CHALMERS
UNIVERSITY OF TECHNOLOGY

Development of a Clad Stress Predictor for PCI Surveillance using Neural Networks

Master's thesis in Nuclear Science and Technology

Otto Gärdin

Department of Physics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2016
CTH-NT-324
ISSN 1663-4662

Abstract

Westinghouse has recently proposed a new methodology to estimate the risk of PCI failure in a reactor, linking the probability for PCI failures to the cladding hoop stress. However, the current thermo-mechanical performance tool, STAV7, is not intended for on-line surveillance, making it too time-consuming to be used for this application. Instead a new approach is attempted, by using a machine learning technique called Artificial Neural Networks. Here, models for clad stress calculations are trained in order to reproduce as close as possible the results from a large number of STAV7 simulations.

In order to create a functioning model, five different phases during the clad stress evolution have been identified. There are three phases during which the power remains constant and the stress evolves over time: Initial reactor power (prior to any power variations), Relaxation (following power increments) and deconditioning (following power reductions). In addition to these, there are two phases during which the power level changes instantaneously: Power increases and power decreases.

It has been demonstrated that it is possible to use neural networks to reproduce the STAV7 clad stress results with high accuracy for the different phases. The calculations are fast enough to be used in a core monitoring system, although more validation, and potentially training, is needed before the networks can be used for reliable application to real operation cases.

Please note that this is the public version of the master thesis report and that certain information has been omitted. For access to the full version contact Westinghouse Electric Sweden AB.

CONTENTS

Nomenclature

ABBREVIATIONS

ANN	Artificial Neural Network
PCI	Pellet-Cladding Interaction
LHGR	Linear Heat Generation Rate
TMOL	Thermo-Mechanical Operating Limit
Backpropagation	Backwards propagation of errors
CR	Control Rod
ERPO	Extended Reduced Power Operation

DEFINITIONS

Power level

The terms “power” and “LHGR” (Linear Heat Generation Rate) used in this report are interchangeable, referring to the same quantity (the unit being fuel rod power per unit length, kW/m).

Burnup

Burnup is a measure of how much the fuel has been utilized. The unit is energy extracted per mass of uranium, MWd/kgU (megawatt days per kg uranium). The burnup thus increases as the fuel is used up.

Average rod burnup

A fuel rod are split into 25 different nodes at which calculations are performed. All calculations in this study are performed at the node with the highest power, and subsequently the node with the highest burnup. However, the power step simulations are defined by the *average rod burnup*. While this does not affect the accuracy of the results, the burnup shown in the graphs will vary slightly from the burnup stated in the corresponding figure texts, due to the difference between average rod burnup and maximum rod burnup.

Clad Stress

The term “Clad Stress” only refers to stresses in the circumferential direction in the cladding, also known as hoop stress.

Clad Creep

Clad Creep refers to the creep of the cladding as a result of the hoop stress in the cladding. This is a slow process.

Cold Gap

The cold gap describes the gap between fuel pellet and cladding at cold conditions, i.e. without the effect of thermal expansion. This is the gap thickness that would be obtained if the power was suddenly dropped to zero.

1 INTRODUCTION

1.1 Background

Nuclear power plants always strive to improve operating performance while at the same time maintaining a safety margin to avoid fuel failures. In order to optimize performance it is thus important to ensure that the accuracy of the models determining the safety margin is as high as possible, so that operations are not restricted by unnecessarily conservative limitations.

One type of fuel failure is caused by Pellet-Cladding Interaction (PCI). Historically, Westinghouse's guidelines have proven to be efficient in avoiding PCI failures, to the point where they have been practically eliminated. However, it is still necessary to place operating restrictions in order to retain the low failure risk. Recent work has been done to reduce these operating restrictions.

The new proposed PCI methodology uses a statistical relationship linking the risk for PCI failure to the hoop stress in the cladding. By continuously calculating the clad stress in the core monitoring system, the remaining operating margin to a potential PCI failure can be directly linked to the current core operation. However, the current method of calculating the clad stress using thermo-mechanical performance codes is very slow and not possible to use for on-line surveillance.

Therefore, a new approach is attempted. By using machine learning, it may be possible to create a model that can predict the clad stress accurately enough without time-consuming calculations as required in a real-time PCI monitoring tool.

Some basic definitions used in the study are explained in Section 2 and the PCI phenomenon, previous recommendations and the new methodology is summarized in Section 3.

Section 4 describes the basic concept of artificial neural networks, the machine learning tool used in this project. This includes both the basic concept of neural networks, the training algorithms used and finally a description of the structure of a trained neural network.

In Section 5, the simulations used to train the neural network, which has been performed in STAV7, are explained. The training setup is described in Section 6. This mainly includes pre-processing of the STAV7 data before it can be used to train the network, first by filtering out irrelevant or in some cases erroneous calculations, and then transforming the data series into a format suited for training.

Finally the results of the machine learning process are presented in Section 7, where the overall variation between the STAV7 calculations and the neural network calculations are shown, together with some example data simulations to better visualize what the different variations mean in practice. In Section 8 the limitations and potential sources of error which occurs as a result of using machine learning are discussed. Suggestions on how to move forward after this project is finished are also given.

1.2 Objectives

The objective of this study is to evaluate the possibilities of using machine learning to estimate the clad stress that occurs as a result of Pellet-Cladding Interaction. The machine learning tool

selected for this purpose is MATLAB's Neural Networks Toolbox. The network will aim to reproduce the clad stress based on calculations performed by Westinghouse's thermo-mechanical performance tool, STAV7 and to obtain a model able to produce reliable results with good accuracy while still being fast enough to perform calculations as a part of a core simulator and a core monitoring system.

The task is split into several steps:

First, the different phases of the clad stress evolution need to be identified. During different operating conditions, the clad stress will vary in different ways. In order for the network to estimate the clad stress accurately, the different phases are modelled separately.

Once the phases have been found, the variables needed to estimate the clad stress for the different cases need to be identified. This is done by a combination of prior knowledge and testing if adding new parameters will impact the results.

In addition to identifying the input parameters, the STAV7 data needs to be pre-processed before being used in the neural network. This includes smoothing out inconsistencies, even out time steps and removing unreasonable or irrelevant simulations.

The networks are then trained against each of the different phases, using simple STAV7 calculations where the phases are clearly defined. Once the simpler cases are working, the separate networks are then integrated into the same model which chooses, depending on the reactor operation and the clad stress phase, which network is to be used to estimate the clad stress at any given point. The model may be tested, and if needed also trained, against more complex scenarios more reminiscent of real operating conditions.

Eventually, the plan is to integrate the neutral network model into Westinghouse's core simulator POLCA7.

2 PELLET-CLADDNING INTERACTION

2.1 PCI Phenomenology

Fuel rods are manufactured with a gap between the fuel pellet and the cladding in a fuel rod (see Figure 2.1). As long as the gap remains open, no clad stress is generated as a result of PCI. However, during operation the gap is gradually closed until the pellet eventually comes into contact with the cladding and stresses start to build.

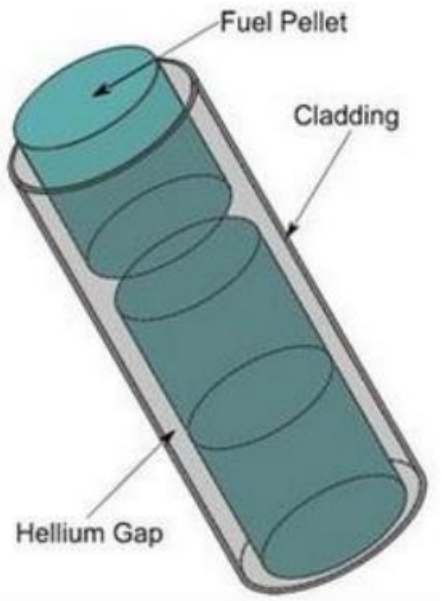


Figure 2.1- Layout of a fuel rod [1].

There are four main factors driving the gap closure:

1) The power generated in the fuel pellet increases the temperature which in turn causes a *thermal expansion* of the pellet. The thermal expansion is basically instantaneous, as soon as the power generated in the reactor increases, the pellets start to expand.

2) As a result of fission products being generated as the pellet is being utilized (burned), a *pellet swelling* will occur. Early on, up to a burnup of about 10 MWd/kgU, the swelling is counteracted by pellet densification and will have no impact on the gap. For the rest of the fuel lifetime the swelling increases linearly as a function of burnup. This process is much slower than the thermal expansion.

3) Stresses in the cladding will result in a *clad creep*. Prior to the gap closure, the surrounding water pressure will cause a negative (inward) stress in the cladding, which in turn causes the cladding to creep inward. Once the gap closes, the thermal expansion and swelling of the pellet will cause stresses to build within the cladding. Once the stresses caused by PCI becomes large enough to overcome the negative stress caused by coolant pressure, the cladding begins to creep outward. The creep is a function of the clad stress and is a slow process at regular stress levels, but fast at large stresses.

4) The pellets will begin to crack as a result of thermal stresses within the pellet. This phenomenon is called *pellet relocation* and will increase the radius of the pellet, thus

reducing the gap. Prior to gap closure, the pellet relocation is a function of both the power level and the burnup of the pellet. Once the gap is closed, the stresses will cause the pellet to be pressed inwards, reducing the size of the relocations, making it a function of power, burnup and contact pressure [2]. The pellet relocations occur on the hour scale, making it slower than the thermal expansion but faster than the pellet swelling (and creep at normal stress levels).

At low reactor power, the thermal expansion is very small. However, the pellet swelling and the clad creep are large enough to close the gap even without a significant thermal expansion. Once the burnup reaches ~50 MWd/kgU, the gap is closed, regardless of thermal expansion.

2.2 Clad Stress

There are three different failure mechanisms due to PCI: stress corrosion cracking, hydriding embrittlement strain failure, and a delayed hydriding cracking started from the outer part of the cladding. Ramp tests show that stress corrosion cracking is the dominating failure mechanism, as it might be initiated earlier than the other failure modes [3]. Thus, the primary objective is to protect the fuel against this failure mechanism.

Stress corrosion cracking (SCC) is a process where cracks occur due to a combination of high local stresses and an aggressive environment. In fuel rods, SCC occurs from the inside, where aggressive fission products such as iodine and cadmium are deposited on the inner surface of the cladding. Once the gap is closed and stresses grow, there is a risk that the combination will cause the cladding to rupture, leading to a fuel failure.

In order to understand how to avoid PCI failures, it is therefore important to understand how the clad hoop stress is affected by varying reactor operations. The reactor operations can be described by three different scenarios: power remaining constant, increasing the power and decreasing the power. Each of these cases has a different impact on the clad stress and therefore all need to be tracked accurately.

2.2.1 Constant Power Operation

At constant power, the clad stress is initially negative due to the gap being open. The surrounding coolant pressure is therefore the only contribution to clad stress and will impose an inward stress. After the gap closes, outward stresses will start to build, increasing continuously with increasing burnup. The increasing clad stress will also increase the outward creep of the cladding, which will then in turn limit the stress build-up. Eventually the clad stress and clad creep reaches equilibrium. This can be seen in Figure 2.2, where the clad stress increases asymptotically, which causes the strain to also become constant. This in turn translates to a more linear clad creep, as can be seen in the lower figure.

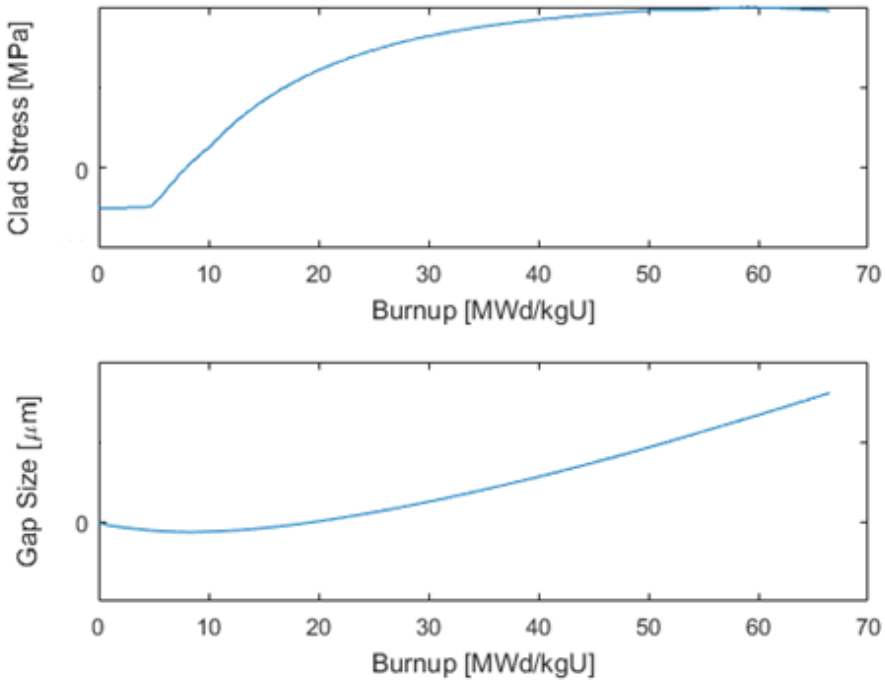


Figure 2.2- STAV7 calculations of clad stress and clad creep at constant power 50kW/m. As can be seen, the creep becomes almost linear as the clad stress apporaches its asymptotic level.

Furthermore, it can also be seen in Figure 2.3 that the equilibrium stress level is not the same for all power levels. At lower power, the clad stress required to reach an asymptotic level is lower than for higher power levels.

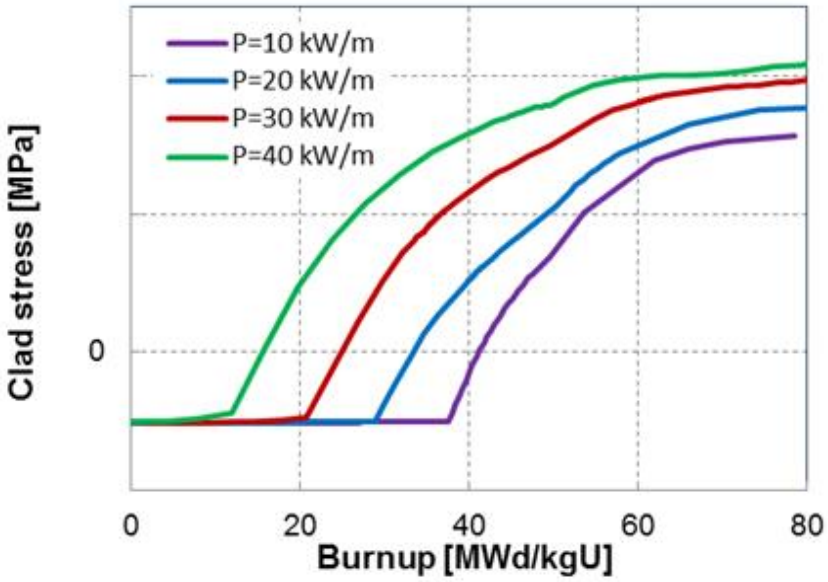


Figure 2.3- Clad stress as a function of burnup for several constant power levels (10-40 kW/m) [3].

2.2.2 Power Increments

While constant power operations reach an asymptotic stress level, increasing the power in the core may cause the stress to reach much higher levels, see e.g. Figure 2.4.

Any power variation in the core is assumed to occur in a step-wise fashion, with the power changing to a new level instantaneously. For that reason, time and/or burnup dependent quantities, such as the swelling of the pellet and the clad creep, will remain unchanged over a power variation. However, as explained in Section 2.1, the thermal expansion of the pellet is basically instantaneous. Increasing the power thus leads to a rapid expansion of the pellet without a corresponding creep, which consequently causes a peak in the clad stress.

A significantly large stress increment will however also increase the clad creep drastically, causing a *relaxation* process. If the clad stress is high, the clad creep will reduce the clad stress back down towards the equilibrium level. By allowing for enough relaxation to occur after a power increment, the power can be increased again, resulting in a much lower clad stress than if the larger step had been taken directly after the previous power increment. For this reason, the relaxation process is also known as *conditioning*, as the new condition allows for new power operations.

Figure 2.4 shows how the condition prior to the ramp can affect the resulting clad stress. The same ramp is performed at six different average rod burnups, ranging from 25 MWd/kgU up to 50 MWd/kgU. As can be seen, the earlier ramps do not cause a stress peak, but as the burnup increases, the resulting clad stress becomes larger and larger. However, the equilibrium level at high burnup is roughly the same for all ramps, regardless of the maximum stress reached.

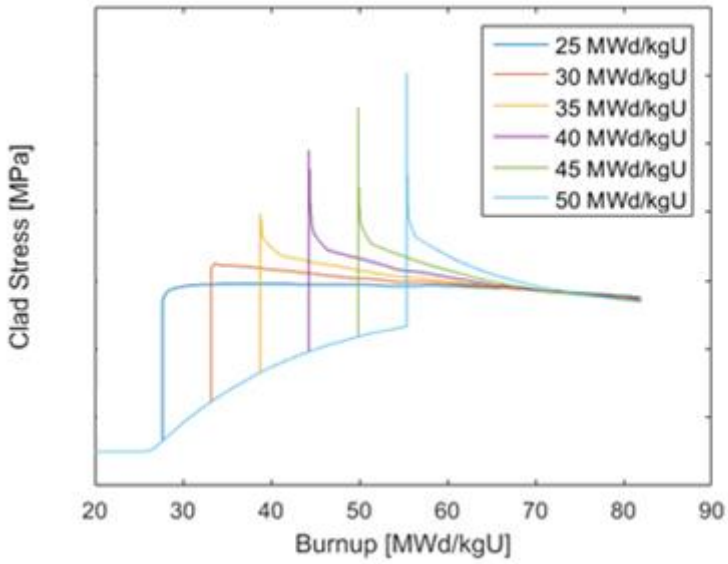


Figure 2.4- STAV7 calculations of clad stress evolution as a function of burnup for the same power ramp (25 kW/m to 40 kW/m) performed at six different average rod burnups

During power increments, the clad stress can be split into three different regimes: soft contact, hard contact and plastic deformation.

In the soft contact region, pellet relocations (cracks) are still present. The cracked pellet is pressed inwards, somewhat counteracting the stress build-up. In this region, the stress increases almost linearly. At a certain stress level, the relocations have been pressed as far inwards as is possible, and the contact transitions to the hard region. The stress build-up is still almost linear, but about four times faster in the hard region. It is the transition into the hard contact region that causes the clad stress to peak drastically between the highest three ramps in Figure 2.4. The last region is the plastic deformation region. As the name suggests, the yield

point of the cladding have been reached and the cladding is subject to plastic deformation. However, this is far above the allowed operating limit of a nuclear power plant.

Figure 2.5 shows the different regions for three different ramp tests. As can be seen, the transition occurs after a smaller power step at larger initial power level.

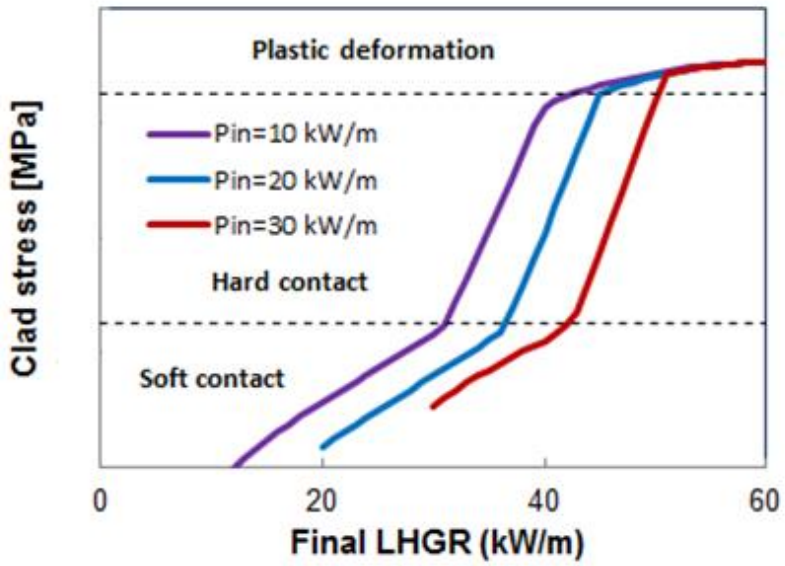


Figure 2.5- Peak Stress as a function of power steps for different initial power levels at average rod burnup 50 MWd/kgU [3].

The long term evolution of the clad stress can be seen in Figure 2.6, which shows four different power increments at the same burnup. As can be seen, if the power step is small in relation to the initial conditions, the clad stress will never exceed the equilibrium level. The resulting creep is not large enough to cause relaxation, and the clad stress will simply continue towards an asymptotic level similar to the one reached during constant power operations. However, the larger power ramps far exceeds the one that would be reached at constant power operation, with the largest power ramp (from 25 kW/m to 55 kW/m) creating a large stress peak. As can be seen, the equilibrium stress level after relaxation is not the same for all ramps, as the final power level varies.

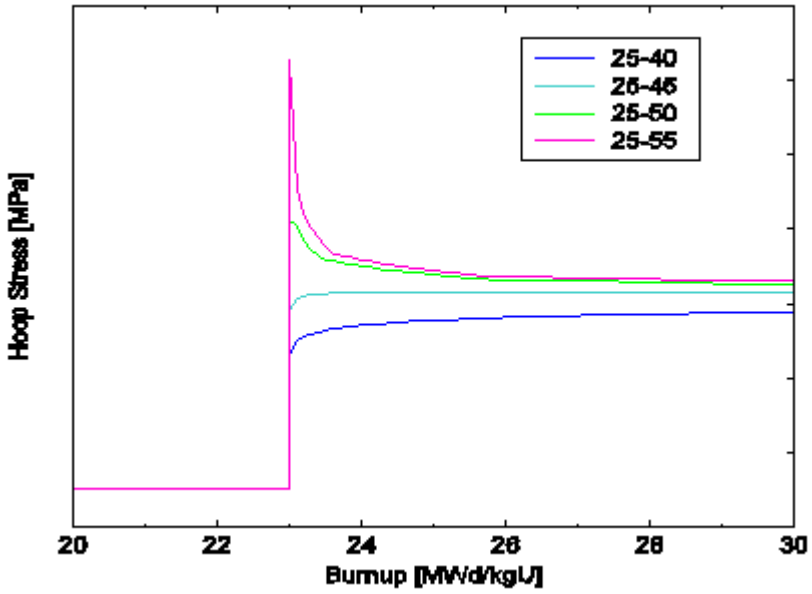


Figure 2.6- STAV7 calculations of clad stress evolution as a function of burnup after several different power steps (40-55 kW/m) starting at 25 kW/m [3].

2.2.3 Power Reduction

Reducing the power will instead decrease the thermal expansion in the fuel pellet, causing the clad stress to drop. However, if the power would then be brought back up to the previous level, a peak in the clad stress may occur.

This process is caused mainly by a reduction in the clad creep rate. As explained in Section 2.2.2, the clad creep depends on the clad stress: if the stress is reduced, the creep growth will slow down. This behavior can be seen in Figure 2.7, where the difference in clad creep between a simulation at constant power and a simulation containing a single power drop is clearly visible. The clad stress and creep are the same in both simulations until the power in one of the simulations is reduced from 25 kW/m down to 15 kW/m, causing the clad stress to drop. After that point, the clad creep evolves at a significantly lower rate than for the constant power simulation.

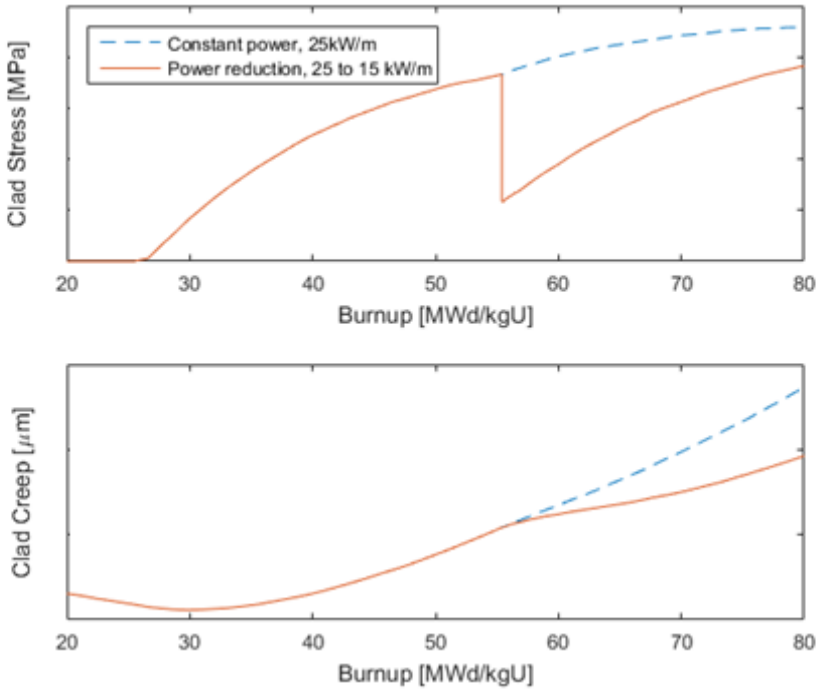


Figure 2.7- STAV7 calculations of clad stress and creep as a function of burnup at constant power (25 kW/m) compared to clad creep and stress during a power reduction: 25 kW/m down to 15 kW/m at average rod burnup 50 kW/m.

This process is called deconditioning, as the cladding loses its previous conditioned state due to the power reduction. As can be seen, the deconditioning grows larger over time, which in turn will cause the clad stress peak to grow more if returning the reactor to its previous power level. However, the deconditioning is generally a slow process. Figure 2.8 shows a simulation of Extended Reduced Power Operation (ERPO) when the power is lowered from 25 kW/m to 15 kW/m and then eventually returned back up to 25 kW/m at different times. As can be seen, the reduced power would need to be retained for several thousand hours in order for a noticeable increase in the clad stress to occur.

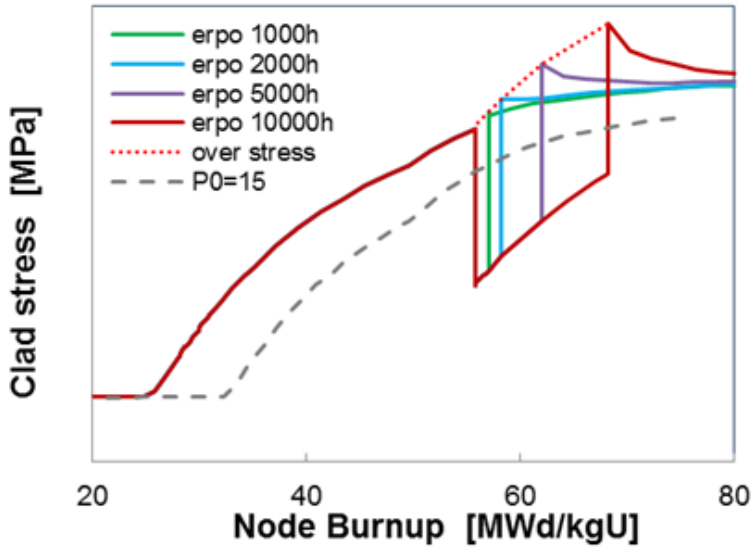


Figure 2.8- Clad stress evolution as a function of burnup for a power reduction without the gap re-opening (25 kW/m to 15 kW/m) and its subsequent return to original power at different times. P0 shows the clad stress evolution at constant power 15 kW/m [3].

If the power reduction is sufficiently large, the gap between pellet and cladding will re-open. Once the gap has opened, the clad stress is once again only affected by the pressure difference inside and outside the fuel rod. Furthermore, unlike when the gap remains closed, the clad stress will not start to build directly after the power reduction. Instead, the stress will remain constant until the gap is closed again, leading to a larger deconditioning. Another ERPO simulation can be seen in Figure 2.9 where the same power reduction takes place at a lower burnup, leading to a re-opening of the gap. As can be seen, returning to original power shortly after a reduction causing a gap re-opening still leads to a noticeable over-stress.

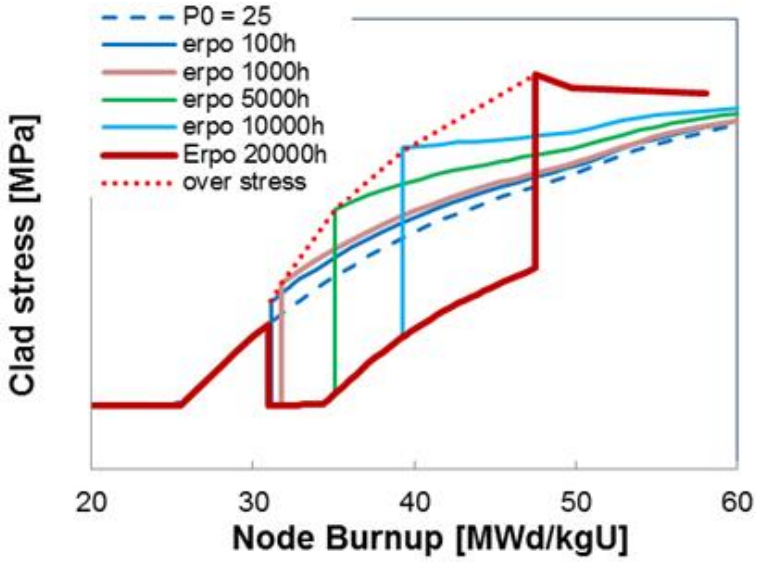


Figure 2.9- Clad stress evolution as a function of burnup for a power reduction resulting in a re-opening of the gap 25 kW/m to 15 kW/m) and its subsequent return to original power after different periods of time [3].

2.3 Current PCI Recommendations

The current PCI recommendations are based on the effects presented above, linking the risk of PCI damage to the power in the core. The recommendations are split into two sets of rules: local rules which restrict the power of the reactor (described in terms of Linear Heat Generation Rate, LHGR. The unit is power generated per unit length, kW/m.) at node level and global rules used for core design and core operation to not exceed the local rules. These rules have been acquired from a combination of ramp tests similar to the ones showed above, experience from commercial reactor operation, engineering judgment and limited fuel rod performance evaluations.

The local rules consist of three main components:

- A burnup dependent threshold level on LHGR. Below this level there are no restrictions on operating at all.
- A restriction on the power ramp rate above the LHGR threshold, limiting the maximum LHGR step to 1.5 kW/m every six hours, or to a continuous LHGR increase of 0.25 kW/m every hour.
- A deconditioning rate as a function of burnup to account for the creep down of the cladding as well as the swelling/relocation of the pellet (set to 5 kW/MWd/kgU) in the case of a power reduction.

Global rules are used as a complement for further insurance at high power operating conditions. These apply restrictions on the entire core in such a way that the local rules are followed without requiring monitoring every node explicitly. These rules include recommendations such as:

- Reducing the core power ramp rate during start-up above 90% core power (max 1% per hour)
- High burnup fuel (>40 MWd/kgU) is not to be used in power or reactivity regulating Control Rod (CR) cells for annual cycles
- Reducing the power to 80% during CR swaps (GE reactors)
- Limiting the control rod withdrawal rate to max 0.5% CR withdrawal per 4 hours
- The same fuel assembly must not be placed in CR position in two consecutive cycles
- Operation limitations in case of a fuel failure.

2.4 Westinghouse's New PCI Methodology

As the clad stress is assumed to be the driving force behind PCI failures, the new proposed methodology for PCI surveillance links the risk of fuel failure to the clad stress instead of LHGR level and steps. By continuously calculating the clad stress during operation, the reactor operator can receive on-line updates on which power variations are allowed given the current circumstances.

Estimating the PCI failure risk directly from the clad stress has several advantages. As covered in Section 2.2, the clad stress is strongly affected by several parameters, such as final LHGR level, the size of the LHGR step and the conditions prior to a power variation. For this reason, covering these aspects separate from each other leads to less accurate results. By taking all effects into account at once, this problem is solved.

In order to calculate the PCI failure probability, 68 ramp tests were performed for different types of fuel (10x10 and 8x8 fuel as well as liner/no liner fuel). Figure 2.10 shows the maximum clad stress reached for the tests against the burnup. The red points signify fuel failures, while blue points signify intact fuel rods. As can be seen, no clear correlation can be found between failure and burnup, which is expected based on the assumption that the clad stress is the main parameter affecting PCI failure [3].

It can also be seen that there is a clear overlap between failure and no failure with respect to clad stress. This is also expected, as a result of the stochastic nature of stress-corrosion cracking.

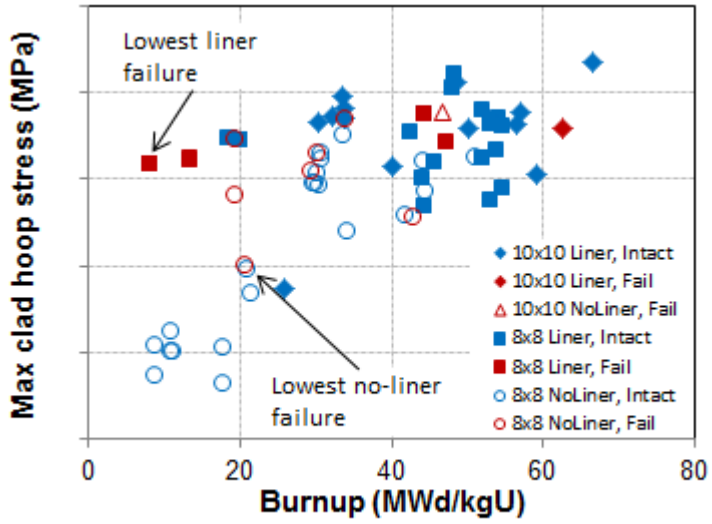


Figure 2.10- Maximum clad stress as a function of burnup for 68 ramp tests, sorted by liner and non-liner fuel, failure and no failure as well as 10x10 and 8x8 fuel [3].

In order to find the PCI failure probability, a statistical evaluation was performed by fitting a cumulative Weibull distribution function to the data.

From the fitting curve, the probability of failure for a given pin during its life time is given by [4]:

$$p_{life}(\sigma) = 1 - \exp\left(-\left(\frac{\sigma}{\lambda_{weib}}\right)^{k_{weib}}\right) \qquad (1)$$

Where σ is the clad stress level, k_{weib} is the shape factor (~12) and λ_{weib} is a scale parameter. The PCI failure probability for a fuel assembly is then:

$$1 - p_{asy} = \prod_{all\ pins} (1 - p_{life}) \quad (2)$$

And subsequently, for a full core:

$$1 - p_{core} = \prod_{assemblies} (1 - p_{asy}) \quad (3)$$

Finally, the failure probability is translated into a generic clad stress limit, allowing for one failure rate per 100 years of operation (1% risk of failure during one full-power year).

2.4.1 Estimating the Clad Stress

The problem associated with estimating the risk for PCI failure using clad stress instead of LHGR is that while the power of the core is being monitored, the clad stress is not. For that reason, a new module needs to be implemented in the core monitoring system that calculates the clad stress.

The clad stress is currently calculated using STAV7, Westinghouse's thermo-mechanical performance tool. However, STAV7 is very calculation heavy, making it too slow to use for on-line surveillance.

Previously, an attempt was made at estimating the clad stress by fitting mathematical expressions to the curve. However, finding the transition point between soft and hard contact has proven to be hard, resulting in an accumulating error when the model is used for multiple subsequent ramps.

An alternative to fitting equations to the data is to use machine learning to train a model to estimate the clad stress. Machine learning is a type of computational optimization which attempts to predict the outputs of a system from a given number of inputs by "training" against sets of known input-output pairs. In this case, a large number of simulations have been created using STAV7, covering the known effects which cause variations in the clad stress. These are then used as the basis for the machine learning process.

One major advantage of using machine learning is that once the training is complete the transfer function from input to output is performed by a simple matrix calculation, which is very fast. A working model would thus solve the time constraint preventing the use of STAV7 in a PCI monitoring tool.

2.4.1.1 The Cold Gap

As seen in both Section 2.2.2 and 2.2.3, the clad stress will be affected differently by the same power variation given different initial conditions. The effect of conditioning and deconditioning implies that not only the current reactor operation, but also the history of previous operations is necessary in order to calculate the clad stress.

The pin history is included in the model by means of a parameter called the cold gap. The cold gap describes, as the name suggests, the gap between pellet and cladding at cold conditions (zero power). Slow effects, such as the pellet swelling, clad creep and relocations are taken into account, while the contribution from thermal expansion is zero. This way, the lingering

effects of previous operations, such as relaxation, deconditioning and relocations, are taken into account when estimating the clad stress. The cold gap is defined as:

$$\Delta_{cold} = \Delta_{creep} + \Delta_{swell} + \Delta_{relocation}$$

3 NEURAL NETWORKS

This study is focused on one subgroup of machine learning, Neural Networks, through use of MATLAB's Neural Networks Toolbox.

Artificial Neural Networks (ANN) are based on the concept of biological neural networks. Just like in its biological counterpart, the ANN consists of a network of connected nodes, or neurons, sorted into layers. Figure 3.1 describes a two-layer neural network. In some literature the inputs sending a signal through the first weights are referred to as a layer as well. However, as there is no transfer function or weights transforming the signal prior to the inputs, they are not considered as a layer in MATLAB [5], and subsequently not in this report either.

As can be seen, each neuron in one layer is connected to every neuron in the following layer through weighted synapses. When an input is sent to the network, each signal is weighted, with weight $w_{n,m}^{1}$ referring to a weight in the first weight matrix connecting to the n^{th} neuron in the first layer from the m^{th} input signal. The weighted signals are summed up together with a bias in the neuron. The sum forms the input to a transfer function f, which generates the output signal of the neuron. The output signal is then once more weighted in the second weight matrix and sent to the next layer. The same procedure is repeated until all layers have been passed through and the system output is found.

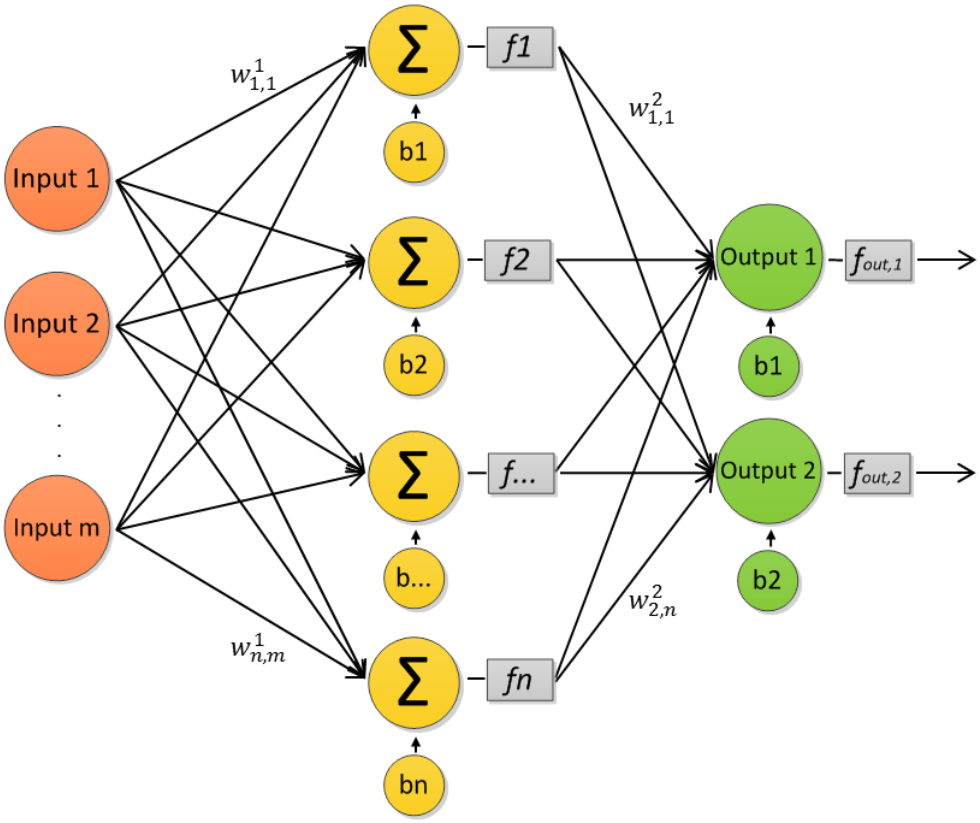
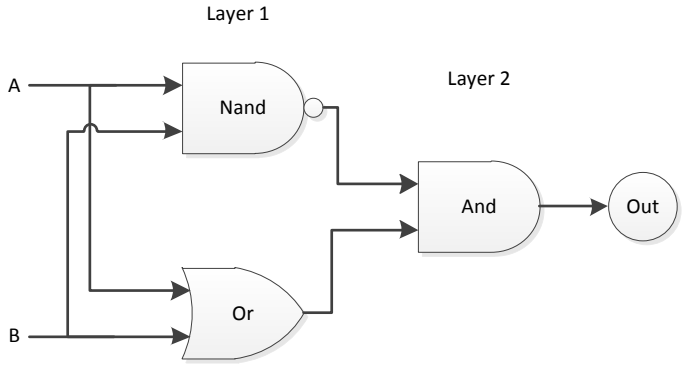


Figure 3.1 - Example of a neural network consisting of an input signal and two layers: one "hidden" layer and one output layer.

Only the inputs to the first layer and the outputs from the final (output) layer are linked to physical quantities. The signals from the layers in-between acts as transfer functions, transforming the input signal into something the final layer can use to compute the expected system response. These middle layers are known as “hidden layers”. A network containing one or more hidden layers is often referred to as a *multi-layer neural network.*

A simple example can be shown by attempting to estimate the logical operator function “XOR” using “OR”, “AND” and “NAND” operators. XOR-functions work by responding “TRUE” if given two different binary inputs and “FALSE” if they are the same. In order to do this, two layers of functions are needed to connect the input to the output (Figure 3.2 below): the first layer contains an OR and a NAND operator in parallel, checking if at least one element (OR) but not both (NAND) are true. By itself, the response from layer 1 does not say anything about the XOR-function, and can thus be seen as a hidden layer. In the second layer, an AND operator is used, transforming the outputs of the first layer into the output of the system, a logical XOR response. It thus corresponds to an output layer of a neural network.



A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

Figure 3.2- A simple two-layer system, logical operator

In the above example, the transfer functions work as simple logical operators. In most applications the output is more complex. In these cases the logic of how the hidden layer works remains the same, but a more complex transfer function is used. Using the correct transfer function, most problems can be solved using only a single hidden layer. According to "Introduction to Neural Networks in Java", by Jeff Heaton:

"A network using a single hidden layer can approximate any function that contains a continuous mapping from one finite space to another" [6].

As this is the case for clad stress estimation, a two-layer network is used like the one described in Figure 3.1. The inputs are connected to one hidden layer, which in turn is connected to an output layer that provides the output of the system.

Figure 3.3 shows the structure of a feedforward neural network in MATLAB, in this case a network with 5 inputs, 2 outputs and 20 neurons in the hidden layer. A feedforward neural network is a network that, given a number of inputs, estimates the output of a system with no feedback or loop mechanism (thus the name feedforward: information only goes in one direction). The transfer function in the neurons of the hidden layer is a hyperbolic tangent sigmoid function, which varies the output signal between -1 and 1 as the input goes from negative to positive infinity:

$$a = \frac{2}{(1 + e^{-2n})} - 1 \qquad (4)$$

Where a is the output of the transfer function (i.e. input to the output layer weight matrix) and n is the sum of weights and biases in the neuron.

The output layer then transforms the responses from the hidden layer into the expected output through a linear scaling function. The transfer functions used are displayed in the layer boxes in Figure 3.3.

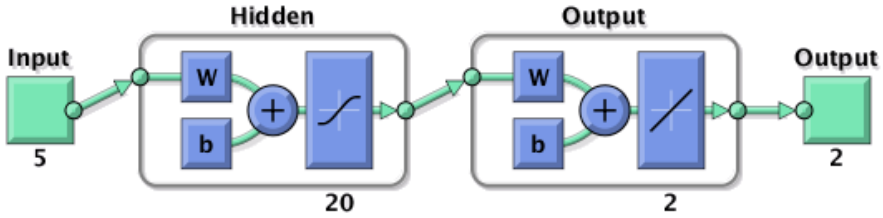


Figure 3.3- Structure of a feedforward neural network in MATLAB, with one hidden layer containing 20 neurons

3.1 Number of Neurons and Layers

The input signals as well as the neurons in the output layer are defined by the number of inputs and outputs connected to the network. There is no set number of neurons in the hidden layer though, and is one of the optimization parameters of the network. Naturally, by adding more neurons to the system, the accuracy of the network is increased. However, adding too many neurons in the hidden layer risks overdefining the network to the training set.

The risk of overdefining the network is in this case solved by using Bayesian regularization (see Section 3.2.4). Using too many neurons is still not always recommended, as a larger network takes considerably longer to train. Using as few neurons as possible without limiting the accuracy is therefore preferred.

As already mentioned, a network using a single layer is capable of approximating any function (given the correct input parameters). While that is the case, there is sometimes a difference in the *representability* and the *learnability* of the network [7]. Representability means that the network can, in principle, find a solution, whereas learnability means that the network is also able to learn to approximate it. In some cases, although a solution does exist, the network process does not manage to identify it. In these cases, a possible approach is to add more hidden layers. The downside of adding more hidden layers is that they also generate additional local error minimas which could lower the accuracy of the model. For this reason, it is recommended to initially estimate a problem with a single hidden layer. However, if that fails it is possible to add a second layer and retry [7].

3.2 Training the Network

As previously mentioned, the concept of using neural networks is to train a system to approximate an output. The network is trained by being given a “test set” of data, containing the inputs to the model as well as the desired targets. The training then consists of updating the weights and biases in order to minimize the error, i.e. the difference between the network response and the desired targets. Training continues until the error reaches a minimum and further training no longer improves the performance of the network.

The performance is defined as the mean squared error:

$$performance = \frac{1}{n}\sum_{i=1}^{n}(t_i - y_i)^2 \qquad (5)$$

Where n is the number of input-output pairs, t are the desired targets and y is the network output.

While the goal of the training is to improve the performance as much as possible, it is equally important to avoid over-fitting. There are two checks in place to prevent over-fitting. The first one is to introduce a regularization term when the training. This is explained in further detail in Section 3.2.4. The second check comes from splitting the data into two groups: training data and test data. As the name implies, the training data is used to train the network. The test data is set aside during training, so that they are independent of the solution. Once the network is trained, it is used against the test set. If the accuracy is poor, the network has overdefined the solution. If the accuracy is good however, it implies that the training is a success and the network has been able to capture the desired behavior. In this project, 20% of the data samples are set aside for testing, while the remaining 80% is used for training.

While the training process itself is very time-consuming, once the network is complete, using it to estimate the desired outputs is very fast, as the finished network works by simple matrix operations.

3.2.1 Supervised Learning

The training process in this project is called supervised learning. Supervised learning is a learning process in which the expected network response (i.e. the target) is known. During the training the expected network response is known from STAV7's output data, and the network will update its weights and biases until the error between STAV7 and the neural network has been minimized.

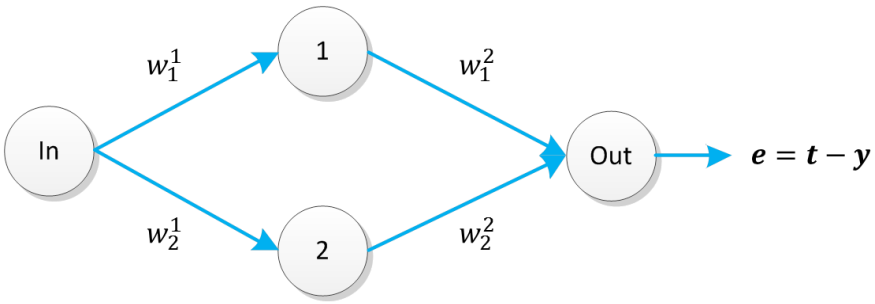
3.2.2 Backwards Propagation of Errors

A major issue in multi-layer neural networks is that it is only possible to directly measure the error signal between the output of the final layer and the target values. The hidden layers have no targets to be compared to, and finding their error signal is therefore more complicated.

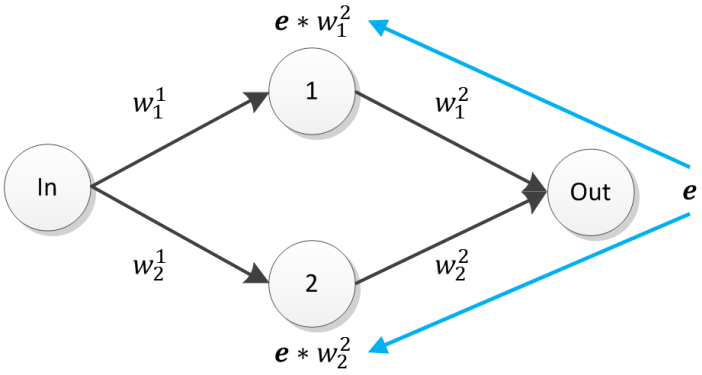
It is however possible to estimate the error of the hidden layers by using backwards propagation of errors, backpropagation for short. Backpropagation links the error of the outputs to the hidden layers by allowing the output error to propagate through the system.

The process follows three steps (explained here with a single input, single output neural network with two neurons in the hidden layer) [8]:

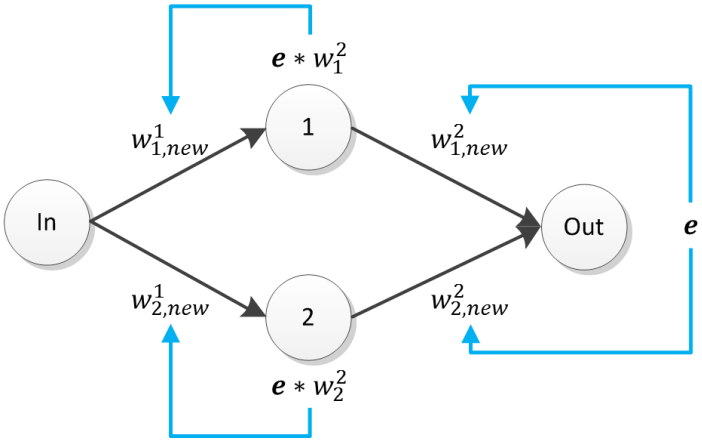
1) A forward pass through the system, using the current weights. The output is compared to the expected response and the error is calculated:



2) A backward pass, sending the error signal through the system using the same weights to find the error in each neuron:



3) Once the error in each neuron is found, the weights are adjusted according to the training algorithm used (the training algorithm used in this project is explained in Sections 3.2.3 and 3.2.4 below):



The process is repeated until the error function has reached a minimum or the training is otherwise cancelled.

A pass through the backpropagation algorithm is called an epoch. By checking the number of epochs after training a network, the number of times the weights and biases have been updated are found.

3.2.3 Levenberg – Marquardt Backpropagation

The backpropagation algorithm used in this project is the Levenberg-Marquardt algorithm. It interpolates between the method of gradient descent and Gauss- Newton's method by means of a damping parameter. Once the errors are found in each neuron, the adjustments are done according to:

$$\Delta \boldsymbol{x} = [\boldsymbol{J}^T(\boldsymbol{x})\boldsymbol{J}(\boldsymbol{x}) + \mu \boldsymbol{I}]^{-1}\boldsymbol{J}^T(\boldsymbol{x})\boldsymbol{e}(\boldsymbol{x}) \quad (6)$$

$$\mathbf{x} = \mathbf{x} + \Delta \boldsymbol{x} \quad (7)$$

Where $\boldsymbol{x}$ is a vector containing all weights and biases, $\boldsymbol{e}(\boldsymbol{x})$ is the error vector, $J(\boldsymbol{x})$ is the Jacobian matrix, μ is the damping parameter and $\Delta\boldsymbol{x}$ is the adjustment made to $\boldsymbol{x}$**.** For a more thorough derivation, see [9].

As μ grows, the algorithm moves closer and closer toward gradient descent, while a smaller μ instead moves the algorithm towards Gauss- Newton. If the performance worsens (the error increases) after updating the weights, μ is multiplied by a factor $\beta > 1$. The process is repeated until the performance is increased, at which point the damping parameter is then instead divided by β:

$$\mu = \mu * \beta \; until \; \boldsymbol{e}(\boldsymbol{x})_{new} < \boldsymbol{e}(\boldsymbol{x})_{old} \tag{8}$$

$$then:$$

$$\mu = \frac{\mu}{\beta} \tag{9}$$

This way, gradient descent is used when far away from the optimum when Gauss-Newton has poor convergence, and Gauss-Newton is used closer to the optimum when gradient descent is slow.

μ is initially set to 0.05 in MATLAB. β is split into two separate parameters, to be able to tune the rate of increasing and decreasing μ separately, with defaults 10 and 0.1 (MATLAB defines the decrease factor as a multiplication term, meaning that they are initially set to be equal).

In order to prevent the algorithm from getting stuck by continuously increasing β if the performance is not improved by updating the weights, the training will stop if μ becomes too large. Generally, this happens close to the minimum when the step size is too large to further improve the network. For this reason, the previous solution is chosen as the finished network if the training breaks due to too high μ.

However, if the limit on μ is set too low, the training may break before convergence is reached due to the weights being far away from their optimum. μ_{max} is initially set to 10^{10}, but through experience it has been found that in some cases this limit is too low, preventing the network from finding a good solution. In some of these cases, μ reached values of about 10^{22} before the solution started to converge. Should the network cancel because of reaching μ_{max} while performance is still poor, increasing μ_{max} could therefore potentially improve the network.

3.2.4 Bayesian Regularization

As mentioned previously, the risk of optimizing a network is that the solution may become overdefined. In order to avoid over-fitting, Bayesian regularization is used together with the Levenberg-Marquardt algorithm.

As explained in the section above, normal Levenberg-Marquardt aims to improve the performance of the system. This is done by minimizing the sum of square errors:

$$E_D = \sum_{i=1}^{n} (t_i - y_i)^2 \tag{10}$$

Where t_i represents the target outputs and y_i represents the neural network response.

Bayesian regularization works by having the network not only minimize the performance function, but also a regularization term. This regularization term is defined as the sum of squares of the network weights:

$$E_W = \sum_{j=1}^{N} w_j^2 \tag{11}$$

Where w_j is the weight coefficient at synapse j. By keeping the weights small, the network response becomes smoother, regularizing the network. Combining these two expressions gives the modified performance function [10]:

$$F = \beta E_D + \alpha E_W \tag{12}$$

Where α and β are performance function parameters. If $\alpha \ll \beta$, the performance function will reduce the size of the errors. If $\alpha \gg \beta$, the performance function will instead reduce the weight size of the network, producing a smoother network response [10].

Finding α and β is done through application of Bayes' rule to neural network training. Assuming Gaussian noise in the input data and a Gaussian weight distribution, the probability densities for the data set and weights respectively is given by:

$$P(D|\boldsymbol{w}, \beta, M) = \frac{1}{Z_D(\beta)} \exp(-\beta E_D) \tag{13}$$

$$P(\boldsymbol{w}|\alpha, M) = \frac{1}{Z_W(\alpha)} \exp(-\beta E_W) \tag{14}$$

Where D represents the data set, M is the neural network model, $\boldsymbol{w}$ is the vector of weights, $Z_D(\beta) = (\pi/\beta)^{n/2}$ and $Z_W = (\pi/\alpha)^{N/2}$ with n being the number of input-output pairs and N being the total number of parameters (weights and biases)[10]. After some derivation, one finds α and β at the minimum point (MP):

$$\alpha^{MP} = \frac{\gamma}{2E_W(\boldsymbol{w}^{MP})} \tag{15}$$

$$\beta^{MP} = \frac{n - \gamma}{2E_D(\boldsymbol{w}^{MP})} \tag{16}$$

Where $\gamma = N - 2\alpha^{MP} tr(\boldsymbol{H}^{MP})^{-1}$ is the effective number of parameters. γ thus describes how many of the weights and biases that are effectively used when estimating the outputs of the network. **H** is the Hessian matrix of the performance function, which using the Gauss-Newton approximation can be approximated using Jacobian matrices [10]:

$$\boldsymbol{H} = 2\beta \boldsymbol{J}^T + 2\alpha \boldsymbol{I}_N \quad (17)$$

The Jacobian matrix is already known from the Levenberg-Marquardt algorithm and can therefore easily be used here.

The full procedure of updating the weights using Levenberg-Marquardt with Bayesian regularization then follows the steps [10]:

0) Define $\alpha = 0$ and $\beta = 1$ when initializing the training, making the first step only depend on the target-output error.

1) Take one step of the Levenberg-Marquardt algorithm to minimize the performance according to: $F = \beta E_D + \alpha E_W$.

2) Calculate the effective number of effective parameters, γ using the Jacobian found in the previous step.

3) Calculate the values for α and β using the new number of effective parameters

4) Repeat steps 1 – 3 until convergence.

In addition to being used for regularization, the effective number of parameters can be used to determine if the network would benefit from using more neurons. If the effective number of parameters are close to the total number of parameters, more neurons could mean that the number of parameters used to estimate the output would increase, thus improving the accuracy of the network. If the effective number of parameters are few compared to the total number of parameters, adding more neurons would not affect the accuracy, as all parameters needed to estimate the output are already being used.

3.2.5 Recurrent Neural Network

In order to estimate the clad stress and cold gap, it was found that a feedforward network like the one described in the beginning of this chapter does not work for most clad stress phases. The current clad stress and cold gap are dependent of their own previous values, requiring a feedback loop in order for the system to work.

A feedback system, also known as a recurrent neural network, varies from a feedforward network in the sense that the previous outputs affect the result of the next calculation. In this case, a nonlinear autoregressive neural network with external input (NARX) will be used.

Naturally, using a recurrent neural network increases the complexity of the network, requiring more time to train and lowering the accuracy compared to a normal feedforward network.

3.2.5.1 Open Feedback Loop

When modelling a system containing a feedback loop, the first step is to train it in open loop form. An open loop network is basically a feedforward network, using the previous clad stress and cold gap from STAV7 as inputs when estimating the current outputs.

Figure 3.4 shows such a configuration. As can be seen, the network has two sets of input parameters that are fed into the model. The numbers in the hidden layer, 0:1 and 1, describes

the delay of the network. In this case, the external variables ($x(t)$) which affects the output are the current inputs (0) as well as the input from one time step ago (1). Likewise, the output from the previous time step (1) is fed into the system as the second set of inputs, $y(t)$. The external inputs are already known parameters that the model does not need to calculate, such as the burnup of the fuel and the power.

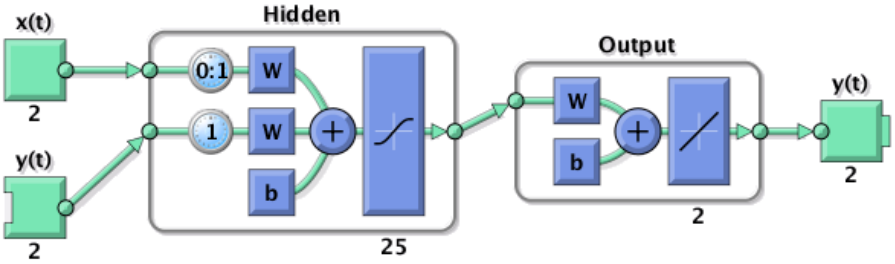


Figure 3.4- Open loop feedback neural network. The previous values are manually added from the outside, not fed back by the neural network.

3.2.5.2 Closed Feedback Loop

When applying the neural network on PCI surveillance, only the initial conditions on clad stress and cold gap are known. An open loop which has been trained using true previous output values as inputs would rapidly diverge when using the calculated outputs due to small prediction errors accumulating over time. Instead, the network needs to be trained to correctly predict the clad stress relying on its own estimation at the previous time step. This is done by closing the network and adjusting the weights so that the outputs are predicted based on the calculated previous outputs, as long as the initial conditions are known. As can be seen in Figure 3.5, only the external inputs are added from the outside once the network has been closed.

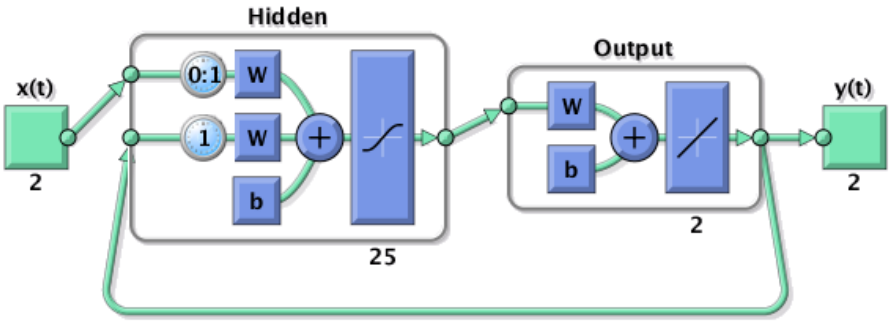


Figure 3.5- Closed loop feedback neural network. The previous values are fed back through the system, calculating the outputs based on its own previous calculations.

The reason why the network is first trained in an open loop configuration is because the closed loop training is much slower. As the feedback signals are affected by the current network, updating the weights will not only alter the output but also the input. This adds an extra layer of complexity that the open loop does not have. Training the network against an open loop thus allows for a faster adjustment of the (initially randomly assigned) weights. The open loop training is run until the accuracy is good: when the mean square error varies by less than 1% over training for 1000 epochs. The network is then closed and retrained, using the weights obtained from the open network as initial weights. By splitting the training process in this

fashion, the time required to reach convergence is much shorter than if the closed loop were to be trained from the beginning.

Due to how much faster an open loop converges towards a solution, it is also a useful tool when determining which input variables are needed for the network to reach a good solution. A closed network will always have a worse accuracy than an open network. This means that if the open network cannot approximate the desired targets, the network needs to be improved further, either by changing the inputs in some way or by adding more neurons to the hidden layer, before attempting to close it. First after the accuracy for the open loop network is deemed good is there something to be gained from closing the network (This type of analysis is performed further down, in Section 5.3).

3.3 Structure of a Neural Network Function File

The training generates a function file consisting of a number of coefficient matrices and some matrix calculations. The structure of this function file is as follows:

When the function receives an input, $\boldsymbol{X}$, consisting of Q input variables, the signal is scaled according to:

$$\boldsymbol{a}_{L1,inputs} = \begin{bmatrix} \left(X_1 - x_{1,offset}\right) * x_{1,gain} - 1 \\ \vdots \\ \left(X_Q - x_{Q,offset}\right) * x_{Q,gain} - 1 \end{bmatrix} \tag{18}$$

Where $\boldsymbol{x}_{offset}$ and $\boldsymbol{x}_{gain}$ are scaling parameters in the function file. The subscript $L1$ refers to quantities associated with the hidden layer, while quantities associated with the output layer uses the subscript $L2$. In the neural networks used in this project, the inputs are defined as:

$$\boldsymbol{X} = \begin{bmatrix} Burnup\ (MWd/kgU) \\ Power\ (kW/m) \end{bmatrix}$$

If the network contains a feedback, the network is initiated by scaling the initial output conditions, $\boldsymbol{Y}_{init}$, in the same way as the input signals. This scaling only needs to be performed the first time the network is called on (for T output variables):

$$\boldsymbol{a}_{L1,feedback} = \begin{bmatrix} \left(Y_{1,init} - y_{1,offset}\right) * y_{1,gain} - 1 \\ \vdots \\ \left(Y_{T,init} - y_{T,offset}\right) * y_{T,gain} - 1 \end{bmatrix} \tag{19}$$

The outputs are in this project defined as:

$$\boldsymbol{Y} = \begin{bmatrix} Clad\ Stress\ (MPa) \\ Cold\ Gap\ (\mu m) \end{bmatrix}$$

Both input signals are then weighted and summed up together with the bias vector:

$$\boldsymbol{n}_{L1} = \boldsymbol{b}_{L1} + \boldsymbol{w}_{L1,inputs}\boldsymbol{a}_{L1,inputs} + \boldsymbol{w}_{L1,feedback}\boldsymbol{a}_{L1,feedback} \quad (20)$$

Where $\boldsymbol{w}_1$ are the weight matrix connected to a hidden layer with N neurons (size NxQ), where and $\boldsymbol{b}_1$ are the bias vector, of size ($Nx1$). The output of the hidden layer is then calculated through a tan-sigmoid transfer function:

$$\boldsymbol{a}_{L2} = \frac{2}{(1 + e^{-2\boldsymbol{n}_{L1}})} - 1 \quad (21)$$

Which creates N output signals from the hidden layer, all scaled down to the interval [-1, 1]. The new signal is then weighted in the output layer in the same way:

$$\boldsymbol{n}_{L2} = \boldsymbol{b}_{L2} + \boldsymbol{w}_{L2} * \boldsymbol{a}_{L2} \quad (22)$$

Here, b_2 is of size ($Tx1$) and $\boldsymbol{w}_2$ of the size (TxN).

Finally, $\boldsymbol{n}_2$ is scaled up and the function returns the calculated output:

$$\boldsymbol{Y} = \begin{bmatrix} \frac{(n_{L2,1} + 1)}{y_{1,gain}} + y_{1,offset} \\ \vdots \\ \frac{(n_{L2,T} + 1)}{y_{T,gain}} + y_{T,offset} \end{bmatrix} \quad (23)$$

$\boldsymbol{n}_2$ corresponds to the scaled-down outputs. In networks containing a feedback, this vector is saved as the feedback input signal for the next time step (instead of first scaling up the output through (23) and then rescale it again through (19)):

$$\boldsymbol{n_2} = \boldsymbol{a}_{1,feedback,new} \quad (24)$$

4 REFERENCE CALCULATIONS USING STAV7

The clad stress is currently calculated using Westinghouse's thermo-mechanical performance tool, STAV7. STAV7 determines the steady-state thermal and mechanical performance in light water reactors, such as the fuel and cladding temperature, pellet-to-clad gap and clad stresses.

As previously explained, STAV7 is used for training the neural network, providing both the inputs and the outputs (targets). Some STAV7 simulations are also used for validation, to see how well the neural network manages to estimate data series not included in the training. All available output parameters generated from STAV7 are listed in Appendix 1.

The calculations are performed at 25 different nodes along a fuel rod. For this project, the neural network is only trained against one of the nodes with the highest rod power, node number 9 from the bottom. Training against additional nodes is not expected to add any extra information.

4.1 STAV7 Simulations

Three different cases have been run in STAV7 to be used for training: constant power, a single power increment and a single power reduction during the lifetime of a single fuel rod. By keeping the model simple the solution converges much faster and troubleshooting becomes significantly easier.

Furthermore, these simple scenarios cover all types of reactor operations: the power can only be increased, decreased or remain constant. Assuming that the cold gap and burnup are enough to capture the history of the pellet, these cases contains all the information needed to estimate the clad stress even for more realistic operating conditions with multiple power variations over the course of the fuel lifetime.

4.1.1 Constant Power

680 STAV7 simulations have been run at different constant power levels over the entire fuel lifetime. Starting at 2.0 kW/m, the next simulation is run at constant power 2.1 kW/m, then 2.2 kW/m etc. increasing in steps of 0.1 kW/m up to a maximum power of 69.9 kW/m.

Most STAV7 simulations are set to run until a burnup of 83.1 MWd/kgU. The calculations are performed continuously every 0.1 MWd/kgU until the burnup limit is reached and the calculation is terminated. As a result, the time steps of the data series are different between the simulations, as a higher reactor power will lead to a faster burnup. The lowest power of 2 kW/m has a time step of about 11 hours between calculations while the highest power of 69.9 kW/m has a time step of only 20 minutes.

Up to 45 kW/m the calculations are performed to the same burnup limit, but at higher powers the limit is gradually lowered. That is to simulate the behavior that the power is allowed to be higher at low burnups. The red area in Figure 4.1 below describes the maximum existing burnup at different power levels. At the highest power, 69.9 kW/m, the burnup limit is set to 0.3 MWd/kgU before the calculation is terminated.

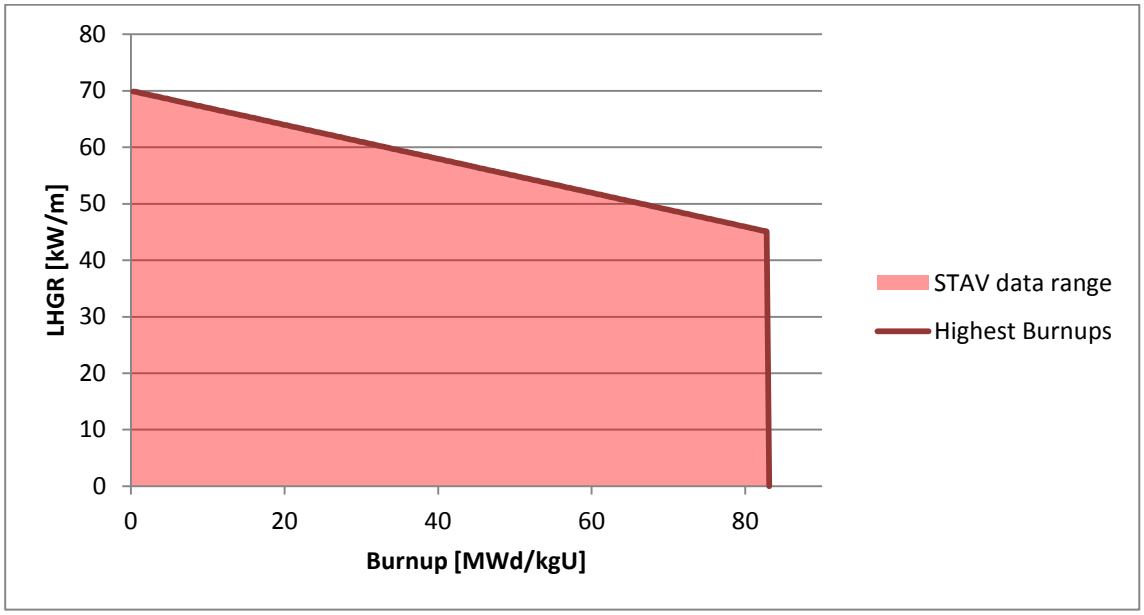


Figure 4.1- STAV7 available data range for constant power simulations. The red area marks the max existing burnup at different power levels.

4.1.2 Single Power Step

The power ramp simulations in STAV7 consist of both increased and decreased power levels. There are three parameters that are being varied in order to cover as many scenarios as possible:

- **The initial power level**. The power at the start of a simulation varies with steps of 5 between the different simulations. The available initial power levels are 5, 10, 15, …, 45.

- **Final power level**. Just like the initial power, the power level after a ramp is varied to all other power levels with steps of 5, from 5 up to 45. This includes both increasing and decreasing power, (e.g. at initial power 5 kW/m there are ramps to 10, 15, 20, …, 45 while at initial power 25 kW/m there are ramps both down to 20, 15, …, 5 kW/m and up to 30, 35, …, 45 kW/m).

- **Burnup at power step**. As seen in Section 2.2.2, the burnup of the fuel pellet has a large impact on the clad stress during a power variation. To be able to capture this phenomenon, the burnup at the time of the power step is varied between simulations. Depending on the initial power level, the burnup at the first ramp also varies. At 5 kW/m, the first ramp does not occur until an average burnup of 43.5 MWd/kgU, while at 45 kW/m the first ramp occurs at 7.7 MWd/kgU (see Table 4.1 for the lowest burnups at different initial power levels). At each burnup, STAV7 simulations to each of the other power levels are performed. After the first ramp, one ramp is performed every 1 MWd/kgU.

Table 4.1- Variation of initial power, final power and average rod burnup at ramp for the power step tests

Initial LHGR [kW/m]	Ramps to: [kW/m]	Burnup at first ramp [MWd/kgU]	Burnup at last ramp [MWd/kgU]
5	10 : 45	43.5	79
10	5, 15 : 45	36.0	79

15	5, 10, 20 : 45	31.0	79
20	5 : 15, 25 : 45	27.4	79
25	5 : 20, 30 : 45	23.9	79
30	5 : 25, 35 : 45	19.4	79
35	5 : 30, 40, 45	14.8	79
40	5 : 35, 45	11.0	79
45	5 : 40	7.7	79

Previous to a power variation, the time steps of the data series are on an hour scale. However, during a power step the power variation, and subsequently the clad stress variation, with respect to time are both instantaneous. While the time is constant, when looking at the discrete calculation steps of the data series, it can be seen that the ramp occurs in steps of 0.1 kW/m, during which the time is constant. A ramp from 5 kW/m to 10 kW/m will thus be increased according to:

$$Q = [5.0, 5.1, 5.2, \dots, 9.9, 10.0]$$
$$t = [t_{ramp}, t_{ramp}, t_{ramp}, \dots, t_{ramp}]$$

Following the ramp the time starts moving again, initially with time steps that are less than a minute to capture the effect of relaxation. As time progresses, the time steps becomes coarser and coarser and after 20 hours following the ramp the time steps have increased to about 10 minutes. At this point, the time scale is reverted back to hour scale for the remainder of the data series. The time step variation can be seen in Figure 4.2. The top right subplot describes the time corresponding to each calculation for the full data series. The steep slope at the beginning describes the time steps prior to the power increment, the flat portion in the middle describes how the time remains constant during multiple calculations over a power variation, and the steep slope at the end describes the time steps when the simulation is back to hour scale. The lower right subplot shows how the time varies after a ramp, with the time being constant at first as the power is increased, and then gradually increasing time steps during early relaxation until an abrupt return to hour scale.

The subplots to the left describes the same thing, but in a different fashion. The upper one simply shows that the time is continuously increasing over a calculation cycle. The lower figure is zoomed in around the area where the power is varied, and each ring describes a calculation. As can be seen, the number of calculations per unit time is much higher during a short span, corresponding to the ramp and the short period after.

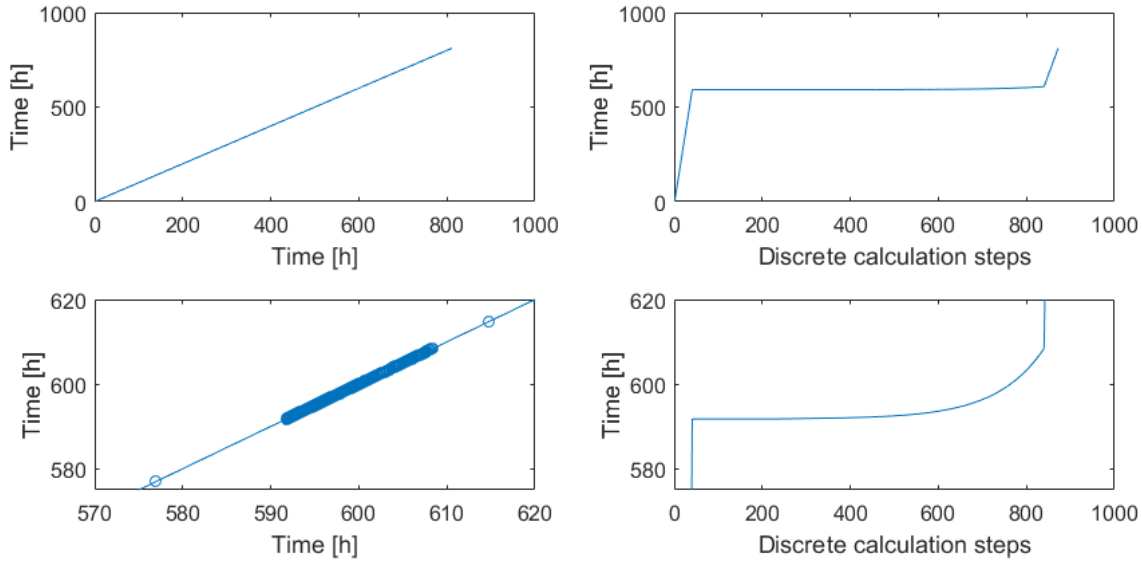


Figure 4.2- Time variation over a data series. In this case, a power ramp from 15 kW/m up to 35 kW/m at average rod burnup 40 MWd/kgU. To the left: time variation as a function of time, to the right: time variation as a function of discrete calculation steps

These intervals can be seen clearer in the upper right subplot of Figure 4.3. The left subplots show the clad stress as a function of continuous time while the right subplots show the clad stress as a function of discrete calculation steps. Due to the coarse time steps during initial constant power, the constant power only makes up for the very small first part of the data series, interval 1. The ramp is instant in the left figures, while power increases linearly in the right figures (interval 2). The clad stress increases gradually with increasing power, and the transition from soft to hard contact is clearly seen at around discrete calculation step 200. Finally, due to the size of the power step, relaxation follows the ramp. Interval 3 marks the shorter time steps after the ramp. The “discontinuity” between interval 3 and 4 occurs due to the transition from short to long time steps. As seen when looking at the time plots in Figure 4.2, it is not a true discontinuity but only appears to be because the time scales are so different.

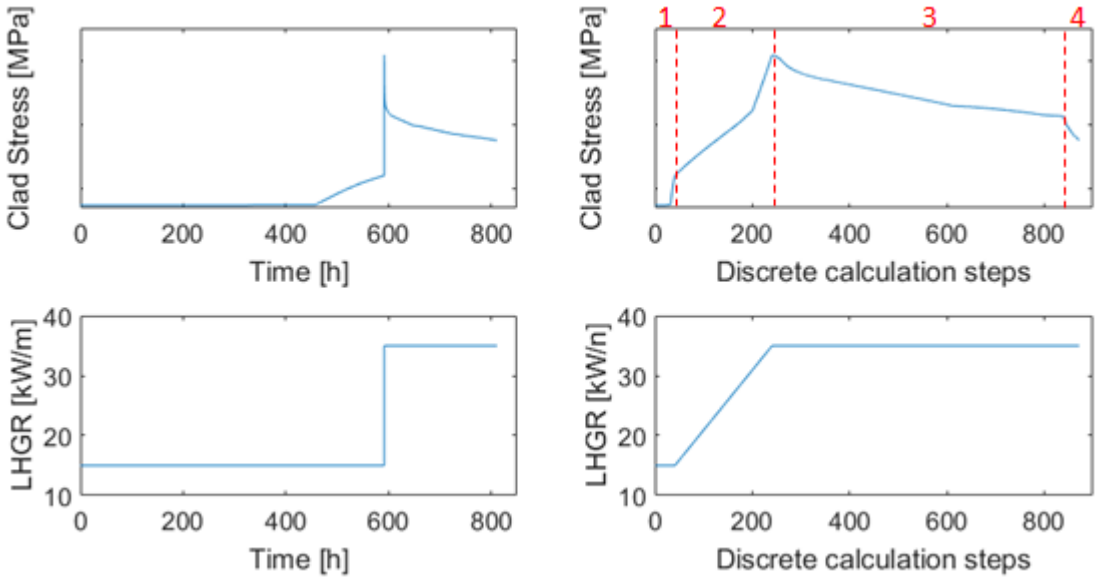


Figure 4.3- Clad stress and power variations for a ramp from 15 kW/m up to 35 kW/m at an average rod burnup of 40 MWd/kgU. To the left: Clad Stress and Power level as a function of continuous time. To the right: Clad Stress and Power level as a function of discrete calculation steps.

5 MODEL SETUP AND TRAINING

5.1 Approach

Initially, a single neural network model was assumed to be able to cover all scenarios. By training the same network multiple times against the different scenarios, it would learn to differentiate between the operations. However, while the open loop performance was good no convergence was found when closing the loop and training the network against the power ramps.

The reason is that the variables are of different importance in the different cases, with the time step being irrelevant during the ramp and the power step being irrelevant previous to and after the ramp (see Section 4.1.2). The variances were too large for the neural network to find a solution and the approach of using a single network to estimate all of the different clad stress phases was abandoned.

Instead, the PCI surveillance system will rely on several different networks, each estimating different cases during operation. Five different phases have been observed:

1) **Initial power**. During constant operation, the clad stress is approaching an asymptotic level (see Section 2.2.1). This type of behavior can be observed prior to any power changes in the core.

2) **Power Increment**. A power increment will cause a peak in the clad stress due to thermal expansion in the fuel pellet. The size of the peak depends on the current power level, the size of the ramp and the condition of the fuel pellet prior to the ramp.

3) **Power Reduction**. Inversely, reducing the power will cause a reduction in the clad stress as the thermal expansion of the pellet is diminished. Unlike the ramp up however, the cold gap is unaffected by the ramp. Furthermore, the clad stress can reach a minimum value, at which point the gap opens again and no PCI interaction occurs. The clad stress at this point only depends on the difference between the internal pressure of the fuel rod and the surrounding coolant pressure.

4) **Relaxation.** After large power increments, the resulting clad stress will exceed the equilibrium stress level. This will cause the clad creep to increase, causing a relaxation in the cladding until the clad stress is back down towards the asymptotic level. In the current design, it is assumed that relaxation is caused if the clad stress following a power increment exceeds a certain set value.

5) **Deconditioning.** Finally, a network is needed to estimate the deconditioning which occurs after a ramp down. As observed in Section 2.2.3, the reduced stress causes a lower clad creep growth, changing the initial conditions in the case of a power increment.

5.2 Filtering the Data

The available training data covers a wider range of power ramps than what is allowed during normal operation of a reactor. Including these when training the ANN would lead to the model approximating the clad stress and cold gap over an unnecessarily large operating range,

potentially reducing the accuracy in the relevant range to try and estimate stresses far outside of realistic operating conditions.

Furthermore, in some cases with large power variations the output data from STAV7 provides unreasonable clad stress behavior. In order to prevent these data series from affecting the neural networks model, the data was searched for unreasonable behavior and the series were removed.

5.2.1 Simulations Exceeding the TMOL Limit

In addition to the limitations due to PCI failure, there is also a Thermo-Mechanical Operating Limit, TMOL, to protect the fuel rod against other failure mechanisms. This limits the maximum allowed LHGR depending on the burnup in the fuel rod. The TMOL used to determine which simulations are filtered away is the limit used for Forsmark 1-3 and Ringhals 1:

Table 5.1- TMOL-curve for SVEA-96 Optima3 with ADOPT

Local Burnup [MWd/kgU]	**Local LHGR [kW/m]**
0	47
35	42
70	25

5.2.1.1 Constant Power

As explained in Section 4.1.1, data series above 45 kW/m powers are run for shorter and shorter steps, with the clad stress for the highest powers only being calculated for a burnup between 0 – 0.3 MWd/kgU. From the table above, these power levels are far above the TMOL curve. To prevent the shorter data series from potentially reducing the accuracy during allowed operating conditions, only constant powers of 50 kW/m and below are used to train the neural network model (the green line in Figure 5.1). As can be seen, all power levels below TMOL are thus still being taken into account.

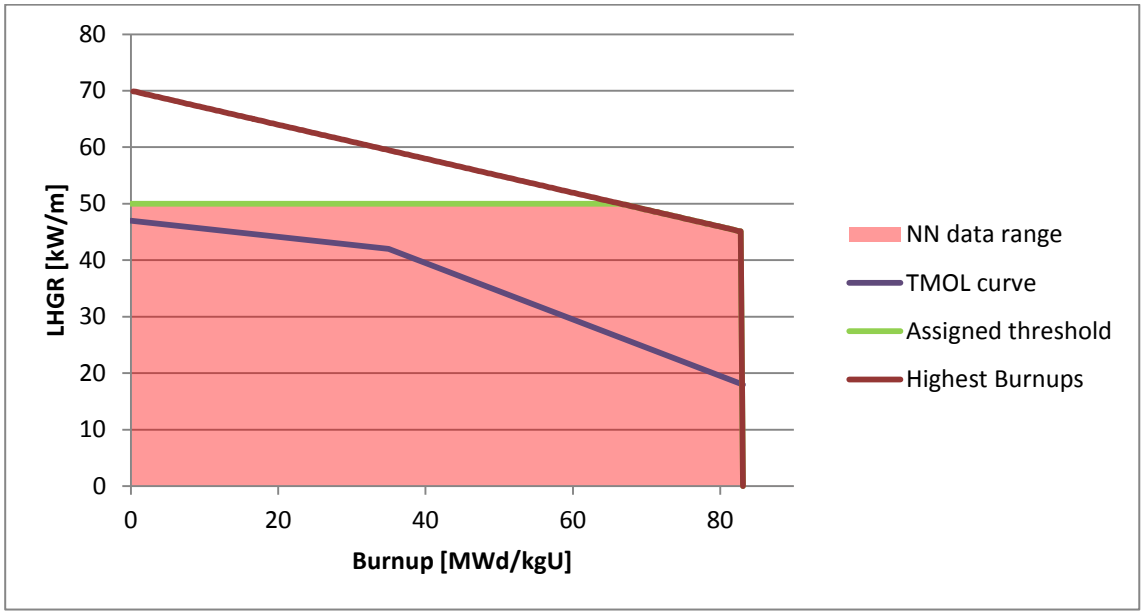


Figure 5.1- Filtering of STAV7 data series for constant power to increase accuracy below the TMOL limit

5.2.1.2 Power Variations

For the power variations, an extra margin of +5 kW/m was added to the TMOL limit so that the model is allowed to train against operations just above the approved limit. The removed simulations are illustrated in Figure 5.2. The area between the red and blue lines describes at which burnups power ramps exists.

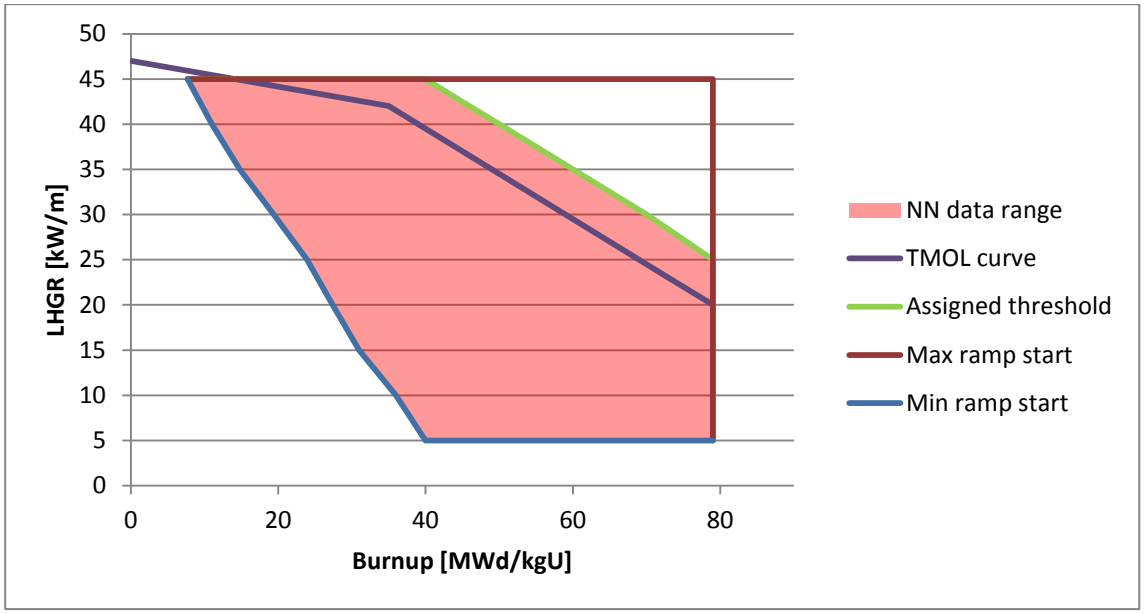


Figure 5.2- Filtration of STAV7 data series with respect to the TMOL limit. Ramps occurring in the upper right triangle are discarded.

As a result of this limitation, all power increments from 5 kW/m up to 45 kW/m are removed from the training. This is because the first ramp from 5 kW/m occurs at a burnup of 43.5 MWd/kgU, which is at too high burnup to allow the power to increase to 45 kW/m.

5.2.2 Unexpected Discontinuities in the Clad Stress

Some power step simulations contained discontinuities in the clad stress following a ramp. One such jump can be seen in Figure 5.3. As soon as the power ramp is completed, the clad stress is instantly reduced from a very high level down to a stress level corresponding to an open gap. This is caused by an instantaneous increase in clad creep, causing the gap to instantly widen to *very* unreasonable levels (as can be seen in Figure 5.3, the gap is much larger than it was at time zero). This most likely occurs as a result of STAV7 using its clad creep model well outside of its intended range, causing an extrapolation error. Due to the unphysical behavior, these simulations were removed as to not affect the neural network model. Furthermore, the high stress level at which these errors occur makes it unlikely that any reactor operation would be performed at these levels regardless.

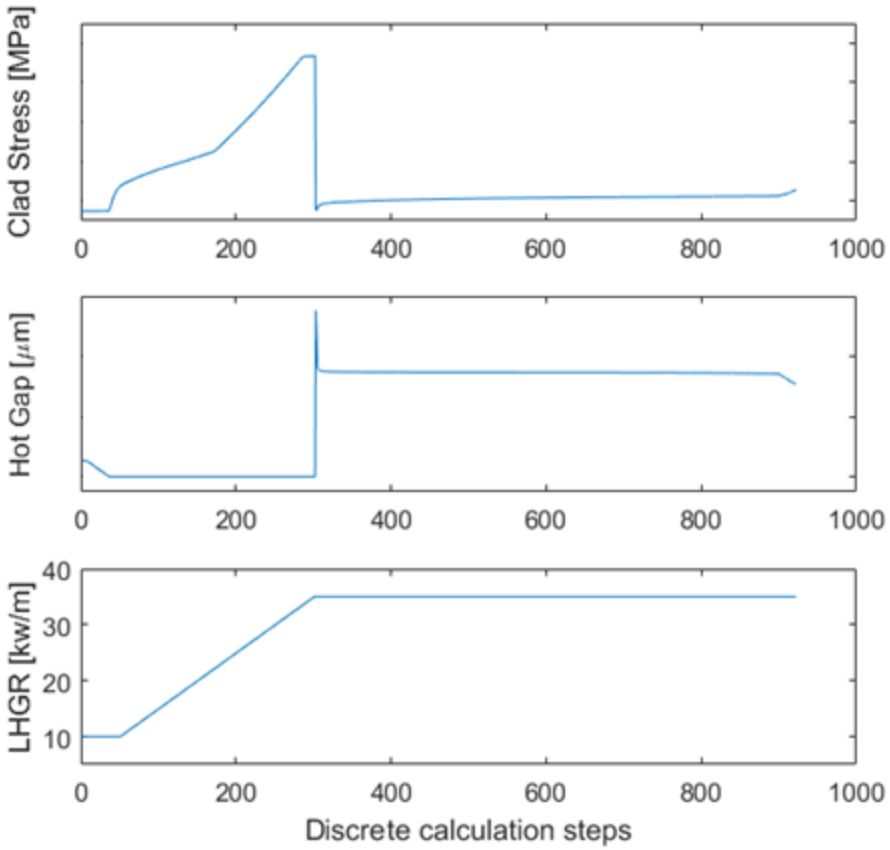


Figure 5.3- STAV7 calculations causing instantaneous stress drop and gap re-opening following a power increment, from 10 kW/m up to 35 kW/m at average rod burnup50 MWd/kgU.

5.2.3 Simulations Resulting in Irrelevant Clad Stress Levels

Some STAV7 simulations cause peak stresses far above the generic clad stress limit discussed in section 2.4, without causing the unreasonable jumps in the clad stress mentioned previously. While being accurate in STAV7, these outliers may cause an unnecessary bias in the neural network, reducing the accuracy in more relevant clad stress ranges. For that reason a safety margin of about 1.5 times the generic stress limit was used, and all simulations with a higher peak stress was removed from the data set.

5.3 The Neural Network Model

As mentioned in Section 5.1, the types of clad stress evolution can be divided into five different phases. Each phase has different inputs and/or restrictions imposed on them. In order for the neural network to obtain a good approximation of the clad stress these cases are separated into five different networks which are then linked together.

5.3.1 Parameters Based on Previous Knowledge

Most inputs are the same for all categories. The current power level and burnup is relevant for all cases. The power is linked to the temperature which in turn affects the thermal expansion of the pellet, while the burnup affects the swelling of the fuel pellet. Both of these have the advantage of already being tracked in the core, so they are readily available.

Furthermore, it has been seen that in addition to being affected by the current condition, the operating history also have a large impact on the clad stress. The history aspect is covered by including the cold gap as an input parameter, see Section 2.4.1.1. Unlike the other inputs however, the cold gap is not a variable that is being tracked by existing surveillance systems and thus first need to be calculated. In addition, the current cold gap is in turn affected by both its own previous state and the current operating condition of the fuel rod. As such, in order to use the cold gap to estimate the clad stress it is included in the neural network as a feedback output variable.

5.3.2 Parameters Found Through Tests

Attempts to estimate the clad stress using only the cold gap, burnup and power showed that some information was lacking in order to properly approximate the clad stress. As seen in Figure 5.4, this network is not very good at capturing the behavior of the clad stress.

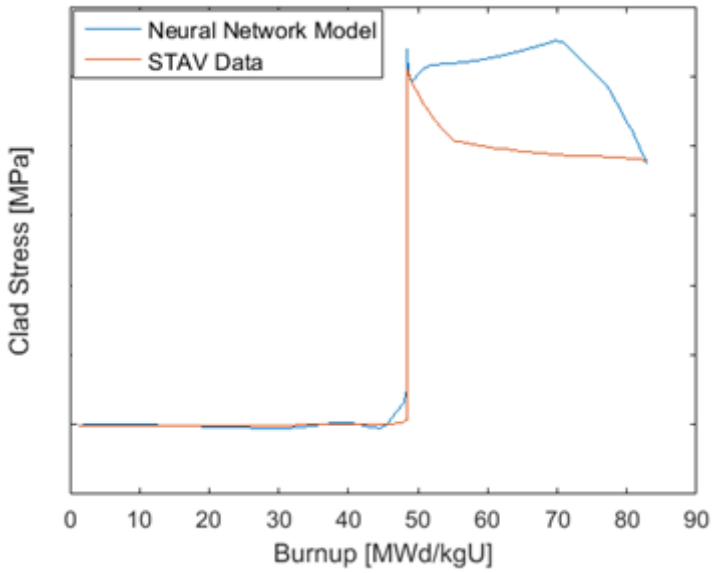


Figure 5.4- Clad Stress (open loop configuration) calculated with the neural network model using only burnup, power and cold gap compared to STAV7 calculations for a power variation from 5 kW/m to 15 KW/m at average rod burnup 43.5 MWd/kgU.

In order to find which parameter is needed to accurately estimate the clad stress, the outputs deemed the most likely to contain the missing information was added one at a time to the neural network. The most likely parameters were deemed to be:

- The internal gas pressure in the fuel rod at cold conditions, p_{cold}
- The gap between pellet and cladding during hot conditions (thermal expansion included), Δ_{hot}
- The contact pressure between pellet and cladding, $p_{contact}$

Table 5.2- Effect on performance by adding additional input variables

Number of new parameters	Input:	Performance (Mean Squared Error)
0	E, Q, ΔQ, Δ_{cold}, t	31.25
1	$Previous + p_{cold}$	22.79
2	$Previous + \Delta_{hot}$	19.65
3	$Previous + p_{contact}$	0.09

As seen in Table 5.2, the cold pressure and hot gap only has a smaller impact on the accuracy. The contact pressure however increases the accuracy significantly. When plotting the clad stress using these extra parameters (Figure 5.5), it can be seen that the clad stress is estimated very well.

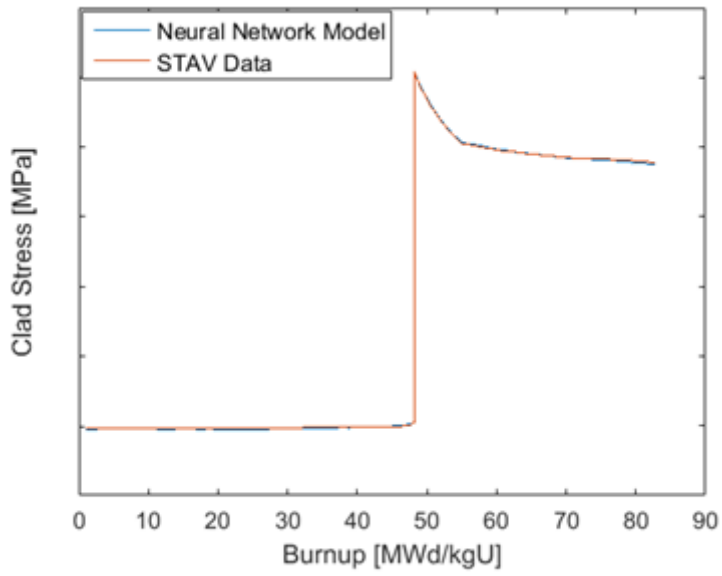


Figure 5.5- Clad Stress (open loop configuration) calculated with the neural network model after adding cold pressure, hot gap and contact pressure as additional input parameters, compared to STAV7 calculations for a power variation from 5 kW/m to 15 KW/m at average rod burnup 43.5 MWd/kgU

Note that both Figure 5.4 and Figure 5.5 are open loop simulations and **not** fully functioning models.

The contact pressure is, just like the cold gap, a parameter that is not currently tracked in the core monitoring system. As seen in Figure 5.6, the shape of the contact pressure is very similar to that of the clad stress itself. Instead of having to first calculate the contact pressure and then the clad stress, it is therefore more intuitive to instead add the clad stress itself as a feedback

signal together with the cold gap. All networks will thus be calculated using both the previous cold gap and clad stress.

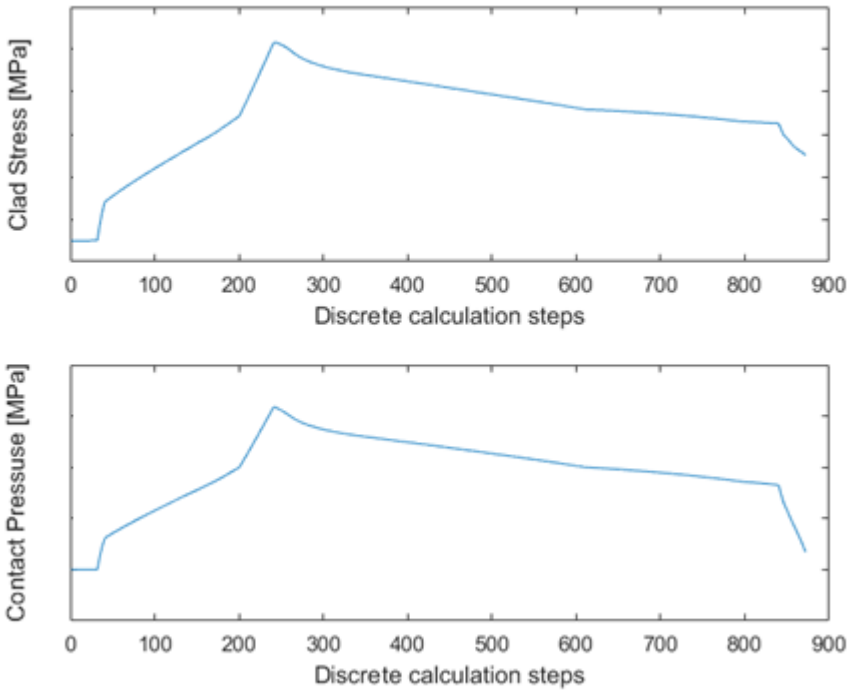


Figure 5.6- STAV7 calculations of clad stress(above) and contact pressure (below) for a power ramp, from 15 kW/m up to 35 KW/m at average rod burnup 40 MWd/kgU

5.3.3 Initial Power

During initial power the clad stress is approaching an asymptotic level. This behavior occurs at the beginning of a cycle before any power variations have taken place.

Due to the power being constant, this model is independent of any power variations. However, as the clad stress increases over time, the time step between the calculations is an important parameter. The full list of inputs and outputs for the initial power model is as follows:

Inputs: Burnup, Power, Time step, previous Clad Stress, previous Cold Gap

Outputs: Clad Stress, Cold Gap

5.3.4 Power Increase

As explained in Section 2.2.2, STAV7 performs a power ramp during a single time step, which means that the ramp is independent of the time step. Instead, the size of the power step is added, as a greater power increment leads to a greater clad stress. The initial conditions are also taken into account through the burnup and the cold gap.

Furthermore, the power steps consist of multiple instantaneous steps of 0.1 kW/m in STAV7 (see Section 2.2.2). In the intended use of the neural network, the ramps will instead consist of a single step, from the initial level directly to the final level. For that reason, the ramps are split into multiple single-step simulations. An example can be seen in Figure 5.7, where a power ramp from 5 to 10 kW/m consisting of 50 data points have been split into 50 separate cases, each estimating a single power variation from 5kW/m → 5.1 kW/m up to 5kW/m → 10 kW/m.

This split is possible as the time remains constant over the power step. The calculated clad stress from STAV7 going between two power levels during a single time step is the same, no matter if it is done in one step or split into multiple small calculations.

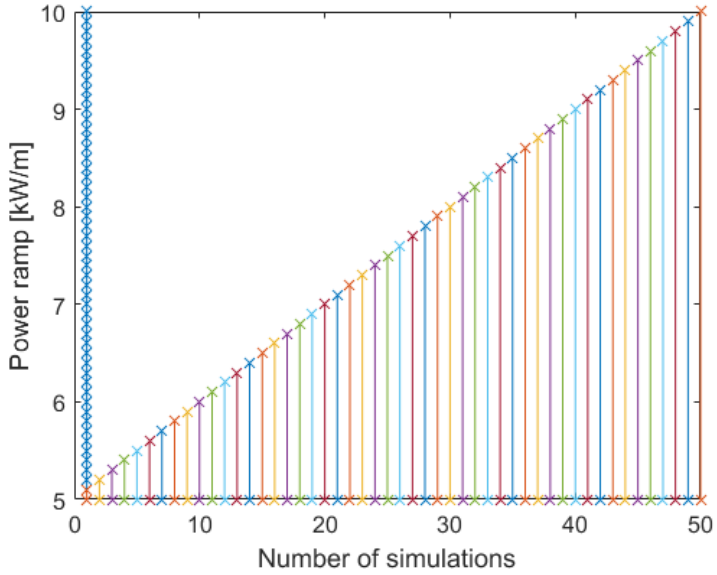


Figure 5.7- Splitting a power ramp into multiple single-step estimations

By splitting the data series into separate calculations, no feedback is needed when estimating the stress as the calculation will only be performed once for each ramp. If the conditions prior to the power increment and the size of the power step are known, the clad stress and cold gap following the power step can be calculated using a feedforward network.

The input and output parameters for the power increment model are:

Inputs: Burnup at ramp, initial Power, initial Clad Stress, initial Cold Gap, Power step

Outputs: Clad Stress, Cold Gap

5.3.5 Power Reduction

The split described in Section 5.3.4 is also performed for power reductions. However, two factors separate a ramp down from a ramp up. Firstly, while a ramp up does not reach an upper limit that cannot be exceeded; a ramp down only reduces the clad stress until the gap re-opens and the stress contribution from pellet-cladding interaction is zero. The second factor is that during ramp down, the cold gap remains constant. It is therefore not calculated and is only used as an input variable:

Inputs: Burnup at ramp, initial Power, initial Clad Stress, initial Cold Gap, Power step

Output: Clad Stress

5.3.6 Relaxation

Once again, the power is constant while the time is varying. Just like for the first model for initial power, the time step will replace the power step as an input variable. Furthermore, as the rate of relaxation is rapid at first and then slows down considerably, the slope of the clad stress is also important. Because the current clad stress is unknown, it is assumed that the derivative between the previous two points is local enough to obtain a good linear approximation. However, as MATLAB does not allow for transformation of the feedback signals, the stress slope cannot be used as a parameter of its own. Instead, the clad stress from both the previous time step and the one before that is fed back. As the time step is also known, the fed back data contains information about the slope without having to define it directly. The inputs and outputs to the relaxation model are thus:

Inputs: Burnup, Power, Time step, two previous Clad Stresses, two previous Cold Gaps

Outputs: Clad Stress, Cold Gap

5.3.7 Deconditioning

The inputs for the deconditioning model are the same as for the initial (constant power) model. However, due to the deconditioning model having to distinguish between cases where the gap is initially closed from cases where the gap is initially opened, a new network is used.

Inputs: Burnup, Power, Time step, previous Clad Stress, previous Cold Gap

Outputs: Clad Stress, Cold Gap

5.4 Differences between MATLAB's ANN Model and STAV7

One of the largest sources of error is the data transformation from the original STAV7 data sets to a format acceptable to MATLAB. STAV7 runs a simulation until a certain burnup has been reached, at which point the program is terminated. Because the fuel has different rates of burnup depending on the power of the core, the data series have very varying lengths.

MATLAB however requires all data series to be of the same length in order to train the network. The way to go about this is to pad the shorter data series using NaN. According to Mathworks documentation [12]:

"When training a network with a concurrent set of sequences, it is required that each sequence be of the same length. If this is not the case, then the shorter sequence inputs and targets should be padded with NaNs, in order to make all sequences the same length. The targets that are assigned values of NaN will be ignored during the calculation of network performance."

Furthermore, when training the recurrent neural networks for the data series requiring feedback (i.e. for initial power, relaxation and deconditioning), MATLAB assumes equidistant time steps between all data points. The time step is defined implicitly in the training algorithm and the time step does not need to be included as a separate input variable.

In the data provided by STAV7, the time steps vary greatly, both between separate simulations, but also within the same simulation as shown in Section 4.1.2. In order for the training to work, these data series must therefore be transformed into having constant time steps.

Finally, MATLAB is also limited in ways not linked to STAV7. One of the larger limitations is that MATLAB does not allow for transforming the output data before sending it back as a feedback signal. It is therefore not possible to calculate for instance the *change* of the clad stress ($\Delta\sigma$) while letting the previous clad stress be an input signal ($\sigma_t = \sigma_{t-1} + \Delta\sigma_t$). Instead, the clad stress has to be calculated directly.

5.5 Defining Inputs and Targets

Two cell arrays are needed in order for MATLAB to train the neural network; one input cell array, denoted X**,** containing the inputs to the system and one target cell array, denoted T**,** containing the desired outputs of the system (the targets). These are defined by looping over all the STAV7 simulations and extracting the required variables for the different cases.

In the networks containing a feedback loop, it is still enough to define the targets in the output cell array only. MATLAB will automatically assign the delayed target as an input when training the networks.

Defining the data for the simulations at constant power is done by limiting the highest allowed power level and loading all data series at a lower power according to Section 5.2.1.1. During constant power, the burnup is proportional to the time (the fuel is being utilized at a constant rate). Instead of using equidistant time steps, equidistant burnup steps are therefore used, with one clad stress calculation being performed every 0.1 MWd/kgU. While this means that the time step varies between different simulations (as stated in Section 4.1.1, 0.1 MWd/kgU corresponds to 11 hours at the lowest powers and 20 minutes at the highest powers), using

burnup instead of time suits better with the way the data series are defined in STAV7, keeping the different simulations closer to the same length.

The power ramp simulations need to be split into several separate cell arrays depending on if they are describing power increments, power reductions, relaxation or deconditioning. This is done according to the scheme in Figure 5.8. The first guideline is to check if a power variation is taking place or not, which is done by comparing the current power with the power one step earlier. Once they are located, the ramps are split into single input-output pairs and are sorted into one of two cell arrays, depending on if the power is increasing or decreasing.

If the power is constant, a check is performed on whether a power step has taken place earlier or not. The data prior to a power variation is discarded, as the case of initial constant power is trained against the more extensive data from the STAV7 constant power simulations.

If a power step has already been performed, the next check is whether the power has been increased or decreased. If the power has been reduced, the remaining data series is sorted into the deconditioning cell array. Just like for initial power, deconditioning is a slow process and the data series is transformed into having equidistant burnup steps of 0.1 MWd/kgU.

Finally, the data following a power increment is studied. Only ramps resulting in a high clad stress will cause relaxation, the rest are assumed to continue in the same fashion as prior to the ramp towards equilibrium. The separation between power increments which does or does not cause relaxation is done by checking if the ramp caused the clad stress to exceed the imposed relaxation limit, σ_{relax}. If the max stress is lower, no relaxation is expected and the data is discarded (the effect is assumed to be covered by returning to the initial power model, with the new clad stress and cold gap). If the clad stress is higher however, the data is saved.

Relaxation is a fast process relative to initial power operation and deconditioning. Only performing one calculation every 0.1 MWd/kgU is too sparse to capture the phenomena. Instead, equidistant time is used, with calculations being performed once every 5 minutes. Furthermore, as relaxation is a fast process, the relaxation process is assumed to be finished after about 200 hours, at which point the clad stress is assumed to have been reduced back down to equilibrium level. For this reason only data for up to 200 hours following a power increment causing relaxation is saved.

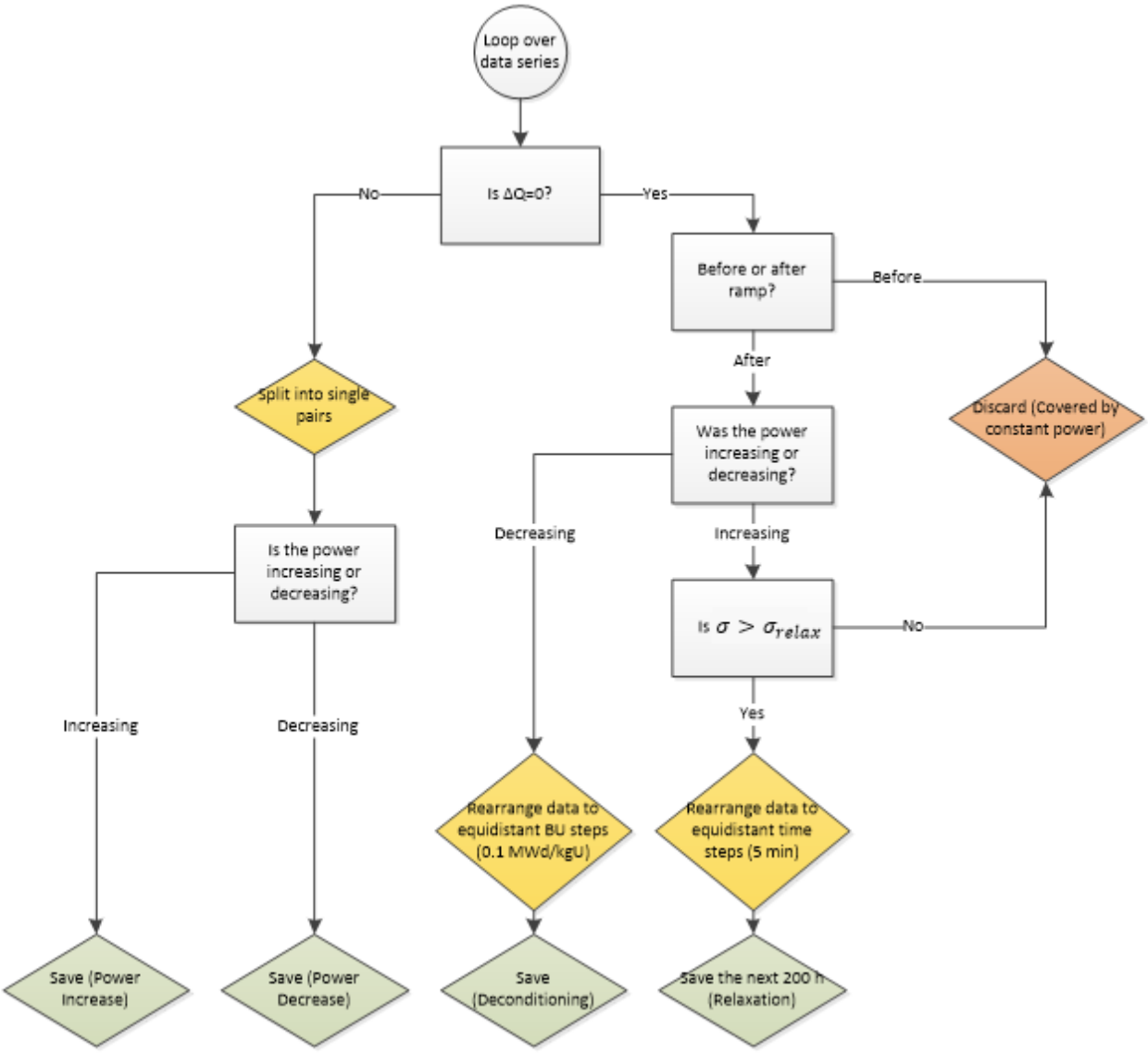


Figure 5.8- Procedure of splitting the STAV7 simulations into different cell arrays. The rectangular boxes represent decisions and the diamonds represents actions.

6 CALCULATION RESULTS

The results of the neural network optimizations are presented below in two ways. First, the accuracy of the full data series is shown through variation plots, where the difference between the clad stresses and cold gaps calculated by the neural network and STAV7 are plotted against their respective targets from the STAV7 calculations:

$$\Delta\sigma = \frac{(\mathrm{T}_{\mathrm{stress}} - Y_{stress})}{Y_{max}} \qquad (25)$$

Where T_{stress} is the STAV7 clad stress, Y_{stress} is the corresponding neural network clad stress and Y_{max} is the highest stress in the simulation.

Figure 6.1 shows an example of such a variation plot. The STAV7 calculations are given on the x-axis and the deviation of the neural network is plotted against the y-axis.

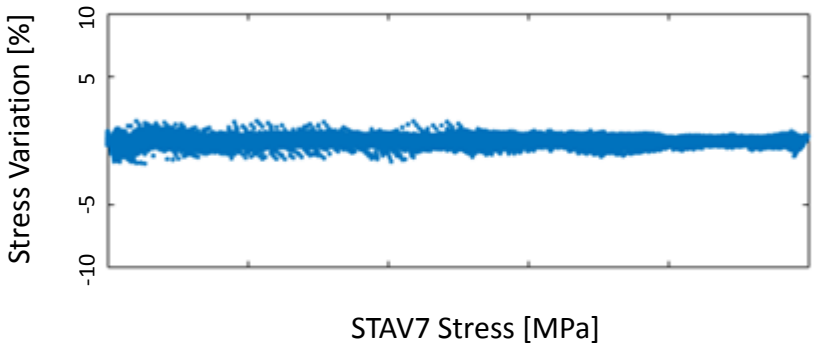


Figure 6.1- Example of a variation plot.

In addition to the variation plots, some specific results are also presented, to give a better grasp on the magnitude of the deviations shown in the regression plots. These are generally chosen to correspond to some of the larger deviations in the regression plots so as to give a more circumstantial understanding of the size of the deviations.

Both the variation plots and the chosen results are calculated using the *test data set*, i.e. the data set independent of the network training.

In the case of initial power, deconditioning and relaxation, only the closed loop results are presented. The open loops are just a first step of the optimization and the results are not interesting more than for making sure that the input parameters contain the information needed in order to estimate the clad stress.

6.1 Initial Power

A neural network using 30 neurons in the hidden layer was used to estimate the clad stress and cold gap for initial power. As can be seen in the variation plots (Figure 6.2), the results are very good. The neural network clad stress never deviates from the STAV7 stress by more than 5% relative to the largest STAV7 stress. The deviations appear to be slightly larger at lower

stresses, which is to be expected due to the sudden stress increase which occurs the moment the gap closes.

Likewise, the neural network cold gap varies by about ±2% relative to the largest STAV7 cold gap. In this case, the larger deviations occur both at the small cold gaps (when the gap closes) but also somewhat close to the top. The larger variations at large cold gaps are due to a discontinuity caused by the transition from pellet densification to pellet swelling.

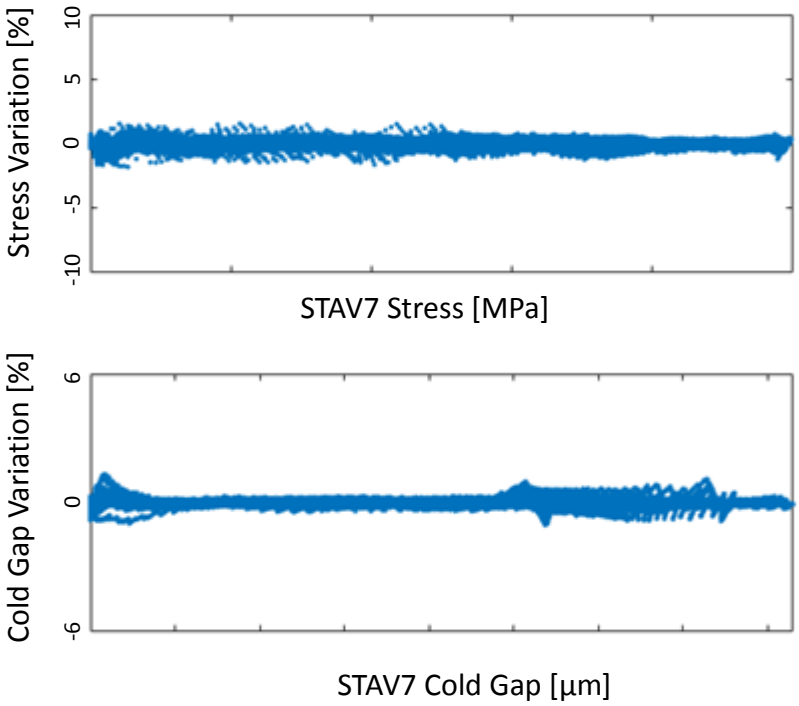


Figure 6.2- Variation plots for the initial power model. Top: Clad stress, bottom: Cold gap

An example can be seen in Figure 6.3. As mentioned, the cold gap can be seen to deviate close to the densification-swelling discontinuity at about 10 MWd/kgU in Figure 6.3. The clad stress variations appear to be too small to be seen however.

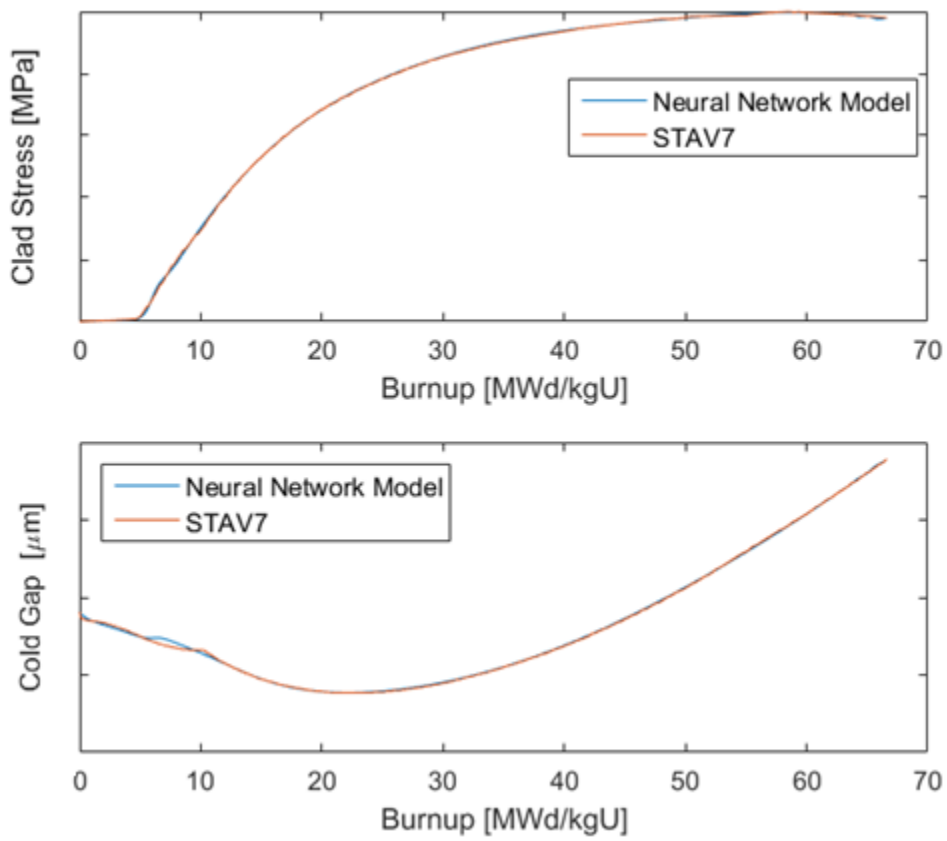


Figure 6.3- Clad stress (left) and cold gap (right) estimation, constant power 49.9 kW/m

6.2 Power Increments

The power increments were modelled using a feedforward network containing 35 neurons in the hidden layer. Due to how much faster feedforward networks converge, it was possible to test against a larger span of neurons without being restricted by the time it took to train the network.

As the variation plots show (Figure 6.4), the network estimates the clad stress within $\pm 2\%$ from the STAV7 calculations. Most of the variations can be seen close to the middle, and is caused by the sharp transition from soft to hard contact. The cold gap never deviates by more than $\pm 1.25\%$ from the STAV7 data, with the largest deviations being close to when the gap closes. Both of these deviations can be attributed to the regularization which smoothens sharp discontinuities.

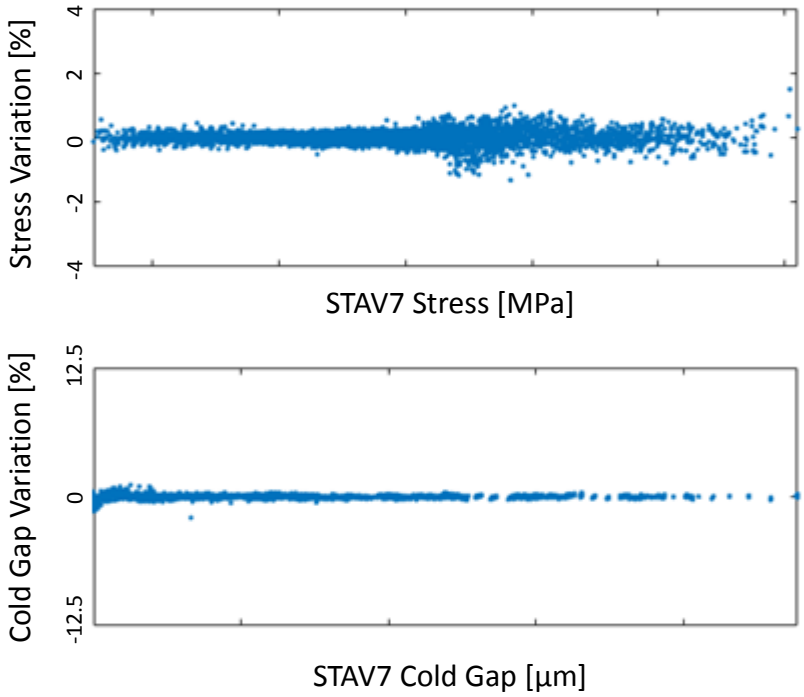


Figure 6.4- Variation plots for the power increment model. Top: Clad stress, bottom: Cold gap

Due to the data series being split, there is no power ramp consisting of only test data. Each output is a separate calculation however, with no direct link to the others. This means that 20% of each ramp is estimated using test data and 80% training data. As a result, the clad stress and cold gap is plotted in Figure 6.5 every 0.6 kW/m, to only use the model against the test data. The clad stress can be seen to follow STAV7's calculations well both prior to and following the transition from soft to hard contact. Just like for initial power, the regularization will make both the transition between hard and soft contact as well as the gap closure smoother. This is more apparent for the cold gap, but can largely be solved by adding a condition on the cold gap to always be equal to or greater than zero.

The clad stress diverges more for power increments than for initial power in terms of MPa, but as seen in Figure 6.4 the accuracy is still very good. The last clad stress calculation in Figure 6.5 corresponds to the largest diverting clad stress in the test data (the point furthest up to the right in Figure 6.4). Relative to the total clad stress, this deviation is effectively negligible.

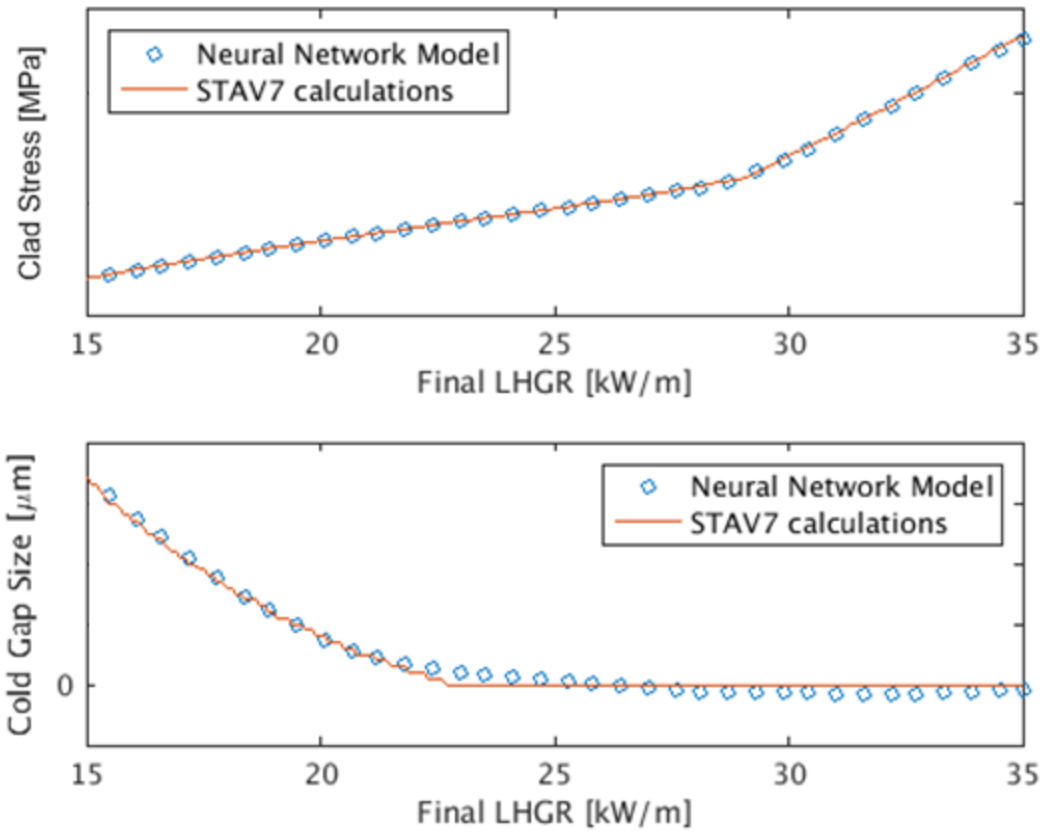


Figure 6.5- Clad stress (left) and cold gap (right) for power steps starting at 15 kW/m up to 35kW/m at average rod burnup 44 MWd/kgU.

6.3 Power Reductions

Similar to the power increments, the power reductions are performed by a feedforward network containing 35 neurons in the hidden layer. As mentioned in Section 5.3.5, the cold gap is constant over the power reduction and it is not calculated by this model. Figure 6.6 shows that the clad stress during ramp down never varies by more than about $\pm$1% from the targets.

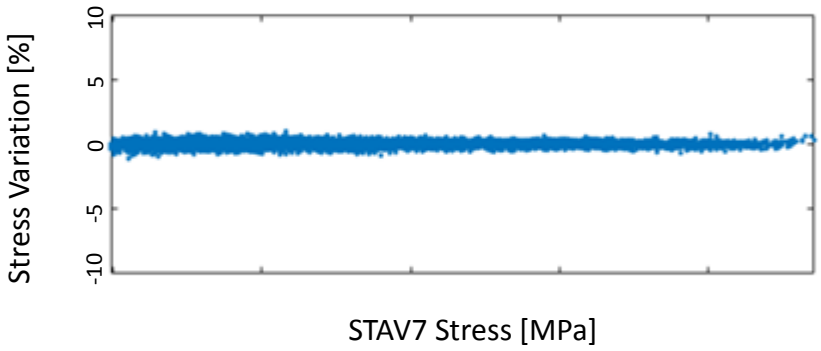


Figure 6.6- Variation plot for the power reduction model.

When plotting an example (Figure 6.7), the test data can be seen to follow the clad stress very well, including the transition to where the gap re-opens (when the power is reduced to just above 10 kW/m).

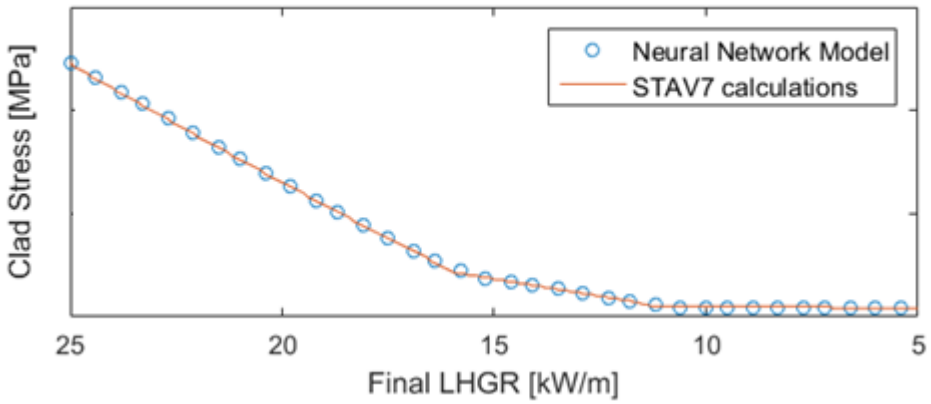


Figure 6.7- Clad stress during power reductions, starting at 25 kW/m, decreasing down to 5 kW/m at an average rod burnup of 36 MWd/kgU.

6.4 Relaxation

Like the model for initial power, the relaxation model has 30 neurons in the hidden layer. However, as can be seen in Figure 6.8, the accuracy for clad stress is about ±5%. The larger variations are mainly observed at high stresses, but there appears to be one data series deviating at lower stress levels as well. While this is worse than for the previous networks, the deviations are still relatively small. Similarly, the cold gap varies by about ±6% which is also worse than previous. In this case, most of the variations occur close to the gap closure, but there appears to be one data series deviating at a higher cold gap as well. This most likely corresponds to the single deviation for low clad stress.

Unlike the earlier models, the relaxation training was stopped prematurely due to time constraints. Given more time and more neurons in the hidden layer, the accuracy may be improved further.

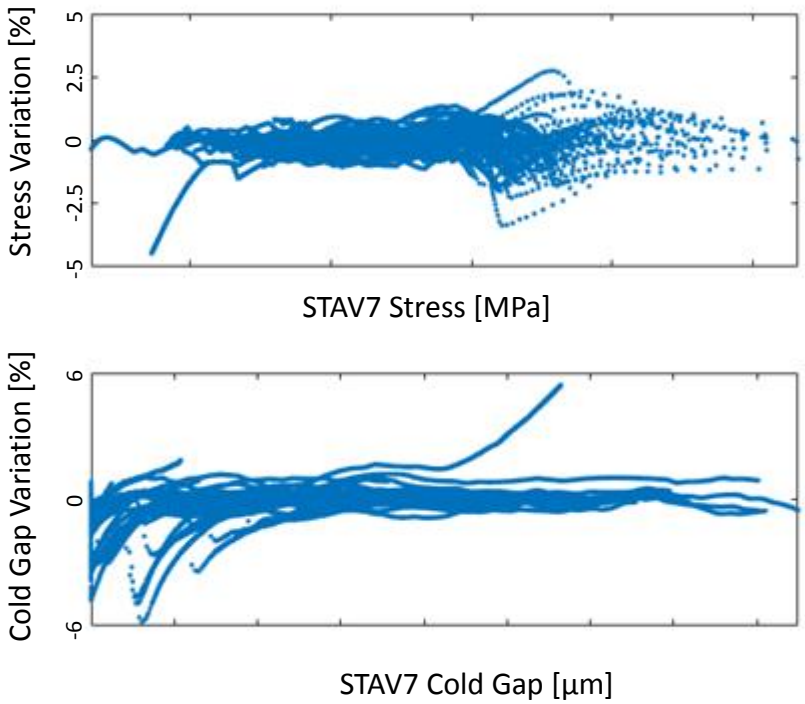


Figure 6.8- Variation plots for relaxation. Top: Clad stress, bottom: Cold gap

The diverging low stress test case is illustrated in Figure 6.9. As can be seen, the estimated clad stress is slightly higher than the expected clad stress. The cold gap appears to increase too rapidly very early, but then drop off, becoming lower than expected. This then changes the conditions for the rest of the relaxation series, which may be the reason why the clad stress decreases faster than expected. This type of divergence has previously occurred during training when it has been manually cancelled before convergence is reached, which suggests that the early stop may be the cause of the poorer accuracy.

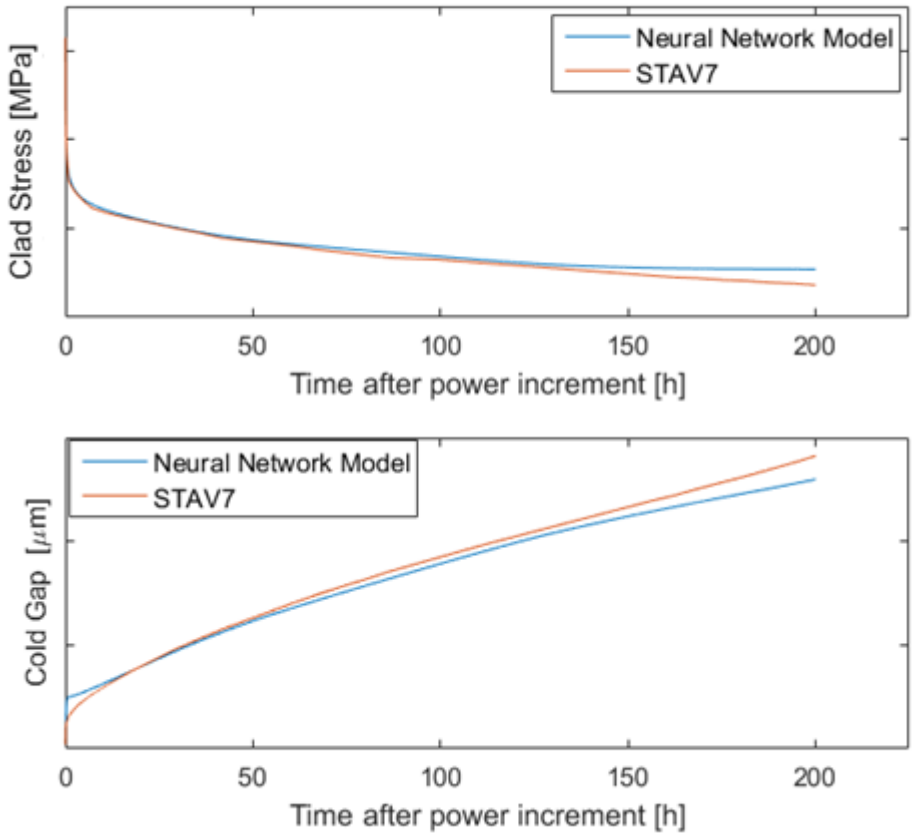


Figure 6.9- Most diverging relaxation case, following a power increment from 10 kW/m up to 35 kW/m at average rod burnup 36 MWd/kgU

6.5 Deconditioning

Similar to relaxation, the network calculating the deconditioning was cancelled due to time constraints before it was fully trained.

While the network, shown in Figure 6.10, is more accurate than the relaxation model, it still exceeds ±5%. The cold gap varies by ±5%, which is also more than for the other networks, barring relaxation.

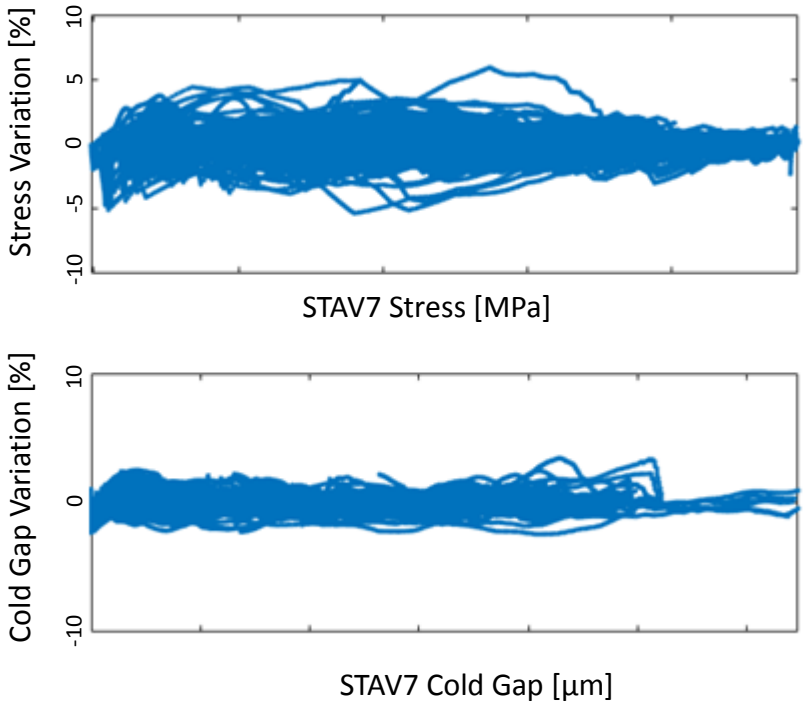


Figure 6.10- Variation plots for the deconditioning model. Top: Clad stress, bottom: Cold gap

The main challenge for the deconditioning model is to be able to distinguish between when the gap has re-opened and when it remains closed. Figure 6.11 and Figure 6.12 show two different test cases, in the first the gap remains shut and in the second the gap has re-opened. As can be seen, the model appears to be able to distinguish well between the cases.

An important note here is that the figures below are described as starting from burnup zero after a power reduction. In reality, the burnup of the fuel rods obviously does not reset when the power of the reactor is altered. This is due to a modelling error when transforming the data from STAV7 to MATLAB and due to time constraints, re-training was not possible during the time frame of the project. Nevertheless, these plots gives a good indication on how well the parameters used can estimate the clad stress and the cold gap.

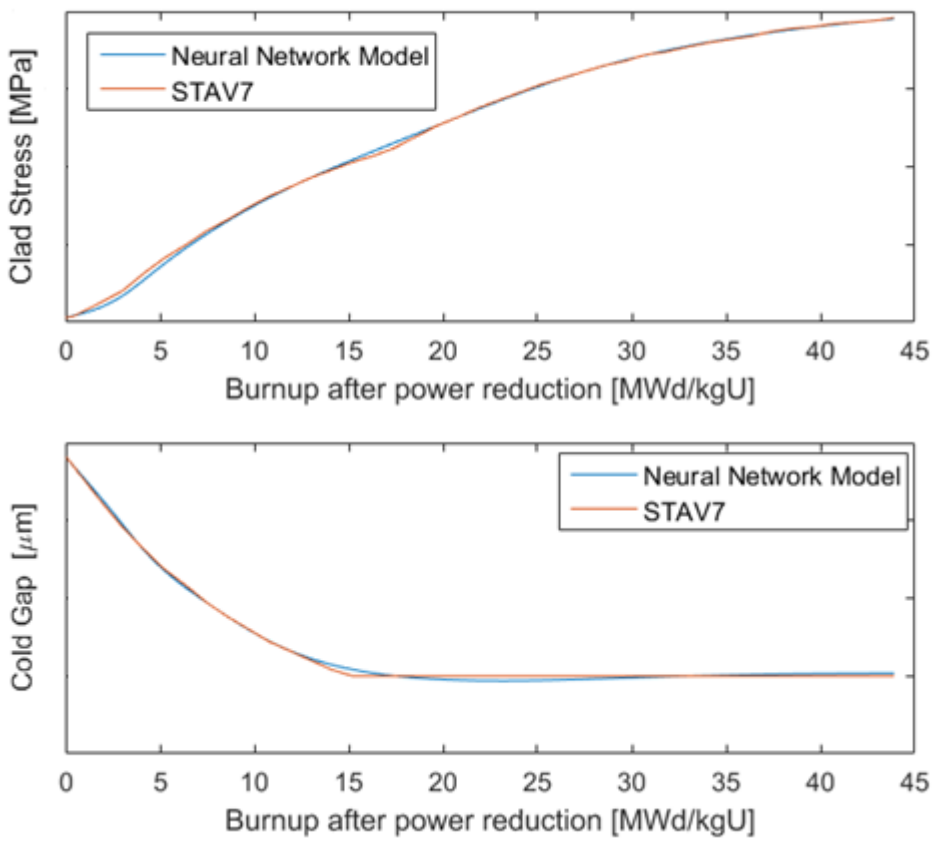


Figure 6.11- Clad Stress and Cold Gap following a power reduction ***not*** *causing the gap to re-open following a power reduction from 25 kW/m down to 10 kW/m at average rod burnup 36 MWd/kgU*

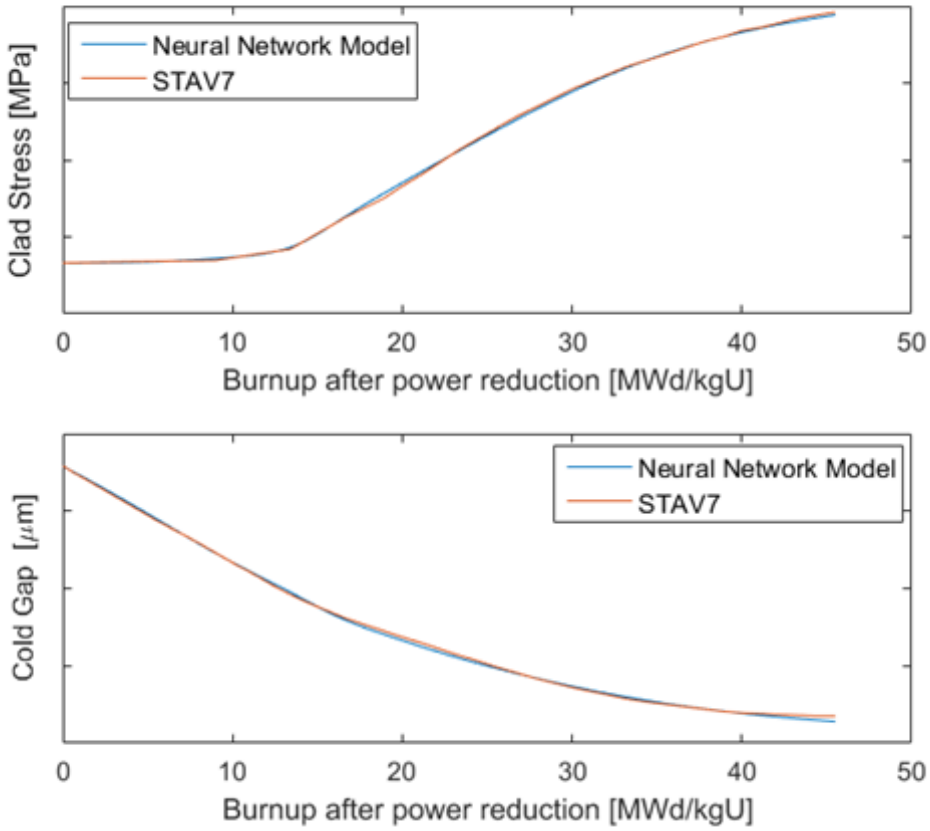


*Figure 6.12- Clad stress and cold gap estimation following a power reduction **with** the gap re-opening following a power reduction from 40 kW/m down to 15 kW/m at average rod burnup 32 MWd/kgU*

7 DISCUSSION

The results of the neutral network models presented in this report have been shown to be of a satisfactory level. However, it is possible that errors will begin to grow over time when switching between the models representing the different clad stress evolution phases. For instance, a slightly larger initial clad stress prior to a ramp could lead to a slightly higher peak stress and so on. For this reason, it is still recommended to train the deconditioning and relaxation models until the performance has converged. Earlier attempts at training the network has shown that stopping the network before it has converged can lead to relatively large errors, even if the performance appears to be good.

7.1 Limitations

While the results are good, there are several limitations imposed on the current model in order for it to work.

During the training of the initial power network, it was found that the cold gap varies at zero burnup depending on the initial power level. This is due to the pellet relocation (cracking), which occurs at all power levels above 4 kW/m, increasing at higher power levels [2]. In order for this model to work, an initial cold gap calculation needs to be performed so that the initial cold gap corresponds to what the network has been trained against.

Possibly, the largest limitation is that the neural network has trained three out of five models using equidistant time steps. In the case of initial power and deconditioning, the variations are so slow that it is assumed that performing one calculation every 0.1 MWd/kgU is enough, while the relaxation is a fast process, requiring calculations every 5 minutes in order to capture the behavior properly. Should the core surveillance have longer time steps, it is possible that the relaxation model will not work properly.

An alternative would be to not rely on MATLAB for the development of the neural network, but instead write the code for training a neural network from scratch. By doing it this way, it would be possible to transform the output variables of the network before returning them as feedbacks, which could help in making the network more suited for this specific task. This approach was not selected in this project though, as the time required would be much larger, with little to no prior experience in using neural networks for this kind of task.

Most of the limitations are concerning the relaxation model. In addition to having much shorter time steps (which requires a finer interpolation), the relaxation model relies on using two previous time steps (Section 5.3.6) in order to capture the time derivative of the clad stress. When the model is initially called upon, only one earlier point is known from the power increment model. Using the point prior to the ramp would mean that the initial derivative would be positive while all the following ones are negative, leading to potential errors. During the current training, it is simply assumed that the initial two points are both known in order to initiate the training and to study the feasibility of the concept.

Another limitation set upon the relaxation model is the way it is called and for how long it lasts. In some cases, (like in Figure 5.5) there is relaxation occurring even if the peak clad stress does not exceed the initiation criteria for the current relaxation model. Similarly, the relaxation may continue for longer than the 200 hours the network is trained against, meaning that both conditions for calling on the network needs refining. A full analysis of the conditions

causing relaxation is outside the scope of this project but is needed before the network can be trained and used for real applications.

7.2 Sources of Error

Because the network model will be based on calculations made with STAV7, the solution will be limited by the accuracy of STAV7. While the data has been filtered to remove possible errors, there is still the risk of some smaller irregularities remaining within the training data.

The data transformation from STAV7 output to MATLAB input could also affect the accuracy in some way. For the coarser data series, extra data points have been added through linear interpolation. Furthermore, the STAV7 data has been smoothed out in places where the data was inconsistent. It is possible that these transformations has somehow excluded or changed some information which affects the results.

In addition to the previously mentioned limitations of the relaxation model, the first point in the training data are in some cases larger than the final clad stress of the power increments. This is due to thermal expansion, which while being almost instantaneous, continues for about a minute after the power has reached its final level. To avoid a single positive derivative, the initial condition for the relaxation model is taken when the clad stress has reached its peak. As shown in Figure 7.1, this can in some cases be noticeably higher than the max clad stress estimated by the power step model. This leads to an initial error in the relaxation model, regardless of how well the power step model works.

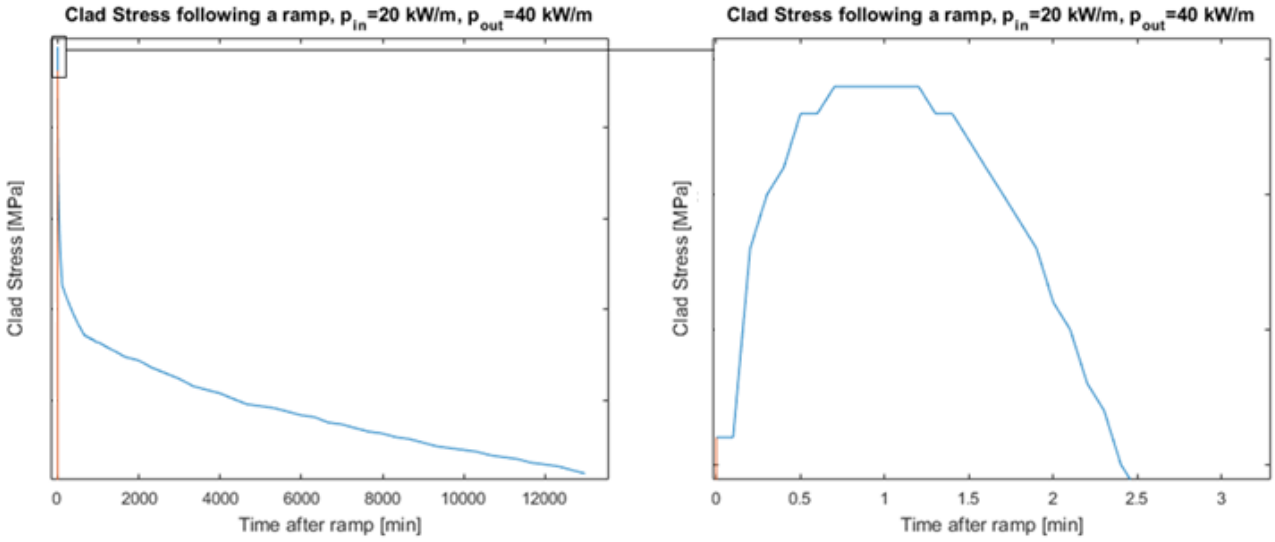


Figure 7.1- STAV7 calculations of clad stress directly following a power step from 20 kW/m up to 40 kW/m at average rod burnup 35 MWd/kgU. The highest stress occurs after 1 minute following the power increment.

7.3 Future work

While not providing a complete solution to clad stress estimation for PCI surveillance, this study shows the potential as well as the limitations of using neural networks instead of relying on more conventional calculation methods.

The current models appear to yield good results. However, if needed, the feedback models could be trained further for further improved accuracy and reliability. Comparing the effective

number of parameters to the total number of parameters in Table 7.1, it can be seen that for almost all cases, the number of effective parameters are close to the total number of parameters. As stated in Section 3.2.4, this is an indication that more neurons could improve the accuracy of the solutions. Furthermore, both the deconditioning and relaxation modelling were interrupted before the solution fully converged. Allowing for more neurons and more time, the accuracy could potentially be improved for all networks.

Table 7.1- Effective number of parameters against total number of parameters for the different networks.

Model	Neurons in hidden layer	Effective # of parameters	Total # of parameters
Initial Power	30	212	212
Power Increment	35	273	282
Power Reduction	35	238	246
Relaxation	30	272	272
Deconditioning	30	211	212

There are still a number of checks that needs to be made before the model can be used for real application. The most important task remaining to be performed is to see how well the model can handle several power variations during the fuel lifetime. The current model is based on the assumption that the cold gap is able to capture the fuel history without the need of any other parameter. The model only just progressed to a state where it can be tested against such a case so the assumption, while reasonable, has yet to be fully proven.

One option that has been discussed in the case of the cold gap not being enough is to separate it into smaller components. Early on in the cycle, the size of the cold gap is mainly determined by the (lack of) swelling in the fuel, while at the end of the cycle, the size of the cold gap is mainly determined by the clad creep. As a result, the same total cold gap can occur at different times during the fuel lifetime due to different effects being dominant. While the cold gap is the same in both cases, the difference in clad stress peak is still is very large. By separating these terms, the solid swelling can be included as a linear burnup-dependent input variable [2], while the clad creep is used as a feedback variable. However, it can be assumed that this effect is already captured by the current model as the burnup is also used as an input parameter.

As mentioned in Section 7.1, the relaxation model requires more development work before it is complete. First, the condition triggering the relaxation model needs to be refined. There are currently three propositions on how to proceed. The first one is to try and train the same model for all scenarios following a power increase, which would be able to distinguish between when a relaxation is triggered or not by itself, much in the same way that the deconditioning model is able to distinguish between when the gap has re-opened and when it remains shut. The other option is to investigate the effect of relaxation closer and finding a more accurate initiation condition. In this case, the power increases not causing relaxation would instead be calculated using the initial power model.

Second, the model requires two previous clad stress and cold gap values, which creates an initiation problem as only one prior clad stress and cold gap value is known from the power increment model. The proposed approach to solve this is to find a correlation between the initial slope at different clad stresses and cold gaps. If such a correlation can be found, an extrapolation function can be created from which a made-up starting point can be calculated to

initiate the model. Such a configuration is shown in Figure 7.2. The relaxation model could then be initiated using the extrapolated point together with the clad stress from the power increment model.

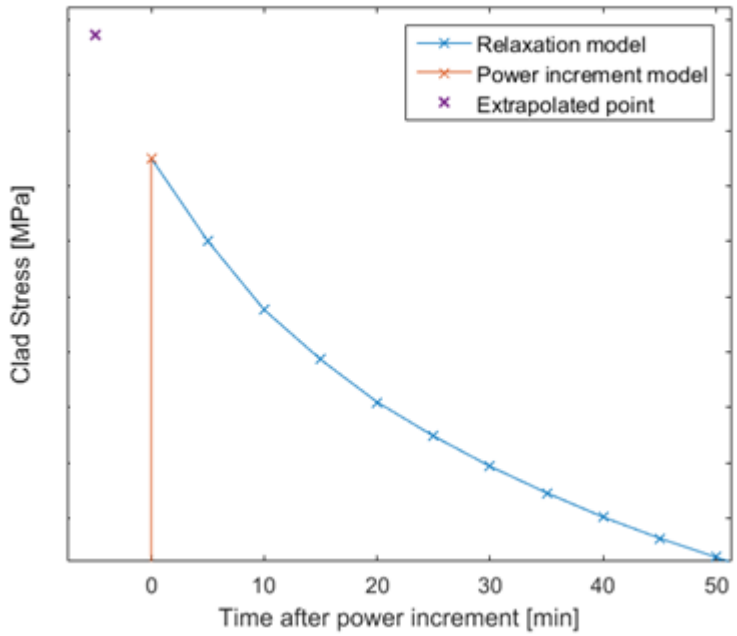


Figure 7.2- Example on how the relaxation model could be initiated by using an extrapolated starting point.

Third, there is currently no network which estimates the clad stress for the period following a ramp which does not cause relaxation. It is assumed that the initial model is able to, with use of the new cold gap, estimate the clad stresses in these scenarios, but this is yet to be confirmed. It is possible that the relaxation model can be trained for all scenarios following a power increment, which would require more extensive training.

Finally, as mentioned in section 6.5, the current model of deconditioning is resetting the burnup of the fuel rods. In order to obtain a useable neural network, this needs to be re-run, using the real burnup for the different test cases. The training was set up to perform this, but due to time constrains the project was finished before this new training had been completed.

8 CONCLUSIONS

This project has increased the understanding on how pellet-cladding interaction affects the clad stress during different reactor operation situations. The different phases of clad stress evolution have been identified, as well as the relevant parameters for each of these phases.

The results show that it is feasible to use neural networks to replicate calculations currently performed by STAV7, such as estimating the clad stress and the cold gap, with satisfactory accuracy and reliability. It is based on these results that it can be concluded that the relevant parameters found do contain all the information needed for the respective phases.

Once trained, the neural network calculates the outputs using matrix calculations. The network is therefore very fast, being able to perform over a million parallel calculations in just a few seconds. The network thus fulfills the criteria on calculation speed as it is definitely fast enough to be used in a core surveillance system.

This project has laid the foundation for further work within the field of neural networks. There are still challenges remaining, especially concerning how to initiate the relaxation model. Furthermore, tests against more complex scenarios are required, both to validate that the cold gap contains enough information about the history of the fuel pellet and that switching between the networks multiple times will not create a prediction error.

Having to use networks containing feedback loops makes the networks somewhat rigid and forces the use of equidistant time steps. Two of the networks, the power steps, are estimated using a feedforward network, however, and does not require any feedback. Should the remaining networks prove to be too rigid, the power step networks may still be useable by combining them with using a different approach for the operation in-between the power variations.

Finally, once satisfied with the performance of the networks, the functions calculating the clad stress can be translated into another language for implementation in POLCA. While the training algorithm is quite complex, the trained network only contains a number of weight and bias matrices and two transfer functions. Translating the network to another language should therefore not prove to be any sort of challenge.

For access to the full report, the reader is referred to the original report, property of Westinghouse Electric Sweden AB.

9 REFERENCES

[1] TIF265 Nuclear Materials (2015)
Lecture 10, Zirconium
Mattias Thuvander

[2] ABB Atom Report BU 97-091 (1997)
STAV7.0 Model Description, *p.42-43*
A.R. Massih and Z. Weiss

[3] Westinghouse Electric Sweden AB Report BT 13-1184, rev 0 (2013)
Methodology for managing PCI failure risks in BWRs
P. Seltborg, J. Casal, G. Zhou, H. Carlsson, P. Jourdain and S-B Johannesson

[4] Westinghouse Electric Sweden AB Report BTE 15-0789, rev 0 (2015)
PCI Model PCI-EYE
S-Ö Lindahl

[5] Mathworks, 2005 [ONLINE] http://matlab.izmiran.ru/help/toolbox/nnet/ *Multiple Layers of neurons*

[6] J. Heaton (2002) "Introduction to Neural Networks with Java", *p.158*

[7] D. Kriesel (2005) "A Brief Introduction to Neural Networks", *p.99-100*

[8] N. Nikolaev, H.Iba (2006) "Adaptive learning of polynomial networks: genetic programming, backpropagation and Bayesian methods", *p. 153*

[9] M.T. Hagan, M.B. Menhaj (1994) "Training Feedforward Networks with the Marquardt Algorithm"

[10] D. F. Foresee, M.T. Hagan (1997) "Gauss-Newton Approximation to Bayesian Learning"

[11] N. Nikolaev, H.Iba (2006) "Adaptive learning of polynomial networks: genetic programming, backpropagation and Bayesian methods", *p. 153*

[12] Mathworks, 2016. [ONLINE] http://se.mathworks.com/help/nnet/ug/multiple-sequences-with-dynamic-neural-networks.html

Appendix 1 – STAV7 Available Data

Effective gap heat transfer coefficient
Pellet-clad contact heat transfer coefficient
Pellet-clad gap gas heat transfer coefficient
Amount of argon
Clad radial creep
Clad specific heat
Clad radial elastic deformation
Cold radial gap size
Pellet-clad contact pressure
Clad radial plastic deformation
Cold rod internal pressure
Crud layer thickness
Clad radial thermal expansion
Average pellet density fractional change
Nodal burnup
Rod average burnup
Effective plastic strain
Effective total strain
Plastic strain in radial direction
Plastic strain in circumferential direction
Plastic strain in axial direction
Total strain in radial direction
Total strain in circumferential direction
Total strain in axial direction
Fast flux factor
Pellet specific heat
Pellet enthalpy
Hot radial gap size
Amount of hydrogen
Amount of helium
Clad average hydrogen concentration
Amount of krypton
Amount of nitrogen
Clad oxide layer thickness
Coolant pressure
Pellet radial densification
Pellet radial gaseous swelling
Pellet radial relocation

Hot rod internal gas pressure
Max pellet radial relocation
Pellet radial solid swelling
Pellet radial thermal expansion
Nodal linear heat generation rate
Rod average linear power density
Nodal fission gas release fraction
Total relative fission gas release fraction
Rod total elongation
Rod irradiation axial growth
Cladding effective stress
Clad stress in radial direction
Clad stress in circumferential direction
Clad stress in axial direction
Yield stress
Pellet average temperature
Pellet centerline temperature
Temperature at clad inner surface
Temperature at clad outer surface
Temperature at crud outer surface
Time
Temperature at pellet outer surface
Cold total void volume
Hot total gas gap volume
Hot total hole volume for hollow pellets
Hot total void volume
Hot total pellet internal volume
Hot plenum volume
Amount of xenon