# Analysis of Switching Dynamics with Competing Neural Networks

Klaus-Robert Müller[†] [††], Jens Kohlmorgen[††] *and* Klaus Pawelzik[†††], *Non-members*

**SUMMARY** We present a framework for the unsupervised segmentation of time series. It applies to non-stationary signals originating from different dynamical systems which alternate in time, a phenomenon which appears in many natural systems. In our approach, predictors compete for data points of a given time series. We combine competition and evolutionary inertia to a learning rule. Under this learning rule the system evolves such that the predictors, which finally survive, unambiguously identify the underlying processes. The segmentation achieved by this method is very precise and transients are included, a fact, which makes our approach promising for future applications.
*key words: neural networks, non-linear dynamics, chaos, time series analysis, prediction, competing neural networks*

## 1. Introduction

Neural networks have been broadly used as approximators of nonlinear functions and as tools for classification and prediction. Trained from input and output examples, they provide a powerful structure for the representation of relations present in data [1, 2]. An important prerequisite for the successful application of such systems, however, is a certain uniformity of the data. In most cases of temporally ordered data, a stationary process is assumed, i.e. the relations remain constant over time. If, on the contrary, the data originate from different sources, the assumption of stationarity has to be discarded. In principle, we can think of three different kinds of non-stationarities: (a) a superposition of many sources or the case, where the underlying system (b) drifts or (c) switches between different dynamics. In all cases standard approaches like simple multi-layer perceptrons are likely to fail to represent the underlying input-output relations. In the present study, we focus on *switching* dynamics with a low switching rate[‡] (case (c)), that is inherent in a raw (unlabeled) data stream. In other words, we consider an unknown number of unknown sources that alternate in time.

While prediction of a stationary dynamical system in principle is only restricted by the largest Lyapunov exponents [5], the reconstruction of different alternating systems is only possible when the switching points are known. Such time series can originate from many kinds of systems, like stochastic dynamics or hidden markov models. Phenomena of this kind are e.g. speech, brain data, and systems which switch their attractors [6].

The well-known *mixtures of experts* approach, proposed in [4], will also fail to capture the non-stationarities, if the information, which is provided by the input data, is not sufficient for the unambiguous determination of the output. In these cases, additional information has to be added to the input in order to discriminate the data. One way to solve this problem is the inclusion of memory effects. But also in *mixture* approaches which include memory [17], a clear cut division of the data according to the underlying sources is not guaranteed. What is missing, is a training scheme, in which finally the data are exclusively distributed among the experts. As a solution of this problem, we proposed two algorithms:

The first one uses hard competition, where the experts segment according to best performance (winner-takes-all) [13, 14]. This approach is able to segment and identify data streams very efficiently (see section 4.1). Yet, if the underlying systems are very similar, the final solution can sensitively depend on the choice of initial conditions and mixings between the similar sources can occur.

The second method controls the degree of diversification of experts. We use a set of competing neural networks for the prediction of the data, where the competition is driven by an evaluation of the prediction performance. The main feature of this approach is an adiabatic enforcement of competition during training, which is described in detail in [16]. Such "annealing of competition" is useful, whenever an unsupervised classification is required. In the following, we call this method "Annealed Competition of Experts" (ACE). It provides a hierarchically ordered representation of "natural" classes, since the resolution of different clusters depends on specific "temperatures" (cf. [16]). This effect of spontaneous organization through bifurcations

[†]The author is with Department of mathematical Engineering and Information Physics, University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo 113, Japan.

[††]The author is with GMD FIRST, Rudower Chaussee 5, 12489 Berlin, Germany.

[†††]The author is with Institut für Theoretische Physik, Universität Frankfurt, 60054 Frankfurt/M., Germany.

[‡]By the assumption of a low switching rate between the different dynamical systems, we strongly simplify the inclusion of memory into the learning rule. Although the asumption of slow switching seems at first sight to be limiting, our algorithm was successfully applied even for switching in every tenth iteration.

also occurs in our application of this training strategy (see section 4.2).

For synthetic data from alternating chaotic systems both approaches provide a very precise segmentation, that leads to accurate classification and nearly optimal prediction (see section 4.). We also applied our method to neurophysiological time series and to speech signals, finding that these data can be naturally classified according to a switching dynamics.

Sections 2 and 3 refer to our segmentation and identification algorithms. In section 4 the simulation results for the applications studied are presented. A short conclusion is given in section 5.

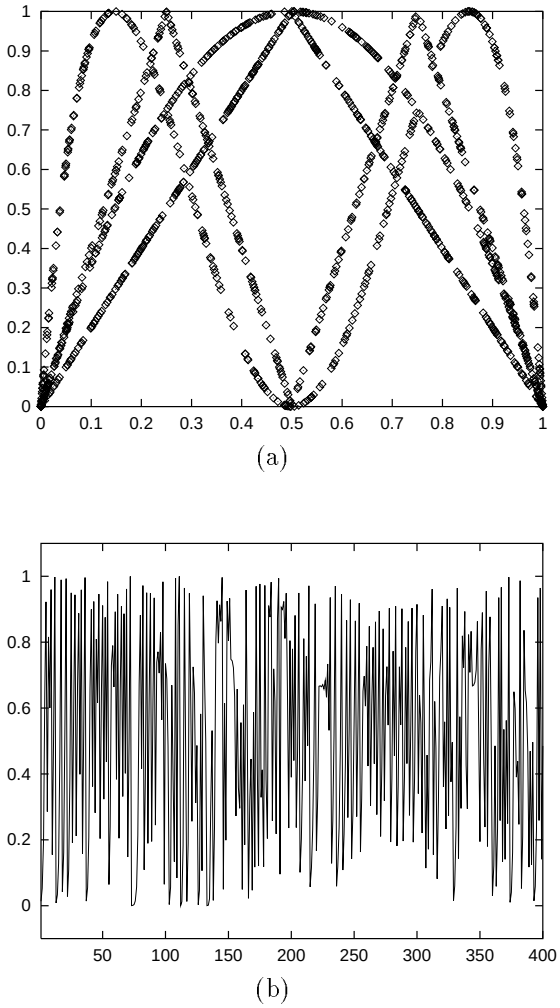



**Fig. 1** *(a) Training data drawn from four chaotic return maps, 300 points for each map. A new map is chosen after every 100 recursions. The first 400 values of the resulting time series are shown in (b).*

## 2. Unmixing of Experts

Data originating from different sources are subject to ambiguity. If input-output relations are considered, this can have at least two interdependent reasons. First, the input domains may overlap. However, it is impossible for a single network to map the same inputs to different outputs without using extra information (e.g. memory). Second, input *and* output of different sources can be identical for a subset of the data. In this latter case, information beyond the input-output pairs is required in order to reassign the data to the sources (for classification).

Let us discuss the first problem with an example of completely overlapping input domains. Consider the case of input-output pairs $(x_t, y_t) = (x_t, f_l(x_t))$, that are a random choice $l = l(t), l = 1,2,3,4$ of one of the four chaotic return maps $f_1(x) = 4x(1-x), x \in [0,1]$ ("logistic map"), $f_2(x) = \{2x$ if $x \in [0,.5)$ and $2(1-x)$, if $x \in [.5,1]\}$ ("tent map"), $f_3 = f_1 \circ f_1$ ("double logistic map") or $f_4 = f_2 \circ f_2$ ("double tent map"). $f \circ f$ denotes the iteration $f(f(x))$. If we set $x_{t+1} = y_t$, then we get a time series $\{x_t\}$ with $x_{t+1} = f_l(x_t)$, see Fig.1. When these maps are alternately used, a given input $x_t$ alone does not determine the appropriate output $y_t$, and a representation of the underlying relations must necessarily contain a division into subtasks. For such data sets, a gating network [4] that only depends on the input, is not helpful.

As a solution of this problem, we propose a set of predictors $\tilde{f}_i, i = 1,...,n$, which compete for the data. The optimal choice of function approximators $\tilde{f}_i$ depends on the specific application. For the present purpose we are using radial basis function networks (RBFN's) of the Moody-Darken type [10], because they offer a fast learning method. The error function

$$E_i = \sum_t \gamma_i^t \epsilon_i^t, \tag{1}$$

used for a gradient-descent, weights the errors of the individual predictors

$$\epsilon_i^t = \left(\tilde{f}_i(x_t) - f_{l(t)}(x_t)\right)^2 \tag{2}$$

by a simple Gaussian assumption for the error distribution (cf. [4, 16])

$$\gamma_i^t = \frac{e^{-\beta \epsilon_i^t}}{\sum_{j=1}^n e^{-\beta \epsilon_j^t}}. \tag{3}$$

The competition is controlled by the parameter $\beta$, similar to other training rules (cf. [4, 16]). A small value of $\beta$ implies that the predictors almost equally share the same data for training. Increasing $\beta$ enforces the competition, thereby driving the predictors to a specialization on different subsets of data. We would like to stress here, that one of the key ingredients of

our ACE algorithm is the adiabatic cooling schedule. Only, if the increase of competition is performed slowly and only after the error has settled (i.e. adiabatically), diversification will occur at particular "temperatures" $T = 1/\beta$ and the network parameters separate abruptly. These phase transitions lead to a significant decrease of the mean error $E = \sum_{i=1}^{n} E_i$ (see Fig.4(f)) and have been described within a statistical mechanics formalism [16]. In the following, the soft competition algorithm is always annealed completely ($T \to 0$). Note however, that it can also be appropriate to stop at a finite "temperature", e.g. when overfitting has to be avoided.

The case $\beta = \infty$ leads to hard competition [13, 14], where the error function of the $i$th predictor is denoted by

$$E_i = \sum_t \delta_i^t \epsilon_i^t \tag{4}$$

with

$$\delta_i^t = \begin{cases} 1 & : \text{ if } \quad \epsilon_i^t < \epsilon_j^t \quad \forall j \neq i \\ 0 & : \text{ otherwise.} \end{cases} \tag{5}$$

So, only the winner at time $t$ with lowest error $\epsilon_i^t$ is considered.

## 3. Inert Predictors

In the above section, we introduced a framework consisting of a set of predictors, which specialize during their competition for the data. However, certain ambiguities, inherent in the data, cannot be resolved so far. In the above example, such ambiguities emerge from the intersections of the four maps. Data points drawn from intersections can originate from different maps. In this case, a correct reassignment to the source requires additional information. Each intersection also gives the possibility of fitting a curve in four ways by a combination of the left and right branches (see Fig.1(a)) of the intersecting maps. A decision which branches "belong together", thereby uniquely identifying the original maps $f_1, ..., f_4$, also requires more information. Such information is implicitly contained in the data $\{(x_t, f_{l(t)}(x_t)\}, t = 1, ..., T$, if the system does not switch its state $l(t)$ every time step, i.e. if the underlying system has memory.

We here consider only the simple case of memory that is present, when the system switches randomly among different subsystems $i, j$ at low rates $r_{ij} < r$. We do not take into account the information contained in the statistics of the transitions, which in principle is possible in terms of a hidden markov model [17]. This limits the scope of our model, but it also gives our method the important advantage to yield reliable results already with very *small* amounts of training data.

For the derivation of our training scheme we again assume that the Gaussian assumption for the error distribution holds. Then the probability that a given part $\{x(t), y(t)\}$, $t = t_0 - \Delta, \ldots, t_0 + \Delta$ of the time series is generated by a particular sequence $\vec{s}_\Delta = (s_{t-\Delta}, \ldots, s_{t+\Delta})$ of functions $f_{s_t}$ is given by

$$p_{\vec{s}_\Delta}^t \propto \prod_{\tau=-\Delta}^{\Delta} \gamma_{s_{t+\tau}}^{t+\tau}. \tag{6}$$

Since we also assume a small bound $r$ on the switching rates, we can neglect the sequences where switchings occur (i.e. where not all $s_{t+\tau} = i$), if we choose $\Delta$ appropriately as a function of $r$. Then, the probability that the data pair $(x(t), y(t))$ is generated by $f_i$ is equivalent to the probability to be in the corresponding sequence:

$$p_i^t \propto \prod_{\tau=-\Delta}^{\Delta} \gamma_i^{t+\tau} \tag{7}$$

and we have by normalization

$$p_i^t = \frac{e^{-\beta \sum_{\tau=-\Delta}^{\Delta} \epsilon_i^{t-\tau}}}{\sum_{j=1}^{n} e^{-\beta \sum_{\tau=-\Delta}^{\Delta} \epsilon_j^{t-\tau}}}. \tag{8}$$

This means, that we can simply use the low-pass filtered errors instead of the raw errors $\epsilon_i^t$ in order to include memory. From the statistics point of view, the use of Eq.(8) corresponds to testing the assumption made on the probability distribution of the transitions, respectively on the length of staying in one dynamics. In the above case, we assume for the simplicity of the analysis that periods of a certain length exist where no switching occurs, i.e. the switching probability is 0. Outside these intervals the probability is 1. In most pratical situations, the probability to stay in the same dynamics will decrease exponentially or with a Poisson or Gamma distribution. For these more general assumptions on the distribution of the switching probabilities, a weighted filter $G^\dagger$ other than the box function has to be introduced in Eq.(8).

$$p_i^t = \frac{e^{-\beta \int G_{\{i\}}^{t-t'} \epsilon_i^{t'} dt'}}{\sum_{j=1}^{n} e^{-\beta \int G_{\{j\}}^{t-t'} \epsilon_j^{t'} dt'}}. \tag{9}$$

Yet, without any knowledge about the characteristics of the time series, the assumption used in equations (7) and (8) seems to be the simplest and at the same time computationally least expensive. Heuristically, Eq.(8) is analogous to evolutionary inertia, since

$^\dagger$The dependence of the kernel $G$ on previous assignments to predictors $\{k\}$ opens the possibility to include higher order information, as e.g. forbidden and/or probable transitions. Different kinds of filter functions as e.g. Gaussians or sombrero-shaped kernels can be chosen for $G$ in order to reflect the inertia, respectively the statistics.

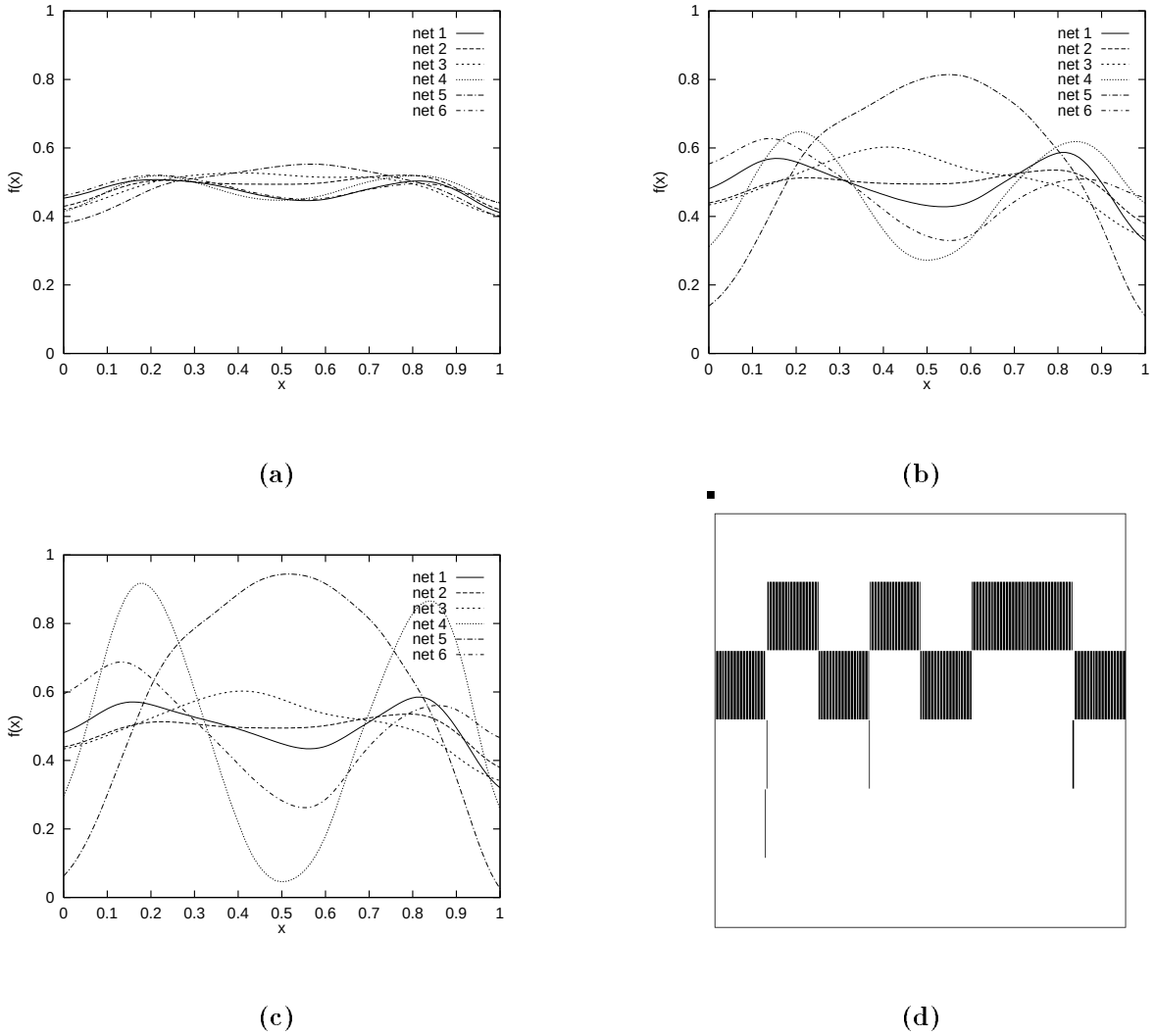



**Fig. 2** *Illustration of the learning process in the one-dimensional case (sine and double-logistic) with additive noise ($\sigma = 0.1$). Clearly this data cannot be represented by a single map. The time series from the two maps above alternates after every 50 steps (except at time $t = 300$). Snapshots of the functions after 1, 2 and 3 steps respectively are shown. (a) After the first training pass some competitors have already achieved advantages in the selection process. (b) and (c) show subsequent steps, where only winners are allowed to improve their approximation quality. Note that the networks who die in the course of the competition are not changing their map anymore. (d) Winner distribution (perfect segmentation) of six predictors after ten iterations of the competition. For each time step a bar indicates which network has won the competition on the time series data.*

once a predictor has performed better than its competitors, it also has an advantage for temporally adjacent data points. Similar as before, training is finally accomplished by a gradient-descent of

$$E_i = \sum_t p_i^t \epsilon_i^t. \tag{10}$$

In the hard competition limit, using eqs.(8) and (10) we arrive at a simple moving average of the winner $i$ at times $t$

$$E_i^t = \sum_{\tau=-\Delta}^{\Delta} \epsilon_i^{t-\tau}, \tag{11}$$

which is summed to become

$$E_i = \sum_t \delta_i^t E_i^t, \tag{12}$$

with

$$\delta_i^t = \begin{cases} 1 & : \text{ if } \quad E_i^t < E_j^t \quad \forall j \neq i \\ 0 & : \text{ otherwise.} \end{cases} \tag{13}$$

So in the hard competition case, the $E_i^t$ are compared in order to determine the winners for times $t$, and only the winners are finally trained on the corresponding parts of the data. The hard competition training is started with identically initialized predictors. For the first iteration, the training set is divided into subsets of equal size, so that each predictor gets its own subset of data to train on. We consider this a fair way to get distinctive predictors for the competition. In the second and each following training pass a hard competition is carried out: Only the predictor with the smallest $E_i^t$ is allowed to train on the pattern at times $t$. As a consequence, unnecessary predictors drop out of the learning process, since they don't win any data after some time.

## 4. Analysing Switching Dynamics

In the following section, the performance of both methods, hard and soft competition, are illustrated with one-dimensional chaotic maps, with the Mackey-Glass equation (a model for blood cell regulation [9]) and with speech data.

### 4.1 Hard Competition

Two points are important in order to find a good solution with hard competition. First, a suitable initialization of the predictors has to be chosen, which is sometimes difficult. Second, the maps to be identified should not be as similar as the logistic map and the tent map in our example (cf. section 2.). With both points kept in mind, we can solve the following three tasks.

#### 4.1.1 One–dimensional Chaotic Maps

In the one-dimensional example, an ensemble of six competing predictors is applied to a time series from two maps, the sine map $f_1(x) = \sin(\pi x)$, and the double logistic map $f_2(x) = f_l(f_l(x))$, $f_l(x) = 4x(1-x)$. We use a time series where the dynamics alternates every 50 time steps except for time $t = 300$. Furthermore Gaussian noise ($\sigma = 0.1$) is added to the signal to hide its deterministic nature. There is no way to include the overall dynamics of this example into a single map; a single approximator would predict an average between the two maps. Higher order statistics is not present in the sequence and we can therefore neglect the dependence of $G$ on predictor assignments $\{k\}$ in eq.(9) and use the simple moving average $E_i^t$ from eq.(11). In this rather simple one-dimensional case, we can easily illustrate the competitive learning process of our algorithm. In figures 2(a)-(c) we show snapshots of the functions after 1, 2 and 3 steps, respectively. After the first training pass the initial symmetry of the networks is broken and some competitors have already achieved advantages in the selection process (cf. 2(a)). Their maps are already a fair approximation to $f_1$ or $f_2$ respectively. The subsequent steps (cf. 2(b) and (c)) show the struggle for data between all competing networks, where only winners are allowed to improve their approximation quality on their share of the training set. The networks who die in the course of the competition will not change their map anymore.

The results clearly indicate that the *unsupervised* segmentation (cf. 2(d)) identifies the underlying dynamics almost perfectly, i.e. it is found out *that* and *which* two systems underlie the observed signal, and it is determined *where* the corresponding dynamics are switched.

#### 4.1.2 The Mackey-Glass Equation

The method can be applied to time series from high-dimensional chaotic systems simply by replacing the scalar input $x$ by a vector, which is obtained by the method of time-delay embedding of the time series [7] and by a corresponding adaptation of the networks (without change of notation). As an example for a high-dimensional chaotic system, we take the Mackey-Glass delay-differential equation

$$\frac{dx(t)}{dt} = -0.1x(t) + \frac{0.2x(t-t_d)}{1+x(t-t_d)^{10}}, \tag{14}$$

originally introduced as a model of blood cell regulation [9]. We generated a time series of $N = 400$ points where we switched the delay parameter $t_d$. For the first and last 100 samples (sampling rate $\tau = 6$) we chose $t_d = 17$, whereas for the second 100 samples we used $t_d = 23$ and for the third $t_d = 30$. To increase the difficulty of the problem, 5% noise was added at each

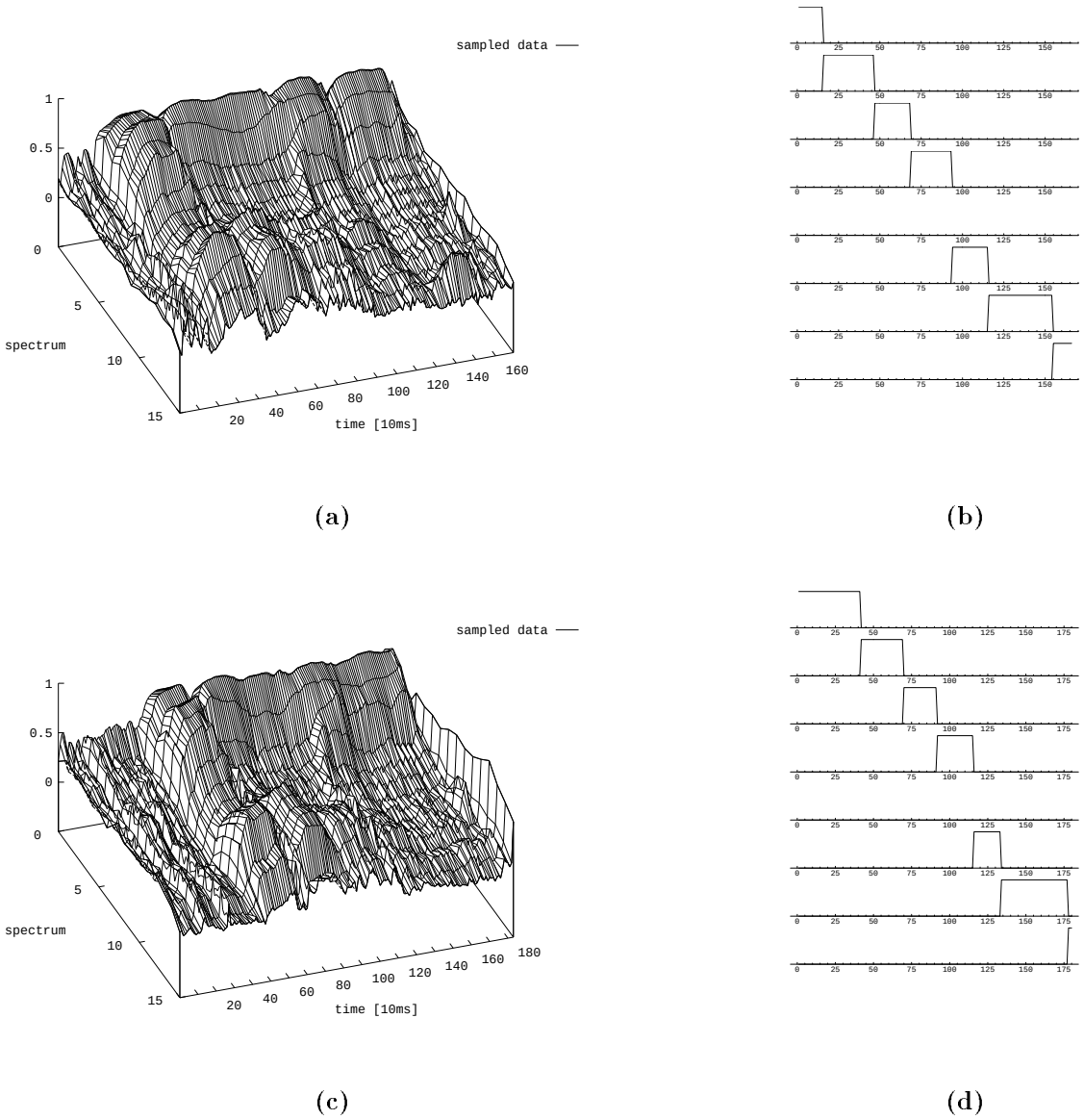



**Fig. 3** *The 16 dimensional mel-scale FFT coefficients (spectrum) of the continuously spoken vowels AEIOU are plotted over time [10ms] (samples (a) and (c)). (b) Winner distribution on the time series from sample (a) after 20 iterations of the competition. Note the clear segmentation for the shown sample. The resulting networks found on sample (a) are tested on sample (c) and the resulting winner distribution is shown in (d) (Generalization test). Note again a clear segmentation for sample (c), although the networks have only been trained on sample (a).*

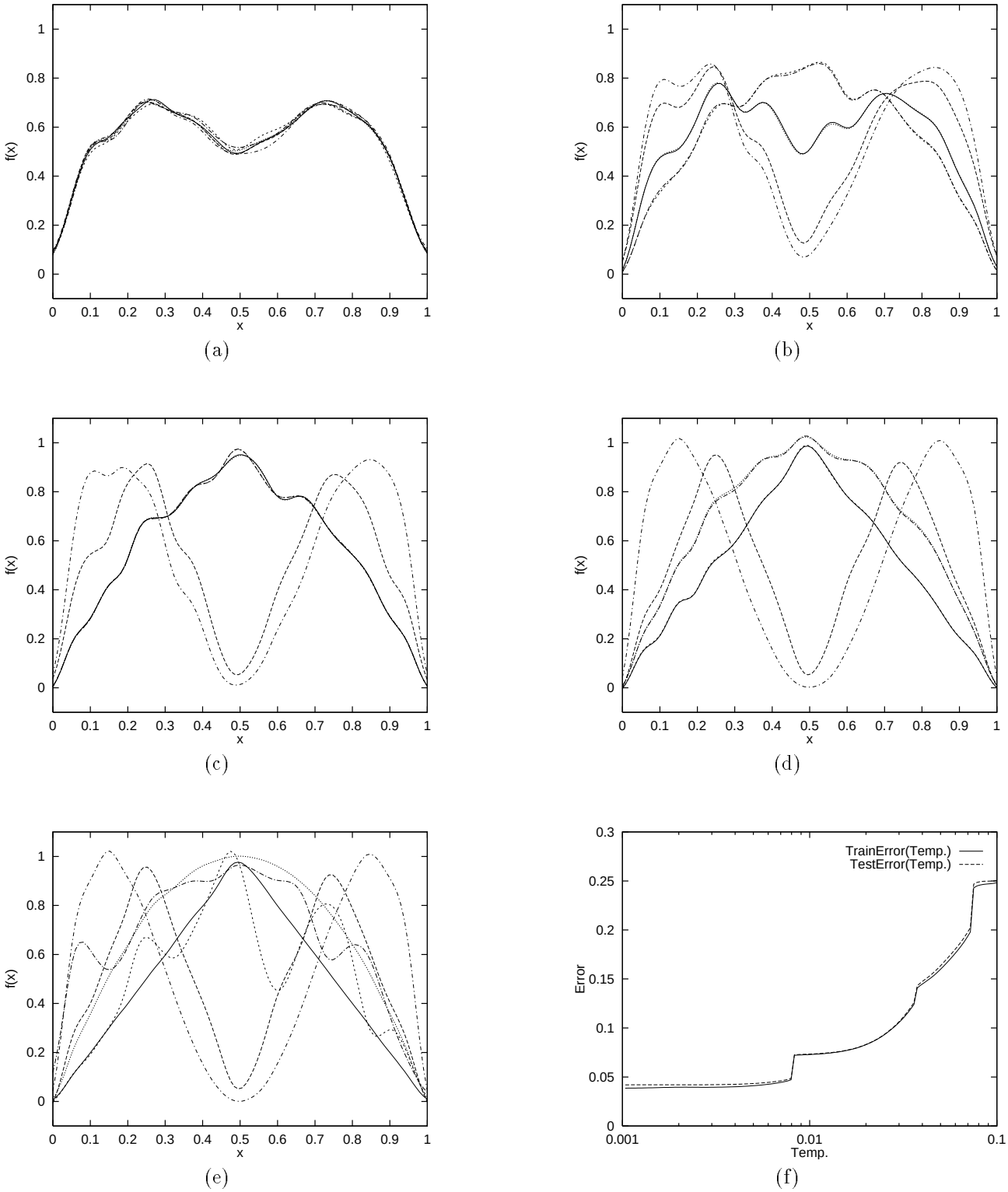
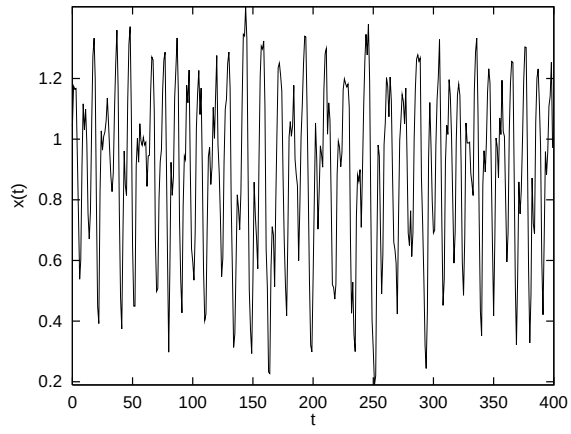



**Fig. 4** *Shown are the maps that have been learned by the predictors, (a) before the first and (b)-(d) after each of three phase transitions. (e) The maps learned by the RBFN's at the end of the process. Four nets have specialized on each of the given dynamics, while two nets dropped off and finally did not contribute to the segmentation and the overall error E. (f) Training and test error (Root mean square error) during the annealing process both indicate phase transitions.*

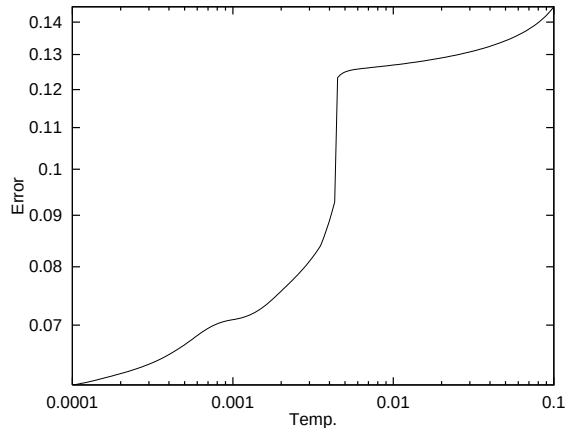
integration step, thereby turning the system stochastic (Fig.5(a)). For the creation of a training set out of this time series, an embedding dimension $m = 6$ was used (cf. [11]).

The width of the moving average parameter $\Delta$ (in eq.(11)) is successively increased while the training proceeds, in order to force the six networks from some initial coexistence to a stronger competition in the end. The distribution of winners on the training set after ten iterations shows a perfect segmentation [13, 14].

In both chaotic cases discussed, only two (rsp. three) predictors survive, each one being able to predict one of the two (rsp. three) chaotic systems represented by the data.




(a)




(b)

**Fig. 5** *(a) A noisy Mackey-Glass time series that includes 3 different dynamics was used for the segmentation task. (b) Adiabatic evolution of the training error for the Mackey-Glass data.*

#### 4.1.3 Dynamics of FFT Speech Data

A well–known example of non–stationary dynamics in the real world is speech. The speech data to be segmented are 16 dimensional mel-scale FFT coefficients, obtained from continuously spoken vowels (AEIOU, single speaker) at a sampling rate of 16kHz. Eight predictors (30 hidden units) are initialized and our hard competition algorithm is performed on a $16\times4$ dimensional input vector, since an embedding of $m = 4$ is chosen. Thus, past information in the time series is incorporated. Both the learning rate and $\Delta$ are successively increased during training. In Fig. 3(b) we see an impressingly accurate segmentation for sample 3(a). After training, five networks represent A,E,I,O,U, respectively, while two nets specialize on silence and one network dies. As a test of the generalization ability, we use these nets as predictors for the speech dynamics of the unknown data set 3(c) (same speaker). We observe a clear segmentation for this and other unknown samples, which shows that our method can indeed detect the vowel dynamics in an *unsupervised* manner and generalize properly. (Note that the testing does not involve additional learning.) In fact, we achieved a perfect generalization, similar to Fig. 3(d), in 14 cases of totally 20 AEIOU samples. All 20 samples are continuously spoken, yet the speed and emphasis is very different. The 6 erroneous generalizations exhibit just a single misidentified subsegment. In most cases this was due to the similarity of E and I, resp. O and U. Especially those samples, where silence is present between the single vowel utterances, show such a generalization error, when tested with the network configuration of sample 3(a).

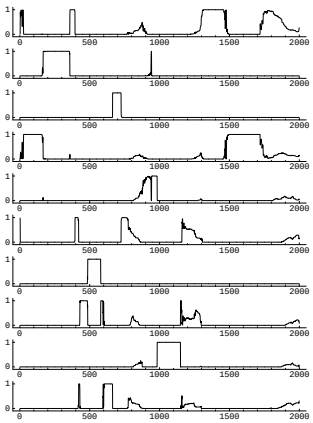
### 4.2 Soft Competition

If the mappings are too similar (see Fig.1(a)), the final result can depend on the choice of initial parameters and a mixing of maps can occur. We solve this initialization problem by adiabatically increasing the degree of soft competition.

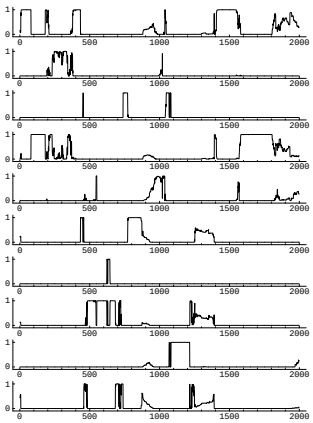
#### 4.2.1 Identification of Switching Chaos

We illustrate this approach with the example of four very similar chaotic maps, which produce a time series of $N = 1200$ points (Fig.1). The maps were alternated at random every 100 iteration steps. Because these maps are ergodic on the support $x \in [0, 1]$, they cannot be distinguished on the basis of their arguments alone. Furthermore, the small switching rate $r_s = 1/100$ guarantees a large probability for short sequences of e.g. length $l = 7$ to contain no alternations of the underlying system, which justifies our simple method of taking memory into account by setting $\Delta = 3$ in Eq.(8).

As in section 4.1.1 we used 6 radial basis function networks of the Moody-Darken type [10] as predictors and decreased the temperature $T = 1/\beta$ adiabatically, i.e. the next smaller value of the temperature is taken, when the overall error $E$ had saturated. The result is shown in Fig.4. The error decreases most during phase transitions (Fig.4(f)) which occur when the different underlying dynamics abruptly become resolved to more detail (Fig.4(a)-(d)). After the relevant structures have been found by the algorithm, no further phase transitions occur and there is only little further decrease of the error when $T$ approaches zero. At $T \simeq 0$, we found that four networks segmented the time series almost exactly at the switching points, which implies that the underlying systems were unambiguously identified.



(a)



(b)

**Fig. 6** *Segmentation $p_i^t$ of the continuously spoken phrase "to be or not to be" with ten competing networks (time scaled from 0 to 2000), (a) for the training sequence, (b) for a test sample. Both instances of the phrase yield a similar segmentation profile, that might be used for an ensuing classification process.*

### 4.2.2 The Mackey-Glass Equation

We use the same setting as in section 4.1.2. During training, two phase transitions occurred (Fig.5(b)), indicating that the system detected the different dynamical systems. The second transition (at $T \approx 0.0007$) becomes more prominent when simpler networks are used. However, this leads to sub–optimal prediction results and was therefore not applied. The removal of three nets, at $T \simeq 0$, did not increase the error significantly, which correctly indicates that three predictors completely describe the source. Segmentation, finally, was perfect.

The performance (convergence speed, segmentation accuracy) of our approach with the high-dimensional Mackey-Glass data was even better than for the one-dimensional maps, which indicates that in higher dimensions segmentation and identification can be easier.

### 4.2.3 Dynamics in raw Speech Data

The soft competition approach can also successfully analyse the vowel data set from section 4.1.3. Another application of our method, using *raw* speech data, might be interesting, since it is quite different from traditional (supervised) classification with hidden markov models (HMM, [18]) or time-delay neural networks (TDNN, [3]). Instead of using any preprocessing, e.g. a fast fourier transform, we just fed the plain A/D-converted signal into the networks, thereby avoiding the loss of possibly relevant information in the preprocessing stage.

We tested the ability of our method to capture the dynamics of continuously spoken words with the phrase "to be or not to be". The text was recorded multiple times (single speaker) on a Sun workstation with an ordinary microphone (sampling rate 8 kHz, resolution 8-bit). We used 10 RBF-networks to compete for the dynamics of only *a single instance* of the phrase. Each network used 10 successive values of the signal to predict the next sampled amplitude ($m = 10$). To further simplify the computation, we reduced the amount of training data from 16000 to 2000 by taking just every 8th training pattern. The annealed competitive learning was performed on the training phrase during 1840 iterations and the resulting segmentation of the phrase is shown in Fig.6(a). The prediction of "to be", that occurs twice in the phrase, is performed by the nets 1,2 and 4 (counted from the top), whereas the prediction of "or not" is distributed among the seven remaining nets. Although both "to be"s are predicted by the same three nets, a specialization on phonemes, which might have been expected, did not happen. Thus, one cannot take out a single net for the recognition of a certain phoneme, at least when using radial basis function networks as models for the phoneme dynamics. Neverthe-

less, when a different instance of the phrase is presented to the ensemble of trained predictors, a similar segmentation is performed (Fig.6(b)). These results suggest, that a word recognition might be possible with the architecture of competing predictors by a classification of the segmentation profiles. This, however, remains to be shown.

### 4.3 Comparing Hard and Soft Competition

From the algorithmic point of view, hard competition is much more efficient than the more advanced ACE algorithm. On the other hand, both the initialization and the similarity problem are solved with the annealed competition of experts algorithm. For small $\beta$, the predictors almost equally have access to all data and a random initialization is sufficient. Increasing $\beta$ adiabatically enforces the competition, which leads to a specialization on different subsets of the data. At particular values of $\beta$, the networks separate and give a more and more refined picture of the underlying dynamical systems. These phase transitions correspond to bifurcation points, where similar dynamics bifurcate, i.e. their difference is detected.

Both the hard competition and the soft competition ansatz have certain advantages, and it depends on the application, which method can be used more effectively.

We favour hard competition for applications, where the dynamical systems involved have different characteristics. In this case, the final segmentation results can be achieved quickly and will not sensitively depend on the choice of initial parameters.

## 5. Summary and Outlook

We presented a framework for the *unsupervised* segmentation of time series. It applies to systems with a nonstationary switching dynamics, a phenomenon which is observed in many natural signals, as e.g. in speech and in brain data [12]. The method is based on hard or soft competition together with a tendency to take advantage of neighborhoods in time (evolutionary inertia). In the soft competition case, the system undergoes a series of phase transitions during the annealing process, where similar dynamics bifurcate. We applied this general idea to time series from alternating maps, switching differential equations and speech data. It was demonstrated that our approach leads to nearly perfect segmentation even in the presence of noise. Note that the goal at this point of our study is not to convince the reader to use our unsupervised segmentation method in his speech recognition system, but to demonstrate its remarkable universality and simplicity on a number of different applications with switching dynamics.

We recently used the ACE method for the prediction of Data Set D from the Santa Fe Time Series Competition [19]. A quantitative comparison with the winners of the Santa Fe Competition, Zhang and Hutchinson [19, pp. 219–241], exhibits a 10% better performance of our method, with much less computational effort (for details see [15]).

Future work will be dedicated to the adaptation of time-delay networks or hidden markov models to our framework.

**Acknowledgement**: K.P acknowledges support of the DFG (grant Pa 569/1-1) and K. -R. M. acknowledges financial support by the European Communities S & T fellowship under contract FTJ3-004. We thank Prof. Waibel's group for supplying the preprocessed speech data set and for valuable discussions and help.

**References**

[1] Amari, S., Mathematical Foundations of Neurocomputing, Proc. of the IEEE, Vol.78, No.9, 1443 (1990)

[2] Rumelhart, D.E., McClelland, J.L., *Parallel distributed processing*, MIT Press, Cambridge Massachusetts (1984).

[3] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. and Lang, K., Phoneme recognition using time-delay neural networks, IEEE int. conf. on acoustics, speech and signal processing (1989).

[4] Jacobs, R.A., Jordan, M.A., Nowlan, S.J., Hinton, G.E., Adaptive Mixtures of Local Experts, Neural Computation **3**, 79-87 (1991).

[5] Schuster, H.G., *Deterministic Chaos*, 2nd Edition, Physik Verlag, Weinheim, (1988).

[6] Kaneko, K., Chaotic but regular posi-nega switch among coded attractors by cluster-size variation, Phys. Rev. Lett. **63**, 219 (1989).

[7] Takens, F., Detecting strange attractors in turbulence, in: Rand, D., Young, L.-S., (Eds.), *Dynamical Systems and Turbulence*, Springer Lecture Notes in Mathematics, **898**, 366 (1981).

[8] Liebert, W., Pawelzik, K., Schuster, H.G., Optimal embeddings of chaotic attractors from topological considerations, Europhys. Lett. **14**, 521 (1991).

[9] Mackey, M., Glass, L., Oscillation and chaos in a physiological control system, Science **197**, 287 (1977).

[10] Moody, J., Darken, C., Fast Learning in Networks of Locally-Tuned Processing Units, Neural Computation **1**, 281-294 (1989).

[11] Casdagli, M., Nonlinear Prediction of Chaotic Time Series, Physica D **35**, 335-356 (1989).

[12] Pawelzik, K., Bauer, H.-U., Deppisch, J. Geisel, T., How oscillatory neuronal responses reflect bistability and switching of the hidden assembly dynamics, NIPS 92, Morgan Kauffmann (1993).

[13] Kohlmorgen, J., Müller, K.-R., Pawelzik, K., Competing predictors segment and identify switching dynamics, in ICANN'94: Proc. of the int. conf. on Artificial Neural Networks, M.Marinaro, P.Morasso (eds.), Springer London, 1045-1048 (1994)

[14] Müller, K.-R., Kohlmorgen, J., Pawelzik, K., Identification of Switching Dynamics with Competing Neural Networks, in NOLTA 94: Kagoshima Symposium on Nonlinear Theory and its Applications, 283-287 (1994)

[15] Pawelzik, K., Kohlmorgen, J., Müller, K.-R., Annealed Competition of Experts for a Segmentation and Classifi-

cation of Switching Dynamics, to appear in *Neural Computation* (1995)

[16] Rose, K., Gurewitz, E., Fox, G. (1990). Statistical Mechanics and Phase Transitions in Clustering. *Phys. Rev. Letters*, Vol. 65, 945–948, 1990.

[17] Cacciatore, T.W., Nowlan, S.J., Mixtures of Controllers for Jump Linear and Non-linear Plants, NIPS 93, Morgan Kauffmann (1994).

[18] Rabiner, L.R. (1988). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Readings in Speech Recognition*, ed. A. Waibel, K. Lee, 267–296. San Mateo: Morgan Kaufmann, 1990.

[19] A.S. Weigend and N.A. Gershenfeld (Eds.), *Time Series Prediction: Forecasting the Future and Understanding the Past*, Santa Fe Institute Studies in the Sciences of Complexity. Addison-Wesley, 1994.