Modelling of a segmented pneumatic soft actuator with integrated sensors using neural networks

Master's thesis in Biomedical Engineering

Christoffer Thorsson

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

Master's thesis report 2024

Modelling of a segmented pneumatic soft actuator with integrated sensors using neural networks

Christoffer Thorsson

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2024

Modelling of a segmented pneumatic soft actuator with integrated sensors using neural networks
CHRISTOFFER THORSSON

Supervisor: Rikard Karlsson, Department of Electrical Engineering
Examiner: Emmanuel Dean, Department of Electrical Engineering

Master's Thesis 2024
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2024

Modelling of a segmented pneumatic soft actuator with integrated sensors using neural networks
CHRISTOFFER THORSSON
Department of Electrical Engineering
Chalmers University of Technology

Abstract

The recent decade has seen a tenfold increase in publications on soft actuators for medical applications. One of the devices which has sprung from this research is the soft rehabilitation glove. However, the gloves developed as of writing has been limited by the lack of accurate control, in turn stemming from a lack of feedback or closed loop controllers. This project aims to evaluate the feasibility of utilising sensor information & neural networks to develop a model for actuator angle prediction which could be used to close the control loop.

During this project a prototype pneumatic soft robotic system was developed, featuring a finger like segmented PneuNet actuator with two integrated flex sensors. Data was then gathered from this actuator which was used for training several neural networks to predict the three joint angles of the actuator.

The results of this project can not definitively state the feasibility of using neural networks for joint angle prediction on a PneuNet actuator with integrated flex sensors. However, the results do suggest that with further development and a broader dataset such a system could possibly be realized. The results also suggest that such a model requires several sources of sensor input, including not only current but also past information.

Keywords: Soft Actuator Glove, Segmented PneuNet, Neural Network Model.

Acknowledgements

I would like to take this time to express my utmost gratitude towards my examiner, Emmanuel Dean & supervisor, Rikard Karlsson, whom allowed me to work upon this thesis & without whose guidance & support this thesis would likely never see the light of day.

I also express gratitude towards the many people I have met across this long & winding road & who took time out of their own schedules to aid in my many difficulties, especially the bachelor group of Sofie Andersson, Felix Gråhn, Alexander Heijkenskjöld, Zacharias Lind, Sackarias Lunman & Anton Strandrot Frid for their help throughout the project.

Christoffer Thorsson, Gothenburg, 10 2024

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

DIP	Distal Interphalangeal
FEM	Finite Element Modeling
MCP	Metacarpophalangeal
PIP	Proximal Interphalangeal
PWM	Pulse-Width Modulation
SRG	Soft Rehabilitation Glove

Contents

List of Figures

List of Tables

1
Introduction

The recent decades have seen a massively increased interest and great advances in soft robotics [1]. In addition to soft grippers for industrial uses, an application which has garnered the attention of researchers is the use of soft robotics for wearable assistive and rehabilitation devices [1]. On this topic, publications have increased tenfold over the last decade [1]. This interest in using soft robotics for medical applications is due to their innate characteristics of being light, affordable, durable and bio-compatible as well as having high conformability to the user's movements compared to traditional devices based on rigid exoskeletons [1] [2] [3]. Designs using soft robotics for various medical applications have already been presented, such as simulating the human stomach [4] and a heart sleeve shown in vivo porcine studies to aid during heart failure [5]. In addition, exoskeletons for elbows and ankles as well as gloves have also been developed [1], the last of which will be the focus of this thesis.

Several research groups have presented systems utilising soft rehabilitation gloves (SRGs). Polygerinos et. al. [3] was one of the earliest reports publicised on the topic showing a pneumatic soft actuator glove that was capable of flexing the user's fingers. The actuator design was a so-called pneumatic network or PneuNet actuator, consisting of a silicone body with raised air chambers that, when inflated, would bulge and push against each other thus creating a uniform bending motion. Four of these were then mounted onto the back of the fingers, the thumb excluded, and controlled via electro-pneumatic valves. Since then, several other SRGs have been presented with actuators whose bending motions mimic the human finger. Yap et. al. [6] and Wang et. al. [7] built upon the PneuNet design but altered the actuator design so it would better conform to the fingers bending motion. Polygerinos et. al. [2] published a new design using fibre-reinforced actuators controlled via hydraulic pressure. Designs using other materials have also been presented, such as the textile actuator gloves developed by Nassour et. al. [8] and Yap et. al. [9]. Finally, this year Kokubu et. al. [10] [11] presented an SRG with a so-called modular actuator composed of individually controlled fibre-reinforced actuators. Kokubu et. al. in particular stress the need for actuators to conform to the bending motion of the individual finger to reduce the risk of injury [10].

However, the systems that have been presented do not contain feedback in regards to the actuator motion, instead relying on open loop control. Several featured feedback for regulating the pressure inside the system [2], [3], [7] and systems such as the one developed by Al-Fahaam et. al. [12] also featured sensors for detecting if the fingers of the user were flexed and if the fingertips were in contact with something. Yet no soft actuator glove has been presented which can relay information regarding the positioning or flexion angle of the actuator, which could pose a potential risk to patient safety. Though it should be noted that Wang et al. did create a bottom-up analytical model for estimating the bending motion given the inflating pressure which they then used in an open loop control [7].

There is therefore a need for sensors to be implemented into these systems in order to improve control and increase patient safety. To quote Yap et al.

> *Feedback system is required for more accurate control. In order to make the device more robust in terms of autonomy and feedback response to external stimuli, different types of sensors such as pressure sensor, soft elastic joint angle sensor, and force sensor are required.* [6]

This need for feedback for accurate control is not unique to glove applications. Other reports such as Elgeneidy et. al. [13], focusing on soft robotic grippers, have already demonstrated that a sensor suite of pressure and flex sensors can be used for controlling a non-segmented soft pneumatic actuator by training a neural network for estimating the actuator bending angle. However, from searches in public databases, no report has been published on using such sensors for modelling and controlling a segmented pneumatic actuator.

1.1 Aim

This thesis aims to explore the feasibility of data-driven modelling of segmented soft actuators. The objective of this thesis is to create a prototype soft actuator system, including an actuator, pneumatic control scheme and sensors with the goal of using data from these sensors to train a neural network to estimate the angle of the joints of an actuator with a finger-like bending motion.

1.2 Research questions

- Can a neural network be trained to estimate the joint angles of a pneumatic actuator with a finger-like bending motion?
- Which sensor signals are adequate inputs for the neural network to learn the joint motions?
- How do different input features affect the network prediction?

1.3 Delimitation

- This work will only consider the flexion and extension movement of a single actuator. Abduction or adduction will not be considered.
- The initial state for the actuator will be assumed to be at maximum extension.
- This work will not consider the impact of the human user upon the actuator movement.

1.4 Thesis outline

This thesis will begin with a theoretical background, providing the prerequisite knowledge to engage with the thesis's main subject fully. This includes a limited state-of-the-art summary of other systems developed as of writing, a summary of the anatomical terminology that will be used relating to the human hand and a brief introduction to neural networks for regression tasks. Following this, the third chapter will chronicle the development process of the prototype system, from concept generation to realized system. This is then followed by the testing chapter, which explains how the system was then used to gather data, how the networks were trained and the evaluation method that was used. After this, the results chapter presents the models created during this project. These are then analysed in the discussion, where trends and behaviours of the different models are explored and conclusions are drawn. The analysis is then followed by a discussion on the various decisions and limitations and how they may have impacted the results before being rounded off with a discussion on what future works may wish to focus on. Finally, a brief conclusion is made, summarizing the aim, the work undertaken, the results garnered and the conclusions drawn during this work.

2

Background

In this chapter, the necessary background information and state-of-the-art regarding soft rehabilitation gloves will be presented. Firstly, a quick summary of the relevant anatomical terminology used in this thesis will be provided and a selection of past works upon which this one is based will be summarized. After this, a brief introduction to neural networks for regression is provided.

2.1 Summary of relevant finger anatomy

The bones which constitute the finger are the phalanges [14]. [1] Each finger is made up of three phalanges, referred to as the proximal, intermediate/middle and distal phalanges according to their proximity to the palm [14]. These can be seen in figure 2.1 coloured green, blue and orange respectively. The phalanx closes to the palm, the proximal is in turn connected to one of the metacarpal bones which make up the middle of the hand [14] and are shown as yellow in figure 2.1.

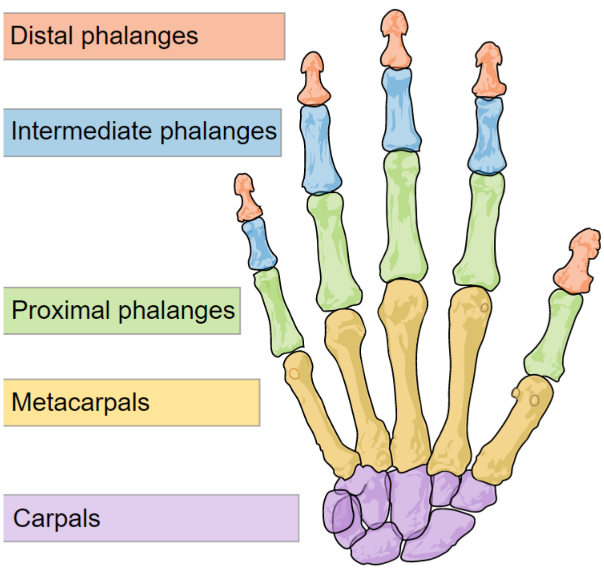


Figure 2.1: *Illustration of the phalanges of the human hand, dorsal view. Adapted from [15]. Licensed under public domain via Wikimedia Commons.*

[1]Note that the information presented pertains only to the index, middle, ring and little finger as the thumb is outside the scope of this work.

Each finger also has three joints, named according to the bone which it interacts with. In order from the palm to the fingertip, they are the metacarpophalangeal (MCP) joint, at the connection between the metacarpophalangeal and the proximal phalanx, the proximal interphalangeal (PIP) joint, between the proximal and the intermediate phalanx and the distal interphalangeal (DIP) joint, between the intermediate and the distant phalanx [14]. The MCP joint is capable of two motions: flexion/extension and abduction/adduction while the PIP and DIP joints are only capable of flexion/extension [16]. A demonstration of flexion can be found in figure 2.2 below.

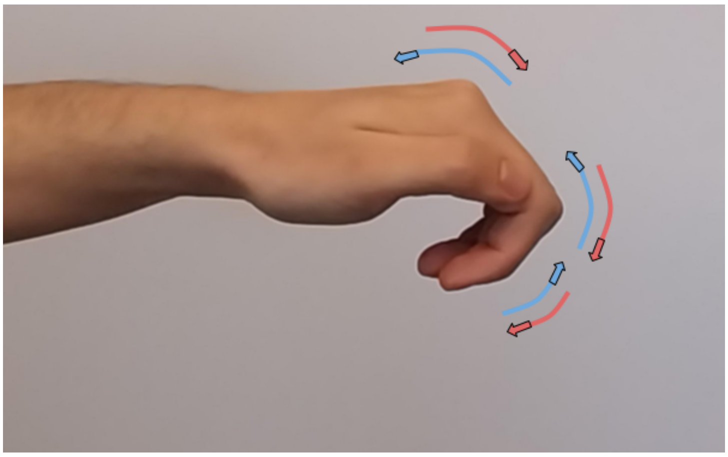


Figure 2.2: *Image demonstrating flexion (red) and extension (blue) motions of the finger. Image created by author*

2.2 Soft rehabilitation gloves

One of the earliest soft rehabilitation gloves (SRGs) was presented by Polygerinos et. al. [3] in 2013. In their report, the authors presented a design of a "hand rehabilitation glove fabricated using soft robotic technology" [3]. The system they developed consisted of a glove with four actuators attached via Velcro to the index, middle, ring and little finger [3]. The actuators were of the so-called PneuNet type [3], shown in figure 2.3. [2] The homogeneous PneuNet actuator consisted of a body made from an elastomer, in their case Elastosil, with a hollow air canal running along the bottom of the actuator and raised, hollow ridges on the top [3]. Under the bottom, a strip of paper, acting as a strain-limiting layer was added to create a flexing motion [3]. A second thin layer of elastomer was then added to seal the air canal and keep the paper strip in place [3].

[2] These will henceforth be referred to as homogeneous PneuNet to distinguish from the segmented PneuNet

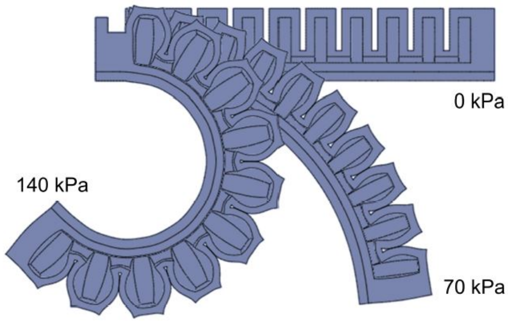


Figure 2.3: *Illustration of the working principle of the homogeneous PneuNet actuator. As the pressure inside the actuator is increased, the outer walls of the ridges begin to bulge. As they continue to bulge they eventually begin to push against their neighbors. This pushing, in conjunction with the strain limiting layer, causes the chamber pair to bend, creating a uniform flexing motion.*
Adapted from [17]

Figure 2.3 illustrates the working principle of the homogeneous PneuNet actuator: When pressurized, the walls of the ridges begin to bulge outwards [7]. When these extend far enough, they begin to push against one another [7]. Due to the strain limiting layer at the bottom, the force results in a flexing motion around the middle of the two ridges and causes the actuator to bend in an arching motion [7].

For controlling the SRG, electro-pneumatic valves were utilised and for gathering data air pressure and airflow sensors were added [3]. For capturing the true flexing angle computer vision was utilised, which was compared to Finite Element Method (FEM) simulations of the actuator behaviour [3].

An alternate design of the homogeneous PneuNet, while designed for robotic grippers but still of note, is the design by Elgeneidy et. al. [13]. In their report, a flex sensor cast into the bottom of the actuator was used alongside a pressure sensor to train a neural network to control the flexing angle of the actuator [13]. The system also utilised a 3/2 solenoid valve [13].

2.2.1 Segmented PneuNet

Two continuations of the PneuNet design were the gloves developed by Yap et. al. [6] and Wang et. al. [7]. Unlike the design by Polygerinos et. al. [3], the PneuNet actuators designed by Yap et. al. were of what the authors referred to as "variable stiffness" as well as featuring an actuator for the thumb [6]. The variable stiffness was realized by not having the raised air chambers along the entire length of the actuator but rather placing them so that the chambers would align with the joints of the finger [6]. Thus the parts of the actuator without chambers would correspond to the finger bones [6]. An illustration showing the flexing behaviour of such an actuator can be seen in figure 2.4. Note how the placement of the chambers creates "joints" leading to a different flexing shape when compared to the actuator used by Polygerinos et. al. in figure 2.3

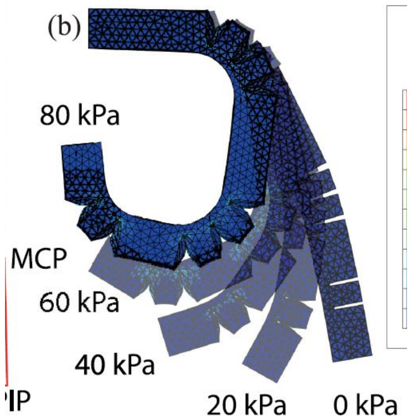


Figure 2.4: *Illustration of the segmented or variable stiffness actuator used by Wang et. al. [7] and Yap et. al. [6] respectively. Unlike the previous PneuNet design, the chambers are placed at specific positions along the actuator to produce specific flexing motions*
Adapted from [7]

According to Yap et. al. [6] the addition of the variable stiffness allows the actuator to better conform to the flexing motion of the user's finger. It would also allow for different exercises to be performed by altering the placement and dimensions of the ridges to control which joints will be flexed and to what extent [6]. They further state would provide better customizability for different hand dimensions [6].

The SRG consisted of five segmented PneuNet actuators, attached to the glove via Velcro [6]. For controlling the system, 3/2 solenoid valves were utilised, one for each actuator [6]. For tracking the motion of the actuator, reflective markers were placed on the actuators from which the positions and angles during actuation could be extracted for each joint [6]. The system utilised no sensors [6].

The SRG presented by Wang et. al. [7] utilised a similar actuator design, there referred to as a "segmented PneuNet" [7]. Like the actuators utilised by Yap et. al. [6], the chambers were placed to align with the joints of the finger [7]. The reason for this was that the previously presented homogeneous PneuNet tended to bend in an arcing motion which would not conform to the human finger [7]. They also state that by conforming to the human finger the actuator would be more comfortable because it would only apply flexing where the joints are located and not towards the finger bones [7]. The actuators used by Wang et. al. [7] featured no bottom layer to seal the actuator. Instead, an inner "bladder" was added, threaded into the actuator body and chambers, which was connected to the air supply [7]. According to the authors, a known issue of previous PneuNet actuators was that the seal between the actuator body and the bottom layer could be prone to leakages, which the air bladder would resolve [7].

The SRG consisted of five actuators, though unlike the ones presented thus far, they were attached via Velcro straps over the inflexible segments [7]. For controlling the glove two manually controlled solenoid valves were utilised [7]. For gathering data, a high-definition camera was used for extracting key nodes and markers from which they could plot the actuators' flexing behaviour [7]. To this end, an air pressure sensor was also used to compare with analytical models and FEM simulations [7].

2.2.2 Fibre reinforced actuators

An alternative to the PneuNet actuator is the fibre-reinforced actuator used by Polygerinos et. al. [2]. The design used various types of strain limiting layers at different locations along the actuator to, as the authors phrase it, "mechanically program" multiple motions into the actuator [2]. This can be seen in figure 2.5, which demonstrates how the direction of the winding of the fibres affects the motion of the actuator.

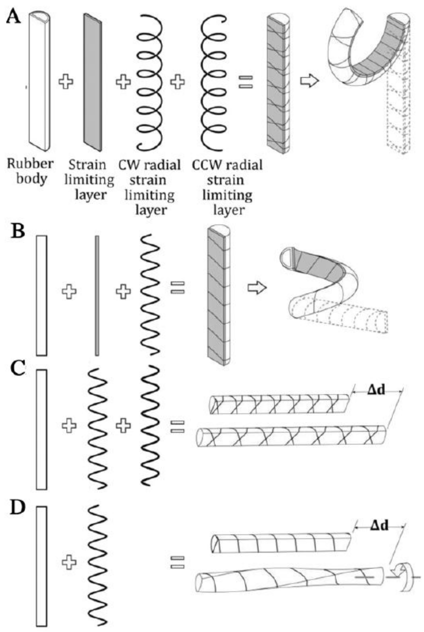


Figure 2.5: *Illustration showing the different programmable motions of a fibre-reinforced actuator. By using combinations of strain-limiting layer and fibre winding it is possible to achieve different motions at specific locations.*
Image adapted from [2]

Through this method, the actuator could perform multiple motions, such as flexing, flex-twist, linearly extending and twisting at different segments [2]. The authors used this method to create one actuator design for flexion of the fingers, consisting of flexing and extending segments, as well as an actuator for flexion of the thumb using bend-twist, twist-extend, bend and extend segments [2]. The first of these can be seen in figure 2.6 below. As has been stated by the previously presented systems, Polygerinos et. al. states that the reason for the segmentation was that the actuator must follow the range of motion of the patient's finger [2].

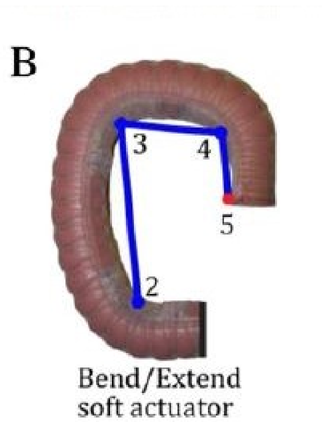


Figure 2.6: *Illustration showing bend and extend motions being used to achieve finger-like bend motion.*
Image adapted from [2]

The SRG consisted of five actuators, four of the bend and extend actuators as well as a specialized thumb actuator, attached via loops along the glove fingers [2]. Unlike the previous systems, the system utilised hydraulics in favour of pneumatics [2]. For regulating pressure, ten solenoid valves (an inlet and outlet per finger) were used, controlled via a feedback loop using a pressure sensor. A notable feature of this system is that it is portable

While not an SRG, another system which utilised the fibre-reinforced actuator was the one developed by Memarian et. al. [18]. Most notable of the system was the use of a gyroscopic sensor mounted at the end of the actuator and used in a control loop for regulating the orientation of the actuator [18]. The pressure was regulated via a 3/2 solenoid valve and a "pneumatic low pass filter" [18].

An interesting alternative to the segmented fibre-reinforced actuator is the modular fibre-reinforced actuator presented by Kokubu et. al. [10] [11]. In their report, the authors presented a design which featured multiple modules [10]. Each module consisted of a fibre-reinforced flexing actuator and 3D printed connectors, which could then be used independently or three units could be coupled together to produce a single unit that would bend similarly to the human finger [10]. An illustration of this can be seen in figure 2.7. Kokubu et. al. then continued development into an SRG where they demonstrated the ability to independently affect the joints [11]. They also compared it to an SRG with non-modular fibre-reinforced actuators and determined the modular design to have superior performance in terms of flexing angles, torque and response time [11].

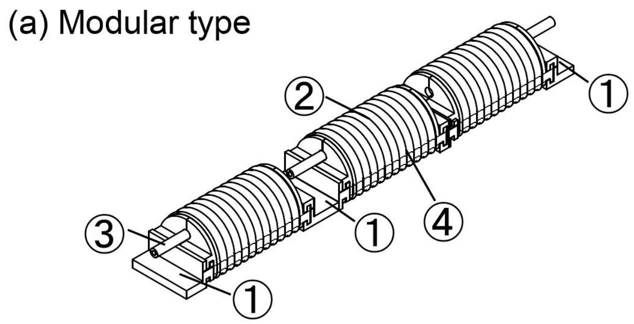


Figure 2.7: *Illustration showing a modular fibre reinforced actuator. 1) denotes the connector, 2) the actuator body, 3) the air tube and 4) the reinforcing fibre* Image adapted from [10]

Kokubu et. al. stated that a critical step for realizing SRGs is to develop methods for customizing the actuator to the patient [10]. Aspects such as the length of the finger and distance between the joints as well as the range of motion and stiffness of each joint must be able to be accommodated [10]. According to the authors, should the actuator not conform to the patient's finger in these regards, there is a risk that the actuator would apply forces which risk damaging the joints or supporting tissues [10]. They in turn cite a report by Shaw and Morris [19], which found that even for healthy people there is variation in the fingers' range of motion as well as finding a potential correlation between the lower range of motion and an increased risk of injury. Kokubu et. al. state that the risk is especially high if the force of the actuator is applied incorrectly or if the force is not applied perpendicular to the joint axis [10].

2.2.3 Textile based actautors

Finally, an alternative to the silicone or rubber actuators presented thus far is the textile actuator glove developed by Nassour et. al. [8]. This design utilised an inflatable tube which is placed inside a fabric housing [8]. By folding the inflatable tube in a zigzag pattern inside the housing and attaching it to another linearly extending actuator they achieved a flexing motion [8]. To resolve issues of applying the force of the actuator in ways that may cause discomfort, the authors presented a "variable curve actuator" [8]. This entailed modifying the housing so that the density of folds was changed according to the desired behaviour, where a higher density of folding led to thinner air chambers and in turn higher flexing force [8]. The manufacturing process of this actuator can be seen in figure 2.8

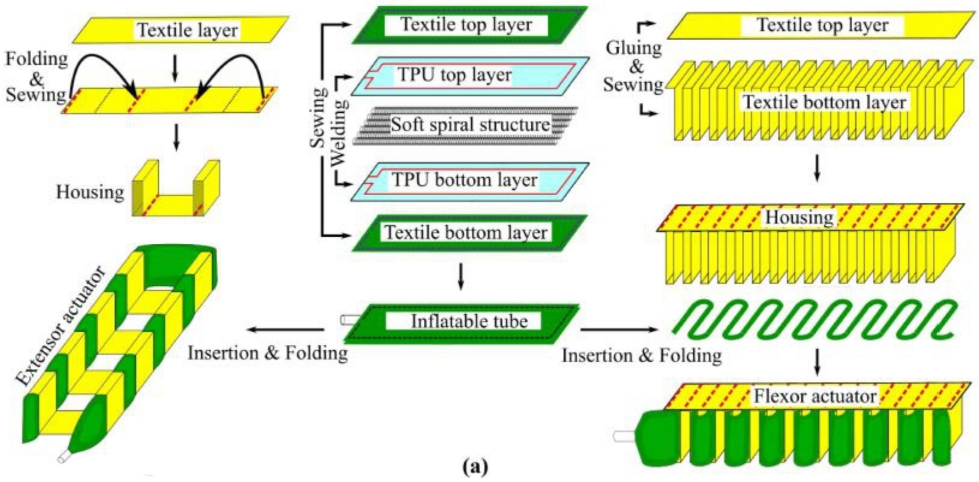


Figure 2.8: *Illustration showing the construction of a textile based actautor.* Image adapted from [8]

2.3 Solenoid valves

For controlling SRGs, most designs employ electro-pneumatic valves such as solenoid valves for regulating pressure [1] [20]. The reason for their popularity are due to their relatively low price, compact size and simple control [1] [20]. According to Huang et. al. the three most common are to use two 2/2 solenoid vales or a single 3/2 or 2/2 proportional valve [20].

A solenoid valve fundamentally consists of three components, illustrated in figure 2.9 by a 3/2 valve: A main body (light grey), a spring-loaded magnetic spool (dark grey) and a coil (yellow) [21]. In a normally closed solenoid valve, the spring is used to keep the spool in the closed state (upper image of figure 2.9) [21]. Once a current is fed to the coil a magnetic field is induced, causing the magnetic spool to move which opens the valve (lower image of figure 2.9) [21]. In a normally open valve, the spring and coil serve the opposite purpose [21].

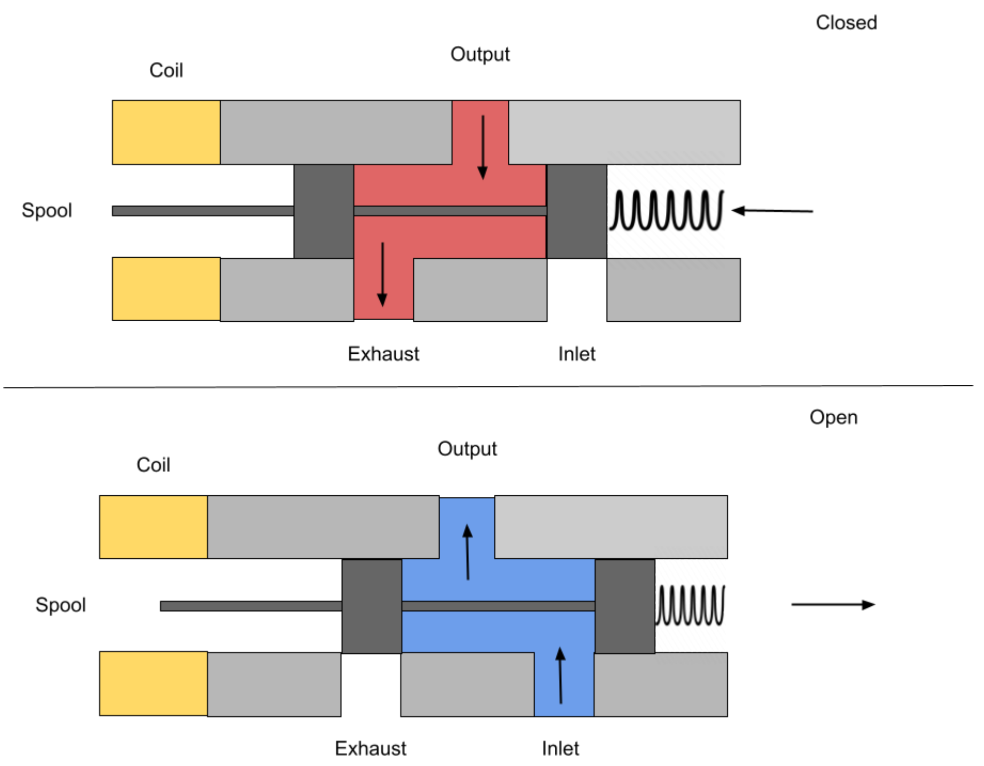


Figure 2.9: *Illustration of 3/2 solenoid valve. Image created by author*

A 2/2 valve has two ports (inlet and outlet) and two states: open, in which ports one and two are connected allowing for air to flow, and closed, in which the ports are not connected, blocking all flow [21]. When using this method, two valves are used: one connects the supply to the outlet, allowing for flow into the actuator, and one connects the outlet to an exhaust, allowing for air to vent from the actuator [20]. Huang et. al. employed two 2/2 valves controlled via Pulse-width modulation (PWM), which acted as a single 3/2 valve [20]. However, other authors such as Wang et. al. [7] and Polygerinos et. al. [2] utilised an alternate system of two 2/2 which did not utilise PWM. In the case of Wang et. al. the valves were controlled via two respective switches [7] and Polygerinos et. al. utilised a control loop where the relevant valves would open and remain open based on the desired vs the current pressure [2].

The 3/2 valve meanwhile has three ports (inlet, outlet and exhaust) [21]. In the open state, the inlet is connected to the outlet, allowing for air to flow into the actuator while the exhaust is disconnected [21]. In the closed state, the inlet is disconnected while the outlet is connected to the exhaust allowing for the venting of air [21]. As the outlet is always connected there is always a flow through the 3/2 solenoid valve. To maintain a certain pressure, the valve must be opened and closed to let an appropriate amount of air in as well as out [13]. For this, a PWM signal is typically used [20]. This method is accomplished by opening and closing the valve at a high rate, typically over 60 Hz, thus allowing the pressure to be regulated according to the duty cycle of the PWM signal [13]. The pressure can be expressed via the equation

$$p_{out} = k \cdot D_{PWM} \cdot p_{in} \tag{2.1}$$

where p_{out} is the output pressure after the valve, k is a constant which must be determined via testing, D_{PWM} is the duty cycle of the PWM signal and p_{in} is the pressure at the inlet [20]. The constant switching of the valves induces oscillations in the pressure [13]. Another consideration when utilising the 3/2 solenoid valve is the so-called "dead time", describing the time required for the valve to fully open [22]. Should the high period of the PWM signal be shorter than the dead time the valve will not open and thus no air will flow from the inlet [22].

Unlike the previously described valves, the proportional solenoid valve does not have distinct states [23]. Instead, its plunger, which controls the flow through the valve, is proportional to the current being fed to the valve, which in turn is related to the duty cycle of the PWM control signal [23]. These valves have been shown to be the best in terms of performance, however, they are by far the most expensive [20].

2.4 Neural Networks for regression models

According to Elgeneidy et. al. [13], modelling and control of soft pneumatic actuators have proved challenging. This is due to the non-linear elastomers most often used for the actuators which pose difficulties in creating accurate models of the material properties [13]. They further state that the manual nature of the actuator fabrication results in variations further complicating analytical models [13]. As such, in their work to model a homogeneous PneuNet actuator, they argue for the use of data-driven models and neural networks for regression [13].

Linear regression involves solving for an unknown value (target) given some set number of input variables (features) with corresponding weights [24]. This can be expressed as

$$\widehat{y} = w_1 \cdot x_1 + ... + w_d \cdot x_d + b \tag{2.2}$$

where $\widehat{y}$ is the estimated value, x are the features, w are the corresponding weights of that feature and b is the bias [24]. The regression task is to set the weights and bias of the above equation so that for each set of feature values the equation results in an approximation of the correct target value. In the work by Elgeneidy et. al. [13], the inputs were the resistance of the flex sensor, the pressure within the actuator and the orientation of the actuator. The output meanwhile was the estimated flexion angle [13].

At the core of the neural network is the artificial neuron [25]. The artificial neuron receives some number of inputs, which it weights according to the importance of the input signal [25]. It then sums all the weighted inputs into a single value, called the activation voltage [25]. If this activation voltage exceeds a set threshold, the neuron is considered to be excited [25]. The difference between the activation voltage and the threshold, called the activation potential is then multiplied with an activation function, which becomes the output of the neuron [25]. The above explanation can be expressed as

$$u = \sum_{i=1}^{n} w_i \cdot x_i - \Theta \tag{2.3}$$

$$y = g(u) \tag{2.4}$$

where u is the activation potential, w_i is the i'th weight corresponding to the input x_i, θ is the activation threshold, y is the neuron output and $g(u)$ is the activation function [25]. There are several types of activation functions, such as step functions, linear amplification and sigmoid functions amongst others [25].

To create the neural network, multiple neurons are grouped into what is called hidden layers [25]. Neurons do not interact within the same layer but instead receive information from the previous layer, evaluate it according to equations 2.3- 2.4, whose output is then transmitted to the next layer [25]. In addition to hidden layers, a network contains an input layer, where the information sent into the network is received [25]. This can be sensor values, images, sequences of data etc. [25]. This input is almost always normalized before being fed into the hidden layers [25]. In the case of Elgeneidy et. al., the input was the readings from the sensors [13]. Finally, the network contains an output layer, also composed of neurons, where the result of the network computations is output [25]. In the case of the network used by Elgeneidy et. al., the output layer was a single neuron corresponding to the predicted actuator angle [13]. A key characteristic of neural networks is the ability for the width (increased amount of neurons in a layer) and depth (increased amount of hidden layers) to be modified to increase network accuracy [26]. However, should the network be made too large there is the risk of overfitting.

The challenge is then to find the weights and parameters in the network so that for each input the data fed through the system the network will produce the corresponding output, i.e. to estimate the relation between the input features and their corresponding outputs [24]. To set these weights the network is trained [25]. The details of neural network training are beyond the scope of this report and thus it will be summarized that during the training process, the network is fed training data with a corresponding "ground truth" and via various methods such as stochastic gradient descent gradually adjusts the network parameters according to a loss function, which acts as a measure of the quality of the model [24] [25]. For regression tasks this typically is the squared error function found below [24]

$$l^{(i)}(\mathbf{w}, b) = \frac{1}{2}(\hat{y}^{(i)} - y^{(i)})^2 \tag{2.5}$$

where $l^{(i)}$ is the loss of the input value i, $\mathbf{w}$ and b are the network weights and biases respectively, $\hat{y}^{(i)}$ is the network output for the input i and y^i is the true value which the model aims to reach [24].

For a more detailed explanation and additional information see [24] [25].

3

Concept generation and system development

In this chapter, the conceptualization and development of the prototype system is recounted. First, the process of gathering information on other systems and evaluating which aspects to include in the prototype is explained. After this, the realization of the system is detailed, including the design of the actuator, the method for controlling the valve and the creation of the testing rig.

3.1 Concept generation

As this project began with no predefined system, the first step was to determine what systems had already been presented, what challenges had been faced and what requirements existed for such a system. This included previous actuator designs, what components had been used, methods for modelling soft actuators, what sensors had been utilised etc. As such, a limited literature overview was conducted. The search was done using Google Scholar, using keywords such as soft actuator glove, soft actuator stroke, PneuNet actuator, segmented PneuNet soft actuator sensor, solenoid valve, PWM control, and soft actuator control. The results of this overview are recounted in chapter 2.2 and 2.3.

From this overview, it was found that the system's core components would be: a segmented actuator, an electronic valve, a 4-power mosfet board for controlling said valves, a control unit, a pressure sensor, and an additional type of sensor. A pressure regulator would also be required to reduce the inlet pressure as well as a bleed valve for emptying the system. Due to availability as well as previous familiarity the Arduino Mega 2560 was initially chosen as the control unit. However, it was later changed into an Arduino Due as the processing power of the Mega was found to be insufficient. The choice was also made to utilise pneumatic actuation, both as a pneumatic supply was readily available as well as concerns regarding potential leaks or ruptures should hydraulics be used.

The pressure sensor was deemed necessary as regardless of which valve was chosen the control variable for the actuator would be the pressure inside it. It would also be required to characterize the actuator. As for what other sensor to utilise, from the overview the two known options were flex sensors [13] or gyroscopic sensors [18]. However, the gyroscopic solution was seen as less suitable due to the more complex computations of calculating the integral of the sensor readings, as well as the readings drifting over time. Flex sensors, a type of sensor whose impedance increases linearly with the bending angle of the sensor, were seen as computationally light and more robust. Previous works have also shown flex sensors to be suitable for capturing finger motions [27] [28].

A system diagram of the complete system can be found in figure 3.1 below and a full-size image can also be found in Appendix C. Red lines show electrical supply lines, blue shows pneumatic connections, green shows data flows and black show control signals.

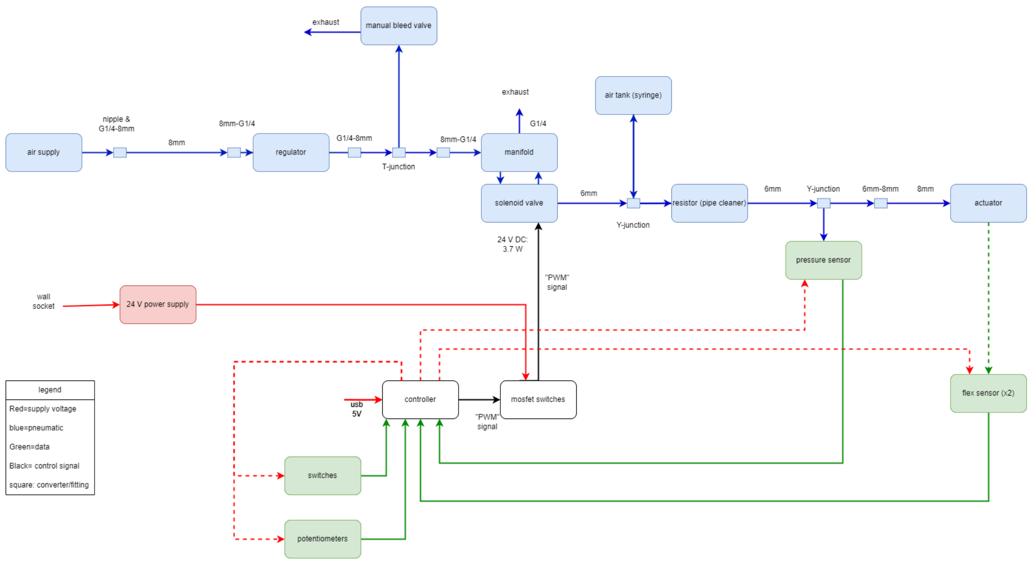


Figure 3.1: System diagram of proposed system

3.1.1 Actuator choice

From the literature overview, four potential designs for actuators were considered for this project: The homogeneous PneuNet, the segmented PneuNet, the fibre-reinforced actuator and the folded textile actuator. The modular fibre-reinforced actuator developed by Kokubu et. al. [10] had not been presented at the time of the design decision and was not considered.

The main criterion was the feasibility of constructing said actuator, as they would have to be made in-house, and secondly the actuator flexion behaviour, i.e. the way the actuator flexes in response to pressure. Additional aspects which were less important for this thesis but would be relevant for future development were weight and bulkiness or the ability to control each joint independently.

The homogeneous PneuNet actuator was ruled out as the flexing motion would not conform to the human finger, which according to Wang et. al. could risk putting a strain upon the phalanges during flexion [7]. Kokubu further stressed the importance of the actuator conforming to the range of motion and stiffness of the human finger joints to avoid discomfort or injury [10]. The fabric actuator was ruled out because, while it had been shown to function as a glove, the actuator had been shown primarily as a power assistive tool and as such the control was unexplored, in addition to being more demanding to produce. The fibre-reinforced actuator was ruled out as, while it could flex in the desired fashion and was able to twist as well as flex, it was found by the bachelor group of Andersson et. al. [29] to be quite labour-intensive to manufacture. In addition, the manual task of winding the fibres would impact the consistency of the actuators which would complicate the modelling. The segmented PneuNet meanwhile was deemed relatively simple to manufacture, capable of finger-like flexing motion as well as being widely used in similar systems, which offers several designs and results to draw inspiration from and compare with. Thus it was chosen.

3.1.2 Valve choice

Based on the findings of the literature overview, it was found that most systems utilised solenoid valves [20], in particular, 2/2 [2] [7], 3/2 [6] [13] [18] or proportional solenoid valves [20] [30]. The reason for this according to Xavier et. al. [1] as well as Huang et. al. [20] were due to their relatively low price, compact size and simple control. As it had been shown to be a viable solution it was decided to also utilise these.

In regards to which to use, all three had desirable traits. The 2/2 solenoid valve offered the simplest control scheme. By using one valve as an inlet and another as an exhaust, the pressure could be controlled by opening the corresponding valve until the desired pressure was reached. In addition, it would be possible to close both valves and seal off the actuator which could prove useful in retaining the set pressure. However, it would require two valves per actuator which was seen as undesirable due to the added weight and size. Also, it would not be possible to control the rate of change. As such, once the valve was opened the pressure could rise rapidly, depending on the flow rate of the system, which may pose a risk to patient safety. Because it was the only valve where the flow rate could not be regulated it was discarded as a viable solution.

The proportional solenoid valve, as shown by Huang et. al. does offer the best performance but is also the most expensive [20]. The proportional valve would allow for control over the flow rate, which is desirable both for control but also for patient safety. However it would also require the implementation of a more complex PWM control. Depending on the valve chosen it would either require two two-way valves or a single three-way valve, with the three-way being more desirable due to weight and size.

The 3/2 solenoid valve would be the most complex to control as not only would PWM be required but also be modified to account for the dead time to ensure proper functioning. It would also not be possible to close off the actuator, requiring the system to be able to maintain the pressure via the PWM signal. The opening and closing would also result in fluctuations in pressure which if severe enough could potentially result in vibrations in the actuator. Nor would it be able to control the rate of change, though it was reasoned that altering the PWM in minor increments would allow for a perceived slower rate of change. It would however be the most low-cost option, as the cost of a 2/2 and 3/2 valves are comparable the 3/2 system would only require a single valve, while offering slightly inferior performance to the proportional valve [20].

Because the 3/2 solenoid valve offered sufficiently high performance at the most affordable price, would only require one valve per actuator as well as there being systems which show the viability of 3/2 solenoids for the intended task, it was seen as the ideal choice. Also, as the 3/2 valve and proportional valve are both controlled via PWM it was reasoned that should the 3/2 be insufficient, changing to a proportional valve would require little modification.

For the project, achieving stable pressure was seen as important. As such, the valve would require a high switching frequency to "smooth out" the variations in pressure due to the valve switching. Therefore, the MHA3-M1H-3/2G-3 valve from Festo was used as it offered a maximum switching speed of 130 Hz, the highest of the available assortment.

3.2 Prototype development

In this section, the development of the prototype system used in this project will be detailed. The section is divided into two subsections, the first describing the design and construction of the soft actuator and the second describing the system for controlling the pressure inside the actuator.

3.2.1 Actuator development

The design of the segmented PneuNet actuator was based on the one developed by Wang et. al. [7]. This design was chosen as it was well documented, with bending profiles and design dimensions included which reduced the need for experimentation with dimensions. However, the balloon which they had fitted as an inner bladder was removed in favour of a bottom layer, such as the one used by Yap et. al. [6], as that would make the implementation of sensors in the style of Elgeneidy et. al. [13] simpler.

Based on the results of Elgeneidy et. al. [13] the decision was made to utilise integrated flex sensors into the bottom of the actuator. However, as the actuator was intended to flex similarly to a human finger, the decision was made to place the flex sensors based on the placement of sensor gloves, more specifically, the CyberGlove II. The design featured a fabric glove with integrated joint angle sensors [27]. Specifically in regards to flexion, the glove uses three of these sensors, mounted dorsally of the fingers with each overlapping one of the three finger joints [28]. As this design had been shown to be capable of use in recognizing sign language [28], a similar sensor placement was utilised. However, as the sensors available were too large as to not overlap multiple joints, in particular the DIP, the decision was made to have two flex sensors, one overlapping the MCP joint and one spanning the PIP and DIP joints, which can be seen in figure 3.2.

Figure 3.2: Image showing flex sensor positioning on the underside of the actautor

As for materials, according to an article by Xavier et. al. [1] the most common material for actuators is silicone rubber [1]. The literature overview of PneuNet actuators also supported this [2][3][6][7][12][13][18]. In particular, silicone rubbers with a hardness rating of between 20A and 30A shore hardness are the most common [1]. Testing by bachelor group Andersson et. al. [29] also found that range to be suitable, with softer actuators lacking the necessary hardness to apply sufficient force and harder actuators having issues of breaking when demolding as well as having far longer cure times. Due to the higher cure time of 30A silicone, the decision was made to utilise a 20A silicone, specifically Dragonskin 20A from Smooth-on, for time-saving reasons.

The manufacturing process was largely as outlined by Yap et. al. [6] and the guide published by the Soft robotics toolkit [31], with the actuator being cast in two pieces using 3D printed moulds. The decision was made to cast the actuator in three pieces, rather than two. One piece would define the segments and air chambers, one piece for sealing the bottom of the actuator and an additional piece beneath the second for placing the sensors and protecting them from outside disturbances. The liquid silicone would be poured into the moulds and then placed in a vacuum chamber to remove any air trapped inside the silicone. Early attempts were made to cast the sensors into the bottom layer. However, this was abandoned as it was found to be difficult to position them and prevent them from moving while in the vacuum chamber. To make connecting air supply easier a 3D-printed resin hose connector was cast into the actuator.

For bonding the pieces, several substances were tested. Silicone glue was initially tried. However, due to it requiring safety training as well as having a very long cure time of 24 hours, it was quickly disregarded. Andersson et. al. also found that the glue, being quite thick, was prone to form lumps which caused leakages [29]. Using the far softer Smooth-on FX pro silicone was also attempted, with it having a very quick cure time and it being reasoned silicone would bond well to other silicone. However, while these bonds were better they were still too weak and higher pressures quickly caused them to fail. In the end, the best result was found to be using the same silicone that the actuator was made of. However, for attaching the sensor layer the FX pro was chosen as the issue of weak bonding was not an issue there as well as offering a way to recover sensors if an actuator had to be discarded.

The final actuator design can be seen in figures 3.3 and 3.4 for the extended and flexed cases respectively. The actuator features two chambers in the MCP, two in the PIP and one in the DIP, separated by solid segments intended to align with the index finger of the author.

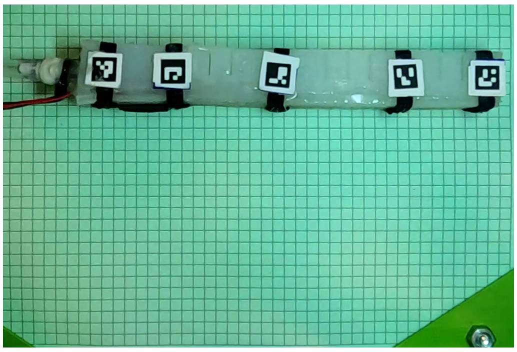


Figure 3.3: *Actuator in fully extended position*

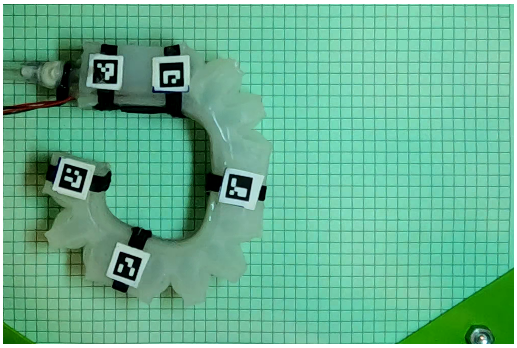


Figure 3.4: *Actuator in flexed position*

3.2.2 Valve control

As mentioned previously in chapter 2.3, a 3/2 solenoid valve such as the one used in this project has only two states: open or closed, i.e. increasing or decreasing in pressure. As such, the way to regulate the pressure is by opening and closing the valve at a high frequency, in some literature referred to as "rapid valve switching" [13]. For this, a PWM signal would be used, where the duty cycle ratio of the signal would correspond to the relative time the valve should remain open during a cycle.

However, as was soon discovered, the valve did not behave as expected. Normally, a solenoid valve, when exposed to a constant voltage should open and remain open, since the duty cycle is 100%. However, studying the valve step response it was discovered that when fed with a constant voltage the valve would instead remain open for approximately 35 ms and then close, which can be seen in figure 3.5.

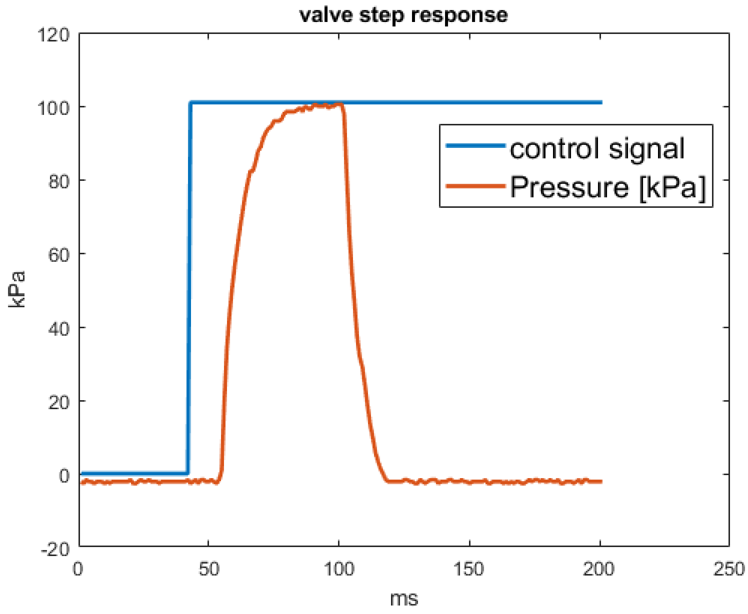


Figure 3.5: *Plot showing air pressure (red) in response to 100% duty cycle (blue).*

The hypothesized explanation is that there is a capacitor within the valve, which causes an open circuit that cuts the power once it has been saturated. Another theory is that Festo previously offered a controller for this valve which as of writing has been discontinued. Additional issues arose with the valve closing time. For PWM to function as desired, the valve must have enough time to both open and close during each cycle. This led to issues with the closing time, since if the frequency or duty cycle was too high, there would not be enough time for the valve to close, essentially locking it open and causing the pressure to increase up to the inlet pressure. There was also an issue regarding the PWM signals generated by the Arduino, as their frequency was set to 490 Hz, while the valve was limited to 130 Hz, and could not be easily modified.

To circumnavigate this a new control method had to be devised. PWM was still used, however, rather than using the frequency and duty cycle ratio as the control parameters, the signal would instead be created using the "on time" and "off time" of the PWM. The on-time was defined as the duration of the high period of a PWM cycle, while off-time was naturally the duration of the low period. This would allow for easier control by ensuring the low period never was too short for the valve to close. An illustration showing the traditional and alternate PWM signals can be found in figure 3.6. In blue, it is shown how the signal can change frequency (second row) to make the wave shorter or longer and change duty cycle ratio (third row) to make the high period longer or shorter. In red it is shown how the alternate signal instead directly modifies the off time (second row) and on time (third row).

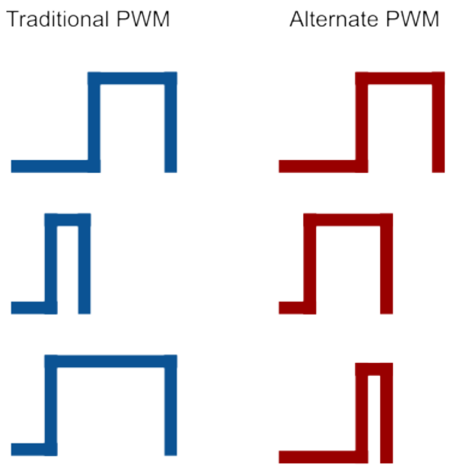


Figure 3.6: *Illustration showing alternate PWM signal*

Testing this new control scheme, an issue regarding the stability of the rapid valve switching soon emerged. As can be seen in the blue line in figure 3.8, showing a plot of the pressure during an 11 ms on and 3 ms off time pulse train, the pressure variations during a cycle were far too high. The large oscillations were not ideal, as they resulted in vibrations and spasms in the actuator. Since this was a known issue with this type of valve, Memarian et. al. [18] implemented a so-called "pneumatic low pass-filter" into their system, consisting of a pneumatic resistor (pipe cleaner) and an air tank (syringe) as in figure 3.7.

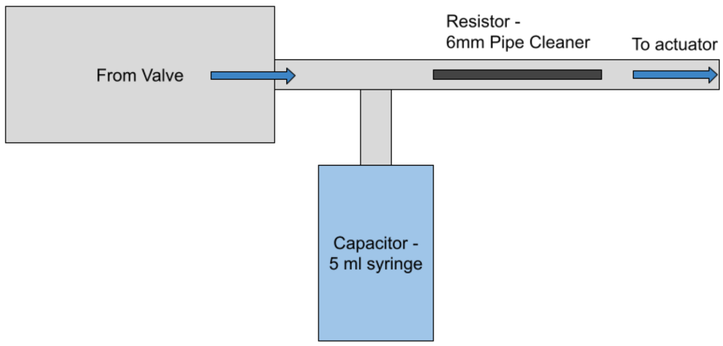


Figure 3.7: *Concept of pneumatic low pass filter using a pipe cleaner and syringe*

The author states that the resistor limits the maximum air flow, thus slowing down the flow rate and preventing large spikes in pressure, while the air tank would offer an additional reserve of air to prevent dips in the pressure. In their report as well as in Elgeneidy et. al. [13], the filter was shown to be successful, so a similar design was added to this system using a 5ml syringe and 6mm pipe cleaners. Testing the filter for this system, the results can be seen in the orange line of figure 3.8. As can be seen, the pressure has become more stable compared to the non-filtered version while the average pressure is nearly unaffected.

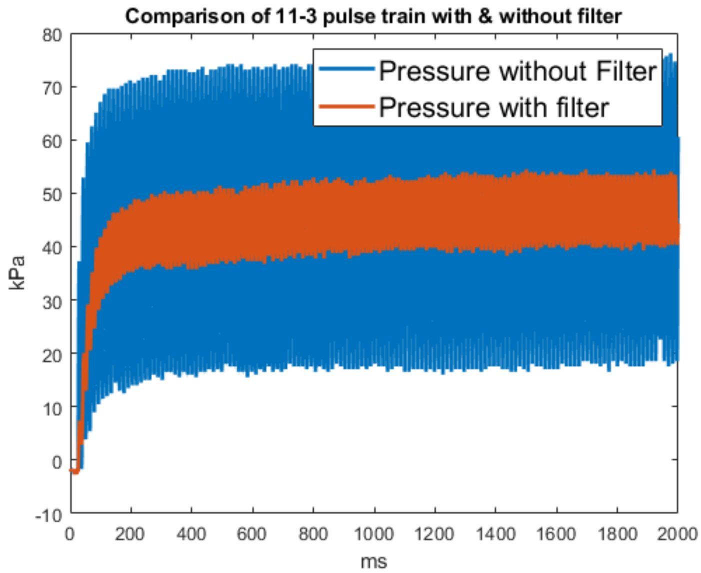


Figure 3.8: *Plot showing pressure during 11-3 pulse train. Blue shows unfiltered while orange shows the same experiment after the implementation of the pneumatic low pass filter seen in figure 3.7*

Having verified that the modified PWM control functioned and stabilised the pressure, the next step was to determine how to best control the pressure. To do this, tests were conducted where one of the parameters (on or off time) was set while the other was increased. Figure 3.9 and 3.10 illustrate the pressure response for a fixed on and off time respectively.

From the results shown in figure 3.9, it was determined that for a fixed time the pressure increases slightly with a decreasing off time, as can be seen with the blue line at the top having the shortest off time (2.5 ms) while the turquoise and green near the bottom having the longest off times (4 and 3.5 ms respectively). However the off time could not be set lower than 2 ms, else the valve would not have time to close leading to the previously mentioned issue of the valve locking open. This can also be seen in the plot with the blue 2.5 ms and orange 2.6 ms off-time plot being far more erratic and prone to peaking compared to the longer off times.

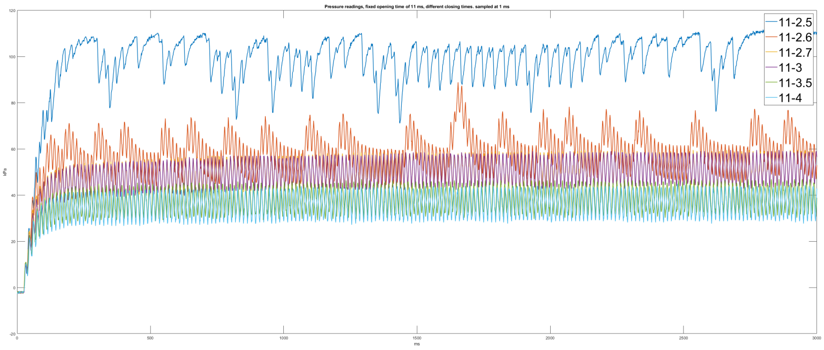


Figure 3.9: *Plot showing pressure response at fixed on time and increasing off time*

Fixing the off time meanwhile, shown in figure 3.10, there is a clear and steady relation between the on time and the mean pressure. While some instability did show, especially for the dark blue 9.5 ms on time, the results overall showed stable behaviour. There was also an advantage in terms of achievable pressure range. While there was a lower limit around 8.5-9 ms on time for the on time, any value between 9 and 15-17 ms would provide a steady response (except some spikes for 9.5) offering from 20 kPa up to the inlet pressure. However, it should be noted that there was also a noticeable increase in the pressure oscillations at higher on-times, as can also be seen in the plot with the lines higher up having a larger magnitude than the ones further down. Thus the choice was to fix the off time to 3 ms and use the on time as the control parameter, opting for a range of 9 to 17 ms on time. However, due to repeated actuator failures at higher pressures, it was decided to limit the range to a maximum of 11 ms when data for the neural network training was gathered.

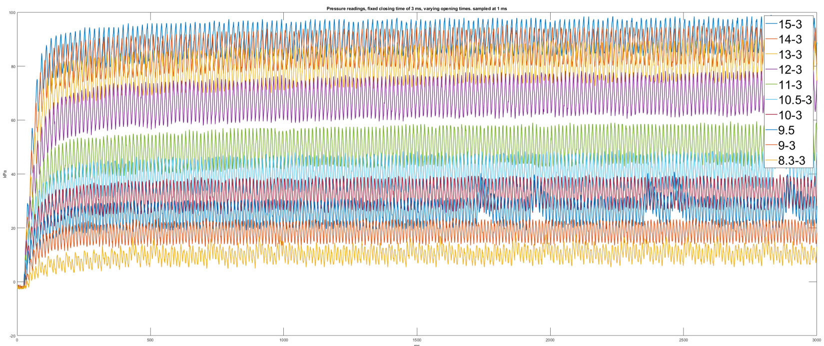


Figure 3.10: *Plot showing pressure response at a fixed off time and increasing on time*

3.3 Testing rig

For creating the ground truth, inspiration was taken from previous reports [6] [17], who utilised markers along the actuator body which they could then track using computer vision. As this project was developed in parallel with the bachelor thesis of Andersson et al. [29], who had created a computer vision system using aruco markers for tracking the MCP, PIP and DIP, they graciously shared their program for this project. In order to utilise this program, aruco markers were attached on the side of the actuator using elastic bands, which can be seen in figure 3.12. A rig was built in which the actuator could be placed for recording. The rig consisted of a bottom plate with a retaining clip for the base of the actuator, a lamp and a top plate with a hole for a GoPro camera. The camera records at a framerate of 120 fps, as higher frame rates caused issues due to file sizes. Thus this was also chosen as the sample rate for the sensors. An image showing the rig in use can be seen in figure 3.11. The angles could then be extracted from the video using Andersson et al.'s program and paired with observations from the system, thus yielding the target angles the network would attempt to estimate.

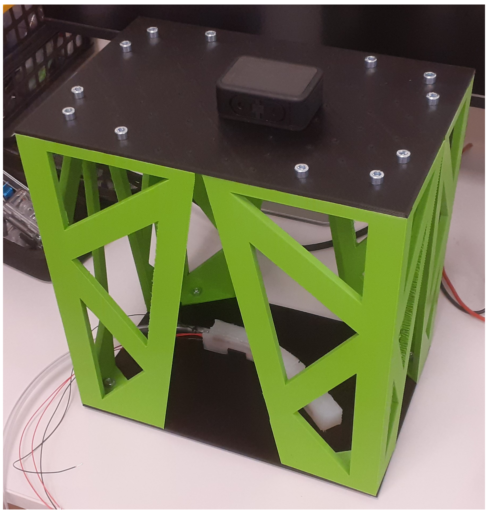


Figure 3.11: *Photograph showing the vision rig used to record the actuator movement. Videos are recorded from above using a GoPro looking down through a hole in the top plate. The actuator is placed inside, though unconnected in this case*

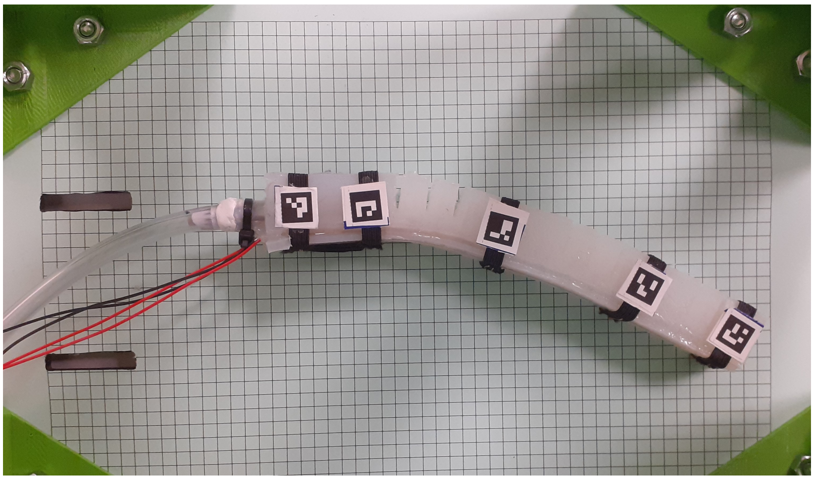


Figure 3.12: *Photograph showing the view through the top plate of the vision rig. The aruco markers placed along the actuator body are used to determine the position and angle of the joints. The actuator is held in place via a retaining clip near the base. Note that in this photo the overhead lighting is not on, hence the shadows*

It should be noted here that the design of the vision rig, similar to the one used by Tennakoon et. al. [17], placed the actuator on a horizontal plane or plate rather than suspending the actuator, such as Yap et. al. [6], Elgeneidy et. al. [13] and Wang et. al. [7]. This was chosen to avoid the impact of gravity as well as being seen as a more practical method. The drawback of this design however was that it introduced friction into the system which could be difficult for the network to learn.

4

Testing

In this chapter, the process of gathering data and training the neural networks, as well as the method for evaluating network performance will be explained. First, the early testing to characterize the system and gather data for the network training is presented. After that, the process of creating the neural networks, including network size, activation functions and parameters is explained. This is then followed by the networks or groups of networks which would be tested, mainly focusing on what features to add and whether to train a single network or train the joints separately. Finally, the method which was used for evaluating the network performance is presented.

4.1 Data analysis and actuator behaviour

It has been reported in numerous publications that when modelling soft pneumatic actuators their behaviour is highly nonlinear [1],[7],[13]. This is both in regards to the actual angle to inflating pressure and also the rate of change in relation to pressure. An additional source of non-linearity in this project was the inclusion of friction brought about by the actuator being tested whilst placed on a plate rather than being suspended.

With this in mind, it was deemed important to analyze the behaviour of the segmented actuator under these conditions. In order to determine whether the behaviour differed between different motions, several datasets were created which can broadly be categorized as steps, ramps and cycling ramps. In the step datasets, the on-time [1] was first set to 9 ms for some time, then raised to a different value before being set back to 9 ms. 9 ms was chosen as it was the lowest the system could stably maintain. An example can be seen in figure 4.1.

[1] On time refers to the time during which the PWM signal is set to high, see 3.2.2

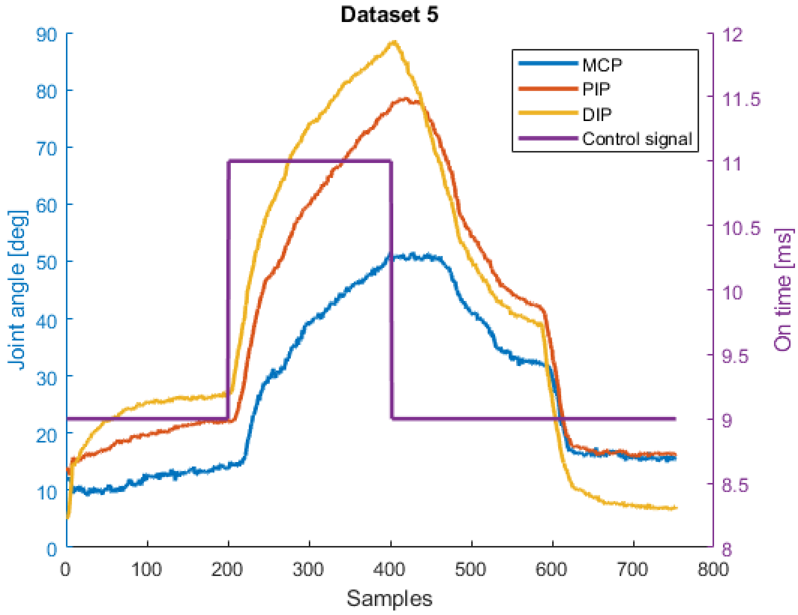


Figure 4.1: *Plot showing dataset 5, a step movement. Plotted are the three vision angles alongside the control signal. Unit for the angles are in degrees (left axis) and milliseconds (right axis)*

In ramp datasets, on time was once again set to 9 ms before being gradually increased by 0.5 ms every second, two seconds or half a second until reaching 11 and then decremented at the same rate. An example of a ramp movement can be seen in figure 4.2 below.

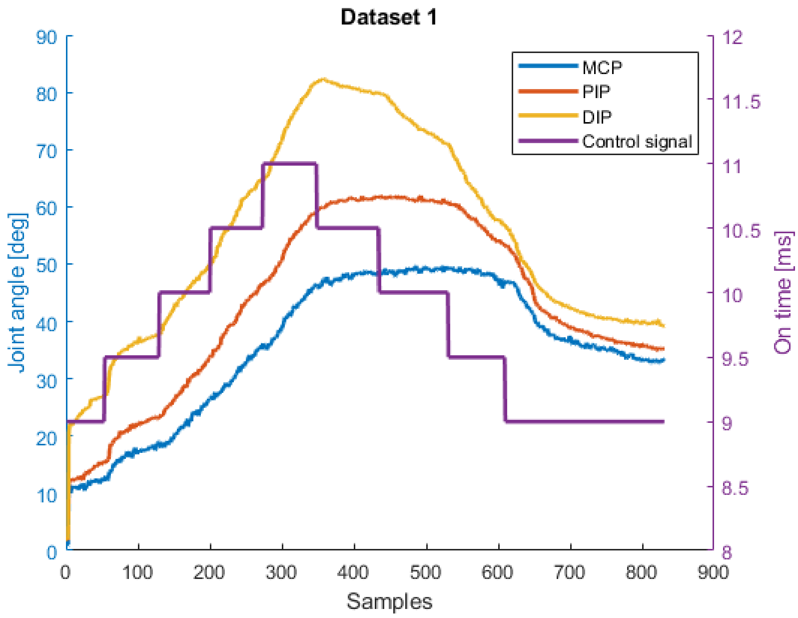


Figure 4.2: *Plot showing dataset 1, a ramp movement which incremented at a rate of 1 second. Plotted are the three vision angles alongside the control signal. Unit for the angles are in degrees (left axis) and milliseconds (right axis)*

Something which can be observed in both figure 4.1 and 4.2 is that when comparing the angles near the beginning and the end of the plot the angles did not return to the same values. Presumably, this was due to the friction induced by the vision rig. This raised questions about whether the joint data would be consistent over multiple flexions and extensions since they would begin at a different starting angle. To evaluate this, the cycling ramp motion was created. In these, the on-time was once again started at 9 ms, then increased by 0.5 ms every second up to a set maximum value before being decremented at the same rate. Once it reached 9 again, the cycle would recommence. A cycling ramp movement from 9 to 10 ms can be seen in figure 4.3 below.

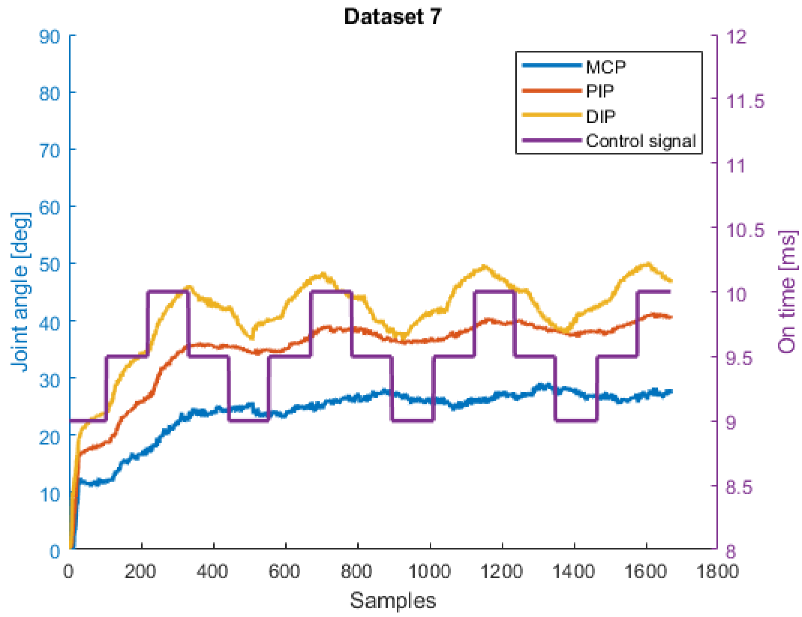


Figure 4.3: *Plot showing dataset 7, a cycling ramp movement. Plotted are the three vision angles alongside the control signal. Unit for the angles are in degrees (left axis) and milliseconds (right axis)*

A complete list of all datasets can be seen in appendix B. Video recordings are also available at https://www.youtube.com/playlist?list=PL2-X-IUuVJh0aJB6ppWigDhvhaM4RHkWy

One aspect of the behaviour which was of interest was whether the bending motion varied depending on whether the actuator was undergoing flexion or extension. In the report by Elgeneidy et. al. [13], figure 8 to be precise, it could be seen that the inflating pressure-flex sensor reading relationship had two distinct curves, though it returned to the same initial value afterwards. This led to speculation that the actuator used in this work would exhibit similar behaviour. A plot of the pressure-vision angle relationship during a ramp movement can be seen in figure 4.4 below.

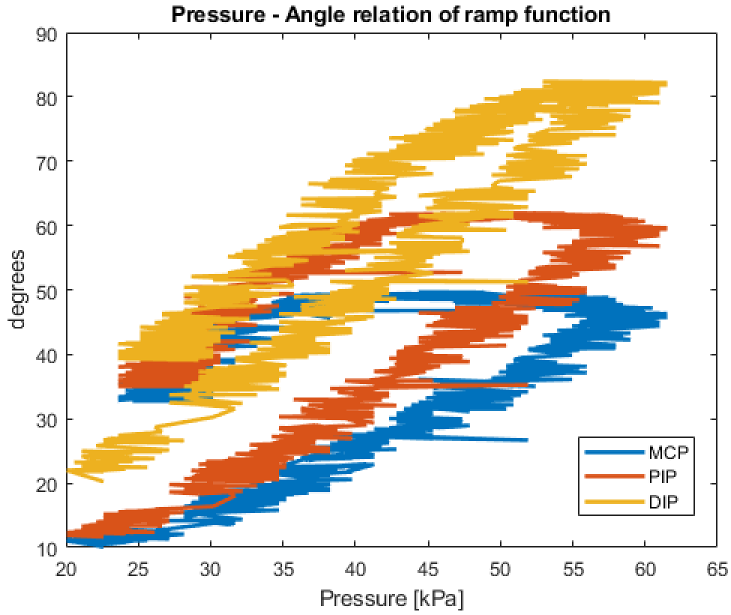


Figure 4.4: *Joint angles recorded by vision system plotted vs pressure during a ramping behaviour. Note that the X-axis begins at 20-25 kPa, as that is the lowest pressure the system can maintain*

In line with the plot from Elgeneidy et. al. [13], the relationships between inflating pressure and the joint angles were non-linear. It can be seen in figure 4.4 that the angles, though less stable than in their plot, still followed two distinct lines. This corresponded to the flexion and extension phase of the ramp. From this, it could be seen that the actuator exhibited different behaviours depending on whether it was in flexion or extension. It can also be seen how the actuator after the flexion part of the movement had been completed, remained stationary for some time during the extension part. This is especially clear for the MCP (blue in 4.4). Between 57 kPa and 40 kPa, the MCP joint remained stationary. After the pressure had decreased sufficiently, the angle would then follow a different line with roughly the same gradient but with an offset. Finally, as can be seen in in figure 4.4 at 20-25 kPa the resting state had changed after flexion, with the angles at the "ends" of the plot altering from 10 to 34, 10 to 35 and 20 to 40 degrees for the MCP, PIP and DIP respectively. This was not in line with the results of Elgeneidy et. al. [13], where the actuator returned to the same resting state after flexion. Presumably, this was due to their actuator being suspended rather than placed on a surface, leading to friction holding the actuator in place.

4.2 Neural network model

For creating the neural networks, Matlab's Deep Network Designer app was used. Initially, the network structure was based on the one utilised by Elgeneidy et. al. [13]. In their paper, a network consisting of a single hidden layer with 7 neurons was used. However, as the activation function was not stated, all activation functions available in Matlab were tested.

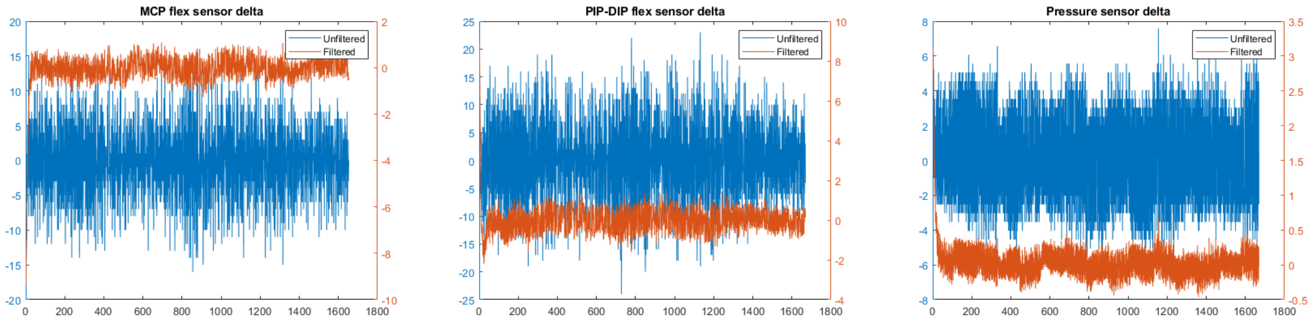
In regards to the datasets used, it was decided to use cycling ramp datasets as the validation and testing datasets as they were deemed to be the least similar to the training dataset and each other. Thus datasets 6 and 7 were to be used for validation and testing respectively. For the training data, the remaining datasets were used. In total, the training data thus consisted of three ramps, four step functions and two cycling ramps. However, it was found during testing that the addition of the step function datasets in the training set negatively impacted the training process, with networks containing step functions tending to have a large spike in the first few predictions. They were therefore removed from the training data.

As the required size of the network and the optimal parameters were not known, testing of different structures and values began. After testing one, two, five and ten hidden layers with varying numbers of neurons and using different activation functions it was found that the best results were found for two hidden layers with ten neurons each using elu activation functions. During this time, other parameters were also gradually decided upon. Batch sizes were increased to 500, max epochs to 400, initial learning rate was set to 0.001 with a decrease of 0.75 every 80 epochs. A batch normalization layer was also added to normalize the input data for each feature. However, the choice of activation function was less clear. Therefore, all permutations of activation function would be tried during testing, resulting in 24 networks per test.

Having decided on the network structure, the first test was in regards to a finding of Elgeneidy et. al. [13]. In their report, they showed that the model improved as more sensor information was added. They initially trained using only the flex sensor data, which in their testing could with some errors track the angle. Then adding the pressure and finally adding the orientation of the base of the actuator (as it was suspended), the error of the model decreased with each additional source of information added. To test if this held for the segmented actuator, a similar test was performed. Networks were trained using only pressure data, only flex sensor data and finally using both pressure and flex sensor data.

4.2.1 Networks trained using sensor delta

Based on the behaviour seen in figure 4.4, it was concluded that the direction of travel does impact the expected behaviour. It was therefore reasoned that to achieve better tracking the model requires not only the current sensor values but also knowledge of the previous state the actuator was in so that it might determine which trajectory it is likely to follow. To investigate this theory additional input data was added in the form of the the delta of the pressure as well as of the flex sensors. However, the delta signals were seen as quite noisy and their sign did not correlate to actuator movement, as was initially hoped. This can be seen in figures 4.5a, 4.5b and 4.5c where the three delta signals are plotted in blue. These correspond to the same ramping dataset seen in figure 4.2. In an attempt to reduce the noise in the signals, they were filtered using an exponential filter. These filtered signals are plotted in orange.



(a) MCP flex delta **(b)** PIP+DIP flex delta **(c)** Pressure delta

Figure 4.5: Plots showing the three delta signals (blue) as well as their filtered counterparts (orange)

This however did not show the behavior that had been hoped for but it was decided to test them regardless since the network could potentially still glean information from it. This resulted in four potential delta inputs: delta pressure, filtered delta pressure, delta pip flex and delta dip flex as well as filtered delta pip flex and filtered delta dip flex. Thus networks with every permutation of activation function were trained using each of these four delta variants as well.

4.2.2 Separately trained networks

Another discussion concerning network training was whether to train one network to solve for all three angles or to train a network for each joint. This concept was conceived after observing a phenomenon in the predictions of the early networks where the estimates appeared linked in some fashion. Despite the three angles from the vision system behaving quite differently, the predictions from the neural network model all appeared to follow a single line with an added offset and dampening. It was then hypothesised that perhaps the network was attempting to optimize for all three joints rather than separating the neurons and training specific neurons for each joint. If so, it was hoped that a network for each joint would firmly separate the models and resolve this issue. For this, two solutions were tested; Three separately trained networks and to separate the networks based on which flex sensor overlaps that joint, i.e. a network trained for the MCP joint and another network trained to solve both PIP and DIP. Thus the above-mentioned tests using all different permutations of activation functions and all four delta inputs were conducted with these two solutions.

4.3 Network evaluation method

In order to evaluate the quality of the networks, the averaged RMSE of the three joints was the primary criterion used. This was chosen as RMSE was seen as a standard measurement that was also simple to implement and would provide a quantitative measurement as opposed to visual analysis. Thus, after the network had been trained it would be tested on the cycling ramp dataset nr.7 (see figure 4.3) using Matlab's predict function. The RMSE would be calculated for each joint and then the network RMSE would be calculated as the sum of these values divided by three.

$$RMSE_{Network} = \frac{\sum_{i=1}^{3} RMSE_i}{3} \tag{4.1}$$

For evaluating a group of networks, such as the single network trained using the delta of the pressure, the above-mentioned scores were calculated for all permutations of delta and the best five were considered. From these scores, the mean x and standard deviation s were calculated, which then became the group's score. For the models using multiple networks, the five best scores for each network would be taken and paired with their respective from the other networks, i.e. the best MCP network would be paired with the best PIP and the best DIP. The mean and standard deviation were then calculated.

$$x = \frac{\sum_{i=1}^{5} RMSE_{Network,i}}{5} \tag{4.2}$$

$$s = \sqrt{\frac{1}{5-1}\sum_{i=1}^{5}(RMSE_{Network,i} - x)^2} \tag{4.3}$$

This method was chosen to test the potential benefit of the separated networks to utilise different activation functions and past information for each joint.

5
Results

In this chapter, the results of the different neural networks described in the previous chapter will be presented. First, the network using the two flex sensors as well as the air pressure as inputs will be shown as the baseline for all networks. Then the networks which utilise past information, i.e. delta values, will be presented and finally, the two separately trained network solutions will be presented. For the sake of brevity, only the plotted prediction from the network with the lowest average RMSE will be shown for each category. However, a complete list of all test results can be found in appendix A

It should also be noted that from this point forward, networks will be referred to according to their activation functions. For example, Elu-Swish refers to the network with an elu activation in its first hidden layer and a swish activation function in its second hidden layer.

5.1 Elgeneidy based network

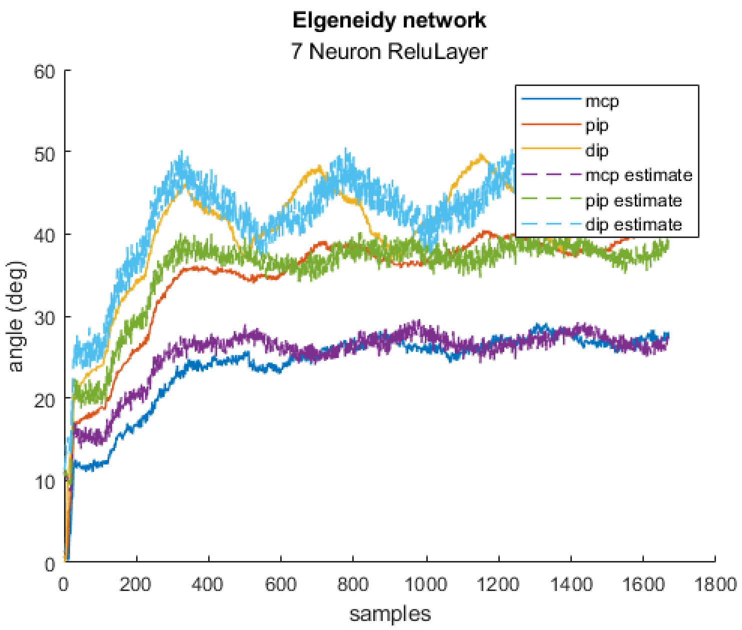


Figure 5.1: *Network structure utilised by Elgeneidy et. al. prediction on dataset 7*

5.2 Single sensor networks

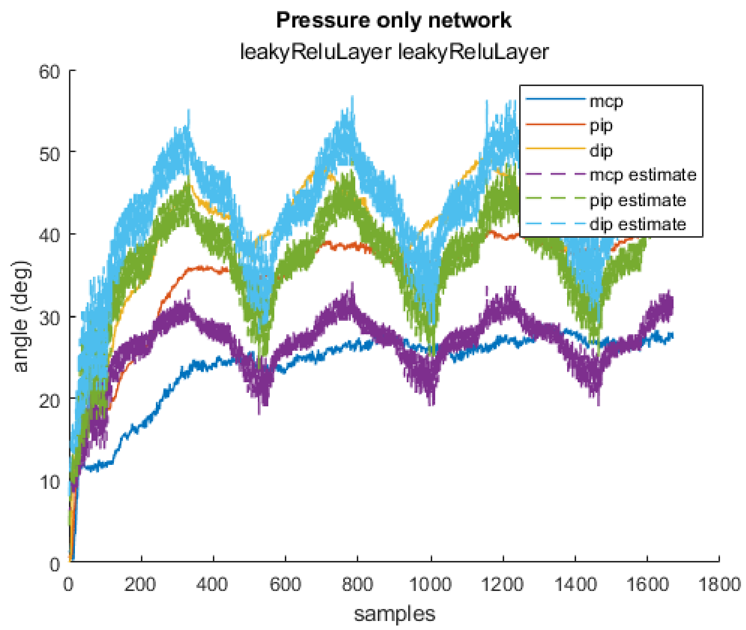


Figure 5.2: *Network with lowest error using only pressure data*

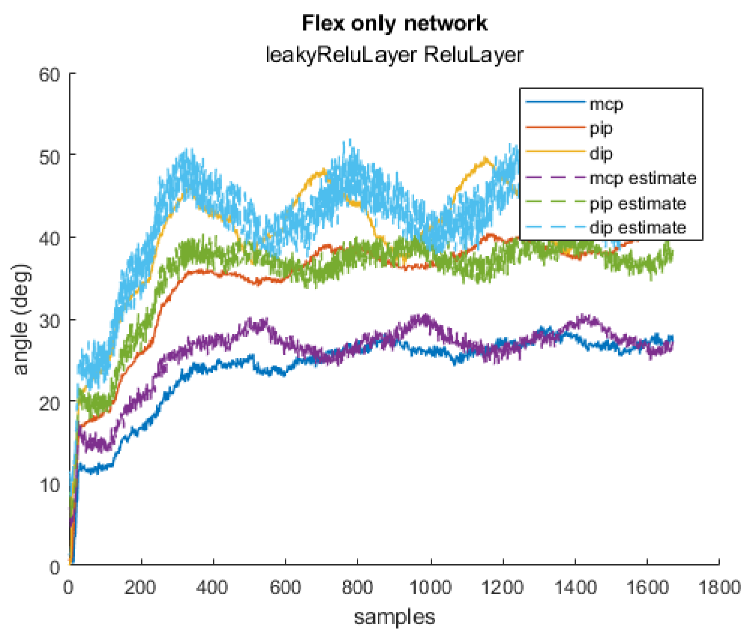


Figure 5.3: *Network with lowest error using only flex data*

5.3 Neural network model without past knowledge

This experiment resulted in 36 networks. However, after it had become clear that networks with neither a sigmoid nor tanh activation function in the second layer would converge they were excluded. Thus 24 networks would be used for each group moving forward. Taking the mean and standard deviation from the five best networks yielded an RMSE of 2.7041 ± 0.0251. The network with the lowest averaged RMSE was the Swish-Swish (2.6700 averaged RMSE), whose prediction can be seen in figure 5.4. Here, the joint angles captured by the vision system are plotted alongside the predicted angles from the neural network model.

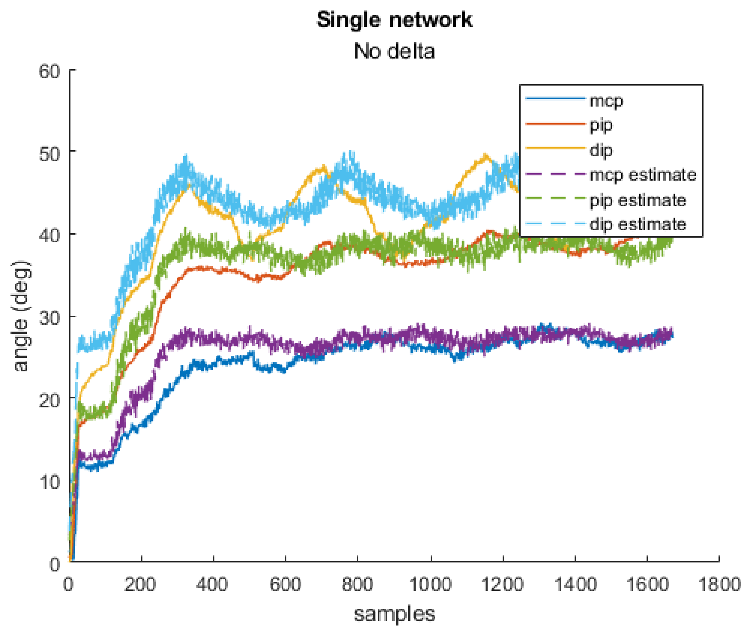


Figure 5.4: *Prediction from Swish-Swish network without past knowledge, which would act as the benchmark moving forward*

5.4 Neural models with past knowledge

As mentioned in section 4.2, four sources of past knowledge were devised from the measurement data: Δflex, Δpressure, filtered Δflex and filtered Δpressure. Evaluating the networks trained on each of these yielded the following four groups.

For the networks which included Δpressure, the network with the mean and variance was 2.6070 ± 0.0398. lowest averaged RMSE was the Relu-Relu network (2.5476), whose prediction can be seen in figure 5.5

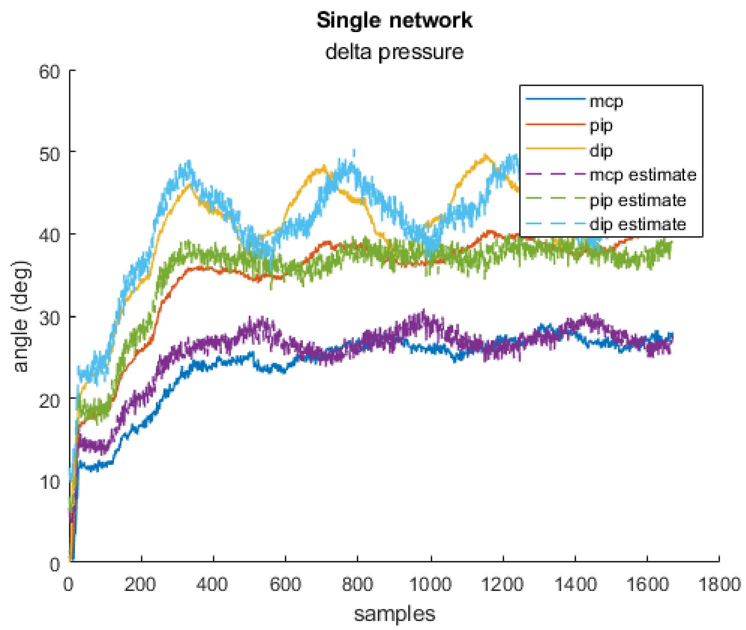


Figure 5.5: *Prediction from Relu-Relu network fed with* Δ*pressure*

For the networks fed with the delta of the two flex sensors, the mean and variance were 2.7376 ± 0.034. The lowest single was the Sigmoid-Swish (2.6861) network. Once again, the Sigmoid-Swish network can be seen in figure 5.6.

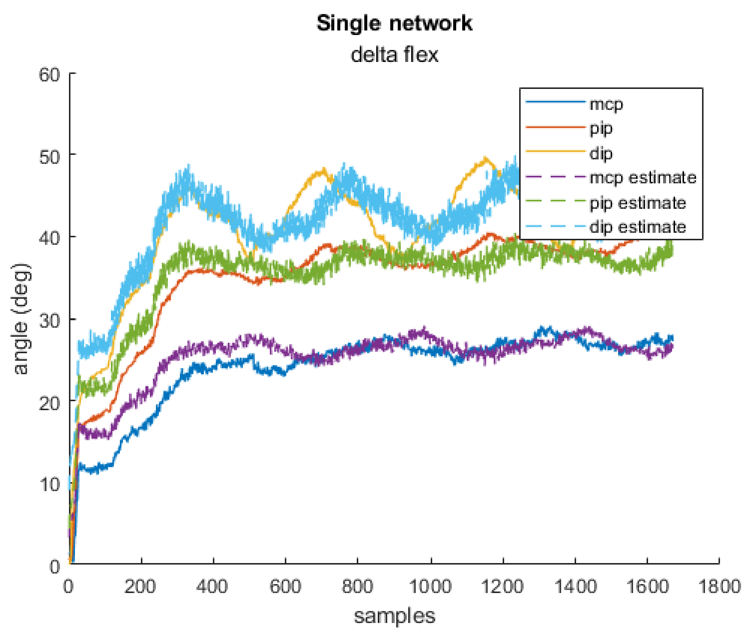


Figure 5.6: *Prediction from Sigmoid-Swish network fed with* Δ*flex*

When filtering the two delta signals, the results were as follows. For the filtered Δpressure the mean and variance was 2.5333 ± 0.09404. The network with the lowest error was the Swish-Swish (2.4136), which also was the lowest score of all networks.

For the filtered Δflex the mean and variance was 2.6341 ± 0.0396 with the lowest score being the Swish-Swish (2.5665). The predictions from the best-performing network using Δpressure and Δflex can be found in figure 5.7 and 5.8 respectively.

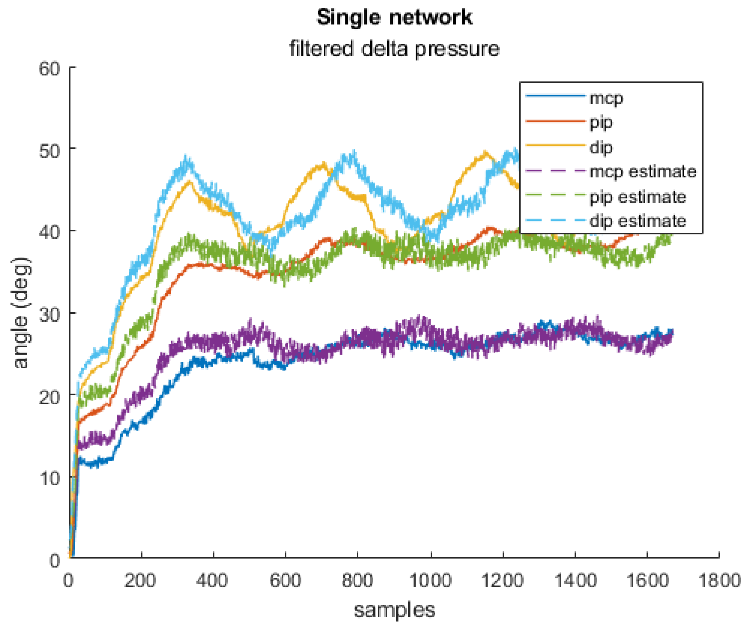


Figure 5.7: *Prediction from Swish-Swish network fed with filtered Δpressure*

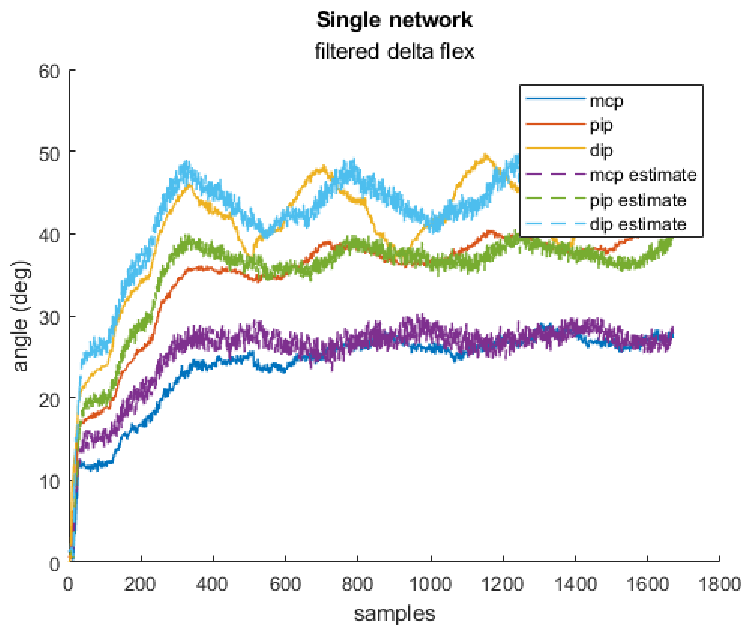


Figure 5.8: *Prediction from Swish-Swish network fed with filtered Δflex*

5.5 Separated neural network

As before, results from the group without past knowledge and the four past knowledge options will be presented separately. Because these networks are trained individually the lowest combination of activation functions for each joint will be presented.

For the networks without past knowledge, the mean and variance were 2.7356 ± 0.0659 with the lowest combination being Relu-Relu (MCP), Elu-Relu (PIP) and Relu-Swish (DIP), with an averaged RMSE of 2.6494.

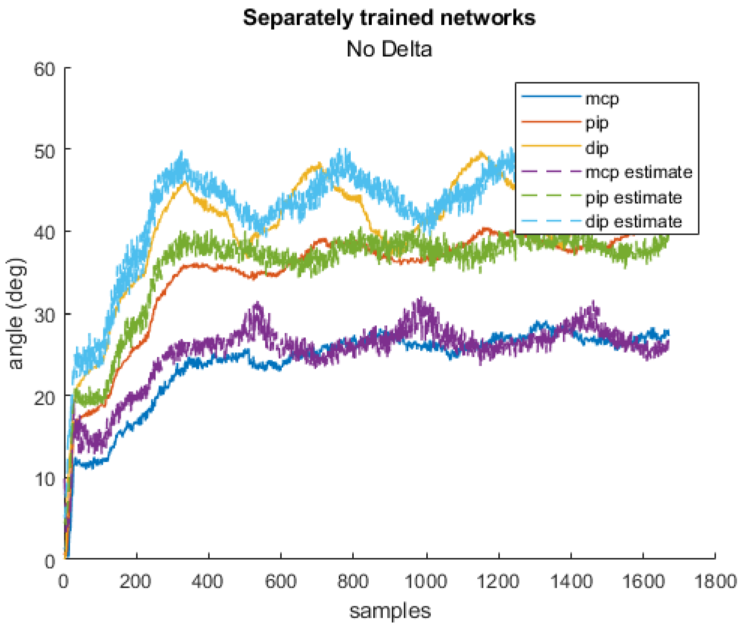


Figure 5.9: *Prediction from separated networks without past knowledge*

For the networks with Δpressure, the mean and variance were 2.5930 ± 0.0257. The lowest combination was Sigmoid-Leaky Relu, Leaky Relu-Swish and Swish-Leaky Relu with an average RMSE of 2.5546.

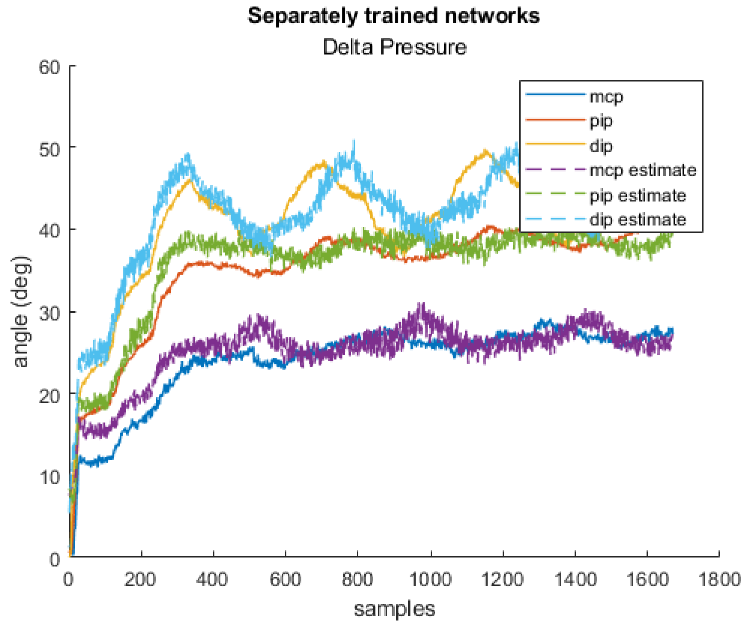


Figure 5.10: *Prediction from separated networks with ∆pressure*

For the networks with ∆flex, the mean and variance were 2.7548±0.0337. The lowest combination was Leaky Relu-Elu, Swish-Elu and Leaky-Relu with an average RMSE of 2.8947.

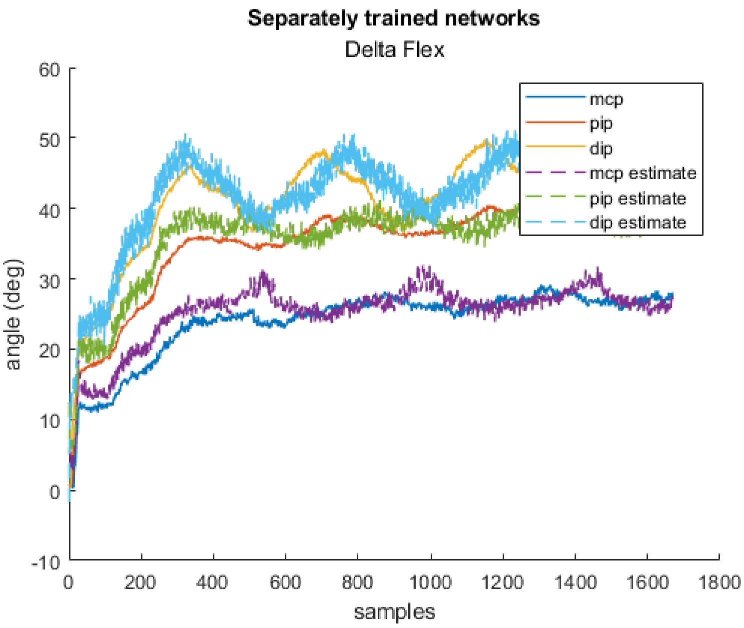


Figure 5.11: *Prediction from separated networks with ∆flex*

For the filtered ∆pressure, mean and variance was 2.5718 ± 0.0525, with the lowest combination being Leaky-Elu, Elu-Elu, Relu-Swish with an averaged error of 2.6719

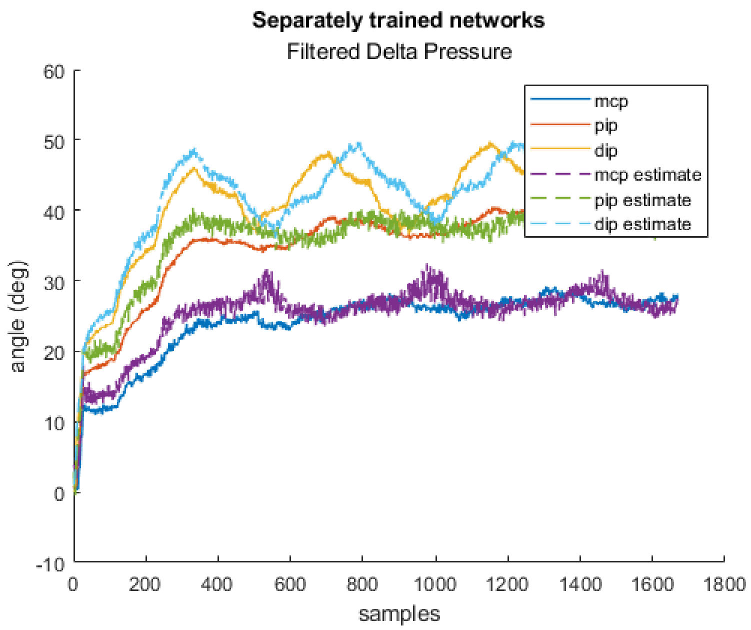


Figure 5.12: *Prediction from separated networks with filtered* Δ*pressure*

For the filtered Δflex, the mean and variance were 2.6228 ± 0.0682. The lowest combination was Relu-Elu, Elu-Leaky Relu, and Elu-Swish with an average error of 2.5663

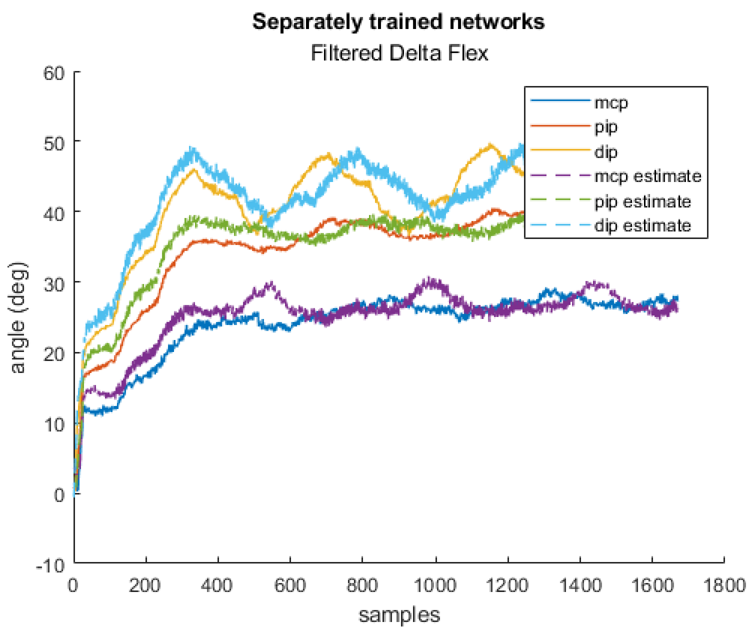


Figure 5.13: *Prediction from separated networks with filtered* Δ*flex*

5.6 Combined PIP DIP network

Finally, the group of networks which trained the MCP individually and the PIP and DIP together. Again, since each prediction contains two networks, the best combination of activation functions for each will be presented.

For the networks without past knowledge, the mean and variance were 2.7107 ± 0.0439. The lowest combination was Relu-Relu (MCP) and Relu-Elu (PIP+DIP), with an average RMSE of 2.6533

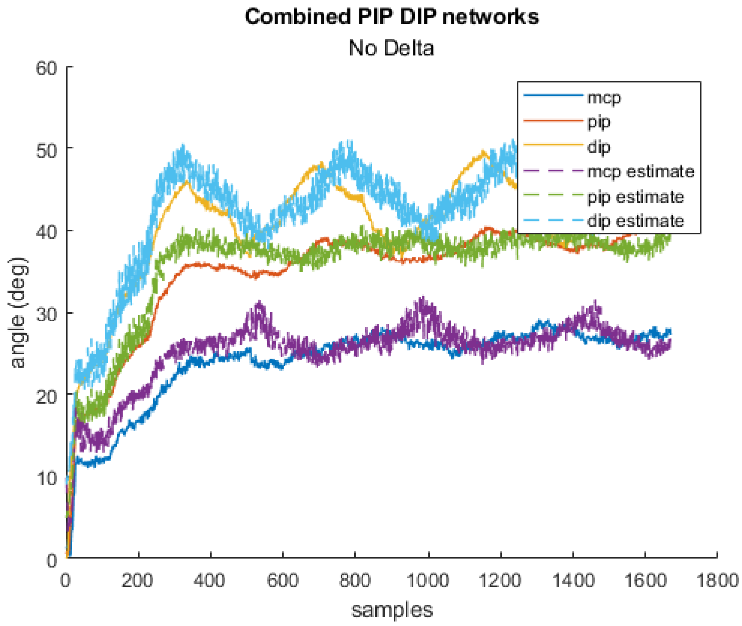


Figure 5.14: *Prediction from combined PIP DIP network without* Δ

For the networks with Δpressure, the mean and variance were 2.5674 ± 0.0363. The lowest combination was Sigmoid-Leaky Relu and Relu-Elu with an averaged RMSE of 2.5127.

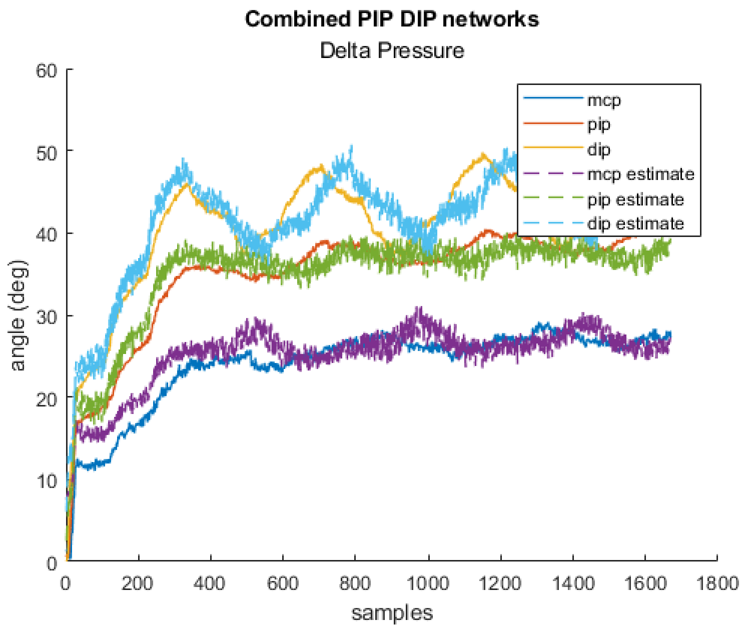


Figure 5.15: *Prediction from combined PIP DIP network with* Δ*pressure*

For the networks with Δflex, the mean and variance were 2.6994 ± 0.0343. The lowest combination was Leaky Relu-Elu and Elu-Relu with an averaged RMSE of 2.6499.

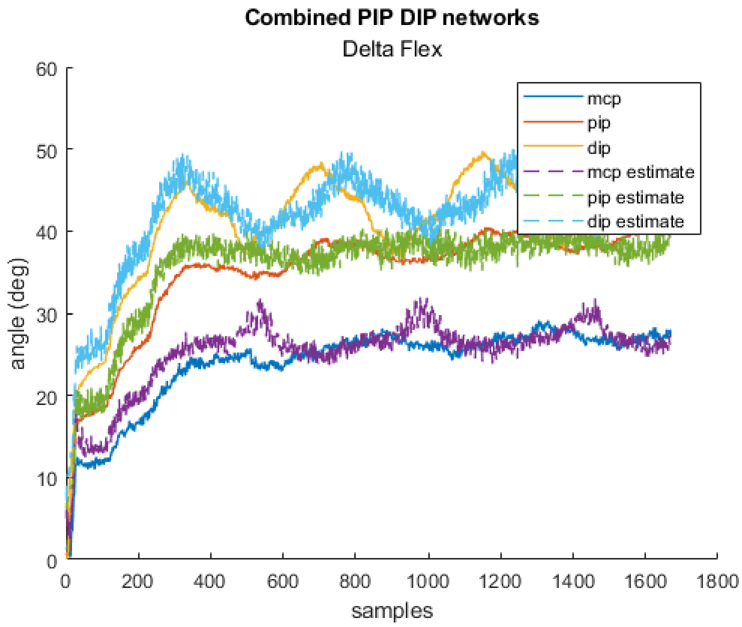


Figure 5.16: *Prediction from combined PIP DIP network with Δflex*

For the filtered Δpressure, the mean and variance were 2.5595 ± 0.046 with the lowest being Leaky-Elu and Elu-Elu with an average error of 2.4973.

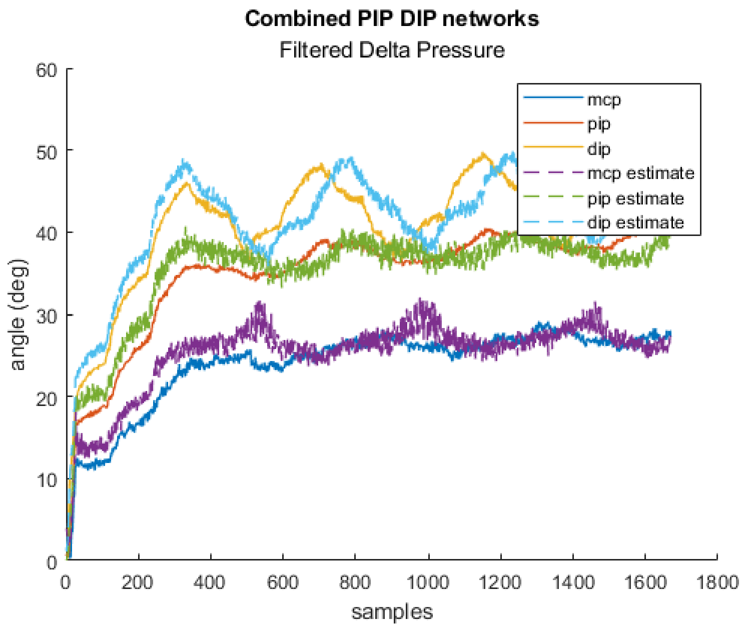


Figure 5.17: *Prediction from combined PIP DIP network with filtered Δpressure*

For the filtered Δflex, the mean and variance were 2.6084 ± 0.0374. The lowest combination was Relu-Elu and Relu-Elu with an average error of 2.5438.

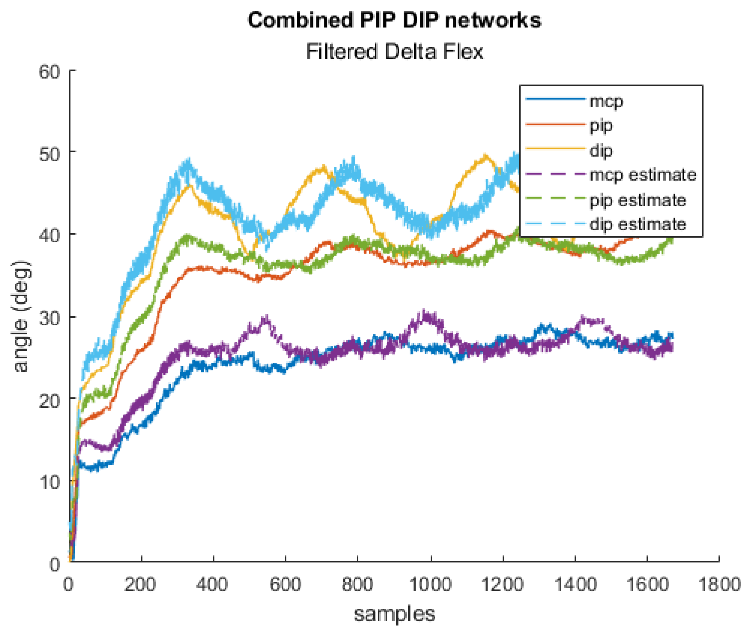


Figure 5.18: *Prediction from combined PIP DIP network with filtered* Δ*flex*

A summary of the mean and variance for each score can be found in table 5.1 below. The complete set of tables containing the scores for all networks can also be found in appendix A

	Single network	Separated network	Flex sensor separated
No Δ	2.7041 ± 0.0251	2.7356 ± 0.0659	2.7107 ± 0.0439
ΔPressure	2.6070 ± 0.0398	2.5930 ± 0.0257	2.5674 ± 0.0363
ΔFlex	2.7376 ± 0.034	2.7548 ± 0.0337	2.6994 ± 0.0343
Filtered ΔPressure	2.5333 ± 0.0904	2.5718 ± 0.0525	2.5595 ± 0.046
Filtered ΔFlex	2.6341 ± 0.0396	2.6228 ± 0.0682	2.6084 ± 0.0374

Table 5.1: Summary of test results, showing the mean and variance taken from the five best networks for each network structure and delta. Rows show what delta the networks were trained with and columns show methods of joint network separation.

6
Discussion

In this chapter, the results will first be analysed to discuss what trends can be seen and what conclusions can be drawn regarding the advantages and disadvantages of the different methods. These will then be used to answer the research questions. Then a broader discussion on the project will follow, including how certain decisions may have influenced the results, what faults lie in the system and finally what future projects can or should focus on based on the conclusions and experiences of this project.

6.1 Analysis

In this section, observations drawn from the results presented in chapter 5.1 will be presented. Following that, a brief discussion of the research questions is presented.

6.1.1 analysing data and figures

Results from the tests with only one sensor input indicate that predicting the angles using only the pressure was not feasible. However, similar to the findings of [13] the networks trained only using the flex sensor data were able to track the joint angles. In particular, the MCP joint was very promising, having both little noise and capturing the shape of the true line well. Meanwhile, the prediction made by the network based on Elgeneidy et. al. was quite promising. Although with an RMSE of 2.8056, it was still higher than all the test groups. This does suggest that increasing the network to two layers does benefit the model.

6.1.1.1 Single networks

Looking towards the single network with no delta input (figure 5.4) as the baseline network, a few observations can be made. Firstly, the DIP prediction is lacking, being unable to follow the shape of the true angle. This is especially noticeable near the dip at c.a. 900. The second observation that can be made is the delay between the vision angle and the estimate after the first peak. During the following decrease and until the end of the test, there is a delay in the vision system. This is most noticeable when analysing features such as the DIP peak at ca. 700 or the beginning of the rise of the PIP at ca. 500. Thirdly, the tracking around the initial ramp-up period (zero to ca.300) has an offset from the vision angle, although there is a noticeable offset. When consulting table 5.1 it is clear that the single network with no delta is not the optimal solution as only two out of the fourteen other networks have a higher error.

In comparison to the network trained only using flex sensor data (see figure 5.3), it is clear that the combination of both flex and pressure data does yield the best result. Compared to the previous benchmark, The DIP is very unstable and the PIP is also slightly less well defined. Tracking ability appears comparable except for the DIP tracking, where the magnitude of the DIP drop is far better. It should also be noted that the MCP is quite promising: being far more well-defined and stable even if the offset is perhaps slightly too high. If one were to speculate, the reason for the stabilization caused by two sensors could possibly be due to the noise of the pressure sensor being relatively low compared to the flex sensors. Referring to figures 4.5a 4.5b and 4.5c, where the delta of each signal is shown in blue. On the blue axis to the left of the figure it can be seen that the pressure varies by $\pm$ 4-5 mostly, while the MCP varies by $\pm$ 7-8 and the PIP+DIP varies by $\pm$ 10-12.

Continuing with the single networks but with some form of past information, from table 5.1 it can be seen that three out of the four delta options resulted in lower errors. Visually, the addition of the delta has changed the behaviour of each model in different ways. For the networks fed delta pressure (figure 5.5) we can see that the DIP prediction is far more stable as well as tracking better. The MCP prediction has also been improved as it better follows the shape of the vision angle, though the offset appears somewhat worse. For the networks fed delta flex (figure 5.6) we can see that the MCP prediction is very stable and that similar to the delta pressure case, the DIP better follows the vision angle, though it is slightly less stable. As for the filtered inputs, for the filtered delta pressure (figure 5.7) the DIP estimate is very stable and follows the shape very well. The PIP and DIP appear largely unchanged. Finally, the filtered delta flex (figure 5.8). Here the PIP is very stable and follows the shape well, the DIP follows the shape but can not follow the valleys and the MCP is quite poor, being both flat as well as less stable than the benchmark.

Overall, the combined networks with some form of past knowledge added all show potential. The addition of a delta did seem to lead to either better tracking or more stable predictions, though none of the networks were able to produce a stable and accurate prediction for all three joints. Also, none of these networks were able to resolve the delay issue.

6.1.1.2 Separated networks

Beginning with the separately trained networks without delta (figure 5.9), as can be seen in table 5.1 the error has risen compared to the benchmark single network without delta. When comparing the predictions, it does seem that the DIP dips slightly lower compared to the combined network. However, in the MCP prediction, the peaks have been greatly amplified. This issue of overestimating the MCP peaks holds for all the separately trained networks. At around 900-1000 one can also see that the PIP does not decrease, another behaviour seen in most of the separated networks.

Moving onto the networks provided past information, out of the four delta options we

can see in table 5.1 that three performed better than the separated network without delta as well as the benchmark. For delta pressure (figure 5.10), we can see that the DIP is far more stable and reaches appropriate magnitudes during the valley. The PIP appears to be unchanged but the MCP peaks have been dampened somewhat. In this case, the addition of a delta has benefited the network. Continuing to the delta flex network (figure 5.11), the error is substantially higher. Despite this, the plot shows that the MCP is very stable, though still with amplified peaks and the DIP, while it does follow the true angle better than the no delta, is more unstable. As for the filtered delta pressure (figure 5.12), which achieved the lowest score of all the separately trained networks, we see the best tracking of the DIP out of all separated networks, being both very stable and good tracking. The PIP appears largely unchanged from the no delta and the MCP is also mostly unchanged. Finally, the filtered delta flex (figure 5.13). Again, we see that the DIP is quite stable, the PIP is very stable and tracks quite well even if there is an offset and the MCP is also stable, though the peaks remain high. It should be noted that this is one of the only network(s) where all three joints have very stable predictions. However, while they are very stable only the DIP tracks the movements well.
Overall, it does seem that the ability of the network to model the joints varies. The DIP predictions were all quite good, the PIP was acceptable but the MCP was worse than for the single network. It would also seem that for separately trained networks the filtering of the delta signal lets them significantly reduce the noise.

6.1.1.3 Flex sensor separated networks

Finally, the networks are separated according to which flex sensor they interact with. As these networks utilise the same MCP networks as in the corresponding separated network, they will be ignored for now as their issues have already been discussed. Firstly, when comparing the network without delta (seen in figure 5.14) to the benchmark in table 5.1, the score is marginally worsened. However, it is an improvement over the three separated networks. Visually, there is not much difference except for the aforementioned MCP issue.

Then regarding the networks using delta, for the first time all four performed better than the one without past information. Beginning with the delta pressure (figure 5.15), the DIP has become more stable and the PIP has also improved with the offset around 200-500 has been corrected. As for the delta flex (figure 5.16), as indicated by the RMSE values it remained largely unchanged, except for the PIP becoming more noisy. Moving onto the filtered delta pressure (figure 5.17, which scored the lowest of the flex sensor separated networks, the DIP has been further stabilized which is now comparable to the ground truth. The PIP has also become slightly more stable and though it is difficult to judge appears to be tracking the decrease between 700-1000. Finally, the filtered delta flex (figure 5.18). Here we see that while the DIP has become less noisy it still is not quite able to track the dip around 900. The PIP meanwhile is very stable as well as tracking quite well, even managing to track the decrease between 700-1000 which so few networks have been able to do.

Overall, this separation does seem to benefit the modelling compared to the three separated networks, as for all the delta options except the no delta the flex sensor separated networks performed better. They all still suffer from the issues of MCP tracking discussed for the separated networks and except for the filtered delta pressure, all exhibit the issue of tracking the PIP decrease. The filtered delta flex is possibly the most stable of all the networks shown while the filtered delta pressure showed both good tracking of all joints except MCP and moderate to excellent stability of all joints.

6.1.1.4 Does delta help

When considering the results of table 5.1 it is clear that the addition of past information does benefit the network models since for each column (i.e. each type of network separation) there is at least one network using delta which performed better than the one without. Also, out of the twelve networks which used delta only two scored higher than the respective no delta network.

It also seems that the four delta options affect the model in different ways. The delta flex options performed worst in all tests and scored the two highest errors of all groups. However, in the predictions of the delta flex for both the single network and the separated networks, seen in figure 5.6 and 5.11 respectively, it can be seen that the delta flex predictions in both cases have practically filtered out the noise for the MCP prediction. However, it should be noted that the individually trained networks did manage comparable results using the filtered delta flex as well. This does seem to indicate that the delta flex, filtered or not, does contain beneficial information for stabilizing the MCP joint but lacks some other information the others provide which helps with the tracking and stabilizing of the other joints.

When analysing the two types of separated networks it also became clear that filtered past information resulted in less noisy predictions, as the predictions in figure 5.12, 5.13 as well as figure 5.17 and 5.18 show the estimates are far less noisy than their respective non-filtered or no delta counterparts. This trend is less clear for the single network models, as while the networks trained on filtered data showed some comparably stable predictions, such as the DIP in figure 5.7 or the PIP in 5.8, similar behaviours were also seen for the other models and can as such not definitively be attributed to the filtering of the past information.

It also seems that the filtered delta flex is beneficial when estimating the PIP. While most of the PIP predictions are quite comparable, the two with the least noise by some margin are the individually trained and the flex sensor separately trained networks when using filtered delta flex. The filtered delta flex appears to stabilize the predictions overall, with all three predictions being very stable, though the DIP was not quite as stable as the filtered delta pressure.

Ultimately, from the results of table 5.1, the filtered delta pressure is the most beneficial past information for the network as it yielded the lowest error for each network training structure which visual analysis of the predictions largely supports. However, as has also been elaborated upon, there does appear to be useful information in the flex and filtered flex signals. Therefore, while it was not explored in this project it may be beneficial to feed multiple delta signals. Then, perhaps it would be possible to gain the benefits of the pressure delta tracking and the noise filtering of the flex data.

6.1.1.5 Should networks be separated

Based on what can be gathered from table 5.1, the results are inconclusive. No separation method performed better for all delta. However, it does appear that three separate networks are not the correct path forward as they never achieved the lowest score.

Comparing the two types of separated networks and the single network, in particular how the separated networks show magnified peaks in the MCP or fail to track the decrease in the PIP, it would seem that these joints correlate with the other joints and training without knowledge of the other joints leads to the behaviour. Meanwhile, the DIP joint can seemingly be modelled quite well using a separated network, presumably due to it being the least dependent on the movements of the other two joints.

There is however also another consideration, which is stability. While the separated networks did result in undesirable tacking behaviours, the separation did seemingly allow the networks to better cope with the noise compared to the single networks. Thus while it does seem the models for the MCP and PIP require knowledge of the other angles to achieve acceptable tracking, the fact that the separated networks can stabilise the predictions in such a way raises a question regarding the required complexity of the network. During this work, comparisons between network separation methods were only done on two-layer networks. The ability of the separated networks to filter out noise could therefore be a result of having the same complexity of network for a simpler task, rather than a benefit of separated networks.

Below is a table summarizing the analysis of the results. The rows represent a particular separation method-delta combination and the columns summarize how well the network prediction exhibits that quality. The two types of characteristics considered are stability, simply how much the prediction fluctuates, and how well the prediction tracks the true angle. This includes capturing the features of the vision angles, peaks, valleys, and plateaus, as well as the magnitudes and offset of the prediction.

		stable MCP	stable PIP	stable DIP	tracking MCP	tracking PIP	tracking DIP
single network	Δ pressure	average	average	average	good	average	good
	Δ flex	good	average	poor	good	good	average
	filtered Δ pressure	average	average	good	average	average	good
	filtered Δ flex	poor	good	average	average	good	poor
separated network	Δ pressure	average	average	average	poor	poor	good
	Δ flex	good	good	poor	poor	poor	good
	filtered Δ pressure	average	good	good	poor	good	good
	filtered Δ flex	good	good	good	poor	average	average
flex sensor separated	Δ pressure	average	poor	average	poor	average	good
	Δ flex	good	poor	average	poor	average	average
	filtered Δ pressure	average	average	good	poor	good	good
	filtered Δ flex	good	good	good	poor	good	average

Table 6.1: Summary of analysis regarding the key characteristics of the networks. Columns show desirable behaviours and rows are grouped first by network separation and then by which past information that network used.

Ultimately, before further testing is done with a sufficiently deep and wide network to determine whether a single network is capable of separating the joints and achieving accurate predictions no conclusions can be made as to whether separation is preferable.

6.1.2 answering research questions

Firstly, in regards to whether it is possible to train a network for estimating the joint angles, the results are sadly inconclusive. While there certainly are trends in the predictions which show promise, none of the predictions are accurate enough to show a definitive yes. Far more data and varied tests are required. As such, the question of which sensor signals are adequate can also not be conclusively answered. It can be concluded that the air pressure is not sufficient and that the predictions improve with the combination of both pressure and flex sensor data. The results also indicate that the inclusion of additional inputs decreased the error. The failures to track at certain points could also indicate that additional sensor inputs are required. Finally, in regards to the effect of different sources of input data, refer to section 6.1.1. To briefly summarize, flex sensor data can be used for an overall shape, with the addition of pressure data and the delta affecting the shape and noise level of the predictions.

6.2 Sources of error and the impact of design choices

6.2.1 Data gathering method

The method by which training data was acquired undoubtedly impacted the resulting model. By placing the actuator onto a horizontal plane and filming from above rather than suspending the actuator in mid-air by anchoring its base, as several previous works had done, the friction between the actuator and the surface altered the behaviour of the actuator. Initially when designing the recording rig it was deemed a more time-efficient and simple setup. The impact of friction compared to gravity was not seen as a major complication. There were also concerns about whether the failing actuator could withstand the heightened pressures required to overcome gravity and fully flex, as it had deteriorated significantly by the time the training data was gathered. While the author can not validate whether this was the correct assumption, it can nonetheless be said that the introduction of friction greatly complicated the modelling, in particular regarding the steady state of the system. As can be seen in several datasets, such as figure 4.1, the resting state of the actuator (i.e. the angles of the actuator when not inflated) differs depending on in what manner and to what extent the actuator was previously inflated. Reasonably meanwhile, gravity would have provided a consistent result regardless of movement, as it is merely a force constantly affecting the actuator rather than a force counteracting any movement the actuator attempts. This is supported by [13] whose tracking of a suspended actuator shows the actuator returns to the original angle after the extension was finished.

The impact of friction being introduced is not something the author can determine. However, the lack of a single resting state and the fact the actuator behaviour depends on the type of motions performed undoubtedly complicated the attempts to create a model which has to comprehend these behaviours. It also adds complications, since to model the actuator behavior the model must also comprehend and characterize the outside environment that causes the friction. Another unforeseen effect of the friction is in regards to the "delay" in the predictions previously discussed. Because there is currently no method for eliminating friction in the current system it is impossible to determine if the delay is related to the friction or some other factors.

Another factor which may have impacted the results was the lack of training data beginning at different resting states. Because the friction prevents the actuator from returning to the original resting state after a ramp movement, the next ramp will begin at a different initial position. This could perhaps have been lessened by also training the networks on data from some of those resting states. Similarly, all datasets used for training showed full movements of all three joints. Without data on how the joints move individually when not subject to the other joints, by for instance restricting the others from flexing, it could potentially be that the network does not understand how the joints function independently as it lacked such data.

6.2.2 the use of RMSE as a measure of quality

In this work, the only objective scoring method used was the averaged RMSE of the three joints. As previously mentioned this was based on the fact that previous works had utilised RMSE as the quality measure of models. However, during the testing process, it soon became clear that other aspects of the predictions were being ignored. While overall, the current measure will favour a stable and accurate prediction, in the case of this project, where not any of the predictions can be considered satisfying, it fails to provide a measure of "quality". As the reader may have seen observing the predictions in the results chapter 5.1 and as discussed above, most all figures show some behaviours which are desirable behaviour. A more comprehensive measure of quality meanwhile could have offered a quantitative measurement of these various behaviours, rather than relying on subjective assessment.

6.2.3 3/2 solenoid valve

In this project, a 3/2 solenoid valve controlled via a modified PWM signal was used for controlling the pressure. It was chosen based on being easy to control, relatively cheap and only requiring one valve per actuator. However, the rapid valve switching has caused several issues for the project. The main impact it has had on the results is the unfortunate combination of pressure oscillations due to the rapid valve switching and the forced sample rate of 8.3 ms leading to a random spread of recorded pressure values. This likely caused further complications during the training process, as the model had to account for the variation of ca. $\pm$ 2.5 kPa for a given pressure reading. Had a proportional valve been used instead these oscillations would not exist and created more stable pressure data.

6.3 Limitations

The findings of this report have clear limitations. Firstly, all data used for the neural network was acquired from a single initial state. As it can not be conclusively determined what causes the discrepancies in the prediction, it can not be refuted that this limitation may have impacted the results. While gathering data from a single initial state did allow for higher repeatability and could be a reason for the predictions being accurate during the initial rise, data from different initial states may have offered the network the ability to learn other behaviours better. However, doing so would require a more advanced rig to ensure a consistent initial state.

Secondly, the impact a human user would have on the system's behaviour was not considered. As such, there are several factors which the models developed in this work do not account for. This does raise further questions regarding the feasibility of using neural network modelling as it is unknown whether the network could be trained to account for these disturbances, however, it was seen as a necessary limitation due to time restrictions and the heightened complexity of such a model. Future studies however will need to determine what effect the actuator being mounted onto a user has on the system behavior.

Meanwhile, the impact of limiting the project to only flexion and extension as well as only considering one finger is believed to have had none to negligible impact on the project results. While multiple actuators could affect the airflow, as the valves are all connected to the same inlet and outlet, the impact of utilising multiple valves is expected to be negligible as the pressure difference will always be greater between the outlet and the exhaust compared to two outlets. As for adduction and abduction, it was not considered relevant as it would require an entirely different actuator and sensor mounting.

6.4 Future work

In this section, suggestions for continued work on this subject and system will be outlined. This includes parts of the prototype design which could be revised and subjects or tests which the author believes could be beneficial for further developing this system.

Firstly, it is recommended to investigate other methods of manufacturing the soft actuator. While the segmented PneuNet actuator remains promising, the repeated failures and leakages due to the weakness of the seam between the two parts must be resolved as such leakages caused far too many interruptions. Designs such as Wag et. al. [7] used a balloon to seal the air, which could be worth considering. Especially as a future step of testing will likely involve the actuators being mounted on a human hand, which requires higher pressures for full flexion. Ruptures are therefore more likely. Also, a larger study on the wear over time of the actuators should be undertaken. During this work, in addition to weakening seams, there was a noticeable difference in the actuator's pressure-flexion behavior over time which will further complicate characterizing the actuator's behavior.

Secondly, it is recommended to replace the existing solenoid valves with either another model of 3/2 solenoid valve or abandon the 3/2 valves in favour of proportional valves. While the pressure controller created for this system is functional it is unconventional compared to ordinary PWM. Replacing them with valves which lack the automatic closing mechanism would make the control more universal and should work with most 3/2 valves, as well as allowing for more refined timing. It should also allow for smoother pressure control, as the current control solution relies on setting the on time by increments which results in noticeable steps in pressure and limited controller accuracy. The current valves are also too loud to be practical for rehabilitative use and if the sound level measurements taken were accurate, long-term exposure would be harmful due to the risk of damaging the patient's hearing. There is also the issue regarding the oscillations in pressure. As discussed earlier in this chapter the rapid valve switching causes the pressure to fluctuate which complicates the training. As Elgeneidy et. al. [13] did not discuss such behaviour however it may be a unique issue of this system. Therefore a different valve should be tested so that hopefully this issue can be resolved. Proportional valves are especially tempting, provided they can be afforded. Firstly, they would offer actual control over the flow rate in the system, allowing for both the system to behave slower but also increasing control accuracy by letting the system be controlled via traditional PWM. Both of these would be beneficial for control as well as patient safety reasons. Secondly, they would resolve the issue of the valve's dead time. As mentioned during the development of the control 3.2.2, the valve requires an on-time of at least 8.5 ms to open. As a result, when beginning actuation the control signal can never be less than 8.5, though in practice 9 ms was found to be the more stable. As a result, there is currently no way to gently increase the pressure from 0, it will always jolt up to the minimal control signal, which in ca. 20 kPa. A proportional valve does not suffer from dead zones and would therefore allow for a smoother initial pressurization. This could prove especially valuable in the case of patients with a very limited range of motion where even the initial jolt of the actuator could be too forceful. Thirdly, it would also remove the oscillations in pressure which may be a contributing factor to the network challenges.

Thirdly, an alternate method for determining the actuator's direction of travel could be the addition of an airflow sensor. This was attempted early on in this work but failed to materialize. The flow sensor could be particularly interesting to investigate should the proportional solenoid valve be implemented, as the sensor would allow additional feedback and characterization. It could also potentially offer a less noisy solution than the delta information used in this work.

In regards to the modelling of the actuator, there is much work to be done. Firstly, a testing rig which allows for the actuator to be suspended should be built and the tests conducted in this project redone to evaluate the impact of friction versus gravity as well as validate the conclusions drawn. As mentioned above, a more detailed list of quantitative quality measures should also be developed besides the RMSE to be applied to these tests, so that the selection process going forward is more quantitative.

Finally, a more comprehensive analysis of deeper and wider neural networks should be conducted. For this work, it was found that two layers with ten neurons yielded the best result, but due to the sheer number of parameters far from all permutations of layers, neurons, activation functions and learning parameters were tested. As described in chapter 2.4, a deeper network would potentially allow the network to create more complex characterizations which may better represent the actuator, especially for more complex use cases and in particular once the human hand is introduced in the future, further complicating matters by adding further unpredictable resistance. Tests using different network depths and widths should also be done to evaluate the theory that the ability of the separately trained network to filter out noise was in part due to having the same number of neurons to allocate to a single joint.

7
Conclusion

The aim of this project was to create a regression model for the joint angles of a segmented soft pneumatic actuator which could later be used for a closed-loop controller of a soft pneumatic glove.

During the course of the project, a prototype pneumatic soft actuator system was developed. It featured a novel segmented PneuNet actuator which aimed to replicate the bending of a human finger by segmenting it into joints and two flex sensors integrated into the bottom layer. Data from the integrated flex sensors as well as air pressure was gathered during various actuator motions. These, in combination with joint angles extracted from recordings of the actuator's movements using computer vision, were then used for training neural networks for regression to model the actuator joint angles.

Different network structures and input data features were tested to evaluate what impact they had on the network predictions. The predictions did show that in the future neural networks may be used for training a model capable of accurately predicting the joint angles. While no clear conclusion could be drawn as to what the optimal structure and training data are, some observations could be made based on trends in the predictions. It was concluded that the MCP and PIP require knowledge of the other joints to produce good predictions while the DIP is relatively independent. It was also concluded that feeding the networks past information resulted in better predictions, however, different features affected the joints to varying degrees resulting in different desirable trends. It could generally be seen that filtering the past information resulted in better predictions overall. Filtered past information of the pressure resulted in better averaged RMSE scores, while filtered past information of the flex sensors resulted in less noisy predictions. Similarly, no conclusion could be drawn as to whether the prediction benefits from training networks separately on different joints. The single network possessed better tracking with no large discrepancies while the separately trained networks showed an inability to track certain movements accurately. The separately trained networks however did result in stable predictions with little noise when trained on filtered past information.

Finally, a set of future development suggestions was presented for consideration, including a deeper study into the effects of deeper and wider networks as well as potential improvements to the system and actuator.

Bibliography

[1] M. S. Xavier, C. D. Tawk, A. Zolfagharian, *et al.*, *Soft pneumatic actuators: A review of design, fabrication, modeling, sensing, control and applications*, 2022. DOI: 10.1109/ACCESS.2022.3179589.

[2] P. Polygerinos, Z. Wang, K. C. Galloway, R. J. Wood, and C. J. Walsh, "Soft robotic glove for combined assistance and at-home rehabilitation," vol. 73, Elsevier B.V., Nov. 2015, pp. 135–143. DOI: 10.1016/j.robot.2014.08.014.

[3] P. Polygerinos, S. Lyne, Z. Wang, *et al.*, "Towards a soft pneumatic glove for hand rehabilitation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1512–1517. DOI: 10.1109/IROS.2013.6696549.

[4] Z. Duanmu, M. Stommel, L. K. Cheng, and W. Xu, "Simulation of solid meal digestion in a soft gastric robot using sofa," in *2021 27th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, 2021, pp. 357–362. DOI: 10.1109/M2VIP49856.2021.9664988.

[5] E. Roche, M. Horvath, I. Wamala, *et al.*, "Soft robotic sleeve supports heart function," *Science Translational Medicine*, vol. 9, eaaf3925, Jan. 2017. DOI: 10.1126/scitranslmed.aaf3925.

[6] H. K. Yap, J. H. Lim, F. Nasrallah, J. C. Goh, and R. C. Yeow, "A soft exoskeleton for hand assistive and rehabilitation application using pneumatic actuators with variable stiffness," vol. 2015-June, Institute of Electrical and Electronics Engineers Inc., Jun. 2015, pp. 4967–4972, ISBN: 9781479969234. DOI: 10.1109/ICRA.2015.7139889.

[7] J. Wang, Y. Fei, and W. Pang, "Design, modeling, and testing of a soft pneumatic glove with segmented pneunets bending actuators," *IEEE/ASME Transactions on Mechatronics*, vol. 24, pp. 990–1001, 3 Jun. 2019, ISSN: 10834435. DOI: 10.1109/TMECH.2019.2911992.

[8] J. Nassour, F. H. Hamker, and G. Cheng, "High-performance perpendicularly-enfolded-textile actuators for soft wearable robots: Design and realization," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 3, pp. 309–319, 2020. DOI: 10.1109/TMRB.2020.3012131.

[9] H. K. Yap, P. M. Khin, T. H. Koh, *et al.*, "A fully fabric-based bidirectional soft robotic glove for assistance and rehabilitation of hand impaired patients," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1383–1390, 2017. DOI: 10.1109/LRA.2017.2669366.

[10] S. Kokubu, P. E. T. Vinocour, and W. Yu, "Development and evaluation of fiber reinforced modular soft actuators and an individualized soft rehabilitation glove," *Robotics and Autonomous Systems*, vol. 171, p. 104 571, 2024,

ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2023.104571. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889023002105.

[11] S. Kokubu, P. E. T. Vinocour, and W. Yu, "Development and evaluation of fiber reinforced modular soft actuators and an individualized soft rehabilitation glove," *Robotics and Autonomous Systems*, vol. 171, p. 104 571, 2024, ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2023.104571. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889023002105.

[12] H. Al-Fahaam, S. Davis, and S. Nefti-Meziani, "Power assistive and rehabilitation wearable robot based on pneumatic soft actuators," Institute of Electrical and Electronics Engineers Inc., Sep. 2016, pp. 472–477, ISBN: 9781509018666. DOI: 10.1109/MMAR.2016.7575181.

[13] K. Elgeneidy, N. Lohse, and M. Jackson, "Bending angle prediction and control of soft pneumatic actuators with embedded flex sensors – a data-driven approach," *Mechatronics*, vol. 50, pp. 234–247, Apr. 2018, ISSN: 09574158. DOI: 10.1016/j.mechatronics.2017.10.005.

[14] emedicinehealth, *Illustration picture of hand strucutres - finger anatomy*, https://www.emedicinehealth.com/image-gallery/finger_anatomy_picture/images.htm, Accessed: 2024-07-24.

[15] M. R. Villarreal, *Ladyofhats - drawing bones of the hand - english labels*, https://anatomytool.org/content/ladyofhats-drawing-bones-hand-english-labels, Acessed: 2024-08-27.

[16] G. Sharp and D. Thompson, *Biomechanics of the hand*, https://ouhsc.edu/bserdac/dthompso/web/namics/hand.htm, Accessed: 2024-07-24.

[17] T. M. Tennakoon, A. L. Kulasekera, D. S. Chathuranga, and R. A. Gopura, "Effect of geometrical parameters on pneunet bending performance," Institute of Electrical and Electronics Engineers Inc., 2021, pp. 738–743, ISBN: 9781665436847. DOI: 10.1109/ICAR53236.2021.9659425.

[18] M. Memarian, R. Gorbet, and D. Kulic, "Control of soft pneumatic finger-like actuators for affective motion generation," vol. 2015-December, Institute of Electrical and Electronics Engineers Inc., Dec. 2015, pp. 1691–1697, ISBN: 9781479999941. DOI: 10.1109/IROS.2015.7353595.

[19] S. J. SHAW and M. A. MORRIS, "The range of motion of the metacarpophalangeal joint of the thumb and its relationship to injury," *Journal of Hand Surgery*, vol. 17, no. 2, pp. 164–166, 1992. DOI: 10.1016/0266-7681(92)90080-L. eprint: https://doi.org/10.1016/0266-7681(92)90080-L. [Online]. Available: https://doi.org/10.1016/0266-7681(92)90080-L.

[20] H. Huang, J. Lin, L. Wu, B. Fang, and F. Sun, "Optimal control scheme for pneumatic soft actuator under comparison of proportional and pwm-solenoid valves," *Photonic Network Communications*, vol. 37, pp. 153–163, 2 Apr. 2019, ISSN: 15728188. DOI: 10.1007/s11107-018-0815-3.

[21] J.-W. Pustjens, Accessed 2024-07-26. [Online]. Available: https://tameson.com/pages/solenoid-valve-types.

[22] K. Ahn and S. Yokota, "Intelligent switching control of pneumatic actuator using on/off solenoid valves," *Mechatronics*, vol. 15, no. 6, pp. 683–702, 2005,

ISSN: 0957-4158. DOI: https://doi.org/10.1016/j.mechatronics.2005.01.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957415805000292.

[23] C. Kolstad, Accessed 2024-07-26. [Online]. Available: https://tameson.com/pages/proportional-solenoid-control-valve.

[24] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *3. linear neural networks for regression*, https://www.d2l.ai/chapter$_l$*inear − regression/index.html*, Accessed: 2024-07-24.

[25] I. N. da Silva, D. H. Spatti, R. A. Flauzino, L. H. B. Liboni, and S. F. dos Reis Alves, *Artificial Neural Networks, A Practical Course*. Springer eBooks, 2017, ch. 1 and 2, pp. 3–28. DOI: 10.1007/978-3-319-43162-8.

[26] T. Nguyen, M. Raghu, and S. Kornblith, "Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=KJNcAkY8tY4.

[27] *Cyberglove ii*, Acessed 2024-07-30. [Online]. Available: https://www.cyberglovesystems.com/cyberglove-ii.

[28] M. A. Mohandes, "Recognition of two-handed arabic signs using the cyberglove," *Arab J Sci Eng 38, 669–677*, 2013. DOI: https://doi.org/10.1007/s13369-012-0378-z.

[29] S. Andersson, F. Gråhn, A. Heijkenskjöld, Z. Lind, S. Lunman, and A. Strandrot Frid, *Pneumatic soft robotic glove for hand rehabilitation*, B.S. thesis Chalmers University of Technology, 2023.

[30] Y. Han, Q. Xu, and F. Wu, "Design of wearable hand rehabilitation glove with bionic fiber-reinforced actuator," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 10, pp. 1–10, 2022. DOI: 10.1109/JTEHM.2022.3196491.

[31] Z. Wang, P. Polygerinos, A. Alazmani, and A. Campo. "Pneunets design." [Accessed: 26 July 2024]. (), [Online]. Available: https://softroboticstoolkit.com/book/pneunets-design.

A

Tables of Neural network testing results

	relu	leaky relu	elu	swish
relu	2.7623	2.7225	2.7044	2.8216
lekay relu	2.7943	3.5379	2.7610	2.9683
elu	3.0170	2.8889	2.9691	2.6907
swish	2.8514	3.0137	2.8822	2.6700
sigmoid	2.8104	2.9783	2.8991	2.8845
tanh	2.7331	3.0932	2.9356	3.3510

Table A.1: Table of averaged RMSE for combined network without delta

	relu	leaky relu	elu	swish
relu	2.5476	3.1052	2.6289	2.8407
lekay relu	2.6894	2.7722	2.9852	2.5873
elu	2.7074	2.6236	2.6478	2.8550
swish	2.8289	2.7198	2.8259	2.7689
sigmoid	2.7980	2.7386	2.7302	2.8379
tanh	2.7745	2.8050	2.7413	2.7060

Table A.2: Table of averaged RMSE for combined network with Δpressure

	relu	leaky relu	elu	swish
relu	2.8190	2.7784	2.8540	2.8159
lekay relu	2.7512	2.8780	2.9774	2.9200
elu	2.9953	2.7955	2.9925	2.7436
swish	2.9606	3.0571	2.7289	2.8402
sigmoid	2.7932	2.8352	3.1952	2.6861
tanh	2.9851	2.8271	2.9152	2.9325

Table A.3: Table of averaged RMSE for combined network with Δflex

	relu	leaky relu	elu	swish
relu	2.8287	2.5582	2.6330	2.8148
lekay relu	2.5933	2.7534	2.4684	2.8487
elu	3.3156	2.7882	2.6603	2.7224
swish	2.6840	2.7037	2.5998	2.4136
sigmoid	2.6868	2.8820	2.7784	2.7013
tanh	3.1875	2.6447	2.6702	3.0096

Table A.4: Table of averaged RMSE for combined network with filtered Δpressure

	relu	leaky relu	elu	swish
relu	2.6563	2.6586	2.6586	2.8493
lekay relu	2.7493	2.6584	3.0972	2.8578
elu	2.6638	2.6891	2.7295	2.6310
swish	2.8818	3.0665	2.8622	2.5665
sigmoid	2.7139	2.7289	2.6997	3.0453
tanh	3.1248	2.7625	2.8908	2.7510

Table A.5: Table of averaged RMSE for combined network with filtered Δflex

	relu	leaky relu	elu	swish
relu	2.4172	3.0809	2.4440	3.0932
lekay relu	2.6138	2.9315	2.9421	3.2145
elu	3.3721	2.6678	2.9901	3.5372
swish	3.1662	3.3516	3.1205	2.9779
sigmoid	2.5297	2.4141	2.5521	2.5807
tanh	3.2863	2.6869	2.5763	2.9995

Table A.6: Table of RMSE for individually trained MCP networks without delta

	relu	leaky relu	elu	swish
relu	2.5365	2.5071	2.9826	2.3660
lekay relu	2.4456	2.5431	2.9441	2.5875
elu	2.2623	2.6991	2.6175	2.5544
swish	2.7213	2.9603	2.5313	2.5329
sigmoid	2.5719	2.5622	2.7110	2.5419
tanh	2.6195	2.6494	2.3155	2.4824

Table A.7: Table of RMSE for individually trained PIP networks without delta

	relu	leaky relu	elu	swish
relu	3.3690	3.6315	3.4153	3.2686
lekay relu	3.4313	3.4908	3.4311	3.4308
elu	3.4281	3.4138	3.4541	3.4552
swish	3.3783	3.5260	3.4036	3.5225
sigmoid	3.5803	3.6850	3.6472	3.5901
tanh	3.3856	3.5001	3.6091	3.4820

Table A.8: Table of RMSE for individually trained DIP networks without delta

	relu	leaky relu	elu	swish
relu	2.7739	2.6281	2.4269	2.4221
lekay relu	3.2187	2.4894	2.5193	2.4944
elu	2.9863	3.3064	2.8459	2.9065
swish	2.8558	2.8030	2.9611	2.9301
sigmoid	2.4289	2.3628	2.4741	2.4289
tanh	2.6307	2.8017	2.7355	2.5550

Table A.9: Table of RMSE for individually trained MCP networks with ΔPressure

	relu	leaky relu	elu	swish
relu	2.2412	2.5496	2.5382	2.3356
lekay relu	2.5045	2.4339	2.1917	2.1420
elu	2.2733	2.3358	2.2473	2.2816
swish	2.5355	2.2300	2.5380	2.2426
sigmoid	2.5319	2.6787	2.6701	2.3376
tanh	2.5205	2.1625	2.5426	2.9553

Table A.10: Table of RMSE for individually trained PIP networks with ΔPressure

	relu	leaky relu	elu	swish
relu	3.1851	3.1898	3.1907	3.1818
lekay relu	3.3235	3.1966	3.3041	3.1681
elu	3.1949	3.3524	3.1644	3.2171
swish	3.1591	3.1943	3.2351	3.1942
sigmoid	3.6290	3.2648	3.5878	3.4384
tanh	3.2618	3.2875	3.2183	3.2465

Table A.11: Table of RMSE for individually trained DIP networks with Δpressure

	relu	leaky relu	elu	swish
relu	2.8771	2.6942	2.4365	2.5768
lekay relu	2.8531	2.5216	2.3463	2.4753
elu	2.7449	2.5890	2.8630	2.8328
swish	3.1504	2.9764	3.2064	2.8719
sigmoid	2.5590	2.4612	2.4910	2.3795
tanh	2.8468	2.9376	3.2084	2.7161

Table A.12: Table of RMSE for individually trained MCP networks with Δflex

	relu	leaky relu	elu	swish
relu	2.5248	2.4927	2.4468	2.4619
lekay relu	2.4286	2.4223	2.5942	2.5010
elu	2.6841	2.5821	2.6549	2.7915
swish	2.4190	2.5293	2.4023	2.6088
sigmoid	2.4912	2.6802	2.5133	2.4080
tanh	2.4249	2.5634	2.5565	2.5212

Table A.13: Table of RMSE for individually trained PIP networks with Δflex

	relu	leaky relu	elu	swish
relu	3.9694	3.4926	3.6246	3.4668
lekay relu	3.3682	3.5575	3.5225	3.5132
elu	3.4801	3.4688	3.4278	3.4597
swish	3.4793	3.6795	3.4709	3.6371
sigmoid	3.7356	3.7141	3.7570	3.4261
tanh	3.4642	3.5996	3.5387	3.5697

Table A.14: Table of RMSE for individually trained DIP networks with Δflex

	relu	leaky relu	elu	swish
relu	2.2760	3.1061	2.6810	2.4507
lekay relu	2.7548	2.9297	2.2453	3.3151
elu	2.9140	3.3236	2.4507	3.0156
swish	2.4810	2.6715	3.0011	2.9924
sigmoid	2.3921	2.4315	2.4992	2.3949
tanh	2.4734	2.7655	3.0718	2.6961

Table A.15: Table of RMSE for individually trained MCP networks with filtered Δ pressure

	relu	leaky relu	elu	swish
relu	2.7005	2.2488	2.5309	2.1686
lekay relu	2.3525	2.1964	2.3262	2.6676
elu	2.3520	2.1602	2.1272	2.0392
swish	2.3170	2.1697	3.0671	2.1684
sigmoid	2.4477	2.4046	2.2415	2.2974
tanh	2.3792	2.3950	2.2892	2.5162

Table A.16: Table of RMSE for individually trained PIP networks with filtered Δ pressure

	relu	leaky relu	elu	swish
relu	3.4309	3.2116	3.3342	3.2072
lekay relu	3.2583	3.2416	3.2956	3.2687
elu	3.2691	3.3353	3.2686	3.2282
swish	3.3142	3.3202	3.3013	3.3121
sigmoid	3.5449	3.5882	3.6129	3.4595
tanh	3.6162	3.4425	3.3220	3.3787

Table A.17: Table of RMSE for individually trained DIP networks with filtered Δ pressure

	relu	leaky relu	elu	swish
relu	2.4312	2.9130	2.1949	2.3313
lekay relu	2.3959	2.4347	2.7515	2.5503
elu	2.6058	3.1114	2.5771	2.5779
swish	2.8631	2.4817	2.6796	2.2564
sigmoid	2.2632	2.2705	2.3624	2.2597
tanh	2.5273	3.5161	2.7867	2.5001

Table A.18: Table of RMSE for individually trained MCP networks with filtered Δ flex

	relu	leaky relu	elu	swish
relu	2.3111	2.3714	2.5520	2.5710
lekay relu	2.3570	2.7020	2.3805	2.4275
elu	2.8062	2.1105	2.4483	2.2186
swish	2.4818	2.4105	2.3012	2.5295
sigmoid	2.1862	2.2804	2.3241	2.4158
tanh	2.3368	2.2628	2.4965	2.4659

Table A.19: Table of RMSE for individually trained PIP networks with filtered Δ flex

	relu	leaky relu	elu	swish
relu	3.5334	3.4574	3.4325	3.6356
lekay relu	3.6049	3.5162	3.5688	3.6768
elu	3.4293	3.3969	3.5087	3.3874
swish	3.3926	3.6287	3.5804	3.4417
sigmoid	3.5819	3.6279	3.8622	3.6799
tanh	3.6716	3.9103	3.5010	3.5606

Table A.20: Table of RMSE for individually trained DIP networks with filtered Δ flex

	relu	leaky relu	elu	swish
relu	2.9949	2.9103	2.7714	3.2150
lekay relu	2.9162	2.9076	2.8543	2.8924
elu	2.8797	2.8560	2.8573	2.8670
swish	2.8127	2.8742	9.6267 *	2.9421
sigmoid	3.4371	3.4041	3.1141	3.1148
tanh	3.3084	2.9260	2.9562	2.8816

Table A.21: Table of RMSE for combined PIP+DIP networks without delta

	relu	leaky relu	elu	swish
relu	2.6854	2.8249	2.5876	2.7551
lekay relu	2.7114	2.6555	2.6897	2.7994
elu	2.9075	2.8741	2.6139	2.7038
swish	2.6783	2.6930	2.6986	2.9008
sigmoid	3.2612	3.1134	3.0299	2.7935
tanh	3.5922	3.1724	3.0443	2.8239

Table A.22: Table of RMSE for combined PIP+DIP networks with Δ pressure

	relu	leaky relu	elu	swish
relu	2.8555	2.9636	2.9069	2.9536
lekay relu	2.9268	2.9308	2.9228	2.8573
elu	2.8017	2.8554	2.9595	2.8265
swish	2.8988	3.0896	2.9299	2.9789
sigmoid	3.0979	2.9997	2.9085	3.0884
tanh	3.1762	2.9957	3.0752	2.9714

Table A.23: Table of RMSE for combined PIP+DIP networks with Δ flex

	relu	leaky relu	elu	swish
relu	2.7646	2.8828	2.8864	2.8424
lekay relu	2.6840	2.8564	2.6642	2.7677
elu	2.7410	2.7631	2.6237	2.8014
swish	2.7849	2.7005	2.6540	2.9164
sigmoid	3.0009	2.9362	3.0729	3.0197
tanh	2.7812	2.7452	2.7305	2.7453

Table A.24: Table of RMSE for combined PIP+DIP networks with filtered Δ pressure

	relu	leaky relu	elu	swish
relu	2.8121	3.2456	2.8450	2.7182
lekay relu	2.9143	2.9602	2.8617	2.7905
elu	2.9223	3.0366	2.9571	2.7985
swish	2.8215	2.8216	2.8330	3.1954
sigmoid	2.8881	2.8288	3.0646	3.0187
tanh	2.9764	3.0533	2.8367	2.9803

Table A.25: Table of RMSE for combined PIP+DIP networks with filtered Δ flex

B

Complete list of datasets

Video recordings are also available at https://www.youtube.com/playlist?list=PL2-X-IUuVJh0aJB6ppWigDhvhaM4RHkWy

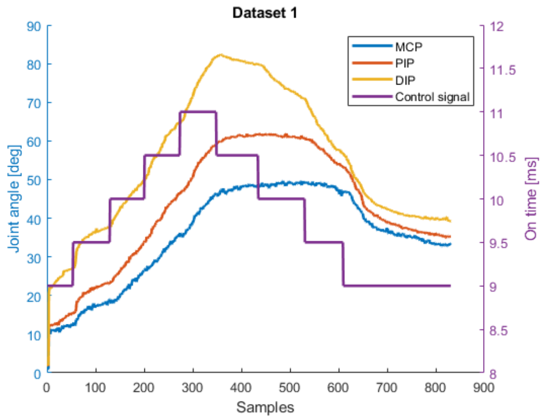


Figure B.1: *Plot showing dataset 1, a ramp movement which incremented at a rate of 1 second. Unit for the angles are in degrees (left axis) and milliseconds (right axis)*

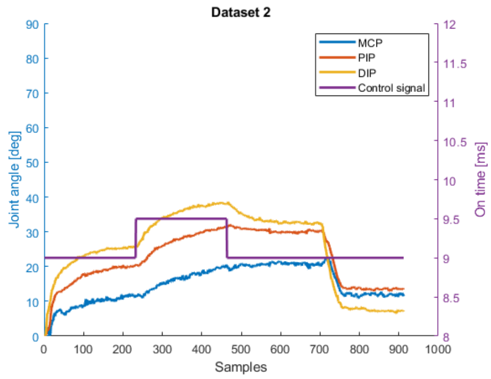


Figure B.2: *Plot showing dataset 2, a step movement which stepped from 9 to 9.5. Unit for the angles are in degrees (left axis) and milliseconds (right axis)*

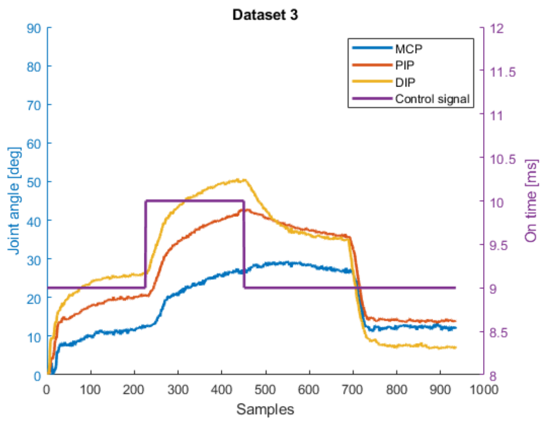


Figure B.3: *Plot showing dataset 3, a step movement which stepped from 9 to 10. Unit for the angles are in degrees (left axis) and milliseconds (right axis)*

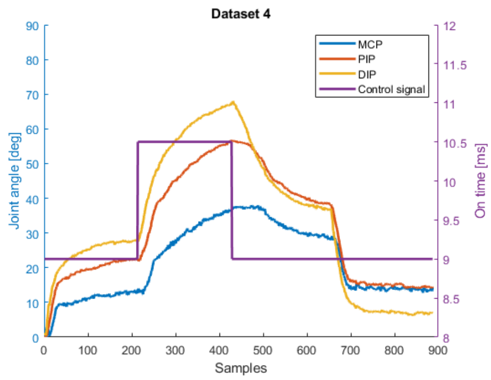


Figure B.4: *Plot showing dataset 4, a step movement which stepped from 9 to 10.5. Unit for the angles are in degrees (left axis) and milliseconds (right axis)*

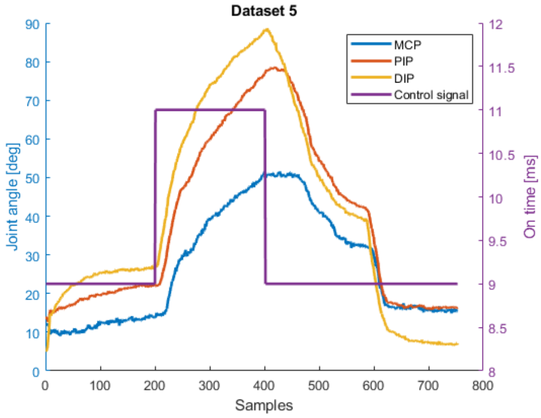


Figure B.5: *Plot showing dataset 5, a step movement, which stepped from 9 to 11. Unit for the angles are in degrees (left axis) and milliseconds (right axis)*

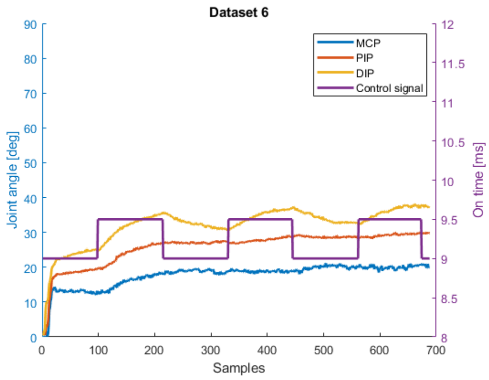


Figure B.6: *Plot showing dataset 6, a cycling ramp movement which incremented from 9 to 9.5. Unit for the angles are in degrees (left axis) and milliseconds (right axis)*

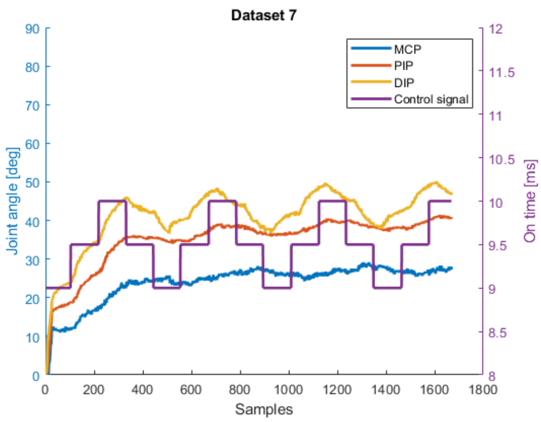


Figure B.7: *Plot showing dataset 7, a cycling ramp movement which incremented from 9 to 10.Unit for the angles are in degrees (left axis) and milliseconds (right axis)*

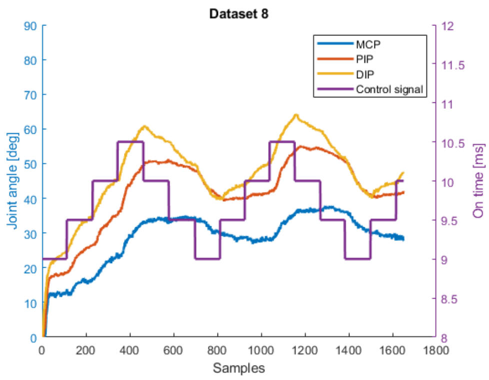


Figure B.8: *Plot showing dataset 8, a cycling ramp movement movement which stepped from 9 to 10.5. Unit for the angles are in degrees (left axis) and milliseconds (right axis)*

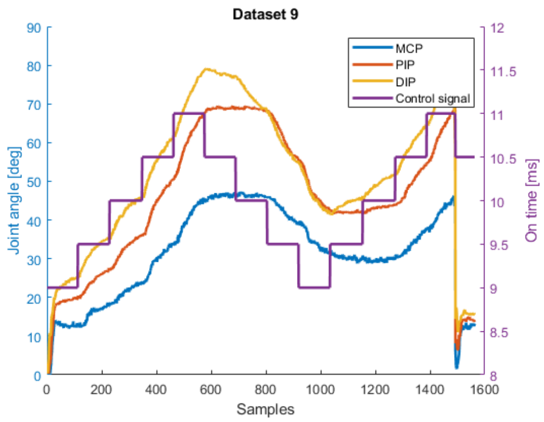


Figure B.9: *Plot showing dataset 9, a cycling ramp movement movement which stepped from 9 to 11. Unit for the angles are in degrees (left axis) and milliseconds (right axis). Note for this one that the end shows a rupture, which whose data was removed during testing.*

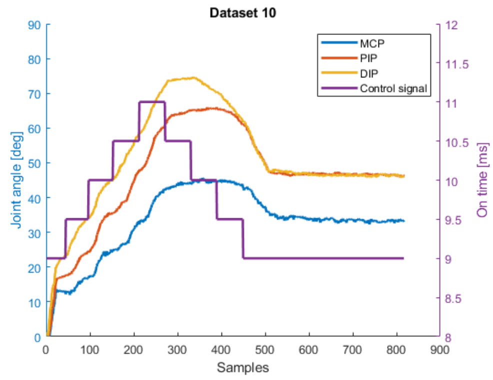


Figure B.10: *Plot showing dataset 10, a ramp movement which incremented at a rate of 0.5 seconds. Unit for the angles are in degrees (left axis) and milliseconds (right axis)*

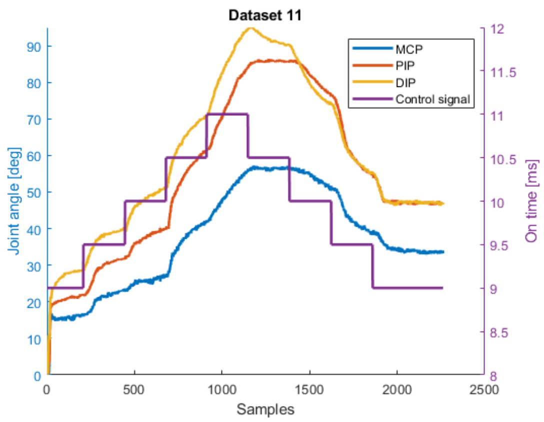
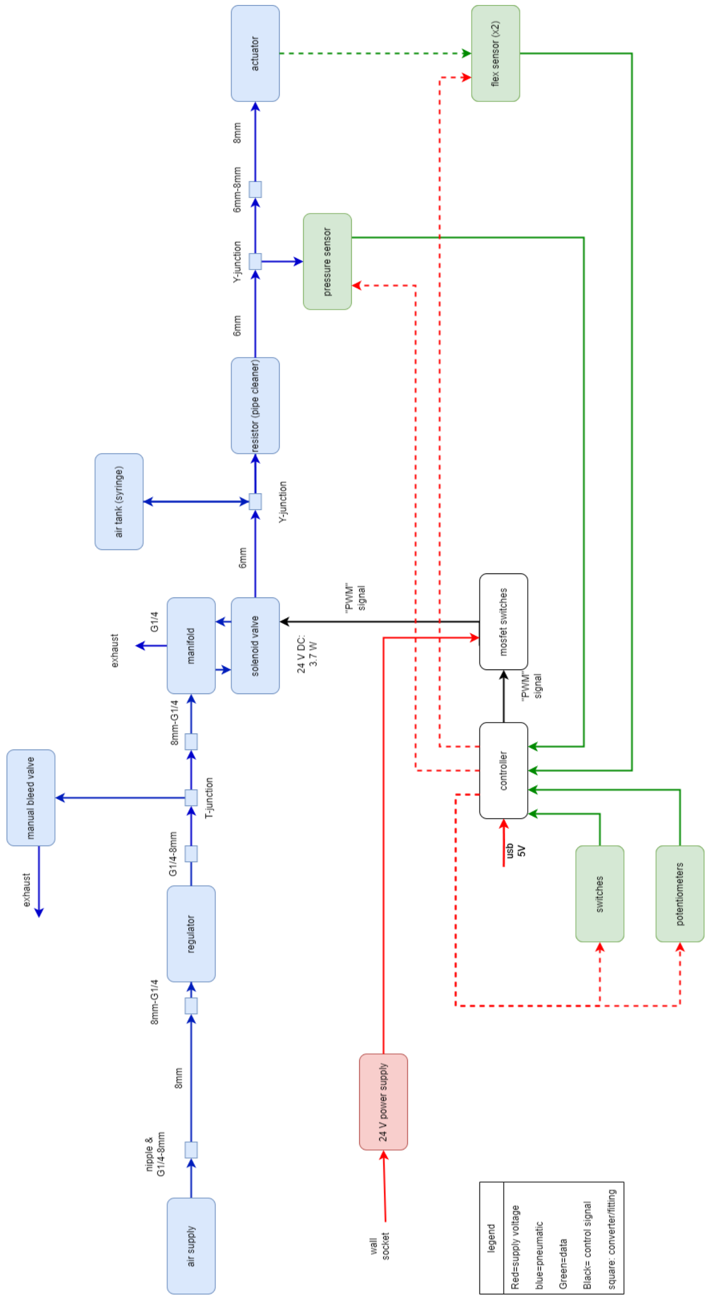


Figure B.11: *Plot showing dataset 11, a ramp movement which incremented at a rate of 2 seconds. Unit for the angles are in degrees (left axis) and milliseconds (right axis)*

C

System diagram



DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY