Question 1

a.

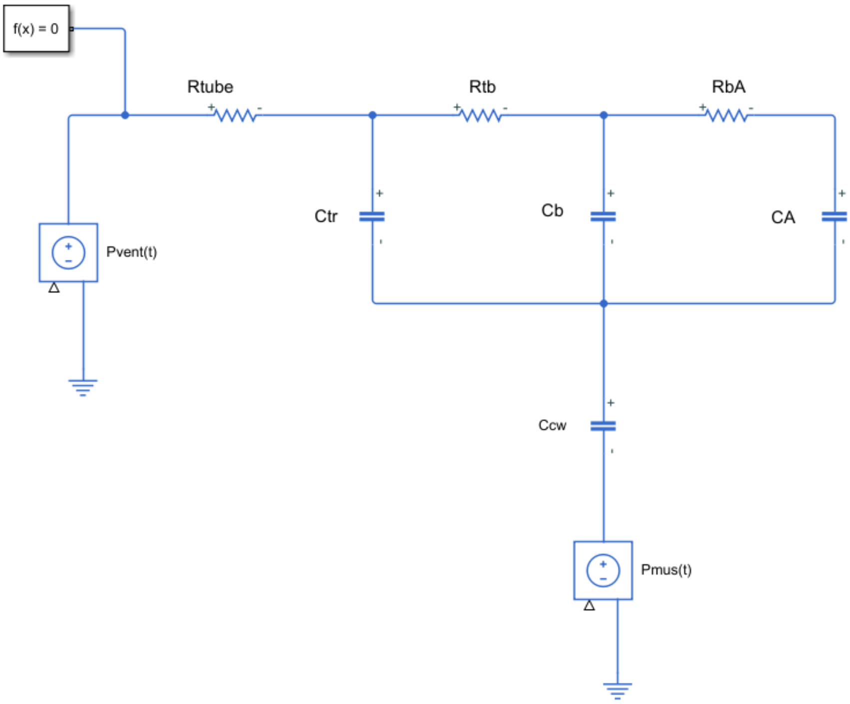


Figure 1: The lung mechanics model with the help of Simscape library (No inputs)

b.

$$P_{mus}(t) = \begin{cases} \dfrac{P_{min}}{T_I\, T_E}(Tt - t^2) & t \in [0,\, T_I] \quad (1.1) \\ \dfrac{P_{min}}{1 - e^{-\frac{T_E}{\tau}}}\left(e^{-\frac{(t-T_I)}{\tau}} - e^{-\frac{T_E}{\tau}}\right) & t \in [T_I, T] \quad (1.2) \end{cases}$$

Where,

$\boldsymbol{T_I}$: Inspiratory Time (s)

$\boldsymbol{T_E}$: Expiratory Time (s)

$\boldsymbol{RR}$: Respiratory Rate (breaths/min)

$\boldsymbol{IE_{ratio}}$: Inspiration/Expiration Ratio

$$T_I + T_E = T = 60/RR \tag{1.3}$$

$$T_I = IE_{ratio} T_E \tag{1.4}$$

$$\tau = T_E/10 \tag{1.5}$$

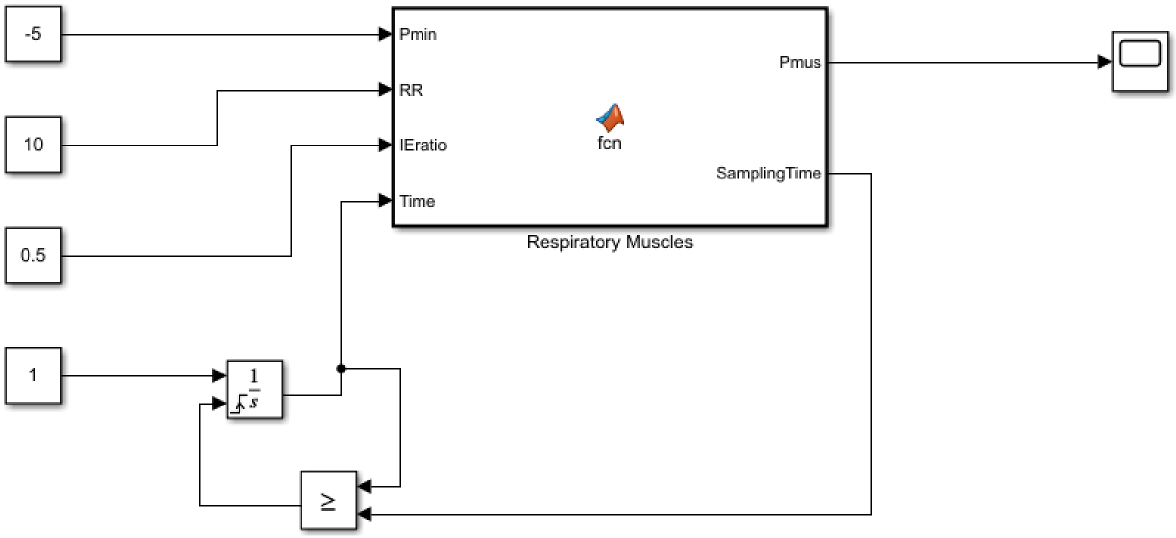


Figure 2: Respiratory Muscles

In order to adjust proper sampling time integrator block is used, and its input set to zero. Also, comparator block is used to achieve different output at the different intervals of the period. Then, output of the comparator block is fed to input of the integrator block as an external reset. Therefore, at interval [0, 6] (Inspiratory time) equation 1.1, and at [6, 10] (Expiratory Time) equation 1.2 is used as output.

Function is created for RR = 10, Pmin = -5 and IE_{ratio} = 1:2 during 60 seconds. Sampling time calculated from equation 1.3 and found 6, then inspiratory time and expiratory time is calculated by using equation 1.3 and 1.4. and T_I is found 4 , T_E is found 2.

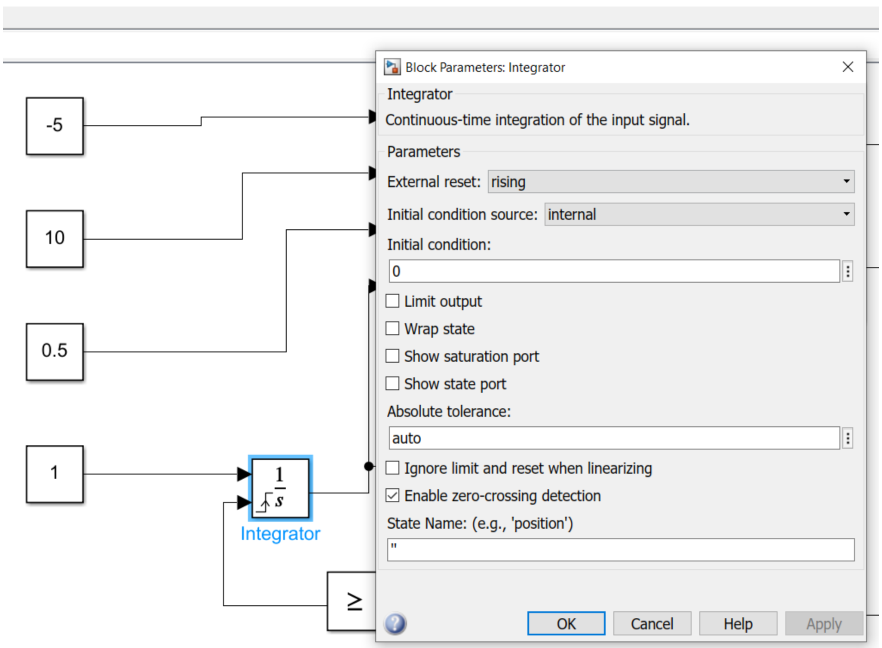


Figure 3: Integrator

```
function [Pmus,SamplingTime] = fcn(Pmin,RR ,IEratio, Time)

% RR standds for Repiratory Rate (breaths/min)
% IEratio stands for Inspiration/Expiration Ratio

T = 60/RR; % Sampling Time (s)
SamplingTime = T;
TE = T/(1 + IEratio); % Expiratory Time (s)
TI = T - TE; % Inspiratory Time (s)
Tau = TE / 10;

if (Time <= TI)

    Pmus = Pmin*(1/(TI*TE))*(T*Time -Time^2)


else

    Pmus = (Pmin/(1 - exp((-TE/Tau))))*(exp(((Time - TI)/(-Tau)))-exp(((-TE)/(Tau))))
end


end
```

Figure 4: Function of the Respiratory Muscle

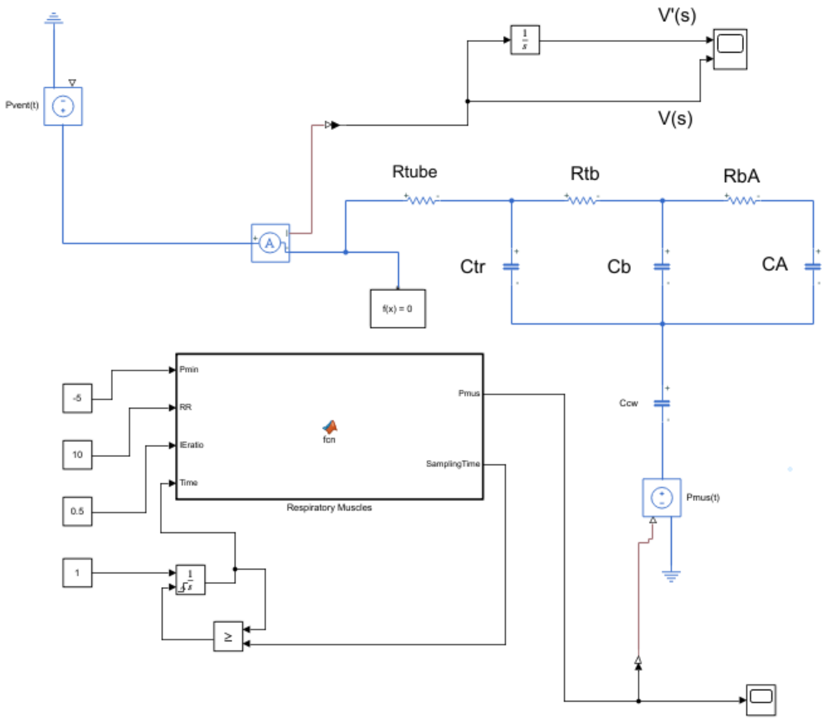


Figure 5: Whole System

c.

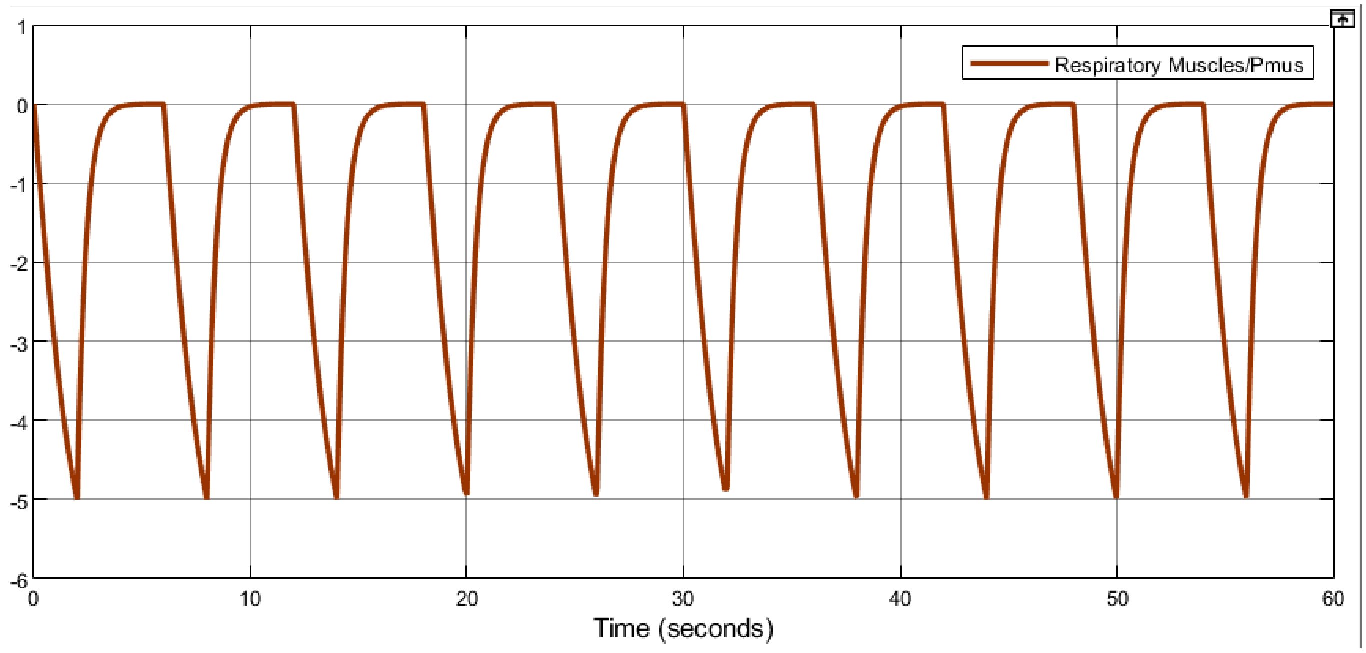


Figure 6 : Pmus(t)

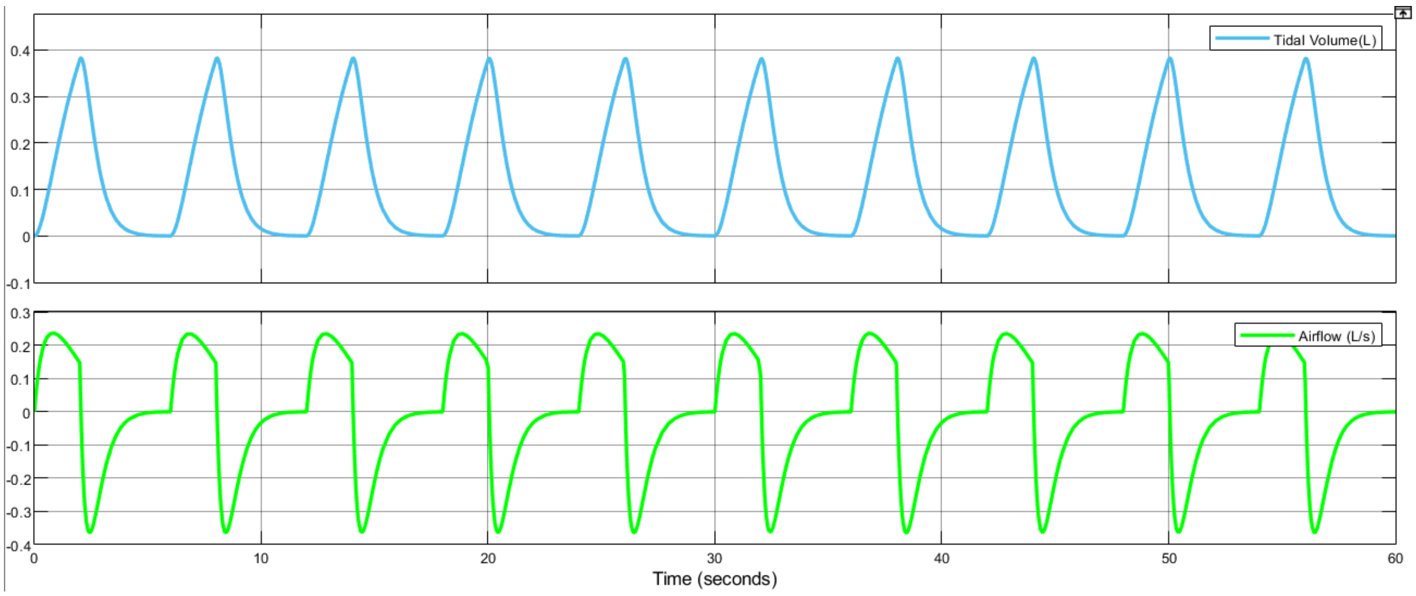


Figure 7: Tidal Volume $V(t)$ and Airflow $V'(t)$ from Simulated System

Airflow is represented as $\boldsymbol{V'(t)}$ and tidal volume is represented as $\boldsymbol{V(t) = \int V'(t)dt}$

d.

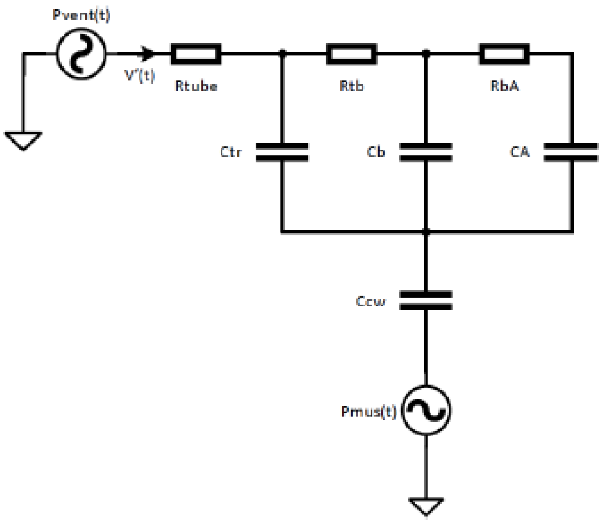
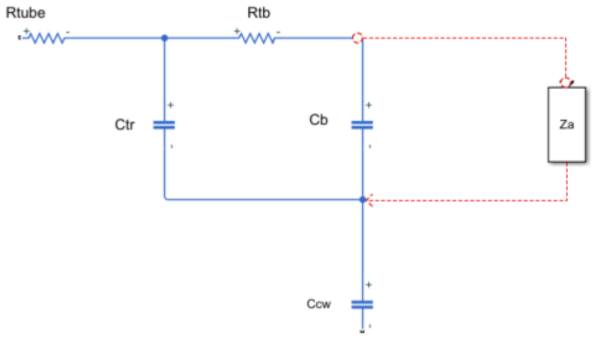
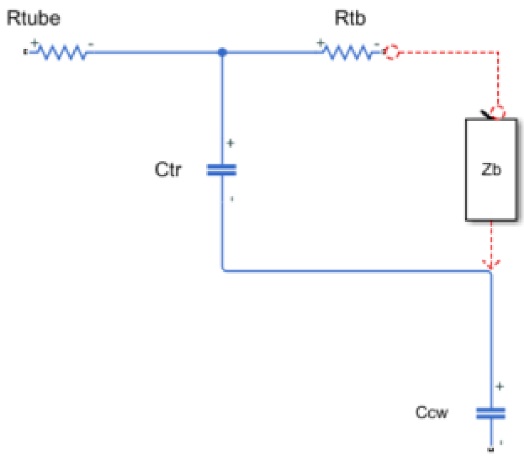


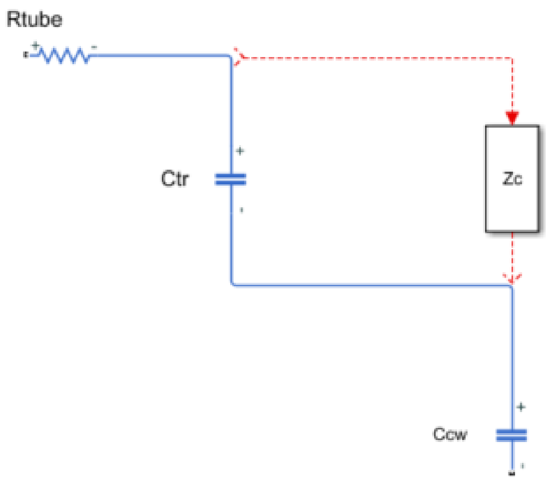
Figure 8: The lung mechanics model given in the project



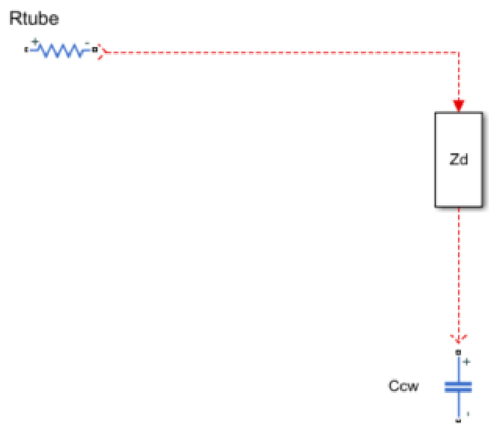
$$Za = Rba + \frac{1}{sCa} \tag{1.6}$$



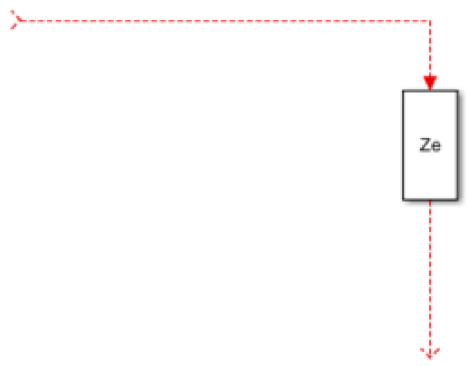
$$Zb = (\frac{Za}{sCb})/(Za + \frac{1}{sCb}) \tag{1.7}$$



$$Zc = Zb + Rtb \tag{1.8}$$



$$Zd = (\frac{Zc}{\mathrm{sCtr}})/(Zc + \frac{1}{\mathrm{sCtr}}) \tag{1.9}$$



$$\mathrm{Ze} = \mathrm{Zd} + \mathrm{Rtube} + \frac{1}{\mathrm{sCcw}} \tag{1.10}$$

Figure 9: Reducing Circuit Representation of the Lung Mechanics

```
Rtube = 4; % Resistance of tube [ cm.H2O.L-1 ]
Rtb = 0.212; % Resistance between trachea and bronchitis [ cm.H2O.L-1 ]
Rba = 0.082; % Resistance between bronchitis and Alveoli [ cm.H2O.L-1 ]
Ctr = 0.0016; % Compliance of trachea [ L/(cm*H2O) ]
Cb = 0.013; % Compliance of bronchitis  [ L/(cm*H2O) ]
Ca = 0.15; % Compliance of Alveoli [ L/(cm*H2O) ]
Ccw = 0.2; % Compliance of chest wall [ L/(cm*H2O) ]

s = tf('s');

% Z stands for empedances

Za = ( Rba + 1/(s*Ca));
Zb = Za*1/(s*Cb) / ( Za + 1/(s*Cb) );
Zc = Zb + Rtb;
Zd = Zc*1/(s*Ctr) / ( Zc + 1/(s*Ctr) );
Ze = Zd + Rtube + 1/(s*Ccw);

Z_sum = Ze;
```

Figure 10: Transfer Function Calculation

$$\mathrm{G(s)} = \frac{V(s)}{Pmus(s)} \tag{1.11}$$

$$Pmuss\ \mathrm{(s)} = Z_{sum}\ x\ V'(s) \tag{1.12}$$

$$G(s) = \frac{V(s)}{Z_{sum}\ x\ V\prime(s)} = \frac{V(s)}{s\ x\ V(s)\ x\ Z_{sum}} \tag{1.13}$$

$$G(s) = \frac{1}{s\ x\ Z_{sum}} \tag{1.14}$$

```
              0.25 s^2 + 1083 s + 7.587e05
G(s) =  --------------------------------------
        s^3 + 4488 s^2 + 3.256e06 s + 8.403e06
```
(1.15)

Open loop transfer function of the system between $P_{mus}(s)$ and V(s) is calculated ($P_{vent}(t)$ is assumed to be 0)

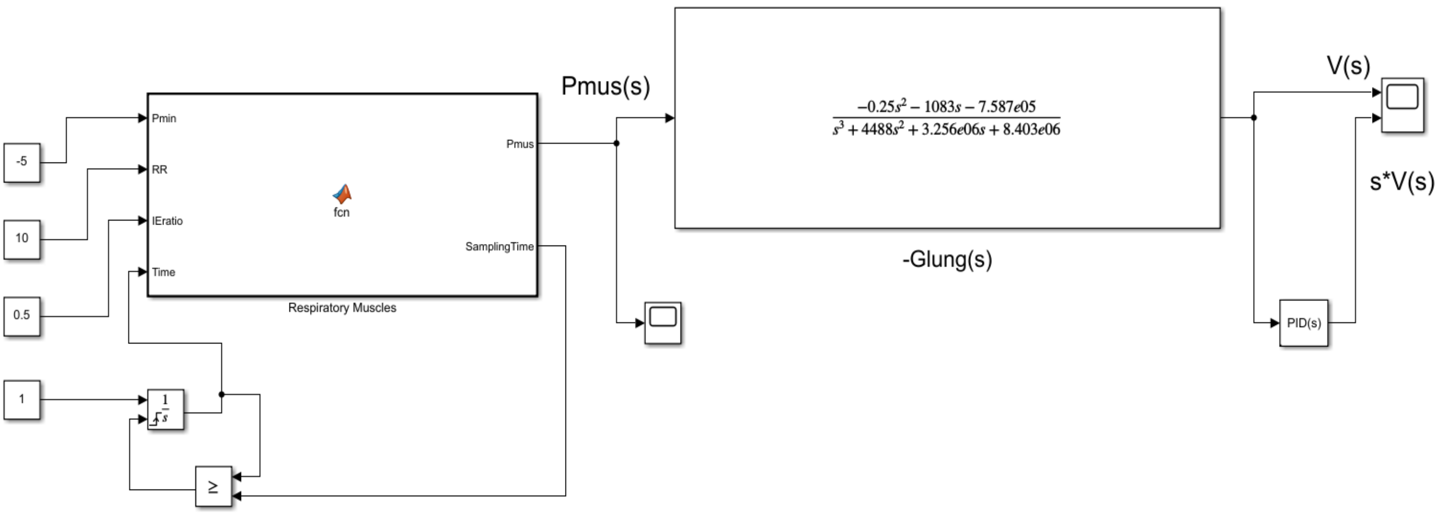


Figure 11: Applying Pmus(t) to Actual Transfer Function of The System

$G_{lung}\,(s)$ is multiplied by negative 1 because of the direction of $V'(s)$.

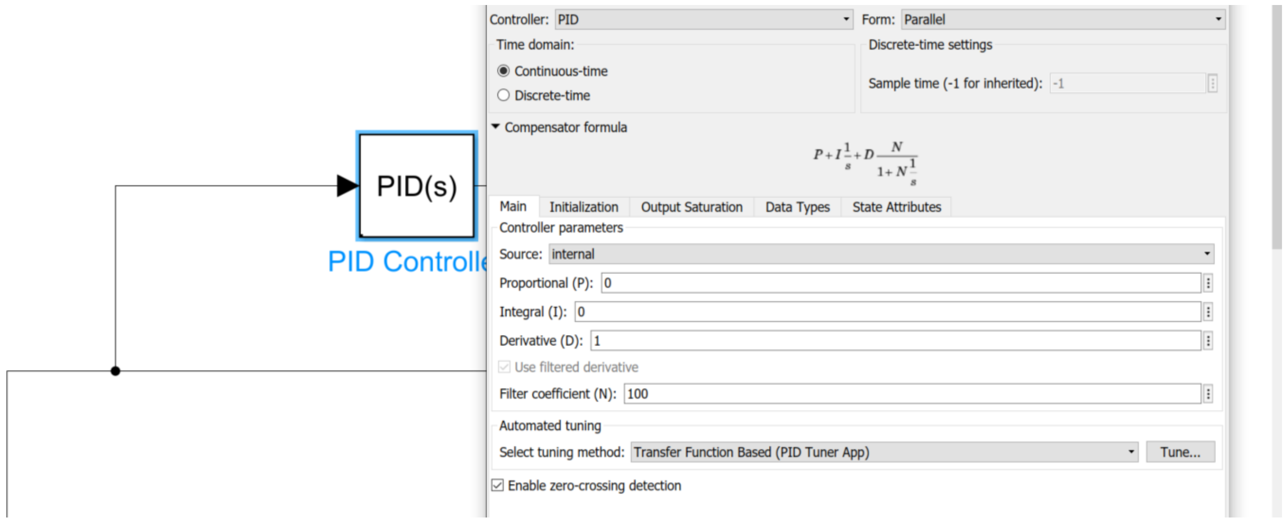


Figure 12: PID Values

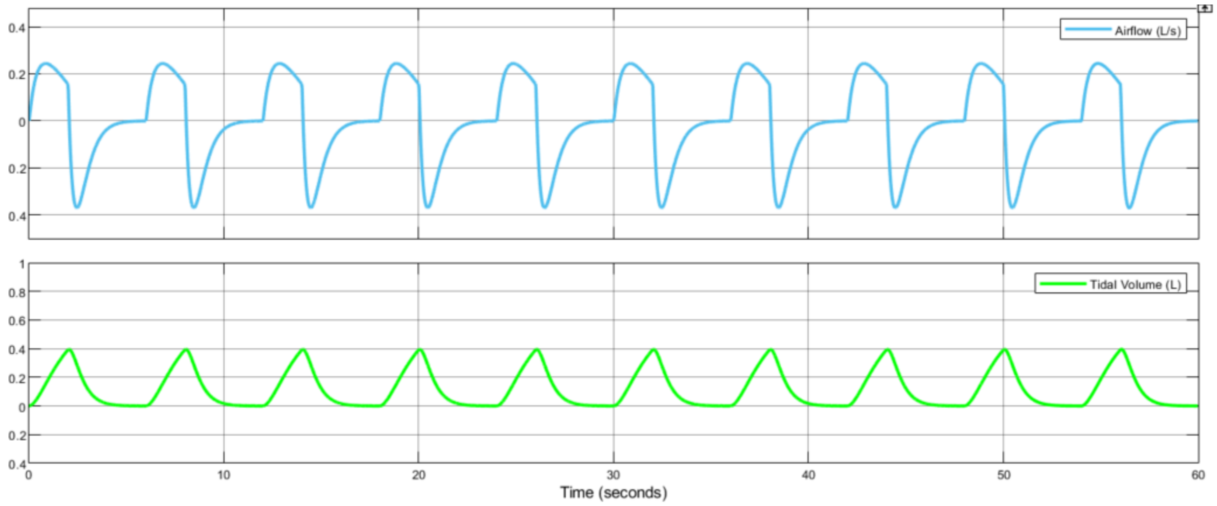


Figure 13: Tidal Volume $V(t)$ and Airflow $V'(t)$from Actual Transfer Function

By comparing figure 7 (previous result) and figure 13, it can be clearly seen that system responses are exactly same.

e.

```
Gs = 1/(s*Z_sum);
Gs = minreal(Gs);
[ p , z ] = pzmap(Gs);
```

Figure 14: Finding Poles and Zeros of the transfer function

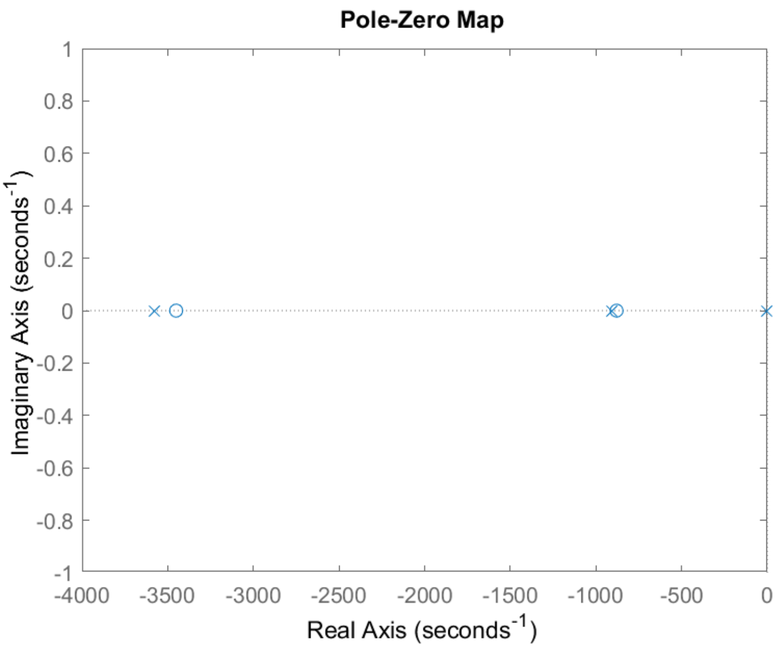


Figure 15: Pole Zero Map

The poles and zeros around -3500 and -1000 are very far from the imaginary axis. In addition, both the -3500 and -1000 poles and zeroes compensate the system response effects since they are close to each other, in this case only one root remains to represent the system.

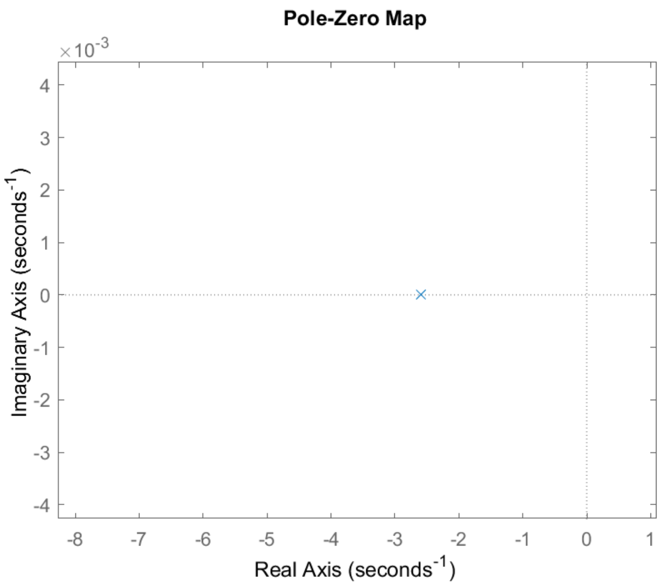


Figure 16: The dominant pole is at -2.589

```
step_value = step( Gs ); % y values of step response of Gs

final_term = step_value(length(step_value)); % to find final term
gain = final_term /1 ; % Gain of Gs
```

Figure 16: Finding The Gain of The System

$$\text{gain} = 0.0903 \tag{1.16}$$

- $H(s) = \frac{\frac{Gain}{Tau}}{s+\frac{1}{Tau}}$ is the structure first order system transfer function

- Here $\frac{1}{Tau}$ is 2.589 (Dominant pole) and gain is 0.0903.

$$G_{lung}(s) = \frac{\frac{1}{R}}{s+\frac{1}{R\,x\,C}} \tag{1.17}$$

$$\frac{1}{R} = \frac{Gain}{Tau} \text{ and } \frac{1}{R\,x\,C} = \frac{1}{Tau} \tag{1.18}$$

```
R = 1/(gain*2.589);
C = 1/(R*2.589);
```

Figure 17 : Total Lung Resistance (R) and Total Lung Compliance (C) Parameters

$$\begin{aligned} R &= 4.278381019601902 \cong 4.278 \\ C &= 0.090279363950629 \cong 0.0903 \end{aligned} \tag{1.19}$$

$$G_{lung}\ (s) = \frac{0.234}{s + 2.589} \tag{1.20}$$

f.

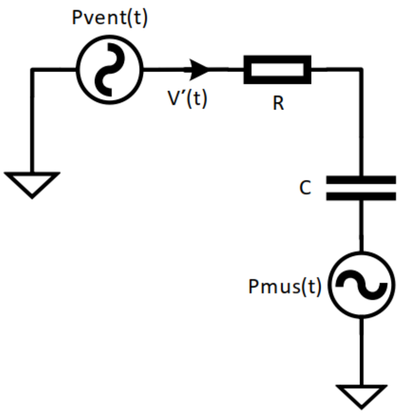


Figure 18: Single compartment model of the respiratory system.

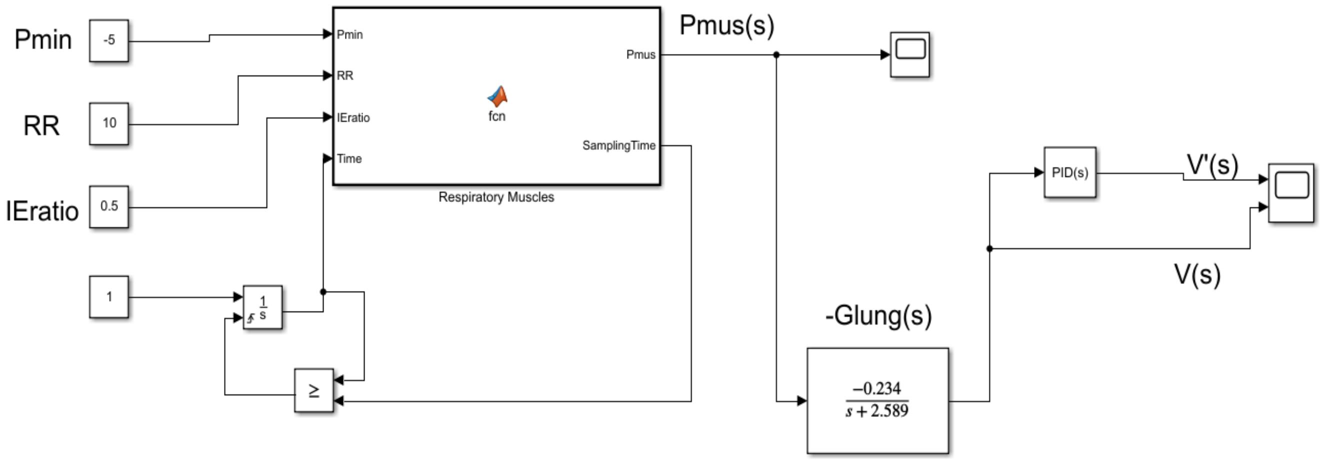


Figure 19: Applying Pmus(t) to Reduced Transfer Function of The System

$G_{lung}\ (s)$ is multiplied by negative 1 because of the direction of $V'(s)$.

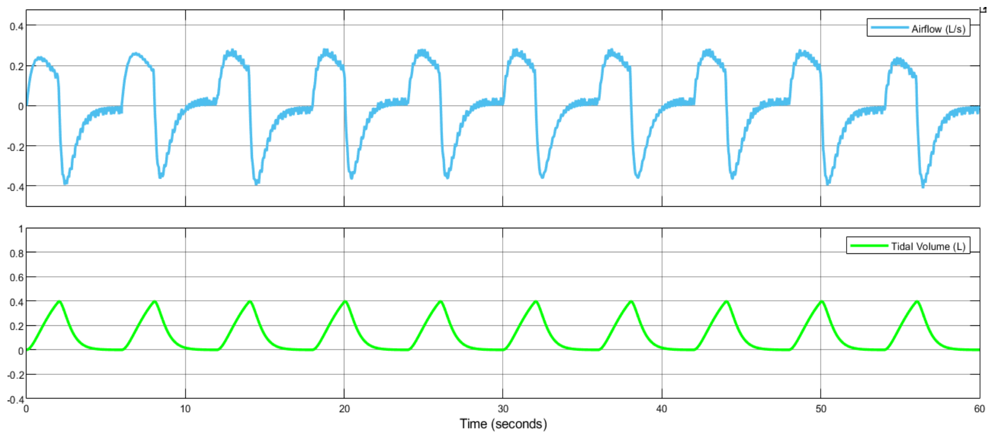


Figure 20: Tidal Volume $V(t)$ and Airflow $V'(t)$from Simplified Transfer Function

By comparing figure 13 (Result derived from exact transfer function) and figure 20 (Result from first order reduced transfer function), it can be seen that figure 20 approximates to figure 13 since effects of the non-dominant poles and zeros are negligible. When the effect of the other poles and zeros are calculated airflow curve become smoother. But it increases dramatically Matlab's response time.

Question 2

a.

```
syms s k w

T=0.1; %Time constant
R=4.278;
C=0.0903;

Glung = (1/R)/(s + (1/(R*C)));
Gf = 1 / ( T * s + 1);
Fs = k/s;

% The closed-loop transfer function
Ts = combine(Fs*Glung*Gf/(1+Fs*Glung*Gf));
```

Figure 21: Finding the closed-loop transfer function

Output of the closed-loop transfer function

$$\mathrm{Ts} = \frac{K}{0.4278s^3 + 5.3861s^2 + 11.07675 + k} \qquad (2.1)$$

```
% The characteristic polynomial
[~,Pcs] = numden(Ts);
```

Figure 22: Finding Characteristic Polynomial

Output of The Closed-Loop System Characteristic Polynomial

$$0.4278s^3 + 5.386s^2 + 11.07675 + k \qquad (2.2)$$

```
syms s k w

T=0.1; %Time constant
R=4.278;
C=0.0903;

Glung = (1/R)/(s + (1/(R*C)));
Gf = 1 / ( T * s + 1);
Fs = k/s;

% The closed-loop transfer function
Ts = combine(Fs*Glung*Gf/(1+Fs*Glung*Gf));

% The characteristic polynomial
[~,Pcs] = numden(Ts);

assume([w k],'real');
% Substitude s -> jw in order to check the stability
Pjw = subs(Pcs,s,1j*w);
% Find the critical frequencies that roots pass to the right half s-plane
solw = solve(imag(Pjw)==0,w)

First_Root=solve(subs(real(Pjw),w,solw(1))==0,k)%For the first root (w1)

fprintf("First_Root= %.2f\n",First_Root);

Second_Root=solve(subs(real(Pjw),w,solw(2))==0,k)%For the second root (w2)

fprintf("Second_Root= %.2f\n",Second_Root);


Third_Root=solve(subs(real(Pjw),w,solw(3))==0,k) % For the first root (w3)

fprintf("Third_Root= %.2f\n",Third_Root);
```

Figure 23: Finding Roots

```
solw =

                                          0
 -29145482717538185^(1/2)/33554432
  29145482717538185^(1/2)/33554432

First_Root= 0.00
Second_Root= 139.41
Third_Root= 139.41
```

Figure 24: Roots of the System

0 < k < 139.41,   0 >k,   k < 139.41 and k > 139.41 intervals are obtained. Within one of those intervals stability is satisfied. It can be understood from figure 24, 0 and 139.41 are critical values.

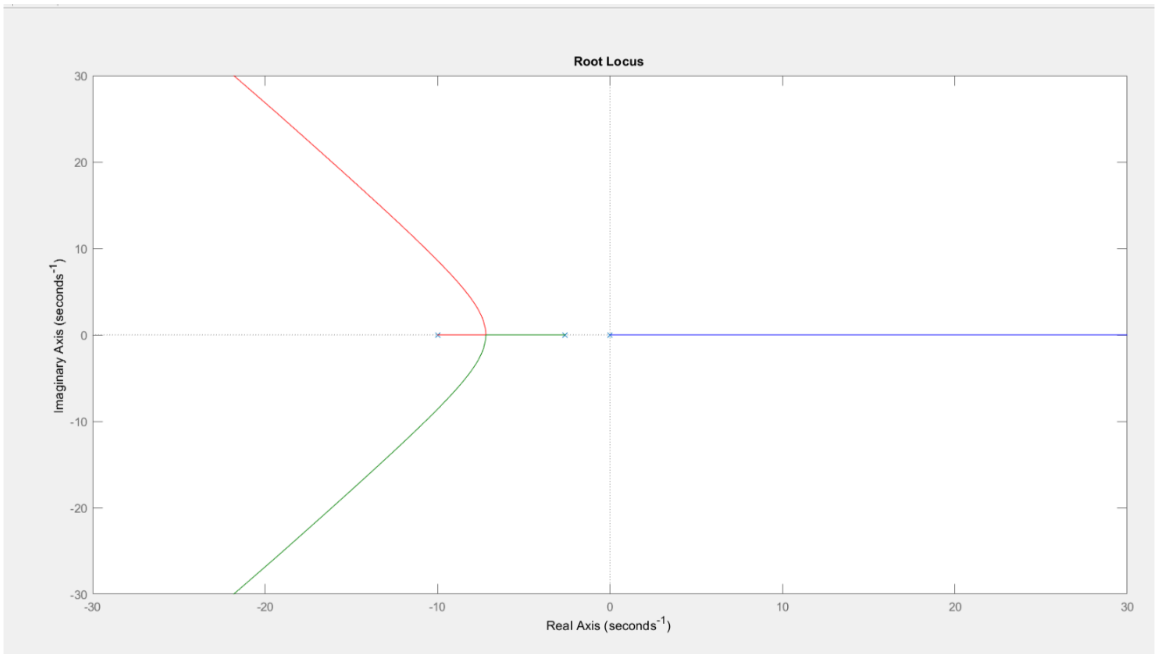


Figure 25: K<0

As it can be seen from the root locus graph, if K is less than zero, the system is unstable because the system has poles on the right half-plane for negative K values.

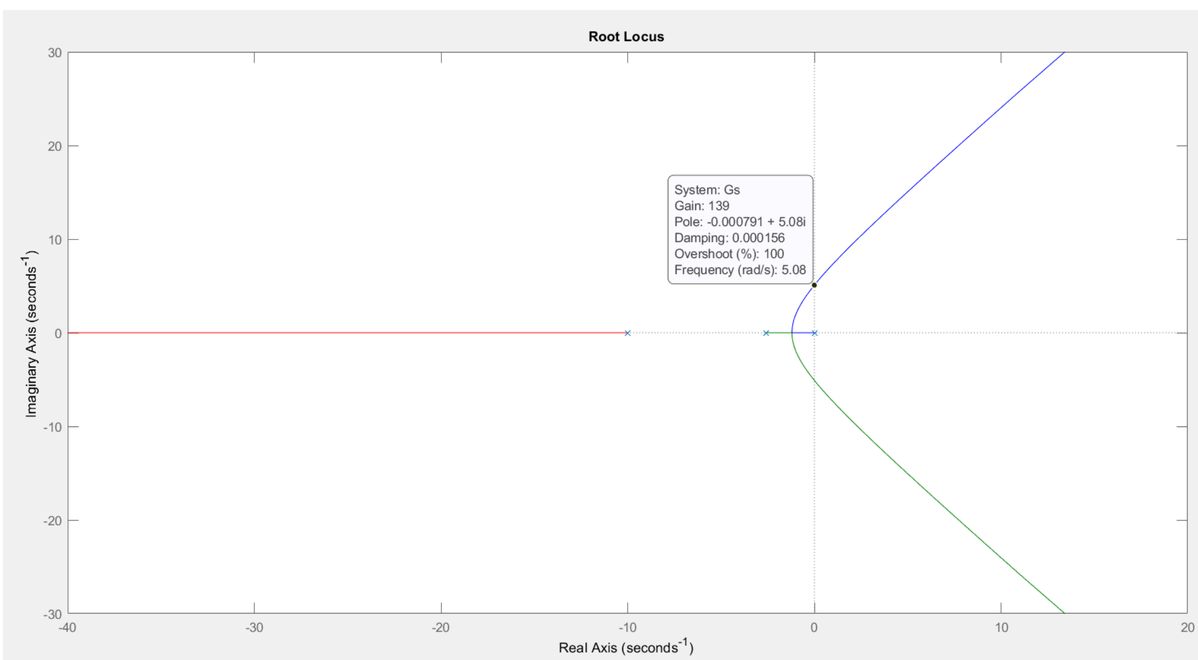


Figure 26: K>0

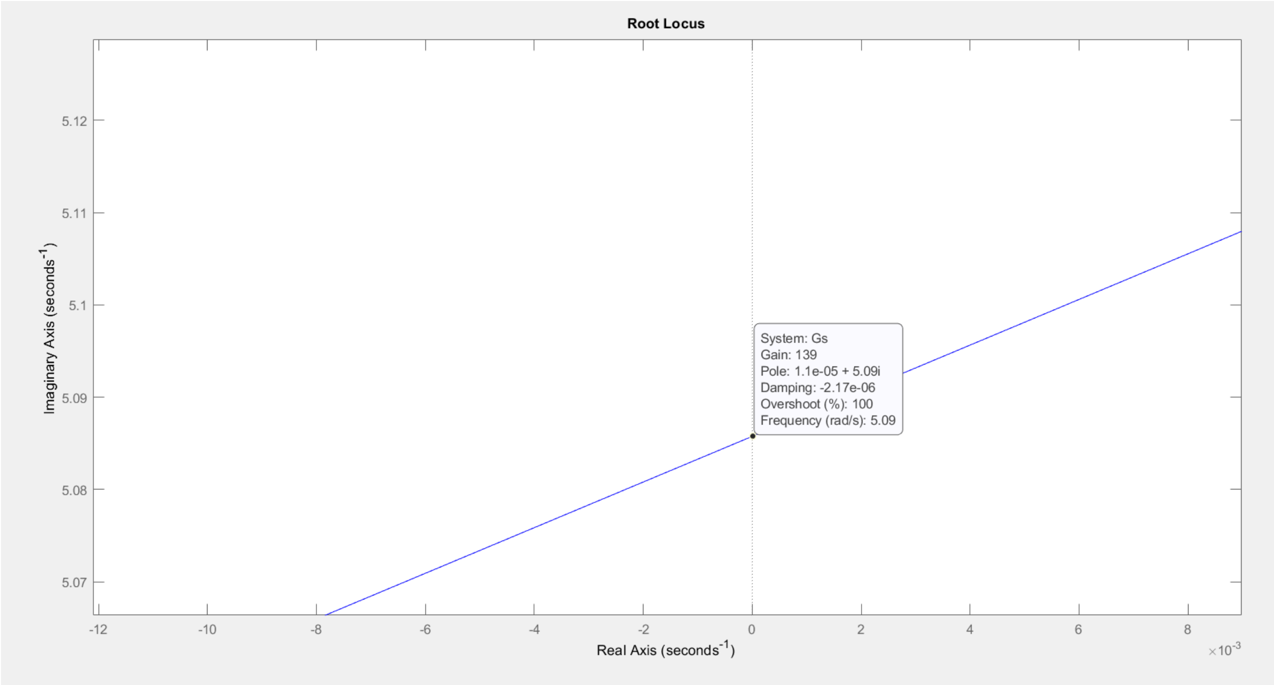


Figure 27: Zooming Root Locus (K>0)

For K >0, the critical gain of the system is 139.41. When the gain of the system is greater than 139.41, the system is unstable because there is one pole in the right half-plane.
As a result, system is stable between 0 and 139.41.

b.

```
syms s K R C Tf;

Glung = (1/R) / ( s + ( 1/(R*C) ));
Gf = 1 / ( Tf * s + 1);
Fs = K;
```

Figure 28: Finding parameters

$$\text{Glung} = \frac{\frac{1}{R}}{s+\frac{1}{RC}} \tag{2.3}$$

$$\text{Gf} = \frac{1}{T_f s+1} \tag{2.4}$$

$$\text{Fs}=K \tag{2.5}$$

```
G_open = Glung*Gf*Fs;
```

Figure 29: Output of G_open(s)

$$\frac{K}{R(s+\frac{1}{RC})(T_f s+1)} \tag{2.6}$$

```
G_closed = pretty(simplify(G_closed))
```

Figure 30: Output of G_closed(s)

$$\frac{CK}{CRT_f s^2+(CR+T_f)s+CK+1} \tag{2.7}$$

Simplified Version of G_Closed(s)

$$\frac{\frac{CK}{CRT_f}}{s^2 + \frac{(CR+T_f)}{CRT_f}s + \frac{(CK+1)}{CRT_f}} \tag{2.8}$$

Reference of the second order systems :

$$\frac{K_{system} {W_n}^2}{s^2 + 2\zeta W_n s + {W_n}^2} \tag{2.9}$$

$\zeta = 1$ → Critically damped systems are the systems with the least settling time among the 2nd degree systems.

$$2\text{zetaWn} = \frac{CR+T_f}{CRT_f} \tag{2.10}$$

$$Wn = \frac{CR+T_f}{2CRT_f} \tag{2.11}$$

$$Wn^2 = \frac{CK+1}{CRT_f} \tag{2.12}$$

$$\frac{(CR+T_f)^2}{4(CRT_f)^2} = \frac{CK+1}{CRT_f} \tag{2.13}$$

$$\frac{(CR+T_f)^2}{4CRT_f} - \frac{1}{C} = K \tag{2.14}$$

What happens to the minimum achievable settling time if the value of is C increased?

$$Wn = \frac{1}{2T_f} + \frac{1}{2RC} \tag{2.15}$$

$$T_S \cong \frac{4}{\zeta W_n} \text{ , (for \%2 tolerance band) (Settling time)} \tag{2.16}$$

If C (Compliances) increases, wn (natural frequency) decreases. Therefore, since wn increase, settling time decreases.

c.

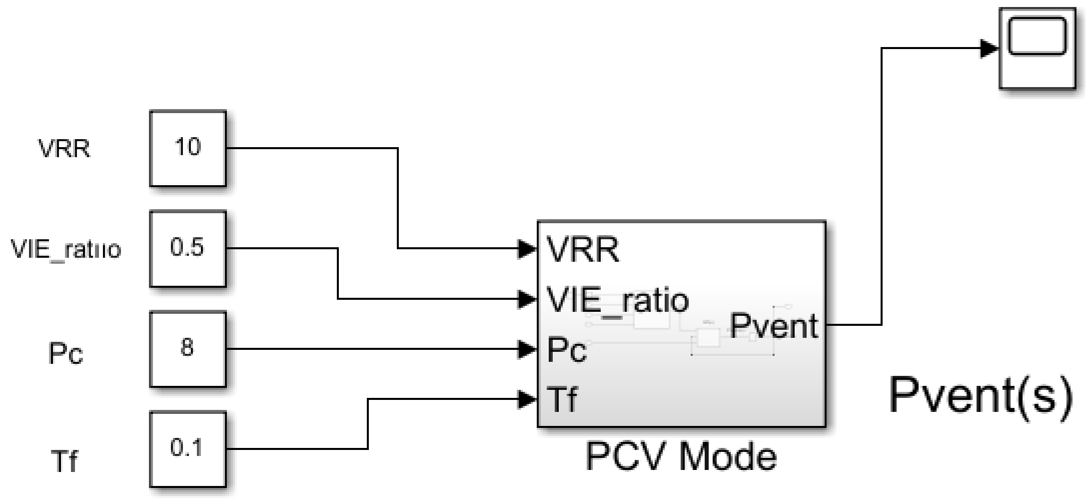


Figure 31: PCV Mode

Subsystem of the PCV Mod as shown in the figure 31. Scope is added in order to see Pvent graph.

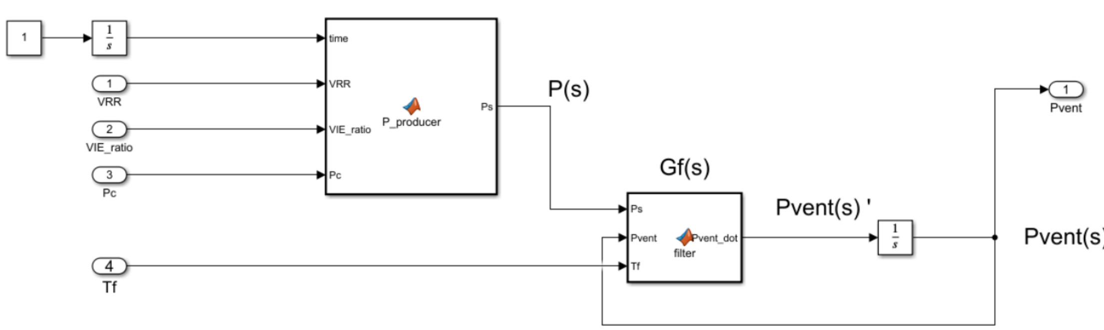


Figure 32: Subsystem of PCV Mode

Codes of the subsystem given figure 33 and 34

```
function Ps = P_producer(time,VRR,VIE_ratio ,Pc)
stop_time = 60; % stop time is given as 60 seconds
% Ti - Inspiratory Time (s)     Te - Expiratory Time (s)
% VRR - Ventilator Respiratory Rate (breaths/min) is 10.
% VIE_ratio ventilator inspiratory/expiratory rate is 0.5.
%
% There are 2 two equations.
%
%   60
% ------ = Ti + Te     and     Ti = Te x VIE
%   VRR
%
% in matrix form;
%  _  _     _              _     _  _
% | 60 |   | 1      1      |   | Ti |
% | 0  | = | 1  -VIE_ratio | * | Te |
%  -  -     -              -     -  -
%
%   b = A.T
%   T = inv(A).b

A = [ 1 1 ; 1 -VIE_ratio];
B = [stop_time/VRR ; 0];

T = inv(A)*B;

Ti = T(1); % Ti
Te = T(2); % Te
% We can set the period duration of the desired square wave
% as [0, Ti + Te] by performing modulus operation.
time = mod(time , Ti + Te );
 if( time <= Ti)
    Ps = Pc;
else
    Ps = 0;
end
end
```

Figure 33: Matlab Code of the P_producer

```
function Pvent_dot = filter(Ps,Pvent,Tf)
% Gf(s) = Pvent/ Ps = 1/( Tf*s + 1)
% Pvent* (Tf*s + 1) = Ps
% Pvent*Tf*s + Pvent = Ps
%
% In time domain
% Pvent(t)' * Tf + Pvent(t) = P(t)
% Pvent(t)' = (P(t)  - Pvent(t))/ Tf
Pvent_dot = (Ps  - Pvent)/ Tf;

end
```

Figure 34: Matlab Code of the Filter $G_f(s)$

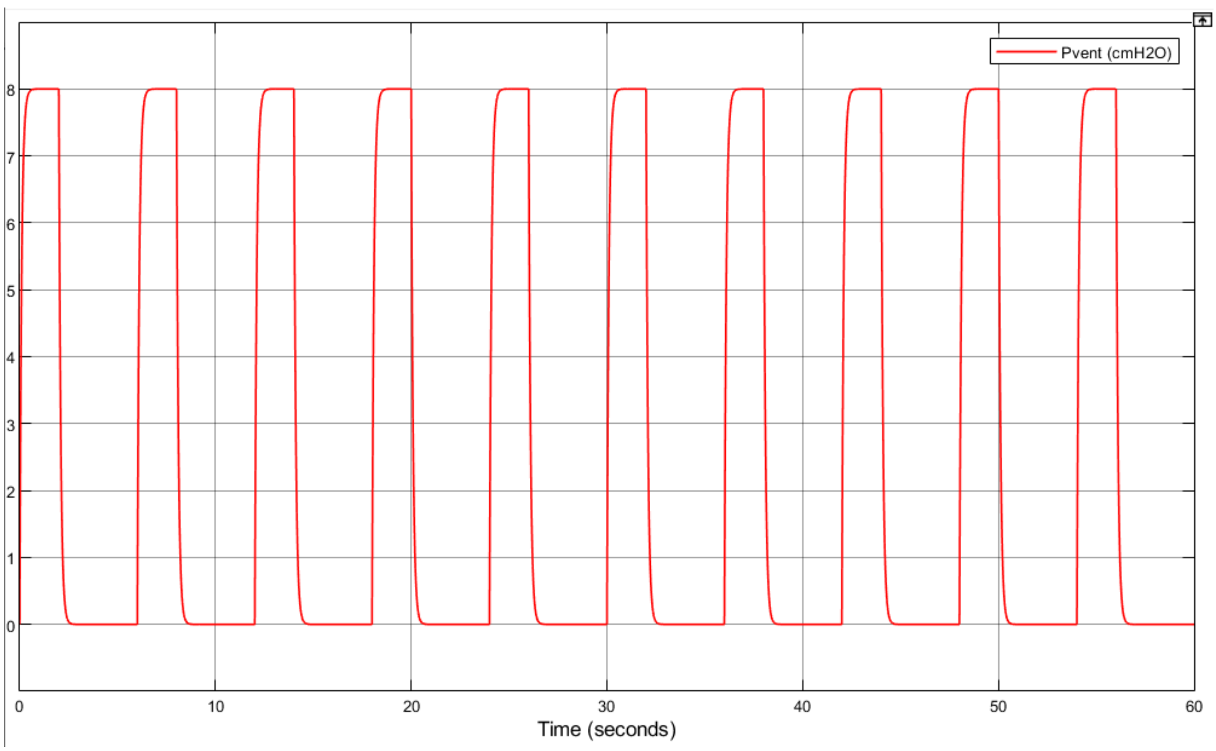


Figure 35: Pvent (cmH2O)

d.

$P_c = 8\ cmH_2O$	$\tau_f = 0.1\ s$
$\mathrm{VRR} = 10\ \mathrm{breaths/min}$	$\mathrm{VIE_{ratio}} = 1:2$

Figure 36: Device Settings

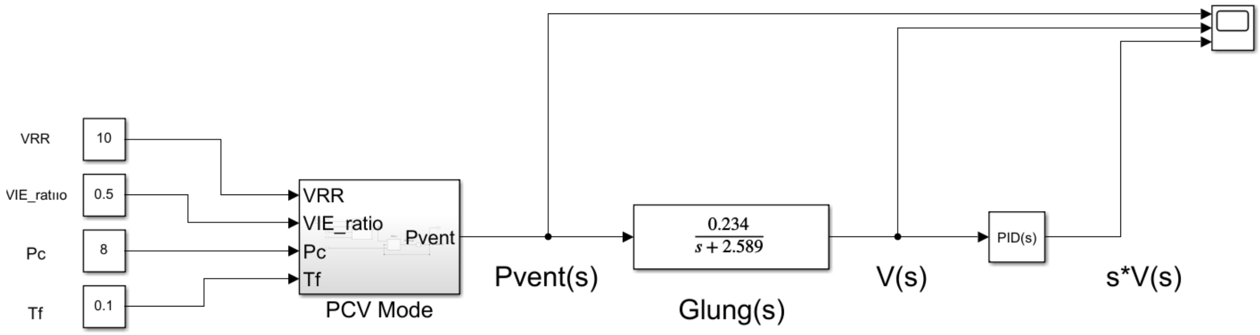


Figure 37: Testing MV subsystem on the first-order respiratory system , $P_c = 8$ cmH2O

$$\mathrm{Pvent(s)} * \mathrm{Glung(s)} = \mathrm{V(s)} \tag{2.17}$$

Vs(s)*s is equal to $V'(t)$ in time domain.

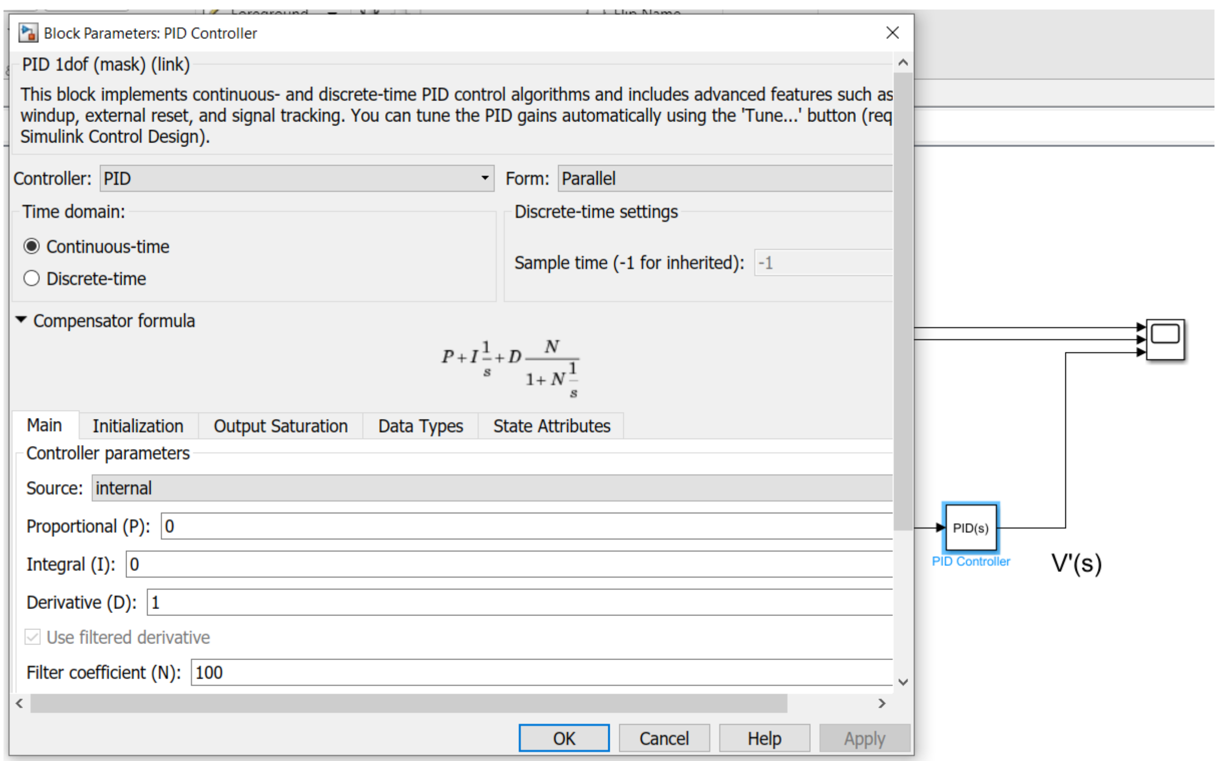


Figure 38: PID includes only derivative (D) parameter

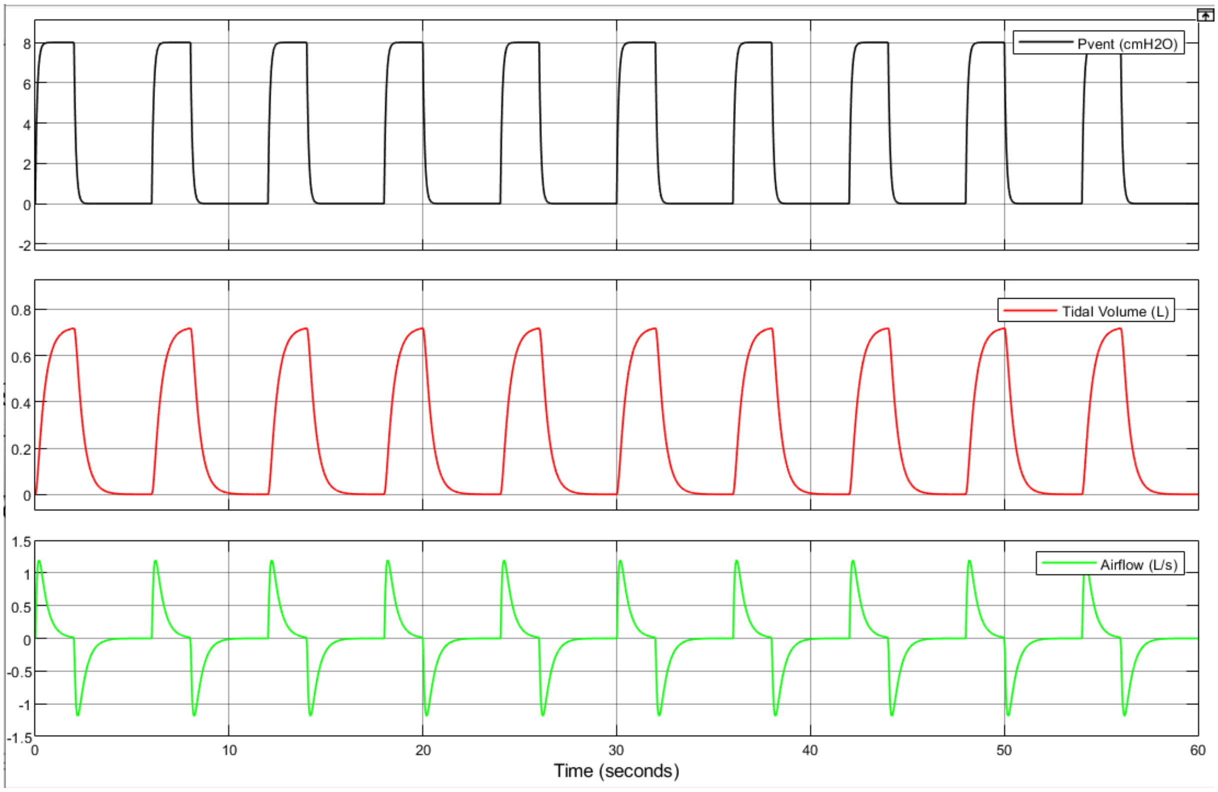


Figure 39: Plotting the produced signal $P_{vent}(t)$, airflow $V'(t)$ and tidal volume $V(t)$ $P_c = 8$ cmH2O

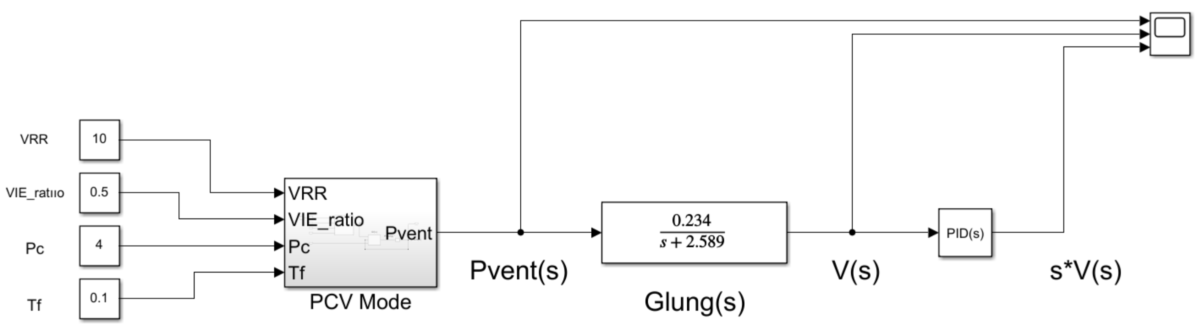


Figure 40: Testing MV subsystem on the first-order respiratory system, $P_c = 4$ cmH2O

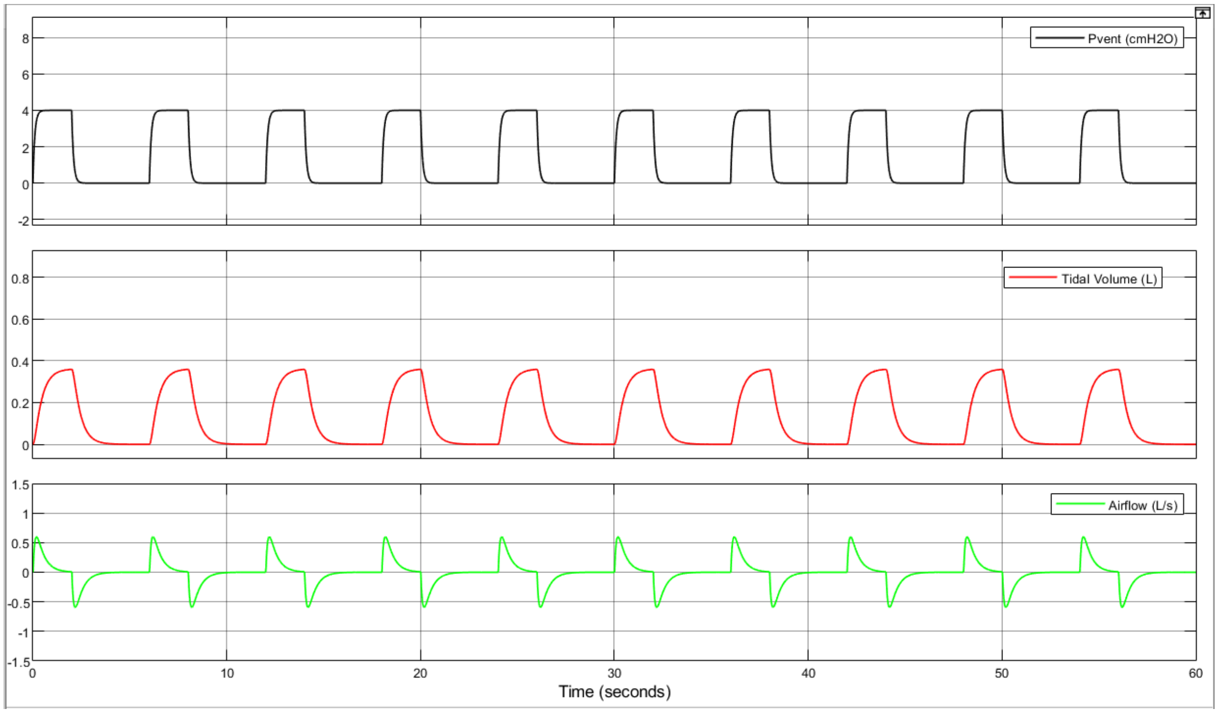


Figure 41: Plotting the produced signal $P_{vent}(t)$, airflow $V'(t)$ and tidal volume $V(t)$ $P_c = 4$ cmH2O

First of all PCV mode is open-loop mod. Since there is no feedback, system can't be controlled. For this reason, if the input signal somehow changed, airflow V'(t) would change, thus this can be result in fatal effects on the patient.

e.

$$\text{Glung(s)} = \frac{0.234}{s+2.589} \tag{2.18}$$

$$\frac{1}{R} = 0.234 \tag{2.19}$$

If the airway resistance (R) is increased by 100%

$$\frac{1}{2R} = 0.117 \tag{2.20}$$

$$\frac{1}{RC} = \frac{2.589}{2} = 1.2945 \tag{2.21}$$

$$\text{Glung}(s) = \frac{0.117}{s+1.2945} \tag{2.22}$$

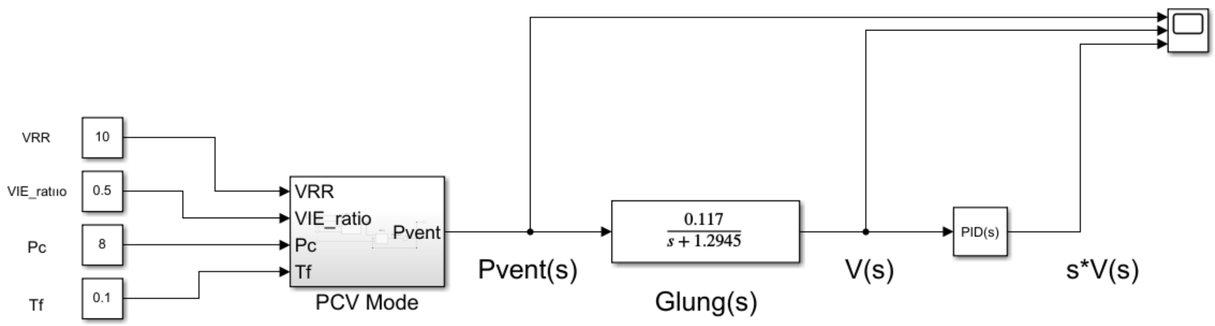


Figure 42: Testing MV subsystem on the first-order respiratory system If the airway resistance (R) is increased by 100%

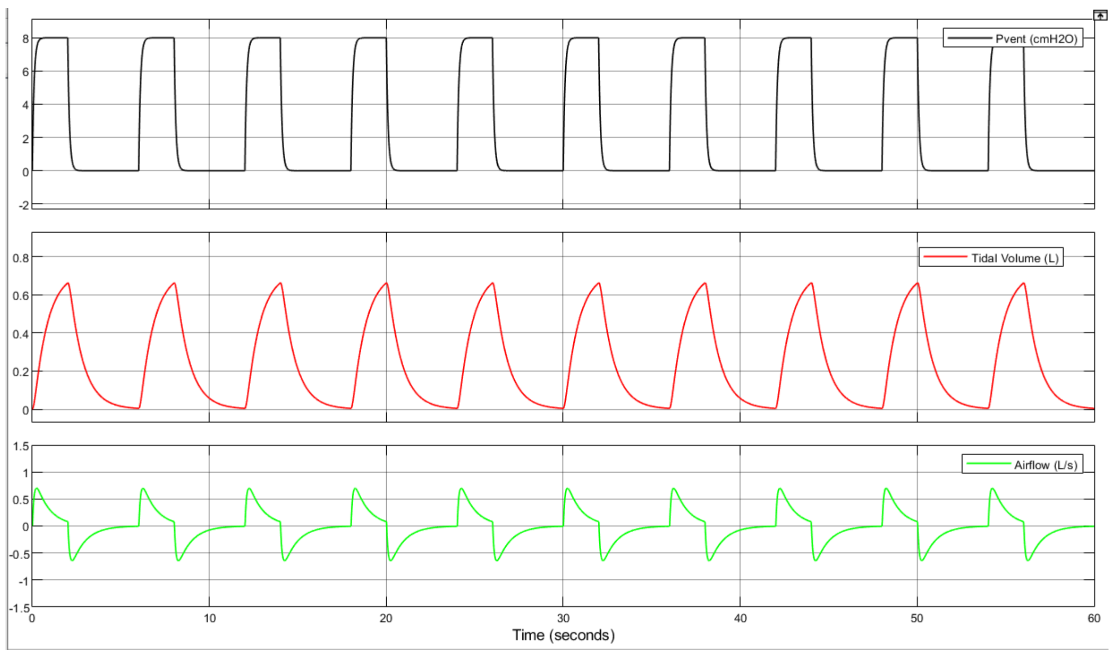


Figure 42: Plotting the produced signal $P_{vent}(t)$, airflow $V'(t)$ and tidal volume $V(t)$, If the airway resistance (R) is increased by 100%

Comparing figure 42 and figure 39

The more airway resistance that lungs have, the less airflow occurs. Also, maximum amplitude of the tidal volume decreases. For better comparation expiratory time and inspiratory time should be examined separately.

1. Inspiratory time
   There is slight change in amplitudes of the the airflow (V'(t)) and tidal volume (V(t)).

2. Expiratory time
   If airway resistance is doubled, the time that it takes for patient to expire increases. Also, in first case graph for expiration time was steeper, and second case graph is flatter.

f.

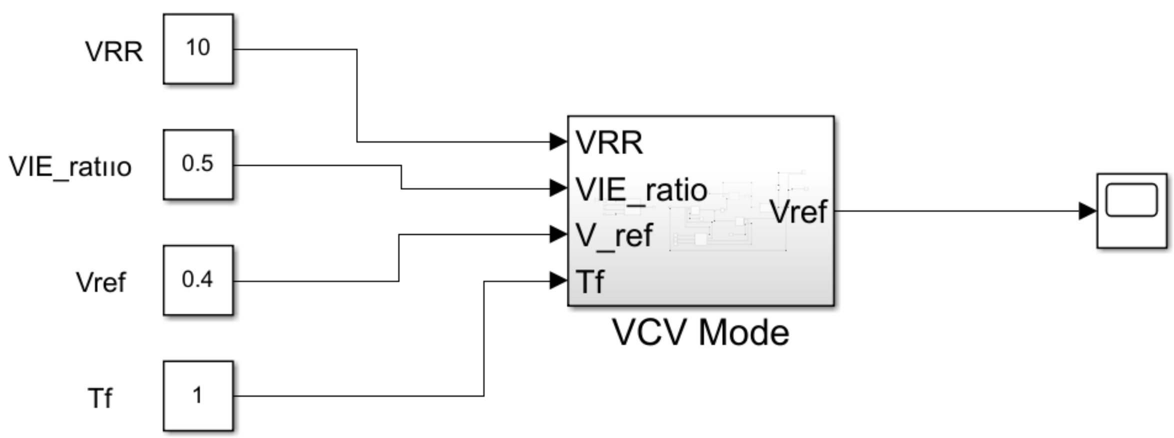


Figure 42: VCV Mode

VCV Mode subsystem includes 4 inputs:

- $V_{Ref\ (L)}$
- VRR (breaths/min)
- VIE_{ratio}
- T_f (s)

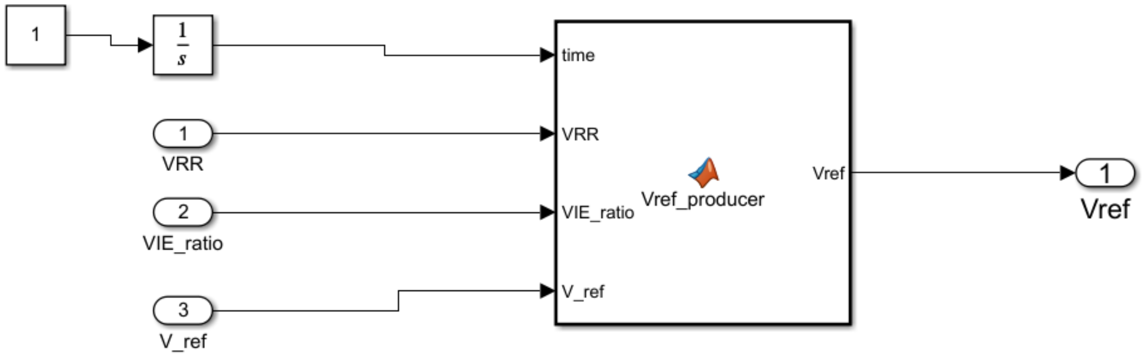


Figure 43: Square Wave Generating Function with Vref Amplitude

```
function Vref = fcn(time,VRR,VIE_ratio ,V_ref)
stop_time = 60; % stop time is given as 60 seconds
% Ti - Inspiratory Time (s)     Te - Expiratory Time (s)
% VRR - Ventilator Respiratory Rate (breaths/min) is 10.
% VIE_ratio ventilator inspiratory/expiratory rate is 0.5.
%
% There are 2 two equations.
%
%   60
% ------ = Ti + Te     and     Ti = Te x VIE_ratio
%   VRR
%
% in matrix form;
%  _  _     _             _     _  _
% | 60 |   | 1      1      |   | Ti |
% | 0  | = | 1  -VIE_ratio | * | Te |
%  -  -     -             -     -  -
%
%   b = A.T
%   T = inv(A).b
A = [ 1 1 ; 1 -VIE_ratio];
B = [stop_time/VRR ; 0];
T = inv(A)*B;
Ti = T(1); % Ti
Te = T(2); % Te
% We can set the period duration of the desired square wave
% as [0, Ti + Te] by performing modulus operation.
time = mod(time , Ti + Te );
if( time <= Ti)
    Vref = V_ref;
else
    Vref = 0;
end
end
```

Figure 44: Matlab Code of the Vref (L) producer

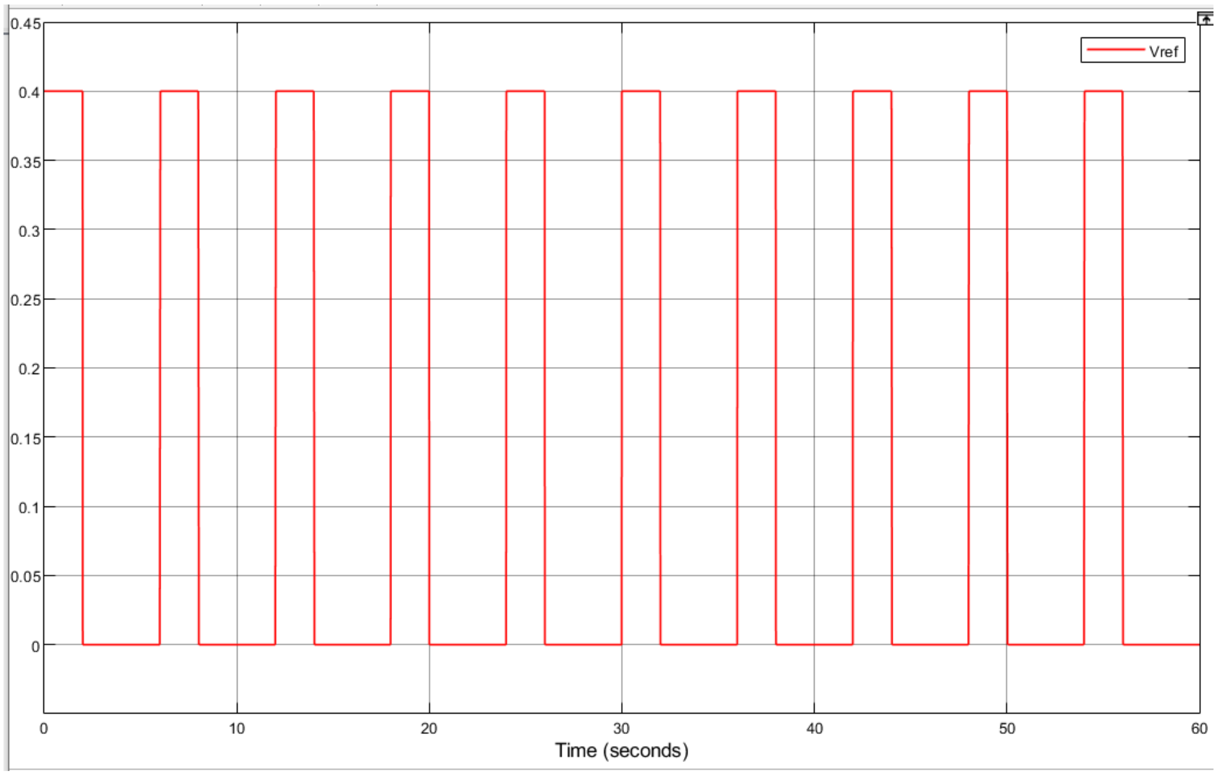


Figure 45: Vref (L) Producer Graph

g.

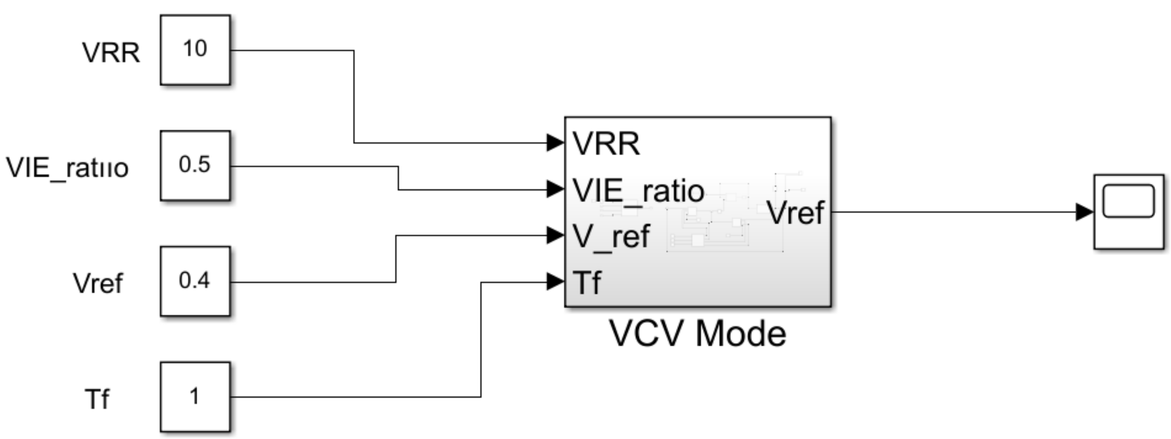


Figure 46: VCV Mode

VCV Mode subsystem includes 4 inputs:

- $V_{Ref\ (L)}$
- VRR (breaths/min)
- VIE_{ratio}
- T_f (s)

$$e(t) = Vref - V \tag{2.23}$$

$$y(t) = V(t) \tag{2.24}$$

$$u(t) = Pvent(t) \tag{2.25}$$

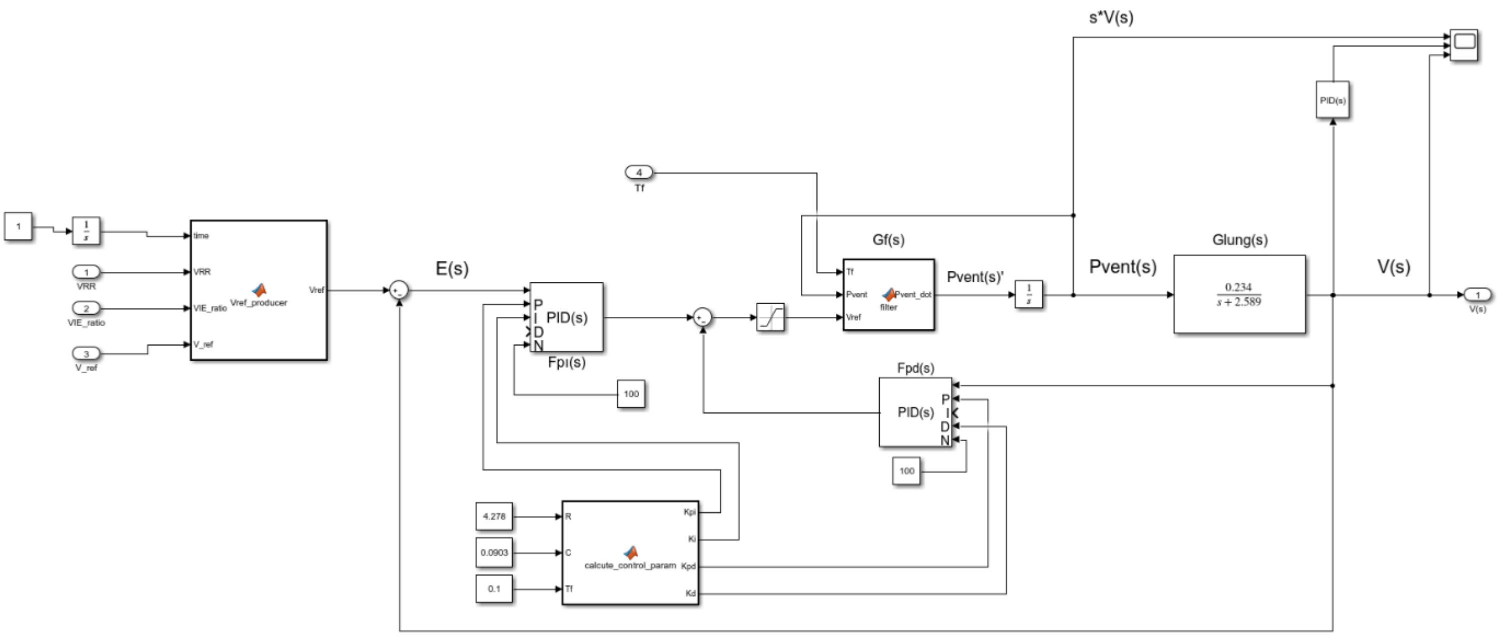


Figure 47: VCV Mode Whole System

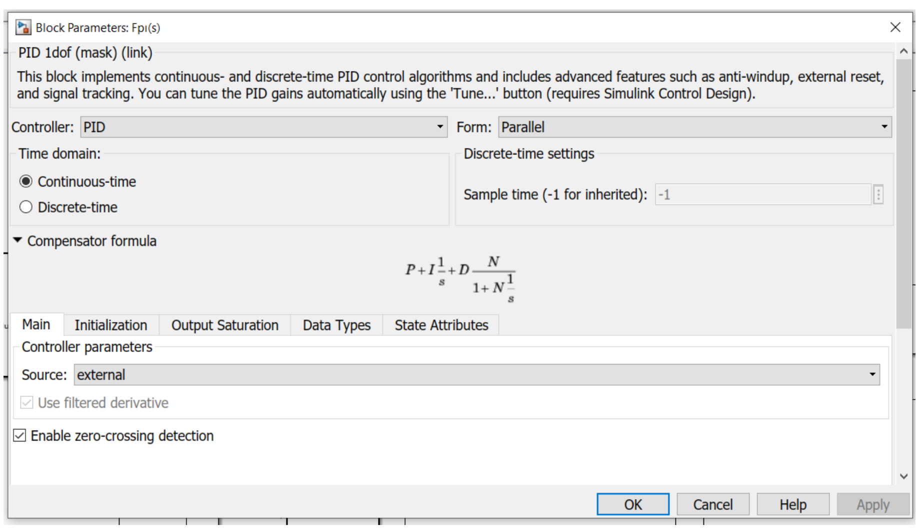


Figure 48: FPI Has External Source

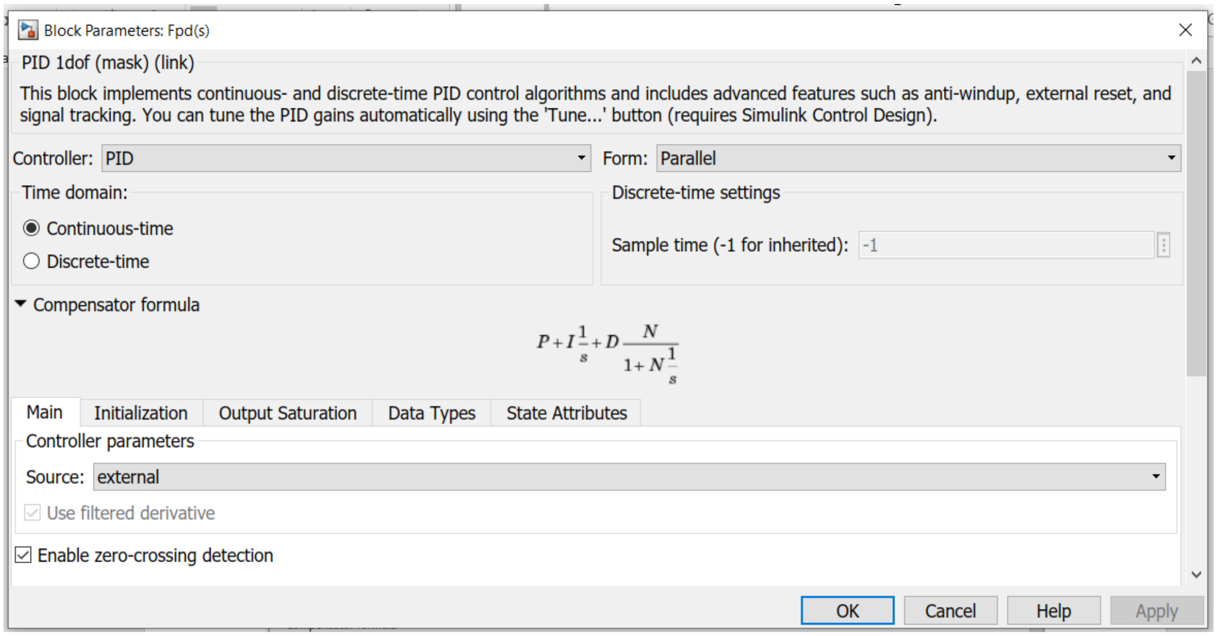


Figure 49: FPD Has External Source

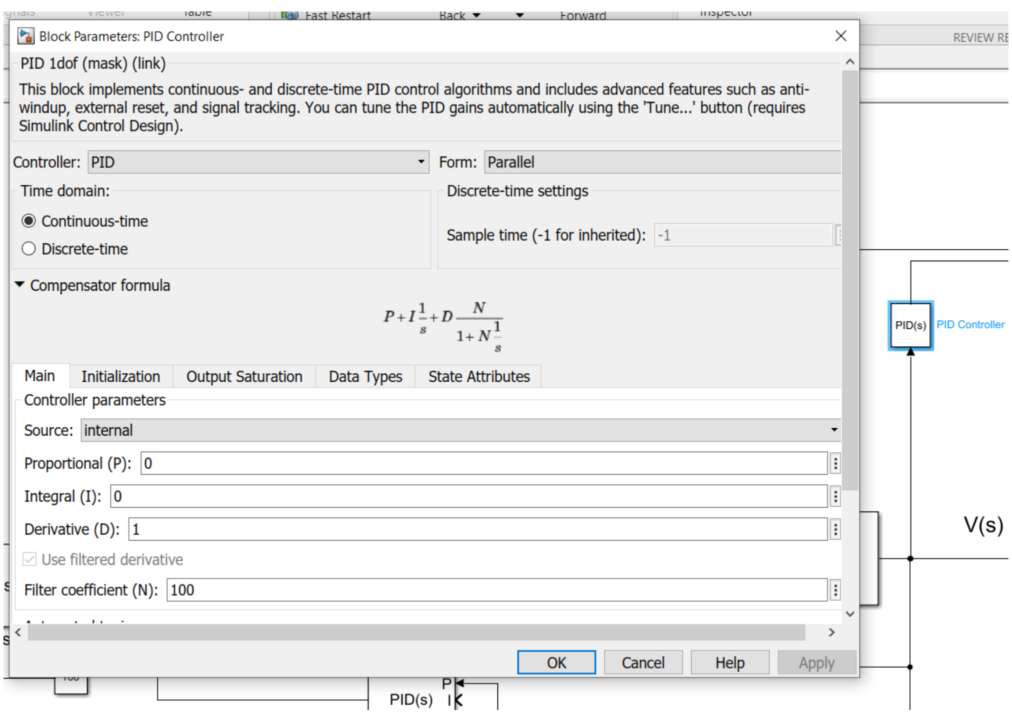


Figure 50: PID Block only used as derivative

$K_{pi} = \frac{16R}{25\tau_f}$	$K_i = \frac{64R}{25\tau_f^2}$
$K_{pd} = \frac{32R}{5\tau_f} - \frac{1}{C}$	$K_d = \frac{23R}{5} - \frac{\tau_f}{C}$

Figure 51: Control Parameters

These parameters are calculated as shown in the figure 52

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


function Pvent_dot = filter(Tf,Pvent,Vref)
% Gf(s) = Pvent/ Vref = 1/( Tf*s + 1)
% Pvent* (Tf*s + 1) = Vref
% Pvent*Tf*s + Pvent = Vref
%
% In time domain
% Pvent(t)' * Tf + V(t) = Vref(t)
% Pvent(t)' = (Vref(t)  - Pvent(t))/ Tf
% Pvent_dot = (Vref  - Pvent)/ Tf;
Pvent_dot = (Vref  - Pvent)/ Tf;

end
```

Figure 52: Calculate Control Parameters

h.

```
R = 4.278; % R value of the Glung
C = 0.0903; % C value of the Glung
Tf = 0.1; % Time constant of Gf - Filter

% Control Parametres calculations
Kpi = (16*R)/(25*Tf);
Kpd = (32*R)/(5*Tf) - 1/C;
Ki = (64*R)/(25*Tf^2);
Kd = (23*R)/5 - Tf/C;

s = tf('s');

Glung = (1/R) / ( s + 1/(R*C) );
Gf = 1 / ( Tf*s + 1 );
Gs = Glung * Gf;

Fpi = Kpi + Ki/s;
Fpd = Kpd + Kd *s;

Ts2 = (Gs*Fpd) / ( 1 + Gs*Fpd ); % Block reduction between output of Fpi(s) and V(s)
Ts =  (Ts2*Fpi) / ( 1 + Ts2*Fpi ); % Transfer function of the system

Ts = minreal(Ts); % to cancel pole-zero pairs in transfer functions
```

Figure 53: Calculating Transfer Function

```
          1189 s^7 + 1.418e05 s^6 + 6.314e06 s^5 + 1.329e08 s^4 + 1.434e09 s^3 + 7.847e09 s^2 + 1.971e10 s + 1.803e10

T(s) = ---------------------------------------------------------------------------------------------------------------------------

       s^8 + 1310 s^7 + 1.473e05 s^6 + 6.433e06 s^5 + 1.343e08 s^4 + 1.441e09 s^3 + 7.866e09 s^2 + 1.973e10 s + 1.803e10
```

```
p =

   1.0e+03 *

 -1.190444250400139 + 0.000000000000000i
 -0.040000000000000 + 0.000000001753627i
 -0.040000000000000 - 0.000000001753627i
 -0.014124088027325 + 0.000000000000000i
 -0.009999999999999 + 0.000000002527509i
 -0.009999999999999 - 0.000000002527509i
 -0.002588638878146 + 0.000000000091238i
 -0.002588638878146 - 0.000000000091238i
```

Figure 54: Poles of Ts

```
z =

-39.999999999999844 + 0.000002842319221i
-39.999999999999844 - 0.000002842319221i
-14.146380010811388 + 0.000000000000000i
-10.000006330049334 + 0.000000000000000i
 -9.999993669971445 + 0.000000000000000i
 -2.588639092975812 + 0.000000000000000i
 -2.588638663316360 + 0.000000000000000i
```

Figure 55: Zeros of Ts

```
pzmap(Ts) % pole-zero map of T(s)
```

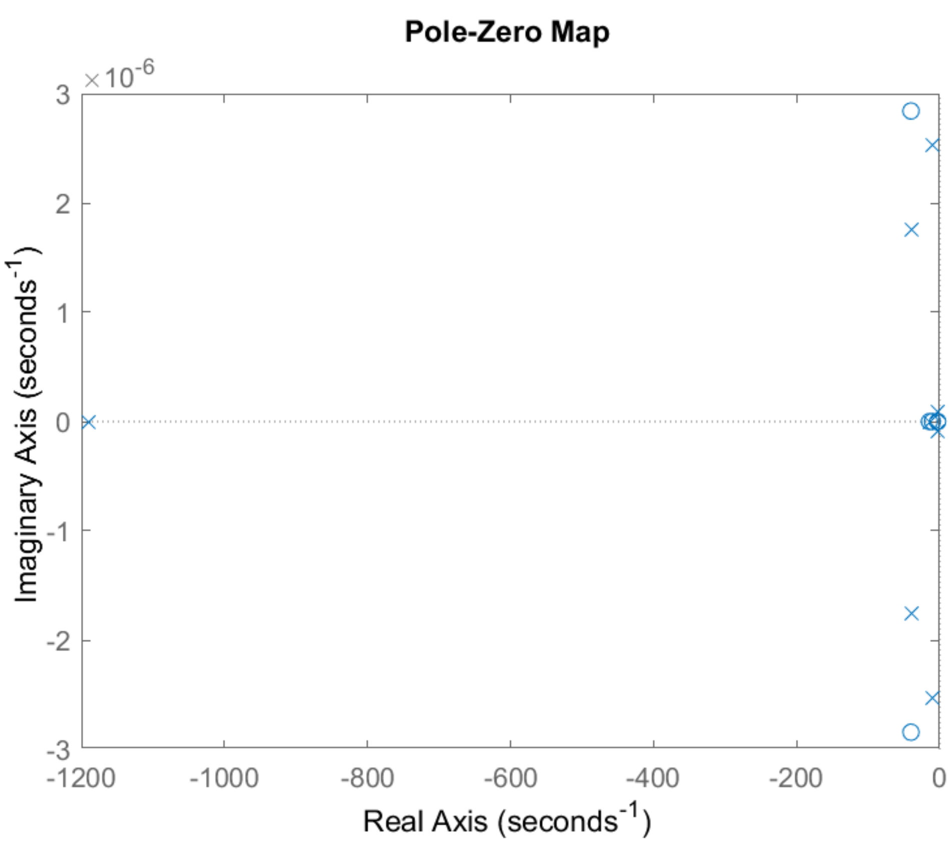


Figure 56: Pole Zero Map

As it can be seen from the figure 56, the y axis values are actually very small because of multiplication by 10^{-6}, therefore the dominant pole must be the pole around -1200. Since the remaining poles and zeros are very close, they eliminate each other system responses.

The dominant pole is -1190.44 .

```
Ts_y_values = step(Ts); % y values of step responses of T(s)

final_term = Ts_y_values(length(Ts_y_values)); % to find final term

gain = final_term / 1; % gain of the T(s), output / input , input is 1.


gain = 0. 998 is obtained

Tau = 1 / 1190.44;

G_reduced = ( gain / Tau ) / ( s + 1/Tau);

T_settling = 4 * Tau; % Estimated settling time of the system T(s) for %2 tolerance band
```

Figure 57: Finding System Parameters

$H(s) = \frac{\frac{Gain}{Tau}}{s+\frac{1}{Tau}}$ is the structure first order system transfer function

Here $\frac{1}{Tau}$ is 1190.44 (Dominant pole) and gain is 0.998.

Tau = $\frac{1}{1190.44} = 8.4\ x\ 10^{-4}$ *can be obtained*

$$G_{reduced}(s) = \frac{1188.06}{s+1190.44}$$

T_settling = 0.00336 s

Since both of the thas first order characteristics, they have no overshoot.

i.

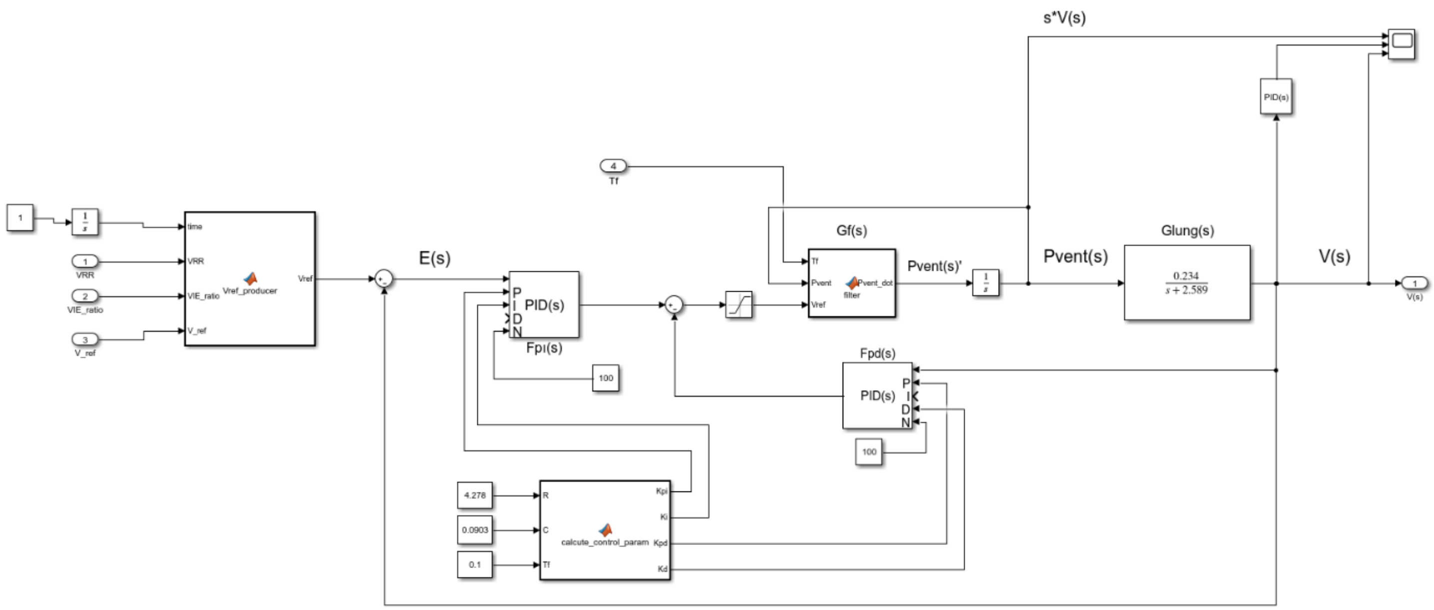


Figure 58: VCV Mode Whole System

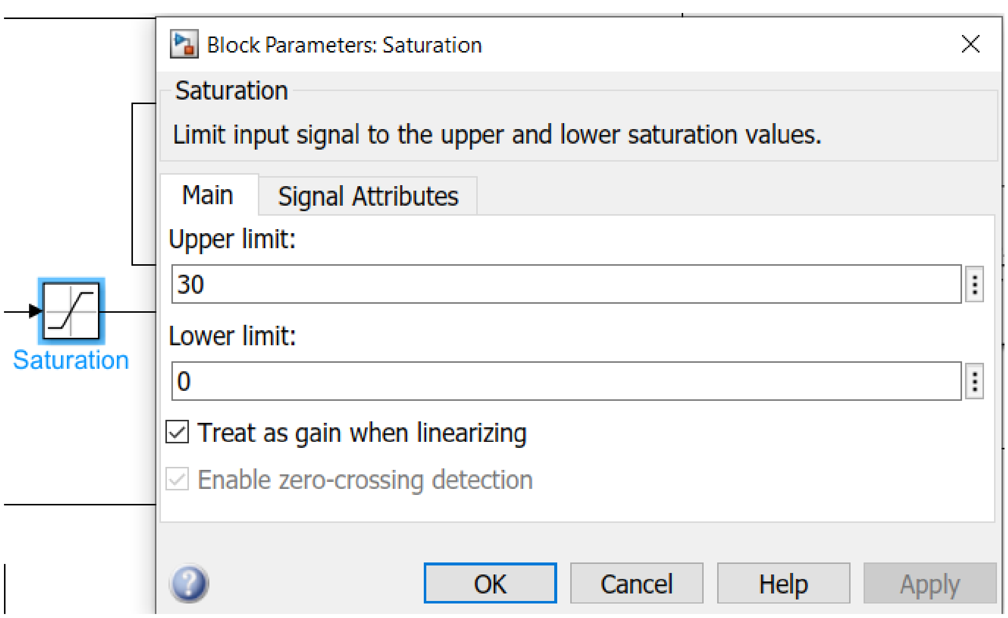


Figure 59: Saturation Block

By using saturation block output U(s) fit to [0,30] cmH20 interval.

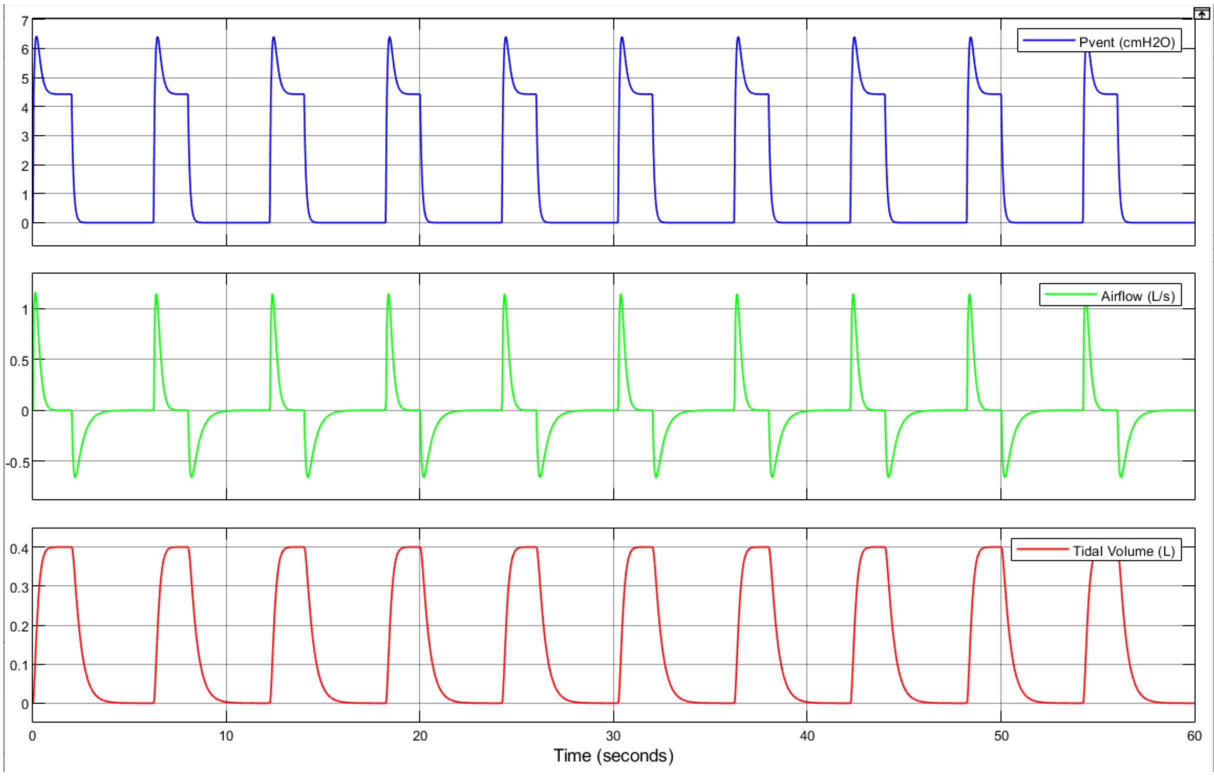


Figure 60: Plotting the produced signal $P_{vent}(t)$, airflow $V'(t)$ and tidal volume $V(t)$

Is the designed PI-PD controller able to control the considered system successfully?

Yes, PI-PD controller able to control the considered system successfully. Because Pvent(t) graph never crosses the negative region, which means that It is not only able to provide airflow to the patient but also it prevents the airflow to pass from patient to ventilator(since Pmust(t) is assumed to be zero).

While the ventilator operates in VCV, PI – PD controllers are designed in such a way that the system has no overshoot. This design restricts the output so that it doesn't exceed the input. Therefore, this property is desirable for our system because we need precision and accuracy since patient health is a concern. Also, PI-PD controllers decrease settling time. It is crucial for responding rapidly to patients varying health states.

j.

The plots of patient who has pulmonary emphysema are given below.

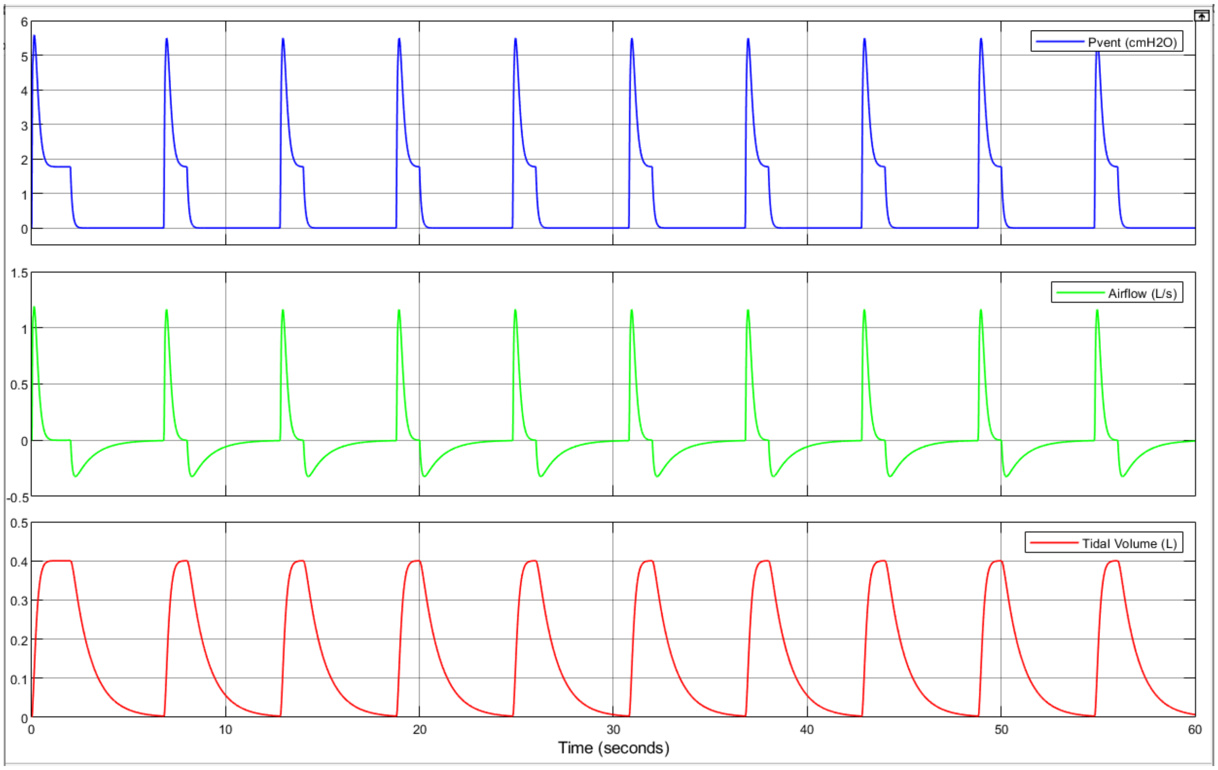


Figure 61: Plotting the produced signal $P_{vent}(t)$, airflow $V'(t)$ and tidal volume $V(t)$

Comparing figure 60 and figure 61

Since the lung capacity increased by %150 leads to keep more air in the lungs of the patient, so it is expected the area under the Pvent(t) graph less than the normal patient. Also, being able to hold more air in his/her lungs causes him to exhale over a longer period of time, that can be understood by looking at Tidal volume (V(t)) graph. Finally, there is slight change in amplitudes of the airflow (V'(t)) on inspiratory time.

Question 3

We analyzed the system by applying the knowledge we learned in the feedback control systems and the system modeling lectures. Also, it reinforced our ability by encouraging us to use our theoretical knowledge on solving a mainstream problem, and our analysis of this problem enabled us to better adopt the nature of the control engineering profession with first hand experience. Furthermore, we think that future generations will find this assignment easy one and we think that more difficult final projects should be given in the coming years. However, we are just limited by matlab. Since other platforms such as ROS can be integrated to matlab, and these platform can be learned individually. It would be much more beneficial to take an assignment about ROS - Matlab integration. In the final project, we think it would be more efficient for future generations to have more than one topic thus people can choose one of them. For example, one question might be about robot arm control while other questions might be about breathing apparatus or aircraft control. it would be very good incentive that if the assignments are presented in LaTeX than students' grades would be increased by ten or five. (Other faculty professors perform this criteria and nearly all students will have learned LaTeX at the end of the semester.)