

Comparative Analysis of Fuzzy ART and ART-2A Network Clustering Performance

Thomas Frank, Karl-Friedrich Kraiss, and Torsten Kuhlen

Abstract—Adaptive resonance theory (ART) describes a family of self-organizing neural networks, capable of clustering arbitrary sequences of input patterns into stable recognition codes. Many different types of ART-networks have been developed to improve clustering capabilities. In this paper we compare clustering performance of different types of ART-networks: Fuzzy ART, ART 2A with and without complement encoded input patterns, and an Euclidean ART 2A-variation. All types are tested with two- and high-dimensional input patterns in order to illustrate general capabilities and characteristics in different system environments. Based on our simulation results, Fuzzy ART seems to be less appropriate whenever input signals are corrupted by additional noise, while ART 2A-type networks keep stable in all inspected environments. Together with other examined features, ART-architectures suited for particular applications can be selected.

Index Terms—Adaptive resonance theory, clustering, clustering analysis, neural networks, self-organization, sensor signals.

I. INTRODUCTION

SEVERAL huge sets of data have to be divided into separate categories. The need for setting up such categories may arise, e.g., from the need to set up recognition codes for complex system-state classes, or to discover separated clusters of data subsets with characteristic similarities (“data mining” [10]). In the field of neural networks, the adaptive resonance theory (ART), introduced and developed by Carpenter *et al.* from the Center for Adaptive Systems, Boston University [3], is a popular representative for self-organized clustering. Some outstanding features of ART, besides its clustering capabilities, attract the attention of application engineers. Among these are performance, economic usage of memory resources and temporal stability of stored knowledge. Neural networks are typically applied when standard statistical clustering methods fail on the interpretation of a given dataset, according to low performance or vast requirements of system resources. However, neural networks follow internal rules, making their applicability to a given problem predictable. The clustering performance of ART-networks is not well documented in the literature. It is assumed that clustering depends not only on the network architecture and parameters, but also on the dimensionality and nature of the clustered data. This

paper concentrates on the comparative analysis of clustering properties for several variants of ART-networks on two types of input patterns. Two-dimensional pattern sets illustrate the geometric characteristics of ART-clustering and the internal representation of knowledge by prototypes. Sampled step responses of second-order systems are used as an example of high-dimensional input patterns, modeling ART properties by clustering the shapes of time-dependent sensor signals.

II. SELF-ORGANIZED CLUSTERING WITH ART-NETWORKS

The common algorithm used for clustering in any kind of ART network is closely related to the well-known k -means algorithm [1]. Both use single prototypes to internally represent and dynamically adapt clusters. The k -means algorithm clusters a given set of input patterns into k groups. The parameter k thus specifies the coarseness of the partition. In contrast, ART uses a minimum required similarity between patterns that are grouped within one cluster. The resulting number k of clusters then depends on the distances (in terms of the applied metric) between all input patterns, presented to the network during training cycles. This similarity parameter is called *vigilance* ρ [3]. Fig. 1 illustrates the main stages of a simplified ART algorithm.

The first step, the *preprocessing* stage, is the creation of an input pattern as an array with a constant number of m elements. ART requires the same pattern size for all patterns, i.e., the dimension of the input space into which all cluster regions shall be placed. Any of the already formed prototypes is of the same dimension m . In addition, the elements of an input pattern must fit constraints concerning, e.g., value bounds or the geometric length of the array as vector. These constraints are characteristics of the different types of ART networks and are needed to make the input comparable to the cluster prototypes. Once the input pattern is formed, it is compared to the n stored prototypes in a *search* stage. If the degree of similarity between current input pattern and best fitting prototype¹ J is at least as high as vigilance ρ , this prototype is chosen to represent the cluster containing the input. The degree of similarity is typically limited to the range [0,1]. If similarity between input pattern and best fitting prototype does not fit into the vigilance interval $[\rho, 1]$, a new cluster has to be installed, where the current input is most commonly used as the first prototype or cluster center. Otherwise, if one of the previously committed clusters matches the input pattern well

Manuscript received July 30, 1997; revised January 6, 1998.

T. Frank is with Siemens Business Services, CC Machine Vision, D-53119 Bonn, Germany.

K.-F. Kraiss and T. Kuhlen are with the Lehrstuhl für Technische Informatik, Aachen University of Technology, D-52074 Aachen, Germany.

Publisher Item Identifier S 1045-9227(98)02766-0.

¹Upper case letters are used to mark the *winning* prototype/cluster of a comparison.

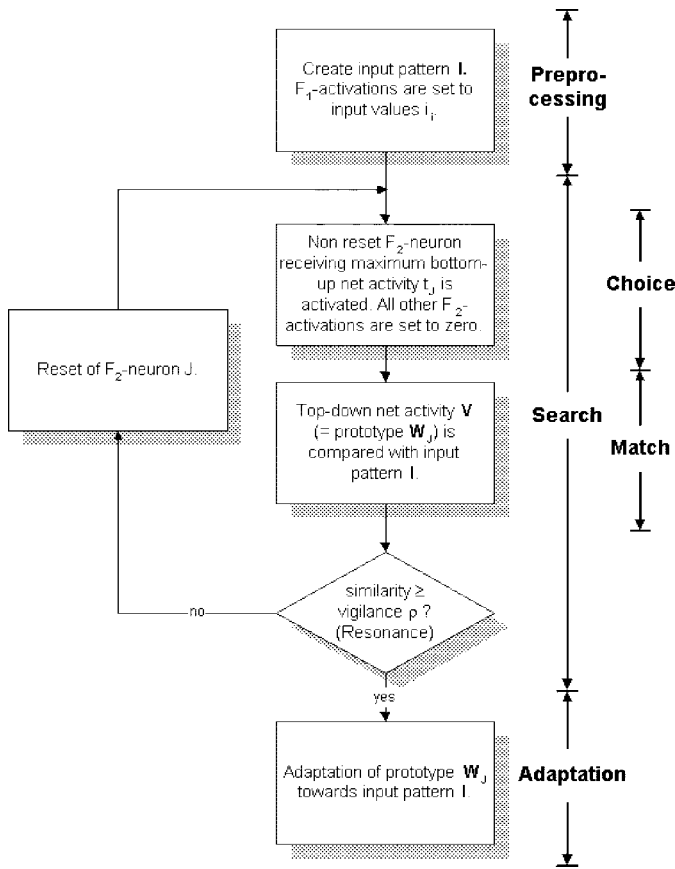


Fig. 1. Pattern processing by an ART network, consisting of a *preprocessing*-, *search*-, and *adaptation*-stage. The search stage will be defined more precisely as a circuit of *choice*, *match*, and *reset*. Termination of the algorithm is guaranteed by the initial values of prototypes.

enough, it is adapted by, e.g., slightly shifting the prototype's values toward the values of the input array.

Specific ART neural networks, such as ART 2 [2] or Fuzzy ART [7], more or less extend this basic layout to show a link between the computational characteristics of the algorithms and the biologically motivated connectionist approach. Concerning the description of the algorithms we used for ART computer simulations in this paper, these extensions are not our primary focus. More detailed information on particular network designs can be found in [2], [3], [7], and [14].

The primary processing module of any ART network is a *competitive learning network*, as shown in Figs. 2 and 3 [3]. The m neurons of an input layer F_1 register the values of an input pattern $\mathbf{I} = (i_1, i_2, \dots, i_m)$. Every neuron of an output layer F_2 receives a *bottom-up* net activity t_j , built from all F_1 -outputs $\mathbf{S} = \mathbf{I}$. The vector elements of $\mathbf{T} = (t_1, \dots, t_n)$ can be seen as results of comparisons between input pattern \mathbf{I} and prototypes $\mathbf{W}_1 = (w_{11}, \dots, w_{1m}), \dots, \mathbf{W}_n = (w_{n1}, \dots, w_{nm})$. These prototypes are stored in the synaptic weights of the connections between F_1 - and F_2 -neurons. Only F_2 -neuron J , receiving the highest net activity t_J , sets its output to one, while all other output neurons remain zero

$$u_j = \begin{cases} 1 & \text{if } t_j > \max(t_k : k \neq j) \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

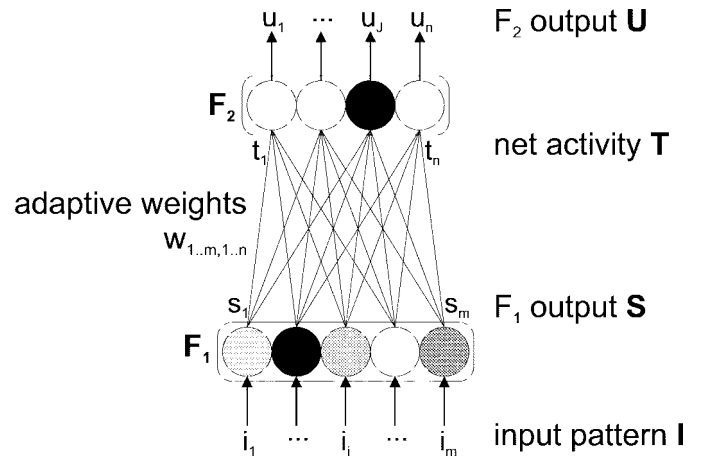


Fig. 2. A competitive learning network. Input layer F_1 adopts the values of input pattern \mathbf{I} . A *winner-take-all* output layer F_2 indicates the according cluster for \mathbf{I} , by the position of its one and only activated neuron J .

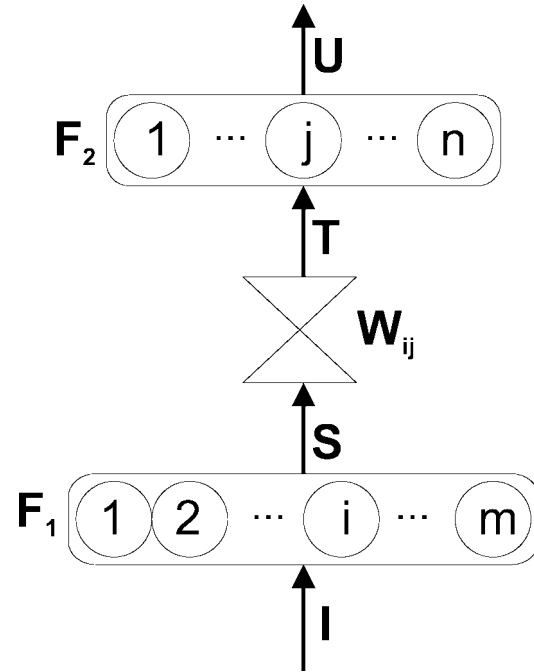


Fig. 3. A simplified representation of the competitive learning network from Fig. 2. All inputs and outputs of F_1 and F_2 are united in one arrow for any input or output vector. The adaptive weight-matrix \mathbf{W}_{ij} of all connections between layer F_1 and F_2 is replaced by the \bowtie -symbol.

One *possible* way to compute net activities t_j , and by that measure the similarity between \mathbf{I} and \mathbf{W}_j , is the weighted sum

$$t_j = \sum_{i=1}^m w_{ij} \cdot i_i. \quad (2)$$

Variations on this measure are often employed because the value t_j exerts great influence on the resulting clusters. After an F_2 -winner J has been found, the corresponding prototype $\mathbf{W}_J = (w_{1J}, \dots, w_{mJ})$ is adapted to the input pattern \mathbf{I} . One suitable method for adaptation is to move \mathbf{W}_J slightly toward input pattern \mathbf{I}

$$\mathbf{W}_J^{(\text{new})} = \eta \cdot \mathbf{I} + (1 - \eta) \cdot \mathbf{W}_J^{(\text{old})}. \quad (3)$$

The constant *learning rate* $\eta \in [0, 1]$ is chosen to prevent prototype \mathbf{W}_J from moving too fast and therefore destabilizing the learning process. Prototypes for this kind of competitive learning network can be initialized either with random values or with values of randomly chosen input patterns from the training sequence.

Competitive learning networks of this kind tend toward unstable categorization whenever the distances between single input patterns vary in too wide a range [3]. Additionally, there is no way to control either the number of clusters produced by the network, or the minimum similarity of patterns in one cluster. The ART solves this problem by extending the competitive learning network as shown in Fig. 4. A second set of connections is added, sending F_2 -output \mathbf{U} back to layer F_1 . The synaptic *top-down* weights \mathbf{W}_{ji} of these connections are, besides a possible scaling factor, identical to the *bottom-up* weights \mathbf{W}_{ij} . The *top-down* net activity \mathbf{V} is usually calculated by

$$v_i = \sum_{j=1}^n u_j \cdot w_{ji}. \quad (4)$$

This leads to

$$\mathbf{V} = \mathbf{U} \cdot \mathbf{W}_{ji} = \mathbf{W}_J \quad (5)$$

because all F_2 -outputs, except u_J , are set to zero [see (1)]. So input layer F_1 receives prototype \mathbf{W}_J , representing the current winning cluster J , as net activity. Now the most complex part of signal processing in ART networks takes place, i.e., matching prototype \mathbf{W}_J with input pattern \mathbf{I} . This task is completed in ways characteristic to the different types of ART networks and, as in ART 2 [2], uses extensions to the internal structure of layer F_1 . This yields a single matching value, that is compared with the *vigilance* ρ , defining the minimum similarity between an input pattern and the prototype of the cluster it is associated with. If the matching value is smaller than vigilance ρ , the current winning F_2 -neuron is removed from the competition by a *reset* signal. The reset signal forces the activation of F_2 -neuron J to zero and another F_2 -neuron is activated, receiving the highest net activity t_j of all nonreset output neurons. Once a prototype is found that leads to a matching value with input pattern \mathbf{I} , at least as high as vigilance ρ , no further reset signal is applied and the network attains *resonance*. The position of the last winning F_2 -neuron indicates the final cluster for input \mathbf{I} , and the associated prototype is adapted. Fig. 1 summarizes these steps of a single pattern processing by an ART-network. None of the output neurons is reset at the beginning.

The initial values of prototypes that have not yet been accessed by an input pattern, provide for two key features.

- 1) Previously accessed prototypes are first compared to the input pattern before an uncommitted prototype is chosen.
- 2) If none of the committed clusters matches the input pattern well enough, search will end with the recruitment of an uncommitted prototype.

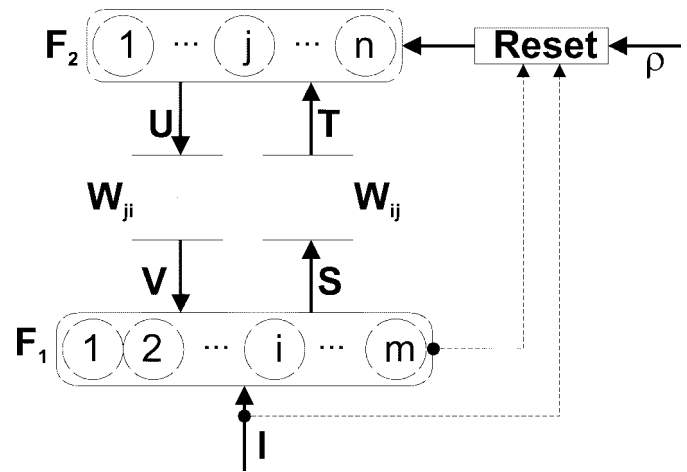


Fig. 4. Basic layout of an ART network. The competitive learning network from Fig. 3 is extended by a second set of connections leading all F_2 -outputs u_j back to the input layer F_1 .

III. DESCRIPTION OF ART-NETWORK ARCHITECTURES

The following section describes different ART-network architectures capable of processing analog input patterns. Properties will be discussed for Fuzzy ART [7] and ART 2A [6], as well as for some modified ART 2A algorithms. The ART 1 algorithm [3] is not discussed, as it is exclusively designed for binary input patterns and therefore not comparable to all other ART variants presented in this chapter. The predecessor of ART 2A, ART 2 [2], is neglected, because ART 2A incorporates nearly the same clustering characteristics while working several orders of magnitude more efficiently in computer simulations [6]. Some features of the neural architectures described in the original publications are skipped, as they are not relevant for this analysis.

A. Fuzzy ART

Referring to Fig. 1, any ART-type net can be characterized by its *preprocessing*-, *choice*-, *match*- and *adaptation*-rule, where *choice* and *match* define the search circuit for a fitting prototype. With Fuzzy ART [7], these rules are as follows.

- **Preprocessing**

All values of an input pattern must fit into the interval $[0, 1]$

$$i_k \in [0, 1] \quad \forall k. \quad (6)$$

- **Choice**

Bottom-up net activities, leading to a preliminary choice of a prototype, are determined using the fuzzy conjunction (\wedge), which is defined by

$$x \wedge y = \min\{x, y\} \\ \mathbf{X} \wedge \mathbf{Y} = (x_1 \wedge y_1, \dots, x_m \wedge y_m).$$

A single net activity t_j can be seen as the degree of prototype \mathbf{W}_j , being a *fuzzy subset* of input pattern \mathbf{I} [14]

$$t_j = \frac{|\mathbf{I} \wedge \mathbf{W}_j|}{\alpha + |\mathbf{W}_j|} \quad (7)$$

where \mathbf{Y} is fuzzy subset of \mathbf{X} , if $\mathbf{X} \wedge \mathbf{Y} = \mathbf{Y}$. The size of a vector ($|\mathbf{X}|$) is determined by its L_1 -norm, the sum of its components. The *choice parameter* α [7] provides a floating point overflow, if $|\mathbf{W}_j| \rightarrow 0$. In [9] some additional properties of Fuzzy ART with variations on α are pointed out, such as, e.g., lowest possible vector size of prototypes. Simulations in this paper are performed with a value of $\alpha \approx 0$.

- **Match**

The similarity of input \mathbf{I} and current winning prototype \mathbf{W}_j is measured by the degree of \mathbf{I} being a fuzzy subset of \mathbf{W}_j . Resonance and adaptation occurs, if

$$\rho \leq \frac{|\mathbf{I} \wedge \mathbf{W}_j|}{|\mathbf{I}|}. \quad (8)$$

- **Adaptation**

The winning prototype \mathbf{W}_j is adapted by moving its values toward the common MIN vector of \mathbf{I} and \mathbf{W}_j

$$\mathbf{W}_j^{(\text{new})} = \eta \cdot (\mathbf{I} \wedge \mathbf{W}_j^{(\text{old})}) + (1 - \eta) \cdot \mathbf{W}_j^{(\text{old})}. \quad (9)$$

The *learning rate* $\eta \in [0, 1]$ defines how quickly prototypes converge to the common minimum of all input patterns assigned to the same cluster. With $\eta \rightarrow 1$ the network is working in a *fast learning* mode [7], stabilizing the network state after a few presentations of all training patterns. In contrast, lower learning rates lead to a *slow learning* mode. ART-networks can simply be run in a pure *classification mode* by setting the learning rate of a previously trained network to zero, which prevents all prototypes from being modified by new input patterns. Uncommitted prototypes are initialized with a constant value

$$w_{ij} \geq 1 \quad \forall i. \quad (10)$$

This ensures that search will end if a previously uncommitted prototype is top-down compared with input \mathbf{I} by (8), since $|\mathbf{I} \wedge \mathbf{W}_j| = |\mathbf{I}|$ then. The higher the initial value for w_{ij} chosen, the lower the bottom-up net activity t_j resulting from an uncommitted prototype (7). By that, initial values $w_{ij} \rightarrow \infty$ guarantee that *all* committed prototypes are compared with the input, before an uncommitted cluster is chosen as winner.

A useful method to accelerate learning in ART networks is to set the learning rate $\eta = 1$ whenever a previously uncommitted cluster is adapted to the current input vector. Then input \mathbf{I} is identically copied as the first prototype of a new cluster if no other stored prototype matches the input well enough. Committed prototypes might then be adapted more slowly ($\eta < 1$), to preserve them from being corrupted by noisy input patterns. This method is called *fast-commit slow-recode* [7] and is used for all simulations of Fuzzy ART networks in this paper.

1) *Complement Coding*: Carpenter and Grossberg mention a problem of cluster proliferation that can occur with Fuzzy ART [7]. Because vector elements of prototypes can only become smaller by adaptation, a fuzzy ART network tends to create more and more prototypes over time that match input patterns with higher values, while prototypes with very low values might never be accessed further on. This behavior is

avoided by normalizing inputs to a constant vector length [7], [9]. One possible method is to use an Euclidean normalization to convert an input pattern \mathbf{A} into a coded input \mathbf{I}

$$\mathbf{I} = \frac{\mathbf{A}}{\|\mathbf{A}\|} = \frac{\mathbf{A}}{\sqrt{\sum_{i=1}^m a_i^2}} \quad a_i \geq 0 \quad \forall i \quad \|\mathbf{A}\| > 0.$$

The main disadvantage of this method is the complete loss of any information stored in the vector length of an input pattern. Therefore, a modified normalization variant called *complement coding* is typically used to set all input patterns to a common vector length [4], [7], [9]. An original vector $\mathbf{A} = (a_1, \dots, a_k)$ is coded into an input pattern $\mathbf{I} = (i_1, \dots, i_m)$ by adding the complements of its elements to the original vector. This doubles the dimension of all input patterns and prototypes

$$\begin{aligned} \mathbf{I} &= (\mathbf{A}, \mathbf{A}^c) \\ &= (a_1, \dots, a_k, 1 - a_1, \dots, 1 - a_k) \quad a_i \in [0, 1] \quad \forall i. \end{aligned} \quad (11)$$

The L_1 -norm² of complement encoded vectors of the same dimension is constant, independent of the values of their elements

$$\begin{aligned} |\mathbf{I}| &= \sum_{i=1}^{2k} i_i \\ &= \sum_{i=1}^k a_i + \sum_{i=1}^k 1 - a_i \\ &= \sum_{i=1}^k a_i + k - \sum_{i=1}^k a_i \\ &= k \\ &= m/2. \end{aligned} \quad (12)$$

Using complement coding, (8) reduces to

$$\rho \leq \frac{|\mathbf{I} \wedge \mathbf{W}_j|}{k}. \quad (13)$$

Uncommitted prototypes are still initialized according to (10). Working in fast learning mode ($\eta \rightarrow 1$), a prototype \mathbf{W}_j in Fuzzy ART represents the common MIN-vector of all l input patterns \mathbf{I}_p , with $p = 1 \dots l$, assigned to the same cluster j , in at least one presentation

$$\begin{aligned} \mathbf{W}_j &= \min\{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_l\} \\ &= (\min\{i_{11}, \dots, i_{l1}\}, \dots, \min\{i_{1m}, \dots, i_{lm}\}). \end{aligned} \quad (14)$$

Using complement coding, input patterns $\mathbf{I}_p = (\mathbf{A}_p, \mathbf{A}_p^c)$ lead to prototypes representing the common MIN- and MAX-vectors of all uncoded patterns \mathbf{A}_p

$$\begin{aligned} \mathbf{W}_j &= (\min\{\mathbf{A}_1, \dots, \mathbf{A}_l\}, \min\{\mathbf{A}_1^c, \dots, \mathbf{A}_l^c\}) \\ &= (\min\{\mathbf{A}_1, \dots, \mathbf{A}_l\}, \max\{\mathbf{A}_1, \dots, \mathbf{A}_l\}^c). \end{aligned} \quad (15)$$

With lower learning rates η , network prototypes converge more slowly to these MIN- and MAX-bounds.

²The L_r -norm is defined by $|\mathbf{X}|^{(r)} = \sqrt[r]{\sum_{i=1}^m x_i^r}$.

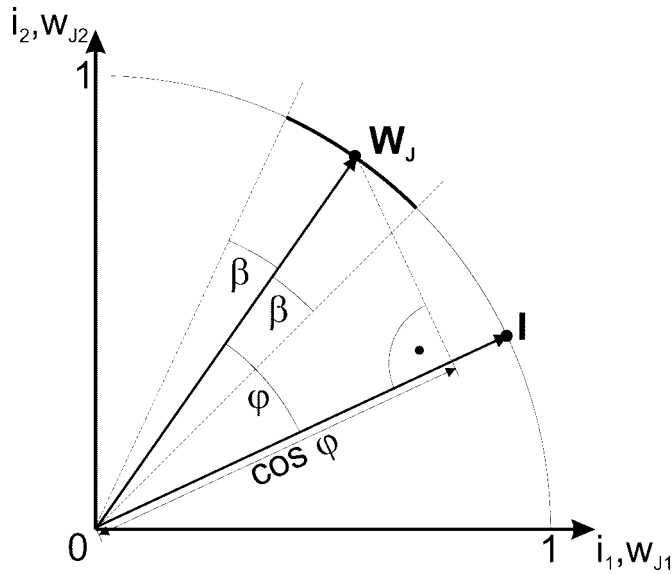


Fig. 5. Similarity in ART 2A is measured by angle φ between input vector \mathbf{I} and prototype \mathbf{W}_J . Input \mathbf{I} is assigned to cluster J if $\cos \varphi \geq \cos \beta = \text{vigilance } \rho$.

B. ART 2A

This section discusses another popular ART algorithm, called ART 2A [6]. In contrast to Fuzzy ART, ART 2A uses the angle between prototype vectors and input pattern to find a fitting cluster. Fig. 5 illustrates the relationships for two-dimensional input patterns.

The central functions of the ART 2A-algorithm, according to Fig. 1, are as follows.

- **Preprocessing**

No negative input values are allowed and all uncoded input vectors \mathbf{A} are normalized to unit Euclidean length, denoted by function symbol \aleph

$$\mathbf{I} = \aleph(\mathbf{A}) = \frac{\mathbf{A}}{\sqrt{\sum_{i=1}^m a_i^2}} = \frac{\mathbf{A}}{\|\mathbf{A}\|} \quad a_i \geq 0 \quad \forall i \quad \|\mathbf{A}\| > 0. \quad (16)$$

Carpenter and Grossberg suggest an additional method of noise suppression to contrast enhance characteristic pattern features by setting all input values to zero, which do not exceed a certain bias θ [2]

$$\mathbf{I} = \aleph(F_0(\aleph(\mathbf{A}))) \quad (17)$$

$$F_0(\mathbf{X})_i = \begin{cases} x_i & \text{if } x_i > \theta \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

$$\|\aleph(F_0(\aleph(\mathbf{A})))\| > 0,$$

This kind of contrast enhancement does only make sense if characteristic features of input patterns, leading to a distribution on different clusters, are coded *exclusively* in their highest values. With θ bounded by

$$0 \leq \theta \leq \frac{1}{\sqrt{m}} \quad (19)$$

the upper limit will lead to complete suppression of all patterns having the same constant value for all elements.

- **Choice**

Bottom-up net activities, leading to the choice of a prototype, are determined by

$$t_j = \begin{cases} \mathbf{I} \cdot \mathbf{W}_j & \text{if } j \text{ indexes a} \\ & \text{committed} \\ & \text{prototype} \\ \alpha \cdot \sum_{i=1}^m i_i & \text{otherwise} \end{cases} \quad (20)$$

$$0 \leq \alpha \leq \frac{1}{\sqrt{m}}. \quad (21)$$

Bottom-up net activities are determined differently for previously committed and uncommitted prototypes. The choice parameter $\alpha \geq 0$ again defines the maximum depth of search for a fitting cluster. With $\alpha = 0$, all committed prototypes are checked before an uncommitted prototype is chosen as winner. The simulations in this paper apply $\alpha = 0$.

- **Match**

Resonance and adaptation occurs either if J is the index of an *uncommitted* prototype or if J is a *committed* prototype and

$$\rho \leq \mathbf{I} \cdot \mathbf{W}_J = t_J. \quad (22)$$

- **Adaptation**

Adaptation of the final winning prototype requires a shift toward the current input pattern

$$\mathbf{W}_J^{(\text{new})} = \aleph(\eta \cdot \mathbf{I} + (1 - \eta) \cdot \mathbf{W}_J^{(\text{old})}) \quad 0 \leq \eta \leq 1. \quad (23)$$

ART 2A-type networks always use *fast-commit slow-encode* mode. Therefore the learning rate is set to $\eta = 1$ if J is an uncommitted prototype and to lower values for further adaptation. If contrast enhancement is used, (22) is modified to

$$\mathbf{W}_J^{(\text{new})} = \aleph(\eta \cdot \aleph(\Psi) + (1 - \eta) \cdot \mathbf{W}_J^{(\text{old})}) \quad (24)$$

$$\psi_i = \begin{cases} i_i & \text{if } w_{ji}^{(\text{old})} > \theta \\ 0 & \text{otherwise.} \end{cases}$$

Since *match* and *choice* do not evaluate the values of uncommitted prototypes, there is no need to initialize them with specific values. ART 2A-related networks should not be used in fast-learning mode with $\eta \cong 1$, because prototypes then begin to “*jump*” between all patterns assigned to their cluster, instead of converging toward their mean.

C. ART 2A-C: Complement Encoding with ART 2A

The main disadvantage of ART 2A for many implementations is the loss of all information coded in the length of an input pattern, because all patterns are normalized to unit Euclidean length. In other words, ART 2A cannot distinguish between two uncoded inputs \mathbf{A}_1 and \mathbf{A}_2 , where $\mathbf{A}_1 = c \cdot \mathbf{A}_2$, with $c > 0$. Using complement encoding, as described in Section III-A1, all information stored in the length of an uncoded vector \mathbf{A} is coded into the direction of the resulting vector $\mathbf{I} = (\mathbf{A}, \mathbf{A}^c)$. Whitely *et al.* used this method to extend

the recognition capabilities of ART 2A in processing analog sensor signals [12]. One way to include complement coding into the ART 2A algorithm, is to use it as an additional preprocessing step before entering the algorithm. Unfortunately, prototypes are normalized to unit length and adapted to normalized input patterns, when doing so. To keep a geometric interpretation of prototypes as a measure of all *uncoded* input patterns assigned to their cluster, normalization is moved from *preprocessing* and *adaptation* to the *choice/match* function. The complete algorithm is as follows.

- **Preprocessing**

$$\mathbf{I} = (\mathbf{A}, \mathbf{A}^c) \quad \alpha_i \in [0, 1] \quad \forall i. \quad (25)$$

- **Choice**

$$t_j = \begin{cases} \mathfrak{N}(\mathbf{I}) \cdot \mathfrak{N}(\mathbf{W}_j) & \text{if } j \text{ indexes a} \\ & \text{committed} \\ & \text{prototype} \\ \alpha \cdot \sum_{i=1}^m i_i & \text{otherwise} \end{cases} \quad (26)$$

$$0 \leq \alpha \leq \frac{1}{\sqrt{m}}. \quad (27)$$

- **Match**

As with ART 2A, resonance and adaptation occurs either if J indexes an *uncommitted* prototype or if J is already *committed* and

$$\rho \leq t_J. \quad (28)$$

- **Adaptation**

$$\mathbf{W}_J^{(\text{new})} = \eta \cdot \mathbf{I} + (1 - \eta) \cdot \mathbf{W}_J^{(\text{old})} \quad 0 \leq \eta \leq 1. \quad (29)$$

In contrast to Fuzzy ART using complement encoding, ART 2A-C prototypes remain a complement encoded pattern, even when adapted to several different input patterns. This is shown for two one-dimensional complement coded input patterns $\mathbf{A} = (a_1, 1 - a_1)$ and $\mathbf{B} = (b_1, 1 - b_1)$. Following (29), a vector $\mathbf{C} = \eta \cdot \mathbf{A} + (1 - \eta) \cdot \mathbf{B}$ is also a complement coded one-dimensional pattern

$$\mathbf{C} = (c_1, c_2)$$

with

$$c_1 = \eta \cdot a_1 + (1 - \eta) \cdot b_1$$

and

$$\begin{aligned} c_2 &= \eta \cdot (1 - a_1) + (1 - \eta) \cdot (1 - b_1) \\ &= 1 - (\eta \cdot a_1 + (1 - \eta) \cdot b_1) \\ &= 1 - c_1. \end{aligned}$$

Hence ART 2A-C prototypes represent a complement coded mean-vector of all input patterns assigned to the individual clusters.

D. ART 2A-E: Euclidean Distances in ART 2A

Another way to preserve vector-length information in ART 2A pattern processing, is to replace the ART 2A-distance metric with a Euclidean measurement of similarity and skip the length normalization of inputs in the preprocessing and adaptation stage. A comparable approach is done in [11]. The Euclidean algorithm used in this section, differs in some ways from [11] to keep it closer to the ART principles.

- **Preprocessing**

All elements of an input vector \mathbf{I} should fit to the interval $[0, 1]$

$$i_i \in [0, 1] \quad \forall i. \quad (30)$$

- **Choice**

Bottom-up net activities are determined using the Euclidean distance, normalized with the dimension m of an input vector. This keeps measurements of similarity independent from the number of vector elements. The distance is subtracted from one to get $t_j = 1$, if input vector and prototype \mathbf{W}_j are identical

$$t_j = 1 - \sqrt{\frac{1}{m} \sum_{i=1}^m (i_i - w_{ji})^2}. \quad (31)$$

Uncommitted prototypes should be initialized with values $w_{ij} \geq 1$ to achieve a sufficiently deep search for a fitting committed prototype.

- **Match**

The match function remains as in (28).

- **Adaptation**

$$\mathbf{W}_J^{(\text{new})} = \eta \cdot \mathbf{I} + (1 - \eta) \cdot \mathbf{W}_J^{(\text{old})} \quad 0 \leq \eta \leq 1. \quad (32)$$

Table I summarizes the ART-algorithms discussed. Choice- and match-functions of ART 2A-type networks are listed only for committed prototypes/clusters. All ART 2A-type network simulations are done in *fast-commit slow-recode* mode, setting $\eta = 1$ when adapting a previously uncommitted prototype and $\eta < 1$, else.

IV. COMPARATIVE ANALYSIS OF CLUSTERING PERFORMANCE

This section presents clustering examples for all ART-architectures discussed in the previous section. Two types of pattern sets, both consisting of 100 different patterns, are used to analyze and compare clustering performances. Patterns are presented in different random orders. A training sequence with a particular network and pattern set is stopped after each pattern has been presented a *minimum* number of times. Therefore not all patterns are necessarily presented the same number of times. This strategy takes into account that in many inspected environments neither the frequency nor the order of particular system states is predictable. Unless it is explicitly mentioned, simulations with different network types or network parameters are performed with the same random but constant order of patterns to keep the results comparable. When a training sequence is completed, the final distribution of the complete pattern set into clusters is tested with a

TABLE I
SUMMARY OF THE ART-ALGORITHMS USED FOR SIMULATIONS IN THIS PAPER

	Fuzzy ART	ART 2A-C	ART 2A-E	ART 2A
Preproc.	$\mathbf{I} = (\mathbf{A}, \mathbf{A}^c)$ $a_i \in [0, 1] \quad \forall i$	$\mathbf{I} = \mathbf{A}$ $a_i \in [0, 1] \quad \forall i$	$\mathbf{I} = \mathbf{A}$ $a_i \in [0, 1] \quad \forall i$	$\mathbf{I} = \frac{\mathbf{A}}{\ \mathbf{A}\ } = \aleph(\mathbf{A})$ $a_i \geq 0 \quad \forall i$ $\ \mathbf{A}\ > 0$
Choice	$t_j = \frac{ \mathbf{I} \wedge \mathbf{W}_j }{ \mathbf{W}_j }$	$t_j = \frac{\mathbf{I} \cdot \mathbf{W}_j}{\ \mathbf{I}\ \cdot \ \mathbf{W}_j\ }$	$t_j = 1 - \sqrt{\frac{\sum_i (i_i - w_{ji})^2}{m}}$	$t_j = \mathbf{I} \cdot \mathbf{W}_j$
Match	$\rho \leq \frac{ \mathbf{I} \wedge \mathbf{W}_j }{ \mathbf{I} }$	$\rho \leq t_j$		
Adapt.	$\mathbf{W}_j^{(new)} =$ $\eta \cdot (\mathbf{I} \wedge \mathbf{W}_j^{(old)})$ $+ (1 - \eta) \cdot \mathbf{W}_j^{(old)}$	$\mathbf{W}_j^{(new)} = \eta \cdot \mathbf{I} + (1 - \eta) \cdot \mathbf{W}_j^{(old)}$		$\mathbf{W}_j^{(new)} = \aleph(\eta \cdot \mathbf{I}$ $+ (1 - \eta) \cdot \mathbf{W}_j^{(old)})$

single presentation of each individual pattern in the set, while learning rate η is set to zero (*classification mode*).

A. Clustering Performance for Two-Dimensional Data

The first inspected pattern set consists of 100 two-dimensional patterns, representing uniformly distributed points in a unit square. Pattern values are taken from the interval $[0.1, 1.0]$ to fit the input restrictions of any described ART-network type. The spatial distribution of the data points does not support an obvious distinction of clusters. So the clustering performance in different simulations demonstrates clearly the different geometric interpretations of the pattern space and the according prototype representations. Fuzzy ART and the ART 2A-type networks are discussed separately, because of their characteristic differences.

1) *Two-Dimensional Clustering with Fuzzy ART*: In the special case of area points as two-dimensional input patterns, (15) defines four-dimensional Fuzzy ART-prototypes as rectangles, with the first two elements representing the lower left corner, and the last two elements representing *the complements* of the upper right coordinates. A stable network state is reached, when all uncoded training patterns $\mathbf{A} = (a_1, a_2)$ are enclosed by at least one of these prototype rectangles. If a complement coded input pattern $\mathbf{I} = (a_1, a_2, 1 - a_1, 1 - a_2)$ defines a point \mathbf{A} inside a prototype rectangle, (9) will lead to no further network modifications, since $\mathbf{I} \wedge \mathbf{W}_j = \mathbf{W}_j$ then. According to (13), resonance and adaptation of \mathbf{W}_j occurs, if

$$\rho \cdot k \leq |\mathbf{I} \wedge \mathbf{W}_j|. \quad (33)$$

Hence the vigilance parameter ρ defines a maximum extension of a single rectangle [7], with a lower bound for the L_1 -norm of any committed Fuzzy ART-prototype

$$\rho \cdot k \leq |\mathbf{W}_j| \leq k. \quad (34)$$

The upper limit results from (12) and defines the initial extension of a prototype after its first adaptation. With learning rate $\eta = 1$, the adaptation of a prototype toward an input pattern not yet lying within its area, stretches the according rectangle to the minimum area, covering all patterns assigned to the same cluster for at least one time. With $\eta = 1$ a stable network state is reached, as soon as all training patterns have been presented just one time (*one-shot learning* [7]). Fig. 6(a) shows an example for one-shot learning of the complete 2-D pattern set. The patterns were presented in a random order, until any pattern was processed by the network at least once. Cluster 5 is divided into two separate areas, 5a and 5b, because its prototype rectangle is partially covered by that of cluster 2.

Bottom-up net activities are set to their maximum value $t_j = 1$, whenever an input pattern defines a point inside a prototype rectangle [see (7)]. If this point is inside more than one rectangle, the prototype with *lowest index j* is chosen as winner³. The prototypes of the network, used in Fig. 6(a), are

$$\begin{aligned} \mathbf{W}_1 &= (0.4, 0.2, 0.0, 0.4) \\ &= ((0.4, 0.2), (1.0, 0.6))^c \\ \mathbf{W}_2 &= ((0.1, 0.4), (0.8, 0.7))^c \\ \mathbf{W}_3 &= ((0.3, 0.7), (1.0, 1.0))^c \\ \mathbf{W}_4 &= ((0.1, 0.1), (1.0, 0.2))^c \\ \mathbf{W}_5 &= ((0.1, 0.3), (0.3, 1.0))^c. \end{aligned}$$

With lower learning rates η in slow-learning mode, prototype rectangles do not tend to overlap that often. While rectangles cover the input space more efficiently up to their maximum size, the overall number of stored prototypes is

³The winning index might also be chosen at random, if more than one prototype leads to a maximum bottom up net activity.

Fuzzy ART

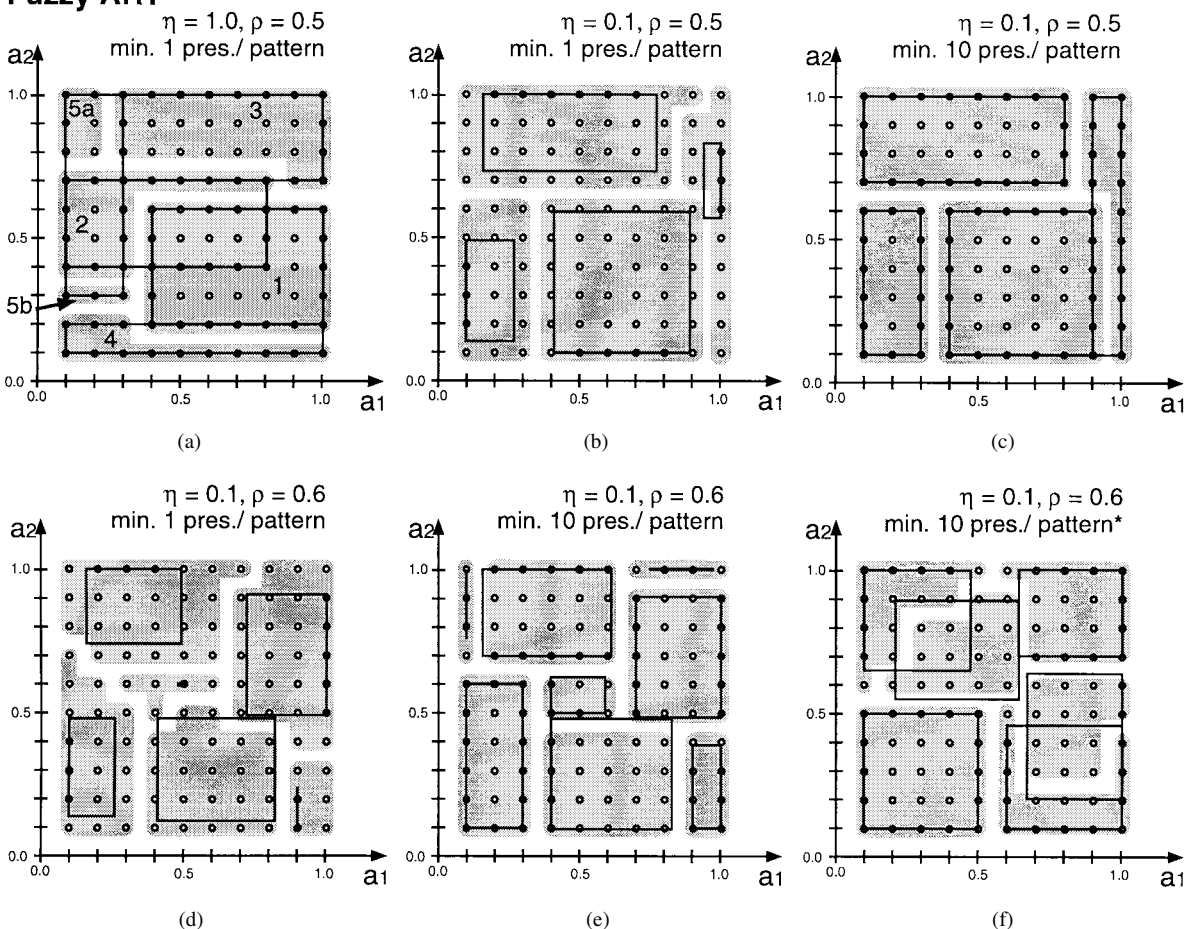


Fig. 6. Two-dimensional clustering performance for Fuzzy ART-networks. Circles mark the spatial positions of uncoded input patterns. Prototypes are printed as rectangles. Patterns assigned to a common cluster are marked with an underlying gray shade. Cluster 5 in (a) is split in two separate areas (5a and 5b). The experiments illustrated in (e) and (f) differ in their random pattern-presentation order.

reduced. Fig. 6(b) shows clusters from a network with the same vigilance, but lower learning rate than in Fig. 6(a), after adaptation to the same random training sequence. Because training stopped when each pattern was presented at least one time, prototype rectangles do not cover all patterns, assigned to the cluster they represent. Fig. 6(c) shows clusters when pattern presentation is continued until each pattern is processed at least ten times. Prototypes now reached their stable values, indicated by the fact, that all training patterns are covered by prototype rectangles.

With learning rates $\eta < 1$, the number of clusters, as well as the distribution of patterns to clusters, might vary throughout pattern presentation, as long as prototypes have not yet reached their stable equilibrium. Fig. 6(d) and 6(e) shows intermediate states of the same network after each pattern of the training sequence has been presented at least one and at least ten times. The number of clusters increases from six to eight. Higher vigilances limit the maximum area in pattern space covered by a single prototype rectangle and increase the total number of clusters on a static pattern set. Fig. 6(f) uses the same values for parameters η and ρ as Fig. 6(e). Each pattern is again presented at least ten times, but in a different random order. The example demonstrates, that even lower vigilances do not necessarily prevent cluster rectangles from overlapping. The

number of resulting clusters depends not only on vigilance parameter ρ but also on the order of pattern presentation.

2) *Two-Dimensional Clustering with ART 2A-Type Networks*: ART 2A prototypes are continually modified as long as the network is presented with input patterns. The prototypes thus never reach stable equilibria. An appropriate time to stop training is reached, when all patterns are assigned to the same clusters over two or more presentations. In most cases this state is reached after few training cycles, independent of the nature and size of the input patterns [12]. With the two-dimensional input patterns used in this section, stability in this sense is always reached with a combination of learning rate $\eta = 0.1$ and a minimum of ten presentations per pattern. The geometric interpretation of ART 2A-type prototypes is a mean-vector of all patterns assigned to the according clusters. Variations on η influence the number of presentations necessary to lead prototypes to this mean point. A rule-of-thumb is to choose η as high as possible, to still achieve stability in the above mentioned sense, with the lowest necessary number of presentations per pattern. As mentioned before, a Fuzzy ART-like *one-shot-learning* with $\eta = 1$ cannot be performed with ART 2A-type networks, since clustering will not stabilize when a prototype is always set to the values of the last input pattern assigned to the cluster it represents. Fig. 8 illustrates

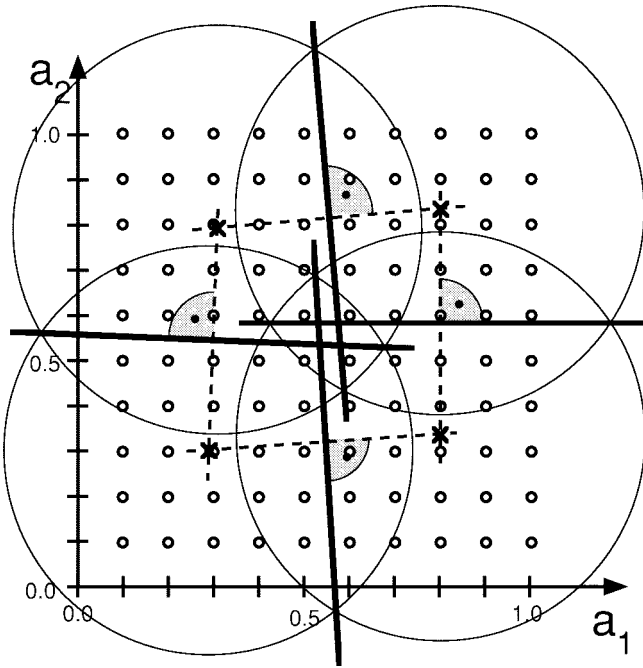


Fig. 7. Cluster borders in ART 2A-E are defined by the centerlines (thick lines) between prototype positions (crosses). Here illustrated for the clustering example of Fig. 8(g).

clustering results of ART 2A and the modified algorithms ART 2A-C and ART 2A-E. The networks were trained with two different values for vigilance parameter ρ and two different random pattern sequences (A and B). Both sequences stopped after each pattern was presented at least ten times. Vigilances were chosen with no other prediction but a number of three or four for the lower and five or six clusters for the higher value, when training with pattern sequence A. Prototype positions are marked with crosses. Even though ART 2A-C stores four-dimensional prototype vectors, the interpretation remains a two-dimensional complement encoded point in pattern space (see Section III-C). For example with Fig. 8(d), prototype values of ART 2A-C are

$$\begin{aligned} \mathbf{W}_1 &= (0.81, 0.33, 0.19, 0.67) \\ &= ((0.81, 0.33), (0.81, 0.33))^c \\ \mathbf{W}_2 &= ((0.33, 0.73), (0.33, 0.73))^c \\ \mathbf{W}_3 &= ((0.80, 0.83), (0.80, 0.83))^c \\ \mathbf{W}_4 &= ((0.28, 0.24), (0.28, 0.24))^c. \end{aligned}$$

The first thing to remark, when looking at Fig. 8, is the way pattern space is separated into clusters by the different network-types. Since ART 2A normalizes all prototypes and uncoded input patterns to unit Euclidean length (17), a pattern $\mathbf{I}_1 = (0.4, 0.5)$ appears the same to the network as a pattern $\mathbf{I}_2 = (0.8, 1.0)$. As a consequence, clusters separate the two-dimensional pattern space along radials [Fig. 8(a)–8(c)]. Vigilance ρ defines a maximum angle between input and prototype vector of $\beta = 20^\circ$ for $\rho = 0.94$ and $\beta = 11.5^\circ$ for $\rho = 0.98$ (see Fig. 5).

ART 2A-C and ART 2A-E behave in a very similar fashion in separating pattern space into clusters, even if ART 2A-C compares the angle between a complement coded input pattern

and a prototype, while ART 2A-E measures a Euclidean distance between input pattern and prototype. This emphasizes the fact that complement coding stores length information of an uncoded pattern in the direction of a coded pattern. With two-dimensional input patterns, the choice/match function (31) of ART 2A-E defines a circle around any stored prototype, with radius

$$r = (1 - \rho) \cdot \sqrt{m} = (1 - \rho) \cdot \sqrt{2}. \quad (35)$$

With Fig. 8(g)–8(i) the radii are $r = 0.45$ for $\rho = 0.68$ and $r = 0.34$ for $\rho = 0.76$. An input pattern is assigned to the cluster with the smallest Euclidean distance to its prototype and the pattern lying inside the according circle area. Otherwise, the input pattern is lying outside any of the already existing cluster circles, a new cluster with a new prototype is set up. Cluster borders within more than one circle area are defined by the centerline between the neighboring prototypes (Fig. 7).

Some properties are common for all ART 2A-type networks. Higher vigilances increase the number of clusters, set up on the same pattern sequence. The number of clusters can vary with the order of pattern presentation, as can be seen in the middle and right column of Fig. 8. In contrast to Fuzzy ART, clusters are always *coherent* in pattern space and never split in two separate areas as with Fig. 6(a).

B. Clustering Performance for Higher Dimensional Data

In Section IV-A ART-clustering with different network variants was illustrated on two-dimensional input patterns. Applications using clustering algorithms on various sources of input data usually deal with input patterns of higher dimensions. One typical scenario is the clustering of analog sensor data. Here, the task is to discover the structure of technical or biological system states as, e.g., in [8] or [12], by analyzing the shapes of specific time dependent sensor signals. Input patterns in these cases do not cover the whole multidimensional input space, but tend to form groups in geometrically separated areas. ART-networks are used to either discover stable categories of patterns with a minimum required similarity [8], or to set up recognition maps of an input space, by supervised assignment of ART clusters to pattern classes, with variants of ARTMAPS [4], [5], [12], [13]. In such applications the results of clustering should *not*, or at most very slightly, depend on the random order in pattern presentation. In addition, the network output should be fairly independent of additional noise, since input patterns built from sensor signals will always vary, even when representing exactly the same system state.

Clustering capabilities of ART networks are examined, using step responses of second-order systems (Fig. 9) as an example for a more general sensor signal shape. The response function is normalized so that the resulting oscillation converges around a value of $f(t) = 0.5$

$$\begin{aligned} f(t) &= 0.5 \cdot [1 - e^{-\zeta \cdot \omega_0 \cdot t} \cdot (\cos g(t) + c \cdot \sin g(t))] \\ g(t) &= \omega_0 \cdot \sqrt{1 - \zeta^2} \cdot t \\ c &= \frac{\zeta}{\sqrt{1 - \zeta^2}}. \end{aligned} \quad (36)$$

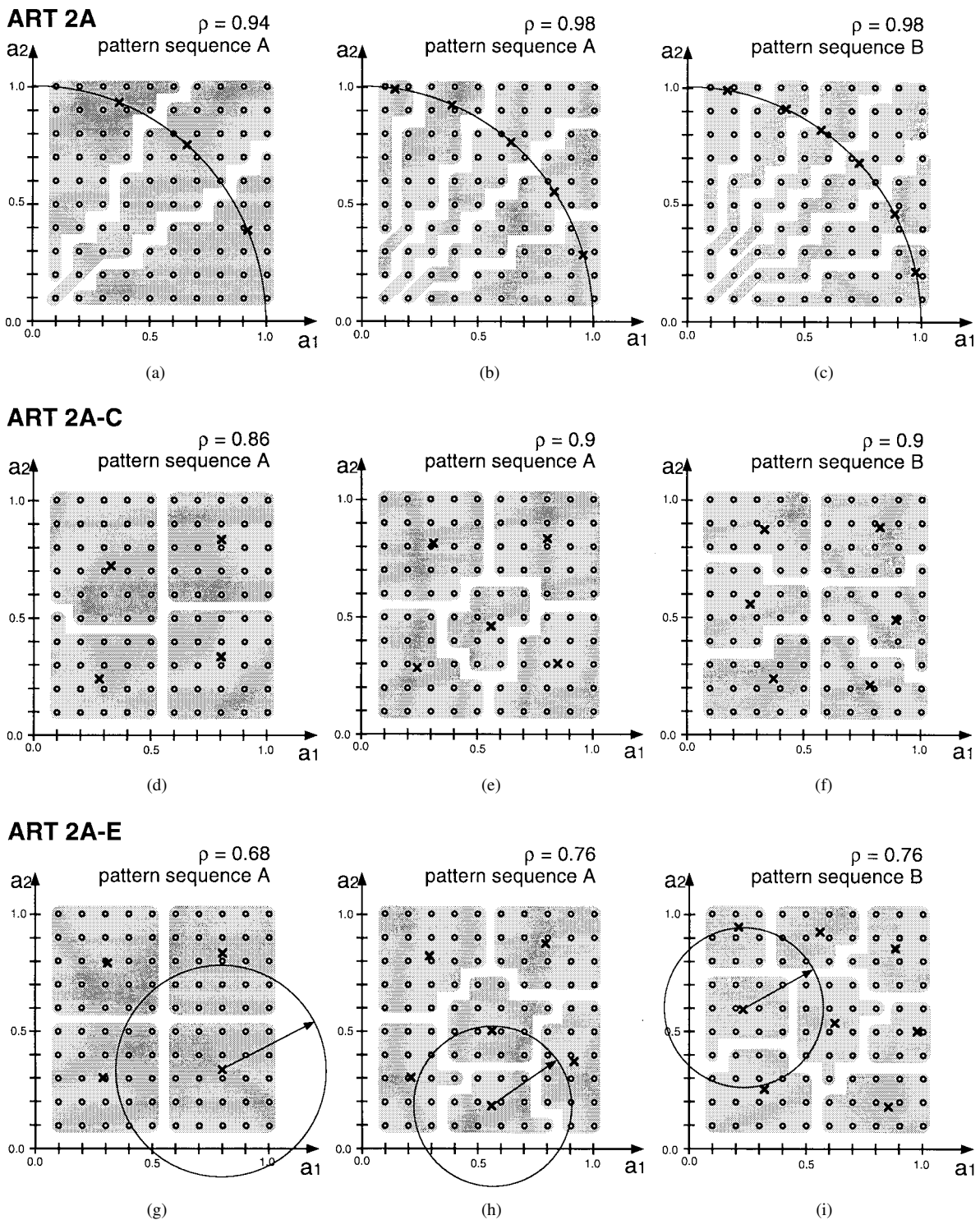


Fig. 8. Comparison of two-dimensional clustering performance for ART 2A-type networks. Circles mark the spatial positions of input patterns; crosses mark the positions of prototypes. Patterns assigned to a common cluster are marked with an underlying gray shade. The common learning rate used for all simulations is $\eta = 0.1$, with a minimum of ten presentations per input pattern in a random sequence.

Input vectors are formed out of 100 consecutive values of $f(t)$ with $t = 1 \dots 100$. A useful property of the step response is the fact that it is completely defined by two physical parameters, eigenfrequency $f_0 = \omega_0/2\pi$ and damping ζ . Therefore, input patterns as shown in Fig. 9, as well as clusters, can be depicted in a two-dimensional *PT2-parameter plane* to illustrate the influence of different network-parameter variations. The period

length of the step response in terms of inverse eigenfrequency is varied from ten to 100 time intervals in steps of ten. The damping is varied from $\zeta = 0.1$ to $\zeta = 0.9$ at ten equidistant intervals on a logarithmic scale. Step responses of the training-pattern set are equally distributed over this physical parameter plane, but represent points in discrete subareas of a 100-dimensional pattern space. So in contrast

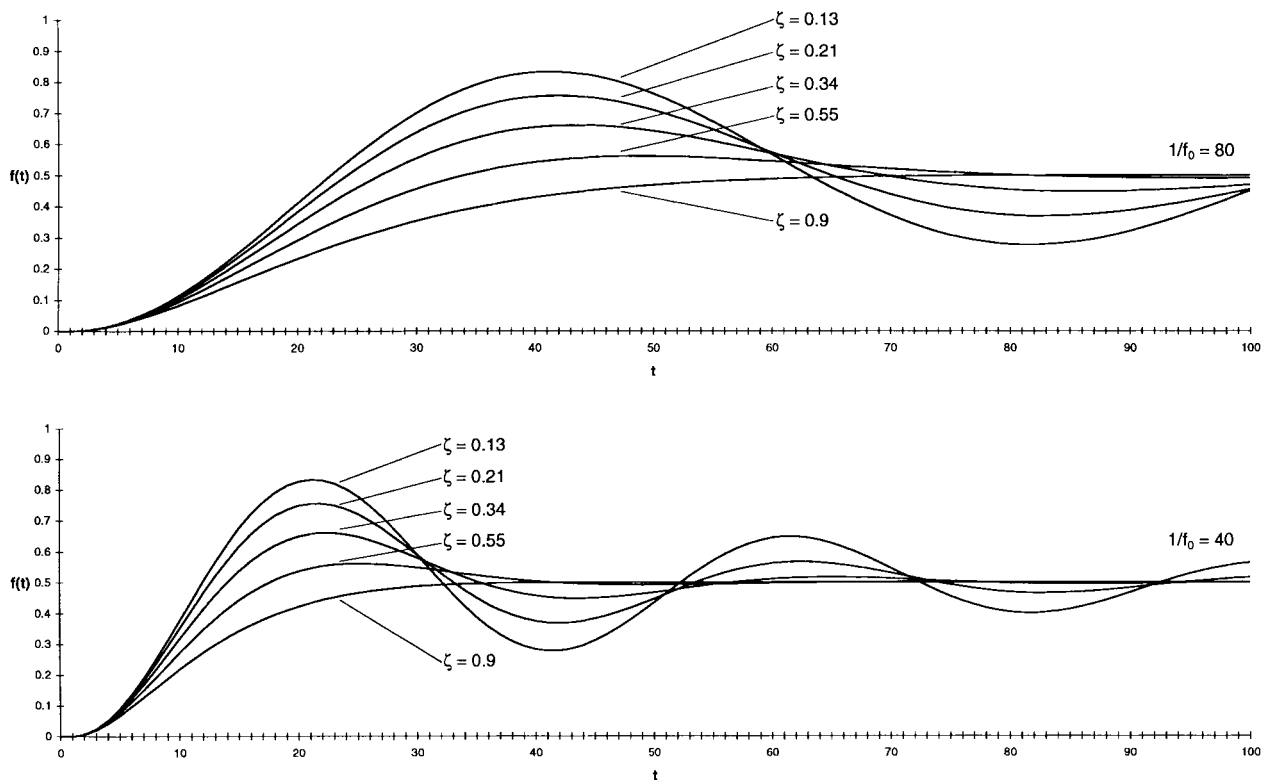


Fig. 9. Step responses of second-order systems (PT2) with different eigenfrequencies and dampings. Input patterns consist of 100 samples, taken at equidistant times.

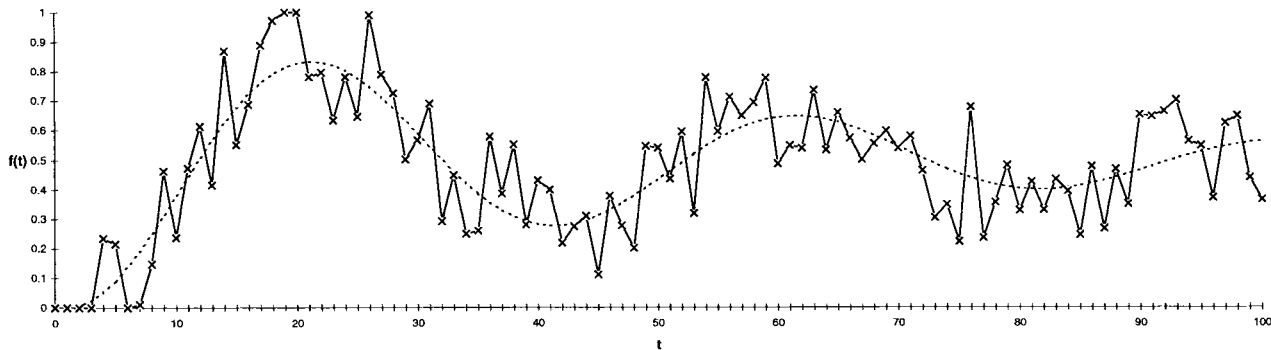


Fig. 10. PT2 step response with period length $1/f_0 = 40$ and damping $\zeta = 0.13$. The original signal (dotted line) is corrupted by a random white noise with maximum amplitude 0.25.

to the two-dimensional pattern set of the previous sections, there are geometrical preferences for clustering, which should be discovered independently of the random order in pattern presentation. Since exclusively damping and eigenfrequency determine the shape of the trained step responses, networks are expected to set up clusters, including shapes referenced by neighboring points in the parameter plane. The training set is presented in random orders, as with the two-dimensional pattern set (see introduction of Section IV).

Generalization capabilities of ART-networks are tested by classifying the pattern set with previously trained networks and learning rate $\eta = 0$, after any pattern has been corrupted with a random white noise (see Fig. 10). The more noisy patterns are assigned to the clusters of their undisturbed origins, the higher is the quality of generalization.

3) *Higher Dimensional Clustering with Fuzzy ART*: Fig. 11 shows clustering examples of step responses with Fuzzy ART in *one-shot-learning* mode, learning rate $\eta = 1$, and *slow-learning* mode, $\eta = 0.01$. The random pattern sequences were presented with a minimum of one presentation per pattern with $\eta = 1$, and 200 presentations per pattern with $\eta = 0.01$. Fig. 11(a) shows an example of one-shot learning with vigilance set to $\rho = 0.65$. The network set up seven clusters on the pattern set. Besides cluster 1, patterns from all other clusters are distributed over up to five separate coherent areas (gray shades) on the parameter plane, as shown for cluster 4. The cluster numbers represent the temporal order during training in which prototypes were accessed for the first time. Clusters in Fig. 11(b) are set up with the same network-parameters but a different random order in pattern

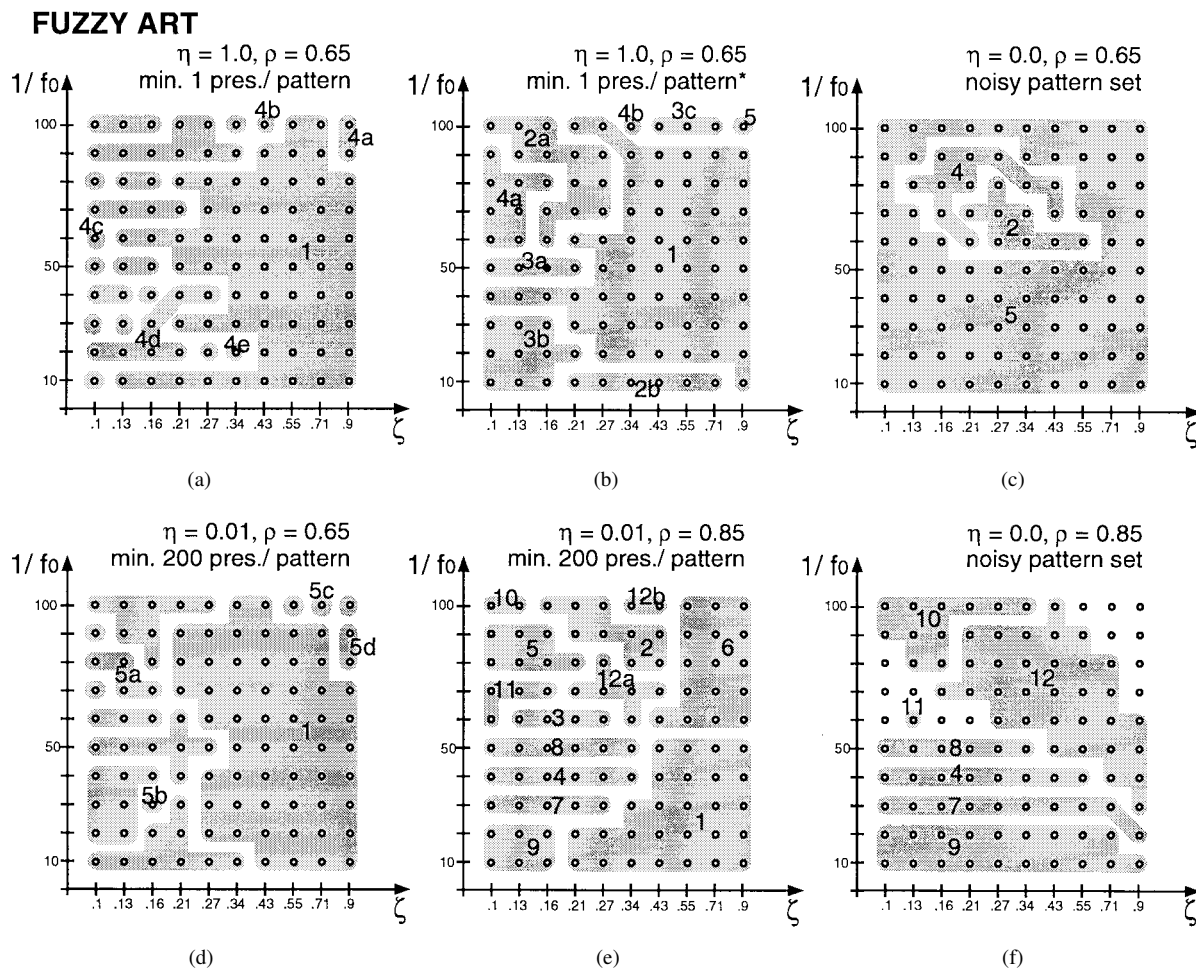


Fig. 11. High-dimensional clustering performance of Fuzzy ART networks. Step responses of second-order systems are defined by their eigenfrequency f_0 and damping ζ , marked with circles on the parameter plane. Gray shades group neighboring patterns in the parameter plane, assigned to the same cluster. Single Fuzzy ART clusters are often represented by more than one coherent area on the parameter plane. Experiments in (a) and (b) differ in the random order of input pattern presentation.

presentation. The scene is again dominated by a huge cluster 1 and four additional clusters, dividing the PT2-parameter plane in distinctly different clusters. Only cluster 1 and 5 are coherent, while clusters 2, 3, and 4 are split in up to three separate areas. Even in slow-learning mode, with at least 200 presentations per pattern, Fuzzy-ART clustering remains incoherent in the physical parameter-plane and highly dependent on the order of pattern presentation. In Fig. 11(d) seven clusters were set up, showing the same characteristics as with (a) and (b) in one-shot learning mode. Clustering tends to become more stable and coherent, when using higher vigilances as in Fig. 11(d), where only cluster 12 is defining two separate coherent areas on the plane. Fig. 11(c) and 11(f) demonstrate the assignment of noisy input patterns to the clusters of Fig. 11(b) and 11(e). All input patterns were corrupted with a random white noise of maximum amplitude 0.1, which is 2.5 times smaller than with Fig. 10. Most of the noisy patterns are assigned to different clusters than their undisturbed trained origins. Some patterns are even not assigned to a cluster at all, as to be seen in Fig. 11(f) with all points/parameter-pairs not included by a gray shade. If we assume that input patterns still represent the same physical state of the underlying second-order system, the recognition

capabilities of Fuzzy ART in real-world environments are rather discouraging.

Properties of Fuzzy ART depend on the state of its prototypes. Fig. 12 shows examples of prototypes, according to the network of Fig. 11(b) and 11(c). Once prototype areas have reached their stable equilibrium on the training pattern set, the MIN- and MAX-components of any Fuzzy ART prototype [see (15)] define the borders of an area, covering all patterns assigned to the associated clusters. The maximum area is again defined by (34). With $\rho = 0.65$ and uncoded patterns, consisting of 100 elements, the limits for prototypes are set to $65 \leq |\mathbf{W}_j| \leq 100$. With Fig. 12 the L_1 -norms are: $|\mathbf{W}_1| = 65.013$, $|\mathbf{W}_2| = 65.214$ and $|\mathbf{W}_5| = 100$. Cluster 5 refers to the last installed prototype during training, adapted to only one input pattern [$\zeta = 0.9$ and $1/f_0 = 100$ in Fig. 11(b)]. Once the L_1 -norm of a prototype reaches the lower limit, no pattern is assigned to its cluster, with even one element lying outside the MIN-/MAX-borders. This is why cluster 1 in Fig. 11(c) was accessed by none of the noisy input patterns, while cluster 5 attracts that many patterns. With a noise-amplitude of 0.25 instead of 0.1, as shown in Fig. 11(c), all noisy patterns are assigned to cluster 5. As a summary, Fuzzy ART recognition properties are highly sensitive to noise,

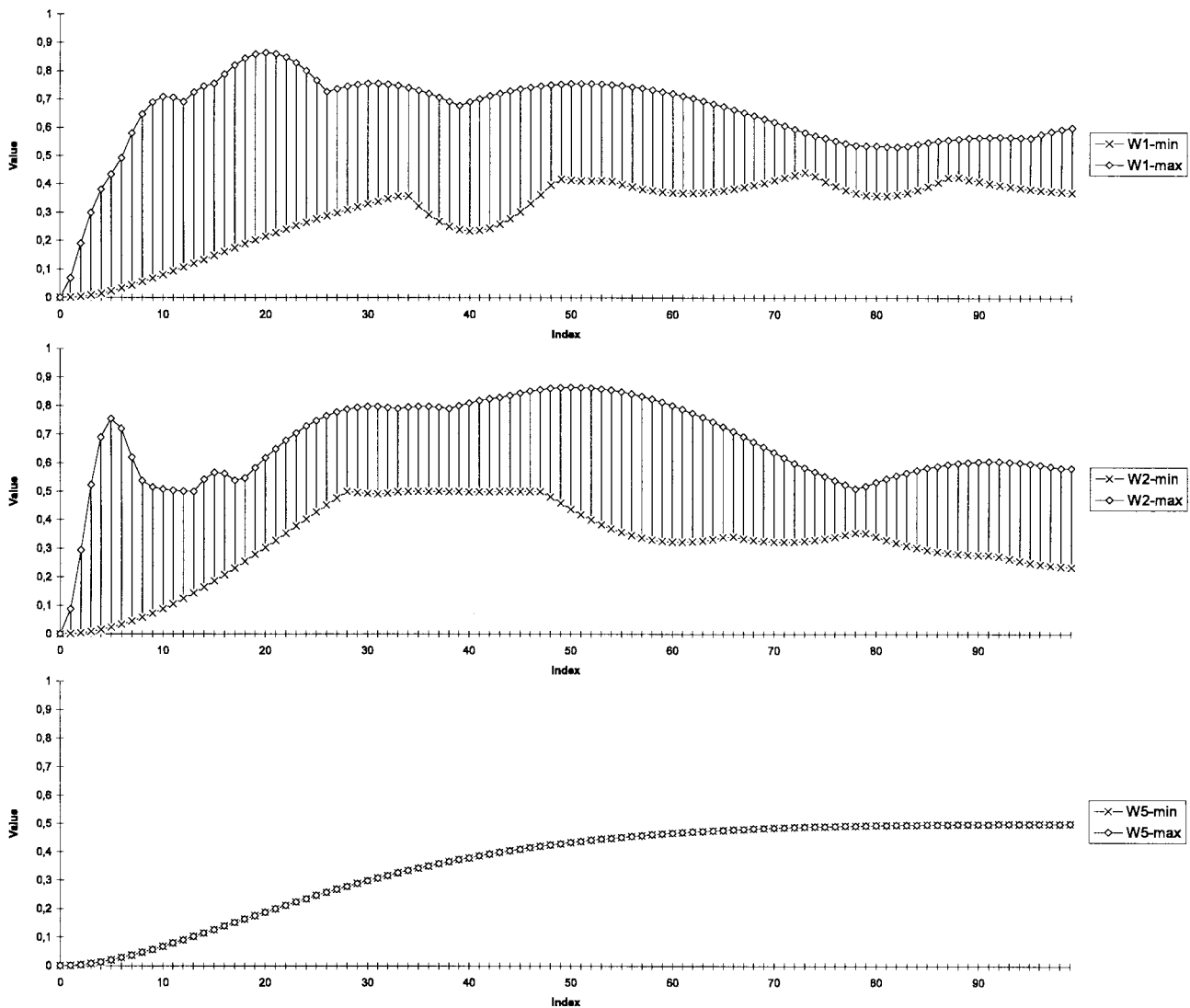


Fig. 12. Fuzzy ART-prototypes W_1 , W_2 , and W_5 , according to Fig. 11(b) and (c). The MIN- and MAX-components of the first and second half of a prototype are drawn separately, defining a hatched area, covering all input patterns assigned to the according cluster.

making its output unpredictable and indefinite, when input patterns in classification mode ($\eta = 0$) differ even slightly from the trained patterns.

Some of the Fuzzy ART properties may change, when the training of a network is terminated before prototype rectangles have reached their stable equilibrium. This method would generally exclude the *one-shot learning* mode, where prototype areas will usually reach their maximum size after a single presentation of a complete patterns set. In addition, Fuzzy ART loses its very useful property of indicating the completion of a training cycle through stability of prototypes. The question then arises, how to appoint an appropriate time to stop training.

4) *Higher-Dimensional Clustering with Art 2A-Type Networks*: Fig. 13 shows clustering examples of the PT2-step responses for ART 2A and the Euclidean ART 2A-E. All ART 2A-type networks, ART 2A with and without complement encoded input patterns and ART 2A-E, behave very similarly in clustering PT2 step responses, in spite of the different distance metrics. ART 2A-C and ART 2A-E already showed similarities when clustering two-dimensional

data (see Fig. 8). Since the Euclidean length of input vectors does not vary in too wide a range, the relative shape of patterns within the training set remains nearly the same, even when normalized to unit Euclidean length. So all ART 2A-type networks detect the same similarities and differences of the presented step responses. Networks were trained with a constant learning-rate of $\eta = 0.1$ and a minimum of 20 presentations per pattern. These values turned out to deliver prototypes close to the means of the assigned input patterns, and led to stable distribution of patterns to the same clusters over several training cycles. Fig. 14 shows prototypes of ART 2A and ART 2A-E, according to Fig. 13(b) and (e). The differences in the quality of the shapes between both network types are minimal. ART 2A-C prototypes approximate those of ART 2A-E, in as far as they have the same adaptation rules. ART 2A-C prototypes consist of 200 elements. The last 100 elements define the complement of the first (see Section III-C).

Clusters of ART 2A-type networks are always coherent in the physical parameter plane, just as they are in pattern space.

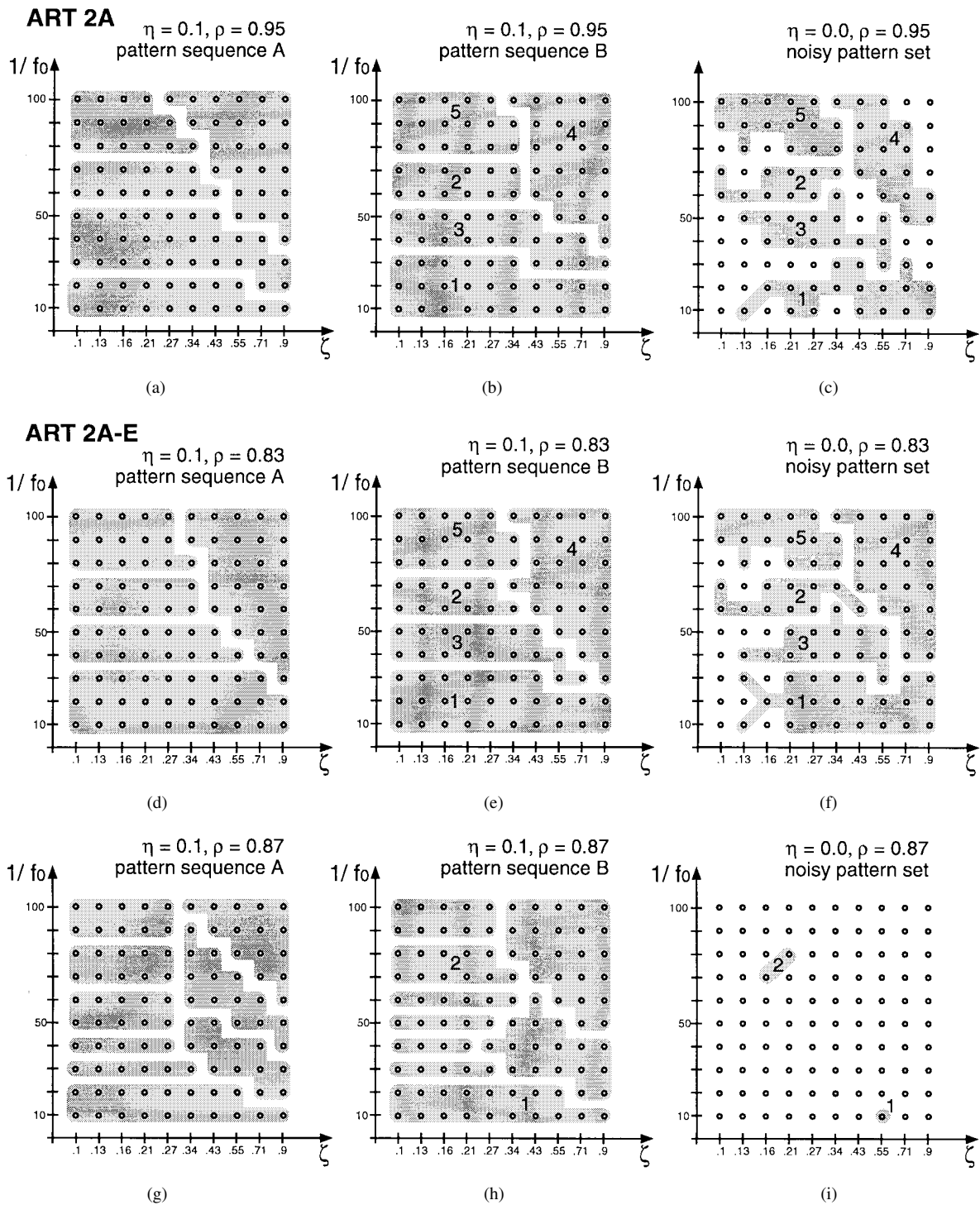


Fig. 13. High-dimensional clustering performance of ART 2A-type networks. All physical parameter-pairs grouped by a gray shade represent a complete cluster of 100-dimensional PT2-step responses. The simulations were done with a learning rate of $\eta = 0.1$ and a minimum of 20 presentations per pattern in two different random sequences *A* and *B*. Noisy patterns were classified with $\eta = 0$ and the trained networks from the middle column of figures. Noise amplitude is set to 0.25.

So ART 2A-clusters do not only reflect geometrical neighborhoods but also similarities in physical signal parameters of the second-order systems. Up to a limit of damping $\zeta \approx 0.3$ the eigenfrequency of the input patterns is the main criterion on which patterns are distinguished. Above that damping limit, clustering becomes more indefinite, with a tendency to a constant relation ζ/f_0 . Vigilances ρ were chosen with

respect to the number of clusters created by a network. The comparison of Fig. 13(a) and (d) and Fig. 13(b) and (e), shows the similarity in the shapes of clusters on the parameter plane for ART 2A and ART 2A-E. The same shapes turn out when clustering the pattern set with ART 2A-C and an appropriately chosen vigilance. Comparing the left and middle column in Fig. 13, clustering turns out to be rather independent of the

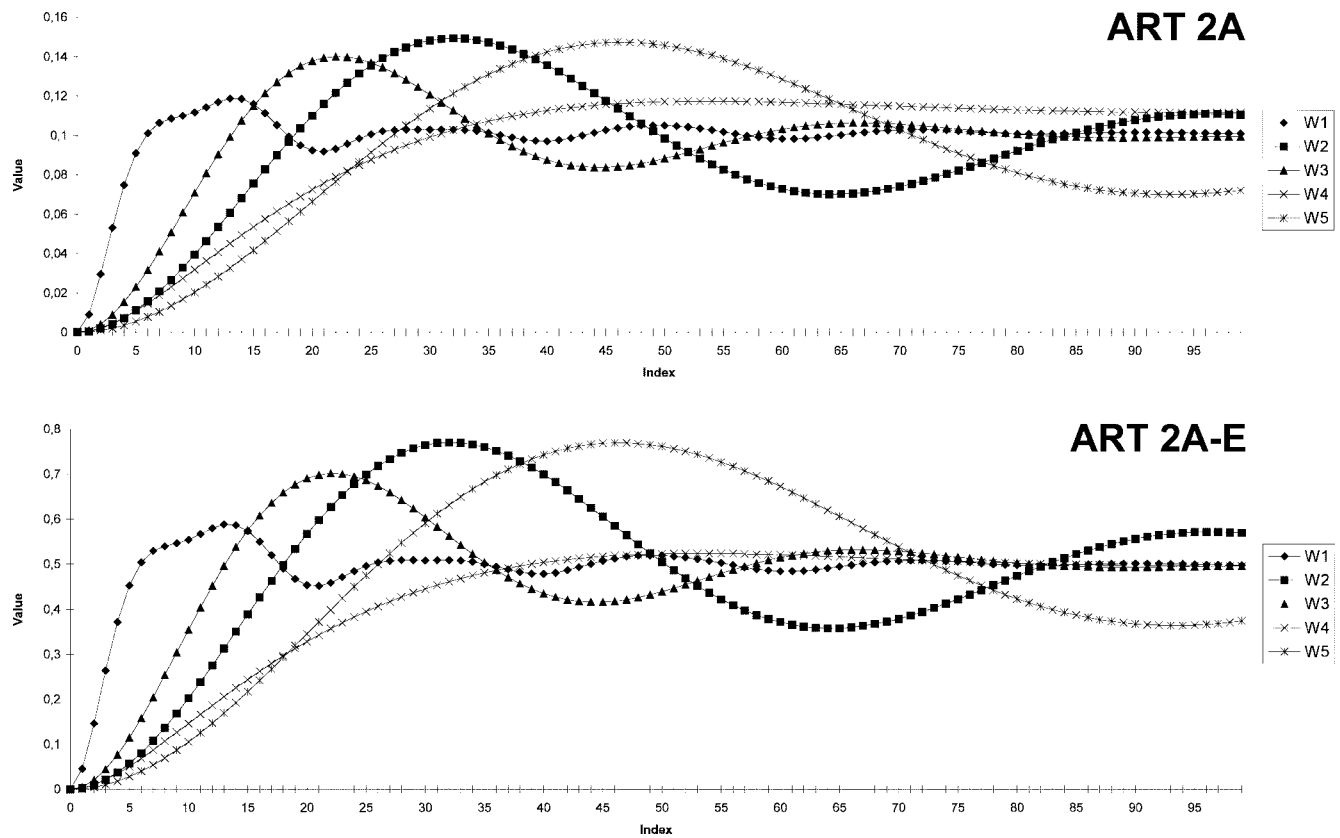


Fig. 14. Prototype representations, referring to the ART 2A clusters of Fig. 13(b) and the ART 2A-E clusters of Fig. 13(e).

random order in pattern presentation, especially for lower vigilances and higher cluster sizes.

Generalization and recognition properties are illustrated with a noise corrupted pattern set, classified by the trained networks from the middle column of Fig. 13. Noise amplitude is set to 0.25 (see Fig. 10). Patterns not assigned to the same clusters as their undisturbed trained origin, are rather assigned to *no* cluster at all than assigned to the *wrong* cluster (right column of Fig. 13). The disturbing influence of additional noise on the assignment of patterns to existing clusters increases, together with vigilance parameter ρ [Fig. 13(i)]. Up to noise amplitudes of about 0.15, pattern distribution did almost not vary from that of the undisturbed patterns. That means, that for many applications ART 2A-like network variants can be treated as rather insensitive against small variations on input patterns, representing the same state of the inspected system.

V. CONCLUSION

Leaving aside the biologically motivated aspects, ART turns out to be an effective, transparent clustering algorithm. Two different types of ART-networks, Fuzzy ART and ART 2A, as well as two ART 2A-modifications, ART 2A-C and ART 2A-E, were inspected. Each variant is characterized by its preprocessing-, choice-, match- and adaptation-rule (see Table I). Two-dimensional pattern sets illustrated the geometric nature of ART-clusters. Fuzzy ART uses the degree of an input pattern being fuzzy subset of a stored prototype to measure the similarity between two patterns. When using complement encoded input patterns, prototypes converge toward

the common MIN- and MAX-values of all patterns assigned to the according cluster. Clusters separate the pattern space along the pattern space axes. In contrast ART 2A measures the cosine of the angle between input- and prototype-vector, separating pattern space along radials. Using complement encoding with ART 2A (ART 2A-C), length and orientation of an uncoded input vector is stored as a higher dimensional coded pattern. The result is a distance metric comparable to the Euclidean metric of ART 2A-E. Properties of ART-networks depend on two main parameters, ρ and η . Vigilance ρ defines the minimum similarity between patterns in one cluster in terms of the applied distance metric. Higher vigilances increase the total number of clusters set up on a static pattern set. If no geometric preferences are given for a specific pattern set, as with the two-dimensional patterns in this paper, the number of clusters is also slightly dependent on the order of pattern presentation. Learning rate η regulates adaptation of stored prototypes toward input patterns.

Fuzzy ART networks reach a state of temporally stable prototypes, indicating the end of a training cycle on a fixed set of patterns. All network weights are fixed, when all training patterns are enclosed by the MIN- and MAX-bounds defined by the prototypes. The extension of prototypes is limited by the vigilance parameter ρ . Once the maximum extension of a prototype has been reached, no further patterns are assigned to the according cluster not lying completely within the MIN- and MAX-borders. This makes Fuzzy ART highly sensitive to additional noise on trained input patterns and its output unpredictable. Even if the geometric distribution of input

patterns in pattern space gives preferences for the distribution of these patterns to clusters, Fuzzy ART clusters remain highly dependent on the random order of pattern presentation and tend to be incoherent in pattern space. The example of sampled PT2-step responses in this paper illustrated that Fuzzy ART-clusters can even be incoherent in the physical parameter space.

For most applications, where pure self-organized clustering of a pattern set is required, ART 2A is the more appropriate solution. If there are geometric preferences within a given pattern set, ART 2A-type networks discover them, independently from the random order of pattern presentation. ART 2A-type clusters are always coherent in pattern space, and in addition are also always coherent in the parameter plane of second-order systems. Even if there is no *stable* network state as with Fuzzy ART, after a few presentations of a pattern set a network state is reached, where single patterns will not change their clusters anymore, and prototypes represent a means of all accorded input patterns. Decisions on which kind of ART 2A-type network should be chosen for a particular application depend on the computational requirements of the algorithms on specific platforms. If all inspected system information is stored in the direction of input vectors, pure ART 2A is a fast alternative; otherwise one of its variations should be chosen.

REFERENCES

- [1] S. Banks, *Signal Processing, Image Processing, and Pattern Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1990.
- [2] G. A. Carpenter and S. Grossberg, "ART 2: Self-organization of stable category recognition codes for analog input patterns," *Appl. Opt.*, vol. 26, no. 23, pp. 4919–4930, 1987.
- [3] ———, "The art of adaptive pattern recognition by a self-organizing neural network," *IEEE Computer*, pp. 77–88, Mar. 1987.
- [4] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: A neural-network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Trans. Neural Networks*, vol. 3, pp. 698–713, 1992.
- [5] G. A. Carpenter, S. Grossberg, and J. H. Reynolds, "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network," *Neural Networks*, vol. 4, pp. 565–588, 1991.
- [6] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition," *Neural Networks*, vol. 4, pp. 493–504, 1991.
- [7] ———, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, pp. 759–771, 1991.
- [8] T. Frank, "Anwendung der theorie der adaptiven resonanz auf die analyse sensorisch gestörter hand-arm bewegungen," Diploma Thesis, Lehrstuhl für Technische Informatik, Aachen Univ. Technol., Germany, 1996 (in German).
- [9] J. Huang, M. Georgiopoulos, and G. L. Heileman, "Fuzzy ART properties," *Neural Networks*, vol. 8, no. 2, pp. 203–213, 1995.

- [10] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, "Engineering applications of the self-organizing map," *Proc. IEEE*, 1996, vol. 84, pp. 1358–1384.
- [11] Y.-H. Pao, *Adaptive Pattern Recognition and Neural Networks*. Reading, MA: Addison-Wesley, 1989.
- [12] J. R. Whitely, J. F. Davis, A. Mehrotra, and S. C. Ahalt, "Observations and problems applying ART2 for dynamic sensor pattern interpretation," *IEEE Trans. Syst., Man, Cybern.*, vol. 26, pp. 423–437, 1996.
- [13] J. R. Williamson, "Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps," *Neural Networks*, vol. 9, no. 5, pp. 881–897, 1996.
- [14] A. Zell, *Simulation Neuronaler Netze*. Reading, MA: Addison-Wesley, 1994 (in German).



Thomas Frank was born in 1967 and received the Diploma degree in electrical engineering from Aachen University of Technology, Germany, in 1996.

He is currently employed by Siemens Business Services CC Machine Vision, Bonn, Germany, as System Engineer for object recognition. His research interests include neural networks, pattern recognition, and image processing.



Karl-Friedrich Kraiss received the Diploma degree in electrical engineering and the Doctorate degree in mechanical engineering from Berlin University of Technology, Germany, in 1966 and 1970, respectively. In 1987 he received the Habilitation for Aerospace Human Factors Engineering degree from Aachen University of Technology, Germany.

Currently he is a Professor at Aachen University of Technology where he holds the Chair for Technical Computer Science in the Department of Electrical Engineering. His research interests include machine intelligence, machine vision, autonomous mobile systems, advanced human machine interfaces, and usability engineering. In these areas he has written numerous scientific publications, among which are three books.

In 1988 Dr. Kraiss received the "Technical Communication" research award from the Alcatel/SEL foundation for research in human machine communication.



Torsten Kuhlen was born in 1964 and received the Diploma degree in Computer Science from Aachen University of Technology, Germany, in 1992.

He is currently pursuing the Ph.D. degree at the Institute of Technical Computer Science in the same University. His research is on advanced human-machine interfaces with applications to robotics and medicine.