university of groningen / faculty of science and engineering

A Comparative Study of Predictive Models for Nafion-based IPMC Soft Actuators

Bachelor's Project Thesis

Kenny Burawudi,
Supervisors: Prof Dr R. Carloni and R. D'Anniballe, MSc

Abstract: Ionic polymer-metal composites are electro-active polymers that, when stimulated by an electric field, convert electrical energy into mechanical energy. The focus of this research is an ionic polymer-metal composite soft actuator that has been realised by Nafion-117 metallised on both sides with platinum. Three models are developed for this study and their predictive ability is compared. The methods used to realise these models are the Multi-layer Perceptron, the curve fitting and a Long-Short Term Memory neural network. The models aim to predict the force with respect to time at different voltages (low to high applied electric fields) and displacements (how much the actuator bends). Their ability to generalise to unseen samples is also evaluated. The Multi-Layer Perceptron produces the best overall results with a root mean squared error of 0.241 mN on data from the unseen sample and computational time for prediction of 1.3 μs.

1 Introduction

Electro-active polymers are a class of soft actuator that, when stimulated by an electric field, convert electrical energy into mechanical energy. More specifically, when an electric field is applied to the polymer, it exhibits a change in shape by contraction, expansion or bending (Chen et al., 2012; Ye et al., 2017).

Ionic polymer metal composites (IPMCs) are a subclass of electro-active polymers. The conversion from electrical energy to mechanical energy is based on mass transfer, i.e. the migration of cations carrying solvent molecules in the polymer which is caused by an applied electric field and facilitated by the conductive property of the IPMC. This means that the actuator has to be (sufficiently) hydrated in order to change its shape and thus produce a force. The actuator produces a force that is very small in magnitude but could be compounded with for greater effect and muscle-like performance (Mirfakhrai et al., 2007). This has many possible biomedical and industrial applications described by Shahinpoor and Kim, 2005. Some of these industrial applications include mechanical grippers, robotic swimming structures and diaphragm pumps. And some biomedical applications include artificial cardiac-assist muscles with more artificial muscles realised in the form of sphincter and ocular muscles. Some characteristics of IPMCs that enable such applications include low actuation voltages, slow actuation, inherent vibration damping and the ability to work in wet conditions.

IPMCs exhibit highly non-linear behaviours (Truong and Ahn, 2011). There have been many successful attempts to model the tip displacement of the an actuator with respect to applied voltage. Some of these models are realised in the form of a white-box model. White-box models are models realised by establishing a meaningful mathematical relationship among the system's variables based on some physical principles (Liu et al., 2017; Enikov et al., 2006; Punning et al., 2009). Other models have been realised by black-box methods for which the exact system dynamics do not have to be known. These dynamics can be learnt by these black-box models given experimental data. This is a particularly useful characteristic to model IPMCs as their underlying dynamics are currently not well known. What is known for certain is that there are mechanical, electrical and chemical properties that dictate an IPMC's system dynamics. Examples of black-box techniques are the general multilayer per-

ceptron neural network (GMLPNN) developed by Truong and Ahn, 2014 and an adaptive neuro fuzzy inference system (ANFIS) and a nonlinear autoregressive with exogenous input (NARX) developed by Annabestani and Naghavi, 2014. There are currently no models that can accurately predict the force produced by an IPMC actuator.

The IPMC actuator that is the focus of this study is Nafion-117 (sodium-ion) with a perfluorinated membrane and platinum electrodes. This study aims to build predictive models capable of predicting the actuation forces of a Nafion-117 IPMC soft actuator. These models are meant to generalise to new samples. Having models that can predict the actuation forces of an IPMC actuator enables the development of controllers and therefore realise the possible applications of IPMCs (Bhat, 2004).

Three black-box models are proposed in this study. The first model contributed by this study is realised as a feed-forward Multi-Layer Perceptron (MLP), the second is a curve fitted model. The third and final model, a novel approach, proposed by this study is a Long-Short Term Memory (LSTM) neural network. In addition, a framework to compare the models' performance against each other is proposed.

This paper addresses the following research question: *"Can predictive models generalise the actuation forces of a Nafion-117 IPMC soft actuator?"*. The findings in this research will contribute to attempts to model actuation forces of IPMC soft actuators.

The remainder of this paper is organised as follows. Section 2 outlines a fabrication process, describes the experiment from which data from a Nafion-117 actuator is collected and details how data is split for training, testing and validating models. Section 3 details the preprocessing steps for collected data, theoretical background for the models used in this study and how the models will be evaluated. Section 4 displays model's performance and comparative data. In section 5 the results are discussed and linked to related studies. Limitations of this study and suggestions for further research are included in this section. Finally, section 6 summarises the results and outlines the conclusions that can be drawn from this study.

2 Material

This section describes the sample fabrication, control variables, experiment setup and the data collection for this study. This is a continuation of work done by Langius, 2019.

The first step in this study is the fabrication of a composite sandwich construction of the Nafion polymer metallised on both sides with platinum to form electrodes. This sandwich construction is then cut into strips of predetermined dimensions. Throughout this paper, such a strip will be referred to as a sample.

The next step is to perform some experiments on the samples. Experimental data is collected from these samples.

This is then followed by preprocessing in order to correct some systematic errors in data collection and prepare for the modeling of this data. Details in section 3.1.

2.1 Sample fabrication

IPMC actuators are realised by the application of a set of chemical processes. The composite sandwich construction used to create samples in this study is fabricated using standards and methodology as shown by De Luca et al., 2013. There are four steps involved, all of which are outlined below:

1. *Surface Treatment:* the Nafion is roughened with sandpaper. This roughening increases the surface area of the polymer in contact with the platinum electrodes. The Nafion is then washed by immersing it in a boiling acid solution. The washing is meant to ensure the Nafion polymer is completely saturated with protons. The Nafion sheet can be seen in figure 2.1.

2. *Ion Exchange:* the Nafion polymer is then submerged in a salt solution containing platinum ions. The platinum cations are absorbed into the polymer.

3. *Primary Plating:* the absorbed platinum cations are then reduced to the metallic state of platinum. These are called nanoparticles and they form the primary electrode. This process can be seen in figure 2.2.

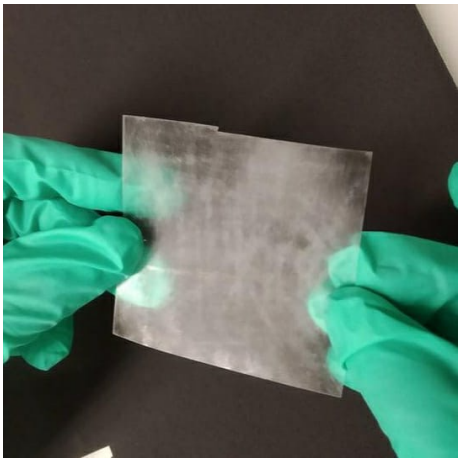

Figure 2.1: Step 1: a roughened, washed Nafion sheet

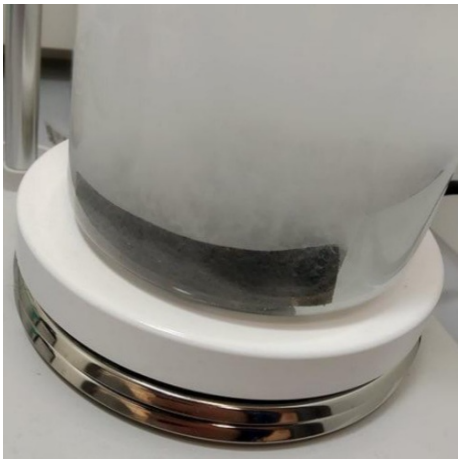

Figure 2.2: Metal ion reduction to form the primary plating

4. *Secondary Plating:* the primary plating is then further developed by additional coating of platinum to form the secondary plating. This step reduces the electrode's resistance, increasing the mass transfer capability.

The samples are then created when the sandwich construction is cut into strips that form samples with dimensions of 65 mm in length and 5 mm in width and a surface area of 455 mm^2 (figure 2.3). Two identically fabricated samples were selected for this study.

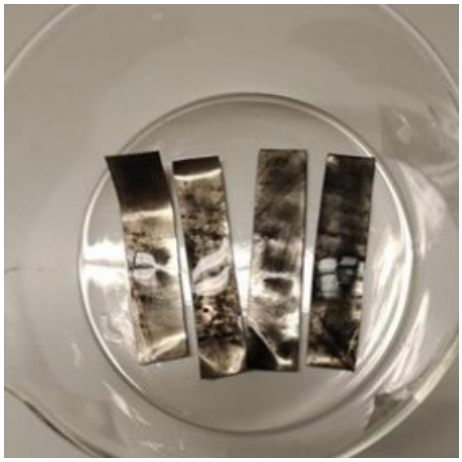

Figure 2.3: Fully formed samples with platinum electrodes

2.2 Control variables

An experiment was designed to observe the behaviour of the samples. Given a step voltage and an initial displacement, the force produced by the sample is measured with respect to time. The variables controlled for this experiment are:

- DC step voltage supplied to the sample, V
- the displacement of the sample above a load cell, d

IPMCs work in low voltages, therefore the voltages (required for actuation) supplied to a sample were limited to 2V, 3V and 4V, in line with previous research (Shahinpoor and Kim, 2001).

IPMCs can bend when an electric field is applied across them. The displacement demonstrates its bending capability and thus, actuation. The displacement was limited to the range of 0mm to 20mm, varied in 2mm increments because the experiment setup can reliably measure data within this displacement range.

2.3 Experiment setup

A number of apparatuses are required for the experiment setup. The experiment is timed and therefore necessitates a timing mechanism. A clamp must hold the sample in place for the duration of the experiment. The clamp is attached to a linear actuator to control the displacement from a load cell. The

load cell is used to measure the force produced by a sample. A voltage amplifier controls the voltage across a sample. A similar setup is demonstrated by Carloni et al., 2018 in which the force-displacement characterisation is replaced by a force-time characterisation for this study.

An Instron ElectroPuls E1000 test instrument encapsulates nearly all the necessary requirements for an experiment of this nature. An additional 3D printed custom mount was installed on the linear actuator of the test instrument. The clamp was attached to the custom mount in order to hold the sample horizontally over the load cell.

The load cell is an Instron static load cell 2530-5N. This is a precision force transducer suitable for materials testing. The load cell has a capacity of 5N and a sensitivity of 1.6mV/V to 2.4mV/V at static rating. This is a suitable load cell for this experiment because of the small (and slow) actuation forces produced by a sample.

This study uses a TREK MODEL 10/10B-HS high-voltage amplifier to apply an electric field across a sample. It has many typical applications including testing of electro-active polymers. Copper tape is used as electrodes from the voltage amplifier to the sample.

The Instron control system enables the automation of data collection and control of variables when the experiment is in progress. This automation allows for a standard experiment setup and data collection procedure. With these features, the displacement of the sample from the load cell is monitored and logged with a specified frequency during the experiment. Additionally, measurements of the force produced by the sample on the load cell are also logged. In this study, all data is logged at a frequency of 10Hz.

The resulting setup can be seen in figure 2.4. Most of the tasks in the experiment setup are automated. With the exceptions of the hydration and attachment of the sample to the clamp and voltage amplifier electrodes.

The steps for setting up the experiment are as follows:

1. Hydrate the sample for one minute. Then dry the surface of the sample with paper

2. Attach the sample to the voltage amplifier and clamp

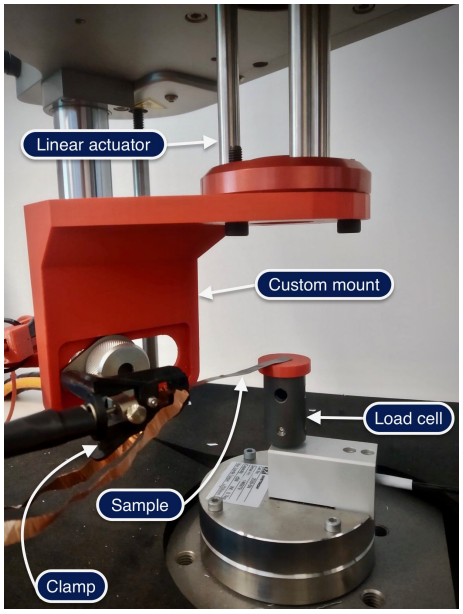

Figure 2.4: The full experiment setup

3. Calibrate the load cell

4. Set the displacement of the sample from the load cell

The experiment can then be run by starting the voltage amplifier and the Instron test instrument at the same time.

2.4 Data collection

Training data is required to build black-box predictive models. A validation dataset can be used for the internal validation of the model i.e to show how well the model performs (if there is any learning done) over the course of training. A test dataset, one unknown to the model during the training process, is also required to test if the model generalises to unseen data. This dataset is important because the relationships between variables learnt by black-box models may be specific only to the training data rather than the phenomena the model is meant to recognise.

Paying attention to this, data collected from one of the samples is used to train and validate models. This is referred to as the training sample. Two rounds of data collection were completed on the training sample to form the training dataset. An

additional round was completed on the same sample validation dataset. Data from the other sample is used to test the models. This is referred to the test sample from which the test dataset is obtained after a single round of data collection.

Since there are three voltage conditions and eleven displacement conditions, each sample therefore undergoes 33 runs under the conditions of every voltage-displacement pair for a complete round of data collection. Runs are limited to 100s in duration, which is considered to be sufficient time to observe the sample response of an initial rise until a peak actuation force before the sample becomes dehydrated due to electrolysis and loses its ability to bend and thus can no longer produce a force. The stage when the sample is dehydrated is referred to as the dehydration phase in this paper and is recognisable as a decay in the actuation force over time.

3 Methods

In this section, the methods used to prepare collected data and to build the models implemented for this study are described. This section also includes a framework to evaluate and interpret the results produced by the models both internally (i.e the predictive ability of the model itself) and compared to other models' predictive ability.

3.1 Data preprocessing

Data is preprocessed before it can be presented to a machine learning method for the purposes of building a model. The preprocessing pipeline is defined as follows:

1. Remove the calibration offset in data from the load cell
2. Extract data from the first 80 seconds of the run
3. Replace missing force measurements with 0N
4. Apply a rolling median over force data collected

Figure 3.1 is an example of preprocessed data overlaid on the collected data.

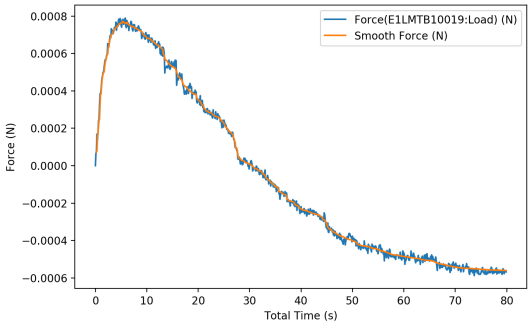

Figure 3.1: Preprocessed data from the training sample overlaid on collected data

Step 1 makes forces positive and ensures that each run begins from 0N regardless of the measurement errors in the load cell. This is done by multiplying force readings by -1 to make readings positive. The minimum force found within the first 5 seconds of the run can then be used as the offset that is subtracted from all force measurements in the run.

Step 2 in the pipeline is motivated by two factors. The first being that the sample response is static by this stage (80s) in the run. The remaining 20 seconds contain no new information for models to learn. The second factor is an error in force measurement inherent to the process. This results in negative or "pulling" force readings. The load cell is calibrated to 0N with the weight of the hydrated sample. The negative force readings are a result of the sample losing mass over the course of the run due to dehydration. This error cannot be removed systematically so step 2 aims to minimise the effect of these inaccurate measurements.

Step 3 deals with missing values in the force readings. Sometimes the test instrument does not log the first 0.1 or 0.2 seconds of the run. Since the experiment begins with a force of 0N, IPMC actuation is slow and the range of forces produced is minuscule (between 0N and 0.0015N), it is reasonable to assume that force readings in those timesteps are or very close to 0N.

Step 4 smooths force readings. Some of the readings can be especially noisy, as shown in figure 3.1, particularly in the dehydration phase. A way to deal with this is to capture the trend of the data between timseteps. In this study, a rolling median

sampling window is run over the data. The sampling window increases in size with respect to time (equation 3.1) because over the course of the experiment, readings become increasingly noisy. The increasing sampling window size attempts to maintain the trend in noisy data. The same scheme is used by Langius, 2019.

$$size = \left\lfloor \frac{10t}{4} \right\rfloor + 5 \tag{3.1}$$

Where t is time in seconds (logged at 10Hz). $\lfloor f(t) \rfloor$ is a floor function on f(t).

Data is further preprocessed for neural networks by scaling data. Scaling data maintains the shape of the distribution of data but limits the data to a given range (typically from 0 to 1) which speeds up computation.

With a training window of 80 seconds and data collected at 10Hz, there are 800 data points for any single run. This results in $800 \times 33 = 26400$ data points for a round of data collection on a sample.

3.2 Multi-layer perceptron

The feed-forward Multi-Layer Perceptron (MLP) is simple and powerful with regards to function approximation (Hornik, 1991; Cybenko, 1992). This makes it a natural choice to model the nonlinear sample response for a Nafion-117 actuator.

In this study, the architecture as in figure 3.2 was implemented, an extension of the work done by Langius, 2019. Where:

- d is displacement of the sample from the load cell
- V is DC voltage across the sample
- t is the current time
- F(t) is the scaled predicted smooth force
- All hidden neurons and the output neuron have a sigmoid activation
- Hidden neurons all have a 20% dropout chance

The input layer represents a vector of predictors at a single time slice. MLP optimisation is realised by finding the set of weights that minimise a loss function. The loss function used in the study is the mean squared Error (MSE) which is typical

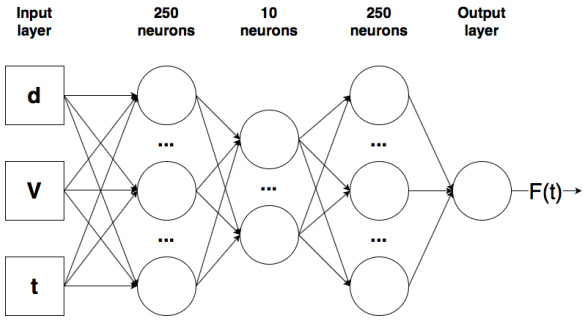

Figure 3.2: MLP architecture used for this study

for least-squares optimisation problems. The optimiser chosen for this study is the Adam optimiser which makes use of adaptive momentum to find a local minimum in the gradient (loss) function. It also has known benefits in terms of computational time, memory usage and faster convergence (Zhang, 2018).

All data, i.e. input and output, are scaled to the range of [0,1] which is typical practice in neural networks. The min-max scaler is used in this study which scales data to the range [0,1] without changing the shape of the distribution of any of the variables. This helps with conversion, making all data fall in the same range. The scaled data thus motivates the use of the sigmoid activation function for neurons. The resulting prediction for a given timestep under different conditions is a sum of sigmoids that is bound within the range [0,1] by the output layer. The root mean square error (RMSE) between the target and prediction can then be considered as a measure of average percentage error in the MLP predictions.

The MLP is trained for 20 epochs during which hidden neurons each have a dropout chance of 20%. This means there is a 20% chance that, each training iteration, a hidden neuron does not contribute to the prediction of the training example and thus, subsequently to the error propagation through the MLP to update neuron weights. This is a simple method to minimise overfitting as described by Srivastava et al., 2014.

3.3 Curve fitting

Curve fitting for this study assumes that the actuation forces for all samples follow a particular behaviour. This assumption relates force to time,

generally as in equation 3.2.

$$F(t) = g(\vec{x}, t) + \epsilon \tag{3.2}$$

Where F(t) is the force observed at time t, ϵ is an error term (assumed to be drawn from a normal distribution) and model parameters are represented as vector $\vec{x}$.

The function g is a predefined function whose parameters, $\vec{x}$ need to be fit. Nafion-117 actuation follows a typical process in which actuator produces a force (the sample response) until a peak force is attained. Dehydration then sets in and the actuator loses the ability to exert a force. This process may be modeled by function g, which is assumed to be a lognormal probability density function (PDF) in t for a given voltage-displacement pair. For simplicity, optimal parameters are found and saved for individual voltage-displacement pairs on the training data. That is, voltage and displacement are treated as ordinal variables. A multivariate function that treats voltage and displacement as numerical variables is beyond the scope of this study.

The function g in equation 3.2 is thus defined as in equation 3.3.

$$g(\vec{x}, t) = lognorm(t, a) + b \tag{3.3}$$

Where parameter a is a distribution scaling parameter, b is a translation parameter and lognorm(t,a) is the standardised PDF, defined as in equation 3.4.

$$lognorm(t, a) = \frac{a^2}{st\sqrt{2\pi}} exp\left(-\frac{ln^2\left(\frac{t}{a}\right)}{2s^2}\right) \tag{3.4}$$

Where s is the shape parameter for the lognormal PDF. This model therefore fits three parameters:

1. distribution scaling parameter, a
2. translating coefficient, b
3. lognormal PDF shape parameter, s

Parameter a scales the range of the lognormal PDF to the range of the force produced by a sample. The translating coefficient, b attempts to account for some of the experimental errors that could not be removed in the preprocessing stage. The shape parameter s determines the shape of the lognormal PDF.

The Levenberg–Marquardt Algorithm (LMA), first described by Levenberg, 1944 and refined by Marquardt, 1963 is the method used to fit the parameters by minimising the error between prediction and target using damped least-squares optimisation. The LMA's optimisation process is iterative during which parameters $\vec{x}$ are adjusted at each step until the LMA finds a local minimum in the gradient function. Iterations are limited to a maximum of 1000.

The LMA interpolates between the gradient descent algorithm (GDA) and the Gauss-Newton algorithm (GNA). It is faster in convergence than the GDA and more robust than the GNA (Zhao et al., 2014) because the LMA is more tolerant to bad initialisation of model parameters. In this study, model parameters are initialised to 1.

3.4 Long-short term memory

Recurrent neural networks (RNN) are an extension of the feed-forward MLP. The extension is that RNNs may have connections that can span adjacent time-steps (Lipton et al., 2015). This means that previously seen information can be incorporated into new information. They therefore exhibit the ability to model dependencies in data and hence are well suited to represent differential equations (Kruse et al., 2013; Sherstinsky, 2020).

Differential equations are often used to describe physical processes. The actuation of a sample over a time may be represented by some system of differential equations. This may prove to be a useful property to model the dynamic system that causes IPMC actuation. Dynamic systems have been proven to be approximated by recurrent neural networks (Funahashi and Nakamura, 1993; Samarasinghe, 2007).

Long-Short Term Memory (LSTM) networks are a class of recurrent neural network. The LSTM is favoured in this study over the traditional recurrent neural network because of a typical problem faced by traditional RNNs. This is the vanishing (or exploding) gradient problem when modelling long-term dependencies first formulated by Bengio et al., 1994. This problem arises when the error gets infinitesimally small, approaching 0 (or infinitely large) and can therefore not be used to update the weights of the network. LSTM neural networks introduced by Hochreiter and Schmidhuber, 1997

manage this problem by introducing a memory cell to each neuron which allows LSTMs to learn long-term dependencies.

Each LSTM cell has its own state. This state is updated at each timestep in the process. The cell state enables a constant gradient flow which prevents vanishing (or exploding) gradients. These properties could be useful particularly in real-world applications such as closed-loop control where the time in which the actuator is operated (possibly in wet conditions) is unbounded.

LSTMs have a vast array of applications, particularly in relation to sequence modeling. They have been used to model many sequence problems before and are currently one of the most sought after solutions for such problems (Li et al., 2020; Lipton et al., 2015).

The sample response may have some long and short-term dependencies (which could be described by an unknown set of differential equations) that the LSTM may be able to learn. A sequence of input vectors is presented to the LSTM for which the resulting forces are predicted. The LSTM maintains an internal state across time steps and can incorporate previous information in the upcoming predictions. Figure 3.3 is a representation of how dependencies within a sequence may be modeled. Red boxes represent the input vector at time t. Blue boxes represent the prediction for time t. Purple boxes are the LSTM layer(s) with a certain state at time t that maintain their state (or part of it) in the next timestep(s) as represented by the horizontal arrows (recurrent connections) connecting them.

An LSTM layer is completely described by a system of equations. Before demonstrating these equations, a few variables must be defined. For gate k, matrix $\vec{W}_k$ is the input weight matrix, matrix $\vec{R}_k$ is the recurrent connection weight matrix and vector $\vec{b}_k$ is the bias vector. The system of equations can then be defined as follows:

$$f_t = \sigma(\vec{W}_f \vec{x}_t + \vec{R}_f \vec{h}_{t-1} + \vec{b}_f) \tag{3.5}$$

$$i_t = \sigma(\vec{W}_i \vec{x}_t + \vec{R}_i \vec{h}_{t-1} + \vec{b}_i) \tag{3.6}$$

$$o_t = \sigma(\vec{W}_o \vec{x}_t + \vec{R}_o \vec{h}_{t-1} + \vec{b}_o) \tag{3.7}$$

$$\bar{c}_t = \tanh(\vec{W}_{\bar{c}} \vec{x}_t + \vec{R}_{\bar{c}} \vec{h}_{t-1} + \vec{b}_{\bar{c}}) \tag{3.8}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \bar{c}_t \tag{3.9}$$

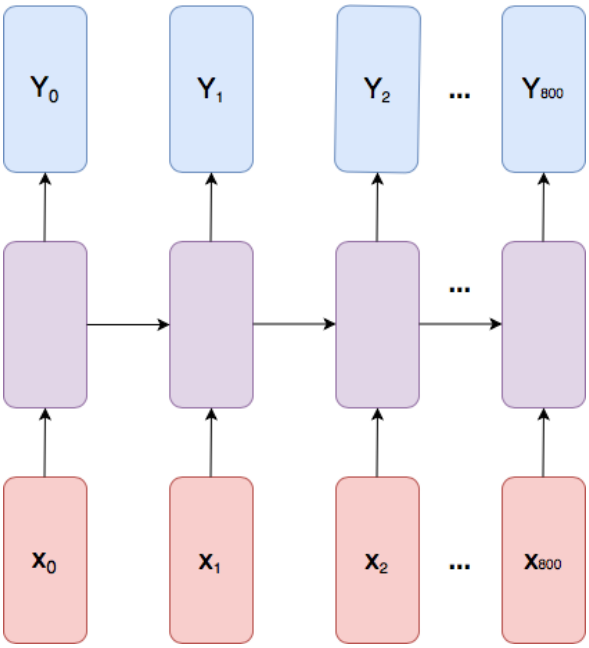

Figure 3.3: Process of actuation as learnt by LSTMs

$$h_t = o_t \circ \tanh(c_t) \tag{3.10}$$

Where $\circ$ is the element-wise product, σ is the sigmoid activation function, tanh is the hyperbolic tangent activation function and $\vec{x}_t$ is the input vector at timestep t.

Equation 3.5 represents the forget gate's activation, 3.6 the input gate's activation and 3.7 the output gate's activation. Equation 3.8 is the LSTM layer's cell input activation vector and 3.9 represents the layer's cell state vector. The hidden state vector of the LSTM cells in a layer is represented by 3.10.

As with the MLP, in this study, data is scaled using the min-max scaler. The architecture used in this study was a neural network consisting of 3 hidden LSTM layers similar to the architechture of the MLP in figure 3.2 except with simple neurons replaced with LSTM cells. The first layer consists of 100 LSTM cells. The second, 10 LSTM cells and the final layer; 100 LSTM cells. Multiple hidden layers can model more complex dependencies (Kruse et al., 2013). All hidden cells within the neural network had a 20% dropout chance to avoid overfitting. The loss (error) function minimised by the network is the mean squared error (MSE) The op-

timiser used to minimise this function was, as with the MLP, the Adam optimiser. The neural network was then trained for 7 epochs. Early stopping determined that larger number of epochs results in no improvement or poorer performance.

3.5 Model evaluation

To compare the performance of each model, a performance metric must be applicable to the problem. The metric must give some indication of how well a model fits given data.

The problem description is a non-linear regression problem between force and time under the control conditions. A metric that can be used for the evaluation of non-linear regression problems is the root-mean squared error (RMSE), as in equation 3.11. This metric is suited to the problem description because it represents the average error in prediction. A lower RMSE implies better model performance.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2} \tag{3.11}$$

Where:

- y_i is the scalar target value
- $\hat{y_i}$ is the scalar predicted value
- n is the number of predictions

A model's ability to generalise to unseen data is evaluated by comparing the RMSE on the test data with the final RMSE on the training and validation data. If they are similar, the models can generalise. Model comparisons are drawn by comparing their test RMSE to other models.

In addition, with the described experiment process, there is a general region in which a sample produces a force, when the sample response is dynamic. This region is in the first 20 seconds of the sample response. This region will be referred to as the region of actuation. It is therefore an important region to consider and models must sufficiently capture the sample response in this region. Therefore an additional condition to the model evaluation is a second RMSE value on the test set for predictions made in the first 20 seconds of each run. A lower RMSE in this region than for the whole run (80s after preprocessing) indicates that the model learns to predict the dynamic part of the sample response. A higher RMSE than for the whole run indicates that the model has not learnt the dynamic part of the sample response and instead minimises the error when the sample is dehydrated which is a mostly static region in the experiment process.

If any two models perform similarly in terms of final test RMSE, a pairwise comparison is made. This comparison is in the form of a scatter plot of the models' RMSE in the region of actuation. These RMSE values will be grouped by voltage condition.

However, since the models are machine learning algorithms, the RMSE can be misleading because a machine learning algorithm can learn a noisy relation (that does not follow the pattern in the data) that results in a low RMSE. A sanity check is therefore necessary. Such a check is overlaying the model's predictions over the actual data which exhibits an initial increase, a peak and decay as a result of sample dehydration. If the predictions exhibit these features, the RMSE can be used to support observations.

The prediction time for each model will also be a metric to evaluate models. Lower prediction times indicate a lower model complexity.

4 Results

In this section, the training results for the Multi-Layer Perceptron (MLP), Long-Short Term Memory (LSTM) neural network and the curve fitted model (CFM) are presented individually. Under the Comparison subsection, the model evaluation metrics are used to compare models in relation to each other.

The scaled root mean squared error (RMSE) losses with respect to training epochs for the neural network models are included in their individual subsections. The RMSE for the training and validation included in these plots is useful to evaluate how much the model has learnt. For the curve fitted model, training and validation losses with respect to iterations in the Levenberg-Marquardt algorithm (LMA) are not included, only the final losses are included.

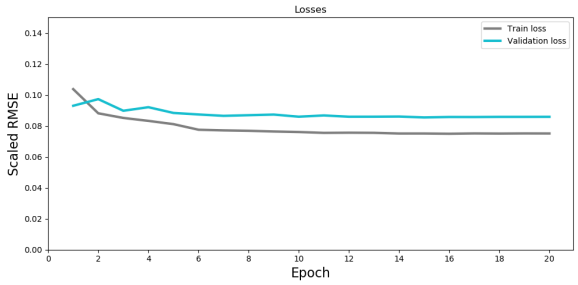

Figure 4.1: MLP root mean squared error loss with respect to epoch

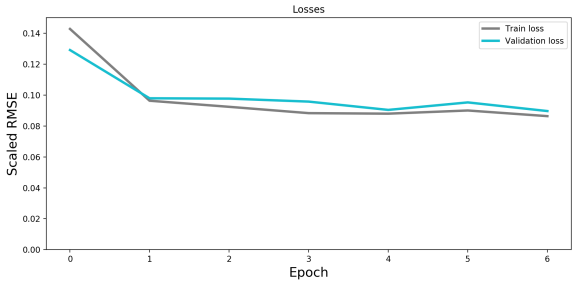

Figure 4.2: LSTM root mean squared error loss with respect to epoch

4.1 Multi-layer perceptron

The decreasing RMSE losses with respect to training epochs in figure 4.1 indicate that the MLP learnt a relationship between the input features and the actuation forces. The MLP had a final scaled training RMSE of 0.07513 (0.133 mN) and a final scaled validation RMSE of 0.08589 (0.152 mN).

The model returned a scaled test RMSE of 0.136 (0.241 mN). This higher RMSE for the test data compared to the training and validation data indicates that the MLP model does not generalise to the test (unseen) sample.

4.2 Long-short Term memory

The results of the LSTM's training losses are in figure 4.2. The model learnt a relationship between the input features and actuation forces. This is indicated by the decrease in (scaled) RMSE losses. There is a close association between the scaled RMSE of 0.0777 (0.137 mN) on the training data and the scaled RMSE of 0.0896 (0.158 mN) on the validation data. This close association also indicates that the LSTM model was indeed able to learn time dependencies in the training sample.

The model returned a scaled test RMSE of 0.142 (0.251 mN). This higher RMSE for the test data compared to the training and validation data indicates that the LSTM model does not generalise to the test (unseen) sample.

4.3 Curve fitted model

The curve fitted model returns a final RMSE of 0.0755 mN and 0.1016 mN on the training and validation data respectively.

The model returned an RMSE of 0.313 mN on the test data. This higher test RMSE on the test data compared to the final training and validation RMSE indicates that the curve fitted model does not generalise to an unseen sample.

4.4 Comparison

The evaluation metrics for model comparison are treated in this subsection. These metrics include the sanity check (overlaying each model's prediction over the test data), a tabulation of test RMSEs for all models and computational time for prediction as described in section 3.5.

4.4.1 Sanity check

Figures that overlay an individual model's predictions on the test data are included in Appendix A for one-to-one comparison. All pass the sanity check (Figures A.1, A.2 and A.3 in Appendix A) described in section 3.5. All models learn the pattern of the sample response.

Figure 4.3 shows that the MLP and LSTM models exhibit similar performance with respect to prediction with similar prediction patterns.

4.4.2 Root mean square error

Table 4.1 shows the RMSE in milli-Newtons (mN) and indicates that the neural network models models predicted the dynamic part of the sample response better than for the entire sample response. It also indicates that for the test sample, the LSTM returns the least expected error in the region of actuation. The MLP returns the least expected error

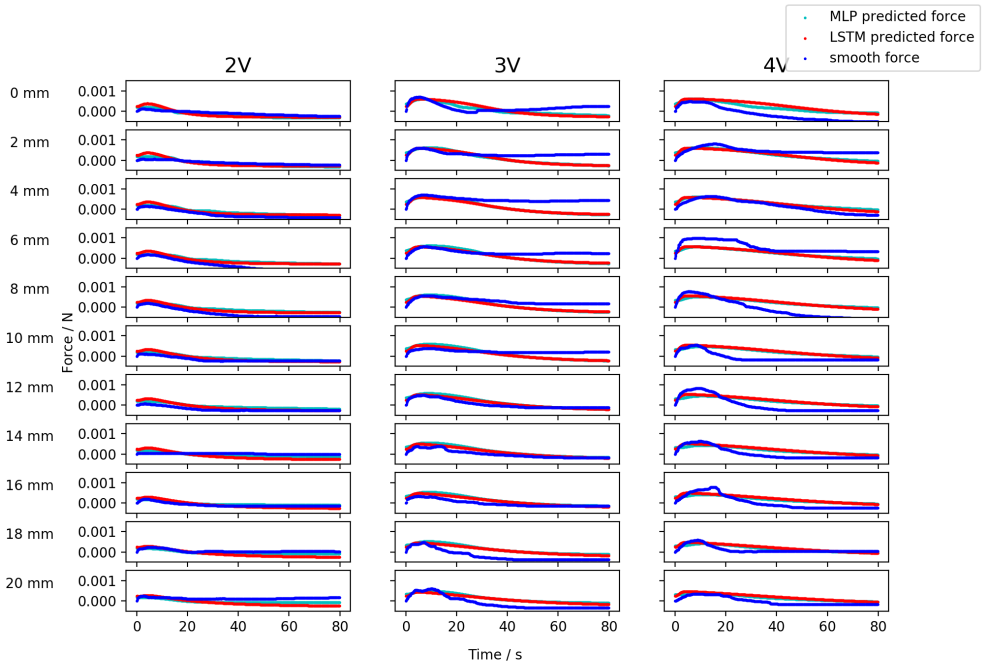

Figure 4.3: MLP and LSTM predictions of test data overlaid on target data.

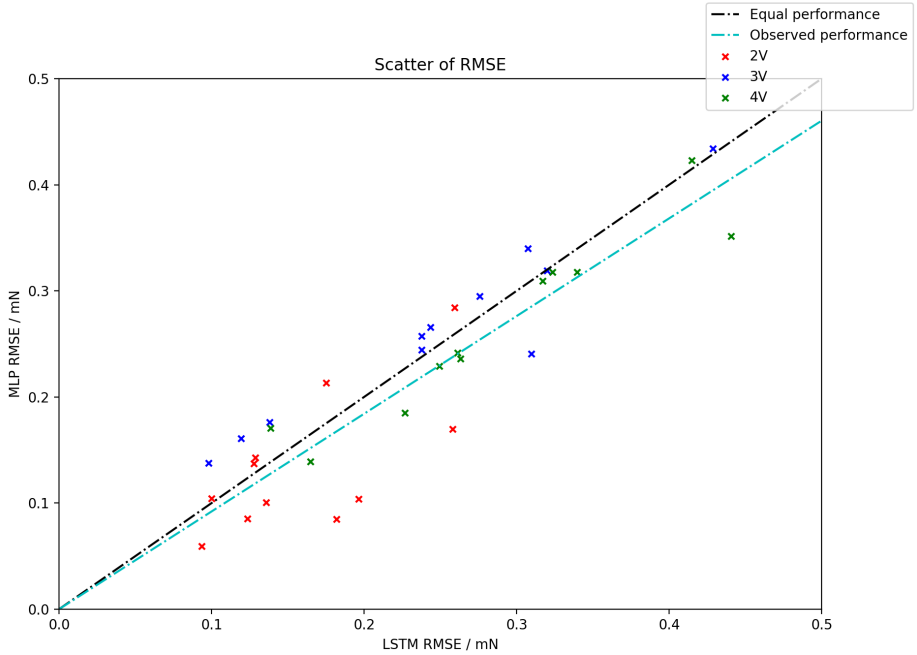

Figure 4.4: Multi-Layer Perceptron against Long-Short Term Memory RMSE. RMSE taken from region of actuation.

	MLP	CFM	LSTM
<20s	0.175	0.326	0.170
0-80s	0.241	0.313	0.251

Table 4.1: Root mean square error on test data. Units are in mN.

MLP	CFM	LSTM
13	1.3	225

Table 4.2: Average computational time for prediction of a timestep. Units are in microseconds (μs).

over the entire process. The curve fitted model performs the poorest out of all three models. The curve fitted model fails to generalise the sample response as seen there is a higher RMSE in the region of actuation than for the entire sample.

The MLP and the LSTM models perform similarly on the test sample with reasonable RMSE values. However, figure 4.4 indicates (by the observed relation between model RMSEs) that the MLP model was more reliable when presented with the unseen sample.

4.4.3 Computational time

Table 4.2 shows the computation times for the prediction of a single input vector with respect to each model. This shows that the fastest predictive model is the curve fitted model with an average prediction time of 1.3 μs.

All models have prediction times at least 440 smaller than 0.1s. They can therefore all realistically be used to predict actuation forces in real time with the logging frequency of 10Hz as used in this study.

5 Discussion

This section includes an interpretation of the results, provides explanations for these results and suggests possible improvements. How (or if) the models could be used for possible real-time predictions in future applications will be made from the prediction time. The limitations of this study and possible avenues for future research are also discussed in this section which subsequently leads into the conclusion of this study.

5.1 Model evaluation

All three models pass the sanity check (described in section 3.5), meaning that the RMSE on all data can therefore be used to evaluate model performance.

The curve fitted model performed the poorest of the three models with a considerably higher RMSE on the test sample when predicting sample response in the region of actuation and over the entire process (see table 4.1). As a result of the translating factor, the predictions start at (in some cases) values less than 0, observable as a "tail" in figure A.2, Appendix A. This contributes to the higher RMSE.

The curve fitted model has the disadvantage of making an assumption on the behaviour of a sample. This is shown by a very low training and validation RMSE but a considerably larger RMSE on the test sample. This shows that the assumption made by the model does not hold. An explanation for this lies in the non-linearities IPMCs exhibit in their actuation as described by Truong and Ahn, 2011.

Of all the models built for this study, the curve fitted model was the simplest with only 99 parameters which made the model worth investigating. This model simplicity also made it the most computationally efficient model with a prediction time of 1.3 μs which would be best suited for real-time predictions if a good model is developed.

The Long-Short Term Memory (LSTM) model was indeed able to learn dependencies in the training sample as demonstrated by similar RMSEs for data from the training and validation datasets which were both drawn from the same sample. The Multi-Layer Perceptron (MLP) makes an assumption of independence between input vectors. Using this knowledge, the notion that the LSTM model learnt dependencies in the training data is further supported by the noticeable split in the training and validation RMSE with respect to training epoch for the MLP in figure 4.1.

It is difficult however, to reconcile the added model complexity for the LSTM model with a negligible improvement in overall prediction performance and a prediction time 17 times higher than that of the MLP. It is worth noticing however, that although the same training data was used for all

methods, the LMA and MLP received individual data points as input coming to a total of 800 × 66 = 52800 "independent" training examples. The LSTM model received training examples as entire runs meaning the model had only 66 independent training examples to learn relationships in the data.

The MLP model is a simplified adaptation of the general multi-layer perceptron neural network (GMLPNN) developed by Truong and Ahn, 2014. It is the most consistent model with generally good performance across all metrics of evaluation as outlined in section 3.5. The model's strongest point is its relative simplicity as a universal function approximator (Hornik, 1991; Cybenko, 1992).

The MLP and LSTM models perform very similarly but the MLP returns more reliable predictions when individual voltage conditions are considered (see figure 4.4) is the more consistent model. The gains made by the added model complexity in the LSTM model are undone by poor performance with respect to the MLP in certain conditions, particularly lower voltage conditions. Applying Occam's razor, the MLP is most likely the better model.

5.2 Limitations

A limitation in this study lies in data collection. It is difficult to collect data due to the long fabrication and hydration process. This makes large experimental datasets for black-box methods difficult to create.

The irremovable calibration error due to a sample's loss of mass over the course of a run which results in unpredictable errors between repeated runs is a limitation in the experiment setup proposed by this study. Modeling the weight loss of an acting sample may help to reduce these errors be considered in the preprocessing stage. Another solution may be to reconfigure the experiment setup to a horizontal configuration. That way, the sample pushes laterally against the load cell and any effects of sample's weight on force readings may be minimised.

The sanity check proposed by this study is a limitation in that for larger datasets, it is not feasible to inspect all overlays in order to verify that models capture the process of actuation. Cross correlation between the target and predictions may also make the verification. A decision threshold may be used to define when to accept that models indeed capture the sample response. The cross correlation may also be used together with root mean square errors to evaluate model performance.

5.3 Future research

Neural network models in particular will benefit from developing knowledge about the dynamics of IPMC soft actuators. It may therefore be useful to consider the variables considered in available white-box methods that attempt to model IPMC behaviour analytically. For example, Liu et al., 2017 consider the IPMC strip's thickness and Young's modulus.

Recurrent neural networks and therefore by extension, LSTM neural networks have been successfully applied to model a number of physical processes (Li et al., 2020; Elsworth and Güttel, 2020; Kruse et al., 2013). The LSTM model's success in modeling the training sample's dependencies but failure to generalise to the unseen test sample indicates that there may be some other system variables that need to be considered in order to create a model that can generalise actuation forces. When more is known about the underlying dynamics of IPMC actuation, it would therefore be useful to investigate the application of LSTMs to model the more complex dependencies. Particularly in the case of dynamic voltage and displacement modulation which would produce more complex behaviours.

6 Conclusion

This study proposed three black-box models to model the actuation forces of a Nafion-117 IPMC actuator. These models were a Multi-Layer Perceptron (MLP), a curve fitted model optimised by the Levenberg-Marquardt Algorithm (LMA) and a Long-Short Term Memory (LSTM) neural network which is a novel approach.

The study also proposed a framework to compare the performance of these models in relation to each other, contributing a model selection paradigm for future IPMC actuation force modeling attempts.

The study showed that the assumption made by the curve fitted model does not hold for unseen samples and that the LSTM neural network and the MLP are better models that perform simi-

larly with regard to prediction. However, the MLP is best suited to model the actuation forces of a Nafion-117 IPMC actuator. This observation was made on balance of three factors. The first being a prediction time of 13 μs. The second being root mean square errors on predictions of a test (unseen) sample of 0.175 mN in the dynamic part of the sample response together with an RMSE of 0.241 mN over the entire sample response. The third being the consistency of predictions across all voltage (applied electric field) conditions.

The research aimed to demonstrate if predictive models generalise the actuation forces produced by a Nafion-117 IPMC soft actuator. The study concluded that although models show the ability to learn the structure of the sample response, they cannot generalise the actuation forces produced by a Nafion-117 IPMC soft actuator.

References

Annabestani, M. and Naghavi, N. (2014). Nonlinear identification of ipmc actuators based on anfis–narx paradigm. *Sensors and Actuators A: Physical*, 209:140 – 148.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–66.

Bhat, N. D. (2004). Modeling and precision control of ionic polymer metal composite. Master's thesis, Texas A&M University.

Carloni, R., Lapp, V. I., Cremonese, A., Belcari, J., and Zucchelli, A. (2018). A variable stiffness joint with electrospun p(vdf-trfe-ctfe) variable stiffness springs. *Ieee Robotics and Automation Letters*, 3(2):973–978.

Chen, Z., Um, T., and Bart-Smith, H. (2012). Ionic polymer-metal composite artificial muscles in bio-inspired engineering research: Underwater propulsion. In Berselli, G., Vertechy, R., and Vassura, G., editors, *Smart Actuation and Sensing Systems*, chapter 10. IntechOpen, Rijeka.

Cybenko, G. (1992). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 5(4):455–455.

De Luca, V., Digiamberardino, P., G., D. P., Graziani, S., Pollicino, A., Umana, E., and Xibilia, M. G. (May 2013). Ionic electroactive polymer metal composites: fabricating, modeling, and applications of postsilicon smart devices. *Journal of Polymer Science Part B: Polymer Physics*, 51(9):699–734.

Elsworth, S. and Güttel, S. (2020). Time series forecasting using lstm networks: A symbolic approach.

Enikov, E. T., Seo, G. S., and Lacy, S. (2006). Numerical analysis of muscle-like ionic polymer actuators. *Biotechnology Progress*, 22(1):96–105.

Funahashi, K. and Nakamura, Y. (1993). Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, 6(6):801–806.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–80.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251—257.

Kruse, R., Borgelt, C., Klawonn, F., Moewes, C., Steinbrecher, M., and Held, P. (2013). *Computational Intelligence: A Methodological Introduction*, chapter 9, pages 159–172. Texts in Computer Science. Springer London, 2nd edition.

Langius, R. (2019). A predictive model for nafion-based ipmc soft actuators. Bachelor's Project.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168.

Li, Z., Yan, H., Zhang, C., and Tsung, F. (2020). Long-short term spatiotemporal tensor prediction for passenger flow profile. *IEEE Robotics and Automation Letters*, 5(4):5010–5017.

Lipton, Z. C., Berkowitz, J., and Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning.

Liu, H., Xiong, K., Bian, K., and Zhu, K. (2017). Experimental study and electromechanical model analysis of the nonlinear deformation behavior of ipmc actuators. *Acta Mechanica Sinica*, 33(2):382–393.

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441.

Mirfakhrai, T., Madden, J. D. W., and Baughman, R. H. (2007). Polymer artificial muscles. *Materials Today*, 10(4):30 – 38.

Punning, A., Johanson, U., Anton, M., Aabloo, A., and Kruusmaa, M. (2009). A distributed model of ionomeric polymer metal composite. *Journal of Intelligent Material Systems and Structures*, 20(14):1711–1724.

Samarasinghe, S. (2007). *Neural networks for applied sciences and engineering : from fundamentals to complex pattern recognition*, chapter 9, pages 437–555. Auerbach Publications, Boca Raton, FL.

Shahinpoor, M. and Kim, K. J. (2001). Ionic polymer-metal composites: I. fundamentals. *Smart Materials and Structures*, 10(4):819–833.

Shahinpoor, M. and Kim, K. J. (2005). Ionic polymer-metal composites: Iv. industrial and medical applications. *Smart Materials and Structures*, 14(1):197–214.

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Truong, D. Q. and Ahn, K. K. (2011). Design and verification of a non-linear black-box model for ionic polymer metal composite actuators. *Journal of Intelligent Material Systems and Structures*, 22(3):253–269.

Truong, D. Q. and Ahn, K. K. (2014). Modeling of an ionic polymer metal composite actuator based on an extended kalman filter trained neural network. *Smart Materials and Structures*, 23(7):074008.

Ye, Z., Hou, P., and Chen, Z. (2017). 2d maneuverable robotic fish propelled by multiple ionic polymer-metal composite artificial fins. *International Journal of Intelligent Robotics and Applications*, 1(2):195–208.

Zhang, Z. (2018). Improved adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, pages 1–2.

Zhao, J., Li, Y., Yu, X., and Zhang, X. (2014). Levenberg-marquardt algorithm for mackey-glass chaotic time series prediction. *Discrete Dynamics in Nature and Society*, page 1–6.

A Appendix

This appendix contains predictions overlaid on the actual data for each method.

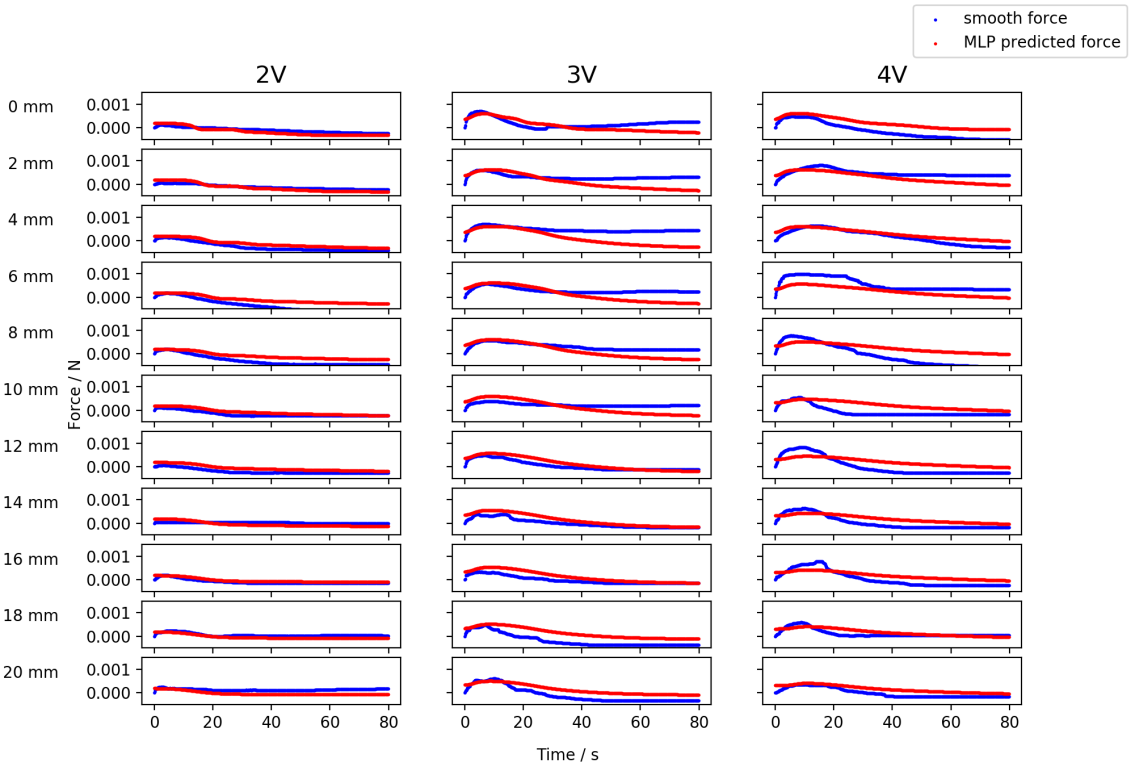

Figure A.1: MLP predictions overlaid on test data

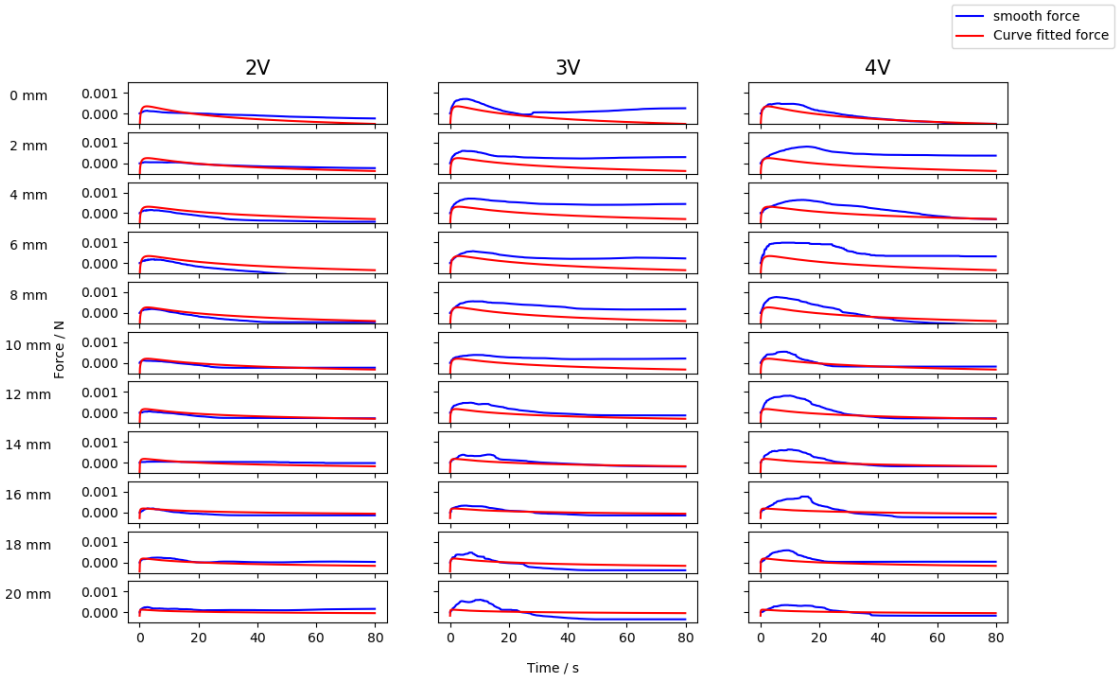

Figure A.2: Curve fitted model predictions overlaid on test data

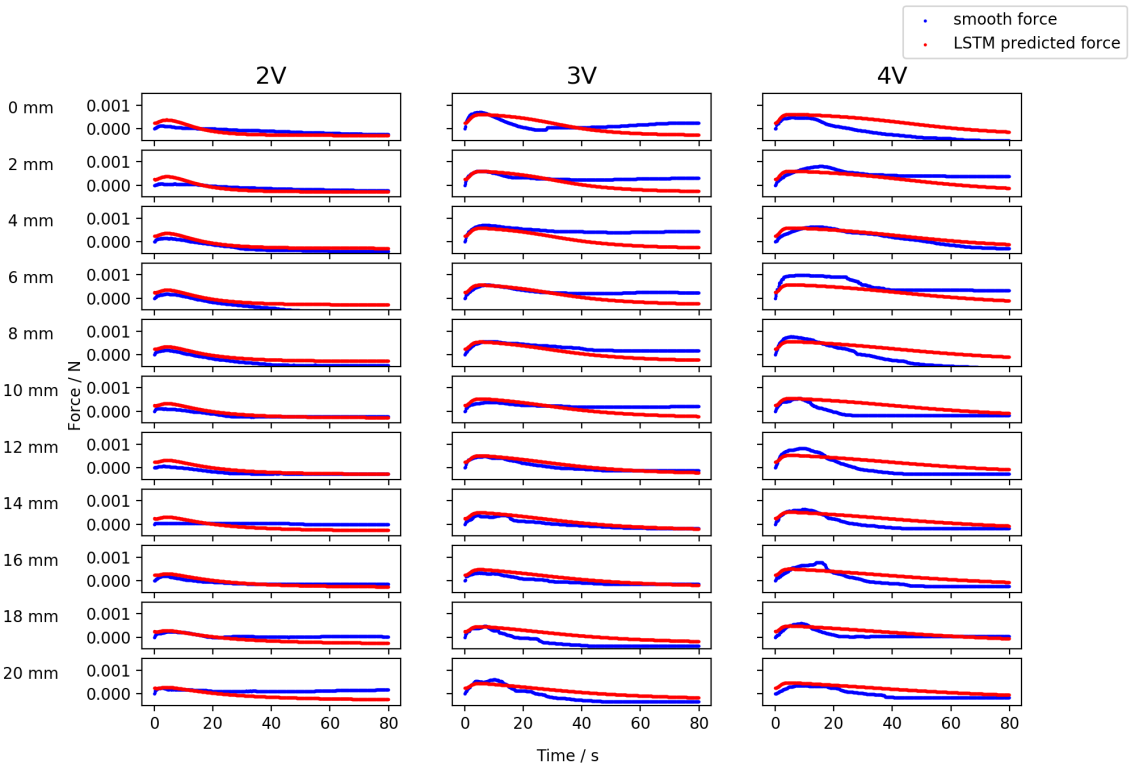

Figure A.3: LSTM predictions overlaid on test data