NEURAL NETWORK CONTROL OF AUTOCLAVE CURING OF COMPOSITE MATERIALS

Thesis Submitted to
The School of Engineering
UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for
The Degree
Master of Science in Chemical Engineering

by
Maria Claudia Baptista

UNIVERSITY OF DAYTON
Dayton, Ohio
December, 1994
ABSTRACT

NEURAL NETWORK CONTROL OF AUTOCLAVE CURING OF COMPOSITE MATERIALS

Baptista, Maria Claudia
University of Dayton, 1994

Advisor: Dr. C. William Lee

A back propagation neural network has been developed to determine the temperature cure cycle of a fiber reinforced epoxy matrix composite material in an autoclave. The self-directed neural network controller performed temperature control by adjusting the set-point of the autoclave based on five sensor inputs. The curing process was simulated by a one-dimensional heat transfer model that included heat source and convective boundary conditions. These two programs, the simulator and the controller, were installed on two separate computers. The neural network controller was developed using NeuroWindows™ in the Visual Basic™ environment.

The neural network controller used for testing consists of an input layer with five neurons that represent the process and material states, one hidden layer with six neurons, and an output layer with a single neuron for temperature set point adjustment. The neural network controller, when trained by a cure cycle for a given thickness panel, was able to
NEURAL NETWORK CONTROL OF AUTOCLAVE CURING OF COMPOSITE MATERIALS

Approved by:

C. William Lee, Ph.D.
Professor
Committee Chairperson

Tony E. Saliba, Ph.D.
Associate Professor
Committee Member

Kevin J. Myers, D.Sc., P.E.
Associate Professor
Committee Member

Donald L. Moon, Ph.D.
Associate Dean
Graduate Engineering Programs & Research
School of Engineering

Joseph Lestingi, D.Eng., P.E.
Dean
School of Engineering
cure the same thickness panel. This was verified by two cases. When a network was trained by a cycle for a thin panel, it was unable to satisfactorily control the cure of a thick panel, and vice versa. When the two training sets are combined to train a network, the resulting controller was unable to interpolate to control the cure of a panel of an intermediate thickness.

It is surmised that the back propagation neural network is capable of capturing an operator's decision mechanism on temperature control of a heat transfer process as it was able to carry out a cure cycle having been given only randomized instances of the process state. The five process variables chosen as the input vector can only represent a particular case of heat transfer process and are not enough for the network to generalize.
ACKNOWLEDGMENTS

I wish to acknowledge and express my appreciation to the members of the committee, Dr. Kevin J. Myers and Dr. Tony E. Saliba. My gratitude goes also to Dr. Ronald A. Servais and Dr. Sarwan Sandhu for their encouragement. In addition, special thanks are due to my advisor, Dr. C. William Lee for his precious guidance and patience during this work.

I thank Fulbright Commission, Encyclopedia Britannica, American Chamber of Commerce in Sao Paulo, University of Dayton Department of Chemical and Materials Engineering, and Rhodia S.A. for the financial support provided during my master’s program at the University of Dayton.

I also would like to thank my family and my many friends for their help and encouragement. And finally to my grandparents, Adelia and Benedicto Gomes, to whom this work is dedicated.
Artificial Intelligence has stimulated the interest of researchers in various fields of knowledge. Some are very interested in finding the analogy with the physiology of physical systems, in understanding how human brain processes information, how learning occurs, where the information is stored and how memory works. There are so many questions surrounding the field and very few answers.

The development of simplified mathematical models also interested engineers looking for new tools to solve practical problems. The interest in applying Artificial Intelligence to solve Chemical Engineering problems has increased substantially lately. Some problems in the Chemical Process Industries cannot be solved by standard mathematical techniques or their solutions can be complicated, costly, and time consuming. In order to solve such problems, alternative methods have been developed. One of the proposed methods is Artificial Neural Networks. It can be applied in problems where the knowledge about the process is available but there is no mathematical model to represent it. An application of neural networks to process control is presented in this work.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>v</td>
</tr>
<tr>
<td>PREFACE</td>
<td>vi</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF ILLUSTRATIONS</td>
<td>ix</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>xi</td>
</tr>
<tr>
<td>LIST OF SYMBOLS/ABBREVIATIONS</td>
<td>xii</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. LITERATURE REVIEW</td>
<td>3</td>
</tr>
<tr>
<td>Autoclave Curing of Composite Materials</td>
<td>3</td>
</tr>
<tr>
<td>Artificial Neural Networks</td>
<td>7</td>
</tr>
<tr>
<td>Back Propagation Neural Network Learning Algorithm</td>
<td>11</td>
</tr>
<tr>
<td>Applications of Neural Networks in Chemical Engineering Processes</td>
<td>14</td>
</tr>
<tr>
<td>III. NEURAL NETWORK CONTROLLER DEVELOPMENT</td>
<td>16</td>
</tr>
<tr>
<td>Neural Network Controller</td>
<td>16</td>
</tr>
<tr>
<td>Autoclave Process Simulator</td>
<td>20</td>
</tr>
<tr>
<td>Process Monitoring</td>
<td>21</td>
</tr>
</tbody>
</table>
LIST OF ILLUSTRATIONS

2.1. Schematic of the Prepreg Lay-up ................................................ 4
2.2. Conventional, Rule-Based/Expert System, and Neural Network Controller 7
2.3. A Neuron-Like Structure .......................................................... 9
2.4. Sigmoidal Activation Function .................................................... 10
2.5. Back Propagation Neural Network Architecture .......................... 11
3.1. NNC Graphical User Interface .................................................... 19
3.2 Lay-Up and Temperature Sensor Locations .................................... 22
3.3. Closed-Loop Self-Directed Neural Network Control System .......... 24
4.1. Neural Network Controller Development Procedure ...................... 26
4.2. Case 1 Training Using Cure Cycle for 32-Ply Laminate ................. 31
4.3. Case 1 RMS Error Variation During Training ............................... 33
4.4. Back Propagation Neural Network Architecture Used .................... 35
4.5. Weight Variations During Training of Case 1 NNC ......................... 36
4.6. Weight values for Case 1 NNC after 2,000 Epochs ......................... 37
4.7. The Stronger Excitatory and Inhibitory Links of Case 1 NNC .......... 38
4.8. Test 1a, Case 1 Trained Network Controlling 32-Ply Laminate .......... 39
4.9. Test 1b, Case 1 Trained Network Controlling 128-Ply Laminate ....... 41
4.10. Test 1c, Case 1 Trained Network Controlling 256-Ply Laminate ................. 42
4.11. Case 2 Training Using Cure Cycle for 256-Ply Laminate .......................... 44
4.12. Case 2 RMS Error Variation During Training ............................................. 45
4.13. Test 2a, Case 2 Trained Network Controlling 256-Ply Laminate .................. 46
4.14. Test 2b, Case 2 Trained Network Controlling 32-Ply Laminate .................... 47
4.15. Test 2c, Case 2 Trained Network Controlling 128-Ply Laminate .................. 48
4.16. Case 3 RMS Error Variation During Training ............................................. 50
4.17. Test 3a, Case 3 Trained Network Controlling 32-Ply Laminate .................... 51
4.18. Test 3b, Case 3 Trained Network Controlling 128-Ply Laminate .................. 52
4.19. Test 3c, Case 3 Trained Network Controlling 256-Ply Laminate .................. 53
LIST OF TABLES

3.1. State Variables ...................................................................................................... 23

4.1. Ranges Used to Scale the Input and Output Data .................................................. 32

4.2. Number of Links as a Function of the Number of Hidden Neurons .................. 34

5.1. Comparison of RMS Error Obtained After 10,000 Epochs .............................. 55

5.2. Summary of the NNC Test Results ........................................................................ 57
LIST OF SYMBOLS/ABBREVIATIONS

C_p \quad \text{composite heat capacity, } J \text{ kg}^{-1} \text{ K}^{-1}

\Delta H \quad \text{resin heat of reaction, } J \text{ kg}^{-1}

E_p \quad \text{RMS error between neural network calculated output and desired output}

F \quad \text{activation function}

h \quad \text{heat transfer coefficient, } J \text{ m}^{-2} \text{ K}^{-1} \text{ s}^{-1}

J \quad \text{error function, sum of RMS errors}

k \quad \text{composite thermal conductivity, } J \text{ m}^{-1} \text{ K}^{-1} \text{ s}^{-1}

r \quad \text{reaction rate, } \text{s}^{-1}

S \quad \text{weighted sum of neuron inputs}

T \quad \text{local temperature, } \text{K}

t \quad \text{time, } \text{s}

w \quad \text{resin mass fraction, dimensionless}

w_i \quad \text{weighting factor for input to a neuron}

x \quad \text{position, } \text{m}

x_i \quad \text{inputs to a neuron}

y \quad \text{neuron network output}

y_c \quad \text{neural network calculated output}
\[ y_d \] desired output

\[ \Delta_p w_{ji} \] change in the weight from node i to node j at pattern p

\[ \rho \] composite density, \ kg m\(^3\)

\[ \eta \] gain, or learning rate

\[ \alpha \] momentum

\[ \delta_{pj} \] error signal, distance from the activation level of node j to its desired level
CHAPTER I
INTRODUCTION

The importance of composite materials has increased substantially in various segments of industry. In the last decade the production of composites grew 40 percent from 0.87 to 1.22 million tons in the U.S. [44]. Advanced composite materials are basically a combination of a fiber phase and a matrix phase. They can be used in a wide variety of applications. There are applications in the aircraft industry, in commercial areas as diverse as sports and recreational equipment, as well as in automotive and construction industries. With such diverse applications, it is necessary to produce parts with different shapes and properties. As a consequence, the manufacturing process used for a given application has to be adapted in order to obtain a final product with the required properties.

In an epoxy resin (matrix) / graphite (fiber) composite, the resin is cured (crosslinked) through heating, typically carried out in an autoclave or a press. During the autoclave curing process the temperature and the pressure are controlled in order to achieve the desired degree of cure of the resin and proper compaction of the part. Conventional controllers use a pre-specified temperature cure cycle. The cure cycle may be given by the material manufacturer, obtained from empirical methods, or from
analytical models. Self-directed control can be obtained with the use of expert systems, which replaces the pre-programmed cycle and performs on-line control of the process based on changes in process and material states. Such self-directed control can accommodate variability in material and processing since the temperature set point of the autoclave is determined on-line. Current expert system control decisions permit only heating, cooling or holding the autoclave temperature [43]. It is suggested that an artificial neural network may capture the decision-making strategy and provide a continuous output of temperature set point adjustment. Although artificial neural networks have been successfully used for modeling non-linear systems [2, 10, 15, 16, 17, 25, 26, 28, 29, 30, 33], it has not been used for such control purpose.

This work studies the feasibility of using a back propagation neural network to replace the expert system decision making mechanism. The network is trained by an operator's decisions on temperature set point adjustments under various process states such as temperature rates, gradients, etc. The trained network is then used to control a simulated autoclave process.
CHAPTER II
LITERATURE REVIEW

In the search for better ways to control the autoclave curing process, various methods were proposed by different researchers. This chapter summarizes some of these works. Although the application of neural networks to solve industrial problems is recent, there is a great amount of work published in the open literature. A brief review of artificial neural networks and their applications to solve chemical engineering problems is also presented.

Autoclave Curing of Composite Materials

The production of composite materials consists basically of arranging the uncured resin-fiber mixture (prepreg) into the desired shape and then curing the material by applying heat and consolidating it to its final shape by applying pressure. Heating is required to initiate and maintain the curing (crosslinking) reaction which hardens the resin. Pressure application not only removes excess resin and consolidates the part, it also prevents void formation due to volatile components in the resin [1].

For producing a flat composite panel in an autoclave, layers of prepreg are packed together with other materials and sealed in a vacuum bag as shown in Figure 2.1. The
bleeder absorbs excess resin from the prepreg layers during the cure. The breather allows uniform application of vacuum over the lay-up and removal of entrapped air or volatiles produced during the cure. The vacuum bag allows application of vacuum in the bag and also autoclave pressure to the lay-up. The dam is used to prevent resin flow from the edges [24]. The lay-up is then placed in an autoclave for the curing and consolidation process.

![Schematic of the Prepreg Lay-up](image)

Figure 2.1. Schematic of the Prepreg Lay-up
Typically, the only controls one has are the autoclave temperature, autoclave pressure, and the bag vacuum. This work concerns the temperature control only. When the temperature is not controlled properly the reaction may not be complete or thermal runaway may occur [14]. In a conventional controller a pre-specified temperature cure cycle is used. There are several ways to select the temperature cure cycle for a given composite material [1]:

1. Cure cycles are recommended by the material manufacturer. Although it is an easy method to implement, the cycles may not be appropriate for most industrial applications since they are determined based on small samples.

2. Empirical methods are used to determine the optimal cure cycle. This trial-and-error method is not very efficient and leads to material losses and waste of operator and equipment time.

3. Analytical models of the process can be developed and used to simulate the process behavior which allows determination of an adequate cure cycle by trial-and-error on this simulated process. An optimal solution is difficult to find due to simplifications made in the model development.

The use of a pre-specified cure cycle based on time does not allow for variations in raw material properties or variations in processing conditions. To overcome these limitations self-directed process control is used. Self-directed control requires monitoring the cure process constantly and manipulating process conditions to obtain the desired cure state. The cure state is defined by the degree of cure (0 to 100%), a critical parameter in
the process control [39], which influences the glass transition temperature and hence the mechanical properties of the composite.

Whether one uses conventional controller or a self-directed controller, sensors are required to monitor the process state. Thermocouples are typically used for temperature measurement [14, 33]. Dielectric sensors [15] are used for resin viscosity, from which the degree of cure is inferred. Fiber optic sensors not only allow determination of resin viscosity but also temperature, pressure, strain and voids [12, 13]. Thermopile sensors have been used for measurements of heat release rate [11]. There are also sensors for thickness and resin pressure measurements [35].

A conventional controller, as shown in Figure 2.2 (a), compares sensor input with pre-programmed temperature cure cycle and makes set-point adjustments to track the pre-programmed cycle. A self-directed controller replaces the pre-programmed cycle by rules of controlling the process as shown in Figure 2.2 (b). Examples of such self-directed systems are given by Lee et al. [43], Saliba et al. [42], Ciriscioli and Springer [1], Wu and Joseph [8], and Buczek [35]. In general, these systems obtain process state through the use of sensors and decide on control actions (set point adjustment) based on programmed rules. The rules are typically programmed under an expert system shell. The systems developed by Saliba et al. [42] and Wu and Joseph [8] also incorporate mathematical models of the process. This study is an attempt to replace the rule-based controller by an artificial neural network as shown in Figure 2.2 (c).
Figure 2.2. Conventional (a), Rule-Based/Expert System (b), and Neural Network Controller (c)

Artificial Neural Networks

The development of artificial neural networks started with an interest in simulating the human brain. The main neural network paradigms available for practical engineering
applications consist basically of a method for nonlinear mapping between a set of input values and a set of desired output values [20]. Some methods can be used for classification problems where the input vector can be represented by binary values (i.e., 0 or 1's). Other methods can be used for mapping continuous input vectors. The initial step of the development of neural networks is the training phase. The training phase consists of repeatedly presenting a set of input data and the desired results to the network and adjusting internal parameters (weights) to minimize the error between the desired output and the network output. During this training phase the neural network learns the function behavior [16]. Neural network paradigms may require supervised or unsupervised learning. In supervised learning the input vector and the desired output vector are presented during the training procedure. For unsupervised learning the input vectors are presented and the neural network groups them in a way that similar inputs will result in similar results.

The training algorithm can be recursive or non-recursive. The non-recursive, or feed-forward algorithm is easier to implement into a computer program and gives good approximations for static mappings. The recursive, or feedback algorithm uses previous information about the system which provides an interesting tool for simulating the dynamic behavior of systems [10].

Most neural network models present a neuron as a basic processing unit. The mathematical model of a neural network is built from a neuron-like structure. The neuron (node) has input connections that are processed by an activity level (activation function) to produce an output value. Each connection has an associated weight that can be excitatory
(greater than zero) or inhibitory (smaller than zero). A simple neuron-like structure is shown in Figure 2.3.

![Neuron-Like Structure](image)

**Figure 2.3.** A Neuron-Like Structure

Referring to Figure 2.3, \( x_i \) is the input vector; \( w_i \) is the weight vector that represents the strength of the connection; \( S \) is the sum of the weighted inputs to that neuron; and \( y \) is the neuron output calculated by using the activation function \( F \). Mathematically these can be expressed as:

\[
S = w_1x_1 + w_2x_2 + \ldots + w_nx_n = \sum_{i=1}^{n} w_ix_i \quad (2.1)
\]

\[
y = F(S) = \frac{1}{1 + e^{-S}} \quad (2.2)
\]

The activation function shown in Equation 2.2 is the most used function. It is a nonlinear function that compresses the range of \( S \) so that the output, \( y \), lies between zero and one.
Several paradigms were developed using different combinations of the neuron-like structure, alternative activation functions, and other modifications by various authors [4, 21, 23, 25]. Only the back propagation neural network will be discussed as it is used for this study.

The back propagation algorithm is by far the most widely used network for practical applications. Approximately 90% of the reported applications of neural networks use back propagation algorithm. Its simple and straightforward technique for minimizing the error is the main reason for its success.

**Backpropagation Neural Network Learning Algorithm**

Backpropagation neural network, proposed by Rumelhart et al. [3], is constructed from the basic neuron structure shown in Figure 2.3. Layers of neurons are connected as shown in Figure 2.5. The input layer does not perform any calculation (broadcaster layer).
The input neurons are connected by links to the neurons in the hidden layer. More than one hidden layer may exist. Neurons from the last hidden layer are connected to neurons in the output layer. More than one neuron may exist in the output layer. The output of a network (output layer) is a function of the weights of the links and of the network inputs. Bias neurons are added to the input layer and hidden layers. The input to a bias neuron is always one.

![Neural Network Architecture](image)

**Figure 2.5.** Back Propagation Neural Network Architecture

During the training phase the weights are adjusted in order to produce the desired outputs. A training vector consists of a set of inputs (x vector) and desired outputs (y vector), the paired input-output vector is called a training pair. A large number of
training pairs should be available to train the network vectors. The group of training pairs used in the training is called a training set. The main steps of the training are [21]:

1. Select training set.

2. Select initial values of the link weights (random small values).

3. Calculate y from Equations 2.1 and 2.2.

4. Calculate the error between the network output and the desired output.

5. Adjust the link weights of the network in a way that minimizes the error.

6. Repeat steps 3-5 until error reaches an acceptable value.

7. When the error reaches an acceptable value the network is said to be trained. The network performance should then be tested by using a testing set different from the training set.

There are several ways to perform step 5. A brief description of the gradient descent technique, which is the error minimization method used for this study, will be given. In the gradient descent technique the weights are updated to minimize the error function (J),

\[ J(w) = \sum_p E_p = \sum_p \sqrt{(y_c - y_d)^2} \]

(2.5)

In Equation 2.5, the subscript \( p \) represents patterns in the training set; \( y_c \) is the calculated output; and \( y_d \) is the desired output. Until the desired error is obtained or the maximum number of iterations (epochs) is achieved, weights are updated according to

\[ \Delta_p w_{ji} = \eta \delta_{pj} + \alpha \Delta_{p-1} w_{ji} \]

(2.6)
where \( \Delta_p w_{ji} \) is the change in the weight from node i to node j at pattern p; \( \eta \) is a gain term also called the learning rate; \( \alpha \) is a momentum term; the error signal \( \delta_{pj} \) is a measure of the distance from the activation level of node j to its desired level.

The learning rate regulates the rate at which weights are changed during training. If the learning rate is set to 0.5 then the weight change is a function of half of the error. Larger values of learning rate produce larger weight changes and faster training. Momentum is used to determine the proportion of the last weight change into the new weight change. It is used to avoid oscillation of weight changes. Momentum provides a smoothing effect and permits the use of larger training rates.

The error signal, \( \delta_{pj} \), for an output node is calculated following the equation

\[
\delta_{pj} = (d_{pj} - y_{pj}) f'_j(a_{pj}) \tag{2.7}
\]

where \( d_{pj} \) is the desired activation level of node j for input pattern p and \( f'_j(a_{pj}) \) is the derivative of the output function for node j with respect to activity generated by the input pattern p. And, for a neuron in the hidden layer,

\[
\delta_{pj} = f'_j(a_{pj}) \sum_k \delta_{pk} w_{kj} \tag{2.8}
\]

where the summation is over all k nodes to which node j sends output.

After equations 2.7 and 2.8 are differentiated the error signal can be calculated; for the hidden layer(s)

\[
\delta_{pj} = y_{pj}(1 - y_{pj}) \sum_k \delta_{pk} w_{kj} \tag{2.9}
\]

and for the output nodes
\[ \delta_{p_j} = (d_{p_j} - y_{p_j})y_{p_j}(1 - y_{p_j}) \]  

(2.10)

The procedure consists of first propagating the inputs forward through the network to calculate the network output. The calculated output is compared to the desired output to produce an error signal. The error is back propagated through the network to update the weights. This procedure can be used to represent several different non-linear function behaviors.

**Applications of Neural Networks in Chemical Engineering Processes**

The ability to represent non-linear function behavior makes neural network a powerful numerical modeling tool in intelligent process control [16] and in process-fault diagnosis. It can also be used in applications such as fault detection, process control, process design and process simulation. Some examples of its application in chemical engineering are described in this section.

Hoskins and Himmelblau [10] indicate that artificial neural networks are particularly suited for chemical engineering tasks requiring pattern recognition or continuous input-output control in processes with uncertain models and data. They have developed a neural network for diagnosis and fault detection for a system of three continuous stirred tank reactors in series. Good results were obtained within the range studied.

A process model of the autoclave curing of composite materials is described by Joseph and Hanratty [2]. A back propagation neural network was used to represent the
batch manufacturing process. The system has been tested on-line. Their network receives several inputs from the process and predicts the laminate thickness and the maximum void size in the laminate.

An isothermal CSTR reaction neural network model was developed by Bhat et al. [26]. They compared the neural network model prediction to previous model results. The neural network input were the scaled feed composition and reactor space time. The outputs were the dimensionless product concentrations. Historical values were used to train the network. Good accuracy was obtained with the network model.

Neural networks are used in process control when the system to be controlled is non-linear and standard techniques do not give satisfactory results. A classification of the applications of neural network to process control was presented by White and Sofge [19].

The system proposed in this work consists of a supervised control system where the process is monitored constantly and control actions are taken according to the process state.
CHAPTER III
NEURAL NETWORK CONTROLLER AND AUTOCLAVE PROCESS SIMULATOR

The controller developed in this study is a neural network that controls the
temperature of a simulated autoclave. A neural network controller (NNC) consisting of a
general purpose back propagation neural network (BPNN) and data input/output (I/O)
support has been developed. Although this NNC can be used in real-time for monitoring
and control with appropriate I/O modifications, the current version uses a simulation
program to simulate the autoclave and composite temperature responses. These tools,
the NNC and the simulator, are described in this chapter along with consideration of
process variables to be monitored.

Neural Network Controller

A general purpose BPNN capable of handling multiple input/output neurons and
multiple hidden layers forms the core of the NNC. The I/O support includes file I/O for
training data input and file storage of trained link weights during the training phase, and
serial port I/O for simulated sensor input and control set-point output when the NNC is
operating in the controller mode. The complete source code listing of this program is
included in Appendix A.
The BPNN was developed using a neural network tool, NeuroWindows™ (NW), from Ward Systems Group, Inc. NW is a Dynamic Link Library (DLL) for the Windows environment. It is a collection of neural network functions that facilitates the construction and training of various types of neural networks. For I/O support under the Windows™ environment, Visual Basic™ (VB) from Microsoft® Corporation was used as the programming language. VB also facilitates the Graphical User Interface (GUI) under the Windows™ environment.

A brief description of the NNC program is given here. Program procedures that are contained in NW are expressed in upper-case and smallcaps letters (e.g. SMALLCAPS) and those developed under this study are in mixed upper- and lower-case letters (e.g. Uppercase). The subroutines associated with the BPNN are CreateNet, TrainNet and RunNet. Subroutine CreateNet constructs the BPNN using the NW functions MAKE_NET, MAKE_SLAB, and MAKE_LINK. Each slab represents a layer of neurons. MAKE_LINK establishes the links between the slabs. NW functions PUT_SLAB, BPPROPAGATE, BPEVALUATE and BPTRAIN are used in subroutine TrainNet. The training set, which contains several hundred training pairs (input vector and desired output vector), is first read in by subroutine ReadTrainingSet. Each input vector is fed to the network by PUT_SLAB. The input vector then propagates through the slabs by BPPROPAGATE. The output vector from the last slab is compared with the desired output vector by BPEVALUATE. Then the network is trained by back propagation (BPTRAIN) to minimize the errors. These steps are performed on each training pair until the entire training set is
processed. Such an iteration is also called an epoch. Many epochs are required to train the network. Subroutine RunNet is used to predict output after the network has been trained. RunNet uses NW functions PUTSLAB to pass input to the network, BPPROPAGATE to propagate through the slabs, and GETSLAB to pass back the output vector.

The NNC program presents to the user a graphical screen (a VB form) as shown in Figure 3.1. On this form, the user enters the total number of epochs (Epochs) to train the network and the number of neurons in the hidden layers of the network (Hidden). For multiple hidden layers, the number of neurons in each layer is separated by a comma. The number of input and output neurons (Inputs and Outputs) are set by the training data file (the TRN file) and can not be changed at run-time. Their values are echoed upon loading training data by selecting Train on the form. A dialog box appears for the user to enter a training data file name. The training data file structure is commented in subroutine ReadTrainingSet. The training data files used for this study are included in Appendix B.

The NNC program allows randomization of the sequence of the training pairs after each epoch by selecting Randomize on the form. The link weights are stored to a file (the LNK file) at an interval specified by the user. This parameter also determines how often the chart display of the network error is updated. Chart displays, for record-keeping purposes, can be printed by double-clicking on the graph.

Selecting Load Links allows the user to retrieve previous trained network link weights in an LNK file. In this case, the value of Epochs on the form specifies which set of link weights in the LNK file is to be loaded. After either Train or Load Links the network
can be exercised by selecting Run - File or Run - Sim. Run - File reads input data from a file, runs the network, and writes the output data to another file. It serves as a general-purpose neural network. For this study it is mainly used to test the network programming by using data generated by a known function. Run - Sim reads input data from a serial port, runs the network, and sends the output data to the serial port. Run - File and Run - Sim differ in I/O support and that Run - Sim is specialized for the simulator.

**Autoclave Process Simulator**

The autoclave process simulator consists of a computer program developed by Lee [18], which is based on the “CURE” program developed by Loos and Springer [38]. The thermochemical model, which describes the part behavior, solves the one dimensional heat transfer equation. Heat source (exothermic reaction) and convective boundary conditions are considered. The model is represented by the partial differential equation:

\[
\rho C_p \frac{\partial T}{\partial t} = - \frac{\partial}{\partial x} \left( -k \frac{\partial T}{\partial x} \right) + \rho r w \Delta H
\]

where \( \rho \) is the composite density (kg m\(^{-3}\)); \( C_p \) is the composite heat capacity (J kg\(^{-1}\) K\(^{-1}\)); \( T \) is the local temperature (K); \( t \) is time (s); \( x \) is position (m); \( k \) is the composite thermal conductivity (J m\(^{-1}\) K\(^{-1}\) s\(^{-1}\)); \( r \) is the reaction rate (s\(^{-1}\)); \( w \) is the resin mass fraction; and \( \Delta H \) is the resin heat of reaction (J kg\(^{-1}\)). Equation 3.1 describes the heat transfer in the part during the curing reaction. The convective boundary condition for autoclave curing is
\[-k \frac{dT}{dx}_{\text{at surface}} = h(T - T_a)\]  

(3.2)

where \(h\) is the heat transfer coefficient (J m\(^{-2}\) K\(^{-1}\) s\(^{-1}\)) and \(T_a\) is the autoclave temperature (K). The simulator was used for generating the training set and for testing the neural network controller. When used for generating training set, the simulator accepts autoclave temperature set-point adjustment from the keyboard and reports laminate temperature at user specified nodes in graphical display. When used with the neural network controller, the data I/O is through the serial port.

**Process Monitoring**

During the curing process the temperature in the laminate and inside the autoclave were monitored. In addition to monitoring the autoclave temperature (Taut), temperatures at four positions within the laminate were also monitored as shown in Figure 3.2. The laminate sensors were located at laminate centerline (Tmid), laminate top (Ttop), and two locations on either side of the centerline at one node distance away from the midpoint (Tm+, Tm-). Data provided by these five sensors were used to obtain the derived data used for control purpose.

These five sensor inputs were chosen because pertinent process states described by the energy balances in Equations 3.1 and 3.2 can be derived from them. The part temperature, which is the controlled variable, is represented by Tmid. The temperature difference between the laminate and the autoclave (Tmid – Taut) provides information on heat transfer by convection across the boundary (Equation 3.2). The temperature
difference between the laminate top and midpoint (\(T_{\text{top}} - T_{\text{mid}}\)) represents the temperature gradient in the part, which is the driving force for heat transfer by conduction in the part. The temperature rate (\(dT/dt\)) represents the dynamics of the system. The last term in Equation 3.1 represents the heat release rate due to the exothermic reaction. The rate is in general dependent on both temperature and the degree of cure of the resin. A quantity proportional to the heat release rate can be obtained from \(dT/dt\) and 
\((T_{\text{m}+} - 2T_{\text{mid}} + T_{\text{m}-})\) according to method described by Lee and Rice [11]. These five derived data, summarized in Table 3.1, were used as the input vector to the neural network controller.
Table 3.1

State Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Analogy in the model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tmid</td>
<td>Actual laminate tempera</td>
<td>Heat transfer by conduction in the laminate</td>
</tr>
<tr>
<td>Tmid-Taut</td>
<td>Temperature gradient</td>
<td>Heat transfer by convection in the autoclave</td>
</tr>
<tr>
<td>Tmid-Ttop</td>
<td>Laminate temperature</td>
<td>Heat transfer by conduction in the laminate and insulation</td>
</tr>
<tr>
<td></td>
<td>gradient</td>
<td></td>
</tr>
<tr>
<td>dT/dt</td>
<td>Temperature rate</td>
<td>Dynamics of the system</td>
</tr>
<tr>
<td>Hr</td>
<td>Heat release rate</td>
<td>Reaction exotherm, source of heat, function of the degree of cure (history)</td>
</tr>
</tbody>
</table>

In summary, the curing process was simulated by a one-dimensional heat transfer model that included heat source and convective boundary conditions. The neural network controller derived the process state from temperature data at five locations to arrive at a control decision in terms of autoclave set-point adjustment. The set-point adjustment was then sent to the simulated autoclave, which set the autoclave temperature for the next time step of the simulator program execution. Figure 3.3 illustrates this closed-loop self-directed control system.
Figure 3.3. Closed-Loop Self-Directed Neural Network Control System
CHAPTER IV

TRAINING AND TESTING OF THE NEURAL NETWORK CONTROLLER

The neural network controller performance is presented in this chapter. First the neural network was trained and then used to control simulated curing of parts of different thicknesses. The sequence used is presented and the results obtained in each step are discussed.

Neural Network Controller Training

The neural network controller (NNC) was developed using the sequence shown in Figure 4.1. The supervised back propagation neural network (BPNN) learning paradigm was used to perform the network training in this study.

The training set was created using the simulator program and the operator’s knowledge. The operator controlled the simulated autoclave temperature to obtain an acceptable cure cycle. This cure cycle, represented by autoclave set point adjustments, constitutes the desired output of the neural network. Five process variables as described in Chapter III form the input vector to the neural network. These input and desired output values were stored at 1-minute intervals during the simulation run. Upon completion of a successful simulation, a training set was created.
Start

Choose Learning Paradigm

Create Training Set

Define Network Structure

Define Network Parameters

Train Network until epoch < max epoch

Evaluate Network Performance

Connect Network to Simulator

Control OK?

Yes ➔ Test Different Thicknesses

Control OK?

Yes ➔ End

No ➔ Control OK?
The control objectives were to heat the part as fast as possible to its cure temperature of 350°F; to avoid excessive temperature rise (not to exceed 500°F) in the part due to reaction exotherm; and to have the part at the cure temperature (or higher) for a long enough period of time to ensure complete cure. There were also constraints on the autoclave temperature range of 80 to 450°F, heat-up rate limit of 10°F/min, and cool-down rate limit of −10°F/min. The operator's decisions concerning the set point adjustments relied on the autoclave/laminate temperature differences, laminate temperature gradients, and heat released by the reaction. All necessary variables were shown on the computer screen, along with strip chart-like graphical display of temperatures, during the simulation run. Although the numerical display provided useful information, it was the operator's intuitive interpretation of the graphical display that lead to the decision making. The operator's intuition and experience had an important role in the decision making. When monitoring the present and past process state (process dynamics) displayed on the computer screen, future conditions were anticipated and control action decisions made before an unacceptable condition could take place. It is this decision making process, which is difficult to quantify, that was attempted to be captured by the neural network.

After an acceptable training set had been created, the next step was to randomize the data. This was done to avoid memorization by the network. The learning algorithm may produce a set of weights that produce exactly the desired output when it has "memorized" the pattern but may not be able to produce a reasonable output when a
slightly different process state (noise) is presented. After the data had been mapped and randomized the neural network was then trained.

The training set defines the network architecture as far as the number of neurons in the first (input) and the last (output) layer of the network. The network at this point consisted of five input neurons and one output neuron corresponding to the training set created by the operator. At training time, one had the choice of the number of hidden layers and the number of neurons in each hidden layer. As the architecture of the network should be kept as simple as possible, the initial trials started with one hidden layer with five neurons. Additional neurons were added to the hidden layer in an attempt to improve the knowledge capture. Up to nine neurons in the hidden layer were used in the training. Also, a case with one additional hidden layer with five neurons in each layer was studied. A bias neuron was added to each layer. All neurons in each layer were connected to neurons in adjacent layers, resulting in a fully connected network. The learning rate was set to 0.7 and the momentum to 0.3 during the training. These values were kept constant for all cases studied.

A file containing training related information was created during the training session. The information saved consists of: links file name, training set file name, network architecture, parameter values, input data ranges, output data range, followed by weight values and root mean square (RMS) error recorded after every 100 epochs of training.

During training, the RMS average error was displayed in the graph form. All cases studied in this work were trained up to 10,000 epochs. Overtraining could occur when
too many iterations were performed. When this happened the network performance as a controller was not acceptable and link weights from a lower number of epochs were used. The weights variation can also be used for the analysis of the network performance.

The RMS average error was based on the desired output set point adjustment. In this study it is not a reliable measure of the network performance since the network is controlling a dynamic process. Rather, one should judge the network performance by the cure cycle developed by the controller. Once the trained BPNN is available it can be used in a straightforward manner to control the simulated autoclave curing process.

Neural Network Controller Testing

The autoclave (simulator) was heated to a maximum temperature of 450°F and cooled to a minimum of 80°F at rates determined by the neural network controller. The control started with the simulator sending the process state to the NNC. After processing the temperature information into the desired input data the NNC computed the set point adjustment and sent it back to the simulator. The same cycle was repeated for 400 steps in time, with each step representing 1 minute.

The NNC was first tested by processing a panel of the same thickness as that used for training. When this was successful, a validation was performed with a panel of different thickness. Otherwise, network link weights from different epoch numbers were tested. When no improvements were observed with all the intermediate trials, a new training set was created and the sequence shown in Figure 4.1 repeated.
Three different prepreg thicknesses were used in this study: 32, 128, and 256 plies. Training sets were created for 32- and 256-ply panels (Cases 1 and 2) and the intermediate 128-ply panel was used to test the NNC performance. The neural network training and testing are presented separately for each case in the following sections.

**Case 1 Training**

The sequence shown in Figure 4.1 was used to develop a NNC based on the simulated 32-ply panel. Input data for the simulator which include laminate thickness and physical properties of the prepreg are presented in Appendix C.

For a thin laminate the temperature gradient was very small. It can be seen from Figure 4.2 that the prepreg top and midpoint temperatures were very close. When developing the training set the heating rate was increased gradually from 1°F/min up to 9°F/min. When the autoclave temperature reached 250°F the heating rate was reduced gradually to 0°F/min. The midpoint temperature was held above 400°F to assure the cure is completed. The autoclave temperature was held constant for about 20 minutes before cooling started. The cooling rate varied form -1°F/min to -10°F/min, when the autoclave reached the minimum set point (80°F) the cooling rate was reduced to 0°F/min. The resulting training set file is presented in Appendix B.

A list of neural network inputs, the corresponding desired output and respective ranges for normalization is shown in Table 4.1. The input data was scaled to the range 0 to 1 and the output data to the range 0.1 to 0.9 as required by the back propagation
Figure 4.2. Case 1 Training Using Cure Cycle for 32-Ply Laminate
training algorithm. The initial ranges were chosen based on the maximum and minimum values from several simulation runs.

Table 4.1
Ranges Used to Scale the Input and Output Data

<table>
<thead>
<tr>
<th>Variable</th>
<th>Initial Range</th>
<th>Mapped to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tmid</td>
<td>80 to 500°F</td>
<td>0 to 1</td>
</tr>
<tr>
<td>Tmid-Taut</td>
<td>-10 to 50°F</td>
<td>0 to 1</td>
</tr>
<tr>
<td>Tmid-Ttop</td>
<td>-100 to 200°F</td>
<td>0 to 1</td>
</tr>
<tr>
<td>dT/dt</td>
<td>-2.5 to 5°F/min</td>
<td>0 to 1</td>
</tr>
<tr>
<td>Hr</td>
<td>0 to 4°F/min</td>
<td>0 to 1</td>
</tr>
<tr>
<td>Tadj</td>
<td>-12 to 12°F/min</td>
<td>0.1 to 0.9</td>
</tr>
</tbody>
</table>

After creating the training set for the 32-ply cure cycle the network was trained with several architectures of different number of neurons in the hidden layer, and in one case, two hidden layers. The RMS error variations for these cases are shown in Figure 4.3 as a function of number of epochs trained. It can be seen from these curves that the errors initially decreased rapidly and then stabilized for all architectures studied. The simplest structure studied had five neurons in the hidden layer. The RMS error after 10,000 epochs for six hidden neurons was slightly lower than the case with five hidden neurons. However, further increase in the number of neurons did not show any improvement. In
Figure 4.3: Case 1 RMS Error Variation During Training
fact, the error was the highest with nine neurons in the hidden layer. The lowest error value was obtained when two hidden layers, with five neurons each, were used. Table 4.2 shows the number of links as a function of the number of hidden neurons. Increasing the

Table 4.2

Number of Links as a Function of the Number of Hidden Neurons

<table>
<thead>
<tr>
<th>Input Layer</th>
<th>Hidden Layer 1</th>
<th>Hidden Layer 2</th>
<th>Output Layer</th>
<th>Total Number of Links</th>
<th>RMS Error After 10,000 Epochs</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5</td>
<td></td>
<td>1</td>
<td>36</td>
<td>0.000661</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td></td>
<td>1</td>
<td>43</td>
<td>0.000651</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td></td>
<td>1</td>
<td>50</td>
<td>0.000718</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td></td>
<td>1</td>
<td>57</td>
<td>0.000864</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td></td>
<td>1</td>
<td>64</td>
<td>0.001160</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>66</td>
<td>0.000509</td>
</tr>
</tbody>
</table>

number of neurons in the hidden layer increases the number of links and consequently the computational time required for training. The number of links created was 36 when five neurons were used in the hidden layer. The additional hidden layer increased the number of links to 66. Even though the two hidden-layer architecture resulted in the lowest RMS error, the increase in computational time did not justify the slight improvement in the RMS error. It was therefore decided that one hidden layer with six neurons would be used for further studies. This structure is shown in Figure 4.4. Weight variations during training for this chosen architecture are shown in Figure 4.5. From the graph it can be seen that the weight variations were in the range between −15 (inhibitory) and 11 (excitatory).
Although training of the neural network was carried out to 10,000 epochs and the RMS error diminished with increased number of epochs, testing of the NNC on the simulated autoclave showed that the best performance was obtained after 2,000 epochs.

![Back Propagation Neural Network Architecture Used](image)

**Figure 4.4.** Back Propagation Neural Network Architecture Used

As was pointed out before, the performance of the NNC can not be judged by the RMS error. This lead to the decision to use the network link weights after 2,000 epochs of training for all subsequent NNC tests. These link weights for the 32-ply case are shown in Figure 4.6. The largest excitatory and inhibitory links are shown in Figure 4.7. It is interesting to see that while the last three inputs received much attention, the first two
Figure 4.5. Weight Variations During Training of Case 1 NNC
Figure 4.6. Weight Values for Case 1 NNC after 2,000 Epochs
especially \((T_{\text{mid}} - T_{\text{top}})\), did not seem to matter much. It is thought that either the first two inputs were unimportant or the network had captured the operator's bias in decision making.

![Diagram](image_url)

**Figure 4.7** The Stronger Excitatory and Inhibitory Links of Case 1 NNC

**Case 1 Testing**

After the BPNN had been trained to control a 32-ply laminate its performance as a controller (Case 1 NNC) was first tested to see if it can reproduce the training cycle (Test 1a). For a 32-ply laminate the NNC was able to reproduce with considerable accuracy the training cycle (Figure 4.2) as can be seen in Figure 4.8. The NNC received the updated process state from the simulator every minute to make a control action.
Figure 4.8. Test 1a, Case 1 Trained Network Controlling 32-Ply Laminate
decision which was then sent back to the simulated autoclave for the next time step. The cycle is repeated for 400 steps (400 simulated minutes). The resulting temperatures were displayed graphically on the computer screen and stored in a file.

Note that the controller was unable to reproduce the temperature hold in the training cycle as the network could not produce a true zero output. However, the final cure cycle obtained was satisfactory and did not lead to any unacceptable condition.

A thicker prepreg with 128-plies was tested (Test 1b) with the same NNC. The results are shown in Figure 4.9. The autoclave temperature increased without slowing down to 450°F, remained constant (as it was the maximum permitted autoclave temperature) for about 30 minutes and then cooled down to 80°F. The cure cycle obtained was not satisfactory for it caused the laminate temperature to exceed 500°F ($T_{\text{mid}}>500°F$) which was an unacceptable condition.

The controller was then tested (Test 1c) with a 256-ply laminate. The resulting autoclave and laminate temperatures are shown in Figure 4.10. The performance of the Case 1 NNC was even worse than it was in Test 1b. In this test, the laminate temperature exceeded 600°F because of the greater thickness.

**Case 2 Training**

As the network trained by cure cycle for a 32-ply laminate did not lead to good generalization when tested with laminates of different thicknesses, another training set was created using a thicker panel of 256 plies (Case 2). The training set is shown in Figure 4.11. During the initial heating the laminate top temperature was higher than that
Figure 4.9. Test 1b, Case 1 Trained Network Controlling 128-Ply Laminate
Figure 4.10. Test 1c, Case 1 Trained Network Controlling 256-Ply Laminate
at the midpoint. Cooling started when the midpoint temperature became higher than that
of the top (Tmid-Ttop). The RMS errors resulting from various number of hidden neurons
are shown in Figure 4.12. They are similar to those of Case 1. A BPNN with one hidden
layer containing six neurons was selected for the NNC testing.

Case 2 Testing

Same procedure as in Case 1 was followed for Case 2 testing. Tests 2a, 2b, and 2c
were made to control curing of 256-, 32-, and 128-ply laminates, respectively. Test 2a in
Figure 4.13, shows that the NNC was able to reproduce the training and control the curing
of a 256-ply laminate. The laminate temperature increased slowly but did not exceed
500°F and an acceptable cure cycle was obtained. However, the network was not able to
control curing of the 32-ply nor the 128-ply laminate as shown in Figures 4.14 and 4.15.
Figure 4.14 shows that, when controlling the curing of a 32-ply laminate, the temperature
increased so slowly that the temperature reached only about 120°F after 280 minutes.
When controlling the curing of a 128-ply laminate the NNC started cooling before the
temperature reached 300°F as can be seen in Figure 4.15.

Case 3 Training

The network was then trained by combining the training sets used in Cases 1 and
2. The RMS errors are shown in Figure 4.16. One hidden layer with six neurons was
chosen, as were in Cases 1 and 2, to demonstrate the NNC performance in this case even
though the network with seven hidden neurons had the lowest error.
Figure 4.11. Case 2: Training Using Cure Cycle for 256-Ply Laminate
Figure 4.12. Case 2 RMS Error Variation During Training
Figure 4.13. Test 2a, Case 2 Trained Network Controlling 256-Ply Laminate
Figure 4.14. Test 2b, Case2 Trained Network Controlling 32-ply Laminate
Figure 4.15. Test 2c, Case 2 Trained Network Controlling 128-Ply Laminate
Case 3 Testing

Again, the same test procedure was followed. Tests 3a, 3b, and 3c were made to control the curing of 32-, 128-, and 256-ply laminates, respectively. The NNC successfully controlled the curing of a 32-ply panel in Test 3a, shown in Figure 4.17. The laminate temperature never exceeded 500°F.

Then, an intermediate thickness of 128 plies was used to test the network performance in Test 3b. As can be seen in Figure 4.18, the temperature increased slowly, taking a long time to cure the panel. Although the laminate temperature never exceeded 500°F and the resulting cure cycle would cure the part, the controller behavior was not acceptable as it raised the autoclave temperature while the reaction was rapidly accelerating.

Finally, the same network was used to control the curing of a 256-ply panel in Test 3c. It can be seen Figure 4.19, that the laminate temperature went slightly over 500°F. However, the controller performance was considered acceptable. This cure cycle is similar to that in Test 2a, but with generally higher temperatures. The inclusion of the 32-ply training set appeared to have degraded the performance of the NNC.
Figure 4.16. Case 3 RMS Error Variation During Training
Figure 4.17. Test 3a, Case 3 Trained Network Controlling 32-Ply Laminate
Figure 4.18. Test 3b, Case 3 Trained Network Controlling 128-Ply Laminate
Figure 4.19. Test 3c, Case 3 Trained Network Controlling 256-Ply Laminate
CHAPTER V
SUMMARY AND CONCLUSIONS

A general purpose back propagation neural network (BPNN) has been developed. The network has been trained by operator generated cure cycles to function as a neural network controller (NNC) for self-directed control of a simulated autoclave process of curing of composite material. The NNC obtains temperature data from the simulator, performs computation, and sends a set point adjustment to the simulator for the next execution step. NeuroWindows™, a Windows™ Dynamic Link Library, was used as a tool for building the network under the Visual Basic Development Environment.

NNCs with various number of hidden neurons were studied. Three NNCs with one hidden layer containing six neurons were tested by controlling a simulated autoclave. They were respectively trained by using operator generated cure cycle for a thin 32-ply panel, a thick 256-ply panel, and the combination of those two cure cycles.

The effect of the number of neurons in the hidden layer was interpreted by examining the RMS errors which were calculated based on the desired output values given in the training set and the network predicted output. The comparisons were based on the RMS errors after 10,000 epochs of training. In all cases, the network structure with two hidden layers containing five neurons in each layer resulted in the smallest RMS error. A
summary of the RMS errors is shown in Table 5.1. Although the error comparisons were made at 10,000 training epochs, the NNCs used for testing used link weights after 2,000 training epochs as initial tests indicated overtraining occurred with 10,000 epochs.

Table 5.1
Comparison of RMS Error Obtained After 10,000 Epochs

<table>
<thead>
<tr>
<th>Number of Neurons in Hidden Layer(s)</th>
<th>Case 1 Trained by 32-Ply Cycle</th>
<th>Case 2 Trained by 256-Ply Cycle</th>
<th>Case 3 Trained by 32- &amp; 256-Ply Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.000661</td>
<td>0.000552</td>
<td>0.000864</td>
</tr>
<tr>
<td>6</td>
<td>0.000651</td>
<td>0.000388</td>
<td>0.000616</td>
</tr>
<tr>
<td>7</td>
<td>0.000718</td>
<td>0.000392</td>
<td>0.000602</td>
</tr>
<tr>
<td>8</td>
<td>0.000864</td>
<td>0.000571</td>
<td>0.000796</td>
</tr>
<tr>
<td>9</td>
<td>0.001160</td>
<td>0.000440</td>
<td>0.000660</td>
</tr>
<tr>
<td>5, 5 (2 layers)</td>
<td>0.000509</td>
<td>0.000167</td>
<td>0.000310</td>
</tr>
</tbody>
</table>

All test cases are summarized in Table 5.2. From the cases studied it can be seen that the NNC performance was good when the same thickness panel used for the training process was used in the testing. When using the neural network for extrapolation, as in Tests 1b and 1c using Case 1 NNC and in Tests 2b and 2c using Case 2 NNC, the results were not promising. When two training sets were combined to train the Case 3 NNC, it reproduced the training in Tests 3a and 3c but failed to interpolate in Test 3b. Failing to
extrapolate was expected. Failing to interpolate indicated that the network was unable to generalize the control strategy. Although it was surmised that the input vector to the NNC contained all necessary information concerning energy balance of the autoclave curing process, and that the desired output of set point adjustments was based on the operator's understanding of energy balance, there was obviously something missing. One possible cause is the lack of dynamic information to the NNC which was crucial to the operator's decision making. Although one of the inputs, the heat release rate, provides rate information, it carries no history information. Incorporation of temperature rate(s) to the input vector is recommended for future study. One other possible cause is the consistency of the operator's decisions. As there are many paths to achieve the desired outcome, different and conflicting input vectors may be associated with the same set point adjustment decision. This is exemplified by the results of Test 3b. Further examination of the training sets is warranted.

In conclusion, it is believed that a back propagation neural network is capable of capturing an operator's decision mechanism on temperature control of a heat transfer process as it was able to generate a cure cycle having been given only randomized instances of the process state. The five process variables chosen as the input vector can only represent a particular case of heat transfer process and are not enough for the network to generalize. The BPNN tool is flexible enough to accommodate future trials with changes in input vector and hidden layers.
### Table 5.2

**Summary of the NNC Test Results**

<table>
<thead>
<tr>
<th>NNC Case</th>
<th>Number of Pies for Training</th>
<th>Test</th>
<th>Number of Pies for Testing</th>
<th>Performance Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>32</td>
<td>la</td>
<td>32</td>
<td>Over training observed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Good results with 6 neurons in one hidden layer and 2,000 epochs training (Figure 4.8)</td>
</tr>
<tr>
<td>1</td>
<td>32</td>
<td>lb</td>
<td>128</td>
<td>Controlled the process but laminate temperature reached 580°F (Figure 4.9)</td>
</tr>
<tr>
<td>1</td>
<td>32</td>
<td>lc</td>
<td>256</td>
<td>Controlled the process but laminate temperature reached 600°F (Figure 4.10)</td>
</tr>
<tr>
<td>2</td>
<td>256</td>
<td>2a</td>
<td>256</td>
<td>Reproduced temperature cure cycle with good accuracy (Figure 4.13)</td>
</tr>
<tr>
<td>2</td>
<td>256</td>
<td>2b</td>
<td>32</td>
<td>Heating was too slow, NNC did not control the process (Figure 4.14)</td>
</tr>
<tr>
<td>2</td>
<td>256</td>
<td>2c</td>
<td>128</td>
<td>Heating was too slow, NNC did not control the process (Figure 4.15)</td>
</tr>
<tr>
<td>3</td>
<td>32 and 256</td>
<td>3a</td>
<td>32</td>
<td>Over training observed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Good results obtained after 8,000 epochs Temperature increased smoothly and control was obtained (Figure 4.17)</td>
</tr>
<tr>
<td>3</td>
<td>32 and 256</td>
<td>3b</td>
<td>128</td>
<td>Temperature changed slowly, taking very long time to cure Unacceptable controller behavior, called for heating while reaction accelerating (Figure 4.18)</td>
</tr>
<tr>
<td>3</td>
<td>32 and 256</td>
<td>3c</td>
<td>256</td>
<td>Part temperature slightly higher than 500°F but good control obtained (Figure 4.19)</td>
</tr>
</tbody>
</table>
REFERENCES


APPENDICES
APPENDIX A

Source Code Listing of Neural Network Controller (NNC)
'This version supports a single net, with any number of layers

'Declarations

Option Explicit

DefInt I-N
DefSng A-H, O-Z

'General def
Const BLACK = &H0&
Const RED = &HFF&
Const GREEN = &HFF00&
Const YELLOW = &HFFFF&
Const BLUE = &HFF0000
Const MAGENTA = &HFF00FF
Const CYAN = &HFFFF00
Const WHITE = &HFFFF00
Const GRAY = &H80000008

Dim t$
DimCrLfS
Dim dummy

'Array Sizes
Const MaxNeuronsPerLayer = 10
Const MaxPatterns = 1000
Const MaxLayers = 10
Const MaxWeights = 200

' total = SumOf [ nNeurons(i)+1) * nNeurons(i+1) ]
' for i = 0 to nLayers-2

'NN def
Const MomentumBP = 2
Const rLearningRate = .7
Const rMomentum = .3

Const Wtlnitial = .3
Const ScaleInLo = 0!
Const ScaleInHi = 1!
Const ScaleOutLo = .1
Const ScaleOutHi = .9

Dim NetNumber
Dim iCode

Dim nLayers, nHiddenLayers
Dim LastLayer, LastHiddenLayer
Dim nNeurons(0 To MaxLayers)
Dim Weights(MaxWeights)

Dim nInputs, nOutputs

'BackPropogation with momentum

'Momentum determines the proportion of the last change
' added into the new weight change to avoid oscillating

'Initial weights

'Scaling ranges for net input and output

'Hanging this shared forces single net for any NeuroWin call
Dim rMinInput(MaxNeuronsPerLayer), rMaxInput(MaxNeuronsPerLayer)
Dim rMinOutput(MaxNeuronsPerLayer), rMaxOutput(MaxNeuronsPerLayer)
Dim InputName$(MaxNeuronsPerLayer), OutputName$(MaxNeuronsPerLayer)

Dim nEpochCount As Long, nEpochCountDefault As Long, LastEpochCount As Long
Dim nSkip, nSkipDefault
Dim ErrorBuffer(MaxPatterns)
Dim LinksFileName$

' flags
Dim HaveCreatedNet As Integer ' Flag used to limit to a single net at any time
Dim GotLinksFile As Integer
Dim HaveTrainedNet As Integer
Dim TrainMore As Integer

Function Between (x, xlo, xhi)
    If x < xlo Then
        Between = xlo
    ElseIf x > xhi Then
        Between = xhi
    Else
        Between = x
    End If
End Function

Static Sub btnLNK_Click (Value As Integer)
Dim LinksCount, b$

    chkRandomIndex.Visible = False
txtLNK.ForeColor = MAGENTA
DisableButtons

ReadLinksFile

If GotLinksFile Then
    txtLNK.ForeColor = RED
    b$ = txtLNK.Text
    LinksCount = LinksCount + 1
    txtLNK.Text = Parm$(b$, "s") + "s" + Str$(LinksCount)
Else
    txtLNK.ForeColor = BLACK
End If

EnableButtons
chkRandomIndex.Visible = True

End Sub

Sub btnRunTestFile_Click (Value As Integer)
    If (Not GotLinksFile) And (Not HaveCreatedNet) Then
        btnLNK.Value = True  ' ReadLinksFile
        If Not GotLinksFile Then Exit Sub
    End If

    If Not HaveCreatedNet Then
        CreateNet
        lblNetNumber.Caption = "Net" + Str$(NetNumber)
        PutLinkWeights
    End If

    DisableButtons
    RunTestFile
    EnableButtons

End Sub

Sub btnSIM_Click (Value As Integer)
    If Not GotLinksFile Then btnLNK.Value = True

    If Not HaveCreatedNet Then
        CreateNet
        lblNetNumber.Caption = "Net" + Str$(NetNumber)
        PutLinkWeights
    End If

    COMM.PortOpen = True

    DisableButtons
    RunSimulator
    EnableButtons

End Sub
Sub btnTRN_Click (Value As Integer)
Dim b$

If (Not HaveTrainedNet) Then
    If GotLinksFile Then
        b$ = "You have loaded links from " + LinksFileName$ + "." + CrLf$
        b$ = b$ + "Do you want to start with this LNK and train more?"
        If MsgBox(b$, 4, "TrainNet") = 6 Then
            TrainMore = 1 ' need to get new TRN and new LNK
            Else
                btnHidden.Value = True
                Exit Sub
        End If
    Else
        ' no links available either
        TrainMore = 0
    End If
Else ' HaveTrainedNet
    b$ = "You have just trained a net." + CrLf$
    b$ = b$ + "Do you want to train more?"
    If MsgBox(b$, 4, "TrainNet") = 6 Then
        TrainMore = 2 ' use the same TRN and LNK
        Else
            btnHidden.Value = True
            Exit Sub
    End If
End If

End Sub
Sub btnViewTRN_Click (Value As Integer)
Dim i, J, nTRN, nPatterns, a$, b$, KeyWord$
Dim Ttop, Tmid, Taut, dTmid, Hr, SPadj
Dim TrainingSetFileName$
ReDim x(MaxPatterns), Y(MaxPatterns)

OpenFile "Training Set File", "trn", "Input", TrainingSetFileName$, nTRN
lblTRN.Caption = TrainingSetFileName$
If TrainingSetFileName$ = "" Then Exit Sub

frmSIM.Show
frmSIM.Caption = "Training set data"
frmSIM.grpTemp.DrawStyle = 1
frmSIM.grpTemp.GraphStyle = 4
frmSIM.grpTemp.PatternedLines = 1
frmSIM.grpTemp.DataReset = 9
frmSIM.grpTemp.AutoInc = False
frmSIM.grpTemp.RandomData = False
frmSIM.grpTemp.NumPoints = 1000
frmSIM.grpTemp.TickEvery = 200
frmSIM.grpTemp.LabelEvery = 200
frmSIM.grpTemp.NumSets = 5

Do
dummy = DoEvents()
If EOF(nTRN) Then Exit Do
KeyWord$ = InputField$("[\]", nTRN)
Select Case KeyWord$
Case "INPUT RANGES"
    i = 0
    Do
        Line Input #nTRN, a$
        If Trim$(a$) = "" Then Exit Do
        i = i + 1
        InputName$(i) = Trim$(Parm$(a$, t$))
        rMinInput(i) = Val(Parm$(a$, t$))
        rMaxInput(i) = Val(a$)
    Loop
    nInputs = i
    txtNinputs.Text = nInputs
    nNeurons(0) = nInputs

Case "OUTPUT RANGES"
    i = 0
    Do
        Line Input #nTRN, a$
        If Trim$(a$) = "" Then Exit Do
        i = i + 1
        OutputName$(i) = Trim$(Parm$(a$, t$))
        rMinOutput(i) = Val(Parm$(a$, t$))
        rMaxOutput(i) = Val(a$)
    Loop

End Do
nOutputs = i:    txtNoutputs.Text = nOutputs

' Hidden Layers from screen input
a$ = txtNhidden.Text
b$ = ""
For i = 1 To MaxLayers - 1
    nNeurons(i) = Val(Parm$(a$, " "))
    b$ = b$ + Str$(nNeurons(i)) + "," 
    If a$ = "" Then Exit For
Next:    txtNhidden.Text = Left$(Trim$(b$), Len(b$) - 2)

nHiddenLayers = i:

nLayers = nHiddenLayers + 2

LastHiddenLayer = i

LastLayer = i + 1

nNeurons(LastLayer) = nOutputs

Case "TRAINING DATA"    ' (nInputs + nOutputs) pieces of data
    J = 0
    Do
        dummy = DoEvents()
        If EOF(nTRN) Then Exit Do
        Line Input #nTRN, a$
        "Text.Text = Str$(J) + " + a$
        If Trim$(a$) <> "" Then
            J = J + 1
            For i = 1 To nInputs
                x(i) = Val(Parm$(a$, t$))
            Next
            For i = 1 To nOutputs
                Y(i) = Val(Parm$(a$, t$))
            Next
            Tmid = x(1)
            Ttop = x(1) - x(2)
            Taut = x(1) - x(3)
            "dTmid = X(4)
            Hr = x(5)
            SPadj = Y(1)
            frmSIM.grpTemp.ThisPoint = J
            frmSIM.grpTemp.ThisSet = 1
            frmSIM.grpTemp.GraphData = Tmid
            frmSIM.grpTemp.ThisSet = 2
            frmSIM.grpTemp.GraphData = Ttop
            frmSIM.grpTemp.ThisSet = 3
            frmSIM.grpTemp.GraphData = Taut
            frmSIM.grpTemp.ThisSet = 4
            frmSIM.grpTemp.GraphData = Hr * 10
            frmSIM.grpTemp.ThisSet = 5
            frmSIM.grpTemp.GraphData = SPadj * 10
        End If
    Loop
nPatterns = J:       txtNpatterns.Text = Str$(nPatterns)
Exit Do
End Select
Loop
Close #nTRN

frmSIM.grpTemp.GraphTitle = TrainingSetFileName$
frmSIM.grpTemp.NumPoints = (nPatterns \ 50 + 1) * 50
frmSIM.grpTemp.TickEvery = frmSIM.grpTemp.NumPoints / 5
frmSIM.grpTemp.LabelEvery = frmSIM.grpTemp.TickEvery
frmSIM.grpTemp.DrawMode = 2

btnHidden.Value = True

End Sub

Sub cmdQUIT_Click ()
iCode = KillNet(NetNumber)
Close
End
End Sub

Sub cmdReset_Click ()

Close
iCode = KillNet(NetNumber)
NetNumber = -1
HaveCreatedNet = False
GotLinksFile = False

nEpochCount = nEpochCountDefault
txtEpochCount = Trim$(Str$(nEpochCount))
nSkip = nSkipDefault
txtNrecords = Trim$(Str$(nSkip))

grpErrors.DataReset = 9
grpErrors.NumPoints = 2
grpErrors.GraphData = 0
grpErrors.BottomTitle = "Epochs/ " + Str$(nSkip)
grpErrors.Refresh
grpErrors.DrawMode = 3

grpLinks.DataReset = 9
grpLinks.NumPoints = 2
grpLinks.GraphData = 0
Sub CreateNet()
' Creates a Momentum BP NN of any number of layers [nLayers]
' with any number of neurons in each single-slab layer [nNeurons()].
' nLayers and nNeurons() must be specified and shared
' NetNumber is established by the Sub

Dim n

If HaveCreatedNet Then ' Current version allows a single net
    Beep
    MsgBox "Net already exist. A new net is created.", 16, "CREATE NET"
End If

' Define the network number
iCode = GetNextNet(NetNumber): If iCode Then GoTo ErrorCreateNet

' Make BackPropagation with momentum and enter the serial number
iCode = MakeNet(NetNumber, MomentumBP, 645213508): If iCode Then GoTo ErrorCreateNet

' Build nLayer slabs in the network, (each layer has a single slab)
iCode = MakeSlab(NetNumber, 0, nNeurons(0), InputLayer%): If iCode Then GoTo ErrorCreateNet
For n = 1 To LastHiddenLayer
    iCode = MakeSlab(NetNumber, n, nNeurons(n), HiddenLayer%): If iCode Then GoTo ErrorCreateNet
Next
iCode = MakeSlab(NetNumber, LastLayer, nNeurons(LastLayer), OutputLayer%): If iCode Then GoTo ErrorCreateNet

' Link slabs together (allocates memory to contain the weights)
' Creates link n between slabs n and n+1
' Initializes the weights to between +/- WtInitial
For n = 0 To LastHiddenLayer
    iCode = MakeLink(NetNumber, n, WtInitial, n, n + 1): If iCode Then GoTo ErrorCreateNet
Next

HaveCreatedNet = True
lblNetNumber.Caption = "Net#" + Trim$(Str$(NetNumber))
Text.Text = Text.Text + "Net#" + Trim$(Str$(NetNumber))
Exit Sub
ErrorCreateNet:
    MsgBox NetError$(iCode), 16, "Error in CreateNet"

End Sub

Sub DisableButtons ()
    btnTRN.Enabled = False
    btnLNK.Enabled = False
    btnRunTestFile.Enabled = False
    btnSIM.Enabled = False
End Sub

Sub EnableButtons ()
    btnTRN.Enabled = True
    btnLNK.Enabled = True
    btnRunTestFile.Enabled = True
    btnSIM.Enabled = True

    btnHidden.Value = True
End Sub

Sub Form_Load ()
Dim i, a$

t$ = Chr$(9)
CrLf$ = Chr$(13) + Chr$(10)

HaveCreatedNet = False
HaveTrainedNet = False
GotLinksFile = False
NetNumber = -1

nEpochCount = txtEpochCount.Text
nEpochCountDefault = txtEpochCount.Text
nSkip = txtNrecords.Text
nSkipDefault = txtNrecords.Text
grpErrors.BottomTitle = "Epochs/ " + Str$(nSkip)
'Communication definitions
COMM.CommPort = Val(txtCOMMport.Text)
COMM.Settings = "9600,N,8,1"
COMM.InputLen = 0
COMM.OutBufferCount = 0
COMM.InBufferCount = 0

End Sub

Sub GetLinkWeights()
Dim J, k, m, n

'iCode = Getlink(NetNumber, 0, Weights(0)):  If iCode Then GoTo ErrorGetLinkWeights
'For n = 1 To LastHiddenLayer
'  j = (nNeurons(n - 1) + 1) * nNeurons(n)
'  iCode = Getlink(NetNumber, n, Weights(j)):  If iCode Then GoTo ErrorGetLinkWeights
'Next

m = 0
For n = 0 To LastLayer - 1       ' links 0, 1, ...
    For J = 0 To nNeurons(n)
        For k = 1 To nNeurons(n + 1)
            iCode = GetWeight(NetNumber, n, J, k, Weights(m)):
                If iCode Then GoTo ErrorGetLinkWeights
            m = m + 1
        Next
    Next
Next

Exit Sub

ErrorGetLinkWeights:
    MsgBox NetError$(iCode), 16, "Error in GetLinkWeights"

End Sub

Function GreaterOf (a, b)
   If a > b Then
       GreaterOf = a
   Else
       GreaterOf = b
   End If
End Function
Sub grpErrors_DblClick()
    grpErrors.PrintStyle = 1
    grpErrors.DrawMode = 5
End Sub

Sub grpLinks_DblClick()
    grpLinks.PrintStyle = 1
    grpLinks.DrawMode = 5
End Sub

Function InputField$(Delimiters$, nFileHandle)
' read from nFileHandle
' Skips lines to specified Delimiters$
' Returns the enclosed text in upper case
Dim L$, R$, a$

    L$ = Left$(Delimiters$, 1)
    R$ = Right$(Delimiters$, 1)

    Do
        Line Input #nFileHandle, a$
        a$ = LTrim$(a$)
        If Left$(a$, 1) = L$ Then
            InputField$ = UCase$(Mid$(a$, 2, InStr(a$, R$) - 2))
            Exit Do
        End If
    Loop

End Function

Sub lblEpochs_Click()
    txtEpochCount.Text = 10000
    nEpochCount = 10000
End Sub
Function Map! (ByVal Value!, min!, max!, x1!, x2!)
' Map variables to the range 0 - 1

If min! = max! Then
    MsgBox "Min and max are equal in Map: " + Str$(min!) + ", " + Str$(max!), 16, ""
    Map! = 0!
End
End If
If Value! < min! Then Value! = min!
If Value! > max! Then Value! = max!
Map! = (x2! - x1!) * (Value! - min!) / (max! - min!) + x1!
End Function

Function NewName$ (GivenName$, NewExt$)
' Change file extension of GivenName$ to NewExt$ion

Dim i

i = InStr(GivenName$, ".")
If i <> 0 Then
    NewName$ = Left$(GivenName$, i) + "," + NewExt$
Else
    NewName$ = GivenName$
End If
End Function

Function nTotalWeights (nL, nn())
Dim i, n

n = 0
For i = 0 To nL - 2
    n = n + (nn(i) + 1) * nn(i + 1)
Next
nTotalWeights = n
End Function

Sub OpenFile (Descript$, ext$, Action$, TheFile$, nFileHandle)
' Get file title from GivenName and construct NewFileName$ with Ext$ 
' Action is "A" for Append, "O" for Output, else for Input 
' Assumes CMDialog object with the name diaFiles

Dim Act$ 

On Error GoTo ErrorOpenFile
diaFiles.DialogTitle = "Select " + Descript$ + " name for " + Action$
diaFiles.Filter = "All Files(*.*)|*.*
" + "|" + ext$ + "|"
daFiles.FilterIndex = 2
Act$ = UCase$(Left$(Action$, 1))
If Act$ = "O" Or Act$ = "A" Then
daFiles.Action = 2
Else
    diaFiles.Action = 1
End If
TheFile$ = diaFiles.Filename
daFiles.Filename = ""
nFileHandle = FreeFile
If Act$ = "O" Then
    Open TheFile$ For Output As nFileHandle
ElseIf Act$ = "A" Then
    Open TheFile$ For Append As nFileHandle
Else
    Open TheFile$ For Input As nFileHandle
End If
dummy = DoEvents()

Exit Sub

ErrorOpenFile:
    If Not diaFiles.CancelError Then
        MsgBox "Error in OpenFile", 16, ""
    End If
    Resume QuitOpenFile

QuitOpenFile:
    TheFile$ = ""

End Sub

Function Parm$ (b$, Delimiter$)
' Extract string to the left of the Delimiter$ as Parm$
' Input string b$ is truncated

Dim i

i = InStr(b$, Delimiter$)
If i <> 0 Then
    Parm$ = Left$(b$, i - 1)
b$ = Right$(b$, Len(b$) - i)
b$ = Trim$(b$)
Else
Sub PrintLinksFileHeader (nLNK, nPatterns, LNKdescription$)
ReDim NName$(MaxLayers, MaxNeuronsPerLayer)
Dim i, j, n

Print #nLNK, "[This File] ": Print #nLNK, LinksFileName$ + ", " + LNKdescription$
Print #nLNK,
Print #nLNK, "[Training Set] ": Print #nLNK, lbITRNCaption
Print #nLNK,
Print #nLNK, "[Patterns]": Print #nLNK, nPatterns
Print #nLNK,
Print #nLNK, "[Learning Rate]": Print #nLNK, rLearningRate
Print #nLNK,
Print #nLNK, "[Momentum]": Print #nLNK, rMomentum
Print #nLNK,
Print #nLNK, "[Layers]": Print #nLNK, nLayers
Print #nLNK,
Print #nLNK, "[Neurons]"
  For n = 0 To LastLayer
    Print #nLNK, nNeurons(n); t$;
  Next: Print #nLNK,
Print #nLNK,
Print #nLNK, "[Input Ranges]"
  For i = 1 To nInputs
    Print #nLNK, InputName$(i); t$; rMinInput(i); t$; rMaxInput(i)
  Next
Print #nLNK,
Print #nLNK, "[Output Ranges]"
  For i = 1 To nOutputs
    Print #nLNK, OutputName$(i); t$; rMinOutput(i); t$; rMaxOutput(i)
  Next
Print #nLNK,

Print #nLNK, "[Weights]"
  Print #nLNK, " "; t$; ' header 1
  For i = 0 To nTotalWeights(nLayers, nNeurons()) - 1
    Print #nLNK, i; t$;
  Next: Print #nLNK,
Print #nLNK, " "; t$; ' header 2, weights identifier
  For n = 0 To LastLayer
    NName$(n, 0) = "Bias" + Trim$(Str$(n))
    For J = 1 To nNeurons(n)
      If n = 0 Then

C:\NN\NNC.FRM

78

NNname$(n, J) = InputName$(J)
ElseIf n = LastLayer Then
  NNname$(n, J) = OutputName$(J)
Else
  NNname$(n, J) = "H" + Trim$(Str$(n)) + "." + Trim$(Str$(J))
End If
Next
Next
For n = 0 To LastHiddenLayer
  For i = 0 To nNeurons(n)
    For J = 1 To nNeurons(n + 1)
      Print #nLNK, NNname$(n, i) + "," + NNname$(n + 1, J) + ":
    Next
  Next
Next 'layer
Print #nLNK,

End Sub

Sub PutLinkWeights()
  Dim J, k, m, n

  ' iCode = PutLink(NetNumber, 0, Weights(0)):  If iCode Then GoTo ErrorPutLinkWeights
  ' For n = 1 To LastHiddenLayer
  '    j = (nNeurons(n - 1) + 1) * nNeurons(n)
  '    iCode = PutLink(NetNumber, n, Weights(j)):  If iCode Then GoTo ErrorPutLinkWeights
  ' Next

  m = 0
  For n = 0 To LastLayer - 1
    ' links 0, 1, ...
    For J = 0 To nNeurons(n)
      For k = 1 To nNeurons(n + 1)
        iCode = PutWeight(NetNumber, n, J, k, Weights(m)):  If iCode Then GoTo ErrorPutLinkWeights
      Next
    Next
  Next 'layer
  Print #nLNK,

End Sub

ErrorPutLinkWeights:
  m = m + 1
Next
Next

Exit Sub

ErrorPutLinkWeights:
  MsgBox NetError$(iCode), 16, "Error in PutLinkWeights"

End Sub
Sub RandomIndex (Limit, IndexArray())
Dim i, MaxIndex, Index, iHold

Randomize Timer

For i = 1 To Limit
    MaxIndex = Limit - i + 1
    Index = Int(Rnd * (MaxIndex)) + 1
    "'(1 <= rnd num <= MaxIndex)
    iHold = IndexArray(Index)
    IndexArray(Index) = IndexArray(MaxIndex)
    IndexArray(MaxIndex) = iHold
Next

End Sub

Sub ReadLinksFile () ' Read and view links
Dim nEp As Long, nEp1 As Long
Dim i, n, nLNKin, nWts, nCount, AvgErr
Dim a$, b$, KeyWord$, LNKdescription$

GotLinksFile = False

' Open and Read Links File
If TrainMore = 2 Then
    nLNKin = FreeFile
    b$ = lblLNK.Caption
    Open Parm$(b$, ",") For Input As #nLNKin
Else
    OpenFile "Links File", "Ink", "Input", LinksFileName$, nLNKin ' All parameters are tab delimited
    If LinksFileName$ = "" Then Exit Sub
    lblLNK.Caption = LinksFileName$
End If

Do
    If EOF(nLNKin) Then Exit Do
dummy = DoEvents()
    KeyWord$ = InputField$("["], nLNKin)

Select Case KeyWord$
Case "THIS FILE" ' 1 line, file name of training set
    Line Input #nLNKin, LNKdescription$
    lblLNK.Caption = LNKdescription$
Case "TRAINING SET" ' 1 line, file name of training set
    Line Input #nLNKin, a$
    Text.Text = "Trained by " + a$
End Select
Case "PATTERNS"  ' 1 line, 1 INTEGER
    Line Input #nLNKin, a$
    txtNpatterns.Text = Val(a$)

Case "LEARNING RATE"  ' 1 line, 1 SINGLE
    Line Input #nLNKin, a$
    '  ignored

Case "MOMENTUM"  ' 1 line, 1 SINGLE
    Line Input #nLNKin, a$
    '  ignored

Case "LAYERS"  ' 1 line, 1 INTEGER
    Line Input #nLNKin, a$
    nLayers = Val(a$)
    nHiddenLayers = nLayers - 2
    LastLayer = nLayers - 1
    LastHiddenLayer = LastLayer - 1

Case "NEURONS"  ' 1 line, nLayers INTEGERs
    Line Input #nLNKin, a$
    For n = 0 To LastLayer  '  number of neurons for each layer
        nNeurons(n) = Val(Parm$(a$, t$))
    Next
    nInputs = nNeurons(0)
    txtNinputs.Text = Str$(nInputs)
    b$ = ""
    For i = 1 To nHiddenLayers
        b$ = b$ + Str$(nNeurons(i)) + ","
    Next
    txtNhidden.Text = Left$(Trim$(b$), Len(b$) - 2)
    nOutputs = nNeurons(LastLayer)
    txtNoutputs.Text = Str$(nOutputs)

    nWts = nTotalWeights(nLayers, nNeurons())

Case "INPUT RANGES"  ' nInputs lines
    For i = 1 To nInputs  '  InputNeuronName, lo, hi
        Line Input #nLNKin, a$
        InputName$(i) = Trim$(Parm$(a$, t$))
        rMinInput(i) = Val(Parm$(a$, t$))
        rMaxInput(i) = Val(Parm$(a$, t$))
    Next

Case "OUTPUT RANGES"  ' nOutputs lines
    For i = 1 To nOutputs  '  OutputNeuronName, lo, hi
        Line Input #nLNKin, a$
        OutputName$(i) = Trim$(Parm$(a$, t$))
        rMinOutput(i) = Val(Parm$(a$, t$))
        rMaxOutput(i) = Val(Parm$(a$, t$))
    Next
Case "WEIGHTS"
  ' Reads to EOF, ignoring blank lines
  ' header 1
  ' header 2
  ' Input Vector... , Output Vector...

  nCount = 0
  Do
    dummy = DoEvents()
    If EOF(nLNKin) Then Exit Do
    Line Input #nLNKin, a$
    If Trim$(a$) <> "" Then
      nEp = Val(Parm$(a$, t$)):
      nCount = nCount + 1
      If nCount = 2 Then
        nEp1 = nEp
      ElseIf nCount = 3 Then
        nSkip = nEp - nEp1:
    End If
    b$ = "nEpochs: " + Str$(nEp) + ": "
    For i = 0 To nWts - 1
      Weights(i) = Val(Parm$(a$, t$))
      b$ = b$ + Str$(Weights(i)) + " / "
    Next:
    If chkDisplay.Value Then Text.Text = b$

    AvgErr = Val(Parm$(a$, t$)):
    ErrorBuffer(nCount) = AvgErr
    End If
  Loop While nEp < nEpochCount
  nEpochCount = nEp:
  LastEpochCount = nEp
  Exit Do
  End Select
  Loop

Close #nLNKin

......................... View AvgErr Data
  grpErrors.DataReset = 9
  ...
  grpErrors.NumPoints = (nCount \ 25 + 1) * 25
  grpErrors.BottomTitle = "Epochs /" + Str$(nSkip)
  grpErrors.GraphTitle = "AvgErr (" + LinksFileNameS + ")"
  grpErrors.TickEvery = grpErrors.NumPoints / 5
  grpErrors.LabelEvery = grpErrors.TickEvery
  For i = 1 To nCount
    grpErrors.GraphData = ErrorBuffer(i)
  Next
  grpLinks.Refresh
  grpErrors.DrawMode = 3
View Links Data

grpLinks.DataReset = 9
grpLinks.NumPoints = nWts
b$ = "Last Loaded Weights after" + Str$(nEpochCount) + " Epochs: 
For i = 0 To nWts - 1
    grpLinks.GraphData = Weights(i)
    b$ = b$ + Str$(Weights(i)) + "/"
Next
Text.Text = Text.Text +CrLfS + b$
grpLinks.GraphTitle = "Weights (" + LinksFileNameS + ") after" + Str$(nEpochCount) + " epochs"
grpLinks.TickEvery = nNeurons(1)
grpLinks.LabelEvery = grpLinks.TickEvery
grpLinks.Refresh
grpLinks.DrawMode = 3

GotLinksFile = True
HaveTrainedNet = False ' Weights have been redefined

End Sub

Sub ReadTrainingSet(nTRN, nPatterns, Xmapped(), Ymapped())
Dim i, J, x, Y, a$, b$, KeyWord$}

Do
dummy = DoEvents()
If EOF(nTRN) Then Exit Do
KeyWord$ = InputField$("["], nTRN)
Select Case KeyWord$
Case "INPUT RANGES"
    i = 0 ' end this group with a blank line
    ' (number of nInputs determined by
    ' the number of lines that follow)
    Do
        Line Input #nTRN, a$
        If Trim$(a$) = "" Then Exit Do ' InputNeuronName, lo, hi
        i = i + 1
        InputName$(i) = Trim$(Parm$(a$, t$))
        rMinInput(i) = Val(Parm$(a$, t$))
        rMaxInput(i) = Val(a$)
    Loop
    nInputs = i: txtNinputs.Text = nInputs
    nNeurons(0) = nInputs

Case "OUTPUT RANGES" ' nOutputs lines
    i = 0 ' end this group with a blank line
    Do
        Line Input #nTRN, a$
        If Trim$(a$) = "" Then Exit Do ' OutputNeuronName, lo, hi
        i = i + 1
        OutputName$(i) = Trim$(Parm$(a$, t$))
    Loop

End Sub
rMinOutput(i) = Val(Parm$(a$, t$))
 rMaxOutput(i) = Val(a$)
Loop
 nOutputs = i: txtNoutputs.Text = nOutputs

' Hidden Layers from screen input
 a$ = txtNhidden.Text
 b$ = ""
For i = 1 To MaxLayers - 1
 nNeurons(i) = Val(Parm$(a$, ",,"))
 b$ = b$ + Str$(nNeurons(i)) + ","
If a$ = "" Then Exit For
Next: txtNhidden.Text = Left$(Trim$(b$), Len(b$) - 2)
 nHiddenLayers = i:
 nLayers = nHiddenLayers + 2
 LastHiddenLayer = i
 LastLayer = i + 1
 nNeurons(LastLayer) = nOutputs

Case "TRAINING DATA"
 J = 0
 Do
dummy = DoEvents()
 If EOF(nTRN) Then Exit Do
 Line Input #nTRN, a$
 If Trim$(a$) <> "" Then
  J = J + 1
 For i = 1 To nInputs
   ' Mapping input vector to xMapped(i,j) and yMapped(i,j)
   x = Val(Parm$(a$, t$))
   Xmapped(i, J) = Map(x, rMinInput(i), rMaxInput(i), ScaleInLo, ScaleInHi)
 Next
 For i = 1 To nOutputs
   Y = Val(Parm$(a$, t$))
   Ymapped(i, J) = Map(Y, rMinOutput(i), rMaxOutput(i), ScaleOutLo, ScaleOutHi)
 Next
 End If
 Loop

 nPatterns = J: txtNpatterns.Text = Str$(nPatterns)
 Exit Do
End Select
Loop
Close #nTRN

End Sub
Sub RunNet (VectorIn(), rMinInput(), rMaxInput(), VectorOut(), rMinOutput(), rMaxOutput())
ReDim rNetInput(MaxNeuronsPerLayer), rNetOutput(MaxNeuronsPerLayer)
Dim i, n

For i = 1 To nInputs
    rNetInput(i) = Map(VectorIn(i), rMinInput(i), rMaxInput(i), ScaleInLo, ScaleInHi)
Next

iCode = PutSlab(NetNumber, 0, rNetInput(1)): If iCode GoTo ErrorRunNet

For n = 1 To LastLayer
    iCode = BpPropagate(NetNumber, n): If iCode GoTo ErrorRunNet
Next

iCode = GetSlab(NetNumber, LastLayer, rNetOutput(1)): If iCode GoTo ErrorRunNet

For i = 1 To nOutputs
    VectorOut(i) = Map(rNetOutput(i), ScaleOutLo, ScaleOutHi, rMinOutput(i), rMaxOutput(i))
Next

Exit Sub

ErrorRunNet:
    MsgBox NetError$(iCode), 16, "Error in RunNet"
End Sub

Sub RunSimulator ()
ReDim VectorIn(MaxNeuronsPerLayer), VectorOut(MaxNeuronsPerLayer)
Dim i, iter, a$, b$, nSIM
Dim SimulationFileName$
Dim Tmid, TmidOld, Ttop, HRtp, StPt, StPtOld, StPtAdj, ActualStPtAdj, StPtMin, StPtMax

SimulationFileName$ = NewName$(LinksFileName$, "SIM")
daFiles.Filename = SimulationFileName$
OpenFile "Simulation Run File", "sim", "Output", SimulationFileName$, nSIM
lblSIM.Caption = SimulationFileName$
If SimulationFileName$ = "" Then lblSIM.Caption = "Simulation Run Not Saved"

Print #nSIM, "Links from: "; LinksFileName$; " After "; nEpochCount; " Epochs"
Print #nSIM, ""
Print #nSIM, "time": t$; "Tmid": t$; "Ttop": t$; "HRtp": t$; "Taut"
Print #nSIM, "0": t$; "80": t$; "80": t$; "0": t$; "80"

' Get NN input state vector from Simulator on COMM port
' RunNet and send resulting SetPointAdjustment to Simulator
' initialize
iter = 0
StPtOld = 80
StPtAdj = 0
StPtMin = 80
StPtMax = 450

' Reset SIM
COMM.Output = "99" + Chr$(13)

frmSIM.Show
frmSIM.Caption = "Simulated Autoclave"
frmSIM.grpTemp.GraphTitle = "T, " + LinksFileName$ + ", after" + Str$(nEpochCount) + " epochs"
frmSIM.grpTemp.NumSets = 2
frmSIM.grpTemp.AutoInc = False

' Do cycles
iter = 0
Do While iter < 400
    iter = iter + 1
    dummy = DoEvents()

    ' Get SIM data from COMM port
    a$ = ""
    Do
        a$ = a$ + COMM.Input
        dummy = DoEvents()
    Loop Until InStr(a$, Chr$(13))
    lblGeneral.Caption = a$
    b$ = "Data From Simulator:"
    For i = 1 To nlnputs
        VectorIn(i) = Val(Parm$(a$, ","))
        b$ = b$ + Chr$(13) + InputName$(i) + ": " + Format$(VectorIn(i), "###0.00")
    Next
    Tmid = VectorIn(1)
    Ttop = VectorIn(1) - VectorIn(2)
    HRtp = VectorIn(5)

    ' Run net and send resulting set point adjustment to SIM
    RunNet VectorIn(), rMinInput(), rMaxInput(), VectorOut(), rMinOutput(), rMaxOutput()
    StPtAdj = VectorOut(1)
    COMM.Output = Str$(StPtAdj) + Chr$(13)

    ' SIM will clip StPt
    StPt = Between((StPtOld + StPtAdj), StPtMin, StPtMax)
    ActualStPtAdj = StPt - StPtOld
Sub RunTestFile()
ReDim ExpectedOut(MaxNeuronsPerLayer), dev(MaxNeuronsPerLayer), devR(MaxNeuronsPerLayer)
ReDim Emin(MaxNeuronsPerLayer), Emax(MaxNeuronsPerLayer), Esum(MaxNeuronsPerLayer)
ReDim VectorIn(MaxNeuronsPerLayer), VectorOut(MaxNeuronsPerLayer)
Dim i, nTST, nTSO, nCount
Dim a$, b0$, b1$, b2$, b3$, b4$, TestFileName$, TestOutputFileName$

    OpenFile "Test Data File", "tst", "Input", TestFileName$, TestOutputFileName$, nTST
If TestFileName$ = "" Then Exit Sub

TestOutputFileName$ = NewName$(LinksFileName$, "TSO")
daFiles.FileName = TestOutputFileName$
OpenFile "Test Output File", "tso", "Output", TestOutputFileName$, nTSO
If TestOutputFileName$ = "" Then Exit Sub

lblMiscFile.Caption = "Write to: " + TestOutputFileName$
Print #nTSO, "Trained by: "; lblTRN.Caption
Print #nTSO, "Links from: "; LinksFileName$; " After "; nEpochCount; " Epochs"
Print #nTSO, "The weights are:"
For i = 0 To nTotalWeights(nLayers, nNeurons()) - 1
    Print #nTSO, Weights(i); t$
Next: Print #nTSO,
Print #nTSO,
Print #nTSO, "The results are:"
For i = 1 To nOutputs
    Print #nTSO, "Expected("; Trim$(Str$(i)); ")"; t$
    Print #nTSO, "NNcalc'd("; Trim$(Str$(i)); ")"; t$
    Print #nTSO, "RelErr("; Trim$(Str$(i)); ")"; t$
Next: Print #nTSO,

For i = 1 To nOutputs
    Emax(i) = -1E+08!
    Emin(i) = 1E+08!
    Esum(i) = 0!
Next

nCount = 0
Do Until EOF(nTST)
    nCount = nCount + 1
dummy = DoEvents()
    Line Input #nTST, a$
For i = 1 To nInputs
    VectorIn(i) = Val(Parm$(a$, t$))
Next
For i = 1 To nOutputs
    ExpectedOut(i) = Val(Parm$(a$, t$))
Next
RunNet VectorIn(), rMinInput(), rMaxInput(), VectorOut(), rMinOutput(), rMaxOutput()
For i = 1 To nOutputs
    dev(i) = (VectorOut(i) - ExpectedOut(i))
    devR(i) = dev(i) / ExpectedOut(i)
    If devR(i) > Emax(i) Then Emax(i) = devR(i)
    If devR(i) < Emin(i) Then Emin(i) = devR(i)
    Esum(i) = Esum(i) + dev(i) ^ 2
    Print #nTSO, ExpectedOut(i); t$; VectorOut(i); t$; devR(i); t$
Next
Print #nTSO,
Loop
Print #nTSO,
b0$ = "Number of test vectors: " + Str$(nCount)
Text.Text = b0$
Print #nTSO, b0$
For i = 1 To nOutputs
    b1$ = "For Output" + Str$(i) + ":"
    b2$ = "MinDev = " + Format$(Emin(i), "00.00%") + ""
    b3$ = "MaxDev = " + Format$(Emax(i), "00.00%") + ""
    b4$ = "E^2/N = " + Format$(Esum(i) ^ 2 / nCount, "0.000E+00")
    Print #nTSO, b1$ + b2$ + b3$ + b4$
End For
Text.Text = b0$
Close #nTST
Close #nTSO

btnRunTestFile.Value = False
MsgBox "RunTestFile Completed.", 64, ""
End Sub

Function SmallerOf (a As Single, b As Single)
    If a < b Then
        SmallerOf = a
    Else
        SmallerOf = b
    End If
End Function

Static Sub TrainNet ()
    ReDim Index(MaxPatterns)
    ReDim Xmapped(MaxNeuronsPerLayer, MaxPatterns), Ymapped(MaxNeuronsPerLayer, MaxPatterns)
    ReDim rNetInput2(MaxNeuronsPerLayer, MaxPatterns), rNetOutput2(MaxNeuronsPerLayer, MaxPatterns)
    Dim nLoop As Long, LastSave As Long
    Dim b$, RNDflag$, TrainingSetFileName$, LNKdescription$
    Dim i, J, n, nTRN, nLNK, nPatterns
    Dim SumError, Emin, Emax, ErrorFactor, AvgError

    If TrainMore = 0 Or TrainMore = 1 Then
        OpenFile "Training Set File", "trn", "Input", TrainingSetFileName$, nTRN
        lblTRN.Caption = TrainingSetFileName$
        If TrainingSetFileName$ = "" Then Exit Sub
    End If

ReadTrainingSet nTRN, nPatterns, Xmapped(), Ymapped()

LNKdescription$ = lblLNK.Caption
OpenFile "Links File", "Ink", "Output", LinksFileName$, nLNK ' See ReadLinksFile
lblLNK.Caption = LinksFileName$
If LinksFileName$ = "" Then Exit Sub

PrintLinksFileHeader nLNK, nPatterns, LNKdescription$

CreateNet
txtTRN.Text = "Train Net" + Str$(NetNumber)
If TrainMore = 1 Then
    PutLinkWeights
End If

Else

nTRN = FreeFile
Open lblTRN.Caption For Input As nTRN

ReadTrainingSet nTRN, nPatterns, Xmapped(), Ymapped()

btnLNK.Value = True
nLNK = FreeFile
b$ = lblLNK.Caption
Open Parm$(b$, ",") For Append As nLNK

End If

If TrainMore = 0 Then
    nLoop = 0
    grpErrors.DataReset = 9
    grpErrors.GraphData = 0
Else
    nLoop = LastEpochCount
    If nLoop <= nEpochCount Then
        nEpochCount = nEpochCount * 2
        txtEpochCount.Text = Trim$(str$(nEpochCount))
    End If
    grpErrors.DataReset = 9
    grpErrors.NumPoints = ((LastEpochCount / nSkip) \ 25 + 1) * 25
    grpErrors.GraphData = ErrorBuffer(1)
    For i = 1 To LastEpochCount / nSkip
        grpErrors.GraphData = ErrorBuffer(i)
    Next
End If
Training

Do Until nLoop >= nEpochCount  'nEpochCount is defined through txtEpochCount.Text
dummy = DoEvents()

nLoop = nLoop + 1:  txtEpochs.Text = nLoop

SumError = 0!
Emin = 1E+08!
Emax = -1E+08!

For i = 1 To nPatterns  'Generate Index Vector
    Index(i) = I  '1, 2, 3, ... nPatterns
Next i

If chkRandomIndex.Value Then
    RNDflag$ = "Randomized"
    RandomIndex nPatterns, Index()  'Randomize indices in Index()
Else
    RNDflag$ = ""
End If

For J = 1 To nPatterns  'Prep data arrays rNetInput2(i,j)
    For i = 1 To nInputs  'and rNetOutput2(i,j)
        rNetInput2(i, Index(J)) = Xmapped(i, J)
    Next i
    For i = 1 To nOutputs
        rNetOutput2(i, Index(J)) = Ymapped(i, J)
    Next i
Next J

For J = 1 To nPatterns
    dummy = DoEvents()

    'Put input pattern
    iCode = PutSlab(NetNumber, 0, rNetInput2(1, J)):  If iCode Then GoTo ErrorTrainNet

    'Propagate through hidden layers
    For n = 1 To LastLayer
        iCode = BpPropagate(NetNumber, n):  If iCode Then GoTo ErrorTrainNet
    Next

    'Evaluate error
    iCode = BpEvaluate(NetNumber, LastLayer, rNetOutput2(1, J), ErrorFactor)
    If iCode Then GoTo ErrorTrainNet

    'Back Propagate
    For n = LastHiddenLayer To 0 Step -1
        iCode = BpTrain(NetNumber, n, rLearningRate, rMomentum, 0)
        If iCode Then GoTo ErrorTrainNet
    Next
SumError = SumError + ErrorFactor
If ErrorFactor > Emax Then Emax = ErrorFactor
If ErrorFactor < Emin Then Emin = ErrorFactor
Next

AvgError = SumError / nPatterns

If (nLoop Mod nSkip) = 0 Then
   grpErrors.NumPoints = ((nLoop / nSkip) \ 25 + 1) * 25
   grpErrors.TickEvery = grpErrors.NumPoints / 5
   grpErrors.LabelEvery = grpErrors.TickEvery
   grpErrors.ThisPoint = nLoop / nSkip + 1
   grpErrors.GraphData = AvgError
   grpErrors.Refresh
   grpErrors.DrawMode = 3

   ' Get link weights of net in a vector
   GetLinkWeights
   LastSave = nLoop

   ' Store weights in LNK file
   b$ = "nLoop=" + Str$(nLoop) + ": "
   Print #nLNK, nLoop;
   For i = 0 To nTotalWeights(nLayers, nNeurons()) - 1
      Print #nLNK, t$; Weights(i);
      b$ = b$ + Str$(Weights(i)) + " / "
   Next
   If (chkDisplay.Value) Then Text.Text = b$
   Print #nLNK, t$; AvgError; t$; Emin; t$; Emax; t$; RNDflag$

End If

Loop
nEpochCount = LastSave
LastEpochCount = nEpochCount
Close #nLNK

' Display last saved net
grpLinks.DataReset = 9
grpLinks.NumPoints = nTotalWeights(nLayers, nNeurons())
For i = 0 To nTotalWeights(nLayers, nNeurons())
   grpLinks.GraphData = Weights(i)
Next
grpLinks.TickEvery = nNeurons(1)
grpLinks.LabelEvery = grpLinks.TickEvery
grpLinks.Refresh
grpLinks.DrawMode = 3

txtTRN.ForeColor = GRAY
HaveTrainedNet = True
TrainMore = 0

MsgBox "Training is Complete.", 64, ""

Exit Sub

ErrorTrainNet:
    MsgBox NetError$(iCode), 16, "Error in TrainNet"
End Sub

Sub txtCOMMport_DblClick()
    If txtCOMMport.Text = 2 Then
        txtCOMMport.Text = 1
        COMM.CommPort = 1
    Else
        txtCOMMport.Text = 2
        COMM.CommPort = 2
    End If
End Sub

Sub txtEpochCount_DblClick()
    nEpochCount = Val(txtEpochCount.Text)
    txtEpochCount.ForeColor = RED
End Sub

Sub txtEpochCount_LostFocus()
    txtEpochCount.ForeColor = BLACK
End Sub

Sub txtNhidden_Change()
    txtNhidden.ForeColor = &H80000008
End Sub
Sub txtNrecords_Change()
    nSkip = txtNrecords.Text
    grpErrors.BottomTitle = "Epochs / " + txtNrecords.Text
End Sub
APPENDIX B

Thin Panel (32 plies) Training Set
Thick Panel (256 plies) Training Set
Thin Panel (32 plies) Training Set

### Input Ranges

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Lower</th>
<th>Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{mid}$</td>
<td>80</td>
<td>500</td>
</tr>
<tr>
<td>$T_{mid} - T_{top}$</td>
<td>-10</td>
<td>50</td>
</tr>
<tr>
<td>$T_{mid} - T_{aut}$</td>
<td>-100</td>
<td>200</td>
</tr>
<tr>
<td>$dT/dt$</td>
<td>-2.5</td>
<td>5</td>
</tr>
<tr>
<td>$H_r$</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

### Output Ranges

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Lower</th>
<th>Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{adj}$</td>
<td>-12</td>
<td>12</td>
</tr>
</tbody>
</table>

### Training Data

<table>
<thead>
<tr>
<th>$T_{mid}$ (Y2)</th>
<th>$T_{adj}$ (Y2)</th>
<th>$dT/dt$ (Y1)</th>
<th>$H_r$ (Y1)</th>
<th>$T_{adj}$ (Y3)</th>
<th>$T_{adj}$ (Y3)</th>
<th>$T_{adj}$ (Y3)</th>
<th>$T_{adj}$ (Y3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>80.03</td>
<td>-0.03</td>
<td>-0.97</td>
<td>0.03</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>5.96E-02</td>
</tr>
<tr>
<td>80.13</td>
<td>-0.08</td>
<td>-1.87</td>
<td>0.10</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.141795</td>
</tr>
<tr>
<td>80.30</td>
<td>-0.12</td>
<td>-2.70</td>
<td>0.17</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.212571</td>
</tr>
<tr>
<td>80.54</td>
<td>-0.15</td>
<td>-3.46</td>
<td>0.24</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.27692</td>
</tr>
<tr>
<td>80.84</td>
<td>-0.19</td>
<td>-4.16</td>
<td>0.30</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.335851</td>
</tr>
<tr>
<td>81.22</td>
<td>-0.25</td>
<td>-5.78</td>
<td>0.38</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0.44768</td>
</tr>
<tr>
<td>81.73</td>
<td>-0.33</td>
<td>-7.27</td>
<td>0.51</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0.579637</td>
</tr>
<tr>
<td>82.35</td>
<td>-0.40</td>
<td>-8.65</td>
<td>0.63</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0.696206</td>
</tr>
<tr>
<td>83.09</td>
<td>-0.46</td>
<td>-9.91</td>
<td>0.74</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0.802616</td>
</tr>
<tr>
<td>83.94</td>
<td>-0.52</td>
<td>-11.06</td>
<td>0.84</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0.900189</td>
</tr>
<tr>
<td>84.90</td>
<td>-0.60</td>
<td>-13.10</td>
<td>0.96</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1.047527</td>
</tr>
<tr>
<td>86.02</td>
<td>-0.70</td>
<td>-14.98</td>
<td>1.12</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1.212119</td>
</tr>
<tr>
<td>87.29</td>
<td>-0.78</td>
<td>-16.71</td>
<td>1.27</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1.358687</td>
</tr>
<tr>
<td>88.70</td>
<td>-0.86</td>
<td>-18.30</td>
<td>1.41</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1.492678</td>
</tr>
<tr>
<td>90.24</td>
<td>-0.94</td>
<td>-19.76</td>
<td>1.54</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1.615615</td>
</tr>
<tr>
<td>91.93</td>
<td>-1.04</td>
<td>-22.07</td>
<td>1.69</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1.786283</td>
</tr>
<tr>
<td>93.80</td>
<td>-1.14</td>
<td>-24.20</td>
<td>1.87</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1.97235</td>
</tr>
<tr>
<td>95.84</td>
<td>-1.24</td>
<td>-26.16</td>
<td>2.04</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>2.138695</td>
</tr>
<tr>
<td>98.04</td>
<td>-1.33</td>
<td>-27.96</td>
<td>2.20</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>2.290918</td>
</tr>
<tr>
<td>100.41</td>
<td>-1.45</td>
<td>-30.59</td>
<td>2.37</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>2.488431</td>
</tr>
<tr>
<td>102.99</td>
<td>-1.56</td>
<td>-33.01</td>
<td>2.58</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>2.699205</td>
</tr>
<tr>
<td>105.77</td>
<td>-1.68</td>
<td>-35.23</td>
<td>2.78</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>2.888351</td>
</tr>
<tr>
<td>108.73</td>
<td>-1.78</td>
<td>-37.27</td>
<td>2.96</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>3.061662</td>
</tr>
<tr>
<td>111.88</td>
<td>-1.90</td>
<td>-40.12</td>
<td>3.15</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>3.278737</td>
</tr>
<tr>
<td>115.26</td>
<td>-2.03</td>
<td>-42.74</td>
<td>3.38</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>3.507734</td>
</tr>
<tr>
<td>118.85</td>
<td>-2.15</td>
<td>-45.15</td>
<td>3.59</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>3.713952</td>
</tr>
<tr>
<td>122.64</td>
<td>-2.26</td>
<td>-47.36</td>
<td>3.79</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>3.903373</td>
</tr>
<tr>
<td>126.65</td>
<td>-2.40</td>
<td>-50.35</td>
<td>4.00</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>4.1358</td>
</tr>
<tr>
<td>130.89</td>
<td>-2.53</td>
<td>-53.11</td>
<td>4.24</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>4.37965</td>
</tr>
<tr>
<td>135.37</td>
<td>-2.66</td>
<td>-55.63</td>
<td>4.48</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>4.600505</td>
</tr>
<tr>
<td>140.06</td>
<td>-2.77</td>
<td>-57.94</td>
<td>4.69</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>4.804676</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>---</td>
<td>----</td>
</tr>
<tr>
<td>144.97</td>
<td>-2.91</td>
<td>-61.03</td>
<td>4.92</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>5.052374</td>
</tr>
<tr>
<td>150.15</td>
<td>-3.05</td>
<td>-63.85</td>
<td>5.17</td>
<td>2.20E-02</td>
<td>8</td>
<td>0</td>
<td>5.312606</td>
</tr>
<tr>
<td>155.57</td>
<td>-3.18</td>
<td>-66.43</td>
<td>5.43</td>
<td>6.43E-02</td>
<td>8</td>
<td>0</td>
<td>5.551676</td>
</tr>
<tr>
<td>161.24</td>
<td>-3.29</td>
<td>-68.76</td>
<td>5.66</td>
<td>0.117212</td>
<td>8</td>
<td>0</td>
<td>5.776798</td>
</tr>
<tr>
<td>167.15</td>
<td>-3.42</td>
<td>-71.85</td>
<td>5.92</td>
<td>0.156627</td>
<td>9</td>
<td>0</td>
<td>6.0494</td>
</tr>
<tr>
<td>173.36</td>
<td>-3.55</td>
<td>-74.64</td>
<td>6.21</td>
<td>0.238865</td>
<td>9</td>
<td>0</td>
<td>6.340301</td>
</tr>
<tr>
<td>179.86</td>
<td>-3.67</td>
<td>-77.14</td>
<td>6.50</td>
<td>0.353867</td>
<td>9</td>
<td>0</td>
<td>6.618149</td>
</tr>
<tr>
<td>186.66</td>
<td>-3.77</td>
<td>-79.34</td>
<td>6.80</td>
<td>0.504955</td>
<td>9</td>
<td>0</td>
<td>6.893203</td>
</tr>
<tr>
<td>193.53</td>
<td>-3.55</td>
<td>-72.47</td>
<td>6.87</td>
<td>0.949279</td>
<td>10</td>
<td>53.10401</td>
<td>6.651446</td>
</tr>
<tr>
<td>200.28</td>
<td>-3.51</td>
<td>-74.72</td>
<td>6.75</td>
<td>0.980708</td>
<td>9</td>
<td>111.9465</td>
<td>6.711061</td>
</tr>
<tr>
<td>207.30</td>
<td>-3.53</td>
<td>-75.70</td>
<td>7.03</td>
<td>1.25982</td>
<td>8</td>
<td>187.5357</td>
<td>7.05021</td>
</tr>
<tr>
<td>214.66</td>
<td>-3.53</td>
<td>-76.34</td>
<td>7.35</td>
<td>1.662427</td>
<td>8</td>
<td>287.2813</td>
<td>7.347373</td>
</tr>
<tr>
<td>222.38</td>
<td>-3.49</td>
<td>-76.62</td>
<td>7.72</td>
<td>2.180229</td>
<td>8</td>
<td>418.095</td>
<td>7.685456</td>
</tr>
<tr>
<td>230.51</td>
<td>-3.38</td>
<td>-75.49</td>
<td>8.13</td>
<td>2.877628</td>
<td>7</td>
<td>590.7527</td>
<td>8.019688</td>
</tr>
<tr>
<td>239.06</td>
<td>-3.22</td>
<td>-73.94</td>
<td>8.56</td>
<td>3.738132</td>
<td>7</td>
<td>815.0406</td>
<td>8.396213</td>
</tr>
<tr>
<td>248.14</td>
<td>-3.01</td>
<td>-71.86</td>
<td>9.08</td>
<td>4.817247</td>
<td>7</td>
<td>1104.075</td>
<td>8.864664</td>
</tr>
<tr>
<td>257.83</td>
<td>-2.70</td>
<td>-68.18</td>
<td>9.68</td>
<td>6.185645</td>
<td>6</td>
<td>1475.214</td>
<td>9.372884</td>
</tr>
<tr>
<td>268.16</td>
<td>-2.31</td>
<td>-63.84</td>
<td>10.34</td>
<td>7.799403</td>
<td>6</td>
<td>1943.178</td>
<td>9.947742</td>
</tr>
<tr>
<td>279.21</td>
<td>-1.86</td>
<td>-58.79</td>
<td>11.05</td>
<td>9.634493</td>
<td>6</td>
<td>2521.248</td>
<td>10.59763</td>
</tr>
<tr>
<td>290.93</td>
<td>-1.33</td>
<td>-52.07</td>
<td>11.72</td>
<td>11.59281</td>
<td>5</td>
<td>3216.816</td>
<td>11.19088</td>
</tr>
<tr>
<td>303.11</td>
<td>-0.78</td>
<td>-44.89</td>
<td>12.17</td>
<td>13.33299</td>
<td>5</td>
<td>4016.796</td>
<td>11.62213</td>
</tr>
<tr>
<td>315.33</td>
<td>-0.29</td>
<td>-37.67</td>
<td>12.22</td>
<td>14.46556</td>
<td>5</td>
<td>4884.729</td>
<td>11.74073</td>
</tr>
<tr>
<td>327.02</td>
<td>0.08</td>
<td>-29.98</td>
<td>11.69</td>
<td>14.65614</td>
<td>4</td>
<td>5764.098</td>
<td>11.3125</td>
</tr>
<tr>
<td>337.55</td>
<td>0.29</td>
<td>-23.45</td>
<td>10.53</td>
<td>13.74067</td>
<td>4</td>
<td>6588.538</td>
<td>10.31927</td>
</tr>
<tr>
<td>346.83</td>
<td>0.40</td>
<td>-18.17</td>
<td>9.28</td>
<td>12.49955</td>
<td>4</td>
<td>7338.511</td>
<td>9.173539</td>
</tr>
<tr>
<td>356.64</td>
<td>0.91</td>
<td>-11.36</td>
<td>9.81</td>
<td>14.04246</td>
<td>3</td>
<td>8181.059</td>
<td>9.304344</td>
</tr>
<tr>
<td>367.04</td>
<td>1.50</td>
<td>-3.96</td>
<td>10.40</td>
<td>15.97306</td>
<td>3</td>
<td>9139.442</td>
<td>9.807139</td>
</tr>
<tr>
<td>378.01</td>
<td>2.13</td>
<td>4.01</td>
<td>10.97</td>
<td>17.99305</td>
<td>3</td>
<td>10219.02</td>
<td>10.34449</td>
</tr>
<tr>
<td>389.53</td>
<td>2.78</td>
<td>12.53</td>
<td>11.52</td>
<td>20.0476</td>
<td>3</td>
<td>11421.88</td>
<td>10.86158</td>
</tr>
<tr>
<td>401.47</td>
<td>3.44</td>
<td>21.47</td>
<td>11.94</td>
<td>21.97001</td>
<td>3</td>
<td>12740.08</td>
<td>11.28354</td>
</tr>
<tr>
<td>413.57</td>
<td>4.05</td>
<td>30.57</td>
<td>12.10</td>
<td>23.48679</td>
<td>3</td>
<td>14149.29</td>
<td>11.48783</td>
</tr>
<tr>
<td>425.38</td>
<td>4.56</td>
<td>39.38</td>
<td>11.81</td>
<td>24.22524</td>
<td>3</td>
<td>15602.8</td>
<td>11.30615</td>
</tr>
<tr>
<td>436.26</td>
<td>4.88</td>
<td>47.26</td>
<td>10.88</td>
<td>23.79906</td>
<td>3</td>
<td>17030.75</td>
<td>10.56142</td>
</tr>
<tr>
<td>452.52</td>
<td>4.80</td>
<td>57.52</td>
<td>7.02</td>
<td>18.98464</td>
<td>3</td>
<td>19489.66</td>
<td>7.173725</td>
</tr>
<tr>
<td>457.09</td>
<td>4.47</td>
<td>59.09</td>
<td>4.58</td>
<td>15.311</td>
<td>3</td>
<td>20408.32</td>
<td>4.910107</td>
</tr>
<tr>
<td>459.42</td>
<td>4.05</td>
<td>58.42</td>
<td>2.32</td>
<td>11.65052</td>
<td>3</td>
<td>21107.35</td>
<td>2.745095</td>
</tr>
<tr>
<td>459.94</td>
<td>3.61</td>
<td>55.94</td>
<td>0.53</td>
<td>8.491117</td>
<td>3</td>
<td>21616.82</td>
<td>0.968264</td>
</tr>
<tr>
<td>459.22</td>
<td>3.19</td>
<td>52.22</td>
<td>-0.73</td>
<td>6.02085</td>
<td>3</td>
<td>21978.07</td>
<td>-0.30846</td>
</tr>
<tr>
<td>457.71</td>
<td>2.80</td>
<td>47.71</td>
<td>-1.50</td>
<td>4.207082</td>
<td>3</td>
<td>22230.49</td>
<td>-1.11694</td>
</tr>
<tr>
<td>455.80</td>
<td>2.45</td>
<td>42.80</td>
<td>-1.91</td>
<td>2.92238</td>
<td>3</td>
<td>22405.84</td>
<td>-1.55491</td>
</tr>
<tr>
<td>453.75</td>
<td>2.12</td>
<td>37.75</td>
<td>-2.05</td>
<td>2.028176</td>
<td>3</td>
<td>22527.53</td>
<td>-1.72787</td>
</tr>
<tr>
<td>451.73</td>
<td>1.82</td>
<td>32.73</td>
<td>-2.02</td>
<td>1.409278</td>
<td>3</td>
<td>22612.08</td>
<td>-1.72333</td>
</tr>
<tr>
<td>449.84</td>
<td>1.54</td>
<td>27.84</td>
<td>-1.88</td>
<td>0.980321</td>
<td>3</td>
<td>22670.9</td>
<td>-1.60572</td>
</tr>
<tr>
<td>448.14</td>
<td>1.32</td>
<td>24.14</td>
<td>-1.70</td>
<td>0.708764</td>
<td>2</td>
<td>22713.43</td>
<td>-1.47743</td>
</tr>
<tr>
<td>446.61</td>
<td>1.12</td>
<td>20.61</td>
<td>-1.53</td>
<td>0.502471</td>
<td>2</td>
<td>22743.58</td>
<td>-1.33526</td>
</tr>
<tr>
<td>445.24</td>
<td>0.97</td>
<td>18.24</td>
<td>-1.37</td>
<td>0.378731</td>
<td>1</td>
<td>22766.3</td>
<td>-1.2205</td>
</tr>
<tr>
<td>444.00</td>
<td>0.85</td>
<td>16.00</td>
<td>-1.24</td>
<td>0.274114</td>
<td>1</td>
<td>22782.75</td>
<td>-1.11749</td>
</tr>
<tr>
<td>442.89</td>
<td>0.74</td>
<td>13.89</td>
<td>-1.11</td>
<td>0.193702</td>
<td>1</td>
<td>22794.37</td>
<td>-0.99931</td>
</tr>
<tr>
<td>441.88</td>
<td>0.67</td>
<td>12.88</td>
<td>-1.01</td>
<td>0.162308</td>
<td>0</td>
<td>22804.11</td>
<td>-0.93425</td>
</tr>
<tr>
<td>440.93</td>
<td>0.61</td>
<td>11.93</td>
<td>-0.95</td>
<td>0.123208</td>
<td>0</td>
<td>22811.5</td>
<td>-0.89372</td>
</tr>
<tr>
<td>440.04</td>
<td>0.56</td>
<td>11.04</td>
<td>-0.89</td>
<td>8.94E-02</td>
<td>0</td>
<td>22816.87</td>
<td>-0.84453</td>
</tr>
<tr>
<td>439.20</td>
<td>0.52</td>
<td>10.20</td>
<td>-0.84</td>
<td>6.41E-02</td>
<td>0</td>
<td>22820.71</td>
<td>-0.79324</td>
</tr>
<tr>
<td>438.42</td>
<td>0.48</td>
<td>9.42</td>
<td>-0.78</td>
<td>4.51E-02</td>
<td>0</td>
<td>22823.12</td>
<td>-0.74188</td>
</tr>
<tr>
<td>437.69</td>
<td>0.44</td>
<td>8.69</td>
<td>-0.73</td>
<td>3.10E-02</td>
<td>0</td>
<td>22825.28</td>
<td>-0.6916</td>
</tr>
<tr>
<td>437.01</td>
<td>0.41</td>
<td>8.01</td>
<td>-0.68</td>
<td>2.05E-02</td>
<td>0</td>
<td>22826.51</td>
<td>-0.64308</td>
</tr>
<tr>
<td>436.39</td>
<td>0.37</td>
<td>7.39</td>
<td>-0.63</td>
<td>1.27E-02</td>
<td>0</td>
<td>22827.27</td>
<td>-0.59676</td>
</tr>
<tr>
<td>435.80</td>
<td>0.34</td>
<td>6.80</td>
<td>-0.58</td>
<td>6.91E-03</td>
<td>0</td>
<td>22827.68</td>
<td>-0.55286</td>
</tr>
<tr>
<td>435.26</td>
<td>0.32</td>
<td>6.26</td>
<td>-0.54</td>
<td>2.68E-03</td>
<td>0</td>
<td>22827.84</td>
<td>-0.51152</td>
</tr>
<tr>
<td>434.77</td>
<td>0.29</td>
<td>5.77</td>
<td>-0.50</td>
<td>0</td>
<td>0</td>
<td>22827.82</td>
<td>-0.47275</td>
</tr>
<tr>
<td>434.31</td>
<td>0.27</td>
<td>5.31</td>
<td>-0.46</td>
<td>0</td>
<td>0</td>
<td>22827.66</td>
<td>-0.43652</td>
</tr>
<tr>
<td>433.88</td>
<td>0.25</td>
<td>4.88</td>
<td>-0.42</td>
<td>0</td>
<td>0</td>
<td>22827.42</td>
<td>-0.40277</td>
</tr>
<tr>
<td>433.49</td>
<td>0.23</td>
<td>4.49</td>
<td>-0.39</td>
<td>0</td>
<td>0</td>
<td>22827.11</td>
<td>-0.3714</td>
</tr>
<tr>
<td>433.13</td>
<td>0.21</td>
<td>4.13</td>
<td>-0.36</td>
<td>0</td>
<td>0</td>
<td>22826.76</td>
<td>-0.34229</td>
</tr>
<tr>
<td>432.80</td>
<td>0.19</td>
<td>3.80</td>
<td>-0.33</td>
<td>0</td>
<td>0</td>
<td>22826.39</td>
<td>-0.31532</td>
</tr>
<tr>
<td>432.49</td>
<td>0.18</td>
<td>3.49</td>
<td>-0.31</td>
<td>0</td>
<td>0</td>
<td>22826.01</td>
<td>-0.29037</td>
</tr>
<tr>
<td>432.21</td>
<td>0.16</td>
<td>3.21</td>
<td>-0.28</td>
<td>0</td>
<td>0</td>
<td>22825.63</td>
<td>-0.26731</td>
</tr>
<tr>
<td>431.95</td>
<td>0.15</td>
<td>2.95</td>
<td>-0.26</td>
<td>0</td>
<td>0</td>
<td>22825.26</td>
<td>-0.24601</td>
</tr>
<tr>
<td>431.71</td>
<td>0.14</td>
<td>2.71</td>
<td>-0.24</td>
<td>0</td>
<td>0</td>
<td>22824.89</td>
<td>-0.22635</td>
</tr>
<tr>
<td>431.49</td>
<td>0.13</td>
<td>2.49</td>
<td>-0.22</td>
<td>0</td>
<td>0</td>
<td>22824.54</td>
<td>-0.20823</td>
</tr>
<tr>
<td>431.29</td>
<td>0.12</td>
<td>2.29</td>
<td>-0.20</td>
<td>0</td>
<td>0</td>
<td>22824.2</td>
<td>-0.19152</td>
</tr>
<tr>
<td>431.11</td>
<td>0.11</td>
<td>2.11</td>
<td>-0.19</td>
<td>0</td>
<td>0</td>
<td>22823.88</td>
<td>-0.17613</td>
</tr>
<tr>
<td>430.91</td>
<td>0.13</td>
<td>2.91</td>
<td>-0.20</td>
<td>2.24E-02</td>
<td>-1</td>
<td>22825.23</td>
<td>-0.2197</td>
</tr>
<tr>
<td>430.63</td>
<td>0.20</td>
<td>4.63</td>
<td>-0.28</td>
<td>0.053943</td>
<td>-2</td>
<td>22828.46</td>
<td>-0.34669</td>
</tr>
<tr>
<td>430.22</td>
<td>0.28</td>
<td>6.22</td>
<td>-0.41</td>
<td>5.59E-02</td>
<td>-2</td>
<td>22831.82</td>
<td>-0.4879</td>
</tr>
<tr>
<td>429.68</td>
<td>0.35</td>
<td>7.68</td>
<td>-0.54</td>
<td>5.16E-02</td>
<td>-2</td>
<td>22834.91</td>
<td>-0.61224</td>
</tr>
<tr>
<td>429.00</td>
<td>0.45</td>
<td>10.00</td>
<td>-0.68</td>
<td>7.47E-02</td>
<td>-3</td>
<td>22839.4</td>
<td>-0.78332</td>
</tr>
<tr>
<td>428.13</td>
<td>0.45</td>
<td>12.13</td>
<td>-0.86</td>
<td>7.48E-02</td>
<td>-3</td>
<td>22843.88</td>
<td>-0.96942</td>
</tr>
<tr>
<td>427.10</td>
<td>0.65</td>
<td>14.10</td>
<td>-1.04</td>
<td>6.89E-02</td>
<td>-3</td>
<td>22848.02</td>
<td>-1.15363</td>
</tr>
<tr>
<td>425.90</td>
<td>0.74</td>
<td>15.90</td>
<td>-1.20</td>
<td>6.33E-02</td>
<td>-3</td>
<td>22851.82</td>
<td>-1.28758</td>
</tr>
<tr>
<td>424.53</td>
<td>0.86</td>
<td>18.53</td>
<td>-1.37</td>
<td>8.55E-02</td>
<td>-4</td>
<td>22856.95</td>
<td>-1.4847</td>
</tr>
<tr>
<td>422.93</td>
<td>1.01</td>
<td>21.93</td>
<td>-1.60</td>
<td>0.112053</td>
<td>-5</td>
<td>22863.67</td>
<td>-1.7527</td>
</tr>
<tr>
<td>421.06</td>
<td>1.17</td>
<td>25.06</td>
<td>-1.87</td>
<td>0.109375</td>
<td>-5</td>
<td>22870.23</td>
<td>-2.02343</td>
</tr>
<tr>
<td>418.94</td>
<td>1.31</td>
<td>27.94</td>
<td>-2.12</td>
<td>0.100782</td>
<td>-5</td>
<td>22876.28</td>
<td>-2.26679</td>
</tr>
<tr>
<td>416.58</td>
<td>1.44</td>
<td>30.58</td>
<td>-2.36</td>
<td>9.26E-02</td>
<td>-5</td>
<td>22881.84</td>
<td>-2.48945</td>
</tr>
<tr>
<td>413.99</td>
<td>1.60</td>
<td>33.99</td>
<td>-2.60</td>
<td>0.112419</td>
<td>-6</td>
<td>22888.58</td>
<td>-2.75148</td>
</tr>
<tr>
<td>411.10</td>
<td>1.79</td>
<td>38.10</td>
<td>-2.89</td>
<td>0.136788</td>
<td>-7</td>
<td>22896.79</td>
<td>-3.07909</td>
</tr>
<tr>
<td>407.85</td>
<td>2.01</td>
<td>42.85</td>
<td>-3.24</td>
<td>0.159482</td>
<td>-8</td>
<td>22906.36</td>
<td>-3.46234</td>
</tr>
<tr>
<td>404.23</td>
<td>2.22</td>
<td>47.23</td>
<td>-3.62</td>
<td>0.152968</td>
<td>-8</td>
<td>22915.54</td>
<td>-3.83828</td>
</tr>
<tr>
<td>400.26</td>
<td>2.43</td>
<td>51.26</td>
<td>-3.98</td>
<td>0.140839</td>
<td>-8</td>
<td>22923.99</td>
<td>-4.17809</td>
</tr>
<tr>
<td>395.95</td>
<td>2.61</td>
<td>54.95</td>
<td>-4.30</td>
<td>0.129391</td>
<td>-8</td>
<td>22931.75</td>
<td>-4.4893</td>
</tr>
<tr>
<td>391.35</td>
<td>2.78</td>
<td>58.35</td>
<td>-4.60</td>
<td>0.118866</td>
<td>-8</td>
<td>22938.88</td>
<td>-4.7749</td>
</tr>
<tr>
<td>386.47</td>
<td>2.94</td>
<td>61.47</td>
<td>-4.88</td>
<td>0.109199</td>
<td>-8</td>
<td>22945.43</td>
<td>-5.03717</td>
</tr>
<tr>
<td>381.34</td>
<td>3.08</td>
<td>64.34</td>
<td>-5.13</td>
<td>0.100319</td>
<td>-8</td>
<td>22951.45</td>
<td>-5.27808</td>
</tr>
<tr>
<td>375.97</td>
<td>3.22</td>
<td>66.97</td>
<td>-5.37</td>
<td>9.22E-02</td>
<td>-8</td>
<td>22956.98</td>
<td>-5.49939</td>
</tr>
<tr>
<td>370.39</td>
<td>3.34</td>
<td>69.39</td>
<td>-5.58</td>
<td>8.47E-02</td>
<td>-8</td>
<td>22962.06</td>
<td>-5.70269</td>
</tr>
<tr>
<td>364.61</td>
<td>3.45</td>
<td>71.61</td>
<td>-5.78</td>
<td>7.78E-02</td>
<td>-8</td>
<td>22966.73</td>
<td>-5.88947</td>
</tr>
<tr>
<td>358.65</td>
<td>3.55</td>
<td>73.65</td>
<td>-5.96</td>
<td>7.15E-02</td>
<td>-8</td>
<td>22971.02</td>
<td>-6.06106</td>
</tr>
<tr>
<td>352.53</td>
<td>3.65</td>
<td>75.53</td>
<td>-6.12</td>
<td>6.56E-02</td>
<td>-8</td>
<td>22974.96</td>
<td>-6.2187</td>
</tr>
<tr>
<td>346.25</td>
<td>3.73</td>
<td>77.25</td>
<td>-6.28</td>
<td>6.03E-02</td>
<td>-8</td>
<td>22978.57</td>
<td>-6.36352</td>
</tr>
<tr>
<td>339.83</td>
<td>3.81</td>
<td>78.83</td>
<td>-6.42</td>
<td>5.54E-02</td>
<td>-8</td>
<td>22981.9</td>
<td>-6.49657</td>
</tr>
<tr>
<td>333.29</td>
<td>3.89</td>
<td>80.29</td>
<td>-6.55</td>
<td>5.09E-02</td>
<td>-8</td>
<td>22984.95</td>
<td>-6.6188</td>
</tr>
<tr>
<td>326.62</td>
<td>3.95</td>
<td>81.62</td>
<td>-6.66</td>
<td>0.046764</td>
<td>-8</td>
<td>22987.76</td>
<td>-6.73109</td>
</tr>
<tr>
<td>319.85</td>
<td>4.02</td>
<td>82.85</td>
<td>-6.77</td>
<td>4.30E-02</td>
<td>-8</td>
<td>22990.34</td>
<td>-6.83426</td>
</tr>
<tr>
<td>312.98</td>
<td>4.07</td>
<td>83.98</td>
<td>-6.87</td>
<td>3.95E-02</td>
<td>-8</td>
<td>22992.71</td>
<td>-6.92903</td>
</tr>
<tr>
<td>306.02</td>
<td>4.13</td>
<td>85.02</td>
<td>-6.96</td>
<td>3.63E-02</td>
<td>-8</td>
<td>22994.88</td>
<td>-7.0161</td>
</tr>
<tr>
<td>298.97</td>
<td>4.17</td>
<td>85.97</td>
<td>-7.05</td>
<td>3.33E-02</td>
<td>-8</td>
<td>22996.88</td>
<td>-7.0961</td>
</tr>
<tr>
<td>291.84</td>
<td>4.22</td>
<td>86.84</td>
<td>-7.13</td>
<td>3.06E-02</td>
<td>-8</td>
<td>22998.71</td>
<td>-7.16958</td>
</tr>
<tr>
<td>284.65</td>
<td>4.26</td>
<td>87.65</td>
<td>-7.20</td>
<td>2.81E-02</td>
<td>-8</td>
<td>23000.4</td>
<td>-7.2371</td>
</tr>
<tr>
<td>277.38</td>
<td>4.29</td>
<td>88.38</td>
<td>-7.26</td>
<td>2.58E-02</td>
<td>-8</td>
<td>23001.95</td>
<td>-7.29912</td>
</tr>
<tr>
<td>270.06</td>
<td>4.33</td>
<td>89.06</td>
<td>-7.32</td>
<td>2.37E-02</td>
<td>-8</td>
<td>23003.38</td>
<td>-7.35611</td>
</tr>
<tr>
<td>262.68</td>
<td>4.36</td>
<td>89.68</td>
<td>-7.38</td>
<td>2.18E-02</td>
<td>-8</td>
<td>23004.68</td>
<td>-7.40846</td>
</tr>
<tr>
<td>255.26</td>
<td>4.39</td>
<td>90.26</td>
<td>-7.43</td>
<td>2.00E-02</td>
<td>-8</td>
<td>23005.88</td>
<td>-7.45655</td>
</tr>
<tr>
<td>247.78</td>
<td>4.42</td>
<td>90.78</td>
<td>-7.47</td>
<td>1.84E-02</td>
<td>-8</td>
<td>23006.99</td>
<td>-7.50073</td>
</tr>
<tr>
<td>240.27</td>
<td>4.44</td>
<td>91.27</td>
<td>-7.52</td>
<td>1.69E-02</td>
<td>-8</td>
<td>23008.75</td>
<td>-7.54132</td>
</tr>
<tr>
<td>232.71</td>
<td>4.46</td>
<td>91.71</td>
<td>-7.56</td>
<td>1.55E-02</td>
<td>-8</td>
<td>23008.94</td>
<td>-7.57861</td>
</tr>
<tr>
<td>225.12</td>
<td>4.48</td>
<td>92.12</td>
<td>-7.59</td>
<td>1.43E-02</td>
<td>-8</td>
<td>23009.79</td>
<td>-7.61287</td>
</tr>
<tr>
<td>217.49</td>
<td>4.50</td>
<td>92.49</td>
<td>-7.63</td>
<td>0.013107</td>
<td>-8</td>
<td>23010.58</td>
<td>-7.64435</td>
</tr>
<tr>
<td>209.84</td>
<td>4.52</td>
<td>92.84</td>
<td>-7.66</td>
<td>0.012041</td>
<td>-8</td>
<td>23011.3</td>
<td>-7.67326</td>
</tr>
<tr>
<td>202.15</td>
<td>4.53</td>
<td>93.15</td>
<td>-7.68</td>
<td>1.11E-02</td>
<td>-8</td>
<td>23011.96</td>
<td>-7.69983</td>
</tr>
<tr>
<td>194.44</td>
<td>4.55</td>
<td>93.44</td>
<td>-7.71</td>
<td>1.02E-02</td>
<td>-8</td>
<td>23012.57</td>
<td>-7.72423</td>
</tr>
<tr>
<td>186.71</td>
<td>4.56</td>
<td>93.71</td>
<td>-7.73</td>
<td>9.34E-03</td>
<td>-8</td>
<td>23013.13</td>
<td>-7.74665</td>
</tr>
<tr>
<td>178.93</td>
<td>4.61</td>
<td>94.93</td>
<td>-7.78</td>
<td>3.59E-02</td>
<td>-9</td>
<td>23015.29</td>
<td>-7.825</td>
</tr>
<tr>
<td>171.11</td>
<td>4.53</td>
<td>91.11</td>
<td>-7.82</td>
<td>0</td>
<td>-10</td>
<td>23009.43</td>
<td>-7.74469</td>
</tr>
<tr>
<td>163.72</td>
<td>4.20</td>
<td>83.72</td>
<td>-7.39</td>
<td>0</td>
<td>-11</td>
<td>22994.58</td>
<td>-7.063</td>
</tr>
<tr>
<td>156.91</td>
<td>3.87</td>
<td>76.91</td>
<td>-6.81</td>
<td>0</td>
<td>-12</td>
<td>22980.32</td>
<td>-6.4715</td>
</tr>
<tr>
<td>150.66</td>
<td>3.56</td>
<td>70.66</td>
<td>-6.25</td>
<td>0</td>
<td>-11</td>
<td>22967.19</td>
<td>-5.94087</td>
</tr>
<tr>
<td>144.91</td>
<td>3.27</td>
<td>64.91</td>
<td>-5.74</td>
<td>0</td>
<td>-10</td>
<td>22955.12</td>
<td>-5.45628</td>
</tr>
<tr>
<td>139.63</td>
<td>3.00</td>
<td>59.63</td>
<td>-5.28</td>
<td>0</td>
<td>-9</td>
<td>22944.04</td>
<td>-5.0121</td>
</tr>
<tr>
<td>134.79</td>
<td>2.76</td>
<td>54.79</td>
<td>-4.85</td>
<td>0</td>
<td>-8</td>
<td>22933.86</td>
<td>-4.60439</td>
</tr>
<tr>
<td>130.33</td>
<td>2.53</td>
<td>50.33</td>
<td>-4.45</td>
<td>0</td>
<td>-7</td>
<td>22924.51</td>
<td>-4.22996</td>
</tr>
<tr>
<td>126.24</td>
<td>2.33</td>
<td>46.24</td>
<td>-4.09</td>
<td>0</td>
<td>-6</td>
<td>22915.91</td>
<td>-3.88603</td>
</tr>
<tr>
<td>122.48</td>
<td>2.14</td>
<td>42.48</td>
<td>-3.76</td>
<td>0</td>
<td>-5</td>
<td>22908.02</td>
<td>-3.57008</td>
</tr>
<tr>
<td>119.03</td>
<td>1.96</td>
<td>39.03</td>
<td>-3.45</td>
<td>0</td>
<td>-4</td>
<td>22900.77</td>
<td>-3.27983</td>
</tr>
<tr>
<td>115.85</td>
<td>1.81</td>
<td>35.85</td>
<td>-3.17</td>
<td>0</td>
<td>-3</td>
<td>22894.11</td>
<td>-3.01317</td>
</tr>
<tr>
<td>112.94</td>
<td>1.66</td>
<td>32.94</td>
<td>-2.91</td>
<td>0</td>
<td>-2</td>
<td>22887.98</td>
<td>-2.7682</td>
</tr>
<tr>
<td>110.26</td>
<td>1.52</td>
<td>30.26</td>
<td>-2.68</td>
<td>0</td>
<td>-2</td>
<td>22882.36</td>
<td>-2.53414</td>
</tr>
<tr>
<td>107.80</td>
<td>1.40</td>
<td>27.80</td>
<td>-2.46</td>
<td>0</td>
<td>-2</td>
<td>22877.2</td>
<td>-2.33638</td>
</tr>
<tr>
<td>105.54</td>
<td>1.29</td>
<td>25.54</td>
<td>-2.26</td>
<td>0</td>
<td>-2</td>
<td>22872.45</td>
<td>-2.14643</td>
</tr>
<tr>
<td>103.46</td>
<td>1.18</td>
<td>23.46</td>
<td>-2.08</td>
<td>0</td>
<td>-2</td>
<td>22868.09</td>
<td>-1.97192</td>
</tr>
<tr>
<td>101.56</td>
<td>1.09</td>
<td>21.56</td>
<td>-1.91</td>
<td>0</td>
<td>-2</td>
<td>22864.08</td>
<td>-1.8116</td>
</tr>
<tr>
<td>99.80</td>
<td>1.00</td>
<td>19.80</td>
<td>-1.75</td>
<td>0</td>
<td>-2</td>
<td>22860.4</td>
<td>-1.66432</td>
</tr>
<tr>
<td>98.19</td>
<td>0.92</td>
<td>18.19</td>
<td>-1.61</td>
<td>0</td>
<td>-2</td>
<td>22857.02</td>
<td>-1.52901</td>
</tr>
<tr>
<td>96.71</td>
<td>0.84</td>
<td>16.71</td>
<td>-1.48</td>
<td>0</td>
<td>-2</td>
<td>22853.92</td>
<td>-1.4047</td>
</tr>
<tr>
<td>95.36</td>
<td>0.77</td>
<td>15.36</td>
<td>-1.36</td>
<td>0</td>
<td>-2</td>
<td>22851.06</td>
<td>-1.29049</td>
</tr>
<tr>
<td>94.11</td>
<td>0.71</td>
<td>14.11</td>
<td>-1.25</td>
<td>0</td>
<td>-2</td>
<td>22848.44</td>
<td>-1.18557</td>
</tr>
<tr>
<td>92.96</td>
<td>0.65</td>
<td>12.96</td>
<td>-1.15</td>
<td>0</td>
<td>-2</td>
<td>22846.04</td>
<td>-1.08919</td>
</tr>
<tr>
<td>91.91</td>
<td>0.60</td>
<td>11.91</td>
<td>-1.05</td>
<td>0</td>
<td>-2</td>
<td>22843.81</td>
<td>-1.00063</td>
</tr>
<tr>
<td>90.94</td>
<td>0.55</td>
<td>10.94</td>
<td>-0.97</td>
<td>0</td>
<td>-2</td>
<td>22841.79</td>
<td>-0.91928</td>
</tr>
<tr>
<td>90.05</td>
<td>0.51</td>
<td>10.05</td>
<td>-0.89</td>
<td>0</td>
<td>-1</td>
<td>22839.92</td>
<td>-0.84454</td>
</tr>
<tr>
<td>89.23</td>
<td>0.46</td>
<td>9.23</td>
<td>-0.82</td>
<td>0</td>
<td>0</td>
<td>22838.21</td>
<td>-0.77588</td>
</tr>
<tr>
<td>88.48</td>
<td>0.43</td>
<td>8.48</td>
<td>-0.75</td>
<td>0</td>
<td>0</td>
<td>22836.63</td>
<td>-0.7128</td>
</tr>
<tr>
<td>87.79</td>
<td>0.39</td>
<td>7.79</td>
<td>-0.69</td>
<td>0</td>
<td>0</td>
<td>22835.18</td>
<td>-0.65485</td>
</tr>
<tr>
<td>87.16</td>
<td>0.36</td>
<td>7.16</td>
<td>-0.63</td>
<td>0</td>
<td>0</td>
<td>22833.85</td>
<td>-0.60161</td>
</tr>
<tr>
<td>86.58</td>
<td>0.33</td>
<td>6.58</td>
<td>-0.58</td>
<td>0</td>
<td>0</td>
<td>22832.63</td>
<td>-0.5527</td>
</tr>
<tr>
<td>86.04</td>
<td>0.30</td>
<td>6.04</td>
<td>-0.53</td>
<td>0</td>
<td>0</td>
<td>22831.51</td>
<td>-0.50776</td>
</tr>
<tr>
<td>85.55</td>
<td>0.28</td>
<td>5.55</td>
<td>-0.49</td>
<td>0</td>
<td>0</td>
<td>22830.48</td>
<td>-0.46648</td>
</tr>
<tr>
<td>85.10</td>
<td>0.26</td>
<td>5.10</td>
<td>-0.45</td>
<td>0</td>
<td>0</td>
<td>22829.53</td>
<td>-0.42856</td>
</tr>
<tr>
<td>84.68</td>
<td>0.24</td>
<td>4.68</td>
<td>-0.41</td>
<td>0</td>
<td>0</td>
<td>22828.66</td>
<td>-0.39371</td>
</tr>
<tr>
<td>84.30</td>
<td>0.22</td>
<td>4.30</td>
<td>-0.38</td>
<td>0</td>
<td>0</td>
<td>22827.86</td>
<td>-0.3617</td>
</tr>
<tr>
<td>83.95</td>
<td>0.20</td>
<td>3.95</td>
<td>-0.35</td>
<td>0</td>
<td>0</td>
<td>22827.12</td>
<td>-0.3323</td>
</tr>
<tr>
<td>83.63</td>
<td>0.18</td>
<td>3.63</td>
<td>-0.32</td>
<td>0</td>
<td>0</td>
<td>22826.45</td>
<td>-0.30528</td>
</tr>
<tr>
<td>83.34</td>
<td>0.17</td>
<td>3.34</td>
<td>-0.30</td>
<td>0</td>
<td>0</td>
<td>22825.83</td>
<td>-0.28046</td>
</tr>
<tr>
<td>83.07</td>
<td>0.15</td>
<td>3.07</td>
<td>-0.27</td>
<td>0</td>
<td>0</td>
<td>22825.26</td>
<td>-0.25766</td>
</tr>
<tr>
<td>82.82</td>
<td>0.14</td>
<td>2.82</td>
<td>-0.25</td>
<td>0</td>
<td>0</td>
<td>22824.73</td>
<td>-0.23671</td>
</tr>
<tr>
<td>82.59</td>
<td>0.13</td>
<td>2.59</td>
<td>-0.23</td>
<td>0</td>
<td>0</td>
<td>22824.25</td>
<td>-0.21747</td>
</tr>
<tr>
<td>82.38</td>
<td>0.12</td>
<td>2.38</td>
<td>-0.21</td>
<td>0</td>
<td>0</td>
<td>22823.81</td>
<td>-0.19979</td>
</tr>
<tr>
<td>82.18</td>
<td>0.11</td>
<td>2.18</td>
<td>-0.19</td>
<td>0</td>
<td>0</td>
<td>22823.41</td>
<td>-0.18354</td>
</tr>
<tr>
<td>82.01</td>
<td>0.10</td>
<td>2.01</td>
<td>-0.18</td>
<td>0</td>
<td>0</td>
<td>22823.03</td>
<td>-0.16862</td>
</tr>
<tr>
<td>81.84</td>
<td>0.09</td>
<td>1.84</td>
<td>-0.16</td>
<td>0</td>
<td>0</td>
<td>22822.69</td>
<td>-0.15491</td>
</tr>
<tr>
<td>81.69</td>
<td>0.09</td>
<td>1.69</td>
<td>-0.15</td>
<td>0</td>
<td>0</td>
<td>22822.38</td>
<td>-0.14232</td>
</tr>
<tr>
<td>81.56</td>
<td>0.08</td>
<td>1.56</td>
<td>-0.14</td>
<td>0</td>
<td>0</td>
<td>22822.09</td>
<td>-0.13075</td>
</tr>
<tr>
<td>81.43</td>
<td>0.07</td>
<td>1.43</td>
<td>-0.13</td>
<td>0</td>
<td>0</td>
<td>22821.82</td>
<td>-0.12012</td>
</tr>
<tr>
<td>81.31</td>
<td>0.07</td>
<td>1.31</td>
<td>-0.12</td>
<td>0</td>
<td>0</td>
<td>22821.58</td>
<td>-0.11035</td>
</tr>
<tr>
<td>81.21</td>
<td>0.06</td>
<td>1.21</td>
<td>-0.11</td>
<td>0</td>
<td>0</td>
<td>22821.35</td>
<td>-0.10138</td>
</tr>
<tr>
<td>81.11</td>
<td>0.06</td>
<td>1.11</td>
<td>-0.10</td>
<td>0</td>
<td>0</td>
<td>22821.15</td>
<td>-0.0932</td>
</tr>
<tr>
<td>81.02</td>
<td>0.05</td>
<td>1.02</td>
<td>-0.09</td>
<td>0</td>
<td>0</td>
<td>22820.96</td>
<td>-0.0856</td>
</tr>
<tr>
<td>80.94</td>
<td>0.05</td>
<td>0.94</td>
<td>-0.08</td>
<td>0</td>
<td>0</td>
<td>22820.79</td>
<td>-0.0786</td>
</tr>
<tr>
<td>80.86</td>
<td>0.04</td>
<td>0.86</td>
<td>-0.08</td>
<td>0</td>
<td>0</td>
<td>22820.63</td>
<td>-0.0722</td>
</tr>
<tr>
<td>80.79</td>
<td>0.04</td>
<td>0.79</td>
<td>-0.07</td>
<td>0</td>
<td>0</td>
<td>22820.48</td>
<td>-0.0663</td>
</tr>
<tr>
<td>80.73</td>
<td>0.04</td>
<td>0.73</td>
<td>-0.06</td>
<td>0</td>
<td>0</td>
<td>22820.34</td>
<td>-0.0601</td>
</tr>
<tr>
<td>80.67</td>
<td>0.03</td>
<td>0.67</td>
<td>-0.06</td>
<td>0</td>
<td>0</td>
<td>22820.22</td>
<td>-0.0560</td>
</tr>
<tr>
<td>80.61</td>
<td>0.03</td>
<td>0.61</td>
<td>-0.05</td>
<td>0</td>
<td>0</td>
<td>22820.11</td>
<td>-0.0514</td>
</tr>
<tr>
<td>80.56</td>
<td>0.03</td>
<td>0.56</td>
<td>-0.05</td>
<td>0</td>
<td>0</td>
<td>22820.01</td>
<td>-0.0473</td>
</tr>
<tr>
<td>80.52</td>
<td>0.03</td>
<td>0.52</td>
<td>-0.05</td>
<td>0</td>
<td>0</td>
<td>22819.91</td>
<td>-0.0434</td>
</tr>
<tr>
<td>80.47</td>
<td>0.02</td>
<td>0.47</td>
<td>-0.04</td>
<td>0</td>
<td>0</td>
<td>22819.82</td>
<td>-0.0399</td>
</tr>
<tr>
<td>80.44</td>
<td>0.02</td>
<td>0.44</td>
<td>-0.04</td>
<td>0</td>
<td>0</td>
<td>22819.74</td>
<td>-0.0366</td>
</tr>
<tr>
<td>80.40</td>
<td>0.02</td>
<td>0.40</td>
<td>-0.04</td>
<td>0</td>
<td>0</td>
<td>22819.66</td>
<td>-0.0337</td>
</tr>
<tr>
<td>80.37</td>
<td>0.02</td>
<td>0.37</td>
<td>-0.03</td>
<td>0</td>
<td>0</td>
<td>22819.59</td>
<td>-0.0309</td>
</tr>
<tr>
<td>80.34</td>
<td>0.02</td>
<td>0.34</td>
<td>-0.03</td>
<td>0</td>
<td>0</td>
<td>22819.53</td>
<td>-0.0284</td>
</tr>
</tbody>
</table>
## Thick Panel (256 plies) Training Set

**[Input Ranges]**

- **Tmid**: 80 to 500
- **Tmid-Top**: -10 to 50
- **Tmid-Taut**: -100 to 200
- **dT/dt**: -2.5 to 5
- **Hr**: 0 to 4

**[Output Ranges]**

- **Tadj**: -12 to 12

**[Training Data]**

<table>
<thead>
<tr>
<th>Tmid (Y2)</th>
<th>id-Top (Y1)</th>
<th>id-Taut (Y1)</th>
<th>dT/dt (Y1)</th>
<th>Hr (Y1)</th>
<th>Tadj (Y1)</th>
<th>not used</th>
<th>not used</th>
</tr>
</thead>
<tbody>
<tr>
<td>80.00</td>
<td>-0.01</td>
<td>-1.00</td>
<td>0.00</td>
<td>3.28E-03</td>
<td>1</td>
<td>0.19692</td>
<td>8.53E-03</td>
</tr>
<tr>
<td>80.01</td>
<td>-0.03</td>
<td>-1.99</td>
<td>0.00</td>
<td>3.28E-03</td>
<td>1</td>
<td>0.393751</td>
<td>3.11E-02</td>
</tr>
<tr>
<td>80.01</td>
<td>-0.09</td>
<td>-2.99</td>
<td>0.00</td>
<td>3.26E-03</td>
<td>1</td>
<td>0.589565</td>
<td>5.93E-02</td>
</tr>
<tr>
<td>80.01</td>
<td>-0.17</td>
<td>-3.99</td>
<td>0.00</td>
<td>3.21E-03</td>
<td>1</td>
<td>0.781992</td>
<td>8.65E-02</td>
</tr>
<tr>
<td>80.02</td>
<td>-0.28</td>
<td>-4.98</td>
<td>0.00</td>
<td>3.09E-03</td>
<td>1</td>
<td>0.967638</td>
<td>0.111426</td>
</tr>
<tr>
<td>80.02</td>
<td>-0.41</td>
<td>-5.98</td>
<td>0.00</td>
<td>2.92E-03</td>
<td>1</td>
<td>1.143074</td>
<td>0.134011</td>
</tr>
<tr>
<td>80.03</td>
<td>-0.56</td>
<td>-6.97</td>
<td>0.01</td>
<td>2.71E-03</td>
<td>1</td>
<td>1.3056</td>
<td>0.154518</td>
</tr>
<tr>
<td>80.03</td>
<td>-0.73</td>
<td>-7.97</td>
<td>0.01</td>
<td>2.47E-03</td>
<td>1</td>
<td>1.453536</td>
<td>0.173236</td>
</tr>
<tr>
<td>80.04</td>
<td>-0.91</td>
<td>-9.96</td>
<td>0.01</td>
<td>2.21E-03</td>
<td>2</td>
<td>1.58618</td>
<td>0.196508</td>
</tr>
<tr>
<td>80.05</td>
<td>-1.14</td>
<td>-11.95</td>
<td>0.01</td>
<td>1.95E-03</td>
<td>1</td>
<td>1.70345</td>
<td>0.235261</td>
</tr>
<tr>
<td>80.07</td>
<td>-1.40</td>
<td>-13.93</td>
<td>0.02</td>
<td>1.69E-03</td>
<td>1</td>
<td>1.804968</td>
<td>0.278227</td>
</tr>
<tr>
<td>80.09</td>
<td>-1.70</td>
<td>-15.91</td>
<td>0.02</td>
<td>1.41E-03</td>
<td>1</td>
<td>1.889311</td>
<td>0.319197</td>
</tr>
<tr>
<td>80.11</td>
<td>-2.03</td>
<td>-17.89</td>
<td>0.02</td>
<td>1.08E-03</td>
<td>1</td>
<td>1.954299</td>
<td>0.356983</td>
</tr>
<tr>
<td>80.14</td>
<td>-2.40</td>
<td>-20.86</td>
<td>0.03</td>
<td>7.26E-04</td>
<td>3</td>
<td>1.997839</td>
<td>0.397759</td>
</tr>
<tr>
<td>80.18</td>
<td>-2.81</td>
<td>-23.82</td>
<td>0.04</td>
<td>3.43E-04</td>
<td>3</td>
<td>2.018418</td>
<td>0.452587</td>
</tr>
<tr>
<td>80.22</td>
<td>-3.28</td>
<td>-26.78</td>
<td>0.04</td>
<td>0</td>
<td>3</td>
<td>2.014651</td>
<td>0.510386</td>
</tr>
<tr>
<td>80.28</td>
<td>-3.79</td>
<td>-29.72</td>
<td>0.05</td>
<td>0</td>
<td>3</td>
<td>1.984655</td>
<td>0.565119</td>
</tr>
<tr>
<td>80.34</td>
<td>-4.35</td>
<td>-32.66</td>
<td>0.06</td>
<td>0</td>
<td>3</td>
<td>1.926286</td>
<td>0.615745</td>
</tr>
<tr>
<td>80.41</td>
<td>-4.94</td>
<td>-35.59</td>
<td>0.07</td>
<td>0</td>
<td>3</td>
<td>1.837867</td>
<td>0.662476</td>
</tr>
<tr>
<td>80.50</td>
<td>-5.56</td>
<td>-39.50</td>
<td>0.09</td>
<td>0</td>
<td>4</td>
<td>1.718714</td>
<td>0.711852</td>
</tr>
<tr>
<td>80.60</td>
<td>-6.24</td>
<td>-43.40</td>
<td>0.10</td>
<td>0</td>
<td>4</td>
<td>1.569107</td>
<td>0.775033</td>
</tr>
<tr>
<td>80.71</td>
<td>-6.97</td>
<td>-47.29</td>
<td>0.11</td>
<td>0</td>
<td>4</td>
<td>1.389451</td>
<td>0.840969</td>
</tr>
<tr>
<td>80.84</td>
<td>-7.74</td>
<td>-51.16</td>
<td>0.13</td>
<td>0</td>
<td>4</td>
<td>1.179459</td>
<td>0.903635</td>
</tr>
<tr>
<td>80.99</td>
<td>-8.56</td>
<td>-56.01</td>
<td>0.15</td>
<td>0</td>
<td>5</td>
<td>0.938333</td>
<td>0.968085</td>
</tr>
<tr>
<td>81.15</td>
<td>-9.44</td>
<td>-60.85</td>
<td>0.16</td>
<td>0</td>
<td>5</td>
<td>0.665336</td>
<td>1.045252</td>
</tr>
<tr>
<td>81.33</td>
<td>-10.38</td>
<td>-65.67</td>
<td>0.18</td>
<td>0</td>
<td>5</td>
<td>0.359677</td>
<td>1.124155</td>
</tr>
<tr>
<td>81.54</td>
<td>-11.38</td>
<td>-70.46</td>
<td>0.21</td>
<td>0</td>
<td>5</td>
<td>0.020133</td>
<td>1.198897</td>
</tr>
<tr>
<td>81.77</td>
<td>-12.42</td>
<td>-76.23</td>
<td>0.23</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>1.274656</td>
</tr>
<tr>
<td>82.02</td>
<td>-13.53</td>
<td>-81.98</td>
<td>0.25</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>1.362474</td>
</tr>
<tr>
<td>82.30</td>
<td>-14.70</td>
<td>-87.70</td>
<td>0.28</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>1.45146</td>
</tr>
<tr>
<td>82.61</td>
<td>-15.93</td>
<td>-93.39</td>
<td>0.31</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>1.535791</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>--------</td>
<td>---------</td>
<td>--------</td>
<td>-----</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>82.94</td>
<td>-17.22</td>
<td>-100.06</td>
<td>0.34</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>1.62071</td>
</tr>
<tr>
<td>83.31</td>
<td>-18.57</td>
<td>-106.69</td>
<td>0.37</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>1.717313</td>
</tr>
<tr>
<td>83.71</td>
<td>-19.98</td>
<td>-113.29</td>
<td>0.40</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>1.814755</td>
</tr>
<tr>
<td>84.14</td>
<td>-21.46</td>
<td>-119.86</td>
<td>0.43</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>1.907256</td>
</tr>
<tr>
<td>84.61</td>
<td>-22.99</td>
<td>-127.39</td>
<td>0.47</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>2.000093</td>
</tr>
<tr>
<td>85.12</td>
<td>-24.58</td>
<td>-134.88</td>
<td>0.51</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>2.104396</td>
</tr>
<tr>
<td>85.67</td>
<td>-26.25</td>
<td>-142.33</td>
<td>0.55</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>2.209354</td>
</tr>
<tr>
<td>86.26</td>
<td>-27.97</td>
<td>-149.74</td>
<td>0.59</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>2.309214</td>
</tr>
<tr>
<td>86.89</td>
<td>-29.74</td>
<td>-158.11</td>
<td>0.63</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>2.409281</td>
</tr>
<tr>
<td>87.57</td>
<td>-31.58</td>
<td>-166.43</td>
<td>0.68</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>2.520716</td>
</tr>
<tr>
<td>88.30</td>
<td>-33.49</td>
<td>-174.70</td>
<td>0.73</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>2.63274</td>
</tr>
<tr>
<td>89.07</td>
<td>-35.45</td>
<td>-181.93</td>
<td>0.78</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>2.73354</td>
</tr>
<tr>
<td>89.90</td>
<td>-37.43</td>
<td>-189.10</td>
<td>0.83</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>2.811645</td>
</tr>
<tr>
<td>90.78</td>
<td>-39.43</td>
<td>-196.22</td>
<td>0.88</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>2.878962</td>
</tr>
<tr>
<td>91.72</td>
<td>-41.43</td>
<td>-202.28</td>
<td>0.93</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>2.936446</td>
</tr>
<tr>
<td>92.71</td>
<td>-43.42</td>
<td>-208.29</td>
<td>0.99</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>2.975345</td>
</tr>
<tr>
<td>93.76</td>
<td>-45.38</td>
<td>-214.24</td>
<td>1.05</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>3.007693</td>
</tr>
<tr>
<td>94.86</td>
<td>-47.30</td>
<td>-219.14</td>
<td>1.11</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>3.034009</td>
</tr>
<tr>
<td>96.03</td>
<td>-49.18</td>
<td>-223.97</td>
<td>1.17</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>3.045035</td>
</tr>
<tr>
<td>97.26</td>
<td>-51.00</td>
<td>-227.74</td>
<td>1.23</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>3.046261</td>
</tr>
<tr>
<td>98.54</td>
<td>-52.74</td>
<td>-230.46</td>
<td>1.29</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>3.027083</td>
</tr>
<tr>
<td>99.89</td>
<td>-54.38</td>
<td>-232.11</td>
<td>1.35</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>2.983234</td>
</tr>
<tr>
<td>101.29</td>
<td>-55.89</td>
<td>-232.71</td>
<td>1.40</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2.915515</td>
</tr>
<tr>
<td>102.76</td>
<td>-57.25</td>
<td>-232.24</td>
<td>1.46</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2.825777</td>
</tr>
<tr>
<td>104.27</td>
<td>-58.46</td>
<td>-231.73</td>
<td>1.52</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2.721967</td>
</tr>
<tr>
<td>105.85</td>
<td>-59.51</td>
<td>-231.15</td>
<td>1.57</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2.622582</td>
</tr>
<tr>
<td>107.47</td>
<td>-60.42</td>
<td>-230.53</td>
<td>1.62</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2.5345</td>
</tr>
<tr>
<td>109.14</td>
<td>-61.21</td>
<td>-229.86</td>
<td>1.67</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2.458039</td>
</tr>
<tr>
<td>110.85</td>
<td>-61.88</td>
<td>-229.15</td>
<td>1.71</td>
<td>0.965E-04</td>
<td>1</td>
<td>0</td>
<td>2.392033</td>
</tr>
<tr>
<td>112.61</td>
<td>-62.46</td>
<td>-227.39</td>
<td>1.76</td>
<td>4.33E-03</td>
<td>0</td>
<td>0</td>
<td>2.329052</td>
</tr>
<tr>
<td>114.40</td>
<td>-62.92</td>
<td>-225.60</td>
<td>1.79</td>
<td>7.87E-03</td>
<td>0</td>
<td>0</td>
<td>2.257083</td>
</tr>
<tr>
<td>116.23</td>
<td>-63.28</td>
<td>-223.77</td>
<td>1.83</td>
<td>1.16E-02</td>
<td>0</td>
<td>0</td>
<td>2.186473</td>
</tr>
<tr>
<td>118.08</td>
<td>-63.55</td>
<td>-221.92</td>
<td>1.86</td>
<td>0.015465</td>
<td>0</td>
<td>0</td>
<td>2.122729</td>
</tr>
<tr>
<td>119.96</td>
<td>-63.74</td>
<td>-220.04</td>
<td>1.88</td>
<td>1.96E-02</td>
<td>0</td>
<td>0</td>
<td>2.066468</td>
</tr>
<tr>
<td>121.87</td>
<td>-63.85</td>
<td>-218.13</td>
<td>1.91</td>
<td>2.39E-02</td>
<td>0</td>
<td>0</td>
<td>2.017116</td>
</tr>
<tr>
<td>123.79</td>
<td>-63.90</td>
<td>-216.21</td>
<td>1.93</td>
<td>0.028433</td>
<td>0</td>
<td>0</td>
<td>1.973916</td>
</tr>
<tr>
<td>125.74</td>
<td>-63.89</td>
<td>-214.26</td>
<td>1.94</td>
<td>3.32E-02</td>
<td>0</td>
<td>0</td>
<td>1.936158</td>
</tr>
<tr>
<td>127.70</td>
<td>-63.83</td>
<td>-212.30</td>
<td>1.96</td>
<td>0.038335</td>
<td>0</td>
<td>0</td>
<td>1.903225</td>
</tr>
<tr>
<td>129.67</td>
<td>-63.73</td>
<td>-210.33</td>
<td>1.97</td>
<td>4.37E-02</td>
<td>0</td>
<td>0</td>
<td>1.874594</td>
</tr>
<tr>
<td>131.65</td>
<td>-63.60</td>
<td>-208.35</td>
<td>1.98</td>
<td>0.049479</td>
<td>0</td>
<td>0</td>
<td>1.849821</td>
</tr>
<tr>
<td>133.65</td>
<td>-63.43</td>
<td>-206.35</td>
<td>1.99</td>
<td>5.56E-02</td>
<td>0</td>
<td>0</td>
<td>1.828529</td>
</tr>
<tr>
<td>135.65</td>
<td>-63.24</td>
<td>-204.35</td>
<td>2.00</td>
<td>6.22E-02</td>
<td>0</td>
<td>0.476559</td>
<td>1.810396</td>
</tr>
<tr>
<td>137.66</td>
<td>-63.02</td>
<td>-202.34</td>
<td>2.01</td>
<td>6.93E-02</td>
<td>0</td>
<td>0.463168</td>
<td>1.795147</td>
</tr>
<tr>
<td>139.68</td>
<td>-62.79</td>
<td>-200.32</td>
<td>2.02</td>
<td>0.07687</td>
<td>0</td>
<td>9.243903</td>
<td>1.782546</td>
</tr>
<tr>
<td>141.71</td>
<td>-62.53</td>
<td>-198.29</td>
<td>2.03</td>
<td>8.51E-02</td>
<td>0</td>
<td>14.35019</td>
<td>1.772391</td>
</tr>
<tr>
<td>143.74</td>
<td>-62.27</td>
<td>-196.26</td>
<td>2.03</td>
<td>9.40E-02</td>
<td>0</td>
<td>19.9916</td>
<td>1.764505</td>
</tr>
<tr>
<td>145.78</td>
<td>-61.99</td>
<td>-194.22</td>
<td>2.04</td>
<td>0.103701</td>
<td>0</td>
<td>26.21366</td>
<td>1.758739</td>
</tr>
<tr>
<td>147.82</td>
<td>-61.70</td>
<td>-192.18</td>
<td>2.05</td>
<td>0.114216</td>
<td>0</td>
<td>33.06662</td>
<td>1.754963</td>
</tr>
<tr>
<td>431.19</td>
<td>95.40</td>
<td>351.19</td>
<td>-3.02</td>
<td>2.25E-03</td>
<td>-6 22906.06</td>
<td>-2.96935</td>
<td></td>
</tr>
<tr>
<td>428.16</td>
<td>95.29</td>
<td>348.16</td>
<td>-3.03</td>
<td>0.001104</td>
<td>-6 22906.12</td>
<td>-2.91696</td>
<td></td>
</tr>
<tr>
<td>425.14</td>
<td>95.13</td>
<td>345.14</td>
<td>-3.03</td>
<td>3.70E-05</td>
<td>-6 22906.13</td>
<td>-2.86508</td>
<td></td>
</tr>
<tr>
<td>422.11</td>
<td>94.92</td>
<td>342.11</td>
<td>-3.03</td>
<td>0</td>
<td>-6 22906.07</td>
<td>-2.81411</td>
<td></td>
</tr>
<tr>
<td>419.09</td>
<td>94.66</td>
<td>339.09</td>
<td>-3.02</td>
<td>0</td>
<td>-6 22905.96</td>
<td>-2.76429</td>
<td></td>
</tr>
<tr>
<td>416.07</td>
<td>94.36</td>
<td>336.07</td>
<td>-3.02</td>
<td>0.001104</td>
<td>-6 22905.79</td>
<td>-2.71581</td>
<td></td>
</tr>
<tr>
<td>413.06</td>
<td>94.02</td>
<td>333.06</td>
<td>-3.01</td>
<td>0</td>
<td>-6 22905.59</td>
<td>-2.66875</td>
<td></td>
</tr>
<tr>
<td>410.06</td>
<td>93.64</td>
<td>330.06</td>
<td>-3.00</td>
<td>3.70E-05</td>
<td>-6 22905.05</td>
<td>-2.5791</td>
<td></td>
</tr>
<tr>
<td>407.07</td>
<td>93.23</td>
<td>327.07</td>
<td>-2.99</td>
<td>0</td>
<td>-6 22904.73</td>
<td>-2.53649</td>
<td></td>
</tr>
<tr>
<td>404.09</td>
<td>92.79</td>
<td>324.09</td>
<td>-2.98</td>
<td>0</td>
<td>-6 22904.37</td>
<td>-2.49534</td>
<td></td>
</tr>
<tr>
<td>401.13</td>
<td>92.32</td>
<td>321.13</td>
<td>-2.96</td>
<td>0</td>
<td>-6 22903.99</td>
<td>-2.45559</td>
<td></td>
</tr>
<tr>
<td>398.18</td>
<td>91.82</td>
<td>318.18</td>
<td>-2.95</td>
<td>0</td>
<td>-6 22903.58</td>
<td>-2.41719</td>
<td></td>
</tr>
<tr>
<td>395.24</td>
<td>91.30</td>
<td>315.24</td>
<td>-2.93</td>
<td>0</td>
<td>-6 22903.15</td>
<td>-2.38008</td>
<td></td>
</tr>
<tr>
<td>392.32</td>
<td>90.77</td>
<td>312.32</td>
<td>-2.92</td>
<td>0</td>
<td>-6 22902.7</td>
<td>-2.34421</td>
<td></td>
</tr>
<tr>
<td>389.42</td>
<td>90.21</td>
<td>309.42</td>
<td>-2.90</td>
<td>0</td>
<td>-6 22902.23</td>
<td>-2.3095</td>
<td></td>
</tr>
<tr>
<td>386.54</td>
<td>89.64</td>
<td>306.54</td>
<td>-2.88</td>
<td>0</td>
<td>-6 22901.75</td>
<td>-2.27591</td>
<td></td>
</tr>
<tr>
<td>383.68</td>
<td>89.05</td>
<td>303.68</td>
<td>-2.86</td>
<td>0</td>
<td>-6 22901.25</td>
<td>-2.24337</td>
<td></td>
</tr>
<tr>
<td>380.84</td>
<td>88.45</td>
<td>300.84</td>
<td>-2.84</td>
<td>0</td>
<td>-6 22900.74</td>
<td>-2.21183</td>
<td></td>
</tr>
<tr>
<td>378.01</td>
<td>87.84</td>
<td>298.01</td>
<td>-2.82</td>
<td>0</td>
<td>-6 22900.22</td>
<td>-2.18123</td>
<td></td>
</tr>
<tr>
<td>375.21</td>
<td>87.22</td>
<td>295.21</td>
<td>-2.80</td>
<td>0</td>
<td>-6 22899.69</td>
<td>-2.15151</td>
<td></td>
</tr>
<tr>
<td>372.43</td>
<td>86.59</td>
<td>292.43</td>
<td>-2.78</td>
<td>0</td>
<td>-6 22899.15</td>
<td>-2.12264</td>
<td></td>
</tr>
<tr>
<td>369.67</td>
<td>85.95</td>
<td>289.67</td>
<td>-2.76</td>
<td>0</td>
<td>-6 22898.6</td>
<td>-2.09456</td>
<td></td>
</tr>
<tr>
<td>366.93</td>
<td>85.31</td>
<td>286.93</td>
<td>-2.74</td>
<td>0</td>
<td>-6 22898.05</td>
<td>-2.06722</td>
<td></td>
</tr>
<tr>
<td>364.21</td>
<td>84.66</td>
<td>284.21</td>
<td>-2.72</td>
<td>0</td>
<td>-6 22897.5</td>
<td>-2.0406</td>
<td></td>
</tr>
<tr>
<td>361.52</td>
<td>84.00</td>
<td>281.52</td>
<td>-2.69</td>
<td>0</td>
<td>-6 22896.94</td>
<td>-2.01465</td>
<td></td>
</tr>
<tr>
<td>358.85</td>
<td>83.35</td>
<td>278.85</td>
<td>-2.67</td>
<td>0</td>
<td>-6 22896.38</td>
<td>-1.98934</td>
<td></td>
</tr>
<tr>
<td>356.20</td>
<td>82.69</td>
<td>276.20</td>
<td>-2.65</td>
<td>0</td>
<td>-6 22895.82</td>
<td>-1.96463</td>
<td></td>
</tr>
<tr>
<td>353.57</td>
<td>82.02</td>
<td>273.57</td>
<td>-2.63</td>
<td>0</td>
<td>-6 22895.26</td>
<td>-1.9405</td>
<td></td>
</tr>
<tr>
<td>350.96</td>
<td>81.36</td>
<td>270.96</td>
<td>-2.61</td>
<td>0</td>
<td>-6 22894.7</td>
<td>-1.9169</td>
<td></td>
</tr>
<tr>
<td>348.38</td>
<td>80.69</td>
<td>268.38</td>
<td>-2.58</td>
<td>0</td>
<td>-6 22894.13</td>
<td>-1.89383</td>
<td></td>
</tr>
<tr>
<td>345.82</td>
<td>80.03</td>
<td>265.82</td>
<td>-2.56</td>
<td>0</td>
<td>-6 22893.57</td>
<td>-1.87124</td>
<td></td>
</tr>
<tr>
<td>343.28</td>
<td>79.36</td>
<td>263.28</td>
<td>-2.54</td>
<td>0</td>
<td>-6 22893.01</td>
<td>-1.84913</td>
<td></td>
</tr>
<tr>
<td>340.77</td>
<td>78.69</td>
<td>260.77</td>
<td>-2.52</td>
<td>0</td>
<td>-6 22892.45</td>
<td>-1.82746</td>
<td></td>
</tr>
<tr>
<td>338.28</td>
<td>78.03</td>
<td>258.28</td>
<td>-2.49</td>
<td>0</td>
<td>-6 22891.89</td>
<td>-1.80623</td>
<td></td>
</tr>
<tr>
<td>335.81</td>
<td>77.36</td>
<td>255.81</td>
<td>-2.47</td>
<td>0</td>
<td>-6 22891.33</td>
<td>-1.7854</td>
<td></td>
</tr>
<tr>
<td>333.36</td>
<td>76.70</td>
<td>253.36</td>
<td>-2.45</td>
<td>0</td>
<td>-6 22890.78</td>
<td>-1.76496</td>
<td></td>
</tr>
<tr>
<td>330.93</td>
<td>76.04</td>
<td>250.93</td>
<td>-2.43</td>
<td>0</td>
<td>-6 22890.23</td>
<td>-1.7449</td>
<td></td>
</tr>
<tr>
<td>328.53</td>
<td>75.38</td>
<td>248.53</td>
<td>-2.40</td>
<td>0</td>
<td>-6 22890.68</td>
<td>-1.7252</td>
<td></td>
</tr>
<tr>
<td>326.15</td>
<td>74.73</td>
<td>246.15</td>
<td>-2.38</td>
<td>0</td>
<td>-6 22889.13</td>
<td>-1.70584</td>
<td></td>
</tr>
<tr>
<td>323.79</td>
<td>74.07</td>
<td>243.79</td>
<td>-2.36</td>
<td>0</td>
<td>-6 22888.59</td>
<td>-1.68682</td>
<td></td>
</tr>
<tr>
<td>321.45</td>
<td>73.42</td>
<td>241.45</td>
<td>-2.34</td>
<td>0</td>
<td>-6 22888.05</td>
<td>-1.66812</td>
<td></td>
</tr>
<tr>
<td>319.13</td>
<td>72.77</td>
<td>239.13</td>
<td>-2.32</td>
<td>0</td>
<td>-6 22887.52</td>
<td>-1.64973</td>
<td></td>
</tr>
<tr>
<td>316.84</td>
<td>72.13</td>
<td>236.84</td>
<td>-2.29</td>
<td>0</td>
<td>-6 22886.98</td>
<td>-1.63164</td>
<td></td>
</tr>
<tr>
<td>314.57</td>
<td>71.49</td>
<td>234.57</td>
<td>-2.27</td>
<td>0</td>
<td>-6 22886.46</td>
<td>-1.61384</td>
<td></td>
</tr>
<tr>
<td>312.31</td>
<td>70.85</td>
<td>232.31</td>
<td>-2.25</td>
<td>0</td>
<td>-6 22885.93</td>
<td>-1.59631</td>
<td></td>
</tr>
<tr>
<td>310.08</td>
<td>70.21</td>
<td>230.08</td>
<td>-2.23</td>
<td>0</td>
<td>-6 22885.41</td>
<td>-1.57905</td>
<td></td>
</tr>
<tr>
<td>307.87</td>
<td>69.58</td>
<td>227.87</td>
<td>-2.21</td>
<td>0</td>
<td>-6 22884.89</td>
<td>-1.56206</td>
<td></td>
</tr>
<tr>
<td>303.51</td>
<td>68.33</td>
<td>223.51</td>
<td>-2.17</td>
<td>0</td>
<td>22884.38</td>
<td>-1.54532</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>--------</td>
<td>-------</td>
<td>---</td>
<td>-----------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>301.37</td>
<td>67.71</td>
<td>221.37</td>
<td>-2.15</td>
<td>0</td>
<td>22883.87</td>
<td>-1.52883</td>
<td></td>
</tr>
<tr>
<td>299.24</td>
<td>67.10</td>
<td>219.24</td>
<td>-2.13</td>
<td>0</td>
<td>22883.37</td>
<td>-1.51257</td>
<td></td>
</tr>
<tr>
<td>297.13</td>
<td>66.49</td>
<td>217.13</td>
<td>-2.11</td>
<td>0</td>
<td>22882.87</td>
<td>-1.49654</td>
<td></td>
</tr>
<tr>
<td>295.04</td>
<td>65.88</td>
<td>215.04</td>
<td>-2.09</td>
<td>0</td>
<td>22882.37</td>
<td>-1.48075</td>
<td></td>
</tr>
<tr>
<td>292.97</td>
<td>65.28</td>
<td>212.97</td>
<td>-2.07</td>
<td>0</td>
<td>22881.88</td>
<td>-1.46517</td>
<td></td>
</tr>
<tr>
<td>290.93</td>
<td>64.68</td>
<td>210.93</td>
<td>-2.05</td>
<td>0</td>
<td>22881.39</td>
<td>-1.4498</td>
<td></td>
</tr>
<tr>
<td>288.90</td>
<td>64.08</td>
<td>208.90</td>
<td>-2.03</td>
<td>0</td>
<td>22880.91</td>
<td>-1.43464</td>
<td></td>
</tr>
<tr>
<td>286.89</td>
<td>63.49</td>
<td>206.89</td>
<td>-2.01</td>
<td>0</td>
<td>22880.43</td>
<td>-1.41969</td>
<td></td>
</tr>
<tr>
<td>284.90</td>
<td>62.91</td>
<td>204.90</td>
<td>-1.99</td>
<td>0</td>
<td>22879.95</td>
<td>-1.40493</td>
<td></td>
</tr>
<tr>
<td>282.92</td>
<td>62.33</td>
<td>202.92</td>
<td>-1.97</td>
<td>0</td>
<td>22879.48</td>
<td>-1.39037</td>
<td></td>
</tr>
<tr>
<td>280.97</td>
<td>61.75</td>
<td>200.97</td>
<td>-1.95</td>
<td>0</td>
<td>22879.01</td>
<td>-1.37599</td>
<td></td>
</tr>
<tr>
<td>279.04</td>
<td>61.18</td>
<td>199.04</td>
<td>-1.93</td>
<td>0</td>
<td>22878.55</td>
<td>-1.3618</td>
<td></td>
</tr>
<tr>
<td>277.12</td>
<td>60.61</td>
<td>197.12</td>
<td>-1.92</td>
<td>0</td>
<td>22878.09</td>
<td>-1.34779</td>
<td></td>
</tr>
<tr>
<td>275.22</td>
<td>60.04</td>
<td>195.22</td>
<td>-1.90</td>
<td>0</td>
<td>22877.63</td>
<td>-1.33396</td>
<td></td>
</tr>
<tr>
<td>273.34</td>
<td>59.49</td>
<td>193.34</td>
<td>-1.88</td>
<td>0</td>
<td>22877.18</td>
<td>-1.3203</td>
<td></td>
</tr>
<tr>
<td>271.48</td>
<td>58.93</td>
<td>191.48</td>
<td>-1.86</td>
<td>0</td>
<td>22876.74</td>
<td>-1.30681</td>
<td></td>
</tr>
<tr>
<td>269.64</td>
<td>58.38</td>
<td>189.64</td>
<td>-1.84</td>
<td>0</td>
<td>22876.29</td>
<td>-1.29348</td>
<td></td>
</tr>
<tr>
<td>267.81</td>
<td>57.83</td>
<td>187.81</td>
<td>-1.83</td>
<td>0</td>
<td>22875.86</td>
<td>-1.28032</td>
<td></td>
</tr>
<tr>
<td>266.01</td>
<td>57.29</td>
<td>186.01</td>
<td>-1.81</td>
<td>0</td>
<td>22875.42</td>
<td>-1.26731</td>
<td></td>
</tr>
<tr>
<td>264.21</td>
<td>56.76</td>
<td>184.21</td>
<td>-1.79</td>
<td>0</td>
<td>22874.99</td>
<td>-1.25446</td>
<td></td>
</tr>
<tr>
<td>262.44</td>
<td>56.22</td>
<td>182.44</td>
<td>-1.77</td>
<td>0</td>
<td>22874.56</td>
<td>-1.24177</td>
<td></td>
</tr>
<tr>
<td>260.68</td>
<td>55.70</td>
<td>180.68</td>
<td>-1.76</td>
<td>0</td>
<td>22874.14</td>
<td>-1.22922</td>
<td></td>
</tr>
<tr>
<td>258.94</td>
<td>55.17</td>
<td>178.94</td>
<td>-1.74</td>
<td>0</td>
<td>22873.72</td>
<td>-1.21682</td>
<td></td>
</tr>
<tr>
<td>257.22</td>
<td>54.65</td>
<td>177.22</td>
<td>-1.72</td>
<td>0</td>
<td>22873.31</td>
<td>-1.20457</td>
<td></td>
</tr>
<tr>
<td>255.51</td>
<td>54.14</td>
<td>175.51</td>
<td>-1.71</td>
<td>0</td>
<td>22872.9</td>
<td>-1.19245</td>
<td></td>
</tr>
<tr>
<td>253.82</td>
<td>53.63</td>
<td>173.82</td>
<td>-1.69</td>
<td>0</td>
<td>22872.49</td>
<td>-1.18048</td>
<td></td>
</tr>
<tr>
<td>252.15</td>
<td>53.12</td>
<td>172.15</td>
<td>-1.67</td>
<td>0</td>
<td>22872.09</td>
<td>-1.16864</td>
<td></td>
</tr>
<tr>
<td>250.49</td>
<td>52.62</td>
<td>170.49</td>
<td>-1.66</td>
<td>0</td>
<td>22871.69</td>
<td>-1.15694</td>
<td></td>
</tr>
<tr>
<td>248.85</td>
<td>52.13</td>
<td>168.85</td>
<td>-1.64</td>
<td>0</td>
<td>22871.29</td>
<td>-1.14537</td>
<td></td>
</tr>
<tr>
<td>247.22</td>
<td>51.63</td>
<td>167.22</td>
<td>-1.63</td>
<td>0</td>
<td>22870.9</td>
<td>-1.13393</td>
<td></td>
</tr>
<tr>
<td>245.61</td>
<td>51.15</td>
<td>165.61</td>
<td>-1.61</td>
<td>0</td>
<td>22870.51</td>
<td>-1.12262</td>
<td></td>
</tr>
<tr>
<td>244.01</td>
<td>50.66</td>
<td>164.01</td>
<td>-1.60</td>
<td>0</td>
<td>22870.13</td>
<td>-1.11144</td>
<td></td>
</tr>
<tr>
<td>242.43</td>
<td>50.18</td>
<td>162.43</td>
<td>-1.58</td>
<td>0</td>
<td>22869.75</td>
<td>-1.10038</td>
<td></td>
</tr>
<tr>
<td>240.87</td>
<td>49.71</td>
<td>160.87</td>
<td>-1.57</td>
<td>0</td>
<td>22869.37</td>
<td>-1.08944</td>
<td></td>
</tr>
<tr>
<td>239.32</td>
<td>49.23</td>
<td>159.32</td>
<td>-1.55</td>
<td>0</td>
<td>22868.99</td>
<td>-1.07862</td>
<td></td>
</tr>
<tr>
<td>237.78</td>
<td>48.77</td>
<td>157.78</td>
<td>-1.54</td>
<td>0</td>
<td>22868.63</td>
<td>-1.06792</td>
<td></td>
</tr>
<tr>
<td>236.26</td>
<td>48.30</td>
<td>156.26</td>
<td>-1.52</td>
<td>0</td>
<td>22868.26</td>
<td>-1.05734</td>
<td></td>
</tr>
<tr>
<td>234.76</td>
<td>47.85</td>
<td>154.76</td>
<td>-1.51</td>
<td>0</td>
<td>22867.89</td>
<td>-1.04687</td>
<td></td>
</tr>
<tr>
<td>233.27</td>
<td>47.39</td>
<td>153.27</td>
<td>-1.49</td>
<td>0</td>
<td>22867.54</td>
<td>-1.03652</td>
<td></td>
</tr>
<tr>
<td>231.79</td>
<td>46.94</td>
<td>151.79</td>
<td>-1.48</td>
<td>0</td>
<td>22867.18</td>
<td>-1.02627</td>
<td></td>
</tr>
<tr>
<td>230.33</td>
<td>46.49</td>
<td>150.33</td>
<td>-1.46</td>
<td>0</td>
<td>22866.83</td>
<td>-1.01614</td>
<td></td>
</tr>
<tr>
<td>228.88</td>
<td>46.05</td>
<td>148.88</td>
<td>-1.45</td>
<td>0</td>
<td>22866.48</td>
<td>-1.00611</td>
<td></td>
</tr>
<tr>
<td>227.44</td>
<td>45.61</td>
<td>147.44</td>
<td>-1.43</td>
<td>0</td>
<td>22866.13</td>
<td>-0.9962</td>
<td></td>
</tr>
<tr>
<td>226.02</td>
<td>45.18</td>
<td>146.02</td>
<td>-1.42</td>
<td>0</td>
<td>22865.79</td>
<td>-0.98638</td>
<td></td>
</tr>
<tr>
<td>224.62</td>
<td>44.75</td>
<td>144.62</td>
<td>-1.41</td>
<td>0</td>
<td>22865.45</td>
<td>-0.97667</td>
<td></td>
</tr>
<tr>
<td>223.22</td>
<td>44.32</td>
<td>143.22</td>
<td>-1.39</td>
<td>0</td>
<td>22865.11</td>
<td>-0.96707</td>
<td></td>
</tr>
<tr>
<td>221.84</td>
<td>43.90</td>
<td>141.84</td>
<td>-1.38</td>
<td>0</td>
<td>22864.78</td>
<td>-0.95756</td>
<td></td>
</tr>
<tr>
<td>220.47</td>
<td>43.48</td>
<td>140.47</td>
<td>-1.37</td>
<td>0</td>
<td>-6</td>
<td>22864.45</td>
<td>-0.94815</td>
</tr>
<tr>
<td>219.12</td>
<td>43.06</td>
<td>139.12</td>
<td>-1.35</td>
<td>0</td>
<td>-6</td>
<td>22864.12</td>
<td>-0.93885</td>
</tr>
<tr>
<td>217.78</td>
<td>42.65</td>
<td>137.78</td>
<td>-1.34</td>
<td>0</td>
<td>-6</td>
<td>22863.8</td>
<td>-0.92964</td>
</tr>
<tr>
<td>216.45</td>
<td>42.25</td>
<td>136.45</td>
<td>-1.33</td>
<td>0</td>
<td>-6</td>
<td>22863.48</td>
<td>-0.92052</td>
</tr>
<tr>
<td>215.14</td>
<td>41.84</td>
<td>135.14</td>
<td>-1.32</td>
<td>0</td>
<td>-6</td>
<td>22863.16</td>
<td>-0.9115</td>
</tr>
<tr>
<td>213.83</td>
<td>41.44</td>
<td>133.83</td>
<td>-1.30</td>
<td>0</td>
<td>-6</td>
<td>22862.84</td>
<td>-0.90257</td>
</tr>
<tr>
<td>212.54</td>
<td>41.05</td>
<td>132.54</td>
<td>-1.29</td>
<td>0</td>
<td>-6</td>
<td>22862.53</td>
<td>-0.89374</td>
</tr>
<tr>
<td>211.27</td>
<td>40.65</td>
<td>131.27</td>
<td>-1.28</td>
<td>0</td>
<td>-6</td>
<td>22862.22</td>
<td>-0.885</td>
</tr>
<tr>
<td>210.00</td>
<td>40.26</td>
<td>130.00</td>
<td>-1.27</td>
<td>0</td>
<td>-6</td>
<td>22861.92</td>
<td>-0.87634</td>
</tr>
<tr>
<td>208.75</td>
<td>39.88</td>
<td>128.75</td>
<td>-1.25</td>
<td>0</td>
<td>-6</td>
<td>22861.62</td>
<td>-0.86778</td>
</tr>
<tr>
<td>207.51</td>
<td>39.50</td>
<td>127.51</td>
<td>-1.24</td>
<td>0</td>
<td>-6</td>
<td>22861.32</td>
<td>-0.8593</td>
</tr>
<tr>
<td>206.28</td>
<td>39.12</td>
<td>126.28</td>
<td>-1.23</td>
<td>0</td>
<td>-6</td>
<td>22861.02</td>
<td>-0.85091</td>
</tr>
<tr>
<td>205.06</td>
<td>38.75</td>
<td>125.06</td>
<td>-1.22</td>
<td>0</td>
<td>-6</td>
<td>22860.72</td>
<td>-0.8426</td>
</tr>
<tr>
<td>203.86</td>
<td>38.37</td>
<td>123.86</td>
<td>-1.21</td>
<td>0</td>
<td>-6</td>
<td>22860.43</td>
<td>-0.83438</td>
</tr>
<tr>
<td>202.66</td>
<td>38.01</td>
<td>122.66</td>
<td>-1.19</td>
<td>0</td>
<td>-6</td>
<td>22860.14</td>
<td>-0.82625</td>
</tr>
<tr>
<td>201.48</td>
<td>37.64</td>
<td>121.48</td>
<td>-1.18</td>
<td>0</td>
<td>-6</td>
<td>22859.86</td>
<td>-0.81819</td>
</tr>
<tr>
<td>200.31</td>
<td>37.28</td>
<td>120.31</td>
<td>-1.17</td>
<td>0</td>
<td>-6</td>
<td>22859.58</td>
<td>-0.81022</td>
</tr>
<tr>
<td>199.15</td>
<td>36.92</td>
<td>119.15</td>
<td>-1.16</td>
<td>0</td>
<td>-6</td>
<td>22859.29</td>
<td>-0.80233</td>
</tr>
<tr>
<td>198.00</td>
<td>36.57</td>
<td>118.00</td>
<td>-1.15</td>
<td>0</td>
<td>-6</td>
<td>22859.02</td>
<td>-0.79451</td>
</tr>
<tr>
<td>196.86</td>
<td>36.22</td>
<td>116.86</td>
<td>-1.14</td>
<td>0</td>
<td>-6</td>
<td>22858.74</td>
<td>-0.78678</td>
</tr>
<tr>
<td>195.74</td>
<td>35.87</td>
<td>115.74</td>
<td>-1.13</td>
<td>0</td>
<td>-6</td>
<td>22858.47</td>
<td>-0.77912</td>
</tr>
<tr>
<td>194.62</td>
<td>35.53</td>
<td>114.62</td>
<td>-1.12</td>
<td>0</td>
<td>-6</td>
<td>22858.2</td>
<td>-0.77154</td>
</tr>
<tr>
<td>193.52</td>
<td>35.19</td>
<td>113.52</td>
<td>-1.10</td>
<td>0</td>
<td>-6</td>
<td>22857.93</td>
<td>-0.76404</td>
</tr>
<tr>
<td>192.42</td>
<td>34.85</td>
<td>112.42</td>
<td>-1.09</td>
<td>0</td>
<td>-6</td>
<td>22857.67</td>
<td>-0.75661</td>
</tr>
<tr>
<td>191.34</td>
<td>34.52</td>
<td>111.34</td>
<td>-1.08</td>
<td>0</td>
<td>-6</td>
<td>22857.41</td>
<td>-0.74926</td>
</tr>
<tr>
<td>190.27</td>
<td>34.18</td>
<td>110.27</td>
<td>-1.07</td>
<td>0</td>
<td>-6</td>
<td>22857.15</td>
<td>-0.74198</td>
</tr>
<tr>
<td>189.20</td>
<td>33.86</td>
<td>109.20</td>
<td>-1.06</td>
<td>0</td>
<td>-6</td>
<td>22856.89</td>
<td>-0.73477</td>
</tr>
<tr>
<td>188.15</td>
<td>33.53</td>
<td>108.15</td>
<td>-1.05</td>
<td>0</td>
<td>-6</td>
<td>22856.64</td>
<td>-0.72763</td>
</tr>
<tr>
<td>187.11</td>
<td>33.21</td>
<td>107.11</td>
<td>-1.04</td>
<td>0</td>
<td>-6</td>
<td>22856.38</td>
<td>-0.72057</td>
</tr>
<tr>
<td>186.07</td>
<td>32.89</td>
<td>106.07</td>
<td>-1.03</td>
<td>0</td>
<td>-6</td>
<td>22856.13</td>
<td>-0.71357</td>
</tr>
<tr>
<td>185.05</td>
<td>32.57</td>
<td>105.05</td>
<td>-1.02</td>
<td>0</td>
<td>-6</td>
<td>22855.89</td>
<td>-0.70665</td>
</tr>
<tr>
<td>184.04</td>
<td>32.26</td>
<td>104.04</td>
<td>-1.01</td>
<td>0</td>
<td>-6</td>
<td>22855.64</td>
<td>-0.69979</td>
</tr>
<tr>
<td>183.04</td>
<td>31.95</td>
<td>103.04</td>
<td>-1.00</td>
<td>0</td>
<td>-6</td>
<td>22855.4</td>
<td>-0.693</td>
</tr>
<tr>
<td>182.04</td>
<td>31.64</td>
<td>102.04</td>
<td>-0.99</td>
<td>0</td>
<td>-5</td>
<td>22855.16</td>
<td>-0.68628</td>
</tr>
<tr>
<td>181.06</td>
<td>31.34</td>
<td>101.06</td>
<td>-0.98</td>
<td>0</td>
<td>-4</td>
<td>22854.92</td>
<td>-0.67963</td>
</tr>
<tr>
<td>180.09</td>
<td>31.04</td>
<td>100.09</td>
<td>-0.97</td>
<td>0</td>
<td>-3</td>
<td>22854.69</td>
<td>-0.67304</td>
</tr>
<tr>
<td>179.12</td>
<td>30.74</td>
<td>99.12</td>
<td>-0.96</td>
<td>0</td>
<td>-2</td>
<td>22854.45</td>
<td>-0.66652</td>
</tr>
<tr>
<td>178.16</td>
<td>30.45</td>
<td>98.16</td>
<td>-0.96</td>
<td>0</td>
<td>-1</td>
<td>22854.22</td>
<td>-0.66006</td>
</tr>
<tr>
<td>177.22</td>
<td>30.15</td>
<td>97.22</td>
<td>-0.95</td>
<td>0</td>
<td>0</td>
<td>22853.99</td>
<td>-0.65366</td>
</tr>
<tr>
<td>176.28</td>
<td>29.86</td>
<td>96.28</td>
<td>-0.94</td>
<td>0</td>
<td>1</td>
<td>22853.77</td>
<td>-0.64124</td>
</tr>
<tr>
<td>175.35</td>
<td>29.55</td>
<td>95.35</td>
<td>-0.93</td>
<td>0</td>
<td>0</td>
<td>22853.54</td>
<td>-0.61817</td>
</tr>
<tr>
<td>174.43</td>
<td>29.23</td>
<td>94.33</td>
<td>-0.92</td>
<td>0</td>
<td>0</td>
<td>22853.32</td>
<td>-0.60661</td>
</tr>
<tr>
<td>173.52</td>
<td>28.93</td>
<td>93.25</td>
<td>-0.91</td>
<td>0</td>
<td>0</td>
<td>22853.1</td>
<td>-0.60147</td>
</tr>
<tr>
<td>172.62</td>
<td>28.62</td>
<td>92.16</td>
<td>-0.9</td>
<td>0</td>
<td>0</td>
<td>22852.87</td>
<td>-0.59771</td>
</tr>
<tr>
<td>171.73</td>
<td>28.32</td>
<td>91.07</td>
<td>-0.89</td>
<td>0</td>
<td>0</td>
<td>22852.64</td>
<td>-0.59401</td>
</tr>
<tr>
<td>170.85</td>
<td>28.03</td>
<td>89.85</td>
<td>-0.88</td>
<td>0</td>
<td>0</td>
<td>22852.42</td>
<td>-0.59012</td>
</tr>
<tr>
<td>169.97</td>
<td>27.74</td>
<td>88.67</td>
<td>-0.87</td>
<td>0</td>
<td>0</td>
<td>22852.19</td>
<td>-0.586</td>
</tr>
<tr>
<td>169.11</td>
<td>27.46</td>
<td>87.49</td>
<td>-0.87</td>
<td>0</td>
<td>0</td>
<td>22851.96</td>
<td>-0.58168</td>
</tr>
<tr>
<td>168.25</td>
<td>27.18</td>
<td>87.25</td>
<td>-0.86</td>
<td>0</td>
<td>0</td>
<td>22851.74</td>
<td>-0.5772</td>
</tr>
<tr>
<td>167.40</td>
<td>26.90</td>
<td>86.40</td>
<td>-0.85</td>
<td>0</td>
<td>0</td>
<td>22851.52</td>
<td>-0.5726</td>
</tr>
<tr>
<td>166.57</td>
<td>26.63</td>
<td>85.57</td>
<td>-0.84</td>
<td>0</td>
<td>0</td>
<td>22851.3</td>
<td>-0.56791</td>
</tr>
<tr>
<td>165.74</td>
<td>26.37</td>
<td>84.74</td>
<td>-0.83</td>
<td>0</td>
<td>0</td>
<td>22851.09</td>
<td>-0.56315</td>
</tr>
<tr>
<td>164.91</td>
<td>26.10</td>
<td>83.91</td>
<td>-0.82</td>
<td>0</td>
<td>0</td>
<td>22850.88</td>
<td>-0.55834</td>
</tr>
<tr>
<td>164.10</td>
<td>25.85</td>
<td>83.10</td>
<td>-0.81</td>
<td>0</td>
<td>0</td>
<td>22850.67</td>
<td>-0.5535</td>
</tr>
<tr>
<td>163.30</td>
<td>25.60</td>
<td>83.30</td>
<td>-0.80</td>
<td>0</td>
<td>-1</td>
<td>22850.47</td>
<td>-0.55472</td>
</tr>
<tr>
<td>162.50</td>
<td>25.37</td>
<td>82.50</td>
<td>-0.80</td>
<td>0</td>
<td>-1</td>
<td>22850.26</td>
<td>-0.56664</td>
</tr>
<tr>
<td>161.71</td>
<td>25.14</td>
<td>81.71</td>
<td>-0.79</td>
<td>0</td>
<td>-1</td>
<td>22850.07</td>
<td>-0.56711</td>
</tr>
<tr>
<td>160.93</td>
<td>24.92</td>
<td>80.93</td>
<td>-0.78</td>
<td>0</td>
<td>-1</td>
<td>22849.88</td>
<td>-0.56123</td>
</tr>
<tr>
<td>160.16</td>
<td>24.71</td>
<td>80.16</td>
<td>-0.77</td>
<td>0</td>
<td>-1</td>
<td>22849.69</td>
<td>-0.55404</td>
</tr>
<tr>
<td>159.39</td>
<td>24.49</td>
<td>79.39</td>
<td>-0.77</td>
<td>0</td>
<td>-1</td>
<td>22849.51</td>
<td>-0.54684</td>
</tr>
<tr>
<td>158.63</td>
<td>24.27</td>
<td>78.63</td>
<td>-0.76</td>
<td>0</td>
<td>0</td>
<td>22849.33</td>
<td>-0.53993</td>
</tr>
<tr>
<td>157.88</td>
<td>24.05</td>
<td>77.88</td>
<td>-0.75</td>
<td>0</td>
<td>0</td>
<td>22849.16</td>
<td>-0.53333</td>
</tr>
<tr>
<td>157.14</td>
<td>23.83</td>
<td>77.14</td>
<td>-0.74</td>
<td>0</td>
<td>0</td>
<td>22848.99</td>
<td>-0.527</td>
</tr>
<tr>
<td>156.40</td>
<td>23.61</td>
<td>76.40</td>
<td>-0.74</td>
<td>0</td>
<td>0</td>
<td>22848.83</td>
<td>-0.52092</td>
</tr>
<tr>
<td>155.67</td>
<td>23.40</td>
<td>75.67</td>
<td>-0.73</td>
<td>0</td>
<td>0</td>
<td>22848.66</td>
<td>-0.51504</td>
</tr>
<tr>
<td>154.94</td>
<td>23.18</td>
<td>74.94</td>
<td>-0.72</td>
<td>0</td>
<td>0</td>
<td>22848.5</td>
<td>-0.50934</td>
</tr>
<tr>
<td>154.22</td>
<td>22.97</td>
<td>74.22</td>
<td>-0.72</td>
<td>0</td>
<td>0</td>
<td>22848.34</td>
<td>-0.5038</td>
</tr>
<tr>
<td>153.51</td>
<td>22.75</td>
<td>73.51</td>
<td>-0.71</td>
<td>0</td>
<td>0</td>
<td>22848.17</td>
<td>-0.49839</td>
</tr>
<tr>
<td>152.81</td>
<td>22.54</td>
<td>72.81</td>
<td>-0.71</td>
<td>0</td>
<td>0</td>
<td>22848.01</td>
<td>-0.49311</td>
</tr>
<tr>
<td>152.11</td>
<td>22.33</td>
<td>72.11</td>
<td>-0.70</td>
<td>0</td>
<td>0</td>
<td>22847.85</td>
<td>-0.48794</td>
</tr>
<tr>
<td>151.42</td>
<td>22.12</td>
<td>71.42</td>
<td>-0.69</td>
<td>0</td>
<td>0</td>
<td>22847.69</td>
<td>-0.48287</td>
</tr>
<tr>
<td>150.73</td>
<td>21.91</td>
<td>70.73</td>
<td>-0.69</td>
<td>0</td>
<td>0</td>
<td>22847.53</td>
<td>-0.47789</td>
</tr>
<tr>
<td>150.05</td>
<td>21.71</td>
<td>70.05</td>
<td>-0.68</td>
<td>0</td>
<td>0</td>
<td>22847.38</td>
<td>-0.473</td>
</tr>
<tr>
<td>149.38</td>
<td>21.50</td>
<td>69.38</td>
<td>-0.67</td>
<td>0</td>
<td>0</td>
<td>22847.22</td>
<td>-0.46819</td>
</tr>
<tr>
<td>148.71</td>
<td>21.30</td>
<td>68.71</td>
<td>-0.67</td>
<td>0</td>
<td>0</td>
<td>22847.06</td>
<td>-0.46345</td>
</tr>
<tr>
<td>148.05</td>
<td>21.09</td>
<td>68.05</td>
<td>-0.66</td>
<td>0</td>
<td>0</td>
<td>22846.91</td>
<td>-0.45878</td>
</tr>
<tr>
<td>147.39</td>
<td>20.89</td>
<td>67.39</td>
<td>-0.65</td>
<td>0</td>
<td>0</td>
<td>22846.75</td>
<td>-0.45419</td>
</tr>
<tr>
<td>146.74</td>
<td>20.70</td>
<td>66.74</td>
<td>-0.65</td>
<td>0</td>
<td>0</td>
<td>22846.6</td>
<td>-0.44965</td>
</tr>
<tr>
<td>146.10</td>
<td>20.50</td>
<td>66.10</td>
<td>-0.64</td>
<td>0</td>
<td>0</td>
<td>22846.45</td>
<td>-0.44518</td>
</tr>
<tr>
<td>145.47</td>
<td>20.30</td>
<td>65.47</td>
<td>-0.64</td>
<td>0</td>
<td>0</td>
<td>22846.3</td>
<td>-0.44077</td>
</tr>
<tr>
<td>144.83</td>
<td>20.11</td>
<td>64.83</td>
<td>-0.63</td>
<td>0</td>
<td>0</td>
<td>22846.14</td>
<td>-0.43641</td>
</tr>
<tr>
<td>144.21</td>
<td>19.92</td>
<td>64.21</td>
<td>-0.62</td>
<td>0</td>
<td>0</td>
<td>22845.99</td>
<td>-0.43211</td>
</tr>
<tr>
<td>143.59</td>
<td>19.73</td>
<td>63.59</td>
<td>-0.62</td>
<td>0</td>
<td>0</td>
<td>22845.85</td>
<td>-0.42786</td>
</tr>
<tr>
<td>142.98</td>
<td>19.54</td>
<td>62.98</td>
<td>-0.61</td>
<td>0</td>
<td>0</td>
<td>22845.7</td>
<td>-0.42366</td>
</tr>
<tr>
<td>142.37</td>
<td>19.35</td>
<td>62.37</td>
<td>-0.61</td>
<td>0</td>
<td>0</td>
<td>22845.55</td>
<td>-0.4195</td>
</tr>
<tr>
<td>141.77</td>
<td>19.16</td>
<td>61.77</td>
<td>-0.60</td>
<td>0</td>
<td>0</td>
<td>22845.41</td>
<td>-0.4154</td>
</tr>
<tr>
<td>141.18</td>
<td>18.98</td>
<td>61.18</td>
<td>-0.60</td>
<td>0</td>
<td>0</td>
<td>22845.27</td>
<td>-0.41134</td>
</tr>
<tr>
<td>140.59</td>
<td>18.80</td>
<td>60.59</td>
<td>-0.59</td>
<td>0</td>
<td>0</td>
<td>22845.13</td>
<td>-0.40733</td>
</tr>
<tr>
<td>140.00</td>
<td>18.62</td>
<td>60.00</td>
<td>-0.58</td>
<td>0</td>
<td>0</td>
<td>22844.99</td>
<td>-0.40336</td>
</tr>
<tr>
<td>139.42</td>
<td>18.44</td>
<td>59.42</td>
<td>-0.58</td>
<td>0</td>
<td>0</td>
<td>22844.85</td>
<td>-0.39943</td>
</tr>
<tr>
<td>138.85</td>
<td>18.26</td>
<td>58.85</td>
<td>-0.57</td>
<td>0</td>
<td>0</td>
<td>22844.71</td>
<td>-0.39554</td>
</tr>
<tr>
<td>138.28</td>
<td>18.09</td>
<td>58.28</td>
<td>-0.57</td>
<td>0</td>
<td>0</td>
<td>22844.57</td>
<td>-0.3917</td>
</tr>
<tr>
<td>137.72</td>
<td>17.91</td>
<td>57.72</td>
<td>-0.56</td>
<td>0</td>
<td>0</td>
<td>22844.44</td>
<td>-0.38789</td>
</tr>
<tr>
<td>137.17</td>
<td>17.74</td>
<td>57.17</td>
<td>-0.56</td>
<td>0</td>
<td>0</td>
<td>22844.3</td>
<td>-0.38413</td>
</tr>
<tr>
<td>136.62</td>
<td>17.57</td>
<td>56.62</td>
<td>-0.55</td>
<td>0</td>
<td>0</td>
<td>22844.17</td>
<td>-0.3804</td>
</tr>
<tr>
<td>136.07</td>
<td>17.40</td>
<td>56.07</td>
<td>-0.55</td>
<td>0</td>
<td>0</td>
<td>22844.04</td>
<td>-0.37671</td>
</tr>
<tr>
<td>135.53</td>
<td>17.23</td>
<td>55.53</td>
<td>-0.54</td>
<td>0</td>
<td>0 22843.91</td>
<td>-0.37306</td>
<td></td>
</tr>
<tr>
<td>134.99</td>
<td>17.07</td>
<td>54.99</td>
<td>-0.54</td>
<td>0</td>
<td>0 22843.78</td>
<td>-0.36945</td>
<td></td>
</tr>
<tr>
<td>134.46</td>
<td>16.90</td>
<td>54.46</td>
<td>-0.53</td>
<td>0</td>
<td>0 22843.65</td>
<td>-0.36587</td>
<td></td>
</tr>
<tr>
<td>133.94</td>
<td>16.74</td>
<td>53.94</td>
<td>-0.53</td>
<td>0</td>
<td>0 22843.52</td>
<td>-0.36233</td>
<td></td>
</tr>
<tr>
<td>133.42</td>
<td>16.58</td>
<td>53.42</td>
<td>-0.52</td>
<td>0</td>
<td>0 22843.4</td>
<td>-0.35883</td>
<td></td>
</tr>
<tr>
<td>132.90</td>
<td>16.42</td>
<td>52.90</td>
<td>-0.51</td>
<td>0</td>
<td>0 22843.27</td>
<td>-0.35536</td>
<td></td>
</tr>
<tr>
<td>132.39</td>
<td>16.26</td>
<td>52.39</td>
<td>-0.51</td>
<td>0</td>
<td>0 22843.15</td>
<td>-0.35192</td>
<td></td>
</tr>
<tr>
<td>131.89</td>
<td>16.10</td>
<td>51.89</td>
<td>-0.51</td>
<td>0</td>
<td>0 22843.03</td>
<td>-0.34852</td>
<td></td>
</tr>
<tr>
<td>131.39</td>
<td>15.95</td>
<td>51.39</td>
<td>-0.50</td>
<td>0</td>
<td>0 22842.91</td>
<td>-0.34515</td>
<td></td>
</tr>
<tr>
<td>130.89</td>
<td>15.79</td>
<td>50.89</td>
<td>-0.50</td>
<td>0</td>
<td>0 22842.79</td>
<td>-0.34181</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX C

Physical Properties Used
Panel Thicknesses Used
Table C.1
Physical Properties Used

Properties of Hercules [38]
(Initial prepreg resin mass fraction = 0.42)

<table>
<thead>
<tr>
<th>Property</th>
<th>Resin</th>
<th>Fiber</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thermal Conductivity (W/mK)</td>
<td>0.1670</td>
<td>26.000</td>
</tr>
<tr>
<td>Density (kg/m³)</td>
<td>1260.0</td>
<td>1790.0</td>
</tr>
<tr>
<td>Specific Heat Capacity (kJ/(kg.K))</td>
<td>1.26</td>
<td>0.712</td>
</tr>
<tr>
<td>Heat of Reaction (J/g)</td>
<td>474</td>
<td></td>
</tr>
</tbody>
</table>

The properties of Hercules AS4/3501-6 Prepreg and Mochburg CW 1850 Thermal Fiber Bleeder Cloth shown in Table C.1 were used in the simulator calculations.
Table C.2
Panel Thicknesses Used

<table>
<thead>
<tr>
<th>Layer</th>
<th># of nodes</th>
<th>32-ply</th>
<th>128-ply</th>
<th>256-ply</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td># of plies</td>
<td>thickness</td>
<td># of plies</td>
</tr>
<tr>
<td>breather</td>
<td>1</td>
<td>1</td>
<td>0.01292</td>
<td>1</td>
</tr>
<tr>
<td>bleeder</td>
<td>2</td>
<td>3</td>
<td>0.00153</td>
<td>12</td>
</tr>
<tr>
<td>laminate</td>
<td>16</td>
<td>32</td>
<td>0.00173</td>
<td>128</td>
</tr>
<tr>
<td>bleeder</td>
<td>2</td>
<td>3</td>
<td>0.00153</td>
<td>12</td>
</tr>
<tr>
<td>breather</td>
<td>1</td>
<td>1</td>
<td>0.01292</td>
<td>1</td>
</tr>
</tbody>
</table>