<table>
<thead>
<tr>
<th>Title</th>
<th>Functional link neural network-based intelligent sensors for harsh environments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Patra, Jagdish Chandra; Chakraborty, Goutam; Mukhopadhyay, Subhas</td>
</tr>
<tr>
<td>Date</td>
<td>2008</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10220/7254">http://hdl.handle.net/10220/7254</a></td>
</tr>
<tr>
<td>Rights</td>
<td>© 2008 IFSA. This paper was published in Sensors &amp; Transducers Journal and is made available as an electronic reprint (preprint) with permission of IFSA. The paper can be found at the following official URL: [<a href="http://www.sensorsportal.com/HTML/DIGEST/P_SI_38.htm">http://www.sensorsportal.com/HTML/DIGEST/P_SI_38.htm</a>]. One print or electronic copy may be made for personal use only. Systematic or multiple reproduction, distribution to multiple locations via electronic or other means, duplication of any material in this paper for a fee or for commercial purposes, or modification of the content of the paper is prohibited and is subject to penalties under law.</td>
</tr>
</tbody>
</table>
Functional Link Neural Network-based Intelligent Sensors for Harsh Environments

Jagdish C. Patra\(^1\), Goutam Chakraborty\(^2\) and Subhas Mukhopadhyay\(^3\)
\(^1\)School of Computer Engineering, Nanyang Technological University, Singapore
\(^2\)Department of Software and Information Sciences, Iwate Prefectural University, Japan
\(^3\)Department of Electrical & Electronic Engineering, Massey University (Turitea), New Zealand
E-mail: aspatra@ntu.edu.sg, goutam@soft.iwate-pu.ac.jp, S.C.Mukhopadhyay@massey.ac.nz

Received: 15 October 2007 / Accepted: 20 February 2008 / Published: 15 April 2008

Abstract: As the use of sensors is widespread, the need to develop intelligent sensors that can automatically carry out calibration, compensate for the nonlinearity and mitigate the undesirable influence of the environmental parameters, is obvious. Smart sensing is needed for accurate and reliable readout of the measurand, especially when the sensor is operating in harsh environments. Here, we propose a novel computationally-efficient functional link neural network (FLNN) that effectively linearizes the response characteristics, compensates for the nonidealities, and calibrates automatically. With an example of a capacitive pressure sensor and through extensive simulation studies, we have shown that the performance of the FLNN-based sensor model is similar to that of a multilayer perceptron (MLP)-based model although the former has much lower computational requirement. The FLNN model is capable of producing linearized readout of the applied pressure with a full-scale error of only ±1.0% over a wide operating range of −50 to 200\(^\circ\)C. Copyright © 2008 IFSA.

Keywords: Smart sensor, Harsh environment, Functional link neural network

1. Introduction

We begin by quoting Brian Betts [1]: "Chances are, your health and happiness rely on sensors, those ubiquitous little devices that tell us if a fridge is too cold, a nuclear reactor’s safety systems are operating, or a factory production line is processing components correctly. But sensors have a dirty little secret: its all too easy for them be in perfect working order, reporting all is well when, in fact, your milk is turning into a frozen block, the reactor’s safety system is impotent, and that factory has filled a warehouse with useless and possibly dangerous products.”
Different types of sensors, for examples, temperature, pressure, flow, humidity, etc., are used in industrial processes, automobiles, robotics, avionics and other systems to monitor and control the system behavior. In addition, precise, accurate and low power sensors are also needed in the recently emerged wireless sensor networks for applications in intelligent homes, habitat monitoring and war-field applications. Therefore, it is of prime importance that the sensor's output truly represents the physical quantity for which it is deployed.

All the sensors exhibit some amount of nonlinear response characteristics. In addition, the sensor characteristics are affected by the environmental conditions in which it operates. The sensor's output depends not only on the primary input that is to be measured, but also on the operating condition. For example, in case of a pressure sensor, its output depends on the applied pressure as well as on the environmental temperature and humidity (disturbing parameters), because of the geometrical structure of the sensor and the sensing material used. Another associated problem is that the dependence of sensor response on the disturbing parameter(s) may not be linear. This further exaggerates the problem of obtaining an accurate, precise and reliable readout from a sensor.

To tackle this problem, several techniques have been proposed. For compensation of offset capacitance, temperature dependence and for auto-calibration, switched capacitor-based techniques [2], and a ROM and over-sampling delta-sigma demodulation techniques [3], [4] have been reported. Some digital signal processing-based, both iterative and noniterative, techniques for pressure sensor compensation can be found in [5]-[7]. Under the assumptions that the range of variation of disturbing parameters is small and that these parameters influence the sensor characteristics linearly, these techniques provide limited solutions to this complex problem. Neural network (NN)-based sensor compensation techniques perform better than those of classical methods of data interpolation and least mean square regression [8]-[10]. Application of NNs, for compensation for environmental dependency and nonlinearities of pressure sensor [11]-[12], magnetic field measurement [13], eddy-current displacement transducer [14] and Wheatstone bridge transducer [15], with superior performance have been reported.

In the above NN techniques, mostly multilayer perceptron (MLP)-based approaches have been proposed. One major drawback of this network is that it is computationally intensive and therefore consumes a large amount of time for its training. In this paper we present a novel computationally efficient functional link neural network (FLNN). The FLNN is single layer architecture. The input signals first undergo a nonlinear transformation using trigonometric polynomials and then the expanded input pattern is applied to a single layer NN. Recently we have shown that FLNN is capable of identification of complex dynamical systems [16] and equalization of digital communication channels [17]. In [18], FLNN-based intelligent models for pressure sensors have been reported. However, the influence of disturbing environmental parameters has not been considered.

In this paper, by taking an example of a capacitive pressure sensor (CPS), we have shown that the performance of the FLNN-based model is similar to that of the MLP-based model, but the former takes only a fraction of computational time for its training. Through extensive computer simulations and by taking three forms of nonlinear dependencies, we have shown that when the pressure sensor is placed in a operating temperature between -50 to 200°C, the maximum full-scale error between the linearized output and the FLNN model output remains within ±1 %.

2. Capacitive Pressure Sensor and Switched Capacitor Interface

A capacitive pressure sensor (CPS) senses the applied pressure in the form of elastic deflection of its diaphragm. The capacitance of the CPS resulting from the applied pressure P at the ambient temperature T is given by:
\[ C(P, T) = C_0(T) + \Delta C(P, T), \]  

where \( \Delta C(P, T) \) is the change in capacitance and \( C_0(T) \) is the offset capacitance, i.e., the zero-pressure capacitance, both at the ambient temperature \( T \). The above capacitance may be expressed in terms of capacitances at the reference temperature \( T_0 \) as:

\[ C(P, T) = C_0(T_0) f_1(T) + \Delta C(P, T_0) f_2(T), \]

where \( C_0(T_0) \) is the offset capacitance and \( \Delta C(P, T_0) \) is the change in capacitance, both at the reference temperature \( T_0 \). The nonlinear functions \( f_1(T) \) and \( f_2(T) \) determine the effect of temperature on the sensor characteristics [3]. This model provides sufficient accuracy in determining the influence of temperature on the sensor response characteristics. When pressure is applied to the CPS, its change in capacitance at the reference temperature \( T_0 \) is given by:

\[ \Delta C(P, T_0) = C_0(T_0) P_N \frac{1 - \tau}{1 - P_N}, \]

where \( \tau \) is a sensitivity parameter, the normalized applied pressure \( P_N \) is given by \( P_N = P / P_{\text{max}} \), and \( P_{\text{max}} \) is the maximum permissible applied pressure. The parameters \( \tau \) and \( P_{\text{max}} \) depend on the geometrical structure and physical dimensions of the CPS. The nonlinear functions \( f_1(T) \) and \( f_2(T) \) control the influence of the ambient temperature on the CPS characteristics, and are given by:

\[ f_i(T) = 1 + \kappa_{i1} T_N + \kappa_{i2} T_N^2 + \kappa_{i3} T_N^3, \]

where \( i = 1 \) and \( 2 \), and the normalized temperature, \( T_N \) is given by \( T_N = (T - T_0)/(T_{\text{max}} - T_{\text{min}}) \). The maximum and the minimum operating temperatures are denoted by \( T_{\text{max}} \) and \( T_{\text{min}} \), respectively. The coefficients, \( \kappa_{ij} \) determine the extent of temperature influence on the sensor characteristics. Note that when \( \kappa_{ij} = 0 \) for \( j = 2 \) and \( 3 \), the influence of the temperature on the CPS response characteristics is linear. Thus, the normalized capacitance at any temperature \( T \) may be expressed as:

\[ C_N = C(P, T) / C_0(T_0), \]

Using (2) and (3) this may be written as:

\[ C_N = f_1(T) + \gamma f_2(T), \]

where \( \gamma = P_N \frac{1 - \tau}{1 - P_N} \). A schematic diagram of a switched capacitor interface (SCI) for the CPS is shown in Fig. 1, in which the CPS is shown as \( C(P) \). The SCI output provides a voltage signal proportional to capacitance change in the CPS due to the applied pressure. The SCI output voltage is given by:

\[ V_0 = K C(P), \]

where \( K = V_R / C_S \). By choosing proper values of the reference voltage \( V_R \) and the reference capacitor


$C_S$, the normalized SCI output voltage $V_N$ may be obtained such that

$$V_N = C_N.$$ \hspace{1cm} (8)

It may be noted that for a fixed applied pressure the SCI output changes when the ambient temperature changes, thus, giving rise to erroneous sensor readout.

3. MLP and FLNN-based CPS Models

Here we describe the MLP and FLNN-based CPS models used to mitigate the adverse effects of the environmental parameters and to linearize the sensor characteristics.

3.1. The MLP

Fig. 2 shows a schematic diagram of an MLP network used in our study. A two-layer MLP architecture is specified by \{I-J-K\}, where I, J and K denote number of neurons at the input, hidden and output layers, respectively. The MLP is trained using the popular backpropagation (BP) learning algorithm [19]. After application of an input pattern, the error at the output layer at the kth instant is found as $e_i(k) = d_i(k) - y_i(k)$, where $e_i(k)$, $d_i(k)$ and $y_i(k)$ denote the error, desired output and MLP output for the $i$th node. The weights of the MLP are updated using this error (BP algorithm) until the mean square error of the network approaches a minimum value.
3.2. The FLNN

The structure of a FLNN is depicted in Fig. 3. It consists of a functional expansion block and a single layer perceptron network. The main purpose of the functional expansion block is to increase the dimension of the input pattern so as to enhance its representation in a high-dimensional space. This enhanced pattern is then used for modeling of the sensor. Let us denote an \(m\)-dimensional input pattern vector at \(k\)th instant by:

\[
X_k = [x_1(k), x_2(k), ..., x_m(k)].
\]

(9)

![Fig. 3. Schematic of a functional link neural network.](image)

Each element of the input vector is expanded into several terms, by using orthogonal trigonometric polynomials. The \(n\)-dimensional expanded pattern vector obtained from \(X_k\) is given by:

\[
X'_k = [x_1(k), \cos(\pi x_1(k)), \sin(\pi x_1(k)), \cos(2\pi x_1(k)), \sin(2\pi x_1(k)), ..., \\
x_2(k), \cos(\pi x_2(k)), \sin(\pi x_2(k)), \cos(2\pi x_2(k)), \sin(2\pi x_2(k)), ...].
\]

(10)

Thus, using trigonometric polynomials, the \(m\)-dimensional input pattern is enhanced into an \(n\)-dimensional \((n > m)\) expanded pattern, which is then applied to a single-layer perceptron. The FLNN schematic shown in Fig. 3, in which \(m = 2\) and \(n = 9\) have been chosen. In addition, sometimes, a few cross-product terms are also included in the expanded pattern, to improve the original pattern representation in the expanded pattern space. The advantage of FLNN over MLP is that the FLNN is computationally more efficient and as such it takes much less time to train than that of the MLP. More details of FLNN may be found in [16], [17], [20].

3.3. Computational Complexity

We present a comparison of computational complexity between MLP and FLNN neural networks, both trained with the BP algorithm. Let us consider a two-layer MLP structure specified by \(I\), \(J\) and \(K\) number of nodes in the input, hidden and output layers, respectively, excluding the bias units. Whereas, the FLNN has \(D\) input nodes and \(K\) output nodes. Three basic computations, i.e., addition, multiplication and computation of \(\tanh(.)\) are involved for updating the weights of the neural networks.
For the MLP, the increased computation burden is due to the back error propagation by calculating square error derivative at each node in the hidden layer. In one iteration, all computations in the network take place in three phases: (i) forward calculation to find the activation values of all nodes of the entire network; (ii) back error propagation for calculation of square error derivatives; and (iii) updating of the weights of the whole network. In addition, \( \sin() / \cos() \) computations are required for the FLNN.

The total number of weights to be updated in one iteration in the 2-layer MLP is \( J + K + J(I + K) \) whereas in case of FLNN it is \( K(D + 1) \). It may be seen from Table 1 that as hidden layer does not exist in the FLNN, its computational complexity is much lower than the MLP network. In addition, the FLNN requires fewer numbers of weights to achieve similar performance as that of MLP.

### Table 1. Comparison of computational complexity between MLP and FLNN.

<table>
<thead>
<tr>
<th>Operation</th>
<th>MLP {I-J-K}</th>
<th>FLNN {D-K}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition</td>
<td>( 4IJ + 3JK )</td>
<td>( 3K(D+1) )</td>
</tr>
<tr>
<td>Multiplication</td>
<td>( 6J(I+K) )</td>
<td>( 6K(D+1) )</td>
</tr>
<tr>
<td>( \tanh() )</td>
<td>( J+K )</td>
<td>( K )</td>
</tr>
<tr>
<td>( \cos() / \sin() )</td>
<td>(-)</td>
<td>( D )</td>
</tr>
</tbody>
</table>

### 3.4. NN-based Sensor Model

A schematic diagram of the NN-based CPS model is provided in Fig. 4. The ambient temperature and the SCI output are used as inputs to the NN. Appropriate scale factors (SFs) are used to keep these values within 1.0. The desired output is the linearized normalized voltage. During the training phase, an input pattern from the training set is applied to the NN and its weights are updated using BP algorithm. At the end of training, the final weights are stored in an EEPROM. During the second phase, the test phase, the stored final weights are loaded into the MLP. An input pattern from the test set is applied to the NN model and its output is computed. If the NN output and the target output match closely, then it may be said that the NN model has learnt the sensor characteristics satisfactorily.

To illustrate the effectiveness of the NN model to compensate for the nonlinear dependency of temperature on sensor characteristics, three forms of nonlinear functions denoted by NL1, NL2 and NL3 have been selected. A linear function denoted by NL0 is also used for comparison purposes. These nonlinear functions are generated by using different sets of coefficients \( \kappa_{ij} \) in Eq. (4). In this study, the temperature information is assumed to be available. It can be obtained by using a temperature sensor. We carried out two sets of experiments by two implementations of the NN-block as shown in Fig. 4, namely, MLP and FLNN, and compared their performances.

![Fig. 4. NN-based CPS modeling.](image)
4. Simulation Studies

Here we provide the details of the simulation studies carried out for performance evaluation of the proposed MLP and FLNN-based CPS models.

4.1. Preparation of Datasets

All the parameters of the CPS, e.g., the ambient temperature, the applied pressure, and the SCI output voltage, were suitably normalized to keep their values within ±1.0. The SCI output voltage $V_N$ was recorded at the reference temperature ($T_0 = 25^0C$) with different known values of normalized pressure ($P_N$) chosen between 0.0 and 0.6 at intervals of 0.05. Thus, these 13 pairs of data ($P_N$ versus $V_N$) constitute one dataset at the reference temperature. To study the influence of temperature on the CPS characteristics, three forms of nonlinear functions NL1, NL2, and NL3, and a linear form NL0 were generated by selecting proper values of $\kappa_{ij}$ in Eq.(4). The selected values of the $\kappa_{ij}$ are tabulated in Table 2.

Table 2. The selected values of $\kappa_{ij}$ for different nonlinear dependencies.

<table>
<thead>
<tr>
<th>NL form</th>
<th>$\kappa_{11}$</th>
<th>$\kappa_{12}$</th>
<th>$\kappa_{13}$</th>
<th>$\kappa_{21}$</th>
<th>$\kappa_{22}$</th>
<th>$\kappa_{23}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL0</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.20</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>NL1</td>
<td>0.25</td>
<td>-0.25</td>
<td>0.10</td>
<td>0.20</td>
<td>-0.40</td>
<td>0.40</td>
</tr>
<tr>
<td>NL2</td>
<td>0.30</td>
<td>0.10</td>
<td>-0.30</td>
<td>0.20</td>
<td>-0.20</td>
<td>-0.10</td>
</tr>
<tr>
<td>NL3</td>
<td>0.40</td>
<td>-0.15</td>
<td>-0.15</td>
<td>0.25</td>
<td>0.30</td>
<td>-0.60</td>
</tr>
</tbody>
</table>

Next, with the knowledge of the dataset at the reference temperature and the chosen values of $\kappa_{ij}$, the response characteristics of the CPS for a specific ambient temperature were generated using Eq. (6). The response characteristics consist of 13 pairs of data ($P_N$ versus $V_N$), and correspond to a dataset at a specific temperature. For temperature from $-50$ to $200^0C$, at an increment of $10^0C$, twenty-six such datasets, each containing 13 data pairs, were generated. Next, these datasets were divided into two groups: the training set and the test set. The training set, used for training the NNs, consists of only five datasets corresponding to $-50$, $10$, $70$, $130$ and $190^0C$, and the remaining twenty one datasets were used as the test set.

The sensor characteristics for the linear (NL0) and nonlinear (NL1) dependencies at different temperatures and the desired linear response are plotted in Fig. 5. It can be seen that the response characteristics of the sensor change nonlinearly over the temperature range. Besides, the change in response characteristics differs substantially for different forms of nonlinear dependencies. However, it is important to note that, in order to have accurate and precise sensor readout, it should provide linear readout of the applied pressure in spite of the nonlinear sensor characteristics, changes in ambient temperature and nonlinear temperature dependency.
4.2. Training and Testing of NN Models

A 2-layer MLP with \{2-5-1\} architecture was chosen in this modeling problem (see Fig. 2). Thus, the number of nodes including the bias units in the input, hidden and the output layers are 3, 6 and 1, respectively. This MLP contains only 21 weights. Its two inputs are the normalized temperature \((T_N)\), and the normalized SCI output voltage \((V_N)\). The linear normalized voltage, \(V_{Lin}\) was used as the target output for the MLP. Initially, all the weights of the MLP were set to some random values within ±0.5. During training, the five datasets were chosen randomly. The learning parameter \(\alpha\) and the momentum factor \(\beta\) used in the BP algorithm, were selected as 0.3 and 0.5, respectively. For effective learning, 50,000 iterations were run to train the MLP model. To improve learning of the NN, the learning parameter was varied with iteration number. In the case of FLNN, the 2-dimensional input pattern was expanded into 14-dimensional pattern by using trigonometric polynomials (10). Both the learning parameter and the momentum factor were chosen as 0.5. The training was continued for 50,000 iterations. Using a Pentium, 1.10 GHz machine, it took 12 seconds to train the MLP, whereas in case of the FLNN it took only 9 seconds. Note that the number of weights in the MLP and FLNN are 21 and 14, respectively.

Fig. 5. Desired linear characteristics and SCI output, i.e., the CPS response characteristics operating at different temperatures (−50, 0, 25, 100, and 200°C): (a) NL0; (b) NL1.
5. Simulation Studies

Here, based on the results of the simulation study, we provide the performance evaluation of the MLP and FLNN-based models for linearization, auto-calibration and auto-compensation of the CPS.

5.1. Linear Response Characteristics

Both the NN-based models were able to produce linear response characteristics. The results obtained for the linear (NL0) and nonlinear (NL2) temperature dependencies are provided in Fig. 6. The response characteristics of the MLP- and FLNN-based models at different temperatures (−40, 100, 150, and 200 °C) are perfectly linear. For comparison purpose, the upper curve shown represents the sensor characteristics (the SCI output) at the reference temperature (T₀ = 25°C). Note that during training phase, the NNs had not seen the sensor characteristics at these values of temperature. It is observed from this figure that both the MLP and FLNN are able to transfer the nonlinear SCI output voltages (see upper curves in Fig. 5) to linearized values quite effectively over a wide range of temperature for NL0 and NL2 dependencies. Similar observations were also made for the nonlinear dependencies NL1 and NL3 (results not shown here).

Fig. 6. Linearized response characteristics obtained by the NN-based models. The response characteristics shown are for different temperatures of the test set: (a) NL0 (MLP); (b) NL2 (MLP); (c) NL0 (FLNN); (d) NL2 (FLNN).
5.2. Full Scale Error

The full-scale (FS) percent error is defined as

\[
FS\\ Error = 100 \frac{(y_{lin} - y_{est})}{y_{fs}}.
\]  

(11)

where \( y_{lin} \) and \( y_{est} \) denote the desired linearized sensor readout and the NN-model output, respectively. As all the values are normalized to ±1.0, the \( y_{fs} \) is selected as 1.0. The FS error for nonlinear temperature dependencies, NL1 and NL3, over the full range of temperature are plotted in Fig. 7. It may be seen that the FS error remains within ±1.0% for a wide range of temperature from −50 to 200°C (at the specified \( P_N \) values). Note that the NNs were trained only with datasets of five temperature values of −50, 10, 70, 130 and 190°C. Similar observations were made at other values of \( P_N \) for NL0 and NL2 (results not shown here). We have observed that the CPS characteristics changes widely when the environmental temperature changes over a range from −50 to 200°C. Additionally, the environmental temperature influences the sensor characteristics nonlinearly. In spite of these facts, the NN-based models are able to provide an accurate linearized readout of the applied pressure. It is shown that between the MLP and FLNN-based models, the performances of both are similar.

![Fig. 7. Full-scale percent error between the linearized and estimated responses at different PN values (\( P_N \) = 0.0, 0.2, 0.4 and 0.6): (a) NL1 (MLP); (b) NL3 (MLP); (c) NL1 (FLNN); (d) NL3 (FLNN).]
5. Conclusions

A novel NN-based smart sensor that is capable of providing linearized readout, auto-calibration and auto-compensation for the nonlinear influence of the environmental parameters on its characteristics, has been proposed. By taking an example of a capacitive pressure sensor, we have shown that the proposed computationally efficient FLNN model provides satisfactory performance even when it is operating in a harsh environment. We have shown the effectiveness of the FLNN-based model with computer simulated experiments for different forms of nonlinear temperature dependences for a temperature range between −50°C to 200°C. The maximum error between the ideal linearized output and the NN model remains within ±1.0% (FS) for both the NN-based models, though the MLP’s FS error is slightly better. However, the FLNN-based model takes less time for its training due its single-layer structure. The FLNN needs less number of weights to achieve similar performance as that of MLP. Such NN-based models, especially, the FLNN, may be applied to other types of sensors to achieve linearized readout, auto-calibration and to mitigate the nonlinear influence of the environmental parameters on their response characteristics.

References

