8-2011

PREDICTING THE THERMAL PERFORMANCE FOR THE MULTI- OBJECTIVE VEHICLE UNDERHOOD PACKING OPTIMIZATION PROBLEM

Ravi teja Katragadda
Clemson University, ravitejakatragadda@gmail.com

Follow this and additional works at: https://tigerprints.clemson.edu/all_theses

Part of the [Engineering Commons](#)

Recommended Citation

Katragadda, Ravi teja, "PREDICTING THE THERMAL PERFORMANCE FOR THE MULTI-OBJECTIVE VEHICLE UNDERHOOD PACKING OPTIMIZATION PROBLEM" (2011). *All Theses*. 1192.
https://tigerprints.clemson.edu/all_theses/1192

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact kokeefe@clemson.edu.

PREDICTING THE THERMAL PERFORMANCE FOR THE MULTI-OBJECTIVE
VEHICLE UNDERHOOD PACKING OPTIMIZATION PROBLEM

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Mechanical Engineering

by
Ravi Teja Katragadda
May 2012

Accepted by:
Dr. Georges M. Fadel, Committee Chair
Dr. Richard S. Miller
Dr. Rui J. Qiao

ABSTRACT

The ever increasing demands towards improvement in vehicle performance and passenger comfort have led the automotive manufacturers to further enhance the design in the early stages of the vehicle development process. Though, these design changes enhance the overall vehicle performance to an extent, the placement of these components under the car hood also plays a vital role in increasing the vehicle performance. The study of the placement of these components in an automobile underhood forms a 3-Dimensional packaging problem. In the past, a study on the automobile underhood packaging problem was conducted and a multi objective optimization routine with three objectives namely, minimizing center of gravity height, maximizing vehicle maintainability and maximizing survivability has been setup to determine the optimal locations of the underhood components. Also in the past, another study was conducted which asserted the need for the inclusion of the thermal performance of the vehicle Underhood as an objective to the optimization routine proposed earlier.

This study makes an assessment of the several available thermal analyses that are performed on the automotive underhood to evaluate the thermal objective. The assessment conducted in this study indicates that these thermal analyses, when included into the rigorous optimization routine, increase the computationally expense by a large amount. Thus an approximate thermal model is presented to evaluate the thermal performance of the vehicle underhood, which when included as an objective into the

optimization routine, does not make it computationally expensive. The approximate thermal model is a Neural Network approximation of the CFD analysis conducted over the automotive underhood. The results obtained from the neural network are compared with the CFD results, showing good agreement. The Neural Network model is now included in the multi objective optimization routine and the results are obtained. A non-deterministic evolutionary multi-objective algorithm (AMGA-2) is used to perform the optimization process.

DEDICATION

This thesis is a dedication to my parents for their love and support.

ACKNOWLEDGMENTS

This thesis would not have been possible without the support of many people. I owe my deepest gratitude to my advisor Dr. Georges M. Fadel for his constant assistance and encouragement throughout my time as a graduate student at Clemson University. I would like to thank Dr. Richard S. Miller and Dr. Rui J. Qiao for being on my advisory committee.

I would also like to thank Dr. Paolo Guarneri and my fellow research mate Sreekanth Reddy Gondipalle for extending their selfless help to assist me in my research.

I would like to acknowledge the ARC (Army Research Centre) for funding this work. Last but not the least I would like to thank my friends and colleagues at Clemson for their continuous support.

TABLE OF CONTENTS

ABSTRACT.....	ii
DEDICATION	iv
ACKNOWLEDGMENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
INTRODUCTION	1
LITERATURE REVIEW	8
2.1 ARTIFICIAL NEURAL NETWORKS – AN INTRODUCTION	8
2.2 APPLICATIONS	9
2.3 CLASSIFICATION	9
2.4 THE FEED-FORWARD ERROR-BACKPROPAGATION NEURAL NETWORK	11
2.5 RELEVANT WORK	12
PROCEDURE.....	16
3.1 INTRODUCTION	16
3.2 PROBLEM DEFINITION AND FORMULATION	17
3.4 DATA PRE-PROCESSING	21
3.5 NEURAL NETWORK TRAINING.....	23
3.6 NEURAL NETWORK GENERALIZATION	27
3.7 DATA POST-PROCESSING.....	31

3.8 SUMMARY	31
3.9 RESULTS	32
OPTIMIZATION	40
4.1 INTRODUCTION	40
4.2 VEHICLE UNDERHOOD THERMAL PERFORMANCE AS AN OBJECTIVE	41
4.3 RESULTS	42
CONCLUSION	47
6.1 CONTRIBUTIONS	47
6.2 FUTURE WORK	48
REFERENCES	49

LIST OF TABLES

Table 3.1: Functions/Parameters used in the Neural Network Development Process	31
Table 3.2: Optimum number of neurons in the hidden layer (Early Stopping)	34
Table 3.3: Optimum number of neurons in the hidden layer (Bayesian Regularization)	38
Table 3.4: Bayesian Vs Early Stopping	39
Table 4.1: Compactness and thermal performance function of the test configurations	46
Table 4.2: Validation of the results obtained from the neural network model	46

LIST OF FIGURES

Figure 2.1: Artificial Neuron	8
Figure 2.2: An illustration of the problems solved by ANN's. (a) Pattern classification. (b) Clustering. (c) Function approximation. (d) Forecasting. (e) Association (e.g., image completion) [15].....	10
Figure 2.3: A Feed-Forward Neural Network.....	12
Figure 2.4: Backpropagation.....	13
Figure 3.1: Steps involved in building a neural network.	16
Figure 3.2: Neural Network model used for average temperature predictions.	17
Figure 3.3: Basic steps involved in the CFD analysis.	19
Figure 3.4: Isometric view of a sample configuration	20
Figure 3.5: Top view of a sample configuration.....	20
Figure 3.6: Neural Network Training Process.	24
Figure 3.7: Tan-Sigmoidal and Linear Transfer Functions [17].....	26
Figure 3.8: Effect of overfitting on network generalization	28
Figure 3.9: Early Stopping Technique	29
Figure 3.10: Training error as a function of the number of neurons in the hidden layer.....	33
Figure 3.11: Network Performance in Predicting the Radiator Temperature with Early Stopping Generalization Technique.....	35
Figure 3.12: Network Performance in Predicting the Coolant Temperature with Early Stopping Generalization Technique.....	35
Figure 3.13: Training error and effective parameter ratio as a function of the number of neurons in the hidden layer (Radiator).	36
Figure 3.14: Training error and effective parameter ratio as a function of the number of neurons in the hidden layer (Airfilter).	37
Figure 3.15: Network Performance in Predicting the Brake Booster Temperature with Bayesian Generalization Technique.	38
Figure 4.1: Optimization procedure.	41

Figure 4.2: Optimization Results (Thermal Objective not included)	43
Figure 4.3: Optimization Results (Thermal Objective included).....	43
Figure 4.4: Optimization Results showing thermal performance as an objective.	44
Figure 4.5: The geometric arrangement of the components under the hood for the solution with a higher thermal performance.....	45
Figure 4.6: The geometric arrangement of the components under the hood for the solution with a lower thermal performance.	45

CHAPTER ONE

INTRODUCTION

In recent years, significant design changes in individual vehicle components are being proposed by their respective design teams during the initial stages of the vehicle development process. Though, these design changes enhance the overall vehicle performance to an extent, the placement of these components under the car hood also plays a vital role in increasing the vehicle performance. Obtaining these optimal locations for the components under the car hood to enhance the vehicle performance forms a multi-objective packing optimization problem. In the past, Gantovnik et al. [1] and others [2]-[7] developed a multi-objective optimization routine for packaging considerations of the components under the car hood. Three objective functions were considered:

- vehicle dynamic performance
- maintainability (accessibility of components)
- survivability (for Army Vehicles subject to attacks)

However, the underhood thermal management is also of vital importance in the vehicle design process. Over the years, advancements in the vehicle design process were typically targeting to improve the performance and fuel economy of the vehicle. Significant changes in the aerodynamic styling, also the introduction of new components reduced the space available in the vehicle underhood. The placement of certain heat

sensitive components, such as the battery, close to the heat sources, such as the engine or the exhaust manifold, can lead to their premature failure.

Traditionally, the means to address the underhood thermal problem is to build vehicle prototypes. This process is extremely time consuming and cost-ineffective. In recent years, new efforts aimed at investigating the problem numerically in the early design stages. Computational fluid dynamics (CFD) is currently the most used numerical technique for the underhood thermal management. Many scientists and researchers successfully implemented the technique achieving a good agreement with the experimental data.

Timothy and Sunil [8] performed CFD analysis for predicting the underhood thermal behavior of a heavy-duty truck and compared the results with test data. In their study, a complete 3-D CAD model of the truck is used for the analysis. The geometry cleanup is done with the Unigraphics CAD software, which is subsequently exported to the IGES format. The Automatic Net-generation for Structural Analysis (ANSA) is used for the mesh generation and FLUENT 5.5 for the CFD analysis. The flow is modeled turbulent and the heat exchangers, condenser and radiator, are modeled using the lumped parameter models available in FLUENT. The cooling fan is modeled using the Multiple Reference Frame (MRF) approach. Radiation effects under the hood are not considered. The radiator coolant heat rejection estimated by the CFD code is accurate to within 4% of the test data.

Salvio et al. [9] include radiation effects in the CFD simulation which was neglected in the previously presented work. The view factor based surface-to-surface (S2S) model in FLUENT is used. This method is much faster in terms of the time taken for iteration than most ray-tracing radiation models. However, the inclusion of the radiation model into the already rigorous CFD simulations is not a viable option, as it increases the computational effort.

Although CFD is an effective tool, it is time consuming. It usually takes several hours to a few weeks for a single simulation. With the available design time in today's market, the design space explored is consequently limited. There is a need for new methods to understand the thermal behavior of a vehicle underhood, which can reduce the design time and enable the engineer to investigate more possibilities. The goal of this work is to develop a quick and efficient approach that can be implemented at the system-level design. Some researchers have come up with different approaches to address the problem.

Li et al. [10] developed a flow network modeling [FNM] method to predict the underhood air flow and temperature distribution. In this method, the underhood environment is represented as a network of flow paths through its various components. The flow and thermal characteristics of the components are specified. The distribution of flow rates and pressures is found by solving a system of algebraic equations for conservation of mass and momentum. The obtained flow rates are then fed to the thermal model to predict the temperature distribution under the hood. This method requires much

less computational effort than a typical computational fluid dynamics (CFD) analysis. However, it necessitates an experimental baseline for calibration of the model before its application to various designs.

Vivek et al. [11] proposed a combined CFD and FNM method approach, towards tackling the under hood thermal management problem. The proposed approach is an alternative method to the full scale CFD analysis performed over the entire underhood flow domain. In this approach, the entire flow domain in the under hood region is broken into small control volumes which are interpreted as nodes in the FNM model. These small control volumes are connected by air flow passages interpreted as the links in the FNM model. For each small control volume, a detailed CFD analysis is conducted to determine the characteristic curve between the air flow rates and its corresponding pressure drop across the control volume. Later, the distribution of these air flow rates and the corresponding pressure drops across each control volume is determined in the FNM method by solving the 1-Dimensional mass and momentum conservation equations across the entire flow network. Also, the average temperature across each control volume is determined by solving the conservation of energy equations over the entire flow network.

Though the 1-Dimensional FNM method used to calculate the distribution of temperature, pressure and flow rates across each individual control volume is fast and is of an order of 10 seconds, it still needs a rigorous CFD analysis to be performed at each individual control volume to determine the characteristic curves relating airflow rates to

pressure drop. Also, discrepancies may creep in while dividing and interpreting the entire flow domain into small control volumes, resulting in a loss of the accuracy in the model devised.

Watanabe et al. [12] developed a similar model combining 1D and 3D fluid flow models using a co-simulation methodology. This model performs a detailed 3D CFD simulation over the individual components such as radiator, engine etc. to determine their characteristic co-efficient i.e. the simple momentum drop element " k " = $2 \frac{|\Delta p|}{\rho v^2}$, where Δp is the pressure drop across the inflow and outflow boundaries of the 3-D component, v is the flow velocity on the boundary and ρ is the density of the fluid. This momentum drop element will then be incorporated into a complete 1D network interpretation of the automobile underhood and solved for the new pressure and mass flow rates across the boundary. The process done so far completes the first iteration of the entire 3-D and 1-D coupling process. Now the newly found pressure and mass flow rates form the boundary conditions for the 3-D CFD simulations in the next iteration. This process is iterated until the error between the thermo-fluid properties obtained on the coupling boundaries for two successive iterations become sufficiently small. This methodology gives the details of the internal flow while conserving the mass flow in the system level. This methodology requires a detailed 3D CFD simulation to be conducted on individual components before performing a 1D simulation over the system level network, which makes this methodology unfeasible for usage in the computationally expensive optimization routines.

As mentioned earlier, considering the computational constraints involved in performing the rigorous multi-objective optimization runs, there is a need to devise an efficient and computationally economic thermal model which can be included into the optimization routine. Though the flow network method that predicts the thermal behavior is relatively fast and fits the bill, it still needs an accurate experimental base line model and results to determine the equivalent pipe flow co-efficient required preparing the model, which might not be an easily available option to every researcher. Thus, a better approach has to be devised, which can deal with these constraints.

Thus, several approaches towards predicting the underhood thermal behavior have been discussed and the need for devising a better approach towards the prediction of underhood thermal behavior to meet the current problems requirements, has been established. Before proceeding further with the development of an approximate thermal model which can be included into the optimization routine, we also need to assess the effects of relocating the components under the hood on its thermal behavior. Since, if the design changes made on the underhood packaging configuration do not cause any effect on its thermal performance, the inclusion of the to be thermal objective into the multi objective optimization routine would not be of much help.

Gondipalle [13] conducted a study to assess the necessity and importance of the inclusion of this thermal objective into the optimization routine developed by Tiwari et al. [1]. CFD simulations were performed to predict the temperatures of the components under the

hood for different configurations. The results indicate that the design changes in the packaging configuration of the underhood have a significant effect on its thermal performance. This justifies the need to include the thermal model into the optimization routine.

In this work, a combined CFD and artificial neural network (ANN) approach is presented to predict the thermal performance for the vehicle underhood which can be included in the optimization routine.

CHAPTER TWO

LITERATURE REVIEW

2.1 ARTIFICIAL NEURAL NETWORKS – AN INTRODUCTION

Artificial neural networks (ANN's) are mathematical simulations of brain function inspired by observations made in the study of biological systems. ANN's may be defined as structures composed of a number of interconnected units (artificial neurons). Each unit has an input/output (I/O) characteristic and implements a local computation or function. The output of any unit is determined by its I/O characteristic, its interconnection to other units, and (possibly) external inputs. The neural network develops an overall functionality through one or more forms of training [14]. Figure 2.1 shows a simple model of an artificial neuron.

Figure 2.1: Artificial Neuron

2.2 APPLICATIONS

ANN's are used in a wide range of applications. They are very good at correlating inputs to outputs. They have been found to be useful in various fields which include function approximation, pattern recognition, clustering, database mining, speech, vision, and control systems. Figure 2.2 [15] illustrates the problems that can be solved by ANN's.

2.3 CLASSIFICATION

ANN's may be classified in various ways on the basis of one or more of their attributes. They may be categorized according to [15]

- The function that the ANN is built for
 - Function approximation
 - Pattern recognition
 - Clustering
- The degree of connectivity of the neurons in the network
 - Partial
 - Full
- The direction of flow of information within the network
 - Recurrent
 - Non-recurrent
- The type of learning style employed
 - Supervised
 - Unsupervised

Figure 2.2: An illustration of the problems solved by ANN's. (a) Pattern classification. (b) Clustering. (c) Function approximation. (d) Forecasting. (e) Association (e.g., image completion) [15].

2.4 THE FEED-FORWARD ERROR-BACKPROPAGATION NEURAL NETWORK

The feed-forward error-backpropagation neural networks are the most widely used type of neural networks and are considered the workhorse of ANN's [16]. The term feed-forward refers to the way how the neural network functions. The term backpropagation describes how the learning process of this neural network is carried out. A feed-forward network typically consist of

- An input layer
- An output layer
- One or more hidden layers

The input layer consists of nodes which represent the input variables of the problem. The output layer consists of nodes which represent dependent variables, for which the network is being modeled. The hidden layers consists of nodes which are used describe the underlying input-output relationships. Some inputs in the network may be of more significance, hence should be given more importance. This is achieved by giving weightage to the inputs. The weights determine the relative importance of the neural network inputs. Inputs which are more relevant are given greater weightage than the others in the network. Figure 2.2 depicts a simple feed-forward neural network. As the name suggests, in the feed-forward neural network, the data are fed forward through the network without any feedback i.e., the flow is unidirectional (non-recurrent). The outputs from the neurons in a layer are not fed back to neurons in the same or the preceding layers.

Figure 2.3: A Feed-Forward Neural Network

A supervised learning process is employed in the training of feed-forward neural networks. In this process, each example of the training data is provided with the target output. The neural network output is now compared with the corresponding target output and the obtained error is used to adjust the weights and biases of the network. Backpropagation is a supervised learning process where the error computed at the output is propagated back from the output layer, to the hidden layer, and finally to the input layer to adjust the weights and biases. Figure 2.4 depicts the process of backpropagation.

2.5 RELEVANT WORK

Researchers in the past have successfully built ANN's from the CFD data for less complex electronics packaging problems. Kargar et al. [17] used Computational fluid

dynamics (CFD) and ANN to examine the cooling performance of two electronic components in an enclosure filled with a Cu-water nanofluid. The heat transfer within the

Figure 2.4: Backpropagation

enclosure is due to laminar natural convection between the heated electronic components mounted on the left and right vertical walls with a relatively lower temperature. The results of a CFD simulation were used to train and validate a series of ANN architectures, which were developed to quickly and accurately carry out this analysis. A comparison study between the results from the CFD simulation and the ANN analysis was made. The study indicates that the ANN accurately predicts the cooling performance of electronic components within the given range of data.

Kuan et al. [18] presented an artificial intelligent system that predicted the temperature distribution of the semiconductor chip with multiple heat sources by integrating the back-

propagation neural network (BNN) and computational fluid dynamics (CFD) techniques. Six randomly generated coordinates of three power sections on the chip die are the inputs and sixty-four temperature monitoring points on the top of the chip die are the outputs. In the methodology presented, one hundred sets of training data obtained from the CFD simulations results were sent to the BNN for the training. Sixteen other input sets of data were generated and used as the validation data. The results of the validation data were then compared to those of the CFD simulations. It was shown that the BNN model was able to accurately estimate the corresponding temperature distribution as well as the maximum temperature values under different power distribution after being well trained.

Also, the ANN's were found to be successful for packaging optimization problems. Hadim and Suwa [19] presented a systematic multidisciplinary electronics packaging design and optimization methodology, which takes into account the complexity of multiple design trade-offs, operated in conjunction with the ANN's technique. This method was applied to a plastic ball grid array package design. Multidisciplinary criteria including thermal, structural, electromagnetic leakage, and cost were optimized simultaneously using key design parameters as variables. ANN's were used for thermal and structural performance predictions which resulted in a large reduction in computational time. The methodology was able to provide the designers a tool for systematic evaluation of the design trade-offs which are represented in the objective function.

In this work, CFD analysis is carried out to predict the temperatures of various components under the hood for different select configurations. Then an ANN is built which can predict the temperature distribution for any given configuration of the components under the hood. A feed-forward error-backpropagation neural network is used for this purpose. They are the most commonly used type of networks for function approximation. Once the ANN is built, it is tied to an optimizer to identify low thermal risk configurations of the components under the hood.

CHAPTER THREE

PROCEDURE

3.1 INTRODUCTION

The procedural steps followed in building the Neural Network are described in this chapter. The Neural Network Tool Box available in MATLAB is used for the purpose.

Figure 3.1 provides a summary of the steps involved in building the Neural Network.

Figure 3.1: Steps involved in building a neural network.

3.2 PROBLEM DEFINITION AND FORMULATION

The problem at hand is to predict the vehicle under hood thermal performance. The input variables are the spatial locations (centroids) in Cartesian coordinates of the components under the hood. Here, seven components are considered which include the engine, radiator, exhaust manifold, battery, airfilter, coolant, and brake booster. The outputs are the average temperatures of those components excluding the engine and exhaust manifold. The average temperatures of the components provide a good measure of the overall thermal performance under the hood. Figure 3.2 illustrates the problem formulation. In this problem, a Neural Network approximation each for the 5 different

Figure 3.2: Neural Network model used for average temperature predictions.

outputs was built. These Neural Networks approximated the values for the five outputs corresponding to the inputs available. The spatial locations of all the components present in this problem constitute for the inputs.

3.3 DATA COLLECTION

Since neural networks are data driven, data collection is an important step in building neural networks. The data collected should be large enough to cover the problem domain i.e. the sample data should represent the population. The process of data collection should be repeatable, reproducible, and stable in order to ensure the accuracy of the samples is maintained [20].

CFD analysis is performed on the vehicle under hood for different configurations of the components to estimate their average temperatures. The temperatures of the engine and the exhaust manifold are set as boundary conditions in the CFD analysis. Hence, they are not included in the network outputs described in section 3.2. The different configurations of the components under the hood are generated randomly, with collision taken care of, in SolidWorks 2010. ANSYS FLUENT 13.0 is used to carry out the analysis. The basic steps involved in the analysis are presented in figure 3.3. The detailed procedure at each step of the analysis is compiled and conducted by Gondipalle [13] for a single configuration. The same is replicated for all the configurations. The simulations are performed on an 8 core, 32 GB processor in parallel on the palmetto cluster. A second order accuracy is employed for the simulations. The average simulation time taken is two hours. Also, some conditions remain constant for all the configurations. They are

- Ambient Temperature (298 K)
- Vehicle speed (55 km/h)
- Radiator heat rejection (28000 W)

Figure 3.3: Basic steps involved in the CFD analysis.

The data for a total of fifty four randomly generated configurations is collected. They typically contain fifty four sets of twenty one inputs (X, Y, Z coordinates of seven components as shown in figure 3.2) and five outputs (Average temperatures of five components as shown in figure 3.2). Figure 3.4 and figure 3.5 show the isometric view and top view respectively of a sample configuration.

Figure 3.4: Isometric view of a sample configuration

Figure 3.5: Top view of a sample configuration

3.4 DATA PRE-PROCESSING

Neural Network training can be made more efficient by performing certain preprocessing techniques on the network inputs and targets. It helps to accelerate convergence of the network. Various preprocessing techniques are usually applied which may include [21]

- Noise Removal
- Reducing Input Dimensions
- Data Normalization
- Deletion of outliers

In the present problem, the data is scaled (normalized) before proceeding to training. Also, the centroid location of the exhaust manifold is not considered reducing the input dimension to eighteen as its position remains constant relative to the engine.

3.4.1 Data Normalization

Normalizing the data within a specified range is often useful (i) to prevent larger numbers from overriding smaller ones i.e. reduce bias, and (ii) to prevent premature saturation of hidden nodes, which impedes the learning process [15]. Several techniques are available for scaling the data which include (a) Statistical or Z-Score Normalization, (b) Min-Max Normalization, (c) Sigmoidal or Softmax Normalization, (d) Energy Normalization, and (e) Principal Components Normalization [20]. MATLAB consists of Statistical Normalization and Min-Max Normalization techniques for scaling the data.

3.4.1.1 Statistical or Z-Score Normalization

The statistical or Z-score normalization technique uses the mean (μ) and standard deviation (σ) for each feature across a set of training data to normalize each input feature vector. The mean and standard deviation are computed for each feature, and then the transformation given in Eq. (3.1) is made to each input feature vector as it is presented. This produces data where each feature has a zero mean and a unit variance. The statistical norm is given by [20]

$$x'_i = \left[\frac{x_i - \mu_i}{\sigma_i} \right] \quad (3.1)$$

In MATLAB the function *mapstd* normalizes the inputs and the targets so that they will have zero mean and unity standard deviation.

3.4.1.2 Min-Max Normalization

In this technique the input and output variables (x_i) are rescaled from one range of values to a new range of values. Most often the input and output variables are rescaled to lie within a range of 0 to 1 or from -1 to 1. The rescaling is done by using a linear interpolation formula given in Eq. (3.2) [20]:

$$x'_i = (\mu_2 - \mu_1) \times \left(\frac{x_i - x_i^{\min}}{x_i^{\max} - x_i^{\min}} \right) + \mu_1 \quad (3.2)$$

where x'_i is the normalized value of x_i , $x_i^{\max}$ and $x_i^{\min}$ are the maximum and minimum values of x_i in the database, and μ_2 and μ_1 are the maximum and minimum values of the new range to which the data has to be scaled. Min-Max normalization has the advantage

of preserving exactly all relationships in the data, and it does not introduce any bias [20]. In MATLAB the function *mapminmax* performs the min-max normalization for the data to lie within a range of -1 to 1. The *mapminmax* function is the default normalization technique which is included in the neural network function.

3.5 NEURAL NETWORK TRAINING

Once the pre-processing is completed, the network weights and biases are initialized, and the network is ready for training. In the present problem, the network is trained for function approximation. A feed-forward error-backpropagation learning algorithm (described in chapter 2) is used for the training process. The new network function *newff* available in MATLAB is used to create the feed-forward backpropagation network. The training is an iterative process where the network weights and biases are adjusted to minimize the network error (network performance). The trained network is then checked for its generalization capability. The iterative process is shown in the figure 3.6.

3.5.1 Neural Network Parameters

Neural Network training requires an optimal selection of various parameters, which may include

- Number of hidden layers
- Number of neurons in each layer
- Transfer functions used
- Training functions used

These parameters are generally determined through trial and error.

Figure 3.6: Neural Network Training Process.

3.5.1.1 Number of Hidden Layers

In 1989, Cybenko [22] proved that the standard multilayer feed-forward network with a single hidden layer, with enough number of hidden neurons, and with arbitrary activation function, can approximate any arbitrary function. This is also called the Universal approximation theorem. Sometimes, two hidden layers are used to speed up convergence. A feed-forward network with more than five hidden layers is not necessary.

3.5.1.2 Number of Neurons in a Hidden Layer

The determination of optimum number of neurons in a hidden layer is an important task in neural network development. If the hidden neurons size is too small, the neural network may not learn the complex pattern of the data. In a similar manner, if the hidden neurons size is too large, the neural network may overfit (memorize) the data. Also, with increasing number of hidden neurons, the size of the associated weight matrix increases, resulting in the need for larger training sets and hence excessive training times.

3.5.1.3 Transfer Functions

The functions that determine the input-output relationship are called transfer functions. The use of backpropagation algorithm for adjusting the weights and biases requires the transfer functions to be differentiable. A variety of transfer functions are available in MATLAB. A tan-sigmoidal (*tansig*) transfer function in the hidden layer and linear (*purelin*) transfer function in the output layer, depicted in the figure 3.7 [23], are used in

the problem. These are transfer functions used by Cybenko [22] in the proof of Universal approximation theorem.

Figure 3.7: Tan-Sigmoidal and Linear Transfer Functions [17].

3.5.1.4 Training Functions

Several backpropagation algorithms are available in MATLAB in training the feed forward networks. The choice is made on the basis of rate of convergence and the type of application of the neural networks. The Levenberg-Marquardt algorithm is reported to have the fastest convergence and to be highly suitable for function approximation problems. In many cases, *trainlm*, which realizes the Levenberg-Marquardt algorithm, is able to obtain lower mean square errors than any of the other algorithms tested [23]. The training functions *trainlm* and *trainbr* (for Bayesian Regularization) are used in the problem.

3.5.2 Network Convergence

Training is stopped when one of the following criterions is attained:

- The number of epochs (iterations) exceeds a specified value
- The network performance function (or network error) reaches a specified goal

- The training time is longer than a specified time
- The error on validation set increases for a specified number of epochs (early stopping, section 3.6.1)

The network performance function, F , typically used for training feed-forward neural networks, is defined as the mean sum of squares of the network errors (Eq. (3.3))

$$F = mse = \frac{1}{N} \sum_{i=1}^N (T_i - z_i)^2 \quad (3.3)$$

where T_i are the target outputs, z_i are the network outputs and N is the number of outputs. Sometimes the sum square of errors, sse , may be used as the network performance function.

3.6 NEURAL NETWORK GENERALIZATION

The common problem that arises during the neural network training is overfitting of the data. The error on the training data is very small, but when a new set of data is presented for verification the network error becomes large. The network memorizes the data, rather than learning the underlying relationships of the inputs and outputs. Thus, it is necessary to examine the best trained neural network for its generalization capability. Figure 3.8 shows the effect of overfitting on network generalization.

The problem of overfitting can be resolved by providing a large enough network size, which results in an optimum fit. But, when there is a limited amount of data, or the optimum network size is unknown, the problem can be addressed by some generalization techniques. The neural network toolbox in MATLAB provides two techniques, (i) Early

stopping, and (ii) Bayesian Regularization for improving the network generalization. The following sections describe the two techniques and their implementation.

Figure 3.8: Effect of overfitting on network generalization

3.6.1 Early Stopping

Early stopping is the most commonly used regularization technique in machine learning. In this method, the available data is split into two subsets namely, training set and validation set. The data division is done by index so that the samples are assigned to test set and validation set successively using *divideind* function in MATLAB. The training set

is used to compute the gradient and update the network weights and biases. The network error is monitored on both the training and validation sets. Initially, the error on the validation set decreases along with the error on the training set. But, when the network starts to memorize the data, the error on the validation set starts to increase. When this error on the validation set increases for a specified number of iterations the training is stopped and the weights and biases at the minimum validation error are used. In MATLAB, the early stopping is automatically included with the network creation functions. Figure 3.9 depicts the early stopping technique applied for the neural network which predicts the temperature of the radiator. The error on the validation set starts to increase after ten iterations continually for five iterations. Hence, the training is stopped and the weights and biases at the tenth iteration are used for the neural network.

Figure 3.9: Early Stopping Technique

3.6.2 Bayesian Regularization

Another method used for neural network generalization is called Bayesian regularization. In this method, the network performance function, which is given by the equation 3.1, is modified. Typically, the modified performance function consists of an additional term which is the mean of the sum of squares of the network weights and biases. This modification improves the generalization capabilities of the trained network. This performance function causes the network to have smaller weights and biases, and this forces the network response to be smoother and less likely to over fit [23]. The regularized performance function, F_{reg} , is given by Eq. (3.4)

$$F_{reg} = msereg = \gamma mse + (1 - \gamma)msw \quad (3.4)$$

where γ is the performance ratio, and

$$msw = \text{mean of sum of squares of network weights and biases} = \frac{1}{n} \sum_{i=1}^n w_i^2$$

The choice of the performance ratio, γ , is determined by trial and error. Typically its value is set to be 0.5. The training function, *trainbr*, available in MATLAB, has Bayesian regularization implemented in it. Alternatively, the regularized performance function, *msereg*, can be used in place of the traditional performance function, *mse*. In the latter case the training function, *trainlm*, can be used instead of *trainbr*. However, in both the cases, since early stopping is automatically implemented in neural networks, the data division should be set to null to avoid its effects.

3.7 DATA POST-PROCESSING

Once the neural network is trained and checked for its generalization capabilities, the outputs obtained from the network are to be scaled back to their original range. This is done by the *mapminmax* function, described in the pre-processing stage.

3.8 SUMMARY

The table 3.1 shown below summarizes the various functions and parameters used in MATLAB during the various stages of the neural network development process.

Table 3.1: Functions/Parameters used in the Neural Network Development Process

Neural Network Development Stage	Function / Parameter
Data Pre-processing <ul style="list-style-type: none">Data Normalization	<i>Mapminmax</i>
Neural Network Training <ul style="list-style-type: none">Neural Network Parameters<ul style="list-style-type: none">Neural Network FunctionNumber of Hidden LayersTransfer FunctionsTraining FunctionNetwork Convergence<ul style="list-style-type: none">Max Number of Epochs	<i>newff</i> 1 <i>tansig</i> (Hidden Layer), <i>purelin</i> (Output Layer) <i>trainlm</i> 1000

<ul style="list-style-type: none"> ○ Network Performance Function 	<i>mse, sse</i>
<p>Neural Network Generalization</p> <ul style="list-style-type: none"> • Early Stopping <ul style="list-style-type: none"> ○ Data division ○ Max number of iterations for which the error on the Validation set can increase • Bayesian Regularization <ul style="list-style-type: none"> ○ Network Performance Function ○ Training Function 	<div><i>divideind</i></div> <div>5</div> <div><i>msereg</i></div> <div><i>trainbr</i></div>
<p>Data Post-processing</p> <ul style="list-style-type: none"> • Scale back Normalized Output Data 	<i>Mapminmax</i>

3.9 RESULTS

A neural network for the underhood thermal problem with the available CFD data is trained by adjusting its various parameter values and is then checked for its

generalization capabilities. Both the generalization techniques presented in the previous section are investigated.

3.9.1 Results – Early Stopping

A single hidden layer is used in the training process. The optimum number of neurons in the hidden layer which gives a good representation of the data is found by trial and error. The sum square of errors, sse , of the network is monitored as a function of the number of neurons in the hidden layer. Figure 3.10 represents the sum square of errors, sse , obtained in training the network corresponding to the average temperature of the battery as an output, with respect to the number of neurons in the hidden layer. Also in the early

Figure 3.10: Training error as a function of the number of neurons in the hidden layer.

stopping technique the error on the validation set is monitored which ensures the network generalization. In the above case optimum network performance was obtained with nine neurons in the hidden layer. Table 3.2 summarizes the optimum number of neurons in the hidden layer, for each of the networks in predicting the corresponding average temperature of the components.

Table 3.2: Optimum number of neurons in the hidden layer (Early Stopping)

Network Output (Early Stopping)	Number of Neurons in the Hidden Layer
Radiator average temperature	3
Coolant average temperature	9
Battery average temperature	9
Brake booster average temperature	12
Airfilter average temperature	2

Figure 3.11 and Figure 3.12 show the performance of the neural network in predicting the temperature of the radiator and the coolant respectively. The targets, T obtained from the CFD are plotted on the x-axis and the corresponding neural network outputs, Y are plotted on the y-axis. The data on the plot includes both the training and validation sets. A regression value of one indicates that the neural network has been trained correctly with good generalization capabilities. The regression of the neural network developed using early stopping for the radiator and coolant are 0.99745 and 0.90878 respectively.

Figure 3.11: Network Performance in Predicting the Radiator Temperature with Early Stopping Generalization Technique.

Figure 3.12: Network Performance in Predicting the Coolant Temperature with Early Stopping Generalization Technique.

3.9.2 Results – Bayesian Regularization

A single hidden layer is used in the training process. The sum square of errors, sse , of the network is monitored as a function of the number of neurons in the hidden layer. In the bayesian regularization technique the effective number of parameters, which are the weights and biases in this case, used during the training is obtained. The ratio of the effective number of parameters to the total number of parameters is also monitored with respect to the number of hidden layers. Figure 3.13 and Figure 3.14 show the training error and effective parameter ratio as a function of the number of neurons for radiator and airfilter respectively.

Figure 3.13: Training error and effective parameter ratio as a function of the number of neurons in the hidden layer (Radiator).

Figure 3.14: Training error and effective parameter ratio as a function of the number of neurons in the hidden layer (Airfilter).

In figure 3.13 the training error reaches its minimum at four neurons in the hidden layer. Also the ratio of number of effective parameters to total parameters decreases. This indicates that the training can be stopped and the optimum number of neurons in the hidden layer in the case of the radiator is four. Similarly the optimum number of neurons in the hidden layer in case of the airfilter is three. Table 3.3 summarizes the optimum number of neurons in the hidden layer, for each of the networks in predicting the corresponding average temperature of the components.

Table 3.3: Optimum number of neurons in the hidden layer (Bayesian Regularization)

Network Output (Bayesian Regularization)	Number of Neurons in the Hidden Layer
Radiator average temperature	4
Coolant average temperature	4
Battery average temperature	7
Brake booster average temperature	8
Airfilter average temperature	3

Figure 3.15 shows the performance of the neural network in predicting the temperature of the brakebooster. The regression in this case is 0.99294

Figure 3.15: Network Performance in Predicting the Brake Booster Temperature with Bayesian Generalization Technique.

Overall the Bayesian regularization technique was found to be more effective in terms of the network performance. Table 3.4 provides a comparison of the two regularization techniques with respect to the network performance.

Table 3.4: Bayesian Vs Early Stopping

Network Output	Regression (Bayesian)	Regression (Early Stopping)
Radiator avg temperature	0.99925	0.99745
Coolant avg temperature	0.95391	0.90878
Battery avg temperature	0.99537	0.8023
Booster avg temperature	0.98389	0.91295
Airfilter avg temperature	0.97058	0.87652

In the following chapter the thermal model, built using the neural networks, is included in the optimization routine.

CHAPTER FOUR

OPTIMIZATION

4.1 INTRODUCTION

As discussed earlier, Gantovnik et al. [1] developed a multi-objective optimization routine for packaging considerations of the components under the car hood. The optimization problem has three objectives: vehicle dynamic performance, survivability and maintainability, subject to two constraints: overlap detection and ground clearance. Optimization is performed using a combination of the modified NSGA-II and packing sequence.

Typically, this work has been extended and several improvements are made in different aspects which include the type of algorithms used in the optimization routine and the objectives considered in the problem. Today, the car underhood packing optimization routine available is a code written in java which addresses various objectives, namely: center of gravity, compactness, survivability, and maintainability subject to no overlap among components and with the enclosure, and minimum ground clearance. A non-deterministic evolutionary multi-objective algorithm (AMGA-2) is used to perform the optimization process. Figure 4.1 shows the procedural steps involved in the optimization process.

Figure 4.1: Optimization procedure.

4.2 VEHICLE UNDERHOOD THERMAL PERFORMANCE AS AN OBJECTIVE

In the previous chapters, the need for including the thermal performance of the vehicle underhood as an objective in the packing optimization problem is established. The thermal performance of the vehicle underhood is evaluated from the average temperatures of the components which can be determined from the developed neural

network model. The thermal performance function, F_{Thermal} , is defined as the root mean square value of the average temperatures of the components under the hood (Eq. (4.1)).

$$F_{\text{Thermal}} = \sqrt{\frac{\sum_{i=1}^N T_i^2}{N}} \quad (4.1)$$

In the Eq. 4.1 above, N is the total number of components under the hood, and T_i is the average temperature of the i 'th component. Now the objective becomes minimizing the thermal performance function, F_{Thermal} . Mathematically, the objective is represented as:

$$\text{Minimize } [F_{\text{Thermal}}] \quad (4.2)$$

4.3 RESULTS

The thermal objective established is included in the multi-objective optimization routine. A total of five thousand runs using non-deterministic evolutionary multi-objective algorithm (AMGA-2) are performed in the optimization routine. The typical running time of the algorithm is around five minutes. The objectives in the vehicle underhood packing are of conflicting in nature. Hence, a single solution to the multi-objective packing problem cannot be obtained, but a set of non-dominated solutions are found. Figure 4.2 shows the results obtained from optimization with the objectives compactness, survivability, maintainability. The thermal objective is not included at this point. This is done to investigate the effect of adding an additional objective to the optimization routine. In figure 4.3 the results obtained after adding the thermal performance as an objective is shown. As the dimension of the objectives increases the number of the design trade-offs increases resulting in more number of non-dominated solutions.

Figure 4.2: Optimization Results (Thermal Objective not included)

Figure 4.3: Optimization Results (Thermal Objective included)

In figure 4.4 the thermal objective is plotted against the three other objectives, compactness, survivability and maintainability. To physically understand the solution in regard to the thermal objective, two solutions are picked of which one has a higher value of the thermal performance function (the solution circled with red in the figure 4.4) and the other has a lower value of the thermal performance function (the solution circled with blue in the figure 4.4). Now the geometric configurations of these solutions are visualized.

Figure 4.4: Optimization Results showing thermal performance as an objective.

Figure 4.5: The geometric arrangement of the components under the hood for the solution with a higher thermal performance

Figure 4.6: The geometric arrangement of the components under the hood for the solution with a lower thermal performance.

From the figures 4.5 and 4.6 it can be seen that in case of the solution with the lower thermal performance, i.e, a higher root mean square temperature, the system is more compact, i.e, a low value of the compactness. The components are closer to the center of the underhood which in turn position them closer to the engine which is a heat source. The compactness values of the two configurations can be seen in the table 4.1.

Table 4.1: Compactness and thermal performance function of the test configurations

Test Configuration	Compactness	Thermal performance function
Configuration Blue	109850	314.8042
Configuration Red	105220	317.0586

To test the validity of the results obtained for the thermal performance function in the optimization routine, which uses the neural network model, a CFD analysis is carried out to the two test configurations and compared with the above solutions (Table 4.2). The

Table 4.2: Validation of the results obtained from the neural network model

Test Configuration	Thermal performance function	
	CFD	Neural Network
Configuration Blue	314.9043	314.8042
Configuration Red	317.3142	317.0586

results are found to be in a good agreement from the ones obtained from the CFD.

CHAPTER FIVE

CONCLUSION

6.1 CONTRIBUTIONS

The multi-objective packing optimization problem of the vehicle underhood is introduced. The importance of the vehicle underhood thermal performance as an objective in the multi-objective packing problem is discussed. The need for a quick and efficient approach in the early design stages to predict vehicle underhood thermal performance is established.

In this work a combined CFD and Neural Network approach to predict the vehicle underhood thermal behavior is presented. A simple and efficient thermal model using CFD and neural networks is developed. This model is successfully integrated into the optimization routine to predict the thermal behavior of the underhood.

The results are obtained for the vehicle underhood packing problem with thermal performance included as an objective. The results obtained for the thermal performance function in the optimization routine are validated by performing CFD simulations to the configurations. The results are found to be in an acceptable agreement with the ones obtained from the Neural Network to justify the use of such a methodology for design purposes.

6.2 FUTURE WORK

The neural network approximation to the rigorous CFD simulations presented in this work corresponds to a specific vehicle driving conditions and atmospheric conditions. In future, considering the computational constraints, analyses can be conducted to determine the significance of the need for approximating different vehicle driving scenarios to capture more realistic data.

Also, the work presented here uses a Feed-Forward neural network approximation technique to construct the numerical approximation to the computationally heavy CFD simulations. However, in future several such neural network approximation techniques, like the Radial Basis Function, can be explored. The accuracy of the results thus obtained can be analyzed to determine the best available approximation technique for the current problem.

Also, the initial design data populated for the problem presented in this work was random. We resorted to the random design data sampling technique to build a feasible design space satisfying the 3-Dimensional geometric constraint, which restricts the location space for the components such that no two of them collide with each other. However with the use of a random technique, there is a chance for the generation of sparsely spaced designs over the complete design space. Thus, in the future an efficient sampling technique that includes the 3-Dimensional geometric constraint can be used which uniformly spreads the data across the entire design space.

REFERENCES

- [1]. V B Gantovnik, S Tiwari, G M Fadel, Y Miao, "Multi-Objective Vehicle Layout Optimization", 11th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA Paper No. 2002-5431, Portsmouth, VA, September 6-8, 2006.
- [2]. P M Grignon, "Configuration Design Optimization Method", Ph.D. thesis, Clemson University, Clemson, SC, May, 1999.
- [3]. P M Grignon, and G M Fadel, "Configuration design optimization method", Proceedings of DETC99, ASME 1999, Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Las Vegas, Nevada, September 12-15, 1999.
- [4]. Y Miao, V Blouin, and G M Fadel, "Multi-objective configuration optimization with vehicle dynamics applied to midsize truck design", Proceedings of DETC03, ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, Illinois, September 2-6, 2003.
- [5]. Y Miao, V Blouin, and G M Fadel, "Design configuration of vehicle components with domain knowledge enhancement", Proceedings of DETC04, ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Salt Lake City, UT, September 28 - October 3, 2004.
- [6]. Y Miao, "Packing Optimization of Engineering Problems, Ph.D. thesis, Clemson University", 2005.
- [7]. Y Miao, and G M Fadel, "Genetic Algorithm for the Packing Problems", Proceedings of DETC05, ASME 2005 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Long Beach, CA, September 24 – 28, 2005.
- [8]. T P Nobel, S K Jain, "A Multi-Dimensional Approach to Truck Underhood Thermal Management", SAE Technical Paper 2001-01-2785, 2001.
- [9]. S Chacko, V Kumar, B Shome, A K Agarwal, "Underbody Simulation for Vehicle Thermal Management", Altair CAE Users Conference, Bangalore, 2006.
- [10]. P Li, G K Chui, J M Glidewell, T H Chue, M C Lai, "A Flow Network Approach to Vehicle Underhood Heat Transfer Problem", SAE Technical Paper 931073, 1993.

- [11]. V Kumar, S Kapoor, G Arora, S K Saha, P Dutta, "A Combined CFD and Flow Network Modeling Approach for Vehicle Underhood Air Flow and Thermal Analysis", SAE Technical Paper 2009-01-1150, 2009.
- [12]. N Watanabe, M Kubo, N Yomoda, "An 1D-3D Integrating Numerical Simulation for Engine Cooling Problem", SAE Technical Paper 2006-01-1603, 2006.
- [13]. S R Gondipalle, "CFD Analysis of the under hood of a car for packaging considerations", Master's Thesis, Clemson University, 2011.
- [14]. R J Schalkoff, "Artificial Neural Networks", The McGraw-Hill Companies, Inc, 1997.
- [15]. I A Basheer, M Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application", Journal of Microbiological Methods, 2000.
- [16]. D E Rumelhart, G E Hinton, R J Williams, "Learning internal representation by error propagation". In: D E Rumelhart, J L McClelland, Parallel Distributed Processing: Exploration in the Microstructure of Cognition, Vol. 1. MIT Press, Cambridge, MA, Chapter 8.
- [17]. A Kargar, B Ghasemi, S M Aminossadati, "An Artificial Neural Network Approach to Cooling Analysis of Electronic Components in Enclosures Filled with Nanofluids", Journal of Electronic Packaging, Vol. 133 / 011010, 2011.
- [18]. Y D Kuan, Y W Hsueh, H C Lien, W P Chen, "Integrating Computational Fluid Dynamics and Neural Networks to Predict Temperature Distribution of the Semiconductor Chip with Multi-heat Sources", Lecture Notes in Computer Science, Volume 3973/2006, 1005-1013: Advances in Neural Networks, 2006.
- [19]. H Hadim, T Suwa, "A Multidisciplinary Design and Optimization Methodology for Ball Grid Array Packages Using Artificial Neural Networks", ASME Transactions, Vol. 127, 2005.
- [20]. K L Priddy, P E Keller, "Artificial Neural Networks: an Introduction", The International Society for Optical Engineering, 2005.
- [21]. F U Dowla, L L Rogers, "Solving Problems in Environmental Engineering and Geosciences With Artificial Neural Networks", The MIT Press, Cambridge, MA, 1995.
- [22]. G V Cybenko, "Approximation by Superpositions of a Sigmoidal Function", Mathematics of Control, Signals, and Systems, 1989.

[23]. MATLAB 7.10.0 (R2010a) Help Manual.