5-2022

# Evaluation of Manually Completed Manufacturing Assembly Processes Through a Wearable Force and Motion Sensing System Integrated Into a Glove

Scott Kerner
skerner@clemson.edu

# Evaluation of Manually Completed Manufacturing Assembly Processes Through a Wearable Force and Motion Sensing System Integrated Into a Glove

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Masters of Science
Automotive Engineering

by
Scott Robert Kerner
August 2022

Accepted by:
Dr. Laine Mears, Committee Chair
Dr. Johnell Brooks
Dr. Divya Srinivasan

# Abstract

The objective of this research is to model the relationship between force, sound, and motion signals in manual assembly environments through a wearable sensor glove and the resultant quality of vehicle connections made on the assembly line.

Many tasks in production assembly are still completed manually due to the intuition needed by the associate, complex automation steps, or time constraints. This is largely observed in automotive assembly environments. With the amount of variability in manually completed processes, the possibility for error increases. These processes include hose and electrical connections which can loosen over time after passing initial quality testing, resulting in costly, time-consuming rework and a diminished brand image.

It is the intent of this work to utilize multidimensional operator force signatures and movements exhibited to understand the primary forces acting in the direction of the connector locking and additional measured forces acting in other directions. The sensor signals feed into the classification algorithm for rapid postprocessing to enable real-time feedback indicating a completed connection or a connection that needs further investigation. These classifications can later act as a steppingstone for automating manually completed manufacturing processes by implementing the findings into autonomous systems to yield an automatic verification of the process.

This research captured data physically exerted by the operator as a means of accountable process quality evaluation where there are limited marketable products and research. The work also introduced a sensor glove system capable of capturing operator applied shear force in a robust and durable way fit for a manufacturing environment. Marketed products and research shear force sensing are extremely limited in breadth, and force sensing gloves are unsuitable for an assembly environment due to cost, measurement capabilities, durability, and/or operator encroachment. The sensing system developed in this research is coupled with a classification algorithm capable of discerning incomplete or rework connections from successful ones demonstrated on an OEM assembly line.

The developed sensor glove capable of capturing shear and normal force, acceleration, and gyroscopic information was successfully tested on an OEM assembly line for 250+ vehicles of work. This includes the completion of hard plastic connections, tool usage, and tasks completed outside of the takt. Five classification models using the gathered data yielded accuracies of 91% or above using a 60/40 train/test split. The best performing model, Naïve Bayes, achieved a balanced accuracy of 97.6%.

# Acknowledgments

I would first like to thank my advisor, Dr. Laine Mears, for his enthusiasm, guidance, and virtue which consistently aided in accelerating my learning experience. Dr. Mears has always been excited about opportunities for himself, his research team, and for individuals which was evident from the start when I spoke with him before becoming one of his students. His excitement towards learning is commendable.

I would like to offer a special thanks to Dr. Matthew Krugh for all of his support in my research. He offers a very intelligent perspective and is well versed with many of the tools I have utilized in my time at Clemson.

I also greatly appreciate the support from my fellow MS students with my research, for being there to discuss ideas, and for the moments outside of schoolwork to destress.

I would like to extend a deep thank you to my family. My parents and siblings have been extremely supportive throughout my entire education and I could not have accomplished what I have without their help. They are constantly looking out for me, staying in touch, and are excited for my future. I look forward to repaying this non-monetary debt to each and everyone of them.

I would also like to thank my committee members for their insight into my research and for adapting with the program. Being the first AuE MS with a thesis has thrown some uncertainty into the experience that did not seem to phase my committee. They have continued to offer their support and looked to understand my investigation to put their best foot forward.

One last thank you to my old friends and the new ones I have made on this journey.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Research Objective and plan

The goal of this study is to provide accountable measures from assembly line operatives to a manufacturer for process quality improvement in manually completed tasks.

The objective of this research is to model the relationship between force, sound, motion and other sensed signals in manual assembly environments and the resultant quality of connections made. The approach entails augmenting the associate with various sensing forms to describe the response of the system as a measurement of quality to minimize defects without overextending process times.

To test this hypothesis, this work will investigate various sensing systems and methods to repeatably gather operator task data, then analyze using a classification algorithm for objective determination of quality. The measurement of process quality will be relayed to the operator. This will assist the manufacturing line in completing work accurately the first time, mitigating rework in the plant, at the distributor, or with the consumer. The coupled system will also provide a level of recordable accountability to the manufacturer.

To test this theory, data were collected in conjunction with BMW Group Plant Spartanburg. This included a series of successfully completed and purposefully improper connections on which data were collected for classification evaluation. A commercial solution that is capable of capturing operator behaviors on the assembly line and evaluating said processes does not currently exist. Investigated sensing solutions that may be capable of capturing operator activation successfully are not robust and durable enough for a harsh manufacturing environment. Therefore, this research can lead to a practical solution that can have economic impact in manufacturing environments.

## 1.2 Motivation

The automotive industry is a large contributor to global innovation and economic prosperity. This is largely due to the competitiveness between Original equipment manufacturers (OEMs) to produce the highest quality product to their targeted audience at a given price point. Therefore, OEMs and suppliers are constantly investigating new methods to streamline processes, maximize quality, and improve worker safety. For this reason, many innovations look towards automation. This could allow a manufacturer to instrument a lights-out manufacturing approach to achieve safer production with little human involvement. Currently, operators are utilized in many parts of manufacturing in the automotive industry, but their predominate commonplace is in final assembly. This can be problematic to final product quality as it is estimated that operators account for 40% of failures resulting in a stoppage of the system from an improper action [1].

Vehicle connections such as electrical, gas, ventilation, or coolant connections are some of the most frequent errors observed in manufacturing at automotive OEMs and suppliers. Due to the process flexibility needed, many connection operations in final assembly are performed manually, introducing the possibility of error while completing and checking the connections. Manual assembly offers a flexible format that can benefit from the associate's reasoning and logic, increasing the potential of what the assembly line worker can determine [2]. Thus, operators are needed to complete these actions where automation is not yet fiscally or technologically feasible. The value of humans in the workplace has been reinforced in recent years when excessive automation has been observed. One of the most notable instances of over-automation is Tesla when they tried and failed at completely automating the Tesla Model 3 assembly line. This was a result of ”too much technology all at once” that came from a ”complex network of conveyor belts [that] was not working, so [they] got rid of whole thing” [3].

Failure to properly perform connections or validation checks can result in missed or incomplete connections, causing rework if the error is found during product testing or potential warranty issues if the product ultimately fails in the field. This may result in extensive rework costs and a diminished brand image.

It is the intent of this work to utilize multidimensional operator force signatures and movements to understand the primary forces acting in the direction of the connector locking and additional measured forces acting in other directions. This feeds into the classification algorithm for rapid post-processing to enable real-time feedback indicating a completed connection or a connection that needs further investigation. These classifications can later act as a stepping stone for automating manually completed manufacturing processes by implementing the findings into autonomous systems to yield an automatic verification of the process.

# Chapter 2

# Related Work

## 2.1 Operator/Machine Collaboration

Human error in manufacturing systems is the source for many defects and rework in a final product. This is caused by qualities such as inconsistencies in operator approach, improper tool use, or degradation of the quality of work as the workers get farther into their shift. An experiment was conducted using historical data on non-compliance (rejected items and rework offline) directly attributed to man, with findings showing human error probability of 2-16% dependent on the task and elapsed time into the worker's shift [4]. As a result of human error, cost, and efficiency, recent efforts in manufacturing are moving towards the automation of repetitive tasks, providing an opportunity to create a societal change in manufacturing, "...improving the quality of life, raising productivity and earnings; making work less dirty, dangerous, physically punishing, and dull; and increasing the value of thinking, creativity, and expertise." [5]. Societal changes shape the automation of repetitive, manually completed tasks performed by workers in automotive production. However, many of these tasks are too complex to automate with existing robot technologies. Current robot designs do not perform well in varying, unstructured environments with people and existing technology [5]. This is due to the knowledge gaps in robotics' adaptability and control when deviating from a programmed function. Robotics lack the flexibility to perform various manual processes because manipulation, mobility, and computation of current robotic technologies are customarily used only for fixed processes [2]. Fixed processes meaning that robots are typically carefully designed and specified by fixed programming methods that cannot handle the process variability of manual automotive assembly.

Manual assembly offers a flexible format that allows for reasoning and logic, increasing the potential of what the assembly line worker can determine [2]. This flexibility in workers' roles can help ensure that variations from an ideal configuration in a production line do not majorly disrupt the production flow or lead to an incomplete or dysfunctional product. In automotive production, common manual tasks include hose connections, wire routing, and subsystem assembly. Much of the value-added work in automotive production line assembly is performed by an operator, generating the possibility of inconsistent or incomplete work. The transforming relations between operator and cyber-physical systems (CPS) resulting from automating the production chain poses the question of where operators will be most effective in a manufacturing environment [6].

The dynamic nature and behaviors exhibited by operators call for proper interfacing between the two elements determined by the environment, performed operation, and the data exchange [6]. This will yield a greater understanding of the product, process, and work system environment engaged in manually conducted helping to bridge the gap between manual work to collaborative manufacturing and collaborative to autonomy [7]. Quantifying characteristics in manual assembly requires perception and intelligent data interpretation [5, 8, 9]. In addition, further investigation in device- and worker-connected environment as well as human-automation symbiosis to aptly accommodate to the employee's skill level. In the current market, a diverse group ranging from long-standing employees to new-hires require various levels on introduction and training to the CPS and machine entities [10]. These advancements can be achieved through greater sensing, vision, and cognition technologies [11]. To provide the necessary inputs for these technological advancements, parametrization and characterization of the manual work are needed. These parametrizations are also utilized for the calibration of solutions such as wearable sensing for automotive assembly. Such solutions will provide the operator with feedback on the process as it is completed, ensuring all steps and verification of each step were completed.

Data about worker abilities, limitations, and variability must be quantified and applied to the design to improve compatibility between workers and product assembly [12]. The stronger the compatibility between the worker and the product being assembled, the greater the understanding of how the process is completed. Part of the solution to manual task classification is understanding the cycle of manual production processes through feedback of information on critical forces [13]. Active force readings based on the operator can yield the status of manually completed work in real-time. A fast-growing technology such as an augmented reality (AR) interface could provide instructive information to the operator enhancing the quality of decision making performed on the assembly line. This will provide a greater connection between the operator and machine while also creating a platform for remote monitoring. AR implementation will require additional information on the task at hand, objects in play, and how intelligence is to be generated [14].

Example applications demonstrated in this paper are hose connections and subsequent verification testing of connectors. These force readings can then be used to provide feedback to an operator or a robot giving subjective analytical contact detection [15]. As connections are manually completed, the operator applies a combination of normal and shear force to one half of the connector to join the two parts, with shear being the driving factor towards completing the connection. These force measurements can then be parameterized by creating target force windows and sample-based statistical probability to numerically prove the outcome of a connection. This yields the categorization of the connections and manual checks for review by the worker. Connectors categorized as incomplete can be reinvestigated, ensuring the components are assembled correctly in initial assembly, minimizing future rework after leaving production rework and field failures.

## 2.2 Quality Assurance for Assembly Connections

Assemblies are simply groupings of systems or components with connections. These connections are critical to functionality as they are the mating components that transfer power, fluids, etc., and as a result, must be completed properly to have a fully functioning system without defects. Product quality in an assembly environment is a critical consideration for a manufacturer. Greater product quality and yield lead to increased competitiveness and capability in a market. Thus, monitoring and assessment tool implementation and research are growing in popularity amongst manufacturers [16]. Methods towards increase product quality range from preventative to corrective in nature. Examples of a preventative ideologies include Design for Manufacturing, Design for Assembly, Design for Manufacturing and Assembly, Poka-Yoke, and Six Sigma. A corrective ideology example is after assembly functional testing. Example methods that fall in between include Augmented Reality (AR) and sensor evaluation systems.

These tools are utilized in both automated and manually completed processes. However, they are objectively more critical when the process is completed manually by a worker. There are many tasks completed in manufacturing that are currently completed by operators due to the intuition needed by associates, complex automation steps, or time constraints. This becomes especially evident when observing the automotive industry. Final assembly is predominately completed by operators because of what they can achieve in a timely manner. Companies such as Tesla learned the value of humans in manufacturing when they experienced ”excessive automation” the manufacturing process [17]. Common tasks completed in final assembly with a more frequent failure rate are connection completions. This includes electrical, venting, gas, and coolant connections. Each of the previously mentioned monitoring and assessment tools will be discussed in with connection defects.

Design for Manufacturing (DFM) is based on the ideology of creating a part or product with manufacturing in mind, merging the design requirements with production methods. This helps to minimize costs and defects with the resulting product. Design for Assembly (DFA) follows a similar premise but focuses on an effective design for ease of assembly. This also progresses towards minimized cost and defects in the part or product. Since the two design methods follow many of the same principles, they are often considered together and referred to as Design for Manufacturing and Assembly (DFMA) [18]. In regards to connection in the automotive industry, these techniques often fall short due to excessive costs. Due to the abundant amount of connections seen on, for example, an assembled vehicle, automotive OEMs and supplier abstain from the use of more expensive, yet more effective, connector types. The price difference may be marginal dissociated from the final product ($0.01-$0.02/connector) but quickly add up in mass production.

A preventative and corrective concept developed by Shigeo Shingo is Poka-yoke. Poka-yoke is a simple concept attempting to mistake-proof a process. It involves avoiding problems by correcting the process making it impossible for errors to occur or ensuring they are obvious when they do occur [19]. Poka-yoke is more of a method and ideology rather than a direct solution. Some studies have shown that this method may only be applicable to simple assemblies [20]. A common example of an implemented Poka-yoke system is a simple electrical plug that can only be connected to its mating pair in one orientation as shown in Fig. 2.1. However, even basic concepts such as the unidirectional electrical connection can be misused and plugged in unsuitable orientations. Thus, a more robust solution for capturing operator physical data is warranted.

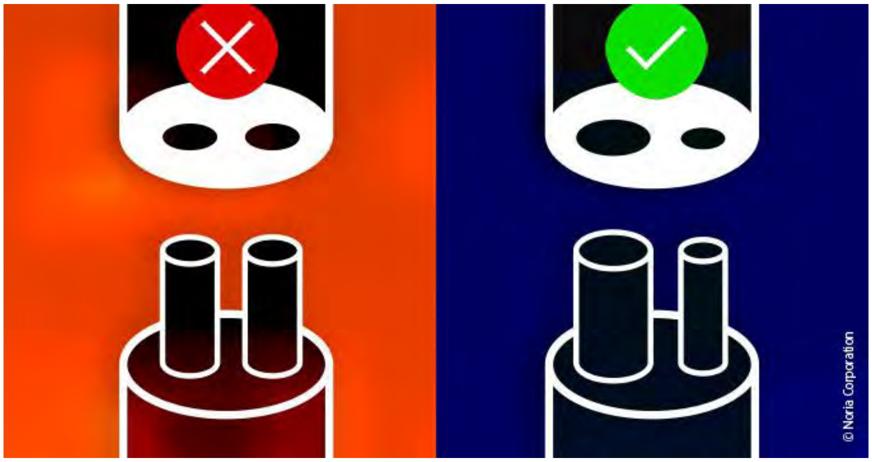


Figure 2.1: Poka-yoke principle applied to an electrical plug [21]

Six Sigma presents a management style to improve an organization's products, services, and processes by continually reducing defects [22]. This minimization of defects statistically equates to less than 3.4 defects per million opportunities or a success rate of 99.9997% [16]. Tools within Six Sigma to eliminate defect causing factors include DMAIC and DFSS. DMAIC stands for define, measure, analyze, improve, and control. This method is a closed-loop process that eliminates unproductive steps, often focuses on new measurements, and applies technology for continuous improvement. This method has a focus on integrating specific tools into each step as a means of problem solving [23]. DFSS, Design for Six Sigma, focuses on the design of processes and products to eliminate corrective steps towards achieving Six Sigma quality levels. This is achieved through tools, training, and measurements [23]. This ideology has proven to be beneficial for a number of companies despite its similarities to previous management approaches. However, this method requires tools for defect mitigation. The management style can benefit a manufacturer as a whole but is less applicable for single process improvement. Thus, an accompanying robust tool is needed for a successful implementation.

Corrective practices are commonly utilized in manufacturing. These corrective approaches normally begin with functional validation testing which may take the form of an electrical test, pressure test, etc.. These tools are used to ensure the product can perform its intended duty without error from assembly or creation. This is an effective method for detecting the vast majority of errors, but can miss critical defects. In regards to connections in manufacturing, a common error that is overlooked by these corrective processes is a partial connection. This partial connection can pass validation testing, then later loosen causing expensive and timely rework. The longer the partial connection goes unnoticed, the more detrimental to the company. This is due to the additional disassembly necessary to access the dis-functional connector and the person(s) in possession at the time of fault. A manufacturer can experience a diminished brand image due to a preventable error if the product fails in the distributor's or consumer's possession. These partial connections can be avoided, but are continuing to be a challenge that plagues the manufacturing industry. This is largely due to the lack of a adaptable solution capable of detecting incomplete connections or processes.

AR has become more common in the manufacturing industry over the recent years. This technology enable the mediation of information in spatial relation to the environmental context [24]. Mainly, this implies adding digital objects and information to a user's real-world view, enabling direct interaction with information which has a direct spatial relation to the real environment. This readily interpretable data can be advantageous in informing the operator of critical information and to facilitate training without waste [25]. While this technology can offer intuitive to the user, it is less applicable to evaluate the status of connectors in manufacturing assembly. It could be used as a supporting system to indicate incomplete connections, but this would inject unwarranted costs into the system. Simpler indication methods can be used.

There are numerous sensing approaches for validation currently marketed. Vision systems and sound recognition are two practices currently leveraged in industry. Companies such as Detect-It and Clarifai offer forms of artificial intelligence as a means of object/defect recognition. Software such as these can train computer algorithms to digitally recognize faults in a system such as an incomplete connection using vision detection. This would require a vast number of images (if visually based) to indicate a complete versus partial connection to the computer. Systems such as these also require strategically placed sensors that may be disrupted by the noisy manufacturing environment if audio based or disruptions from moving parts if visual based. As the processes grow in complexity, the number of sensors required grows, and their may not be viable locations available to accurately observe the connections [26, 27, 28].

Wearable systems are another approach to monitor, evaluate, or capture operator actions. This technology involves sensing directly affixed/on the operator. This will be further discussed in the next section.

## 2.3 Wearable Sensing for Process Confirmation

To enable Industry 4.0 approaches, many researchers and companies are exploring the realm of increased sensing capabilities for process monitoring. This varies in application with integration options including sensors on the machine, operator, or in the environment. The processes utilize the captured data to ensure the process is being completed correctly, perform predictive maintenance, and inspire a level of accountability through saved data [29].

Sensors incorporated onto the operator have a growing popularity due to their ability to measure what was previously subjective due to the need for operator intuition, the eased workload on the operator, provide critical information for preservation of the operator, and enable a human-machine symbiosis [30, 31, 32]. With the changing tides in manufacturing, operators are being recognized as a manufacturing capability that possess the intuitive and flexible nature that creates value over a robotic counterpart [33]. Thus, wearable sensing platforms are increasing in research and implementation, but they are not always successful. This often stems from worker flow and human factors related shortcomings. Tools and programs developed for manufacturing workers can perform their function with high accuracy but forgo necessary human factors considerations to promote usage of the tool or process. The tool or process should be easy to use and not interrupt workflow to ensure the quality improvement measure maintains its place in production.

Research conducted by Schönig et al. creates a decentralized production environment that offers control and support to the workers through stored and acquired production data. This allows the workers to access user-specific information and subsidies as desired [34]. The data exchange platforms empowered through wearable process monitoring sensing can be captured through a wide array of devices such as gloves, smartwatches, prosthesis, and more detailed by Muzny et al. [31]. This collection can then be delivered in equally as many ways through most often visual, audible, and tactile feedback loops. Examples of these technologies include wearable scanners such as the ProGlove and Bosch iGlove [35, 36], health monitoring devices such as smart watches, body worn motion sensing [37], and inner body activation such as tendon monitoring for loading evaluation [38]. Another technology, shown in Fig. 2.2, is the GM-NASA produced force augmenting glove which amplifies operator gripping capabilities [39]; this is one of the few models that incorporate actuation augmentation together with sensing.

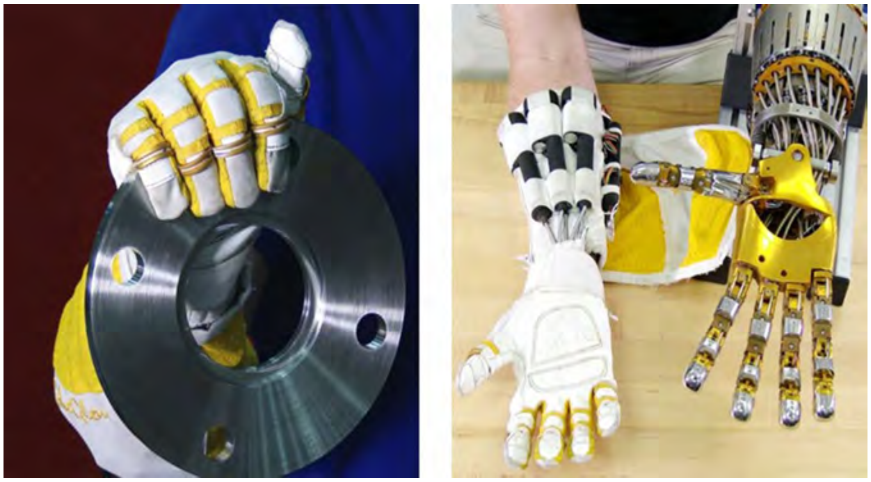


Figure 2.2: GM-Nasa Robo-Glove: Wearable technology that reduces the force needed to operate tools

Analysis of the large quantities of data generated through the use of such sensors has benefited from recent and rapid advances in the capability and accessibility of machine learning technologies [40]. This application helps expedite the processing of collected data in controlled experiments [41], in addition to enabling analysis supporting the real-time monitoring of workers' ergonomics [37, 42].

## 2.4 Force and Pressure Sensing Gloves

Throughout manufacturing, operators play a critical role in final assembly, and often times they wear gloves for added protection, comfort, or utility. This offers a seamless opportunity to introduce process monitoring technologies directly on or in the glove that the associates already wear. A handful of products have been released targeting the market of force sensing gloves. Many rely on different operating principles than the glove proposed in this research. Most marketed solutions utilize pressure sensors to measure pressure distribution or activation forces where the glove presented in this paper offers shear and normal force measurement capabilities [43, 44, 45, 46, 47]. These solutions can provide valuable feedback in a research environment, but they are unfit for the wear and tear introduced by a production environment. This is due to their limited robustness in durability and constrained data processing circuit for general applications. Further discussions on marketed and research oriented products are found below.

The Grip Pressure Sensor Glove from Tactilus can display pressure profiles through a series of pressure sensors located along all five fingers. These piezoresistive sensors can each measure up to 100 psi at a maximum of 1000 Hz. The sensor glove offers a repeatability of ±4% and an accuracy of ±10%. However, the greatest flaw of this sensor glove are the invasive sensor leads. The glove has fives sensors for each finger and 4 for the thumb. This is a total of 24 sensors with long leads that interferes with natural operator movement. This renders the sensor glove unfit for an assembly environment as the leads could easily be snagged during operation [48]. The sensor glove is shown below in Fig. 2.3.

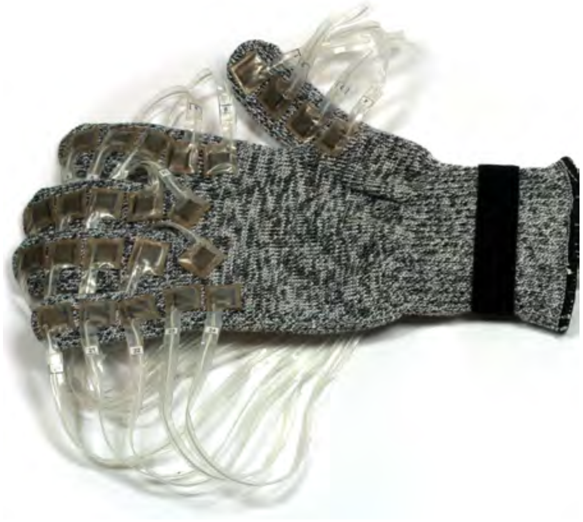


Figure 2.3: Tactilus grip pressure sensor glove with 24 sensors

Tactilus offers another version of their Grip Pressure Sensor Glove, shown in Fig. 2.4, that utilizes only one sensor on the index, middle, ring, and pinky fingers. The sensor glove boasts the same measurement range, scan speed, and repeatability. It does not provide an accuracy measure. The sensor glove can take a maximum of 9.04 lbs (4.1 kg) and claims to be waterproof. This sensor glove is unfit for most assembly environments due to the limited endurable force and measurement capability (limited to normal force alone). The sensor glove is advertised for recreational use [49].

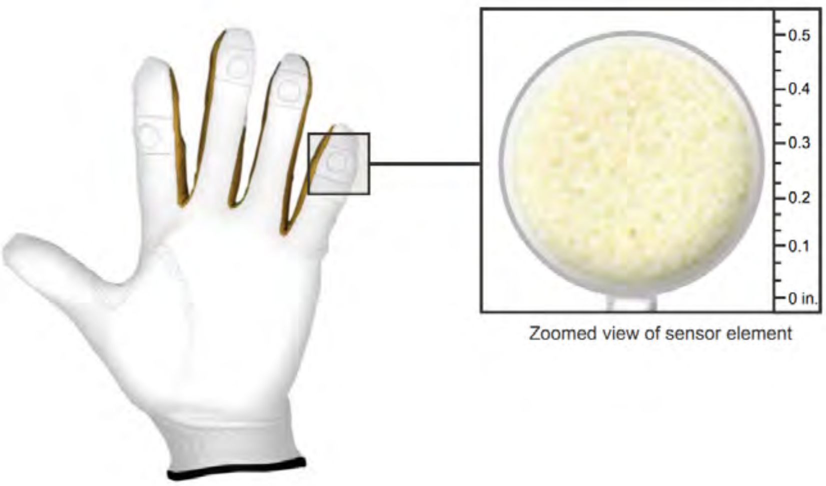


Figure 2.4: Tactilus grip pressure sensor glove with 4 sensors

Pressure Profile Systems (PPS) offers full hand pressure measurements with the addition of normal force measurements. The sensor has low measurement capabilities (0.1oz/3g) and can measure up to 80 psi with its 65 sensing elements. The sensor glove can record wirelessly but cannot display results until connected to a computer. The sensor glove is advertised for manufacturing safety and worker's comp assessments. Applications extend to assisting with ergonomic product design. This product is useful in a controlled environment such as research or the applications listed beforehand. The glove is unfit for manufacturing production because of its high price tag ($25,000) and limited durability [44]. The sensor glove is shown in Fig. 2.5

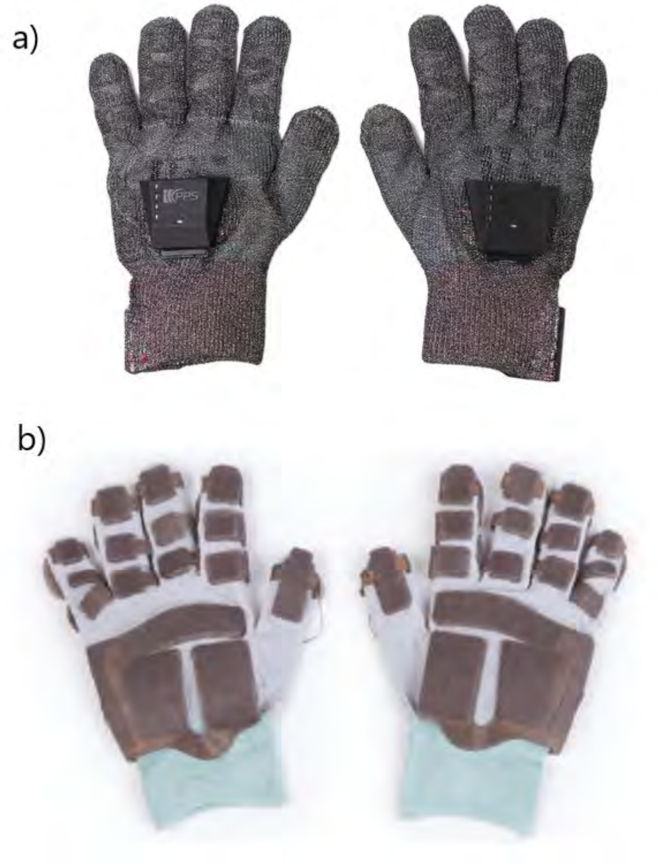


Figure 2.5: PPS TactileGlove - hand pressure and force measurement a) fully assembly glove b) inner glove lining with sensors affixed

Novel.de offers a similar solution to PPS with their Pliance Glove Sensor as shown in Fig. 2.6. This minimally invasive also collects full hand pressure measurements. 256 Sensors are applied to each fingertip and a larger sheet to the palm. This data is recorded onto an SD card for later processing or analyzing in their Pliance software. Possible applications are also limited to controlled environments, mainly research. The sensor glove is unfit for manufacturing production because of their high price tag ($20,000) and limited durability [46].

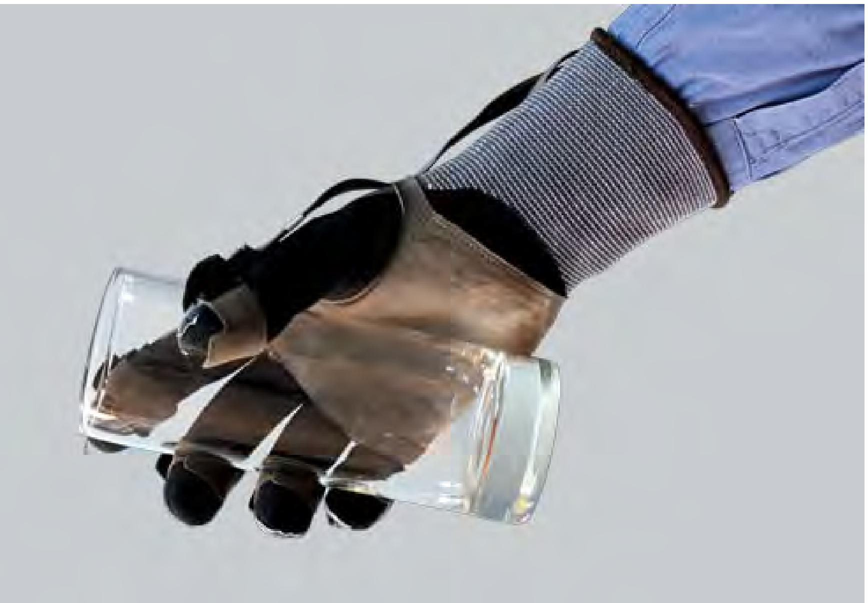


Figure 2.6: Novel.de Pliance glove sensor

The Hoggan Scientific ergoPAK ergoGlove, shown in Fig. 2.7 offers a more reasonable price at $4,795. The sensor glove is composed of up to 8 normal force sensors with a wrist mounted collection unit that can measure up to 100 lbs. Sensors are not directly affixed to the glove. The manufacturer provides latex finger cots to hold the sensors in place. The sensor glove must also be within 25 feet of the operating computer to collect data. This sensor glove has glaring issues with durability as the sensors are not fixed to the glove in a secure way. The measurement is also limited to normal force alone. Cost is still relatively high to be implemented in a harsh assembly environment even though it is a fraction of the other solutions. [45, 50].

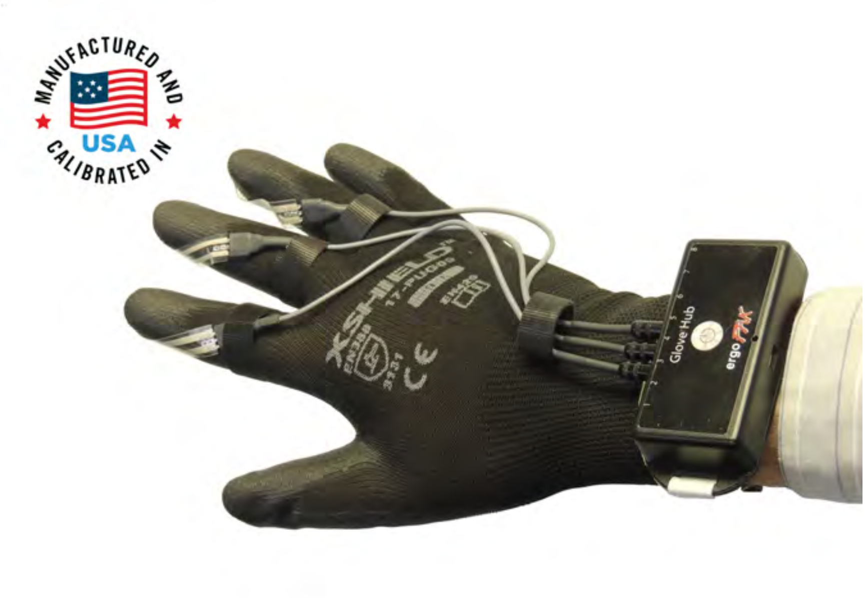


Figure 2.7: Hoggan Scientific ergoPAK ergoGlove

An overview of marketed sensor glove products included advantages and shortcomings can be found in Table 2.1 below. BMW assembly line workers reported using anywhere from two pairs of gloves per day to one pair lasting a week. This is takt dependent. With the frequency being relatively high, this further validates the need for a cost-effective, durable solution.

Table 2.1: Marketed force and pressure sensing glove overview

| Glove | Advantages | Shortcomings |
|---|---|---|
| Tactilus Grip Pressure Sensor Glove – Research | High accuracy and repeatability, large measurement area | Invasive sensor leads, durability, fine measurements, only normal force |
| Tactilus Grip Pressure Sensor Glove – Recreational | High accuracy and repeatability, waterproof, durable, minimally invasive | Accuracy, low measurement capability, only normal force |
| Pressure Profile Systems (PPS) pressure sensor glove | Full hand measurement, fine measurements, wireless recording, minimally invasive | Only pressure/normal force, durability, high price tag ($25,000), research environment use |
| Novel.de pliance sensor glove | Full hand measurement, wireless recording, minimally invasive | Only pressure/normal force, durability, high price tag ($25,000), research environment use |
| Hoggan Scientific ergoPAK ergoGlove | More reasonable price ($4,795), high measurement range, wireless recording | Only normal force, sensors not directly affixed to glove, durability, limited wireless range (25 ft), repeatability, still a high price |

Research done by Hammond III *et al.* on a soft sensor embedded pressure system is minimally invasive and captures motion and tactile data from the operator's finger [43]. The finger glove, shown in Fig. 2.8, is composed of three strain sensors to measure joint motion and three pressure sensors to measure interaction dynamics. The system is designed for use in a research environment to understand gesturing and grasping mechanics. The sensors are cast liquid-metal embedded elastomer. The research claims to be able to capture hundreds of Newtons and maintain repeatability. However, the finger glove is vulnerable to hazards in production (sharp corners, excessive forces). The researchers reported interference with grasp acquisition and sensor failure under high loads [43].

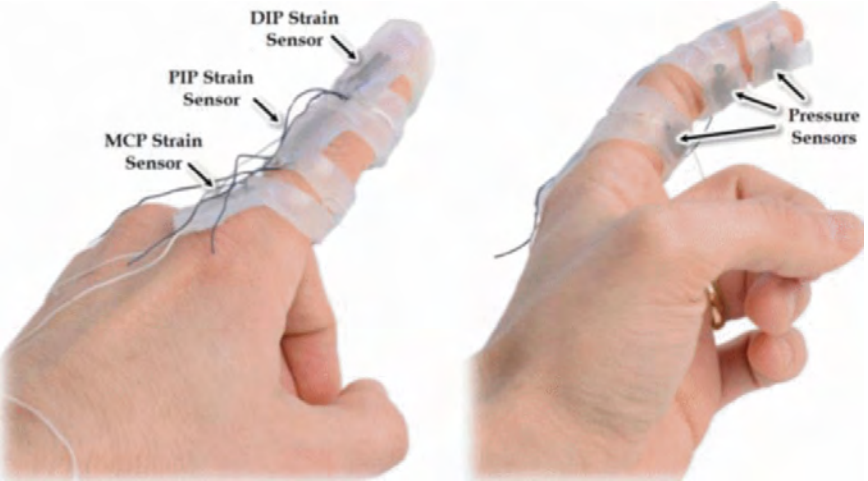


Figure 2.8: Finger glove sensor system developed by Hammond III *et al.*

Research by Yin *et al.* investigates what is needed to create a high performing wearable system that presents flexibility and stretchability to connect the user to machines in the environment [51]. Their application focuses on the integration with Human Machine Interaction (HMI) devices. These HMI wearables enable real-time acquisition of physiology and surrounding signals through accurate and flexible sensors. The research provides a breakdown of sensing systems, formats, and communication methods for HMI development [51]. Shown below is an example use of a proposed HMI wearable device by Yin *et al.* in Fig. 2.9.

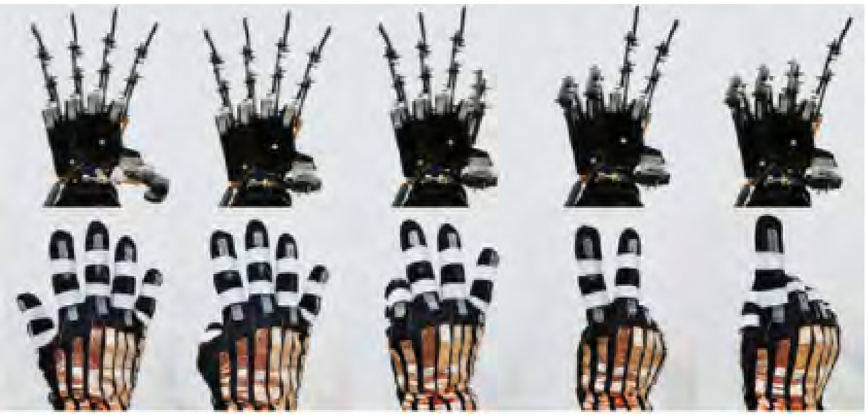


Figure 2.9: Robot hand gesturing control through sensor glove by Yin *et al.* [52]

Research conducted by Lin *et al.* investigated a sensor glove with 18 IMUs and 5 force sensing resistors affixed to the fingertips [53]. The IMUs were place across the hand with a focus on the fingers. One IMU is place on the wrist portion of the glove. The combined sensors measure hand kinematics and fingertip force. The glove utilized 3D printed disks to focus the force onto the Flexiforce resistive sensors. The force sensors were linear and relatively accurate to actual applied force with a mean absolute error of 1.47 N. While the sensor glove was noted as highly accurate for joint angle and applied force estimation by the authors, the durability and invasion of the glove is problematic. This final form is not suitable outside of a controlled environment [53]. The developed sensor glove is shown in Fig. 2.10

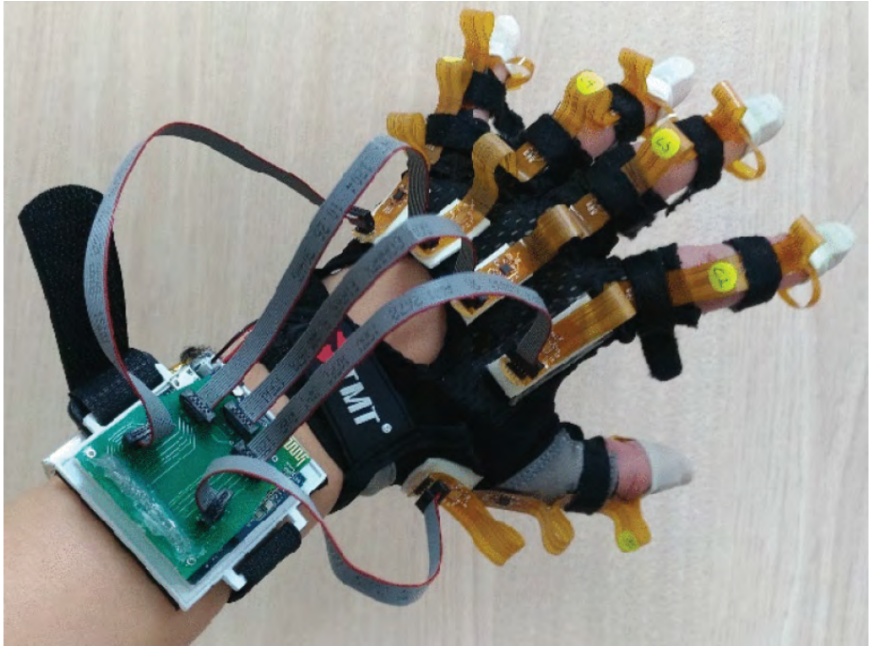


Figure 2.10: Developed force and motion sensing glove by Lin *et al.*

In this work, Austin *et al.* created a force sensing glove using 14 Flexiforce sensors affixed to the 5 fingers is shown in Fig. 2.11 [54]. This research investigates forces required for common daily activities for eventual actuator sourcing for prosthetics, exoskeletons, and orthotics. The use of commercial sensors yielded stable and repeatable results. The sensor glove experienced issues with activation during the recorded tasks because of the sensor placements and small measurement area of each sensor. The sensor leads all run across the palm side of the hand leaving them vulnerable to damage. This makes the sensor glove more suitable for controlled research environments [54].

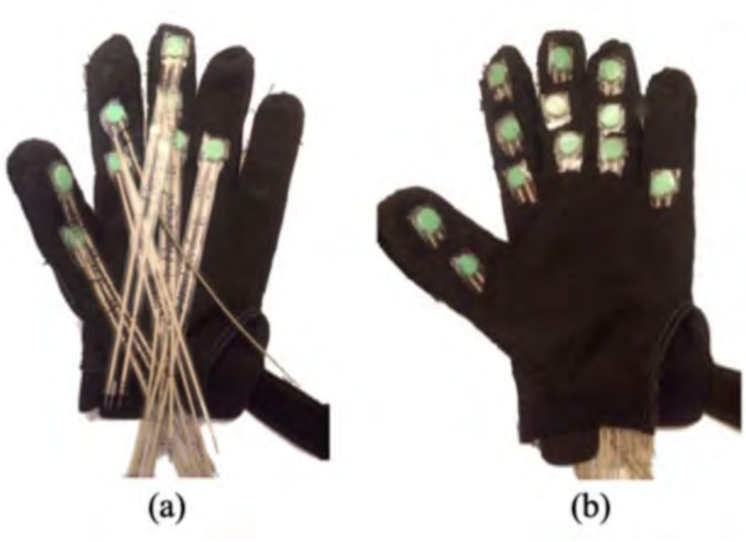


Figure 2.11: Normal force sensing glove from the research conducted by Austin *et al.* a) uncovered sensor lead view b) covered sensor lead view [54]

Similar to the work conducted by Yin *et al.*, Meng *et al.* investigated a sensor glove for use in HMI. A screen printed flexible electrode based force sensor is paired with a Functional Liquid Metal (FLM) strain sensor as shown in Fig. 2.12 [55]. The research claims the screen printed sensor to measure both normal and tangential forces. The tangential force refers to the shear force discussed in this thesis work. Both sensors offer linear measurements, high sensitivity, and a low cost. However, the sensor glove lacks in durability and measurement range. It is sufficient for an HMI application of movement replication, but not for an assembly line where higher forces are achieved around potentially damaging edges and tools [55].

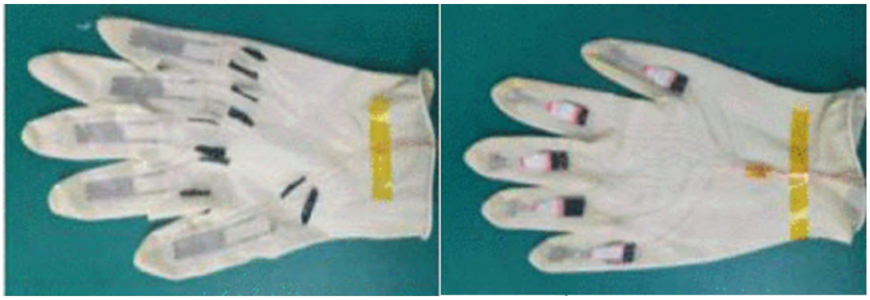


Figure 2.12: Strain sensors integrated onto the back of the finger with force sensors integrated to the finger tips from Meng *et al.* [55]

Borik *et al.* focuses on the measurement of hand press force (grip forces) with their proposed sensor glove. This sensor glove utilized force sensing resistors placed at the locations shown in Fig. 2.13 [56]. When measuring up to 50 N, an observable error of ±5 N. Higher force show less error with the minimum suggested force being 12 N. Therefore, the glove is not suitable for fine tool manipulation. A final version of the sensor glove is not shown in the research suggesting that there is much more development needed to use the sensor glove outside of a research environment [56].

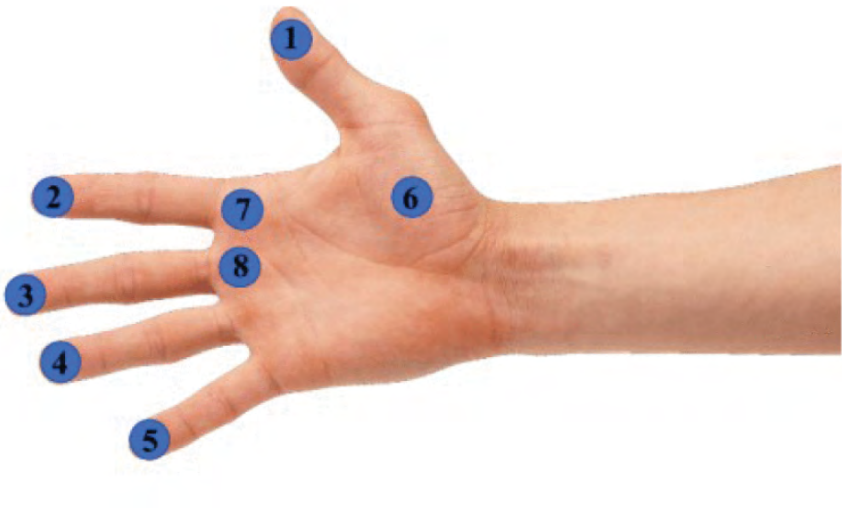


Figure 2.13: Force sensing resistor placement on the hand for Borik *et al.* [56]

Park *et al.* created a force sensor glove that utilizes a capacitive approach rather than resistive [57]. This sensor glove is meant to capture grasping and manipulation from the operator. Sensors are affixed to the thumb, index, and middle fingers as shown in Fig. 2.14. This sensor glove utilizes an open-pad structure which leaves a small portion of skin open for tactile feel. This resulted in lower grip forces exerted from the operators to perform the same function. Two capacitive sensors are placed on the sides of the fingers to allow for this open-pad structure. Capacitive sensor were selected for their high sensitivity, high measurement range, and low hysterisis. Durability concerns arose from the authors with the side mounted capacitive sensors. They also specified a wide dynamic range, but only achieved up to 15 N of force. This is below the necessary threshold for an automotive environment. The open-pad structure is also not ideal for an assembly line as the gloves were originally introduced as a means of protection [57].

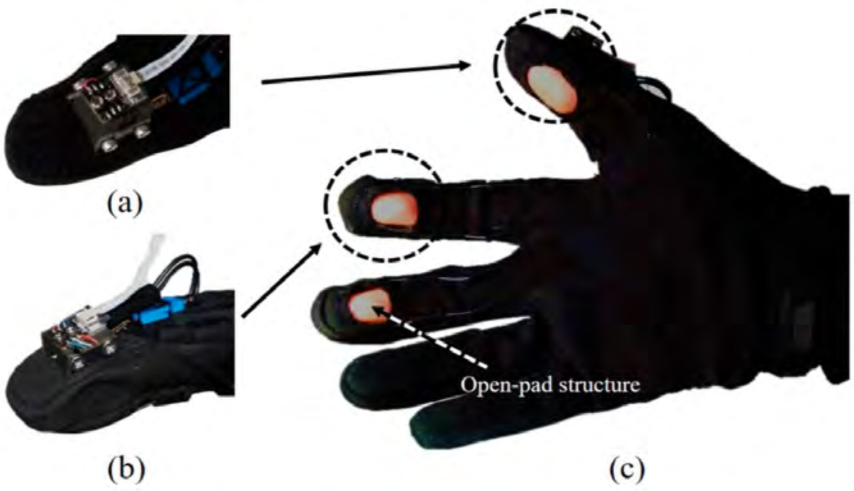


Figure 2.14: Park *et al.* capacitive force sensor glove a) back of the thumb view, b) side of the thumb view, and c) palm up view [57]

An overview of research sensor glove products included advantages and shortcomings can be found in Table 2.2 below.

Table 2.2: Research force and pressure sensing glove overview

| **Researchers (et al.)** | **Advantages** | **Shortcomings** |
|---|---|---|
| Frank Hammond III | Minimally invasive, motion and force, high measurement range, repeatable | Only normal force, durability, research environment, delicate production process |
| Ruiyang Yin | Real-time acquisition, high accuracy | Focus on Human-machine interaction use, durability, only normal force, research environment |
| Bor-Shing Lin | Motion and force measurement, high accuracy | Limited measurement area, durability, invasive, research environment |
| Edward Austin | High accuracy and repeatability, measures all five fingers | Only normal force, invasive sensor leads, durability, sensor activation issues, research environment |
| Xingyou Meng | Normal and shear force measurement, linear measurement, high sensitivity | Focus on Human-machine interaction use, durability, measurement range, research environment |
| Stefan Borik | Grip force measurement, measurement area | Low accuracy, measurement range, only normal force, durability, underdeveloped, research environment |
| Junghoon Park | Allows for tactile feel, promotes lower force utilization, high accuracy | Measurement range, limited protection, durability, research environment |

## 2.5 Operator Applied Shear Force Sensing

Marketed solutions and conducted research on shear force sensing is rather limited. This becomes even more apparent when searching for shear force sensing solutions capable of measuring operator outputs. Solutions that are presented are bulky, fragile, not flexible, have directional limitations, or have small measurement ranges [58, 59, 60, 61, 62].

In manufacturing, force sensors are encountered in machine monitoring, assembly line production, and quality testing. Force sensing wearables can improve training and development, enhance communication, reduce rework and push informed decision-making to line employees [63]. For example, force sensors can be found on machining equipment to detect when the machine, tool, or workpiece may be deviating from expected behavior. On the assembly line specifically, manufacturers are implementing sensing technologies into the operator-driven tasks to increase the operator's physical, sensing, and cognitive capabilities [64]. Commonly force sensing is commonly utilized, providing feedback such as finger activation and area of contact for the force [65]. This sensing can provide real-time data for the operator, decreasing human errors in the final product [63].

Sensor types developed for operator tactile sensing can be classified as invasive and non-invasive forms. Invasive forms could be extreme, like a physical implant incorporated into a worker, or more moderate, like physical augmentative technology such as wearable sensing [66]. Non-invasive technology can also be a form of motion capture or audible signatures [13, 67].

Many sensors for augmenting operator abilities utilized in manufacturing are capable of measuring normal force through capacitive or resistive materials. Relatively few sensors have documented findings showing the ability to measure shear force. Work has been conducted to develop and research sensor capabilities and applications with a recent focus on reducing sensor size and thickness for a less invasive solution, optimizing cost, and making the sensors more tractable [59, 43, 58]. There have been successful efforts in reducing sensor profiles, but the work is incomplete. Many efforts have looked towards the implementation of force sensing into a wearable sensing platform. Formats include a glove worn by the user to allow them to continue to perform work with minimal interference or inconvenience [13, 15, 59, 43, 68].

Advancements in the realm of measuring shear force acting between multiple surfaces or bodies effectively are much less complete than normal force sensing. Shear force sensors have been developed offering compact size, variable measurement, linear readings, and high sensitivity, but one that encompasses all attributes to a high degree does not exist [59]. Applications for these shear force sensors vary widely as well. A multi-axis sensor was developed to evaluate car seat comfort, which had a linear measurement with good sensitivity [69]. However, this design is not compact or thin enough for a glove application. Piezoresistive sensors have also been developed, which provide a highly sensitive measurement and quick response times, but fabrication techniques are costly [70]. 3D printed shear force sensors are another researched variation. Applications include thin, almost 2D designs or 3D layouts [71]. There is a need for a shear force sensor that can be used in a wearable format, that is minimally invasive to avoid disruption to workers and possesses an acceptable working range and sensitivity.

## 2.6 Data Classification Approaches

The growing presence of technology in industry and academia has enabled the likes of *Industry 4.0* and *Industry 5.0* to provide a new perspective on traditional methods. As a result, there is a more significant of background computing, monitoring, and evaluation of machines, processes, and persons. This is to minimize disruptions from stoppages or breakdowns through preventative and predictive measures. A common accompaniment to these background enterprises is the likes of a machine learning, artificial intelligence, or deep learning entity. These tools can be leveraged to perform duties and even provide judgement at rates and accuracies higher than traditional methods. One prevalent use of these tools is a classification approach. These output a probabilistic interpretation rather than a deterministic guess.

Classification algorithms have a seemingly endless number of configurations through the use of established tool packs or unique coding from scratch. There are also many methods in which a user can evaluate the classification. These methods can be more basic such as utilizing images versus sensory input, or they can be more detailed like the kernel type or learning rate. Research completed by Wang *et al.*, Yen *et al.*, and others show the evaluation of multiple learning methods [72, 73, 74]. These papers create the baseline code, then enable tools which have been proven to work to evaluate their system. This is a common practice as it is relatively easy to implement and is an effective way to find the learning method which is best at predicting for the data set. Commercial tools can also be implemented effectively, such as Watson from IBM, which received its claim to fame from participating against the best contestants on Jeopardy [75].

Countless research papers investigate machine monitoring. Studies have been conducted to measure various parameters sourced from the tool, workpiece, or machine. Most are done to minimize the use of a worn or broken machine to mitigate failure or insufficient products [76, 77, 78, 79, 80]. There are also opportunities in industry for classification algorithms to provide an extra level of protection or reinforcement to the expert's analysis. For example, the research conducted by Abhilash and Chakradhar investigated a wire EDM process to address issues that could not be corrected through process opmtimization [77].

The use of classification measures such as the ones discussed previously require some form of data stream. This could be historical data or a continuous stream of data. Nevertheless, gathered data can require significant amounts of storage. This also means that it can require a substantial amount of computing time. To alleviate storage and processing times, data reduction methods are used, such as joint dimension reduction, linear discriminant analysis, and principal component analysis [81, 73, 82].

One study conducted by Nishino *et al.* emphasized the use of a method, Taguchi's T-method, to enable data analysis early into a project. This paper showed that there can be an adequate amount of data to run a diagnosis with few datapoints. This means that analysis can begin sooner than normally conducted, helping to establish a baseline before significant amounts of data are acquired [83].

Other interesting and uncommon approaches are the works conducted by Rohani *et al.* and Kazunori *et al.*. Rohani's research team built a Gaussian process classifier that used probabilistic modeling and variational Bayesian inference. Unlike most, they utilized utilized samples with missing features rather than discarding them for training [84]. Kazunori's team highlighted the development by Toshiba Group where they used two types of deep learning methods. One was weakly supervised and the other was a transfer learning method to reduce deterioration of classification performance when reliable labeled data are lacking. This helped them to automatically classify each defect in scanning electron microscope images [85].

## 2.7 Chapter Summary

The literature review encompasses an understanding of current methodologies and tools utilized for quality assurance or operator activation measurements. The proposed research problem involves incomplete vehicle connections in automotive assembly. Wearable force sensing was determined to be a valid approach towards overcoming incomplete connections. The current marketed products and research endeavors for wearable force sensing gloves yield an insufficient platform fit for use in a manufacturing environment. Thus, a new or improved development is required. An example classification algorithm is proposed later in this research. The classification review was utilized for model selection and algorithm approaches.

# Chapter 3

# Experimental Design

This research investigates multimodal sensing on the objective determination of connection quality of three BMW connector types utilized in joining engine hose components used as venting, vacuum, or fluid transfer lines are shown in Fig. 3.1. These connectors are referred to as Type 1, Type 2, and Type 3 throughout this research. Each connector type has a locking mechanism that engages upon full completion of the two mating connector halves. This prevents the connector halves from separating under use.

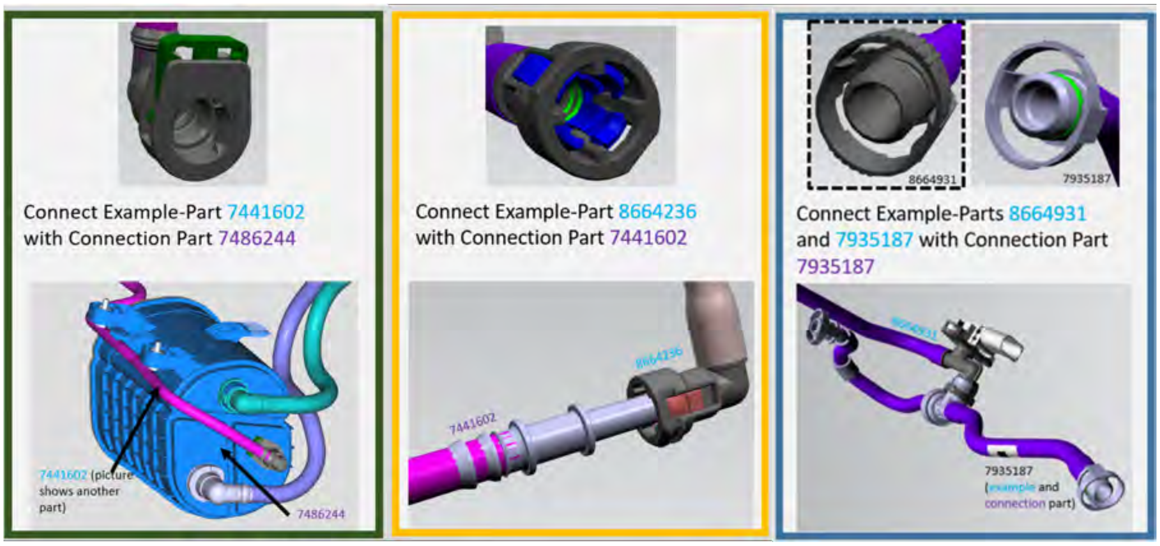

Connect Example-Part 7441602
with Connection Part 7486244
7441602 (picture
shows another
part)
7486244
Connect Example-Part 8664236
with Connection Part 7441602
8664236
7441602
8664931
7935187
Connect Example-Parts 8664931
and 7935187 with Connection Part
7935187
8664931
7935187
(example and
connection part)

Figure 3.1: Investigated BMW connectors for quality, left) vent line 1 (Type 1), middle) vent line 2 (Type 2), and right) vacuum line (Type 3)

Evaluation data is collected through a sensor glove system for real-time feedback of assembly line connection quality. Development of said sensor glove is detailed in the successive chapter. Sensors are implemented to measure a combined loading of worker applied shear and normal force coupled with acceleration and gyroscopic information. It is important to differentiate that the shear loading discussed in this research occurs across a planar motion as shown in Fig. 3.2 rather than a torsional motion. To measure forces in the shear plane a variable resistance approach is utilized for the sensor. The sensor glove system will be implemented in critical production operations that currently utilize subjective measures dependent on the operator to verify connection success. This verification testing may not indicate an incomplete connection if insufficient force is used or the operation is not completed.

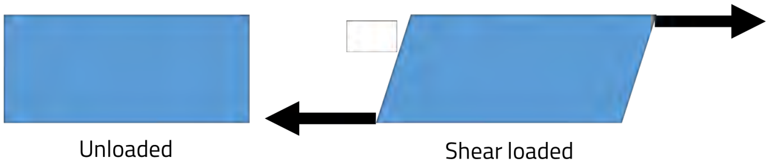


Figure 3.2: Applied shear force visual representation

The sensor glove will augment the operators in their critical role as much of final assembly in automotive manufacturing requires operator intuition and dexterity to achieve a level of efficiency and complete complex processes. The multimodal sensing will increase the confidence level of the feedback provided to the operator, ensuring the connection are successfully evaluated. This is achieved through multiple sensors requiring accountable targets before outputting the successful connection prompt.

Throughout the proposed research, a series of test methodologies and studies were conducted to ensure the investigated processes were warranted, create a better understanding of working principles, and aid in the development of the glove system. Test setups were created to understand the forces applied by the associate by mimicking movements seen on the assembly line. The test fixtures were designed to isolate applied forces and control movements exhibited when completing a connection. Types of tests completed are detailed in this section.

## 3.1 Process Failure Mode Effects Analysis

The first development was the creation of a Process Failure Mode Effects Analysis (PFMEA) spreadsheet for the connection process at BMW. This identified the critical failure modes that were needed to investigate each connector type. This analysis yielded a Risk Priority Number (RPN) which indicated the relative risk ranking of the failure modes. Higher numbers indicated higher risk. The RPN is calculated using severity, occurrence, and detection rankings as shown in Eq. 3.1. RPN is not a linear measurement as it is relative to each PFMEA. Thus, numbers should only be compared within the same failure mode analysis.

$$RPN = Severity * Occurrence * Detection \tag{3.1}$$

Two critical failure modes that were common across the board were an insufficient applied force and a misalignment while making the connection as shown in Fig. 3.3. The angular misalignment failure mode involves a sufficient amount of force that is not applied in the direction of the connection. The force is applied at an angle. The linear offset failure mode is similar, but rather than the force being applied at an angle, it is applied at a point not concentric with the connector. Of the two misalignment cases, the angular misalignment was deemed more probable on the assembly line. Connector Type 1 had a special failure case since its locking mechanism was a plastic clip on the female end which could cause a defect if it were loose when the connection was made. The secondary locking clip which is engaged by the associate after mating the two connector halves could even obstruct the opening of the connector. Another issue that could arise is misplaced force that could exhibit the push-pull-push, but it is not applied in the correct place to complete the connection. To test these failure modes, multiple participants performed tests for each connector and failure mode combination. All the tests were done manually so that test conditions could be as close as possible to the real-world application. To see all the failure modes investigated for each investigated connector type and the tools utilized for evaluation, reference the PFMEA spreadsheet in Appendix A.

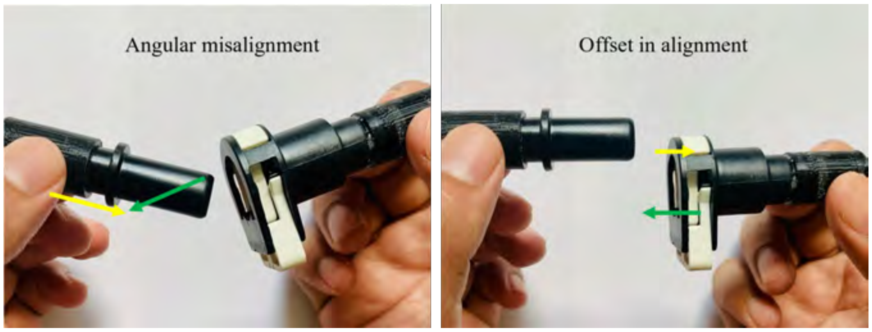


Figure 3.3: PFMEA examples with angular misalignment (left) and linear offset (right) shown

## 3.2 Insertion Force Fixture

Testing on the insertion force fixture began with bench top testing utilizing a servo motor before moving into manually completed connections by lab members. To measure applied forces from the sensor, the insertion force fixture secured the male half of the investigated BMW connectors to a sliding aluminium block that restricted motion in all directions except the *x-direction*, as seen in Fig. 3.4. This allows the aluminium block to slide freely with the force sensor, recording the force in the *x-direction* while completing the connection. To prepare for the testing, connectors were separated from hoses and collets were 3D printed to hold them in place in the fixture. To understand the predominant shear force, the data is first collected with the insertion force fixture in a direction aligned with the connection axis. The insertion force needed to complete each connection can then be applied towards correlating the shear force necessary. The force sensor used in the study was a Mark-10 series 5i force indicator and series MR01-100 transducer with a dynamic range from 0.5 N to 10 kN ± 0.15% of full scale [86].

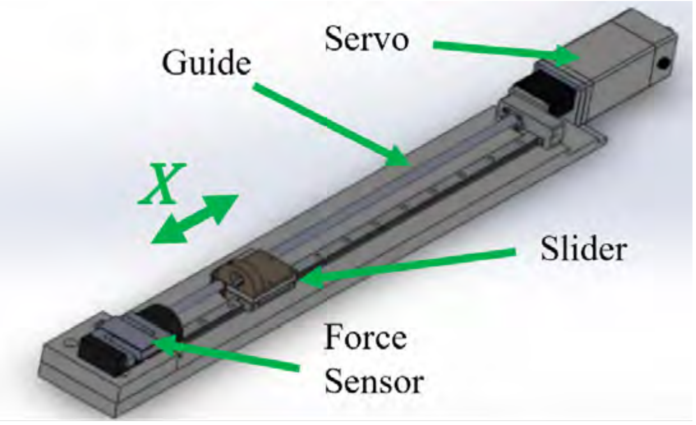


Figure 3.4: Normal force fixture setup acting in the *x-direction*

Tests began with a series of controlled speed tests were conducted across all three BMW connector types with three replications of testing, using a new connector for each replication. The testing began with completing the connections using controlled speeds via a servo motor at 2, 10, and 50 mm/s. The low speeds were selected to provide a controlled and precise connection. 50 mm/s was selected to mimic the speed at which the assembler completes the connection. An example force profile during a 50 mm/s connection is shown in Fig. 3.5. The controlled speed tests yielded activation forces necessary to successfully complete each connector type.The servo motor utilized in testing is the Applied Motion Products TSM23S-3RG with velocity control [87]. This allowed for the establishment of a controlled baseline parameters for the force necessary to complete each connector type and a curve profile for each. Samples were collected through MATLAB at a sample rate of 250 Hz to provide sufficient data that captured critical points in the observed force profiles.

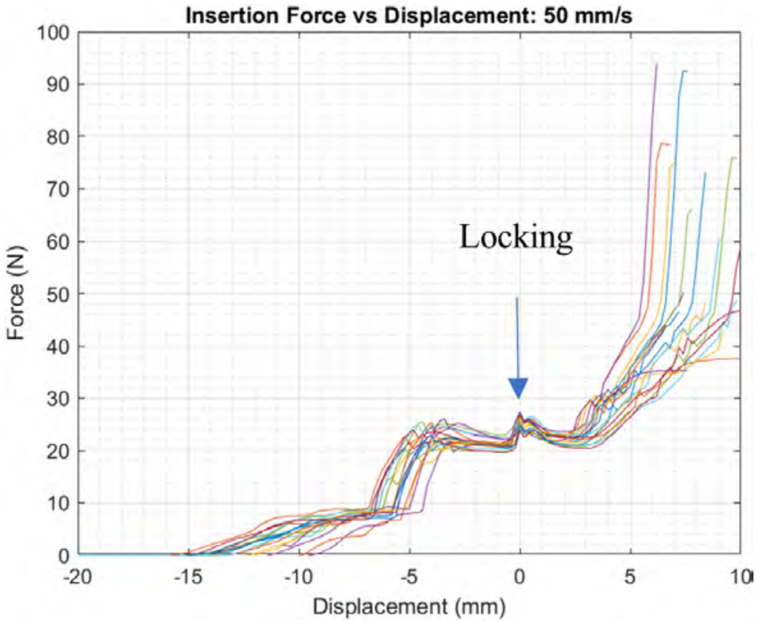


Figure 3.5: Force plots for motor driven connection at 50mm/s for connector Type 2

Later testing utilized lab participants to complete connections of the insertion force fixture without use of the servo motor or the slider referenced in 3.4. Here, participants tried to mimic the movements, speed, and timing of a BMW production line associate. The manual connections were used to establish the initial push and characterize the connection push-pull-push check. To complete all tests, including the push-pull-push check, the aluminum block in the normal force fixture was secured to one side of the force sensor. The other end of the block was fixed, restricting movement in all directions. This allowed for the force sensor to flex and record forces acting on the connector and for a relative pushing and pulling motion to be completed.

As the operator grips the hose or connectors, a normal force is applied perpendicular to the *x-direction*, which keeps the fingers from slipping. The shear force is applied in the *x-direction*, which is the same direction that the force sensor measures data. As the associate grips the half of the connector that is not secured, they utilize a grip that exhibits the shear force mentioned earlier, as seen in Fig. 3.6. The associate force acting on the connector is translated to the force sensor and recorded. This force is later used in conjunction with the developed shear force sensor understand the working principles of the sensor glove.

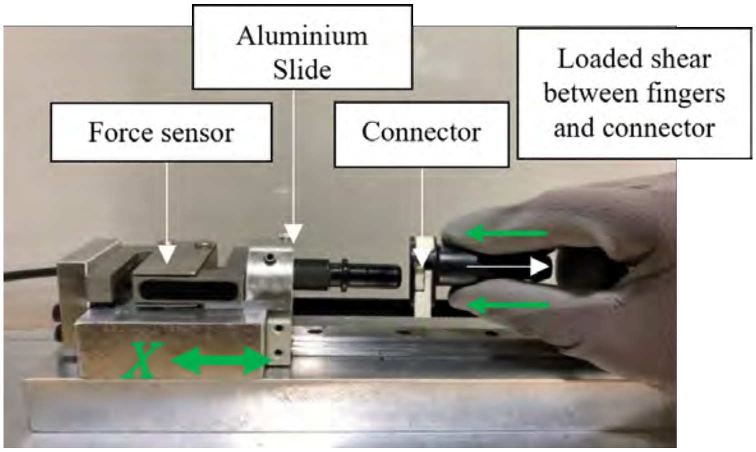


Figure 3.6: Finger engagement demonstrating the utilization of shear force in grip

## 3.3 Shear Force Fixture

To accurately quantify the shear force and allow for calibrating initial sensor designs, the go/no-go and deconstructed variable resistance sensors, a shear force fixture was developed. This fixture design was used in the beginning phases of development before the intact soft potentiometer was used in the sensor glove, such as the one seen on the fixture in the bottom image of Fig. 3.7. This fixture isolated the applied shear force by applying a force only in the x-direction on a thin sensor design. The sliding part of the fixture (orange and green, which are fixed together) moves along the blue plate and applies a known force to the block via a cord attached to the eyelet, as shown in Fig. 3.7. Friction between the two surfaces was minimized by using shims on the guide rails to decrease the contact patch. This cord was drawn over the pulley only to act parallel to the relative motion and weights are applied to the cord's end to impart known forces. The displacement was also recorded to determine the operating range of initial sensor designs.

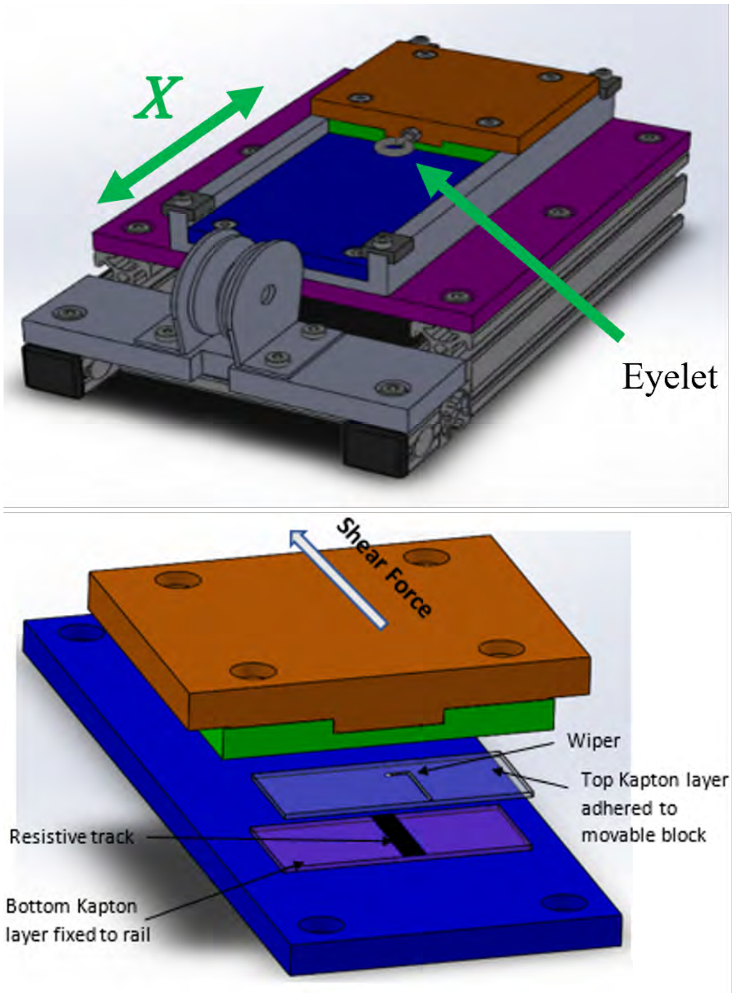

X
Eyelet
Shear Force
Wiper
Top Kapton layer adhered to movable block
Resistive track
Bottom Kapton layer fixed to rail

Figure 3.7: Shear force fixture setup acting in the x-direction

The shear sensor prototype was adhered fully to the bottom rail surface and was attached to the slide by an adhesive area of 5 mm diameter as shown in Fig. 3.8. An Arduino Uno board was used to collect displacement data and display sensor output by connecting to a PC.

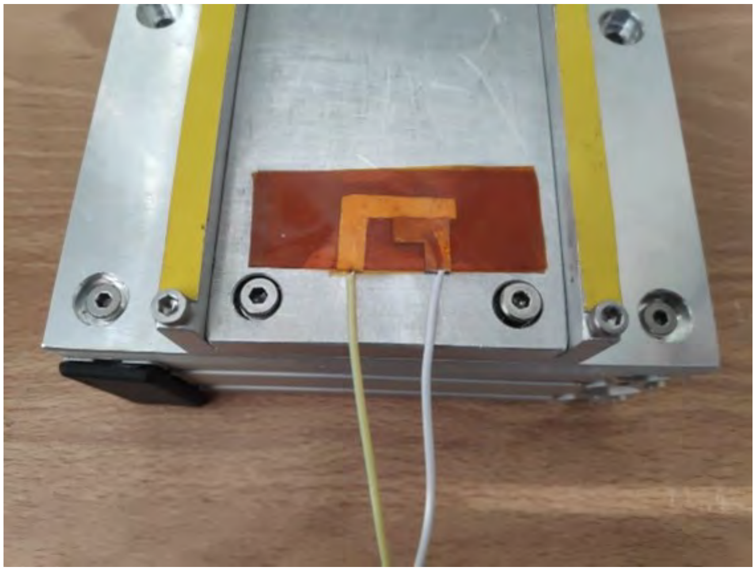


Figure 3.8: Assembled prototype placed on shear force test fixture

Testing done on the fixture focused on two areas, the repeatability of the prototype design and calibrating activation forces to inter electrode distances. Fixture components are shown in Fig. 3.9. The fixture was also utilized to determine activation forces for the go/no-go sensor. These results can be found in section 4.1.1 Go/no-go Sensor.

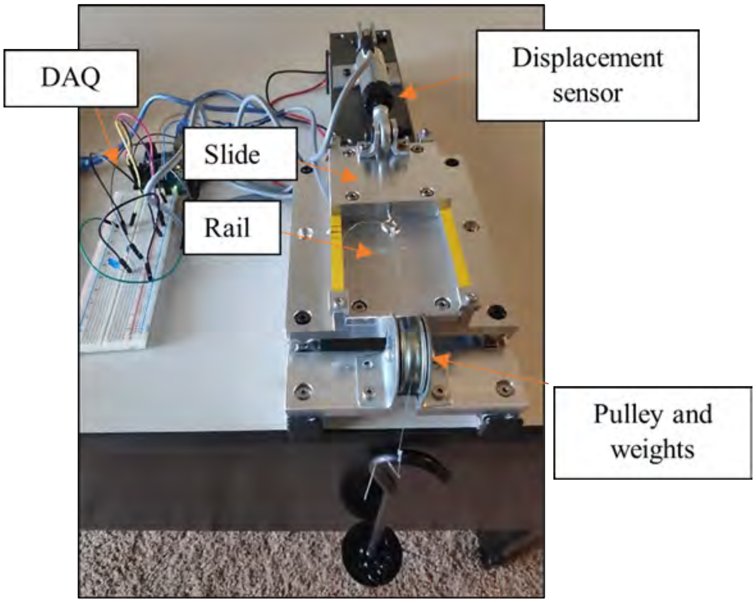


Figure 3.9: Shear force fixture data collection components

## 3.4 Shear-Normal Fixture

The shear-normal fixture was developed to understand how the sensor system reacts under a combined shear and normal loading. This helped to develop best fit curves for production line prediction and calibration measures for coupling with a robust feedback method. The concept of the fixture is similar to that of the shear force fixture. the main difference is where the relative movement takes place. On the shear force fixture, the top plate moves which allows for controlled shear forces, but applied normal forces cannot accurately be measured for the variable resistance sensor that uses the intact soft potentiometer. This is due to the small area on the puck in which the force is applied. Normal forces applied on the puck cause rocking motions and the upper green plate from Fig. 3.7 to rest on surfaces other than the puck. Therefore, the full applied normal force is not translated onto the sensor. The shear-normal fixture uses relative motion of the bottom plate to counteract this.

The developed fixture isolates the applied forces to allow monitoring of the sensor signal responses. Normal force is applied via weights placed on a platform which is then transferred in the y-direction to a guided unthreaded bolt which sits on the puck of the stacked sensor. At the bottom of the unthreaded bolt is another plate that allows relative movement of the puck while still applying the full normal force. 300FN Kapton material is placed on the underside of the bottom plate and on top of the puck to mitigate error due to friction. The fixture is shown in Fig. 3.10

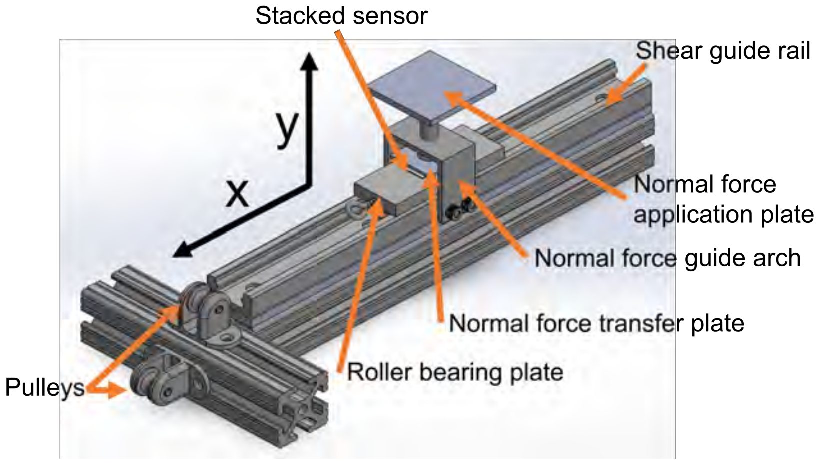


Figure 3.10: Shear and normal force isolating fixture

The stacked sensor is affixed to a guided plate with roller bearings that moves in the x-direction. An eye-bolt connected to the plate holds the cable which runs over the pulleys to apply the shear loading. A closer look at the stacked sensor placement relative to the applied normal force is shown in Fig. 3.11.

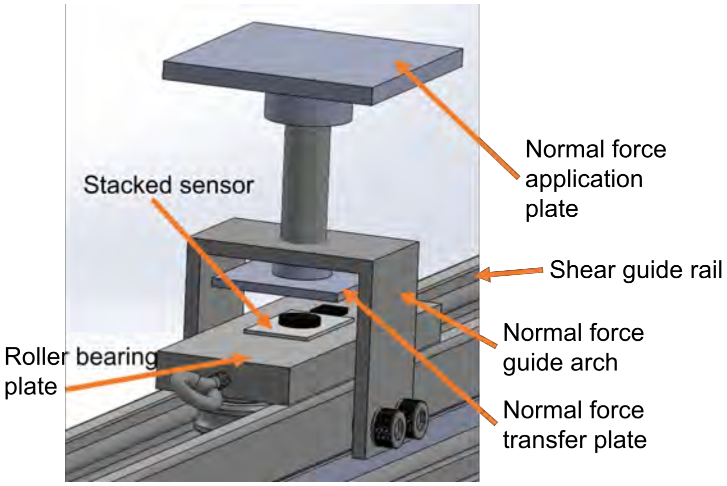


Figure 3.11: Stacked sensor placement in shear-normal fixture

The data collected from the Shear-Normal Fixture was used as a means of calibration the stacked sensor. The results from this experiment can be found in the Calibration chapter under Fixture Isolating Calibration. The physical fixture is shown in Fig. 3.12.

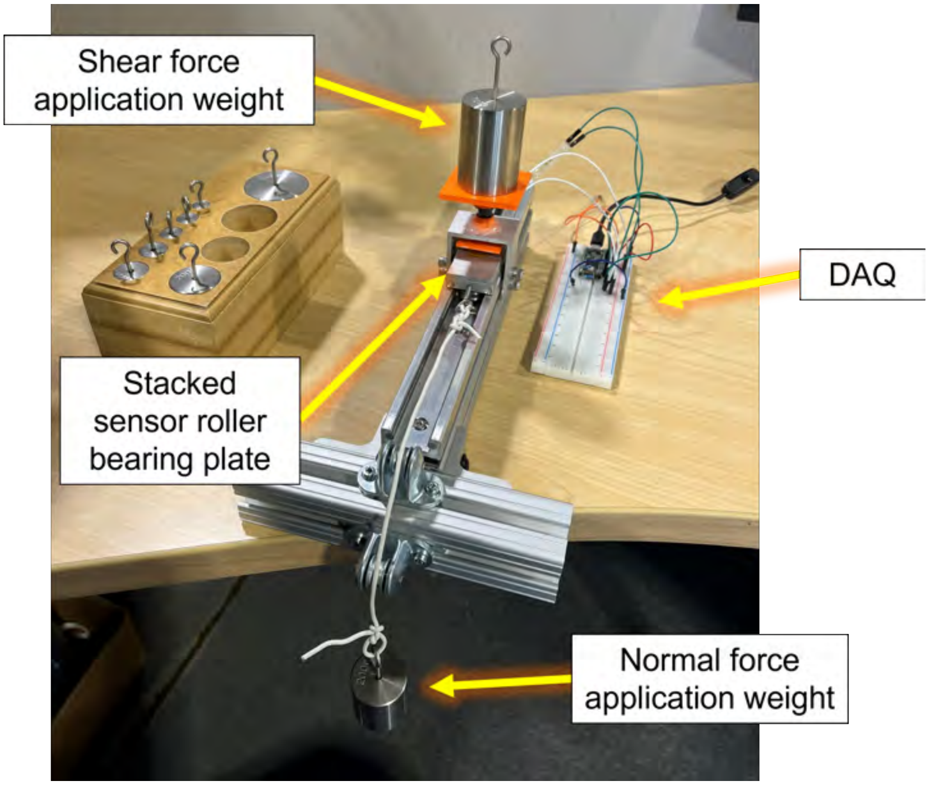

Shear force application weight
DAQ
Stacked sensor roller bearing plate
Normal force application weight

Figure 3.12: Constructed shear and normal force isolating fixture with weights applied

## 3.5 Sensor Location Study

Multiple line visits observing associates on the BMW assembly line completing the investigated connectors were conducted. These efforts were to develop viable locations for implementation of the developed force sensor to measure the forces exhibited by the associate. It also helped to understand the approaches the associates use when making connections so they could be replicated in the lab. All three BMW connector types being completed were observed at various areas of assembly, also indicating various locations of the connectors on the vehicle.

The finger and hand engagement for each connector type is shown below in Fig. 3.13. Finger and hand engagement is dependent on the associate and the location of the connector. For initial testing of the sensor, a common point of engagement was used. Once initial testing with the sensor was verified, further research efforts could be conducted to determine how to adapt the sensor and glove implementation to measure forces at the other engagement areas. This allowed opportunities to apply the sensor glove to other processes in final assembly as well.

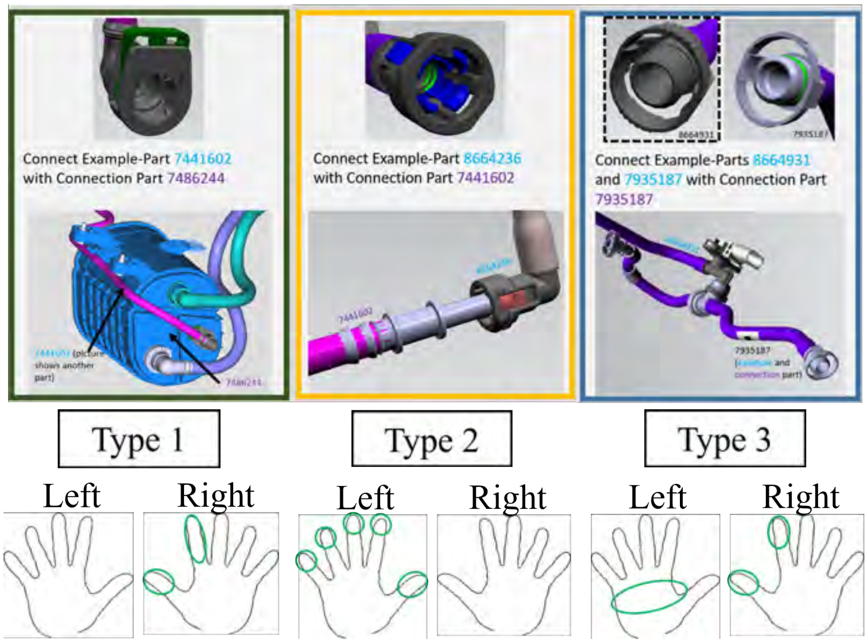

Connect Example-Part 7441602
with Connection Part 7486244
(picture shows another part)
Connect Example-Part 8664236
with Connection Part 7441602
7441602
8664931
7935187
Connect Example-Parts 8664931
and 7935187 with Connection Part
7935187
7935187
and
part)
Type 1
Type 2
Type 3
Left
Right
Left
Right
Left
Right

Figure 3.13: Finger and hand engagement for each connector type from line observations

A more detailed breakdown of the results from the study are depicted in Table 3.1. Here, the operator approach and hand/finger engagement for each connector type is detailed. This indicates how the highlighted regions from the previous figure are activated as connections are completed. Common engagement across all three connector types is the thumbprint utilization.

Table 3.1: Type 1, 2, and 3 verbal representation of finger and hand engagement when completing connections

| **Type 1** | |
|---|---|
| Approach | Head and waist height, directly in front of associate |
| Primary hand | Right dominant, left used when associate pushes both sides of connector |
| Finger engagement | Primarily thumb and pointer |
| Parts of finger | Primarily thumbprint, and side of pointer |
| **Type 2** | |
| Approach | Waist height, reaching towards right rear of engine bay |
| Primary hand | Left |
| Finger engagement | Thumb and other finger tips |
| Parts of finger | Thumbprint and other finger prints |
| **Type 3** | |
| Approach | Top and right side of engine |
| Primary hand | Right dominant, left used when associate pushes both sides of connector |
| Finger engagement | Primarily thumb and pointer, some palm |
| Parts of finger | Primarily thumbprint, some pointer print and side |

## 3.6 Chapter Summary

The experimental work began with a PFMEA to identify critical failure modes of the investigated vehicle connectors. This yielded two main failure modes of linear offset and angular misalignment which are used later for testing and the collection of purposeful failure data on the assembly line.

A series of fixtures were developed to understand the developed sensors and investigated connectors. The insertion force fixture was used to determine connection force required, connector degradation, and repeatability. The shear force fixture helped to determine working principles and operating ranges. The shear-normal fixture isolated applied shear and normal forces under a combined loading and was used for calibration efforts.

A sensor location study which took place on the BMW assembly was conducted to understand operator activation with each connector type and location. These observations were also translated to the lab environment for recreating operator approach during experimentation and testing.

# Chapter 4

# Design of the Sensor Glove

The Sensor Evolution section within this chapter details stages of the sensor development. This is followed by the progress with the proposed printed shear force sensor. Lastly, is the evolution of the coupling wrist unit which houses the circuitry for collecting the glove data. The resistive sensor showed more promise than the go/no-go when the commercial soft potentiometer was used in its original form (intact) for shear force sensing. The printed sensor shows promise, but requires further development to achieve a robustness and durability fit for an assembly environment.

## 4.1 Sensor Evolution

### 4.1.1 Go/no-go Sensor

Go/no-go is a simple type of force sensor which can measure if the applied force has crossed a certain threshold value. It is primarily measuring displacement achieved on the application of a particular amount of force. The preliminary concept developed for this type of sensor had a construction as shown below. The working concept for the go/no-go sensor is shown in Fig. 4.1.

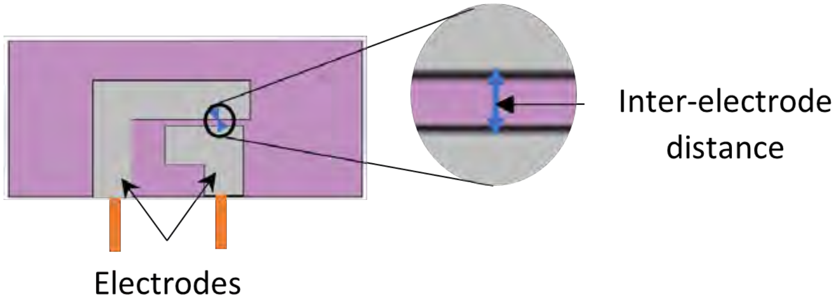


Figure 4.1: Go/no-go sensor working principle

The sensor has two electrodes which are a specific distance apart from each other. This inter-electrode distance can be set based upon the threshold force of the application. The electrodes are attached to two separate layers of Kapton material. Kapton was selected due to its dimensional stability and repeatability. The two layers of Kapton are joined together by means of an adhesive which provides flexibility for displacement of the layers in a shear direction. When the displacement reaches or exceeds the threshold amount, the electrodes make contact, completing the circuit and transmitting output to the DAQ system. Parameters of the sensor prototype used are in Table 4.1.

Table 4.1: Go/no-go prototype parameters

| **Attribute** | **Description** |
|---|---|
| Sensor top and bottom layer material | Kapton HS |
| Electrode material | Aluminum tape |
| Sensor thickness | 1 mm |
| Sensor surface area | 45 mm x 20 mm |

For the repeatability test, a single prototype was tested 20 times. The inter-electrode distance was kept as 0.2mm. Force was applied incrementally, and each test cycle was performed after an interval of 2 minutes after completion of the previous cycle. The data obtained showed that the average maximum displacement achieved for the 20 tests for a force of 18.6N was 0.943 mm with a standard deviation of 0.038 mm. No drift or trend was observed when looking at the data sequentially. A visual representation is shown in Fig. 4.2.

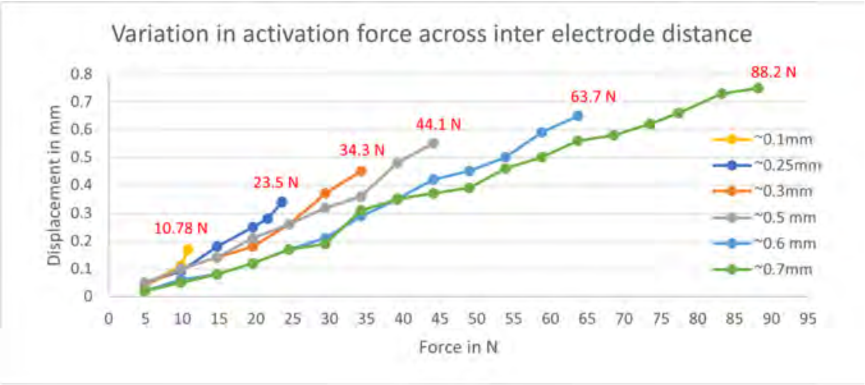


Figure 4.2: Variation in activation force across inter electrode distance, go/no-go Sensor test

To learn more about the force threshold's relationship with inter-electrode distance, a series of tests were conducted with prototypes having various inter-electrode distances starting from 0.1mm up to 0.7mm. Force was applied incrementally until the force threshold was reached. We were able to reach up to 88.2 N threshold force for the sample having 0.7mm inter-electrode distance. A visual representation is shown in Fig. 4.3.

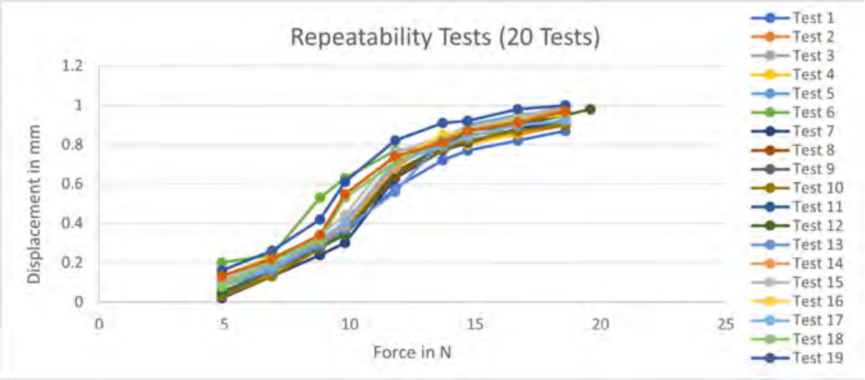


Figure 4.3: Repeatability tests for 0.2mm inter-electrode distance, go/no-go sensor test

### 4.1.2 Deconstructed Variable Resistance Sensor

While the Go - No Go type sensor is simple in construction, it is limited in its functional scope. It can only confirm whether the applied force crosses the set threshold but does not provide any kind of qualitative data. Thus, there was a requirement for a sensor that can overcome that limitation. The variable resistance shear force sensor changes its resistance output based on the input force. Thus, it can be used to get continuous force data to study any process and characterize force. There are multiple variable resistance force sensors available in the market for forces in the normal direction but not many viable options for shear force measurement. Options that were investigated are shown in Fig. 4.4. The basic design of a variable resistance sensor has a resistive element which is in contact with a contact or wiper. The relative motion between these, changes resistance of the sensor circuit. An alternative design uses a flexible conductive material like conductive fabric. The flexing of fabric changes its resistance. However, flexible conductive fabrics tested were found to have poor stability and hence were not suitable for our application. A variety of resistive elements were tested including wire wound track, rotary potentiometer track and soft linear potentiometer track. A constantan foil track was tested for performance as a resistive element, but it had poor resolution. It was found to have about 1.5 per mm change in resistance. Resistive track from a linear potentiometer made by Spectra Symbol was found suitable for our application as it provided stable output and offered a good resolution of 12.9 k across a 26.71 cm strip ( 48 per mm). The rotary potentiometer track performed similarly ( 43 per mm) but would be difficult to implement for measuring forces in the shear direction.

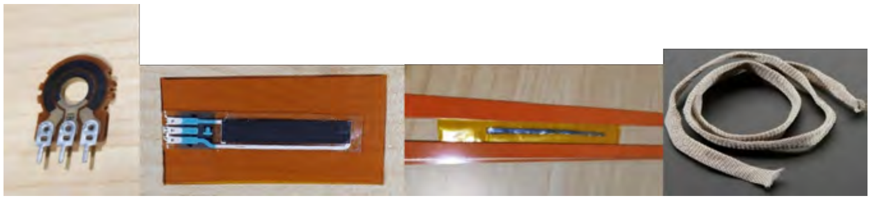


Figure 4.4: Resistive elements tested in order from left to right: rotary potentiometer, linear potentiometer, constantan foil, and conductive fabric

Once it was decided to use the resistive track from the Spectra Symbol sensor, the prototype design for the variable resistance sensor was developed. The design used Kapton sheets to hold all the sensor elements. The top layer of Kapton had the wiper attached to it while the bottom layer of Kapton held the resistive track. The Kapton pieces are held together by adhesive applied on their periphery which also provides elasticity for motion between the two layers. Once assembled, the wiper and track will be in contact with each other. When force in the shear direction is applied to the sensor, a relative motion between the wiper and resistive track will occur which will change resistance of the circuit. The variable resistance sensor prototype and component description are shown in Fig. 4.5.

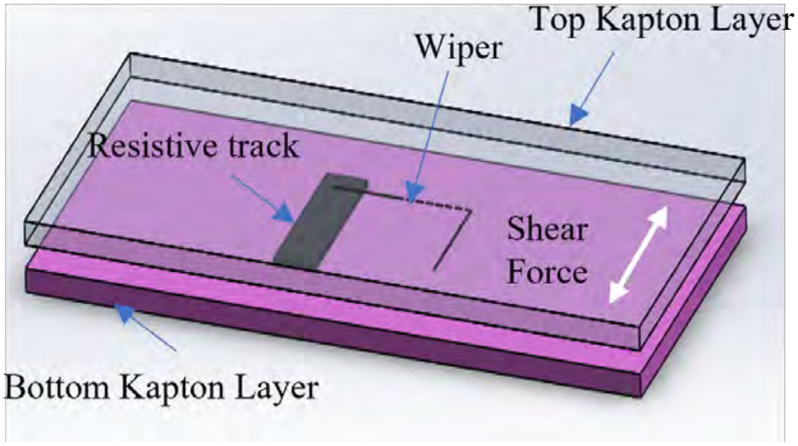


Figure 4.5: Prototype design for variable resistance sensor

There were many iterations of the deconstructed soft potentiometer variable resistance sensor. The evolution of the sensor is shown in Fig. 4.6. There were many stability issues with this sensor type. In general, fairly stable measurements were recorded when the sensors first start being uses, but they deteriorate quickly to unreadable data. Many efforts were taken to attempt to fix this issue including, creating hardline terminals, decreasing the wiper size, decreasing the resistive track width, and changing the wiper material.

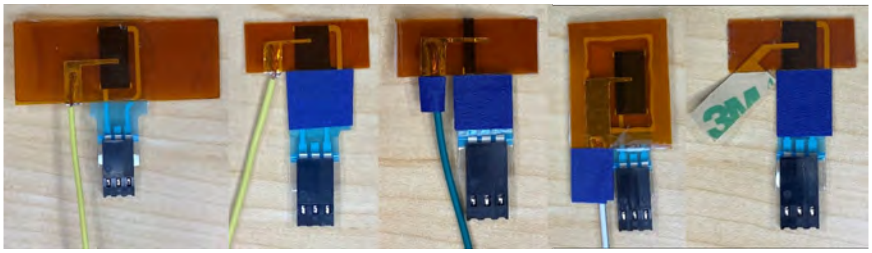


Figure 4.6: Variable resistance sensor with deconstructed soft potentiometer evolution (left to right)

When the sensors were tested on the insertion force fixture, the setup with the sensor on the glove is what is depicted in Fig. 4.7.

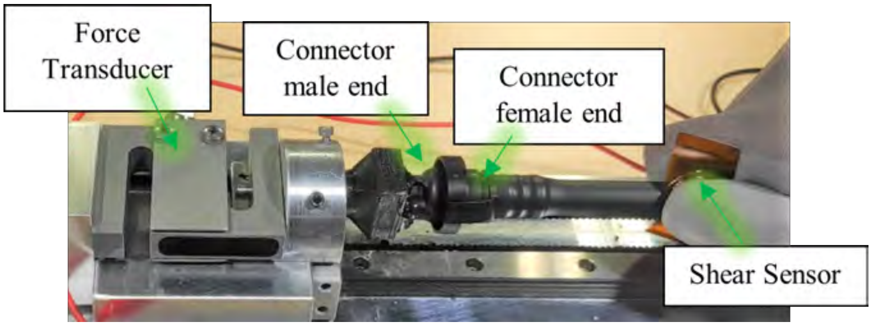


Figure 4.7: Deconstructed soft potentiometer sensor on glove for testing

The initial tests on the prototype showed a linear performance across a displacement of ¿ 3mm. This was tested on multiple prototypes. The testing was carried on the shear force fixture which had been used previously for the go/no-go sensor. An example of a test with a fresh sensor is shown in Fig. 4.8. Multiple prototypes were tested to investigate parameters like width of the wiper, adhesive to use for assembling and size of Kapton layers housing the sensor elements. Once the parameters were decided, the sensor was attached to the thumb of a nitrile dipped work glove to be used for testing by an operator. The sensor was fixed to the glove by means of hot melt adhesive to have a temporary bond and easy swapping of samples. To test the accuracy of the sensor output, a test fixture containing a force transducer was used. The fixture was developed to generate force characterization data on a set of hose connectors. The male end of the connector pair was placed on the fixture while female end was held in the hand of the lab member.

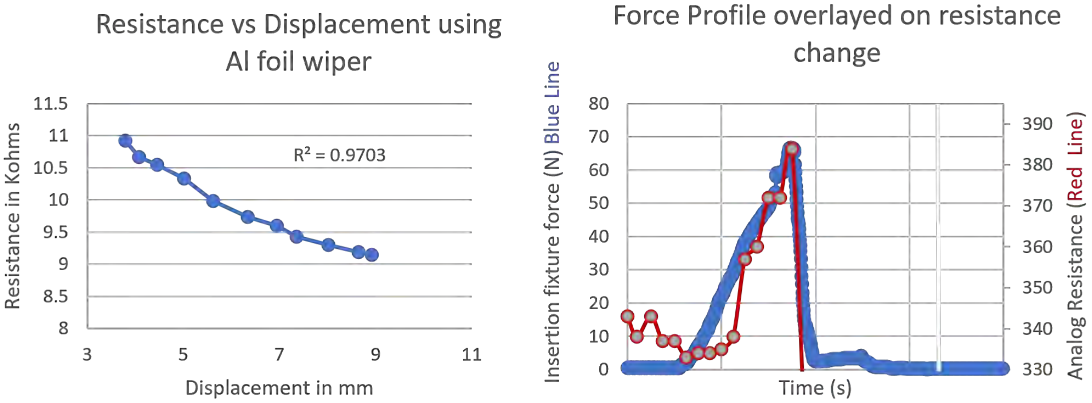


Figure 4.8: Example of output achieved from aluminum foil wiper (left) and the shear sensor output overlayed on force transducer output (right)

Even with sometime promising outputs initially, the sensors always deteriorated to output unstable measurements. Thus, an investigation was conducted to determine the cause of this. After examining microscope images, it was found that the wiper material was gouging and scratching the resistive track. This creates discontinuities on the track, skewing the measured resistance. Microscope images are shown in Fig. 4.9. The left image shows a new resistive track, and the right image is after the wiper material moved across the track.

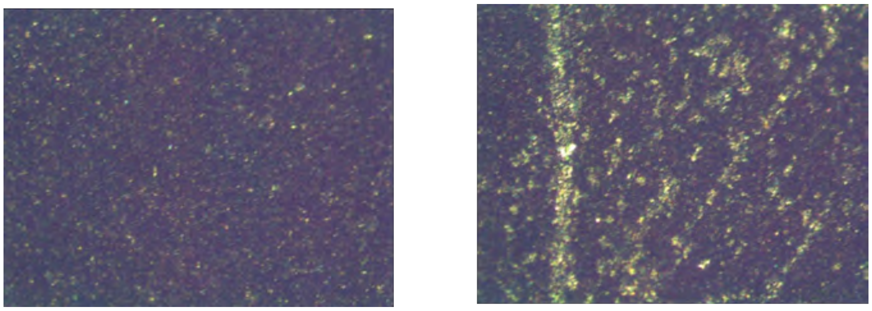


Figure 4.9: Microscope images (20x) of unused resistive track (left) and scratching on resistive track from wiper material after use (right)

### 4.1.3 Variable Resistance Guided Rail Sensor

Previous design iterations utilized relative movement between a resistive track from a soft potentiometer and an aluminum wiper to measure voltage change under a shear load. The developed sensor could successfully capture the applied shear force under controlled conditions, but the sensor lacked robustness, repeatability, and durability. The sensor rapidly deteriorated with each test creating more instability due to the aluminum wiper abrading the resistive track. The sensor was unfit for the harsh assembly environment of automotive production due to the geometry of the upper and lower Kapton layers required to achieve a significant relative movement [88].

Another proposed design for the variable resistance sensor was to use the commercial linear potentiometer intact and having a mechanism such that it can measure shear forces. The sensor prototype is shown in Fig. 4.10. This is done by mounting a tab over the soft potentiometer track which makes contact with the sensor as normal and shear forces are applied. The tab will have its motion regulated by means of a rail and end stop. In the prototype created to test the concept, we used a plastic rail, a Spectra Symbol soft potentiometer, a plastic tab and PDMS end stops. The tab was secured in place using some thread temporarily.

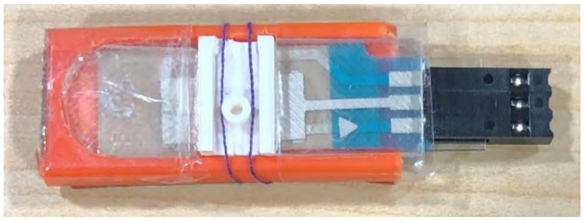


Figure 4.10: Orange rail sensor prototype with commercial soft potentiometer intact

One more alternate design was explored which used a layer of elastic/deformable material in between two layers of Kapton in our original design along with a puck on top as shown in Fig. 4.11. The puck helped activate the sensor only when actual force was being applied by the user. The elastic layer provided the motion in shear direction required to measure forces. To test the design out, multiple mix ratios of PDMS and multiple durometer hardness silicone rubber sheets were tested. The materials tested were not able to provide the magnitude of displacement required in the shear direction. This design inspired future printing designs.

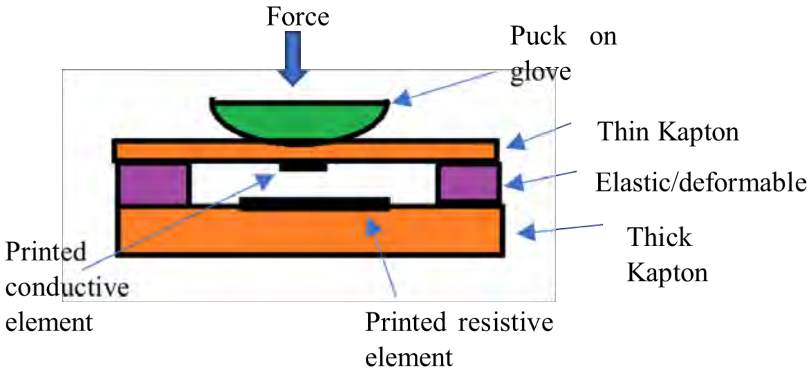


Figure 4.11: Alternate sensor design having puck and elastic layer

### 4.1.4 Stacked Sensor

The new sensor design eliminated the resistive track with a conductive wiper. This old design was troublesome because the wiping action between the two layers scored and scratched the resistive track creating inconsistencies in measurements or continuity errors disrupting any measurable resistive output. It followed a similar concept to the orange rail sensor, but without all the extra components except for the puck. This provided the same functional output.

The focus then moved towards utilizing a commercial linear soft potentiometer from Spectra Symbol (SP-L-0012-103-1%-RH). When a small puck was placed on the outer surface of the potentiometer, the sensor reacted when under shear force. It operated similar to the old design with a changing contact patch moving along the length of the sensor, but forewent the wiping action with its enclosed design.

One issue with the soft potentiometer was its ability to measure both shear and normal force. This created issues of trying to discern the signals to equate them to their shear and normal force components. To counteract this, a Flexiforce normal sensor (a201) was added in line with the soft potentiometer. This sensor reacted only to normal forces which allowed for the filtering out the shear component from the soft potentiometer. It ultimately ended up matching the expected output (when compared to the load cell in the insertion force fixture) when completing a connection of a median of 95% as well. This means that the normal force sensor can predict shear force applied in some scenarios due to the fluctuating normal force as shear is applied.

To complete the stacked sensor, an outer layer material to protect the sensors and provide grip was affixed. This material initially started as a 3M adhesive foam to provide exceptional short-term grip ensuring there is no slip when applying shear force. The layers of the stacked sensor can be seen below in Fig. 4.12.

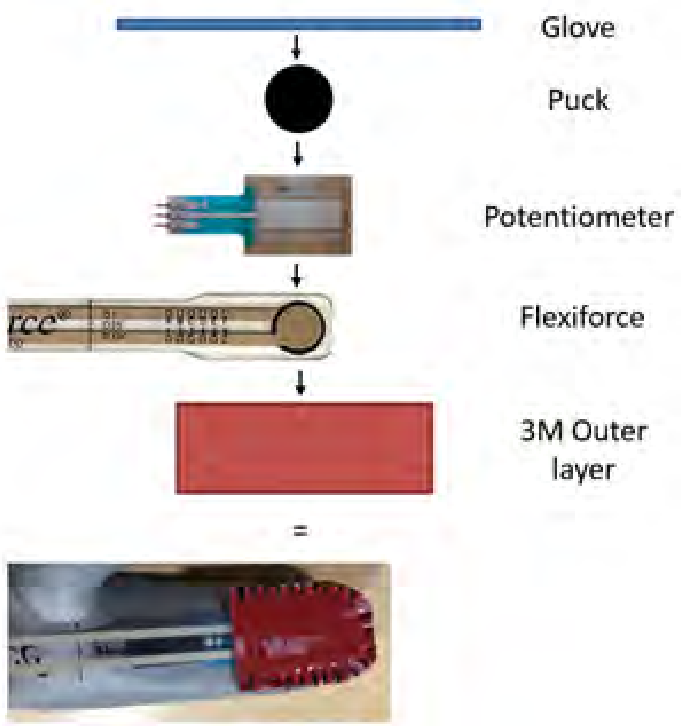


Figure 4.12: Prototype 1 stacked sensor layers

More images of the first prototype of the stacked sensor are shown below in Fig. 4.13. The sensor gloves were constructed by assembling the layers then using a thread and needle to sew the stack onto the glove. During assembly, all sensor leads and measurement areas were carefully avoided to ensure a proper reading.

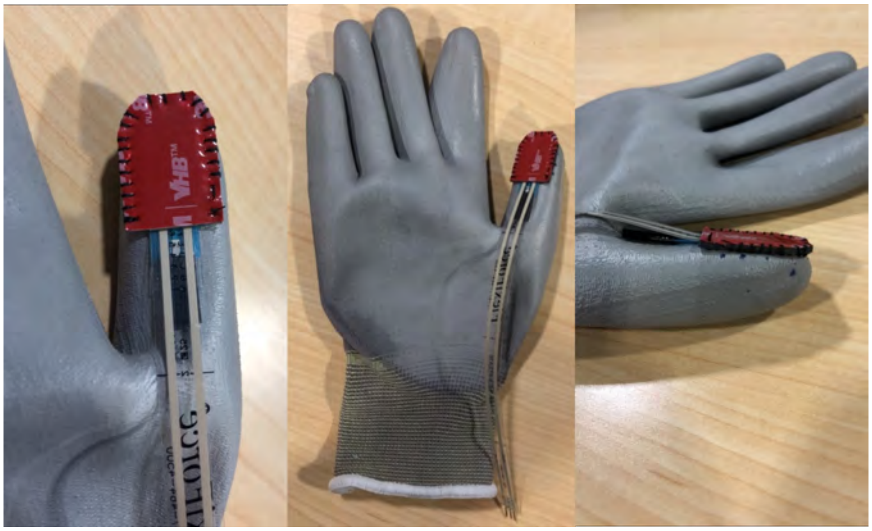


Figure 4.13: Prototype 1 of the stacked sensor affixed to the glove

The second stacked sensor prototype followed a similar format to the first one with a few substitutions and additions highlighted in green in Fig. 4.14. The Setex material provides enhanced grip for the associate when completing connections ensuring the measured shear force was accurate. The flexible plastic ring helps to protect the Setex material from tearing. The sensor terminals were also rerouted to the back of the hand using printed sensor leads.

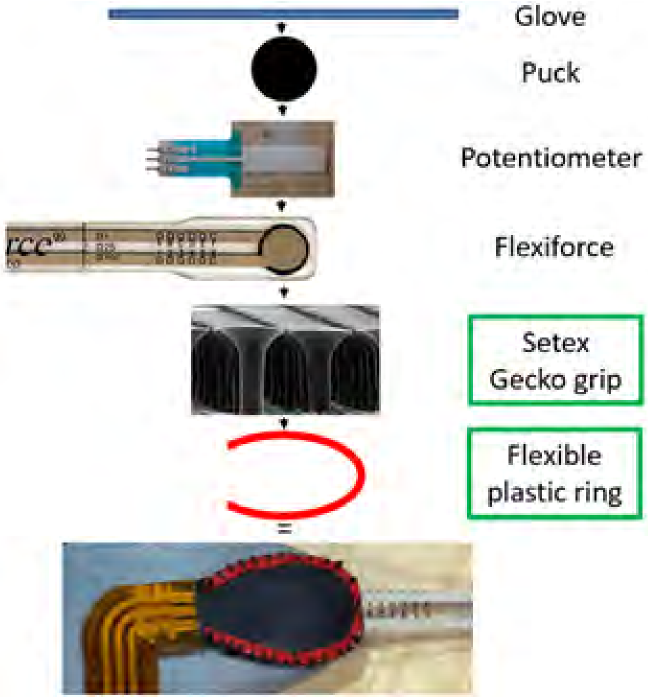


Figure 4.14: Prototype 2 stacked sensor layers

The second stacked sensor affixed to the glove is shown in Fig. 4.15. Assembly was the same as stated previously for Fig. 4.13. The thread went through the thin plastic ring to protect the soft rubber Setex material from tearing.

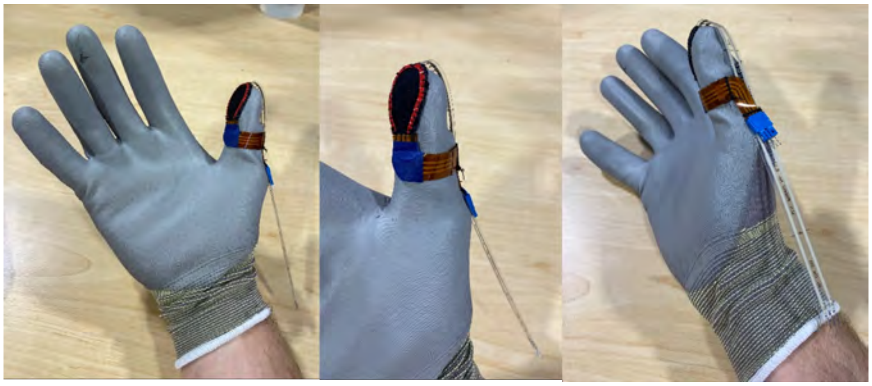


Figure 4.15: Stacked sensor glove prototype 2

The third iteration of the stacked sensor utilized the same Setex material and flexible ring, but it forewent the terminal rerouting to provide increased durability as the leads from the second prototype tore when tested on the assembly line. The normal force sensor is also angled so the lead has a reduced risk of snagging or bending. The glove sensor prototype is shown in Fig. 4.16.

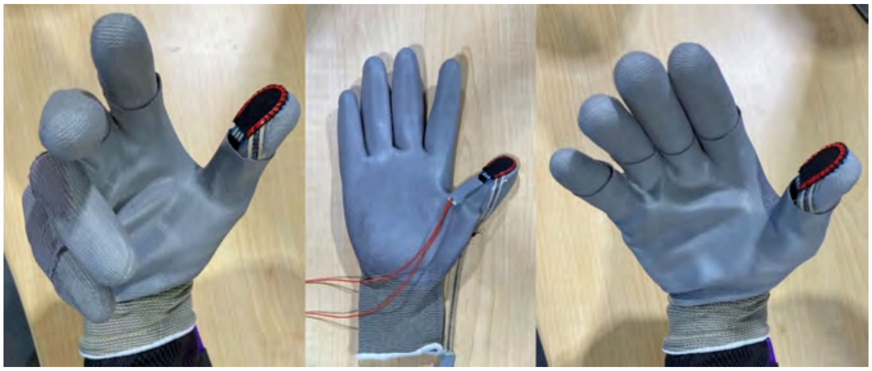


Figure 4.16: Stacked sensor glove prototype 3

## 4.2 Printed Sensor Development

The printed sensor was designed to operate similar to the commercial soft potentiometer, but it can measure shear force applied in any direction rather than being restricted to one axis. This is done by measuring the change in resistance from the original center to any point in the resistive circle to the outer silver trace. The silver trace has zero resistance and carries the signal back to the sensor terminal. As a greater shear force is applied, the contact patch will draw closer to the outer edge of the circle (and closer to the silver trace), decreasing the resistive output. The schematic for the printed sensor layers is shown in Fig. 4.17. The final printed sensor with the black being resistive ink and silver being silver, non-resistive, ink is shown below in Fig. 4.18.

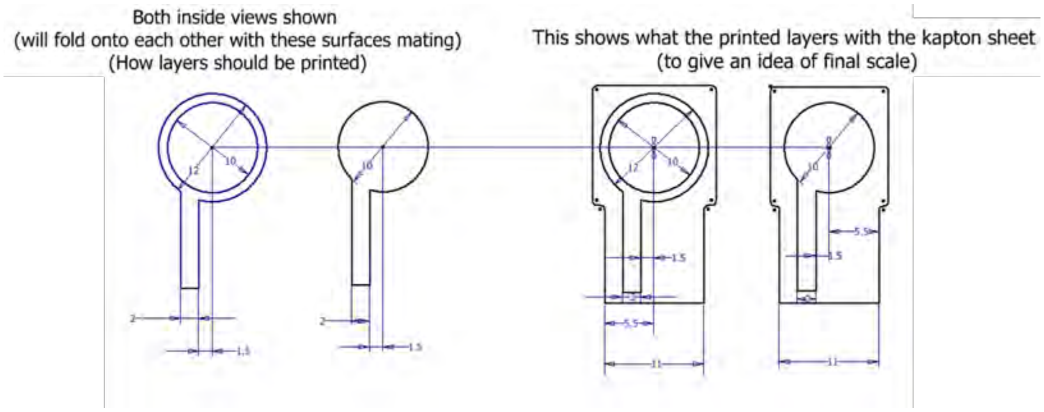

Both inside views shown
(will fold onto each other with these surfaces mating)
(How layers should be printed)
This shows what the printed layers with the kapton sheet
(to give an idea of final scale)
12
10
10
2
1.5
2
1.5
12
10
10
1.5
2
5.5
11
5.5
1.5
11

Figure 4.17: Printed sensor drawing with dimensions

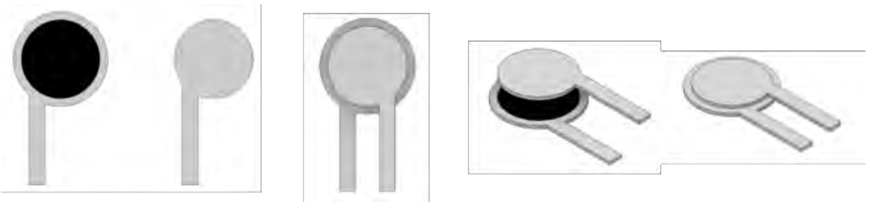


Figure 4.18: Printed sensor material layers

The first iteration for assembly of the printed sensor is shown in Fig. 4.19. It has a middle layer with a small thickness to provide spacing between the two layers, so there is no resistance reading when the two layers are not activated (not in contact). There is also a small cutout from the middle layer to allow for displacement of the air inside when under load.

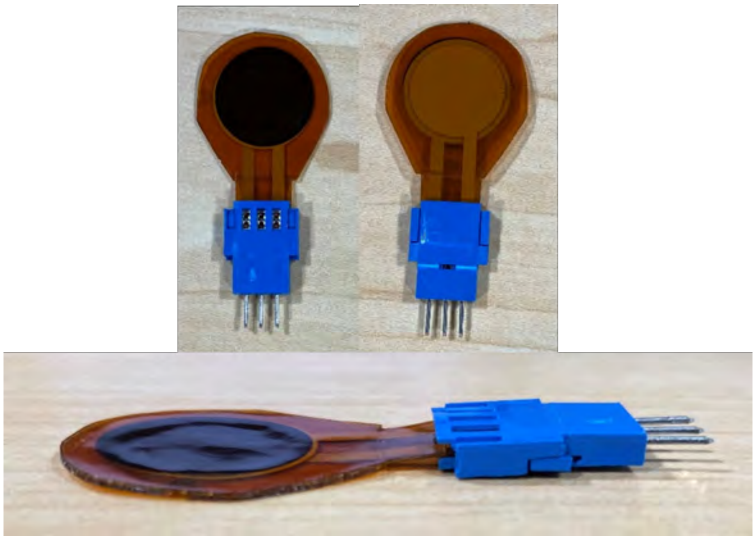


Figure 4.19: Printed sensor prototype 1

The second iteration of the printed sensor utilized a thicker middle layer to provide a separation between the two layers, increasing the probability of the materials separating when not under a load. The outer layer materials were also formed into a thicker substrate by using clear tape to also increase spring-back when not under a load. The second prototype is shown in Fig. 4.20. While the separation problems were solved, the resistance reading when under a shear load was not what was expected. The output from the sensor remained relatively constant. This was likely due to the contact patch between the two layers being relatively large and the low measurement range of the resistive ink.

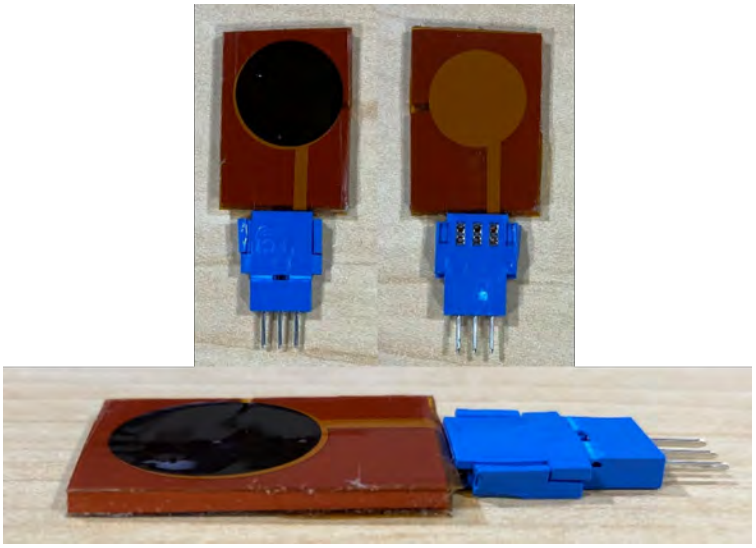


Figure 4.20: Printed sensor prototype 2

For the third printed sensor prototype, another layer of the resistive ink was printed onto the existing sensors. This can be seen in Fig. 4.21. This was done because upon further investigation of commercial sensors, the resistive ink is printed on top of the silver ink rather than underneath.

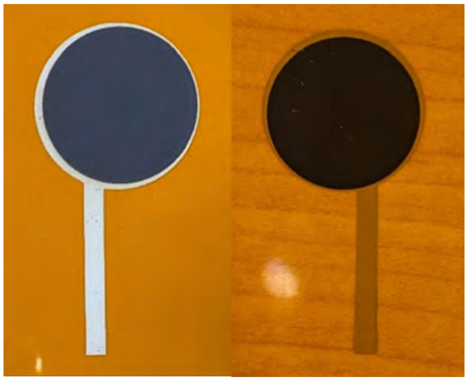


Figure 4.21: Printed resistive half top (left) and bottom (right)

The printed sensors were tested for measurement range as shown in Fig. 4.22. The center reading output has a higher resistance than the outer edges, which is expected. As the sensor is under a shear load, the contact patch will reach closer to the outer edges of the circle decreasing the resistance measurement, allowing shear to be measure in any direction. However, even though the resistive ink used show initial promise, it does not have a sufficient measurement range to give an adequate reading that could be equated to force as shear loading changes. This could be corrected with a higher resistive ink.

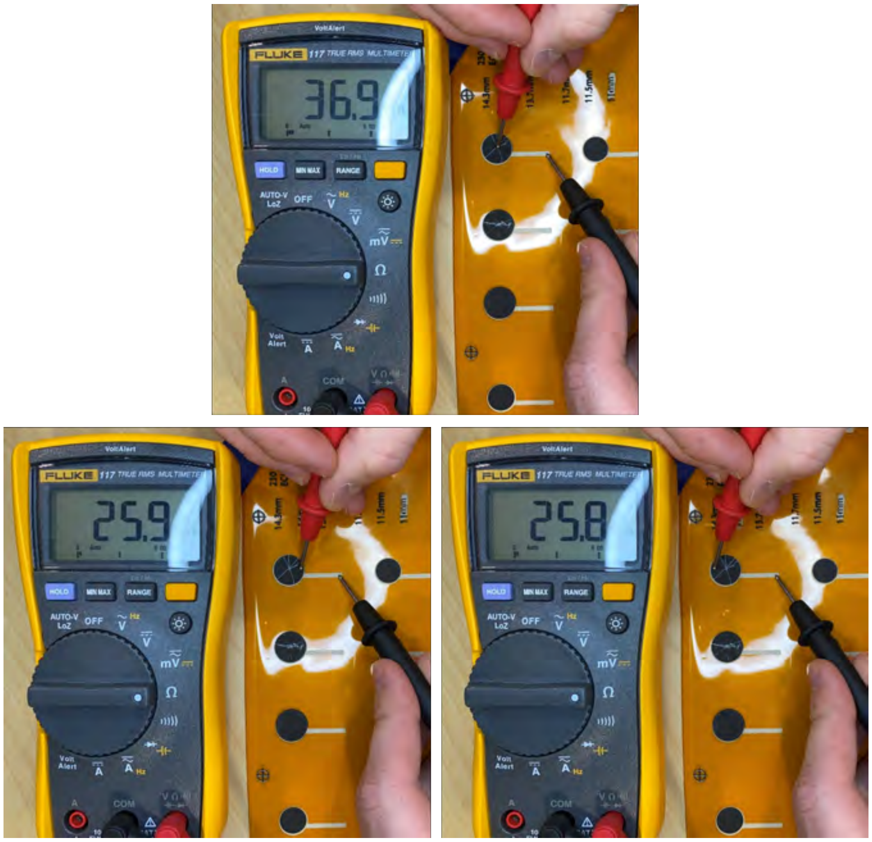


Figure 4.22: Resistive working range for printed sensor: center reading under normal force alone (top), edge reading under shear load simulating push (bottom left), and edge reading under shear load simulating pull (bottom right)

Sensors were assembled with the new double layer prints. The sensors shown in Fig. 4.23 were assembled the same way with a thicker outer substrate and middle layer, but two sizes were created. The diameters of the overlap between the silver top layer and resistive bottom layer are 11 mm and 15 mm for the two separate sensors. The sensors reacted as expected, but with the current geometry and materials used, they did not perform well enough for glove testing. Modifications need to be made to the sensor which may include, but are not limited to, utilizing a higher resistive ink, controlling the contact patch between the two layers, utilizing a sealed enclosure with air or dielectric grease within, and changing the geometries of the middle layer before affixing the sensor to the glove. This is included in the future work of the printed sensor

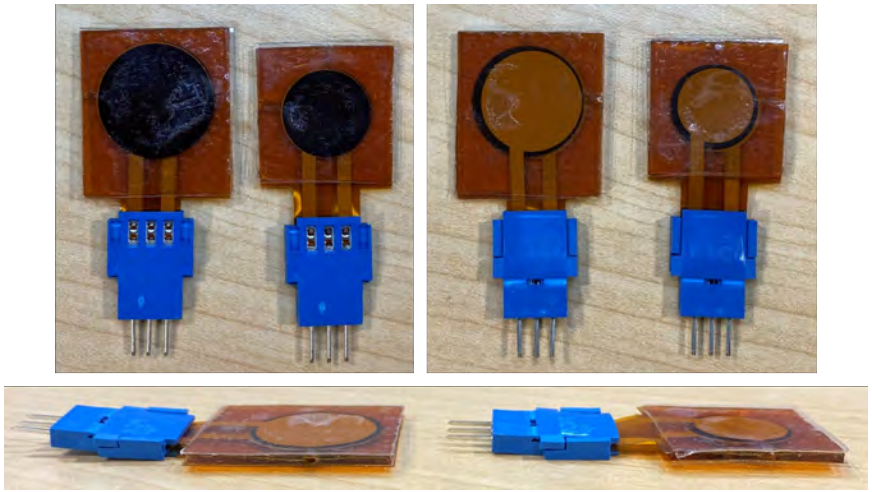


Figure 4.23: Printed sensor prototype 3 (left in pictures) 15 mm diameter and sensor prototype 4 (right in pictures) 11 mm diameter

An example geometry that will control the contact patch and still allow for multidirectional measurement is shown in Fig. 4.24. The sensor currently used a 3D layer with adhesives that borders the outer black resistive ring. The proposed new design will utilize the same border middle layer will an additional geometry overlayed on the resistive ink. In the figure, this proposed geometry is highlighted in blue. This example will allow for measurement along 3 axes on a horizontal plane as show by the green arrows. This will also reduce the contact patch since there will be less overhang between the two sensor layers to lay on top of each other.

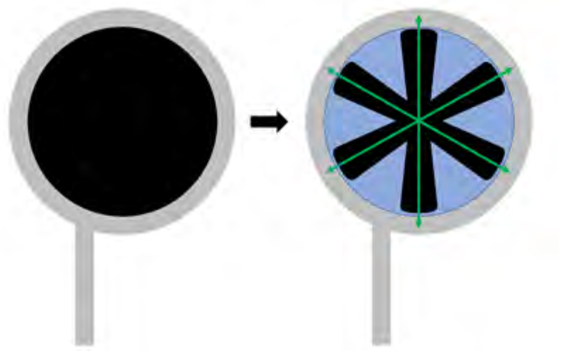


Figure 4.24: Middle layer adjustments to control contact patch between two layers and allow for multidirectional measurement: Before on the left, adjusted on the right

## 4.3 Wrist Unit Development

Once a working glove prototype was developed, efforts were put towards preparing for testing on the assembly line. To accomplish this, a new sensor glove was made along with a control module to collect and save data.

The control module is a circuit comprised of a Teensy 3.2 board, an accelerometer, and a Precision RTC. The Teensy board is how the system connects to an outside computer for data collection and saving. This is also where power is drawn to excite the other components in the control module. Data is fed from the shear and normal sensors to the analog inputs of the Teensy board.

The accelerometer provides both acceleration data in x, y, and z and gyroscopic data around those axes. This can be used as a reinforcement for determining when a connection is completed so the real-time feedback can be supplied to the associate.

The Precision RTC is a real-time clock that is used to record a timestamp with each data collection point so it can be matched with the data labeling system used on the iPad. The data labeling system was our method of indicating which type of connection was completed at which time. We later parsed through the data and added the respective label to each dataset. The entire module is shown in Fig. 4.25.

Overall, the control module coupled with the sensor glove recorded voltage differences, which were equated to resistance changes from the shear and normal force sensors, acceleration in x, y, and z, gyro readings in x, y, and z, and the date and time the measurement took place.

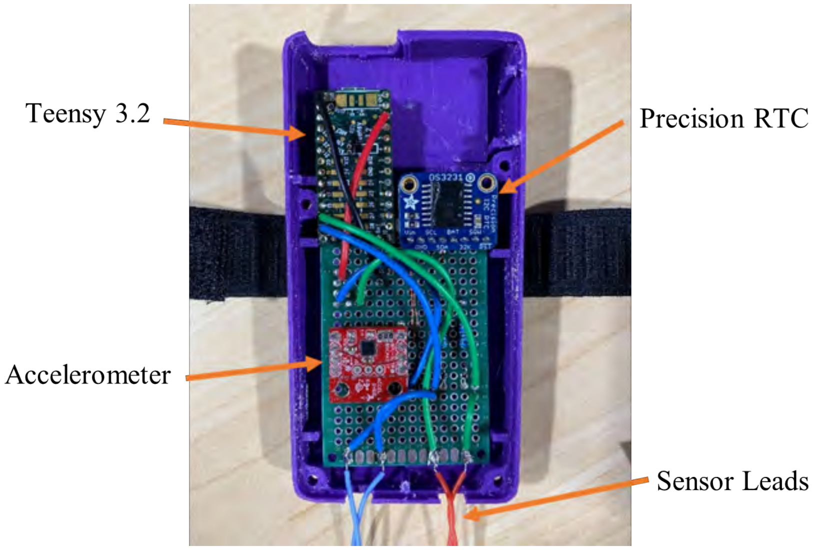


Figure 4.25: Sensor glove control module for data collection and saving

The control module was designed to fit into a forearm sleeve like the existing BMW thumb unit sensor system. The complete setup which was worn by the associate during line testing is shown in Fig. 4.26.

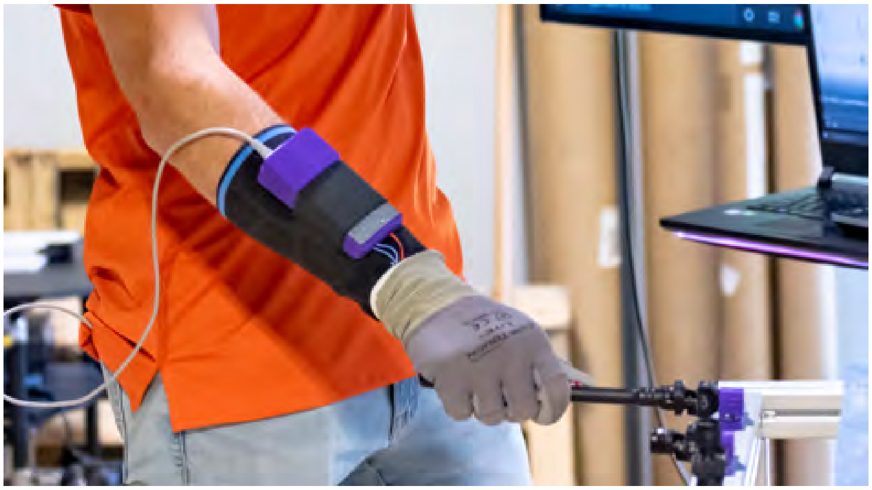


Figure 4.26: Complete sensor system used for line testing with sensor glove

After the first round of testing, adjustments detailed below were made to the hardware and software to create a better experience for the associate and create an easier process for the data analysis. A new sensor glove prototype was made, making this the third prototype. This adjusted positioning of the sensors and utilized the outer layer material. The sensor control module also received some upgrades, utilizing an amplifying circuit to achieve a greater measurement range for both sensors, but for the normal force sensor in particular. The previous working range for the normal force sensor was rather small, being approximately 75-88 analog points. The new internals for the control module are shown below in Fig. 4.27. The third iteration of the sensor glove is reiterated below in Fig. 4.28.

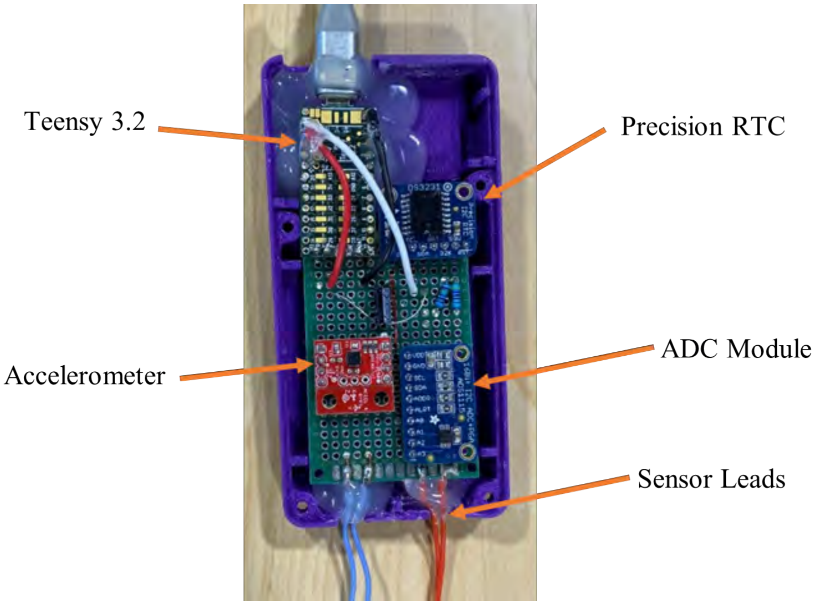


Figure 4.27: Second iteration of sensor glove control module

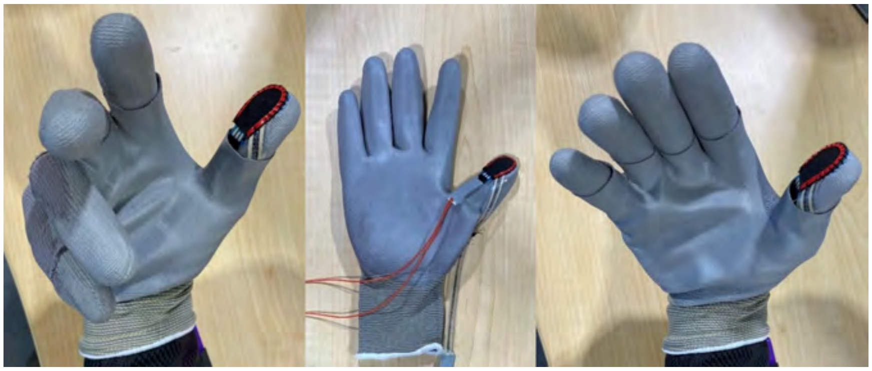


Figure 4.28: Stacked sensor glove prototype 3

The working principle of the sensor glove system is shown in Fig. 4.29. This indicates the delivery of the circuit components to the Teensy, which then flows into a student computer. As noted previously, the data is currently saved on a laptop while collecting data on the assembly line.

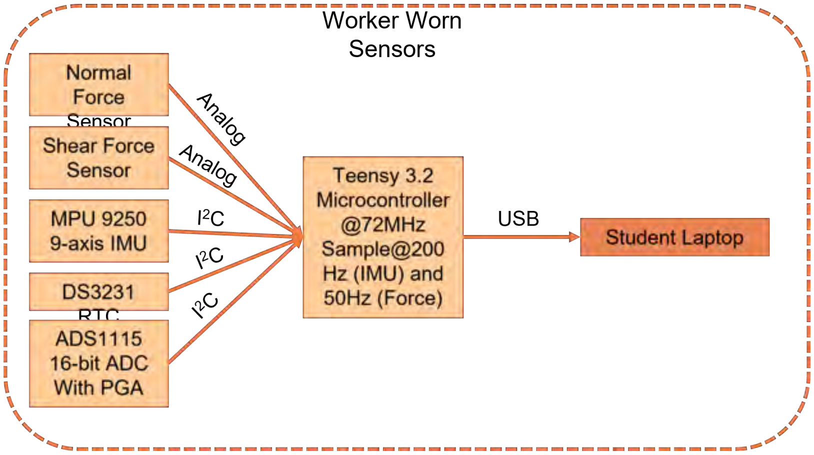


Figure 4.29: Current sensor system circuit operating principle

The next steps for the sensor system will create an independent system that can collect and analyze data without augmentation from a student. Further details of the next steps can be found in the Future Work section. The goal for the next operating principle is shown in Fig. 4.30.

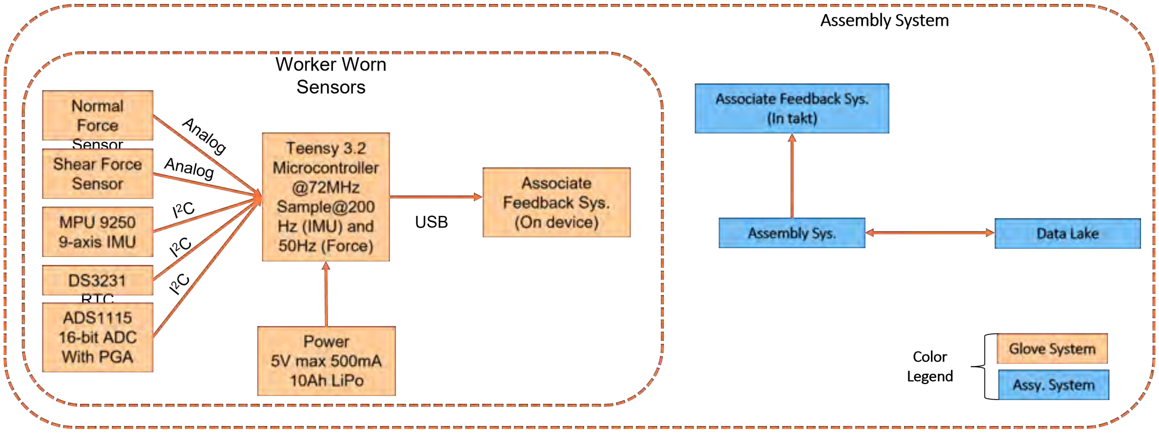

Assembly System
Worker Worn Sensors
Normal Force Sensor
Shear Force Sensor
MPU 9250 9-axis IMU
DS3231 RTC
ADS1115 16-bit ADC With PGA
Analog
Analog
I2C
I2C
I2C
Teensy 3.2 Microcontroller @72MHz Sample@200 Hz (IMU) and 50Hz (Force)
USB
Associate Feedback Sys. (On device)
Power 5V max 500mA 10Ah LiPo
Associate Feedback Sys. (In takt)
Assembly Sys.
Data Lake
Color Legend
Glove System
Assy. System

Figure 4.30: Future sensor system circuit operating principle with internal processing and power

The wiring diagram for the second control module utilized in assembly line testing with the amplifying circuit is shown below in Fig. 4.31.

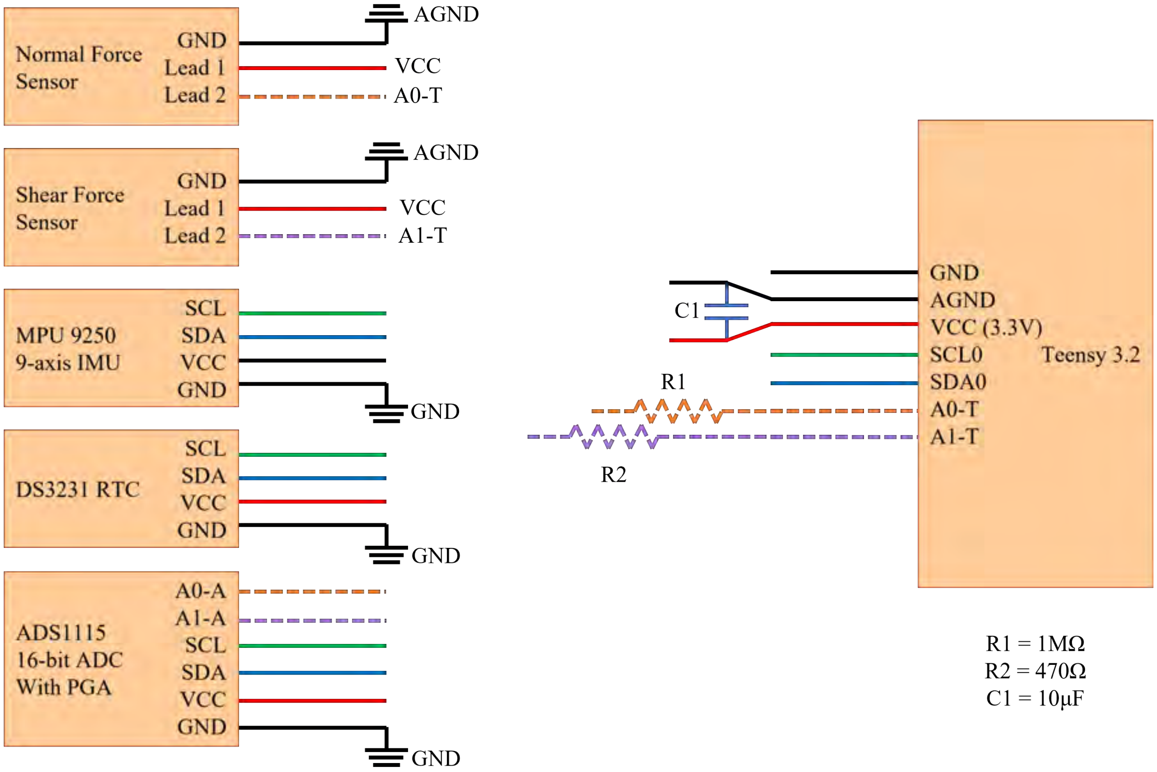


Figure 4.31: Wiring diagram for the second iteration of the sensor glove circuit utilized during lab and assembly line testing

# Chapter 5

# Sensor Glove Calibration

## 5.1 Calibration

The sensor glove system utilizes a changing voltage output via a variable resistance under loading for the objective determination of connection quality. These changing resistances can capture shear and normal forces applied by an associate. Similar marketed standalone sensors are coupled with a calibration document to equate a voltage output to a given force. Similar tests were conducted with the developed sensor glove discussed in the paper. However, there is a fundamental issue preventing the direct correlation of a voltage output to force. This being the capturing of both shear and normal forces simultaneously. To try and overcome these challenges, three separate calibration methods were performed. We will refer to the methods as direct calibration, subtractive calibration, and fixture isolating calibration.

### 5.1.1 Direct calibration

The first method was a direct calibration. This involved taking the sensor system and applying known forces in the shear and normal directions. This was completed in the glove format on a person and with an isolated sensor.

Tests began with attempting to calibrate normal forces first, then shear. Normal forces were applied to a stacked sensor that was placed on a table. This was done by applying a series of known weights to the puck on the stacked sensor. Tests were also conducted in the glove format using the

Mark-10 load cell in the insertion force fixture. The person simply pushed against the load cell until a desired normal force was achieved. Once voltage outputs were equated to known normal force values, efforts moved towards doing the same for shear forces. However, to apply a shear force, a normal force must also be applied to mitigate slip. Further discussions of shear calibration in the glove format are in section 5.1.2 Subtractive Calibration.

The isolated sensor was placed in the shear force fixture for calibration. The shear force fixture could apply known shear forces, but the applied normal force were inconsistent due to the loading mechanism. The top plate of the shear force fixture, highlighted in orange in Fig. 5.1, sat on top of the puck and guide rails to direct the movement. These guide rails created friction which influenced the applied shear force. Without the guide rails, the plate was unstable on only the puck, so the normal forces were inconsistent.

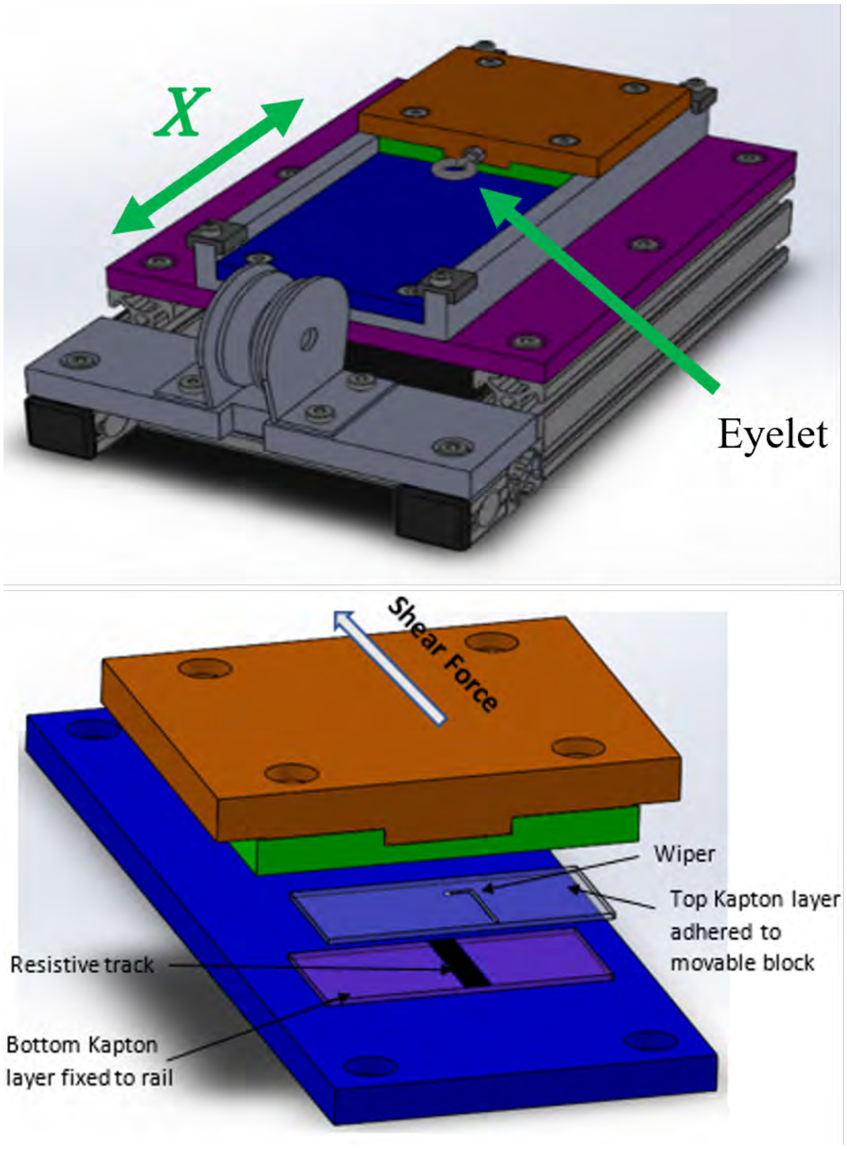

X
Eyelet
Shear Force
Wiper
Top Kapton layer adhered to movable block
Resistive track
Bottom Kapton layer fixed to rail

Figure 5.1: Shear force fixture

A major roadblock preventing the calibration of the shear force sensor was the working principle of said sensor. The sensor is a resistive track. As a contact patch moved up or down the resistive track under a shear force, the voltage output increased or decreased. The puck was put in place to focus force applications onto the centers of each sensor, but this did not eliminate fluctuation. The shear sensor output under loading was heavily dependent on the starting position of the person (the starting contact patch). Once initial contact was made, there was an observable voltage change as shear was applied.

### 5.1.2 Subtractive Calibration

We have already discussed how the shear sensor under a normal loading can fluctuate based on initial contact patch, rendering the voltage output inequitable to an applied normal force (repeatably). As a result, the normal force sensor was placed in line with the shear force sensor. This was originally done to become determine the shear component of the combined loading, but was later kept as a form of multi-modal sensing for increased confidence.

The idea was to calibrate the commercial, off-the-shelf, normal force sensor to normal loadings. This calibration data would then be used to "subtract" the normal component from the shear sensor to capture the shear component. This practice faltered due to the normal force sensor reacting to shear force as well. Therefore, the individual components of the combined loading could not be determined through the stacked sensor system alone.

### 5.1.3 Fixture Isolating Calibration

The most recent effort towards calibration the stacked sensor system utilized the shear-normal fixture. The research consisted of three experiment types: repeated normal loading (Normal LU), repeated shear loading (Shear LU), and incremental shear loading (Shear Inc). The repeated normal loading was a series of loading and unloading an applied varying normal force to the sensor. The repeated shear loading utilized a constant normal force with the loading and unloading of a varying shear loading. The incremental shear loading maintained a constant normal force and applied additional shear force without unloading of the previous force. This test was run until failure (*i.e.* the roller bearing slide overcame normal force from applied shear and slipped out from underneath normal loading).

All forces were exerted through weight which were equated into forces. Shear loading was applied in increments of 0.1 kg or 0.981 N. A breakdown of all 70 experiments are shown in Table 5.1. Each experiment was repeated 5 times.

Table 5.1: Shear and normal force fixture experiment breakdown

| Experiment | Applied Normal (N) | Applied Shear (N) |
|---|---|---|
| Normal LU | 9.81 | 0 |
| | 11.77 | 0 |
| | 14.72 | 0 |
| | 16.68 | 0 |
| | 19.62 | 0 |
| | 21.58 | 0 |
| | 24.53 | 0 |
| | 26.49 | 0 |
| | 29.43 | 0 |
| Shear LU | 9.81 | 0 - 4.91 |
| | 14.72 | 0 - 5.89 |
| | 19.62 | 0 - 7.85 |
| | 24.53 | 0 - 7.85 |
| Shear Inc | 24.53 | 0 - 6.87 |

Based on the behavior of the stacked sensor in the shear-normal fixture, we expect the stacked sensor in the glove format to follow a similar trend. This being both sensors responding under shear and normal forces exhibiting a linear response. The linearity in the sensor under separate normal and shear loading will create a more friendly sensor output for adapting to other processes and/or people. The normal force response in the shear sensor are detailed in Fig. 5.2 with error bars indicating repeatability.

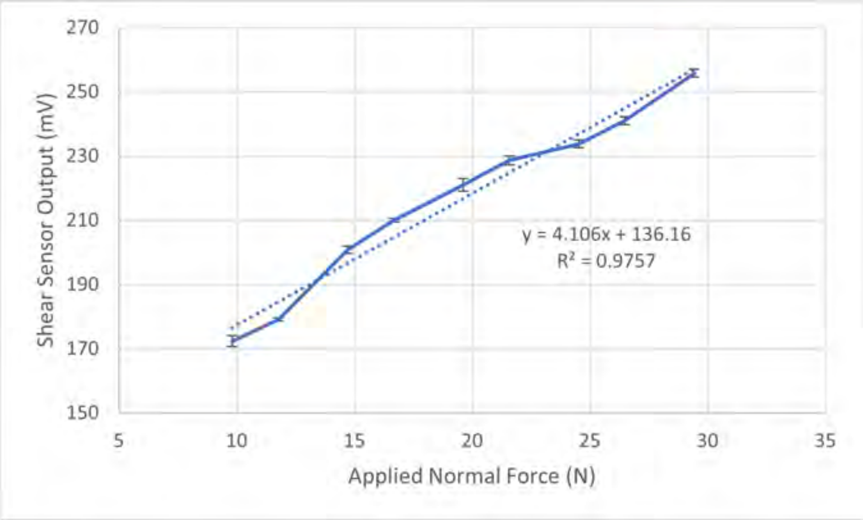


Figure 5.2: Shear sensor sensor output under normal loading a) sensor response and b) sensor repeatability

The graph shown in Fig. 5.3 gives the sensor outputs of various shear loading under varying normal forces (Shear LU data). Each line indicates a different applied constant normal force. Each data point on the line marks a different experimental shear loading. When the shear force overcomes the applied normal force, causing the roller bearing plate to slip from underneath the normal force application plate, the testing was stopped. All loadings exhibit a relatively linear trend.

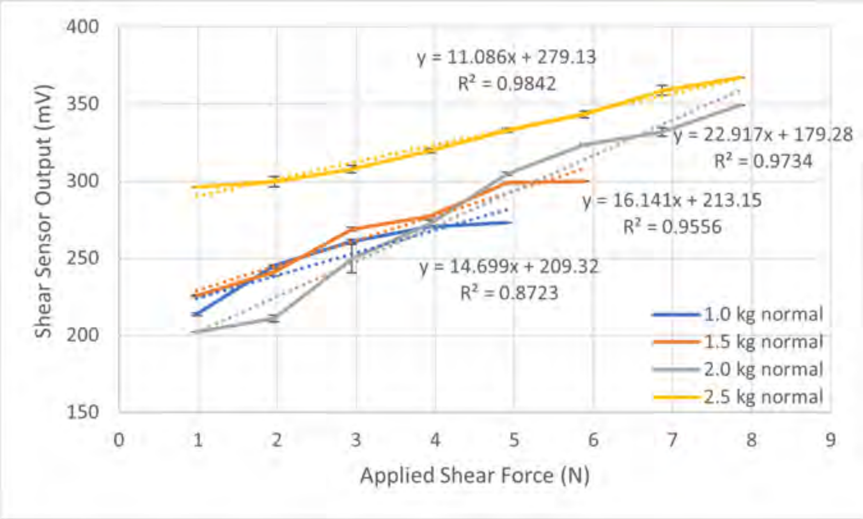


Figure 5.3: Shear sensor sensor output under combined shear and normal loading

As you can see in the graph, the sensor outputs were not always higher with a greater shear force or combination of shear and normal. This was again due to the nature of the resistive track in the shear sensor. The resistive track bottom layer made contact with the silver trace (approximately zero resistivity) top layer. As the contact patch moved along the track, the resistance, and therefore output voltage, changed. As a result, the force profiles exhibited in the glove format became the critical component rather than a force target or threshold alone to determine the quality of connections completed on the assembly line. Once activated, the sensor showed an increase in voltage output as shear force increased as shown in Fig. 5.4. The green lines indicate the steady-state point of each step in the incremental loading. The additional motion and acceleration sensors also helped to determine the operator approach increasing the confidence of feedback provided to the operator.

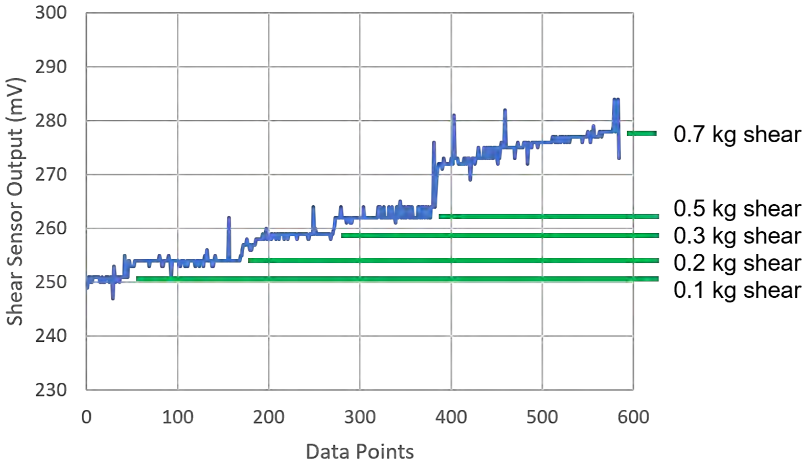


Figure 5.4: Shear sensor sensor output under a constant normal load of 2.5 kg and an incremental shear load (Shear Inc)

Despite calibration efforts with the fixture, the variability in the sensors under a combined loading were too high to accurately equate voltage outputs to force. Tests did however prove the linear relationships within each sensor response. The other sensor in the stacked sensor system was the normal force sensor, and the manufacturer claims a linear response under normal loading. This was verified as shown in the sensor output graph of Fig. 5.5.

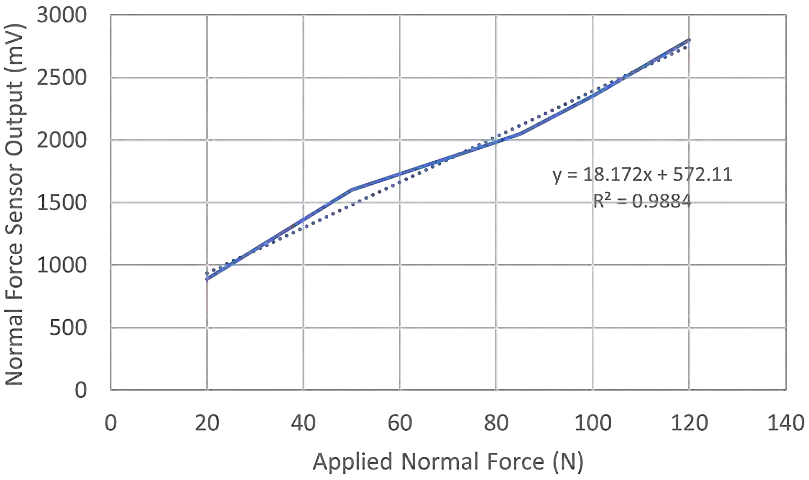


Figure 5.5: Normal force sensor linear signal output curve

As a result of the profound variability, the direct voltage output of the sensors was utilized for signal processing. The live output and the classifier utilized the voltage response.

## 5.2 Sensor Behavior Over Time

The sensor glove was tested hundreds of times in he lab and on the BMW assembly line. Tests in the lab were usually isolated to the connection alone. Tests conducted on the assembly line were worn for the duration of the BMW associate's work which included other processes within the takt, such as tool use and plug installation, and other non-work tasks, such as taking a drink or resting their hand.

Throughout testing at BMW, three sensor gloves were used. The first two sensor gloves were previous iterations that had functional or durability flaws. These flaws were amended, resulting in the third glove, and current, iteration for assembly line use. This sensor glove has been tested on the assembly line for 250+ vehicles. This includes 3+ investigated connections per vehicle and other accompanying tasks that make up the remainder of the takt time. Throughout testing, the sensor glove maintained functionality and durability in the physically demanding, harsh environment. There was no observable degradation in sensor performance and repeatability.

Repeatability tests were also conducted with the BMW connectors to investigate degradation in the material and structure which could result in lesser or greater forces being required to complete each connector type. No observable change was recorded over hundreds of tests. Changes in connection force were resultant of operator variability.

## 5.3 Sensor Working Range

The working range of a sensor denotes the range of concentration that can be adequately determined by the instrument. In simpler terms, it indicates the operating range the sensor is capable of measuring. The Flexiforce a201 normal sensor from Digi-key offers three measurement capabilities: 0-4.4 N (1 lb), 0-111 N (25 lb), and 0-445 N (100 lb). Measurements can exceed 445 N as this sensor is said to be capable of measuring up to 4,448 N (1,000 lb) if a lower drive voltage and lower resistance for the feedback resistor is used. For the intents and purposes of this research, the middle range, 0-111 N, sensor was selected. This allowed for a greater sensitivity in the normal operating range for the assembly line workers. In the Python code, this 0-111 N measurement range was equated to a 0-3300 mV working range.

The shear sensor working range has more variability. This soft potentiometer was adapted for measuring shear forces, and therefore, the manufacturer did not directly calibrate for this. As discussed above in the Section 5.1, the voltage output cannot directly be equated to force as the sensor is sensitive to shear and normal forces, and the output is dependent on the starting contact point. Therefore, the working range of the shear sensor was given as a voltage output and was driven by the Python code. Similar to the normal sensor, the measurement range of the sensor was set to 0-3300 mV in the code. This can be approximated to be a force equated working range with a minimum of 1-3 N to a maximum of 50-125 N.

# Chapter 6

# Data Collection

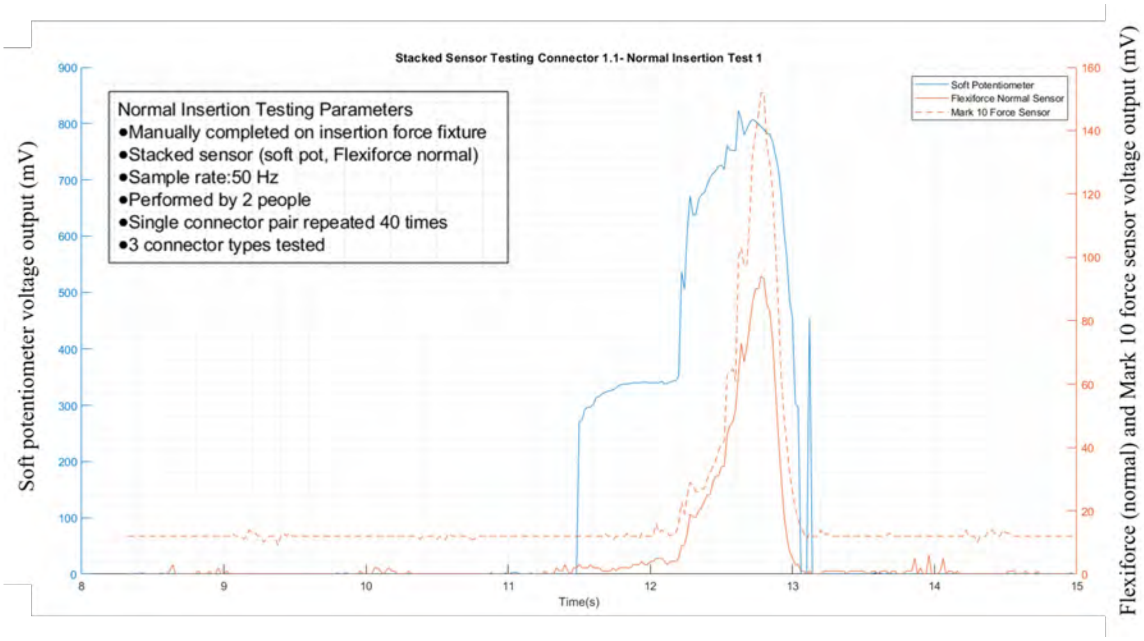
All data collected and discussed in the following sections utilize the stacked sensor in the sensing glove system. The stacked sensor was paired with the first wrist unit design iteration for lab collected data and the first round of assembly line testing with a BMW associate. Later tests utilized the second wrist unit iteration for a more dynamic measurement and increased working ranges for the sensors. Some figures discussed in this chapter are recaps of figures from previous section for reading convenience.

## 6.1 Lab Collected Data - BMW Connectors

Lab captured data was completed by 3 Clemson students who were part of the research project. Lab testing with the sensor glove took place over the course of 2.5 months (mid-February to April 2021). Two individuals were Master's students and a postdoctoral research fellow. All lab participants were part of the research group.

A representation of the assembly line was created in the lab to try and mimic movements, approaches, and grips of the BMW associates. Tests were completed on the insertion force fixture by multiple students to create a diverse data set. Tests were completed by mating the female end of the connector with the male end attached to the fixture.

Initial testing was completed with this stacked sensor glove on the insertion force fixture so the signals from the two sensors could be compared to the load cell output. An example output of a normal insertion is shown in Fig. 6.1. A normal insertion indicates that the initial push to complete the connection was made without the subsequent pull-push verification testing. Testing parameters are overlayed on the figure.



Normal Insertion Testing Parameters
- Manually completed on insertion force fixture
- Stacked sensor (soft pot, Flexiforce normal)
- Sample rate:50 Hz
- Performed by 2 people
- Single connector pair repeated 40 times
- 3 connector types tested

Figure 6.1: Normal insertion profile using the first prototype of the stacked sensor glove

Fig. 6.2 is an example plot of a push-pull-push test with the stacked sensor glove. Testing parameters are also overlayed on the figure. There is also a green dividing line indicating where the initial push stops and the verification pull-push begins. This is for the regression which is discussed after the figure.

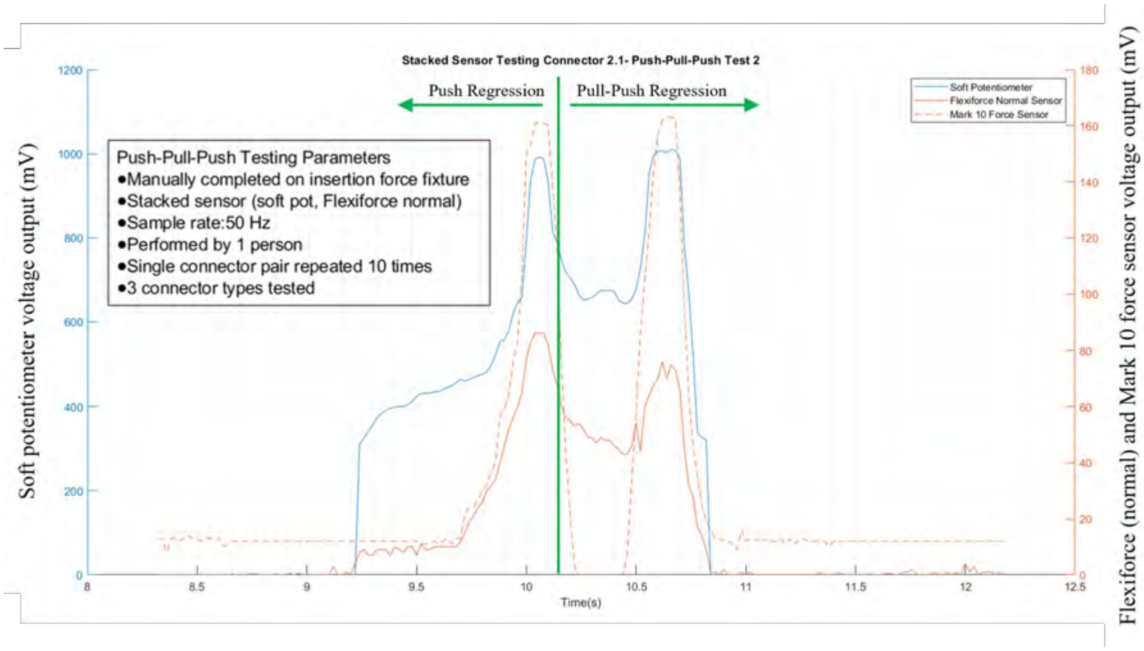


Figure 6.2: Push-pull-push profile using the first prototype of the stacked sensor glove

## 6.2 Regression Analysis - Sensors to Load Cell

The relationship of normal and shear force sensor with Mark 10 was quantified using regression analysis in Minitab. The R-squared value represents how well the normal force sensor and shear sensor follow the load cell output curve. This is an analysis for a single test to show how well the sensor outputs match the Mark 10 load cell as the connection is completed against the Mark 10. Comparisons between multiple tests in the lab, line, and comparing the two settings are discussed later in Section 7.1.

The normal force sensor shows a better correlation of fitting the load cell curve 90% or more for most tests. For the example output in Fig. 6.2, the normal curve matches the expected to 95.5%. However, the soft potentiometer shows greater promise of capturing the inflection for the verification testing.

Additionally, the lower value of Mallows Cp represents that the model is relatively precise (has small variance) in estimating the true regression coefficients and predicting future responses. Mallows Cp is a measure to pick the best regression model and can be calculated as shown in Eq. 6.1. $RSS_p$ is the residual sum of squares for a model with p predictor variables, $S^2$ is the residual mean square for the model (estimated by MSE), N is the sample size, and P is the number of predictor variables [89]. The desired Mallows Cp value is as close to the number of variables as possible (1 for sensors alone and 2 for both sensors).

$$Cp = RSS_p/S^2–N + 2(P + 1) \tag{6.1}$$

Utilizing two sensors gives greater confidence for all aspects of the connection completion and also confidence in the regression with the lower Mallows Cp value. The regression analysis for Fig. 6.2 can be seen below in Table 6.1.

Table 6.1: Normal insertion regression of shear and normal force sensors compared to the load cell output

| **Normal Insertion Regression** | | |
|---|---|---|
| *Variables* | *R-squared Adjusted* | *Mallows Cp* |
| Normal Sensor | 95.5 | 31.7 |
| Shear Sensor | 51.7 | 2010.1 |
| Normal and Shear Sensors | 96.1 | 3.0 |

The initial push exemplified the same regression trend as the normal insertion, but the following pull-push did not. Since we are examining inflection for confirming the verification test was complete, the poor correlation is allowable. These numbers are reflected in Table 6.2. This regression analysis further justifies the use of multimodal sensing. The normal force sensor is better at capturing the response during the initial push, the locking phase of the connection. The shear sensor is better at capturing the push-pull-push profile to ensure validation checks are completed. These coupled with motion sensors can be used to determine operator approach and forces applied to ensure success of the connection.

Table 6.2: Push and push-pull-push regression of shear and normal force sensors compared to the load cell output

| **Push Regression** | | |
|---|---|---|
| *Variables* | *R-squared Adjusted* | *Mallows Cp* |
| Normal Sensor | 95.6 | 57.5 |
| Shear Sensor | 61.0 | 716.3 |
| Normal and Shear Sensors | 92.9 | 3.0 |
| **Push-Pull-Push Regression** | | |
| *Variables* | *R-squared Adjusted* | *Mallows Cp* |
| Normal Sensor | 46.9 | 2.4 |
| Shear Sensor | 44.4 | 7.0 |
| Normal and Shear Sensors | 47.1 | 3.0 |

## 6.3 BMW Assembly Line Collected Data

Assembly line captured data was completed by 7 BMW associates. Assembly line testing with the sensor glove took place over the course of 16 days in March 2021 and one additional visit in both April and May of 2021. Three days of assembly line tests took place in March and one in both April and May. Associates were selected based on their willingness to be a part of the study and their role at BMW. Associates who normally complete the investigated connector types were selected. All participants on the assembly line were male. Sensor glove iteration 1, 2, and 3 were tested on approximately 66, 15, and 250+ vehicles respectively. This included processes outside of the investigated connections. Two associates took place in the study twice, and the remainder just once. The sensor glove and wrist unit forearm sleeve were disinfected with Clorox wipes between use. The testing conducted in the month of April on the BMW assembly line was not utilized in any data analysis as both force sensors were disconnected from the circuit. This issue was quickly remedied for future testing. This indicates the need for increased durability of the sensor glove system before final deployment on the assembly. All studies conducted with participants outside of the research group were operating under IRB2018-114.

### 6.3.1 Testing: Industrial First Phase

The new sensor glove, prototype 2, as mentioned previously in the section 4.1.4 Stacked Sensor, rerouted the sensor terminals and utilized an outside layer with better gripping strength. Prototype 2 of the sensor glove is shown below in Fig. 6.3. This was coupled with the first rendition of the wrist control unit.

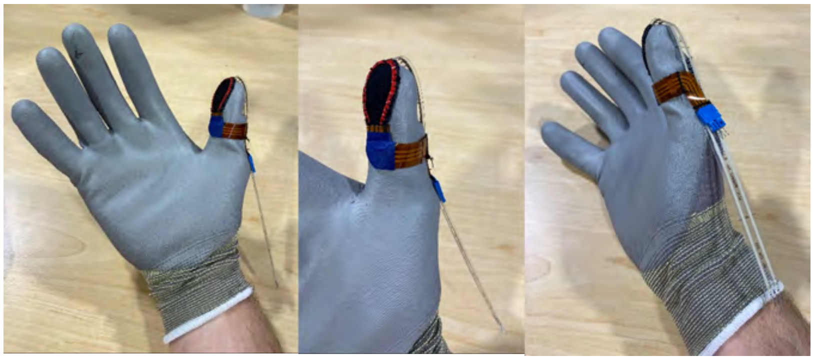


Figure 6.3: Stacked sensor prototype 2 recap

On the assembly line, the team worked with the BMW associates to coordinate the data collection to occur when the investigated connections were being completed. The control module Teensy board was connected to a laptop. A custom program on the laptop designated when to start and stop data collection. Another tablet device was used by another member of the Clemson team to label the data as it was being collected on the laptop. This utilized another custom program to record the timestamp. The data collection through the control module also had an associated timestamp, so these values were matched up later to determine which type of connection was being completed for each data recording.

The associates were asked to complete the work as they normally would so the glove data would be able to represent a normal work day. Therefore, there was no training involved. The takt time for data collection was slightly longer than normal due to communication time between the BMW associate and Clemson team, but the connections were completed in the duration expected on any given day.

Unfortunately, the sensor leads snagged during testing, ripping the leads from the soft potentiometer (shear sensor). Therefore, the sensor glove prototype 1 was used in place with the control module to continue line testing. An example output from the sensing system from line testing with the prototype 1 sensing glove is shown in Fig. 6.4.

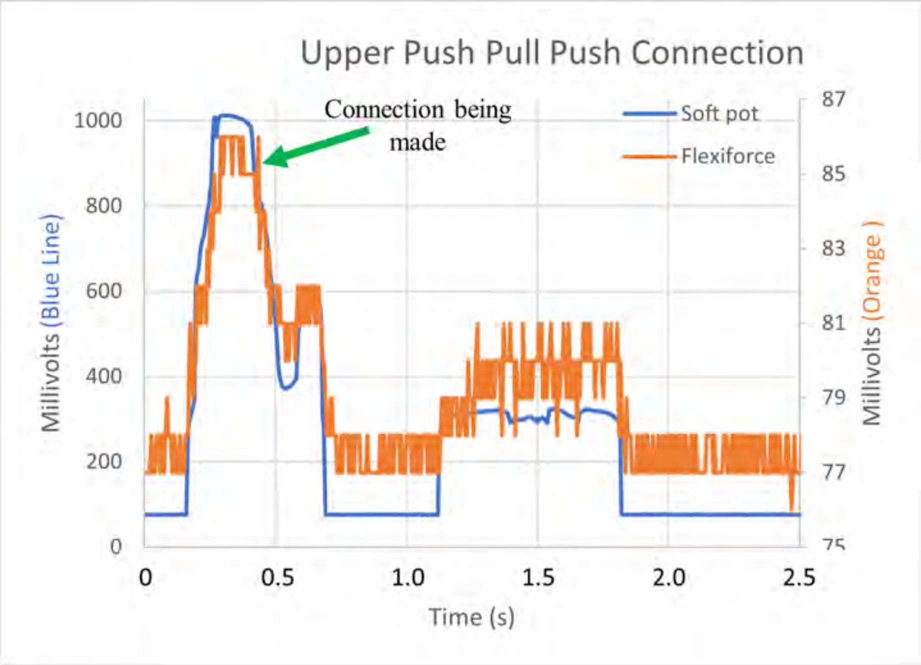


Figure 6.4: Line testing data collection example (Type 1 connector)

Once the data were collected and labeled, each set was segmented. The segmentation was conducted to separate the push-pull-push from the rest of the data. We wanted to investigate the push-pull-push alone to compare tests, classify, and predict connection quality before adding in noise and outside movements into the data analysis. This segmentation was completed by plotting all datasets in MATLAB and looking for the inflection of the push-pull-push in the graph. Since only short spans of data were collected at a time, the inflection was noticeable for the majority of the line data. The sets that were questionable or indeterminable were kept, but not utilized in the data analysis.

Below in Fig. 6.5 is the segmented plot from the data collection shown in Fig. 6.4. This shows where the push-pull-push is completed during testing.

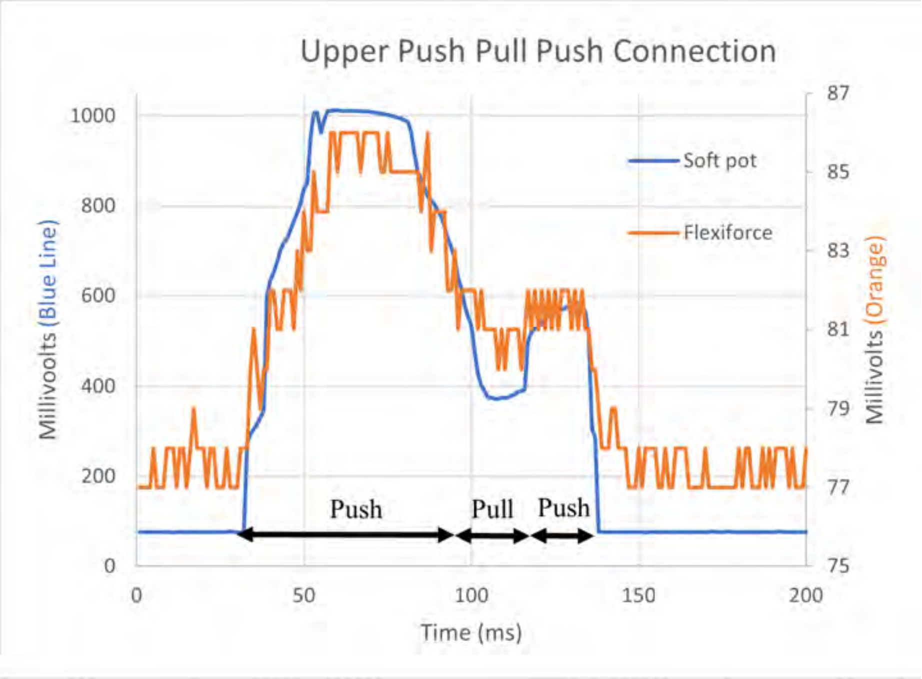


Figure 6.5: Segmented line testing data collection example (Type 1 connector)

### 6.3.2 First Phase Testing Lessons

After the first round of testing the sensor glove on the assembly line, adjustments to the stacked sensor hardware were needed to provide additional durability and ease of use for the associate as the second prototype leads tore, rendering the glove unable to collect the data we needed. Efforts toward simplifying the data collection and labeling would also be beneficial as there was an intermediate step needed to understand the meaning of the BMW terminology.

The team was able to collect useful data exemplifying completed connections that could be used for data classification and modeling of the connection quality. It was also determined that the response from the normal force sensor could use amplification to give a wider measurement range. This would allow for the associate activation to have a greater sensitivity and increase the confidence of the connection status based on the force input.

### 6.3.3 Testing: Industrial Second Phase

As stated in the Design of the Sensor Glove chapter, adjustments were made to the hardware and software after the first round of testing. The main differences were the adjusted positioning of the sensors, the outer layer material, and the circuit amplification which increased measurement ranges.

Below are the successful connections using the new sensor glove and control module. Fig. 6.6 and Fig. 6.7 show the full and segmented output respectively.

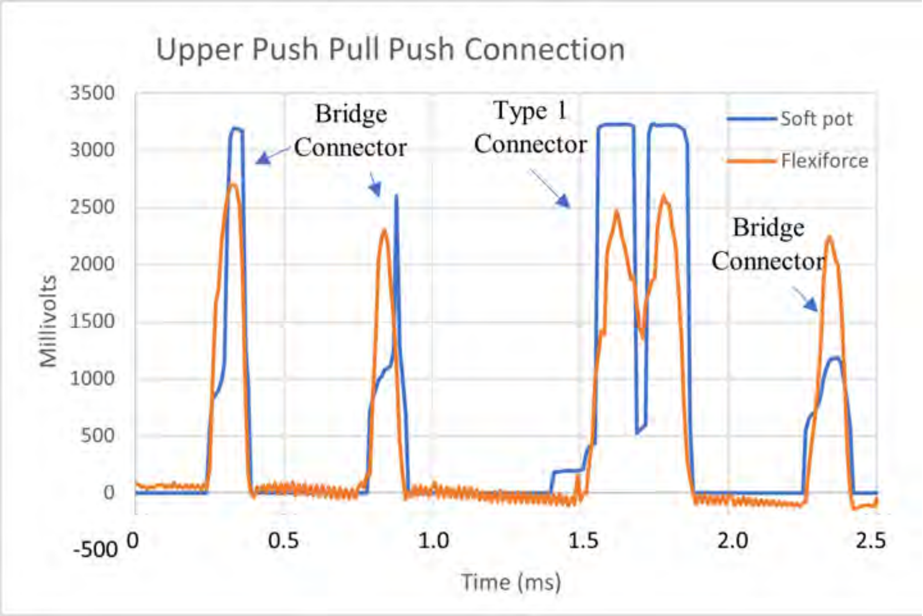


Figure 6.6: Phase two - line testing completed connection example (Type 1 connector)

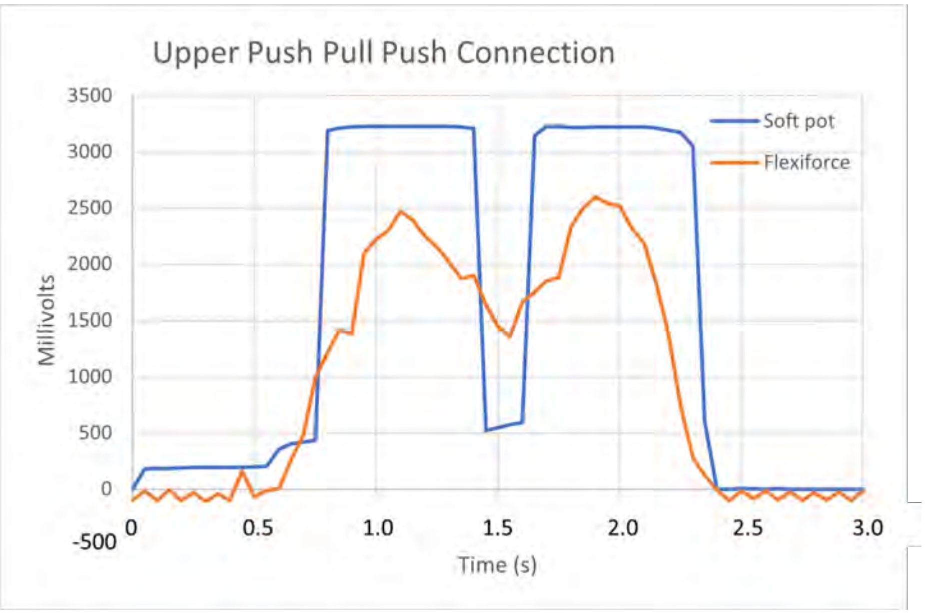


Figure 6.7: Segmented phase two - line testing completed connection example (Type 1 connector)

### 6.3.4 Second Phase Testing - Purposeful Failure

The second round of testing was focused on failed connections. We investigated with the BMW team some of the failure modes detailed on the PFMEA spreadsheet. The two investigated were misplaced force and low force. Both of these result in an incomplete connection. Misplaced force may see the force profile we expect, but since it is not applied on the connector, we can use motion data to determine if the connection was actually completed.

Example plots of low force and misplaced force are shown in Fig. 6.8 and Fig. 6.9 respectively. It was observed that associates were more inclined to not complete the push-pull-push verification testing when applying a misplaced force.

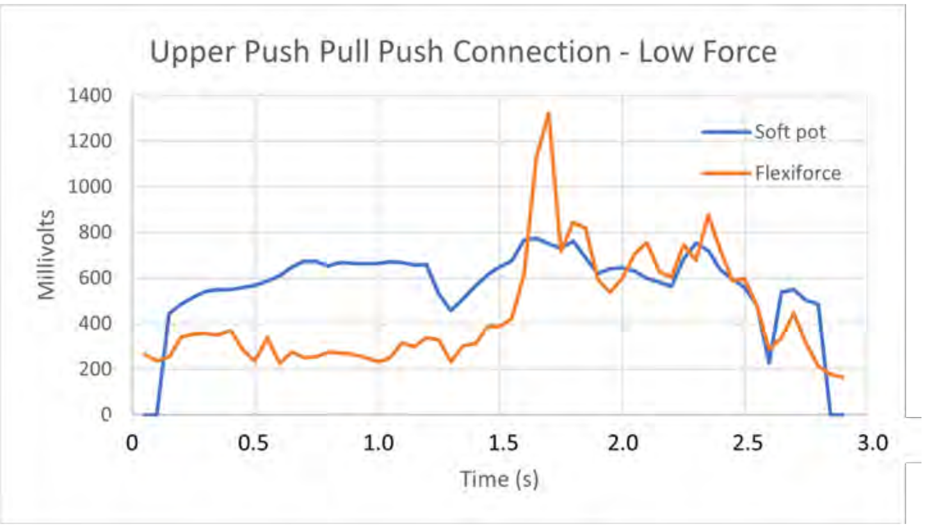


Figure 6.8: Phase two – purposeful failure, line testing data collection low force example (Type 1 connector)

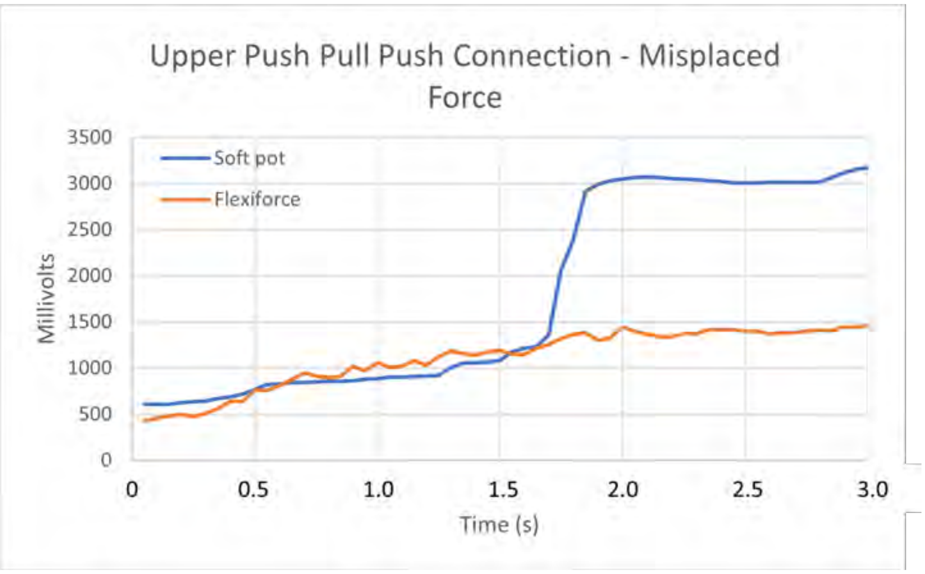


Figure 6.9: Phase two - segmented line testing data collection example (Type 1 connector)

### 6.3.5 Second Phase Testing Lessons

The second round of line testing showed a greater response from the sensors with the new control module than the first iteration. This is due to the amplifying circuit. This was especially noticeable for the normal force sensor as the analog measurement range went from approximately 75-88 to 0-3800 with a greater response from the sensor itself.

Feedback from the associates determined that the comfort of the sensor glove setup could be improved. Associates were asked questions regarding usability and comfort of the device such as: ”Are there any restrictions to motion?”, ”Did you experience any levels of discomfort during testing?”, ”Do you believe this device would interrupt your workflow?”, and ”Do you have any improvement suggestions?”. One associate noted that the forearm sleeve was itchy after extended use. Associates did not directly mention that any component, in the sensor glove’s current state, would interrupt the process other than the wired component to the student laptop. This component will be eliminated before unmonitored deployment on the assembly line. Another element the associates did not mention directly but needs to be improved is the terminal routing. Previous efforts in re-routing the terminals were successful, but they did not maintain the durability of the commercial terminal locations. The current glove prototype 3 utilizes the commercial terminals as durability was deemed more important than a slight hindrance to the associate’s movements. This is because the data collection stability was critical, and the hindrance could be corrected later. Associate feedback did not state that the terminals blocked them from flexing their hand to the extent that they normally would.

The “bad” connections yielded useful data that helped determine connection quality and to classify the connections being made. Further discussions on this are found below in Chapter 7 Data Analysis.

# Chapter 7

# Data Analysis

Data collected from lab and line tests contain noise due to movements involved other than making a connection. Output data from the shear and normal sensors were segmented to get desired data where the actual connection is made, and further repeatability of the sensor verified using the segmented data. Only the push-pull-push part from the signal is separated by analyzing each signal manually. Six lab and seven line output data files were selected randomly to check the correlation between the signals. The linear regression analysis was run on the data. The regression model gives the quantitative relationship between two variables. It explains how one variable change according to the change of another variable. Hence, to check the degree of fit between output datasets, two outputs were compared simultaneously using regression analysis.

## 7.1 Signal Correlation

While studying the regression analysis of sensor outputs, the following factors were considered:

- $R^2$ (also called coefficient of determination): $R^2$ value measures the proportion of variability in the data that the model can explain.

- $R^2_{Adjusted}$ value: $R^2_{Adjusted}$ is a modified version of $R^2$ that has been adjusted for the number of predictors in the model.

- P-Value: P-Value checks whether the model is statistically significant. If the p-value is greater than 0.05, then the model is not statistically significant at the 5% significance level.
- Root Mean Square Error: RMSE is the standard deviation of the residuals. Residuals are the difference between the actual value and fitted value which is a measure of how far from the regression line data points are, where RMSE is a measure of how spread out these residuals are.
- Akaike's information criterion (AIC): AIC compares the quality of a set of statistical models to each other. The AIC will take each model and rank them from best to worst. The "best" model will be the one that neither under-fits nor over-fits.
- AICc is an Akaike information criterion corrected for the sample size.

RMSE is calculated as shown below in Eq. 7.1. $\hat{x}_i$ are predicted values, $x_i$ are observed values, and n is the number of observations [90].

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} \left(\hat{x}_i - x_i\right)^2}{n}} \tag{7.1}$$

AIC is calculated as shown in Eq. 7.2. $k$ is the number of parameters in the model and $\hat{L}$ is maximum value of the likelihood function for the model [91].

$$AIC = 2k - 2\ln(\hat{L}) \tag{7.2}$$

The samples from randomly selected output data files of shear and normal sensors are compared to each other by selecting one of them as a dependent variable and the other one as the independent variable.

### 7.1.1 Dynamic Time Warping

Due to segmentation, data files had different sample sizes, and the same sample size is required to run Regression Analysis. Hence, Dynamic Time Warping function was used to resample the data by stretching data points to achieve the same sample size without altering the signal output. The idea is to compare arrays with different lengths in a one-to-one fashion to investigate similarities in the signals regardless of time. Fig. 7.1 shows the benefit of DTW. The red and blue traces show very similar profiles with varying time. If data points were compared using a Euclidean Matching, the data points would not be synced up which can results in improper results on similarities. DTW compares like data points by finding the optimal match for troughs and peaks in the signals [92].

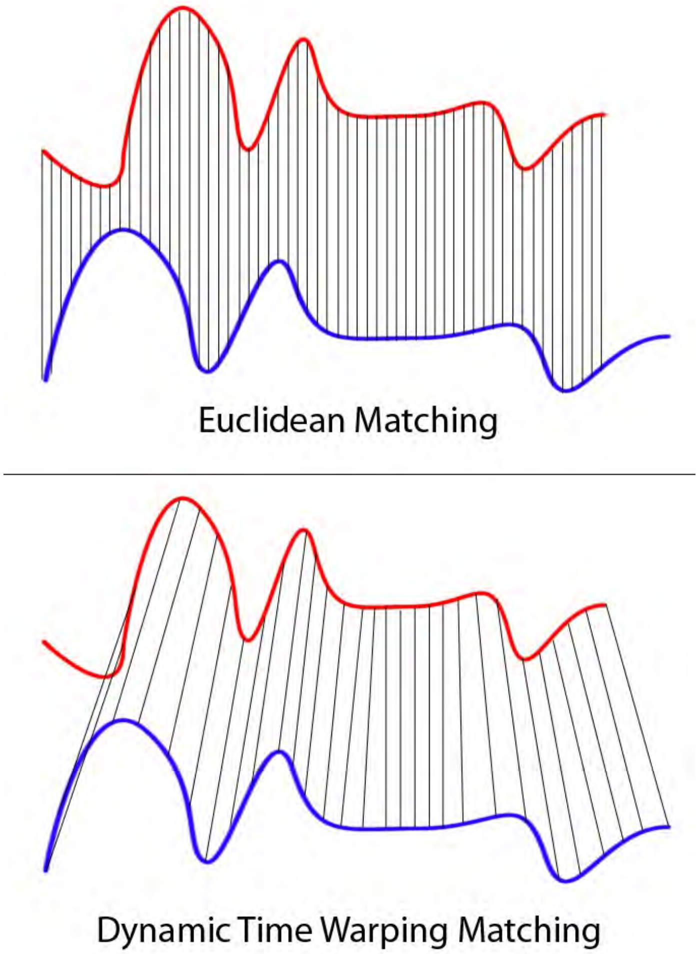


Figure 7.1: Euclidean matching vs dynamic time warping for signal comparison [92]

An example output with one signal that has been altered to match another is shown in Fig. 7.2.

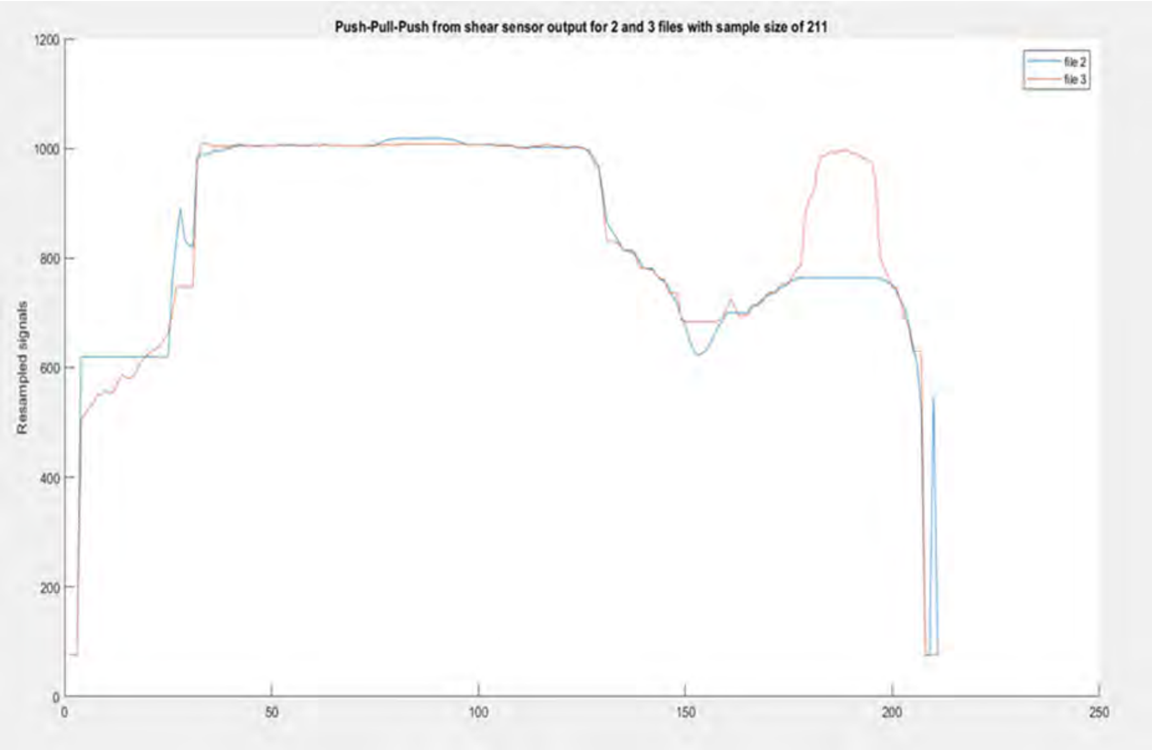


Figure 7.2: Resampling of data with dynamic time warping function

## 7.1.2 Lab vs. Lab

### 7.1.2.1 Shear Sensor

Labs output files are compared with each other for normal and shear sensor output, shown in Table 7.1. The F-test's p-value is lower than the $\alpha$ level (0.05) for all the cases, indicating that the model is statistically significant. $R^2_{Adjusted}$ values are above 89% for all the combinations of output data files. The following table shows few regression models. Test 4 has $R^2_{Adjusted}$ value equal to 89% means 89% of the variation in the output from the 5th sample can be explained by the output from the 1st sample. In simple words, the predictor (sample 5) is 89% accurate to the response (Sample 1).

Table 7.1: Regression coefficients for shear sensor lab vs lab data

| Regression test number | Sample | Comparison sample | Rˆ2 | Rˆ2 Adjusted | P-value | RMSE | AIC | AICc |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 0.991625 | 0.991528 | 3.83E-92 | 27.29739 | 843.157 | 843.2966 |
| 2 | 1 | 3 | 0.971306 | 0.97105 | 3.30E-88 | 44.08063 | 1188.713 | 1188.821 |
| 3 | 1 | 4 | 0.935868 | 0.93529 | 4.84E-68 | 63.79703 | 1261.852 | 1261.961 |
| 4 | 1 | 5 | 0.891297 | 0.890242 | 1.93E-51 | 86.72539 | 1237.135 | 1237.252 |
| 5 | 1 | 6 | 0.915709 | 0.914831 | 2.32E-53 | 81.86453 | 1143.484 | 1143.61 |

The following graph shows $R^2_{Adjusted}$ value for all the Lab vs Lab tests. The $R^2_{Adjusted}$ is essentially the percentage in which the compared signals match each other. This is reflected in Fig. 7.3 below and the subsequent regression analysis graphs as well.

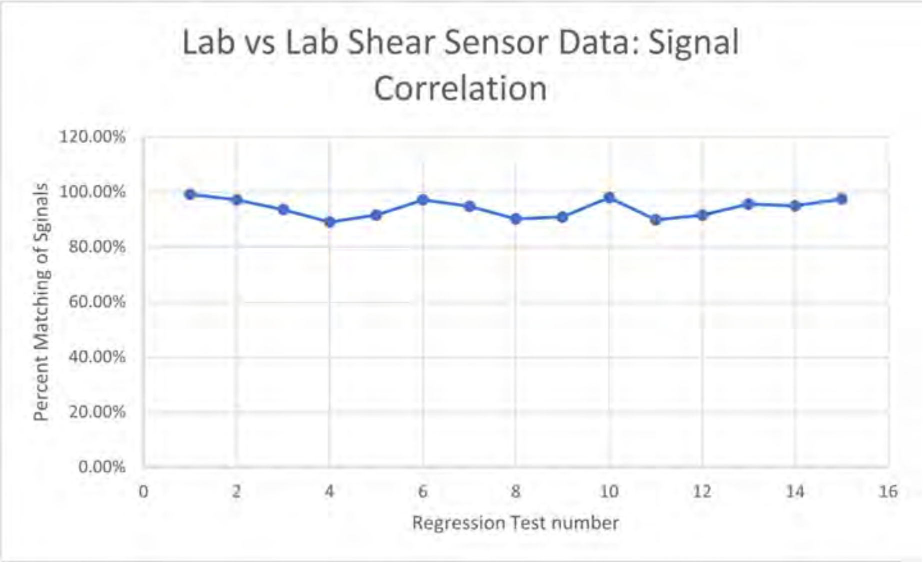


Figure 7.3: $R^2_{Adjusted}$ value for shear sensor lab vs lab data

All lab tests show similar output or a better fit with each other.

#### 7.1.2.2 Normal Sensor

Similarly, for the normal sensor, the F-test's P-value is lower than the $\alpha$ level (0.05) for all the cases, indicating that the model is statistically significant. Table 7.2 shows regression model coefficients for normal sensor output for the first lab sample with all remaining lab samples.

Table 7.2: Regression coefficients for normal sensor lab vs lab data

| Regression test number | Sample | Comparison sample | Rˆ2 | Rˆ2 Adjusted | P-value | RMSE | AIC | AICc |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 0.9837553 | 0.983577 | 3.25E-83 | 3.072282 | 474.67099 | 474.8043 |
| 2 | 1 | 3 | 0.9150573 | 0.914241 | 1.68E-57 | 7.220639 | 721.90788 | 722.0244 |
| 3 | 1 | 4 | 0.9312243 | 0.930581 | 5.06E-64 | 6.209114 | 709.382 | 709.4952 |
| 4 | 1 | 5 | 0.9329458 | 0.932232 | 5.90E-57 | 5.475754 | 600.87841 | 601.0074 |
| 5 | 1 | 6 | 0.9066443 | 0.905692 | 2.90E-52 | 6.693992 | 666.00953 | 666.1332 |

The following graph, Fig. 7.4, shows $R^2_{Adjusted}$ value for all the Lab vs Lab tests.

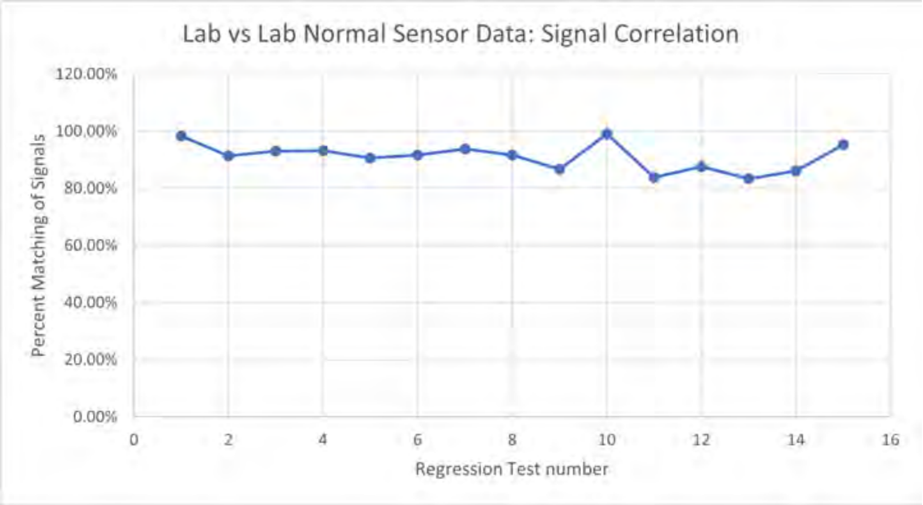


Figure 7.4: $R^2_{Adjusted}$ value for normal sensor lab vs lab data

$R^2_{Adjusted}$ values are above 83% for all the combinations of output data files. Hence, we can conclude that all lab tests for normal sensors show similar output or a better fit.

### 7.1.3 Line vs. Line

#### 7.1.3.1 Shear Sensor

Total seven random shear sensor line output files are compared with each. The F-test's p-value is lower than the $\alpha$ level (0.05) for all the cases, indicating that the model is statistically significant. Table 7.3 shows regression model coefficients for shear sensor output for the first line sample with all remaining line samples.

Table 7.3: Regression coefficients for shear sensor line vs line data

| Regression test number | Sample | Comparison sample | Rˆ2 | Rˆ2 Adjusted | P-value | RMSE | AIC | AICc |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 0.852523 | 0.851738 | 4.56E-80 | 78.19076 | 2197.664 | 2197.728 |
| 2 | 1 | 3 | 0.970897 | 0.970712 | 2.88E-123 | 35.69632 | 1600.063 | 1600.139 |
| 3 | 1 | 4 | 0.848516 | 0.847557 | 1.22E-66 | 88.94596 | 1892.217 | 1892.293 |
| 4 | 1 | 5 | 0.943816 | 0.943489 | 1.83E-109 | 52.32218 | 1872.961 | 1873.032 |
| 5 | 1 | 6 | 0.953805 | 0.953489 | 2.21E-99 | 49.70847 | 1578.22 | 1578.303 |
| 6 | 1 | 7 | 0.938571 | 0.93828 | 8.44E-130 | 53.93655 | 2305.265 | 2305.322 |

Fig. 7.5 shows $R^2_{Adjusted}$ value for all the Line vs Line tests.

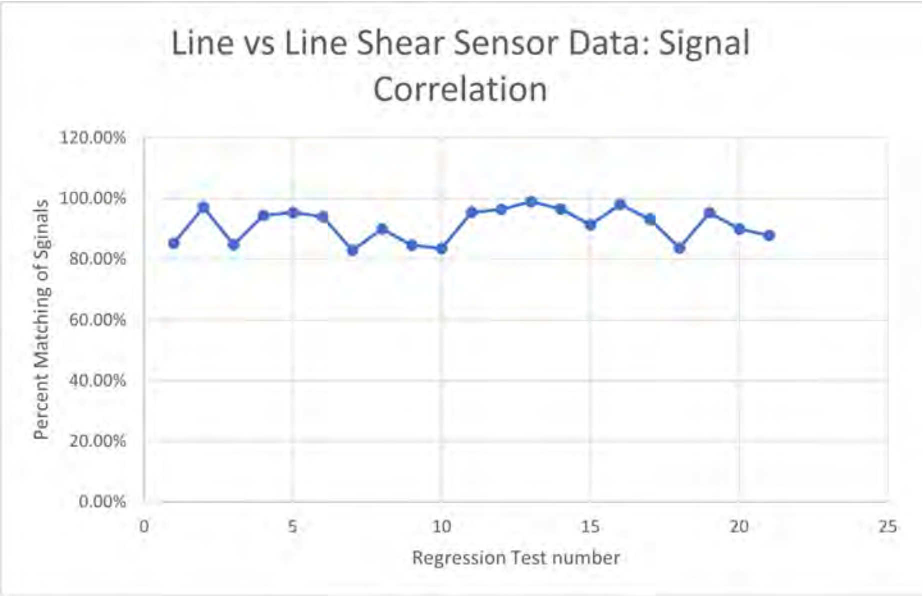


Figure 7.5: $R^2_{Adjusted}$ value for shear sensor line vs line data

$R^2_{Adjusted}$ values are above 83% for all the combinations of output data files, representing that line tests output datasets give a better fit with each other.

#### 7.1.3.2 Normal Sensor

In this case, the F-test's p-value is lower than the $\alpha$ level (0.05) for all the cases, indicating that the model is statistically significant. Table 7.4 shows regression model coefficients for normal sensor output for the first-line sample with all remaining line samples.

Table 7.4: Regression coefficients for normal sensor line vs line data

| Regression test number | Sample | Comparison sample | R^2 | R^2 Adjusted | P-value | RMSE | AIC | AICc |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 0.8541316 | 0.853303 | 1.74E-75 | 1.113619 | 545.44172 | 545.5103 |
| 2 | 1 | 3 | 0.9634408 | 0.963178 | 9.25E-102 | 0.515705 | 215.38042 | 215.4674 |
| 3 | 1 | 4 | 0.9530397 | 0.952727 | 1.60E-101 | 0.594228 | 275.11428 | 275.1948 |
| 4 | 1 | 5 | 0.9089056 | 0.908336 | 3.79E-85 | 1.029707 | 471.20862 | 471.2841 |
| 5 | 1 | 6 | 0.6734936 | 0.671346 | 8.92E-39 | 1.931597 | 641.79096 | 641.8704 |
| 6 | 1 | 7 | 0.9034291 | 0.902978 | 1.37E-110 | 0.792293 | 514.39189 | 514.4482 |

Fig. 7.6 shows $R^2_{Adjusted}$ value for all the Line vs Line tests.

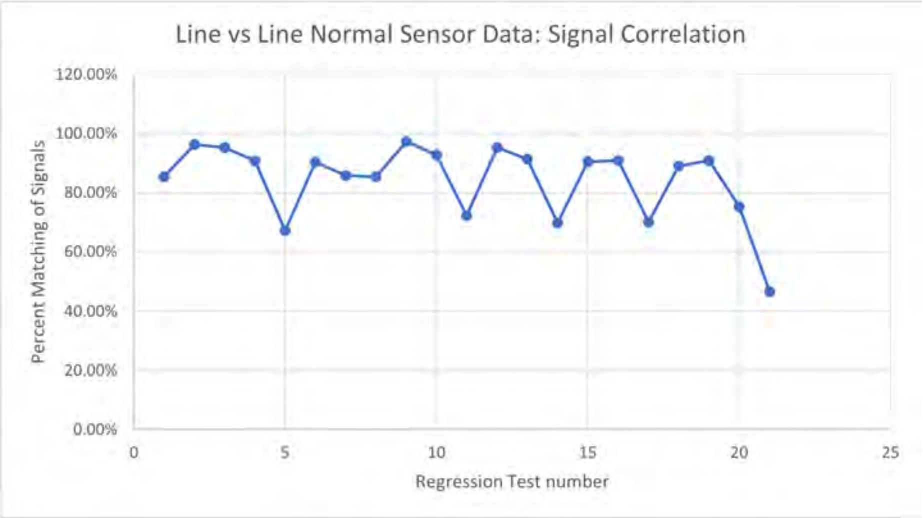


Figure 7.6: $R^2_{Adjusted}$ value for normal sensor line vs line data

$R^2_{Adjusted}$ values are above 70% for all the combinations of output data files except for tests 5 and 21. This shows that the normal force sensor shows a better fit for almost all the cases, but few cases have an average percentage of fit.

### 7.1.4 Lab vs. Line

#### 7.1.4.1 Shear Sensor

Lab and Line output files are compared with each other for normal and shear sensor output. The F-test's p-value is lower than the $\alpha$ level (0.05) for all the cases, indicating that the model is statistically significant. Table 7.5 shows regression model coefficients for shear sensor output for the first lab sample with all line samples.

Table 7.5: Regression coefficients for shear sensor lab vs line data

| Regression test number | Sample | Comparison sample | R^2 | R^2 Adjusted | P-value | RMSE | AIC | AICc |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 0.915345 | 0.914622 | 1.42E-64 | 69.34122 | 1348.582 | 1348.685 |
| 2 | 1 | 3 | 0.966736 | 0.966542 | 2.63E-128 | 41.86462 | 1785.058 | 1785.128 |
| 3 | 1 | 4 | 0.909712 | 0.908966 | 4.96E-65 | 71.37986 | 1400.974 | 1401.074 |
| 4 | 1 | 5 | 0.922371 | 0.921842 | 1.79E-83 | 68.7332 | 1685.439 | 1685.522 |
| 5 | 1 | 6 | 0.92096 | 0.920415 | 8.55E-82 | 70.3507 | 1669.681 | 1669.764 |
| 6 | 1 | 7 | 0.895143 | 0.894355 | 5.40E-67 | 78.33279 | 1562.559 | 1562.65 |
| 7 | 1 | 7 | 0.957662 | 0.95744 | 4.20E-133 | 47.50619 | 2039.992 | 2040.055 |

Fig. 7.7 shows $R^2_{Adjusted}$ value for all the Lab vs Line tests.

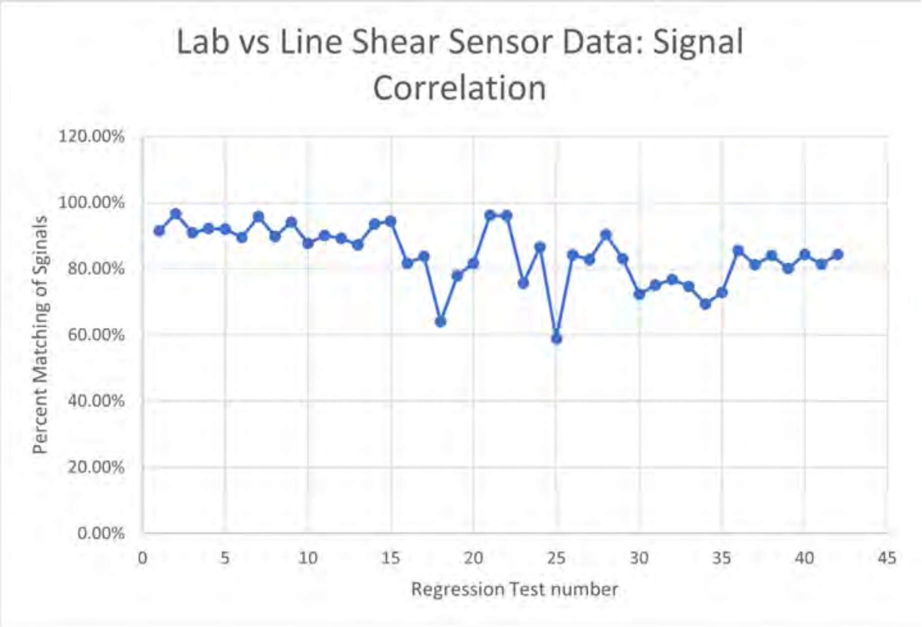


Figure 7.7: $R^2_{Adjusted}$ value for shear sensor lab vs line data

$R^2_{Adjusted}$ values are above 70% for all the combinations of output data files except for tests 18 and 25 out of 42 tests. This shows that shear sensor output from lab tests matches better with almost all the line tests, except few tests show an average fit between lab and line tests.

#### 7.1.4.2 Normal Sensor

The F-test's p-value is lower than the $\alpha$ level (0.05) for all the cases, indicating that the model is statistically significant. Table 7.6 shows regression model coefficients for normal sensor output for the first lab sample with all line samples.

Table 7.6: Regression coefficients for normal sensor lab vs line data

| Regression test number | Sample | Comparison sample | R^2 | R^2 Adjusted | P-value | RMSE | AIC | AICc |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 0.5648337 | 0.562164 | 2.94E-31 | 2.622262 | 788.36982 | 788.4439 |
| 2 | 1 | 3 | 0.4059839 | 0.403271 | 1.43E-26 | 3.454026 | 1177.0387 | 1177.094 |
| 3 | 1 | 4 | 0.6031959 | 0.600776 | 9.66E-35 | 2.230168 | 739.36502 | 739.4386 |
| 4 | 1 | 5 | 0.5206095 | 0.517946 | 1.50E-30 | 2.475185 | 848.38127 | 848.4483 |
| 5 | 1 | 6 | 0.4194545 | 0.416211 | 6.66E-23 | 4.043973 | 1021.441 | 1021.508 |
| 6 | 1 | 7 | 0.3426734 | 0.338807 | 3.35E-17 | 7.094066 | 1164.0881 | 1164.159 |
| 7 | 1 | 7 | 0.4872189 | 0.48498 | 4.60E-35 | 2.161028 | 1013.5507 | 1013.603 |

Fig. 7.8 shows $R^2_{Adjusted}$ value for all the Lab vs Line tests.

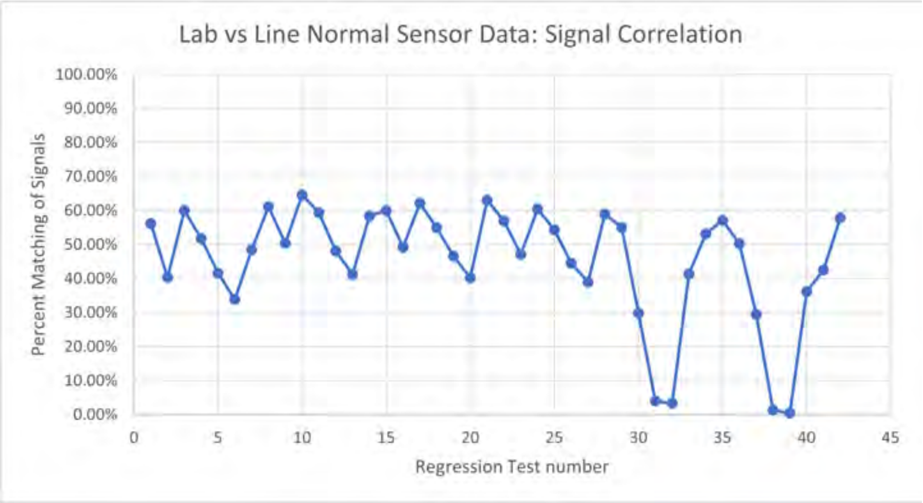


Figure 7.8: $R^2_{Adjusted}$ value for normal sensor lab vs line data

$R^2_{Adjusted}$ values for this model are below 65% for all the combinations of output data files, indicating that normal sensor performs differently on the line when compared to the lab performance.

## 7.2 Data Classification

The data classification to evaluate connection quality was separated into multiple binary go/no-go steps. The overall data classification flow is depicted in Fig. 7.9 and a summary provided here. Additional detail and results for step 3 and 4 are provided in the subsequent sections.

The measured data are recorded in Step 1 by the wrist worn device and subsequently segmented and labeled as described in section 6.3. The data were then passed to Step 3 for evaluation as either a hose connection or not-a-hose connection. The not-a-hose connection class included measurements of body plugs and bridge connectors that were installed within the same takt as the hose connections. The primary focus of this work is Step 4 which determined the quality of hose connections that were attempted. Finally, the hose connection quality evaluation was output as either a good or rework connection (go/no-go).

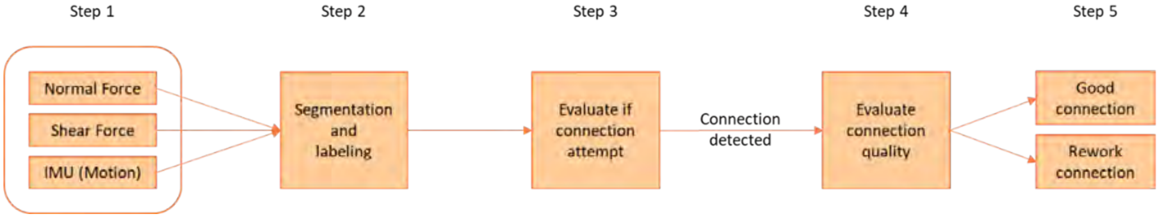


Figure 7.9: Data processing approach from sensor measurement (left) to final output (right)

The datasets used were processed entirely offline in this work. By doing so, Step 1 and 2 were completed by the experimenters during the data gathering, segmenting, and labeling activities. This involved manually investigating each bit of gathered data to segment the data around the push-pull-push. If the connection were a rework connection, it was segmented around activation points. Each segmented dataset were labeled as a ”good” or ”rework” connection. The evaluation methodology was designed with a switch to an online approach in mind, hence the inclusion of Step 1 and 2, which is detailed in the section Scaling of Data Processing. Using Bulling *et al.* taxonomy of Human Activity Recognition (HAR), the system characteristics were it will be an offline user-independent system that recognizes segmented sporadic using a stateless model [93]. The description of each characteristic is provided in Table 7.7.

Table 7.7: Design characteristics of classification model using taxonomy by (Bulling et al., 2014)

| **Type** | **Characteristic** | **Description** |
|---|---|---|
| Execution | Offline | The system records the sensor data first and recognition is performed afterwards |
| Generalization | User independent | The system is optimized to work with a large number of users |
| Recognition | Isolated (Segmented) | The system assumes that the sensor data stream is segmented at the start and end of a gesture by an oracle. Each segment is only classified into one class. The oracle can be an external system or the experimenter |
| Activities | Sporadic | The activity or gesture occurs sporadically, interspersed with other activities |
| System Model | Stateless | The system does not model the state of the world. Activities are recognized by spotting specific sensor signals. |

All data preprocessing and modeling was completed using Python 3.8.5, sklearn 0.24.1, scipy 1.6.1, numpy 1.19.2, pandas 1.2.3, seaborn 0.11.1, and matplotlib 3.3.4. The random state variable for modeling was set to 7 where applicable for repeatability.

### 7.2.1 Scaling of Data Processing

The two force inputs were checked in Step 2 that the average signal of each exceeds a specified threshold detection level. Thresholding reduced the amount of data sent for evaluation in the case that nothing was being held by the associate or only lightly held, indicating that a part installation was not taking place. If the threshold was exceeded, the data were passed to Step 3 for evaluation as either a hose connection or not-a-hose connection.

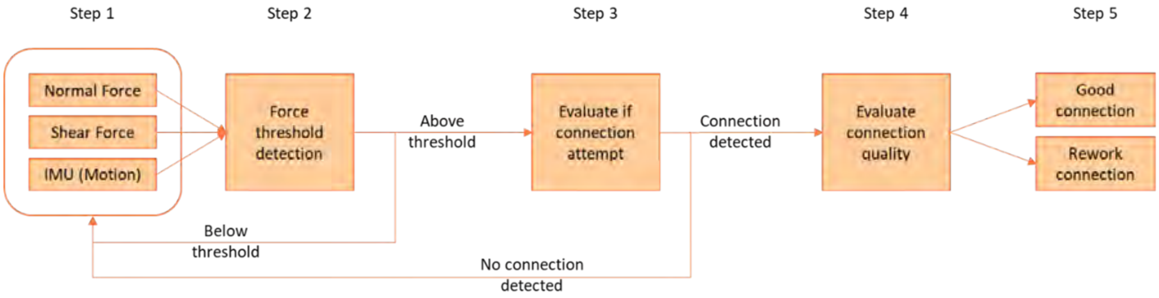


Figure 7.10: Data processing approach from sensor measurement (left) to final output (right)

## 7.2.2 Go/no-go classification of connection quality

### 7.2.2.1 Feature Extraction

Feature extraction was performed on raw data to provide additional derived data that may have been able to better inform the learning and generalization of the classifier. The features used were selected based on past literature in characterizing body worn sensors and included the signal mean, variance, minimum value, maximum value, and for the force sensors only, area under the curve. Area under the curve was calculated using the composite trapezoidal rule to integrate the signal. The area under the curve was included to inform the classifier of the amount of “energy” contained in the force signals. This was hypothesized to assist with differentiating variations in the level of applied connection force. Each feature was computed across the entire segmented connection event which varied in duration between samples. A summary of the features used per sensor is included in Table 7.8.

Table 7.8: Features used in model training and classification

| **Sensor** | **Feature** |
|---|---|
| Normal Force<br>Shear Force | Area under curve |
| | Mean |
| | Variance |
| | Minimum |
| | Maximum |
| Accelerometer | Mean |
| | Variance |
| | Minimum |
| | Maximum |
| Gyroscope | Mean |
| | Variance |
| | Minimum |
| | Maximum |

The data were further scaled to a 0-1 range based on the known sensor output limits and any outliers were removed. Scaling the input data better accounts for the varied ranges of input feature scale from each of the sensors, this preprocessing step was used to match the relative magnitudes between the features. Depending on the algorithm used, a weight feature with units of kg could be given a higher weight or bias than one with units of mg due to the common method of using the Euclidean distance between two data points in an algorithm's computation. Existing methods include standardization, scaling, min-max scaling, and unit vector approaches. Scaling was chosen for this work since the sensor outputs had known limits and the data encompassed the full range during measurement. An example of scaling was the force sensor output limits ranged from 0-3.3 volts before scaling and after scaling ranged from 0-1.

### 7.2.3 Data Modeling and Summary Model Description

The data used included 52 segmented connections from both the Vehicle Assembly Center laboratory and Plant Spartanburg Hall 52 production line. The 30 good connections were gathered from the laboratory and the 22 rework connections were gathered from the active Hall 52 production line. The breakdown in rework connection type is included in Table 7.9. Also collected were 6 bridge connector signature, but those were left out of the quality classification dataset. This dataset is a relatively small number of data points but enough to test initial classification performance. A larger number of connections would be needed to validate classification performance and to classify by rework connection type rather than a binary go/no-go.

Table 7.9: Dataset class composition

| **Binary Type** | **Connection Label** | **Count** |
|---|---|---|
| Good | Good Connection | 30 |
| Rework | Lower Low Force | 2 |
| | Lower Misplaced Force | 2 |
| | Upper Low Force | 8 |
| | Upper Misplaced Force | 10 |

Multiple model types were trained on the dataset and were selected based on prior literature and past experiences with modeling body worn sensor data. Classifier modules were simply imported to output results with minimal extra code and therefore, processing time. Hyperparameters were set for each classifier module including kernel type, alpha values, gamma values, number of estimators, etc.. Classifiers were cycled through with corresponding results. Processing time is already rather quick, but will continue to improve as the window function is further implemented from a live stream of data rather than importing the data.

The five investigated models are presented in order of performance: Gaussian Naïve Bayes, Random Forest, Support Vector Machine (SVM) using a Radial Basis Function (RBF) kernel, Gaussian Process Classifier, and Multi-Layer Perceptron classifier. The following is a summary description of the models. Performance results of the models is discussed further in the next section.

Naïve Bayes classifiers are a group of probabilistic classifiers based around applying Bayes' theorem with the assumption of strong independence between features. Training of Naïve Bayes models is done by evaluating a closed-form expression to determine the maximum-likelihood estimation or determining the parameters of a probability distribution that maximize a given likelihood function using the observed data. Maximum-likelihood estimation takes linear time to solve, compared to more computationally expensive iterative approximation methods used in other types of classifiers. Naïve Bayes classifiers are also highly scalable in their application as they require several parameters linear in the number of variables (features/predictors) in a learning problem.

A random forest estimator fits a number of decision tree classifiers on sub-sets of the total dataset. A decision tree is a structure of questions where a data instance descends through each branch, using the features in each level of the tree until reaching a leaf or end node, which contains the predicted class. Averaging across the resulting decision trees is used to improve predictive accuracy and control over-fitting.

SVM is a machine learning modeling approach that utilizes algorithms for two-group classification problems. The model learns from labeled data and returns the class based on a determined hyperplane. It can do this at very fast rates compared to newer algorithms such as neural networks and has better performance with smaller data sets [94]. The RBF kernel is commonly used in conjunction with SVM models and is widely used becuase of its similarity to the Guassian distribution [95].

The Gaussian Process Classifier is another machine learning algorithm that are a generalized by the Gaussian distribution. Like SVM, they are a kernel type, but unlike SVM, the Gaussian Process Classifier can predict highly calibrated class probabilities. Gaussian Process Classifier is a non-parametric algorithm, and similarly to SVM it can be applied to binary classification tasks [96].

A MLP is a supervised learning algorithm class of feedforward artificial neural network. A single perceptron is a neuron model that feeds into larger neural networks [97]. The MLP classifier is trained using gradient descent and the gradients are calculated using backpropagation and the minimum of the Cross-Entropy loss function. This algorithm utilizes relu, sgd uses adam solver, and a constant learning rate.

### 7.2.4 Model Classification Evaluation Metrics and Results

Metrics for evaluating binary HAR and other machine learning algorithms are well established. These metrics are based on values measured from testing the model on pre-characterized test data and are derived from four standard values. Below, the case of binary classification is used to present the equations, however, these can be expanded to included two or more classes. The below measures represent the confusion matrix possibilities.

- True Positive (TP): The number of positive instances classified as positive.

- True Negative (TN): The number of negative instances that were classified as negative.
- False Positive (FP): The number of negative instances that were classified as positive.
- False Negative (FN): The number of positive instances that were classified as negative.

Accuracy (ACC) or the overall classification performance is the ratio between the number of correctly classified samples over all samples as in Eq. 7.3. Accuracy is a quick way to gauge model performance and scales easily to non-binary classification. However, in cases of imbalanced training data where the number of samples in one class is much larger than the number of samples in the other classes, accuracy is not reliable as it tends to overestimate the ability of the classifier to predict the majority class or the class with the highest number of samples (Akosa, 2017).

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{7.3}$$

When calculating accuracy of a model on datasets with unequal distributions, the balanced accuracy (Bal ACC) is commonly used. The balanced accuracy is based on additional metrics that answer the manufacturing questions "How many truly defective products did I recall?" and "How many truly non-defective products did I recall?". When the classes of a training dataset are imbalanced, it is preferred to use the balanced accuracy over the standard accuracy metric. If the dataset classes are balanced than the balanced accuracy is equal to the accuracy.

$$TruePositiveRate(TPR) = TP/(TP + FN) \tag{7.4}$$

$$TrueNegativeRate(TNR) = TN/(TN + FP) \tag{7.5}$$

$$BalancedAccuracy = (TPR + TNR)/2 \tag{7.6}$$

A metric not as widely reported however well-representative of classifier performance is the Matthew's Correlation Coefficient (MCC) which is a measure of the quality of binary classification which includes both true and false positives and negatives. MCC is generally regarded as providing a more balanced measure of classification model performance even when the classes are very different sizes [98, 99].

$$MCC = (TPTN - FPFN)/((TP + FP)(TP + FN)(TN + FP)(TN + FN)) \tag{7.7}$$

These metrics can also be used for non-binary classification models, where more than two classes are present in a single problem, by measuring them for each individual class rather than evaluating them for the model as a whole.

To further validate the performance of the top models, a *k-fold* cross validation was completed for each model using 3 folds, shuffled data, and all three performance metrics were calculated for each. *K-fold* cross-validation holds out a portion of the dataset during training of the classifier. The partitioning used is propagated through the dataset by splitting the dataset into $k$ smaller sets or folds and training the model using *k-1* folds. The model is then iteratively trained on each split of folds until all folds have been used for both training and testing as in Fig. 7.11. The resulting performance metrics for each iteration or split are used to compute the average value.

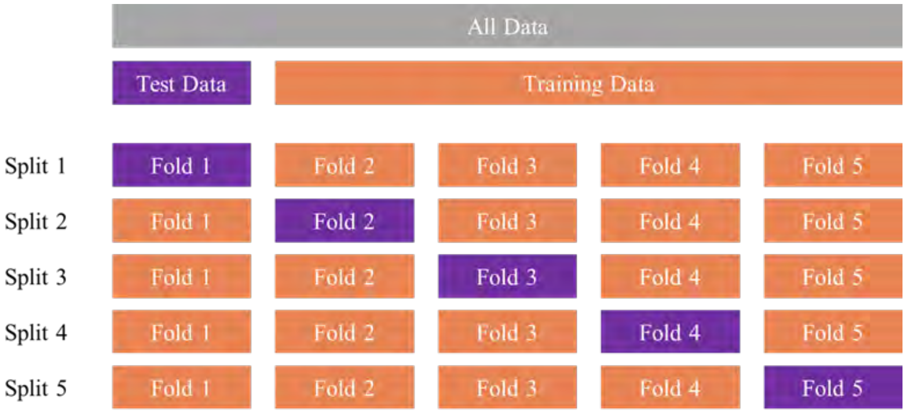


Figure 7.11: Representation of k-fold Cross Validation with k=5 splits

The *k-fold* method provides an increased percentage of training data allowed the resulting classifier to learn from more data points, potentially encompassing more data variability when compared to the more common fixed 60% training/40% testing dataset split used in many machine learning applications. The smaller size of the *k-fold* test dataset increases the chance of the model being overfit to the test data. The potential for overfitting is diminished by iteratively splitting the dataset and averaging the model performance over all iterations. A second motivation for the *k-fold* cross-validation as a check in selecting the top models was that the dataset encompassed time series data. Time series data changes over time and so different areas of the dataset contain differing distributions of information. By iterating over the full dataset, the resulting classifier is validated against all variability in the collected dataset.

The *k-fold* model performance and standard deviation are summarized in Fig. 7.12 and are ordered by MCC value.

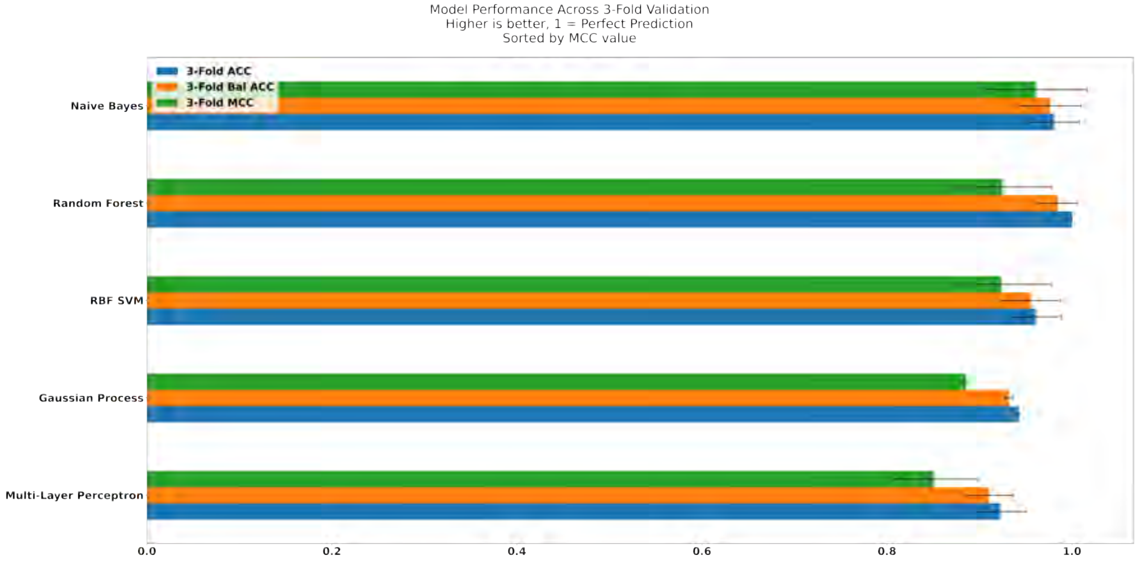


Figure 7.12: Top 5 model classification performance

From Fig. 7.13, all five models presented performed very well in classifying the collected data. All models had above 92% balanced accuracy (orange bar) including one standard deviation of the *k-fold* mean. Additionally, all models had above 0.85 MCC value with 1 being a perfect prediction, including one standard deviation from the k-fold mean. Fig. 7.13 proposes a solution which utilizes multiple classifier methods to create a balanced prediction increasing the confidence in the feedback to the operator.

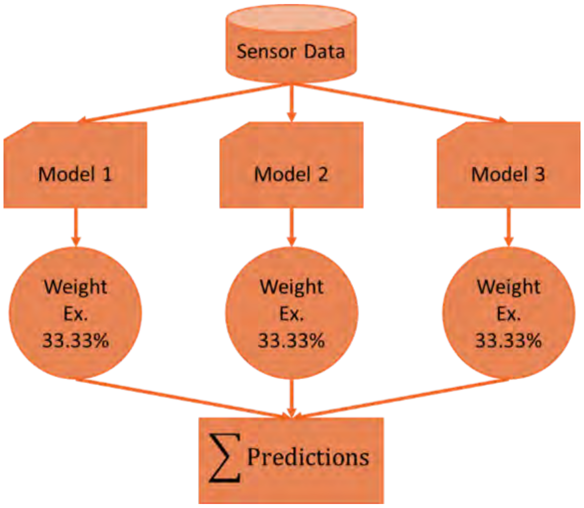


Figure 7.13: Ensemble of three equally weighted models to improve the performance of the resulting classifier

# Chapter 8

# Summary of Contributions

The goal of this study was to provide accountable measures from assembly line operatives to a manufacturer for process quality improvement in manually completed tasks. To accomplish this, quality improvement methods were evaluated which led to a wearable sensing system validation system. After a market and literature review, a sensing glove was investigated. Methodologies for the sensor glove can be applied to other format and operations, not to limit the findings in this research to only a sensor glove format. The objective of this research is to model the relationship between force, sound, motion and other sensed signals in manual assembly environments and the resultant quality of connections made. The proposed sensor glove offers a unique sensing measurement capability and approach that offers a level of robustness and continuance to be successfully incorporated into harsh environments in manufacturing. A coupled wrist unit moves necessary electronics away from operator movement points and potentially damaging surroundings. The sensor glove captures shear force, normal force, acceleration, and gyroscopic information for process evaluation. These measure can be used to determine operator approach and engagement for an objective determination of quality. The sensor system considered interviewed associates after assembly line testing and users reported little to no disturbance in their workflow and a minimally invasive approach.

The sensor glove is paired with a well-conditioned classification algorithm which evaluates segmented data and discerns a connection as "good" or "rework". The classifier applies 5 methods to the data; Random Forest, Naive Bayes, RBF SVM, Gaussian Process, and Multi-layer Perception. Utilizing multiple models for process evaluation provides a more encompassing approach that is not sensitive to outlier phenomena for increased feedback confidence.

The sensor glove is investigated for the use of connector quality evaluation on an automotive OEM's assembly line. Throughout the investigation, data were collected in the lab and on the assembly line. This included good connections and purposeful failure at both locations. The signals were run through a regression analysis to determine the correlation of data collected in a research environment to what is seen on the assembly line. The multimodal sensing assisted with achieving a repeatable and measurable process for the classifier. The classifier results reported a high accuracy in the 5 model methods. The methods had an average balanced accuracy of 94.8% with Naive Bayes performing the best with a 97.6% balanced accuracy.

The idea is to augment the associate with various sensing forms to describe the response of the system as a measurement of quality to minimize defects without overextending process times. The investigation was successful in this venture providing a low-cost, wearable sensing approach that is minimally invasive to the operator and to process times.

### 8.0.1 Key Knowledge Contributions

- Design and validation of a flexible, durable, and low-cost shear force sensor
- Architecture of a wearable, non-intrusive force sensing system for manufacturing assembly line use
- A multi-modal sensor fusion classifier for assembly connections
- Capture of intuitive final assembly operations through sensor signal relationships that require complex movements that cannot yet be automated

## 8.1 Recommendations for the Graduate School at Clemson University

Continue to offer funded opportunities through GTA, GRA, fellowships, etc..; The more you have the better. This was a key element I considered when researching graduate schools. I amassed a rather large amount of student loan debt for my undergraduate schooling. I was not prepared to take on additional debt while my old debt continued to gain interest.

Incorporate deadlines and deliverables for the Automotive Engineering MS with a thesis into the grad student handbook. I am told that I am the first Automotive Engineering student to complete the MS with a thesis degree. As a results, everyone involved was learning what to do with me. For future students, this should be incorporated into the handbook just like the other options. I think this will also entice other students to look into a MS with a thesis as well, especially those already involved in research.

# Chapter 9

# Future Work

Future developments of the physical data capture and evaluation system for process quality improvement can be broken down into 4 main categories; Sensor glove industrialization, extended line trials, classification algorithm refinement, printed sensor development, and a durability study. Each of the aforementioned sections are discussed below indicating key elements to progress the sensing system towards widespread implementation into manufacturing environments.

Throughout the next steps of this work, it is important to consider the human factors before reaching decisions. This will ensure the sensing system endorses usability and comfort for the operators. The system in place should not be a burden to the operator or manufacturer, but rather a robust tool that can perform its function well without requiring extensive training, attention, or instruction.

## 9.1 Sensor Glove Industrialization

Unlike many wearables found in research or on the current market, the proposed sensor glove already achieves a high level of robustness and durability that a manufacturing environment requires. This is a great stage to launch the system into an industrialization phase focusing on the weaker points of the sensing system to ensure long-term usability. Durability should be increased before deployment to avoid any disconnects in the circuit resulting in all data not being collected and processes. This was an issue briefly had during one day of assembly line testing that should be mitigated in future endeavors.

One of the most vulnerable points for damage on the sensor glove system are the sensor leads, specifically the shear sensor. The shear sensor currently has a terminal with a plastic housing just below the thumbprint of the operator. This restricts the operator from fully closing their fist. During line trials at BMW, associates were asked to review their experience with the glove including comfort remarks. The associates reported no hindrance with their work despite the plastic terminal cover in line with their thumb. This is most likely due to the associate not needing to fully close their hand when completing their work in the investigated areas. However, the current state is not ideal for implementation across other processes at BMW or other manufacturing OEMs and suppliers. Therefore, the sensor leads should be amended before widespread implementation.

The shear sensor leads should ideally be like the normal force sensor leads; printed and encased in a flexible plastic. Concurrent with this development, there should be a sensor lead rerouting of both sensors. This should take the leads away from the inside surfaces of articulating joints to inhibit bending. Routing the leads to the back of the hand to the wrist unit will keep them out of harms way from articulation and tool usage. Integrating the leads into the glove will forgo the current secondary glove which is in place to provide and additional layer of protection to the loose wires and leads.

Adding in a disconnect between the wrist unit and sensor glove can also provide easier access to the operators and prevent the leads connecting the two from being ripped or snagged. This could be solved by simply integrating a locking connection allowing the user to put on the arm sleeve and glove separately without the other hanging from the leads.

The sensor glove wrist unit requires further advancement to be implemented onto the assembly line without an accompanying laptop or researcher. These developments include an internal power supply, internal postprocessing (automatically segmenting the data and running it through the classifier), and integrating a feedback method to the operator.

The feedback method should not provide any information to the operator that requires analyzing. The method should be straightforward and only required when a fault is detected. Allowing the operator to complete their work without needing to look at the system for feedback every time will empower long-term usage as it will be minimally invasive to them. Development of this should consider providing feedback to the operator wearing the sensing system and the associates downstream of the completed work. The corrective measure based on a detected fault does not necessarily have to be completed by the one wearing the sensing system. Providing feedback to an operator decoupled from the sensor glove down the assembly line can allow for unused takt times to be filled and a fresh perspective when examining connections labeled as ”rework”. Should the feedback method be implemented downstream of where the connections are made, there must be sufficient access to the investigated connectors. For example, if the connectors are completed inside on the vehicle floor, rework connections must be checked before any components such as the carpets are put in place.

In consideration of human factors, there should be an evaluation of the sensor glove system's overall usability in addition to the points mentioned above. Before completing corrective measures regarding these advancements, surveys should be conducted to attain the pro's and con's of the current system to lead development in the right direction. Other studies should be completed to ensure the system is easy and non-intrusive to use. This should be conducted with those who are unfamiliar with the sensing system to determine difficult to use components.

The feedback method to the operator should also undergo a human factors study. Ideally, this should not rely on the operator's judgement but it should relay easily discernable information in a convenient method. For example, a haptic motor or visual display of lights can be utilized. These could both properly indicate a good or rework connection to the operator. The final implementation should consider which feedback method cannot be mistaken as the opposite outcome. A haptic motor can be missed as the associates are constantly moving and engaged in the process. A light indicator may be missed if no input of acknowledgement is given from the associate.

## 9.2 Extended Line Trials

Once the sensing system is developed to the point where a researcher does not need to be present to collect data, efforts should look towards extended line trials. This will involve a long-term collection of data with assembly line operator wearing the sensing system as they complete their work normally. Data should automatically be segmented and stored during the collection process to minimize postprocessing times. This will also allow a greater number of connection data to be stored in the device as it would not be saving a continuous data stream. After collection periods, system performance and durability can be evaluated. Associates who complete the extended line trials should be interviewed after regarding usability and comfort with the sensing system.

## 9.3 Classification Algorithm Refinement

The classification algorithm needs to be refined further before completing extended line trials. This starts with further development of the window function to only save critical points (ex. push-pull-push at BMW). This will automatically segment and save data from a continuous data stream. The data that is not deemed a critical point will not be saved allowing more space on the internal storage. A depiction of the window function concept is shown in Fig. 9.1. The window function uses a given window size and step size to investigate a segment of the incoming continuous data stream for given parameters. These windows can overlap to ensure investigated parameters are not missed, but this increases processing time. Process minimum and maximum duration should be utilized to determine step and window size.

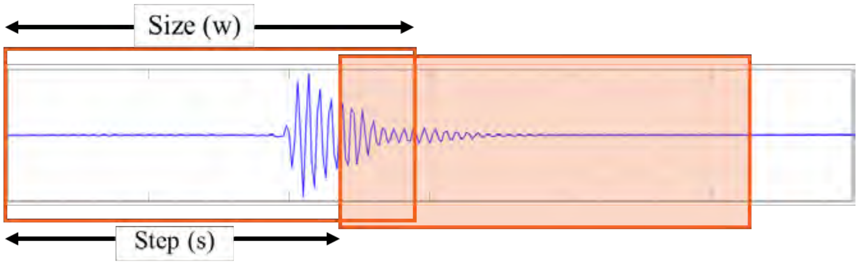


Figure 9.1: Window function methodology [100]

Once the window function development is achieved along with some other ones mentioned above, the extended line trials can begin. The data gathered during this can be run through the existing classifier. It is important to maintain communication with the manufacturer and to have them record when a fault is detected. Classifier performance can then be evaluated and subsequently excite refinement to the algorithm. Processing time of the data should also be reduced to achieve a real-time feedback is desirable.

Other algorithmic or smart tools can also be investigated. Examples of these tools include open source artificial intelligence software such as IBM Watson or Clarifai. Other methods such as image classification can be evaluated for performance.

## 9.4 Printed Sensor Development

The printed sensor design shows promise, but it needs an ample amount of further development to be suitable for a manufacturing environment. Efforts include increasing the measurement range, refining the middle layer geometry, and an evaluation and sourcing of materials. Increasing the measurement range can be done through the utilization of a different resistive ink. This will change the resistance of the sensor creating more stark differences under shear loads. The middle layer geometry left off on an asterisk (*) like shape. This enables multi-directional measurement with a controlled contact patch for a greater measurement range. Other geometries and thicknesses should be evaluated. Lastly, the materials within the sensor need to perform better. The current sensor maintains flexibility but lacks durability and structural integrity under shear loading. This seems to mainly stems from the adhesives and the middle layer material used. Sourcing and testing of replacement or augmenting materials should be had which may include discussions with current flexible sensor manufacturers.

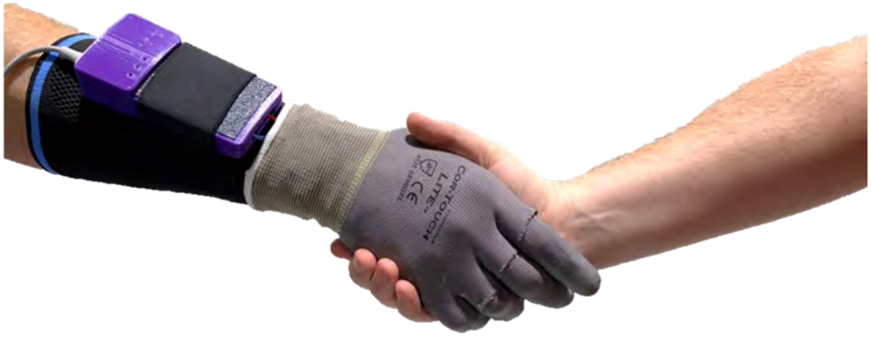


Figure 9.2: Sensing system architecture being worn and shaking hands

## 9.5 Durability Study

To further understand the capabilities of the developed sensor glove system, an investigation in the durability of the system should take place. This should begin with a Gage Repeatability and Reproducibility (GRR) study to determine the sensor glove system's accuracy to ensure measurements are repeatable and reproducible. The GRR study should be concurrently developed with a study to determine critical points of failure in the system (i.e. what is wearing out first?). This will determine weak points that require further development to reach a desired lifespan. The findings from the aforementioned investigations can then be compared to the durability of marketed sensor glove solutions to determine the applicability outside of a research environment. The literature review yielded information that led to the marketed sensor glove solutions being unfit for a manufacturing environment due to durability concerns, but this should be verified after fully understanding the durability limits of the developed sensor glove. This high level comparison of sensor gloves can yield a metric detailing operational limits regarding durability for future solutions to quantify. This will give manufacturers and those alike an idea of applicable solutions for their concerns.

# Chapter 10

# Publications

## 10.1 Journal

1. **Scott Kerner**, Suryanarayanan Gunasekar, Rishabh Mulesh Vedant, Matthew Krugh, Laine Mears. "Parametrization of manual work in automotive assembly for wearable force sensing", *Journal of Manufacturing Systems*. Vol. 59. pp. 686-700. 2021.

2. Yabin Liao, Ihab Ragai, Ziyun Huang, **Scott Kerner**. "Manufacturing process monitoring using time-frequency representation and transfer learning of deep neural networks", *Journal of Manufacturing Processes*. Vol. 68, Part A. pp. 231-248. 2021.

3. **Scott Kerner**, Matthew Krugh, Laine Mears. "Wearable shear and normal force sensing glove development for real-time feedback on assembly line processes", *Journal of Manufacturing Systems*. In-press. 2022.

## 10.2 Conference

1. Flanagan Waldherr, Suruchi Gujarathi, **Scott Kerner**, Matthew Krugh, Laine Mears. "Wearable motion and force sensing to determine force exertion and task recognition for ergonomic analysis", *Institute of Industrial and Systems Engineers Conference (IISE 2022)*. May 21-24, 2022.

2. **Scott Kerner**, Matthew Krugh, Laine Mears. "Wearable shear and normal force sensing glove development for real-time feedback on assembly line processes", *North American Manufacturing Research Conference (NAMRC 50)*. June 27-July 1, 2022.

3. Ethan Wescoat, **Scott Kerner**, Laine Mears. "A comparative study of different algorithms using contrived failure data to detect robot anomalies", *International Conference on Industry 4.0 and Smart Manufacturing (ISM 2021)*. November 17-19, 2021.

4. **Scott Kerner**, Zachery Deabenderfer, Katherine Korn, Ihab Ragai, Yabin Liao, David Loker. "Preliminary investigation on the acoustic characteristics of turning processes", *International Mechanical Engineering Congress  Exposition (IMECE 2021)*. November 1-4, 2021.

5. Suryanarayanan Gunasekar, **Scott Kerner**, Matthew Krugh, Laine Mears. "Wearable shear force-sensing for augmenting manual hose connections in an automotive assembly", *North American Manufacturing Research Conference (NAMRC 49)*. June 21-25, 2021.

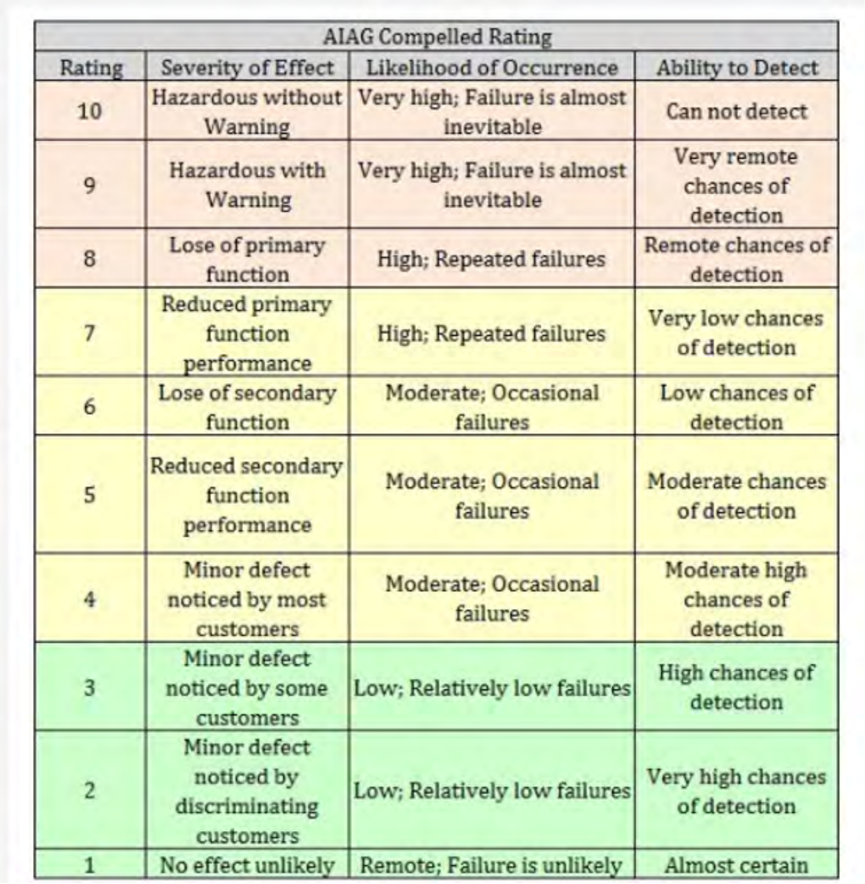
## 10.3 Poster

1. Wearable Force Sensing Glove for Manual Work in Automotive Assembly. *MSEC*. 2021.

# Appendices

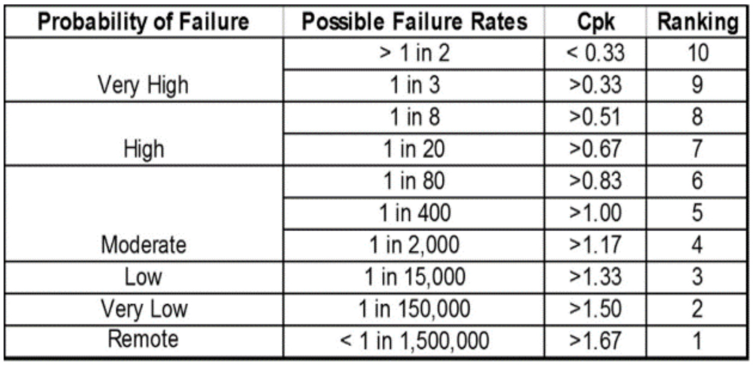
# Appendix A Process Failure Mode Effects Analysis

## A.1 PFMEA Evaluation Measures



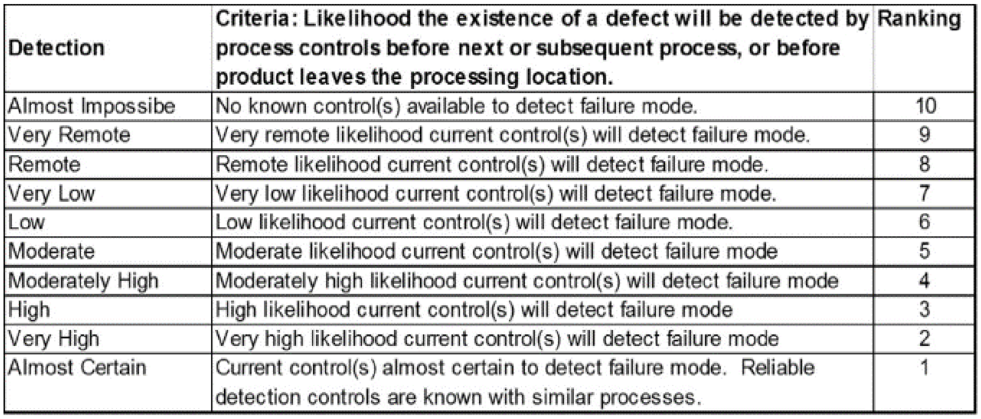
| AIAG Compelled Rating | | | |
|---|---|---|---|
| Rating | Severity of Effect | Likelihood of Occurrence | Ability to Detect |
| 10 | Hazardous without Warning | Very high; Failure is almost inevitable | Can not detect |
| 9 | Hazardous with Warning | Very high; Failure is almost inevitable | Very remote chances of detection |
| 8 | Lose of primary function | High; Repeated failures | Remote chances of detection |
| 7 | Reduced primary function performance | High; Repeated failures | Very low chances of detection |
| 6 | Lose of secondary function | Moderate; Occasional failures | Low chances of detection |
| 5 | Reduced secondary function performance | Moderate; Occasional failures | Moderate chances of detection |
| 4 | Minor defect noticed by most customers | Moderate; Occasional failures | Moderate high chances of detection |
| 3 | Minor defect noticed by some customers | Low; Relatively low failures | High chances of detection |
| 2 | Minor defect noticed by discriminating customers | Low; Relatively low failures | Very high chances of detection |
| 1 | No effect unlikely | Remote; Failure is unlikely | Almost certain |

Figure 1: PFMEA severity ranking table of failure mode [101]



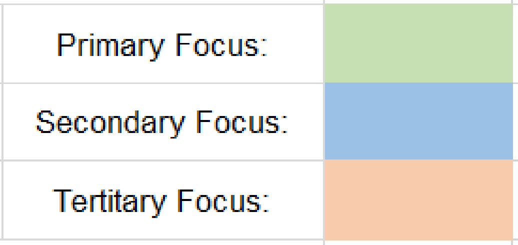
| Probability of Failure | Possible Failure Rates | Cpk | Ranking |
|---|---|---|---|
| Very High | > 1 in 2 | < 0.33 | 10 |
| | 1 in 3 | >0.33 | 9 |
| High | 1 in 8 | >0.51 | 8 |
| | 1 in 20 | >0.67 | 7 |
| Moderate | 1 in 80 | >0.83 | 6 |
| | 1 in 400 | >1.00 | 5 |
| | 1 in 2,000 | >1.17 | 4 |
| Low | 1 in 15,000 | >1.33 | 3 |
| Very Low | 1 in 150,000 | >1.50 | 2 |
| Remote | < 1 in 1,500,000 | >1.67 | 1 |

Figure 2: PFMEA occurrence rate ranking table of failure mode [101]



| Detection | Criteria: Likelihood the existence of a defect will be detected by process controls before next or subsequent process, or before product leaves the processing location. | Ranking |
|---|---|---|
| Almost Impossibe | No known control(s) available to detect failure mode. | 10 |
| Very Remote | Very remote likelihood current control(s) will detect failure mode. | 9 |
| Remote | Remote likelihood current control(s) will detect failure mode. | 8 |
| Very Low | Very low likelihood current control(s) will detect failure mode. | 7 |
| Low | Low likelihood current control(s) will detect failure mode. | 6 |
| Moderate | Moderate likelihood current control(s) will detect failure mode | 5 |
| Moderately High | Moderately high likelihood current control(s) will detect failure mode | 4 |
| High | High likelihood current control(s) will detect failure mode | 3 |
| Very High | Very high likelihood current control(s) will detect failure mode | 2 |
| Almost Certain | Current control(s) almost certain to detect failure mode. Reliable detection controls are known with similar processes. | 1 |

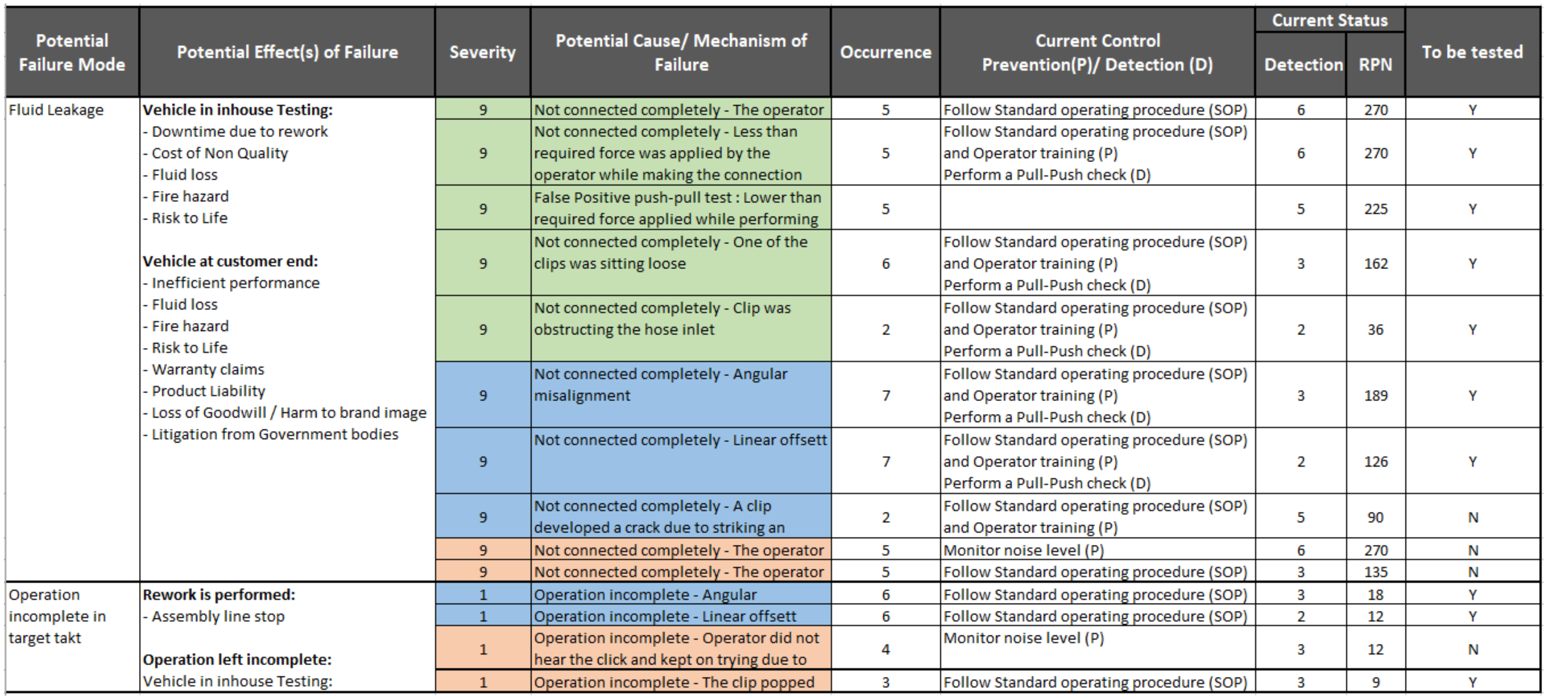
Figure 3: PFMEA detection capability ranking table of failure mode [101]



| | |
|---|---|
| Primary Focus: | 🟩 |
| Secondary Focus: | 🟦 |
| Tertitary Focus: | 🟧 |

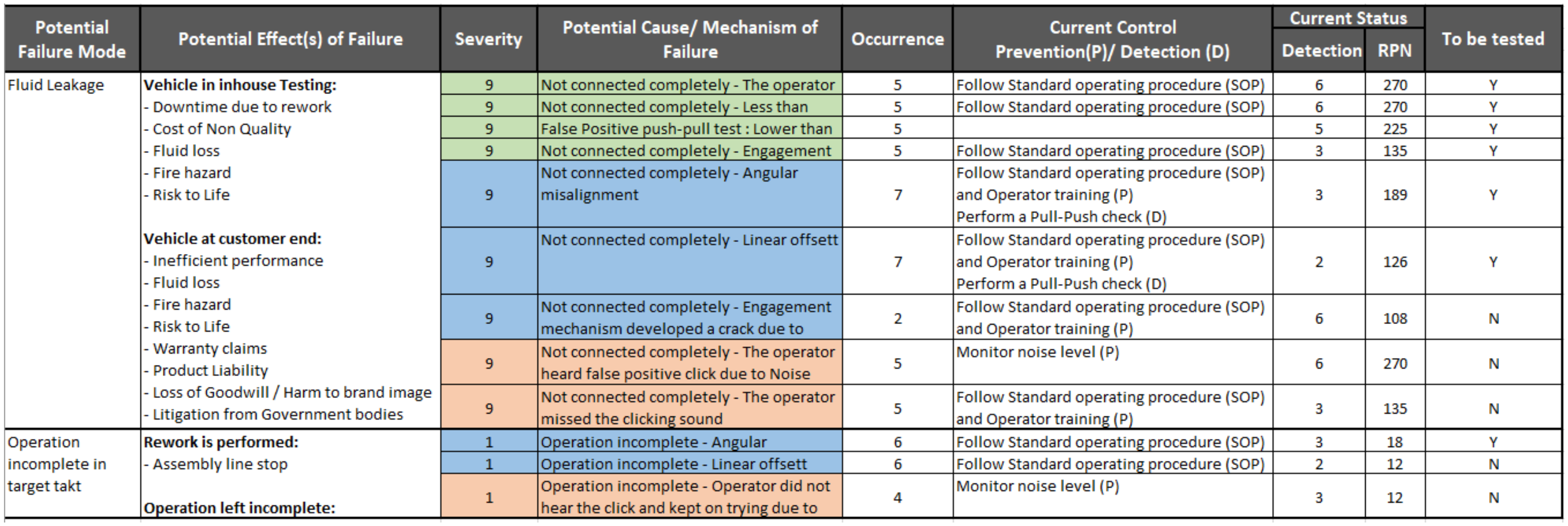
Figure 4: Focus color key breakdown for failure modes in larger PFMEA table for each connector type

## A.2 PFMEA BMW Connector Types 1-3



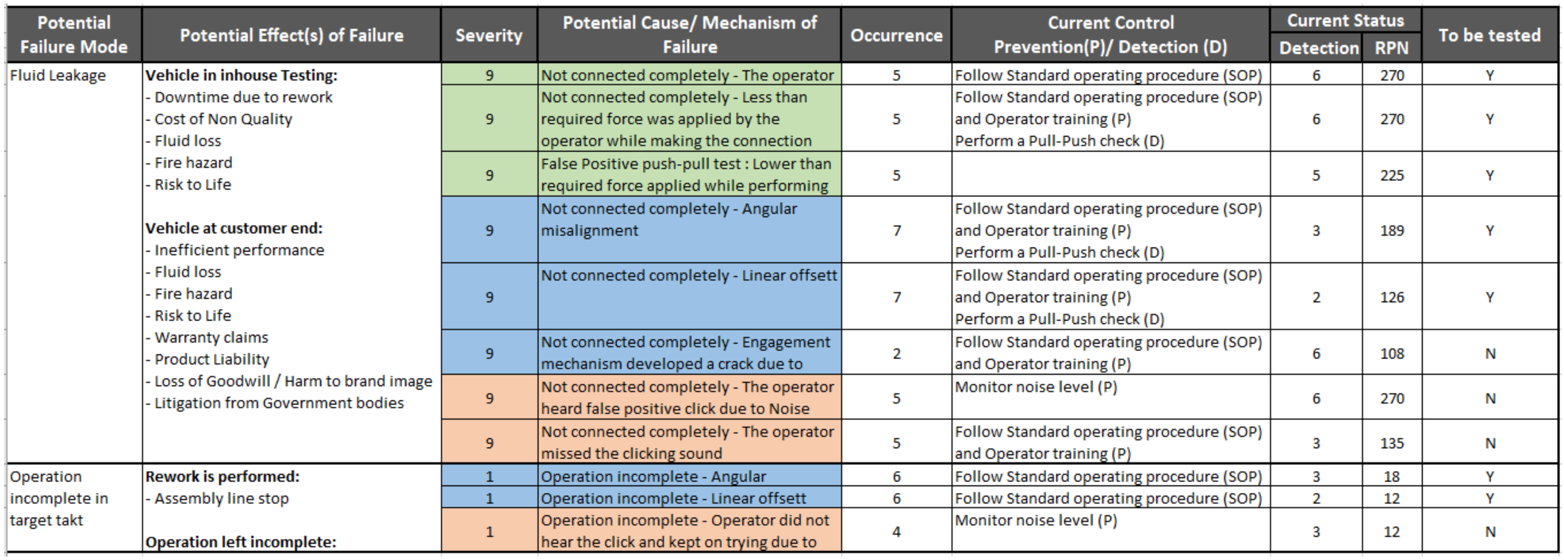
| Potential Failure Mode | Potential Effect(s) of Failure | Severity | Potential Cause/ Mechanism of Failure | Occurrence | Current Control Prevention(P)/ Detection (D) | Current Status | | To be tested |
|---|---|---|---|---|---|---|---|---|
| | | | | | | Detection | RPN | |
| Fluid Leakage | **Vehicle in inhouse Testing:**<br>- Downtime due to rework<br>- Cost of Non Quality<br>- Fluid loss<br>- Fire hazard<br>- Risk to Life<br><br>**Vehicle at customer end:**<br>- Inefficient performance<br>- Fluid loss<br>- Fire hazard<br>- Risk to Life<br>- Warranty claims<br>- Product Liability<br>- Loss of Goodwill / Harm to brand image<br>- Litigation from Government bodies | 9 | Not connected completely - The operator | 5 | Follow Standard operating procedure (SOP) | 6 | 270 | Y |
| | | 9 | Not connected completely - Less than required force was applied by the operator while making the connection | 5 | Follow Standard operating procedure (SOP) and Operator training (P)<br>Perform a Pull-Push check (D) | 6 | 270 | Y |
| | | 9 | False Positive push-pull test : Lower than required force applied while performing | 5 | | 5 | 225 | Y |
| | | 9 | Not connected completely - One of the clips was sitting loose | 6 | Follow Standard operating procedure (SOP) and Operator training (P)<br>Perform a Pull-Push check (D) | 3 | 162 | Y |
| | | 9 | Not connected completely - Clip was obstructing the hose inlet | 2 | Follow Standard operating procedure (SOP) and Operator training (P)<br>Perform a Pull-Push check (D) | 2 | 36 | Y |
| | | 9 | Not connected completely - Angular misalignment | 7 | Follow Standard operating procedure (SOP) and Operator training (P)<br>Perform a Pull-Push check (D) | 3 | 189 | Y |
| | | 9 | Not connected completely - Linear offsett | 7 | Follow Standard operating procedure (SOP) and Operator training (P)<br>Perform a Pull-Push check (D) | 2 | 126 | Y |
| | | 9 | Not connected completely - A clip developed a crack due to striking an | 2 | Follow Standard operating procedure (SOP) and Operator training (P) | 5 | 90 | N |
| | | 9 | Not connected completely - The operator | 5 | Monitor noise level (P) | 6 | 270 | N |
| | | 9 | Not connected completely - The operator | 5 | Follow Standard operating procedure (SOP) | 3 | 135 | N |
| Operation incomplete in target takt | **Rework is performed:**<br>- Assembly line stop<br><br>**Operation left incomplete:**<br>Vehicle in inhouse Testing: | 1 | Operation incomplete - Angular | 6 | Follow Standard operating procedure (SOP) | 3 | 18 | Y |
| | | 1 | Operation incomplete - Linear offsett | 6 | Follow Standard operating procedure (SOP) | 2 | 12 | Y |
| | | 1 | Operation incomplete - Operator did not hear the click and kept on trying due to | 4 | Monitor noise level (P) | 3 | 12 | N |
| | | 1 | Operation incomplete - The clip popped | 3 | Follow Standard operating procedure (SOP) | 3 | 9 | Y |

Figure 5: PFMEA for BMW connector Type 1 indicating severity, occurrence, and detection



| Potential Failure Mode | Potential Effect(s) of Failure | Severity | Potential Cause/ Mechanism of Failure | Occurrence | Current Control Prevention(P)/ Detection (D) | Current Status | | To be tested |
|---|---|---|---|---|---|---|---|---|
| | | | | | | Detection | RPN | |
| Fluid Leakage | **Vehicle in inhouse Testing:**<br>- Downtime due to rework<br>- Cost of Non Quality<br>- Fluid loss<br>- Fire hazard<br>- Risk to Life<br><br>**Vehicle at customer end:**<br>- Inefficient performance<br>- Fluid loss<br>- Fire hazard<br>- Risk to Life<br>- Warranty claims<br>- Product Liability<br>- Loss of Goodwill / Harm to brand image<br>- Litigation from Government bodies | 9 | Not connected completely - The operator | 5 | Follow Standard operating procedure (SOP) | 6 | 270 | Y |
| | | 9 | Not connected completely - Less than | 5 | Follow Standard operating procedure (SOP) | 6 | 270 | Y |
| | | 9 | False Positive push-pull test : Lower than | 5 | | 5 | 225 | Y |
| | | 9 | Not connected completely - Engagement | 5 | Follow Standard operating procedure (SOP) | 3 | 135 | Y |
| | | 9 | Not connected completely - Angular misalignment | 7 | Follow Standard operating procedure (SOP) and Operator training (P)<br>Perform a Pull-Push check (D) | 3 | 189 | Y |
| | | 9 | Not connected completely - Linear offsett | 7 | Follow Standard operating procedure (SOP) and Operator training (P)<br>Perform a Pull-Push check (D) | 2 | 126 | Y |
| | | 9 | Not connected completely - Engagement mechanism developed a crack due to | 2 | Follow Standard operating procedure (SOP) and Operator training (P) | 6 | 108 | N |
| | | 9 | Not connected completely - The operator heard false positive click due to Noise | 5 | Monitor noise level (P) | 6 | 270 | N |
| | | 9 | Not connected completely - The operator missed the clicking sound | 5 | Follow Standard operating procedure (SOP) and Operator training (P) | 3 | 135 | N |
| Operation incomplete in target takt | **Rework is performed:**<br>- Assembly line stop<br><br>**Operation left incomplete:** | 1 | Operation incomplete - Angular | 6 | Follow Standard operating procedure (SOP) | 3 | 18 | Y |
| | | 1 | Operation incomplete - Linear offsett | 6 | Follow Standard operating procedure (SOP) | 2 | 12 | N |
| | | 1 | Operation incomplete - Operator did not hear the click and kept on trying due to | 4 | Monitor noise level (P) | 3 | 12 | N |

Figure 6: PFMEA for BMW connector Type 2 indicating severity, occurrence, and detection

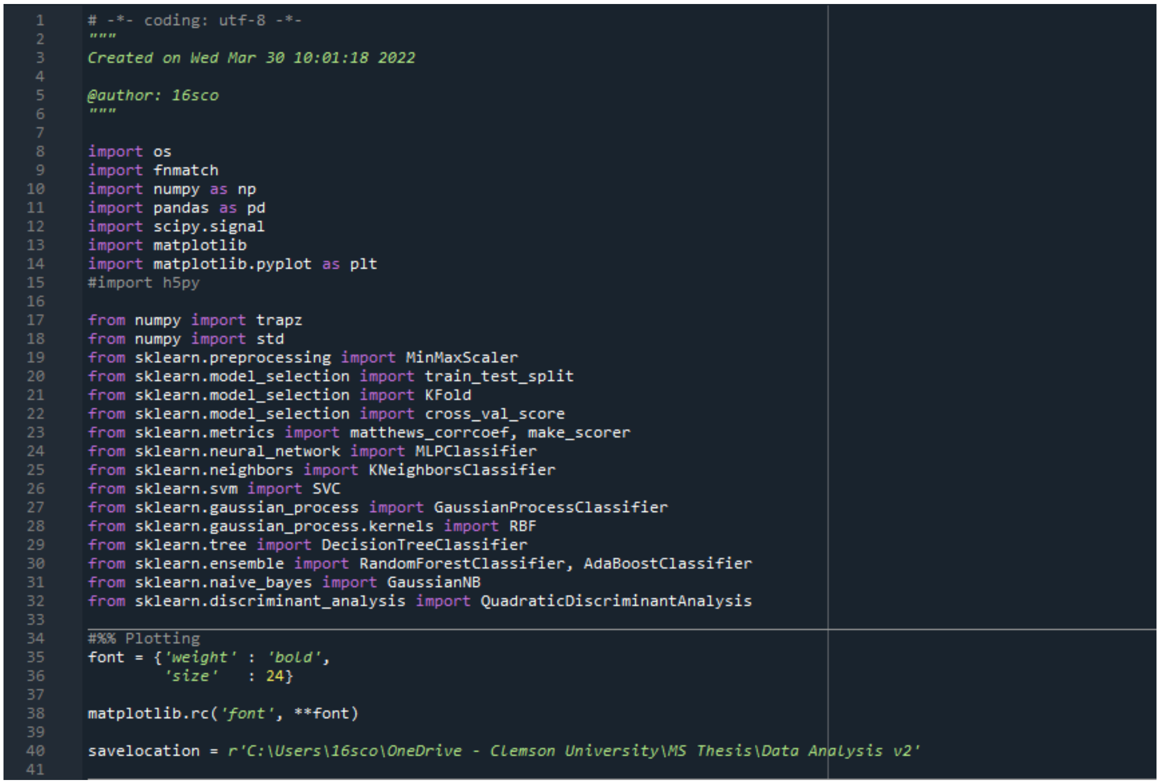


| Potential Failure Mode | Potential Effect(s) of Failure | Severity | Potential Cause/ Mechanism of Failure | Occurrence | Current Control Prevention(P)/ Detection (D) | Current Status | | To be tested |
|---|---|---|---|---|---|---|---|---|
| | | | | | | Detection | RPN | |
| Fluid Leakage | **Vehicle in inhouse Testing:**<br>- Downtime due to rework<br>- Cost of Non Quality<br>- Fluid loss<br>- Fire hazard<br>- Risk to Life<br><br>**Vehicle at customer end:**<br>- Inefficient performance<br>- Fluid loss<br>- Fire hazard<br>- Risk to Life<br>- Warranty claims<br>- Product Liability<br>- Loss of Goodwill / Harm to brand image<br>- Litigation from Government bodies | 9 | Not connected completely - The operator | 5 | Follow Standard operating procedure (SOP) | 6 | 270 | Y |
| | | 9 | Not connected completely - Less than required force was applied by the operator while making the connection | 5 | Follow Standard operating procedure (SOP) and Operator training (P)<br>Perform a Pull-Push check (D) | 6 | 270 | Y |
| | | 9 | False Positive push-pull test : Lower than required force applied while performing | 5 | | 5 | 225 | Y |
| | | 9 | Not connected completely - Angular misalignment | 7 | Follow Standard operating procedure (SOP) and Operator training (P)<br>Perform a Pull-Push check (D) | 3 | 189 | Y |
| | | 9 | Not connected completely - Linear offsett | 7 | Follow Standard operating procedure (SOP) and Operator training (P)<br>Perform a Pull-Push check (D) | 2 | 126 | Y |
| | | 9 | Not connected completely - Engagement mechanism developed a crack due to | 2 | Follow Standard operating procedure (SOP) and Operator training (P) | 6 | 108 | N |
| | | 9 | Not connected completely - The operator heard false positive click due to Noise | 5 | Monitor noise level (P) | 6 | 270 | N |
| | | 9 | Not connected completely - The operator missed the clicking sound | 5 | Follow Standard operating procedure (SOP) and Operator training (P) | 3 | 135 | N |
| Operation incomplete in target takt | **Rework is performed:**<br>- Assembly line stop | 1 | Operation incomplete - Angular | 6 | Follow Standard operating procedure (SOP) | 3 | 18 | Y |
| | | 1 | Operation incomplete - Linear offsett | 6 | Follow Standard operating procedure (SOP) | 2 | 12 | Y |
| | **Operation left incomplete:** | 1 | Operation incomplete - Operator did not hear the click and kept on trying due to | 4 | Monitor noise level (P) | 3 | 12 | N |

Figure 7: PFMEA for BMW connector Type 3 indicating severity, occurrence, and detection

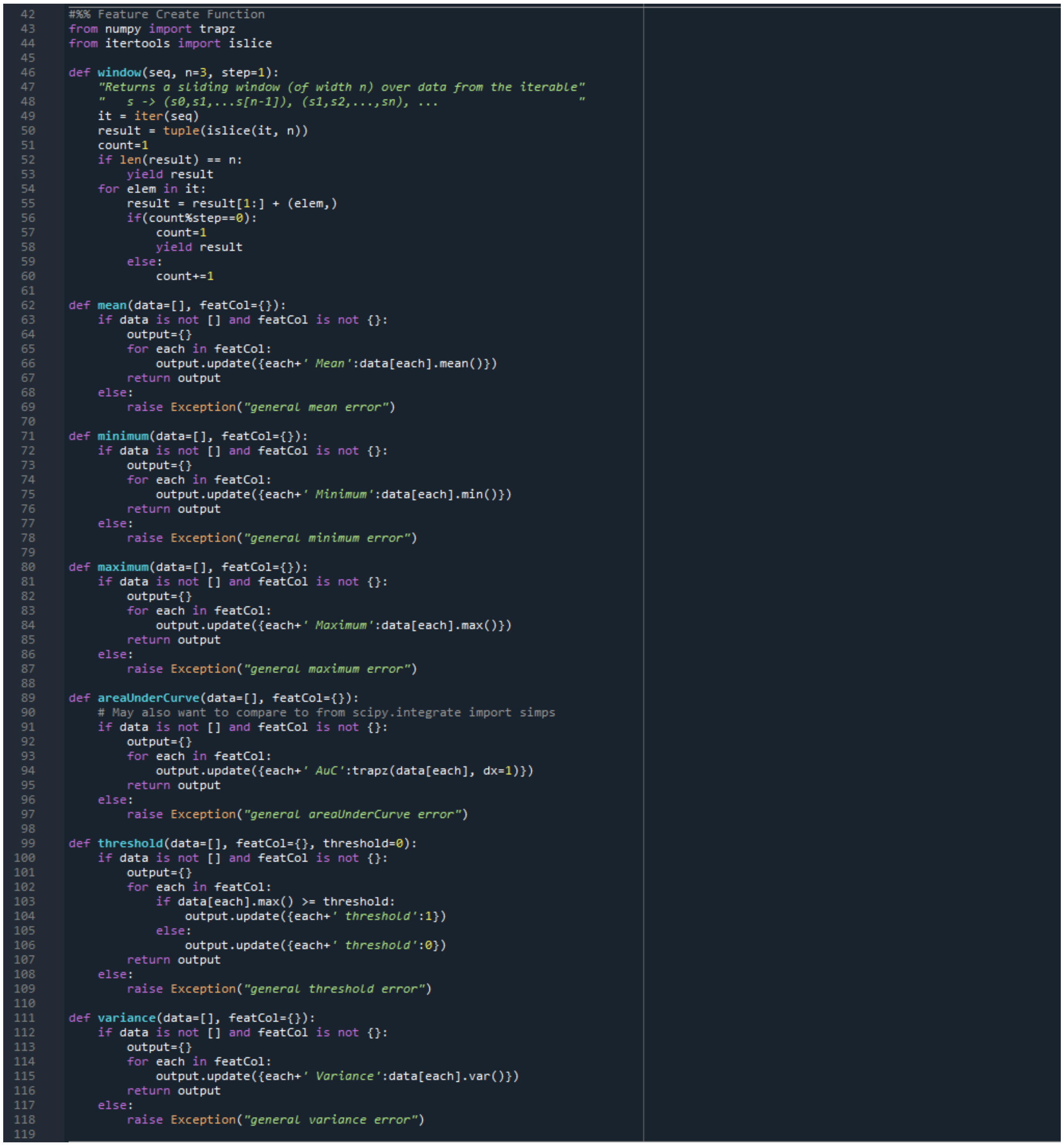
# Appendix B Data Classification Code

## B.1 Module Importing and Plotting



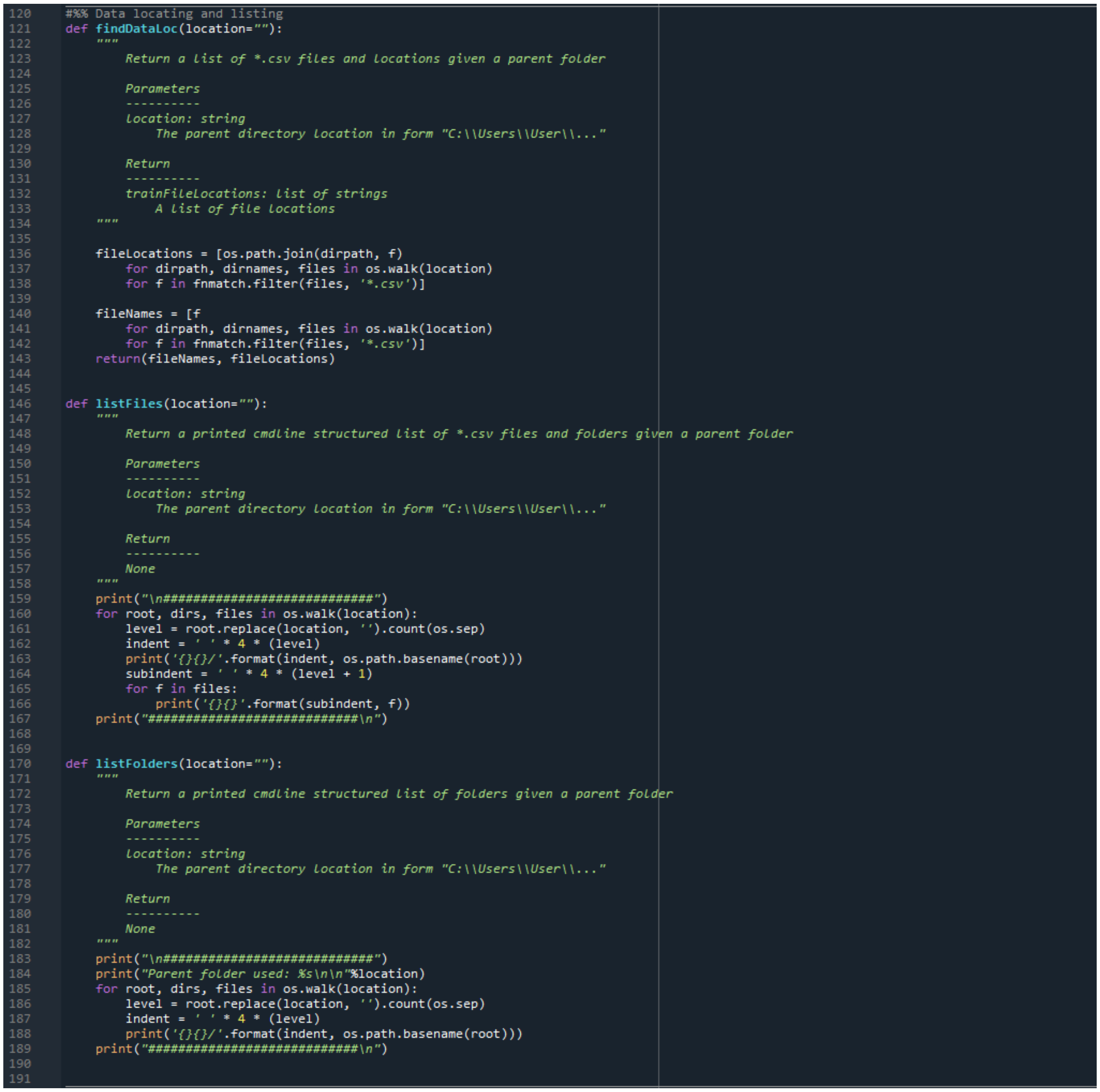
```python
# -*- coding: utf-8 -*-
"""
Created on Wed Mar 30 10:01:18 2022

@author: 16sco
"""

import os
import fnmatch
import numpy as np
import pandas as pd
import scipy.signal
import matplotlib
import matplotlib.pyplot as plt
#import h5py

from numpy import trapz
from numpy import std
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.metrics import matthews_corrcoef, make_scorer
from sklearn.neural_network import MLPClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.gaussian_process import GaussianProcessClassifier
from sklearn.gaussian_process.kernels import RBF
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis

#%% Plotting
font = {'weight' : 'bold',
        'size'   : 24}

matplotlib.rc('font', **font)

savelocation = r'C:\Users\16sco\OneDrive - Clemson University\MS Thesis\Data Analysis v2'

```

## B.2 Feature Create Function



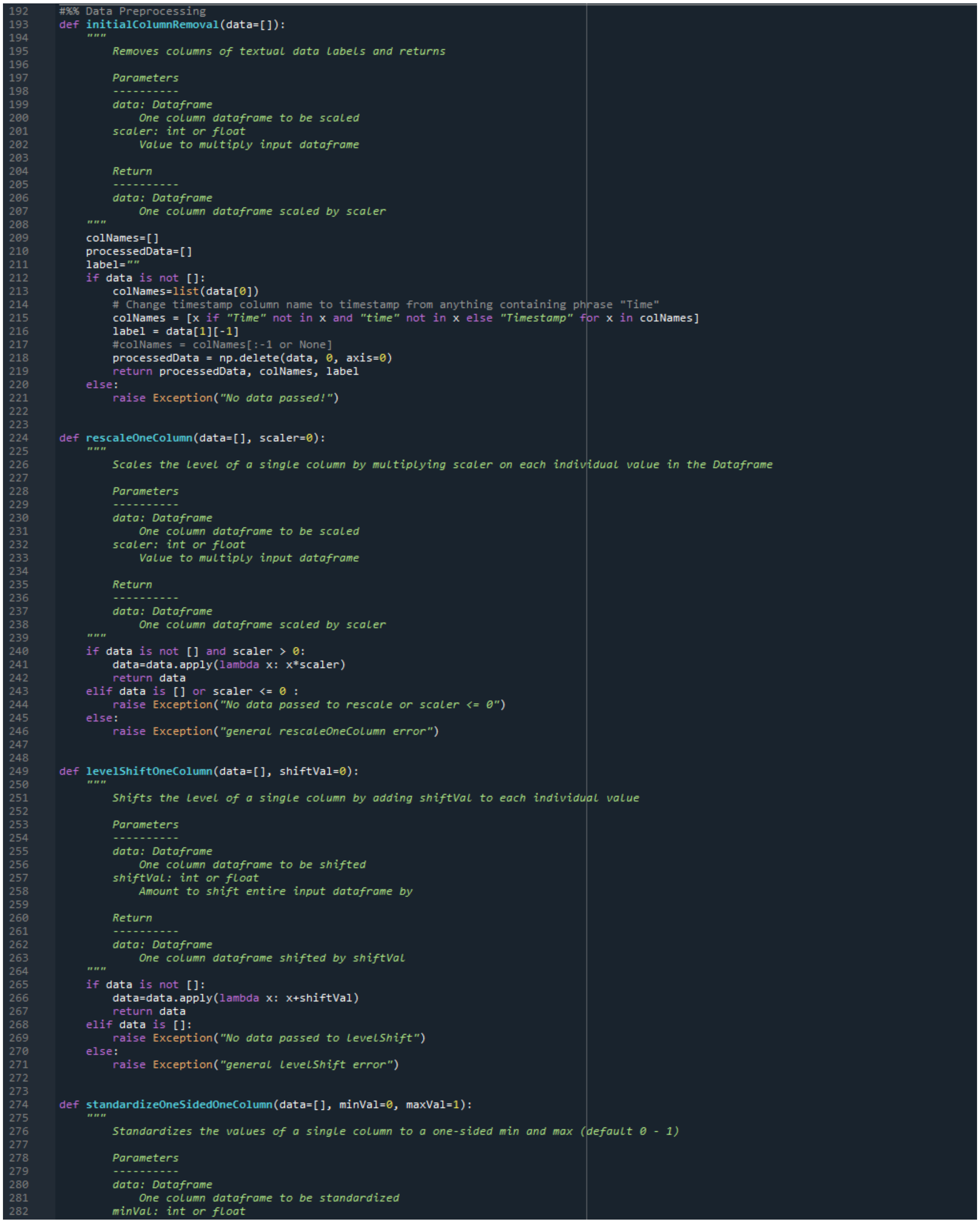
```
#%% Feature Create Function
from numpy import trapz
from itertools import islice

def window(seq, n=3, step=1):
    "Returns a sliding window (of width n) over data from the iterable"
    "   s -> (s0,s1,...s[n-1]), (s1,s2,...,sn), ...                    "
    it = iter(seq)
    result = tuple(islice(it, n))
    count=1
    if len(result) == n:
        yield result
    for elem in it:
        result = result[1:] + (elem,)
        if(count%step==0):
            count=1
            yield result
        else:
            count+=1

def mean(data=[], featCol={}):
    if data is not [] and featCol is not {}:
        output={}
        for each in featCol:
            output.update({each+' Mean':data[each].mean()})
        return output
    else:
        raise Exception("general mean error")

def minimum(data=[], featCol={}):
    if data is not [] and featCol is not {}:
        output={}
        for each in featCol:
            output.update({each+' Minimum':data[each].min()})
        return output
    else:
        raise Exception("general minimum error")

def maximum(data=[], featCol={}):
    if data is not [] and featCol is not {}:
        output={}
        for each in featCol:
            output.update({each+' Maximum':data[each].max()})
        return output
    else:
        raise Exception("general maximum error")

def areaUnderCurve(data=[], featCol={}):
    # May also want to compare to from scipy.integrate import simps
    if data is not [] and featCol is not {}:
        output={}
        for each in featCol:
            output.update({each+' AuC':trapz(data[each], dx=1)})
        return output
    else:
        raise Exception("general areaUnderCurve error")

def threshold(data=[], featCol={}, threshold=0):
    if data is not [] and featCol is not {}:
        output={}
        for each in featCol:
            if data[each].max() >= threshold:
                output.update({each+' threshold':1})
            else:
                output.update({each+' threshold':0})
        return output
    else:
        raise Exception("general threshold error")

def variance(data=[], featCol={}):
    if data is not [] and featCol is not {}:
        output={}
        for each in featCol:
            output.update({each+' Variance':data[each].var()})
        return output
    else:
        raise Exception("general variance error")
```

## B.3 Data Locating and Listing

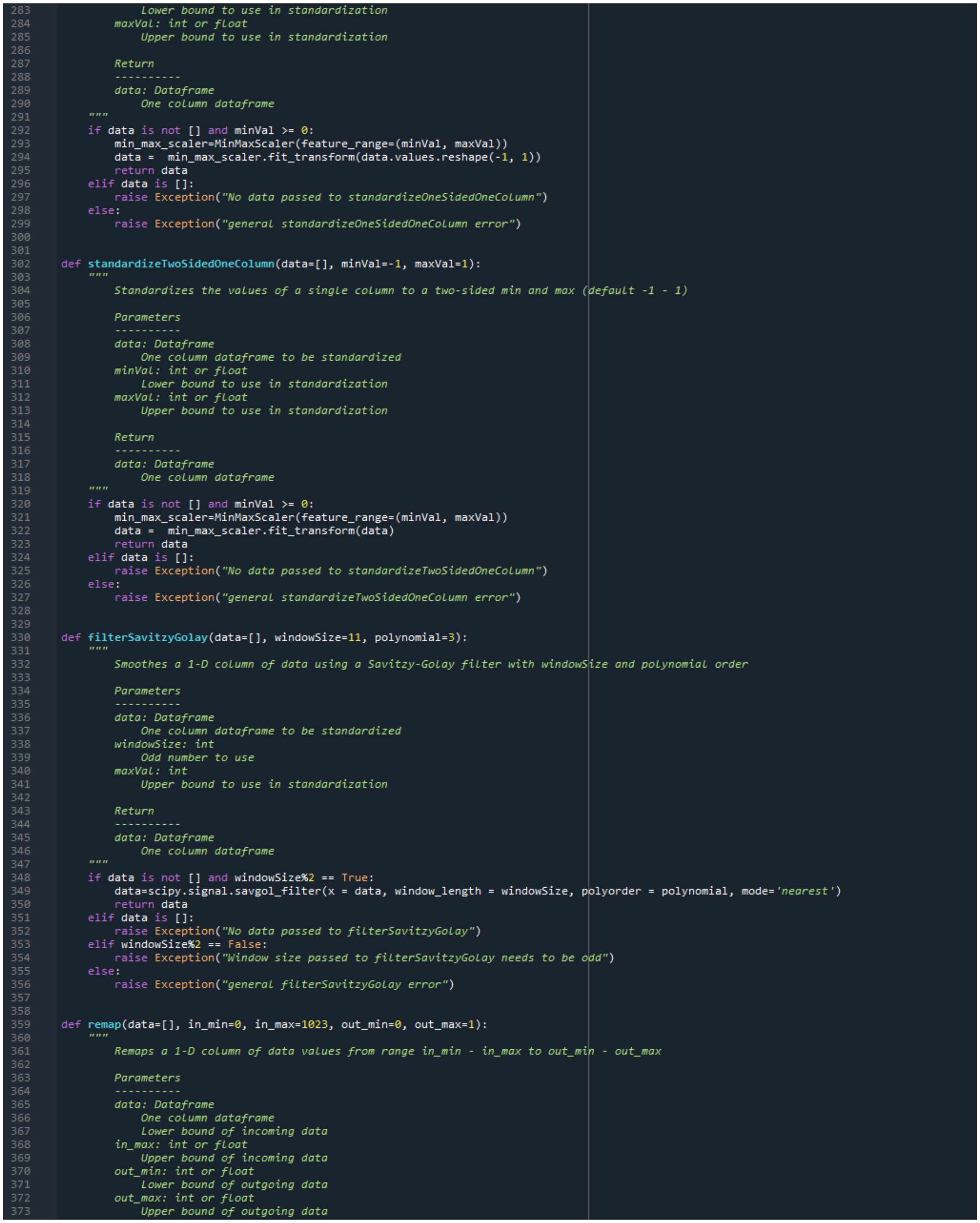


```
#%% Data locating and listing
def findDataLoc(location=""):
    """
        Return a list of *.csv files and locations given a parent folder

        Parameters
        ----------
        location: string
            The parent directory location in form "C:\\Users\\User\\..."

        Return
        ----------
        trainFileLocations: list of strings
            A list of file locations
    """

    fileLocations = [os.path.join(dirpath, f)
        for dirpath, dirnames, files in os.walk(location)
        for f in fnmatch.filter(files, '*.csv')]

    fileNames = [f
        for dirpath, dirnames, files in os.walk(location)
        for f in fnmatch.filter(files, '*.csv')]
    return(fileNames, fileLocations)


def listFiles(location=""):
    """
        Return a printed cmdline structured list of *.csv files and folders given a parent folder

        Parameters
        ----------
        location: string
            The parent directory location in form "C:\\Users\\User\\..."

        Return
        ----------
        None
    """
    print("\n############################")
    for root, dirs, files in os.walk(location):
        level = root.replace(location, '').count(os.sep)
        indent = ' ' * 4 * (level)
        print('{}{}/'.format(indent, os.path.basename(root)))
        subindent = ' ' * 4 * (level + 1)
        for f in files:
            print('{}{}'.format(subindent, f))
    print("############################\n")


def listFolders(location=""):
    """
        Return a printed cmdline structured list of folders given a parent folder

        Parameters
        ----------
        location: string
            The parent directory location in form "C:\\Users\\User\\..."

        Return
        ----------
        None
    """
    print("\n############################")
    print("Parent folder used: %s\n\n"%location)
    for root, dirs, files in os.walk(location):
        level = root.replace(location, '').count(os.sep)
        indent = ' ' * 4 * (level)
        print('{}{}/'.format(indent, os.path.basename(root)))
    print("############################\n")
```

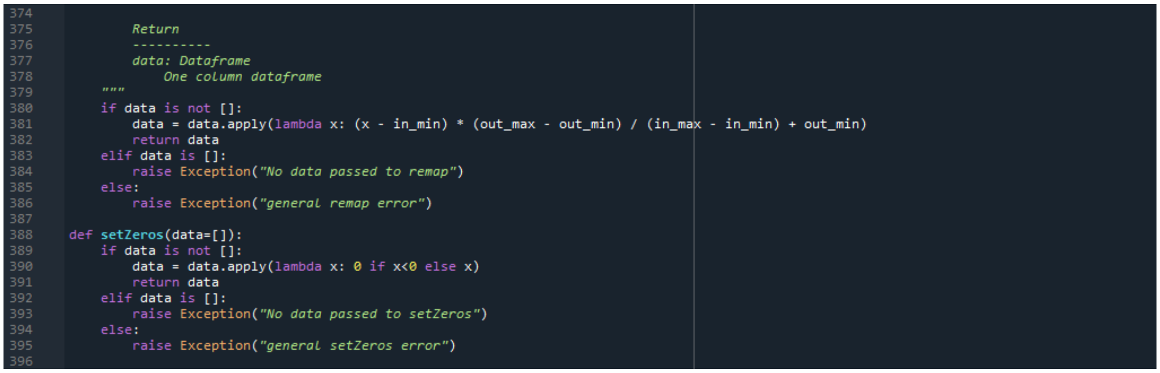
## B.4 Data Preprocessing



```python
#%% Data Preprocessing
def initialColumnRemoval(data=[]):
    """
        Removes columns of textual data labels and returns

        Parameters
        ----------
        data: Dataframe
            One column dataframe to be scaled
        scaler: int or float
            Value to multiply input dataframe

        Return
        ----------
        data: Dataframe
            One column dataframe scaled by scaler
    """
    colNames=[]
    processedData=[]
    label=""
    if data is not []:
        colNames=list(data[0])
        # Change timestamp column name to timestamp from anything containing phrase "Time"
        colNames = [x if "Time" not in x and "time" not in x else "Timestamp" for x in colNames]
        label = data[1][-1]
        #colNames = colNames[:-1 or None]
        processedData = np.delete(data, 0, axis=0)
        return processedData, colNames, label
    else:
        raise Exception("No data passed!")


def rescaleOneColumn(data=[], scaler=0):
    """
        Scales the level of a single column by multiplying scaler on each individual value in the Dataframe

        Parameters
        ----------
        data: Dataframe
            One column dataframe to be scaled
        scaler: int or float
            Value to multiply input dataframe

        Return
        ----------
        data: Dataframe
            One column dataframe scaled by scaler
    """
    if data is not [] and scaler > 0:
        data=data.apply(lambda x: x*scaler)
        return data
    elif data is [] or scaler <= 0 :
        raise Exception("No data passed to rescale or scaler <= 0")
    else:
        raise Exception("general rescaleOneColumn error")


def levelShiftOneColumn(data=[], shiftVal=0):
    """
        Shifts the level of a single column by adding shiftVal to each individual value

        Parameters
        ----------
        data: Dataframe
            One column dataframe to be shifted
        shiftVal: int or float
            Amount to shift entire input dataframe by

        Return
        ----------
        data: Dataframe
            One column dataframe shifted by shiftVal
    """
    if data is not []:
        data=data.apply(lambda x: x+shiftVal)
        return data
    elif data is []:
        raise Exception("No data passed to levelShift")
    else:
        raise Exception("general levelShift error")


def standardizeOneSidedOneColumn(data=[], minVal=0, maxVal=1):
    """
        Standardizes the values of a single column to a one-sided min and max (default 0 - 1)

        Parameters
        ----------
        data: Dataframe
            One column dataframe to be standardized
        minVal: int or float
```

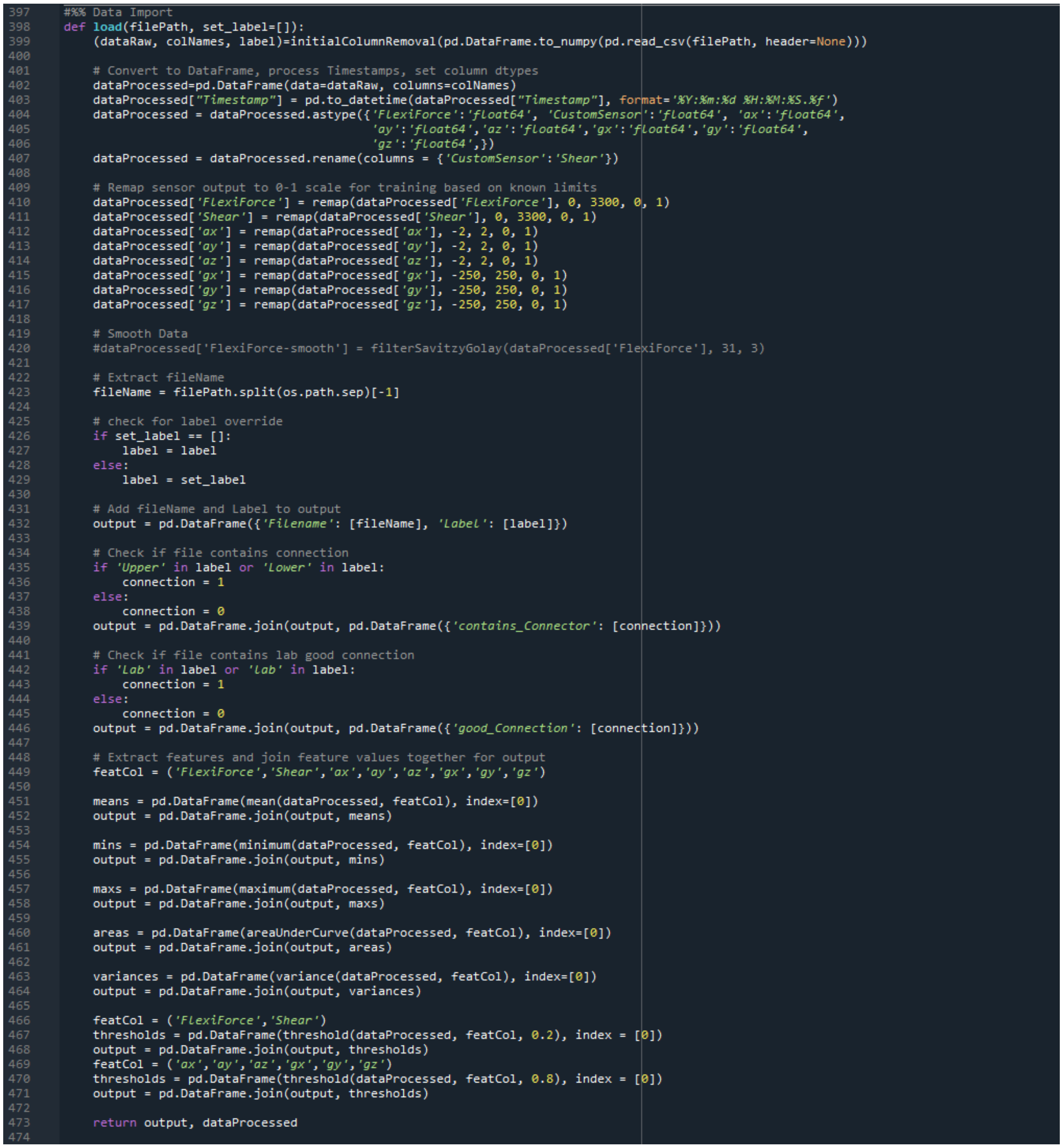


```
            Lower bound to use in standardization
        maxVal: int or float
            Upper bound to use in standardization

        Return
        ----------
        data: Dataframe
            One column dataframe
    """
    if data is not [] and minVal >= 0:
        min_max_scaler=MinMaxScaler(feature_range=(minVal, maxVal))
        data =  min_max_scaler.fit_transform(data.values.reshape(-1, 1))
        return data
    elif data is []:
        raise Exception("No data passed to standardizeOneSidedOneColumn")
    else:
        raise Exception("general standardizeOneSidedOneColumn error")


def standardizeTwoSidedOneColumn(data=[], minVal=-1, maxVal=1):
    """
        Standardizes the values of a single column to a two-sided min and max (default -1 - 1)

        Parameters
        ----------
        data: Dataframe
            One column dataframe to be standardized
        minVal: int or float
            Lower bound to use in standardization
        maxVal: int or float
            Upper bound to use in standardization

        Return
        ----------
        data: Dataframe
            One column dataframe
    """
    if data is not [] and minVal >= 0:
        min_max_scaler=MinMaxScaler(feature_range=(minVal, maxVal))
        data =  min_max_scaler.fit_transform(data)
        return data
    elif data is []:
        raise Exception("No data passed to standardizeTwoSidedOneColumn")
    else:
        raise Exception("general standardizeTwoSidedOneColumn error")


def filterSavitzyGolay(data=[], windowSize=11, polynomial=3):
    """
        Smoothes a 1-D column of data using a Savitzy-Golay filter with windowSize and polynomial order

        Parameters
        ----------
        data: Dataframe
            One column dataframe to be standardized
        windowSize: int
            Odd number to use
        maxVal: int
            Upper bound to use in standardization

        Return
        ----------
        data: Dataframe
            One column dataframe
    """
    if data is not [] and windowSize%2 == True:
        data=scipy.signal.savgol_filter(x = data, window_length = windowSize, polyorder = polynomial, mode='nearest')
        return data
    elif data is []:
        raise Exception("No data passed to filterSavitzyGolay")
    elif windowSize%2 == False:
        raise Exception("Window size passed to filterSavitzyGolay needs to be odd")
    else:
        raise Exception("general filterSavitzyGolay error")


def remap(data=[], in_min=0, in_max=1023, out_min=0, out_max=1):
    """
        Remaps a 1-D column of data values from range in_min - in_max to out_min - out_max

        Parameters
        ----------
        data: Dataframe
            One column dataframe
            Lower bound of incoming data
        in_max: int or float
            Upper bound of incoming data
        out_min: int or float
            Lower bound of outgoing data
        out_max: int or float
            Upper bound of outgoing data
```



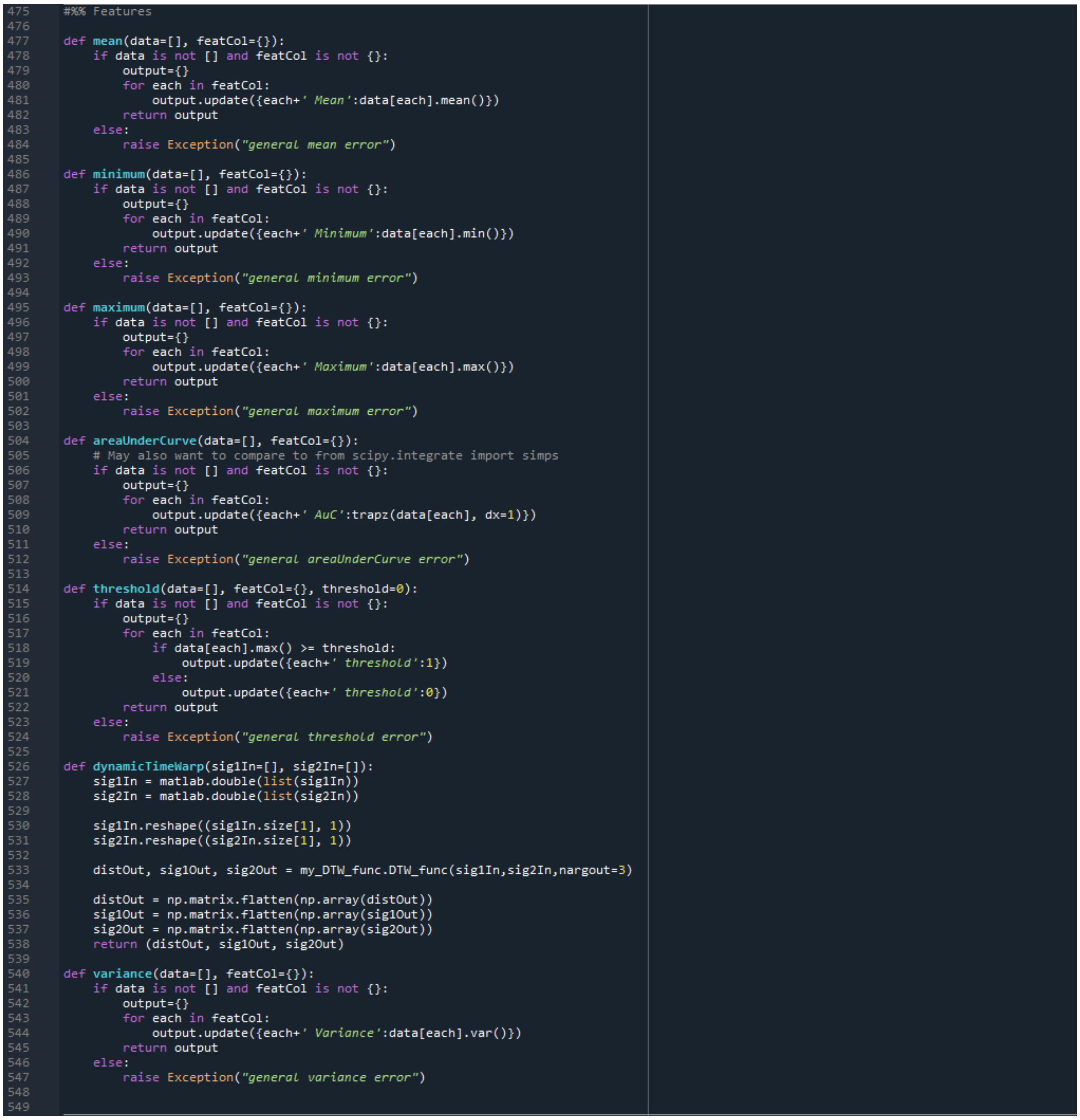
```
        Return
        ----------
        data: Dataframe
            One column dataframe
    """
    if data is not []:
        data = data.apply(lambda x: (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min)
        return data
    elif data is []:
        raise Exception("No data passed to remap")
    else:
        raise Exception("general remap error")

def setZeros(data=[]):
    if data is not []:
        data = data.apply(lambda x: 0 if x<0 else x)
        return data
    elif data is []:
        raise Exception("No data passed to setZeros")
    else:
        raise Exception("general setZeros error")
```

## B.5 Data Import



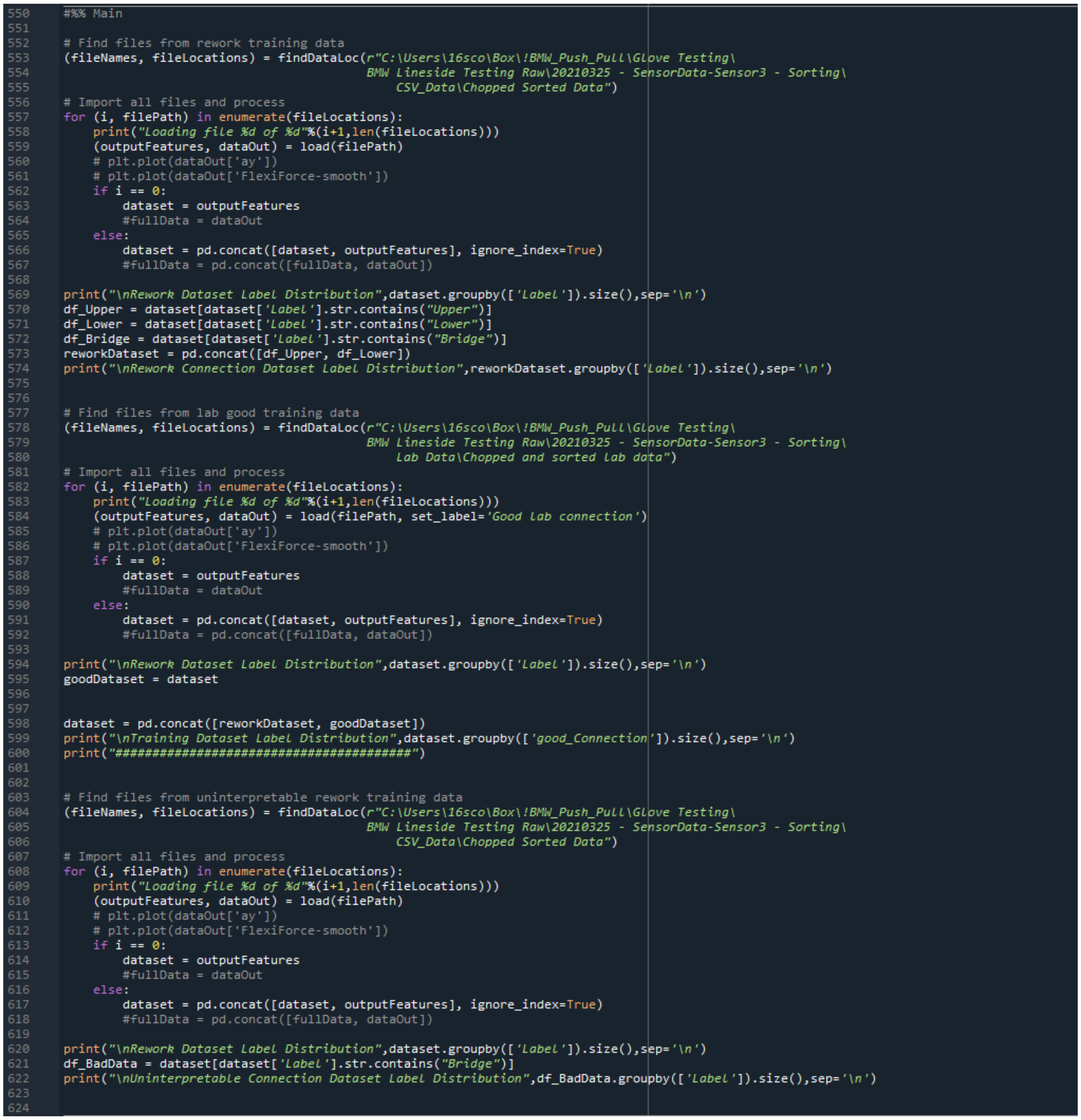
```python
#%% Data Import
def load(filePath, set_label=[]):
    (dataRaw, colNames, label)=initialColumnRemoval(pd.DataFrame.to_numpy(pd.read_csv(filePath, header=None)))

    # Convert to DataFrame, process Timestamps, set column dtypes
    dataProcessed=pd.DataFrame(data=dataRaw, columns=colNames)
    dataProcessed["Timestamp"] = pd.to_datetime(dataProcessed["Timestamp"], format='%Y:%m:%d %H:%M:%S.%f')
    dataProcessed = dataProcessed.astype({'FlexiForce':'float64', 'CustomSensor':'float64', 'ax':'float64',
                                          'ay':'float64','az':'float64','gx':'float64','gy':'float64',
                                          'gz':'float64',})
    dataProcessed = dataProcessed.rename(columns = {'CustomSensor':'Shear'})

    # Remap sensor output to 0-1 scale for training based on known limits
    dataProcessed['FlexiForce'] = remap(dataProcessed['FlexiForce'], 0, 3300, 0, 1)
    dataProcessed['Shear'] = remap(dataProcessed['Shear'], 0, 3300, 0, 1)
    dataProcessed['ax'] = remap(dataProcessed['ax'], -2, 2, 0, 1)
    dataProcessed['ay'] = remap(dataProcessed['ay'], -2, 2, 0, 1)
    dataProcessed['az'] = remap(dataProcessed['az'], -2, 2, 0, 1)
    dataProcessed['gx'] = remap(dataProcessed['gx'], -250, 250, 0, 1)
    dataProcessed['gy'] = remap(dataProcessed['gy'], -250, 250, 0, 1)
    dataProcessed['gz'] = remap(dataProcessed['gz'], -250, 250, 0, 1)

    # Smooth Data
    #dataProcessed['FlexiForce-smooth'] = filterSavitzyGolay(dataProcessed['FlexiForce'], 31, 3)

    # Extract fileName
    fileName = filePath.split(os.path.sep)[-1]

    # check for label override
    if set_label == []:
        label = label
    else:
        label = set_label

    # Add fileName and Label to output
    output = pd.DataFrame({'Filename': [fileName], 'Label': [label]})

    # Check if file contains connection
    if 'Upper' in label or 'Lower' in label:
        connection = 1
    else:
        connection = 0
    output = pd.DataFrame.join(output, pd.DataFrame({'contains_Connector': [connection]}))

    # Check if file contains lab good connection
    if 'Lab' in label or 'lab' in label:
        connection = 1
    else:
        connection = 0
    output = pd.DataFrame.join(output, pd.DataFrame({'good_Connection': [connection]}))

    # Extract features and join feature values together for output
    featCol = ('FlexiForce','Shear','ax','ay','az','gx','gy','gz')

    means = pd.DataFrame(mean(dataProcessed, featCol), index=[0])
    output = pd.DataFrame.join(output, means)

    mins = pd.DataFrame(minimum(dataProcessed, featCol), index=[0])
    output = pd.DataFrame.join(output, mins)

    maxs = pd.DataFrame(maximum(dataProcessed, featCol), index=[0])
    output = pd.DataFrame.join(output, maxs)

    areas = pd.DataFrame(areaUnderCurve(dataProcessed, featCol), index=[0])
    output = pd.DataFrame.join(output, areas)

    variances = pd.DataFrame(variance(dataProcessed, featCol), index=[0])
    output = pd.DataFrame.join(output, variances)

    featCol = ('FlexiForce','Shear')
    thresholds = pd.DataFrame(threshold(dataProcessed, featCol, 0.2), index = [0])
    output = pd.DataFrame.join(output, thresholds)
    featCol = ('ax','ay','az','gx','gy','gz')
    thresholds = pd.DataFrame(threshold(dataProcessed, featCol, 0.8), index = [0])
    output = pd.DataFrame.join(output, thresholds)

    return output, dataProcessed

```

## B.6 Feature Extraction



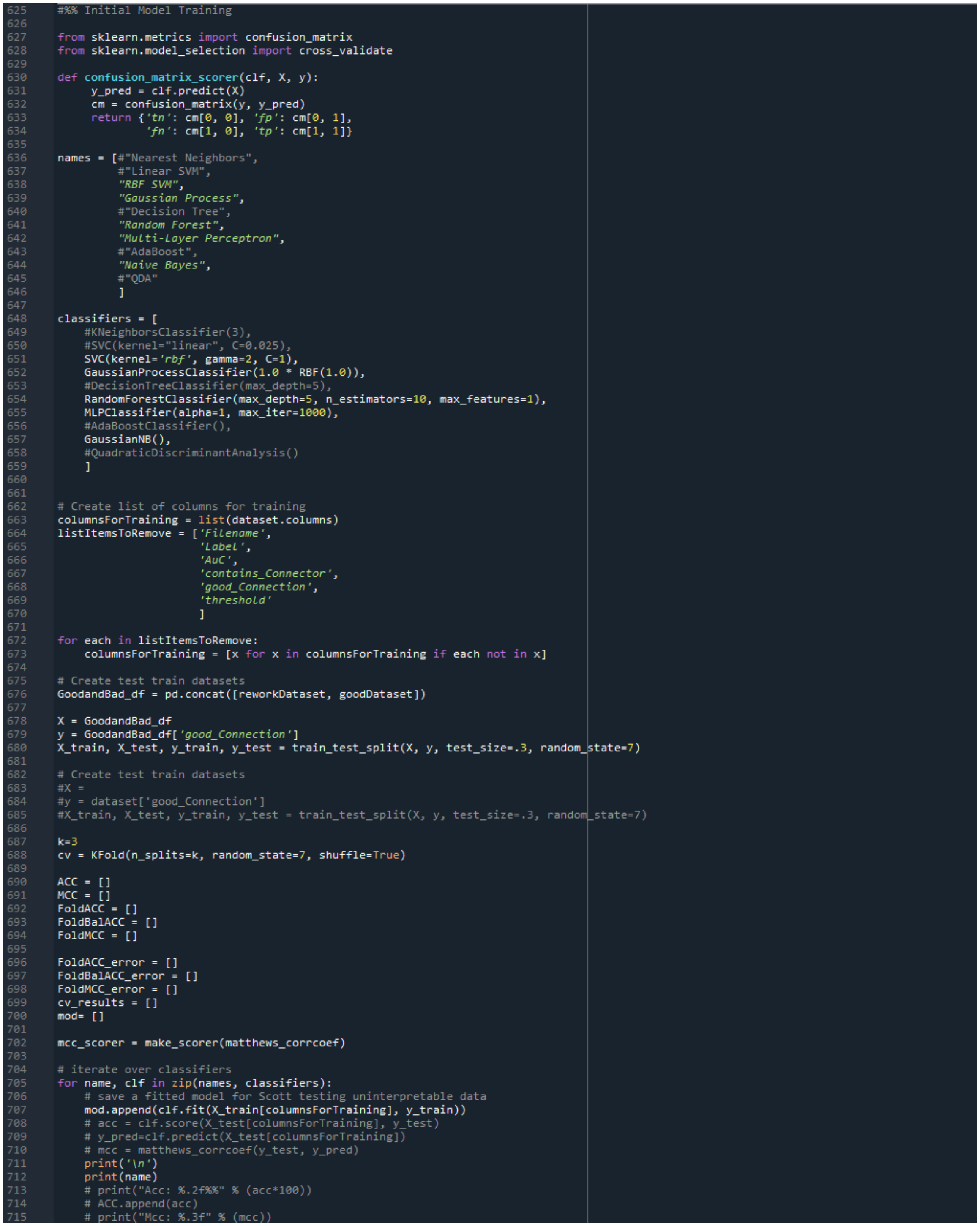
```python
#%% Features

def mean(data=[], featCol={}):
    if data is not [] and featCol is not {}:
        output={}
        for each in featCol:
            output.update({each+' Mean':data[each].mean()})
        return output
    else:
        raise Exception("general mean error")

def minimum(data=[], featCol={}):
    if data is not [] and featCol is not {}:
        output={}
        for each in featCol:
            output.update({each+' Minimum':data[each].min()})
        return output
    else:
        raise Exception("general minimum error")

def maximum(data=[], featCol={}):
    if data is not [] and featCol is not {}:
        output={}
        for each in featCol:
            output.update({each+' Maximum':data[each].max()})
        return output
    else:
        raise Exception("general maximum error")

def areaUnderCurve(data=[], featCol={}):
    # May also want to compare to from scipy.integrate import simps
    if data is not [] and featCol is not {}:
        output={}
        for each in featCol:
            output.update({each+' AuC':trapz(data[each], dx=1)})
        return output
    else:
        raise Exception("general areaUnderCurve error")

def threshold(data=[], featCol={}, threshold=0):
    if data is not [] and featCol is not {}:
        output={}
        for each in featCol:
            if data[each].max() >= threshold:
                output.update({each+' threshold':1})
            else:
                output.update({each+' threshold':0})
        return output
    else:
        raise Exception("general threshold error")

def dynamicTimeWarp(sig1In=[], sig2In=[]):
    sig1In = matlab.double(list(sig1In))
    sig2In = matlab.double(list(sig2In))

    sig1In.reshape((sig1In.size[1], 1))
    sig2In.reshape((sig2In.size[1], 1))

    distOut, sig1Out, sig2Out = my_DTW_func.DTW_func(sig1In,sig2In,nargout=3)

    distOut = np.matrix.flatten(np.array(distOut))
    sig1Out = np.matrix.flatten(np.array(sig1Out))
    sig2Out = np.matrix.flatten(np.array(sig2Out))
    return (distOut, sig1Out, sig2Out)

def variance(data=[], featCol={}):
    if data is not [] and featCol is not {}:
        output={}
        for each in featCol:
            output.update({each+' Variance':data[each].var()})
        return output
    else:
        raise Exception("general variance error")
```

## B.7 Main Function

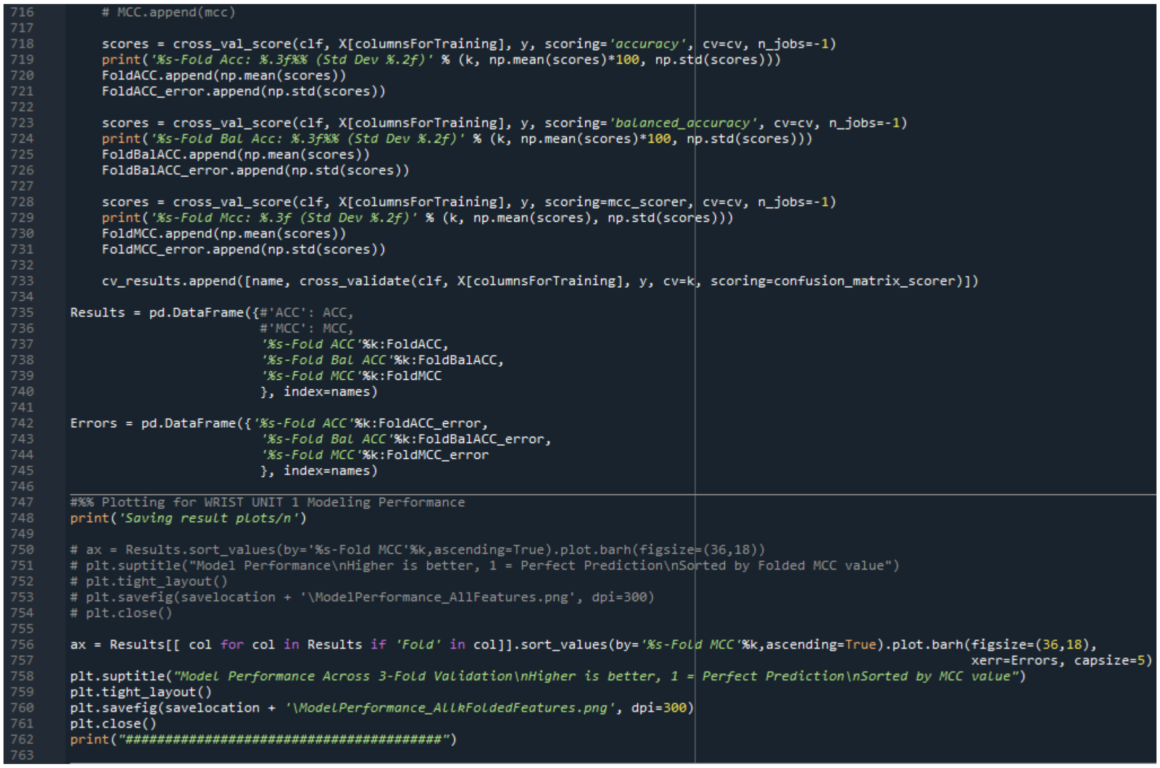


```python
#%% Main

# Find files from rework training data
(fileNames, fileLocations) = findDataLoc(r"C:\Users\16sco\Box\!BMW_Push_Pull\Glove Testing\
                                         BMW Lineside Testing Raw\20210325 - SensorData-Sensor3 - Sorting\
                                             CSV_Data\Chopped Sorted Data")
# Import all files and process
for (i, filePath) in enumerate(fileLocations):
    print("Loading file %d of %d"%(i+1,len(fileLocations)))
    (outputFeatures, dataOut) = load(filePath)
    # plt.plot(dataOut['ay'])
    # plt.plot(dataOut['FlexiForce-smooth'])
    if i == 0:
        dataset = outputFeatures
        #fullData = dataOut
    else:
        dataset = pd.concat([dataset, outputFeatures], ignore_index=True)
        #fullData = pd.concat([fullData, dataOut])

print("\nRework Dataset Label Distribution",dataset.groupby(['Label']).size(),sep='\n')
df_Upper = dataset[dataset['Label'].str.contains("Upper")]
df_Lower = dataset[dataset['Label'].str.contains("Lower")]
df_Bridge = dataset[dataset['Label'].str.contains("Bridge")]
reworkDataset = pd.concat([df_Upper, df_Lower])
print("\nRework Connection Dataset Label Distribution",reworkDataset.groupby(['Label']).size(),sep='\n')


# Find files from lab good training data
(fileNames, fileLocations) = findDataLoc(r"C:\Users\16sco\Box\!BMW_Push_Pull\Glove Testing\
                                         BMW Lineside Testing Raw\20210325 - SensorData-Sensor3 - Sorting\
                                             Lab Data\Chopped and sorted lab data")
# Import all files and process
for (i, filePath) in enumerate(fileLocations):
    print("Loading file %d of %d"%(i+1,len(fileLocations)))
    (outputFeatures, dataOut) = load(filePath, set_label='Good Lab connection')
    # plt.plot(dataOut['ay'])
    # plt.plot(dataOut['FlexiForce-smooth'])
    if i == 0:
        dataset = outputFeatures
        #fullData = dataOut
    else:
        dataset = pd.concat([dataset, outputFeatures], ignore_index=True)
        #fullData = pd.concat([fullData, dataOut])

print("\nRework Dataset Label Distribution",dataset.groupby(['Label']).size(),sep='\n')
goodDataset = dataset


dataset = pd.concat([reworkDataset, goodDataset])
print("\nTraining Dataset Label Distribution",dataset.groupby(['good_Connection']).size(),sep='\n')
print("########################################")


# Find files from uninterpretable rework training data
(fileNames, fileLocations) = findDataLoc(r"C:\Users\16sco\Box\!BMW_Push_Pull\Glove Testing\
                                         BMW Lineside Testing Raw\20210325 - SensorData-Sensor3 - Sorting\
                                             CSV_Data\Chopped Sorted Data")
# Import all files and process
for (i, filePath) in enumerate(fileLocations):
    print("Loading file %d of %d"%(i+1,len(fileLocations)))
    (outputFeatures, dataOut) = load(filePath)
    # plt.plot(dataOut['ay'])
    # plt.plot(dataOut['FlexiForce-smooth'])
    if i == 0:
        dataset = outputFeatures
        #fullData = dataOut
    else:
        dataset = pd.concat([dataset, outputFeatures], ignore_index=True)
        #fullData = pd.concat([fullData, dataOut])

print("\nRework Dataset Label Distribution",dataset.groupby(['Label']).size(),sep='\n')
df_BadData = dataset[dataset['Label'].str.contains("Bridge")]
print("\nUninterpretable Connection Dataset Label Distribution",df_BadData.groupby(['Label']).size(),sep='\n')
```

## B.8 Initial Model Training

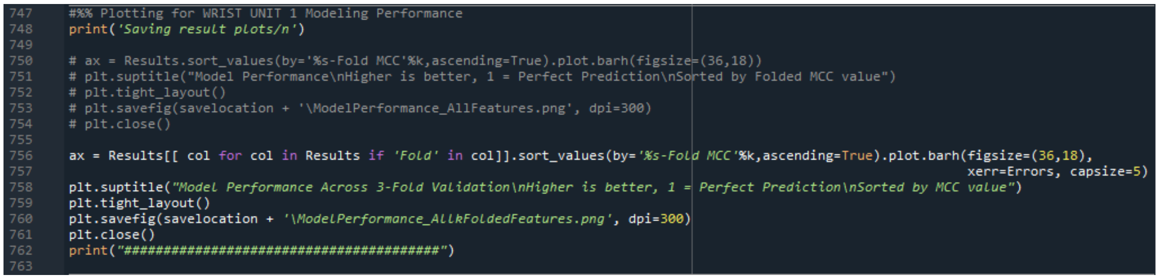


```python
#%% Initial Model Training

from sklearn.metrics import confusion_matrix
from sklearn.model_selection import cross_validate

def confusion_matrix_scorer(clf, X, y):
     y_pred = clf.predict(X)
     cm = confusion_matrix(y, y_pred)
     return {'tn': cm[0, 0], 'fp': cm[0, 1],
             'fn': cm[1, 0], 'tp': cm[1, 1]}

names = [#"Nearest Neighbors",
         #"Linear SVM",
         "RBF SVM",
         "Gaussian Process",
         #"Decision Tree",
         "Random Forest",
         "Multi-Layer Perceptron",
         #"AdaBoost",
         "Naive Bayes",
         #"QDA"
         ]

classifiers = [
    #KNeighborsClassifier(3),
    #SVC(kernel="linear", C=0.025),
    SVC(kernel='rbf', gamma=2, C=1),
    GaussianProcessClassifier(1.0 * RBF(1.0)),
    #DecisionTreeClassifier(max_depth=5),
    RandomForestClassifier(max_depth=5, n_estimators=10, max_features=1),
    MLPClassifier(alpha=1, max_iter=1000),
    #AdaBoostClassifier(),
    GaussianNB(),
    #QuadraticDiscriminantAnalysis()
    ]


# Create list of columns for training
columnsForTraining = list(dataset.columns)
listItemsToRemove = ['Filename',
                     'Label',
                     'AuC',
                     'contains_Connector',
                     'good_Connection',
                     'threshold'
                     ]

for each in listItemsToRemove:
    columnsForTraining = [x for x in columnsForTraining if each not in x]

# Create test train datasets
GoodandBad_df = pd.concat([reworkDataset, goodDataset])

X = GoodandBad_df
y = GoodandBad_df['good_Connection']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.3, random_state=7)

# Create test train datasets
#X =
#y = dataset['good_Connection']
#X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.3, random_state=7)

k=3
cv = KFold(n_splits=k, random_state=7, shuffle=True)

ACC = []
MCC = []
FoldACC = []
FoldBalACC = []
FoldMCC = []

FoldACC_error = []
FoldBalACC_error = []
FoldMCC_error = []
cv_results = []
mod= []

mcc_scorer = make_scorer(matthews_corrcoef)

# iterate over classifiers
for name, clf in zip(names, classifiers):
    # save a fitted model for Scott testing uninterpretable data
    mod.append(clf.fit(X_train[columnsForTraining], y_train))
    # acc = clf.score(X_test[columnsForTraining], y_test)
    # y_pred=clf.predict(X_test[columnsForTraining])
    # mcc = matthews_corrcoef(y_test, y_pred)
    print('\n')
    print(name)
    # print("Acc: %.2f%%" % (acc*100))
    # ACC.append(acc)
    # print("Mcc: %.3f" % (mcc))
```



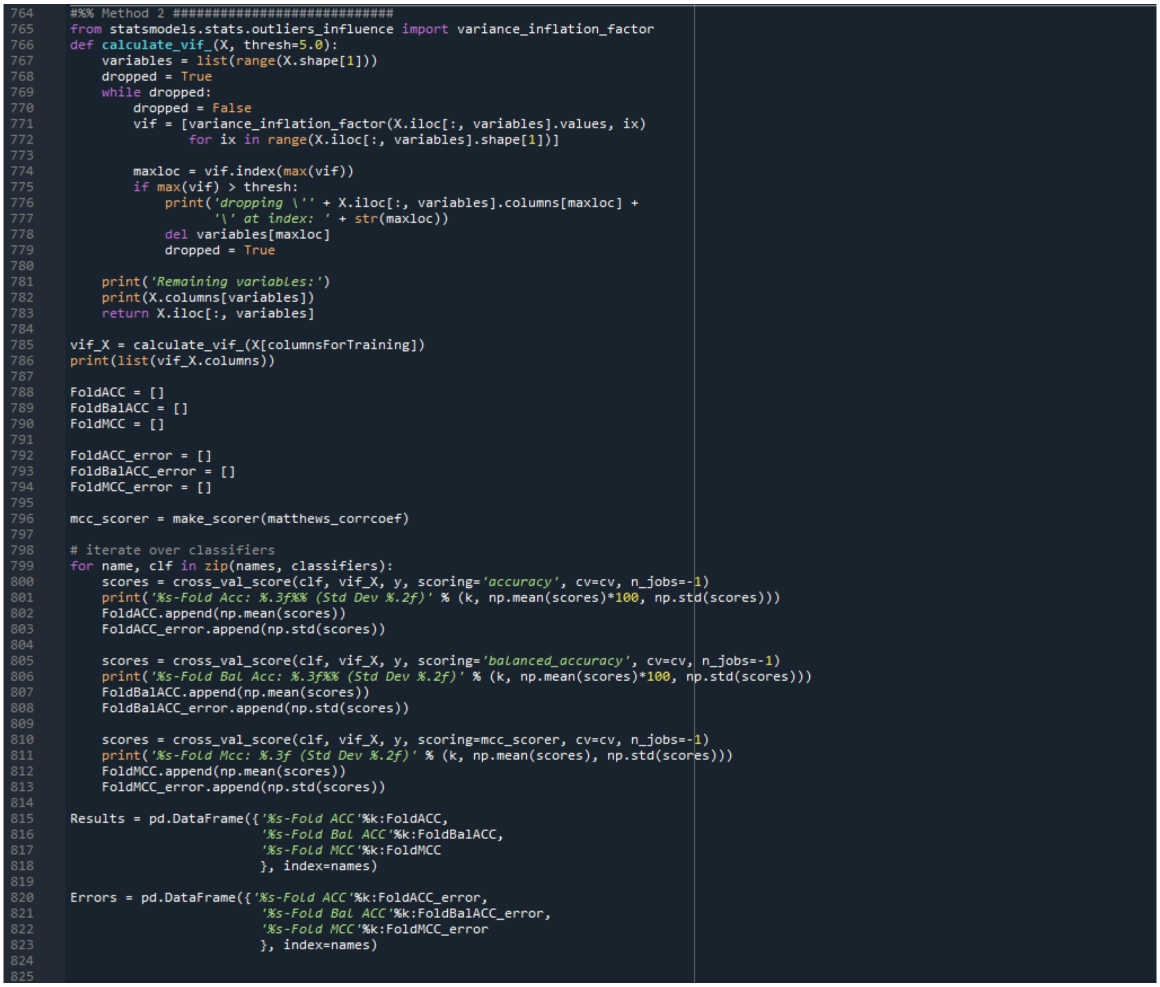
```python
    # MCC.append(mcc)

    scores = cross_val_score(clf, X[columnsForTraining], y, scoring='accuracy', cv=cv, n_jobs=-1)
    print('%s-Fold Acc: %.3f%% (Std Dev %.2f)' % (k, np.mean(scores)*100, np.std(scores)))
    FoldACC.append(np.mean(scores))
    FoldACC_error.append(np.std(scores))

    scores = cross_val_score(clf, X[columnsForTraining], y, scoring='balanced_accuracy', cv=cv, n_jobs=-1)
    print('%s-Fold Bal Acc: %.3f%% (Std Dev %.2f)' % (k, np.mean(scores)*100, np.std(scores)))
    FoldBalACC.append(np.mean(scores))
    FoldBalACC_error.append(np.std(scores))

    scores = cross_val_score(clf, X[columnsForTraining], y, scoring=mcc_scorer, cv=cv, n_jobs=-1)
    print('%s-Fold Mcc: %.3f (Std Dev %.2f)' % (k, np.mean(scores), np.std(scores)))
    FoldMCC.append(np.mean(scores))
    FoldMCC_error.append(np.std(scores))

    cv_results.append([name, cross_validate(clf, X[columnsForTraining], y, cv=k, scoring=confusion_matrix_scorer)])

Results = pd.DataFrame({#'ACC': ACC,
                        #'MCC': MCC,
                        '%s-Fold ACC'%k:FoldACC,
                        '%s-Fold Bal ACC'%k:FoldBalACC,
                        '%s-Fold MCC'%k:FoldMCC
                        }, index=names)

Errors = pd.DataFrame({'%s-Fold ACC'%k:FoldACC_error,
                        '%s-Fold Bal ACC'%k:FoldBalACC_error,
                        '%s-Fold MCC'%k:FoldMCC_error
                        }, index=names)

#%% Plotting for WRIST UNIT 1 Modeling Performance
print('Saving result plots/n')

# ax = Results.sort_values(by='%s-Fold MCC'%k,ascending=True).plot.barh(figsize=(36,18))
# plt.suptitle("Model Performance\nHigher is better, 1 = Perfect Prediction\nSorted by Folded MCC value")
# plt.tight_layout()
# plt.savefig(savelocation + '\ModelPerformance_AllFeatures.png', dpi=300)
# plt.close()

ax = Results[[ col for col in Results if 'Fold' in col]].sort_values(by='%s-Fold MCC'%k,ascending=True).plot.barh(figsize=(36,18),
                                                                                                                  xerr=Errors, capsize=5)
plt.suptitle("Model Performance Across 3-Fold Validation\nHigher is better, 1 = Perfect Prediction\nSorted by MCC value")
plt.tight_layout()
plt.savefig(savelocation + '\ModelPerformance_AllkFoldedFeatures.png', dpi=300)
plt.close()
print("##########################################")
```

## B.9 Plotting for Wrist Unit 1 Model Performance - Method 1



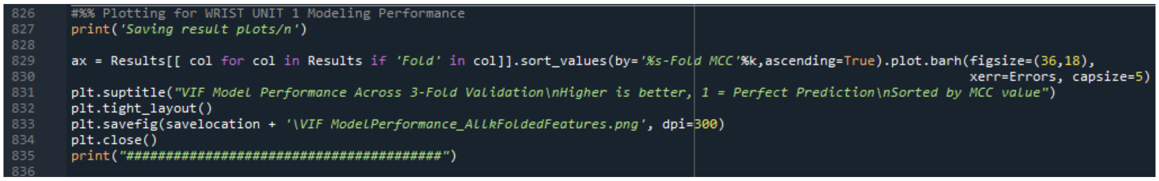
```python
#%% Plotting for WRIST UNIT 1 Modeling Performance
print('Saving result plots/n')

# ax = Results.sort_values(by='%s-Fold MCC'%k,ascending=True).plot.barh(figsize=(36,18))
# plt.suptitle("Model Performance\nHigher is better, 1 = Perfect Prediction\nSorted by Folded MCC value")
# plt.tight_layout()
# plt.savefig(savelocation + '\ModelPerformance_AllFeatures.png', dpi=300)
# plt.close()

ax = Results[[ col for col in Results if 'Fold' in col]].sort_values(by='%s-Fold MCC'%k,ascending=True).plot.barh(figsize=(36,18),
                                                                                                                  xerr=Errors, capsize=5)
plt.suptitle("Model Performance Across 3-Fold Validation\nHigher is better, 1 = Perfect Prediction\nSorted by MCC value")
plt.tight_layout()
plt.savefig(savelocation + '\ModelPerformance_AllkFoldedFeatures.png', dpi=300)
plt.close()
print("##########################################")
```

## B.10 Classification Method 2



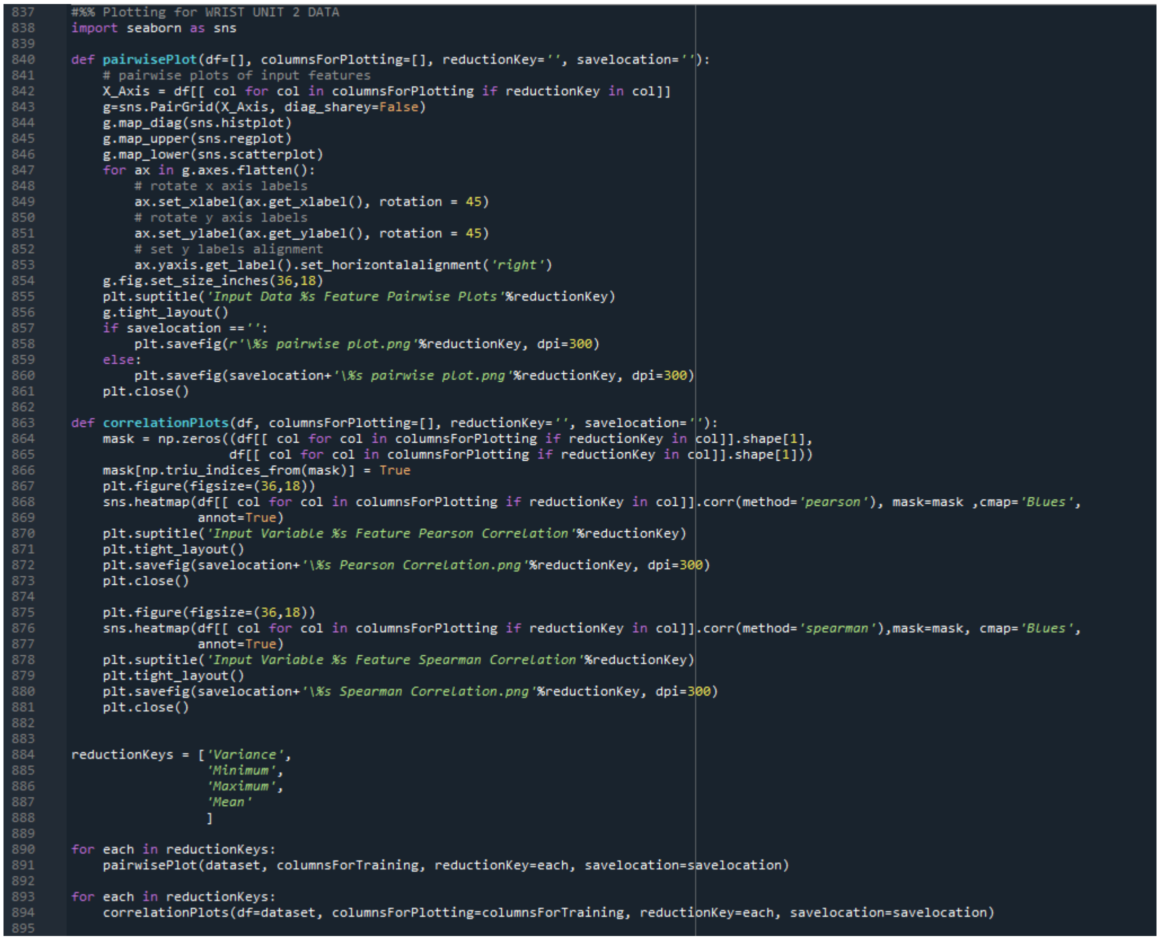
```python
#%% Method 2 ############################
from statsmodels.stats.outliers_influence import variance_inflation_factor
def calculate_vif_(X, thresh=5.0):
    variables = list(range(X.shape[1]))
    dropped = True
    while dropped:
        dropped = False
        vif = [variance_inflation_factor(X.iloc[:, variables].values, ix)
               for ix in range(X.iloc[:, variables].shape[1])]

        maxloc = vif.index(max(vif))
        if max(vif) > thresh:
            print('dropping \'' + X.iloc[:, variables].columns[maxloc] +
                  '\' at index: ' + str(maxloc))
            del variables[maxloc]
            dropped = True

    print('Remaining variables:')
    print(X.columns[variables])
    return X.iloc[:, variables]

vif_X = calculate_vif_(X[columnsForTraining])
print(list(vif_X.columns))

FoldACC = []
FoldBalACC = []
FoldMCC = []

FoldACC_error = []
FoldBalACC_error = []
FoldMCC_error = []

mcc_scorer = make_scorer(matthews_corrcoef)

# iterate over classifiers
for name, clf in zip(names, classifiers):
    scores = cross_val_score(clf, vif_X, y, scoring='accuracy', cv=cv, n_jobs=-1)
    print('%s-Fold Acc: %.3f%% (Std Dev %.2f)' % (k, np.mean(scores)*100, np.std(scores)))
    FoldACC.append(np.mean(scores))
    FoldACC_error.append(np.std(scores))

    scores = cross_val_score(clf, vif_X, y, scoring='balanced_accuracy', cv=cv, n_jobs=-1)
    print('%s-Fold Bal Acc: %.3f%% (Std Dev %.2f)' % (k, np.mean(scores)*100, np.std(scores)))
    FoldBalACC.append(np.mean(scores))
    FoldBalACC_error.append(np.std(scores))

    scores = cross_val_score(clf, vif_X, y, scoring=mcc_scorer, cv=cv, n_jobs=-1)
    print('%s-Fold Mcc: %.3f (Std Dev %.2f)' % (k, np.mean(scores), np.std(scores)))
    FoldMCC.append(np.mean(scores))
    FoldMCC_error.append(np.std(scores))

Results = pd.DataFrame({'%s-Fold ACC'%k:FoldACC,
                        '%s-Fold Bal ACC'%k:FoldBalACC,
                        '%s-Fold MCC'%k:FoldMCC
                        }, index=names)

Errors = pd.DataFrame({'%s-Fold ACC'%k:FoldACC_error,
                       '%s-Fold Bal ACC'%k:FoldBalACC_error,
                       '%s-Fold MCC'%k:FoldMCC_error
                       }, index=names)

```

## B.11 Plotting for Wrist Unit 1 Model Performance - Method 2



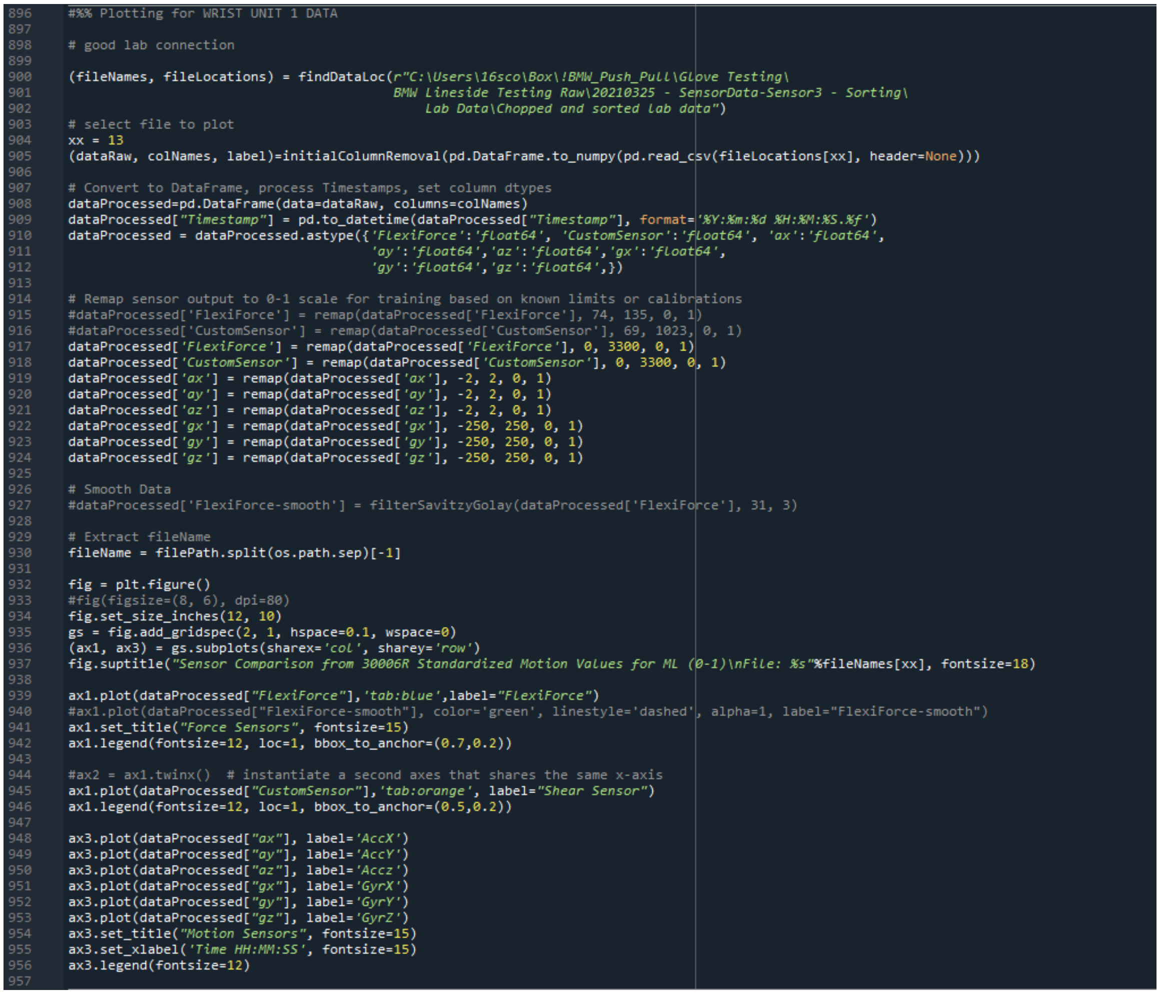
```python
#%% Plotting for WRIST UNIT 1 Modeling Performance
print('Saving result plots/n')

ax = Results[[ col for col in Results if 'Fold' in col]].sort_values(by='%s-Fold MCC'%k,ascending=True).plot.barh(figsize=(36,18),
                                                                                                                  xerr=Errors, capsize=5)
plt.suptitle("VIF Model Performance Across 3-Fold Validation\nHigher is better, 1 = Perfect Prediction\nSorted by MCC value")
plt.tight_layout()
plt.savefig(savelocation + '\VIF ModelPerformance_AllkFoldedFeatures.png', dpi=300)
plt.close()
print("########################################")

```

## B.12 Plotting for Wrist Unit 2 Data



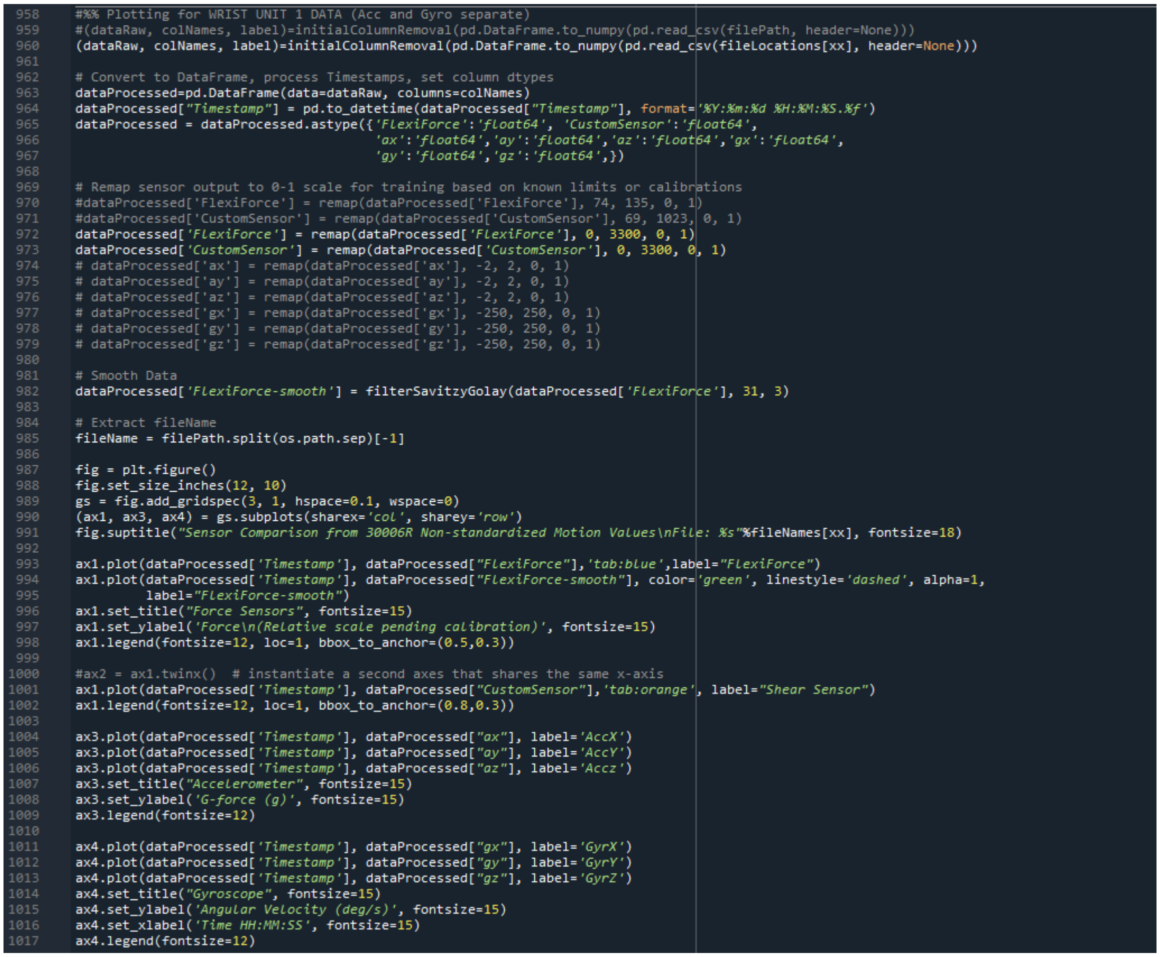
```python
#%% Plotting for WRIST UNIT 2 DATA
import seaborn as sns

def pairwisePlot(df=[], columnsForPlotting=[], reductionKey='', savelocation=''):
    # pairwise plots of input features
    X_Axis = df[[ col for col in columnsForPlotting if reductionKey in col]]
    g=sns.PairGrid(X_Axis, diag_sharey=False)
    g.map_diag(sns.histplot)
    g.map_upper(sns.regplot)
    g.map_lower(sns.scatterplot)
    for ax in g.axes.flatten():
        # rotate x axis labels
        ax.set_xlabel(ax.get_xlabel(), rotation = 45)
        # rotate y axis labels
        ax.set_ylabel(ax.get_ylabel(), rotation = 45)
        # set y labels alignment
        ax.yaxis.get_label().set_horizontalalignment('right')
    g.fig.set_size_inches(36,18)
    plt.suptitle('Input Data %s Feature Pairwise Plots'%reductionKey)
    g.tight_layout()
    if savelocation =='':
        plt.savefig(r'\%s pairwise plot.png'%reductionKey, dpi=300)
    else:
        plt.savefig(savelocation+'\%s pairwise plot.png'%reductionKey, dpi=300)
    plt.close()

def correlationPlots(df, columnsForPlotting=[], reductionKey='', savelocation=''):
    mask = np.zeros((df[[ col for col in columnsForPlotting if reductionKey in col]].shape[1],
                     df[[ col for col in columnsForPlotting if reductionKey in col]].shape[1]))
    mask[np.triu_indices_from(mask)] = True
    plt.figure(figsize=(36,18))
    sns.heatmap(df[[ col for col in columnsForPlotting if reductionKey in col]].corr(method='pearson'), mask=mask ,cmap='Blues',
                annot=True)
    plt.suptitle('Input Variable %s Feature Pearson Correlation'%reductionKey)
    plt.tight_layout()
    plt.savefig(savelocation+'\%s Pearson Correlation.png'%reductionKey, dpi=300)
    plt.close()

    plt.figure(figsize=(36,18))
    sns.heatmap(df[[ col for col in columnsForPlotting if reductionKey in col]].corr(method='spearman'),mask=mask, cmap='Blues',
                annot=True)
    plt.suptitle('Input Variable %s Feature Spearman Correlation'%reductionKey)
    plt.tight_layout()
    plt.savefig(savelocation+'\%s Spearman Correlation.png'%reductionKey, dpi=300)
    plt.close()


reductionKeys = ['Variance',
                 'Minimum',
                 'Maximum',
                 'Mean'
                 ]

for each in reductionKeys:
    pairwisePlot(dataset, columnsForTraining, reductionKey=each, savelocation=savelocation)

for each in reductionKeys:
    correlationPlots(df=dataset, columnsForPlotting=columnsForTraining, reductionKey=each, savelocation=savelocation)
```

## B.13 Plotting for Wrist Unit 1 Data



```
#%% Plotting for WRIST UNIT 1 DATA

# good lab connection

(fileNames, fileLocations) = findDataLoc(r"C:\Users\16sco\Box\!BMW_Push_Pull\Glove Testing\
                                         BMW Lineside Testing Raw\20210325 - SensorData-Sensor3 - Sorting\
                                             Lab Data\Chopped and sorted lab data")
# select file to plot
xx = 13
(dataRaw, colNames, label)=initialColumnRemoval(pd.DataFrame.to_numpy(pd.read_csv(fileLocations[xx], header=None)))

# Convert to DataFrame, process Timestamps, set column dtypes
dataProcessed=pd.DataFrame(data=dataRaw, columns=colNames)
dataProcessed["Timestamp"] = pd.to_datetime(dataProcessed["Timestamp"], format='%Y:%m:%d %H:%M:%S.%f')
dataProcessed = dataProcessed.astype({'FlexiForce':'float64', 'CustomSensor':'float64', 'ax':'float64',
                                      'ay':'float64','az':'float64','gx':'float64',
                                      'gy':'float64','gz':'float64',})

# Remap sensor output to 0-1 scale for training based on known limits or calibrations
#dataProcessed['FlexiForce'] = remap(dataProcessed['FlexiForce'], 74, 135, 0, 1)
#dataProcessed['CustomSensor'] = remap(dataProcessed['CustomSensor'], 69, 1023, 0, 1)
dataProcessed['FlexiForce'] = remap(dataProcessed['FlexiForce'], 0, 3300, 0, 1)
dataProcessed['CustomSensor'] = remap(dataProcessed['CustomSensor'], 0, 3300, 0, 1)
dataProcessed['ax'] = remap(dataProcessed['ax'], -2, 2, 0, 1)
dataProcessed['ay'] = remap(dataProcessed['ay'], -2, 2, 0, 1)
dataProcessed['az'] = remap(dataProcessed['az'], -2, 2, 0, 1)
dataProcessed['gx'] = remap(dataProcessed['gx'], -250, 250, 0, 1)
dataProcessed['gy'] = remap(dataProcessed['gy'], -250, 250, 0, 1)
dataProcessed['gz'] = remap(dataProcessed['gz'], -250, 250, 0, 1)

# Smooth Data
#dataProcessed['FlexiForce-smooth'] = filterSavitzyGolay(dataProcessed['FlexiForce'], 31, 3)

# Extract fileName
fileName = filePath.split(os.path.sep)[-1]

fig = plt.figure()
#fig(figsize=(8, 6), dpi=80)
fig.set_size_inches(12, 10)
gs = fig.add_gridspec(2, 1, hspace=0.1, wspace=0)
(ax1, ax3) = gs.subplots(sharex='col', sharey='row')
fig.suptitle("Sensor Comparison from 30006R Standardized Motion Values for ML (0-1)\nFile: %s"%fileNames[xx], fontsize=18)

ax1.plot(dataProcessed["FlexiForce"],'tab:blue',label="FlexiForce")
#ax1.plot(dataProcessed["FlexiForce-smooth"], color='green', linestyle='dashed', alpha=1, label="FlexiForce-smooth")
ax1.set_title("Force Sensors", fontsize=15)
ax1.legend(fontsize=12, loc=1, bbox_to_anchor=(0.7,0.2))

#ax2 = ax1.twinx()  # instantiate a second axes that shares the same x-axis
ax1.plot(dataProcessed["CustomSensor"],'tab:orange', label="Shear Sensor")
ax1.legend(fontsize=12, loc=1, bbox_to_anchor=(0.5,0.2))

ax3.plot(dataProcessed["ax"], label='AccX')
ax3.plot(dataProcessed["ay"], label='AccY')
ax3.plot(dataProcessed["az"], label='Accz')
ax3.plot(dataProcessed["gx"], label='GyrX')
ax3.plot(dataProcessed["gy"], label='GyrY')
ax3.plot(dataProcessed["gz"], label='GyrZ')
ax3.set_title("Motion Sensors", fontsize=15)
ax3.set_xlabel('Time HH:MM:SS', fontsize=15)
ax3.legend(fontsize=12)
```

## B.14 Plotting for Wrist Unit 1 Data - Acceleration and Gyro Separate



```python
#%% Plotting for WRIST UNIT 1 DATA (Acc and Gyro separate)
#(dataRaw, colNames, label)=initialColumnRemoval(pd.DataFrame.to_numpy(pd.read_csv(filePath, header=None)))
(dataRaw, colNames, label)=initialColumnRemoval(pd.DataFrame.to_numpy(pd.read_csv(fileLocations[xx], header=None)))

# Convert to DataFrame, process Timestamps, set column dtypes
dataProcessed=pd.DataFrame(data=dataRaw, columns=colNames)
dataProcessed["Timestamp"] = pd.to_datetime(dataProcessed["Timestamp"], format='%Y:%m:%d %H:%M:%S.%f')
dataProcessed = dataProcessed.astype({'FlexiForce':'float64', 'CustomSensor':'float64',
                                      'ax':'float64','ay':'float64','az':'float64','gx':'float64',
                                      'gy':'float64','gz':'float64',})

# Remap sensor output to 0-1 scale for training based on known limits or calibrations
#dataProcessed['FlexiForce'] = remap(dataProcessed['FlexiForce'], 74, 135, 0, 1)
#dataProcessed['CustomSensor'] = remap(dataProcessed['CustomSensor'], 69, 1023, 0, 1)
dataProcessed['FlexiForce'] = remap(dataProcessed['FlexiForce'], 0, 3300, 0, 1)
dataProcessed['CustomSensor'] = remap(dataProcessed['CustomSensor'], 0, 3300, 0, 1)
# dataProcessed['ax'] = remap(dataProcessed['ax'], -2, 2, 0, 1)
# dataProcessed['ay'] = remap(dataProcessed['ay'], -2, 2, 0, 1)
# dataProcessed['az'] = remap(dataProcessed['az'], -2, 2, 0, 1)
# dataProcessed['gx'] = remap(dataProcessed['gx'], -250, 250, 0, 1)
# dataProcessed['gy'] = remap(dataProcessed['gy'], -250, 250, 0, 1)
# dataProcessed['gz'] = remap(dataProcessed['gz'], -250, 250, 0, 1)

# Smooth Data
dataProcessed['FlexiForce-smooth'] = filterSavitzyGolay(dataProcessed['FlexiForce'], 31, 3)

# Extract fileName
fileName = filePath.split(os.path.sep)[-1]

fig = plt.figure()
fig.set_size_inches(12, 10)
gs = fig.add_gridspec(3, 1, hspace=0.1, wspace=0)
(ax1, ax3, ax4) = gs.subplots(sharex='col', sharey='row')
fig.suptitle("Sensor Comparison from 30006R Non-standardized Motion Values\nFile: %s"%fileNames[xx], fontsize=18)

ax1.plot(dataProcessed['Timestamp'], dataProcessed["FlexiForce"],'tab:blue',label="FlexiForce")
ax1.plot(dataProcessed['Timestamp'], dataProcessed["FlexiForce-smooth"], color='green', linestyle='dashed', alpha=1,
         label="FlexiForce-smooth")
ax1.set_title("Force Sensors", fontsize=15)
ax1.set_ylabel('Force\n(Relative scale pending calibration)', fontsize=15)
ax1.legend(fontsize=12, loc=1, bbox_to_anchor=(0.5,0.3))

#ax2 = ax1.twinx()  # instantiate a second axes that shares the same x-axis
ax1.plot(dataProcessed['Timestamp'], dataProcessed["CustomSensor"],'tab:orange', label="Shear Sensor")
ax1.legend(fontsize=12, loc=1, bbox_to_anchor=(0.8,0.3))

ax3.plot(dataProcessed['Timestamp'], dataProcessed["ax"], label='AccX')
ax3.plot(dataProcessed['Timestamp'], dataProcessed["ay"], label='AccY')
ax3.plot(dataProcessed['Timestamp'], dataProcessed["az"], label='Accz')
ax3.set_title("Accelerometer", fontsize=15)
ax3.set_ylabel('G-force (g)', fontsize=15)
ax3.legend(fontsize=12)

ax4.plot(dataProcessed['Timestamp'], dataProcessed["gx"], label='GyrX')
ax4.plot(dataProcessed['Timestamp'], dataProcessed["gy"], label='GyrY')
ax4.plot(dataProcessed['Timestamp'], dataProcessed["gz"], label='GyrZ')
ax4.set_title("Gyroscope", fontsize=15)
ax4.set_ylabel('Angular Velocity (deg/s)', fontsize=15)
ax4.set_xlabel('Time HH:MM:SS', fontsize=15)
ax4.legend(fontsize=12)
```

# Bibliography

[1] Michael Vineyard, Kwasi Amoako-Gyampah, and Jack R Meredith. Failure rate distributions for flexible manufacturing systems: An empirical study. *European Journal of Operational Research*, 116(1):139–155, 7 1999.

[2] Martin Wilkie, Ji Cheong, Charles Armitage, Alex Chang, Johanna Chua, Pavan Daswani, and Kota Ezawa. Factory of the Future: Flexible, Digitalized, and Sustainable. Technical report, Citigroup, 7 2019.

[3] Laura Geggel. Elon Musk Says 'Humans Are Underrated'underrated, 2018.

[4] Valentina Di Pasquale, Chiara Franciosi, Alfredo Lambiase, and Salvatore Miranda. Methodology for the analysis and quantification of human error probability in manufacturing systems. In *2016 IEEE Student Conference on Research and Development (SCOReD)*, pages 1–5. IEEE, 12 2016.

[5] Suzanne Berger, Erik Brynjolfsson, John Gabrieli, John Hart, Yasheng Huang, Jason Jackson, and Thomas Kochan. The Work of the Future: Shaping Technology and Institutions. Technical report, Massachusetts Institute of Technology, Cambridge, 11 2019.

[6] Doruk Şahinel, Cem Akpolat, O. Can Görür, Fikret Sivrikaya, and Sahin Albayrak. Human modeling and interaction in cyber-physical systems: A reference framework. *Journal of Manufacturing Systems*, 59:367–385, 4 2021.

[7] André Hengstebeck, Kirsten Weisner, Maike Klöckner, Jochen Deuse, Bernd Kuhlenkötter, and Jürgen Roßmann. Formal Modelling of Manual Work Processes for the Application of Industrial Service Robotics. *Procedia CIRP*, 41:364–369, 2016.

[8] Marco Bortolini, Maurizio Faccio, Francesco Gabriele Galizia, Mauro Gamberi, and Francesco Pilati. Design, engineering and testing of an innovative adaptive automation assembly system. *Assembly Automation*, 40(3), 2020.

[9] Ted Wackler, Chloe Kontos, Walter Copan, Paul Dabbar, Michael Kratsios, Lloyd Whitman, Michael Molnar, Abigail Slater, and Said Jahanmir. Strategy for American Leadership in Advanced Manufacturing. Technical Report October, 2018.

[10] Benedikt G. Mark, Erwin Rauch, and Dominik T. Matt. Worker assistance systems in manufacturing: A review of the state of the art and future directions. *Journal of Manufacturing Systems*, 59:228–250, 4 2021.

[11] Liying Xu and Feng Yu. Factors that influence robot acceptance. *Kexue Tongbao/Chinese Science Bulletin*, 65(6), 2020.

[12] M. Reinvee and K. Jansen. Utilisation of tactile sensors in ergonomic assessment of hand-handle interface: A review. *Agronomy Research*, 12(3):907–914, 2014.

[13] Takashi Sagisaka, Yoshiyuki Ohmura, Yasuo Kuniyoshi, Akihiko Nagakubo, and Kazuyuki Ozaki. High-density conformable tactile sensing glove. *IEEE-RAS International Conference on Humanoid Robots*, pages 537–542, 2011.

[14] Dawi Karomati Baroroh, Chih-Hsing Chu, and Lihui Wang. Systematic literature review on augmented reality in smart manufacturing: Collaboration between human and computational intelligence. *Journal of Manufacturing Systems*, 11 2020.

[15] Gereon H. Büscher, Risto Kõiva, Carsten Schürmann, Robert Haschke, and Helge J. Ritter. Flexible and stretchable fabric-based tactile sensor. *Robotics and Autonomous Systems*, 63:244–252, 1 2015.

[16] Kuen-Suan Chen and Tsang-Chuan Chang. A Modified Approach for Six Sigma Quality Assessment of Product with Multiple Characteristics in Intelligent Manufacturing Environments. *Journal of Testing and Evaluation*, 49(5), 9 2021.

[17] Katie Roof. Elon Musk says 'humans are underrated,' calls Tesla's 'excessive automation' a 'mistake', 2018.

[18] G. Boothroyd. Product design for manufacture and assembly. *Computer Aided Design*, 26(7), 7 1994.

[19] K.R. Ghadge and P.P. Mone. PLC Based POKE-YOKE System for Output Shaft Assembly. *Applied Mechanics and Materials*, 110-116, 2012.

[20] Geoffrey Boothroyd. Assembly 2000: Quality and automation will shape design. *ASSEMBLY-OAK-BROOK-42*, pages 18–19, 1999.

[21] Jonathan Trout. Poka-Yoke Explained.

[22] Young Hoon Kwak and Frank T. Anbari. Benefits, obstacles, and future of six sigma approach. *Technovation*, 26(5-6):708–715, 6 2006.

[23] Roger G. Schroeder, Kevin Linderman, Charles Liedtke, and Adrian S. Choo. Six Sigma: Definition and underlying theory. *Journal of Operations Management*, 26(4):536–554, 7 2008.

[24] Volker Paelke. Augmented reality in the smart factory: Supporting workers in an industry 4.0. environment. In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. IEEE, 9 2014.

[25] George Michalos, Panagiotis Karagiannis, Sotiris Makris, Önder Tokçalar, and George Chryssolouris. Augmented Reality (AR) Applications for Supporting Human-robot Interactive Cooperation. *Procedia CIRP*, 41:370–375, 1 2016.

[26] Bernardo Cassimiro Fonseca de Oliveira, Antonio Luiz Schalata Pacheco, Rodolfo Cesar Costa Flesch, and Miguel Burg Demay. Detection of defects in the manufacturing of electric motor stators using vision systems: Electrical connectors. In *2016 12th IEEE International Conference on Industry Applications (INDUSCON)*, pages 1–6. IEEE, 11 2016.

[27] Detect-It Industrial Product Line, 2021.

[28] Clarifai: The World's AI, 2022.

[29] K.K. John, S.N. Adarsh, and V. Pattali. Workers to super workers: A brief discussion on important technologies for industry 5.0 manufacturing systems. *AIP Conference Proceedings*, 2311, 12 2020.

[30] David Romero, Johan Stahre, Thorsten Wuest, Ovidiu Noran, Peter Bernus, Aring;sa Fast-Berglund, and Dominic Gorecky. Towards an operator 4.0 typology: A human-centric perspective on the fourth industrial revolution technologies. *CIE 2016: 46th International Conferences on Computers and Industrial Engineering*, 2016.

[31] M. Muzny, A. Henriksen, A. Giordanengo, J. Muzik, A. Groslashttland, H. Blixgaringrd, G. Hartvigsen, and E. Aringrsand. Wearable sensors with possibilities for data exchange: analyzing status and needs of different actors in mobile health monitoring systems. *International Journal of Medical Informatics*, 133, 1 2020.

[32] Alessandro Greco, Mariarosaria Muoio, Monica Lamberti, Salvatore Gerbino, Francesco Caputo, and Nadia Miraglia. Integrated wearable devices for evaluating the biomechanical overload in manufacturing. *2019 IEEE International Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2019 - Proceedings*, pages 93–97, 6 2019.

[33] X.T.R. Kong, Hao Luo, G.Q. Huang, and Xuan Yang. Industrial wearable system: the human-centric empowering technology in Industry 4.0. *Journal of Intelligent Manufacturing*, 30(8), 12 2019.

[34] S. Schönig, S. Jablonski, A. Ermer, and A.P. Aires. Digital Connected Production: Wearable Manufacturing Information Systems. *On the Move to Meaningful Internet Systems. OTM 2017 Workshops. Confederated International Workshops EI2N, FBM, ICSP, Meta4eS, OTMA 2017 and ODBASE Posters 2017. Revised Selected Papers: LNCS 10697*, 2018.

[35] ProGlove. ProGlove Wearable Scanner, 2021.

[36] Martin W. Buchenau. Projekt "IGLOVE": Bosch und die magische Hand, 8 2016.

[37] Nipun D. Nath, Theodora Chaspari, and Amir H. Behzadan. Automated ergonomic risk monitoring using body-mounted sensors and machine learning. *Advanced Engineering Informatics*, 38:514–526, 10 2018.

[38] Hojoon Kim, Kanghyeon Lee, Geonhui Jo, Jun-Sik Kim, Myo Taeg Lim, and Youngsu Cha. Tendon-Inspired Piezoelectric Sensor for Biometric Application. *IEEE/ASME Transactions on Mechatronics*, 26(5):2538–2547, 10 2021.

[39] GM Corporate Newsroom. GM-NASA Space Robot Partnership Brings 'Power' Glove to Life, 7 2016.

[40] Sujee Lee, Li Liu, Robert Radwin, and Jingshan Li. Machine Learning in Manufacturing Ergonomics: Recent Advances, Challenges, and Opportunities. *IEEE Robotics and Automation Letters*, 6(3):5745–5752, 7 2021.

[41] Alessandro Tognetti, Nicola Carbonaro, Giuseppe Zupone, and Danilo De Rossi. Characterization of a novel data glove based on textile integrated sensors. *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, pages 2510–2513, 2006.

[42] J. Oyekan, Y. Chen, C. Turner, and A. Tiwari. Applying a fusion of wearable sensors and a cognitive inspired architecture to real-time ergonomics analysis of manual assembly tasks. *Journal of Manufacturing Systems*, 61:391–405, 10 2021.

[43] Frank L. Hammond, Yiit Menguc, and Robert J. Wood. Toward a modular soft sensor-embedded glove for human hand motion and tactile pressure measurement. *IEEE International Conference on Intelligent Robots and Systems*, (Iros):4000–4007, 2014.

[44] Pressure Profile Systems TactileGlove - Hand Pressure And Force Measurement, 2020.

[45] ergoPAK™ ergoGLOVE, 2020.

[46] Novel.de manugraphy® – Hand assessment, 2021.

[47] Grip Pressure Sensor Glove, 2021.

[48] Tactilus Grip Pressure Sensor Glove - 24 Sensors, 2021.

[49] Tactilus Grip Pressure Sensor Glove - Recreational, 2014.

[50] Ondřej Fuksa. Options of Using ergoPAK Portable Analysis Kit for Application Purposes ( Možnosti využití sady ergoPAK pro aplikační účely). Technical report, University of West Bohemia in Pilsen, University in Pilsen, 2012.

[51] Ruiyang Yin, Depeng Wang, Shufang Zhao, Zheng Lou, and Guozhen Shen. Wearable Sensors-Enabled Human&ndash;Machine Interaction Systems: From Design to Application. *Advanced Functional Materials*, 31(11), 3 2021.

[52] Hongjian Zhang, Wenqi Han, Kui Xu, Yu Zhang, Yufei Lu, Zhentao Nie, Yuhang Du, Jixin Zhu, and Wei Huang. Metallic Sandwiched-Aerogel Hybrids Enabling Flexible and Stretchable Intelligent Sensor. *Nano Letters*, 20(5):3449–3458, 5 2020.

[53] Bor-Shing Lin, I-Jung Lee, and Jean-Lon Chen. Novel Assembled Sensorized Glove Platform for Comprehensive Hand Function Assessment by Using Inertial Sensors and Force Sensing Resistors. *IEEE Sensors Journal*, 20(6):3379–3389, 3 2020.

[54] Edward F. Austin, Charlotte P. Kearney, Pedro J. Chacon, Sara A. Winges, Prasanna Acharya, and Jin-Woo Choi. A Fabricated Force Glove That Measures Hand Forces during Activities of Daily Living. *Sensors*, 22(4), 2022.

[55] Xingyou Meng, Hanyang Li, Kui Zhang, Yuyang Sun, and Huicong Liu. A Glove-Based Human-Machine Interface Assisted by Ultra-Stretchable Strain Sensors and Three-Axis Force Sensors. *2021 21st International Conference on Solid-State Sensors, Actuators and Microsystems (Transducers)*, 2021.

[56] S. Borik, A. Kmecova, M. Gasova, and M. Gaso. Smart glove to measure a grip force of the workers. *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, 2019.

[57] Junghoon Park, Pilwon Heo, Jung Kim, and Youngjin Na. A Finger Grip Force Sensor with an Open-Pad Structure for Glove-Type Assistive Devices. *Sensors*, 20(1), 1 2020.

[58] Steven Cramp, Cam Maccoll, and R. Bruce Wallace. Preliminary results for novel shear force sensor using force sensitive resistors. In *I2MTC 2020 - International Instrumentation and Measurement Technology Conference, Proceedings*, 2020.

[59] Jason Mertodikromo, Fedor Zorin, and Chris J. Lee. A Low-Profile Shear Force Sensor for Wearable Applications. *IEEE Sensors Journal*, 20(18), 2020.

[60] Ye Zhang, A. Serdar Sezen, and Rajesh Rajamani. A Low-Profile Supercapacitor-Based Normal and Shear Force Sensor. *IEEE Sensors Journal*, 21(1):239–249, 1 2021.

[61] Lucie Viry, Alessandro Levi, Massimo Totaro, Alessio Mondini, Virgilio Mattoli, Barbara Mazzolai, and Lucia Beccai. Flexible Three-Axial Force Sensor for Soft and Highly Sensitive Artificial Touch. *Advanced Materials*, 26(17):2659–2664, 5 2014.

[62] Alexi Charalambides and Sarah Bergbreiter. A novel all-elastomer MEMS tactile sensor for high dynamic range shear and normal force sensing. *Journal of Micromechanics and Microengineering*, 25(9):095009, 9 2015.

[63] Ilker Kucukoglu, Hilal Atici-Ulusu, Tulin Gunduz, and Onder Tokcalar. Application of the artificial neural network method to detect defective assembling processes by using a wearable technology. *Journal of Manufacturing Systems*, 49:163–171, 10 2018.

[64] Iveta Zolotová, Peter Papcun, Erik Kajáti, Martin Miškuf, and Jozef Mocnej. Smart and cognitive solutions for Operator 4.0: Laboratory H-CPPS case studies. *Computers and Industrial Engineering*, 139, 2020.

[65] Matthew Krugh, Rishabh Mulesh Vedant, Ravi Shankar Garimella, Adithya Baburaj, Ethan Wescoat, and Laine Mears. Associate finger engagement during manual assembly in automotive production for smart wearable systems. In *Procedia Manufacturing*, volume 39, 2019.

[66] Vaughan G. Macefield, Charlotte Häger-Ross, and Roland S. Johansson. Control of grip force during restraint of an object held between finger and thumb: Responses of cutaneous afferents from the digits. *Experimental Brain Research*, 108(1), 1996.

[67] Kyung Sun Lee and Myung Chul Jung. Investigation of hand postures in manufacturing industries according to hand and object properties. *International Journal of Industrial Ergonomics*, 46, 2015.

[68] Anton Polishchuk, William Taube Navaraj, Hadi Heidari, and Ravinder Dahiya. Multisensory Smart Glove for Tactile Feedback in Prosthetic Hand. *Procedia Engineering*, 168(0):1605–1608, 2016.

[69] Seong Guk Kim, Chang Yong Ko, Dong Hyun Kim, Ye Eun Song, Tae Uk Kang, Sungwoo Ahn, Dohyung Lim, and Han Sung Kim. Development of a shear force measurement dummy for seat comfort. *PLoS ONE*, 12(11), 2017.

[70] Sang Min Won, Heling Wang, Bong Hoon Kim, Kunhyuck Lee, Hokyung Jang, Kyeongha Kwon, Mengdi Han, Kaitlyn E. Crawford, Haibo Li, Yechan Lee, Xuebo Yuan, Sung Bong Kim, Yong Suk Oh, Woo Jin Jang, Jong Yoon Lee, Seungyong Han, Jeonghyun Kim, Xueju Wang, Zhaoqian Xie, Yihui Zhang, Yonggang Huang, and John A. Rogers. Multimodal Sensing with a Three-Dimensional Piezoresistive Structure. *ACS Nano*, 13(10), 2019.

[71] Changyong Liu, Ninggui Huang, Feng Xu, Junda Tong, Zhangwei Chen, Xuchun Gui, Yuelong Fu, and Changshi Lao. 3D printing technologies for flexible tactile sensors toward wearable electronics and electronic skin, 2018.

[72] Lin Wang, Tim Littler, and Xueqin Liu. Gaussian Process Multi-Class Classification for Transformer Fault Diagnosis Using Dissolved Gas Analysis. *IEEE Transactions on Dielectrics and Electrical Insulation*, 28(5):1703–1712, 10 2021.

[73] Chih-Ta Yen and Jia-De Lin. Human body activity recognition using wearable inertial sensors integrated with a feature extraction&ndash;based machine-learning classification algorithm. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 2020.

[74] M.A. Hossain, M.I. Sabik, M.M. Rahman, S.N. Sakiba, A.K.M. Muzahidul Islam, S. Shatabda, S. Islam, and A. Ahmed. An Effective Leukemia Prediction Technique Using Supervised Machine Learning Classification Algorithm. *Proceedings of International Conference on Trends in Computational and Cognitive Engineering. Proceedings of TCCE 2020. Advances in Intelligent Systems and Computing (AISC 1309)*, 2021.

[75] Hari Trivedi, Joseph Mesterhazy, Benjamin Laguna, Thienkhai Vu, and Jae Ho Sohn. Automatic Determination of the Need for Intravenous Contrast in Musculoskeletal MRI Examinations Using IBM Watson&rsquo;s Natural Language Processing Algorithm. *Journal of Digital Imaging*, 31(2):245–251, 4 2018.

[76] M. Kusy, R. Zajdel, J. Kluska, and T. Zabinski. Fusion of Feature Selection Methods for Improving Model Accuracy in the Milling Process Data Classification Problem. *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020.

[77] Abhilash P. M and Chakradhar D. Prediction and analysis of process failures by ANN classification during wire-EDM of Inconel 718. *Advances in Manufacturing*, 8(4):519–536, 12 2020.

[78] Albina Jegorowa, Jaroslstrok;aw Kurek, Izabella Antoniuk, Wioleta Do&lstrok;owa, Michalstrok; Bukowski, and Pawelstrok; Czarniak. Deep learning methods for drill wear classification based on images of holes drilled in melamine faced chipboard. *Wood Science and Technology*, 55(1):271–293, 1 2021.

[79] P. Illavarason and B.K. Sundaram. A Study of Intrusion Detection System using Machine Learning Classification Algorithm based on different feature selection approach. *2019 Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2019.

[80] Se-Yun Hwang, Kwang-Sik Kim, Hyung-Jin Kim, Hong-Bae Jun, and Jang-Hyun Lee. Application of PCA and Classification for Fault Diagnosis of MAB Installed in Petrochemical Plant Process Facilities. *Applied Sciences*, 11(9), 2021.

[81] Yanxia Li, Yi Chai, Han Zhou, and Hongpeng Yin. A novel dimension reduction and dictionary learning framework for high-dimensional data classification. *Pattern Recognition*, 112, 4 2021.

[82] A. Jim&#233;nez-Cordero and S. Maldonado. Automatic feature scaling and selection for support vector machine classification with functional data. *Applied Intelligence. The International Journal of Research on Intelligent Systems for Real Life Complex Problems*, 51(1), 1 2021.

[83] Keisuke Nishino, Arata Suzuki, and Daisuke Fujita. Classification method based on Taguchi&rsquo;s T-method for small sample sizes. *Journal of Advanced Mechanical Design, Systems and Manufacturing*, 15(2), 2021.

[84] N. Rohani, P. Ruiz, R. Molina, and A.K. Katsaggelos. Variational Gaussian process for multisensor classification problems. *Pattern Recognition Letters*, 116, 12 2018.

[85] I. Kazunori and N. Tomohiro. Automated Defect Classification System for Semiconductor Manufacturing Processes Using Deep Learning. *Toshiba Review*, 74(5), 2019.

[86] Series 5 - Mark-10 Force and Torque Measurement.

[87] TSM23S-3RG — Applied Motion, 2022.

[88] Scott Kerner, Suryanarayanan Gunasekar, Rishabh Mulesh Vedant, Matthew Krugh, and Laine Mears. Parametrization of manual work in automotive assembly for wearable force sensing. *Journal of Manufacturing Systems*, 59, 4 2021.

[89] Zach. What is Mallows' Cp? (Defintion & Example), 5 2019.

[90] James Moody. What does RMSE really mean?, 9 2019.

[91] Sachin Date. The Akaike Information Criterion: A goodness of fit measure that is based on Information Theory.

[92] Jeremy Zhang. Dynamic Time Warping: Explanation and Code Implementation, 2 2020.

[93] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys*, 46(3):1–33, 1 2014.

[94] Bruno Stecanella. Support Vector Machines (SVM) Algorithm Explained, 6 2017.

[95] Sushanth Sreenivasa. Radial Basis Function (RBF) Kernel: The Go-To Kernel, 10 2020.

[96] Jason Brownlee. Gaussian Processes for Classification With Python, 10 2020.

[97] Jason Brownlee. Crash Course On Multi-Layer Perceptron Neural Networks, 5 2016.

[98] Sabri Boughorbel, Fethi Jarray, and Mohammed El-Anbari. Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. *PLOS ONE*, 12(6):e0177678, 6 2017.

[99] Davide Chicco and Giuseppe Jurman. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):6, 12 2020.

[100] Alejandro Baldominos, Alejandro Cervantes, Yago Saez, and Pedro Isasi. A Comparison of Machine Learning and Deep Learning Techniques for Activity Recognition using Mobile Devices. *Sensors*, 19(3):521, 1 2019.

[101] SSDSI Blog. FMEA? What is that?, 2020.