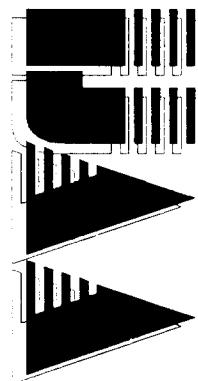


INTERNATIONAL JOURNAL

HEFT 1

Jg. 4, (1996)

AUTOMATION AUSTRIA



INHALT

	Seite
HOFBAUER, A.; HEISS, M. The Origin of Spikes During Online Adaptation of Membership Functions	1
FESSEL, C. Low Cost Bildverarbeitung und ihre industrielle Anwendung	15
SCHRÖDL, M.; SCHERMANN, M. Fully Digital Induction Motor Drive for Positioning Applications Without Measuring Electrical Quantities	26
WEINMANN, A. Neural Networks. Algorithms for Learning and Strategies of Application	40
SCHLAGLICHTER	57
DISSERTATIONSKURZFASSUNGEN	61
BUCHBESPRECHUNGEN	63
VORSCHAU	66
EIN INSTITUT STELLT SICH VOR Institut für Handhabungsgeräte und Robotertechnik der Fakultät Maschinenbau an der TU Wien	67
AUSSCHREIBUNG DES FRED MARGULIES PREISES	70
VERANSTALTUNGSKALENDER	71

The Origin of Spikes During Online Adaptation of Membership Functions

Andreas Hofbauer und Michael Heiss, Wien*

Manuscript received: March 31, 1995

Abstract

The adaptation of membership functions in a fuzzy system is a nonlinear optimization problem. Thus, the convergence of online learning algorithms is questionable. We demonstrate the convergence problems by analyzing two types of spikes, the narrow basis function spikes and the non-monotonic basis function spikes, which can occur during the online adaptation. Further, we show how these spikes can be avoided by restricting the parameter variations of the widths and the distances of the membership functions. According to these restrictions we have to conclude that in most cases it is better solely to adapt the rule conclusions than to adapt the membership functions.

Keywords Fuzzy Control, Learning Control, Basis Functions, Spikes, Back-Propagation Algorithm.

1 Introduction

Adaptive fuzzy controllers are able to control slowly time-varying nonlinear systems. The most popular parameters for adaptation are: the rule conclusions, the widths of the input membership functions and the locations of the input membership functions.

The adaptation of the rule conclusions is a linear adaptation when Takagi and Sugeno's type of fuzzy system [17] with linear rule conclusions [9, 10] is used (Section 2). Convergence proofs for these linear adaptations are available. Nevertheless from an approximation theory point of view an additional adaptation of the locations and the widths of the membership functions could result in a better approximation of the desired input-output behavior of the fuzzy controller. Some of the literature in this field gives the impression that "you can simply take the back-propagation algorithm and everything would be

* Univ.-Doz. Dipl.-Ing. Dr.techn. Michael Heiss was with the University of Technology Vienna and is now with SIEMENS AG Austria, PSE NLT ECANSE, Gudrunstr. 11, A-1100 Wien, Internet: <http://www.siemens.at/~ecanse/ecanse.html>, E-mail: michael.heiss@siemens.at. Dipl.-Ing. Andreas Hofbauer, Goethestr. 16, A-3910 Zwettl. This work was supported by the Austrian "Fonds zur Förderung der wissenschaftlichen Forschung" under contract P8914-TEC. The paper was presented at the *International Symposium on Fuzzy Logic (ISFL'95)*, ETH Zürich, 1995.

fine”, no matter whether the adaptation is performed online or offline. Nevertheless it is known that the adaptation of the nonlinear parameters is a multi-dimensional nonlinear optimization problem which is very difficult to solve, even offline (Section 3).

In our paper we analyze why spikes occur in the input-output characteristic of the fuzzy controller when an online¹ adaptation of membership functions is realized and show that such an adaptation works properly only for some “well behaved” applications. We concentrate in our analysis on Gaussian membership functions, but similar results would have been accomplished with B-Splines or other smooth “hill”-shaped membership functions. Two types of spikes are investigated, called *Non-Monotonic Basis Function Spikes* and *Narrow Basis Function Spikes* (Section 4).

Non-Monotonic Basis Function Spikes: According to [11, 17] we interpret the defuzzification as a normalization of the input membership functions. The multiplication of the normalized membership functions yields the fuzzy basis functions [19]. Based on this interpretation we show that even if the membership functions are monotonic, the corresponding fuzzy basis functions can be non-monotonic due to the adaptation. This violation of monotony can cause spikes in the input-output characteristic. Limits for the distances and widths of the membership functions are derived such that this effect does not occur.

Narrow Basis Function Spikes: Another reason for spikes is membership functions that become thinner if adaptations take place only in the local neighborhood of the membership function, and not near the center of this membership function. Once the membership function is very thin, the probability of an update of this membership function becomes very small.

These undesired properties of the adaptation due to the “short sightedness” of the back-propagation algorithm are illustrated with examples (Section 5).

Due to the complexity of the problem it is most likely impossible to provide a general adaptation algorithm with guaranteed convergence. A “nice” convergence behavior is achieved when the variation ranges of the nonlinear parameters are limited to small variations. On the other hand, if only small variations are allowed, it may need less memory to use more densely placed membership functions with fixed nonlinear parameters and to learn only the remaining linear parameters. This rentability check of the improved approximation quality versus the required additional effort (Section 6) leads us to the conclusion that we cannot recommend the learning of nonlinear parameters (locations and widths) of membership functions (Section 7).

2 Fuzzy System

Every controller can be seen to be equivalent to an input-output map, if the controller determines the control output $y(\mathbf{x})$ from n inputs x_i (e.g. state space variables) in a deterministic and memoryless way. The good old PID-controller, a feedforward neural network [4], or a fuzzy controller [1, 11] can be interpreted to be simply a different representation of input-output maps (Fig. 1).

¹The term *online* is used in the sense of typical online applications where the learner plays no role in obtaining information about the desired input-output characteristic, but is just collecting and processing input-output pairs at the current operating point.

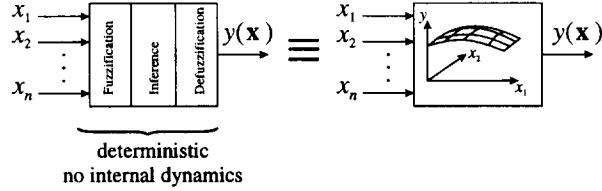


Fig. 1: The input-output behavior of a fuzzy controller is equivalent to an input-output map.

As long as linear controllers (like the PID) are considered we gain no additional insight when interpreting them as mappings. As soon as nonlinear controllers like the fuzzy controllers are considered, the map based interpretation can be helpful. In [12, 13] we analyzed how to set up a fuzzy system in order to get a smooth input-output mapping which properly represents the rule base of the N rules²

$$R^i : \text{IF } (x_1 = X_1^i) \text{ and } (x_2 = X_2^i) \text{ and } \dots (x_n = X_n^i) \text{ THEN } (y = C^i), \quad \forall i \in \{1, \dots, N\} \quad (1)$$

where each linguistic term X_l^i is defined by a membership function $\mu_l^i(x_l)$. The linguistic term C^i for the rule conclusion is defined by a singleton-shaped membership function at $y = c_i$. Let $\mathbf{x} = (x_1 \ x_2 \ \dots \ x_m)^T$ and define the multiplication as **and**-operator, then the premise can be calculated as

$$\mu_i(\mathbf{x}) = \mu_1^i(x_1)\mu_2^i(x_2)\cdots\mu_n^i(x_n). \quad (2)$$

Output singletons c_i and the center-of-gravity defuzzification [2] (also called center average defuzzifier [18] or weighted average) yield the fuzzy controller output ([17], [18, p. 4], [2, p. 187])

$$y(\mathbf{x}) = \frac{\sum_{i=1}^N c_i \mu_i(\mathbf{x})}{\sum_{i=1}^N \mu_i(\mathbf{x})}. \quad (3)$$

Defining the *fuzzy basis functions* [19]

$$b_i(\mathbf{x}) = \frac{\mu_i(\mathbf{x})}{\sum_{j=1}^N \mu_j(\mathbf{x})} \quad (4)$$

allows us to represent the fuzzy controller output as a linear combination

$$y(\mathbf{x}) = \sum_{i=1}^N c_i b_i(\mathbf{x}) \quad \forall \mathbf{x} \in \mathbb{R}^n \quad (5)$$

of fuzzy basis functions $b_i(\mathbf{x})$ (Fig. 2). This fuzzy system is a special type (constants as rule conclusions) of Takagi and Sugeno's fuzzy system [17]. The basis function representation was introduced as *standard additive model* by Kosko 1992 [11]. The name *fuzzy basis function* was introduced by Wang 1992 [19].

²The index i denotes throughout the whole paper the number of the rule. In order to avoid an index confusion we use the superscripted index i in X_l^i , C^i , $\mu_l^i(x_l)$ whereas we use a regularly placed index for $\mu_i(\mathbf{x})$, c_i , $b_i(\mathbf{x})$, ...

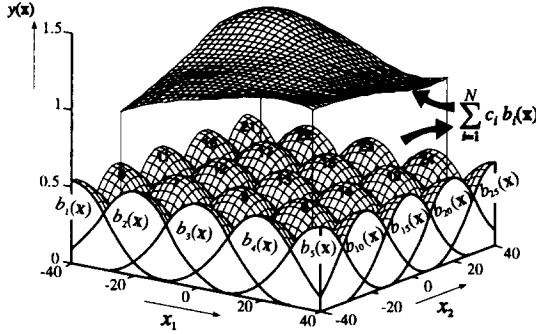


Fig. 2: The output of the fuzzy controller $y(\mathbf{x})$ is the weighted (c_i) sum of the fuzzy basis functions $b_i(\mathbf{x})$. Each basis function represents a rule and the height c_i is the rule conclusion.

According to Eq. (5) the input-output characteristic is smooth as long as the basis functions $b_i(\mathbf{x})$ are smooth.

When discussing in the following sections the problems of adapting membership functions, these problems can even be demonstrated for the one-dimensional case ($n = 1$). Thus, for simplicity we restrict the discussion to the one-dimensional case.

We use Gaussian membership functions [18], [16]

$$\mu_i(x) = \exp\left[-\frac{(x - \bar{x}_i)^2}{\sigma_i^2}\right] \quad \forall i \in \{1, \dots, N\} \quad (6)$$

where \bar{x}_i is the location of the center (mean value) and σ_i is a measure for the width (standard deviation) of the membership function. Similar results would have been achieved for other smooth membership functions.

3 Adaptation

The fuzzy model Eq. (5) is linear in its rule conclusions c_i . We could therefore adapt these rule conclusions by one of the standard learning algorithms like the LMS algorithm [20], the back-propagation algorithm [15] or the gradient descent algorithm [1], all resulting in the same adaptation rule

$$c_i^{new} = c_i^{old} + \lambda e(\xi) b_i(\xi), \quad (7)$$

where λ is the learning rate, $x = \xi$ is the location where the update is performed and

$$e(\xi) = y_d(\xi) - y(\xi) \quad (8)$$

is the deviation between the desired output $y_d(\xi)$ and the actual output $y(\xi)$ of the fuzzy controller [6]. The convergence of algorithm (7) has been proved under real world conditions [1, 5].

Nevertheless from an approximation theory point of view it would be preferred to adapt all three parameters c_i , \bar{x}_i , and σ_i in order to be flexible enough to describe the

desired input-output characteristic $y_d(x)$ of the fuzzy controller with a minimum number of rules as accurately as possible. Note that the adaptation of nonlinear parameters can be seen as a multi-dimensional nonlinear optimization problem which is known to be very hard to solve, even offline.

If the learning is performed offline (Brown and Harris call it: *batch learning* [1]) the update locations ξ can be chosen to be evenly distributed over the whole input space. In this case there is a good chance but no guarantee for the convergence of the back-propagation algorithm [15].

On the contrary, if the learning must be performed online during the regular operation of the system (Brown and Harris call it: *instantaneous learning*), the choice of the update locations ξ cannot be influenced. Thus, the learning algorithm is in some way “short sighted” on the actual operating point ξ and has no possibility to overview the global situation.

This short sightedness causes no major troubles in the linear case as the global error surface is convex, but causes serious troubles if the global error surface has many local minima.

In the following we show how we apply the classical back-propagation algorithm to learn the parameters c_i , \bar{x}_i , and σ_i and how unsatisfying the convergence behavior can be, even for a trivial looking example.

3.1 Back Propagation Algorithm

With the local error function

$$I(\xi) = \frac{1}{2}e^2(\xi) \quad (9)$$

and the gradient descent learning rule³

$$p^{new} = p^{old} - \lambda \frac{\partial I}{\partial p} \quad (\lambda > 0) \quad (10)$$

applied to the parameters $p = c_i, \bar{x}_i, \sigma_i$, the learning rules

$$c_i^{new} = c_i^{old} + \lambda e(\xi) b_i(\xi) \quad \forall i \in \{1, \dots, N\} \quad (11)$$

$$\bar{x}_i^{new} = \bar{x}_i^{old} + \lambda e(\xi) b_i(\xi) \frac{\xi - \bar{x}_i}{\sigma_i^2} 2 [c_i - y(\xi)] \quad \forall i \in \{2, \dots, N-1\} \quad (12)$$

$$\sigma_i^{new} = \sigma_i^{old} + \lambda e(\xi) b_i(\xi) \frac{(\xi - \bar{x}_i)^2}{\sigma_i^3} 2 [c_i - y(\xi)] \quad \forall i \in \{1, \dots, N\} \quad (13)$$

are obtained, where \bar{x}_i and σ_i must be restricted to $\bar{x}_i < \bar{x}_{i+1}$ and $\sigma_i > 0$. Note, that according to Eq. (12) the centers \bar{x}_1 and \bar{x}_N at the boundaries are not changed by the learning algorithm.

3.2 The Spike Problem

The learning algorithm (11) to (13) is now applied to the following example (Fig. 3): The desired input-output characteristic of the fuzzy controller is the smooth function

³In the single layer case the gradient descent learning rule is equivalent to the back-propagation algorithm.

$y_d(x) = 1 + x - x^2 + 0.25x^3 - 0.019x^4 + 0.0003x^5$ with $0 \leq x \leq 10$. The initial rule conclusions $c_i(t=0)$ for all i are set to $c_i(0) = 5$, the initial widths to $\sigma_i(0) = 0.7$ and the initial centers \bar{x}_i are evenly spaced on $\bar{x}_i = 0, 1, 2, 3, \dots, 10$. Thus, 11 rules are stated. The learning rate is chosen to be⁴ $\gamma = 0.5$. During the learning process $0.01 \leq \sigma_i < \infty$ and $0.01 \leq d_i < \infty$ are allowed, where

$$d_i = \bar{x}_{i+1} - \bar{x}_i \quad (14)$$

is the distance between adjacent centers of Gaussians. The operating points ξ are chosen randomly (MATLAB random-seed = 0).

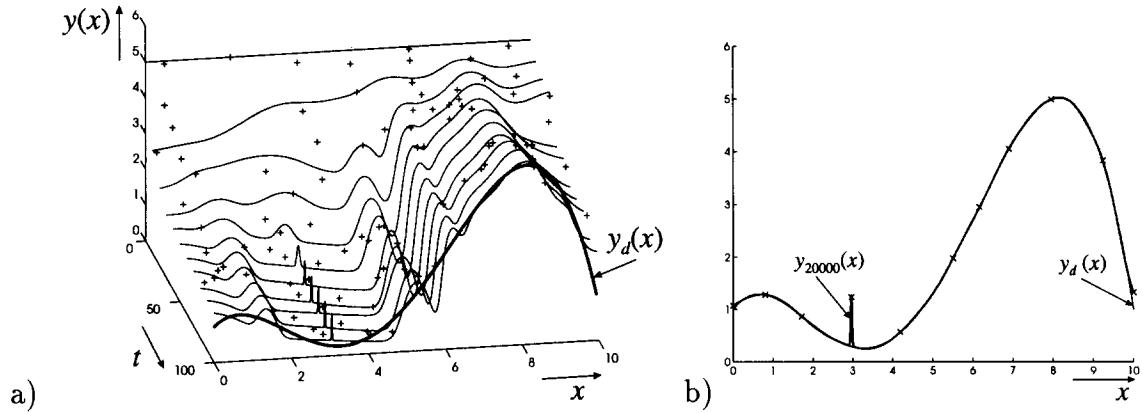


Fig. 3: (a) Convergence behavior over time $0 \leq t \leq 100$ to the desired function $y_d(x)$.

The operating points ξ are marked with '+'. A line is plotted for every 10th adaptation only. (b) Result $y_{20000}(x)$ after 20000 updates. The centers \bar{x}_i are marked with 'x'. [$\sigma_i(0) = 0.7$, $\gamma = 0.5$, $\sigma_{\min} = 0.01$, $\sigma_{\max} = \infty$, $d_{\min} = 0.01$, $d_{\max} = \infty$]

The stated example looks simple as $y_d(x)$ is smooth and all parameters look to be properly chosen. Nevertheless, the convergence result is very disappointing (Fig. 3) as two types of spikes occur. In the next section we discuss how these spikes occur and how they can be avoided.

4 Spikes

Two types of spikes can be distinguished: *Narrow Basis Function Spikes* and *Non-Monotonic Basis Function Spikes*.

4.1 Narrow Basis Function Spikes

The spike at $x \approx 3$ in Fig. 3 is a narrow basis function spike. The adaptation according to Eqs. (11), (12), (13) causes the membership function $\mu_4(x)$ and therefore the basis

⁴With $\lambda = \frac{\gamma}{2 \sum_{j=1}^N b_j^2(\xi) \left\{ 1 + 4[c_j - y(\xi)]^2 \left(\frac{\xi - \bar{x}_j}{\sigma_j^2} \right)^2 \left[1 + \left(\frac{\xi - \bar{x}_j}{\sigma_j} \right)^2 \right] \right\}}$ the reduction of the local error function $I(\xi)$ is approximately $\gamma I(\xi)$ (see also [9, 10]).

function $b_4(x)$ to become thinner and thinner (see also Fig. 6). The thinner the basis function, the smaller the probability of a correct update of this narrow basis function. In the following section we show how the “short sighted” reduction of the local error function $I(\xi)$ causes this unsatisfying effect.

4.1.1 Origin of the Spike

All three parameters c_i , \bar{x}_i , and σ_i are adapted proportional to the learning rate λ , proportional to the local error $e(\xi)$, and proportional to the contribution $b_i(\xi)$ to the output $y(\xi)$ of the corresponding basis function $b_i(x)$ (Eqs. (11)-(13)). The operating point ξ must therefore be in the neighborhood of \bar{x}_i in order to significantly change one of the parameters c_i , \bar{x}_i , or σ_i .

The parameters \bar{x}_i and σ_i are adapted proportional to $c_i - y(\xi)$ (Eqs. (12), (13)). If the operating point ξ is very different from \bar{x}_i then $b_i(\xi)$ is very small and $y(\xi)$ of Eq. (5) is not dominated by $c_i b_i(\xi)$. In this case, the term $c_i - y(\xi)$ includes no relevant information for adapting the parameters of $b_i(\xi)$ and the parameters are almost randomly changed. The small value of $b_i(\xi)$ helps to reduce the bad influence of these adaptations.

On the other hand, if the operating point ξ is close to \bar{x}_i then $b_i(\xi)$ dominates over the other basis functions and $y(\xi)$ is mainly defined by the term $c_i b_i(\xi)$.

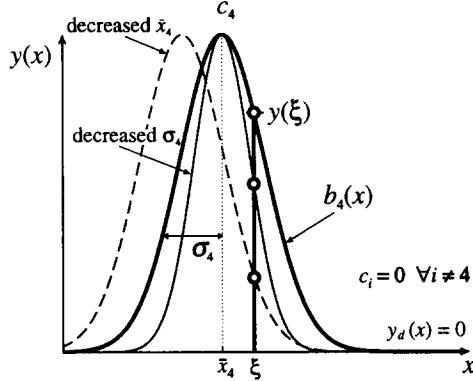


Fig. 4: The local deviation $e(\xi) = y_d(\xi) - y(\xi)$ and also the error function $I(\xi)$ is decreased if σ_4 is decreased and/or the center \bar{x}_4 of the basis function $b_4(x)$ is shifted to increase the distance between \bar{x}_4 and ξ . Both of these adaptations conserve the spikes.

If now $\text{sign}(c_i - y(\xi)) \neq \text{sign}(e(\xi))$ then σ is decreased (Eq. (13)) and the center \bar{x}_i is shifted away from the operating point ξ (Eq. (12)) (see Fig. 3a or illustration in Fig. 4). We call this the “short sightedness” of the learning algorithm. It is correct that the strategy helps to reduce the local error function $I(\xi)$ and $e(\xi)$, but from a global point of view it is by no means obvious that $y(x)$ approximates $y_d(x)$ any better after the update than before. Note also that the adaptation of \bar{x}_i is proportional to $\frac{1}{\sigma_i^2}$ and the adaptation of σ_i is even proportional to $\frac{1}{\sigma_i^3}$. Consequently, the thinner the spike already is, the faster it becomes even more thin.

Both parameters \bar{x}_i and σ_i are not changed if the operating point ξ lies exactly at the center \bar{x}_i but are adapted proportional to $\xi - \bar{x}_i$ and $(\xi - \bar{x}_i)^2$, respectively. If the distance

between ξ and \bar{x}_i is too large, the factor $b_i(\xi)$ is small, such that the maximal change is achieved at a distance of $\frac{\sigma}{\sqrt{2}}$ or σ , respectively (Fig. 5).

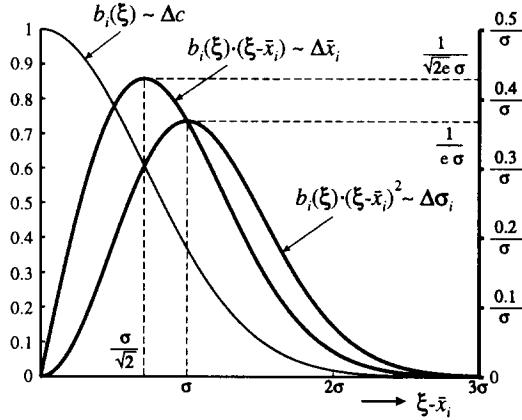


Fig. 5: Illustration of the distance dependent factors in Eqs. (11), (12), and (13).

The best way to destroy the spikes is to perform an adaptation directly on or very close to the center \bar{x}_i of the spike. If by chance no operating point ξ is near the center at the time when the spike is formed, then the spike will become thinner and thinner, and the probability of such an operating point decreases. In the example of Fig. 3 the spike does not vanish, even after 20000 updates (Fig. 3b).

4.1.2 Avoiding the Spike

One method to avoid the narrow basis function spikes is simply to limit the widths of the membership functions to $\sigma_i \geq \sigma_{\min}$ (Fig. 9). The lower limit σ_{\min} must be chosen according to the smoothness of the desired input-output characteristic. If σ_{\min} is too small there is again a risk of spikes as the probability of a proper update decreases.

As mentioned above, $\text{sign}(c_i - y(\xi)) \neq \text{sign}(e(\xi))$ is one of the main requirements for generating a spike. In the example of Fig. 3 the initialization of the rule conclusions $c_i(0)$ was very different from the desired solution. In this case the requirement is easily fulfilled in less frequently updated regions and a spike is generated. If the initial rule conclusions already represent the desired input-output characteristic and the learning algorithm has just to track a very slowly varying desired characteristic $y_d(x, t)$, then it is much harder to find situations where spikes occur.

4.2 Non-Monotonic Basis Functions

The undesired local minimum at $x \approx 5.7$ or at $x \approx 6.4$ in Fig. 3a is called a *non-monotonic basis function spike* as it is generated by a non-monotonic basis function. Note that the membership functions (6) are Gaussian and are therefore by definition monotonic⁵ but the basis functions

$$b_i(x) = \frac{\mu_i(x)}{\sum_{j=1}^N \mu_j(x)} \quad (15)$$

⁵We define a membership function $\mu_i(x)$ or a basis function $b_i(x)$ to be monotonic if the function is monotone decreasing with increasing distance from its maximum.

are non-monotonic. The denominator term of Eq. (15) can be seen as a normalization [1] of the membership function $\mu_i(x)$ to

$$\sum_{i=1}^N b_i(x) = 1 , \quad \forall x \in \text{IR} . \quad (16)$$

The normalization can cause a wide membership function (e.g. μ_5 with $\sigma_5 = 1.9451$ in Fig. 6) to dominate other membership functions in regions where the rule conclusion corresponding to this wide membership function is not appropriate ($x = 5.7$ or $x = 6.4$ in Fig. 6 are too distant from the center of $\mu_5(x)$ so that the rule conclusion c_5 of the rule R^5 is relevant in this region). This domination of $\mu_i(x)$ in more distant regions causes $b_i(x)$ to be large in these regions according to Eq. (15). In other words, the basis function of such a membership function can become non-monotonic (e.g. $b_5(x)$ in Fig. 6).

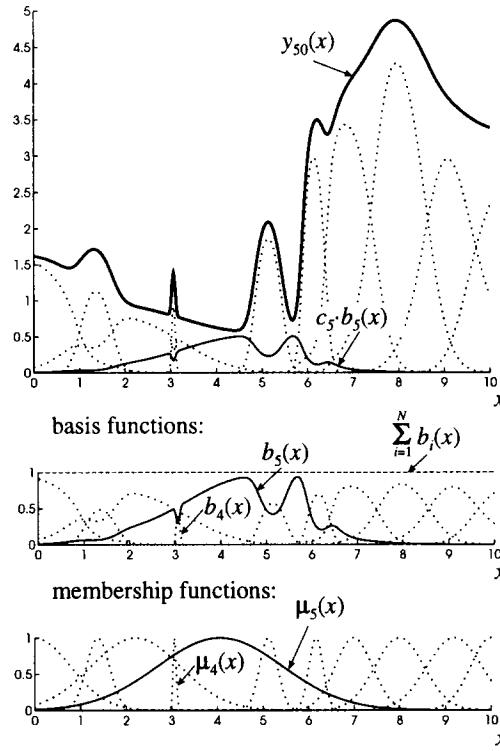


Fig. 6: Composition of $y_{50}(x)$ in Fig. 3. The domination of $\mu_5(x)$ for $x \approx 5.7$ causes the basis function $b_5(x)$ to be non-monotonic.

Note also that the adaptation of \bar{x}_i or σ_i for any single membership function changes the shapes of all basis functions according to Eq. (15). Hence, the effect of changing these nonlinear parameters on the output $y(x)$ of the fuzzy-controller is very complex.

4.2.1 Why Monotonic?

For real world applications of the learning algorithm it is important that a unique solution of the corresponding interpolation/approximation problem exists [14]. It is known

from the approximation theory that $b(\sqrt{x})$ being completely monotonic⁶ assures the existence of a unique solution [14]. The Gaussian basis function with $b(x) = \exp\left(-\frac{x^2}{\sigma^2}\right)$ and $b(\sqrt{x}) = \exp\left(-\frac{x}{\sigma^2}\right)$ being completely monotonic fulfills this requirement. It is most likely impossible to guarantee completely monotony when adapting the nonlinear parameters \bar{x}_i and σ_i .

The minimal requirement for a robustly working learning algorithm is at least the monotony of the basis functions $b_i(x)$. It would be counter-intuitive and inconsistent with our experience with learning algorithms to allow non-monotonic basis functions. Just consider the case in the example of Fig. 3 and Fig. 6 where we adapt the rule conclusion c_5 of the rule R^5 due to a deviation at the operating point $\xi = 4$. As the corresponding non-monotonic basis function $b_5(x)$ is at $x = 5.7$ even larger than for $x = 4$ the change of c_5 affects the output $y(x)$ at $x = 5.7$ more than at the location $x = 4$ where the deviation was recognized.

Therefore we show under which conditions the basis functions are monotonic.

4.2.2 Limits of \bar{x}_i and σ_i for Monotonic Basis Functions

Define the limits σ_{\min} , σ_{\max} by

$$\sigma_{\min} \leq \sigma_i \leq \sigma_{\max} \quad (17)$$

and the limits d_{\min} , d_{\max} for the distances (14) of adjacent centers by

$$d_{\min} \leq d_i \leq d_{\max} . \quad (18)$$

Further, define the typical values

$$\sigma_{\text{typ}} = \sqrt{\sigma_{\max}\sigma_{\min}} \quad (19)$$

and

$$d_{\text{typ}} = \sqrt{d_{\max}d_{\min}} . \quad (20)$$

In [7] we derived the relationship

$$\frac{d_{\min}}{d_{\max}} \geq \frac{1}{2} \left[\left(\frac{\sigma_{\max}}{\sigma_{\min}} \right)^2 - 1 \right] \quad \text{with} \quad \sigma_{\text{typ}} \leq d_{\text{typ}} \quad (21)$$

as a “sufficient” condition for monotonic membership functions $b_i(x)$. The main part of this result was analytically derived. Due to the transcendental nature of the Gaussian functions we were not able to complete the analyses without empirical argumentations. Consequently, the term *sufficient* cannot be used in the exact mathematical sense but only in an empirical sense.

The experiments (Fig. 7) confirm the result (21). For a given set of membership functions there can always be determined a σ_{\min} , σ_{\max} , d_{\min} , and d_{\max} . Thus, every set of membership functions yields a point within the diagram $\frac{d_{\min}}{d_{\max}}$ vs. $\frac{\sigma_{\min}}{\sigma_{\max}}$ of Fig. 7. None of the 50000 random⁷ distributions of membership functions violates condition (21). If σ_{typ} is not restricted to be smaller than d_{typ} according to (21), then the left equation of (21) is no longer a sufficient condition (Fig. 8).

⁶Let $b(\|\mathbf{x}\|)$ be a zero centered basis function such that the basis functions $b_i(\mathbf{x}) = b(\|\mathbf{x} - \bar{\mathbf{x}}_i\|)$ are just shifted versions of $b(\|\mathbf{x}\|)$. According to Schoenberg 1938 a function $f(r)$ is said to be completely monotonic on $(0, \infty)$ provided that it is $C^\infty(0, \infty)$ and $(-1)^j \frac{\partial^j f(r)}{\partial r^j} \geq 0$, $\forall x \in (0, \infty)$, $\forall j \in \mathbb{N}_0$.

⁷Worst case situation: d_i is solely d_{\min} or d_{\max} , σ_i is solely σ_{\min} or σ_{\max} .

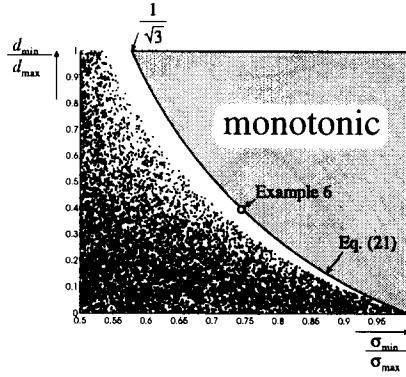


Fig. 7: Out of 50000 randomly distributed sets of membership functions with $\sigma_{\text{typ}} \leq d_{\text{typ}}$, 8813 sets (\bullet) are non-monotonic. None of them violates condition (21). 41187 are monotonic (not shown in the plot as it would yield a black rectangle).

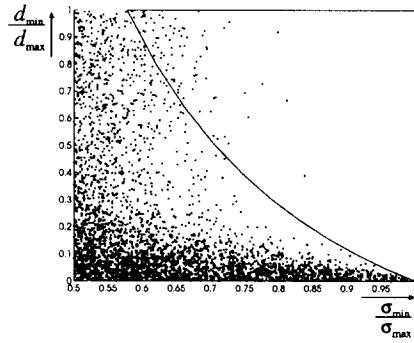


Fig. 8: Out of 50000 randomly distributed sets of membership functions with $\sigma_{\text{typ}} > d_{\text{typ}}$ (in conflict with Eq. (21)), 3598 sets (\bullet) are non-monotonic, 46402 are monotonic (not shown in the plot as it would yield a black rectangle).

5 Examples

Table 1: Overview of Examples 1 to 6

#	γ	$\sigma_i(0)$	σ_{\min}	σ_{\max}	d_{\min}	d_{\max}	Results
1	0.5	0.7	0.01	∞	0.01	∞	Fig. 3, both types of spikes, not robust
2	0.5	1	0.01	∞	0.01	∞	similar to Fig. 9, no spikes, not robust
3	0.01	0.7	0.01	∞	0.01	∞	little worse than Fig. 9 with narrow BF spike, not robust
4	0.5	0.7	0.1	1	0.33	3	similar to Fig. 3 but no narrow BF spike, not robust
5	0.5	0.7	0.33	1	0.33	3	similar to Fig. 3 but no spikes, not robust
6	0.5	0.86	0.86	1.16	0.63	1.59	Fig. 9, no spikes

Based on the unsatisfying example of Fig. 3 and 6 we apply now the results of Section 4 to improve the performance of the learning algorithm. The setup of these examples is exactly the same as in Section 3.2. Even the operating points are the same. The only changed parameters are listed in Table 1.

Note that all these examples were performed with randomly distributed operating

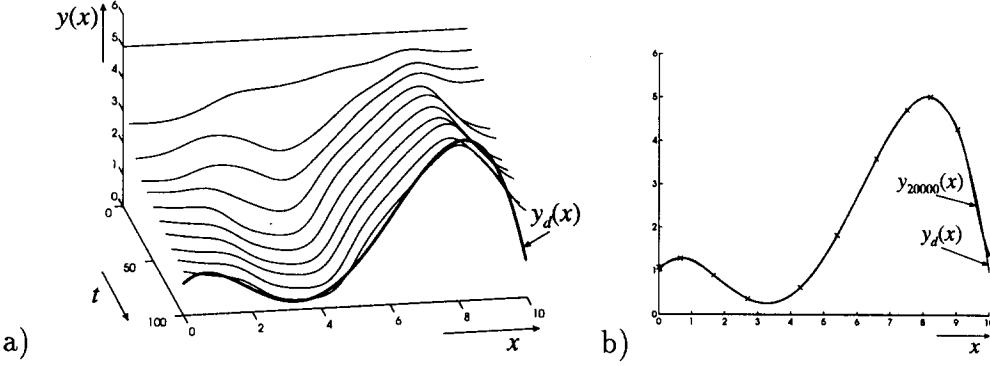


Fig. 9: Example 6: (a) Convergence behavior over time $0 \leq t \leq 100$ to the desired function $y_d(x)$. A line is plotted for every 10th adaptation only. (b) Result $y_{20000}(x)$ after 20000 updates. The centers \bar{x}_i are marked with 'x'. [$\sigma_i(0) = 0.86$, $\gamma = 0.5$, $\sigma_{\min} = 0.86$, $\sigma_{\max} = 1.16$, $d_{\min} = 0.63$, $d_{\max} = 1.59$]

points ξ . Under these circumstances it is possible to avoid the spikes by some minor modifications of the parameters (e.g. Example 3), but this would not be possible if the worst possible operating points are applied (e.g.: to decrease the learning rate like in Example 3 but to apply each operating point several times would be equivalent to Example 1). The term “not robust” in Table 1 should emphasize this sensitivity of the choice of operating points.

A deterministic improvement is achieved if the range of σ_i and d_i is limited according to (21). The particular values chosen for Example 6 are determined by maximizing the area $(1 - \frac{d_{\min}}{d_{\max}})(1 - \frac{\sigma_{\min}}{\sigma_{\max}})$ in Fig. 7 of allowed parameter variations according to (21) and setting $\sigma_{\text{typ}} = d_{\text{typ}} = 1$ (Fig. 9). The robustness of the learning algorithm with these small parameter variations is significantly more robust than for the other cases.

6 Rentability Check

When just the linear parameters (the rule conclusions c_i) are adapted, only one memory is required for each rule. Thus, N memories are required overall. If also the nonlinear parameters \bar{x}_i and σ_i are adapted, then $3N$ memories are required in the one-dimensional case ($n = 1$). At the same time the computational effort for evaluating the learning algorithm Eqs. (11), (12), (13) is approximately three times higher than evaluating (7) for the linear case.

Let us assume to approximate a function $y_d(x)$ with a given accuracy. In the best case the function has only one detail where a relatively narrow (σ_{\min}) basis function is required and the rest of the function can be approximated with relatively wide (σ_{\max}) basis functions.

In the case of the linear adaptation we need m basis functions with $\sigma = \sigma_{\min}$, provided we do not know the location of this single detail (otherwise different but constant σ_i , \bar{x}_i could be used). The convergence of the linear learning algorithm is guaranteed.

In the case of the nonlinear adaptation it is sufficient to use one basis function with σ_{\min} and the rest of the basis functions with σ_{\max} . According to (21) σ_{\max} can at most

be $\sqrt{3}$ times larger than σ_{\min} in order to achieve reasonable robustness of the learning algorithm, but still no convergence is guaranteed. Approximately $\frac{m}{\sqrt{3}}$ basis functions are utilized and $3\frac{m}{\sqrt{3}} = m\sqrt{3}$ memories are required to store the corresponding parameters. Consequently, the linear adaptation is the more efficient and more robust method.

A similar relation would hold for $n > 1$ if for each of the N rules in each of the n dimensions a separate membership function were defined. In commonly used fuzzy systems this is not the case as only one set of membership functions is used regardless of the rules. The number of additional parameters \bar{x}_i, σ_i increases linearly with the dimension n . If n is sufficiently high it is more efficient from the memory point of view to use the nonlinear configuration. On the other hand, the robustness of the online learning algorithm drastically decreases as one membership function is then responsible for many basis functions, and changing a single nonlinear parameter directly affects the corresponding set of basis functions. The locally bounded influence of an adaptation, which is required for a robust online learning algorithm, is then not provided.

7 Conclusion

The goal of the work was to analyze if the nonlinear online adaptation of membership functions has any practical relevance. We made every effort to do the analyses as objectively as possible so that the reader can draw his or her own conclusions. Nevertheless, in our opinion the risk of convergence problems (like the spikes analyzed in this paper) dominates over the questionable advantages and therefore we do not recommend the nonlinear online adaptation.

The more the parameter variation is bounded the better is the convergence behavior but the poorer is the memory advantage.

The situation is completely different if the parameter optimization is performed offline. Other learning algorithms like genetic algorithms [3] are successfully applied for offline optimization.

Acknowledgements

The authors wish to thank Dr. G. Brasseur and Prof. F. Paschke for their general support.

References

- [1] M. Brown and Ch. Harris. *Neurofuzzy Adaptive Modelling and Control*. Prentice Hall, New York, 1994.
- [2] D. Driankov, H. Hellendoorn, and M. Reinfrank. *An Introduction to Fuzzy Control*. Springer-Verlag, New York, 1993.
- [3] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, 1989.
- [4] S. Haykin. *Neural Networks*. IEEE Press and Macmillan, New York, 1994.

- [5] M. Heiss, D. Heiss, and S. Kampl. Lernen linear interpolierter Kennlinien. *Automatisierungstechnik at*, 42(11):479–506, 1994.
- [6] M. Heiss and J. Leichtfried. Selflearning fuzzy controller with smooth transfer characteristic and guaranteed convergence. In *IEEE Conference on Control Applications (CCA 94)*, pages 1251–1256, Glasgow, 1994.
- [7] A. Hofbauer. The Origin of Spikes During Online Adaptation of Membership Functions (in German). Master’s thesis, University of Technology Vienna, November 1994.
- [8] A. Hofbauer and M. Heiss. The origin of spikes during online adaptation of membership functions. In *Int. Symp. on Fuzzy Logic (ISFL’95)*, paper # 101–007, ETH Zürich, Mai 1995. Academic Press.
- [9] J.-S. R. Jang. Self-learning fuzzy controllers based on temporal back propagation. *IEEE Transactions on Neural Networks*, 3(5):714 – 723, September 1992.
- [10] J.-S. R. Jang. Anfis: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3):665 – 685, May/June 1993.
- [11] B. Kosko. *Neural Networks and Fuzzy Systems*. Prentice Hall, New Jersey, 1992.
- [12] J. Leichtfried and M. Heiss. Ein kennfeldorientiertes Konzept für Fuzzy-Regler. *Automatisierungstechnik at*, 43(1):31–40, 1995.
- [13] J. Leichtfried and M. Heiss. Fuzzy-Regler als smoother Regel-Interpolator. *Int. J. of Automation Austria*, 3(2):47–61, 1995.
- [14] T. Poggio and F. Girosi. A theory for approximation and learning. A.I. Memo 1140, M.I.T., 1989.
- [15] D. E. Rumelhart et al. *Parallel Distributed Processing*, volume 1. MIT-Press, Boston, 1986.
- [16] R. M. Sanner and J.-J. E. Slotine. Gaussian networks for direct adaptive control. *IEEE Trans. on Neural Networks*, 3(6):837–863, November 1992.
- [17] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man, and Cyberrn.*, 15(1):116–132, 1985.
- [18] L.-X. Wang. *Adaptive Fuzzy Systems and Control*. Prentice Hall, New Jersey, 1994.
- [19] L. X. Wang and Jerry M. Mendel. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE Transactions on Neural Networks*, 3(5):807–814, 1992.
- [20] B. Widrow and S. D. Stearns. *Adaptive Signal Processing*. Prentice-Hall, New Jersey, 1985.

Low Cost Bildverarbeitung und ihre industrielle Anwendung

Ch. Fessel

Institut für Handhabungsgeräte und Robotertechnik
Technische Universität Wien

Kurzfassung: Low Cost Bildverarbeitungssysteme können sich infolge ihrer vielfältigen Einsatzmöglichkeit etablieren. Dieser Bericht analysiert Grundkonzepte der digitalen Bilddatenbearbeitung- und analyse, und gibt Einblick in die gängigsten Verfahren. In weiterer Folge wird die zur Low Cost Bildverarbeitung erforderliche Hardware, sowie deren Funktion besprochen. Mit der Erläuterung der umfangreichen Bildverarbeitungssoftware, zur Realisierung der eingangs erläuterten Verfahren, schließt diese Arbeit ab.

1 Einleitung

Low Cost Bildverarbeitungssysteme (*Vision System*) gewinnen in der Robotik und der Automatisierungstechnik sehr stark an Bedeutung. So gibt es neben Sortieraufgaben an Förderbändern vor allem Fragestellungen, bei denen es gilt, Lage, Form und Identität von Teilen zu erkennen, zu lösen. Obwohl das Vision System eine komplizierte, verantwortungsvolle und rechenaufwendige Aufgabe zu erfüllen hat, stellt es nur einen kleinen Teil in einer komplexen Fertigungsstraße dar. Aus diesem Grund sind preisgünstige, überschaubare und wartbare Kompaktlösungen mit kurzen Zykluszeiten und hoher Systemgeschwindigkeit gefordert, die an die individuellen Kundenwünsche und Anforderungen angepaßt werden.

Um ein Vision System mit geringem Investitionsaufwand und hoher Leistungsfähigkeit zu realisieren, müssen Standardkomponenten anstelle von Spezialentwicklungen verwendet werden. Daher wird als Bildverarbeitungsrechner ein konventioneller PC eingesetzt. Versuche haben ergeben, daß *Central Processing Units* der „80486“ oder „Pentium“- Leistungsklasse schneller und leistungsfähiger als spezielle Bildverarbeitungsprozessoren sind. Um diesen Geschwindigkeitsvorteil auszunützen, ist ein spezielles Speicherkonzept zur Bildspeicherung und Bearbeitung erforderlich. Ein weiterer Vorteil eines PC basierten Vision Systems ist die universelle Einsetzbarkeit des PCs, und damit die Möglichkeit zur Koppelung in einem übergeordneten *Intelligent Manufacturing System (IMS)*.

Die Programmierung des PC basierten Vision Systems erfolgt durch objektorientierte Programmierung. Die objektorientierte Bildverarbeitung bringt Vorteile wie klare, deutliche und übersichtliche Programmstrukturen mit sich. Dadurch können Programme geschaffen werden, die vom Betreiber selbst gewartet und erweitert werden können. Unterschiedlichste Anwendungen erfordern unterschiedlichste Analyse- und Bearbeitungsalgorithmen. Durch die objektorientierte Bildverarbeitung ist Ihre Implementierung und Adaption sehr einfach, wodurch das System vom Benutzer selbst erweitert werden kann. Die zugängliche

Programmierung ermöglicht die Implementierung von Fuzzy- und Neuromethoden zur Bildidentifikation und Analyse, was immer mehr an Bedeutung gewinnt. Letztendlich ist es das Ziel, ein PC basiertes Vision System soweit zu entwickeln, daß unter der Verwendung von problemspezifisch optimierten Analysealgorithmen die Steuerung eines übergeordneten Prozesses mit Daten versorgt werden kann.

2 Bildbearbeitung und Bildanalyse als Teilgebiete der Digitalen Bilddatenverarbeitung

Um konkret auf die Verfahren der digitalen Bilddatenverarbeitung, kurz auch Bildverarbeitung bezeichnet, eingehen zu können, ist zunächst eine Abgrenzung dieser Wissenschaft gegenüber verwandten Disziplinen erforderlich. Der Literatur ist unter anderem die nachfolgende Definition (Klette, R., et al., 1992) zu entnehmen, die sehr klar die Abgrenzung der Bildverarbeitung zur Computergrafik hervorhebt. Dementsprechend beschäftigt sich die Computergrafik im Gegensatz zur Bildverarbeitung mit der Erstellung von künstlichen Bilddaten zur Veranschaulichung des vorliegenden Sachverhaltes.

Die digitale Bildverarbeitung ist jener Teilbereich der Informatik, der auf analytische Fragestellungen im Zusammenhang mit bildhaften Daten ausgerichtet ist. Im Unterschied hierzu ist die Computergrafik auf Fragestellungen für synthetisch erzeugte Bilddaten orientiert. Die **Bildbearbeitung** und die **Bildanalyse** sind die beiden wesentlichen Teildisziplinen der digitalen Bildverarbeitung.

Der obigen Definition sind Bildbearbeitung und Bildanalyse als Teilgebiete der Bildverarbeitung zu entnehmen. Beschäftigt sich die Bildbearbeitung mit der Berechnung von neuen Bilddaten aus bekannten Bilddaten, um zum Beispiel ein kontrastreicheres oder schärferes Bild zu erstellen, so versteht man unter der Bildanalyse die Interpretation von Bilddaten, das meint eine Informationsgewinnung, deren Gehalt in einer anderen Form als es das digitale Bild ist, zur Verfügung gestellt wird. Der Zusammenhang der beiden Verfahren liegt darin, daß die Bildbearbeitung dazu verwendet wird, um ein, für die anschließend stattfindende Analyse, aussagekräftiges Bild zu erzeugen. Die Analyse des Bildes kann auch durch einen menschlichen Auswerter erfolgen, geschieht sie jedoch durch Rechenalgorithmen, so ist dies jene Klasse von Verfahren, die oben als Bildanalyse deklariert wurden.

2.1 Bildbearbeitung

Werden bei den Verfahren der Bildbearbeitung neue Bilddaten, als Ergebnis eines implementierten Algorithmus aus alten Bilddaten berechnet, so ist darauf zu achten, daß unter Befolgung einer Grundregel der Bildbearbeitung die Ergebnisse immer in ein neues Bild, und nie in das Ursprungsbild geschrieben werden. Dieser Vorgang erfordert erhöhten Speicherbedarf. Dies ist aber bei Operatoren, die zur Berechnung eines neuen Pixelwertes andere, eventuell in einer gewählten Abtastreihenfolge davor liegende, Pixel betrachten, unumgänglich.

Ziele der Bildbearbeitung sind im allgemeinen die Bildverbesserung, Bildrestauration und zum Teil eine Bildsegmentation. Theoretisch kann diese Bildbearbeitung in Form eines parallelen oder seriellen Prozesses erfolgen. Unter Nutzung von Parallelprozessoren, Parallelspeichern

oder Transputern können auf Bilddaten mehrere Operationen gleichzeitig angewendet werden, das führt also zu einer parallelen Bearbeitung des Bildes. Dabei ist aber zu berücksichtigen, daß dies nur für Operatoren verwendet werden kann, die sich auf eine abgeschlossene Einheit, zum Beispiel ein Pixel, beziehen. Jede andere Anwendung würde eine Vermischung zwischen ursprünglichen Bilddaten und bereits geänderten Bilddaten innerhalb einer lokalen Operatoranwendung durch einen parallel laufenden Operator zur Folge haben.

Neben der großen Gruppe der **Punktoperationen**, deren signifikante Eigenschaft der Bezug auf einen einzigen Pixelwert, unabhängig von dessen Position im Bild, ist, stellen die **lokalen und globalen Operatoren** eine weitere wichtige Gruppe dar. Ein Überblick über typische Verfahren, welche den einzelnen Gruppen zugeordnet werden können, gibt Tabelle 1.

Punktoperationen:	lokale und globale Operatoren:
Grauwertkorrektur Schwellwertoperation Look Up Table Kontrastverbesserungen Falschfarbendarstellung Grauerthistogramm erstellen	Mittelwertoperator Gradientenfilter Medianfilter Fouriertransformation

Tab. 1 Beispiele zur Zuordnung typischer Bildbearbeitungsprozeduren

Als Spezialfall der Verfahren zur Grauwertkorrektur sollte die **Kontrastverbesserung** erwähnt werden. Es handelt sich dabei um eine Streichung jener Grauwerte, die mit definitionsgemäß geringster Häufigkeit im Bild vorhanden sind. Dies wird mittels Grauerthistogramm erfaßt, und durch eine anschließende, im zur Verfügung stehenden Grauwertbereich durchgeführte, ergebnisorientierten Streckung der Grauwertskala, ausgeglichen. Zur Kontrastbeurteilung kann der mittlere Grauwert eines Bildes herangezogen werden. Allerdings läßt der mittlere Grauwert, gebildet als das arithmetische Mittel der Grauwerte, nur eine subjektive Beurteilung der Bildhelligkeit zu. Es kann daher nur ausgesagt werden, ob es sich um ein helles oder dunkles Bild handelt. Die mittlere quadratische Abweichung hingegen, die als Differenz der Grauwerte gegen den mittleren Grauwert berechnet wird, ermöglicht eine eindeutigere Aussage über den Kontrast im Bild.

Bei lokalen und globalen Operatoren werden sowohl der Einfluß der Nachbarschaftspixel, sowie die Position des Pixels im Bild berücksichtigt, um entsprechend der Operatorgleichnungen den neuen Grauwert zu berechnen. Dabei ist der lokale Charakter einer Operation eine wesentliche Eigenschaft. Diese Eigenschaft wird dann erreicht, wenn das Operatorfenster, vorstellbar durch ein Loch in einem Blatt Papier, mittels welchem das zu betrachtende Bild abgetastet wird, gegenüber den Bildabmessungen klein ist. Mathematisch wird dieses Operatorfenster durch eine Matrix, welche das Bild zeilenförmig abtastet, und in jedem Abtastschritt den neuen Pixelwert berechnet, realisiert. Als Beispiele für derartige Operatoren können der **Mittelwertoperator**, der zum Glätten der Grauwertspitzen eines Bildes verwendet wird, sowie der **Gradientenfilter** angeführt werden. Der Mittelwertoperator besitzt Tiefpaßeigenschaften, was bedeutet, daß zwar Grauwertspitzen geglättet werden, allerdings auch bedeutungsvolle Grauwertdifferenzen, die Objektkanten repräsentieren, verwischt werden. Das hat zur Folge, daß, abhängig von der Größe des gewählten Operatorfensters, nach dem Glätten das gesamte Bild störungsfrei, aber unscharf erscheint. Der

Gradientenfilter hingegen, der Hochpaßeigenschaften besitzt, verstärkt Grauwertdifferenzen und wirkt daher als so bezeichneter „Kantenschärfer“.

Globale Operationen benötigen zur Durchführung die Grauwerte sämtlicher Pixel eines Bildes, es werden also die Werte aller vorhandenen Stützstellen verwendet. Ein Beispiel für eine derartige globale Operation ist die **Fouriertransformation**. Die Fouriertransformation wird in der Bildbearbeitung vorrangig zur Bildaufbesserung, wie zum Beispiel zur Rauschunterdrückung und zur Verminderung von Unschärfen durch Defokusierung oder zu rascher Bewegung verwendet.

2.2 Bildanalyse

Bei der Bildanalyse, deren Ergebnisse im allgemeinen nicht in Form eines digitalen Bildes, sondern als Steuerungsinformationen vorliegen, ist darauf zu achten, daß zunächst Bildmerkmale erkannt werden müssen, um sie im Anschluß interpretieren zu können. So ist es zum Beispiel beim Detektieren einer Flächenkante in einem Bild zunächst erforderlich, die Grauwertdifferenzen der unterschiedlich hellen Flächen zu erkennen. Erst dann können Bereiche gleicher Eigenschaften als zusammenhängende Bereiche erkannt werden. Nachfolgend befindet sich eine Aufstellung der unterschiedlichen Verfahren der Bildanalyse:

- ☒ Bereichssegmentierung
- ☒ Kontursegmentierung
- ☒ Hough-Transformation
- ☒ Morphologische Bildverarbeitung
- ☒ Texturanalyse
- ☒ Mustererkennung
- ☒ Bildfolgenverarbeitung

Die Verfahren der **Bereichssegmentierung** gliedern sich in die Schritte Schwellwertoperation, Zusammenhangsanalyse und Merkmalsextraktion. Mittels Schwellwertoperation werden jenen Bereichen eines Bildes, die gleiche Eigenschaften aufweisen, gleiche Klassen bzw. Identifikationen zugeordnet. Zum Beispiel kann in der Schwellwertoperation hinterlegt werden, daß den Pixeln des Bildes, abhängig davon, ob sie einen größeren oder kleineren Grauwert als der Schwellwert aufweisen, die Klasse 0 oder 1 bzw. hell oder dunkel zugewiesen werden. In der anschließenden Zusammenhangsanalyse ist hinterlegt, daß alle Pixel gleicher Klasse zu einem Bereich gehören. Ebenso muß es dem Algorithmus möglich sein, alle Elemente eines Bereiches anzugeben. Somit kann das Programm eine Merkmalsextraktion durchführen, bei welcher alle gefundenen Bereiche betrachtet werden, und aufgrund typischer Bereichsmerkmale, das kann zum Beispiel die Bereichsgröße, die Bereichsform u. v. m. sein, miteinander verglichen und beurteilt werden.

Die **Kontursegmentierung** beschäftigt sich mit der Auffindung von in der Bildvorlage enthaltenen Kanten. Wobei die Ausgabe nicht in Form eines Bildes, sondern explizit durch die Angabe der Koordinaten einer Geraden, die die gefundene Kante approximiert, erfolgen soll. Diese Bedingung ist sehr wesentlich, da sie die Anwendung der Verfahren der Bildanalyse gegenüber den Verfahren der Bildbearbeitung, wie es ein einfacher Gradientenfilter, der als Kantenschärfer wirkt, bewerkstelligen würde, erfordert. Die Aufgabe, Kanten aus einem Vorlagebild zu detektieren ist eine, die in der digitalen Bildverarbeitung sehr häufig gestellt wird. Da sich Eingabebilder für derartige Probleme aber durch unterschiedliche

Rauschcharakteristik auszeichnen, muß der für den jeweiligen Anwendungsfall optimale Operator durch Tests ermittelt werden. Ein Grundkonzept, welches den meisten Kantenoperatoren innenwohnt, ist die Kombination der Verfahren der Detektion der Konturpunkte, einer anschließenden Verdünnung der Kontur sowie der Verkettung und Konturapproximation.

3 Komponenten und Architektur eines Low Cost Bildverarbeitungssystems

Bildaufnahmegerät und Bildverarbeitungscomputer stellen die zentralen Komponenten eines Bildverarbeitungssystems, welches durch den Anschluß von Aktoren, sowie anderen Eingabe- und Ausgabegeräten erweitert werden kann, dar. Allgemein besteht die Tendenz, das Bildverarbeitungssystem konkret an das Problem angepaßt zu entwickeln. Dabei entstehen Minimallösungen, deren Vorteil hohe Bearbeitungsgeschwindigkeit und gute Zugänglichkeit sind. Folgende Komponenten sind im allgemeinen Bestandteile eines Low Cost Bildverarbeitungssystems:

- ☒ CCD Flächen- oder Zeilenkamera (SW- oder Farbkamera)
- ☒ Objektiv (gegebenenfalls Autofocus- oder Zoomobjektiv)
- ☒ Bildverarbeitungscomputer mit Bilderfassungskarte und Bildspeicher
- ☒ Software (Bildbearbeitungs- und Analysesoftware)

Gemäß Abbildung 1 wird durch das Bildaufnahmegerät (i.a. Kamera mit Objektiv) ein digitales Bild der zu betrachtenden Szenerie aufgenommen und an die Bilderfassungskarte (*Frame Grabber*) im Computer gesendet. Die Bilderfassungskarte digitalisiert die analog transportierte Bildinformation und schreibt das Bild in den digitalen Bildspeicher. Somit steht das Bild der Bearbeitung und Analyse durch den Bildverarbeitungscomputer zur Verfügung.

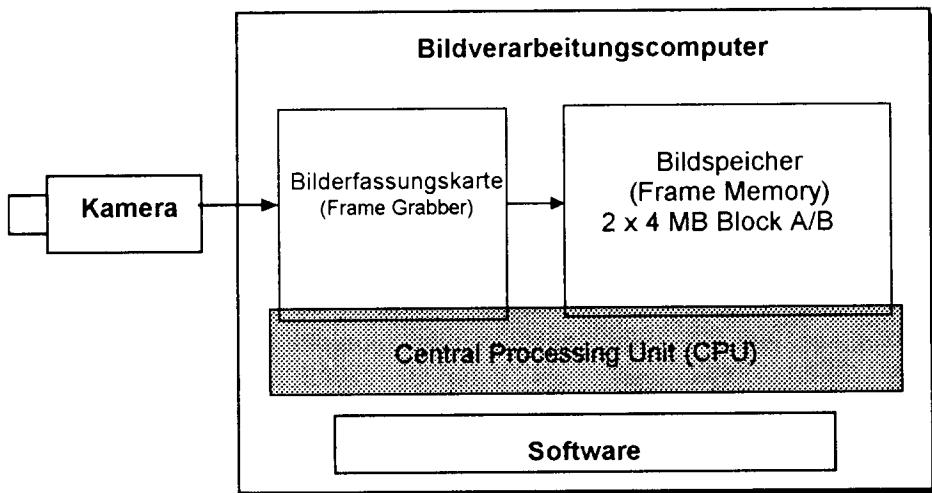


Abb. 1 Komponenten und Funktion eines Low Cost Bildverarbeitungssystems

Je nach Verwendungszweck kann das Bild nach eventuellen Filterungen, Einfärbungen und dgl. an einem Bildschirm angezeigt, oder ohne Anzeige weiterverarbeitet werden. Die aus der Verarbeitung resultierenden Ergebnisse werden an den übergeordneten Steuerrechner der Fertigungszelle gesendet. Der Steuerrechner interpretiert die erhaltenen Ergebnisse, und steuert weitere Aktoren und Sensoren an.

3.1 CCD Flächen- und Zeilenkameras

Die prinzipielle Aufgabe des Bildaufnahmegerätes besteht in der Bereitstellung eines Bildes zur Digitalisierung der Bildvorlage. Das kann über eine Analogkamera mit nachgeschaltetem Frame Grabber, eine Digitalkamera oder einen Scanner erfolgen. Dabei werden die Grauwerte bzw. Farbwerte des Bildes, also eine optische Information, in ein elektronisches Signal umgewandelt.

Wurden zu diesem Zweck früher Röhrenkameras verwendet, so tritt an ihre Stelle heute die *CCD-Kamera*, deren Funktion auf dem CCD Prinzip (*charge coupled device*) basiert. Zur Durchführung der oben beschriebenen Wandlung wird eine Diodenmatrix (MOS Transistoren) als lichtempfindliches Element verwendet. Durch diese Matrix erreicht man eine Rasterung der lichtempfindlichen Fläche, die jener des digitalen Bildes entspricht.

Die verwendeten Halbleitersensoren zeichnen sich durch ihre geringe Größe aus, was dazu führt, daß CCD-Bildaufnahmegeräte sehr klein bauen, und ein geringes Gewicht aufweisen. Vor allem die kleine Baugröße und das geringe Gewicht derartiger Kameras macht es möglich, daß Bildverarbeitung auch zur Steuerung bewegter Systeme, auf denen die Kamera montiert werden muß, eingesetzt wird. Verwendet wird dieses CCD-Prinzip sowohl in Flächenkameras, als auch in Zeilenkameras.

Die Größe der lichtempfindlichen Fläche eines Bildaufnahmegerätes wirkt sich auf die Helligkeit des aufgenommenen Bildes aus. Es wird zwar die Helligkeit maßgeblich durch die Wahl des Objektives, der entsprechenden Blendeneinstellung sowie durch die externe Beleuchtung bestimmt, jedoch kann bei extrem schlechtem Lichteinfall, durch eine größere Aufnahmefläche eine gleichmäßige Ausleuchtung erzielt werden.

Wird das Bildverarbeitungssystem in Kombination mit einem mobilen Roboter eingesetzt, so ist die Implementierung einer Autofocuskamera mit Zoomobjektiv denkbar. Konventionelle Anwendungen erfordern dies nicht. Hier wird von einem definierten Bildaufnahmegerät ausgegangen.

3.2 Bilderfassungskarte (Frame Grabber)

Wie bereits im Zuge der Abbildung 1 erläutert, schreibt der Frame Grabber das von der Kamera aufgenommene Bild in den Bildspeicher. Das Bild wird im allgemeinen analog in Form eines BAS Signals (Bild-Austastlücke-Synchronisation) transportiert, und muß daher vom Frame Grabber analog / digital (A/D) gewandelt werden. Das BAS Signal, auch als *Composite Video* bezeichnet, ist ein SW Summensignal, das sowohl die Bildinformation (also den Spannungswert für den Grauwert), als auch die Information über Austastlücke und Synchronisation beinhaltet. Dabei werden die negativen Synchronisationssignale zu den positiven Bilddaten addiert. Einerseits kann dieses BAS-Signal, wenn es gemeinsam mit der Masse (*GND*) über eine Cinch-Buchse (=Video Out) geführt wird, auf Monitoren bzw. Schirmen mit einem Video-In Eingang direkt angezeigt werden. Andererseits kann dieses Signal A/D gewandelt, und anschließend in einen Bildspeicher geschrieben werden.

Der Frame Grabber (FG) ist das Herzstück des Bildverarbeitungscomputers und ist als PC-Steckkarte ausgeführt. Durch die Adaption der Basissoftware wird er vom Benutzer auf seine Umgebung (Kamera und Computer) eingestellt. Nachdem dem Frame Grabber durch diese Adaption die Standardeinstellung der Kamera mitgeteilt wurde, schreibt er selbsttätig, auf-

grund des definierten Bildformates, die erhaltenen Bilddaten über ein 64-poliges Flachkabel in den Speicher des Bildverarbeitungscomputers. Der Frame Grabber stellt aber auch die Schnittstelle zwischen Computer bzw. Anwender und der Kamera her. Soll die Kamera angesteuert werden, so erfolgt dies üblicherweise über den Frame Grabber. Mögliche Einsatzbeispiele sind die Integration, die Kurzzeitaufnahme (*Shutter*) oder die Ringschaltung mehrerer Kameras.

Der Shutter ist eine elektronisch geregelte, zeitliche Begrenzung der Akkumulierung der Ladungen am CCD-Chip, und wird daher verwendet, um beeinflußbare Belichtungszeiten zu realisieren. Entgegen der nachfolgend angeführten Integration wird beim Shutterbetrieb (Shutterzeiten liegen in der Größenordnung von 1/60 bis 1/31000 Sekunden) die Belichtungszeit durch eine Reduktion der Integrationszeit verkürzt.

Bei der Aufnahme von sehr düsteren Szenerien kann aber durchaus eine überdurchschnittliche Belichtungszeit erforderlich sein. Dann muß das Verfahren der Integration angewendet werden. Dabei wird ein sich nicht änderndes Bild mehrmals hintereinander mit der selben Shutterzeit (circa 1/60 Sek.) belichtet, wodurch sich eine Aufhellung des Bildes ergibt. Das Objekt darf sich dabei nicht bewegen, da sonst ein „verschmiertes“ Bild produziert wird. Zu starkes Aufintegrieren bringt infolge des Dunkelstroms und des Rauschens Probleme mit sich. Daher soll nicht öfter als 10 mal belichtet werden. Mit Spezialkameras können allerdings Integrationszeiten bis zu 30 Minuten realisiert werden.

Neben einer eventuellen Bildanzeige oder Bildbearbeitung beeinflußt der Vorgang der Bildspeicherung wesentlich die Geschwindigkeit des Visionsystems. Dabei ist die Schreibgeschwindigkeit vom Speicherbedarf des Bildes, und daher von der Bildgröße abhängig. So erfordert ein Bild mit dem Format von z.B. 765 x 581 Pixel bei einem SW-Bild, bei welchem pro Pixel ein Byte gespeichert wird, ungefähr 445 kByte. Die Speicherung eines gleich großen Farbbildes, hier müssen 3 Byte pro Pixel gespeichert werden, hingegen bedarf ca. 1.3 MByte. Da zwischen Frame Grabber und Kamera kein Puffer besteht, hat der FG die zentrale Aufgabe, die Bilddaten möglichst in Videoechtzeit in den Bildspeicher zu schreiben. Geschieht dies nicht, so gehen Bilddaten zwischen Kamera und Bildspeicher verloren.

3.3 Bildverarbeitungscomputer

Der installierte Bildverarbeitungscomputer weist die nachfolgend angeführten Komponenten auf:

- ☒ Mylex MVE 486-DX2 / 66 Motherboard
EISA System Board with VESA Local Bus,
- ☒ 16 MB RAM,
- ☒ 2 x 4 MB Frame Memory (Block A and Block B),
- ☒ Frame Grabber,
- ☒ VGA Grafikkarte V7 Mirage,
- ☒ Harddisk.

So wie der Frame Grabber ist auch der Bildspeicher (vgl. Abbildung 1) als Steckkarte ausgeführt, und steht daher direkt mit der *Central Processing Unit* (CPU) in Kontakt. Dies ist einerseits zum Ansprechen des Frame Grabbers, um Kameranummer sowie Kameramode

u.v.m. einzustellen, erforderlich, andererseits muß die CPU die Bilddaten vom Bildspeicher lesen können. Erst durch den Zugang der CPU zum Bildspeicher ist das Anzeigen oder Bearbeiten der Bilddaten möglich. Die Darstellung gemäß Abbildung 1 erweckt den Eindruck, daß der Bildspeicher zur gleichen Zeit dem Frame Grabber und der CPU zugänglich ist. Dies ist jedoch nicht der Fall, da der Bildspeicher, in Block A und B getrennt, umgeschaltet wird. Damit kann der, das Speicherhandling betreffenden Forderung, laufend Bilder einzuschreiben und gleichzeitig diese lesen zu können, gerecht werden.

Der Bildspeicher ist dabei Teil des PC-Hauptspeichers, und sitzt daher an den *SIMM Modulen*, die für eine konventionelle Speichererweiterung vorgesehen sind. Die Bilddaten werden via internen Bus direkt ins RAM (*Random Access Memory*) geschrieben, wodurch nicht der langsamere *PCI Bus* zum Transfer verwendet werden muß.

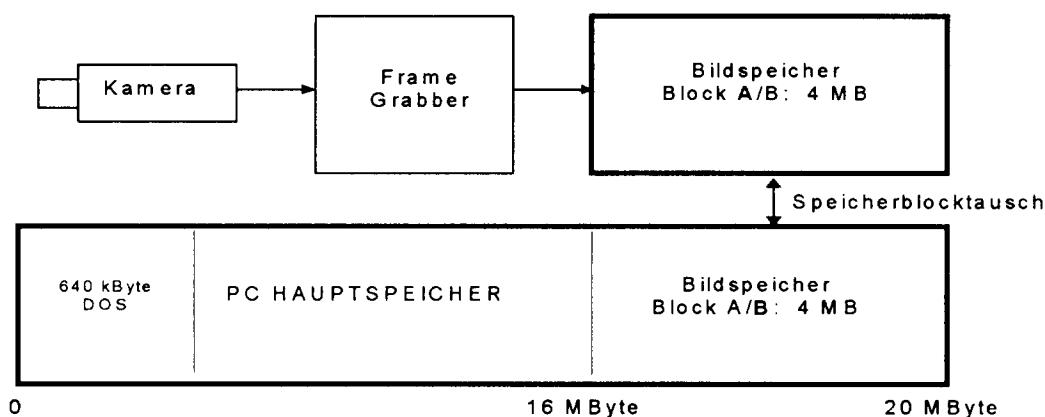


Abb. 2 Konzept für den Speicherblocktausch

Wie bereits oben erläutert, und in Abbildung 2 dargestellt, wird der Bildspeicher geteilt und besteht aus je 4 MB. Während nun eine Bank von der Kamera mit Bilddaten beschrieben wird, ist die andere Bank Bestandteil des CPU Hauptspeichers. Durch einen „out-Befehl“ erfolgt der Austausch der Speicherblöcke. Lokal betrachtet, ist immer ein Speicherblock mit der CPU und der andere mit dem Frame Grabber verbunden. Wird der Bildspeicher global betrachtet, erscheint er von zwei Seiten gleichzeitig zugänglich.

Bei der Bildverarbeitung werden die 4 MB Bildspeicher für das jeweilige Betriebssystem ausgespart, und nur der Anwendung selbst zur Verfügung gestellt. Der Bildspeicher kann je nach Speicherkonfiguration des Motherboards am Ende oder auch in der Mitte des physikalisch vorhandenen Hauptspeichers liegen. Die ersten 4 MByte bleiben jedoch immer für das Betriebssystem reserviert.

4 Bildverarbeitungssoftware und ihr industrieller Einsatz

Eine der Komponenten eines Bildverarbeitungssystems, die sich maßgeblich auf die Verarbeitungsgeschwindigkeit auswirkt, ist die implementierte Software. Die Bildverarbeitung bringt es mit sich, daß sehr große Datenmengen in möglichst kurzer Zeit bearbeitet werden müssen. Aus diesem Grund werden zeitaufwendige Arbeiten sowie der Datentransfer zwischen

Bildspeicher und Rechner in Assembler programmiert. Routinen zur Lösung anwenderspezifischer Probleme sollen aber, um eine gute Zugänglichkeit zu gewährleisten, in einer Hochsprache, wie z.B.: C, Fortran, Pascal und dgl. realisiert werden.

Damit läßt sich die Bildverarbeitungssoftware in die in Assembler programmierte, so bezeichnete

- **Basissoftware**, und in die
- **Applikationssoftware**

unterteilen. Schnittstellen zwischen den beiden Softwarepaketen sind in Assembler programmierte, und vom Anwender einsetzbare Funktionen. Derartige Funktionsaufrufe werden getätigt, um zum Beispiel die Initialisierung des Frame Grabbers sowie einen Speicherblocktausch durchzuführen. Die dabei wesentlichste Funktion ist jene, die den Startpunkt der linear im Bildspeicher liegenden Bilddaten zurückgibt. Weiters stellt die Basissoftware Variablen zur Ablage der kameraspezifischen Parameter zur Verfügung. Diese Kommunikation zwischen Basissoftware und Applikationssoftware ist der Abbildung 3 zu entnehmen.

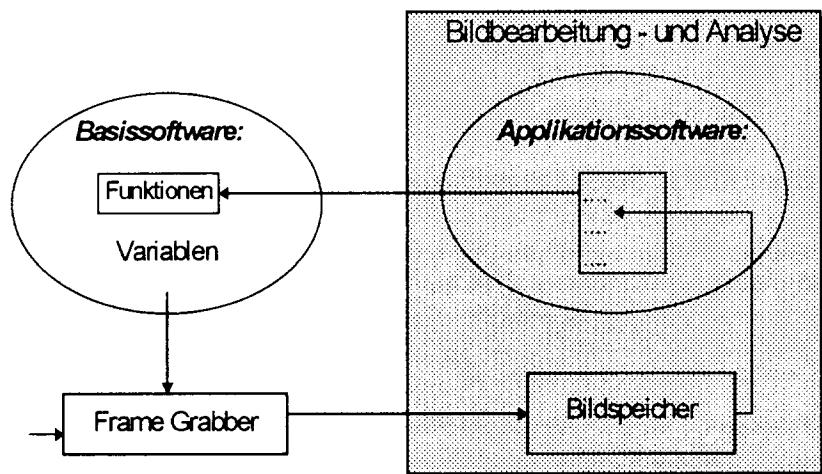


Abb. 3 Bildbearbeitungs- und Analysesoftware

An die Applikationssoftware stellt sich die Aufgabe, in möglichst kurzer Zeit brauchbare Ergebnisse zu liefern. Je zielsicherer und genauer ein Algorithmus arbeitet, um so höher ist der mathematische Aufwand und um so länger die Rechenzeit. Durch Austesten muß der für den konkreten Anwendungsfall günstigste Algorithmus gefunden werden.

Die Applikationssoftware wurde objektorientiert unter Verwendung der Hochsprache C⁺⁺ als Windowsanwendung programmiert. Objektorientierte Programmierung wird oft als ein Prozeß beschrieben, bei welchem ein Rechenmodell eines realen Prozesses erstellt wird. Das Rechenmodell besteht dabei aus einzelnen Objekten, welche durch Versendung von Nachrichten untereinander kommunizieren. Somit kann das Zusammenfassen von Daten und Funktionen, die zur Bearbeitung dieser Daten benötigt werden, als das wesentliche Prinzip der Objektorientierten Programmierung (OOP) bezeichnet werden. Die dadurch gebildete Einheit von Elementen nennt man Klasse. Ein Objekt ist nun ein Konstrukt, welches die Eigenschaften und Fähigkeiten einer Klasse zugewiesen bekommt. Der Vorteil darin liegt, daß ein Objekt nicht alle Eigenschaften einer Klasse übernehmen muß, sondern einzelne Eigenschaften für

dieses Objekt gänzlich oder nur teilweise neu definiert werden können. Weiters besteht die Möglichkeit, mehrere Objekte, abgeleitet von einer Klasse zu definieren. Es kann auch eine Klasse von einer anderen abgeleitet werden, um zum Beispiel eine eigene Klasse zu definieren.

Den grundsätzlichen Ablauf einer Applikation, beginnend mit der Initialisierung und der Deklaration der kameraspezifischen Parameter bis hin zum Rücksetzen der Interrupt Serviceroutine, enthält die nachfolgende Abbildung 4.

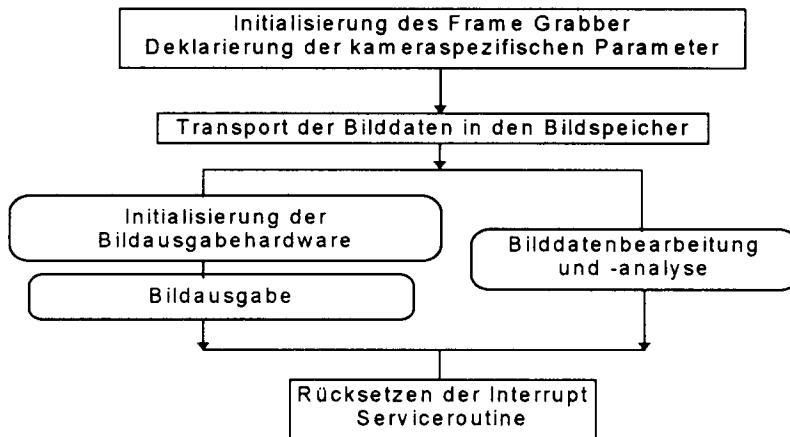


Abb. 4 Ablauf einer Bildverarbeitungsapplikation

In der Bildverarbeitung ist es im allgemeinen nicht üblich, parallel zur Bildbearbeitung und der laufenden Analyse, das aufgenommene Bild am Monitor anzuzeigen. Allerdings kann dies zur Realisierung einer übergeordneten Kontrolle einer Montage- oder Demontagezelle verwendet werden.

Die **Bildbearbeitungs- und Analysesoftware** muß entsprechend der gestellten Anforderungen entwickelt und gegebenenfalls modifiziert werden. Gängige Fragestellungen sind die Konturfindung entlang einer definierten Geraden zufolge der Detektion eines Grauwertsprunges, sowie etwaige Kantenapproximationen und Bereichsdetektionen.

Ein **Kantenextraktor** wurde basierend auf einer Gradientberechnung realisiert, und umfaßt die Schritte der Kontorpunktdetektion, Konturverdünnung, Verbindung und der abschließenden Konturapproximation. Die angesprochene **Bereichssegmentierung** schließt an die häufig erforderliche Bildbearbeitung und Bildverkleinerung an. Das anschließend erstellte Grauerthistogramm dient als Basis zur Schwellwertfindung und der Erstellung eines quasi Binärbildes. Das quasi Binärbild ist Grundlage der Bereichssegmentierung, deren Ergebnisse in weiterer Folge zur Extrahierung von Charakteristiken verwendet werden.

5 Zusammenfassung und Ausblick

Die Bildverarbeitung ist als Werkzeug in der intelligenten Fertigungsautomation nicht mehr wegzudenken. Gehört heute die klassische Sortieraufgabe, als Pionierarbeit in der Bildverarbeitung, der Vergangenheit an, so gilt es nun, Probleme der Objektidentifikation und

Umgebungsidentifikation unter Verwendung der Bildverarbeitung zu lösen. Zahlreiche Entwicklungen der Vergangenheit tragen ihre Früchte, wodurch es möglich ist, Bildverarbeitung auch in Form einer Low Cost Lösung einzusetzen. Zentrale Stellung genießt dabei der Bildverarbeitungscomputer, der einen im Hauptspeicher plazierten Bildspeicher verwaltet, und die Berechnung der erforderlichen Bildverarbeitungsroutinen tätigt. Die hierzu erforderlichen Hauptkomponenten sind der Frame Grabber und der Bildspeicher. Weiters zum Einsatz kommen heute gängige Industrie-(CCD) Kameras. Das weitreichende Anwendungsfeld der Low Cost Bildverarbeitung bringt die Forderung nach unterschiedlichsten Analysealgorithmen, die für die entsprechende Anwendung getestet und optimiert werden müssen, mit sich. Bei der Implementierung eines Bildverarbeitungssystems muß weiters auf die Realisierung von definierten Beleuchtungszuständen und die mögliche Ausschaltung ungünstiger Umgebungseinflüsse geachtet werden.

Literatur

- Bässmann, H., Besslich, Ph.W. (1993). Bildverarbeitung - Ad Oculos. Springer Verlag, Berlin.
- Breuckmann, B., 1993, Bildverarbeitung und optische Meßtechnik in der industriellen Praxis, Franzis' Verlag, München.
- Fessel, Ch. (1995). Test und Inbetriebnahme eines Low Cost Bildverarbeitungssystems, Diplomarbeit, Institut für Handhabungsgeräte und Robotertechnik TU Wien, Wien.
- Fessel, Ch. and R. Probst (1995). Low Cost Vision System in Robotics: A New Approach. In: Preprints of the 4th International Workshop Robotics in the Alpe-Adria-Region RAAD'95, Vol.2, p.275-278. Pörtschach, Austria.
- Fisher, R. (1989). From surfaces to objects: computer vision and the three dimensional scene analysis, Wiley.
- Klette, R., Zamperoni, P., 1992, Handbuch der Operatoren für die Bildbearbeitung. Vieweg Verlag. Braunschweig, Wiesbaden.
- Steinbrecher R., 1993, Bildverarbeitung in der Praxis. Oldenbourg Verlag, München.

Fully Digital Induction Motor Drive for Positioning Applications Without Measuring Electrical Quantities

Manfred Schrödl, Markus Schermann
Technische Universität Wien

Manuscript received January 17, 1996

Abstract

The paper deals with a new method for highly-dynamic control of an induction motor (IM) without measuring electrical quantities. The drive system can be realized in fully digital technology because neither phase currents nor voltages are used for control. The one and only feedback signal is the angular position of the rotor which is used to calculate proper reference values of the motor voltages from given torque-, velocity- or position commands. The model of the induction motor is based on indirect field-orientation. The voltages are generated by a voltage source inverter (VSI) using pulselwidth modulation (PWM). Measurements on a 1.5 kW drive verify the excellent behaviour of the drive in position-controlled operation.

1 Introduction

Usually, drives without measuring electrical quantities are realized in open loop. They are widely used in industrial applications, if there are no special demands concerning transient and low-speed behaviour (e.g. pumps, fans). In such systems, the IM is fed by a VSI varying the voltage as a function of synchronous frequency (V/f-control). Since there is no feedback, drives with V/f-characteristics are characterized by bad dynamic properties, unsatisfactory torque production at low synchronous speed and weak damping in a certain operating range.

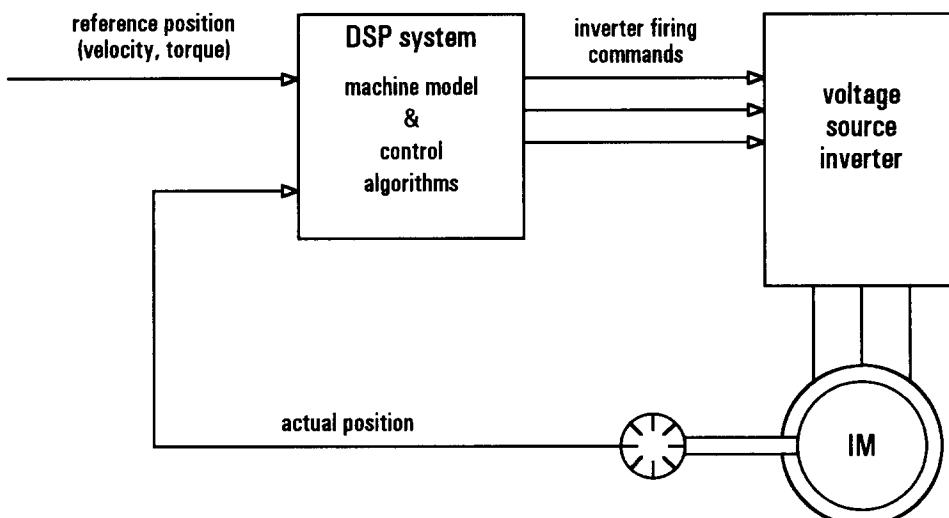


Fig. 1: Induction motor drive with fully digital control

The properties of the drive presented in this paper depend strongly on the quality of the motor current waveforms. The switching lag-times of the voltage source inverter cause serious voltage deformations, particularly at low frequencies. Hence, a dead-time compensating method is introduced to guarantee proper current waveforms. A special calculation of the pulselwidths ensures a good utilization of the DC link voltage with only small computational expense.

The proposed drive is well suited to high performance servo applications because only a position sensor is used for control (Fig. 1). The control system consists of few components resulting in minimal hardware expense. Using a standard induction motor, the presented system guarantees high robustness and low costs.

2 Model of the Induction Motor

The dynamical behaviour of the IM is described by space-phasor equations according to (Kovacs, P.K., 1984). The model introduced in this chapter enables the realization of a fully digital and highly dynamic drive. The following basic equations refer to a general reference frame rotating at angular velocity ω_k . All variables are used in per unit (p.u.) form.

The space phasor $\underline{\xi}$ built from phase quantities ξ_i , $i=1,2,3$ (ξ represents current i , voltage v or flux linkage ψ) is defined in the following way:

$$\underline{\xi} = \frac{2}{3} \cdot (\xi_1 + \xi_2 \cdot e^{j\frac{2\pi}{3}} + \xi_3 \cdot e^{j\frac{4\pi}{3}}). \quad (1)$$

The stator voltage equation describes the relation between stator voltage, stator current and stator flux linkage:

$$\underline{v}_S = r_S \underline{i}_S + \frac{d\underline{\psi}_S}{d\tau} + j\omega_k \underline{\psi}_S. \quad (2)$$

The rotor voltage equation gives the relation between rotor current and rotor flux linkage, taking into account that the squirrel-cage rotor represents a short circuit ($\underline{v}_R = 0$):

$$0 = r_R \underline{i}_R + \frac{d\underline{\psi}_R}{d\tau} + j(\omega_k - \omega_M) \underline{\psi}_R. \quad (3)$$

Magnetic coupling of rotor and stator is described by the flux linkage equations:

$$\underline{\psi}_S = x_S (\underline{i}_S + (1 - \sigma) \underline{i}_R) \quad (4)$$

$$\underline{\psi}_R = x_S (1 - \sigma) (\underline{i}_S + \underline{i}_R) = \underline{\psi}_S - \sigma x_S \underline{i}_S. \quad (5)$$

The leakage coefficient σ is defined in the following way:

$$\sigma = 1 - \frac{x_{SR}^2}{x_S x_R}, \quad (6)$$

with x_{SR} as mutual p.u. inductance between rotor and stator. Additionally, the rotor reference quantities are chosen so as to obtain (that does not mean "rotor has no leakage"!)

$$x_R = x_{SR} = x_S (1 - \sigma). \quad (7)$$

Finally, the mechanical equation describes the relation between mechanical velocity and torque. The p.u. starting time τ_M represents the moment of inertia. It is defined as the duration the IM takes to accelerate from standstill to reference angular velocity ($\omega_M = 1$) at reference torque. The difference of the IM's torque t_i and the load torque t_L accelerates the rotor:

$$\tau_M \frac{d\omega_M}{d\tau} = t_i - t_L . \quad (8)$$

The torque produced by the IM is determined by the product of (rotor) flux magnitude and perpendicular stator current component:

$$t_i = |i_s| |\psi_R| \sin \xi , \quad (9)$$

where ξ is the angle between stator current and rotor flux space-phasor.

From now on, these basic equations of the IM are mainly used in a rotor flux-oriented reference frame (indices x, y, Fig. 2). As subsequently shown, this reference frame enables splitting the stator current into a component determining the flux magnitude and a component defining the torque.

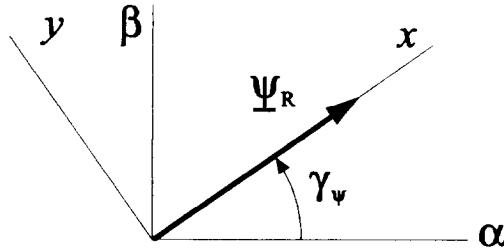


Fig. 2: Reference frame definition

In order to eliminate the rotor current (which is not measured and calculated), the rotor equation (3) is combined with (5). The result is a relation between stator current and rotor flux. Splitting the space-phasors into their x,y-components, the relation is expressed by two scalar equations:

$$i_{sx} = \frac{1}{(1-\sigma)x_s} \left(\tau_R \frac{d\psi_{Rx}}{d\tau} + \psi_{Rx} \right) , \quad (10)$$

with $\tau_R = \frac{x_R}{r_R}$

$$i_{sy} = \frac{\tau_R}{(1-\sigma)x_s} \psi_{Rx} \omega_{slip} , \quad (11)$$

with $\omega_{slip} = \omega_\psi - \omega_M$

τ_R is the rotor time constant, ω_{slip} is the difference between the angular velocities of flux and rotor.

As can be seen, i_{sx} is the component of the stator current that determines the flux via a first order delay (rotor time constant τ_R). On the other hand, i_{sy} determines the torque produced by the IM. At fixed flux level, these quantities are proportional, which can be seen by writing (9) in the following way:

$$t_i = i_{sy} \psi_{Rx} . \quad (12)$$

Combining (11) and (12), a direct relationship between the IM's torque and the slip angular velocity is obtained:

$$\tau_i = \frac{\tau_R}{(1-\sigma)x_S} \psi_{Rx}^2 \omega_{slip} . \quad (13)$$

Hence, at fixed flux level, the torque production can be controlled by the slip angular velocity without any delay ("indirect field orientation").

3 Field Orientation

Considering (12) and the unsatisfactory results of unintelligent systems, there is need for a concept that orients the stator current space-phasor \underline{i}_S to the rotor flux space-phasor $\underline{\psi}_R$. This approach is called field orientation and can be realized by applying equations (10) and (11).

Direct Field Orientation

An immediate application of field orientation is feasible using a current controlled IM (Schrödl, M., 1992). For this purpose, the rotor flux phasor must be known to determine the position of the x,y-frame. Proceeding from measurable machine quantities, the rotor flux can be determined by evaluating (2) - (5) ("flux model"). However, at least two of the measurable quantities "stator current", "stator voltage" or "rotor position" have to be measured. Because of the necessity of measuring analog quantities, a fully digital drive cannot be realized in that way.

Indirect Field Orientation with Inverse Flux Model

A flux model cannot only be used in the feedback branch as previously shown. The rotor flux can also be considered as an input quantity of a feed-forward flux model. The output of such an inversely used flux model is a proper reference quantity that can be produced by the VSI. In the following, a flux model is introduced that provides two voltage components in the x,y-frame as output quantities. The model does not need information about the actual stator current, so that machine control is possible without current measurement.

Proceeding from the basic equations considered in the x,y-frame with $\omega_k = \omega_\psi$, the stator flux in (2) and the rotor current in (3) can be eliminated:

$$\underline{v}_S = r_S \underline{i}_S + \sigma x_S \frac{d\underline{i}_S}{d\tau} + \frac{d\underline{\psi}_R}{d\tau} + j\omega_\psi (\sigma x_S \underline{i}_S + \underline{\psi}_R) \quad (14)$$

$$0 = \frac{1}{\tau_R} (\underline{\psi}_R - (1-\sigma)x_S \underline{i}_S) + \frac{d\underline{\psi}_R}{d\tau} + j(\omega_\psi - \omega_M) \underline{\psi}_R . \quad (15)$$

The rotor flux, which is an input quantity of the model, is usually kept constant in the base speed range. Hence, the first derivation of the rotor flux space-phasor disappears. In addition, the slip angular velocity is introduced in (15), leading with (7) to

$$\underline{i}_S = \frac{1}{x_R} \underline{\psi}_R + j \frac{\tau_R}{x_R} \omega_{slip} \underline{\psi}_R . \quad (16)$$

Replacing the current terms in (14) by (16) yields an equation without \underline{i}_S :

$$\begin{aligned} \underline{v}_S = & \frac{r_S}{x_R} \underline{\psi}_R + j r_S \frac{\tau_R}{x_R} \omega_{slip} \underline{\psi}_R + j \sigma x_S \frac{\tau_R}{x_R} \frac{d\omega_{slip}}{d\tau} \underline{\psi}_R + \\ & + j \omega_\psi \sigma \frac{x_S}{x_R} \underline{\psi}_R - \omega_\psi \sigma x_S \frac{\tau_R}{x_R} \omega_{slip} \underline{\psi}_R + j \omega_\psi \underline{\psi}_R \end{aligned} \quad (17)$$

Now, (17) can be split into x,y - components by using (7):

$$\begin{aligned} v_{Sx} &= \left(\frac{r_S}{x_S} - \omega_\psi \omega_{slip} \sigma \tau_R \right) \frac{1}{1-\sigma} \psi_{Rx} \\ v_{Sy} &= \left(\frac{r_S \tau_R}{x_S} \omega_{slip} + \sigma \tau_R \frac{d\omega_{slip}}{d\tau} + \omega_\psi \right) \frac{1}{1-\sigma} \psi_{Rx} \end{aligned} \quad (18)$$

These equations represent an inverse flux model that enables indirect field orientation without current measurement. The open-loop structure is illustrated in Fig. 3.

It is useful to set the input quantity "magnitude of rotor flux" $\psi_{Rx} = 1$ in the base speed range for good torque utilization. The other input quantity is the slip angular velocity ω_{slip} , which is obtained from superimposed speed- and position control loops. Hence, the torque produced by the IM is directly controlled according to (13). The sum of mechanical and slip velocity is ω_ψ which yields the flux angular position γ_ψ by integration. γ_ψ is used for transforming the stator voltage components from the x,y-frame to the stator-oriented α,β -frame. From α,β -components the phase voltages are calculated. By an appropriate PWM-algorithm (chapter 4), inverter firing commands are generated.

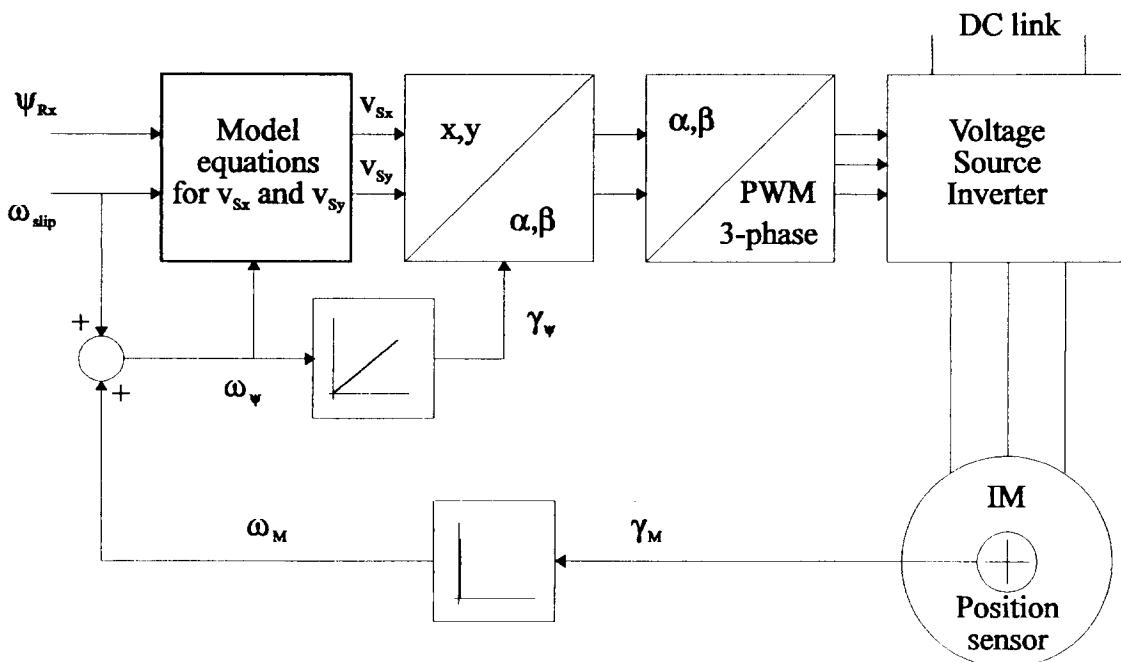


Fig. 3: Indirect field orientation with inverse flux model

Inverse flux model with current interface

For some purposes, it is useful to know the x,y-oriented reference values of the stator current since they correspond closely to the model input quantities ψ_{Rx} and ω_{slip} . The stator current space-phasor calculated in that way can be used for test purposes (for instance, to verify the model quality by comparing actual with reference current during control optimization). On the other hand, the argument of i_S is necessary for the inverter's dead-time compensation introduced in chapter 5 (Fig. 11).

A current interface can be installed in the model by including (10) and (11) with constant rotor flux magnitude assumed. ψ_{Rx} calculated from (10) and ω_{slip} obtained from (11) can be applied to the model equations (18) leading to

$$\begin{aligned} v_{Sx} &= r_S i_{Sx} - \sigma x_S \omega_\psi i_{Sy} \\ v_{Sy} &= r_S i_{Sy} + \sigma x_S \frac{di_{Sy}}{d\tau} + x_S \omega_\psi i_{Sx} \end{aligned} \quad (19)$$

Model equations (18) and (19) contain first derivatives with respect to time. Tests at the real-time system have shown that there is no noticeable difference in transient behavior of the drive with and without using the differential terms. Hence, with respect to disturbing effects, the differential terms are neglected.

4 Voltage Space Phasor Generation

In a drive system without current measurement, the quality of motor voltage control plays an important role. As the proposed control method calculates a reference voltage space phasor from flux and torque input commands, the inverter switching strategy has to fulfill the problem of approximating this general voltage space phasor by switching between the seven possible inverter voltage output states. Furthermore, the pulse sequence should optimize DC link voltage utilization.

In recent years, a variety of methods for PWM has been introduced (Fukuda, S. et al., 1990; Kolar, J.W., et al., 1991; Murai, Y. et al., 1987; Trenner, A., 1991). In the presented control method, a PWM method with zero-sequence space-phasor quantity will be used, since it is ideally suited to transient operation of the motor.

Restrictions due to DC link voltage

As shown in Fig. 4, the DC link voltage V_D (p.u. value v_D , respectively) restricts the possible magnitude of voltage space phasors generated by the inverter. Referring to half DC link potential ('0' in Fig. 4), $V_D/2$ represents the maximum phase voltage that can be obtained in any way.

According to (1), the voltage space-phasor in p.u. form is given by the following equation:

$$\underline{v} = \frac{2}{3} (v_1 + v_2 e^{j\frac{2\pi}{3}} + v_3 e^{j\frac{4\pi}{3}}) = |\underline{v}| e^{j\phi}. \quad (20)$$

The magnitude $|\underline{v}|$ is either 2/3 of p.u. DC link voltage in any "active" inverter switching state or zero in the two "short circuit" states (all inverter branches at the same potential). The possible arguments ϕ of the "active" states are $k \cdot 60^\circ$ ($k = 0, 1, \dots, 5$). A general voltage space phasor is hence approximated by a sequence of these possible space phasors.

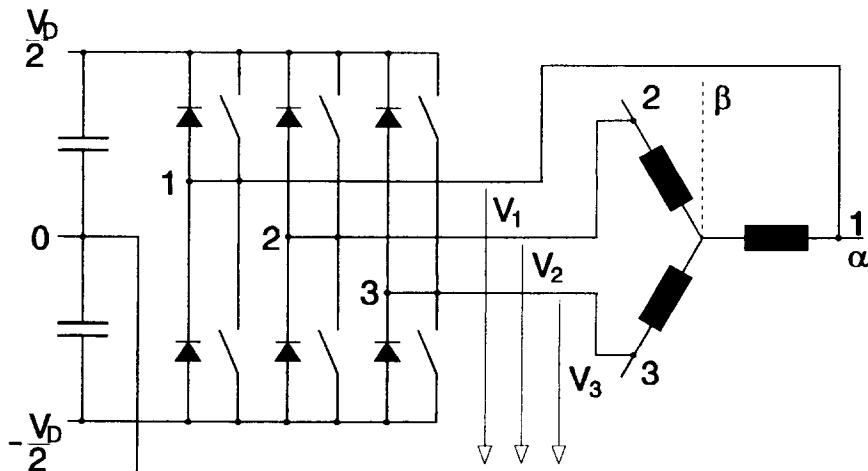


Fig. 4: Voltage Source Inverter

Zero-sequence voltage

A value v_0 added to the phase values v_1 , v_2 , v_3 does not cause any change in the space-phasor \underline{v} . Such a value is called zero-sequence voltage.

$$\underline{v} = \frac{2}{3} \left[(v_1 + v_0) + (v_2 + v_0) e^{j\frac{2\pi}{3}} + (v_3 + v_0) e^{j\frac{4\pi}{3}} \right] \quad (21)$$

A voltage difference between reference ground and neutral point of the machine (Fig. 4) is obviously a zero-sequence quantity and has consequently no influence on the voltage space-phasor. Hence, from space-phasor point of view it is correct to call V_1 etc. in Fig. 4 "phase voltages". On the other hand, adding zero-sequence voltages offers a degree of freedom in calculating reference phase voltage values from given voltage space phasor.

Simple PWM

Proceeding from the reference voltage space-phasor calculated by the control algorithm, the pulsewidths in phases 1,2,3 have to be determined. In the case of simple PWM, the duty-cycle of each phase, τ_{phase} (duration of positive DC link potential within one cycle at phase terminal related to cycle duration), is derived from pure projections of the space-phasor onto the phase axes. The following linear relation is obvious:

$$V_{phase} = \frac{1}{2} V_D (-1 + 2 \tau_{phase}), \quad 0 \leq \tau_{phase} \leq 1 \quad (22)$$

It can be shown that $\frac{1}{2} V_D \sqrt{3}$ is the maximum line voltage (averaged over a duty cycle) achieved. On the other hand, Fig. 4 shows that the line voltage between two machine terminals is either zero or $\pm V_D$. Obviously, simple PWM utilizes only 86.6 % of DC link voltage.

PWM with Zero-Sequence Quantity

By adding an appropriate zero-sequence voltage v_0 to the p.u. phase values v_1, v_2, v_3 , the utilization of the DC link voltage can be enhanced. The technique was introduced for pure harmonic functions at steady-state operation in (Buja, G. et al., 1975; Houldsworth, J.A., 1984), but it will subsequently be generalized for space-phasor consideration (Schermann, M., Schrödl, M., 1993).

For determining an appropriate zero-sequence voltage, the range $0 \leq \phi \leq 60^\circ$ is considered by studying the relations between voltage space phasor \underline{v} , phase voltages v_1 and v_3 and line voltage v_{13} . For optimal utilization, the line voltage (averaged over a duty cycle) has to reach full DC link voltage, as can be seen from Fig. 4. With this requirement, modified phase voltages v_{Z1} and v_{Z3} are introduced by adding a zero-sequence voltage v_0 :

$$\begin{aligned} v_{Z1} &= v_1 + v_0 = |\underline{v}| \cos \phi + v_0 \\ v_{Z3} &= v_3 + v_0 = -|\underline{v}| \cos(60^\circ - \phi) + v_0 \end{aligned} \quad (23)$$

However, to find appropriate v_0 -values, it has to be taken into account that the phase voltages (averaged over a duty cycle and related to "0" in Fig. 4) cannot exceed half DC link voltage. Choosing (without loosing generality) half DC link voltage as reference value for the p.u. voltages ($\frac{1}{2}v_D := 1$) leads with (23) to

$$\begin{aligned} v_{0\max} &= 1 - |\underline{v}| \cos \phi \\ v_{0\min} &= -1 + |\underline{v}| \cos(60^\circ - \phi) \end{aligned} \quad (24)$$

Each $v_0(\phi)$ that meets $v_{0\min}(\phi) \leq v_0(\phi) \leq v_{0\max}(\phi)$ is admissible (valid for arbitrary transient operation!). However, in the case of maximum DC link voltage utilization, $v_{0\min}(\phi)$ and $v_{0\max}(\phi)$ have a common tangent at $\phi = 30^\circ$ (Fig. 5). Hence, each admissible (space-dependent) function $v_0 = v_0(\phi)$ must have the same tangent at this point. The gradient of the tangent is:

$$\left. \frac{dv_{0\min}}{d\phi} \right|_{30^\circ} = \left. \frac{dv_{0\max}}{d\phi} \right|_{30^\circ} = \frac{1}{2} |\underline{v}| . \quad (25)$$

Admissible zero-sequence voltage functions $v_0(\phi)$ are for example the triangular function $v_{0,T}$ with a period of 120° and a magnitude $v_{0,T}(60^\circ)$ of

$$v_{0,T}(60^\circ) = \frac{\pi}{12} |\underline{v}| = 0.262 |\underline{v}| \quad (26)$$

or a cosine-function $v_{0,C}$ (Fig. 5). The tangent relation eq. (25) yields

$$v_{0,C} = -\frac{1}{6} \cos(3\phi) |\underline{v}| . \quad (27)$$

According to (25), there is no option to optimize the magnitudes of the cosine and the triangular function with respect to DC link voltage utilization. At steady-state operation, $v_{0,C}$ represents a third harmonic.

From given phase voltages v_{Z1}, v_{Z2}, v_{Z3} , respective pulselwidths τ_{phase} are calculated using (22). Although v_{Z1}, v_{Z2}, v_{Z3} do not exceed the value of 1, the DC link voltage utilization is increased from 86.6 % to 100 % compared to simple PWM. This is the same utilization as obtained by using space phasor modulation (Schermann, M. et al., 1993), which is another well-known modulating method.

Further increasing of the reference voltage space phasor leads to overmodulation. That means, there is no way to find a possible solution for a v_0 -value which fulfills the line voltage requirements (24) (at least in a range around $\phi = 30^\circ$). As an example, Fig. 5 shows the relations for a voltage space phasor of 110 % of maximum magnitude. In the shaded section,

there is no solution for an admissible zero-sequence voltage. However, around the "k.60°-corners" ($\phi = k \cdot 60^\circ$, $k = 0..5$), there exist solutions for v_0 .

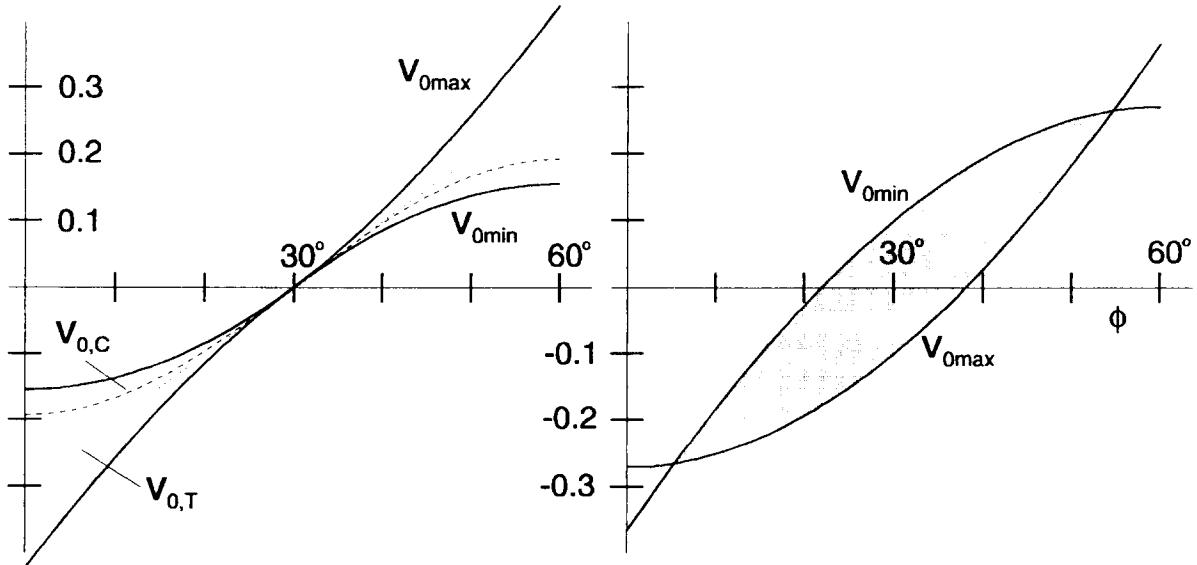


Fig. 5: Limits for the zero-sequence function

The short processing expense makes this algorithm well-suited for implementation in a fully digital drive system. However, the quality of inverter output waveforms depends significantly on the switching lag times between upper and lower semiconductors. In the following it is shown how to solve this problem within indirect field orientation.

5 Dead-time Compensation

The duration time from turning off one semiconductor of an inverter branch to turning on the counteracting semiconductor is called "dead-time" or "switching lag time" because it represents a delay necessary for preventing short circuits in the branch.

In recent years, inverter carrier frequency has been considerably increased in order to reduce motor losses, acoustic noise, torque ripple and so on. Modern fast-switching devices like FETs and IGBTs perform switching frequencies of up to 20 kHz. However, these devices still require switching dead-times of several μs (Murai, Y. et al., 1992), causing serious waveform distortion of inverter output voltages, particularly at low motor speed.

As shown in Fig. 6, the resultant pulse of the line voltage is very short in this operating range. Therefore, the dead-time and the pulselength are roughly of the same extent - the output pulselength can even disappear.

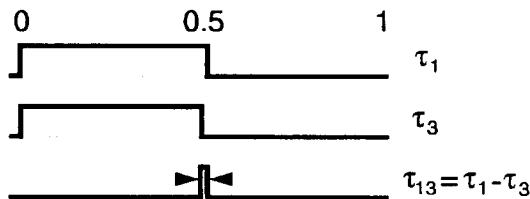


Fig. 6: Pulsewidths at low voltages

This effect occurs especially in feed-forward systems. Since the drive presented here has no closed current or voltage loop, an unadulterated output voltage is of great importance.

Waveform Distortion

Fig. 7 illustrates the dead-time effect considering one inverter branch loaded by an inductance. It shows that the voltage error depends on the polarity of the output current. With $i > 0$, the lower diode conveys the current, and the voltage error is negative. With $i < 0$, the upper diode conveys the current and the voltage error is positive. The magnitude of the voltage error is always full DC link voltage.

The error is accumulated during half a cycle of output frequency. That means, the higher the carrier frequency related to the output frequency, the higher the voltage error. Thus, the relation of carrier frequency to output frequency determines essentially the deviation from the ideal voltage waveform.

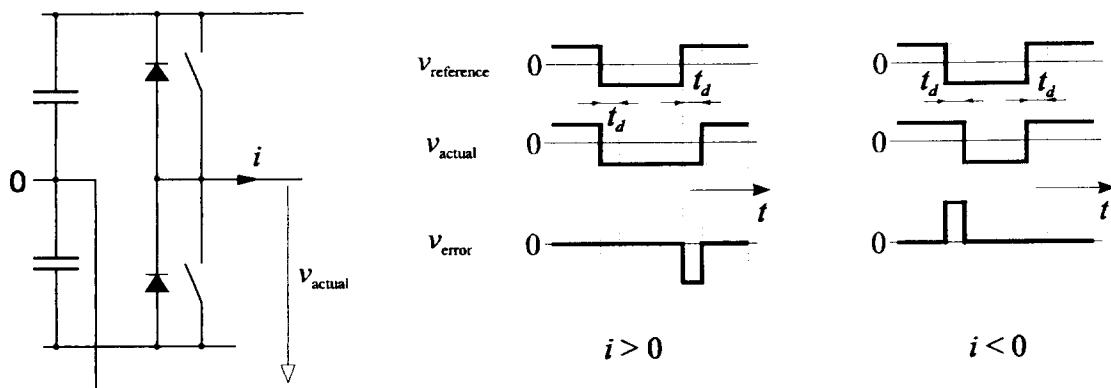


Fig. 7: Voltage error caused by dead-time

The effects of the uncompensated dead-time at the test stand are documented in Fig. 8. The VSI is controlled in feed-forward way at constant output frequency and voltage using PWM with zero-sequence voltage. (Other modulation methods yield similar or worse results).

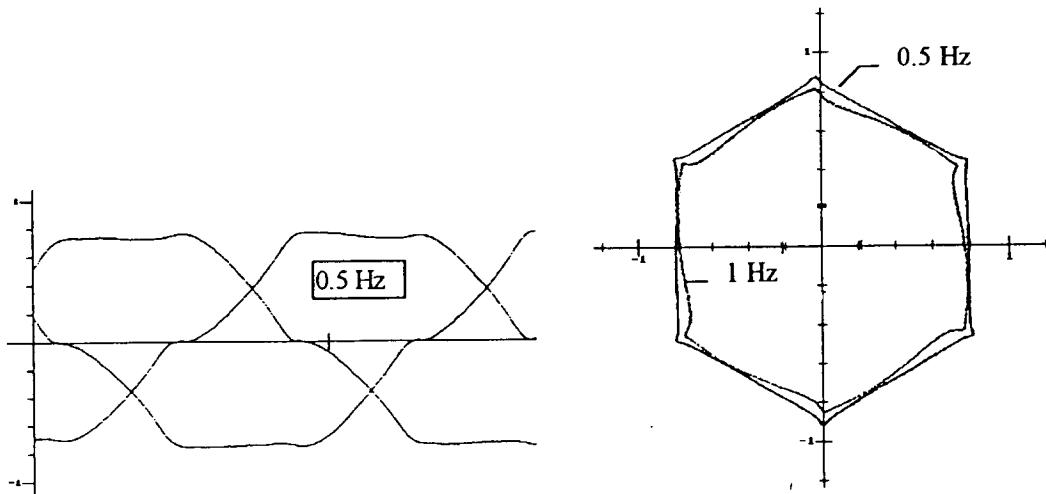


Fig. 8: Phase currents and current space-phasor at low frequencies

Although different PWM-methods cause different voltage and current waveforms, there is a common effect that occurs with all methods: The current seems to be clamped to zero when it crosses zero. Considering the complex plane of the current space-phasor, current clamping causes the current phasor to remain longer in certain positions. The clamping period depends strongly on the power factor, that is, the smaller the phase angle between voltage and current is, the longer the clamping period lasts.

A New Compensating Method

In general, there are both software and hardware counteractions that are proposed in the literature. Proceeding on the fact that the effect of dead-time is known, one can take the voltage error into account during calculation of the pulsewidths (Sukegawa, T. et al., 1991). Therein, compensating components are calculated that are added to the voltage phase values. However, only a constant correcting phasor is used within a 60° -sector.

In this paper, an extended method is presented at which compensating components for each angular position of current space phasor are calculated. Fig. 9 shows the realized correcting function at the test stand. The dotted line is a theoretical approximation of the dead-time effect, the continuous line is the result from an identification procedure.

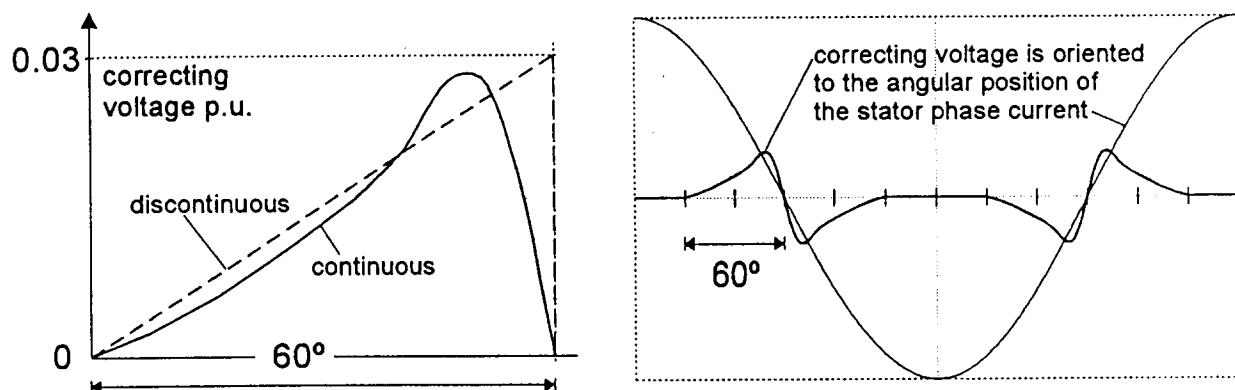


Fig. 9: Compensating function of one phase at the test stand

With this function, the current distortion is compensated correctly at each moment via a correcting voltage. The results of the method are illustrated in Fig. 10, showing current waveforms at 0.5 Hz (similar to Fig. 8). Even at lowest speeds, the distortion can be reduced almost ideally.

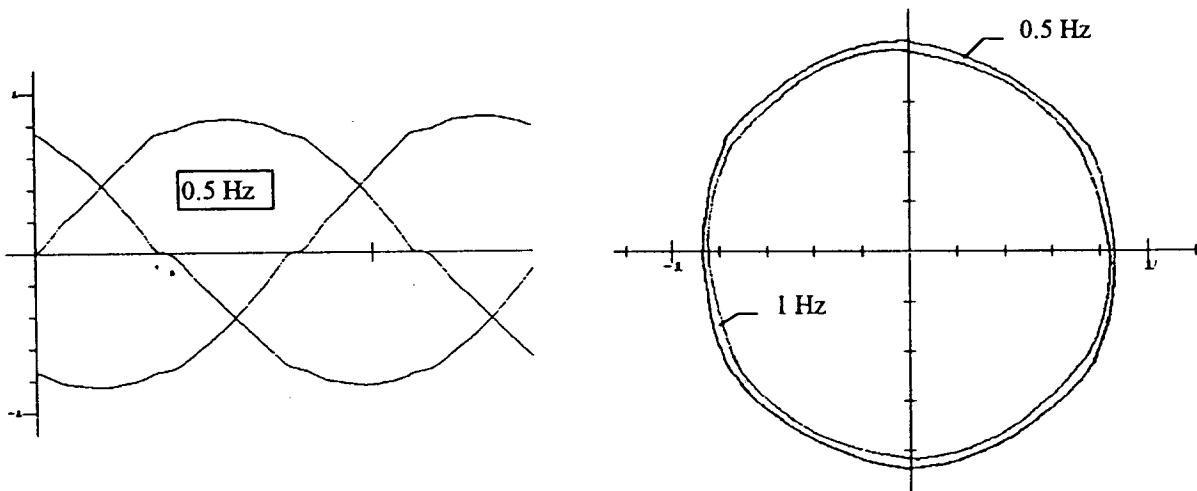


Fig. 10: Currents in the compensated system (measurement)

The method is independent of load variations due to the following fact. First, dead-time effects are critical only in the low-speed range since the dead-time voltage error is of the same magnitude as the (averaged) terminal voltage itself. In this range, the current is more and more determined by the ohmic stator resistance, and the current space-phasor direction

does not depend on the load condition (exactly at flux standstill). Second, because the compensating function is oriented to the angular position of the current space-phasor and is independent of the current magnitude, the system works as well at full load as at no load.

Inverse flux model with current interface and dead-time compensation

Obviously, the presented correcting method needs information about the actual current angular position. Since the current in the used system is not measured, it has to be calculated according to (4.9) and (4.10) from reference flux and slip angular velocity (or torque, respectively), yielding the control structure Fig. 11.

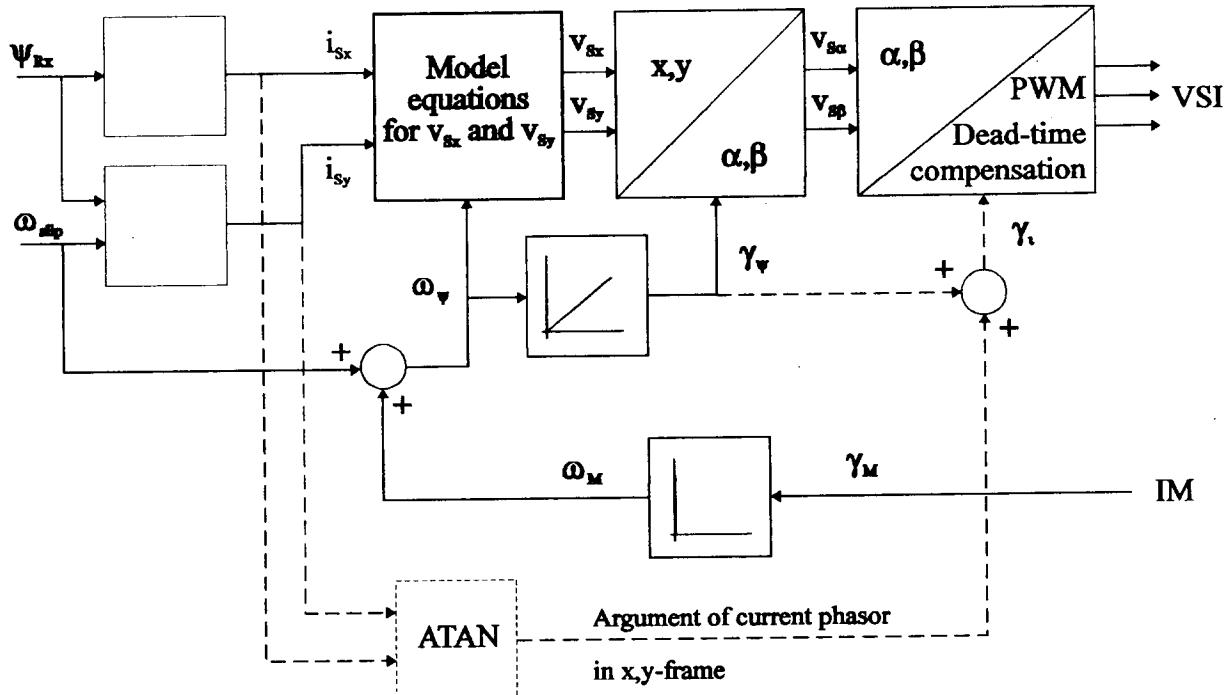


Fig. 11: Inverse flux model with current interface and dead-time compensation

The flux level in the induction motor can be kept constant only up to base speed due to limited DC link voltage. Beyond this limitation, the stator flux linkage has to be reduced in order to keep the magnitude of the terminal voltage constant (field-weakening range). However, the structure given in Fig. 11 ensures automatic field-weakening with a quite good dynamic behaviour and a satisfactory decoupling of flux and torque production (Schermann, M., 1994).

6 Measurement Results at Position Control

The behaviour of the realized system according to Fig. 11 (rated values of IM: 220 V, 3.5 A, 50 Hz, 4 poles) was tested in position-controlled mode. In the DSP-based system (low-cost CPU TMS 32014), all algorithms were implemented by software (indirect field orientation, PWM with zero-sequence voltage, dead-time compensation). Fig. 12 shows the response of the system to a step in the reference position and the reaction to a load step.

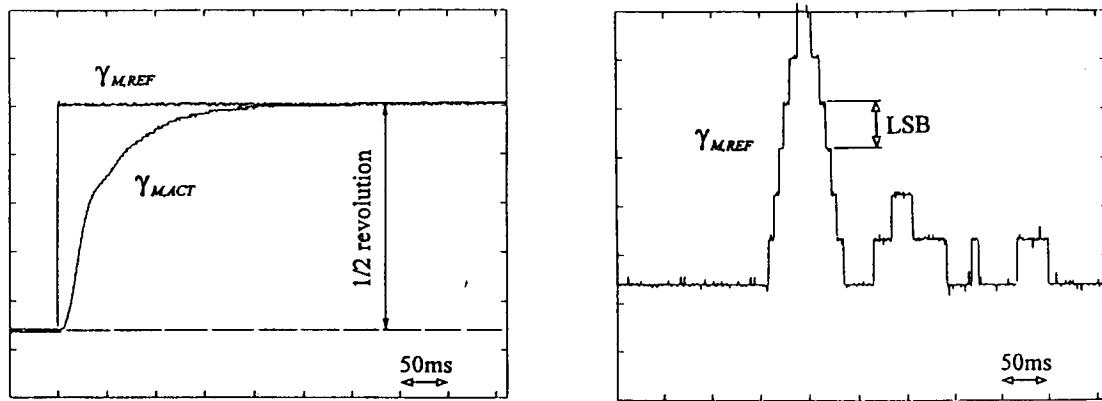


Fig. 12: Step response of the position-controlled system

- a) step in reference position (half a revolution)
- b) load step (torque from zero to rated value)

The phase currents of the position-controlled system operated at rated load are quite sinusoidal due to the presented dead-time compensation (Fig. 13). The current space phasor is almost a circle.

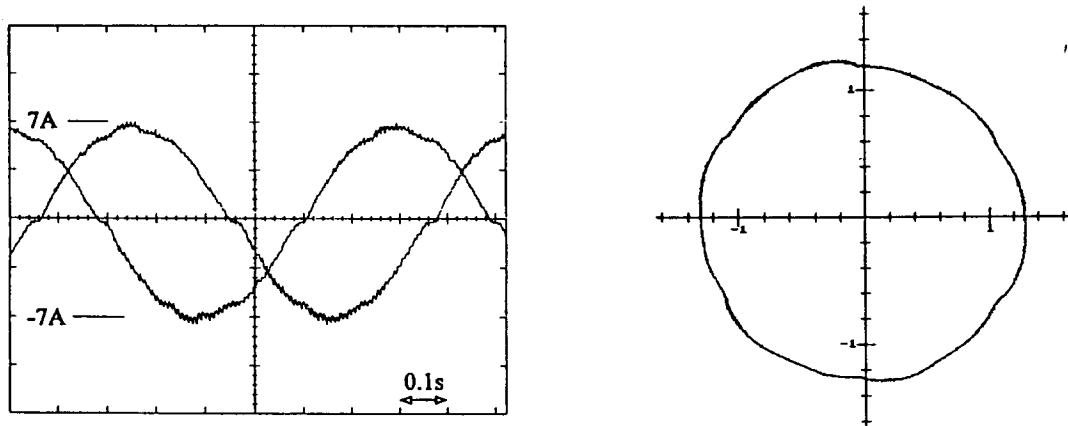


Fig. 13: Phase currents at rated load torque and constant rotor position

7 Conclusion

A method for fully digital control of induction machines has been presented. Indirect field-oriented control ensures highly dynamic operation. The inverter dead-time is compensated by a novel correcting algorithm based on calculation of the actual current space phasor. Measurement results show the good performance of the system especially in position-controlled operation. The system is hence well-suited to servo drives and robotic applications.

References

- Buja, G.; Indri, G., 1975, Improvement of Pulse Width Modulation Techniques, Archiv für Elektrotechnik 57, pp. 281-289.
- Fukuda, S.; Iwaji, Y.; Hasegawa, H., 1990, PWM technique for inverter with sinusoidal output current, IEEE Trans. on Power Electronics 5/1, pp. 54-61.
- Houldsworth, J.A.; Grant, D.A., 1984, The use of harmonic distortion to increase the output voltage of a three-phase PWM inverter, IEEE Trans. on Industry Applications 20/5, pp. 1224-1228.
- Kolar, J.W.; Ertl, H.; Zach, F.C., 1991, Influence of the modulation method on the conduction and switching losses of a PWM converter system, IEEE Trans. on Ind. Appl. 27/6, pp. 1063-1075.
- Kovacs, P.K., 1984, Transient phenomena in electrical machines, Elsevier Science Publishing Co., New York-Budapest.
- Murai, Y.; Ohashi, K.; Hosono, I., 1987, New PWM method for fully digitized inverters, IEEE Trans. on Ind. Appl. 23/5, pp. 887-893.
- Murai, Y.; Riyanto, A.; Nakamura, H., 1992, PWM strategy for high frequency carrier inverters eliminating current-clamps during switching dead-time, Proc. IEEE Ind. Appl. Annual meeting, pp. 317-322.
- Schermann, M., 1994, Digital induction motor positioning drive without measuring electrical quantities. Ph.D. thesis, University of Technology, Vienna.
- Schermann, M.; Schrödl, M., 1993, Methods of generating the voltage space-vector by fast real-time Pulsewidth Modulation, Proc. Power Conversion Conference, Yokohama, pp. 322-327.
- Schrödl, M., 1992, Sensorless control of A.C. machines, VDI Fortschrittberichte 21/17, VDI Verlag, Düsseldorf.
- Sukegawa, T.; Kamiyama, K.; Mizuno, K.; Matsui, T.; Okuyama, T., 1991, Fully digital vector-controlled PWM VSI-fed ac drives with an inverter dead-time compensation strategy, IEEE Trans. on Ind. Appl. 27/3, pp. 552-559.
- Trenner, A., 1991, Optimierung der Spannungspulsmuster und Sensitivitätsüberlegungen bei hochdynamischen Antrieben mit Asynchronmaschinen. Ph.D. Thesis, University of Technology, Vienna.

Neural Networks. Algorithms for Learning and Strategies of Application

Alexander Weinmann, Senior Member IEEE *

Manuscript received November 30, 1995

Abstract

The concept of artificial neural networks, their performance learning and training algorithms are presented. The algorithms are used to compare supervised and unsupervised learning algorithms as implemented in multilayer perceptron networks and Kohonen networks, respectively. The main properties of artificial neural networks lead to various strategies for combining artificial neural networks with real-world processes in order to learn the behavioural actions of the process or to carry out optimization, prediction and control problems.

1 Introduction. Principles in Cybernetics

Several real-world actions and behavioural patterns primarily are only known as black or grey boxes. That means that inside the boxes there exists no or only little information about the behaviour. In some cases it is feasible to find a description of the relations inside the black box, provided input and output signals in time domain (or frequency domain) are available and measurable. A basic problem, e.g., is to determine the transfer characteristic in order to describe the behaviour of the black box and to change one's knowledge from black to white.

Artificial neural networks are designed to be trained by input and output signals. Before learning they correspond to a black or grey box. Having trained they can be considered as a white box. The knowledge obtained is stored by many factors in a memory. These factors are denoted as weighting factors.

Training can be performed by one or several input output relations presupposing that they are not in contradiction.

Learning, training and natural evolution primarily are basic biological activities. They are used as a shining example for computer-aided modelling of activities in the field of engineering and economics.

Artificial neural networks are best suited for general modelling procedures since they can model any process irrespective if it is linear or nonlinear, time-dependent or time-varying.

*Expanded version of a two and a half hours lecture presented on November 24, 1995, in the "Engineering Management Executive Education Program" at Scientific Academy of Lower Austria, Krems, in cooperation with University of Technology, Vienna, and Oakland University, Rochester/Michigan

2 Basic Facts of Artificial Neural Systems and Neurocomputing

Artificial neural networks aim at the simulation of natural neural nets by being simulated on powerful computers, by parallel computation or by specific neural computers (*Lippmann, R.P., 1987; Anlauf, J., Gröber, A., und Knapp, B., 1995; Thurner, E., 1995; Budil, M., und Krenn, G., 1995; ECANSE*).

Originally, artificial neural networks are implemented for static relations between input and output signals. By using additional memory and delay operations, increments with respect to time can be included, hence enabling the artificial neural net to simulate dynamical behaviour of systems. A specified class of artificial neural networks are recurrent neural networks.

Classical systems comprise the analytic relations of mass, energy or information flow and are oriented to implement the relations in formulas and programs. Unlike to classical systems, artificial neural networks are based on a specific neural structure and multitude of weighting factors (*Hafner, S., Geiger, H., Kreßel, U., 1992; Ritter, H., Schulten, K., und Martinez, T., 1989*).

3 Use of Math Representation

Why does one use and even prefer descriptions and explanations based on mathematical formulas? The reason is that mathematical formulas are a very concise, clear and precise form; they provide a consistent overview; they allow for analytic calculation, for reduction, for the use of an equivalent structure (e.g. recursive formula derived from a one-shot formula); they lead to numeric evaluation and computation. Aiming at implementing artificial neural networks on digital computers, mathematical representations are an excellent preparation, irrespective if analysis or algorithms are preferred.

It should be emphasized that mathematical formulas are only describing a system and not explaining it, i.e., a physical or a biological or a neural process can only be modelled by a mathematical structure. Nevertheless, the math model is very useful within the limits of definition and validation.

4 Learning Laws. General Remarks

Artificial neural networks without being trained are black boxes. Only training and learning produces an active, useful and intelligent object just as a model, operator, controller, predictor etc.

Learning processes can be performed either by

- supervised learning, using a teacher (e.g. backpropagation algorithm), by
- reinforcement learning, characterized by a teacher who only decides if the learning process succeeded or failed, or, finally, by
- unsupervised learning, carried out without a teacher by means of a selforganizing activity, i.e., the network separates similar input signals into separate categories.

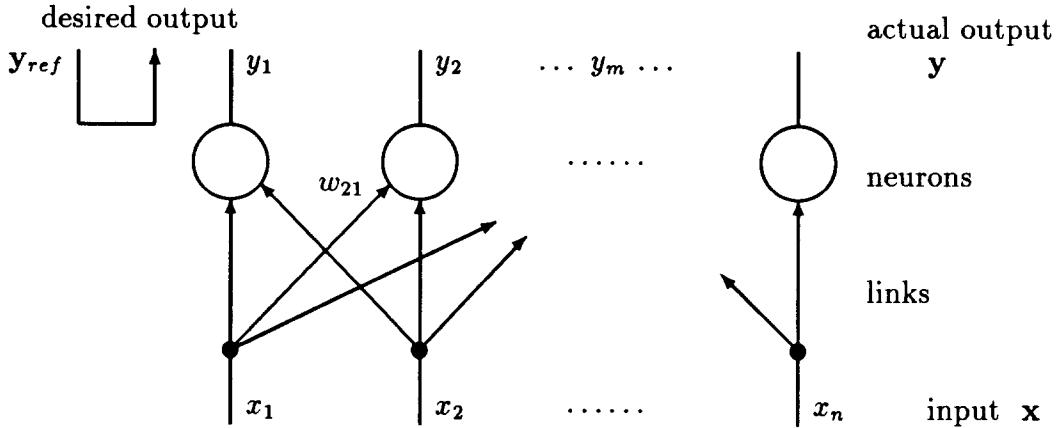


Figure 1: Linear associator neural network

5 Learning and Application Phase

During the learning phase, consistent data of input and desired output are forwarded to the system in a proper sequence. Data of sufficient quantity have to be supplied. The more data the better the result will be, usually. But putting more data into work requires a longer training phase.

In the application phase (also named as recall phase), new inputs, which have not yet been supplied, are fed to the network. The network will separate and identify to which class the new information belongs presupposing that the new data are within the scope of the trained data. Yet unknown inputs, i.e., data beyond the trained orbit, are associated with that class which is optimally close or probably remain unidentified at all.

Referring to Fig. 1 there are neurons (or nodes, cells or processing elements) responsible for the transformation or processing of information from the input to the output. The node in an artificial neural network is the simplified model of a biological neuron. Biological nets are characterized by thousands and millions of nodes, artificial ones by dozens and hundreds, only.

Additionally, links provide the transportation of information inside the artificial neural network and to and from the output and input. The direction is characterized by arrows. The vector-valued input \mathbf{x} corresponds to the dendrites in biological neurons. The neuron produces an output (axon) \mathbf{y} with entries y_i . Each output element y_i is modelled by a function of the sum of the weighted inputs; a function which is known as the activating function f_A . Learning is performed by adapting weighting factors w_{ij} , the so-called synapses, attaching the significance of the inputs to the nodes.

How is learning performed in principle? There is the possibility of creating new links or nodes, deleting links or nodes, altering w_{ij} , modifying the activating function f_A or, eventually, automatic searching of a better or an optimum structure of higher complexity.

An input vector (an input time series) \mathbf{x} with elements x_i is fed to the network. The actual output \mathbf{y} is to approach the reference (or desired output) y_{ref} during the training phase.

6 Linear Learning Laws

Denoting the input (entry) and output as $\mathbf{x} = \text{vec} \{x_i\} \in \mathcal{R}^n$ and $\mathbf{y} = \text{vec} \{y_i\} \in \mathcal{R}^m$, respectively, a weighting matrix is set

$$\mathbf{W} \triangleq (\mathbf{w}_1 \ \mathbf{w}_2 \dots \mathbf{w}_m)^T \in \mathcal{R}^{m \times n}, \quad \mathbf{w}_i \triangleq (w_{i1} \dots w_{in})^T \quad \forall i = 1 \dots m. \quad (1)$$

The output $\mathbf{y} = \mathbf{Wx}$ is assumed to be a linear combination of the inputs x_k (see Fig. 1). The output \mathbf{y} has to approach \mathbf{y}_{ref} , i.e., $\mathbf{y} \rightarrow \mathbf{y}_{ref}$ where \mathbf{y}_{ref} is the training setpoint. The question is if \mathbf{W} exists and which \mathbf{W} should be selected. If such a matrix \mathbf{W} exists then the learning procedure turns out as an operation of simple matrix multiplication.

6.1 Hebb's Outer-Product Algorithm

During the training phase, the linear associator neural network is to learn L pairs of input output vectors $\mathbf{x}_1 \ \mathbf{y}_1$ through $\mathbf{x}_L \ \mathbf{y}_L$. If the vectors \mathbf{x}_μ are orthonormal, we claim that the outer-product sum is the solution, i.e.,

$$\mathbf{W} \triangleq \sum_{\mu=1}^L \mathbf{y}_{ref,\mu} \ \mathbf{x}_\mu^T. \quad (2)$$

Then, in fact, the result is $\mathbf{y}_{ref,k} = \mathbf{Wx}_k$. This is Hebb's outer-product learning law. The proof is easy, since by orthonormality $\mathbf{x}_i \mathbf{x}_j = \delta_{ij}$ (Kronecker symbol)

$$\mathbf{Wx}_k = \left(\sum_{\mu=1}^L \mathbf{y}_{ref,\mu} \ \mathbf{x}_\mu^T \right) \mathbf{x}_k = \mathbf{y}_{ref,k} \cdot 1 + \mathbf{0} = \mathbf{y}_{ref,k} \quad (3)$$

in the recall phase¹. The weighting matrix \mathbf{W} in incremental form is

$$\mathbf{W}_{\text{new}} = \mathbf{W}_{\text{old}} + \mathbf{y}_{ref,k} \ \mathbf{x}_k^T, \quad (4)$$

as long as $L \leq n$ (*Hecht-Nielsen, R., 1989*).

Example:

$$\begin{aligned} \mathbf{x}_1 &= \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{x}_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{x}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \mathbf{y}_{ref,1} = \begin{pmatrix} 8 \\ 2 \\ 1 \end{pmatrix}, \quad \mathbf{y}_{ref,2} = \begin{pmatrix} 4 \\ 4 \\ 4 \end{pmatrix}, \quad \mathbf{y}_{ref,3} = \begin{pmatrix} -7 \\ 3 \\ 9 \end{pmatrix} \\ \rightsquigarrow \mathbf{W} &= \begin{pmatrix} 8 \\ 2 \\ 1 \end{pmatrix} (0 \ 1 \ 0) + \dots = \begin{pmatrix} 4 & 8 & -7 \\ 4 & 2 & 3 \\ 4 & 1 & 9 \end{pmatrix} \quad \text{and} \quad \mathbf{Wx}_k = \mathbf{y}_k = \mathbf{y}_{ref,k} \quad \text{is satisfied.} \quad \blacksquare \end{aligned}$$

6.2 Pseudo Inverse Formula

Joining \mathbf{x}_k and \mathbf{y}_k in a matrix \mathbf{X} and \mathbf{Y} , respectively, then as before $\mathbf{Y} = \mathbf{WX}$ and the aim is $\mathbf{Y} \rightarrow \mathbf{Y}_{ref}$. A performance measure g is used

$$g(\mathbf{W}) \triangleq \frac{1}{L} \|\mathbf{Y}_{ref} - \mathbf{WX}\|_F \quad (5)$$

¹If the inputs are only orthogonal then $\mathbf{y}_{ref,k} = \mathbf{Wx}_k (\mathbf{x}_k^T \mathbf{x}_k)^{-1}$.

where

$$\mathbf{X} \triangleq (\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_L) \in \mathcal{R}^{n \times L} \quad (6)$$

$$\mathbf{Y}_{ref} \triangleq (\mathbf{y}_{ref,1} \ \mathbf{y}_{ref,2} \ \dots \ \mathbf{y}_{ref,L}) \in \mathcal{R}^{m \times L} \quad (7)$$

$$\mathbf{Y} \triangleq (\mathbf{y}_1 \ \mathbf{y}_2 \ \dots \ \mathbf{y}_L) \in \mathcal{R}^{m \times L}. \quad (8)$$

The definition and the target of the optimum weighting matrix is

$$\mathbf{W}_{opt} \triangleq \arg \min_{\mathbf{W}} g(\mathbf{W}) \in \mathcal{R}^{m \times n}. \quad (9)$$

The result of the calculation as the result of the training phase is

$$\mathbf{W}_{opt} = \mathbf{Y}_{ref} \mathbf{X}^\dagger. \quad (10)$$

In the definitions above, $\|\mathbf{A}\|_F \triangleq \sum_{ij} A_{ij}^2$ is the Frobenius (or Euler) norm and \dagger denotes the pseudo inverse, see, e.g., Weinmann, A., 1991, Eq.(34.20).

Example:

(i) Underdetermined case $L \leq n$, column like \mathbf{X} : Using Eq.(9),

$$\mathbf{X}^\dagger = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \quad \mathbf{Y} = \mathbf{W}_{opt} \mathbf{X} = \mathbf{Y}_{ref} \mathbf{X}^\dagger \mathbf{X} = \mathbf{Y}_{ref} \quad \rightsquigarrow \quad \mathbf{Y} = \mathbf{Y}_{ref}. \quad (11)$$

In the special case of orthogonal \mathbf{x} , one has $\mathbf{X}^T \mathbf{X} = \mathbf{I}$, $\mathbf{X}^\dagger = \mathbf{X}^T$ and the outer-product algorithms is achieved.

(ii) Overdetermined case $L \geq n$, row like \mathbf{X} : Using Eq.(9) again,

$$\mathbf{X}^\dagger = \mathbf{X}^T (\mathbf{X} \mathbf{X}^T)^{-1} \quad \mathbf{Y} = \mathbf{W}_{opt} \mathbf{X} = \mathbf{Y}_{ref} \mathbf{X}^\dagger \mathbf{X} = \mathbf{Y}_{ref} \mathbf{X}^T (\mathbf{X} \mathbf{X}^T)^{-1} \mathbf{X} \quad (12)$$

$$\rightsquigarrow \mathbf{Y}_{ref} - \mathbf{Y} = \mathbf{Y}_{ref} [\mathbf{I} - \mathbf{X}^T (\mathbf{X} \mathbf{X}^T)^{-1} \mathbf{X}] \quad (\text{denoted as the residual}). \quad \blacksquare \quad (13)$$

Expanding $\mathbf{Y}_{ref} \mathbf{X}^\dagger$ in a Taylor series, $\mathbf{Y}_{ref} \mathbf{X}^T$ is the first term and equates to a matrix representaton of the outer-product matrix algorithm.

In the recall phase, one has $\mathbf{y} = \mathbf{W}_{opt} \mathbf{x}$ for an arbitrary input.

The incremental version corresponding to Eq.(10) also exists but is of complex character.

7 Performance Learning

We use the linear combination given by the scalar or inner product $y = \mathbf{w}^T \mathbf{x}$, the signals y_{ref} and \mathbf{x} are given, \mathbf{w} is unknown. If a vector-valued output \mathbf{y} should be achieved, the following derivations are rewritten to matrix-valued \mathbf{W} and vector-valued \mathbf{y} and \mathbf{y}_{ref} .

Considering a sequence \mathbf{x}_k of N vectors and the actual and the desired output y_k and $y_{ref,k}$, respectively, the mean squared error is

$$g(\mathbf{w}) \triangleq \frac{1}{N} \sum_{k=1}^N (y_{ref,k} - y_k)^2. \quad (14)$$

Taking a high number of vectors into account, the expectation operator $E\{\cdot\}$ can be applied

$$g(\mathbf{w}) \triangleq E\{(y_{ref,k} - y_k)^2\} = E\{(y_{ref,k} - \mathbf{w}^T \mathbf{x}_k)^2\} = \quad (15)$$

$$= E\{y_{ref,k}^2\} - 2\mathbf{w}^T E\{y_{ref,k} \mathbf{x}_k\} + \mathbf{w}^T E\{\mathbf{x}_k \mathbf{x}_k^T\} \mathbf{w}. \quad (16)$$

Searching the optimum, we use the gradient of the weighting function with respect to \mathbf{w} and evaluate its zero, i.e.,

$$g(\mathbf{w}) \rightarrow \min_{\mathbf{w}} \quad \rightsquigarrow \quad \frac{\partial g(\mathbf{w})}{\partial \mathbf{w}} = \mathbf{0}$$

$$\frac{\partial g(\mathbf{w})}{\partial \mathbf{w}} = -2E\{y_{ref,k}\mathbf{x}_k\} + 2E\{\mathbf{x}_k\mathbf{x}_k^T\}\mathbf{w} = \mathbf{0} \quad \rightsquigarrow \quad \mathbf{w} = \mathbf{w}_{opt} \quad (17)$$

$$\mathbf{w}_{opt} = (E\{\mathbf{x}_k\mathbf{x}_k^T\})^{-1} \cdot E\{y_{ref,k}\mathbf{x}_k\}. \quad (18)$$

Moreover, the difference quotient is

$$\frac{\Delta g(\mathbf{w})}{\Delta \mathbf{w}} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=1}^N 2(y_{ref,k} - y_k)(-\mathbf{x}_k) = -2E\{\delta_k \mathbf{x}_k\} \quad (19)$$

where $\delta_k \triangleq y_{ref,k} - y_k$.

Considering the case that the input signals \mathbf{x}_k are supplied stepwise, the increment $\Delta \mathbf{w}$ can be set proportional to the gradient in Eq.(19). *Widrow, B., and Hoff, M.E., 1960*, postulated a so-called least-mean-square learning law with an increment directly proportional to $\delta_k \mathbf{x}_k$. Then, the resulting Widrow-Hoff learning law is

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha \delta_k \mathbf{x}_k \quad (20)$$

which is always converging to \mathbf{w}_{opt} from any \mathbf{w}_o . The factor α is a positive constant in order to obtain minimum g . Widrow-Hoff learning is also known as adaptive linear element learning. It is numerically simple and only a global minimum exists (*Hecht-Nielsen, R., 1989*).

The change of weights is achieved proportional to the expectation of the difference between the current and desired (teaching) output δ_k multiplied by the input \mathbf{x}_k and a learning rate α .

The result of Eq.(20) also provides a clear insight into the general Hebb's learning rule. Consider the cell j being excited by cell i . The general Hebb's rule of learning (*Hebb, D., 1949,*) is then given by

$$\Delta w_{ji} = \alpha o_i s_j \quad (21)$$

where the link from i to j is used; o_i is the output of the sourcing cell i and s_j is the activation of the consecutive cell j . If both the o_i and the s_j are high then Δw_{ji} is to be increased.

A generalization leads to

$$\Delta w_{ji} = \alpha o_i (y_{ref} - y_i) \quad (22)$$

where y_{ref} is the teaching input or expected activation and y_i is the current activation (*Zell, A., 1994*).

8 Competitive Learning. Kohonen Learning. Selforganizing Maps. Associative Memory

The Kohonen model is based on the fundamental idea that neurons in a layer are sensitive for specified properties of signals. Assume that there exist many nodes with random distribution of properties. The property is considered vector-valued and denoted as \mathbf{w}_i for

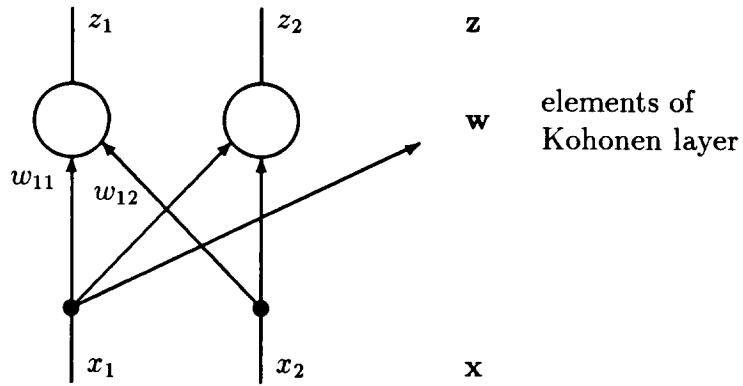


Figure 2: Kohonen layer

the neuron i . During the learning phase the input vector \mathbf{x} is fed to all the neurons. There will be a neuron the sensitivity of which is closest to the input, i.e., there is an output node which is activated most. The difference between the input \mathbf{x} and the weighting vector \mathbf{w} is minimum and is detected by their difference. The vector-valued difference is treated by any norm, e.g. the Frobenius norm. This optimum reaction is also transferred to neurons in some local neighbourhood, thus effecting their adaptation to the property of the optimum neuron.

If another input is supplied to the Kohonen network for teaching purpose, another group will react optimally and store its specific property. In such a way, by persistent teaching each significant property of the input signal is put into a specified group of neurons. Without supervision the neurons adapt to the input (*Kohonen, T., 1989*).

The processing elements compete at any input \mathbf{x} entering. Which of them has its weight vector \mathbf{w}_i closest to \mathbf{x} (measured by a certain scalar distance)? The closest element i is selected, in order to perform learning. An algorithm can be stated as follows: The winner takes all, i.e., the winning neuron i emits a signal $z_i = 1$, the others $z_{j(\neq i)} = 0$, see Fig. 2.

Unlike other learning algorithms the weight \mathbf{w}_i is to approach \mathbf{x} . Widrow and Hebb are trained by a reference, Kohonen is able to dispense with a reference (or teacher).

A distance metric performance produced by each element of the Kohonen layer is

$$g_i = \|\mathbf{w}_i - \mathbf{x}\|_F . \quad (23)$$

The Kohonen learning law results from

$$\mathbf{w}_i \text{ new} = \mathbf{w}_i \text{ old} + \alpha(\mathbf{x} - \mathbf{w}_i \text{ old})z_i \quad (0 < \alpha \leq 1) . \quad (24)$$

For the winning element one has $z_i = 1$ and $\mathbf{w}_i \text{ new} = (1 - \alpha)\mathbf{w}_i \text{ old} + \alpha\mathbf{x}$. For all the losing elements $z_{j(\neq i)} = 0$ is achieved and the weights are not altered, i.e., $\mathbf{w}_j \text{ new} = \mathbf{w}_j \text{ old}$. Only the nearest weight vector is attracted.

In the beginning of the training phase, the learning factor α is chosen near 1 in order to guarantee quick training start-up, in advancing the training procedure α is reduced to a small number (choice of receding learning rate).

Example: Consider four neurons with initial sensitivities \mathbf{w}_1 through \mathbf{w}_4 . Only the closest neuron 2 is changed from $\mathbf{w}_{2\text{old}}$ to $\mathbf{w}_{2\text{new}}$, according to Eqs.(23) and (24), see Fig. 3. No additional relation to the neighbouring neurons is taken into consideration. ■

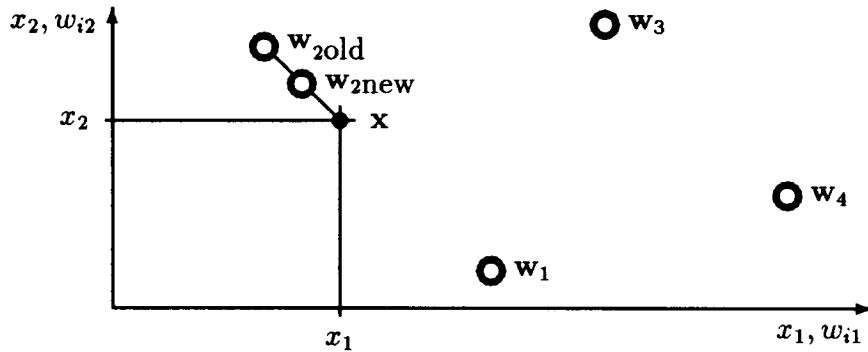


Figure 3: Weight plane and adaptation of the weights of neuron 1 through 4 when a single \mathbf{x} is supplied and used for learning with $\alpha = 0.5$

The Kohonen network can be considered as a learning vector quantization, valuating the relation to the neighbours. Only a single layer of active neurons is needed. In the space of input vectors, \mathbf{w}_{ref} should be distributed in such a way that the input vectors which are expected during learning phase can be represented by \mathbf{w}_{ref} .

In addition, (i) one can attain all the losing elements with an output $z_i \neq 0 > 0$ in order to include all the losing elements in the process of learning or (ii) one selects losing neighbouring neurons in a predetermined region where attraction should take place. In most Kohonen networks the neurons are distributed in a grid of low dimension (1, 2 or 3).

Another representation of the learning rule is

$$\|\mathbf{x} - \mathbf{w}_c\|_F \stackrel{\Delta}{=} \min_i \|\mathbf{x} - \mathbf{w}_i\|_F \quad \text{where } c = \arg \min_i \|\mathbf{x} - \mathbf{w}_i\|_F .$$

9 Nonlinear Learning Laws

Hitherto, neurons have been considered in a structure of being equally entitled in one single layer. In Fig. 4, a multilayer network is depicted. The layer organization stands for the property that within the layer there is no connection. All the connections are referred to another layer or to the input or output. A feedforward network is characterized by only one direction of data transfer, corresponding to the resulting direction from the input to the output. This structure or pattern is also known as the perceptron.

The sensors are connected to the input layer with weighting unity, feeding it thus with the input \mathbf{x} . The output layer with output \mathbf{y} is connected to the actors. In between, there is one or there are more hidden layers (Ritter, H., et al. 1990). Referring to Kolmogorow, for the approximation of a continuous function only one hidden layer is required. By means of a perceptron all linear separable functions are trainable. Most of the neurons are not connected to input or output and, hence, are named hidden. The activity of a neuron in each layer is denoted by s and an appropriate index, i.e., i , j , k for output, hidden and input layer, respectively. The input is \mathbf{x} where $\mathbf{x} = \text{vec}[x_k] = \text{vec}[s_k]$. Analogously the output is $\mathbf{y} = \text{vec}[y_i] = \text{vec}[s_i]$. Using the activation function f_A , the relations are

$$s_i = f_A \left(\sum_j w_{ij} s_j \right) \quad \text{and} \quad s_j = f_A \left(\sum_k w_{jk} s_k \right) . \quad (25)$$

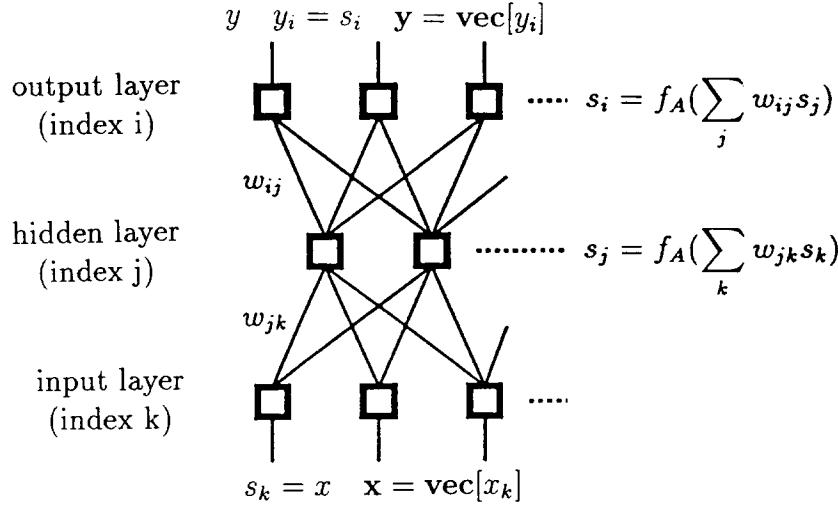


Figure 4: Neuronal net with three layers (Perceptron)

In addition, a bias or offset input could be taken into account in order to evaluate an overall displacement. The activating function f_A is a nonlinear function, especially a selected sigmoid function. Alternative activating functions are identity, scaling (linear) or threshold (linear or hard saturation). In order to emphasize the nonlinear activating function, the symbols of the nodes are squares in Fig. 4.

The aim of the neural network is to learn the weighting factors w of each path. Assuming a supervising performance g , a gradient method facilitates the adaptation of the weighting factors. A wide-spread method is the backpropagation method by which the weighting factors are learned progressively, starting from the supervised output and its associated weighting factors, and proceeding to the input weighting factors. The approach operates in backward direction. Based on the available training data, a resulting performance is defined by a quadratic function g , see Eq.(26). The increments are derived from its gradient.

First of all, the relations are written in such a form as if the the output, input and weighting factors were given. But these relations are only utilized in order to find out the weighting factors evaluating the given training input and output. The learning algorithm uses a multitude r of inputs characterized by superscript μ . The purpose of the learning algorithm is to learn the weighting factors w_{ij} und w_{jk} supervised by a performance g to be minimized. Learning is performed by comparison and incremental change in the direction of the gradient, and that in the following steps:

- input data are fed to the input,
- then initialize w_{ij} and w_{jk} and find y and $y_{ref} - y$,
- run the following gradient descent (backpropagation algorithm)

$$g = \sum_{\mu=1}^r \sum_i [y_{ref,i}^\mu - s_i(\mathbf{x}^\mu)]^2 \quad (26)$$

$$\text{where the output activity is } s_i = f_A(s_j) , \quad (27)$$

$$\text{the hidden activity is } s_j = f_A(s_k) \quad (28)$$

$$\text{and the input (and its activity) } \mathbf{x} = \text{vec}[s_k] . \quad (29)$$

The weighting factors w_{ij} are changed referring to

$$\Delta w_{ij} \propto -\frac{\partial g}{\partial w_{ij}} \quad \rightsquigarrow \quad \Delta g \propto \sum_{ij} \frac{\partial g}{\partial w_{ij}} \Delta w_{ij} \propto -\sum_{ij} \left(\frac{\partial g}{\partial w_{ij}} \right)^2 \leq 0 . \quad (30)$$

Differentiating the performance with respect to the weighting factors associated with the output, one has

$$\frac{\partial g}{\partial w_{ij}} = -2 \sum_{\mu} [y_{ref,i} - s_i(\mathbf{x}^{\mu})] \frac{\partial s_i(\mathbf{x}^{\mu})}{\partial w_{ij}} = - \sum_{\mu} [y_{ref,i} - s_i(\mathbf{x}^{\mu})] f'_A \left(\sum_{\nu} w_{i\nu} s_{\nu} \right) s_j \quad (31)$$

where ' is the derivation with respect to the argument. The activating function f_A is considered differentiable. Calculating the derivative with respect to the weighting factors associated with the input, the intermediate differentiation with respect to s_j has to be taken into account. Out of this, the result is

$$\begin{aligned} \frac{\partial g}{\partial w_{jk}} &= -2 \sum_{\mu} \sum_i [y_{ref,i}^{\mu} - s_i(\mathbf{x}^{\mu})] \left[f'_A \left(\sum_{\nu} w_{i\nu} s_{\nu} \right) \right] w_{ij} \frac{\partial s_j}{\partial w_{jk}} = \\ &= -2 \sum_{\mu} \sum_i [y_{ref,i}^{\mu} - s_i(\mathbf{x}^{\mu})] \left[f'_A \left(\sum_{\nu} w_{i\nu} s_{\nu} \right) \right] w_{ij} \left[f'_A \left(\sum_{\nu} w_{j\nu} s_{\nu} \right) \right] s_k . \end{aligned} \quad (32)$$

From the foregoing analysis the gradients $\frac{\partial g}{\partial w}$ result, yielding the basis for changing and learning the increments Δw in Eq.(30). The operations in Eqs.(31) and (32) demonstrate the numerical effort. Parallelling the computations is an urgent need as far as time for learning is concerned.

The elements of the vectors \mathbf{y}_{ref} and \mathbf{x} are given. It seems adequate to use a larger set of input and output vectors for better results. The topology of the network is predetermined according to the designer's experience. If the training procedure does not yield adequate results valuated by g , the topology of the pattern has to be changed.

10 Alternative Algorithms

10.1 Recurrent Neural Networks

Recurrent artificial neural networks are characterized by the property that information is also transferred in the opposite direction (feedback). By this means, information processing and memory is combined and no external memory is needed. If the network is only partly recurrent these networks are known as Elman networks. The input of the hidden layer is fed both by the artificial neural network input and the preceding output which is stored in so-called context cells. In such a way the state-space description of a discrete-time system is implemented easily by means of an artificial neural network.

10.2 Backpercolation

The backpercolation method differs from the backpropagation method in that each neuron is equipped with its own performance. Unlike the backpropagation method it is said to be capable of breaking through hills of the overall performance function in order to avoid local minima and to arrive at the global minimum.

10.3 Counterpropagation

Supervised learning by counterpropagation is based on a Kohonen and a Grossberg layer. Grossberg layer neurons possess the same activating functions as Kohonen networks except the winner-takes-all characteristic. The property of such a network is quicker learning in comparison to exclusive Kohonen and perceptron networks. The operation is similar to a generalized look-up table.

10.4 Radial Basis Functions

Radial basis functions are feedforward networks with only one hidden layer. Each neuron of this layer is characterized by a radial symmetric activating function. The number of neurons corresponds to the number of training samples. The activating cluster is characterized by

$$y = \sum_i^N h_i e^{-\gamma \|x - x_i\|_F} \quad x \in \mathcal{R}^n \quad (33)$$

where the x_i 's are the given and supporting input samples. The setting γ determines the shape of the plane of the radial basis function, the h_i 's are unknown. The network produces a transformation from \mathcal{R}^n to \mathcal{R} by portraying x_i to y . During the training phase for given y_μ and x_μ

$$y = y_\mu \quad \leadsto \quad y_\mu = h_\mu + \sum_{i \neq \mu}^N h_i e^{-\gamma \|x_\mu - x_i\|_F} \quad \forall \mu = 1, 2, \dots, N. \quad (34)$$

The coefficients h_μ are achieved in a direct (non-iterative) way, since Eq.(34) is a system with N unknowns in N linear equations. For a given data more than N , the pseudo inverse formula of Eq.(10) is adapted with respect to h_i . The exponential activity function provides remarkable contribution only if x is close to x_i which can be considered as an advantage in order to guarantee good approximation properties.

Example: Radial basis functions, $N = 2$:

$$x_i : \quad x_1 = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \quad x_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \gamma = 1. \quad (35)$$

Information for the training phase:

$$x_\mu : \quad \mu = 1, x_1 = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \quad y_1 = 2.8; \quad \mu = 2, x_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad y_2 = 2. \quad (36)$$

Calculation for the training phase:

$$\mu = 1 : \quad y_1 = 2.8 = h_1 + h_2 e^{-\|(-1)-\begin{pmatrix} 1 \\ 2 \end{pmatrix}\|_F} = h_1 + h_2 \cdot 0.059. \quad (37)$$

Similarly, $y_2 = 2 = h_2 + h_1 \cdot 0.059$. The result is $h_1 = 2.69$ and $h_2 = 1.84$. The resulting plane $y(x)$ is depicted in Fig. 5. The training samples correspond to the peaks of the plane of the radial basis functions.

■

10.5 Genetic Algorithms

Genetic algorithms are based on transposing parameters of technical systems into clusters of genes. Starting an optimization problem, a group of parameter constellations is selected

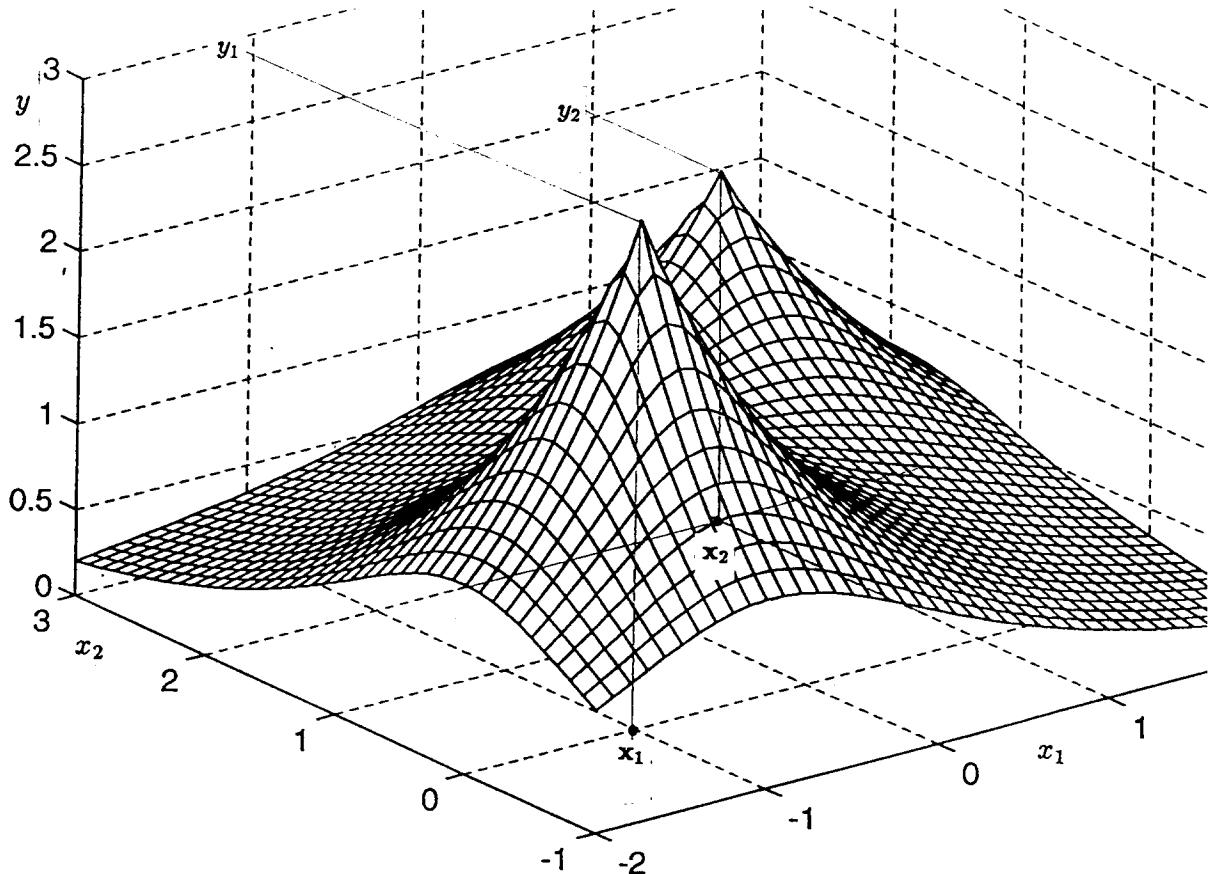


Figure 5: Radial basis function network

and defined by genes. This corresponds to a group of individuals. The individuals are subject to an evolution by computer simulation. Natural evolution is simulated by the transfer of properties from generation to generation, by natural heredity from parents to children. Both the natural selection and the natural mutation are simulated. Each individual is characterized by a fitness function. Selection is based on this fitness function, in order to find the probability of survival and to decide which individual should survive. The principle is "survival of the fittest" (see, e.g., Grünberger, T., 1995).

11 Strategies For Combining Artificial Neural Networks and Processes

In technical systems, one is used to combine real-world systems with models and to adapt the model optimally for the purpose of identification, adaptive controllers and related topics. Management systems, e.g., can be considered as control loops with a manager acting as a controller. The control action is based on his decisions after having evaluated the industrial and economic data of his enterprise and predicting the market. That means that he is confronted with a remarkable amount of data, partly fuzzy data, and with the efficiency and correctness of data reduction system.

The object of any learning strategy is to generate and to train a cluster or network, operating as a model, controller and/or compensator.

Learning is performed by combining the artificial neural network and the real process or system S. The methods of combining are similar to those in adaptive controllers (Miller, W.T., Sutton, R.S., and Werbos, P.J., 1990; Narendra, K.S., and Parthasarathy, K., 1990).

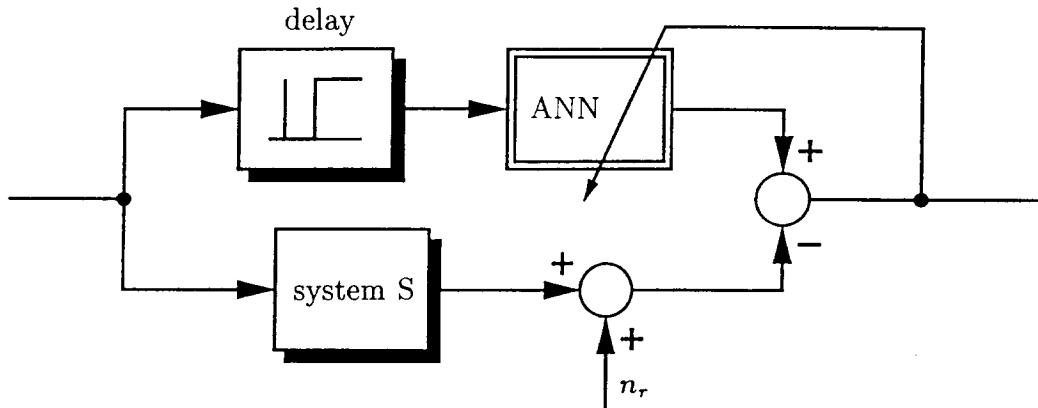


Figure 6: Parallel connection of process and artificial neural network ANN

As carried out in the following subsections, there are various possibilities for combining and comparing the artificial neural network and the real-world process (*Miesbach, S., und Schürmann, B., 1991*).

The artificial neural network and the process inherently are nonlinear, hence it is difficult to analyze, predict and design the stability degree both during the adaptation (learning) and with respect to the performance robustness (*Weinmann, A., 1991*).

11.1 Parallelling Including Predictive Facilities

Combining in parallel means identical input signals. The error between system and artificial neural network is utilized for adapting the artificial neural network, see Fig. 6. The arrow crossing a block element represents an action in that the parameters of the block are adapted, governed by the link feeding the arrow. The artificial neural network learns the input output relation of the system S or the order and the coefficients of the differential equation of S. The input to the artificial neural network is given by the last input and output values of S. The number of learning steps increases with the order of the difference equation (*Mistry, S.I., and Nair, S.S., 1994*).

By preconnecting a dead time delay, the artificial neural network is used as a predictor, see Subsection 11.5.

The system S is given either by a process G or by a skilled operator K. Note that the measured signals are under the influence of measurement noise n_r .

11.2 Inverse Model Associated with the Output or the Input

By connecting an artificial neural network with inverse behaviour to the output of a process G (Fig. 7), one can expect an output signal of the artificial neural network which is equal to the input of the process. The error e is used for adapting the artificial neural network.

Abb. 7 is not “goal oriented”, since the output of the process is not used for comparison purposes. The input u_B has to be selected carefully in order to offer an adequate excitation to the process; adequate as far as amount, velocity or frequency richness are concerned.

In Fig. 8 the inverse ANN model is associated with the input of the process, presupposing that there exists a sufficient initial state of the artificial neural network in order to supply a suitable input to the process. For selecting the signal u_S in Fig. 8, some experience and initial identification results should exist. This choice is characterized by the opposite advantages in comparison to Fig. 7.

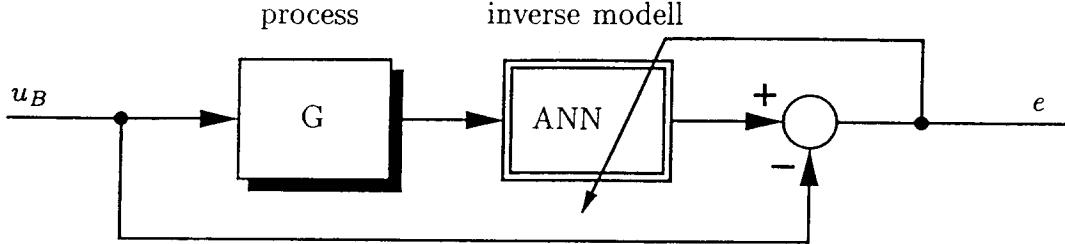


Figure 7: Inverse model trained by the observed plant output

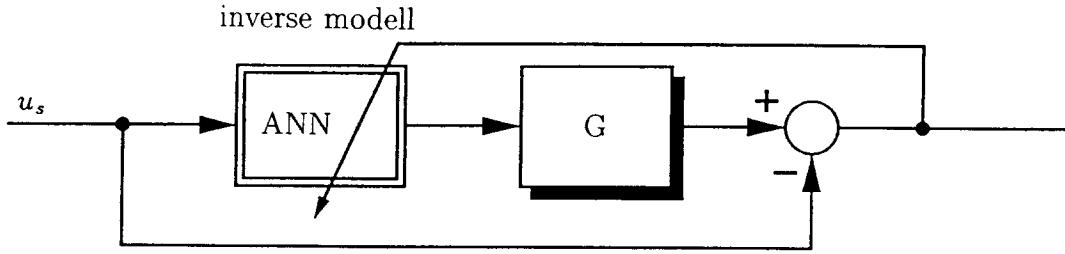


Figure 8: Inverse model, training executed in the input path of the process

In most cases only reinforcement learning can be applied because the process is observed by the input or output, only.

11.3 Internal Model Control

Inverse and parallel model are combined in a block assembly named *Internal Model Control* in Fig. 9.

If $\text{ANN}_2 \rightarrow G(s)$ has been settled perfectly then v equals the reaction w_d of the disturbance to the process $G(s)$. The signal v also consists of uncertainties in $G(s)$ and differences between ANN_2 and $G(s)$; as far as they could have been excited by u .

The reference y_{ref} is augmented by w_d , and u will turn out as an appropriate manipulating variable in order to enable y to approach y_{ref} . This will happen in the ideal case of $\text{ANN}_1 \rightarrow \frac{1}{G(s)}$. The transfer function $T(s)$ comprises the desired input output relation from y_{ref} to y . The signal e is used for adapting ANN_1 , the feedback signal v for changing ANN_2 (Hunt, K.J., Sbarbaro, D., Zbikowski, R., and Gawthrop, P.J., 1992; Morari, M., and Zafriou, E., 1989).

11.4 Optimization

For optimization purposes an arrangement of Fig. 10 can be applied. First, an optimum actuating variable u_{opt} has to be found by using an artificial neural network of the process G , see Fig. 10a. This is performed by time-lapse in most cases. In parallel, the deviation $e(k)$ is evaluated which actually results from G being excited by u_{opt} , see Fig. 10b. This value $e(k)$ is utilized as the error $e(k - 1)$ for the next optimization step.

11.5 Predictive Control

Consider Fig. 6 once more and set the delay equal to βT where T is the sampling interval characterizing two consecutive computations and β is a positive integer. According to the

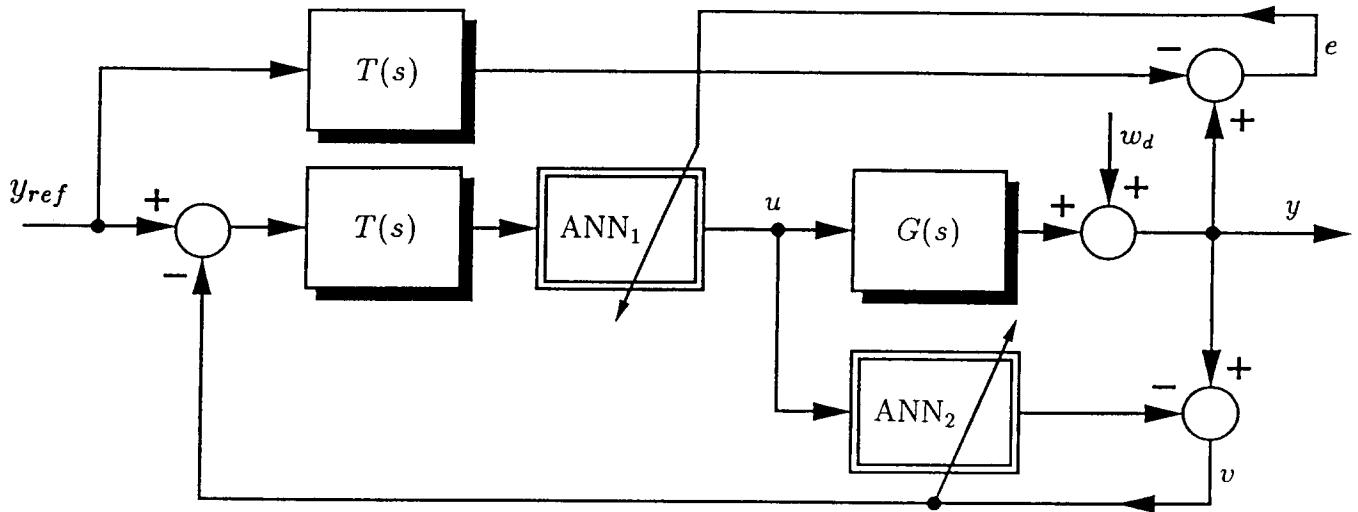


Figure 9: Internal model control

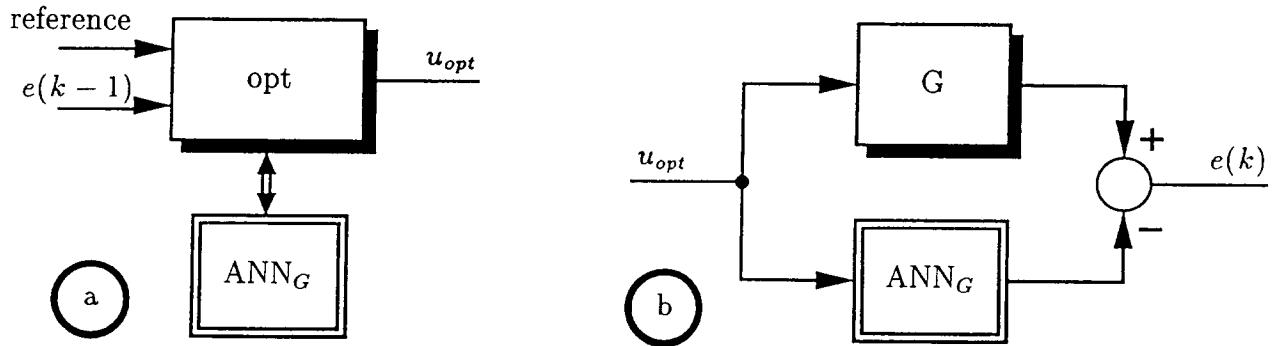


Figure 10: Optimization carried out by an artificial neural network ANN

fact that the input of the artificial neural network is delayed artificially but the output is compared to the real system output, the artificial neural network dynamics will try to compensate for the delay as long as the system is not confronted with unpredictable situations.

When transferring the artificial neural network parameters to an additional network ANN_P , which is not depicted in Fig. 6, excited by the undelayed input of the system S , then the output of ANN_P is a β -step-ahead predictor of S .

12 Special Features of Artificial Neural Networks

Above all, artificial neural networks are characterized by the following properties:

- Application is favourable if the process is not known sufficiently or if the mathematical model is incomplete as far as the nonlinear transformation function, the coefficients or order of the differential equation are concerned.
- Many input and output data of the process are available in order to train the artificial neural network.

- Evaluation by an artificial neural network is performed by highly parallel computation, hence the results, e.g. for image detection or pattern recognition, are found out promptly even if each computational operation is relatively slow.
- Knowledge about the process need not be put into formulas when initiating the system, e.g., pattern recognition and selection becomes feasible. On the other hand, it is difficult to insert preknowledge by formulas.
- The main fields of application are given by data compression in image processing, by classification and by function approximation.
- Diagnosis of failures is feasible, e.g., detecting the type of gear defect out of the gear noise (*Schöneburg, E., 1992,*). A “therapy” (maintainance or repair) can be suggested. Care has to be taken of process measurement noise.
- Prognosis referring to quality and accuracy is feasible.
- Combination with fuzzy control is advantageous, e.g., learning or improving the membership functions via an artificial neural network, or demonstrating and illustrating the artificial neural network results by fuzzy methods (via membership functions or inference) (*Preuß, H.-P., und Tresp, V., 1994*).
- Increase of knowledge can only be performed by learning. The analysis of the knowledge obtained is hardly possible. Deductions are difficult to be drawn.
- Learning may turn out as slow. By means of particular strategies, e.g., momentum terms, convergence and learning can be accelerated.
- Danger of disregarding basic knowledge in well-known formulas.
- Danger of passing over the limits gained by the training process (risk of undue extrapolation).
- Danger of overtraining: An overtraining effect arises if by too extensive training the network happens to learn details of the input data which are not essential, e.g., measurement or process noise. Overtraining can be detected by applying test data to the artificial neural network during or after the training phase. Training is no more useful and should be interrupted before the sum of squares errors referring to the test data might increase as a consequence of more training. More neurons support the overtraining effect.
- Danger of overvalueing process simulation.

13 Conclusion

Algorithms characterizing the basic types of artificial neural networks have been presented, pointing out the advantages and computational facilites for simulation purposes.

The algorithms are the basis of several software tools and strategies combining the real-world process and the artificial neural network.

The particular advantages and the individual characteristics concerning the application of artificial neural networks are overviewed in order to estimate or predict the opportunities of exceptionally applying artificial neural networks in various fields.

References

- Anlauf, J., Gröber, A., und Knapp, B., 1995, Neurocomputer SYNAPSE1 für vernetzte Lösungen mit künstlichen neuronalen Netzen, e&i (Elektrotechnik und Informationstechnik) 112, H. 7/8, S. 376-384*
- Budil, M., und Krenn, G., 1995, Neurohardware — Ein optischer Computer und der Neurocomputer SYNAPSE-1, e&i (Elektrotechnik und Informationstechnik) 112, H. 7/8, S. 370-376*
- ECANSE, Environment for Computer Aided Neural Software Engineering (Siemens AG Österreich, Programm- und Systementwicklung, Wien)*
- Grünberger, T., 1995, Optimierung neuronaler Regler mit Hilfe genetischer Algorithmen, e&i (Elektrotechnik und Informationstechnik) 112, H. 7/8, S. 338-344*
- Hafner, S., Geiger, H., Kreßel, U., 1992, Anwendungsstand Künstlicher Neuronaler Netze in der Automatisierungstechnik, Teil 1, Einführung, Automatisierungstechnische Praxis 34, S. 592-599*
- Hebb, D., 1949, The Organization of Behavior (Wiley, New York)*
- Hecht-Nielsen, R., 1989, Neurocomputing (Addison-Wesley, Reading)*
- Hunt, K.J., Sbarbaro, D., Zbikowski, R., and Gawthrop, P.J., 1992, Neural Networks for Control Systems — A Survey, Automatica 28, pp. 1083 - 1112*
- Kohonen, T., 1989, Self-Organization and Associative Memory (Springer-Verlag, Berlin)*
- Lippmann, R.P., 1987, An Introduction to Computing with Neural Nets, IEEE ASSP Mag., pp. 4-22*
- Miesbach, S., und Schürmann, B., 1991, Wenn Roboter arbeiten lernen: Ideen und Methoden der Neuroinformatik zur Regelung und Steuerung, Informationstechnik it 33, S. 300-309*
- Miller, W.T., Sutton, R.S., and Werbos, P.J., 1990, Neural Networks for Control (MIT Press, Cambridge Mass.)*
- Mistry, S.I., and Nair, S.S., 1994, Identification and Control Experiments Using Neural Designs, IEEE Control Systems 14, pp. 48-57*
- Morari, M., and Zafiriou, E., 1989, Robust Process Control (Prentice Hall, Englewood Cliffs)*
- Narendra, K.S., and Parthasarathy, K., 1990, Identification and Control of Dynamical Systems Using Neural Networks, IEEE Trans. on Neural Networks 1, pp. 4-27*
- Preuß, H.-P., und Tresp, V., 1994, Neuro-Fuzzy, Automatisierungstechnische Praxis 36, S. 10-24*
- Ritter, H., Schulten, K., und Martinez, T., 1989, Eine Einführung in die Neuroinformatik (Springer, Berlin)*
- Schöneburg, E., 1992, Anwendungsstand Künstlicher Neuronaler Netze in der Automatisierungstechnik, Teil 3: Diagnose mit Neuronalen Netzen, Automatisierungstechnische Praxis 35, S. 161-166*
- Thurner, E., 1995, Environment for Computer Aided Neural Software Engineering, e&i (Elektrotechnik und Informationstechnik) 112, H. 7/8, S. 385-392*
- Weinmann, A., 1991, Uncertain Models and Robust Control (Springer, New York and Vienna)*
- Widrow, B., and Hoff, M.E., 1960, Adaptive switching circuits, IRE Wescon Convention Record, pp. 96-104*
- Zell, A., 1994, Simulation Neuronaler Netze (Addison-Wesley Deutschland)*

O. Univ.Prof. Dipl.-Ing. Dr.techn. Alexander Weinmann
Head of the Institute on electrical control engineering

Vienna University of Technology

Gußhausstraße 27, A-1040 Vienna, Tel. 58801*3905, Fax 5058907
email WEINMANN@IERT.TUWIEN.AC.AT

4. IFAC-Symposium on „Low Cost Automation - LCA '95

P. Kopacek

13.-15. September 1995
Buenos Aires, Argentinien

Nach Valencia, Mailand und Wien fand das vierte für die Automatisierung in Klein- und Mittelbetrieben bedeutende Symposium in Buenos Aires statt. Ein Grundgedanke dabei war, der argentinischen IFAC Mitgliedsorganisation „Asociacion Argentina de Control Automatico - AADECA“ eine Möglichkeit zu geben die Automatisierungstechnik in diesem sowie in anderen südamerikanischen Ländern zu popularisieren. Diese ist in diesen Ländern nur in bestimmten Industriezweigen (Stahl, Nahrungsmittel, Landwirtschaft usw.) bis zu einem gewissen Grad eingeführt. Die Mittel dazu entsprechen nur teilweise dem derzeitigen Standard.

Naturgemäß erstreckte sich das Spektrum der Beiträge über das ganze Gebiet der Automatisierungstechnik. In drei Surveypapers wurden zunächst Fuzzy-Regelung, „low cost“ CIM sowie „low cost“ Automatisierung in einem Stahlwerk behandelt. Fuzzy-Methoden - in der Automatisierungstechnik bereits ein unentbehrliches Werkzeug - stellen sich immer mehr als geeignet für „low cost“ Lösungen heraus. CIM Lösungen, basierend auf PC's und LAN's (Lokale Netze) sind bereits eine Selbstverständlichkeit. Neu dabei ist - wie im zweiten Übersichtsvortrag erläutert - die Nachrüstung von Werkzeugmaschinen mit numerischen Steuerungen. Diese so entstehenden CNC-Maschinen können in ein CIM-Konzept eingebunden werden. PC's als Benutzerinterface in der Prozeßautomatisierung setzen sich als „low cost“ Lösungen immer mehr durch. Das Problem des Softwarepreises - bei den Herstellern der Geräte üblicherweise sehr hoch - wird durch Eigenprogrammierung von Modulen für einen ganz speziellen Anwendungsfall gelöst. Im dritten Übersichtsvortrag wurden solche Lösungen für ein Stahlwerk präsentiert.

Von den 66 Vorträgen können mehr als 70% der Prozeßautomatisierung - Automatisierung kontinuierlicher Prozesse - zugeordnet werden. Dies ist nicht verwunderlich, da auf den ersten beiden Symposien 1986 und 1989 fast ausschließlich Papers dieses Gebietes auf dem Programm standen. Das Spektrum erstreckte sich dabei von Geräten wie Sensoren, Aktoren und Reglern über theoretische Beiträge - neuronale Netze und Expertensysteme - bis zur Simulation und Anwendungen. Nur zwei der 14 Sessions waren der Fertigungsautomatisierung gewidmet. Hier dominierten überraschenderweise Arbeiten über Roboter.

Die Teilnehmerzahl war mit 102 etwa gleich wie 1992 in Wien, wobei aber diesmal das Veranstaltungsland mit 48 fast 50% der Teilnehmer stellte. Es bleibt zu hoffen, daß das nächste LCA - 1998 in Shenyang (PRC) - ebenso gut organisiert ist wie dieses.

Human-Oriented Design of Advanced Robotics Systems DARS '95

P. Kopacek, W. Schachner

IFAC - Workshop Vienna, Austria
September 19 - 20, 1995

The first IFAC-workshop on this topic took place at the Technical University in the heart of Vienna the capital of Austria. It was organized by the Institute of Handling Devices and Robotics (TU Vienna) and sponsored by IFAC and the IFAC-TC HAC.

Novel approaches for advanced robotics systems will be applied in the industrial production and in many other areas in the near future. For the employees to be able to work with these systems, new ergonomical, social and cultural aspects must be taken under consideration. At this workshop (DARS'95), new aspects of design were overviewed and discussed.

The objectives of the Workshop were to provide a forum for the exchange of the latest knowledge and new ideas of science and technology in the field of new ergonomical, social and cultural aspects of design of robotics systems. The main topics of the scientific programme were

-) the system design, specifically of the man-machine interface, for autonomous, semi-autonomous and tele-operated mode and for tele-existence;
-) the organizational and social aspects with respect to the environment in which the system is embedded;
-) the cultural aspects due to different living and working traditions and conditions of the people involved;
-) the economic aspects.

The following numbers of papers were given in the below areas:

- ⇒ 4 Survey
- ⇒ 4 Modelling
- ⇒ 4 Man-Machine Interface Design
- ⇒ 3 Various Aspects
- ⇒ 4 Manufacturing
- ⇒ 6 Assisted Vehicle Driving
- ⇒ 13 Planing Control and Sensors/Recognition

The opening address was given by Prof. H.B. Matthias, vice rector of the Technical University of Vienna.

The workshop included four Survey Papers and 34 regular papers. The 46 participants came from 13 different countries. The next workshop on DARS will take place 1998.

Supplementary Ways for Improving International Stability (SWIIS '95)

Peter Kopacek, Gernot Kronreif

IFAC Conference
Vienna, Austria Sept. 29th - Oct. 1st, 1995.

The IFAC event on „Supplementary Ways for Improving International Stability“ - SWIIS'95 was held in Vienna, Austria from September 29th to October 1st, 1995. This fifth event in the SWIIS series was organized by the Institute for Handling Devices and Robotics of the Technical University of Vienna. The conference was sponsored by IFAC and by the IFAC-TC on Supplementary Ways for Improving International Stability (SWIIS). Several other IFAC-TCs as well as the Technical University of Vienna (TU Wien) and the Creditanstalt Bankverein Austria (CA) supported SWIIS'95 as co-sponsors.

International stability refers to conditions in which nations, in an interdependent way, interact with one another in ways, which permit gradual changes with time in a mutually acceptable scale and direction. This development under stable conditions is considered with respect to social, political, ecological, national and international, regional and global aspects.

The conference continued the tradition set in the earlier four SWIIS meetings. The goal was the beneficial application of systems engineering methods onto description of conditions, in which nations or groups interact with one another. Scientists from other fields as political science, economics, social science, and international studies should have a platform to present and discuss their ideas. Perhaps, this SWIIS event differed from earlier SWIIS meetings in the efforts to attract a younger generation to the work in the framework of this TC.

The conference was attended by 38 participants from 14 different countries (e.g. South Africa, USA, Germany, UK). In the technical program, 3 Survey Papers and 22 regular papers - grouped into 7 sessions - were presented to the very interested audience. Papers were given in the following areas:

- ⇒ Methodological analysis
- ⇒ Investigation of development: stability, sustainable development
- ⇒ Modelling of stability
- ⇒ Application of control principles to international stability
- ⇒ International policy co-operation
- ⇒ Cultural and educational aspects in international stability
- ⇒ The new world order

The many positive comments made by the participants of this conference may confirm its success in meeting its objectives.

Arbeitskreis „Informatik und Automatisierung“ der Österreichischen Computergesellschaft (OCG)

R. Genser

Der technische Fortschritt bewirkt ein Zusammenwachsen der Bereiche Informatik, Telekommunikation und Leittechnik beziehungsweise Automatisierungstechnik. Vor allem die Franzosen bringen dies durch den neuen Begriff Telematik zum Ausdruck, in Japan spricht man vom Zeitalter der 3C (Computers, Communications and Control). Dieser Vorgang hat aber auch Auswirkungen auf Organisationsstrukturen, sei es im Bereich der Forschung und Entwicklung oder in der Betriebs- und Nutzungsphase selbst.

Die fünf Schwesterorganisationen IFAC, IFIP, IFORS, IMEKO und IMACS hatten schon während ihrer Gründungsphase die Notwendigkeit einer Koordinierung und des Informationstransfers untereinander als wesentlich erkannt. Der Kommission der Europäischen Gemeinschaften wurde aber auch bewußt, daß neben einer Zusammenarbeit der Forschung, Industrie und Großnutzern, wie Militärwesen, Eisenbahnen, Fernmeldeunternehmen usw., auch das Wissen breiterer Nutzerkreise einbezogen werden muß, um eine Verbesserung der Lebensqualität zu erreichen und technische Lösungsmöglichkeiten effizient zu nutzen.

Die österreichischen Gegebenheiten - überwiegend Klein- und Mittelbetriebe - und die Anforderungen der heutigen Entwicklung (EU) bewogen daher den Vorstand der Österreichischen Computergesellschaft den Arbeitskreis „Informatik und Automatisierung“ am 7. Juni 1995 zu gründen. Insbesonders soll das Zusammenwachsen der Bereiche Informatik, Telekommunikation und Leittechnik durch die technische Entwicklung, sei es bei der Produktion, beim Verkehr oder im Haushalt durch diesen Arbeitskreis im Rahmen der OCG berücksichtigt werden.

Als Ziele und Aufgaben wurden festgelegt:

- Aufbereitung von Neuland im technischen, organisatorischen und sozialen Bereich
- Heranführen der Lösungsmöglichkeiten an österreichische Gegebenheiten
- Erarbeiten von Gesichtspunkten für zukunftssichere Lösungen.

Dies soll in breiter nationaler und internationaler Kooperation mit Normungsgremien, wissenschaftlichen und wirtschaftlichen Kreisen unter Einbindung von Anwendergruppen durch Arbeitstreffen, Tagungen und andere Informations- und Diskussionsveranstaltungen gefördert werden.

Die Leitung dieses Arbeitskreises durch Prof. Dr. Peter Kopacek, Vorstand des Institutes für Handhabungsgeräte und Robotertechnik an der Technischen Universität Wien, und der beiden Stellvertreter Dr. Martin Zauner (Bereich Informatik) und Dr. Robert Genser (Bereich Automatisierung) sollen die Brücke zwischen Informatik, Telekommunikation und Automatisierung, aber auch zwischen der International Federation of Automatic Control (IFAC) und der International Federation for Information Processing (IFIP) auf nationaler Ebene gewährleisten. Es ist die Gründung einer IFIP Working Group im TC5 geplant.

Es bleibt zu hoffen, daß dieser Arbeitskreis für Informatik und Automatisierung vor allem von bisher weniger erfaßten Gruppen genutzt wird. Die nächsten Sitzungen sind für den 13. März und 22. Mai 1996 bei der OCG, Wollzeile 1, 1010 Wien, vereinbart.

Identifikation strukturierter Modellunsicherheiten aus verrauschten Meßreihen

Rudolf Hornischer

Begutachter: O.Univ.Prof. Dr. A. Weinmann und Doz. Dr. R. Noisser

Die vorliegende Arbeit beschreibt ein rekursives Verfahren zur Identifikation dynamischer Systeme aus verrauschten Meßdaten. Ausgangspunkt ist ein aus der Literatur bekanntes, auf funktionalanalytischen Überlegungen beruhendes Verfahren, welches in seiner ursprünglichen Form jedoch nur bei rauschfreien Messungen angewendet werden kann.

Im Gegensatz zu den auf der Methode der kleinsten Quadrate beruhenden Algorithmen ist bei diesem Verfahren eine genaue Kenntnis der Systemordnung nicht erforderlich. An ihre Stelle tritt ein vom Anwender vorgegebenes, nominelles parametrisches Modell, das durch eine ebenfalls vom Anwender zu wählende unstrukturierte multiplikative Modellunsicherheit ergänzt wird. Diese multiplikative Unsicherheit beschreibt den maximalen Modellfehler im Frequenzbereich und kann somit als obere Schranke des Modellfehlers betrachtet werden. Der Ansatz erscheint sinnvoll, da in der Praxis oft der Fall auftritt, daß Teilbereiche eines Systems gut bekannt und somit a priori durch ein parametrisches Modell zu beschreiben sind während man von anderen Bereichen der zu regelnden Anlage nur eine vage Vorstellung, zB. über deren Frequenzgang, besitzt. Da der nichtparametrische Teil des Modells für den Modellfehler nur eine obere Schranke darstellt, ist der zugehörige parametrische Modellteil nicht mehr eindeutig bestimmt. Vielmehr bilden alle parametrischen Modelle, die zu einem Modellfehler unterhalb dieser Schranke führen, eine Modellmenge, die im Parameterraum im allgemeinen eine unregelmäßige Begrenzung aufweist. Das vorgestellte Identifikationsverfahren liefert für diese Modellmenge eine Abschätzung in Form einer elliptischen oder hyperbolischen Hüllmenge.

Im ersten Teil der Arbeit wird der bekannte Algorithmus für den rauschfreien Fall, da er für das weitere Verständnis notwendig ist, nochmals hergeleitet. Daran anschließend folgt eine systematische Untersuchung über günstige Anregungssignale, die eine wertvolle Hilfestellung für den praktischen Einsatz des Verfahrens liefert. So kann z.B. für sinusförmige Anregung ein Frequenzbereich angegeben werden, der mit Sicherheit zu elliptischen Lösungsmengen führt. Eine Aussage darüber, wie konservativ die Schätzung ausfallen wird, ist ebenfalls möglich.

Im zweiten Teil der Arbeit wird der Algorithmus für die Anwendung auf verrauschte Meßreihen erweitert. In diesem Fall wird das tatsächlich auftretende Meßrauschen auf gefiltertes weißes Rauschen zurückgeführt, Vorwissen über die Struktur des Störfilters und die Varianz des Rauschens wird vorausgesetzt. Betrachtungen über den Einfluß des

Signal-Geräuschverhältnisses auf die Ausdehnung der aus der Identifikation resultierenden Übermenge runden das Kapitel ab.

Schließlich wird an Hand von Beispielen die Anwendung des Verfahrens gezeigt. Die Auswirkung verschiedener Störmodelle sowie unterschiedlicher multiplikativer Unsicherheiten wird durch Simulationen verdeutlicht. Abschließend bleibt zu sagen, daß der Algorithmus gutes Konvergenzverhalten besitzt und vom Rechenaufwand mit der Methode der kleinsten Quadrate vergleichbar ist. Sensibel reagiert der Algorithmus hingegen auf die nichtparametrische Unsicherheit. Eine unglückliche Wahl dieser Fehlerschranke führt schnell zu konservativen Ergebnissen.

Schwache L_2 -Stabilität und Fuzzy Control

Oliver König

Begutachter: Doz. Dr. R. Noisser und o.Univ.Prof. Dr. A. Weinmann

Die Dissertation präsentiert ein neues Analyse- und Synthesewerkzeug für eine Klasse nichtlinearer Regelungssysteme. Es handelt sich dabei um die sogenannte schwache L_2 -Stabilität. Die Methode gestattet Stabilitätsuntersuchungen an Systemen, die sich in der Form des nichtlinearen Standardregelkreises, d. h. als rückgekoppelte Serienschaltung eines nichtlinearen statischen Systems und eines linearen dynamischen Systems darstellen lassen.

Der Einsatz der Methode zu Analysezwecken liefert neben der hinreichenden Stabilitätsaussage für ein gegebenes System zusätzlich eine Abschätzung für die Amplitude einer potentiellen Stationärabweichung. Bei Anwendung der Methode als Synthesehilfsmittel läßt sich für eine geforderte maximale Stationärabweichung und ein gegebenes lineares System ein zulässiger (stabilitätsgarantierender) Sektor angeben, in dem sich die Kennfläche des nichtlinearen Systems befinden muß. Diese Entwurfsmethodik führt im Fall von nichtlinearen Systemen mit einem Eingang und einem Ausgang (nichtlineare Kennlinie) auf ein graphisches Werkzeug, das sogenannte schwache Kreiskriterium, welches auf der Ortskurve des linearen Systems basiert.

In weiterer Folge wird die Theorie zur Stabilitätsanalyse von zeit- und amplitudendiskreten Fuzzy-Regelkreisen eingesetzt, wobei die Regelstrecke linear angenommen wird. Ein Vergleich der Ergebnisse des neuen Stabilitätsanalyseverfahrens mit Zeitbereichssimulationen zeigt, daß die Methode sich vorwiegend für schwach ausgeprägte Nichtlinearitäten eignet, da die Konservativität der Stationärabschätzungen mit zunehmender Nichtlinearität des Kennfeldes stark zunimmt. Andererseits darf man bei Einsatz des Verfahrens zu Synthesezwecken davon ausgehen, daß die vorgegebene Stationärschranke, aufgrund der Konservativität des Verfahrens, auch bei unzureichender Modellkenntnis eingehalten wird.

Produkthaftung und Risikominderung

Eine allgemein verständliche Einführung

Egon SATTLER

Carl Hansen Verlag, München und Wien, 1995
338 Seiten. DM 86.-

Die Sicherung der Qualität von Serien- und Einzelprodukten und die damit verbundene Verantwortung im Sinne der Produkthaftung hat in den letzten Jahren überproportional an Bedeutung zugenommen. Die Problematik erstreckt sich auf Industrie, Gewerbe und Dienstleistungsbetriebe. In diesen Bereichen hatte man sich mit dem mit jeder Lieferung verbundenen Risiko gründlich auseinanderzusetzen und muß dies in Schulungen auch weiter intensiv besorgen.

Das vorliegende Buch vermittelt sehr ausgewogen und leicht faßlich die Grundlagen aus dem Bereich des Rechtswesens, des Allgemeinen Bürgerlichen Gesetzbuches, der allgemeinen Geschäftsbedingungen, des Handelsgesetzbuches und der Schutzrechte.

Verfahren der eigenen Risikobeurteilung und der Risikominderung, im internen Bereich und nach außen, sind wichtige Maßnahmen.

Ein Kapitel über typische Fragen und Antworten rundet die Materie ab.

Man ist überrascht, wie abwechslungsreich die trocken anmutenden Inhalte präsentiert werden; sicher Ausdruck der Erfahrung des Autors aus zahlreichen selbst geleiteten Seminaren.

A. Weinmann

Fuzzy-Regelungen

R. Strietzel

R. Oldenbourg Verlag GmbH, München, 1996
192 Seiten, 102 Bilder, 36 Tabellen

Das vorliegende Buch zur Anwendung von Methoden der Fuzzy Theorie auf die Gebiete der Steuerungen und Regelungen stellt sich das Ziel dem Leser sowohl eine Einführung in die Grundlagen der Fuzzy Verarbeitungsmechanismen als auch in die regelungstechnische Anwendung der Methodik zu geben. Ausgehend von den Begriffen der Fuzzy-Mengen, der Fuzzy-Operationen und den Interferenztechniken werden insbesondere die Transformationen zwischen linguistischen/qualitativen Größen und den quantitativen Größen in Form der Fuzzyfizierung und Defuzzyfizierung betrachtet. Diese beiden Transformationen sind für die Lösung regelungstechnischer Aufgaben besonders wichtig, da sie die Schnittstellen zum realen Prozeß darstellen. Den Schwerpunkt des Buches bilden Betrachtungen zum Problem der Struktur für den Einsatz von Fuzzy-Reglern und des Übertragungsverhaltens selbst. Vorgestellt werden Strukturen für einfache und vermaschte Regelungsaufgaben, einschließlich adaptiver Konzepte. Beim Übertragungsverhalten wird insbesondere das nichtlineare statische Verhalten dargelegt und untersucht. Vermisst werden in diesem Abschnitt Aussagen zum Einfluß der Testperiode, der Wahl von Gütekriterien und Aussagen zur Stabilitätsanalyse. Betrachtungen zur Gestaltung von Neuro-Fuzzy-Systemen schließen das Buch ab.

Insgesamt kann eingeschätzt werden, daß das vorliegende Buch eine gute Einführung und einen ersten Überblick über Arbeiten auf dem Gebiet der Fuzzy-Regelungen vermittelt. Das Buch kann den Studierenden und in der Praxis tätigen Ingenieuren zur Lösung von Automatisierungsaufgaben auf der Basis von qualitativem Wissen oder Erfahrungen empfohlen werden. Dem Leser sollte ebenfalls bewußt sein, daß zu Fragen des optimalen Entwurfs, der Stabilitätsprüfung der Entwurfswerkzeuge und der Hardwareimplementierung weiterführende Literatur erforderlich ist.

J. Wernstedt

„Informatik für Maschinenbauer“

P. Kopacek, R. Probst und M. Zauner

307 Seiten, 126 Abbildungen
Springer Verlag Wien, 1995

Die Tätigkeit des Absolventen im Bereich der Ingenieurwissenschaften erfordert fundierte Kenntnisse, auch auf dem Gebiet der Informatik. Hard- und Software sind sowohl für den Entwurf erforderlich, als auch integrierter Bestandteil moderner Geräte und Anlagen. Ebenfalls erfordert die rechnergestützte Produktion leistungsfähige Datenverarbeitungssysteme.

Das vorliegende Buch gibt eine hervorragende Einführung in die Grundlagen der Hardwarearchitekturen, der Betriebssysteme MS-DOS und UNIX sowie des Softwareengineering. Der Leser findet detaillierte Einführungen in die derzeit am häufigsten verwendeten Programmiersprachen. Dabei wird stets auf die Anwendung moderner Programmiermethoden, wie z.B. die objektierte Programmierung eingegangen. Die ausführliche und gelungene Vorstellung der Programmiersprache C wird für zukünftige Entwicklungen als wertvoll angesehen. Die gleiche Wertung kann für die übersichtliche, kompakte Darstellung von Datenbanksystemkonzepten, lokalen Netzen und Fragen der Mensch-Maschine-Kommunikation ausgesagt werden.

Die langjährigen praktischen Erfahrungen von Hard- und Softwarekomponenten im Maschinenbau werden in dem Abschnitt „Rechner im Maschinenbau“ deutlich. Es wird der Versuch unternommen, dem Leser die sich in rasanter Entwicklung befindlichen Gebiete wie Computergrafik, CIM-Konzepte, Robotik, wissensbasierte Systeme und künstliche neuronale Netze in kurzen Einführungen näher zu bringen und auf weiterführende Literatur hinzuweisen.

Das vorliegende Buch erfüllt voll seinen Anspruch als Lehrbuch für Studierende technischer Studienrichtung, insbesondere für Studiengänge des Maschinenbaus. Es zeichnet sich durch eine gelungene Kombination von theoretischen Grundlagen, Beispielen und Darstellung von Praxisprojekten aus.

Das insgesamt gelungene Konzept ist darauf zurückzuführen, daß die Autoren als Maschinenbauer die Gebiete der Informatik gekonnt ausgewählt haben, die für die zukünftige Entwicklung des Maschinenbaus unverzichtbar sind.

Th. Rauschenbach

IARP Workshop „Medical Robots“

1-3 Oktober 1996, Wien

Die Anwendung von Robotersystemen in der Medizin erweist sich vermehrt als ein Forschungs- und Entwicklungsbereich mit enorm steigender Wachstumsrate. Der Einzug des „Doktor Roboter“ eröffnet dabei ein immens großes Potential zur Reduktion der Kosten des Gesundheitswesens.

Die Entwicklungstendenzen und ersten Anwendungen können in einige verschiedene Bereiche gegliedert werden, wie z.Bsp. Rehabilitation, Transportaufgaben, Hilfsdienste während der Operation oder auch „Mikroroboter“ zur Anwendung „im Inneren des Körpers“. Weitere Entwicklungen liegen im Gebiet der roboter-unterstützten Diagnose oder der roboter-unterstützten Radio- und Lasertherapie.

Die Identifikation neuer Anwendungsbereiche sowie die Verbesserung der Roboter-anwendungen setzt eine enge Zusammenarbeit zwischen Ingenieurtechnik, Wissenschaft und Medizin voraus. Zweck dieses IARP (International Advanced Robotics Programme) Workshops „Medical Robots“ soll es sein, Experten aus den obgenannten Fachgebieten sowie Vertretern der Industrie ein Forum zur Definition des „state of the art“ und zur Diskussion neuer Anwendungsmöglichkeiten von Robotertechnik in der Medizin zu ermöglichen.

Es sind Vorträge zu folgenden Themenbereichen eingeladen:

- ⌚ Operationsplanung und -simulation
- ⌚ Sicherheit von Roboteranwendungen in Medizin und Chirurgie
- ⌚ Bildverarbeitung in der Therapie
- ⌚ Mensch-Maschine Schnittstelle für Roboter in der Medizin
- ⌚ Tele-Chirurgie und Tele-Medizin
- ⌚ Robotersysteme für medizinische Anwendungen
- ⌚ Mikro-Roboter für Medizin und Chirurgie
- ⌚ „Virtuelle Realität“ in der Medizin
- ⌚ Klinik-Anwendungen für „Medical Robots“ und praktische Erfahrung
- ⌚ Unterstützung für Behinderte
- ⌚ Robotik in der Altenpflege
- ⌚ Roboter in der Rehabilitation
- ⌚ Robotertechnik für Transportaufgaben

Weitere Informationen:

Institut für Handhabungsgeräte und Robotertechnik
Technische Universität Wien
„Medical Robots“
Floragasse 7A
A-1040 Wien
Tel: (+43-1-) 504 18 35, FAX: (+43-1-) 504 18 35/9
E-mail: medrob@ihrt1.ihrt.tuwien.ac.at

Ein Institut stellt sich vor:

Institut für Handhabungsgeräte und Robotertechnik der Fakultät Maschinenbau an der TU Wien

P. Kopacek

Das zur Fakultät für Maschinenbau gehörende Institut für *Handhabungsgeräte und Robotertechnik* wurde am 4. Jänner 1990 gegründet und nahm am 1. März 1990 seine Tätigkeit auf. Es ist derzeit in Büroräumen in der Floragasse 7A und in Laborräumen am Möllwaldplatz 5 (PC-Labor) bzw. Wiedner Hauptstraße 8 - 10 (Roboter-Labor) untergebracht. Der Personalbestand setzt sich derzeit aus 1 ordentlichen Universitätsprofessor, 3 Universitätsassistenten, 2 allgemeinen Bediensteten und einem Gastwissenschaftler zusammen. Im Sommersemester 1996 sind weiters 4 Gastprofessoren und 7 Lektoren im Rahmen von Lehrveranstaltungen am Institut tätig.

Das Institut beschäftigt sich sowohl theoretisch als auch praktisch mit dem Fachgebiet der Roboter- und Handhabungssysteme im weitesten Sinne. Theoretische Untersuchungen beziehen sich auf die Kinematik und Kinetik von Industrierobotern sowie die Anwendung *fortgeschrittenener* Regelalgorithmen. Weitere Arbeitsschwerpunkte sind die Konstruktion von Leichtrobotern, Verfahren zur Bilderkennung sowie Anwendung der Methoden der künstlichen Intelligenz, insbesonders neuronale Netze.

Breiten Raum nimmt der Einsatz von Industrierobotern in der österreichischen Industrie - und hier vor allem in Klein- und Mittelbetrieben, insbesonders für Montage- und Demontageaufgaben, ein. Da diese Geräte üblicherweise in Verbindung mit CIM-Konzepten eingesetzt werden, wird auch dieses Gebiet bearbeitet. Zur praktischen Erprobung stehen dazu im Labor 4 Industrieroboter, verbunden durch ein Transportsystem, zur Verfügung. Weiters finden 3 Lehr- und Übungsroboter in verschiedenen Lehrveranstaltungen Verwendung.

Lehrveranstaltungen

Das Institut bietet Lehrveranstaltungen zu aktuellen Gebieten der Automatisierungstechnik an und ist zusätzlich für die Ausbildung der Maschinenbaustudenten in Informatik verantwortlich.

Einen Überblick über die angebotenen LVAs bietet der Institutsfolder. Dieser Folder wurde zum ersten Male im Jahre 1991 erstellt und ist seitdem zu Beginn jedes Semesters in aktualisierter Form verfügbar. Er enthält neben kurzen Beschreibungen der LVAs auch Prüfungs- und Vorbesprechungstermine, Telefonnummern, email-Adresse und Lageplan. Im Sommersemester 1996 werden bereits 29 verschiedene Lehrveranstaltungen abgehalten.

Innerhalb von Pflichtlehrveranstaltungen betreut das Institut alle Studienrichtungen der Fakultät Maschinenbau. Im Rahmen der Wahllehrveranstaltungen, Diplomarbeiten und Dissertationen betreut das Institut aufgrund der Interdisziplinarität allerdings auch immer mehr Studenten der Informatik und Elektrotechnik.

Ebenfalls gerne von den Studenten besucht sind die jährlich durchgeführten Exkursionen. So konnte man auf diese Weise in den letzten Jahren Firmen in Österreich, Deutschland, Schweiz, Schweden, Italien und Slowenien kennenlernen.

Weiterführende Ausbildung

In Zusammenarbeit mit der Abteilung "System- und Automatisierungstechnik" an der Landesakademie für NÖ werden zahlreiche postgraduale Ausbildungsprogramme durchgeführt (siehe Tätigkeitsbericht), wovon das "Engineering Management" Executive Education Program, in Kooperation mit der Oakland University in Rochester/Michigan, USA, besonders hervorzuheben ist. Dieses zielt darauf ab eine Persönlichkeit mit interdisziplinärer Ausbildung in den Fachbereichen Wirtschaft, Recht und Technik als flexiblen, kompetenten Engineering Manager für die Unternehmen der Zukunft auszubilden.

In den USA hat dieses neue Modell der Unternehmungsführung bereits zu starker Nachfrage nach entsprechenden Ausbildungsprogrammen geführt. Jetzt gibt es auch in Europa die Ausbildung zum Master of Engineering Management. Das Engineering Management Executive Education Program besteht aus einem Hauptprogramm und zwei Vorbereitungskursen. Das Hauptprogramm ist für alle Teilnehmer verpflichtend. Je nach Ausbildungsstand der Teilnehmer müssen zusätzlich bestimmte Module der Vorbereitungskurse absolviert werden.

Die Absolventen erhalten ein Zeugnis der Oakland University und der Technischen Universität Wien, sowie ein Diplom der Landesakademie für NÖ. Unter bestimmten Umständen können Universitätsabsolventen auch den Abschluß „Master of Science in Engineering Management“ der Oakland University erwerben.

Forschungstätigkeit

Das Institut hat es sich zur Aufgabe gemacht auf theoretischen Gebieten der Robotertechnik nur insofern tätig zu werden, als dies für die industrielle Anwendung zukunftsträchtig erscheint. Es sollen einerseits Forschungsergebnisse für die Industrie nutzbar gemacht und diese durch eigene Arbeiten ergänzt werden. Hauptstoßrichtung sind dabei Klein- und Mittelbetriebe. Daraus ergeben sich folgende Forschungsschwerpunkte:

- Kinematik und Kinetik von Industrierobotern
- *Fortgeschrittene* Regelalgorithmen (Fuzzy und neuronale Netze)
- Optimale Roboterbahnen
- Koordination mehrerer Roboter
- Sensoren (Mikrosysteme)
- Roboteranwendungen
- Roboter in CIM Konzepten
- Montage und Demontage mit Industrierobotern
- Serviceroboter, insbesonders für medizinische Anwendungen
- Technische Anwendungen der Automatisierungstechnik
- „low cost“ Bilderkennung
- Montagegerechte und demontagegerechte Konstruktion
- Methoden der künstlichen Intelligenz in der Robotik
- *Low cost* Automatisierung

Zusätzlich werden auf systemtechnischem Gebiet Forschungen in folgenden interdisziplinären Richtungen betrieben:

- Systemtechnische Methoden zur Konfliktforschung
- Soziale und kulturelle Aspekte der Automatisierungstechnik

Internationale Aktivitäten

Das Institut organisierte bereits mehrere internationale Konferenzen und Veranstaltungen. Für diese wurden teilweise auch WWW-Seiten eingerichtet, die über den Infoserver der TU (<http://www.info.tuwien.ac.at>) - siehe Internationale Aktivitäten - abrufbar sind. Nachfolgende Liste enthält die bedeutendsten Ereignisse der letzten 2 Jahre:

- Internationale Summerschool "Rechnergestützte Produktion und Robotertechnik", TU-Wien und Wissenschaftliche Landesakademie für NÖ, 2. - 20.9.1991; 23.8. - 12.9.1992; 23.8. - 11.9.1993; 10. - 30.7.1994; 9. - 22.7.1995
- 2. IFAC-Workshop on "Intelligent Manufacturing Systems - IMS'94", Wien, 13. - 15.6.1994
- 10. Österreichischer Automatisierungstag, TU-Wien, 20.10.1994
- 4. Intern. Workshop "Robotics in Alpe-Adria Region - RAA'95", Pörtschach, 6. - 8.7.1995
- IFAC-Workshop on "Human-Oriented Design of Advanced Robotics Systems - DARS'95", Wien, 19. - 20.9.1995
- 11. Österreichischer Automatisierungstag, Messe Congress Centrum, Wien, 28.9.1995
- IFAC Conference on "Supplementary Ways for Increasing International Stability - SWIIS'95", Wien, 29.9. - 1.10.1995

Kooperationen

Zur Verbesserung der Forschungsmöglichkeiten bestehen folgende nationale und internationale Kooperationen: Fraunhofer Institut Berlin; TU Ilmenau (Deutschland); TU Dresden (Deutschland); Instituto Politecnico Nacional, Mexico City (Mexiko); Universität Linz; Landesakademie für Niederösterreich, Krems; Texas A & M University, College Station (USA); Universidad de Valencia (Spanien); Universidad de Canarias, Las Palmas (Gran Canaria); Oakland University, Rochester (USA).

Weiters ist das Institut Mitglied des Forschungsnetzes ERNET (European Robotics Network), in dem 26 europäische Institute intensiv zusammenarbeiten.

Publikationen

Die Veröffentlichungen des Instituts seit seiner Gründung Anfang 1990 umfassen Fachvorträge (derzeit über 85) sowie Beiträge zu Konferenzen und Fachzeitschriften (derzeit über 110). Genaue Angaben sind dem jährlich erscheinenden Institutsbericht zu entnehmen. Zusätzlich konnten bis jetzt 6 Bücher publiziert und 29 mal als Herausgeber die Verantwortung übernommen werden. Das zuletzt veröffentlichte Buch (Kopacek, P.; Probst, R. und Zauner, M.: Informatik für Maschinenbauer, Springer-Verlag Wien, 1995) findet in der Pflichtvorlesung "Grundzüge und Methoden der EDV" großen Zuspruch bei den Maschinenbaustudenten.

**Ausschreibung
des
Fred Margulies-Preises
für Forschungsarbeiten auf dem Gebiet der
Automatisierungstechnik
mit besonderer Berücksichtigung der sozialen Aspekte**

Anlässlich des zehnjährigen Bestehens des IFAC-Beirates Österreich verleiht dieser jährlich zusammen mit der Gewerkschaft der Privatangestellten den Fred-Margulies-Preis. Der Preis wird für hervorragende österreichische Arbeiten auf dem Gebiet der Automatisierungstechnik mit besonderer Berücksichtigung der sozialen Aspekte verliehen. Kriterien für die Bewertung sind im wesentlichen Innovationsgehalt, Wissenschaftlichkeit, wirtschaftliche Bedeutung sowie Interdisziplinarität des Ansatzes und insbesondere soziale Relevanz. Der Preis in der Höhe von

öS 60.000,-- pro Jahr

wird je zur Hälfte von den beiden Stiftern geleistet und wird von einer Jury vergeben, welcher neben Persönlichkeiten aus Wissenschaft und Wirtschaft auch Vertreter der Gewerkschaft angehören. Der Fred Margulies-Preis wird vom Leiter der Forschungssektion des Bundesministeriums für Wissenschaft und Forschung, im Rahmen des "Österreichischen Automatisierungstages" verliehen. Dieser findet heuer am 24. Oktober 1996 an der Technischen Universität Wien statt.

Eingereicht können Studien werden, welche innerhalb der letzten 5 Jahre in Österreich abgeschlossen wurden. Diese Arbeiten sind bis spätestens 30. Juni 1996 in Form einer Studie in dreifacher Ausfertigung sowie einer Kurzfassung (max. 10 Seiten) in dreifacher Ausfertigung beim

**Generalsekretariat des IFAC-Beirates c/o Institut für Handhabungsgeräte- und Robotertechnik) der Technischen Universität, Floragasse 7a, 1040 Wien
einzureichen.**

Für weitere Anfragen steht das Sekretariat des IFAC-Beirates unter obiger Adresse zur Verfügung.

<i>Datum</i>	<i>Veranstaltung</i>	<i>Ort</i>	<i>Weitere Informationen erhältlich bei:</i>
22.4. - 28.4. 1996	1996 IEEE International Conference on Robotics and Automation	<i>Minneapolis, Minnesota, USA</i>	Prof. Norman Caplan FAX: +1 703 306 0312 Email: ncaplan@note.nsf.gov
27.5. - 30.5. 1996	2nd World Automation Congress (WAC '96)	<i>Montpellier, France</i>	Dr. Mo Jamshidi Univ. of Mexico USA FAX: +1 505 277 1439 Email: jamshid@houdini.unm.edu
29.5. - 31.5. 1996	CIMAT '96 Fifth International Conference on Computer Integrated Manufacturing and Automation Technology	<i>Grenoble, France</i>	Rene David FAX: +33 76 82 63 88 Email: david@lag.grenet.fr
10.6. - 13.6. 1996	5th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD '96)	<i>Budapest, Hungary</i>	Imre J. Rudas FAX: +36 1 133 9183 Email: raad@zeus.banki.hu
1.7. - 5.7. 1996	IFAC WORLD CONGRESS	<i>San Francisco, California, USA</i>	IFAC'96 POB 111, Mabelton GA 30059 USA Phone: +1/708/491-3641 FAX: +1/708/491-4455 Email: ifac96@nwu.edu
26.9. - 28.9. 1996	Second ECPD International Conference on Advanced Robotics, Intelligent Automation and Active Systems	<i>Vienna, Austria</i>	Branislav Borovac Phone: +381 21 55 011 FAX: +381 21 59 536 Email: borovac@uns.ns.ac.yu
1.10. - 3.10. 1996	IARP Workshop Medical Robots	<i>Vienna, Austria</i>	Institute for Handling Devices and Robotics Technical University of Vienna Floragasse 7A A-1040 Vienna, Austria Phone: +43 1 504 18 35 FAX: +43 1 504 18 35 9 Email: medrob@ihrt1.ihrt.tuwien.ac.at

Datum	Veranstaltung	Ort	Weitere Informationen erhältlich bei:
6.10. - 8.10. 1996	27th ISIR Robotics Towards 2000	<i>Milan, Italy</i>	ISIR '96 Secretariat CEU Centro Esposizioni UCIMU Spa Viale Fulvio Testi 128 I-20092 Cinisello Balsami MI Phone: +39 2 26 25 51 FAX: +39 2 26 25 52 14 349 Email: mofino@dimec.unige.it
24.10.1996	12. Österreichischer Automatisierungstag	<i>Vienna, Austria</i>	Institute for Handling Devices and Robotics Technical University of Vienna Floragasse 7A A-1040 Vienna, Austria Phone: +43 1 504 18 35 FAX: +43 1 504 18 35 9 Email: e318@ihrt1.ihrt.tuwien.ac.at
3.2. - 5.2. 1997	IFAC Workshop Manufacturing Systems: Modelling, Management and Control (MIM '97)	<i>Vienna, Austria</i>	Institute for Handling Devices and Robotics Technical University of Vienna Floragasse 7A A-1040 Vienna, Austria Phone: +43 1 504 18 35 FAX: +43 1 504 18 35 9 Email: mim@ihrt1.ihrt.tuwien.ac.at
5.2. - 7.2. 1997	IMACS/IFAC Symposium Mathematic Modelling MATHMOD '97	<i>Vienna, Austria</i>	Prof. I. Troch Technical University of Vienna Wiedner Hauptstr. 8 -10 A-1040 Vienna, Austria FAX: +43 1 586 8093 Email: itroch@email.tuwien.ac.at
23.3. - 26.3. 1997	IFAC Symposium Modelling and Control of Biomedical Systems including Biological Systems	<i>Warwick, UK</i>	Prof. K. Godfrey Dept. of Engineering University of Warwick Coventry CV4 7AL, UK FAX: +44 1203 418922
28.4. - 30.4. 1997	IFAC/IEEE Symposium Computer-Aided Control Systems Design - CACSD '97	<i>Ghent, Belgium</i>	Prof. Luc Boullart University of Ghent Campus Ardoyen Technologiepark Zwijnaarde, 9 B-9052 Zwijnaarde, Belgium FAX: +32 9 264 5839 Email: boullart@autoctrl.rug.ac.be