Development of Control System for Open-Source Low-Cost Ventilators

Heena Shrestha, The University of Western Ontario

Supervisor: Trejos, Ana L., The University of Western Ontario
Co-Supervisor: DeGroot, Christopher, The University of Western Ontario

A thesis submitted in partial fulfillment of the requirements for the Master of Engineering Science degree in Electrical and Computer Engineering
© Heena Shrestha 2021

Follow this and additional works at: https://ir.lib.uwo.ca/etd

Recommended Citation
https://ir.lib.uwo.ca/etd/8340

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlsadmin@uwo.ca.
Abstract

Respiratory illness and diseases are ranked five out of the thirty most common causes of death worldwide. Thus, the use of a ventilator becomes a must for assisting patients suffering from respiratory disorders. The COVID-19 outbreak has aggravated this situation. Difficulty breathing is one of the most common symptoms of COVID-19. Out of 20% symptomatic COVID-19 patients that require hospitalization, about 5% ended up in the Intensive Care Unit (ICU), most of them requiring ventilation. The world saw a scarcity of ventilators when COVID-19 was at its first peak, with manually ventilating patients using an AMBU bag being the only option left. The lack of ventilators was a struggle for every country around the world. One of the leading factors for the shortage of ventilators was due to their cost. This situation affected developing and underdeveloped countries the most.

The objective of this entire project is to develop a low-cost emergency use ventilator device that can be readily mass-produced and used during an emergency such as COVID-19. The main work done in this thesis is towards the development of the controller for the project. This thesis describes the design, development, data collection, analysis, and validation of the controller used for a low-cost mechanical ventilator device designed using a bag valve mask (BVM) and a microcontroller.

A list of requirements and specifications needed for the design of an emergency respirator was created after consulting with expert respiratory therapists and clinicians. A first prototype was built based on the first version of the mechanical ventilator developed at Robarts Research Institute. The prototype was tested for different settings of Tidal Volume ($V_t$), Breaths per Minute (BPM), and Inspiration: Expiration ratio (I:E). A safety alarm and a flow sensor were added to the existing prototype for real-time data collection and to check the accuracy of the device. The device monitors the inspiratory and expiratory pressure, $V_t$, BPM, I:E ratio, cycle (for the motor) and peak pressure.

The work done in this thesis improves the functionality (output tidal volume can be determined) and safety of the system. A low-cost ventilator device has a potential to be produced into a commercialized regulatory approved version so that it can be easily accessible to medical centers all around the world when an outbreak such as COVID-19 occurs again. However, more
research is necessary in order to add more functionality such as multiple modes of ventilation, measurement of plateau pressure, monitoring and measurement of fraction of inspired oxygen (FiO₂) and improving the specificity of the ventilation model.

Keywords: Mechanical Ventilation, Respiratory Diseases, COVID-19, Arduino Due
Lay Abstract

Respiratory Diseases are most common causes of death worldwide. In extreme cases, the use of ventilators becomes a necessity for assisting patients suffering from respiratory diseases. Mechanical ventilators are devices that assist a patient to breathe when they are having difficulties breathing on their own. In the recent times, COVID-19 outbreak caused scarcity of the ventilators in many developing and under-developed countries. One of the most common symptoms of COVID-19 is difficulty in breathing. Out of 20% COVID-19 patients, about 5% end up in ICU and most of them require ventilation.

The objective of this project as a whole is to develop a low-cost ventilator device. A low-cost ventilator is made using free licensed design and freely available components that are generally not expensive. These types of ventilators can be easily mass-produced since they are low-cost devices and used during emergency situations. This thesis describes the development of the controller of low-cost ventilators using a mask that is connected to squeezable bag to help the patient breath in and breath out. The squeezing of the bag is controlled by a simplified computer device. Several experiments were conducted to check the safety and the operation of the device.
Acknowledgements

I would like to express my appreciation for the guidance provided by my supervisors, Dr. Ana Luisa Trejos and Dr. Christopher DeGroot. Their support and encouragement throughout the past one year have motivated me to acquire the skills and knowledge to complete this thesis. They have been a constant source of inspiration to me.

I am deeply indebted to Dr. David Holdsworth, who provided me with insight and expertise towards this project and assisted me with testing the device. I was very fortunate to have his support and feedback for data analysis and interpretation. The feedback provided by Dr. Gordon Campbell for the contents of Chapter 4 of the thesis is greatly appreciated.

I would like to express my gratitude to Dr. James Lacefield for his help with useful comments to make the thesis better and for his insights as a thesis examiner. The input provided by my other thesis examiners, Dr. George K. Knopf and Dr. Vijay Prasa, is also greatly appreciated.

I am extremely fortunate to have been a part of the Wearable Biomechatronics Laboratory. I would like to thank all of the colleagues I have worked with over the past years for their assistance, encouragement, expertise, and friendship.

The research in this thesis was funded by the Scotiabank and Western University.
# Table of Contents

Abstract........................................................................................................................................... i
Lay Abstract ................................................................................................................................. iii
Acknowledgements ....................................................................................................................... iv
List of Tables .................................................................................................................................. vii
List of Figures ............................................................................................................................... viii
Nomenclature ............................................................................................................................... ix

Chapter 1 ....................................................................................................................................... 1

Introduction .................................................................................................................................. 1

1.1 Motivation ............................................................................................................................... 2
1.2 General Problem Statement ................................................................................................. 3
1.3 Research Scope and Objectives ............................................................................................. 4
1.4 Overview of the Thesis ........................................................................................................... 4

Chapter 2 ....................................................................................................................................... 6

Literature Review ......................................................................................................................... 6

2.1 Introduction ............................................................................................................................. 6
2.2 Breathing Anatomy .................................................................................................................. 6

2.2.1 Mechanism of Breathing .................................................................................................. 7
2.2.2 Respiratory Mechanics ..................................................................................................... 8
2.3 Mechanical Ventilators .......................................................................................................... 9

2.3.1. Types of Mechanical Ventilation ............................................................................... 10
2.3.2 Working Mechanism of the Mechanical Ventilator ....................................................... 16
2.3.3 Uses of Mechanical Ventilators ..................................................................................... 19
2.4 Obstructive and Restrictive Lung Disease ............................................................................ 19

2.4.1 COVID-19 and Need for Emergency Ventilators ......................................................... 20
2.4.2 COVID-19 and Mechanical Ventilation ........................................................................ 23
2.5 Low-Cost Open-source Ventilators ..................................................................................... 23

2.5.1 Types of Open-Source Ventilators ................................................................................. 24
2.5.2 Western University Ventilator (WUV) ........................................................................ 28
2.6 Conclusion .............................................................................................................................. 29

Chapter 3 ....................................................................................................................................... 30

Design and Development of the Control System ........................................................................ 30

3.1 Introduction ............................................................................................................................. 30
3.2 Western University Ventilator (WUV) ................................................................................ 30
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3 Design Process</td>
<td>3.3.1 Components of the Existing WUV</td>
<td>34</td>
</tr>
<tr>
<td>3.3 Design Process</td>
<td>3.3.2 Components of the Modified WUV</td>
<td>37</td>
</tr>
<tr>
<td>3.3 Design Process</td>
<td>3.3.3 Control Implementation</td>
<td>39</td>
</tr>
<tr>
<td>3.4 Physical Implementation</td>
<td></td>
<td>42</td>
</tr>
<tr>
<td>Chapter 4</td>
<td></td>
<td>43</td>
</tr>
<tr>
<td>Results and Discussion</td>
<td>4.1 Motivation and Objective of the Experiment</td>
<td>44</td>
</tr>
<tr>
<td>Results and Discussion</td>
<td>4.2 Tidal Volume Test</td>
<td>44</td>
</tr>
<tr>
<td>Results and Discussion</td>
<td>4.2.1 Methods</td>
<td>45</td>
</tr>
<tr>
<td>Results and Discussion</td>
<td>4.2.2 Results</td>
<td>45</td>
</tr>
<tr>
<td>Results and Discussion</td>
<td>4.4 Sensor Validation</td>
<td>49</td>
</tr>
<tr>
<td>Results and Discussion</td>
<td>4.5 Safety Alarm Testing</td>
<td>57</td>
</tr>
<tr>
<td>Chapter 5</td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>Conclusion</td>
<td>5.1 Concluding Remarks</td>
<td>60</td>
</tr>
<tr>
<td>Conclusion</td>
<td>5.2 Contribution</td>
<td>61</td>
</tr>
<tr>
<td>Conclusion</td>
<td>5.3 Future Work</td>
<td>61</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>Appendices</td>
<td></td>
<td>69</td>
</tr>
<tr>
<td>Appendix A</td>
<td>Permissions and Approvals</td>
<td>69</td>
</tr>
<tr>
<td>Appendix B</td>
<td></td>
<td>72</td>
</tr>
<tr>
<td>Appendix B</td>
<td>Section B.1</td>
<td>72</td>
</tr>
<tr>
<td>Appendix B</td>
<td>Section B.2</td>
<td>95</td>
</tr>
<tr>
<td>Appendix B</td>
<td>Section B.3</td>
<td>97</td>
</tr>
<tr>
<td>Curriculum Vitae</td>
<td></td>
<td>110</td>
</tr>
</tbody>
</table>
List of Tables

Table 2.1: Advantages and disadvantages of the volume and pressure control mode. ................................................. 15
Table 2.2: Summary of types of open-source ventilators. ............................................................................................................. 26
Table 3.1: Alert conditions implemented in the control system. ........................................................................................................ 39
Table 4.1: Recorded data for different lung settings .................................................................................................................. 46
Table 4.2: Recorded data for different lung settings .................................................................................................................. 47
Table 4.3: Recorded data for different lung settings .................................................................................................................. 48
Table 4.4: Portion of data from Appendix B, Table 1. .................................................................................................................... 50
Table 4.5: Calculated error data for Input and output seconds per breath .................................................................................. 51
Table 4.6: Calculated error data for Input and output seconds per breath .................................................................................. 52
Table 4.7: Calculated error data for Input and output seconds per breath .................................................................................. 53
Table 4.8: Safety Alarm Test Conditions ........................................................................................................................................ 57
List of Figures

Figure 2.1: Views of thoracic anatomy during breathing ....................................................... 7
Figure 2.2: Components of changes in airway pressure during breathing cycle ........................ 9
Figure 2.3: Diagram of an Iron lung NVP ............................................................................. 11
Figure 2.4: Example waveform of volume-controlled ventilation waveform .......................... 13
Figure 2.5: Example waveform for pressure-controlled ventilation waveform ..................... 14
Figure 2.6: Example waveform to describe PV Relationship .................................................. 17
Figure 2.7: COVID-19 illustration created by Centers for Disease Control and Prevention ...... 21
Figure 2.8: Using a Blower for Ventilation System Design ................................................... 25
Figure 2.9: Using a AMBU bag for ventilation system design ................................................ 26
Figure 2.10: Western University low-cost ventilator device .................................................. 28

Figure 3.1: Western University low-cost ventilator device .................................................... 30
Figure 3.2: Squeezing of the BVM at start position ............................................................... 31
Figure 3.3: Squeezing of the BVM at full excursion ............................................................ 31
Figure 3.4: Bag valve mask .................................................................................................. 32
Figure 3.5: Display screen that shows the user input data .................................................... 33
Figure 3.6: Stepper motor .................................................................................................... 34
Figure 3.7: Stepper motor driver ......................................................................................... 35
Figure 3.8: MAX30105 high-sensitivity optical sensor ........................................................ 35
Figure 3.9: MPS20N001 pressure sensor ............................................................................. 36
Figure 3.10: LCD display ..................................................................................................... 36
Figure 3.11: Arduino due microcontroller .......................................................................... 37
Figure 3.12: SFM 3300-D Flow sensor ................................................................................. 38
Figure 3.13: Piezo Buzzer .................................................................................................... 38
Figure 3.14: Ventilator controller loop ................................................................................ 41
Figure 3.15: Flow diagram of physical implementation ....................................................... 42

Figure 4.1: Test setup with the lung simulator showing the Western University Ventilator ........ 43
Figure 4.2: Input Vt and Output Vt for Test Procedure 1 ....................................................... 47
Figure 4.3: Input Vt and Output Vt for Test Procedure 2 ....................................................... 48
Figure 4.4: Input Vt and Output Vt for Test Procedure 3 ....................................................... 49
Figure 4.5: Pressure and time graph for Test Procedure 1 ................................................... 54
Figure 4.6: Pressure and time graph for Test Procedure 2 ................................................... 55
Figure 4.7: Pressure and time graph for Test Procedure 3 ................................................... 56
Figure 4.8: The LED turns on for Test 1 ............................................................................. 58
Figure 4.9: The LED turns on for Test 2 ............................................................................. 58
Figure 4.10: The LED turns on for Tests 3 and 4 ................................................................. 59
Nomenclature

TB – Tuberculosis

COPD – Chronic Obstructive Pulmonary Disease

DALYs – Disability-adjusted life-years

ICU – Intensive Care Unit

SARS-CoV-2 – Severe acute respiratory syndrome coronavirus 2

COVID-19 – Coronavirus disease 2019

ARDS – Acute Respiratory Distress Syndrome

$P_{atm}$ – Atmospheric Pressure

$P_{ip}$ – Intrapleural Pressure

$P_{aiv}$ – Intra-alveolar Pressure

$Pa_o$ – Peak airway Pressure

NVP – Negative Pressure Ventilators

CPAP – Continuous Positive Airway Pressure

OSA – Obstructive Sleep Apnea

APAP – Auto titrating (adjustable) Positive Airway Pressure

BiPAP – Bilevel Positive Airway Pressure

IPAP – Inhalation Positive Airway Pressure

EPAP – Exhalation Positive Airway Pressure

PC – Pressure Controlled

SIMV – Synchronized intermittent mechanical ventilation

AC – Assist-control

PSV – Pressure support ventilation
RR – Respiratory Rate
IBW – Ideal Body Weight
MERS – Middle East respiratory syndrome
CDC – Centers for Disease Control and Prevention
AMBU – Artificial Manual Breathing Unit
BVM – Bag-valve mask
MATLAB - Matrix Laboratory
ARM – Advanced RISC (Reduced Instruction Set Computing) Machines
RAM – Random Access Memory
ROM – Read Only Memory
GIPO – General-Purpose Input/Output
LCD – Liquid Crystal Display
VC – Volume Control
LED – Light Emitting Diode
PIP – Peak inspiratory Pressure
WUV – Western University Ventilator
LMIC – Low- and Middle-Income Countries
$V_t$ – Tidal Volume
BPM – Breaths per Minute
I/E ratio – Inspiration–Expiration ratio
$FiO_2$ – Fraction of Inspired Oxygen
$EtCO_2$ – End-tidal Carbon Dioxide
PEEP – Positive End Expiratory Pressure
Chapter 1

Introduction

Respiratory disorders or lung diseases are conditions such as asthma, lung cancer, cystic fibrosis, pulmonary hypertension, tuberculosis (TB), emphysema, mesothelioma, to name a few [20]. These are either inherited genetically or are caused by the long-term exposure to external irritants that damage the lungs and the airways. Respiratory diseases produce health complications and life-threatening conditions if they are left untreated [20]. They make up to five out of the thirty most common causes of death around the world: Chronic Obstructive Pulmonary Disease (COPD), lower respiratory tract infection, lung cancer, TB, and asthma are ranked third, fourth, sixth, twelfth, and twenty-eighth, respectively [1].

More than one billion people around the world suffer from acute or chronic respiratory conditions [1]. Respiratory disorders are the third leading cause of death in Canada and the United States both in adults and infants [2-3]. In Canada, one in five people have a respiratory disorder. Over two million Canadians suffer from asthma, one of the leading causes of hospital admissions among children [4]. In 2014, lung cancer caused more cancer deaths among Canadians than colorectal, breast and prostate cancer combined [5]. The leading cause of hospitalization among adults is COPD, which also accounts for more than 10% of all disability-adjusted life-years (DALYs). DALYs is a metric that estimates the amount of active and productive life lost due to a condition [8]. Respiratory diseases impose a significant burden on the Canadian economy, particularly COPD, asthma, and lung cancer. Respiratory disorders account for 6% of annual healthcare costs in Canada. In 2014, respiratory disorders were estimated to cost the Canadian economy over $12 billion every year according to an analysis by the Conference Board of Canada [8].

The internal organs that are most vulnerable to injury and infection from the external environment are the lungs. The constant exposure to chemicals, such as toxic smoke of biomass fuel, particles, inhaling polluted outdoor air, and infectious organisms in the air cause lung infection. By 2030, it is estimated that 12 million Canadians will be affected annually by respiratory disorders [6] if there are no further enhancements and strategies made for dealing with respiratory diseases. The annual economic burden is also projected to be double by 2030.
Innovative strategies, policies to further reduce and modify the risk factors and treatments must be developed in order to reduce the imminent burden of these conditions on the economy and the health care systems. A potential area of research is in the development of technologies to assist patients with respiratory disorders.

1.1 Motivation

Extreme cases of respiratory diseases such as acute exacerbation of asthma, significant progression of COPD, TB patients who require care in the ICU and lung cancer patients who develop acute illness requiring ICU admission that can lead to respiratory failure, often require intubation and ventilatory assistance. Mechanical ventilators are there to support patients with respiratory illness but are available in a limited number. When the SARS-CoV-2 virus, a contagious airborne respiratory virus causing the disease known as COVID-19, was identified it took the entire world by storm. A key challenge caused by the virus was the shortage of mechanical ventilators worldwide.

One of the symptoms of the virus includes difficulty in breathing, and in extreme cases, patients required ventilation. In the spring and early summer of 2020, the number of infected people exceeded the number of hospital beds available and their access to medical infrastructure was therefore limited [21]. Many hospitals claim they cannot find enough devices that assist patients breathe and can mean the difference between life and death for individuals suffering from the coronavirus's most severe respiratory effects [39]. Doctors had to assess and prioritize resources such as ventilators to ensure infected people who had the best chance of survival received care first [39]. There was a serious scarcity of life-saving machines in the United States and other countries, with no straightforward means to increase manufacturing [39]. The hospitals did not have a sufficient number of additional ventilators stored for assisting every patient, as the cost of ventilators is exorbitant and hospital budgets are limited. In the COVID-19 case, there were 2,586 COVID-19 cases per 100,000 people in the world (as of July 2021) [7]. It was simply impossible to generate and distribute a large number of ventilators in such a short period of time. Mechanical ventilators are expensive machines. Not all countries in the world can manage to have ventilators stored up for emergency situations. The main motivation behind the current research project is a
result of these factors, especially the costs of ventilators, their accessibility, and the time required to make the devices. Furthermore, for rapid deployment, the dimensions and mass of the device were considered important as well.

The ongoing COVID-19 pandemic demonstrated the potential need for a low-cost ventilator system that can be rapidly deployed. Low-cost ventilators are easy-to-use emergency ventilators. An ideal low-cost ventilator uses standard components that are easily accessible to the public. The system can be easily recreated by anyone due to its open-source compatibility. It can also be designed to be able to ventilate Acute Respiratory Distress Syndrome (ARDS) patients. The low production cost can make these ventilators available when and where they are needed.

1.2 General Problem Statement
Open-source low-cost ventilators are disaster-situation\(^1\) machines that are made using readily available parts and components and using freely licensed designs. These ventilators have the potential to overcome the scarcity of ventilators that the world faced during the COVID-19 pandemic. There were many attempts to make a low-cost device in the past but none of them were adequate to be used for an emergency situation, nor did any of the devices have regulatory approval as a medical device. There are now many designs available, but obtaining regulatory approvals remains a significant challenge. The work being done in this project as a whole is directed towards making a low-cost ventilator device and testing it using protocols that are most relevant to eventual regulatory approval.

The existing open-source low-cost ventilator machines are not intelligent devices and require much work in order to achieve medical grade status. This is the reason why, despite so many open-source ventilator designs shared on the web, the world could not use the low-cost ventilator as a last resort when the COVID-19 virus was out of control. The purpose of this thesis is to develop a smart controller for the low-cost ventilator device. The work in this report proposes all the design specifications and processes used in order to form the controller so that in future if there were to be another respiratory virus that the world has to face again, a low-cost ventilator can be established. The affordable cost of this device can make it possible for under-developed

\(^1\) Disaster-situation: serious disruption situations where the functioning of a community exceeds its capacity to cope using its own available resources
and developing countries to use it as an emergency ventilator as well and not just during a pandemic.

1.3 Research Scope and Objectives
A volume-control ventilator design was considered for the project and both physical and software components were implemented. 

The primary objectives of this thesis were as follows:
1. To program a micro-controller device that can control the system as per the user input data.
2. To obtain and analyze the input data from the prototype device for various parameters and examine the output generated based on the input provided to check the efficiency of the system.
3. To evaluate and validate the sensor used for different waveforms generated from resistance values for healthy and non-healthy lung properties by connecting the device through an artificial lung device.
4. To add safety features to protect the patients from different scenario as explained in Chapter 4.
5. To connect relevant components and sensors to gather more information, which could help the clinicians further assess the patient’s condition.

1.4 Overview of the Thesis
The structure of the thesis is as follows:

Chapter 1 Introduction: The introduction chapter with motivation, problem statement, scope, and objective of the project.

Chapter 2 Literature Review: A review of breathing anatomy and mechanism, mechanical ventilators, types and working mechanisms, uses and risk of the mechanical ventilator device and types of low-cost open-source ventilators.
Chapter 3  Design and development of the Control System: working principle for the commercial and low-cost device, design specification and process and physical implementation discussions.

Chapter 4  Results and Discussion: Presenting the resulting waveforms of the analyzed output data and explaining their significance.

Chapter 5  Conclusion: Highlighting the contribution of this work and providing recommendations for future work.

Appendix A  Permissions: Permissions for images.

Appendix B  Codes: Arduino code used for the control system and MATLAB code.
Chapter 2

Literature Review

2.1 Introduction

This literature review provides a background into mechanical ventilation, and a discussion about key ventilator settings and waveforms for reading and analyzing output signals. A literature search was conducted using Google Scholar for literature published in the range from 2010 to 2021. The topics covered in this chapter include breathing anatomy in Section 2.2, under which mechanism of breathing, respiratory modes and factors that affect breathing are discussed. Mechanical ventilators, including types, working mechanisms, ventilator settings, uses, and risks are discussed in Section 2.3. Section 2.4 discusses the current COVID-19 situation and the need for emergency ventilators, and Section 2.5 briefly outlines the existing open-source ventilator designs.

2.2 Breathing Anatomy

Breathing is essential for survival. The human body can live without food for 3 weeks and water for 3 days, but only 3 minutes without air. Normal bodily functions cease to occur when brain is starved of oxygen. The act of breathing is called pulmonary ventilation [1] and is described as the process of air flow into and out of the lungs from the atmosphere during inspiration (breathing in air) and expiration (breathing out air) [11]. The air movements inside the lungs are governed by the principles of the gas laws. Pulmonary ventilation is dependent on three types of pressure [14]: atmospheric pressure ($P_{atm}$); intrapleural pressure ($P_{ip}$), the pressure within the pleural cavity; and intra-alveolar pressure ($P_{air}$), the pressure within the alveoli.

The air flows into the lungs due to the difference in the pressure. The air flows down a pressure gradient from an area of higher pressure to an area of lower pressure. Atmospheric pressure is greater than intra-alveolar pressure and intra-alveolar pressure is greater than intrapleural pressure [12]. The same principle applies during expiration, when air flows out of the lungs. During exhalation, pressure within lungs becomes greater than the atmospheric pressure.
2.2.1 Mechanism of Breathing

The two major steps involved during respiration are inspiration and expiration. When air enters the lungs, the process is called inspiration, and when the air leaves the lungs, it is called expiration. This process is shown in Figure 2.1. One full sequence of expiration and inspiration is called a respiratory cycle. The two general muscle groups that are used during normal inspiration are the diaphragm and the external intercostal muscles [9]. When a person takes bigger breath, additional muscles are required. The diaphragm moves towards the abdominal cavity when it contracts, this creates a larger thoracic cavity and hence there is more space for the lungs. The rib cage then expands and the volume of the thoracic cavity increases due to the contraction of the external intercostal muscles [19]. Ribs move upward and outward due to the contraction of the external intercostal muscles. This increase in volume leads to a decrease in intra-alveolar pressure [1]. Hence, a pressure lower than atmospheric pressure is created. This creates a pressure gradient that
then drives air into the lungs. The expansion and contraction of the thoracic cavity causes inspiration and expiration respectively.

The intra-alveolar (i.e., the pressure within the alveoli) and intrapleural pressures (i.e. the pressure within the pleural cavity) are dependent on certain physical features of the lung [19]. However, the ability to breathe i.e., to have air enter and leave the lungs during inspiration and expiration, respectively, is dependent on the air pressure of the atmosphere and the air pressure within the lungs [11].

2.2.2 Respiratory Mechanics

During normal inspiration, negative intrapleural pressure is generated. In other words, a pressure gradient is created between the atmosphere and alveoli, resulting in airflow. In mechanical ventilation, however, the pressure gradient results from increased (positive) pressure of the air source.

At the airway opening, peak airway pressure (\(P_{a_o}\)) is measured. It is the total pressure needed to push a volume of gas into the lungs [14]. This pressure is composed of the elastic recoil of the lung and chest wall (elastic pressure), the inspiratory flow resistance (resistive pressure) and the alveolar pressure present at the beginning of the breath (positive end-expiratory pressure [PEEP]) as seen in Figure 2.2. Hence, peak airway pressure is equal to the resistive pressure plus the elastic pressure plus the PEEP [15].

End-expiratory pressure in the alveoli is normally the same as atmospheric pressure. When alveoli fails to empty completely, the end-expiratory pressure may be positive relative to atmosphere. This pressure is called intrinsic PEEP or AutoPEEP. The end-inspiratory pressure represents the elastic pressure once PEEP is subtracted. The difference between the peak and the plateau pressure is the resistive pressure.
The main factors that affect breathing are the levels of oxygen and carbon dioxide in the blood, the blood’s pH level, and any respiratory disorders such as COPD, asthma, lung infection or collapsed lung.

### 2.3 Mechanical Ventilators

A mechanical ventilator is a device that aids a patient to breathe when they are having difficulties; for example, if they are recovering from a surgery or serious sickness or have difficulty breathing on their own for any reason (e.g. a critical illness). When using a ventilator, a hollow tube or a mask is placed in the patient’s mouth to connect them to the ventilator. Airflow is then pushed into patients’ lungs via the mechanical ventilator to help them breathe. Patients remain on the ventilator until their condition is improved or until they can breathe on their own. The two types of mechanical ventilation include the following:
• Invasive ventilation: usually performed in the intensive care unit with a tube inserted into the patient’s airway.
• Noninvasive ventilation: usually a mask that goes around a person’s mouth; it can be used at home by people that are facing respiratory difficulties.

2.3.1. Types of Mechanical Ventilation

There are two types of mechanical ventilator positive-pressure ventilation, in which air is pushed into the lungs; and negative-pressure ventilation, in which air is sucked into the lungs by making the chest expand and contract. These are further discussed in the following sections.

2.3.1.1 Negative Pressure Ventilation (NVP)

Negative pressure ventilators (NVP) were the first ventilators invented and they are rarely in use now. There are several types of NVPs, as discussed below:

• **Iron lung:** The iron lung is a metal cylinder that wraps the patient completely up to the neck as shown in Figure 2.3. The patient lays inside the cylinder. The opening through which the head is protruding, is sealed around the neck to avoid air leaking. An electric motor is connected by mechanical linkage to a flexible diaphragm (yellow) as shown in Figure 2.3. This diaphragm expands and contracts varying the air pressure inside the chamber, causing the chest to expand and contract.

• **Chest cuirass:** In order to create a negative pressure, a small shell is strapped to the patient’s chest. It is a more compact version of the iron lung. It encloses only the patient’s torso and is sealed around the neck and the waist. It is depressurized and repressurized by an external pump.
2.3.1.2 Positive Pressure Ventilation

Positive pressure ventilators were developed in the early 1950s. They were developed to treat polio patients with respiratory paralysis. In these types of ventilators, air is blown into the patient’s lung via a tube. They can be invasive or non-invasive.

Invasive ventilation is positive pressure delivered to the patient’s lungs via an endotracheal tube or a tracheostomy tube and these tubes are described further in detail below:

- **Endotracheal intubation**: In this case a tube is inserted into a patient’s trachea through the mouth or nose. Patients are sedated before putting them on invasive ventilators.
- **Tracheostomy**: In this case the tube is inserted through a hole that is made into the patient’s airway.

Non-invasive ventilation modes are those that use a mask rather than a tube. They can be used at home as well as in healthcare settings. There are three kinds of noninvasive mechanical ventilators:

- **Continuous positive airway pressure (CPAP)**: A device that delivers a constant and steady level of air pressure that is greater than the atmospheric pressure. It is applied to the upper respiratory tract of a person. People with all severities of obstructive sleep apnea (OSA) use CPAP.
• **Auto titrating (adjustable) positive airway pressure (APAP):** A device that can change air pressure according to the breathing pattern of a patient. The pressure setting is not limited to a single pressure, but to a range of pressures. It is built in such a way that it fluctuates within this range of pressures automatically while it delivers air.

• **Bilevel positive airway pressure (BiPAP):** A device that delivers air with different pressures for inhalation (IPAP) and lower pressure for exhalation (EPAP). This machine can be set to match a person’s breaths per minute. It can sense a significant shift in the patients breathing pattern and can adjust pressures accordingly.

The common form of mechanical ventilation currently used in hospitals is positive pressure ventilation. It pushes the air into the patient’s airways continuously and stops in regular pressure cycles, which enables the lungs to receive oxygen and expel carbon dioxide. These positive-pressure ventilators can be controlled in different ways, including volume- and pressure-controlled.

In volume-controlled mode, a preset volume of air (tidal volume) is delivered into the patient’s trachea regardless of the pressure in the airway. When the flow stops, the chest recoils and expels the air out. Volume control modes are generally constant flow modes, which means a constant flow is delivered, and this constant flow stops when the desired volume is achieved. Volume-control gives an advantage to the control of ventilation.

Figure 2.4 is an example of a volume control waveform. In this graph the pressure waveform is variable. The shape changes depending on lung compliance and airway resistance. Since pressure is not regulated or controlled in any way, the waveform created by the pressure takes a parabolic shape as the lungs distend during inspiration.
In pressure-controlled (PC) mode, inspiratory pressure is the control variable. It delivers air until the airway pressure limit is reached, and until the valve opens to expel air. This results in the square pressure waveform as shown in Figure 2.5, which increases the area under the pressure/time graph (the mean airway pressure is greater). Depending on the airway resistance and lung capacity, the volume of air delivered may vary.

During inspiration, a high inspiratory flow is provided to achieve the pressure limit (PC above PEEP). The flow rate needs to decrease over the course of inspiration in order to maintain this pressure. This generally takes the shape of a down sloping ramp. Flow will reach zero if the inspiratory time is long enough, as shown in the figure. In the absence of flow the prescribed constant pressure is in equilibrium with the peak alveolar pressure and equals plateau pressure at the end of the breath.
In dual control mode combines the advantages of volume control and pressure control, and delivers the airflow based on the requirement and response of the patient.

Table 2.1 summarizes the advantages and disadvantages of the volume and pressure control mode of positive pressure ventilation.

Figure 2.5: Example waveform for pressure-controlled ventilation waveform [22].
Table 2.1: Advantages and disadvantages of the volume and pressure control mode.

<table>
<thead>
<tr>
<th></th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume-Control Mode</td>
<td>• Since the tidal volume is guaranteed, it produces a more stable minute</td>
<td>• The mean airway pressure is lower and can serve as a disadvantage in patients</td>
</tr>
<tr>
<td></td>
<td>volume (i.e. it is the air inhaled and exhaled in a minute).</td>
<td>that suffer from severe hypoxia.</td>
</tr>
<tr>
<td></td>
<td>• It can maintain reliable minute volume if the airway resistance fluctuates</td>
<td>• Before the ventilator cycles to expiration, there is little time for gas</td>
</tr>
<tr>
<td></td>
<td>significantly.</td>
<td>exchange.</td>
</tr>
<tr>
<td></td>
<td>• Compared to the pressure control mode, the initial flow rate is lower.</td>
<td>• The mean airway pressure may be unstable during a leak. There will be no</td>
</tr>
<tr>
<td></td>
<td></td>
<td>volume delivered if the leak flow rate is equal to inspiratory flow rate.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Patient–ventilator dyssynchrony may occur due to insufficient flow.</td>
</tr>
<tr>
<td>Pressure-Control Mode</td>
<td>• Oxygenation is improved due to increased mean airway pressure.</td>
<td>• Tidal volume is variable since it is dependent on the respiratory compliance.</td>
</tr>
<tr>
<td></td>
<td>• Better gas exchange due to increased duration of alveolar recruitment.</td>
<td>• Since the volume is not controlled, it may result in trauma.</td>
</tr>
<tr>
<td></td>
<td>• This mode protects against pressure-induced alveolar injury, as the pressure</td>
<td>• A high initial flow may breach the pressure limit causing the alarms to go</td>
</tr>
<tr>
<td></td>
<td>level is controlled.</td>
<td>off.</td>
</tr>
<tr>
<td></td>
<td>• Patient’s comfort and work of breathing may be improved.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.3.2 Working Mechanism of the Mechanical Ventilator

Mechanical ventilators are dependent on the compliance and the resistance of the airway, and work by applying a positive pressure breath [7]. The volume of air that enters the lung during inhalation is called tidal volume ($V_t$) [7]. Resistance and compliance are affected by the state of the disease. In order to choose proper ventilator settings, understanding the compliance and resistance is important.

Four stages of mechanical ventilation include trigger, inspiratory, cycling and expiratory phase. Initiation of an inhalation either triggered by patients or set parameters in the ventilator by clinician is called the trigger phase. The inspiratory phase is when the inhalation of air into the patient’s lung takes place. The momentary pause stage after inspiration and before exhalation begins is called the cycling phase. The passive exhalation of air from the patient is the expiratory phase [7].

Once a patient is placed on mechanical ventilation, the clinicians have to set up the ventilator. There are different options that have to be selected depending on a patient’s needs. There are many different modes, such as synchronized intermittent mechanical ventilation (SIMV), assist-control (AC), and pressure support ventilation (PSV) [7]. The ventilator can be set to provide a given volume or pressure [7]. Volume assist control ventilators are recommended to use, as they are safe, simple to use and also provide complete ventilator support.

The rest of the parameters have to be set on the ventilator after choosing the mode. These parameters are the $V_t$, the breaths per minute (BPM), the respiratory rate (RR), the fraction of inspired oxygen ($\text{FiO}_2$), and the PEEP. Finally, the head of the bed of all patients on mechanical ventilation should elevated to at least 30 degrees and have continuous waveform end-tidal carbon dioxide ($\text{CO}_2$) (EtCO$_2$) monitoring. Section 2.3.2.1 and 2.3.2.2 further discuss PV (Pressure-Volume) relationships and ventilator settings.

2.3.2.1 Pressure-Volume (PV) Relationship

Compliance is equal to change the in volume over the change in pressure ($C' = \Delta V / \Delta P$ ; $C'$=compliance, $\Delta V$=change in volume, $\Delta P$=change in pressure), which means that when a small change in pressure causes a large change in volume, it means that the lung is very compliant. In
the case of a low compliant lung, it will take lot of pressure to make small amount of change in volume. If the compliance of the lung decreases, it results in higher pressure requirements, and vice versa. The waveform in Figure 2.6 explains this concept. In this example, the reading was taken where BPM is 18, $V_t$ is 600 ml and PEEP is 5 cmH$_2$O. In this case since the patient is not taking a voluntary breath there is no negative pressure. The ventilator starts and volume goes up to a certain pre-set tidal volume. As the volume starts to go into lungs the pressure will go up in the lung until it reaches maximum. The flow starts at that time and reaches a user defined preset. When the flow stops and comes out of the patient the pressure in the lung falls back down and volume will come out of the lung again and the cycle repeats.

![Figure 2.6: Example waveform to describe PV Relationship [23].](image)
2.3.2.2 Ventilator Settings

Ventilator settings are different for each individual. Adjustable ventilator settings differ with mode but include respiratory rate, tidal volume, trigger sensitivity, flow rate, waveform and inspiratory/expiratory (I/E) ratio.

Tidal volume and respiratory rate set minute ventilation. Excess volume risks overinflation and insufficient volume allows for atelectasis which is a condition where lungs collapse partially or completely. Too high of a rate risks hyperventilation and auto PEEP along with inadequate respiratory time. Too low of a rate risks respiratory acidosis and inadequate minute ventilation. A tidal volume of 6 to 8 mL/kg ideal body weight (IBW) was initially recommended for patients with acute respiratory distress syndrome (ARDS) [26]. Such low tidal volume is only suitable for certain patients who have normal lung mechanics, other patients such as those with trauma, obtundation, severe acidosis may have to started at slightly higher tidal volume (such as, 8 to 10 mL/kg) [26]. IBW is used to determine the appropriate tidal volume for patients rather than actual body weight who are receiving mechanical ventilation and have lung disease, as follows [24]:

- IBW (kg) males: 50 + 2.3 (height in inches - 60) or 50 + 0.91 (height in cm - 152.4)
- IBW (kg) females: 45.5 + 2.3 (height in inches - 60) or 45.5 + 0.91 (height in cm - 152.4)

The I:E ratio (inspiratory: expiratory ratio) is the ratio inhalation time versus exhalation time. The I:E ratio can also be adjusted in some modes of ventilation. The normal setting for patients with normal mechanics is a 1:2 ratio [27][30]. Patients with asthma or exacerbations of COPD (chronic obstructive pulmonary disease) should have ratios of 1:4 or even more to limit the degree of autoPEEP [24].

Either the inspiratory flow rate or the I:E ratio can be adjusted as part of the ventilator settings, but not both. The inspiratory flow is generally set at about 60 L/minute but can be increased up to 120 L/minute for patients with airflow limitation [27]. This facilitates having more time in exhalation, thereby limiting autoPEEP.

FiO₂ (fraction of inspired oxygen) is initially set at 1.0 (100% oxygen) [28] and is subsequently decreased to the lowest level necessary to maintain adequate oxygenation. PEEP can be applied as one of the settings in any ventilator mode. PEEP reduces airspace closure at the end of expiration and increases end-expired lung volume. Patients undergoing mechanical ventilation
benefit from the application of PEEP at 5 \( cmH_2O \) [29], as it limits the atelectasis that frequently accompanies endotracheal intubation, sedation, paralysis, and/or supine positioning. Oxygenation improves in higher levels of PEEP in disorders such as cardiogenic pulmonary edema and ARDS.

2.3.3 Uses of Mechanical Ventilators

The mechanical ventilator decreases the work of breathing of a patient until they recover or can breathe on their own. This machine makes sure that the patients are receiving adequate amount of oxygen and that the carbon dioxide is removed. This device can be used for short term respiratory assistance in surgeries and also while patient have difficulty in breathing where the respiratory muscle is not properly functioning. The advancement of mechanical ventilation has helped doctors meet individual requirements of patients. They can be used for

- short-term periods for respiratory assistance during surgeries,
- longer periods for critically ill patients, or
- at home by people who have difficulties in breathing normally.

The advantages of mechanical ventilations are the following:

- The patients can rest their respiratory muscles as they do not have to work as hard to breathe.
- It helps patients get adequate amount of oxygen and removes carbon dioxide.
- It prevents injury from aspiration by preserving a stable airway.

Mechanical Ventilation does not heal patients. It allows them to be stable while other treatments and medication helps them to recover.

The major risk of using mechanical ventilation is an infection. The breathing tube may allow germs to enter the lungs. The longer a patient is on a ventilator, the higher the risk of getting an infection. Lung damage caused by over inflation or repetitive opening and collapsing of alveoli is another risk associated with using mechanical ventilation. Some patients may require prolonged support when they are unable to be weaned off of a ventilator. This may result in a tracheostomy.

2.4 Obstructive and Restrictive Lung Disease

Lung conditions have been classified as obstructive or restrictive lung diseases by doctors [38]. The condition that makes it hard to exhale the air in the lungs is obstructive lung disease and the
condition where people find it difficult to fully expand their lungs with air is restrictive lung disease [38]. Both the diseases share the same main symptom which is shortness of breath during exertion.

People with restrictive lung diseases cannot fully fill their lungs with air hence their lungs are restricted from fully expanding [38]. This disease results from the condition causing stiffness in the lungs, chest wall, weak muscles or damaged nerves that may cause restriction in lung expansion [38].

People with obstructive lung diseases have shortness of breath caused due to difficulty exhaling all the air from the lungs [38]. In this condition exhaled air comes out more slowly than normal. The most common causes of obstructive lung diseases are COPD, asthma, cystic fibrosis to name a few.

2.4.1 COVID-19 and Need for Emergency Ventilators

The ongoing global COVID-19 pandemic, caused by a novel severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), is an obstructive lung disease which was first identified in December 2019 in Wuhan, China [1]. A public health emergency regarding COVID-19 of international concern was declared by the World Health Organization on January 30th, 2020. It was later declared as a pandemic on March 11th, 2020 [2]. As of June 7th, 2021, there are more than 174 million [21] confirmed cases of COVID-19. This airborne virus overwhelmed the current medical infrastructure around the world. It is considered very dangerous, as it caused spikes in mortality rate. There are more than 3.74 million [3] confirmed deaths attributed to COVID-19.

SARS-CoV-2 is a type of coronavirus which is named due to the appearance of a “corona” which means “crown”. The outer layer of this virus is covered with spike proteins, which surrounds it like a crown as shown in Figure 2.7 Coronaviruses are common in different animals. An animal coronavirus rarely infects humans [10]. There are many different kinds of coronaviruses, some of which cause serious diseases, including severe acute respiratory syndrome (SARS) and Middle East respiratory syndrome (MERS). COVID-19 identified in 2019 [19] has caused a global pandemic of respiratory illness [10]. COVID-19 can also cause lasting health problems in some people that have survived the illness. This virus is presumed to be originated in an animal and later mutated to cause illness in people and is diagnosed with the help of a laboratory test. It is spread from person to person through air when an infected person sneezes, coughs, laughs, sings, talks or
breathes [10]. The larger droplets of virus particles may fall to the ground within a few seconds, but tiny particles linger in the air [10] for a longer period of time. Virus particles can accumulate in indoor places where there is poor ventilation and there are a lot of people gathered, which is the reason why physical distancing; hand hygiene and mask-wearing are considered essential for the prevention of COVID-19.

Figure 2.7: COVID-19 illustration created by Centers for Disease Control and Prevention (CDC) (Image adapted from Wikipedia, Public Domain Image).

When a person is exposed to the virus, the symptoms show up within two to 14 days [31]. The infected person then becomes contagious to others and should remain in quarantine for 10 to 20 days depending upon their immune system, severity of their illness and the negative test results for the virus. COVID-19 symptoms are as follows [31]:

- Chills or Fever
- Difficulty breathing or shortness of breath
- Cough
- Body or muscle aches
- Loss of taste or smell
- Headaches
- Sore Throat
- Vomiting or nausea
- Diarrhea
- New fatigue
- Runny nose or congestion
Some infected people have mild symptoms, while others have no symptoms at all. In some cases, it can lead to respiratory failure, nervous system problems, lasting lung and heart muscle damage, kidney failure or death [10].

Some people suffering from COVID-19 got affected more than others. In extreme cases, it got so dangerous that people were unable to properly oxygenate their body, thereby requiring a ventilator. Since, the disease is so contagious, COVID patients were usually separated from other ICU (Intensive Care Unit) patients. Hospitals with greater number of patients around the world had to set up separate ICUs for COVID and non-COVID patients, which resulted in exhausting available resources [32].

The coronavirus that causes COVID-19, like other viruses, can mutate (change). Different variants of this virus have emerged and are spreading globally. B.1.1.7 was first identified in the United Kingdom [10] and then later spread to over 120 countries. P.1 was detected in Brazil and has spread to over 50 countries. B.1.351, which was first detected in South Africa, has spread to more than 80 countries [34]. B.1.617 (Delta variant) is another variation of the virus that was recently discovered in India and has spread to over 40 countries [33]. Waves of cases show up in a pattern, such as it has been seen in other virus pandemics before. When COVID-19 was first dominant around the world, all countries saw a number of increasing cases. Some countries saw higher number of infections in the beginning followed by a decline. Later when the virus mutated, the infected countries encountered a second wave of increased cases. Some countries, such as Canada, also experienced third wave of increased cases. And now has entered into the fourth wave [34]. One of the reasons for the increased number of cases is the mutation of the virus. It enabled coronavirus to spread faster from person to person and caused more severe diseases.

Treatment of symptoms of COVID-19 are include over-the-counter medications and fever reducers for mild cases. Severe cases may require hospitalization and may receive treatments that include oxygen and ventilation support. Severe COVID-19 cases can be fatal. COVID-19 vaccines are developed to defeat the virus and are being rolled out around the world with active vaccination programs in progress.
2.4.2 COVID-19 and Mechanical Ventilation

Ventilators are an example of the medical infrastructure that is generally available. However, they do not exist in a high enough density to handle the volume of patients associated with pandemics. It is a technology that is currently in critical short supply, and one of the reasons why many fatalities were reported throughout the world. Ventilators are essential for treating COVID-19 patients with severe respiratory failure.

Past studies have also showed that during a massive pandemic, hospitals will not have sufficient resources to treat patients requiring ventilator support, and patients be allocated to ventilators on a first-come first-served basis.

Ventilator shortages have been experienced around the world [5]. In order to overcome the ventilator shortage situation, there have been innovative solutions with promising results, such as the use of a single ventilator to support multiple patients during disasters, and rapidly presenting solution with open-source ventilator designs, which would eliminate the need for rationing ventilators.

The current health care system relies on mass-manufactured ventilators from a small selection of suppliers. These ventilators are expensive, ranging from $5,000 to $50,000. Hence, it is not possible for hospital or medical centers to stock on ventilator machines. Developing and underdeveloped countries suffer the most as they are low- and middle-income countries. This supply model fails when there is a high demand for such a low-volume specialty product, like ventilators. Hence, low-cost open-source ventilators are one solution that can help in a situation like COVID-19.

2.5 Low-Cost Open-source Ventilators

An open-source ventilator is made by using freely-licensed design and is often thought to be used for disaster situation, the sources include the components that are freely available. Parts can be 3D-printed instead of purchasing to help keep the cost low. Documentation and testing of open-source ventilators has been going on since the start of the COVID-19 pandemic. Section 2.5.1 discusses the different types of low-cost ventilator systems that can be used to treat COVID-19 patients.
2.5.1 Types of Open-Source Ventilators

There has been a lot of research going on designing low-cost open-source ventilators system since late 2020’s. The conceptual designs of field portable ventilators for domestic and military emergency response exists, but there is limited information on how to reconstruct the ventilators as the software used for making them was not shared openly and some of the codes were written in assembly language [19] which is not easy to replicate for most users. However, design considerations that might be useful for future designers has become available [35][36][37] and although COVID-19 motivated most of the work on emergency ventilators, the concept of developing a low-cost ventilator device is not new. Most of the open-source solutions can be categorized as part of four groups, as shown in the subsections below.

2.5.1.1 Centralized Air Ventilation Systems

Centralized air ventilation systems are designed to supply air to several patients at once. In order to provide air to 10–20 patients at a time, an industrial oil-free air compressor is used. The air is mixed with oxygen, which is extracted from standard oxygen bottles. The air that is supposed to be received by each patient is regulated by individual valves. These valves are connected to microcontrollers and monitored for individual patients. This solution is suitable for intubated patients. The design of the terminal that allows individual patients to be connected can differ depending on the design.

2.5.1.2 Using a Blower for Ventilation Systems

This type of design uses a blower to feed the air to the individual patients. It is a compact and low-cost mask respirator. This solution has been designed and prototyped successfully [17]. It is a low power device and integrated sensor for airway pressure can detect leakage and occlusion. This solution is suitable for patients that need to be moved from one place to another. Figure 2.8 proposes such solution with a miniature turbine where oxygen and air intake are mixed in a single step. The housing box is 3D printed. A standard car blower can also be used for this design, which can be controlled by microcontrollers, such as an Arduino.
2.5.1.3 The Artificial Manual Breathing Unit (AMBU) Bag Ventilation System Design
This type of design uses a standard AMBU bag to assist the patients with breathing that do not require intubation. This design is not based on constant blower use [18]. The AMBU bag can be automated to be able to adjust breathing rate and volume of air. It can also be made to regulate inspiration to expiration ratio and PEEP rate. Other design solutions include using a DC motor and a microcontroller by compressing a bag-valve mask (BVM) to achieve a higher degree of control over the respiration process [18]. Automating the system would eliminate the need for a person to push the BVM. These solutions however require a certain manufacturing infrastructure. Figure 2.9 shows a AMBU bag that can be used as a short-term solution for manual resuscitation.
2.5.1.4 The Air Pump Type

This solution is similar to those design that uses a blower. It needs air pumps that are easily available like aquarium air pumps to supply necessary airflow. Meeting the maximum airflow requirements is one of the critical factors that these solutions need to ensure. Hence, it might not be suitable for the patients in dire conditions. Design solutions like using a repurposed air pump from a soldering station with Arduino to create ventilator have been proposed [25]. The advantages and disadvantages of open-source ventilators discussed in the section are presented in the table 2.2.

Table 2.2: Summary of types of open-source ventilators.

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| Centralized air ventilation systems | • It uses readily available components and is inexpensive.  
• It can provide air to 10–20 patients at a time.  
• It is most suitable for intubated patients. |
<p>| | • These solution designs are not available as open-source content. Hence, it is difficult to replicate the system for open-source design. |</p>
<table>
<thead>
<tr>
<th>Using a blower for ventilation system</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- It is a low-power device.</td>
<td>- It treats one patient at a time. Hence, it may not be as cost-effective as the centralized air ventilation systems when it comes to treating multiple patients at once.</td>
</tr>
<tr>
<td></td>
<td>- It has a blower unit to provide adequate ventilation to patients.</td>
<td>- Successfully prototyped design [11] exhibits very few details, which is not enough to be considered full open hardware source, or to be easily replicated.</td>
</tr>
<tr>
<td></td>
<td>- It can be wirelessly operated, as well as with batteries.</td>
<td></td>
</tr>
<tr>
<td>The AMBU bag ventilation system design</td>
<td>The main advantage of this solution is that the materials required are readily available in the hospital.</td>
<td>- It is a temporary solution.</td>
</tr>
<tr>
<td></td>
<td>- This design runs using an open source Arduino microcontroller [17], and there are enough details published to work as a guide to build a similar device.</td>
<td>- If the automated BVM stops running, manual pressing of the bag is required. Even with proper training, only few individuals can maintain adequate mask seal and a patent airway with one hand while squeezing the bag fully to achieve the 800 ml to 1200 mL standard volume.</td>
</tr>
<tr>
<td></td>
<td>- The automated BVM eliminates the need for the person pushing on BVM. It can determine tidal volume, breaths per minute and inhalation and exhalation ratio.</td>
<td></td>
</tr>
<tr>
<td>The air pump type</td>
<td>- It is similar to the blower type design.</td>
<td>- It is not suitable for patients with vulnerable conditions.</td>
</tr>
<tr>
<td></td>
<td>- It is a low-cost device, and the materials used are readily available.</td>
<td>- It is not easy to replicate as these solution designs are not available as open-source content.</td>
</tr>
</tbody>
</table>
There are several approaches that are being attempted in order to create an open-source ventilator that can be used readily around the world when there is scarcity of ventilators again. The design solution used in open ventilator community include pressure regulators, pneumatic systems, servo gas modules, pumps, bellows, screw compressors and fan blowers. The use of BVMs/AMBU bags are the most favored design solutions by both the academic literature and the maker community. Western University ventilator is also an open-source BVM ventilator that uses a stepper motor and an Arduino microcontroller designed for this project. Section 2.5.3 discusses more about the Western University ventilator.

2.5.2 Western University Ventilator (WUV)

WUV is a low-cost volume controlled positive pressure ventilator device. Figure 2.10 is the picture of the first prototype that was developed at Robarts Research Institute. As seen in the figure it consists of a BVM also called as AMBU bag. It uses a stepper motor for the movement of the arm. The arm is connected to the motor using a nylon thread. The arm is positioned at pre-set start position before the squeezing of the BVM takes place. When the device is turned on, the arm reaches full excursion, stops momentarily and reverses away from BVM towards start position. It stops and holds its position again and then the cycle continues. The function and feature of the WUV are further described in Chapter 3.

![Western University low-cost ventilator device.](image)

Figure 2.10: Western University low-cost ventilator device.
2.6 Conclusion

This chapter discussed breathing anatomy, including the mechanism of breathing and normal respiratory mechanics. Mechanical ventilators, its types and the working mechanism, uses and risk involved while using mechanical ventilators were discussed. A brief introduction was made of COVID-19 and how a mechanical ventilator helps patients with COVID-19. Different types of low-cost open-source existing ventilator designs were discussed and compared. The Western University ventilator (WUV) was introduced in this chapter. In Chapter 3, more detailed discussion about the WUV and the control system design for the WUV.
Chapter 3
Design and Development of the Control System

3.1 Introduction
This chapter describes the process used for the design and development of the control system for the low-cost ventilator system. The sections in this chapter outline the working principle of the low-cost ventilator or WUV (Western University Ventilator), design specification and processes used for the selection of the sensors, simulation of the device in MATLAB and physical implementation of the machine.

3.2 Western University Ventilator (WUV)
The WUV is a low-cost volume controlled positive pressure ventilator device. Figure 3.1 is the picture of the first prototype that was developed at Robarts Research Institute. The device uses a stepper motor for the movement of the arm. The arm is connected to the motor using a nylon thread.

Figure 3.1: Western University low-cost ventilator device.
The arm is positioned at a pre-set start position before the squeezing of the BVM takes place. The squeezing operation, that compresses the BVM is described as follows:

- The actions of the moving arm are as follows:
  1. The arm is positioned at pre-set start/stop position, as shown in Figure 3.2.
  2. The arm then moves towards the BVM until it reaches maximum squeezing position as shown in Figure 3.3. The arm then stops momentarily and reverses away from the BVM towards the start/stop position.
  3. The arm stops and stays at the start/stop position and the cycle repeats.

Figure 3.2: Squeezing of the BVM at start position.

Figure 3.3: Squeezing of the BVM at full excursion.
• The actions of the moving arm motion toward the BVM are as follows:
  1. The air-in one-way valve or duckbill valve (Air Inlet One-way valve & O2 Reservoir Socket as shown in Figure 3.4) in the BVM is closed.
  2. The air-out one-way valve or expiratory valve to patient in the BVM as illustrated in Figure 3.4 is open.
  3. The air flow is center patient port and into patient.
  4. The side port one-way valve in the BVM is closed resulting in no flow. A single one-way valve controls both the center port flow and the side port flow through different channels.
• The actions of the moving arm motion away from the BVM (refilling bag)
  1. The air-in one-way valve in the BVM is open.
  2. The air-out one-way valve or expiratory valve to patient in the BVM is closed.
  3. The air flow is channelled from the patient and out the side port of BVM through the PEEP valve.
  4. The side port one-way valve in the BVM is open.
  5. Note that specific details of the BVM may vary; one unit that is being using has a pressure-relief valve set to a maximum of 40 cm H2O.

Figure 3.4: Bag valve mask (image from litfl.com, image free to use).
The following are the functions that exist in the WUV:

- The user can select a value for desired breaths per minute (BPM), Inspiratory to Expiratory ratio (I:E) and Tidal Volume ($V_t$). The BPM ranges from 10 to 30 (in 1 BPM steps). I:E ratio can be set as 1:1, 1:2, 1:3, 1:4 as shown on the display in Figure 3.5 as 0.5, 0.33, 0.25, 0.2. $V_t$ ranges from 100 ml to 900 ml (in steps of 100 ml).

- The ventilator starts working when the enable switch is switched ON. When the enable switch is on, the parameters are then adjusted, the WUV first finishes the current cycle and then starts the next cycle with new parameters. Enable switch has to be switched OFF for the ventilator to stop working. When the enable switch is off and then the parameters are adjusted, the WUV can start immediately with the new parameters when the enable switch is turned back on.

![Ventilator Display](image)

**Figure 3.5:** Display screen that shows the user input data. In the display above, BPM is 18, TV is 700 ml, I:E is 0.50, PPS is 6197 pulse/second (motor speed), Inspiratory pressure is 0.8 cmH$_2$O, Expiratory pressure is 0.9 cmH$_2$O, and the number of cycles.
3.3 Design Process
This section describes the components used for making the WUV and the sensor considered to make the control system smarter. This section further illustrates the flow chart for the controller of the device.

3.3.1 Components of the Existing WUV
The following components listed below are currently used in the WUV. They were already part of the WUV.

- Stepper Motor
The motor selected for this project was a NEMA 23 frame motor, with nominal torque of 126 N-cm (1.26 N-m or 178 oz-in) as shown in Figure 3.6. The motor has excellent response to starting, stopping and reversing. A wide range of rotational speeds can be realized with this motor. It has full torque at standstill and precise positioning and repeatability of the movement.

![Figure 3.6: Stepper motor.](image)

- Stepper Motor Driver
The TB6560 micro-stepping driver was used as a stepper motor driver for the project as shown in Figure 3.7, which provides a maximum of 3A per phase, and micro-steps up to 1/16. It receives step and direction input form the Arduino. However, there is a heat sink in this unit, and it requires a fan.
• Proximity Detector

The proximity detector used was the MAX30105 (Maxim Integrated high-sensitivity optical sensor) as shown in Figure 3.8. This optical sensor is mounted on the bottom of the arm. The proximity sensor is used to detect the presence of the squeezer mechanism at the start position. This sensor is very effective in determining whether the arm has returned back to its original or start position.

Figure 3.7: Stepper motor drive in right which is later enclosed in a 3D printed box attached to the base of the WUV in left.

Figure 3.8: MAX30105 high-sensitivity optical sensor.
• **Pressure Sensor**

The pressure sensor used in this project is MPS20N001 pressure sensor as shown in Figure 3.9. This pressure sensor has a higher range up to 40 kPa. It works well in the range from 0 to 60 cm$H_2O$. This range is close to the value that is needed in the project to observe the pressure during testing for different ventilator parameters such as Vt, BPM and I:E ratio. This pressure sensor is mounted in the adapter of the BVM in-patient port.

![MPS20N001 pressure sensor](image)

Figure 3.9: MPS20N001 pressure sensor.

• **LCD display:**

Figure 3.10 shows the 20×4 character display device that is currently used in the WUV. It is used for displaying the user input data and the output pressure data for inspiration and expiration.

![LCD display](image)

Figure 3.10: LCD display.
3.3.2 Components of the Modified WUV

The objective of this project is to make the control system intelligent and smarter. In order to make the controller smart, five alerts were added to the current design. A flow sensor was also added to determine the flow at the patient-side port. The microcontroller and other sensors that were selected for this project in order to make the change to the existing WUV to make the control system smarter are explained in this subsection.

- Arduino Due

The microcontroller selected for this project is an Arduino Due as shown in Figure 3.11. Arduino Due is one of the larger Arduino boards with 12 PWM (Pulse Width Modulation) channels, 54 digital I/O (Input/Output) pins, 12 analog inputs and two analog outputs. This board is powered by an ARM (Advanced RISC (Reduced Instruction Set Computing) Machines) processor. The smaller size of the ARM processor, its reduced complexity, and lower power consumption makes the Arduino Due suitable for miniaturized devices such as the WUV. Arduino Due is most suitable for this project due to its powerful ARM core and large Random Access Memory (RAM) (96 kB) and Read Only Memory (ROM) (512 kB). This project requires a number of General-Purpose Input/Output (GPIO). In case of this project the number of input and output are from the motor, the motor driver, the LCD display, the pressure sensor, the optical sensor and the buzzer, hence the Due is most suitable choice.

![Arduino Due microcontroller](image)

Figure 3.11: Arduino due microcontroller.
• **Flow Sensor**

Figure 3.12 shows a Sensirion SFM 3300-D digital flow meter that was added and used in the WUV. This device measures the flow rate of air. The SFM 3300-D is the disposable single use version. There is a reusable version that withstands washing and autoclaving procedures, the SFM 3300-AW[41]. Both versions are well suited for proximal flow measurement in medical ventilation and other respiratory applications [41]. The SFM 3300-D measures bidirectional flow volumes of up to ±250 sLm (standard liter per minute) [40]. It has extremely fast update time of 0.5ms and has small dead space (<10ml) [40]. This sensor features medical cones for pneumatic connection to the standard breathing circuit. The sensor also has a mechanical interface for a user-friendly electrical connection [40]. This sensor is mounted after the pressure sensor in the adapter of the BVM in-patient port of the WUV.

![Figure 3.12: SFM 3300-D Flow sensor.](image)

• **Buzzer**

Figure 3.13 shows a piezo buzzer that was added to the WUV. The purpose of using this buzzer is to provide alerts when the WUV is not functioning properly. There are number of alerts as shown in Table 3.1 that have been added to the programming of the system. When the system does not satisfy the condition as described in the table, the alarm goes off and it produces a sound to notify the users.

![Figure 3.13: Piezo Buzzer.](image)
Table 3.1: Alert conditions implemented in the control system.

<table>
<thead>
<tr>
<th>Alarm Name</th>
<th>Detection</th>
<th>To stop</th>
<th>Display Message in LCD Screen</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIP pressure exceeded</td>
<td>If the pressure increases to more than 40 cm $H_2O$, as this pressure damages the lungs.</td>
<td>One complete breathing cycle with no over pressure.</td>
<td>HIGH PRESSURE ERR!</td>
</tr>
<tr>
<td>Under pressure</td>
<td>This is for any leaks in the breathing lube, which can be checked by the clinicians if the alarm goes off.</td>
<td>Normal Plateau pressure should be obtained.</td>
<td>LOW PRESSURE ERR!</td>
</tr>
<tr>
<td>Wire break</td>
<td>When wire connecting the motor and arm breaks, the arm does not move. Hence the squeezing of the BVM does not take place. This has to be immediately addressed by clinicians when alarm goes off.</td>
<td>The wire has to be fixed.</td>
<td>WIRE BREAK ERR!</td>
</tr>
<tr>
<td>Motor not working</td>
<td>This is when the motor stops moving.</td>
<td>The motor has to be checked to fix the condition. If not fixable, should be replaced.</td>
<td>MOTOR STUCK ERR!</td>
</tr>
</tbody>
</table>

3.3.3 Control Implementation

The WUV provides assured tidal volumes when operating in the volume-control (VC) mode. The operator selects the required tidal volume as required by the patient. It is usually 6 to 8 mL/kg of ideal body weight and a minimum respiratory rate. This provides a minimum assured minute ventilation. The advantage of VC mode is that the delivered tidal volumes produce a more stable minute volume, which meets the physiological needs for adequate gas exchange. A disadvantage
is that an insufficient flow may give rise to patient–ventilator dyssynchrony. However, this issue can be addressed by adjusting the required tidal volume and respiratory rate.

As described in Section 3.3.2, an Arduino Due microcontroller board was selected to control the device. The microcontroller runs a simple loop to achieve the user-prescribed performance. The control loop is triggered by the internal timer set by the user inputs. Once the input tidal volume is set and the device is powered on, the arm moves towards the BVM and then squeezes the BVM until it reaches the input tidal volume. The stepper motor used in this project moves the arm in the constant inspiration speed as determined by the potentiometers. Inspiration position for the arm is determined by reading the tidal volume potentiometer. Once the actuator has reached maximum position, it records the time taken for inspiration (This algorithm used for motor actuation can be further explored in Appendix A where the program codes have been provided.). This loop keeps repeating to deliver the required breaths to the patient. A flow sensor has been added to check the flow of the air received at the output port. The analysis for flow rate is done in the next chapter. If the loop is interrupted by any one of the alert conditions in Table 3.1, the safety alarm goes off and immediate supervision is needed in order to resolve the issue.

Figure 3.14 shows the loop that the ventilator controller follows. The patient’s safety is ensured through the continuous monitoring of the airway pressure with a pressure sensor. This pressure sensor is connected to the output of the BVM. This pressure sensor also triggers an alarm if the pressure is too high or low as stated in Table 3.1 to alert the attending physicians. There is no way to differentiate the alarm conditions for when the motor is not working and when there is a wire break (as stated in Table 3.1) at the moment. Differentiating these alarm conditions can be one of the future work directions. The logic used for both the alarm conditions is to detect that the squeezing of the bag does not take place, which can be caused by the wire breaking or if the motor stops working. Hence, once the bag is not squeezed by the arm, an alert is raised. These added features, such as the alert system for different conditions and the flow sensor to check the actual flow rate to the patient’s lungs adds feature to the current WUV compared to the previous version of the WUV.
Figure 3.14: Ventilator controller loop.
3.4 Physical Implementation

The sensors described in Section 3.3.3 were added to the prototype and programmed accordingly. A flow diagram in Figure 3.15 shows how the flow sensor is interfaced with other components of the ventilator. The flow sensor is mounted on the patient port of AMBU bag and is connected to the test lung via breathing circuit. This chapter describes the components used in the previous version of WUV and the new components that are added in the current version of the WUV. It also describes WUV working mechanisms. In the next chapter, the testing of system was conducted, and data were recorded from the flow meter to check the accuracy of the system. The testing part and results are further explained in the next chapter.

![Figure 3.15: Flow diagram of physical implementation.](image-url)
Chapter 4

Results and Discussion

This chapter describes the tests that were conducted in the prototype device after the flow sensor and alert codes were added to the microcontroller. The flow data were collected when the flow sensor was attached to the patient port of the BVM. The flow sensor reads real-time flow data. These flow data are important because they determines the airflow at the patient port of the BVM. The flow sensor was connected to the arduino, which monitored the ventilator performance parameters such as pressure, load, current, voltage and airflow. Flow data were collected every 30 ms. The BVM and a QuickLung breathing simulator (test lung) were connected as shown in Figure 4.1 by connecting a breathing circuit with a flow meter between them. The topics discussed in this chapter include motivation of the experiment in Section 4.1, the objective of the experiment in Section 4.2, device-based test with goals and results discussion in Section 4.3, safety alarm test in Section 4.4, followed by the conclusion in Section 4.5.

Figure 4.1: Test setup with the lung simulator showing the Western University Ventilator. The BVM has a manually settable PEEP valve and a flow sensor connected.
4.1 Motivation and Objective of the Experiment

The motivation for doing the experiments is to make sure that the ventilator performs under different lungs and ventilator parameters. It is also to ensure that the ventilator is safe to use.

The objective of conducting the experiment is to evaluate and validate the sensor used for the device under different ventilator and lung settings. A micro-controller device was programmed so that the controller of the system works as per the user’s input data. Various experiments in the device with different settings were conducted to analyze and examine the output data generated based on the input provided, in order to check the efficiency of the system and the accuracy of the sensor used. The objective was to check whether the input volume of air is the same as the output volume of air at the BVM port before it goes to the test lungs and whether the sensor is collecting data for the desired duration of a breath cycle. Testing the added safety alarms for various conditions is also one of the objectives of the experiments. Various tests were done to gather more information, which could help the clinicians further assess the patient’s condition.

4.2 Tidal Volume Test

General requirements from ISO 80601-2-79:2018 (Medical Electrical Equipment) specify that for volume-controlled breath types, during the testing, the error of the delivered volume of individual breaths shall not deviate by more than 35%, and that the delivered volume averaged over a one-minute interval shall not deviate by more than 25%.

In order to conduct the tidal volume test, a flow meter was connected to the patient port of the BVM in the WUV as shown in Figure 4.1. The test lung and the ventilator device are connected by a breathing circuit. When the ventilator is switched on, the BVM is squeezed by the arm, up to a certain level, according to the input ventilator parameters. When squeezing the BVM, the air passes from the BVM to the patient port. The flow sensor is attached to the patient port. The air flows through the flow sensor and the flow rate is detected by the sensor. This flow rate data are recorded to further evaluate the performance of the ventilator device. The pressure values are also being recorded simultaneously along with flow rate values. The pressure during inspiration increases as the BVM squeezes and reaches a peak inspiratory value. The pressure then decreases during expiration.
Tidal volume ($V_t$) is a ventilator input setting, which determines the volume of gas delivered by the mechanical ventilator. The tidal volume input must correlate to the actual volume delivered to ensure that the tidal volume requirements are accurately met. The volume of gas delivered by the mechanical ventilator to the lungs is dependent on the compression mechanism on the BVM, the compliance and resistance of the lung, the I:E ratio limits and the torque of the motor providing the mechanism movement.

4.2.1 Methods

The test equipment is set up as shown in Figure 4.1. Once, the input tidal volume is set, the device runs for 1–5 cycles. As the BVM is compressed and released, the air that comes out of the BVM passes through the flow sensor through the breathing circuit and goes to the test lung. The flow data from the flow sensor are then recorded. The instantaneous peak inspiratory and expiratory pressures are displayed in the 20×4 display of the device. The pressure observed at every point is also recorded. Ventilator parameters, such as $V_t$ and BPM, were varied, test lung parameters such as resistance were adjusted, and flow and pressure readings were recorded.

4.2.2 Results

The data collected from the flow sensor were used to compute output tidal volume. The inspiratory and expiratory pressure, output volume and error between input and output volume for various test settings are shown in Table 4.1, 4.2 and 4.3. Figures 4.2, 4.3 and 4.4 respectively shows the bar graph for the input $V_t$ and output $V_t$ obtained from the flow data for different test procedures, which makes the comparison easier. The blue bar and orange bar represent input and output tidal volume, respectively. The output $V_t$ is calculated by taking the integration of the set of flow data for each condition using MATLAB (MATLAB codes used for calculation are presented in Appendix B). The integration is done using the trapezoidal rule by approximating the region under the flow graph and calculating its area.

The error percentage is calculated taking the percentage difference between the input and output tidal volumes. From the table, it can be observed that the error percentage for ventilator performance for 5 (cmH$_2$O/ (l/s)) and 10 (cmH$_2$O/ (l/s)) resistance is less than 15%. However, for higher resistance 20 (cmH$_2$O/ (l/s)), the error percentage is more than 15%. This is because by applying different lung resistance values, the lung volume at the end of the exhalation is increased.
In presence of increased airway resistance, an increased transpulmonary pressure (the difference between the alveolar pressure and the intrapleural pressure in the pleural cavity) is required to produce a given tidal volume, and therefore the work of breathing is increased. Different control parameters can also create issues when they are at their maximum values.

Table 4.1: Recorded data for different lung settings.

Test Procedure 1: Compliance: 20 (ml/cmH₂O), Resistance: 20 (cmH₂O/(l/s)), PEEP: 10 cmH₂O, I:E ratio: 1:2

<table>
<thead>
<tr>
<th>BPM</th>
<th>Input Tidal Volume (V_t) (ml)</th>
<th>Inspiration Pressure (cmH₂O)</th>
<th>Expiration Pressure (cmH₂O)</th>
<th>Output V_t (ml)</th>
<th>Error %</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>300</td>
<td>21.6</td>
<td>14.0</td>
<td>260</td>
<td>13.33</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>30</td>
<td>13.9</td>
<td>370</td>
<td>7.50</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>36.1</td>
<td>14.3</td>
<td>470</td>
<td>6.00</td>
</tr>
<tr>
<td></td>
<td>600</td>
<td>39.1</td>
<td>13.5</td>
<td>550</td>
<td>8.33</td>
</tr>
<tr>
<td>14</td>
<td>300</td>
<td>23.6</td>
<td>14.7</td>
<td>270</td>
<td>10.00</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>29.4</td>
<td>13.9</td>
<td>380</td>
<td>5.00</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>35.2</td>
<td>41.5</td>
<td>480</td>
<td>4.00</td>
</tr>
<tr>
<td></td>
<td>600</td>
<td>42.0</td>
<td>13.3</td>
<td>550</td>
<td>8.33</td>
</tr>
<tr>
<td>16</td>
<td>300</td>
<td>25.8</td>
<td>13.7</td>
<td>280</td>
<td>6.67</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>33.4</td>
<td>14.3</td>
<td>390</td>
<td>2.50</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>41.5</td>
<td>42.6</td>
<td>480</td>
<td>4.00</td>
</tr>
<tr>
<td></td>
<td>600</td>
<td>44.8</td>
<td>13.6</td>
<td>560</td>
<td>6.67</td>
</tr>
<tr>
<td>18</td>
<td>300</td>
<td>26.5</td>
<td>14.0</td>
<td>280</td>
<td>6.67</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>34.1</td>
<td>13.3</td>
<td>390</td>
<td>2.50</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>42.6</td>
<td>13</td>
<td>490</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>600</td>
<td>47.3</td>
<td>13.2</td>
<td>580</td>
<td>3.33</td>
</tr>
</tbody>
</table>
Figure 4.2: Input $V_t$ and Output $V_t$ for Test Procedure 1.

Table 4.2: Recorded data for different lung settings.

Test Procedure 2: Compliance: 20 (ml/cmH$_2$O), Resistance: 5 (cmH$_2$O /l/s), PEEP: 10 cmH$_2$O, I:E ratio: 1:2

<table>
<thead>
<tr>
<th>BPM</th>
<th>Input Tidal Volume ($V_t$)</th>
<th>Inspiration Pressure cmH$_2$O</th>
<th>Expiration Pressure cmH$_2$O</th>
<th>Output $V_t$ (ml)</th>
<th>Error %</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>300</td>
<td>21.9</td>
<td>11.9</td>
<td>260</td>
<td>13.33</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>27.3</td>
<td>12.4</td>
<td>380</td>
<td>5.00</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>31.7</td>
<td>12.1</td>
<td>460</td>
<td>8.00</td>
</tr>
<tr>
<td></td>
<td>600</td>
<td>35.2</td>
<td>12.4</td>
<td>560</td>
<td>6.67</td>
</tr>
<tr>
<td>14</td>
<td>300</td>
<td>21.7</td>
<td>12.0</td>
<td>260</td>
<td>13.33</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>27.5</td>
<td>12.4</td>
<td>380</td>
<td>5.00</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>32.4</td>
<td>12.0</td>
<td>480</td>
<td>4.00</td>
</tr>
<tr>
<td></td>
<td>600</td>
<td>36.6</td>
<td>12.3</td>
<td>580</td>
<td>3.33</td>
</tr>
<tr>
<td>16</td>
<td>300</td>
<td>21.7</td>
<td>12.2</td>
<td>280</td>
<td>6.67</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>28.3</td>
<td>12.4</td>
<td>390</td>
<td>2.50</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>33.6</td>
<td>35.7</td>
<td>490</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>600</td>
<td>37.8</td>
<td>12.5</td>
<td>580</td>
<td>3.33</td>
</tr>
<tr>
<td>18</td>
<td>300</td>
<td>23.2</td>
<td>12.4</td>
<td>280</td>
<td>6.67</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>29.4</td>
<td>12.2</td>
<td>390</td>
<td>2.50</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>35.7</td>
<td>12.5</td>
<td>500</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>600</td>
<td>39.1</td>
<td>12.5</td>
<td>590</td>
<td>1.67</td>
</tr>
</tbody>
</table>
Figure 4.3: Input $V_t$ and output $V_t$ for Test Procedure 2.

Table 4.3: Recorded data for different lung settings.

Test Procedure 3: Compliance: 20 (ml/cmH$_2$O), Resistance: 50 (cmH$_2$O /l/s)), PEEP: 10 cmH$_2$O, I:E ratio: 1:2

<table>
<thead>
<tr>
<th>BPM</th>
<th>Input Tidal Volume ($V_t$)</th>
<th>Inspiration Pressure cmH$_2$O</th>
<th>Expiration Pressure cmH$_2$O</th>
<th>Output $V_t$ (ml)</th>
<th>Error %</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>300</td>
<td>29.9</td>
<td>15.0</td>
<td>250</td>
<td>16.67</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>41.3</td>
<td>14.2</td>
<td>340</td>
<td>15.00</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>54.4</td>
<td>14.3</td>
<td>400</td>
<td>20.00</td>
</tr>
<tr>
<td></td>
<td>600</td>
<td>62.6</td>
<td>15.1</td>
<td>450</td>
<td>25.00</td>
</tr>
<tr>
<td>14</td>
<td>300</td>
<td>33.7</td>
<td>14.9</td>
<td>250</td>
<td>16.67</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>48.2</td>
<td>14.7</td>
<td>350</td>
<td>12.50</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>60.2</td>
<td>15.1</td>
<td>400</td>
<td>20.00</td>
</tr>
<tr>
<td></td>
<td>600</td>
<td>71.1</td>
<td>15.2</td>
<td>480</td>
<td>20.00</td>
</tr>
<tr>
<td>16</td>
<td>300</td>
<td>36.1</td>
<td>14.7</td>
<td>250</td>
<td>16.67</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>52.7</td>
<td>14.2</td>
<td>350</td>
<td>12.50</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>68.4</td>
<td>15.0</td>
<td>420</td>
<td>16.00</td>
</tr>
<tr>
<td></td>
<td>600</td>
<td>76.4</td>
<td>15.4</td>
<td>480</td>
<td>20.00</td>
</tr>
<tr>
<td>18</td>
<td>300</td>
<td>41.3</td>
<td>14.8</td>
<td>250</td>
<td>16.67</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>58.3</td>
<td>14.3</td>
<td>320</td>
<td>20.00</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>75.0</td>
<td>15.2</td>
<td>400</td>
<td>20.00</td>
</tr>
<tr>
<td></td>
<td>600</td>
<td>87.1</td>
<td>13.1</td>
<td>500</td>
<td>16.67</td>
</tr>
</tbody>
</table>
Figure 4.4: Input $V_t$ and output $V_t$ for Test Procedure 3.

In all the results obtained while testing with different ventilator and lung parameters, the error percentage between set or input tidal volume and output volume is less than 35%, which fulfils the condition from the general requirements from ISO 80601-2-79:2018 (Medical Electrical Equipment) Part 2-79: Particular requirements for basic safety and essential performance of ventilatory support equipment for ventilatory impairment.

4.4 Sensor Validation

This section presents how the data collected for the Test Procedures 1, 2 and 3 were cross-checked. In order to examine the accuracy of the flow sensor, input BPM and output BPM were compared. This comparison shows whether the sensor is working for the desired duration of a breath cycle.

The data collected during the test procedure were used to calculate the output BPM. There are five types of data collected each cycle when the arm squeezes the BVM. They are voltage, current, load, pressure, and airflow. The current draw was used to determine one full breathing cycle. The current was relatively low when the BVM was not squeezed and high when the BVM was squeezed. An example of the calculation is as follows: from the set of data from Test Procedure 1, where $V_t$ of 300 ml and BPM of 12 is input to the system. In order to detect one cycle, current data was taken as reference, it can be observed from the table 4.4 (is the portion of the table which refers to the data that was taken) that current data changes at Row 8 (0.67A) and it changes to high
(up until high current value it is one cycle) and back to low current at Row 177 (next cycle begins). So, in this case 169 samples are taken during this interval. The sampling time used in the program is 30 ms. Multiplying 169 samples by 30 ms gives us the duration of the interval which is 5.07 s. However, the initial condition of the device was set to a BPM of 12, which is 5 seconds per breath. The error between the input and output breathing duration for one cycle is 1.4\% \((\frac{5.07-5}{5})\times100\%) \). All other conditions were calculated in a similar manner. The full table is presented in the Appendix B as Table 1.

Table 4.4: Portion of data from Appendix B, Table 1.

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Voltage</th>
<th>Current</th>
<th>Load</th>
<th>Pressure</th>
<th>Airflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.</td>
<td>12.03</td>
<td>1.55</td>
<td>0.15</td>
<td>13.98</td>
<td>-8.87</td>
</tr>
<tr>
<td>8.</td>
<td>12.54</td>
<td>0.67</td>
<td>0.15</td>
<td>13.72</td>
<td>-7.63</td>
</tr>
<tr>
<td>176.</td>
<td>12.52</td>
<td>1</td>
<td>0.14</td>
<td>13.99</td>
<td>-6.53</td>
</tr>
<tr>
<td>177.</td>
<td>12.54</td>
<td>0.64</td>
<td>0.15</td>
<td>13.61</td>
<td>-4.73</td>
</tr>
</tbody>
</table>
Table 4.5: Calculated error data for Input and output seconds per breath.
Test Procedure 1: Compliance: 20 (ml/cmH₂O), Resistance: 20 (cmH₂O /l/s), PEEP: 10 cmH₂O, I:E ratio: 1:2

<table>
<thead>
<tr>
<th>Tidal Volume (ml)</th>
<th>BPM</th>
<th>Input second per breaths</th>
<th>Output second per breath</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>5</td>
<td>5.07</td>
<td>1.40</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>4.28</td>
<td>4.35</td>
<td>1.64</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3.75</td>
<td>3.81</td>
<td>1.60</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>3.33</td>
<td>3.39</td>
<td>1.80</td>
</tr>
<tr>
<td>400</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>5</td>
<td>5.01</td>
<td>0.20</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>4.28</td>
<td>4.32</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3.75</td>
<td>3.81</td>
<td>1.60</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>3.33</td>
<td>3.39</td>
<td>1.80</td>
</tr>
<tr>
<td>500</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>5</td>
<td>5.07</td>
<td>1.40</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>4.28</td>
<td>4.35</td>
<td>1.64</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3.75</td>
<td>3.75</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>3.33</td>
<td>3.39</td>
<td>1.80</td>
</tr>
<tr>
<td>600</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>5</td>
<td>5.01</td>
<td>0.20</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>4.28</td>
<td>4.32</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3.75</td>
<td>3.81</td>
<td>1.60</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>3.33</td>
<td>3.39</td>
<td>1.80</td>
</tr>
</tbody>
</table>
Table 4.6: Calculated error data for Input and output seconds per breath.

Test Procedure 2: Compliance: 20 (ml/cmH$_2$O), Resistance: 5 (cmH$_2$O /l/s), PEEP: 10 cmH$_2$O, I:E ratio: 1:2

<table>
<thead>
<tr>
<th>Tidal Volume (ml)</th>
<th>BPM</th>
<th>Input second per breaths</th>
<th>Output second per breath</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>12</td>
<td>5</td>
<td>5.01</td>
<td>0.20</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>4.28</td>
<td>4.32</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3.75</td>
<td>3.81</td>
<td>1.60</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>3.33</td>
<td>3.39</td>
<td>1.80</td>
</tr>
<tr>
<td>400</td>
<td>12</td>
<td>5</td>
<td>5.04</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>4.28</td>
<td>4.35</td>
<td>1.64</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3.75</td>
<td>3.75</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>3.33</td>
<td>3.36</td>
<td>0.90</td>
</tr>
<tr>
<td>500</td>
<td>12</td>
<td>5</td>
<td>5.04</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>4.28</td>
<td>4.32</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3.75</td>
<td>3.78</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>3.33</td>
<td>3.39</td>
<td>1.80</td>
</tr>
<tr>
<td>600</td>
<td>12</td>
<td>5</td>
<td>5.04</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>4.28</td>
<td>4.32</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3.75</td>
<td>3.81</td>
<td>1.60</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>3.33</td>
<td>3.36</td>
<td>0.90</td>
</tr>
</tbody>
</table>
Table 4.7: Calculated error data for Input and output seconds per breath.

Test Procedure 3: Compliance: 20 (ml/cmH\textsubscript{2}O), Resistance: 50 (cmH\textsubscript{2}O/(l/s)), PEEP: 10 cmH\textsubscript{2}O, I:E ratio: 1:2

<table>
<thead>
<tr>
<th>Tidal Volume (ml)</th>
<th>BPM</th>
<th>Input second per breaths</th>
<th>Output second per breath</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>12</td>
<td>5</td>
<td>5.04</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>4.28</td>
<td>4.32</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3.75</td>
<td>3.81</td>
<td>1.60</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>3.33</td>
<td>3.39</td>
<td>1.80</td>
</tr>
<tr>
<td>400</td>
<td>12</td>
<td>5</td>
<td>5.07</td>
<td>1.40</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>4.28</td>
<td>4.32</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3.75</td>
<td>3.81</td>
<td>1.60</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>3.33</td>
<td>3.39</td>
<td>1.80</td>
</tr>
<tr>
<td>500</td>
<td>12</td>
<td>5</td>
<td>5.07</td>
<td>1.40</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>4.28</td>
<td>4.35</td>
<td>1.64</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3.75</td>
<td>3.81</td>
<td>1.60</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>3.33</td>
<td>3.36</td>
<td>0.90</td>
</tr>
<tr>
<td>600</td>
<td>12</td>
<td>5</td>
<td>5.04</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>4.28</td>
<td>4.32</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3.75</td>
<td>3.78</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>3.33</td>
<td>3.39</td>
<td>1.80</td>
</tr>
</tbody>
</table>

The mean error and standard deviation for all the error between input and output time for each breathing cycle calculated for all three set of test procedure is 1.17% and 0.51%, respectively. Hence, the range of error for the sensor for input and output breathing cycle duration is between 0.66% to 1.68% (1.17 ± 0.51)%.

The inspiratory and expiratory pressure at every point is measured and recorded as well, which provides the clinicians with not just the peak pressure but a range and a pattern of inspiratory and expiratory pressure values. The clinicians can further assess these values to evaluate the patient’s condition. Some of the inspiratory and expiratory pressure waveforms are shown in Figures 4.5, 4.6 and 4.7. These waveforms resulted from data collected in three different conditions. From these
waveforms, it can be seen that as the resistance of the lung increases, the inspiratory and expiratory pressures increase as well. Similar patterns were recorded for all of the tests done (The data can be found in Appendix B.). When the lung resistance was 5 cmH₂O as shown in Figure 4.6, the peak inspiratory pressure was around 32 cmH₂O, and the peak expiratory pressure was around 13 cmH₂O. The mean of this condition is 31.83 cmH₂O, and the range of variability is between 31.66 cmH₂O and 31.997 cmH₂O. As the lung resistance increased from 5 cmH₂O to 20 cmH₂O as shown in Figure 4.5, the peak inspiratory and peak expiratory pressure also increased to 35 cmH₂O and 15 cmH₂O, respectively. The mean of this condition is 35.17 cmH₂O, and the range of variability is between 35 cmH₂O and 35.34 cmH₂O. Similarly, when the lung resistance is 50 cmH₂O, as shown in Figure 4.7, the peak inspiratory and expiratory pressures increased to 55cmH₂O and 16 cmH₂O, respectively. The mean of this condition is 54.75 cmH₂O, and the range of variability is between 54.68 cmH₂O and 54.82 cmH₂O.

Figure 4.5: Pressure and time graph for input $V_t$ of 400 ml, 14 BPM and PEEP of 10 cmH₂O for Test Procedure 1.
Figure 4.6: Pressure and time graph for input $V_t$ of 400 ml, 14 BPM and PEEP of 10 cmH$_2$O for Test Procedure 2.
The tidal volume testing showed that the input volume of air was almost equal to the output volume, and the error percentage met the requirements from ISO 80601-2-79:2018 (Medical Electrical Equipment) for volume-controlled breath types. Input breathing duration and output breathing duration was compared, and the range of error was found for the sensor used to check its accuracy. However, more rigorous tests should be done for a longer period of time to check the durability and effectiveness of the system.

The combination of control parameters, such as BPM, $I$, $E$, $V_t$, compliance, resistance, and PEEP, can create issues when they are at the maximum values for the device. The purpose of testing the device in low and extreme conditions is to check and determine if the interplay of ventilatory parameters causes malfunction or dysfunction of the ventilator.

Figure 4.7: Pressure and time graph for input $V_t$ of 400 ml, 14 BPM and PEEP of 10 cmH$_2$O for Test Procedure 3.
4.5 Safety Alarm Testing

Safety alarms were added to the ventilator device as stated in the objective of the experiment in Section 4.1. These alarms were added to conditions such as when the peak inspiratory pressure exceeds or is under pressure, when the wire that moves the arm breaks and when the motor is not working. For these four different alarm settings, testing was conducted to check whether the alarms go off when these conditions are met. These conditions are critical, as they determine the safety of the patient. A buzzer was added to the physical system. When the buzzer starts to make a sound, a clinician can address the issue. A Light Emitting Diode (LED) was used instead of the buzzer to show that the alarm goes off when the conditions are met. Figures 4.8, 4.9 and 4.10 show the testing performed according to the conditions described by the test numbers in Table 4.8. In all of the conditions mentioned in the table, the alarm is turned on and hence the LED is on.

Table 4.8: Safety Alarm Test Conditions.

<table>
<thead>
<tr>
<th>Test No.</th>
<th>Alarm Name</th>
<th>Test Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Peak inspiratory Pressure (PIP) Exceeded</td>
<td>Letting the pressure be higher than 40 cmH₂O by holding the patient port of BVM as shown in Figure 4.9.</td>
</tr>
<tr>
<td>2.</td>
<td>Under Pressure</td>
<td>To detect leaks in the breathing tube, the condition was tested with a breathing tube not attached as shown in Figure 4.10.</td>
</tr>
<tr>
<td>3.</td>
<td>Wire Break</td>
<td>The wire was manually held, so the arm does not move as shown in Figure 4.11.</td>
</tr>
<tr>
<td>4.</td>
<td>Motor not working</td>
<td>The arm was made not to squeeze the bag by holding the wire as shown in Figure 4.11 to test the alarm condition.</td>
</tr>
</tbody>
</table>
Figure 4.8: The LED turns on when the PIP is higher than $40 \text{ cmH}_2\text{O}$ (as defined in the program), as shown in the picture for Test 1.

Figure 4.9: The LED turns on when the PIP is lower than $5 \text{ cmH}_2\text{O}$ (as defined in the program), as shown in the picture for Test 2.
Figure 4.10: The LED turns on when the wire breaks as the squeezing of the bag will not take place, which also satisfies the condition that the motor is not working since the bag will not be squeezed shown in the picture for Tests 3 and 4.

In conclusion, these tests were conducted to determine if the high pressure, low pressure, wire break and non-operating motor will activate an alarm for the safety of the patient. These conditions that activate the alarm can be checked by the clinicians so that the issue can be addressed and fixed.
Chapter 5

Conclusion

5.1 Concluding Remarks
The work presented in this thesis is towards the development of an intelligent and smarter controller module for the WUV (Western University Ventilator). A literature review was performed to understand the working principle of the ventilator device, and to identify the different types of existing low-cost open-source ventilators designs. A working prototype was developed for the controller module, and various sensors that were considered for the design were discussed and added. The prototype has been tested on the test lungs and the results were discussed in Chapter 4. The ventilator operates effectively even with the combination of different ventilator and lung parameters. The flow sensor provided a fairly consistent result for delivered volume with an average deviation of 15%. The inspiration and expiration pressure is measured at all times and the pressure pattern can be used by the clinicians to evaluate a patient’s condition. Safety alarms were tested for four conditions, and it worked competently.

The current WUV prototype that was worked on during this thesis includes user-controlled breath rate and tidal volume. It is a volume control device, and it has a number of alerts to make the system smart. A flow sensor was added to measure the flow rate at the output of the ventilator. This is important because knowing the flow rate ensures the correct amount of air that is being pushed into the lungs. Low volume and high volume of air in the lungs can both damage the lungs and put the patient’s life at risk. Hence, adding a flow sensor and observing the output data makes sure that the volume of air provided in the lungs is adequate[16].

The on-going COVID-19 pandemic was a huge motivation for the start of this project. The COVID-19 pandemic resulted in limited healthcare resources. The risk of limited ventilators was one of the greatest concerns. Ventilator shortage affected every country around the world but mostly Low- and Middle-Income Countries (LMIC). A simple easy to build low-cost ventilator device such as WUV, which can be made with readily available components and can provide a reliable ventilator solution, can be used in case of surge demand [11]. The relative low cost of this
project could potentially provide a ventilator solution in other resource constricted environments as well [11].

5.2 Contribution

The contributions of the work presented in this thesis are as follows:

1. An Arduino Due micro-controller device was programmed to control the system. The user inputs TV, BPM and I/E ratio. These input data determine the speed of the motor, which was controlled via the microcontroller.

2. Control system features for alarms capable of identifying issues were identified and added. This includes situations such as over pressure and under pressure, wire break and non-operating motor. A flow sensor was added to the device at the output port. This was done to make sure that the same volume of air is pushed into the lungs as the input data.

3. Testing was conducted with various conditions, which included different lung parameters settings to check the reliability of the system. The device was evaluated for various resistance of the lungs through an artificial lung device. The output generated was examined for various input parameters from the prototype device.

5.3 Future Work

The work performed in this project can be further enhanced. Some changes that can be made for future versions of prototype are listed below:

1. Sensors for measuring the fraction of inspired oxygen \( (FiO_2) \), end-tidal \( CO_2 \) and PEEP value can be added to improve the functionality of the device. Measurement of \( FiO_2 \) provides an estimation of the oxygen content during inhalation. Interpreting \( FiO_2 \) values and understanding oxygen delivery are vital for proper treatment of people. Et\( CO_2 \) values provides clues about respiratory efforts made by the patient.

2. Multiple modes of ventilation such as pressure control, assist control, can be included to make the system more intelligent.

3. A real-time monitoring system can be connected to get the data for real time usage of the system. These data can be made into waveforms so that the clinician will be able to know the pattern for TV, inspiratory and expiratory pressure,

4. User factors can be improved by adding a touch screen display device to enhance the system.
5. The device has to be tested on more rigorous conditions (such as in different weather conditions, running the device to its maximum capacity for a longer period of time, taking data for more cycles and comparing the results and so on) to check the durability of the system. Extensive testing of the ventilator’s repeatability should also be conducted.
References


[33] India. Ministry of Health and Family Welfare. Genome sequencing by ISACOG shows variants of concern and a novel variant in India [Internet]. 24 Mar 2021. New Delhi:


Appendices

Appendix A

Permissions and Approvals

The permission for use of images are presented in this Appendix:

For Figure 2.1

---

Kenhub feedback

Fri 5/28/2021 12:18 PM

To: Heena Shrestha

Hi there,

Thanks for reaching out to us.

We give permission to use our illustrations in a dissertation or thesis for free if they don’t get published. Otherwise, a paid license is required. If your thesis will not be published (this does not include the university library), we can gladly allow you to use up to 10 of our images for free if you name Kenhub correctly as the source of them.

Please make sure to use the correct reference for the images, it is "© Kenhub (www.kenhub.com); Illustrator: XY". You can find the illustrator when you hover over the images in our atlas.

Let me know if you have any further questions about this!

Cheers,

---

How would you rate my reply?
For Figure 2.2

"Permission to use Images and contents from opentextbc.ca website" from BCcampus Website Contact Us Form

BCcampus IT Support
Thu 5/27/2021 10:32 PM
To: Heena Shrestha

Hi Heena,

Permission is not required for you to use openly licensed materials. We endeavour to provide proper attribution for each open textbook in the BC Open Textbook Collection published by BCcampus.

For Anatomy And Breathing – The Process of Breathing, scroll to the very bottom of the online home page and under "Metadata" you'll see the suggested attribution statement for this book. "The CC licence permits you to retain, reuse, copy, redistribute, and revise this book – in whole or in part – for free providing the author is attributed as follows": Anatomy and Physiology by OSCRiceUniversity is licensed under a Creative Commons Attribution 4.0 International License.

For more information, see our Answer "May I obtain permission to use an image or text from a resource in the BC Open Textbook Collection?" You can find this FAQ by choosing "Permissions" from the list of Main Topics on the Open Ed Answer Guide page here: https://open.bccampus.ca/help/

Sincerely,
For Figure 2.3, 2.7, 2.8 and 2.9
Appendix B
This appendix includes the Arduino code used to conduct alarm test and add and read data from flow sensors. This appendix also includes MATLAB codes used for integration of observed flow data to calculate output tidal volume used for analysis work presented in this thesis. The appendix is divided into three sections Alarm code (Section B.1), Flow Sensor code (Section B.2), MATLAB Codes (Section B.3) and Table 1.

Section B.1

Alarm Codes

/*
* This code is part of the Western University emergency ventilator project
* The intent is to develop and test a 3D-printed, rapidly deployed
* ventilation-assist device, which could automate compression of a conventional bag valve mask
* This version incorporates a homing routine that will adjust the start position
* of the actuator cam
*
* Position sensing uses the MAX30105 using the IR reflection as a proximity measurement.
* Hardware Connections for the MAX30105 (Breakoutboard to Arduino):
* -5V = 5V (3.3V is allowed)
* -GND = GND
* -SDA = A4 (or SDA pin 20 on Due)
* -SCL = A5 (or SCL pin 21 on Due)
* -INT = Not connected
* The MAX30105 Breakout can handle 5V or 3.3V I2C logic. We are running at 5V; the Due seems to tolerate the
5V logic signals on SDA and SCL, and the 3.3V Due pulses are adequate to trigger the stepper driver.

* 5V logic signals on SDA and SCL, and the 3.3V Due pulses are adequate to trigger the stepper driver.

* /

#include <AccelStepper.h> // required for control of the stepper driver
#include <Wire.h> // required for communication with the MAX30105 sensor
#include "MAX30105.h" // required for control of the MAX30105 sensor
#include "HX711.h" // required to read the MPS20N004D pressure sensor, via HX711 signal conditioner
#include <LiquidCrystal_I2C.h> // required for LCD display

// Define a stepper driver and the pins it will use

// note that we are using an established stepper driver routine that works well with a step and direction indexer

// this approach would also work with a servo motor driver that can accept step and direction input, if needed

AccelStepper stepper(1, 12, 13); // this sets AccelStepper code to step (pin 12) and direction (pin 13) mode

// Initiate the MAX30105 sensor library

MAX30105 particleSensor; // the MAX30105 is used in this application as an IR proximity sensor

// Initiate the LCD display library

LiquidCrystal_I2C lcd(0x27, 20, 4); // set the LCD address to 0x27 for a blue 20 x 4 display

// Initiate the library for the HX71 signal conditioner; this is normally used for strain gauges

HX711 scale;
#define MIN_PRESSURE 5
#define MAX_PRESSURE 40

// HX711 circuit wiring
const int LOADCELL_DOUT_PIN = 2;  // HX711 is two-wire communication; pin 2 is data out
const int LOADCELL_SCK_PIN = 3;   // HX711 is two-wire communication; pin 3 is clock

//const float slope = 0.00001809;   // calibration constants for pressure sensor MPS20N 001;
// ADU to cm H2O
float intercept = 0;               // calibration constants for pressure sensor MPS20N 001; ADU to cm H2O
const float slope = 0.00001726;   // calibration constants for pressure sensor MPS20N 002; ADU to cm H2O

//const float intercept = -9.489;   // calibration constants for pressure sensor MPS20N 002; ADU to cm H2O
//const float slope = 0.00001629;   // calibration constants for pressure sensor MPS20N 003; ADU to cm H2O
//const float intercept = -40.489;  // calibration constants for pressure sensor MPS20N 003; ADU to cm H2O

// define constants
const long error_tolerance = 50;          // 50 microsteps = about 10 degrees motor rotation; this is allowed actuator position error each cycle
const int cycle_time_tolerance = 10;      // error if achieved cycle time is not within 10 ms of requested
const long start_position = 1350;         // calculated for 15mm capstan; can be set to whatever is appropriate for BVM
const long home_position = 1350;          // calculated for 15mm capstan
const long homing_speed = 10000; // this is the speed that will be used while the actuator is homing

const long max_allowed_speed = 12000; // do not exceed this speed in pulses per second; the motor will likely stall

const long expiration_speed = 7000; // note that if this speed is much greater, the bag does not have time to expand

const long acceleration = 40000; // all motion cycles are executed with this acceleration in pps per second

const int enable_pin = 5; // the pin assigned to the enable/disable switch

const long max_position = 7900; // microsteps; plateau position, calculated for 15mm capstan; actuator never reaches this

const long never_exceed_position = 7500; // microsteps; calculated for 15mm capstan

const unsigned long inspiration_hold = 0; // time in ms to hold at peak inspiration; variable is not used but could be added functionality

unsigned long inspiration_speed = 7000; // inspiration cycle speed in pulses per second; derived by input parameters

long derived_position; // this is the estimated actuator position, derived from the proximity sensor

float proximity = 0; // proximity is the raw value returned from the IR channel of the optical sensor

int trigger_pin = 10; // a digital trigger can be set on this pin, to coordinate other events

int BPM_pin = 0; // this is the analog pin used to read the BPM potentiometer

int BPM = 10; // this is the value for BPM; set by default at 10 breaths per minute

int TV_pin = 1; // this is the analog pin used to read the TV potentiometer
int TV = 500; // this is the value for TV; set by default at 500 ml
int IE_pin = 2; // this is the analog pin used to read the I:E potentiometer
float IE = 0.33; // this is the value for IE; set by default at 0.33 (i.e. 1:3)
long insp_position; // microsteps to reach desired inspiration volume; calculated for 15mm capstan
long mid_position;
long total_cycles = 0; // total number of cycles since program initiation
long correction_cycles = 0; // total number of cycles for which actuator did not return to within error_tolerance
long error_cycles = 0; // total number of cycles for which system did not achieve cycle period within cycle_time_tolerance
unsigned long start_time = 0; // this is the start time of the current cycle, in ms
unsigned long cycle_period = 2000; // this sets the cycle time in ms; default value of 2 seconds
unsigned long time_to_inspiration; // this is the time in ms that is recorded to reach max inspiration
unsigned long time_to_expiration; // this is the time in ms that is recorded to retract actuator from inspiration to rest
unsigned long elapsed_time; // time in ms that as elapsed since beginning of current respiratory cycle
unsigned long cycle_time; // time in ms that has elapsed for current compete respiratory cycle; should equal cycle_period
float inspiration_time; // this is the time in ms that is calculated as required to reach max inspiration
unsigned long stroke; // this is the actuator stroke (in motor micro-steps) to reach the required TV
float peak_insp_pressure; // pressure value at the peak of inspiration (i.e. max actuator position) cm H2O
float end_insp_pressure; // pressure value at the end of inspiration (i.e. actuator retracted) cm H2O
float initial_pressure; // initial pressure reading (required for wire-break condition checking)
float average_pressure; // average pressure value during continuous monitoring cm H2O
int DEBUG = 1; // set this to 1 to have testing results sent to serial

// these values are the derived constants to predict actuator position from proximity reflectance
// these are consistent with calibration using a nominal 15 mm capstan diameter
const float Plateau = -17.76;
const float SpanFast = 2436.5;
const float KFast = 0.00009773;
const float SpanSlow = 1260.2;
const float KSlow = 0.00001314;

// these values are the derived constants to predict the required motor displacement to reach TV
// these are based on an exponential association curve; this is just a rough estimate, awaiting calibration
const float Y0 = -17.76;
const float TV_plateau = (max_position - start_position);
const float K_TV = 0.003044;
bool motor_notMoved_err = false;       // error for motor not moving
bool wire_break_err = false;            // error for wire break
bool cycle_time_err = false;            // error for time duration of the cycle
bool max_pressure_err = false;          // error for maximum pressure
bool min_pressure_err = false;          // error for minimum pressure
bool initial_position_err = false;      // error for not returning to initial position

void setup() {
  Serial.begin(9600);                           // enable serial communications for debugging to
  pinMode(enable_pin, INPUT);                    // enable to input pin to accept values from
                                                      // the toggle switch
  Serial.println("Ventilator Assist V5.6");
  Serial.println(" ");
  lcd.init();                                     // initialize the lcd
  lcd.backlight();                                 // turn on the LCD backlight
  display_splash_screen();                       // display a brief splash screen to the LCD
  // initialize pressure sensor
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
  intercept = getIntercept();
// Initialize optical reflectance sensor

if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C port, 400kHz speed
{
    Serial.println("MAX30105 was not found. Please check wiring/power.");
    //while (1);
}

//Setup to sense the reflection from the underside of the actuator arm

byte ledBrightness = 0xFF; //Options: 0=Off to 255=50mA; I have set to 255 for maximum performance

byte sampleAverage = 8; //Options: 1, 2, 4, 8, 16, 32

byte ledMode = 2; //Options: 1 = Red only, 2 = Red + IR, 3 = Red + IR + Green

int sampleRate = 100; //Options: 50, 100, 200, 400, 800, 1000, 1600, 3200

int pulseWidth = 411; //Options: 69, 118, 215, 411

int adcRange = 16384; //Options: 2048, 4096, 8192, 16384

particleSensor.setup(ledBrightness, sampleAverage, ledMode, sampleRate, pulseWidth, adcRange); //Configure sensor with these settings

//Setup initial values for the speed and acceleration of the stepper motor

stepper.setMaxSpeed(homing_speed); // calculated for 15mm capstan

stepper.setAcceleration(acceleration); // calculated for 15mm capstan

while (digitalRead(enable_pin) == LOW) { // check status of enable toggle and loop until enabled
    read_potentiometers(); // while waiting for enable signal, read potentiometers

    refresh_LCD(); // while waiting for enable signal, update LCD display

    delay(1); // loop delay
}
delay(10);

// we don't know the actuator position at start up, so we need to use the optical proximity sensor to set home
home_actuator();

if (DEBUG == 1) {
    log_parameters();
}

void loop()
{
    while (digitalRead(enable_pin) == LOW) {  // wait in this loop if the actuator is set to disabled on the console
        average_pressure = 0;
        for (long i = 0; i <= 9; i++) {
            average_pressure += read_pressure();    // read the instantaneous pressure and increment average
            read_potentiometers();
            refresh_LCD();
            delay(100);
        }
        peak_insp_pressure = average_pressure / 10; // average 10 pressure readings to reduce noise
        refresh_LCD();
    }
read_potentiometers();

refresh_LCD();

initial_pressure = read_pressure(); // baseline pressure

start_time = millis();

// the next two commands will set the actuator speed and drive it to the inspiration position; code is blocking
stepper.setMaxSpeed(inspiration_speed); // inspiration speed has been determined by reading the pots
stepper.runToNewPosition(insp_position); // inspiration position has been determined by reading the TV pot
time_to_inpiration = millis() - start_time; // once the actuator has reached max position, record time to inspiration

//send_trigger();

digitalWrite(trigger_pin, HIGH); // digital trigger is sent, in case other devices need to monitor
peak_insp_pressure = read_pressure(); // read the instantaneous peak pressure
proximity = particleSensor.getIR(); // reading position in peak pressure
mid_position = ProximityToStepPosition(proximity);
delay(inspiration_hold); // hold max inspiration position, if desired
digitalWrite(trigger_pin, LOW);

// the next two commands will set the actuator speed and drive it to the start position; code is blocking
stepper.setMaxSpeed(expiration_speed); // expiration speed has been determined by reading the pots
stepper.runToNewPosition(start_position);

end_insp_pressure = read_pressure(); // read the instantaneous pressure when actuator has been retracted

time_to_expiration = millis() - start_time; // once the actuator has retracted to start position, record time to inspiration

refresh_LCD();

proximity = particleSensor.getIR(); // the actuator should be back at the start position, but we must check this with the proximity sensor

derived_position = ProximityToStepPosition(proximity);

if (abs(derived_position - start_position) >= error_tolerance) { // if the actuator position is not within tolerance, re-home

  lcd.clear();
  lcd.setCursor(0, 0); // set the cursor to column 0, line 0
  lcd.print("ERROR:"); // Print a message to the LCD.
  lcd.setCursor(0, 2); // set the cursor to column 0, line 0
  lcd.print("INIT-POS ERR!"); // Print a message to the LCD.
  home_actuator();
  stepper.runToNewPosition(start_position);
  correction_cycles = correction_cycles + 1; // update the count of cycles for which actuator did not return exactly to start position
}

elapsed_time = millis(); // check the lapsed time since beginning of cycle
while (elapsed_time - start_time <= cycle_period) { // wait in this loop until it is time to start
the next cycle

    end_insp_pressure = read_pressure(); // read the instantaneous pressure when
actuator has been retracted

    if (end_insp_pressure <= 0.0) {
        elapsed_time = start_time + cycle_period + 1000;
    }

    else {
        elapsed_time = millis();
    }

}

total_cycles = total_cycles + 1; // increment count of total cycles since initiation

if (mid_position <= abs(start_position - error_tolerance)){ // error condition for motor
movement

    lcd.clear();
    lcd.setCursor(0, 0); // set the cursor to column 0, line 0
    lcd.print("ERROR:"); // Print a message to the LCD.
    lcd.setCursor(0, 2); // set the cursor to column 0, line 2
    lcd.print("MOTOR STUCK ERR!"); // Print a message to the LCD.
    while(1);
}

if((peak_insp_pressure - initial_pressure <= 0) && (mid_position >= abs(start_position -
error_tolerance))){ // error condition for wire break

    lcd.clear();
lcd.setCursor(0, 0);  // set the cursor to column 0, line 0
lcd.print("ERROR:");  // Print a message to the LCD.
lcd.setCursor(0, 2);  // set the cursor to column 0, line 2
lcd.print("WIRE BREAK ERR!");  // Print a message to the LCD.
while(1);
}

if(initial_pressure < MIN_PRESSURE){
  lcd.clear();
lcd.setCursor(0, 0);  // set the cursor to column 0, line 0
lcd.print("ERROR:");  // Print a message to the LCD.
lcd.setCursor(0, 2);  // set the cursor to column 0, line 2
lcd.print("LOW PRESSURE ERR!");  // Print a message to the LCD.
delay(2000);
}

if(peak_insp_pressure > MAX_PRESSURE){
  lcd.clear();
lcd.setCursor(0, 0);  // set the cursor to column 0, line 0
lcd.print("ERROR:");  // Print a message to the LCD.
lcd.setCursor(0, 2);  // set the cursor to column 0, line 2
lcd.print("HIGH PRESSURE ERR!");  // Print a message to the LCD.
delay(2000);
}
cycle_time = elapsed_time - start_time;  // calculate the actual cycle time and compare
with requested cycle period

if ((cycle_time) > cycle_period + cycle_time_tolerance) {
    error_cycles = error_cycles + 1;  // increment count of cycles that did not achieve 
desired period

    lcd.clear();
    lcd.setCursor(0, 0); // set the cursor to column 0, line 0
    lcd.print("ERROR:");  // Print a message to the LCD.
    lcd.setCursor(0, 2); // set the cursor to column 0, line 2
    lcd.print("CYC TIME ERR!");  // Print a message to the LCD.
    delay(2000);
}

if (DEBUG == 1) {
    print_report();
}
}

int ProximityToStepPosition(float sensor_input) {

    // this routine calculates the motor step position, based on the IR reflectance signal coming from
the MAX30105

    // the calculation is based on a fit to a two-stage exponention, using five constants defined 
above

    int derived_position;

    derived_position = Plateau + SpanFast * exp(- KFast * proximity) + SpanSlow * exp(- KSlow * 
proximity);
return derived_position;
}

void home_actuator() {

    // this function uses the reflectance signal from the MAX30105 to estimate the current step position
    // it then iterates until it gets to within 15 microsteps of the home position
    // note that we are using a 1/8 step micro-stepper, so 15 steps is around 3 degrees of motor rotation
    proximity = particleSensor.getIR();
    derived_position = ProximityToStepPosition(proximity);
    while (abs(derived_position - home_position) >= 15) {
        stepper.setCurrentPosition(derived_position);
        stepper.runToNewPosition(home_position);                  // this routine iteratively moves the actuator closer and closer to home
        proximity = particleSensor.getIR();
        derived_position = ProximityToStepPosition(proximity);
    }
    stepper.setCurrentPosition(home_position);                  // when complete, the current position is updated to agree with home position
}

float getIntercept() {

    // this routine reads the pressure signal from the HX711 when no pressure is applied
    // this will compensate for atmospheric pressure fluctuations
    // average of 50 readings is taken
float pressure = 0;
for (long i = 0; i <= 49; i++) {
    long reading = scale.read();
    pressure += reading * slope;
    delay(20);
}
float offset = (pressure / 50) * -1;
return offset;
}

void read_potentiometers() {
    // this routine calculates the requested BPM, based on the potentiometer attached to analog pin
    // the readings are averaged to reduce noise
    int val = 0;
    for (long i = 0; i <= 999; i++) {
        val += analogRead(BPM_pin);    // read the BPM value from the potentiometer
    }
    val = val / 1000;    // average 1000 readings to reduce noise
    BPM = (val / 50) + 10;
    cycle_period = (60 * 1000 / BPM);
    // this routine calculates the requested TV, based on the potentiometer attached to analog pin 1
    // the readings are averaged to reduce noise
val = 0;
for (long i = 0; i <= 999; i++) {
    val += analogRead(TV_pin);  // read the TV value from the potentiometer
}
val = val / 1000;               // average 1000 readings to reduce noise

TV = ((val / 125) * 100) + 100;

stroke = Y0 + ((TV_plateau - Y0) * (1 - exp(-1 * K_TV * TV)));

insp_position = start_position + stroke;

if (insp_position >= never_exceed_position) {
    insp_position = never_exceed_position;
}

// this routine calculates the requested I/E, based on the potentiometer attached to analog pin 2
// the readings are averaged to reduce noise
val = 0;
for (long i = 0; i <= 999; i++) {
    val += analogRead(IE_pin);  // read the I/E ratio value from the potentiometer
}
val = val / 1000;               // average 1000 readings to reduce noise

switch (val / 256) {
    case 0:
        IE = 1.0;
        break;
    case 1:
IE = 0.5;
break;
case 2:
    IE = 0.33;
    break;
case 3:
    IE = 0.25;
    break;
}

inspiration_time = (cycle_period * (IE / (IE + 1) )) - 120;  /**<the value of 120ms is a correction factor*/

inspiration_speed = (stroke / inspiration_time) * 1000;  /**<inspiration speed is in pulses per second*/

if (inspiration_speed >= max_allowed_speed) {
    inspiration_speed = max_allowed_speed;  /**< don't ask the motor to go over stall speed*/
}

void refresh_LCD() {
    lcd.setCursor(0, 0);  /**< set the cursor to column 0, line 0*/
    lcd.print("BPM ");  /**< Print a message to the LCD.*/
    lcd.setCursor(4, 0);  /**< set the cursor to column 0, line 0*/
    print_LCD_int(BPM, 6);
    lcd.setCursor(0, 1);  /**< set the cursor to column 0, line 0*/
lcd.print("TV "); // Print a message to the LCD.
lcd.setCursor(4, 1); // set the cursor to column 0, line 0
print_LCD_int(TV, 6);
lcd.setCursor(0, 2); // set the cursor to column 0, line 0
lcd.print("I/E "); // Print a message to the LCD.
lcd.setCursor(4, 2); // set the cursor to column 0, line 0
print_LCD_float(IE, 4, 2);
lcd.setCursor(0, 3); // set the cursor to column 0, line 0
lcd.print("PPS "); // Print a message to the LCD.
lcd.setCursor(4, 3); // set the cursor to column 0, line 0
print_LCD_int(inspiration_speed, 6);
lcd.setCursor(10, 3); // set the cursor to column 0, line 0
lcd.print("Cyc "); // Print a message to the LCD.
lcd.setCursor(14, 3); // set the cursor to column 0, line 0
print_LCD_int(total_cycles, 6);
lcd.setCursor(10, 0); // set the cursor to column 0, line 0
lcd.print("I Pr "); // Print a message to the LCD.
lcd.setCursor(15, 0); // set the cursor to column 0, line 0
print_LCD_float(peak_insp_pressure, 5, 1);
lcd.setCursor(10, 1); // set the cursor to column 0, line 0
lcd.print("E Pr "); // Print a message to the LCD.
lcd.setCursor(15, 1); // set the cursor to column 0, line 0
print_LCD_float(end_insp_pressure, 5, 1);
void send_trigger() { // this routine will send a 1ms pulse to a digital pin, when needed to sync other devices

digitalWrite(trigger_pin, HIGH);

delay(1);

digitalWrite(trigger_pin, LOW);

delay(1);
float read_pressure() {

    // this routine reads the pressure signal from the HX711
    // pressure readings are ready every 100 ms

    float pressure;

    if (scale.is_ready()) {
        long reading = scale.read();
        pressure = reading * slope + intercept;
    }

    return pressure;
}

void log_parameters() {

    Serial.print("Error Tolerance (microsteps) ");
    Serial.println(error_tolerance);

    Serial.printf("Cycle Time Tolerance (ms )");
    Serial.println(cycle_time_tolerance);

    Serial.print("Acceleration (microsteps/s/s )");
    Serial.println(acceleration);

    Serial.print("Inspiration position (microsteps )");
    Serial.println(max_position);

    Serial.print("Expiration Position (microsteps )");
    Serial.println(start_position);

    Serial.print("Home position (microsteps )");
}
Serial.println(home_position);

Serial.print("Start position (microsteps )");

Serial.println(start_position);

Serial.print("Inspiration Speed (microsteps/s)");

Serial.println(inspiration_speed);

Serial.print("Expiration Speed (microsteps/s)");

Serial.println(expiration_speed);

Serial.print("Cycle Period (ms)");

Serial.println(cycle_period);

Serial.println(" ");

}

void print_report() {

Serial.print(total_cycles);

Serial.print(" ");

Serial.print(time_to_inpiration);

Serial.print(" ");

Serial.print(time_to_expiration);

Serial.print(" ");

Serial.print(cycle_time);

Serial.print(" ");

Serial.print(float(time_to_inpiration) / (cycle_time - time_to_inpiration));

Serial.print(" ");

Serial.print(peak_insp_pressure);
Serial.print(" ");
Serial.print(end_insp_pressure);
Serial.print(" ");
Serial.print(correction_cycles);
Serial.print(" ");
Serial.println(error_cycles);
}

static void print_LCD_int(unsigned long val, int len)
{
    char sz[32];
    sprintf(sz, "%ld", val);
    sz[len] = 0;
    for (int i = strlen(sz); i < len; ++i)
    {
        sz[i] = ' ';
    }
    if (len > 0)
    {
        sz[len - 1] = ' ';
    }
    lcd.print(sz);
}

static void print_LCD_float(float val, int len, int prec)
{
    lcd.print(val, prec);
    int vi = abs((int)val);
    int flen = prec + (val < 0.0 ? 2 : 1); // . and -
flen += vi >= 1000 ? 4 : vi >= 100 ? 3 : vi >= 10 ? 2 : 1;

for (int i = flen; i < len; ++i)
    lcd.print(' ');
}

Section B.2

Flow Sensor Codes

#include <Wire.h>

/* Inspired by :

* https://stackoverflow.com/questions/60741196/difference-i2c-sensor-reading-raspberry-pi-and-arduino

*/

#define sfm3300i2c 0x40

float volume = 0;

float air_flow;

const int integration_time = 3750;

const int sample_time = 30;

long start;

void setup() {

    Wire.begin();
    Serial.begin(115200);
while(!Serial) {} // let serial console settle

Wire.beginTransmission(sfm3300i2c);

Wire.write(0x10); // start continuous measurement

Wire.write(0x00); // command 0x1000

Wire.endTransmission();

}

void loop() {

volume = 0;

start = millis();

while ((millis() - start) < integration_time) {

volume = volume + (ReadFlow_mls() * sample_time /1000);

delay(sample_time);

}

Serial.println(volume);

}

float ReadFlow_mls(){

// this routine reads air flow from the Sensirion SFM3300-D

// flow readings are ready every 10 ms

// flow is in standard liters per minute
float flow;

if (2 == Wire.requestFrom(sfm3300i2c, 2)) { // just keep reading SLM (Standard Liter per Minute)
    uint16_t a = Wire.read(); // only two bytes need to be read
    uint8_t b = Wire.read(); // if we don't care about CRC
    a = (a<<8) | b;
    flow = ((float)a - 32768) / 7.200;  // standard ml per second
}

return flow;

}

Section B.3

MATLAB Codes

%% Flowrate to Volume Algorithm for Ventilator system
% clear data and figures
clc

clear

close all
% Initialize data

data=readmatrix('peep15bpm16tv500.csv');
% Initialize the sample time

Ts=30;  % ms
% Time vector is ms

t=0:Ts:(length(data)-1)*Ts;

% Time vector in minute

t=t/60000; % minute

% FlowRate

flow=data(:,5)';

% number of steps

N=numel(t);

% Compute Volume using Integration

for i=2:N

    % Summation of 1st and last segment

    T=flow(1)+flow(i);

    % Define Summation

    R=0;

    % Summation of all summation between 1st and last segment

    for itr = 1:i

        R=R+flow(itr);

    end

    % Integration using Trapezoidal Rule

    vol(i)=((Ts/60000)/2)*(T+2*R);

end
% Plot Flow rate
figure
subplot(211)
plot(t,flow)
grid on
xlabel('time (min)')
ylabel('Flow Rate (lit/min)')
% Plot Volume
subplot(212)
plot(t,vol+RV)
grid on
xlabel('time (min)')
ylabel('Volume (lit)')
Table 1

The recorded data for one condition i.e., 300 ml of $V_t$, 12 BPM and 1:2 I:E ratio for Test Procedure 1 as mentioned in Section 4.4 of Chapter 4 for the purpose of calculating sample time is mentioned below in Table B.1. S.No. 8 and 177 are start and end, which is one complete cycle breathing cycle.

Table B.1. Data collected for test procedure 1: 300 ml of $V_t$, 12 BPM

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Voltage</th>
<th>Current</th>
<th>Load</th>
<th>Pressure</th>
<th>Airflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11.97</td>
<td>1.53</td>
<td>0.09</td>
<td>14.68</td>
<td>-8.2</td>
</tr>
<tr>
<td>2</td>
<td>11.84</td>
<td>1.87</td>
<td>0.09</td>
<td>14.62</td>
<td>-5.07</td>
</tr>
<tr>
<td>3</td>
<td>11.92</td>
<td>1.82</td>
<td>0.11</td>
<td>14.35</td>
<td>-12</td>
</tr>
<tr>
<td>4</td>
<td>11.99</td>
<td>1.67</td>
<td>0.11</td>
<td>14.35</td>
<td>-5.3</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>1.58</td>
<td>0.12</td>
<td>14.15</td>
<td>-3.9</td>
</tr>
<tr>
<td>6</td>
<td>11.99</td>
<td>1.55</td>
<td>0.13</td>
<td>13.84</td>
<td>-6.9</td>
</tr>
<tr>
<td>7</td>
<td>12.03</td>
<td>1.55</td>
<td>0.15</td>
<td>13.98</td>
<td>-8.87</td>
</tr>
<tr>
<td>8</td>
<td>12.54</td>
<td>0.67</td>
<td>0.15</td>
<td>13.72</td>
<td>-7.63</td>
</tr>
<tr>
<td>9</td>
<td>12.55</td>
<td>0.64</td>
<td>0.15</td>
<td>13.5</td>
<td>-4.63</td>
</tr>
<tr>
<td>10</td>
<td>12.56</td>
<td>0.64</td>
<td>0.15</td>
<td>13.47</td>
<td>-3.4</td>
</tr>
<tr>
<td>11</td>
<td>12.56</td>
<td>0.64</td>
<td>0.15</td>
<td>13.47</td>
<td>-1.5</td>
</tr>
<tr>
<td>12</td>
<td>12.57</td>
<td>0.64</td>
<td>0.15</td>
<td>13.46</td>
<td>-2.93</td>
</tr>
<tr>
<td>13</td>
<td>12.57</td>
<td>0.64</td>
<td>0.15</td>
<td>13.4</td>
<td>-3.27</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-----</td>
<td>----</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>14.</td>
<td>12.57</td>
<td>0.64</td>
<td>0.15</td>
<td>13.39</td>
<td>-2.2</td>
</tr>
<tr>
<td>15.</td>
<td>12.58</td>
<td>0.65</td>
<td>0.15</td>
<td>13.34</td>
<td>-1.9</td>
</tr>
<tr>
<td>16.</td>
<td>12.58</td>
<td>0.64</td>
<td>0.15</td>
<td>13.31</td>
<td>-1.8</td>
</tr>
<tr>
<td>17.</td>
<td>12.58</td>
<td>0.64</td>
<td>0.15</td>
<td>13.26</td>
<td>-1.83</td>
</tr>
<tr>
<td>18.</td>
<td>12.58</td>
<td>0.65</td>
<td>0.15</td>
<td>13.21</td>
<td>-1.83</td>
</tr>
<tr>
<td>19.</td>
<td>12.58</td>
<td>0.64</td>
<td>0.15</td>
<td>13.16</td>
<td>-1.8</td>
</tr>
<tr>
<td>20.</td>
<td>12.58</td>
<td>0.64</td>
<td>0.15</td>
<td>13.12</td>
<td>-1.83</td>
</tr>
<tr>
<td>21.</td>
<td>12.58</td>
<td>0.64</td>
<td>0.15</td>
<td>13.06</td>
<td>-1.8</td>
</tr>
<tr>
<td>22.</td>
<td>12.59</td>
<td>0.64</td>
<td>0.15</td>
<td>13.01</td>
<td>-1.83</td>
</tr>
<tr>
<td>23.</td>
<td>12.59</td>
<td>0.64</td>
<td>0.15</td>
<td>12.96</td>
<td>-1.83</td>
</tr>
<tr>
<td>24.</td>
<td>12.59</td>
<td>0.64</td>
<td>0.15</td>
<td>12.92</td>
<td>-1.8</td>
</tr>
<tr>
<td>25.</td>
<td>12.59</td>
<td>0.64</td>
<td>0.15</td>
<td>12.87</td>
<td>-1.8</td>
</tr>
<tr>
<td>26.</td>
<td>12.59</td>
<td>0.64</td>
<td>0.15</td>
<td>12.81</td>
<td>-1.8</td>
</tr>
<tr>
<td>27.</td>
<td>12.59</td>
<td>0.64</td>
<td>0.15</td>
<td>12.77</td>
<td>-1.8</td>
</tr>
<tr>
<td>28.</td>
<td>12.59</td>
<td>0.64</td>
<td>0.15</td>
<td>12.72</td>
<td>-1.8</td>
</tr>
<tr>
<td>29.</td>
<td>12.59</td>
<td>0.64</td>
<td>0.16</td>
<td>12.67</td>
<td>-1.8</td>
</tr>
<tr>
<td>30.</td>
<td>12.59</td>
<td>0.64</td>
<td>0.15</td>
<td>12.64</td>
<td>-1.8</td>
</tr>
<tr>
<td>31.</td>
<td>12.59</td>
<td>0.65</td>
<td>0.15</td>
<td>12.58</td>
<td>-1.8</td>
</tr>
<tr>
<td>32.</td>
<td>12.59</td>
<td>0.64</td>
<td>0.15</td>
<td>12.53</td>
<td>-1.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>33</td>
<td>12.59</td>
<td>0.64</td>
<td>0.15</td>
<td>12.5</td>
<td>-1.77</td>
</tr>
<tr>
<td>34</td>
<td>12.6</td>
<td>0.64</td>
<td>0.15</td>
<td>12.44</td>
<td>-1.77</td>
</tr>
<tr>
<td>35</td>
<td>12.6</td>
<td>0.64</td>
<td>0.15</td>
<td>12.39</td>
<td>-1.77</td>
</tr>
<tr>
<td>36</td>
<td>12.6</td>
<td>0.65</td>
<td>0.15</td>
<td>12.39</td>
<td>-1.77</td>
</tr>
<tr>
<td>37</td>
<td>12.6</td>
<td>0.64</td>
<td>0.15</td>
<td>12.31</td>
<td>-1.77</td>
</tr>
<tr>
<td>38</td>
<td>12.59</td>
<td>0.63</td>
<td>0.15</td>
<td>12.25</td>
<td>-1.77</td>
</tr>
<tr>
<td>39</td>
<td>12.6</td>
<td>0.64</td>
<td>0.15</td>
<td>12.2</td>
<td>-1.77</td>
</tr>
<tr>
<td>40</td>
<td>12.6</td>
<td>0.64</td>
<td>0.16</td>
<td>12.16</td>
<td>-1.73</td>
</tr>
<tr>
<td>41</td>
<td>12.6</td>
<td>0.64</td>
<td>0.16</td>
<td>12.11</td>
<td>-1.73</td>
</tr>
<tr>
<td>42</td>
<td>12.6</td>
<td>0.65</td>
<td>0.15</td>
<td>12.06</td>
<td>-1.73</td>
</tr>
<tr>
<td>43</td>
<td>12.6</td>
<td>0.64</td>
<td>0.15</td>
<td>12.02</td>
<td>-1.73</td>
</tr>
<tr>
<td>44</td>
<td>12.6</td>
<td>0.64</td>
<td>0.15</td>
<td>11.97</td>
<td>-1.7</td>
</tr>
<tr>
<td>45</td>
<td>12.6</td>
<td>0.64</td>
<td>0.16</td>
<td>11.92</td>
<td>-1.73</td>
</tr>
<tr>
<td>46</td>
<td>12.6</td>
<td>0.65</td>
<td>0.16</td>
<td>11.87</td>
<td>-1.7</td>
</tr>
<tr>
<td>47</td>
<td>12.6</td>
<td>0.65</td>
<td>0.16</td>
<td>11.84</td>
<td>-1.7</td>
</tr>
<tr>
<td>48</td>
<td>12.6</td>
<td>0.64</td>
<td>0.16</td>
<td>11.78</td>
<td>-1.7</td>
</tr>
<tr>
<td>49</td>
<td>12.6</td>
<td>0.64</td>
<td>0.16</td>
<td>11.73</td>
<td>-1.7</td>
</tr>
<tr>
<td>50</td>
<td>12.6</td>
<td>0.64</td>
<td>0.16</td>
<td>11.69</td>
<td>-1.7</td>
</tr>
<tr>
<td>51</td>
<td>12.6</td>
<td>0.64</td>
<td>0.16</td>
<td>11.64</td>
<td>-1.67</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>52.</td>
<td>12.6</td>
<td>0.64</td>
<td>0.16</td>
<td>11.59</td>
<td>-1.7</td>
</tr>
<tr>
<td>53.</td>
<td>12.6</td>
<td>0.64</td>
<td>0.16</td>
<td>11.56</td>
<td>-1.7</td>
</tr>
<tr>
<td>54.</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>11.51</td>
<td>-1.7</td>
</tr>
<tr>
<td>55.</td>
<td>12.6</td>
<td>0.64</td>
<td>0.16</td>
<td>11.46</td>
<td>-1.67</td>
</tr>
<tr>
<td>56.</td>
<td>12.6</td>
<td>0.65</td>
<td>0.16</td>
<td>11.41</td>
<td>-1.67</td>
</tr>
<tr>
<td>57.</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>11.37</td>
<td>-1.67</td>
</tr>
<tr>
<td>58.</td>
<td>12.61</td>
<td>0.63</td>
<td>0.16</td>
<td>11.32</td>
<td>-1.67</td>
</tr>
<tr>
<td>59.</td>
<td>12.61</td>
<td>0.65</td>
<td>0.16</td>
<td>11.27</td>
<td>-1.63</td>
</tr>
<tr>
<td>60.</td>
<td>12.6</td>
<td>0.64</td>
<td>0.16</td>
<td>11.24</td>
<td>-1.67</td>
</tr>
<tr>
<td>61.</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>11.18</td>
<td>-1.63</td>
</tr>
<tr>
<td>62.</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>11.14</td>
<td>-1.63</td>
</tr>
<tr>
<td>63.</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>11.11</td>
<td>-1.63</td>
</tr>
<tr>
<td>64.</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>11.06</td>
<td>-1.63</td>
</tr>
<tr>
<td>65.</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>11</td>
<td>-1.63</td>
</tr>
<tr>
<td>66.</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>10.97</td>
<td>-1.63</td>
</tr>
<tr>
<td>67.</td>
<td>12.6</td>
<td>0.64</td>
<td>0.16</td>
<td>10.92</td>
<td>-1.63</td>
</tr>
<tr>
<td>68.</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>10.87</td>
<td>-1.63</td>
</tr>
<tr>
<td>69.</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>10.82</td>
<td>-1.6</td>
</tr>
<tr>
<td>70.</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>10.78</td>
<td>-1.63</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>-----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>71</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>10.73</td>
<td>-1.6</td>
</tr>
<tr>
<td>72</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>10.68</td>
<td>-1.6</td>
</tr>
<tr>
<td>73</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>10.63</td>
<td>-1.6</td>
</tr>
<tr>
<td>74</td>
<td>12.61</td>
<td>0.63</td>
<td>0.16</td>
<td>10.58</td>
<td>-1.6</td>
</tr>
<tr>
<td>75</td>
<td>12.61</td>
<td>0.63</td>
<td>0.16</td>
<td>10.52</td>
<td>-1.6</td>
</tr>
<tr>
<td>76</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>10.49</td>
<td>-1.57</td>
</tr>
<tr>
<td>77</td>
<td>12.61</td>
<td>0.64</td>
<td>0.15</td>
<td>10.44</td>
<td>-1.57</td>
</tr>
<tr>
<td>78</td>
<td>12.61</td>
<td>0.64</td>
<td>0.15</td>
<td>10.39</td>
<td>-1.57</td>
</tr>
<tr>
<td>79</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>10.34</td>
<td>-1.57</td>
</tr>
<tr>
<td>80</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>10.29</td>
<td>-1.57</td>
</tr>
<tr>
<td>81</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>10.24</td>
<td>-1.53</td>
</tr>
<tr>
<td>82</td>
<td>12.61</td>
<td>0.65</td>
<td>0.16</td>
<td>10.18</td>
<td>-1.57</td>
</tr>
<tr>
<td>83</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>10.15</td>
<td>-1.53</td>
</tr>
<tr>
<td>84</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>10.08</td>
<td>-1.53</td>
</tr>
<tr>
<td>85</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>10.01</td>
<td>-1.5</td>
</tr>
<tr>
<td>86</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>9.95</td>
<td>-1.5</td>
</tr>
<tr>
<td>87</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>9.88</td>
<td>-1.5</td>
</tr>
<tr>
<td>88</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>9.81</td>
<td>-1.5</td>
</tr>
<tr>
<td>89</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>9.77</td>
<td>-1.5</td>
</tr>
<tr>
<td></td>
<td>12.62</td>
<td>0.63</td>
<td>0.16</td>
<td>9.69</td>
<td>-1.47</td>
</tr>
<tr>
<td>---</td>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>91.</td>
<td>12.62</td>
<td>0.64</td>
<td>0.16</td>
<td>9.63</td>
<td>-1.43</td>
</tr>
<tr>
<td>92.</td>
<td>12.62</td>
<td>0.64</td>
<td>0.16</td>
<td>9.43</td>
<td>-1.43</td>
</tr>
<tr>
<td>93.</td>
<td>12.61</td>
<td>0.64</td>
<td>0.16</td>
<td>9.27</td>
<td>-1.4</td>
</tr>
<tr>
<td>94.</td>
<td>12.62</td>
<td>0.64</td>
<td>0.16</td>
<td>9.05</td>
<td>-1.4</td>
</tr>
<tr>
<td>95.</td>
<td>12.62</td>
<td>0.65</td>
<td>0.16</td>
<td>8.84</td>
<td>-1.37</td>
</tr>
<tr>
<td>96.</td>
<td>12.62</td>
<td>0.64</td>
<td>0.16</td>
<td>8.7</td>
<td>-1.33</td>
</tr>
<tr>
<td>97.</td>
<td>12.1</td>
<td>1.19</td>
<td>0.16</td>
<td>8.51</td>
<td>-1.33</td>
</tr>
<tr>
<td>98.</td>
<td>12.02</td>
<td>1.6</td>
<td>0.16</td>
<td>8.33</td>
<td>-1.6</td>
</tr>
<tr>
<td>99.</td>
<td>12.01</td>
<td>1.64</td>
<td>0.2</td>
<td>8.21</td>
<td>-1.03</td>
</tr>
<tr>
<td>100.</td>
<td>12.02</td>
<td>1.66</td>
<td>0.22</td>
<td>8.04</td>
<td>-0.93</td>
</tr>
<tr>
<td>101.</td>
<td>12</td>
<td>1.64</td>
<td>0.22</td>
<td>8.04</td>
<td>-0.77</td>
</tr>
<tr>
<td>102.</td>
<td>12.01</td>
<td>1.68</td>
<td>0.27</td>
<td>7.9</td>
<td>-0.5</td>
</tr>
<tr>
<td>103.</td>
<td>11.97</td>
<td>1.7</td>
<td>0.31</td>
<td>7.76</td>
<td>-0.37</td>
</tr>
<tr>
<td>104.</td>
<td>11.99</td>
<td>1.69</td>
<td>0.36</td>
<td>7.69</td>
<td>-0.03</td>
</tr>
<tr>
<td>105.</td>
<td>11.98</td>
<td>1.68</td>
<td>0.39</td>
<td>7.69</td>
<td>2.67</td>
</tr>
<tr>
<td>106.</td>
<td>11.98</td>
<td>1.67</td>
<td>0.44</td>
<td>7.69</td>
<td>1.6</td>
</tr>
<tr>
<td>107.</td>
<td>11.98</td>
<td>1.71</td>
<td>0.49</td>
<td>8.17</td>
<td>3.53</td>
</tr>
<tr>
<td>108.</td>
<td>11.99</td>
<td>1.68</td>
<td>0.52</td>
<td>8.69</td>
<td>4.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>109</td>
<td>11.97</td>
<td>1.7</td>
<td>0.57</td>
<td>9.12</td>
<td>4.57</td>
</tr>
<tr>
<td>110</td>
<td>11.96</td>
<td>1.68</td>
<td>0.63</td>
<td>9.86</td>
<td>5.7</td>
</tr>
<tr>
<td>111</td>
<td>11.95</td>
<td>1.72</td>
<td>0.68</td>
<td>10.61</td>
<td>6.77</td>
</tr>
<tr>
<td>112</td>
<td>11.96</td>
<td>1.68</td>
<td>0.73</td>
<td>10.98</td>
<td>7.37</td>
</tr>
<tr>
<td>113</td>
<td>11.96</td>
<td>1.71</td>
<td>0.76</td>
<td>11.38</td>
<td>8.03</td>
</tr>
<tr>
<td>114</td>
<td>11.97</td>
<td>1.69</td>
<td>0.81</td>
<td>11.71</td>
<td>8.27</td>
</tr>
<tr>
<td>115</td>
<td>11.95</td>
<td>1.71</td>
<td>0.85</td>
<td>12.02</td>
<td>8.63</td>
</tr>
<tr>
<td>116</td>
<td>11.96</td>
<td>1.7</td>
<td>0.9</td>
<td>12.22</td>
<td>8.93</td>
</tr>
<tr>
<td>117</td>
<td>11.95</td>
<td>1.73</td>
<td>0.93</td>
<td>12.55</td>
<td>9.17</td>
</tr>
<tr>
<td>118</td>
<td>11.95</td>
<td>1.71</td>
<td>0.98</td>
<td>12.86</td>
<td>9.5</td>
</tr>
<tr>
<td>119</td>
<td>11.93</td>
<td>1.74</td>
<td>1.03</td>
<td>13.05</td>
<td>9.57</td>
</tr>
<tr>
<td>120</td>
<td>11.94</td>
<td>1.71</td>
<td>1.08</td>
<td>13.36</td>
<td>9.63</td>
</tr>
<tr>
<td>121</td>
<td>11.93</td>
<td>1.73</td>
<td>1.12</td>
<td>13.67</td>
<td>9.43</td>
</tr>
<tr>
<td>122</td>
<td>11.94</td>
<td>1.71</td>
<td>1.17</td>
<td>13.85</td>
<td>9.7</td>
</tr>
<tr>
<td>123</td>
<td>11.93</td>
<td>1.75</td>
<td>1.23</td>
<td>14.12</td>
<td>9.4</td>
</tr>
<tr>
<td>124</td>
<td>11.91</td>
<td>1.75</td>
<td>1.3</td>
<td>14.38</td>
<td>9.2</td>
</tr>
<tr>
<td>125</td>
<td>11.93</td>
<td>1.77</td>
<td>1.34</td>
<td>14.54</td>
<td>9.2</td>
</tr>
<tr>
<td>126</td>
<td>11.9</td>
<td>1.76</td>
<td>1.41</td>
<td>14.8</td>
<td>8.97</td>
</tr>
<tr>
<td>127</td>
<td>11.92</td>
<td>1.76</td>
<td>1.48</td>
<td>15.04</td>
<td>8.67</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>129</td>
<td>130</td>
<td>131</td>
<td>132</td>
</tr>
<tr>
<td>----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td></td>
<td>11.91</td>
<td>11.91</td>
<td>11.89</td>
<td>11.9</td>
<td>11.89</td>
</tr>
<tr>
<td></td>
<td>1.75</td>
<td>1.77</td>
<td>1.8</td>
<td>1.79</td>
<td>1.81</td>
</tr>
<tr>
<td></td>
<td>1.52</td>
<td>1.59</td>
<td>1.66</td>
<td>1.75</td>
<td>1.83</td>
</tr>
<tr>
<td></td>
<td>8.9</td>
<td>9.37</td>
<td>10.03</td>
<td>11.03</td>
<td>12.17</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>166</td>
<td>12.07</td>
<td>1.46</td>
<td>0.13</td>
<td>14.67</td>
<td>-12.37</td>
</tr>
<tr>
<td>167</td>
<td>12.08</td>
<td>1.49</td>
<td>0.11</td>
<td>14.49</td>
<td>-11.8</td>
</tr>
<tr>
<td>168</td>
<td>12.07</td>
<td>1.48</td>
<td>0.1</td>
<td>14.39</td>
<td>-10.43</td>
</tr>
<tr>
<td>169</td>
<td>12</td>
<td>1.48</td>
<td>0.08</td>
<td>14.24</td>
<td>-9.7</td>
</tr>
<tr>
<td>170</td>
<td>11.84</td>
<td>1.82</td>
<td>0.08</td>
<td>14.12</td>
<td>-8.83</td>
</tr>
<tr>
<td>171</td>
<td>11.89</td>
<td>1.84</td>
<td>0.1</td>
<td>14.05</td>
<td>-7.37</td>
</tr>
<tr>
<td>172</td>
<td>11.96</td>
<td>1.74</td>
<td>0.1</td>
<td>14.01</td>
<td>-5.8</td>
</tr>
<tr>
<td>173</td>
<td>12.02</td>
<td>1.63</td>
<td>0.12</td>
<td>14.01</td>
<td>-3.27</td>
</tr>
<tr>
<td>174</td>
<td>12.01</td>
<td>1.58</td>
<td>0.12</td>
<td>14.09</td>
<td>-3.7</td>
</tr>
<tr>
<td>175</td>
<td>12.01</td>
<td>1.56</td>
<td>0.14</td>
<td>14.09</td>
<td>-6.43</td>
</tr>
<tr>
<td>176</td>
<td>12.52</td>
<td>1</td>
<td>0.14</td>
<td>13.99</td>
<td>-6.53</td>
</tr>
<tr>
<td>177</td>
<td>12.54</td>
<td>0.64</td>
<td>0.15</td>
<td>13.61</td>
<td>-4.73</td>
</tr>
<tr>
<td>178</td>
<td>12.55</td>
<td>0.64</td>
<td>0.15</td>
<td>13.58</td>
<td>-2.53</td>
</tr>
<tr>
<td>179</td>
<td>12.56</td>
<td>0.64</td>
<td>0.15</td>
<td>13.55</td>
<td>-2.37</td>
</tr>
<tr>
<td>180</td>
<td>12.56</td>
<td>0.63</td>
<td>0.15</td>
<td>13.48</td>
<td>-3.17</td>
</tr>
<tr>
<td>181</td>
<td>12.56</td>
<td>0.64</td>
<td>0.15</td>
<td>13.44</td>
<td>-3.33</td>
</tr>
<tr>
<td>182</td>
<td>12.57</td>
<td>0.64</td>
<td>0.15</td>
<td>13.39</td>
<td>-2.07</td>
</tr>
<tr>
<td>183</td>
<td>12.57</td>
<td>0.65</td>
<td>0.15</td>
<td>13.36</td>
<td>-1.97</td>
</tr>
<tr>
<td>184</td>
<td>12.57</td>
<td>0.64</td>
<td>0.15</td>
<td>13.33</td>
<td>-1.93</td>
</tr>
</tbody>
</table>
Curriculum Vitae

Name: Heena Shrestha

Post-secondary Education and Degrees: Birla Institute Technology and Science Pilani, Dubai Campus, Dubai, United Arab Emirates 2014–2018 B.E.Sc. Electrical and Electronics Engineering

The University of Western Ontario, London, Ontario, Canada 2019–2021 M.E.Sc Electrical and Computer Engineering

Related Work Experience: Teaching Assistant ECE 4460B – Real-time Embedded Control System

Graduate Research Student The University of Western Ontario 2020–2021