

# Parametric CMAC Networks: Fundamentals and Applications of a Fast Convergence Neural Structure

Paulo E. M. Almeida, *Student Member, IEEE*, and Marcelo Godoy Simões, *Senior Member, IEEE*

**Abstract**—This paper shows fundamentals and applications of the parametric cerebellar model arithmetic computer (P-CMAC) network: a neural structure derived from the Albus CMAC algorithm and Takagi–Sugeno–Kang parametric fuzzy inference systems. It resembles the original CMAC proposed by Albus in the sense that it is a local network, (i.e., for a given input vector, only a few of the networks nodes—or neurons—will be active and will effectively contribute to the corresponding network output). The internal mapping structure is built in such a way that it implements, for each CMAC memory location, one linear parametric equation of the network input strengths. This mapping can be corresponded to a hidden layer in a multilayer perceptron (MLP) structure. The output of the active equations are then weighted and averaged to generate the actual outputs to the network. A practical comparison between the proposed network and other structures is, thus, accomplished. P-CMAC, MLP, and CMAC networks are applied to approximate a nonlinear function. Results show advantages of the proposed algorithm based on the computational efforts needed by each network to perform nonlinear function approximation. Also, P-CMAC is used to solve a practical problem at mobile telephony, approximating an RF mapping at a given region to help operational people while maintaining service quality.

**Index Terms**—Cerebellar model arithmetic computers (CMACs), communication systems, mobile communication, neural networks.

## I. INTRODUCTION

SINCE the introduction of cerebellar model arithmetic computer (CMAC) networks by Albus [1], [2], these biologically inspired models have been widely used. Most of the time, they have worked as function approximators to support an adaptive approach. From the first applications in robotic manipulators [9], [14], to more complex approaches in the fields of industrial processes [15], robotics [18], and missile control [8], some modifications and adaptations have been proposed to improve the behavior and the performance of the original CMAC model.

Paper MSDAD-A 03–08, presented at the 2002 Industry Applications Society Annual Meeting, Pittsburgh, PA, October 13–18, and approved for publication in the IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS by the Industrial Automation and Control Committee of the IEEE Industry Applications Society. Manuscript submitted for review October 15, 2002 and released for publication June 13, 2003. This work was supported in part by CAPES under Project BEX 0384/00-3, by FINEP under the RECOPE-IA Project, by FAPEMIG, by the Colorado School of Mines, and by the National Science Foundation under Grant ECS 0134130.

P. E. M. Almeida is with the Centro Federal de Educação Tecnológica de Minas Gerais, 30.510-000 Belo Horizonte, Brazil (e-mail: paulo@dppg.cefetmg.br).

M. G. Simões is with the Division of Engineering, Colorado School of Mines, Golden, CO 80401-1887 USA (e-mail: m.g.simo@ieee.org).

Digital Object Identifier 10.1109/TIA.2003.816543

New learning algorithms, modified structures, and replacement of the original binary activation functions have been the most popular proposed improvements [6], [7], [11].

This work presents a new neural network structure, mainly derived from CMAC algorithm and Takagi–Sugeno–Kang parametric fuzzy inference systems (fuzzy TSK) [17], [19]. The universal approximation properties exhibited by these algorithms (see [12] and [22]) make them very suitable for employment in conjunction with neural networks. The so-called parametric CMAC network (P-CMAC) resembles the original CMAC proposed by Albus, in the sense that it is a local network (i.e., for a given input vector, only a few of the networks nodes—or neurons—will be active and will effectively contribute to the corresponding network output). In addition, the internal mapping structure is built in such a way that it implements, for each CMAC memory location, one linear parametric equation of the network input strengths. This mapping can be corresponded to a hidden layer in a multilayer perceptron (MLP) structure (for more details about neural networks algorithms and their applications, the reader is referred to [10] and [13]). The outputs of the active equations are then weighted and averaged to generate actual outputs [4].

Implementation details of P-CMAC networks are discussed, along with practical results that show potential advantages of this algorithm when compared to classical neural structures. A practical application of P-CMAC to mobile telephony monitoring services is described in detail and the corresponding results are shown. Here, the proposed network proved to be quite powerful, giving important feedback information to the operational personal at the telephony company.

This paper is organized into six sections. The fundamentals of P-CMAC network are detailed in Section II, where they are also compared to original and modified CMAC networks. Section III addresses training algorithms for P-CMAC. Section IV shows results obtained with the new structure. Section IV also compares P-CMAC performance and computational requirements with those of original CMAC and MLP structures and suggests other applications of the proposed algorithm. Section V discusses the application of P-CMAC to mobile telephony monitoring and shows practical results, and Section VI concludes the paper.

## II. BACKGROUND AND P-CMAC BASICS

A CMAC neural network, as described by Albus, can be depicted in three mappings, from its input to its output: an activa-

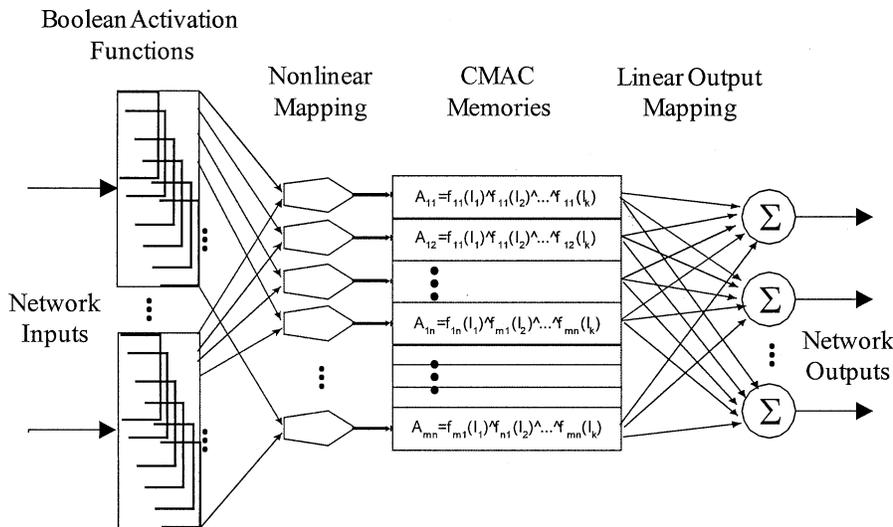


Fig. 1. Albus CMAC network basic structure.

tion mapping at the input space, which determines which nodes or neurons will remain active for the current input vector and at which strength they will be processed; an internal mapping, which generates the outputs for the CMAC memories, based on the activation of the input neurons and on the parameters of the linear equations; and, a linear output mapping, which performs a weighted sum of the outputs of the active CMAC memories and generates the overall outputs to the network.

The input mapping can be seen as a set of multidimensional interlaced receptive fields, each one with finite and sharp borders. Any input vector to the network excites some of these fields, while the majority of the receptive fields remain unexcited (not contributing to the corresponding output). Additionally, the weighted average of the excited receptive fields will form the network output. Fig. 1 shows a schematic diagram of this structure. This figure depicts the nonlinear input mapping in the Albus approach, but omits a hashing operation that can be performed to decrease the amount of memory needed to implement the CMAC memories in the internal mapping.

CMAC networks are considered local algorithms because, for a given input vector, only a few receptive fields will be active and contribute to the corresponding network output. In the same way, the training algorithm for a CMAC network should affect only the weights corresponding to active fields, excluding the majority of inactive fields in the network. This increases the efficiency of the training process and minimizes the computational efforts needed to perform adaptation in the whole network. They present the following features.

- The input mapping can be chosen by the designer, through the definition of the superimposing pattern for the receptive fields.
- The network outputs are linearly dependent to the weight set, as opposed to MLP model where the input mapping transformed by the weight set is applied to a nonlinear output activation function.

The signal flow through a CMAC network is represented by (1)–(5). These equations show a generalization parameter  $C$ , which determines the number of active fields for each input vector  $S$ . The first step is to process the input vector, creating

a normalized version of it, called  $S'$ . In (2),  $q_i$  the normalized components of  $S'$  and  $\Delta_i$  are normalization factors for each receptive field of the network used in the normalization process

$$S = [s_1, s_2, \dots, s_n] \quad (1)$$

$$S' = [q_1, q_2, \dots, q_n] \\ = \left[ \text{int} \left( \frac{s_1}{\Delta_1} \right), \text{int} \left( \frac{s_2}{\Delta_2} \right), \dots, \text{int} \left( \frac{s_n}{\Delta_n} \right) \right]. \quad (2)$$

Second, the address vectors  $A$  are formed from the  $C$  receptive fields excited by the normalized input  $S'$ . The index  $i$  on (3) denotes the  $C$  superimposed activation functions that cover the input space  $\mathfrak{R}^n$

$$A_i = [a_{i1}, a_{i2}, \dots, a_{in}] \quad i = 1, 2, \dots, C \\ = [q_1 - \text{mod}((q_1 - i), C), q_2, \dots, q_n \\ - \text{mod}((q_n - i), C)]. \quad (3)$$

Each  $A_i$  vector is the address for an extremity of the hypercube defined by the receptive field excited in the layer  $i$ . As the total number of receptive fields in this  $n$ -dimensional hyperspace can be very large,  $A_i$  vectors can be considered virtual CMAC memories. In this case, a second input mapping based on memory hashing is accomplished, to relate the virtual memory to the physical memory available. This way,  $A'$  vector is obtained from the  $A$  vectors, as shown by (4)

$$A' = h(A, M). \quad (4)$$

In the above equation,  $h(\cdot)$  represents any memory hashing function able to map the  $a_{ij}$  components of  $A$ —which points to virtual addresses—into uniformly distributed actual addresses of the available physical memory of size  $M$ . Finally, the CMAC outputs  $y(A')$  are obtained by the weighted average of the active memories, as shown by (5)

$$y(A') = \frac{1}{C} \sum_{i=1}^C W \cdot A'_i. \quad (5)$$

The training algorithm proposed by Albus adjusts all the weights that have contributed to the actual output by the same

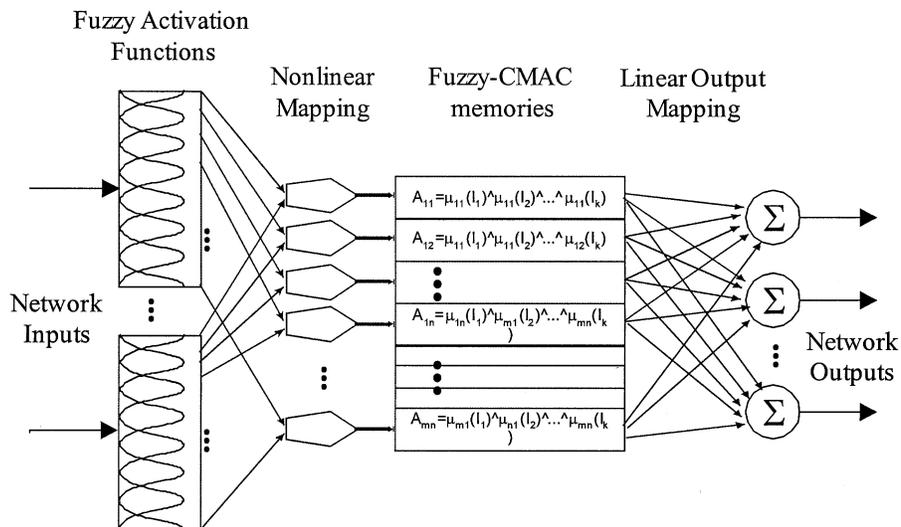


Fig. 2. Fuzzy-CMAC network basic structure.

adjusting factor calculated from the output error observed at this output. Therefore, the training equation can be written as

$$dW = \delta \cdot E \cdot A' = \delta \cdot (y - \hat{y}) \cdot A' \quad (6)$$

where  $\delta$  is a learning coefficient which varies within [0,1] and determines the correction gain,  $E$  is the output error, and  $A'$  is the physical memory activated by the corresponding input.

As opposed to the global training scheme verified on MLP networks when trained by backpropagation algorithm (where all weights from all layers are adjusted at each training step), at CMAC training, the weight corrections will occur only for the weights that effectively contributed to the output  $y$ . That indicates a more efficient algorithm and reduced computational costs.

A drawback of the original CMAC algorithm is the binary behavior of the activation functions that form the receptive fields to the network. In terms of the signal flow through the network, this means that a given actual memory position is either associated to a given input vector (and so effectively contributing to the corresponding output), or it is completely not associated to this input vector (not contributing at all to the output). In some cases, this kind of relationship can cause discontinuities at the network output, even when a smooth signal is continually presented to the inputs. To eliminate this undesired behavior and, thus, to reduce the corresponding output error at the network output, a very large number of activation fields must be used, which increases the algorithm complexity and can compromise its real-time execution.

CMAC input mapping can also be done by replacing the original binary functions by continuous or other differently shaped functions (see [7] and [11]). At their approaches, B-splines and Gaussian curves are used as activation functions to the receptive fields. When any kind of continuous function is used, CMAC input mapping resembles the activation mapping that occurs at a fuzzy system (for more details, see [3] and [16]), when a crisp input is fuzzified into a fuzzy partition in the input space.

In this case, the CMAC memories receive not a binary value, but the corresponding output of the activation functions with

respect to the input vector. Hence, the network outputs are now the average of the weighted summation of the output weights multiplied to the activation strengths expressed by the CMAC memories. Formally, the network output was shown by (7), as a function of its input vector  $S$

$$y(S) = \frac{1}{M} \sum_{i=1}^M w_i \cdot \mu_i(S). \quad (7)$$

In the above equation,  $w_i$  represents each output weight in the modified CMAC network and  $\mu_i(S)$  is the output of the  $M$  activation functions defined over the input space. Fig. 2 shows a schematic diagram of the modified CMAC network as described here. Some authors tend to call this architecture a “fuzzy-CMAC network,” because of the resemblance just discussed [8].

The training scheme used in fuzzy-CMAC networks is very similar to the method used in the original CMAC, except that now the weight corrections can take into account the actual strength of each CMAC memory that contributes to the network error. Thus, the learning equation is expressed by

$$dw_i = \delta \cdot E \cdot \mu_i(S) = \delta \cdot (y - \hat{y}) \cdot \mu_i(S). \quad (8)$$

In the case of P-CMAC, the activation mapping has been modified with respect to Albus’ CMAC algorithm. In our approach, fuzzy activation functions are employed. This allows each neuron to be activated with strengths different from the unity and then improves the undesired Boolean mapping that exists at the original algorithm. The result is a smoother output for smooth inputs, as opposed to what occurred before with CMAC networks.

The internal mapping at P-CMAC is an original improvement. Each bite-sized memory location existing in CMAC is replaced for one linear parametric equation of the input strengths. The idea of introducing parametric equations here came from fuzzy TSK models, which have such equations at their rule consequents. This mapping corresponds to a hidden layer in an MLP structure. Therefore, the  $k$ th neuron now implements (9), where  $P_{ij}$  are real parameters and  $I_i$  are the inputs of the network. It

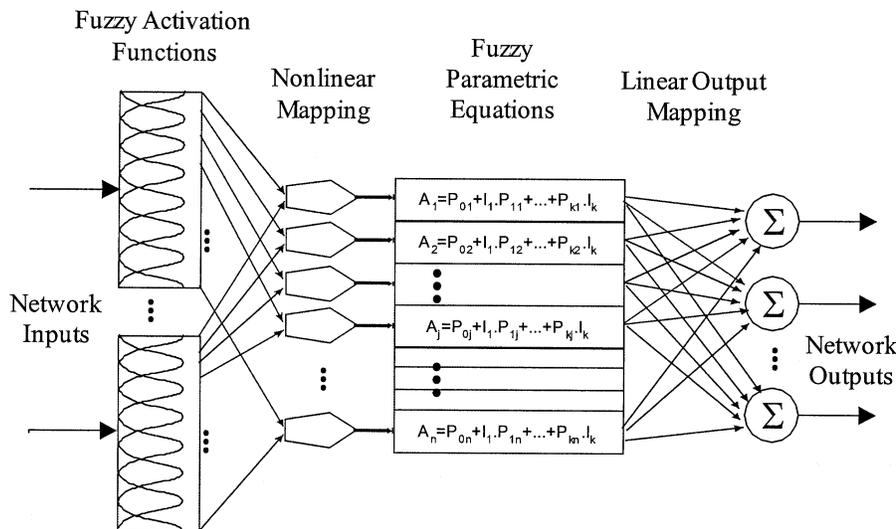


Fig. 3. P-CMAC network basic structure.

tends to increase the representation potential of the resulting algorithm, at the cost of a greater computational effort necessary to process the network mappings

$$A_k = P_{10} + P_{11} \cdot I_1 + \dots + P_{1k} \cdot I_k. \quad (9)$$

The output mapping is executed just as it is in the CMAC, (i.e., the network output vector is calculated through a weighted average of the active memories). Fig. 3 shows a schematic diagram of the P-CMAC network just described.

### III. LEARNING IN P-CMAC NETWORKS

Training in P-CMAC can be performed in a variety of ways, depending on the specific application. Two procedures will be discussed here. First, a supervised schema based on the delta rule is suggested, to adjust the output weights. Second, a recursive least-squares algorithm is recommended, to estimate the optimal parameters of the internal mapping equations. The whole training process is divided into three steps:

- choice of the input space fuzzy partition, defining the input mapping to be used;
- adjustment of the  $P$  parameters in the internal mapping;
- weight adaptation, to define the output mapping.

This process can be iteratively accomplished to achieve adequate results. Optimization of the fuzzy partition can also be incorporated into the parameter estimation algorithm, by including the parameters of each membership defined along with the consequent parameter vector in the recursive least squares scheme. Practical evaluation has yet to be accomplished to certify that an approach like this will not overload the estimation procedure and degrade the system's response time.

Alternatively, one can train a P-CMAC network by means of the classical back propagation algorithm, considering that the network can be viewed as an ordered system (for more details on ordered systems, (see [20], [21]). In this case,  $P$  parameters are treated as normal connection weights inside the network, and partial error derivatives back propagated from the outputs are used to adjust the network behavior. While the first method is more complex and time consuming, it seems to be less sensitive

to local minima, as two different optimization techniques are concurrently employed and tend to improve the search task.

### IV. COMPARISON RESULTS

For the purpose of comparison between the proposed network and existing structures, a nonlinear function was used. It had to be approximated by a CMAC algorithm, an MLP trained with backpropagation (BP) algorithm, and a P-CMAC network. All networks used six-neuron structures. Fig. 4 shows the results accomplished by each, averaged over 100 simulations for each network for MLP and P-CMAC cases, where weight initialization is randomly performed. Each  $X$  axis represents the closed real-valued interval between  $-1$  and  $1$ , and the  $Y$  axes show network outputs (above) and corresponding error (below).

The P-CMAC and MLP presented superior performance when compared to Albus' algorithm. But, the computational requirements of CMAC are much smaller than the other two networks. Table I shows a comparison of the average amount of calculation required, by each of the three algorithms, to perform the discussed approximation (quadratic error  $< 1e-4$ ). Results show the advantages of using the proposed structure over the original CMAC.

Although executing one training cycle of the CMAC is faster, it needs many more cycles to achieve the same accuracy of the P-CMAC and MLP networks, resulting in a larger accumulated computational cost. In comparison to MLP, P-CMAC results are similar in terms of accuracy, while the computational effort required for MLP is larger. This occurs because of the local nature of the P-CMAC training algorithm. For a single training step, considerably fewer weights are adjusted by the P-CMAC algorithm than by the corresponding BP algorithm used in the MLP network, for the same number of nodes in the networks.

### V. APPLICATION TO MOBILE TELEPHONY

Mobile telephony is based on voice and/or data transmission by means of RF electromagnetic waves, which can be generated at cellular phones (called mobile stations) and received at radio-base stations (RBSs), operated by the telephony provider

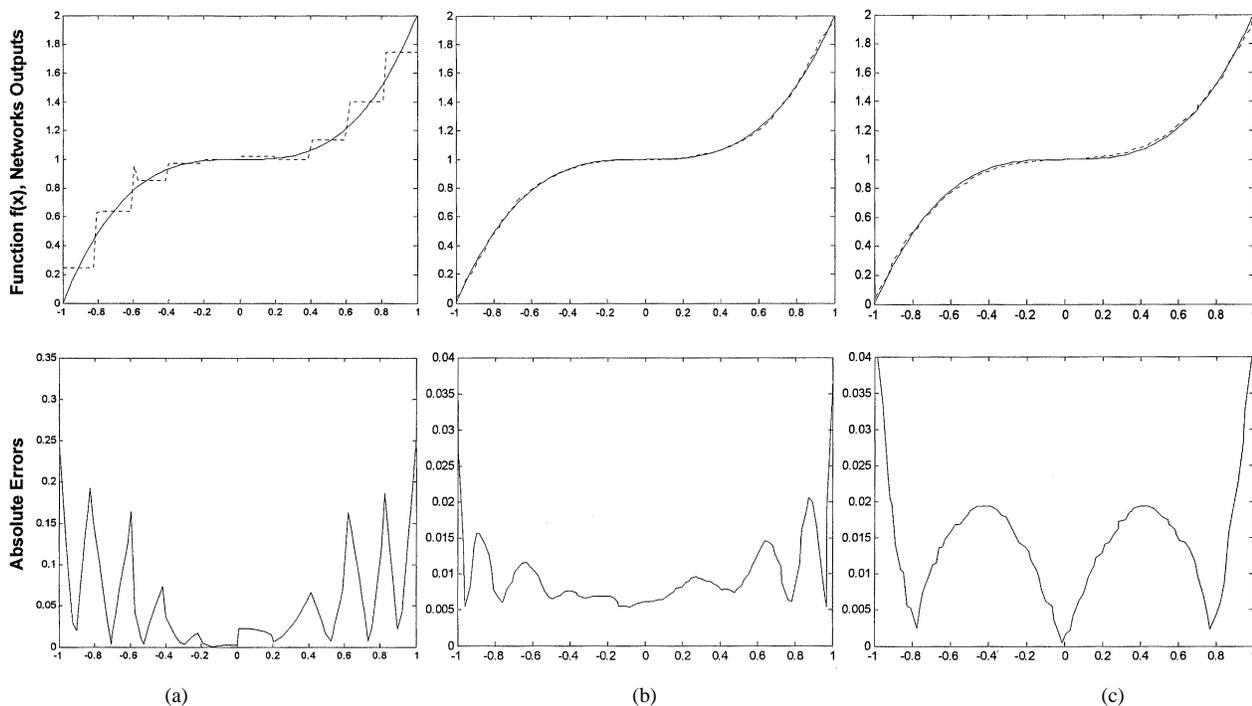


Fig. 4. Comparison between (a) CMAC, (b) MLP, and (c) P-CMAC with BP training.

TABLE I  
COMPUTATIONAL COSTS (IN MEGA-FLOPS) AND RMS ERROR BY CMAC, MLP, AND P-CMAC NETWORKS

	One Cycle Training	Total Training Effort (for RMS Error < 1.0e-4)	Effective Error Reached
CMAC	0.30	8.51	6.89e-3
MLP	0.95	9.35	3.12e-4
P-CMAC	1.43	1.43	5.94e-4

company) or vice-versa. The term *cellular* stands for the operational area of an individual RBS, since the radial signals correspond to a cell around this RBS. The set of all cells within a given region form the overall service coverage area for that region [5].

Very often, a mobile telephony provider needs to measure the RF power levels at a given region of its covering area, to ensure that the system is running well, to maintain communication quality and performance indexes, and to avoid disturbance, noise, and regions with low RF level. For instance, low RF levels can cause communication cutoff and interruptions in a conversation in course. This analyzing process is very tedious and imprecise, since it is necessary to acquire the existing signals from all the interest regions, to organize and to process the acquired data. Moreover, the limitations involved with data acquisition procedure itself and with the available equipment turn the mapping problem into a very complex task, especially when it is necessary to analyze the whole system or large coverage areas.

Given a geographic area covered by a mobile telephony provider, RF mapping operation can be divided into two steps: data acquisition and data processing. Like any other measuring process, this one requires modern and accurate equipment,

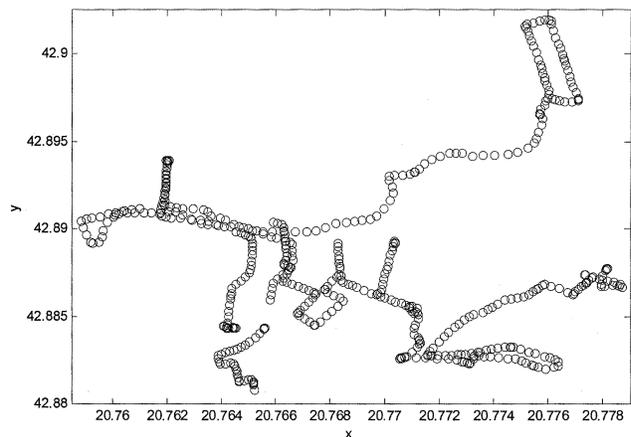


Fig. 5. RF measuring diagram of Brazil (data source: Telemig Celular).

which must provide good noise rejection and promote a reliable and deterministic mapping between actual and measured data.

The major problem at the acquisition step is not the equipment characteristics, but is the adopted acquisition procedure (which is highly asymmetric). This procedure is accomplished as follows. An equipped vehicle runs around the streets at the region of interest, while an acquisition and recording unit remains collecting RF data at constant time intervals; during the acquisition process, the vehicle's speed and direction are changed very often, according to flux directions and traffic conditions at the region being scanned.

The collected data are not equally spaced with respect to geographic positioning and cannot represent, in a symmetrical way, the delimited region of interest for the mapping process. Moreover, there is no way to collect data at areas where access is difficult or impossible for the measuring vehicle (e.g., inside a quarter, private farms, lakes, rivers, industrial plants, etc). Fig. 5 shows a typical measuring diagram obtained from real data col-

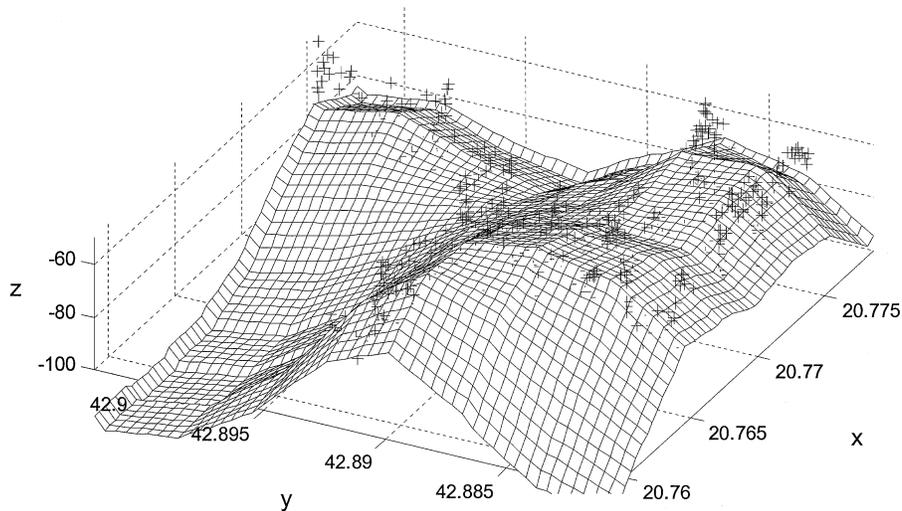


Fig. 6. Approximated surface of RF level, obtained by a P-CMAC network (measured in dBm).

lected by *Telemig Celular*, a mobile telephony provider for the State of Minas Gerais, Brazil. The small circles correspond to the geographic points where the measuring process has been effective, while the total area of the graphic corresponds to the region of interest. Longitude is shown at the  $X$  axis and latitude is shown at the  $Y$  axis.

The second step of the operation—data processing—cannot be accomplished with simple numerical methods. As shown in Fig. 5, the data set obtained at the acquisition process is highly ill conditioned and does not fill the whole region of interest. It is very difficult to find a solution by means of traditional numerical methods described by the approximation theory, as polynomial interpolation and multiple regressions. Another way to obtain an approximated surface that represents the RF level at all the regions of interest is to use a nonlinear approximator (e.g., a neural network algorithm).

Matlab version 5.3, by The MathWorks Inc., Natick, MA, was used to implement the P-CMAC algorithm. Each network had two inputs, (corresponding to the geographical positions recorded by the acquisition equipment), and one output, (corresponding to the RF level to be approximated in the region of interest). At the training phase, these triples were presented to the adapting algorithm and the adjusting parameters were calculated. After the end of the learning phase, the coordinates of all the regions of interest were presented to the inputs, and an approximation surface was obtained at the output. The same procedure described above was applied with the fuzzy-CMAC algorithm.

RF mapping approximation results obtained with P-CMAC are presented (see Fig. 6).  $X$  and  $Y$  axes are the same as in Fig. 5, and the  $Z$  axis represents resulting RF power mapping at each geographical point in the area of interest, measured in dBm (decibels relative to 1 mW). At each computation, a training set of approximately 3000 points was used. The networks sizes were defined empirically, on a trial-and-error basis. Analysis of the accomplished performance and of the computational effort at the calculations dictated changes in the topologies. After 100 experiments with each algorithm, the fuzzy-CMAC algorithm presented an average quadratic error of  $8.22 \times 10^{-2}$ , while

P-CMAC presented an average quadratic error of  $6.94 \times 10^{-3}$ , almost ten times smaller, for equally sized networks [5].

As can be seen in Fig. 6, the approximation process performed well, even if the difficulties due to the ill-conditioned data set are taken into account. For instance, some regions with no measurements, in the neighborhood existing in the data set, always showed low values for RF levels. This could indicate that the P-CMAC network is not adequately activated for these areas, as there is no information available. For those cases, the RF approximation is not reliable. But, for those regions lying closer to measured points, the obtained approximations seem to be consistent. Unfortunately, existing measurement equipment cannot fully support this statement.

With such mapping in hand, operational personnel can have a fairly good idea about the RF power distribution along the area of interest, and can then apply the suitable corrective or preventive actions in order to ensure proper operation of the telephony system. Without this approach, it was much more difficult, and a lot less intuitive for the engineer to make technical decisions about the situation.

## VI. CONCLUSION

P-CMAC networks were described in detail, from their inspiration in CMAC and fuzzy TSK algorithms, to their performance comparison with other neural networks architectures. It is our belief that this architecture is original and innovative, standing as a promising alternative to classical CMAC algorithms for some classes of problems where the performance of such networks is not suitable. Analysis of the comparison experiment results demonstrated advantages of using the proposed architecture over the original CMAC and MLP algorithms. For the described experiment, P-CMAC networks proved to be more powerful than Albus' algorithm and presented a computational cost compatible with the MLP network.

The mobile telephony RF level mapping process was discussed. It is a common problem in the operational quotidian for mobile telephony providers around the world.

P-CMAC was employed to solve the RF mapping problem and the results were very encouraging. Although these results are not yet conclusive, they indeed demonstrate that using alternative techniques such as neural networks to solve complex problems stands today as an important choice for field operational engineers.

#### ACKNOWLEDGMENT

The authors wish to thank Telemig Celular, Brazil, and P. T. Pereira for their valuable contributions, cession of operational data, and description of commonly used measurement procedures.

#### REFERENCES

- [1] J. S. Albus, "A new approach to manipulator control: the cerebellar model articulation controller," *Trans. ASME, J. Dyn. Syst., Meas. Control*, pp. 220–227, Sept. 1975.
- [2] —, "Data storage in the cerebellar model articulation controller," *Trans. ASME, J. Dyn. Syst., Meas. Control*, pp. 228–233, Sept. 1975.
- [3] P. E. M. Almeida, C. S. Filho, and F. G. Jota, "A novel real time integrated environment for the design of practical fuzzy controllers," in *Proc. IEEE 2nd Int. Conf. Knowledge-Based Intelligent Electronic Systems*, Adelaide, Australia, 1998, pp. 155–164.
- [4] P. E. M. Almeida and M. G. Simões, "Fundamentals of a fast convergence parametric CMAC network," in *Proc. IEEE-INNS IJCNN*, vol. 3, Washington, DC, 2001, pp. 3015–3020.
- [5] P. E. M. Almeida, P. T. Pereira, and M. G. Simões, "Mobile telephony RF mapping using fuzzy-CMAC neural networks," in *Proc. 27th IEEE IECON*, Denver, CO, 2001, pp. 1844–1849.
- [6] R. Dunay and G. Horváth, "Modified CMAC neural network architectures for nonlinear dynamic system modeling," in *Proc. Int. Conf. Artificial Neural Networks (ICANN'93)*, Amsterdam, The Netherlands, 1993, p. 473.
- [7] M. Eldracher, A. Staller, and R. Pompl, "Function approximation with continuous valued activation functions in CMAC," *Inst. Informatik, Technische Univ. München, Munich, Germany, Forschungsberichte Kunstliche Intelligenz-94*, 1994.
- [8] Z. J. Geng and C. L. McCullough, "Missile control using fuzzy CMAC neural networks," *AIAA J. Guid., Control, Dyn.*, vol. 20, no. 3, May–June 1997.
- [9] D. P. W. Graham and G. M. T. D'Eleuterio, "Robotic control using a modular architecture of cooperative artificial neural networks," in *Proceedings of the 1991 International Conference on Artificial Neural Networks (ICANN'91)*. Amsterdam, The Netherlands: Elsevier, 1991, pp. 365–370.
- [10] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan/IEEE Press, 1994.
- [11] S. H. Lane, D. A. Handelman, and J. J. Gelfand, "Theory and development of higher-order CMAC neural networks," *IEEE Contr. Syst. Mag.*, vol. 12, pp. 23–30, Apr. 1992.
- [12] J.-C. Lo and Y.-M. Chen, "Stability issues on Takagi-Sugeno fuzzy model-parametric approach," *IEEE Trans. Fuzzy Syst.*, vol. 7, pp. 597–608, Oct. 1999.
- [13] M. R. G. Meireles, P. E. M. Almeida, and M. Godoy Simões, "A comprehensive review for industrial applicability of artificial neural networks," *IEEE Trans. Ind. Electron.*, vol. 50, pp. 585–601, June 2003.
- [14] W. T. Miller, III, F. H. Glanz, and L. G. Kraft, III, "Application of a general learning algorithm to the control of robotic manipulators," *Int. J. Robot. Res.*, vol. 6, no. 2, pp. 84–98, 1987.
- [15] W. S. Mischo, M. Hormel, and H. Tolle, "Neurally inspired associative memories for learning control: A comparison," in *Proceedings of the 1991 International Conference on Artificial Neural Networks (ICANN'91)*. Amsterdam, The Netherlands: Elsevier, 1991.

- [16] M. G. Simões and B. K. Bose, "Application of fuzzy logic in the estimation of power electronic waveforms," in *Conf. Rec. IEEE-IAS Annu. Meeting*, vol. 6, Toronto, ON, Canada, 1993, pp. 853–861.
- [17] M. Sugeno and G. T. Kang, "Structure identification of a fuzzy model," *Fuzzy Sets Syst.*, vol. 28, no. 1, pp. 15–33, 1988.
- [18] R. S. Sutton, "Generalization in reinforcement learning: successful examples using sparse coarse coding," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1996, vol. 8, pp. 1038–1044.
- [19] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, pp. 116–132, Jan./Feb. 1985.
- [20] P. J. Werbos, "Maximizing long-term gas industry profits in two minutes in lotus using neural network methods," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, pp. 315–333, Mar./Apr. 1989.
- [21] P. J. Werbos, T. McAvoy, and T. Su, "Neural networks, system identification and control in the chemical process industries," in *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*, D. A. White and D. A. Sofge, Eds. New York: Van Nostrand Reinhold, 1992.
- [22] H. Ying, "General SISO Takagi-Sugeno fuzzy systems with linear rule consequent are universal approximators," *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 582–587, Nov. 1998.



**Paulo E. M. Almeida** (S'00) received the B.E. and M.Sc. degrees in electrical engineering from the Federal University of Minas Gerais, Belo Horizonte, Brazil, in 1992 and 1996, respectively. He is currently working toward the Dr.E. degree at São Paulo University, São Paulo, Brazil.

He is an Assistant Professor at the Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, Brazil. His research interests are applied artificial intelligence, intelligent control systems, and industrial automation. In 2000–2001, he was a Visiting Scholar in the Division of Engineering, Colorado School of Mines, Golden, where he conducted research in the Mechatronics Laboratory.

Mr. Almeida received a Student Award and a Best Presentation Award from the IEEE Industrial Electronics Society at the 2001 IEEE IECON, held in Denver, CO. He is a member of the Brazilian Automatic Control Society.



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

He joined the faculty of the Colorado School of Mines, Golden, in 2000, and has been working to establish research and education activities in the development of intelligent control for high-power electronics applications in renewable and distributed energy systems.

Dr. Simões was a recipient of a National Science Foundation (NSF) Faculty Early Career Development (CAREER) Award in 2002. It is the NSF's most prestigious award for new faculty members, recognizing activities of teacher/scholars who are considered most likely to become the academic leaders of the 21st century. He is currently serving as IEEE Power Electronics Society Intersociety Chairman, Associate Editor for Energy Conversion as well as Editor for Intelligent Systems of the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, and also serving as Associate Editor for Power Electronics and Drives of the IEEE TRANSACTIONS ON POWER ELECTRONICS. He has been actively involved in the Steering and Organization Committee of the IEEE/DOE/DOD 2003 International Future Energy Challenge.