Recurrent Neural Network Dual Resistance Control of Multiple Memory Shape Memory Alloys

by

Igor Ruvinov

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2018

© Igor Ruvinov 2018

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Shape memory alloys (SMAs) are materials with extraordinary thermomechanical properties which have caused numerous engineering advances. NiTi SMAs in particular have been studied for decades revealing many useful characteristics relative to other SMA compositions. Their application has correspondingly been widespread, seeing use in the robotics, automotive, and aerospace industries, among others. Nevertheless, several limitations inherent to SMAs exist which inhibit their applicability, including their inherent single transformation temperature and their complex hysteretic actuation behaviour.

To overcome the former challenge, one method utilizes high energy laser processing to perform localized vaporization of nickel and accurately adjust its transformation temperatures. This method can reliably produce NiTi SMAs with multiple monolithic transformation memories. There have also been attempts to overcome the latter of the aforementioned challenges by designing systems which model NiTi's hysteretic behaviour. When applied to actuators with a single transformation memory, these methods require the use of external sensors for modeling actuators with varying current and load, driving up the cost, weight, and complexity of the actuator. Embedding a second transformation memory with different phase into NiTi actuators can overcome this issue. By measuring electrical resistance across the two phases, sufficient information can be extracted for differentiating events caused by heating from those caused by applied load. The current study examines NiTi wires with two embedded transformation memories and utilizes recurrent neural networks for interpreting the sensed data. The knowledge gained through this study was used to create a recurrent neural network-based model which can accurately estimate the position and force applied to the NiTi actuator without the use of external sensors.

The first part of the research focused on obtaining a comprehensive thermomechanical characterization of laser processed and thermomechanically post-processed NiTi wires with two embedded transformation memories, with one memory exhibiting full SME and the second partial PE at room temperature. A second objective of this section was to acquire cycling data from the processed wires which would be used for training the artificial neural networks in the following section of the study. The selected laser processing and post-processing parameters resulted in a transformation temperature increase of $61.5°C$ and $35.3°C$ for A_f and M_s, respectively, relative to base metal. Furthermore, the post-processing was found to successfully restore the majority of the lost mechanical properties, with the ultimate tensile strength recovered to 84% of its corresponding base metal value. This research resulted in the fabrication of NiTi wires with two distinct embedded transformation memories, exhibiting sufficient mechanical and cyclic properties for the next phase of the research.

Once an acceptable amount of NiTi actuation cycling data was acquired, the second part of the research consisted of training multiple recurrent neural network architectures with varying hyperparameters on the data and selecting the model which achieved the best performance. The hyperparameter optimization was performed on data with constant applied load, resulting in a model which successfully estimated the actuator's position with 99.2% accuracy. The optimized hyperparameters were then used to create a recurrent neural network model which was trained to estimate both position and force using the full acquired data set, capitalizing on the two embedded memories. The model achieved overall position and force estimation accuracy of 98.5% and 96.0%, respectively, on data used to train it, and 96.6% and 89.8%, respectively, on data it had never before encountered. The result of this study was the successful development of an accurate RNN-based position and force estimation model for NiTi actuators with two embedded phases. Using this model, a position controller was implemented which resulted in 95.9% position accuracy under varying applied loads.

Acknowledgements

I would like to thank my supervisors, Dr. Norman Zhou and Dr. Adrian Gerlich, for their kind support, encouragement provided during the study, and freedom for pursuing my research interests. I would also like to thank the Centre for Advanced Materials Joining (CAMJ) and its members for providing helpful insight and equipment which allowed this research to successfully progress.

I owe my deepest gratitude to Dr. M. Ibraheem Khan and the rest of the Smarter Alloys members for their endless guidance and generosity and for allowing me to spend countless hours completing this work at their facilities. This research would not have been possible without their support. I am particularly grateful to Nima Zamani for his selfless mentorship and contagious passion which inspired me to push the boundaries in the personal and professional aspects of my life.

To my fiancée Biljana, brother Ivan, parents, family, friends, colleagues, and others who I have not mentioned, I thank you for keeping me sane and well-fed throughout this journey. I am sincerely grateful for your patience, support, and the values and lessons you have instilled in me.

Finally, I would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) Industrial Partnership Program and the Mitacs Accelerate program for supporting this work.

I dedicate this to my mother and father,
Margarita and Nikola

You sacrificed everything to plant your garden
in an unknown world full of promise.

Enjoy their beauty;
the flowers bloom for you.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Shape memory alloys (SMAs) are materials which exhibit fascinating mechanical properties resulting from reversible lattice structure transformations. These properties, named the shape memory effect and pseudoelasticity, have been observed and studied as early as 1932 and can be exhibited by alloys with varying compositions. The most widely used composition is NiTi given its useful characteristics such as superior mechanical properties, low cost, and safe handleability, as well as up to 10% achievable strain recovery [1, 2]. These properties, in addition to their high energy density and large actuation stress and strain, have enabled the widespread applicability of SMAs in the robotics, automotive, aerospace, and medical fields [3, 4, 5, 6, 7]. However, the complex, stress-dependent behaviour of the inherent transformation hysteresis present in SMAs limits their use in applications which require accurate and precise actuation control.

Traditional SMAs contain a single transformation temperature (memory) attributed to their constant, homogeneous material composition. Embedding multiple transformations in a monolithic SMA would further broaden its applicability, enabling technologies such as actuators containing sections with distinct mechanical properties and varying actuation profiles at different temperatures. Past efforts have been taken to induce multiple transformation temperatures by locally altering SMA properties, resulting in using various methods such as gradient annealing [8] and laser annealing [9], all of which poorly affect the mechanical and fatigue properties of SMAs. A promising technology based on localized composition adjustment using high-powered laser processing has recently been developed by researchers at the University of Waterloo which yields monolithic NiTi SMAs with

multiple embedded transformation temperatures and relatively low impact on mechanical properties [10]. This method has led to novel configurations which have the potential to greatly improve the controllability of NiTi SMAs without the use of external sensors.

Artificial neural networks are another increasingly popular technology which have found great success in mathematical modeling and controls applications. Although theorized as early as 1957 as a computational version of biological neural networks found in the brain, recent advances in computational power have facilitated the effective use of the computationally-hungry neural networks [11]. The goal of this work is to create an algorithm for accurately and precisely controlling NiTi SMA wire actuators with two embedded monolithic transformation temperatures using artificial neural networks. Developing this fundamental tool will immensely expand the applicability of NiTi SMA wires within actuation applications.

1.2 Objectives

The first goal of this work was to perform an extensive analysis of laser-processed NiTi SMA wires with two embedded memories, including the effects of processing parameters on the thermal and mechanical properties of the actuator. Next, the cyclic behaviour of the processed NiTi actuators was to be fully characterized under varying applied currents and loads using custom-built tensile testing setups. The final goal was to analyze the cyclic actuation data acquired from the custom tensile testers and utilize artificial neural networks to develop an accurate and precise control model for the laser processed NiTi actuators that successfully models both major and minor hysteresis loops. More specifically, the following tasks were performed:

1. Mechanical and thermoanalytical characterization of the influence of laser processing and thermomechanical post-processing operations on NiTi wires.

2. Acquisition and analysis of thermal cycling behaviour of NiTi wire actuator with two embedded transformation memories, quantifying the effects of varying applied currents and loads on actuation and transformation behaviour.

3. Development of a neural network-based position and force estimation model for NiTi actuators using the acquired thermal cycling data.

4. Application of the neural network-based model in a NiTi actuator position controller.

1.3 Justification

The knowledge, methods, and tools resulting from this work will have a meaningful impact on the fabrication and implementation of NiTi actuators with multiple embedded memories. Studying the effects of laser processing will help construct guidelines for the creation of custom NiTi wires with multiple embedded transformations. Furthermore, exploring the factors which influence the actuation behaviour of NiTi will lead to a better understanding of the fundamental mechanisms behind the material. Finally, the developed controls model will enable the use of laser-processed NiTi actuators within applications requiring high precision and accuracy. Additionally, the model will be more accurate and adaptable relative to similar past models. Altogether, the goal of this work is to advance the state of NiTi actuators toward mass implementation throughout all possible fields.

1.4 Thesis Organization

This thesis is divided into six chapters.

Chapter 1 introduces the work performed in the study while providing the motivation and justification behind it.

Chapter 2 presents fundamental information about NiTi SMAs, describing their unique properties and characteristics. The most recent developments in NiTi laser processing are reviewed. Artificial neural networks are also introduced, including various state-of-the-art network architectures. The underlying theory behind their application is also discussed.

Chapter 3 outlines the experimental methods and equipment employed in this work.

Chapter 4 discusses the fabrication and characterization of monolithic NiTi wires with two embedded transformation memories, including results obtained from NiTi laser processing, thermomechanical post-processing, training, and cyclic actuation. The results include differential scanning calorimetry curves for transformation temperature characterization, optical microscope images for identification of voids and microstructure features, and tensile tests for evaluating the mechanical and actuation properties of the wire.

Chapter 5 utilizes the actuation data acquired in Chapter 4 to create a neural network-based model for position and force estimation of monolithic NiTi wires with two embedded transformation memories. The application of the best performing model in a position controller is also discussed.

Finally, Chapter 6 discusses the conclusions reached through this work and provides recommendations for future study.

Chapter 2

Literature Review

The unique behaviour exhibited by shape memory alloys (SMAs) can be attributed to their equally unique properties. This chapter aims to provide the fundamental knowledge necessary for understanding the functionality of SMAs by examining their microstructure, phase transformation dynamics, mechanical properties, and processing methods. Several challenges concerning SMAs relevant to this work are also outlined. In addition, basic concepts surrounding artificial neural networks are discussed in this chapter in order to provide insight regarding their use for SMA actuator controls. The following literature review was invaluable to building the foundation for the research discussed in the remaining chapters.

2.1 NiTi Shape Memory Alloys

Shape memory alloys are a type of intermetallic compounds which exhibit unique properties largely attributed to their reversible lattice phase transformation. This diffusionless temperature-dependent solid-to-solid transformation occurs between two crystal lattice phases, namely austenite (high temperature cubic phase) and martensite (low temperature monoclinic phase) [12, 13]. The driving factor behind the phase transformations is the temperature of the SMA, with each phase occurring within a certain temperature range in order to minimize the Helmholtz free energy of the system [14]. Phase transformations occur at specific transformation temperatures which primarily depend on the SMA's composition and thermomechanical processing history. In addition to varying temperature, the phase transformations can also be induced by applying a load to the SMA which places

strain on the crystal lattice, causing the martensite phase to be increasingly favourable [15].

Many different SMA compositions exist, including Cu-Al-Ni, Fe-Mn-Si, Au-Cd, and the most commonly used NiTi [16]. First discovered in 1962, NiTi was given the commercial name Nitinol to reflect its location of origin (the Naval Ordinance Laboratory [17]. NiTi SMAs have a near-equiatomic chemical composition (close to 50 Ni and Ti at.%) at room temperature, with increasing atomic percentage of either constituent leading to the formation of non-SMA intermetallic compounds. This effect can be seen in Figure 2.1, which shows the binary phase diagram for NiTi. The formation of the Ti_2Ni phase intermetallic compounds in Ti-rich NiTi is particularly undesirable as it leads to diminished tensile strength and elongation in NiTi, and as a result the Ti at.% should be kept below the threshold value for inducing this phase [18, 19]. Compared to other SMA compositions, NiTi possesses superior mechanical properties, corrosion resistance, biocompatibility, and higher electrical resistivity, making it a preferable candidate for applications in a variety of products [1, 20].

2.1.1 Martensitic Phase Transformation

Martensitic phase transformations are a diffusionless process which alters the material's phase, resulting in a product phase with identical chemical composition [2, 22]. Martensitic transformations are not exclusive to SMAs, as they also often occur in steel alloys. However, unlike SMAs, the transformation in steel is not reversible due to large changes in volume. In order to accommodate such changes in volume, the crystal lattice structure must accordingly adjust through either slip or twinning (refer to Figure 2.2) [23]. The first type of self-accommodation is lattice plane slipping, which is an irreversible deformation generally resulting in broken bonds and rearranged atomic structure. Slipping is able to accommodate variations in both shape and volume. The martensitic transformation occurring in steel results in slip deformation, making the process irreversible.

The second self-accommodation process is twinning, which solely accommodates changes in shape. SMAs experience twinning accommodation during martensitic transformation, making the transformation a reversible process. Due to twinning during transformation, SMAs experience a change in shape while maintaining a constant volume. As can be seen in Figure 2.2, twinning results in mirrored atomic structures with respect to the plane of deformation. All bonds remain intact during the twinning process, making the process more energetically favourable than slipping [23]. As a result, twinned lattice structures can reorient themselves in response to an applied stress, resulting in considerable amounts

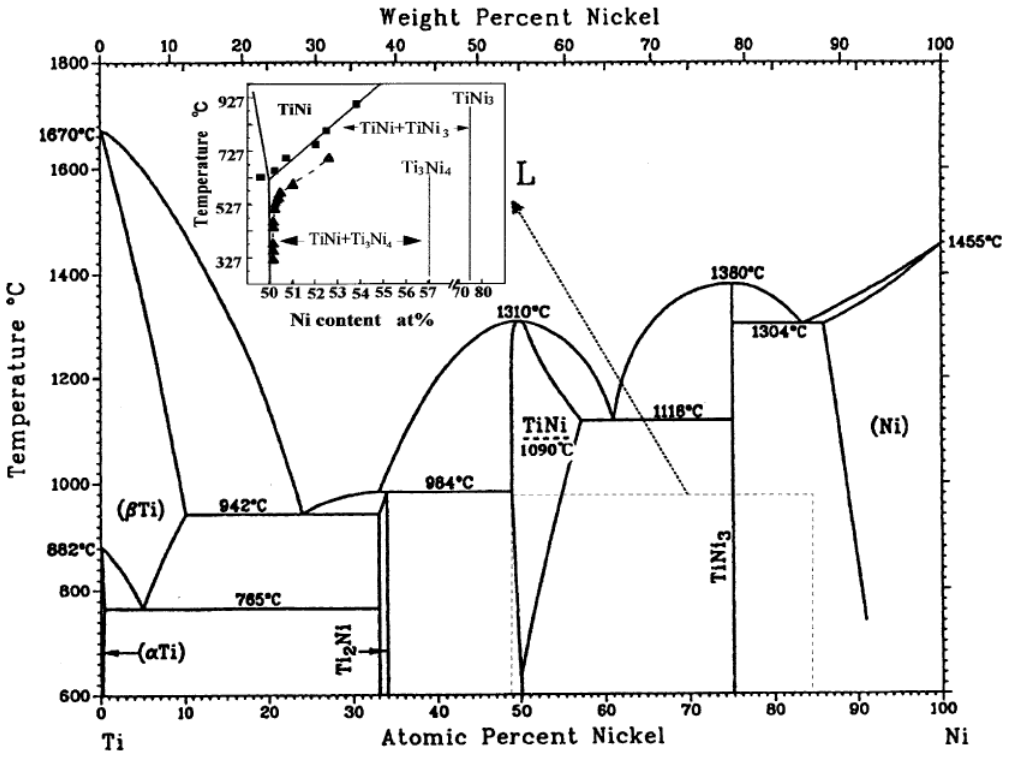

Figure 2.1: Binary phase diagram of NiTi [21].

of strain - this process is called detwinning and is depicted in Figure 2.3 [23]. Further applying stress after detwinning can result in plastic deformation [24].

As previously mentioned, the SMA phase transformations occur at temperatures characteristic to the individual material. There are four distinct characteristic temperatures, namely martensite start (M_s), martensite finish (M_f), austenite start (A_s), and austenite finish (A_f) - refer to Figure 2.4. During heating, A_s and A_f represent the temperature where the martensite-to-austenite transformation begins and ends, respectively. Similarly, M_s and M_f are the characteristic temperatures during cooling which respecitvely represent the beginning and ending of the austenite-to-martensite transformation. During both heating and cooling, the transformation process occurs over a temperature range which generally spans $5 - 30K$ [25]. Temperatures within this range represent partial phase transformations. Another effect visible in Figure 2.4 is that the heating and cooling transformations do not occur at the same temperatures - M_s usually occurs at a lower temperature than A_f. This effect is known as the transformation hysteresis. This temperature

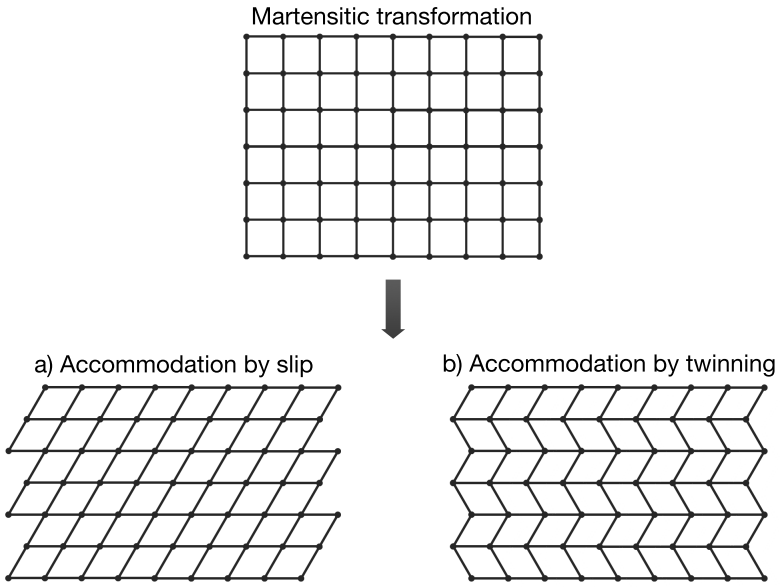

Figure 2.2: Illustration of martensitic transformation volume change accommodation by a) slip and b) twinning.

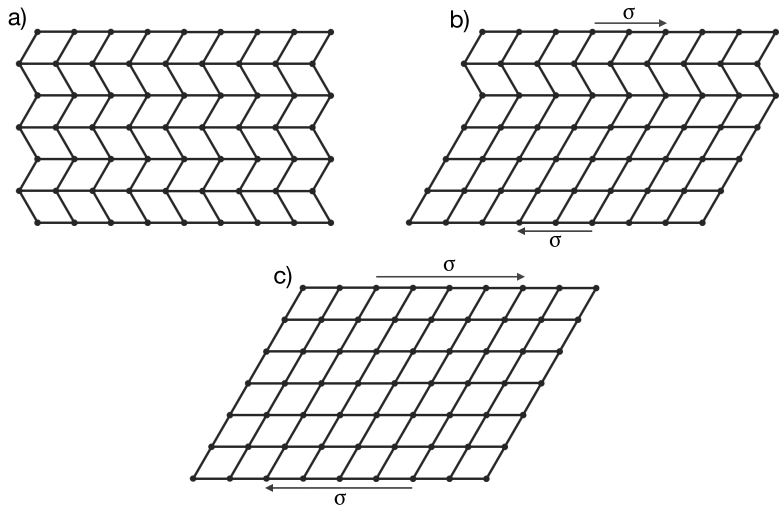

Figure 2.3: Illustration of shear stress accommodation by detwinning showing a) twinned (no stress), b) partially detwinned (moderate stress) and c) detwinned (high stress) microstructure.

variation between heating and cooling can range between 10-50K and depends on many factors including chemical composition and thermomechanical processing [25, 17, 26]. Understanding transformation hysteresis is essential for this work, and so the topic is explored in detail in Section 2.1.5.

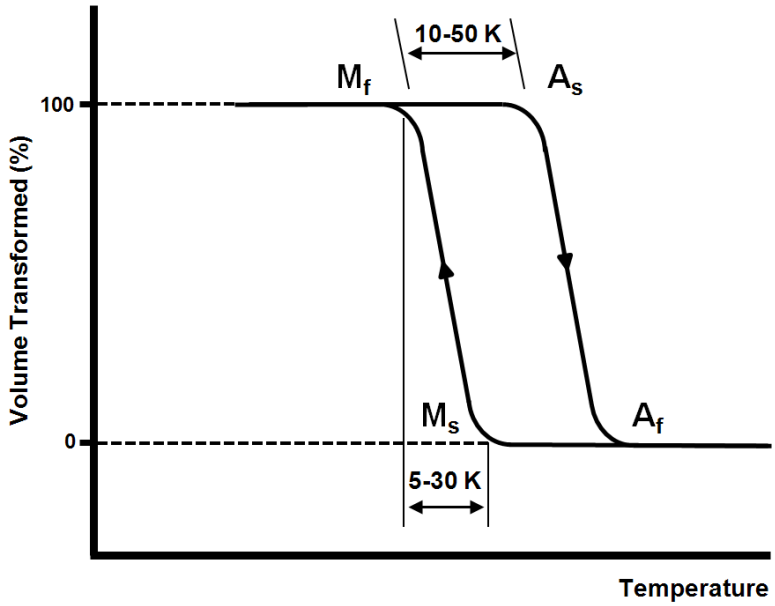

Figure 2.4: Fraction of transformed volume as a function of temperature. Characteristic transformation temperatures are also shown [25].

The characteristic phase transformation temperatures depend on various factors which affect the freedom of motion between the crystal lattice structures within the material [26]. Two common methods for determining the transformation temperatures of SMAs are differential scanning calorimetry (DSC) and electrical resistance measurement, both of which are shown in Figure 2.5 [27, 26]. One factor which affects these transformation temperatures is thermomechanical processing which inhibits twin boundary mobility by breaking down larger grains, producing a more refined microstructure with an increased number of twin boundaries [15, 27]. This in turn improves the mechanical properties of the material, resulting in superior yield strength and fatigue life. Another important factor is the chemical composition of the SMA. In the case of NiTi, the Ni/Ti ratio has been shown to have significant impact on its transformation temperature [23, 26]. This effect is illustrated in Figure 2.6, which shows a significant decrease in M_s transformation temperature as the Ni at.% increases from 49.7% to 51% [28, 24]. Below Ni 49.7 at.%

NiTi becomes Ti-rich, resulting in the creation of Ti_2Ni intermetallic compounds which deplete the excess Ti. Unlike Ni_3Ti and other Ni-rich intermetallic compounds, Ti_2Ni does not affect the transformation temperature of NiTi. In fact, the introduction of Ti_2Ni into the NiTi microstructure results in a deterioration of mechanical properties such as fatigue life [18, 19]. Various ways exist for altering the composition of NiTi, including forming of secondary phases, bulk alloying, and laser processing (discussed in Section 2.1.4).

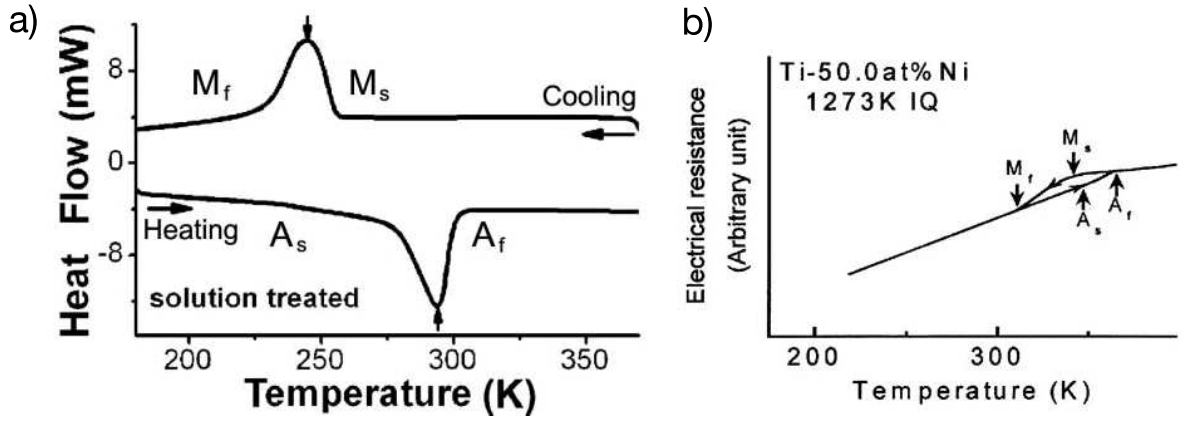

Figure 2.5: Determination of characteristic transformation temperatures using a) DSC [26] and b) electrical resistance measurement [27].

Other factors affecting the transformation temperature of SMAs are thermal/mechanical cycling and applied stress. Thermal cycling leads to the creation of dislocations as a result of incomplete transformation accommodation [29, 30]. This effect, called transformation induced plasticity, results in an increased phase transformation driving force, effectively lowering the transformation temperature. Transformation induced plasticity also leads to unrecoverable plastic strain and lower actuation strain [31]. Additional effects of thermal cycling are decreasing actuation strain and permanent (plastic) elongation. Figure 2.7 shows the thermal cycling of a NiTi wire under constant load. The application of stress on SMAs also influences their transformation temperature - this results in the effect known as pseudoelasticity which is further discussed in Section 2.1.2.

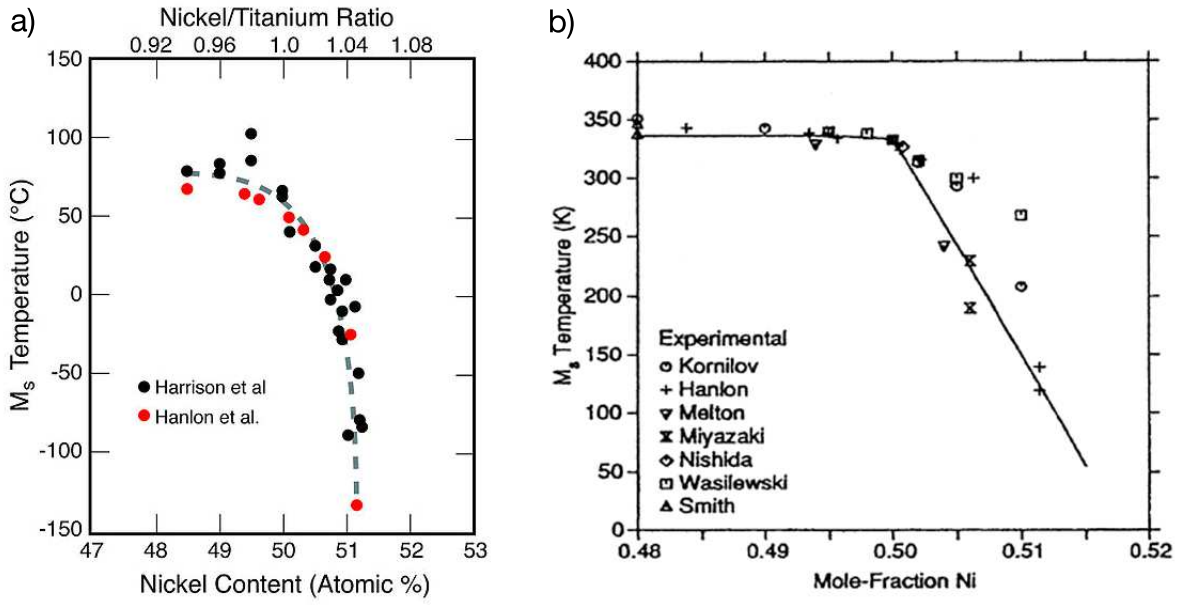

Figure 2.6: Effect of NiTi composition on M_s transformation temperature as reported by a) Duerig et al [23] and b) Tang et al [26].

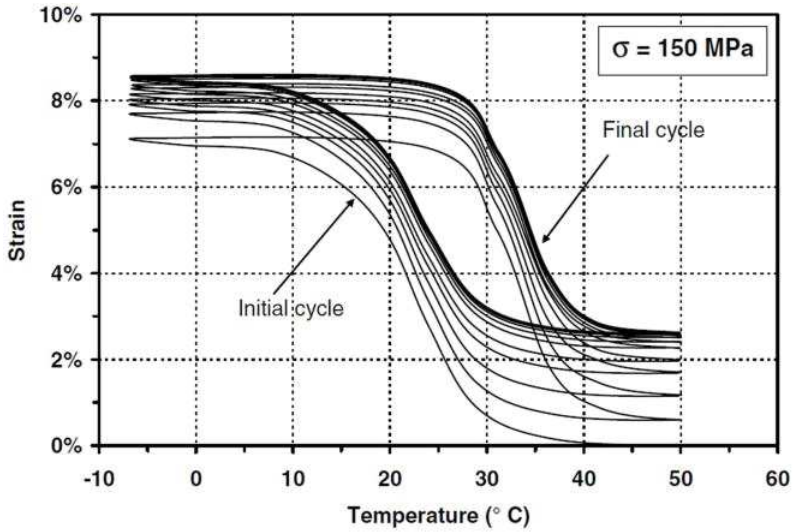

Figure 2.7: Thermal cycling of NiTi SMA wire with constant applied load of 150 MPa [31].

2.1.2 Pseudoelasticity

As mentioned in Section 2.1.5, the application of stress on SMAs causes an increase in their characteristic transformation temperatures. The application of stress adds energy to the material, effectively lowering the amount of energy needed to achieve the martensite phase [44]. If sufficient stress is applied, assuming constant temperature, an SMA's phase can be fully transformed from austenite to martensite [45]. Removal of the applied stress causes the transformation temperatures to decrease to their original values, causing the material to return to the austenite phase and effectively reverting the transformation. Two criteria must be met for stress-induced martensite to occur, the first being application of sufficient stress. The second criterion is that the SMA must be in the austenite phase at the temperature of interest, meaning the testing temperature T must fall within the $A_f < T < M_d$ temperature range for the specific SMA. M_d represents the temperature above which stress-induced martensite is no longer observed and is known as the intersection of critical stresses between slip and martensitic transformation [46, 47]. If $T < A_f$, the SMA phase will not be fully austenite and so only the austenite portion will experience the stress-induced martensite transformation. On the other hand, if $T > M_d$, because the critical stress exceeds stresses which would cause plastic deformation in austenite, the SMA will behave like a regular material exhibiting plastic deformation [48].

The typical stress-strain curve of a pseudoelastic NiTi material is illustrated in Figure 2.8. Pseudoelastic (PE) NiTi behaves very differently compared to conventional elastic materials as a result of stress-induced martensite. Firstly, due to the hysteretic nature of NiTi, two stress-strain curves are necessary in order to fully characterize its tensile properties: a loading curve (application of stress) and an unloading curve (removal of stress). The loading curve occurs at a higher stress than the unloading curve, as shown in Figure 2.8. The relatively flat region in both the loading and unloading stress-strain curves around $2 - 4\%$ strain corresponds to the stress-induced phase transformation. During loading, stress-induced martensite occurs at a sufficiently high applied stress - the stresses at which the stress-induced martensite transformation begins and ends are σ^{M_s} and σ^{M_f}, respectively. Similarly, during stress unloading, the reverse transformation occurs from martensite to austenite at the characteristic stresses σ^{A_s} and σ^{A_f} for the start and finish of the transformation, respectively. For a specific NiTi composition, these values are easily obtained from the material's stress-strain curve as shown in Figure 2.8.

Pseudoelasticity is a very useful property in practice, as it results in a high-strain region while maintaining a relatively constant stress. Compared to conventional elastic materials, applying a stress equal to or greater than σ^{M_f} to NiTi results in significantly greater strain due to the austenite-to-martensite transformation. NiTi can reach maximum PE strain of

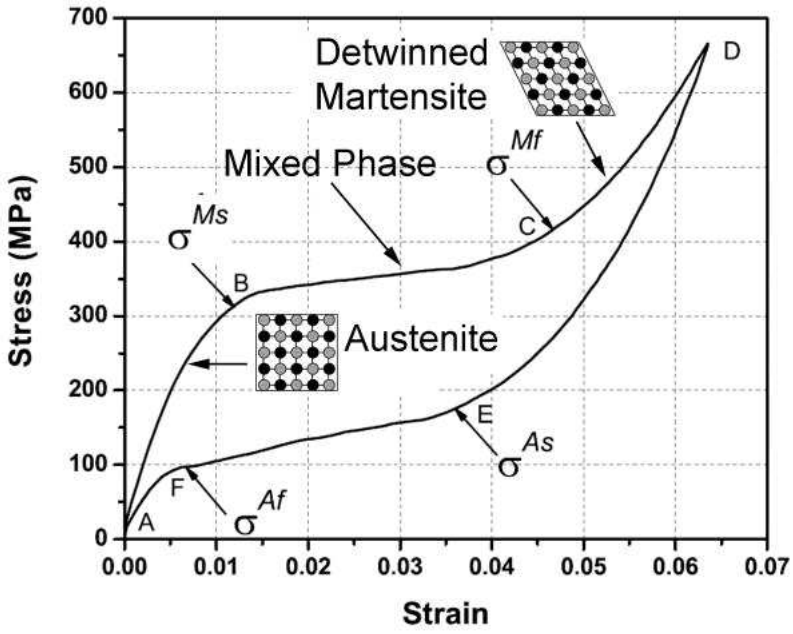

Figure 2.8: Typical pseudoelastic stress-strain curve of NiTi depicting the location of critical stresses for inducing martensite. The loading curve is labeled as $A \rightarrow B \rightarrow C \rightarrow D$, whereas the unloading curve is labeled as $D \rightarrow E \rightarrow F \rightarrow A$ [31].

13% before inducing plastic deformation [49].

2.1.3 Shape Memory Effect

While in the austenite phase, NiTi SMAs experience pseudoelasticity with sufficient applied force - this effect is not observed for martensitic NiTi. The equivalent effect for NiTi in the martensite phase results in the shape memory effect (SME) [16]. Applying sufficient stress to twinned martensitic NiTi causes detwinning to occur, leading to a stress plateau exhibiting large amounts of strain at relatively constant stress. However, unlike pseudoelasticity, unloading the stress does not cause the material to return to its original shape. Instead, the martensite lattice remains detwinned upon unloading, resulting in a significant amount of unrecovered strain. This effect is shown in Figure 2.9, which compares the stress-strain curves of PE and SME NiTi [50]. Once the detwinned NiTi is heated past A_f, transformation to the austenite phase causes the material to return to its original length, effectively recovering the detwinning strain. The cycle of loading, straining, unloading,

and finally heating to induce and recover large amounts of strain in NiTi is known as the shape memory effect.

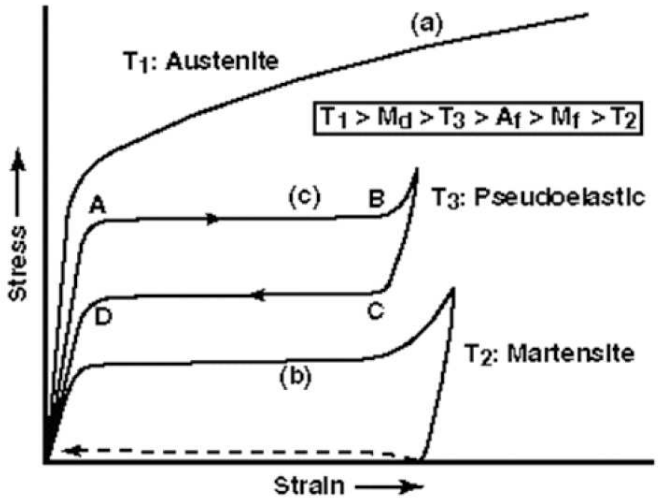

Figure 2.9: Illustration of NiTi stress-strain curves exhibiting a) plastic deformation ($T > M_d$), b) shape memory effect ($T < M_f$), and c) pseudoelasticity ($A_f < T < M_d$) [50].

2.1.4 Multiple Embedded Memory NiTi

Traditional NiTi SMAs are composed of a constant Ni/Ti ratio within the monolithic material, resulting in a single set of characteristic transformation temperatures. In order to further advance the functional properties of NiTi, efforts have been made to create a monolithic material with multiple sets of transformation temperatures resulting from multiple chemical compositions at different locations in the material. Methods used to achieve this feat include joining of multiple alloys [51, 52, 53], powder metallurgy [54, 55], laser annealing [9], and gradient annealing [8], among others. Despite the list of possible methods for achieving this feat being extensive, each of these methods faces unique challenges which limits their ability to effectively produce monolithic multiple memory NiTi. Although effective for simple geometries with similar compositions, joining alloys has proven to be difficult for complex shapes and differing compositions, leading to the creation of undesirable intermetallic compounds. The heat treatment methods come with

their own set of drawbacks, including poor controllability of resulting properties and large time requirements.

A novel, recently-developed technique is able to embed multiple transformation temperatures in monolithic NiTi SMAs through localized composition changes induced by laser beam processing [10, 56]. This method is made possible by the dissimilar vapour pressures of Ni and Ti as shown in Figure 2.10. It can be seen that the vapour pressure of Ni is significantly larger than that of Ti, especially in the temperature range of $1700 - 2500K$ where Ti does not experience noticeable vaporization. Due to these characteristics, the localized laser processing of near-equiatomic NiTi yields a greater vaporization flux of Ni than that of Ti, resulting in controlled removal of Ni from the local alloy composition through vaporization. As discussed in Section 2.1.1, the phase transformation temperatures of NiTi closely follow the ratio of Ni to Ti within the material composition. As a result, this method can locally vaporize Ni in order to change the Ni/Ti ratio, effectively adjusting the transformation temperatures of the NiTi SMA [57, 58].

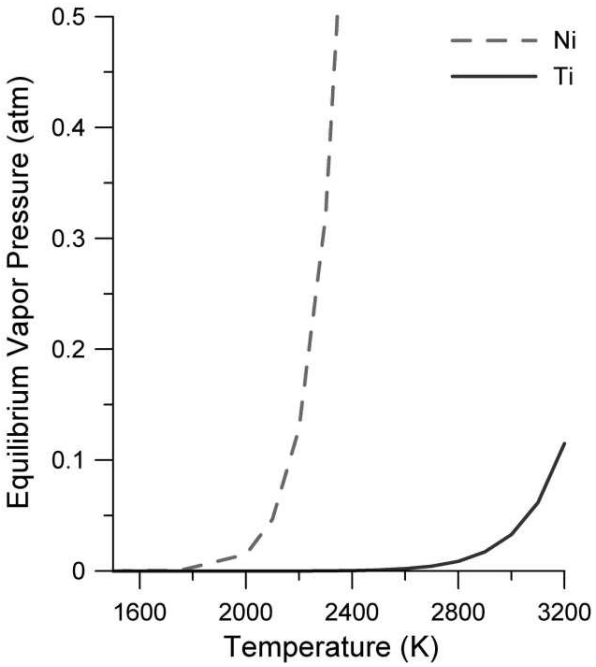

Figure 2.10: Vapour pressures of Nickel and Titanium from equiatomic NiTi [10].

According to Figure 2.6, reducing the Ni content yields increased transformation temperatures until $49.7at.\%$ Ni is reached. Below $49.7at.\%$, the material becomes Ti-rich and

does not experience further rise in transformation temperatures through changes in composition. The Ti-saturation therefore acts as an upper limit for transformation temperatures achievable through this method. This upper limit was experimentally found to be $340K$ for the particular tested NiTi composition [10].

The degree of Ni vaporization can be controlled by adjusting various laser parameters, including the number of pulses per location and pulse duration. Increasing both the number and duration of pulses has been shown to increase the NiTi transformation temperature [10]. Figure 2.12 a) illustrates the effect of number of pulses on the transformation temperature. As the number of pulses increases, the second transformation peak in the differential scanning calorimetry curves becomes more intense and its peak shifts toward higher temperatures, whereas the base metal peak appears to shrink while remaining at the same temperature. The effect of pulse duration on transformation is shown in Figure 2.12 b), and it can be seen that increased duration also results in an increased processed NiTi peak intensity. Additionally, it was found that shorter pulse duration results in larger transformation temperature compared to longer duration due to vaporization flux decreasing with pulse duration [59]. Using this technology, monolithic NiTi SMAs which contain multiple transformation memories have been successfully produced [57, 60, 61, 62, 63]. The proof of concept for the technology is shown in Figure 2.11, which shows two separate memories achieved at different applied temperatures.

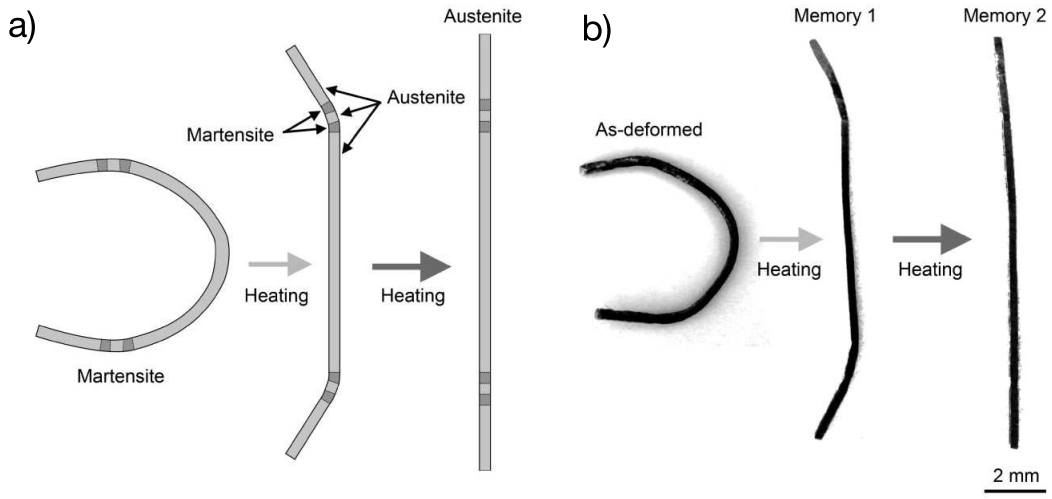

Figure 2.11: Proof of concept showing a) schematic illustration and b) actual images of two transformation temperatures (T_1, T_2 where $T_2 > T_1$) embedded into a monolithic NiTi wire heated to two different temperatures ($T_1 > T > T_2$ and $T > T_2$) [10].

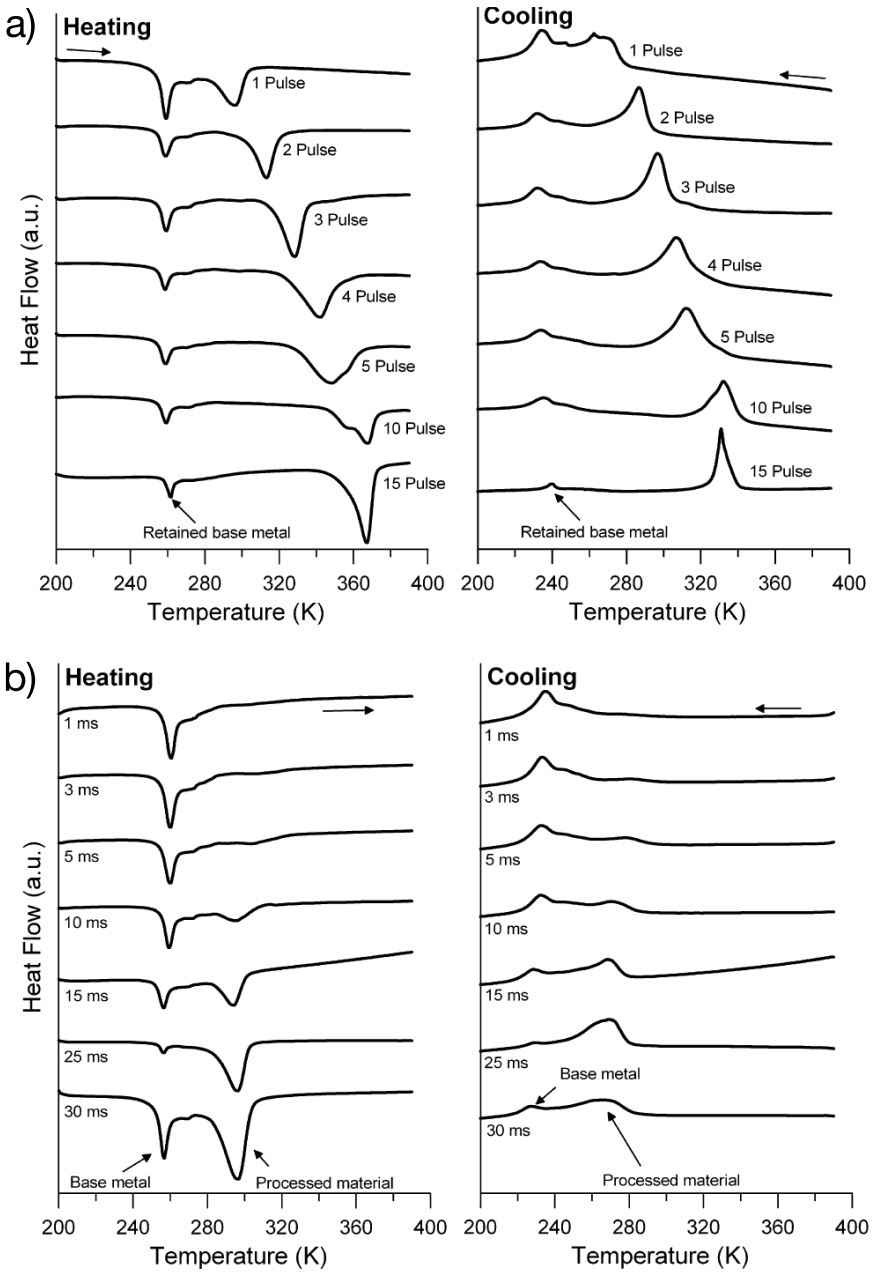

Figure 2.12: Effect of a) number (0.6 kW peak power with 30 ms pulse duration) and b) length (single pulse) of laser processing pulses on the DSC curves of NiTi [10].

2.1.5 Transformation Hysteresis

The existence of the transformation hysteresis curves in the temperature-related actuation behaviour of SMAs introduces a significant level of complexity for the creation of accurate and precise control models. During phase transformations, the crystal lattice moves in order to accommodate changes in the SMA's shape. The motion of the austenite-martensite and martensite-martensite phase interfaces results in frictional effects which are the main cause of transformation hysteresis [17]. In addition, since the distribution of defects in the lattice affects the transformation hysteresis, the cyclic loading history experienced by an SMA also influences its hysteretic behaviour.

Stress applied to the SMA also significantly alters its transformation characteristics as shown in Figure 2.13, where the four lines corresponding to A_s, A_f, M_s, and M_f show how NiTi's transformation temperatures change with varying applied loads [32, 33]. All four characteristic temperatures exhibit an increase in transformation temperature with increased applied load. The application of load to SMAs also introduces further complexity in the hysteretic behaviour of SMAs. From Figure 2.13 it can be seen that the slopes of A_s and A_f lines are larger than those of M_s and M_f, meaning that applied stress causes a larger increase in the austenite-to-martensite cooling transformation temperatures than their heating counterparts. As a result, the hysteresis curve shrinks with respect to temperature as stress increases. This effect occurs up to a certain limit (in terms of both stress and temperature), after which permanent deformation occurs [34]. As SMA actuators will likely experience varying stresses within their applications, this stress-dependent hysteretic behaviour acts as a major hurdle for the creation of accurate SMA control models.

Despite the aforementioned challenges, many attempts have been made to create a control system for SMAs which accurately takes into account their hysteretic behaviour. Bo and Lagoudas proposed a thermomechanical model based on the Preisach hysteresis model which resulted in good estimation of minor SMA hysteresis loops [35]. Madill and Wang expand on the traditional proportional-integral-derivative (PID) controller by including the theoretical stability of SMAs, resulting in the ability to model minor hysteresis loops [36]. Numerous attempts have been made to create a neural network-based model with varying architectures including feed-forward neural network [37], nonlinear autoregressive model with exogeneous inputs (NARX) recurrent neural network [38], hysteresis functional link artificial neural network (HFLANN) [39], and more [40, 41, 42, 43]. Although many of these neural network-based control models succeeded in achieving accurate SMA control with various applied loads, they were performed on traditional NiTi with a single embedded memory therefore limiting the amount of information that can be obtained without the use of secondary sensors.

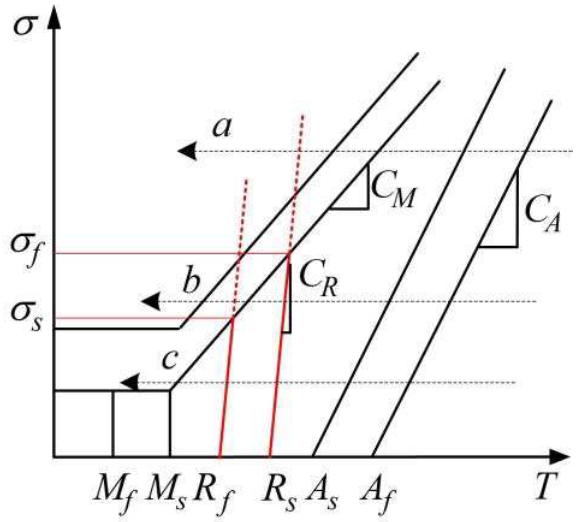

Figure 2.13: The effect of applied stress on the characteristic transformation temperatures of NiTi [33].

A recently developed model performs position estimation of NiTi actuators with two distinct monolithic phases (martensite and austenite) by measuring the electrical resistance across each phase [100]. Position estimation of the actuator was performed using a mathematical model based on the unique thermomechanical properties of each wire section. Despite reportedly achieving accurate sensor-less position estimation, several key disadvantages were identified which limit the model's applicability. Firstly, no physical model is able to perfectly capture reality, resulting in estimation errors. In the case of this model, significant deviations from reality were observed at the end of the heating cycle, which would significantly lower the model's accuracy. Furthermore, the model greatly relies on the material properties of the actuator, as it assumes that the two embedded phases perfectly exhibit PE and SME behaviour. Any deviations from ideal properties will cause more inaccuracies in the model. Due to this, the model is restricted to modeling pure PE and SME NiTi actuators. Differences in material composition would also cause the model to break down, as different hysteretic behaviour would be exhibited by the actuator. In order to be used universally for SMA actuators, unique models would have to be developed for each individual actuator material. The control method developed through this research addresses all of these drawbacks, improving the applicability of SMA actuators.

2.1.6 Post-Processing

In order to successfully vaporize Ni, laser processing must locally increase the temperature of the processed region to upwards of $1700°C$ (see Figure 2.10). This rapid heating and subsequent cooling of the metal leads to the creation of coarse grains across the heat-affected zone in addition to solidification fronts, both of which allow cracks to propagate more easily. Another effect caused by laser processing is the creation of intermetallic phases throughout the wire due to the localized increase in temperature required for nickel vaporization [18, 10]. Both of these effects are detrimental to the mechanical properties of the wire. In order to recover the properties of base NiTi, sufficient thermomechanical post-processing treatment was performed on the wire to prevent the generation and propagation of cracks [64].

Wire Drawing

Stock NiTi wires are not commercially available in any desired diameter, and so wire drawing can be performed to obtain wires with diameters specific for their corresponding applications [65]. The typical wire drawing process is depicted in Figure 2.14, where a wire (in this case of circular cross-section) passes through a die with a decreasing diameter. This process consequently reduces the wire's cross-sectional area and increases its length through conservation of volume. Another side effect of wire drawing is the breaking down of the metal's crystal lattice into very fine structures and eliminating large grains and solidification fronts created by the laser processing. This effect results in the loss of NiTi's unique lattice-dependent properties such as pseudoelasticity and shape memory effect [66, 67].

As the wire passes through the die, it experiences three types of stress: uniform work (W_U), redundant work (W_R), and frictional work ($W_{friction}$). The useful work necessary for achieving wire elongation and diameter reduction is W_U, whereas W_R and $W_{friction}$ result from the material flow change through the die and the friction between the wire and die surface, respectively. Both W_R and $W_{friction}$ are detrimental to the wire drawing process and impose an upper limit of $30 - 35\%$ on the achievable area reduction with a single pass. The total stress experienced by the wire (D) is the sum of W_U, W_R and $W_{friction}$ as shown in Equation 2.1 [68, 65]. High drawing stresses can fracture the wire during area reduction, and so appropriate precautions must be taken to prevent this including lubrication, intermediate thermal annealing, and limiting area reduction.

$$D = W_U + W_R + W_{friction} \tag{2.1}$$

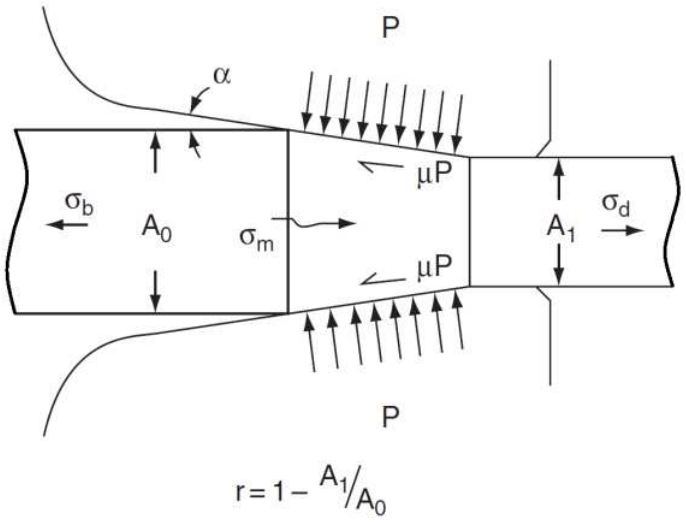

Figure 2.14: Illustration of wire passing through a wire drawing die [68].

Heat Treatment

As mentioned in Section 2.1.6, wire drawing breaks down the crystal lattice of the NiTi wire into a very fine microstructure, effectively eliminating the shape memory effect and pseudoelasticity. In order to recover NiTi's characteristic properties, heat treatment must be performed immediately after wire drawing. Heat treatment causes the material's grain structure to grow and form large crystal lattice capable of exhibiting macroscopic amounts of strain upon phase transformations. Other benefits of heat treatment after cold working are resistance to crack evolution and propagation and strengthening through formation of Ni-rich precipitates [69]. Heat treatment temperatures of $350 - 475°C$ have found to result in maximized formation of Ni-rich precipitates through optimization of diffusion and nucleation processes [70].

2.2 Artificial Neural Networks

The human brain is made up of billions of neurons connected together in a complex network resulting in functionality beyond our full understanding [71]. Neurons are cells which transmit and receive electrical impulse signals to and from neighbouring neurons. The structure of a neuron is shown in Figure 2.15. As shown in the illustration, each neuron

contains multiple branches called axons which further branch out many times into dendrites. The connections of neurons terminate at synapses, which are nodes where neurons connect to other neurons. As each neuron has many connections to neighbouring neurons (ranging from tens to several thousand synapses), the various inputs are merged into a single input detected by the neuron. In order for a signal to be produced within the axon of a neuron, the combined signals from each dendrite must surpass a certain threshold value. If an electrical pulse is received by the neuron, its output depends non-linearly on this combined input. The synaptic connections between neurons can vary in strength and can be chemically adjusted by the brain through exposure to appropriate stimuli, and this process occurring at a large scale is thought to result in learning.

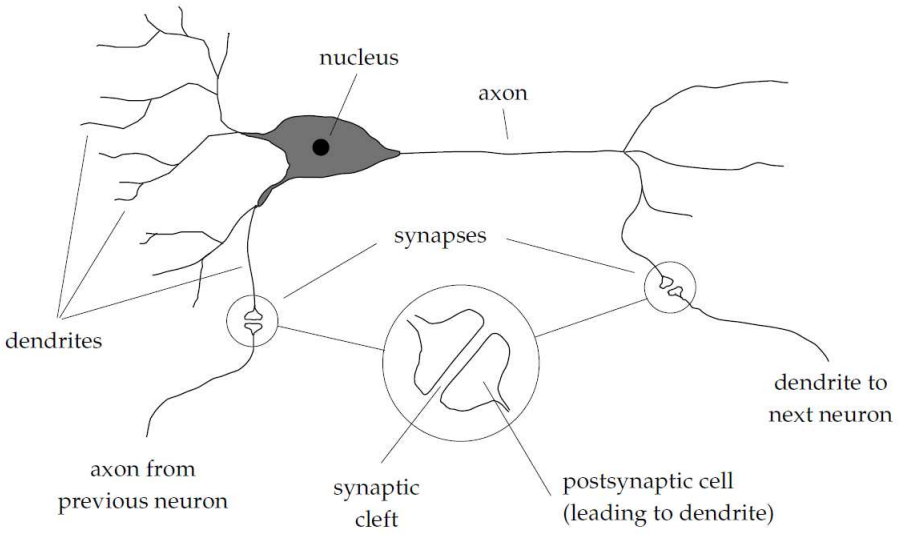

Figure 2.15: Structure of a biological neuron found in the brain [71].

Artificial neural networks (ANNs) are designed to mimic many of the aforementioned characteristics found in biological neural networks [72, 73]. Similarly to the brain, ANNs are made up of networks of many individual processing units (artificial neurons) linked with varying connection strengths. The processing unit which represents an artificial neuron in ANNs is shown in Figure 2.16 [74]. Each artificial neuron (also known as perceptron) can accept multiple numerical inputs which are summed together and passed into the neuron, analogous to their cell counterparts. If this summed input is significant relative to a certain threshold, a nonlinear activation function is applied to the input summation in order to transform the sum and generate the neuron's output value.

As is the case with synapses in biological neurons, the connection strength between

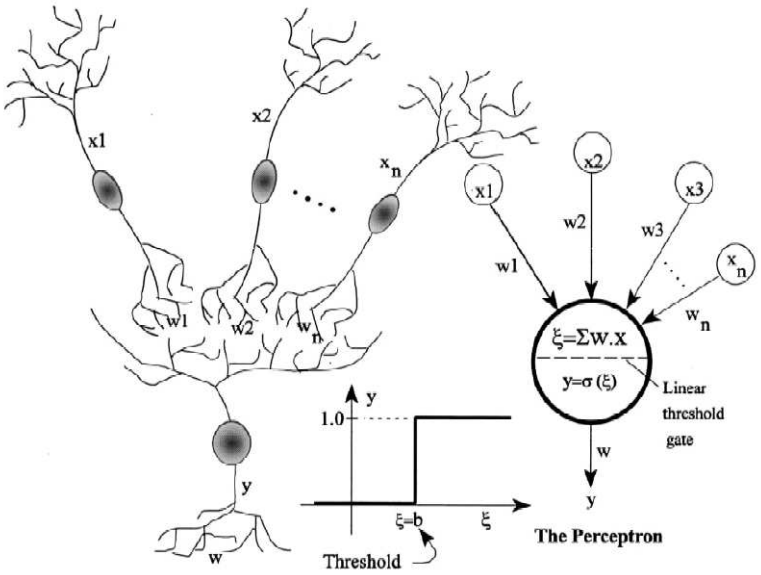

Figure 2.16: Structure of an artificial neuron found in artificial neural networks [74].

neighbouring artificial neurons can vary. ANNs emulate the variable synapse connections by assigning a weight to each artificial neural connection and multiplying each input passed to the neuron through the input's corresponding connection weight [74, 71]. The weight values fall in the range of $[0, 1]$, where a weight of 0 is effectively equal to no connection existing between the two adjacent neurons whereas a weight of 1 transmits the entire input value to the neuron. Once the input values are summed, a non-linear function (called the activation function) is applied to the sum in order to generate the input [74]. Figure 2.17 shows various activation functions commonly used in ANNs, with the most common functions being sigmoid and rectified linear unit (ReLU) [75]. In order for an output to be generated by the neuron, the input sum must produce a value after application of the activation function. For example, the sum of inputs must be greater than zero when using the ReLU activation function, otherwise no output will be generated by the neuron. This information transfer process through each neuron is mathematically represented by Equation 2.2 for k inputs, where $x_{1,n}, x_{2,n}, ..., x_{k,n}$ are the inputs received by the neuron, $w_{1,n}, w_{2,n}, ..., w_{k,n}$ are the weights of each connection to adjacent neurons, y_n is the output, and $f_n()$ is the activation function.

$$y_n = f_n(w_{1,n}x_{1,n} + w_{2,n}x_{2,n} + ... + w_{k,n}x_{k,n}) \tag{2.2}$$

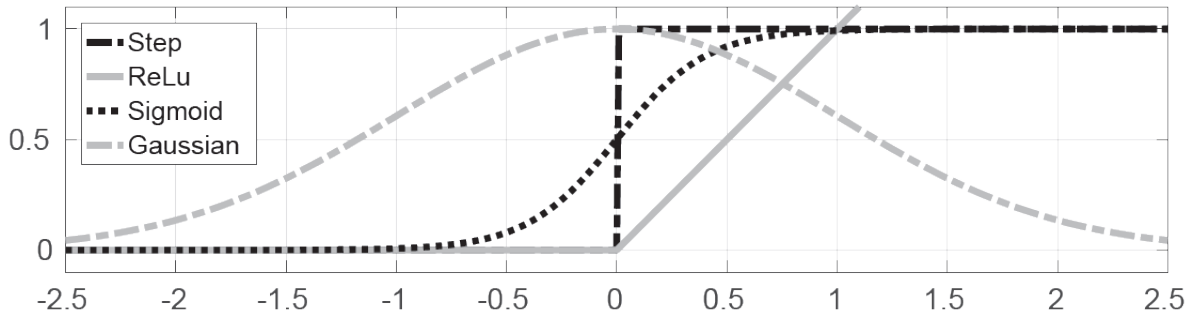

Figure 2.17: Various activation functions commonly used in neural networks [75].

Although neurons do not appear impressive when examined in isolation, their capability vastly increases when a larger number are used to create an interconnected network [71, 74]. These networks of neurons are known as neural networks (or NNs) and consist of layers of neurons as shown in Figure 2.18. Neurons in each layer are only connected to neurons in adjacent layers, with varying degrees of connectivity depending on the network architecture. One such network configuration is the fully connected NN, where each neuron is connected to all neurons in the adjacent layers as shown in Figure 2.18.

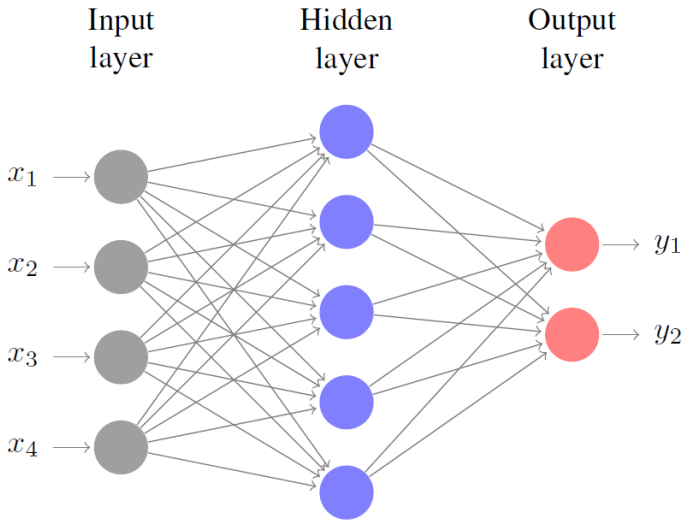

Figure 2.18: Image depicting a fully connected neural network with one hidden layer where x_1, x_2, x_3, x_4 are the inputs and y_1, y_2 are the outputs [76].

The first layer in a NN is known as the input layer, which is where the input data enters

the network. Similarly, the final layer yields the outputs calculated by the network and is known as the output layer. All other layers found between the input and output are known as hidden layers. Any NN which contains more than one hidden layer is known as a deep neural network (DNN) [71]. In traditional feed-forward NNs, the data propagates forward through the network until the output layer is reached. During the forward propagation, each neuron in the current layer sums its weighted inputs and applies its activation function, passing on the value to all neurons in the following layer to which a connection exists. This process continues until the final layer is reached, where each output neuron produces an output for the network.

It is important to recall that each propagation through a node results in the multiplication of data with the corresponding node's weight. In fact, NNs are trained to produce specific outputs from selected input data by appropriately adjusting the weights of all connections between the neurons. One fundamental method for weight adjustment is backpropagation using gradient descent, where the difference between the expected and calculated value is obtained and propagated backward through the network in the direction of the maximum cost gradient with respect to any weight [77, 78]. Recently, gradient descent has been replaced by faster, more efficient methods such as Adam and AdaGrad [79]. Using backpropagation (or other similar algorithms), the NN weights are adjusted in order to minimize the difference between the predicted and actual outputs. This process is called training and continues until the NN's calculated output is within a desired error margin compared to the expected output.

2.2.1 Recurrent Neural Networks

It is well-known that a multilayer NN can model any multiple input nonlinear function with arbitrary precision [73]. This property makes NNs a prime candidate for modeling the hysteresis behaviour of SMAs. However, feed-forward NNs are structured for function approximation and therefore poorly model dynamic and temporal-based systems [80]. It is important in real-time controls for the NN to be able to predict sequences of future output values based on inputs and previous outputs. A special class of NNs called recurrent neural networks (RNNs) take into account the state from previous outputs and effectively incorporate a temporal aspect to traditional feed-forward NNs [81]. As a result, RNNs are ideal for modeling time series and forecasting future function behaviour based on historical data. These properties make RNNs ideal for modeling and predicting SMA behaviour. Figure 2.19 depicts how RNNs function over time, with x_t representing the input, A representing the RNN state, and h_t representing the output, all occurring at time t.

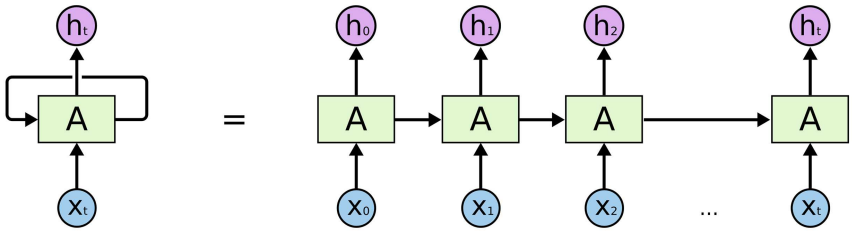

Figure 2.19: Illustration showing a single looping (left) and unrolled (right) recurrent neural network cell [82].

Conventional RNNs use feedback connections to effectively remember recent inputs by storing their representations in the form of activation functions. However, this gradient-based backpropagation through time tends to yield either vanishing or exploding gradients. This behaviour makes it difficult for RNNs to backpropagate through longer periods of time, limiting the networks' modeling capabilities to 5-10 time steps [83].

One method for overcoming this backpropagation limit is the use of an RNN architecture called long short-term memory (LSTM) [80]. The LSTM architecture consists of memory blocks, which are units found in the recurrent hidden layer. In order to recall previous network states, the memory blocks contain self-connected memory cells. The flow of information into and out of the LSTM memory blocks is controlled by three types of gates, namely the input, output, and forget gates. The input gate handles activation information being input into the memory block, and similarly, the output gate handles the activation information flowing out of the memory block and into the rest of the network. Early LSTM architectures did not include the forget gate and suffered from the inability to process continuous inputs which were not split into subsequences. The forget gate was added to address this issue, allowing the memory block to reset its state at the beginning of subsequences [83]. In addition to these, peephole connections were added in order to learn precise output timings. The standard LSTM cell is shown in Figure 2.20.

Another recently proposed architecture for RNNs is the gated recurrent unit (GRU), which is a simplified variation of the LSTM architecture [84]. The GRU only contains one reset and one update gate, compared to the more complex LSTM cell which contains three gates - see Figure 2.21 [85]. Furthermore, the GRU does not incorporate the LSTM memory cell. Even with these simplifications, it has been found that GRUs exhibit comparable performance to LSTM units for polyphonic music modeling and speech signal modeling [84]. In addition, both LSTM and GRU were found to outperform traditional tanh cell RNNs in the aforementioned modeling tasks.

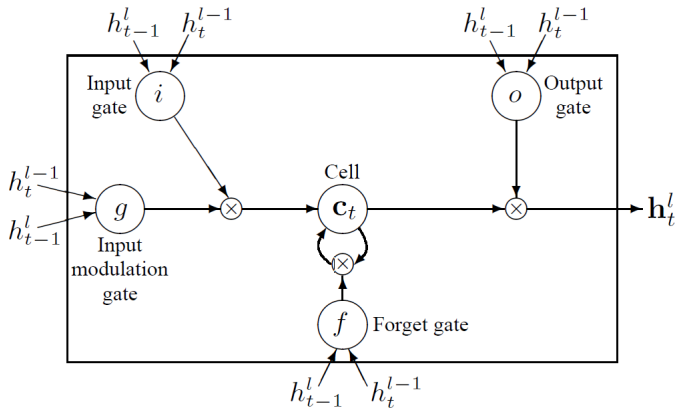

Figure 2.20: Standard LSTM memory cell containing input, forget, and output gates [81]

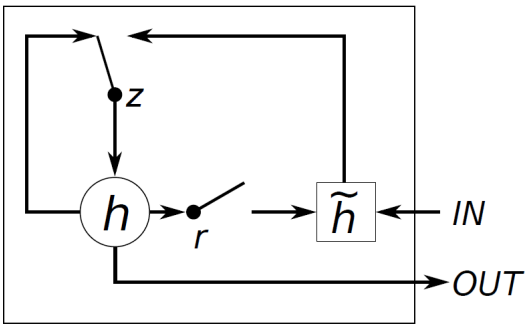

Figure 2.21: Standard GRU cell containing reset gate r and update gate z [84]

By applying the LSTM or GRU architectures, the RNN can be trained on many input sequences in order to learn the function shape and can predict future outputs based on historic output sequences. Using this trained network, the actuation behaviour of SMA wires can be estimated in the present and predicted several time steps into the future according to previous inputs and outputs which are stored in the RNN's internal memory states. In practical terms, for example, the model can estimate the current position of the NiTi actuator given past current and electrical resistance values. Furthermore, the model can also predict future position values based on its historic position profile. RNN-based controls methods have the potential to enable accurate control of SMA actuators, greatly increasing their versatility and applicability.

Chapter 3

Experimental Methods and Conditions

3.1 Materials

The material examined in this study was commercially available (item WSE001550000SG) pseudoelastic 0.0155" (0.394 mm) dia. mechanically polished Nitinol wire obtained from Confluent Medical Technologies in a straight, mechanically polished condition. The composition of the wire was $50.6 - 51wt.\%$ Ni. The as-received base metal wire was cleaned from impurities using ethanol in order to minimize their incorporation during laser processing. The DSC curves of the base metal wire are shown in Figure 3.1. From the results, it can be seen that the M_s and A_f transformation temperatures are $17.4°C$ and $21.1°C$, respectively, meaning that the wire exhibits pseudoelastic properties at room temperature.

Cross-sectional images of the wire were obtained as per the procedure discussed in Section 3.2. The base metal NiTi wire lengthwise cross-section is shown in Figure 3.2. It can be seen that the base metal wires consist of a homogeneous microstructure which appears to lack voids or other imperfections. This image is assumed to be representative of the entire wire length, as no imperfections were observed in any base metal cross-section sample.

Tensile testing was also performed on the base metal NiTi to characterize its stress-strain properties prior to laser processing. As the base metal wire is austenite at room temperature, the stress plateau observed in the stress-strain curve corresponds to the induction of stress-induced martensite. It can be seen in Figure 3.3 that the stress plateau of the as-received base metal wire occurs at $537MPa$.

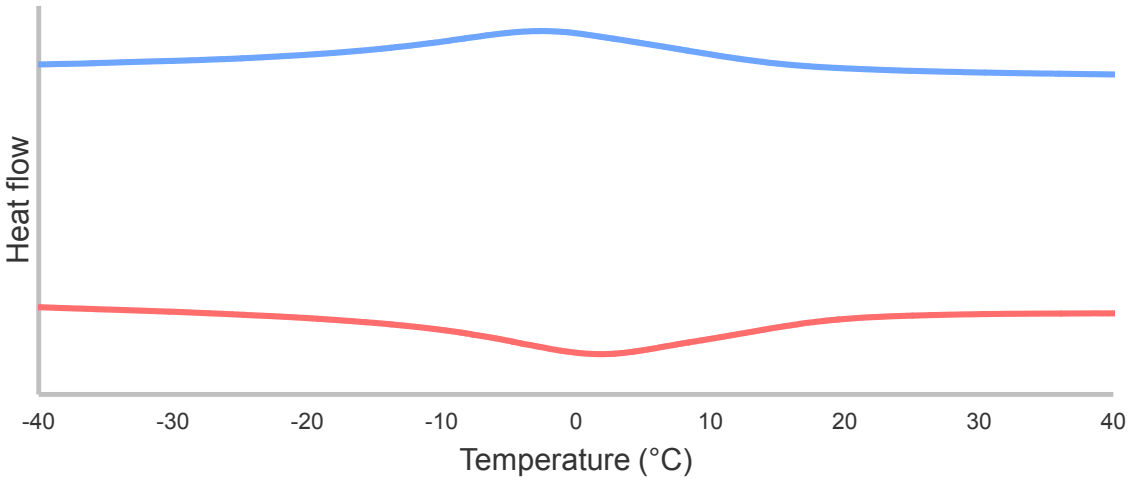

Figure 3.1: DSC plot of base metal NiTi wire used in this study.

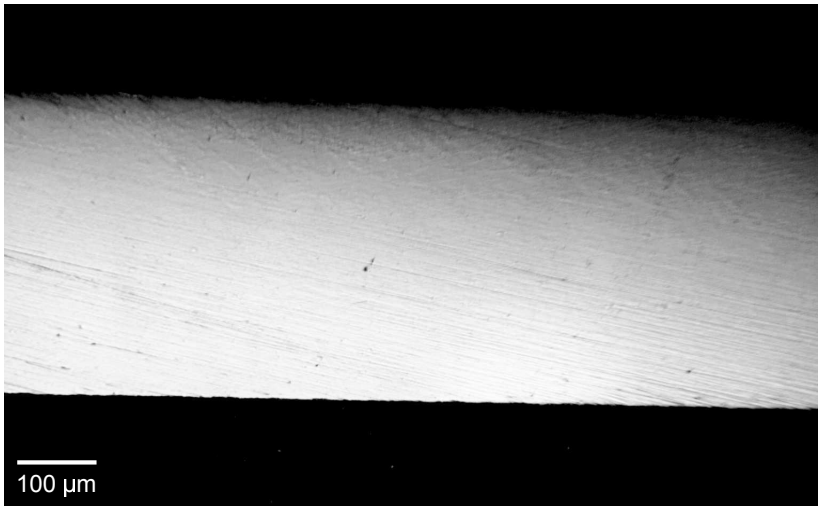

Figure 3.2: Optical microscope image of the base NiTi wire lengthwise cross-section.

3.2 Characterization

As mentioned in Section 2.1.1, DSC was used for thermal characterization of the NiTi wires. A TA Instruments Discovery series DSC was used in this study to determine the transformation temperatures of base, processed, and post-processed NiTi wires by cycling

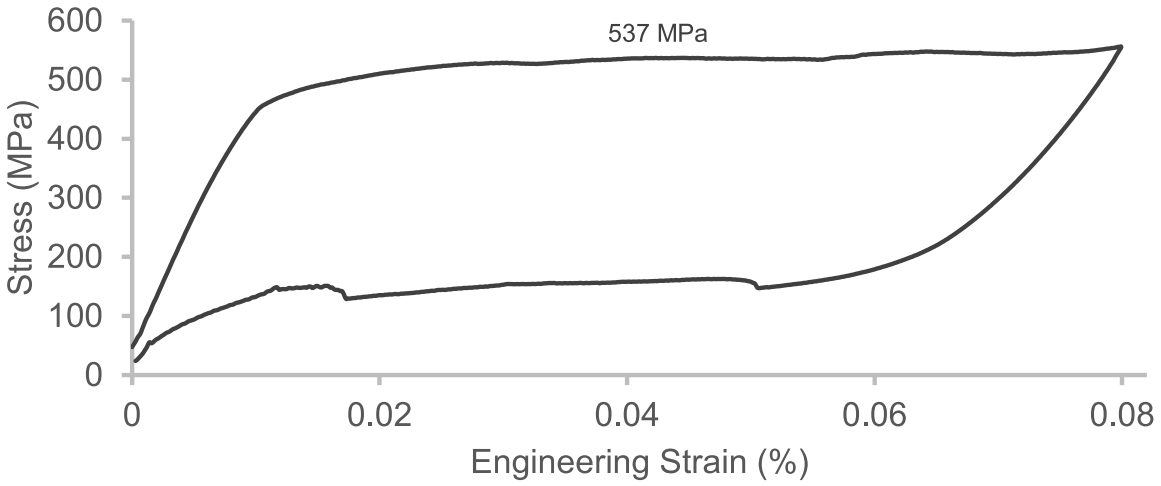

Figure 3.3: Stress-strain curve of base NiTi wire to 8% strain. Stress-induced martensite is observed at stress of $537MPa$.

between $-50°C$ and $120°C$ at a rate of $5°C/min$. The DSC was equipped with a TA Instruments Refrigerated Cooling System capable of reaching temperatures as low as $-75°C$. Ultra high purity (grade 5.0) nitrogen gas was connected to the refrigerated cooling system in order to cool the DSC sample.

Physical characterization of the NiTi wires was performed using an Olympus BX51M optical microscope with up to 40x magnification capability. The microscope was primarily used to acquire images of the lengthwise cross-section of the wire in order to spot imperfections such as voids. Sample wires were cut to size and fastened using a sample holder, which was placed in a sample mold filled with a 7 to 1 weight ratio of EpoFix epoxy resin to hardener (triethylenetetramine). The resin hardened after 24 hours, resulting in a hard, transparent sample. In order to image the cross-section of the wires, the hardened resin sample was sanded and polished using 600, 1200 coarse, and 1200 fine grit sandpaper discs (in that order) on a Struers LaboPol-1 grinding machine until the desired cross-section location was reached. Figure 3.4 shows a polished sample prepared for microscopy imaging.

Finally, the stress-strain characteristics of the NiTi wires were evaluated using an Instron 5565 Advanced Tensile Tester with a $500N$ load cell, as the custom-built tensile testing setups were not able to apply sufficient stress to reach the austenite plateau of the $394\mu m$ wires [86]. The wires were loaded until reaching $7-8\%$ strain, after which the stress was removed and the wire elastically recovered part (or all, in the case of PE) of

Figure 3.4: Optical imaging sample showing five NiTi wires with visible cross-sections ready for optical imaging after mounting and polishing.

its strain. These tests were performed to compare the shape of the stress-strain curves and location of the stress plateaus for the base metal and post-processed wires in order to determine the effect of laser processing on their mechanical properties.

3.3 Laser Processing

The base NiTi wires were processed using an IPG Photonics Ytterbium rack-mount fiber laser with quasi-continuous 3kW peak power, $400\mu m$ fiber diameter, and $1,070nm$ wavelength. The laser was incorporated into a custom system designed for laser processing of wires (shown in Figure 3.5). The system includes 3-axis (XYZ) laser head control with a built-in camera for laser head alignment assistance and a live view of the wire processing. The system is capable of automatic wire feeding, which is useful for processing any desired lengths of wire in addition to remote process parameter programming and operation. Argon gas with flow rate of $5L/min$ was used to shield the NiTi wire with nozzles releasing gas from the bottom and side simultaneously in order to prevent exposure to air during processing, minimizing oxidation. A schematic depicting the main functionality of the wire processing system is shown in Figure 3.6.

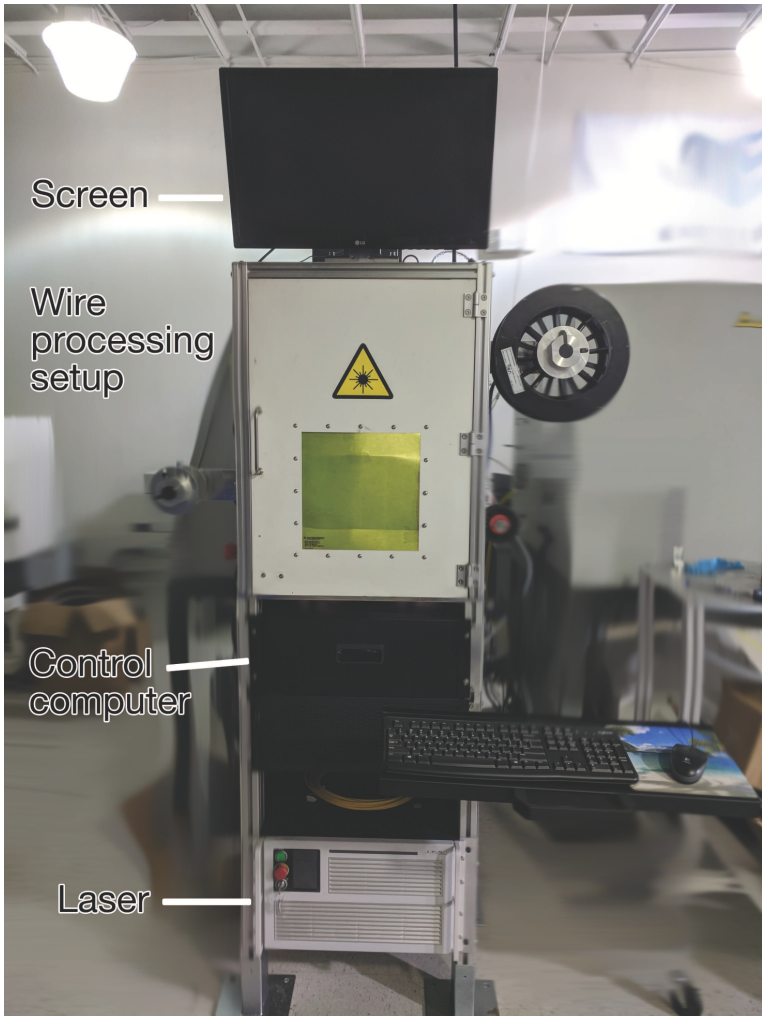

Figure 3.5: System used for producing laser processed NiTi wires.

The final laser parameters used to process the wires were selected on the basis of achieving full laser depth of penetration in addition to significant nickel vaporization, leading to drastic changes in the transformation temperatures. Full penetration of the laser into the wire diameter was required during processing to vaporize nickel from the entire cross-sectional area of the wire. Achieving partial penetration was undesirable as it results in a combination of base and processed NiTi properties. Although full penetration is necessary, it is more difficult to practically achieve compared to partial penetration due to the momentary wire separation caused by the melting of the full wire cross-section. In

order to overcome this issue, springs were used to hold the separated wire pieces in place (as shown in Figure 3.6), eliminating relative motion between the wire sections during nickel expulsion and solidification of the processed region. Incorporating this feature into the laser processing system enabled continuous full penetration processing of NiTi wires.

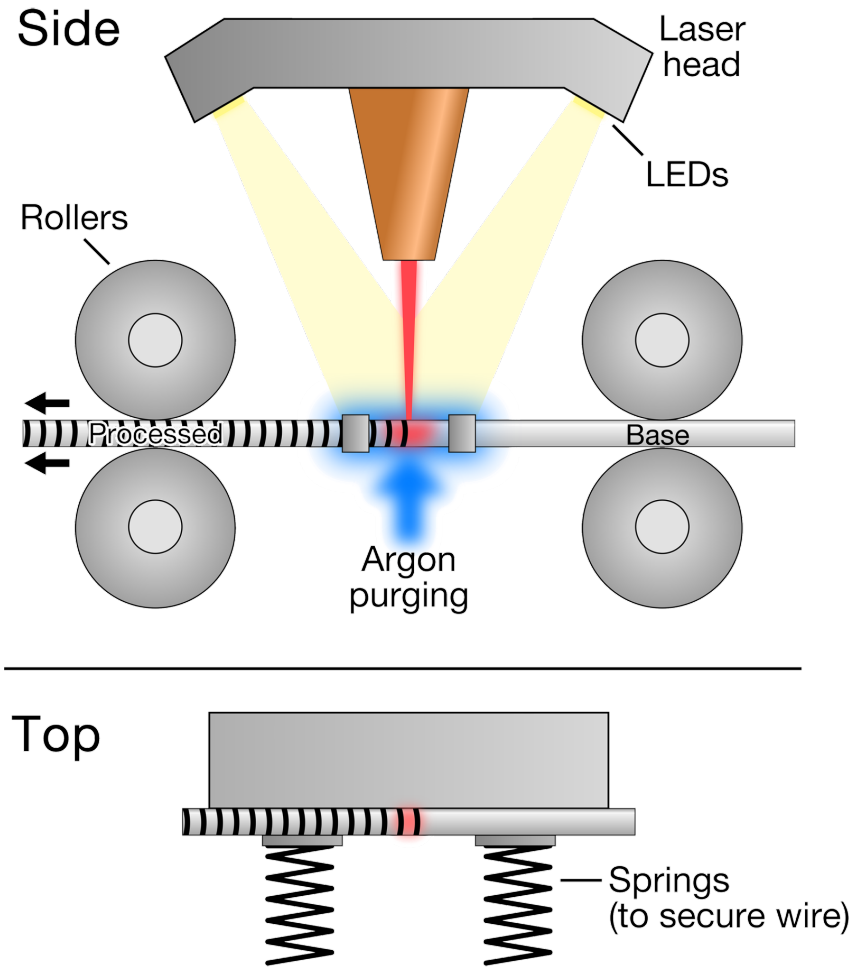

Figure 3.6: Schematic of the system used for laser processing NiTi wires.

The NiTi wires used in this study contained a processed and non-processed region, resulting in a monolithic wire sample with two separate phases (martensite and austenite). Processing part of an austenite wire can locally increase the transformation temperatures so that the processed part of the wire more favourably adopts the martensite phase. This results in a NiTi wire which can contain both martensite and austenite phases at a single

temperature. The goal was to produce monolithic NiTi actuators containing two embedded memories, with one exhibiting SME and the other PE at room temperature. Each wire sample was $7cm$ in length before post-processing, with $5cm$ processed NiTi (high transformation temperatures) and $2cm$ base metal NiTi (low transformation temperatures). This configuration was selected in order to maximize the amount of strain recovery from the SME section while ensuring that sufficient PE section length is present for obtaining a significant drop in electrical resistance.

Due to dissimilar electrical resistivities and transformation properties in the austenite and martensite phases, having a single wire with both phases proves to be useful for controls purposes as separate resistance values can be measured across each phase and used to estimate the stress and position of the NiTi wire. As a result, using electrical resistance measurements across both phases of the NiTi wire, the PE section of the wire acted like a built-in force gauge, obviating the need for external position or force sensors.

3.4 Post-Processing

3.4.1 Wire Drawing

Immediately after laser processing the wires were solutionized by applying $5A$ of current, after which they were left to cool in an ambient environment. According to the phase diagram of Ni and Ti shown in Section 2.1, heating the metal to temperatures above $750°C$ produces an equiatomic NiTi phase. Resistive heating using $5A$ applied current caused yellow/red light to be emitted from the wire, corresponding to temperatures exceeding $750°C$, as traditional solutionization was performed at $750°C$ in a furnace without observing any visible light emission. In addition to growing the grains and lowering the wire drawing forces, solutionization is performed to dissolve intermetallic phases present in the wire and break them down into the fundamental NiTi phase. Eliminating intermetallic phases improves the mechanical properties of the wire, allowing it to withstand the large stresses experienced during wire drawing.

A custom wire drawing setup was built for assisting with the wire drawing process. The wire drawing setup includes a steel spool attached to a large servo motor for pulling the wire through the die, a smaller servo motor for reciprocating the die holder along the length of the spool to wind the wire as it is drawn, and an electric oil dispensing system containing a pump with adjustable flow rate for constant pumping of lubricant to the wire prior to passing through the die. Using this setup, wires of arbitrary length can be continuously

drawn to any desired diameter simply by changing the dies in the die holder. A schematic of the wire drawing setup is shown in Figure 3.7. The dies used were purchased from Advanced Wire Die Limited and were fabricated from polycrystalline diamond in order to minimize die wear and maintain constant wire drawing properties throughout the length of the study.

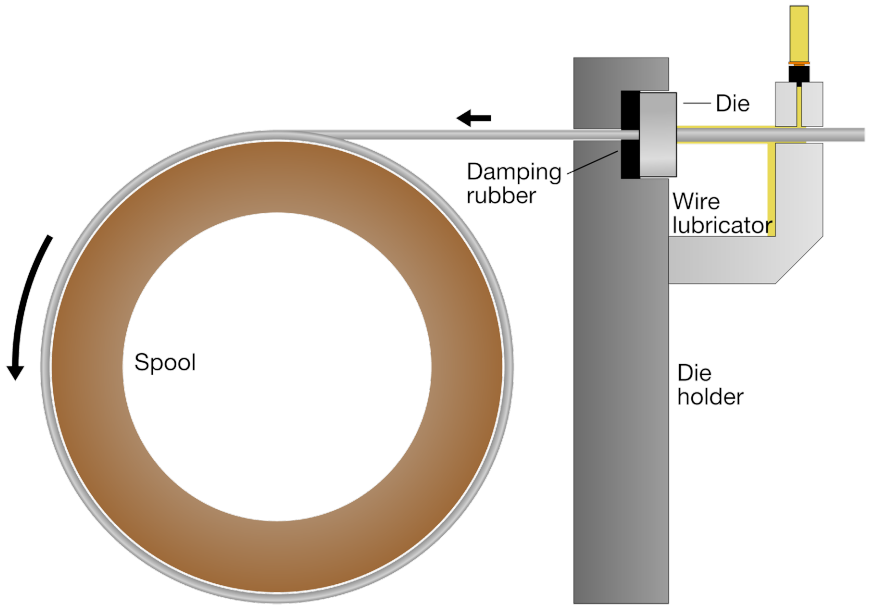

Figure 3.7: Schematic showing the wire drawing setup.

Sharpening of the front end of the wire must be performed in order to feed an initial section through the die and commence drawing. Tip sharpening is traditionally performed by swaging the tip of the wire, resulting in a reduction in cross-sectional area sufficient for entering the die [65]. Since no such equipment was available for use, the tip sharpening was performed by placing the wire in the Instron tensile tester, applying $8A$ of current through the clamped section of the wire, and performing a constant extension of $2mm/min$. As the wire is extended, necking begins to occur at some point along the heated section of the wire, causing a decrease in the cross-sectional area and a direct increase in the local electrical resistance. The larger resistance results in locally increased Joule heating, dramatically increasing the ductility of the wire and causing the extension to create a sharp tip upon breakage. The wire breaking setup is shown in Figure 3.8, with the electrical contacts fastened to the tensile tester clamps rather than the wire itself in order to avoid passing current through the wire at a concentrated contact point.

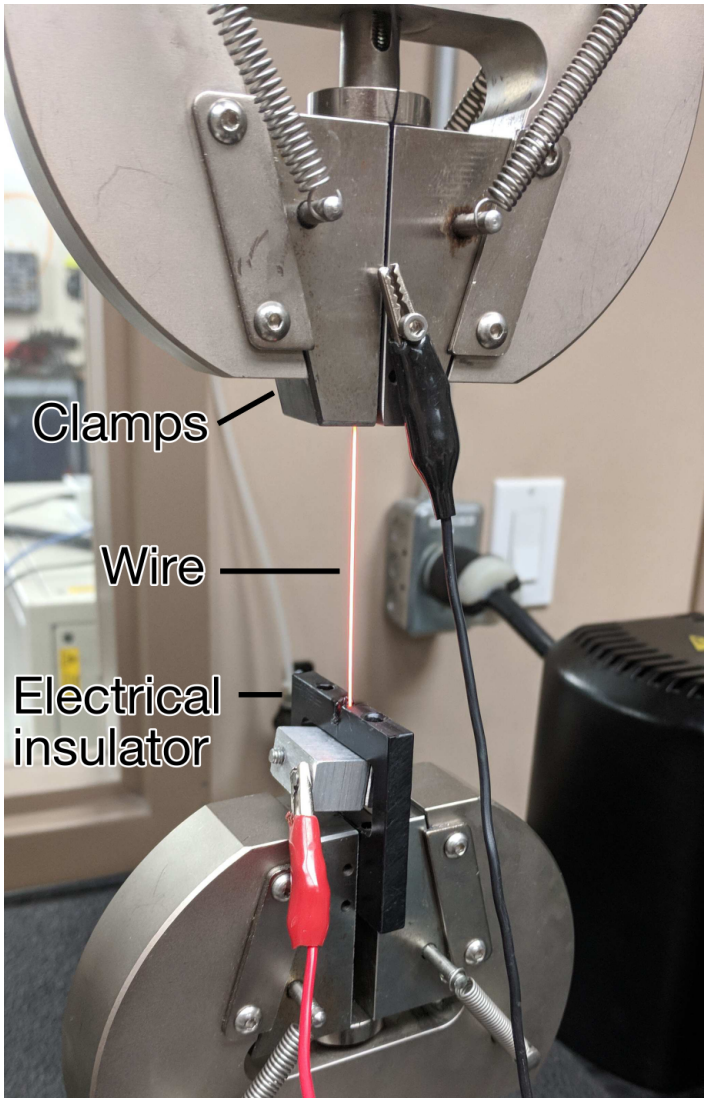

Figure 3.8: Setup used for breaking the wire, creating a sharp tip that can be fed through the wire drawing dies. Current is passing through the wire, causing it to glow due to Joule heating.

After passing the tip of the wire through the die, the initial $4 - 5$ inches of wire are manually pulled through the die using wire drawing tongs. Once a sufficient length of wire has been pulled through the die, the wire is fed through the oil pump channel and clamped to the wire drawing spool. The wire drawing servo motor was set to a constant speed of

$2RPM$. This process was repeated for each die until the desired final wire diameter was reached.

The wires were drawn from $394\mu m$ to the final diameter of $310\mu m$, resulting in a total area reduction of 38.1%. Each wire drawing pass was limited to $10-20\%$ area reduction in order to minimize both the number of passes (time) and the stress experienced by the wire. Machining lubricant was used to minimize the $W_{friction}$ stress component experienced by the wire in order to prevent breakage [87]. The wire drawing sequence for achieving each of the three aforementioned diameters is shown in Table 3.1.

Table 3.1: Die diameter sequence used for wire drawing NiTi samples.

Diameter	Step Area Reduction (%)	Total Area Reduction (%)
$394\mu m$	/	/
$370\mu m$	11.8	11.8
$340\mu m$	15.6	25.5
$310\mu m$	16.9	38.1

3.4.2 Heat Treatment

Numerous studies have been performed to determine heat treatment parameters which result in optimal mechanical properties after wire drawing [88, 89]. For samples cold worked by 30% it was found that heat treatment at temperatures in the range of $400-450°C$ resulted in the largest ultimate tensile stress and lowest stress plateau compared to temperatures exceeding $450°C$ - refer to Figure 3.9 [88, 90]. In addition, heat treatments performed at larger temperatures were found to result in increased transformation temperatures. Using these values as a guideline, the final wire heat treatment was selected based on the trade-off between mechanical properties and characteristic transformation temperatures of the processed, cold-worked wire. The wire was quenched immediately after heat treatment in order to increase surface hardness and minimize phase transformation hysteresis by preventing the formation of Ni_3Ti [91].

3.5 Mechanical Testing

The Instron tensile tester was used to evaluate the mechanical properties of the base, processed, and post-processed wires, including stress-strain curves and tensile failure tests

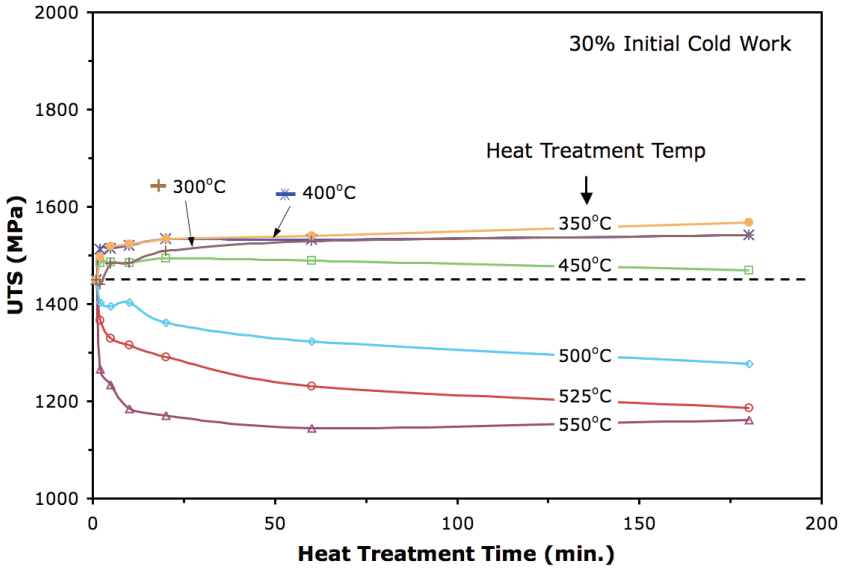

Figure 3.9: The effect of heat treatment at varying temperatures on the UTS of 30% cold worked NiTi wires [92]. The dashed line represents the UTS of the cold worked material prior to heat treatment.

to calculate the ultimate tensile strength (UTS) of the wires. As mentioned previously, the stress-strain curves provide useful information such as phase information (austenite vs. martensite) and identifying the stress necessary for reaching the characteristic stress-strain plateau. Furthermore, the tensile failure tests will gauge the ability of the processed and post-processed wires to endure applied stress in comparison to the base metal wires, determining the degree of mechanical property recovery through post-processing.

The variety and quantity of data required for properly training neural networks exceeds what a traditional tensile tester can acquire within a practical amount of time. In order to circumvent this obstacle, three custom tensile testing setups based on the design proposed by Zamani et al. were redesigned and built for the purpose of data acquisition [100]. Several improvements to the design were made, including incorporation of air bushings to eliminate noise caused by ball bearings, servo motors for programatically applying variable load, and custom control boards for improved controllability and data acquisition performance. Each of the three tensile setups were identical and included the features described in Table 3.2. In addition to these commercially available features, a custom control printed circuit board (PCB) was designed and fabricated specifically for the tensile setups, with the schematic shown in Figure 3.11. The custom PCB enables functionality such as

controlling output current and voltage, sending outputs to the servo motor, reading data from the load cell, position encoder, and temperature sensor, and sensing two separate resistance values. The top wire clamp and load cell were attached to a linear ball bearing shaft with a threaded knob for vertical adjustability. Altogether, the tensile setup enabled all required functionality for acquiring the necessary NiTi wire actuation data. The tensile testers, along with a corresponding schematic depicting their major features, are shown in Figure 3.10. The wires were fastened to the tensile testers using stainless steel clamps to which electrical contacts were also connected. Copper crimps with a through hole were machined and crimped at the PE/SME boundary of the wire to allow for an electrical connection to be made. This electrical contact at the boundary allows for separate resistance values to be measured for the two distinct memories. Plexiglas tubes were also placed around the wire samples in order to isolate them from air streams which may affect its thermal state. Appendix A shows detailed images of various aforementioned tensile tester characteristics.

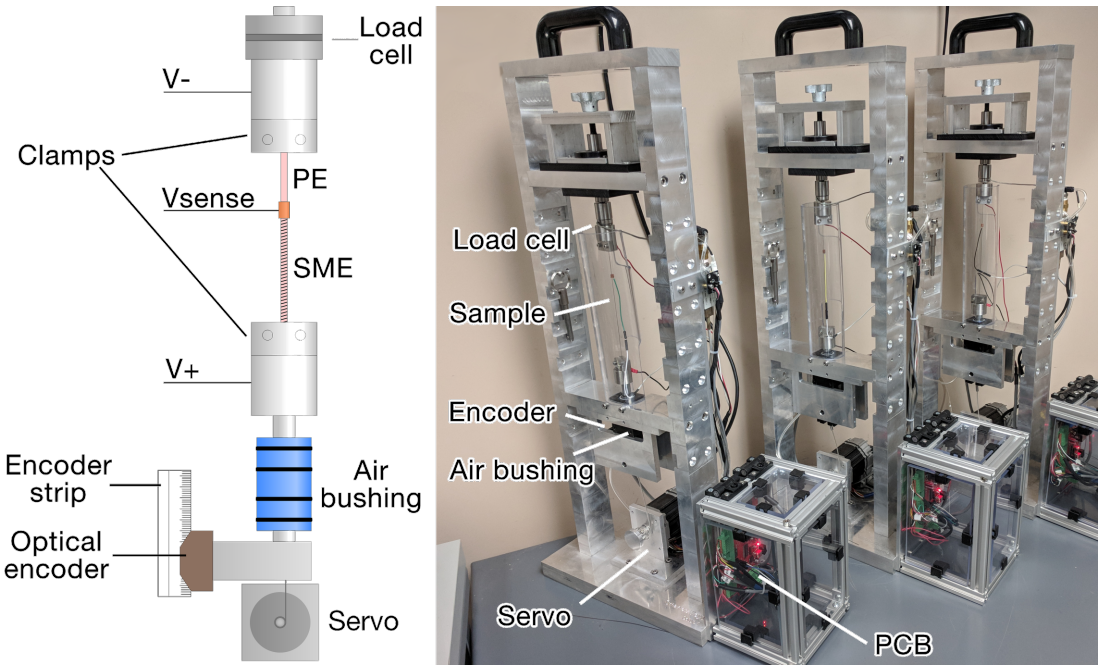

Figure 3.10: Schematic (left) and image (right) of the three custom built tensile testers with samples set up for data acquisition.

The load cell was calibrated using $500 \pm 0.3g$ and $1,000 \pm 0.47g$ stainless steel test weights (meeting Class 7 ASTM and ANSI tolerances) by fitting a linear equation relating

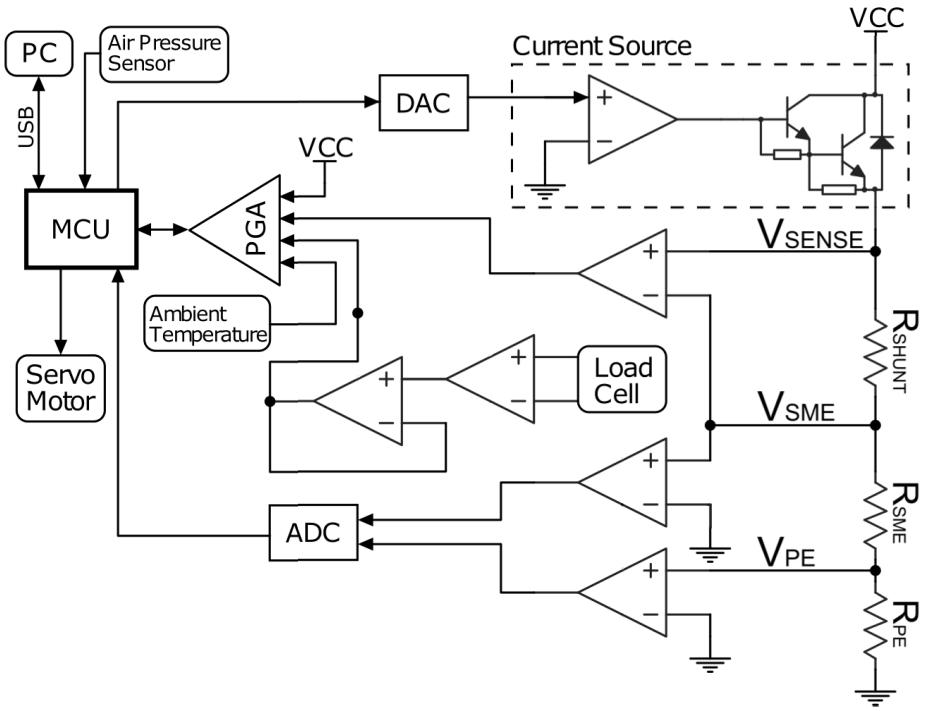

Figure 3.11: Electrical schematic of PCB used in custom tensile testers for actuation and data acquisition.

the load cell's voltage output to the measured weight. The current and voltage outputted by the control board were calibrated through measurement across a 1Ω resistor with 0.1% tolerance using an Agilent 34401A $6\frac{1}{2}$ digit multimeter. The actual resistance (including wires) of the resistor was measured using the four-point resistance measurement technique in order to maximize measurement accuracy. Since the sensed resistance is calculated by taking current and voltage measurements, the sensed current and voltage were both calibrated by measuring the voltage and current across the PCB terminals using the aforementioned DMM and fitting a corresponding linear function relating the measured current and voltage to the PCB's sensed current and voltage, respectively. The servo motor was set to constant torque mode, controlled by the PCB using pulse width modulation (PWM) with maximum applied load limited to $20N$ in order to avoid accidental overloading and possible breaking of the NiTi wire samples. The servo motor was also tuned to account for any mechanical attachments placed on the shaft in order to ensure correct load application on the wire.

Table 3.2: Features included in the custom-built tensile testing setups.

Feature	Item #	Description
Load Cell	LCMFD-500N	$\pm 500N$ range with $\pm 0.15\%$ combined linearity and hysteresis accuracy
Air Bushing	S301301	Eliminate noise caused by ball bearings
Linear Position Encoder	EM2-0-2000-N	Achievable $3.175\mu m$ resolution using 2000 lines/inch strip and x4 quadrature mode
Temperature Sensor		Calibrated to $\pm 0.1°C$
Servo Motor	CPM-MCPV-2310S-RLN	1.6 $N \cdot m$ peak torque with 0.375" shaft diameter. Applies force to wire.
Air Pressure Sensor		Ensures air is flowing into bushing.

3.5.1 Training

Each processed NiTi wire was thermally cycled with an applied load to ensure consistent behaviour during data acquisition. This process is useful for aligning the grains in the desired direction of actuation, resulting in repeatable actuation behaviour [93, 94]. This behaviour can be observed in Figure 3.12, which shows 30 cycles of mechanical training performed on a base metal NiTi wire using the Instron tensile tester. It can be seen that the stress-induced martensite plateau initially occurs at $560MPa$, with the plateau stress decreasing to a value of $346MPa$ as the wire is trained further. Another effect resulting from training is the presence of residual strain (around 1% in Figure 3.12) which cannot be recovered by the wire's pseudoelastic properties.

Training was performed on the three custom-built tensile testers using the custom developed training control program shown in Figure 3.13 [16]. The applied load and current along with stop conditions are specified in the program, allowing the user to adjust the training procedure accordingly. The wire training parameters and stop criteria used in this study are shown in Table 3.3. Wire training was stopped when the wire position threshold (δ_{pos}) between cycles was reached during both heating and cooling for a consecutive number of cycles.

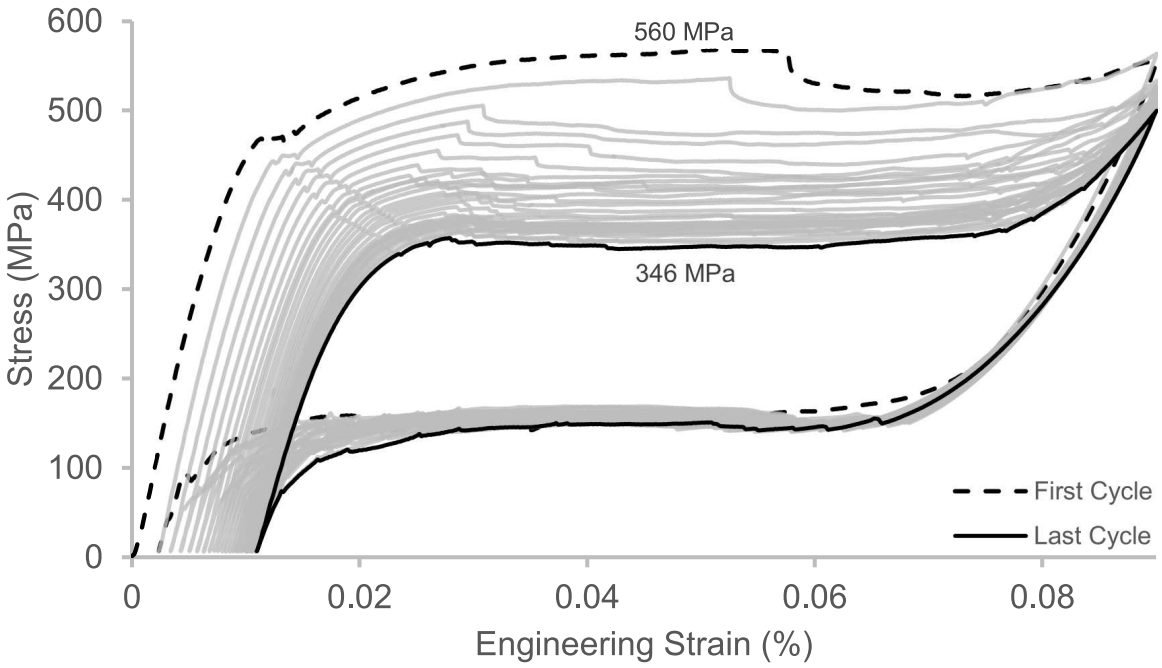

Figure 3.12: Mechanical cycling performed on base NiTi wire for 30 cycles.

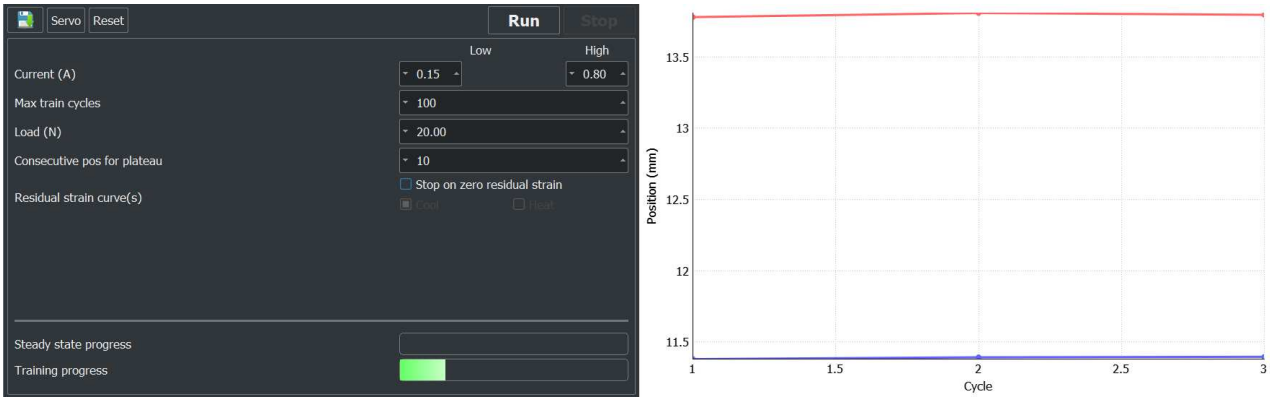

Figure 3.13: Program used to perform thermal cycling of NiTi wires using the custom-built tensile setups.

Table 3.3: Parameters used during wire training.

Parameter		Value
Load	(N)	20
Current - High	(A)	0.8
Current - Low	(A)	0.15
Position threshold δ_{pos}	(mm)	0.1

3.5.2 Data Acquisition

A custom software controller was developed specifically for data acquisition and actuation of NiTi wires. In order for the neural network controller to learn the hysteretic behaviour of NiTi wires with varying load and heating (including major and minor hysteresis loops), data was acquired over a range of applied forces and currents. The data acquisition program developed for this study accepts a number of inputs including minimum and maximum current, maximum load (with minimum being zero load), number of applied loads, and number of applied currents. The parameters used for data acquisition in this study are shown in Table 3.4.

By specifying the maximum load and number of loads to be tested, the software evenly interpolates between the minimum and maximum load values in order to generate the applied loads. However, the same is not true for current - instead, in order to randomize the testing procedure experienced by each NiTi wire, a mean and standard deviation are specified for the heating and cooling values. The software then randomly generates the desired number of data points following a normal distribution based on the corresponding mean and standard deviation values. To add to the randomization, the order of the interpolated load values is also randomized during data generation. The randomization of the actuation schedule experienced by each wire is important, as the data will be used to train neural networks. Training a neural network with data which has inherent patterns (such as steadily increasing load or predictable applied currents) may result in the neural network detecting these patterns and treating them as a characteristic of the inherent data set. This is undesirable, as the neural network is meant to be a universal model capable of achieving accurate NiTi wire control which follows any desired actuation path. Randomization greatly helps prevent overfitting and achieve controller universality [95].

The controller was run after data generation, applying the first load to the wire. The servo is placed in constant torque mode and ensure that the desired force is constantly applied to the wire. A different list of currents is generated for each load, and the load is kept constant until the software has applied each of the currents. After each load or current

change, the software ensures that the wire position reaches steady-state before moving on to the next current or load. Steady state position is reached by remaining within a defined position range for a certain number of time steps. In addition, when a new load is applied, the current is initially set to the minimum value to act as a reference between varying loads.

The controller takes various input values (such as applied current ranges, loads, etc.) and displays the generated current and load cycling recipe as shown in Figure 3.14. The generated currents are graphed in the top plot, with a green bar representing the overall controller progress. Similarly, the loads are shown in the bottom plot as a bar graph with a green overlaid bar representing the current applied load.

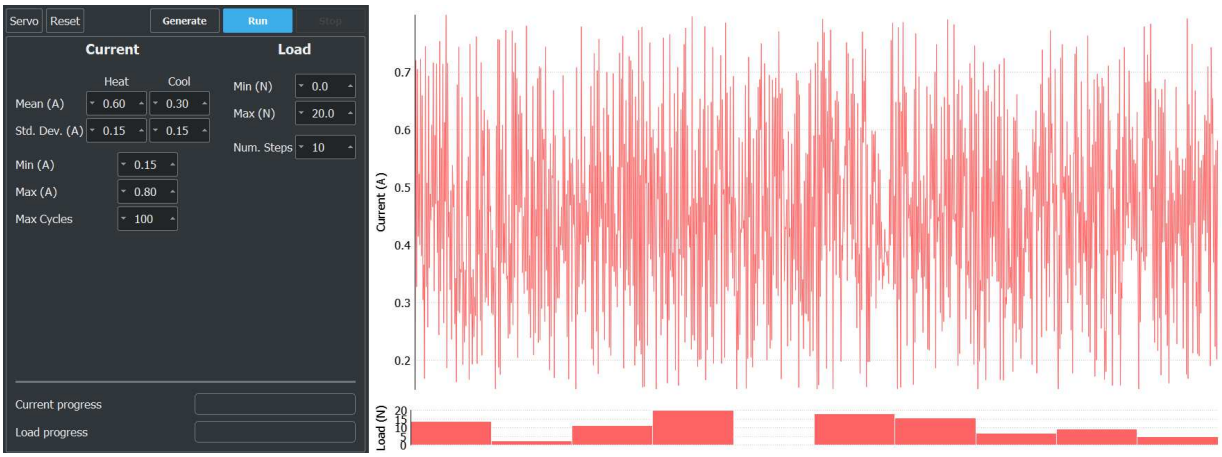

Figure 3.14: Program used to generate random current and load values for data acquisition using the custom-built tensile testers.

Table 3.4: Parameters used for data acquisition of NiTi actuation.

Parameter		Value
Maximum load	(N)	20
Minimum current	(A)	0.15
Maximum current	(A)	0.8
Mean current - heating	(A)	0.6
Standard dev. current - heating	(A)	0.15
Mean current - cooling	(A)	0.3
Standard dev. current - cooling	(A)	0.15

3.6 Neural Networks

Python 3.5.4 was the language of choice for programming the neural network-based model. The development of various types of neural networks was simplified through the use of the Python machine learning library TensorFlow 1.1.0 with graphics processing unit (GPU) support, which provides native implementations of multiple neural network architectures including traditional deep neural networks, recurrent neural networks (with regular, LSTM, and GRU cells), and convolutional neural networks [96]. In order to accelerate the training of neural networks, an NVIDIA GeForce GTX 1060 GPU was used along with NVIDIA's parallel computing platform CUDA 9.0 and the corresponding deep neural network library cuDNN 7.1. Since the training of neural networks consists of massive numbers of simple mathematical evaluations (such as addition and multiplication), GPUs have been shown to increase training speed by an order of 10 compared to performing sequential mathematical evaluations using the computer's central processing unit (CPU) [97]. The reason for this performance discrepancy is that GPUs can contain thousands of cores with small amounts of processing power sufficient for parallel computation of the simple mathematical expressions present in neural network training. In contrast, CPUs generally contain 4-8 cores, significantly decreasing the number of operations which can be performed in parallel. The GeForce GTX GPU contains 1152 CUDA cores compared to the 4 cores or (8 threads) found in the Intel i7-6700 3.40GHz CPU present in the same computer.

The two RNN architectures tested were LSTM and GRU. Due to the large sequences of past data required for describing the hysteresis of NiTi, regular RNNs were not examined as they would likely suffer from vanishing/exploding gradients. The accuracy of the neural network predictions was evaluated by calculating the mean of the sum of squared errors, also known as the mean squared error (MSE). The sum of squared errors calculation is shown in Equation 3.1, where $R(\theta)$ represents the measure of fit for the weights, K is the number of output neurons, M is the number of vectors of unknown parameters (weights), $f_k(x_i)$ is the output predicted by output neuron k, and y_{ik} is the desired output value [98]. The Adam optimizer was used for stochastic optimization of the neural network weights and biases, as this method has been found to be more effective than other popular techniques such as stochastic gradient descent and AdaGrad [79]. A constant learning rate of 0.01 was used for all neural network training. The data set used for training was normalized to the range $[0, 1]$ in order to improve the neural network performance [99]. In addition, the data point batches (chunks of past values) were fed into the neural network in random order during training in order to prevent the neural networks from learning any patterns originating from the data acquisition process which may not be present during actuation. The training process was repeated for a predetermined number of training epoch, with

each epoch representing a full training cycle using the entire training data subset.

$$R(\theta) = \sum_{k=1}^{K} \sum_{i=1}^{N} (y_{ik} - f_k(x_i))^2 \tag{3.1}$$

The acquired data was split into training (70%), validation (10%), and testing (20%) subsets for the purpose of evaluating the models' performance on data it has previously not encountered. The model is expected to perform well on data it has previously seen, but how well it generalized to data never before encountered was evaluated using the validation and testing data subsets. The performance of the model on the validation data set is evaluated at the end of each training epoch, and on the testing data set after training has fully concluded. The data splitting was performed individually for each applied load to ensure that a sufficient number of data points were used for training at each load. The data sets were split randomly, and each resulting subset was randomized so that training is performed in random order. Each position and force data point fed into the RNN was accompanied by a number of past measured current and resistance values equal to the look back length of the current experiment.

Due to the sheer amount of data used in the study (over 1 million data points for each measured parameter), training the neural networks required a significant amount of time. As a result, the neural network optimization described in Section 5.1 was performed under constant load of $10N$, which significantly speeds up the training time by using less data. All parameter variations performed during optimization are summarized in Table 3.5 along with the base parameters kept constant across all tests. Section 5.1.7 describes position estimation performed using constant load, evaluating the best performing models from the hyperparameter optimization on the entire data set. Finally, position and load estimation was carried out under varying load using the best performing RNN, with the results detailed in Section 5.2. As the resistances across both actuator phases were used as inputs in the variable load model evaluation, the RNN architecture was adjusted to have 3 inputs (measured R_{SME}, R_{PE}, and current) as well as two outputs (predicted position and load).

The activation function and number of hidden units are parameters which directly affect the structure of the RNN. Details regarding the activation functions used can be found in the TensorFlow library documentation [96]. The number of epochs represents the number of times the network is trained on the entire training data set. Look back length refers to the number of past time steps fed into the RNN when the force and position at the current time step - the larger the number, the more resource-intensive the position estimation. Training is performed in batches with multiple data points fed at the same

Table 3.5: Parameters used for optimization of the position estimation RNN.

Parameter	Base	Variations				
Activation Function	ReLU	tanh	sigmoid	softplus	ReLU6	elu
Batch Size	256	128	512			
# Hidden RNN States	100	50	125			
# Epochs	20	1	5	10	30	
Look back length	1500	500	2000			
Sparsity	3	5	10			

time, and so the batch size explores the effect of different batch sizes on the RNN training performance. Finally, the sparsity represents skipping data points - a sparsity of 2 leads to the use of one in every two points, effectively reducing the number of past data points used for estimation in half. Because data acquisition was performed at 50 data points per second, the acquisition speed may be too high relative to the speed of the actuator, and therefore skipping points may result in comparable training performance while immensely increasing training speed. The RNN with optimized parameters was then used in Section 5.2 to perform position and force estimation using two measured resistances.

Chapter 4

Thermomechanical Properties of Laser-Processed NiTi

Before performing data acquisition and developing control algorithms, a thorough investigation of the thermal and mechanical properties of laser processed NiTi with two embedded memories was performed. This chapter explores the effects of varying laser processing and thermomechanical post-processing parameters on the final NiTi wire characteristics. The choice of laser processing parameters had a significant effect on both the characteristic transformation properties and the surface morphology of the processed wires. Optimization of laser processing parameters led to increased transformation temperatures without the creation of Ti-rich intermetallics as well as a smooth morphology, both of which are important for successful wire drawing of the processed wire. Tuning the thermomechanical post-processing treatment also proved to be key for recovering mechanical performance lost during laser processing.

4.1 Metallurgical Properties

Compared to the smooth base metal morphology, NiTi laser processing produces a rough surface morphology such as the one shown in Figure 4.1. The rough surface morphology of the laser processed wire stems from the turbulent nature of the Ni vaporization expulsion. The surface roughness was found to be controllable through laser processing parameters to an extent, with an increase in parameters such as laser power and pulse frequency resulting in significantly more textured morphology. As a result, the laser properties were selected

with the goal of minimizing the surface roughness present at the laser processed region in order to reduce the stress experienced by the wire during wire drawing.

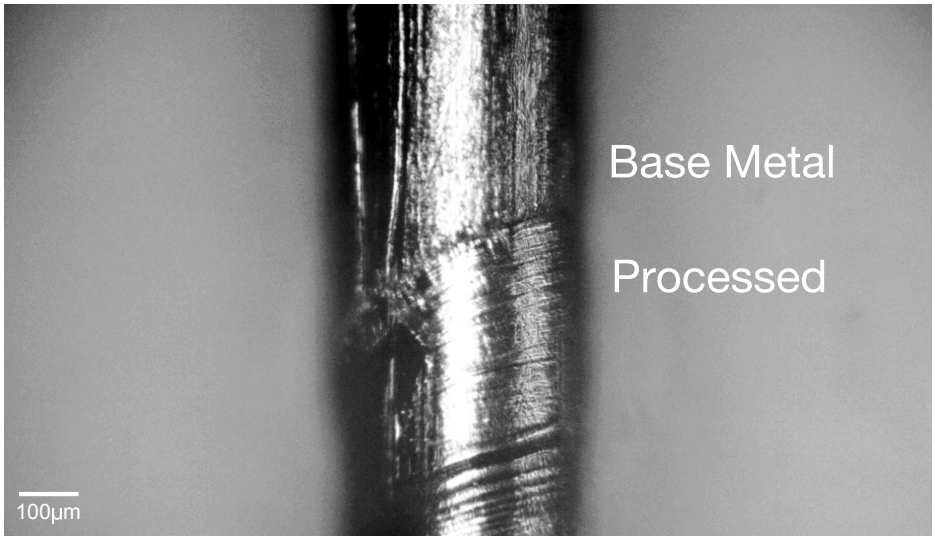

Figure 4.1: Boundary of base metal and processed NiTi wire showing difference in surface morphology.

The optical images of the lengthwise cross-section of the as-processed and wire drawn wires are shown in Figure 4.2. The as-processed wire was observed to have various visible defects throughout its cross-section, including what appear to be voids. In contrast, the microstructure of the post-processed wire appears to be significantly more homogeneous, with slight imperfections still present sparsely throughout the wire. Nevertheless, there is a discernible difference between the as-processed and post-processed wires likely due to wire drawing breaking down the microstructure and filling in most voids or other imperfections resulting from laser processing. The microstructural defects translate into poor mechanical properties for the as-processed wire as discussed in Section 4.3, with post-processing leading to considerable recovery of mechanical properties through elimination of such defects.

4.2 Thermal Properties

A comprehensive DSC study was performed to determine and confirm the effects of various laser processing parameters on the transformation characteristics of the laser processed NiTi wires. For reference, the transformation temperatures for each of the DSC studies

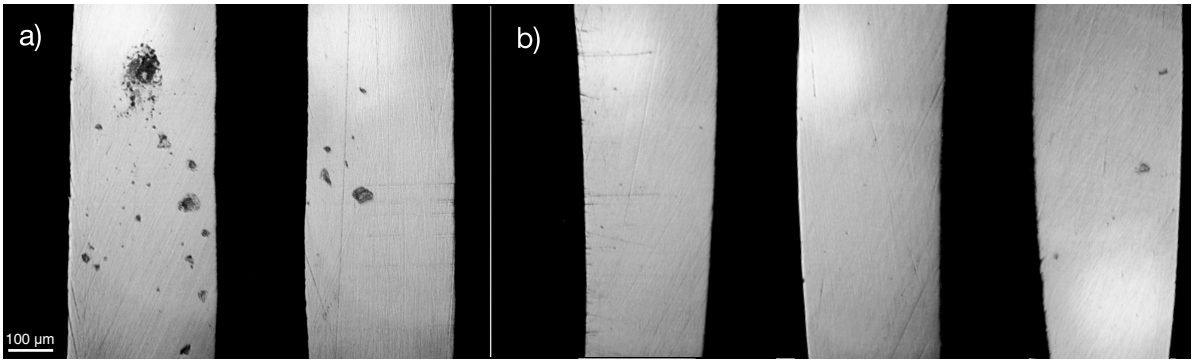

Figure 4.2: Cross-sectional images of a) as-processed and b) wire drawn wires.

performed in this section are listed in Table 4.2. Figure 4.3 shows the effect of laser power on the transformation properties, and a clear rise in transformation temperatures is observed as laser power is increased from $370W$ to $410W$ with other parameters held constant. These results are in accordance with those reported by Zamani et al [100]. Furthermore, the high temperature section of the heating curve appears somewhat sharper for the wire processed with $410W$, hinting that the transformation temperature increase caused by Ni vaporization may be beginning to reach saturation. Further increase in processing power may yield negligible changes in transformation temperature while further reducing the Ni composition, causing the formation of undesirable Ti intermetallics. As a result, laser processing power of $410W$ was selected for the study.

The effect of pulse duration on the transformation temperatures of processed NiTi wire is examined in Figure 4.4 using power of $390W$. As reported by Khan et al, an increase in the laser processing pulse duration causes the transformation temperatures to rise, eventually plateauing as the material becomes Ti-rich [10]. The largest rise in transformation temperatures is seen in the increase from $5ms$ to $6ms$, with further increases in pulse duration causing marginal transformation changes. Given the selected laser power of $410W$, a pulse duration of $5ms$ was found to result in maximized transformation temperatures while remaining on the verge of becoming Ti-rich and was therefore selected for the study. The remaining laser processing parameters used for fabricating the samples examined in the following sections are outlined in Table 4.1.

Solutionization was performed immediately after laser processing as per the procedure discussed in Section 3.4. It was found that the number of solutionization cycles had no significant effect on the transformation properties of the wire, meaning that the majority of grain growth and intermetallic dissolution occurred during the first solutionization cycle.

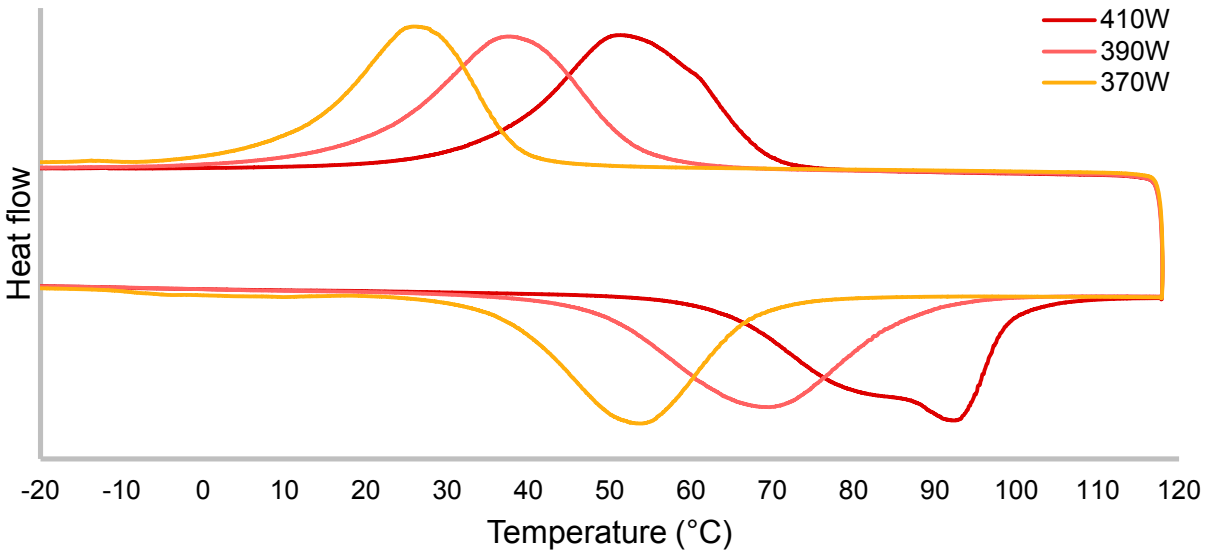

Figure 4.3: DSC plots of NiTi wires as-processed with laser powers of 370W, 390W, and 410W.

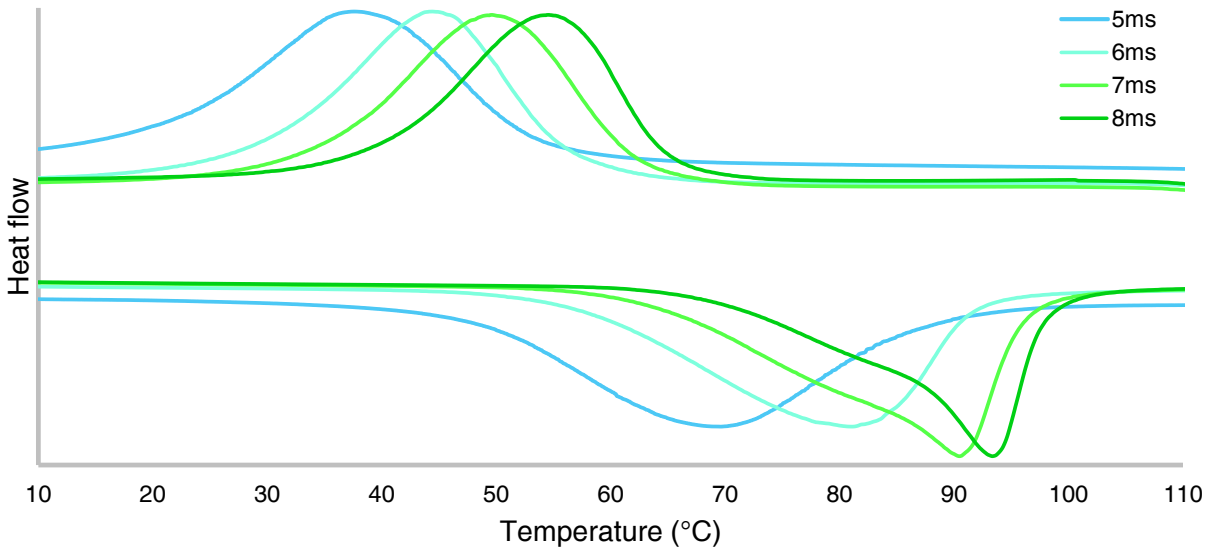

Figure 4.4: DSC plots of NiTi wires as-processed with laser pulse duration of 5, 6, 7, and 8 ms.

Table 4.1: Parameter used for laser processing of NiTi wires in order to achieve full penetration and Ni vaporization.

Parameter		**Value**
Power	(W)	410
Overlap	$(\%)$	85
Spot size	(μm)	1000
Pulse time	(μs)	5000
Pulse frequency	(Hz)	1
Pulses per spot		1

However, each solutionization cycle performed in an ambient environment contributed to the thickness of the oxide layer grown on the NiTi wires. As a result, increasing the number of solutionization cycles led to a direct increase in the oxide layer thickness. It was found that performing 10 solutionization cycles with $5A$ of current severely deteriorated the mechanical properties of the NiTi wire compared to 1-4 cycles, causing the wire to fracture when experiencing relatively small bend radii. This result may be attributed to the significant growth of the brittle oxide layer, consuming the malleable NiTi metal.

The heat treatment performed on the cold worked wires was also found to have a significant effect on the transformation properties of both the base metal and processed regions of the NiTi wire. As is visible in Figure 4.5, increasing the heat treatment temperature causes a rise in transformation temperatures for the laser processed NiTi wire. Although heat treatment at $500°C$ yielded transformation property improvements compared to $450°C$, the final wires broke during training. Interestingly, heat treatment at $480°C$ yielded larger A_f than at $500°C$.

The effect of heat treatment on the unprocessed base metal section of the wire was also studied, as heat treatment of cold worked Ni-rich NiTi has been shown to significantly increase the material's transformation temperatures through the formation of nickel precipitates [92]. The DSC plots of base metal post-processed wires heat treated for 2 hours at temperatures of $480°C$ and $500°C$ are shown in Figure 4.6. Compared to the as-received base metal NiTi, the post-processed base metal sections saw a significant increase in transformation temperatures. However, heat treatment at $500°C$ yields lower A_f than at $480°C$. This effect is due to the presence of R-phase - increasing heat treatment temperature appears to cause the R/martensite peak to increase while the R/austenite peak simultaneously decreases. At $500°C$, the R/martensite and R/austenite peaks appear to have combined, resulting in a taller, narrower peak than those acquired from heat treatment at $480°C$. The transformation temperatures for the base metal heat treated at $480°C$ are around room

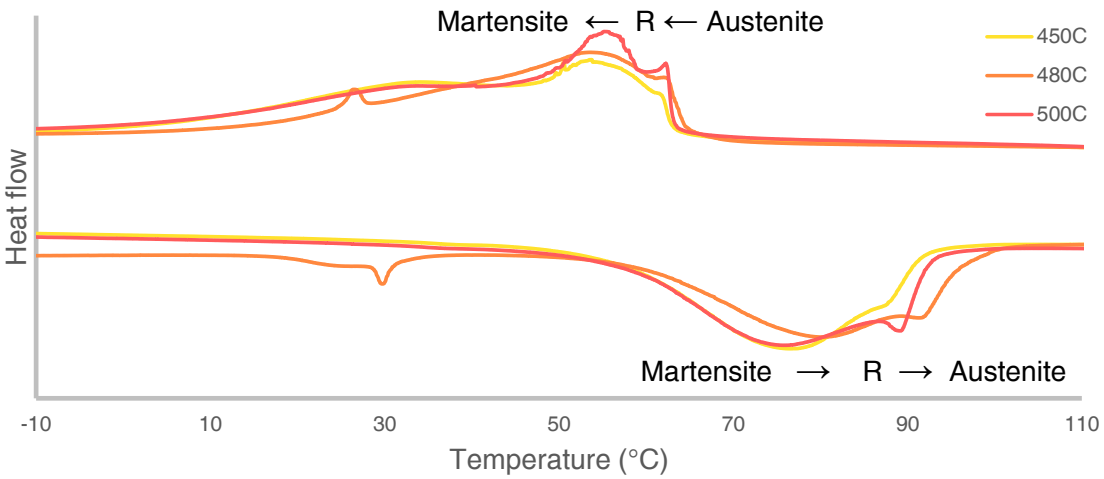

Figure 4.5: DSC plots of processed NiTi wires cold worked to 310μm and heat treated at various temperatures for 2 hours.

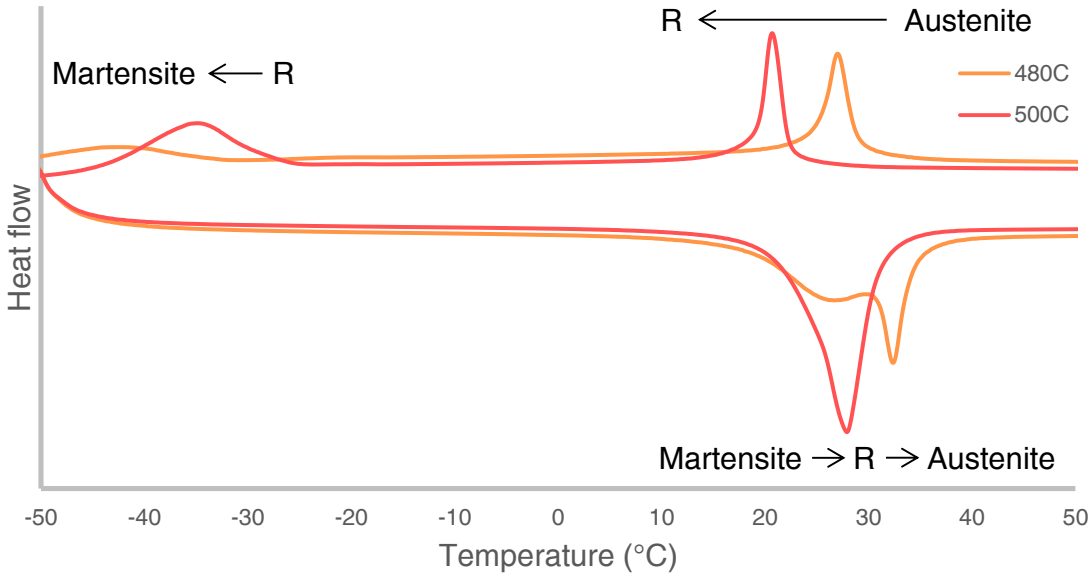

Figure 4.6: DSC plots of base metal NiTi wires cold worked to 310μm and heat treated at various temperatures for 2 hours.

temperature, which yields partial transformation and therefore mixed PE/SME behaviour rather than pure PE. However, training and applied stress should both cause a significant enough decrease in the transformation temperatures to differentiate the distinct electrical and mechanical properties of the processed and unprocessed memories.

Table 4.2: DSC temperatures for all NiTi wire processing and post-processing studies discussed in this section.

Study		Value	A_s	A_f	M_s	M_f
Pulse Power	(W)					
		370	35.2	68.4	38.8	9.79
		390	48.0	89.0	54.1	19.2
		410	63.1	98.7	69.4	34.7
Time	(ms)					
		5	48.5	87.4	53.7	20.1
		6	65.2	92.1	58.3	25.9
		7	63.0	96.7	63.6	33.8
		8	67.7	98.3	65.8	38.5
Heat Treatment Temperature	$(^\circ C)$					
		450	51.8	92.3	63.1	3.7
		480	59.2	97.1	64.5	23.3
		500	66.5	100.3	65.8	40.8
Heat Treatment Temperature, Base Metal	$(^\circ C)$					
		480	17.3	35.6	29.2	23.7
		500	17.3	32.4	22.5	18.4

The process discussed in this section produced monolithic wires containing two distinct sets of transformation temperatures, with one section exhibiting full SME and the other mainly PE at room temperature. DSC curves corresponding to the final processing and post-processing parameters for the base metal, as-processed, solutionized, and heat treated NiTi wires are shown in Figure 4.7. DSC characterization on the final wire was also performed after training, with the results shown in Figure 4.8. It is apparent that training has led to the sharpening of the transformation peaks, along with translating toward lower transformation temperatures.

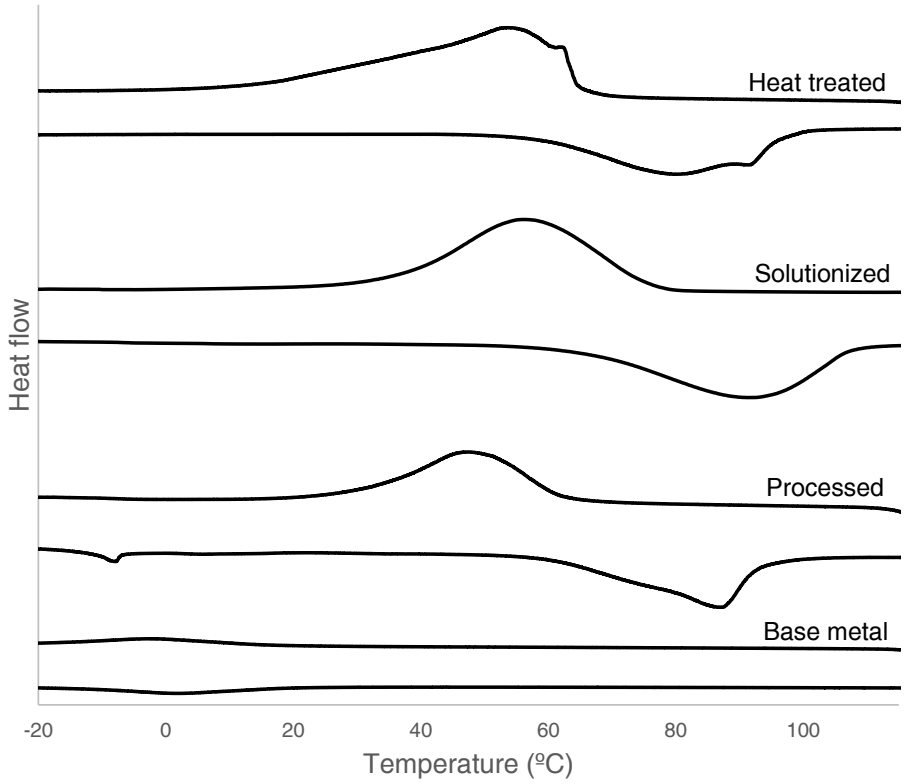

Figure 4.7: DSC plot of base metal, laser processed, solutionized, and heat treated NiTi wires corresponding to the final properties used to fabricate wires for data acquisition.

4.3 Mechanical Properties

The forces required to pull the processed wire through each die during wire drawing are displayed in Figure 4.9. The drawing stresses experienced during the first wire drawing pass remained below the recommended upper stress limit of 0.6 times the UTS of the wire (upper limit of $351MPa$ for the as-processed wire) [101]. As the area reduction increased with each step, the wire experienced increasing stresses. Furthermore, due to the textured morphology of the processed area, the wire drawing forces fluctuated significantly more at the processed region compared to the base metal region. However, the textured morphology slowly disappeared with each wire drawing pass, which is evident from the drop in drawing stress fluctuations. The rough laser processing morphology was fully eliminated in the final wire drawn product, resulting in a wire with smooth morphology.

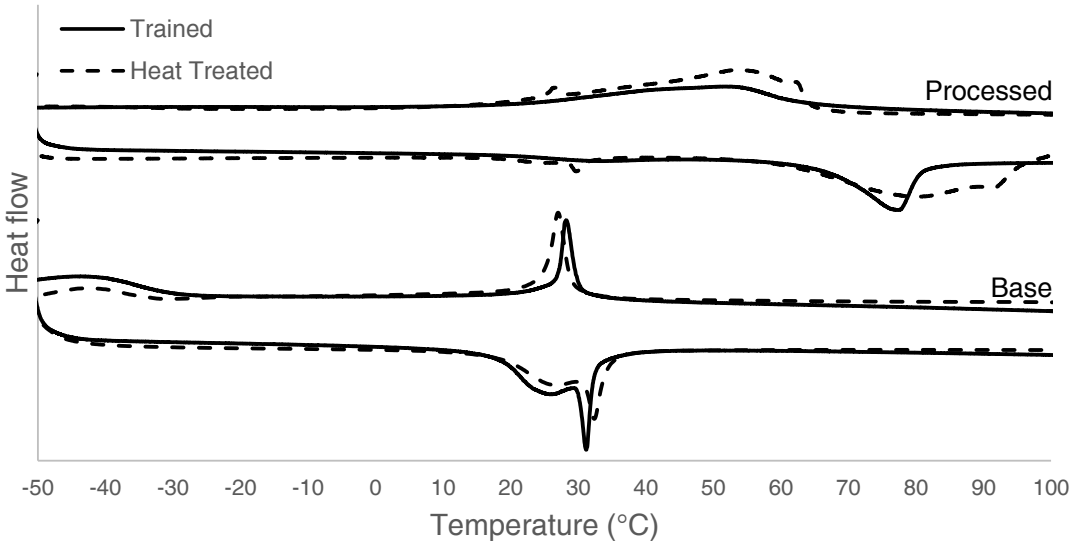

Figure 4.8: DSC plots of base metal and processed NiTi wires after heat treating and training.

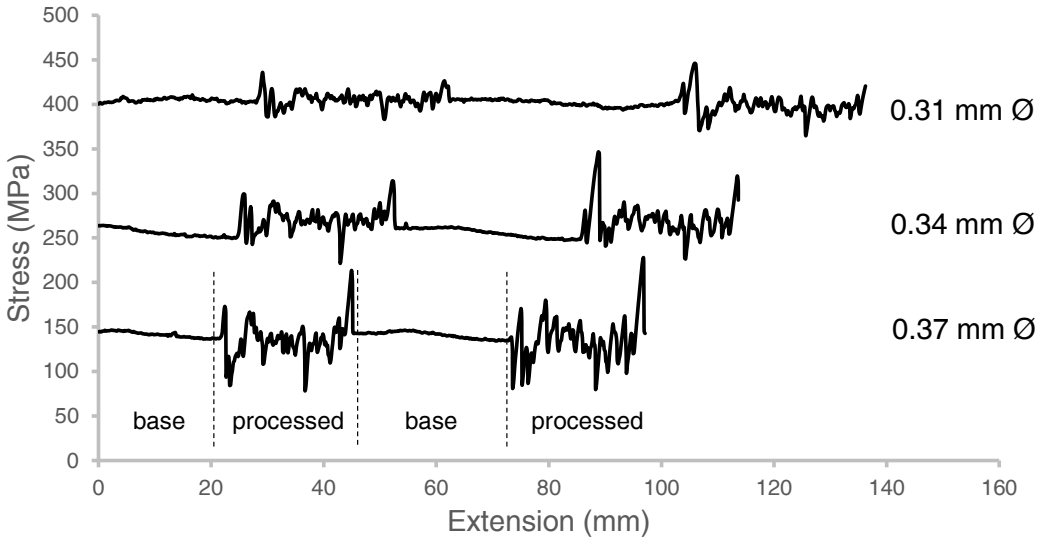

Figure 4.9: Wire drawing forces of base metal and laser processed sections of a monolithic NiTi wire sample through die diameters of $0.37mm$, $0.34mm$, and $0.31mm$.

The tensile tests performed on the heat treated wires are shown in Figure 4.10. Unlike the base metal NiTi, straining the fully laser processed and post-processed wires did not result in a significant amount of strain recovery due to the wires exhibiting SME at room temperature. Accordingly, the tensile results prove that the laser processing and post-processing successfully altered the room temperature microstructure of the NiTi wires from austenite to martensite. The detwinning plateau stress also appears to depend on the heat treatment temperature, with increasing temperature resulting in a lower stress plateau. This effect is visible in Figure 4.10, with the stress plateau dropping from an average of $202.0MPa$ to $183.9MPa$ as heat treatment temperature increases from $450°C$ to $500°C$, respectively.

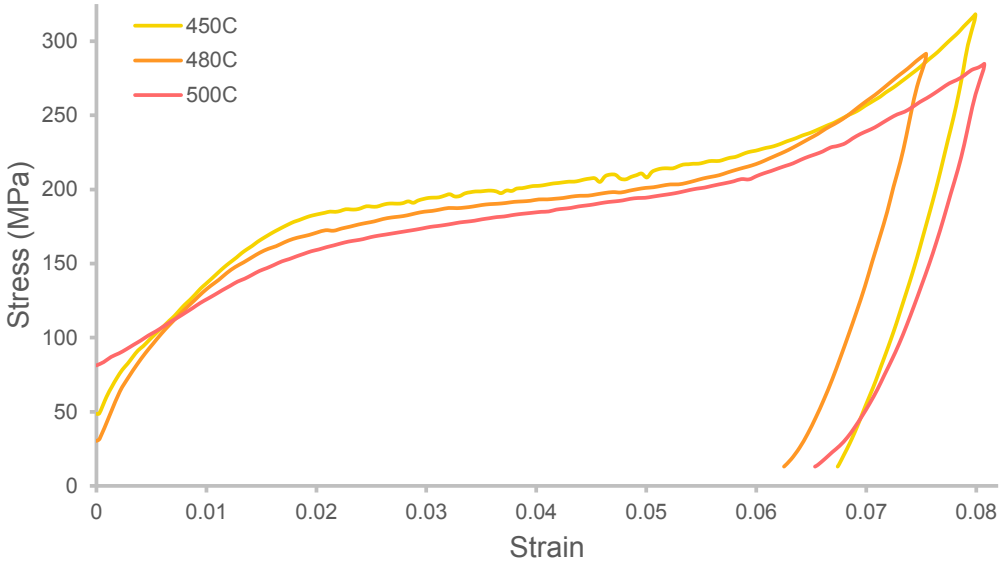

Figure 4.10: Stress-strain curves of laser processed and post-processed wires with heat treatment time of $2h$ and temperatures of $450°C$, $480°C$, and $500°C$.

The tensile failure curves of the heat treated wires are shown in Figure 4.11. As expected from literature, increasing the heat treatment temperature appears to lower the UTS of the wire from $1,073MPa$ to $1,027MPa$ [92]. Increasing the heat treatment time from $450°C$ to $480°C$ led to a relatively negligible UTS decrease of $2MPa$. There did, however, appear to be a significant effect on ductility, with the wires reaching strain values of 13.3%, 26.3%, and 38.5% as heat treatment temperatures increased from $450°C$ to $500°C$.

The failure characteristics of the $480°C$ heat treatment wire selected for the study were

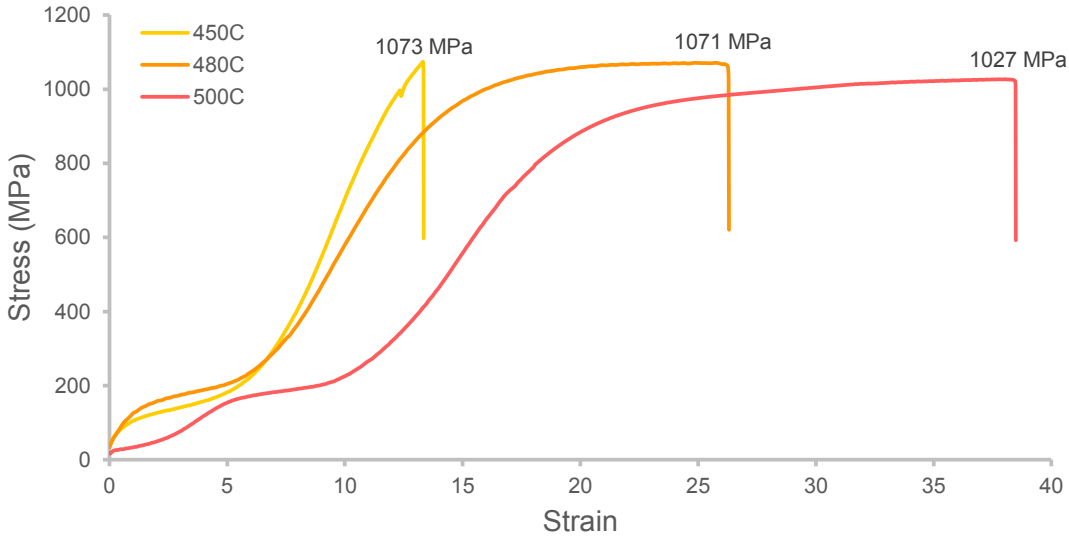

Figure 4.11: Tensile failure curves of laser processed and post-processed wires with heat treatment time of $2h$ and temperatures of $450°C$, $480°C$, and $500°C$.

also compared to those of the base metal and as-processed wires in Figure 4.12. Laser processing causes a significant drop in mechanical properties, with the as-processed wire retaining only 46% of its base metal UTS while gaining some ductility. Cold working and heat treatment at $480°C$ of the laser processed wire increased the UTS to 84% of the base metal value, which is a substantial improvement. Compared to the base metal and as-processed variants, the heat treated wire also saw an increase in ductility. The mechanical properties resulting from processing and post-processing as discussed in this section are summarized in Table 4.3. Due to the aforementioned trade-off between rise in transformation temperatures and decay of mechanical properties caused by increased heat treatment temperatures, a heat treatment of $480°C$ was found to result in optimal thermal and mechanical properties relative to the base metal. As a result, the samples examined in the following sections of this study were heat treated at $480°C$ for $2hrs$ following cold working.

4.4 Cyclic Properties

The results of 35 cycles of thermal cycling with $18N(238.5MPa)$ applied load performed on a final laser processed, heat treated multiple embedded memory wire sample are shown

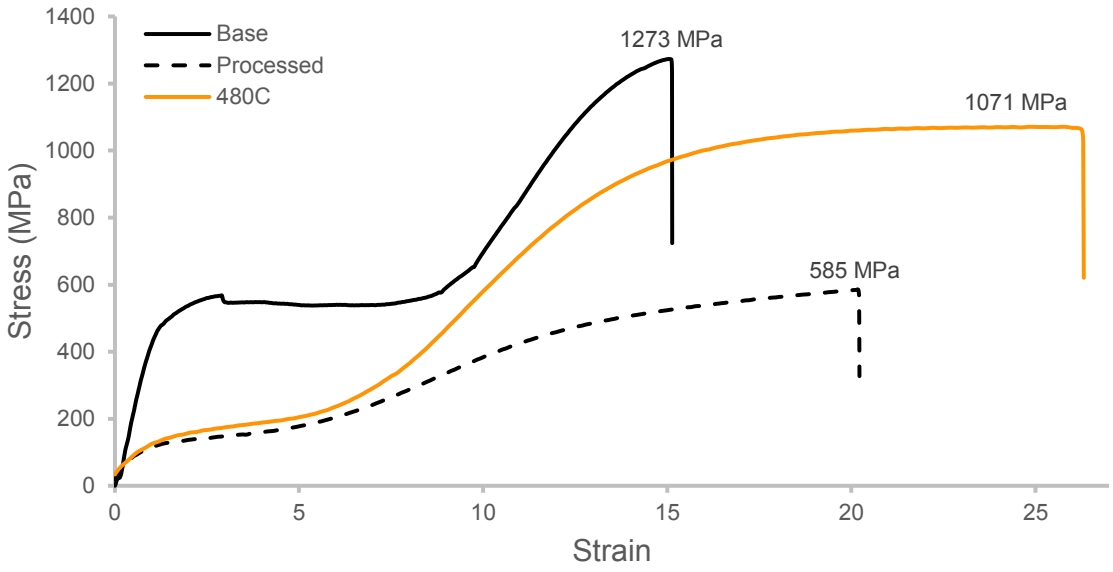

Figure 4.12: Tensile failure curves of base metal, as-processed, 480°C heat treated NiTi wires.

Table 4.3: UTS and ductility comparison of as-processed and heat treated wires relative to base metal NiTi.

Wire	UTS(%)	Ductility (%)
As-processed	46.0	134.7
450°C	84.3	88.7
480°C	84.1	175.3
500°C	80.7	256.7

in Figure 4.14. The steady-state position of the training cycles is also shown in Figure 4.13. The thermal training resulted in an unrecoverable extension of 3.9mm, which corresponds to residual strain of 2.6% relative to the sample length of 15.2cm. However, the residual strain appeared to plateau relatively quickly, reaching steady-state behaviour in as little as 20 thermal cycles. This performance is also reflected in Figure 4.14, which shows the resistance vs. position curves converging to a stable, repeatable cycle after about 20 cycles for both the PE and SME sections of the wire. The PE section of the wire exhibited distinct resistance behaviour relative to the SME section, likely due to partial transformation at room temperature as predicted from the DSC analysis. Nevertheless, the electrical resistance properties of the two memories differ greatly under identical applied

load and current. Thermal cycling appears to have a more significant effect on the resistance properties of the SME section, causing an increase of more than 0.15Ω. In contrast, the resistance properties of the PE section remained relatively constant throughout the thermal cycling.

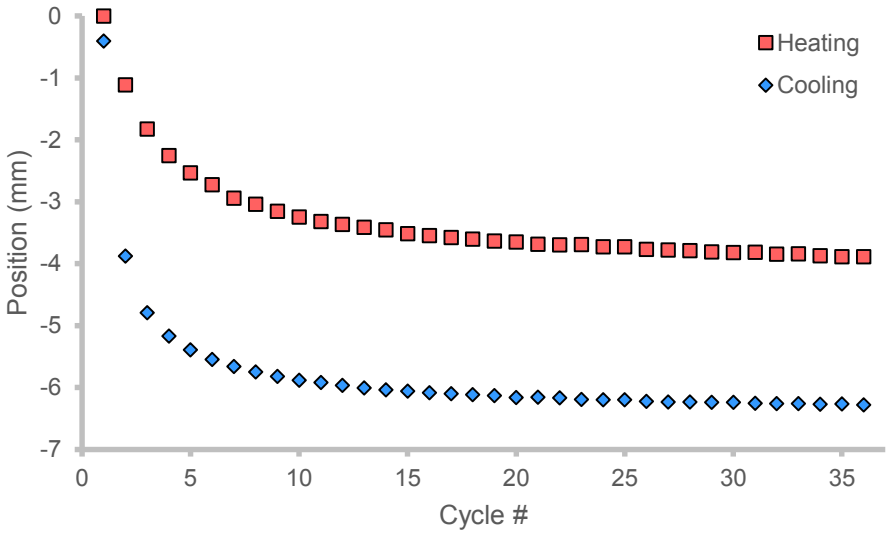

Figure 4.13: Steady-state position during heating and cooling cycles of multiple memory wire training.

As mentioned in section 2.1.5, the hysteretic nature of NiTi is complex and depends on many factors including applied stress. The cyclic behaviour of the multiple memory wire at various applied stresses is shown in Figure 4.15 for both the PE and SME sections with low and high applied currents of $0.15A$ and $0.8A$, respectively. Increase in stress applied to the wire causes a rise in the measured resistance, likely due to reduction in the cross-sectional area of the wire from increased extension. As this effect is based on physical deformation of the material, it is visible in both the PE and SME sections. Another consequence is the elongation of the hysteresis curve with increased stress. As the stress increases from $35.5MPa$ to $197.1MPa$, the positional length of the SME hysteresis curve effectively doubles likely due to an increased amount of detwinning. This effect is also observed in the partial PE section, although it is not as pronounced. Finally, increased applied stress causes the resistance range of the top and bottom of the hysteresis loops (corresponding to the temperature-induced change in resistance) to shrink only for the SME section while remaining somewhat constant for the PE section.

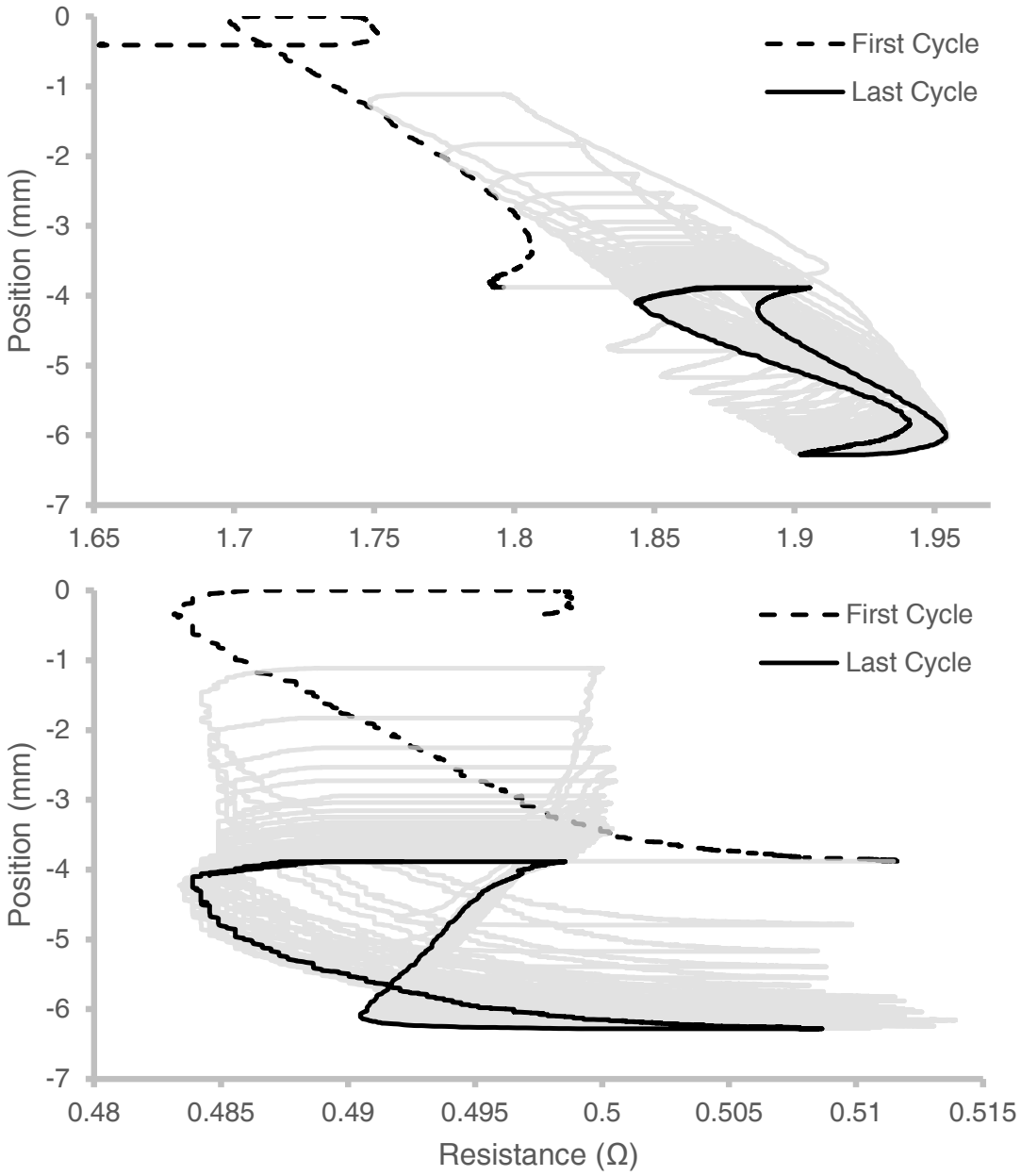

Figure 4.14: Thermal cycling of post-processed multiple memory wire showing resulting resistance vs. position curves of the SME (top) and partial PE (bottom) sections on the monolithic multiple memory wire.

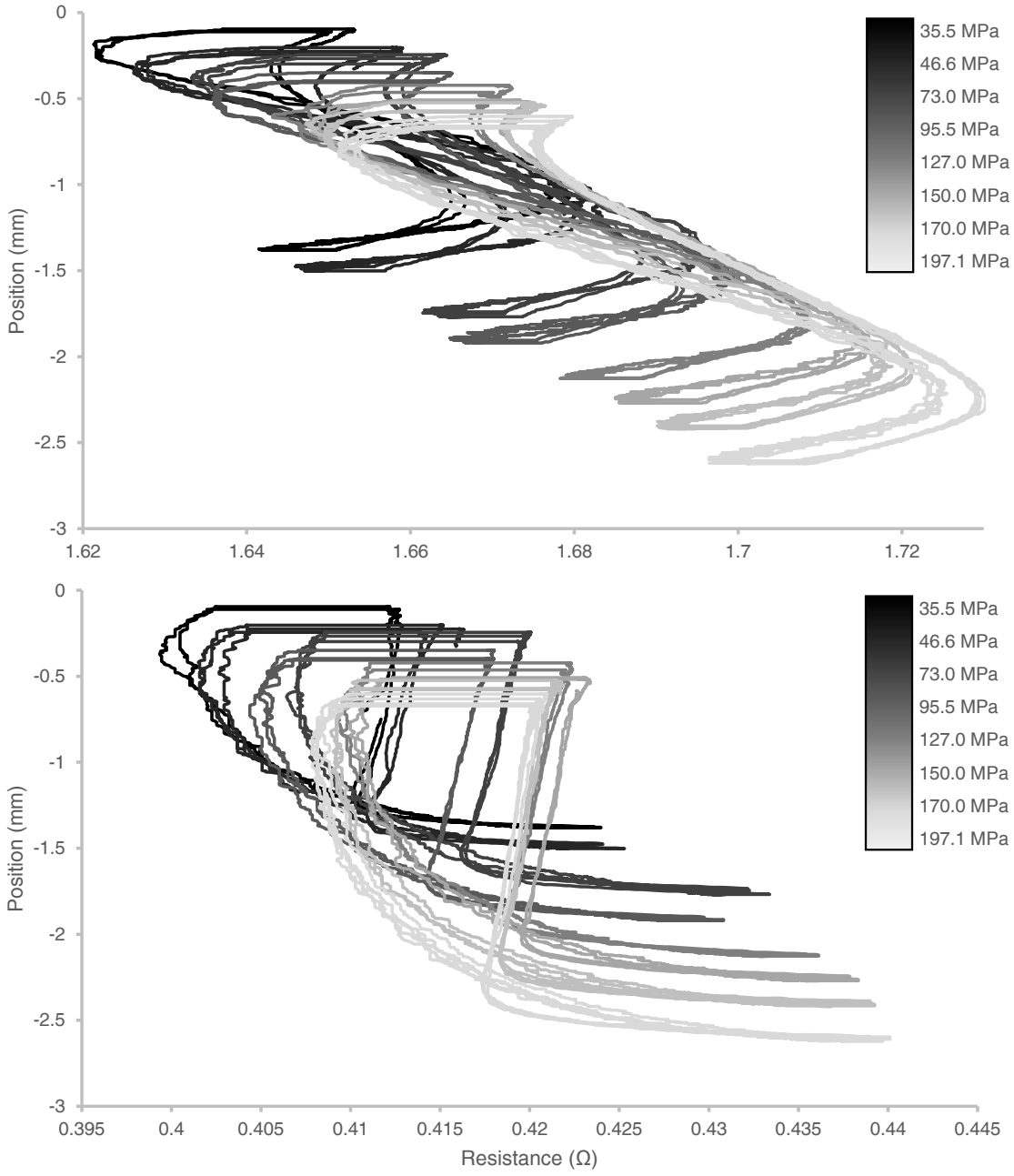

Figure 4.15: Cyclic behaviour of SME (top) and partial PE (bottom) sections of the fabricated multiple memory wire at various applied loads.

4.5 Chapter Summary

This chapter analyzed various thermomechanical properties of base metal and laser processed NiTi SMAs prior to and following post-processing. The effects of various laser processing parameters, including laser power and pulse time, on the transformation properties of NiTi were explored and optimized to successfully embed a second SME memory into PE NiTi wire. The post-processing techniques, including wire drawing and heat treatment, were also tuned to produce a final monolithic NiTi wire with one fully SME section and one partially PE section at room temperature. Furthermore, it was found that various metallurgical imperfections (such as voids) observed immediately after processing were not present after wire drawing and heat treatment. Compared to the base metal, the laser processed section of the final wire heat treated at $480°C$ for $2hrs$ exhibited M_s and A_f transformation temperature increases of $61.5°C$ and $35.3°C$, respectively.

Mechanical evaluation of the actuators was also performed, and it was found that laser processing causes a significant decrease in ultimate tensile strength. Using the optimal heat treatment of $480°C$ at $2hrs$, 84% of the base metal UTS was recovered. The mechanical stability of the actuator was apparent also during thermal cycling with applied load, with the actuator's cyclic behaviour stabilizing after about 15 cycles. Once the cyclic behaviour became fully stable, sufficient amounts of data were acquired which characterized the cyclic profile of the multiple memory NiTi actuator at varying loads and currents.

Chapter 5

Neural Network Position and Force Estimation of Multiple Memory Shape Memory Alloys

The complex, hysteretic, load-dependent behaviour of NiTi SMAs was demonstrated and characterized in Chapter 4. From the acquired results, it is evident that achieving accurate control of such an actuator is far from intuitive. This chapter will explore the use of recurrent neural networks with various architectures and hyperparameters for estimating the present position and load state of a NiTi actuator with two embedded memories using only the applied current and resistances across each memory. This method preserves the inherent benefits of SMA actuator - such as low cost and weight - by eliminating the use of external sensors within the actuator control system. The neural networks will be trained using the data acquired in the previous chapter. Two scenarios will be explored in this chapter: position estimation of a single embedded memory NiTi actuator with constant force, and force and position estimation of a NiTi actuator with two embedded memories and varying applied force.

5.1 Constant force position estimation using single resistance measurement

As previously mentioned, this section will focus on optimizing the hyperparameters of the RNN model performing position estimation with a constant applied force. Figure 5.1 illus-

trates the system into which the RNN was incorporated for performing position estimation. Due to time constraints, given the length of time necessary for training neural networks on such a large data sets, a limited number of variations were performed on each hyperparameter. Memory was a limiting factor in hyperparameter selection as well, as it was found that certain hyperparameter combinations required more memory than the system allowed (such as the combination of batch size of $1,024$, look back length of 1500, sparsity of 3, and 100 hidden RNN states, requiring the storage of $51,200,000$ points in a $[1,024 \times 500 \times 100]$ tensor). As a result, large batch sizes had to be balanced by smaller numbers of past time step data points and less hidden RNN states, and vice-versa.

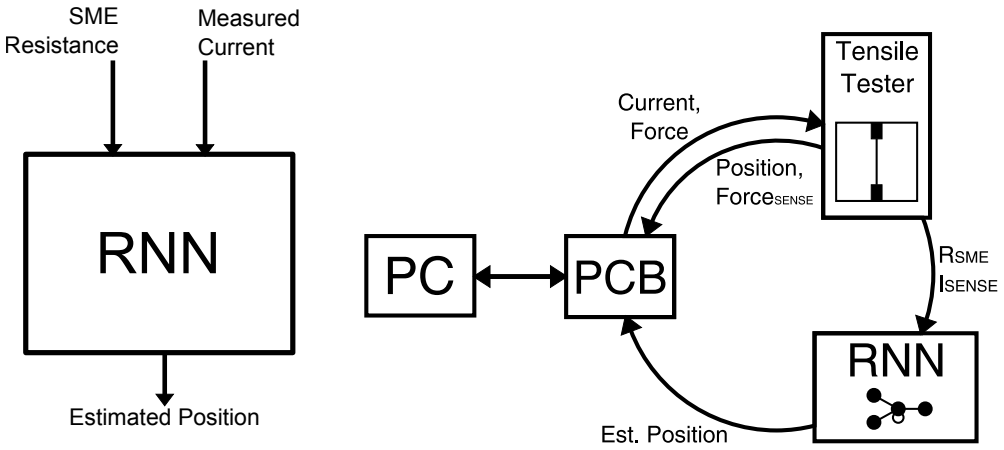

Figure 5.1: Schematic showing RNN inputs and outputs (left) and position estimation setup using single resistance RNN with constant applied load (right).

Each set of hyperparameter variations was performed on both the LSTM and GRU RNN architectures in an effort to find the optimally performing model. The base RNN model was also considered, although it was found to exhibit unstable gradient behaviour during backpropagation due to the large number of look back values necessary for this type of data. This behaviour was somewhat expected, as one of the main benefits of using LSTM and GRU is the reduction of this vanishing/exploding gradient problem [83]. Other ways of overcoming this challenge are to lower the learning rate, normalize the data to lower the difference between time steps, and decrease the look back length. Lowering the learning rate to as low as 0.0001 was found to still result in NaN loss values during model training (hinting at unstable gradients), with the data normalized and the large look back length necessary for properly capture the hysteretic behaviour of the NiTi actuator. As a result, the base RNN architecture was not considered in this hyperparameter optimization study.

Another discovery during the initial stages of the RNN optimization study was that, given the large number of look back points, the LSTM architecture also exhibited unstable gradients when attempting to use relatively higher learning rates (such as 0.01 or 0.001). In order for the LSTM architecture to properly train, the learning rate had to be lowered to 0.0001. On the other hand, the GRU architecture successfully trained with a learning rate of 0.01. As this is an important difference between the two neural network architectures, learning rates of 0.0001 and 0.01 were used for the LSTM and GRU architectures, respectively. Due to gradient stability with larger learning rates, the GRU architecture was found to generally converge to its minimum loss value significantly faster than the LSTM architecture, resulting in better performance in the same number of training epochs. The results comparing LSTM and GRU will be discussed in the following subsections, along with each hyperparameter variation.

5.1.1 Activation Function

The effect of various activation functions on the MSE during training of the RNN is shown in Appendix B.1 for both GRU and LSTM architectures. Figure 5.2 also shows the variance of each corresponding training curve for GRU and LSTM. The variance is calculated after 1000 training steps in order to circumvent the initial noise inherent to the training process.

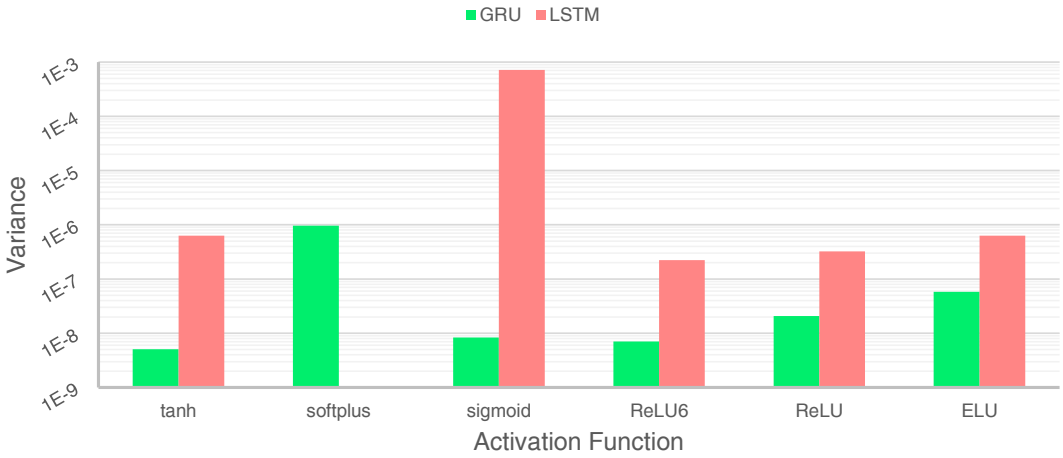

Figure 5.2: Variance of training curves for GRU and LSTM architectures using various activation functions.

For the GRU architecture, it is immediately visible that the *softplus* activation function

performs significantly worse than the others. ELU also stands out, as it appears to converge to the minimum MSE after $3000 - 4000$ training steps, which is significantly longer than the remaining activation functions. $ReLU$, $ReLU6$, $sigmoid$, and $tanh$ all showed similar convergence characteristics, with MSE reaching its minimum value after only 1000 training steps. $Softplus$ was also found to have the largest observed variance, more than two orders of magnitude larger than the variances of $tanh$, $sigmoid$, and $ReLU6$. ELU and $ReLU$ also showed relatively high variance, hinting at lower training stability.

Similar observations can be made for the LSTM architecture, with $tanh$, $sigmoid$, and $ReLU6$ showing the fastest convergence. Training was not possible using $softplus$, as it resulted in NaN MSE values. $Sigmoid$ also performed significantly worse, with MSE orders of magnitude larger than the remaining functions. As a result, the $sigmoid$ curves were not visible in the results as their MSE was too high. It also appears that the LSTM RNNs are converging to larger minimum MSE values for each activation function compared to their GRU counterparts.

Aside from the differences between individual activation functions, the training behaviour of the LSTM architecture differs significantly from that of GRU. It can be seen that convergence did not occur until around $3000 - 4000$ training steps for the best performing activation functions, showing significantly slower convergence compared to GRU. This effect was likely caused by the slower learning rate used for LSTM in order to enable training. Furthermore, the variance for each activation function's training curve is about one order of magnitude larger relative to their GRU counterpart. Overall, the training behaviour was slower and more unstable using the LSTM architecture.

The performance of the RNN was also validated on previously unseen data during training for each activation function. The results of this analysis are shown in Appendix C.1 for both GRU and LSTM. For the GRU architecture, $tanh$, $sigmoid$, $ReLU6$, and $ReLU$ showed similar performance, reaching the minimum MSE within 10 training epochs and maintaining a fairly constant value throughout the remainder of the RNN training. The poorest performance activation function was $softplus$, with ELU showing initially poor performance and later recovering to converge after 15 epochs. For the LSTM architecture, $ReLU6$ and $ReLU$ appeared to be the most stable activation functions in addition to converging to a relatively low MSE. ELU and $tanh$ had comparatively good performance compared but were less stable than $ReLU$ and also converged more slowly. Due to the poor performance of $sigmoid$, its MSE is so massive that it is not visible in the plot. Altogether, as was the case for training behaviour, the GRU architecture yielded significantly more desirable training results than LSTM when observing validation performance.

Finally, the performance of the activation functions after fully completing RNN training

was also evaluated, and the results are summarized in Figure 5.3. The overall best performance was shown by *tanh* using the GRU architecture for training, validation, and testing MSE. *ReLU*, *ReLU*6, and *sigmoid* exhibited comparable results, with only marginally higher MSE values. As expected from the training behaviour, *softplus* proved to be the worst performing activation function, which was observed across both RNN architectures. GRU was also found to consistently outperform LSTM across all activation functions. Furthermore, the training, validation, and testing MSE for the best performing activation functions (*tanh*, *sigmoid*, and *ReLU*6) appeared to be relatively equal, meaning the RNNs trained using the training data subset can successfully generalize to data they have never previously seen and produce low MSE values. Taking everything into account, *tanh* and *ReLU* activation functions using GRU architecture were selected to be used in the final position prediction RNN testing. Of these two functions, the one which performed best with the remaining optimal hyperparameters was selected for the final position prediction model.

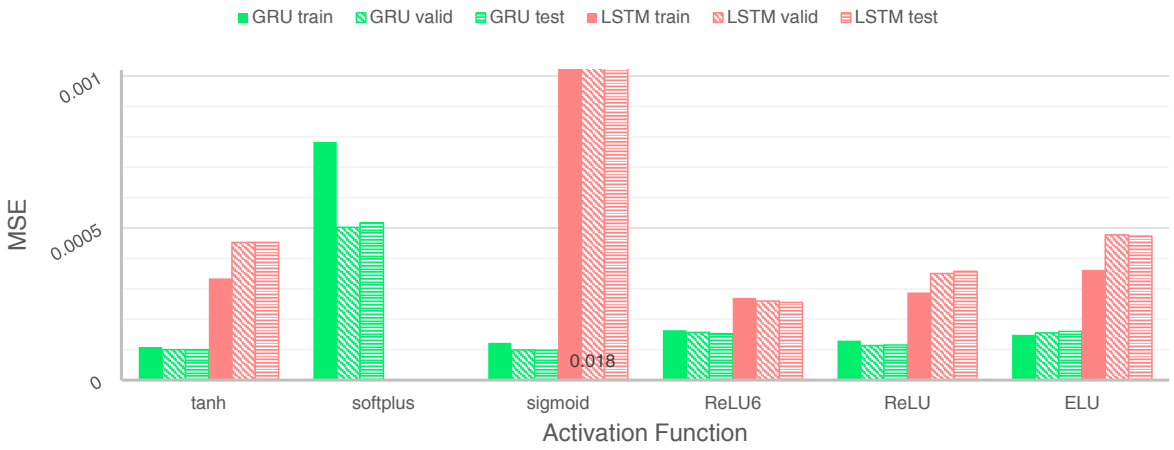

Figure 5.3: Performance of fully trained RNNs with various activation functions on training, validation, and testing data sets.

5.1.2 Batch Size

The size of batches used during training was the second hyperparameter optimized during this study. The training behaviour of each batch size for GRU and LSTM architectures is shown in Appendix B.2. Compared to the activation function variation previously studied,

the behaviour of all three batch sizes appeared to be comparable. However, some differences were still observed, one of which is that larger batch sizes converged in less training steps than smaller batch sizes. This observation is expected, as larger batch sizes result in training on larger amounts of data within one training step. A batch size of 512 contains twice as many data points as a batch size of 256, meaning that an RNN with 512 batch size will have been trained on twice as many data points as one with 256 batch size after the same number of training steps. As a result, larger batch sizes are expected to lead to faster convergence. In addition, because more data is used per training step and the arithmetic operations calculated during each training step should be performed in parallel on the GPU, less overall training time should be required for larger batch sizes. As can be seen in Figure 5.4, larger batch size did indeed significantly lower the required training time for both GRU and LSTM. This effect did not appear to be linear, as lowering the batch size from 256 to 128 increased training time from $3.5hrs$ to $9.6hrs$ for the GRU architecture. The LSTM architecture saw a less dramatic increase in training time, increasing from $4.2hrs$ to $7.4hrs$ for the same change in batch size.

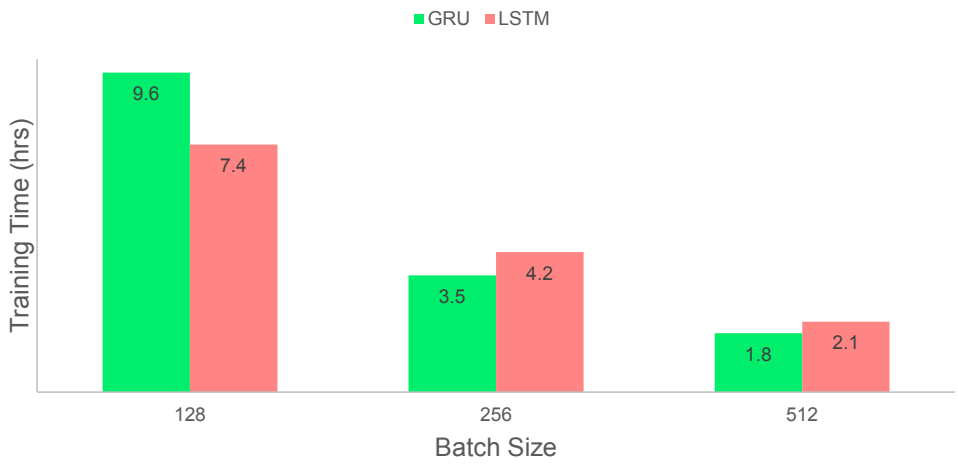

Figure 5.4: Training time of RNNs with GRU and LSTM architectures using varying batch sizes.

Despite the numerous benefits acquired from increasing the batch size, there is also a major drawback. Because more data points are used during each training step, less overall training steps are performed on the RNN. This means that large batch sizes may deteriorate the final performance of a neural network. There is therefore a trade-off between training speed and final neural network performance. This effect is visible in the training behaviour

for both GRU and LSTM architectures, as the 512 batch sizes appeared as though they would have continued converging to a smaller MSE value like the lower batch sizes had there been more training steps. Although the RNNs appeared to perform somewhat similarly across all batch sizes during training, this was not true when evaluating their performance on validation data as can be seen in Appendix C.2. From the results, it appears that increase in batch size resulted in larger validation MSE for both GRU and LSTM. The batch sizes of 128 and 256 yielded similar MSE, whereas batch size of 512 caused a considerable increase in MSE. The performance of LSTM compared to GRU was again inferior, with significantly larger final MSE values and much slower convergence. Furthermore, as is visible in both training and validation curves for LSTM, the training cycle with batch size of 128 did not fully complete as the system ran out of memory after 17 training epochs.

The performance of the fully trained GRU and LSTM RNNs with varying batch sizes is shown in Figure 5.5 on training, validation, and testing data sets. For both architectures, the previously described effect holds for the testing evaluation. Increasing batch size caused the resulting MSE to also increase, with the effect appearing to have exponential behaviour. For GRU, increasing batch size from 256 to 512 caused a MSE increase of more than 3 times the one resulting from increasing batch size from 128 to 256. Taking all factors into account, lowering batch size to 128 yielded marginal improvements in MSE while significantly increasing training time for GRU. For LSTM, slightly larger improvements in MSE were attained for the same batch size adjustment. On the other hand, for both architectures, increasing the batch size to 512 caused a considerable rise in MSE while lowering the training time by a less significant amount. The optimal choice in this scenario was a batch size of 256, which was used in the final position estimation model. This batch size balances training cycles steps with training time in order to produce low MSE and reduced training time.

5.1.3 Number of Hidden RNN States

The third hyperparameter analyzed was the number of hidden states in the GRU/LSTM recurrent cells, with the training performance for each variation outlined in Appendix B.3. After sufficient training steps, RNN state sizes converged to virtually the same final MSE value. For GRU, state size of 50 converged rather quickly but spiked after $2,000$ iterations which may hint at training instability. Variance of the 125 state size RNN also appeared to be higher than the other sizes. On the other hand, for LSTM, the 125 state size saw the fastest convergence and smallest variance. However, a state size of 125 caused the system to run out of memory after 17 training epochs, ending training prematurely. It was also found that changing the number of hidden states did not result in significant change in

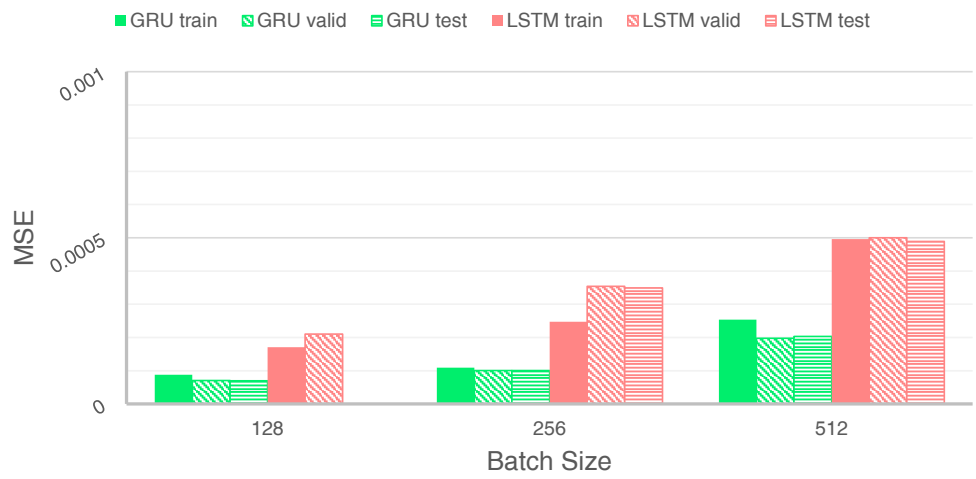

Figure 5.5: Performance of fully trained RNNs with varying batch sizes on training, validation, and testing data sets.

training time due to parallelization. Overall, the RNN with hidden state size of 100 with GRU shows the best stability and convergence behaviour.

The validation performance shown in Appendix C.3 echoed the previously mentioned observations in that the final MSE value reached was practically equal for all state sizes. For GRU, the 125 state size RNN required longer training time to converge and exhibited unstable behaviour during the initial training epochs but stabilized after 10 epochs. In contrast, the 50 state size RNN converged relatively quickly but appeared somewhat unstable. For LSTM, 125 neurons yielded fastest convergence but prevented the network from fully training, with 50 hidden states resulting in slowest and least stable convergence. The fastest convergence and largest stability were exhibited by the 100 state size RNN.

The fully trained RNNs' performance with varying state sizes is displayed in Figure 5.6. Again, constant MSE performance was observed over all hidden state variations. Furthermore, the train, validation, and test MSE were approximately constant relative to each other, meaning that the network has good generalizability. Taking into account the slight performance differences observed during validation and the lack of other major discrepancies between the state sizes, the final RNN used for position prediction was selected to have a hidden state size of 100 using the GRU architecture.

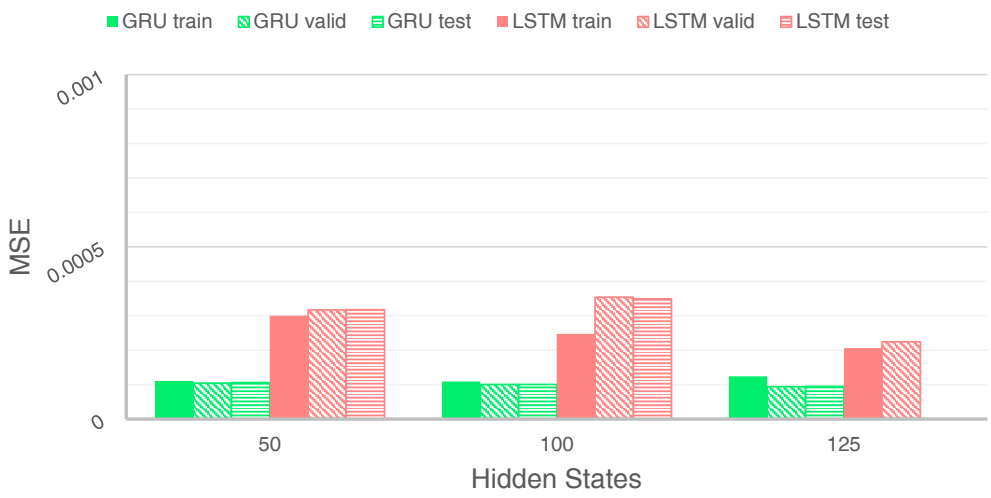

Figure 5.6: Performance of fully trained RNNs with varying numbers of hidden RNN states on training, validation, and testing data sets.

5.1.4 Number of Epochs

The result of varying the number of training epochs on the RNNs' training performance is illustrated in Appendix B.4 for both GRU and LSTM architectures. Because the remaining hyperparameters are identical, virtually the same performance was observed over all numbers of epochs for GRU. However, significant differences were observed in LSTM, possibly due to instabilities in the model. The GRU RNNs trained stably to the minimum MSE value, with the number of training epochs determining the MSE that is ultimately reached. As the number of epochs increases from 1 to 30, the minimum MSE continued to fall during training. As training continued, the training stability also improved, especially after 10 epochs - this can be seen from the decreasing variance in MSE. As for LSTM, all curves eventually converged to the identical MSE value, as expected. Other than the initial differences, similar behaviour was observed as for the GRU architecture, although with significantly more spikes in MSE.

When observing the RNNs' behaviour on validation data in Appendix C.4, similar conclusions were made as for the training data. As the number of epochs increased, the RNNs' performance on never before seen data also significantly improved. Although this is expected to plateau after a large enough number of epochs, a significant drop in MSE was obtained by raising the number of epochs to 30 for both GRU and LSTM. Additional

improvements in MSE are likely attainable by further increasing the number of epochs, but as is visible in Figure 5.7, increasing the number of epochs caused an exponential rise in training time. An increase from 20 to 30 epochs caused training time to increase from $3.5hrs$ to $7.9hrs$ for GRU and from 4.2 to 8.5 hours for LSTM. This trend is expected to continue, so an increase to 40 epochs may require upwards of 16 hours to train. Of course, the improvements in MSE are also expected to eventually plateau, meaning any gains in model performance would be marginal after an adequate number of epochs.

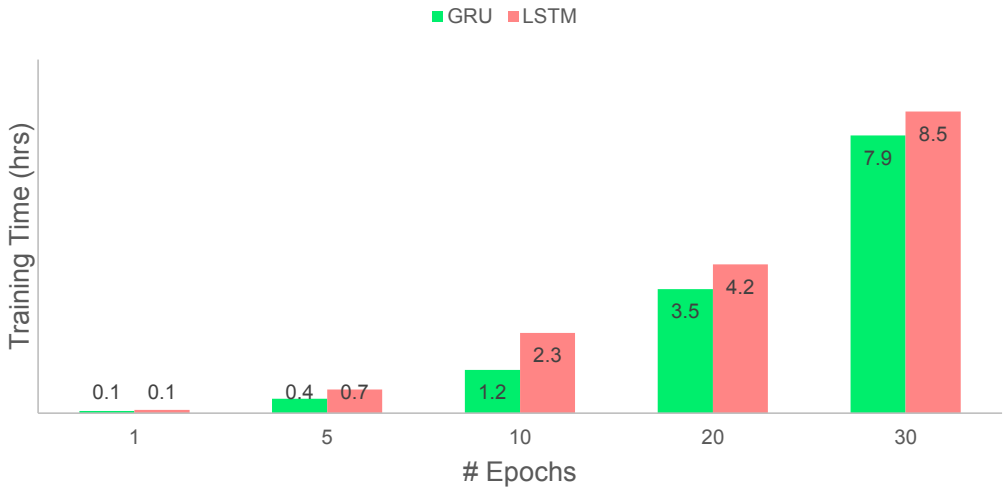

Figure 5.7: Training time of RNNs with GRU and LSTM architectures for various numbers of training epochs.

Finally, the performance of the fully trained model on training, validation, and testing data sets is shown in Figure 5.8. Again, increasing the number of training epochs caused the final MSE to drop in the case of all three data sets. The effect appears to be most dramatic when increasing from 1 to 5 epochs, after which marginal gains in performance were acquired. Comparing the two architectures, it can be seen that LSTM showed much higher MSE values at low numbers of epochs, meaning GRU has superior training performance at low epochs. Increasing to 30 epochs caused a significant drop in MSE relative to 20 epochs for all three data sets. Furthermore, the training, validation, and testing MSEs were fairly constant, with some variations in the training MSE for lower epoch numbers likely caused by the increased variance observed early in the training. Still, this means that the RNNs can generalize well to new data. Overall, 30 epochs resulted in the best performance while maintaining acceptable training time. As a result, the final position estimation RNN was

trained for 30 epochs.

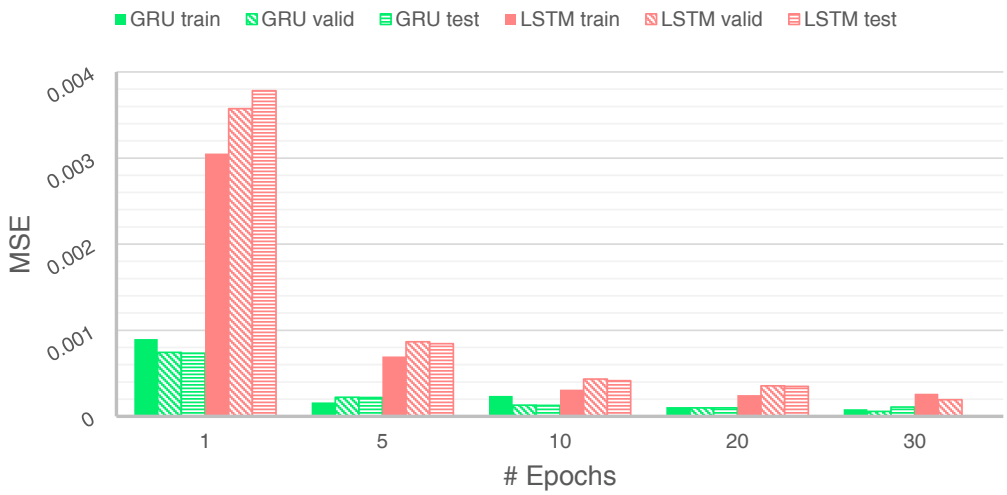

Figure 5.8: Performance of fully trained RNNs with varying numbers of training epochs on training, validation, and testing data sets.

5.1.5 Data Sparsity

The effect of adjusting the look back data sparsity on the RNN performance is examined in this section, with the training curves for various levels of sparsity used during training shown in Appendix B.5 for both GRU and LSTM. For GRU, all levels of sparsity resulted in similar convergence behaviour. However, using every 3^{rd} data point was shown to result in a significantly more stable MSE training curve than using every 5^{th} or 10^{th} data point. Nevertheless, as training progressed, all sparsities reached a fairly constant training MSE variance. The LSTM architecture showed somewhat different training behaviour, with slower MSE convergence exhibited by using every 3^{rd} data point compared to the remaining levels of sparsity. The variance behaviour of the aforementioned sparsity was also significantly worse than the GRU counterpart. Furthermore, all levels of sparsity converge more slowly in LSTM than GRU and ultimately reached a larger MSE value. It was also found that the level of sparsity did not significantly affect RNN training time, as all data points were processed in parallel. However, lower levels of sparsity required the simultaneous storage of much larger amounts of data, so a sparsity of 1 (using every data

point) was not achievable using the base batch size of 256 and look back length of $1,500$ due to limited computational processing power.

Looking at the validation MSE of GRU in Appendix C.5, it is clear that using every 10^{th} data point for training does not translate well to data never previously encountered. The MSE of the 10^{th} data point sparsity showed greater instability, especially between $5-15$ epochs. Furthermore, the final MSE value reached was significantly higher than using every 3^{rd} or 5^{th} data point. This poor model performance may be caused by the RNN's failure of capturing the actuator behaviour when trained with highly sparse data. A similar effect is observed in the LSTM architecture, with unstable behaviour observed between $5-10$ epochs for the 10^{th} data point sparsity. Nevertheless, the 3^{rd} data point results in larger MSE, possibly due to training instabilities during the final training epochs. LSTM was in general less stable and again arrived at a larger MSE than GRU. Furthermore, due to the slower convergence behaviour of LSTM, it appears that increasing the number of training epochs would yield further MSE improvements given the downward trend of the validation MSE curves.

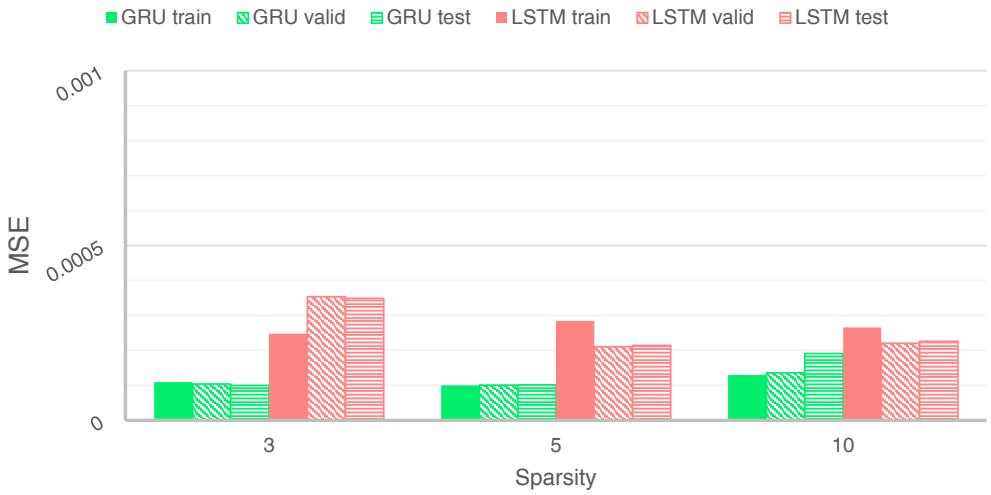

Figure 5.9: Performance of fully trained RNNs with varying levels of look back sparsity on training, validation, and testing data sets.

Evaluation of the fully trained RNNs on the training, validation, and testing data sets was also performed, and the results are displayed in Figure 5.9. Overall, LSTM shows poor MSE performance relative to GRU. For the GRU architecture, the 3^{rd} and 5^{th} data point levels of sparsity yielded similar MSE results, with constant MSE across the three data

sets showing good model generalizability. However, increasing the sparsity to every 10^{th} data point not only caused the MSE to increase, but also resulted in larger testing and validation MSE compared to the training MSE. This shows that the model with the largest level of sparsity was not able to generalize well to new data, which is highly undesirable. As for LSTM, using every 5^{th} data point yielded the lowest MSE, with significant increases in MSE when either lowering or increasing the training data sparsity. These effects are believed to result from model instability caused by the LSTM architecture. As the GRU architecture showed better overall performance, the 3^{rd} data point level of sparsity was be used in the final position estimation RNN architecture.

5.1.6 Look Back Length

The final hyperparameter optimized in this study is the look back length during position estimation, and the results for this optimization are displayed in Appendix B.6. In the case of GRU, look back length of 500 yielded greatly more unstable behaviour relative to the other values. Interestingly, look back length of $2,000$ also appeared slightly more unstable than look back length of $1,500$. The reason for this may be the fact that such a large look back may be exiting the current hysteresis loop and entering the previous loop, misleading the RNN and producing lower prediction accuracy. As for LSTM, look back lengths of $1,500$ and $2,000$ exhibited similar training performance, with 500 clearly showing comparatively poor performance. Altogether, the look back length of $1,500$ yielded best training performance.

The RNNs' performance was also evaluated on validation data during training, and the results are shown in Appendix C.6. For GRU look back length of $1,500$ yielded the best results - it was the most stable and converged to the lowest MSE after 20 epochs. Look back of $2,000$ was slightly more unstable and yielded a relatively small increase in MSE. The opposite was true for LSTM, with look back of $2,000$ showing superior performance and $1,500$ closely trailing. From the results, it is also clear that look back length of 500 was not enough to properly capture the hysteretic behaviour of the actuator - for both architectures, the validation convergence was significantly slower and more unstable than the larger look back lengths.

The results of evaluating the fully trained RNNs on the training, validation, and testing data sets are summarized in Figure 5.10. For GRU, the best performance resulted from look back length of $1,500$ with a low MSE consistent over all data sets. Look back length of $2,000$ yielded slightly worse MSE while also somewhat maintaining model generalizability. As previously discussed, a look back length of 500 showed significantly deteriorated

performance, especially on the validation and testing data sets. LSTM saw best MSE performance from look back of $2,000$, although GRU performed better overall for all look back lengths. Altogether, the look back length of $1,500$ for the GRU architecture yielded optimal MSE and stability across the three data sets.

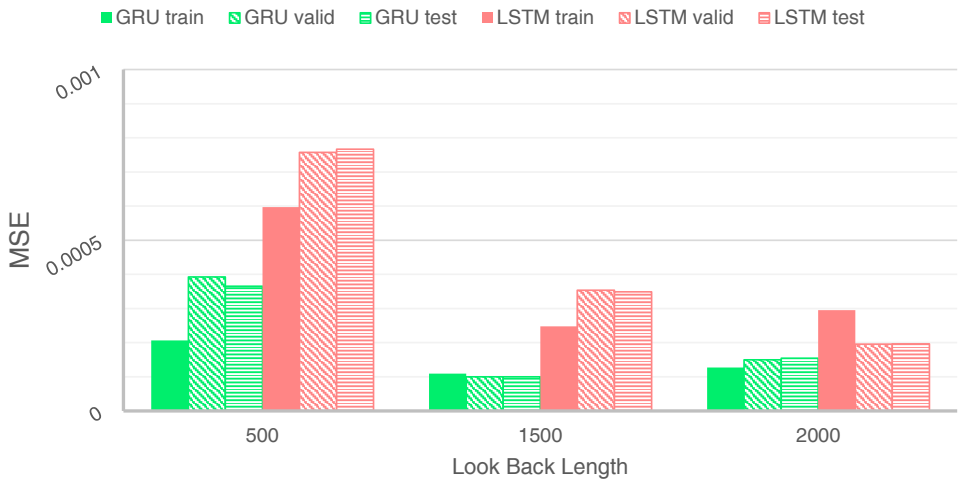

Figure 5.10: Performance of fully trained RNNs with varying look back lengths on training, validation, and testing data sets.

5.1.7 Position Estimation Results

Taking all previously examined factors into account, it was found that the RNN model hyperparameters listed in Table 5.1 resulted in the best position estimation performance on the acquired NiTi actuator cycling data. Using these hyperparameters, two separate models (one for each activation function) were trained on the training data set used in Section 5.1.1.

Table 5.1: RNN model hyperparameters resulting in best position estimation model performance.

Hyperparameter	Value
Activation Functions	*tanh* and *ReLU*
Batch size	256
Hidden RNN States	100
Number of Epochs	30
Data Sparsity	3
Look Back Length	1500

The position estimation results for the two RNN models are shown in Figure 5.11 on a subsection of the entire $10N$ applied force data set. It can be seen that both models are able to successfully estimate the real position of the actuator with excellent performance. Furthermore, it is clear that the RNN models have truly captured the hysteretic behaviour of the NiTi actuator, successfully predicting both major and minor hysteresis loops. Although a small subset of data is shown, this estimation performance persisted throughout the entire data set. Modeling of the major and minor hysteresis loops is also shown in Figure 5.12, which again shows the models' success in learning NiTi's hysteretic nature.

From the accuracy plot shown in Figure 5.11, it is clear that the RNN models almost perfectly estimate the actuator's position given past measured current and R_{SME} data. In fact, across the entire data set, the average accuracy for the *ReLU* position estimation model was 99.2%, with the *tanh* model trailing slightly at 98.7%. Even though the models perform almost identically, due to the slight edge in accuracy, the optimal final RNN model for constant load position estimation of a NiTi actuator with arbitrary applied current was the model using *ReLU* along with the remaining optimized hyperparameters.

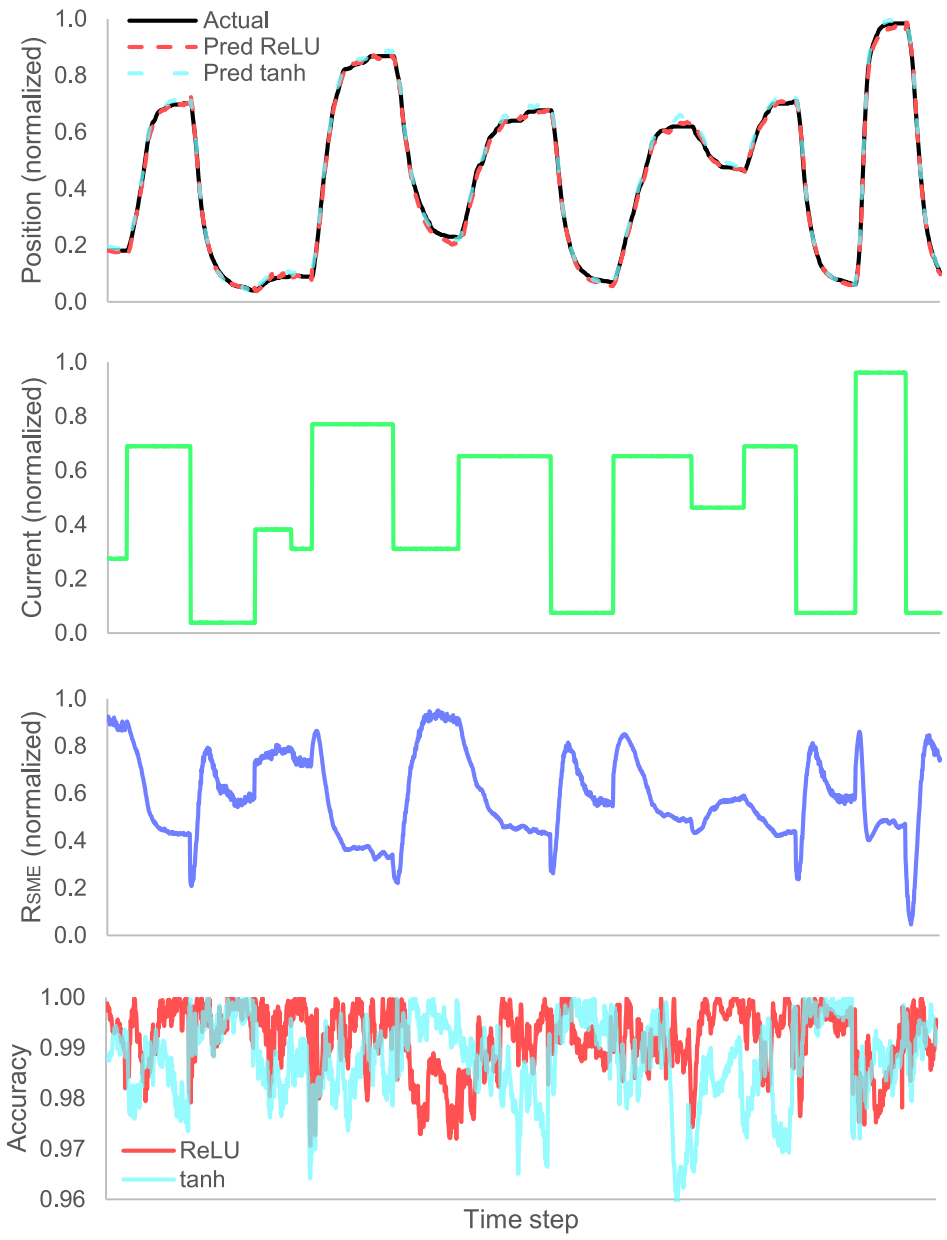

Figure 5.11: Starting at the top: position estimation of RNN models with *tanh* and *ReLU* activation functions vs. real position, along with corresponding applied current, measured R_{SME}, and prediction accuracy of both models, all with respect to time.

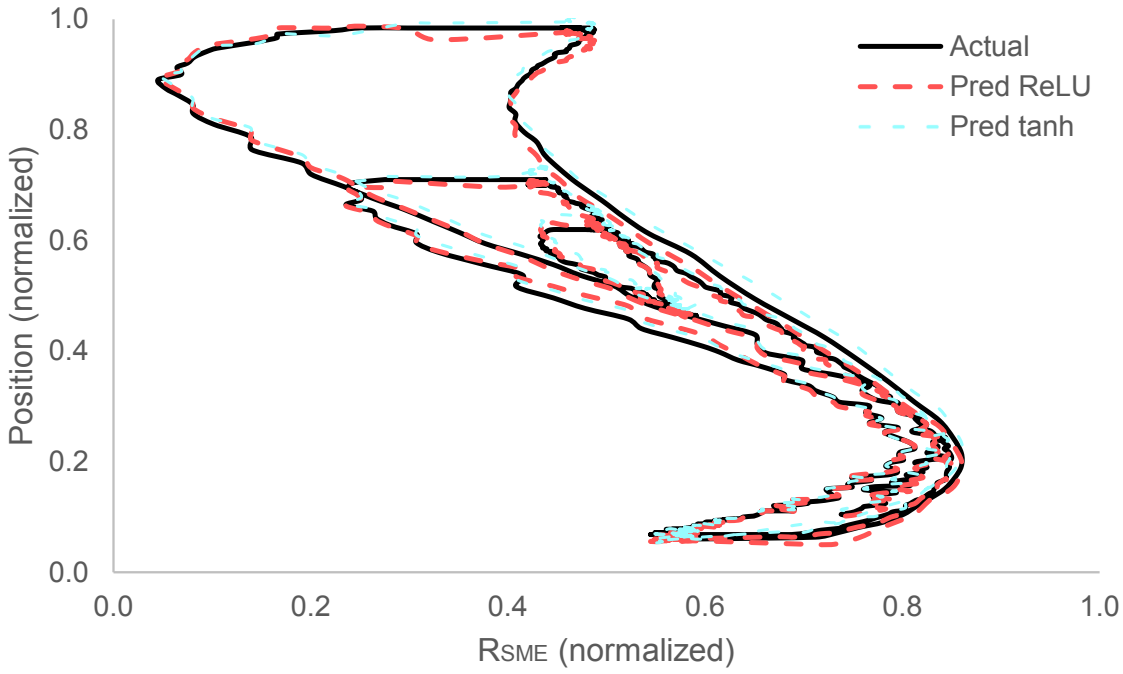

Figure 5.12: Position estimation using *tanh* and *ReLU* RNNs resulting in successful tracking of NiTi hysteresis curves.

5.2 Position and Force Estimation Using Dual Resistance Measurements

The system used for performing dual resistance RNN position and force estimation is illustrated in Figure 5.13. As the models analyzed in Section 5.1.7 exhibited essentially identical position estimation performance, only the *ReLU* activation function RNN was studied in this section due to achieving marginally higher accuracy. The position and load estimation results under two applied loads ($4N$ and $12N$) using the training data are shown in Figure 5.14. As before, it is apparent that the RNN model can effectively perform estimation of both force and position on data used for training, which is reflected in the high accuracy values. Other parameters corresponding to these actuation cycles (such as current and measured resistances) can be found in Appendix D.1.

The estimation accuracy for each applied load data set is shown in Table 5.2. It can

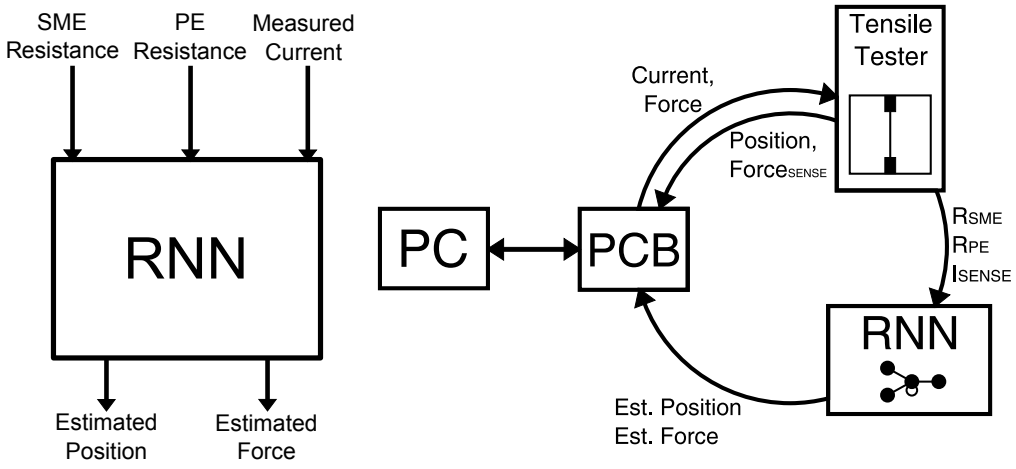

Figure 5.13: Schematic showing RNN inputs and outputs (left) and position estimation setup using single resistance RNN with variable applied load (right).

be seen that the overall estimation accuracy is 98.5% and 96.0% for position and force, respectively. It appears that position estimation consistently outperforms force estimation, albeit by only a few percent accuracy. Overall, the RNN model achieves excellent prediction performance on training data with varying applied loads. Figure 5.15 shows the prediction of the minor and major hysteresis loops for $4N$ and $12N$ applied loads using training data. Again, the model appears to successfully capture the hysteretic behaviour of NiTi even with varying applied load. Furthermore, several position prediction imperfections can be seen from the hysteresis loops which are not expected to significantly hurt the actuator performance in practice as the estimated value eventually recovers to the true position.

The performance of the RNN model was also evaluated on varying applied load data sets which the model had never previously seen (testing data), and the results for applied loads of $4N$ and $9N$ are shown in Figure 5.16. The model performed very well given that it had not previously the sequence of applied currents and measured resistances given in the test data sets. As can be seen, the force estimation for $4N$ applied load yielded significantly lower accuracy than for $9N$. This hints that the actuation sequences utilized for model training may not be long enough to fully characterize the actuator's cyclic behaviour. Therefore the discrepancy between $4N$ and $9N$ may be due to the data set randomness. Acquiring more data or adjusting the model architecture may result in significant performance improvements. On the other hand, the position estimation performance was excellent in both scenarios. Other parameters corresponding to these actuation cycles (such as current and measured resistance) can be found in Appendix D.2.

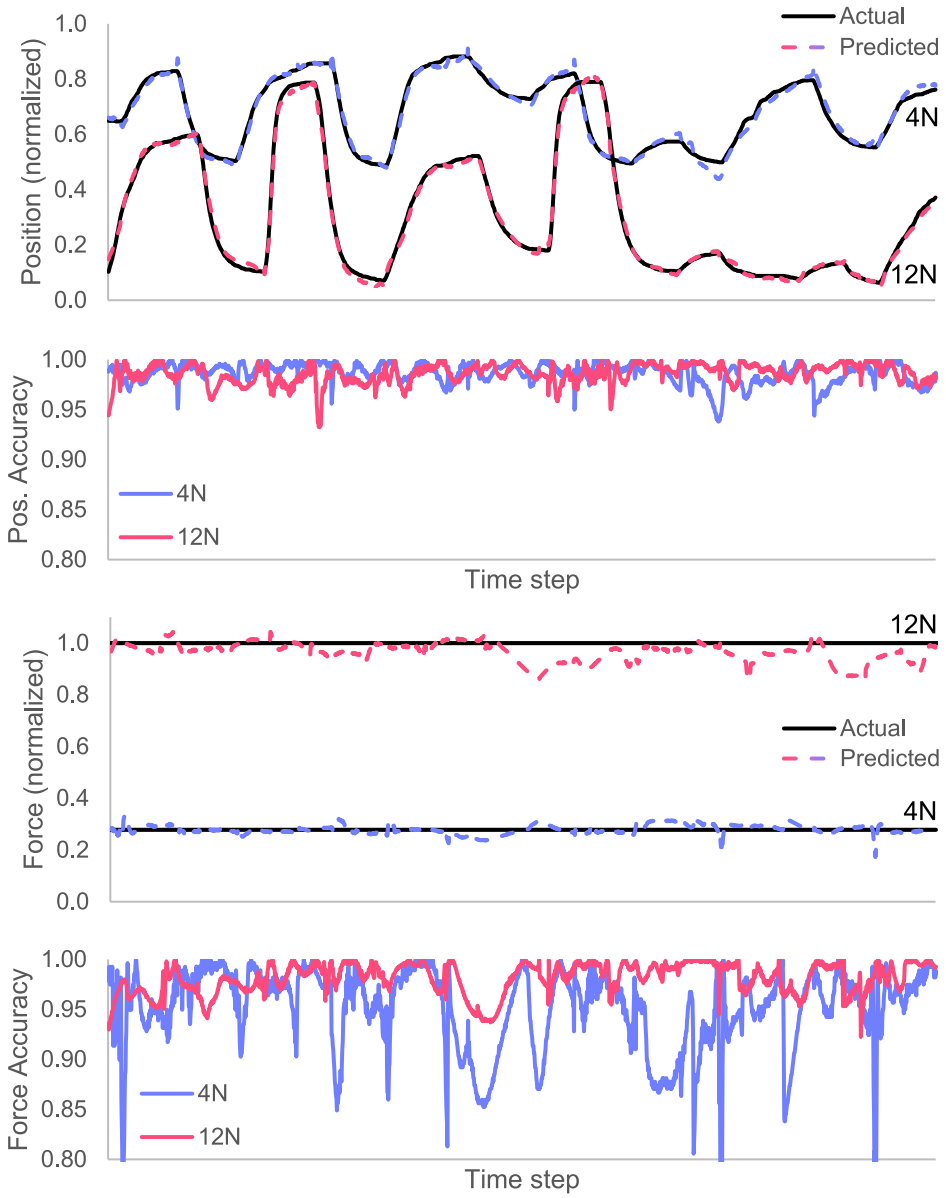

Figure 5.14: Starting at the top: position estimation of RNN model vs. real position under $4N$ and $12N$ applied load along with corresponding position estimation accuracy, and force estimation of RNN model vs. real $4N$ and $12N$ applied loads along with force estimation accuracy. Evaluation was performed using training data sets.

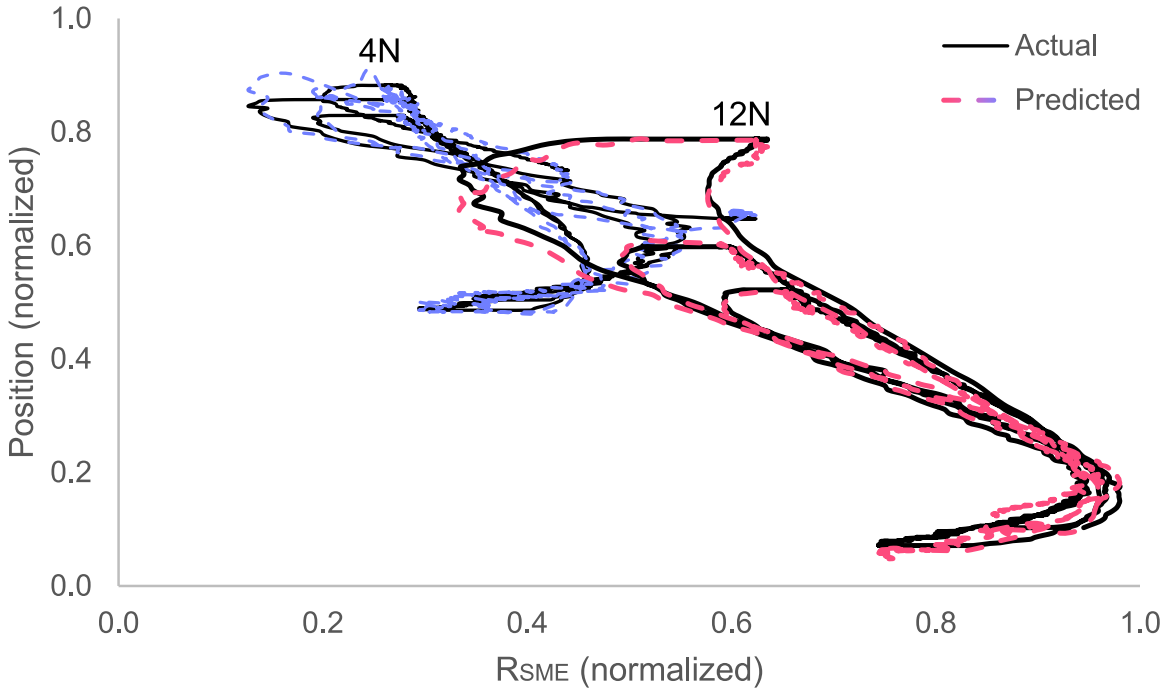

Figure 5.15: Position hysteresis curve estimation results on training data with $4N$ and $12N$ applied force.

Table 5.2: Position and force estimation accuracy of RNN model on the entire varying load training data sets.

Data Set Applied Load (N)	Position Accuracy (%)	Force Accuracy (%)
3	98.3	92.9
4	98.6	93.6
5	99.0	97.3
7	98.3	97.2
8	98.6	97.5
9	98.4	97.2
12	98.4	96.6
Overall:	98.5	96.0

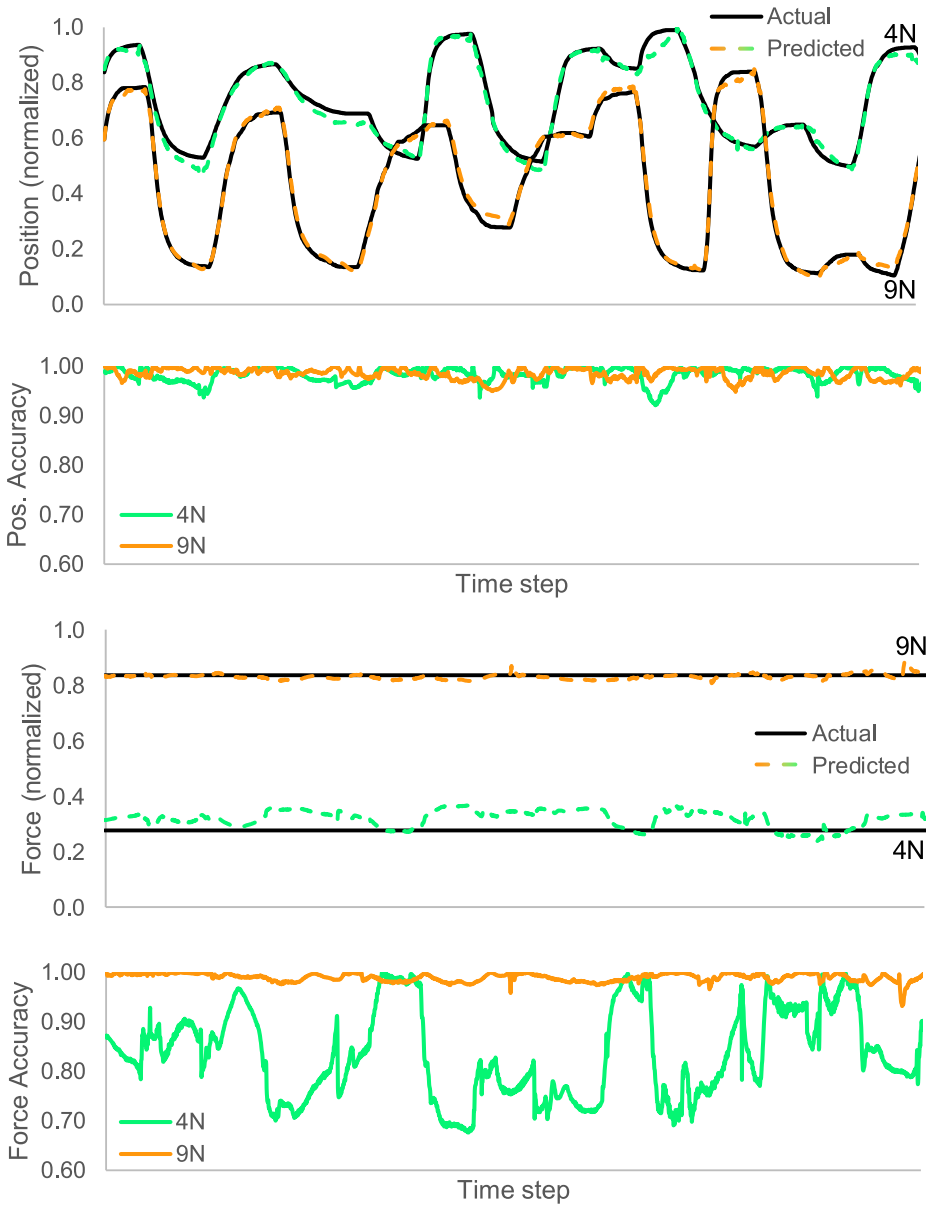

Figure 5.16: Starting at the top: position estimation of RNN model vs. real position under $4N$ and $12N$ applied load along with corresponding position estimation accuracy, and force estimation of RNN model vs. real $4N$ and $12N$ applied loads along with force estimation accuracy. Evaluation was performed using testing data sets.

The calculated prediction accuracy values for each testing data set are shown in Table 5.3. Significant variations in accuracy are visible, especially for force estimation. Nevertheless, the model successfully estimated the NiTi actuators' position (including minor loops) as well as the applied load. The overall model accuracy on testing data was 96.6 and 89.8 for position and force estimation, respectively. The performance is not as high as that of training data, meaning the model does not perfectly generalize to new data. Still, the model generalization is sufficient for accurate position control and rough force estimation. Figure 5.17 also illustrates the model's ability to estimate the actuator's minor hysteresis loops at different applied loads. The aforementioned changes (longer cycling sequences, optimized RNN architecture for varying loads) may further improve the model's estimation performance.

Compared to the mathematical model proposed by Zamani et al. ([100]) which requires ideal PE and SME phases, not only does the RNN model achieve higher accuracy on a significantly larger set of actuation data by capturing the full actuator behaviour, but it was also able to utilize a partially transformed PE phase to successfully perform position estimation. Consequently, the optimized RNN architecture can be trained on data acquired from SMA actuators with varying compositions, geometries, and phases and likely result in accurate position and force estimation models. It can therefore be concluded that the RNN model exhibits better performance and generalizability relative to the mathematical model.

Table 5.3: Position and force estimation accuracy of RNN model on the varying load testing data sets.

Data Set Applied Load (N)	Position Accuracy (%)	Force Accuracy (%)
3	98.3	83.9
4	88.7	84.4
5	99.1	96.8
7	98.4	96.6
8	98.4	91.9
9	98.4	94.4
12	94.8	80.4
Overall:	96.6	89.8

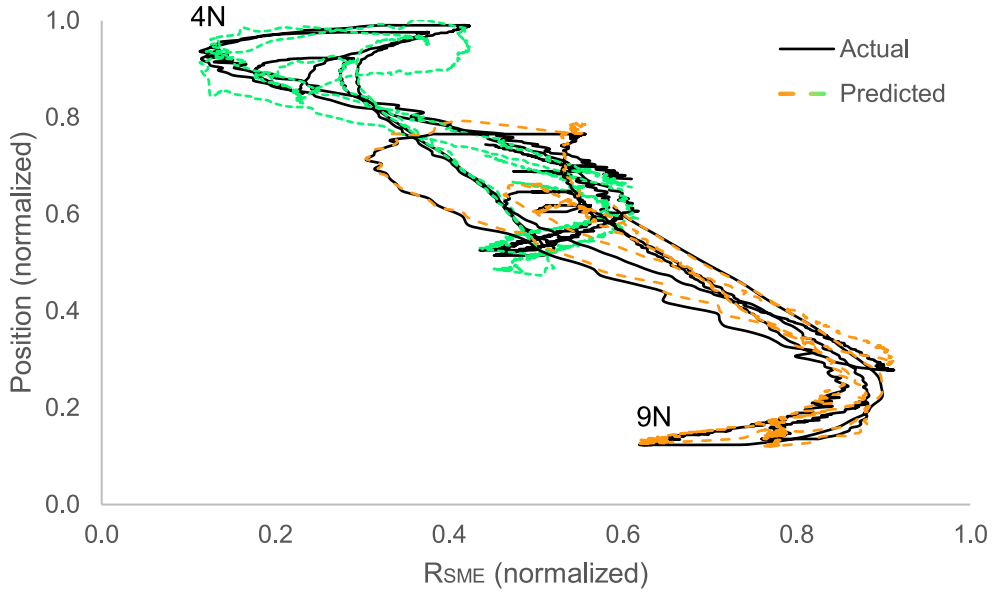

Figure 5.17: Position hysteresis curve estimation results on testing data with $4N$ and $9N$ applied force.

5.3 Position Control Using RNN Model Under Varying Applied Force

Using the model discussed in Section 5.2, the estimated position and force of the dual resistance NiTi actuator can be used to control the actuator's position under varying load by adjusting the applied current. In order to demonstrate the model's use in practice, a proportional integral derivative (PID) controller was implemented into the custom tensile tester control software. Figure 5.18 illustrates the PID controller used for position control under varying forces (the system disturbance). The controller calculated the error value $e(t)$ by subtracting the estimated from the desired position value. Following partial optimization, the PID controller constants used were $K_p = 2.0$, $K_i = 0.0005$, and $K_d = 0.001$. These constants resulted in relatively fast, stable actuator response.

The results of the RNN position estimator-based PID controller are shown in Figure 5.19. The controller attempted to accurately set the position of the NiTi actuator under varying applied forces by accordingly adjusting the applied current. It can be seen that the largest position control inaccuracies are observed when the force applied to the

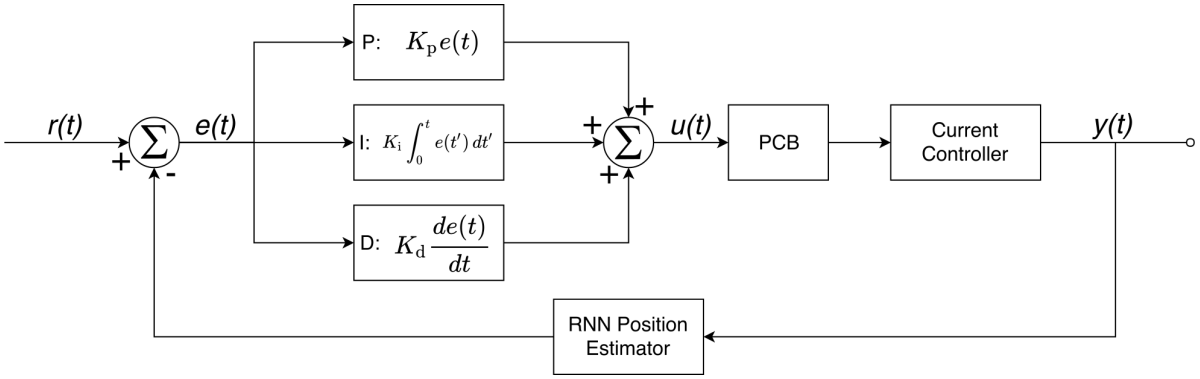

Figure 5.18: Schematic of the implemented PID controller.

actuator is changed. Furthermore, many of the inaccuracies result from the controller's inability to rapidly cool the wire. When the applied force is decreased, the controller responds by decreasing the applied current. Once the current reaches the minimum value of $0.15A$, which results in passive cooling dictated by conduction and convection, the cooling rate is maximized and the controller must wait until the desired actuator temperature is reached. This effect resulted in larger errors when the applied force was lowered. Given all of these inaccuracies, the model achieved accurate position control of the dual resistance NiTi actuator, resulting in an average position accuracy of 95.9%.

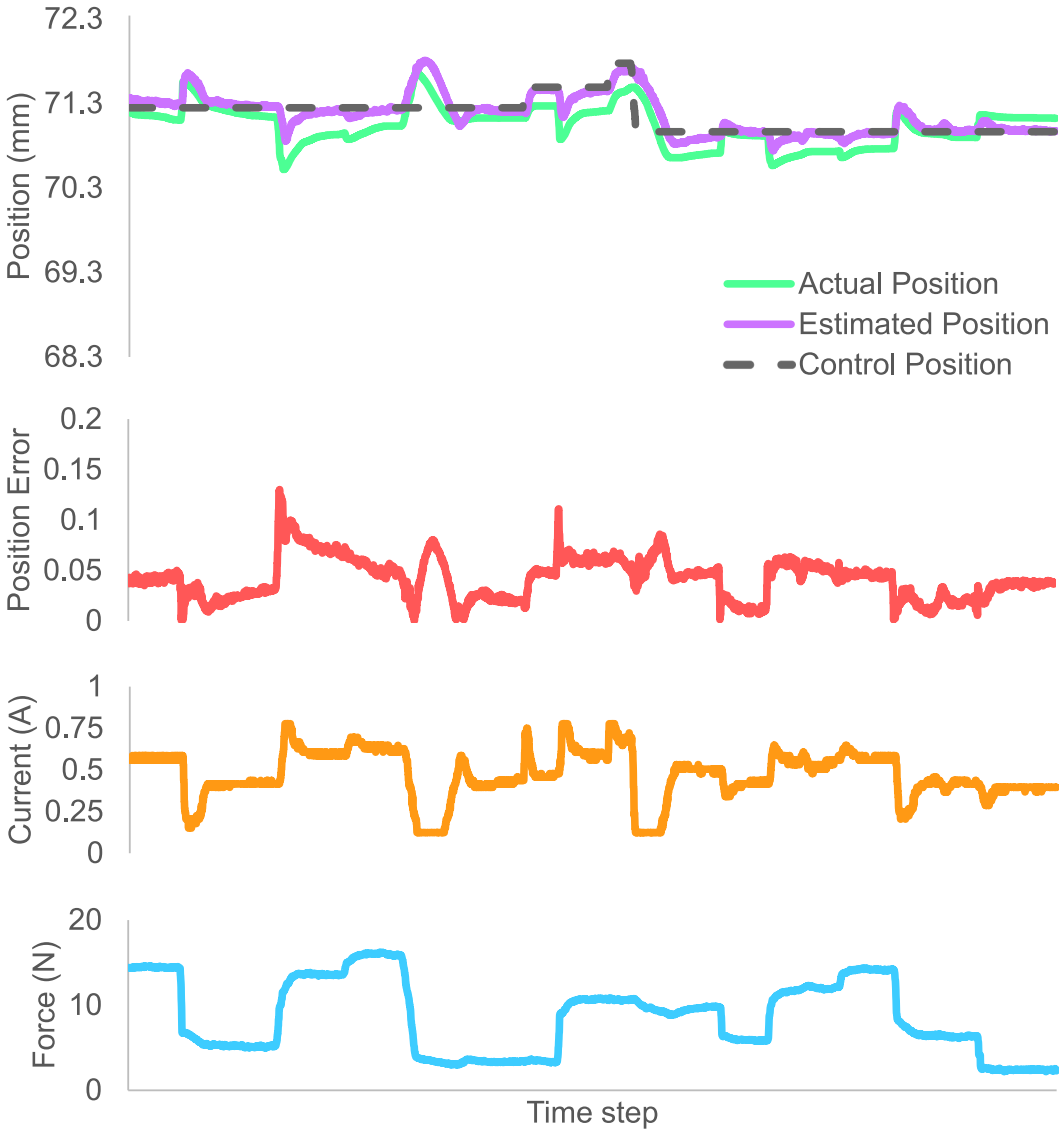

Figure 5.19: PID controller results using RNN position estimation model. Starting from top: actual, estimated, and control (desired) position during PID control, estimated vs. control position error, current applied by the controller, and force experienced by the actuator.

5.4 Chapter Summary

This chapter utilized the cycling data acquired in Chapter 4 in order to create a multiple memory SMA actuator control model based on recurrent neural networks for position and force estimation. The neural network architecture was optimized by individually examining six model hyperparameters, resulting in a final RNN with maximized achievable accuracy. This model was then used to successfully perform position estimation on an a multiple memory SMA actuator with constant applied load using only the input current and measured SME resistance as model inputs, resulting in prediction accuracy of 99.2%. Finally, the model was adapted to predict both position and force using a second input resistance across the PE section of the actuator, resulting in successful predictions with 98.5% and 96.0% overall estimation accuracy for training data position and force, respectively. The model also achieved 96.6% and 89.9% position and force estimation accuracy, respectively, on never before seen data, showing successful generalization and good understanding of the inherent nature of the actuator. The RNN model was successfully applied in a PID controller which yielded 95.9% position control accuracy under varying applied forces.

Chapter 6

Conclusions and Future Outlook

The following sections summarize the main conclusions resulting from this work and build on them by providing several recommendations for future work.

6.1 Conclusions

6.1.1 Thermomechanical Properties

During the thermomechanical processing, post-processing, training, and data acquisition of the NiTi actuators with two embedded memories, the following conclusions were reached:

1. Thermal characterization of the laser processed NiTi revealed significant increases in transformation temperature, with the final selected laser parameters caused A_f and M_s to increase by $77.6°C$ and $52.0°C$, respectively. Increases in pulse power and pulse time were also shown to cause transformation temperatures to rise.

2. Evaluation of UTS showed a deterioration in mechanical properties resulting from laser processing, with the NiTi processed using the selected laser parameters maintaining only 46% of the corresponding base metal UTS. Optical images of the wire's cross section revealed the presence of defects in the processed wire, most of which were shown to be eliminated by cold working. Thermal cycling was performed on the post-processed wire, showing positional actuation stabilization within 20 cycles.

3. Larger heat treatment temperatures increased transformation temperatures while lowering the detwinning stress plateau and UTS, confirming the trade-off between mechanical and thermal properties. Heat treatment temperature of $480°C$ resulted in a good balance between thermal and mechanical properties, increasing A_f and M_s by $61.5°C$ and $35.3°C$, respectively, while retaining 84.1% of UTS relative to base metal.

4. The base metal and processed NiTi sections were found to have drastically different mechanical and electrical characteristics during cycling caused by dissimilar phases at room temperature. The processed section showed full transformation in response to applied heat and load due to increased transformation temperatures, whereas the base metal only achieved partial transformation. This difference in properties confirmed the successful embedding of a second transformation memory into the monolithic actuator.

6.1.2 Neural Network Position and Force Estimation of Multiple Memory NiTi

Using the acquired NiTi cycling data, various recurrent neural network-based models were developed for predicting the position of the NiTi actuator with two embedded memories. The following conclusions can be made from analyzing the optimization and performance of the neural networks:

1. The GRU architecture was found to result in greatly improved performance relative to the LSTM architecture after equal training steps. LSTM proved to be unstable during training, requiring learning rate several orders of magnitude lower than that of GRU in order to successfully train. As a result, GRU resulted in lower MSE, faster convergence, and more stable training behaviour. Multiple hyperparameters were optimized for GRU, leading to a final RNN model with good training characteristics and excellent generalizability to unseen validation and testing data.

2. Using the optimized hyperparameters, an RNN model was trained on NiTi actuator cycling data at a constant load. The model successfully learned the actuator's hysteretic behaviour, including major and minor hysteresis loops. The final model achieved average position prediction accuracy of 99.2% across the entire data set.

3. The RNN with same architecture and hyperparameters was successfully trained to predict position and force under varying applied loads. The model achieved overall

accuracy of 98.5% and 96.0% for position and force estimation, respectively, across all training data sets. Performance evaluation was also performed using testing data never before seen by the model, resulting in accuracy of 96.6% and 89.8% for position and force estimation, respectively. Overall, an accurate RNN-based NiTi actuator position and force estimation model for varying loads was successfully developed. The RNN model was successfully applied by developing a PID position controller which achieved position accuracy of 95.9% under varying applied loads.

6.2 Future Work

Based on the results of this study, several recommendations for future research can be made:

1. This study achieved successful, repeatable fabrication of NiTi wires with two distinct embedded memories. However, due to cold working and heat treatment, the unprocessed section of the actuator also saw an increase in transformation temperatures. Further laser processing optimization should be performed in the future to minimize the required post-processing, reducing the magnitude of the base metal transformation temperature increase. Ideally, an actuator with two memories exhibiting full SME and full PE behaviour should be processed in order to better emphasize the difference in transformation properties between the two sections.

2. Rather than performing heat treatment on the full wire after cold working, methods (such as Joule heating) for selectively heat treating only the processed sections of the wire should be explored. By selectively heat treating the laser processed sections, the cold worked base metal sections will not recover their transformation properties and will therefore act like a regular material. As a result, the electrical resistance of the base metal will directly reflect the temperature of the material, whereas the processed section will exhibit rises and falls in resistance depending on transformation.

3. Although the RNN architectures and hyperparameters used were found to successfully perform position and force estimation on the multiple memory NiTi actuators, more complex models exist which were not thoroughly explored due to time constraints. More advanced architectures (such as multilayer GRU and LSTM neural networks), different optimizers (AdaGrad, Ftrl, RMSProp, Gradient Descent), and other advanced training strategies (batch normalization, simulated annealing) should

be explored as they may further improve the RNN performance. Furthermore, various other inputs (such as ambient temperature, wire geometry, and wire composition) can be introduced to the neural network to produce a more advanced model which works in diverse environments with any NiTi wire geometry and composition.

4. Using the estimated position and force values, along with past current and measured resistance values, an additional neural network can be developed which predicts the future position of the NiTi actuator. Position control systems for such a NiTi actuator would greatly benefit from this model by gaining the ability to predict and correct the actuator's position trajectory.

References

[1] W. J. Buehler, J. V. Gilfrich, and R. C. Wiley, "Effect of low-temperature phase changes on the mechanical properties of alloys near composition tini," *Journal of Applied Physics*, vol. 34, no. 5, pp. 1475–1477, 1963. [Online]. Available: https://doi.org/10.1063/1.1729603

[2] W. J. Buehler and F. E. Wang, "A summary of recent research on the nitinol alloys and their potential application in ocean engineering," *Ocean Engineering*, vol. 1, no. 1, pp. 105 – 120, 1968. [Online]. Available: http://www.sciencedirect.com/science/article/pii/002980186890019X

[3] Y. Furuya, "Design and material evaluation of shape memory composites," *Journal of Intelligent Material Systems and Structures*, vol. 7, no. 3, pp. 321 – 330, 1996. [Online]. Available: http://journals.sagepub.com/doi/10.1177/1045389X9600700313

[4] G. N. S. J. B. Donald J. Leo, Craig Weddle, "Vehicular applications of smart material systems," *Proc.SPIE*, vol. 3326, pp. 3326 – 3326 – 11, 1998. [Online]. Available: https://doi.org/10.1117/12.310625

[5] C. Bil, K. Massey, and J. E. Abdullah, "Wing morphing control with shape memory alloy actuators," *Journal of Intelligent Material Systems and Structures*, vol. 24, no. 7, pp. 879 – 898, 2013. [Online]. Available: http://journals.sagepub.com/doi/10.1177/1045389X12471866

[6] J. V. Humbeeck, "Non-medical applications of shape memory alloys," *Materials Science and Engineering: A*, vol. 273-275, pp. 134 – 148, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921509399002932

[7] M. M. Kheirikhah, S. Rabiee, and M. E. Edalat, "A review of shape memory alloy actuators in robotics," pp. 206–217, 2011.

[8] A. S. Mahmud, Y. Liu, and T. hyun Nam, "Gradient anneal of functionally graded niti," *Smart Materials and Structures*, vol. 17, no. 1, p. 015031, 2008. [Online]. Available: http://stacks.iop.org/0964-1726/17/i=1/a=015031

[9] Q. Meng, Y. Liu, H. Yang, B. S. Shariat, and T. hyun Nam, "Functionally graded niti strips prepared by laser surface anneal," *Acta Materialia*, vol. 60, no. 4, pp. 1658 – 1668, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1359645411008524

[10] M. I. Khan, A. Pequegnat, and Y. N. Zhou, "Multiple memory shape memory alloys," *Advanced Engineering Materials*, vol. 15, no. 5, pp. 386–393, 2013.

[11] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain." *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958. [Online]. Available: http://psycnet.apa.org/record/1959-09865-001

[12] L. Sun and W. M. Huang, "Nature of the multistage transformation in shape memory alloys upon heating," *Metal Science and Heat Treatment*, vol. 51, no. 11, pp. 573–578, Nov 2009. [Online]. Available: https://doi.org/10.1007/s11041-010-9213-x

[13] K. Otsuka, T. Sawamura, and K. Shimizu, "Crystal structure and internal defects of equiatomic tini martensite," *Physica Status Solidi*, vol. 5, no. 2, pp. 457–470, May 1971.

[14] K. Bhattacharya, "Microstructure of martensite. why it forms and how give rise to the shape-memory effect," pp. 1–208, 2003.

[15] S. Miyazaki and K. Otsuka, "Deformation and transition behavior associated with ther-phase in ti-ni alloys," *Metallurgical Transactions A*, vol. 17, no. 1, pp. 53–63, April 1985.

[16] J. Mohd Jani, M. Leary, A. Subic, and M. A. Gibson, "A review of shape memory alloy research, applications and opportunities," *Materials and Design*, vol. 56, pp. 1078–1113, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.matdes.2013.11.084

[17] G. B. Kauffmann and I. Mayo, "The story of nitinol: The serendipitous discovery of the memory metal and its applications," *The Chemical Educator*, vol. 2, no. 2, pp. 1–21, Jun 1997. [Online]. Available: https://doi.org/10.1007/s00897970111a

[18] P. Schlossmacher, T. Haas, and A. Schüssler, "Laser-welding of a ni-rich tini shape memory alloy : Mechanical behavior," *Journal de Physique Archives*, vol. 7, no. C5, pp. 251–256, November 1997.

[19] P. Schlossmacher, T. Haas, and A. Schussler, "Laser-welding of a ni-rich tini shape memory alloy: pseudoelastic properties," *Proceedings of the 2nd International Conference on Shape Memory and Superelastic Technologies*, pp. 137–142, March 1998.

[20] B. Tam, A. Pequegnat, M. Khan, and Y. Zhou, "Resistance microwelding of ti-55.8 wt pct ni nitinol wires and the effects of pseudoelasticity," *Metallurgical and Materials Transactions A*, vol. 43, no. 8, pp. 2969–2978, 03 2012.

[21] T. Massalski, H. Okamoto, P. Subramanian, and L. Kacprzak, *Binary Phase Diagrams*, 2nd ed. Materials Park, OH, USA: ASM International, 1990.

[22] K. Otsuka and C. Wayman, "Mechanism of shape memory effect and superelasticity," in *Shape memory materials*. Cambridge University Press, November 1998, pp. 27–48.

[23] T. Duerig, K. Melton, and D. Stockel, *Engineering Aspects of Shape Memory Alloys*. Oxford: Butterworth-Heinemann, 1990.

[24] K. Otsuka and X. Ren, "Physical metallurgy of Ti-Ni-based shape memory alloys," *Progress in Materials Science*, vol. 50, no. 5, pp. 511–678, 2005. [Online]. Available: http://dx.doi.org/10.1016/j.pmatsci.2004.10.001

[25] J. V. Humbeeck, "Shape Memory Alloys: AMaterial and a Technology," *Advanced Engineering Materials*, no. 11, pp. 837–850, 2001.

[26] W. Tang, "Thermodynamic study of the low-temperature phase b19 and the martensitic transformation in near-equiatomic ti-ni shape memory alloys," *Metallurgical and Materials Transactions A*, vol. 28, no. 3, pp. 537–544, Mar 1997. [Online]. Available: https://doi.org/10.1007/s11661-997-0041-6

[27] C. Urbina, S. De la Flor, and F. Ferrando, "Effect of thermal cycling on the thermomechanical behaviour of niti shape memory alloys," vol. 501, pp. 197–206, 02 2009.

[28] X. B. Ren and K. Otsuka, "Why does the martensitic transformation temperature strongly depend on composition?" in *Shape Memory Materials*, ser. Materials Science Forum, vol. 327. Trans Tech Publications, 1 2000, pp. 429–432.

[29] T. Simon, A. Kröger, C. Somsen, A. Dlouhy, and G. Eggeler, "On the multiplication of dislocations during martensitic transformations in niti shape memory alloys," *Acta Materialia*, vol. 58, no. 5, pp. 1850 – 1860, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1359645409008076

[30] A. R. Pelton, "Nitinol fatigue: A review of microstructures and mechanisms," *Journal of Materials Engineering and Performance*, vol. 20, no. 4, pp. 613–617, Jul 2011. [Online]. Available: https://doi.org/10.1007/s11665-011-9864-9

[31] D. C. Lagoudas, *Shape Memory Alloys: Modeling and Engineering Applications*. New York: Springer, 2008.

[32] S. Miyazaki, K. Otsuka, and Y. Suzuki, "Transformation pseudoelasticity and deformation behavior in a ti-50.6at%ni alloy," *Scripta Metallurgica*, vol. 15, no. 3, pp. 287 – 292, 1981. [Online]. Available: http://www.sciencedirect.com/science/article/pii/003697488190346X

[33] J. J. Zhang, Y. H. Yin, and J. Y. Zhu, "Electrical resistivity-based study of self-sensing properties for shape memory alloy-actuated artificial muscle." *Sensors (Basel, Switzerland)*, vol. 13, no. 10, pp. 12 958–12 974, 2013.

[34] S. Miyazaki, Y. Ohmi, K. Otsuka, and Y. Suzuki, "Characteristics of deformation and transformation pseudoelasticity in Ti-Ni Alloys," *Journal de Physique*, vol. 43, no. C4, pp. C4225–260, 1982.

[35] Z. Bo and D. C. Lagoudas, "Thermomechanical modeling of polycrystalline SMAs under cyclic loading, Part IV: Modeling of minor hysteresis loops," *International Journal of Engineering Science*, vol. 37, no. 9, pp. 1205–1249, 1999.

[36] D. R. Madill and D. Wang, "Modeling and L2-Stability of a Shape Memory Alloy Position Control System," *TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY*, vol. 6, no. 4, pp. 473–481, 1998.

[37] L. Chuntao and T. Yonghong, "A neural networks model for hysteresis nonlinearity," *Sensors and Actuators, A: Physical*, vol. 112, no. 1, pp. 49–54, 2004.

[38] H. Wang and G. Song, "Innovative NARX recurrent neural network model for ultra-thin shape memory alloy wire," *Neurocomputing*, vol. 134, pp. 289–295, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2013.09.050

[39] N. T. Tai and K. K. Ahn, "A hysteresis functional link artificial neural network for identification and model predictive control of SMA actuator," *Journal of Process Control*, vol. 22, no. 4, pp. 766–777, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.jprocont.2012.02.007

[40] a.a. Adly and S. Abd-El-Hafiz, "Using neural networks in the identification of Preisach-type\nhysteresis models," *IEEE Transactions on Magnetics*, vol. 34, no. 3, pp. 629–635, 1998.

[41] X. Dang and Y. Tan, "RBF neural networks hysteresis modelling for piezoceramic actuator using hybrid model," *Mechanical Systems and Signal Processing*, vol. 21, no. 1, pp. 430–440, 2007.

[42] N. Nikdel, P. Nikdel, M. A. Badamchizadeh, and I. Hassanzadeh, "Using neural network model predictive control for controlling shape memory alloy-based manipulator," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 3, pp. 1394–1401, 2014.

[43] X. Zhang, Y. Tan, M. Su, and Y. Xie, "Neural networks based identification and compensation of rate-dependent hysteresis in piezoelectric actuators," *Physica B: Condensed Matter*, vol. 405, no. 12, pp. 2687–2693, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.physb.2010.03.050

[44] D. Stoeckel, "The shape memory effect - phenomenon, alloys and applications." Shape Memory Alloys for Power Systems EPRI, 1995, pp. 1 – 13.

[45] K. Otsuka and K. Shimizu, "Pseudoelasticity and shape memory effects in alloys," vol. 31, pp. 93–114, 01 1986.

[46] T. Duerig and A. Pelton, "Ti-ni shape memory alloys," in *Materials Properties Handbook, Titanium Alloys*. Cambridge University Press, 1994, pp. 1035–1048.

[47] M. H. Elahinia, *Shape Memory Alloy Actuators: Design, Fabrication, and Experimental Evaluation*, 2016.

[48] Y. Zheng, F. Jiang, l. li, H. Yang, and Y. Liu, "Effect of ageing treatment on the transformation behaviour of ti–50.9at.736–745, 02 2008.

[49] K. Wada and Y. Liu, "On the two-way shape memory behavior in niti alloy—an experimental analysis," *Acta Materialia*, vol. 56, no. 13, pp. 3266 – 3277, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1359645408002048

[50] M. I. Khan, "Pulsed Nd:YAG Laser Processing of Nitinol," 2011.

[51] P. Sevilla, F. Martorell, C. Libenson, J. A. Planell, and F. Gil, "Laser welding of niti orthodontic archwires for selective force application," vol. 19, pp. 525–9, 03 2008.

[52] B. Panton, J. Oliveira, Z. Zeng, Y. Zhou, and M. Khan, "Thermomechanical fatigue of post-weld heat treated niti shape memory alloy wires," *International Journal of Fatigue*, vol. 92, pp. 1 – 7, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0142112316301566

[53] B. Panton, Z. Zeng, Y. Zhou, and M. Khan, "The effect of laser welds on the thermomechanical fatigue of niti shape memory alloys," *ASME Proceedings: Mechanics Behavior of Active Materials*, pp. 1–6, 09 2014.

[54] J. Hey and A. Jardine, "Shape memory tini synthesis from elemental powders," *Materials Science and Engineering: A*, vol. 188, no. 1, pp. 291 – 300, 1994.

[55] M. Whitney, S. Corbin, and R. Gorbet, "Investigation of the mechanisms of reactive sintering and combustion synthesis of niti using differential scanning calorimetry and microstructural analysis," *Acta Materialia*, vol. 56, no. 3, pp. 559 – 570, 2008.

[56] M. Daly, A. Pequegnat, Y. Zhou, and M. Khan, "Fabrication of a novel laser-processed niti shape memory microgripper with enhanced thermomechanical functionality," *Journal of Intelligent Material Systems and Structures*, vol. 24, no. 8, pp. 984–990, May 2012.

[57] A. Pequegnat, M. Daly, J. Wang, and M. Khan, "Dynamic actuation of a novel laser-processed niti linear actuator," *Smart Materials and Structures*, vol. 21, no. 9, pp. 1–7, 08 2012.

[58] A. Pequegnat, A. Michael, J. Wang, K. Lian, Y. Zhou, and M. Khan, "Surface characterizations of laser modified biomedical grade niti shape memory alloys," *Materials Science and Engineering: C, Materials for Biological Applications*, vol. 50, pp. 367–378, 05 2015.

[59] M. Jandaghi, P. Parvin, M. J. Torkamany, and J. Sabbaghzadeh, "Alloying element losses in pulsed nd : yag laser welding of stainless steel 316," *Journal of Physics D: Applied Physics*, vol. 41, no. 23, p. 235503, 2008. [Online]. Available: http://stacks.iop.org/0022-3727/41/i=23/a=235503

[60] A. Pequegnat, B. Panton, Y. N. Zhou, and M. I. Khan, "Local composition and microstructure control for multiple pseudoelastic plateau and hybrid self-biasing shape memory alloys," *Materials and Design*, vol. 92, pp. 802 – 813, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0264127515309370

[61] A. Pequegnat, M. Vlascov, M. Daly, Y. Zhou, and M. Khan, "Dynamic actuation of a multiple memory material processed nitinol linear actuator," vol. 1, pp. 1–6, 01 2011.

[62] B. Panton, Y. N Zhou, and M. Khan, "A stabilized, high stress self-biasing shape memory alloy actuator," vol. 25, p. 095027, 09 2016.

[63] A. Michael, Y. N. Zhou, and M. I. Khan, "Experimental validation of a one-dimensional model for monolithic shape memory alloys with multiple pseudoelastic plateaus," *Journal of Intelligent Material Systems and Structures*, vol. 27, no. 15, pp. 2102–2111, 2016. [Online]. Available: https://doi.org/10.1177/1045389X15620044

[64] B. Panton, A. Michael, Y. Zhou, and M. Khan, "Effects of post-processing on the thermomechanical fatigue properties of laser modified niti," *International Journal of Fatigue*, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0142112317304383

[65] S. Kalpakjian and S. Schmid, *Manufacturing Processes for Engineering Materials*, 5th ed. Jurong, Singapore: Prentice Hall Pearson Education, 2008.

[66] J. E. Schaffer and D. L. Plumley, "Fatigue performance of nitinol round wire with varying cold work reductions," *Journal of Materials Engineering and Performance*, vol. 18, no. 5, p. 563, Feb 2009. [Online]. Available: https://doi.org/10.1007/s11665-009-9363-4

[67] X. Lei, W. Rui, and L. Yong, "The optimization of annealing and cold-drawing in the manufacture of the ni–ti shape memory alloy ultra-thin wire," *The International Journal of Advanced Manufacturing Technology*, vol. 55, no. 9, pp. 905–910, Aug 2011. [Online]. Available: https://doi.org/10.1007/s00170-010-3116-2

[68] R. Wright, *Wire Technology: Process Engineering and Metallurgy: Second Edition*. Butterworth-Heinemann, 11 2010.

[69] K. Gall, J. Tyber, G. Wilkesanders, S. W. Robertson, R. O. Ritchie, and H. J. Maier, "Effect of microstructure on the fatigue of hot-rolled and cold-drawn niti shape memory alloys," *Materials Science and Engineering:*

A, vol. 486, no. 1, pp. 389 – 403, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921509307018242

[70] H. Aaronson, Y. Lee, and K. Russell, "Precipitation processes in solids." The Minerals, Metals & Materials Society, American Institute of Mining, Metallurgical, and Petroleum Engineers, 1978, pp. 31–86.

[71] C. A. L. Bailer-Jones, R. Gupta, and H. P. Singh, "An introduction to artificial neural networks," pp. 1–18, 2001. [Online]. Available: http://arxiv.org/abs/astro-ph/0102224

[72] M. M. Botvinick, M. M. Botvinick, D. C. Plaut, and D. C. Plaut, "Short-term memory for serial order: a recurrent neural network model," *Psychological review*, vol. 113, no. 2, pp. 201–33, 2006. [Online]. Available: http://doi.apa.org/getdoi.cfm?doi=10.1037/0033-295X.113.2.201{%}5Cnhttp://www.ncbi.nlm.nih.gov/pubmed/16637760

[73] J. Le, H. El-Askary, M. Allali, and D. Struppa, "Application of recurrent neural networks for drought projections in California," *Atmospheric Research*, vol. 188, pp. 100–106, 2017. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0169809517300157

[74] I. Basheer and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *Journal of Microbiological Methods*, vol. 43, no. 1, pp. 3 – 31, 2000, neural Computting in Micrbiology.

[75] A. Hagg, M. Mensing, and A. Asteroth, "Evolving parsimonious networks by mixing activation functions," *Neural and Evolutionary Computing*, pp. 1 – 8, 2017. [Online]. Available: https://arxiv.org/abs/1703.07122

[76] R. Vidal, J. Buna, R. Giryes, and S. Soatto, "Mathematics of deep learning," *Computer Vision and Pattern Recognition*, pp. 1 – 10, 2017. [Online]. Available: https://arxiv.org/pdf/1712.04741.pdf

[77] A. S. Sebag, M. Schoenauer, and M. Sebag, "Stochastic gradient descent: Going as fast as possible but not faster," *Machine Learning*, pp. 1 – 16, 2017. [Online]. Available: https://arxiv.org/pdf/1709.01427.pdf

[78] S. Ruder, "An overview of gradient descent optimization algorithms," *Computer Science Learning*, pp. 1 – 14, 2017. [Online]. Available: https://arxiv.org/pdf/1609.04747.pdf

[79] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," pp. 1–15, 1 2017. [Online]. Available: https://arxiv.org/pdf/1412.6980.pdf

[80] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural computation*, vol. 9, no. 8, pp. 1735–80, 1997. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/9377276

[81] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent Neural Network Regularization," *Iclr*, no. 2013, pp. 1–8, 2014. [Online]. Available: http://arxiv.org/abs/1409.2329

[82] C. Olah, "Understanding lstm networks," 2015. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[83] H. Sak, A. Senior, and F. Beaufays, "Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition," *Interspeech*, no. Cd, pp. 338–342, 2014. [Online]. Available: http://arxiv.org/abs/1402.1128

[84] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *arXiv*, pp. 1–9, 2014.

[85] K. Irie, Z. Tuske, T. Alkhouli, R. Schluter, and H. Ney, "LSTM, GRU, highway and a bit of attention: An empirical overview for language modeling in speech recognition," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 08-12-Sept, pp. 3519–3523, 2016.

[86] A. Michael, Y. Zhou, and M. Khan, "Novel method to analyse tensile properties of ultra-fine niti wires with a visual extensometer," *Materials Letters*, vol. 182, pp. 177 – 180, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167577X16310345

[87] R. Wright, "Center bursts - a review of criteria." Conference Proceedings,Wire Association International, Inc., 2008, p. 15.

[88] F. Jiang, Y. Liu, H. Yang, L. Li, and Y. Zheng, "Effect of ageing treatment on the deformation behaviour of Ti-50.9 at.% Ni," *Acta Materialia*, vol. 57, no. 16, pp. 4773–4781, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.actamat.2009.06.059

[89] J. E. Schaffer and D. L. Plumley, "Fatigue performance of nitinol round wire with varying cold work reductions," *Journal of Materials Engineering and Performance*, vol. 18, no. 5-6, pp. 563–568, 2009.

[90] A. R. Pelton, J. Dicello, and S. Miyazaki, "Optimisation of processing and properties of medical grade Nitinol wire," *Minimally Invasive Therapy & Allied Technologies*, vol. 9, no. 2, pp. 107–118, 2000. [Online]. Available: http://www.tandfonline.com/doi/full/10.3109/13645700009063057

[91] Y. Motemani, M. Nili-Ahmadabadi, M. Tan, M. Bornapour, and S. Rayagan, "Effect of cooling rate on the phase transformation behavior and mechanical properties of ni-rich niti shape memory alloy," *Journal of Alloys and Compounds*, vol. 469, no. 1, pp. 164 – 168, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925838808002120

[92] M. Drexel, G. Selvaduray, and A. Pelton, "The effects of cold work and heat treatment on the properties of nitinol wire," *Proceedings of the international conference on shape memory and superelastic technologies*, pp. 447–454, 2006. [Online]. Available: http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1592796

[93] S. Eucken and T. Duerig, "The effects of pseudoelastic prestraining on the tensile behaviour and two-way shape memory effect in aged niti," *Acta Metallurgica*, vol. 37, no. 8, pp. 2245 – 2252, 1989. [Online]. Available: http://www.sciencedirect.com/science/article/pii/000161608990151X

[94] S. Miyazaki, T. Imai, Y. Igo, and K. Otsuka, "Effect of cyclic deformation on the pseudoelasticity characteristics of ti-ni alloys," *Metallurgical Transactions A*, vol. 17, no. 1, pp. 115–120, Jan 1986. [Online]. Available: https://doi.org/10.1007/BF02644447

[95] X. Sun, W. Sun, S. Ma, X. Ren, Y. Zhang, W. Li, and H. Wang, "Complex structure leads to overfitting: A structure regularization decoding method for natural language processing," *Computation and Language*, pp. 1 – 37, 2017. [Online]. Available: https://arxiv.org/pdf/1711.10331.pdf

[96] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: https://www.tensorflow.org/

[97] X. Sierra-Canto, F. Madera-Ramirez, and V. Uc-Cetina, "Parallel training of a back-propagation neural network using cuda." 2010 Ninth International Conference on Machine Learning and Applications, 12 2010.

[98] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer-Verlag, 2009.

[99] Q. Zhang and S. Sun, "Weighted data normalization based on eigenvalues for artificial neural network classification," pp. 1–8, 12 2017. [Online]. Available: https://arxiv.org/pdf/1712.08885.pdf

[100] N. Zamani, M. Behrad, and M. Ibraheem, "Sensors and Actuators A : Physical Novel laser processed shape memory alloy actuator design with an embedded strain gauge sensor using dual resistance measurements . Part I : Fabrication and model-based position estimation," *Sensors & Actuators: A. Physical*, vol. 263, pp. 234–245, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.sna.2017.03.033

[101] R. N. Wright, *Wire Technology Process Engineering and Metallurgy*. Burlington, MA: Butterworth-Heinemann, 2011.

Appendices

Appendix A

Custom Tensile Tester Images

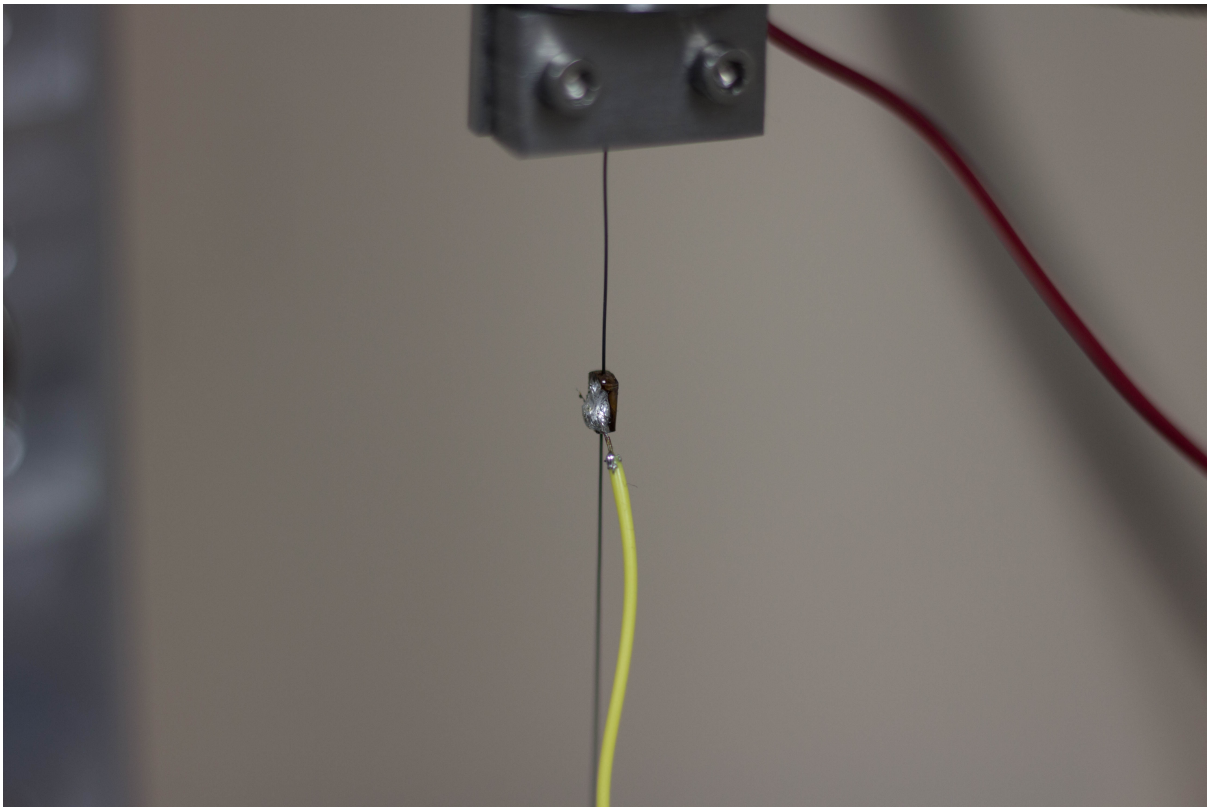

Figure A.1: Image of custom tensile tester showing copper crimp on NiTi actuator with PE and SME sections above and below the crimp, respectively.

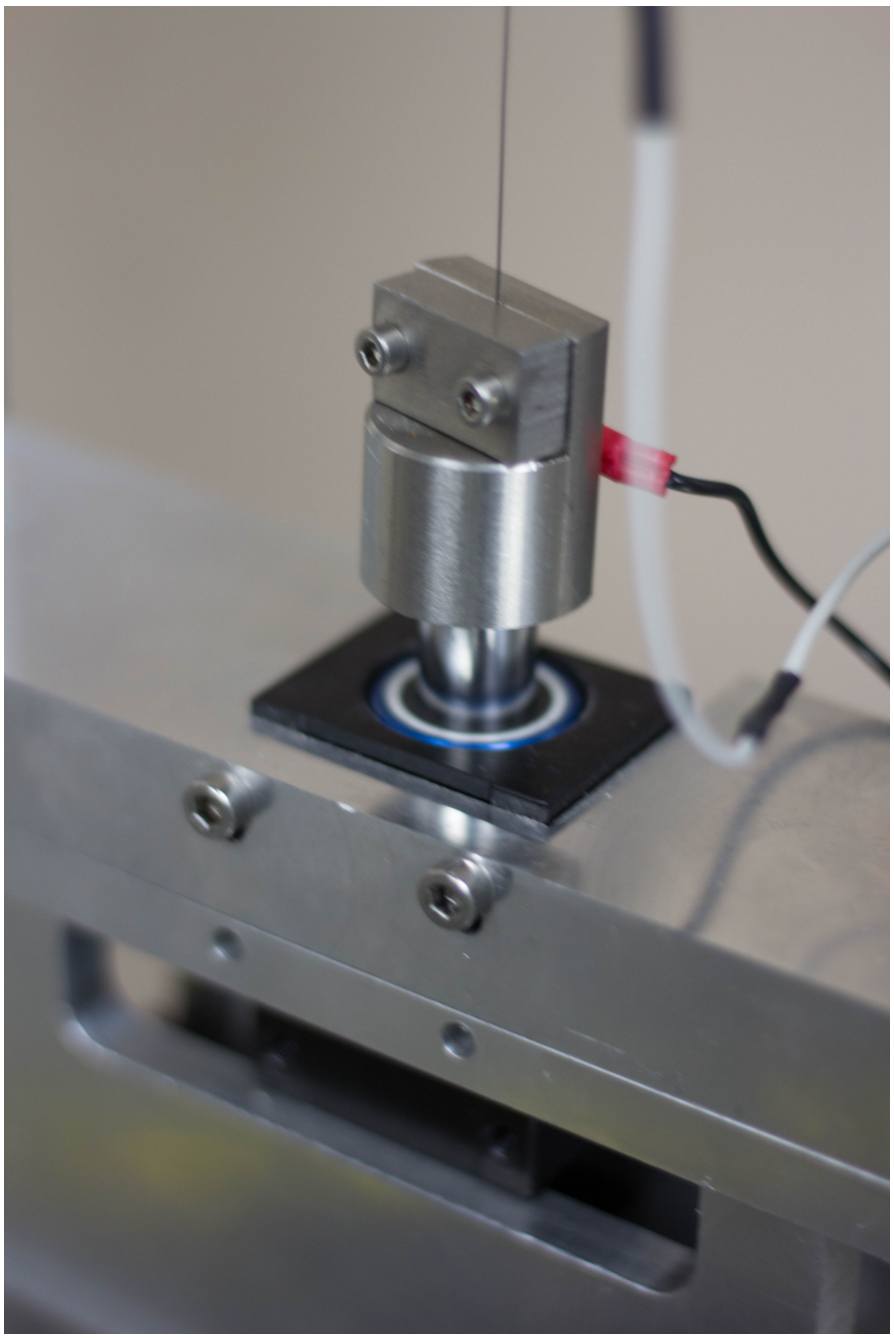

Figure A.2: Image of custom tensile tester showing air bushing with bottom clamp attached to the shaft.

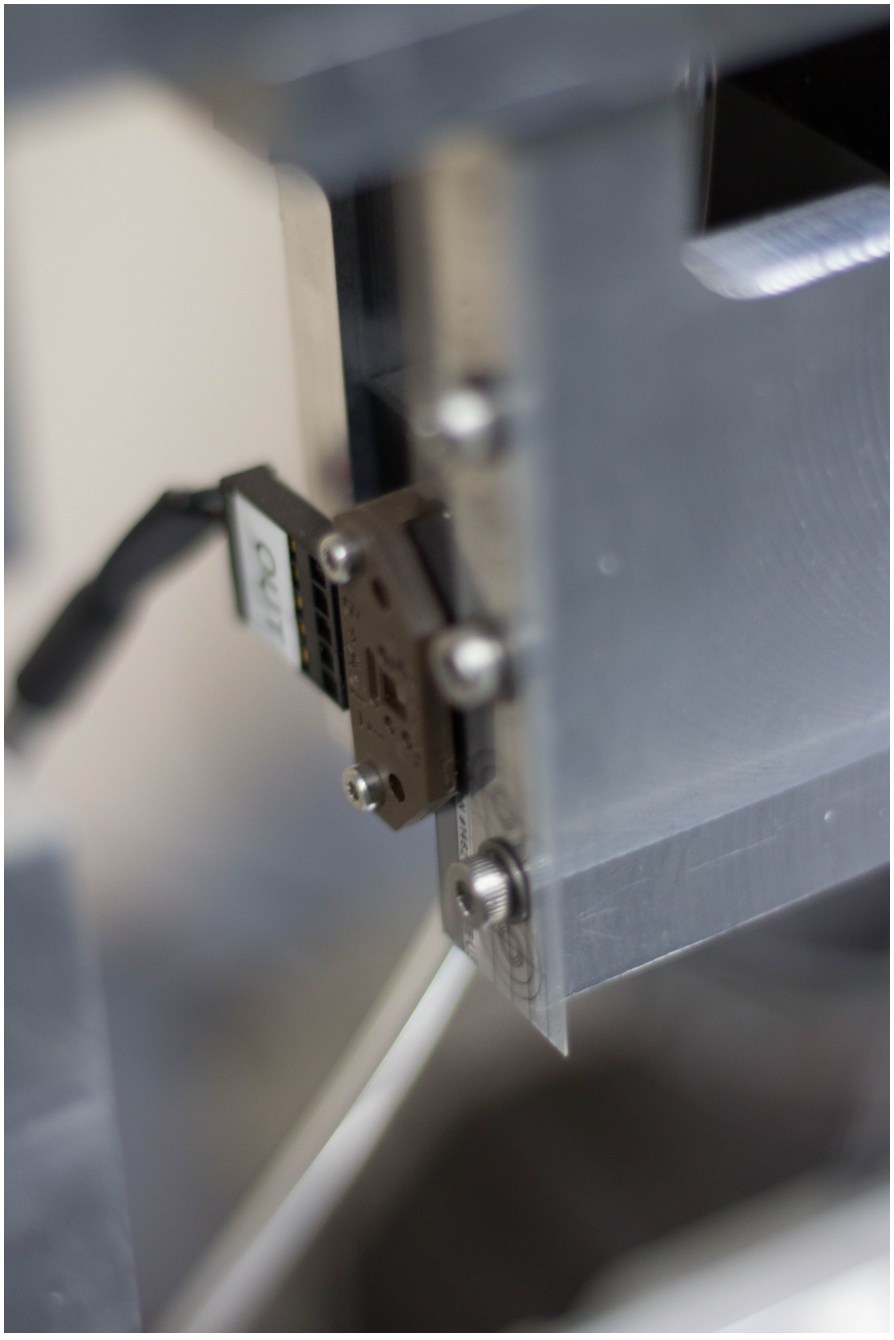

Figure A.3: Image of custom tensile tester showing optical encoder and corresponding encoder strip.

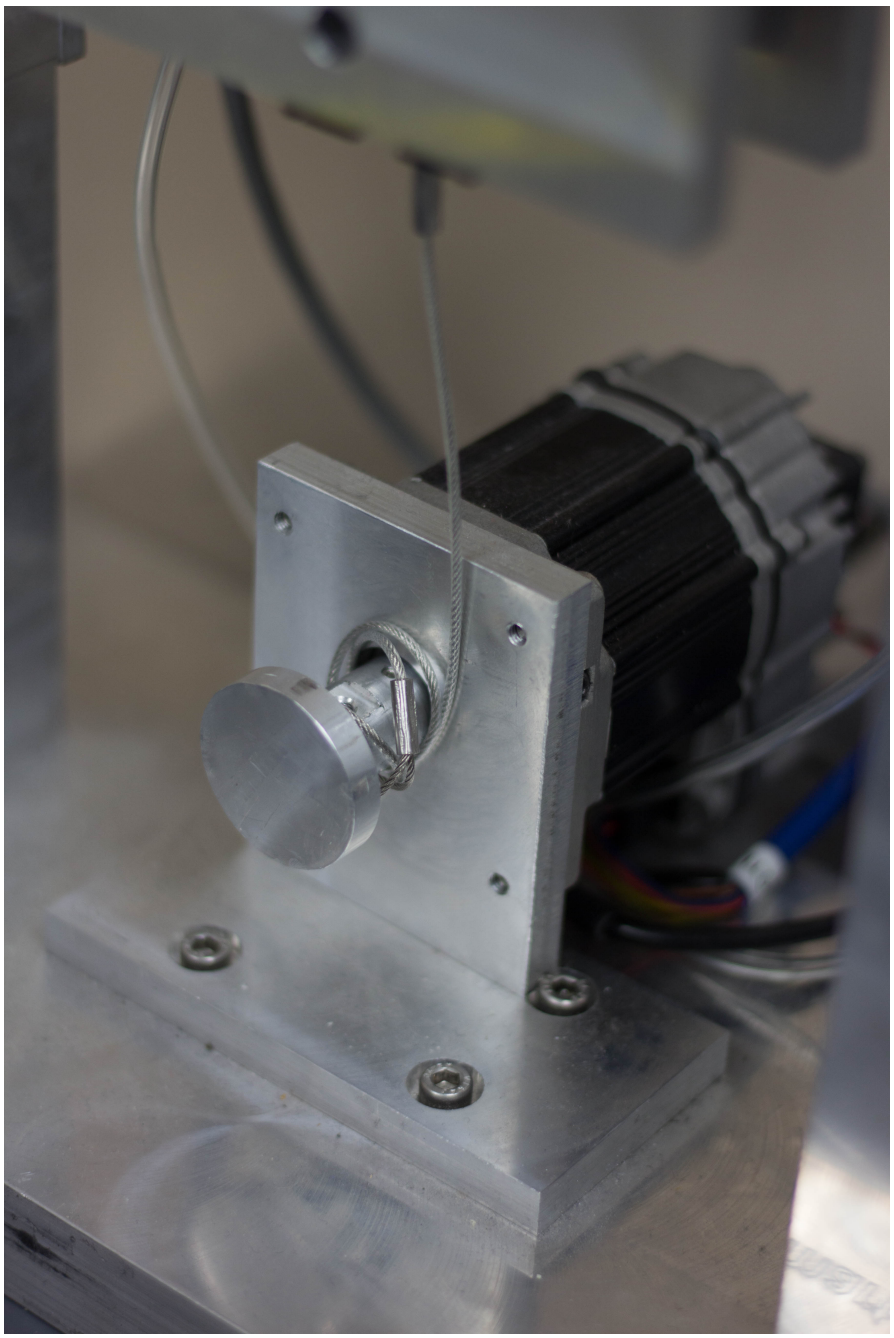

Figure A.4: Image of custom tensile tester showing load-applying servo connected to the bottom shaft.

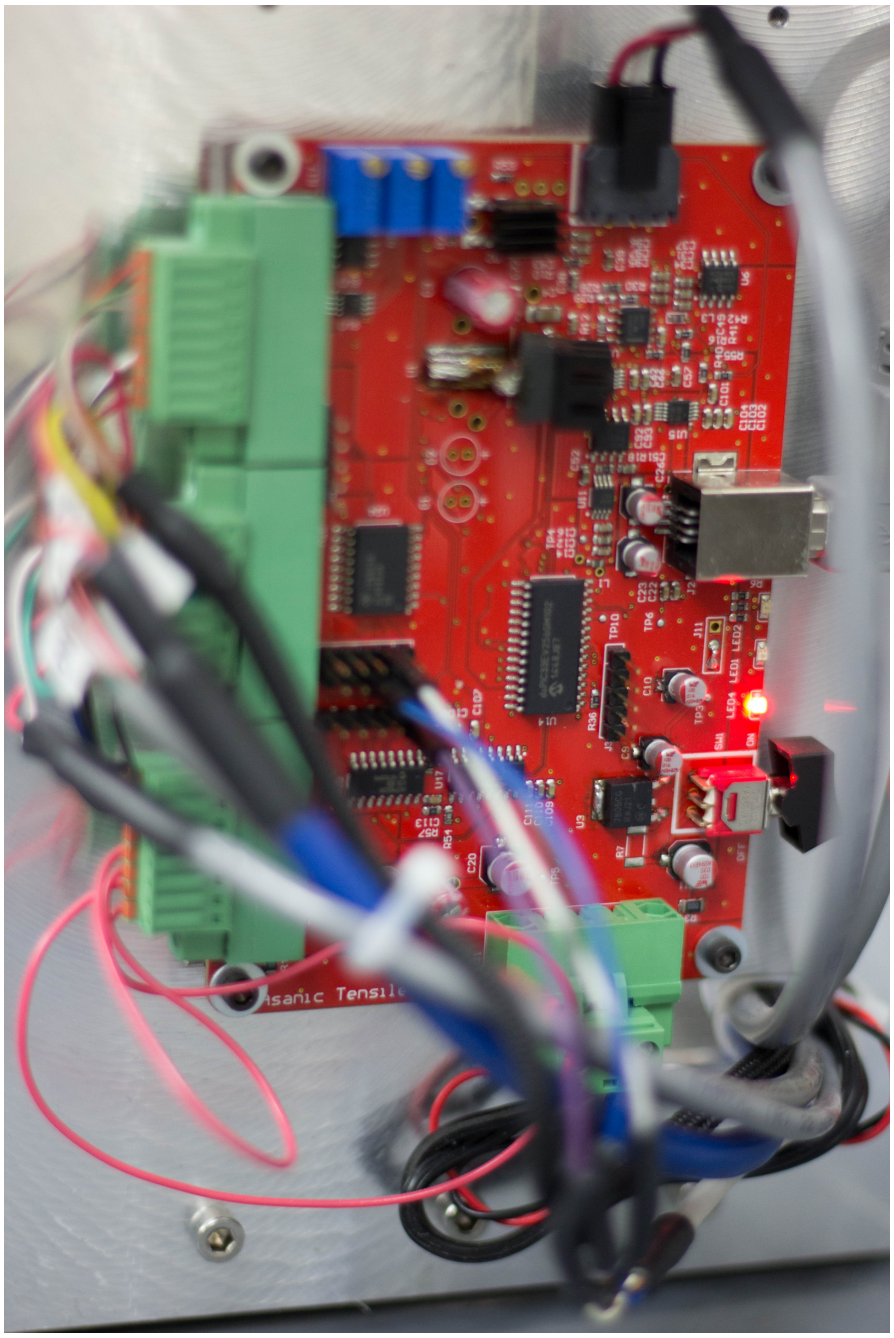

Figure A.5: Image of custom tensile tester control board.

Appendix B

RNN Training Performance on Constant Load Data

B.1 Activation Functions

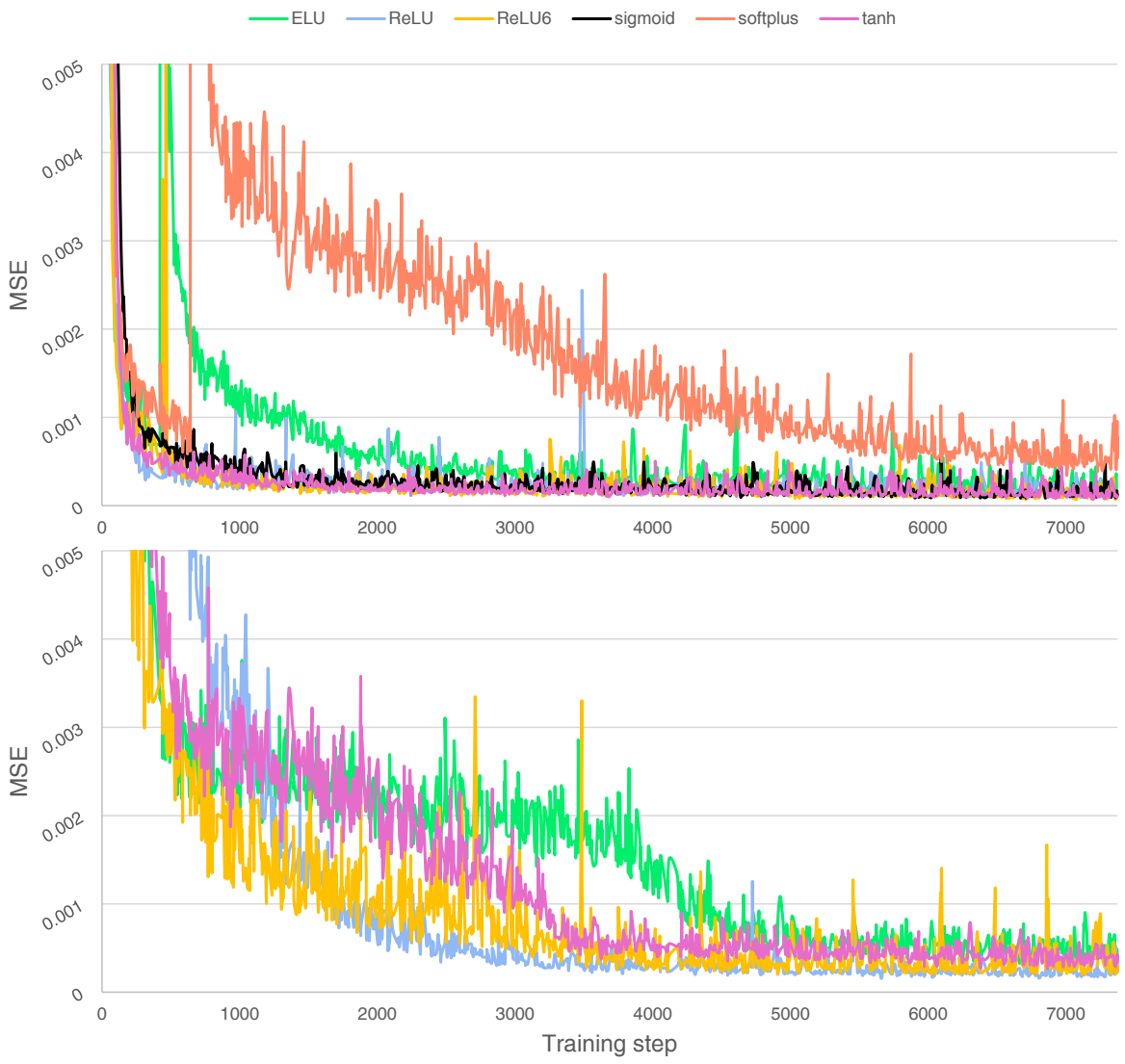

Figure B.1: Training curves for GRU (top) and LSTM (bottom) RNN architectures using various activation functions.

B.2 Batch Sizes

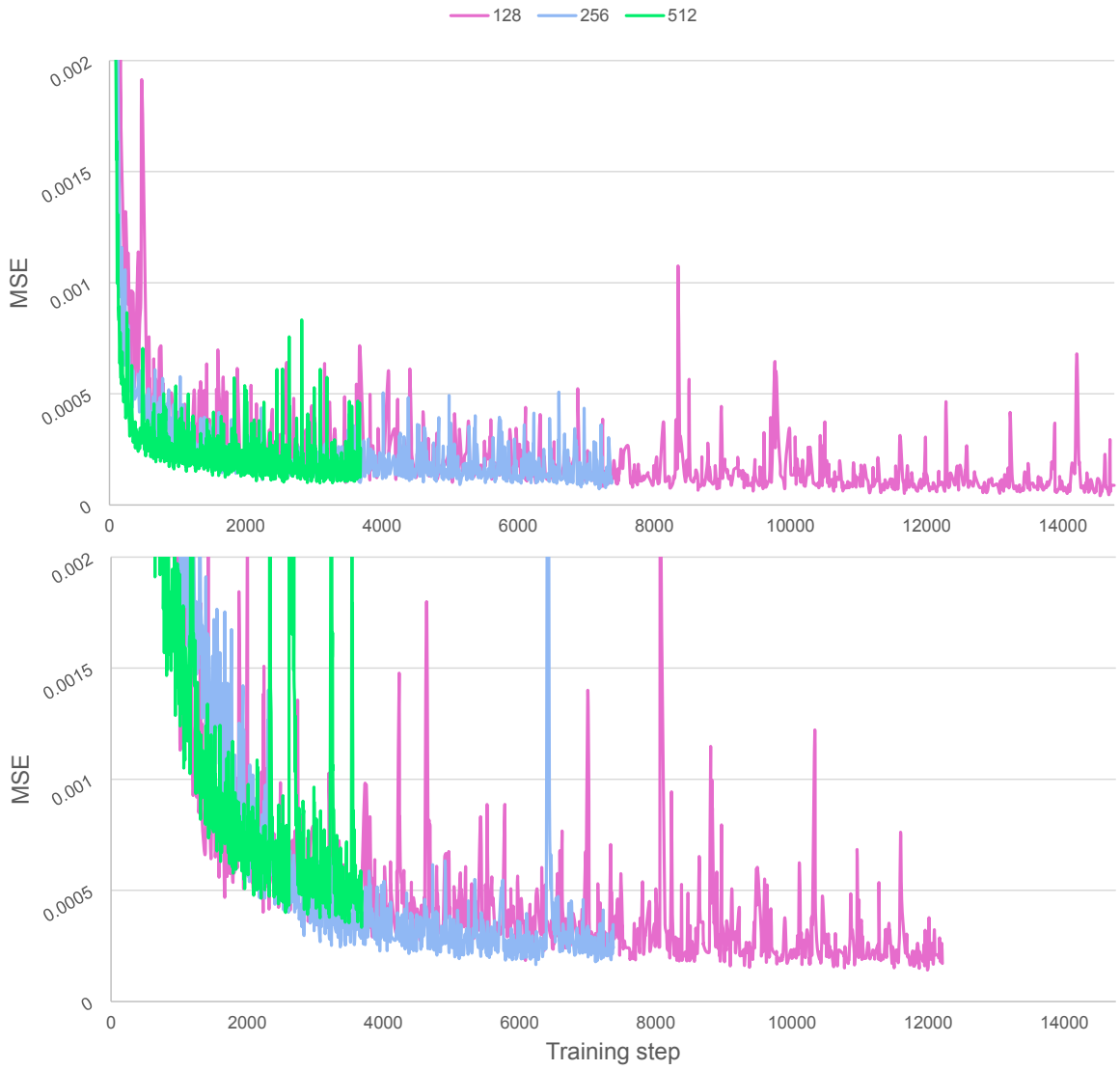

Figure B.2: Training curves for GRU (top) and LSTM (bottom) RNN architectures using various batch sizes.

B.3 Number of Hidden RNN States

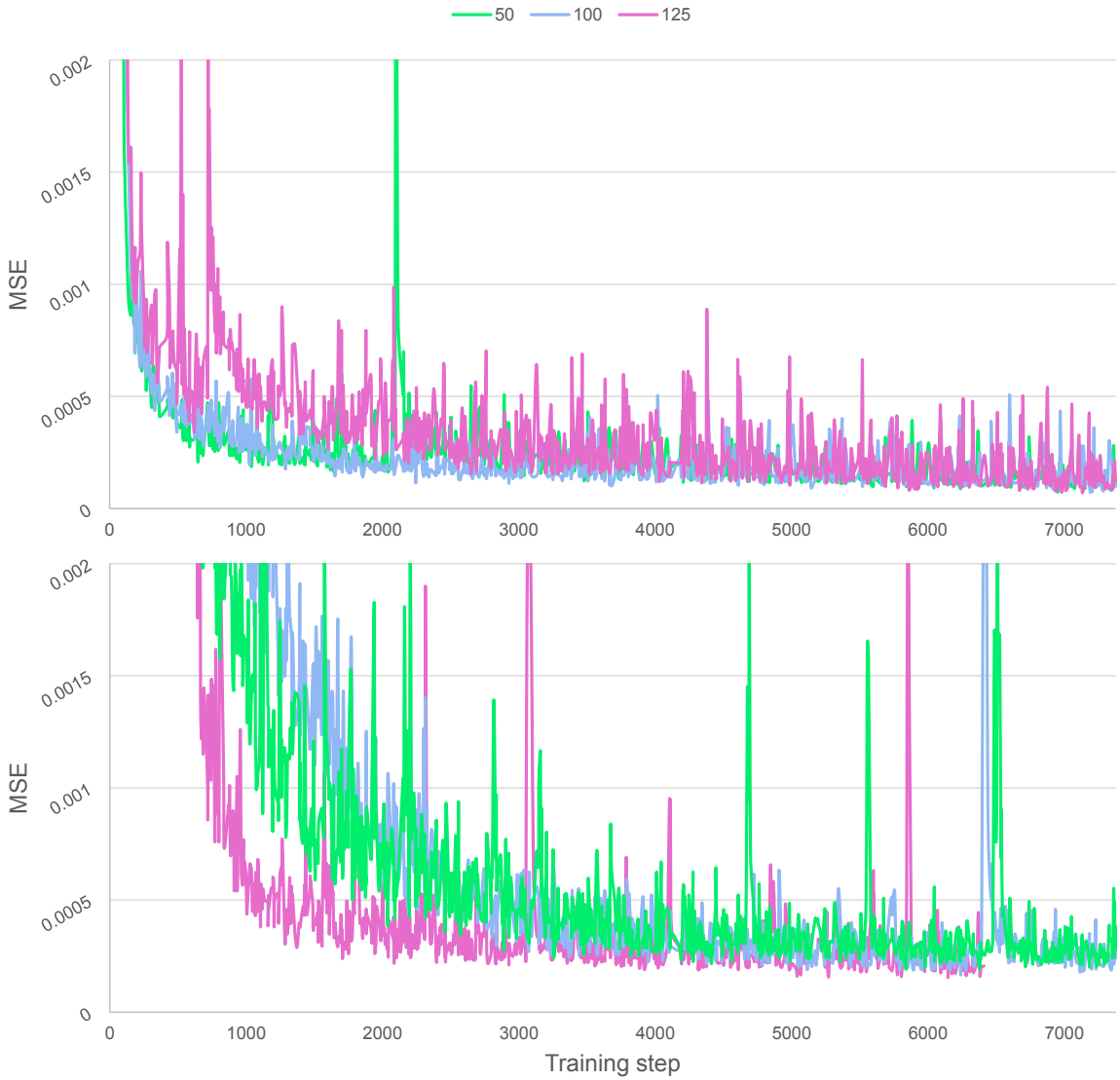

Figure B.3: Training curves for GRU (top) and LSTM (bottom) RNN architectures using varying numbers of hidden RNN states.

B.4 Number of Epochs

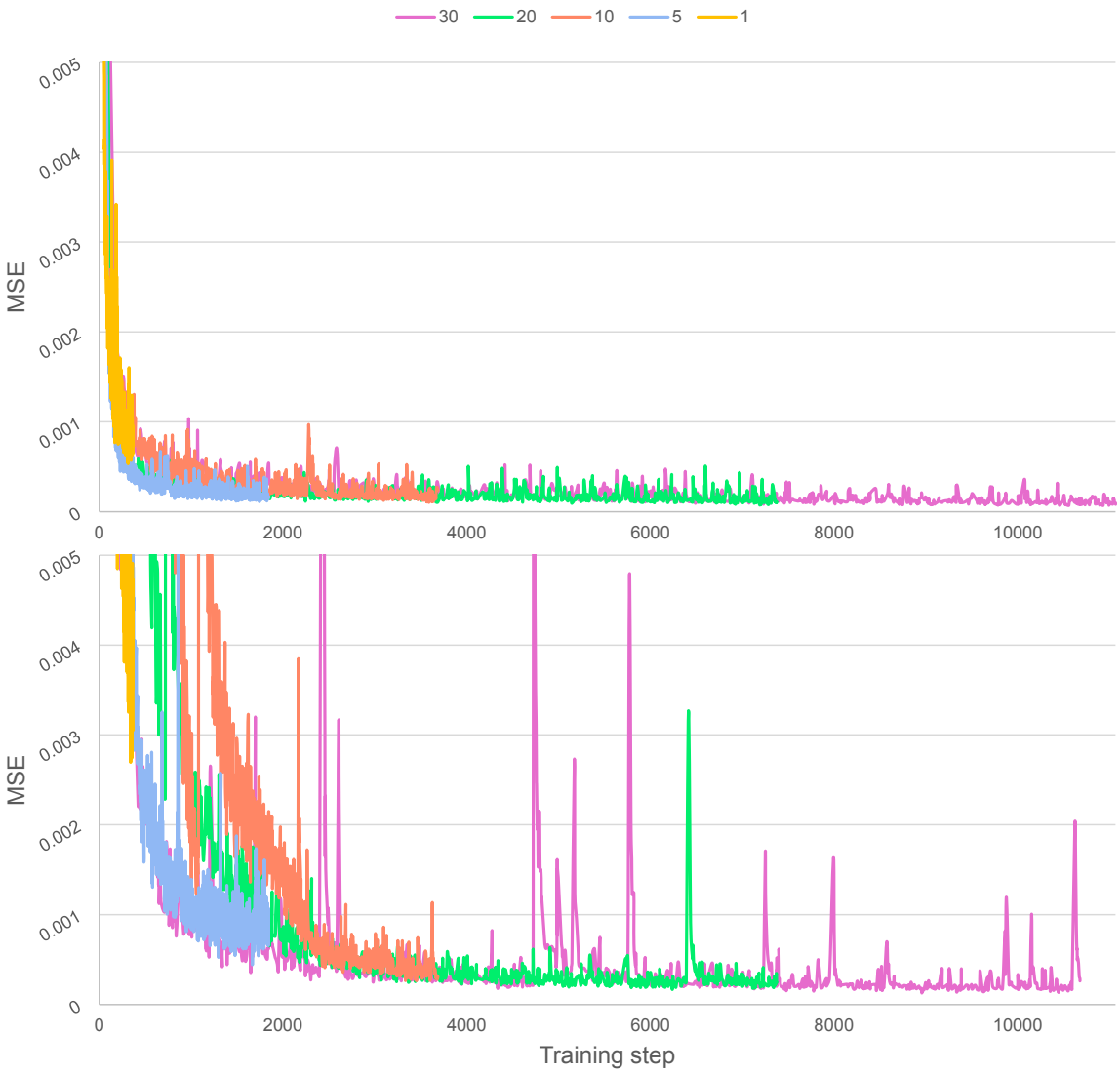

Figure B.4: Training curves for GRU (top) and LSTM (bottom) RNN architectures trained for varying numbers of epochs.

B.5 Levels of Sparsity

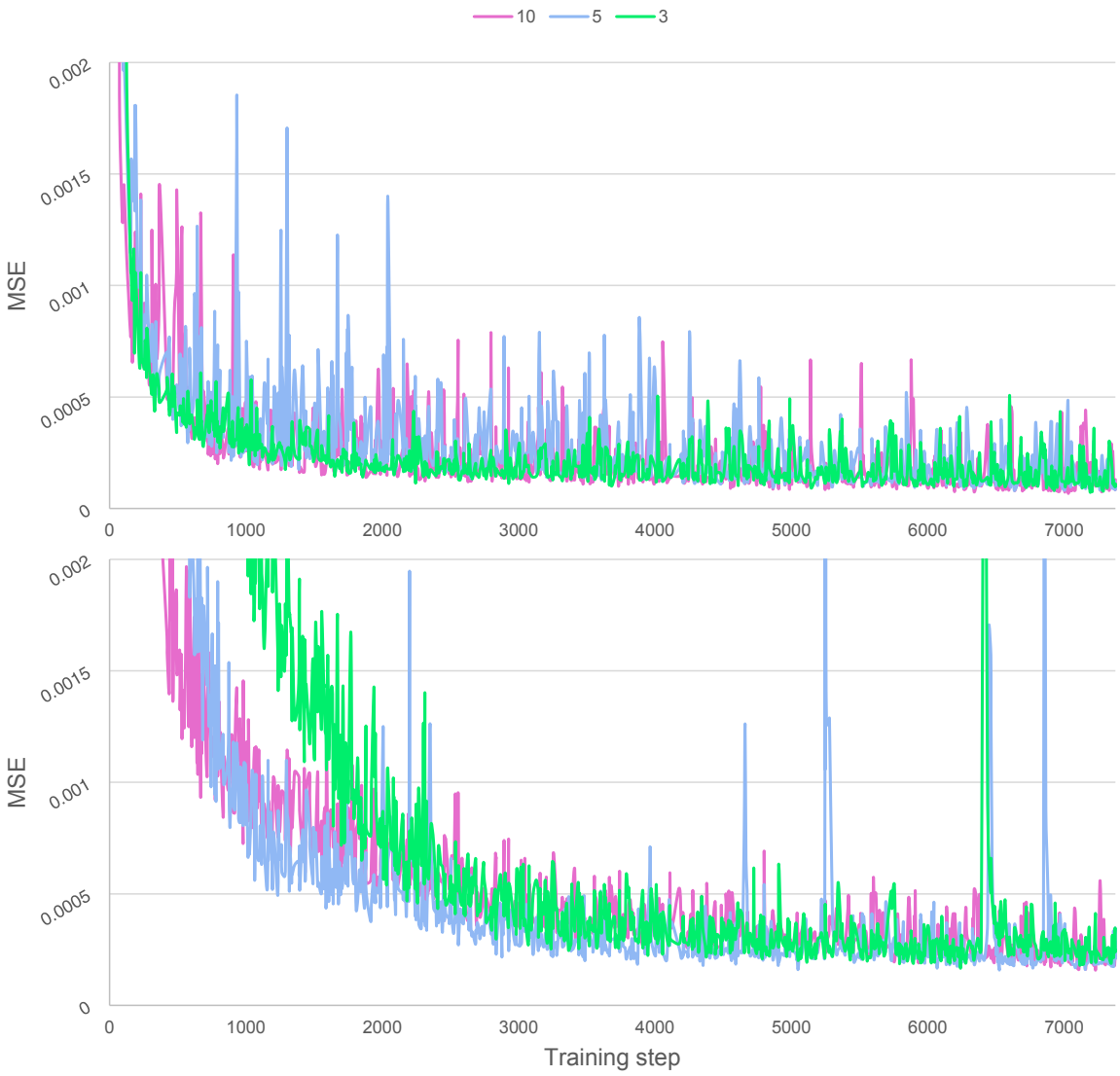

Figure B.5: Training curves for GRU (top) and LSTM (bottom) RNN architectures using various levels of look back sparsity.

B.6 Look Back Length

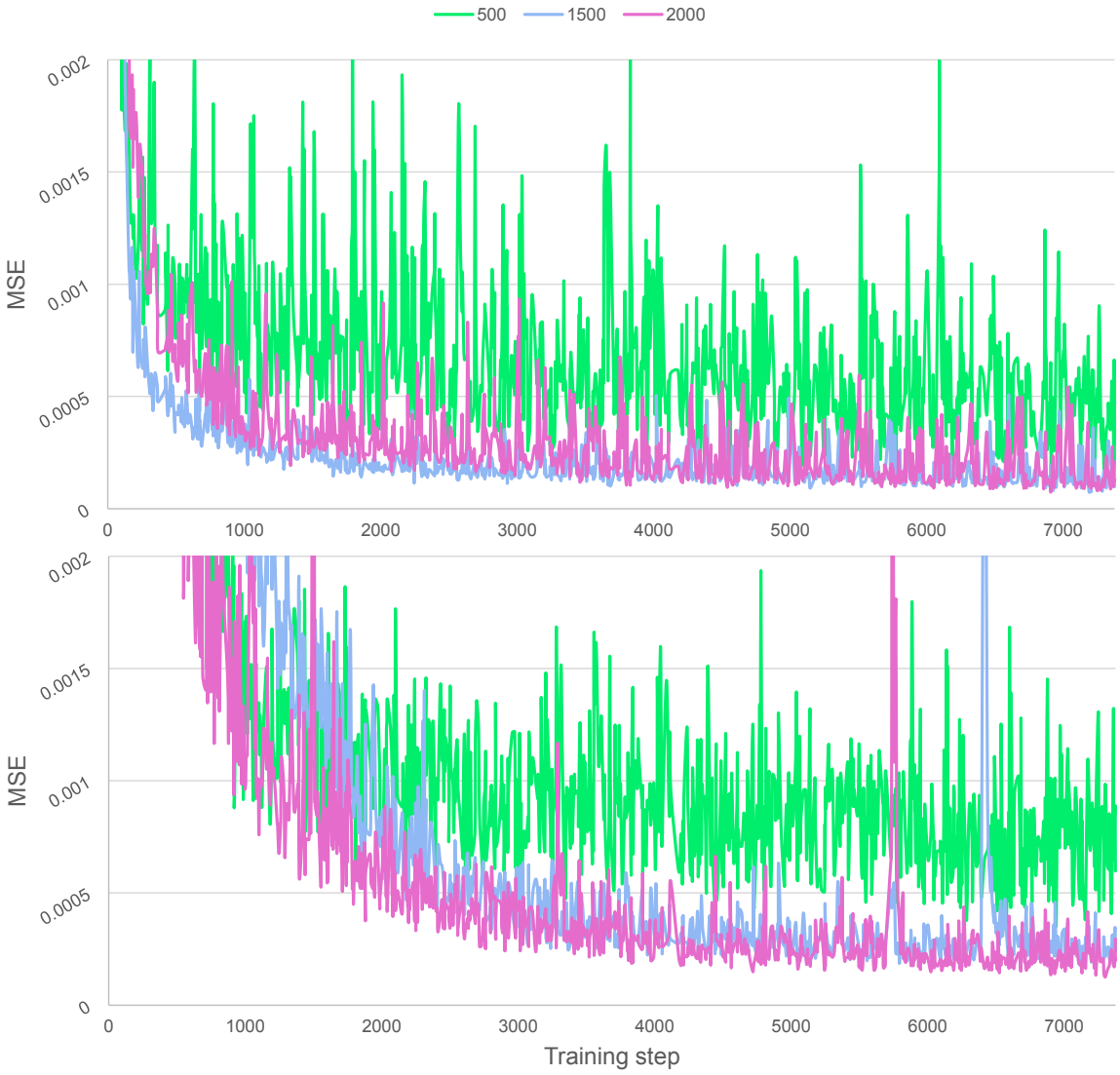

Figure B.6: Training curves for GRU (top) and LSTM (bottom) RNN architectures using varying look back lengths.

Appendix C

RNN Validation Performance on Constant Load Data

C.1 Activation Functions

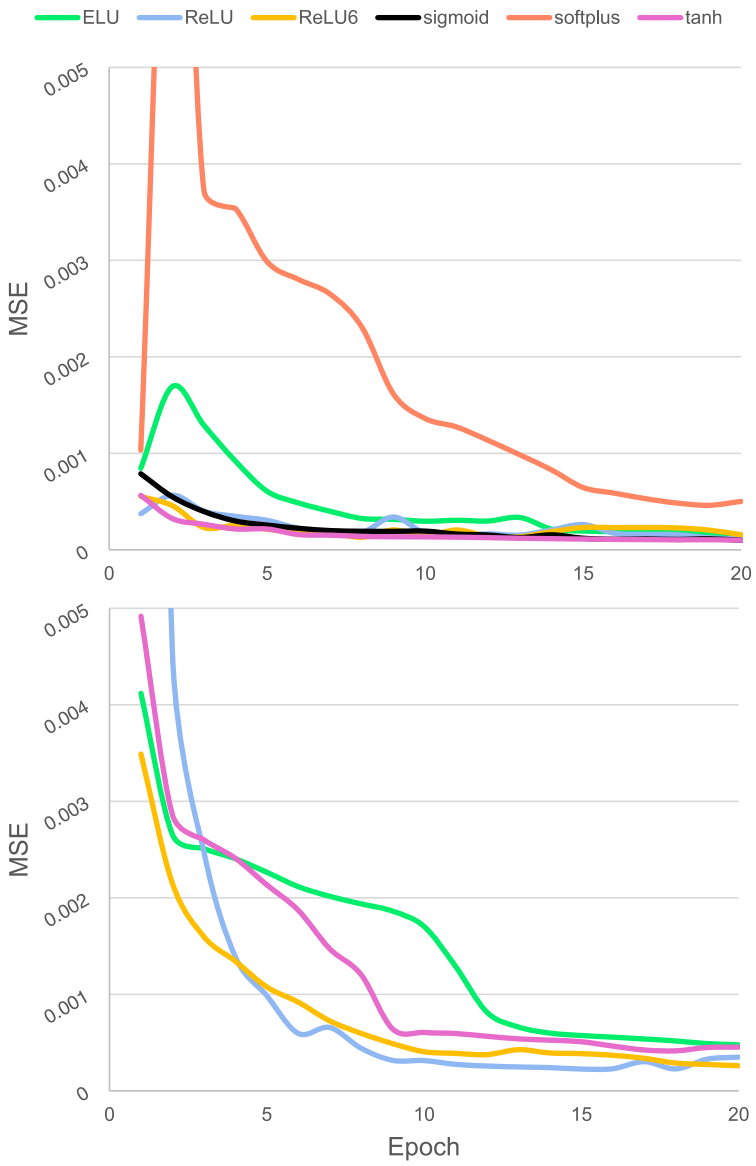

Figure C.1: Validation MSE curves during training for GRU (top) and LSTM (bottom) architectures using various activation functions.

C.2 Batch Sizes

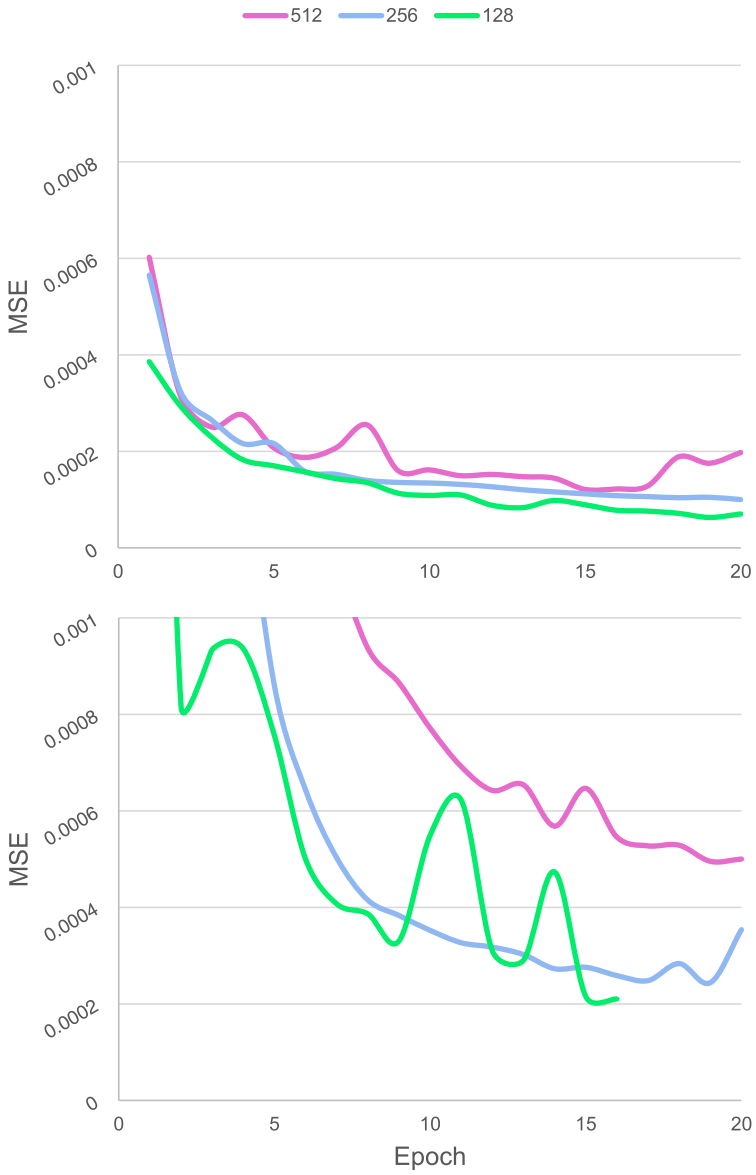

Figure C.2: Validation MSE curves during training for GRU (top) and LSTM (bottom) architectures using varying batch sizes.

C.3 Number of Hidden RNN States

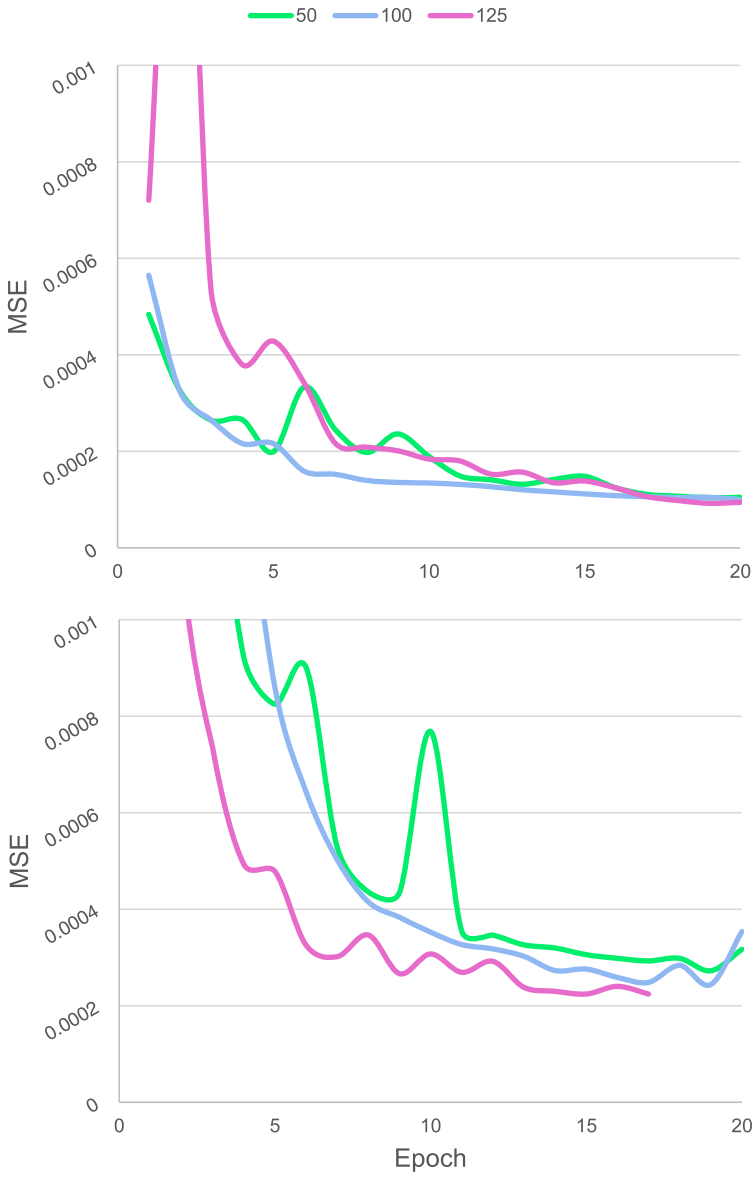

Figure C.3: Validation MSE curves during training for GRU (top) and LSTM (bottom) architectures using varying numbers of hidden RNN states.

C.4 Number of Epochs

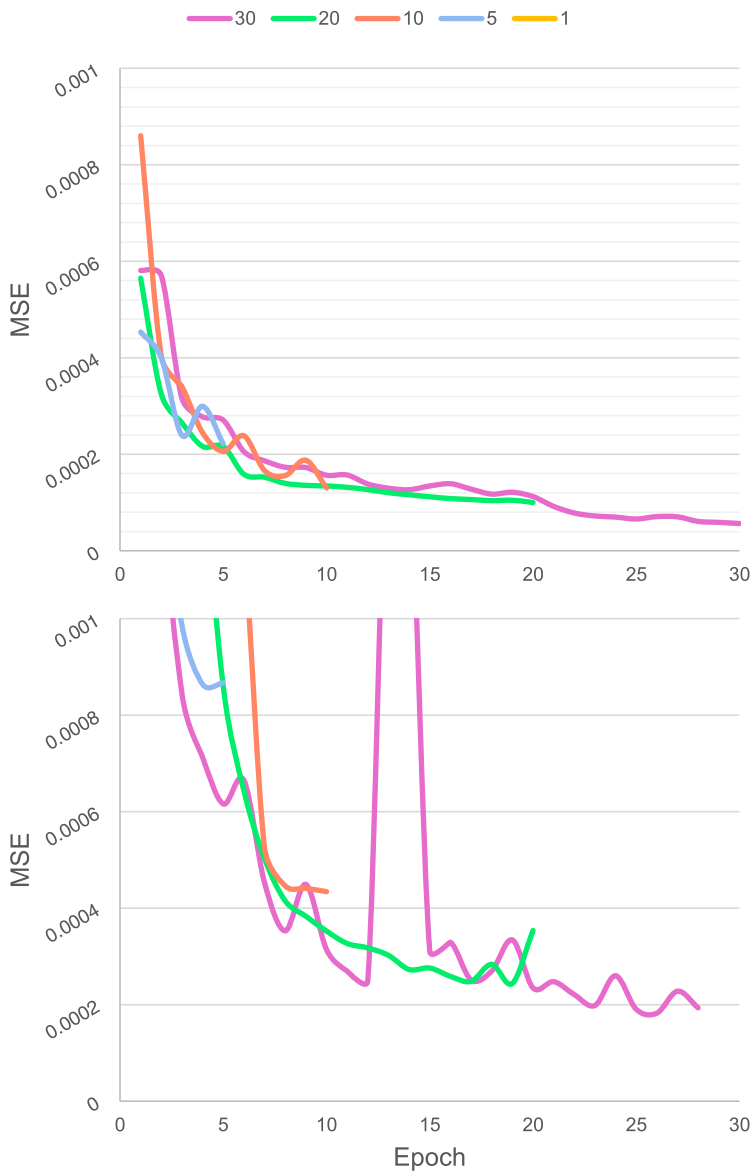

Figure C.4: Validation MSE curves during training for GRU (top) and LSTM (bottom) architectures using varying numbers of training epochs.

C.5 Levels of Sparsity

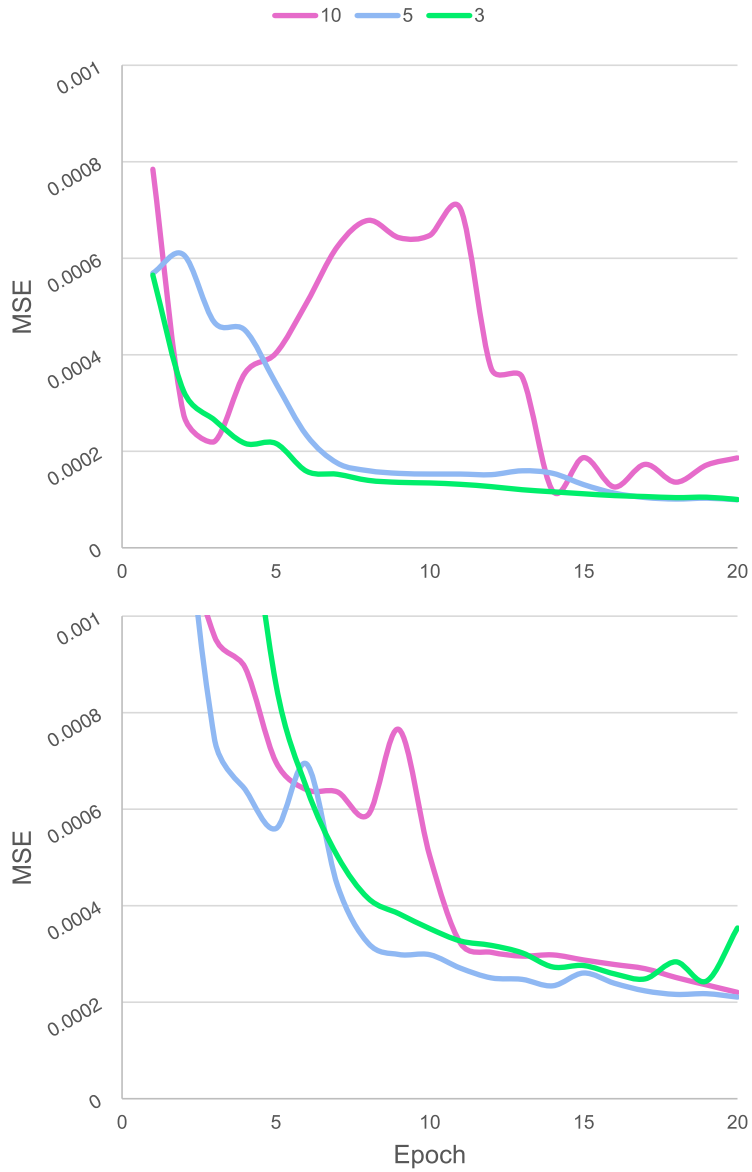

Figure C.5: Validation MSE curves during training for GRU (top) and LSTM (bottom) architectures using varying levels of look back sparsity.

C.6 Look Back Length

Figure C.6: Validation MSE curves during training for GRU (top) and LSTM (bottom) architectures using varying look back lengths.

Appendix D

Applied Current and Measured Resistance for RNN Model Estimation Data

D.1 Position and Force Estimation on Training Data

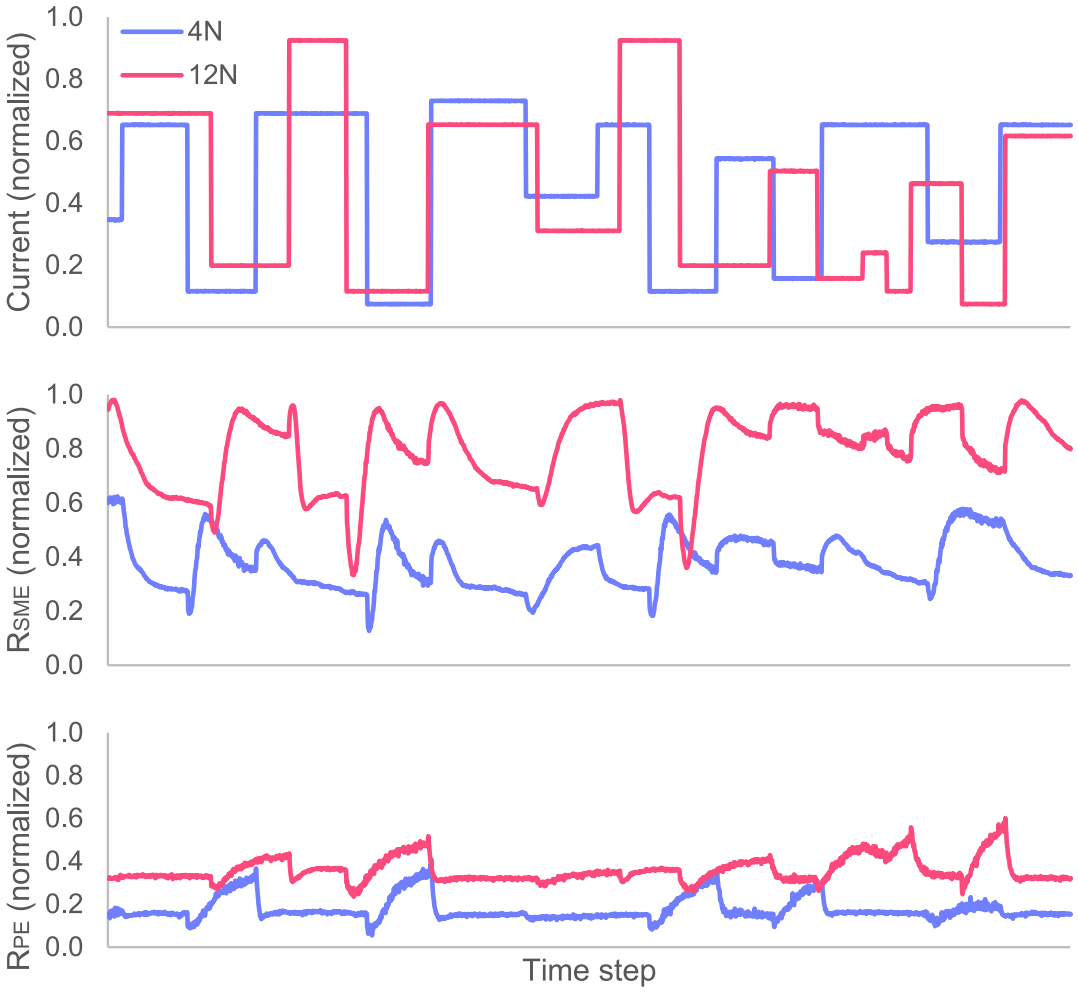

Figure D.1: Starting at the top: Current applied to variable load NiTi actuators modeled using RNN in Section 5.2, along with measured SME and PE resistances. Training data set was used for all model estimations.

D.2 Position and Force Estimation on Testing Data

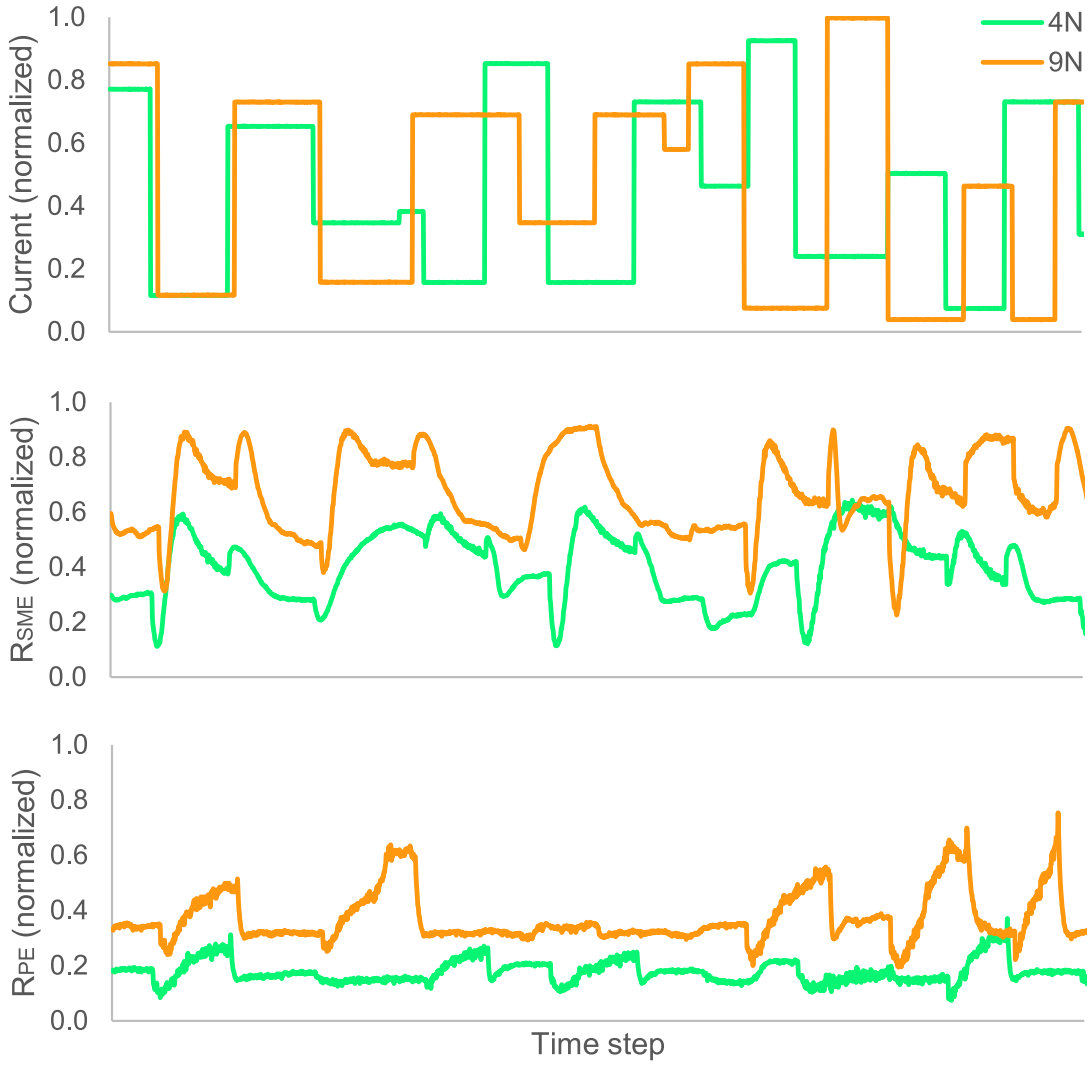

Figure D.2: Starting at the top: Current applied to variable load NiTi actuators modeled using RNN in Section 5.2, along with measured SME and PE resistances. Testing data set was used for all model estimations.