

## Methodology for developing a neural network leaf spring model

**Corresponding Author:** Dr. Cor-Jacques Kat  
**Position:** Senior lecturer  
**Department:** Department of Mechanical and Aeronautical Engineering  
**Institution:** University of Pretoria  
**Postal address:** Department of Mechanical and Aeronautical Engineering,  
University of Pretoria, Private Bag x20 Hatfield, 0028, South  
Africa  
**E-mail:** cor-jacques.kat@up.ac.za

**Author:** Dr. Jennifer L. Johrendt  
**Position:** Associate professor  
**Department:** Department Mechanical, Automotive, and Materials  
Engineering  
**Institution:** University of Windsor  
**Postal address:** 401 Sunset Avenue, Windsor, ON, Canada, N9B 3P4  
**E-mail:** j.johrendt@uwindsor.ca

**Author:** Prof. Pieter Schalk Els  
**Position:** Associate professor  
**Department:** Department of Mechanical and Aeronautical Engineering  
**Institution:** University of Pretoria  
**Postal address:** Department of Mechanical and Aeronautical Engineering,  
University of Pretoria, Private Bag x20 Hatfield, 0028, South  
Africa  
**E-mail:** schalk.els@up.ac.za

## **Biographical notes**

**Dr Kat** holds degrees in Mechanical Engineering from the University of Pretoria (B.Eng, B.Eng (Hons), M.Eng, Ph.D). His Masters and Ph.D research was focused on vehicle system and component modelling and testing. His professional experience includes work as a research and development engineer in the automotive industry and he is currently employed by the University of Pretoria as a senior lecturer. His current research includes vehicle system modelling and testing, human factors in vehicle dynamics and biomechanics.

**Dr. Johrendt** holds degrees in Mathematics and Engineering from Queen's University in Kingston (B.Sc., M.A.Sc.) and Mechanical Engineering from the University of Windsor (Ph.D.). She began her career in automotive research as a product development engineer with Chrysler Canada (now FCA Canada) conducting vehicle durability studies and associated research activities in the Road Test Simulation laboratory. In 2005, she joined the University of Windsor as an Experiential Learning Specialist, focusing on teaching and educational research in hands-on learning and cooperative education as it relates to undergraduate engineering. Dr. Johrendt is an Associate Professor of Mechanical and Automotive Engineering at the University of Windsor in Ontario, Canada. She has applied neural network modelling techniques to accelerate product development processes in addition to models describing driver behaviour as it relates to risk and composites processing parameters and their effect on material properties.

**Prof Els** worked in industry, developing and testing military wheeled vehicles for 5 years, where he was actively involved in new technology projects and applied research. Highlights include semi-active dampers for heavy vehicles, including the world's first semi-active hydraulic rotary damper (with Horstman Defence). In 1999 he joined the Department of Mechanical and Aeronautical Engineering at the University of Pretoria as a permanent staff member and obtained a PhD in 2006. His current research interests include semi-active dampers, semi-active springs and height adjustment to improve ride comfort, handling, rollover propensity and life of heavy off-road vehicles.

**Abstract** –This paper describes the development of a neural network that is able to emulate the vertical force-displacement behaviour of a leaf spring. Special emphasis is placed on aspects that affect the predictive capability of a neural network such as type, structure, inputs and ability to generalize. These aspects are investigated in order to enable the effective use of it to model leaf spring behaviour. The results show that with the correct selection of inputs and network architecture, the neural network’s ability to generalize can be improved and also reduce the required training data. The resulting 2-15-1 feed forward neural network is shown to generalize well and requires minimal data to be trained. Experimental data was used to train and validate the network. The methodology followed is not limited to the application of leaf springs only but should apply to various other applications especially ones with similar non-linear characteristics.

**Keywords** – leaf spring modelling; multi-leaf spring; neural networks; generalization; experimental training data; experimental validation

## 1. Introduction

Virtual prototyping, or simulation, has been playing an increasing role in the vehicle development process and is used in combination with physical prototyping and testing. In order to use virtual prototyping successfully in the development process, accurate and efficient simulation models of the various systems, subsystems and components are required. This paper will focus on the component level with attention given to the modelling of the vertical behaviour of a multi-leaf spring for use in vehicle dynamic simulations.

Even though leaf springs are frequently used in practice they still hold great challenges in creating accurate mathematical models. Many different methods exist that can be used to model the leaf spring’s force-displacement characteristics. Fancher et al. (1980) and Cebon

(1986) present an **analytical model** that uses algebraic equations that are able to fit the behaviour of the experimentally obtained characteristics of the leaf-spring. An **equivalent model** which models the leaf spring as an equivalent system using a vertical spring (or a combination of series and parallel springs) with damper and/or friction elements is used in studies by Hoyle (2004) and ElMadany (1987). Another method discretises the leaf spring into rigid elements. The rigid elements are connected by torsional springs and dampers. This method is known as the **discrete or finite segment method** and examples of its use in vehicle simulations can be found in Huhtala et al. (1994), Yang et al. (2007), Jayakumar et al. (2005) and Ekici (2005). A similar technique to the discrete method is the **finite element method** which models the leaf spring using finite element techniques. The use of finite element leaf spring models can be found in Moon et al. (2007), Li and Li (2004) and Qin et al. (2002). Other techniques that exist are the **elasto-plastic leaf spring model** (Kat and Els, 2011) and models that use **neural networks** (NN) to emulate the behaviour of the leaf spring (Ghazi Zadeh et al., 2000). These models have various advantages and disadvantages. The comparison of these models, w.r.t. accuracy, computational efficiency and ease of parameterization, is outside the scope of this paper. This study will consider the neural network technique. It is expected that this technique will be computationally efficient and allow for easy parameterization.

The study by Ghazi Zadeh et al. (2000) showed that a neural network can be trained to accurately emulate the typical nonlinear, hysteretic behaviour of a leaf spring. The leaf spring data considered in their study was derived from an analytical model (Fancher et al., 1980) representing the typical force-displacement characteristics of a leaf spring. No experimental leaf spring data was used. They used two recurrent neural networks with similar architecture; one emulating the loading behaviour and the other one the unloading behaviour of the leaf spring. A switching algorithm was used to determine which one of the networks should be

used depending on whether the spring is loaded or unloaded. Each of the loading and unloading neural networks has an architecture of 3 x 10 x 20 x 1. The neural network was shown to approximate the analytical model well.

In the current study, experimental data obtained from a physical multi-leaf spring is used to train and validate a neural network model. This data contains typical experimental noise. Aspects that affect the predictive capability of the neural network are investigated in order to enable the effective use of this approach to model the vertical behaviour of leaf springs. Particular focus is placed on the methodology for developing a suitable neural network including type, structure, appropriate inputs and ability to generalize. An aim of the methodology is to develop a neural network model that requires minimal (experimental) data to train it.

## **2. Multi-leaf spring characteristics**

The multi-leaf spring considered in this study consists of eight blades with uniform cross-section. The total weight of the multi-leaf spring is 49kg. Figure 1 shows the in-service setup of the spring. Figure 2 shows a typical force-displacement characteristic of a multi-leaf spring in compression and tension. In general, the multi-leaf spring will seldom be in tension as this occurs when the wheels lose contact with the road. This situation may have a higher possibility of occurring in off-road and very rough road conditions than on smooth on-road conditions. The two major aspects of the leaf spring that a model has to capture for the compression part of the cycle are: 1) the spring stiffness and 2) the hysteresis behaviour.

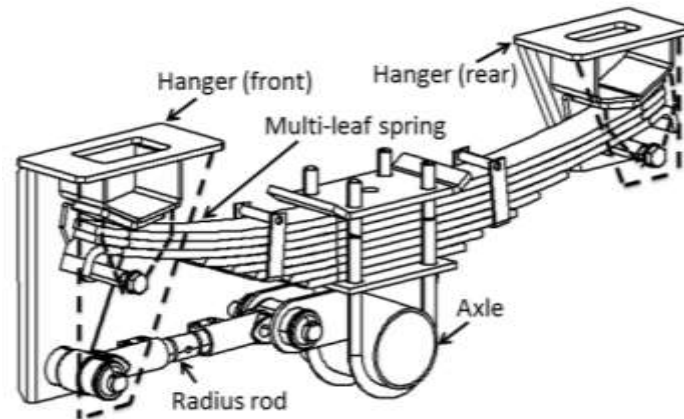


Figure 1. In-service setup of the leaf spring

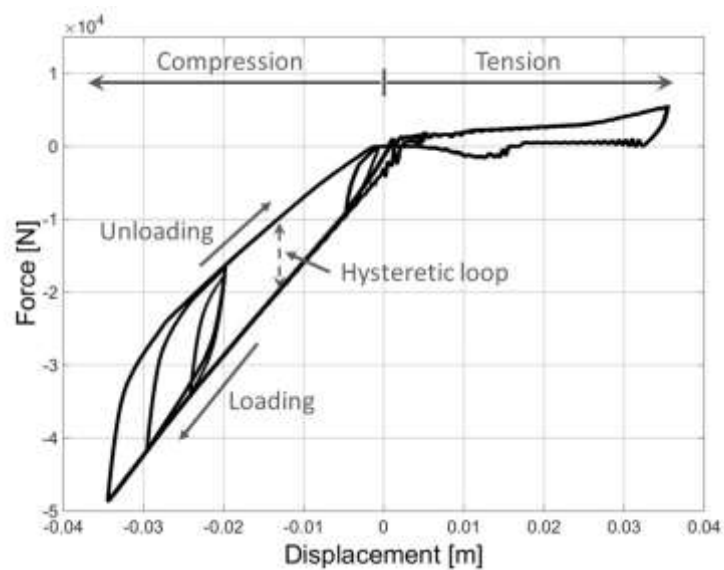


Figure 2. Typical force-displacement characteristic of a multi-leaf spring

According to Fancher et al. (1980) the force exerted by the leaf spring when deflected, is a function of the loading on the spring and the amplitude of the imposed displacement. Other factors include leaf beam bending stiffness, friction between each leaf, loaded length, as well as boundary friction (between the leaf and the supporting structure) and leaf material structural damping. Fancher et al. (1980) and Cebon (1986) found that the spring force is not dependent on the frequency of the imposed displacement for excitation frequencies up to 18Hz. This implies that the force displacement characteristics do not need to be obtained at

different input frequencies. Figure 3 shows the force-displacement characteristics of the multi-leaf spring used in this study when the amplitude of the displacement input is kept the same but the excitation frequency is swept from 0.05 Hz to 4 Hz. From this figure it can be observed that the force-displacement characteristic stays the same irrespective of the excitation frequency in that frequency range. This is similar to the findings of Fancher et al. (1980) and Cebon (1986).

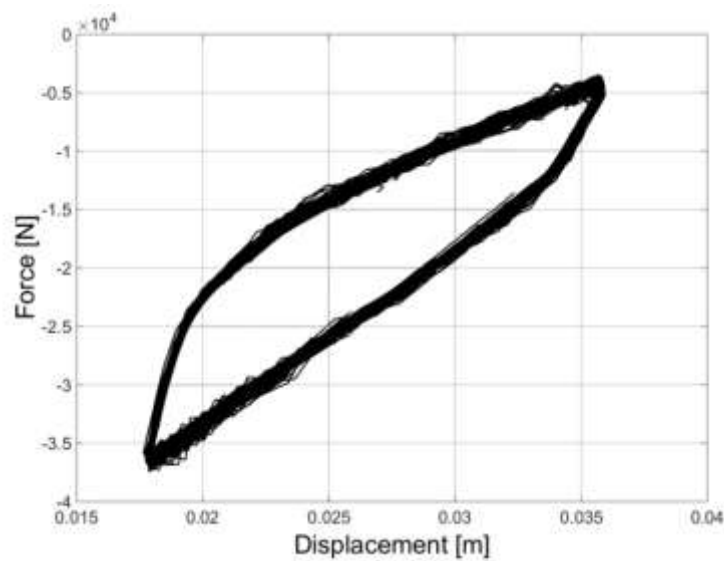


Figure 3. Force-displacement characteristic of a multi-leaf spring subjected to a sinusoidal displacement with frequencies ranging from 0.05 Hz to 4 Hz

### 3. Artificial neural networks

Artificial neural networks are inspired by the biological networks found in the brain. Artificial neural networks are mathematical simplifications of the biological counterparts on which they are based. For details on biological neurons and the networks they form the reader is referred to the book by Müller et al. (1995). A model of a simple artificial neuron with one input and one output is shown in Figure 4 which has the following mathematical representation for the output of the neuron:

$$a = f(wp+b)$$

Similar to neurons being the building blocks of biological networks, the artificial neurons are the building blocks of artificial neural networks. The neuron input  $p$  is multiplied by the weight  $w$  to form the term  $wp$  which is sent to the summer. The bias value,  $b$ , is also sent to the summer and added to the term  $wp$  to form the value  $n$  which is sent to the transfer function (or activation function)  $f$  and produces the neuron output  $a$ . The transfer function  $f$  may be a linear or nonlinear function with the most common transfer functions being a hard limit transfer function, a linear transfer function and a log-sigmoid transfer function. Transfer functions that are continuously differentiable are desirable in neural networks as they allow for the back propagation of the error during the training phase which is necessary for weight and bias adjustments to achieve convergence. Modelling non-linear behaviour requires the use of non-linear transfer functions. The neuron shown in Figure 4 is a single-input neuron and can be extended to have multiple inputs. These multiple input neurons can be connected in parallel to form layers and layers connected in series to form various network architectures (Hagan et al., 1996). The network architecture that will be used in this study is a feed forward multilayer network. Other networks that incorporate a feedback of an output to an input are called recurrent networks.

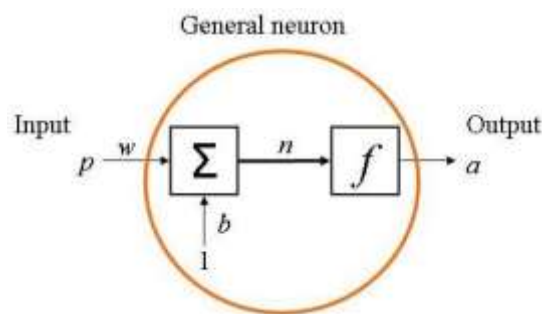


Figure 4. Simple artificial neuron



The combination of various neurons into networks will result in a set of weights ( $w$ ) and biases ( $b$ ). The parameters  $w$  and  $b$  are the variables of the neural network that are adjusted by a learning rule so that the network's input/output relationship reflects that of the data used for training the network. The error between the network output ( $a$ ) and the targets that are given to the neural network via the training set is quantified using a performance index such as the Mean Square Error (MSE). The MSE is minimised by means of a back propagation algorithm wherein the network parameters (i.e. the weights ( $w$ ) and biases ( $b$ )) are adjusted in order to minimize the error between the network output and the target values. Further details about this procedure can be found in Hagan et al. (1996), Dreyfus (2005) and Johrendt et al. (2008).

This brief introduction on neural networks has discussed the concept, the different architectures and how they are trained. It has not discussed how to design a neural network. The application of a neural network to emulate the behaviour of the multi-leaf spring is essentially the approximation of a non-linear function. It is known that neural networks can approximate any non-linear function (Dreyfus, 2005). Dreyfus (2005) states that the following aspects have to be considered in designing a neural network that will be able to approximate a given function:

1. Find the relevant inputs
2. Collect data necessary for training and testing of the neural network
3. Train the network
4. Find appropriate complexity of the network (i.e. number of layers, number of neurons per layer)
5. Assess generalization ability of neural network

These guidelines will be followed in order to obtain a neural network that is able to emulate the vertical force-displacement characteristics of a multi-leaf spring. The five aspects are discussed in the following paragraphs.

### **3.1. Relevant inputs and network architecture**

It was mentioned in section 2 that the force exerted by the leaf spring is a function of various factors. The neural network is not a physics-based approach to modelling of a physical system, so the most effective use of the neural network is to use it to relate the input of interest (i.e. displacement) to the output of interest (i.e. spring force) of the leaf spring. For the purpose of this research, the leaf spring displacement will be used to predict the spring force, as the single model output. The combined effect of the various factors is taken into account by the neural network, even if they're not explicitly used as network inputs.

The inputs used by Ghazi Zadeh et al. (2000) were the deflection at the current time step, the absolute value of the deflection change and the force at the previous time step (this was the recurrent input). The selection of these input parameters was based on the analytical equation of Fancher et al. (1980). Deflection, or displacement, of the spring will also be used as one of the input parameters to the neural network in this study. Velocity will be used as the other input parameter. The choice of using the displacement as input is obvious as any spring develops a force due to it being deflected. The velocity is chosen as it is expected to indicate to the neural network whether it's being loaded or unloaded, as we know that the hysteretic nature of the spring force-displacement would make it possible to associate one or more values of spring force for each spring deflection point.

There is no clear method by which the architecture of a neural network should be determined, but there are some general guidelines which can be followed. For instance, the number of neurons that should be used in order to obtain good generalization from the neural network

should be enough to capture the behaviour but no more (Hagan et al., 1996; Dreyfus, 2005). The nonlinear function observed in the force-displacement characteristic is not that complex and it is assumed that a single hidden layer of 35 neurons will enable the neural network to emulate the non-linear behaviour and will therefore serve as a starting point. A higher number of neurons (and layers) will result in a higher level of nonlinearity in the network.

The higher nonlinearity may be needed to capture the nonlinear relationship between the inputs and outputs of the system being emulated. However, a higher level of nonlinearity may result in the neural network having too many parameters (i.e. weights ( $w$ ) and biases ( $b$ )), causing the output of the network to fit the training data very accurately, including the noise in the data, but provide an inaccurate output to inputs not in the training data (Dreyfus, 2005). This is known as overfitting. Overfitting can affect the ability of the network to generalize. Generalization refers to the ability of the neural network to give a sufficiently accurate prediction of the system behaviour for situations that were not present in the training data (Dreyfus, 2005). This will be discussed further in subsequent sections.

The initial network is a feed forward neural network with an architecture of two inputs, one hidden layer with 35 neurons and one output being the spring force (referred to as a 2-35-1 network). The transfer functions used in the hidden layer are tan-sigmoid (tansig) functions with the output layer using a linear (purelin) transfer function. This is a much simpler neural network than the one used by Ghazi Zadeh et al. (2000) described earlier. An advantage of using a feed forward network over a recurrent network is that the training of the feed forward network is faster.

### **3.2. Collect necessary data for training and testing of neural network**

Numerical data was generated from an analytical function (Fancher et al, 1980) in order to train and test the neural network in Ghazi Zadeh et al. (2000). In this study experimental data

(with its inherent noise) was obtained and used to train, test and validate the neural network. The experimental setup used to obtain the force-displacement characteristic of the multi-leaf spring in compression is shown in Figure 5. The leaf spring is connected to the dummy axle which is connected through an interface plate and U-joint to the actuator via a load cell. The leaf spring is simply supported on bearings by the front and rear supports. This is a simplified version of the in-service setup of the leaf spring shown in Figure 1. The actuator imposes a predefined displacement onto the multi-leaf spring to which the leaf spring then exerts a corresponding force. The force exerted by the leaf spring is measured by the load cell that is located between the actuator and the multi-leaf spring.

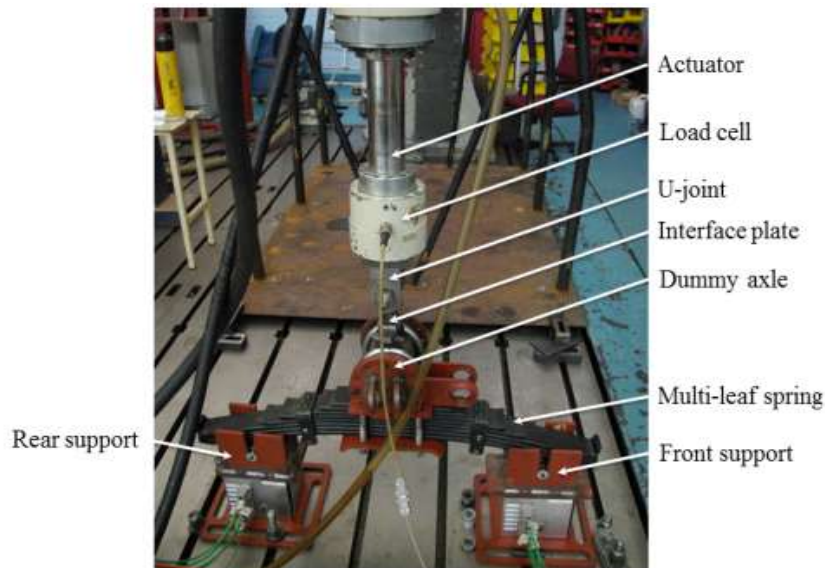


Figure 5. Experimental setup of the multi-leaf spring.

Figure 6(a) shows the force-displacement characteristic of the multi-leaf spring that is obtained when subjected to the displacement signal shown in Figure 6(b). The frequency of the displacement signal used during testing was 0.25Hz. The inertial loads resulting from the maximum acceleration and mass of components, was negligible. The displacement signal results in three observable loops on the force-displacement characteristic, i.e. an outer, inner

and end loop. Each closed loop corresponds to the force produced by the leaf spring when subjected to a reciprocating deflection input of different amplitudes. The three loops can be associated with three different vertical loads with different displacement amplitudes. The training set that will be used to train the neural network will be constructed from the experimental data as discussed in the following section.

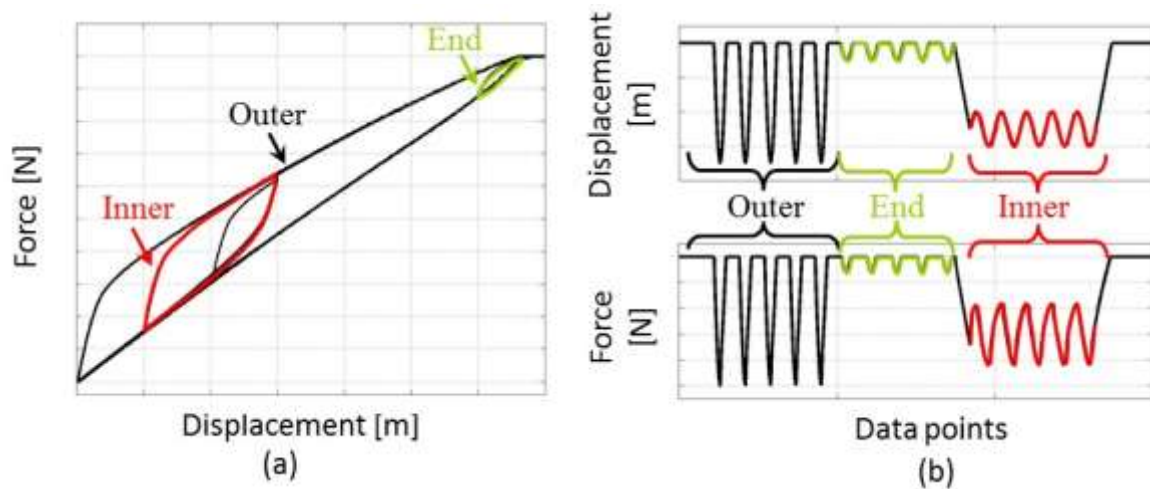


Figure 6. Three different loops shown on (a) the force-displacement characteristic and (b) force and displacement versus data point plots

### 3.3. Training the network

Two training sets were constructed from the experimental data. Training set 1 used data that consisted of only the outer force-displacement loop. Training set 2 used data that consisted of all the loops in the force-displacement characteristic (see Figure 6). The Levenberg-Marquardt training algorithm is used in MATLAB<sup>®</sup> as it is regarded as being the fastest algorithm for small to medium sized networks (Mathworks, R2016a). The performance index used is the Mean Square Error. Early stopping was also used during the training of the network to avoid overfitting and ensure that good generalization is achieved by the network.

Two 2-35-1 neural networks were trained, one using Training set 1 and the other using Training set 2. The two neural networks will be referred to as 2-35-1 TS1 and 2-35-1 TS2, respectively. After the two networks have been trained, both are simulated by giving them the full displacement input signal shown in Figure 6 (b). Figure 7(a) shows the result of the 2-35-1 TS1 neural network compared to the experimental data of the actual leaf spring subjected to the same displacement. It can be observed that the neural network emulates the leaf spring well for the outer loop for which it was trained. However, for the other loops it gives inaccurate predictions. When we use the neural network that was trained using all the loops in the force-displacement characteristic (i.e. 2-35-1 TS2) and compare its prediction to the experimentally measured force-displacement characteristic (see Figure 7(b), it can be observed that this neural network is able to better predict the force for all the loops. From the results in Figure 7 it can be concluded that the neural network has difficulty in generalizing. In other words, it is not able to correctly predict the force for inputs (in this case displacements and velocities) it was not trained with. This implies that in order for the neural network to be able to emulate the behaviour of the leaf spring it has to be trained with data over its entire working range. It would be more advantageous to have a network with a better ability to generalize as this would mean that less experimental data is required and implies that a less complex experimental testing procedure would be needed. The ability of the network to generalize and the noise present on the network's predictions are addressed further in subsequent sections.

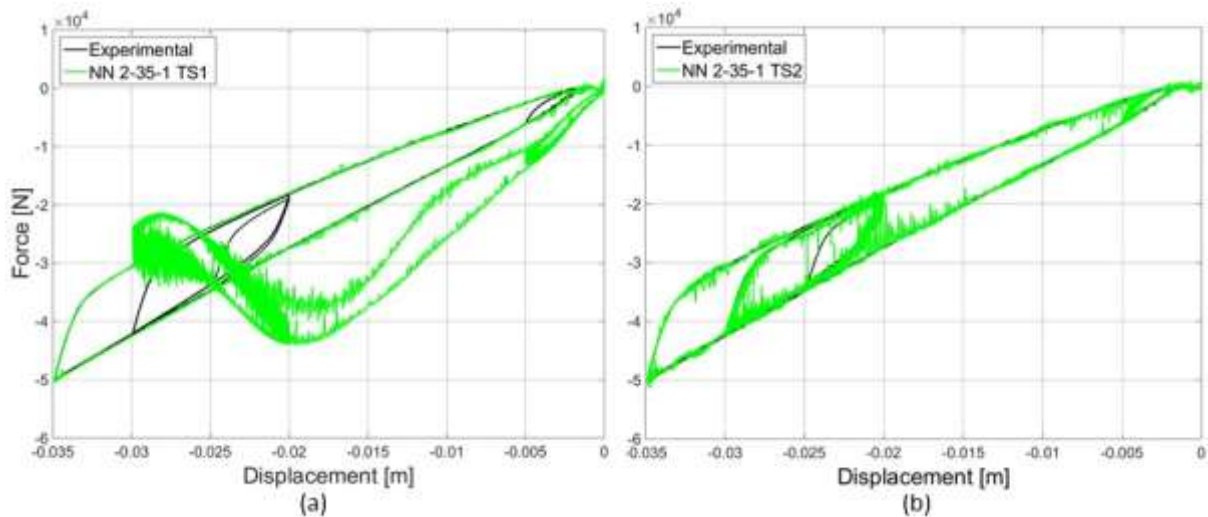


Figure 7. Comparison of experimental data and neural network predictions. NN 2-35-1 trained (a) with outer loop and (b) all loops of force-displacement characteristic

### 3.4. Appropriate complexity of the neural network

The ability of the neural network to generalize and the noise on its predictions may be linked to the complexity of the network. In this case reducing the number of hidden neurons (i.e. degrees of freedom of the network) may address both issues. The 35 hidden neurons in the network were reduced to 15, 9 and 4. Reducing the number of neurons did not reduce the noise on the network's predictions.

The velocity input signal is obtained by differentiating the displacement. When the velocity signal is viewed it is observed that the signal has a lot of noise present. This noise on the velocity signal seems to be the source of the noise observed in the predictions of the neural network in Figure 7. The velocity signal was smoothed by applying a four point moving average to the signal. Figure 8 shows the velocity signal before and after the four point moving average is applied. Figure 9 shows the results obtained from the two 2-35-1 neural networks (2-35-1 TS1 and 2-35-1 TS2) when the four point moving average is applied to the velocity signal. The prediction from the neural network has a lot less noise when applying the four point moving average to the velocity input signal. Note, when comparing the results

of neural network 2-35-1 TS1 between Figure 7(a) and Figure Figure 9(a) it is not only observed that there is less noise on the predictions of the neural network but also a noticeable difference in the prediction of the force associated with the displacements of the inner and end loops. This has to do with the network's ability to generalize, which is discussed in detail in section 3.5.

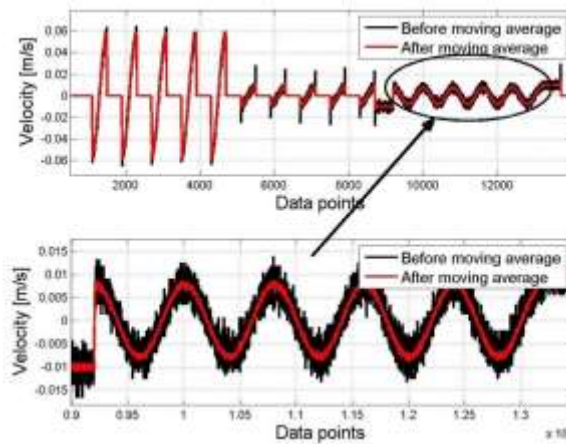


Figure 8. Velocity input signal before and after applying the four-point moving average

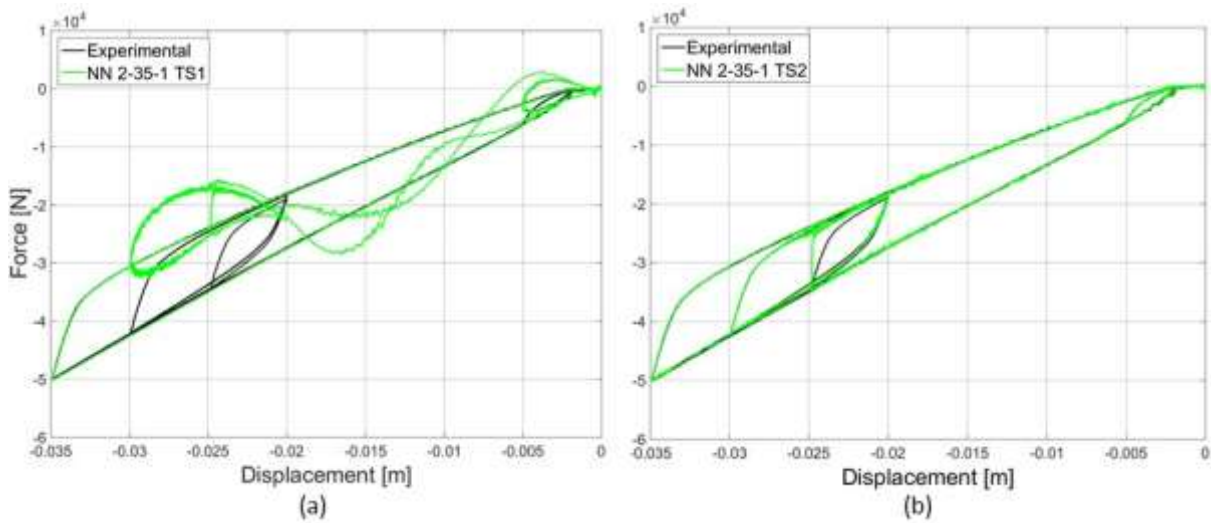


Figure 9. Comparison of experimental data and neural network predictions. Four point moving average applied to the velocity input signal. NN 2-35-1 trained with (a) outer loop and (b) all loops of force-displacement characteristic

The results in Figure 9(b) show that the 2-35-1 network is able to accurately emulate the behaviour of the multi-leaf spring when it is trained with a training set containing input-output data from all the loops. It was observed in Figure 9(a) that the 2-35-1 network had



difficulties in generalizing the behaviour and was unable to correctly predict the response when given inputs that were not included in the training set. As mentioned, generalization is expected to improve when the network is as simple as possible while still being able to adequately represent the training set (Hagan et al., 1996; Dreyfus, 2005).

Therefore, the 2-35-1 network's neurons are reduced. Figure 10 shows how the predictions of the neural network (trained with Training set 2) compares to the experimental data as the number of neurons is reduced. It can be seen that the number of neurons can be reduced until the network has nine neurons at which point the network fails to accurately represent the training data. The network with 15 neurons results in a simple network that is still able to accurately represent the training data. This architecture should therefore give us a network that will be able to generalize better. The ability of the network to generalize is discussed further in the following section.

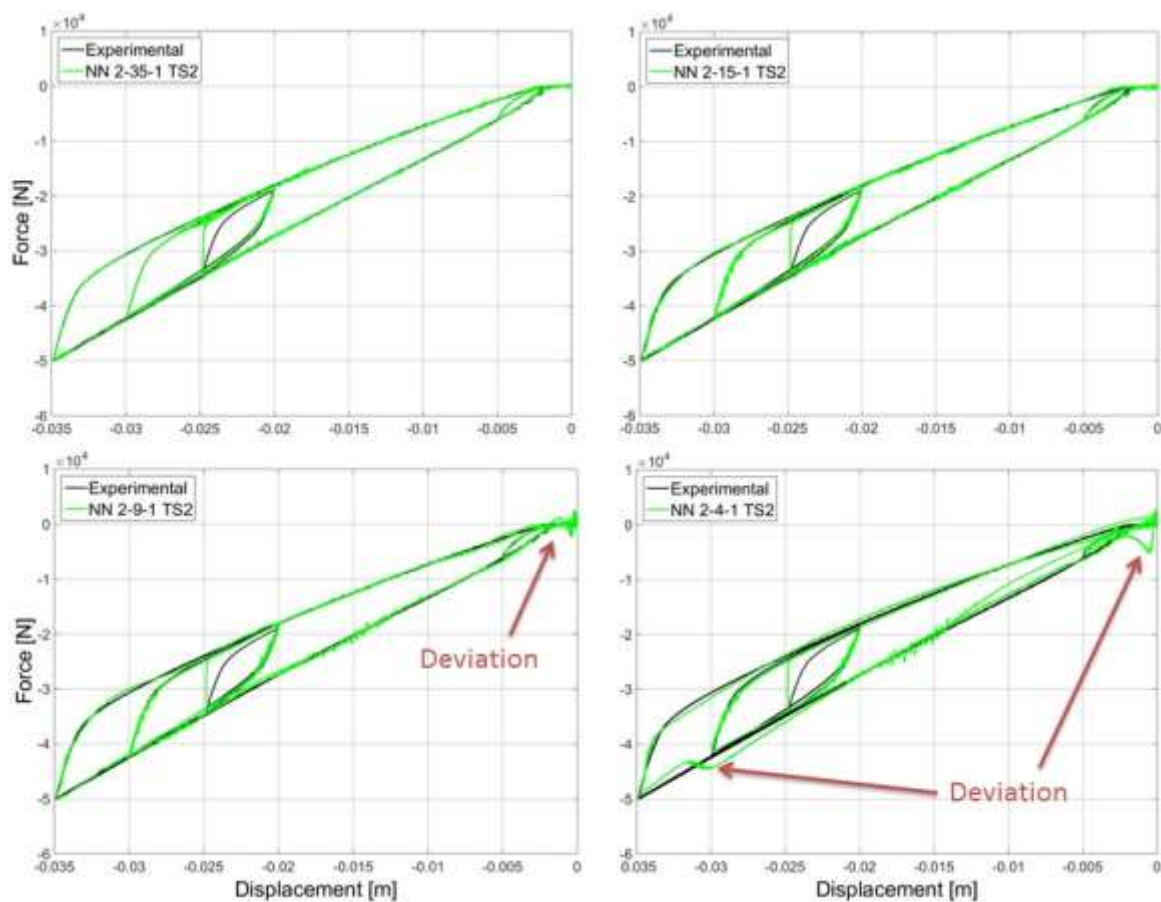


Figure 10. Effect of reducing neurons on the predictions of the neural network

### 3.5. Assess generalization

The results in Figure 9(b) show that the neural network is able to predict the spring force due to a given displacement when it was trained with data that covered the entire working range of the inputs. In Figure 9(a) however, it was shown that the neural network has difficulty in predicting the correct force for inputs not included in the training data. This was referred to as an inability of the neural network to generalize. It should be noted that generalization is often used to refer to the ability of the neural network to interpolate correctly between the supplied training data, whereas, in the case shown in Figure 9(a) the neural network is actually trying to extrapolate. In other words, **generalization with respect to interpolation** means the ability of the neural network to give the correct output value for input values that was not part of the training set but falls within the range of the training set values for the input parameter(s). **Generalization with respect to extrapolation** means the ability of the neural network to give the correct output value for input values that was not part of the training set and falls outside the range of the of the training set values for the input parameter(s).

The difference between the generalization ability of the neural network with respect to interpolation and extrapolation is discussed using a simple example similar to the one used in Hagan et al. (1996). A 1-9-1 feed forward neural network was used to emulate an analytical function ( $y = 1 + \sin(\frac{\pi}{4}x)$ ). After this network was trained it was given a set of inputs.

The inputs include points that were not part of the training set but are within the range of the data in the training set. This network architecture results in the neural network having many more adjustable parameters (weights and biases) in comparison to the data points in the training set and therefore does not generalize well (see Figure 11(a)). To improve the generalization of the neural network the number of neurons can be reduced to give the

simplest model that is able to adequately represent the training set (Hagan et al., 1996; Dreyfus, 2005). When the network architecture is reduced to one having two neurons (i.e. 1-2-1) the generalization is improved as shown in Figure 11(b). Figure 11(b) shows that for any input(s) lying between the data points of the training set, the neural network will give good predictions. With the lower level of nonlinearity of NN 1-2-1 overfitting is avoided. Using the 1-2-1 network, the ability of the neural network to extrapolate is shown in Figure 12. The 1-2-1 network was trained three separate times. From this figure it is clear that the network is not able to predict the correct response for data outside the range used in the training set. It can be noted that for the region that the network has to extrapolate the network predictions differ for each of the three training runs. This is because the training process results in different combinations of values for the weights and biases. The region where each of these networks interpolates yields similar results, but the region where the networks extrapolate show markedly different results. This example indicated that the 1-2-1 network is able to generalize well with respect to interpolation but not extrapolation.

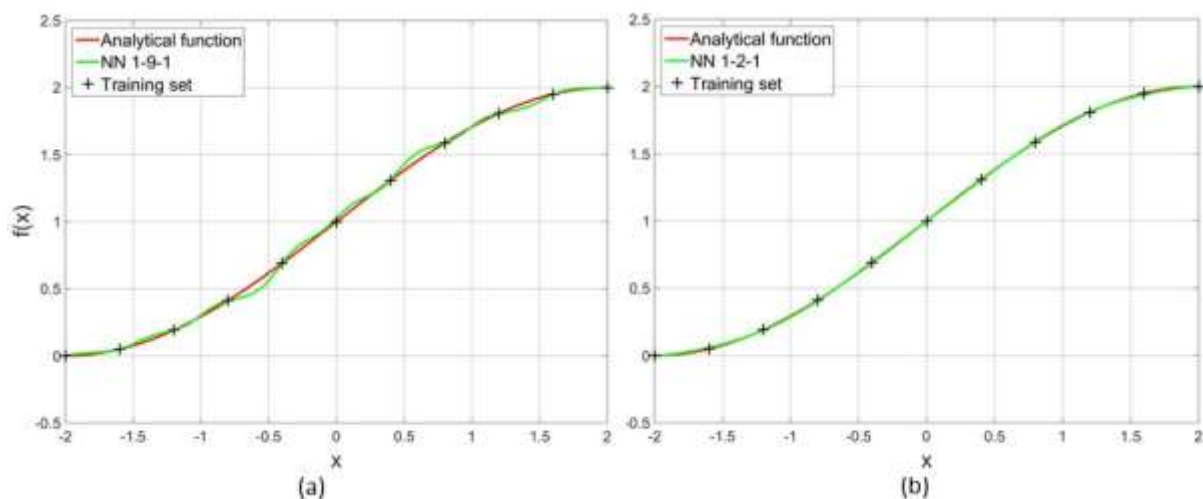


Figure 11. Results of (a) NN 1-9-1 and (b) NN 1-2-1 when interpolating (similar to Hagan, Demuth and Beale (1996))

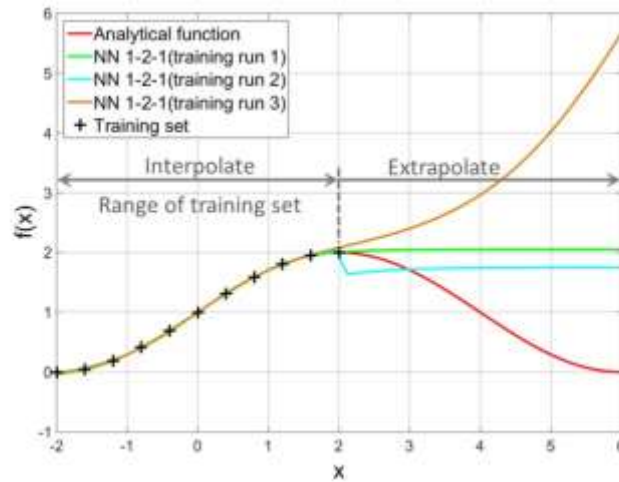


Figure 12. Results of 1-2-1 network when interpolating and extrapolating

The ability of the feed forward 2-35-1 neural network to generalize the behaviour of the multi-leaf spring was shown in Figure 9(a) to be unacceptable. The network's neurons were reduced from 35 to 15 in paragraph 3.4 and it was shown to be the simplest network which was still able to adequately capture the behaviour of the multi-leaf spring. This should improve the generalization ability of the network. The generalization ability of the 2-15-1 network is now assessed with attention given to both the interpolation and extrapolation ability of the network.

### 3.5.1. Generalization with respect to interpolation

Figure 13(a) shows the results for the 2-15-1 TS1 network that was trained on three occasions. It was simulated with a displacement signal having displacements associated with all the loops of the force-displacement characteristic. The displacement signal does not have any of the displacements used in the training set (Training set 1). The results from this figure show that the network has good generalization with respect to interpolation. It is able to predict the outer loop of the force-displacement characteristic well.

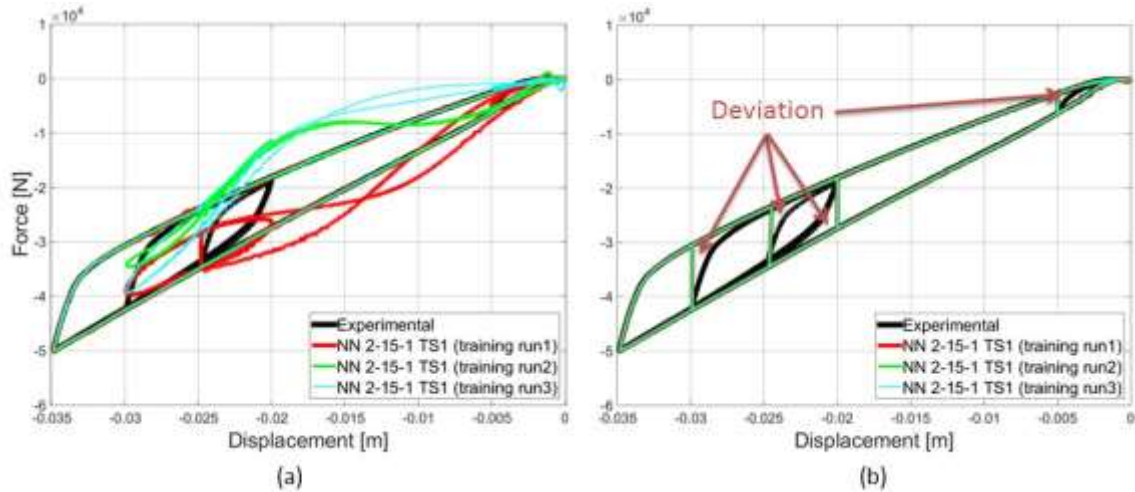


Figure 13. Results of 2-15-1 network using (a) displacement and velocity and (b) displacement and  $\text{sign}(x_i - x_{i-1})$  as inputs, when interpolating and extrapolating

### 3.5.2. Generalization with respect to extrapolation

The predictions of NN 2-15-1 TS1 in Figure 13(a) shows that the network is not able to predict the spring force for displacements associated with the inner and end loops. It should be noted that in using the outer loop for training, the displacement values of the inner and end loops do fall within the range of the training set. However, when the velocity of the three loops are considered the velocity of the inner and end loops fall outside the range of the velocity of the outer loop that was used for training. The network therefore has to extrapolate. Similarly, the network will have to extrapolate when given displacements associated with the outer loop at different excitation frequencies to what was used for the outer loop training data. The experimental force-displacement characteristic (see Figure 6) that is used to generate the training set was obtained using an excitation frequency of 0.25Hz.

Figure 14(a) shows the results when the 2-15-1 TS1 neural network is simulated with the outer loop displacement signal but with three different excitation frequencies (0.26, 0.3 and 0.5Hz). The network gives good predictions when it is simulated with the displacement signal having the same excitation frequency of 0.25Hz as used for training. The predictions deteriorate as the excitation frequency of the displacement input moves away from the

excitation frequency used during training. The results in Figure 13(a) are similar to what was observed in Figure 12, i.e. the network has good generalization with respect to interpolation but not with respect to extrapolation. As stated previously, this implies that, in order for the neural network to be able to emulate the behaviour of the leaf spring, it has to be trained with data over its entire working range (i.e. spring displacements and excitation frequencies). It would be more advantageous to have a network with a better ability to generalize as this would require less experimental data for training.

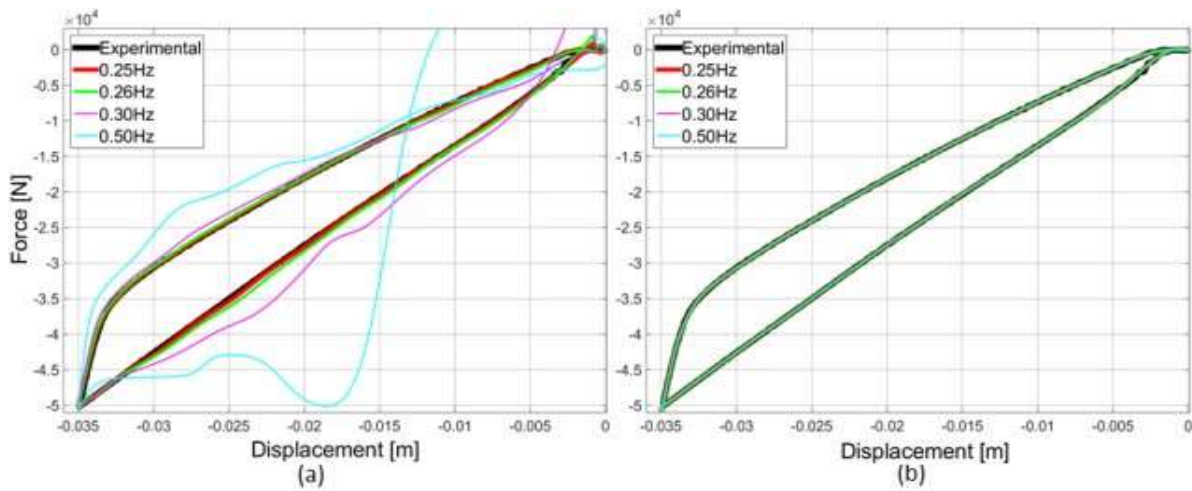


Figure 14. Results of 2-15-1 network using (a) displacement and velocity and (b) displacement and  $\text{sign}(x_i - x_{i-1})$  as inputs, for different excitation frequencies

The dependency of the neural network on the excitation frequency is due to the use of velocity as one of the inputs. Figure 15 shows the displacement and velocity inputs to the neural network for displacement associated with the outer loop (as indicated in Figure 6) at different excitation frequencies. Note that the displacement and velocity time histories have been shifted such that the point where the loading direction on the spring changes from loading to unloading, coincides. The difference between the velocities for the displacement signal having the same amplitude but different frequencies can be seen from

Figure 15. This difference in the velocities causes the network to extrapolate when given different excitation frequencies. It is interesting to note that the neural network gives good force predictions at the point of maximum deflection (see Figure 14(a)). This is the moment where the loading on the leaf spring changes and the velocity is zero. At this point the velocity, for all the different excitation frequencies, is similar and therefore the force prediction of the network is good for all the excitation frequencies.

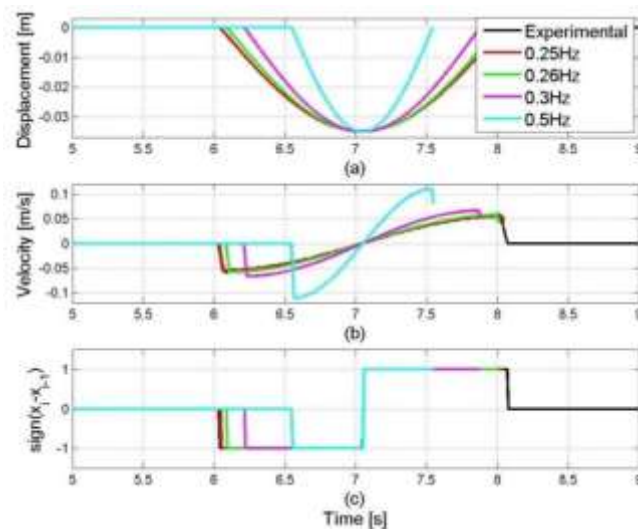


Figure 15. Characteristics of input parameters to neural network for displacements associated with the outer loop at different frequencies

#### 4. Effect of inputs on generalization

The results shown in Figure 14(a) clearly indicate the dependency of the neural network on the excitation frequency of the displacement input, which is contradictory to the behaviour of the physical leaf spring. It was shown in section 2 that the force-displacement characteristic of the leaf spring is not dependent on the excitation frequency of the displacement input to the leaf spring. This contradiction seems to indicate that velocity may

not be a good input to use in order to create a neural network that is able to emulate a multi-leaf spring.

The effect of the input(s) on the generalization ability of the neural network is demonstrated by using the same 2-15-1 neural network as used in the previous paragraphs but with different inputs. The network considered up to now uses the displacement and velocity as inputs. The choice of displacement as input, for a spring, is obvious. Velocity was initially chosen as it indicates to the neural network whether the leaf spring is being loaded or unloaded. The use of velocity causes the network to be dependent on the excitation frequency of the displacement signal. This is in contradiction to the behaviour of the physical leaf spring for which the characteristics have been shown to be velocity independent (see Figure 3).

Velocity may therefore not be the best choice for use as an input. The velocity input is replaced by a signal that will still indicate to the network whether it is being loaded or unloaded but will not make the network dependent on the excitation frequency of the displacement signal. This signal takes the sign of the difference between the displacement at the current time step and the previous time step (i.e.  $\text{sign}(x_i - x_{i-1})$ ). Figure 15(c) shows the characteristics of this input parameter. Comparing this to the characteristic of velocity (in Figure 15(b), it can be seen that the outer loop at 0.25 spans the range of the input parameter and has the same amplitude for different excitation frequencies. This leads to a signal that indicates to the network whether the spring is being loaded or unloaded and is not affected by different excitation frequencies.

The 2-15-1 network using displacement and  $\text{sign}(x_i - x_{i-1})$  as inputs was trained using the outer loop of the experimental force-displacement characteristic. This network was again simulated with displacement signals having different excitation frequencies to that used for training. The results are shown in Figure 14(b). It can be observed that the network gives similar



force predictions for displacement signals having different excitation frequencies. The network is therefore no longer dependent on the excitation frequency of the displacement signal. Figure 13(b) shows the results of this network when trained using the outer loop (Training set 1) but simulated using all the loops. Comparing Figure 13(b) to Figure 13(a), a significant improvement in the ability of the network to generalize, particularly with respect to extrapolating beyond the range of the training set, can be observed.

The substitution of velocity with the sign of the difference in displacements at the current and previous time steps ( $\text{sign}(x_i - x_{i-1})$ ), greatly improved the ability of the neural network to generalize. This improvement was due to the fact that the outer loop spans the range of data and has the same amplitude for different excitation frequencies for the input parameter  $\text{sign}(x_i - x_{i-1})$ . There are some aspects with respect to the force predictions of the network, using displacement and the  $\text{sign}(x_i - x_{i-1})$  as inputs, that have to be further investigated. These aspects include the spikes observed at the points where the direction of loading changes with certain displacement inputs as well as the deviation from the experimental force-displacement characteristic in certain areas of the inner and end loops (see Figure 13(b)). These deviations are most likely due to the input parameter  $\text{sign}(x_i - x_{i-1})$  used and further illustrate the effect of the inputs on the results of the neural network.

## 5. Application to other leaf springs and setups

A 2-15-1 neural network, with displacement and the  $\text{sign}(x_i - x_{i-1})$  as inputs, was derived for the multi-leaf spring setup as shown in Figure 5. The 2-15-1 neural network is used to emulate the force displacement characteristics of the multi-leaf spring in the in-service setup (Figure 1) as well as of a 3-blade parabolic leaf spring in the in-service setup. The blades in the parabolic leaf spring are spaced such that there is only contact between the blades

at their ends. The in-service setup of the parabolic leaf spring is similar to the in-service setup of the multi-leaf spring shown in Figure 1. The 2-15-1 neural network is trained with a force-displacement data set containing the maximum deflections of the spring, in both compression and extension, for the in-service setup of the multi-leaf and parabolic leaf spring, respectively. The force-displacement characteristics for these two springs in the in-service setups are obtained using an experimental setup similar to the one shown in Figure 5. Figure 16 shows the results of the neural network for the in-service setups of the multi-leaf and parabolic leaf spring simulated with a displacement input contacting different amplitudes. A qualitative comparison between the neural network and the experimental data shows good agreement.

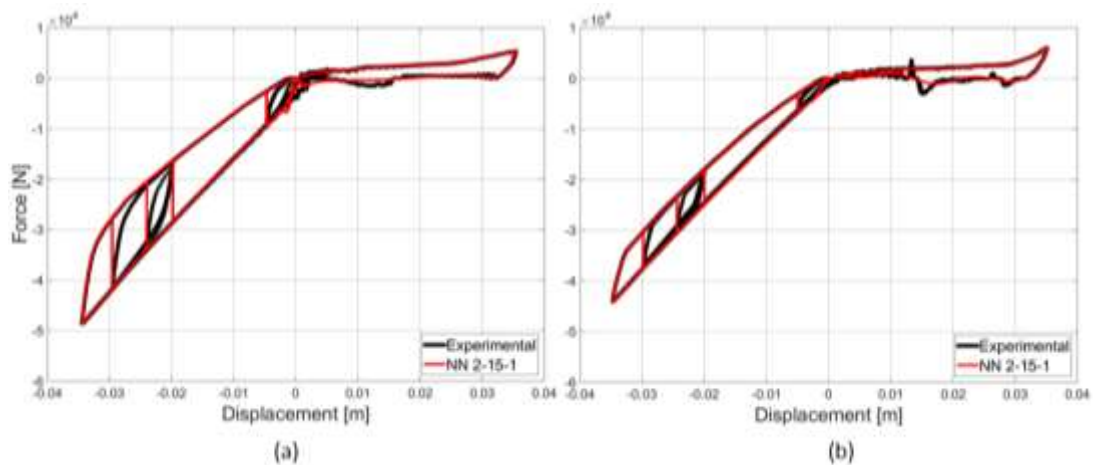


Figure 16. Results of 2-15-1 network using displacement and  $\text{sign}(x_i - x_{i-1})$  as inputs, in emulating the in-service setups of (a) the multi-leaf spring and (b) a parabolic leaf spring.

## 6. Conclusion

The methodology to develop a neural network multi-leaf spring model has been presented. Aspects such as choosing relevant inputs, network architecture, network complexity, training of the network and the generalization ability of the network were investigated. The effect of the inputs on the generalization ability of the neural network was demonstrated using two 2-

15-1 feed forward neural networks, one having displacement and velocity as network inputs and the other displacement and the sign of the difference in displacements at the current and previous time steps ( $\text{sign}(x_i - x_{i-1})$ ). It was shown that the generalization ability of the network can be improved by the selection of appropriate network inputs. The advantage of a neural network with better generalization is that less training data is required. This may not be an advantage when training data is generated from an analytical model as in the study by Ghazi Zadeh et al. (2000). However, when experimental data is to be used for the training data it would be advantageous if a minimal amount of data is sufficient to train the neural network. In the case of the leaf spring it means that only data of the outer loop, or in other words, force-displacement data of the leaf spring's maximum displacement range is required. Furthermore, appropriate inputs will also focus the network toward interpolation instead of extrapolation which, from the results in this study, is a desirable situation. The generalization of the network that has to interpolate can be improved by selecting the least amount of neurons that is still able to capture the response of the spring.

The methodology followed resulted in a simple feed forward network with two inputs (displacement and  $\text{sign}(x_i - x_{i-1})$ ) and one hidden layer with 15 neurons that is able to emulate the vertical force-displacement behaviour of the multi-leaf spring. This neural network requires minimal (experimental) data for training and validation. The neural network created in this study gives a component level model of the leaf spring which can be extended to create more detailed models in order to represent the suspension system. From the results obtained in this study it is postulated that the neural network inputs strongly affect the "intelligence" of the neural network and influence the generalization ability of the network. Well-chosen inputs can improve the generalization of the neural network (especially when extrapolating) and may reduce the required range of the training set.

As stated, neural networks can approximate any non-linear function (Dreyfus, 2005). In designing a neural network to approximate a given function (or emulate the characteristics of a physical component/system) certain aspects need to be considered (i.e. finding the relevant inputs, data necessary for training and testing the neural network, training the network and finding the appropriate complexity of the network). By taking these aspects into account one should be able to obtain a neural network that can approximate any given non-linear function. Considering leaf spring force-displacement characteristics, various leaf spring types (multi-leaf spring, parabolic leaf spring) and configurations have similar characteristics. It is therefore expected that this methodology will be applicable to various leaf springs and configurations. Section 5 showed that the neural network model developed is able to emulate the force-displacement characteristics of the in-service setups of an 8-blade multi-leaf spring and a 3-blade parabolic leaf spring. This is not to say that the methodology, applied to other leaf springs (and their boundary conditions), will result in the same neural network architectures as developed in this study. The methodology presented is not limited to the application of leaf springs only but should apply to various other applications especially those with similar non-linear characteristics.

## **7. Future work**

A thorough validation of the neural network must be done by comparing the predictions of the neural network model to experimental data of the leaf spring obtained from subjecting the leaf spring to white noise excitation as well as more structured road profile data containing a range of amplitudes and frequencies that span the full behaviour of the spring.

## **8. References**

**Cebon, D. (1986)**, „Simulation of the Response of Leaf Springs to Broad Band Random Excitation“, *Vehicle System Dynamics*, Vol. 15, Issue 6, pp. 375-392

- Dreyfus, G. (2005)**, *Neural networks – Methodology and applications*, 1<sup>st</sup> ed., Springer-verlag, Heidelberg, New York
- Ekici, B. (2005)**, „Multi-response optimisation in a three-link leaf-spring model“, *International Journal of Vehicle Design*, Vol. 38, No.4, pp. 326-346
- ElMadany, M.M. (1987)**, „Nonlinear ride analysis of heavy trucks“, *Computers & Structures*, Vol. 25, No. 1, pp. 69-82
- Fancher, P.S., Ervin, R.D., MacAdam, C.C. and Winkler, C.B. (1980)**, „Measurement and Representation of the Mechanical Properties of Truck Leaf Springs“, SAE Technical Paper 800905, SAE International, Warrendale, PA, USA
- Ghazi Zadeh, A., Fahim, A. and El-Gindy, M. (2000)**, „Neuro-leaf spring“, *Heavy Vehicle Systems, A Series of the International Journal of Vehicle Design*, Vol. 7, No. 4, pp. 317-335
- Hagan, M.T., Demuth, H.B. and Beale, M. (1996)**, *Neural network design*, 1<sup>st</sup> ed., PWS publishing company, Boston, MA
- Hoyle, J.B. (2004)**, „Modelling the static and dynamic frequency response characteristics of a leaf spring truck suspension.“, *Proceedings of the Institution Mechanical Engineers*, Vol. 218, Part D: Journal of Automobile Engineering, pp. 259-278
- Huhtala, M., Vesimaki, M. and Halonen, P. (1994)**, „Computer Simulation of Road-Vehicle Dynamic Interaction Forces of Three-and Four-Axle Trucks“, *Vehicle – Road Interaction*, ASTM STP 1225, B.T. Kulakowski, Ed., American Society for Testing and Materials, Philadelphia, pp. 36-51
- Jayakumar, P., Alanoly, J. and Johnson, R. (2005)**, „Three-Link Leaf-Spring Model for Road Loads“, *SAE Technical Paper 2005-01-0625*, SAE International, Warrendale, PA, USA
- Johrendt, J.L., Frise, P.R. and Malik, M.A. (2008)**, „Streamlining automotive product development using neural networks“, *International Journal of Vehicle Design*, Vol. 47, Nos. 1/2/3/4, pp.19-36
- Kat, C-J. and Els, P.S. (2011)**, „Elasto-plastic leaf spring model“, *International Journal of Engineering Systems Modelling and Simulation*, Vol. 3, Nos. 3/4, pp.126-139
- Li, Q. and Li, W. (2004)**, „A Contact finite element algorithm for the multileaf spring of vehicle suspension systems“, *Proceedings of the Institution Mechanical Engineers*, Vol. 218, Part D: Journal of Automobile Engineering, pp. 305-314
- Mathworks (R2016a)**, *MATLAB Neural Network Toolbox User's Guide*, <http://www.mathworks.com/help/nnet/> [Accessed on 2016/07/27]
- Moon, I.D., Kim, G.T, Lim, J.H. and Hwang, Y., (2007)**,“A Flexible Multi-Body Dynamic

Model for Analyzing the Hysteretic Characteristics and the Dynamic Stress of a Taper Leaf Spring", *SAE Technical Paper 2007-01-0852*, SAE International, Warrendale, PA, USA

**Müller, B., Reinhardt, J. and Strickland, M.T. (1995)**, *Neural networks – An Introduction*, 2<sup>nd</sup> ed., Springer-Verlag, Heidelberg, New York.

**Qin, P., Dentel, G. and Mesh, M. (2002)**, „Multi-leaf spring and Hotchkiss suspension CAE simulation“, Paper Presented at the 2002 ABAQUS Users“ Conference. 28-31 May 2002. Newport, Rhode Island, USA.

**Yang, Y., Ren, W., Chen, L., Jiang, M. and Yang, Y. (2007)**, „Study on ride comfort of tractor with tandem suspension based on multi-body system dynamics“, *Applied mathematical modelling*, doi:10.1016/j.apm.2007.10.011