PREDICTION OF WAVEFORMS UNDER THE VARIATION OF INPUT PARAMETERS USING NEURAL NETWORKS

Gabriel OLTEAN*, Alex PRODAN*, Monica RAFAILĂ**, Laura IVANCIU*,
*Technical University of Cluj-Napoca, Romania, str. Memorandumului, nr.28, Cluj-Napoca
**Infineon Technologies, Neubiberg, Germany
Gabriel.Oltean@bel.utcluj.ro

Abstract: This paper introduces a modeling flow for predicting waveforms as a function of parameters, variables in the system generating the waveforms. In order to achieve this goal, a neural network is involved. The model is developed using early-stage simulation data from the automotive industry. Usually, a large amount of data is necessary in order to properly create such a model and successfully train a neural network, and this can be problematic. To address this issue, a model which can be trained with just a handful of characteristics is proposed. The results obtained show that our model can predict waveforms based on input factors with high accuracy.

Keywords: *waveform prediction, neural network, modeling, automotive industry waveforms.*

I. INTRODUCTION

The industry of automotive electronics must reduce the time-to-market while increasing the complexity, therefore the pressure on a thorough verification is high, especially for safety-relevant systems. To cope with this, pre-silicon verification, by modeling and simulation, starting from a system-level, has gained a lot of interest because of the opportunity to verify at early stages the application-fitness [1, 2]. While digital circuitry and software have established means of high-coverage verification, analog and mixed-signal systems need to be verified under all allowed operating conditions, given variations of components and loads. This introduces a lot of overhead since it is not possible to formally (analytically) verify such systems, and a reasonable coverage, when the number of variations is high, translates into a high number of long simulations [3], [4].

Methods which apply heuristics to the outputs of the simulation, in order to predict not-yet simulated points exist: [5], [6]. Such methods are applied here, but not to optimize/study only static outputs, e.g. performance characteristics, but rather full output signals, over the complete time-scale. Predictive models are built, which can estimate the output signals, given a small variation of the system parameters. In this way, the methods are not limited to one-objective optimization, which would focus on only one output (e.g. one time sample) of the system, but rather predict the multi-response output, formed by the time samples of the signal. To address this range-based verification, some methods are available, such as semi-symbolic simulation [7], which are harder to apply on a system-level.

Some existing approaches imply applying statistics on the output values [8], but do not build predictive models of the signal of interest, therefore can only assess the quality of the system, not improve it.

The goal here is to predict output (transient) signals of the system, in any point inside a multi-dimensional, continuous verification space, as a cheap and fast substitute to a new extensive simulation. This must be applied on signals common in control loops of smart power systems, in the automotive industry [9]. Therefore, we focus on signals of reduced complexity i.e. not many transitions between samples (low frequency), and for which some characteristics such as delay, slew rate, maximum value e.g. overshoot, settle time, are the main characteristics of interest, measured and assessed to be within specification limits.

Other important aspects are:

-the number of samples needed to build the model must be reduced i.e. order of magnitude smaller than the Monte-Carlo circuit simulation methods;

-the evaluation time of the underlying results-model must be much smaller, in comparison to the simulation time;

-such model must be of reasonable size, and must be sufficient to estimate new output signals, with respect to the input variations;

The system under study here is a typical ECU (electronic control unit) for which the control signal is influenced both by the DUT (device under test) as well as the load variation, as described in [10]. The exact switch-on time, given as value and pulse duration, is crucial when it comes to driving the squib of the airbag. The SystemC-AMS model is subject to simulations, to extract the output signals corresponding to applied variations on the DUT and Load parameters (Table II in [10]).

In the literature, there are some examples of using Artificial Neural Network (ANN) to address different aspect in waveform processing. Back Propagation Neural Network and Radial Basis Function Neural Network are used to develop behavioural models of a RF power amplifier, the predicted output signal corresponding to sampling points of the amplifier output waveform value [11]. In [12], an ANN is used for detection and classification of electrical

Manuscript received October 28, 2013; revised December 12, 2013

disturbances in three-phase systems. Automatic detection of spikes in electroencephalograms (EEG) can be solved using neural network as it is presented in [13]. In [14], a neural network provides a means of determining a degree of belief for each identified disturbance waveform in Power System.

In order to extract the waveform features, ANNs are usually combined with mathematical analysis, such as Fourier and wavelet transforms, for the generation of signal features which serve as inputs of the neural network [14 - 17].

In this paper we evaluate Neural Networks as an efficient way to develop a model that can predict waveforms (signals), based only on an input set of parameters (factors), given that a significant training data set is provided. One problem is that the necessary data required to properly train a neural network, can vary in size and complexity. To capture the characteristics of a certain set of waveforms, common and differentiating, a very large data set is required. The waveforms to be predicted are represented by their time samples. Using a high enough sample frequency we can collect a lot of data points. The reverse of the medal is the fact that a very large data training set can pose problems in using computer memory (out of memory type errors) and can dramatically increase the training time.

The approach presented in this paper performs a transformation, from the time domain into the frequency domain, followed by a feature selection operation. This way we can select only the most meaningful characteristics, thus reducing the size of the training set. Finally, we need a reverse transformation and waveform reconstruction, once the neural network is properly trained.

II. DEVELOPMENT OF THE MODEL

The block diagram of the model is presented in Figure 1. The model has a number of *N* inputs representing the input parameters (*p*) of the entire data set. *Q* is the number of the outputs representing the samples describing the predicted waveform (s_1, s_2, ..., s_Q).

The development of the entire model, involves three phases:

1. Training data preparation;
2. Neural network training;
3. Output waveform reconstruction.

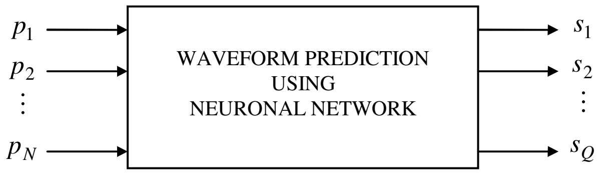

Figure 1. Block diagram of the model.

The detailed model developing diagram that also highlights the phases is presented in Figure 2.

II.1. Training Data Preparation

In order to reduce the complexity of the data used to train the neural network, we resorted to a nominal waveform represented in terms of its time samples S_{nom} (see Figure 2). Applying the Fast Fourier Transform (FFT) on this waveform resulted in changing its domain, from time samples to a new domain consisting of FFT coefficients C_{nom} . A coefficient (feature) selection algorithm was used to select from the new domain, those coefficients that were considered most important ones (according to the magnitude), of 1st order, denoted IC_{nom} . For these 1st order coefficients we are not interested in their value, but in their indices (positions), that are stored as a vector idx_IC_{nom} . The remaining coefficients, considered as 2nd order, IIC_{nom} (values and indices) are stored for later use in the waveform reconstruction phase of the model.

The waveforms in the data set are determined by a combination of input parameters. For each individual set of parameters *p*, the corresponding waveform is selected (see Figure 2). The coefficients C_w result from applying the FFT on the time samples of waveform S_w. From these coefficients only the ones corresponding to the 1st order indices extracted from the nominal waveform were taken into account (IC_w). In this way, the size of the resulted data set can decrease substantially. Because the resulted data set is composed of complex numbers and the neural network cannot be trained with values of this format, an Inverse Fast Fourier Transform (IFFT) is applied to the new data set, which is further used to train the neural network. This way, the output of the neural network will be represented by a collection of predicted time samples ($pred\_IS_w$) corresponding to the position of 1st order coefficients extracted from the nominal waveform.

II.2. Neural Network Training

To develop our model, a fitting neural network [18] is used, with one hidden layer with a sigmoid activation function. The output layer uses a linear activation function. Supervised training is used, meaning that in each training epoch, the parameters of the network (weights and biases) are adapted based on an error calculated as a distance between the target (original waveform samples) and the output (predicted output samples) computed by the neural network.

The input of the network is provided by matrix *P* (equation (1)), composed of elements representing the combinations of parameters that describe each of the waveforms in the data set. The number of rows represents the number of combinations of the input parameters (*M* – length of the data set). The number of columns is given by the number of input parameters that are varied (*N*). The target of the neural network is the *T* matrix (equation (2)) that has a number of rows equal to the number of combinations of the input parameters (*M*). The number of columns represents the number of samples describing each waveform (*Q*).

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \vdots & \vdots & & \vdots \\ p_{M1} & p_{M2} & & p_{MN} \end{bmatrix} \quad (1)$$

$$T = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1Q} \\ t_{21} & t_{22} & \cdots & t_{2Q} \\ \vdots & \vdots & & \vdots \\ t_{M1} & t_{M2} & & t_{MQ} \end{bmatrix} \quad (2)$$

Details regarding the structure and training of the neural network are provided in Section III.

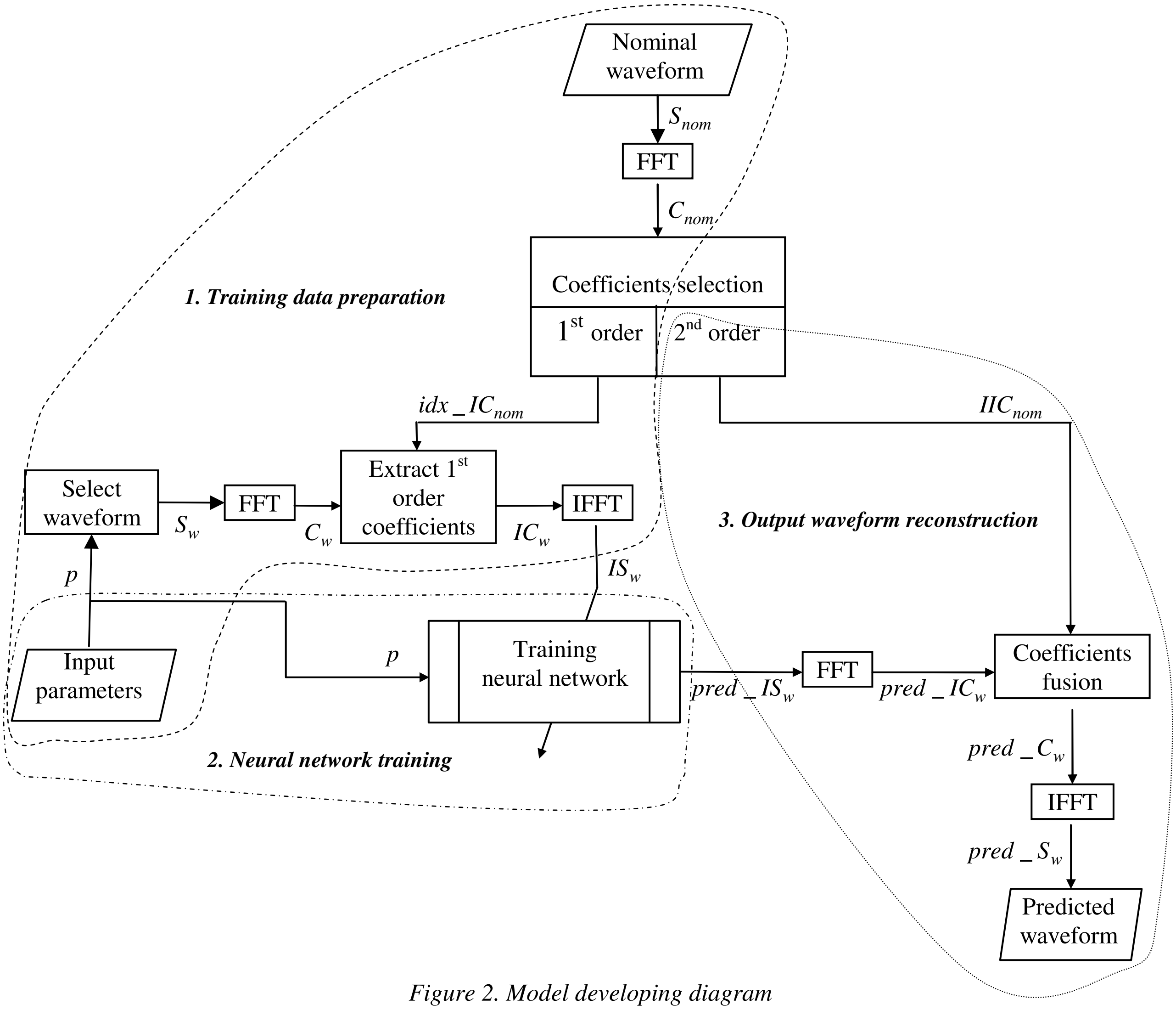

Figure 2. Model developing diagram

II.3. Output Waveform Reconstruction

The reconstruction of the output waveform is realized using both the 2nd order components stored in the data preparation phase, and the 1st order components provided by the output of the neural network.

To do that, a fusion operation (Coefficient fusion – see Figure 2) between 1st order coefficients ($pred\_IC_w$) and 2nd order coefficients (IIC_{nom}) is involved. The 1st order coefficients were obtained by applying a FFT operation on the 1st order samples ($pred\_IS_w$) predicted by the neural network.

Finally, the predicted output waveform is generated by its time samples ($pred\_S_w$) by applying the IFFT once again on the predicted coefficients $pred\_C_w$.

III. EXPERIMENTAL RESULTS

The data used for conducting the experiment consists of a set of 200 waveforms ($M = 200$), generated by 200 combinations of 10 input parameters (N=10). The variation of these factors takes values between -1 and 1. These numbers do not represent absolute values of the parameters, but normalized values, -1 representing the minimum value, and +1 representing the maximum value. Supplementary, a nominal waveform has been provided in order to develop this system. Each of the waveforms used to train the neural network are described by a number of ca.13600 samples, presented in Figure 3.

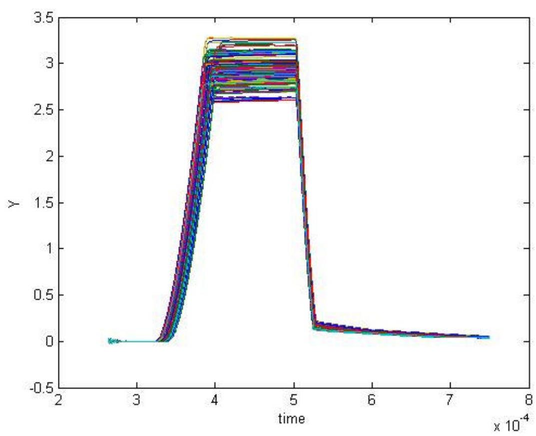

Figure 3. Training Data Set described by 200 waveforms sampled in 13600 points.

According to the model development procedure presented in the diagram in Figure 2, a number of 105 1st order coefficients are selected from the nominal waveform, the rest of the coefficients representing 2nd order components, used to reconstruct the waveform. Undergoing the procedure presented in the diagram, the data set used to train the network contains only 105 samples, instead of 13600. This translates into a substantial data dimensionality reduction, by a factor of 130, which ensures the avoidance of risk concerning memory issues, and also provides a reduced time for neural network training.

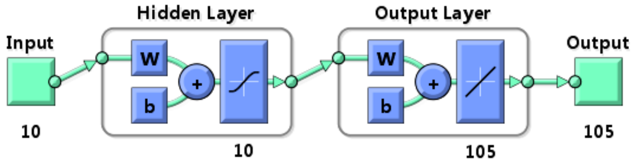

Figure 4. Structure of the neural network.

The structure of the neural network is illustrated in Figure 4. The number of neurons in the hidden layer was selected from a series of trial runs in order to obtain an optimal network structure with minimum error. Finally, the network has 10 neurons in the hidden layer and 105 neurons in the output layer. The input set is provided by the parameter matrix P (equation (1)), consisting of 200 combinations of 10 parameters. The full data set consists of 200 waveforms corresponding to the combinations of the input parameters, sampled in 105 points, describing the main characteristics of the waveforms. For the training procedure, the full data set was split into three data subsets: training subset (70% of the data set), validation subset (15% of the data set) and testing subset (15% of the data set). The training subset is used to train the neural network, adapting its parameters (weights and biases). The validation subset supervises the training, detecting the overfitting phenomenon. The testing subset, considered as an independent data subset, measures the performance of the neural network, inasmuch as that is not at all involved in the training process.

After successfully training the neural network, some analysis concerning the performances of this process was made. The performance validation graph is presented in Figure 5. This figure illustrates the evolution of the mean squared error within the three subsets, over the duration of the training.

In the first 10 training epochs one can see a steep improvement (reduction) of the errors in all data subsets. Then, the training enters the phase of "fine tuning". When the overfitting phenomenon occurs (after 19 epochs), the training stops.

A regression analysis is also performed, the results being presented in Figure 6. The graph illustrates the linear regression of targets (as reference values) relative to the outputs (predicted values) of the neural network. The regression equation is:

$$Target = a \cdot Output + b \qquad (3)$$

where a is the slope of regression fit and b is the offset of regression fit.

An ideal fit (network outputs match the targets exactly) means a=1 and b=0, and a regression value R=1. It is easy to see that our neural network presents extremely good fitting performances in all data subsets. The slope of the regression fit is a ideal one (1 for all subsets), while the offsets have very low values for all subsets (0,0095 for the training subset, 0.22 for the validation subset and 0.059 for the testing subset). The regression value is almost equal to 1 in all cases: $R = 0.99994$ for training, $R = 0.99982$ for validation, $R = 0.99987$ for testing, and $R = 0.99991$ for the entire data set.

This means that our neural network provides very good generalization capabilities; in particular it can correctly predict a waveform corresponding to a new combination of input parameters, combination that was not included in the

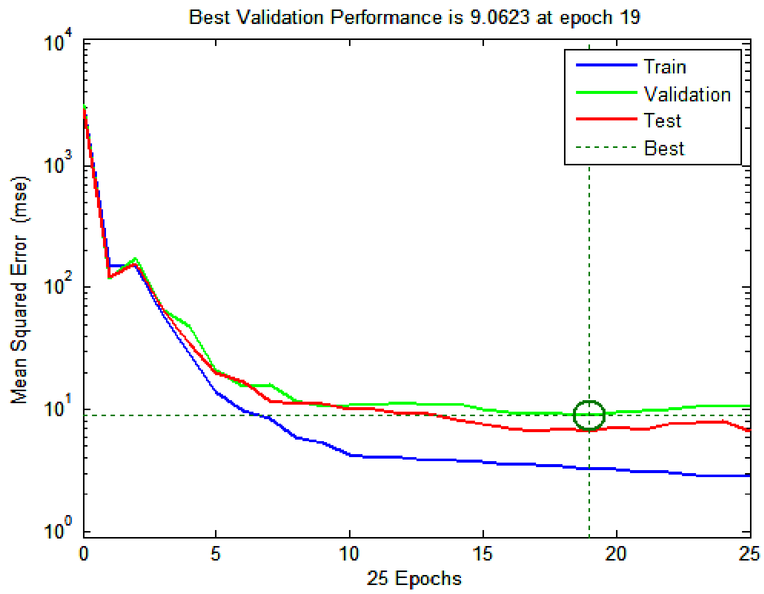

Figure 5. Performance validation graph.

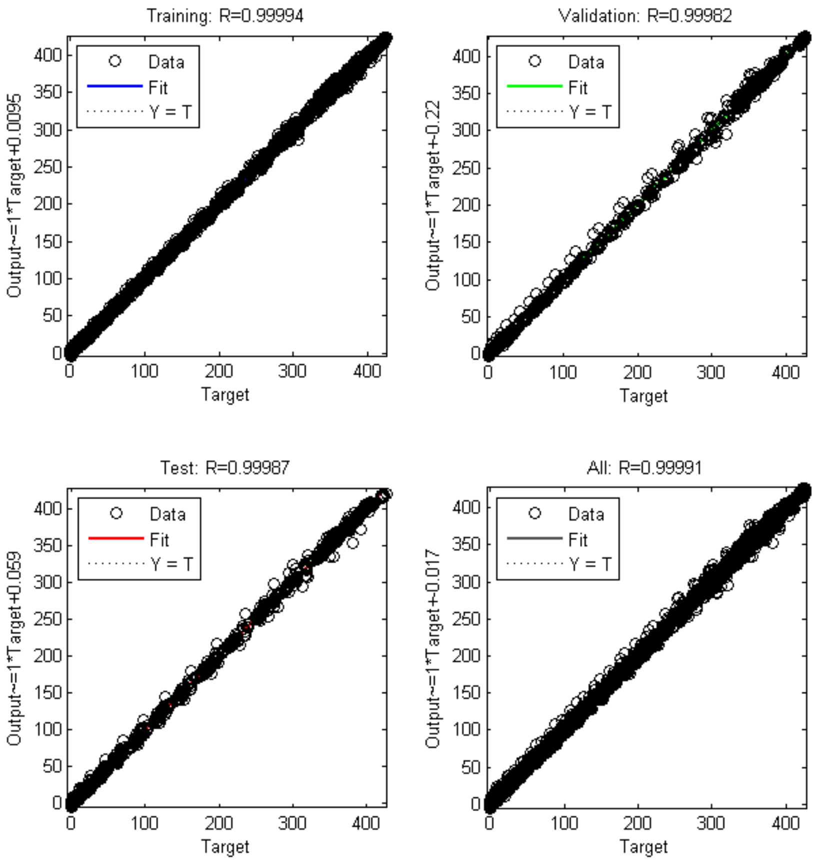

Figure 6. Regression analysis of the neural network

data set that actually trained the network. For all 30 waveforms from the testing subset, the results confirm the previous statement (see graph in lower left side of Figure 6).

To appreciate the goodness of fit of the predicted waveform compared with the corresponding reference waveform, Figure 7 illustrates both an arbitrary chosen reference waveform from the testing subset and the corresponding waveform predicted by the model. We can very easily see the quality of the prediction, namely the two waveforms are almost identical.

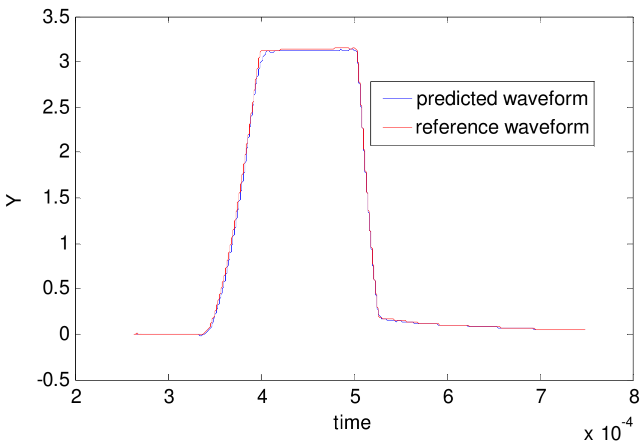

Figure 7. Simulation results

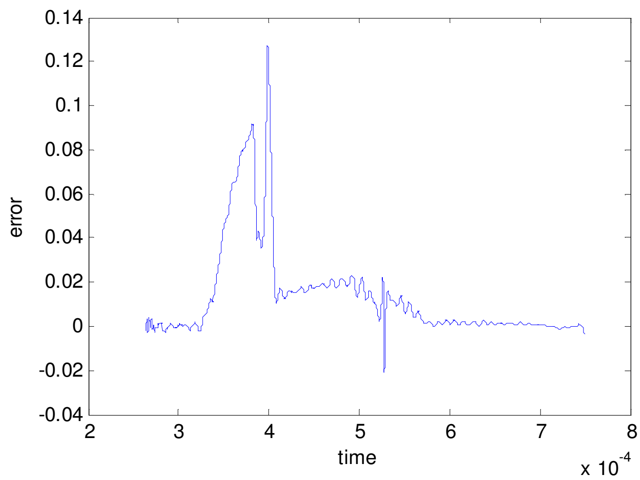

Figure 8. Sample-by-sample error

Figure 8 presents the sample-by-sample error between the reference waveform and the predicted waveform. The error values are both positive and negative, around 0. The maximum error is 0.127, as a difference between the reference values and predicted values, for an expected value of 3.12. In this point we also have the maximum relative error of 4%. It is worth mentioning that the maximum error happens (as expected) in the most difficult region of the waveform with the maximum nonlinearity, where the waveform changes its shape from an almost vertical segment to a horizontal segment. The mean value of the absolute error of the whole waveform is 0.014.

For further analysis of the performance and generalization capabilities of the resulted model, all our 30 waveforms in the testing subset have been used to compare the reference version with the predicted one. Figure 9 illustrates the mean value of the absolute value for all these 30 waveforms. The errors are very small, lying in the [0.0037, 0.019] range, with a maximum error value of 0.019 for the 29th waveform.

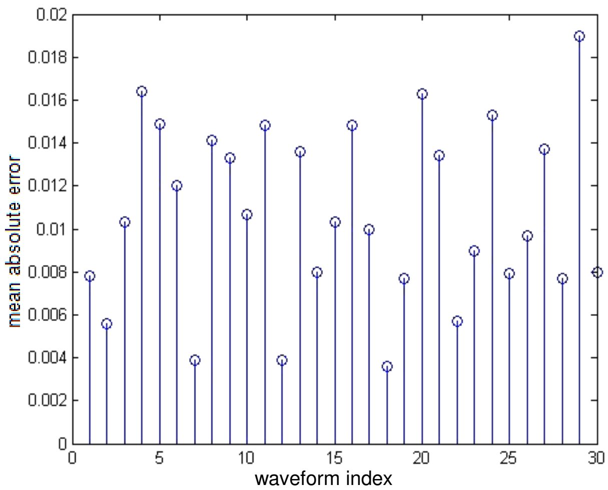

Figure 9. Mean absolute errors for 30 testing waveforms

The above results show that our model can effectively predict accurate waveforms, for any new combination of parameters that were not used at all in the model development phase.

IV. CONCLUSIONS

This paper proposes a model based on a neural network for predicting waveforms, with respect to an input set of parameters. By applying some transformations to the initial data set, which was very large, in terms of samples, we obtained a reduced size of data, further used to train the neural network. Based on the experimental results, our model features very good generalization capabilities. It is able to correctly predict waveforms corresponding to any new combination of input parameters, combination that was not included in the data set used to train the model.

One further research direction to improve the model developing procedure and the prediction accuracy consists in implementing an adaptive coefficient selection algorithm, to assure a better discrimination between the 1st order and 2nd order coefficients of FFT transform. Also, to simplify the procedure, the possibility to train the neural network with real and complex parts of FFT coefficients should be investigated; this way one FFT and one IFFT transform can be eliminated from the model developing procedure.

REFERENCES

[1] MathWorks, *MATLAB online documentation*, Available: http://www.mathworks.com/help/techdoc/, 2010.
[2] SystemC-AMS Working Group. *Requirements specification for SystemC Analog Mixed Signal (AMS) Extensions*, Available: http://www.systemc.org/downloads/standards/, 2008.
[3] SystemC Working Group. *SystemC Verification Standard Specification*, Version 1.0e, Available: http://www.systemc.org/downloads/standards/, 2003.
[4] http://en.wikipedia.org/wiki/Monte_Carlo_method, 2013.
[5] T.W. Simpson, J.D. Peplinski, P.N. Koch, J.K. Allen, “Metamodels for Computer-based Engineering Design: Survey and recommendations”, *Engineering with Computers*, vol. 17, issue 2, pp. 129-150, 2001.
[6] M. Rafaila, C. Decker, C. Grimm, G. Pelz, “New Methods for System-level Verification using SystemC-AMS Extensions: Application to an Automotive ECU”, *12. Workshop Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen*, 2009.
[7] Ch. Grimm, W. Heupke, K. Waldschmidt, “Refinement of Mixed-Signal Systems with Affine Arithmetic”, *Proceedings of the Conference on Design, Automation and Test in Europe*, Volume 1, 2004.
[8] A. Nigam, Q. Tang, A. Zjajo, M. Berkelaar, N. van der Meijs, “Pseudo Circuit Model for Representing Uncertainty in Waveforms”, *Design, Automation & Test in Europe Conference & Exhibition*, 2011.
[9] R. Stone, J. Ball, *Automotive Engineering Fundamentals*, SAE International, 2004.
[10] M. Rafaila, C. Decker, C. Grimm, G. Pelz, Simulation-based “Sensitivity and Worst-Case Analyses of Automotive Electronics”, *IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems*, 2010.
[11] L. Jiuchao, N. Jingchang, Z. Jingmei, “Study and simulation of RF power amplifier behavioral model based on RBF Neural Network”, *Proceedings of the International Conference on Microwave and Millimeter Wave Technology,* pp. 1465 – 1467, 2010.
[12] J.C. Montano, C. Leon, A. Garcia, A. Lopez, I.Monedero, E. Personal, “Random Generation of Arbitrary Waveforms for Emulating Three-Phase Systems”, *IEEE Transactions on Industrial Electronics*, vol. 59, issue 11, pp. 4032 – 4040, 2012.
[13] H.N. Suresh, V. Balasubramanyam, “Wavelet transforms and neural network approach for epileptical EEG”, *Proceedings of the IEEE 3rd International Advance Computing Conference,* pp. 12 – 17, 2013.
[14] K. Shanlin, Z. Huanzhen, K. Yuzhe, “Application of signal processing and neural network for transient waveform recognition in power system”, *Proceedings of the Chinese Control and Decision Conference,* pp. 2481 - 2484, 2010.
[15] I. Monedero, C. Leon, J. Ropero, A. Garcia,J.M. Elena, J.C. Montano, “Classification of Electrical Disturbances in Real Time Using Neural Networks”, *IEEE Transactions on Power Delivery*, vol. 22, issue 3, pp. 1288 - 1296, 2007.
[16] H.M. Rai, A. Trivedi, “ECG signal classification using wavelet transform and Back Propagation Neural Network”, *Proceedings of the 5th International Conference on Computers and Devices for Communication,* pp. 1 - 4, 2012.
[17] A.N. Kannan, A. Rathinam, “High Impedance Fault Classification Using Wavelet Transform and Artificial Neural Network”, *Proceedings of the Fourth International Conference on Computational Intelligence and Communication Networks,* pp. 831 - 837, 2012.
[18] MathWorks, *MATLAB R2013b Documentation. Function Approximation and Nonlinear Regression*, Available: http://www.mathworks.com/help/nnet/function-approximation-and-nonlinear-regression.html, 2013.