Bangladesh University of Engineering and Technology

Department of Electrical and Electronic Engineering

Course No: EEE 318
Course Name: Control Systems Laboratory

Project Title:

Variable-Gain Control for Respiratory Systems[1]

Submitted To:

Rajat Chakraborty, Assistant Professor, Dept. of EEE

Sadman Sakib Ahbab, Lecturer, Dept. of EEE

Submitted By:

Mohseu Rashid Subah, Farhan Sadik, Kaushik Biswas, Sadiah Ahmed Moon, and Ankan Ghosh Dastider

❖ Table of Contents

Objective:

The objective of this project is to explore and reproduce the results of the publication by Hunnekens et. al. 2018: “Variable-gain control for respiratory systems." on *IEEE Transactions on Control Systems Technology* following the answers of some given questions.

Introduction:

Mechanical ventilation is imperative for the patients facing difficulties in breathing by themselves. The scenario has become even more germane during the current COVD-19 time. The basic working method of a mechanical ventilator is quite straightforward – it increases the pressure to fill the lungs with air during an inspiration, and decreased subsequently to release the air from the lungs during expiration. However, maintaining the patient-ventilator synchrony is quite important as the ventilator should be able to accurately track the patient’s condition.

In order to achieve accurate tracking i.e. rapid release and buildup of pressure, high-gain pressure controller is preferred. But to eliminate the oscillations partly, low-gain pressure controller is preferred. So there should be a tradeoff between a high-gain and a low-gain controller, for which the paper dealt with a variable-gain control method, which has been reproduced in this project following the analysis of the answers of the give questions.

Task-1:

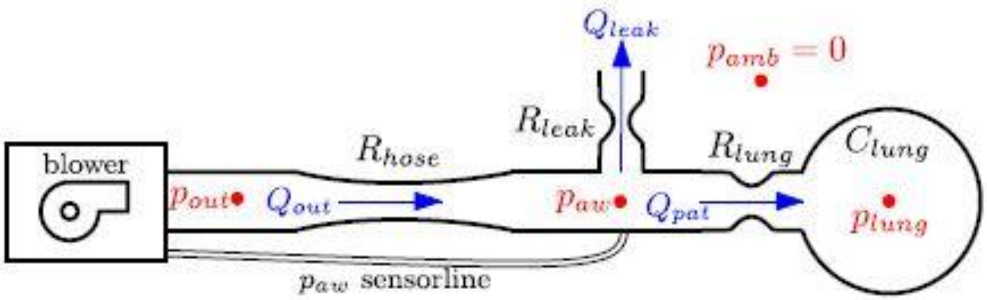


Figure 1: Schematic diagram of a respiratory system

From figure-1,

$$Q_{pat} = Q_{out} - Q_{leak} \tag{1}$$

$$Q_{out} = \frac{p_{out} - p_{aw}}{R_{hose}}$$

$$Q_{leak} = \frac{p_{aw}}{R_{leak}}$$

$$Q_{pat} = \frac{p_{aw} - p_{lung}}{R_{lung}} \tag{2}$$

The lung pressure satisfies the following differential equation:

$$\dot{p}_{\text{lung}} = \frac{Q_{\text{pat}}}{C_{\text{lung}}} = \frac{p_{\text{aw}} - p_{\text{lung}}}{C_{\text{lung}} R_{\text{lung}}} \tag{3}$$

From equations (1) & (2)

$$p_{\text{aw}} = \frac{\frac{p_{lung}}{R_{\text{lung}}} + \frac{p_{out}}{R_{hose}}}{\frac{1}{R_{\text{lung}}} + \frac{1}{R_{hose}} + \frac{1}{R_{leak}}} \tag{4}$$

$$\dot{p}_{\text{lung}} = \frac{-\left(\frac{1}{R_{hose}} + \frac{1}{R_{leak}}\right) p_{lung} + \frac{p_{out}}{R_{hose}}}{R_{\text{lung}} C_{\text{lung}} \left(\frac{1}{R_{\text{lung}}} + \frac{1}{R_{hose}} + \frac{1}{R_{leak}}\right)} \tag{5}$$

From equations (2), (5), and (6), the patient and hose system can be written as a linear state-space system with input p_{out}, outputs $[p_{aw}, Q_{pat}]^T$ and state p_{lung}.

$$\dot{p}_{\text{lung}} = A_h p_{\text{lung}} + B_h p_{\text{out}} \tag{6}$$

$$\begin{bmatrix} paw \\ Q\text{pat} \end{bmatrix} = C_h p_{\text{lung}} + D_h p_{\text{out}} \tag{7}$$

where,

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

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

$$C_h = \left[\frac{\frac{1}{R_{lung}}}{\frac{1}{R_{lung}}+\frac{1}{R_{hose}}+\frac{1}{R_{leak}}} - \frac{\frac{1}{R_{hose}}+\frac{1}{R_{leak}}}{R_{lung}\left(\frac{1}{R_{lung}}+\frac{1}{R_{hose}}+\frac{1}{R_{leak}}\right)}\right]^T$$

$$D_h = \left[\frac{\frac{1}{R_{hose}}}{\frac{1}{R_{lung}}+\frac{1}{R_{hose}}+\frac{1}{R_{leak}}} \quad \frac{\frac{1}{R_{hose}}}{R_{lung}\left(\frac{1}{R_{lung}}+\frac{1}{R_{hose}}+\frac{1}{R_{leak}}\right)}\right]^T \tag{8}$$

which can be written as transfer function, $H(s) = C_h(sI - A_h)^{-1}B_h + D_h$ (9)

The following parameters in Table-1 are used in the simulation.

Variable	**Value**	**Unit**
R_{lung}	5/1000	$\frac{mbar}{\frac{mL}{s}}$
C_{lung}	20	$\frac{mL}{mbar}$
R_{leak}	60/1000	$\frac{mbar}{\frac{mL}{s}}$
R_{hose}	4.5/1000	$\frac{mbar}{\frac{mL}{s}}$
ω_n	$2\pi 30$	$\frac{rad}{s}$

Table 1: Parameters needed for the simulation

Code:

```
%% -----Task 1: Determining H(s)----- %%
R_lung = 5/1000;
C_lung = 20;
R_leak = 60/1000;
R_hose = 4.5/1000;

A_h = - (1/R_hose + 1/R_leak)/(R_lung * C_lung * (1/R_lung + 1/R_hose + 1/R_leak));
B_h =  (1/R_hose)/(R_lung * C_lung * (1/R_lung + 1/R_hose + 1/R_leak));
C_h = [(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))];
D_h = [(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');
Hs = C_h * inv(s-A_h) * B_h + D_h;
```

Figure 2: MATLAB code for Task-1

Output:

The code in Figure 2 is used to solve (8) and the values are as follows.

$$A_h = -5.4430$$

$$B_h = 5.0633$$

$$C_h = \begin{bmatrix} 0.4557 \\ -108.8608 \end{bmatrix}$$

$$D_h = \begin{bmatrix} 0.5063 \\ 101.2658 \end{bmatrix}$$

Hence, the transfer function, *H(s)* can be written as,

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

Task-2:

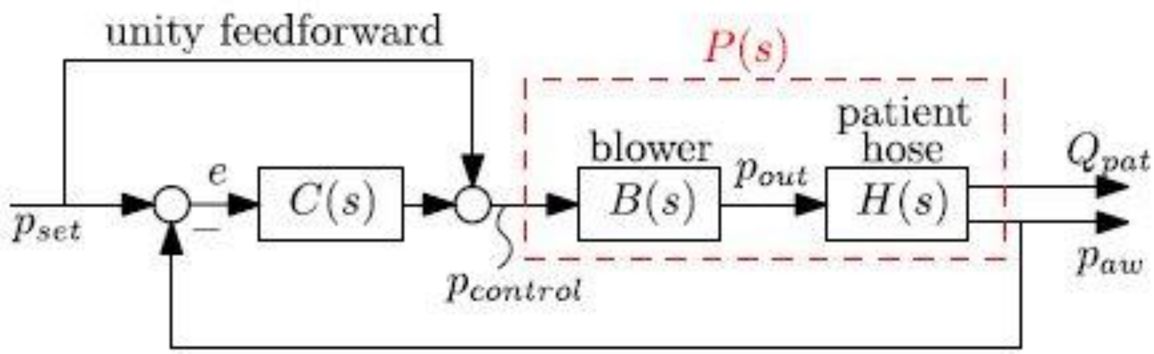


Figure 3: Closed loop control system with a linear controller

However, the blower is a dynamical system with inertia; therefore, the actual system has roll-off for high frequencies, which can be modeled as a second-order low-pass filter,

$$B(s) = \frac{p_{out}(s)}{p_{control}(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{10}$$

Here, $\omega_n = 2\pi 30$ and damping ratio, $\zeta = 1$.

In the state space format, it can be written as,

$$\dot{x}_b = A_b x_b + B_b p_{control}$$

$$p_{out} = C_b x_b$$

with state $x_b \in R2$, output p_{out}, and control input $p_{control}$, and system matrices,

$$A_b = \begin{pmatrix} -2\zeta\omega_n & -\omega_n^2 \\ 1 & 0 \end{pmatrix}$$

$$B_b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$C_b = \begin{pmatrix} 0 & \omega_n^2 \end{pmatrix}$$

By coupling the patient hose system dynamics (6) and the blower dynamics (9), the general state-space form of the plant *P(s)* (to be controlled by the feedback controller; see Fig. 3) can be formulated as

$$\dot{x}_p = \begin{bmatrix} \dot{x}_b \\ \dot{p}_{lung} \end{bmatrix} = \begin{bmatrix} A_b & 0 \\ B_h C_b & A_h \end{bmatrix} \begin{bmatrix} x_b \\ p_{lung} \end{bmatrix} + \begin{bmatrix} B_b \\ 0 \end{bmatrix} p_{control}$$

and $z = \begin{bmatrix} p_{aw} \\ Q_{pat} \end{bmatrix} = \begin{bmatrix} D_b C_b & C_h \end{bmatrix} \begin{bmatrix} x_b \\ p_{lung} \end{bmatrix}$

So, the transfer function will be,

$$P(s) := \begin{bmatrix} P_1(s) \\ P_2(s) \end{bmatrix} = B(s)H(s) = C_P(sI - A_P)^{-1}B_P. \qquad (11)$$

where, $A_P = \begin{bmatrix} A_b & 0 \\ B_h C_b & A_h \end{bmatrix}, B_P = \begin{bmatrix} B_b \\ 0 \end{bmatrix}, C_p = \begin{bmatrix} D_b C_b & C_h \end{bmatrix}$

From the closed loop control system in figure-3, we can write

$$e = P_{set} - P_{aw}$$

$$P_{control} = P_{set} + eC(s)$$

$$P_{aw} = P_{control} \times P_1(s) \qquad (12)$$

$$Q_{pat} = P_{control} \times P_2(s) \qquad (13)$$

From equation (12),

$$P_{aw} = (P_{set} + eC(s)) \times P_1(s)$$

$$= (P_{set} + (P_{set} - P_{aw}) \times C(s)) \times P_1(s)$$

$$= P_{set}P_1(s) + P_{set}P_1(s)C(s) - P_{aw}P_1(s)C(s)$$

$$P_{aw}\big(1 + P_1(s)C(s)\big) = P_{set}(P_1(s) + P_1(s)C(s))$$

$$\frac{P_{aw}}{P_{set}} = \frac{P_1(s)(1 + C(s))}{\big(1 + P_1(s)C(s)\big)}$$

$$T_1(s) = \frac{P_1(s)(1+C(s))}{\big(1+ P_1(s)C(s)\big)} \qquad (14)$$

Again, from equation (13),

$$Q_{pat} = (P_{set} + eC(s)) \times P_2(s)$$

$$= (P_{set} + (P_{set} - P_{aw}) \times C(s)) \times P_2(s)$$

$$= P_{set}P_2(s) + P_{set}P_2(s)C(s) - P_{aw}P_2(s)C(s)$$

$$= P_{set}P_2(s) + P_{set}P_2(s)C(s) - T_1(s)P_{set}P_2(s)C(s)$$

$$= P_{set}(P_2(s) + P_2(s)C(s) - T_1(s)P_2(s)C(s))$$

$$\frac{Q_{pat}}{P_{set}} = P_2(s)(1 + C(s) - T_1(s)C(s))$$

$$T_2(s) = P_2(s)(1 + C(s)(1 - T_1(s)))$$

$$= P_2(s)(1 + C(s)(1 - \frac{P_1(s)(1+C(s))}{(1+P_1(s)C(s))}))$$

$$= \frac{P_2(s)(1+C(s))}{(1+P_1(s)C(s))} \tag{15}$$

Equations (14) & (15) are basically the transfer functions for the closed loop system shown in Figure-3.

The linear integral feedback controller, $C(s) = \frac{k_i}{s}$; where k_i is assumed 0.4

Code:

```
%% -----Task 2: Determining T(s)----- %%

%Determining B(s)
w_n = 2*pi*30;
zeta = 1;

Bs = w_n^2 / (s^2 + 2*zeta*w_n*s + w_n^2);

%Determining P(s)
Ps = Hs*Bs;

%Determining T(s) for a particular ki
ki = 0.4;
Cs = ki/s;

Ts1 = (Ps(1)*(Cs+1))/(1+Cs*Ps(1));
Ts2 = (Ps(2)*(Cs+1))/(1+Cs*Ps(1));
```

Figure 4: MATLAB code for task-2

Output:

The code in figure-4 is used to solve (10), (11), (14) & (15). The values are as follows:

$$B(s) = \frac{3.553e^4}{s^2 + 377s + 3.553e^4}$$

$$P(s) = \begin{bmatrix} \dfrac{1.799e^4s + 1.799e^5}{s^3 + 382.4\,s^2 + 3.758e^4s + 1.934e^5} \\ \dfrac{3.598e^6s + 4.039e^{-9}}{s^3 + 382.4\,s^2 + 3.758e^4s + 1.934e^5} \end{bmatrix}$$

$$T_1(s) = \frac{1.799e^4s^6 + 7.067e^6s^5 + 7.477e^8s^4 + 1.054e^{10}s^3 + 3.889e^{10}s^2 + 1.392e^{10}s}{s^8 + 764.9s^7 + 2.214e^5s^6 + 2.914e^7s^5 + 1.563e^9s^4 + 1.483e^{10}s^3 + 4.15e^{10}s^2 + 1.392e^{10}s}$$

$$T_2(s) = \frac{1.799e^4s^6 + 7.067e^6s^5 + 7.477e^8s^4 + 1.054e^{10}s^3 + 3.889e^{10}s^2 + 1.392e^{10}s}{s^8 + 764.9s^7 + 2.214e^5s^6 + 2.914e^7s^5 + 1.563e^9s^4 + 1.483e^{10}s^3 + 4.15e^{10}s^2 + 1.392e^{10}s}$$

So, the transfer function of the closed loop system in figure-3, can be written as

$$T(s) = \begin{bmatrix} T_1(s) \\ T_2(s) \end{bmatrix} = \begin{bmatrix} \dfrac{1.799e^4s^6 + 7.067e^6s^5 + 7.477e^8s^4 + 1.054e^{10}s^3 + 3.889e^{10}s^2 + 1.392e^{10}s}{s^8 + 764.9s^7 + 2.214e^5s^6 + 2.914e^7s^5 + 1.563e^9s^4 + 1.483e^{10}s^3 + 4.15e^{10}s^2 + 1.392e^{10}s} \\ \dfrac{1.799e^4s^6 + 7.067e^6s^5 + 7.477e^8s^4 + 1.054e^{10}s^3 + 3.889e^{10}s^2 + 1.392e^{10}s}{s^8 + 764.9s^7 + 2.214e^5s^6 + 2.914e^7s^5 + 1.563e^9s^4 + 1.483e^{10}s^3 + 4.15e^{10}s^2 + 1.392e^{10}s} \end{bmatrix}$$

Task - 3

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

There are two different transfer functions for the two outputs Q_pat and P_aw.

$$T_1(s) = \frac{P_1(s)\big(1 + C(s)\big)}{\big(1 + P_1(s)C(s)\big)}$$

$$T_2(s) = \frac{P_2(s)\big(1 + C(s)\big)}{\big(1 + P_1(s)C(s)\big)}$$

The linear integral feedback controller, $(s) = \frac{Ki}{s}$. For Ki >0 , we plot the root locus for $\frac{P1(s)}{s}$.

Code:

```
%% -----Task 3: Sketching the root locus----- %%
rlocus(Ps(1)/s)
xlim([-200 20]);
ylim([-300 300]);

```

Root Locus:

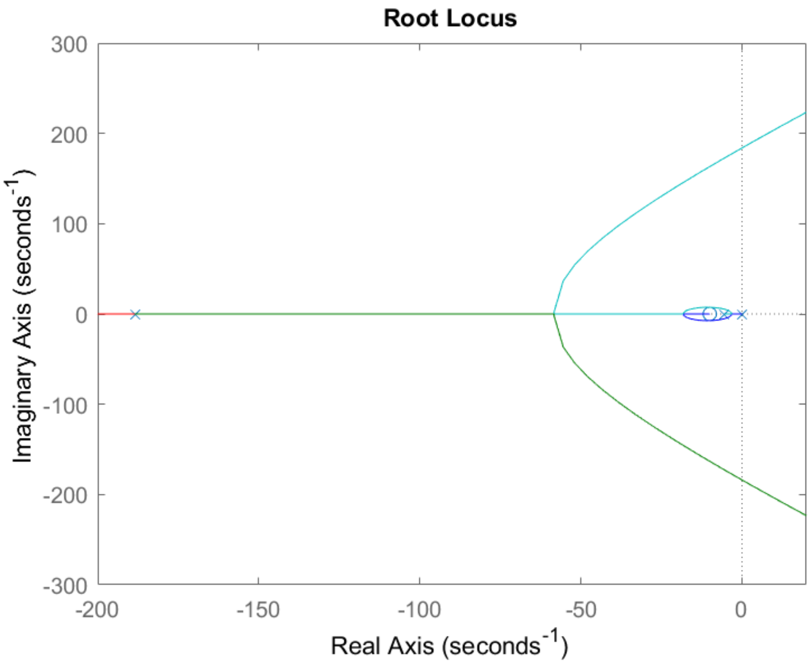


Figure 5 Root locus

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.

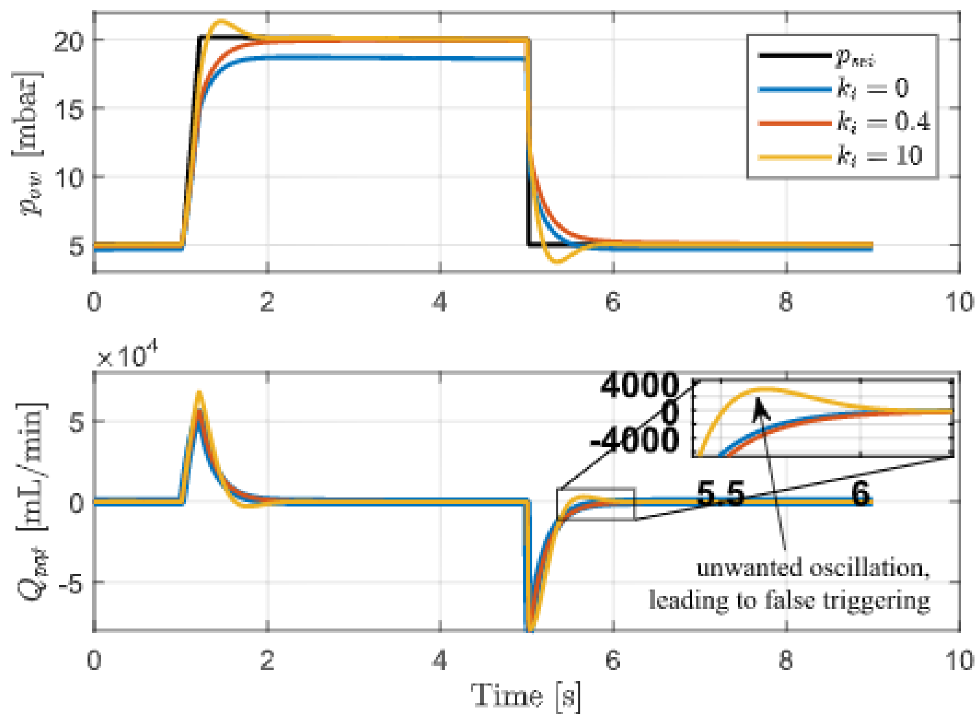
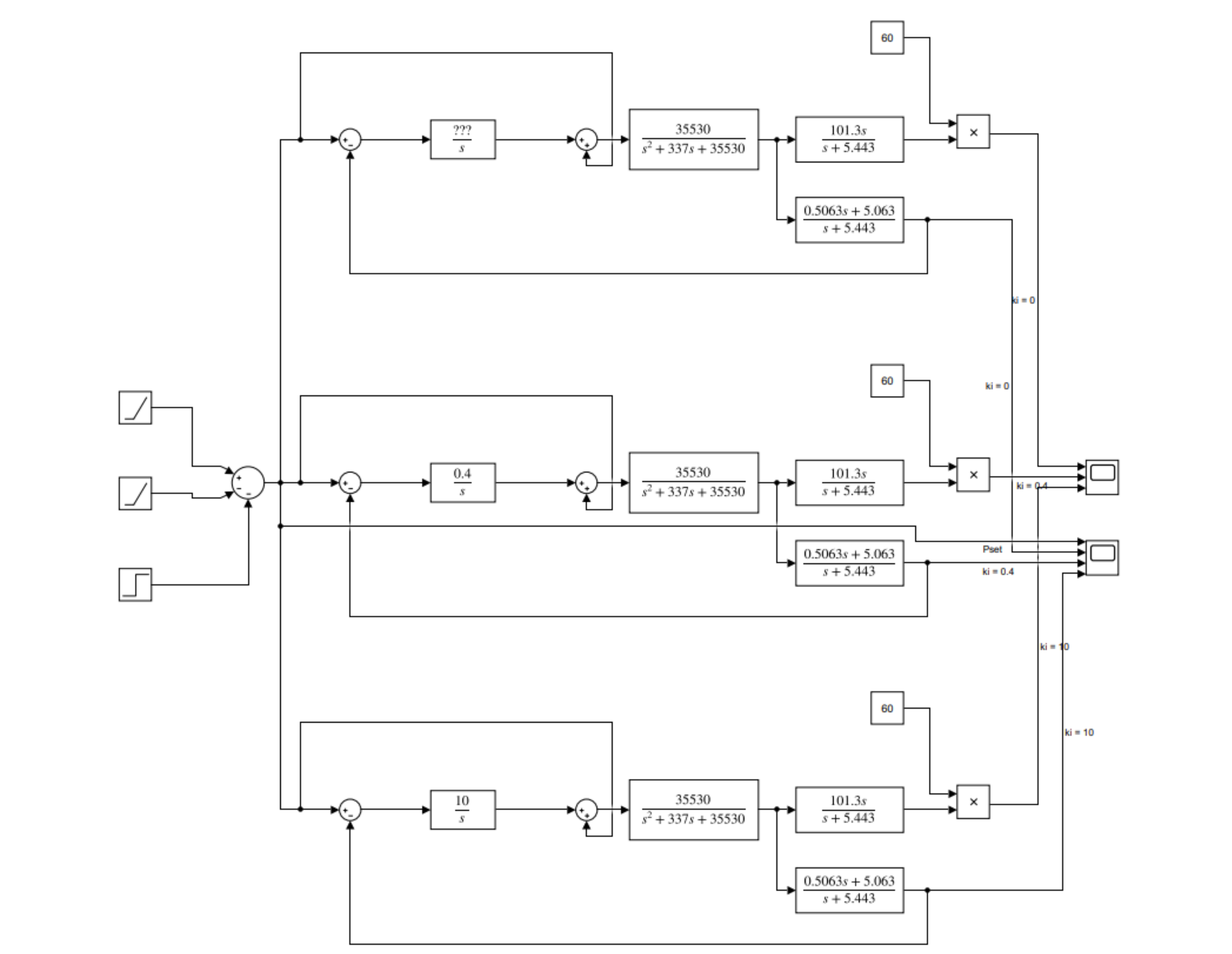
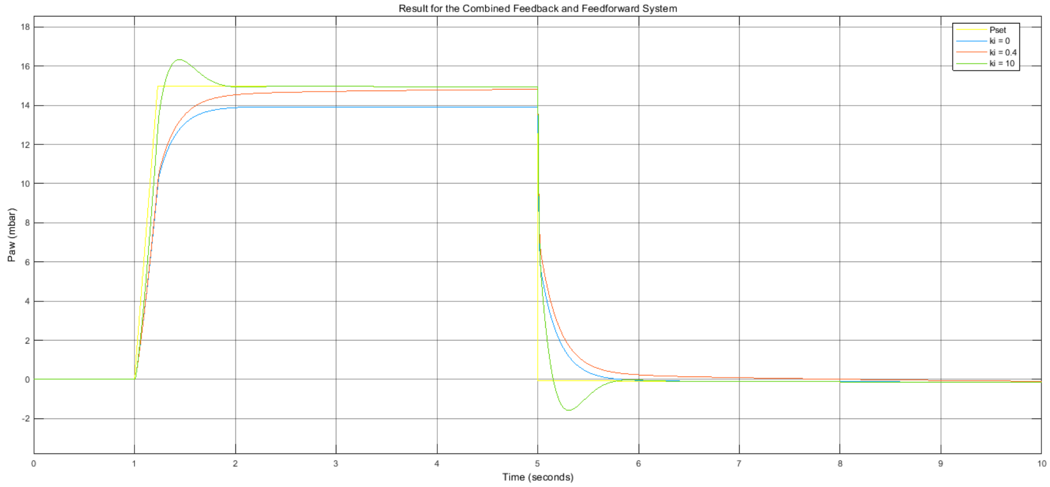


Fig. 7. Simulation result of the closed-loop system using no controller, a low-gain controller ($k_i = 0.4$), and a high-gain controller ($k_i = 10$).

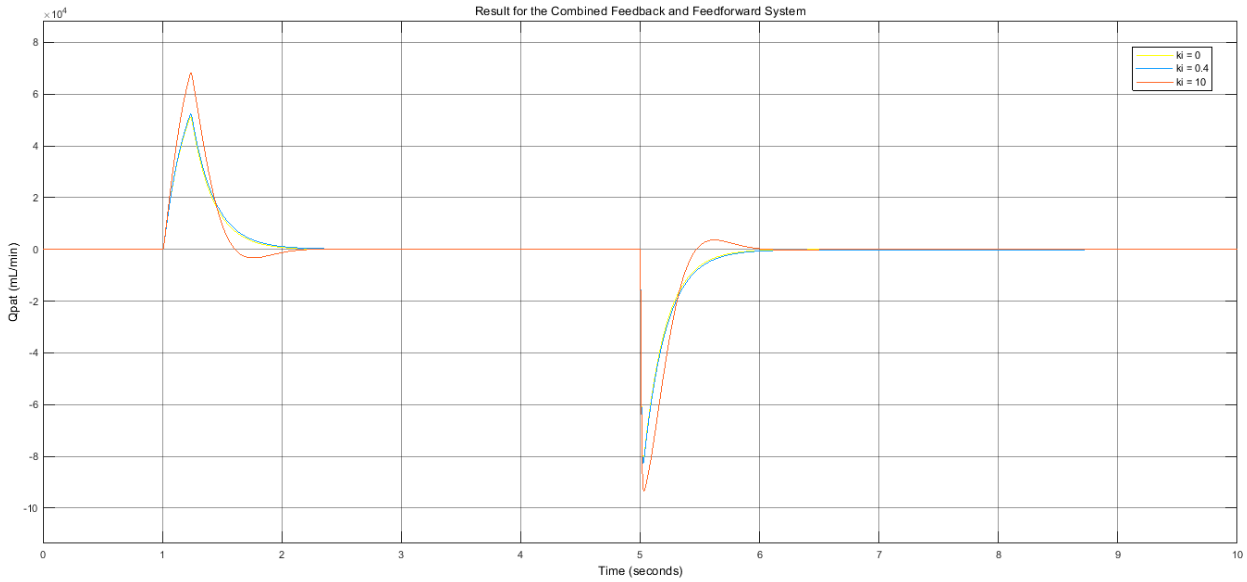
Simulink Diagram:



Output for P_aw:



Output for Q_pat:



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

For ki = 0 we have Open-loop control or C(s) = 0. According to the simulation, it cannot reach the target pressure for 0 value of ki.

Next we change the parameter to ki=0.4 and achieve a low-gain integral feedback controller c(s). In this case, the output eventually reaches the target pressure. However the rise time, T_r is significantly low. Therefore, the response is slower. However, due to low gain, it has zero overshoot.

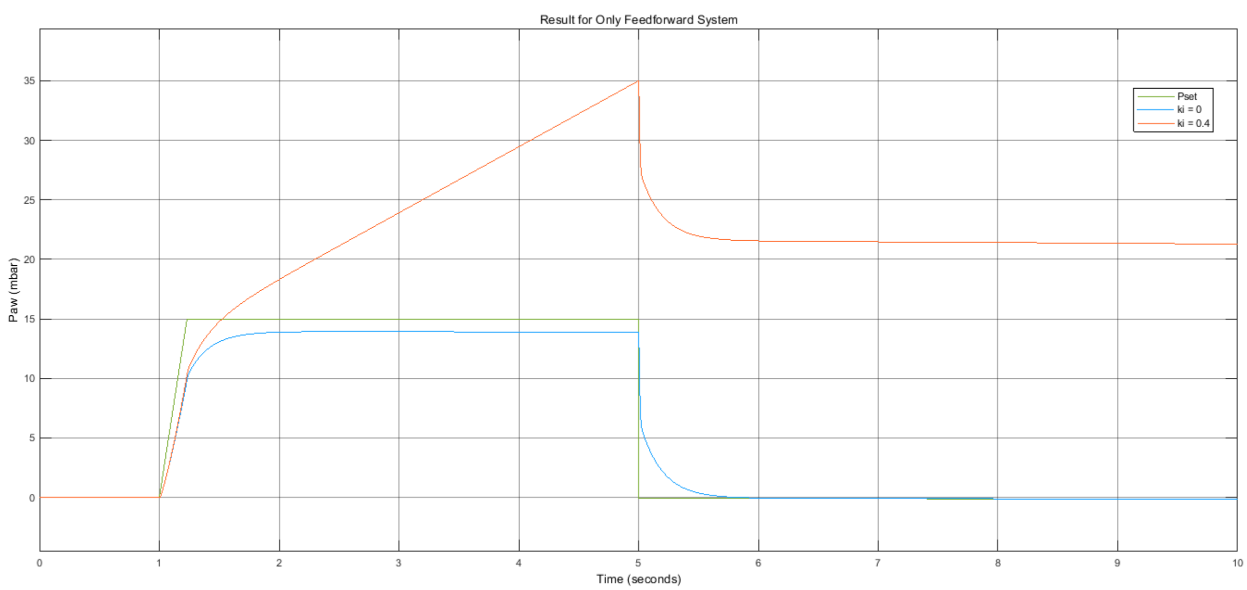
Up next, For ki = 10 we achieve a high-gain integral feedback controller C(s). Due to the high-gain, the system reaches the target pressure faster but creates an unwanted oscillation in the patient flow. This might result in false patient flow triggering.

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.

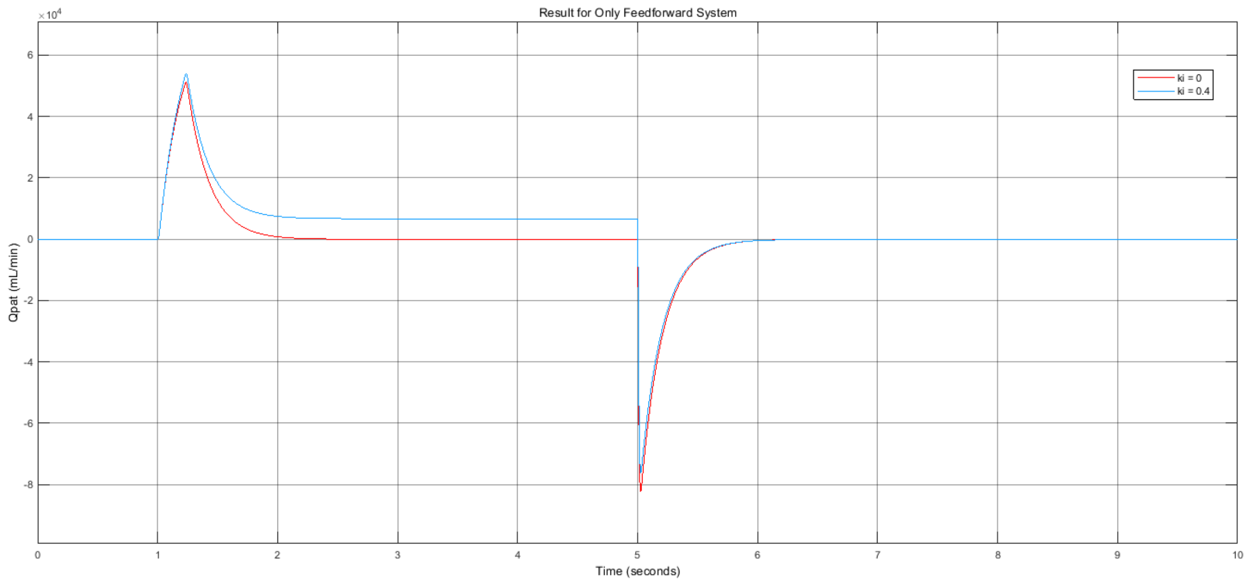
Without Feedback:

We eliminate the feedback from the system and observe the output that is given below.

Output for P_aw:



Output for Q_pat:

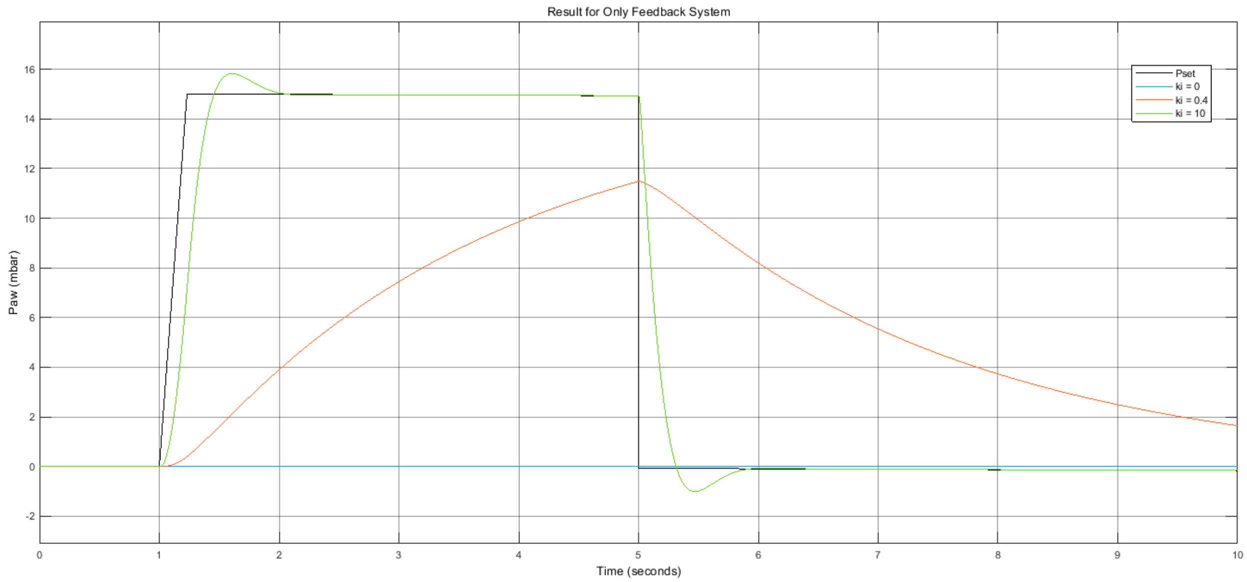


According to the simulation result, it is very clear that there is a large value of error in the system. The goal of the system is to achieve sufficiently fast pressure buildup and accurate tracking of the desired pressure profile pset. However, without feedback a steady state error always persists. 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 – p_aw 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.

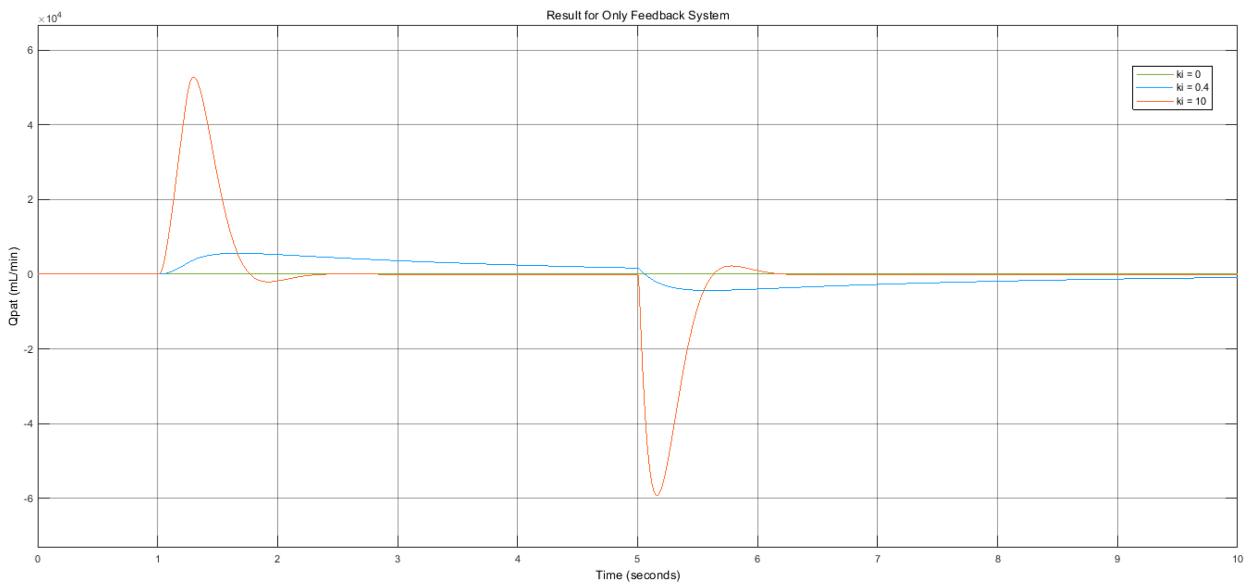
Without Feedforward:

We eliminate the feedforward circuit from the system and observe the output that is given below.

Output for P_aw:



Output for Q_pat:



These simulation results indicate that without the unity feed forward, output pressure pout cannot accurately track the target pressure pset. According to the instructions, T_r from 10% to 90% of a pressure set point should be approximately 200 ms. For ki = 0 and 0.4 the system is far from meeting the specifications. For ki = 10 the system can somewhat meet the specifications, but the overshoot is higher. Therefore, the unity feedforward system is also necessary.

From analyzing the simulation results and the required specifications, we come to the conclusion that both unity feedback and unity forward circuit is absolutely necessary for the system.

Task - 5

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.

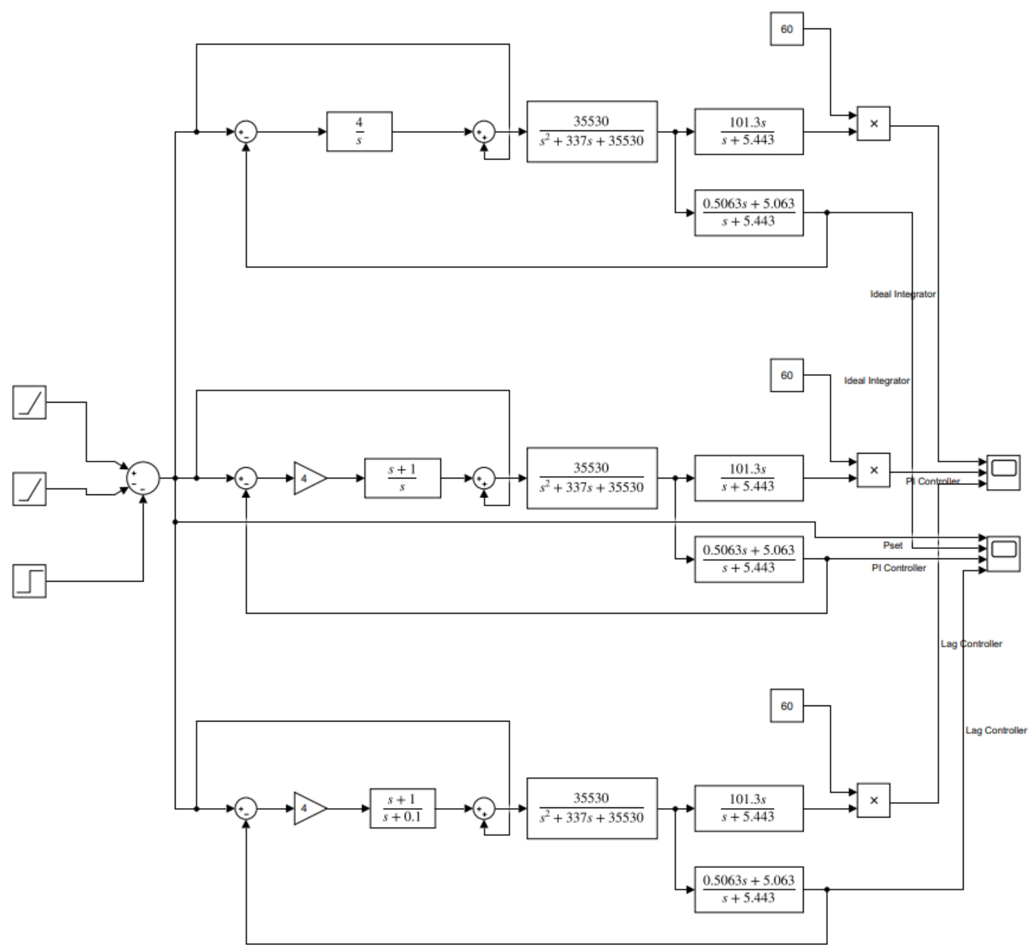
Here. We have used:

Ideal integrator controller, $C_1(s) = \frac{4}{s}$

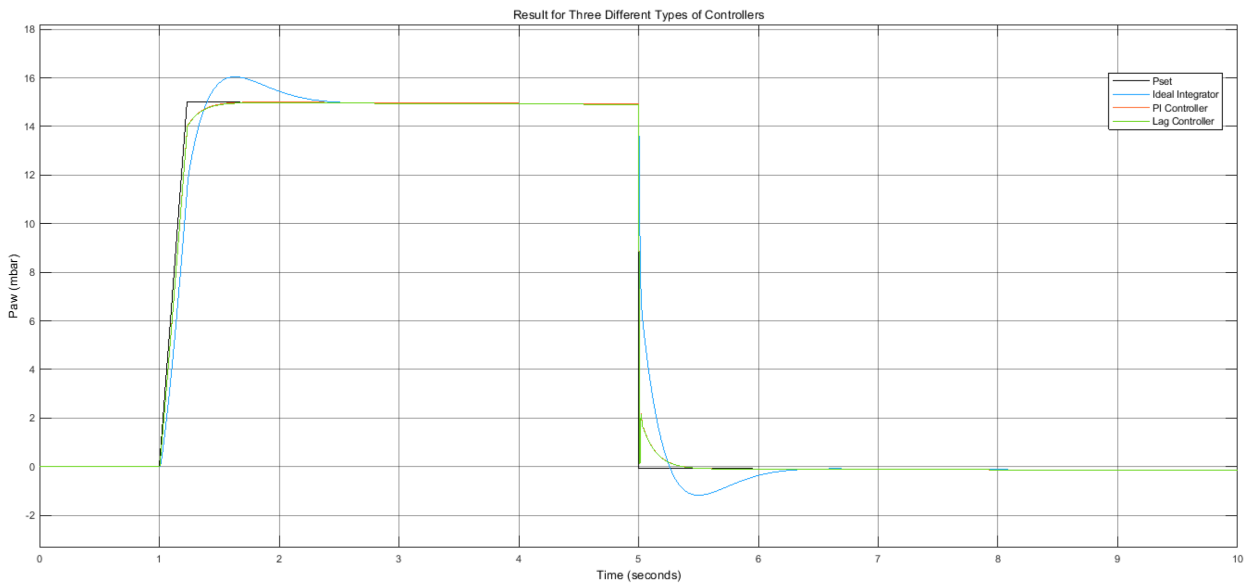
PI controller, $C_2(s) = \frac{4(s+1)}{s}$

Lag controller, $C_3(s) = \frac{4(s+1)}{s+0.1}$

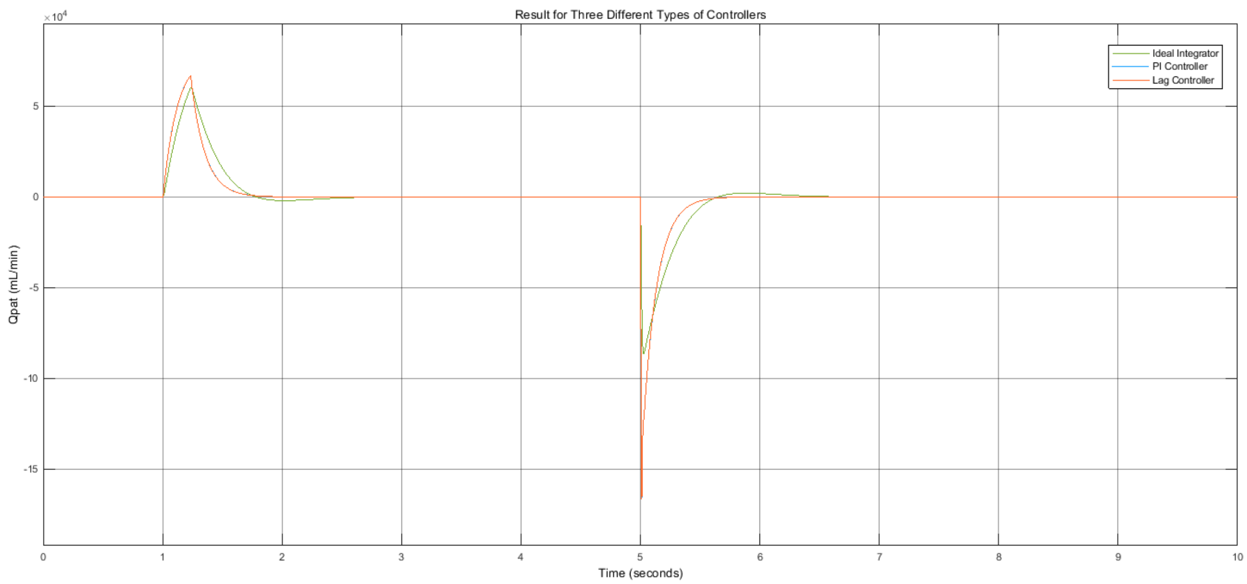
Simulink Diagram:



Output for P_aw:



Output for Q_pat:



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. Whereas, ideal integrator has a rise time outside the specified range, PI and Lag controller has T_r within 200ms. The overshoot is also lower for PI and Lag controllers. We have a similar value of overshoot however, PI controller gives 0 error whereas there is a minimum error in Lag controller.

Therefore, PI controller is preferable.

The specifications are given below.

	Overshoot (%OS)	**Rise Time (T_r)**	**Error**
Ideal Integrator	2.088 L/min	254 ms	0 mbar
PI	0	196 ms	0 mbar
Lag	0	196 ms	0.02 mbar

Task - 6

Designing the preferred linear controller 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.

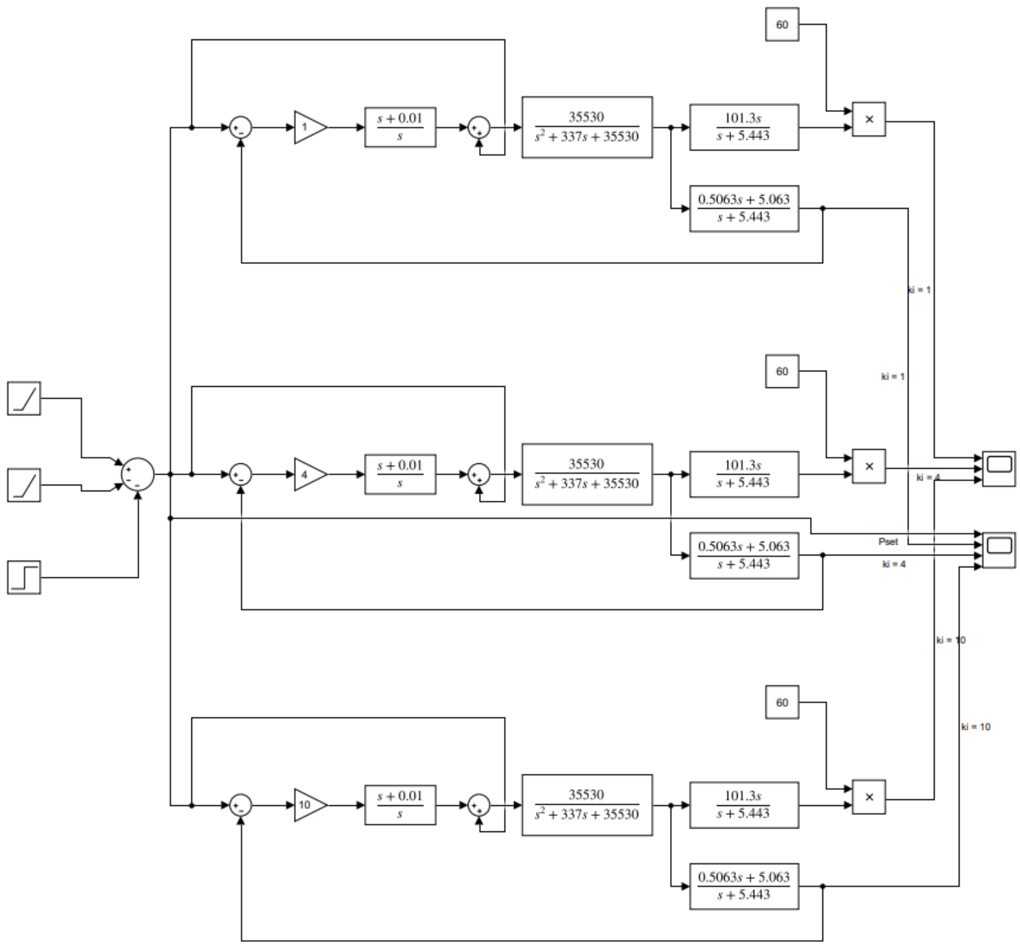
2. The pressure at the end of an inspiration, the so-called plateau pressure, should be within a pressure band of ±2 mbar of the pressure set point.

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.

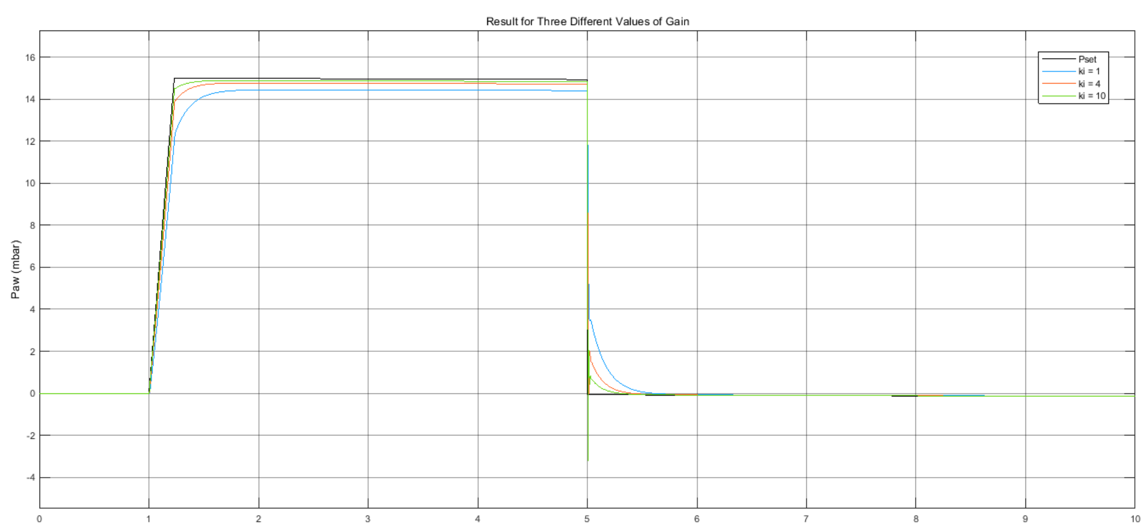
As discussed in the previous section, PI controller is the most preferable one for our system. On this task we used different values of K_i to determine the optimum value that meets the required specification. The value of $Z_c = 0.01$ for the PI controller. Among the many variations we have experimented with, $K_i = 1$, 4 and 10 gave satisfactory results.

	Rise Time (T_r)	**Overshoot (%OS)**	**Error**
$(s + 0.01)/s$	311 ms	0	0.52 mbar
$4(s + 0.01)/s$	198 ms	0	0.21 mbar
$10(s + 0.01)/s$	190 ms	0	0.09 mbar

Simulink Diagram:



Output for P_aw:



Output for Q_pat:

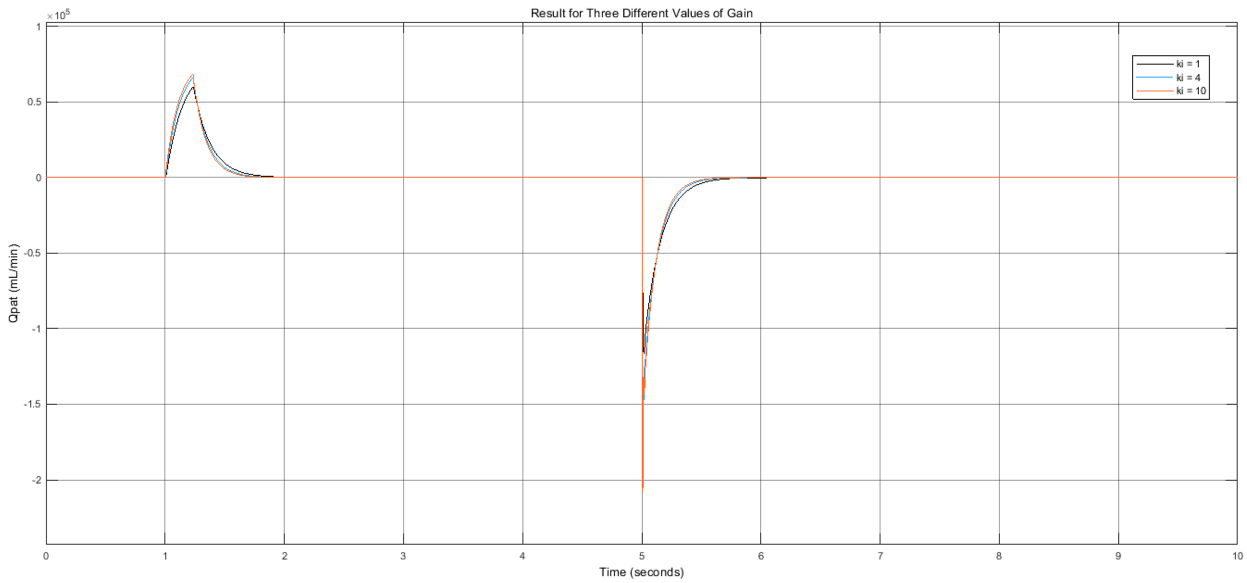


Figure 6 Output for Q_{pat}

From the simulation diagram, we observe that for $K_i = 10$, we have the fastest rise time value and lowest error. However, during the end of the cycle, at 5 second, it gives an unwanted oscillation that might result in false patient flow triggering. That is why, $K_i = 4$ is our preferable input as it meets the specifications while not creating undesired oscillation.

Therefore, the preferred PI controller is $4(s + 0.01)/s$.

Task-7:

Reproducing the results shown in Fig. 14 in the paper for linear and variable gain controllers and discussing the pros and cons of nonlinear control over linear control:

Expected output:

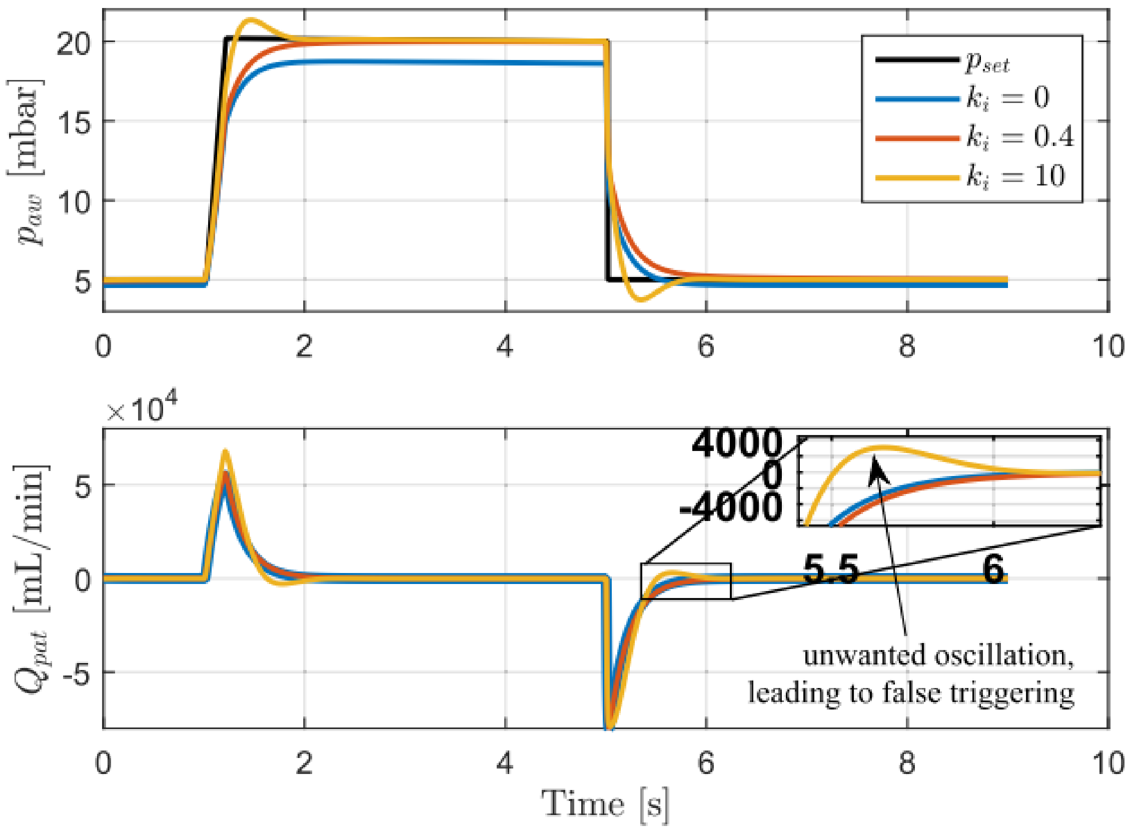


Figure 7. Expected output for Non-Linear case

For this we built the following circuit using Matlab-Simulink:

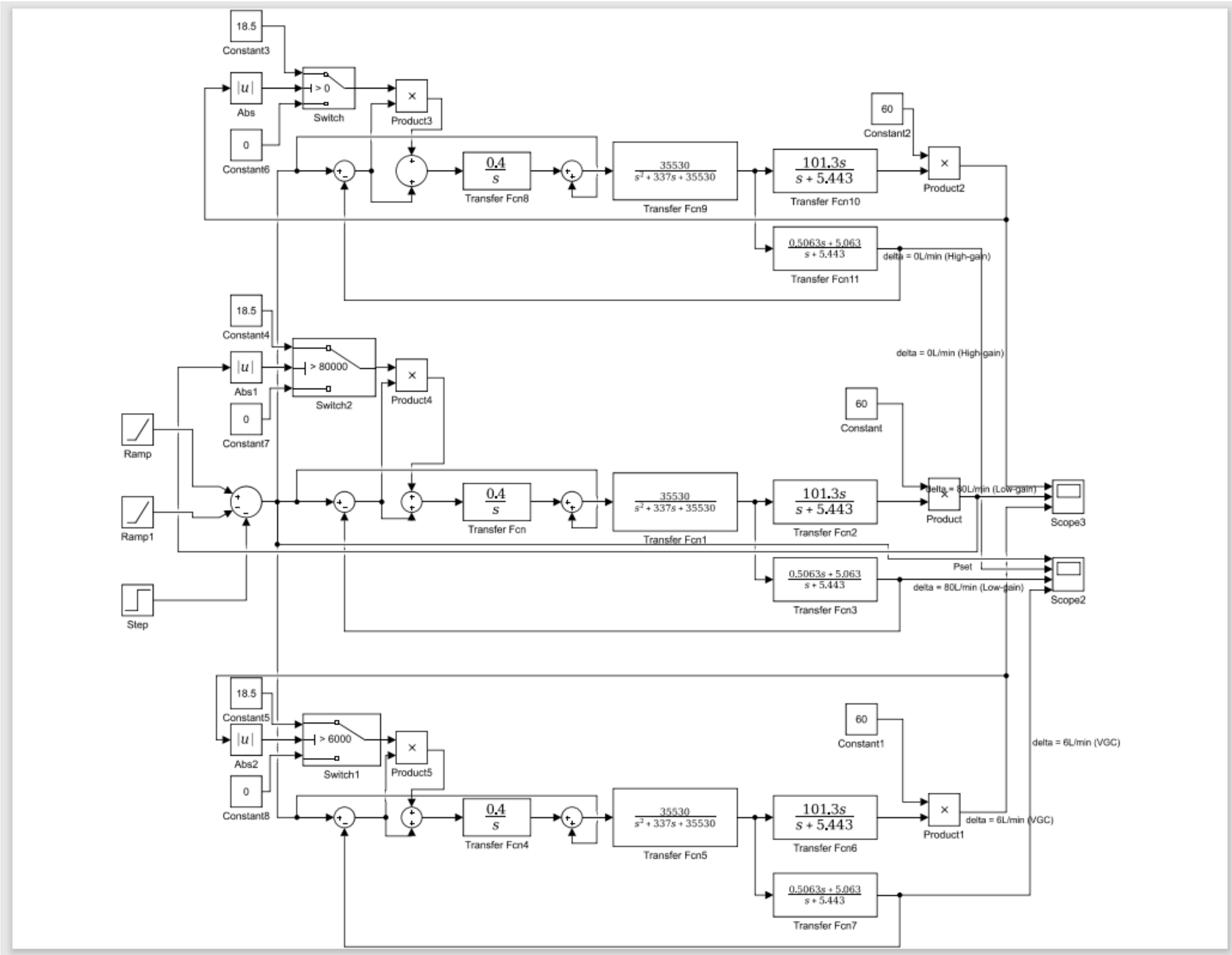


Figure 8. Model for Non-Linear case

Results:

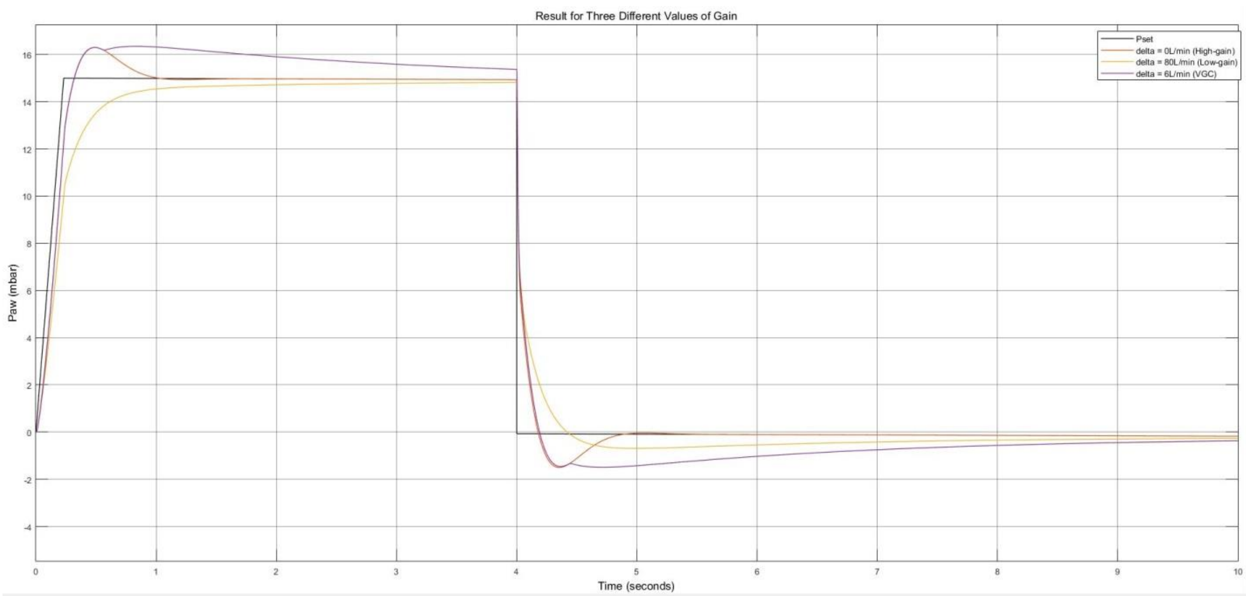


Figure 9: P_{aw} for three different values of gain

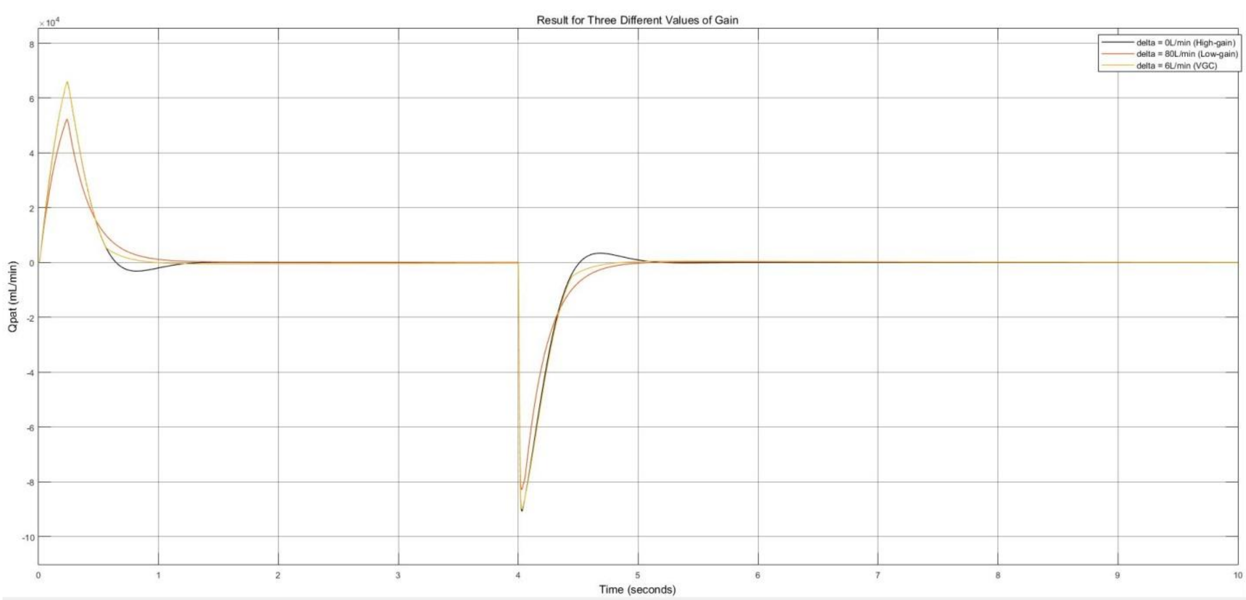


Figure 10: Q_{pat} for three different values of gain

Here, the flows are considered separately. For large flows, however, during pressure buildup and pressure release, a higher controller gain is desired to compensate for the pressure drop along the hose quickly. Therefore, switching the controller gain based on the patient flow Qpat is the optimum option.

$$\varphi(e, Q_{\text{pat}}) = \phi(Q_{\text{pat}})e$$

in which φ(Qpat) is designed to be large for large flows (high-gain) and small for small flows (low gain). The choice for the nonlinear gain can be a "switch" nonlinearity like:

$$\phi(Q_{\text{pat}}) = \begin{cases} 0, & \text{if } |Q_{\text{pat}}| \leq \delta \\ \alpha, & \text{if } |Q_{\text{pat}}| > \delta \end{cases}$$

Here,

when $\delta = 0$, it is operating with a high gain controller. As a result, the step response is expected to have low rise time and high overshoot.

When $\delta = 80$ L/min, it is operating with a low gain controller. As a result, the step response is expected to have high rise time but low overshoot.

When $\delta = 6$ L/min, it is operating with a variable gain controller.

The paper's novelty lies in the variable-gain control scheme, the controller can switch between a high-gain controller and a low-gain controller in order to balance the tradeoff between both of them. To ensure rapid response in pressure, high gain is desired. But it brings unwanted oscillations and needs to be eliminated like the case in low gain. To get the best of both, variable gain system is considered.

The controller is non-linear and it can identify the change in the gain regarding the pressure controller. If the flow exceeds a threshold, as shown in the above equation, gain is raised to compensate for the drop in the hose. The gain falls to ensure stability in the flow without overshoot during the time when the lung is full and Qpat reaches nearly to zero.

There also exists some drawback of the nonlinear system. The optimal value of delta obtained from the figure of $\boldsymbol{Qpat}$, introduces a high error in the pressure $\boldsymbol{Pa}$**w.** It also 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 it yields more realistic results than the linear one.

Task-8:

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

Here, we changed some of the parameters of the system including C_{lung} and R_{lung}, system parameter R_{leak} and the parameter R_{hose}. Same blower is used and therefore w_n is same.

The values are given here:

P_{lung}	C_{lung}
0.001	10
0.005	20
0.1	40

R_{leak}	R_{hose}
0.03	0.003
0.06	0.0045
0.09	0.009

The outputs are as follows:

Linear:

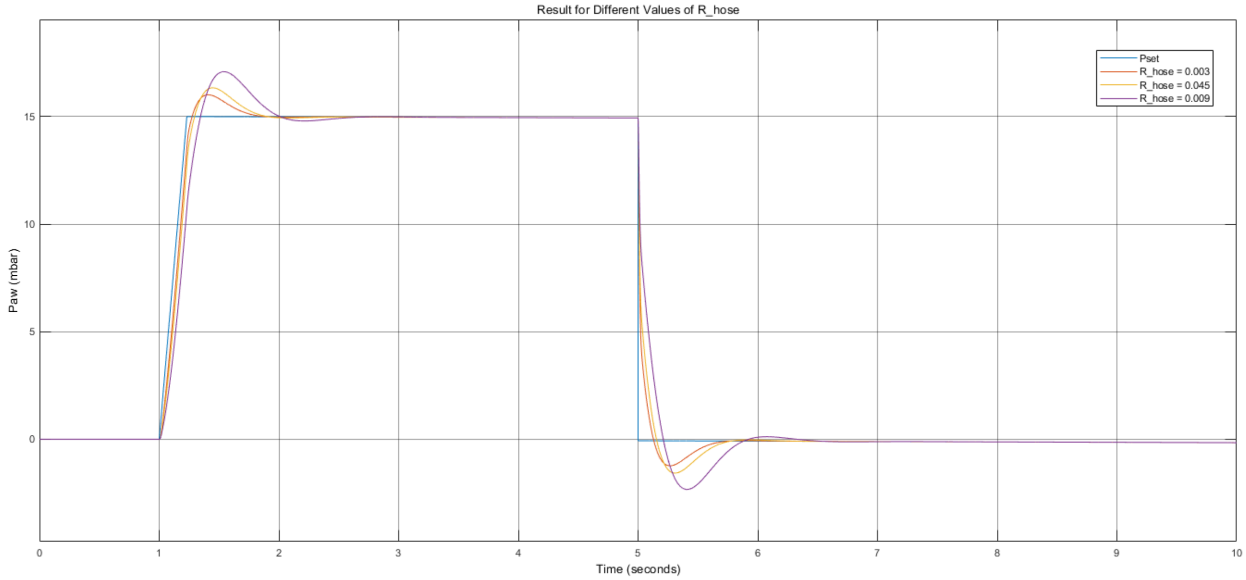


Figure 11. P_{aw} for changing R_{hose},

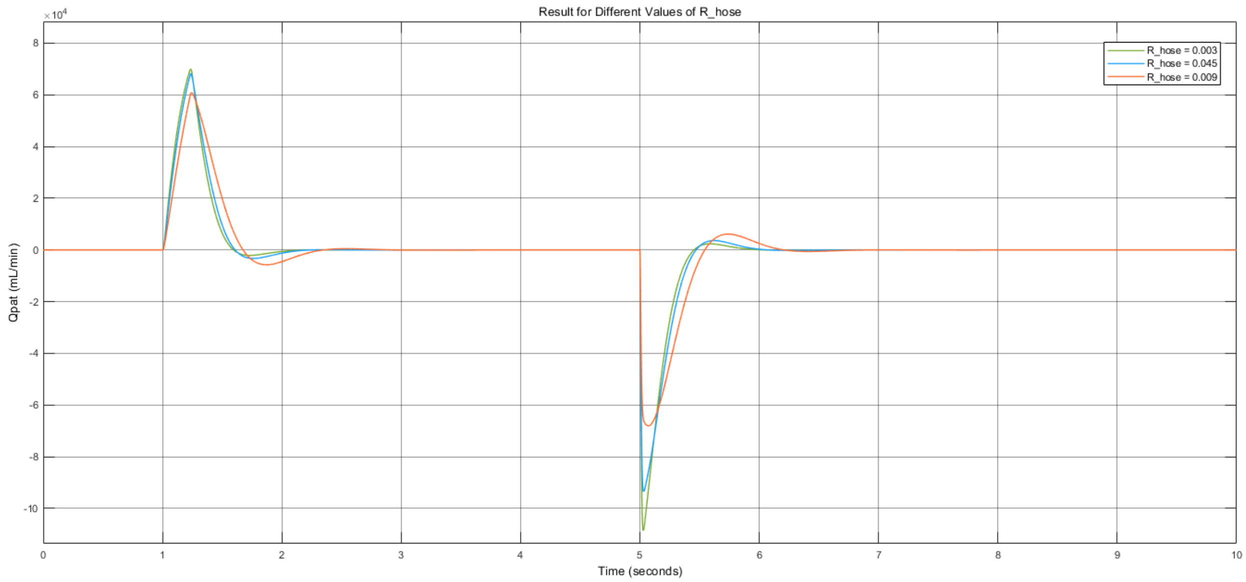


Figure 12. Q_{pat} for changing R_{hose},

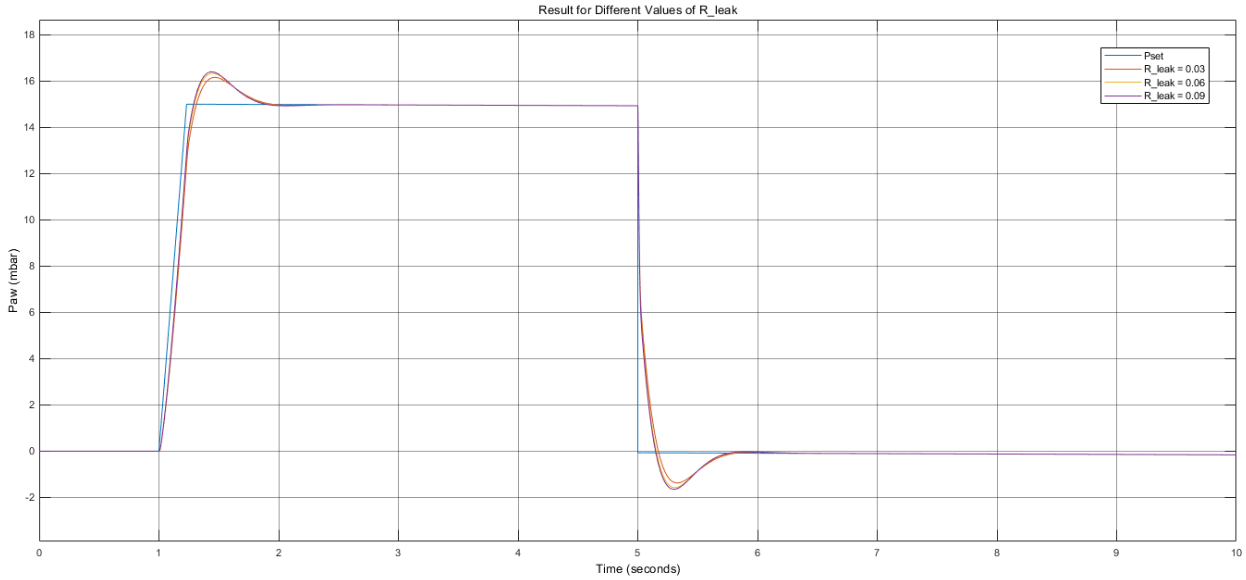


Figure 13. P_{aw} for changing R_{leak}

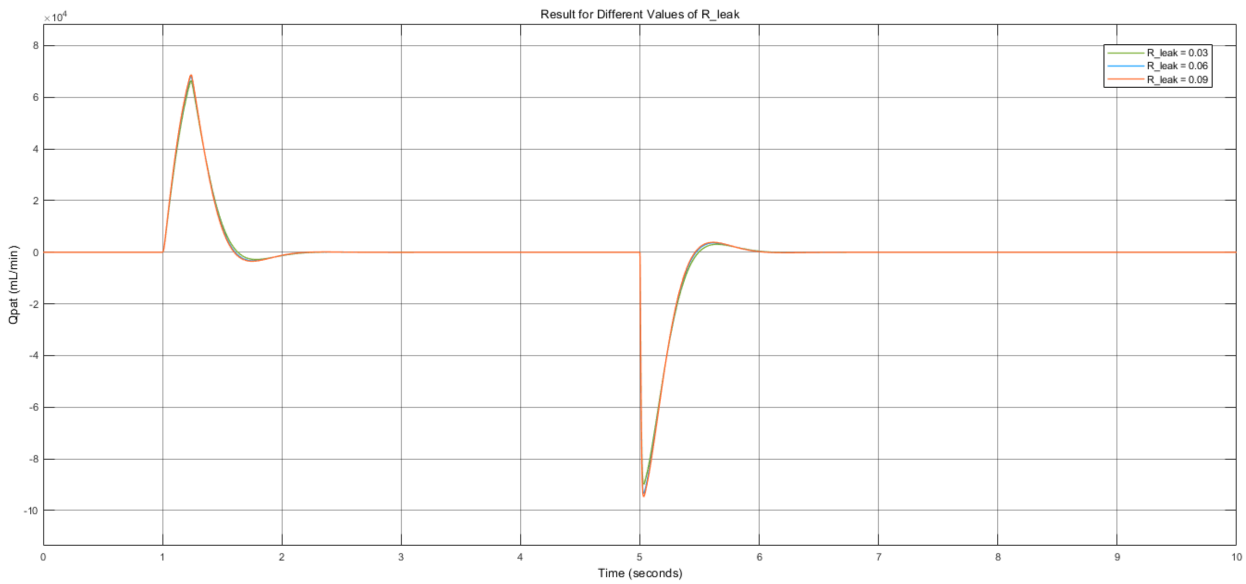


Figure 14. Q_{pat} for changing R_{leak}

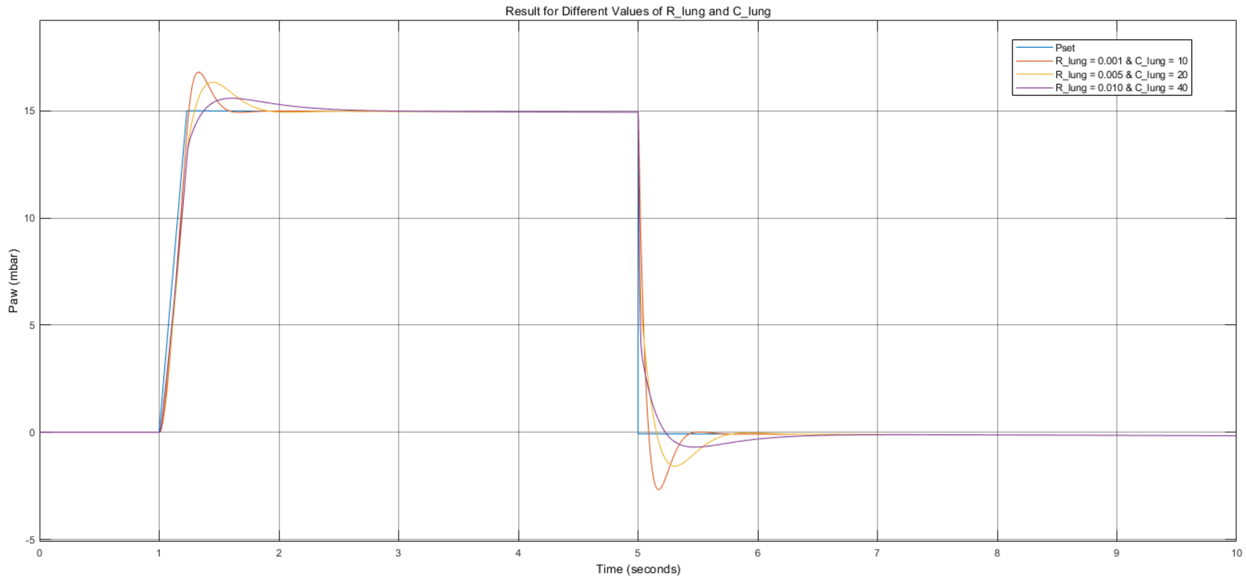


Figure 15. P_{aw} for changing R_{lung}, C_{lung}

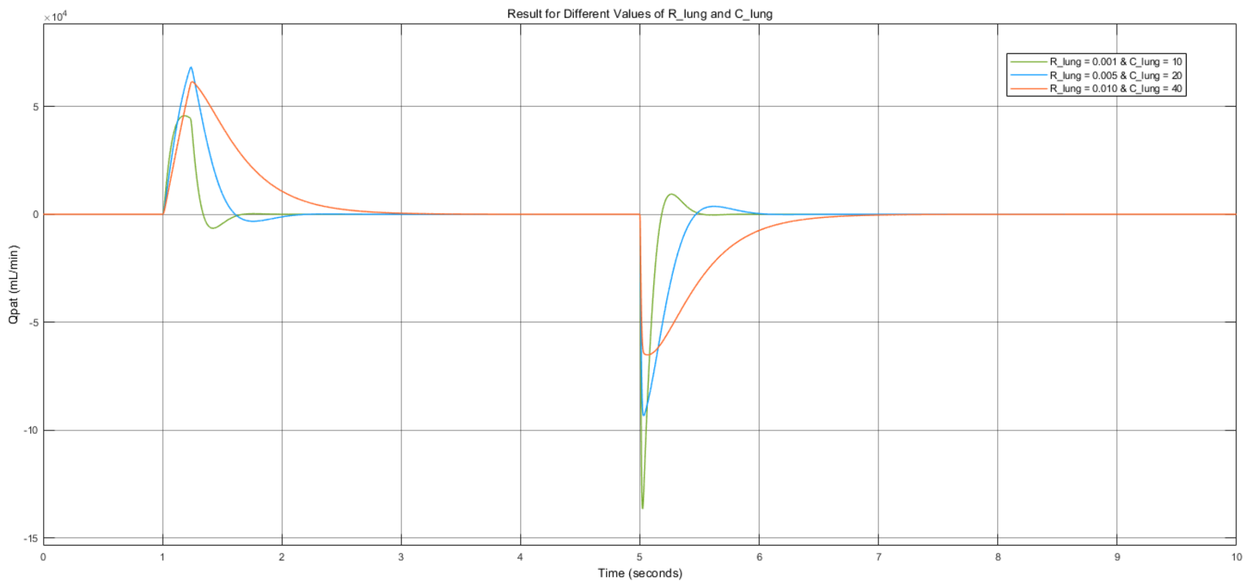


Figure 16. Q_{pat} for changing R_{lung}, C_{lung}

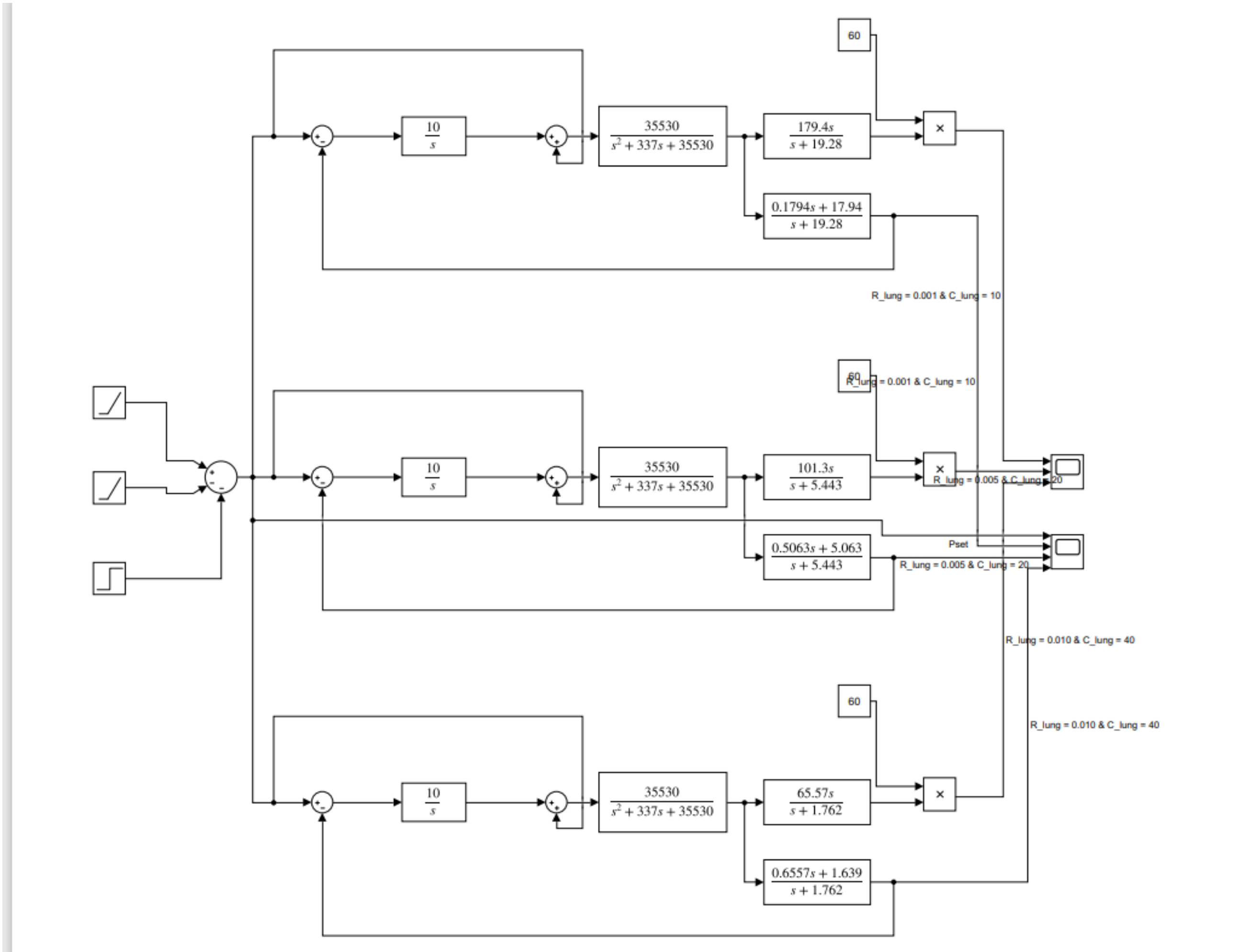


Figure 17. Model for changing R_{lung} C_{lung}

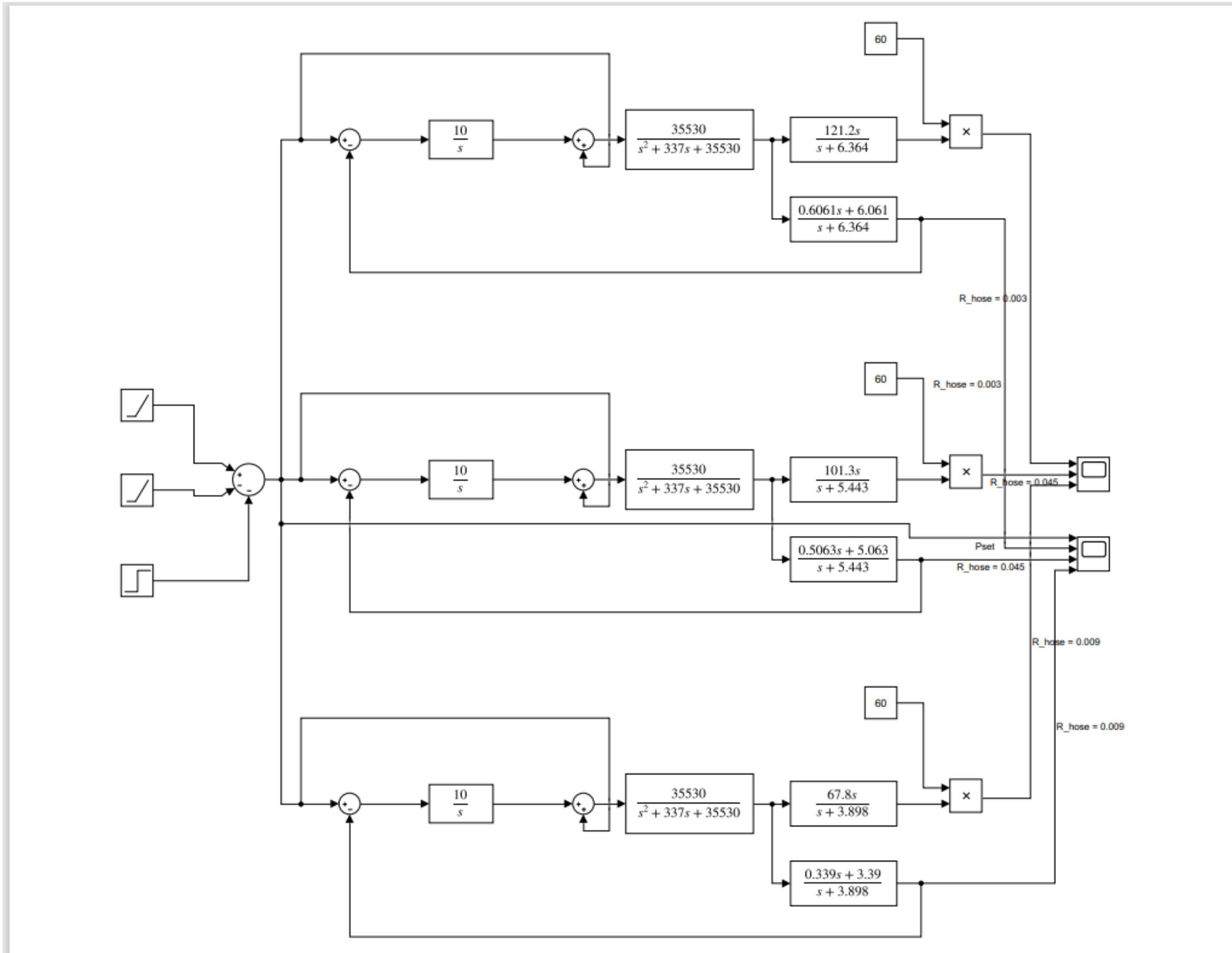


Figure 18. Model for changing R_{hose}

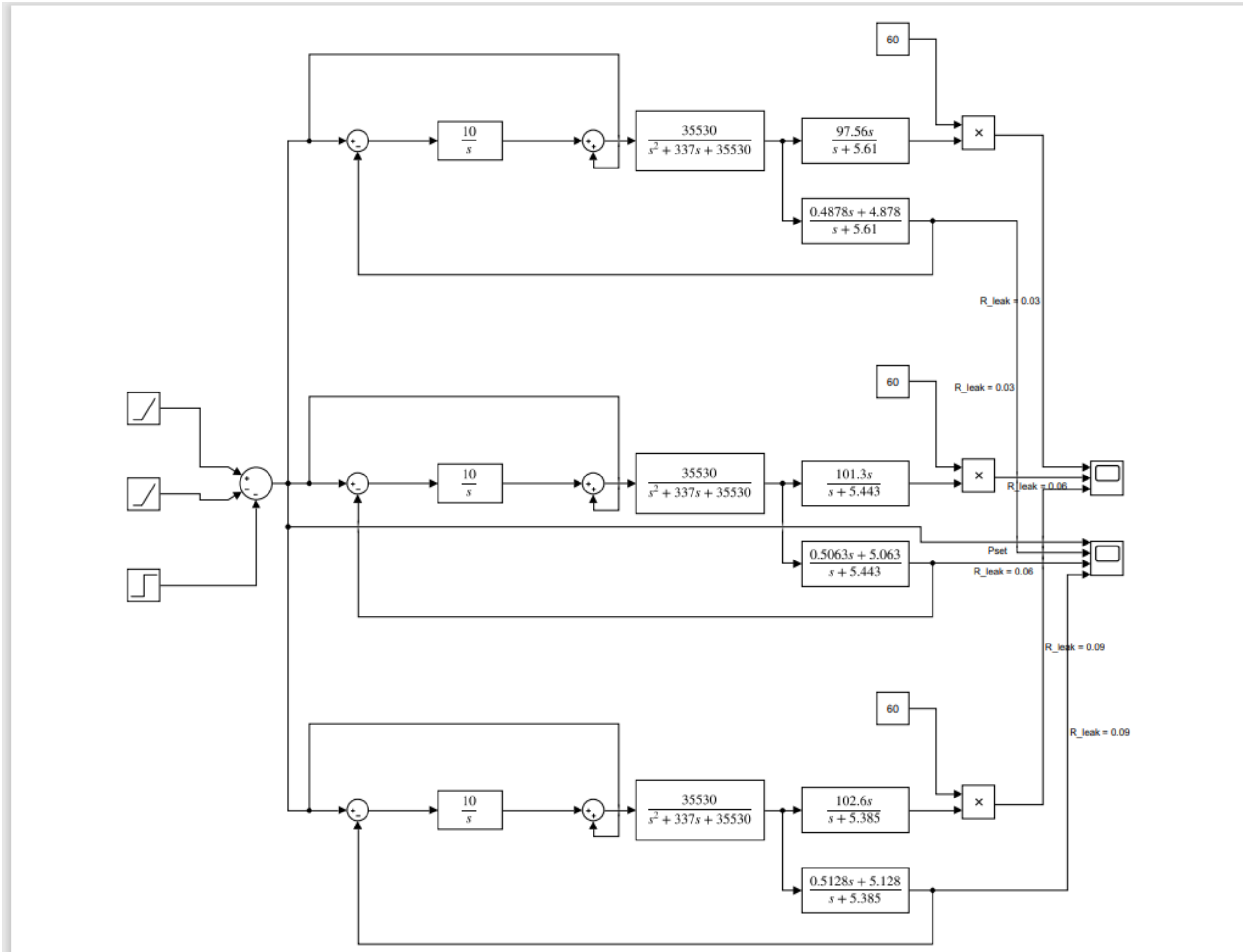


Figure 19. Model for changing R_{leak}

Non-Linear:

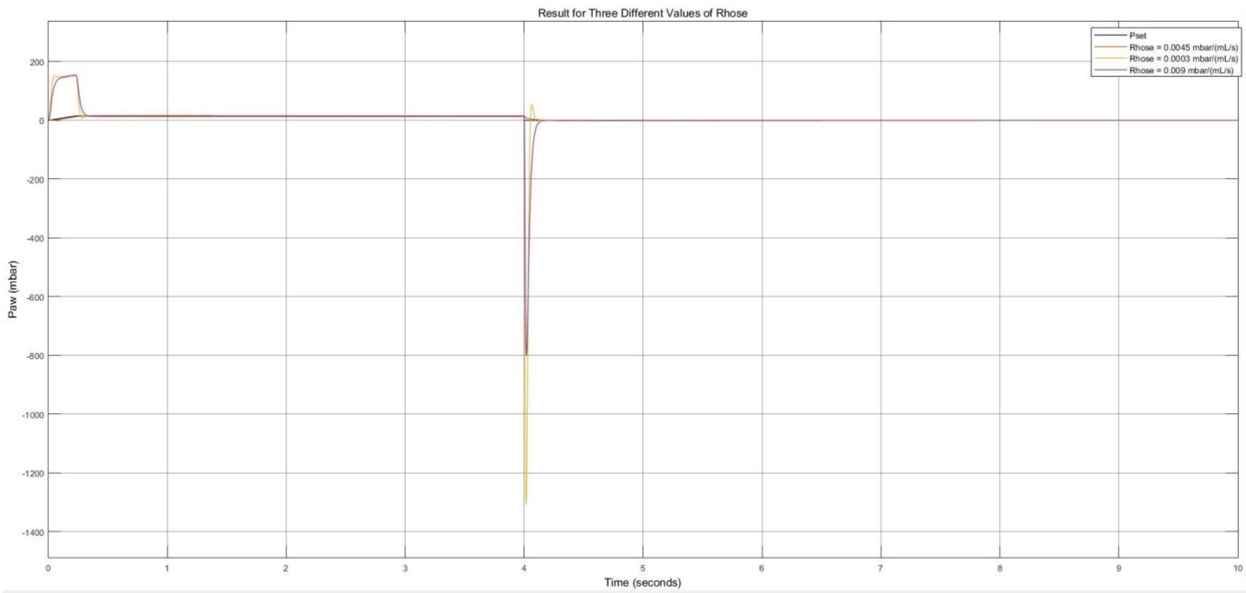


Figure 20. P_{aw} for changing R_{hose}

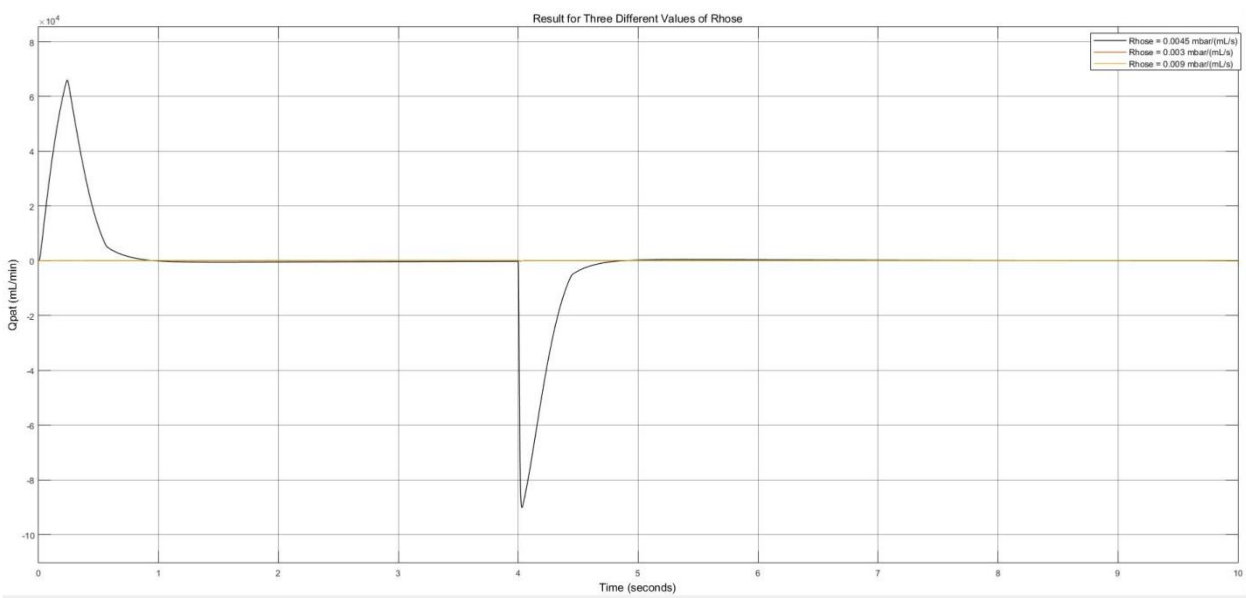


Figure 21. Q_{pat} for changing R_{hose}

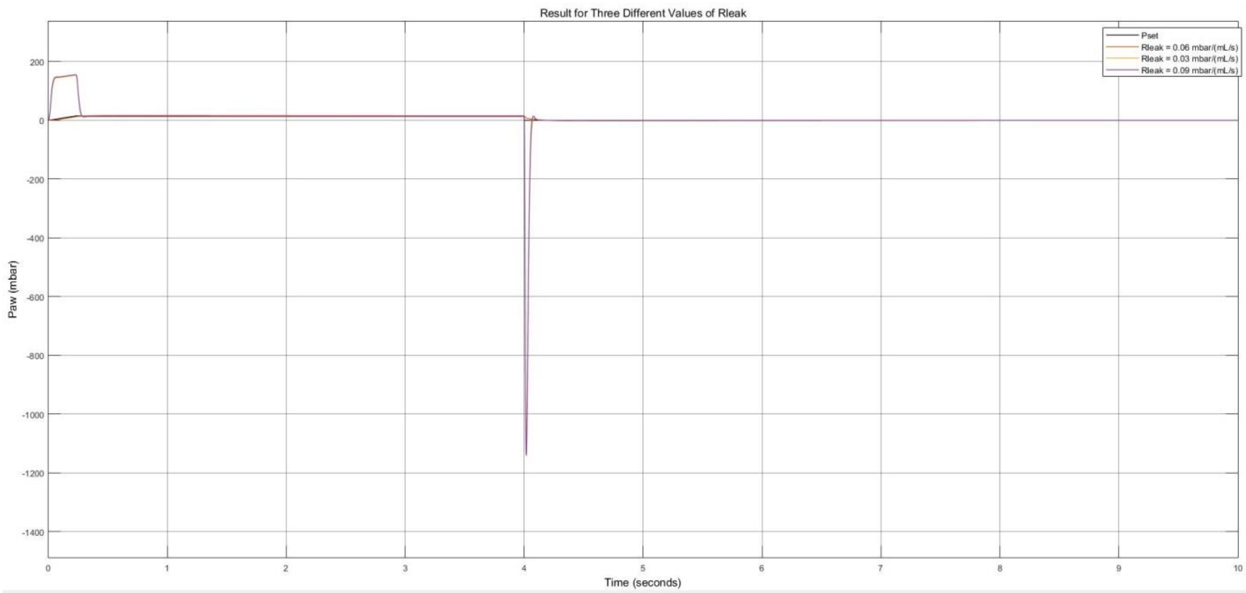


Figure 22. P_{aw} for changing R_{leak}

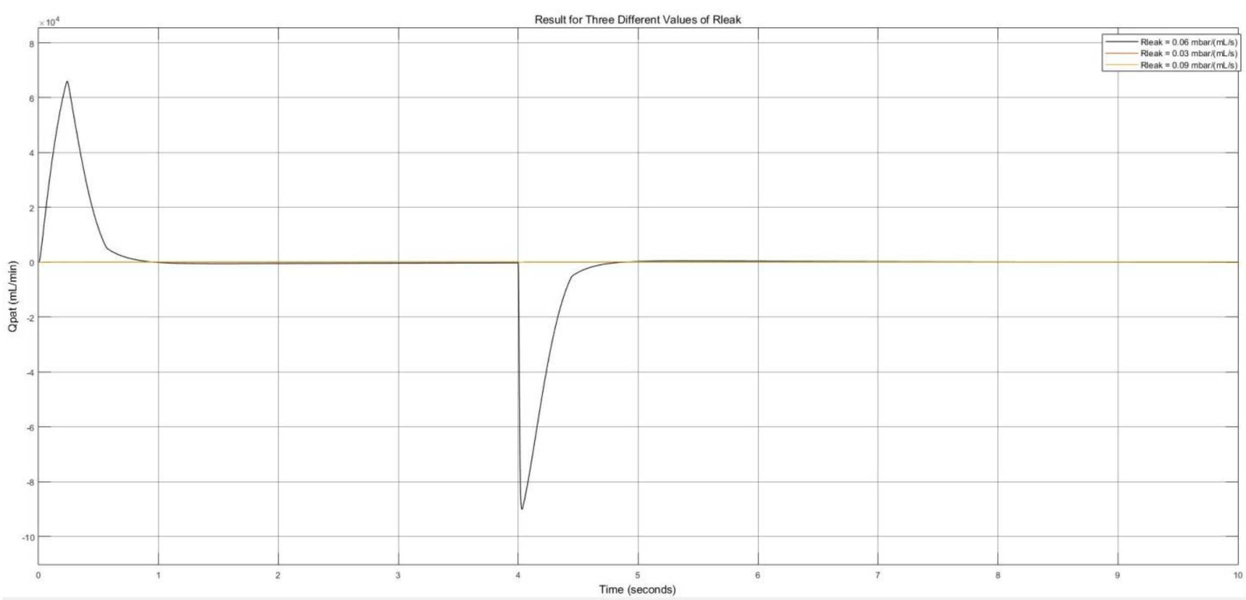


Figure 23. Q_{pat} for changing R_{leak}

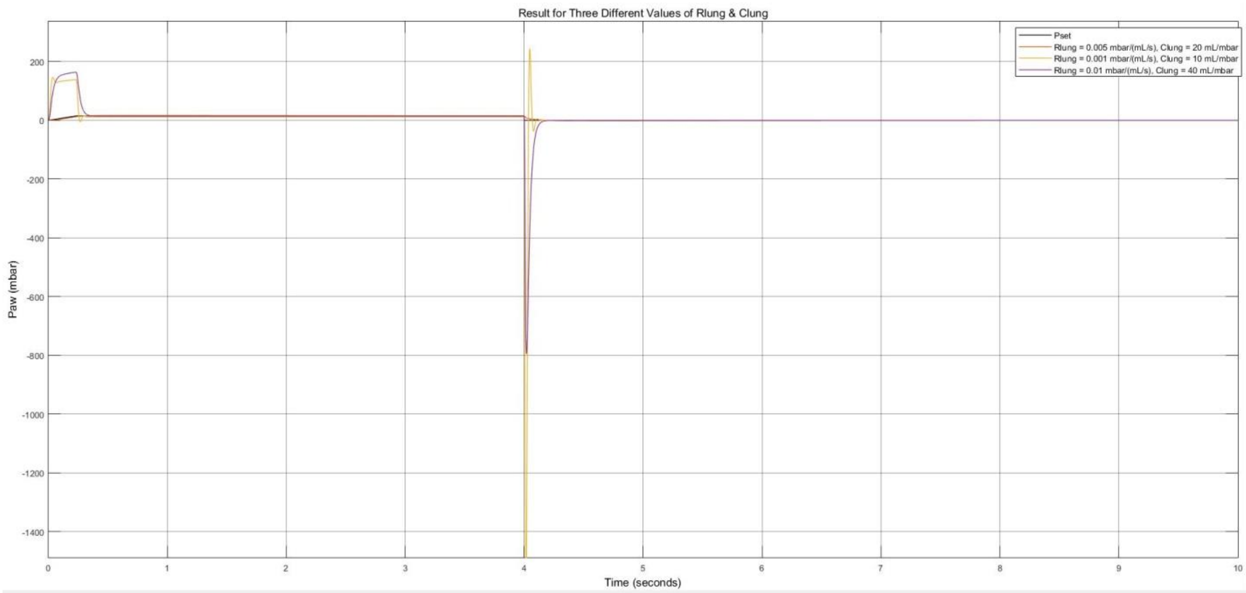


Figure 24. P_{aw} for changing R_{lung}, C_{lung}

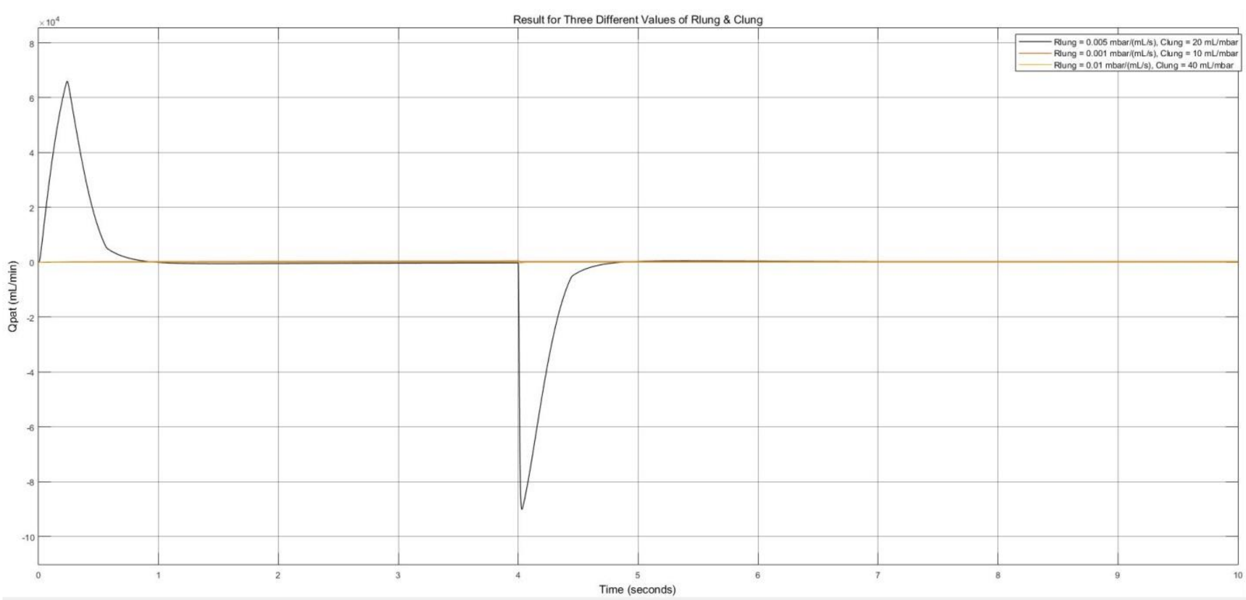


Figure 25. Q_{pat} for changing R_{lung}, C_{lung}

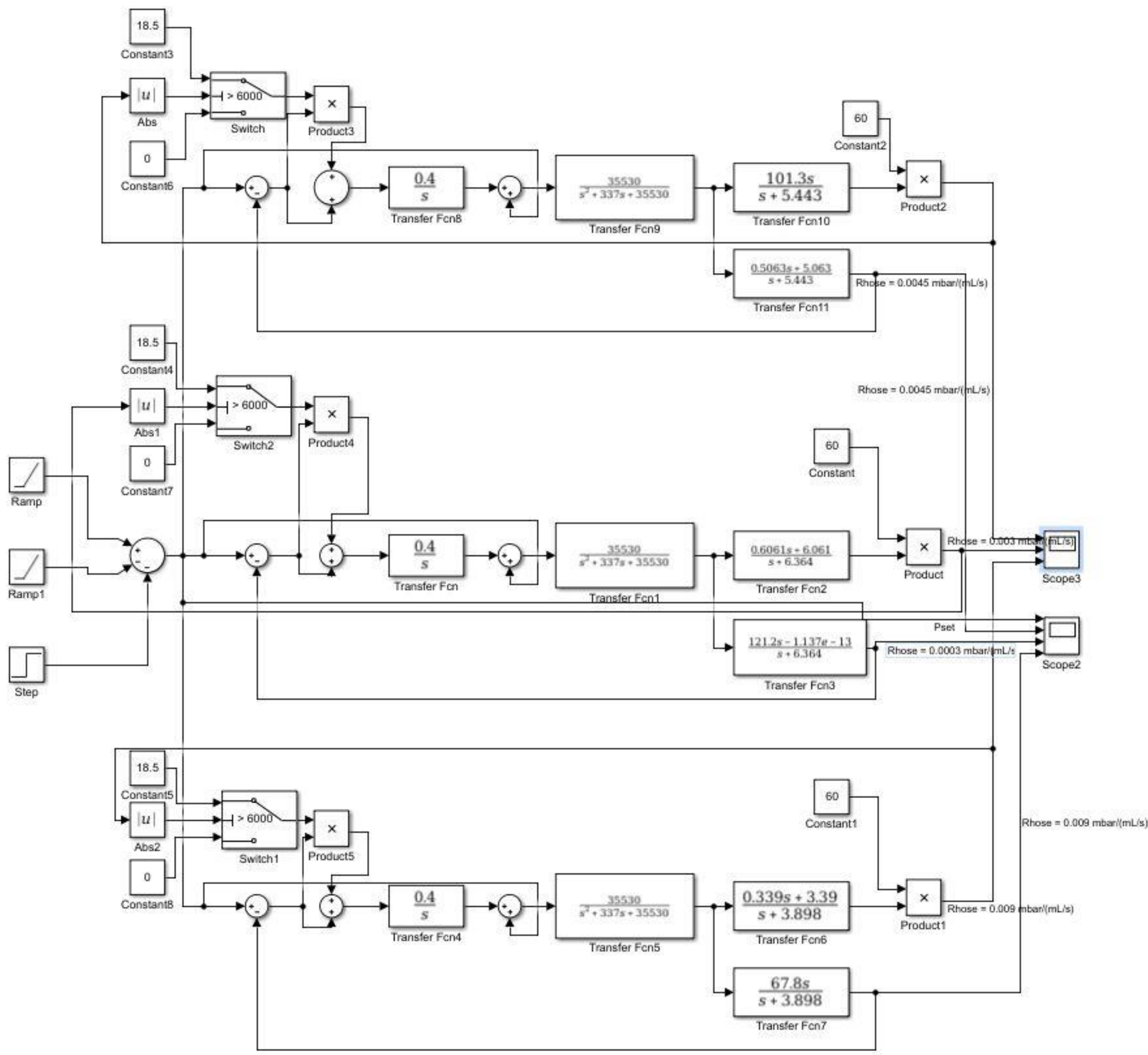


Figure 26 Model for changing R_{hose}

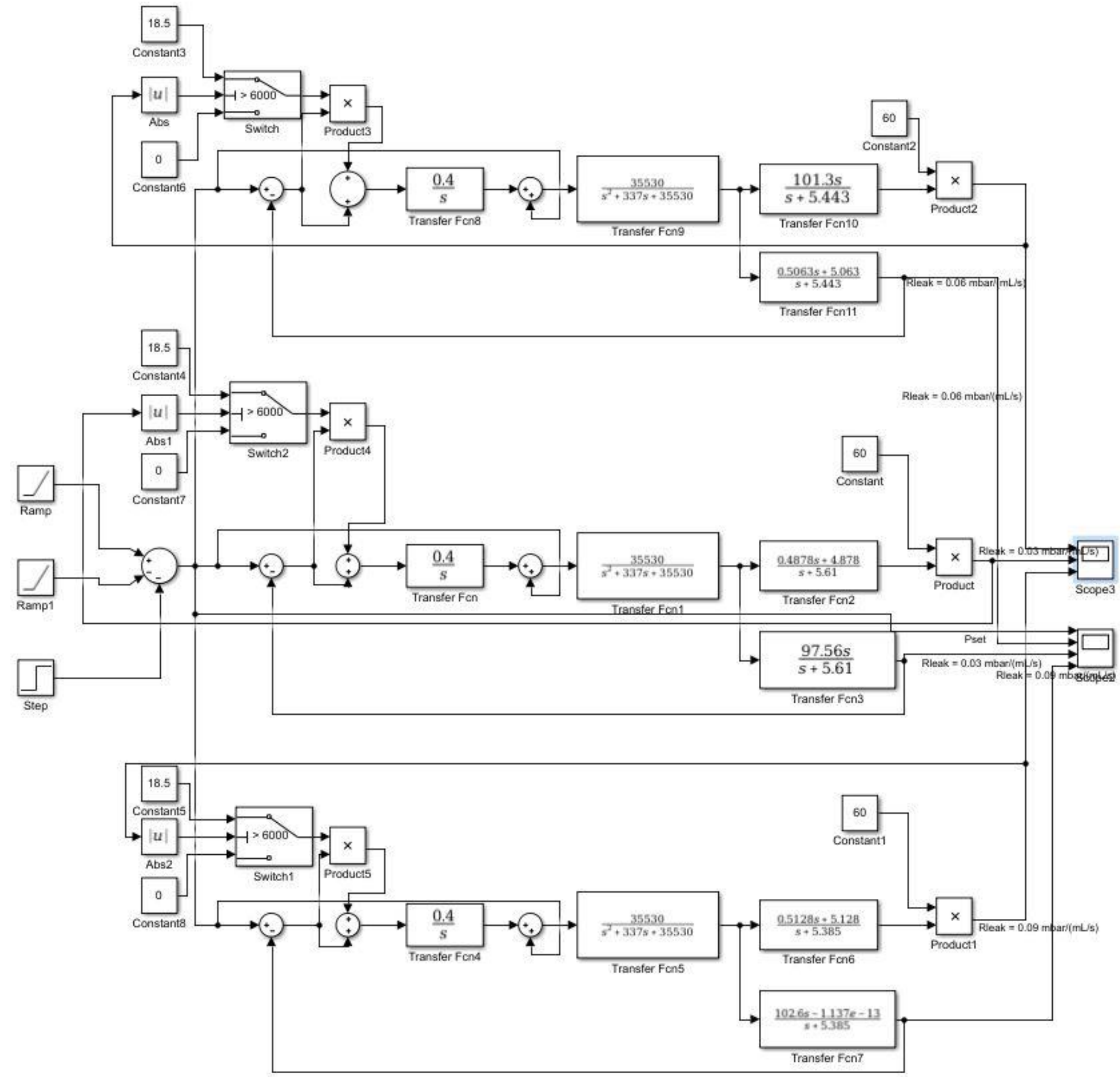


Figure 27. Model for changing R_{leak}

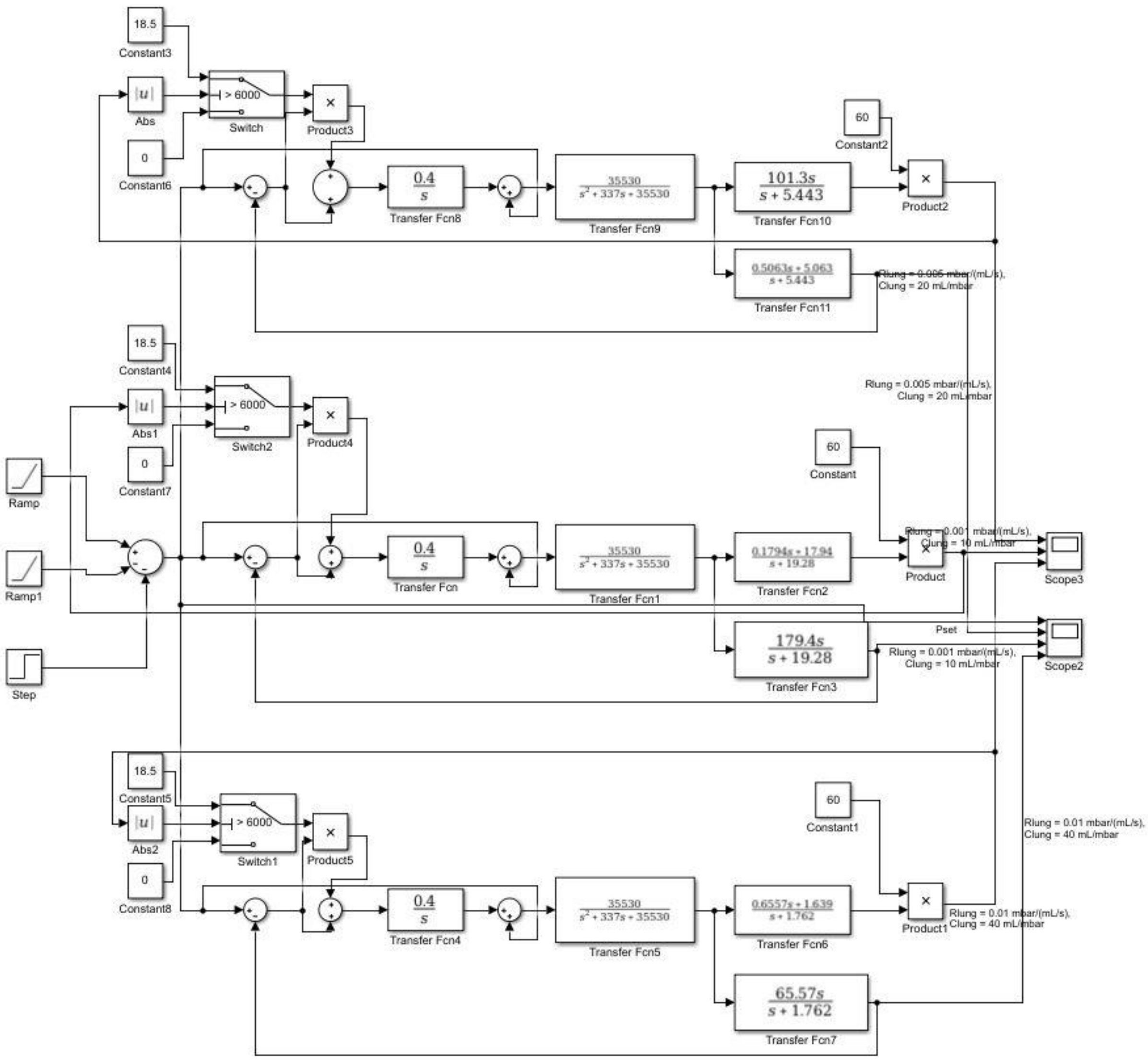


Figure 28. Model for changing R_{lung}, C_{lung}

It is apparent that the linear-case results are better than that of Non-linear condition. It is to be noted that increasing Clung results in increased amount of rise time, but decreased overshoot. Also, the other results are expected, for example increasing R_{hose} increases Q_{pat}, as the patient will get higher flow of air.

Conclusion:

In this project, we replicated the results presented in [1] to build a variable-gain controller system for ventilation. The result matched with the result presented in the paper, so we can say that it has been reproduced efficaciously. In the last task, results are shown by changing the parameters and displaying results accordingly. It is evident that linear system provides better result in this case.

References:

[1] B. Hunnekens, S. Kamps and N. Van De Wouw, "Variable-Gain Control for Respiratory Systems," in IEEE Transactions on Control Systems Technology, vol. 28, no. 1, pp. 163-171, Jan. 2020, doi: 10.1109/TCST.2018.2871002.

***** Nise, Norman S. *Control Systems Engineering*. John Wiley & Sons, 2020.