ISTANBUL TECHNICAL UNIVERSITY
DEPT. OF CONTROL AND AUTOMATION ENGINEERING
KON305E – PROGRAMMING TECHNIQUES IN CONTROL
FINAL PROJECT

Team #20

Students,

AGAH ENES SOYALP 040190616

AHMET FATİH OLGUNÖZ 040190766

FAİK ENES ALTUN 040190614

HALİL FARUK KURT 040200753

SALİH SERDAR ÖNCEL 040180604

27/01/2022

First Question: Modelling of the Lungs

In this part, we will model the lungs without the addition of ventilators. This model will then be reduced to a much simpler first degree model.

A-

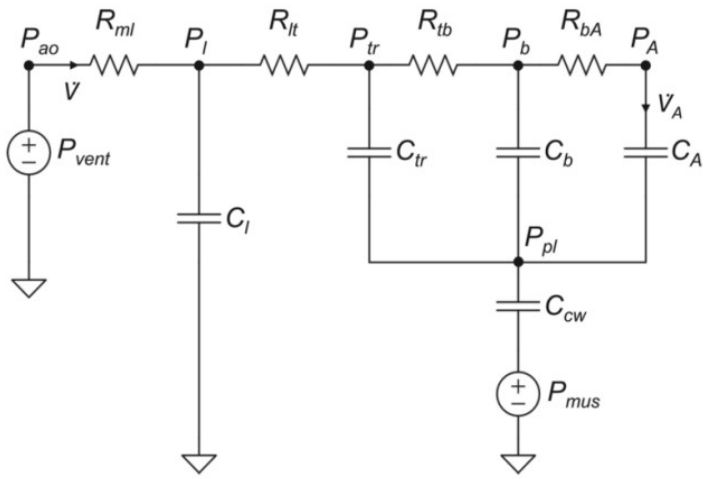


Figure 1 - The mechanics model of the lungs

Respiratory system is modelled with electrical components in Simulink.

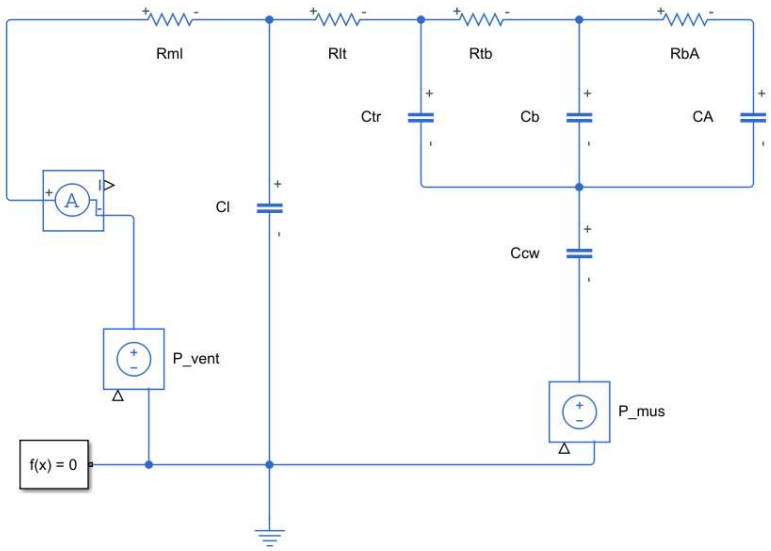


Figure 2 - Simulink model of the system's mechanics model.

B-

Pressure applied by the diaphragm muscle is modelled as a MATLAB function in Simulink.

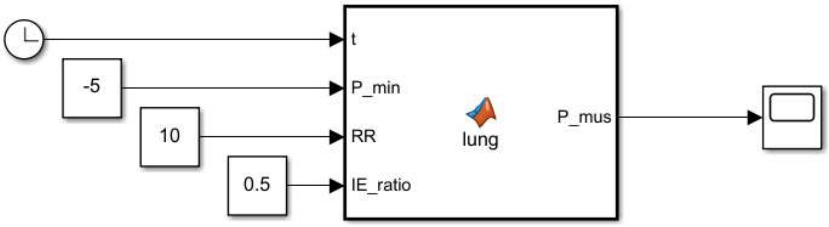


Figure 3 - Simulink Subsystem

In this model there are four inputs. Three of them are $P_{mus,min}$, RR and IE_{ratio}. Fourth one is a clock for the signal to be periodic.

The code in the function is as follows:

```
function P_mus=lung(t,P_min,RR,IE_ratio)
    T=60/RR;
    TI=IE_ratio*T/(1+IE_ratio);
    TE=T-TI;
    tau=TE/5;
    if (0<=rem(t,T) && rem(t,T)<=TI+0.06)
        P_mus=P_min/(TI*TE)*(T*rem(t,T)-rem(t,T)^2);
    else
        if (TI+0.06<=rem(t,T) && rem(t,T)<=T)
            P_mus=P_min/(1-exp(-TE/rem(t,T)))*(exp(-(rem(t,T)-TI)/tau)- exp(-(TE/tau)));
        else
            P_mus=0;
        end
    end
end
```

Output of the muscle pressure can be observed in this graph.

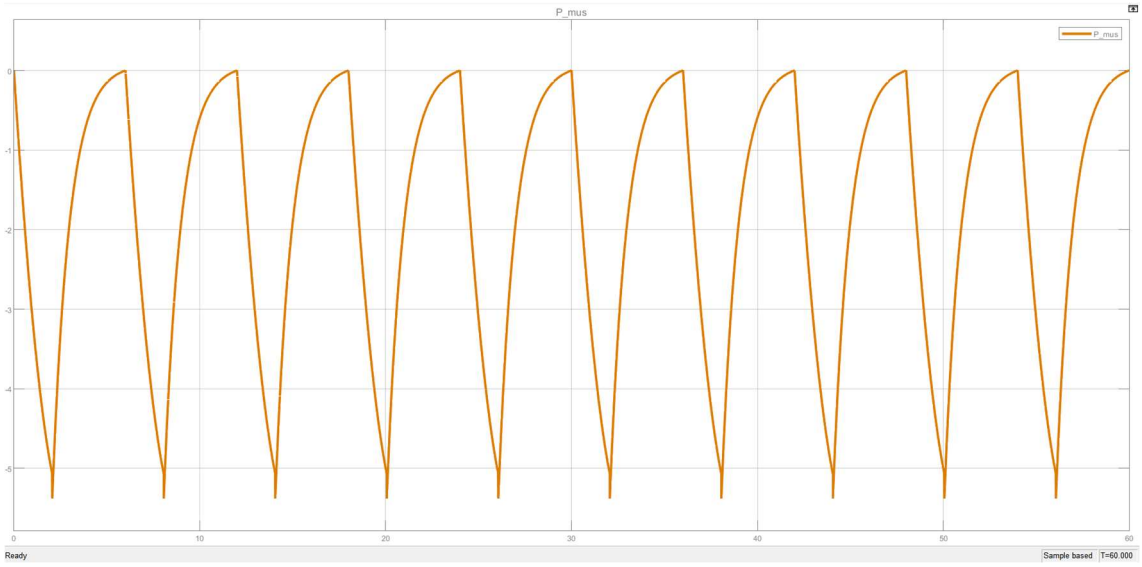


Figure 4 - Pmus Output

C-

Simulink model for the for $P_{vent} = 0$

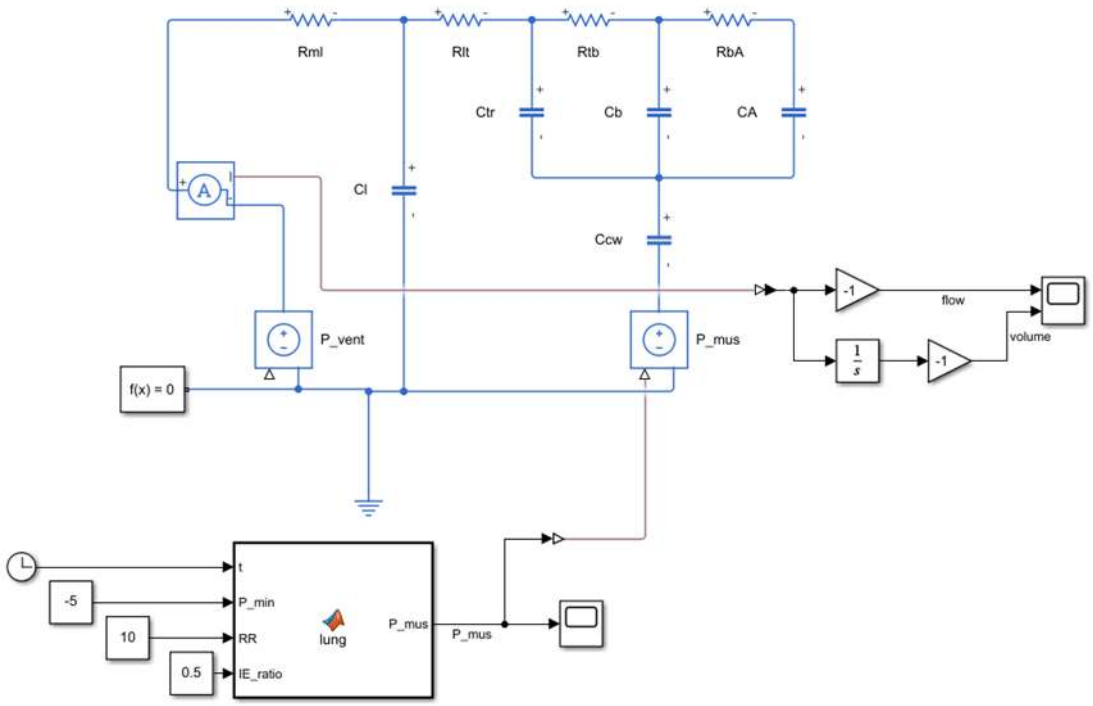


Figure 5 - Lung system and subsystem simulink

In order to get the model simulation to work without problems, a fixed step size (0.0005) was set. The output graphs came out as expected.

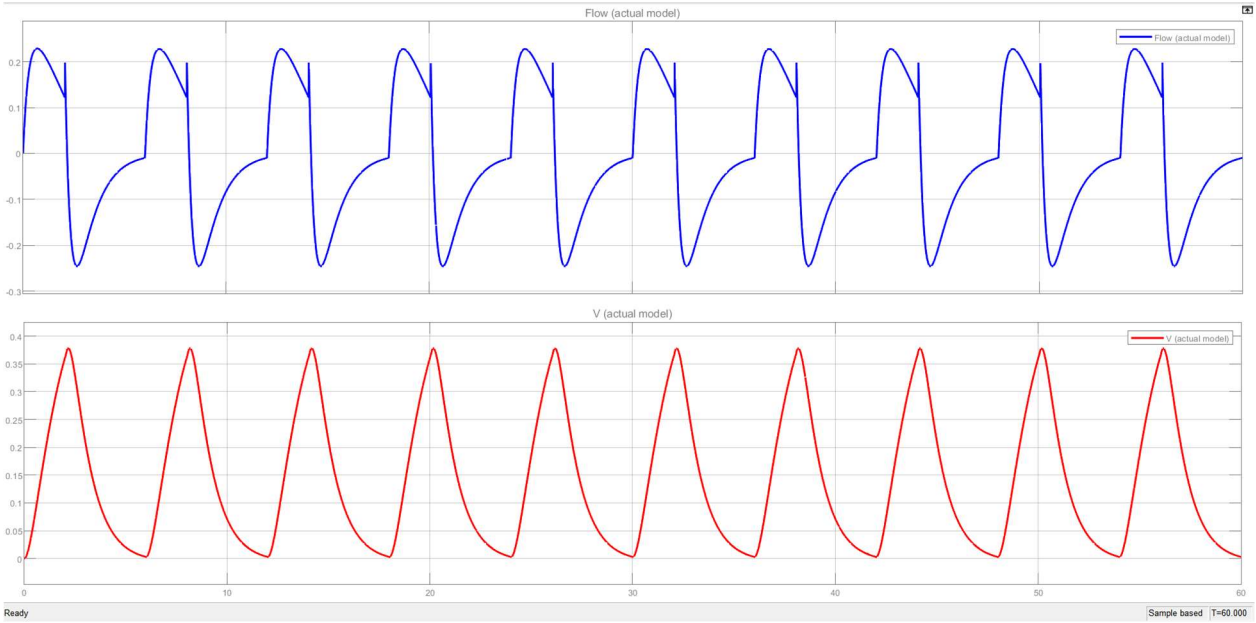


Figure 6 - Volume and Flow Graph

D-

Transfer function is observed via Model Linearizer tool. After selecting open loop input and open loop output as P_{mus} and V(s) we obtain the transfer function.

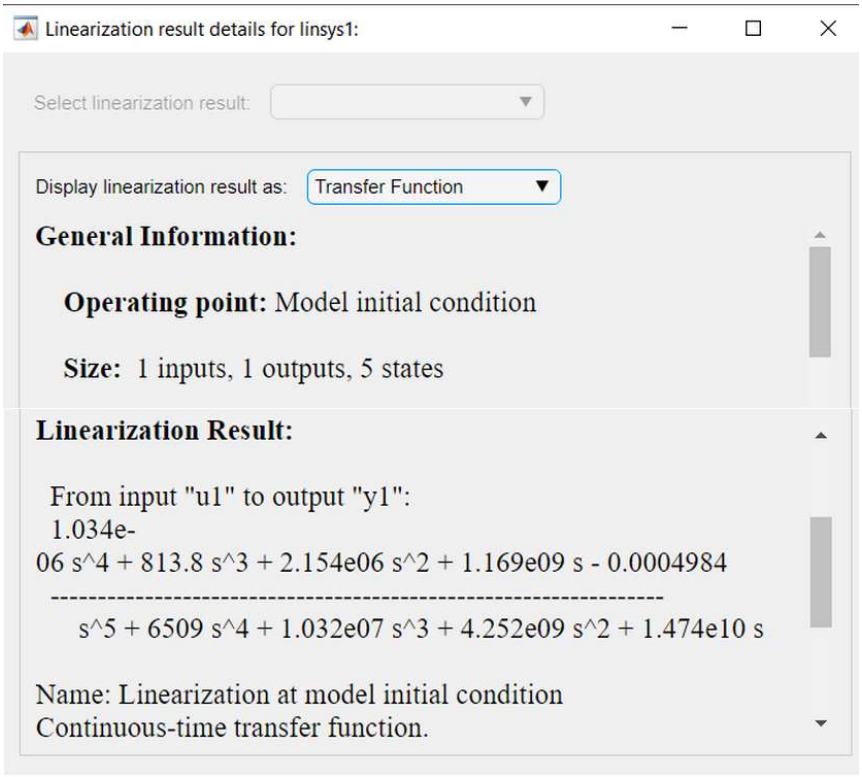


Figure 7 - Open Loop Transfer Function between P_mus and V_s

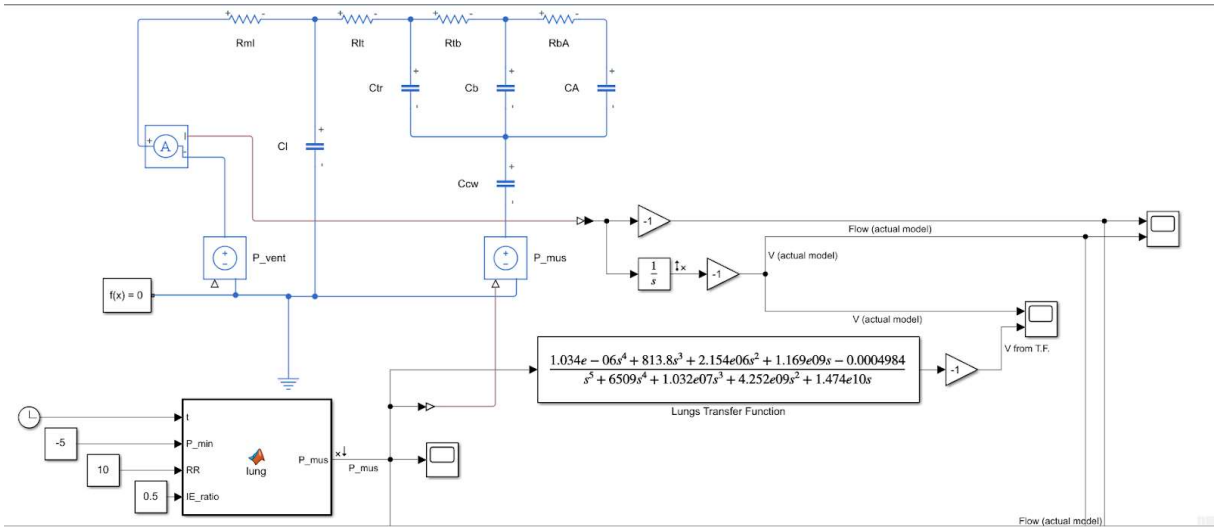


Figure 8 - Simulink with transfer function

It is seen that actual system and output of the transfer function is very similar to each other.

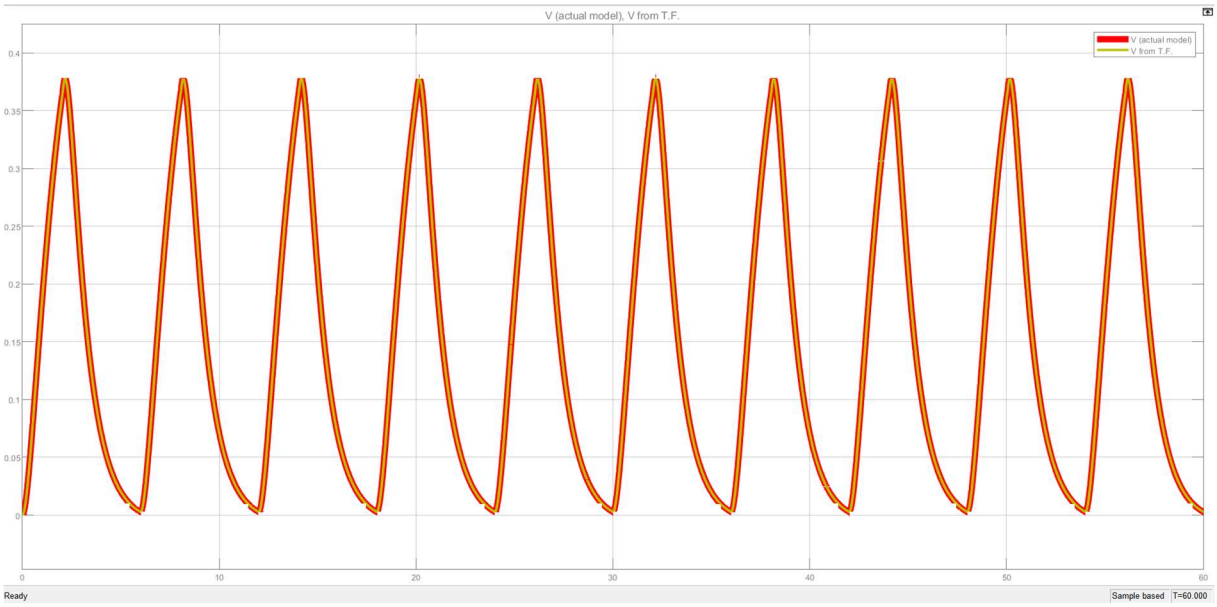


Figure 9 - Output Graph of the Actual System and the system with Transfer Function

E-

MATLAB code for finding the dominant pole:

final1.m | polezerofinder.m

```
clear;
clc;

num=[1.034e-06 813.8 2.154e06 1.169e09 -0.0004984];
den=[1 6509 1.032e07 4.252e09 1.474e10 0];

Gs= tf(num,den);

zpk(Gs)
pzmap(Gs)
```

Transfer Function of the system expressed with pole, zero and gain:

Command Window

```
ans =

  1.034e-06 s (s+7.87e08) (s+1885) (s+762.2)
  ------------------------------------------
    s (s+4370) (s+1487) (s+649) (s+3.496)

Continuous-time zero/pole/gain model.

fx >>
```

Observing the dominant pole via pole zero map. Poles and zeros that are not close to the jw axis are neglected. However, their impact to the transfer function should be calculated as residues.

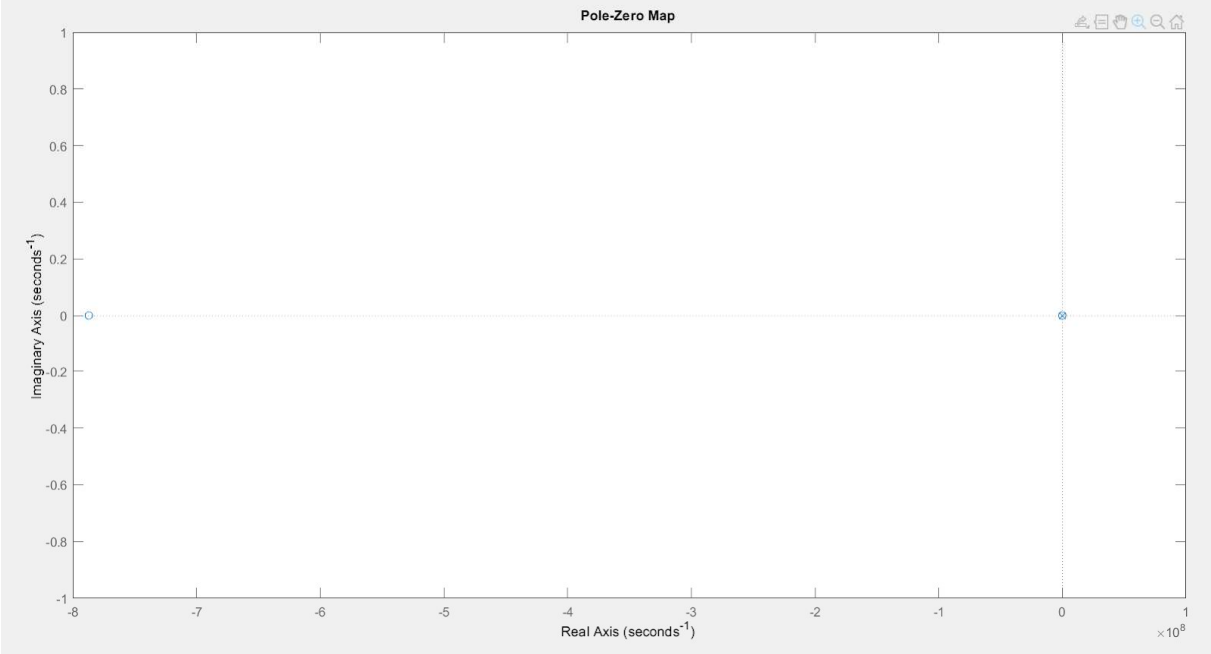


Figure 10 - PZMAP for G(s)

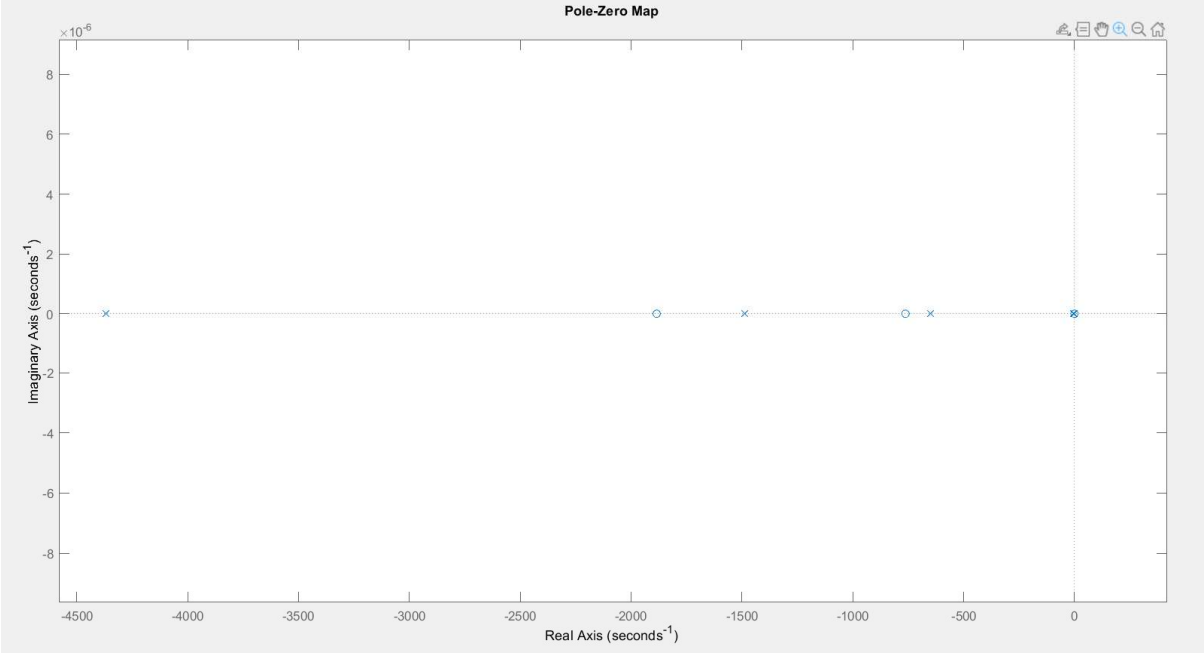


Figure 11 - PZMAP for G(s) Zoomed In

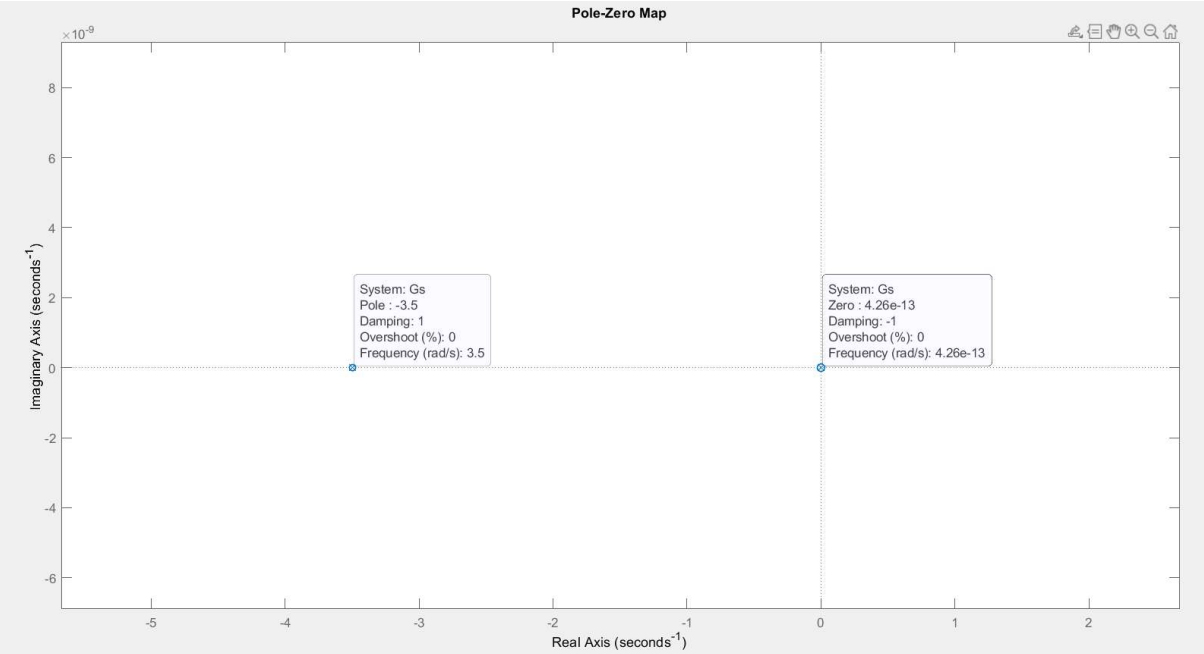


Figure 12- PZMAP for G(s) Zoomed In Further

It is easy to see that the system actually has one dominant pole at -3.5, meaning that it can be reduced to a first order model. The pole-zero pair at the origin takes each other out.

F-

Expressing the transfer function as a first order system by reducing the order of the transfer function to a first order system. The residues are taken into account to fix the gain.

$$\frac{1.034x10^{-6}x - 7.87x10^{8}\ x\ 1885\ x\ 762.2}{(-4370)\ x(-1487)x(-649)} = 0.27722$$

$$\frac{0.27722}{s+3.5} = \frac{\frac{1}{R}}{s+\frac{1}{RC}} => R = 3.6\Omega\ and\ C = \ 0.07939F$$

G-

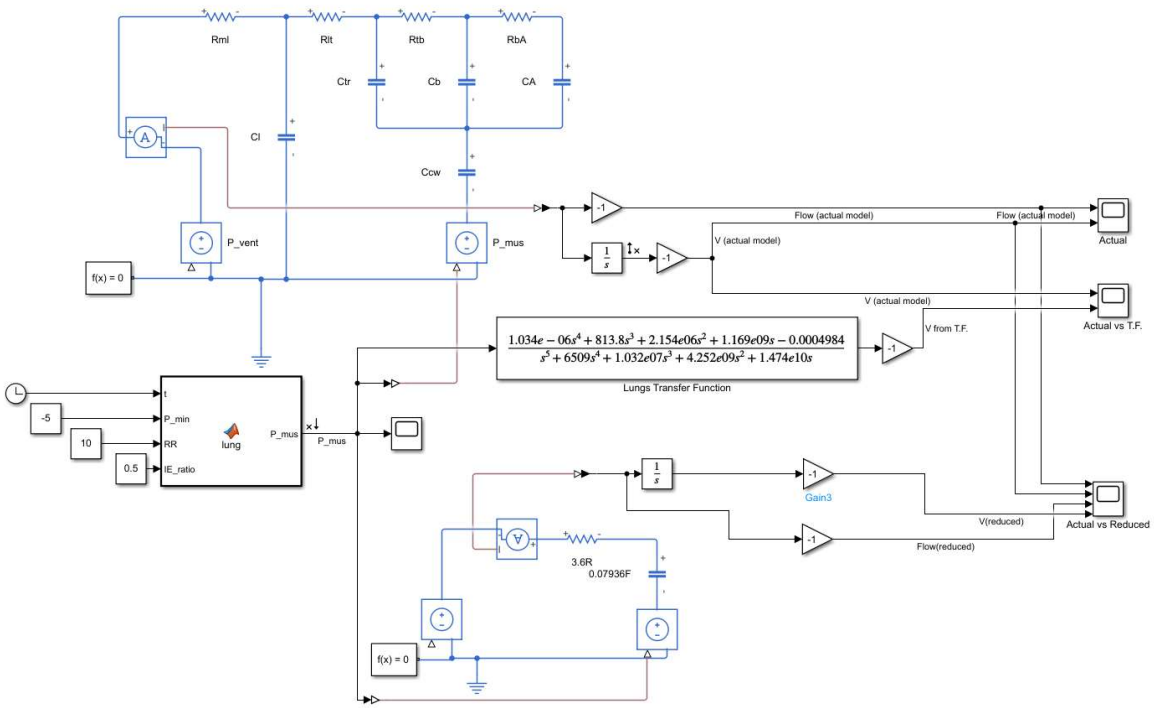


Figure 13 - Simulink Model

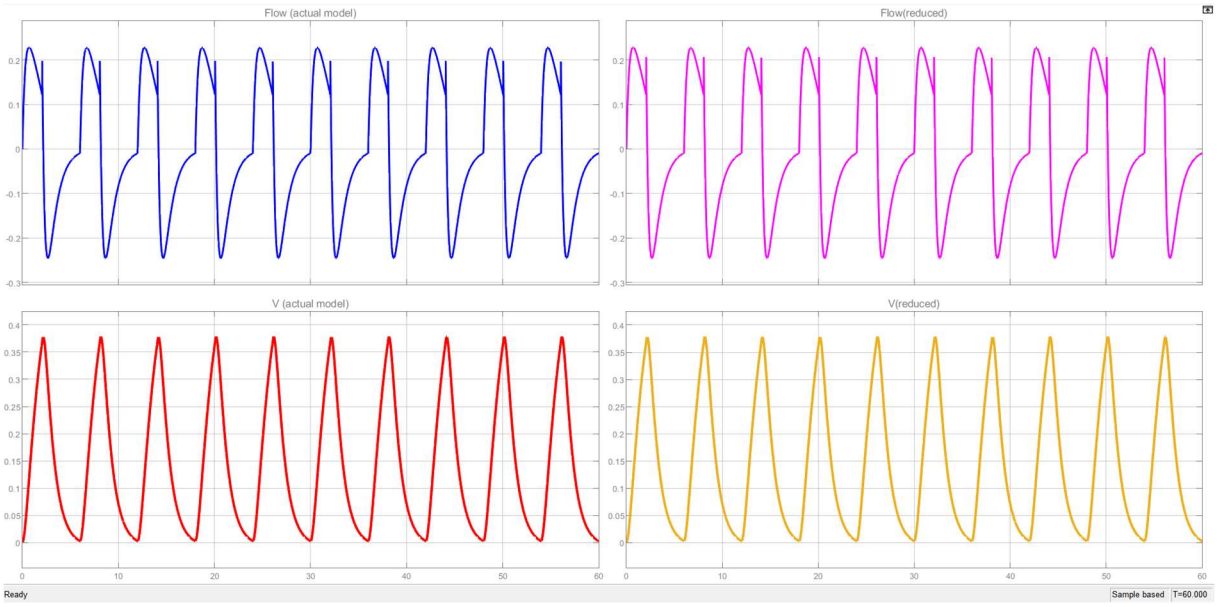


Figure 14 - Output graphs of the models

These two plots are very similar. Thus, we can say that the reduced first order model is consistent with the actual model, meaning this reduced model can be used for further applications.

H-

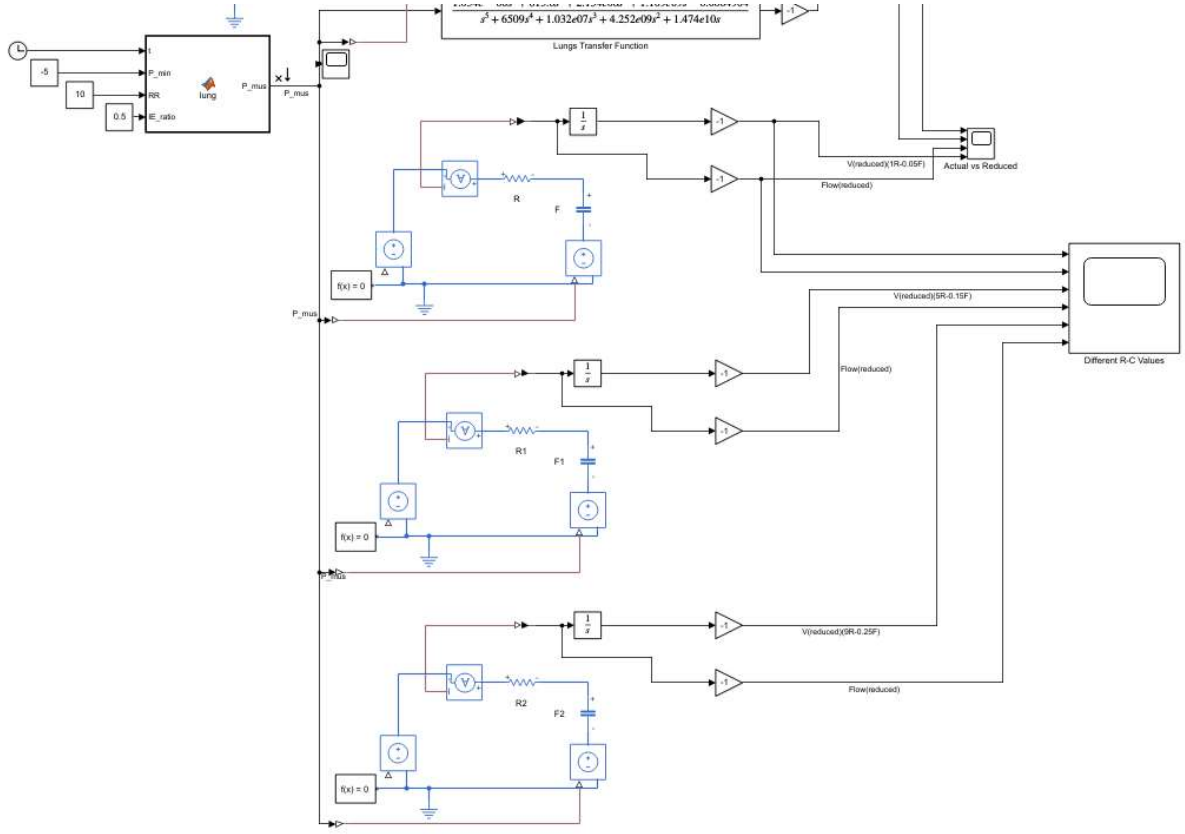
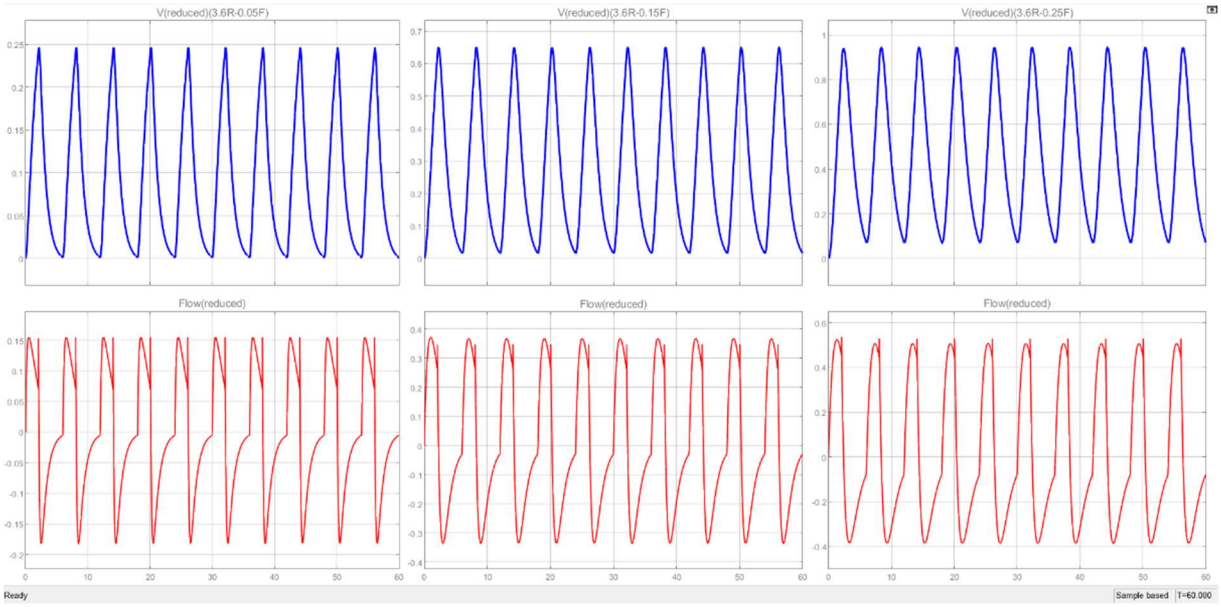
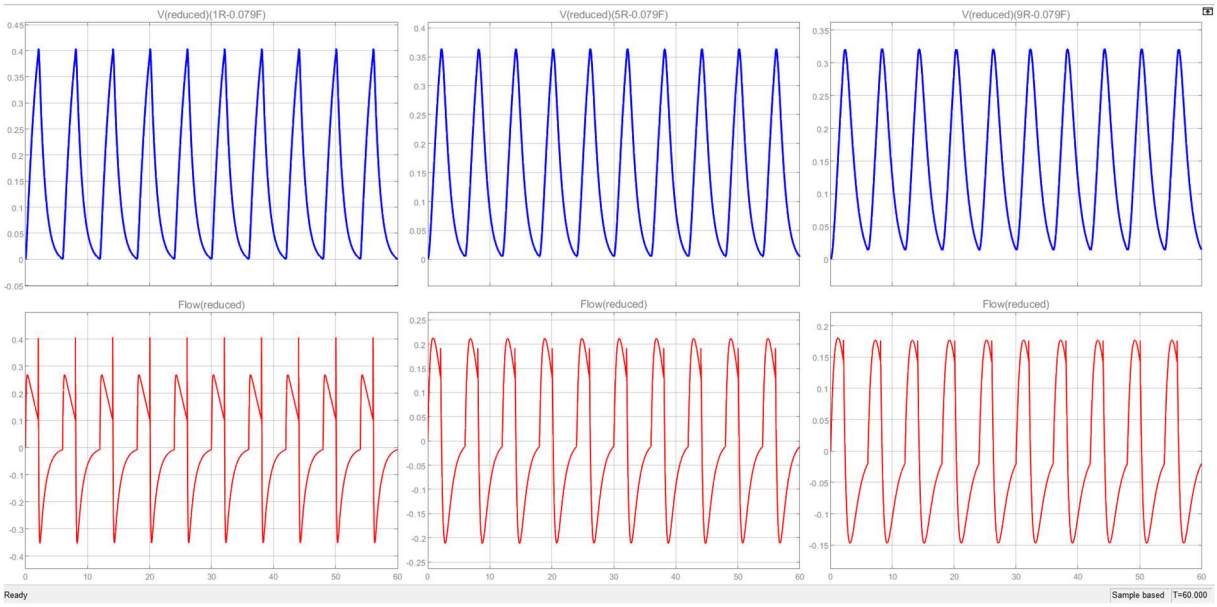


Figure 15 - Simulink Model for Comparing Different R-C Values. Three Seperate Models were used to compare results on the same Scope.

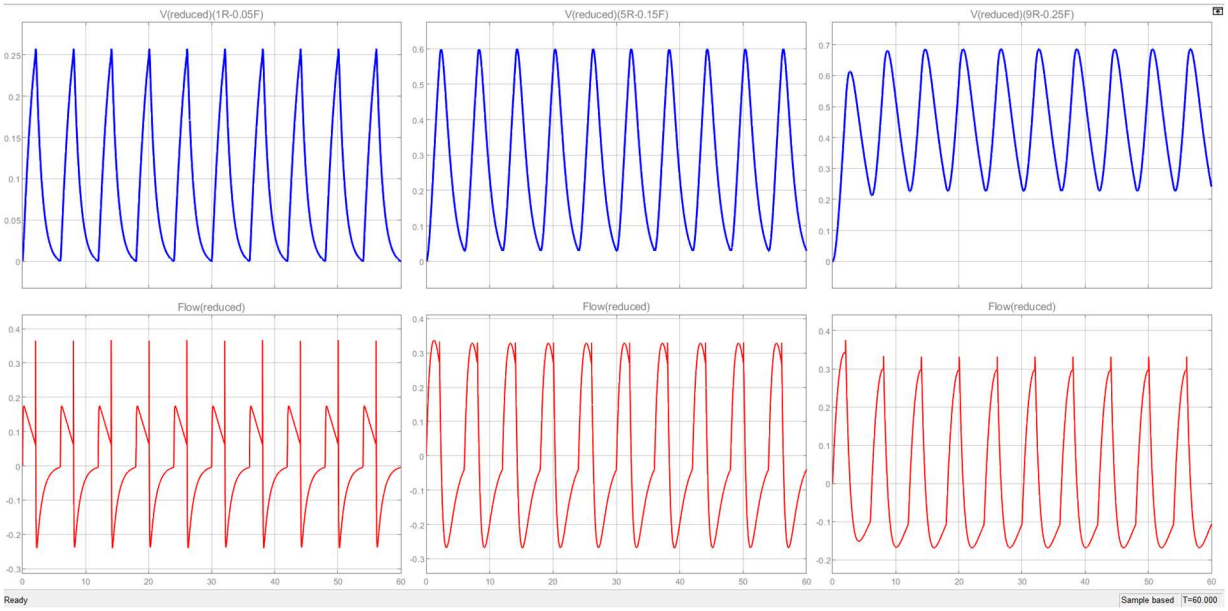
For different C values (0.05-0.15-0.25) when R is constant:



For different R values(1-5-9) when C is constant:



For different R values(1-5-9) and C (0.05-0.15-0.25) values:



In the first two settings, only one value was increased to demonstrate the isolated effect of R and C. In the last setting, both values were increased gradually to show the combined effect. With higher C values, the total capacity we can fill with equal muscle pressure increased. With higher R values, the maximum volume change decreased as expected, caused by the increase in loss to resistance. In the final setting it is shown that, with R and C values great enough, the system charges up and can not return to zero value.

Second Question: Pressure and Volume Controlled Ventilation Modes

A- Pressure Controlled Ventilation

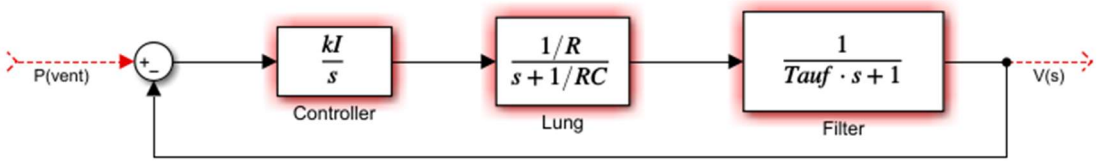


Figure 16 - Block Diagram For Transfer Function

$$G_{openloop}(s) = \frac{V(s)}{P\_vent(s)} = \frac{\frac{1}{R}}{\left(s + \frac{1}{RC}\right)} * \frac{1}{\tau_f s + 1}$$

$$G_{openloop}(s) = \frac{C}{(\tau_f RC)s^2 + (\tau_f + RC)s + 1}$$

Characteristic Equation:

$$1 + F(s)G_{openloop}(s) = 0$$

$$F(s) = \frac{k_I}{s}$$

$$Characteristic\ Polynomial => \ (\tau_f R)s^3 + \left(\frac{RC + \tau_f}{C}\right)s^2 + \left(\frac{1}{C}\right)s + k_I$$

Routh-Hurwitz:

s^3	$\tau_f R$	$\frac{1}{C}$
s^2	$\frac{RC + \tau_f}{C}$	k_I
s^1	b_1	0
s^0	b_2	0

Figure 17 - Routh Hurwitz Table

$$b_1 = \left(\frac{RC + \tau}{C^2} - R\tau_f k_I\right) * \frac{C}{(RC) + \tau_f}$$

$$b_2 = k_I$$

For stability all indices of the first coloumn of Routh-Hurwitz must be positive, Resulting:

$$0 < k_I < \frac{(\tau_f + RC)}{\tau_f RC^2}$$

For R=3.6, C=0.07936

$$\frac{(\tau_f + RC)}{\tau_f RC^2} = \frac{0.385696}{0{,}002267283456} = 170.11$$

$$0 < k_I < 170.11$$

```
clc

clear all % KON305E
close all
format long
syms tao R C k w s
assume([w k],'real')
R=3.6;
C=0.07936;
tao=0.1;
%kl = 1;
s = tf('s');
%closed loop characteristic polynomial
% 1 + F(s)G(s)
Fs = 1/s;
Gs = (1/R)/(s + 1/(R*C));
Hs = 1/(tao*s+1);
%Gs_characteristic = 1 + Fs*Gs;
Gs_rlocus = Fs*Gs*Hs;
%rlocus(Gs_rlocus)
```

Figure 18 - MATLAB Code for Root Locus

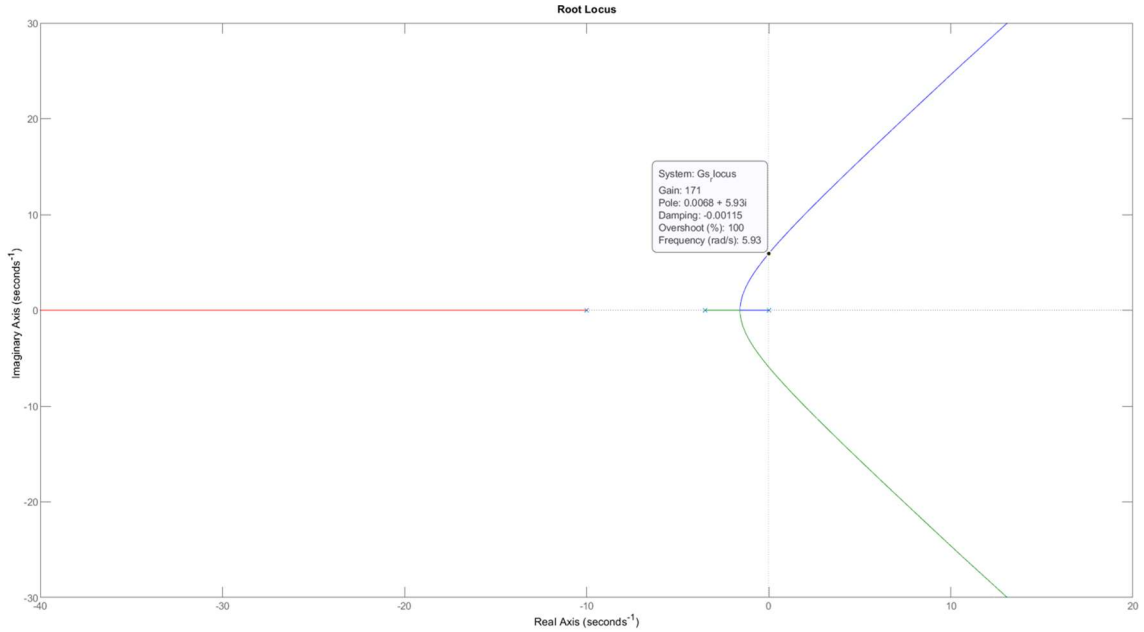


Figure 19 - Root Locus

B-

If F(s) = k, the closed loop transfer function will become:

$$G_{closedloop}(s) = \frac{(G_{openloop}(s).k)}{1 + (G_{openloop}(s).k)} = \frac{Ck}{(\tau_f RC)s^2 + (\tau_f + RC)s + Ck + 1}$$

In order to have minimum settling time and no overshoot, damping ratio ζ must be 1.

General second order transfer function:

$$TF = \frac{Kw_n^2}{s^2 + 2w_n\zeta s + w_n^2}$$

Equating the denominator of closed loop system with general second order function denominator.

$$s^2 + 2w_n\zeta s + w_n^2 = (\tau_f RC)s^2 + (\tau_f + RC)s + Ck + 1$$

$$s^2 + 2w_n\zeta s + w_n^2 = s^2 + \frac{(\tau_f + RC)}{(\tau_f RC)}s + \frac{Ck + 1}{(\tau_f RC)}$$

$$\frac{(\tau_f + RC)}{(\tau_f RC)} = 2w_n\zeta$$

$$w_n^2 = \frac{Ck + 1}{(\tau_f RC)}$$

Taking $\zeta = 1$ from first part,

$$w_n = \frac{(\tau_f + RC)}{\tau_f RC}$$

$$w_n^2 = \frac{Ck + 1}{(\tau_f RC)} = \left(\frac{\tau_f + RC}{\tau_f RC}\right)^2$$

Leaving k alone,

$$k = \frac{(\tau_f - RC)^2}{4\tau_f RC^2}$$

We can divide w_n to two parts to see the effect of C easily:

$$w_n = \frac{(\tau_f + RC)}{\tau_f RC} = \frac{1}{RC} + \frac{1}{2\tau_f}$$

As seen in equation, C has a inverse effect on w_n. Knowing that w_n and settling time is inversely proportional, as C is increased, settling time will increase.

C-

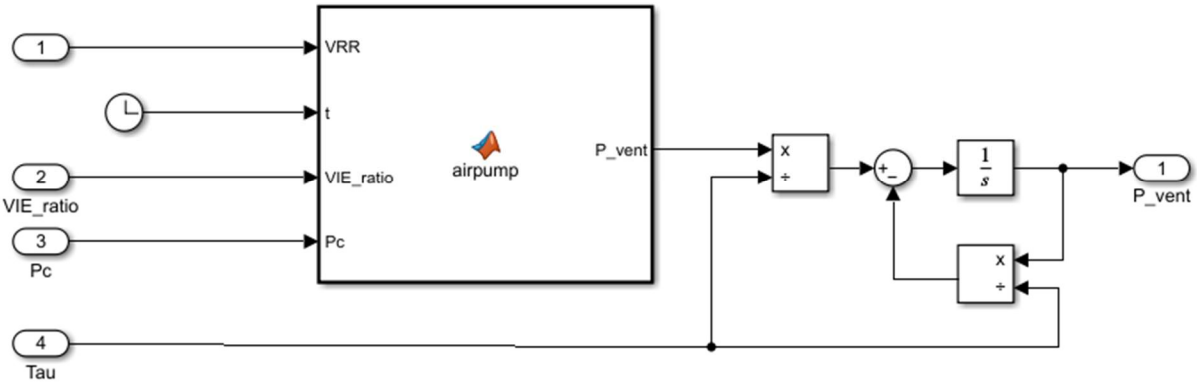


Figure 20 - PCV Mode Pvent Subsystem wi

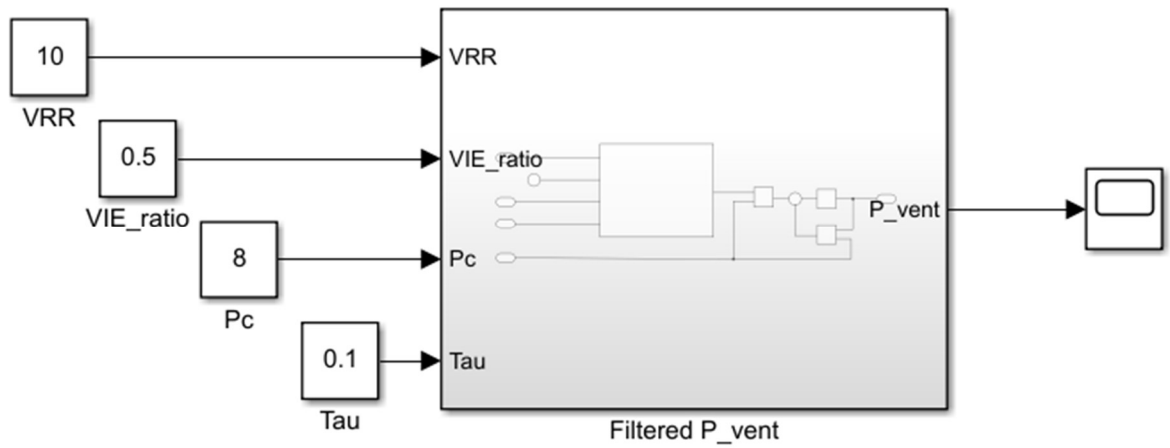


Figure 21 - PCV Mode

D-

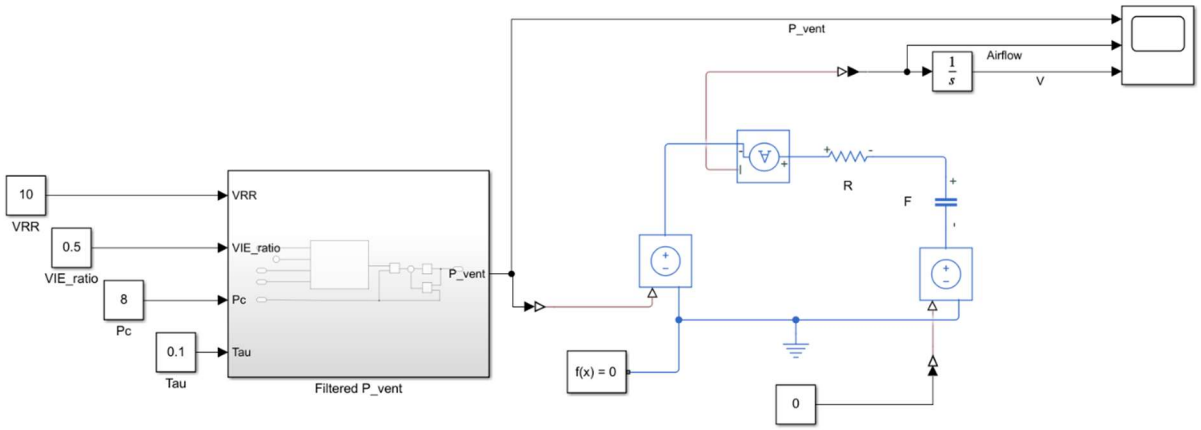


Figure 22 - Simulink Diagram For PCV Mode

$For\ P_c = 8cmH_20$

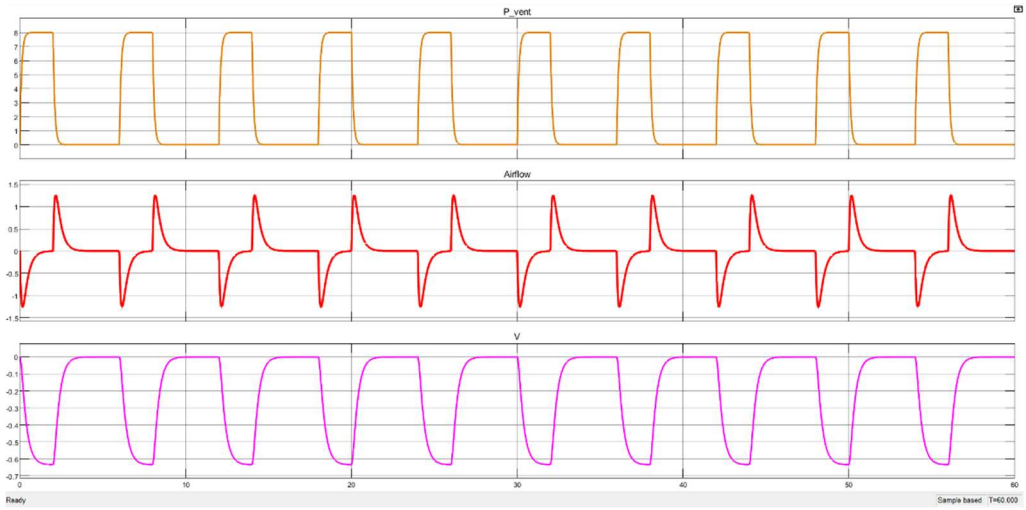


Figure 23 - Output for Pc = 8cmH20

For $P_c = 4cmH\_2O$

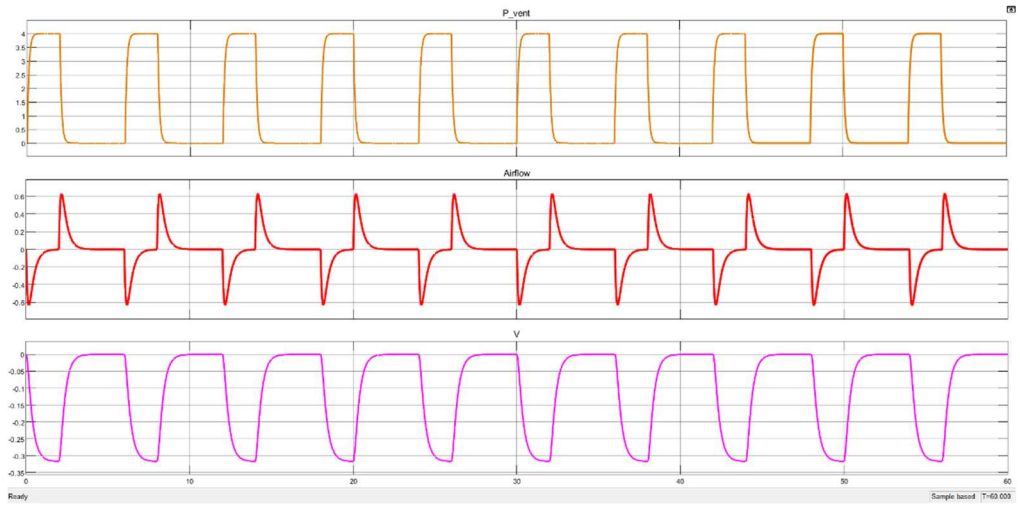


Figure 24 - Output for Pc = 4cmH20

E-

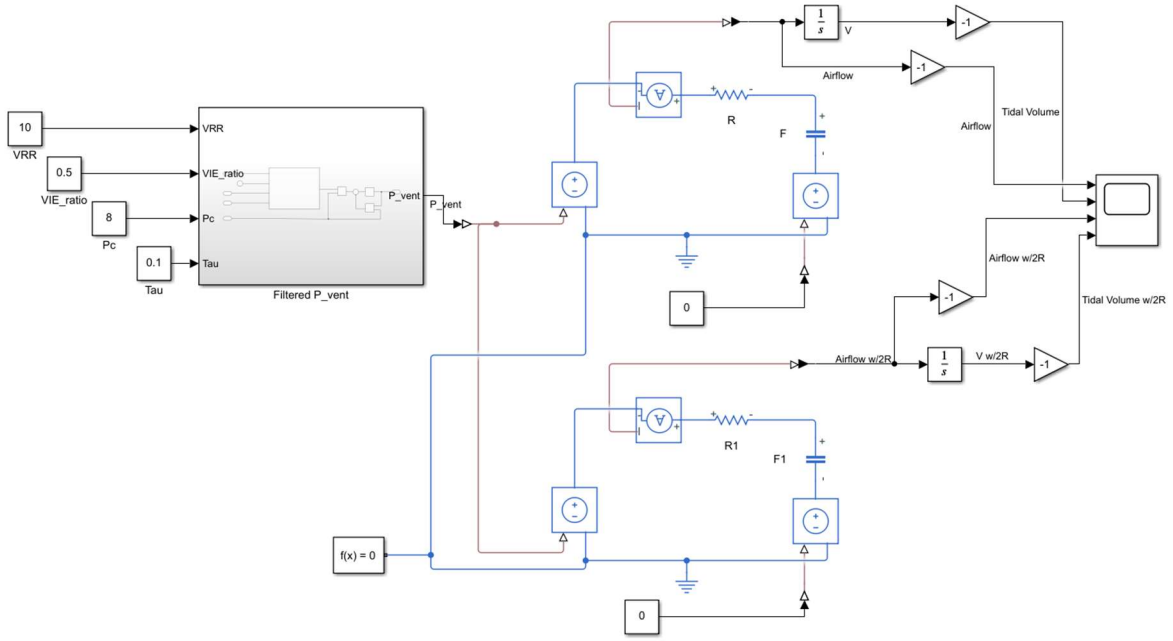


Figure 25 - PCV Mode Comparing Different R Values

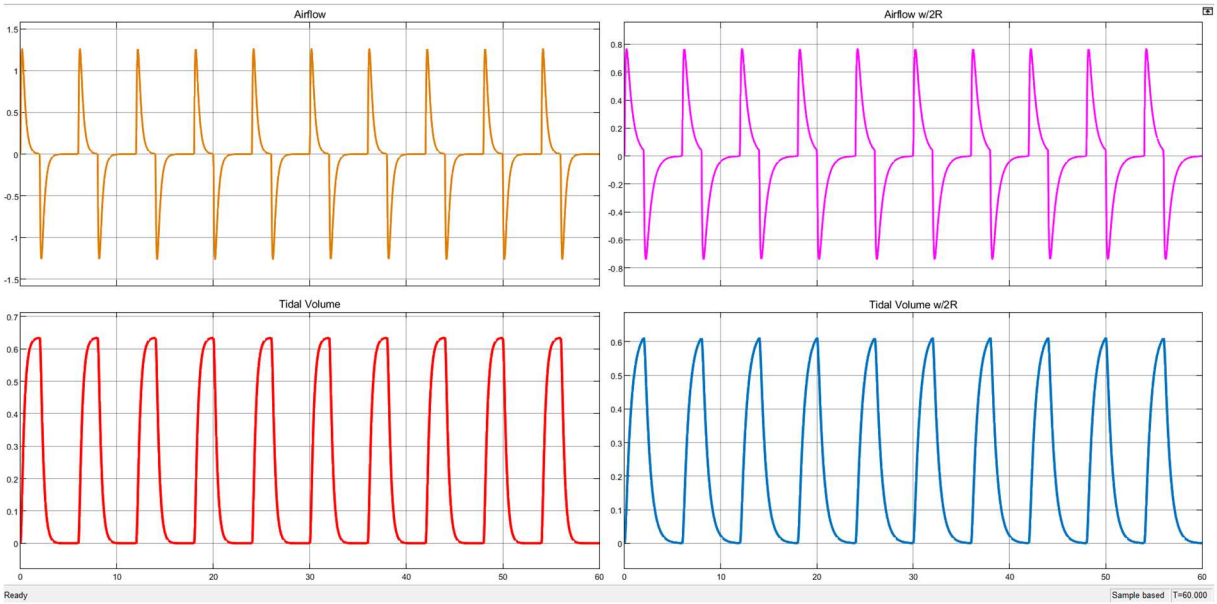


Figure 26 - Output of Different R Values

With the resistance increased to twice its value, the peak airflow has decreased to nearly half of it, which was expected as resistance and current (airflow) is inversely proportional. Tidal volume on the other hand, which is the integral of airflow, decreased only slightly, meaning the sum of flown air has not reduced as much and the capacitance also retains some energy.

F- Volume Controlled Ventilation

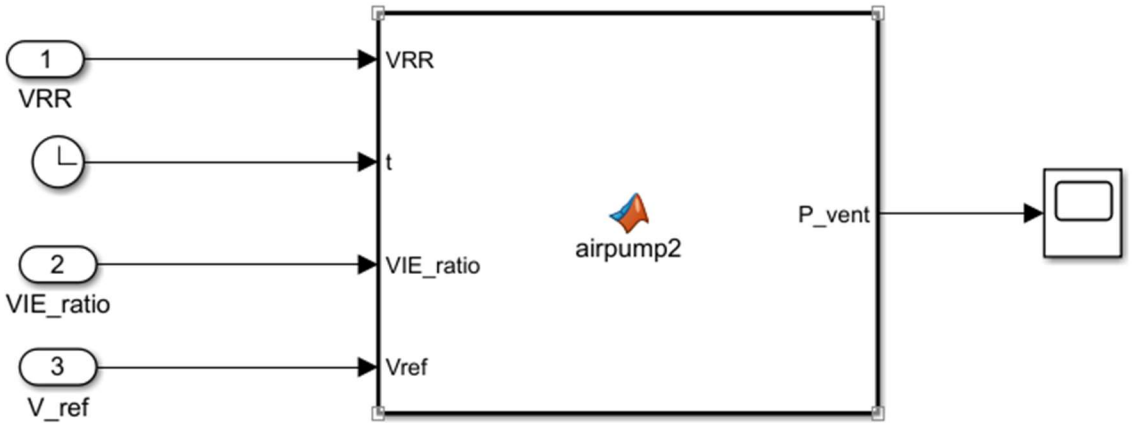


Figure 27 - VCV Mode Subsystem

For Vref = 8, VRR = 10:

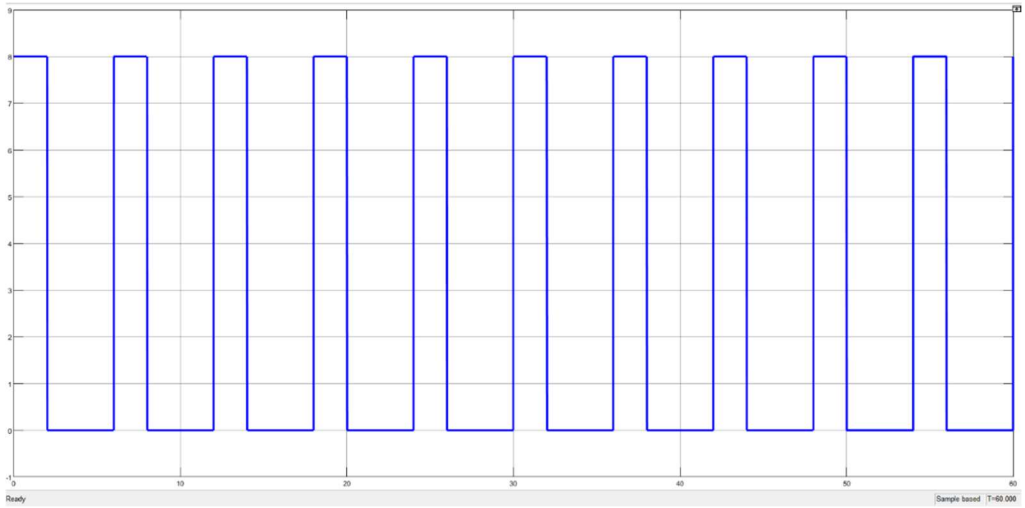


Figure 28 - For Vref = 8, VRR = 10

For Vref = 4, VRR 15:

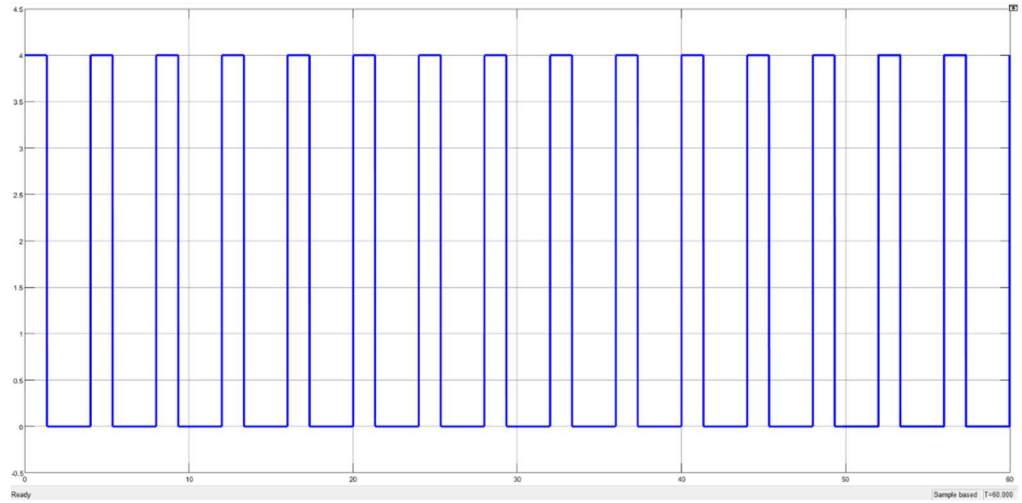


Figure 29 - For Vref = 4, VRR 15

Adding filter for practical applications:

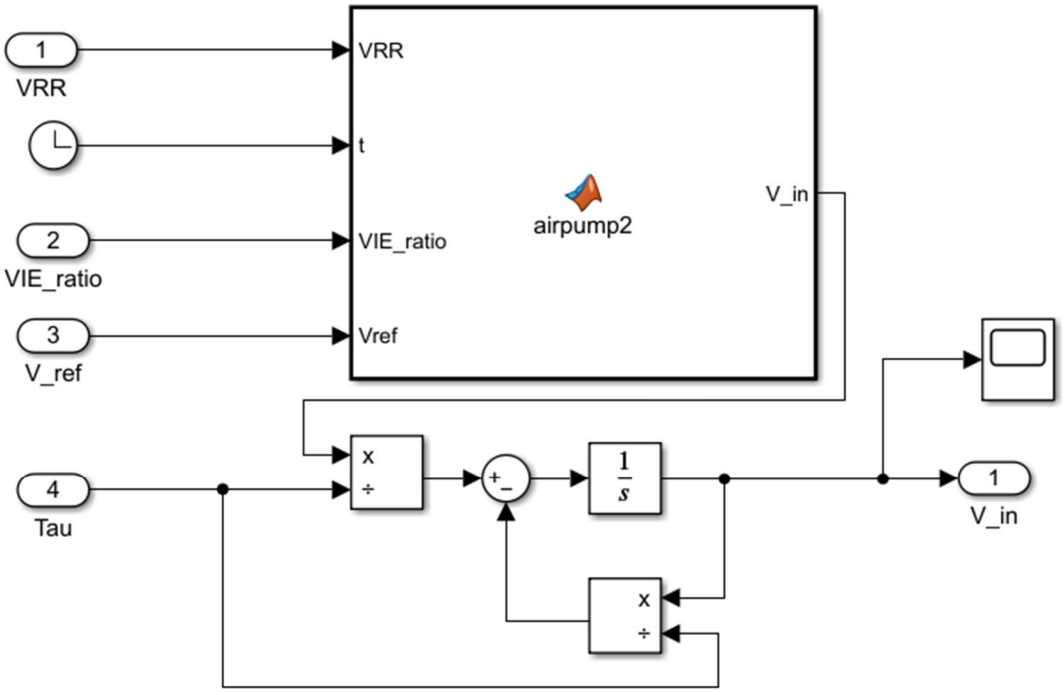


Figure 30 - Adding Filter to VCV Subsystem

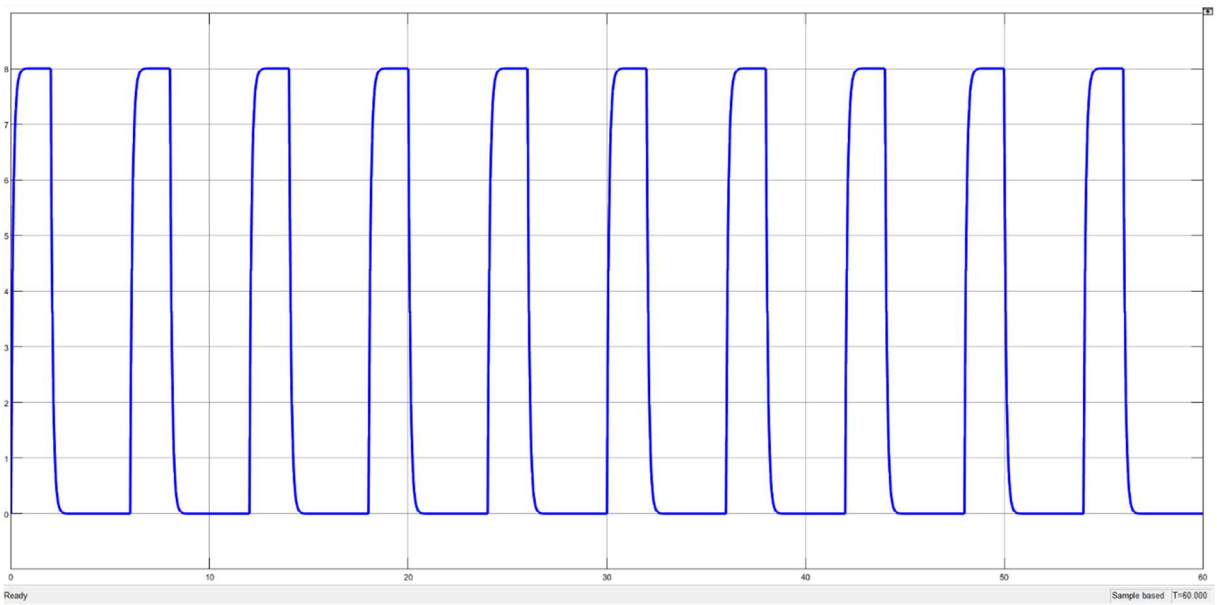


Figure 31 - VCV Mode Reference Input with Filter

G-

Matlab Funcition to find PID parameters:

```
function [Kpi, Kpd, Ki, Kd] = PID_Finder(R, C, tau)
Kpi= (16*R)/(25*tau);
Kpd= ((32*R)/(5*tau))-(1/C);
Ki= (64*R)/(25*tau^2);
Kd= (23*R/5)-(tau/C);
end
```

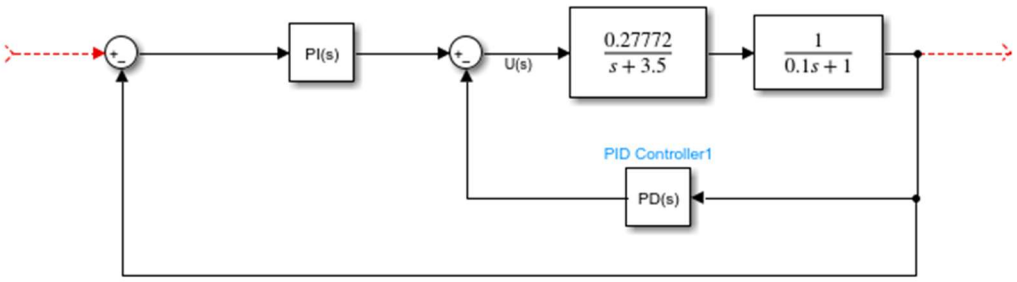


Figure 32 - Block Diagram with PI PD Blocks

H-

Obtaining the transfer function of the closed loop system:

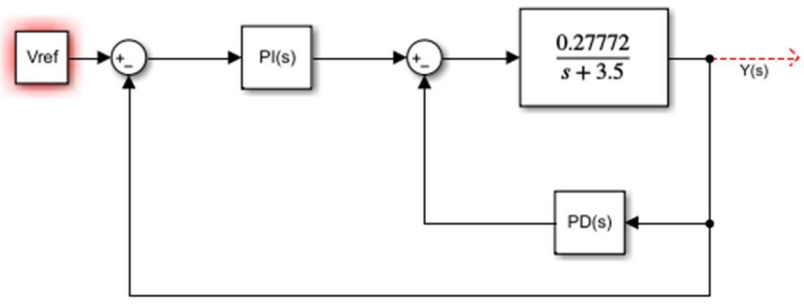


Figure 33 - Overall System,

Zooming into the inner structure, we can see there is a feedback system.

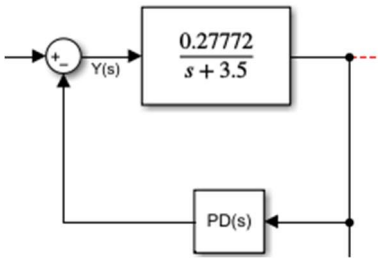


Figure 34 - Inner Feedback Loop

The system will be taken as G(s).

$$inner\ closed\ loop: \frac{G(s)}{1 + F_{PD}(s)G(s)}$$

Then the outer system becomes:

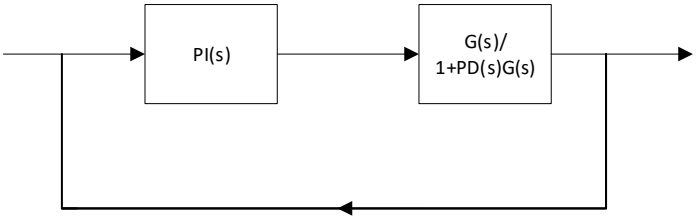


Figure 35 - Unit Feedback Form

$$\frac{PI(s) * \frac{G(s)}{1 + PD(s)G(s)}}{1 + PI(s) * \frac{G(s)}{1 + PD(s)G(s)}}$$

$$G(s) = \frac{C}{(\tau_f RC)s^2 + (\tau_f + RC)s + 1}$$

$$K_{PI} = 23.04$$

$$K_{PD} = 2.178$$

$$K_i = 9.216$$

$$K_D = 15.3$$

$$PI(s) = K_{PI} + \frac{K_i}{s} = 23.04 + \frac{9.216}{s}$$

$$PD(s) = K_{PD} + K_D s = 2.178 + 15.3s$$

$$= \frac{K_{PI} + \frac{K_i}{s} * \frac{\frac{C}{(\tau_f RC)s^2 + (\tau_f + RC)s + 1}}{1 + \frac{(K_{PD} + K_D s)C}{(\tau_f RC)s^2 + (\tau_f + RC)s + 1}}}{1 + K_{PI} + \frac{K_i}{s} * \frac{\frac{C}{(\tau_f RC)s^2 + (\tau_f + RC)s + 1}}{1 + \frac{(K_{PD} + K_D s)C}{(\tau_f RC)s^2 + (\tau_f + RC)s + 1}}}$$

$$= \frac{\left(K_{PI} + \frac{K_i}{s}\right) * \frac{C}{(\tau_f RC)s^2 + (\tau_f + RC)s + 1 + (K_{PD} + K_D s)C}}{1 + \left(\frac{\left(K_{PI} + \frac{K_i}{s}\right)C}{(\tau_f RC)s^2 + (\tau_f + RC)s + 1 + (K_{PD} + K_D s)C}\right)}$$

$$= \frac{\left(K_{PI} + \frac{K_i}{s}\right) * C}{(\tau_f RC)s^2 + (\tau_f + RC)s + 1 + (K_{PD} + K_D s)C + \left(K_{PI} + \frac{K_i}{s}\right) * C}$$

$$T(s) = \frac{K_{PI}Cs + K_i C}{(\tau_f RC)s^3 + (\tau_f + RC + K_D C)s^2 + (1 + K_{PD}C + K_{PI}C)s + (K_i C)}$$

Substituting the variables R,C, τ and K values.

$$T(s) = \frac{1.828s + 73.14}{0.02857s^3 + 1.6s^2 + 20.11s + 73.14}$$

$$T(s) = \frac{64s + 896}{s^3 + 56s^2 + 703.8s + 896}$$

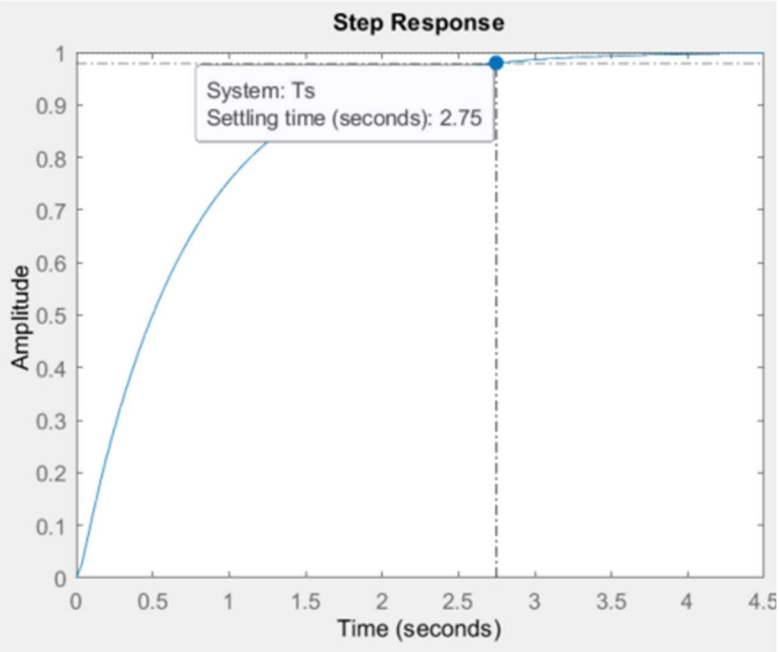


Figure 36 - Step Response of T(s)

Poles of T(s):

```
Continuous-time transfer function.

>> pole(Ts)

ans =

 -38.181686862986638
 -16.386209690308775
  -1.432103446704568
```

-38.181686862986638

-16.386209690308775

-1.432103446704568

Zeros of T(s):

```
>> zero(Ts)

ans =

   -14
```

-14

Due to the fact that the pole at -16.38 is close to the zero at -14, they are simplified. After this step only two poles are left and the corresponding residue is calculated.

$$T_{simplified}(s) = \frac{14}{16.3862} * \frac{64}{(s + 39.181) * (s + 1.43221)}$$

$$= \frac{54.68}{s^2 + 40.61s + 56.12}$$

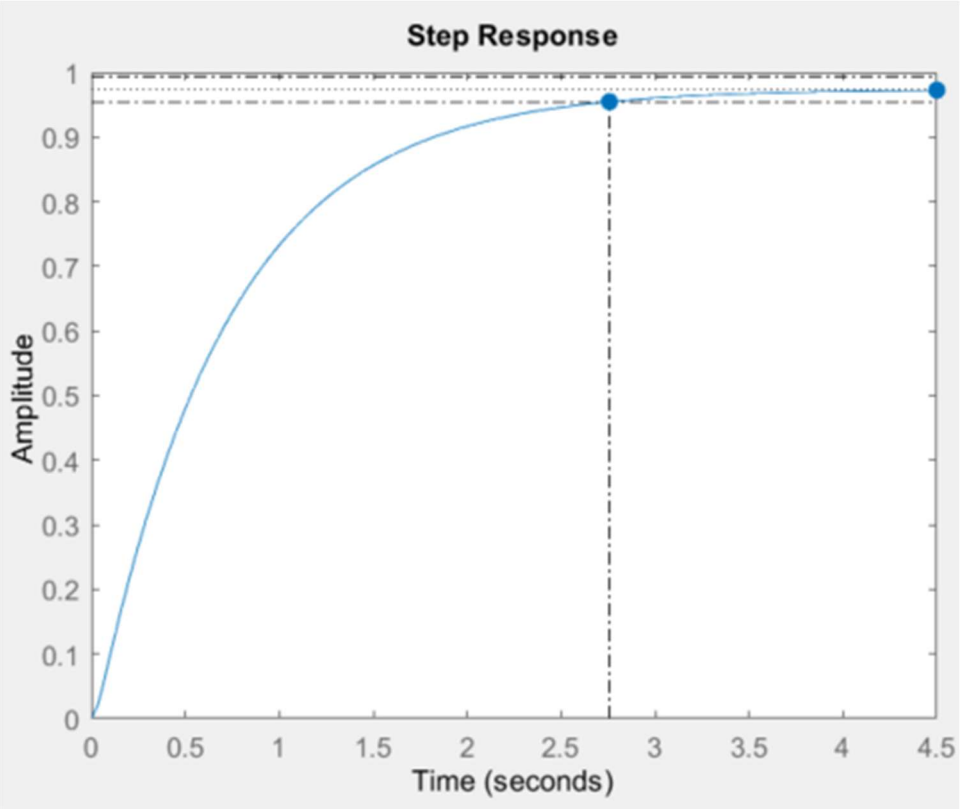


Figure 37 - Step Response Of Tsimplified(s)

I-

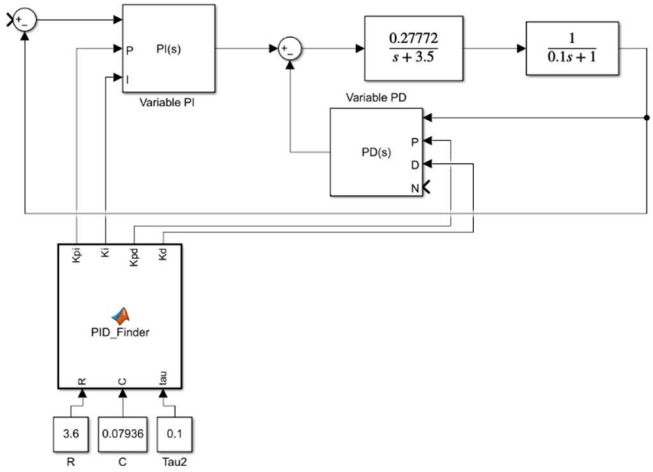


Figure 38 - Varying PID Controller with PID_Finder

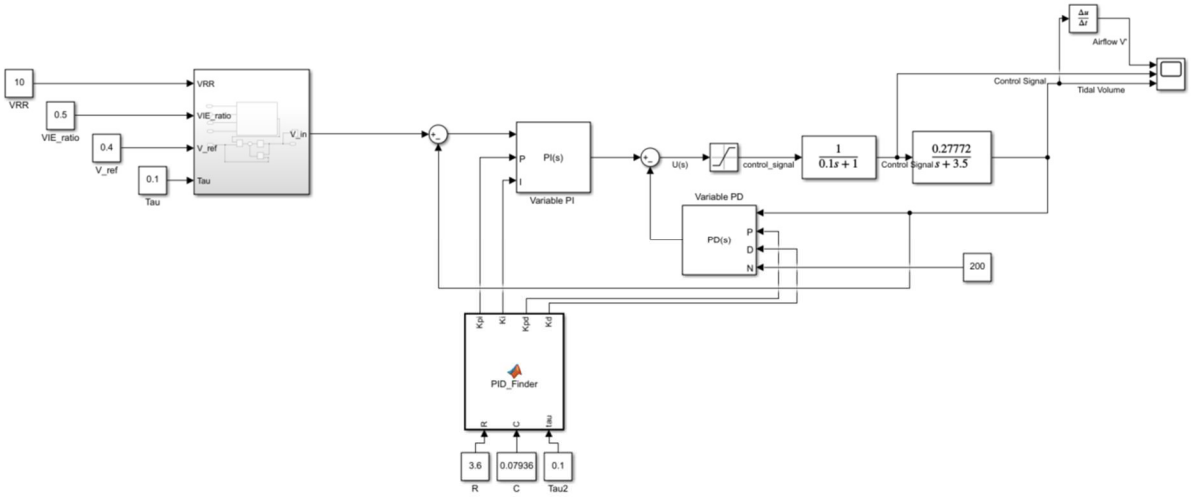


Figure 39 - Final VCV System

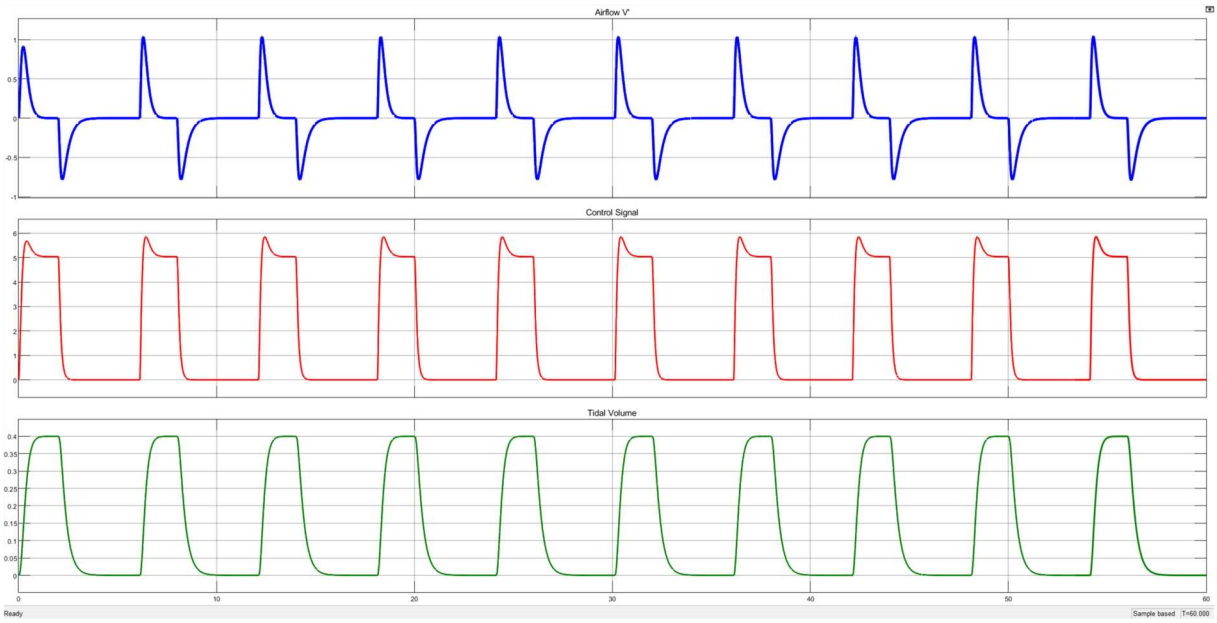


Figure 40 - VCV System Output

Observing the output of VCV System, we can clearly say that the desired tidal volume graph is obtained, which is the output of the whole system. It is nearly identical to the reference signal V_{ref}, so the system is successfully controlled. The PI-PD blocks generate the required control signal to match the output with reference.

J-

In this part, we test our system on a patient with Pulmonary Emphysema, which is a type of chronic obstructive pulmonary disease (COPD). The lung compliance is increased by %150. We run the system without changing anything other than the C value, and observe the output tidal volume.

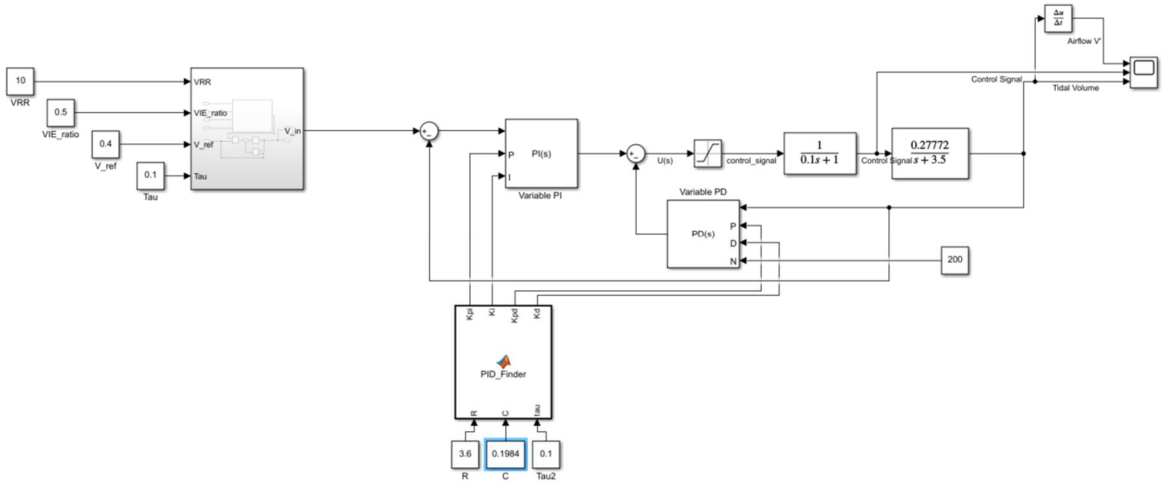


Figure 41 - VCV System with C Value Increased (Pulmonary Emphysema)

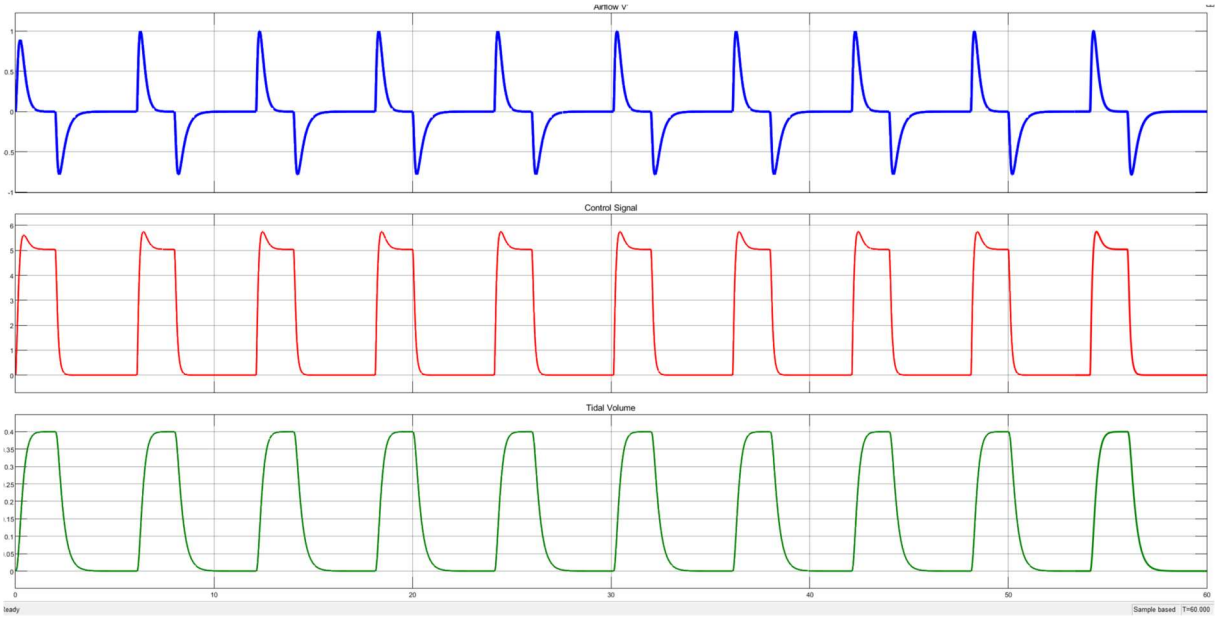


Figure 42 – VCV System (Pulmonary Emphysema) Output

As seen in Figure 42 , when the lung compliance is increased by %150 (C'= 2.5C) the control system is changing Kp, Ki, Kd parameters accordingly, to generate the desired tidal volume graph, same as the initial healthy lung. This shows that the control system can be used for Pulmonary Emphysema disease.

Third Question: PCV and VCV Modes with Simultaneous Breathing

A-

```
function P_vent = airpump(VRR,t,VIE_ratio,Pc,
Vprime, Thresh)
    %%VRR, Ventilator Respiratory Rate
    %%VIE_ratio, inhale/exhale rate
    %%Pc, control pressure
    %%t, filter time constant
    %%Vprime, measured airflow
    %%Thresh, flow threshold
    T = 60/VRR;
    Te = T/(1+VIE_ratio);
    Ti = Te*VIE_ratio;

    if(rem(t,T)<Ti || Vprime > Thresh )
            P_vent = Pc;
            t = 0;
    else
        P_vent = 0;
    end
end
```

Figure 43 - MATLAB Code for Spontaneous Breathing PCV

Here, this simple code initiates the Ventilator Pressure by checking whether the mandatory breathing time has come or the patient tries to take a breath with an OR command. As the assisted breathing is applied, the timer for mandatory breathing *t* is also set to 0, which was required in the project.

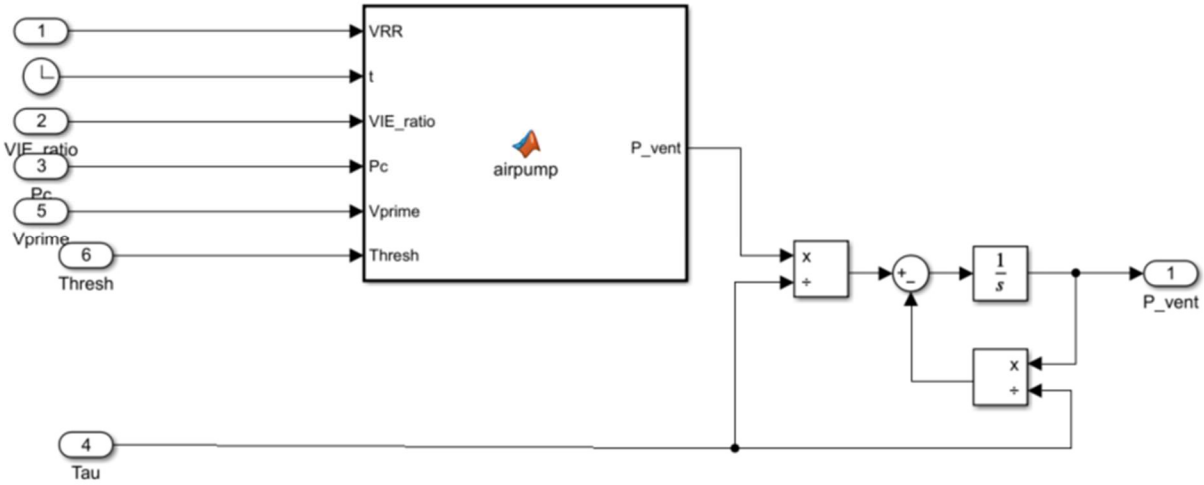


Figure 44 - PCV System with Measured Airflow and Flow Threshold

The filter, $\frac{1}{\tau s+1}$, is created with appropriate blocks in the subsystem.

B-

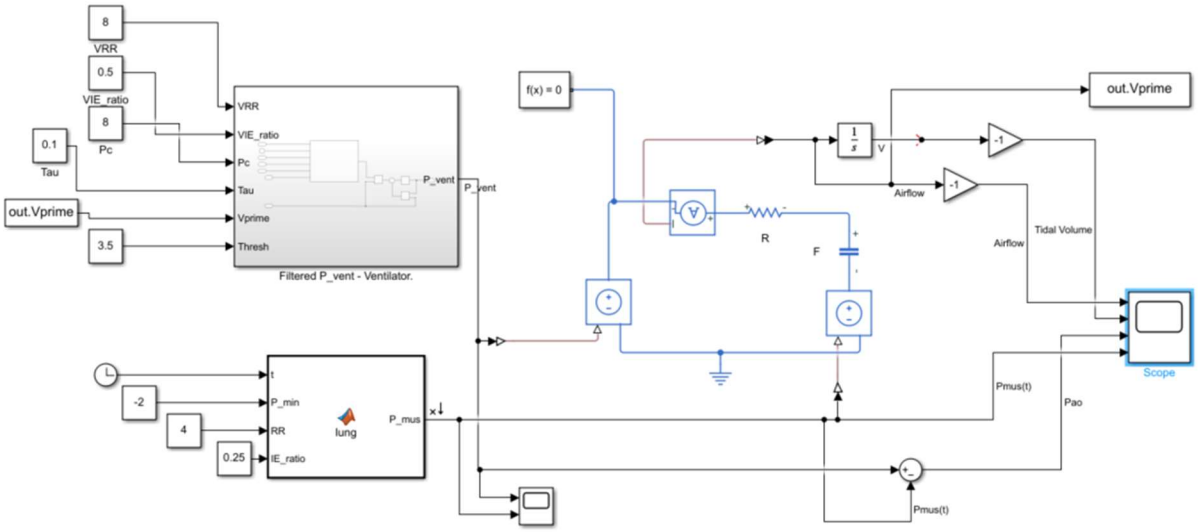


Figure 45 -PCV System with Spontaneous Breathing

The diaphragm muscle and the Ventilator works simultaneously, which is shown in this scope.

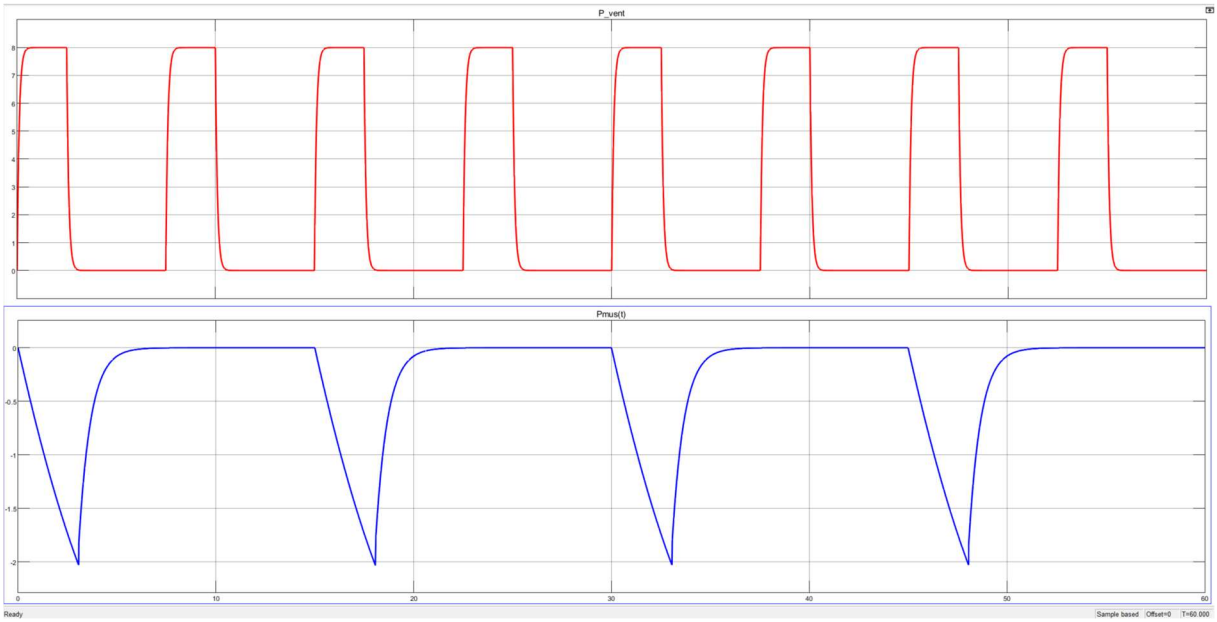


Figure 46 - Pmus and Pvent Graphs

C-

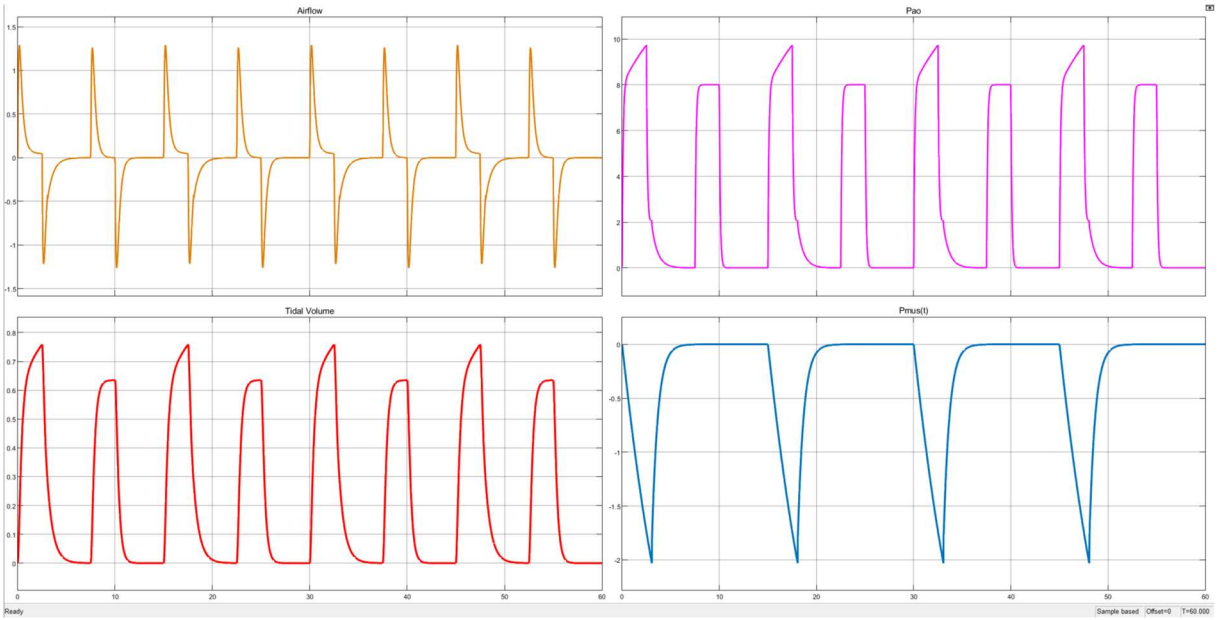


Figure 47 - Output of PCV with Spontaneous Breathing

Here we can see that, as the patient tries to take a breath, the the ventilator machine initiates the assisting breath. This was shown clearly in code, however, the patient activity and ventilator activity given in the project description has the same fundamental frequencies, thus the instances the patient tries to take a breath are at the same instance that the ventilator is already giving assisted breaths. Other activity scenarios for the patient were tried, (for example with patient RR= 5 breaths/min) with barely satisfying results. The important part here is that when the patient tries to take a breath, this is detected through the measured airflow, with a well tuned treshhold set, that is lower than the airflow created by the effort of the patient. The second important aspect here is that the mandatory breathing timer is set to 0, so that regardless of the patient, the ventilator works regularly.

D-

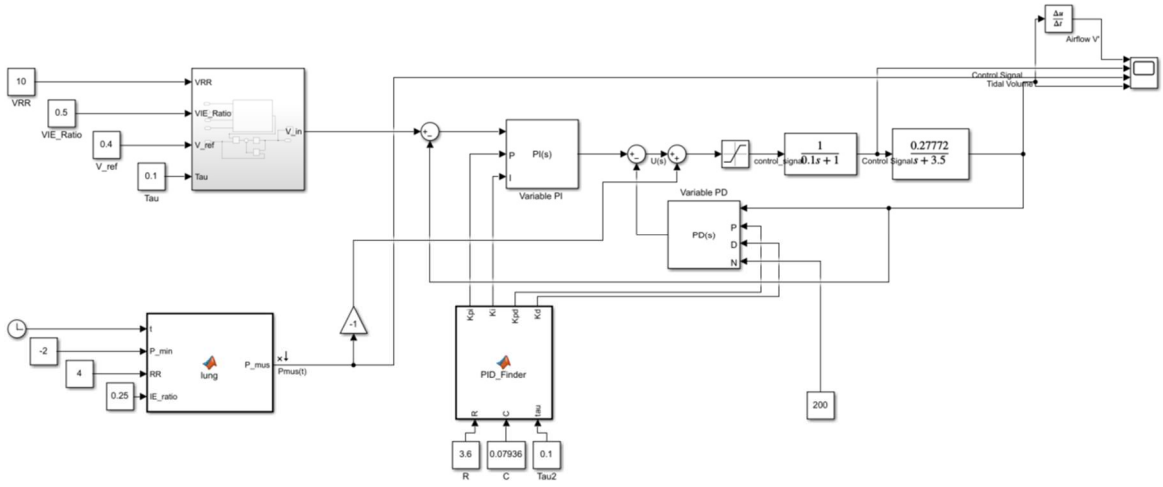


Figure 48 - VCV System with Spontaneous Breathing

Here, the VCV system is updated with the requested P_{mus} model. N input of Varying-PD is set to 200, by trial and error. The effort made by the patient is added to the input of the lung model at U(s). It is also multiplied by -1 so that the diaphragm and the ventilator works in the same direction.

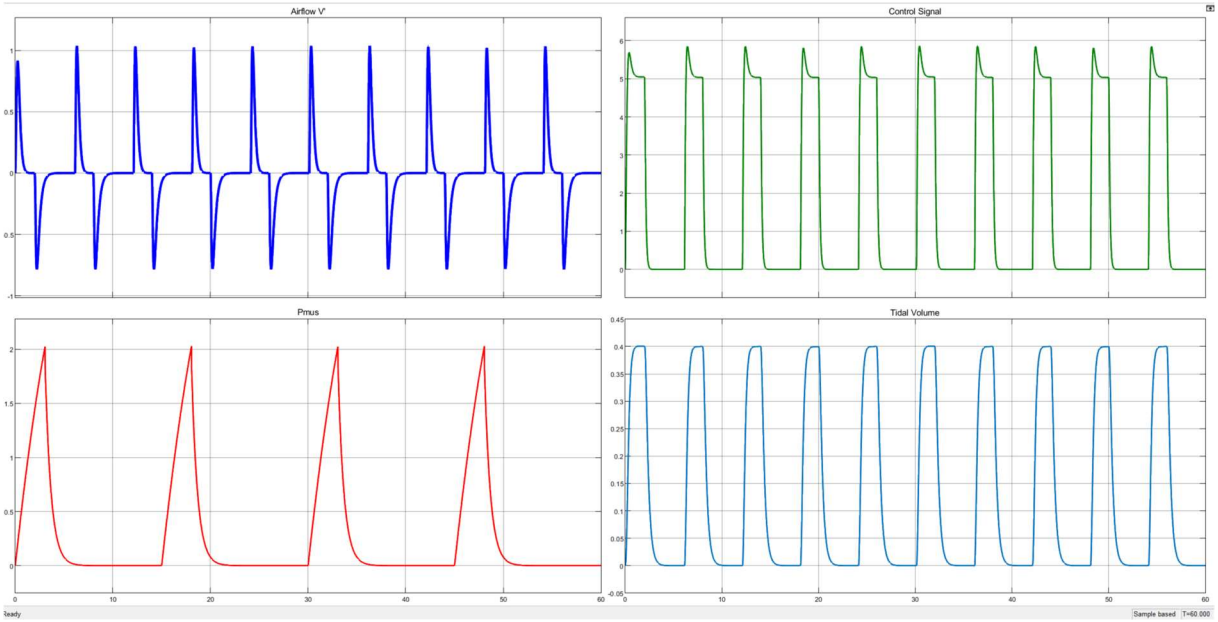


Figure 49 - Output of VCV with Spontaneous Breathing

The tidal volume graph shows that the desired volume reference signal V_{ref} is the same as the output tidal volume, which means that the ventilator works as required even if the patient tries to take a breath.

E-

We were requested to design a Mechanical Ventilator, which is used to aid patients in extensive care units, who are struggling to breath on their own due to some medical complications.

The project started with the modelling of the lungs, in order to try the designed ventilator models on. For this, we first created a Simulink Model with the electrical circuit equivalent given in the project description. Then we acquired a transfer function which is the direct conversion of the electrical circuit and compared this transfer function to our actual model. Then we reduced this fifth order transfer function to a first order one, with the knowledge we gained in Controls Engineering Courses and with the help of MATLAB. Finally we created the equivalent electrical circuit of this reduced transfer function, and used this simple model onwards.

In the second part, we were requested to design to different modes for the Mechanical Ventilator, namely Pressure Controlled and Volume Controlled Ventilation. Both were created by following the steps carefully. The results were promising, the output graphs of both the models were as desired, generating a healthy tidal volume plot.

For the third and last part, we finally added the patient diaphragm activity to both PCV and VCV models, as these ventilators will sometimes be used on half-awake patients, who might start to breath on their own, often inadequately. We again succesfully generated the desired tidal volume graphs, under different lung resistances and capacitances.

Finalizing our project report, we would like to thank our course instructors for giving us an in-depth education on computer aided techniques on control engineering, and a detailed term project to test and demonstrate the acquired skills.