



This is a repository copy of *Fusion ARTMAP: Clarification, Implementation and Developments*.

White Rose Research Online URL for this paper:
<http://eprints.whiterose.ac.uk/80099/>

Monograph:

Harrison, R.F. and Borges, J.M. (1995) *Fusion ARTMAP: Clarification, Implementation and Developments*. Research Report. ACSE Research Report 589 . Department of Automatic Control and Systems Engineering

Reuse

Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher's website.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

X
629
.8
(S)

Fusion ARTMAP: Clarification, Implementation and Developments.

R.F. Harrison and J.M. Borges

Department of Automatic Control and Systems Engineering

The University of Sheffield

Mappin Street, Sheffield S1 3JD

United Kingdom

E-mail r.f.harrison@sheffield.ac.uk, j.m.borges@sheffield.ac.uk

Research Report No. 589

August 1995

I. Introduction.

This report aims to clarify how the fusion ARTMAP neural network operates. Here we explore the case of multiple-input sensors with single-teacher channel architecture and ART sensor modules using independent baseline vigilance parameter. Two similar algorithms have been interpreted and implemented from the fusion ARTMAP architecture proposed by Asfour et al [1],[2]. One uses unsupervised Fuzzy ART [3] modules to form a compressed code within each channel before activation of a global recognition code. The other generates simultaneously the compressed, and the global, recognition codes. The latter algorithm seems more closely to resemble the fusion ARTMAP architecture as set out in [1],[2]. Simulations are shown and compared with fuzzy ARTMAP [5].

II. Fusion ARTMAP Neural Network.

Fusion ARTMAP is a self-organising neural network architecture for multi-channel or multi-sensor data fusion [1],[2]. That is, it is designed to classify input data using information from multiple independent channels or sensors regardless of its source or type. This network generalises the fuzzy ARTMAP architecture by incorporating an individual sensor classifier for each input channel and by extending the ARTMAP match tracking mechanism so that when a predictive error occurs, vigilance parameters in multiple ART modules are simultaneously increased until reset is triggered in one of them. This channel hereby is considered to be the most likely source of the predictive error and is defined as the channel with the minimum predictive confidence. This selective reset process, by assigning blame to one channel only, is considered to be a type of credit assignment that efficiently shares codes subsets across categories in the learned network (i.e. one intra-channel code can contribute to several global codes), because predictively effective channels are not unnecessarily reset to correct errors caused by ineffective channels [1], [2]. It is argued that this approach retains system predictive accuracy while reducing total network connectivity by maximising compression within each channel [1],[2].

The algorithms that are presented here are based on multiple-input sensors with single-teacher channel fusion ARTMAP architecture (figure 1) using equal baseline vigilance parameter for all the sensor channels and an independent one for the global channel.

A fusion ARTMAP architecture consists of two components: an individual sensor classifier and a global classifier. The former are fuzzy ART modules [3]. Each channel or fuzzy ART module serves as an individual classifier for each sensor, since each sensor is assigned to an individual channel. The latter includes a fuzzy ARTMAP module [5] that serves as a global classifier which makes global predictions using *compressed recognition codes* as input from each individual classifier. From figure 1, the module ART_c represents each one of the individual sensor classifier and the modules ART_a , ART_b and the Map Field F^{ab} form a fuzzy ARTMAP system that constitutes the global classifier. The fuzzy ARTMAP system internally controls code formation via a non-specific feedback signal sent in parallel to the fuzzy ART systems of the individual channels. This process is called *parallel match tracking* because it generalises the ARTMAP match tracking mechanism [1],[2].

In fusion ARTMAP, the ART_c module receives a stream $\{c\}$ of input patterns, ART_a receives a stream $\{a\}$ of compressed codes and ART_b receives a stream $\{b\}$ of input patterns, where b is the correct prediction given c , and a is the *compressed categorical input* formed by the compressed code from each channel, given c . At the start of each input presentation, the vigilance parameter, ρ , of each sensor channel and ART_a module equals a fixed baseline vigilance $\bar{\rho}_c$ and $\bar{\rho}_a$ respectively. When an input to ART_c makes an erroneous prediction of the actual ART_b input, the Map Field orienting subsystem becomes active. This mismatch, detected at the Map Field, between the ART_a category activated by the compressed categorical input a and the ART_b category activated by the input b , activates the parallel match tracking mechanism. This control strategy simultaneously raises the vigilance of the multiple ART_c modules (figure 2) until reset is triggered in just one of them. That module has the poorest match between the bottom-up input and the top-down prototype and is defined as the channel with the least predictive confidence [1],[2]. As a result, a search ensues in that module alone and a new category or compressed code is then chosen and used to replace the old contribution of this channel into the compressed categorical input to ART_a , preserving other input channel categories of the previously active pattern. Parallel match tracking continues until an active ART_a category satisfies both the ART_a matching criterion and the Map field matching criterion.

Owing to the sketchy information and the lack of mathematical and algorithmic details in [1] and [2], we had to clarify and define some of the terminology used for this architecture such as the compressed categorical input and the compressed recognition code as well as develop a strategy for the operation of the parallel match tracking mechanism that led us to some important modification. The following two sub-sections state the strategy and definitions commented here.

A. Parallel Match Tracking.

Let us define the minimum matching value γ as follow:

$$\gamma = \min_i \left(\frac{|C_i \wedge w_{L_i}^{c_i}|}{|C_i|} \right) \quad \text{for } i=1, \dots, n \quad (1)$$

Where

$w_{L_i}^{c_i}$ is the L th weight vector of the i th ART_{c_i} sensor channel, and

$C_i = (c_i, c_i^c)$ is the input vector in complement code form to the i th ART_{c_i} field $F_1^{c_i}$, and

n : number of sensor channels.

The channel with the minimum matching value has the poorest match between the bottom-up input and the top-down prototype and is defined as the channel with the least predictive confidence, ART_{c^*} .

Since all the ART_c modules have equal baseline vigilance parameter

$$\rho_{c_i} = \rho_{c_j} = \rho_c \quad \text{for } i \neq j \quad (2)$$

Parallel match tracking increases ρ_c in multiple ART_c modules until it is slightly larger than γ , that is

$$\rho_c^{(new)} = \min(\gamma + \delta, 1) \quad (3)$$

where δ is a positive bias small enough to reset the least confident channel.

Now, let us define the increment in confidence Δ as:

$$\Delta = \gamma - \rho_c^{(old)} + \delta \quad (4)$$

This equation represents the minimum amount by which the parallel match tracking mechanism raises ρ_c in order to reset the least confident channel and, in turn, represents the proportion that should be added to ρ_a to track the adjustment in ART_c . This leads to the following equation:

$$\rho_a^{(new)} = \min(\rho_a^{(old)} + \Delta, 1) \quad (5)$$

When $\gamma=1$ and a predictive error occurs, no more blame can be assigned to the ART_c module since $\gamma=1$ implies a perfect matching between the bottom-up input and the top-down prototype. Therefore, the fusion ARTMAP architecture behaves as the fuzzy ARTMAP architecture because the compressed categorical input to ART_a can not be anymore modified. Hence, equation (5) can be extended to give:

$$\rho_a^{(new)} = \begin{cases} \frac{|\mathbf{A} \wedge \mathbf{w}_j^a|}{|\mathbf{A}|} + \delta & \text{if } \gamma = 1 \\ \min(\rho_a^{(old)} + \Delta, 1) & \text{otherwise} \end{cases} \quad (6)$$

Note that, when the individual classifier and the global classifier have equal baseline vigilance parameter, equation(6) reduces to equation (3) for $\gamma \neq 1$.

In order to illustrate why we have used the minimum operation in equation (3), let us assume the following scenario: parallel match tracking has been activated and ART_a receives a compressed categorical input \mathbf{a} that is formed by a set of compressed codes arising from uncommitted nodes; then, an ART_a category may satisfy the ART_a match value criterion but fail to meet the parallel match tracking criterion which is the same as the ARTMAP match tracking criterion [4],[5]. In this situation ρ_c becomes one, since all match values from the sensor channels satisfy the following equation:

$$\frac{|C_i \wedge w_{L_i}^c|}{|C_i|} = \frac{|C_i \wedge 1|}{|C_i|} = 1 \quad (7)$$

In this case, no more blame can be assigned to the ART_c module since the prototype becomes the exemplar. Parallel match tracking cannot increase any more the value of ρ_c because this value is confined to the interval [0,1] [6]. There are two alternatives to solve this situation. One is as proposed in [4],[5] for the ARTMAP system, if no such node exists that satisfies both criteria then ART_a search leads to the shutdown of F_2^a for the remainder of the input presentation. In other words, we have to remove the present input pattern hence avoiding learning the active input presentation. This situation also arises from time to time in fusion ARTMAP according to the simulation done. The other is to set ρ_c to 1 as indicated in equation (3). This has the advantage of establishing a correct ART_a \rightarrow ART_b prediction by adjusting the value of ρ_a according to the ARTMAP match tracking mechanism given by equation (6) for $\gamma = 1$.

It is worth noting that, when a predictive error occurs and $\gamma \neq 1$, the j th chosen category at ART_a is not reset in the usual way as may be expected, i.e. ρ_a is increased just enough to ensure $\frac{|A \wedge w_j|}{|A|} \geq \rho_a$. Instead, the vigilance parameter ρ_a is increased by the minimum difference to reset the least confident channel, equation (6), as we do not impute this error to the compressed categorical input (input to ART_a) but to the compressed recognition code of the least confident channel. Two possibilities arise with this method:

- i. $\rho_a^{(new)} > \rho_a^{(old)}$, this is equivalent to resetting the j th chosen category in ART_a and finding a better F_2^a recognition code, or selecting an uncommitted node.
- ii. $\rho_a^{(new)} < \rho_a^{(old)}$, unnecessary nodes are not reset.

To illustrate how the parallel match tracking mechanism works, let us describe a cycle of ρ_a and ρ_c adjustment. When ART_a makes a prediction that is incompatible with the actual ART_b input, The MAP field orienting subsystem becomes active. This mismatch event activates the parallel match tracking mechanism. In fusion ARTMAP a mismatch at F^{ab} while

F_2^b is active triggers an inter-ART reset signal to the ART_a and the ART_c orienting subsystems, identical to [4], [5]. This occurs whenever

$$|\mathbf{x}^{ab}| < \rho_{ab} |\mathbf{y}^b| \quad (8)$$

where

\mathbf{x}^{ab} denotes the F_{ab} output vector,

\mathbf{y}^b denotes the F_2^b output vector, and

ρ_{ab} denotes the Map Field vigilance parameter.

At the start of each input presentation, ρ_c and ρ_a equal a fixed baseline vigilance $\bar{\rho}_c$ and $\bar{\rho}_a$ respectively. When an input c_i activates an $F_2^{c_i}$ category node, L_i , and resonance is established at each one of the ART_c channels according to [4], [5],

$$|\mathbf{x}^{c_i}| = |\mathbf{C}_i \wedge \mathbf{w}_{L_i}^{c_i}| \geq \rho_c |\mathbf{C}_i| \quad \text{for } i=1, \dots, n \quad (9)$$

Then, ART_a receives the compressed categorical input, \mathbf{a} , from each channel separately. When this input \mathbf{a} activates an F_2^a category node, J , and resonance is established at the ART_a module,

$$|\mathbf{x}^a| = |\mathbf{A} \wedge \mathbf{w}_J| \geq \rho_a |\mathbf{A}| \quad (10)$$

as in equation (9). If the ART_b category predicted by \mathbf{a} fails to match the active ART_b category by equation (8), an inter-ART reset signal is sent to the multiple ART modules. The inter-ART reset signal simultaneously raises the vigilance of the multiple ART modules to a value that is just enough to cause equation (9) to fail in just one of them and the new matching values for ρ_c and ρ_a are adjusted by equation (3) and (6) respectively. Node L of the ART_c channel is therefore reset and a search ensues in that module alone. Parallel match tracking continues until an active ART_a category satisfy both the ART_a matching criterion equation (10) and the map field matching criterion equation (8).

B. Compressed Recognition Code.

In the adaptive resonance theory, the compressed recognition code is referred to as the category or symbol to represent input \mathbf{I} at level F_2 [3]. In this context, field F_1 acts to match the learned expectation \mathbf{V} against the active input vector \mathbf{I} . The result of this matching process (once a resonant state has been reached) is the re-constructed feature vector \mathbf{X}^* across field F_2 that encodes the pattern of features relevant to \mathbf{I} and determines the future course of learning and recognition by the network [6]. Thus, the compressed recognition code that is used as input to ART_a depends on the selection of the learning rule. ART systems use the following learning rule [3]:

$$\mathbf{w}_J^{(new)} = \beta(\mathbf{I} \wedge \mathbf{w}_J^{(old)}) + (1 - \beta)\mathbf{w}_J^{(old)} \quad (11)$$

where fast learning corresponds to setting $\beta = 1$. For the fast-commit slow-recode option, $\mathbf{w}_J^{(new)} = \mathbf{I}$ the first time category J becomes committed, otherwise equation (3) with $\beta < 1$ is used.

The compressed categorical input vector \mathbf{a} is defined by:

$$\mathbf{a} = \mathbf{z}^{c_1} \cup \mathbf{z}^{c_2} \cup \dots \cup \mathbf{z}^{c_n} \quad (12)$$

where

\mathbf{z}^{c_i} is the compressed recognition code vector of the i th ART_{c_i} sensor channel,

n : number of sensor channels, and

\cup : union operator.

and

\mathbf{z}^{c_i} obeys equation (11), so that

$$\mathbf{z}^{c_i} = \beta(\mathbf{I}_i \wedge \mathbf{w}_{L_i}^{c_i(oid)}) + (1 - \beta)\mathbf{w}_{L_i}^{c_i(oid)} \quad (13)$$

where

$\mathbf{w}_{L_i}^{c_i(oid)}$ is the L th weight vector of the i th ART_{c_i} sensor channel,

β denotes the learning rate parameter, and

$\mathbf{I}_i = \mathbf{C}_i = (\mathbf{c}_i, \mathbf{c}_i^c)$ is the input vector in complement code form to the i th ART $_{c_i}$ field $F_1^{c_i}$.

In the case of fast learning ($\beta=1$), the compressed recognition code \mathbf{z}^{c_i} , equals the activity vector \mathbf{x}^{c_i} , since

$$\mathbf{z}^{c_i} = (\mathbf{I}_i \wedge \mathbf{w}_{L_i}^{c_i(\text{old})}) = \mathbf{x}^{c_i} \quad (14)$$

III. Fusion ARTMAP Algorithm

Two algorithms designed to reproduce the functionality of that proposed in [1] and [2] have been implemented. The main difference between the two versions is that, initially, in one of them, the sensor channels are allowed to self-organise into category codes in response to a stream $\{\mathbf{c}\}$ of input patterns as if they were disconnected from the rest of the system, thus during this phase, each channel forms an unsupervised fuzzy ART classifier. Therefore, during training the input pattern set is presented twice. The first time, to form a set of compressed codes for each channel separately and then to begin the association between the compressed code of ART $_a$ and the compressed code of exemplar \mathbf{b} in order to make a global prediction. Incidentally, such an approach is inherently “off-line” and thus diminishes the utility of fusion ARTMAP from the perspective of “on-line” operation in a changing environment. In the discussion below, this version is referred to as the “two-pass” algorithm.

In the second version the input pattern set is presented just once. As an input \mathbf{c} arrives at each sensor channel, a compressed recognition code is created in each channel separately. These codes are merged into a single vector \mathbf{a} and sent as input to the ART $_a$ module. Then, the fuzzy ARTMAP system organises the multi-channel recognition code via the parallel match tracking mechanism. In the discussion below, this version is referred to as the “one-pass” algorithm. This retains the option of on-line operation and thus remains true to the spirit of Adaptive Resonance Theory.

Both algorithms use complement coding since this prevents the problem of category proliferation as too many adaptive weights converge to zero [3]. This pre-processing method is used at the ART_c field, F_0^c , for the input pattern \mathbf{c} , at the ART_b field, F_0^b , for the input pattern \mathbf{b} and at the ART_a field, F_0^a , for the input pattern \mathbf{a} . Also any other method that normalises the inputs at a pre-processing stage can be used to avoid this problem of category proliferation; i.e. performing fuzzification of the input domain using symmetric membership functions.

A. The One-Pass Algorithm.

At the start of each input presentation, $\rho_c = \bar{\rho}_c$ and $\rho_a = \bar{\rho}_a$.

1. Present an input \mathbf{c} to each ART_c sensor channel.
2. When an input \mathbf{c} activates an F_2^c category node L and resonance is established, a compressed recognition code, \mathbf{z}^c , is generated in each channel given by equation (13).
3. Form the compressed categorical input, \mathbf{a} , to ART_a given by equation (12).
4. Present an input \mathbf{b} to the ART_b module.
5. Search for an F_2^b category node, K , such that resonance can be established in the ART_b module.
6. Search for an F_2^a category node, J , such that resonance can be established in the ART_a module.
7. Test if ART_a predicts the same category as ART_b.

If equation (8) is true then

the prediction \mathbf{w}_J^{ab} is disconfirmed by \mathbf{y}^b and parallel match tracking is activated.

Else

go to step 12.

8. Parallel match tracking raises the vigilance of multiple sensors in ART_c , so that the new baseline vigilance values are given by equation (3) and (6).

9. Reset node L of the least confident channel, ART_{c^*} , and a search ensues in that module alone, leading to activation of another F_2^c node. Once a resonant state has been established according to equation (9), a new compressed recognition code, $z^{c(new)}$, is generated by equation (13).

10. Replace the previously compressed recognition code, $z^{c(old)}$, of ART_{c^*} with a new $z^{c(new)}$ code in the compressed categorical input \mathbf{a} , preserving other input channel categories of the previously active pattern.

11. Go to step 6.

12. Learning proceeds updating the connection weights to the winning nodes J, K and L at the F_2 layer according to the chosen learning rule, and updating the Map Field weights w_{jk}^{ab} in $F_2^a \rightarrow F_2^b$ paths.

13. Remove input vector \mathbf{c} and \mathbf{b} . Return to step 1 with a new input vector.

B. The Two-Pass Algorithm.

1. The ART_c module is disconnected from the rest of the fusion ARTMAP system so that it behaves as a fuzzy ART system with ρ_c equal to the chosen baseline vigilance, $\bar{\rho}_c$. Stable recognition categories for each individual ART_c channel are established in response to a stream $\{\mathbf{c}\}$ of input patterns.

2. Repeat all steps of the one-pass algorithm.

IV. Simulations.

A. Circle-in-the-Square

The circle in the square problem requires a system to identify which points of a square lie inside, and which lie outside, a circle whose area equals half that of the square [5].

A single-channel Fusion ARTMAP system was trained to recognise whether a point within a unit square was inside or outside a circle of one-half unit area. The results were compared with those using fuzzy ARTMAP. The main goal of this task was to evaluate the predictive accuracy of the one-pass and two-pass algorithms and to make comparisons with fuzzy ARTMAP since during supervised learning of single-channel signal and teaching input, fusion ARTMAP is functionally equivalent to fuzzy ARTMAP for analogue inputs [1]. According to the same simulations described in [1], the performance of the two systems was identical and fusion ARTMAP produced more ART_{ab} category nodes than did fuzzy ARTMAP at F^{ab} . However, no simulation results are shown in either any setting for the network parameters.

•Simulation A1

Fusion ARTMAP was trained on a data set size of 1000 input patterns. Inputs in complement coding form, equal baseline vigilance parameters for the ART_c and the ART_a modules, and single and multiple training epoch were used. Both the training and test sets consist of 1000 input patterns. For this simulation, the choice parameter was set to 0.001; i.e. $\alpha = 0.001$, fast learning was used; i.e. $\beta = 1$, and different values for the baseline vigilance parameter were used. With fast learning at the Map Field, its vigilance parameter, ρ_{ab} , can be set to any value between 0 and 1 without affecting fast-learn results. Fuzzy ARTMAP was trained using the same data set and network parameters.

Typical results are shown in table I. In general, we can observe similar accuracy between the two architectures. Note that, for $\bar{\rho} = 0$, the two-pass algorithm had a better accuracy than the others. However, fuzzy ARTMAP was better in general. Overall, we cannot

generalise, since the training process depends on the statistical nature and the size of the training sample. This simulation shows that the one-pass and two-pass algorithms are good candidates to represent the fusion ARTMAP architecture proposed in [1],[2].

•Simulation A2

Fusion ARTMAP and fuzzy ARTMAP were trained on the same data set and network parameters used in simulation A1 but using fuzzification of inputs. Since a priori information about the inputs is available, network training accuracy can be significantly improved.

Typical results are shown in table II. Note the effect of the fuzzification of inputs on the improvement on the test accuracy and the significant reduction of the ART_{ab} category nodes at F^{ab} as $\bar{\rho}$ is increased (# F_2^a nodes at table II) in the one-pass and two-pass algorithms. For example, when comparing table I and II for $\bar{\rho} = 0.7$ and three training epochs, the test accuracy for the one-pass algorithm is increased from 95% to 97.4% (42 ART_{ab} category nodes) while the test accuracy for fuzzy ARTMAP is increased from 95.4% to 97.4% (65 ART_{ab} category nodes). Fuzzy ARTMAP created more codes because the dimension of the input vector was increased, leading to the formation of more category nodes at field F_2^a .

•Simulation A3

Fusion ARTMAP and fuzzy ARTMAP were trained on the same data set used in simulation A1 and A2 using fuzzification of inputs as for simulation A2. For fusion ARTMAP, the vigilance parameter of the ART_c was set to 0.7 while the matching value for the ART_a module was varied from 0 to 0.9. The other network parameters were left unchanged.

Table III summarises the results. Note the beneficial effect in terms of the reduction of ART_{ab} category nodes at F^{ab} when $\rho_a < \rho_c$. When $\rho_a > \rho_c$, the number of connections at ART_{ab} is increased owing to the stricter matching condition at the global classifier that leads to the creation of more category nodes.

B. Cylinder-in-the-cube

The cylinder in the cube problem generalises the idea of the circle in the square to a three-dimensional problem. In principle, It requires a system to identify which points of a cube lie inside, and which lie outside, a cylinder whose volume equals half that of the cube.

The motivation for developing this task was to confirm the ability of fusion ARTMAP to classify objects using information from multiple data sources of any type [1], [2].

This task can be seen as a requirement for a system to identify a cylinder of a particular size or volume; i.e., a cylinder whose volume is or is not equal to half the volume of the cube. In terms of this simulation, we can think of having two sensors to find the volume of a cylinder and allocating two channels of the fusion ARTMAP system to these sensors. Thus, one channel receives information from an area-sensor (square units) where it is first classified into area codes while the other gathers information from a distance-sensor (linear units) where it is classified into length codes. The compressed area and length codes become inputs to a global classifier which predicts the type of cylinder; i.e. appropriate size or inappropriate size. The program that implemented this task was set up to generate two classes with approximately the same probability.

• Simulation B1

A two-channel fusion ARTMAP system was trained to recognise whether a point within a unit cube was inside or outside a cylinder of one-half unit volume. Thus, the cylinder is identified implicitly. Inputs in complement coded form, equal baseline vigilance parameter for the ART_c and the ART_a modules, and single and multiple training epoch were used. Both the training and test sets consisted of 2000 input patterns each. For this simulation, the choice parameter was set to 0.001; i.e. $\alpha = 0.001$, fast learning was used; i.e. $\beta = 1$, and different values for the baseline vigilance parameter were used. With fast learning at the Map Field, its vigilance parameter, ρ_{ab} , can be set to any value between 0 and 1 without affecting fast-learn

results. The fuzzy ARTMAP system was trained on the same data set and network parameters but using a concatenated input vector.

Typical results are shown in table IV. Note that, the one-pass and two-pass algorithms had lower accuracy on the testing set than that of fuzzy ARTMAP. Also note the difference between what the network learned and what it predicted. Taking the ratio of the test set accuracy to the training set accuracy gives us an indication of how good the performance of the network is for the purpose of this analysis. For example, for one training epoch and $\bar{\rho} = 0$, the one-pass gives 0.9351, two-pass: 0.9408 and fuzzy ARTMAP: 0.9545 which are approximately equal. Also this ratio decreases as $\bar{\rho}$ increases. For one training epoch and $\bar{\rho} = 0.7$, the one-pass gives 0.9270, two-pass: 0.9214 and fuzzy ARTMAP: 0.9382 which is not unfavourable. However, this performance appears to be due to the way that the fusion ARTMAP system learns. As remembered, fusion ARTMAP learns prototype from prototype that has been previously learned in the ART_c module. This is a consequence of the way ART systems learn. ART systems learn prototypes rather than exemplars, because the reconstructed feature vector \mathbf{X}^* , rather than \mathbf{I} itself, is learned [3]. Thus, the category prototype in the field F_2^a of ART_a is in fact the *prototype of a cluster of prototypes*. Therefore, if we have good cluster formation at field F_2^c that represents well enough the universe of the sample pattern, we can improve the compressed code formation at the global classifier which will be better able to predict the actual ART_b input. For better accuracy, we should increase the training set size, or tune the network more appropriately (i.e. a combination of high vigilance threshold at the ART_c module and low vigilance threshold at the global classifier), implement another method such as a voting strategy suitable for small or incomplete training sets [5], or use any a priori knowledge about the inputs; i.e. fuzzification of inputs. The results of using these alternatives are illustrated in the simulations below.

Note that, the number of ART_{ab} category nodes is larger than that generated by fuzzy ARTMAP. This is because of the stricter matching criterion used at both levels (see last row results of table IV where different values of vigilance parameters between the two classifiers were used). Setting $\rho_a < \rho_c$ results in a good combination since we can have more prototypes

at F_2^c , hence better cluster formation for the input c that, in turn, represents the future input to the global classifier, and good accuracy due to the capability of the network to generalise at the global classifier, hence fewer connection paths at ART_{ab} . As a result, more codes are created in field F_2^c with fewer paths and weights in F^{ab} .

• Simulation B2

Fusion ARTMAP and fuzzy ARTMAP were trained on the same data set and network parameters used in simulation B1 but using fuzzification of inputs. Simulation results are summarised in table V.

Note the improvement in the test accuracy for $\bar{p} = 0.7$ with one and three training epochs, where the one-pass algorithm had better accuracy than the others. Also note that, at least one more training epoch is needed to reach 100% in the training set which implies that a few more categories would be established and a probable improvement in the test accuracy would be achieved. Also, the number of ART_{ab} category nodes is reduced by a ratio of 1.4 when comparing the one-pass algorithm with fusion ARTMAP. Once again, a priori information about the inputs improves the performance of the fusion ARTMAP system.

Simulation B3

Fusion ARTMAP and fuzzy ARTMAP were trained on data sets ranging in size from 125 to 2000 input patterns for a single epoch. A fixed vigilance value of 0.7 and the same network parameters as for simulation B1 were used. The test set was as the same size as the training set.

Table VI shows the evolution of test set errors as the training set is increased in size. This shows how test set error is reduced from 33.6% to 19.95% as training size increases from 125 to 2000 for the one-pass algorithm. Note also that, larger training sets are required to achieved a correct prediction rate of over 80% for the one-pass algorithm. This simulation

shows that a large training set size is needed for fusion ARTMAP to achieve good results on-line.

The Two-pass algorithm behaves erratically. This appears to be the combination of the earlier unsupervised and later supervised learning of the two-pass algorithm.

•Simulation B4

Table VII summarises simulation results that repeat the training conditions of simulation B3 except that fuzzification of inputs was used. This table shows the evolution of test set errors as the training set is increased in size. This shows how a test set error is reduced from 26.4% to 8.95% as training set size increases from 125 to 2000 for the one-pass algorithm. Better accuracy is achieved as the training set size is increased.

Note also, the reduction in ART_{ab} category nodes, compared with fuzzy ARTMAP, as the training set size is increased in size. This reduction is increased from a ratio of 1.36 to 1.88 as training set size is increased from 125 to 2000 for the one-pass algorithm. Further reductions can be achieved if the baseline vigilance between the individual and global classifiers is selected as indicated in simulation B1; i.e. $\rho_a < \rho_c$.

•Simulation B5

Fusion ARTMAP and fuzzy ARTMAP were trained on a data set of 250 patterns and with one training epoch. Their baseline vigilance parameters were set to 0 and the other network parameters as for the simulation B1. They were run for five independent simulations. Each one had different randomly chosen presentation orders for the 250 training patterns. The test set was as the same size as the training set.

Typical results are shown in table VIII. For the one-pass algorithm, the test set error varied from 31.1% to 17.6% and for the fuzzy ARTMAP, from 13.6% to 24.8%. This shows that fusion ARTMAP is more sensitive than fuzzy ARTMAP to training on small sets. This table also shows that, the sampling of the training set or, as the first run, good luck in the selection of representative samples, can dramatically alter early success rates. Observe that,

training on small sets illustrates the statistical nature of the learning or coding process; i.e. from 39 to 20 ART_{ab} category nodes for one-pass, from 30 to 16 to two-pass, and from 11 to 21 for fusion ARTMAP.

•Simulation B6

Fusion ARTMAP was trained on a data set of 1000 input patterns using fuzzification of inputs. The vigilance parameter of the ART_c was set to 0.7 while the matching value for the ART_a module was varied from 0 to 0.8. The rest of the network parameters were set as for the simulation B1. Fuzzy ARTMAP was trained on the same conditions but fixing the matching value to 0.7. Thus, it is used as a reference point when comparing with fusion ARTMAP using different matching values between ART_a and ART_c. The test set was the same size as the training set.

Table IX summarises the results. Note again, the beneficial effect in terms of the reduction of ART_{ab} category nodes at F^{ab} when $\rho_a < \rho_c$. When $\rho_a > \rho_c$, the numbers of connections at ART_{ab} is increased owing to the stricter matching condition at the global classifier that leads to the creation of more category nodes. Note that, when using $\rho_a = 0.4$ and $\rho_c = 0.7$, for the one-pass algorithm, a reduction of ART_{ab} category nodes of 4.43 is achieved as well as a good test accuracy of 88.1% (c.f. 88.8% of fuzzy ARTMAP), bearing in mind that, there are still patterns remaining to be learned (training accuracy of 93% compared with 99.8% of fuzzy ARTMAP).

This once again shows that, a good selection for the vigilance parameters of the ART_a and ART_c modules of fusion ARTMAP plays an important role in the accuracy and in the number of compressed codes that can be created at the global classifier.

•Simulation B7

The one-pass algorithm was trained on a data set of 250 input patterns using one-training epoch. Its baseline vigilance parameter was set to 0 and the other network parameters were set as for the simulation B1. The one-pass algorithm was run for three and five

independent simulations, each with a different input order. The aim of this simulations was to apply the voting strategy proposed in [5]. The test set was the same size as the training set.

Table X shows how voting improves performance of the one-pass algorithm with one-training epoch. In the two cases, voting performance was better than the performance of any of the individual simulations. For example, applying voting for five runs caused the error rate to drop to 11.2% from a five-run average of 23.52%.

C. Summary of experimental results and observations.

- Simulation results confirm that fusion ARTMAP is able to classify objects using information from different sources of disparate type.

- Large training sets are needed to train fusion ARTMAP on-line. This is because Fusion ARTMAP codes prototypes using prototypes that have been previously coded at the ART_c module. Therefore, it is very important to have a good representation of cluster of patterns at field F_2^c . Otherwise, this lack of categorisation at field F_2^c would propagate to the global classifier resulting in a detriment in the accuracy.

- Simulation results show that, if previous knowledge of the inputs is available, network training accuracy can be significantly improved. A voting strategy can also be used in fusion ARTMAP to improve the performance of the network when small or incomplete training sets are given.

- Simulation results show that, the threshold for ρ_a should be lower than that for ρ_c . This configuration allows the creation of reliable information codes for each sensor channel, resulting in better global prediction. In turn, increased code compression can be obtained at the global classifier owing to the smaller matching value used at this level. As noted in [1],[2], fusion ARTMAP created the necessary numbers of codes for each channel and then the global classifier organised them in such a manner that one intra-channel code can contribute to several global codes. Otherwise, we would have a kind of probabilistic combination that in the worst case would be:

$$\#F_2^a \text{ node} = \prod_{i=1}^n \#F_2^{c_i} \text{ nodes, where } n: \text{ number of sensor channels}$$

Therefore, more codes are created in F_2^c with fewer codes in F_2^a and hence fewer paths in F^{ab} .

•Simulation results show that the one-pass and two-pass algorithms are suitable candidates to represent the fusion ARTMAP architecture proposed in [1],[2]. The one-pass algorithm seems to have the more similar behaviour to that of fuzzy ARTMAP in terms of accuracy and the monotonic decrease in test set errors as the training set is increased in size. On the basis of this propositions and the simulation results, we can say that the predictive accuracy of the one-pass algorithm is retained compared with that of fuzzy ARTMAP as well as a neural network architecture than can be used for multi-channel, or multi-sensor, data fusion.

V. Conclusions

Two algorithms have been implemented that follow the operational philosophy described in [1] and [2]. The case of the multiple-input sensors with single-teacher channel fusion ARTMAP architecture with individual and global classifiers using independent baseline vigilance parameter is explored, achieving a better understanding of this system through the development of these two algorithms and the description of some of their operations. Owing to the lack of mathematical and algorithmic details in [1] and [2], a number of details required derivation before successful implementation could be achieved. Among this were:

- i. Definition of the compressed recognition code, \mathbf{z}^f .
- ii. Definition of the compressed categorical input \mathbf{a} to ART_a .
- iii. Modification to the parallel matching criterion.
- iv. Development of a control strategy for using these have been successfully implemented leading to the following results:
 - a. Two algorithms have been developed with acceptable accuracy compared with that of fuzzy ARTMAP. However, the one-pass algorithm seems to be a better candidate than the two-pass algorithm to represent the fusion ARTMAP architecture.

b. Directions for tuning the network in terms of the selection of the baseline matching values are given, ($\rho_a < \rho_c$). The importance, during the training process, of the sample size and some ways to overcome this limitation; i.e. using voting strategy if sample size is incomplete or incorporating a priori information about the inputs if it is available, are noted.

At this point we can conclude that, fusion ARTMAP is designed to obtain a compressed code using information (compressed recognition codes) from multiple data sources of any type, while retaining system predictive accuracy [2]. As asserted in [1], fusion ARTMAP creates more parsimonious codes (in field F_2^c) with fewer paths and weights (in F^{ab}), than would be needed by the single-channel, fuzzy ARTMAP, recognition system.

References

1. Y.R. Asfour, G.A. Carpenter, S. Grossberg, and G. Leshner, 1993, "Fusion ARTMAP: A neural network architecture for multi-channel data fusion and classification", *Technical report CAS/CNS-93-006*. Boston, MA: Boston University.
2. Y.R. Asfour, G.A. Carpenter, S. Grossberg, and G. Leshner, 1993, "Fusion ARTMAP: An adaptive fuzzy network for multi-channel classification", *Technical report CAS/CNS-93-052*. Boston, MA: Boston University.
3. G.A. Carpenter, S. Grossberg, and D.B. Rosen, 1991, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system", *Neural Networks*, **4**, 759-771.
4. G.A. Carpenter, S. Grossberg, and J.H. Reynolds, 1991, "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organising neural network", *Neural Networks*, **4**, 565-588.
5. G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, and D.B. Rosen, 1992, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps", *IEEE Transactions on Neural Networks*, **3**, 698-713.
6. G.A. Carpenter, S. Grossberg, 1987, "A massively parallel architecture for a self-organising neural network pattern recognition machine", *Computer Vision, Graphics, and Image processing*, **37**, 54-115

TABLE I

Comparison of system performance after training for circle-in-the-square using complement coding as input pre-processing. Training and test set size: 1000.

| No. of Categories | One-Pass Accuracy (%) | | $\bar{\rho}$ | No. of Categories | Two-Pass Accuracy (%) | | $\bar{\rho}$ | No. of Categories | Fuzzy ARTMAP Accuracy (%) | | $\bar{\rho}$ | Note |
|-------------------|-----------------------|------|--------------|-------------------|-----------------------|------|--------------|-------------------|---------------------------|------|--------------|---------|
| | Training | Test | | | Training | Test | | | Training | Test | | |
| a: 14 c: 18 | 95.9 | 92.5 | 0 | a: 18 c: 22 | 97.1 | 94.6 | 0 | 19 | 96.4 | 92.8 | 0 | 1 epoch |
| a: 17 c: 24 | 100 | 93.6 | 0 | a: 21 c: 27 | 99.6 | 95.6 | 0 | 24 | 100 | 95.4 | 0 | 3 epoch |
| a: 23 c: 32 | 94.5 | 90.1 | 0.5 | a: 18 c: 27 | 97 | 93.6 | 0.5 | 16 | 97.3 | 94.8 | 0.5 | 1 epoch |
| a: 26 c: 36 | 100 | 94.2 | 0.5 | a: 23 c: 33 | 100 | 95.8 | 0.5 | 21 | 100 | 96.1 | 0.5 | 3 epoch |
| a: 25 c: 31 | 95.8 | 93.3 | 0.7 | a: 23 c: 46 | 96.9 | 92.9 | 0.7 | 30 | 97.2 | 95 | 0.7 | 1 epoch |
| a: 31 c: 41 | 100 | 95 | 0.7 | a: 28 c: 52 | 100 | 94.4 | 0.7 | 36 | 99.6 | 95.4 | 0.7 | 3 epoch |

TABLE II

Comparison of system performance after training for circle-in-the-square using fuzzification as input pre-processing. Training and test set size: 1000.

| No. of Categories | One-Pass Accuracy (%) | | $\bar{\rho}$ | No. of Categories | Two-Pass Accuracy (%) | | $\bar{\rho}$ | No. of Categories | Fuzzy ARTMAP Accuracy (%) | | $\bar{\rho}$ | Note |
|-------------------|-----------------------|------|--------------|-------------------|-----------------------|------|--------------|-------------------|---------------------------|------|--------------|---------|
| | Training | Test | | | Training | Test | | | Training | Test | | |
| a: 20 c: 27 | 97.4 | 94.4 | 0 | a: 13 c: 23 | 97.8 | 96.3 | 0 | 18 | 97.7 | 95.2 | 0 | 1 epoch |
| a: 23 c: 24 | 100 | 95.3 | 0 | a: 16 c: 26 | 99.8 | 96.5 | 0 | 30 | 99.3 | 93.9 | 0 | 3 epoch |
| a: 40 c: 72 | 98.5 | 96.8 | 0.7 | a: 29 c: 75 | 98.5 | 94.9 | 0.7 | 64 | 99.7 | 97.1 | 0.7 | 1 epoch |
| a: 42 c: 77 | 100 | 97.4 | 0.7 | a: 31 c: 78 | 100 | 95.4 | 0.7 | 65 | 100 | 97.4 | 0.7 | 3 epoch |

TABLE III

Comparison of system performance after one training epoch for circle-in-the-square using different baseline matching parameters between ARTc and ARTa and fuzzification as input pre-processing. Training and testing set size: 1000.

| No. of Categories | One-Pass | | | No. of Categories | Two-Pass | | | Fuzzy ARTMAP | | | | |
|-------------------|-----------------------|-------------------|----------------|-------------------|-----------------------|-------------------|----------------|-------------------|-----------------------|-------------------|--------------|-----|
| | Accuracy (%) training | Accuracy (%) Test | $\bar{\rho}$ | | Accuracy (%) Training | Accuracy (%) Test | $\bar{\rho}$ | No. of Categories | Accuracy (%) Training | Accuracy (%) Test | $\bar{\rho}$ | |
| a: 16 c1:102 | 93.3 | 92.4 | a:0 c:0.7 | a: 13 c1: 102 | 98.1 | 94.3 | a:0 c:0.7 | | | | | N/A |
| a: 18 c1: 74 | 97.7 | 94.1 | a:0.4 c:0.7 | a: 14 c1: 85 | 98 | 94.2 | a:0.4 c:0.7 | 64 | 99.7 | 97.1 | 0.7 | |
| a: 40 c1: 72 | 98.5 | 96.8 | a:0.7 c:0.7 | a: 29 c1: 75 | 98.5 | 94.9 | a:0.7 c:0.7 | | | | | N/A |
| a: 98 c1: 67 | 95.7 | 95.4 | a:0.9 c:0.7 | a: 64 c1: 76 | 97.7 | 94.3 | a:0.9 c:0.7 | | | | | N/A |

TABLE IV

Comparison of system performance after training for cylinder-in-the-cube using complement coding as input pre-processing. Training and test set size: 2000.

| No. of Categories | One-Pass | | | No. of Categories | Two-Pass | | | Fuzzy ARTMAP | | | | Note |
|----------------------------|-----------------------|-------------------|----------------|----------------------------|-----------------------|-------------------|----------------|-------------------|-----------------------|-------------------|--------------|---------|
| | Accuracy (%) training | Accuracy (%) Test | $\bar{\rho}$ | | Accuracy (%) Training | Accuracy (%) Test | $\bar{\rho}$ | No. of Categories | Accuracy (%) Training | Accuracy (%) Test | $\bar{\rho}$ | |
| a: 91 c1: 82 c2:33 | 87.1 | 81.45 | 0 | a: 77 c1: 69 c2:28 | 88.65 | 83.4 | 0 | 26 | 95 | 90.65 | 0 | 1 epoch |
| a: 168 c1:135 c2:51 | 95.55 | 86.4 | 0 | a: 154 c1: 130 c2:40 | 94.3 | 85.5 | 0 | 37 | 99.6 | 92.6 | 0 | 3 epoch |
| a: 136 c1: 99 c2:37 | 86.35 | 80.05 | 0.7 | a: 112 c1: 95 c2:33 | 92.9 | 85.6 | 0.7 | 113 | 95.5 | 89.6 | 0.7 | 1 epoch |
| a: 253 c1: 183 c2:57 | 96 | 85.4 | 0.7 | a: 203 c1: 153 c2:55 | 95.3 | 85.9 | 0.7 | 124 | 99.9 | 91.9 | 0.7 | 3 epoch |
| a: 18 c1: 187 c2:140 | 99.55 | 85.45 | a: 0 c: 0.7 | a: 19 c1: 216 c2:165 | 99.5 | 88.25 | a: 0 c: 0.7 | | | | | 3 epoch |

TABLE V

Comparison of system performance after training for cylinder-in-the-cube using fuzzification as input pre-processing. Training and test set size: 2000.

| No. of Categories | One-Pass | | | No. of Categories | Two-Pass | | | No. of Categories | Fuzzy ARTMAP | | | Note |
|----------------------------|----------|--------------|--------------|----------------------------|----------|--------------|--------------|-------------------|--------------|--------------|--------------|---------|
| | training | Accuracy (%) | $\bar{\rho}$ | | Training | Accuracy (%) | $\bar{\rho}$ | | Training | Accuracy (%) | $\bar{\rho}$ | |
| a: 74 c1: 88 c2:34 | 92.4 | 87.2 | 0 | a: 38 c1: 52 c2:24 | 92.4 | 89.4 | 0 | 25 | 93.75 | 92.3 | 0 | 1 epoch |
| a: 143 c1:165 c2:59 | 98.55 | 88.95 | 0 | a: 110 c1: 121 c2:47 | 98.3 | 91.3 | 0 | 58 | 99.2 | 93.45 | 0 | 3 epoch |
| a: 136 c1: 122 c2:39 | 95.9 | 91.05 | 0.7 | a: 152 c1: 148 c2:44 | 92.7 | 85.25 | 0.7 | 256 | 98.9 | 90.35 | 0.7 | 1 epoch |
| a: 186 c1: 163 c2:52 | 98.7 | 91.75 | 0.7 | a: 237 c1: 214 c2:69 | 98.95 | 87.7 | 0.7 | 261 | 100 | 90.65 | 0.7 | 3 epoch |

TABLE VI

Comparison of system performance after one training epoch on input sets ranging in size from 125 to 2000 for cylinder-in-the-cube task using complement coding as input pre-processing. $\rho=0.7$.

| Training set size | One-Pass | | | Two-Pass | | | Fuzzy ARTMAP | | |
|-------------------|-------------------|---------------------------------|-------|-------------------|---------------------------------|-------|-------------------|---------------------------------|------|
| | No. of Categories | % test set correct incorrect | | No. of Categories | % test set correct incorrect | | No. of Categories | % test set correct incorrect | |
| 125 | a:25 | 66.4 | 33.6 | a: 27 | 76 | 24 | 26 | 71.2 | 28.8 |
| | c1:20 | | | c1: 29 | | | | | |
| | c2:10 | | | c2:14 | | | | | |
| 250 | a:45 | 66.8 | 33.2 | a: 36 | 69.2 | 30.8 | 44 | 73.6 | 26.4 |
| | c1:33 | | | c1: 36 | | | | | |
| | c2:16 | | | c2:16 | | | | | |
| 500 | a:61 | 68.4 | 31.6 | a: 50 | 80 | 20 | 64 | 84 | 16 |
| | c1:44 | | | c1: 44 | | | | | |
| | c2:20 | | | c2:17 | | | | | |
| 750 | a:80 | 72.53 | 27.47 | a: 57 | 78.67 | 21.33 | 75 | 84.8 | 15.2 |
| | c1:58 | | | c1: 50 | | | | | |
| | c2:23 | | | c2:18 | | | | | |
| 1000 | a: 95 | 75.9 | 24.1 | a: 81 | 78 | 22 | 84 | 87.5 | 12.2 |
| | c1: 70 | | | c1: 67 | | | | | |
| | c2:26 | | | c2:25 | | | | | |
| 2000 | a: 136 | 80.05 | 19.95 | a: 112 | 85.6 | 14.4 | 113 | 89.6 | 10.4 |
| | c1: 99 | | | c1: 95 | | | | | |
| | c2:37 | | | c2:33 | | | | | |

TABLE VII

Comparison of system performance after one training epoch on input sets ranging in size from 125 to 2000 for cylinder-in-the-cube task using fuzzification as input pre-processing. $\rho=0.7$.

| Trainin g set size | One-Pass | | | Two-Pass | | | Fuzzy ARTMAP | | |
|--------------------------|---------------------------|---------------------------------|------|---------------------------|---------------------------------|-------|---------------------------|---------------------------------|------|
| | No. of Catego- ries | % test set correct incorrect | | No. of Catego- ries | % test set correct incorrect | | No. of Catego- ries | % test set correct incorrect | |
| 125 | a:36 | 73.6 | 26.4 | a: 32 | 68.8 | 31.2 | 49 | 64.8 | 35.2 |
| | c1:36 | | | c1: 47 | | | | | |
| | c2:19 | | | c2:18 | | | | | |
| 250 | a:56 | 76.8 | 23.2 | a: 43 | 70.4 | 29.2 | 81 | 70.8 | 29.2 |
| | c1:56 | | | c1: 62 | | | | | |
| | c2:26 | | | c2:21 | | | | | |
| 500 | a:82 | 79.8 | 20.2 | a: 70 | 76.2 | 23.8 | 125 | 80.2 | 19.8 |
| | c1:79 | | | c1: 96 | | | | | |
| | c2:31 | | | c2:30 | | | | | |
| 750 | a:98 | 83.6 | 16.4 | a: 74 | 86.13 | 13.87 | 158 | 84.4 | 15.6 |
| | c1:90 | | | c1: 95 | | | | | |
| | c2:33 | | | c2:28 | | | | | |
| 1000 | a: 107 | 86.6 | 13.4 | a: 98 | 88.7 | 11.3 | 190 | 87.1 | 12.9 |
| | c1: 99 | | | c1: 112 | | | | | |
| | c2:34 | | | c2:37 | | | | | |
| 2000 | a: 136 | 91.05 | 8.95 | a: 152 | 85.25 | 14.75 | 256 | 90.35 | 9.65 |
| | c1: 132 | | | c1: 148 | | | | | |
| | c2:39 | | | c2:44 | | | | | |

TABLE VIII

Comparison of system performance after one training epoch on inputs presented in a different random order for cylinder-in-the-cube task using complement coding as input pre-processing. Training size: 250. $\rho=0$.

| Run No. | One-Pass | | | Two-Pass | | | Fuzzy ARTMAP | | |
|---------|----------------------|------------|-----------|----------------------|------------|-----------|----------------------|------------|-----------|
| | F ₂ Nodes | % test set | | F ₂ Nodes | % test set | | F ₂ Nodes | % test set | |
| | | correct | incorrect | | correct | incorrect | | correct | incorrect |
| 1 | a:29 | 82.4 | 17.6 | a: 16 | 74.4 | 25.6 | 15 | 84.8 | 15.2 |
| | c1:26 | | | c1: 15 | | | | | |
| | c2:14 | | | c2:10 | | | | | |
| 2 | a:20 | 70.4 | 29.6 | a: 17 | 66.4 | 33.6 | 16 | 86.8 | 13.2 |
| | c1:21 | | | c1: 17 | | | | | |
| | c2:12 | | | c2:12 | | | | | |
| 3 | a:39 | 76 | 24 | a: 20 | 78 | 22 | 11 | 84.8 | 15.2 |
| | c1:30 | | | c1: 20 | | | | | |
| | c2:18 | | | c2:13 | | | | | |
| 4 | a:25 | 68 | 32 | a: 30 | 86.13 | 13.87 | 9 | 86.4 | 13.6 |
| | c1:23 | | | c1: 26 | | | | | |
| | c2:14 | | | c2:28 | | | | | |
| 5 | a: 31 | 68.8 | 31.2 | a: 18 | 61.6 | 38.4 | 21 | 75.2 | 24.8 |
| | c1: 26 | | | c1: 18 | | | | | |
| | c2:16 | | | c2:10 | | | | | |

TABLE IX

Comparison of system performance after one training epoch for cylinder-in-the-cube task using different baseline matching parameters between ARTc and ARTa and fuzzification as input pre-processing. Training and testing set size: 1000.

| No. of Categories | One-Pass | | | No. of Categories | Two-Pass | | | Fuzzy ARTMAP | | | |
|----------------------------|--------------|------|----------------|----------------------------|--------------|------|----------------|-------------------|--------------|------|--------------|
| | Accuracy (%) | Test | $\bar{\rho}$ | | Accuracy (%) | Test | $\bar{\rho}$ | No. of Categories | Accuracy (%) | Test | $\bar{\rho}$ |
| a: 17 c1:137 c2:98 | 88.5 | 83.5 | a:0 c:0.7 | a: 11 c1: 110 c2:52 | 88.2 | 82.9 | a:0 c:0.7 | | | | N/A |
| a: 42 c1: 103 c2:59 | 93 | 88.1 | a:0.4 c:0.7 | a: 38 c1: 127 c2:50 | 92.9 | 83.9 | a:0.4 c:0.7 | | | | N/A |
| a: 105 c1: 100 c2:39 | 93.7 | 87.8 | a:0.7 c:0.7 | a: 93 c1: 113 c2:33 | 92.8 | 86.5 | a:0.7 c:0.7 | 186 | 99.8 | 88.8 | 0.7 |
| a: 174 c1: 95 c2:32 | 88.3 | 79.7 | a:0.8 c:0.7 | a: 173 c1: 122 c2:30 | 87.1 | 78.1 | a:0.8 c:0.7 | | | | N/A |

TABLE X

Voting strategy applied to set of three and five Fusion ARTMAP (One-pass) simulations of cylinder-in-the-cube task using complement coding as input pre-processing. Training and testing set size: 250. $\rho=0$.

| | % correct Test Set Predictions |
|---------------|--------------------------------|
| 3 simulations | |
| Average | 77.06 |
| Range | 82.4%-70.4% |
| Voting | 84.8 |
| 5 simulations | |
| Average | 76.48 |
| Range | 82.4%-68.8% |
| Voting | 88.8 |

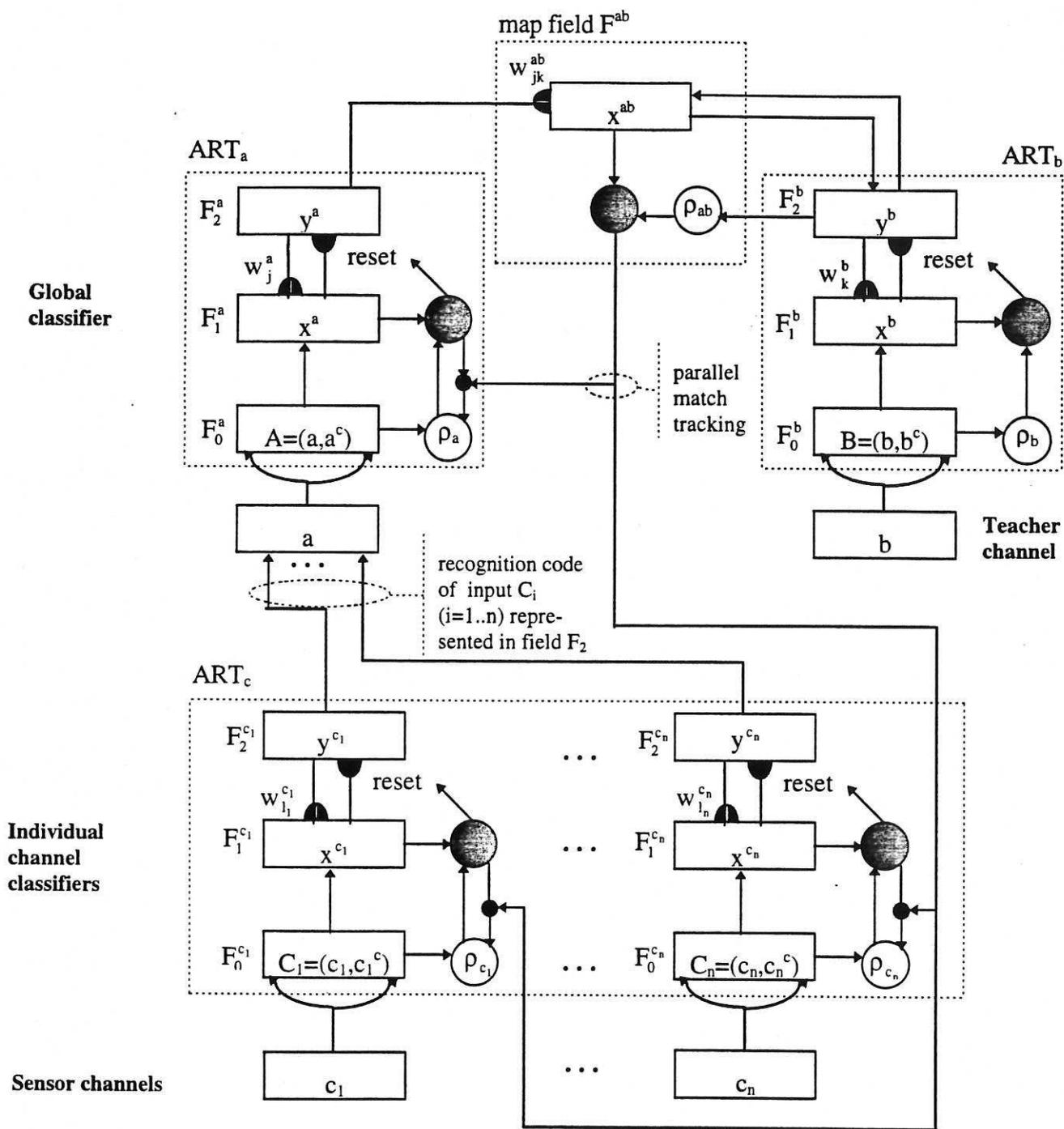


Figure 1. Fusion ARTMAP architecture: Multiple-input sensors with single-teacher channel.

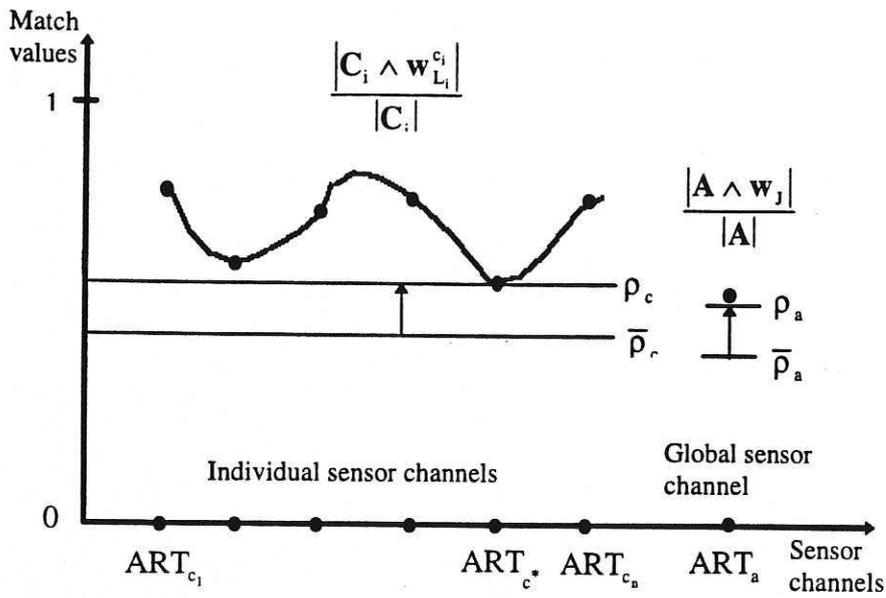


Figure 2. Parallel match tracking mechanism. When a predictive error occurs, parallel match tracking raises multiple vigilance values simultaneously until reset occurs in the ART module most likely to have caused the error.

