A Novel Competitive Learning Neural Network Based Acoustic Transmission System for Oil-Well Monitoring

Marcelo Godoy Simões, Senior Member, IEEE, Celso Massatoshi Furukawa, Alexander T. Mafra, and Julio Cezar Adamowski, Member, IEEE

Abstract—The optimal operation of an oil well requires the periodic measurement of temperature and pressure at the downhole. In this paper, acoustic waves are used to transmit data to the surface through the pipeline column of the well, making up a wireless transmission system. Binary data is transmitted in two frequencies, using frequency-shift keying modulation. Such transmission faces problems with noise, attenuation, and, at pipeline joints, multiple reflections and nonlinear distortion. Hence, conventional demodulation techniques do not work well in this case. The neural network presented here classifies signals received by the receiver to estimate transmitted data, using a linear-vector-quantization-based network, with the help of a preprocessing procedure that transforms time-domain incoming signals in three-dimensional images. The results have been successfully verified. The neural network estimation principles presented in this paper can be easily applied to other patterns and time-domain recognition applications.

Index Terms—Acoustic data transmission, neural network, oil pipeline.

I. INTRODUCTION

A SIMPLIFIED diagram of an oil extraction column used in deep-sea exploration is shown in Fig. 1. Oil flows up through the oil pipeline, i.e., the inner tube, which is made up of tube segments connected by thread joints. The figure also shows an external tube, called a jacket, which surrounds the pipeline. The space between the pipeline and the jacket is filled with water. A column length used in deep-sea exploration might ordinarily be longer than 3000 m. Two different types of equipment are used to monitor the temperature and pressure at the oil-well downhole.

1) One is an autonomous data logger, which is used from the beginning of the well operation, to store data from the initial conditions up to a stabilized situation. Since the logger cannot be used online in the measurement, it is impossible to know the time that the well enters the steady state, leading to overestimation of the acquisition time.

Therefore, the data logger remains in operation longer than necessary, delaying the beginning of full operation of the well.

2) Another conception of monitoring system replaces the data logger for the whole operating life of the well; it is called a permanent downhole sensor (PDS). The operator measures the temperature and pressure every day and injects water into the system in order to keep the well operating under optimum conditions. However, temperature and pressure sensors are usually connected to the surface with cables, which makes the system prone to damage and expensive to install and maintain.

This work describes a battery-powered wireless transmission system for this application. With the elimination of cabling, significant cost savings and increased reliability can be attained. In addition, a wireless downhole sensor could be used to monitor the well at the beginning of operation, replacing the data logger.

To implement wireless transmission along an oil-well column, underground sensors must be placed at a depth of several meters. Radio waves are not appropriate in this case, since they would have to propagate in the water between the pipeline and the external jacket with consequent large attenuation. On the other hand, acoustic waves constitute an alternative for transmitting data in this situation. Acoustic modems that transmit data in water have been successfully developed for submarine applications and ocean oil platforms [1]. Acoustic transmission through the walls of a drilling tube has been reported where data were received from tilt sensors...
that measured the tilt angle of the drill tip [2], [3]. In the case of the downhole sensor, data would be traveling along the oil pipeline during full operation of the well.

A simple modulation technique used to transmit binary data is frequency-shift keying (FSK), that employs two different frequencies, \(f_1\) and \(f_2\). A burst signal at frequency \(f_1\) is emitted when a bit “1” has to be transmitted, while frequency \(f_2\) is used for transmitting a bit “0.” Although the encoding principle is simple, the correct detection of ones and zeros at the receiver side of a downhole measuring system is quite difficult. The acoustic signal is strongly distorted and attenuated after traveling along the oil pipeline. Multiple reflections take place at the pipeline junctions, and propagation through thread joints presents nonlinear characteristics, resulting in a very complex acoustic signal both in time and frequency domains. The oil flow also produces vibration, which, in its turn, constitutes a source of acoustic noise that influences the arriving signal. Such characteristics strongly deteriorate the signal–noise ratio.

The usual correlation techniques for demodulation are not appropriate for this case, since their utilization would require increased transmission power to reduce the high error rate of the communication channel, consequently reducing the autonomy of the downhole sensor. Such nature motivated the development of a neural network, to automatically cope with the signal transmission and overcome its nonlinear nature.

Two fundamental aspects must be considered critical in the design of this system: the piezoelectric acoustic transducer, responsible for generating the acoustic waves, and the understanding of the wave propagation through the steel pipeline elements. In addition, operating temperature and pressure (100 °C/300 atm), size (within a cylinder of 30-mm diameter and maximum length of 8 m), and noninterrupted operation of up to five years are challenging requirements. Much care must be taken for power consumption optimization, by designing the acoustic transducers and managing the sleeping state (idling) of electronic circuits. A wakeup signal sent from the top of the pipeline commands the system to get out of idling.

**II. PIPELINE ACOUSTIC TRANSMISSION**

Fig. 2 illustrates the oil pipeline, actually the inner tube inside the external jacket where the oil flows. The pipeline is assembled from steel tubes, with length \(d_1\) and cross-sectional area \(a_1\), connected by threaded tool joints, with length \(d_2\) and cross-sectional area \(a_2\). The acoustic wave propagating on such mechanical structure has phase and group velocities depending on the frequency; some frequencies are blocked for propagation within certain periodic bands. Therefore, the acoustic wave propagates at the expense of high distortion due to the variation of phase and group velocities in terms of frequency. The relationship between the angular frequency \(\omega\) and the wave number \(k\) is given by [3]

\[
\begin{align*}
\cos k(d_1 + d_2) &= \cos \left( \frac{\omega d_1}{c_2} \right) \cos \left( \frac{\omega d_2}{c_2} \right) - \frac{1}{2} \left( \frac{a_1}{c_2} + \frac{a_2}{c_2} \right) \\
&\cdot \sin \left( \frac{\omega d_1}{c_2} \right) \sin \left( \frac{\omega d_2}{c_2} \right)
\end{align*}
\]

where \(c_1 = c_2\) is the extensional wave velocity in steel, \(c_f = \omega/k\) is the phase velocity, and \(c_g = \frac{dc_f}{dk}\) is the group velocity. Fig. 3 shows the behavior of the phase velocity for an oil pipeline, composed of pipes of 9-m length. The acoustic waves can propagate within the frequency range indicated in Fig. 3 where the phase velocity changes from imaginary (represented by the value zero) to infinity, being blocked after the phase comes back to imaginary again. The figure indicates several bandpass ranges, the fourth range has been analyzed, showing a 180-Hz width. The pipe length might vary up to 0.5 m, leading to shifting bandpass ranges around 50 Hz. For longer lengths, the central frequency range is smaller and the bandwidth has such nature along all the frequency range. In order to choose the FSK operating frequencies, a tradeoff study of transducer power and attenuation has been made, and the frequencies 4375 and 4425 Hz have been selected.

**III. COMPETITIVE LEARNING FUNDAMENTALS**

Neural networks are normally used in applications where it is required to learn hidden patterns and store them in an associative memory. If the neural network has self-organizing capabilities, it can modify the connection strengths based only on input characteristics; in this case, the storage is autoassociative. The Kohonen network is a self-organizing system, with a single layer, having highly interconnected neurons. There is an inverse
distance function that provides a feedback signal so as to render excitatory connections to the neurons in the vicinity, as well as inhibitory connections to the neurons further away [4], [5]. The combination of a Kohonen network and a Grossberg outstar makes up the counterpropagation network (CPN) shown in Fig. 4. Such a blending creates a powerful network that can function as an adaptive lookup table in pattern recognition, pattern completion, and signal enhancement. It contains a supervised learning process by virtue of the association of input vectors with the corresponding output vectors [6]. Even though it is as robust as a regular backpropagation neural network, it has rapid training and saved computational time, via the construction of a statistical model of the input vector environment.

Given \( x_i \) and \( y_j \) as correspondent input and output vectors, a training data base is constituted by the set of patterns \((x_1,y_1),(x_2,y_2),\ldots,(x_L,y_L)\). The CPN learns to associate the input layer vector \( x \) with the output layer vector \( y \). If the relationship between them could be described by continuous function \( \Phi \), as \( y = \Phi(x) \), the CPN could learn the mapping of any \( x \) within the range of the training input and also the inverse mapping. When an input vector is applied to layer \#1, each neuron on layer \#2 calculates the net input value, and there is a competition among the neurons to determine which one will send an excitation to the output neurons.

Four modules are working in this neural network: the input layer—which processes input data, the instar layer, the inner competitive network, and the outstar layer—the output structure. The input layer might contain buffers responsible for delivering data to the neurons in the following layer; it can also preprocess incoming data, as well as performing the necessary input data scaling. Fig. 5 shows the usual way of representing a processing element; on the other hand, if such a neuron is seen as an instar element as in Fig. 5(b), the output is no longer different from the inputs.

Therefore, (2) determines the input pattern of the total intensity for each neuron. To every input pattern is assigned a vector \((\theta_1,\theta_2,\ldots,\theta_n)^T\) in (3), called the normalized reflected pattern

\[
I = i \times w = \sum_{j=1}^{n} i_j w_j \tag{2}
\]

where \( w \) is the weight vector and \( i \) is the input vector.

The hidden layer in the CPN is constructed out of instar processing elements. The net value is calculated by the inner product \( i \times w \), assuming that input data has been prescaled to unity length. Therefore, both \( i \) and \( w \) could be represented in a multidimensional unit of space vector, as in Fig. 6. The quantity \((i - w)\) is a vector that points the weight vector toward the input, and the instar can learn such a pattern by rotating the weight vector until it becomes aligned to the input pattern, as indicated by

\[
\theta_i = I_i \left( \sum_{j=1}^{n} I_j \right)^{-1} \tag{3}
\]

Although this simple learning algorithm assumes the instar layer as a “winner-takes-all” competition, there are other training methods termed LVQ2, LVQ2.1, and LVQ3, where the network structure remains intact [5], but the training is no longer restricted to the winning neuron alone. In the final implementation of this work, only the neuron closest to the pattern sends a single +1 output to the outstar layer, shown in Fig. 4. The outstar neurons are then trained by either delta or Hebbian rule [6].

This CPN is done in two steps. First, the Kohonen middle layer is trained until it recognizes the input patterns and categorizes them into an input feature map. It selects the input vector pattern randomly and then calculates \( \alpha(i - w!) \) to update the weight vector in accordance with (4). After presentation of a
pattern in the input layer, the units in the hidden layer sum their inputs and then compete among them to respond to the input pattern. The unity with the highest net input wins and its activation is set to 1 while all others are set to 0; the process is repeated to cover all input space vectors and after competition the output layer sums weighted outputs of the hidden layer. Once the middle layer is adequately trained, the weights between the input layer and the middle layer are frozen and the outstar is trained with supervised Hebbian training [6], [7], i.e., by increasing the learning constants until it can reproduce the outputs. When the network is fully trained, the presentation of a pattern in \( \mathbf{x} \) gives trained \( y_j \). A commercial neural network simulator—NeuralWorks Professional—was used to initially develop the topology and strategies before writing the final code. Issues like initialization of the neural network, pruning, generalization, and tests are hard to do without a graphical simulator.

The assumption that all input vectors are of constant norm 1 is achieved by inserting a projection layer, which projects the input vectors to a hypersphere of radius \( R \) in a space one dimension higher than the input space; then, the Euclidean distance between a training vector and a cluster center is given by

\[
|\mathbf{c}_k - \mathbf{x}|^2 = 2(R^2 - \mathbf{c}_k \cdot \mathbf{x})
\]

where \( \mathbf{x} \) is the input vector, \( \mathbf{c}_k \) is the cluster center, and \( R \) is the hypersphere radius.

In the context of NeuralWorks Professional, a unidirectional CPN can be implemented like a radial basis function (RBF) network through the setting of the hidden layer for competitive learning with \( P \) nearest neighbors heuristic, i.e., a given cluster center \( \mathbf{c}_k \), let \( k_1, \ldots, k_P \) indexes the \( P \) nearest neighboring cluster centers. The width of a Gaussian transfer function \( \sigma_k \) is then set to the root-mean-square distance of a given cluster center to the \( P \) nearest neighboring cluster centers

\[
\sigma_k = \sqrt{\frac{1}{P} \sum_{j=1}^{P} ||\mathbf{c}_k - \mathbf{c}_j||^2}.
\]

After developing the neural network strategies, a final customized algorithm was implemented in C++ language, so as to run in the microcomputer embedded acoustic communication reception system.

IV. INPUT SIGNAL PREPROCESSING FOR CPN ESTIMATION

In order to improve the training convergence of the CPN network and the correspondent classification performance, the raw acoustic signal was preprocessed before being fed to the network inputs. A time window was programmed to find significant ultrasound activity, i.e., the transmitted burst was sampled and acquired. The discrete vector of \( \eta_a \) samples, \( \mathbf{f} \), was transformed into a three-dimensional image with \( m \times m \times m \) pixels, where \( m \) is the number of discrete levels used to sample the signal. Two new vectors, \( \mathbf{u} \) and \( \mathbf{v} \), were created from \( \mathbf{f} \) as indicated by

\[
\mathbf{u}[k] = \mathbf{f}[k],
\]

\[
\mathbf{v}[k] = \mathbf{f}[k + l], \quad \text{for } k = 0, 1, \ldots, \eta_a.
\]

The ordered pairs \( (\mathbf{u}[k], \mathbf{v}[k]) \) defined \( \eta_a \) coordinates of the image domain \( (\eta_x, \eta_y) \). A histogram matrix \( \mathbf{H}(\eta_x, \eta_y) \) was constructed using these coordinates, by counting how many times each pixel was addressed by \( \mathbf{u} \) and \( \mathbf{v} \). Such histograms were then used as the images to be classified initially. However, it was found that they showed pronounced peaks around the origin, masking out other features of the image. Therefore, a geometric series generator was used to compress the histogram peaks and reinforce other points of the image [8]. The geometric series has two main advantages over other compression functions (like the log function \( \ln \)): 1) it is less computationally expensive to calculate and 2) it offers control on the asymptotic behavior. The compressor \( \mathbf{Z} \) applied to the histogram \( \mathbf{H}(\eta_x, \eta_y) \) is given by

\[
\mathbf{Z}(\eta_x, \eta_y) = \frac{1 - 0.5 \mathbf{H}(\eta_x, \eta_y)}{1 - 0.5}.
\]

Equation (9) makes up a geometric series with initial value 1 and ratio 0.5. The series convergence is 2, i.e., for large values of \( \mathbf{H}(\eta_x, \eta_y) \).

A. Application to Distorted Waveform Estimation

Before applying this technique to the present work, some tests were performed in the study of ac line harmonics estimation. The standards adopted by IEC-555 and IEEE-519 strictly limit the total harmonic distortion and maximum individual harmonics injected by the power consumers. Harmonic currents may cause distortion of the voltage waveform via the power system series impedance, exciting resonances in some distance far from their source, while odd triple harmonics lead to large neutral currents in three-phase systems. The technique presented here of: 1) using the compression function \( \mathbf{Z}(\eta_x, \eta_y) \) and 2) employment of a CPN was used to detect the thresholds standardized by IEEE-519. Fig. 7(a)–(c) shows the mapping of a 60-Hz sinusoidal waveform and the correspondent third and fifth harmonics pollution. It is easily seen that the surface generated by \( \mathbf{Z}(\eta_x, \eta_y) \) encoded the signal dynamics. A description of such investigation is not within the scope of this paper, but it was very important for debugging the proposed methodology; it successfully confirmed the proposed ideas of this paper.

V. EXPERIMENTAL SETUP

A transducer system with all of the required electronics was installed in the laboratory. This system was used to study the transmission and also to settle down the hardware and software before installation in the oil extraction system. An oil pipeline of approximately 100 m in length, made of 9.5-m-long steel tube segments which were connected in a chain by means of thread joints, was used in the laboratory. The diameter of the tubes was 4 in and the wall thickness was 0.5 in. The whole structure rested horizontally on the ground.

Acoustic signals were generated by a piezoelectric transducer with output power of 150 W, connected to one end of the pipeline through a coupler, as shown in Fig. 8. On the other end, an accelerometer was responsible for signal reception and delivery to an analog–digital converter interfaced to a personal computer. The transducer emitted one bit as a burst of ten cycles using 4375 kHz for a “zero” and 4425 kHz for a “one.” Bits were separated by 500 ms of gaps to avoid intersymbolic
interference. The transducer was installed 200 m down the pipeline in a prototype installation.

Fig. 9 shows the frequency response of the transmission channel in such conditions. Fig. 10 shows a signal received by the accelerometer. The amplitude samples had a resolution of 8 bits, i.e., in a binary range of from $-128$ to $+128$. Multiple reflections due to the pipeline joints are clearly observed in Fig. 10. The signal envelop presents several local maxima and the time span of the signal is quite long when compared to the short excitation (ten cycles) imposed by the emitter. It has been observed that those interactions at the joints added nonlinear distortion, leading to difficulties in using standard correlator detectors for FSK estimation. The three-dimensional figure shown in Fig. 11 is an example of the images obtained with the described algorithm for the acoustic signal presented in Fig. 10. The accelerometer signal was sampled with 5 bits (i.e., $m = 32$). The surface in Fig. 11 was padded with a contour of zeros to permit a better view of the inner features.

A. Algorithm Implementation

The network was implemented in C++. Such decision was due the flexibility of operating system implementation and to the function overloading possibilities, i.e., allowing functions that perform similar tasks operating with different data types of objects; in addition, C++ provides encapsulation, inheritance, and dynamic run-time binding, allowing reusable code [9]. A flow chart for the network program is shown in Fig. 12, where two learning phases, namely, the Kohonen layer and the Grossberg outstar, are described. The network weights were initialized with the average pattern of the whole data set. It was observed that this procedure helped the training convergence. The training was controlled by the learning parameter ($\alpha$) which was decreased as the training went on. Convergence was attained when the system classified the stimulus pattern correctly, or eventually stopped by a prescribed maximum number of training steps. In turn, the program saved the weights used for recalling. Therefore, the network received a preprocessed pattern, classifying it into a stored pattern that resembled the input.

The neural network data structure is given in Fig. 13. CPN is the main variable, which is made up of three lists linked dynamically: CPN[input], CPN[hidden], and CPN[output], each being a record with the following elements: $A_k$ (neuron activation) and $\omega_{ij}$ (connection weights from $i$ elements and $j$ elements on the preceding layer). The software modules determine memory allocation, image width and height, scaling of images, and the class routines for classification and training control. The network was trained using a set of 40-bit transmission experiments, each with 8192 sampled values. Frequencies indicating
“ones” and “zeros” were transmitted and the signals received by the accelerometer were recorded. Although the transducer peak power was in the range of 4.4–4.5 kHz, it was excited from 1000 Hz, every 100 Hz, up to 5000 Hz, making a rich and broad response of the transducer for the training set. The CPN network was trained with only 70% of the acquired signals. In turn, the remaining 30% of data was deliberated corrupted with 40% of Gaussian noise, yet the network was still able to classify frequency 4.375 kHz (“zero”) correctly in 99.17% and frequency 4.425 kHz (“one”) in 97.50% of the cases. It was also observed that such estimation approach is immune to phase shift on the input signal, because the image mapping equations, given by (7)–(9), eliminate such time relationship. Fig. 14 shows a screen image of network training. The training time was on the order of several minutes, but the recalling time was very fast, just a few tenths of seconds for the PC processing.

A CPN-based estimation system has a huge amount of input neurons, leading to a possible overfitting of the input/output mapping. Training to an excessively low error tolerance can result in overall poor performance by the network. In effect, the network begins memorizing the training set and might lose its ability to generalize. The conventional test of convergence requires that the overall error over the ensemble of training vectors be less than some specified minimum value.

By sequentially training and testing with an independent data test set, the error can be compared to the learning phase total error and the generalization loss GL as (10) indicates a relation between the current error (E_{cu}) and the smallest error observed during a batch test distinct from the training set (E_{bt}). The generalization loss can then be used heuristically to determine whether there are too many hidden units and the hidden layer size should be decreased, or the training set does not adequately represent the decision class and should be augmented.

\[
GL = 100 \cdot \left( \frac{E_{cu}}{E_{bt}} - 1 \right) \%.
\]
It is possible for some promising methods for generalization optimization to be further addressed, like noise injection and weight decaying [10], or like statistics tests to estimate whether a certain weight will become zero during a future training step [11]. Those suggestions are being taken into consideration for future improvements in this estimation algorithm.

VI. CONCLUSION

A CPN neural network algorithm was designed for the estimation of acoustic signal transmission in an oil-pipeline extraction system, where the well pressure and temperature were sensed and sent to the top of the line by the ultrasonic signals that performed a FSK-type of digital modulation, with two frequencies within the ultrasonic transducer range. The oil pipeline has junctures around every 10 m, which cause multirefections and nonlinear distortion, resulting in a complex acoustic signal in the time and frequency domains. The usual correlation techniques are not efficient enough for demodulation, and one needs several field measurements for setting up the channel communication model. A CPN neural network was used to classify the acoustic signals into two classes of binary transmission, overcoming such nonlinearities and model settings. A preprocessing method of raw input acoustic signal was proposed with a chaotic discrete mapping, so as to build a three-dimensional image of the incoming signal, relating the surface to the time signal.

The system was tested in a laboratory with a 100-m-long oil-pipeline. Acoustic signals were generated by a 150-W piezoelectric transducer and an accelerometer was responsible for signal reception and delivery to an analog–digital converter interfaced to a PC. The neural network algorithm was written in C++ and ran in the PC for training and estimation. The input data signals were deliberately corrupted with 40% of noise and the network was still able to correctly estimate original data with 2.5% of average correctness, demonstrating its robustness and stability. This present system opens new horizons for oil-well PDS monitoring systems. Periodically, the PDS can send known patterns for the equipment on the top of the pipeline, which will train the CPN-based estimation algorithm, making the system parameter insensitive and more reliable in regard to the natural environmental degradation. It is important to observe that, in an acoustic transmission system with a neural network capability like this, the channel communication model is not necessary. The CPN-based system can be easily translated to any oil well as well, and even retrofitted to the existing ones. The system is being fitted into a Brazilian sea oil extraction system and can be also used in other pattern recognition applications.

REFERENCES

Marcelo Godoy Simões (S’89–M’95–SM’98) received the B.S. and M.S. degrees from the University of São Paulo, São Paulo, Brazil, the Ph.D. degree from the University of Tennessee, Knoxville, and the “Livre-Docência” degree (D.Sc.) from the University of São Paulo, in 1985, 1990, 1995, and 1998, respectively.

For the last ten years, he has been a Professor of Engineering at the Polytechnic School, University of São Paulo (EPUSP), involved in organizing undergraduate courses in the automation and systems area, teaching graduate courses in power electronics and drives, and also conducting research on applications of intelligent systems. He was a Visiting Associate Professor at Georgia Institute of Technology, Atlanta, from January to March 1999, teaching graduate courses and conducting research. He has been working on research concerning fuzzy logic and neural networks applications to power electronics, drives, and machines control. He has authored numerous papers published in international journals and conference proceedings. His interests include research and development of intelligent applications (neural network, fuzzy logic, and genetic algorithms) for renewable energy systems, diagnostics, sensor fusion techniques, robotics, rapid prototyping, manufacturing systems, and industrial applications. In addition to his academic career, he has initiated several intelligent-based control system industrial applications in Brazil, including applications in the oil extraction industry, photovoltaic applications, and electric vehicles and drive systems applications. He organized the SBAI—Brazilian Intelligent Automation Seminar, sponsored by IFAC and IFSA, held in September 1999.

Dr. Simões organized a session on “Industrial Applications of Intelligent Systems” at the IEEE 2nd International Conference on Knowledge-Based Intelligent Electronic Systems, organized by IEEE South Australia in 1998 in Adelaide, Australia. He is also involved with the IEEE Industry Applications Society (IAS), leading technical activities in the IAS São Paulo Chapter. He organized the 1998 Brazilian IEEE-IAS INDUSCON Meeting in São Paulo, Brazil. He is a Registered Professional Engineer in the State of São Paulo, Brazil.

Celso Massatoshi Furukawa received the Engineering degree and the Masters degree in electronics from the University of São Paulo, São Paulo, Brazil, in 1987 and 1992, respectively, and the Doctoral degree from The University of Tokyo, Tokyo, Japan, in 1996.

Since 1989, he has been with the Polytechnic School, University of São Paulo, where he is currently an Assistant Professor, engaged in teaching the undergraduate course on mechatronics in the Department of Mechanical Engineering. His research interests include industrial and medical applications of ultrasound, industrial automation, electronic sensors, and solar vehicles.

Prof. Furukawa is a member of the American Society of Mechanical Engineers.

Alexandre T. Mafra received the B.E. degree from the University of São Paulo, São Paulo, Brazil, where he is currently working toward the Master degree.

During his undergraduate studies, he was engaged in research on neural networks applications to pattern recognition, such as fingerprint recognition and digital filtering for FSK signals with high noise levels. For the last two years, he has been an IT Consultant, developing factory floor monitoring and control real-time integrated systems and warehouse management systems at Uniconsult Systems and Services, São Paulo, Brazil. In addition, for eight years, he has managed his own high school (Exatus Colegio e Vestibulares), where he teaches physics. His interests include research on neural network applications for pattern recognition in dynamic signals.

Julio Cezar Adamowski (M’95) was born in Paraná, Brazil, in 1954. He received the Aeronautical Mechanical Engineer degree from the Instituto Tecnológico de Aeronáutica, São José dos Campos, Brazil, the Masters degree in precision machinery engineering from the Faculty of Engineering, The University of Tokyo, Tokyo, Japan, and the Doctor degree from University of São Paulo, São Paulo, Brazil, in 1980, 1985, and 1993, respectively.

He joined the Mechanical Engineering Department, Polytechnic School, University of São Paulo, in 1988 and became a Full Professor in 1998. He has worked on several research projects concerning applications of sound and ultrasound since 1988. His research interests include the modeling of elastic-wave propagation in solids, the ultrasonic characterization of liquids, and ultrasonic NDE.