# Sound Source Identification in Vehicles Using a Combined Linear-Evolutionary Product Unit Neural Network Model

Mª Dolores Redel-Macías, Pedro Antonio Gutiérrez,
Antonio José Cubero-Atienza, and César Hervás-Martínez

**Abstract.** The successful development of new products relies on the capacity to assess the performance of conceptual design alternatives in an early phase. In recent years, major progress has been made hereto, based on the extensive use of prediction models, particularly in the automotive industry. The proposal of this paper is to develop a novel vehicle noise prediction model based on the combination of a Linear Regression model and Evolutionary Product Unit Neural Networks (EPUNNs). Several methods were compared depending on the frequency of the noise since the system has a linear behavior at low frequencies and a more random one at high frequencies. The accuracy of all models has been evaluated in terms of the Mean Squared Error (MSE) and the Standard Error of Prediction (SEP) obtaining the smallest value for both measures when using the combined model (Linear Regression at low frequencies and EPUNN at high frequencies).

## 1 Introduction

The successful development of new products relies on the capacity to assess the performance of conceptual design alternatives in an early design phase. Complex outside noise in a vehicle is originated from various sources. It is vital for automotive engineers to identify the dominant noise sources such as engine noise, exhaust noise and tire noise, etc. Their vehicles have to comply with noise emission

Mª Dolores Redel-Macías · Antonio José Cubero-Atienza
Department of Rural Engineering, University of Cordoba,
Campus of Rabanales,14017, Cordoba, Spain
e-mail: {mdredel,ir1cuata}@uco.es

Pedro Antonio Gutiérrez · César Hervás-Martínez
Department of Computer Science and Numerical Analysis,
University of Cordoba, Campus of Rabanales,14017, Cordoba, Spain
e-mail: {pagutierrez,chervas}@uco.es

E. Corchado et al. (Eds.): SOCO 2011, AISC 87, pp. 379–386, 2011.
springerlink.com 

standards by measuring noise levels according to procedures defined by International Standards, commonly known as Pass-by and Coast-by noise tests. The enhancement effort may be very difficult and time-consuming if one relies only on a 'trial and error' method without prior knowledge about the sources themselves. This important change in viewpoint results in a need for new prediction models and noise abatement techniques. In this context, auralization techniques will play an important role. Sound synthesis models aim at auralizing the sound produced by a physical sound source at an arbitrary receiver location, e.g. [1].

Recently, Artificial Neural Networks (ANNs) have emerged as a powerful learning technique to perform complex tasks in highly nonlinear dynamic environments. Due to their enhanced temporal capabilities, they have been extensively employed in several real world applications, including noise identification and control [2]. In this sense, different types of ANNs are nowadays being used for regression purposes [3], including, among others: MultiLayer Perceptron (MLP) neural networks, where the transfer functions are logistic or hyperbolic tangent functions, Radial Basis Functions (RBFs), General Regression Neural Networks proposed by Specht [4]; Product Unit Neural Networks (PUNNs) [5], etc.

The multiplicative neural networks are among the most recent and interesting neural network models, and they contain nodes that multiply their inputs instead of adding them, which allows inputs to interact non-linearly. This class of multiplicative neural networks comprises such types as sigma-pi networks and product unit (PU) networks [6]. Therefore, this paper investigates the performance of hybrid evolutionary product unit-based neural networks as statistical techniques in the auralization of sound to simulate the listening experience at a given position in the modeled space. Specifically, we present a hybrid evolutionary programming algorithm for automatically obtaining the structure and weights of a PUNN. The proposed method is compared to other standard modeling techniques, in order to assess its performance. The paper is organized as follows: Section 2 introduces the different modeling techniques considered; Section 3 describes the experiments; Section 4 presents the results and Section 5 states the conclusions of the paper.

## 2 Modeling Techniques

Three different models are considered: a linear model, an Evolutionary PUNN (EPUNN) non-linear model and a combined linear-EPUNN model with a linear model for low frequencies and an EPUNN model for high frequencies.

### 2.1 Linear Model

A Linear Regression (LR) model is the best model when a linear (or quasi-linear) relationship can be assumed between the independent variables and the dependent one. This is a staple method in statistics, where the idea is to express the target variable as a linear combination of the attributes, with predetermined weights $f_{LR}(\mathbf{x},\beta_0,\ldots,\beta_k)=\beta_0+\sum_{j=1}^{k}\beta_j x_k$, where $\beta_0,\ldots,\beta_k$ are the weights to be estimated and $\mathbf{x}=(x_1,\ldots,x_k)$ are the input variables. In this work, a ridged linear regression has been performed, using the Akaike Information Criterion (AIC) for selecting the variables of the model. This method is available in the WEKA [7].



### 2.2 Evolutionary Product Neural Networks

Product Unit Neural Networks (PUNNs) are an alternative to MLP or RBF neural networks, and are based on multiplicative nodes instead of additive ones. A multiplicative node is given by $\prod_{i=1}^{k} x_i^{w_{ji}}$, where $k$ is the number of the inputs. As mentioned above, PUNNs have a major handicap: they have more local minima and a higher probability of becoming trapped in them [8].The main reason for this difficulty is that small changes in the exponents can cause large changes in the total error surface. The Back Propagation algorithm, which is the most common algorithm for training multilayer neural networks, does not work very well with the product-units because of its complex error surface. The structure of the neural network considered is the following: an input layer with $k$ nodes, a node for every input variable, a hidden layer with $m$ nodes and an output layer with one node. There are no connections between the nodes of a layer and none between the input and output layers either. The activation function of the $j$-th node in the hidden layer is given by $\Pi_j(\mathbf{x},\mathbf{w}_j)=\prod_{i=1}^{k} x_i^{w_{ji}}$, where $w_{ji}$ is the weight of the connection between input node $i$ and hidden node $j$ and $\mathbf{w}_j=(w_{j1},\ldots,w_{jk})$ the weight vector. The activation function of the output node is given by: $f_{PUNN}(\mathbf{x},\boldsymbol{\theta})=\beta_0+\sum_{j=1}^{m}\beta_j\Pi_j(\mathbf{x},\mathbf{w}_j)$, where $\beta_j$ is the weight of the connection between the hidden node $j$ and the output node. The transfer function of all hidden and output nodes is the identity function.

#### 2.2.1 Evolutionary Algorithm

In this section, we present the EA used to estimate the parameters and the structure of the PUNNs minimizing the prediction error function. The algorithm, similar to that proposed by Martínez-Estudillo et al. [9] begins with the random generation of $N_P$ individuals. Then the evolution process starts and a population-update algorithm is applied. Since the algorithm falls into the class of Evolutionary Programming (EP) paradigm [10], the population is subject to replication and mutation operations, but crossover is not considered, as this operation is usually regarded as being less effective for ANN evolution [6]. Although there are different training methodologies for this purpose, most researchers agree that EP is the most suitable Evolutionary Computation paradigm for evolving neural nets [11]. Fitness of a neural network of the population that implements a function $f(x)$, is calculated using a $D=\{(\mathbf{x}_l, y_l): l=1,2,...,n_T\}$ training dataset, where the number of samples is $n_T$. Mean Squared Error (MSE) of $f(x)$ is considered: $MSE(f) = \frac{1}{n_T}\sum_{l=1}^{n_T}(y_l - f(x_l))^2$, where the $y_l$ are the observed values, and $f(x_l)$ the predicted values. The fitness function $A(f)$ is defined by means of a strictly decreasing transformation of the MSE: $A(f) = 1/1 + MSE(f)$, $0 < A(f) \le 1$. The adjustment of both weights and structure of the PUNNs is performed by the complementary action of two mutation operators: parametric and structural mutation. More details about the EP operators can be found in [9].

### 2.2.2 Hybrid Algorithms

In this work, different variants of hybrid EAs have been applied, all of them proposed by Martínez-Estudillo et al. [9]. The EP algorithm is the EA exposed in the previous section without either a local search or a clustering process. In the Hybrid EP (HEP), the EP is run without the local optimization algorithm and then it is applied to the best solution obtained by the EP in the final generation. This allows the precise local optimum around the final solution to be found. Another version of hybrid EA is the HEP with the Clustering algorithm (HEPC), which applies the clustering process over a large enough subset of the best individuals in the final population. The number of individuals in this subset and the number of clusters to be created are important parameters of the clustering process. Once clusters have been determined, the best individual in each cluster is selected and then optimized using the local search algorithm. The clustering method selected is *k-means* clustering, using a distance measure defined for the vectors of the different values obtained for each individual over the training dataset. Further information can be found in [9]. The local optimization procedure considered is the *improved Rprop* algorithm [12], which is based on Rprop. The Rprop algorithm is believed to be a fast and robust learning algorithm. It employs a sign-based scheme to update the weights in order to eliminate harmful influences of the derivatives' magnitude on the weight updates, i.e., the magnitude of the update along each weight direction only depends on the sign of the corresponding derivative. The size of the update step along a weight direction is exclusively determined. The improved Rprop by a weight-specific "update-value" algorithm (denoted by iRprop+) applies a backtracking strategy. The HEP and HEPC hybrid algorithms are described in Ref. [9].

## 2.3 *Combined Linear Regression and EPUNN Model*

A combined linear regression and EPUNN model is proposed in this paper, following the behavior of the system regarding the frequency:



$$f(\mathbf{x}, h, \boldsymbol{\theta}) = \begin{cases} f_{\mathrm{LR}}(\mathbf{x}, \beta_0, ..., \beta_k), & \text{if } h \le 2.5kHz \\ f_{\mathrm{PUNN}}(\mathbf{x}, \boldsymbol{\theta}), & \text{if } h > 2.5kHz \end{cases}$$

where $f_{\mathrm{LR}}(\mathbf{x}, \beta_0, ..., \beta_k)$ and $f_{\mathrm{PUNN}}(\mathbf{x}, \boldsymbol{\theta})$ models are the linear regression and the EPUNN models, respectively, presented in the sections 2.1 and 2.2. For obtaining this model, only low frequencies are used for fitting $f_{\mathrm{LR}}(\mathbf{x}, \beta_0, ..., \beta_k)$ and only high frequencies are used for adjusting $f_{\mathrm{PUNN}}(\mathbf{x}, \boldsymbol{\theta})$.

## 3 Description of the Dataset and the Experimental Design

The noise sources are located in the vehicle as presented in Fig 1a. The set-up used for this research consisted of a rectangular box with outer dimensions 902 mm x 602 mm x 190mm, with 24 loudspeakers separated from it at a distance of 100 mm. An LMS instrumentation series, consisting of a portable and multi-channel SCADAS meter, a Brüel and Kjaer (B&K) prepolarized free-field half-inch microphone and a prepolarized free-field quarter-inch microphone was employed as the measuring device. LMS Test.Lab was the measurement software package and all the microphones were calibrated with a B&K calibrator. All recordings were carried out inside a semi anechoic chamber. For the engine mock-up sound measurements, the height of the one free-field microphones above the ground was 1.65 m. Additionally, the sound emitted by the source was a random burst noise of between 100-10000 Hz. The sound produced by the source was recorded by the microphone array and by the microphone situated in the receptor position at a distance of 0.10 m from the loudspeaker cabinet, see Fig 1b. The objective was to identify the sound source at the receptor position using the signals of the microphone array situated at 10 cm. The signal registered at receptor position was used for checking the accuracy of the model. The main drawback of the traditional methods is their low resolution at high frequencies and their linear performance at low frequencies. Figure 2 shows the behavior of the system with the

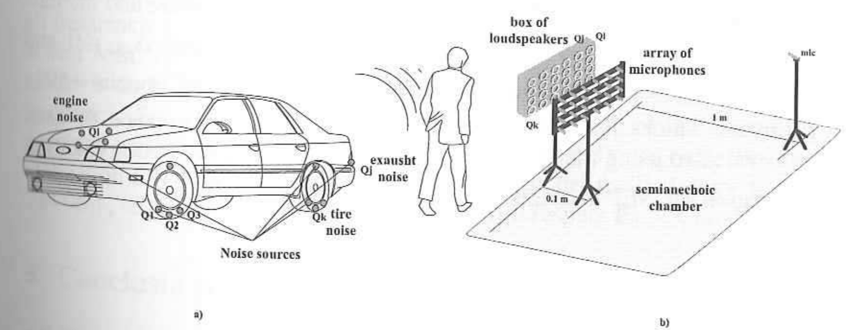


Fig. 1 a) Simulation of auralization process; b) Measurements set-up frequency: the system has a more or less linear behavior for less than 2.5 kHz, changing above 2.5 kHz, where it follows a more random behavior. Therefore, from 8193 registered data, for our purpose, we have employed 6143 data for training, 1500 data of which corresponded to low frequencies and 4643 data corresponded to high frequencies, and 2050 for generalization or test, 501 data at low frequencies and 1549 data at high frequencies, respectively.

The different EPUNN experiments were conducted using a software package developed in JAVA. To start processing data in the EPUNN model, each of the input variables was scaled in the rank [1, 2] by a simple linear rescaling, with $x_i^*$ being the transformed variables.The parameters used in the evolutionary algorithm for learning the EPUNN models are common to all methodologies: the $\mathbf{w}_j$ vector and the coefficients $\beta_j$ are initialized in the [–5, 5] interval; the maximum number of hidden nodes is $m = 4$ ; the size of the population is $N_P = 1{,}000$ .

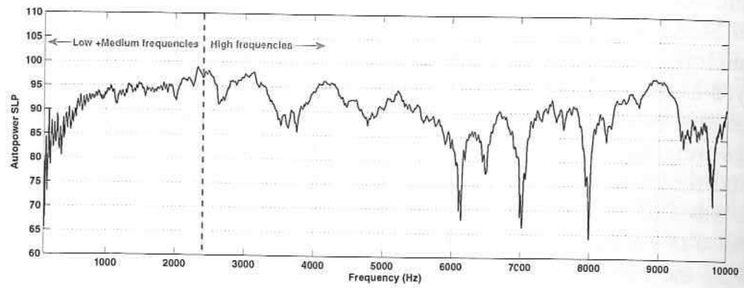


**Fig. 2** Behavior of the system with the frequency

The number of nodes that can be added or removed is within the [1, 3] interval, whereas the number of connections that can be added or removed in a structural mutation is within the [1, 7] interval. The *k-means* algorithm is applied to $N_C = 200$ best individuals in the population. The number of clusters is 4. For iRprop+, the adopted parameters are $\eta^- = 0.2$ and $\eta^+ = 1.2$ (decreasing and increasing factor for stepsize), $\Delta_0 = 0.0125$ (the initial value of the step), $\Delta_{\min} = 0$ and $\Delta_{\min} = 50$ (minimum and maximum stepsize for the weights), and $\text{epochs} = 500$. These parameter values have been taken from [17]. The performance of each model was evaluated using the MSE and the Standard Error of Prediction (SEP) in the generalization set. SEP is defined as: $SEP = (100/|\bar{y}|) \times \sqrt{MSE}$ .

## 4 Results and Discussion

Table 1 presents the results obtained by using the EPUNN models by considering the different hybrid algorithms (EP, HEP and HEPC). It is important to note that two different experiments have been performed with this algorithm: the first one considers only high frequencies in order to obtain a non-linear part for the combined model presented in Section 2.3; the second one considers the whole range of frequencies in order to compare its performance with respect to the combined model. The EP algorithm used for the EPUNN model is a stochastic method and, consequently, we repeat all the methods 30 times and then record the mean and the standard deviation of the error corresponding to the 30 models. The generalization results included in Table 1 clearly establish the HEPC method as the best EPUNN optimization alternative.



**Table 1** Statistical results of the different evolutionary methodologies for the EPUNN model

| | MSE (Mean±SD) | | | |
|---|---|---|---|---|
| | High frequencies | | Low and High frequencies | |
| Method | Training | Generalization | Training | Generalization |
| EP | 0.0204±0.0004 | 0.0196±0.0012 | 0.0222±0.0006 | 0.0227±0.0005 |
| HEP | 0.0147±0.0028 | 0.0149±0.0021 | 0.0143±0.0014 | 0.0159±0.0045 |
| HEPC | 0.0143±0.0029 | 0.0148±0.0021 | 0.0136±0.0007 | 0.0148±0.0009 |

**Table 2** Comparative performance of the LR, EPUNN and combined models

| | Model Performance | | | |
|---|---|---|---|---|
| | Training | | Generalization | |
| Model | MSE | SEP (%) | MSE | SEP (%) |
| LR | 0.019600 | 292.8735 | 0.019200 | 299.7228 |
| EPUNN | 0.011603 | 209.6200 | 0.012811 | 243.7573 |
| LR+EPUNN | 0.012402 | 232.5014 | 0.011708 | 234.0557 |

Consequently, we selected the best HEPC model from the 30 high frequency models and we combined it with a LR model trained using only the low frequency data. The performance of this combined model over the whole range of frequencies was compared with the EPUNN and LR models obtained by considering also all frequencies. The comparison is shown in Table 2, by using MSE and SEP. The lowest MSE for generalization was obtained with the combined model since this model is defined according to the real behavior of the system depending on the frequency. Since MSE values depend on the magnitude of the data, SEP values provided better comparison between different models. In our case, the lowest generalization SEP was achieved by the combined model methodology, resulting in a 234.06%, reducing the SEP by around 78% with respect the LR model.

## 5 Conclusions

An engine noise was experimentally characterized by means of different methods depending on the behavior of the system with the frequency. The data corresponding to noise at low frequencies follow a linear behavior and high frequencies follow a random behavior. Examining this behavior, we have developed three models to ascertain which the best one is for identifying engine noise in vehicles: a linear regression model, an EPUNN and combined model. The combined model obtained a better performance both in MSE and SEP than that obtained with other models for the prediction of engine noise.

**Acknowledgments.** This work has been partially subsidized by the TIN 2008-06681-C06-03 project of the Spanish Inter-Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P08-TIC-3745 project of the "Junta de Andalucía" (Spain). The section corresponding to measurements about noise was supported by Research Group of Noise and Vibration from Katholieke Universiteit Leuven with the supervision of the Professor Paul Sas.

## References

1. Redel-Macías, M.D., Berckmans, D., Cubero-Atienza, A.: Model of identification of sound source. Application to noise engine RIAI 34(7) (2010)
2. Hu, H.: HRTFpersonalitation based on artificial neural network in individual virtual auditory space. Applied Acoustic 69, 163–172 (2008)
3. Haykin, S.: Neural Networks and learning machines, 3rd edn. Prentice-Hall, Englewood Cliffs (2009)
4. Specht, D.F.: A general regression neural network. IEEE Transactions on Neural Networks 2(6), 568–576 (1991)
5. Durbin, R., Rumelhart, D.: Product units: A computationally powerful and biologically plausible extension to backpropagation networks. Neural Computation 1, 133–142 (1989)
6. Angeline, P.J., Aunders, G.M., Pollack, J.B.: An evolutionary algorithm that constructs recurrent neural networks. IEEE Trans. Neural Networks 5(1), 54–65 (1994)
7. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Data Management Systems. Morgan Kaufmann, Elsevier (2005)
8. Ismail, A., Engelbrecht, A.P.: Global optimization algorithms for training product unit neural networks. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN 2000), Como, Italy, vol. 1, pp. 132–137 (2000)
9. Martínez-Estudillo, A.C., Hervás-Martínez, C., Martínez-Estudillo, F.J., García-Pedrajas, N.: Hybridization of evolutionary algorithms and local search by means of a clustering method. IEEE Trans. Syst. Man Cybernetics 36(3), 534–545 (2006)
10. Fogel, D.B.: Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. IEEE Press, New York (1995)
11. Koehn, P.: Combining genetic algorithms and neural networks: The encoding problem. PhD Thesis. University of Erlangen and University of Tennessee, Knoxville (1994)
12. Igel, C., Hüsken, M.: Empirical evaluation of the improved Rprop learning algorithms. Neurocomputing 50(6), 105–123 (2003)

# Predicting the Passenger Request in the Elevator Dispatching Problem

Maite Beamurgia and Rosa Basagoiti

**Abstract.** The elevator group control system is a problem where new approaches are being used to optimize the cab assignment problem. Soft Computing methods can be useful to assist the existing dispatching algorithm, predicting the passengers stop floor, or detecting the type of traffic pattern (up-peak, down-peak, interfloor). In this work, neural networks has been used for predicting from where is going to come the next hall call and then this information is used to park the cabs adequately. An evaluation is carried out, using Dynamic Sectoring algorithm and different service time analyzed. The results show that service level can be improved using neural network for the demand prediction.

## 1 Introduction

Currently elevators are installed in all types of buildings, hospitals, residences and residential buildings. Nowadays, it is unthinkable to build a building without an elevation system. The people that enter and leave the building are users of those elevators. The users do not mind the kind of building where they are, the only thing that they want is to have a good service.

Depending the characteristics of a building such as physical available site, building type or passengers demand, it is common to have more than one elevator. The good service is expected for a single elevator or a group of elevators.

A passenger waiting time from the request of a call until the answer of an elevator should be as less as possible. For this passengers, the least time that they have to wait for the elevator, is better for them, because no so much people like waiting to the elevators for a large period. So, if the passengers do not wait a large period, they consider that the elevator gives them a good service. There is another point to be considered, the energy spent. Nowadays, is increasing the energy

Maite Beamurgia · Rosa Basagoiti
Mondragon University, Electronics and Computing Department, Goiru 2, 20500 Arrasate-Mondragon, Spain
e-mail: {mbeamurgia,rbasagoiti}@eps.mondragon.edu

E. Corchado et al. (Eds.): SOCO 2011, AISC 87, pp. 387–394, 2011.
springerlink.com 