Designing an Artificial Lung System that Models Asthma and Asthma Medication

By: Sanad Afaneh 1002307078

Omar Hegazi 1002311116

Yushan Tu 1001215758

For: BME344

Prof. Paul Yoo

TA: Jae Lee

December 1st, 2017

Abstract

Asthma is characterized by airway contraction due inflammatory response and the resulting hyperinflation of the lung. Constructing an adaptable lung model that accurately simulates the lung volume during normal respiration and during an asthma attack is very useful for clinical studies. Herein we constructed and validated a lung model based on Delawari and Doelman, which converted mouth pressure into lung volume, and designed a disturbance input that resembled different degrees of asthma symptoms, such as increased airway resistance, and a second input that characterized the effects of asthma medication. It was observed that the asthma disturbance input resulted in a positive vertical shift and amplitude decrease of the original output, and the asthma medication produced a reverse effect. This is similar to results observed in the literature and thus, we conclude that the proposed model is suitable for asthma studies.

1. Introduction

Asthma is a chronic inflammatory disease of the airways of the lungs that affects more than 2.4 million Canadians [3]. Symptoms include shortness of breath, wheezing, and chest tightness [1]. An asthma attack starts when the highly sensitive airway is inflamed by environmental triggers. This leads to an increase in the contractibility of airway smooth muscles and causes a period of airway constriction. The difference in airway flow rate causes the patient to inhale at total lung capacity, making the lung hyperinflated [2].

To study the lung volume of asthma patients and the effectiveness of various treatment methods, a lung model designed by Delawari and Doelman is used as a platform for simulating the normal respiration process of the lung. The lung model is a closed-loop system with mouth pressure (p_m) as input and lung volume (V_L) as output. The input is generated by a ventilator that is represented by square wave pulses of mouth pressure [10]. The system has several major compartments: the lung model, the controller, the voice coil, the piston, and the air compartment. The mouth pressure input is converted by the lung model and the controller into movement of the voice coil (sleeve position), which translates to lateral movement of the piston and the compression and expansion of the air compartment (the volume of the air compartment is the output). The Lung model is denoted by equation $I_L \frac{d^2}{dt^2} V_L(t) + R_L \frac{d}{dt} V_L(t) + \frac{1}{C_L} V_L(t) = p_m(t)$, where R_L is the resistance to air flow in the lung, C_L is the lung's ability to stretch during inhalation, and I_L is the acceleration of airflow and muscle in the lung. The air compartment is modeled through the equation $E_a(V(t) - V_o(t)) = -p_m(t)$, where E_a is air compressibility, $V(t)$ is the current volume of the compartment, and $V_o(t)$ is the volume of the outgoing air. The piston is represented by equation $V(t) = A_p p_s x(t)$, where A_p is the piston area, p_s is the gain of the lead screw pitch, and $x(t)$ is the horizontal displacement of the piston. The equations for the voice coil and the controller are more complicated and are discussed later in the Methods section [10].

The system proposed by Delawari and Doelman was built into a Simulink model and the output of the model was validated by comparing it to a literature model derived from Nunn's Applied Respiratory Physiology [9]. Later, the Delawari and Doelman system was modified and combined with two disturbance inputs in a situation of an asthma attack and drug injection. The asthma disturbance input was added after the lung model, which represented an increase in airflow

resistance during airway inflammation. The drug disturbance input was added after the voice coil, which resembled the dilation of the airway and decrease in the airflow resistance. Through Simulink, the design of asthma and medication input were simulated to test the accuracy of the model.

2. Methods

2.1. Model Construction

The model proposed by Delawari and Doelman was reconstructed using Simulink as shown in Figure 1. It used a periodic square wave input of period 4 seconds and amplitude 1, alongside a constant step size of 0.01 to provide a simplified representation of the mouth pressure during respiration [10].

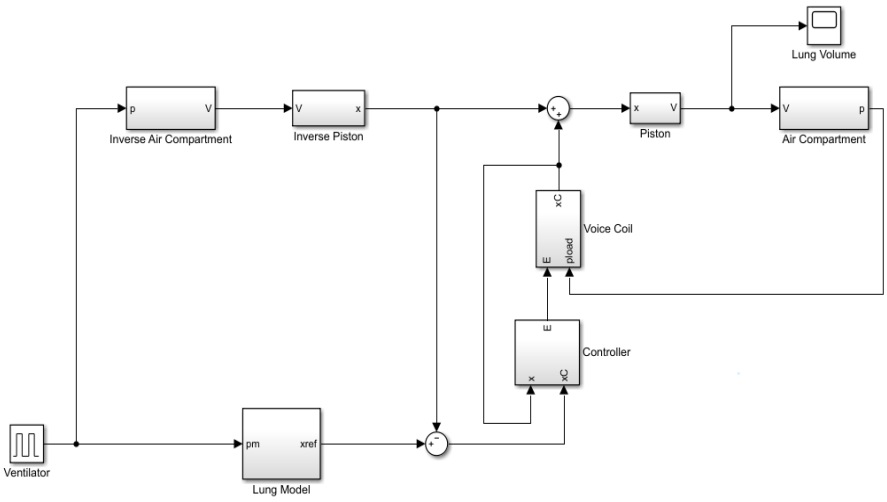


Figure 1 The Simulink model proposed by Delawari and Doelman.

The input represents mouth pressure during respiration, and the output is the corresponding lung volume. The input goes to both the "Lung Model", and the "Inverse Air Compartment". The "Lung Model" converts mouth pressure to displacement. The "Inverse Air compartment" converts mouth pressure to volume, which is then converted to displacement (x) in the "Inverse Piston"; the result is subtracted from the output of the "Lung Model" (x_{ref}). Such resulting output is then directed to the "Controller", which converts displacement signals to energy, followed by the "Voice Coil", which converts both energy and pressure (the feedback of the "Air Compartment") into displacement of the piston. The result, after summation, is then converted to lung volume.

It was necessary to derive some elements ("Lung Model", "Inverse Piston", "Inverse Air Compartment", and "Controller") of the proposed model that were not explicitly defined, then validate the model through comparison with the literature model derived from "Nunn's Applied Respiratory Physiology" [9]. We also sought to examine how this model behaves when a disturbance input that results in asthma symptoms is introduced to this system. Finally, we introduced an input representing a medication for asthma to observe how the system behavior changes.

2.2. The "Lung Model" and the "Inverse Piston"

The "Lung Model" incorporated in the model proposed by Delawari and Doelman converted mouth pressure into displacement, instead of converting it to volume. Hence, it was inferred that the component is composed of two sub-models; one converts mouth pressure into

volume, denoted by the transfer function $\frac{V(s)}{p_m(s)} = \frac{C_L}{R_L C_L s+1}$, where C_L is lung compliance (0.5 L kPa^{-1}), and R_L is lung resistance (1.1 kPa L^{-1} s^{-1}). The other sub-model converts volume to displacement and was determined to be the "Inverse Piston", a constant gain and a reciprocal of piston area, denoted by $\frac{x(s)}{V(s)} = \frac{1}{A_p p_s}$, where A_p is the area of a piston derived by Delawari and Doelman (2.4829 dm^2), and p_s is the displacement of that piston per screw revolution (0.025 dm).

2.3. The "Inverse Air Compartment"

The transfer function was estimated to be the reciprocal of the transfer function associated with the "Air Compartment"; hence, it was estimated to be:

$$\frac{V(s)}{p(s)} = \frac{s^2 + \frac{R_a}{I_a} s + \frac{E_a}{I_a}}{E_a s^2 + \frac{E_a R_a}{I_a} s}$$

where R_a is air flow resistance (0.8 kPa L^{-1} s^{-1}), I_a is air inertance (0.12 kg m^{-4}), and E_a is air elastance (100 kPa). The function was estimated by constructing one model, with a step input multiplied by the transfer function of the "Air Compartment". The analytical function of the output was derived using MATLAB. This function served as the input for another model and was multiplied by the reciprocal of the transfer function of the "Air Compartment". The output was graphically compared to the input of the first model, such that their root mean square error was insignificant (0.0179). Appendix A offers a detailed illustration of the approximation.

2.4 Controller Selection

P controller, PI controller, and PID controller were investigated as possible candidates for the "Controller". Manually tuning each of the three controllers resulted in gains that greatly exceeded the range of controller gain values (1000 to 2500) that were suggested by Delawari and Doelman (calculations provided Appendix B). Therefore, MATLAB was used to find the gain for each controller that has the lowest root mean square error (RMSE) value [5] by comparing the system outputs from Figure 1 in each case to the output of the literature model [9]:

$$\frac{V(s)}{p_m(s)} = \frac{1.83549}{(0.19607843137 \times 1.83549)\, s+1}$$

where the values of compliance and resistance were converted to kPa^{-1} and kPa L^{-1} s, respectively [8].

In addition, the gain margin and the phase margin for each controller were obtained from the Bode plot via MATLAB. The controller with phase margin that is furthest from 180 degrees was chosen as it provided the highest stability.

After obtaining the transfer functions for the unknown components, they were incorporated into the proposed model by Delawari and Doelman. The model's output was compared and validated by graphing its output and the literature value on the same axis.

2.5. Constructing a Disturbance Input of Asthma

No mathematical model was found in literature that relates asthma symptoms to lung volume. Hence, we constructed our own model. One of the effects that asthma yields on the human lungs is hyperinflation. Air gets trapped inside the lungs with each successive breath, as airway smooth muscle contraction makes expulsion difficult. Hence, a disturbance input, with an associated gain, representing such change should be added to the model.

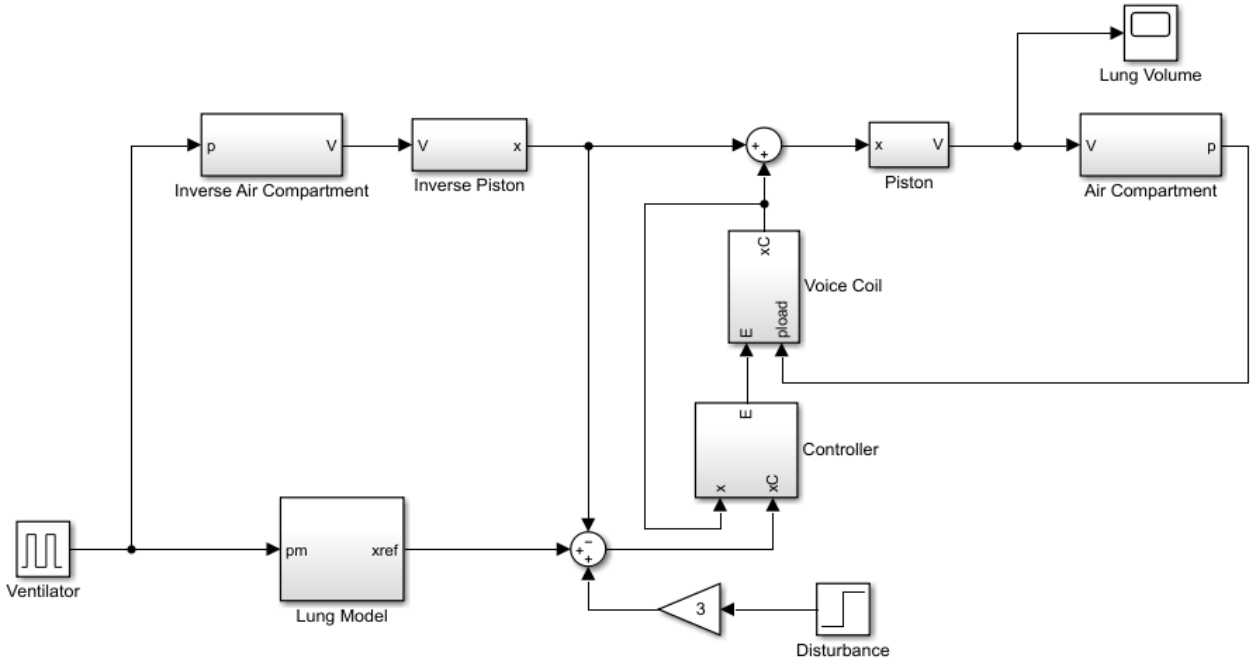


Figure 3 Adding a disturbance to model for asthma. Note that the gain value of 3 is not unique; it must be any constant greater than 1.

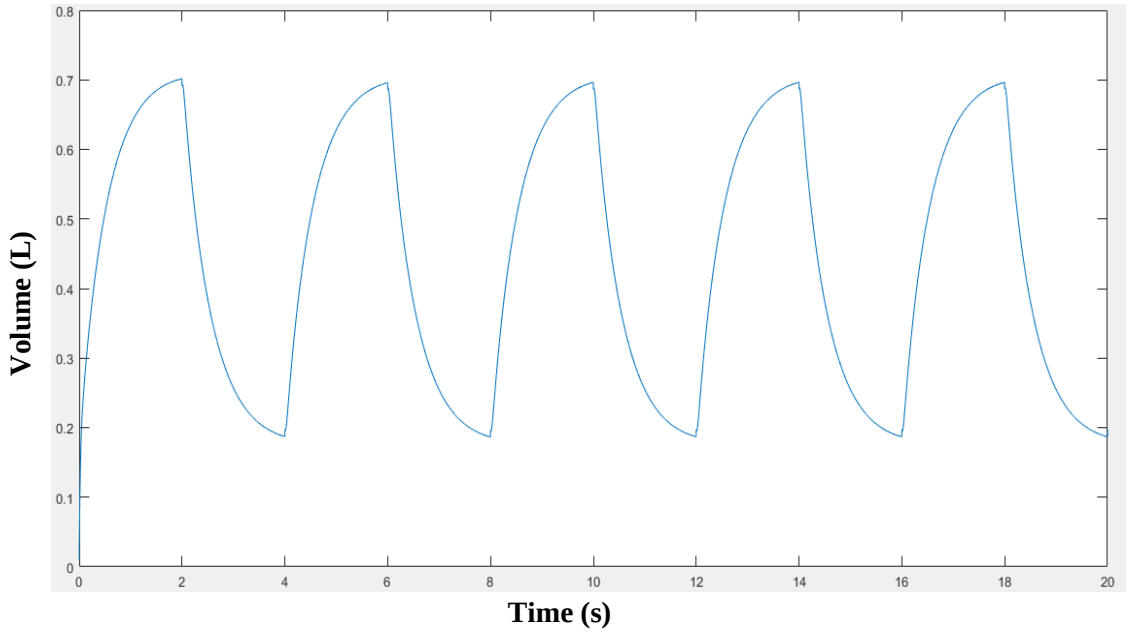


Figure 4 Output for the model in Figure 3.

From another point, asthma results in inflammation of the airways, such that they become swollen. The muscles around the airways tighten, so the airways become narrower; hence, less air flows to the lungs [7]. A gain block was incorporated into the "Lung Model" in Figure 3, such that it was placed right after the transfer function, to yield such change.

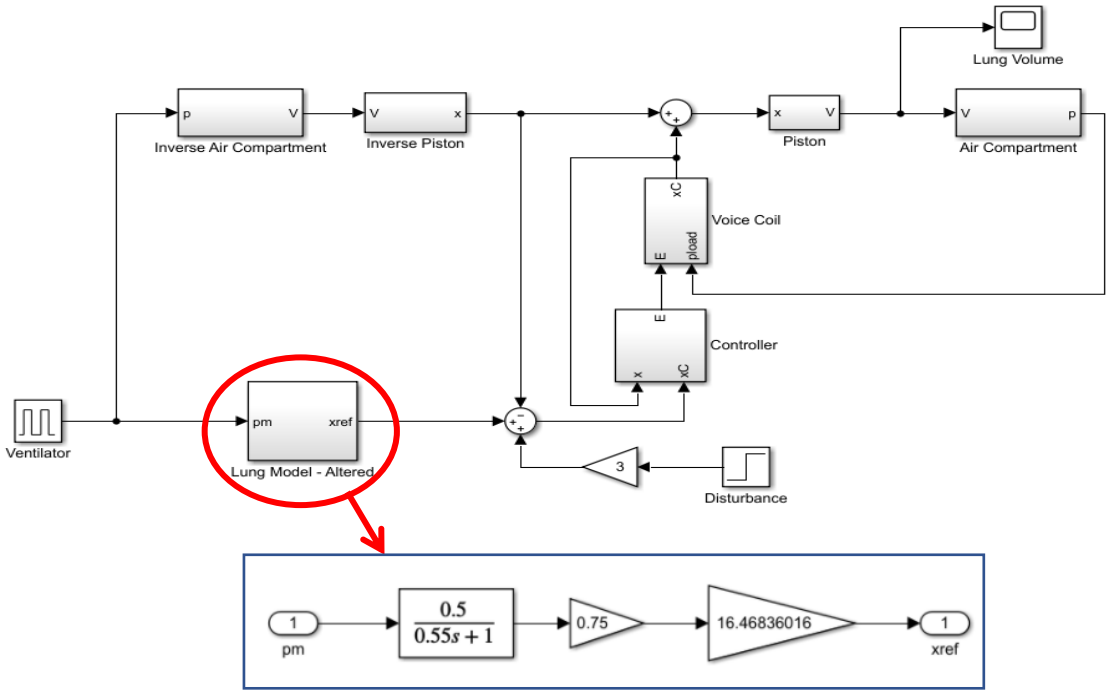


Figure 5 The modified Lung Model block to represent inflammation of the airways.

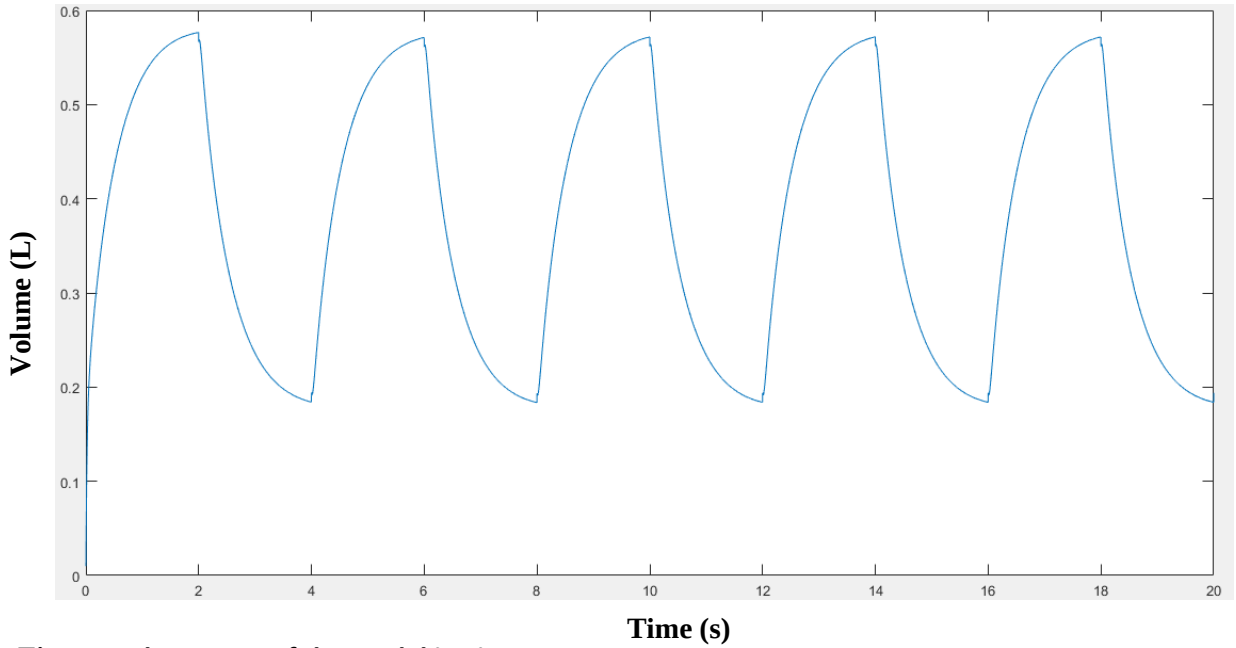


Figure 6 The output of the model in Figure 5.

Parameter sweeps were manually performed for the asthma model; the parameters altered were the gain introduced to the "Lung Model" (the gain block of value 0.75), and the gain associated with the disturbance input (the gain block of value 3). By linearly increasing the values of the gain introduced to the "Lung Model" from 0.125 to 1, with a constant increase of 0.125 each time (Figure 7), the lung volume increased, indicating that such output is proportional to the introduced gain. Since asthma is an obstructive disease, the lung volume decreases and this makes it harder to breathe. Therefore, the value of the gain introduced to the "Lung Model" must be lower than 1.

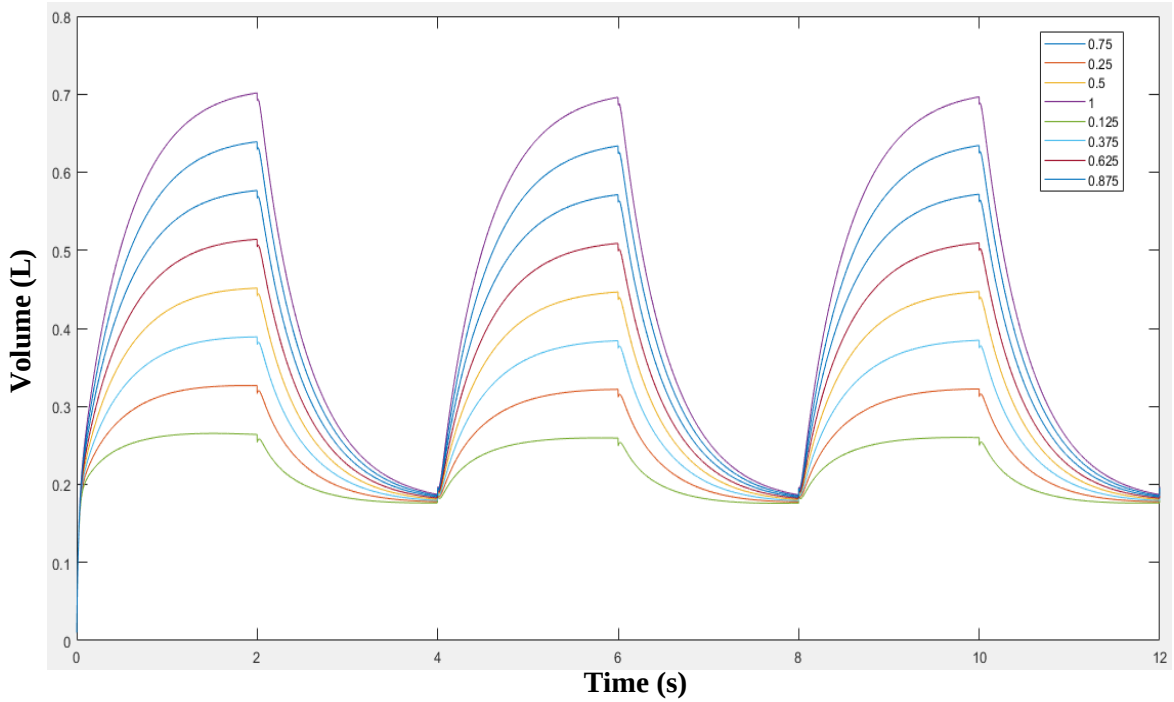


Figure 7 Parametric sweep results for the gain block of value 0.75 in Figure 5.

Moreover, the values of the gain block associated with the step input (disturbance) were changed from 0.5 to 5 (Figure 8), with a constant increase of 0.5 each time. Altering such parameter resulted in a vertical shift for $t > 0$ as shown. The larger the value of such parameter, the greater the shift will be. Such shifts are a result of hyperinflation of the lungs during an asthma attack.

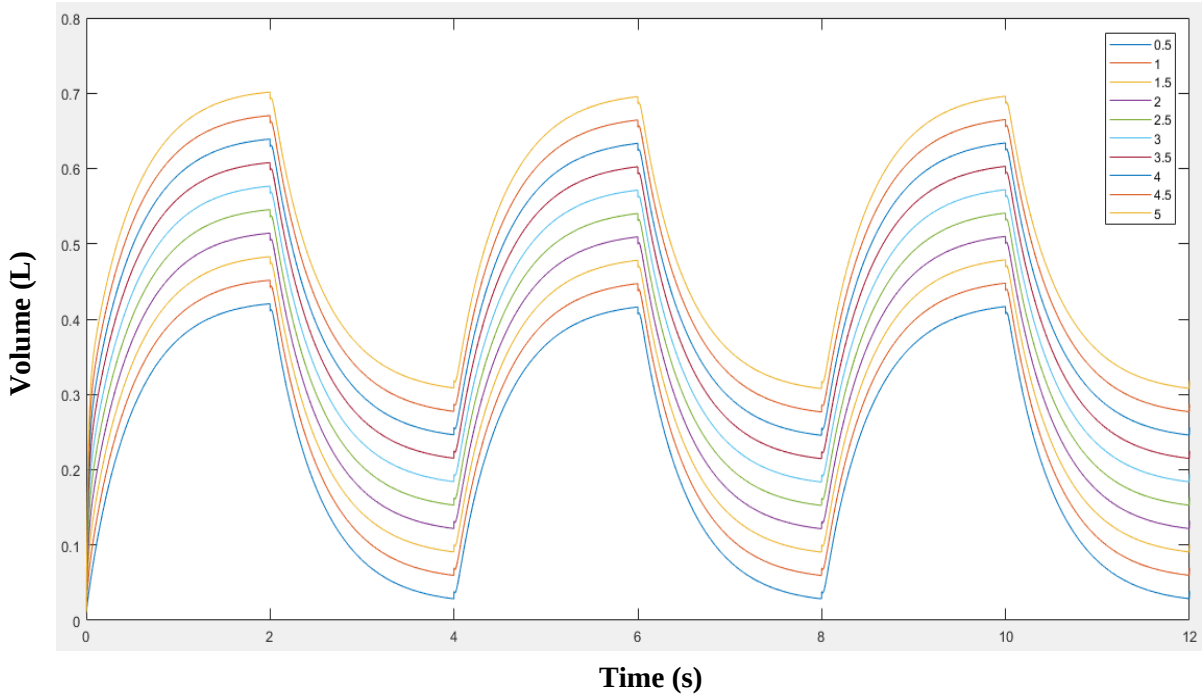


Figure 8 Parametric sweep results for the gain block of value 3 in Figure 5.

2.6 Constructing a disturbance input of asthma medication

The current medications for acute asthma symptoms are beta2-adrenoceptor agonists (e.g. Salbutamol) that are delivered into the respiratory system via a metered-dose inhaler. Beta2-adrenoceptor agonists act on the beta2-adrenergic receptor in the airway and cause smooth muscle relaxation and dilation. The activated receptors are coupled with a stimulatory G protein of adenylyl cyclase, which produces cAMP. cAMP decreases calcium concentration in smooth muscle cells, thus inactivate myosin light-chain kinase and active myosin light-chain phosphatase. These cause smooth muscle cells to relax [12].

We used trial and error method to construct the drug input based on data from the model proposed by Wang et al (Figure 9). In the model, the agonist influences the calcium (Ca2+), myosin light chain kinase (MLCK), and myosin light chain phosphatase (MLCP) level in the airway smooth muscle cells, thus regulates airway contraction and its area [14].

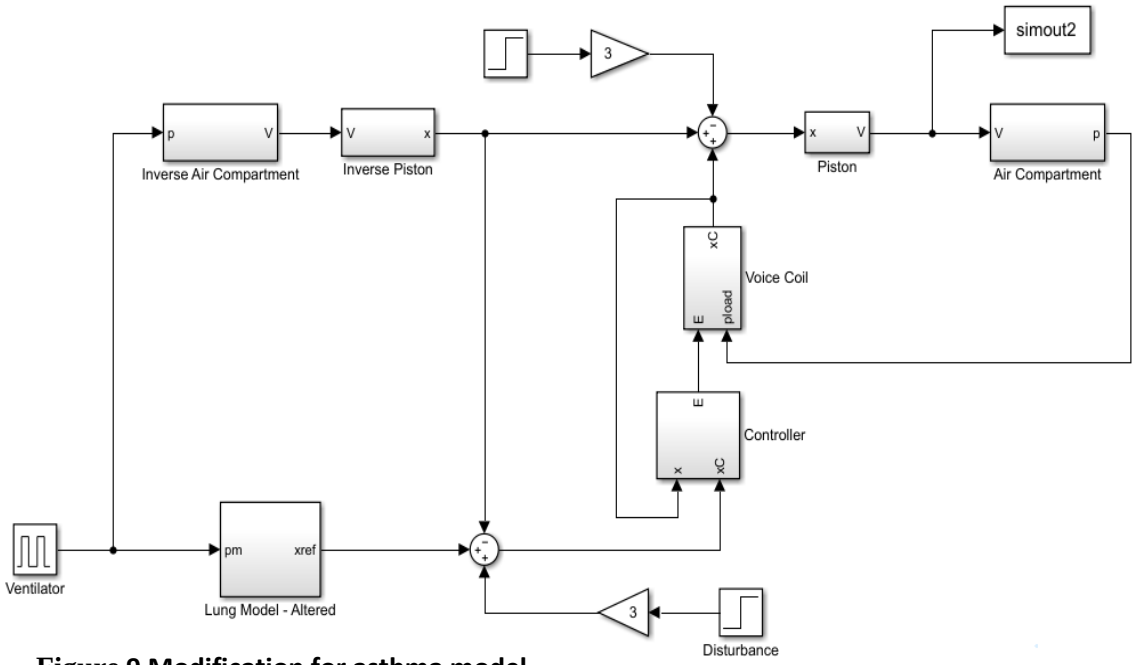


Figure 9 Modification for asthma model.

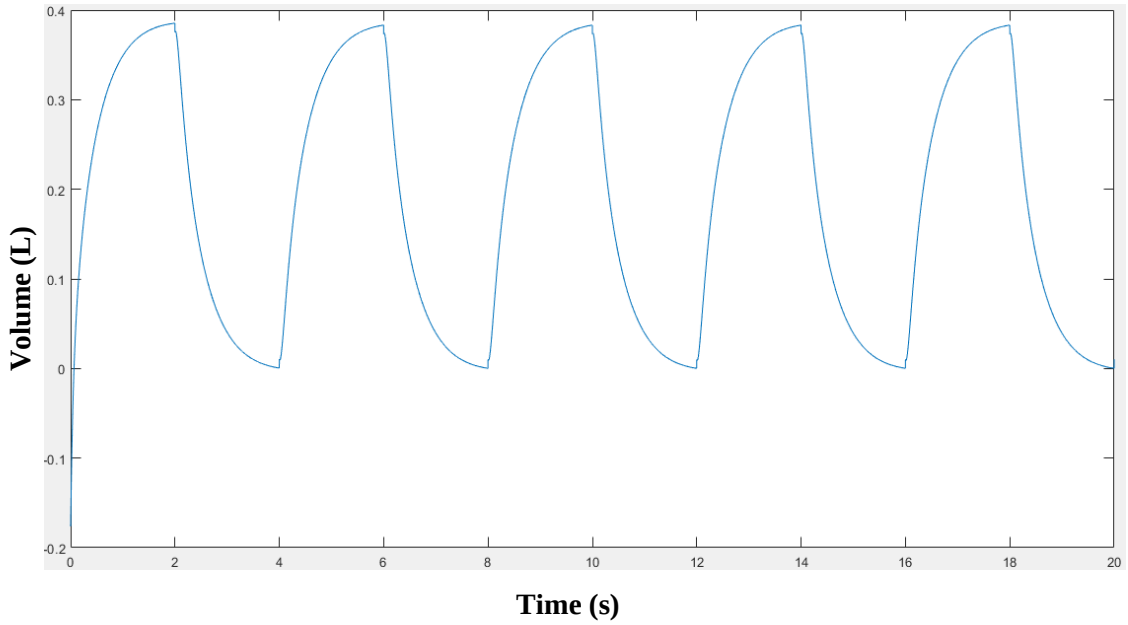


Figure 10 Graphical output of modified model for asthma.

A comparative fit of the model simulations to experimental data from Wang et al predicts that an increase of contractile agonist concentration from 0 to 6 will decrease the calcium concentration in smooth muscle cells from 0.65 μM to 0.1 μM after 200 seconds, and inhibits

MLCP. During agonist presence, the level of myosin that is attached to actin (denoted by AM) also drops to the minimum level (smooth muscle cell contracts while myosin attaches to actin). The relative airway area from time zero of medication injection to time 400 seconds can be approximated by the Equation $A = e^{\frac{t}{4000}}$ [14]. The expansion of airway area inversely proportional airway pressure, which is approximated by adding a step input with the associated gain at the summing point after the voice coil. In addition, a constant gain was added after the piston (Figure 11).

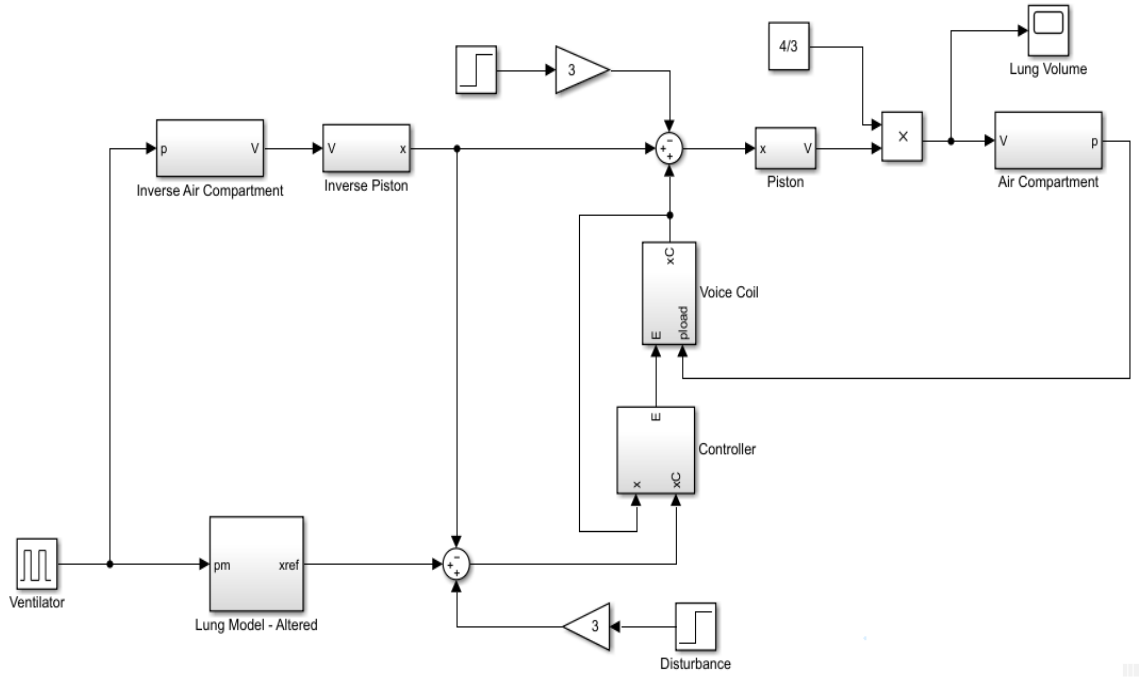


Figure 11 Adding a gain to the output of the Piston to eliminate the Lung Model's gain effect.

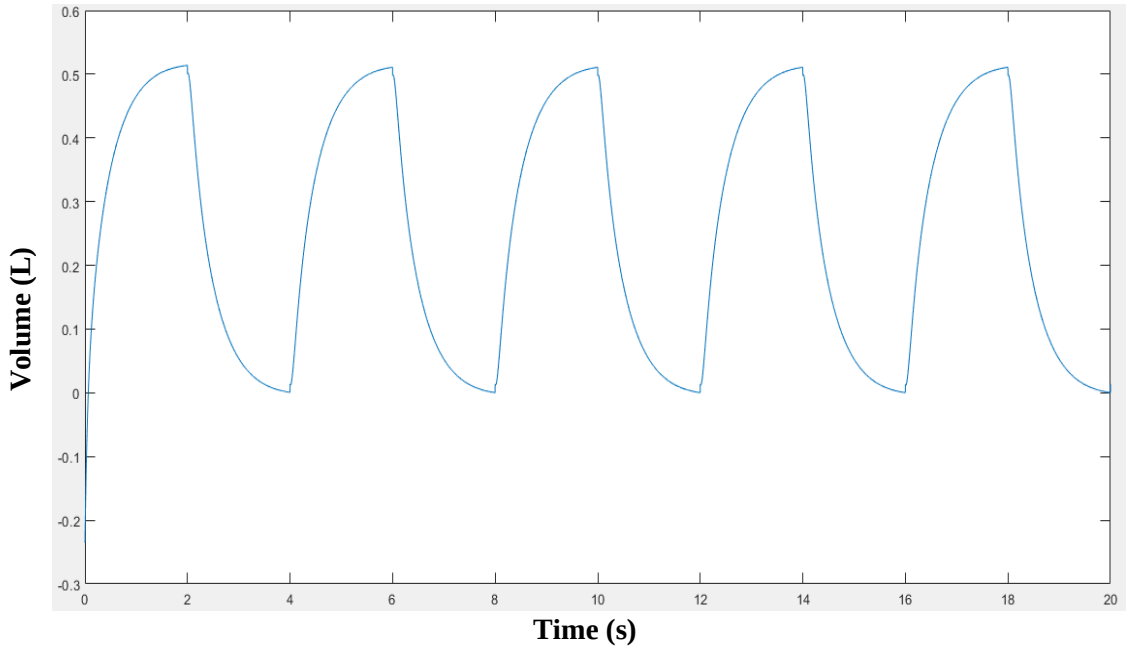


Figure 12 The modified output for the model in Figure 11.

3. Results & Discussion

3.1. Controller Selection and model Verification

Phase and gain margins were acquired for the closed loop systems containing the "Voice Coil" and each controllers (P, PI and PID).

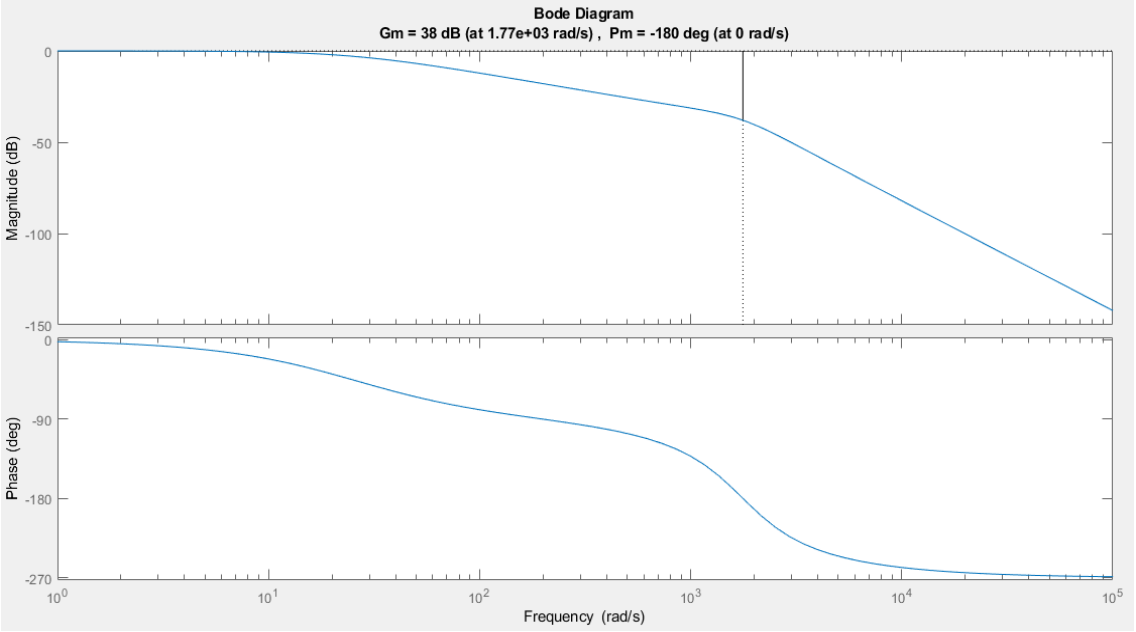


Figure 13: Bode diagram for the closed loop containing P.

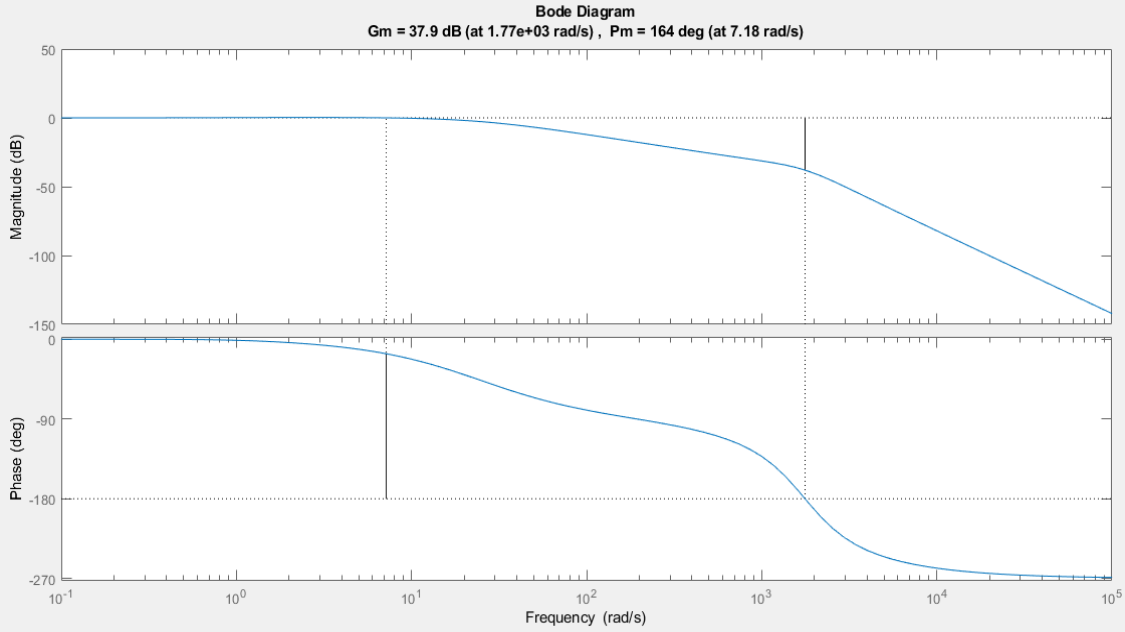


Figure 14: Bode diagram for the closed loop containing PI.

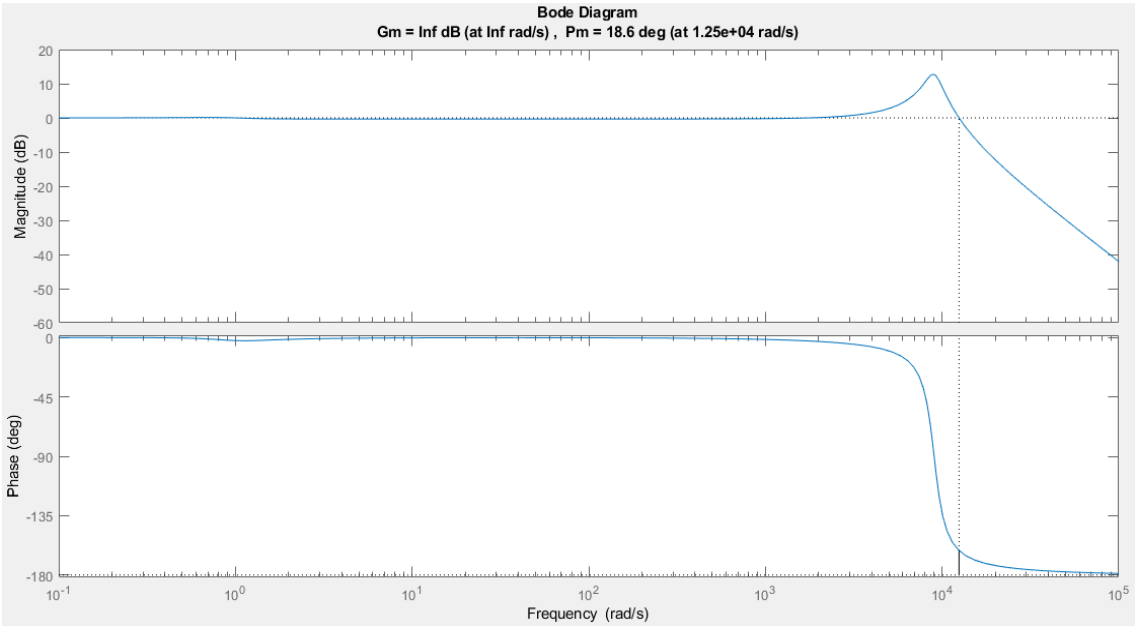


Figure 15: Bode diagram for the closed loop containing PID.

Although the system with the P controller possessed a reasonable gain margin, 38 dB, it had a negative phase margin (-180°); hence, such choice was eliminated. When comparing the system containing the PI controller with that containing the PID controller, it was observed that the latter system's gain margin is infinite and has a more reasonable value of phase margin (18.6°). In addition, the Bode phase plot of PID controller never touches -180° (the phase Bode plot of the system with the PI controller reaches -270°). Therefore, the PID controller was determined to be a more suitable choice for the system, as it yields the most stable system [11].

When plotting the proposed model containing the PID controller, on the same axes as the literature model derived from Nun's book, the following results can be examined, where the red curve denotes the literature model, and the blue curve denotes the proposed model with the PID controller. The two models are approximately the same as the root mean square error was insignificant (0.0508).

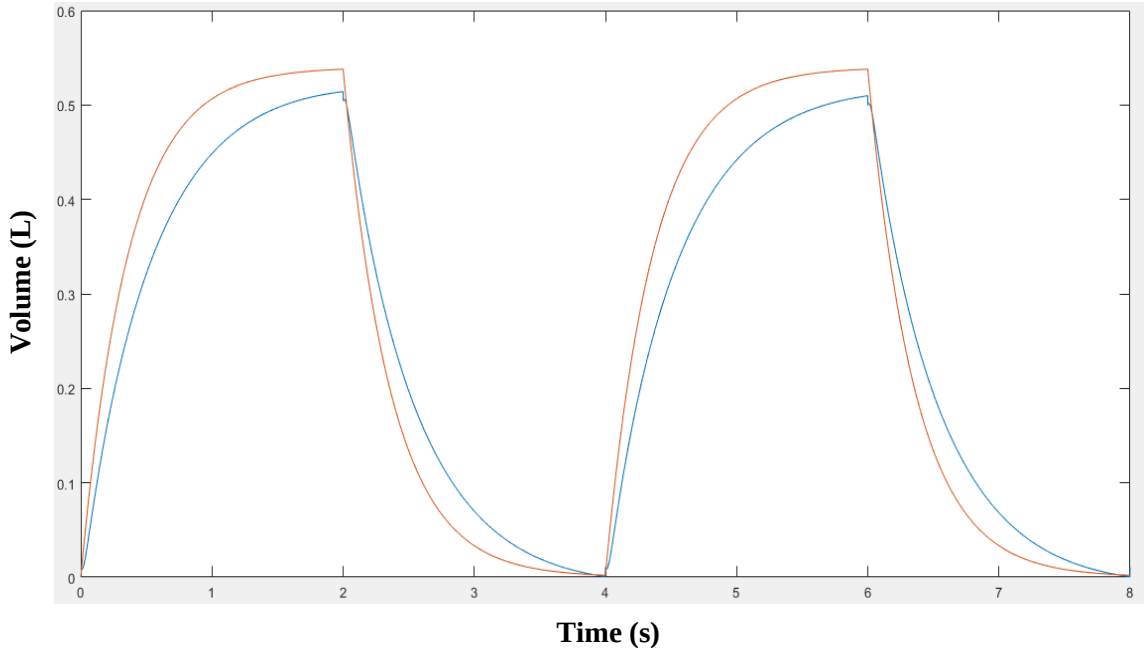


Figure 16 Proposed model (blue) VS literature model (red).

3.2. Asthma model behavior

We were unable to obtain the literature data points of lung volume for an asthma patient, a spirogram denoting lung volume change of patients suffering from an obstructive disease (e.g. asthma), was used to validate the results of the asthma model.

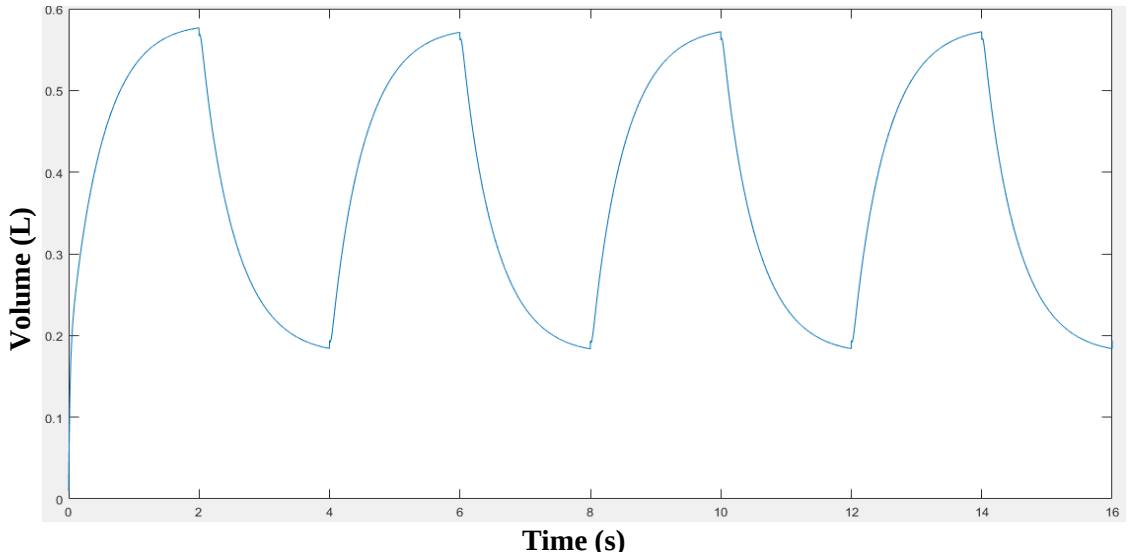


Figure 17: Graphical output of asthma model.

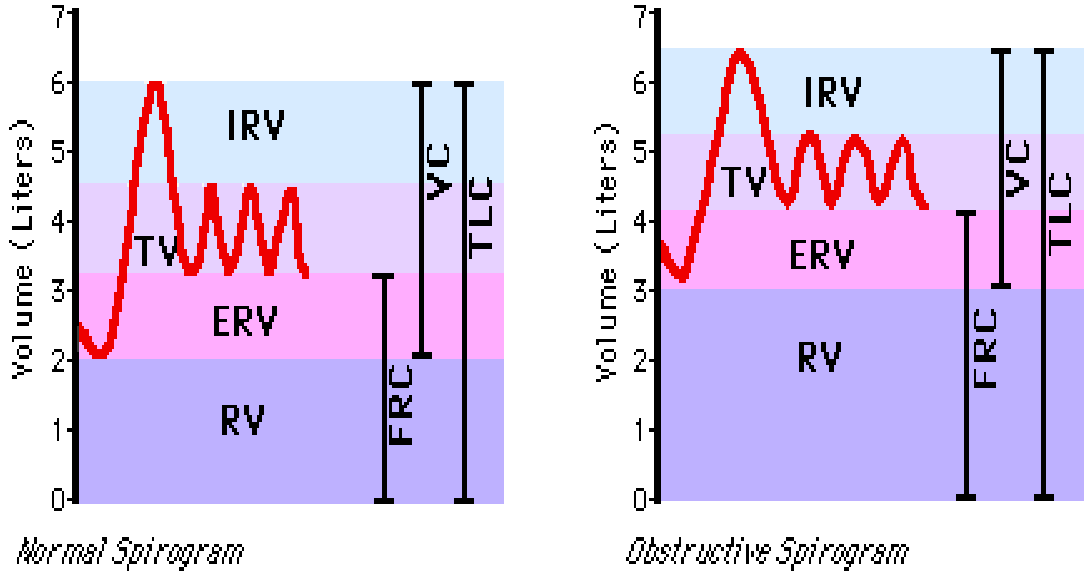


Figure 18: Spirogram for the lung volume of a normal patient (left), and a patient suffering from an obstructive lung disease (right)

When comparing Figure 17 with the rightmost portion of Figure 18, it can be observed that both plots behave similarly. Because of the perturbation input in the asthma model, a vertical shift occurred for $t > 0$, and the gain with value between 0 and 1 incorporated to that model resulted in a decrease in amplitude of the output signal. Such observations can also be observed in the "Obstructive Spirogram" of Figure 16 [4].

3.3. Validation the proposed medication model

The graphical output of the medication model was compared to the outputs of both the literature model, and the proposed model containing the PID controller.

To assess the medication model, a root mean square error (RMSE) test was conducted, such that an average RMSE value was calculated for two cases; the first when the output of the medication model was compared to the output of the proposed model, and the second when the output of the medication model was compared to the output of the literature model. For each case, the simulation was run for 100 seconds, such that the average RMSE value was calculated for 25 successive periods (the period of each output is 4 seconds). The RMSE value from t = 0 to t = n was approximated as the average of the RMSE value of each successive period lying between 0 and n seconds. For example:-

$$RMSE_{T_1} = RMSE_{0\ to\ 4\,\mathrm{sec}}$$

$$RMSE_{0\ to\ 8\ sec} = \frac{RMSE_{T_1} + RMSE_{T_2}}{2}$$

$$RMSE_{T_2} = 2(RMSE_{0\ to\ 8\,\mathrm{sec}}) - RMSE_{T_1}$$

$$RMSE_{0\ to\ 12\ sec} = \frac{RMSE_{T_1} + RMSE_{T_2} + RMSE_{T_3}}{3}$$

$$RMSE_{T_3} = 3(RMSE_{0\ to\ 12\,\mathrm{sec}}) - (RMSE_{T_1} + RMSE_{T_2})$$

And so on up to 25 periods for each of the two cases.

Such calculations led us to obtaining the following tables:

Table 1: RMSE values obtained for simulation times (starting at zero in each case and linearly increasing by a constant of 4 seconds in each preceding case).

Time	Model		Time	Model	
	Proposed model	Literature model		Proposed model	Literature model
	RMSE	RMSE		RMSE	RMSE
0 - 4 sec	0.0243	0.0495	**0 - 56 sec**	0.0078	0.0476
0 - 8 sec	0.0175	0.0485	**0 - 60 sec**	0.0076	0.0476
0 - 12 sec	0.0145	0.0482	**0 - 64 sec**	0.0074	0.0476
0 - 16 sec	0.0127	0.048	**0 - 68 sec**	0.0073	0.0476
0 - 20 sec	0.0116	0.0479	**0 - 72 sec**	0.0071	0.0476
0 - 24 sec	0.0107	0.0479	**0 - 76 sec**	0.0070	0.0476
0 - 28 sec	0.0101	0.0478	**0 - 80 sec**	0.0069	0.0476
0 - 32 sec	0.0095	0.0477	**0 - 84 sec**	0.0068	0.0476
0 - 36 sec	0.0091	0.0477	**0 - 88 sec**	0.0067	0.0476
0 - 40 sec	0.0087	0.0477	**0 - 92 sec**	0.0066	0.0475
0 - 44 sec	0.0084	0.0477	**0 - 96 sec**	0.0066	0.0475
0 - 48 sec	0.0082	0.0476	**0 - 100 sec**	0.0065	0.0475
0 - 52 sec	0.0080	0.0476			

Table 2: RMSE values obtained for each of the 25 periods separately, using the method previously outlined, the average of such RMSE values was calculated.

Time	Model		Time	Model	
	Proposed model	Literature model		Proposed model	Literature model
	RMSE	RMSE		RMSE	RMSE
0 - 4 sec	0.0243	0.0495	**52 - 56 sec**	0.0052	0.0476
4 - 8 sec	0.0107	0.0475	**56 - 60 sec**	0.0048	0.0476
8 - 12 sec	0.0085	0.0476	**60 - 64 sec**	0.0044	0.0476
12 - 16 sec	0.0073	0.0474	**64 - 68 sec**	0.0057	0.0476
16 - 20 sec	0.0072	0.0475	**68 - 72 sec**	0.0037	0.0476
20 - 24 sec	0.0062	0.0479	**72 - 76 sec**	0.0052	0.0476
24 - 28 sec	0.0065	0.0472	**76 - 80 sec**	0.0050	0.0476
28 - 32 sec	0.0053	0.047	**80 - 84 sec**	0.0048	0.0476
32 - 36 sec	0.0059	0.0477	**84 - 88 sec**	0.0046	0.0476
36 - 40 sec	0.0051	0.0477	**88 - 92 sec**	0.0044	0.0453
40 - 44 sec	0.0054	0.0477	**92 - 96 sec**	0.0066	0.0475
44 - 48 sec	0.0060	0.0465	**96 - 100 sec**	0.0041	0.0475
48 - 52 sec	0.0056	0.0476	**Average**	**0.0065**	**0.0475**

The average RMSE value obtained when comparing the output of the medication model with that of the proposed model was 0.0065, and the average RMSE value obtained when comparing the output of the medication model with that of the literature model was 0.0475. Both values are low, indicating that the medication model can effectively model the behavior of a human respiratory system suffering from asthma, when a medication is taken.

Conclusion

An existing Simulink model for a respiratory system has been validated and modified to model the effect of a disturbance representing an asthma attack, and model the behaviour when an input representing medication is incorporated. However, the main challenge we faced was the lack of sufficient literature models or data. We were able to qualitatively validate our asthma model, through comparison with a spirogram, representing the lungs of a patient suffering from an obstructive lung disease. Parameter sweeps enabled us to approximate reasonable values for the gains associated with the asthma model. Moreover, the root mean square error test, previously outlined, enabled us to assess our medication model.

Appendix A: Model for Estimating the Transfer Function of the Inverse Air Compartment

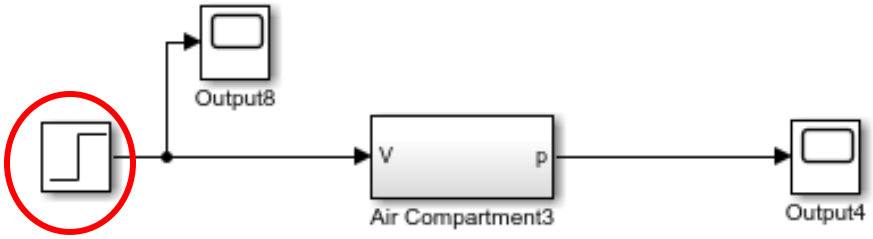
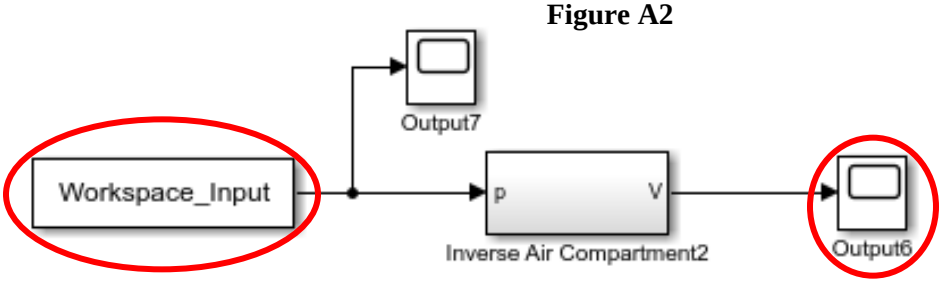


Figure A1

The input is a step input with amplitude 1 and no time delays

Figure A2



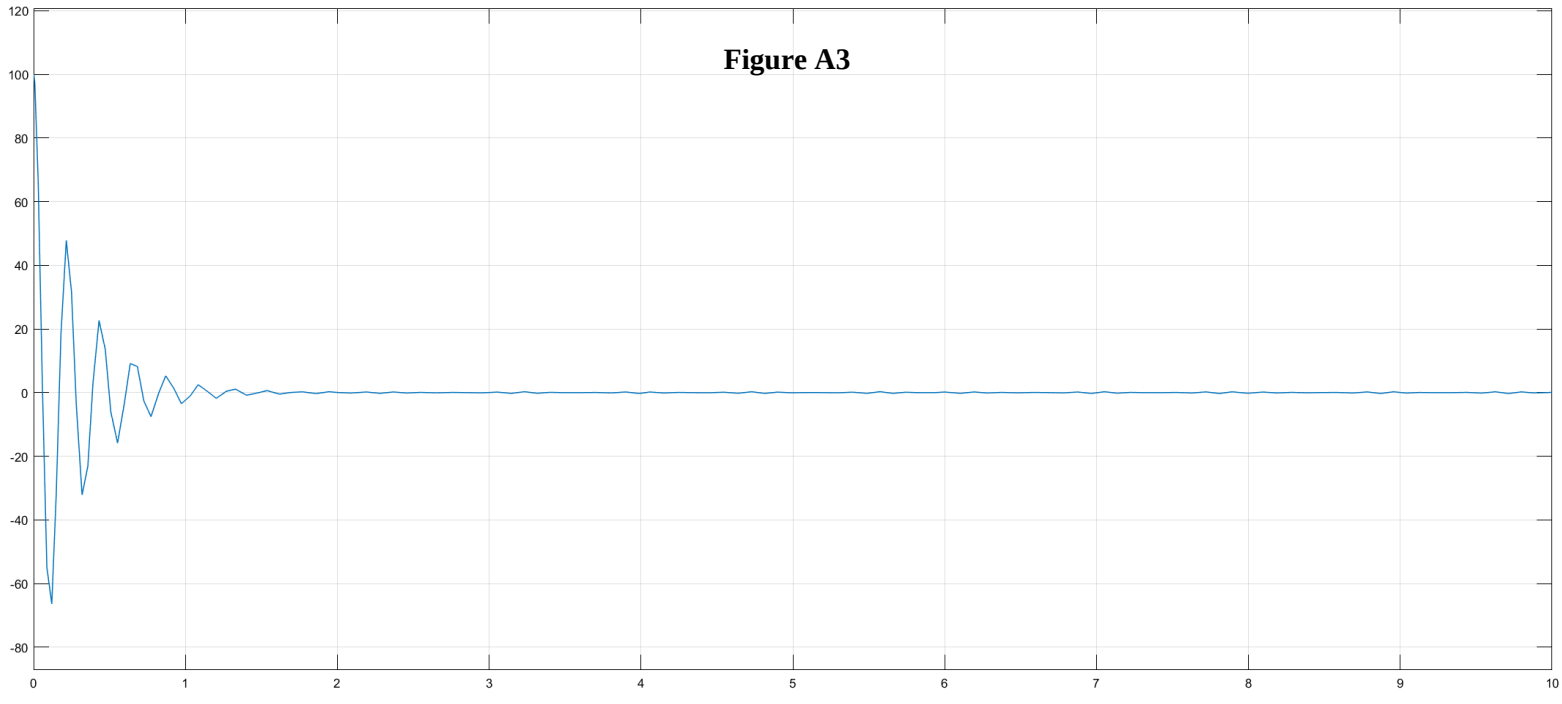
Output6 with Output8 were compared

The analytical function associated is

$100exp\left(-\frac{10t}{3}\right)\left(cos\left(\frac{10\sqrt{74}t}{3}\right)+\frac{\sqrt{74}sin\left(\frac{10\sqrt{74}t}{3}\right)}{74}\right)$, which is the analytical function associated with Output4

The is the graph associated with both Output4 and "Workspace_Input".

Figure A3



This is the graph associated with Output6:-

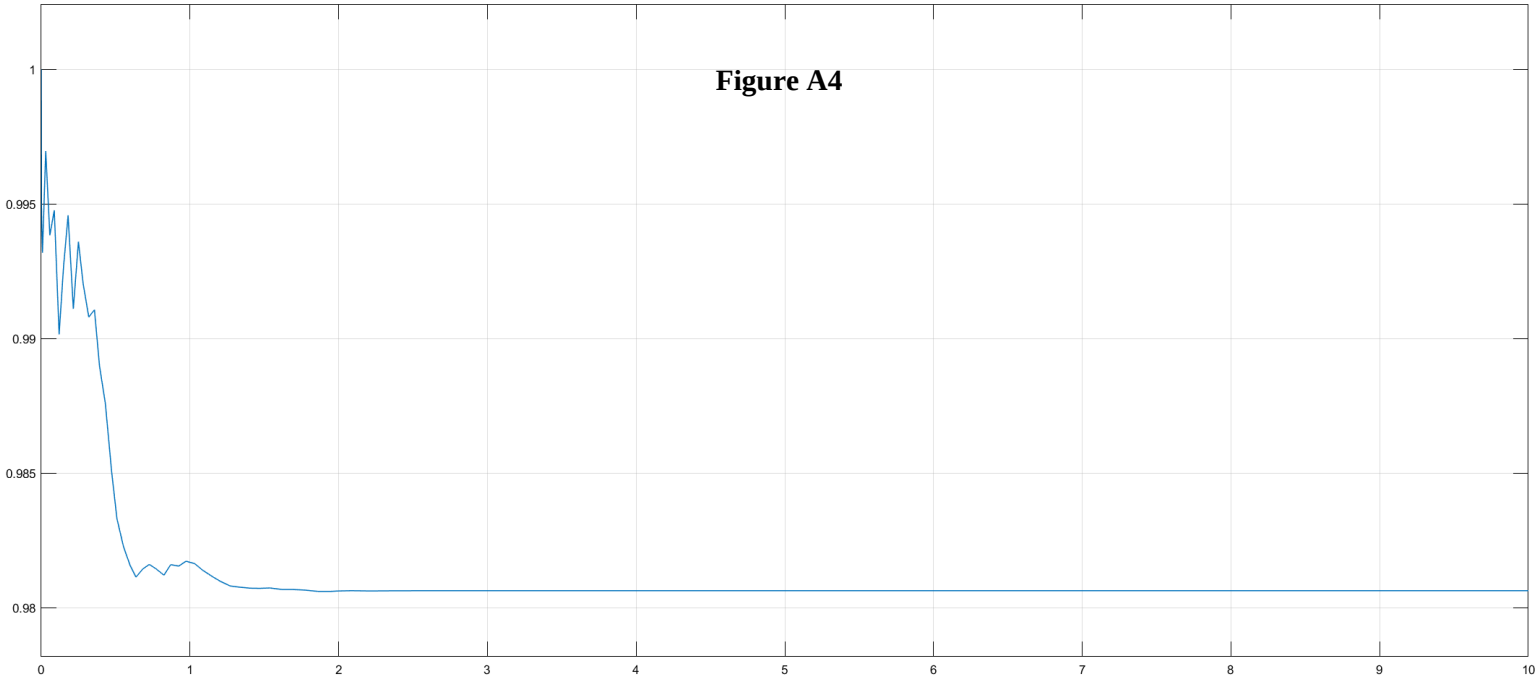


Figure A4

This is the graph obtained when plotting both Output6 and Output8 on the same plot; the blue curve denotes Output8, and the orange curve denotes Output6. Notice that the two curves look very similar to each other.

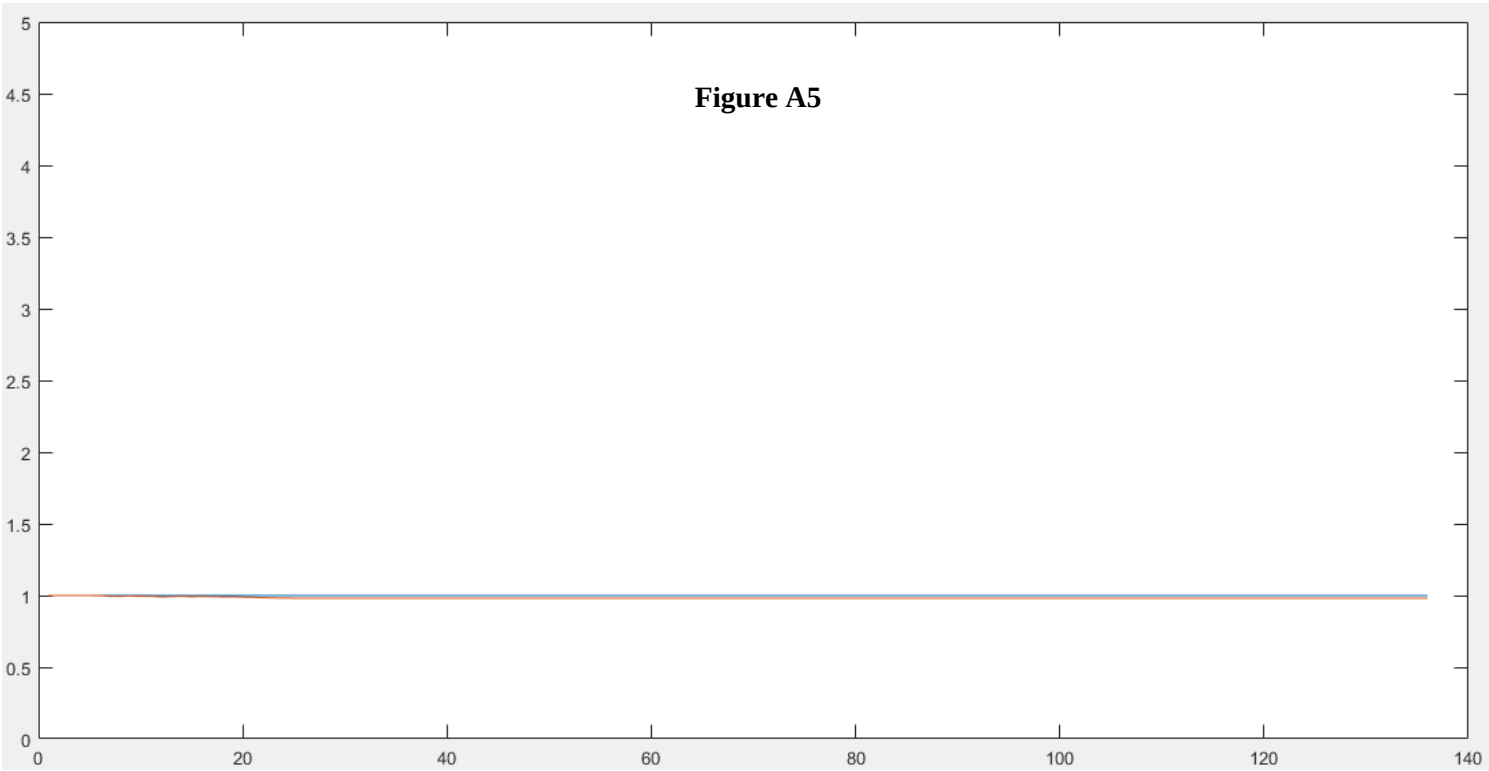


Figure A5

Appendix B: Ziegler-Nichols Calculations for PID Controller Tuning

Model 1: P Controller:-

The associated transfer function in this case (derived from a unity feedback block diagram with gain K_U) is as follows:-

$$T(s) = \frac{100K_U}{0.0032s^3 + 6.4s^2 + 10000s + 100K_U}$$

Where the characteristic equation is:-

$$q(s) = 0.0032s^3 + 6.4s^2 + 10000s + 100K_U$$

The Routh array constructed is:-

$\mathbf{s^3}$	0.0032	10000
$\mathbf{s^2}$	6.4	$100K_U$
$\mathbf{s^1}$	x	0
$\mathbf{s^0}$	$100K_U$	0

$$x = \frac{(6.4 \times 10000) - (0.0032 \times 100K_U)}{6.4} = 0$$

Rearranging yields $K_U = 200000$

$K_P = 0.5K_U = 100000$

Model 2: PI Controller:-

From the calculations above, the characteristic equation can be written as:-

$$q(s) = 0.0032s^3 + 6.4s^2 + 10000s + 20000000$$

Such that K_I is set to 0.

To find the roots of q(s), the following MATLAB code is used:-

```
charac_eq = [0.0032 6.4 10000 20000000]
roots(charac_eq)
```

The roots were:-

$s_1 = -2000$, $s_2 = 1.7678j$, $s_3 = -1.7678j$

Frequency can be calculated as f = 1.7678 rad/ sec = 0.28135410872 Hz

Period can be calculated as

$T_U = \frac{1}{f} = 3.5542399027$

$K_P = 0.45 \times 200000 = 90000$

$K_I = \frac{0.54 \times 200000}{T_U} = 30386.24375$

Model 3: PID Controller:-

Using the previous calculations:-

$K_P = 0.6K_U = 120000$

$K_I = \frac{1.2K_U}{T_U} = 67524.98609$

$K_D = \frac{0.6K_U T_U}{8} = 5313.59855$

[13]

Appendix C: MATLAB Code

The following code was utilized to examine the input of Figure A1 and output of Figure A2 from Appendix A.

```
syms s;    % to create a symbolic function

% The output of the first model in this simulation is denoted as
% Y(s) = U(s)T(s), where U(s) is the input, and T(s) is the transfer
% function
Y = (1/s)*(((100*s^2) + (2000/3)*s)/(s^2 + (60/9)*s + (2500/3)));
disp(Y);

% To find the inverse Laplace (which gives the analytical function)
y = ilaplace(Y);
disp(y);

T0=0;    % starting time
Tf=10;   % ending time

fs=100;  % sampling frequency
dt=1/fs; % step size

t=(T0:dt:Tf)';

% To make the analytical function derived as the input for the second model
Workspace_Input = [t 100.*exp(-(10.*t)/3).*(cos((10.*74^(1/2).*t)/3) +
(74^(1/2).*sin((10*74^(1/2).*t)/3))/74)];
sim('test_inv_AC.slx');
```

To validate our findings from the code above and Appendix A, the following code was used:-

```
plot(yout); % A "To Workspace" block called "yout" was added to the input
            % of Figure A1 to obtain graphical results in MATLAB environment

hold on; % To plot both curves on the same graph

plot(yout1); % A "To Workspace" block called "yout1" was added to the output
             % of Figure A2 to obtain graphical results in MATLAB environment

ylim([0 5]);  % To set the limits of the y-axis (from 0 to 5)

disp(rms(yout - yout1));  % the root mean square error value here is 0.0179
```

The following is the code used to find the value of K_P corresponding to the lowest root mean square error value for the model in Figure 1 such that a P controller is used.

```
Gain = [1000:2500];   % Range of gain values provided in thesis

for n = 1:length(Gain)    % There is a block in the corresponding simulink
model called 'Gain'
    K = Gain(n);          % K can take any integer value from 1000 to 2500
    sim('find_optimal_Kp.slx');    % To initiate the simulation

    % Formula for root mean square error (the functions rms cannot be used
    % here because the graphical outputs from the simulation are time
    % series
    RMSE = sqrt(mean((simout - simout1)*(simout - simout1)));

    % Concerned with the minimum root mean square error value
    min_RMSE = min(RMSE);
 end

disp(min_RMSE);

% To display the value of K that has the lowest root mean square error value
disp(K);
```

The following code was used to find the values of K_P and K_I that together correspond to the lowest root mean square error value for the model in Figure 1, if a PI controller is utilized. Eventually, this code kept running for multiple hours without obtaining the desired values of gain; hence, the values of K_P and K_I were not obtained.

```
% There are two gain blocks; one associated with the proportional part, and
% the other with the integral part of the PI controller
Gain = [1000:2500];
Gain1 = [1000:2500];

for n = 1:length(Gain)
    for m = 1:length(Gain1)
        Kp = Gain(n);    % Proportional gain
        Ki = Gain1(m);   % Integral gain
        sim('find_optimal_Kp_Ki.slx');
        RMSE = sqrt(mean((simout - simout1)*(simout - simout1)));
        min_RMSE = min(RMSE);
    end
 end

disp(min_RMSE);
disp(Kp);
disp(Ki);
```

Given each of the two gains in the code above can take any integer values from 1000 to 2500, MATLAB would have to attempt 1500^2 simulations and compare them with one another to obtain the desired values of gain. If we ought to optimize the gain of K_P, K_I, and K_D assuming Figure 1 utilized a PID controller, MATLAB would have to attempt 1500^3 simulations and compare them

with one another to obtain the desired values of gain. Hence, it was estimated that $K_P = K_I$ for the case of the PI controller, and $K_P = K_I = K_D$ for the case of the PID controller.

The following code was used to find the value of gain K corresponding to the lowest root mean square error value, for the model in Figure 1, assuming the controller used was a PID controller with equal proportional, integral, and derivative gains.

```
for n = 1:length(Gain)
    K = Gain(n);
    sim('find_optimal_K_PID.slx');
    RMSE = sqrt(mean((simout - simout1)*(simout - simout1)));
    min_RMSE = min(RMSE);
 end

disp(min_RMSE);
disp(K);
```

The same logic was utilized to optimize for gain K of Figure 1, assuming a PI controller was used.

The following is a sample code snippet used to find the root mean square error value for a certain time period:-

```
set_param('test_stoptime','StartTime','0','StopTime','20');
% test_stoptime is the name of the target Simulink file
% StartTime is the time at which the simulation should start (0 sec in this
case)
% StopTime is the time at which the simulation should stop (20 sec in this
case)

figure(1);

% To plot both curves on the same graph
plot(simout1);
hold on;
plot(simout);

% To find root mean square error
RMSE = sqrt(mean((simout - simout1)*(simout - simout1)));
disp(RMSE);
```

The following code was utilized to generate the Bode plots with phase and gain margins for the P, PI, and PID controllers incorporated in a closed loop system with the “Voice Coil”:-

```
Kf = 100; % Electromagnetic force constant
Kg = Kf; % Electromotive force constant
L = 0.002; % Inductance of coil
m = 1.6; % Mass of coil
R = 4; % Resistance of coil
b = 0; % Coefficient of friction
```

```
% Controller gains or functions:-
K_p = 2500;   % For P controller
K_pi = 2500*tf(1, [1 0]) + 2500;  % For PI controller
K_pid = 2500 + 2500*tf(1, [1 0]) + 2500*tf([1 0], [0 1]); % For PID
controller

% Transfer function of "Voice Coil":-
voice_coil_tf = tf(Kf, [(L*m) (R*m + L*b) (R*b + Kf*Kg) 0]);

% Bode plots with gain and phase margins:-

% To display Bode plot
figure(1);
bode((voice_coil_tf * K_p)/ (1 + voice_coil_tf*K_p));
title('Closed loop Bode plot for voice coil with P controller');
% To display gain and phase margin on Bode plot
figure(2);
margin((voice_coil_tf * K_p)/ (1 + voice_coil_tf*K_p));

figure(3);
bode((voice_coil_tf * K_pi)/(1 + voice_coil_tf*K_pi));
title('Closed loop Bode plot for voice coil with PI controller');
figure(4);
margin((voice_coil_tf * K_pi)/(1 + voice_coil_tf*K_pi));

figure(5);
bode((voice_coil_tf * K_pid)/(1 + voice_coil_tf*K_pid));
title('Closed loop Bode plot for voice coil with PID controller');
figure(6);
margin((voice_coil_tf * K_pid)/(1 + voice_coil_tf*K_pid));
```

Appendix D: References

[1]: "Asthma | AAAAI", *The American Academy of Allergy, Asthma & Immunology*, 2017. [Online]. Available: http://www.aaaai.org/conditions-and-treatments/asthma. [Accessed: 17- Oct- 2017].

[2]: "Asthma", *Oac.med.jhmi.edu*, 2017. [Online]. Available: http://oac.med.jhmi.edu/res_phys/Encyclopedia/Asthma/Asthma.HTML. [Accessed: 17- Oct- 2017].

[3]: "Asthma, 2014", *Statcan.gc.ca*, 2017. [Online]. Available: https://www.statcan.gc.ca/pub/82-625-x/2015001/article/14179-eng.htm. [Accessed: 17- Oct- 2017].

[4]: "Occupational Medicine", Iaff.org, 2017. [Online]. Available: http://www.iaff.org/hs/Resi/pulmonary%20function%20tests.htm. [Accessed: 17- Oct- 2017].

[5]: "RMS Error", Statweb.stanford.edu, 2017. [Online]. Available: http://statweb.stanford.edu/~susan/courses/s60/split/node60.html. [Accessed: 01- Dec- 2017].

[6]: "What are Hyperinflated Lungs and What Causes It? | Inogen Blog", Inogen, 2017. [Online]. Available: https://www.inogen.com/blog/what-are-hyperinflated-lungs-and-what-causes-it/. [Accessed: 01- Dec- 2017].

[7]: "What Is Asthma? - NHLBI, NIH", Nhlbi.nih.gov, 2017. [Online]. Available: https://www.nhlbi.nih.gov/health/health-topics/topics/asthma/. [Accessed: 01- Dec- 2017].

[8]: (1997). Système International: Examples of Conversions Commonly Used in Respiratory Physiology and Respiratory Care, (30), 639–640.

[9]: A. Lumb and J. Nunn, Nunn's Applied Respiratory Physiology. Edinburgh: Churchill Livingstone/Elsevier, 2010.

[10]: Delawari, Anton; Doelman, R. (2010). Simulation of an Artificial Respiratory System. Lloydia (Cincinnati), 72.

[11]: Mysore, Shreesh. (2002). “What is a negative phase margin; what does it mean?”. [Online]. Available: http://www.cds.caltech.edu/~murray/courses/cds101/fa02/faq/02-11-18_negativepm.html. [Accessed: 01- Dec- 2017].

[12]: National Asthma Education and Prevention Program (2007). "Expert Panel Report 3: Guidelines for the Diagnosis and Management of Asthma". National Heart Lung and Blood Institute.

[13]: R. Dorf and R. Bishop, Modern control systems. Harlow, England: Pearson, 2017.

[14]: Wang, Inga et al. “A Mathematical Model of Airway and Pulmonary Arteriole Smooth Muscle.” Biophysical Journal 94.6 (2008): 2053–2064. PMC. Web. 30 Nov. 2017.