Development of Laguerre neural-network-based intelligent sensors for wireless sensor networks

Patra, Jagdish Chandra; Meher, Pramod Kumar; Chakraborty, Goutam

2011

Patra, J. C., Meher, P. K., & Chakrabortty, G. (2011). Development of Laguerre Neural Network-based Intelligent Sensors for Wireless Sensor Networks. IEEE Transactions on Instrumentation and Measurement, 60(3), 725-734.

https://hdl.handle.net/10356/94366

https://doi.org/10.1109/TIM.2010.2082390

Development of Laguerre Neural-Network-Based Intelligent Sensors for Wireless Sensor Networks

Jagdish Chandra Patra, *Member, IEEE*, Pramod Kumar Meher, *Senior Member, IEEE*, and
Goutam Chakraborty, *Senior Member, IEEE*

***Abstract*—The node of a wireless sensor network (WSN), which contains a sensor module with one or more physical sensors, may be exposed to widely varying environmental conditions, e.g., temperature, pressure, humidity, etc. Most of the sensor response characteristics are nonlinear, and in addition to that, other environmental parameters influence the sensor output nonlinearly. Therefore, to obtain accurate information from the sensors, it is important to linearize the sensor response and compensate for the undesirable environmental influences. In this paper, we present an intelligent technique using a novel computationally efficient Laguerre neural network (LaNN) to compensate for the inherent sensor nonlinearity and the environmental influences. Using the example of a capacitive pressure sensor, we have shown through extensive computer simulations that the proposed LaNN-based sensor can provide highly linearized output, such that the maximum full-scale error remains within ±1.0% over a wide temperature range from -50 °C to 200 °C for three different types of nonlinear dependences. We have carried out its performance comparison with a multilayer-perceptron-based sensor model. We have also proposed a reduced-complexity run-time implementation scheme for the LaNN-based sensor model, which can save about 50% of the hardware and reduce the execution time by four times, thus making it suitable for the energy-constrained WSN applications.**

***Index Terms*—Autocompensation, harsh environment, Laguerre neural networks (LaNNs), linearization, smart sensors, wireless sensor networks (WSNs).**

I. Introduction

Wireless sensor networks (WSNs) have a wide range of civil and military applications. In civilian applications, WSNs can be used in the development of smart environments, such as smart homes, smart utilities, and smart transport systems. Some other practical applications of WSN include monitoring of environmental conditions for crops and livestock, irrigation, forest fire detection, flood detection, pollution studies, etc. Smart sensor nodes and actuators embedded in home appliances, such as vacuum cleaners, microwave ovens, refrigerators, and video cassette players and recorders in smart homes, can interact with each other and with any external network via the Internet, so that the end users can manage their home devices locally and remotely [1].

A WSN typically comprises an array of sensor nodes (also called motes) of diverse types interconnected by a wireless communication network. Sensor data are shared between these nodes and used as inputs to a distributed estimation system which extracts relevant information from the environment [1]. Each node consists of three main modules, namely, sensor, processor, and communication modules. The sensor module consists of several sensors, e.g., temperature sensor, pressure sensor, humidity sensor, etc. The analog electrical output of the sensors in response to the physical environmental changes is converted to digital form by the analog-to-digital converters and fed to the processing module. The processor module consists of a microcontroller or a dedicated processor, and the wireless module provides wireless interface for communication.

The functions of WSN generally include the measurement or detection of relevant physical quantities, monitoring and collecting the data, information processing, and generating necessary alerts. The information needed by smart environments is provided by distributed sensor motes, which are responsible for sensing the environment [2]. The important design considerations of a WSN include reliability, accuracy, flexibility, cost effectiveness, and ease of deployment. One of the major challenges of WSN is to maximize the battery life by optimizing different stages of information processing, e.g., data collection, data processing, routing, clustering, and medium access control, modulation, and coding [3], [4]. Recently, several studies have been made on the development of energy-efficient communication protocols [5], clustering and routing [6]–[8], power management [9]–[14], fault tolerance [15], and other issues pertaining to WSN-based systems [16]–[18]. Recently, various other challenges, including the need of accurate sensor readout in the integration of RFID and WSN [19] and underwater sensor networks [20], [21], have been reported. The need of accurate and reliable sensor output is more prominent in fault-tolerant mobile robot networks, as the robots may be operating in hazardous and extreme environmental conditions [22].

Moreover, in order to utilize the precious battery resource optimally, it is necessary to minimize the hardware and the computations needed for data processing in the WSN node. Therefore, in a smart sensor, it is important to reduce the number of computations needed to obtain reliable and accurate readout. The main problem in obtaining accurate readout of

J. C. Patra is with the School of Computer Engineering, Nanyang Technological University, Singapore 639798 (e-mail: aspatra@ntu.edu.sg).

P. K. Meher is with the Department of Embedded Systems, Institute for Infocomm Research, Singapore 138632 (e-mail: pkmeher@i2r.a-star.edu.sg).

G. Chakraborty is with the Department of Software and Information Science, Iwate Prefectural University, Iwate 020-0193, Japan (e-mail: goutam@soft.iwate-pu.ac.jp).

the measurand from a sensor is its inherent nonlinear response characteristics. Therefore, frequent calibration may be needed, particularly when the operating conditions are changed. Another problem is the dependence of sensor characteristics on the environmental parameters. For example, a pressure sensor's output depends not only on the applied pressure (the measurand) but also on the environmental temperature and humidity. Often, due to the physical properties of the sensor, the environmental temperature affects the sensor characteristics nonlinearly, particularly when the environmental temperature undergoes large variations.

In order to get accurate sensor readout, several DSP- and hardware-based compensation schemes have been proposed [23]–[27]. However, the performances of these schemes are not adequate, particularly when the sensor is operating in a harsh environment in which there is a large variation in ambient conditions. Recently, artificial neural networks (ANNs), due to their adaptability, nonlinear processing, fault tolerance, and generalization characteristics, have been successfully applied for solving highly nonlinear problems in pattern recognition, prediction, system identification, and control [28]. It is shown that, in general, ANN techniques perform well for fitting transducer characteristics to measured data [29]. Several interesting applications of ANN-based intelligent systems in the field of instrumentation and measurement have been suggested in [30]–[37]. Using a multilayer perceptron (MLP) network, the compensation for nonidealities in sensors has been proposed in [38]–[41]. These studies show the effectiveness of ANN-based models in providing accurate sensor readout when the sensor is deployed in harsh environments.

In this paper, we propose a computationally efficient ANN, named as Laguerre neural network (LaNN), which can provide similar performance as that of an MLP while involving much lower computational resources. A LaNN-based smart sensor with preliminary results has been reported in [42]. In an energy-constrained wireless sensor node, the computational complexity of smart sensors is an important issue since power consumption increases with computational complexity. In the simulation study, we have taken an example of capacitive pressure sensor (CPS) because it is widely used to sense external pressure due to its high sensitivity and low power consumption. However, the problem with CPS is that its response is highly nonlinear with a large offset voltage and its characteristics depend on the ambient temperature. We have shown, through extensive computer simulations, that the LaNN can linearize the sensor response, with a maximum full-scale (FS) error of only ±1.0% over a temperature range of -50 °C–200 °C for different nonlinear environment models. We have shown that the performance of the LaNN-based sensor is similar to that of an MLP-based model, but the former requires much less computational load than the MLP-based model.

Another major contribution of this paper is the proposed computationally efficient implementation technique which substantially reduces the run-time computations of the LaNN-based sensor. The proposed technique reduces the hardware requirement by about 50% over the direct implementation. The computational savings in LaNN-based sensors is of great advantage in WSNs, as it helps in energy savings as well.

In Section II, we briefly explain the operating principle of a CPS, its temperature dependence, and the switched capacitor interface (SCI). In Section III, we describe the theory of the LaNN and show its effectiveness in two examples of nonlinear system identification problem. In Section IV, we propose a run-time implementation scheme which can provide substantial savings in hardware and execution time. The experimental results of ANN-based sensor models are provided in Section V, and the proposed run-time implementation scheme of the LaNN-based sensor is presented in Section VI. Finally, conclusions of this study are presented in Section VII.

II. CPS and Interface

A CPS senses the applied pressure in the form of elastic deflection of its diaphragm. The capacitance $C(P_N, T_N)$ of a CPS, which is a function of the applied pressure P and ambient temperature T, is given by [25], [41]

$$C(P_N, T_N) = C_0(T_0)f_1(T_N) + \Delta C(P_N, T_0)f_2(T_N) \qquad (1)$$

where the normalized temperature $T_N = (T - T_0)/(T_{max} - T_{min})$, and T_0, T_{min}, and T_{max} denote the reference room temperature and the minimum and maximum ambient temperatures, respectively. The normalized applied pressure is given by $P_N = P/P_{max}$, where P_{max} is the maximum allowed pressure. The offset capacitance at T_0 is denoted by $C_0(T_0)$. The functions $f_1(T_N)$ and $f_2(T_N)$ determine the influence of ambient temperature on the sensor characteristics and are given by

$$f_i(T_N) = 1 + \kappa_{i1}T_N + \kappa_{i2}T_N^2 + \kappa_{i3}T_N^3 \qquad (2)$$

where $i = 1, 2$. The coefficients κ_{ij}, where $i = 1, 2$ and $j = 1, 2, 3$, determine the extent of nonlinear influence of the temperature on the sensor characteristics. Note that, when $\kappa_{i2} = \kappa_{i3} = 0$, the temperature influences the CPS response linearly. The normalized capacitance $C_N = C(P_N, T_N)/C_0(T_0)$ can be written as

$$C_N = f_1(T_N) + \gamma f_2(T_N) \qquad (3)$$

where $\gamma = P_N(1 - \tau)/(1 - P_N)$.

An SCI [25], [43] for the CPS is shown in Fig. 1, where the CPS is shown as $C(P)$. The SCI output provides a voltage signal that is proportional to the change of capacitance of the CPS due to applied pressure. The SCI output voltage is given by

$$V_O = K \cdot C(P_N, T_N) \qquad (4)$$

where $K = V_R/C_S$. By selecting proper values of the reference capacitor C_S and reference voltage V_R, the normalized SCI output V_N may be obtained as

$$V_N = C_N. \qquad (5)$$

Note that the SCI output changes with ambient temperature even if the applied pressure is unchanged, thus giving rise to erroneous sensor readout.

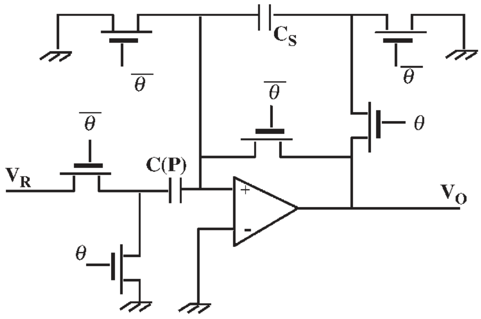


Fig. 1. SCI for a CPS.

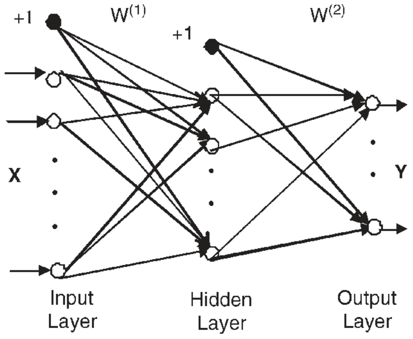


Fig. 2. Schematic diagram of an MLP neural network.

III. LaNN

The operating principle of the LaNN and its computational advantage over MLP are demonstrated here using two examples of nonlinear system identification problem.

A. MLP

Fig. 2 shows a schematic diagram of an MLP neural network. A two-layer MLP architecture is specified by $(I - J - K)$, where I, J, and K denote the number of neurons (excluding the bias unit) in the input and the hidden and output layers, respectively. The MLP is trained using the popular back-propagation (BP) learning algorithm. Let $y(k)$ be the output of MLP for a training input $x(k)$ and $d(k)$ be the desired output at the kth instant. The error at the output layer is found as $e(k) = d(k) - y(k)$. The network weight updating procedure is repeated until the related mean square error (mse) is less than a prespecified value. Details of BP algorithm and MLP training procedure can be found in [28].

B. LaNN

The structure of a LaNN is shown 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 input pattern using the Laguerre polynomials so as to enhance its representation in a higher dimensional space. The enhanced patterns are then used for the modeling of sensor.

The idea of LaNN originated from the functional-link ANNs proposed by Pao and Philips [44], [45]. Laguerre-polynomial-based control algorithms have been found to provide better performance compared to conventional algorithms because these polynomials exhibit good approximation capability for the variances of system time delay, order and other structural parameters, and low computational complexity [46], [47].

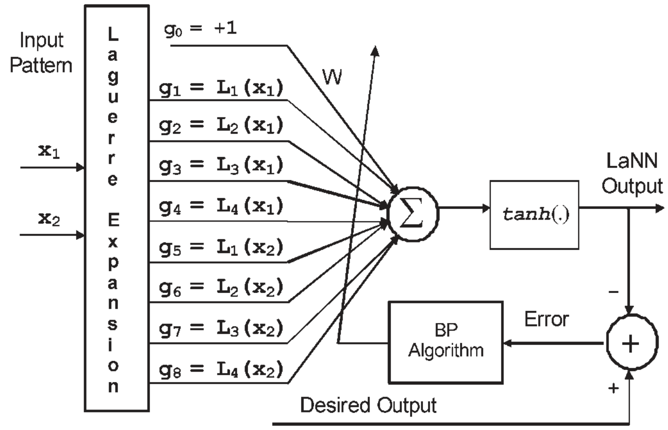


Fig. 3. Schematic diagram of the proposed LaNN.

Laguerre's differential equation, which arises in the treatment of harmonic oscillator in quantum mechanics, is given by

$$xy'' + (1 - x)y' + ny = 0 \tag{6}$$

where n is a nonnegative integer. The Laguerre polynomials $L_n(x)$, where $n = 0, 1, 2, \ldots$, are the solutions to the Laguerre's differential equation. These polynomials form a complete orthogonal set on the interval $0 < x < \infty$ with respect to the weighting function e^{-x}. The first few Laguerre polynomials are given by [48]

$$\begin{aligned} L_0(x) &= 1 \qquad L_1(x) = -x + 1 \\ L_2(x) &= x^2/2 - 2x + 1 \\ L_3(x) &= -x^3/6 + 3x^2/2 - 3x + 1 \\ L_4(x) &= x^4/24 - 2x^3/3 + 3x^2 - 4x + 1. \end{aligned} \tag{7}$$

The higher order Laguerre polynomials may be generated using a recursive formula

$$(n + 1)L_{n+1}(x) = (2n + 1 - x)L_n(x) - nL_{n-1}(x). \tag{8}$$

Let us denote an m-dimensional input pattern vector by

$$\mathbf{x} = [x_1, x_2, \ldots, x_m]. \tag{9}$$

Each element of the input vector is expanded into several terms using the Laguerre polynomials (7) to generate an n-dimensional $(n > m)$ enhanced vector **g**, given by

$$\mathbf{g} = [g_1, g_2, \ldots, g_n]. \tag{10}$$

The expanded pattern **g** is then applied to a single-layer perceptron. In the LaNN schematic in Fig. 3, we have chosen $m = 2$ and $n = 8$, and the bias input is shown as $g_0 = 1$. In some applications, a few cross-product terms (not shown in Fig. 3) may be included to improve the pattern representation in the expanded pattern space. However, it may be noted that there is no straightforward method to determine the order of Laguerre's

TABLE I
COMPUTATIONAL COMPLEXITIES OF MLP AND LaNN

No. of Operations	MLP (I-J-K)	LaNN (D-K)
Addition (N_{ADD})	2IJ + 3K(J+1)	2DK + 3K
Multiplication (N_{MUL})	3IJ + 4JK + 3J + 5K	3DK + 5K
tanh() (N_{TAN})	J+K	K

expansion or the selection of cross-product terms. The optimum selection of these functions is problem dependent, and they are usually selected after several trial experiments. This is an important issue which needs further investigation. Since LaNN has no hidden layer, it is computationally more efficient and takes less training time compared to the MLP network.

C. Training-Time Computational Complexity

Here, we present a comparison of computational complexities between LaNN- and MLP-based training procedures using the BP algorithm. A two-layer MLP is represented by $(I - J - K)$, where I, J, and K denote the number of nodes (excluding the bias unit) in the input and the hidden and output layers, respectively. A LaNN is represented by $(D - K)$, where D and K denote the numbers of input and output nodes, respectively. Three basic computations, i.e., addition, multiplication, and computation of $\tanh(.)$, are involved for updating the weights of ANNs. In the case of MLP, the increased computational burden is mainly due to the computation of $\tanh()$ and the calculation of square-error derivative for each node in the hidden layer [49].

In each iteration, computations are carried out in three phases: 1) forward calculation to find the activation value of all nodes of the network; 2) back error propagation for the calculation of square-error derivatives; and 3) updating of weights of the whole network. The total number of weights to be updated in one iteration in a two-layer MLP is $J(I + 1) + K(J + 1)$, whereas in the case of LaNN, it is $K(D + 1)$. Table I provides the number of computations, i.e., the number of additions (N_{ADD}), the number of multiplications (N_{MUL}), and the number of $\tanh()$ computations (N_{TAN}) for MLP and LaNN in one iteration of BP algorithm. It can be seen in Table I that, due to the absence of the hidden layer in LaNN, its computational complexity is lower than that of the MLP network.

D. LaNN-Based Static System Identification

To compare the training performances and to highlight the computational advantage of LANN over MLP, we have carried out experiments for the identification of two static systems taken from [49]. The input–output relationship of the two systems with $x \in [-1, 1]$ is given by

$$\begin{aligned} f_1(x) &= x^3 + 0.3x^2 - 0.4x \\ f_2(x) &= \frac{4.0x^3 - 1.2x^2 - 3.0x + 1.2}{0.4x^5 + 0.8x^4 - 1.2x^3 + 0.2x^2 - 3.0}. \end{aligned} \tag{11}$$

A schematic diagram of the system identification is shown in Fig. 4. The two nonlinear functions $f_1(x)$ and $f_2(x)$ are identified by using ANN models, first with an MLP and then with a LaNN. A two-layer MLP with (1-4-1) architecture, which provides the best results, was selected after several trials. During training, input x was taken randomly from a uniform distribution in the range of $[-1, 1]$. The BP algorithm with a learning parameter $\alpha = 0.2$ and a momentum factor $\beta = 0.5$ was selected for the training of MLP. The training was carried out for 50 000 iterations.

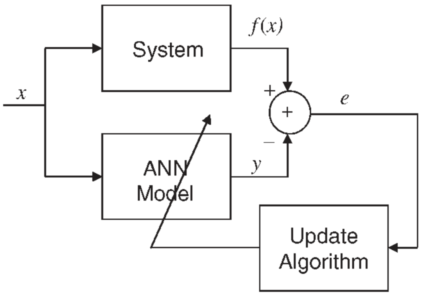


Fig. 4. Schematic diagram of system identification.

In the case of LaNN, the random input x was first translated to a range of [0, 4] and then expanded to g_i, where $i = 1, 2, \ldots, 6$, using the Laguerre polynomials (7), as given in the following:

$$\begin{aligned} x_1 &= 2(x + 1) \qquad x_2 = x x_1 \\ g_1 &= L_1(x_1) \qquad g_2 = L_2(x_1) \qquad g_3 = L_3(x_1) \\ g_4 &= L_1(x_2) \qquad g_5 = L_2(x_2) \qquad g_6 = L_3(x_2) \end{aligned} \tag{12}$$

with $g_0 = 1$ as the bias input. Thus, the LaNN architecture is given by (6-1), where 6 and 1 denote the input dimension (n) and the number of output nodes of the LaNN, respectively. The training of LANN continued for 50 000 iterations using the BP algorithm with $\alpha = \beta = 0.5$. During the test phase, an input in the range $[-1, 1]$ was applied to the ANN model. The estimated outputs from the MLP and LaNN models are shown in Fig. 5. The actual system output and the ANN model estimated output are denoted by "tru" and "ann," respectively, and the error between the two are denoted by "err." The mse in decibels between the actual and estimated outputs is also shown in this figure. In the case of $f_1()$, the LaNN model performs better than the MLP model, whereas in the case of $f_2()$, the performances of both ANN models are comparable.

The numbers of weights in LaNN (6-1) and MLP (1-4-1) are given by 7 and 13, respectively. From Table I, we can find that, in the case of LaNN, the N_{ADD}, N_{MUL}, and N_{TAN} in one iteration of training are 15, 23, and 1, respectively. On the other hand, in the case of MLP, the N_{ADD}, N_{MUL}, and N_{TAN} are 23, 45, and 5, respectively. Thus, there is a large computational savings during the training of LaNN. The training of ANNs was carried out using an Intel Core Duo CPU personal computer with 3.24-GB RAM operating at 2.20-GHz clock. One iteration of BP algorithm to train the LaNN and MLP took 0.92 and 1.52 μs, respectively. Thus, LaNN requires nearly 40% less training time than the MLP.

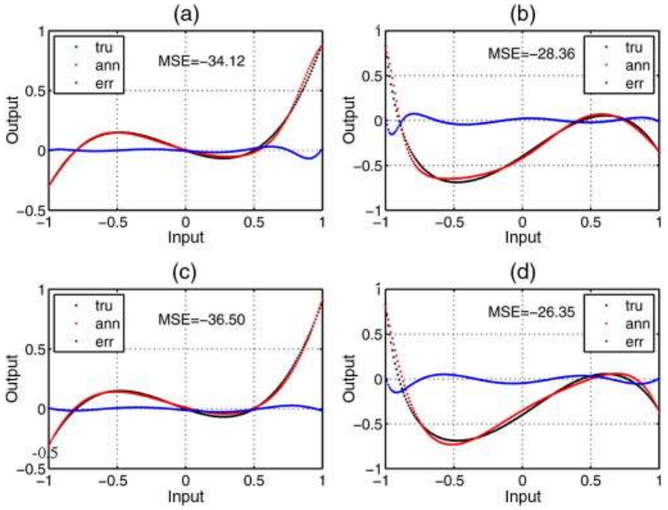
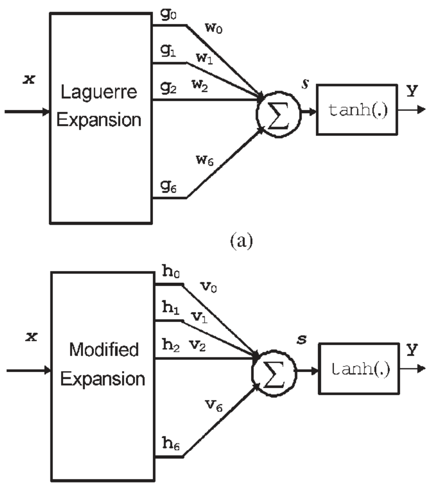


Fig. 5. Actual and estimated results of the ANN models. (a) MLP—$f_1()$. (b) MLP—$f_2()$. (c) LaNN—$f_1()$. (d) LaNN—$f_2()$.



(a)

(b)

Fig. 6. Run-time implementation schemes. (a) Direct computation. (b) Modified computation.

IV. Run-Time Implementation Scheme

In most of the applications, the sensors may be trained offline; therefore, the computational savings in training phase may not be of high significance. However, in energy-constrained WSN applications, computational efficiency is of prime importance during its operations. In this section, we propose a computationally efficient implementation technique for LaNN that will substantially reduce the computations during the run-time operations. After the completion of training, the weights of LaNN are saved in a ROM. The implementation schemes using direct computation and the proposed modified computation are shown in Fig. 6. In the direct implementation scheme, the input x is expanded to g_i, where $i = 1, 2, \ldots, 6$, using (12) and (7). The linear output s is computed as

$$s = \sum_{i=0}^{6} g_i w_i \tag{13}$$

where $g_0 = 1$ is the bias unit and w_i, with $i = 0, 1, \ldots, 6$, denotes the trained weights of the LaNN. The output of the LaNN is computed as $y = \tanh(s)$.

In order to reduce the run-time computations, we propose the following scheme to compute the linear sum s. After expanding (13) and using (7), we can see that s consists of the sum of several terms, each of which is a product of w_i and the sum of polynomials of x_1 or x_2. Rearranging the coefficients of the same power of x_1 or x_2 together, we can compute s as follows:

$$s = \sum_{i=0}^{6} h_i v_i \tag{14}$$

where h_i consists of powers of x_1 or x_2, and v_i consists of the sum of different weights, as given by

$$\begin{aligned} x_1 &= 2(x+1) & x_2 &= x x_1 \\ h_1 &= x_1 & h_2 &= x_2 \\ h_3 &= x_1^2 & h_4 &= x_2^2 \\ h_5 &= x_1^3 & h_6 &= x_2^3 \end{aligned} \tag{15}$$

with $h_0 = 1$ as the bias unit and

$$\begin{aligned} v_0 &= w_0 + w_1 + w_2 + w_3 + w_4 + w_5 + w_6 \\ v_1 &= -(w_1 + 2w_2 + 3w_3) \\ v_2 &= -(w_4 + 2w_5 + 3w_6) \\ v_3 &= w_2/2 + 3w_3/2 \\ v_4 &= w_5/2 + 3w_6/2 \\ v_5 &= -w_3/2 \qquad v_6 = -w_5/2. \end{aligned} \tag{16}$$

Note that the modified weights v_i are computed only once and can be stored in a ROM. The direct computations (13) requires 29 multiplications and 19 additions to obtain the linear sum s, whereas using the proposed modified computations (14)–(16), only 13 multiplications and 7 additions are needed. Thus, a savings of about 50% hardware is achieved by using the proposed implementation scheme. Using an Intel Core Duo CPU personal computer with 3.24-GB RAM operating at 2.20-GHz clock, the direct and modified implementations took 37.9 and 23.9 ns, respectively, thus providing a savings of nearly 37% of execution time. The most computationally expensive circuit in LaNN is the computation of $\tanh()$. However, in LaNN, we need only one such computation. With some approximations, the computation of $\tanh()$ can be carried out efficiently. We plan to present a more detailed discussion on various implementation issues of LANN in a future paper.

V. Experiments With Neural-Network-Based Sensor Models

A schematic diagram of the ANN-based CPS model is shown in Fig. 7, in which the pair of inputs to the ANN is the ambient temperature and the SCI output. To illustrate the effectiveness of the ANN model for mitigating the nonlinear dependence of temperature on sensor characteristics, a linear function denoted by $NL0$ and three nonlinear functions denoted by $NL1$, $NL2$, and $NL3$, representing different environmental models, have been selected. These functions are generated using a set of

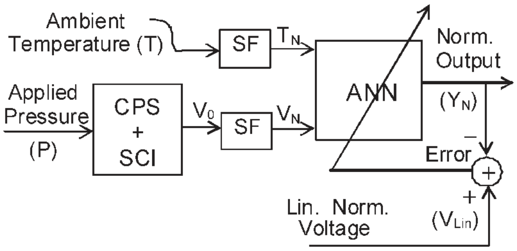


Fig. 7. Schematic diagram of the proposed ANN-based sensor model.

TABLE II
VALUES OF κ_{ij} FOR LINEAR AND NONLINEAR ENVIRONMENTAL MODELS

Environ. Model	κ_{11}	κ_{12}	κ_{13}	κ_{21}	κ_{22}	κ_{23}
$NL0$	0.10	0.00	0.00	0.20	0.00	0.00
$NL1$	0.25	-0.25	0.10	0.20	-0.40	0.40
$NL3$	0.30	0.10	-0.30	0.20	-0.20	-0.10
$NL4$	0.40	-0.15	-0.15	0.25	0.30	-0.60

coefficients κ_{ij} as given in Table II and using (2). The set of coefficients was selected arbitrarily to illustrate the effect of different nonlinear environmental conditions. In this paper, the temperature information is assumed to be available.

A. Generation of Data Sets

All the parameters of the CPS, such as ambient temperature, applied pressure, and SCI output voltage, used in the simulation study, were suitably normalized by selecting appropriate scale factors. The measured CPS response characteristics at room temperature ($T_0 = 25$ °C) were taken from [43]. The SCI output voltage (V_N) was recorded at the reference temperature, 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 data set at the reference temperature.

The CPS response for other temperature values was generated by using the measured response characteristics at the reference temperature and applying the selected values of κ_{ij} (see Table II) in (3). Twenty-six data sets, each containing 13 data pairs for a temperature range from -50 °C to 200 °C with an increment of 10 °C, were generated. Next, these data sets were divided into two sets: training and test sets. The training set consists of only five data sets corresponding to -50 °C, 10 °C, 70 °C, 130 °C, and 190 °C, each with 13 pressure measurement points. We found that, using these training data, we can obtain satisfactory performance of ANN-based sensors in terms of FS error and linearization of sensor characteristics. The remaining twenty-one data sets were used as the test set.

Fig. 8 shows the desired linear response and the actual sensor characteristics (the SCI outputs) for the four environmental models ($NL0$, $NL1$, $NL2$, and $NL3$) at different temperatures. It can be seen that the sensor characteristics change nonlinearly over the temperature range. Moreover, the response characteristics differ substantially between the linear ($NL0$) and the nonlinear ($NL1$, $NL2$, and $NL3$) environmental models. The main objective of the development of an intelligent sensor is that the CPS should provide linear response characteristics (bottom straight line denoted as "des" in Fig. 8) irrespective of its nonlinear characteristic, changes in ambient temperature, and its nonlinear temperature dependence. The focus of this study is to achieve this objective by using the LaNN modeling technique with computationally efficient implementation.

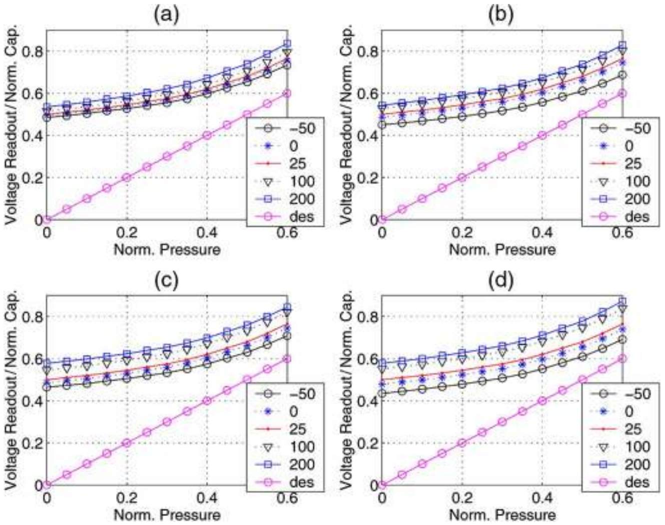


Fig. 8. (Bottom solid line) Desired linear characteristics and the SCI output voltage, i.e., the actual CPS response characteristics, operating at -50 °C, 0 °C, 25 °C, 100 °C, and 200 °C for the four environmental models: (a) $NL0$, (b) $NL1$, (c) $NL2$, and (d) $NL3$.

B. Training and Testing of MLP and LaNN

A two-layer MLP with (2-5-1) architecture (see Fig. 2) was chosen in this modeling problem. We have selected this MLP architecture after several experiments due to its best performance results. The T_N and V_N were used as inputs to the MLP, and the linear normalized voltage V_{Lin} was used as the target output. Initially, all the weights of the MLP were set to random values lying within ±0.5. The learning parameter α and the momentum factor β used in the BP algorithm, were selected as 0.3 and 0.5, respectively. The completion of weight adaptation for the 13 data pairs of all the five training data sets constitutes one iteration. For effective learning, 50 000 iterations were run to train the MLP model. Usually, the learning and momentum parameter values lie between zero to one and are selected by trial [28]. Furthermore, it is seen that the learning of ANN improves when the learning parameter decreases with iteration [28], [41]. Therefore, we used a slowly decreasing learning parameter in which the initial value α_0 was modified in each iteration as

$$\alpha_k = \alpha_0(1 - k/N_t) \qquad (17)$$

where k is the current iteration number and N_t is the total number of iterations (in this case, $N_t = 50\,000$).

In the case of LaNN (see Fig. 3), an (11-1) architecture was selected by expanding the 2-D input pattern to an 11-D pattern, as given in (19) using the Laguerre polynomials (7). Here, both T_N and P_N were expanded using up to fourth-order Laguerre polynomials. In addition, four cross-product terms were also included. The training continued for 50 000 iterations with the initial learning rate and the momentum parameter set at 0.5.

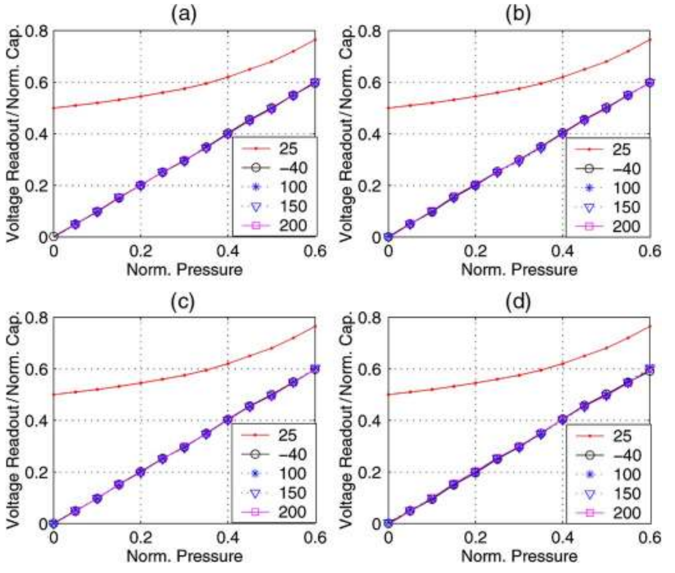


Fig. 9. Linearized response characteristics obtained by the ANN-based models. (a) $NL0$ (MLP). (b) $NL1$ (MLP). (c) $NL0$ (LaNN). (d) $NL1$ (LaNN).

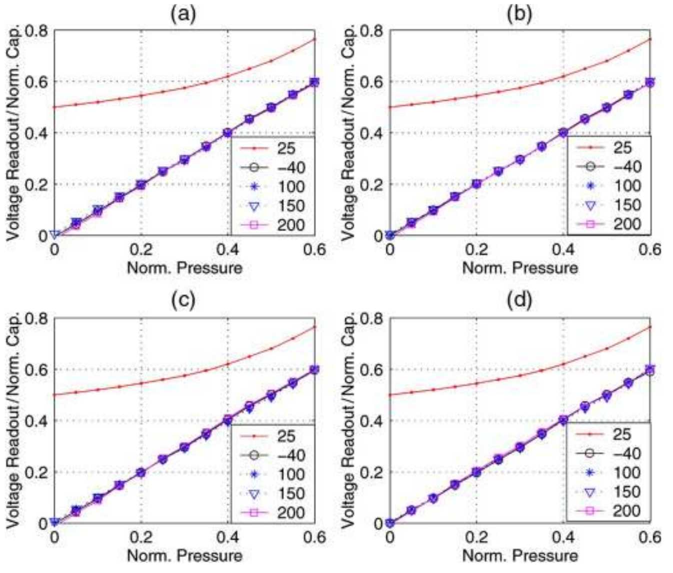


Fig. 10. Linearized response characteristics obtained by the ANN-based models. (a) $NL2$ (MLP). (b) $NL3$ (MLP). (c) $NL2$ (LaNN). (d) $NL3$ (LaNN).

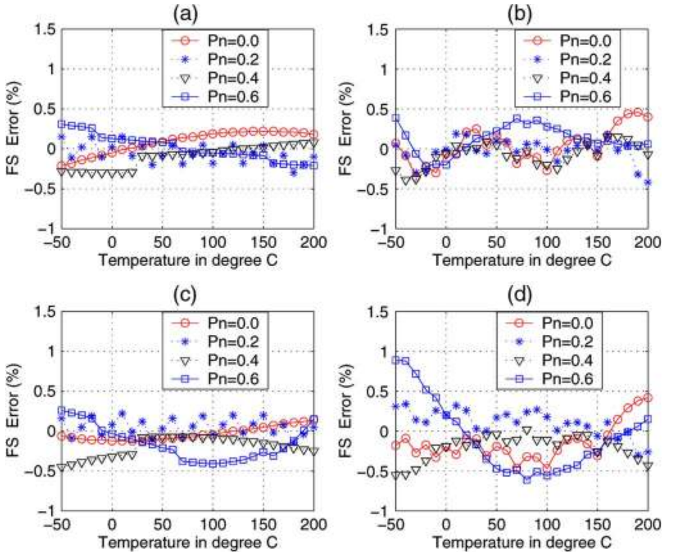


Fig. 11. FS percent errors at $P_N = 0.0, 0.2, 0.4$ and 0.6. (a) $NL0$ (MLP). (b) $NL1$ (MLP). (c) $NL0$ (LaNN). (d) $NL1$ (LaNN).

From Table I, we find that, in the case of MLP, N_{ADD}, N_{MUL}, and N_{TAN} are 38, 70, and 5, respectively. On the other hand, in the case of LaNN, they are 25, 38, and 1, respectively. Using an Intel Core Duo CPU personal computer with 3.24-GB RAM operating at 2.20-GHz clock, it took 83.02 and 43.54 μs to train the MLP and LaNN, respectively, for each iteration. Note that the numbers of weights in the MLP and LaNN are 21 and 12, respectively.

C. Linear Response Characteristics

The results obtained for the two environmental models (linear $NL0$ and nonlinear $NL1$) at different temperatures are shown in Fig. 9. The linearized response characteristics for $NL2$ and $NL3$ are shown in Fig. 10. For comparison purposes, the sensor characteristics (the SCI output) at the reference temperature ($T_0 = 25$ °C) are shown as the upper curve in these figures. As can be seen from these figures, the response characteristics of the MLP- and LaNN-based models are almost linear. Both the MLP and LaNN are able to transform the nonlinear SCI output voltages (upper curves in Fig. 8) to linearized values quite effectively, over a wide range of temperatures, for the linear and the three nonlinear models.

D. FS Error

The FS percent error is defined as

$$FS\ Error = 100(y_{\mathrm{lin}} - y_{\mathrm{est}})/y_{\mathrm{fs}} \tag{18}$$

where y_{lin} and y_{est} denote the desired linearized sensor readout and the ANN-model output, respectively. As all the values are normalized to ±1.0, the y_{fs} is selected as 1.0. The FS errors for the linear $(NL0)$ and nonlinear $(NL1)$ environmental models over the full range of temperature at different P_N values are shown in Fig. 11. Those for $NL2$ and $NL3$ are shown in Fig. 12. From these figures, one can observe that, in the case of linear dependence $(NL0)$, the FS error remains within ±0.5% for both ANN models. However, for the nonlinear environmental models ($NL1$, $NL2$, and $NL3$), the FS error becomes slightly higher but remains within ±1.0%. Note that both the MLP and the LaNN were trained using only five data sets corresponding to −50 °C, 10 °C, 70 °C, 130 °C, and 190 °C.

The FS errors between the estimated and desired responses at specific values of temperature are shown in Fig. 13 (for $NL0$ and $NL1$) and in Fig. 14 (for $NL2$ and $NL3$). From these figures, one can see that the FS error remains within ±0.5% for $NL0$. On the other hand, in the cases of $NL1$, $NL2$, and $NL3$, the FS errors remain within ±1.0% for both ANN models.

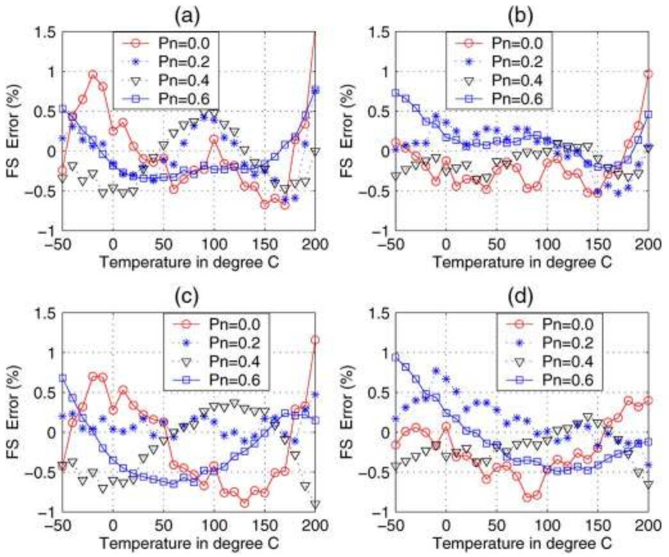


Fig. 12. FS percent errors at $P_N = 0.0, 0.2, 0.4$ and 0.6. (a) $NL2$ (MLP). (b) $NL3$ (MLP). (c) $NL2$ (LaNN). (d) $NL3$ (LaNN).

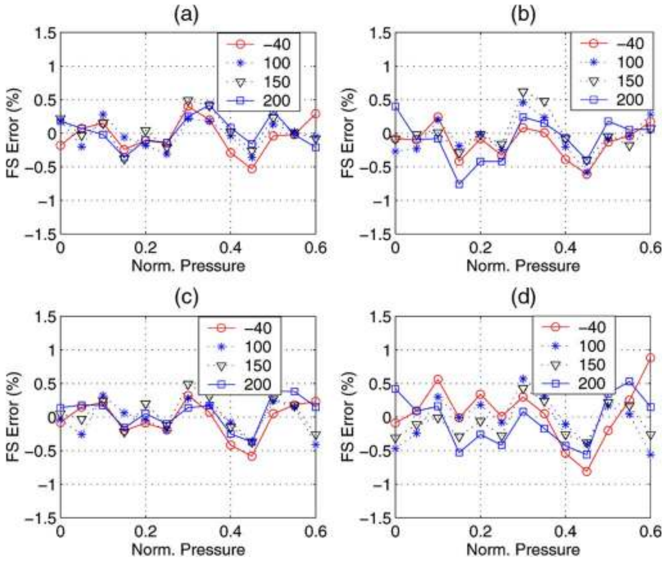


Fig. 13. FS percent error at −40 °C, 100 °C, 150 °C, and 200 °C. (a) $NL0$ (MLP). (b) $NL1$ (MLP). (c) $NL0$ (LaNN). (d) $NL1$ (LaNN).

However, in the case of $NL2$, only at 200 °C, the FS error approaches 1.5% for $P_N = 0$.

VI. Run-Time Implementation of Sensor

In the preceding section, we have shown that LaNN-based CPS models are capable of effectively compensating for nonlinear characteristics and nonlinear environmental dependence. Consequently, they provide the sensor readout with the desired accuracy. Due to limited battery life in a wireless mote, the LaNN may be trained offline. During the actual use, the trained weights of LaNN are stored in the ROM of the microcontroller of the mote. Since the computational complexity of LaNN is much less than that of MLP, the LaNN-based sensor model is preferable in the wireless mote.

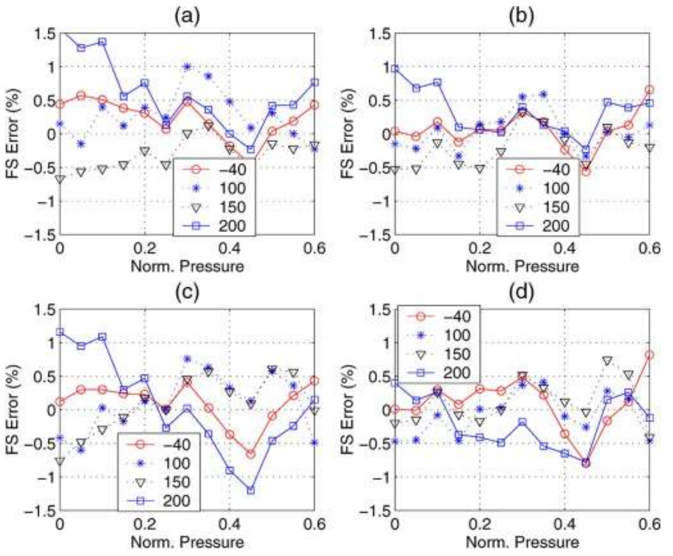


Fig. 14. FS percent error at −40 °C, 100 °C, 150 °C, and 200 °C. (a) $NL2$ (MLP). (b) $NL3$ (MLP). (c) $NL2$ (LaNN). (d) $NL3$ (LaNN).

In Section IV, we have proposed a computationally efficient scheme for the implementation of LaNN. Here, we describe the run-time implementation of LaNN-based sensor model. Let us denote the two inputs of the LaNN sensor model (see Fig. 7) V_N and T_N by x and t, respectively. In the run-time implementation scheme (Fig. 6), there will be two inputs x and t. Using the Laguerre polynomials (7), the 2-D input pattern is expanded to 11-D pattern g_i, where $i = 1, 2, \ldots, 11$, as follows:

$$\begin{aligned} g_1 &= L_1(t) & g_2 &= L_2(t) & g_3 &= L_3(t)\\ g_4 &= L_1(x) & g_5 &= L_2(x) & g_6 &= L_3(x)\\ g_7 &= L_4(x) & g_8 &= xt & g_9 &= g_5 t\\ g_{10} &= g_5 g_2 & g_{11} &= g_6 t. \end{aligned} \tag{19}$$

After the completion of training, let the weights of the LaNN be given by w_i, where $i = 0, 1, \ldots, 11$. The linear sum s [see Fig. 6(a)] is given by

$$s = \sum_{i=0}^{11} g_i w_i \tag{20}$$

where $g_0 = 1$ is the bias unit.

In order to improve the run-time computational efficiency, the modified implementation scheme, as proposed in Section IV, is carried out. Referring to Fig. 6(b) and using (22) and (23), the linear sum s is computed as

$$s = \sum_{i=0}^{12} h_i v_i. \tag{21}$$

The modified expansion is carried out as follows:

$$\begin{aligned} h_1 &= t & h_2 &= x & h_3 &= t^2\\ h_4 &= x^2 & h_5 &= t^3 & h_6 &= x^3\\ h_7 &= x^4 & h_8 &= xt & h_9 &= x^2 t\\ h_{10} &= xt^2 & h_{11} &= x^2 t^2 & h_{12} &= x^3 t \end{aligned} \tag{22}$$

with $h_0 = 1$ as the bias unit. After grouping the coefficients of h_i together, the modified weights v_i are computed as follows:

$$\begin{aligned}
v_0 &= w_0 + w_1 + w_2 + w_3 + w_4 + w_5 + w_6 + w_7 + w_{10} \\
v_1 &= -\, w_1 - 2w_2 - 3w_3 + w_9 - 2w_{10} + w_{11} \\
v_2 &= -\, w_4 - 2w_5 - 3w_6 - 4w_7 - 2w_{10} \\
v_3 &= w_2/2 + 3w_3/2 + w_{10}/2 \\
v_4 &= w_5/2 + 3w_6/2 + 3w_7 + w_{10}/2 \\
v_5 &= -\, w_3/6 \qquad v_6 = -w_6/6 - 2w_7/3 \\
v_7 &= w_7/24 \qquad v_8 = w_8 - 2w_9 + 4w_{10} - 3w_{11} \\
v_9 &= w_9/2 - w_{10} + 3w_{11}/2 \\
v_{10} &= -\, w_{10} \qquad v_{11} = w_{10}/4 \qquad v_{12} = -w_{11}/6.
\end{aligned} \tag{23}$$

Note that the modified weights v_i are computed only once using the known values of the trained weights w_i and then stored in the ROM of the microcontroller. The direct computation of s (20) involves 49 multipliers and 41 adders, whereas using the modified scheme (21), the numbers of multipliers and adders needed are 22 and 12, respectively. This saves about 50% of hardware requirements. Using an Intel Core Duo CPU personal computer with 3.24-GB RAM operating at 2.20-GHz clock, the execution times for the direct and modified implementations are found to be 193.9 and 48.1 ns, respectively. Thus, the proposed implementation scheme provides a computational advantage by a factor of four over the direct implementation, which makes it more attractive for WSN applications.

VII. Conclusion

We have proposed a novel Laguerre ANN-based computationally efficient smart sensor model, which could be used with less computational complexity in WSNs. By taking an example of a CPS, we have shown that the proposed model can provide accurate linearized readout and is able to autocompensate for the nonlinear influence of the environmental parameters on its characteristics. We have compared its performance with that of an MLP-based model and shown the effectiveness of the LaNN-based model under linear and nonlinear influences of the ambient temperature. We have considered a temperature range varying from -50 °C to 200 °C and shown that the FS error of the linearized readout of the ANN models remains within $\pm 1.0\%$ under four different forms of temperature dependences. In order to enhance the battery life of the wireless mote, we have also suggested an implementation scheme of the LaNN-based sensor, which can reduce the run-time computational complexity by a factor of four and save hardware requirements by more than 50% compared to the direct implementation.

One of the issues to be further considered in this modeling technique is the effect of noise, as most of the sensors operate in noisy environments. Another issue is to improve computational efficiency by approximating the $\tanh()$ function. We intend to probe into these aspects in our future study. Because of the computational efficiency, flexibility, fault tolerance, and effectiveness to adapt to dynamic environments, the LaNN-based models may be used for optimizing other aspects of WSNs.

Acknowledgment

The authors would like to thank the anonymous reviewers for their encouraging and positive comments which helped in enhancing the quality of this paper.

References

[1] M. I. Abd-El-Barr, M. A. M. Youssef, and M. M. Al-Otaibi, "Wireless sensor networks—Part I: Topology and design issues," in *Proc. IEEE CCECE/CCGEI*, Saskatoon, SK, Canada, May 2005, pp. 1165–1168.

[2] F. L. Lewis, *Smart Environments: Technologies, Protocols, and Applications*, D. J. Cook and S. K. Das, Eds. New York: Wiley, 2004.

[3] C.-Y. Chong and S. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proc. IEEE*, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.

[4] A. J. Goldsmith and S. Wicker, "Design challenges for energy-constrained ad hoc wireless networks," *IEEE Wireless Commun.*, vol. 9, no. 4, pp. 8–27, Aug. 2002.

[5] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493–506, Jun. 2004.

[6] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Comput. Commun.*, vol. 30, no. 14/15, pp. 2826–2841, Oct. 2007.

[7] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. IEEE INFOCOM*, New York, Jun. 2002, pp. 1567–1576.

[8] P. K. K. Loh, H. W. Jing, and Y. Pan, "Performance evaluation of efficient and reliable routing protocols for fixed-power sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 5, pp. 2328–2335, May 2009.

[9] A. Chamam and S. Pierre, "On the planning of wireless sensor networks: Energy-efficient clustering under the joint routing and coverage constraint," *IEEE Trans. Mobile Comput.*, vol. 8, no. 8, pp. 1077–1086, Aug. 2009.

[10] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1333–1344, Aug. 1999.

[11] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *IEEE Des. Test Comput.*, vol. 18, no. 2, pp. 62–74, Mar./Apr. 2001.

[12] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Resource management in energy-limited, bandwidth-limited, transceiver-limited wireless networks for session-based multicasting," *Comput. Netw.*, vol. 39, no. 2, pp. 113–131, Jun. 2002.

[13] X. Wang, J. Ma, S. Wang, and D. Bi, "Distributed energy optimization for target tracking in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 1, pp. 73–86, Jan. 2010.

[14] X. Tang and J. Xu, "Optimizing lifetime for continuous data aggregation with precision guarantees in wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 4, pp. 904–917, Aug. 2008.

[15] C. Shen, C. Srisathapornphat, and C. Jaikaeo, "Sensor information networking architecture and applications," *IEEE Pers. Commun.*, vol. 8, no. 4, pp. 52–59, Aug. 2001.

[16] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[17] D. Ganesan, "Networking issues in sensor networks," *J. Parallel Distrib. Comput.*, vol. 64, pp. 799–814, Jul. 2004.

[18] R. Rao, S. Vrudhula, and D. Rakhmatov, "Battery models for energy aware system design," *Computer*, vol. 36, no. 12, pp. 77–87, Dec. 2003.

[19] H. Liu, M. Bolic, A. Nayak, and I. Stojmenovic, "Taxonomy and challenges of the integration of RFID and wireless sensor networks," *IEEE Netw.*, vol. 22, no. 6, pp. 26–32, Nov./Dec. 2008.

[20] L. Liu, S. Zhou, and J.-H. Cui, "Prospects and problems of wireless communication for underwater sensor networks," *Wireless Commun. Mobile Comput.*, vol. 8, no. 8, pp. 977–994, Oct. 2008.

[21] Y. Huang, W. Liang, H.-B. Yu, and Y. Xiao, "Target tracking based on a distributed particle filter in underwater sensor networks," *Wireless Commun. Mobile Comput.*, vol. 8, no. 8, pp. 1023–1033, Oct. 2008.

[22] S. Das, H. Liu, A. Nayak, and I. Stojmenovic, "Localized movement control for fault tolerance of mobile robot networks," in *IFIP International Federation for Information Processing, Wireless Sensor and Actor Networks*, L. Orozco-Barbosa, T. Olivares, R. Casado, and A. Bermudez, Eds. Boston, MA: Springer-Verlag, 2007, pp. 1–12.

[23] P. Hille, R. Hohler, and H. Strack, "A linearization and compensation method for integrated sensors," *Sens. Actuators A, Phys.*, vol. 44, no. 2, pp. 95–102, Aug. 1994.

[24] X. Li and G. C. Meijer, "An accurate interface for capacitive sensors," *IEEE Trans. Instrum. Meas.*, vol. 51, no. 5, pp. 935–939, Oct. 2002.

[25] M. Yamada and K. Watanabe, "A capacitive pressure sensor interface using oversampling $\Delta - \Sigma$ demodulation techniques," *IEEE Trans. Instrum. Meas.*, vol. 46, no. 1, pp. 3–7, Feb. 1997.

[26] K. F. Lyahou, G. van der Horn, and J. H. Huijsing, "Noniterative polynomial 2-D calibration method implemented in a microcontroller," *IEEE Trans. Instrum. Meas.*, vol. 46, no. 4, pp. 752–757, Aug. 1997.

[27] J. Rivera, G. Herrera, M. Chacón, P. Acosta, and M. Carrillo, "Improved progressive polynomial algorithm for self-adjustment and optimal response in intelligent sensors," *Sensors*, vol. 8, no. 11, pp. 7410–7427, 2008.

[28] S. Haykin, *Neural Networks,* 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.

[29] J. M. Dias Pereira, P. M. B. Silva Girao, and O. Postolache, "Fitting transducer characteristics to measured data," *IEEE Instrum. Meas. Mag.*, vol. 4, no. 4, pp. 26–39, Dec. 2001.

[30] J. M. Dias Pereira, O. Postolache, and P. M. B. Silva Girao, "A temperature-compensated system for magnetic field measurements based on artificial neural networks," *IEEE Trans. Instrum. Meas.*, vol. 47, no. 2, pp. 494–498, Apr. 1998.

[31] P. Arpia, P. Daponte, D. Grimaldi, and L. Michaeli, "ANN-based error reduction for experimentally modeled sensors," *IEEE Trans. Instrum. Meas.*, vol. 51, no. 1, pp. 23–30, Feb. 2002.

[32] A. P. Singh, S. Kumar, and T. S. Kamal, "Virtual compensator for correcting the disturbing variable effect in transducers," *Sens. Actuators A, Phys.*, vol. 116, no. 1, pp. 1–9, Oct. 2004.

[33] P. Daponte and D. Grimaldi, "Artificial neural networks in measurements," *Measurement*, vol. 23, no. 2, pp. 93–115, Mar. 1998.

[34] A. Marconato, M. Hu, C. Marzadro, A. Boni, and D. Petri, "A resource-constrained sensor dynamic compensation using a learning-from-examples approach," in *Proc. IEEE Instrum. Meas. Technol. Conf.*, Warsaw, Poland, May 2007, pp. 1–6.

[35] A. Marconato, M. Hu, A. Boni, and D. Petri, "Dynamic compensation of nonlinear sensors by a learning-from-examples approach," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 8, pp. 1689–1694, Aug. 2008.

[36] J. Rivera, M. Carrillo, M. Chacón, G. Herrera, and G. Bojorquez, "Self-calibration and optimal response in intelligent sensors design based on artificial neural networks," *Sensors*, vol. 7, no. 8, pp. 1509–1529, 2007.

[37] S. Mekid, "Further structural intelligence for sensors cluster technology in manufacturing," *Sensors*, vol. 6, no. 6, pp. 557–577, 2006.

[38] J. C. Patra and G. Panda, "ANN-based intelligent pressure sensor in noisy environment," *Measurement*, vol. 23, no. 4, pp. 229–238, Jun. 1998.

[39] J. C. Patra and A. van den Bos, "Modeling and development of an ANN-based smart pressure sensor in a dynamic environment," *Measurement*, vol. 26, no. 4, pp. 249–262, Dec. 1999.

[40] J. C. Patra, A. C. Kot, and G. Panda, "An intelligent pressure sensor using neural networks," *IEEE Trans. Instrum. Meas.*, vol. 49, no. 4, pp. 829–834, Aug. 2004.

[41] J. C. Patra, G. Chakraborty, and P. K. Meher, "Neural-network-based robust linearization and compensation technique for sensors under nonlinear environmental influences," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 5, pp. 1316–1327, Jun. 2008.

[42] J. C. Patra, C. Bornand, and P. K. Meher, "Laguerre neural network-based smart sensors for wireless sensor networks," in *Proc. IEEE Int. IMTC*, Singapore, May 2009, pp. 832–837.

[43] M. Yamada, T. Takebayashi, S. Notoyama, and K. Watanabe, "A switched-capacitor interface for capacitive pressure sensors," *IEEE Trans. Instrum. Meas.*, vol. 41, no. 1, pp. 81–86, Feb. 1992.

[44] Y.-H. Pao, *Adaptive Pattern Recognition and Neural Networks*. Reading, MA: Addison-Wesley, 1989.

[45] Y.-H. Pao and S. M. Philips, "The functional link net and learning optimal control," *Neurocomputing*, vol. 9, no. 2, pp. 149–164, Oct. 1995.

[46] H. Zhang, Z. Chen, Y. Wang, M. Li, and T. Qin, "Adaptive predictive control algorithm based on Laguerre functional model," *Int. J. Adapt. Control Signal Process.*, vol. 20, no. 2, pp. 53–76, Mar. 2006.

[47] L. Wang, "Discrete model predictive controller design using Laguerre functions," *J. Process Control*, vol. 14, no. 2, pp. 131–142, Mar. 2004.

[48] R. E. Attar, *Special Functions and Orthogonal Polynomials*. Morrisville, NC: Lulu Press, 2006.

[49] J. C. Patra, R. N. Pal, B. N. Chatterji, and G. Panda, "Identification of nonlinear dynamic systems using functional link artificial neural networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 2, pp. 254–262, Apr. 1999.

Jagdish Chandra Patra (M'97) received Ph.D. degree from the Indian Institute of Technology, Kharagpur, India, in 1996.

In 1999, he joined the School of Electrical and Electronics Engineering, Nanyang Technological University (NTU), Singapore, as a Research Fellow. He is currently an Assistant Professor with the School of Computer Engineering, NTU. His research interests include intelligent information processing using neural networks in the areas of information security and sensor networks.

Pramod Kumar Meher (SM'03) received the Ph.D. degree from Sambalpur University, Sambalpur, India, in 1996.

He is currently a Senior Scientist with the Institute for Infocomm Research, Singapore. His research interest includes the design of dedicated and reconfigurable architectures for computation-intensive algorithms pertaining to signal processing, image processing, and intelligent computing.

Dr. Meher is a Fellow of the Institution of Electronics and Telecommunication Engineers, India, and the Institution of Engineering and Technology, U.K.

Goutam Chakraborty (SM'05) received the Ph.D. degree from Tohoku University, Sendai, Japan, in 1993.

He is currently a Professor and the Head of the Intelligent Informatics Laboratory, Department of Software and Information Science, Iwate Prefectural University, Iwate, Japan. His main research interests are soft-computing algorithms and their applications to solve pattern recognition, prediction, scheduling, and optimization problems, including the applications in wired and wireless networking problems.