BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

BUET

Department of Electrical and Electronic Engineering

PROJECT REPORT

Course: EEE 318

Course Title: Control System Sessional

Group No: 02

Group Member: Ishtiaque Ahmed Showmik (1606035)

Mumtahina Islam Sukanya (1606039)

Sami Muhtasim (1606052)

Mohammad Ali Kazmi (1606061)

Md Adnan Faisal Hossain (1606063)

Section: A2

Level/Term: 3/2

Date of Submission: 13/12/2020

Contents

Question 01:

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

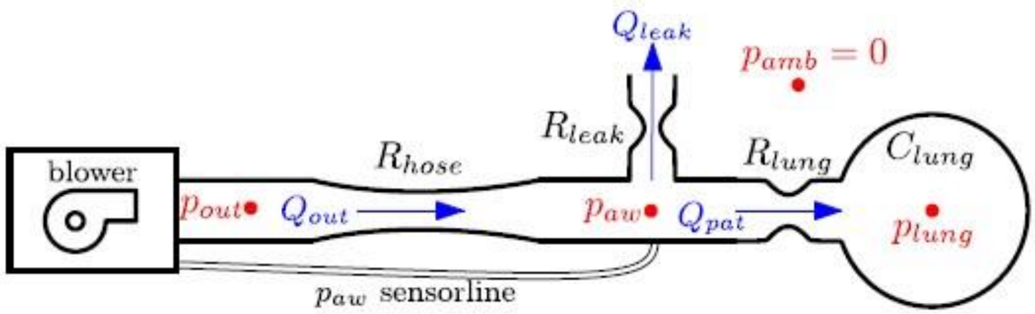
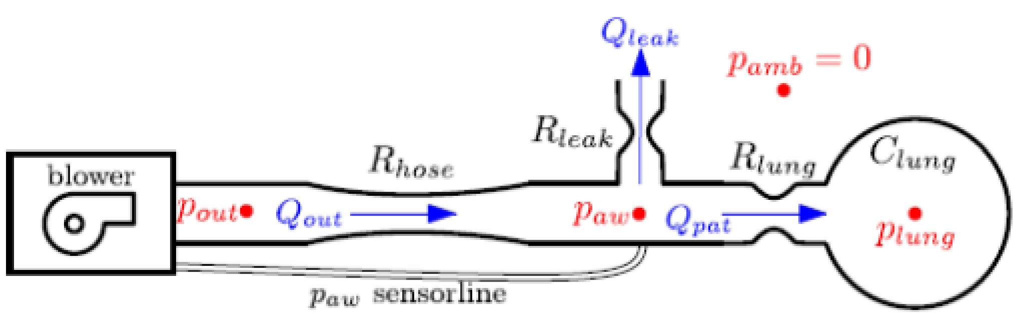
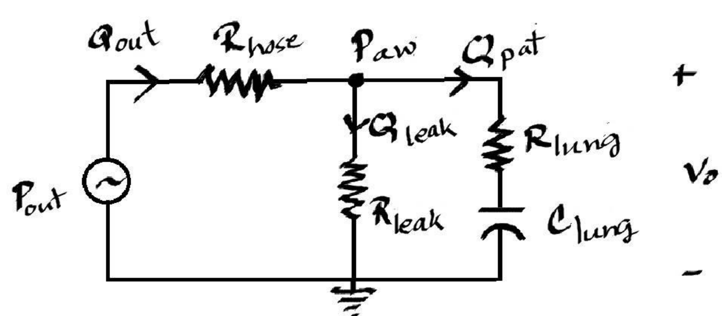


Fig. 3. Schematic of a respiratory system showing the different pressures (red), flows (blue), and resistances and compliance (black).

Answer:



The equivalent circuit is



Symbol	Meaning
P	Voltage
Q	Current
C	Capacitance
R	Resistance

We know, $Q_{out} = \frac{P_{out}-P_{aw}}{R_{hose}}$ and $Q_{leak} = \frac{P_{aw}}{R_{leak}}$

Hence, $Q_{pat} = Q_{out} - Q_{leak} = \frac{P_{aw}-P_{lung}}{R_{lung}}$ ....(1)

Also, $\dot{P}_{lung} = \frac{dP_{lung}}{dt} = \frac{\dot{Q}_{pat}}{C_{lung}} = \frac{P_{aw}-P_{lung}}{R_{lung}C_{lung}}$ ....(2)

We shall apply KCL at node 1,

$$Q_{out} = Q_{leak} + Q_{pat}$$

$$\Rightarrow \frac{P_{out} - P_{aw}}{R_{hose}} = \frac{P_{aw}}{R_{leak}} + \frac{P_{aw} - P_{lung}}{R_{lung}}$$

$$\Rightarrow P_{aw}\left(\frac{1}{R_{leak}} + \frac{1}{R_{lung}} + \frac{1}{R_{hose}}\right) = \frac{P_{out}}{R_{hose}} + \frac{P_{lung}}{R_{lung}}$$

$$\Rightarrow P_{aw} = \left[\frac{\frac{1}{R_{hose}}}{\frac{1}{R_{leak}} + \frac{1}{R_{lung}} + \frac{1}{R_{hose}}}\right]P_{out} + \left[\frac{\frac{1}{R_{lung}}}{\frac{1}{R_{leak}} + \frac{1}{R_{lung}} + \frac{1}{R_{hose}}}\right]P_{lung}$$

Now if we put the value of P_{aw} in equation (1), we get

$$Q_{pat} = \frac{\frac{P_{lung}}{R_{lung}} + \frac{P_{out}}{R_{hose}}}{R_{lung}\left(\frac{1}{R_{lung}} + \frac{1}{R_{hose}} + \frac{1}{R_{leak}}\right)} - \frac{P_{lung}}{R_{lung}}$$

$$\Rightarrow Q_{pat} = \frac{\frac{1}{R_{hose}}}{R_{lung}\left(\frac{1}{R_{lung}} + \frac{1}{R_{hose}} + \frac{1}{R_{leak}}\right)}P_{out} + \frac{\frac{1}{R_{lung}} - \left(\frac{1}{R_{lung}} + \frac{1}{R_{hose}} + \frac{1}{R_{leak}}\right)}{R_{lung}\left(\frac{1}{R_{lung}} + \frac{1}{R_{hose}} + \frac{1}{R_{leak}}\right)}P_{lung}$$

$$\Rightarrow Q_{pat} = \left[\frac{\frac{1}{R_{hose}}}{R_{lung}\left(\frac{1}{R_{lung}} + \frac{1}{R_{hose}} + \frac{1}{R_{leak}}\right)}\right]P_{out} - \left[\frac{\frac{1}{R_{hose}} + \frac{1}{R_{leak}}}{R_{lung}\left(\frac{1}{R_{lung}} + \frac{1}{R_{hose}} + \frac{1}{R_{leak}}\right)}\right]P_{lung}$$

We now put the value of P_{aw} in the equation (2)

$$\dot{P}_{lung} = \frac{\frac{P_{lung}}{R_{lung}} + \frac{P_{out}}{R_{hose}}}{C_{lung}R_{lung}\left(\frac{1}{R_{lung}} + \frac{1}{R_{hose}} + \frac{1}{R_{leak}}\right)} - \frac{P_{lung}}{R_{lung}C_{lung}}$$

$$\Rightarrow \dot{P}_{lung} = \left[\frac{\frac{1}{R_{hose}}}{C_{lung}R_{lung}\left(\frac{1}{R_{lung}} + \frac{1}{R_{hose}} + \frac{1}{R_{leak}}\right)}\right]P_{out} - \left[\frac{\frac{1}{R_{hose}} + \frac{1}{R_{leak}}}{C_{lung}R_{lung}\left(\frac{1}{R_{lung}} + \frac{1}{R_{hose}} + \frac{1}{R_{leak}}\right)}\right]P_{lung}$$

Here, the system is SIMO system with one input and two outputs. Input is P_{out} and outputs are $\begin{bmatrix} P_{aw} \\ Q_{pat} \end{bmatrix}$, while state is P_{lung}

Now, $\dot{P}_{lung} = A_h P_{lung} + B_h P_{out}$

$$\begin{bmatrix} P_{aw} \\ Q_{pat} \end{bmatrix} = C_h P_{lung} + D_h P_{out}$$

Here, $A_h = -\frac{\frac{1}{R_{hose}} + \frac{1}{R_{leak}}}{C_{lung}R_{lung}\left(\frac{1}{R_{lung}} + \frac{1}{R_{hose}} + \frac{1}{R_{leak}}\right)}$

$$B_h = \frac{\frac{1}{R_{hose}}}{C_{lung}R_{lung}\left(\frac{1}{R_{lung}} + \frac{1}{R_{hose}} + \frac{1}{R_{leak}}\right)}$$

$$C_h = \begin{bmatrix} \dfrac{\frac{1}{R_{lung}}}{\frac{1}{R_{leak}} + \frac{1}{R_{lung}} + \frac{1}{R_{hose}}} \\ -\dfrac{\frac{1}{R_{hose}} + \frac{1}{R_{leak}}}{C_{lung}R_{lung}\left(\frac{1}{R_{lung}} + \frac{1}{R_{hose}} + \frac{1}{R_{leak}}\right)} \end{bmatrix}$$

$$D_h = \begin{bmatrix} \dfrac{\frac{1}{R_{hose}}}{\frac{1}{R_{leak}} + \frac{1}{R_{lung}} + \frac{1}{R_{hose}}} \\ \dfrac{\frac{1}{R_{hose}}}{R_{lung}\left(\frac{1}{R_{lung}} + \frac{1}{R_{hose}} + \frac{1}{R_{leak}}\right)} \end{bmatrix}$$

If we denote Y(s) as the output matrix and U(s) as the input matrix, we can write,

$$Y(s) = [C_h(sI - A_h)^{-1}B_h + D_h]u(s)$$

$$\therefore H(s) = \frac{Y(s)}{u(s)} = [C_h(sI - A_h)^{-1}B_h + D_h]$$

The values of the parameters are given below:

Variables	Values
$\boldsymbol{R_{lung}}$	$\frac{5}{1000}$
$\boldsymbol{C_{lung}}$	20
$\boldsymbol{R_{leak}}$	$\frac{60}{1000}$
$\boldsymbol{R_{hose}}$	$\frac{4.5}{1000}$
$\boldsymbol{\omega_n}$	$2\pi 30$

Calculating the values, we get

$A_h = -5.443, B_h = 5.063$

$C_h = \begin{bmatrix} 0.4557 \\ -108.86 \end{bmatrix}, \; D_h = \begin{bmatrix} 0.5063 \\ 101.266 \end{bmatrix}$

$\therefore C_h = (sI - A_h)^{-1} B_h = \begin{bmatrix} \dfrac{2.3072}{s + 5.443} \\ -\dfrac{551.1582}{s + 5.443} \end{bmatrix}$

$\therefore H(s) = C_h \, (sI - A_h)^{-1} B_h + D_h = \begin{bmatrix} \dfrac{0.5063(s + 10)}{s + 5.443} \\ \dfrac{101.266(s + 0.322 \times 10^{-3})}{s + 5.443} \end{bmatrix}$

This is our transfer function.

Question 02:

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

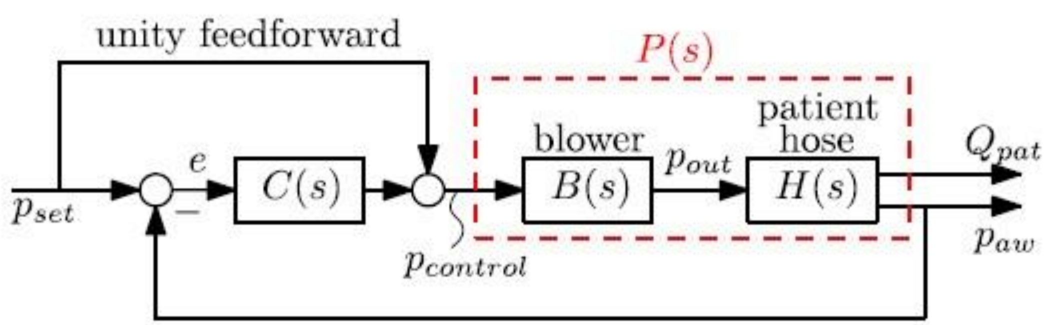


Fig. 4. Closed-loop control scheme with a linear controller $C(s)$.

Answer:

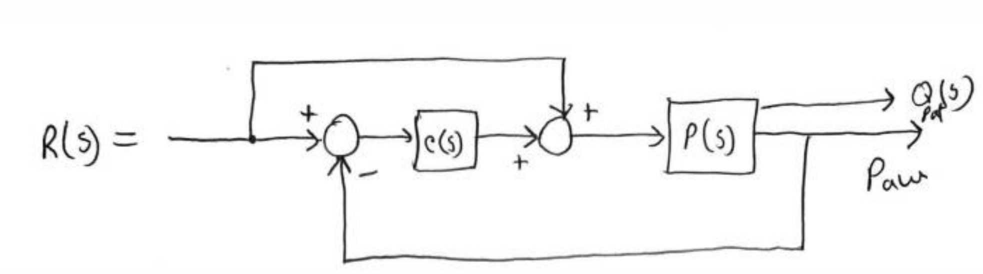
Here, $B(s) = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$

$$\omega_n = 2\pi 30 \ and \ \zeta = 1$$

$$\therefore B(s) = \frac{35530.742}{s^2 + 376.992s + 35530.742}$$

$$Now, P(s) = H(s).B(s) = \begin{bmatrix} \dfrac{0.5063(s+10)}{s+5.443} \\ \dfrac{101.266(s+0.322\times 10^{-3})}{s+5.443} \end{bmatrix} \cdot \frac{35530.742}{s^2 + 376.992s + 35530.742}$$

$$= \begin{bmatrix} \dfrac{17989.21(s+10)}{s^3 + 382.44s^2 + 37582.715s + 193393.83} \\ \dfrac{3598056.119(s+0.322\times 10^{-3})}{s^3 + 382.44s^2 + 37582.715s + 193393.83} \end{bmatrix}$$



Using Mason's rule:

$T_1 = C(s)P(s), T_2 = P(s), L_1 = -C(s)P(s)$

$\therefore \Delta = 1 - L_1 = 1 + C(s)P(s)$

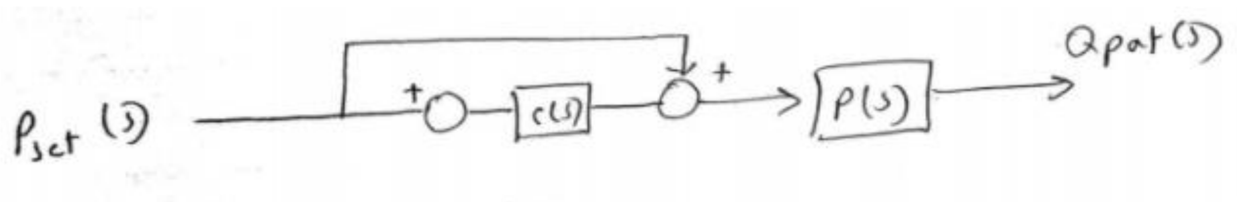
$$\therefore \frac{P_{out}(s)}{R(s)} = T(s) = \frac{P(s)[C(s) +]}{[1 + C(s)P(s)]}$$

$$Here, C(s) = \frac{k_i}{s}$$

Assuming k_i = 1, $C(s) = \frac{1}{s}$

$$T_1(s) = \frac{P_{aw}(s)}{P_{set}(s)} = \frac{P(s)[C(s) + 1]}{1 + C(s)} = \frac{17989.21(s + 10)(s + 1)}{s^4 + 382.44s^3 + 37582.71s^2 + 211383.04s + 179892}$$

For output $Q_{pat}(s)$,



$$\therefore T_2(s) = [1 + C(s)]P(s) = \frac{3598056.119(s + 1)(s + 0.332 \times 10^{-3})}{s^4 + 382.44s^3 + 37582.71s^2 + 193393.83s}$$

Question 03:

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

Answer:

As we can see we have two different transfer functions in this system. But it can be seen from the transfer functions that the root locus related to output, Q_{pat} of the system will have a constant zero and a pole alongside the root locus related to output, P_{aw}. Therefore, we are just showing the root locus related to output, P_{aw} of the system.

MATLAB Code for plotting the root locus:

```
s = tf('s');
sys = (17989.21*(s+10))/((s^4)+(382.44*(s^3))+(37582.71*(s^2))+(193393.83*s))
figure(1);
rlocus(sys)
```

We convert the system into unity feedback system:

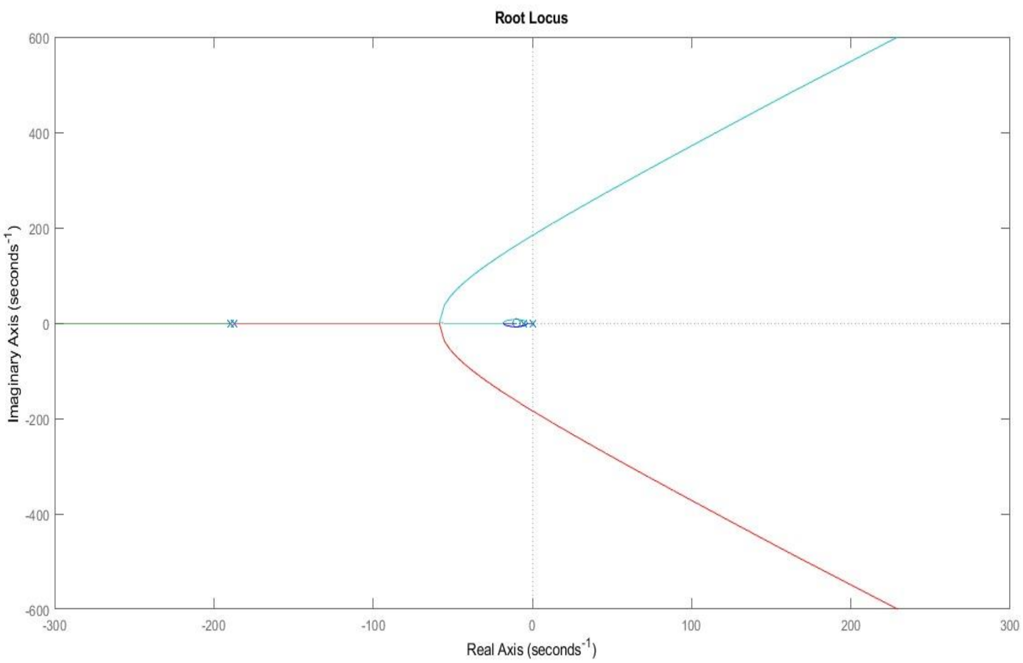
$$G(s) = \frac{P(s)[C(s)+1]}{1-P(s)} = \frac{P(s)\left[\frac{s+k_i}{s}\right]}{1-P(s)}$$

We put the values and calculate to find that

$$G(s) = \left[\frac{\frac{17989.21[s^2+(10+k_i)s+10k_i]}{s^4+382.44s^3+19593.5s^2+13501.73s}}{\frac{3598056.119[s^2+(0.32\times10^{-3}+k_i)s+(0.32\times10^{-3}k_i)]}{s^4+382.44s^3-3560473.41s^2+192235.26s)}}\right]$$

Also, $T(s) = \frac{\frac{P(s)[s+k_i]}{s[1-P(s)]}}{1+\frac{P(s)(s+k_i)}{s[1-P(s)]}} = \frac{\frac{P(s)(s+k_i)}{s}}{1+\frac{k_i P(s)}{s}}$

We need to plot the root locus of P(s)/s


Root Locus
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
600
400
200
0
-200
-400
-600
-300
-200
-100
0
100
200
300

Question 04:

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

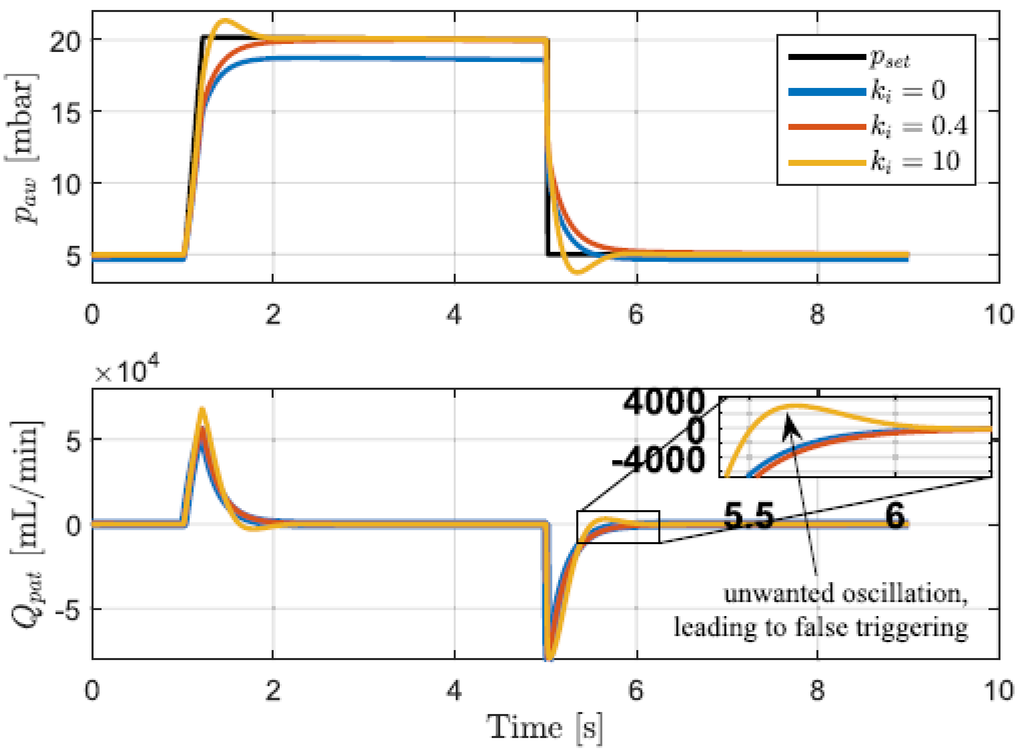
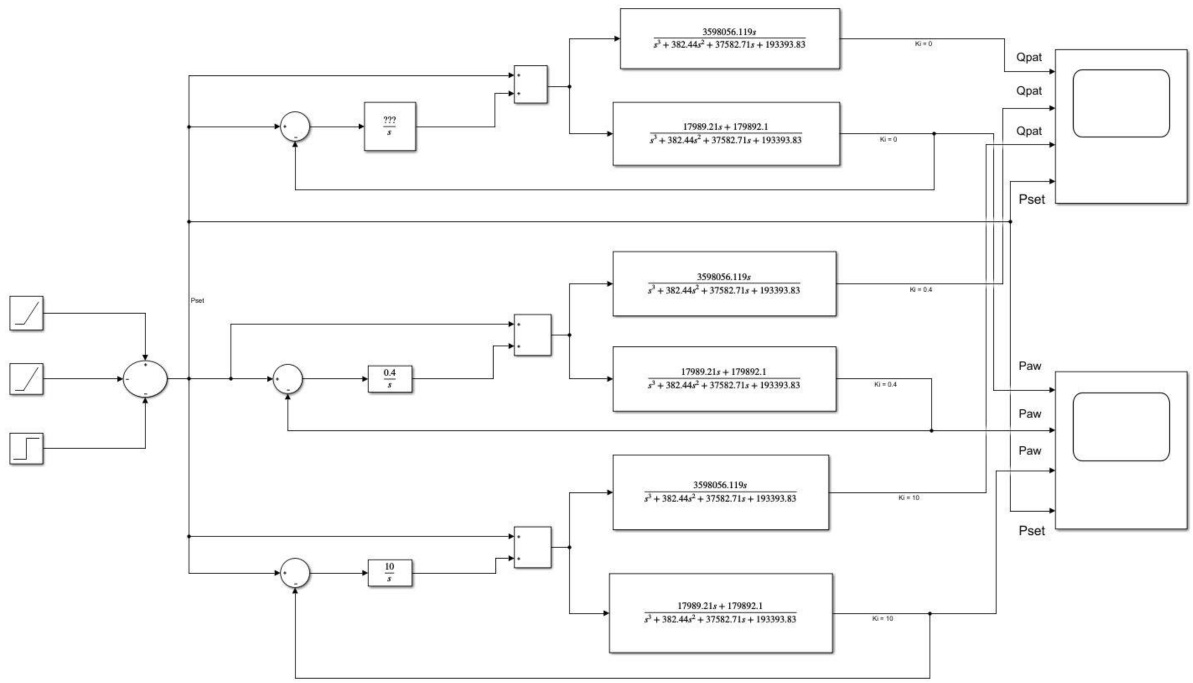


Fig. 7. Simulation result of the closed-loop system using no controller, a low-gain controller ($k_i = 0.4$), and a high-gain controller ($k_i = 10$).

Simulink Model:



Observed Output:

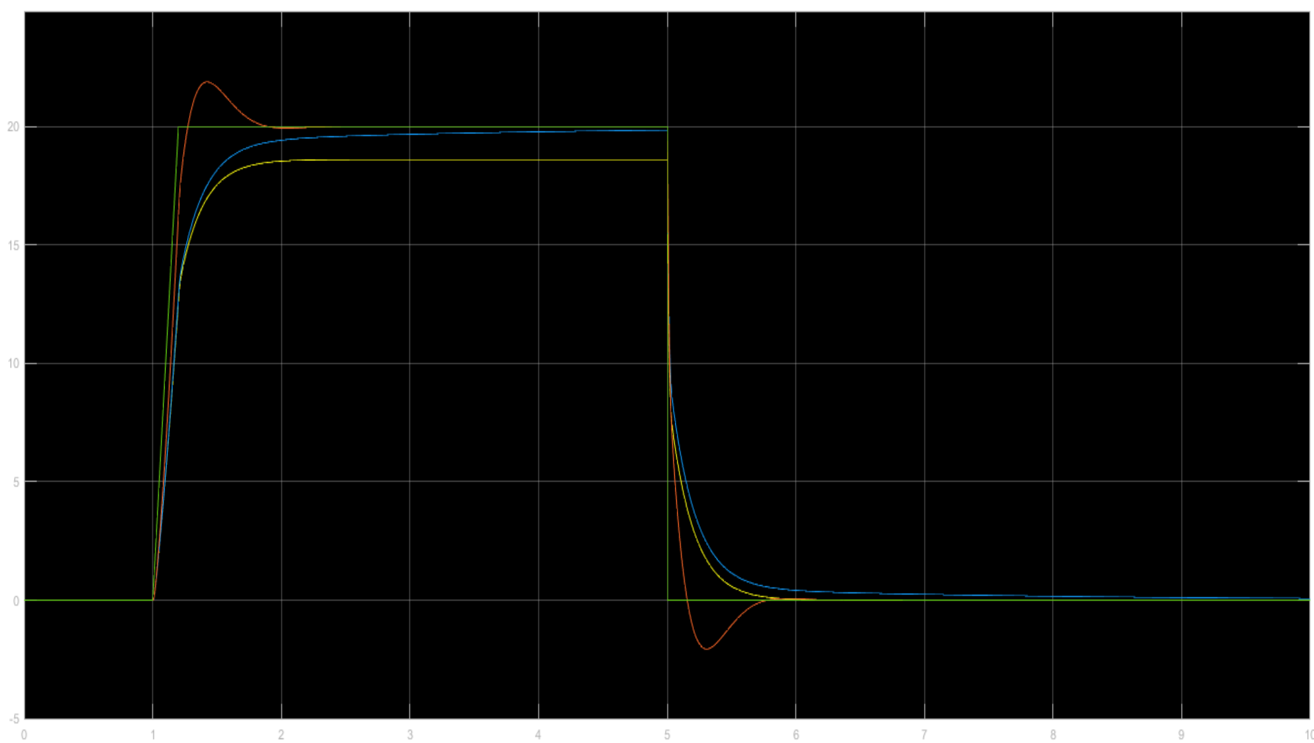


Figure: P_{aw} (mbar) vs time(s) using 3 different values of K_i (0-yellow,0.4-blue,10-red)

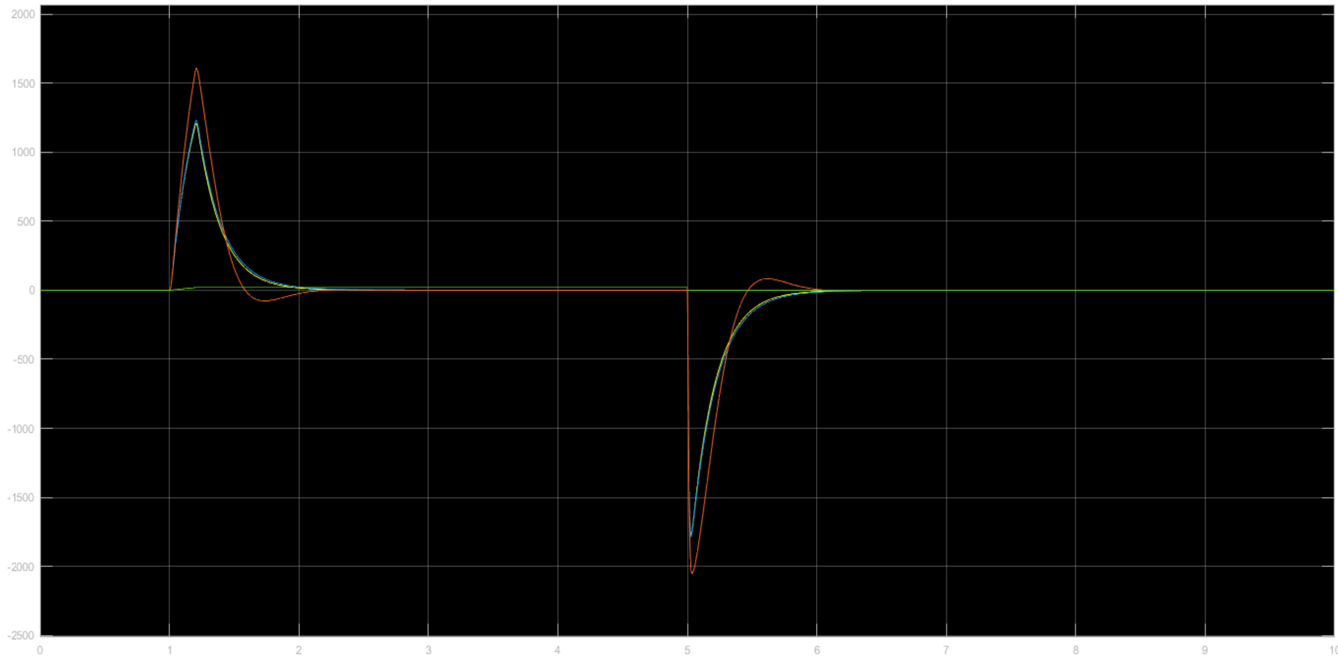


Figure: Q_{pat} (mL/s) vs time (s) using 3 different values of K_i (0-yellow,0.4-blue,10-red)

Here we have shown three different plots for integral controller where the controllers gain parameter ki is changed.

For ki = 0 we have Open-loop control or C(s) = 0. This simulation result shows that without feedback the system cannot reach the target pressure specified.

For ki = 0.4 we have a low-gain integral feedback controller C(s). As expected, T_R is low for this controller compared to other designs, so the response is slower. But the low gain induces zero overshoot. Overshoot has to be minimized in our system in order to avoid false inspiration triggers.

For ki = 10 we have a high-gain integral feedback controller C(s). The high-gain controller induces an unwanted oscillation in the patient flow. Although the system reaches its target pressure rapidly, the unwanted high value of overshoot cannot be ignored. Because high value of overshoot may result in false patient flow triggering.

To illustrate the importance of both feedback & feedforward control, we will simulate the system without any one of them. The impacts are clearly shown on the observed outputs.

Observed Output without Feedback:

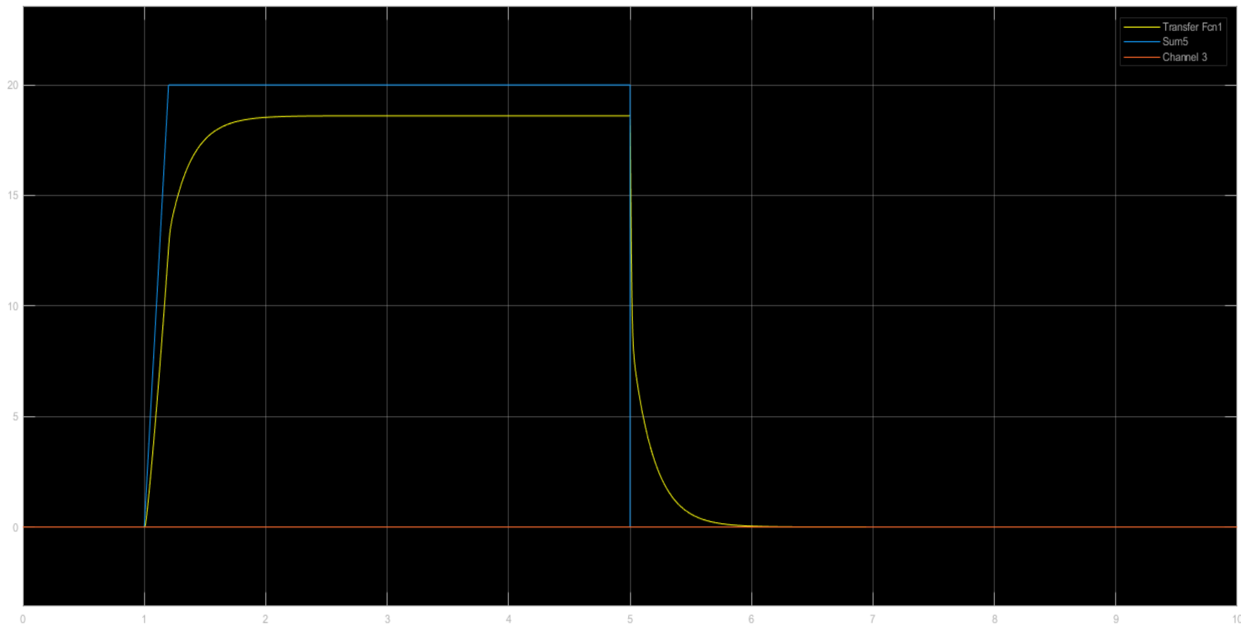


Figure: P_{aw} vs time without feedback

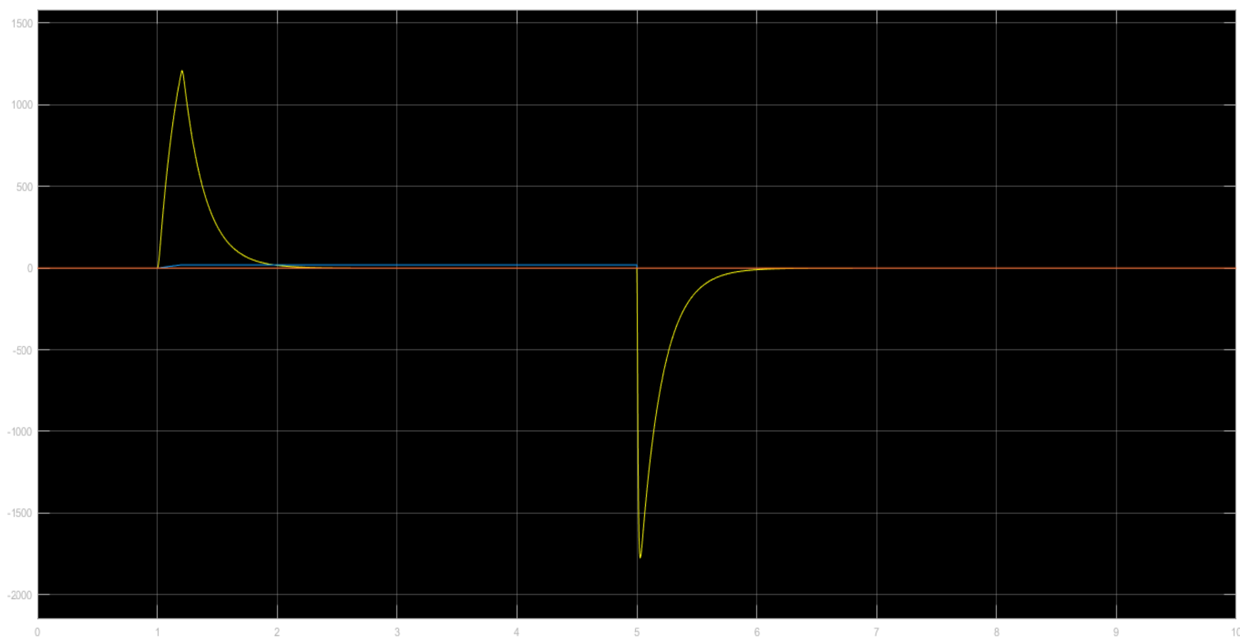


Figure: Q_{pat} vs time without feedback

For simulating the system without any feedback, we have set ki = 0. So the system becomes an open loop system without any feedback.

The simulation result clearly shows why we need a feedback controller for this system. The goal of the system is to achieve sufficiently fast pressure buildup and accurate tracking of the desired pressure profile pset. However, without feedback a steady state error always persists.

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. But a pressure drop pout – paw exists along the hose and for the variations across the blower characteristics. As a result, the system without any feedback cannot track the desired steady state pset value accurately.

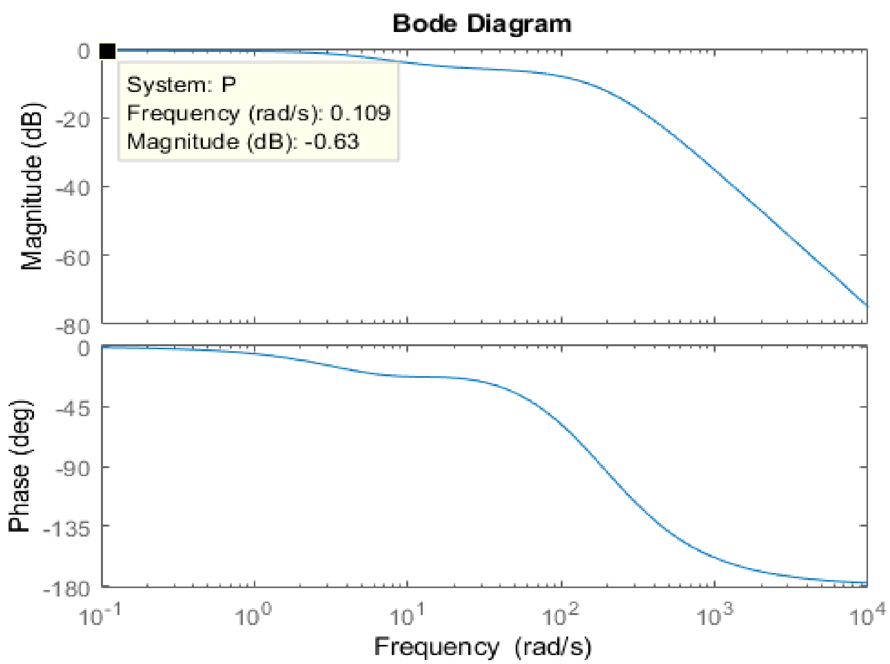


Figure: Bode plot of the plant

The requirement of a feedback control can be further illustrated using the bode plot of the plant. It can be seen that P (ω = 0) < 1 (in DB P (ω = 0) < 0). This phenomenon is due to the pressure drop along the hose. Given a constant pressure $P_{control}$, there will be a leakage flow Q_{leak} through the leakage hole in the hose, which results in a pressure drop along the hose. So $P_{aw}/P_{control} < 1$ in steady state

Observed Output without Feedforward Control:

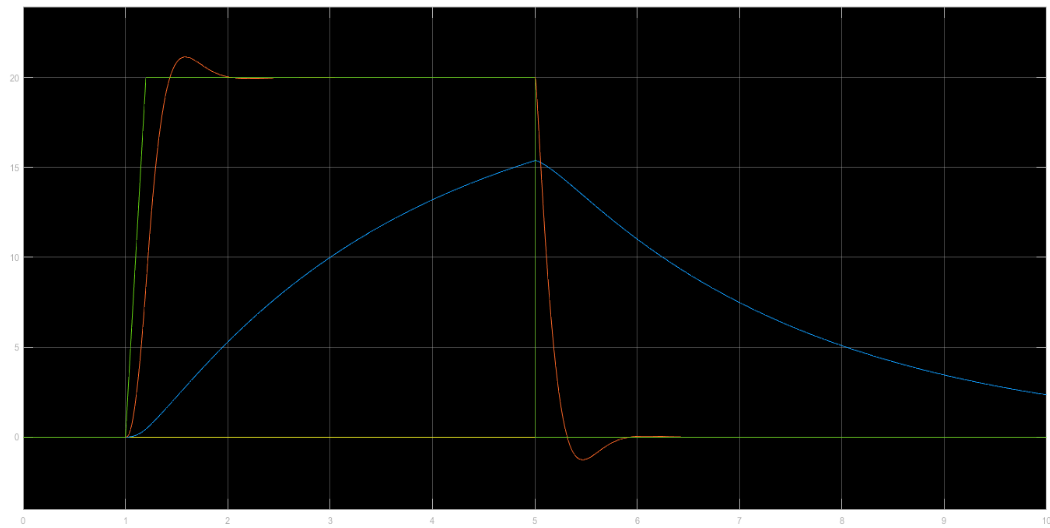


Figure: P_{aw} vs time without feedforward (0-yellow,0.4-blue,10-red)

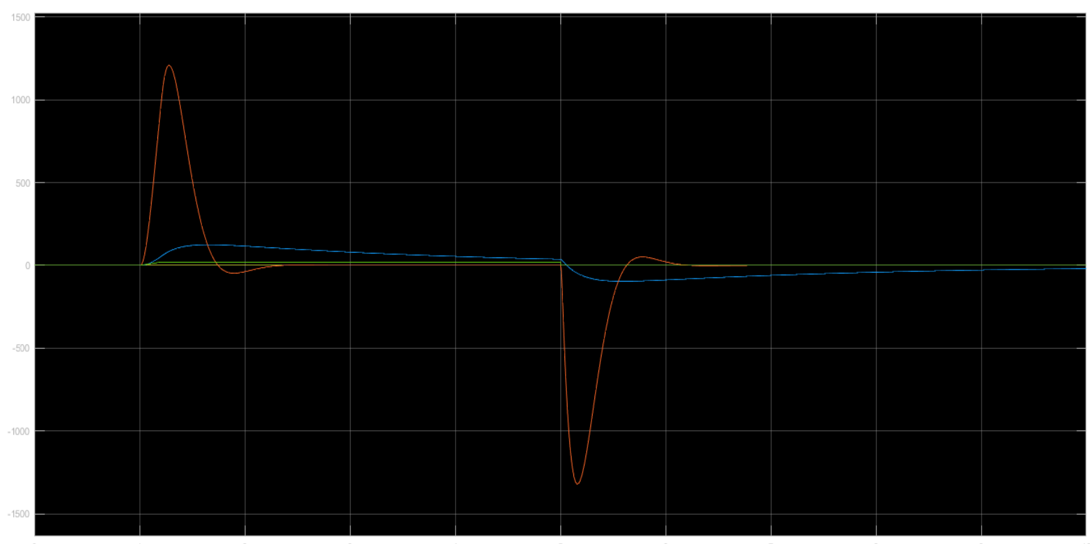


Figure: Q_{pat} vs time without feedback (0-yellow,0.4-blue,10-red)

These simulation results indicate that without the unity feed forward, output pressure pout cannot accurately track the target pressure pset. The rise time from 10% to 90% of a pressure set point should be approximately 200 ms. For ki = 0 and 0.4 specifications cannot be met. For ki = 10 we get better result to some extent, but the overshoot is higher. 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. So, the unity feed forward is necessary.

Question 05:

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

To illustrate the effects of using three different type controllers, we are simulating our Simulink model with an ideal integrator, a PI and a lag controller. The different effects of the controllers can be observed in the outputs.

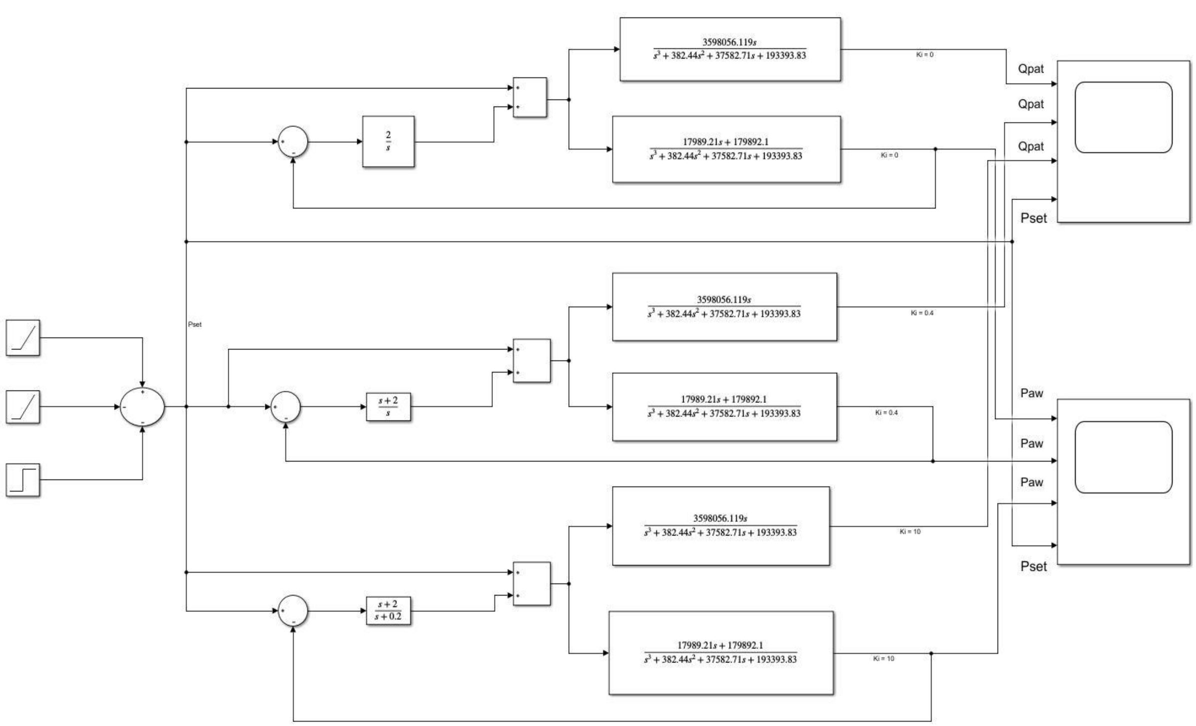
Here. We have used:

Ideal integrator controller, $C_1(s) = 2/s$

PI controller, $C_2(s) = (s+2)/s$

Lag controller, $C_3(s) = (s+2)/(s+0.2)$

Simulink Model:



Observed Output:

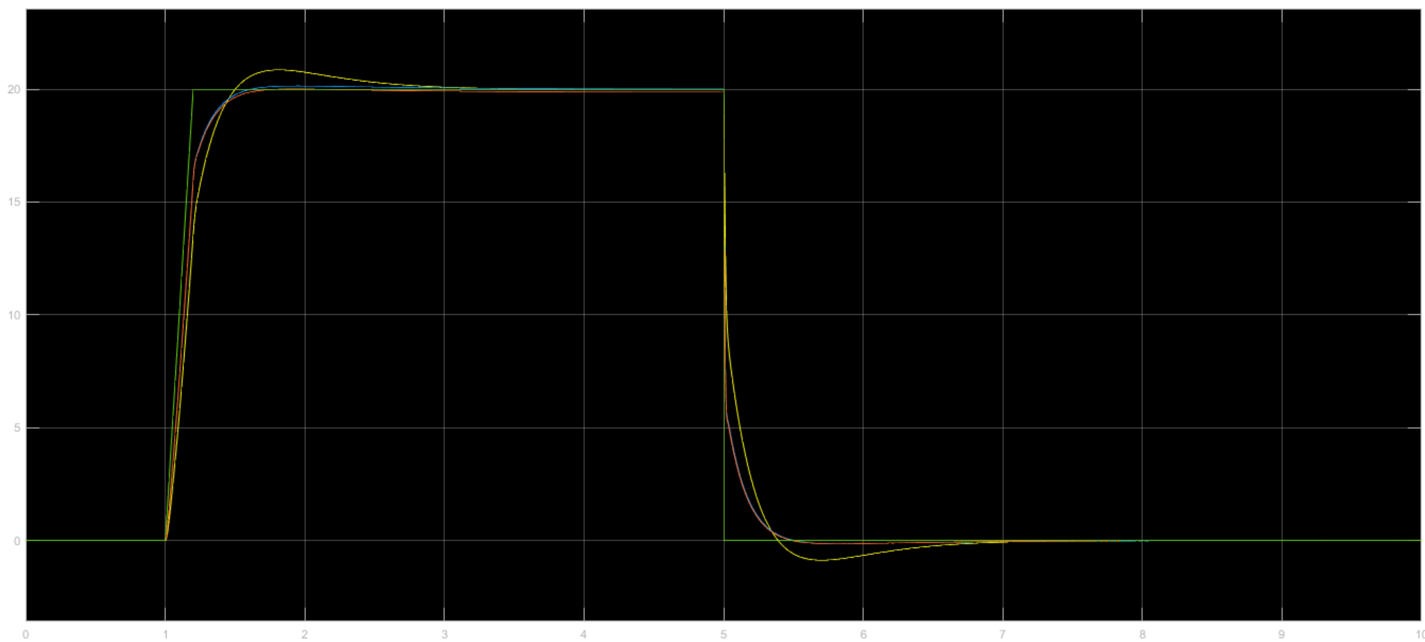


Figure: P_{aw} vs time(yellow – Ideal integrator, blue – PI, red – lag)

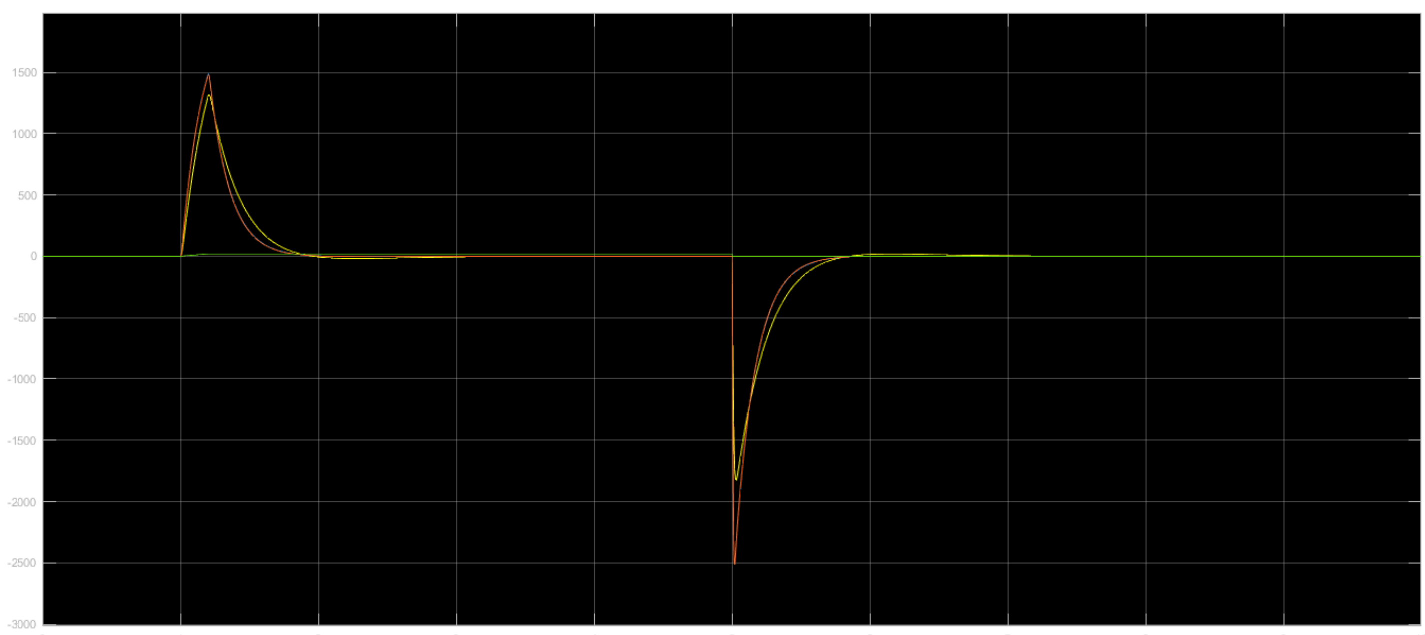


Figure: Q_{pat} vs time(yellow – Ideal integrator, blue – PI, red – lag)

Controller	Rise Time(ms)	Overshoot(mL/s)
Ideal integrator	298.43	18.85
PI	243.72	2.985
Lag	240.61	2.978

From the graphs, we can see that the rise time of the step response for PI and Lag controller is lower than the ideal integrator. Although, all of them are outside the specified range. The overshoot is also lower for PI and Lag controller and all of them are inside the specified range.

But the problem with using PI or Lag controller is that this plant requires a controller that can ensure low frequency disturbance supression, high-frequency roll off and a stabilizing -1 slope across the bandwidth of the system.

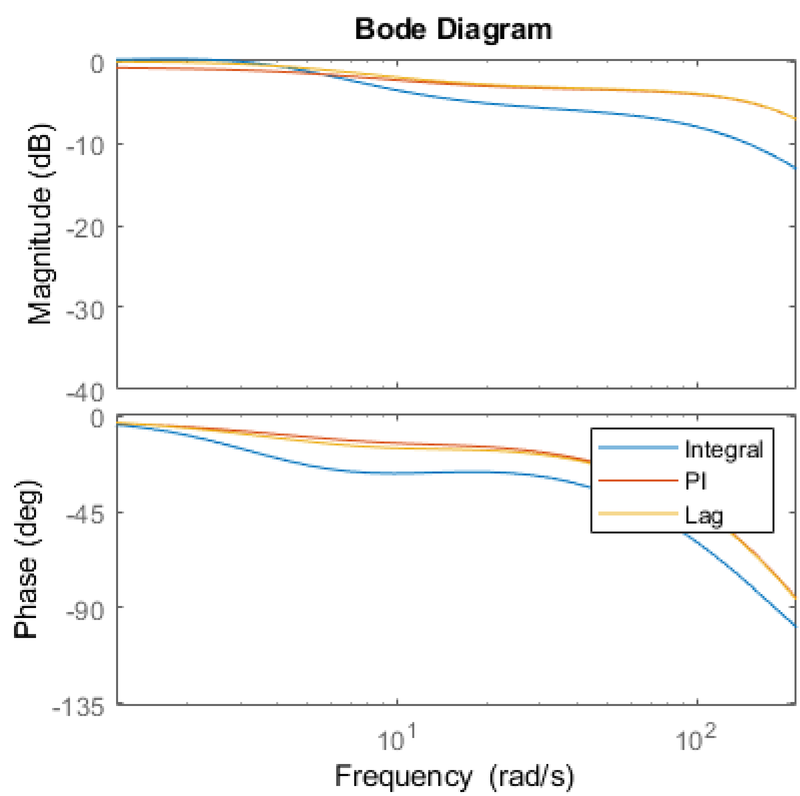


Figure: Bode plot of the Close Loop System

From the bode plot we can state that for the integral controller we get the highest amount of roll off for frequencies outside the bandwidth. The PI and Lag controller introduces unnecessary less roll off for frequency above the bandwidth. Therefore, they are not considered desirable particularly for this plant.

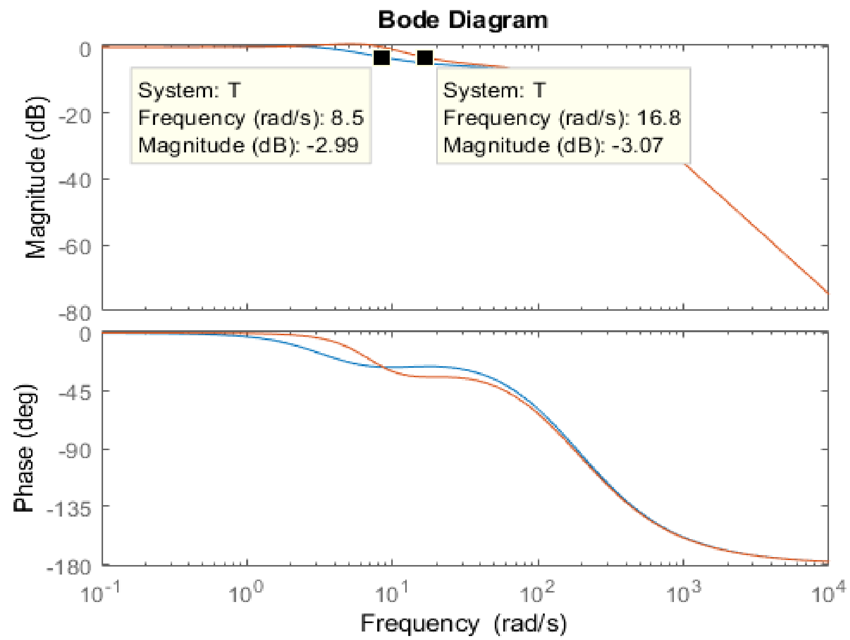


Figure: Bode plot for Integral Control for ki = 2(blue line) and ki = 10(red line). Here BW changes from 8.5 Hz to 16.8 Hz

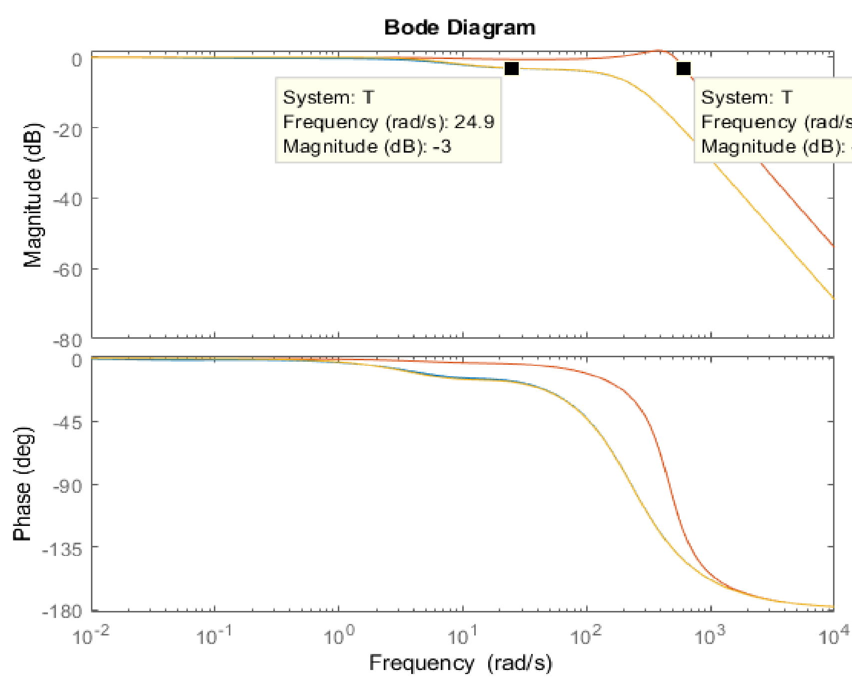
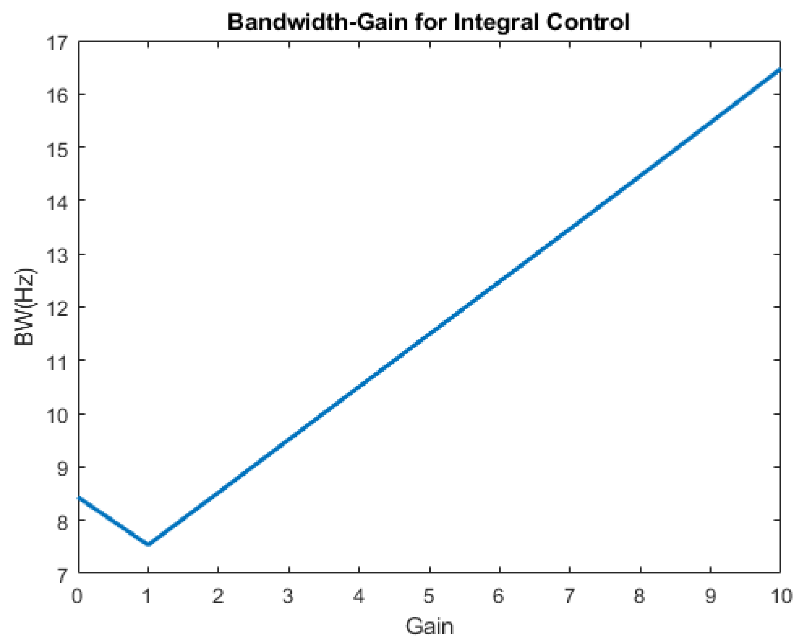
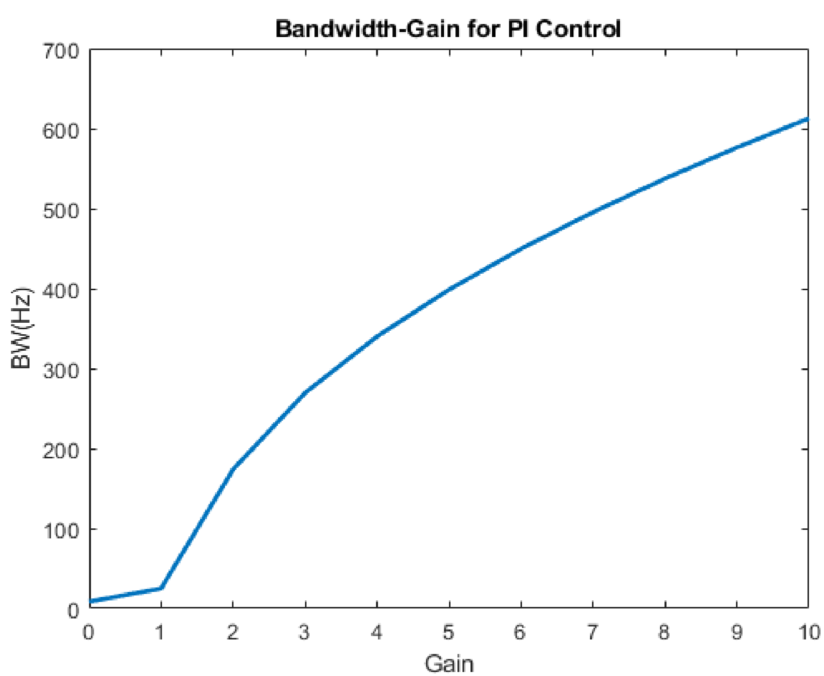
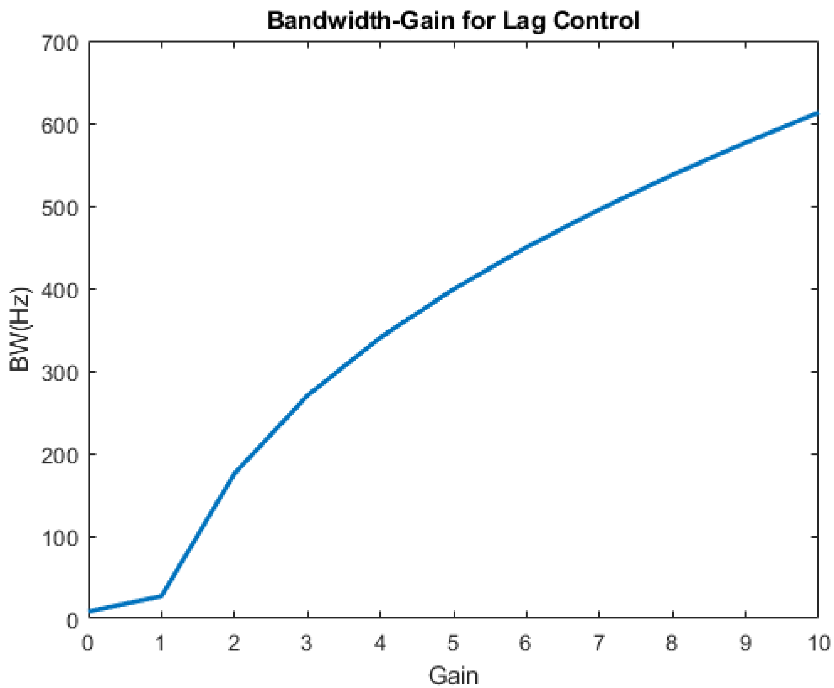


Figure: Bode plot for PI Control for ki = 2(blue line) and ki = 10(red line). Here BW changes from 25 Hz to 612 Hz

The need for high amount of roll off for frequencies above bandwidth can be explained using the figures. When we change the gain of our controller for both Integral and PI controller, bandwidth increase for PI control is much higher than Integral control. Same can be demonstrated for Lag controller also.





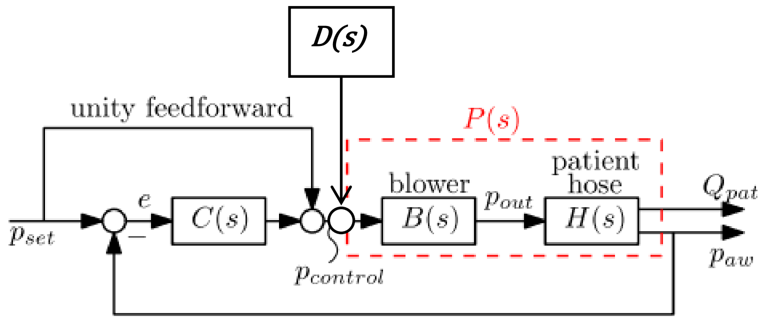


From the above plot we can see that for integral control bandwidth changes linearly with gain.

But for PI and lag controller Bandwidth vs Gain follows a nonlinear trend. For low gain BW increases rapidly. This trend is due to the less roll off of PI and Lag controller.

As the gain of the controller is changed, bandwidth changes. This changes have effect on Ts and Tr values. Due to rapid change in bandwidth for PI and Lag controller rise time changes abruptly, which has an adverse effect on the robustness of the system. The soul purpose of controller design is to ensure that the specifications are met in all scenarios. So we prefer Integral controller for the robustness of the system.

For analysis of disturbance input we have considered the following system



Transfer function for disturbance input = $\dfrac{B(s)H(s)}{1+C(s)B(s)H(s)} = \dfrac{P(s))}{1+C(s)P(s)}$

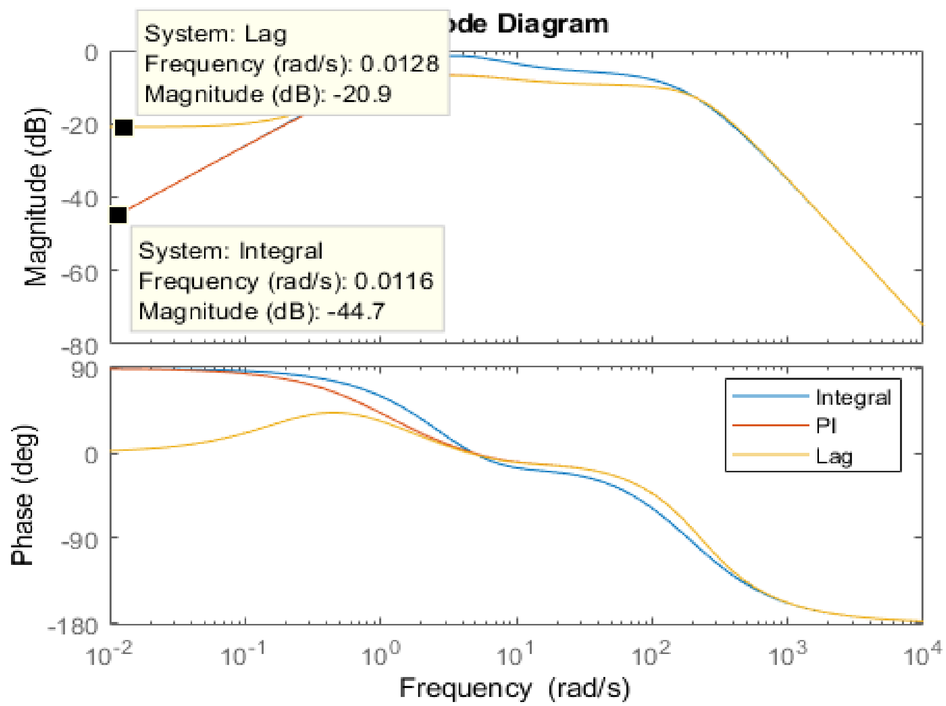


Figure: Bode plot for disturbance input transfer function

From the frequency response of the disturbance input we can state that for Lag controller low frequency disturbance is much higher than PI or Integral controller. Our feedback control system is supposed to compensate for the disterbances or unwanted inputs that enter the system. Integral controller is a suitable controller type for this system since it results in low-frequency disturbance suppression.

Question 06:

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

Answer:

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.

Here, three linear controllers are designed which meet the specifications. Each of them has some advantages and some drawbacks. So, there is a trade off between all of the controllers designed.

Ideal Integral controller (C1):

We used an ideal integrator controller, $C(s) = \frac{3}{s}$. By using a gain K_i = 3, we get the best response among all the ideal integrator controllers. Still, all the specifications are not exactly met.

Lag controller (C2):

We have chosen a lag controller $C(s) = \frac{s+8}{s+0.1}$ for this purpose. The rise time of the step response is 185.11 ms and the overshoot is 25mL/s. The plateau pressure is also less than the desired value. Therefore, this controller meets all the three specifications stated.

PID controller with noise reduction filter (C3):

We designed a third filter with $C(s) = P + I\frac{1}{s} + D\frac{N}{1+\frac{N}{s}}$.

The values of the parameters used are:

Proportional gain, P =14.075

Integral gain, I = 1286.53

Derivative gain, D = 0.0384

Filter coefficient, N = 78915.12

We used the PID controller block of simulink to tune the parameter values and achieve our final values which give the best reponse for our purpose. This PID controller has a Low pass filter in series with the Derivative gain block to reduce system noise.

Simulink Model:

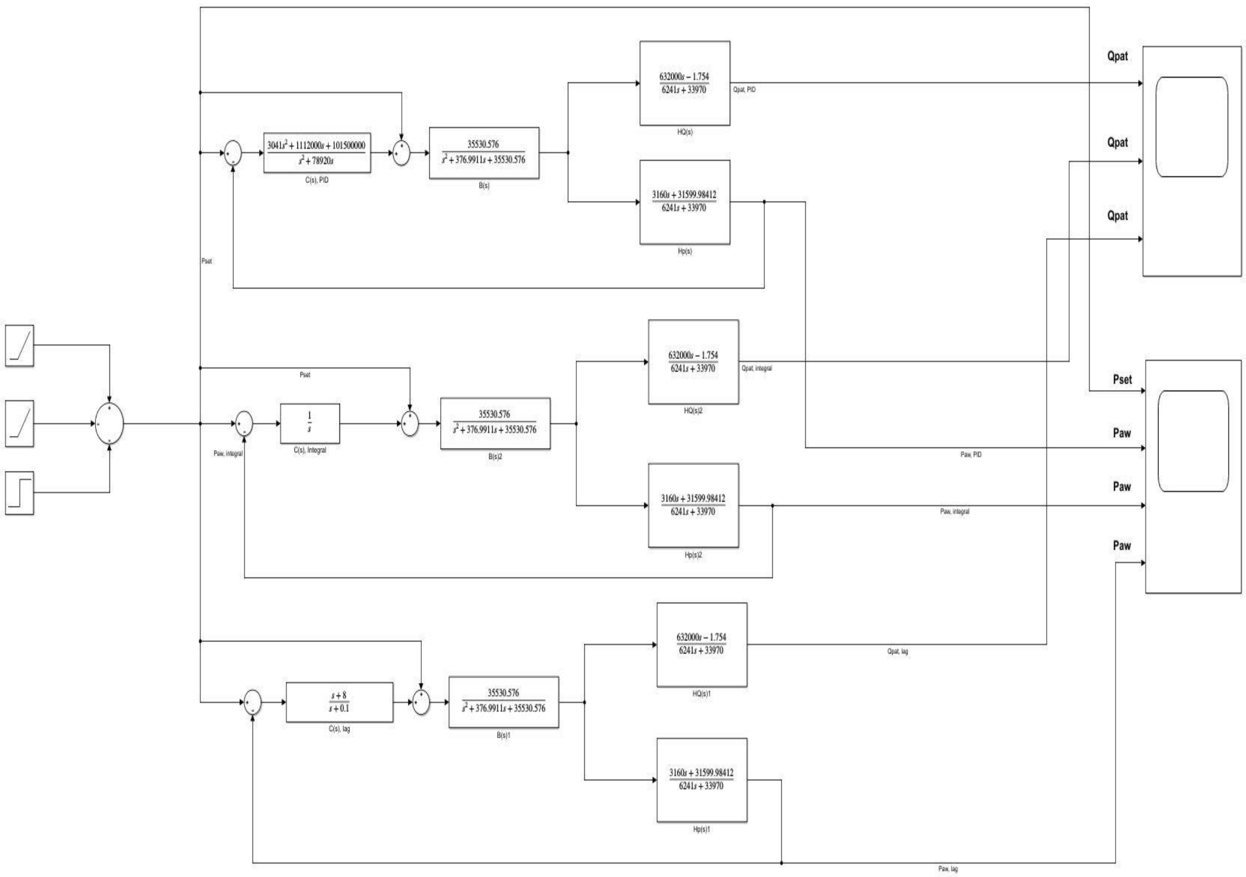


Table with with system responses for different controllers:

	Controller Type		
Transient response	**PID**	**Integral**	**Lag**
Rise Time T_r, (ms)	**159.416**	**260.226**	**185.11**
Plateau Pressure (mbar)	**0**	**0.02**	**-0.02**
Overshoot (mL/s)	**0**	**35**	**25**

From the table we can see that both the PID and lag controller meet the rise time specification, but the integral control does not by a long margin. All the controllers meet the plateu pressure specifications. The lag controller introduces some overshoot for air flow response Q_{pat}, but it is within the acceptable range. The PID and Integral give no overshoot. Therefore, both the PID and lag controllers can be applied here, but the PID controller is a better choice.

Output for pressure P_{aw} :

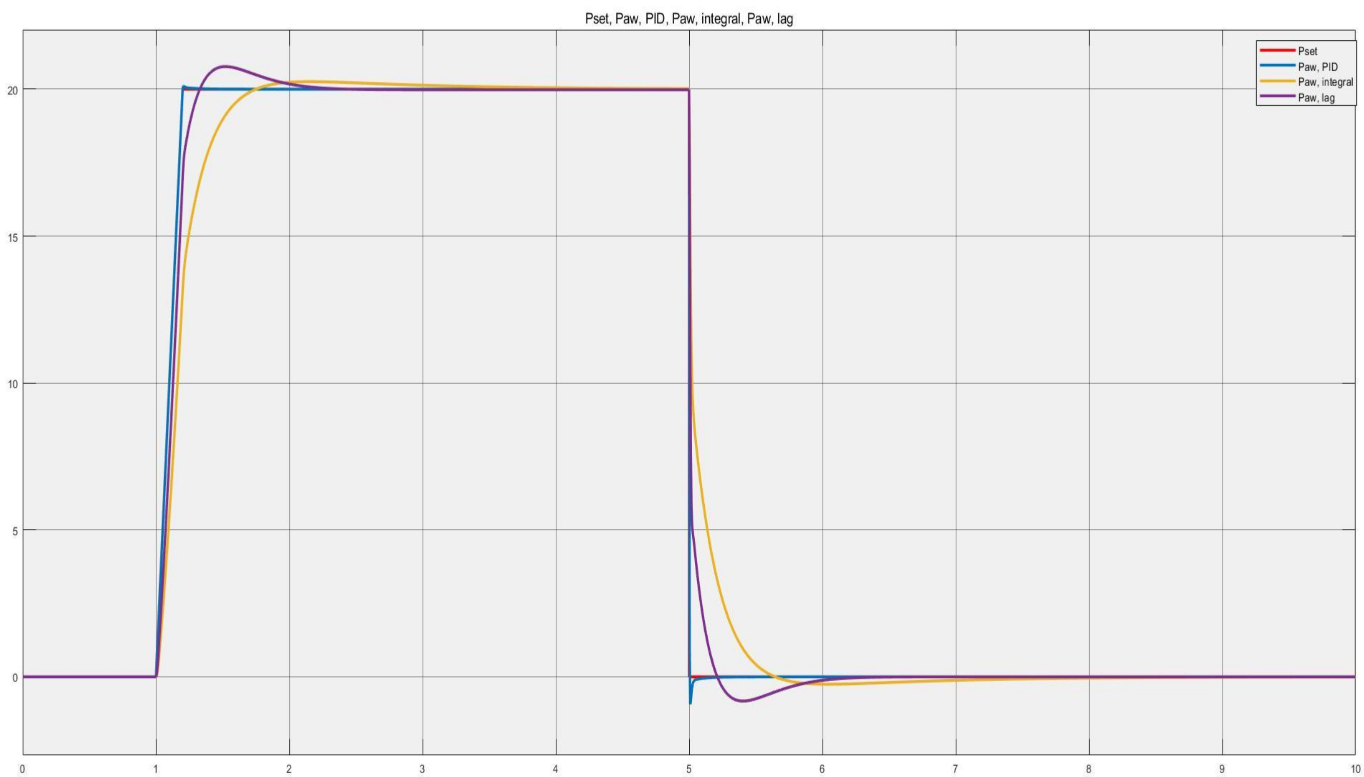


Figure : P_{aw} vs time (C1-yellow, C2-purple, C3-blue)

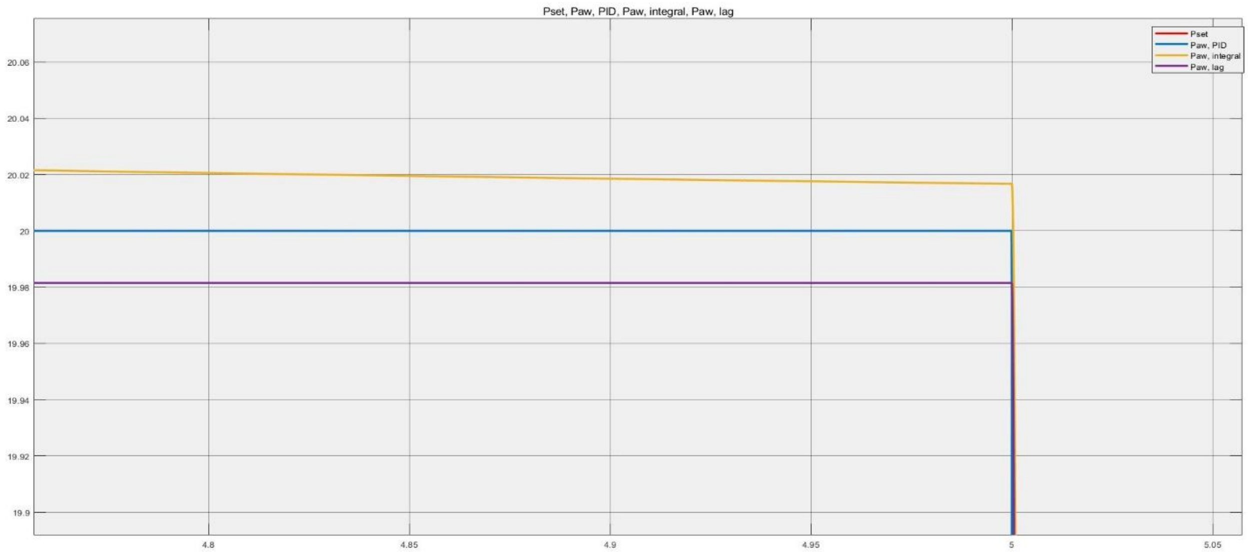


Figure: Zoomed in version of P_{aw} vs time showing the plateu pressure (color scheme same as previous)

Output for flow Q_{pat} :

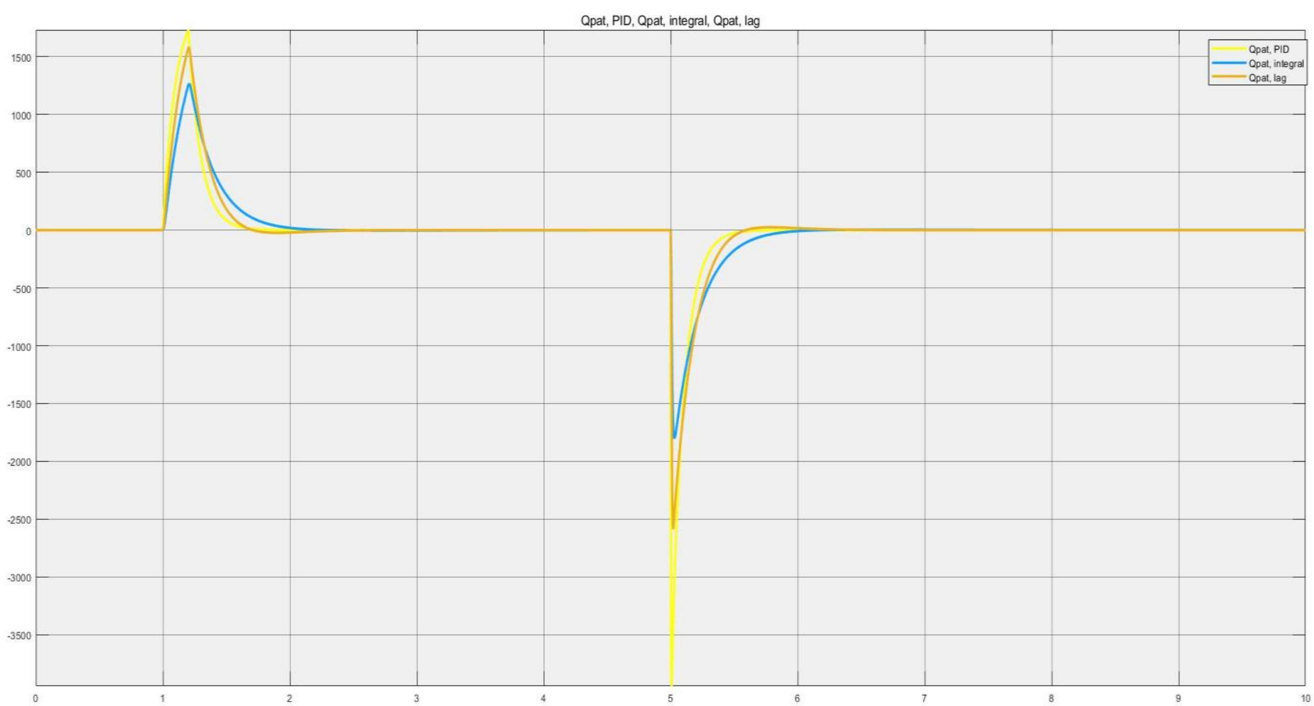


Figure: Q_{pat} vs time (C1-blue, C2-brown, C3-yellow)

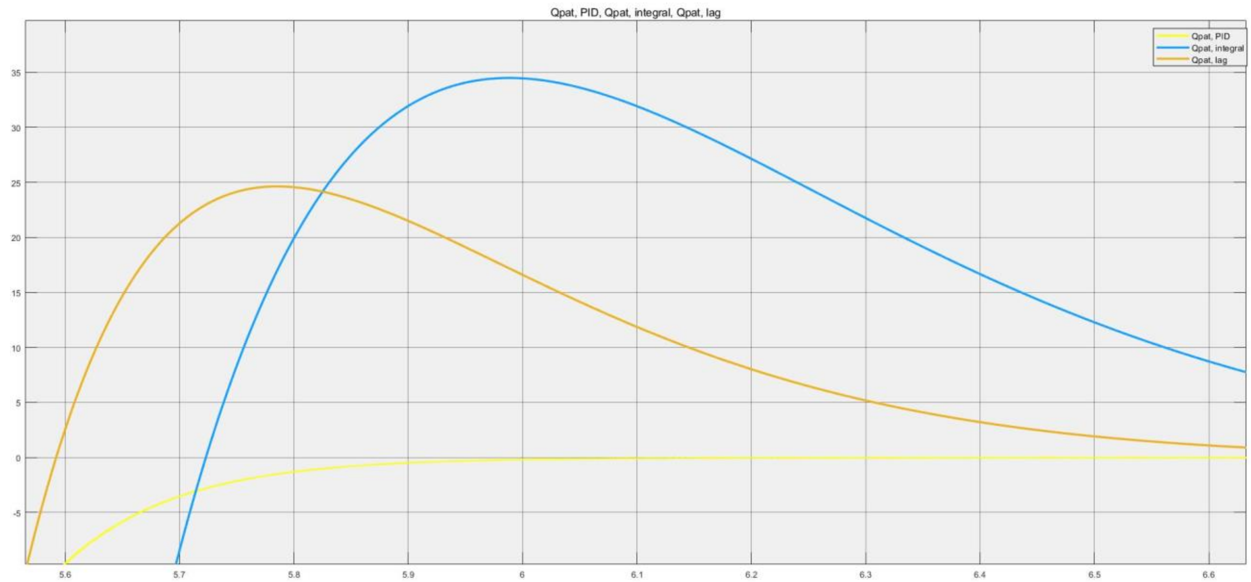


Figure: Zoomed in version of Q_{pat} vs time showing the overshoot (color scheme same as previous)

From the above simulation results we can see that the PID controller gives the best response for our purpose with Lag controller second, and the Integral controller last (it does not meet the rise time specifications).

Question 07:

Reproduce the results shown in Fig. 14 for linear and variable gain controllers. What are the pros and cons of nonlinear control over linear control?

Answer:

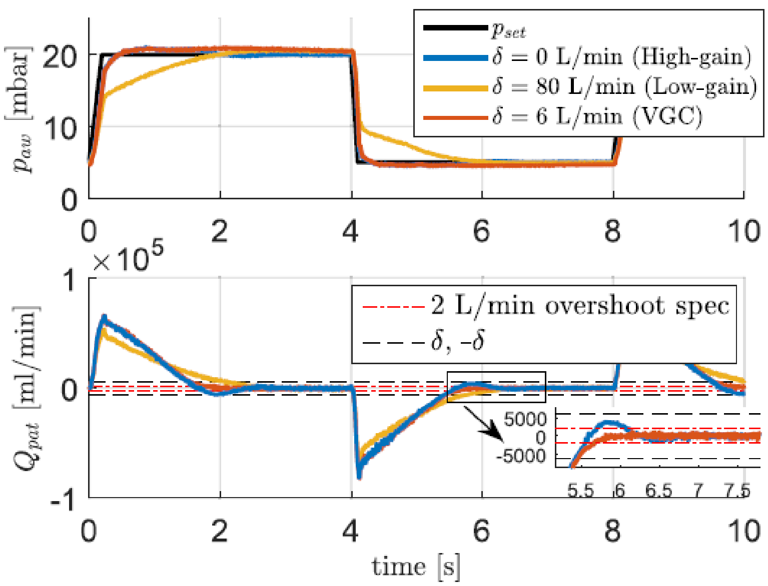


Fig. 14. Experimental time-domain response of the linear controllers and a variable-gain controller with $\delta = 6$ L/min. Note that $\delta = 0$ represents the linear high-gain controller and $\delta = 80$ L/min represents the linear low-gain controller.

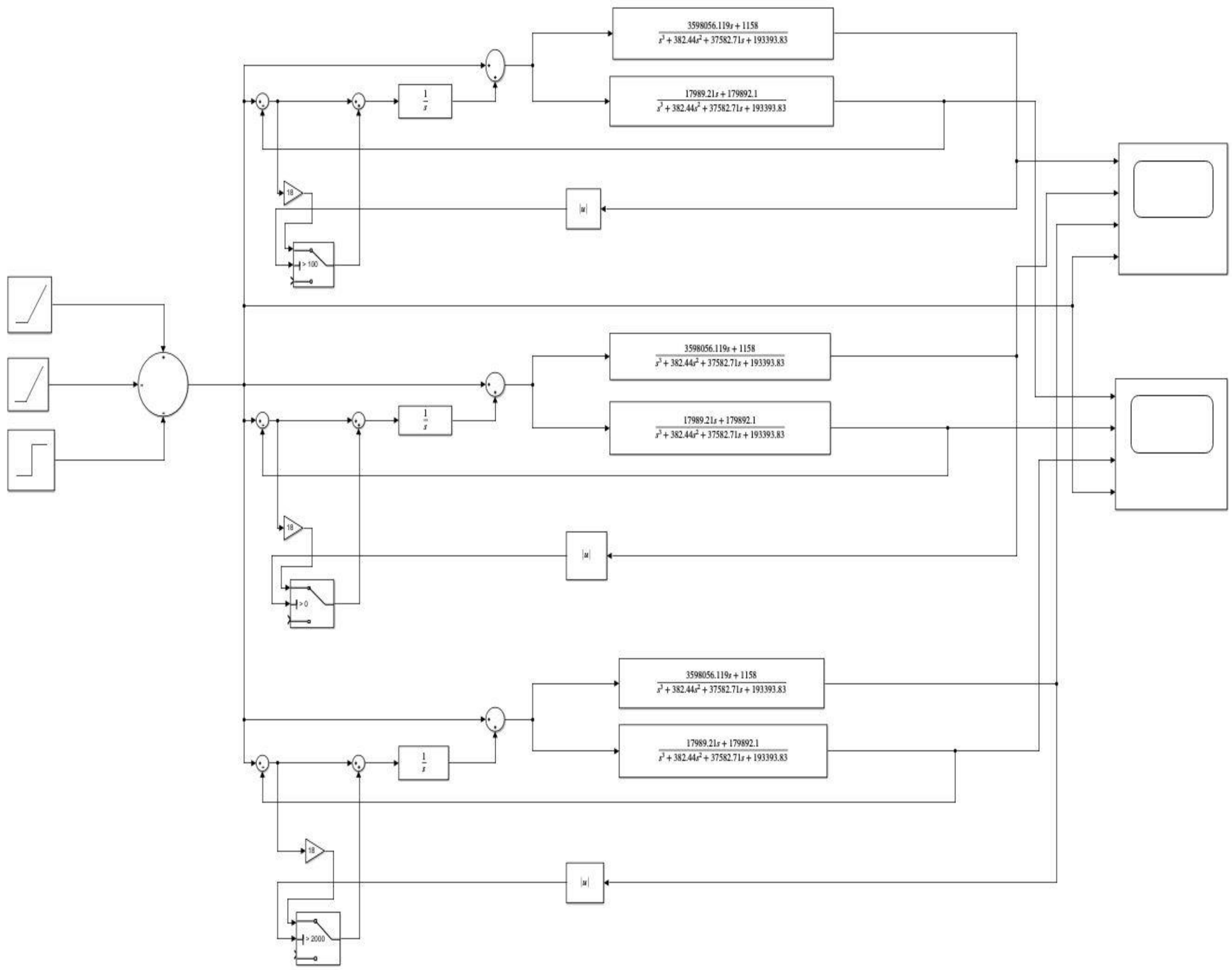
For simulation, we have used 3 different values of the switch threshold, δ = 0,100,2000 mL/sec.

Here, when δ = 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.

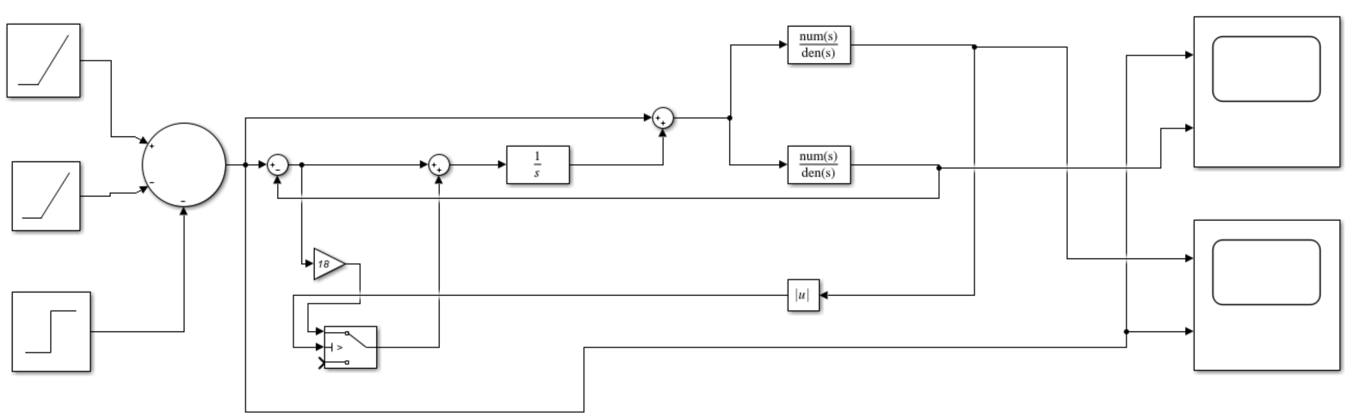
When δ = 2000, we are operating with a low gain controller. Thus, the step response for this case should have low overshoot value but high time.

When δ = 100, we are operating with a variable gain controller. For example, when the switch is triggered, the gain of the controller is 19 for a fixed controller, 1/s and so operating on high gain. On the other hand, when the switch is not triggered, the gain of the controller is 1 and so operating on low gain. Therefore, the step response for this case should have high rise time and low overshoot value. This is observed on the output graphs below.

Simulink Model:



Zoomed in view:



Observed Output:

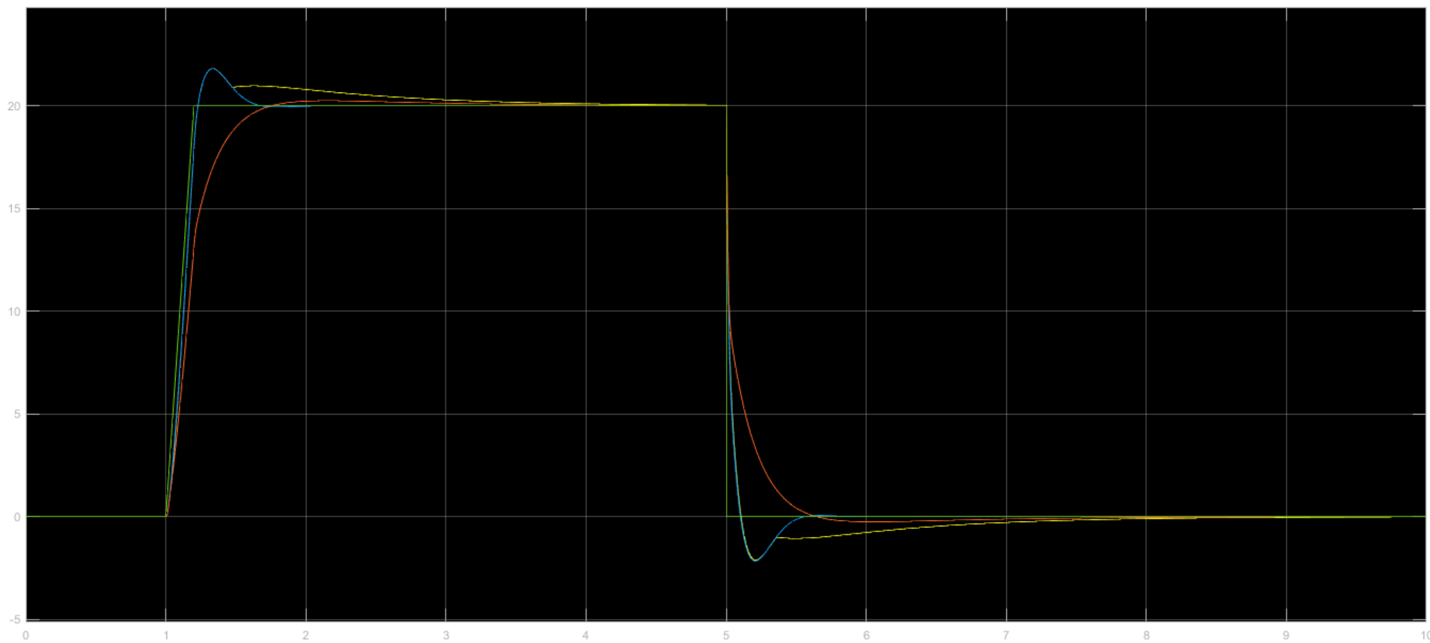


Figure: P_{aw} vs time(yellow-VGC controller,blue-high gain controller,red-low gain controller)

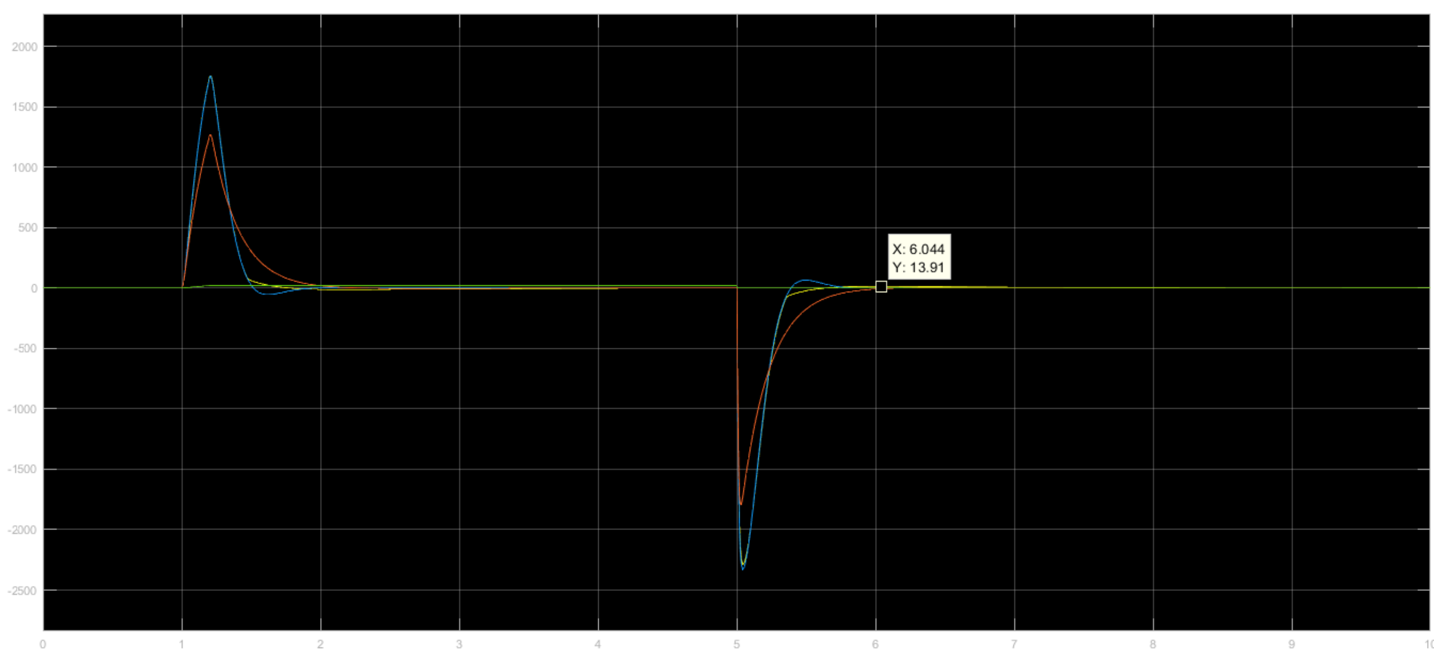


Figure: Q_{pat} vs time(yellow-VGC controller,blue-high gain controller,red-low gain controller)

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, R_{lung} 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 k_i that satisfies both rise time and overshoot specifications. K_i cannot be set to be 2 due to the fact that a slightly larger k_i will cause a massive overshoot while a slightly lower k_i 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 Q_{pat} 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.

Question 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} and system parameter R_{leak}. We did not change the parameter R_{hose} because we consider that we will use the same air-flow pipe. We did not change ω_n because we consider we will use the same blower.

Original System parameters:

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

System response for original system parameters:

	Controller Type		
Transient response	**Non-linear ($\delta = 100$)**	**PID**	**Integral**
Rise Time T_r, (ms)	**158.553**	**159.416**	**260.241**
Plateau Pressure (mbar)	**0.04**	**0**	**0.02**
Overshoot (mL/s)	**14**	**0**	**35**

MATLAB code:

We used the following MATLAB code to calculate the system transfer function for each set of parameter values.

Code:

```
clear all;
close all;
clc;

R_lung = 1/5000;
C_lung = 100;
R_leak = 60/1000;
R_hose = 4.5/1000;
omega = 2*pi*30;
zeta = 1;

Ah = -
((1/R_hose)+(1/R_leak))/(R_lung*C_lung*((1/R_lung)+(1/R_hose)+(1/R_leak)))
Bh = (1/R_hose)/(R_lung*C_lung*((1/R_lung)+(1/R_hose)+(1/R_leak)))
Ch = [(1/R_lung)/((1/R_lung)+(1/R_hose)+(1/R_leak)) ; ...
    -((1/R_hose)+(1/R_leak))/(R_lung*((1/R_lung)+(1/R_hose)+(1/R_leak)))]
Dh = [(1/R_hose)/((1/R_lung)+(1/R_hose)+(1/R_leak)) ; ...
    (1/R_hose)/(R_lung*((1/R_lung)+(1/R_hose)+(1/R_leak)))]

s = tf('s');
H = Ch*((s-Ah)^-1)*Bh + Dh

B = (omega^2)/(s^2 + 2*zeta*omega*s + omega^2)

P = H*B
```

Output:

```
P =

  From input to output...
                  1507 s + 7.536e04
   1:  --------------------------------------
       s^3 + 379.3 s^2 + 3.639e04 s + 8.101e04

                     7.536e06 s
   2:  --------------------------------------
       s^3 + 379.3 s^2 + 3.639e04 s + 8.101e04

Continuous-time transfer function.
```

Simulink models:

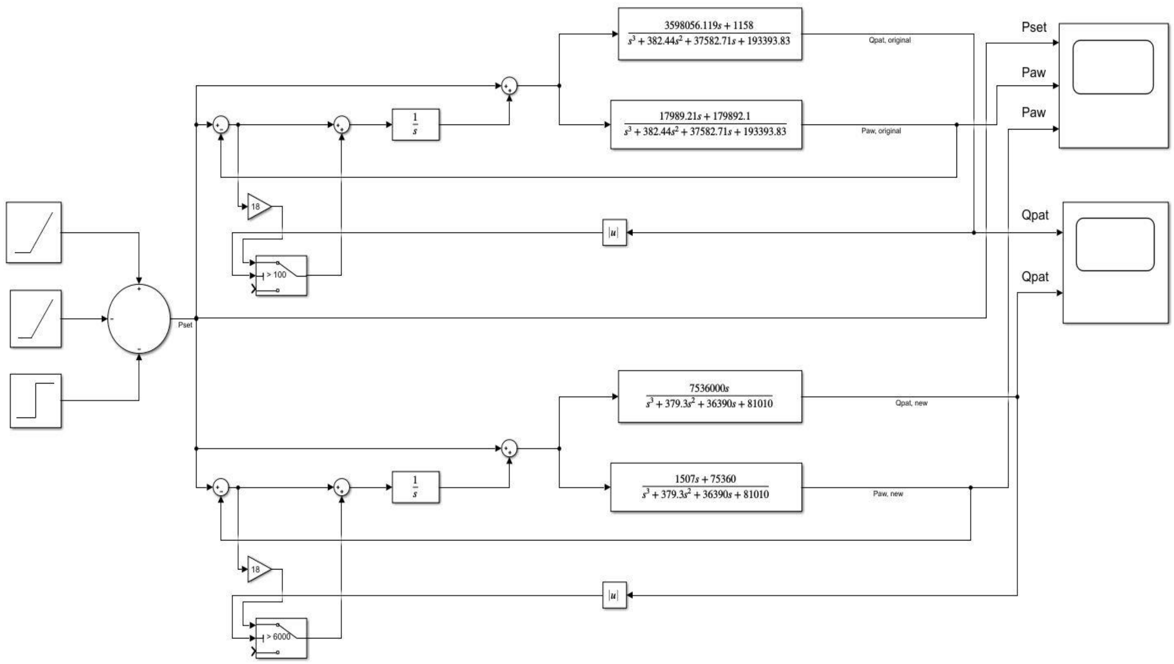


Figure: Non-linear controller (top one is original system and bottom one is system with changed parameters)

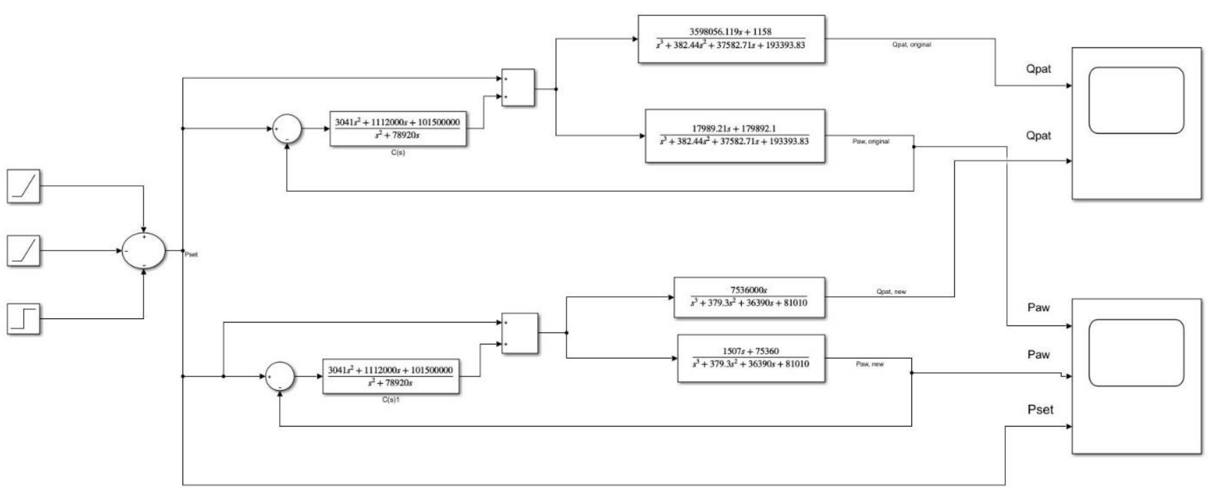


Figure: PID controller (top one is original system and bottom one is system with changed parameters)

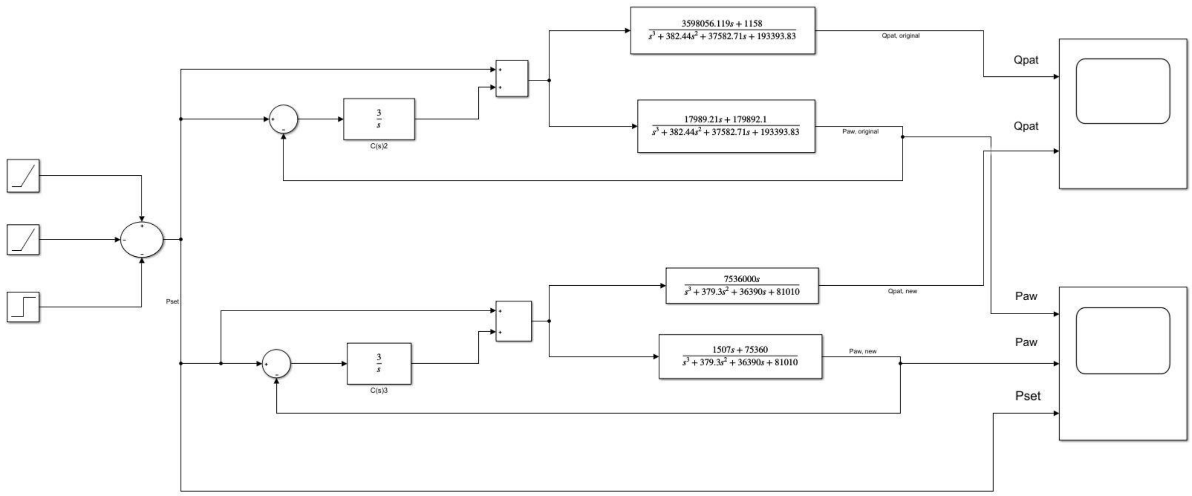


Figure: Lag controller (top one is original system and bottom one is system with changed parameters)

We check the robustness of the system for change in different system parameters for each of the three types of controllers used.

1. R_{lung} is changed to $\frac{25}{1000}$ and C_{lung} is changed to 50. The rest of the parameters are unchanged.

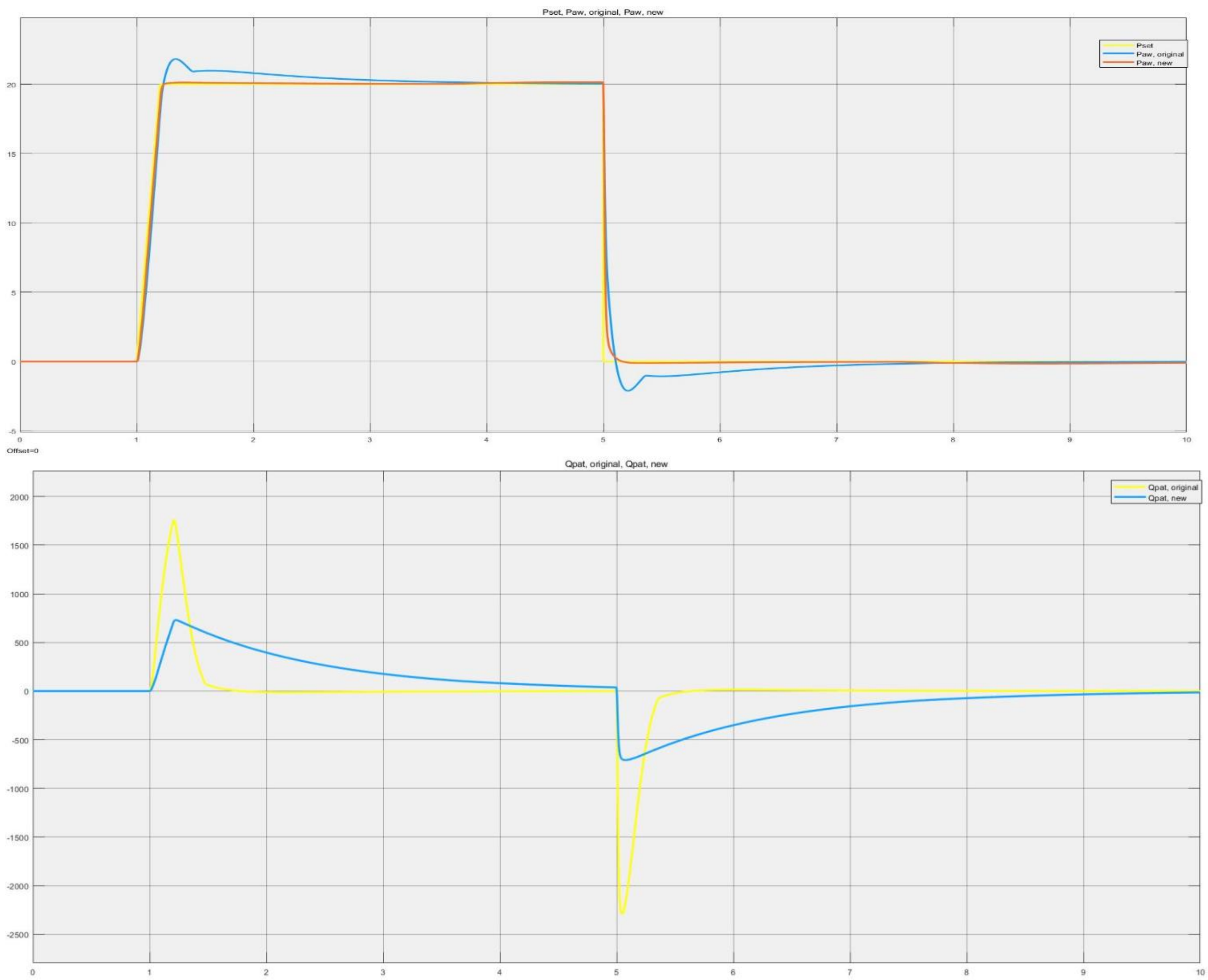
System response for changed system parameters:

	Controller Type		
Transient response	**Non-linear ($\delta = 100$)**	**PID**	**Integral**
Rise Time $T_{r,}$ (ms)	**158.341**	**159.417**	**299.009**
Plateau Pressure (mbar)	**0.15**	**0**	**0.07**
Overshoot (mL/s)	**0**	**0**	**0**

In this example we have slightly increased the lung parameters to model a patient with larger lungs than the standard one. Here we can see that the rise time changes by 30 ms for the integral controlled system. There is no significant change in rise time for the other systems. There is no significant change in plateau pressure for any of the systems. The overshoot decreases to 0 for all the systems. However, there is a large change in overshoot from 35 to 0 for the integral controller and since it also experiences a change in rise time we can conclude it is the least robust system from this example.

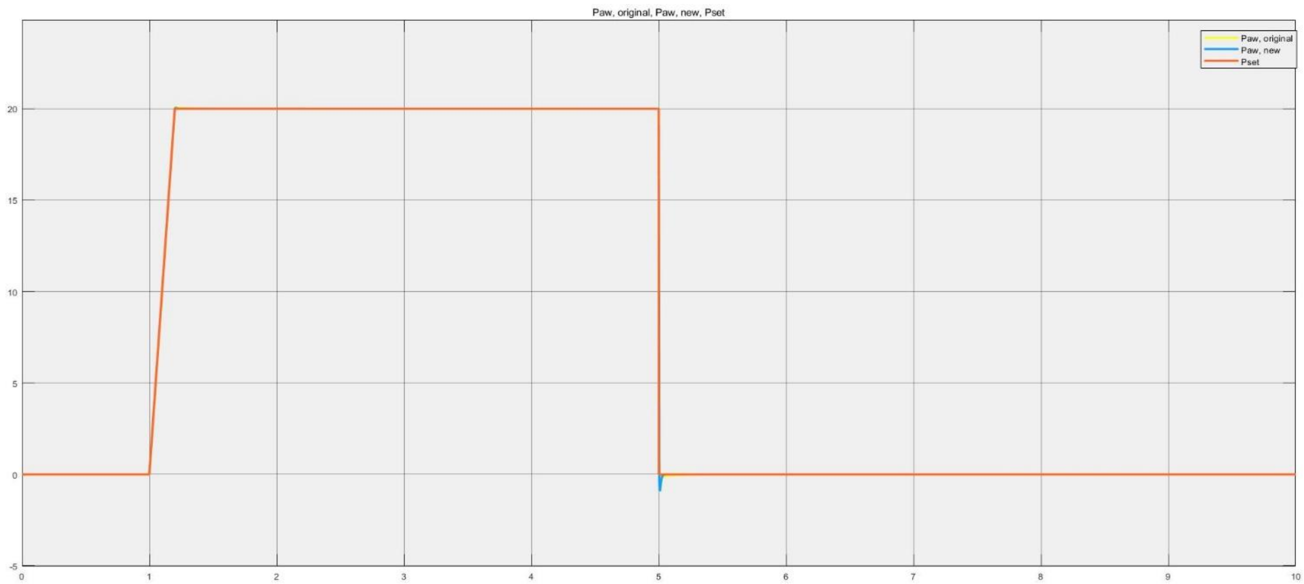
Simulation output for the system:

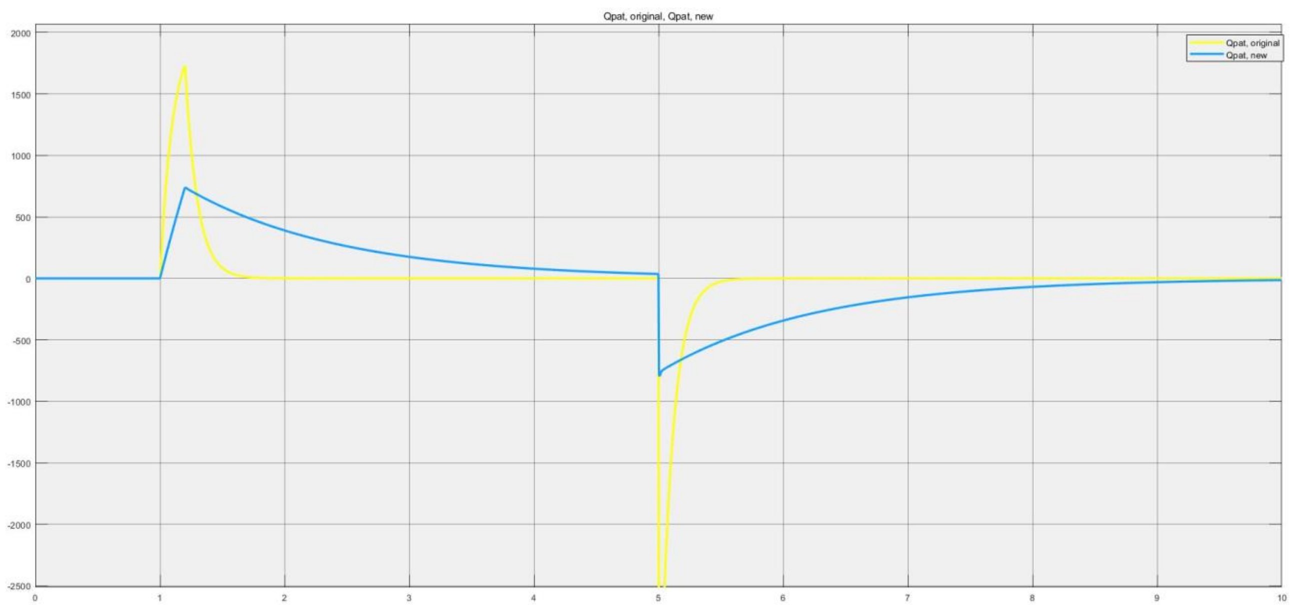
Non-linear controller:



We can see that the rise time is unaffected, but the overshoot is reduced for the non-linear controller.

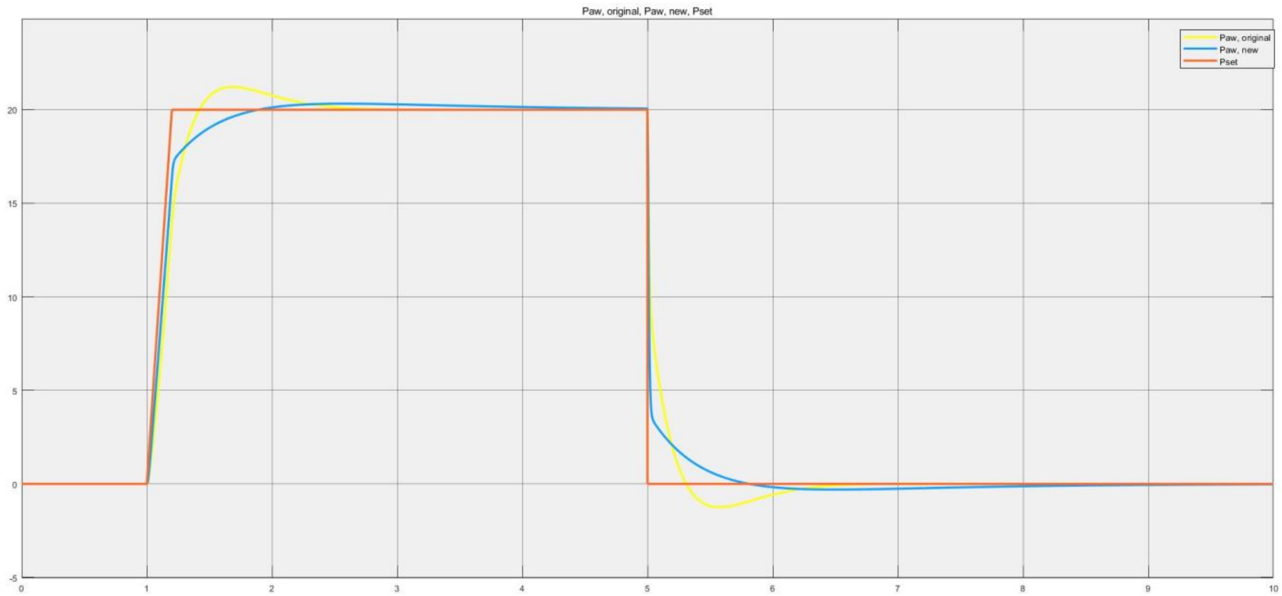
PID controller:

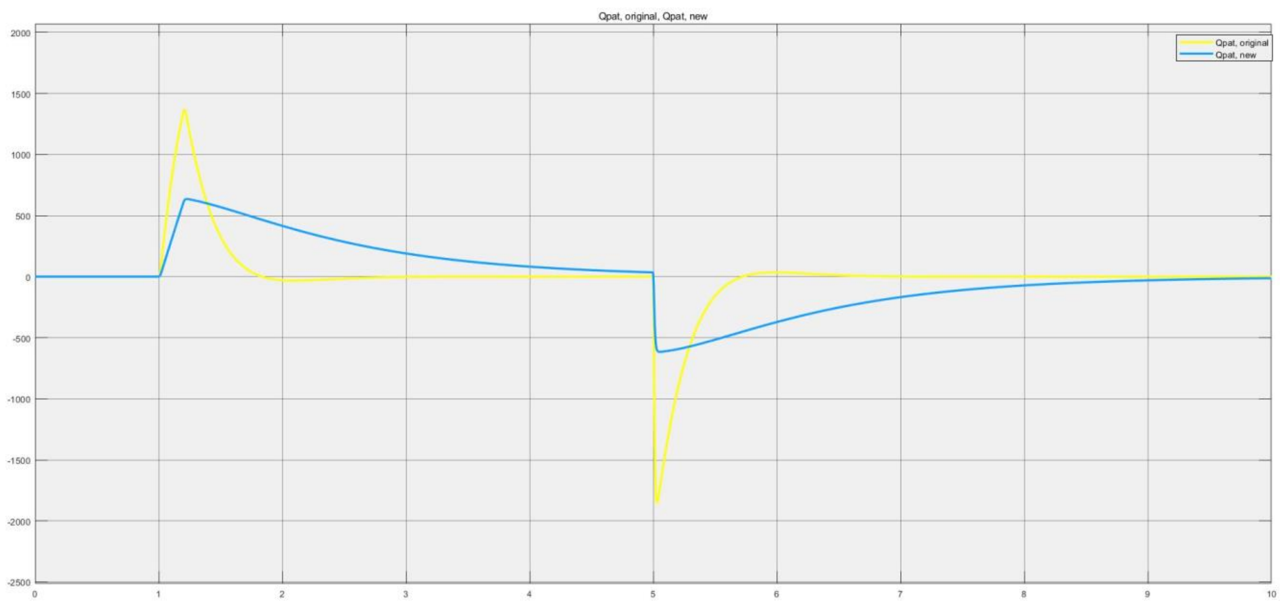




We can see that the rise time is almost unaffected and there is no introduction of overshoot for the PID controller.

Integral controller:





We can see from the plot that the rise time has increased slightly. There has also been a decrease in overshoot for this system.

2. R_{leak} is changed to $\frac{120}{1000}$. The rest of the parameters are unchanged.

System response for changed system parameters:

	Controller Type		
Transient response	**Non-linear ($\delta = 100$)**	**PID**	**Integral**
Rise Time T_r, (ms)	**158.512**	**159.474**	**245.474**
Plateau Pressure (mbar)	**0.03**	**0**	**0**
Overshoot (mL/s)	**14**	**0**	**42**

In this experiment we have increased the leakage resistance to account for a decrease in leakage airflow. Consistent with our previous example we can see that here too the rise time only changes significantly for the integral control. The plateau pressure does not change significantly and remains within the acceptable range. There is no change in overshoot for the non-linear or PID controller, however the overshoot for the integral controller increases to 42 form 35. Hence, we can again conclude that the integral controller is the least robust controller. We have still not gathered enough evidence to compare between the non-linear and PID controller.

3. R_{lung} is changed to $\frac{1}{1000}$ and C_{lung} is changed to 10 . The rest of the parameters are unchanged.

System response for changed system parameters:

	Controller Type		
Transient response	**Non-linear ($\delta = 100$)**	**PID**	**Integral**
Rise Time T_r, (ms)	**137.639**	**158.377**	**177.870**

Plateau Pressure (mbar)	**0.08**	**0**	**0**
Overshoot (mL/s)	**82**	**240**	**22**

In this example we slightly decreased the lung parameters R_{lung} and C_{lung} to model smaller lungs in patients. This time although the rise time for the PID controller is unchanged, the rise time for the other two controller's changes significantly. The overshoot for both the non-linear and PID controllers rise significantly and beyond the acceptable range. However, the overshoot for the non-linear controller can easily be reduced by adjusting the threshold of the switch. This can be done very easily in a practical system by adjusting a knob. We may experience slight increase in rise time, but it should remain under the acceptable range. However, the PID controller parameters cannot be easily changed in a practical system. In this aspect, we can conclude that the non-linear controller is more robust than the PID controller.

We changed the threshold from 100 to 250 for the non-linear controller and achieved the below system response.

Transient response	**Non-linear controller ($\boldsymbol{\delta = 100}$)**
Rise Time T_r, (ms)	**139.731**
Plateau Pressure (mbar)	**0.08**
Overshoot (mL/s)	**45**

Therefore, we can reduce the overshoot by compromising slightly with the rise time.

Simulation output:

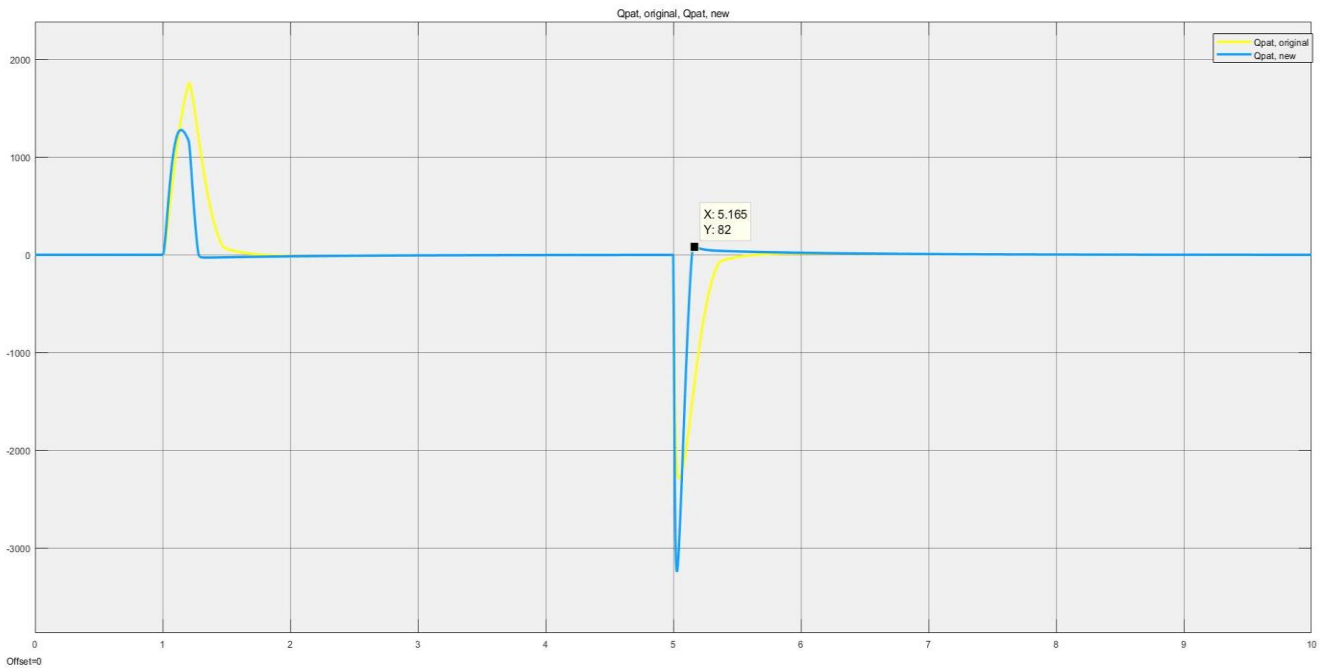


Figure: Q_{pat} and overshoot for threshold = 100

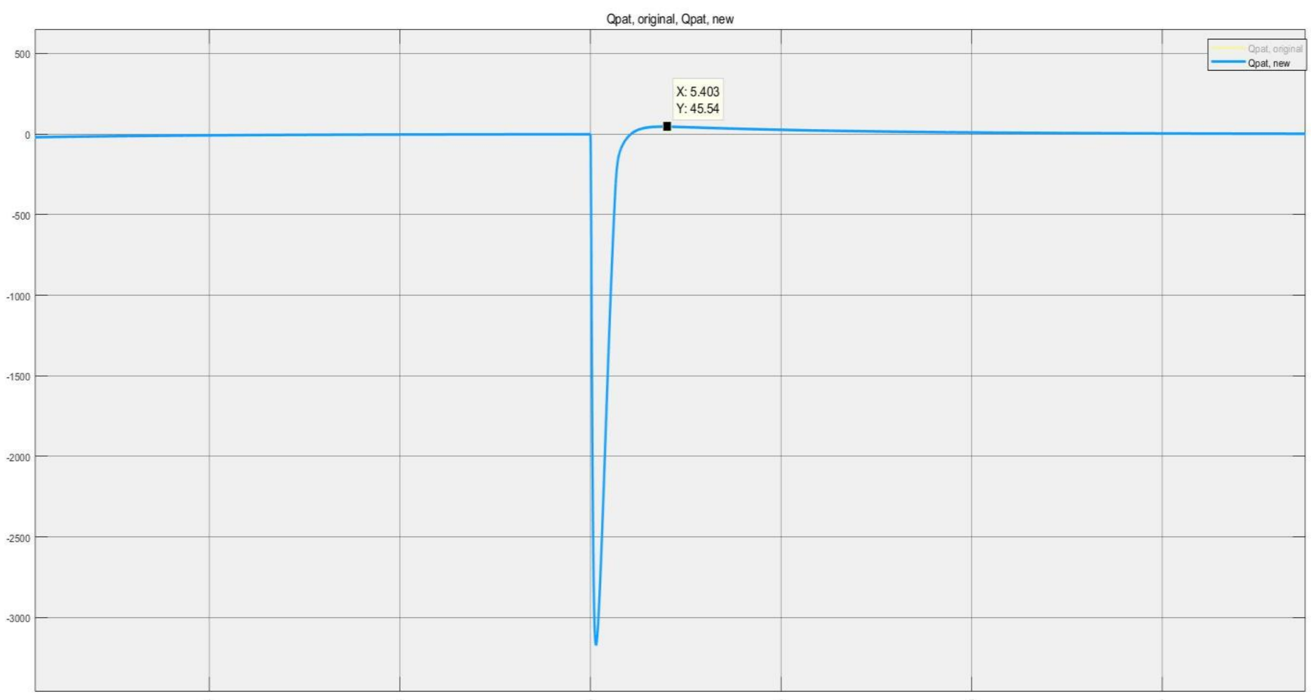


Figure: Q_{pat} and overshoot for threshold = 250

4. R_{lung} is changed to $\frac{1}{5000}$ and C_{lung} is changed to 100 . The rest of the parameters are unchanged.

System response for changed system parameters:

	Controller Type		
Transient response	**Non-linear ($\delta = 100$)**	**PID**	**Integral**
Rise Time $T_{r,}$ (ms)	**189.247**	**150.643**	**394.524**
Plateau Pressure (mbar)	**-0.3**	**0**	**0.2**
Overshoot (mL/s)	**5030**	**7100**	**845**

In this example, the system parameters are changed extensively. This is done to model abnormal conditions, and how the system reacts to these conditions. It can be seen that the rise time still does not change much for the PID controller. We can hence conclude that the PID controller is most effective at maintaining rise time among the three controllers. The rise time for the non-linear controller rises but stays within the acceptable range. The rise time for the integral controller is way above the acceptable limit. We can see that the overshoot has significantly increased for all the controllers. In fact, the values are so large that none of these systems can be successfully applied. However, as we have seen previously the non-linear controller system threshold can be changed to manipulate the overshoot. After careful tuning we found that the lowest overshoot is achieved for threshold = 6000.

Transient response	**Non-linear controller ($\delta = 6000$)**	**Non-linear controller ($\delta = 1000$)**
Rise Time $T_{r,}$ (ms)	**842.165**	**197.744**
Overshoot (mL/s)	**187**	**686**

From the table above we can see that lowest value of overshoot achievable is 190 mL/s, and it comes at a cost of increasing the rise time to 842 ms. Both of these values are much larger than the acceptable range and hence cannot be used. While keeping the rise time within an acceptable range, the lowest possible value of overshoot that can be reached is 680 mL/s. This can be achieved by setting the threshold value to 1000. Therefore, under extremely different conditions, none of the controllers are likely to perform well.

Simulation output:

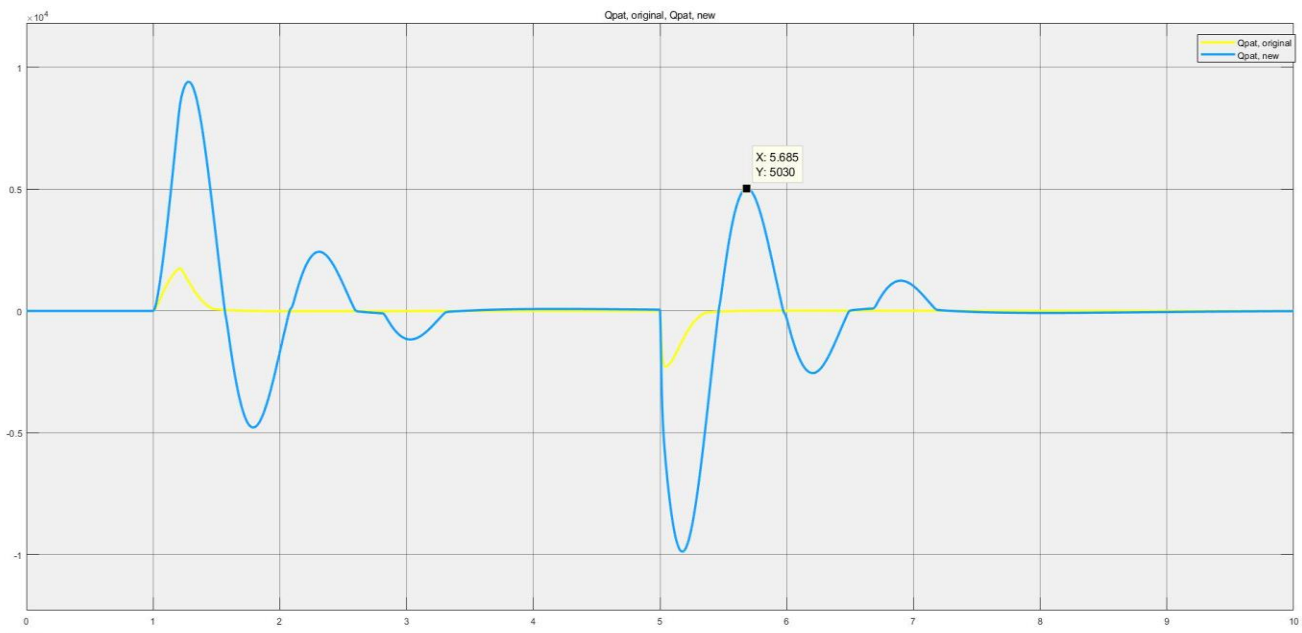


Figure: Q_{pat} for non- linear controller for threshold = 100

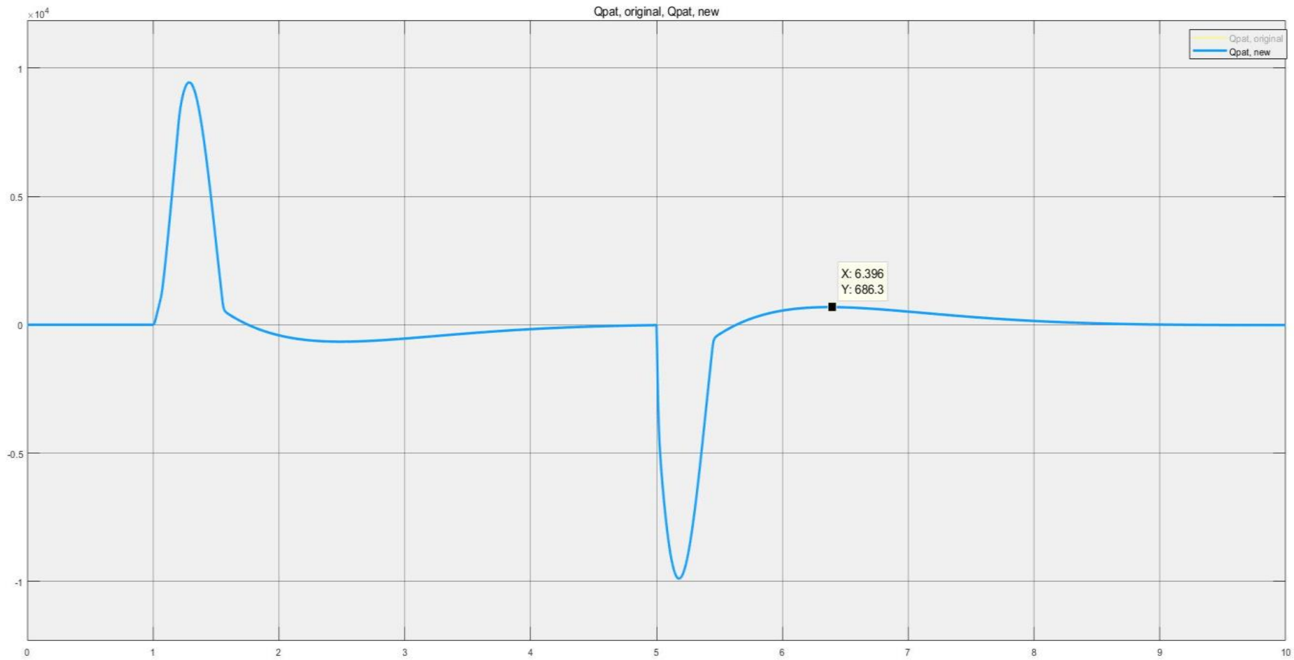


Figure: Q_{pat} for non- linear controller for threshold = 1000

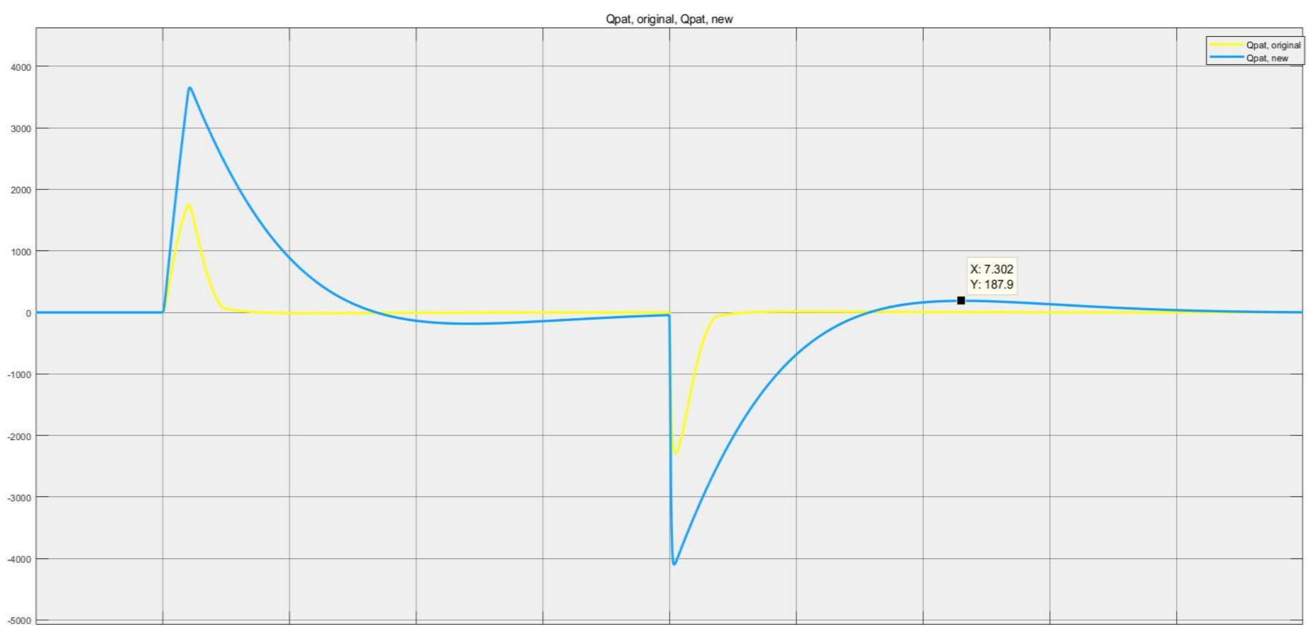


Figure: Q_{pat} for non- linear controller for threshold = 6000

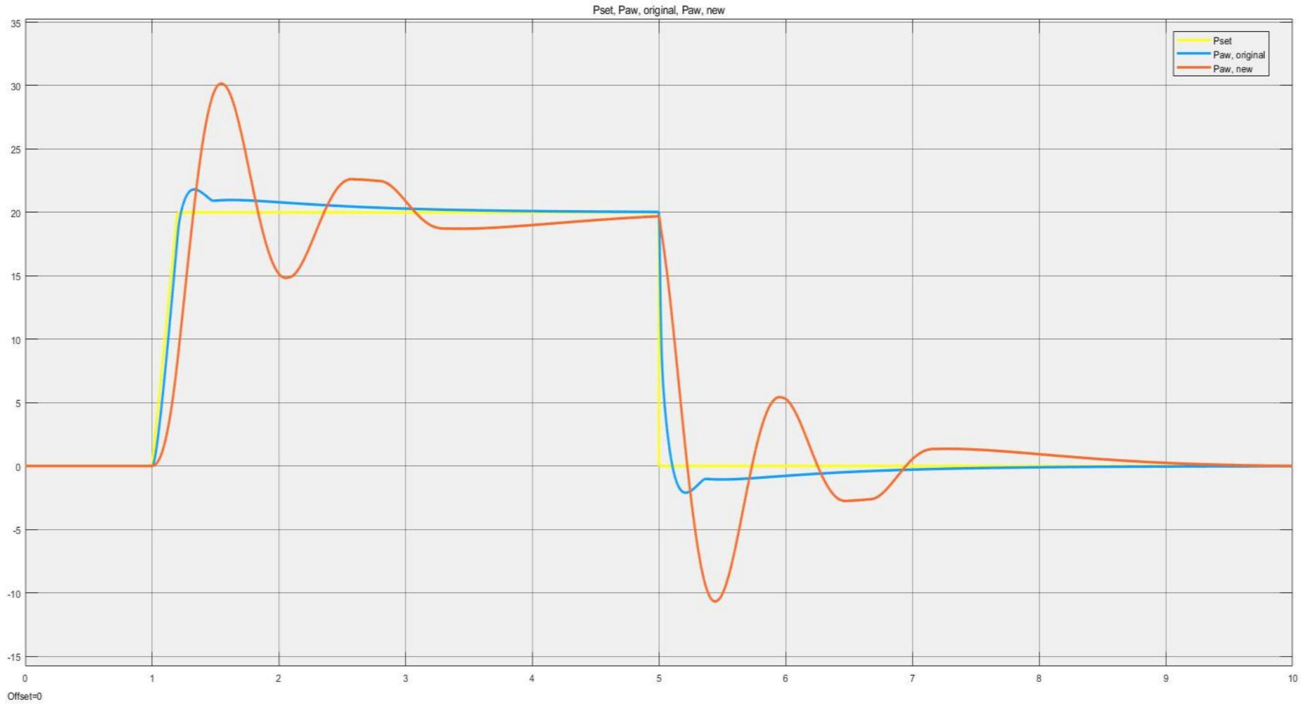


Figure: P_{aw} for non- linear controller for threshold = 100

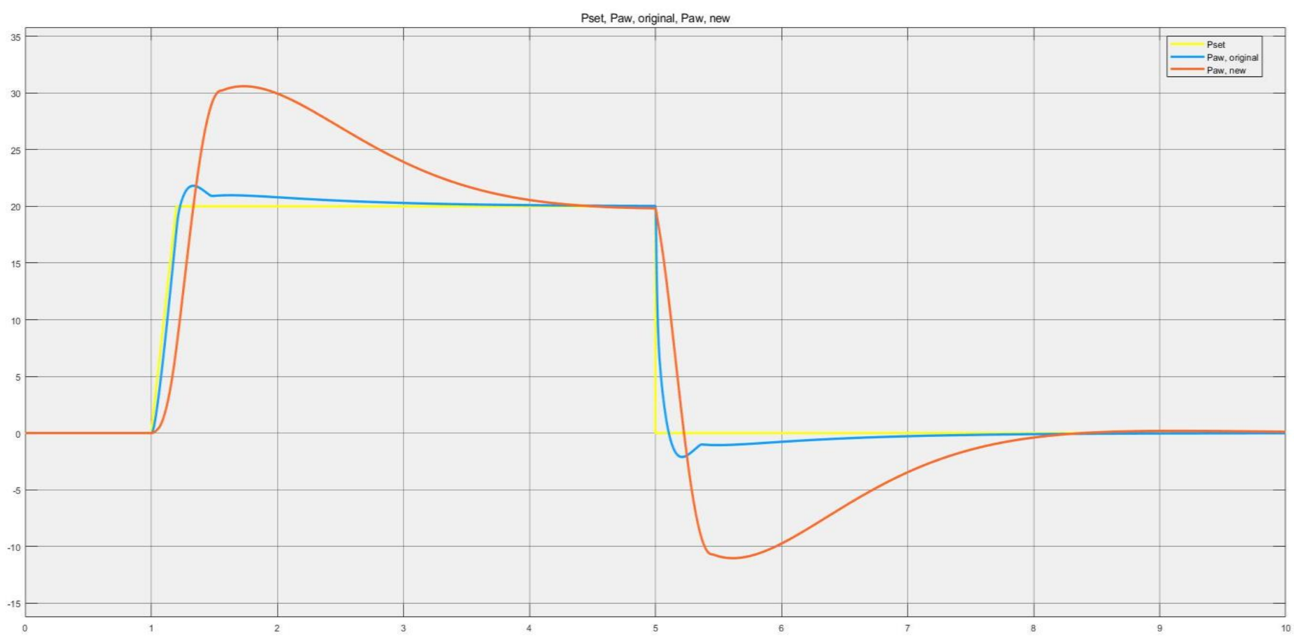


Figure: P_{aw} for non- linear controller for threshold = 1000

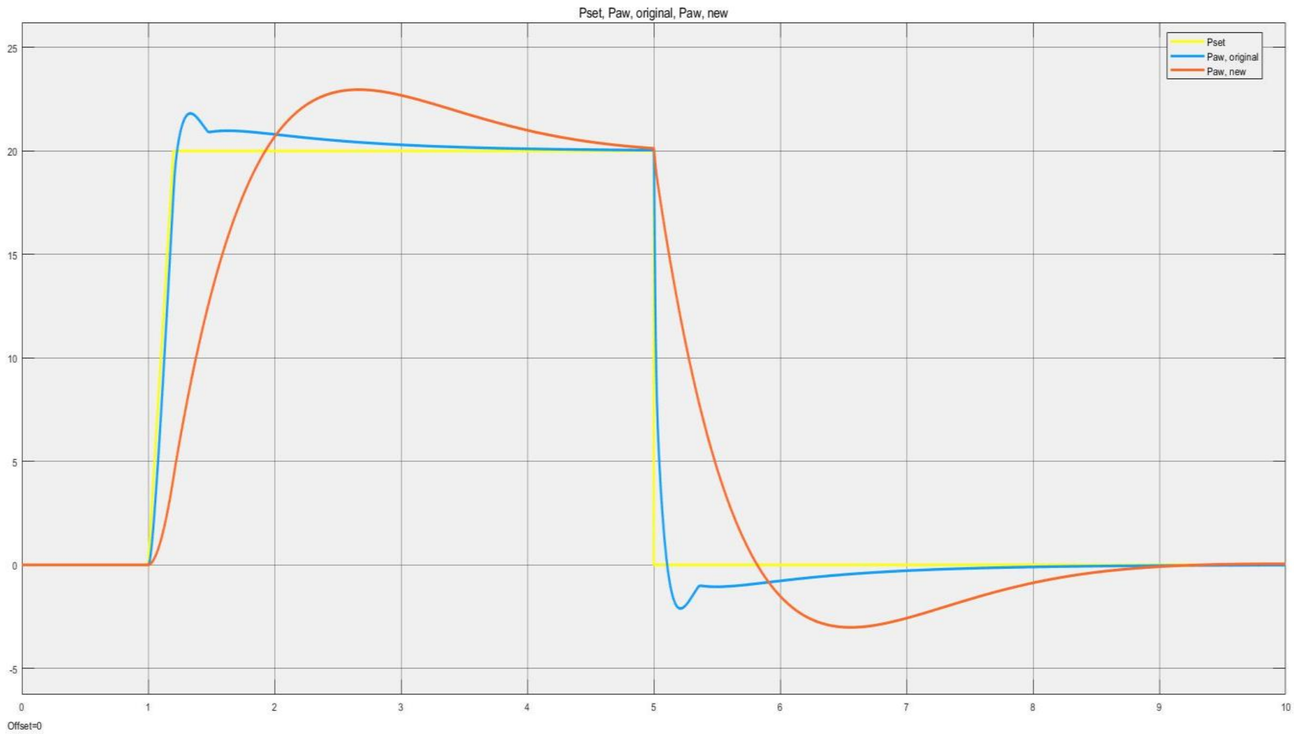


Figure: P_{aw} for non- linear controller for threshold = 6000

From the above graphs we can clearly see how changing the threshold value can reduce the overshoot by increasing the rise time.

Contributions:

1606035 – Explored the possibilities of using a PI or a lag controller in place of the ideal integrator that was originally used for problem 5. Also sketched the root locus in problem 3.

1606039 - Reproduced figure 7 in problem 4 and compared the impact of feedback and feedforward path in the control system. Performed the calculation of the root locus for problem 3.

1606052 – Reproduced the results for both linear and variable gain controllers to draw a comparison between nonlinear and linear control in problem 7.

1606061 – Calculated the transfer functions for problem 1 and 2, designed both the lag and ideal integrator controller in problem 6.

1606063 – Designed the PID controller in problem 6. Analyzed the performances of both linear and nonlinear control systems in presence of varying uncertainties and parameters in problem 8.