A Comparison of Systems Identification Techniques for Dynamic Modelling of EMG to Force Data

System Identification is a process of deriving a mathematical model of a predefined part of the world, using observations [3]. The context of this investigation is force control of prosthetic devices using a dynamic model that relates surface electromyography (EMG) signals to generated output forces [1]. EMG-force models are commonly used to extract ‘user-intention’ from surface EMG signals and transform the EMG signals into a more useful force signal that can be used to control prosthetic devices, e.g. as the reference signal to a motor in a prosthetic.

During the course of this investigation three different methods for mapping the input EMG signals to the output force and deriving a generalised model of this mapping will be implemented and their performances will be evaluated and analysed using established metrics such as the mean absolute error and root mean squared error between the desired reference output produced by an actual human appendage, and those produced by the various models.

The three system identification techniques under consideration are:
1. Standard Systems Identification Algorithms (MATLAB SystemsID Toolbox)
2. Adaptive Fuzzy Network-based Fuzzy Inference System (MATLAB ANFIS Toolbox)
3. Neural Network Identification (MATLAB Neural Fitting Toolbox)

The standard algorithms will serve as the experimental control/benchmark.
The data utilised was obtained from the NinaPro Database on Non-Invasive Adaptive Prosthetics [2].

EMG Signal Processing: Low-Pass Filter Design

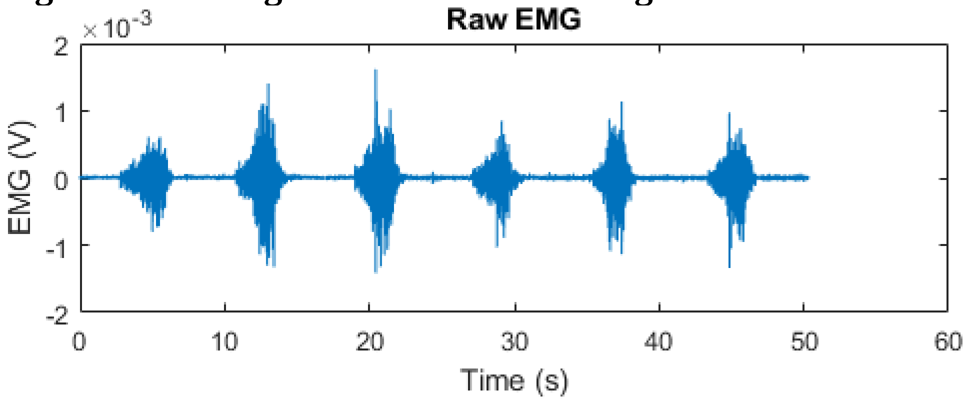

fig.1

Before EMG Data can be used in generating systems model, it must first be digitally processed. This is done to remove the unwanted elements of the signal and to make it easier to map to an output function. The raw EMG data is first rectified as only the positive elements are useful, this rectified data is then passed through a series of Low-pass filters in a Biquad configuration. The signal is then normalised to fit in range 0 to 1.

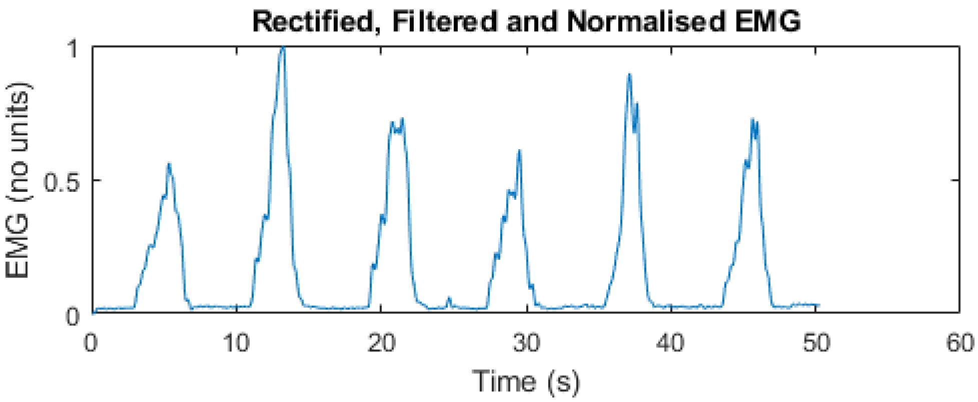

fig.2

The filter was designed leveraging the built in MATLAB Filter design toolbox:

```
Fs = 2000;              % Sampling Frequency
Fpass = 2;              % Passband Frequency
Fstop = 10;             % Stopband Frequency
Apass = 1;              % Passband Ripple (dB)
Astop = 80;             % Stopband Attenuation (dB)
match = 'stopband';     % Band to match exactly

% Construct an FDESIGN object and call its BUTTER method.
h  = fdesign.lowpass(Fpass, Fstop, Apass, Astop, Fs);
Hd = design(h, 'butter', 'MatchExactly', match);

rEMG = abs(emg(:,2));         % EMG signal rectification
filtEMG = filter(Hd,rEMG);    % low pass filter
maxEMG = max(filtEMG);        % max EMG value
normEMG = filtEMG./maxEMG;    % signal normalisation
```

fig.3

The filter is then implemented in Simulink via these Block

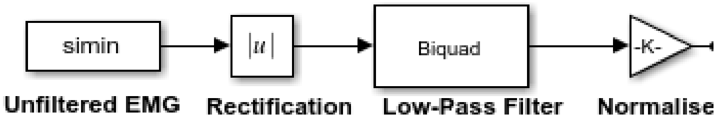

fig.4

The normalised filter output is then used as the input for the various systems identification procedures.

System Identification Techniques

In system identification many different approaches can be applied depending on the prior information about the system to be modelled available. The relation between inputs and outputs of a system is formulated in a general mathematical form. This mathematical form defines the structure of the model and defines a set of parameters, the values of which are determined during the identification process. The techniques selected for deriving the EMG to Force model all utilises a black box approach. In black box modelling, an input – output mapping of the system to be modelled can be described by a continuous function y=f(x) but where any aspect eluding to the structure of the function is unknown and models are constructed based on observations about the system's behaviour, the input-output data, alone [3].

ANFIS

An Adaptive Fuzzy Network based Inference System or ANFIS is a hybrid architecture that combines the interpolating abilities of an Artificial Neural Network to optimise the TSK-based fuzzy rules of a fuzzy system through the use of input and output data. In order to utilise ANFIS input and output data most be provided in this case, and with all the system identification models the input data is the normalised EMG, and the desired output is the force generated [4].

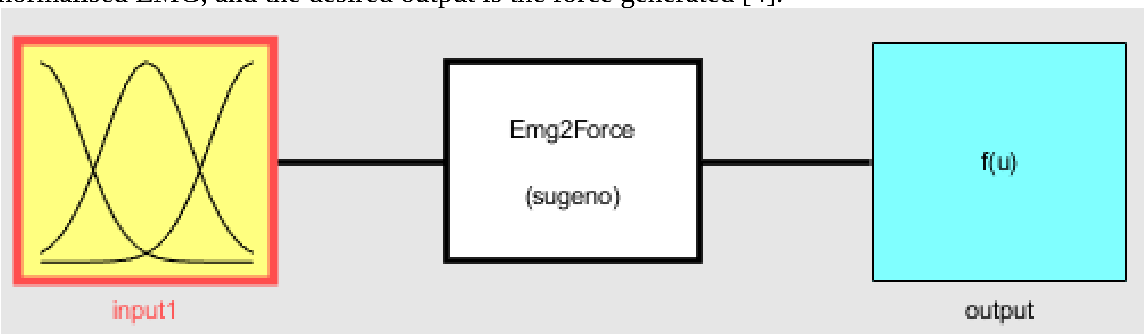

fig.5

Neural Network Identification

Neural networks are distributed information processing systems made up of a great number of highly interconnected identical or similar simple processing units, which are doing local processing, and are arranged in ordered topology whereby its knowledge is acquired from its environment through an adaptive process called learning. Learning is an iterative process, when the values of the network are adjusted step by step until we can achieve the best fit between observed data and the model. The construction of neural networks uses this iterative process instead of applying conventional computational steps. Most neural network architectures mimic biological neural networks. The neural networks used for system modelling usually apply a basic processing element, the perceptron which is a nonlinear model of a neuron.
This simple neural model consists of two basic parts:

- A Linear Combiner
- A Nonlinear Activation Function

The linear combiner computes the scalar product of the input vector of the neuron and a parameter or weight vector [3].

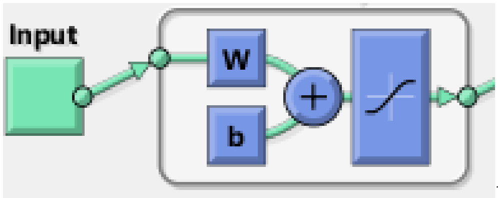

fig.6

Experimental Results

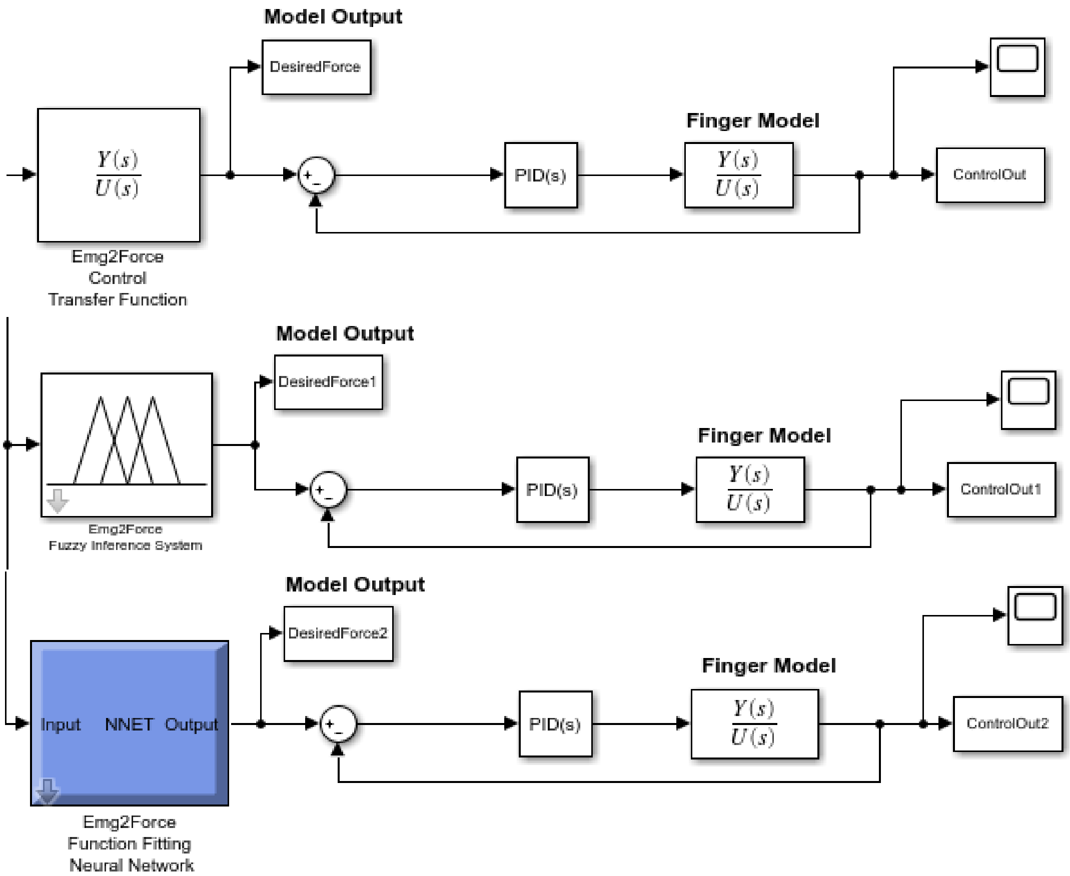

fig.7

Result Plots

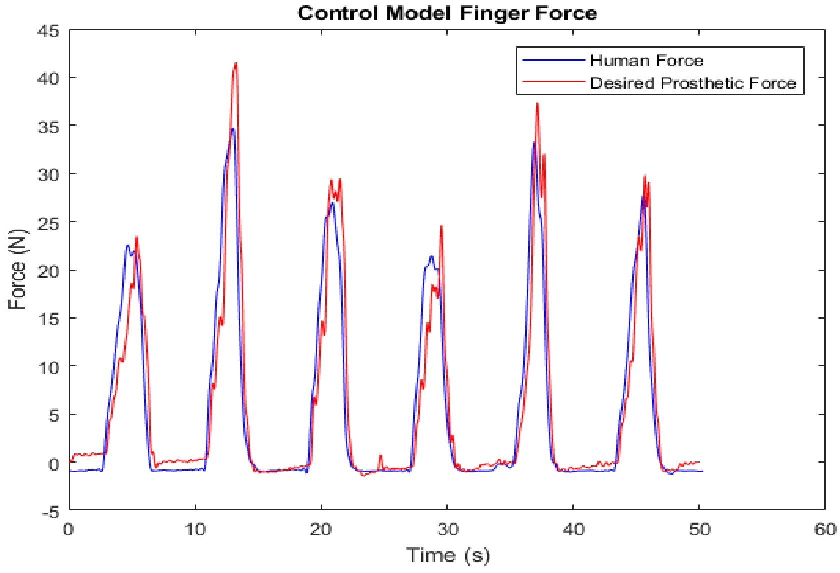

fig8

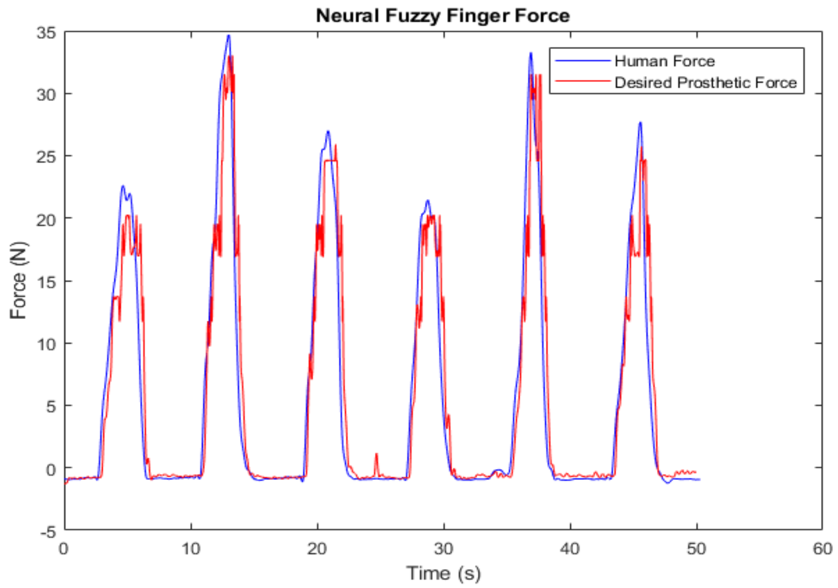

fig.9

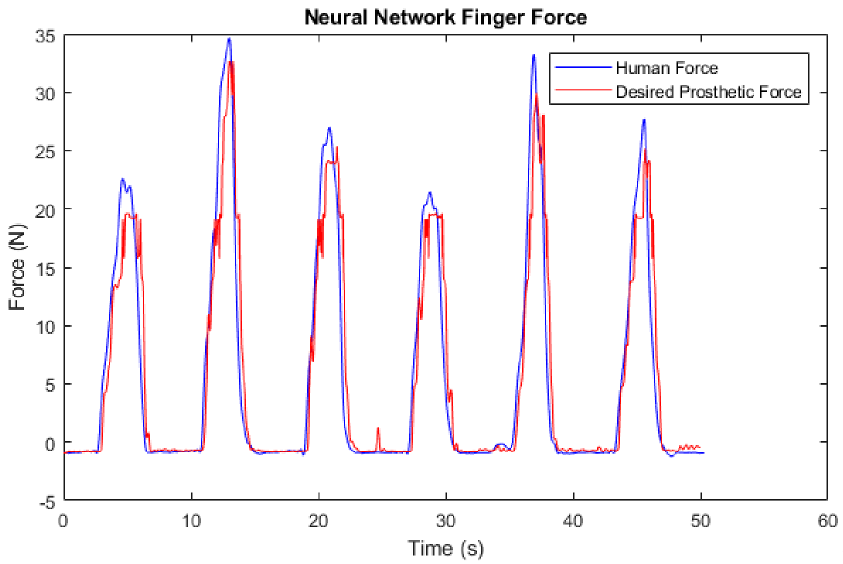

fig.10

The plots above show the human generated force against the forces derived from the 3 models. In analysing the performance of the models, I calculated the Mean Average Error (MAE) and Root Mean Squared Error (RMSE) between the human force value and those generated by each identification method to show closeness of fit as shown below:

```
ControlError = force(1:length(DesiredForce.Data),2) - DesiredForce.Data;
ANFISError = force(1:length(DesiredForce1.Data),2) - DesiredForce1.Data;
NNError = force(1:length(DesiredForce2.Data),2) - DesiredForce2.Data;

MeanAbsErrorControl = mae(ControlError);
MeanAbsErrorANFIS = mae(ANFISError);
MeanAbsErrorNN = mae(NNError);

RootMeanSqErrorControl = sqrt(mean((ControlError).^2));
RootMeanSqErrorANFIS = sqrt(mean((ANFISError).^2));
RootMeanSqErrorNN = sqrt(mean((NNError).^2));

PercentImprovementANFIS = 100*((RootMeanSqErrorControl-RootMeanSqErrorANFIS)/RootMeanSqErrorControl);
PercentImprovementNN = 100*((RootMeanSqErrorControl-RootMeanSqErrorNN)/RootMeanSqErrorControl);
```

fig.11

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It’s the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

fig.12

RMSE is a quadratic scoring rule that also measures the average magnitude of the error. It’s the square root of the average of squared differences between prediction and actual observation.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

fig.13

RMSE gives a relatively high weight to large errors. This means the RMSE is more useful when large errors are particularly undesirable, such as when precise prosthetic actuator commands are required [5].

The MAE between the Systems Identification toolbox Control model and the human finger force was: 2.4533

The RMSE was: 4.0005

Using the above RMSE value as a benchmark, I was also able to calculate the Percentage Improvement (PI), if any, of both the ANFIS and NN fitting methods over the Control.

The MAE of the ANFIS Method was: 1.8839

The RMSE of the ANFIS method was: 3.4104

This led to a PI of: 14.7522% over the Control

The MAE of the NN Method was: 1.8769

The RMSE of the NN method was: 3.3442

This led to a PI of: 16.4061% over the Control

Conclusion

In conclusion, the universal approximation abilities of both the ANFIS architecture, with 10 non-linear fuzzy membership functions, and the Neural Network with 10 hidden layer nodes able to generate a model mapping the EMG to the desired force better than that of the control model when used with the training data.

Below is a plot of the non linear function derived by the ANFIS system which describes the data mapping.

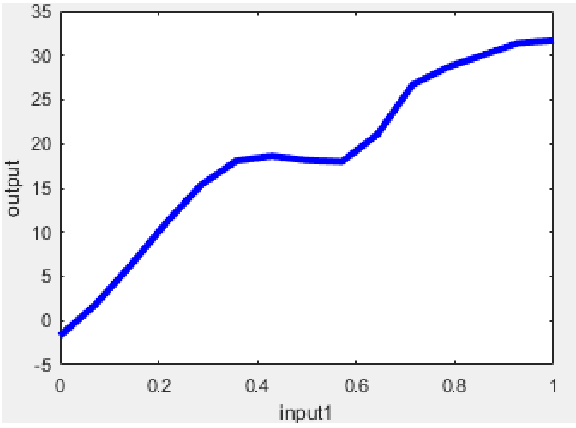

fig.14

The results show, with a percentage improvement of 16.4061, that the neural network was, giving the constraints of then nodes, better able to map the input EMG to the output force when compared to the 14.7522% improvement managed by the ANFIS architecture.

I believe this 2% disparity is due to the 25 epoch limit I had set for the ANFIS training in order to avoid over training the model and maintain generalisation. whereas the Neural Networks epochs where automatically set and adjusted, with a limit of 1000 epochs, to find the minimal error. I believe that giving enough cycles it would be possible to train the ANFIS architecture to a higher performance level.

References

[1] Castellini, C., & van der Smagt, P. (2009). Surface EMG in advanced hand prosthetics. Biological cybernetics, 100(1), 35-47

[2] NinaPro (Non-Invasive Adaptive Prosthetics) project: http://ninapro.hevs.ch/

[3] Gábor HORVÁTH, Department of Measurement and Information Systems Budapest University of Technology and Economics Magyar, Neural Networks in System Identification.

[4] Erik Cuevas, Primitivo D´ıaz ,Omar Avalos, Daniel Zald´ıvar, Marco Perez-Cisneros, Nonlinear system identification based on ANFIS-Hammerstein model using Gravitational search algorithm.

[5] MAE and RMSE—Which Metric is Better?
https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d