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.

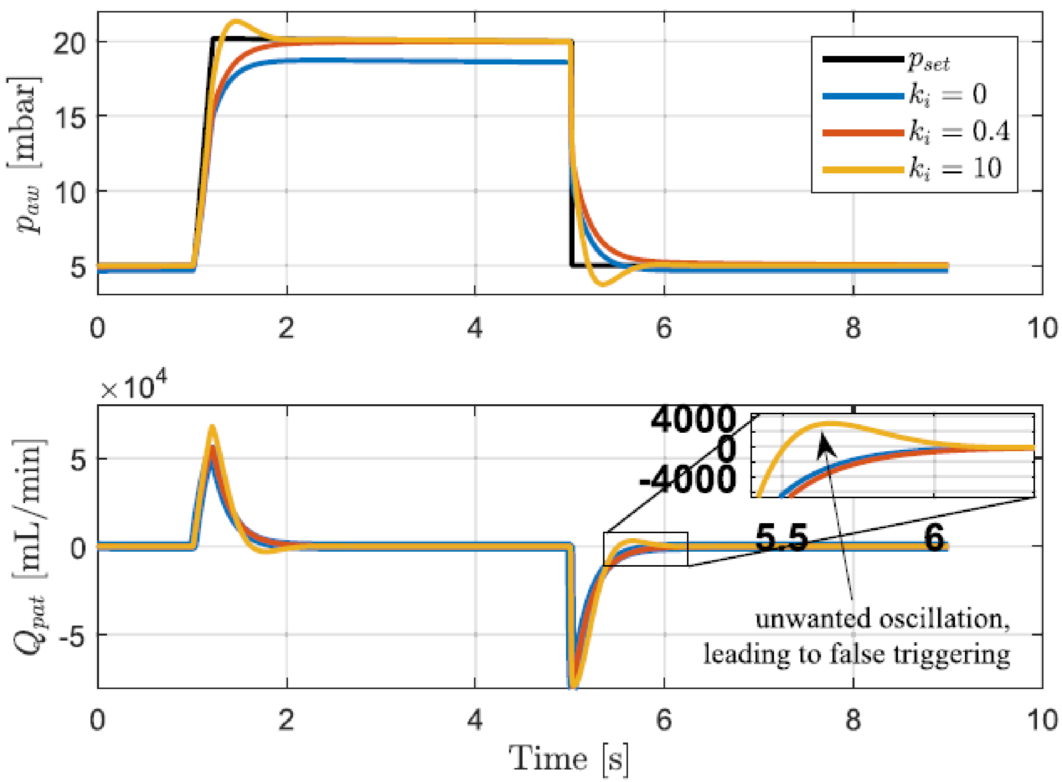
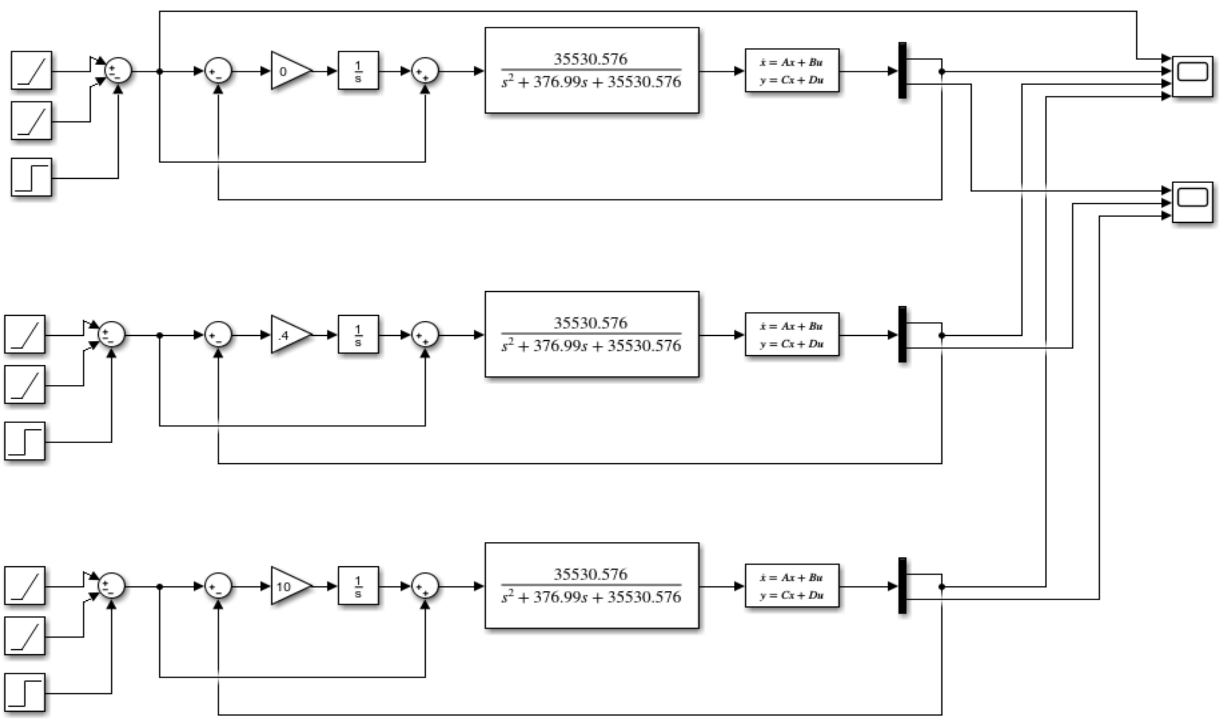
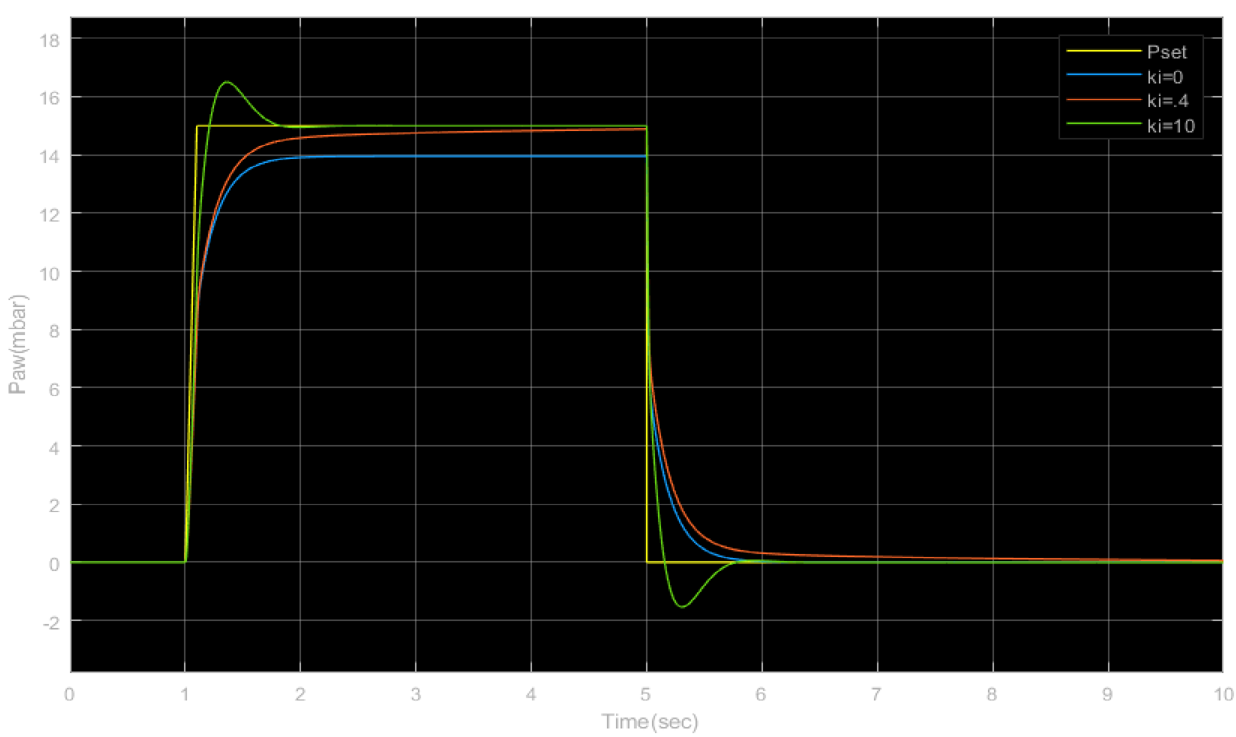


Fig. 7. Simulation result of the closed-loop system using no controller, a low-gain controller ($k_i = 0.4$), and a high-gain controller ($k_i = 10$).

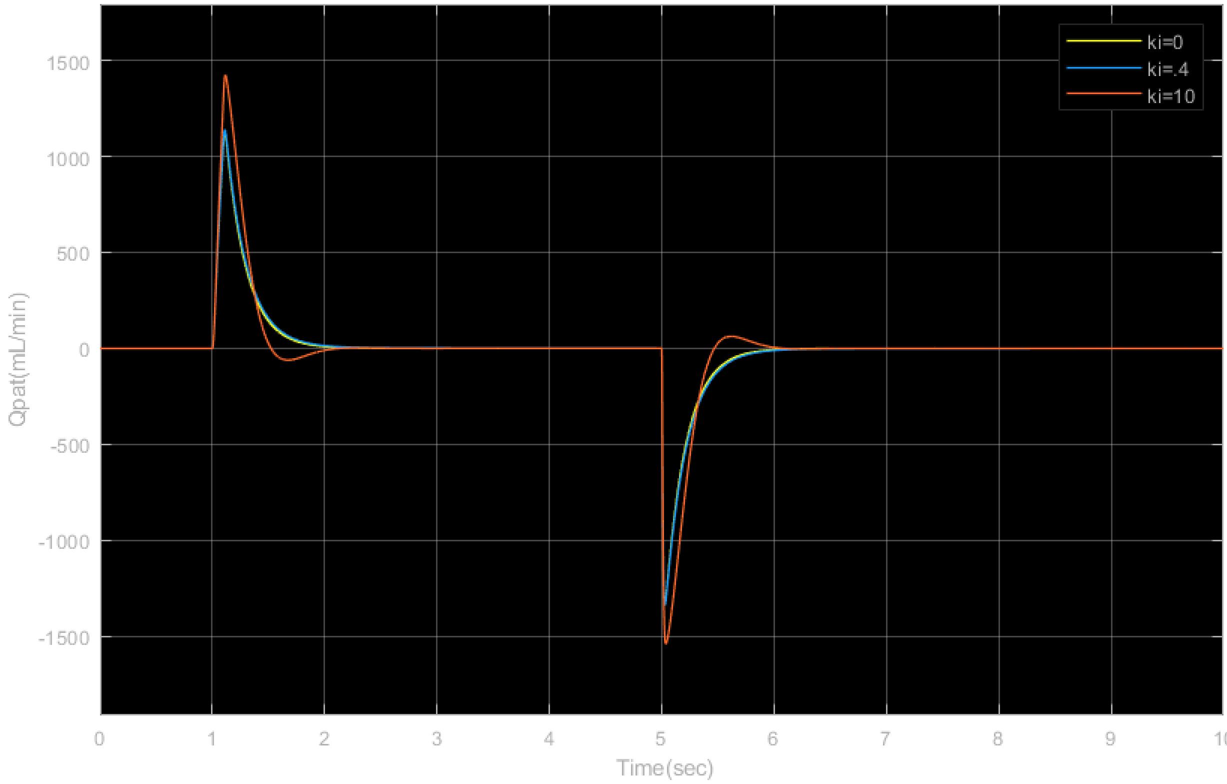
Simulink Diagram:



Plot for output Paw:



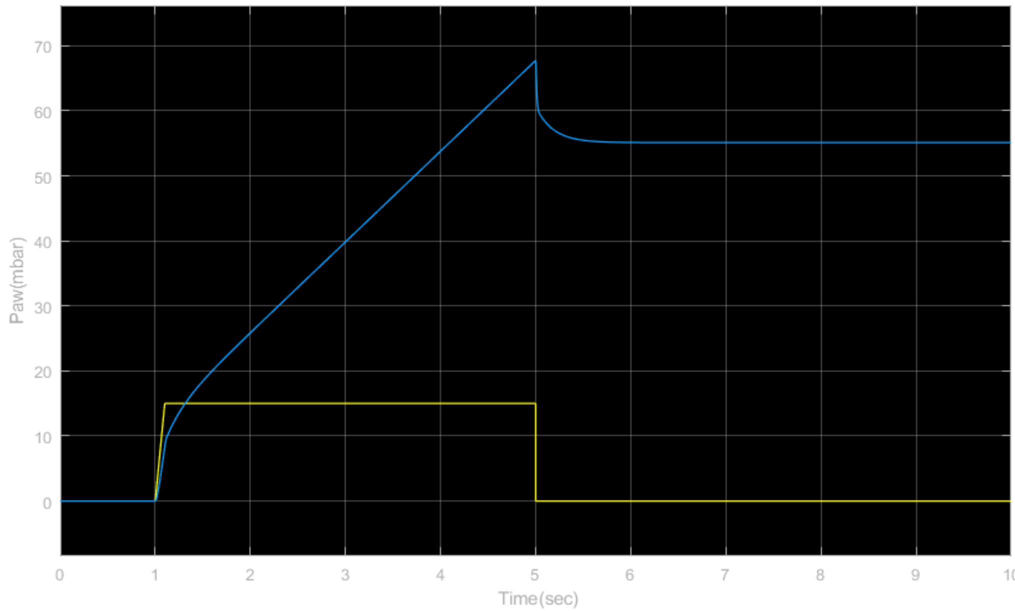
<u>Plot for output Qpat:</u>

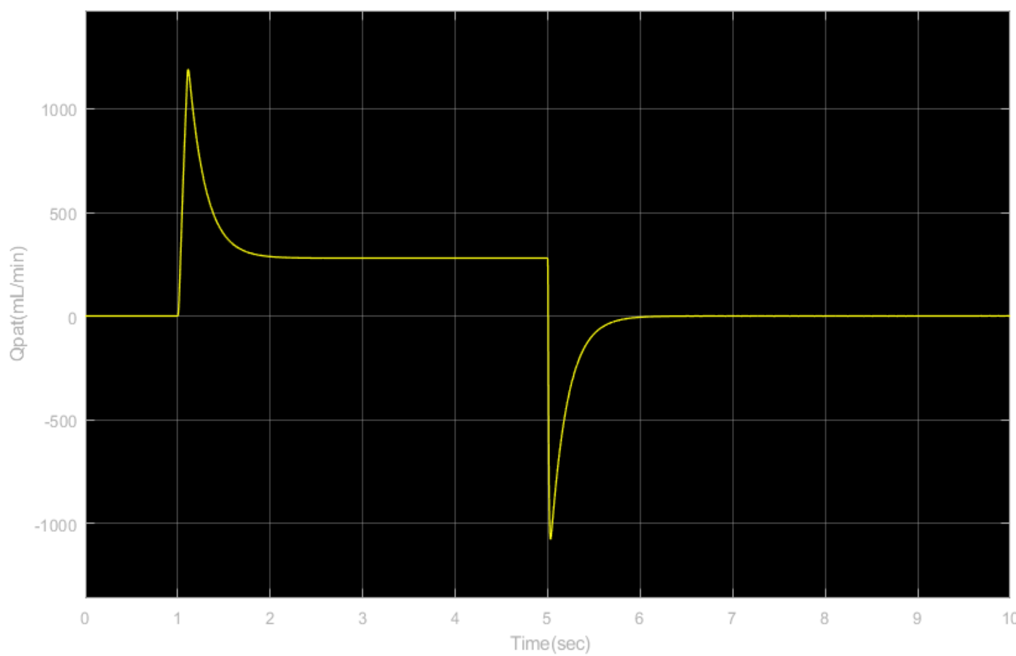


Just as in the paper, we have used ki = 0, 0.4, 10, as controller gain. From the plot, it is clear that at low values of gain, the output air-way pressure can hardly, if at all, reach the pressure set point. On the other hand, for high values of gain, the system can build up pressure fast, at the cost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

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

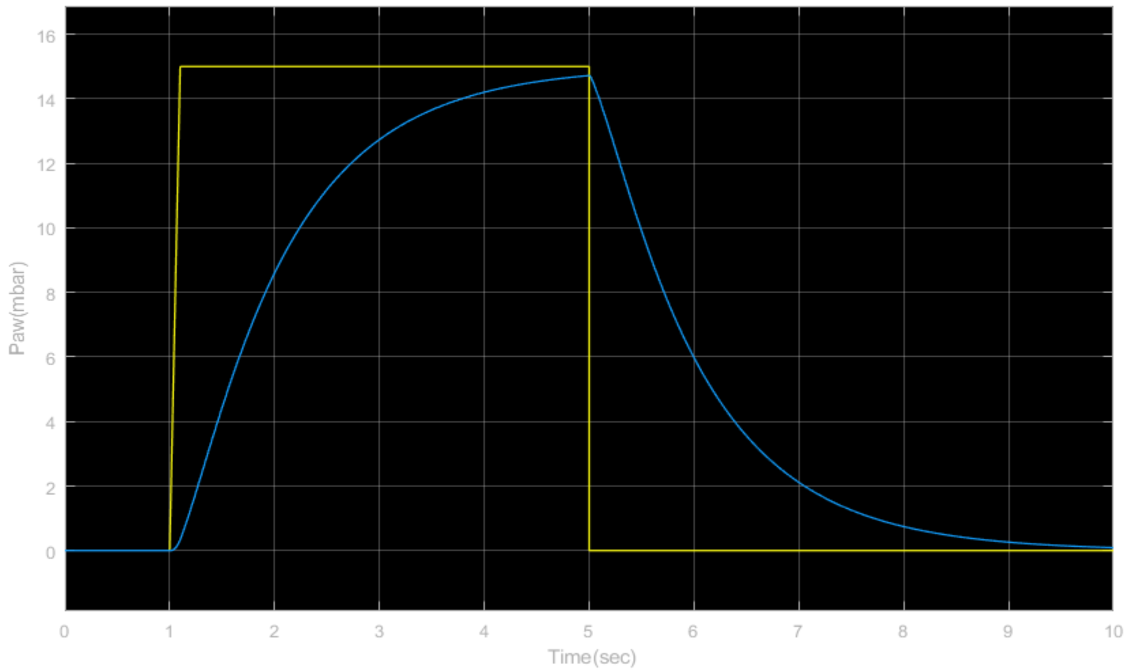
Without feedback:

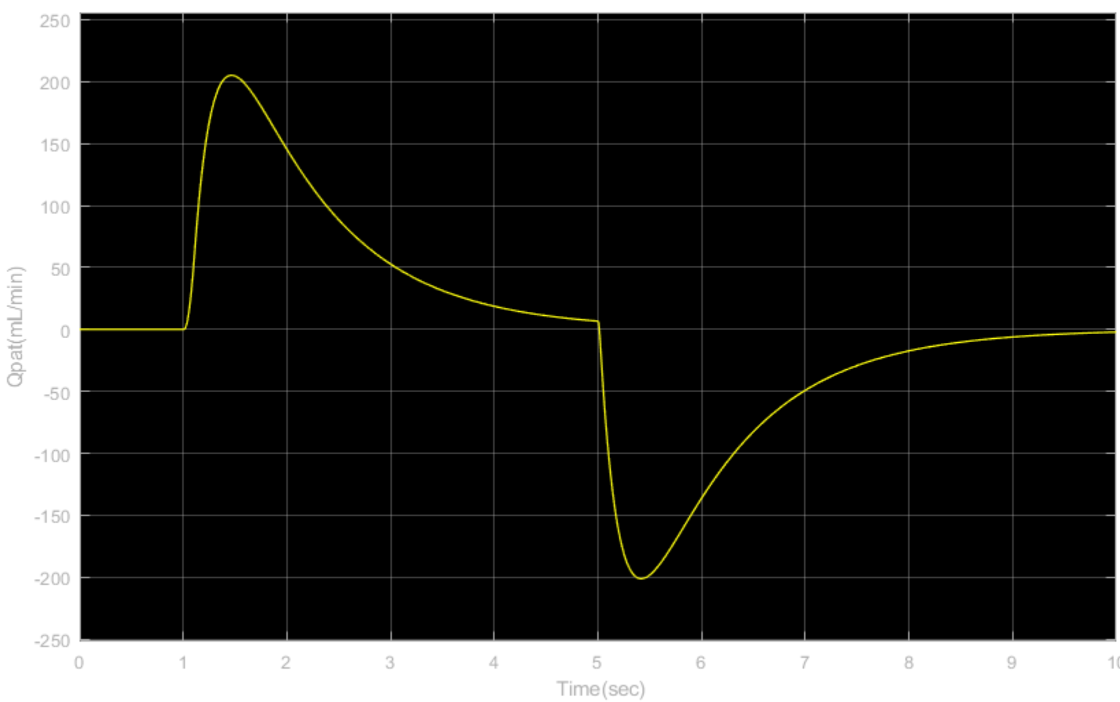




Without feedback, the system isn't aware of the error it is generating. Resulting in very large values of error for both the airway pressure and the airflow.

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?

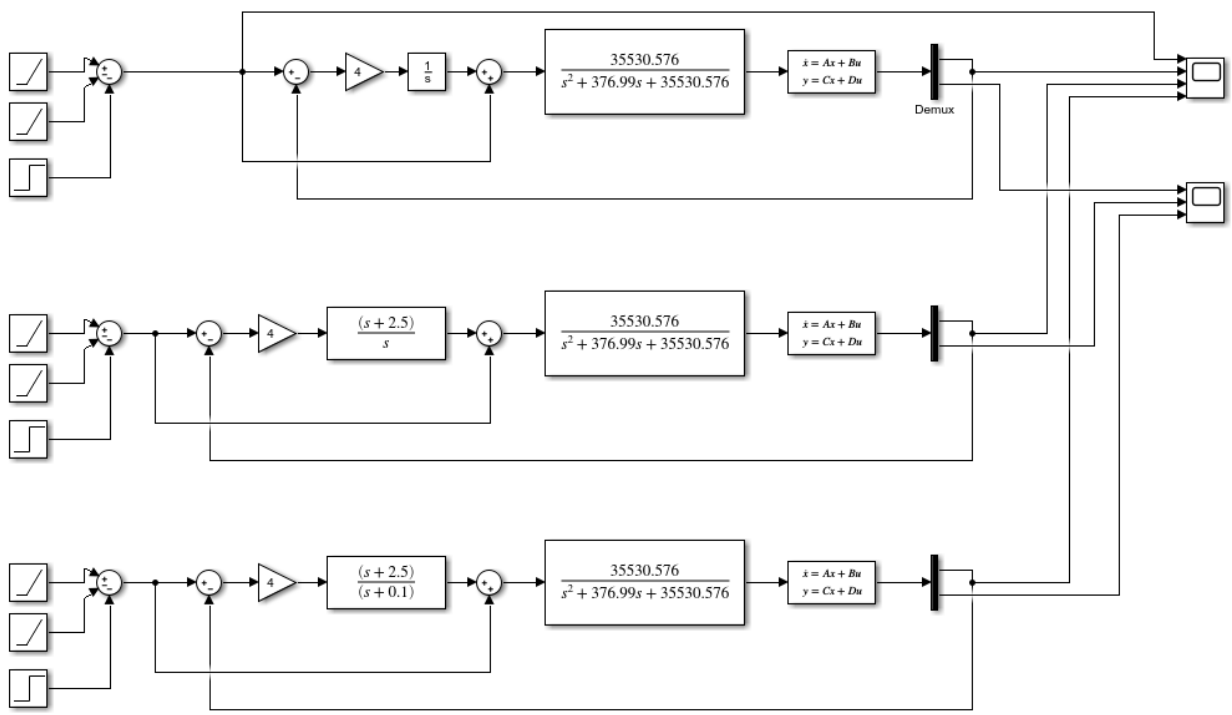
First, we observe the performance of the three types of controllers. Here, we have used,

Ideal integrator: 4/s

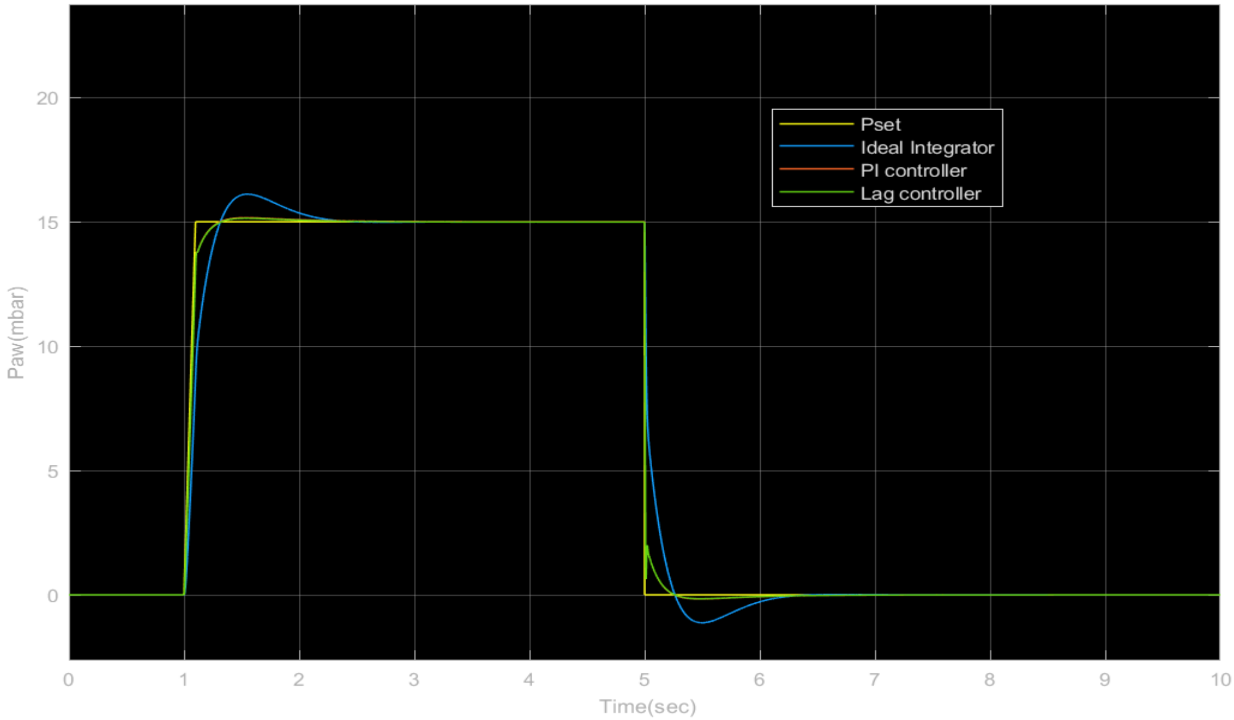
PI controller: 4*(s+2.5)/s

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

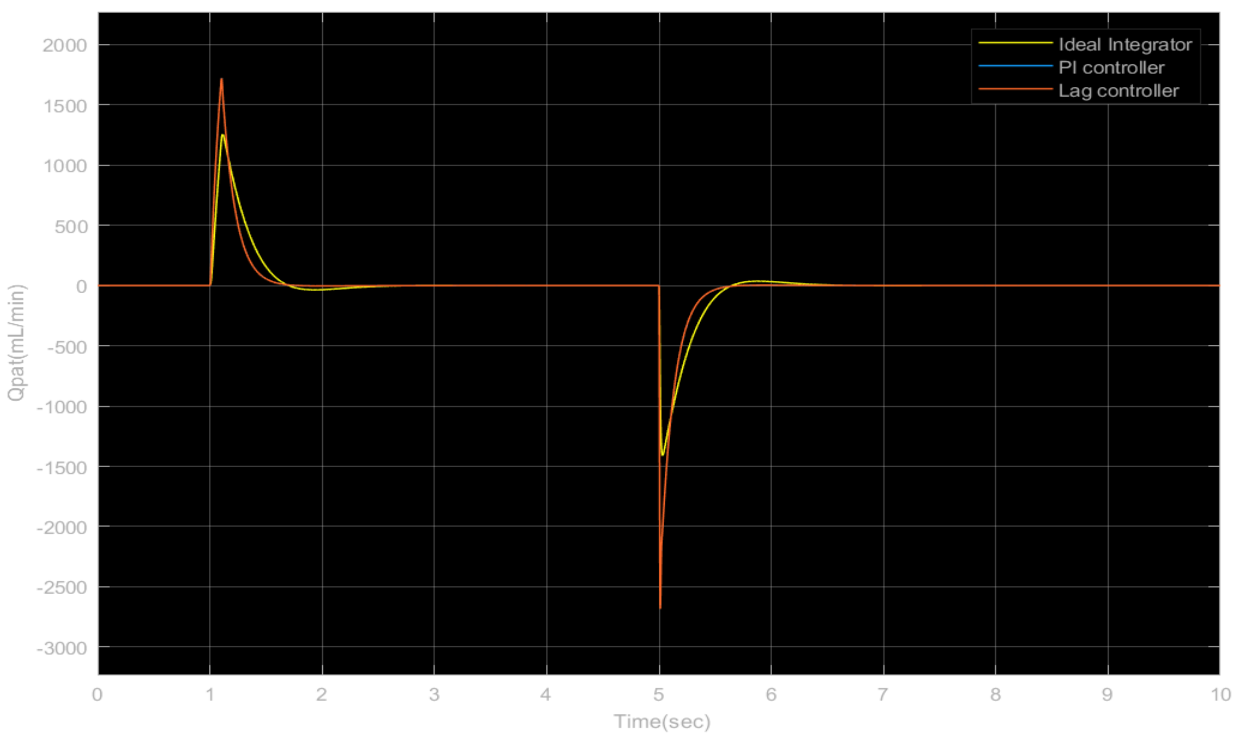
Simulink Diagram:



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

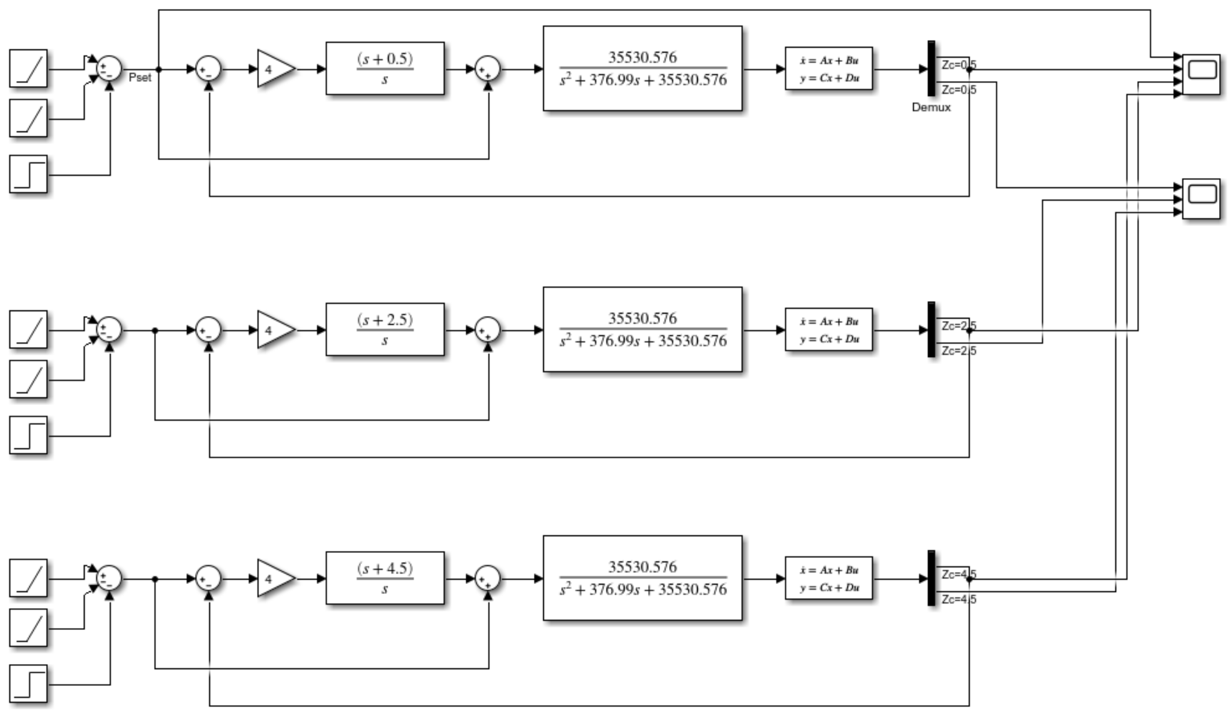
Since Zc = 0.5 gives zero overshoot and error with a significantly quick rise time, we settled on Zc = 0.5 for this controller.

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

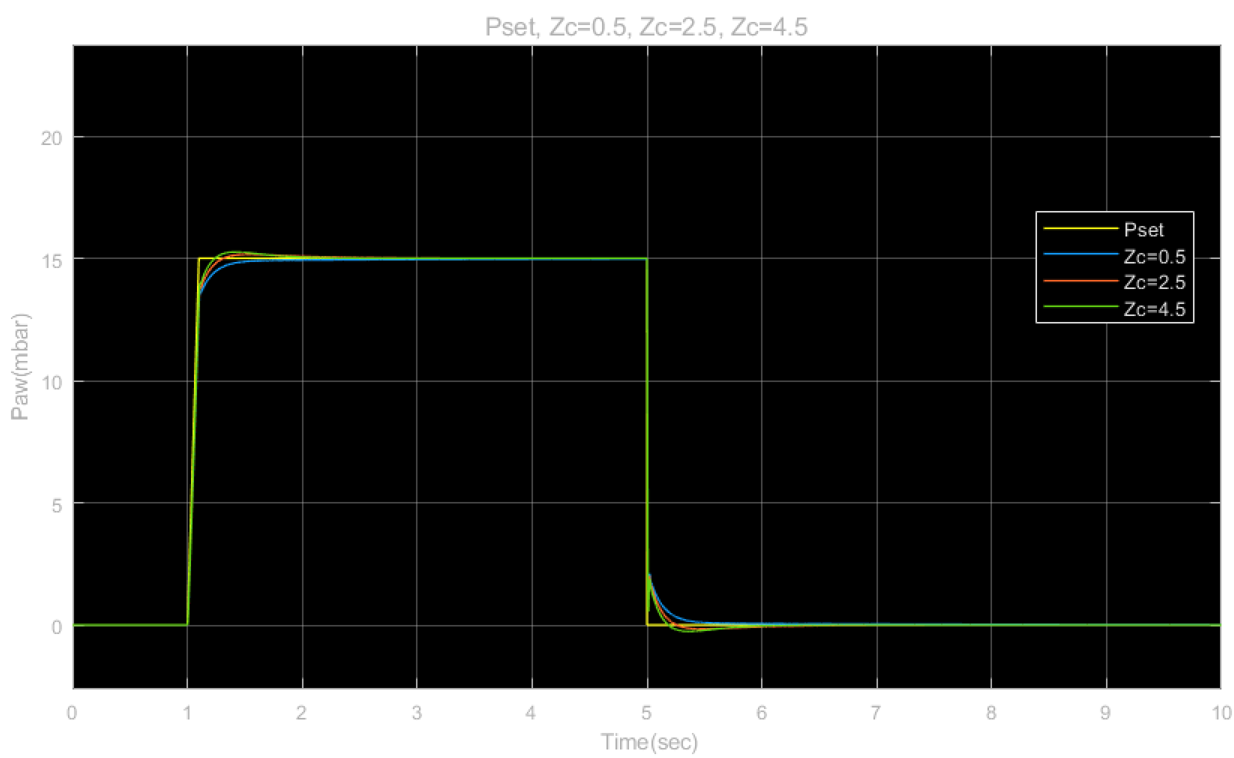
Zc = 0.5	Rise Time(ms)	Error(mbar)	Overshoot(mL/sec)
Ki = 4.5	89	0	0
Ki = 5	87	0	0
Ki = 8	85	0	0

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.

Simulink Diagram:



Plot for output Paw:



Plot for output Qpat: