

AUGMENTED REALITY AND VIRTUAL REALITY BASED SIMULATION SETUPS FOR ACCELERATED LIFE TESTING

M.Tech. Thesis

By

UMESH VASANT MALI



**DISCIPLINE OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE**

JUNE 2021

AUGMENTED REALITY AND VIRTUAL REALITY BASED SIMULATION SETUPS FOR ACCELERATED LIFE TESTING

A THESIS

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

of

Master of Technology

by

UMESH VASANT MALI



**DISCIPLINE OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE**

JUNE 2021



INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **AUGMENTED REALITY AND VIRTUAL REALITY BASED SIMULATION SETUPS FOR ACCELERATED LIFE TESTING** in the partial fulfillment of the requirements for the award of the degree of **MASTER OF TECHNOLOGY** and submitted in the **DISCIPLINE OF MECHANICAL ENGINEERING, Indian Institute of Technology Indore**, is an authentic record of my own work carried out during the time period from July 2019 to June 2021 under the supervision of Dr. Bhupesh Kumar Lad, Associate Professor, (PhD).

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

Umesh
3-6-21

Signature of the student with date
(Umesh Vasant Mali)

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

Bh
4/6/21

Signature of the Supervisor of
M.Tech. thesis
(Dr. B. K. Lad)

Umesh Vasant Mali has successfully given his/her M.Tech. Oral Examination held on **4th June 2021**.

Bh
4/6/21

Signature of Supervisor of M.Tech. thesis

Date:

Dr. B. K. Lad

Signature of PSPC Member

Date: 4/6/2021

Satyanarayan

Convener, DPGC

Date: 06/06/2021

anil

Signature of PSPC Member

Date: 06 June 2021

ACKNOWLEDGEMENTS

I take this opportunity to express my profound sense of gratitude and respect to all those who helped me through the completion of this work. I would first like to thank my thesis supervisor Dr. Bhupesh Kumar Lad, for his guidance and support. Professor helped me a lot whenever I ran into any tough situation or had a question about my research or writing. He provided me with the freedom to work on this project but steered me in the right direction whenever he thought I needed it. His trust in me pushed and inspired me during the completion of my work. A special thanks to Mr. Romit Patil for helping me with the AR and VR-related work. His consistent guidance has helped a lot during the completion of my work. Also, I would like to thank Mr. Ram for his consistent guidance and support. Finally, I am thankful to my friends and colleagues here at IIT Indore for their help and encouragement.

Umesh Vasant Mali

M. Tech. (Production & Industrial Engineering)

The discipline of Mechanical Engineering

IIT Indore.

DEDICATION

I dedicate this thesis

to

My beloved family

My friends

and

My guide.

Abstract

Life testing under nominal operating conditions of mechanical parts with a high mean lifetime between failure often consumes a significant amount of time and resources, therefore rendering such procedures is expensive and impractical. As a result, the technology of accelerated life testing (ALT) has been developed to obtain life characteristics of components within a reasonable amount of time.

Accelerated life testing involves testing the components at two or more accelerated levels of stress to induce failures quickly. However, the testing involved in the accelerated life testing requires the building of costly setups to perform run to failure tests on components. As these setups are costly, the setups may not be available in most of the institution. Due to the unavailability of these setups, students and researchers may not get hands-on training with the ALT setups. Another challenge associated with these setups is that ALT setups are not remotely accessible, as it requires a physical presence of an operator to perform tests.

The objective of this thesis is to address these challenges associated with the building of costly setups for accelerated life testing. For this purpose, a virtual setup can be developed for accelerated life testing instead of costly setups. A simulation algorithm can be incorporated with the virtual setup to simulate the failure behavior of components under the test. Finally, these virtual setups can also be used for the remote control and monitoring of the actual setups in real-time.

The virtual setup is developed in such a way that, it provides step-by-step instructions regarding the experimental procedure which is to be followed for testing of components. In this way, hands-on training is provided with the virtual setup. Along with virtual

setup, a generic simulation algorithm is used for the simulation of failure behavior of the component under test. The algorithm learns from actual failure data and identifies the features like failure trend, inherent noise and, abrupt jumps to generate simulated failure data. The primary purpose of the algorithm is to simulate the stochastic nature of failure involved in the failure behavior of components under test. The data generated by the simulation algorithm is then visualized in the virtual setup for the simulation of the failure behavior of a component in the VR-based setup.

For the validation of the concept proposed in this thesis, a virtual setup for an accelerated life testing of shape memory alloy springs is developed. The data generation process of generic simulation is validated by generating simulated failure data for the SMA springs undergoing thermo-mechanical fatigue. Results obtained from the simulation algorithm are compared with actual failure datasets and results obtained are discussed in chapter 5. Results show the similarity between actual datasets and simulated datasets which is illustrated using graphical representations of simulated data with the original datasets along with calculation of similarity indices and performing Anova tests on actual and simulated failure data.

For the remote control of the actual setup, an API (Application Programming Interface) is developed. The working procedure of this API is discussed in detail in chapter 4. The procedure for remote control of PPS is discussed for remote control of the setup. For the remote monitoring of the actual setup in real-time, the failure data is needed to be monitored continuously. For this purpose, the web publishing tool of LabView is used along with the WebView plugin of Unity. The detailed procedure for remote monitoring is discussed in chapter 4. The developed LabView application is controlled remotely using the VR-based setup for remote monitoring of the actual setup.

Table of Contents

Abstract	x
LIST OF FIGURES	xv
LIST OF TABLES	xviii
ACRONYMS	xix
Chapter 1. Introduction	1
1.1 Accelerated Life Testing (ALT)	1
1.2 Accelerated Life Testing Strategy	3
1.3 Importance of Accelerated Life Testing	5
1.4 Challenges in Accelerated Life Testing.....	6
1.4.1. Accelerated life testing involves building costly test setups.	6
1.4.2. Operators are not trained to perform tests on the ALT setups.....	7
1.4.3. The accessibility of ALT setups is limited.	7
1.5 Problem Statement	8
1.6 Organization of the Thesis	8
Chapter 2. Literature Review	10
2.1 Literature Survey	10
2.2 Summary of Literature Survey	14
2.3 Research Objectives.....	14
Chapter 3. Introduction to AR and VR	17

3.1 Virtual Reality (VR)	17
3.2 Augmented Reality (AR)	23
Chapter 4. Methodology	28
4.1 Selection of ALT Setup	28
4.2 Actual ALT Setup for SMA Springs.	31
4.3 Development of the VR Based Setup	34
4.3.1. Modelling the components of virtual setup using Blender 3D.....	34
4.3.2. Working on the visuals of virtual setup in Unity Game Engine.	37
4.3.3. Implementing functionality to the virtual setup. .	39
4.3 Developing Generic Simulation Algorithm.....	43
4.3.1. Actual failure data of the SMA springs.....	44
4.3.2. Features of actual failure data.....	47
4.3.3. Time to failure (TTF) of simulated component...	48
4.3.4. Estimation of random start mean.....	49
4.3.5. Estimation of threshold.....	51
4.3.6. Trend simulation.....	52
4.3.7. Incorporation of noise in simulated data.	54
4.3.8. Incorporation of abrupt jumps in the data.	56
4.4 Interfacing Simulation Algorithm with Virtual Setup.	58

4.5 Visualization of Simulated Failure Data in Virtual Setup.	60
4.6 Real-time Control and Monitoring of the Actual Setup.	63
4.6.1. Remote control of the programmable power supply.	63
4.6.2. Real-time data acquisition using a VR-based setup from a remote location.	66
Chapter 5. Results and Discussions	70
5.1 Graphical Comparison of Actual and Simulated Failure Data.	70
5.2 Calculation of Jaccard and Cosine Similarity Indices.	73
5.2.1 Calculation of cosine similarity index.	74
5.2.2. Jaccard Similarity Index	76
5.3 Analysis of Variance (ANOVA) Test.....	79
5.3.1. Anova test on TTF data from two populations....	80
5.3.2. Anova test on failure data obtained at TS 2.	82
Chapter 6. Conclusion	85
Chapter 7. Future Scope.....	88
Appendix A	90
REFERENCES.....	107

LIST OF FIGURES

Figure 1: ALT Strategy	3
Figure 2: Accelerated life testing setup	6
Figure 3: Overview of the working of virtual setup.	15
Figure 4: VR gear and VR controller.....	18
Figure 5: CAVE system	20
Figure 6: Fully immersive VR environment.....	21
Figure 7: Virtual factory layout	22
Figure 8: Virtual machine tool.....	22
Figure 9: (a) Designer checking car in an immersive environment (b) and evaluating car assembly	23
Figure 10: Tablet scenario	24
Figure 11: HoloLens glasses scenario.....	24
Figure 12: Procedure for overlaying virtual object on an AR marker .	25
Figure 13: AR instructional system for assembly operations. Multiple types of instructions are rendered through display including texts, graphics, and animations.....	26
Figure 14: User interface developed with AR for production monitoring and scheduling	26
Figure 15: Schematic overview of ALT setup for SMA springs.....	29
Figure 16: Thermo-elastic martensitic transformation	30
Figure 17: ALT setup for the SMA springs.....	32
Figure 18: Modelled Data acquisition system	35
Figure 19: Modelled programmable power supply.....	35
Figure 20: Modelled SMA spring.....	35
Figure 21: Modelled laser displacement sensor.....	36
Figure 22: Modeled desktop	36
Figure 23: Modeled pulley.....	36
Figure 24: Modeled test fixture.....	37
Figure 25: Modeled weights	37

Figure 26: Connections of SMA spring	38
Figure 27: Overall experimental setup in the Unity Game Engine.....	39
Figure 28: VR hands	40
Figure 29: Fixing spring in the VR setup and text-based information	40
Figure 30: Connecting wires in VR setup.....	41
Figure 31: Tutorial videos in VR setup.....	41
Figure 32: LabView software introduction in VR setup.....	42
Figure 33: Operating electronic equipment along with detailed specifications.....	42
Figure 34: Operation guide for electronic equipment.....	43
Figure 35: Actual failure data for SMA springs at (a) 3.5V, (b) 3.75V and (c) 4V	46
Figure 36: Scatter plot of initial values of elongation.....	50
Figure 37: Variation in the initial values of elongation of spring.....	51
Figure 38: Trend line for $n < 1$ & $k < 0$	53
Figure 39: Trend line for $n = 1$ & $k < 0$	53
Figure 40: Trend line for $n > 1$ & $k < 0$	54
Figure 41: Noise in the actual data	55
Figure 42: Random nature of abrupt jumps in the data	57
Figure 43: Working of the web	58
Figure 44: Data transfer process between two APIs.....	59
Figure 45: Simulated elongation value for 1 st cycle at 3.5V.....	60
Figure 46: Simulated elongation value for 2 nd cycle at 3.5V.....	61
Figure 47: Simulated elongation value for 3 rd & 4 th cycle at 3.5V	61
Figure 48: Simulated elongation value for 5 th cycle and life of SMA spring at 3.5V.....	61
Figure 49: Simulated elongation values and life of virtual spring at 3.75V.....	62
Figure 50: Simulated elongation values and life of virtual spring at 4V	62
Figure 51: Remote control of PPS	65
Figure 52: Block diagram for data acquisition in LabView	67

Figure 53: Front panel for data acquisition in LabView.....	67
Figure 54: Front panel for remote data acquisition opened in a web browser.....	68
Figure 55: Elongation VS the number of cycles plot for simulated data at 3.5V for simulation no. 1	70
Figure 56: Elongation VS number of cycles plot for simulated data at 3.5V for simulation no. 2	71
Figure 57: Elongation VS number of cycles plot for simulated data at 3.75V for simulation no. 1	71
Figure 58: Elongation VS number of cycles plot for simulated data at 3.75V for simulation no. 2	72
Figure 59: Elongation VS number of cycles plot for simulated data at 4V for simulation no. 1	72
Figure 60: Elongation VS number of cycles plot for simulated data at 4V for simulation no. 3	73
Figure 61: Cosine similarity indices for simulation results at 3.5V.	75
Figure 62: Cosine similarity indices for simulation results at 3.75V ..	76
Figure 63: Cosine similarity indices for simulation results at 4V	76
Figure 64: Jaccard similarity indices for simulation results at 3.5V....	78
Figure 65: Jaccard similarity indices for simulation results at 3.75V..	78
Figure 66: Jaccard similarity indices for simulation results at 4V.....	79
Figure 67: Remote control of actual ALT setup using a robotic arm ..	88

LIST OF TABLES

Table 1: Sample elongation data for SMA springs at 3.5V	46
Table 2: Sample elongation data for SMA springs at 3.75V	46
Table 3: Sample elongation data for SMA springs at 4V	47
Table 4: Time to failure data of SMA springs	48
Table 5: Failure rate for different values of β	49
Table 6: Cosine similarity index	74
Table 7: Jaccard similarity index	77
Table 8: Anova table for TTF data at 3.5V	80
Table 9: Anova table for TTF data at 3.75V	81
Table 10: Anova table for TTF data at 4V	81
Table 11: Anova table for simulated TTF data at 3.5V & 4V	82
Table 12: Anova table for failure data at TS 2 for 3.5V	83
Table 13: Anova table for failure data at TS 2 for 3.75V	83
Table 14: Anova table for failure data at TS 2 for 4V	83
Table 15: Anova table for failure data at TS 2 for 3.5V& 4V	84

ACRONYMS

ALT	Accelerated Life Testing
AR	Augmented Reality
VR	Virtual Reality
SMA	Shape Memory Alloys
HMI	Human Machine Interface
xR	Extended Reality
DAQ	Data Acquisition System
PPS	Programmable Power Supply
LDS	Laser Displacement Sensor
RTF	Run to Failure
TS	Time Stamp
TTF	Time to Failure
SD	Sample Data
CIM	Change in Mean
ANOVA	Analysis of Variance
API	Application Programming Interface

Chapter 1

Introduction

Reliability testing is done to ensure that product is reliable, it performs its specified function for the specified period, under specified operating conditions and, is capable of rendering fault-free operations. All the tests involved in reliability testing are run to failure tests i.e. components are tested till failure occurs. Often the amount of time available for testing is considerably less than the expected lifetime of the components. Also, for highly reliable components testing under normal operating conditions would generate very few failures in a reasonable time, which will not be sufficient for reliability analysis. Hence, accelerated life testing is done to obtain the failure data within a reasonable period of testing time.

1.1 Accelerated Life Testing (ALT)

Accelerated life testing of products, components, and materials is used to get information quickly on specific lives, life distributions, failure rates, mean lives, and reliabilities. Accelerated testing is achieved by subjecting the test units to conditions, or application and operation stress levels, that are more severe than normal or use conditions, to shorten lives or their times-to-failure. If the results can be extrapolated to the use conditions, they yield estimates of the lives and reliabilities under use conditions. Such testing provides savings in time and expense since, for many products, components, and materials, life under use conditions is so great that testing under those conditions is not timewise and economically feasible. In the accelerated life testing, accelerated test conditions are typically produced by testing units at higher levels of temperature, voltage, pressure, vibration amplitude, frequency, cycling rate, loads,

humidity, etc., or some combination thereof, than are encountered under use conditions. For most of the applications, testing time is considerably less than that of operational time hence, accelerated life testing is of great importance in product development.

Accelerated life testing can be defined as the process of estimating the reliability of highly reliable components by using time compression techniques and extrapolation. Time compression techniques used in accelerated life testing are as follows:

1. Increasing number of units on the test:

Additional units can be placed under test so that a greater number of failures can be obtained within the given time. But, this method of accelerated life testing is not suitable for expensive components.

2. Usage rate acceleration:

This method of accelerated life testing is most suitable for components that do not operate continuously. In this method, the number of cycles per unit time is increased for components to make failures happen quicker than if it were used normally.

3. Increasing stress that generates failure:

This method of accelerated life testing is used as the backup for acceleration for usage rate and used if usage rate acceleration cannot be used for the testing of components. This method is most suitable for components that operate continuously. In this method, stress that is responsible for failure is increased more than normal operating stress conditions to obtain failures quickly. While accelerating failure stress of the components certain points are needed to be considered these are known as assumptions in the accelerated life testing and are as follows:

- i. Same failure mode as operating conditions:

Increasing failure stress should not introduce a new failure mode in the component and the failure mode should be the same as that of the failure mode encountered during the normal working conditions of components.

- ii. Accelerating failure mode dominant at normal operating conditions:

Increasing failure mode should accelerate the failure mode which is dominant at normal working conditions without accelerating any other failure mode.

- iii. Same statistical model at accelerated conditions:

The statistical model which is applicable at normal operating conditions should also be applicable at accelerated conditions.

After using any one of the above time compression techniques, life data is obtained at accelerated test conditions. To obtain life data at normal operating conditions extrapolation is done.

1.2 Accelerated Life Testing Strategy

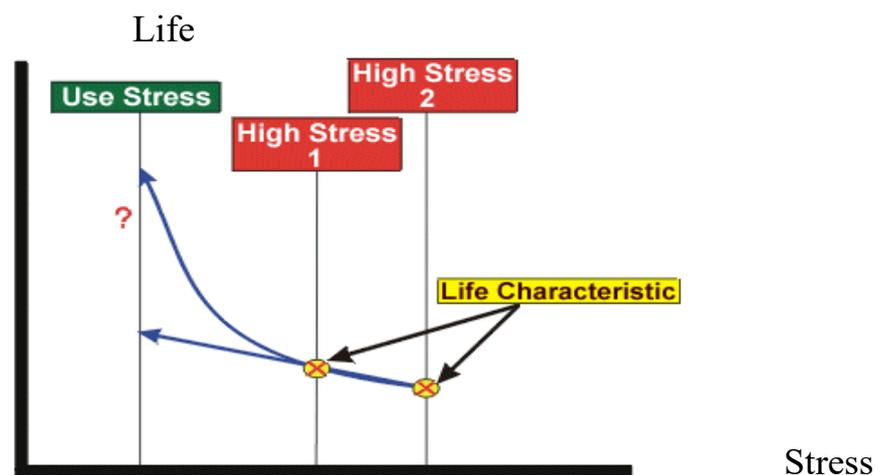


Figure 1: ALT Strategy [1]

Figure 1 shows the strategy adopted while performing accelerated life testing when using accelerating failure stress as a time compression technique. The first step involves testing a certain number of components at two or more accelerated levels of stress. The next step is to determine life characteristics at these accelerated stress conditions. Life characteristics are obtained by fitting suitable reliability distribution to the time to failure data obtained at each accelerated level of stress. The number of components to be tested at each accelerated level of stress is determined based on the accuracy of fit desired for particular reliability distribution. The higher the number of components tested more is failure data available for fitting the distribution and the greater will be the accuracy of the distribution. However, the higher the number of components tested at each stress level more will be the cost associated with testing. Hence, the optimum number of components to be tested is selected by taking into account both accuracies desired and the cost for testing. After selecting appropriate life distribution for estimation of life characteristics next step is parameter estimation. Parameter estimation involves fitting the model to time to failure data and solving parameters that describe the model. Various techniques available for the estimation of parameters of the model include the graphical method, least square method, and maximum likelihood estimation method. After parameter estimation, the next step is to define the life-stress relationship to obtain life characteristics from accelerated stress levels to normal operating stress levels. After estimation of life characteristics like mean life, median life, reliability function and failure function at normal operating conditions final step is to determine reliability information of the tested component. Reliability information that is to be derived includes warranty time, instantaneous failure rate and failure function of component tested.

From the strategy shown in figure1, it can be comprehended that testing of the components till the failure is the first and important step

in the accelerated life testing. For the testing of the component a physical setup is required. Hence, the building test setup is an essential and inseparable part of the accelerated life testing.

1.3 Importance of Accelerated Life Testing

Accelerated life testing is the process by which a product is forced to fail more quickly than it would under normal operating conditions. Forcing a product to fail more quickly reduces test time and still allows us to understand the life characteristics of the product. In today's competitive market if a company can release a reliable product before its competitors can release theirs, then the company gains a greater advantage over its competitors. Accelerated life testing can play a very important role for a company to gain this advantage. Hence, the importance of accelerated life testing can be understood from the following advantages:

- The most obvious advantage is time savings, due to a decrease in test duration because of an increase in failure stress.
- Reduction in testing times results in reduced time to market, lower product development costs, and lower warranty costs.
- ALT enables the user to predict product performance and to find the life of a component or product.
- Life prediction helps in the marketing of products and to predict demand for the product in the future.
- Constant monitoring of the Accelerated Life Testing results helps to improve the design quality of the new product development cycle.
- Knowing reliability and MTBF (mean time between failures) of product, helps in deciding warranty plan and costs for the product.

1.4 Challenges in Accelerated Life Testing

The first step in the accelerated life testing is the building of the setup for testing. Even though accelerated life testing is a time-saving technique, there are certain challenges associated with the building of the test setups. These challenges are discussed in this section and are as follows:

1.4.1. Accelerated life testing involves building costly test setups.



Figure 2: Accelerated life testing setup [2]

Figure 2 shows an accelerated life testing setup for the fatigue testing of an aerospace product called supple platinum. Experimental setup for testing consists of an engine for providing mechanical energy to apply load on specimen, command installation to switch ON and OFF the engine, electronic apparatus for reading frequencies and cycle counter, and data acquisition system for collecting data. All the components involved in the testing are costly and it is not economically feasible to build setups as shown in figure 2 for training and educational purpose.

Similarly, most of the setups involved in the accelerated life testing are costly setups. Most of the institutions have limited resources,

hence these institutions are unable to make these facilities available for their students and researchers. Due to these limitations of costly setups, these setups are not available in most institutions. Also, building multiple setups of accelerated life testing for training and educational purpose is not possible. Hence, students and researchers are unable to get hands-on training with the accelerated life testing setups.

1.4.2. Operators are not trained to perform tests on the ALT setups.

Costly setups of accelerated life testing are not available in most institutions. To make these costly facilities available to students and researchers' government of India has come up with an initiative called I-STEM (Indian Science Technology and Engineering Facilities Map). With this initiative all the costly facilities are made available on one portal, researchers can log in to this portal, search for their required facilities and get the information regarding where these facilities are available. Researchers can book a slot and perform tests on these setups which are necessary for their research work. Even if with this initiative, as researchers do not have any hands-on training with these setups, they do not have any prior knowledge of the setup. Due to this, researchers are not able to make the best possible use of these facilities for their research work.

1.4.3. The accessibility of ALT setups is limited.

To perform tests on actual setups, the physical presence of the operator and trainer is necessary in the labs, as these setups are not remotely accessible. To perform the tests on the setup operator has to go to the laboratory where these setups are located which may not be possible in pandemic situations like COVID-19. Hence, the accessibility of these test setups is limited.

1.5 Problem Statement

Performing run-to-failure tests is an essential part of reliability analysis as well as accelerated life testing, which most of the time involves the building of costly setups. Most institutions can't build these costly setups. Even if with the building of these setups it is not economically feasible to use these setups to provide hands-on training to students and researchers. Another challenge associated with the building of physical setups is that these setups are not remotely accessible, and the physical presence of students and trainers is necessary for performing the tests, which may not be possible in pandemic situations like COVID-19. Because of these challenges, students and researchers have limited knowledge and training about accelerated life testing.

These critical challenges associated with accelerated life testing are addressed by this thesis. The problem statement of this thesis is to design and develop a mechanism that can help to provide hands-on training with costly setups of accelerated life testing. A mechanism is also needed to be developed, which will simulate the failure behavior of components subjected to testing using the historical failure data. The developed mechanism can then also be used for remote control and remote monitoring of the actual setups of accelerated life testing.

1.6 Organization of the Thesis

The previous chapter gives a brief introduction regarding the accelerated life testing and also discusses its significance as a time-saving technique in product development. Chapter 1 also discusses the importance of ALT in reliability analysis along with the various challenges associated with accelerated life testing.

The next chapter discusses some plausible solutions like augmented reality and virtual reality, to tackle the challenges faced in the accelerated life testing. Chapter 2 also mentions some important research papers which provide some better insights regarding the applicability of augmented reality and virtual reality concepts in the fields related to my research problem.

To apply augmented reality and virtual reality concepts to develop simulation setups for accelerated life testing, one needs to have basic knowledge about these technologies. Chapter 3 provides a brief overview of these concepts.

Chapter 4 includes the methodology of the thesis and discusses the development procedure for generic simulation algorithm for simulation of failure behavior tested components.

Chapter 5 validates methodology by the results and discussions.

Chapter 6 concludes the thesis work and discusses the scope for future work for the improvisation of the process.

Chapter 2

Literature Review

To overcome the challenges faced in the accelerated life testing a comprehensive literature survey has been done to search for plausible solutions. Among all the possible solutions, the augmented reality and virtual reality were found to be promising technologies to address challenges discussed in the previous chapter. This literature survey includes some important applications of augmented reality and virtual reality in the training, simulation, and educational fields.

2.1 Literature Survey

L Perez, et al. (2019) [3] used virtual reality interfaces for industrial robot control and operator training. The VR-based system was used to replace the human-machine interface (HMI) of the industrial robot. VR system was used to virtually visualize trajectories of the real robot for beforehand error detection and accident prediction. The authors found the VR-based approach usable and friendly, and it reduced the training time of the operator. Operators were already familiarized with the actual working environment and also the effectiveness of training was higher for their real tasks. Testing robot trajectories and programs in VR allowed them to avoid risks and improve safety in industrial facilities. The VR-based HMI is an integrated application for training, simulation, and programming. M. Giannuzzi, et al. (2020) [4] investigated the logic interface of computer-aided manufacturing in virtual reality. Authors simulated machining operation in a VR system. They used CAM software Mastercam to find G-codes from 3D CAD drawings to define machine kinematics. Instead of testing these NC part programs directly on CNC machines, they were tested for any errors using the

VR system. The authors concluded that integration of computer-aided manufacturing and virtual reality helped in the simulation of machining operation in a realistic way to improve understanding of the machining process. Z. Guo, et al. (2020) [5] investigated applications of virtual reality in maintenance during the industrial product life cycle. Systematic literature review methodology was used to review primary studies related to virtual reality. The authors concluded that VR is a very useful tool for serving the entire lifecycle of industrial products, optimizing development processes, improving maintenance efficiency and safety, and also for reducing lifecycle costs.

Osama Halabi (2019) [6] studied the use of VR to enforce teaching in engineering education. He instructed students to design a kitchen for a person in a wheelchair using a traditional approach and VR-based approach. He observed that VR based approach helped students in the achievement of project goals, increased their cumulative project grades. It also enhanced the engagement and motivation of students and improved course outcomes significantly. He concluded that VR will improve the teaching-learning environment significantly and provide students of engineering with excellent tools to examine, design, develop new concepts and help them in developing the conceptual framework for engineering design.

Z-H Lai, et al. (2020) [7] introduced a Smart augmented reality instructional system for mechanical assembly towards worker-centered intelligent manufacturing. Authors used AR technology for workforce training for mechanical assembly. AR-based instructions for assembly were provided by tracking the progress of ongoing assembly using cameras and these instructions were provided in three different modalities viz. texts, graphics, and animations. The region-based convolutional neural network was used for identifying the correct tool required for assembly. The developed system was tested

for three different types of assembly errors viz. tool/part selection error, assembly sequential order errors, and installation errors. Comparison of AR-based system was done with the conventional system for the rate of errors and completion time. It was observed that by following smart AR instructions, the completion time for the assembly was reduced by 33.2% whereas the rate of errors was decreased by 32.4%. Tool/part selection errors and assembly sequential order errors were reduced by 72.7% and 100% respectively. M. M. L. Chang, et al. (2017) [8] described the procedure to develop an AR-based instructional system for mechanical disassembly. AR guided product disassembly framework consisted of three major modules namely disassembly sequence planning, automatic content generation, and AR interface. The disassembly sequence table was first generated based on the ontology of the product. Next, an automatic content generation domain was used to link the generated sequence with appropriate virtual information based on a taxonomy of visual cues. The final virtual content was rendered on the real scene which served as a straightforward step-by-step guide to the human operator. The developed AR system was then tested to perform the disassembly operation. The authors found that the efficiency of disassembly was considerably improved with the AR system. The authors concluded that with the use of AR guidance in disassembling a product, the quality of maintenance services was greatly improved and disassembly operation was successfully completed without expert intervention.

Dimitris Mourtzis, et al (2020) [9] coupled production scheduling and monitoring along with augmented reality. An AR-based application was used to handle and display production scheduling and machining data. The application was designed and implemented, to dynamically exchange information from digital to physical production system and back. The physiology of the application was

based on AR technology had great advantages of AR devices within manufacturing, such as user-friendly, easily interactable, and more comfortable (non-static) application. Hirosuke Horii and Yohei Miyajima (2013) [10] used an AR-based system for teaching hand-drawn mechanical drawing. 3D CAD models of educational materials with realistic textures were developed which offered an easy and intuitive user experience. The AR-based system achieved a reduction of time, effort, and monetary cost for preparing various mechanical parts as educational materials by replacing actual materials with virtual ones. Boeing is using AR technology to give technicians real-time, hands-free, interactive 3D wiring diagrams right in front of their eyes. Their studies have shown a 90% improvement in first-time quality when compared to using two-dimensional information on the airplane, along with a 30% reduction in time spent doing a job [11]. Renault trucks have tested AR technology for providing step-by-step instructions for engine assembly for the reduction in duration of inspection operations, improving quality and flexibility of operations on assembly lines [12].

Asa-Fast-Berglund, et al. (2018) [13] studied different strategies regarding when and where to implement AR and VR technology in manufacturing. AR and VR technologies were implemented at each stage of manufacturing along with conventional methods for the comparative studies. Authors found that AR can give instant feedback hence it is almost impossible to do the wrong assembly which resulted in high quality. They concluded that AR is suitable for learning phases like operator training and remote guidance, operational phases like complex assemblies, and disruptive phase like maintenance and disassembly in the manufacturing system. Whereas, VR is very useful in the design phase, product development, and operator training.

2.2 Summary of Literature Survey

From the literature survey, it can be comprehended that VR technology offers a way to simulate reality. Hence, VR technology can be used to create an immersive virtual environment for an operator to get familiarized with actual working conditions. VR technology can also be used to develop simulations to improve the understanding of the various processes and working of the setups. Also, AR and VR systems can be used for giving instructions in real-time, while performing complex experimental procedures, or while doing complex assemblies. AR and VR-based smart instructional systems can be used for operator training and educational purpose.

From the literature survey, it can also be concluded that virtual setups can be developed instead of costly setups of accelerated life testing for training and educational purpose. These virtual setups can also help in understanding the working of the actual setups and getting prior knowledge of the setups before working on the actual setups. These virtual setups can be easily accessible from any location using VR gear, and a VR controller. Hence, the challenges faced in accelerated life testing can be well addressed using AR and VR technologies.

2.3 Research Objectives

1. Objective I: To develop a virtual reality-based setup for accelerated life testing of the mechanical component.

Virtual setup developed using VR technology will be similar to the actual setup of accelerated life testing and the operator will be interacting with virtual setup in a similar manner as he or she will

interact with the actual setup for testing. In this way, hands-on training will be provided with an accelerated life testing rig.

2. Objective II: To develop a generic simulation algorithm to simulate the failure behavior of the components under the test.

After completing the experimental procedure and making necessary connections in the virtual setup, the component will be subjected to testing in the virtual setup. To incorporate the failure behavior followed by the actual component in the virtual setup an algorithm will be required, which will run in the backend of the virtual setup to generate simulated failure data which will be visualized in the virtual setup. The algorithm will learn from actual failure data to identify and extract the features in actual data to generate simulated failure data.

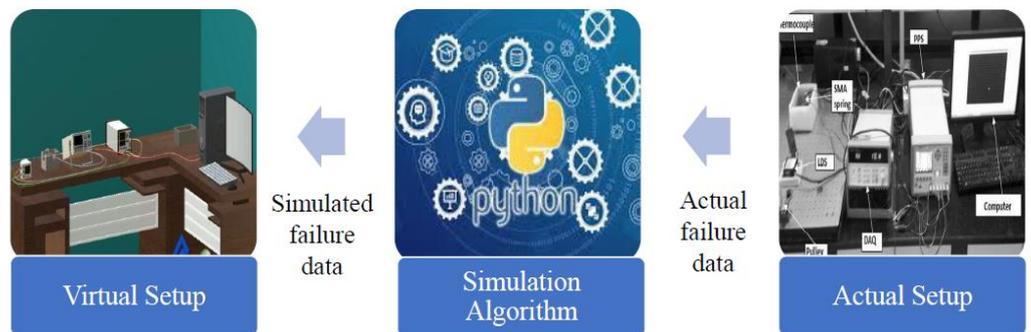


Figure 3: Overview of the working of virtual setup.

Figure 3 elaborates objective I and Objective II

3. Objective III: To monitor the working of the actual setup using the VR-based setup in real-time from a remote location.

For the third objective of the research, the operator in the lab will make all the necessary connections for the testing. A new mode

will be introduced in the virtual which will be used for the remote control of the instruments of the setup. To monitor the working of the actual setup, the data from the sensors of the actual setup will be visualized in the virtual setup in real-time from the remote location.

Chapter 3

Introduction to AR and VR

Digital manufacturing has been considered as a highly promising set of technologies for reducing costs and product development times as well as for addressing the need for customization, increased product quality, and faster response to the market. Bridging digital and physical worlds can mean a lot of time savings in almost all areas of manufacturing like design, training, prototyping, learning, marketing, logistics, maintenance, set-ups, remote guidance, and assembly, etc. Technologies like Augmented Reality (AR) and Virtual Reality (VR) could be useful to accomplish this bridge as well as to increase time-room flexibility i.e. the need not to be at the same place at the same time when working on a project. Cyber-physical systems are growing and are an important part of the industry 4.0 evolution [14].

3.1 Virtual Reality (VR)

Virtual reality (VR) is a popular information technology area that provides an indirect experience by creating a virtual space that interacts with the human sensory systems and overcomes spatial and physical constraints of the real world. VR technology can be categorized as follows: (1) expression technology for stimulating human sensory systems, (2) interaction technology for interfacing reality with VR, (3) authoring technology for developing VR content, and (4) collaboration technology that networks multiple participants within VR. VR technology has various applications, from the manufacturing industry to education, training, and entertainment [15]. In simple words, VR can be defined as “The computer-generated simulation of a three-dimensional environment that can be interacted with in a seemingly real or physical way by a person using special electronic equipment, such as a VR gear with a screen inside and VR controller” (refer Figure 4). Virtual reality

allows the user to step through the computer screen into a three-dimensional (3D) world. The user can look at, move around, and interact with these worlds as if they were real. Virtual reality can also be considered as a new media for information and knowledge acquisition, and representing concepts of ideas in ways not previously possible. VR applications can provide the users not only immersive sight beyond reality but also hearing, touch, and even the ability to interact with virtual objects.



Figure 4: VR gear and VR controller

An integral VR system usually consists of hardware that can be classified into three categories based on their effects, namely, display devices, motion capture devices, and interactive devices. A display device is the essential element of a VR system, which outputs stereoscopic images to users. The ultimate output effects will make the images users see, the same as those of reality. Display wall, PC, CAVE, the portable device are commonly used display devices. Among these display walls, PC and portable devices are most common because of their low cost and low space requirements as compared to CAVE. Despite its high cost and bulkiness CAVE has high resolution and offers a strong immersive experience.

The main function of motion capture devices also called mocap devices is to capture and process the motions of the user to provide tracking for

the user's view and positioning for interaction. the common types of mocap are optical, inertial, mechanical, and magnetic. Portable VR devices usually embed sensors and algorithms to realize motion tracking. In addition, somatosensory devices, such as Kinect and Leap Motion, can position and capture motion in specific spaces, and they have relatively low costs.

In addition to lifelike visual effects, interactive devices are important media for enhancing the reality of VR environments, through which a user can interact with objects in a virtual environment (VE) and receive sensory feedback, such as haptic and auditory feelings. The use of an interactive device usually requires cooperation with a mocap device to obtain accurate positions for command execution. Joysticks or wands are the most commonly used interaction devices. They enable the user to hover in the VE, move objects and execute many other interactions. Data gloves and haptic devices are other frequently used interaction devices, with which a user can perform more elaborate operations, such as grasping and picking. They have been widely used in industrial applications, especially in some cases that have high requirements for elaborate operations. Figure 5 shows a complete VR environment that includes a CAVE, mocap, and interactive device.

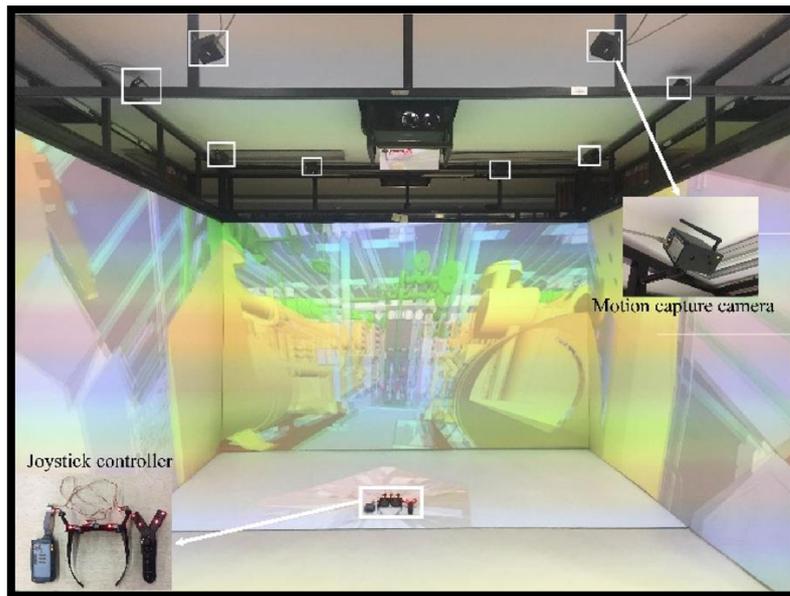


Figure 5: CAVE system [5]

Based on the sense of immersion or degree of presence provided by VR configuration VR systems can be classified into three main categories viz. Non-immersive VR, semi-immersive VR, and fully immersive VR. Non-immersive VR systems use computer systems like monitors as output devices and keyboard or mouse as input devices. Even though the extent of immersion is the least in these systems but these are the lowest cost VR systems. Semi-immersive systems use projectors or large screens as output devices whereas data gloves, joystick, etc. as input devices. Fully immersive systems use head-mounted displays (HMD) or CAVE as output devices and gloves, voice command, etc. as input devices. These systems are very expensive but provide the highest level of immersion. Figure 6 shows a fully immersive VR system.

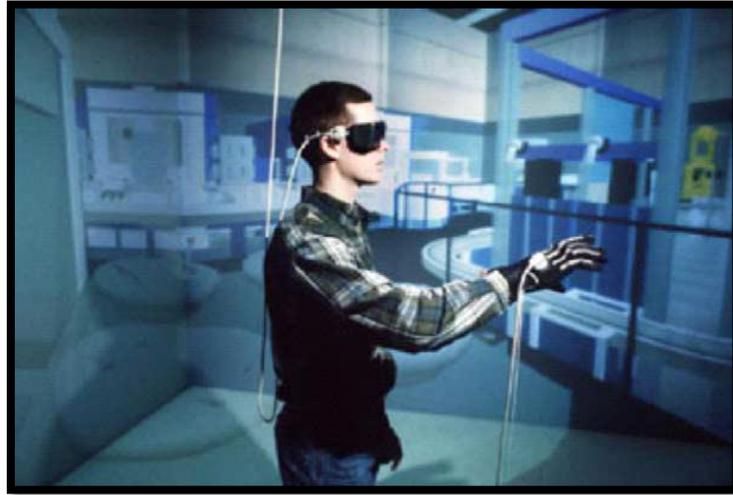


Figure 6: Fully immersive VR environment [16]

Virtual manufacturing (VM) is one of the applications of applying VR technology in manufacturing applications. Virtual manufacturing is defined as a computer system that is capable of generating information about the structure, status, and behavior of a manufacturing system as can be observed in a real manufacturing environment [17]. The vision of virtual manufacturing is to provide a capability to “manufacture in the computer”. That means VM will provide a modeling and simulation environment so powerful that the fabrication/ assembly of any product, including the associated manufacturing processes, can be simulated in the computer. The advances in virtual reality have provided the impetus for applying VR to different engineering applications such as product design, modeling, shop floor controls, process simulation, manufacturing planning, training, testing, and verification. In the design phase, VR systems can be primarily used for product design to evaluate multiple designs, to interact with the design model, and to conduct ergonomics studies and in prototyping for testing and evaluating specific characteristics of the design. In the operation management phase, VR systems can be used in factory layout planning, simulation, and operator training. Figure 7 shows the virtual factory layout created using VR. In the manufacturing phase, VR systems are useful in the machining, assembly, and inspection phase. Figure 8 shows the application of VR

in the machining process simulation. Figure 9 shows the application of VR in inspection and assembly operations.



Figure 7: Virtual factory layout [16]

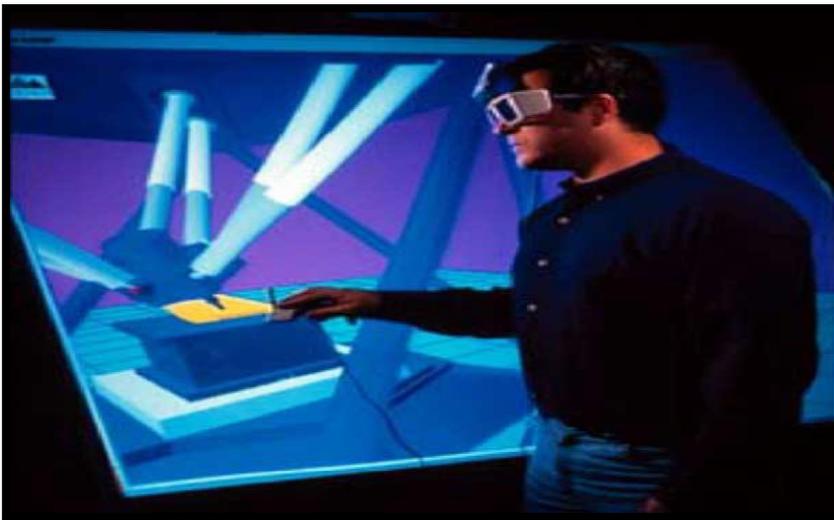


Figure 8: Virtual machine tool [16]

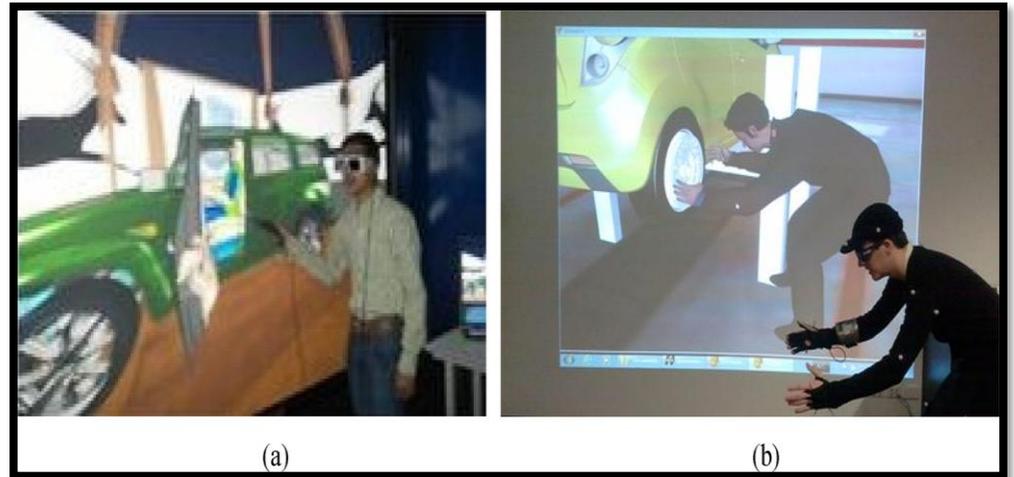


Figure 9: (a) Designer checking car in an immersive environment (b) and evaluating car assembly [5]

3.2 Augmented Reality (AR)

Augmented reality (AR) is an emerging human-computer interaction technology that renders virtual information on a real scene. AR can be defined as a technique that enriches reality by superimposing a layer of information and numerical contents. In other words, AR is used to ‘augment’ the visual field of the user with the information necessary in the performance of the current task. In simple words, AR refers to placing a digital object in the real world. A typical AR system fulfills the following three properties, namely, (a) combining real and virtual content in a real environment, (b) running in real-time and in an interactive manner, and (c) registering virtual content in the 3D environment [18]. Due to these properties, AR is considered as the upcoming way to display a variety of data in a more easily perceivable and interactable way. The stationary and mobile AR systems involve common material architecture like a camera filming the scene viewed by the user like a tablet (refer figure 10) or a semi-transparent helmet worn by the user like HoloLens glasses (refer figure 11), a computer to generate virtual entities, a numerical display, and sensors for sensing the position of the user and objects in a real environment.



Figure 10: Tablet scenario [19]



Figure 11: HoloLens glasses scenario [19]

A typical AR application consists of five modules, namely, registration, tracking, rendering, interaction, and content generation. In short, computer-generated information such as annotations, graphics, and 3D models, should be rendered and registered on the real scene with accurate tracking and alignment, followed by user-friendly interaction modes, such as gesture-based input, speech input, or with the help of external input devices, such as data gloves, ray casting using the mouse, etc. Lastly, relevant content in response to a specific request or task should be generated and displayed to the users. For the registration, a marker is placed on the real scene. A virtual object is overlaid on an AR marker which is recognized by the digital camera. The position and

orientation of the virtual object are determined by recognizing the position and orientation of the AR marker. Figure 12 shows one of the methods adopted for overlaying the virtual object in the real scene.

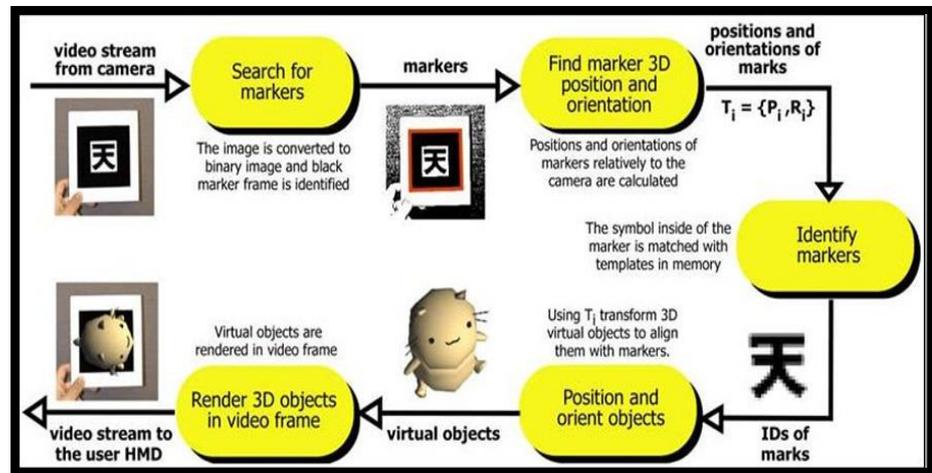


Figure 12: Procedure for overlaying virtual object on an AR marker [10]

AR systems have been widely used in different sectors because of their ability to bring the right information at right time. In the industries, AR systems are widely used in assembly operations (refer figure 13) and for maintenance. AR systems can also be useful in remote guidance, operator training, and production scheduling and monitoring (refer figure 14). Training is one of the favorite domains of an AR system because this technique allows double support real & virtual to the activity of the learner by giving him contextualized information. Hence, AR systems are most common in educational fields. AR can help visualize the phenomenon which is invisible to the naked eyes for example invisible flows on real objects like airflow or magnetic fields, in this way AR helps in better understanding the physical phenomenon. AR can also help in the understanding of complex drawings and assemblies.

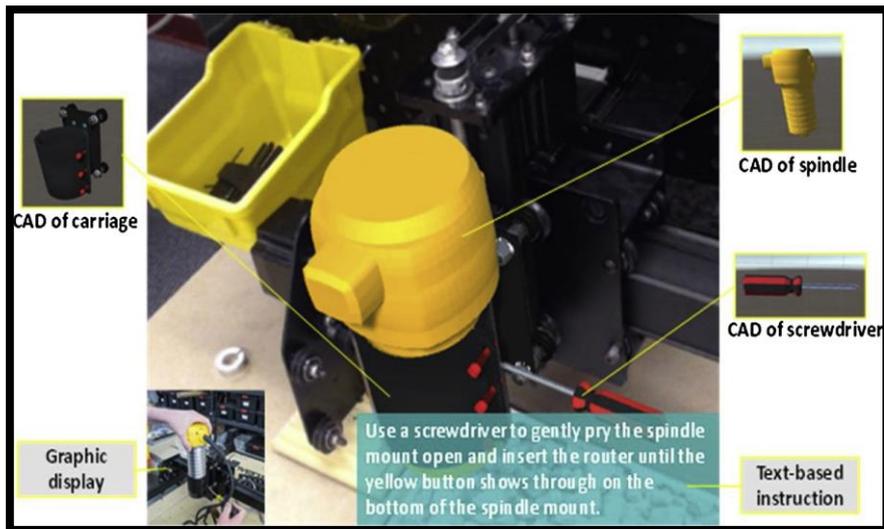


Figure 13: AR instructional system for assembly operations. Multiple types of instructions are rendered through display including texts, graphics, and animations [7]

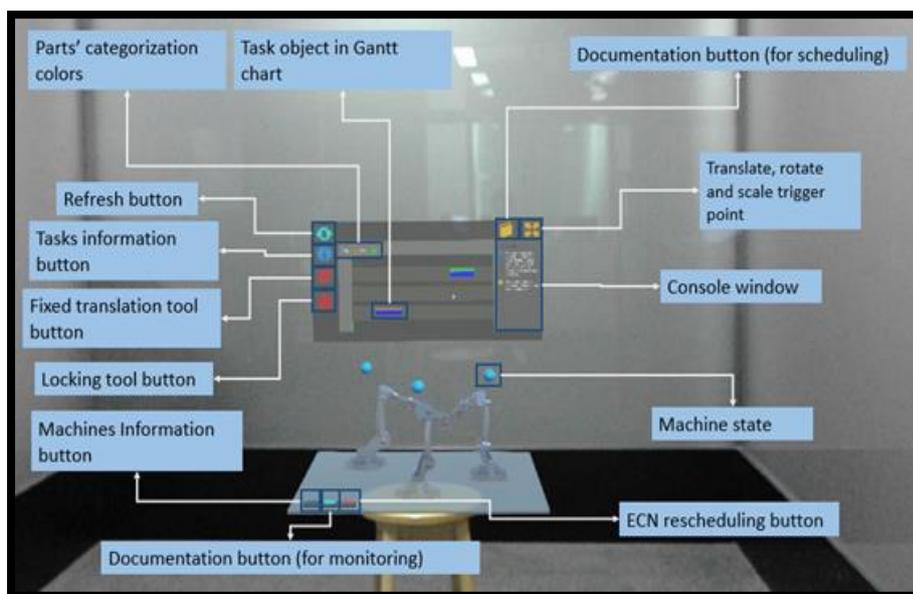


Figure 14: User interface developed with AR for production monitoring and scheduling [9]

AR and VR systems are important because of their ability to make work easier and make complex things better understandable. These technologies have already found their way in many different domains like medicine, the automotive industry, mechanical engineering, pilot training, etc. Even though AR and VR are emerging technologies for the

manufacturing field, but these technologies are here to stay. As an emerging technology VR and AR still faces several challenges like their user-friendliness needs to be improved, portability and robustness needs to be improved, more importantly, the software paradigm needs to be improved to seamlessly integrate these technologies in various industrial applications. Recent developments in AR and VR technologies have improved the quality of visualization along with making these systems more affordable. Various efforts and research are being conducted to develop smart manufacturing systems, such as “Industry 4.0” and “Factory of future” in manufacturing industries. Research to develop a cyber-physical system is being carried out and much research is focused on AR and VR technologies as fusion technology between physical and IT systems. Hence, AR and VR technologies are considered as advanced front-end technologies in Industry 4.0 for supporting smart working.

Chapter 4

Methodology

The methodology consists of complete detailed information regarding the modeling and development of virtual setup. The development procedure of the simulation algorithm which will run in the backend of virtual setup for generating simulated failure data is also discussed in the present chapter. The simulation algorithm along with the virtual setup will provide hands-on training with an accelerated life testing rig. Finally, the procedure for remote control and monitoring the actual setup using the virtual setup is discussed.

4.1 Selection of ALT Setup

The first important step in the methodology is to select the appropriate accelerated life testing setup for the validation of the proposed concept. Various setups are available for different components for the accelerated life testing and it is beyond the scope of this project work to develop a virtual setup for all of these available setups. Hence, one setup is selected for the validation of the concept. While selecting the setup all the challenges discussed previously are taken into the consideration and according to that most suitable setup is selected.

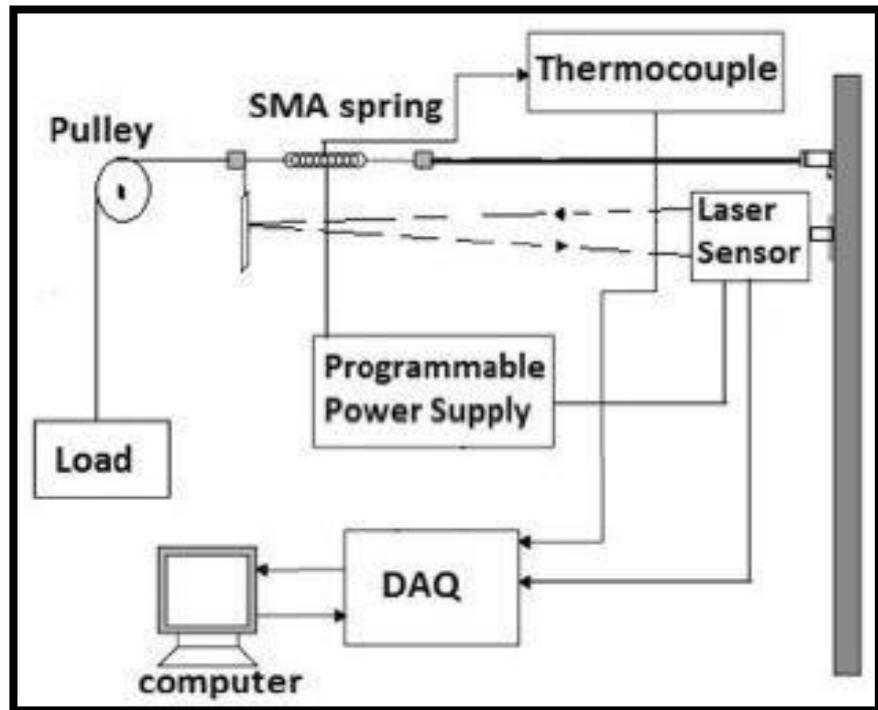


Figure 15: Schematic overview of ALT setup for SMA springs [20]

The accelerated life testing setup for shape memory alloys (SMA) springs is selected for concept validation. Fig. 15 shows the schematic diagram of the selected setup. ALT setup for SMA springs is selected for the validation of the concept because, as can be seen from figure 15 setup involves the components like programmable power supply, data acquisition systems, and laser displacement sensors which are costly components. Hence, the overall setup cost is very high. Most of the institutions will not be able to afford the building of this costly setup because of limited resources. Even with the building of this costly setup, it is not possible and not economically feasible for every student to perform accelerated life testing using this setup to gain hands-on training with this test rig. Also, the ALT setup for SMA springs is not remotely accessible, students and trainers are needed to be present in the lab for performing the tests using the setup. Due to all these challenges associated with setup ALT setup for the SMA springs is the best possible choice for the validation of the concept.

To understand the working of the accelerated life testing setup, it is important to understand the shape memory alloys. Shape memory alloy is an alloy that has a unique property that, it “remembers” its original shape, i.e., when it is deformed by applying an external force, it regains its original shape when some external stimulus like heating, magnetism, etc. is applied to it. This property of shape memory alloys is due to its crystalline phase change known as “thermo-elastic martensitic transformation”. This phenomenon is explained in figure 16.

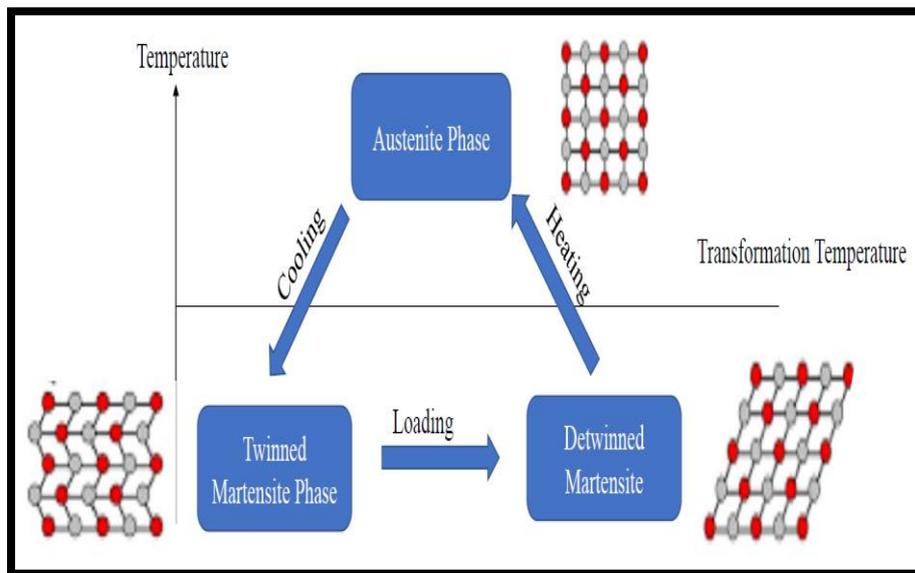


Figure 16: Thermo-elastic martensitic transformation [21]

At room temperature i.e., below transformation temperature spring material exists in the twinned martensite phase. At this phase, the spring material can be deformed to any shape by detwinning hence, when the load is applied to the spring material, it is converted to the detwinned martensite phase. When the spring is heated above transformation temperature, the material regains its original shape, as the detwinned martensite is converted to the austenite phase. After cooling to room temperature spring material is again converted to the twinned martensite phase below transformation temperature, which can be easily deformed by detwinning. This property of SMA spring makes them suitable for the applications like actuators, valves, etc. These self-actuating actuators

and valves are replacing conventional pneumatic and hydraulic valves and actuators in most applications due to their properties like lightweight, high power to weight ratio, ease of actuation, noiseless operation. Hence, shape memory alloys springs are used as an actuator or valves in various fields like robotics, biomedical, aerospace and automotive, etc.

4.2 Actual ALT Setup for SMA Springs.

To develop the virtual setup, it is necessary to understand the working of the actual setup. Also, to provide hands-on training with the setup, it is important to understand the experimental procedure to be followed on the actual setup. For this purpose, the actual setup for the accelerated life testing of SMA springs is built. The experimental procedure and the working of the actual setup are discussed in this section.

Figure 17 shows the actual experimental setup for the accelerated life testing of the SMA springs. The setup consists of NiTi (Nickel-Titanium) shape memory alloy spring, weights to provide external load, to keep the spring in the extended position using a pulley and copper wire as the spring is trained to contract upon actuation, programmable power supply (PPS) to supply specific energy required during heating and cooling cycles of spring, laser displacement sensor (LDS) to measure the micro-displacements during each cycle and data acquisition system to collect the analog signals from the LDS to convert them into digital values. LabVIEW (Laboratory Virtual Instrument Engineering Workbench) platform is then used for the data acquisition in the computer system. The output of DAQ is visualized in the LabVIEW application and the output is saved as an excel file to the computer.

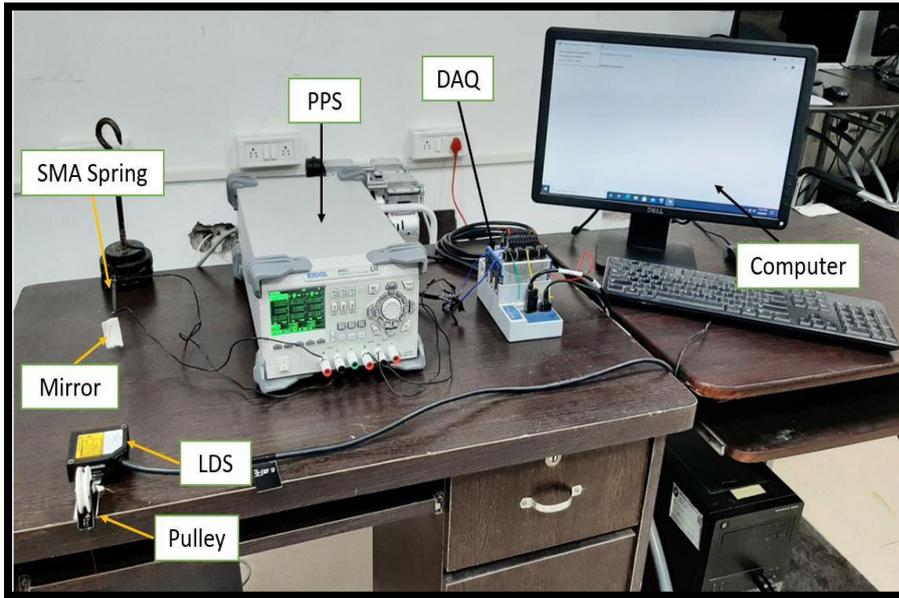


Figure 17: ALT setup for the SMA springs

The experimental procedure for the testing of the SMA spring involves the following steps, the first step is fixing the spring for the testing. For this purpose, the copper wire is used to fix one end of the spring to the table and the other end of the spring is connected to the dead weight using the copper wire and pulley. Next, supply cables are connected from channel 1 (+8V @ 5A) of the PPS across the SMA spring to generate the voltage necessary for heating the spring. To measure the micro-displacements during each cycle of operation LDS is placed between the pulley and the spring and the mirror is attached to copper wire passing over the pulley from which the laser beam is reflected to measure displacements. The DC voltage is used as the power source for the LDS for this purpose, supply cable is connected from LDS to channel 2 (+30V @ 2A) of the PPS. To convert analog signals from LDS to digital values LDS is connected to DAQ. These digital values from DAQ are stored in the computer with LabVIEW software, for this purpose, one cable is connected between the DAQ and CPU of the computer system. After making all of the above connections, PPS is switched ON. Channel 2 is selected to supply voltage to the LDS, the voltage value is set to 25V, and Channel 2 is switched ON. A ray of laser is projected

from LDS to mirror. Next, channel 1 is selected to apply a voltage across the spring. A rectangular waveform is generated for heating and cooling cycles using the timer function of PPS and supply through channel 1 is switched ON.

Initially, spring is kept in the extended position using the weights which is a deformed state for spring i.e., deformed martensite. When the voltage is applied across the spring, the spring is heated due to resistive heating. Due to heating, the spring regains its original shape due to the shape memory effect of lifting weights against gravity. This process is known as the actuation of spring. After the actuation, the power supply to the spring is cut off and the spring is allowed to cool to room temperature. During the cooling, the weights attached to the spring will force the spring back to its extended position. This process is called the release of the spring. This actuation and release combined will be considered as the one cycle of the operation. In the initial stages, multiple iterations will be performed to estimate the time required for the heating and cooling cycles. Based on the estimated times a rectangular waveform will be fed to the PPS to determine the ON-OFF times, so that setup can run on its own through the multiple cycles of operation until the spring fails. The calibrated LDS will be fixed to the platform to record the displacement of the spring. The measurements from the LDS will be recorded by DAQ. These data will be stored in the computer for further use.

After understanding the components and working of the actual setup, the next step will be to model these components to develop the virtual setup and to implement the functionality to the virtual setup to perform the experimental procedure discussed above on the virtual setup. This procedure will be discussed in detail in the next sections.

4.3 Development of the VR Based Setup

The procedure to develop the VR-based setup is divided into three sections. The first step includes the modeling of individual components of the setup which will be followed by working on the visuals of the setup and the last step will be to implement functionality to the virtual setup. All these steps are discussed in detail in this section.

4.3.1. Modelling the components of virtual setup using Blender 3D.

For modeling the individual components of the virtual setup Blender 3D software is used. Blender is an open-source 3D creation suite. Blender is chosen for the modeling of the setup components because models created using the blender can be easily imported into the Unity game engine which is used in the further development of the virtual setup. Blender is the dedicated software for 3D modeling, sculpting, UV mapping, and texturing to create realistic models. The setup equipment like data acquisition system (refer figure 18), programmable power supply (refer figure 19), SMA spring (refer figure 20), laser displacement sensor (refer figure 21), desktop (refer figure 22), pulley (refer figure 23), test fixture (refer figure 24), and weights (refer figure 25) are modeled using the blender. While modeling the emphasis was given that all the components look realistic and all the materials and textures are selected accordingly. Blender has versatile rendering systems called cycles and eevee which is used for the visualization of all the modeled components. Figures 18 through 25 show the rendered images of the modeled components of all setup equipment. These images are rendered using Blender's cycle rendering system. The selection of realistic materials and textures helps to create an immersive virtual environment that will improve the user's hands-on experience and will subsequently help the user while working on the actual setup. The next step is to export these models from blender to unity game

engine to develop a virtual experimental setup which is discussed in next section 4.2.2.



Figure 18: Modelled Data acquisition system



Figure 19: Modelled programmable power supply



Figure 20: Modelled SMA spring

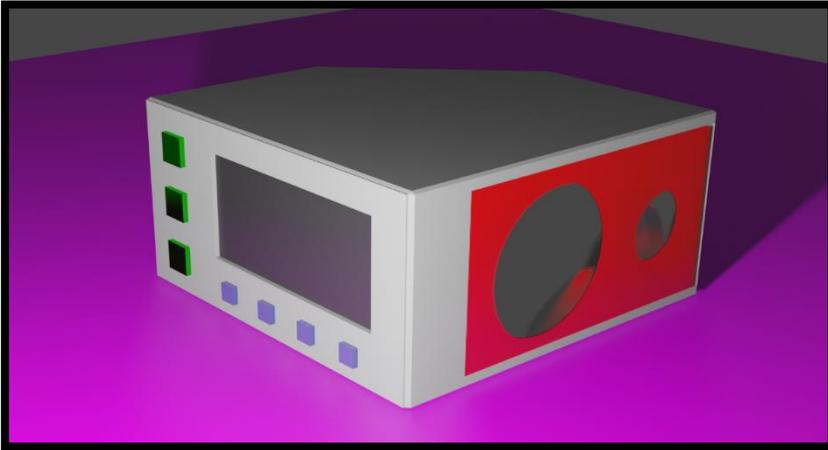


Figure 21: Modelled laser displacement sensor



Figure 22: Modeled desktop

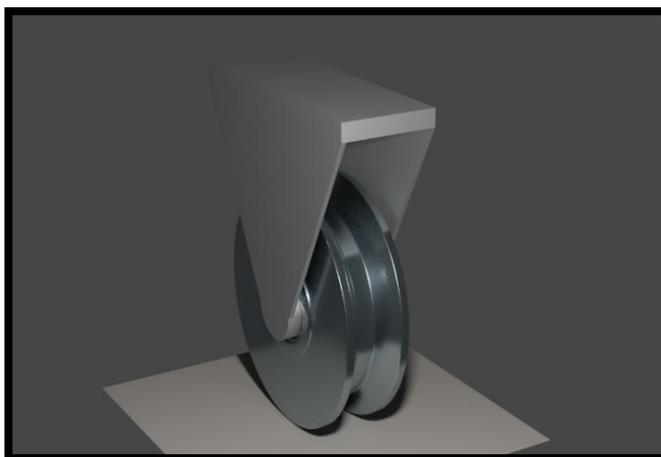


Figure 23: Modeled pulley

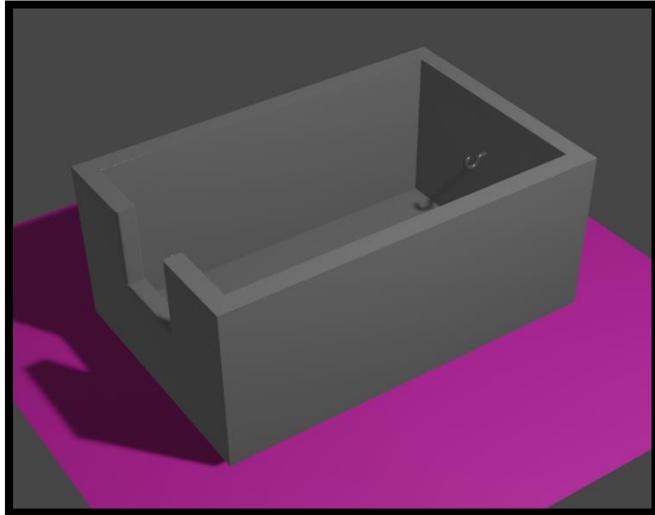


Figure 24: Modeled test fixture



Figure 25: Modeled weights

4.3.2. Working on the visuals of virtual setup in Unity Game Engine.

Before importing the modeled components in the unity game engine, the lab of appropriate size for the housing the setup with suitable lighting is modeled in the unity. After modeling of the lab is complete the auxiliary components like tables, wires, etc. are modeled in the unity game engine. The components modeled in the blender are imported into the

unity game engine. Unity 3D is the real-time 3D development tool and powerful cross-platform 3D engine. Unity also has its scripting language called C# for the implementation of various functionalities to the game objects in unity. After importing, the components are arranged and connected according to the requirement of the experimental setup. The wires are connected according to the schematic overview of the experimental setup to get the final view of the experimental setup as shown in figures 26 & 27.

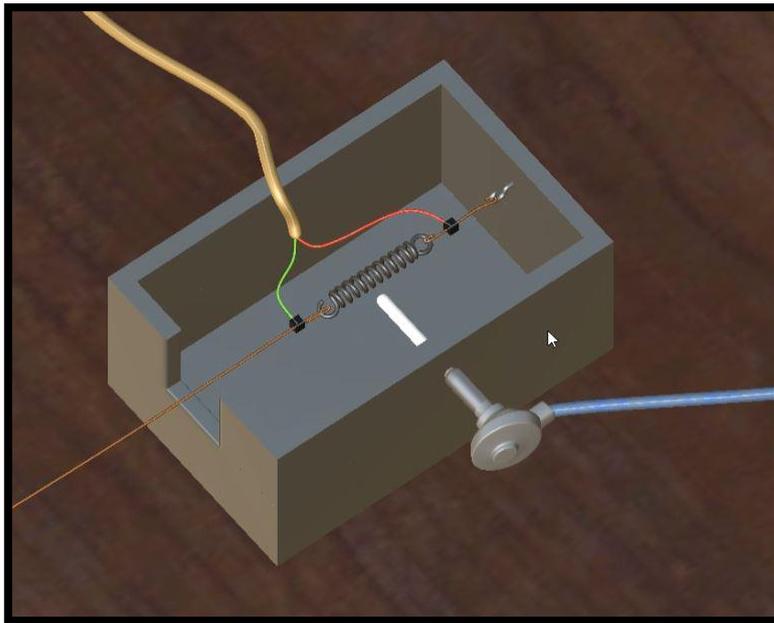


Figure 26: Connections of SMA spring

Figure 26 shows the spring box connections. One end of the spring is fixed to the hook and the other end is connected to the weights using the copper wire and pulley as shown in figure 27. The yellow wire is the power supply to generate voltages across the spring from the programmable power supply.



Figure 27: Overall experimental setup in the Unity Game Engine

4.3.3. Implementing functionality to the virtual setup.

To provide hands-on training with the virtual setup, the experimental procedure which is performed on the actual setup is needed to be performed in the virtual setup. Implementation of functionality to the virtual setup involves prompting the user to perform steps in a similar sequence and the same fashion as that of the actual setup. For this purpose, the user is needed to be guided through the experimental procedure in the virtual setup. The experimental procedure to be followed is discussed in section 4.2. The procedure for the implementation of functionality to the virtual setup is discussed in detail in this section.

For the user to be able to perform an experimental procedure in the virtual setup, virtual hands are used. Virtual hands (refer figure 28) help in grabbing the different gameobjects i.e., setup components like SMA springs, wires, etc. For the user to be able to grab required gameobjects rigid body is needed to be added to these gameobjects. To guide the user through various experimental steps text-based information (refer

figure 29 & 30) necessary in performing the current task is rendered in the visual field of the user. Various triggers are used to render appropriate and correct information in the visual field of the user. Along with the text-based information tutorial videos are also provided to improve the user's understanding of the setup. Tutorial videos related to a brief introduction of SMA springs (refer figure 31) and the working principle of LDS are included in the virtual setup.

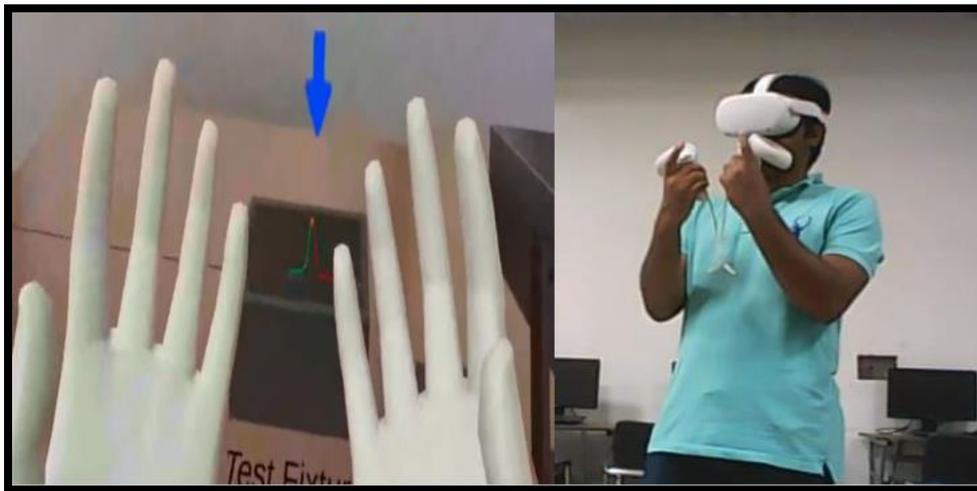


Figure 28: VR hands

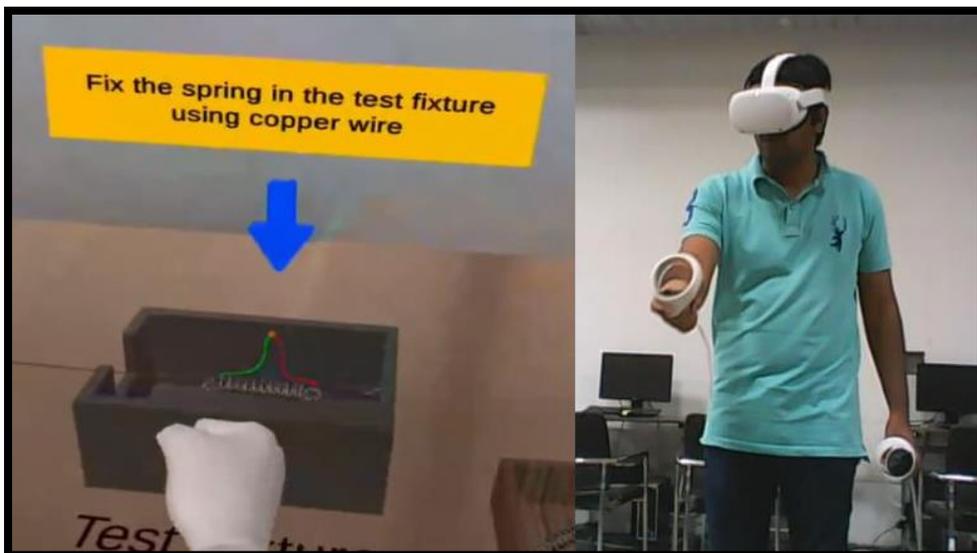


Figure 29: Fixing spring in the VR setup and text-based information

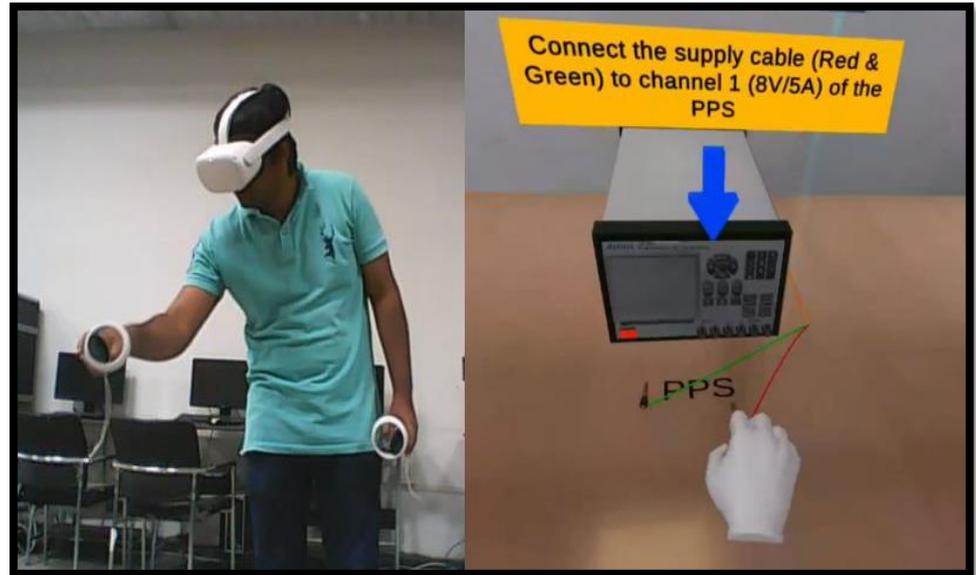


Figure 30: Connecting wires in VR setup



Figure 31: Tutorial videos in VR setup

The data acquisition in the actual setup involves the use of LabView software. The virtual setup also provides step-by-step information regarding the steps to be followed in the data acquisition process in the LabView (refer figure 32). Most of the ALT setups include the use of electronic equipment. The VR setup can also guide through the various steps required for the operating of these instruments in the setup. The

VR setup also includes a panel to provide the detailed specifications of the electronic equipment (refer figure 33 & 34).

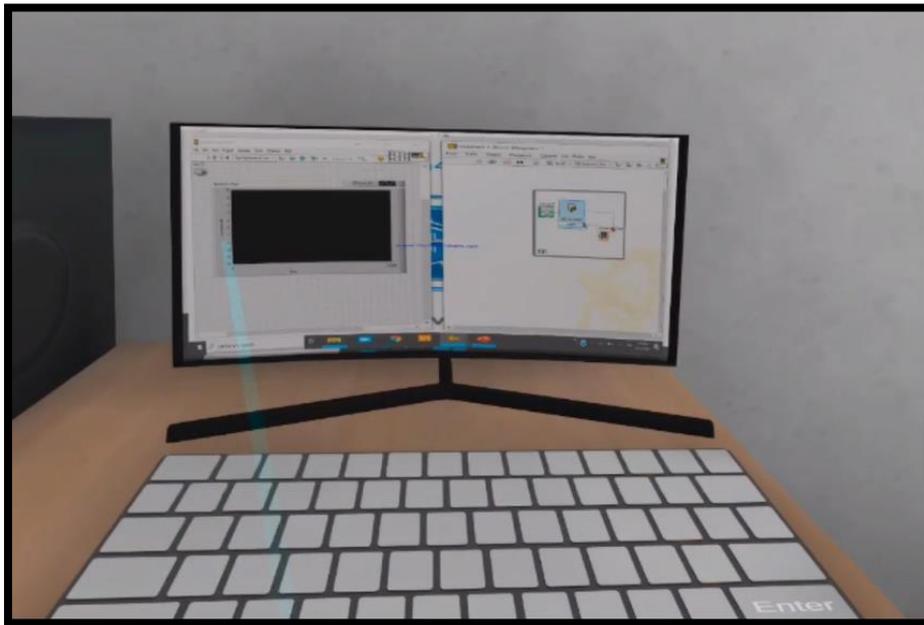


Figure 32: LabView software introduction in VR setup

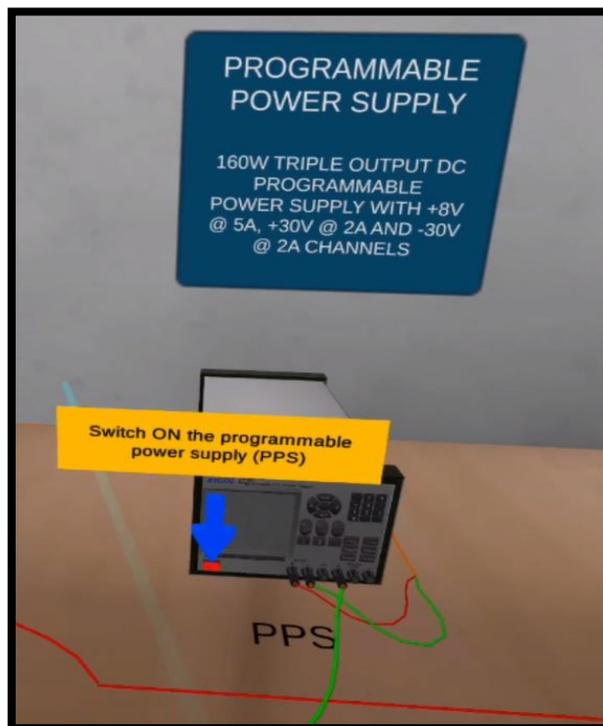


Figure 33: Operating electronic equipment along with detailed specifications

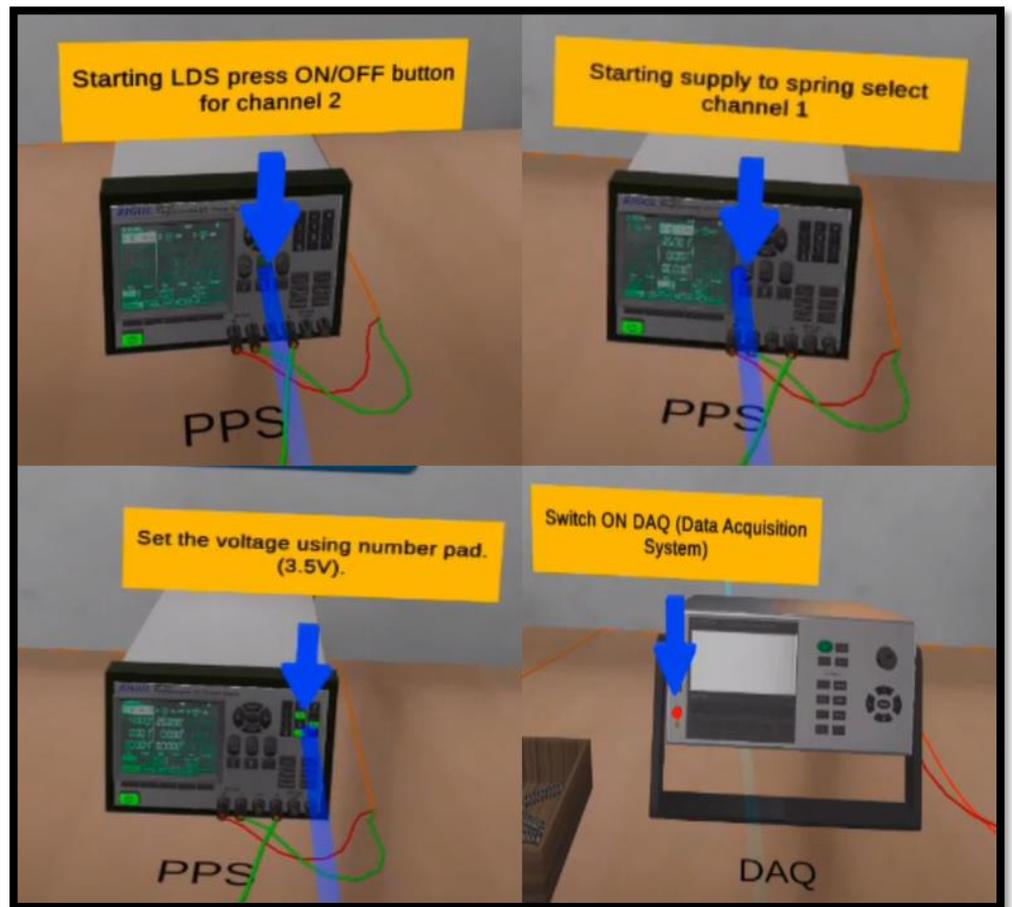


Figure 34: Operation guide for electronic equipment

After completing all the steps discussed in section 4.2 in the virtual setup, the component is subjected to testing. In the actual setup, the component follows a certain failure behavior after it is subjected to testing. To incorporate this failure behavior in the virtual setup a simulation algorithm is needed. The development procedure for the simulation algorithm is discussed in detail in the next section.

4.3 Developing Generic Simulation Algorithm.

While testing the component, a particular parameter is measured using the sensors to determine the degradation of the material under the testing, this parameter is known as the failure parameter. In some cases, when the failure parameter crosses some threshold value the component

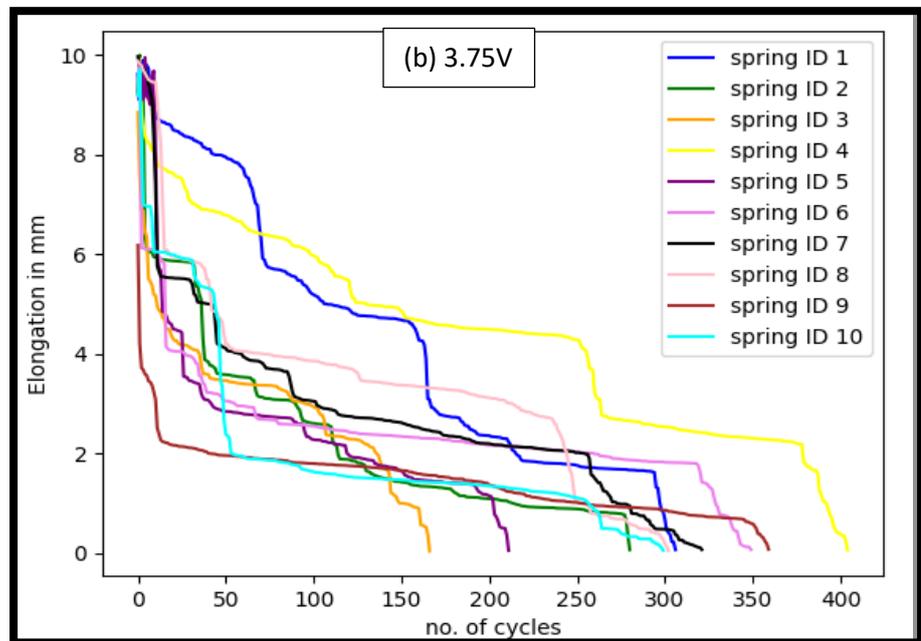
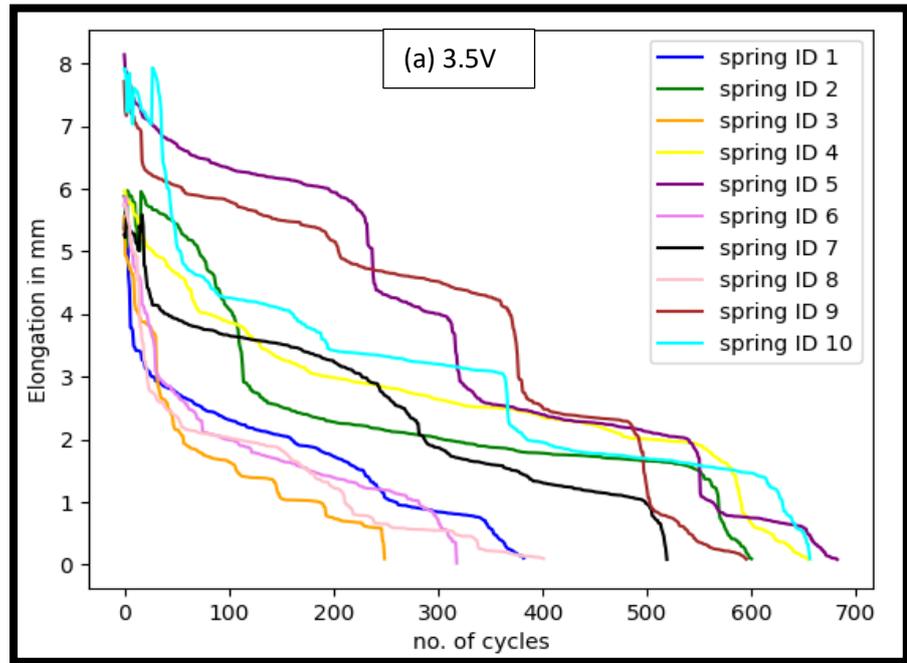
is considered to be failed. Hence, monitoring or measuring this failure parameter is important while performing the accelerated life testing of the components. This failure parameter determines the failure behavior of the component under test. To implement this failure behavior of the component under test in the virtual setup a simulation algorithm is necessary. The purpose of the simulation algorithm is to learn from past actual failure data to identify and extract the failure features in that data to generate simulated failure data. This data is then visualized in the virtual setup. Even though all the components are of the same specifications and subjected to testing under similar conditions, these components will not fail at the same time. The failure parameter which is measured during each cycle of operation will have different values for the different components. This is due to the stochastic nature of the failure involved in the fatigue failure behavior of the components. Hence, the purpose of the simulation algorithm is to simulate this failure behavior of the components under the test.

After making all the necessary connections in the virtual setup the component will be subjected to testing and simulated failure data generated by the simulation algorithm will be visualized in the virtual setup. The development procedure for the simulation algorithm is discussed in detail in this section. Since the ALT setup for the SMA springs is selected for the validation of the concept, the failure data for the SMA springs is used for developing the simulation algorithm. But due consideration is given so that the developed algorithm is generic and can be used for the different components under the test. The current section focuses on the detailed development procedure of the simulation algorithm based on the failure data of SMA springs.

4.3.1. Actual failure data of the SMA springs.

The actual failure data of the SMA springs consists of the elongation of springs in mm measured for each cycle for a particular value of the

voltage. NiTi SMA springs with the same specifications are tested at three different levels of voltage viz. 3.5V, 3.75V, and 4V. The actual failure data consists of failure data for 30 NiTi SMA springs with 10 springs tested at each level of voltage as shown in figure 35. The sample data of the elongation of 10 springs are shown in Tables 1 to 3.



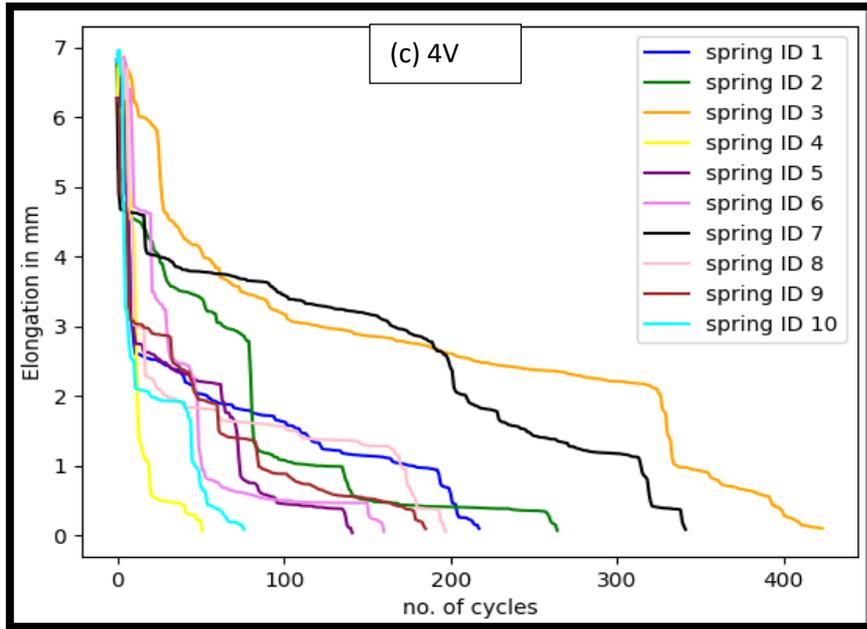


Figure 35: Actual failure data for SMA springs at (a) 3.5V, (b) 3.75V and (c) 4V

Table 1: Sample elongation data for SMA springs at 3.5V

Spring ID 1	Spring ID 2	Spring ID 3	Spring ID 4	Spring ID 5	Spring ID 6	Spring ID 7	Spring ID 8	Spring ID 9	Spring ID 10
5.3622	5.883	5.5596	5.9662	8.1417	5.7286	5.2652	5.753	7.7137	7.9116
5.6321	5.8112	4.9342	5.9527	7.8820	5.8708	5.2214	5.7136	7.2516	7.9174
5.3276	5.0754	4.9204	5.8474	7.8645	5.7276	5.5344	5.6597	7.1665	7.6535
5.3002	5.9724	4.8754	5.8263	7.7815	5.4503	5.4277	5.5449	7.3089	7.2211
4.6222	5.939	4.8075	5.7825	7.7105	5.3772	5.4050	5.4796	7.2894	7.5908
4.4032	5.8967	4.7265	5.7752	7.7054	5.2249	5.3591	5.3955	7.2625	7.8419
3.7866	5.8860	4.7246	5.7711	7.6192	5.0969	5.3206	5.1687	7.2184	7.3285
3.7838	5.8810	4.6477	5.7606	7.5094	4.9994	5.3099	5.0823	7.1984	7.2999
3.7333	5.7978	4.6442	5.7433	7.4624	4.9812	5.2942	5.0191	7.1854	7.0311
.
.
.

Table 2: Sample elongation data for SMA springs at 3.75V

Spring ID 1	Spring ID 2	Spring ID 3	Spring ID 4	Spring ID 5	Spring ID 6	Spring ID 7	Spring ID 8	Spring ID 9	Spring ID 10
9.6174	9.2487	8.8498	9.1915	9.2589	9.1992	9.9705	9.8792	6.1802	9.1874
9.0886	9.9967	7.6206	9.9299	9.8245	9.8870	9.7969	9.8138	4.2161	9.7034
9.8944	9.6311	6.8792	8.8493	9.4088	6.1285	9.7881	9.7887	3.7160	8.2752
9.8875	9.1615	6.8690	8.5738	9.1036	6.1229	9.7437	9.6949	3.6790	6.9989

9.8265	6.3129	6.4433	8.3815	9.9511	6.1058	9.6859	9.6406	3.5902	6.9832
9.7927	6.0989	6.3568	8.2863	9.1648	6.1026	9.6189	9.5723	3.5667	6.9628
9.7243	6.0494	5.5030	8.2479	9.1255	6.0998	9.4199	9.5026	3.4530	6.96242
9.0980	6.0490	5.4550	8.1846	9.0049	6.0993	9.2789	9.4809	3.4322	6.96010
.
.
.

Table 3: Sample elongation data for SMA springs at 4V

Spring ID 1	Spring ID 2	Spring ID 3	Spring ID 4	Spring ID 5	Spring ID 6	Spring ID 7	Spring ID 8	Spring ID 9	Spring ID 10
6.8292	6.7536	6.6923	6.0415	6.2721	6.9073	6.2741	6.9369	6.270	6.8077
6.3432	5.2590	6.672	6.7816	6.2412	6.8536	4.9192	6.8782	6.213	6.9702
6.1612	5.2548	6.2869	6.7262	6.2365	6.7991	4.6793	6.8376	4.926	6.6473
6.1492	5.1212	6.2655	6.875	6.7534	6.7982	4.6691	6.8269	4.8138	6.5893
6.72976	5.0801	6.7789	6.7303	5.9329	6.8655	4.6585	6.4165	4.7676	4.4070
6.14376	4.9867	6.7775	6.3809	4.9123	6.7739	4.6507	6.2710	4.7422	3.2667
5.26457	4.9322	6.7017	5.0745	4.8684	6.2789	4.6493	6.7043	4.6100	3.1881
4.21943	4.7463	6.6573	5.0685	3.5218	6.2317	4.6474	4.5957	3.3032	2.7919
3.95131	4.6470	6.6060	4.767	3.3767	6.4094	4.6373	4.5619	3.0924	2.5548
.
.

From figure 35, it can be seen that even though all the springs are of the same specifications and 10 springs each are tested at the same voltage, the failure times for each spring are different. Also, elongation values for a particular cycle are different for each spring. This is due to the stochastic nature of failure involved in the failure behavior of SMA spring. Hence, the purpose of the simulation algorithm for this data is to simulate the thermo-mechanical fatigue behavior of SMA springs.

4.3.2. Features of actual failure data.

To understand typical features of actual failure data different failure datasets were studied like thermo-mechanical fatigue failure data of SMA springs, Wear data of milling cutter, and RMS vibration data of milling cutter. After studying these datasets, it was found that three

important features are present in every failure data and these are failure trend, noise, and abrupt jumps. Hence, these features were selected for the generation of the simulated failure data. Therefore, the purpose of the simulation algorithm is to identify and extract these failure features like failure trends, inherent noise, and abrupt jumps from actual failure data to generate simulated failure data.

4.3.3. Time to failure (TTF) of simulated component

The first step in the generation of simulated failure data is to find the life i.e., time to failure of the simulated component. The life of the simulated component can be expressed as the number of cycles or time in seconds, minutes, hours, or days for which the component will survive. In the case of SMA springs, it is expressed as the number of cycles for which the spring will survive. Table 4 shows the TTF data for 30 springs at three different levels of voltage.

Table 4: Time to failure data of SMA springs

Spring ID / Voltage	1	2	3	4	5	6	7	8	9	10
3.5V	384	601	250	655	683	319	520	402	596	657
3.75V	307	281	167	405	212	350	322	303	360	300
4V	218	265	424	52	142	161	342	198	186	77

From table 4, it can be seen that each spring failed at a different time even though the voltage applied is the same. To determine the time to failure of the simulated component probability distribution is fitted to lifetime data of the component. In reliability engineering, the Weibull distribution is the most widely used distribution for lifetime data. Weibull distribution is selected for determination of the life of simulated component because it takes into account all the three different types of failure rates viz. increasing, decreasing, and constant failure rate.

Weibull distribution can be 2 parameters or 3 parameters Weibull distribution. The algorithm uses two-parameter Weibull distribution for lifetime data. Two parameters involved are the scale parameter (η) and shape parameter (β). The scale parameter is also known as characteristic life it is the time at which reliability is 36.8% or percentage failures are 63.2%. The shape parameter gives information regarding the failure rate of the component. Table 5 shows the failure rate for different values of β .

Table 5: Failure rate for different values of β

Shape Factor (β)	Failure Rate
$\beta < 1$	Decreasing failure rate
$\beta = 1$	Constant failure rate
$\beta > 1$	Increasing failure rate

In the simulation algorithm, these parameters of the Weibull distribution are determined using the maximum likelihood estimation. The reliability function for the Weibull distribution is given by equation 1.

$$R(t) = e^{-\left(\frac{t}{\eta}\right)^\beta} \dots (1)$$

Where $R(t)$ is the reliability of the component.

η & β are the estimated parameters of the fitted distribution.

t is the time to failure (TTF) of the simulated component.

Hence, by selecting any random value of reliability between 0 to 1 and substituting estimated parameter values in equation 1 the TTF of the simulated component is determined in the simulation algorithm.

4.3.4. Estimation of random start mean

The random start mean is the starting point for the failure trend of the simulated component. The random start is mean is the initial value of displacement for the SMA springs. The random start mean is determined

using the initial values of displacement in the actual failure data. Ideally, the initial value of the failure parameter must be the same for the component with the same specifications and manufactured under similar manufacturing conditions. But any manufacturing process cannot produce the component with the exact dimensional accuracy. For this purpose, the concept of tolerances is introduced in manufacturing. Therefore, it is impossible to produce the mechanical component with exact specifications and there are deviations in the manufacturing conditions during manufacturing. Because of all of these parameters initial values of failure parameters are different for different components with the same specifications which can also be seen from the scatter plot of the initial values of displacement of springs in figure 36. All of these deviations that are involved in the manufacturing process and operating conditions are always centered around the actual value of operating parameters which can also be seen from the box plots shown in figure 37.

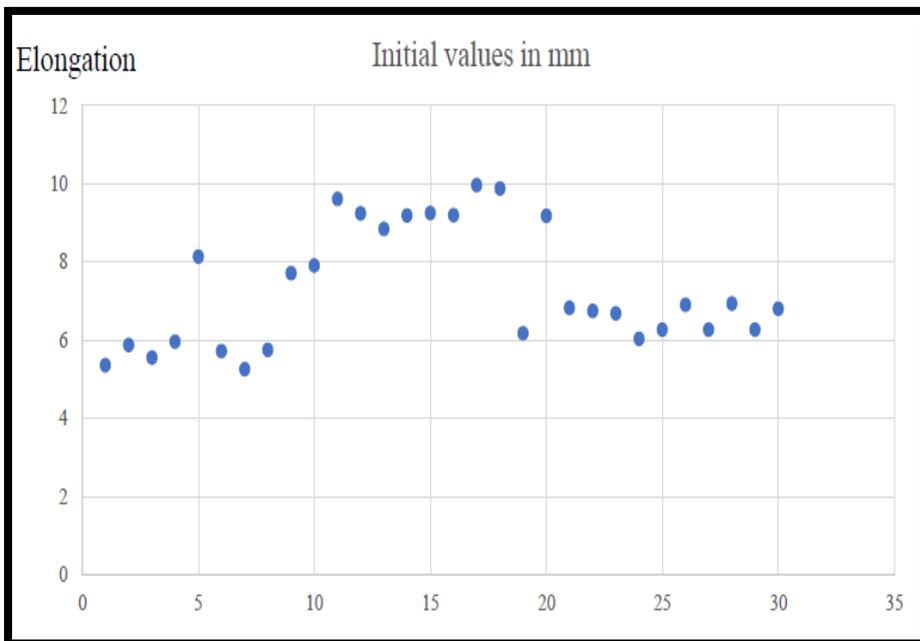


Figure 36: Scatter plot of initial values of elongation

From figure 37 it can be seen that there is a small variation in the initial values of elongation for the spring tested at each level of voltage like

3.75V and 4V. Even though there is a slight variation in initial values of displacement for the springs tested at 3.5V, more than 60% values of elongation lied between 5.5mm to 6mm which is evident from the box plot.

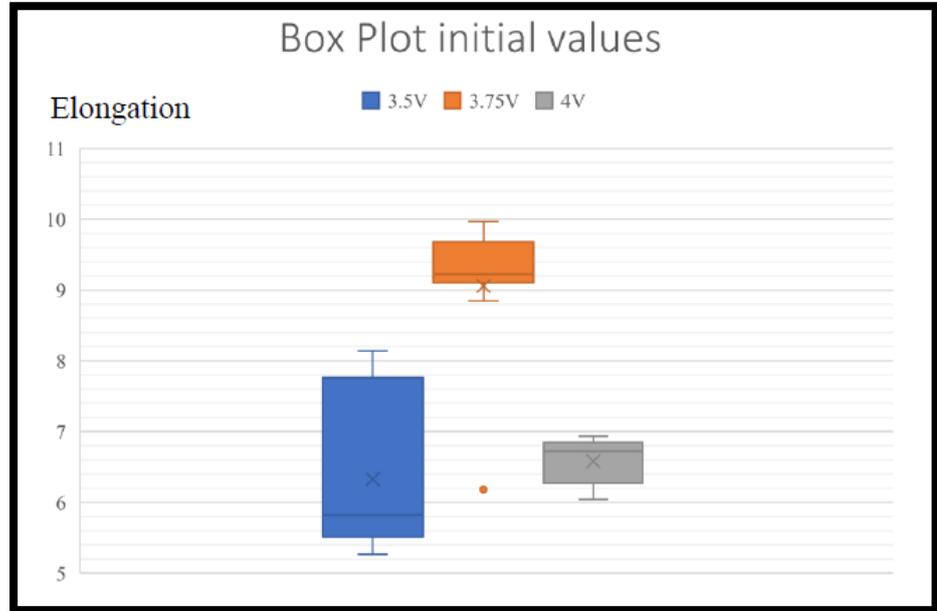


Figure 37: Variation in the initial values of elongation of spring

From the box plot in figure 37 it can be seen that despite small variations all the initial values of elongation are centered around the true value hence, the normal distribution is fitted to the actual values of initial displacements. Equation 2 shows the equation used in the simulation algorithm for the estimation of the initial value of displacement for the simulated component.

$$\text{Random Start Mean} = \text{normalinverse}(\text{rand}(0, 1), \mu, \sigma) \dots (2)$$

In equation 2, μ is the mean of initial values of elongation
 σ is the standard deviation of initial values of elongation.

4.3.5. Estimation of threshold

The threshold is the endpoint of the trend line of the simulation. In the simulation algorithm, while selecting the threshold of the simulation

both noise and the abrupt jumps in the actual data are taken into the consideration. The threshold point is not the endpoint of the simulation, it just acts as a reference for obtaining the best-fit trend line for the simulated data. Equation 3 shows the formula for the threshold.

$$\text{Threshold} = \min \text{ or } \max (\text{final values}) \mp \text{avg. jumps} \\ * \text{jump threshold} \dots (3)$$

In equation 3, the minimum of final values is selected for the increasing trend, and the maximum of final values is selected for decreasing trend to account for the noise in data. The trend is increasing if the random start mean is less than the threshold and decreasing if the random start mean is greater than the threshold. Avg. jumps are the average number of jumps in the actual data and the jump threshold is determined in the simulation of abrupt jumps which is explained in a further section. To account for jumps, the positive sign in equation 3 is used for decreasing trend and the negative sign is used for increasing trend.

4.3.6. Trend simulation

The trend line is the line connecting the random start mean and threshold for the simulated data. To find the best-fit trend line first mathematical multiplier is calculated using equation 4.

$$k = \frac{\text{Threshold} - \text{Random Start Mean}}{TTF^n} \dots (4)$$

If the value of the mathematical multiplier (k) is positive then the trend is increasing with time and if k is negative then the trend is decreasing with time. To find values of failure parameter at each timestamp change in the mean is calculated using equation 5.

$$\text{Change In Mean (CIM)} = \text{Random Start Mean} + k * TS^n \dots (5)$$

In equation 5, TS is the time stamp that varies between 1 to TTF. The equation of change in mean generates simulated values of elongation for the springs at each timestamp.

Both equations 4 & 5 contain exponent n , the effect of this exponent n on the simulated data is as shown in figures 38 to 40. For $k < 0$, if the value of exponent n is less than 1 then, the trend line with the trend points decreasing at decreasing rate is created as shown in figure 38. For $k < 0$ & $n = 1$, the trend line with the trend points decreasing at a constant rate is created as shown in figure 39. For $k < 0$ & $n > 1$, the trend line with trend points decreasing at an increasing rate is created as shown in figure 40. The value of n is varied to obtain the best fit line for the trend line to generate the simulated failure data.

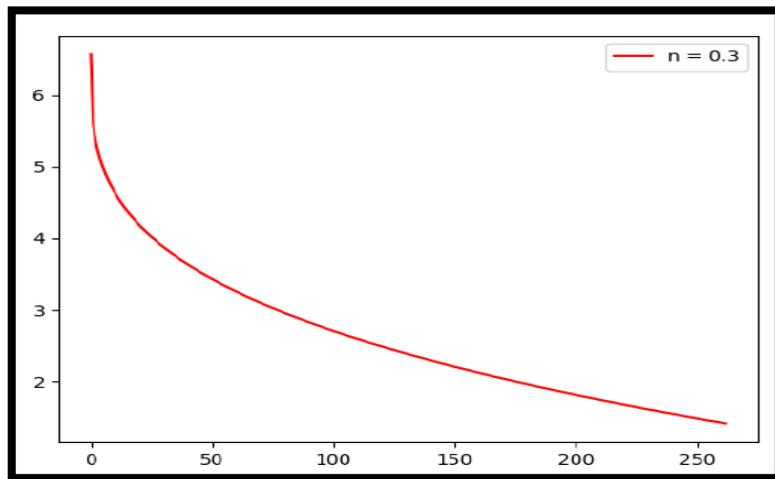


Figure 38: Trend line for $n < 1$ & $k < 0$

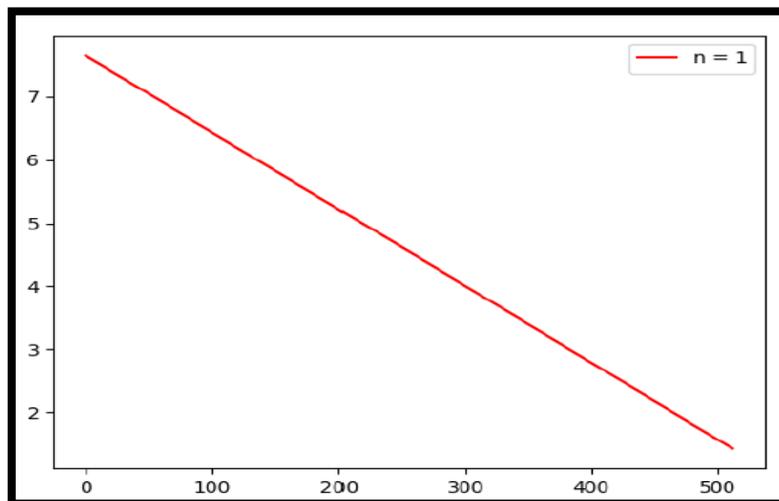


Figure 39: Trend line for $n = 1$ & $k < 0$

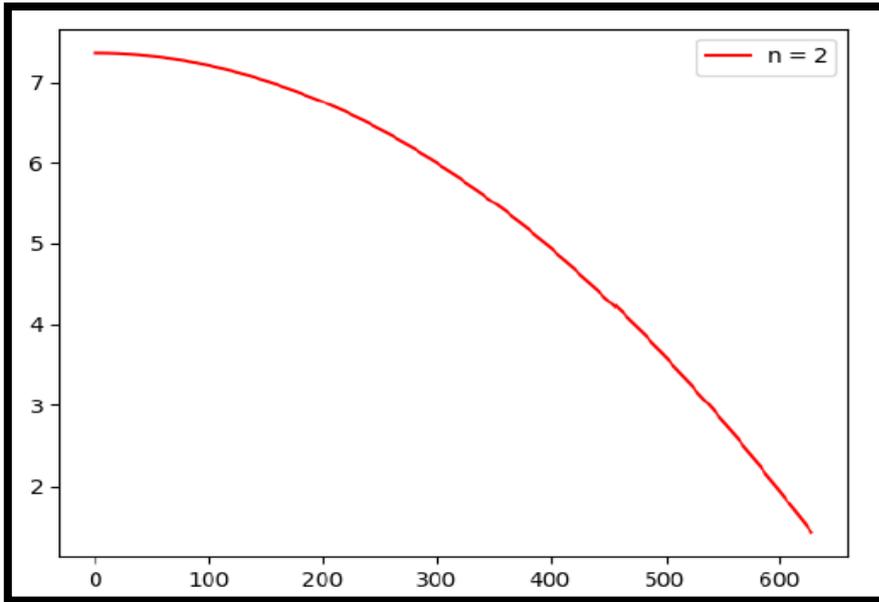


Figure 40: Trend line for $n > 1$ & $k < 0$

The value of n for the best fit line will depend on the random start mean, threshold, and time to failure estimated for that particular simulation.

4.3.7. Incorporation of noise in simulated data.

While performing the accelerated life testing certain surrounding conditions cannot be controlled. Due to these variations in the surrounding conditions, there is some unevenness in the trend line of the actual data as shown in figure 41.

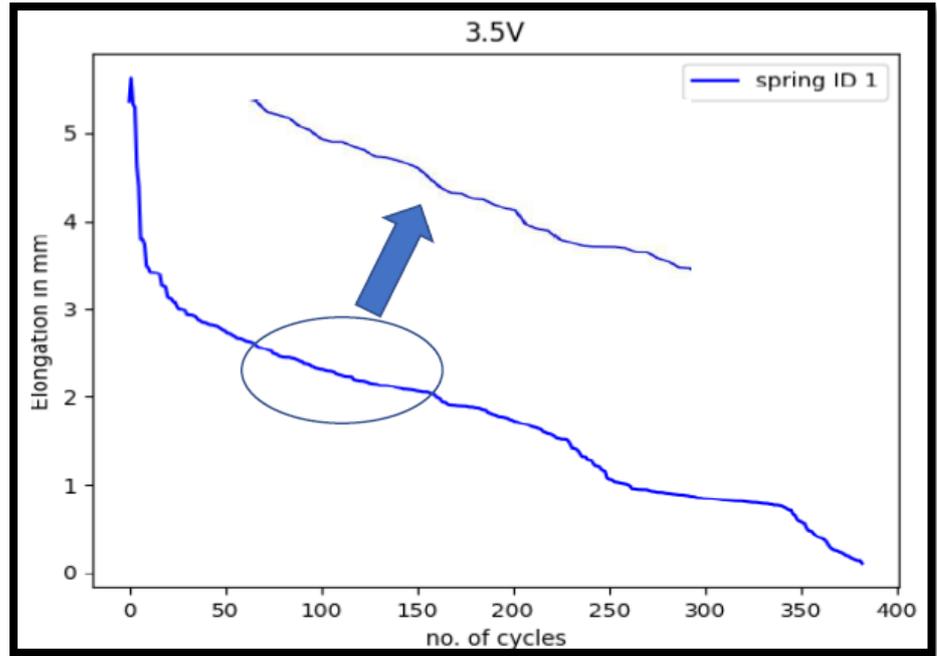


Figure 41: Noise in the actual data

In the fig. 41 when we zoom-in in the encircled portion we see unevenness in the trend line. This unevenness in the trend line is due to the noise that is present in the actual data. Noise is an inseparable part of actual failure data. The presence of noise in the data may be due to inaccuracies in the measurement of the sensors due to uncontrollable conditions. This noise is random. Hence, noise is incorporated in the simulated failure data in the algorithm in a random fashion. For this purpose, the inverse normal function is used which is shown in equation 6.

Noise Addition

$$= \text{normalinverse}(\text{rand}(0, 1), CIM, CIM * A) \dots (6)$$

In equation 6, change in mean (CIM) values at each timestamp are used as the mean of the distribution, and (CIM * A) is considered as the standard deviation of normal distribution. Where, A is the maximum percentage deviation of the trend point i.e., elongation value from the respective trend line for the actual data.

4.3.8. Incorporation of abrupt jumps in the data.

As the component undergoes testing, the degradation of the failure parameter exhibits multi-phase features because of sudden shocks and physical mutations. These sudden changes in the failure parameter sometimes may be due to changes in the system, surrounding, or operating environment. If some failure data exhibits such features, they must be taken into account by the simulation algorithm and must be incorporated in the simulated failure data. The algorithm uses the following methodology to incorporate abrupt jumps in the simulated failure data. First, the slopes are calculated at each timestamp i.e., the failure parameter value at a particular timestamp is subtracted from the previous timestamp. Next, the jump threshold is calculated. The jump threshold is the minimum value of slope from the actual data that is considered as a jump. The jump threshold is determined by the formula given in equation 7.

$$\text{Jump threshold} = \min(\text{initial elongation} - \text{final elongation}) * \text{Jump factor} \dots (7)$$

Equation 7 consists of the jump factor; the jump factor determines what minimum percentage of the total slope is considered as jump.

After calculation of the jump threshold, all the slopes with slope values greater than the jump threshold are considered as jumps. Based on this jump probability is calculated using equation 8.

$$\text{Jump Probability} = \frac{\text{No. of slopes with slope value greater than jump threshold}}{\text{Total number of slopes}} \dots (8)$$

The jump probability is calculated because all the abrupt jumps involved in the actual failure data are random. Figure 42 shows elongation Vs the number of cycles plot for 10 springs tested at the same voltage level.

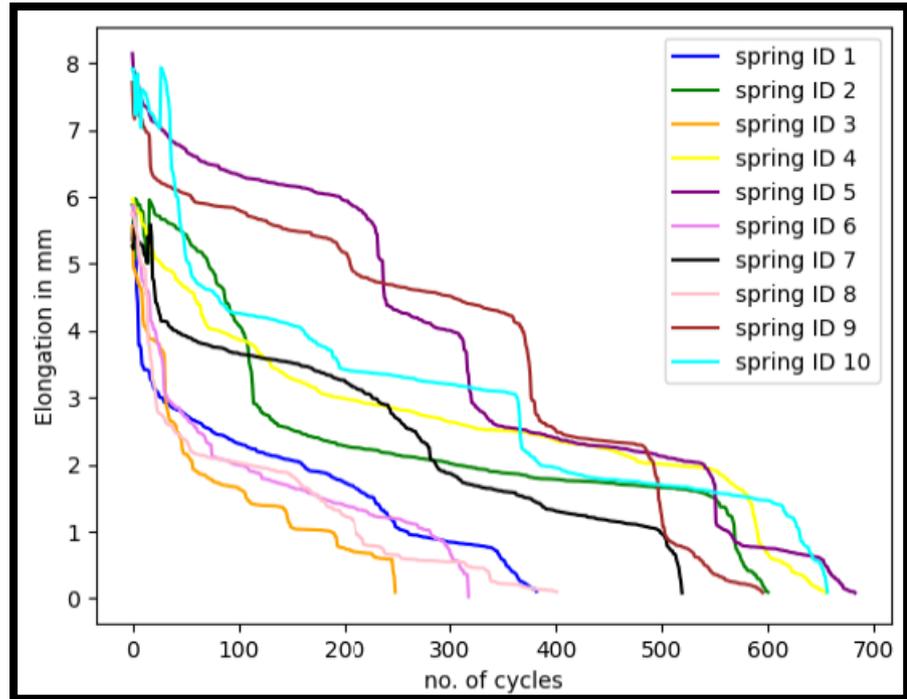


Figure 42: Random nature of abrupt jumps in the data

From figure 42 it can be inferred that there is no fixed elongation value at which these jumps occur and also there is no fixed number of cycles for which these jumps occur. Hence, these jumps are random. Hence, the jumps are added randomly in the simulated failure data with the probability equal to jump probability. After the incorporation of jumps the simulated failure data is generated which is the output of the simulation algorithm.

The output of the generic simulation algorithm is the simulated failure data, which will be further visualized in a virtual setup. For this purpose, the algorithm is needed to be interfaced with the virtual setup to run in the back-end of the setup to generate simulated failure data for the component under the test.

4.4 Interfacing Simulation Algorithm with Virtual Setup.

For interfacing of the simulation algorithm with the virtual setup an application programmable interface i.e., API is developed. API is a set of programming codes that enables data transmission between one software application and another. API is used when users need to access data in real-time. API acts as a programming interface where the user doesn't need to understand the working of the particular software, the user just sends a request for the data which is required for his particular application. This request is then handled by the software's API which performs complex functions to generate the required data which is then sent to the user. To understand the working of API, it is important to understand the basic working of the web. Figure 43 shows the basic working of the web.

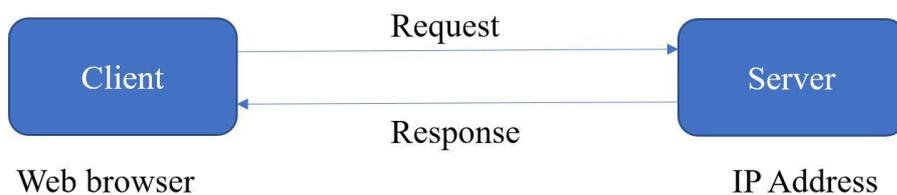


Figure 43: Working of the web

The client first sends a request to the server, the two basic types of requests are GET request and POST request. A GET request is used to retrieve the data from the server whereas a POST request is used to send the data to the server. Each server has its specific IP address to which the request is sent. Based on the request from the client, the server sends a response to the client, the response may be anything image, texts or data, etc.

For the present application, an API is created using the 'flask' module of python, and the data generated from the simulation algorithm is uploaded to the localhost server which can be accessed by using its IP address. To incorporate this data in the virtual setup, another API is

created which will send the voltage and email data. This data is then processed by the simulation algorithm's API and corresponding to the received voltage the simulation algorithm will generate simulated failure data which will be then sent to the virtual setup's API for the visualization. Figure 44 elaborates the data transfer process between the two APIs.

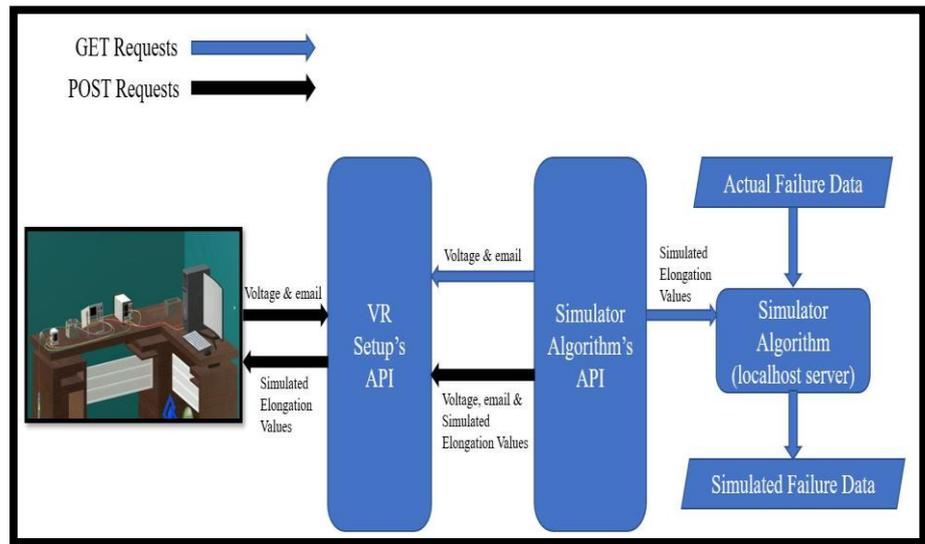


Figure 44: Data transfer process between two APIs

As shown in fig. 44, first the POST request is sent from the VR setup to the VR setup's API to update voltage value and email address. The simulator algorithm's API sends the GET request to fetch the voltage and email data. After receiving the voltage value Simulator algorithm's API sends the GET request to the simulator algorithm server to get the elongation values, then it sends the POST request to VR setup's API to send the simulated elongation values, then this API sends POST to request to update these elongation values to the VR setup. The next step is the visualization of these simulated values of elongation in the VR setup.

4.5 Visualization of Simulated Failure Data in Virtual Setup.

The data received from the API is in the form of the JSON (JavaScript Object Notation) file. It needed to be parsed in unity. After parsing and formatting the data from the simulation algorithm is ready for visualization. When the SMA spring is subjected to the testing, it undergoes multiple heating and cooling cycles during which it contracts and expands respectively. The data obtained from the simulation algorithm is the elongation value obtained after each heating and cooling cycle. These simulated elongation values are needed to be displaced in the virtual setup. These values are displayed on the screen of LDS after each cycle of operation as shown in figures 45 to 47. After a certain number of cycles, the elongation value falls below the threshold and the spring is considered to be failed. The number of cycles after which spring fails is the life of the simulated component, it is displayed in the virtual setup as shown in fig. 48. Similarly, the simulated elongation values are generated at three different levels of voltages and are shown in figures 49 & 50.

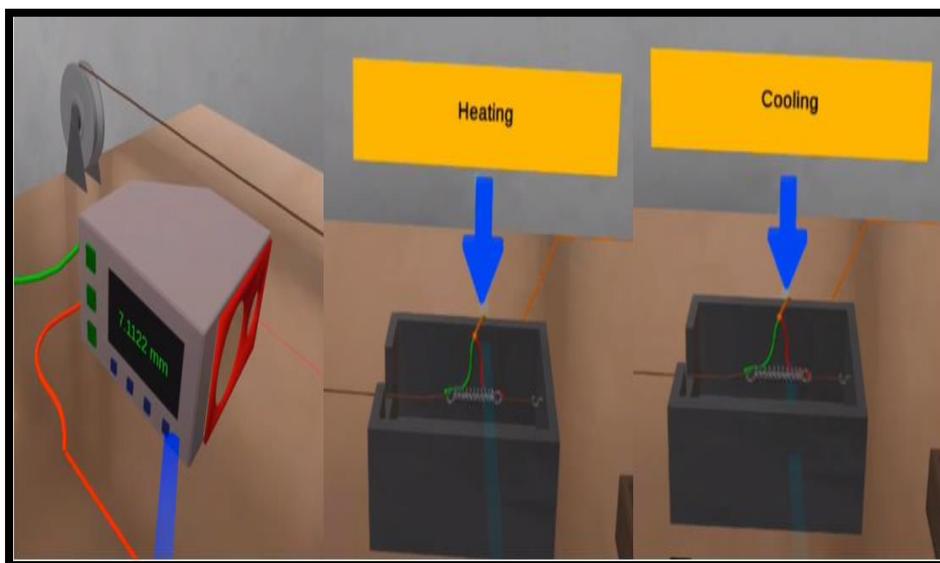


Figure 45: Simulated elongation value for 1st cycle at 3.5V

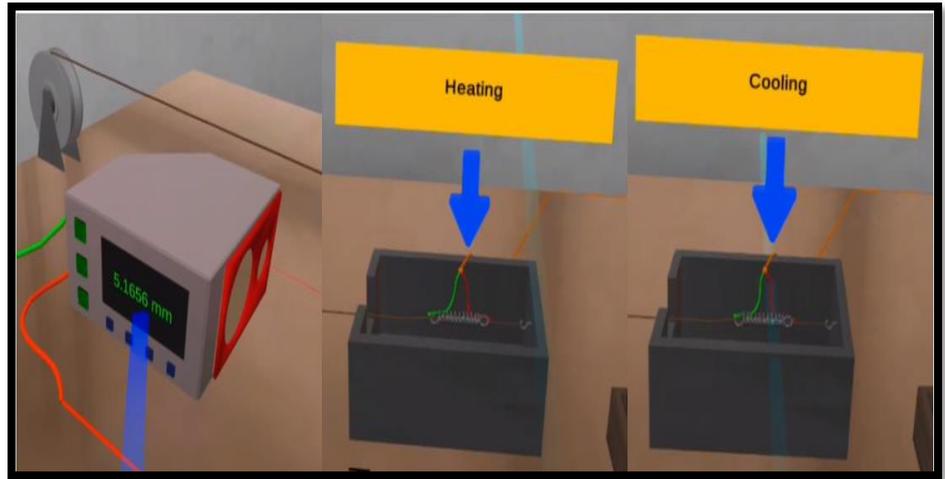


Figure 46: Simulated elongation value for 2nd cycle at 3.5V

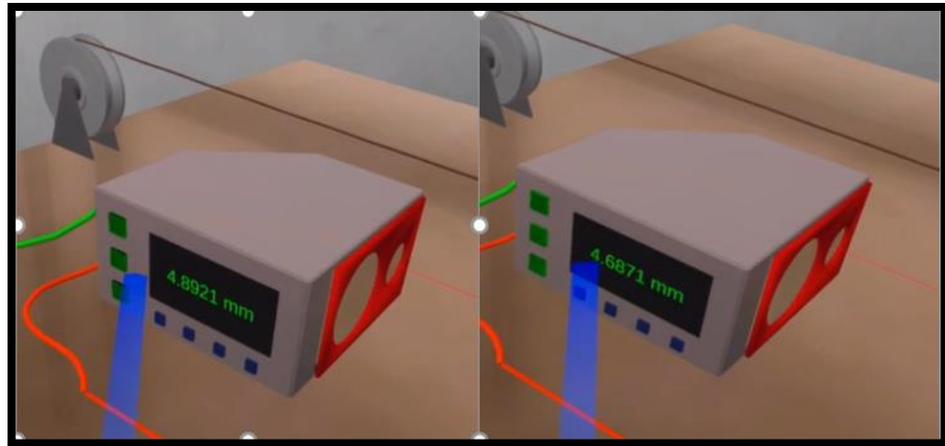


Figure 47: Simulated elongation value for 3rd & 4th cycle at 3.5V

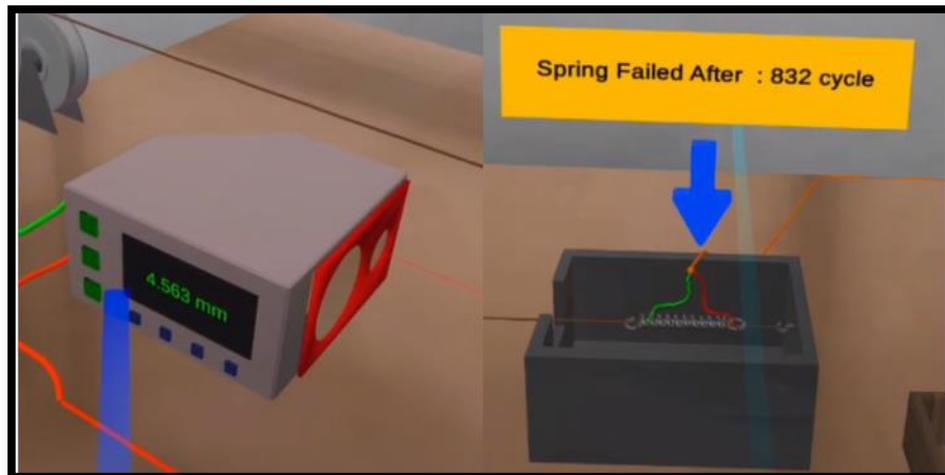


Figure 48: Simulated elongation value for 5th cycle and life of SMA spring at 3.5V

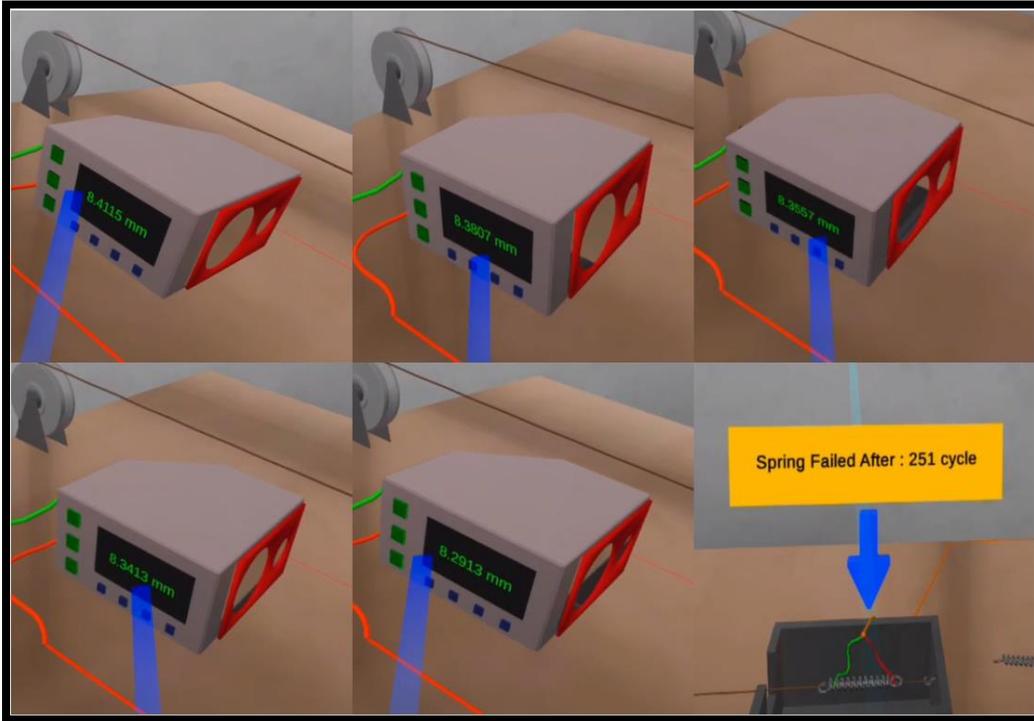


Figure 49: Simulated elongation values and life of virtual spring at 3.75V

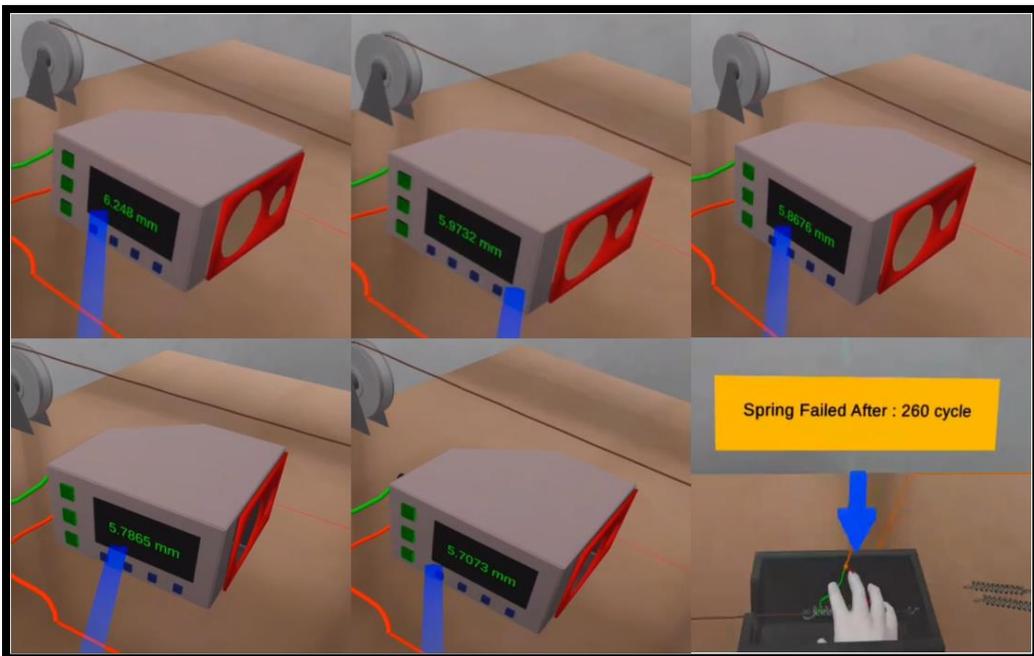


Figure 50: Simulated elongation values and life of virtual spring at 4V

As shown in figures 45 to 50 simulated elongation values for the first five cycles are shown in the virtual setup to save time. After five cycles spring is assumed to be failed and the life of the spring is shown to the user. This process is repeated for the three different values of voltage to provide complete hands-on training with the setup.

4.6 Real-time Control and Monitoring of the Actual Setup.

The final challenge associated with the actual setup of accelerated life testing is that it is not remotely accessible. The virtual setup can be used to access the actual setup remotely. For the remote control of the actual setup, the person in the lab will make all the necessary connections of the setup. The virtual setup will then be used to remotely control the PPS and real-time data acquisition of data from the setup. The procedure for remote control and monitoring is discussed in detail in this section. For the ALT setup of the SMA spring, a remote-control procedure is divided into two sections namely remote control of programmable power supply and real-time data acquisition from the actual setup.

4.6.1. Remote control of the programmable power supply.

The programmable power supply used in the actual setup is Rigol DP 831 with three-channel DC programmable power output. The three channels are channel 1 with 8V & 5A, channel 2 is 30V & 2A and channel 3 with -30V & 2A. Channel 1 has the timer function which can be used to generate rectangular wave to provide DC voltage for the specified time. Therefore, channel one is connected across the SMA spring to supply voltage for heating and cooling cycles. Channel 2 of the PPS is connected to LDS to supply constant 24V DC output.

For the remote control of the PPS, SCPI commands (Standard Commands for Programmable Instruments) were used. The PPS is connected to the computer system used in the actual setup using the USB. Python's pyvisa module is used to send commands to the PPS using the PC connected to the PPS. The API developed for sending the data of the simulation algorithm is modified for sending the SCPI commands to the PPS. The commands for the remote control of the PPS are sent as requests from the VR setup to the API. The API then handles these requests by sending appropriate commands to the PPS.

The new online mode is added to the VR setup for the remote control and monitoring of the actual setup. The PPS in the VR setup is used as a trigger for sending particular requests to the system's API. The buttons of the virtual PPS were given the same functionality as that of the actual PPS. The remote control of PPS is shown in figure 51. The display is used to see the current parameters set on the PPS. These parameters include the currently selected channel, ON or OFF status of each channel, voltage and current value set for each channel, and the status of the timer. These parameters are updated at the interval of 5 sec to check for any changes done remotely or even from the lab. The IF on the display stands for the input field which stores the value set from the number pad of the virtual PPS. Along with the interval of 5 sec, the display is updated when the buttons of virtual PPS are pressed to get the instantaneous display of triggers from pressing the buttons from the virtual PPS.

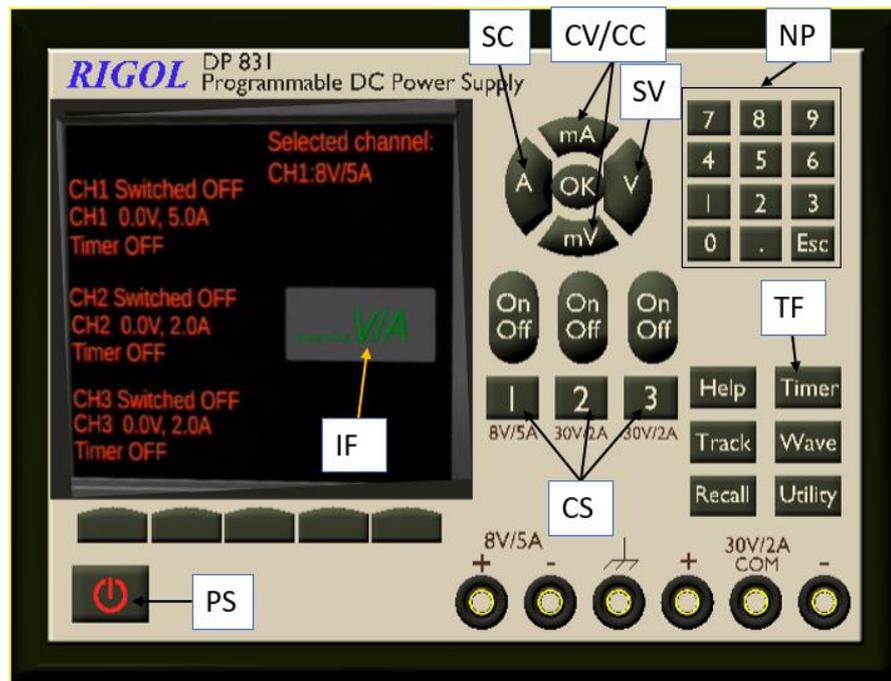


Figure 51: Remote control of PPS

In figure 51, each button has the functionality same as that of the buttons on the actual PPS. The button PS is used to switch ON or OFF the display of the actual PPS. The buttons named CS are used to select the channel marked on each button on the actual setup. On Off buttons above each CS button are used to switch ON or Off the corresponding channel of actual PPS. CV or CC buttons of virtual PPS are used to set constant voltage or constant current output respectively on the actual setup. the NP indicates the number pad which is used to select the particular desired value of the voltage or current up to one decimal place. This value from the number pad is updated and displayed on the IF i.e., the input field on the display if the user presses the SV button, then the corresponding value is set as a voltage else if, SC button is pressed then the value on the IF is set as the current on the actual setup. TF stands for the timer function which is used to switch On or Off the timer to send the wave output.

The functionality discussed in the above paragraph is used for the remote functioning of the actual setup for the accelerated life testing. First, the PS i.e., power supply button is used to turn on the display of the actual setup. In the actual setup, LDS is connected to channel 2 of the PPS and channel 1 is used to apply a voltage across the SMA spring. First, 24V is supplied to channel 2 of the actual PPS to supply voltage to the LDS. To start the LDS channel 2 is selected by pressing the corresponding CS button on the virtual PPS. The number pad (NP) is used to set 24 value in the input field (IF), then the SV button is pressed in the virtual setup to set the voltage of channel 2 to 24V then On Off button above the channel 2 is pressed to start the supply to the LDS in the actual setup. The next step is to start the power supply to the SMA spring. For this purpose, the rectangular waveform is generated using the timer function of the PPS. The TF button on the virtual PPS is pressed to start the timer then the On Off button on the virtual PPS is pressed to start the power supply to the SMA spring. In this way, the SMA spring can be subjected to testing remotely using the VR setup. The next step is to monitor the progress of the actual setup remotely in real-time using VR setup the procedure for which is discussed in detail in the next section.

4.6.2. Real-time data acquisition using a VR-based setup from a remote location.

As discussed in the previous section the LabView software is used for data acquisition. The block diagram and the front panel used for data acquisition are shown in figures 52& 53. The LabView uses a graphical programming interface in the form of block diagrams to implement logic used for data acquisition. The front panel is used for the representation of data in the form of graphs or charts. It also uses various buttons for the control of the data acquisition process.

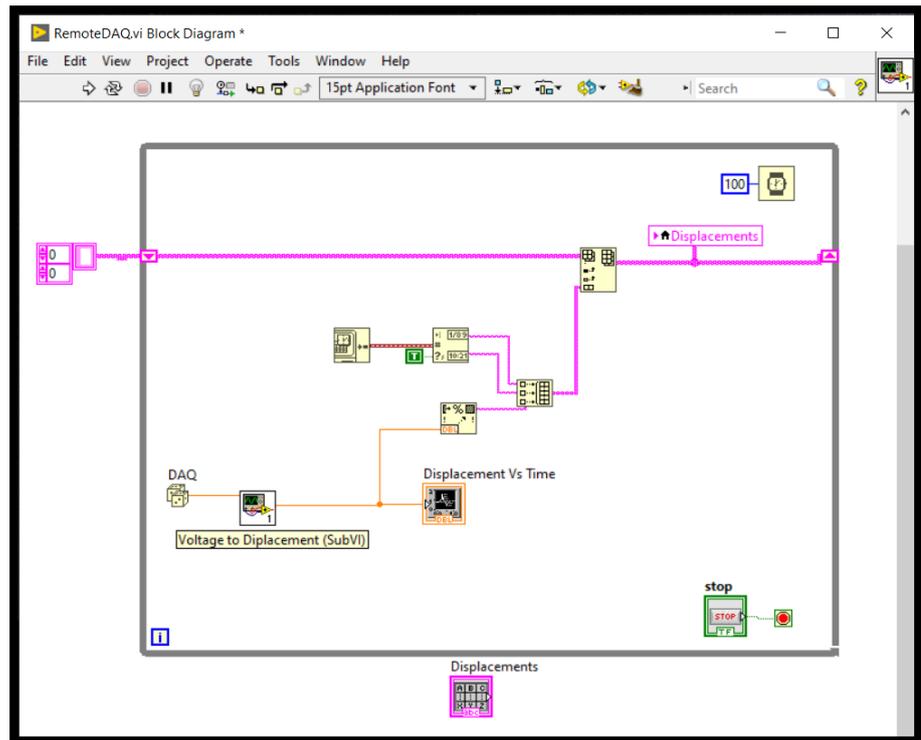


Figure 52: Block diagram for data acquisition in LabView

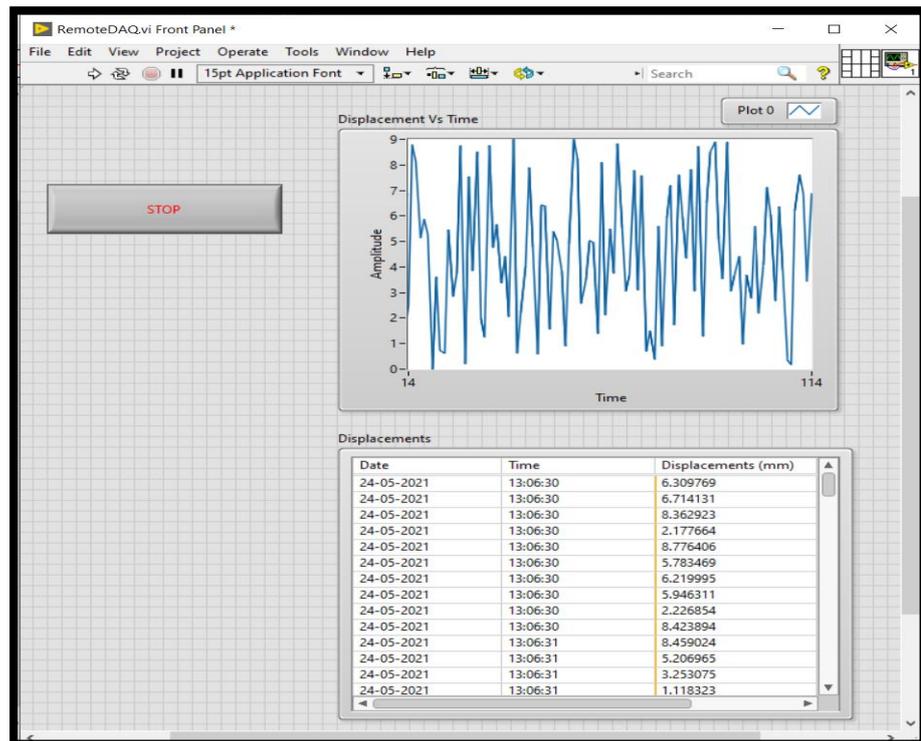


Figure 53: Front panel for data acquisition in LabView

The web publishing tool of LabView is used for remote data acquisition. With a web publishing tool, the execution of the program is controlled

from another computer from a remote location. Without any additional programming, the user can configure any LabView application for remote control through the common web browser. The user needs to simply point the web browser to the webpage associated with the application. The remote user can access fully the user interface that appears in the web browser. The acquisition still occurs on the host computer, but the remote user has complete control of the application. The front panel opened in the browser window opened in the remote computer is shown in figure 54.

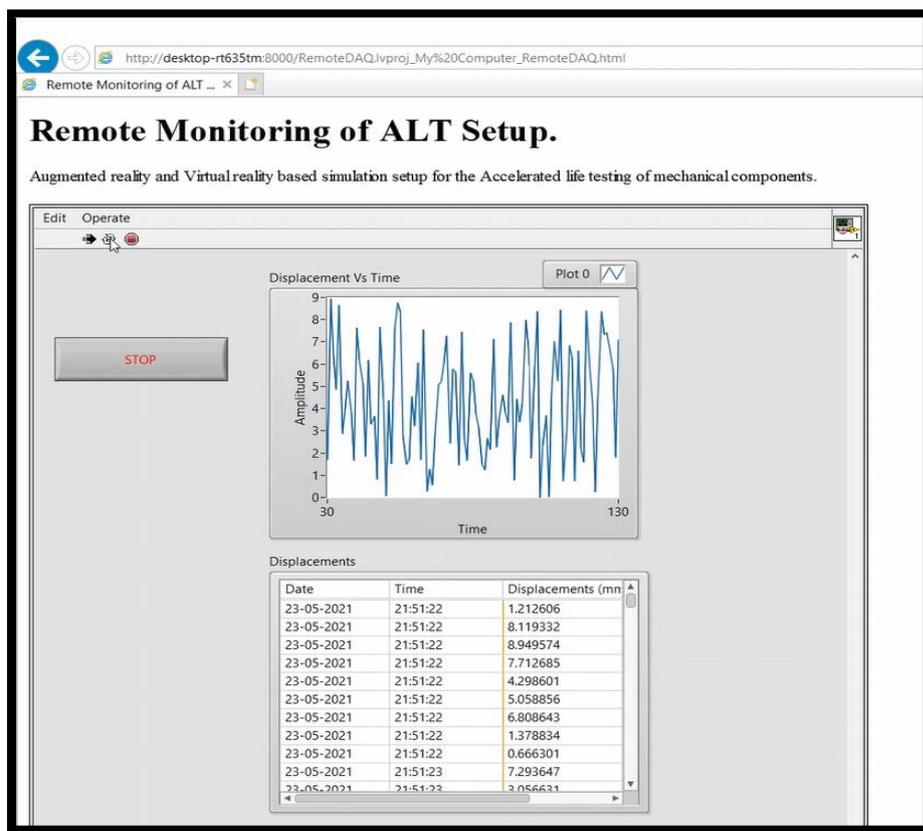


Figure 54: Front panel for remote data acquisition opened in a web browser

The last step for remote data acquisition is to open a browser window for the front panel for the LabView application into VR setup. For this purpose, a screen is a setup in the virtual setup to open the browser window. Unity has a plugin named webview which is used to open browser windows without leaving the application. This plugin is used to

open the application window of LabView in the VR setup. This helps in the remote data acquisition from the actual setup. In this way, the virtual setup can be easily accessed and monitored remotely using the VR-based setup which partially addresses the challenge of limited remote accessibility of actual setups for accelerated life testing. The procedure for complete remote access of the actual setup is beyond the scope of this project work. One of the procedures for remote control is discussed in the future scope section of the thesis in chapter 7.

Chapter 5

Results and Discussions

The procedure discussed in chapter 4.3 is used to generate the simulated failure data. The data generation process of the simulated algorithm is validated in this chapter. Simulated data generated using the simulation algorithm is compared with the actual failure data of SMA springs to check the accuracy of feature extraction. The results obtained are discussed in the present chapter.

5.1 Graphical Comparison of Actual and Simulated Failure Data.

Figure 55-60 shows the plot of simulated failure data generated using the simulation algorithm along with the actual failure data for the graphical comparison.

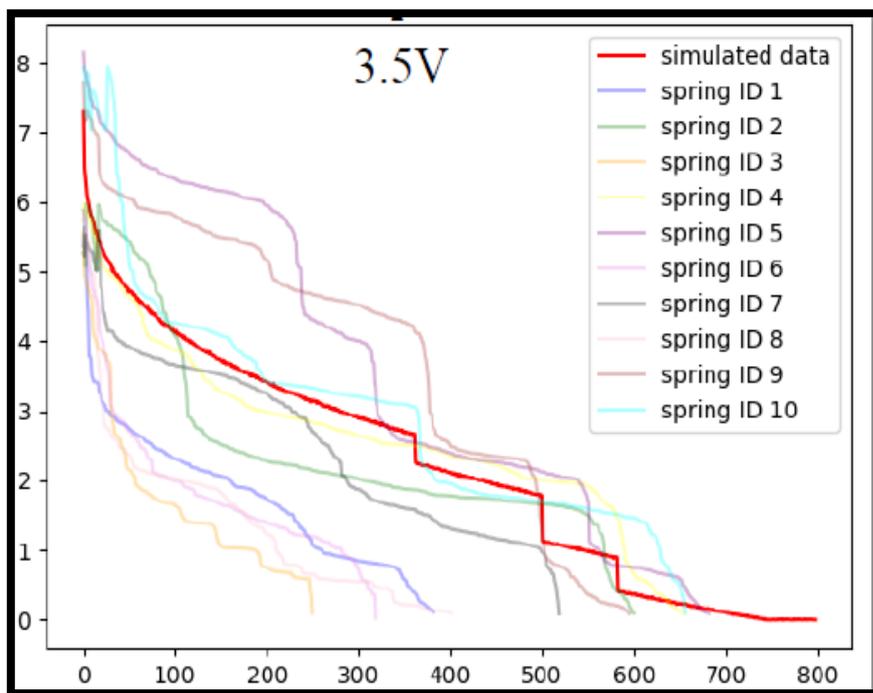


Figure 55: Elongation VS the number of cycles plot for simulated data at 3.5V for simulation no. 1

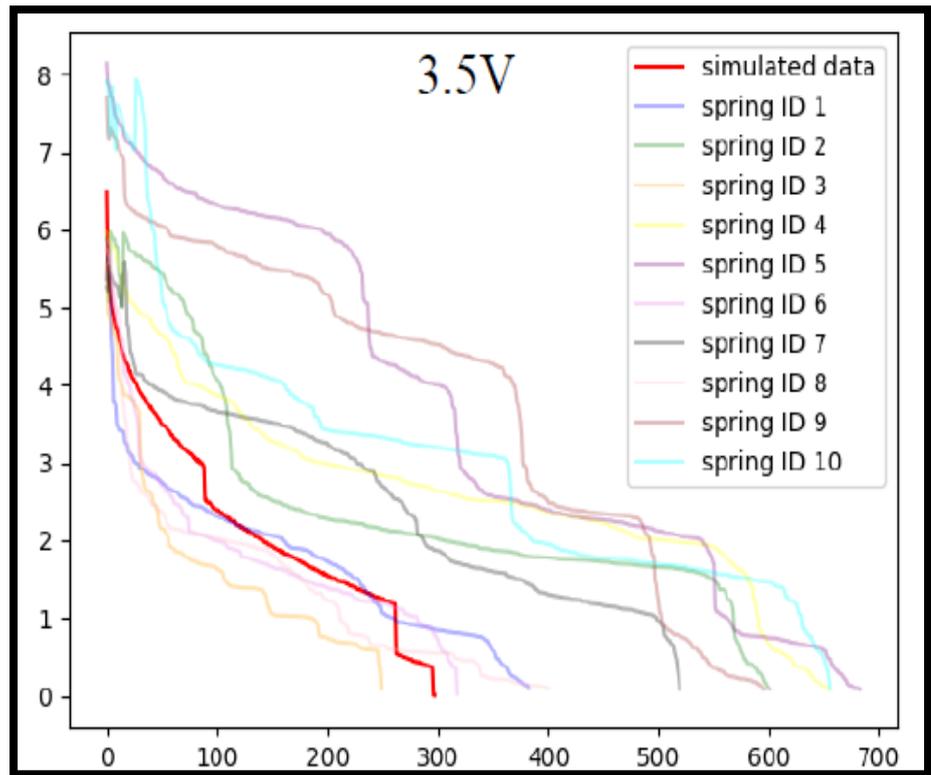


Figure 56: Elongation VS number of cycles plot for simulated data at 3.5V for simulation no. 2

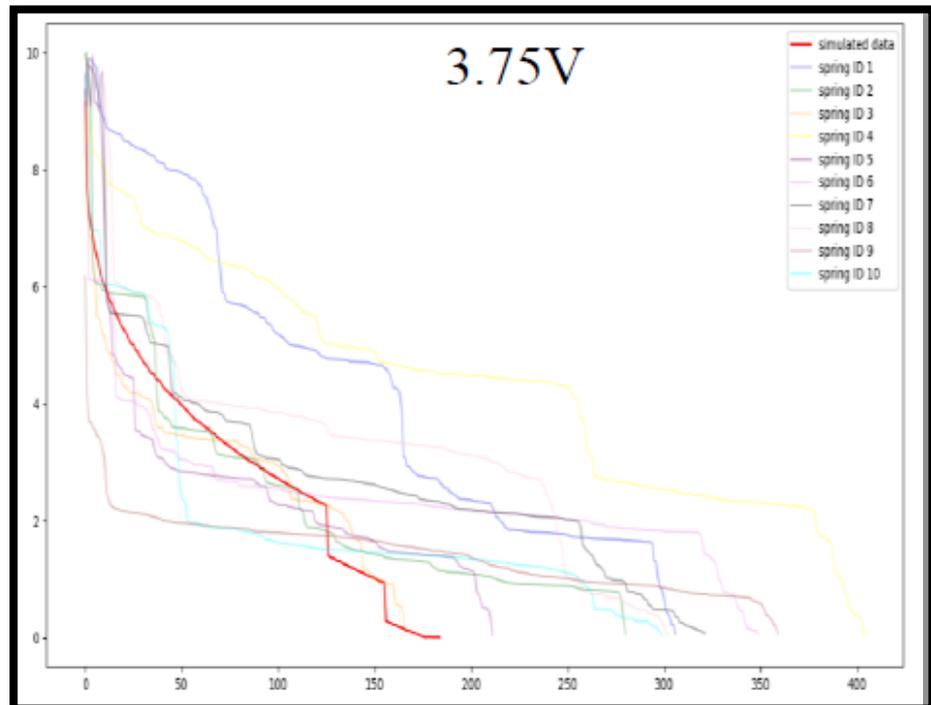


Figure 57: Elongation VS number of cycles plot for simulated data at 3.75V for simulation no. 1

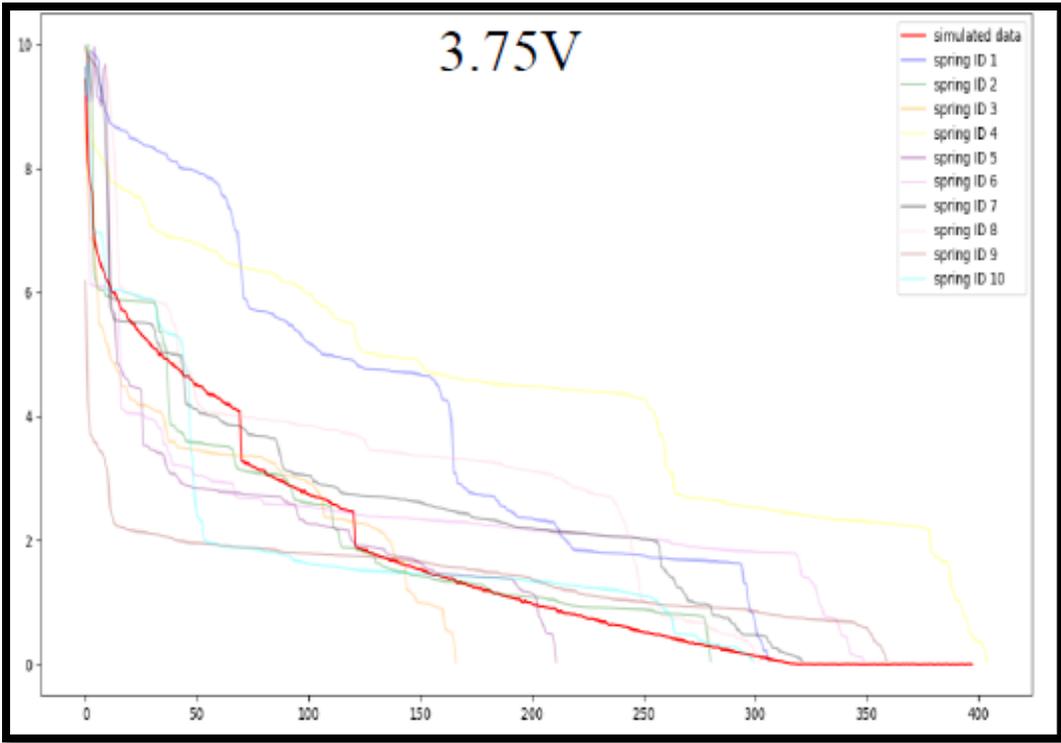


Figure 58: Elongation VS number of cycles plot for simulated data at 3.75V for simulation no. 2

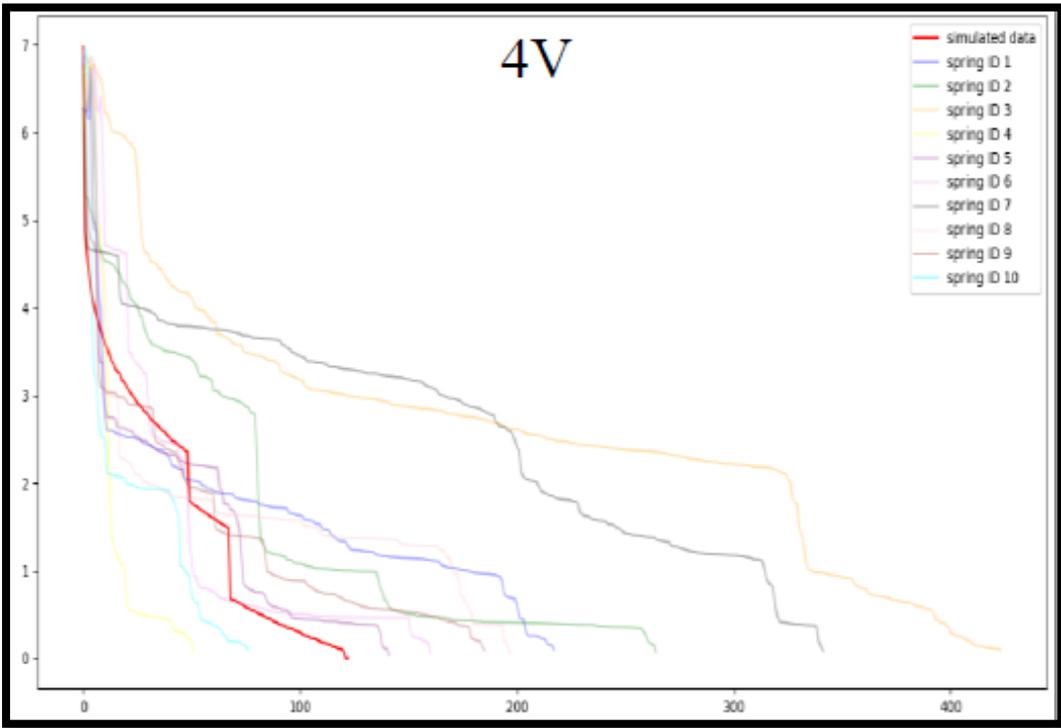


Figure 59: Elongation VS number of cycles plot for simulated data at 4V for simulation no. 1

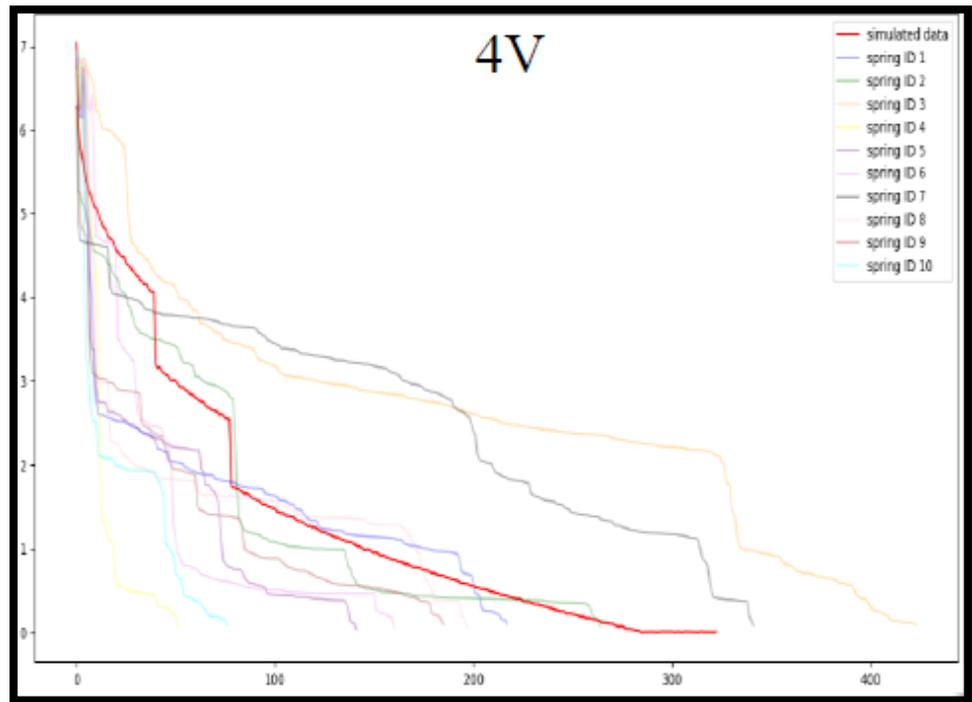


Figure 60: Elongation VS number of cycles plot for simulated data at 4V for simulation no. 3

In the fig. 55 to 60, the dark red line represents simulated failure data generated by the simulation algorithm and the other faint colored lines represent actual failure data of the springs at each level of voltages. From each of the figures, it can be inferred that the red curve of the simulated failure data follows the trend which is similar to the one or more actual failure data of the SMA springs. It means that the failure behavior shown by the simulated component is similar to the failure behavior shown by one or more actual SMA springs. Visually, it can be inferred that the algorithm is correctly extracting features from actual data to generate simulated failure data. However, further investigation is needed to determine the accuracy of the simulation algorithm.

5.2 Calculation of Jaccard and Cosine Similarity Indices.

From the graphs in fig. 55 to 60, it can be seen that the failure behavior of the simulated component is similar to the actual failure behavior

shown by the SMA spring. To mathematically measure this similarity index is calculated. The actual data is considered as one dataset and simulated data is considered as the second dataset and cosine and Jaccard similarity indices are calculated to mathematically determine the similarity between two datasets.

5.2.1 Calculation of cosine similarity index.

The cosine similarity index measures the similarity of two datasets based on the orientation of data points of two datasets from fixed reference. For this purpose, the cosine of the angle between the two data points is measured. If two datasets are similar to each other orientation of their data points will be similar, lower will be the angle between their data points and higher will be the cosine of the angle. Hence, the cosine similarity index is measured for actual data and simulated data using equation 9.

$$\text{Cosine Similarity} = \frac{\sum A_i B_i}{\sqrt{\sum A_i^2} * \sqrt{\sum B_i^2}} \dots (9)$$

Where A & B are simulated and actual datasets respectively and A_i & B_i are corresponding datapoints

From fig. 55, it can be seen that simulated failure data (red curve) is following the failure behavior similar to spring IDs 6, 1 & 8. Hence, the cosine similarity index is calculated with respect to these spring IDs and is shown in table 6.

Table 6: Cosine similarity index

Spring ID	Cosine Similarity Index
6	0.9850
1	0.9837
8	0.9685

From the table, it can be seen that simulated failure data is following failure behavior which is very similar to spring ID 6, 1 & 8. By following the same procedure, the cosine similarity index is calculated for 30 different simulation results obtained at each level of voltage. The maximum cosine similarity index is plotted and shown in figures 61 to 63.

From figures 61 to 63, it can be seen that cosine indices at each level of voltage varied between 0.95 to 0.99 for most of the simulation results. Even though the values of these similarity indices are very high, the cosine similarity index doesn't take into account the magnitude of data points to determine the similarity. Hence, the cosine similarity index alone is not sufficient to determine the similarity between actual and simulated data. Therefore, further investigation is needed.

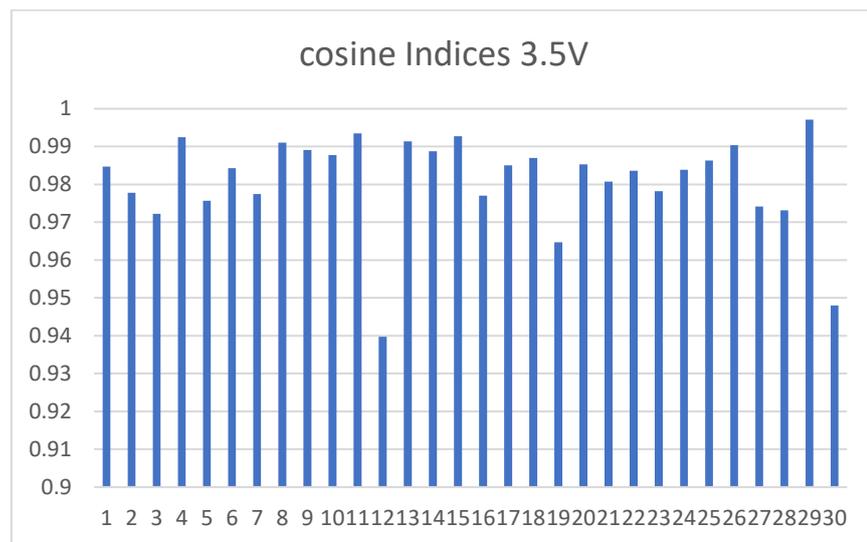


Figure 61: Cosine similarity indices for simulation results at 3.5V.

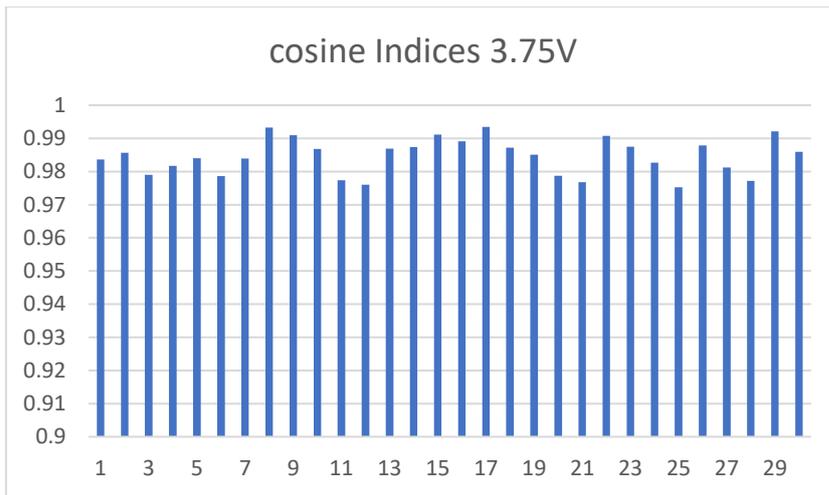


Figure 62: Cosine similarity indices for simulation results at 3.75V

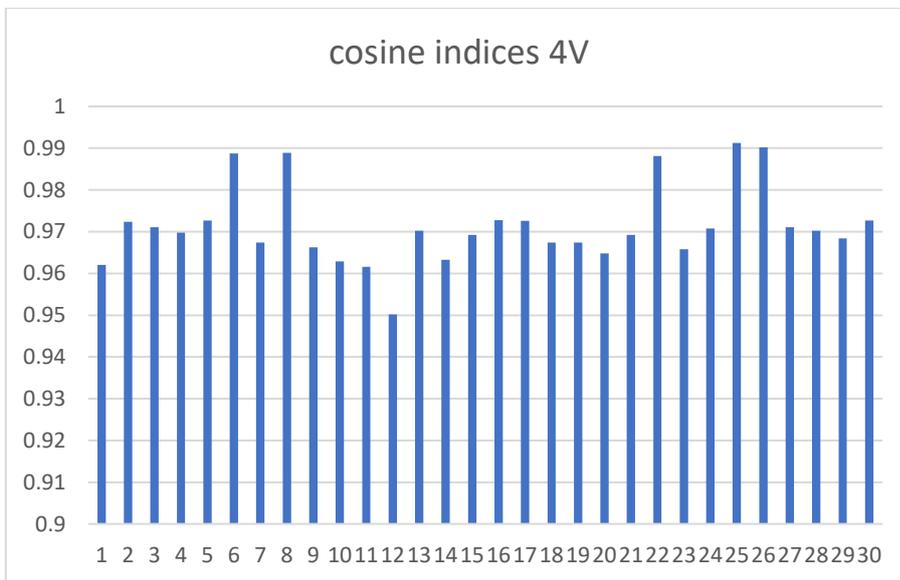


Figure 63: Cosine similarity indices for simulation results at 4V

5.2.2. Jaccard Similarity Index

As cosine similarity only considers the orientation of data points, the Jaccard similarity index which is based on the magnitudes of data points is determined for the spring IDs 6, 1 & 8 with respect to simulated data from figure 55 using equation 10.

$$\text{Jaccard Similarity Index} = \frac{A \cap B}{A \cup B} =$$

$$\frac{\text{Common points to both datasets}}{\text{All points in datasets}}$$

$$J(A, B) = \frac{\sum_i \min(A_i, B_i)}{\sum_i \max(A_i, B_i)} \dots (10)$$

Jaccard similarity indices are shown in table 7.

Table 7: Jaccard similarity index

Spring ID	Jaccard Similarity Index
6	0.7464
1	0.7483
8	0.6469

From the table, it can be inferred that simulated failure data is 74.64%, 74.83%, and 64.69% similar to actual failure data of spring ID 6, 1, and 8 respectively. By following the same procedure Jaccard similarity indices are obtained for 30 different simulation results obtained at each level of voltage and results are shown in figures 62 to 64. From the figure, it can be inferred that Jaccard similarity indices for most of the simulation results varied between 0.65 to 0.9. As the Jaccard similarity index gauges, the similarity based on the magnitudes of data points, these values obtained for the simulation results are sufficient to conclude that data obtained from the simulation algorithm follows the failure behavior similar to actual failure data of SMA springs. Higher values of Jaccard similarity indices i.e., above 0.95 are not good as they indicate overfitting of both datasets. As the failure is stochastic in nature elongation values obtained from actual data as well as for simulation results should be different for each iteration. Hence, Jaccard indices in the range of 0.65 to 0.9 are sufficient to prove that algorithm is correctly extracting the features and generating simulated failure data which

follows the failure behavior similar to actual failure data of the SMA springs.

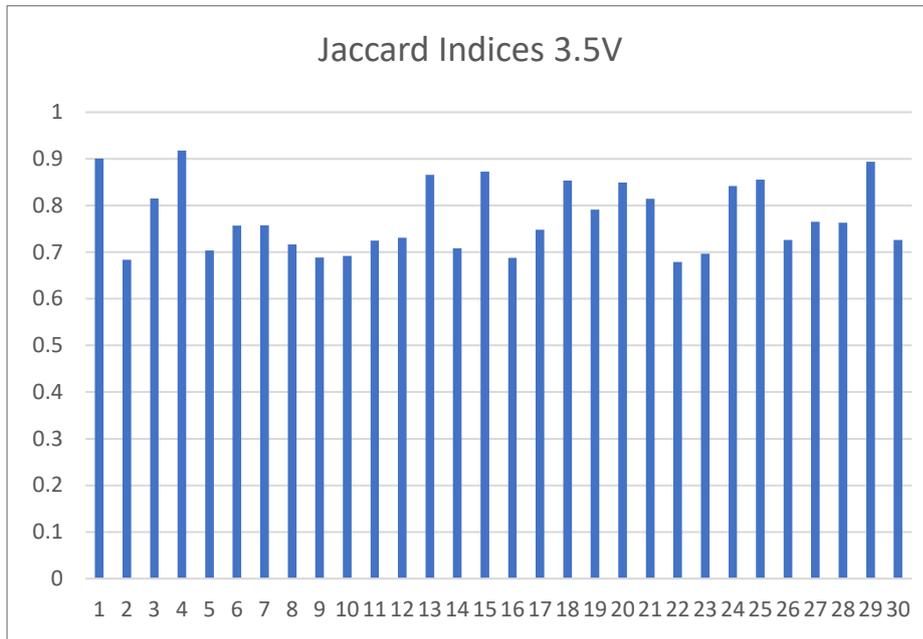


Figure 64: Jaccard similarity indices for simulation results at 3.5V

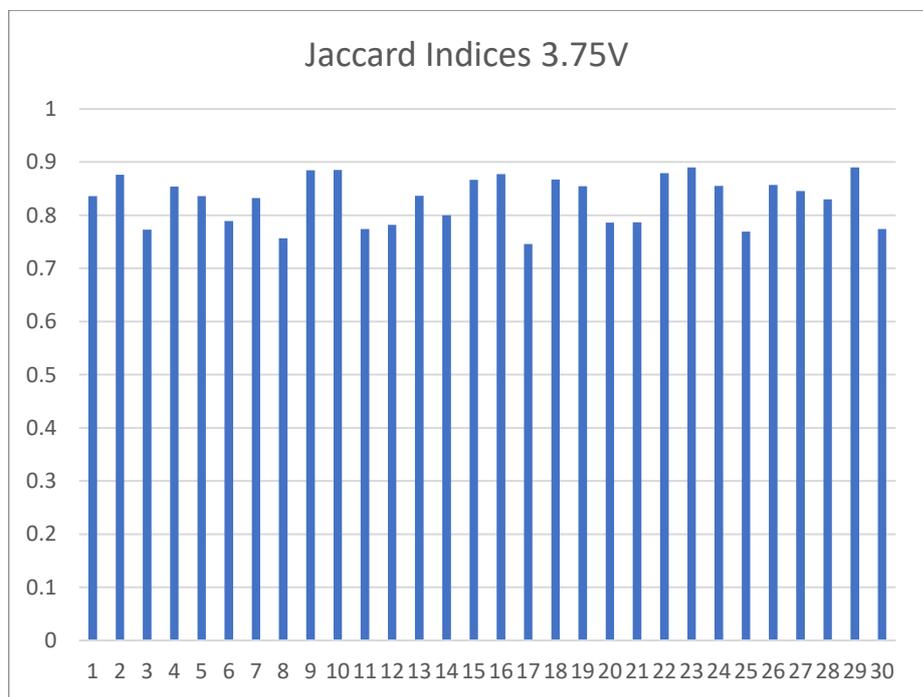


Figure 65: Jaccard similarity indices for simulation results at 3.75V

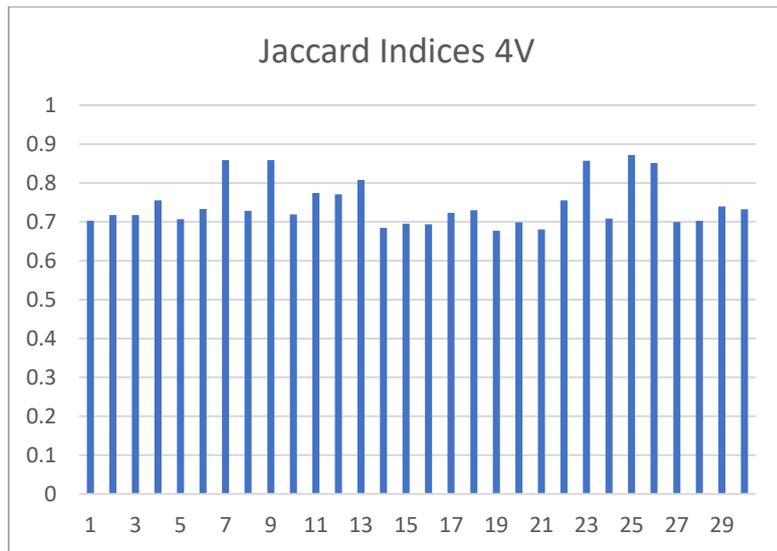


Figure 66: Jaccard similarity indices for simulation results at 4V

5.3 Analysis of Variance (ANOVA) Test

Analysis of variance test i.e., Anova test is used to determine if there is a significant statistical difference between the two populations. It is used to compare the equality of two or more population means. It uses the F-statistics and p-value to accept or reject the null hypothesis. If the F-statistics for the ANOVA test is greater than the critical F-value then populations are said to be statistically different.

For comparison of the actual and simulated failure data following parameters are considered.

- Independent variable: Mode of data generation.
- Populations or treatments:
 1. Failure data generated by performing tests i.e., actual failure data.
 2. Failure data generated using a simulation algorithm i.e., simulated failure data.
- *Null Hypothesis* (H_0): $\mu_1 = \mu_2$ (No significant statistical difference)
- *Alternate Hypothesis* (H_a): $\mu_1 \neq \mu_2$ (Significant statistical difference)

- $F - statistics = \frac{Variance\ between\ the\ groups}{Variance\ within\ the\ group}$

If the variance between the groups is greater than the variance within the groups i.e., populations then the value of F statistics is higher, which means that there is a significant statistical difference between the two populations and vice versa.

- P-value: It is called statistical significance. It determines how significant the results are to reject the null hypothesis. It is also called the probability that the difference between the means occurs by pure chance.

5.3.1. Anova test on TTF data from two populations.

First, the anova test is performed on time to failure (TTF) data for 10 simulation results and actual failure data of 10 springs tested at each level of voltage. First, the anova test is performed on the TTF data obtained from two populations, and the results obtained are shown in table 8.

Table 8: Anova table for TTF data at 3.5V

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	37845	1	37845	1.174138	0.292854	4.413873
Within Groups	580178.8	18	32232.16			
Total	618023.8	19				

In table 8, the first column represents the source of variation which may be between the groups i.e., populations or within the groups. SS represents the squared sum of variations for both populations, df is degrees of freedom for both populations. MS is the mean sum of variations for both populations which is the ratio of the squared sum to the degrees of freedom. F is the value of F-statistics which is the ratio of the mean sum of variation between the populations and the mean sum of variation within the populations. P-value is the probability of getting a given F score for given degrees of freedom for both populations and a 95% confidence interval. F_{crit} is the critical F value found from the F –

distribution for given degrees of freedom for both populations and 95% confidence interval.

From the results of the anova test, it can be inferred that obtained F-value is less than the critical F-value and the p-value is greater than 5% (level of significance). Hence, the null hypothesis is not rejected. Therefore, there is no significant statistical difference between the time to failure data obtained from the simulation algorithm and actual failure data.

Similarly, anova test is performed on the TTF data obtained at 3.75V and 4V, and results are shown in Tables 9 & 10.

Table 9: Anova table for TTF data at 3.75V

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	10951.2	1	10951.2	2.297171	0.146974	4.413873
Within Groups	85810.6	18	4767.256			
Total	96761.8	19				

Table 10: Anova table for TTF data at 4V

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	20352.2	1	20352.2	1.77634	0.199225	4.413873
Within Groups	206232.8	18	11457.38			
Total	226585	19				

For both the cases, F-value is found to be less than the critical F-value and the p-value is greater than 5% (level of significance). Hence, the null hypothesis is not rejected. Therefore, there is no significant statistical difference between the time to failure data obtained from the simulation algorithm and actual failure data.

To check the applicability of anova test, that it is not always accepting the null hypothesis, the anova test is performed on the simulated failure data obtained at two different levels of voltage. Here, the voltage is considered as the independent variable, and TTF data obtained from the

simulation algorithm at 3.5V and 4V are considered as the two populations. The results of anova test are shown in table 11.

Table 11: Anova table for simulated TTF data at 3.5V & 4V

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	368832.8	1	368832.8	14.42964	0.001315	4.413873
Within Groups	460094	18	25560.78			
Total	828926.8	19				

From the table, it can be inferred that F-value from the anova test is greater than the critical F value and the p-value is less than 5% (level of significance). Hence, the null hypothesis is rejected. Therefore, there is a significant statistical difference between the simulated failure data obtained at 3.5V and 4V. These results verify the applicability of anova test as it correctly rejects the null hypothesis for simulated data at two different voltage values as both the data are obtained from different populations. Therefore, from the anova test on TTF data, it can be concluded that there is no significant statistical difference between the TTF data obtained from the simulation algorithm and actual failure data. However, to determine the accuracy of the simulation algorithm further investigation is needed.

5.3.2. Anova test on failure data obtained at TS 2.

To further evaluate the accuracy of the simulation algorithm the failure data, obtained at timestamp 2 for the simulation algorithm and actual failure data is used to perform the anova test. The data at timestamp 2 is chosen because the data at timestamp 1 is obtained from the same normal distribution for the simulated data. Also, the time to failure for each spring is different hence, as the timestamp value increases the variance will go on increasing. For this purpose, anova test is performed on the failure data obtained at timestamp 2. By considering the mode of failure data generation as the independent variable anova test is

performed for actual and simulated failure data at TS 2. The results of anova test are shown in table 12.

Table 12: Anova table for failure data at TS 2 for 3.5V

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	0.035462582	1	0.035463	0.053206	0.820176	4.413873
Within Groups	11.99717453	18	0.66651			
Total	12.03263711	19				

From the anova table, it can be seen that the calculated F value is less than the critical F-value, and the p-value is significantly higher than 5%. Hence, the null hypothesis is not rejected. Therefore, there is no significant statistical difference between the failure data at timestamp 2 for actual and simulated failure data.

Similarly, anova test is performed for the failure data at 3.75V and 4V. The results of the test are shown in tables 13 and 14.

Table 13: Anova table for failure data at TS 2 for 3.75V

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	4.82895E+12	1	4.83E+12	1.000004	0.330564	4.413873
Within Groups	8.69208E+13	18	4.83E+12			
Total	9.17498E+13	19				

Table 14: Anova table for failure data at TS 2 for 4V

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	1.26299E+12	1	1.26E+12	1.000003	0.330564	4.413873
Within Groups	2.27337E+13	18	1.26E+12			
Total	2.39966E+13	19				

From Tables 13 & 14, it can be seen that the calculated F-value from anova test is less than the critical F-value and the p-value is greater than 5%. Hence, the null hypothesis is not rejected. Therefore, there is no significant statistical difference between the failure data at timestamp 2 for actual and simulated failure data.

To check the applicability of anova test, that it is not always accepting the null hypothesis, the anova test is performed on the simulated failure data obtained at two different levels of voltage. Here, the voltage is considered as the independent variable, and TTF data obtained from the simulation algorithm at 3.5V and 4V are considered as the two populations. The results of anova test are shown in table 15.

Table 15: Anova table for failure data at TS 2 for 3.5V& 4V

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	1.680182899	1	1.680183	11.0308	0.003799	4.413873
Within Groups	2.741714331	18	0.152317			
Total	4.42189723	19				

From the table, it can be inferred that F-value from the anova test is greater than the critical F value and the p-value is less than 5% (level of significance). Hence, the null hypothesis is rejected. Therefore, there is a significant statistical difference between the simulated failure data obtained at timestamp 2 for 3.5V and 4V. These results verify the applicability of anova test as it correctly rejects the null hypothesis for simulated data at two different voltage values, as both the data are obtained from different populations. Therefore, from the anova test on failure data, it can be concluded that there is no significant statistical difference between the failure data obtained from the simulation algorithm and actual failure data.

Hence, it can be concluded that the simulation algorithm is correctly identifying the failure features from actual data to generate simulated failure data. The simulated failure data also shows the failure behavior similar to the actual failure behavior shown by the SMA springs, which is also validated by the calculation of similarity indices.

Chapter 6

Conclusion

The Accelerated life testing is an essential part of reliability analysis. There are several challenges associated with the accelerated life testing which are discussed in chapter 1 of the thesis. This thesis tries to provide the solution to the some of challenges faced in the accelerated life testing. The challenges like costly setups for testing, no hands-on training with the setups, and no remote access to the ALT setups are addressed by this thesis.

The first step in the accelerated life testing is building the test setups to perform run to failure tests on the components. Most of the time these setups and components are costly, due to which hands-on training with the setups is not possible. Hence, while searching for plausible solutions to address these challenges the idea of building VR based setup came into existence. The present thesis comprises the complete process for the building of VR-based setups. The VR-based setup incorporates a complete experimental procedure followed in the actual setup for testing to provide hands-on training with the setup. As the virtual environment in the VR setup resembles the actual working environment, it helps the user to completely get familiarize with the actual working environment. This subsequently helps the user to work efficiently while working with the actual setups of the accelerated life testing. This also helps students to gain knowledge of the ALT setups which may not be available in their institutions as they are costly.

For the complete hands-on training with the ALT setups, the failure behavior of the component under test is needed to be simulated in the virtual setup. For this purpose, the simulation algorithm is developed and this thesis includes the complete process for making such a simulator. This thesis tries to incorporate various failure features to make it more realistic. Studies of various failure data were done to find different failure features. The different failure features like failure trend,

random jumps, and noise were considered while developing the algorithm. The validation of simulated failure data generated by the simulation algorithm was done by comparing it with actual failure data. For this purpose, cosine similarity and Jaccard similarity indices were calculated along with the performance of the ANOVA test. After the validation of the data generation process, the performance of the simulation algorithm was found up to the mark. The data generated by the simulation algorithm is then visualized in a virtual setup to incorporate the failure behavior of the component under test.

To address the challenge of the limited accessibility of actual setups, the thesis discusses one of the procedures for the remote control of the actual setups for accelerated life testing. The complete procedure for remote control of the electronic instrument (Programmable Power Supply) is discussed in the thesis. The actual setup for the accelerated life testing of the SMA spring is controlled remotely using the VR-based setup by controlling the PPS remotely. The thesis also provides an idea for the remote monitoring of the actual setup using the VR-based setup. This process involves real-time data acquisition from the actual setup using a VR-based setup from the remote location.

In this process our idea of developing a VR-based setup for hands-on training and education was successful. Implementation of VR concepts in mechanical fields for training and education will open new avenues. The VR-based setups can replace actual setups for accelerated life testing in educational institutions. Instead of working on predefined data which is common in most VR-based setups, the incorporation of a simulation algorithm has allowed our VR setup to run on the real-time data generated by the simulation algorithm. This ensured the stochastic nature of the failure behavior of the component under test. VR setup running on real-time data will open the door for various new possibilities in VR-based systems.

The idea of remote control and monitoring of actual setups using VR-based setups was also successful and will be helpful in pandemic

situations like COVID-19. This idea will also open the door for new possibilities in the mechanical field for cyber-physical systems. AR and VR concepts are growing will play a vital role in mechanical fields to replace physical systems. The idea discussed in the thesis can be the starting step towards the use of AR and VR systems in reliability engineering.

Chapter 7

Future Scope

To address the challenge of remote access of the actual setup for the accelerated life testing using the VR-based setup experimental procedure must be done remotely before using the currently developed VR setup. For this purpose, modifications are needed to be done to the VR setup and the currently developed API. One of the solutions can be to use a robotic arm for performing the experimental procedure. This concept is represented in figure 67.

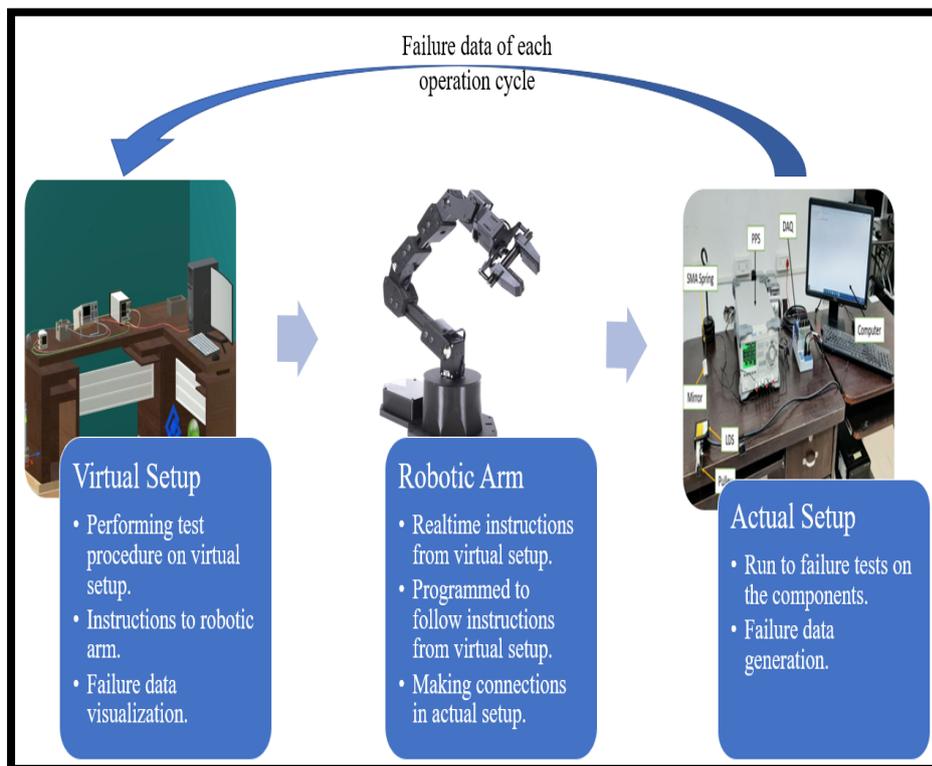


Figure 67: Remote control of actual ALT setup using a robotic arm

As shown in figure 67, the experimental procedure can be performed in the virtual setup by following the experimental procedure discussed in the previous section. While performing the experimental procedure, the instructions will be sent to the robotic arm to perform a similar procedure on the actual setup. This can be done by sending requests from

VR setup to the API of a robotic arm or computer system. The robotic arm will receive these instructions in real-time from virtual setup and it will be programmed to follow these instructions to make necessary connections in the actual setup. After making all the necessary connections the component will be subjected to testing on the actual setup. The failure data generated during the testing will be monitored in real-time using a VR-based setup from a remote location.

In this way, the actual setup for the accelerated life testing can be controlled and monitored remotely using the VR-based setup without the operator being physically present in the lab for performing tests. By following this concept discussed above all the experimental procedures will be done remotely using a VR-based setup and robotic arm.

Appendix A

1. Code of the simulation algorithm.

```
# Importing libraries
import openpyxl
import random
import math
import statistics
from reliability.Fitters import Fit_Weibull_2P
import numpy as np

# function to calculate Jaccard similarity index

def similarity_jaccard(actual, simulated):
    numerators = []
    denominators = []
    a = actual.copy()
    b = simulated.copy()
    if len(a) < len(b):
        for x in range((len(b) - len(a))):
            a.append(0)
    else:
        for z in range((len(a) - len(b))):
            b.append(0)
    for x in range(len(a)):
        numerator = min(a[x], b[x])
        numerators.append(numerator)
        denominator = max(a[x], b[x])
        denominators.append(denominator)
    return sum(numerators) / sum(denominators)
```

```
# function to calculate cosine similarity index
```

```
def cosine_similarity(actual, simulated):  
    numerators = []  
    denominators1 = []  
    denominators2 = []  
    a = actual.copy()  
    b = simulated.copy()  
    if len(a) < len(b):  
        for x in range((len(b) - len(a))):  
            a.append(0)  
    else:  
        for x in range((len(a) - len(b))):  
            b.append(0)  
    for x in range(len(a)):  
        numerator = a[x] * b[x]  
        denominator1 = (a[x] ** 2)  
        denominator2 = (b[x] ** 2)  
        numerators.append(numerator)  
        denominators1.append(denominator1)  
        denominators2.append(denominator2)  
    denom1 = np.sqrt(sum(denominators1))  
    denom2 = np.sqrt(sum(denominators2))  
    return sum(numerators) / (denom1 * denom2)
```

```
# Simulation algorithm function
```

```
def simulation_alg(sheet_num):  
    # path for failure data sheet  
    path = "Spring Data Trend.xlsx"  
    wb = openpyxl.load_workbook(path)
```

```

# sheet having failure data
failure_data_sheet = wb.worksheets[sheet_num]

# Setting jump factor for jump simulation
if sheet_num == 0:
    jump_factor = 0.03
elif sheet_num == 1:
    jump_factor = 0.05
else:
    jump_factor = 0.06

# list containing failure data
all_failure_data = []
failure_data = []
# list containing life of component
life = []
# list containing starting point of data
initial_values = []
# list containing final values before failure
final_values = []

# extracting failure data for analysis from excel sheet

for i in range(1, failure_data_sheet.max_column + 1):
    for j in range(1, failure_data_sheet.max_row + 1):
        if failure_data_sheet.cell(j, i).value is not None:
            failure_data.append(failure_data_sheet.cell(j, i).value)

    life.append(len(failure_data))
    initial_values.append(failure_data[0])
    final_values.append(failure_data[-1])
    all_failure_data.append(list(failure_data))

```

```

failure_data.clear()

# finding TTF of component

ttf_dist = Fit_Weibull_2P(life, print_results=False,
show_probability_plot=False)

eta = ttf_dist.alpha
beta = ttf_dist.beta
rel = random.uniform(0, 1)
ttf = math.ceil(eta * (- math.log(rel)) ** (1 / beta))

# check for increasing or decreasing trend

total_diff_list = []
for i in range(len(all_failure_data)):
    total_diff = (all_failure_data[i][-1] - all_failure_data[i][0])
    total_diff_list.append(total_diff)
increasing_trend = True
if all(x < 0 for x in total_diff_list):
    increasing_trend = False

# simulation of jump

all_slopes_list = []
slopes_list = []
temp_list = []
for i in range(len(all_failure_data)):
    for j in range(1, len(all_failure_data[i])):
        slope = (all_failure_data[i][j] - all_failure_data[i][j - 1])
        slopes_list.append(slope)
        temp_list.append(slope)

```

```

all_slopes_list.append(list(temp_list))
temp_list.clear()

jump_threshold = min(total_diff_list) * jump_factor

# Extracting jump data
# list containing jump magnitudes
jump_data = []
# list containing jump times
all_jump_points = []
jump_points = []
temp_list = []

for i in range(len(all_slopes_list)):
    for j in range(len(all_slopes_list[i])):
        if increasing_trend:
            if all_slopes_list[i][j] > jump_threshold:
                jump_points.append(j + 1)
                temp_list.append(j + 1)
                jump_data.append(all_slopes_list[i][j])
            else:
                if all_slopes_list[i][j] < jump_threshold:
                    jump_points.append(j + 1)
                    temp_list.append(j + 1)
                    jump_data.append(all_slopes_list[i][j])

all_jump_points.append(list(temp_list))
temp_list.clear()

avg_jumps = math.ceil(len(jump_points) / len(all_jump_points))

random_jump_list = []

```

```

if avg_jumps != 0:
    for i in range(len(all_jump_points)):
        if all(x > 0.5 * life[i] for x in all_jump_points[i]):
            random_jump_list.append(False)
        else:
            random_jump_list.append(True)

    if all(x == False for x in random_jump_list):
        random_jump = False
        jump_probability = 0
    else:
        random_jump = True
        jump_probability = len(jump_points) / len(slopes_list)
else:
    jump_probability = 0
    random_jump = True

if avg_jumps > 0:
    if increasing_trend:
        threshold = min(final_values) - avg_jumps *
jump_threshold
    else:
        threshold = max(final_values) - avg_jumps *
jump_threshold
    else:
        threshold = random.uniform(min(final_values),
max(final_values))

mean = statistics.mean(initial_values)
std = statistics.pstdev(initial_values)
random_start_mean = 0

```

```

while random_start_mean < min(initial_values) or
random_start_mean > max(initial_values):
    random_start_mean = np.random.normal(mean, std)

# simulation of trend

all_trend_points = []
trend_simulated_data = [random_start_mean]
exp_list = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.2, 1.5, 2,
2.5, 3, 3.5, 4, 4.5, 5]
max_jacard_list = []
jacard_indices = []
max_index = []
for n in exp_list:
    k = (threshold - random_start_mean) / ttf ** n

    for i in range(1, ttf):
        CIM = random_start_mean + k * i ** n
        trend_simulated_data.append(CIM)
        all_trend_points.append(list(trend_simulated_data))
        for i in range(len(all_failure_data)):

            jacard_indices.append(similarity_jacard(all_failure_data[i],
            trend_simulated_data))
            max_jacard_list.append(max(jacard_indices))

max_index.append(jacard_indices.index(max(jacard_indices)))
jacard_indices.clear()
trend_simulated_data.clear()
n = exp_list[max_jacard_list.index(max(max_jacard_list))]

# generation of trend points

```

```

simulated_data = [random_start_mean]

k = (threshold - random_start_mean) / ttf ** n

jump_magnitude = 0
jump_times = []
if not random_jump:
    max_jumps = max([len(jumps) for jumps in all_jump_points])
    min_jumps = min([len(jumps) for jumps in all_jump_points])
    no_jumps = random.randint(min_jumps, max_jumps)
    for i in range(no_jumps):
        jump_time = random.randint(math.ceil(0.5 * ttf), ttf - 1)
        jump_times.append(jump_time)
    jump_times.sort()
j = 0

for i in range(1, ttf):
    CIM = random_start_mean + k * i ** n
    NA = np.random.normal(CIM, CIM * 0.002)
    rand = random.uniform(0, 1)
    if random_jump:
        if rand < jump_probability:
            jump = random.uniform(max(min(jump_data), -1),
max(jump_data))
            jump_magnitude += jump
            trend_point = NA + jump_magnitude
        else:
            trend_point = NA + jump_magnitude
    else:
        if i in jump_times:
            jump = random.uniform(min(jump_data),
max(jump_data) - 0.1)

```

```

        jump_magnitude += jump
        trend_point = NA + jump_magnitude
        j += 1
    else:
        trend_point = NA + jump_magnitude

simulated_data.append(trend_point)

# removing negative values

for i in range(len(simulated_data)):
    if simulated_data[i] <= 0:
        simulated_data[i] = random.uniform(0, 0.009)
# output is simulated failure dat
return simulated_data

```

2. Code for remote control of Programmable Power Supply.

```

# Importing pyvisa library

import pyvisa

rm = pyvisa.ResourceManager()
pps = rm.open_resource(rm.list_resources()[0])

# Function for switching the channel for PPS

def on_off_channel(channel):
    output = stat = pps.query(":OUTP? " + str(channel))
    if stat.split()[0] == 'ON':
        pps.write(':OUTP ' + str(channel) + ',OFF')

```

```

        output = str(channel) + " switched " + pps.query(':OUTP? ' +
str(channel))
        elif stat.split()[0] == 'OFF':
            pps.write(':OUTP ' + str(channel) + ',ON')
            output = str(channel) + " switched " + pps.query(':OUTP? ' +
str(channel))
        return output

```

Function for checking the on-off status of each channel of PPS

```

def on_off_info():
    ch1 = pps.query('OUTP? CH1')
    ch2 = pps.query('OUTP? CH2')
    ch3 = pps.query('OUTP? CH3')
    return "CH1 switched " + ch1 + "; CH2 switched " + ch2 + ";
CH3 switched " + ch3

```

function for selecting a channel for PPS

```

def channel_select(channel_num):
    pps.write(':INST CH' + str(channel_num))
    output = "Selected channel: " + pps.query(':INST? CH' +
str(channel_num))
    return output

```

function for checking currently selected channel

```

def channel_select_info():
    info = pps.query(':INST? ')
    return "Selected Channel: " + info

```

```
# Function for setting constant voltage output
```

```
def set_const_voltage(volt, curr, ocp):  
    channel = pps.query(':INST? ').split(':')[0]  
    pps.write(':APPL ' + str(channel) + ',' + str(volt) + ',' + str(curr))  
    pps.write('CURR:PROT ' + str(ocp))  
    pps.write('CURR:PROT:STAT ON')  
    output = pps.query(':APPL? ' + str(channel)).split(',')  
    return 'CH1: ' + output[1][:-2] + 'V, ' + output[2][:-4] + 'A'
```

```
# Function for setting constant current output
```

```
def set_const_current(volt, curr, ovp):  
    channel = pps.query(':INST? ').split(':')[0]  
    pps.write(':APPL ' + str(channel) + ',' + str(volt) + ',' + str(curr))  
    pps.write('VOLT:PROT ' + str(ovp))  
    pps.write('VOLT:PROT:STAT ON')  
    output = pps.query(':APPL? ' + str(channel)).split(',')  
    return output[0] + ', ' + output[1] + 'V, ' + output[2] + 'A'
```

```
# Function for switching timer
```

```
def start_timer(file_no):  
    timer_stat = pps.query(':TIME?')  
    if timer_stat.split()[0] == 'OFF':  
        pps.write(':MEM:LOAD RTF,' + str(file_no))  
        pps.write(':TIME ON')  
        timer_stat = "Timer " + pps.query(':TIME?')  
    else:  
        pps.write(':TIME OFF')  
        timer_stat = "Timer " + pps.query(':TIME?')
```

```

    return timer_stat

# Function for checking timer status

def timer_info():
    return "Timer " + pps.query('TIME?')

# Function for setting any value of voltage

def set_voltage(volt):
    channel = pps.query(':INST? ').split(':')[0]
    pps.write(':APPL ' + str(channel) + ',' + str(volt))
    output = pps.query(':APPL? ' + str(channel)).split(',')
    return channel + ' ' + output[1][:-2] + 'V, ' + output[2][:-4] + 'A'

# Function for setting any value of current

def set_current(curr):
    channel = pps.query(':INST? ').split(':')[0]
    voltage = pps.query(':APPL? ' + str(channel)).split(',')[1][:-1]
    pps.write(':APPL ' + str(channel) + ',' + str(voltage) + ',' +
str(curr))
    output = pps.query(':APPL? ' + str(channel)).split(',')
    return channel + ' ' + output[1][:-2] + 'V, ' + output[2][:-4] + 'A'

# Function for switching the display of PPS

def switch_display():
    stat = pps.query(':DISP?').split()[0]
    if stat == 'ON':
        pps.write(':DISP OFF')
        output = 'PPS switched off'

```

```

else:
    pps.write(':DISP ON')
    output = 'PPS switched ON'
return output

# Function for checking current value of voltage and current

def channel_info():
    ch1 = pps.query(':APPL? CH1')
    ch2 = pps.query(':APPL? CH2')
    ch3 = pps.query(':APPL? CH3')
    return ch1 + ";" + ch2 + ";" + ch3

```

3. Code for API (Application Programming Interface)

```

# Importing libraries

from flask import Flask, request, jsonify

# Importing code for remote control of PPS

import PPS_remote_actuation

# Importing code for simulation algorithm

import Vr_model_interface_program

app = Flask(__name__)

@app.route('/')

```

```

def index():
    elongation_data = {
        '3.5V': Vr_model_interface_program.simulation_alg(0),
        '3.75V': Vr_model_interface_program.simulation_alg(1),
        '4V': Vr_model_interface_program.simulation_alg(2)
    }

    return jsonify(elongation_data)

```

```

@app.route('/pps_on')
def pps_switch_on():
    """IP_address:port_name/pps_on?channel=channel_name"""
    args = request.args
    channel = args['channel']
    return PPS_remote_actuation.on_off_channel(channel)

```

```

@app.route('/channel_select')
def channel_select():
    """IP_address:port_name/channel_select?channel=channel_num"""
    args = request.args
    channel = args['channel']
    return PPS_remote_actuation.channel_select(channel)

```

```

@app.route('/const_voltage')
def voltage_setting():
    """IP_address:port_name/const_voltage?v=voltage&i=current&oc
p=over-current protection"""

```

```

args = request.args
volt = args['v']
curr = args['i']
ocp = args['ocp']
return PPS_remote_actuation.set_const_voltage(volt, curr, ocp)

@app.route('/const_current')
def current_setting():

    """IP_address:port_name/const_current?v=voltage&i=current&ovp
    =over-voltage protection"""

    args = request.args
    volt = args['v']
    curr = args['i']
    ovp = args['ovp']
    return PPS_remote_actuation.set_const_current(volt, curr, ovp)

@app.route('/timer')
def timer():

    """IP_address:port_name/timer?f=file_num"""

    args = request.args
    file = args['f']
    return PPS_remote_actuation.start_timer(file)

@app.route('/voltage')
def set_volt():

    """IP_address:port_name/voltage?v=voltage"""

    args = request.args
    # channel = args['channel']

```

```
voltage = args['v']
return PPS_remote_actuation.set_voltage(voltage)
```

```
@app.route('/current')
def set_curr():
    """IP_address:port_name/current?i=current"""
    args = request.args
    # channel = args['channel']
    current = args['i']
    return PPS_remote_actuation.set_current(current)
```

```
@app.route('/display')
def display():
    """IP_address:port_name/display"""
    return PPS_remote_actuation.switch_display()
```

```
@app.route('/info')
def get_info():
    """IP_address:port_name/info"""
    return PPS_remote_actuation.channel_info()
```

```
@app.route('/info/on_off')
def get_on_off_info():
    """IP_address:port_name/info/on_off"""
    return PPS_remote_actuation.on_off_info()
```

```
@app.route('/info/channel')
```

```
def get_channel_info():
    """IP_address:port_name/info/channel"""
    return PPS_remote_actuation.channel_select_info()
```

```
@app.route('/info/timer')
def get_timer_info():
    """IP_address:port_name/info/timer"""
    return PPS_remote_actuation.timer_info()
```

```
if __name__ == "__main__":
    app.run(host='0.0.0.0', port=5000, debug=False)
```

REFERENCES

- [1] Jun-Yang Li, Jia-Xu Wang, Guang-Wu Zhou, Wei Pu and Zhi-Hua Wang, “Accelerated life testing of harmonic driver in space lubrication”, *journal of engineering tribology*, 2015, Vol. 229(12) 1491–1502.

- [2] Sebastian Marian Zaharia, Ionel Martinescu, Cristin Olimpiu Morariu, “Life time prediction using accelerated test data of the specimens from mechanical element”, *Maintenance and Reliability* 2012; 14 (2): 99–106.

- [3] Luis Péreza, Eduardo Dieza, Rubén Usamentiagab, Daniel F. Garciab, “Industrial robot control and operator training using virtual reality interfaces”, *Computers in Industry* 109 (2019) 114–120.

- [4] Michele Giannuzi, Gabriele Papadia, Claudio Pascarelli, “IC.IDO as a tool for displaying machining process. The logic interface between Computer aided manufacturing and virtual reality”, *Procedia CIRP* 88 (2020) 145-150.

- [5] Ziyue Guo, Dong Zhou, Qidi Zhou, Xin Zhang, Jie Geng, Shengkui Zeng, Chuan Lv, Aimin Hao, “Applications of virtual reality in maintenance during the industrial product lifecycle: A systematic review”.

- [6] Osama Halabi, “Immersive virtual reality to enforce teaching in engineering education”, *Multimedia Tools and Applications* (2020) 79:2987–3004.

- [7] Ze-Hao Lai, Wenjin Tao, Ming C. Leu, Zhaozheng Yin, “Smart augmented reality instructional system for mechanical assembly toward worker-centered intelligent manufacturing”, *Journal of Manufacturing Systems* 55 (2020) 69–81.
- [8] M. M. L. Chang, S. K. Ong, A. Y. C. Nee, “AR-guided Product Disassembly for Maintenance and Remanufacturing”, *Procedia CIRP* 61 (2017) 299 – 304.
- [9] Dimitris Mourtzis, Vasilis Siatras, Vasilios Zogopoulos, “Augmented reality visualization of production scheduling and monitoring”, *Procedia CIRP* 88 (2020) 151-156.
- [10] Hirosuke HORII, Yohei MIYAJIMA, “Augmented reality-based support system for teaching hand drawn mechanical drawing”, *Procedia - Social and Behavioural Sciences* 103 (2013) 174 – 180.
- [11] BOEING. "Boeing tests augmented reality in the factory", <http://www.boeing.com>, Jan. 2018.
- [12] BELGNAOUI Y. "Renault Trucks teste la réalité mixte", *Arts et Métiers Magazine*, nov. 2018.
- [13] Asa Fast-Berglund, Liang Gong, Dan Li, “Testing and validating extended reality (xR) technologies in manufacturing”, *Procedia Manufacturing* 25 (2018) 31-38.
- [14] J. Lee, B. Bagheri, and H.-A. Kao, “A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing

systems”, *Manufacturing Letters*, vol. 3, pp. 18-23, 2015/01/01/ 2015.

- [15] SangSu Choi, Kiwook Jung and Sang Do Noh, “Virtual reality applications in manufacturing industries: Past research, present findings, and future directions”, 2015, *Concurrent Engineering: Research and Applications*, 1–24.
- [16] T.S. Mujber, T. Szecsi, M.S.J. Hashmi, “Virtual reality applications in manufacturing process simulation”, *Journal of Materials Processing Technology* 155–156 (2004) 1834–1838.
- [17] K. Iwata, M. Onosato, K. Teramoto, S. Osaki, “Virtual manufacturing systems as advanced information infrastructure for integrated manufacturing resources and activities”, 46 (1997) 335–338.
- [18] AZUMA RT. “A survey of augmented reality”, *Presence*, Volume 6, pp. 355-385, 1997.
- [19] Dominique Scaravetti, Dominique Doroszewski, “Augmented reality experiment in higher education, for complex system appropriation in mechanical design”, *Procedia CIRP* 84 (2019) 197-202.
- [20] Pradeep Kundu, Tameshwer Nath, I.A. Palani, and Bhupesh K. Lad, “Integrating GLL Weibull Distribution Within a Bayesian Framework for Life Prediction of Shape Memory Alloy Spring Undergoing Thermo Mechanical Fatigue”, *Journal of Materials Engineering and Performance* Volume 27(7) July 2018 3655.

- [21] Mahdi Baniasadi, Alireza Foyouzat, Mostafa Baghani, “Multiple Shape Memory Effect for Smart Helical Springs with Variable Stiffness over Time and Temperature”, *International Journal of Mechanical Sciences* (2020).