed loop transfer functions,

KGsHs=KiP1(s)s

So, the root locus will be drawn for P(s)s for ki>0

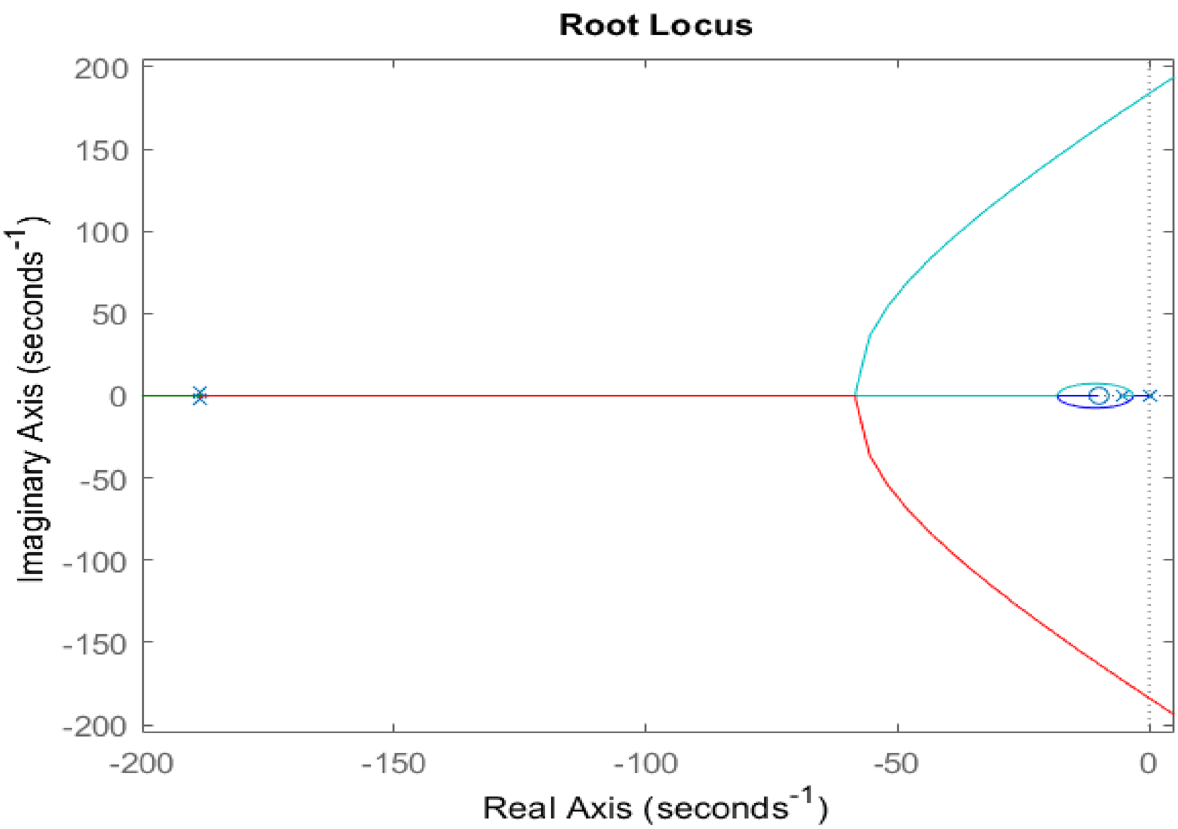
<u>Code for plotting root locus:</u>

```
clc;
clearvars;

    s = tf('s');
Ps = [(1.799e04*s+1.799e05)/(s^3+382.4*s^2+3.758e04*s+1.934e05);...
   (3.599e06*s+4.04e-09)/(s^3+382.4*s^2+3.758e04*s+1.934e05)];

figure(1)
rlocus(Ps(1)/s);
axis([-200 5 -205 205]);
```

Root Locus:



Task 4:

Reproduce the results shown in Fig. 7 for the combined feedback and feedforward control system. Discuss the necessity of both feedback and feedforward control.

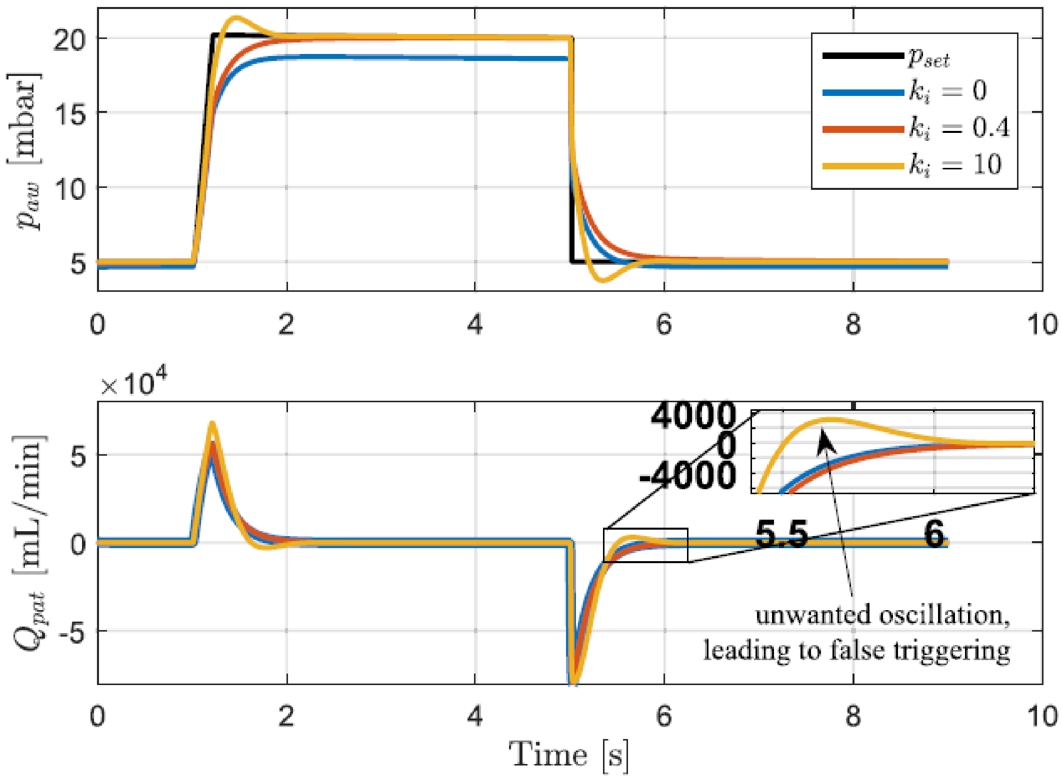
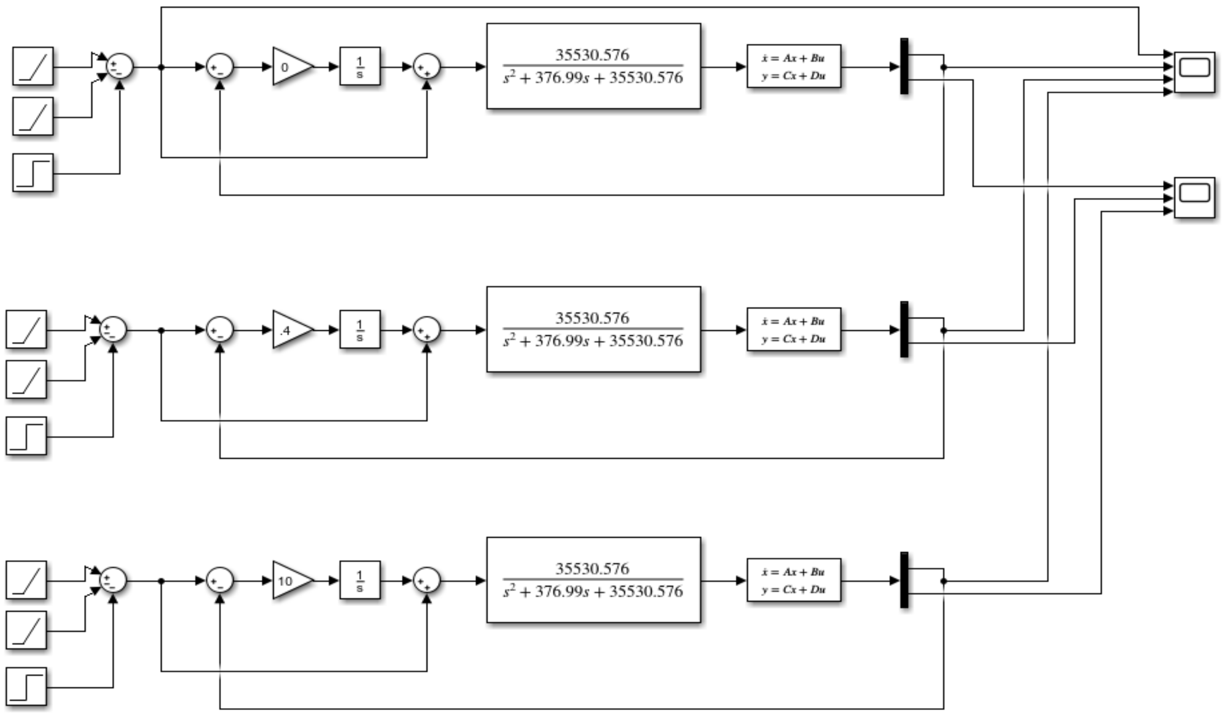
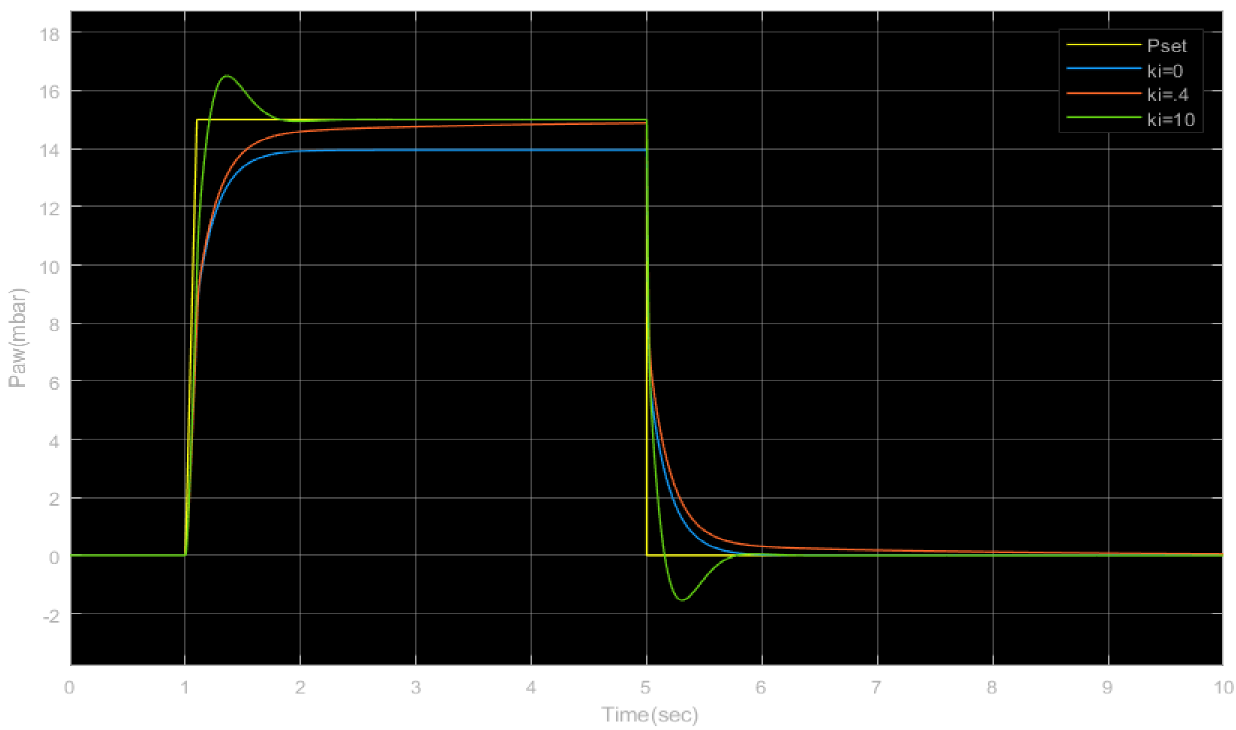


Fig. 7. Simulation result of the closed-loop system using no controller, a low-gain controller ($k_i = 0.4$), and a high-gain controller ($k_i = 10$).

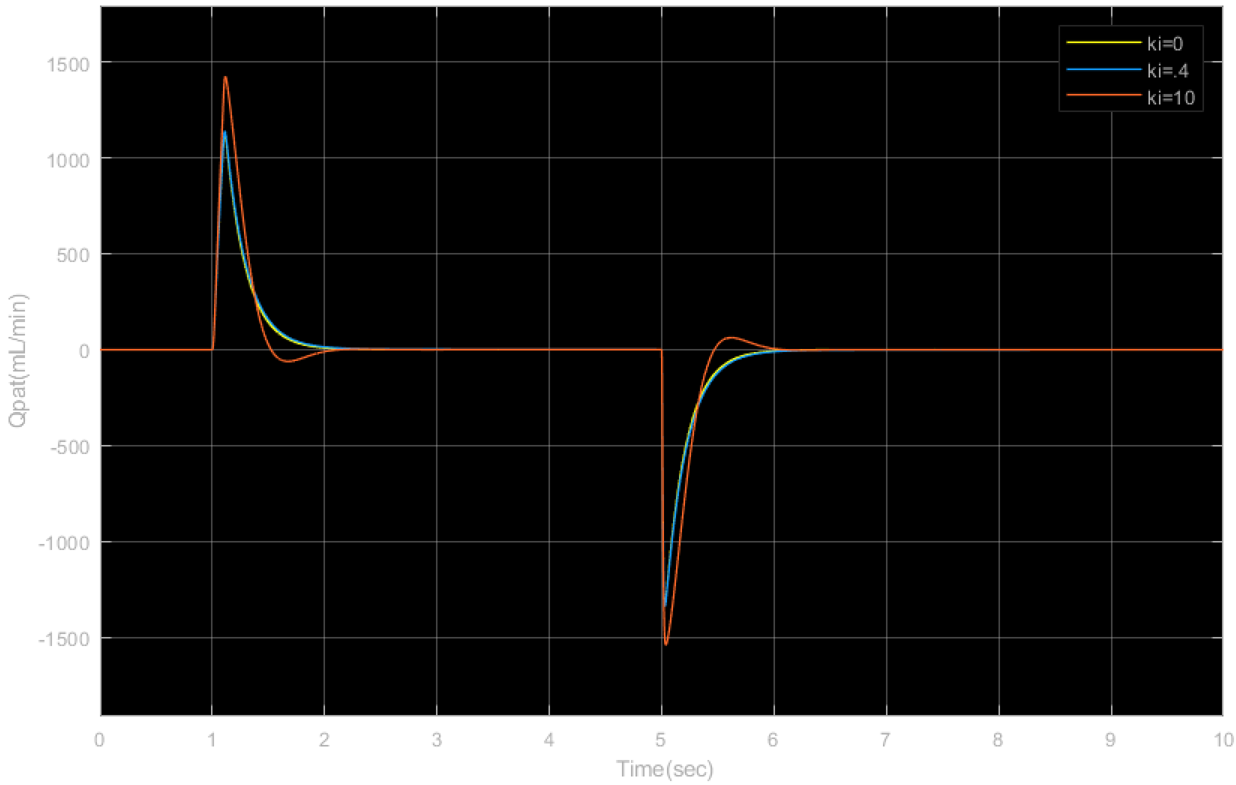
Simulink Diagram:



Plot for output Paw:



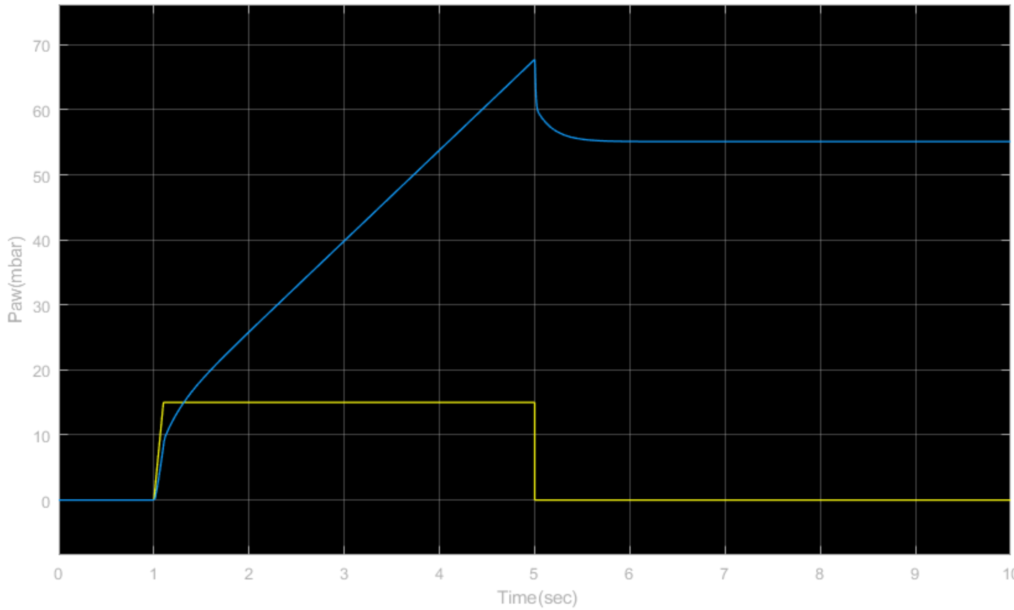
Plot for output Qpat:

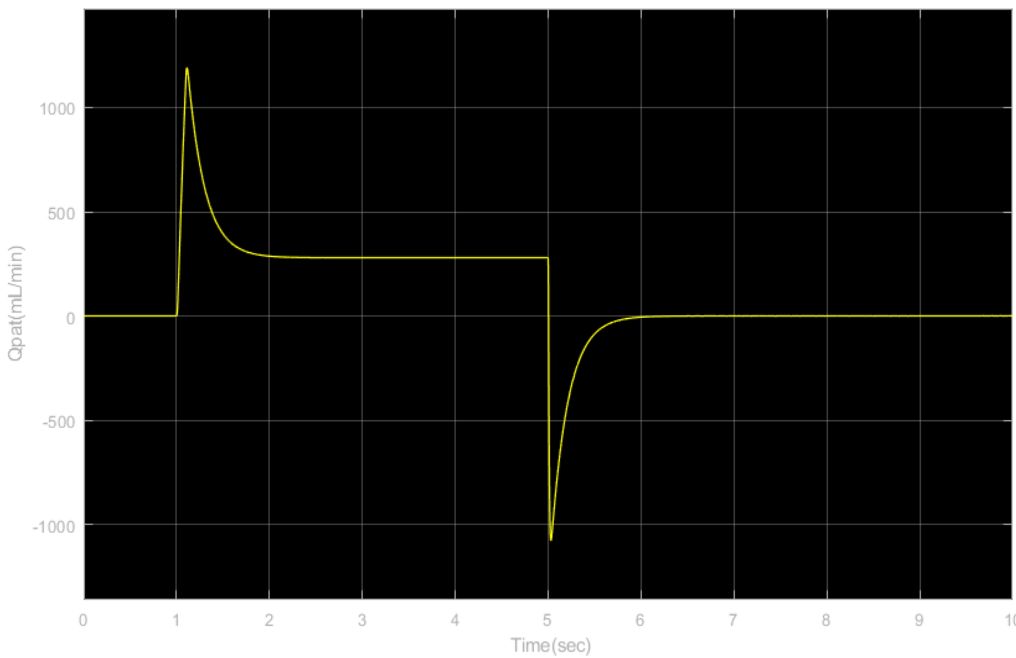


Just as in the paper, we have used ki = 0, 0.4, 10, as controller gain. From the plot, it is clear that at low values of gain, the output air-way pressure can hardly, if at all, reach the pressure set point. On the other hand, for high values of gain, the system can build up pressure fast, at the cost of producing high overshoot in airflow. Which is why designing a better controller is necessary.

<u>The importance of both feedback and feed-forward:</u> To discuss the importance of feedback and feed-forward, we first observed the outputs without feedback and without feed-forward.

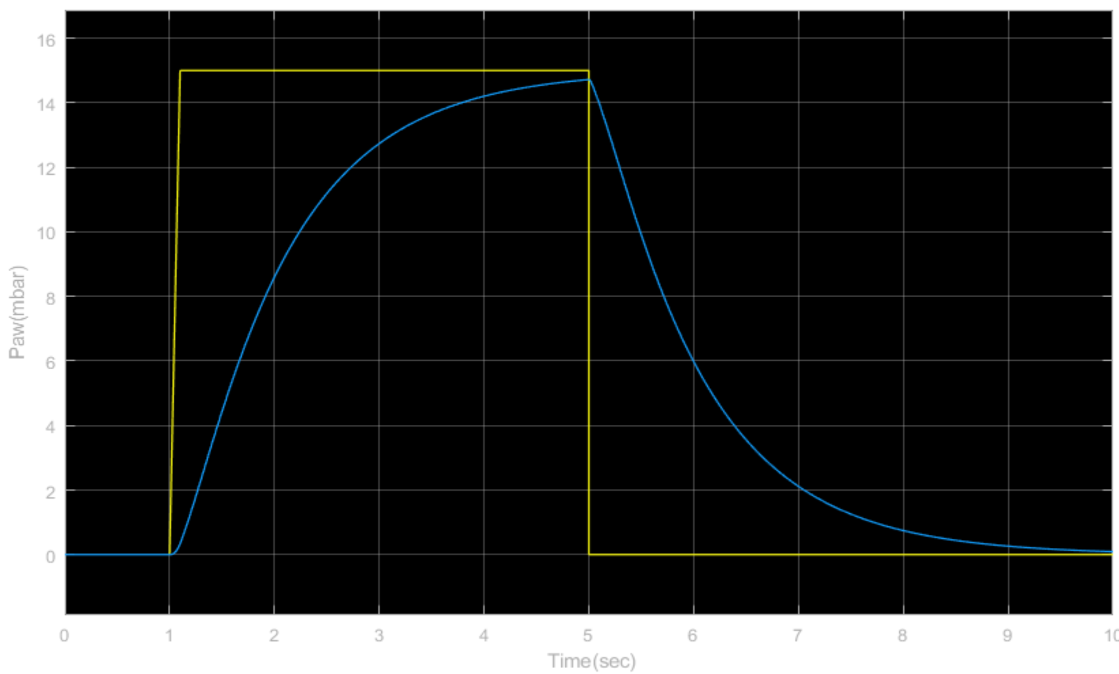
Without feedback:

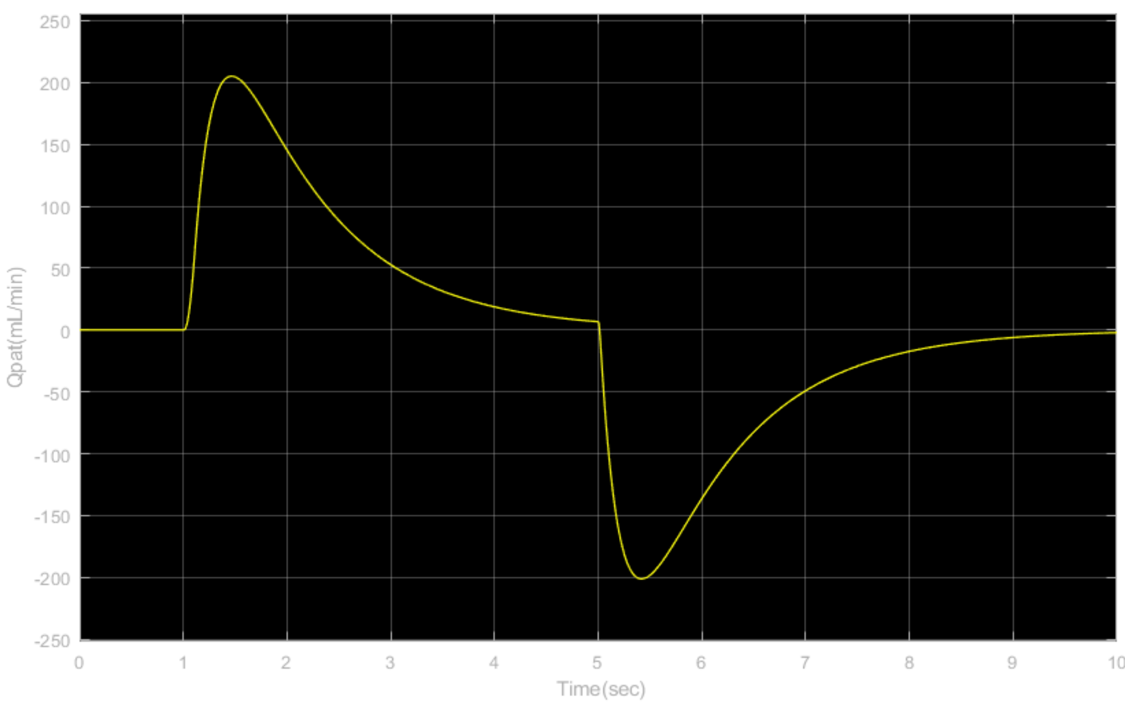




Without feedback, the system isn't aware of the error it is generating. Resulting in very large values of error for both the airway pressure and the airflow.

Without feed-forward:





It clear that without feed-forward the system fails to build up pressure quick enough to follow the set pressure, which is one of the main requirements of the system along with low error and overshoot.

So, we can see that without both feedback and feed-forward the system fail to meet the requirements by large margin. That is why both feedback and feed-forward are necessary.

Task 5:

In this design, the feedback controller is an ideal integrator. Do you prefer a PI or lag controller? Why or why not?

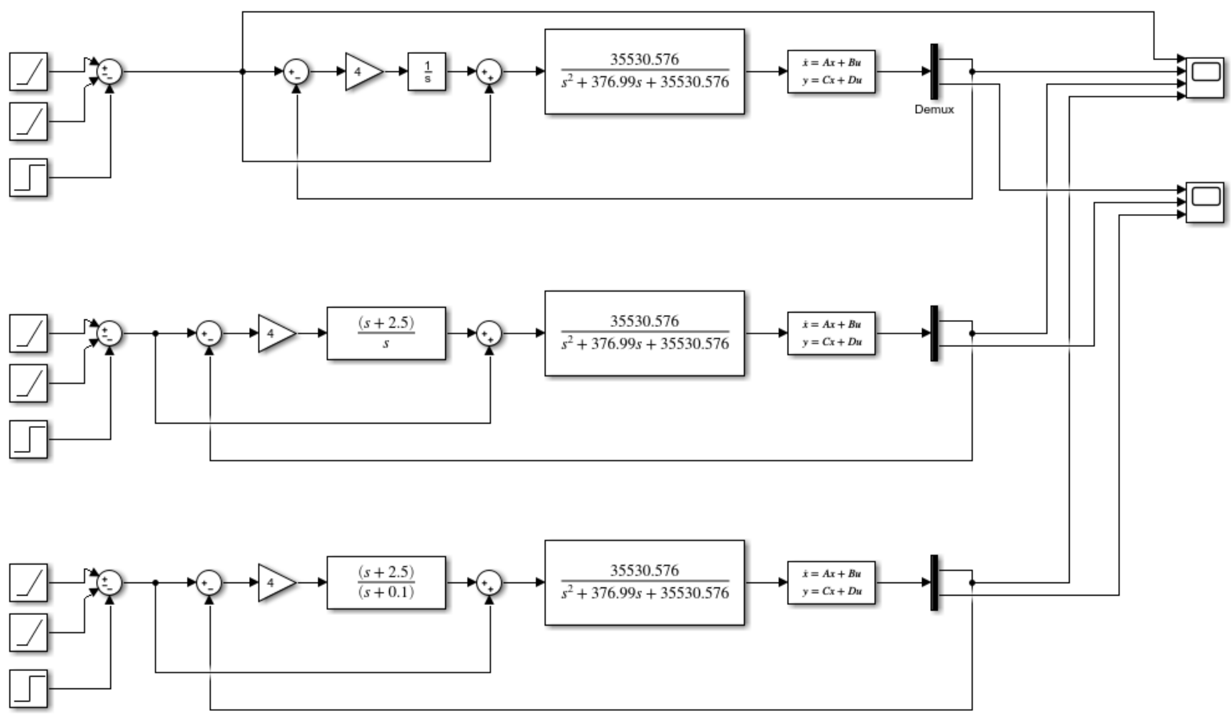
First, we observe the performance of the three types of controllers. Here, we have used,

Ideal integrator: 4/s

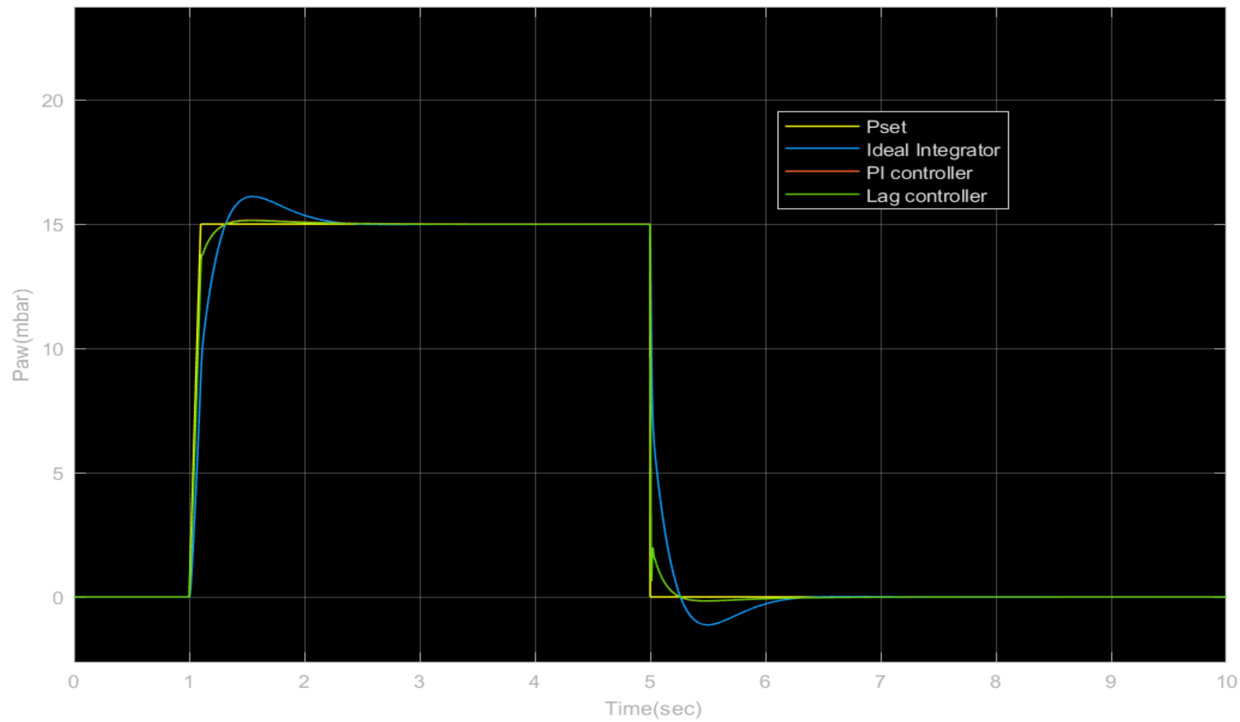
PI controller: 4*(s+2.5)/s

Lag controller: 4*(s+2.5)/(s+0.1)

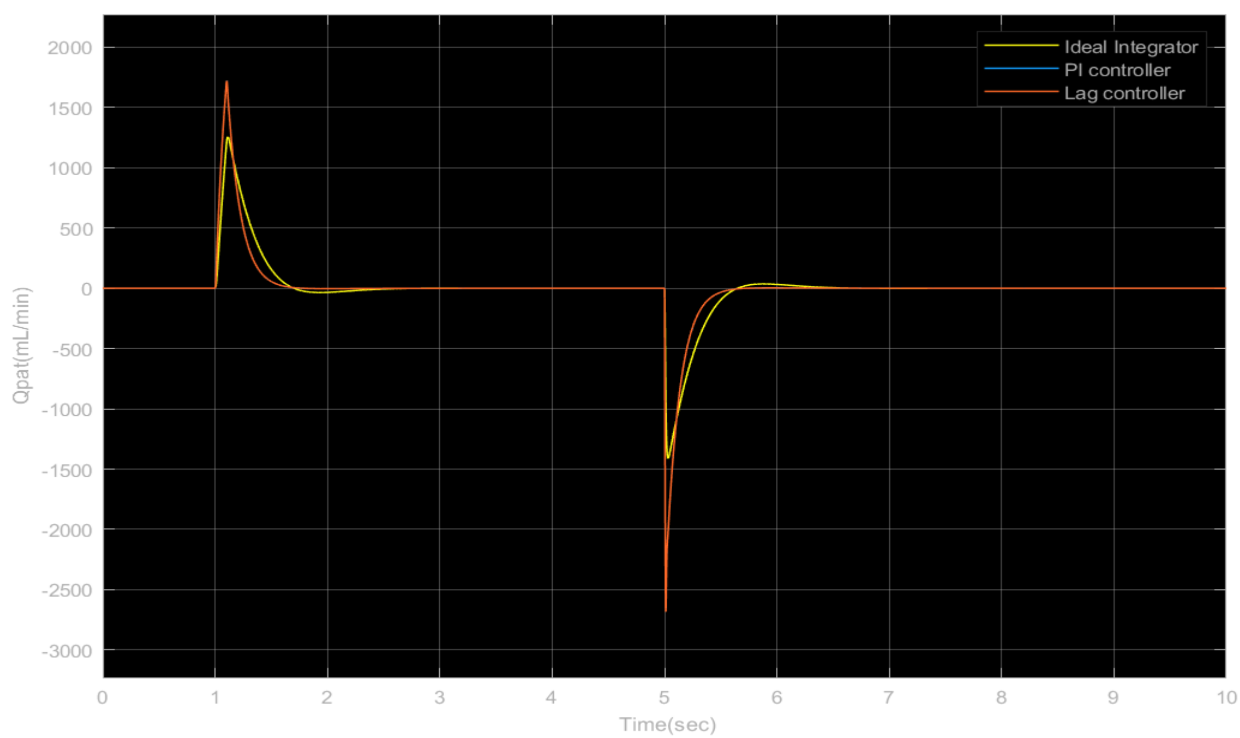
Simulink Diagram:



Plot for output Paw:



Plot for output Qpat:



From our investigations of the performance of the three controllers for different parameters such as, different gains ki, locations of zeros and poles Zc and Pc, we have found that, For different values of ki, the ideal integrator has large overshoot in comparison to PI and lag controllers, exceeding the tolerance value of overshoot which is 2mL/min.

For different values of ki, and the same Zc, rise time for PI and lag controllers are almost the same with negligible difference. When Pc of the lag controller is very close to origin, all the values of concern (overshoot, rise time, error) are almost equal for both PI and lag controller. However, as Pc is moved away from the origin, the overshoot decreases for lag controller in comparison to PI controller at the cost of increases error at plateau pressure. The rise time still remains almost equal.

Considering all these facts we chose to prefer the PI controller over the other two.

The performances of the three controllers are shown in the table below for rise time of

Pset = 80ms.

	Rise Time(ms)	Error(mbar)	Overshoot(mL/sec)
Ideal Integrator	198	0	35.96
PI controller	88	0	3.27
Lag controller	88	0.01	3.11

Task 6:

Design your preferred linear controller in order to meet the specifications stated in page 166 between column 1 and 2.

Specifications:
1.The rise time from 10% to 90% of a pressure set point should be approximately 200 ms.
2.The pressure at the end of an inspiration, the so-called plateau pressure, should be within a pressure band of ±2 mbar of the pressure set point.
3. The overshoot in the flow during an expiration should be below the triggering value set by the clinician, and a typical value is 2 L/min or 33.33 mL/s.

In the previous task, we have decided to use the PI controller in our design. First, we observed the performance of PI controller for ki = 4 and Zc = 0.5, 2.5, 4.5. The observed performance for this configuration is given in the following table.

Ki = 4	Rise Time(ms)	Error(mbar)	Overshoot(mL/sec)
Zc = 0.5	91	0	0
Zc = 2.5	88	0	3.274
Zc = 4.5	87	0	6.193

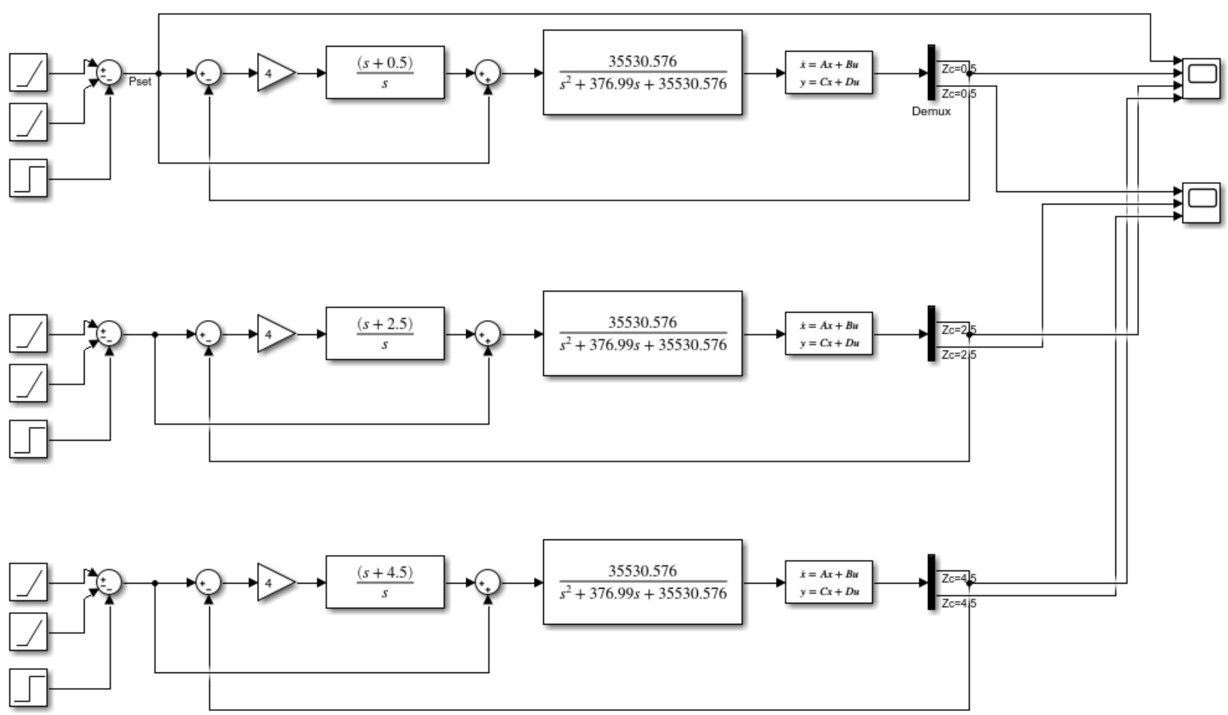
Since Zc = 0.5 gives zero overshoot and error with a significantly quick rise time, we settled on Zc = 0.5 for this controller.

Next, for Zc = 0.5, and ki = 4.5, 5, 8, the performance is given in the table below.

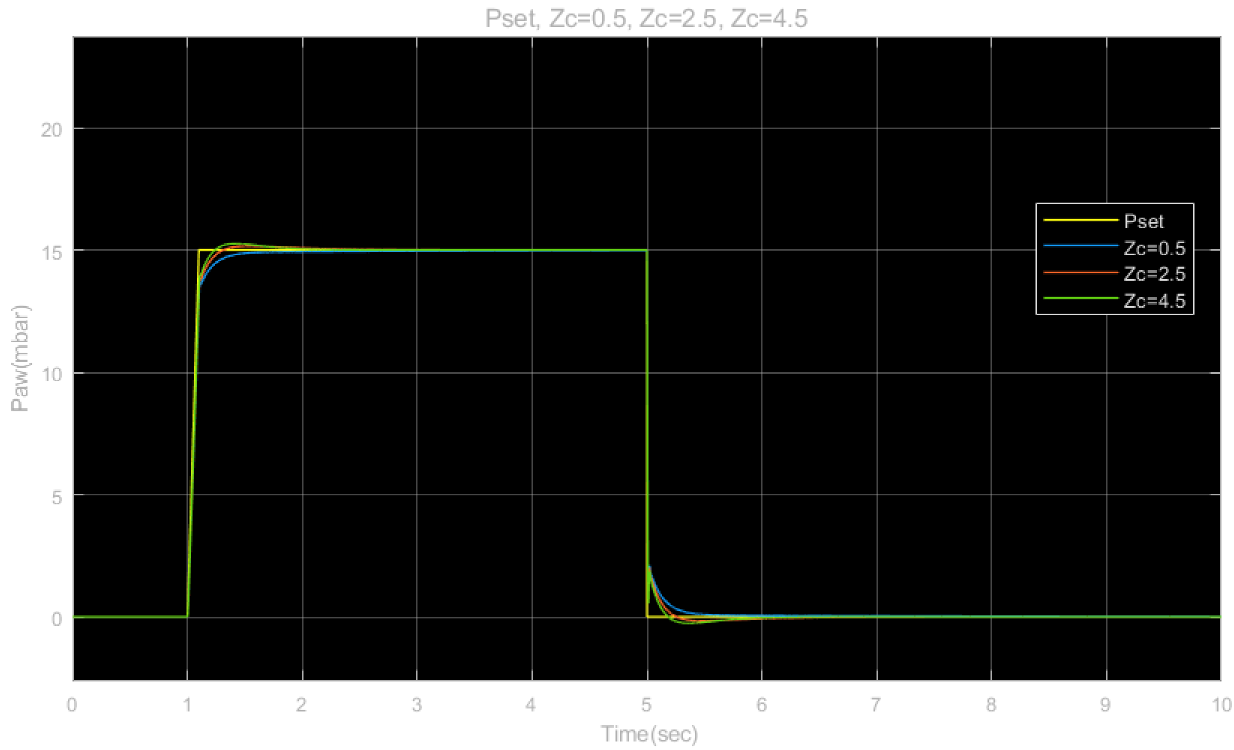
Zc = 0.5	Rise Time(ms)	Error(mbar)	Overshoot(mL/sec)
Ki = 4.5	89	0	0
Ki = 5	87	0	0
Ki = 8	85	0	0

We found the best result to be for the PI controller 8*(s+0.5)/s, with rise time = 85ms, error = 0, and overshoot = 0.

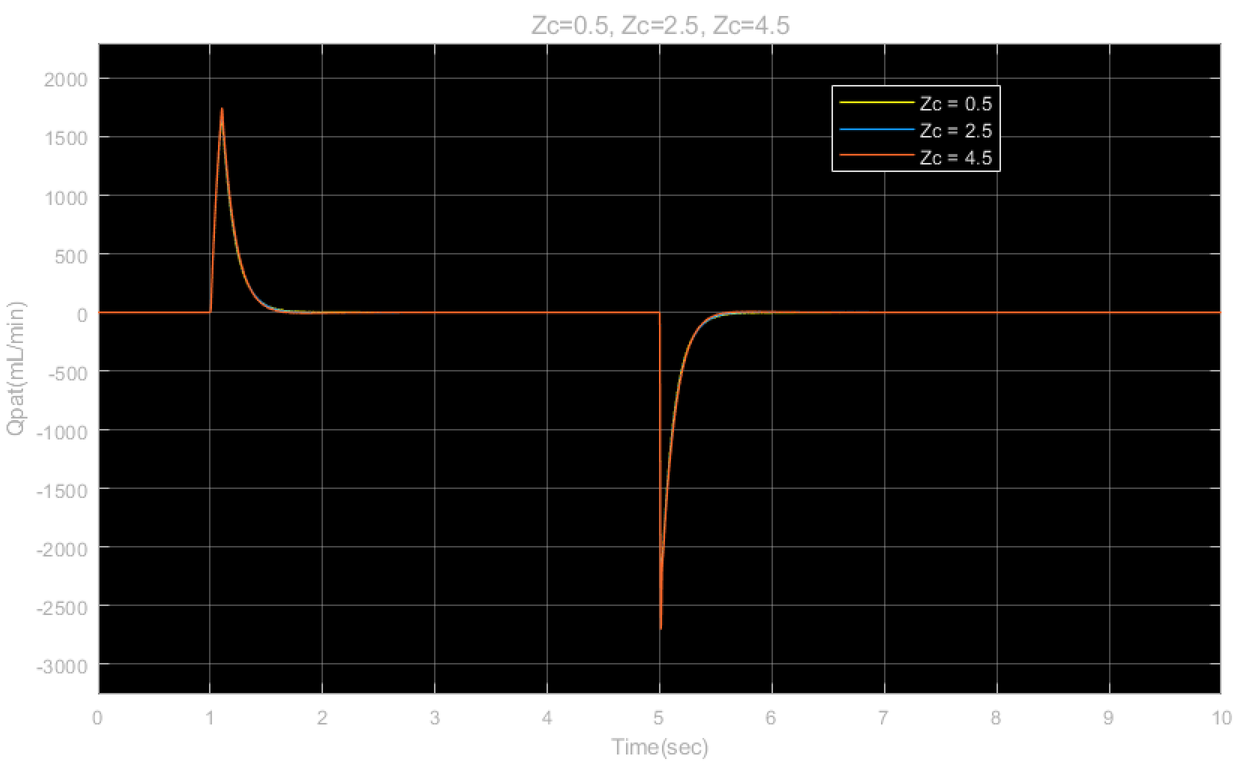
<u>Simulink Diagram:</u>



<u>Plot for output Paw:</u>



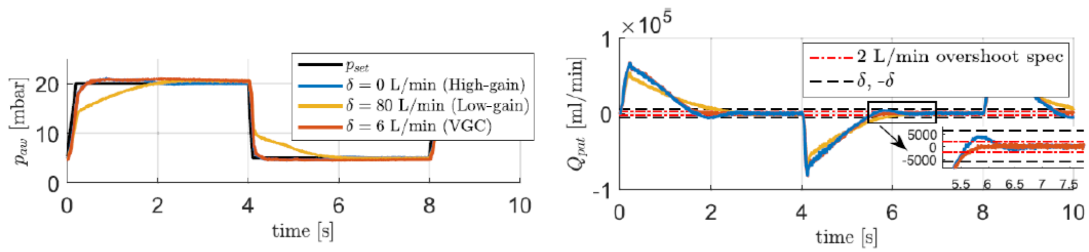
Plot for output Qpat:



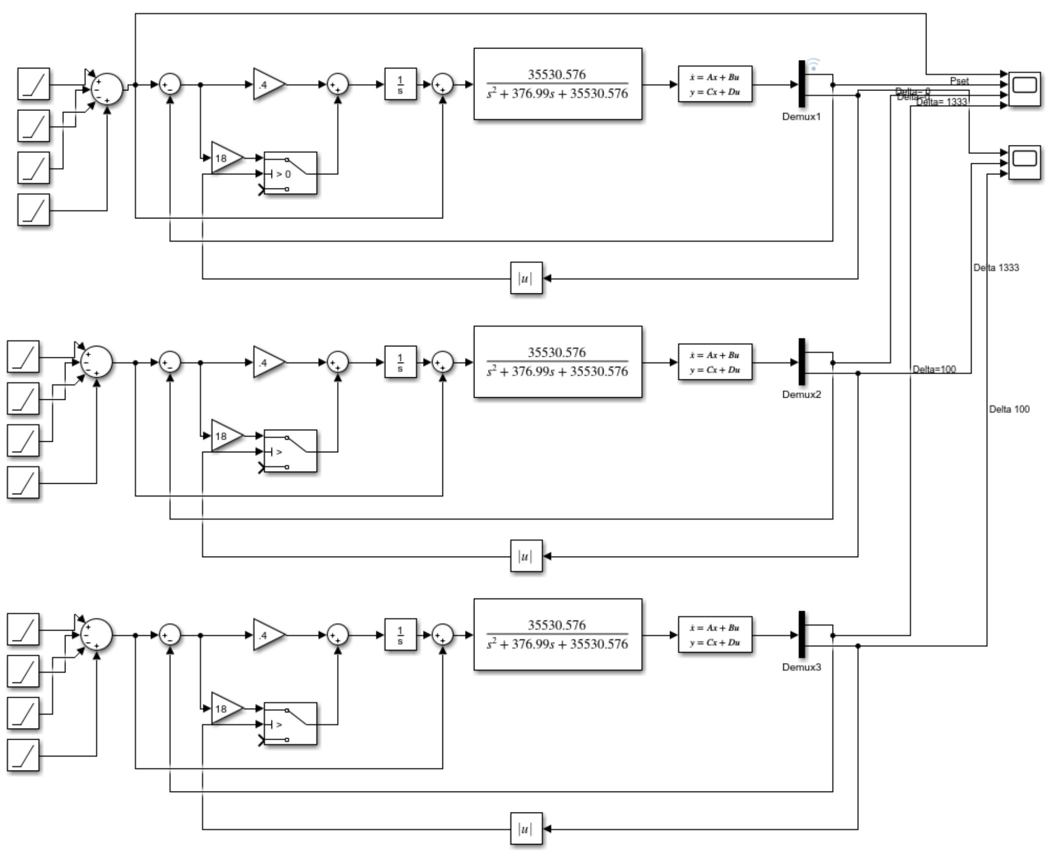
Task7:

Reproducing the following results with both linear and non-linear controllers as well as the discussion on the pros and cons of both controller with respect to each other.

Expected Output:



For this we implemented the following circuit and got these results:



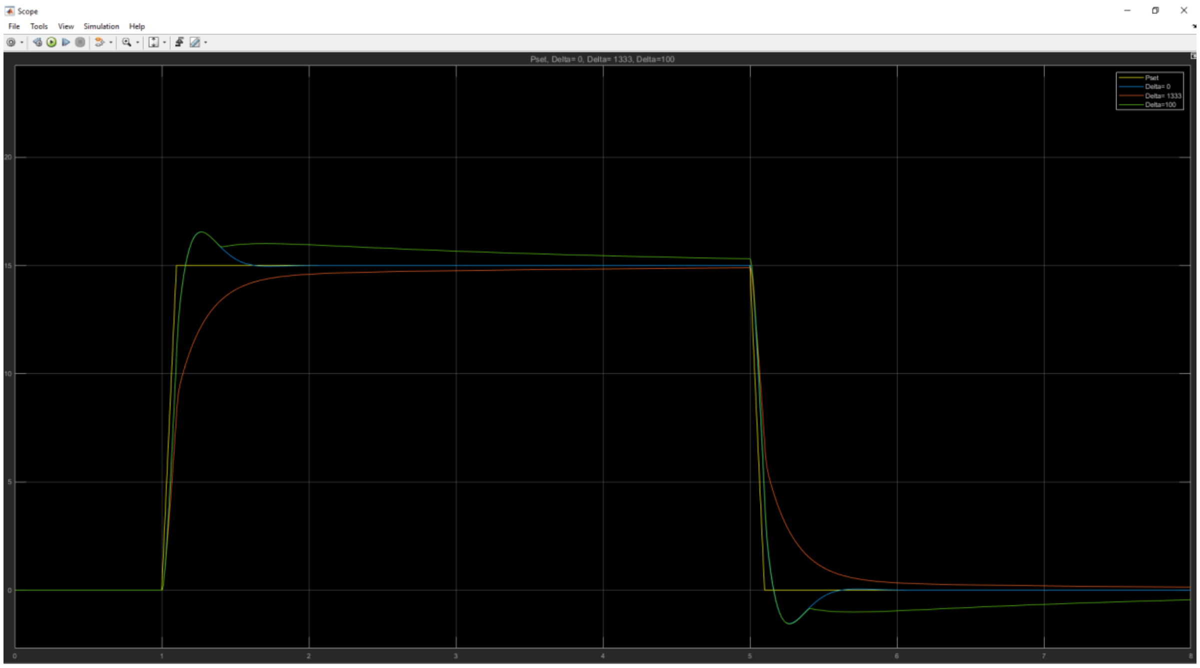


Fig. Paw vs time

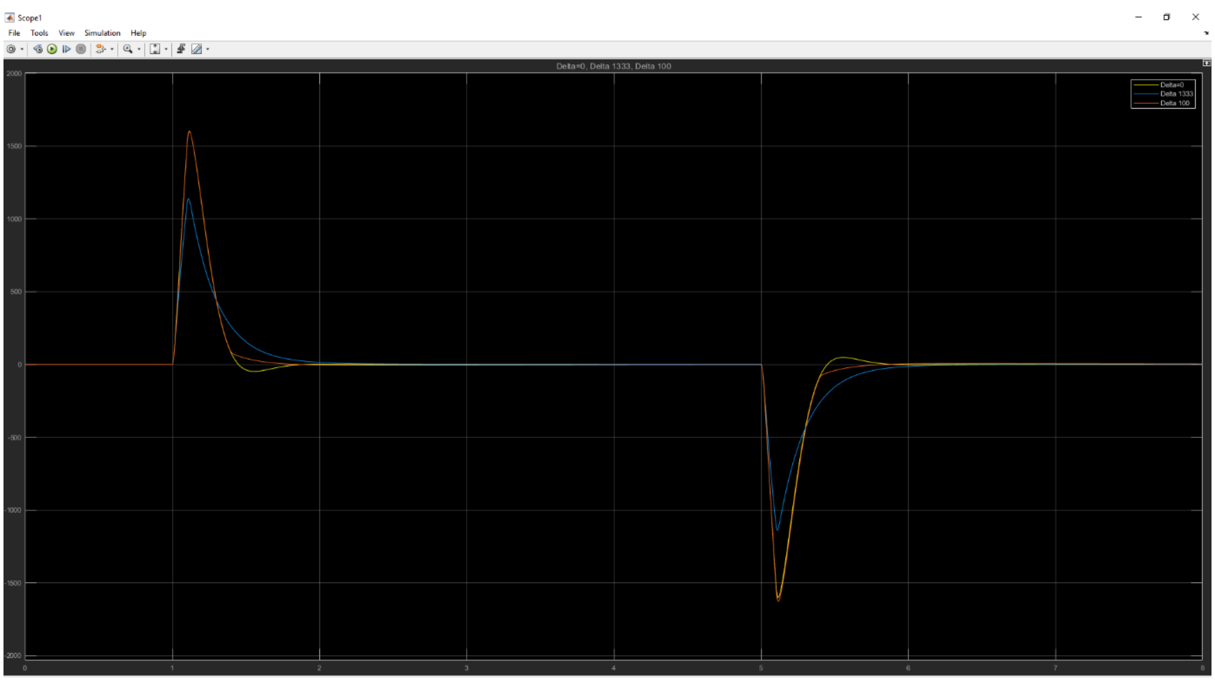


Fig. Qpat vs time

Here, when $\delta = 0$, we are operating with a high gain controller. Thus, the step response for this case should have low rise time and high overshoot value.

When $\delta = 1333$, we are operating with a low gain controller. Thus, the step response for this case should have low overshoot value but high time.

When $\delta = 100$, we are operating with a variable gain controller.

The designed system was required to match the flexibility of the currently available ventilator systems. For a respiratory system, a high gain pressure controller is required to ensure faster pace in pressure buildup and release. However, the unwanted oscillation in the patient flow must be kept smaller. So, a low gain pressure controller is preferred. To solve this dilemma, a variable gain controller is chosen.

For a linear system, in the high gain operation where the pressure builds up, the experimental data shows that the overshoot and the rise time balances each other. However, the data may match with the simulation qualitatively, but the quantitative values are different. This is due to the fact that the lung resistance, Rlung is a quadratic resistance. So, the linear system with a linear lung resistance showed a deflection in the values. The human lung resistance may show linearity or a mix of linear and quadratic behavior.

Moreover, there are no proper value of ki that satisfies both rise time and overshoot specifications. Ki cannot be set to be 2 due to the fact that a slightly larger ki will cause a massive overshoot while a slightly lower ki will cause a larger rise time.

To solve these issues, a variable gain controller was implemented. The controller is non-linear and able to identify the change in the gain of the pressure controller. If the flow exceeds a threshold, gain is risen to compensate for the drop in the pressure hose. When the lung is full and Qpat reaches nearly to zero, the gain falls to ensure stability in the flow without overshoot.
The drawback of the nonlinear system is it sets a boundary to the patient flow overshoot when the low gain controller is activated. However, this is not an issue in a real system. Also, the simulation requires more computational resources, and the controller was more difficult to design than the linear controller. But to ensure quality of performance, the resource issue is to be sacrificed.

Task 08:

Discuss the performance of both linear and nonlinear control systems in presence of uncertainties such as different lung parameters, pressure drop etc.

In this part of the report, we changed some of the parameters of the system such as patient lung parameters C_{lung} and R_{lung}, system parameter R_{leak} and the parameter R_{hose}. We did not change w_n because we consider we will use the same blower.

Original System parameters:

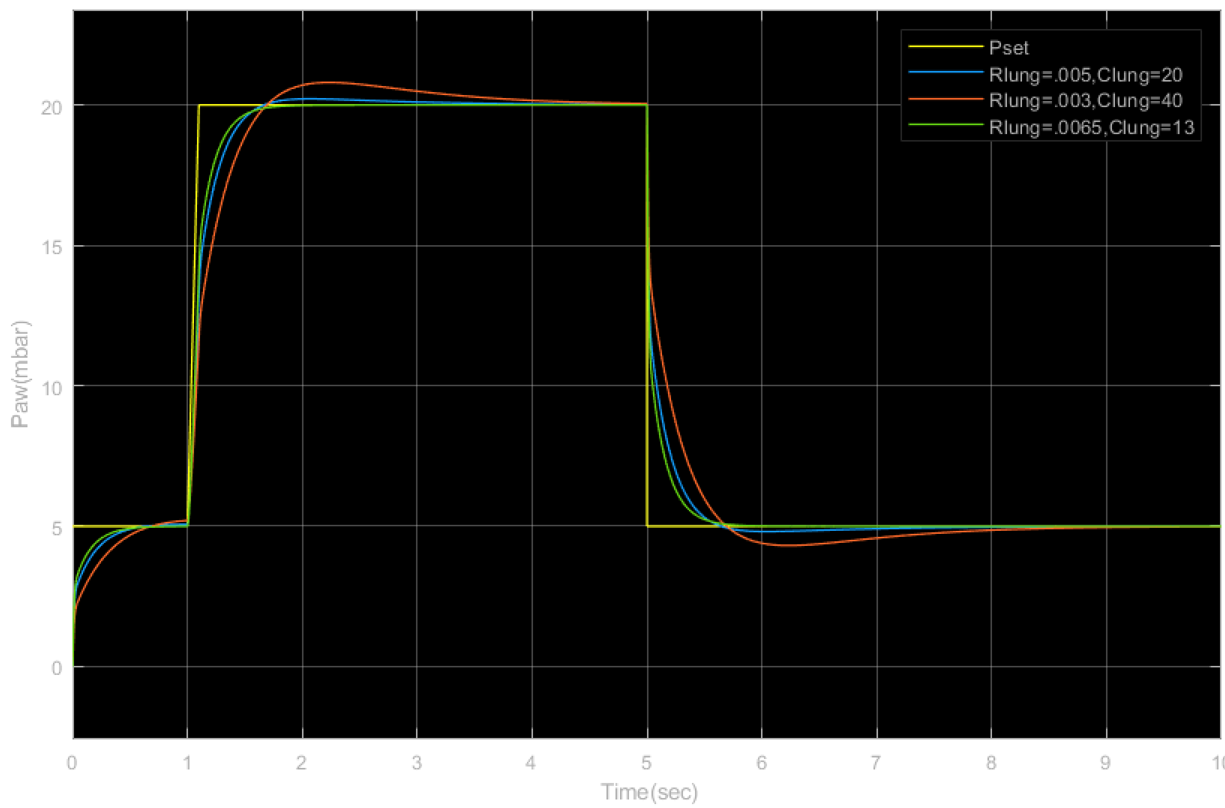
Variable	**Value**
R_{lung}	5/1000
C_{lung}	20
R_{leak}	60/1000
R_{hose}	4.5/1000
w_n	$2\pi(30)$

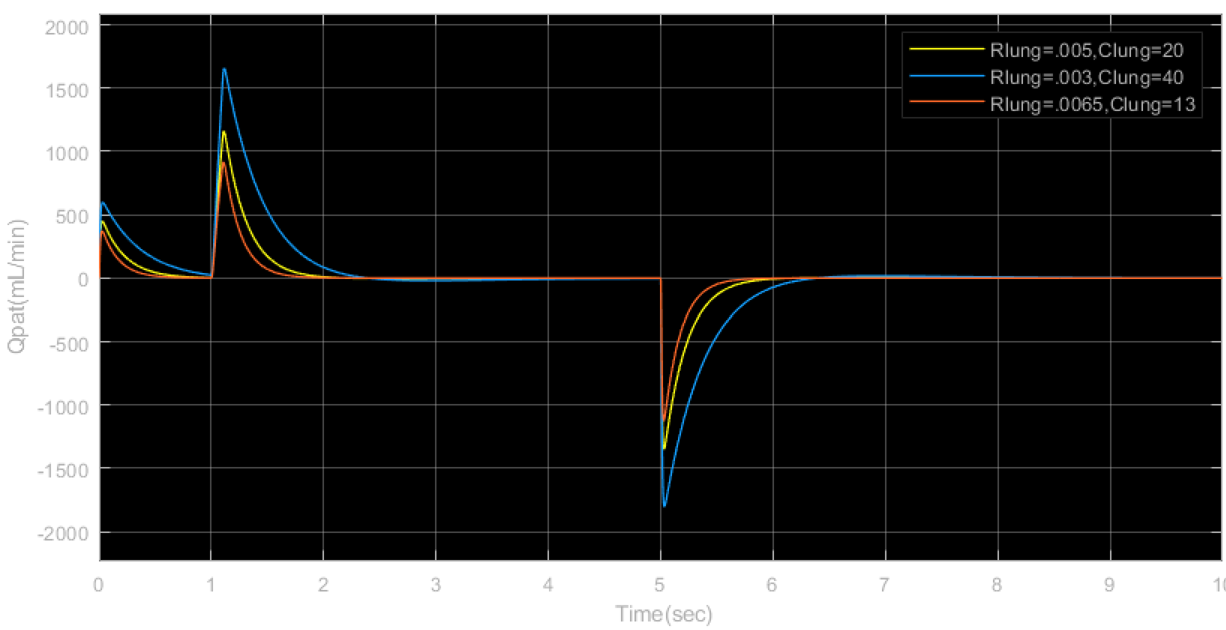
Firstly, we used the R_{lung} and C_{lung} value that is used in the paper, then we decreased the value of R_{lung} and increased the value of C_{lung} and lastly we increased the value of R_{lung} and decreased the value of C_{lung}. In this three cases the output is given below.

The values we used are given below in a table:

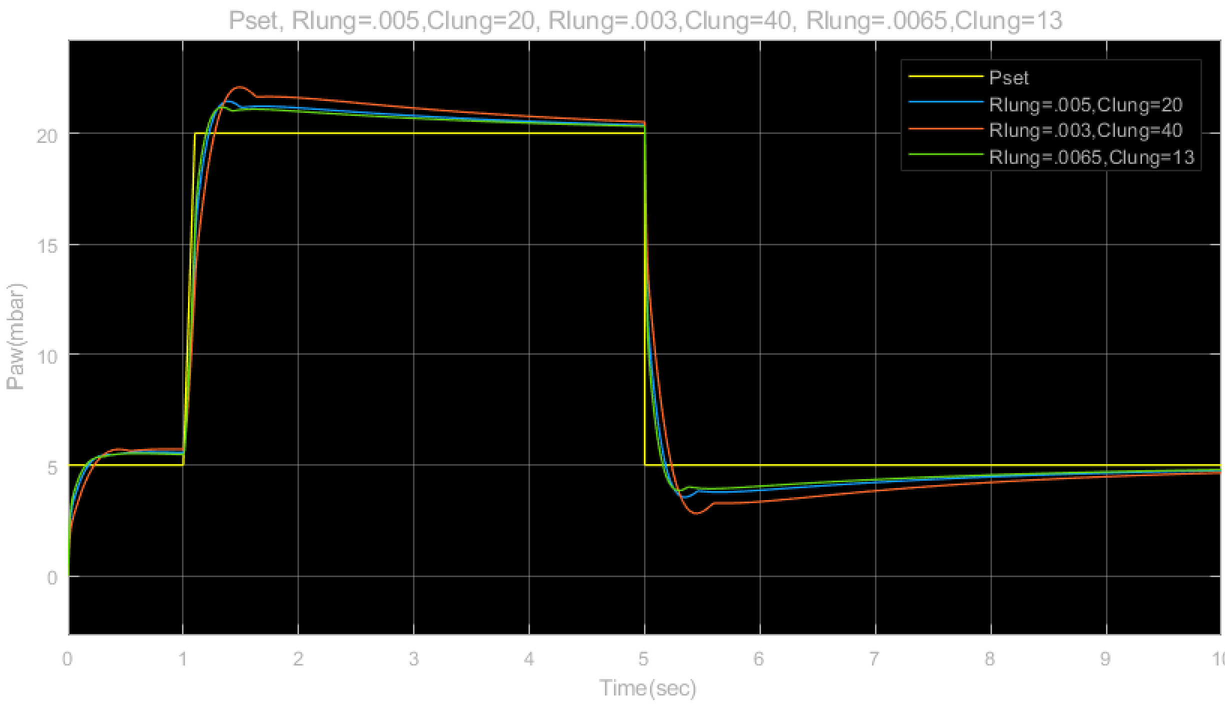
R_{lung}	C_{lung}
0.005	20
.003	40
0.065	13

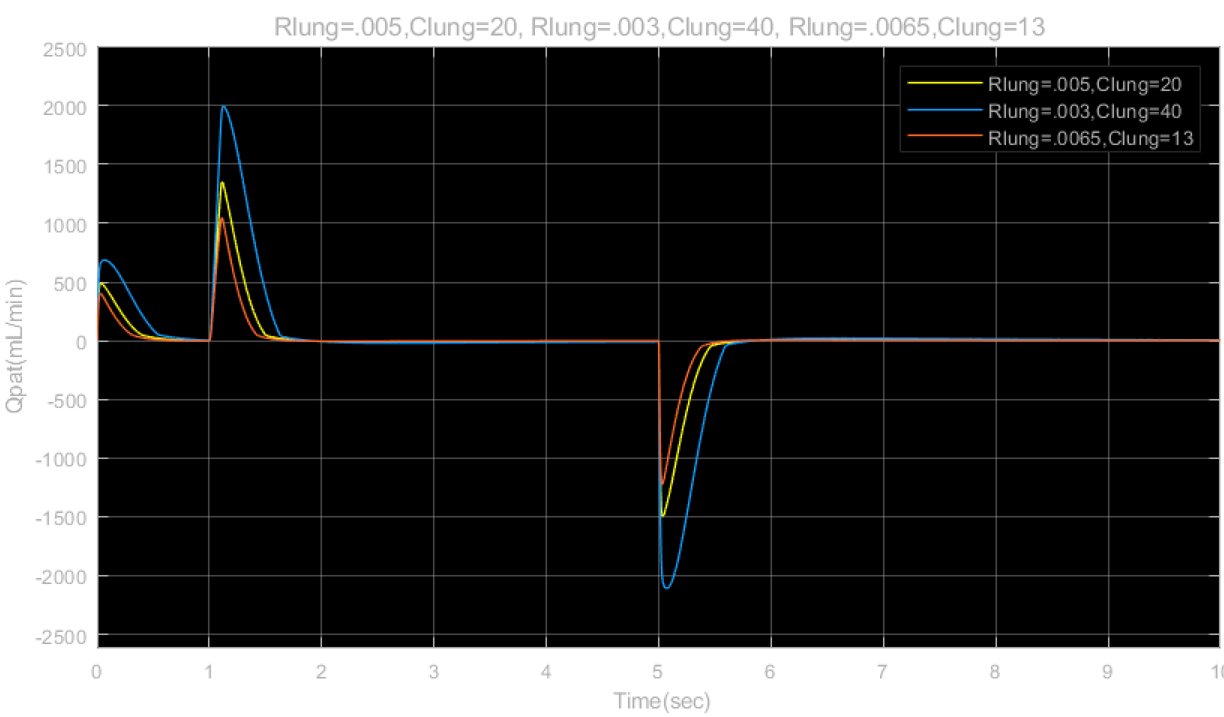
For Linear Gain Controller:





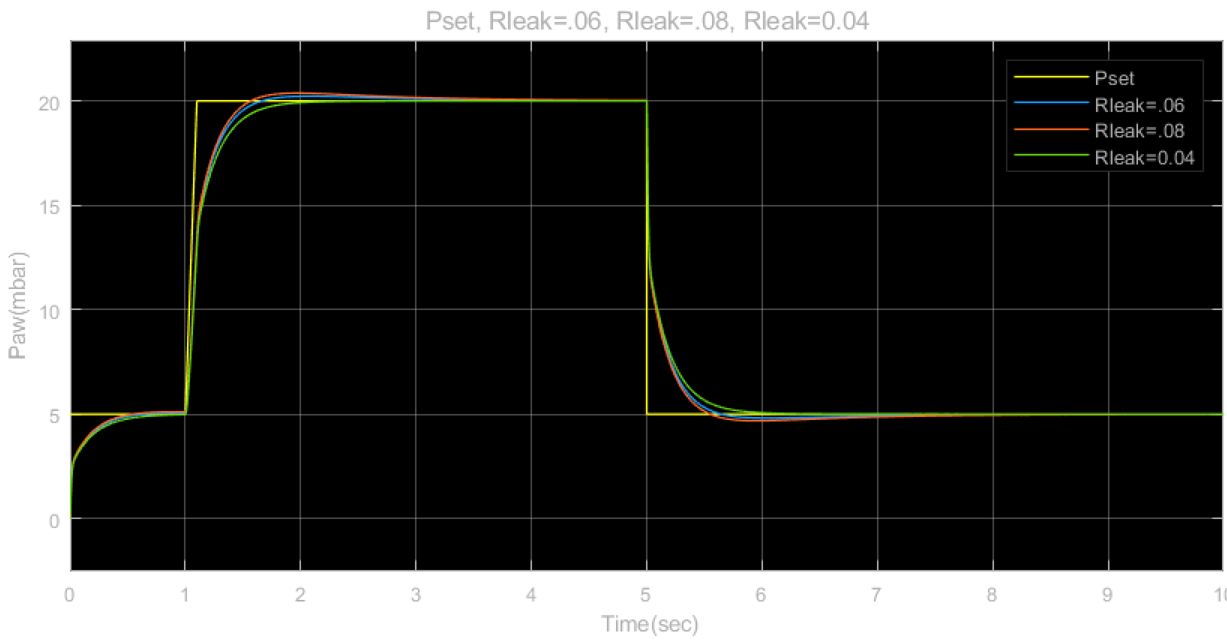
For Variable Gain Controller:

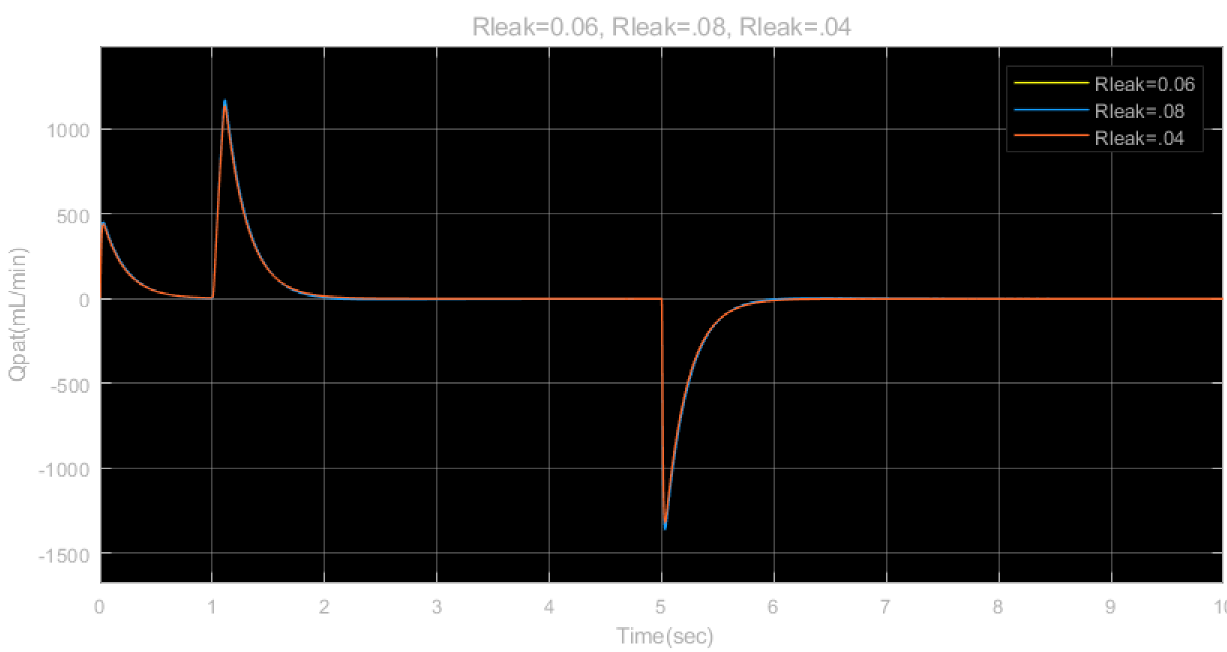




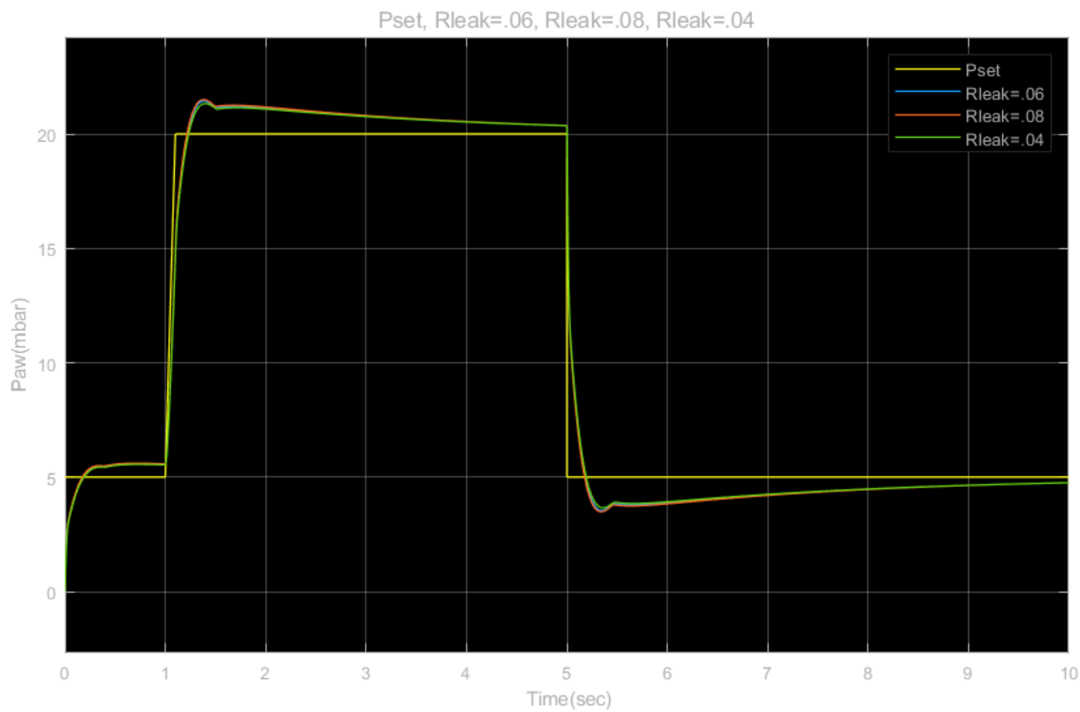
After this we changed the value of R_{leak}. For different R_{leak} we get different value. The output are show below.

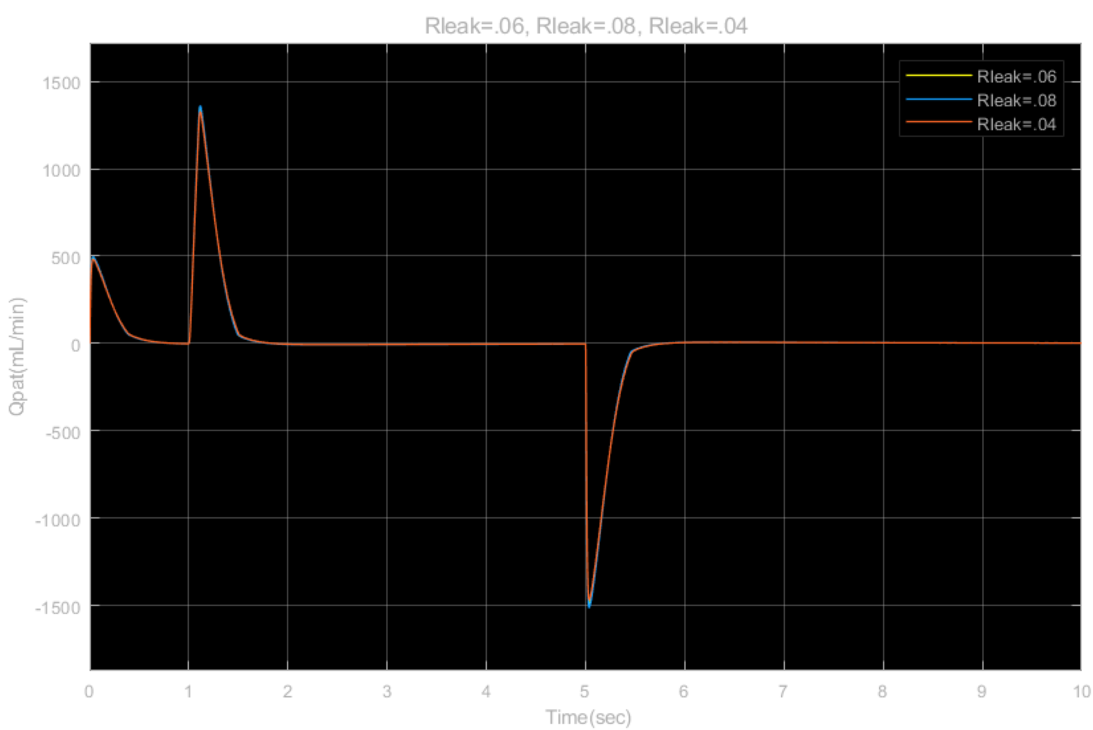
For linear gain controller:





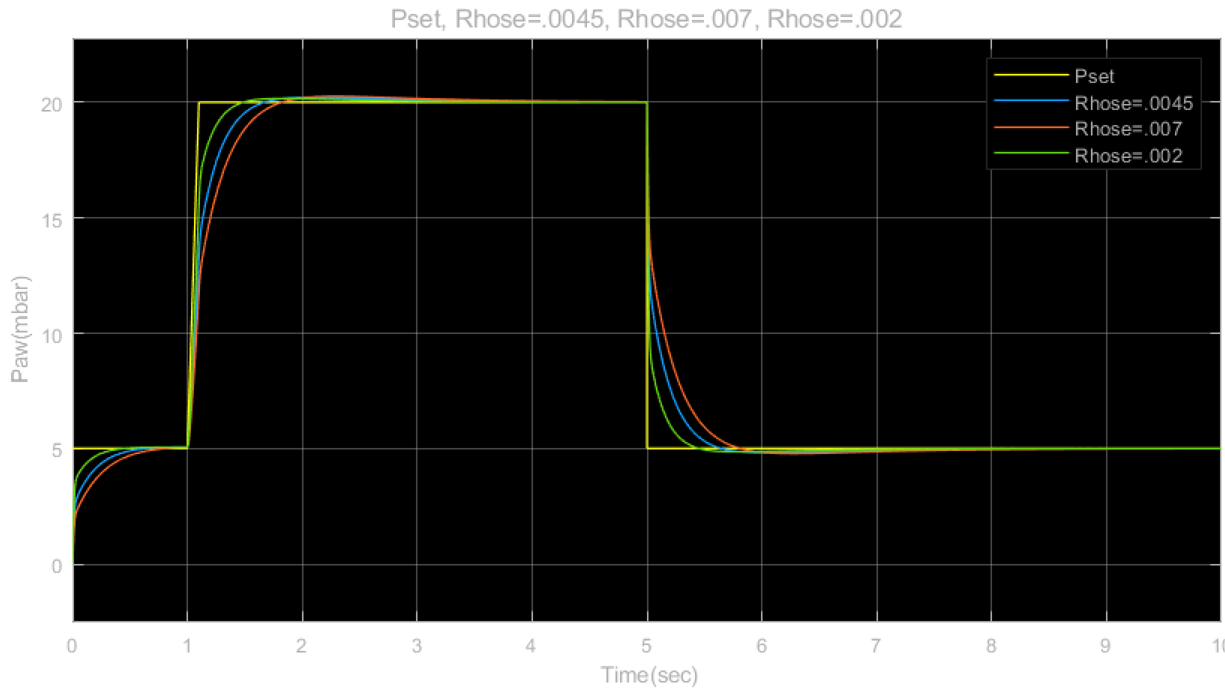
For variable gain controller:

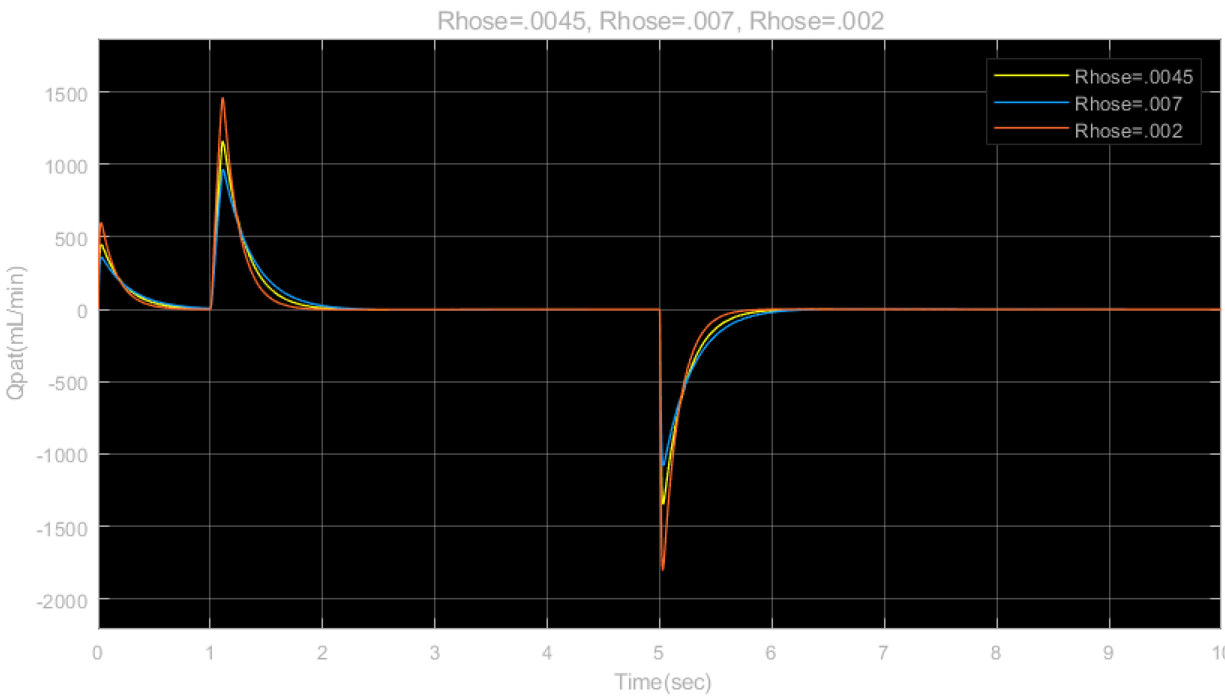




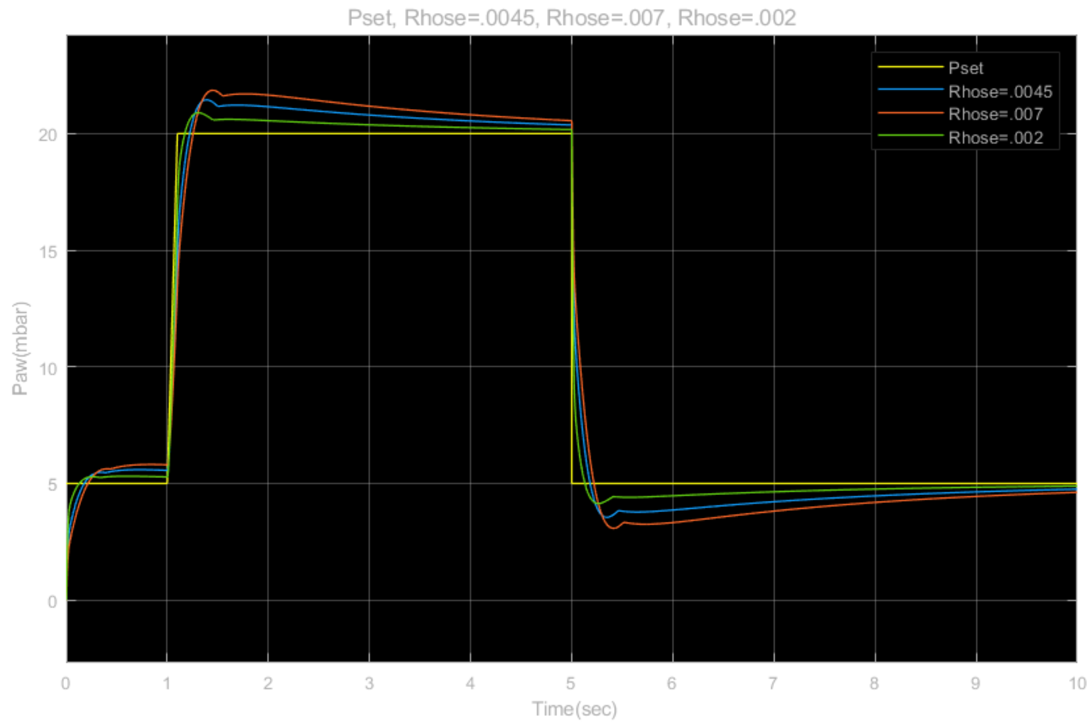
After that we changed the R_{hose}. For different value of R_{hose} the outputs are given below.

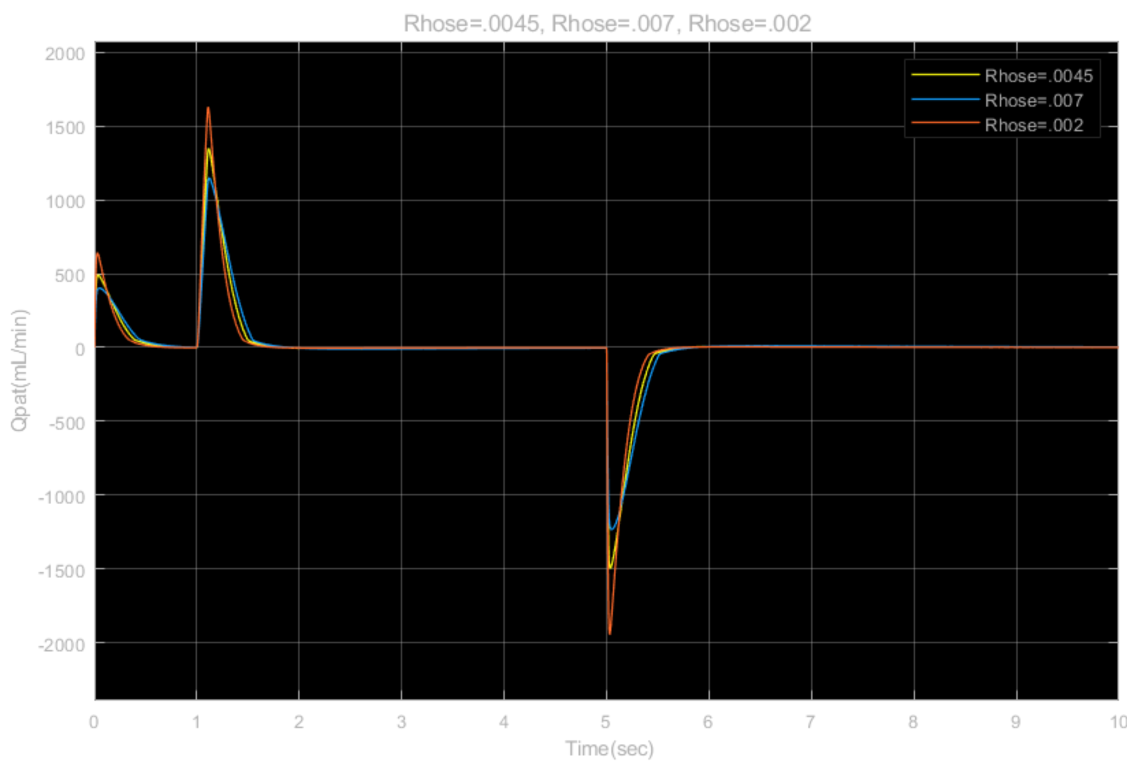
For linear gain controller:





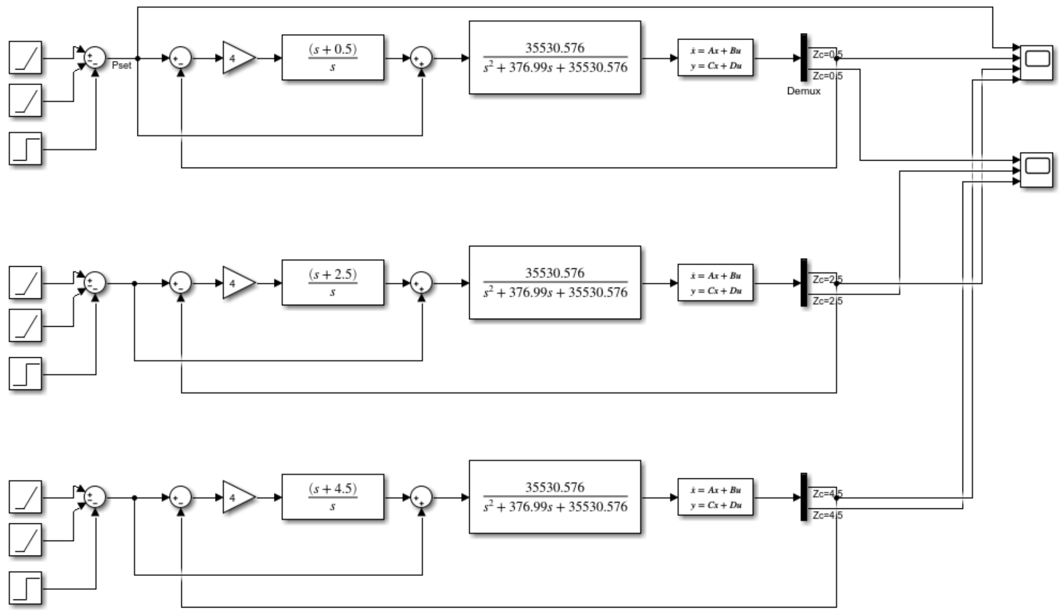
For variable gain controller:



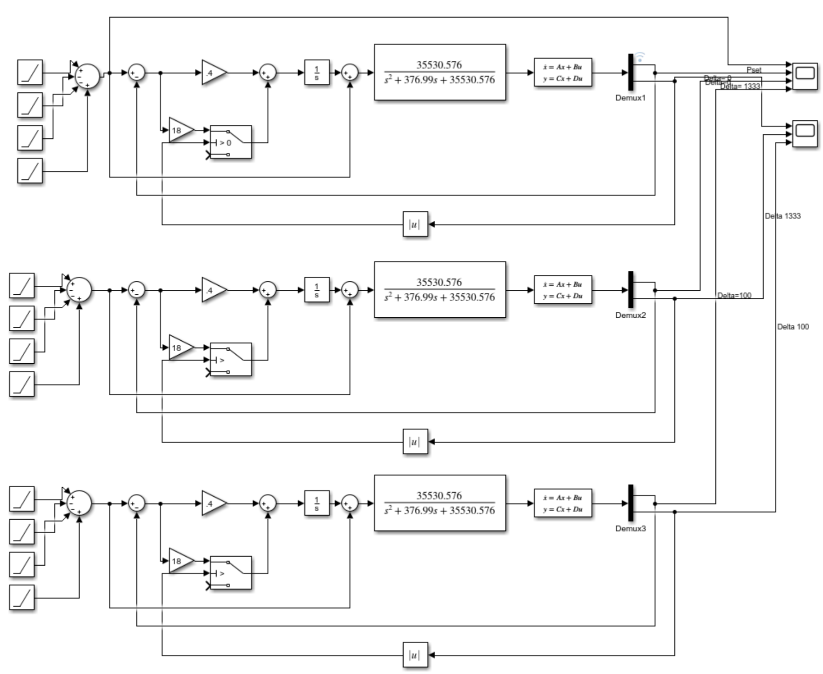


The Circuit Diagrams are that we used are given below:

Linear gain controller:



variable gain controller:



In this way we changed the value of different parameters and observed the changes in the graphs.

Contributions:

1606099- Reproduced the results shown in Fig.7 of the reference paper, discussed the necessity of both feedback and feedforward control.Explored the performances for Ideal intigrator ,PI and Lag controller and designed the preferred linear controller in order to meet the given specifications.

1606100- Completed the task 8. In this task the the value of lung parameters C_{lung} and R_{lung}, system parameter R_{leak} and the parameter R_{hose} were changed and for different value observed the changed in the graph.

1606101- Completed task 2 and 3. Figured out the overall closed loop transfer function and plotted the root locus.

1606103- Worked to simulate the circuit with three different feedback controllers and also compare themselves to find out which controller is best in the system.

1606104- I completed task 1. The task was to find out the overall transfer function.

1606119- Worked in solving linear & variable gain controller system and simulated the corresponding output. Also helped in solving task 8 varying parameters & finding simulation outputs.