# A NEURAL NETWORK BASED MODEL FOR PREDICTING ACCELERATION AND GRADEABILITY PERFORMANCES OF TRANSIT BUSES

**Saravanan Muthiah and Bohdan T. Kulakowski**

Pennsylvania Transportation Institute, The Pennsylvania State University
University Park, PA 16802, USA

## Abstract

Slow moving vehicles can cause traffic congestion and prolonged intersection clearance times. Adequate acceleration and gradeability performances are hence essential to keep vehicular traffic flowing in a smooth manner and to minimize accidents. While acceleration tests are relatively straightforward to perform, gradeability tests have some inherent difficulties associated with them. One such difficulty is the seemingly impossible scenario of having all possible road grades available for testing. Computer simulation of acceleration and gradeability performances hence appears appealing as they are closely related, except for the fact that gradeability is a steady state calculation and therefore the static weight of a vehicle is used rather than the effective weight.

In this paper, an Artificial Neural Network (ANN) based method is used to predict gradeability and acceleration performances of transit buses. The ANN method used here has an automatic means for generating empirical formulae consisting of product units and their sums (if desired). This alleviates the most common problem associated with Neural Networks – "Neural Networks are like black boxes and are hard to interpret!"

Often, it is also of interest to know the range of possible vehicle configurations that can provide a desired acceleration or gradeability performance. To this effect, an inversion method based on non-linear programming has also been examined. The result of such an inversion is a set of bus design parameters that can be used to achieve the desired performance.

# A NEURAL NETWORK BASED MODEL FOR PREDICTING ACCELERATION AND GRADEABILITY PERFORMANCES OF TRANSIT BUSES

**Saravanan Muthiah and Bohdan T. Kulakowski**

Pennsylvania Transportation Institute, The Pennsylvania State University
University Park, PA 16802, USA

## 1. INTRODUCTION

In the United States, the primary evaluations used for measuring vehicle performance are acceleration, gradeability and top speed. Acceleration capability of a vehicle can be measured either in terms of the amount of time needed by a vehicle to reach a given speed or as acceleration rates at given speeds. Gradeability, on the other hand, is defined as the maximum grade a vehicle can negotiate at any given speed. Adequate acceleration and gradeability performances are essential to keep vehicular traffic flowing in a smooth manner and to reduce accidents. Slow moving vehicles can cause traffic congestion and prolonged intersection times.

Both acceleration and gradeability tests are technically simple. However, test track requirements for either test can be prohibitive in terms of costs of construction and maintenance. Computer modeling of acceleration and gradeability behaviors is hence very appealing. In this study, a Artificial Neural Network (ANN) model based on a modified version of Saito et al.'s RF5 (Rule extraction from Facts, version 5) algorithm was developed to predict acceleration performance of transit buses. This model has an in built mechanism for generating simple equations relating inputs to outputs and thus alleviates the black-box nature of neural networks.

Acceleration and gradeability performances are closely related, except for the fact that gradeability is a steady state calculation. Hence, gradeability performance can be calculated from acceleration values at a given speed.

## 2. MATHEMATICAL APPROACH

Block diagram of a simplified vehicle system is shown in Figure 1 (Kulakowski, 2003). In this diagram, a heavy vehicle is represented by a vector of design parameters, $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots\ldots\theta_p)$ that could include parameters such as length, width, height, wheel base, axle loads, engine power etc. The other two inputs to the vehicle block are driving vector, $\mathbf{u} = (u_1,u_2,\ldots\ldots u_r)$ and the road vector, $\mathbf{x} = (x_1,x_2,\ldots\ldots\ldots x_k)$. Driving vector includes actions of the driver such as steering, braking and accelerating, while road input vector comprises of road geometry, surface friction, pavement roughness etc. The output signal is a vector, $\mathbf{y} = (y_1, y_2,\ldots\ldots, y_n)$, that includes vehicle performance characteristics. For the purposes of this study, $\mathbf{y}$ is a scalar as only one output i.e. acceleration value at a given speed is modeled.

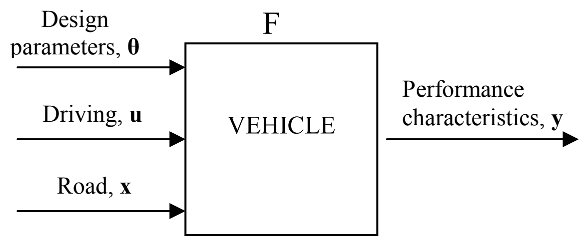


**Figure 1. Block diagram of vehicle system.**

Mathematically, the relationships can be written in vector form as

$$\mathbf{y} = F(\mathbf{u},\mathbf{x},\boldsymbol{\theta}) \tag{1}$$

If driver input, **u** and the road input, **x**, are kept constant, they can be effectively eliminated from active inputs affecting vehicle performance characteristics i.e. $\mathbf{y}=F(\boldsymbol{\theta})$. The knowledge of transformation $F$ is of great value in developing and implementing vehicle regulations since we can then predict to a reasonable degree the performance of a vehicle given its design parameters. Also, if $F$ is known accurately then we can apply reverse engineering concepts (assuming inverse exists) to get corresponding vehicle design parameters that would result in a desired acceleration performance. This can be expressed as

$$\boldsymbol{\theta}^* = F^{-1}(\mathbf{y}^*) \tag{2}$$

where, $\boldsymbol{\theta}^*$ is a vector of vehicle design parameters that would ensure desired acceleration performance given by $\mathbf{y}^*$ .

## 3. FORWARD MODEL USING ARTIFICIAL NEURAL NETWORK

In this study, an artificial neural network based method called RF5 (Rule extraction from Facts version 5) was chosen to model the vehicle transformation, $F$. This was done because of the universal function approximation and other favorable characteristics offered by neural networks (Muthiah, 2006). RF5 is a connectionist method for numerical law discovery (Saito et al., 1997). It consists of a combination of 3 techniques:

1) Use of product unit networks
2) Employment of a second-order learning algorithm called BPQ
3) Adoption of Rissanen's Minimum Description Length (MDL) criterion for finding an adequate number of hidden neurons

Let **x** be an n-dimensional input vector and y be a target value corresponding to **x**. RF5 is capable of discovering a class of numeric laws that can be expressed as

$$y = c_0 + \sum_{i=1}^{H}\left(c_i \prod_{j=1}^{n} x_j^{w_{ij}}\right) \tag{3}$$

where parameters $c_0$, $c_i$ and $w_{ij}$ are real numbers and $H$ is an integer. Now if the inputs are positive, then we can write equation (3) as

$$y = c_0 + \sum_{i=1}^{H} c_i \, exp\left( \sum_{j=1}^{n} w_{ij} ln(x_j) \right) \tag{4}$$

Equation (4) can be regarded as the feed-forward computation of a 3-layer (including input layer) neural network where the activation function of each hidden unit is the exponential function, namely exp(s) = $e^s$, and the output layer has just one linear neuron. $H$, $\mathbf{w}_{ij}$ and $c_i$ denote the number of neurons in the hidden layer, the weights between an input $j$ and hidden neuron $i$, and the weights between the hidden neuron $i$ and the output unit respectively. The first term $c_0$ represents the bias of the output neuron. The hidden neurons have no biases. Saito's RF5 employs a second-order quasi-Newton learning algorithm called BPQ.

One design issue with neural networks is finding the number of neurons, H, in the hidden layer. RF5 uses Rissanen's MDL criterion for this purpose.

$$MDL = 0.5m \, log(MSE) + 0.5\tilde{N} \, log(m) \tag{5}$$

where m is the number of samples, MSE is the mean square error, $\tilde{N}$ is given by

$$\tilde{N} = nH + H + 1 \tag{6}$$

Using the ANN solution with the lowest MDL ensures an optimal balance between model accuracy (MSE) and model complexity (number of hidden neurons, $H$). The main advantage of RF5 is that most numerical law discovery methods are incapable of finding equations like (3) without preparing appropriate prototype functions before hand. Preparing such functions essentially amounts to pre-guessing the form of the solution! RF5 is also resistant to the presence of irrelevant input variables and noise.

In this study, few modifications were made to the original RF5 algorithm. Normalization of inputs, i.e. centering by subtracting the mean value and dividing by standard deviation is a standard procedure used in neural networks to avoid an ill-conditioned error surface arising out of a vast difference in magnitude between inputs. However, this renders input values in both positive and negative domain. Since equation (4) requires logarithmic values of inputs, this would result in complex values for the output. To circumvent this problem, in this study, all inputs were normalized by dividing with their respective means. This results in normalized input values close to one (assuming input variances are not large) and is very suitable for optimization on the error surface (Oost, 2002). The second modification was to use Levenberg-Marquardt algorithm (Trainlm in MATLAB Neural Network Toolbox) for training instead of BPQ algorithm. Trainlm is known to have the fastest convergence on function approximation problems for networks that contain up to a few hundred weights (Mathworks, 2005).

## 4. NON-LINEAR PROGRAMMING BASED INVERSE MODEL

For us to be able to predict the required input design parameters of a diesel transit bus for a desired acceleration performance, the trained neural network needs to be inverted. However, the inverse problem as given in equation (2) is an ill-posed problem since the inverse mapping is usually a one-to-many mapping. In general, the inverse problem is locally ill-posed in the sense that it has no unique solution and globally ill-posed because there are multiple solutions. One way of tackling this problem is to use non-linear programming (NLP) techniques (Lu et al., 1999). If we consider the problem of inverting the trained feed-forward network, $\mathbf{y} = \boldsymbol{F}(\boldsymbol{\theta})$, the problem is to basically find an input vector **θ** which yields a given output $\bar{y}$. To find various designated inversions for a given output, the inverse problem is formulated as:

Minimize

$$P(\boldsymbol{\theta}) \tag{7}$$

Subject to equality constraint

$$F(W;\boldsymbol{\theta}) - \bar{y} = 0 \tag{8}$$

and inequality constraint

$$\boldsymbol{\rho} \leq \boldsymbol{\theta} \leq \boldsymbol{\gamma} \tag{9}$$

where $P(\boldsymbol{\theta})$ is the objective function to be minimized, $\boldsymbol{\rho}$ and $\boldsymbol{\gamma}$ are the constant vectors representing the lower and upper bounds for the inputs, W is the weight matrix of the forward ANN and **θ** is the input vector. The equality constraint ensures that the forward mapping of the ANN is satisfied. The inequality constraint on the input vector is introduced to restrain the obtained inversions within a meaningful range of network inputs i.e. design parameters that are expected to result in a transit bus! The nature of the objective function determines the kind of inversions that are computed. In this study, objective function was chosen as $P(\boldsymbol{\theta}) = \| \boldsymbol{\theta} - \boldsymbol{r} \|^2$ , so as to result in an inversion that is nearest to a reference point **r.** The reference point vector was sequentially selected as the input vector corresponding to each bus that was used for training the forward ANN. Hence, in this manner, a number of inversions (equal to the total number of buses in the training set) were obtained. After rejecting the solutions wherein the algorithm had not converged, the solution that was closest to a 'known' bus in terms of Euclidian distance was selected as the optimal solution. It is felt that this shall result in a configuration that would give the desired output with only minimum changes being made to the configuration of a 'previously known' bus. This feature is desirable from the point of view of manufacturability of the 'optimal bus configuration'.

## 5. DATA USED FOR MODELING

Acceleration test data collected on 110 two-axle diesel transit buses, from model year 1990 to 2004, at Pennsylvania Transportation Institute (PTI) have been used in this study. Data were obtained for a total of 21 inputs variables and 4 outputs. The four outputs include acceleration values at speeds of 16, 32, 48 and 64 km/hr. The salient characteristics of the data are given in Tables 1 and 2. The input variable 'Weight to Power Ratio' is a calculated ratio of 'Seated Load Weight (SLW)-Total' and 'Engine Power'.

**Table 1. List of output variables.**

| Sr. No. | Output | Min. | Max. | Mean | Std. Dev. |
|---|---|---|---|---|---|
| 1 | Acceleration at a speed of 16 km/hr, ($m/s^2$) | 0.70 | 2.19 | 1.31 | 0.34 |
| 2 | Acceleration at a speed of 32 km/hr, ($m/s^2$) | 0.60 | 1.80 | 1.05 | 0.27 |
| 3 | Acceleration at a speed of 48 km/hr, ($m/s^2$) | 0.46 | 1.52 | 0.81 | 0.22 |
| 4 | Acceleration at a speed of 64 km/hr, ($m/s^2$) | 0.27 | 1.28 | 0.58 | 0.20 |

**Table 2. List of input variables.**

| Sr. No. | Input | Min. | Max. | Mean | Std. Dev. |
|---|---|---|---|---|---|
| 1 | Length (m) | 6.17 | 12.54 | 9.99 | 1.75 |
| 2 | Width (m) | 2.16 | 2.69 | 2.49 | 0.08 |
| 3 | Height (m) | 2.56 | 3.50 | 3.05 | 0.17 |
| 4 | Wheel Base (m) | 3.10 | 7.59 | 5.44 | 1.22 |
| 5 | Front Overhang (m) | 0.69 | 3.04 | 1.67 | 0.69 |
| 6 | Rear Overhang (m) | 0.762 | 5.54 | 2.86 | 0.52 |
| 7 | Ground Clearance (m) | 0.08 | 0.34 | 0.22 | 0.05 |
| 8 | Curb Weight - Front (kN) | 14.02 | 46.06 | 30.47 | 8.65 |
| 9 | Curb Weight - Rear (kN) | 19.58 | 89.89 | 53.21 | 20.82 |
| 10 | Curb Weight - Total (kN) | 34.26 | 128.87 | 83.69 | 27.83 |
| 11 | Seated Load Weight - Front (kN) | 14.15 | 57.32 | 35.68 | 11.65 |
| 12 | Seated Load Weight - Rear (kN) | 24.59 | 109.96 | 68.72 | 22.55 |
| 13 | Seated Load Weight - Total (kN) | 41.16 | 158.24 | 104.35 | 32.75 |
| 14 | Gross Vehicle Weight - Front (kN) | 14.02 | 64.97 | 39.73 | 14.76 |
| 15 | Gross Vehicle Weight - Rear (kN) | 24.59 | 123.84 | 75.55 | 25.63 |
| 16 | Gross Vehicle Weight - Total (kN) | 41.61 | 192.95 | 115.33 | 39.55 |
| 17 | Axle Ratio | 2.9 | 6.6 | 4.4 | 0.7 |
| 18 | Engine Power (kW) | 123.09 | 223.8 | 166 | 25.22 |
| 19 | Engine Displacement ($cm^3$) | 4250.8 | 10831.8 | 7211.5 | 1368.3 |
| 20 | Wheel Diameter (m) | 0.41 | 0.57 | 0.53 | 0.06 |
| 21 | Weight to Power Ratio (kN/kW) | 0.26 | 0.92 | 0.62 | 0.15 |

## 6. INPUT SELECTION

Input selection is of crucial importance for any model. If relevant inputs are neglected, then model accuracy is sacrificed. On the other hand, inclusion of many irrelevant inputs can result in unnecessary complexity, wastage of computing resources and decrease in model accuracy due to misleading of the learning process. So a proper balance is required between these two conflicting requirements. A two stage approach has been used for input selection in this study. The first stage involves use of correlation matrix to select those inputs that are least correlated with other inputs. A cutoff value magnitude of 0.7 was chosen for the correlation coefficient since a value of 0.7 means that less than 50 % of the variation in one variable can be explained by the other (Muthiah, 2006). The second stage uses Information Theoretic Subset Selection (ITSS) method that is based on Shannon's information theory (Sridhar et al., 1998).

Shannon's information theory provides a means for quantifying the information content of any input vector $\mathbf{x}$. Let, $\mathbf{x}$ take on M discrete values $\mathbf{x_1}$, $\mathbf{x_2}$……$\mathbf{x_M}$. If $N_i$ is the number of occurrences of the vector $\mathbf{x_i}$ in the data set and N is the total number of samples in the data set, then the probability that $\mathbf{x}$ takes a value $\mathbf{x_i}$ can be defined as

$$p_i = \frac{N_i}{N} \tag{10}$$

The entropy or information presented by variable **x** can then be written as

$$H(\boldsymbol{x}) = -\sum_i p_i \ln(p_i) \tag{11}$$

The joint entropy of two vectors **x** and **y** (output) can now be similarly defined as

$$H(\boldsymbol{x}, \boldsymbol{y}) = -\sum_{ij} p_{ij} \ln(p_{ij}) \tag{12}$$

where $p_{ij}$ is the probability that **x** will take on the value $\mathbf{x_i}$ and **y** will take on the value $\mathbf{y_j}$ simultaneously. The entropy of **y** given **x**, $H(\mathbf{y}|\mathbf{x})$, is a measure of the information in the vector **y** when **x** is known. It can be shown that

$$H(\boldsymbol{y} \mid \boldsymbol{x}) = H(\boldsymbol{x}, \boldsymbol{y}) - H(\boldsymbol{x}) \tag{13}$$

Sridhar et al. then define an asymmetric dependency coefficient (ADC), $U(\mathbf{y}|\mathbf{x})$ that measures the dependency of **y** on **x**

$$U(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{H(\boldsymbol{y}) - H(\boldsymbol{y} \mid \boldsymbol{x})}{H(\boldsymbol{y})} \tag{14}$$

ADC measures the extent to which knowledge of **x** provides information about **y**. Since the input variables in this study are continuous, the input domain was divided into a finite number of regions (called bins) within which $p(\mathbf{x})$ is assumed to be constant. By doing this, the methodology described above for discrete variables was made applicable to continuous variables.

The general methodology followed in ITSS is given below:

1) Calculate $U(\mathbf{y}|\mathbf{x})$
2) Generate a candidate input subset $\mathbf{x_{sp}}$ and determine $U(\mathbf{y}|\mathbf{x_{sp}})$
3) Check if the candidate subset is satisfactory using equation (15) where $\varepsilon$ is an acceptable small loss of information in the input space with respect to predicting the output.

$$U(\boldsymbol{y} \mid \boldsymbol{x}_{sp}) - U(\boldsymbol{y} \mid \boldsymbol{x}) < \varepsilon \tag{15}$$

4) If candidate subset is satisfactory then stop, else go to step (2)

In this study, use of correlation matrix resulted in reducing the number of inputs from the 21 (given in Table 2) to 9. These were width, height, wheel base, rear overhang, ground clearance, seated load weight –total, axle ratio, engine power and displacement. Using ITSS method with 5 bins, this was further reduced to 7 inputs with the rejection of width and rear overhang.

## 7. RESULTS AND DISCUSSION

Data pertaining to a total of 99 buses (90%) were used for training the ANN while the remaining data for 11 buses were used for testing. The forward neural network model was run with the 7 inputs selected using the ITSS method. The number of neurons in the hidden layer was varied from 1 to 5. Five simulation runs were conducted for each hidden neuron number setting. The optimal model was then selected using the Minimum Description Length (MDL) criterion as given in equation (5). For all four outputs listed in Table 1, MDL criterion resulted in just one neuron in the hidden layer. Hence using equation (3), there were only 2 terms in the empirical formulae for acceleration values. The obtained formulae are of the following form

$$A = c_0 + c_1 ( h^a w^b G^c W^d r^e P^f D^g ) \tag{16}$$

where

| | |
|---|---|
| A | - acceleration in $m/s^2$ |
| $c_0$ and $c_1$ | - coefficients in $m/s^2$ |
| h | - normalized height |
| w | - normalized wheel-base |
| G | - normalized ground clearance |
| W | - normalized seated load weight-total |
| r | - normalized axle ratio |
| P | - normalized engine power |
| D | - normalized engine displacement |
| a to g | - respective exponents |

The values obtained for the exponents of 7 normalized input variables and the two coefficients are as shown in Table 3.

**Table 3. Exponents and coefficients obtained using RF5 algorithm.**

| Exponent/ Coefficient | Acceleration (at 16 km/hr) | Acceleration (at 32 km/hr) | Acceleration (at 48 km/hr) | Acceleration (at 64 km/hr) |
|---|---|---|---|---|
| a | 0.1225 | -0.2506 | -0.7361 | -1.1350 |
| b | 0.4892 | 0.4419 | 0.2653 | 0.1428 |
| c | 0.2502 | 0.2375 | 0.2218 | 0.2791 |
| d | -1.6110 | -1.6231 | -1.4010 | -1.3415 |
| e | 0.5231 | 0.4978 | 0.3338 | 0.2776 |
| f | 0.3910 | 0.5535 | 0.8015 | 1.1112 |
| g | -0.1744 | -0.0402 | 0.0890 | 0.0755 |
| $c_0$ | 0.5566 | 0.4373 | 0.4070 | 0.3437 |
| $c_1$ | 0.6734 | 0.5510 | 0.3519 | 0.1937 |

The following observations can be made from Table 3:

1) Seated load weight stands out as the most important variable and its exponent (‘d’) is negative since acceleration capability decreases with increasing weight.
2) As expected, ground clearance (exponent ‘c’) and engine displacement (exponent ‘g’) seem to have little impact on acceleration capabilities of a bus.
3) The influence of vehicle height (exponent ‘a’) on acceleration values increases with increasing speed. This is due to the fact that aerodynamic drag force which is dependent on vehicle frontal area is negligible at low speeds and increases with speed.
4) The exponent (‘b’) for wheelbase decreases with increasing speed. This probably results from the fact that longitudinal load transfer plays a more important role at high acceleration values which occur at lower gears i.e. lower vehicle speeds.

5) The value of the exponent ('e') for axle ratio decreases with increasing speed. Axle ratio has a two fold effect on net tractive force. Firstly, it magnifies the torque coming from the engine to the wheels. Secondly, it plays a major part in determining the loss of tractive force due to inertia of rotating components (engine and drive-train). Since higher speed corresponds to a numerical lower axle ratio, the results seem consistent with known theory.

6) Engine power (exponent 'f') becomes a dominant factor as the speed increases. It is interesting to note that the ratio of the exponents for seated load weight and power approaches -1 as speed increases i.e. acceleration becomes proportional to power to weight (P/W) ratio. The P/W ratio is commonly used to denote acceleration capability (Gillespie, 1992).

Figure 2 shows a sample comparison plot between the actual acceleration values and the ANN output values on the test set of 11 buses for the "Acceleration at a speed of 48 km/hr" case.

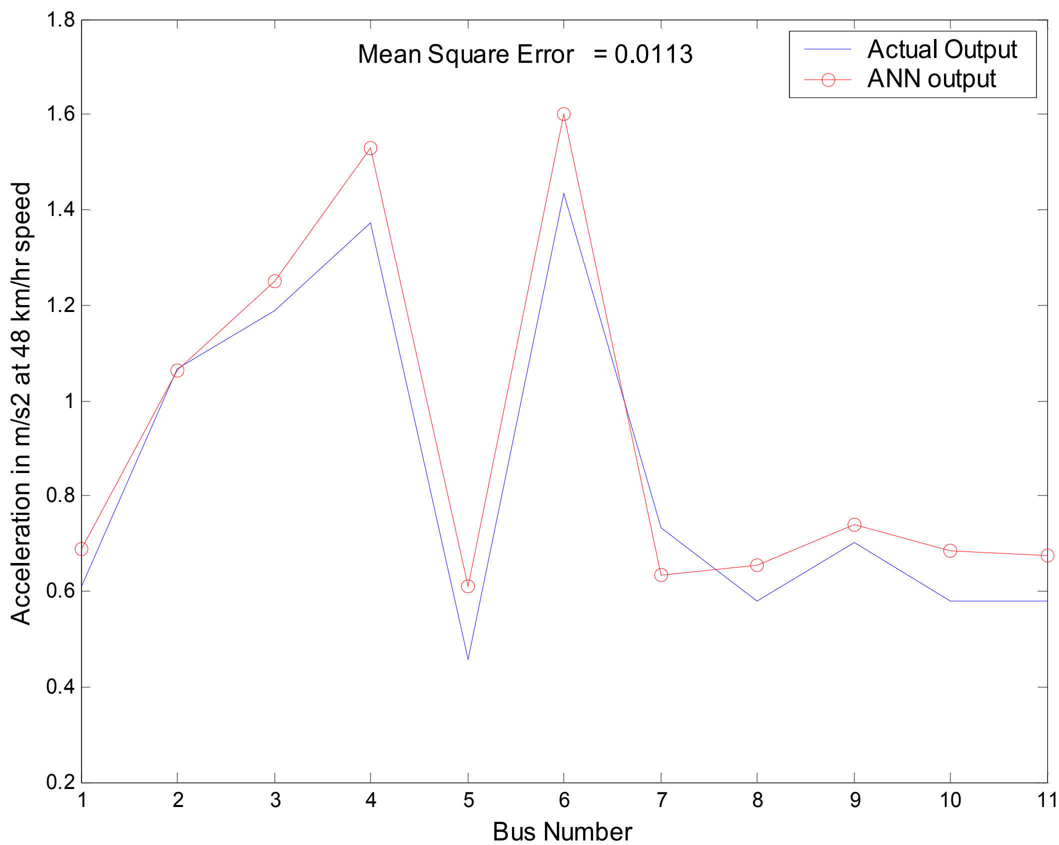


**Figure 2. Comparison plot – Target output and ANN output.**

Once the forward ANN model was developed, it was inverted using the nonlinear programming technique discussed in section 4. A sample simulation was carried out to search for a bus configuration that would give a desired acceleration of 1.0363 m/s$^2$ at 48 km/hr. The results showing the optimal bus configuration and the closest existing bus configuration are given in Table 4.

**Table 4. Optimal bus configuration for a desired acceleration value.**

| Input No. | Input Name | Optimal Bus Configuration | Closest Existing Bus Configuration |
|---|---|---|---|
| 1 | Height (m) | 2.87 | 2.87 |
| 2 | Wheelbase (m) | 4.49 | 4.49 |
| 3 | Ground Clearance (m) | 0.18 | 0.18 |
| 4 | Seated Load Weight – Total (kN) | 60.74 | 60.63 |
| 5 | Axle Ratio | 4.10 | 4.10 |
| 6 | Engine Power (kW) | 137.9 | 138.0 |
| 7 | Engine Displacement ($cm^3$) | 7275 | 7275.9 |

The closest bus configuration had an acceleration value of 1.0516 $m/s^2$ at 48km/hr. Since the other changes are negligible, the result seems to show that the easiest way of achieving the desired acceleration value is to increase the Seated Load Weight on an existing bus by 0.11 kN. This seems reasonable.

It should be noted that equation (16) can easily be modified to accommodate inputs values without normalization by using the mean input values given in Table 2 and adjusting $c_1$ accordingly. At PTI, gradeability values are calculated from acceleration results. Hence one can calculate gradeability from the acceleration value predicted by the neural network model. Alternatively, if gradeability test results are available, one could use them to develop a separate model by following the procedure outlined in this paper.

## 8. CONCLUSIONS

Good acceleration capability and gradeability are required for smooth flow of traffic and minimizing the number of accidents. However, measuring these values is cumbersome. This paper outlines a simulation model for acceleration performance of a transit bus. The model is based on Artificial Neural Network (ANN) and can easily be interpreted as an equation involving product units. This alleviates the common perception of an ANN being a black-box. A procedure for finding the design configuration (if it exists) of a transit bus that results in a desired acceleration performance is also discussed. The inversion model uses nonlinear programming to achieve this objective.

It is hoped that both vehicle designers and regulatory authorities would find this method of analysis useful in their respective fields of designing better vehicles and drafting standards that are optimized to benefit the society.

## REFERENCES

Gillespie, T.D. (1992). Fundamentals of Vehicle Dynamics, “Acceleration Peformance”, pp 21-43. Society of Automotive Engineers, Inc.

Kulakowski, B.T. (2003). "Performance Based Standards – The Time has come!", PBS Seminar: Performance-Based Standards – Moving from Theory to Practice, Melbourne, Australia, February 2003.

Lu, B.L., Kita, H., and Nishikawa, Y. (1999). "Inverting Feedforward Neural Networks using Linear and Nonlinear Programming", IEEE Transactions on Neural Networks, Vol. 10, No. 6, November 1999.

Mathworks (2005). Neural Network Toolbox – Documentation, "Backpropagation – Speed and Memory Comparison".

Muthiah, S., and Kulakowski, B.T. (2006). "Artificial Neural Network (ANN) – An Alternative Numerical Modeling Technique to Evaluate Compliance to Performance Based Standards", Proceedings of the 9th International Symposium on Heavy Vehicle Weights and Dimensions, University Park, PA, USA, June 2006.

Oost, E. (2002). "Opening Pandora's Box, Bottom Side Up", Master's Thesis, University of Amsterdam, The Netherlands, August 2002.

Saito, K., and Nakano, R. (1997). "Law Discovery using Neural Networks", Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97), pp 1078-1083, San Francisco, USA.

Sridhar, D.V., Bartlett, E.B., and Seagrave, R.C. (1998). "Information Theoretic Subset Selection for Neural Network Models", Computers in Chemical Engineering Vol. 22, No. 4/5, pp 613-626, Elsevier Science Ltd., 1998