Department of Electrical & Electronics Engineering

A Project Report on

Simulation Study of Cloud Connected Non-Invasive Ventilator with Interactive Web Portal

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering in Electrical and Electronics Engineering

By

ANUSHUA BANERJEE	**1MS17EE008**
KUSUMA R	**1MS17EE032**
SIMRAN A BATHIJA	**1MS17EE056**
TEJASWINI K	**1MS17EE059**

Under the guidance of

SRI. VICTOR GEORGE

Assistant Professor

EEE Dept, MSRIT

M S RAMAIAH INSTITUTE OF TECHNOLOGY

(Autonomous Institute, Affiliated To VTU)

Bengaluru – 560054

June 2021

Department of Electrical & Electronics Engineering

CERTIFICATE

Certified that the project work entitled **"Simulation study of cloud connected non-invasive ventilator with interactive web portal"** is carried out by **Anushua Banerjee, 1MS17EE008, Kusuma R, 1MS17EE032, Simran A Bathija, 1MS17EE056, Tejaswini K, 1MS17EE059** bonafide students of M **S Ramaiah Institute Of Technology, Bengaluru** in partial fulfillment for the award of **Bachelor of Engineering in Electrical and Electronics Engineering** of Visvesvaraya Technological University, Belagavi during the year **2020-21**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Project Guide	**Head of the Department**
Sri. Victor George	**Dr. Pradipkumar Dixit**

External Examiners

Name of Examiners: **Signature with Date**

1.

2.

RAMAIAH Institute of Technology

Department of Electrical & Electronics Engineering

DECLARATION

We, hereby, declare that the entire work embodied in this project report has been carried out by us at M S Ramaiah Institute of Technology, Bengaluru, under the supervision of **Sri. Victor George.** This report has not been submitted in part or full for the award of any diploma or degree of this or any other University.

Anushua Banerjee **1MS17EE008**

Kusuma R **1MS17EE032**

Simran A Bathija **1MS17EE056**

Tejaswini K **1MS17EE059**

ACKNOWLEDGEMENT

We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We would like to express our profound gratitude to **Dr. N.V.R Naidu**, Principal, RIT, Bengaluru, for providing us with the opportunity to explore our potential. We extend our heartfelt gratitude to **Dr. Pradipkumar Dixit**, Head of the Department of EEE, RIT, Bengaluru for constant support and guidance. We also thank our project coordinator **Smt. Dawnee**, Assistant Professor, Department EEE, RIT, Bengaluru for her immense support and encouragement. We would also thank our teaching and non-teaching staff of Department EEE, RIT, Bengaluru for their valuable inputs and support.

We wholeheartedly thank our project guide **Sri. Victor George**, Assistant Professor, Department EEE, RIT, Bengaluru for providing us with the confidence and strength to overcome every obstacle at each step of the project and inspiring us to do the best of our potential. We also thank him for his direction and insight during the project.

We would like to thank the teaching and non-teaching staff members of the Department of Electrical and Electronics Engineering for providing us with the skills and knowledge which we applied in our project.

Finally, we would express our gratitude to our parents, seniors and friends for their constant support during the course of work.

Anushua Banerjee

Kusuma R

Simran A Bathija

Tejaswini K

ABSTRACT

Ventilators have become the need of the hour in view of the pandemic COVID-19. Hospitals all over the world have faced an acute shortage of the same. Ventilators allow a person to breathe through assistance given to the lungs in order to inhale and exhale. It is when the patient suffers from respiratory distress that support has to be administered through the airway in the form of a tube that goes down the trachea- thereby termed as Invasive Ventilation. A ventilator, in general has two functions- supplying of oxygen rich air to the respiratory system and removal of carbon dioxide from the same system- the former is known as oxygenation and the latter is termed ventilation. Non-Invasive Ventilation (NIV) involves offering breathing support in the form of a mask which can be a nasal or a face mask or a helmet. Hence an endotracheal airway isn't required. This project offers to indicate an availability of ventilators through the usage of a simulated ventilator model on Simulink conveying information to ThingSpeak, an IoT based analytics platform from which exported data in csv format can be used to check availability of ventilators through a developed website.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

A ventilator is a machine that aids in the breathing process in human beings. It helps in pumping oxygen in and out of the lungs. In Non-Invasive Ventilation (NIV) ventilator support is provided without using an invasive artificial airway which makes it more flexible.

A ventilator supplies airflow into the lungs during inspiration and removes air from the lungs during expiration. Airflow from a ventilator is controlled by an inspiratory flow control valve and expiratory flow control valve. Ventilators provide a mixture of air and oxygen to the patients with pressure and flow controls.

Helmet based ventilator uses a helmet that is transparent and made of latex-free polyvinyl chloride (PVC). It is supported over a sturdy metallic ring to which is attached a soft collar of latex or silicon that sits around the neck, sealing the helmet to prevent any air leakage. It provides pressure support ventilation (PSV) to patients with acute respiratory disorder and is therefore more effective when compared to the conventional face mask.

Over the years, a vast range of simulator devices has been utilized for extensive analysis of the respiratory system through simulation and computational modeling. It plays a significant role in guiding medical professionals to diagnose the varying conditions that affect the patient. Furthermore, the results of the simulation and models from these simulators are of great educational purpose in providing detailed knowledge of the course of treatment and therapy, assessment of state-of-the-art medical instruments, and understanding the procedures that require authorization and several trials.

This is of utmost priority to avoid coming to any inaccurate conclusions which can be fatal to the patient. In this project, the working of the ventilator is analyzed using the MATLAB/Simulink software. In this, the base of the lung structure is represented as a mechanical model whose parameters are fed into the model and assessed based on the patient's medical conditions

1.2 LITERATURE REVIEW

Literature review was conducted around papers having information about helmet-based ventilators as well as those conducting analysis on models of ventilators on a particular software.

Helmet based non-invasive ventilators are getting more attention in recent times. One example of this ventilators uses a direct current fan blower as a source of the air and oxygen mixture to provide good air flow and pressure.[1] The design consists of inlet and outlet pipes fitted with two pressure sensors, two viral filters to contain severe acute respiratory syndrome coronavirus 2(SARS-CoV-2) and a positive end-expiratory pressure (PEEP) valve at the outlet.

Along with the high-pressure blower, they consist of a driver, two pressure transducers and an Arduino Nano controller.[2] A digital display informs the where pressure and flow were the output measured continuously at the outlet of the ventilator. This is taken as the output and is further fed into controller. Additionally, a custom-made code is used to detect inspirations and expirations. Accordingly, it triggers the inspiratory and expiratory pressures generated by blower. The performance is evaluated using bench test and volunteers. The advantages of using this is that, it’s low-cost, easily assembled and can operate in spontaneous timed out mode.

To study the ventilator parameters, it models a simple ICU ventilator circuit [3] that intends to use a rubber test lung bag which represents a human lung in volume-controlled ventilator (VCV) signal mode. The rubber test lung bag is used to evaluate three mechanical translation components, i.e., resistance, elastance and volume. Gas pressure and gas flow both in and out of this test lung bag during inspiration and expiration phase are monitored using two sensors in the sensor subsystem module. This type of mathematical model acts as a simple tool in testing and studying the ventilator parameters like Tidal volume, airway pressure and flow rate.

To simulate these results, a Simulink model of a ventilator is further developed [4]. It is useful in studying the effect of the different modes of ventilation on the respiratory system. PCV or Pressure Controlled Mode of ventilation has been used in this model. A complete model is obtained by combining the mathematical model of the PCV signal with that of the lung. Parameters such as resistances and compliances are measured through addition of resistors and capacitors in the model. Input variables of the PCV signals have parameters such as inspiratory pressure to which changes are made to observe the results.

The pressure gradient in the human respiratory tract [5] mentions how during inhaling and exhaling, there is a resistance to the flow of air caused by the airways in the respiratory tract. There is a pressure slope present, which is necessary to cause the transportation of air from the nasal passage to the pulmonary alveoli. In order to understand the amount of aerosol deposited in the respiratory tract, this slope calculation is needed. Based on this, by defining the boundary conditions, simulations are carried using a software tool called Computational Fluid Dynamics (CFD)

1.2.1 LITERATURE REVIEW SUMMARY:

Based on the literature survey conducted, it was derived that the proposed method of the non-invasive ventilators in the papers, were not able to collect and relay data of patient in real-time. It means that, ventilator parameters could not be communicated in real-time. They did not attend to the patients on a critical need basis, that shows the availability of ventilators for effective ventilator distribution. Hence, the proposed prototype aims to have the following features:

- It will convey “real-time” data about patients such as the airway pressure, flow rate and tidal volume so that the most critical ones get immediate care and are tended to with the ventilators
- The patient data can be stored on a database where will be accessible on a cloud platform
- Additionally, the difference in pressure level inside the helmet of the ventilator will be communicated to the platform
- The stored database will contain the real-time data of the patient’s condition which can be exported and used for further medical analysis
- This data will be displayed a website to show the availability in emergency situations and will vary in the availability according to the patient’s criticality. It is useful for better deployment of the ventilators which ensures efficient patient management

1.3 PROBLEM STATEMENT

The proposed work is ‘To develop a Simulink model of a non-invasive ventilator capable of communicating with the cloud to ensure efficient management through a website’.

1.4 PROJECT OBJECTIVES

The objectives of this project are listed below

1. To design a 3D model of a Non-Invasive Helmet Ventilator using CAD Fusion360
2. To observe total deformation and equivalent stress in the helmet ventilator model using Ansys software
3. To observe parameters like volume, airway pressure and flow rate for different tidal volumes and pressure losses within the helmet using a Simulink model
4. To develop a website to show availability of ventilators using the data from Thinkspeak cloud platform

1.5 PROPOSED METHODOLOGY

Fig 1.1 shows the block diagram of the method of implementation

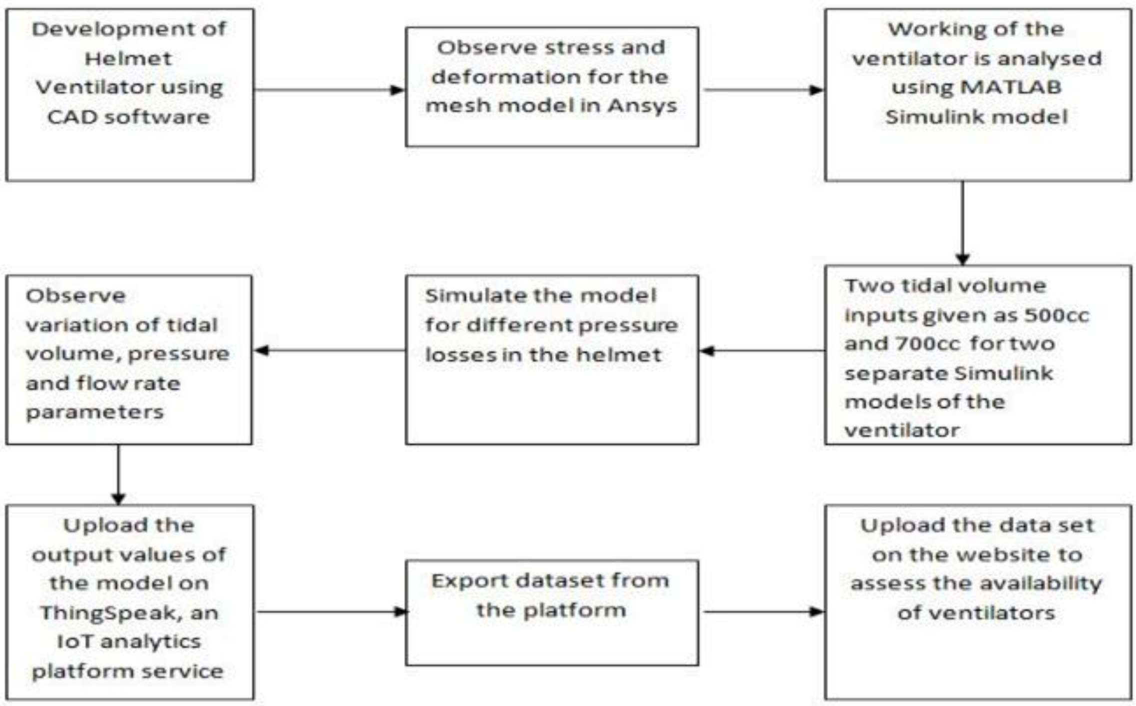


Fig. 1.1 Methodology Block Diagram

A non-invasive helmet ventilator is designed as a CAD model using Fusion 360 software. The model of the ventilator prototype is designed according to the premeditated dimensions of each of the components. By developing such a kind of accurate model design, the analysis of the ventilator working will be conducted smoothly.

The prototype is further developed as a mesh model in 3D using ANSYS simulation software. In this software, the stress and deformation of the mesh model is analysed using atmospheric pressure value.

The working of the ventilator is analysed using the MATLAB/Simulink model. Various parameters like tidal volume, airway pressure, and flow rate are obtained from the model which acts as the inputs to the lung model. Two cases of the nature of outputs are simulated and observed based on two input values of the tidal volume, i.e.,500cc and 700cc. Based on these values, the difference in the pressure inside the helmet hood of the ventilator is assessed. The difference is pressure inside the hood of the helmet is with respect to the pressure drop as soon as the patient completes one cycle of breathing.

The muscle pressure force causes the expansion of the lung volume, which lowers the lung pressure to draw in air from the helmet. When the patient breathes the oxygen, there is tightening of the diaphragm and other muscles which inhales the air into the lungs. The lung ventilator takes air from the room, and delivers the sterilised air(by humidifying and heating in the tubes) to the patient. Based on the predefined values for the three parameters of tidal volume, airway pressure and flow rate, the pressure level of the air is controlled volumetrically with sensors being attached to the outlet pipe.

There is an IoT block module of ThingSpeak connected to the scopes of all the output parameters in the model. This will communicate the scope data in real time from the ventilator to the ThingSpeak cloud platform. They are stored on cloud and will be monitored using an external monitor which will display the data in the form of simulations. Finally, the dataset from the platform is then exported onto the website which can be used to manage the ventilators in a better way and assessing the availability of ventilators.

1.6 ORGANISATION OF THE REPORT

The chapters in this report are segregated as follows:

Chapter 1 deals with the brief introduction into the objectives and implementation of this project

Chapter 2 explains the design and computations of the software's used for the CAD model of the prototype and the simulation

Chapter 3 details the methodology and implementation of the MATALB Simulink model

Chapter 4 discusses the results obtained after analysis

Chapter 5 finally concludes the work and discusses the future scope of it.

CHAPTER 2

DESIGN AND COMPUTATIONAL CONSIDERATIONS OF THE HELMET VENTILATOR

The 3D model of a non-invasive helmet ventilator was designed with the help of Autodesk Fusion360 and the total deformation and equivalent stress was observed using Ansys.

2.1 CAD MODEL

CAD or Computer-aided design is a technology to aid in creation, analysis and optimization of a design. It uses computer software to create technical drawings. CAD can be used to design models in either a two-dimensional space or a three-dimensional space. It is an important industrial art with applications across various fields.

Fusion 360 is a cloud-based CAD tool created by Autodesk. It can connect the entire product development into a cloud-based platform. Fusion 360 focuses on modeling in a 3D space.

A 3D CAD model was created using Autodesk Fusion 360. It consists of a transparent dome shape helmet made of PVC supported over a silicone neck seal and a neck ring. The silicone neck seal prevents leakage of air from the helmet. It has two port provisions made for inlet and outlet pipes. Oxygen enters the helmet from the inlet pipe. The outlet pipe has a PEEP valve and viral filters attached to it.

The PEEP valve helps maintain the Positive End Expiratory Pressure (PEEP) and prevents the lung from collapsing. The viral filter helps contain the virus within the helmet. Fig. 2.1 shows the 3D model of the CAD prototype model designed using Fusion 360.

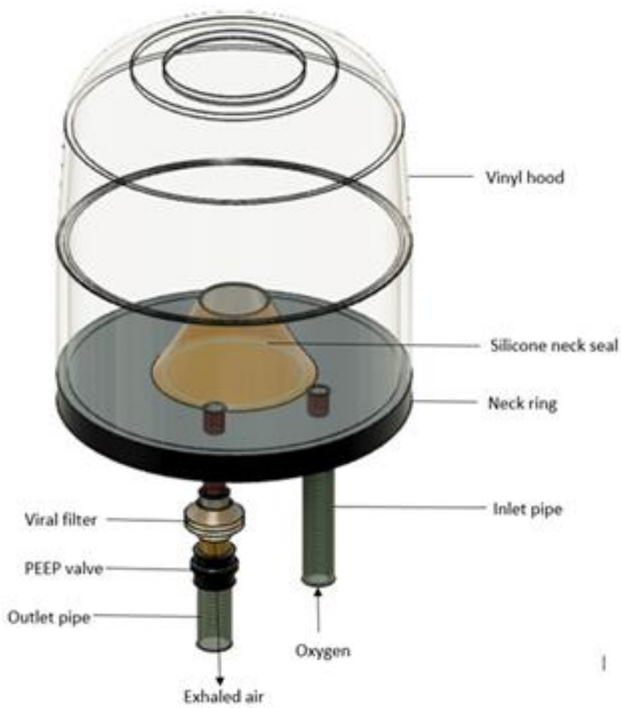


Fig. 2.1 CAD model of prototype

The various views of the CAD model along with the dimensions are indicated in the below figure. The diameter of the outer neck ring is 300mm, the diameter of the inner circle of the neck ring is 194mm to which the neck seal is connected. The height of the helmet from the neck ring is 306.812mm.

The model was then exported from Fusion 360 in an IGES format to observe total deformation and equivalent stress of the helmet using Ansys. Fig. 2.2 shows the different views of the CAD model prototype with the various dimensions.

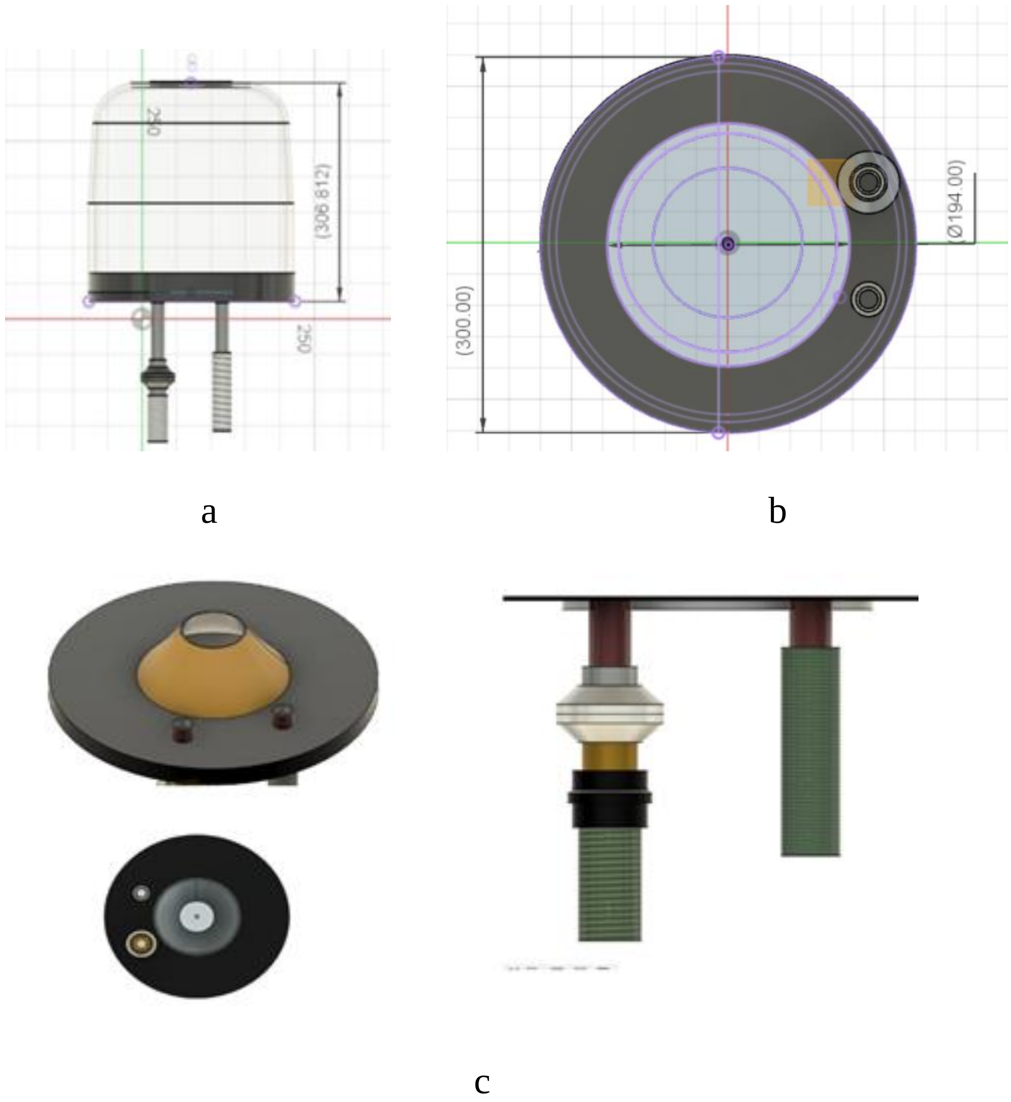


Fig. 2.2 Views of the CAD model of the prototype. a. Front view b. Bottom view with dimensions c. View of the inlet/outlet pipes, neck ring and seal.

2.2 ANSYS MODEL

Ansys is a software package and is used to simulate computer models of structures, electronics, or machine components for analysing strength, toughness, elasticity, temperature distribution, electromagnetism, fluid flow, and other attributes.

Ansys works on the principle of finite element analysis. The material of PVC foam was selected for the model and a mesh was then generated. Meshing is an integral part of the simulation process. The mesh influences the convergence, accuracy and speed of the solution. Intersection points in the mesh are known as nodes. Taking the atmospheric pressure as 101325 Pa, pressure in the inlet pipe as 9900 Pa and pressure within the helmet as 9900 Pa while keeping the base of the helmet as the fixed support, the total deformation and equivalent stress was observed.

Deformation is any alteration in the form of shape that the model exists in. Total deformation is taken as the vector sum of all the directional displacements in the system. The minimum, maximum and average deformation was observed and is as shown in Table 2.1.

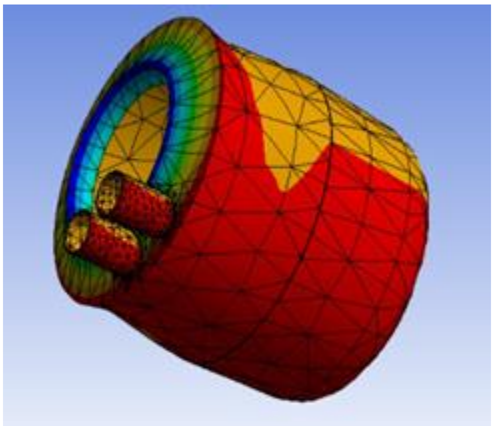


Fig. 2.3 Ansys model depicting deformation

Fig. 2.3 is the Ansys model that depicts the total deformation within the helmet prototype. The red colour in the model shows us the maximum deformation and the blue colour shows minimum deformation.

Table 2.1 Deformation as expressed through the Ansys model

MINIMUM(m)	MAXIMUM(m)	AVERAGE(m)
0	3.1115e-003	2.4851e-003

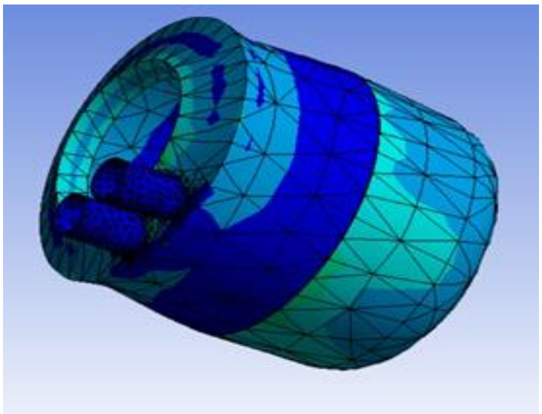


Fig. 2.4 Ansys model depicting equivalent stress

Table 2.2 Equivalent stress as expressed through the Ansys model

MINIMUM(Pa)	MAXIMUM(Pa)	AVERAGE(Pa)
498.02	3.1695e+006	2.8423e+005

Equivalent stress was calculated at each of the node points and the remaining part of it was obtained through interpolation. The stress values converge after multiple iterations. Fig. 2.4 is the Ansys model that depicts the equivalent stress within the helmet prototype. The darker shade of blue indicates where the stress is minimum and as it gets lighter, the stress increases at those points.

CHAPTER 3

METHODOLOGY OF IMPLEMENTATION

3.1 MATLAB SIMULINK MODEL

MATLAB combines a desktop environment tuned for iterative analysis and design processes with a programming language that expresses matrix and array mathematics directly. It includes the Live Editor for creating scripts that combine code, output, and formatted text in an executable notebook [6].

SIMULINK is a block diagram environment for multidomain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modelling and simulating dynamic systems. It is integrated with MATLAB [7].

Elaborated in this section is the MATLAB Simulink model that is used to analyze the working of the ventilator.

It models a positive pressure ventilator system implying that the flow of air is a combination of oxygen with different gases into the lungs with this particular pressure. The Simulink model mainly comprises four subsystems, the inlet, outlet, helmet pressure and the lung subsystems.

The lungs represented in the model are dependent on a Translational Mechanical Converter (MA) organization, which changes over the process into the translational type of motion.

A predefined controlled rate is provided to the patient. To analyse the parameters of tidal volume flow rate and airway pressure, inputs of tidal volumes, respiration rate, and I/E ratio are given to the control signal. The control signal function is to control the air supply flowing in and out of the inspiratory and expiratory tubes. For observing tidal volumes each of 500cc and 700cc, two different ventilator Simulink models have been used.

There is also a sensor block, which contains three main sensors that sense the volumetric flow rate, temperature, and humidity along with traces of gases if any, that is present in the air entering from the supply.

While defining the Interface cross-sectional region, to a unit of one, the mechanical translational network parameters of displacement, force, spring constant and damping coefficient substitute for the lung parameters of volume, pressure, elastance and resistance respectively. Fig 3.1 shows the MATLAB SIMULINK model of the helmet non-invasive ventilator with the four subsystems.

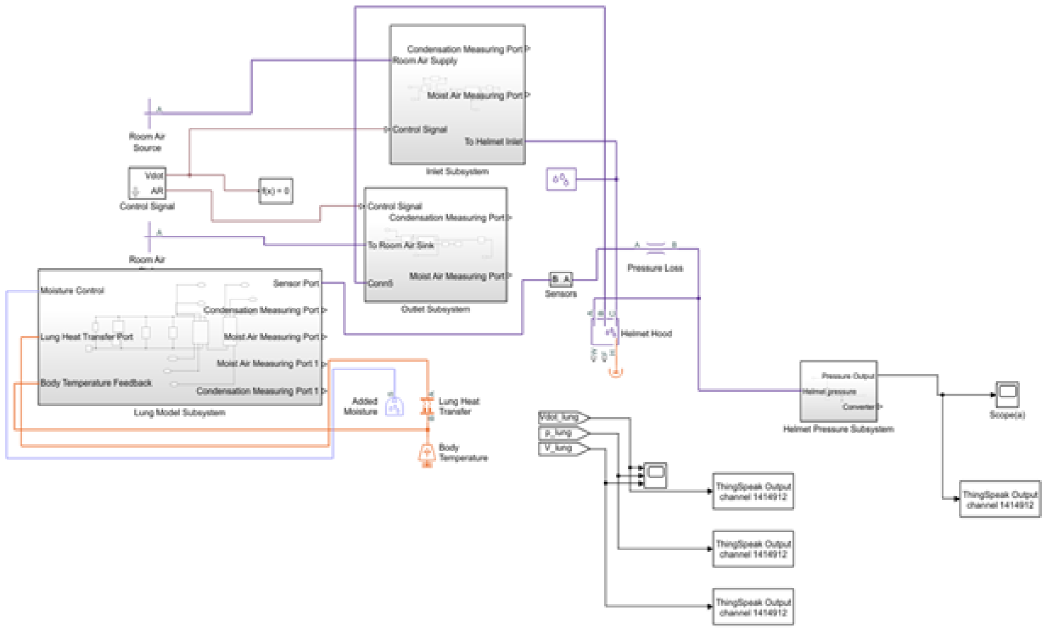


Fig. 3.1 Non-Invasive Ventilator model

3.1.1 Inlet Subsystem

The inlet subsystem controls the flow of air into the helmet ventilator. The air supply from the room enters this subsystem and goes to the volumetric supply block.

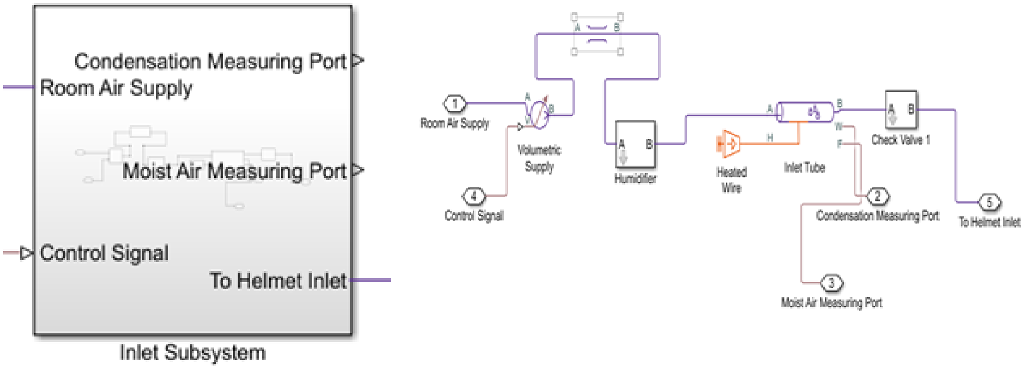


Fig. 3.2 Inlet Subsystem

The Inlet subsystem is designed as shown in Fig. 3.2. The volumetric supply block is a mechanical energy source that maintains a controlled mixture volumetric flow rate at its output regardless of the differential pressure. The flow rate is controlled but the input physical signal at port V. The positive volumetric flow rate causes the moist air to move from port A to port B. Considering there is no heat loss and the entropy remains a constant the power added parameter is set to isentropic. The standard pressure, standard temperature and standard relative humidity are set as 0.101325MPa, 20°C and 0.65 respectively. From the volumetric supply block, the air goes to the humidifier. There is a resistance offered to the flow of air and there is some generic loss of pressure. This is represented by a Flow resistance with a nominal pressure drop of 100Pa (1.0197 cm of H20).

The humidifier subsystem models an active Passover humidifier that increases the moisture in the inspiratory flow. Fig 3.3 is the Humidifier subsystem with its various parts.

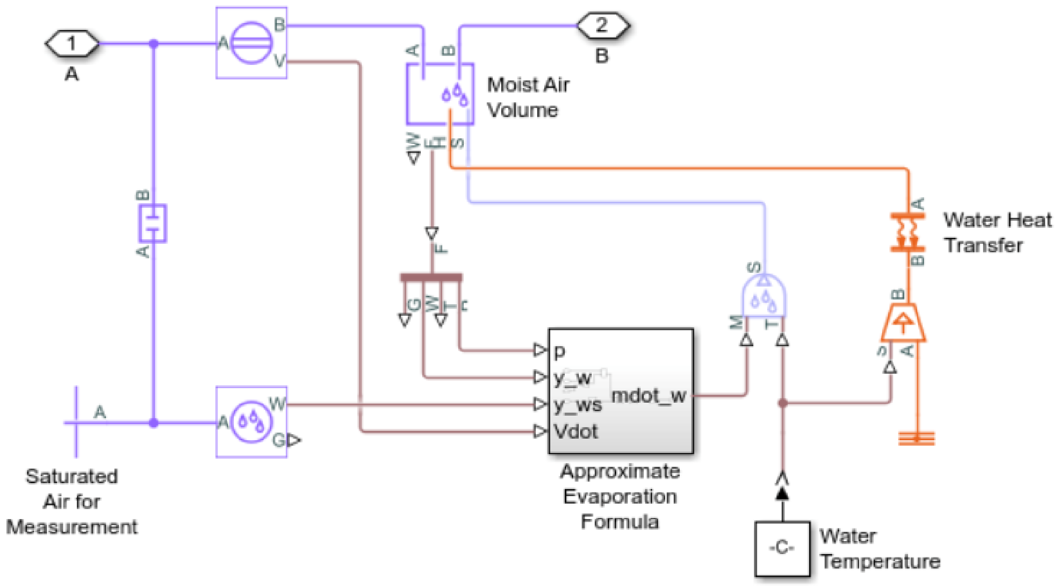


Fig. 3.3 Humidifier Subsystem

The humidifier subsystem has a volumetric flow rate sensor to measure the volumetric flow rate of the moist air and humidity & a trace gas sensor to measure the amount of moisture and air trace gas present. The Moist Air Volume block models the mass and energy storage in the air network. This chamber can exchange mass and energy with the connected air network and it has a constant volume of gas.

Port A and B are the chamber inlet ports to the moist air volume block. The number of inlet ports can vary from 1 to 4

Port W can be used to measure the rate of condensation

Port H is the thermal conserving port that is connected to the Water Heat Transfer

Port S is the moist air conserving post that can add or remove moisture and trace gas based on the requirements. This is connected to a controlled moisture source.

Port F is a physical signal port that reports the moist air volume measurements. It is connected to a Measurement selector block

The measurement selector block extracts the moist volume measurement from Port F of the Moist Air volume block. The signal ports P, T, W, and G report the value of pressure, temperature, moisture, and trace gas levels of the is air connected with it. The moisture measurement of the saturated air and the moist air, volume measurement, and pressure measurement are fed to an Approximate Evaporation Formula as shown in Fig. 3.4.

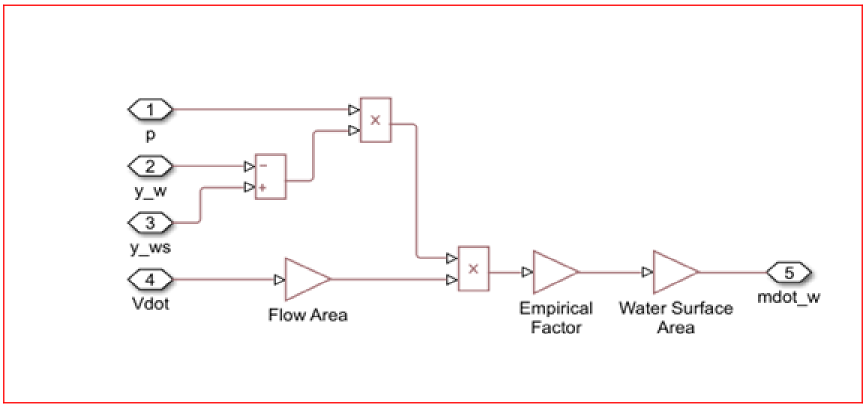


Fig. 3.4 Approximate Evaporation Formula

The Water Heat Transfer block models the heat transfer by convection. The heat transfer depends on the temperature difference, heat transfer coefficient, and the surface area in contact. From the humidifier, the moist respiratory air passes through the inlet tube which models the pipe flow dynamics due to viscous friction loss and convective heat transfer with the pipe wall. The pipe has some volume of moist air and the temperature and pressure evolve based on the compressibility and thermal capacity of the moist air. The air now passes through the check valve 1 subsystem as shown in Fig. 3.5 which allows it to flow into the helmet.

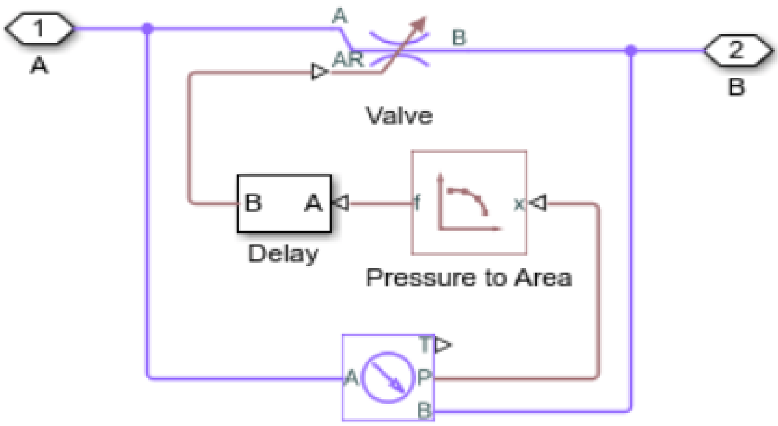


Fig. 3.5 Check Valve 1

The check valve system checks the inspiratory airflow before letting it into the helmet to ensure proper working of the helmet ventilator.

3.1.2 Outlet Subsystem

The outlet subsystem is similar in design to the inlet subsystem but distinctive in the process. In this subsystem, the expired air is again humidified and heated by the humidifier and the thermal network, as it leaves the outlet pipe

This air is pressure controlled by the expiratory peep valve and collects into the room air sink with the accumulation of condensed water in the expiratory tube, which should be drained periodically. Fig. 3.6 shows the Outlet subsystem within which there is a Check Valve 2 Subsystem as shown in Fig. 3.7.

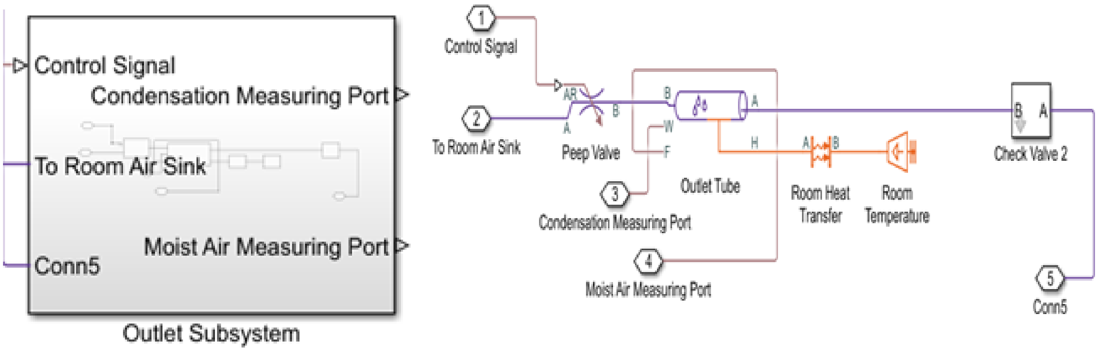


Fig. 3.6 Outlet Subsystem

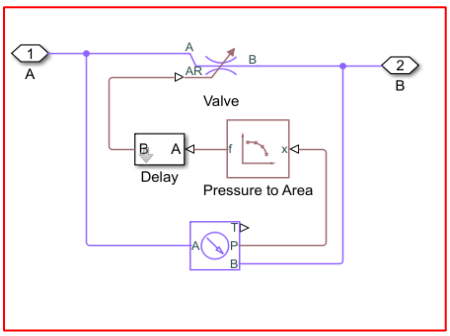


Fig. 3.7 Check valve 2

3.1.3 Helmet Pressure Subsystem

This model represents a helmet pressure subsystem shown in Fig. 3.8. This subsystem is used to simulate the different pressure losses inside the helmet. The difference in the pressure is with respect to the difference between the inhaled air and exhaled air by the patient breathing inside the hood of the helmet. The values are input in the Flow resistance block using a defined range of values for the pressure drop which acts as input values to the helmet pressure subsystem. We have used a pressure range of 31 to 293 Pa for simulation. There is a sensor that is used for measuring the temperature and pressure when the air composed of moisture is present in the helmet hood on account of breathing in of the patient. The input values to the sensor are converted from mechanical units to electrical units with the help of a converter used in the model.

In addition to this, the room pressure is taken into consideration with the standard value of 1 atmospheric pressure, which is further converted from Pa to cm H20. This is then given as output to the scope for analysis.

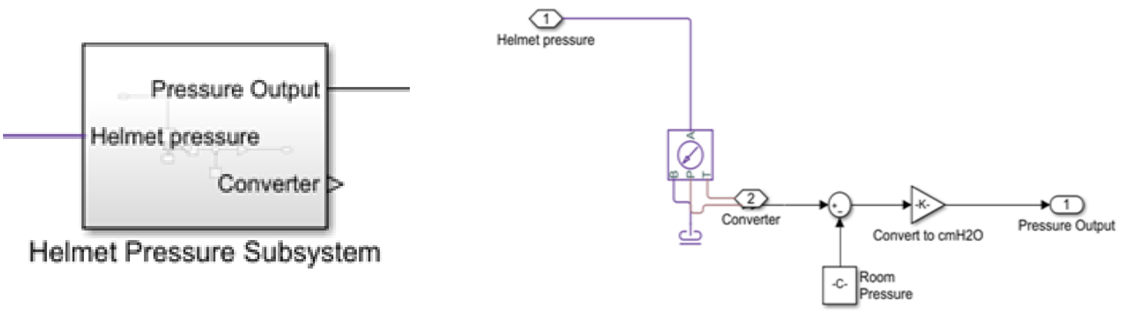


Fig. 3.8 Helmet Pressure Subsystem

3.1.4 Lung Model Subsystem

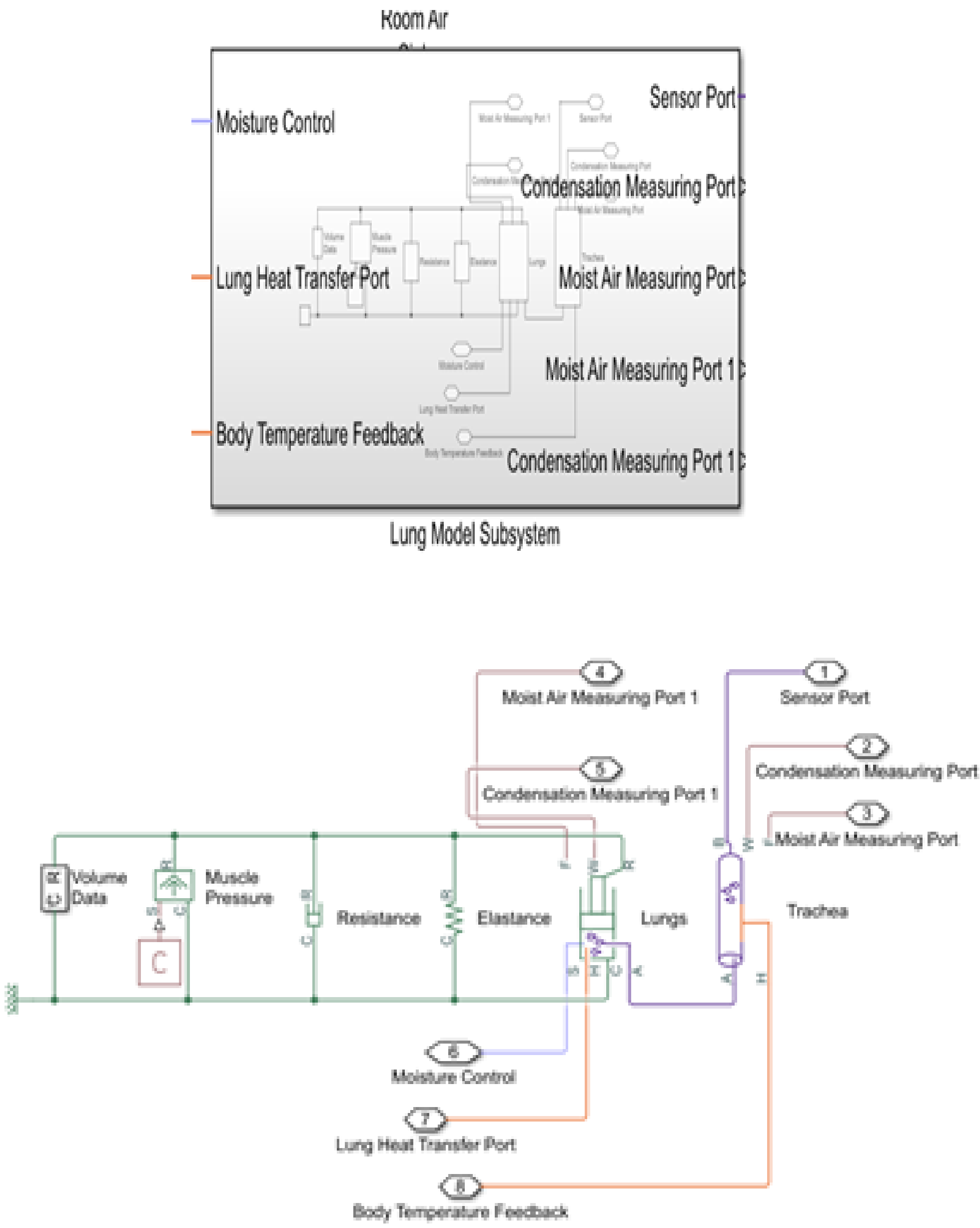


Fig. 3.9 Lung Model Subsystem

The model shown in Fig 3.9 represents a Simulink lung model subsystem. The relation Between airflow volumes and various pressure in the lung is considered in the lung mechanics model. It also, considers different parameters like airflow resistance R , pressure P ,compliances C of airflow compartments. The moist air pressure can be converted into translational motion using a translational mechanical converter in a Simulink lung mechanic model. Also, it is used to measure humidity and temperature in ventilator tubes with the help of sensor port connected to sensor block and condensation in the respiratory tube is measured with the condensation measuring port . The moisture which acts like a source of sink for the moist air volume connected is represented by the Moisture constant block. Moisture can be added or removed to maintain constant moisture. Heat transfer model is connected to lung heat transfer port in a thermal network by convection due to fluid motion. The rate of heat transfer depends on various factors like temperature difference, heat transfer coefficient, and surface area in contact with the fluid. Ideal energy source in a network can maintain a constant absolute temperature at the lung heat transfer port regardless of the heat flow rate.

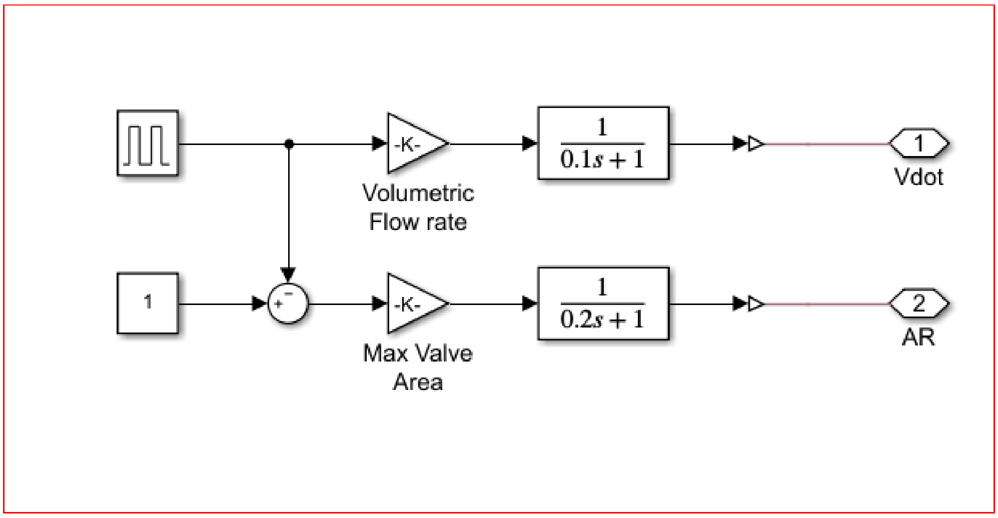


Fig. 3.10 Control signal

To analyse the parameters of tidal volume flow rate and airway pressure, the inputs of tidal volumes, respiration rate and I/E ratio are given to the control signal. Control signal function, shown in Fig 3.10 is to control the air supply flowing in and out of the inspiratory and expiratory tubes. Tidal volumes each of 500cc and 700cc is observed

CHAPTER 4

RESULTS AND ANALYSIS

4.1 SIMULINK MODEL OUTPUT

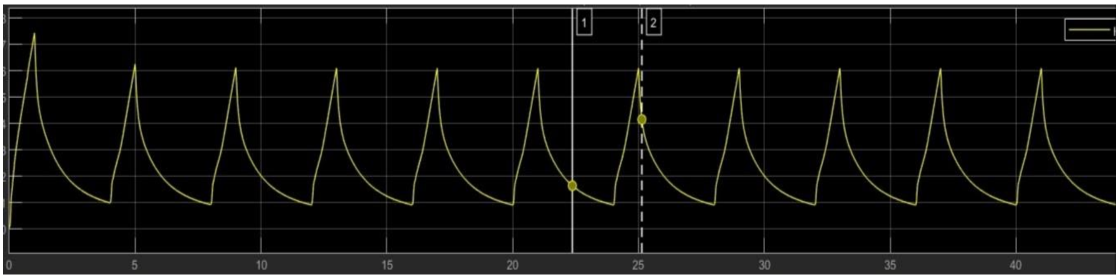


Fig. 4.1 Pressure waveform

Fig.4.1 shows the sample waveform for monitoring the pressure loss for 31pa and tidal volume of 500cc. Pressure loss is the difference in pressure inside the helmet, after one cycle of breathing. Generally, when the patient exhales, the pressure is higher.

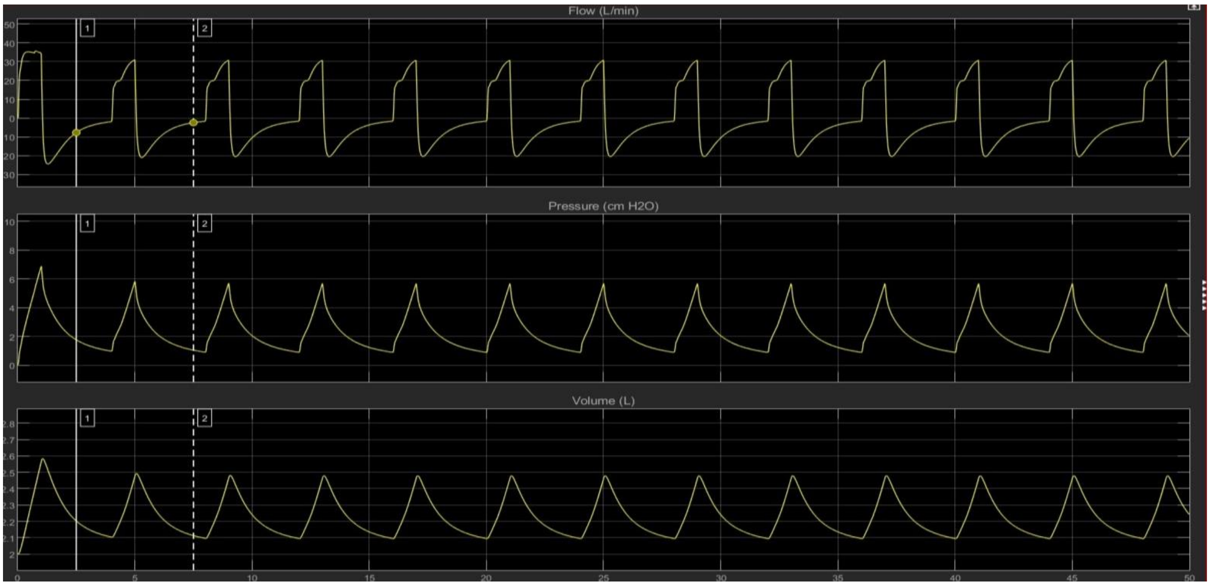


Fig. 4.2 Flow rate, Airway pressure and volume waveform

Fig.4.2 shows the sample waveform to monitor ventilator parameters like flow rate, pressure, and volume for the tidal volume of 500cc and pressure drop of 31pa. tidal volume is a certain amount of air that goes into the patient. When a patient inhales with the help of ventilator circuits it generates pressure, which is positive pressure. But in the case of actual breathing, the pressure is negative as a person inhales air from the environment without the help of a ventilator. The maximum flow at which a set tidal volume breath is delivered by the ventilator is a flow rate. All these basic ventilator parameters are represented in the waveform.

4.2 THINGSPEAK IOT OUTPUT

Fig. 4.3 ThingSpeak Output waveform

Fig 4.3 shows the sample graph for the pressure drop of 31pa and 500 tidal volume of 500 cc. When output data from the Simulink model is uploaded on ThingSpeak it provides visualizations of the data uploaded as shown in the figure.

There are four fields here; the first field is the output obtained from the vdot_lung which represents the airflow in the form of a waveform. Similarly, the other 3 waveforms obtained from p-lung, v-lung and p-loss represents pressure, volume and pressure loss respectively in the form of waveform.

Value	500cc	Pressure	31Pa	31Pa	31Pa	31Pa
created_at	entry_id		field1	field2	field3	field4
2021-06-13 13:1	1					
2021-06-13 13:1	2		0	-1.48E-13	2	-1.48E-13
2021-06-13 13:1	3		-10.5298	2.042119	2.240154	1.993842
2021-06-13 13:1	4		-1.65183	0.891909	2.094825	0.890722
2021-06-13 13:1	5		-10.5296	2.042084	2.24015	1.993808
2021-06-13 13:1	6					
2021-06-13 13:1	7		0	-1.48E-13	2	-1.48E-13
2021-06-13 13:1	8					
2021-06-13 13:1	9		0	-1.48E-13	2	-1.48E-13
2021-06-13 13:1	10					
2021-06-13 13:2	11		0	-1.48E-13	2	-1.48E-13
2021-06-13 13:2	12					
2021-06-13 13:2	13		0	-1.48E-13	2	-1.48E-13
2021-06-13 13:2	14		-10.5298	2.042119	2.240154	1.993842
2021-06-13 13:2	15		-1.65183	0.891909	2.094825	0.890722
2021-06-13 13:2	16		-10.5296	2.042084	2.24015	1.993808

100Pa	100Pa	100Pa	100Pa	100Pa	100Pa
created_at	entry_id	field1	field2	field3	field4
2021-06-13 12:5	1				
2021-06-13 12:5	2	0	-1.48E-13	2	-1.48E-13
2021-06-13 12:5	3	-10.3475	2.135703	2.248865	1.985289
2021-06-13 12:5	4	-1.7698	0.917005	2.097737	0.912615
2021-06-13 12:5	5	-10.3467	2.135605	2.248852	1.985213
2021-06-13 12:5	6	-1.77005	0.917063	2.097744	0.912671
2021-06-13 12:5	7	-10.3465	2.13558	2.248849	1.985194
2021-06-13 13:0	8	-1.76995	0.917053	2.097742	0.912661
2021-06-13 13:0	9	-10.347	2.135638	2.248857	1.985239
2021-06-13 13:0	10	-1.76991	0.917038	2.097741	0.912646
2021-06-13 13:0	11	-10.3474	2.135691	2.248863	1.98528
2021-06-13 13:0	12	-1.7705	0.917146	2.097753	0.912751
2021-06-13 13:0	13	-10.3464	2.135562	2.248847	1.985179
2021-06-13 13:0	14	-1.77027	0.917111	2.097749	0.912715
2021-06-13 13:0	15	-10.3472	2.135676	2.248861	1.985268
2021-06-13 13:0	16	-1.76989	0.917038	2.097741	0.912646
2021-06-13 13:0	17	-10.3471	2.135661	2.248859	1.985258
2021-06-13 13:0	18	-1.76982	0.91702	2.097739	0.912629
2021-06-13 13:0	19	-10.3467	2.135597	2.248851	1.985207
2021-06-13 13:1	20	-1.77058	0.917162	2.097755	0.912768

Fig. 4.4 Dataset from ThingSpeak

Sample datasets that were exported from the ThingSpeak are shown in Fig 4.4. The first figure shows the dataset for 500cc tidal volume and the pressure drop of 100pa. Similarly, another figure is for a pressure drop of 31 pa.

4.3 WEBSITE

The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Website is developed using Hypertext markup language or HTML , CSS and JavaScript. Where, HTML acts as a building block to develop a web. It defines the meaning and structure of web content . Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation of a document written in HTML or XML. Both HTML and CSS gives structure and style to web pages whereas Javascript is a text based programming language used to make a web page interactive so as to engage a user.

Index.html

```
<!DOCTYPE html>

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>Group 4-Ventilator project</title>
</head>
```

The code snippet allows the ventilator data file to be uploaded onto the website in the form of the extension for the scripts. The html head will contain the properties of the file and the unique identifier of the particular file that needs to be uploaded. The websites' attributes and extensions of what all data need to be uploaded and displayed is written into the code. This forms a part of the html head.

In addition to this, the style of the document on how it will be represented on the website is written in the html file. The category of <style> tag used in the code, is defined by the "type" attribute. This will take in as inputs, the font colour, font size, font style and any table with rows and columns that are defined. In this attribute, the contents of the style tag and its identifier are parts of it.

```
<body>

  <h1>Upload the Ventilator data</h1>
  <input type="file" id="upload-file">
  <table></table>


  <script
    src="https://unpkg.com/read-excel-file@4.1.0/bundle/read-excel-file.min.js"
  ></script>
  <script src='./main.js'></script>
```

```
</body>
</html>
```

This is the main.js script of the ventilator data readable in JavaScript format and is defined in the body of the code. It is tagged in the body as a syntax of <body>, containing the main content of the document.

The main.js file script contains the code which contains the ventilator data, is added as an extension. In this, read- excel- file library in JavaScript imported as a package from the Node Package Manager (NPM) to read the excel file in JavaScript format.

NPM is a package in JavaScript acting as a default package manager role for Node.js. All these are processed through a CDN which is short for Content Delivery Network. It is a distributive network consisting of servers and centres for data. Its main function is help deliver and distribute the web content from the respective content developers to all their end users. Their advantages are that they can provide a user-friendly experience of fast delivery of content without major glitches and from remote locations too.

Main.js

```
const input = document. querySelector('#upload-file');
```

Introduction to JavaScript querySelector() and querySelectorAll() methods. The querySelector() is a method of the Element interface. The querySelector() allows you to find the first element that matches one or more CSS selectors.[8] You can call the querySelector() method on the document or any HTML element used to upload file.[8] It takes the file that we upload by clicking on the choose file button as the input.[8]

```
input. addEventListener ('change', function (e) {
  const result = {};
  let currentVentilator;
  readXlsxFile (e. target.files[0]).then((rows) => {
    rows. forEach ((row, index) => {
      if (row [0] && row [0]. indexOf('Ventilator') > -1)
```

The data from the excel sheet is uploaded into the javascript file into the .js format by the readXlsxFile command. It is then converted into arrays, containing the ventilator data as a list.

```
  rows. forEach ((row, index) => {
    if (row [0] && row [0]. indexOf('Ventilator') > -1) {

   if (currentVentilator) {

        result[currentVentilator] =
          rows [index - 1][2] === 0? 'available': 'not available';
      }
      currentVentilator = row [0];
    }
  });
  if (currentVentilator) {
    result[currentVentilator] = rows [rows. length - 1][2]
      ? 'available'
      : 'not available';
  }
  generateTable(result);
  console.log(result);
 });
});
```

Rows and columns are formed for displaying in the website. The index number of each row, keeps increasing depending on the number of ventilators. Hence, the current ventilator's value keeps getting assigned, when a new ventilator data is added, for displaying availability. In the if loop, the objects are stored from an array to the rows of the number of ventilators. The loop runs through each row with the help of index number assigned to the ventilator and checks if the condition is 0 or a value of the ventilator. If the data is recorded as 0 in the dataset, the ventilator is displayed as “not available”. If there is a value of one of the parameters like pressure and/or volume mentioned in the dataset, then it displays “available”. If it finds an index number, containing an empty row with no data, then it goes to the immediate previous row, to check what type of data is entered. Finally, it assumes this to be the current ventilator number and the current status which is displayed on the website page.

```
function generateTable(result) {
 const table = document.querySelector('table');

 table. innerHTML = '';



 const thead = table. createTHead ();
 const row = thead.insertRow();
 let th1 = document.createElement('th');
 th1.innerText = 'Name';
 let th2 = document.createElement('th');
 th2.innerText = 'Status'; of ventilator
 row.appendChild(th1);
 row.appendChild(th2);



 Object.keys(result). forEach((item) => {
  const newRow = table.insertRow(-1);
  const td1 = document.createElement('td');
  td1.innerText = item;
  const td2 = document.createElement('td');
  td2.innerText = result[item];
  newRow.appendChild(td1);
  newRow.appendChild(td2);
```

Once the loop runs for inspecting the data of the ventilator, the table of data is generated. This can be reset according to the type and number of inputs provided. More number of patient and ventilator parameters can be added according to the end user's application. Accordingly, the table headings are generated with columns headings like "Name" and "Status" created as part of the table. Following this, the final table with body contents displaying the availability of the ventilator can be visible on website.

<u>Style.css</u>

The style.css styles the webpage by specifying various parameters like margin style, font style and size, padding etc.

A website was developed to analyse the data from ThingSpeak to check the availability of ventilators. Fig 4.5 shows the dataset that is to be uploaded on the website.

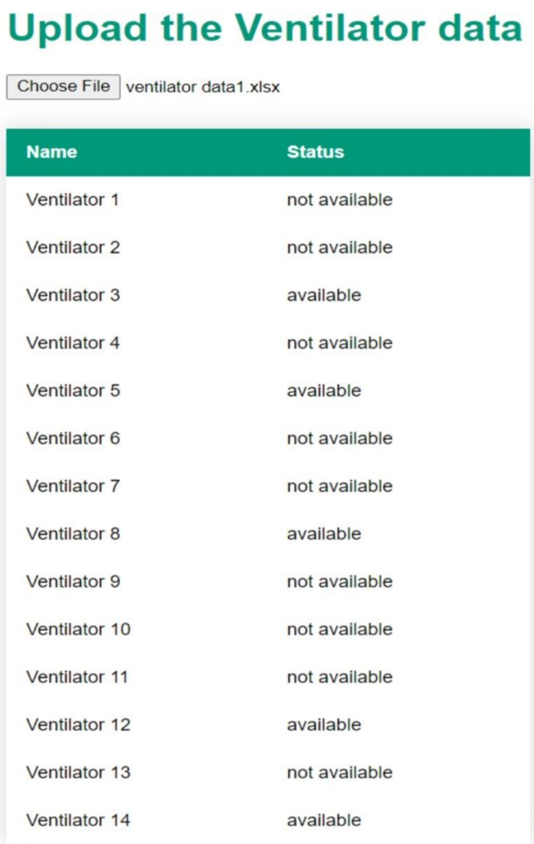
Ventilator 2						
created_at	entry_id	V_dot lung	p_lung	v_lung	p_loss	
2021-06-13 10:45:41 UTC	1	-2.21897	0.996624	2.107232	0.993102	
2021-06-13 10:46:29 UTC	2	-2.21897	0.996624	2.107232	0.993102	
2021-06-13 10:46:47 UTC	3	-4.61E-34	-1.48E-13	2	-1.48E-13	
2021-06-13 10:47:47 UTC	4	-15.655	2.68751	2.322127	2.511727	
2021-06-13 10:48:28 UTC	5	-2.21897	0.996624	2.107232	0.993102	
2021-06-13 10:49:27 UTC	6	-15.6555	2.687569	2.322135	2.511773	
2021-06-13 10:50:07 UTC	7	-2.21966	0.996747	2.107247	0.993223	
2021-06-13 10:51:07 UTC	8	-15.6549	2.687503	2.322127	2.511721	
2021-06-13 10:51:47 UTC	9	-2.21973	0.996756	2.107248	0.993232	
2021-06-13 10:52:47 UTC	10	-15.6562	2.687661	2.322146	2.511851	
2021-06-13 10:53:27 UTC	11	-2.21964	0.996726	2.107245	0.993204	
2021-06-13 10:54:46 UTC	12	-15.6562	2.687661	2.322146	2.511851	
2021-06-13 10:55:08 UTC	13	-4.61E-34	-1.48E-13	2	-1.48E-13	

Ventilator 3						
created_at	entry_id	V_dot lung	p_lung	v_lung	p_loss	
2021-06-13 11:05:33 UTC	1	0	0	0	0	
2021-06-13 11:06:33 UTC	2	0	0	0	0	
2021-06-13 11:07:22 UTC	3	0	0	0	0	
2021-06-13 11:07:45 UTC	4	0	0	0	0	
2021-06-13 11:08:45 UTC	5	0	0	0	0	
2021-06-13 11:09:25 UTC	6	0	0	0	0	
2021-06-13 11:10:25 UTC	7	0	0	0	0	
2021-06-13 11:11:05 UTC	8	0	0	0	0	
2021-06-13 11:12:05 UTC	9	0	0	0	0	
2021-06-13 11:12:45 UTC	10	0	0	0	0	

Fig. 4.5 Dataset to upload on website

Fig. 4.6 Website before uploading dataset

Fig 4.6 is an image of the website before uploading the dataset. The Choose File option is to be clicked on and the dataset is uploaded. Once the dataset is chosen the dataset is analysed and based on whether there is an air flow or not the ventilator availability is displayed on the website as shown in Fig 4.7.



Name	Status
Ventilator 1	not available
Ventilator 2	not available
Ventilator 3	available
Ventilator 4	not available
Ventilator 5	available
Ventilator 6	not available
Ventilator 7	not available
Ventilator 8	available
Ventilator 9	not available
Ventilator 10	not available
Ventilator 11	not available
Ventilator 12	available
Ventilator 13	not available
Ventilator 14	available

Fig. 4.7 Displaying availability of ventilators on the website

The data were collected on the basis of changing different ventilator parameters like pressure drop tidal volume.

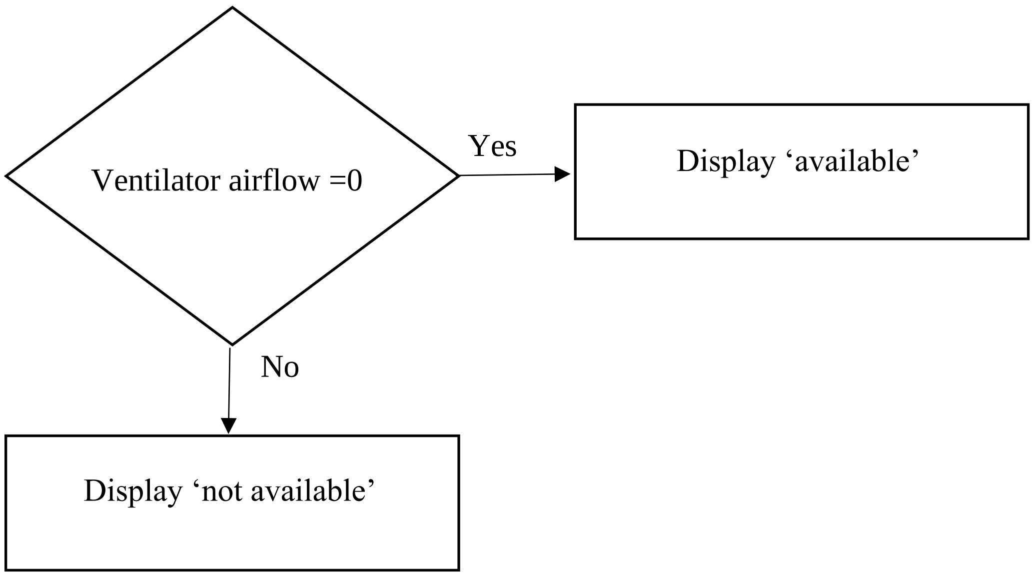


Fig. 4.8 Flowchart that determines availability of ventilator

These datasets which were exported from the ThingSpeak is used to check the availability of the ventilator on a website developed on the condition when the ventilator is in use there’s some airflow so there will be some vdot-lung value as in the case of ventilator 2 shown in the Fig 4.5 there’s some vdot-lung so ventilator 2 is in use which means it is not available whereas in ventilator 3 there’s no airflow so the value is 0 it indicates that ventilator 3 is not in use so it displays available on the website as shown in the Fig 4.6

There’s a choose file option on the website to upload the dataset once it is uploaded it checks the availability of the ventilator.

4.4 DATA FLOW

Fig. 4.8 shows the data flow diagram. The data from MATLAB SIMULINK is uploaded on ThingSpeak which is then extracted into an excel file to create a dataset to upload on the website to show availability of ventilators.

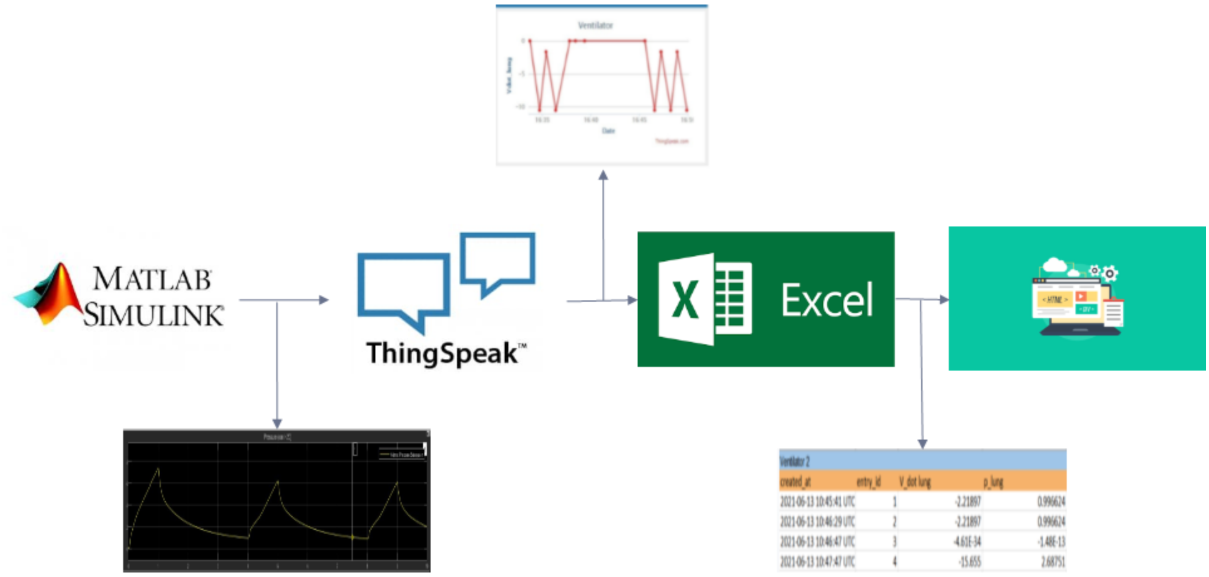


Fig. 4.9 Data flow diagram

The MATLAB SIMULINK was used in simulating the ventilator model to monitor basic ventilator parameters like pressure, airflow and lung volume. This output data of the Simulink model will be uploaded in real-time continuously after every 15 seconds in the ThingSpeak which is an IoT analytics platform. ThingSpeak stores and analyzes the data in the cloud. Data being analyzed is the plot shown in the figure. stored data from ThingSpeak is exported in CSV format and can be viewed in excel. The figure below shows the data exported. This dataset exported from the ThingSpeak is used to check the availability of the ventilator on a website developed.

CHAPTER 5

CONCLUSION

Ventilatory support was administered via a helmet eliminating the need of an endotracheal airway. A 3D model of the non-invasive helmet-based ventilator was designed with specific dimensions using Autodesk Fusion360, a CAD software. This model was exported from Fusion360 in IGES file format to ANSYS software and the minimum, maximum and average equivalent stress and total deformation was observed.

The working of the ventilator model was simulated using the MATLAB/Simulink software. For tidal volumes of 500cc and 700cc and by varying the pressure loss in the helmet from the range of 31Pa to 293Pa, the output parameters of airway pressure, respiratory flow and volume was observed.

The airway pressure, volume, respiratory flow and pressure loss in the helmet were communicated to the ThingSpeak Cloud platform in real time. This data was stored and it can be used for further analysis of the condition of the patient. The data was then extracted from ThingSpeak in a csv format. This data was then uploaded on a website, where it was analysed. The respiratory flow values were checked. Whenever there was no flow, the ventilator was considered to be available to another patient for use. The website would display this availability of ventilators.

The ventilator model was designed using CAD and few parameters were observed using ANSYS. The working of the model was analysed using MATLAB Simulink and the output parameters from Simulink were uploaded in real time on ThingSpeak. This data was then uploaded on the website and analysed to display the availability of ventilators on the website.

FUTURE SCOPE

To conclude further with the proposed methodology of the non-invasive helmet ventilator, the following ideas can be put to test for analysis:

1. Automatic, real time update of values of ventilator directly into the website
2. Displaying ventilator availability for a large number of ventilators and patients
3. Analyzing the ventilator Simulink model with other medical parameters such as ECG rate, Spo2 level, pulse rate, etc.
4. Analyzing the body temperature loss of the patient through the outlet of the ventilator pre-set with the temperature limit

REFERENCES

[1] Yasser Khan, Hossain Mohammad Fahad, Sifat Muin, Karthik Gopalan, “A low-cost, helmet-based, non-invasive ventilator for COVID-19” in Medical Physics, Citation: arXiv:2005.11008[physics.med-ph], 2020

[2] Onintza Garmendia, Miguel A. Rodríguez-Lazaro, Jorge Otero, Phuong Phan, Alexandrina Stoyanova, Anh Tuan Dinh-Xuan, David Gozal, Daniel Navajas, Joseph M. Montserrat and Ramon Farré, Eur Respir J,” Low-cost, easy-to-build non-invasive pressure support ventilator for under-resourced regions: open-source hardware description, performance and feasibility testing”; European Respiratory Journal 55,2020

[3] Mohammad Jaber, Lara Hamawy, Mohamad Hajj-Hassan, Mohamad Abou Ali and Abdallah Kassem, “MATLAB/Simulink Mathematical Model for Lung and Ventilator”, 32nd International Conference on Microelectronics (ICM), 2020

[4] Sanpanich, Y. Kajornpredanon, W. Sroykham, K. Petsarb, C. Phairoh, W. Angkhananuwat, P. Bunluechockchai Institute of Molecular Biosciences, Mahidol University, Nakhon Pathom, “A ventilation parameters waveform study by using a simple ICU ventilator circuit”, The 2018 Biomedical Engineering International Conference (BMEiCON), 201

[5] Michaela Chovancová and Jakub Elcner, “The pressure gradient in the human respiratory tract”, EPJ Web of Conferences, 2014

[6] https://www.mathworks.com/products/matlab.html

[7] https://www.mathworks.com/help/simulink/

[8] https://www.javascripttutorial.net/javascript-dom/javascript-queryselector/

APPENDIX

CODE FOR VENTILATOR AVAILABILITY WEBSITE

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>Group 4-Ventilator project</title>
</head>
<body>
  <h1>Upload the Ventilator data</h1>
  <input type="file" id="upload-file">
  <table></table>

  <script
    src="https://unpkg.com/read-excel-file@4.1.0/bundle/read-excel-file.min.js"
  ></script>
  <script src='./main.js'></script>
</body>
</html>
```

Main.js

```
const input = document.querySelector('#upload-file');


input.addEventListener('change', function (e) {
 const result = {};
let currentVentilator;
 readXlsxFile(e.target.files[0]).then((rows) => {
 rows.forEach((row, index) => {
 if (row[0] && row[0].indexOf('Ventilator') > -1) {
       // if currentVentilator's value is unassigned, then set the status
   if (currentVentilator) {
   result[currentVentilator] =
   rows[index - 1][2] === 0 ? 'available' : 'not available';
       }
       currentVentilator = row[0];
     }
    });
    if (currentVentilator) {
      result[currentVentilator] = rows[rows.length - 1][2]
       ? 'available'
       : 'not available';
    }
    generateTable(result);
    console.log(result);
  });
```

```
});


function generateTable(result) {
  const table = document.querySelector('table');
  // Reset table before each upload
  table.innerHTML = '';


  // Generate Table Headings
  const thead = table.createTHead();
  const row = thead.insertRow();
  let th1 = document.createElement('th');
  th1.innerText = 'Name';
  let th2 = document.createElement('th');
  th2.innerText = 'Status';
  row.appendChild(th1);
  row.appendChild(th2);


  // Generate Table Body
  Object.keys(result).forEach((item) => {
  const newRow = table.insertRow(-1);
  const td1 = document.createElement('td');
  td1.innerText = item;
  const td2 = document.createElement('td');
  td2.innerText = result[item];
    newRow.appendChild(td1);
```

```
    newRow.appendChild(td2);

    table.appendChild(newRow);

  });

}
```

Style.css

```
/* Style the table */

table {

  border-collapse: collapse;

  margin: 25px 0;

  font-size: 0.9em;

  font-family: sans-serif;

  min-width: 400px;

  box-shadow: 0 0 20px rgba(0, 0, 0, 0.15);

}


thead tr {

background-color: #009879;

color: #ffffff;

text-align: left;

}


th,

td {

 padding: 12px 15px;
```

```
}

tbody tr {
border-bottom: 5px solid #dddddd;
}

tbody tr:nth-of-type(even) {
background-color: #f3f3f3;
}
 tbody tr.active-row {
 font-weight: bold;
 color: #009879;
}

/* Style heading */

h1 {
color: #009879;
font-family: sans-serif;
            }
```