

## dFasArt: Dynamic neural processing in FasArt model

Jose-Manuel Cano-Izquierdo\*, Miguel Almonacid, Miguel Pinzolas, Julio Ibarrola

Department of Systems Engineering and Automatic Control, School of Industrial Engineering, Technical University of Cartagena, Campus Muralla del Mar, 30202 Cartagena, Murcia, Spain

### ARTICLE INFO

#### Article history:

Received 9 October 2006

Accepted 15 September 2008

#### Keywords:

Adaptive resonance theory

Neuronal dynamics

FasArt

Dynamic pattern recognition

### ABSTRACT

The temporal character of the input is, generally, not taken into account in the neural models. This paper presents an extension of the FasArt model focused on the treatment of temporal signals.

FasArt model is proposed as an integration of the characteristic elements of the Fuzzy System Theory in an ART architecture. A duality between the activation concept and membership function is established. FasArt maintains the structure of the Fuzzy ARTMAP architecture, implying a static character since the dynamic response of the input is not considered.

The proposed novel model, dynamic FasArt (dFasArt), uses dynamic equations for the processing stages of FasArt: activation, matching and learning. The new formulation of dFasArt includes time as another characteristic of the input. This allows the activation of the units to have a history-dependent character instead of being only a function of the last input value. Therefore, dFasArt model is robust to spurious values and noisy inputs.

As experimental work, some cases have been used to check the robustness of dFasArt. A possible application has been proposed for the detection of variations in the system dynamics.

© 2008 Elsevier Ltd. All rights reserved.

### 1. Introduction

Dealing with the problem of recognition of temporal patterns, the arriving order of data constitutes an important part of the information in order to establish a classification of time series. It is necessary to consider the temporal arrangement of the patterns as part of their characteristics. Usually, the methods proposed for pattern classification do not make use of this information, (Bezdek & Pal, 1992; Devijver & Kittler, 1982; Duda & Hart, 1973; Nigrin, 1993; Pao, 1989). It can be said that patterns are processed independently of their temporal disposition. However, the temporal concept is taken into account in the Adaptive Resonance Theory (ART) because of the dynamic character of their activation equations. This is suggested in Rajmakers and Molenaar (1997) and Carpenter, Grossberg, and Rosen (1991b), but usually the algorithm used just considers the stationary solution of the model differential equations (Carpenter, Grossberg, & Rossen, 1991c). Specific architectures have been proposed, like the STORE model (Bradski, Carpenter, & Grossberg, 1994) for the recognition of temporal sequences or the ART-EMAP model (Carpenter & Ross, 1995) that allows temporal evidence accumulation. ART 3 (Carpenter & Grossberg, 1990) might be the model that best

approaches the problem that outlines the recognition of temporal patterns because it allows the tracking of time-variant patterns. This requires the inclusion of new dynamics associated to the liberation of neurotransmitters on the neural synapses.

Different combinations of Fuzzy Set Theory have been included in the neural models. The Fuzzy ART architecture (Carpenter, Grossberg, & Rosen, 1991a) and Fuzzy ARTMAP (Carpenter, Grossberg, Markuzon, Reynolds, & Rosen, 1992) are the most important models among the ART ones. The FasArt model (Cano, Dimitriadis, Araúzo, & Coronado, 1996) and its derivation FasBack (Cano, Dimitriadis, & Coronado, 1997) maintain the ARTMAP structure but rigorously including the principles of fuzzy sets. This allows establishing a dual vision of the model as a neural architecture and as a Fuzzy Logic System (FLS). At first, the FasArt/FasBack models were applied to identification and control system tasks (Arauzo et al., 2004; Cano, Dimitriadis, Gómez, & Coronado, 2001), and to the pattern recognition problems (Gómez et al., 2001). An unsupervised version called UFasArt (Sainz, Dimitriadis, Cano, Gómez, & Parrado, 2000; Sainz, Dimitriadis, Sanz, & Cano, 2002) has been also proposed associated with this issue.

The paper is organized as follows. Section 2 describes the main characteristics of FasArt model within the Fuzzy ART architectures in the frame of the ART Theory. Section 3 proposes the dynamic equations defining the new model. Several examples have been selected to test its behaviour under different inputs including inputs with noise. An application of dFasArt model, described in Section 4, is the detection of changes in the system dynamics. The

\* Corresponding author.

E-mail addresses: [JoseM.Cano@upct.es](mailto:JoseM.Cano@upct.es) (J.-M. Cano-Izquierdo), [Miguel.Almonacid@upct.es](mailto:Miguel.Almonacid@upct.es) (M. Almonacid), [Miguel.Pinzolas@upct.es](mailto:Miguel.Pinzolas@upct.es) (M. Pinzolas), [JulioJose.Ibarrola@upct.es](mailto:JulioJose.Ibarrola@upct.es) (J. Ibarrola).

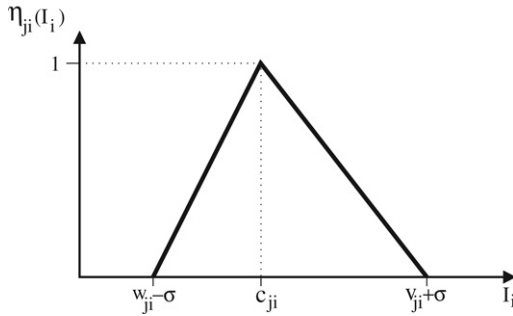


Fig. 1. Activation/membership function in FasArt model.

plant is a simulated DC motor controlled in a closed loop. Finally, conclusions are given in Section 5.

## 2. FasArt model: ART based fuzzy adaptive system

FasArt model links the ART architecture with Fuzzy Logic Systems, establishing a relationship among the unit activation function and the membership function of a fuzzy set. On the one hand this allows interpreting each of the FasArt unit as a fuzzy class defined by the membership–activation function associated to the representing unit. On the other hand, the rules that relate the different classes are determined by the connection weights among the units.

FasArt uses an activation function determined by the weights of the unit as the membership function of a fuzzy set. The signal activation is calculated as the AND of the activations of each one of the dimensions when a multidimensional signal is considered. This AND is implemented using the product as a T-norm. Hence, the activity of unit  $j$  ( $T_j$ ) for an  $M$ -dimensional input  $\vec{I} = (I_1 \dots I_M)$  is given by:

$$T_j = \prod_{i=1}^M \eta_{ji}(I_i) \quad (1)$$

where  $\eta_{ji}$  is the membership function associated to the  $i$ th-dimension of unit  $j$ , determined by the weights  $w_{ji}$ ,  $c_{ji}$  and  $v_{ji}$ , it is shown in Fig. 1. The  $\sigma$  parameter determines the fuzziness of the class associated to the unit.

The election of the winning unit  $J$  is carried out following the winner-takes-all rule, hence:

$$T_j = \max_j \{T_j\}. \quad (2)$$

The learning process starts when the winning unit meets a criterion. That is usually related with the size of the class, considering the input as a pattern attached to this class. This size is calculated by the sum of those sides of the resulting hyperbox:

$$R_j = \sum_{i=1}^M (\max(v_{ji}, I_i) - \min(w_{ji}, I_i)). \quad (3)$$

This  $R_j$  value, called RESET level, is compared with a function  $h(\rho)$  of the  $\rho$  design parameter, so that:

- If:  $R_j \leq h(\rho)$  (4)

the matching between the input and the weight vector of the unit is good, and the learning task starts.

- If:  $R_j > h(\rho)$  (5)

there is not enough similarity, so the RESET mechanism is fired. This inhibits the activation of unit  $J$ , returning to the election of a new winning unit.

In the case of Fuzzy ART and FasArt this function comes defined as:

$$h(\rho) = \rho M. \quad (6)$$

If the winning unit achieves the matching condition (RESET) the learning phase is activated and the unit modifies its weights. The *Fast-Learning* concept is commonly used. When the winning unit represents a class that had performed some other learning cycle (committed unit) the weights are updated according to the equations:

$$\begin{aligned} \vec{W}_j^{NEW} &= \min(\vec{I}, \vec{W}_j^{OLD}) \\ \vec{C}_j^{NEW} &= \vec{C}_j^{OLD} + \beta (\vec{I} - \vec{C}_j^{OLD}) \\ \vec{V}_j^{NEW} &= \max(\vec{I}, \vec{V}_j^{OLD}). \end{aligned} \quad (7)$$

For the case of the uncommitted units, the class is initialized with the first categorized value, hence *Fast-Commit*:

$$\begin{aligned} \vec{W}_j^{NEW} &= \vec{I} \\ \vec{C}_j^{NEW} &= \vec{I} \\ \vec{V}_j^{NEW} &= \vec{I}. \end{aligned} \quad (8)$$

Related with the FasArt equations it can be realized that the learning task and the RESET calculus coincide with Fuzzy ART (Carpenter et al., 1991a). However, the formulation proposed in the case of FasArt denotes the different philosophy that inspires both models. In the case of FasArt,  $\min(\vec{I}, \vec{W}_j)$  is written since what is calculated is the minimum between two vectors, so it can not be computed as  $\vec{I} \wedge \vec{W}$  (representing the intersection of two fuzzy sets).  $\vec{I}$  and  $\vec{W}$  are not fuzzy sets, i.e. their membership functions are not defined at any time. Therefore, the fuzzy character of Fuzzy ART is questionable, as pointed out in Simpson (1993). As a difference, FasArt maintains a formal analogy with a fuzzy logic system, relating each unit with a fuzzy set (fuzzy class) whose membership function corresponds to the activation function.

### 2.1. Normalization of the inputs in FasArt

The complementary code used in Fuzzy ARTMAP model (Carpenter et al., 1992) allows the normalization of the inputs maintaining the relative importance of each component. Nevertheless, the use of this coding implies that the components of the input vector should be inside the  $[0, 1]$  interval. When using generic signals, this condition is often not satisfied and a pre-normalization of the inputs is necessary. This pre-processing implies a previous knowledge of the maximum and minimum values of the signal. This is not always fulfilled. Therefore, the requirement for the input of being inside the  $[0, 1]$  range, limits the on-line adaptive character of the system since it implies a previous signal pre-processing.

The constraint on the input values is eliminated when using the proposed formulation for FasArt, since the fuzzy sets can be defined along the whole real axis. In contrast, the necessity of determining the values of  $\sigma$  and  $h(\rho)$  is introduced, so that the variability of the maximum and minimum values in the inputs and that they are not known a priori, is assumed. The value of  $\sigma$  is the level of generalization assigned to the fuzzy set. On the one hand, it is necessary to remember that the  $w_i$  and  $v_i$  values have had inputs that have been classified in the unit at any time and they have contributed to its definition. On the other hand, the  $c_i$  value represents a typical value of the category; so the generalization can be computed as a part of it. Hence,  $\sigma$  is proposed as:

$$\sigma = \sigma^* |2c_j| + \epsilon \quad (9)$$

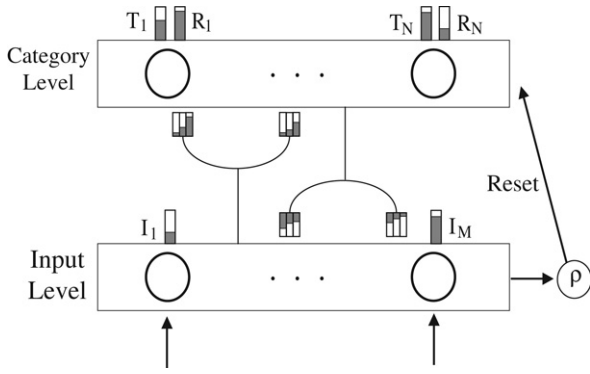


Fig. 2. dFasArt structure.

where the  $\sigma^*$  value is the generality level of the associated fuzzy set: values of  $\sigma^* \rightarrow 0$  make the set less fuzzy (more crisp), while values of  $\sigma^* \rightarrow \infty$  increase set fuzziness. The parameter  $\epsilon > 0$  fixes the minimum value of  $\sigma$ .

Another aspect when considering unconstrained input values is the calculus of the RESET level. As the maximum value of each individual length of the hyperbox is not limited to 1 and taking into account that each component contributes with the same equivalent information, the size of the hyperbox must be weighted by its order of magnitude. Thus:

$$R_j = \sum_{i=1}^M \left( \frac{\max(v_{ji}, I_i) - \min(w_{ji}, I_i)}{|2c_{ji}| + \epsilon} \right). \quad (10)$$

This set of changes allows to process signals at any range, without a previous knowledge of them. This feature increases the operation range of the model achieving a truly adaptive and on-line character.

### 3. FasArt with dynamics: dFasArt

In other to use FasArt for clustering temporal inputs, a dynamic formulation for the operation equations of FasArt is proposed. This allows considering some temporal features of the input (arriving order, persistence) which are as relevant as the signal value.

The dFasArt architecture, shown in Fig. 2, maintains the ART structure but adding dynamic equations for the RESET, weights and the activation calculus. This last one directly derives from the activity equation in FasArt by:

$$\frac{dT_j}{dt} = -A_T T_j + B_T \prod_{i=1}^M \eta_{ji}(I_i(t)). \quad (11)$$

Eq. (11) includes a term of passive decay with decay rate:  $A_T$ , that reduces the unit activity when the signal moves away from the hyperbox that defines it and a term of excitation with excitation gain  $B_T$ . If the condition  $A_T = B_T$  is imposed in this equation and if input  $\vec{I}(t)$  remains time-constant, then  $T_j \rightarrow \prod_{i=1}^M \eta_{ji}(I_i)$  when  $t \rightarrow \infty$ . So, the system behaves as the original FasArt model.

Let us consider no dynamics to calculate the RESET level, since this mechanism generates a typical characteristic of the classes and therefore an aspect that is not generally time-dependent. It can be said that the maximum size of the category/hyperbox is a system characteristic of the same type than the input pattern dimension ( $M$ ). This implies that the RESET signal is integrated at a much higher rate than the activation one. So the differential equation which describes the RESET activation can be taken as an algebraic equation. However, dealing with noisy data, it is appropriate to use a dynamic formulation, as is shown in dFasArt structure (Fig. 2):

$$\frac{dR_j}{dt} = -A_R R_j + B_R \sum_{i=1}^M \left( \frac{\max(v_{ji}, I_i) - \min(w_{ji}, I_i)}{|2c_{ji}| + \epsilon} \right). \quad (12)$$

If the condition of having the decay rate the same value as excitation gain ( $A_R = B_R$ ) is imposed, the system behaviour approaches in time to that of the FasArt static model in the case of constant input.

Dynamics is also considered in the learning equations, since it seems evident that the persistence or duration of a certain value as input signal should influence in the learning, as can be seen in Fig. 2. To do this, a learning dynamics based on the principle of *Fast-Commit Slow-Recode* is outlined. The initialization in FasArt is maintained for uncommitted nodes, so that the category is initialized with the input which generated it. When the learning category is a previously created committed node, its weights are modified according to the following dynamic equations:

$$\begin{aligned} \frac{d\vec{W}}{dt} &= -A_W \vec{W} + B_W \min(\vec{I}(t), \vec{W}) \\ \frac{d\vec{C}}{dt} &= \beta(\vec{I} - \vec{C}) \\ \frac{d\vec{V}}{dt} &= -A_V \vec{V} + B_V \max(\vec{I}(t), \vec{V}) \end{aligned} \quad (13)$$

where  $A_W$  and  $A_V$  are passive decay rates,  $B_W$  and  $B_V$  excitations gains and  $\beta$  is the learning rate.

The  $A_W = B_W$  and  $A_V = B_V$  conditions assure that the weights approach the values of the FasArt static model when the input signal remains time-constant.

#### 3.1. Activation and reset in dFasArt

Let the input be a sinusoidal signal  $I(t) = \sin(0.5\pi t)$  (Fig. 3(a)) and a unit with weight vector  $w = 0.5$ ,  $c = 0.6$  y  $v = 0.7$ . The activation equation (11) has been considered with a forgetting factor rate  $A_R = 0.99$ , an excitation gain  $B_R = 1$  and a minimum fuzziness  $\sigma^* = 0$ . Simulation results of the unit behaviour are shown in Fig. 3(b)(c). The unit begins to respond (increasing activity) when input  $I(t)$  reaches the weight value, ( $I(t) = w(t) = 0.5$ ). At this point, the neuron starts to increase its activity and it continues doing so until the input comes out of its hyperbox influence (in this case, the segment  $[0.5, 0.7]$ ). Starting from the output, the unit begins losing its activity, increasing it again when it passes through the hyperbox again. A memory effect takes place in this second passing, since there has been recently an activation. This effect is observed in Fig. 3(b), where the new peak of maximum activation is bigger than the previous one. This clearly reflects the dynamic character of the net. Faced with the same input, the system operates in a different way (different level of maximum activity) depending on the past history (temporal distance of the last activation increase). This is shown in the second sinusoidal wave: as time passes, the neuron relaxes and its response is weaker.

The results obtained in this simulation can be related to the experimental results of pyramidal neurons submitted to a double pulse of excitement with a short time interval between pulse and pulse. In Polsky, Mel, and Schiller (2004) it can be clearly seen under the mentioned circumstances how the response to the second stimulus overcomes the one given to the first one.

According to the RESET level value shown in Fig. 3(c), it is observed that it remains constant and with a minimum value when the signal remains in the hyperbox that defines the category. The RESET level is increased or decreased when the signal passes away or comes closer to the hyperbox with the same speed of that increase/decrease according to the case. Evidently, any memory effect is taken into account given the algebraic character of the equation that defines it.

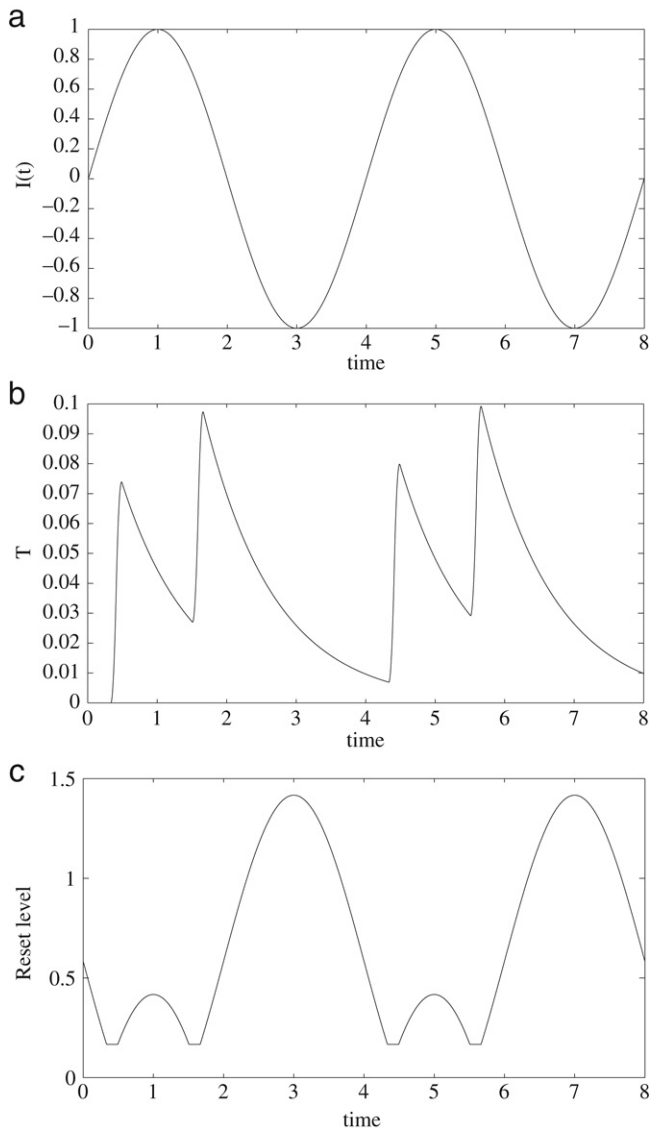


Fig. 3. Activation and RESET level of the unit in response to a sinusoidal signal.

### 3.2. Noisy data

The appearance of noise is a frequent problem dealing with experimental data. Noise is a serious inconvenience in ARTMAP (Carpenter, 1997) architectures, since it can generate proliferation of categories. Although FasArt model allows processing data with noise (Cano et al., 2001) problems can arise due to an excess of categories when the activation value becomes zero because of the noise. An example of this behaviour is illustrated in Fig. 4(a) where a Gaussian noise of zero mean and 0.05 variance has been added to the original sinusoidal signal. For the activation and the RESET calculus no dynamics have been considered. Comparing figures Figs. 3(b) and 4(b) it can be seen that the activation falls to zero many times. This is because the noise added to the original signal yields the input  $I(t)$  out of the range  $[w - \sigma, v + \sigma]$  where the activation is not zero, just as Fig. 1 reflects. This kind of behaviour would produce superimposed multiple categories (that would be superimposed).

When dynamic equations are used to calculate both, the unit activation and the RESET level, there appears a filtering effect at certain input frequencies (Fig. 5). This has been observed in the behaviour of biological neurons, (Mel, 1994). It allows the noise effect becoming considerably attenuated when dynamic character

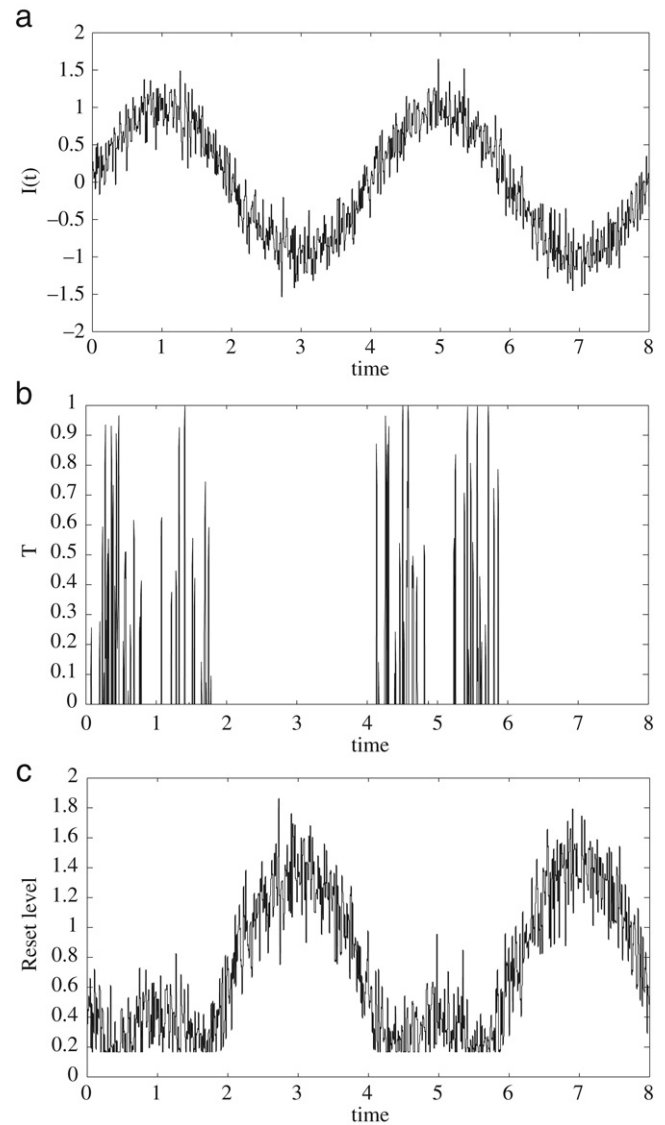


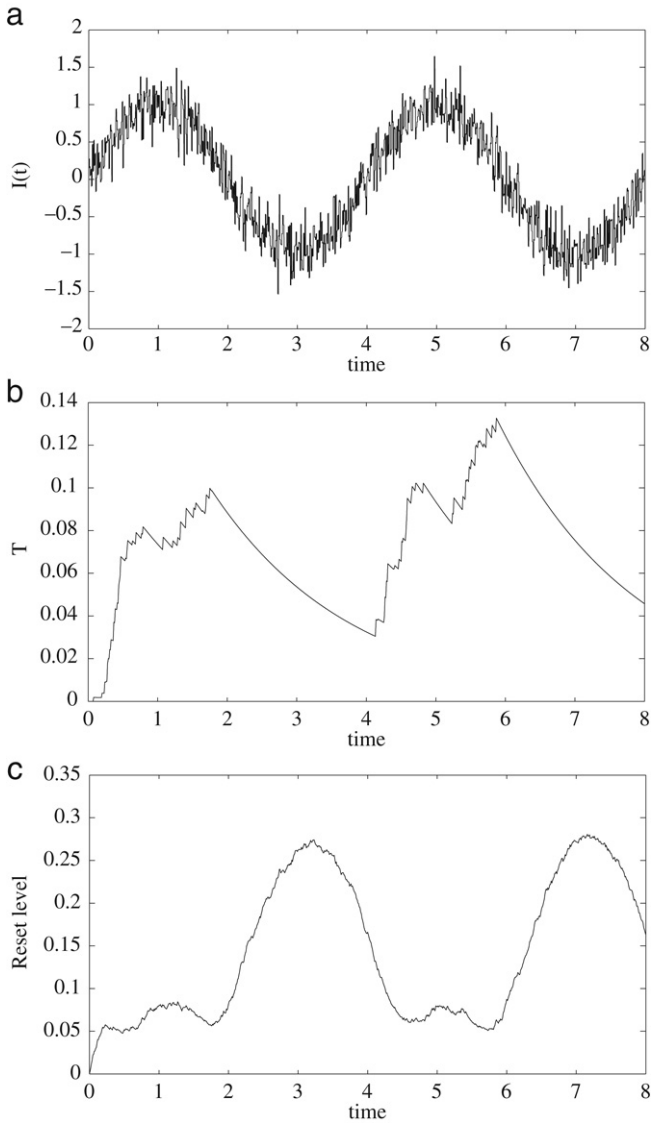
Fig. 4. Unit activation (b) and RESET level (c) responses to a sinusoidal input signal (a) with additive noise when dynamics is not considered in neither the activity equation nor the RESET.

is taken into account in the activation unit and RESET signal calculus. Values of  $A_T = 0.5$  and  $B_T = 1$  was set for the  $T_j$  equation and  $A_R = 5$ ,  $B_R = 1$  for the RESET equation, are used in Fig. 5.

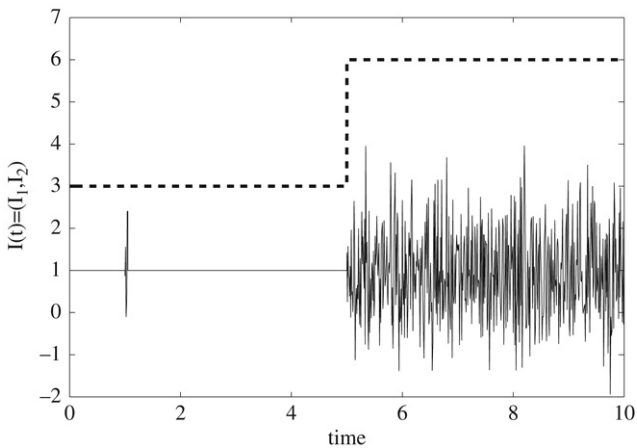
### 3.3. Weight learning

A system with a two-dimensional input  $I(t) = (I_1, I_2)$  is selected in order to illustrate the performance of the learning equations. The first one is a step input ( $I_1$ ), passing from value:  $I_1(t) = 3$  in  $t \in [0, 5)$  to value:  $I_1(t) = 6$  in  $t \in [5, 10]$  (Fig. 6, dotted line). The second input is a constant value:  $I_2(t) = 1$ , but in  $t \in [1, 1.1)$  and  $t \in [5, 10]$  a Gaussian noise (mean = 0 and variance = 1)  $e(t) = N(0, 1)$  is added (Fig. 6, solid line).

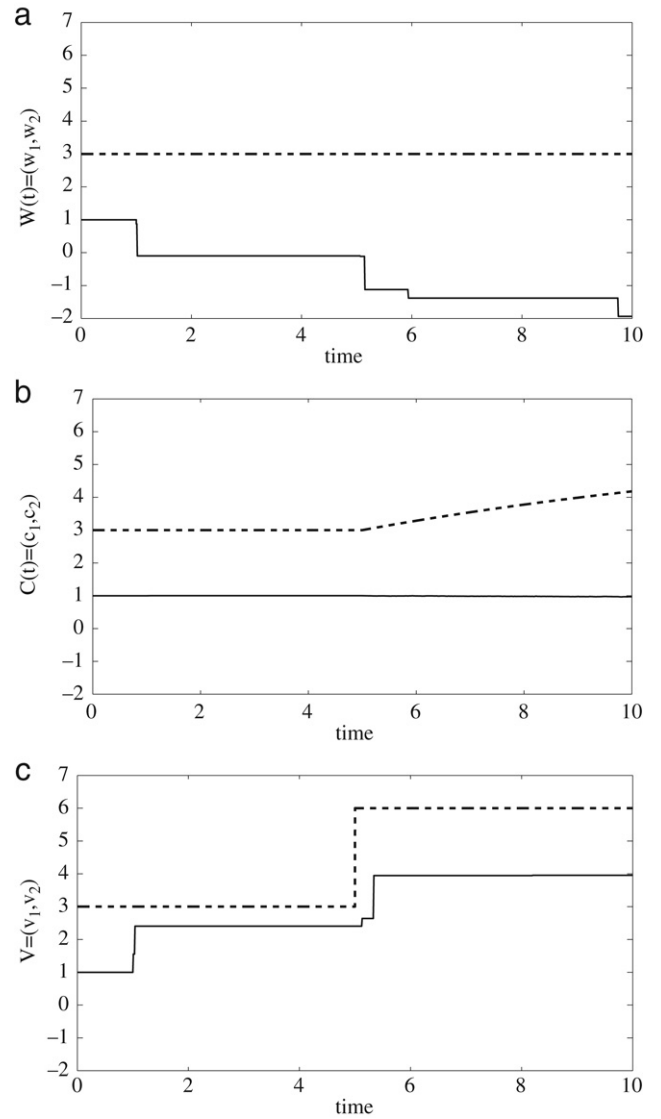
It is considered that a unit with initial weight vector:  $\vec{W}(0) = \vec{C}(0) = \vec{V}(0) = (1, 3)$  is learning all the time by the input ( $A_W = B_W = A_V = B_V = 0.9$  and  $\beta = 0.1$ ). Taken into account the input vector characteristics it is expected that the change in the input value  $I(1)$  will be reflected in the first component of the weights ( $w_1, c_1, v_1$ ). Fig. 7 shows the weight behaviour when FasArt equations are used. It is observed that the weight corresponding with the minimum learned value remains the same



**Fig. 5.** Unit activation (b) and RESET level (c) responses to a sinusoidal input signal (a) with additive noise when dynamics is considered in the activity equation and in the RESET.



**Fig. 6.** Two-dimensional input vector. One dimension is represented by the dotted line and the other one by the continuous line. At time  $t = 5$ , a change in the input values is produced.



**Fig. 7.** Weight learning for two-dimensional input vector. One dimension is represented by the dotted line and the other one by the continuous line. At time  $t = 5$ , a change in the input values is produced. (a)  $w_1$  and  $w_2$  weights, (b)  $c_1$  and  $c_2$  weights,  $v_1$  and  $v_2$  weights.

(Fig. 7(a), dotted line), the one corresponding with the maximum value (Fig. 7(c), dotted line) reflects the change in the input, and the weight corresponding to the value of maximum membership (Fig. 7(b), dotted line) is modified taking a value between the maximum and minimum.

When the proposed dynamic learning equations are used, a similar behaviour is observed with some differences (Fig. 8, dotted lines). The change in weight  $V_1$  (Fig. 8(c), dotted line) is not instantaneous, a continued presence of the input is necessary for this weight to get the final value.

The particular characteristics of the dynamic learning are clearly presented in the learning process of the input second component  $I_2(t)$  (Fig. 6, solid line). This input presents at  $t = 1$  a small disturbance that produces an important change in the weights  $w_2$  and  $v_2$  in the case of fast learning (without dynamics) as shown in Fig. 7(b)(c) solid lines, and nothing when dynamic learning is considered. This short time disturbance is filtered, as shown in Fig. 8(b)(c) solid lines. When the noise in the input is persistent ( $t > 5$ ) the weight variations are high in the case of fast learning (Fig. 7(b)(c), solid line) and smooth in the case of dynamic learning (Fig. 8(b)(c), solid line).

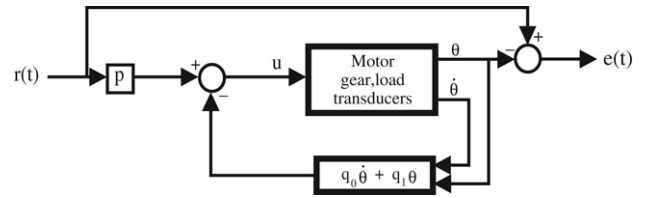


Fig. 9. PD control scheme for the DC motor.

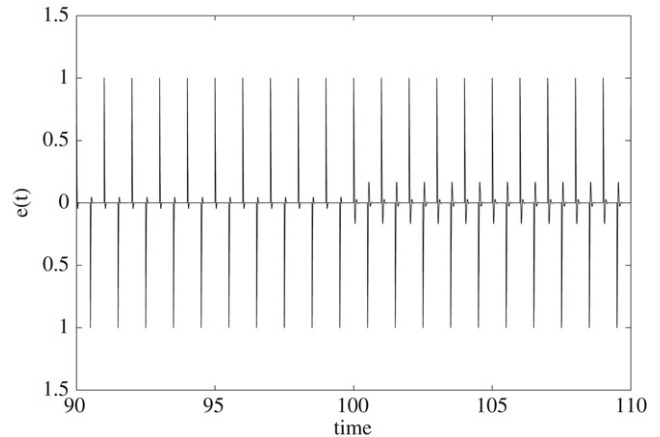


Fig. 10. Error signal at time interval  $90 < t < 110$ .

Fig. 8. Weight learning for two-dimensional input vector. One dimension is represented by the dotted line and the other one by the continuous line. At time  $t = 5$ , a change in the input values is produced. (a)  $w_1$  and  $w_2$  weights, (b)  $c_1$  and  $c_2$  weights,  $v_1$  and  $v_2$  weights.

In both cases the learning equation of the weight  $c_2$ , which determines the maximum membership point, makes the system to maintain a same “prototype” of the input  $c_2$ , filtering the influence of the noise.

**4. Experimental results**

A DC motor with a PD control has been considered as a possible practical application for the neural processing system. A change in the parameters of the motor will imply a non-stationary state in the process. The aim is to study the nature of the system by output observation. The fact that the global system is controlled in a closed loop makes its study difficult, since the controller tends to cancel the effect of changes in the parameters of the motor, in order to maintain the global system behaviour (i.e. the tracking of a certain reference). Dealing with closed loop systems allow a closer approach to industrial processes, where data usually comes from controlled variables. Model identification in closed loop systems is very important and has been studied in different works (Landau, 2001; Zhu & Butoyi, 2002).

A DC motor is considered, where the input voltage is applied to the armature circuit of the motor, while fixed voltage is applied to the field winding (Matko, Karba, & Zupancic, 1992). This motor can be modelled as a dynamic system where the input  $u(t)$  is the voltage applied to the motor and the output  $\theta(t)$  is the angular position:

$$J \frac{d^2\theta}{dt^2} + c_f \frac{d\theta}{dt} = \frac{k_T}{R} \left( u - k_e \frac{d\theta}{dt} \right) \tag{14}$$

where  $k_T$  is the motor torque constant,  $k_e$  is the back emf constant,  $J$  is the inertia of the combination load and gear train referenced to the motor shaft and  $c_f$  is the viscous-friction coefficient of the whole system, also referenced to the motor shell. The numeric values are the same proposed in Matko et al. (1992):  $k_T = 0.1146 \text{ Nm/A}$ ,  $c_f = 0.001 \text{ N ms/rad}$ ,  $k_e = 0.1435 \text{ V s/rad}$ ,  $J = 1.5 \times 10^{-4} \text{ kg m}^2/\text{rad}$ ,  $R = 0.93 \text{ }\Omega$ .

The position control of the motor is carried out using a PD controller (Fig. 9). The controller parameters are:  $p = 12.17$ ,  $q_0 = 0.0188$ ,  $q_1 = 12.17$ . The reference for angular position  $r(t)$  of the motor is a square signal with a period of  $T = 1$ , amplitude  $A_r = 1$ , and a pulse width of 50% of the period.

At time  $t = 100$ , a change in the load inertia of the motor is considered, from  $J(t \leq 100) = 1.5 \times 10^{-4}$  to a new value  $J(t > 100) = 3 \times 10^{-4}$ . This variation produces a change in the behaviour of the closed loop system. This change should be detected by studying the time series of the control error  $e(t)$ , obtained subtracting the output signal from the reference, which it is the angular position of the motor. The error signal  $e(t)$  is represented in Fig. 10 at time interval  $90 \leq t \leq 110$ . Note that the action of the controller prevents the output from moving away from the reference even with the change in the parameter of the motor.

The change in the value of the load inertia causes a variation in the global system (motor and controller) dynamics. This variation is reflected in the appearance of a non-stationary state in the error variable which can be studied with the frequency composition of the signal. The power spectrum is used, which







