Implementing Temporal Radial Basis Function for Reactive Navigation of Mobile Robot

Mesbahi Larbi [1], Hendel Fatiha [2], Berrached Nasr-Eddine[2], and Abdelkader Benyettou [1]
[1] Laboratoire SIMPA, University of Sciences and Technology of Oran, Algeria
[2] Laboratoire LARESI, University of Sciences and Technology of Oran, Algeria

Abstract: *We present in this article, the realisation of reactive navigation module based on neural networks Like Temporal Radial Basis Functions (TRBF), with respect of security constraints and inherent robustness while using an Orthogonal Least Square algorithm (OLS). Applied to a structured type like interior of building, the mobile robot must assure its task of navigation mildly all while avoiding obstacles without wandering, with the possibility to take into account the taken decisions in its past lasting trajectory.*

Keywords: *Reactive navigation, TRBF, OLS, mobile robot.*

Received May 18, 2004; accepted August 15, 2004

1. Introduction

Problems concerning traffic mobility, safety, and energy consumption have become more serious in most developed countries in recent years [2]. Add to that the increased need of human to machine which helps him in most delicate cases like nuclear site, submarine depth, and helps to handicaps with a minimum of extern intervention [4].

In this sense different command architectures proposed for mobile robot, meeting in literature can be separated into three classes:

- Deliberative approach: Uses the environment representation to plan in advance the commands e. g., Shakey robot and Stanford Cart architecture.
- Reactive approach: Rests on close coupling between captors and actors, e. g., Brooks's architecture and Rosenblatt propositions.
- Hybrid approach: Combines the above two approaches to take part of their advantages and uses different techniques like (Modules, Supervisor, Scheduler..), e. g., we find different architectures like (Payton, TCA "Task control architecture", SSS "Symbolic Subsumption Servo", Atlantis and Sharp project)[4].

Our navigation type is localised in second approach, it is considered like a basis function of complete system for mobile robot while basing on information nature of the environment.

The general idea consists in associating an elementary displacement of the robot to information of situation. This information is the same type used in environment recognition phase. It is a vector of inputs of distances robot- measures counter environment among the 1024 possible [1]. To this inputs, the network should associate an elementary displacement like: Advance, turn on right or on left [9].

For the phase of recognition, the robot must have a panoramic vision on 360 degrees, independent of its movement [5]. Here in opposite, the taken sense of displacement of the mobile seems indispensable. In addition, it is necessary to recall that achieving neural module is foreseen to function in link with the system of perception and the system of actioneers commands, while following the decision chain [1] (see Figure1).

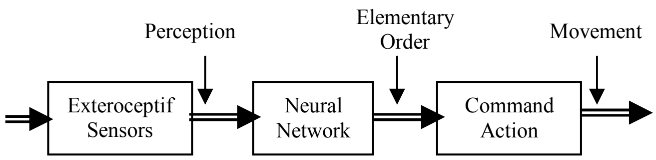


Figure1. Module of navigation.

We are not evidently here in case of generation and follow-up of an optimal trajectory; we ask nevertheless for the chain of navigation to be globally surest possible [11]. We have opts for a neuronal module based on temporal radial basis functions using Orthogonal Least Square contexts, because we think that it adjusts good with this type of problem (overlapping or oscillations owing a confused situation), in this case the taken decision inserted in the past becomes important and useful. So that we can integrate the temporal notion, we needed to play on optimisation of all parameters, in object:

- To have an integral solution in a reliable hardware carried easily and fast.
- To use less expensive sensors.
- Hardiness towards noises and the unforeseen shortcomings.

After this introduction we pass in section 2 to define the application about the navigation of mobile robot. In section 3 we see how to use a network which uses temporal RBF in chain decision, based on OLS algorithm. Results and simulation are presented in the section 4. We finish with conclusion concerning the application while proposing some perspectives.

2. Application

2.1. Problem Definition

We consider eleven elementary situations that a robot can frequently meet inside a building: Passage, impasse, corner, piece, wall, left angle, input, right angle, crossing, T-crossing, and output (see Figure 2).

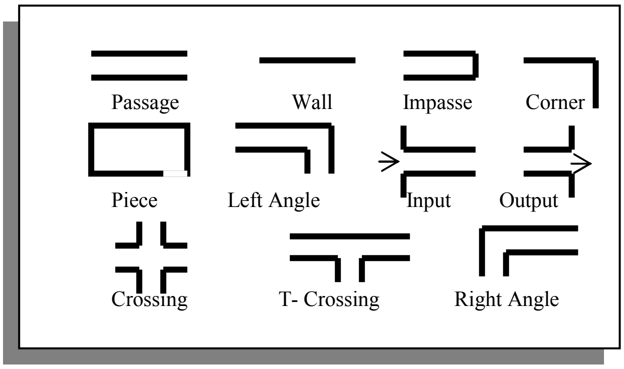


Figure 2. Elementary situations.

In this environment, mobile robot must assure its stain of navigation mildly all while choosing an optimal course: To this effect we can introduce the probabilistic notion in action with hold in amount of the temporal aspect.

We must recall here that the sense of displacement of the robot has an influence on the creation of the training basis and in this goal we must consider the half-plan like source of information for actions of robot (turn to right or turn to left or Advancing).

2.2. Preparing of Learning Basis

For many situations of environment, the mobile is placed in uncertain way in *Np* different positions, with an uncertain initial orientation. Then R rotations of a step θ given are done there, creating R_i "examples" for every Np_i position. We have thus: $(\sum_{i=1} Np_i * (R_i + 1))$ examples with i: 1.. s; s being the number of chosen situations for the navigation.

For every example, the vector of information containing N distances is recorded and a decision of elementary order of movement is chosen (see Figure 3, it shows the taken decisions according to the main direction of the axis of the robot in an environment like passage).

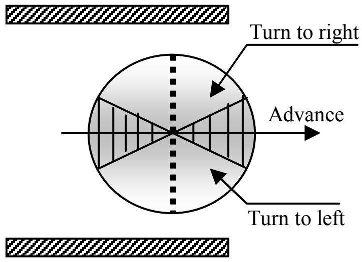


Figure 3. Decisions of displacement according to robot orientation in passage.

At the time of the creation of the basis, we chose decisions in order to direct the mobile toward a trajectory situated toward the middle of the environment (for reason of displacement security).

Figure 4 shows how to take the different orders in particular environment. We distributed this environment in 4 zones: Z1, Z2, Z3, and Z4.

- If the robot is in the zone (Z1), from its position and its initial direction, we will create R (examples) by P rotation of 10°. For every case, the chosen order will be compliant, according to the direction of the main axis, to the Figure 3.
- Suppose now that the robot is in a zone (Z2), with an initial orientation in direction of the left wall. We create as much then of "examples" by rotations of 10° toward the right that it is necessary so that the axis of the robot rejoins the axis of the passage. For every case an order "Turn to Right" will be associated.

If the initial orientation moves away the mobile of the wall, we will associate an order "Advance" and none example won't be creates. By duality, the same procedure has been used for the zone (Z3). If the robot is in the zone (Z4), we will make it turns until it is in a compatible direction with the trajectory indicated on the face.

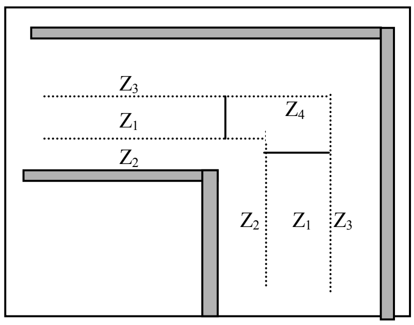


Figure 4. Decisions of displacement in function of position of the robot in angle.

Therefore, we get an organized basis of a stationary number of examples; every example understands 9 measures on the ahead half-plan (180°), knowing that we used situations like (passage, crossing, T-crossing, wall, impasse , corner, angle, door, and piece).

2.3. Advantages of Temporal Approach

This approach permits us to associate to resource our network a certain degree of confidence under a certain probabilistic angle, while basing on neural approximation to take the optimal decision, in order to avoid to knock itself to the wall or to ride.

Another problem concerning the oscillation of the robot especially in the impasse, in this case one of solutions is to introduce the vector of examples composed of measures sequence holds in stationary time delay, as we choose a number of units known in the training for all actions to undertake.

3. Temporal RBF Approach

The capability of multilayer feedforward networks has been theoretically studied. In previous work, Funahashi, Hornik have concluded that "standard multilayer feedforward networks with as few as one hidden layer using arbitrary squashing functions are capable of approximating any Borel measurable function from one finite dimensional space to another to any desired degree of accuracy, provided sufficiently many hidden units are available. In this sense, multilayer feedforward networks are belonging to class of universal approximators" [6].

The classic RBF can be formed to accomplish tasks of the recognition of shapes with no linear and complex contours; they are limited to treat some static models, rather than to treat shapes that are in temporal nature [6]. The Temporal RBF, as Adaptive Time Delay Neural Network (ATDNN), Local Short Term Memory (LSTM)..., is proposed to defeat this limitation. Networks with this capacity can play an important role in domain of applications that has properties varying like temporal signals and dynamic shapes. As we take part of advantages of the classic RBF in approximation and recognition, the objective come closer toward a behaviour wanted by a collection of functions, named kernels [3]. A kernel is characterized by a centre and a receptor field r; these kernels can be chosen by k-means or by vector quantification technique.

In general, the temporal discrimination functions of K classes are written under the following form [6]:

$$y_k(t_n)=\sum_{j=1}^{m1} w_{jk}\,\varphi\Big(\sum_{l=0}^{p} w_j(l)\,x(t_n-l)+b_j\Big)+b_0 \qquad (1)$$

where $x(t_n) = [x(t_n), x(t_n - 1),\ldots x(t_n - p)]^t$; b_j: The bias; p: Is the memory order and m1 is the dimension of the hidden layer. In the following part we describe the network architecture and the training algorithm.

3.1. Network Architecture

Figure 5 shows how estimating the decision functions according to neural network architecture.

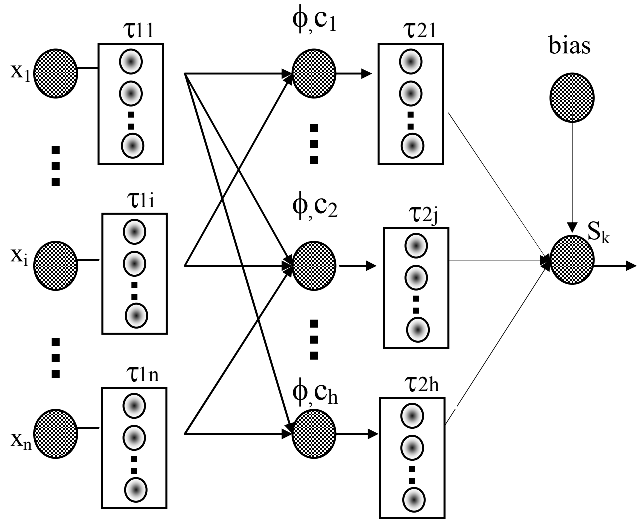


Figure 5. Representation of TRBF network.

Now we define:

$$S[k]=\varphi\Big[W_{ok}+\sum_{j=1}^{m}\sum_{i=1}^{h} Yh[i][j]\,W_{kij}\Big] \qquad (2)$$

where φ: Is an activation function (Linear or Sigmoid) and W_{0k} are the bias.

We can demonstrate that our TRBF is a kind of TDNN network since this last can approximate any spatiotemporal function to a desired accuracy by using the Stone-Weirstrass theorem [6], under some restrictions:

- If ϕ is a bounded and monotone increasing differentiable function.
- At least one hidden layer of N hidden units.
- Having d-times delays elements in each input hidden connections pairs.

Now we describe each block delay (see Figure 6).

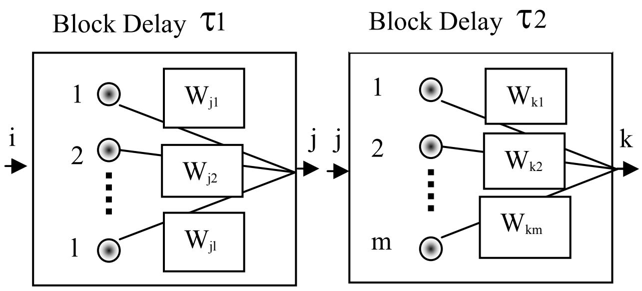


Figure 6. Representation of delay block for (τ1, τ2).

The kernel function is:

$$Yh[i][j] = \phi_{j,\sigma i}(\|C_i - X\|) \qquad (3)$$

where:
i: 1.. h, h: is the number of centres.

j: 1.. m, m: is the dimension of the hidden time delay ($\tau 2$).
Dimension of Here = Dimension of X = n x l.
n: number of feature of the input vector.
l: is the dimension of time delay of input ($\tau 1$).
$\phi_{j,\ \sigma i}$: The kernel function characterized by the time delay of j delay with receiving field σ_i (see Figure 7).

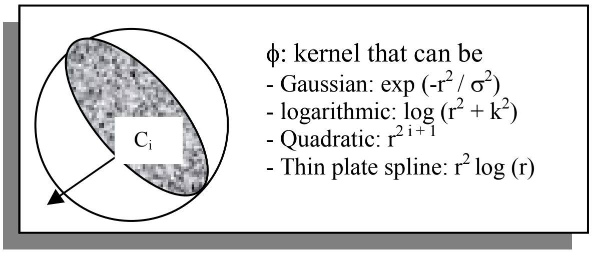


Figure7. Kernel example.

3.2. OLS Training Algorithm

The OLS algorithm is a sequential method, conceived in origin for the identification of no linear systems and permits to make an incremental training [3, 7, 8]. OLS is adapted from the Gram-Schmidt orthogonalization procedure. The algorithm starts by considering all *N* data of the class Ω as centres, then orders them from the most to the less relevant centres: At the first step, the relevance of each centre is estimated by measuring the angle between target d and each column vector P_i. The remaining centres and the output are orthogonalized with respect to the column vector P_i related to the first selected centre, and the procedure is repeated until all centres are ranked. The algorithm must be carried out separately for each subnet [10]. This algorithm can be applied to RBF network that can be considered like a particular case of the regression model linear defined by:

$$d = (P * \theta + E) \tag{4}$$

W is the orthogonal image of *P:*

$$d = (W * A * \theta + E) \tag{5}$$

This equation is used for iterative construction of the RBF network as criteria of selection. Hence an initial whole of *M* centers, the network is constructed iteratively, by the addition of the center that possesses the value [maximal error] and we take the corresponding G_i.

$$G = A * \theta \tag{6}$$

Iteratively we calculate elements of α and W by:

$$\alpha^i_{j,k} = \frac{W^t_j * P_i}{W^t_j * W_j} \tag{7}$$

$$W^i_k = P_i - \sum_{j=1}^{j=k-1} \alpha^i_{j,k} * W_j \tag{8}$$

The criteria of iteration stop known of Akaike:

$$1\text{-}\sum_{i=1}^{M} err_i > \varepsilon \tag{9}$$

In end of iterations, we calculate weights θ_i according to the system, A contains α values:

$$G = A * \theta \tag{10}$$

4. Results and Simulation

4.1. Parameters of Training

We used the method of function networks to combined temporal radial basis with the OLS, applied at every creation of a corresponding network to such action.

We use data normalized in a vector of entrance to 9 measurements with time delay of 2 unit delay to the entrance with hidden layer that calculates its number of hidden neurons following an incremental approach, while following the criteria of Akaike (1-sum_errors > threshold) for the corresponding stop. However, we have limited the training by the choice of only one Gaussian kernel instead of a mixture of kernels (risk of a big complexity).

We played on the value of the receiving field of the kernel that is worth between 1 and the spread of the training basis. In order to accelerate the process of count we have made some modifications on the training basis, like the method of center-reduce data or by shift the data according to following formula [1, 8]: (*x' = 2 * x - 1*).

4.2. Choice of Kernels

Certain authors proposing to choose a variety of kernels in the training like: Thin plate spline, the Gaussian and multiquadratic kernels, but it needs to an enormous count time to choose the best center with the best kernel, in our survey, we fixed the kernel.

4.3. Training and Test Rates

First, these results have been obtained in both training and test by dividing the number of success decision over the total number of proposed examples and respectively by category.

Second, looking at Table 1, we notice after these results that the navigation reacts well with the TRBF training. It comes back to the reduced class number that enters in conflict (3 classes) on the first hand and in other hand to the fact that examples of training base have been chosen minutely.

Table.1 Training rates.

Action	Turn to Left	Advance	Turn to Right	Global Rate
Training rate	95.88%	96.46%	97.9%	96.74%
Test rate	94.23%	95.88%	97.53%	95.88%

4.4. Tolerance to Noise

Table 2 presents the gotten results while adding to data of the validation basis a Gaussian noise of spread (variable between 0.01 and 0.1. we note that until β = 0.05 there nearly is not any reduction of performances. It assures practically that we could replace the laser telemeter by another sensor. When choosing a Gaussian noise we got these results:

Table 2. Comparison between different noised data with a Gaussian noise, while playing on the factor of spreadβ .

Action	Turn to left	Advanc e	Turn to right	Global rate
β = 0.01	94.23%	95.88%	97.53%	96.74%
β = 0.03	95.06%	93.41%	96.7%	95.05%
β = 0.05	93.8%	92.5%	95.47%	93.92%
β = 0.08	90.9%	89.3%	93.41%	91.20%
β = 0.1	88.88%	87.65%	92.18%	89.57%

After these results, we notice that Gaussian noise, in spite of increase of the spread type, didn't drag a total deterioration on the global rate until a spread β = 0.1. We conclude that the margin of the spread type that keeps the best performances belongs to the interval [0.01, 0.05].

4.5. Simulation

4.5.1. Simulation in Structured Environment

We notice seeing Figure 8 that the critic transition through door between two pieces, followed by the entrance in passage, is perfectly realized.

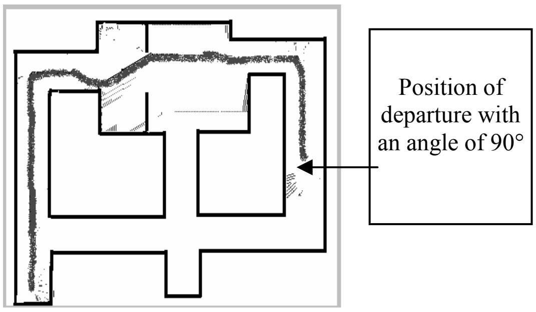


Figure 8. Structured environment test.

In particular, we observe that anticipation capacity of robot in movement is very mildly in center. We recall that there is not any goal attraction strategy.

4.5.2. Simulation in Unknown Environment

The good results in environments of training incited us to do other tests to validate the capacity of the network to make sail correctly the robot in very different situations.

Figure 9 shows that results gotten in a course through passage of shapes bent and of variable widths obliging the trajectory to have some various curvature radiuses. We note a good navigation practically with a regular trajectory without to-stroke. All small obstacle placed close to the centre of the scene is perfectly avoided.

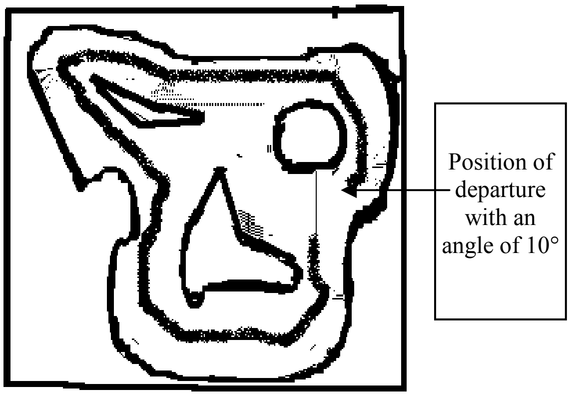


Figure 9. Unknown environment test.

5. Conclusions

We presented in this article the realization of reactive navigation module based on the use of network like TRBF while introducing a stationary time delay to the vector of measures. A training basis and a validation basis have been worked out from elementary environments. In every case, a decision of displacement is chosen among three possibilities: Advance, turn on the right or turn on the left. The done choice generally aims to bring closer the robot for a median trajectory in the environment to cross. We tested various disruption influences then. While adding to measures of distance with a Gaussian noise of variable spread, we showed the hardiness of network screw to screw of a measure noise can go until 5%. It guarantees that our network will be able to function correctly with sensors of modest performance.

Finally, the whole of simulations has been done while letting the robot sail in buckle closed under the conduct of the network. Beginning by simple labyrinths, organized of various assemblies of situations learned, these simulations continued in cases, more and more distant of those of the training basis.

In general, the navigation took place with success. The described trajectory is always very soft; avoid all met obstacles, permits robot turns there in impasse in general way, resemble to what would make an animal or a man in the same situation. In spite of its extreme simplicity, the developed solution for the reactive navigation of our method seems robust, comfortably efficient, and transposable towards other sensors.

References

[1] Alan S. A., "Recognition of Environment and Reactive Navigation of an Autonomous Mobile Robot by Neural Network," *PhD Thesis*, Evry Val d'Essonne, 1996.

[2] Bertozzi M., Broggi A., and Cellario M., "Artificial Vision in Road Vehicles," *IEEE*, vol. 90, no. 7, pp. 1258-1271, July 2002.

[3] Demian V., "Conception and Parallel Algorithm Analysis for Neural Networks Like Kohonen and Radial Basis Function (RBF)," *PhD Thesis*, University of Claude Bernard Lyon, 1995.

[4] Gautier E., "Using of Artificial Neural Network for Autonomic Vehicle Command," *PhD Thesis*, University of Grenoble, January 1999.

[5] Laugier C., Philip Garnier T. F., Parantchic I., and Scheuer A., "Sensor Based Control Architecture for a Car Like Vehicle," RR 3552, INRIA, 1998.

[6] Lin D. T., "The Adaptive Time Delay Neural Network Characterization and Application to Pattern Recognition, Prediction and Signal Processing," *PhD Thesis*, University of Maryland, 1994.

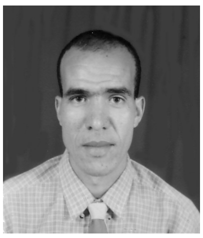
[7] Mesbahi L. and Benyettou A., "A New Contribution Towards a Temporal Radial Basis Function Applied to Mackey-Glass Time Serie," *in Proceedings of ICNNAI'03*, Minsk, Byelorussia, pp. 72-78, 12-14 Nov 2003.

[8] Mesbahi L. and Benyettou A., "Application of Radial Basis Functions to the Recognition of Environment of a Mobile Robot," *in Proceedings of RFIA'04*, Toulouse, France.

[9] Moez C., "Scheduling of Movements for an Autonomous Mobile Robot All Terrain: An Approach by Using Physic Models," *PhD Thesis*, Polytechnic School of Grenoble, 1995.

[10] Oukhellou L. and Aknin P., "Optimization of Radial Basis Function Network for Classification Tasks," *in Proceedings of Neurap'98, IV International Conference on Neural Network and their Applications*, Marseille, pp. 303-307, 1998.

[11] Soruchiari E., "Self Organizing Neural Network for Trajectory Control and Task Coordination of a Mobile Robot," *Connection Science*, vol. 2, no. 3, 1990.



Mesbahi Larbi obtained his Engineering diploma in computer science 1994 from Oran University, and MSc in electronic option vision and pattern recognition in 2001. Currently, in final phase of preparing the PhD in computer science. His research interests include pattern recognition application, neural network, mobile robot applications, and speech recognition.

Hendel Fatiha obtained her engineering diploma in electronic in 1986, and her MSc in electronic in 1991. Now, she is in final phase of PhD in electronic. She teaches in Electronic Department in USTO, Algeria. Her research area includes mobile robots applications, satellite images processing and neural network theories.

Berrached Nasr-Eddine received the BSc of engineering degree in electronics from the University of Sciences and Technology of Oran, Algeria in 1978, and the Diploma of engineering degree in electronics from the University of Sciences and Technology of Lille, France in 1982. He also received the Diploma of engineering degree in computer science from Tokyo Institute of Technology, Japan in 1992 and the PhD degree in electronics from University of Sciences and Technology of Oran, in 1994. He joined University of Sciences and Technology of Oran in 1982, where he is now a Professor at the Electronic Department. From 1986 to 2000, he led the Laboratory of robotics, and since 2000, he is the creator director of the Intelligent Systems Research Laboratory (LARESI). His interests include interface systems, computer vision, pattern recognition, inverse problems, teleoperation, mobile robots, and intelligent systems.

Abdelkader Benyettou received his BSc of engineering in 1982 from the Institute of Telecommunications of Oran, and the MSc degree in 1986 from the University of Sciences and Technology of Oran, Algeria. In 1987, he joined the Computer Sciences Research Center of Nancy, France, where he worked until 1991 on Arabic speech recognition by expert systems and received his PhD in electrical engineering in 1993 from the University of Sciences and Technology of Oran. Since 2003, he has been a professor at the University of Sciences and Technology of Oran. His interests are in the area of speech and image processing, Arabic speech recognition, neural networks, and machine learning. He has been the director of the Signal-Speech-Image– SIMPA Laboratory, Department of Computer Sciences, Faculty of Sciences, University of Sciences and Technology of Oran, since 2002.