Forschungszentrum Karlsruhe

Technik und Umwelt

Wissenschaftliche Berichte

FZKA 6509

Proceedings
10. Workshop Fuzzy Control
des GMA-FA 5.22

Dortmund, 18. - 20. Oktober 2000

Ralf Mikut, Jens Jäkel (Hrsg.)

Forschungszentrum Karlsruhe GmbH, Karlsruhe
2000

VORWORT

Dieser Tagungsband enthält die Beiträge des 10. Workshops des Fachausschusses 5.22 Fuzzy Control der VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik (GMA), der vom 19.-20. Oktober 2000 im Haus Bommerholz Dortmund stattfindet.

Der jährliche Workshop unseres Fachausschusses bietet ein Forum zur Diskussion neuer methodischer Ansätze und industrieller Anwendungen auf dem Gebiet der Fuzzy-Logik und in angrenzenden Gebieten wie Künstlichen Neuronalen Netzen und Evolutionären Algorithmen. Besondere Schwerpunkte sind automatisierungstechnische Anwendungen, z.B. in der Verfahrenstechnik, Energietechnik, Kfz-Technik, Robotik und Medizintechnik, aber auch Lösungen in anderen Problemgebieten (z.B. Data Mining für technische und nichttechnische Anwendungen) sind von Interesse.

Die Ergebnisse werden von den ca. 50 Mitgliedern und Gästen aus Hochschulen, Forschungseinrichtungen und der Industrie präsentiert und in Klausuratmosphäre intensiv diskutiert.

Nähere Informationen zum Fachausschuss erhalten Sie unter

http://wwwserv2.iai.fzk.de/Institut/SK/Gang/gma/index.html.

Die Herausgeber bedanken sich an dieser Stelle bei allen Autoren und Rednern sowie bei Herrn Dr. Kroll (ABB Heidelberg), der maßgeblich an der Vorbereitung des Workshops beteiligt war.

Ralf Mikut und Jens Jäkel

INHALTSVERZEICHNIS

Implizite Modellierung, inkrementeller Relevanzindex und Rauigkeitsmaß: neue Strategieelemente für die datenbasierte Modellierung

H. Kiendl
Universität Dortmund
Fakultät für Elektrotechnik und Informationstechnik
Lehrstuhl für Elektrische Steuerung und Regelung
Otto-Hahn-Str. 4, D-44221 Dortmund
Tel.: +49.231.755-2760
Fax: +49.231.755-2752
e-mail: kiendl@esr.e-technik.uni-dortmund.de

Kurzfassung: Die Anwendungen in Industrie und Wissenschaft verlangen nach immer leistungsfähigeren Methoden zur datenbasierten Modellierung. Hierfür werden drei neue Strategieelemente vorgestellt. Das Konzept der impliziten Modellierung verhindert, dass in den Daten vorhandene Mehrdeutigkeiten sich störend auf den Prozess der eigentlichen Modellierung auswirken. Statt dessen wird die Möglichkeit eröffnet, erst nach Erstellung des sogenannten impliziten Modells durch entsprechende Auslegung der Defuzzifizierungsvorschrift festzulegen, ob die Mehrdeutigkeiten eher im Sinne eines Kompromisses oder einer Entscheidung für den am besten durch die Daten gestützten Ausgangsgrößenwert zu berücksichtigen sind. Der inkrementelle Relevanzindex dient bei der datenbasierten Generierung zum Hypothesentest und zur Bewertung von Fuzzy-Regeln. Insbesondere kann damit das resultierende Fuzzy-Modell nachträglich an unterschiedliche anwendungsspezifische Anforderungen angepasst und gewisse Nachteile bekannter Test- und Bewertungsverfahren können vermieden werden. Das Rauigkeitsmaß dient zur Bewertung und ggf. Glättung erhobener Daten bzw. zur Bewertung und zum Vergleich von datenbasiert generierten Modellen.

1 Einführung

Anwender verlangen nach immer leistungsfähigeren Methoden zur datenbasierten Modellierung, denn je leistungsfähiger die Modellierungsmethode ist, desto komplexere Probleme lassen sich lösen, insbesondere auch dann, wenn nur sehr wenig Vorwissen verfügbar ist. In diesem Beitrag werden drei neue Strategieelemente für die datenbasierte Modellierung vorgestellt.

Das Konzept der impliziten Modellierung verhindert, dass sich etwaige in den Daten vorhandene Mehrdeutigkeiten bzw. Widersprüchlichkeiten auf den eigentlichen Model-

lierungsprozess störend auswirken. Statt dessen wird die Möglichkeit eröffnet, erst nach der Modellerstellung durch Auslegung einer entsprechenden Defuzzifizierungsvorschrift festzulegen, wie die Mehrdeutigkeiten zu berücksichtigen sind. Das Konzept der impliziten Modellierung kann eigenständig oder in Verbindung mit dem Fuzzy-ROSA-Verfahren genutzt werden. Der inkrementelle Relevanzindex dient im Rahmen des Fuzzy-ROSA-Verfahrens zum Hypothesentest und zur Bewertung von Fuzzy-Regeln und erlaubt eine nachträgliche anforderungsspezifische Anpassung des resultierenden Fuzzy-Modells. Hiermit lassen sich gewisse Nachteile bekannter Test- und Bewertungsverfahren für Regeln vermeiden. Das Rauigkeitsmaß dient zur Voranalyse und ggf. Glättung der Daten, auf denen eine datenbasierte Modellierung aufsetzen soll. Weiterhin erlaubt es eine Bewertung und einen Vergleich der resultierenden Modelle.

2 Datenbasierte Modellierung

2.1 Grundaufgabe

Wir gehen von der Grundaufgabe aus, dass Datensätze (Messdatenvektoren) der Form

$$\boldsymbol{z}_j = (\boldsymbol{x}_j, y_j),\ j = 1,2,...,N \tag{1}$$

mit

$$\boldsymbol{x}_j^T = (x_{1,j}, x_{2,j}, ..., x_{n,j}) \tag{2}$$

vorliegen. Darin sind die $\boldsymbol{x}_j$ die zu einem Modelleingangsgrößenvektor zusammengefassten, im Datensatz $\boldsymbol{z}_j$ enthaltenen Werte der Eingangsgrößen $x_1, x_2, ..., x_n$ eines zu modellierenden Originalsystems und y_j die dazugehörigen Werte der Ausgangsgröße *y* (Bild 1).

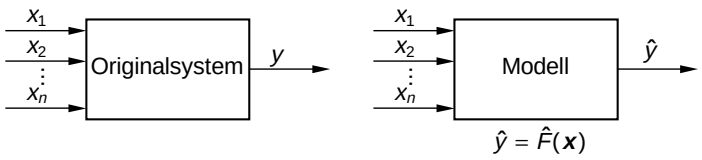


Bild 1: Originalsystem, an dem Datensätze erhoben werden (links). Gesuchtes Modell, das datenbasiert generiert werden soll (rechts).

Die Aufgabe besteht darin, ein Modell zu erstellen, dessen Verhalten $\hat{y} = \hat{F}(\boldsymbol{x})$ mit dem Eingangs-Ausgangsverhalten des zu modellierenden Systems möglichst gut übereinstimmt: Zunächst ist eine gute Übereinstimmung hinsichtlich der als Lerndaten verwendeten Messdaten erwünscht. Darüber hinaus soll das Modell ein sinnvolles interpolatorisches und extrapolatorisches Verhalten aufweisen, d. h. es soll auch für Eingangsgrößenvektoren, die nicht als Lerndaten verwendet worden sind, plausibel sein.

2.2 Anwendungsgrenzen bekannter Modellierungsverfahren bei mehrdeutigen Daten

Die meisten bekannten Verfahren zur datenbasierten Modellierung gehen davon aus, dass der Zusammenhang zwischen dem jeweiligen Ausgangsgrößenwert y_j des Originalsystems und dem dazugehörigen Eingangsgrößenvektor $\boldsymbol{x}_j$ einer expliziten *eindeu-*

tigen Funktion $y = F(\boldsymbol{x})$ genügt. Zur Modellerstellung wird zunächst eine *Strukturwahl* vorgenommen, beispielsweise in Form eines neuronalen Netzes. Anschließend werden die in der Modellstruktur noch einstellbaren Parameterwerte durch eine Parameteroptimierung so eingestellt, dass das Modellverhalten $\hat{y} = \hat{F}(\boldsymbol{x})$ möglichst gut mit dem durch die Datensätze beschriebenen Zusammenhang übereinstimmt. Die so arbeitenden Verfahren haben allerdings zwei bekannte Nachteile: Erstens ist das resultierende Modell meistens intransparent. Zweitens stößt man auf Anwendungsgrenzen, wenn die Datensätze keinen eindeutigen, sondern einen mehrdeutigen Zusammenhang zwischen einem Eingangsgrößenvektor $\boldsymbol{x}$ und einer Ausgangsgröße *y* zeigen (Bild 2). Diese Verfahren beseitigen nämlich solche Mehrdeutigkeiten durch einen nur bedingt durchschaubaren und daher nicht notwendigerweise sachgerechten Kompromiss.

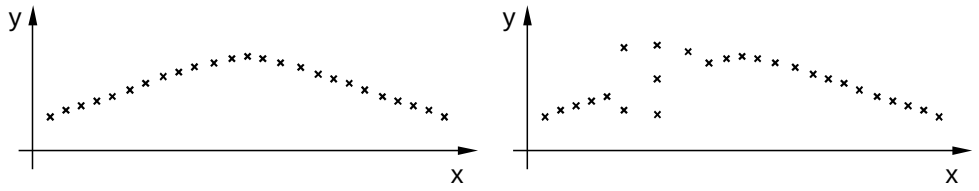


Bild 2: Datensätze mit und ohne Mehrdeutigkeit für ein System mit einer Eingangsgröße x und einer Ausgangsgröße y (links bzw. rechts).

Ein konzeptionell anderes Verfahren – das Fuzzy-ROSA-Verfahren – zielt auf die datenbasierte Generierung von möglichst transparenten Modellen ab [1]. Hierzu wird die Nachbildung des durch die Datensätze beschriebenen Verhaltens des Originalsystems dadurch modularisiert, dass zunächst statistisch abgesicherte Regeln bestimmt werden, die jeweils nur einen Teilaspekt dieses Zusammenhangs beschreiben. Anschließend werden diese Regeln zu dem gesuchten Gesamtmodell zusammengefügt. Wenn allerdings Mehrdeutigkeiten in den Datensätzen vorliegen, hat auch dieses Verfahren Anwendungsgrenzen: Die Regelgenerierung wird nämlich für jeden vorgesehenen linguistischen Ausgangsgrößenwert, wie *verschwindend*, *klein*, *mittel*, *groß* und *sehr groß*, jeweils voneinander getrennt vorgenommen. Deshalb führen Mehrdeutigkeiten u. U. dazu, dass nicht alle benötigten Regeln durch ausreichend viele Datensätze statistisch abgesichert werden können und daher auch nicht generiert werden (Bild 3).

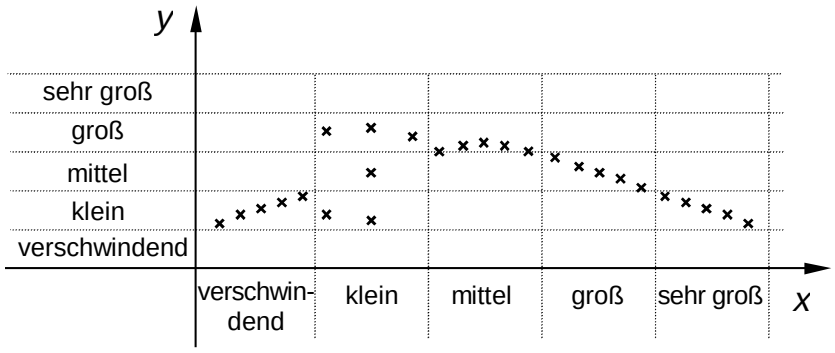


Bild 3: Die Widersprüchlichkeit in den dargestellten Datensätzen behindert die statistische Absicherung von Regeln mit der Prämisse *x = klein*.

3 Implizite Modellierung

3.1 Das Grundkonzept

Das Grundkonzept der impliziten Modellierung zielt eigens auf die datenbasierte Modellierung von Eingangs-Ausgangszusammenhängen ab, die Mehrdeutigkeiten zeigen. Das Konzept sieht zwei Modellierungsschritte vor. Im ersten Schritt wird auf der Basis der Datensätze (1) ein Modell mit den Eingangsgrößen $x_1, x_2, ..., x_n$ sowie y und der Ausgangsgröße μ erzeugt. Das Konstruktionsprinzip dieses Modells besteht darin, dass der Ausgangsgrößenwert $\mu(\boldsymbol{z})$ dann und nur dann groß ist, wenn der Vektor $\boldsymbol{z}$ in der Nachbarschaft mindestens eines der als Lerndaten verwendeten Messdatenvektoren $\boldsymbol{z}_j$ liegt. Anderenfalls ist der Ausgangsgrößenwert klein. Dieses Modell hat damit die Eigenschaft, dass der gesuchte explizite Zusammenhang $\hat{y} = \hat{F}(\boldsymbol{x})$ implizit in der qualitativen Beziehung $\mu(\boldsymbol{x}, y) \approx$ *groß* steckt. Deswegen wird dieses Modell im folgenden *implizites Modell* genannt. Für die Festlegung der Nachbarschaft zwischen zwei Vektoren $\boldsymbol{z}$ und $\boldsymbol{z}_j$ sind wählbare Abstandsfunktionen $d(\boldsymbol{z}_j, \boldsymbol{z})$ bzw. $d_j(\boldsymbol{z}_j, \boldsymbol{z})$ vorgesehen, die einen um so größeren Funktionswert annehmen, je kleiner der Abstand zwischen den Vektoren $\boldsymbol{z}_j$ und $\boldsymbol{z}$ nach Maßgabe einer wählbaren Norm $\| \boldsymbol{z} - \boldsymbol{z}_j \|$ ist. Dieser erste Modellierungsschritt enspricht der in der Statistik behandelten Aufgabenstellung, ausgehend von Messdatenvektoren (1) eine Wahrscheinlichkeitsdichte $\mu(\boldsymbol{z})$ zu ermitteln.

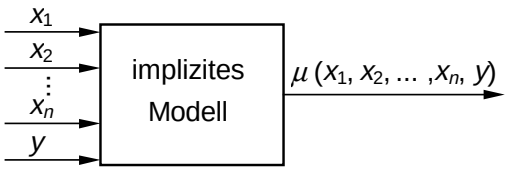


Bild 4: Implizites Modell. Es liefert genau für solche Eingangsvektoren $\boldsymbol{z} = (\boldsymbol{x}, y)$ große Ausgangsgrößenwerte, die in der Nähe zumindest eines Datenvektors $\boldsymbol{z}_j = (\boldsymbol{x}_j, y_j)$ liegen.

Die nach diesem Prinzip konstruierte Funktion $\mu(x_1, x_2, ..., x_n, y)$ des impliziten Modells gibt für beliebig vorgegebene Werte $x_1, x_2, ..., x_n$ und y Auskunft darüber, in welchem Grade durch die Datensätze gestützt wird, dass das Originalsystem für die Eingangsgrößenwerte $x_1, x_2, ..., x_n$ den Ausgangsgrößenwert y liefert. Die Funktion $\mu(x_1, x_2, ..., x_n, y)$ lässt sich daher als Fuzzy-Zugehörigkeitsfunktion interpretieren. In einem zweiten Konstruktionsschritt wird das implizite Modell zur Konstruktion des Gesamtmodells mit den Eingangsgrößen $x_1, x_2, ..., x_n$ und der Ausgangsgröße $\hat{y}$ genutzt (Bild 5). Hierzu werden beispielsweise nacheinander jeweils die aktuell gegebenen Werte $x_1, x_2, ..., x_n$ zusammen mit verschiedenen möglichen Werten $y(i)$ auf das implizite Modell geschaltet. Aus den Werten $y(i)$ und den vom impliziten Modell gelieferten Ausgangsgrößenwerten $\mu(x_1, x_2, ..., x_n, y(i))$, abgekürzt $\mu(y(i))$, erzeugt eine Defuzzifizierungseinrichtung den Ausgangsgrößenwert $\hat{y}$.

Der Vorteil dieses Grundkonzeptes zur datenbasierten Modellierung liegt darin, dass die Erstellung des impliziten Modells nicht durch Mehrdeutigkeiten in den Daten (d. h. durch das Vorkommen von Datenvektoren $(\boldsymbol{x}_i, y_i)$ und $(\boldsymbol{x}_k, y_k)$ mit $\boldsymbol{x}_i \approx \boldsymbol{x}_k$ und $y_i \neq y_k$ tangiert wird: Der Zusammenhang $\mu(\boldsymbol{x}, y)$, den das implizite Modell beschrei-

ben soll, ist stets eindeutig. Etwaige, in den Datenvektoren enthaltene Mehrdeutigkeiten der o. a. Form werden durch das implizite Modell erfasst und an das Defuzzifizierungsmodul weitergeleitet. Hierdurch wird die Möglichkeit eröffnet, Mehrdeutigkeiten bei ungeändertem impliziten Modell durch eine entsprechende Auslegung der Defuzzifizierungsvorschrift anwendungsgerecht zu berücksichtigen. Besonders geeignet hierfür ist das Inferenzfilterverfahren, da damit stufenlose Kompromisse zwischen herkömmlichen Defuzzifizierungsverfahren einstellbar sind [2].

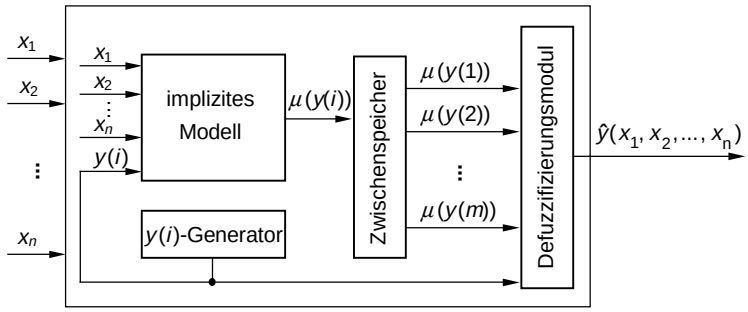


Bild 5: Nutzung des impliziten Modells zum Aufbau des gesuchten Modells $\hat{y} = \hat{F}(\boldsymbol{x})$

3.2 Modularer Aufbau des impliziten Modells

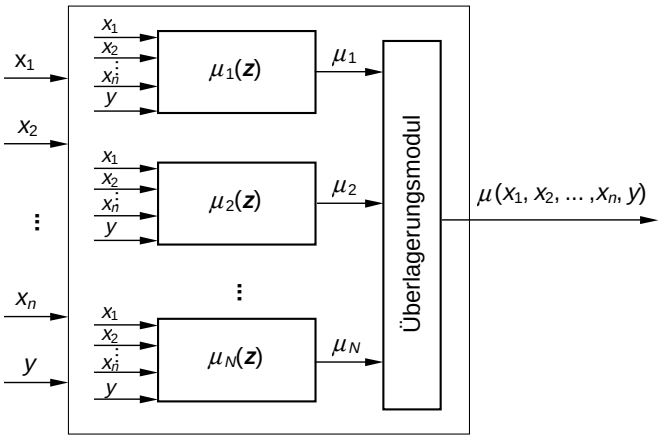


Bild 6: Modularer Aufbau des impliziten Modells aus Teilfunktionen $y_j(\boldsymbol{z})$.

Ein mögliches Konzept zur Konstruktion der gesuchten Funktion $\mu(\boldsymbol{z})$ besteht darin, für jeden Punkt (Datensatz) $\boldsymbol{z}_j$ eine Teilfunktion $\mu_j(\boldsymbol{z})$ mit $0 \le \mu_j(\boldsymbol{z}) \le 1$ anzusetzen, die für $\boldsymbol{z} = \boldsymbol{z}_j$ den Funktionswert 1 und für $\boldsymbol{z} \neq \boldsymbol{z}_j$ umso kleinere Funktionswerte annimmt, je größer der Wert einer geeignet gewählten Abstandsfunktion $d(\boldsymbol{z}_j, \boldsymbol{z})$ bzw. $d_j(\boldsymbol{z}_j, \boldsymbol{z})$ ist. Die gesuchte Funktion $\mu(\boldsymbol{z})$ wird durch Überlagerung dieser Teilfunktionen $\mu_j(\boldsymbol{z})$ - beispielsweise durch die gewöhnliche Summe oder einen Fuzzy-ODER-Operator – erzeugt (Bild 6). Diese Vorgehensweise entspricht dem Parzen-Windows-Verfahren zur Approximation von Wahrscheinlichkeitsdichten, ausgehend von Datenpunkten [3]. Beispiele für solche Funktionsansätze sind

$$\mu_j(\boldsymbol{z}) = e^{-q_j \|\boldsymbol{z}_j - \boldsymbol{z}\|} \tag{3}$$

mit $q_j > 0$ und einer wählbaren Norm $\|...\|$ sowie

$$\mu_j(\boldsymbol{z}) = e^{-(\boldsymbol{z}_j-\boldsymbol{z})^T \boldsymbol{Q}_j (\boldsymbol{z}_j-\boldsymbol{z})} \tag{4}$$

mit einer positiv definiten symmetrischen Matrix $\boldsymbol{Q}_j$. Eine solche punktbasierte Konstruktion von $\mu(\boldsymbol{z})$ ist allerdings bei einer großen Anzahl von Lerndatensätzen ungünstig. Diesem Nachteil lässt sich entgegenwirken, indem Cluster von eng benachbarten Datenpunkten durch je einen einzigen Punkt, z. B. durch ein Clusterzentrum, ersetzt werden (Bild 7).

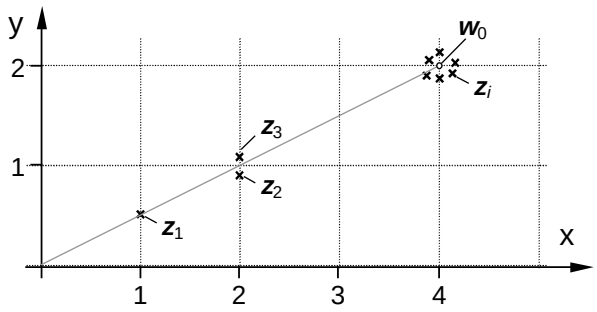


Bild 7: Lerndaten $\boldsymbol{z}_1, \boldsymbol{z}_2$ und $\boldsymbol{z}_3$ sowie *r* weitere, eng benachbarte Lerndaten $\boldsymbol{z}_i$. Diese Punkte $\boldsymbol{z}_i$ werden durch den Punkt $\boldsymbol{w}_0$ ersetzt.

Bei einer solchen Ersetzung vieler Lerndatensätze durch einen einzigen Punkt entsteht die im nächsten Abschnitt aufgegriffene Frage, ob dieser Punkt bei der nachfolgenden Modellierung im Vergleich zu den sonstigen Punkten mit einem größeren Gewicht zu berücksichtigen ist oder nicht.

3.3 Abstandsbasierte Datengewichtung

In vielen – nicht allen – Anwendungen kann unterstellt werden, dass die wiederholte Beobachtung ein und desselben Datenpunktes keine zusätzlichen Informationen liefert. In diesen Fällen ist es folgerichtig, die für die Modellierung verwendeten Datensätze $\boldsymbol{z}_j$ so zu gewichten, dass die in der Nachbarschaft vieler anderer Datenpunkte liegenden Datenpunkte $\boldsymbol{z}_j$ im Vergleich zu isoliert liegenden Punkten mit einem geringeren Gewicht berücksichtigt werden. Dies leistet beispielsweise die Gewichtung gemäß

$$p_j = \frac{1}{N_j} = \frac{1}{\sum_{i=1}^{N} e^{-q\|\boldsymbol{z}_j-\boldsymbol{z}_i\|}} \tag{5}$$

mit $q > 0$. Die Größe N_j ist nämlich ein Maß dafür, wie viele Punkte (neben $\boldsymbol{z}_j$ selbst) in der Nachbarschaft von $\boldsymbol{z}_j$ liegen.

Diese abstandsbasierte Datengewichtung wird direkt auf die ursprünglichen Messdaten $\boldsymbol{z}_j$ angewendet. Werden dann gewisse Messdaten durch ein Clusterzentrum ersetzt, so erhält dieses das Gesamtgewicht der ersetzten Punkte.

3.4 Strukturwahl für die Teilfunktionen

Die im Bild 7 gezeigten Lerndaten legen ein Modell mit dem Verhalten $\hat{y} = 0{,}5x$ nahe. Bei Anwendung des Konzepts der impliziten Modellierung (Bild 5) mit Defuzzifizierung nach der Maximummethode wird dieses gewünschte Modellerhalten dann erhalten,

wenn das implizite Modell $\mu(x,y)$ für jeden konstantgehaltenen Wert von *x* das absolute Maximum an der Stelle $y = 0{,}5x$ annimmt. Mit kugelsymmetrischen Teilfunktionen, wie beispielsweise Gl. (3), lässt sich dieses Verhalten nur näherungsweise erreichen. Die resultierende Modellfunktion $\hat{y} = \hat{F}(x)$ zeigt einen mehr oder minder stark ausgeprägten wellenförmigen Verlauf. Im Gegensatz dazu erlaubt es der Ansatz (4), die Teilfunktionen so an die Lerndaten anzupassen, dass sie große Funktionswerte bevorzugt auf der genannten Geraden $y = 0{,}5x$ annehmen. Dieser Ansatz erhält jedoch bei größeren Werten der Dimension *n* des Eingangsraumes mit $n(n+1)/2$ eine inpraktikabel große Anzahl von einstellbaren Parametern. Durch Beschränkung auf Diagonalmatrizen $\boldsymbol{Q}$ kann zwar die Anzahl der Parameter auf *n* verringert werden. Allerdings führen derartige Matrizen zu einer unerwünschten Bevorzugung von achsenparallelen Datenmustern. Dieser Nachteil lässt sich durch Verwendung von Matrizen der Form

$$\boldsymbol{Q}(a,b,\boldsymbol{h}) = \left(\frac{1}{a^2}\boldsymbol{h}\boldsymbol{h}^T + \frac{1}{b^2}\left(\boldsymbol{E} - \boldsymbol{h}\boldsymbol{h}^T\right)\right) \tag{6}$$

mit $|\boldsymbol{h}| = 1$ vermeiden. Diese Matrizen enthalten nur *n*+1 wählbare Parameter, lassen sich aber dennoch flexibel an die Daten anpassen: Die Niveauflächen der zugehörigen quadratischen Form $\boldsymbol{x}^T\boldsymbol{Q}(a,b,\boldsymbol{h})\boldsymbol{x} = 1$ sind Ellipsoide, die in Bezug auf $\boldsymbol{h}$ rotationssymmetrisch sind und in Richtung von $\boldsymbol{h}$ die Halbachse *a* und senkrecht dazu stets die Halbachse *b* aufweisen (Bild 8).

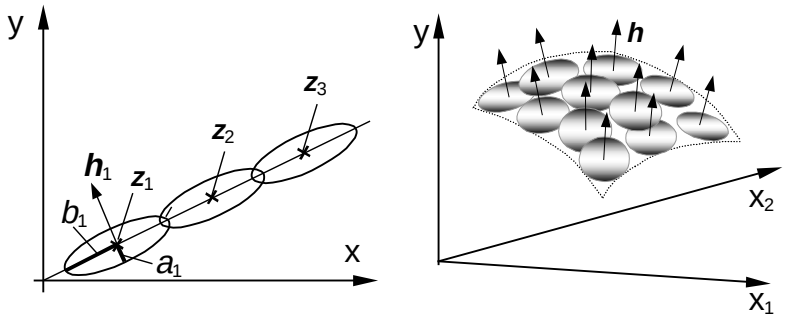


Bild 8: Illustration der Flexibilität von Matrizen der Form (6) zum Aufbau von Teilfunktionen $\mu_k(\boldsymbol{z})$, ausgehend von Lerndaten $\boldsymbol{z}_i$. Für den Fall *n* = 1 (links) sind die Niveaulinien der Teilfunktionen, für den Fall *n* = 2 (rechts) Projektionen der Niveauflächen auf die gekrümmte Fläche, in der die nicht eingezeichneten Datenvektoren $\boldsymbol{z}_j$ liegen, dargestellt.

3.5 Heuristische Konzepte zur Optimierung der Teilfunktionen

Sind, ausgehend von den Lerndaten, gewisse Punkte $\boldsymbol{z}_j$ für den Ansatz von Teilfunktionen beispielsweise mit der Form (3) oder (4) ausgewählt worden, so verbleibt die Aufgabe, die darin noch freien Parameter geeignet zu wählen. Hierzu können folgende Erwägungen dienen:

Zu den Lerndaten $\boldsymbol{z}_j$ lässt sich stets ein kleinstes quaderförmiges Arbeitsgebiet *G* mit dem dazugehörigen Volumen *V*, in dem die Lerndaten liegen, angeben. Die Größe

$$V_j = \frac{1}{V}\int \mu_j(\boldsymbol{z})\, d\,\boldsymbol{z} \tag{7}$$

lässt sich als Maß für die Größe des *Wirkungsbereiches* der Teilfunktion $\mu_j(\boldsymbol{z})$ interpretieren. Es ist zu fordern, dass jede Teilfunktion einen angemessenen Wirkungsbereich hat und dass die Wirkungsbereiche sämtlicher Teilfunktionen zusammengenommen das Arbeitsgebiet *G* akzeptabel ausschöpfen. Weiterhin liefert die Größe

$$s_j = \frac{1}{V_j\, p} \sum_{i=1}^{N} p_i \mu_j(\boldsymbol{z}_i) \tag{8}$$

mit $p = \sum p_i$ ein Maß dafür, in welchem Grade jede Volumeneinheit des Wirkungsbereiches einer Teilfunktion durch die Lerndaten unterstützt wird. Es ist zu fordern, dass s_j für alle Teilfunktionen einen möglichst hohen Wert annimmt und dass sämtliche Teilfunktionen zusammengenommen die Lerndaten möglichst gut abdecken.

Die genannten Forderungen sind gegenläufig. Dementsprechend sind sinnvolle Kompromisse zu schließen. Hierfür sind unterschiedliche Heuristiken naheliegend, von denen hier zwei skizziert werden: Zur Wahl eines geeigneten Wertes von q_j in Gl. (3) werden der Reihe nach diejenigen Punkte $\boldsymbol{z}_i$ bestimmt, die dem Punkt $\boldsymbol{z}_j$ am nächsten liegen. Von diesen Punkten werden so viele berücksichtigt, bis ihr Gesamtgewicht jeweils einen vorgegebenen Wert *c*, z. B. c = *n*, annimmt. Wird dieses Gewicht überschritten, wird das Gewicht des letzten Punktes entsprechend reduziert. Für die so bestimmten Nachbarn $\boldsymbol{z}_i$ des Punktes $\boldsymbol{z}_j$ wird der Mittelwert der mit p_i gewichteten Abstände zum Punkt $\boldsymbol{z}_j$ bestimmt. Der reziproke Wert dieses Mittelwertes liefert einen Anhaltspunkt für die Wahl von q_j. Zur Wahl der Matrix $\boldsymbol{Q}_j$ gemäß Gl. (6) für die Teilfunktion (4) wird diejenige Hyperebene $\boldsymbol{h}^T(\boldsymbol{z} - \boldsymbol{z}_j) = 0$ bestimmt, die bestmöglich durch die Punkte $\boldsymbol{z}_i$ verläuft. Dabei ergibt sich der Vektor $\boldsymbol{h}$ analytisch durch Minimierung eines quadratischen Fehlers, der durch Aufsummierung der gewichteten Fehlerbeiträge $(\boldsymbol{h}^T(\boldsymbol{z}_i - \boldsymbol{z}_j))^2$ entsteht. Zur Fehlergewichtung werden einerseits die Gewichte p_i der jeweiligen Punkte $\boldsymbol{z}_i$ sowie eine Größe – beispielsweise $\exp(-q\,|\,\boldsymbol{z}_j - \boldsymbol{z}_i\,|)$ – herangezogen, die umso kleinere Werte annimmt, je größer der Abstand zwischen den Punkten $\boldsymbol{z}_i$ und $\boldsymbol{z}_j$ ist. Die Größe des verbleibenden minimalen Fehlers liefert einen Anhaltspunkt für die Wahl des Verhältnisses a/b. Anhaltspunkte für eine sinnvolle Wahl des Absolutwertes dieser Größen ergeben sich aus Erwägungen analog zur oben skizzierten Möglichkeit zur Wahl von q_j in Gl. (3).

3.6 Relevanzbasierte Optimierung der Teilfunktionen

Im folgenden wird skizziert, dass die Optimierung der Teilfunktionen durch Nutzung der vom Fuzzy-ROSA-Verfahren her bekannten Maße zum Hypothesentest und zur Bewertung von Regeln systematisiert werden kann. Ausgangspunkt hierfür ist der Gedanke, dass sich jede Teilfunktion $\mu_j(\boldsymbol{z})$ als eine Regel der Form

WENN < $\mu(\boldsymbol{z}) = gross$ >
DANN < Es gibt Lerndatensätze $\boldsymbol{z}_j$ in der Nachbarschaft von $\boldsymbol{z}$ > (9)

interpretieren lässt [4]. Deshalb kann mit den vom Fuzzy-ROSA-Verfahren her bekannten Verfahren zum Test von Hypothesen und zur Bewertung von Regeln festge-

stellt werden, ob eine Teilfunktion $\mu_j(\boldsymbol{z})$ in Anbetracht der Lerndaten als eine statistisch relevante Regel anzusehen und wie groß ggf. ihre Relevanz ist. Dies eröffnet die Möglichkeit, jede parametrisierte Teilfunktion $\mu_j(\boldsymbol{z})$ im Sinne einer Maximierung ihres Relevanzgrades zu optimieren.

Diese Übertragung des Fuzzy-ROSA-Verfahrens wird im folgenden anhand des durchsichtigen Sonderfalls, dass die Teilfunktionen $\mu_j(\boldsymbol{z})$ nur einen der beiden Funktionswerte 0 oder 1 annehmen können, illustriert. Hierzu wird unterstellt, dass die Erhebung der *N* Datensätze $\boldsymbol{z}_j$ am Originalsystem aus der Untersuchung einer zufällig ausgewählten Stichprobe im Umfang von insgesamt $M > N$ Einzeluntersuchungen hervorgegangen ist und dass dabei das Ereignis „Der Stichprobenvektor $\boldsymbol{z} = (\boldsymbol{x}, y)$ wird durch das Originalsystem in dem Sinne bestätigt, dass bei Anliegen des Eingangsvektors $\boldsymbol{x}$ der Ausgangswert *y* auftritt“ *N*-mal auftrat und $M - N$ -mal nicht. Die Originaldaten geben nur wieder, welche Datenpunkte $\boldsymbol{z}_j$ im Eingangs-Ausgangsraum beobachtet worden sind. Sie geben keine Auskunft über das Nichtauftreten von Ausgangswerten y_j bei Anlegen von Eingangsvektoren $\boldsymbol{x}_j$. Deshalb muß über dieses Nichtauftreten durch die Wahl von *M* prolemabhängig eine Annahme getroffen werden. Die relative Häufigkeit für das Auftreten dieses Ereignisses im Arbeitsgebiet *G* ist durch $\hat{p} = N / M$ gegeben. Hieraus und aus dem Stichprobenumfang *M* kann analog zur Vorgehensweise beim Fuzzy-ROSA-Verfahren ein Konfidenzintervall *I* für die Wahrscheinlichkeit *p* des Auftretens des o. a. Ereignisses im Arbeitsgebiet *G* ermittelt werden. Der auf das Volumen V_j des Wirkungskreises von $\mu_j(\boldsymbol{z})$ entfallende anteilige Stichprobenumfang ist $M_j = M V_j / V$, wobei *V* das Volumen des Arbeitsgebietes ist. Hieraus und aus der Anzahl N_j der Datensätze $\boldsymbol{z}_j$, die in dem Wirkungsbereich der Teilfunktion liegen, lässt sich analog ein Konfidenzintervall I_j für die Wahrscheinlichkeit p_j des Auftretens des o. a. Ereignisses in diesem Wirkungsbereich ermitteln. Aufsetzend auf diesen Konfidenzintervallen *I* und I_j lässt sich das Relevanzkonzept des Fuzzy-ROSA-Verfahrens direkt zur Bewertung jeder Teilfunktion $\mu_j(\boldsymbol{z})$ anwenden.

Bei der bisher üblichen Anwendung des Fuzzy-ROSA-Verfahrens zur direkten Aufstellung eines expliziten Modells $\hat{y} = \hat{F}(\boldsymbol{z})$ wird die Regelgenerierung für jeden vorgesehenen linguistischen Ausgangsgrößenwert getrennt voneinander vorgenommen. Im Gegensatz dazu findet die Regelgenerierung für die Erstellung der impliziten Modellierung en bloc im gesamten Eingangs-Ausgangsraum statt. Damit können Lerndaten, die sich auf unterschiedliche Ausgangsgrößenwerte beziehen, gemeinsam zur Regelgenerierung beitragen. Dies unterstützt die Generierung von Regeln, die sich auch auf nicht in den Lerndaten vorkommende Ausgangsgrößenwerte y_j erstrecken

4 Inkrementeller Relevanzindex

4.1 Relevanzindizes des Fuzzy-ROSA-Verfahrens

Zur datenbasierten Lösung der in Abschnitt 2.1 formulierten Grundaufgabe werden mit dem Fuzzy-ROSA-Verfahren Hypothesen der Form

WENN < Prämisse > DANN < $y = L_i$ > (10)

aufgestellt. Darin wird der Wahrheitswert der Prämisse durch den Eingangsgrößenvektor $\boldsymbol{x}$ definiert. Die Konklusion bezieht sich auf die Ausgangsgröße *y*. Zur Vereinfachung der Darstellung wird im folgenden davon ausgegangen, dass diese Wahrheitswerte nur die Werte 0 oder 1 annehmen können. Der allgemeine Fall lässt sich analog behandeln.

Die im Rahmen des Fuzzy-ROSA-Verfahrens entwickelten Relevanzmaße stützen sich auf folgende Größen: Zum einen wird anhand der Lerndaten ermittelt, wie groß die relative Häufigkeit $\hat{p}_K$ dafür ist, dass die Konklusion erfüllt ist (Bild 9, links). Unter Berücksichtigung des Datenumfangs *N* wird hierzu das Konfidenzintervall I_K ermittelt, das den Wert der Wahrscheinlichkeit p_K für die Erfülltheit der Konklusion abschätzt. Entsprechend wird anhand der Daten die relative Häufigkeit $\hat{p}_{K|P}$ dafür, dass die Konklusion bei Erfülltheit der Prämisse wahr ist, ermittelt. Es wird wiederum das dazugehörige Konfidenzintervall $I_{K|P}$, das die Wahrscheinlichkeit $p_{K|P}$ abschätzt, bestimmt. Von diesen Werten gehen die zentralen Konzepte des Fuzzy-ROSA-Verfahrens zum Test von Hypothesen aus [5, 6]. Falls die Konfidenzintervalle I_K und $I_{K|P}$ disjunkt sind, unterscheiden sich die Wahrscheinlichkeiten p_K und $p_{K|P}$ signifikant. Dann ist die Hypothese (10) eine statistisch relevante Regel, und zwar im Falle $\hat{p}_{K|P} > \hat{p}_K$ eine positive und im Falle $\hat{p}_{K|P} < \hat{p}_K$ eine negative Regel. Als Maß für die Relevanz wird der geeignet normierte Abstand der Konfidenzintervalle verwendet. Der so erklärte Relevanzindex hat zwei Schwachpunkte: Gilt beispielsweise $\hat{p}_K = 0{,}8$ und sind die genannten Konfidenzintervalle nicht disjunkt, wird die Hypothese (10) nicht zur Regel erhoben, obwohl sie eine hohe Trefferquote hat. Sind die Konfidenzintervalle disjunkt und ist $\hat{p}_{K|P}$ nur geringfügig kleiner als $\hat{p}_K$, wird eine negative Regel generiert. Sie besagt, dass der Ausgangsgrößenwert $y = L_i$ bei Erfülltheit der Prämisse nicht vorliegt, obwohl dies überwiegend der Fall ist. Zur Beseitigung des ersten Mangels wurde die normierte bzw. die konfidente Trefferquote eingeführt. Sie orientiert sich allein an der relativen Häufigkeit $\hat{p}_{K|P}$ bzw. an dem dazugehörigen Konfidenzintervall $I_{K|P}$. Wenn $\hat{p} > 0{,}5$ gilt bzw. der linke Rand des Konfidenzintervalls größer als 0,5 ist, wird die Hypothese (10) zu einer positiven Regel erhoben. Wenn $\hat{p}_{K|P} < 0{,}5$ gilt bzw. der rechte Rand des dazugehörigen Konfidenzintervalls kleiner als 0,5 ist, wird die Hypothese (10) zu einer negativen Regel erhoben. Auch dieses Konzept hat Schwachpunkte: Erstens zielt es nur darauf ab, wie erfolgreich eine Regel ist, unabhängig davon, ob es einen statistisch abgesicherten Zusammenhang zwischen Prämisse und Konklusion gibt. Zweitens ist die Schwelle von 0,5 % für das Akzeptieren einer positiven bzw. negativen Regel fragwürdig. Sind beispielsweise drei linguistische Ausgangsgrößenwerte vorgesehen, auf die sich die Daten gleichmäßig verteilen, so gilt für jede Konklusion $p_K = 0{,}33$. In diesem Fall ist das Kriterium "$> 0{,}5$" für das Akzeptieren einer positiven Regel sinnvoll, das Kriterium "$< 0{,}5$" für das Akzeptieren einer negativen Regel jedoch nicht.

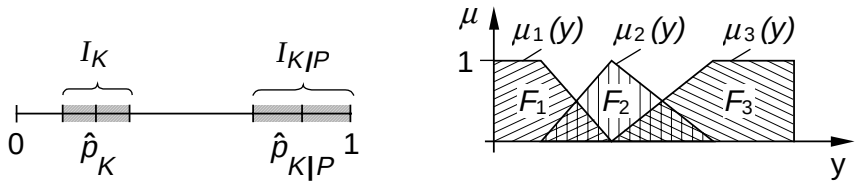


Bild 9: Zur Definition der Relevanzindizes des Fuzzy-ROSA-Verfahrens (links), Zugehörigkeitsfunktionen $\mu_i(y)$ zur Modellierung der linguistischen Werte L_i (rechts)

Es hängt von der Anwendungssituation ab, welches dieser Maße in Abwägung der Vorzüge und Nachteile am angemessensten ist. Da dies meist nicht vorab entscheidbar ist, muß die datenbasierte Modellierung mit dem Fuzzy-ROSA-Verfahren in der Praxis meist mehrfach für unterschiedliche Test- und Bewertungsmaße durchgeführt werden.

4.2 Inkrementeller Relevanzindex

Der inkrementelle Relevanzindex zielt auf eine Überwindung der o. a. Schwachstellen ab. Zur Erläuterung der Grundidee betrachten wir den Fall, dass Hypothesen der Form (10) zu testen sind und dabei insgesamt drei linguistische Werte L_i mit Zugehörigkeitsfunktionen $\mu_i(y)$ in Form eines Fuzzy-Informationssystems vorgesehen sind (Bild 9, rechts). Die unter den Graphen dieser Funktionen liegenden Flächen sind mit F_i, ihre Summe mit *F* bezeichnet. Ausgangspunkt ist der Gedanke, neben Regeln (10) zusätzliche universelle Regeln der Form

$$\text{WENN} < 1 = 1 > \quad \text{DANN} \quad y = L_i \tag{11}$$

einzuführen, deren Prämissen stets erfüllt sind. Diesen Regeln wird der Bewertungsindex

$$\rho_i = \lambda\, q\, \hat{p}_{y=L_i}\, \frac{F}{F_i} \tag{12}$$

zugeteilt. Darin ist λ ein Entwurfsparameter mit $0 \le \lambda \le 1$, mit dem eingestellt werden kann, in welchem Maße die universellen Regeln (11) berücksichtigt werden sollen. Der Faktor *q* ergibt sich aus der Normierungsbedingung $\sum \rho_i = \lambda$. Für die Generierung der üblichen Regeln vom Typ (10) werden die relativen Häufigkeiten $\hat{p}_K$ und $\hat{p}_{K|P}$ sowie die dazugehörigen Konfidenzintervalle bestimmt und nur solche Hypothesen als relevante positive bzw. negative Regeln akzeptiert, für die die Konfidenzintervalle disjunkt sind. Anders als üblich wird hier allerdings als Relevanzgrad die Differenz σ_i zwischen der unbedingten und der bedingten relativen Häufigkeit $\hat{p}_{y=L_i}$ und $\hat{p}_{y=L_i|Prämisse}$ (inkrementelle Trefferquote) bzw. alternativ die Differenz zwischen $\hat{p}_{y=L_i}$ und dem nächstgelegenen Rand des zu $\hat{p}_{y=L_i|Prämisse}$ gehörigen Konfidenzintervalls (konfidente inkrementelle Trefferquote) verwendet. Erweist sich eine Hypothese mit dem linguistischen Ausgangsgrößenwert L_i als relevante positive bzw. negative Regel mit dem Relevanzgrad σ_i, werden alle Regeln (10), die sich auf die übrigen linguistischen Werte beziehen, als relevante negative bzw. positive Regeln erklärt. Dabei werden die zugehörigen Relevanzgrade so festgesetzt, dass ihre Summe gerade σ_i ist und die Verhältnisse der Einzelwerte den Verhältnissen der Werte (12) für $\lambda = 1$ entsprechen. Somit liegen die

Regeln (11) im allgemeinen sowohl in positiver wie auch in negativer Form vor. Die Relevanzgrade werden als inkrementelle Werte behandelt: Sie werden für die positiven und die negativen Regeln jeweils voneinander getrennt zu einem resultierenden gesamten Relevanzgrad addiert. Im Gegensatz zur Trefferquote behandelt diese Vorgehensweise positive und negative Regeln gleichberechtigt. Ferner kann hier – im Gegensatz zu [6] – noch nach erfolgter Modellierung durch Wahl des Parameters λ festgelegt werden, welcher Kompromiss zwischen den gegenläufigen Gesichtspunkten „Erfolg einer Regel“ und „statistische Absicherung“ geschlossen werden soll.

5 Rauigkeitsmaß

5.1 Motivierung

Bei der datenbasierten Modellierung eines Systems mit nur je einer Eingangs- und Ausgangsgröße ist es sinnvoll, sich vorab die Verteilung der Lerndaten im *x*-*y*-Raum anzusehen. So kann erkannt werden, ob die Daten ggf. Ausreißer enthalten und ob der zu modellierende Zusammenhang gutmütig oder eher bizarr ist. Daraus ergeben sich Hinweise auf den Schwierigkeitsgrad der Modellierungsaufgabe sowie auf sinnvolle Strukturansätze für die Modellierung. Ebenso ist es nach Durchführung der Modellierung sinnvoll, das Modellverhalten grafisch zu veranschaulichen. Hieraus ist ablesbar, wie groß die Abweichungen zwischen dem Verhalten des Originalsystems und des Modells sind. Darüber hinaus ist aber auch zu erkennen, ob die Glattheit bzw. Rauigkeit der Modellfunktion $\hat{y} = \hat{F}(\boldsymbol{x})$ dem Verhalten des Originalsystems entspricht.

Meistens wird nicht nur ein möglichst kleiner Fehler zwischen Modell und Originalsystem angestrebt, sondern es wird darüber hinaus gewünscht, dass das Modellverhalten möglichst glatt ist. Wird beispielsweise durch datenbasierte Modellierung des Verhaltens eines Prozessbedieners ein Fuzzy-Regler entwickelt, so ist es im Hinblick auf die Beanspruchung des Stellgliedes im allgemeinen erwünscht, dass kleine Variationen der Eingangsgrößen des Reglers nicht zu großen Variationen der Ausgangsgröße führen. Ähnliches gilt beim datenbasierten Entwurf eines Fuzzy-Moduls zur Gütebewertung eines Prozesses. In diesem Fall ist es unerwünscht, dass kleine Änderungen der Eingangsgrößenwerte des Gütemaßes (d. h. kleine Änderungen der Merkmalsausprägungen) zu unmotivierten großen Änderungen des ermittelten Gütewertes führen. Insbesondere ist es für die Prozessoptimierung abträglich, wenn die Ausgangsgröße des Fuzzy-Moduls ein unmotiviertes, nicht monotones Verhalten zeigt.

5.2 Rauigkeitsmaß für den *n*-dimensionalen Fall

Im Fall von $n > 2$ Eingangsgrößen scheidet eine visuelle Inspektion der Daten sowie des Modellverhaltens aus. Im folgenden wird ein Rauigkeitsmaß angegeben, das auch im höherdimensionalen Fall einfach berechenbar ist.

Zu jedem Datensatz $\boldsymbol{z}_j = (\boldsymbol{x}_j, y_j)$ werden im *n*-dimensionalen $\boldsymbol{x}$-Raum die (*n*+1) nächsten Nachbarn $\boldsymbol{z}_{j,1}$, $\boldsymbol{z}_{j,2}$,..., $\boldsymbol{z}_{j,n}$, $\boldsymbol{z}_{j,n+1}$ bestimmt. Zu je *n* dieser Punkte, die daraus durch Weglassen je eines Nachbarn $\boldsymbol{z}_{j,i}$ hervorgehen, und jeweils dem Aufpunkt $\boldsymbol{z}_j$ wird im (*n*+1)-dimensionalen $\boldsymbol{z}$-Raum die durch diese Punkte laufende Hyperebene und deren Normalenvektor $\boldsymbol{h}_{j,i}$ bestimmt. Dies geschieht durch Auflösen eines linearen Gleichungssystems. Die so erhaltenen $n+1$ Vektoren $\boldsymbol{h}_{j,i}$ werden mit einem Maß für die Kondition dieses linearen Gleichungssystem gewichtet. Aus den so gewichteten Vekto-

ren wird der mittlere Normalenvektor $\bar{\boldsymbol{h}}_j$ bestimmt. Sodann wird der Mittelwert $\Delta\varphi$ der Winkel zwischen den $\boldsymbol{h}_{j,i}$ mit $1 \le i \le n+1$ und $\bar{\boldsymbol{h}}_j$ bestimmt. Wenn der Aufpunkt und seine Nachbarpunkte im $\boldsymbol{z}$-Raum sämtlich in einer Hyperebene liegen, gilt $\Delta\varphi = 0$. Je weniger gut diese Punkte in einer Hyperebene liegen, desto größer wird $\Delta\varphi$. Zum Abfangen des Sonderfalls, dass die Bestimmung der o. g. *n* nächsten Nachbarn nicht eindeutig ist, weil es mehrere Punkte mit gleichem größtem Abstand gibt, werden besondere Vorkehrungen getroffen.

Das so definierte Maß dient zur Beurteilung der lokalen Glattheit. Wird es auf sämtliche Lerndaten $\boldsymbol{z}_j$ angewendet, so lässt sich die globale Rauigkeit der Lerndaten durch den Mittelwert, die Varianz bzw. durch den Maximalwert der Einzelwerte charakterisieren. Entsprechend lässt sich das Verhalten eines Modells durch Auswertung der Modellgleichung $\hat{y} = \hat{F}(\boldsymbol{x})$ für eine repräsentative Menge von Punkten $(\boldsymbol{x}_i, \hat{F}(\boldsymbol{x}_i))$ bewerten. Sind die Lerndaten rauer als aufgrund des verfügbaren Prozessvorwissens zu erwarten ist, kann geschlossen werden, dass die Datenerhebung u. U. fehlerhaft ist. In diesem Fall bietet es sich an, die Daten vor der eigentlich durchzuführenden Modellierung wie folgt zu glätten: Für sämtliche Punkte $\boldsymbol{z}_j$, bei denen der Grad der Rauigkeit eine vorzugebende Schwelle überschreitet, wird der in den Daten gegebene Ausgangsgrößenwert y_j geringfügig im Sinne einer Verringerung der Rauigkeit verändert. So entstehen aus den ursprünglichen Lerndaten modifizierte Lerndaten, auf die die Glättungsprozedur erneut angewendet wird. Dies wird wiederholt, bis schließlich die lokale Rauigkeit überall unterhalb einer vorgegebenen Schwelle liegt.

6 Beispiel

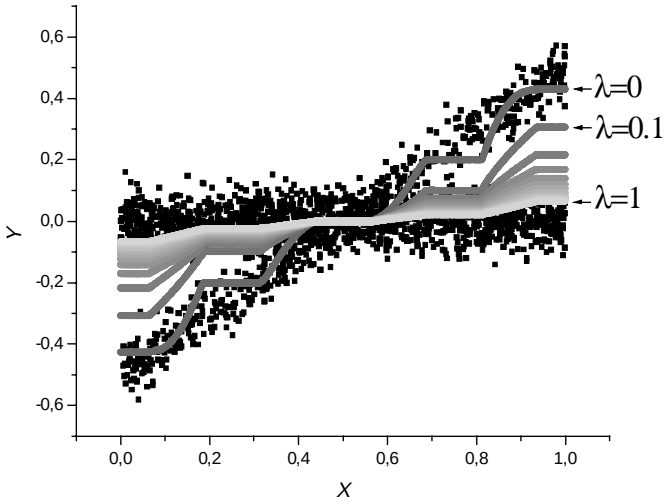


Bild 10: Datenbasierte Modellierung durch Verwendung von universellen Regeln und Regelgenerierung mit dem inkrementellen Relevanzindex. Dargestellt sind die Datenpunkte (teilweise verdeckt) und die Ergebnisse für die Werte 0, 0.1, 0.2, ..., 1 des Anpassungsparameters λ.

Wir beschränken uns auf ein einziges Beispiel. Bild 10 zeigt eine Verteilung von Lerndatensätzen $\boldsymbol{z}_j = (x_j, y_j)$ im $x-y-$Raum. Für 70 % davon gilt näherungsweise $y_j = 0$, für die restlichen 30 % gilt näherungsweise $y_j = -0{,}5 + x_j$. Der Mittelwert al-

ler Werte y_j ist $\bar{y} = 0$. Ausgehend von diesen Lerndaten werden Universalregeln (10) mit dem dazugehörigen Bewertungsindex (12) bestimmt. Zusätzlich werden, basierend auf der o. a. konfidenten inkrementellen Trefferquote σ_i Regeln vom Typ (11) generiert. Für die Verarbeitung der positiven und negativen Regeln wird ein zweisträngiges Fuzzy-System (vgl. [2]) verwendet. Dabei wird zur Akkumulation die gewöhnliche Summe und zur Defuzzifizierung die Schwerpunktmethode eingesetzt. Bild 10 zeigt die Modellierungsergebnisse für unterschiedliche Werte von λ. Für $\lambda = 0$ sind nur die statistisch abgesicherten Regeln (11) wirksam. Damit werden die Daten, die näherungsweise auf der Geraden $y = -0{,}5 + x$ liegen, nachgebildet. Mit steigendem Wert von λ werden zunehmend auch die Universalregeln aktiviert. Anders als in [7], wo ein vergleichbares Beispiel mit den bisher üblichen Regelbewertungen behandelt wird, ist hier nach erfolgter Modellierung einstellbar, welchem Kompromiss zwischen den Aspekten statistische Absicherung und Erfolg der Regeln der Ausgangsgrößenwert des Fuzzy-Systems entsprechen soll.

7 Ausblick

Die vorgeschlagenen Strategieelemente dehnen den Anwendungsbereich bekannter Verfahren zur datenbasierten Modellierung in unterschiedliche Richtungen aus. Dies betrifft insbesondere Modellierungsprobleme mit Mehrdeutigkeiten oder Widersprüchlichkeiten in den Daten. Die beschriebenen formelmäßigen Ausgestaltungen der Strategieelemente sind als erste, bisher nur punktuell erprobte Ansätze zu verstehen. Breitere Erprobungen dürften noch zu Modifikationen im Detail führen.

Literatur

[1] Krone, A.: Datenbasierte Generierung von relevanten Fuzzy-Regeln zur Modellierung von Prozesszusammenhängen und Bedienstrategien. Fortschrittsberichte VDI, Reihe 10, Nr. 615, VDI, Düsseldorf, 1999

[2] Kiendl, H.: Fuzzy Control methodenorientiert. Oldenbourg Verlag, München, Wien, 1997. S. 231 - 264

[3] Duda, R. O., Hart, P. E.: Pattern Classification and Scene Analysis. John Wiley & Sons, New York, Chichester, Brisbane, Toronto, Singapore, 1973, S. 88 – 95

[4] Kiendl, H.: Verfahren zur datenbasierten Konstruktion eines Fuzzy-Moduls und Fuzzy-Modul hierfür. Patentanmeldung DE 10013509.9, 2000

[5] Krone, A., Taeger, H.: Relevance test for fuzzy rules. In *Computational Intelligence*, CI-40/98, ISSN 1433-3325, Collaborative Research Center 531, University of Dortmund, Germany, 1998

[6] Jessen, H., Slawinski, T.: Test and rating strategies for data-based rule generation. In *Computational Intelligence*, CI-39/98, ISSN 1433-3325, Collaborative Research Center 531, University of Dortmund, Germany, 1998

[7] Slawinski, T., Jessen, H., Praczyk, J., Krause, P., Krone, A., Kiendl, H.: Einsatz der datenbasierten Fuzzy-Modellierung für komplexe Anwendungen. GMA/GI Fachtagung *Computational Intelligence* im industriellen Einsatz. VDI Bericht Nr. 1526. Baden-Baden, 2000, S. 119 - 124

Multikriterielle, robuste Fuzzy-Optimierung der Parameter bei der Regelung eines verfahrenstechnischen Prozesses mit großer Messtotzeit

Thomas Bernard, Markoto Sajidman

Fraunhofer-Institut für Informations- und Datenverarbeitung IITB
Fraunhoferstr. 1, 76131 Karlsruhe
Tel. +49 (0) 721 / 60 91-360
Fax +49 (0) 721 / 60 91-413
Email: bnd@iitb.fhg.de, saj@iitb.fhg.de

Zusammenfassung

Zur Lösung des multikriteriellen Optimierungsproblems wurde ein Verfahren zur unscharfen Gewichtung und Verknüpfung der Gütekriterien entwickelt, das auf dem bekannten *Fuzzy Decision Making* [4] basiert. Am konkreten Fallbeispiel wird die Generierung von Fuzzy-Gütekriterien zu Störverhalten, Führungsverhalten, Stabilität und Stellaufwand vorgestellt sowie deren Tauglichkeit zur Lösung des Optimierungsproblems untersucht. Zur Sicherstellung der Robustheit der Lösung wird ein einfacher Ansatz vorgestellt. Abschließend werden Ergebnisse der erfolgreichen Regelparameteroptimierung diskutiert. Der vorliegende Beitrag vermittelt Erfahrungen, die bei der Regleroptimierung für einen komplexen, stark gestörten Industrieprozess mit großen Messtotzeiten und unterschiedlichen, zum Teil gegensätzlichen Güteanforderungen, gewonnen wurden.

1 Motivation

Bei der Optimierung von Regelungssystemen sind in der Praxis oft mehrere unterschiedliche, sich zum Teil widersprechende Gütekriterien mit unterschiedlichen Gewichtungsfaktoren zu berücksichtigen. Bei verfahrenstechnischen Prozessen sind dies zum Beispiel:

- gutes Störverhalten der Regelung, d.h. auftretende Störungen sollen so gut wie möglich gedämpft oder kompensiert werden,
- gutes Führungsverhalten zur Gewährleistung der stationären Genauigkeit durch Kompensation von Drifts und/oder Parameterschwankungen,
- Stabilität des geschlossenen Regelkreises,
- geringer Stellaufwand zur Minimierung des Energieeinsatzes oder zur Vermeidung zusätzlicher Systemanregungen und
- Robustheit, d.h. auch bei nicht genau bekannten Streckenparametern ist das Regelverhalten noch zufriedenstellend.

Im Laufe der Untersuchungen hat sich gezeigt, dass die ersten vier Teilkriterien (Störverhalten, Führungsverhalten, Stabilität, Stellverhalten) explizit im gemeinsamen

Gütemaß zu berücksichtigen sind. Robustheit ist als weitere wichtige Eigenschaft von den optimierten Reglerparametern zu fordern. Somit liegt ein Problem der multikriteriellen Optimierung vor. Von besonderer Bedeutung sind dabei die Aggregationsmechanismen zur Verknüpfung der Teilkriterien sowie eine transparente Gewichtung der Teilkriterien. Ein Ziel des Verfahrens ist es sicherzustellen, dass die einzelnen Teilziele auch wirklich erfüllt werden.

Die meisten bisher veröffentlichten multikriteriellen Regelungsentwurfsverfahren beruhen auf der Optimierung eines vektoriellen Gütekriteriums (vgl. z. B. [1], [2], [3]). Hierbei werden einzelne Teilkriterien zu einem Gütevektor zusammengefasst. Zur Lösung des Optimierungsproblems werden unterschiedliche Möglichkeiten vorgeschlagen, wobei eine Methode darin besteht, eine Lösungsmenge zu bestimmen, bei der jedes Teilkriterium einem bestimmten Anspruchsniveau genügt [2]. Hierbei ergeben sich jedoch oft komplexe Optimierungsprobleme, zu deren Lösung z. B. Evolutionsstrategien eingesetzt werden können [3].

Eine weitere Lösungsmöglichkeit besteht darin, die Teilkriterien zu einem Ersatzkriterium zusammenzufassen und somit das multikriterielle in ein monokriterielles Optimierungsproblem zu überführen [3]. Bei der Erstellung eines Ersatzkriteriums besteht jedoch die Gefahr, dass sich konkurrierende Teilkriterien kompensieren. Da in einem solchen Fall die erhaltene "optimale" Lösung kaum noch einen Rückschluss auf die Erfüllung der einzelnen Teilkriterien zulässt, ist die gegenseitige Kompensation der Gütekriterien oft unerwünscht. In vielen Fällen ist es sinnvoll, eine Fuzzy-UND-Verknüpfung der Gütekriterien zu realisieren, wodurch eine Lösung erhalten wird, die alle Teilkriterien in etwa gleich erfüllt und pareto-optimal ist [2].

In diesem Beitrag wird daher ein Ansatz des Fuzzy Decision Making vorgestellt, bei dem das gemeinsame Gütekriterium durch eine Fuzzy-UND-Verknüpfung der einzelnen Teilkriterien erfolgt. Der verwendete Algorithmus erlaubt eine transparente Gewichtung der einzelnen Teilkriterien entsprechend ihrer Bedeutung. Angewendet wurde das fuzzy-basierte Entwurfskonzept auf die Regleroptimierung eines komplexen, stark gestörten Industrieprozesses mit großen Messtotzeiten.

2 Multikriterielle Optimierung durch Fuzzy Decision Making

2.1 Grundidee

Das in diesem Beitrag vorgestellte Konzept zur Lösung eines multikriteriellen Optimierungsproblems basiert darauf, dass die verschiedenen Teilkriterien durch Fuzzy-Zugehörigkeitsfunktionen dargestellt und unscharf verknüpft werden. In einem ersten Schritt werden konventionelle Gütekriterien für die Optimierungsziele "gutes Störverhalten", "gutes Führungsverhalten", "Stabilität" und "geringer Stellaufwand" definiert. Im zweiten Schritt werden diese dann durch geeignete Transformationen in Fuzzy-Gütekriterien umgewandelt. Die Parameter k_P und k_I eines modellbasierten, linearen Reglers werden schließlich nach einem Algorithmus des Fuzzy Decision Making optimiert.

Im folgenden Abschnitt 2.2 wird zunächst der Algorithmus des Fuzzy Decision Making vorgestellt. Abschnitt 2.3 beschreibt ein einfaches Verfahren zur Gewährleistung einer robusten Lösung.

2.2 Fuzzy Decision Making mit Gewichtung der Gütekriterien

Ansätze des Fuzzy Decision Making sind immer dann sinnvoll, wenn Gütekriterien und Restriktionen als Fuzzy-Zugehörigkeitsfunktionen μ_i darstellbar sind. Sie eignen sich sehr gut bei multikriteriellen Problemstellungen und bieten transparente Mechanismen zur Gewichtung der einzelnen Teilkriterien [4], [5], [6]. Angestrebt wird jeweils ein möglichst hoher Erfüllungsgrad der einzelnen Teilkriterien μ_i.

Zur Herleitung des Algorithmus wird zunächst angenommen, dass nur *ein* Parameter p optimiert werden soll. Es seien N Gütekriterien μ_1 ... μ_N gegeben, die jeweils von einer Hilfsgröße y_i abhängen:

$$\begin{aligned} \mu_1 &= f_1(y_1) \\ &\dots \\ \mu_N &= f_N(y_N) \end{aligned} \tag{1}$$

Die Größen y_i können beispielsweise gewöhnliche Gütekriterien (z.B. Güteintegrale) oder andere, den Prozess charakterisierende Größen sein. Es wird vorausgesetzt, dass die y_i vom zu optimierenden Parameter p abhängen. Somit gilt:

$$\begin{aligned} \mu_1 &= f_1(y_1(p)) \equiv \mu_1(p) \\ &\dots \\ \mu_N &= f_N(y_N(p)) \equiv \mu_N(p) \end{aligned} \tag{2}$$

Es wird die *Fuzzy-Entscheidung* μ_D definiert über eine Fuzzy-UND-Verknüpfung der N Gütekriterien μ_1, ... ,μ_N [6]:

$$\mu_D(p) = \mu_1(p) \wedge \dots \wedge \mu_N(p) \tag{3}$$

Zur Realisierung des Fuzzy-UND-Operators $\wedge$ gibt es viele unterschiedliche Möglichkeiten [6], [7]. In Verbindung mit Gewichtungsfaktoren λ_i zur Bewertung der Gütekriterien hat sich der Minimum-Operator als besonders transparent erwiesen [5]:

$$\mu_D(p) = \min(\lambda_1\mu_1(p),\dots,\lambda_N\mu_N(p)) \tag{4}$$

Dabei wird das Gütekriterium μ_i um so stärker gewichtet, je *kleiner* λ_i ist. Die relative Gewichtung zweier Gütekriterien μ_j und μ_k wird durch das Verhältnis λ_j/λ_k repräsentiert. Für $\lambda_j/\lambda_k \to 0$ wird μ_j sehr viel stärker als μ_k gewichtet.

Der optimale Parameter p^* berechnet sich über die Maximierung von μ_D:

$$\mu_D(p^*) = \max_p \mu_D(p) \tag{5}$$

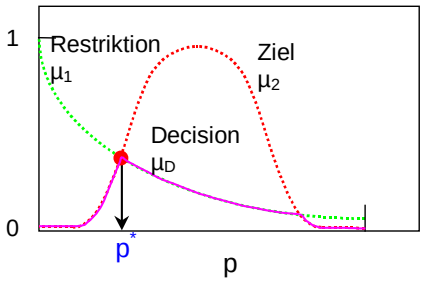


Abb. 1: Zum Grundprinzip des Fuzzy Decision Making

Zur Sicherstellung einer eindeutigen Entscheidung für den Fall, dass ein oder mehrere Gütekriterien ein nicht eindeutiges Maximum aufweisen, wird Gleichung (4) um einen Korrekturterm $\varepsilon(\mu_1 \cdot ... \cdot \mu_N)$ erweitert [5]:

$$\mu_D(p) = \min(\lambda_1\mu_1(p),\ldots,\lambda_N\mu_N(p)) + \underbrace{\varepsilon\mu_1(p)\cdot\ldots\cdot\mu_N(p)}_{Korrekturterm}, \quad 0<\lambda_i<1,\ 0<\varepsilon<<1 \tag{6}$$

Die Wirkung des Korrekturterms wird beispielhaft in Abb. 2 verdeutlicht. Das mit dem Gewichtungsfaktor $\lambda = 0.75$ stärker gewichtete Gütekriterium μ_1 führt zunächst zu einem gleichwertigen Bereich in μ_D, der jedoch durch den Korrekturterm zu einer eindeutigen Entscheidung überführt wird.

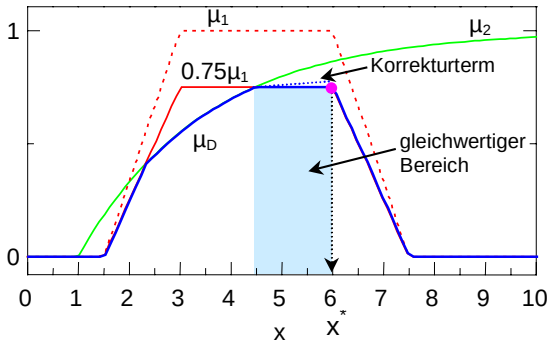


Abb. 2: Zum Mechanismus des Korrekturterms zur Sicherstellung einer eindeutigen Entscheidung

Der Algorithmus lässt sich ohne weiteres auf die hier gegebene Problemstellung der gleichzeitigen Optimierung von zwei Parametern k_P, k_I eines linearen PI-Reglers anwenden.

2.3 Erweiterung hinsichtlich Robustheitsanforderungen

Da bei der Berechnung der Gütekriterien gemäß den Gleichungen (2) i. a. Modellgleichungen verwendet werden, muss die Robustheit der Lösung sichergestellt sein. Dies bedeutet, dass auch bei nicht genau bekannten Streckenparametern das Regelverhalten noch zufriedenstellend ist.

Im Algorithmus des Fuzzy Decision Making kann eine Robustheit der Lösung dadurch gewährleistet werden, dass in die Modellgleichungen (2) Fehlerparameter Δm integriert werden, die die Abweichung des Modells vom realen System beschreiben (z.B. Unsicherheit in der Streckenverstärkung oder Zeitkonstante). Es werden nun charakteristische Werte des Fehlerparameters bei der Berechnung der Gütekriterien eingesetzt und die erhaltenen Zugehörigkeitsfunktionen UND-verknüpft:

$$\begin{aligned} \tilde{\mu}_1(p) &= \bigwedge_{\Delta m} f_1(y_1(\Delta m, p)) \\ &\ldots \\ \tilde{\mu}_N(p) &= \bigwedge_{\Delta m} f_N(y_N(\Delta m, p)) \end{aligned} \tag{7}$$

Die dadurch erhaltenen Gütekriterien $\tilde{\mu}_1,\ldots,\tilde{\mu}_N$ werden zur Berechnung der Fuzzy-Entscheidung μ_D in Gleichung (6) eingesetzt. Die UND-Verknüpfung in (7) kann ggf. mit Gewichtungsfaktoren erfolgen, so dass die Modellabweichungen entsprechend ihrer Bedeutung berücksichtigt werden können.

3 Anwendung auf einen industriellen verfahrenstechnischen Prozess mit großer Messtotzeit und starkem Rauscheinfluss

3.1 Prozessbeschreibung

Der untersuchte Industrieprozess wurde bereits in [9] vorgestellt. Hierbei handelt es sich um ein Verfahren zur Produktion hochwertiger Glasrohre, wobei ein zylinderförmiger Rohling mit einer konstanten Vorschubgeschwindigkeit in einen Ofen geführt, über seine Erweichungstemperatur erhitzt und mit einer regelbaren Geschwindigkeit v als Rohrstrang aus dem Ofen abgezogen wird. In den Rohling wird ein Inertgas mit einem regelbaren Volumendurchsatz eingeleitet, was zum Aufbau eines Differenzdruckes p zwischen dem Inneren der Verformungszone und der Umgebungsluft dient. Die Aufgabe besteht darin, den Durchmesser D und die Wandstärke W des produzierten Rohrstranges über v und p zu regeln.

Dieser Prozess ist durch eine Reihe von Schwierigkeiten gekennzeichnet, wie z.B. nichtlinear gekoppelte Regelgrößen, nichtlineare, zeitvariante Systemdynamik, große Messtotzeiten, stochastische Störungen im stationären Betrieb sowie großen Drift-störungen in den transienten Prozessphasen.

3.2 Prozessmodell

Näherungsweise kann der Prozess durch ein nichtlineares statisches Modell mit linearen dynamischen Gliedern (Wienermodell) beschrieben werden. Im folgenden wird hieraus eine linearisierte Teilstrecke betrachtet, die den stationären Betrieb in einem Arbeitspunkt befriedigend beschreibt. Die Übertragungsfunktionen der Strecke und des Modells sind in Tabelle 1 dargestellt. Die Abweichungen von Übertragungsfaktor k_s und Zeitkonstante T_s der realen Strecke gegenüber dem angenommenen Streckenmodell werden über die Fehlerfaktoren Δk_{rel} und ΔT_{rel} berücksichtigt, welche mit k_s bzw. T_s multipliziert werden. Für $\Delta k_{rel} = \Delta T_{rel} = 1$ ist das Modell mit der Strecke identisch. Da besonders die Zeitkonstante T_s z. T. erheblich variiert, werden die Werte $\Delta k_{rel} = \{0.7, 1, 1.3\}$ und $\Delta T_{rel} = \{0.5, 1, 1.5\}$ in den Untersuchungen verwendet. Es wird also bezüglich der Streckenverstärkung von einer Modellunsicherheit von $\pm$ 30 % und bezüglich der Zeitkonstante von $\pm$ 50 % ausgegangen.

	Übertragungsfunktion	**Parameter**
VZ2-Strecke	$S(s) = \frac{\Delta k_{rel} k_s}{(1 + \Delta T_{rel} T_s s)^2}$	k_s=+1, T_s=32 sec Δk_{rel} ={0.7, 1,1.3}, ΔT_{rel} ={0.5, 1,1.5}
VZ2-Modell	$M(s) = \frac{k_s}{(1 + T_s s)^2}$	
Totzeit	$T(s) = \exp(-T_{tot} s)$	$T_{tot}=L_{tot}/v_0$

Tabelle 1: Streckenmodelle mit Parametern

Übersicht über die verwendeten Reglerstrukturen

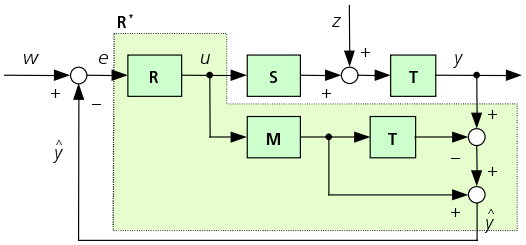


Abb. 3: Smith-Prädiktor mit PI-Grundregler

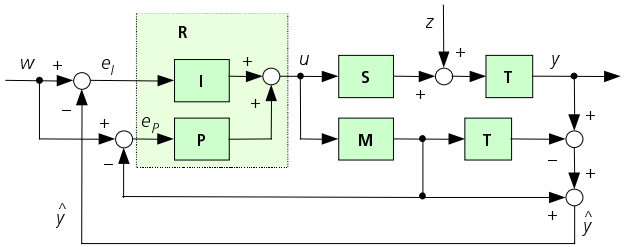


Abb. 4: Smith-Prädiktor-Regelkreis mit selektiven P- und I- Zweigen

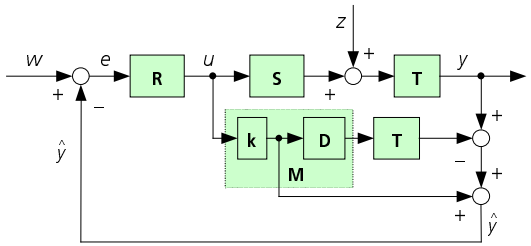


Abb. 5: Smith-Prädiktor-Regelkreis mit gesplittetem Modell

	Smith-Prädiktor mit PI-Grundregler	**Smith-Prädiktor mit selektiven P&I-Zweigen**	**Smith-Prädiktor mit gesplittetem Modell**
Grundregler	$R(s) = k_p(1 + \frac{k_I}{s})$	$P(s) = k_P,\ \ I(s) = \frac{k_I}{s},\ \ R(s) = P(s) + I(s)$	$R(s) = \frac{k_I}{s}$
Ersatzregler R*	$R^* = \frac{u}{e} = \frac{R}{1 + RM(1-T)}$	$R^*_{stör} = \frac{u}{e} = \frac{I}{1 + PM + IM(1-T)}$ $R^*_{Führ} = \frac{u}{w} = \frac{P + I}{1 + PM + IM(1-T)}$	$R^* = \frac{u}{e} = \frac{R}{1 + R(k - MT)}$
Führungs-ÜF F_w	$F_w = \frac{y}{w} = \frac{RST}{1 + RM + R(S-M)T}$	$F_w = \frac{y}{w} = \frac{RST}{1 + RM + I(S-M)T}$	$F_w = \frac{y}{w} = \frac{RST}{1 + Rk + R(S-M)T}$
Stör -ÜF F_z	$F_z = \frac{y}{z} = \frac{1 + RM(1-T)}{1 + RM + R(S-M)T} \cdot T$	$F_z = \frac{y}{z} = \frac{1 + RM - IMT)}{1 + RM + I(S-M)T} \cdot T$	$F_z = \frac{y}{z} = \frac{1 + RM(1-T)}{1 + Rk + R(S-M)T} \cdot T$

Tabelle 2: Untersuchte Reglergrundstrukturen (ÜF = Übertragungsfunktion)

3.3 Regelungskonzept

Das Regelungskonzept für die betrachtete Teilstrecke basiert auf einer Variante der Smith-Prädiktor-Regelkreisstruktur [10]. Das Optimierungsverfahren wird auf die drei in Tabelle 2 dargestellten Strukturvarianten angewandt:

1. Gewöhnlicher Smith-Prädiktor mit linearem Streckenmodell und PI-Regler (Abb. 3)
2. Regelkreis mit selektiven P- und I-Zweigen: Hierbei wird die Tiefpasscharakteristik des I-Anteils ausgenutzt, um die gemessene Regelgröße zu filtern, so dass der Stellaufwand implizit reduziert wird. Der P-Anteil des Reglers wird lediglich zur Gegenkopplung des Modellsignals verwendet (Abb. 4).
3. Smith-Prädiktor mit gesplittetem Modell: In diesem Fall wird der statische Modellanteil rückgekoppelt und der dynamische Modellanteil lediglich zum Abgleich der gemessenen Regelgröße verwendet. Bei diesem Ansatz reduziert sich der Regler auf einen I-Regler (Abb. 5).

3.4 Entwicklung der Fuzzy-Gütekriterien

Zur Optimierung der Reglerparameter k_P, k_I werden Fuzzy-Gütekriterien bezüglich Störverhalten, Führungsverhalten, Stabilität und Stellverhalten berücksichtigt. Dabei werden zunächst gewöhnliche Güteintegrale bzw. Gütefunktionen $J_{stör}$, $J_{führ}$, J_{stab}, J_{stell} definiert und durch eine Transformation in das Intervall $[0, 1]$ in entsprechende Zugehörigkeitsfunktion $\mu_{stör}$, $\mu_{führ}$, μ_{stab}, μ_{stell} überführt. Diese Darstellung hat den Vorteil von großer Transparenz, da die Zugehörigkeitswerte leicht interpretiert werden können. Der Wert $\mu = 1$ entspricht einer sehr guten Beurteilung, der Wert $\mu = 0$ entspricht einem nicht akzeptablem Niveau. Entsprechend sind die Zugehörigkeitsfunktionen zu entwerfen.

3.4.1 Gütekriterium 1: Störverhalten

Um gutes Störverhalten der Regelung zu erreichen muss die Störübertragungsfunktion F_z im relevanten Frequenzbereich möglichst starke Dämpfung aufweisen. Zur Entwicklung des Gütekriteriums wird die Amplitude von F_z im logarithmischen Maßstab betrachtet (dB-Skala). In Abb. 6 ist eine typische Störübertragungsfunktion des Smith-Prädiktors gezeigt.

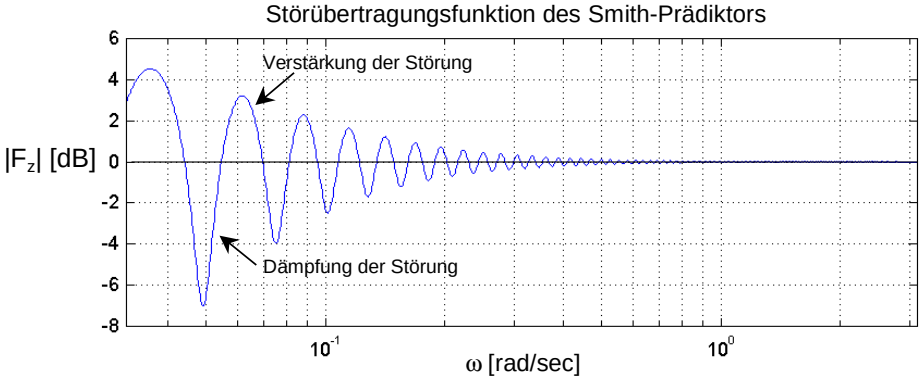


Abb. 6: Typische Störübertragungsfunktion des Smith-Prädiktorreglers

Typisch sind ausgeprägte "Resonanzhörner" im negativen Bereich, d.h. diese Frequenzbereiche werden gut weggedämpft. Zwischen diesen Hörnern gibt es jedoch auch Bereiche mit positiver Amplitude, d.h. Verstärkung der Störung. Wünschenswert ist ein möglichst gleichmäßiger Verlauf von $\log(|F_z|)$ im negativen Bereich. Daher werden positive Werte von $\log(|F_z|)$ im Gütekriterium $J_{stör}$ durch eine Quadrierung besonders bestraft (Gleichung (8)). Zusätzlich wird ein weiterer Term zur Bestrafung starker Krümmung der Störübertragungsfunktion $|F_z|$ addiert. Dadurch soll ein gleichmäßiger Verlauf von $|F_z|$ über den Frequenzbereich erreicht werden. Somit wird folgender zu minimierende Güteindex $J_{stör}$ definiert:

$$J_{stör} = \int_{\omega_l}^{\omega_g} \left(\log(|F_z(\omega)|)_{|F_z|<1} + \underbrace{\log(|F_z(\omega)|)^2_{|F_z|>1}}_{\text{Straftermfür Werte von |Fz|>1}} + \alpha \cdot \underbrace{\left| \frac{\partial^2 |F_z(\omega)|}{\partial \omega^2} \right|}_{\text{Straftermfür Krümmung von Fz}} \right) d\omega \qquad (8)$$

Die Integrationsgrenzen ω_l und ω_g entsprechen über $\omega = 2\pi/T$ den Periodendauern $T_1 = 100 \cdot T_A = 200$ sec und $T_g = 2 \cdot T_A = 4$ sec (ω_g: Grenzfrequenz nach Shannon'schen Abtasttheorem). Somit wurde mit dem Intervall $[\omega_l, \omega_g]$ der für den Prozess relevante Frequenzbereich ausgewählt. Die Integration wird numerisch durchgeführt.

Anschaulich kann $J_{stör}$ interpretiert werden als ein Maß für die Standardabweichung des Ausgangssignals y. Das ergibt sich aus folgender Überlegung. Es gilt für die Standard-abweichung σ_y

$$\sigma_y^2 = \Theta_{yy}(0) - \bar{y}^2 \qquad (9)$$

Dabei bezeichnen $\Theta_{yy}(\tau)$ die Autokorrelationsfunktion und $\bar{y}$ den Mittelwert von y. $\Theta_{yy}(0)$ berechnet sich zu

$$\Theta_{yy}(0) = \frac{1}{\pi} \int_0^\infty |F_z(\omega)|^2 \, S_{zz}(\omega) d\omega, \qquad (10)$$

wobei $S_{zz}(\omega)$ das Leistungsdichtespektrum des Rauschsignals bezeichnet [11]. Für weißes Rauschen ($S_{zz}(\omega)$=const.) ergibt sich zumindest näherungsweise ein Ausdruck vom Typ (8). Für stark negative Werte von $J_{stör}$ kann das Störverhalten als optimal ($\mu_{stör} = 1$), für stark positive Werte als nicht akzeptabel ($\mu_{stör} = 0$) beurteilt werden. Entsprechend wird folgende Definition vorgeschlagen (vgl. Abb. 7):

$$\mu_{stör}(J_{stör}) = \min(1, \exp(-\frac{J_{stör} - J_1}{J_2})) \qquad (11)$$

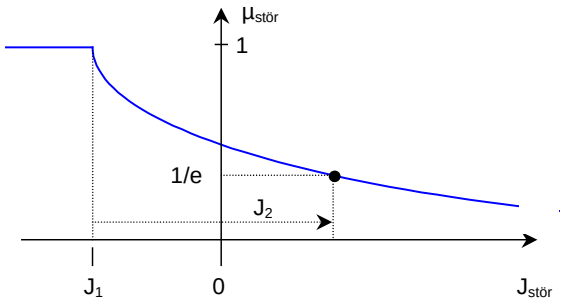


Abb. 7: Fuzzy-Gütekriterium bezüglich Störverhalten ($\mu_{stör}$)

Die Parameter J_1, J_2 wurden zu $J_1 = -3$ und $J_2 = 6$ gewählt. Die Parameter sind relativ leicht zu interpretieren, da $J_{stör}$ in etwa den Mittelwert der Störübertragungsfunktion $|F_z|$ in dB-Einheiten darstellt (vgl. Gleichung (8)).

3.4.2 Gütekriterium 2: Führungsverhalten

Um stationäre Genauigkeit zu erhalten, muss der Betrag der Führungsübertragungsfunktion F_w für kleine Frequenzen etwa gleich 1 sein, d.h. im logarithmischen Maßstab gleich 0. Werte $\log(|F_w|) > 0$ entsprechen überschwingendem Verhalten und sind ebenso ungünstig wie Werte $\log(|F_w|) < 0$, d.h. langsames Erreichen des stationären Zustandes. Es wird daher folgender zu minimierende Teilindex $J_{führ}$ als Mittelwert von F_w bei kleinen Frequenzen definiert:

$$J_{führ} = \frac{1}{\omega_2 - \omega_1} \int_{\omega_1}^{\omega_2} |\log(|F_w(\omega)|)| d\omega, \qquad \omega_2 \approx \omega_1 \tag{12}$$

Bei der rechentechnischen Implementierung wurde $\omega_1 = 0.031$ rad/sec und $\omega_2 = 0.055$ rad/sec gewählt. Dies entspricht den Periodendauern $T_1 = 100 \cdot T_A = 200$ sec und $T_2 = 97.5 \cdot T_A = 115$ sec.

Das Führungsverhalten kann als optimal bezeichnet werden, wenn das definierte Gütekriterium $J_{führ}$ (=Mittelwert von $|F_w|$ in dB-Einheiten) nahe bei Null liegt. Da in $J_{führ}$ sowohl positive als auch negative Werte von $\log(|F_w|)$ zu positiven Werten führen (vgl. (13)), ist die Definition folgender Zugehörigkeitsfunktion sinnvoll

$$\mu_{führ}(J_{führ}) = \exp(-\frac{J_{führ}}{J_1}), \tag{13}$$

wobei der Parameter J_1 zu $J_1 = 5$ dB gewählt wurde.

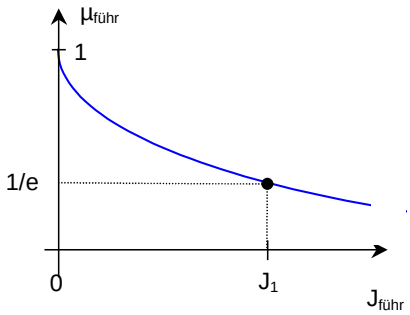


Abb. 8: Fuzzy-Gütekriterium bezüglich Führungsverhalten ($\mu_{führ}$)

3.4.3 Gütekriterium 3: Stabilität

Zur Beurteilung der Stabilität ist die Lage der Pole des geschlossenen Regelkreises wesentlich (vgl. Übertragungsfunktionen in Tabelle 2). Befinden sich alle Pole in der negativen Halbebene (Realteil $Re(s) < 0$), so ist der Regelkreis stabil. Falls ein oder mehrere Pole in der rechten Halbebene ($Re(s) > 0$), so ist der Regelkreis instabil. Falls ein Pol auf der Imaginärachse liegt und alle anderen sich in der linken Halbebene befinden, ist der Regelkreis grenzstabil [8]. Zur Bewertung der Stabilität wird daher die Lage des Pols mit dem größten Realteil herangezogen ($\max Re(s)$). Es erscheint als sinnvoll, einen optimalen Wert $Re_{opt} < 0$ zu definieren mit $\mu_{stab} = 1$ für

$\max Re(s) < Re_{opt}$. Weiter ist es plausibel, für den Fall, dass $\max Re(s) > 0$ gilt, $\mu_{stab} = 0$ zu setzen. Im Zwischenbereich wird linear interpoliert. Dadurch ergibt sich folgende Definition der Zugehörigkeitsfunktion μ_{stab}:

$$\mu_{stab} = \begin{cases} 1 & \text{für } \max Re(s) < Re_{opt} < 0 \\ \dfrac{\max Re(s)}{Re_{opt}} & \text{für } Re_{opt} < \max Re(s) < 0 \\ 0 & \text{für } 0 < \max Re(s) \end{cases} \tag{14}$$

Der optimale Wert $Re_{opt} < 0$ wurde entsprechend der erreichbaren Stabilität bei dem VZ2-Modell zu $Re_{opt} = -0.05$ gewählt.

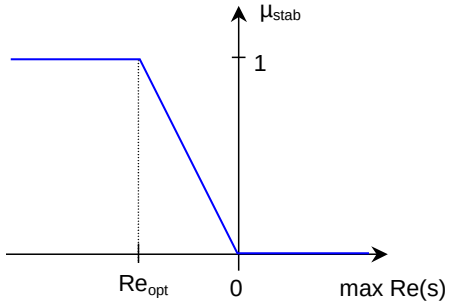


Abb. 9: Fuzzy-Gütekriterium bezüglich Stabilität (μ_{stab})

Von entscheidender Bedeutung ist, dass die Zugehörigkeitsfunktion μ_{stab} im Gegensatz zu $\mu_{stör}$ und $\mu_{führ}$ den Wert Null annehmen kann. Wegen der UND-Verknüpfung der Teilkriterien (vgl. (6)) werden die entsprechenden Lösungswerte k_P und k_I ausgeschlossen, die wegen $\max Re(s)$ zu instabilen Lösungen führen würden. Dieser transparente Mechanismus ist ein wesentlicher Vorteil des hier vorgestellten Ansatzes.

3.4.4 Gütekriterium 4: Stellverhalten

Bezüglich des Stellverhaltens sind möglichst geringe Stelleingriffe anzustreben. Da beim vorliegenden stark gestörten Glasziehprozess in der stationären Phase das Fehlersignal $e = w - \hat{y}$ überwiegend stochastischer Natur ist, besteht das Ziel darin, die Standardabweichung von u zu Minimieren. Entsprechend der Überlegung bei der Störoptimierung wird dies über die Minimierung des Integrals über die Ersatzreglerübertragungsfunktion

$$R^*(s) = \frac{u(s)}{e(s)} \tag{15}$$

erreicht (vgl. Störübertragungsfunktionen in Tabelle 2). Es wird daher folgender zu minimierende Güteindex J_u definiert:

$$J_{stell} = \int_{\omega_l}^{\omega_2} \log(|R^*(\omega)|) d\omega \tag{16}$$

Somit werden also im Mittel möglichst kleine Werte von $\log(|R^*(\omega)|)$ angestrebt, was einem geringem Stellaufwand entspricht.

Das Fuzzy-Gütekriterium bezüglich Stellverhalten (μ_{stell}) wird in vollkommen analoger Weise wie das Gütekriterium bezüglich Störverhalten ($\mu_{stör}$) definiert, da auch hier für stark negative Werte von J_{stell} das Stellverhalten als optimal ($\mu_{stell} = 1$), für stark positive Werte als nicht akzeptabel ($\mu_{stell} = 0$) beurteilt werden kann. Entsprechend Gleichung (11) wird folgende Definition vorgeschlagen:

$$\mu_{stell}(J_{stör}) = \min(1, \exp(-\frac{J_{stell} - J_1}{J_2})) \tag{17}$$

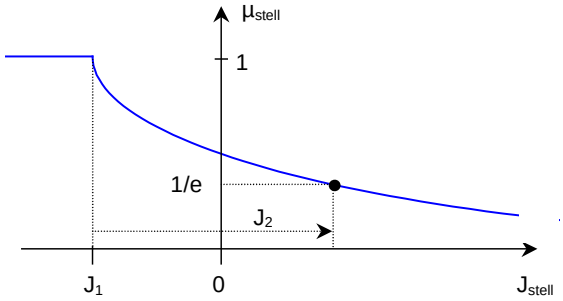


Abb. 10: Fuzzy-Gütekriterium bezüglich Stellverhalten (μ_{stell})

Die Parameter J_1, J_2 wurden zu $J_1 = -5$ und $J_2 = 5$ gewählt. Die Parameter entziehen sich hier allerdings einer einfachen Interpretation. Es ist daher zunächst der Wertebereich von J_{stell} festzustellen und schließlich J_1 und J_2 entsprechend der erreichbaren Güte festzulegen.

3.5 Robustheit gegenüber Modellungenauigkeiten

Die verwendeten Reglerstrukturen erbringen naturgemäß die besten Ergebnisse für den Fall der idealen Übereinstimmung des Streckenmodells M mit der tatsächlichen Strecke S ($S = M$). Es gilt beispielsweise für die Führungsübertragungsfunktion des Smith-Prädiktorregelkreises mit PI-Grundregler (siehe Tabelle 2):

$$F_w = \frac{y}{w} = \frac{RST}{1 + RM + R(S - M)T} \tag{18}$$

Die Abweichung des Modells von der Strecke wird offensichtlich durch den Term $(S - M)$ beschrieben. Nur für $S = M$, d.h. ideale Übereinstimmung des Modells mit dem Streckenverhalten, wird die Totzeit der Strecke kompensiert.

Die Abweichung vom Modell gegenüber dem Streckenverhalten werden über relative Fehlerparameter Δk_{rel}, ΔT_{rel} modelliert (vgl. Abschnitt 3.2). Es gilt für die VZ2-Strecke

$$S(s) = \frac{\Delta k_{rel} k_s}{(1 + \Delta T_{rel} T_s s)^2}. \tag{19}$$

Mit S sind auch die Übertragungsfunktionen F_z, F_w, F_u von Δk_{rel} und ΔT_{rel} abhängig und dadurch auch die Gütekriterien $\mu_{stör}$, $\mu_{führ}$, μ_{stab}, μ_{stell}.

Zur Berechnung von Fuzzy-Gütekriterien, die auch die möglichen Modellungenauigkeiten enthalten, werden die einzelnen Gütekriterien $\mu_{stör}$, $\mu_{führ}$, μ_{stab}, μ_{stell} in Abhängigkeit von Δk_{rel} und ΔT_{rel} berechnet und jeweils durch den Fuzzy-UND-Operator miteinander verknüpft (20).

$$
\begin{aligned}
\tilde{\mu}_{stör} &= \underset{\Delta k_{rel},\Delta T_{rel}}{\wedge} \mu_{stör}(\Delta k_{rel},\Delta T_{rel}), \qquad \wedge = \min \\
\tilde{\mu}_{führ} &= \underset{\Delta k_{rel},\Delta T_{rel}}{\wedge} \mu_{führ}(\Delta k_{rel},\Delta T_{rel}) \\
\tilde{\mu}_{stab} &= \underset{\Delta k_{rel},\Delta T_{rel}}{\wedge} \mu_{stab}(\Delta k_{rel},\Delta T_{rel}) \\
\tilde{\mu}_{stell} &= \underset{\Delta k_{rel},\Delta T_{rel}}{\wedge} \mu_{stell}(\Delta k_{rel},\Delta T_{rel})
\end{aligned}
\tag{20}
$$

Die Aggregation der somit erhaltenen Teilkriterien $\tilde{\mu}_{stör}$, $\tilde{\mu}_{führ}$, $\tilde{\mu}_{ab}$, $\tilde{\mu}_{ell}$ erfolgt entsprechend Gleichung (6) gemäß

$$
\mu_D(K_P,K_I) = \min(\lambda_{stör}\tilde{\mu}_{stör},\lambda_{führ}\tilde{\mu}_{führ},\lambda_{stab}\tilde{\mu}_{stab},\lambda_{stell}\tilde{\mu}_{stell}) + \varepsilon\tilde{\mu}_{stör}\tilde{\mu}_{führ}\tilde{\mu}_{stab}\tilde{\mu}_{stell} \tag{21}
$$

4 Ergebnisse der Parameteroptimierung

Zur Veranschaulichung des vorgestellten Optimierungskonzeptes werden im folgenden die Ergebnisse zu Regelkreisstruktur 2 (Smith-Prädiktor mit selektiven P- und I-Zweigen, s. Abb.4) betrachtet.

In Abb. 11 sind die Fuzzy-Gütekriterien bezüglich Störverhalten $\mu_{stör}$, Führungsverhalten $\mu_{führ}$, Stellaufwand μ_{stell} und Stabilität μ_{stab} sowie die resultierende Fuzzy-Entscheidung μ_D in Abhängigkeit der zu optimierenden Parameter k_P und k_I dargestellt. Das Ergebnis wurde für T_{tot} = 235 sec erzielt. Aus den unterschiedlichen Positionen der Maxima der einzelnen Gütekriterien wird deutlich, dass gegensätzliche Güteanforderungen vorliegen. So wird gutes Störverhalten eher für "kleine" k_I-Werte erreicht, während für gutes Führungsverhalten "große" k_I-Werte günstig sind. Beim Gütekriterium zur Stabilität μ_{stab} ist zu erkennen, dass für große Bereiche des [k_P, k_I]-Parameterraums sich instabiles Verhalten ergeben würde, was durch Bereiche mit $\mu_{stab} = 0$ gekennzeichnet ist. Durch die UND-Verknüpfung der Gütekriterien wird sichergestellt, dass diese Werte nicht zur Lösungsmenge gehören können.

Kennlinien der optimierten Parameter k^*_P, k^*_I sowie der Zugehörigkeitsgrad $\mu_D(k^*_P, k^*_I)$ sind in Abb. 12 als Funktion der Messtotzeit T_{tot} dargestellt, die in Abhängigkeit der Abzuggeschwindigkeit v variiert. Bei dickwandigen Rohren treten sehr langsame Abzuggeschwindigkeiten und somit große Totzeiten auf, bei dünnwandigen Rohren sind die Totzeiten bis zu einem Faktor von 10 geringer. Durch Hilfslinien sind die konstanten Parameterwerte vor der Optimierung gekennzeichnet. Es ist ersichtlich, dass der Zugehörigkeitsgrad μ_D, der als Gütemaß der optimierten Werte interpretiert werden kann, durch die Optimierung deutlich gesteigert wurde.

Interessant ist die Tatsache, dass für kleine Totzeiten das Gütemaß abnimmt. Dies ist dadurch zu erklären, dass für Totzeiten in der Größenordnung der Zeitkonstante ($T_{tot} \approx T_S$ = 32 sec) der hier berücksichtigte relative Modellfehler in T_S von $\pm$ 50 % sich stärker bemerkbar macht als für $T_{tot} > T_S$. Für den Fall sehr großer Totzeiten ($T_{tot} >> T_S$) nimmt die Robustheit des Smith-Prädiktors hingegen auch ab, so dass das Gütemaß hier ebenfalls niedrigere Werte annimmt.

Die Ergebnisse dieser Optimierung wurde in Form von Lookup-Tabellen in der Prozessregelungssoftware implementiert und seither mit Erfolg eingesetzt. Hierbei wurden Verbesserungen in der Regelgüte von bis zu 30% gegenüber heuristisch optimierten Regelparametern erzielt. Gegenüber den herkömmlichen vektoriellen Gütekriterien

gestattet das vorgestellte fuzzy-basierte Regleroptimierungskonzept eine individuellere Berücksichtigung in der Praxis auftretender, heuristischer Güte- und Robustheits-forderungen sowie Beschränkungen.

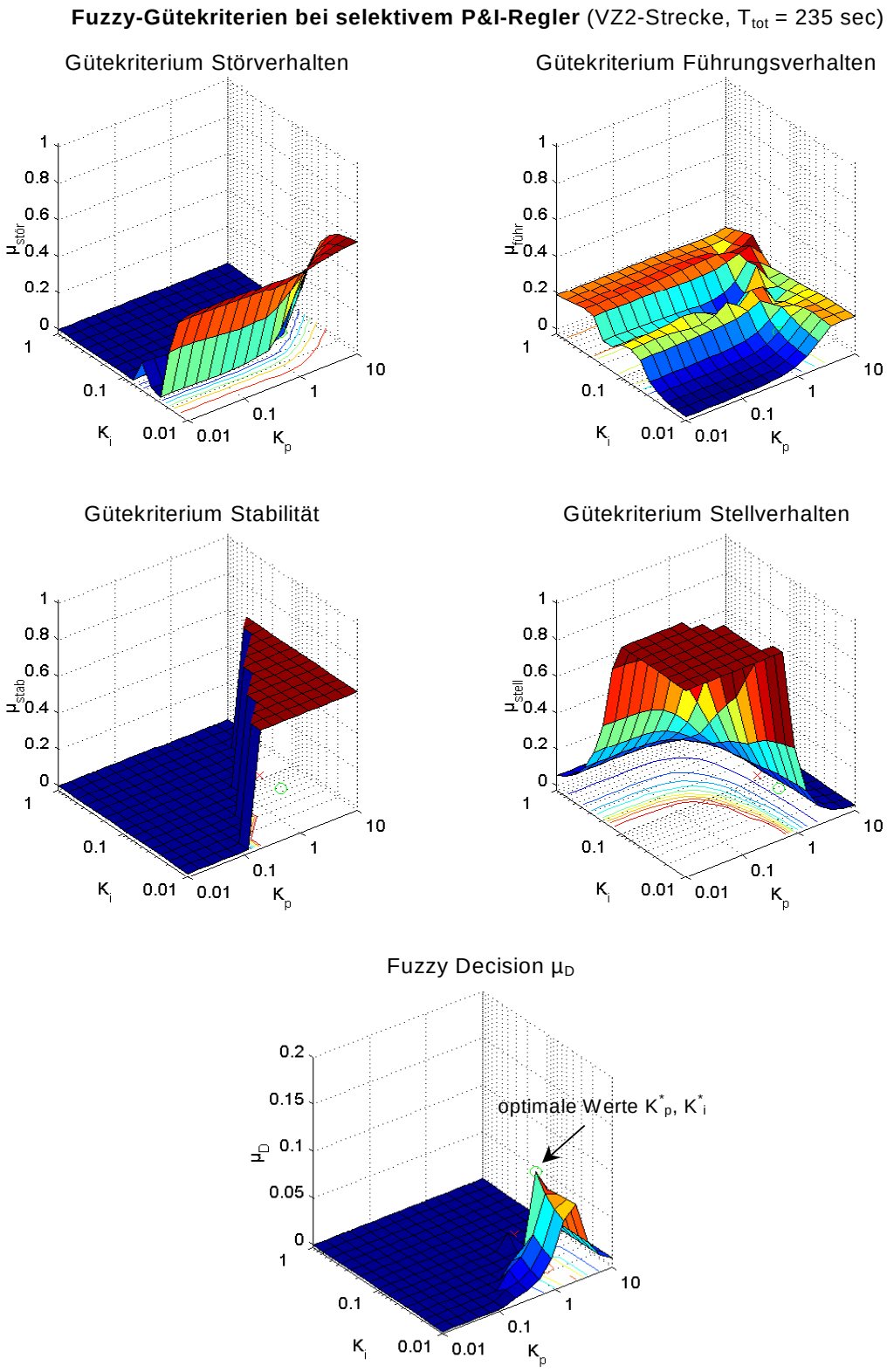


Abb. 11: Fuzzy-Gütekriterien bei VZ2-Strecke (T_{tot} = 235 sec)

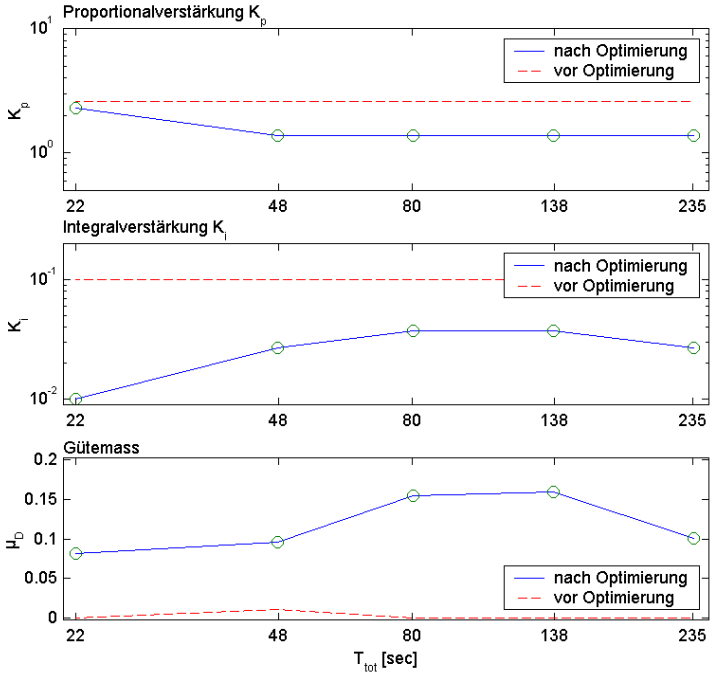


Abb. 12: Optimierte Parameter k_P und k_I sowie resultierendes Gütemaß μ_D für eine VZ2-Strecke als Funktion der Messtotzeit

5 Literatur

[1] Kreisselmeier, G; Steinhauser, R.: Systematische Auslegung von Reglern durch Optimierung eines vektoriellen Gütekriteriums, Regelungstechnik 27, pp. 76-79, 1979

[2] Ackermann, J.: Robuste Regelung. Springer, Heidelberg 1993

[3] Kahlert, J.: Vektorielle Optimierung mit Evolutionsstrategien und Anwendungen in der Regelungstechnik. VDI-Verlag, Düsseldorf, 1991

[4] Bellman, R.E.; Zadeh, L.A.: Decision Making In A Fuzzy Environment, *Management Science*, 17 (1970), S. 141-163

[5] Bernard, T.: Ein Beitrag zur gewichteten multikriteriellen Optimierung von Heizungs- und Lüftungsregelkreisen auf Grundlage des Fuzzy Decision Making. Universität Karlsruhe, Fakultät für Maschinenbau, Dissertation 2000, online: http://thbernard.leute.server.de/diss

[6] Rommelfanger, H.: Fuzzy Decision Support Systeme. Springer-Verlag, Heidelberg 1994

[7] Kiendl, H.: Fuzzy Control methodenorientiert. Oldenbourg-Verlag, München; 1997

[8] Föllinger, O.: Regelungstechnik: Einführung in die Methoden und ihre Anwendung, Hüthig-Verlag, Heidelberg 1994

[9] Sajidman, M.; Kuntze, H.B.: Fuzzy-Regelung stark gestörter, verfahrens-technischer Prozesse mit großer Messtotzeit; Proc. 5. Workshop "Fuzzy Control" des GMA-UA 1.4.2, Dortmund 1995

[10] Smith, O. J.: A Controller to Overcome Deadtime, *ISA Journal*, 6 (1959) 2, pp. 28-33

[11] Isermann, R.: Digitale Regelsysteme. Berlin, Springer, 1977

FUZZY-ADAPTIERTE MODELLGESTÜTZTE MESSVERFAHREN

A. Traichel, W. Kästner, R. Hampel

Institut für Prozeßtechnik, Prozeßautomatisierung und Meßtechnik
FG Meßtechnik/Prozeßautomatisierung
an der Hochschule Zittau/Görlitz (FH)
D-02763 Zittau, Theodor-Körner-Allee 16
Tel: +49-(0)3583-61-1547, Fax: +49-(0)3583-61-1288
E-mail: r.hampel@hs-zigr.de

1 Einführung

Für eine kontiniuerliche Prozeßüberwachung und –beobachtung in sicherheitsrelevanten Systemen ist die Bereitstellung des aktuellen Prozeßzustandes in Form von meßbaren und nichtmeßbaren Zustandsgrößen erforderlich. Dies ist besonders erschwert in komplexen Prozessen, die durch starke dynamische Nichtlinearitäten und damit Unsicherheiten im Prozeßverlauf infolge hochtransienter Übergangsvorgänge gekennzeichnet sind.

Es existieren häufig analytisch schwer beschreibbare Unschärfen durch die Vielzahl der auftretenden Abhängigkeiten, der Gesamtheit aller möglichen Abläufe und Einzeleffekte (Fuzziness von komplexen Systemen). Trotzdem laufen die Prozesse in einem reproduzierbaren technologischen Rahmen ab. Aufgrund dessen kann zur Beschreibung auch Erfahrungs- bzw. Expertenwissen in geeigneter, aufbereiteter Form eingesetzt werden.

Die höheren Anforderungen an die Sicherheit, Zuverlässigkeit (Verfügbarkeit) und Güte der Prozeßinformationen erfordern den Einsatz moderner, intelligenter Methoden der Signalverarbeitung, wie die Fuzzy Set Theorie oder Künstliche Neuronale Netze. Dieser Aspekt wird auch besonders in [16] hervorgehoben. Ausgangspunkt des vorliegenden Beitrages ist die Annahme, daß durch die Kombination analytischer und wissensbasierter Verfahren eine neue Qualität zur Prozeßbeobachtung und -überwachung erreicht werden kann. Als analytisches Verfahren kommen klassische modellgestützte Meßverfahren -MMV- in Form von Nichtlinearen Beobachtern zur Anwendung. Die Qualität und Gültigkeit der eingesetzten klassischen Beobachter ist limitiert durch [13]:

⇒ die starken Nichtlinearitäten des Prozesses und damit des Prozeßmodells,
⇒ Prozeßstörungen,
⇒ stochastischen Störungen der Meßgrößen
⇒ Anwendbarkeit auf bestimmte Klassen von Systemen.

Das globale Ziel besteht darin, die Fuzzy Set Theorie zur Modellierung der Unsicherheiten und Unschärfen bei der Prozeßbeschreibung innerhalb der MMV einzusetzen. Im Ergebnis entsteht aus einer Kombination aus analytischem Modellansatz (DGL) und Fuzzy Modell eine hybride Struktur. Als hybride Strukturen werden hier solche bezeichnet, in denen zur Repräsentation des Wissens verschiedene Formalismen zur Verfügung stehen [9]. In Bild 1.1 sind die Varianten der entwickelten, fortgeschrittenen Beobachter als Hybridverfahren mit fuzzy-basierter Parameteradaption dargestellt. Zunächst wurde, basierend auf einem linearen Prozeßmodell, ein fuzzy-

unterstützter Beobachter zur Modellierung der Nichtlinaritäten des Prozesses entworfen. In diesem Fall wurden die Modellmatrizen mit Fuzzy Modellen nachgeführt. Auf diesem Wege wurde eine verbesserte Rekonstruktionsgüte der Zustandsgrößen des Prozesses erzielt [13].

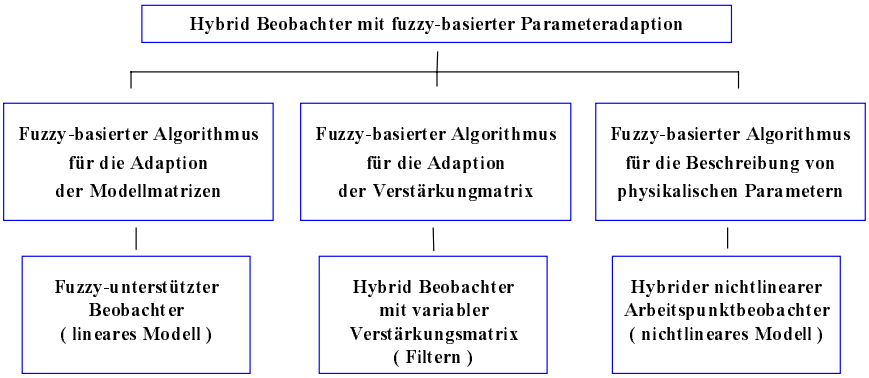


Bild 1.1: Varianten der fuzzy-basierten Parameteradaption in Modellgestützten Meßverfahren (Beobachter)

Um die stochastischen Störungen der Meßgrößen zu kompensieren, wurde ein Hybrid Beobachter mit variabler Verstärkungsmatrix entworfen, in dem eine fuzzy-basierte Adaption der Verstärkungsmatrix in Abhängigkeit der stochastischen Eigenschaften des Rekonstruktionsfehlers auf der Basis statistischer Kenngrößen vorgenommen wird. Mit diesem Hybrid Beobachter werden die Beobachter Problematiken Konvergenz und Filterung mit hoher Güte gelöst [13]. Gegenstand dieses Artikels ist die Beschreibung des Entwurfes des Hybriden nichtlinearen Arbeitspunkt-Beobachters, der durch ein nichtlineares Prozeßmodell und eine fuzzy-basierte Beschreibung der physikalischen Größe Wärmestrom gekennzeichnet ist [3]. Diese Art von Hybrid Beobachtern ist besonders geeignet für die Beschreibung von Prozessen mit Prozeßstörungen und inhärenten Rückkopplungen. Für die Nachbildung des hochtransienten Verlaufs einer Prozeßstörung (Leck) wird eine neue Methode angewendet, ein Fuzzy Modell mit externer Dynamik, welches sich durch eine hohe Transparenz auszeichnet.

Für die genannten Hybdrid Beobachter werden in [7], [8], [10], [11], [12], [14] und [15] vergleichbare Vorgehensweisen für Fuzzy Modelle (Fuzzy Controller) und Neuronale Netze vorgeschlagen. Eine Fuzzy Struktur mit interner Dynamik zur Beschreibung dynamischer Prozesse wird in [4] dargestellt.

2 Beschreibung des nichtlinearen Prozesses – Füllstandsmessung im Druckbehälter

Die Intention des Einsatzes nichtlinearer Modellgestützter Meßverfahren (Beobachter) ist im vorliegenden Fall die Rekonstruktion der meßbaren und nichtmeßbaren Massenfüllstände in definierten Zonen eines Druckbehälters mit Wasser-Dampf Gemisch während negativer Drucktransienten. Bild 2.1 zeigt einen Druckbehälter (z. B. Druckhalter, Dampferzeuger, Reaktordruckgefäß) mit dem entsprechenden hydrostatischen Füllstandsmeßsystem. Mit Hilfe des hydrostatischen Differenzdruckmeßverfahrens wird der Massenfüllstand zwischen den Einbindungen (obere und untere Einbindung am Druckbehälter) bestimmt.

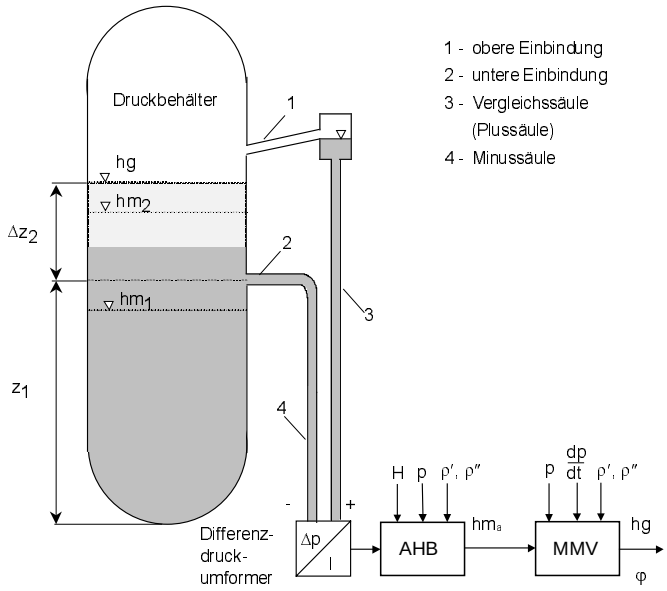


Bild 2.1: Bestimmung des Massenfüllstandes in Druckbehältern

Das Volumen des Wasser-Dampf Gemisches im Druckbehälter (charakterisiert durch den Gemischspiegel hg) läßt sich im einfachsten Fall in zwei Zonen unterteilen:

Zone 1: Druckbehältervolumen unterhalb der unteren Einbindung des Meßsystems (konstante Zonenhöhe z_1)

Zone 2: Volumen des Wasser-Dampf Gemisches zwischen den Einbindungen des Meßsystems (variable Zonenhöhe Δz_2)

Für die interessierenden Massenfüllstände hm

und hm_2 der Zonen 1 und 2 gelten ausgehend von den Massen- und Energiebilanzen des Wasser-Dampf Gemisches die folgenden nichtlinearen Differentialgleichungen (DGL):

$$\frac{dhm_1}{dt} = k_w - k_1 \cdot hm_1 - \left(2k_2 + k_3\right) \cdot hm_1 \cdot \frac{dp}{dt} + k_2 \cdot 2z_1 \cdot \frac{dp}{dt} - k_4 \cdot \dot{Q}_1 \qquad (1)$$

$$\frac{dhm_2}{dt} = -k_w + k_1 \cdot hm_1 + \left(k_6 + k_7 + k_2\right) \cdot hm_1 \cdot \frac{dp}{dt} - k_2 \cdot 2z_1 \cdot \frac{dp}{dt} - k_8 \cdot hm_2 \cdot \frac{dp}{dt} + \\ - k_5 \cdot \dot{Q}_2 + \left(k_4 - k_5\right) \cdot \dot{Q}_1 \qquad (2)$$

hm_1	-	Massenfüllstand Zone 1, nichtmeßbare Zustandsgröße
hm_2	-	Massenfüllstand Zone 2, meßbare Zustandsgröße (Ausgangsgröße)
p, dp / dt	-	Druck, Druckänderungsgeschwindigkeit, meßbare Eingangsgrößen
$\dot{Q}_1, \dot{Q}_2$	-	Wärmestrom zwischen Behälterwand und Wasser-Dampf Gemisch in den Zonen 1 bzw. 2, nichtmeßbare Eingangsgrößen
k	-	nichtlineare Koeffizienten (Abhängigkeit vom Druck)

Aufgrund der Anwendung von nichtlinearen Beobachtern wurde für das zweidimensionale nichtlineare Modell entsprechend den Gleichungen (1) und (2), der in den Gleichungen (3) bis (5) dargestellte Ansatz für einen nichtlinearen PI-Arbeitspunkt-Beobachter entworfen, der aus dem nichtlinearen Streckenmodell und einer linearen Rückführung des Rekonstruktionsfehlers mit einer konstanten Verstärkung besteht [2]:

$$\dot{\hat{x}}_1 = k_w - k_1 \cdot \hat{x}_1 - k_a \cdot \hat{x}_1 \cdot u_1 + k_b \cdot u_1 - k_4 \cdot u_2 + K_{0P} \cdot (y - \hat{x}_2) + K_{0I} \cdot \int (y - \hat{x}_2) dt \quad (3)$$

$$\dot{\hat{x}}_2 = -k_w + k_1 \cdot \hat{x}_1 + k_c \cdot \hat{x}_1 \cdot u_1 - k_b \cdot u_1 - k_8 \cdot \hat{x}_2 \cdot u_1 - k_5 \cdot u_3 + k_d \cdot u_2 + \\ + K_{1P} \cdot (y - \hat{x}_2) + K_{1I} \cdot \int (y - \hat{x}_2) dt \quad (4)$$

$$\hat{y} = \hat{x}_2 \quad (5)$$

Die Eingangs-, Ausgangs- und Zustandsgrößen des Beobachtermodells werden den physikalischen Größen entsprechend Beziehung (6) zugeordnet (vgl. Gleichungen (1) und (2)):

$$\begin{array}{llll} \hat{x}_1 \mathrel{\hat{=}} hm_1 & u_1 \mathrel{\hat{=}} \dfrac{dp}{dt} & u_2 \mathrel{\hat{=}} \dot{Q}_1 & \mathbf{K}_P \mathrel{\hat{=}} \text{proportionale Verstärkung} \\ \hat{x}_2 \mathrel{\hat{=}} hm_2 & & u_3 \mathrel{\hat{=}} \dot{Q}_2 & \mathbf{K}_I \mathrel{\hat{=}} \text{int egrale Verstärkung} \end{array} \quad (6)$$

Während eines Druckentlastungsvorganges (z. B. Leck) wird dem Wasser-Dampf Gemisch durch die Behälterwand Wärmeenergie in Form der Wärmestrome $\dot{Q}_1$ und $\dot{Q}_2$ zugeführt. Die starke Druckänderung und die hohe Druckänderungsgeschwindigkeit führen zu einer Differenz zwischen der Wandtemperatur des Behälters und der Fluidtemperatur. Dieser Effekt hat Einfluß auf die Energiebilanz und somit auf die Gleichungen (1) und (2).

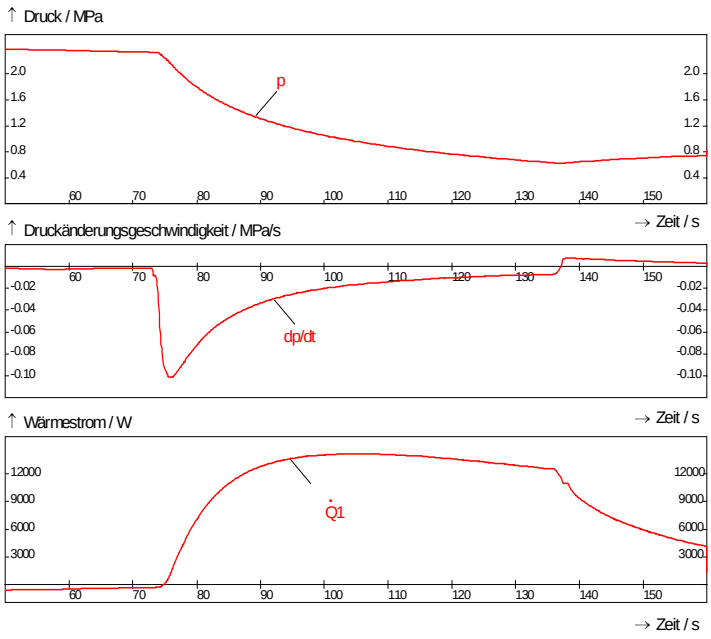


Bild 2.2: Zeitliche Verläufe der Größen Druck (p), Druckänderungsgeschwindigkeit (dp/dt) und Wärmestrom der Zone 1 ($\dot{Q}_1$) für ein Druckentlastungs-experiment ($p_0 = 2.4$ MPa)

Bild 2.2 veranschaulicht den zeitlichen Verlauf des Wärmestromes $\dot{Q}_1$ der Zone 1 für ein Druckentlastungsexperiment an der Versuchsanlage Druckhaltermodell. Weiterhin werden die Größen Druck (p) und Druckänderungsgeschwindigkeit (dp/dt) dargestellt. Die Nachrechnung des Experimentes und die Bereitstellung der zeitlichen Verläufe der Prozeßgrößen erfolgte mit Hilfe des komplexen Thermohydraulikcodes ATHLET. Die Wärmeströme $\dot{Q}_1$ und $\dot{Q}_2$ der Zonen 1 und 2 stehen dem Beobachter als meßbare

Eingangsgrößen u_2 und u_3 nicht zur Verfügung. Eine analytische Bestimmung des Wärmestromes setzt die Kenntnis des Wärmeübertragungsmodells voraus. Hierin ist die nichtmeßbare Zustandsgröße Temperatur der Behälterwand zu berechnen. Der Wärmeübergangskoeffizient zwischen Behälterwand und Wasser-Dampf Gemisch muß bekannt sein oder aus empirischen oder halbempirischen Gleichungen ermittelt werden, welche für bestimmte Randbedingungen (z. B. Geometrie, Strömungsform) gelten.

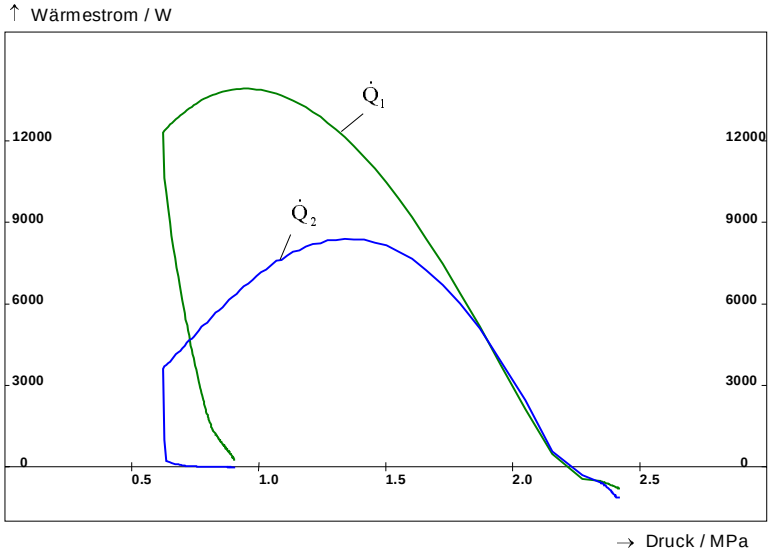


Bild 2.3: Nichtlineare Charakteristik der Wärmeströme $\dot{Q}_1$, $\dot{Q}_2$ in Abhängigkeit des Druckes p für ein Druckentlastungsexperiment (p_0 = 2.4 MPa)

Zur Veranschaulichung der nichtlinearen Charakteristik des Prozesses werden im Bild 2.3 die Wärmeströme $\dot{Q}_1$, $\dot{Q}_2$ in Abhängigkeit des Druckes p für ein Druckentlastungsexperiment (p_0 = 2.4 MPa) dargestellt.

3 Zustandsrekonstruktion mittels des hybriden nichtlinearen Arbeitspunkt-Beobachters

Da die unbekannten Wärmeströme einen wesentlichen Einfluß auf die Rekonstruktionsgüte der Massenfüllstände hm_1 und hm_2 haben, wird deren Beschreibung mit Hilfe wissensbasierter Komponenten vorgeschlagen. Zur Anwendung kommt ein fuzzybasiertes Modell, das in den Beobachteralgorithmus eingebunden wird. Im Ergebnis entsteht der Hybride nichtlineare Arbeitspunkt-Beobachter (Bild 3.1).

Der Hybride nichtlineare Arbeitspunkt-Beobachter ist gekennzeichnet durch:

⇒ ein nichtlineares Prozeßmodell mit den Zustandsvariablen ($\hat{x}_1$ und $\hat{x}_2$) und den Eingangsgrößen Druckänderungsgeschwindigkeit (u_1) und Wärmeströme der Zonen 1und 2 (u_2 und u_3),

⇒ eine konstante Verstärkungsmatrix **K**,

⇒ eine fuzzy-basierte Adaption der Wärmeströme (u_2 und u_3), die als zusätzliche Eingangsgrößen des Beobachtermodells interpretiert werden.

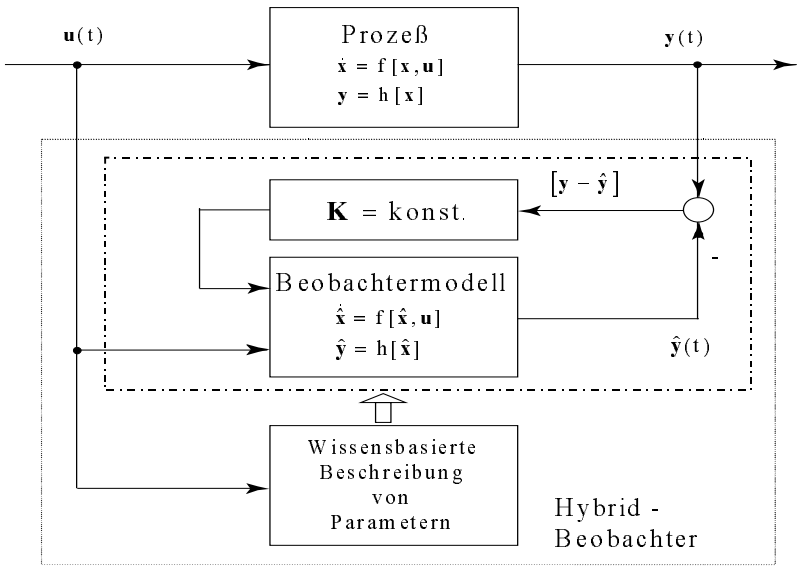


Bild 3.1: Struktur des Hybriden nichtlinearen Arbeitspunkt-Beobachters

Bild 3.2 zeigt die Rekonstruktionsergebnisse der Zustandsgrößen Massenfüllstand in den Zonen 1 und 2 eines Druckbehälters mit Wasser-Dampf Gemisch im Vergleich zum Prozeßverlauf (ATHLET) für die folgenden Beobachtervarianten:

⇒ klassischer nichtlinearer PI-Arbeitspunkt-Beobachter ($\hat{x}1\_BEO$, $\hat{x}2\_BEO$) (Modell ohne Berücksichtigung des Wärmestromes)

⇒ hybrider nichtlinearer PI-Arbeitspunkt-Beobachter ($\hat{x}1\_h\_BEO$, $\hat{x}2\_h\_BEO$) Bestimmung des Wärmestromes durch ein Fuzzy-Kennfeld)

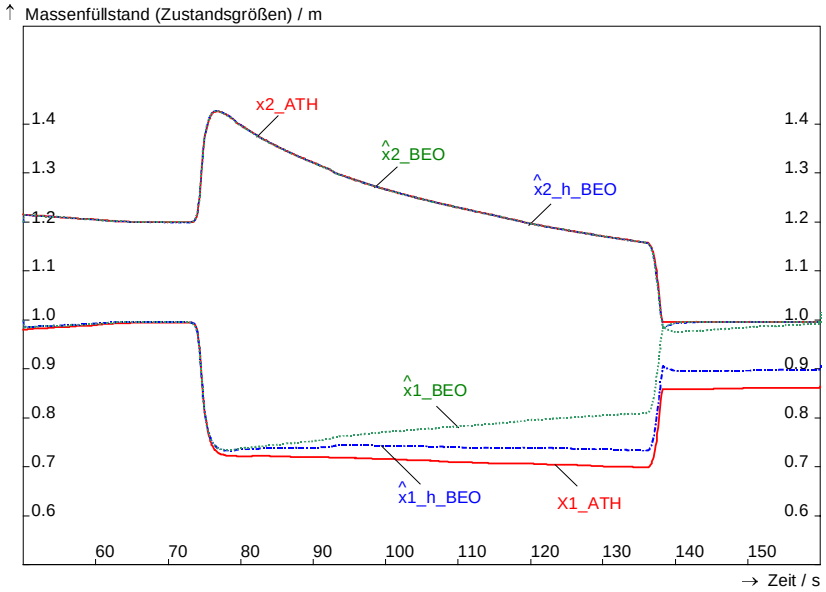


Bild 3.2: Zeitlicher Verlauf der Zustandsgrößen (Massenfüllstand der Zonen 1 und 2) von Prozeß (ATH), klassischem nichtlinearen PI-Arbeitspunkt-Beobachter (BEO) und hybridem nichtlinearen PI-Arbeitspunkt-Beobachter (h_BEO) für ein Druckentlastungsexperiment ($p_0 = 2.4$ MPa)

Im Ergebnis der fuzzy-basierten Ermittlung des Wärmestromes ist der Hybridbeobachter durch eine deutliche Verbesserung der Rekonstruktionsgüte der nichtmeßbaren Zustandsgröße x1 (Massenfüllstand in der Zone 1) gegenüber dem klassischen Beobachter gekennzeichnet.

Im weiteren wird der Schwerpunkt auf das Fuzzy Modell zur Nachbildung des Wärmestromes gerichtet. Dabei ist zu klären, welche Eingangsgrößen / Ausgangsgrößen und welche Struktur für das Fuzzy Modell zu wählen sind, um eine hinreichend genaue Rekonstruktion der nichtlinearen dynamischen Charakteristik des Wärmestromes zu gewährleisten.

4 Beschreibung physikalischer Parameter mittels Fuzzy Algorithmen

4.1 Fuzzy Modellierung

Für die Beschreibung physikalischer Parameter mittels Fuzzy Algorithmen wird eine hinreichende Anzahl aussagekräftiger Prozeßgrößen und Datensätze benötigt. Mit Hilfe von Simulationsrechnungen mittels des ATHLET-Codes wurde eine Datenbasis generiert, die den Zusammenhang zwischen dem Wärmestrom und den Prozeßgrößen Druck und Druckänderungsgeschwindigkeit für unterschiedliche Parameterbereiche als Kennfeld charakterisiert. Die Parameterbereiche wurden aufbauend auf durchgeführten Experimenten an der Druckbehälterversuchsanlage (DHM) innerhalb des für den vorliegenden Prozeß sinnvollen technologischen Rahmens gewählt:

⇒ gleicher Anfangswert für den Massenfüllstand (hm_0 = 1.20 m)
⇒ Variation des Anfangsdruckes (p_0 = 2.4 MPa ,..., p_0 =1.4MPa).

Bezogen auf eine mögliche analytische Modellierung des Wärmestromes werden mit dem Fuzzy Modell folgende Vereinfachungen erzielt:

⇒ Modellierung der Wärmeübergangszahl und der Strömung ist enthalten
⇒ Ersetzung der DGL für die Temperaturänderung.

Damit erfolgt die Formulierung eines globalen Zusammenhanges (Übertragungsverhalten) dieser Größen. Dies bedeutet eine erhebliche Parameterreduktion für ein vollständiges Zustandsraummodell zu einem Zweizonenmodell mit teilweise fuzzybasierter Modellierung, sowie eine Erhöhung des Gültigkeitsbereiches auf den gesamten Parameterbereich.

In dem generierten Kennfeld ist das vollständige Wissen über den Wärmeübergangsprozess abgelegt. Mit Hilfe der Fuzzy Set Theorie eröffnet sich eine einfache und transparente Möglichkeit, das Kennfeld auf die Anlagenverhältnisse, Experimente und Simulationen anzupassen. Die Transparenz wird im Gegensatz zu einem analytischen Modell mit einer Vielzahl von Parametern durch die Verwendung von Regeln erhöht, die daten- und erfahrungsgestützt sind.

4.1.1 Fuzzy Kenngrößen

Aufgrund der Erfahrungen aus vorangegangenen Untersuchungen [13] wurden die folgenden Randbedingungen für die Fuzzy Modelle gewählt:

⇒ Überlappungsgrad: 100%
⇒ T-Norm (Inferenz): T2
⇒ S-Norm (Akkumulation): S2
⇒ Defuzzifizierung: Singleton
⇒ Form der Zugehörigkeitsfunktion: lambda

Die Modellierung und Abarbeitung der Fuzzy Algorithmen erfolgt mit dem Simulationsprogramm DynStar [1]. Es werden ausschließlich Fuzzy Modelle nach Mamdani verwendet [17].

4.1.2 Wahl der Ein- und Ausgangsgrößen

Als Modell wurde ein einfaches 2-D Fuzzy Modell mit zwei Eingangsgrößen und einer Ausgangsgröße gewählt, um die Funktionsfähigkeit der Fuzzy Algorithmen und die Möglichkeit der Rekonstruktion des Wärmestromes zu demonstrieren. Aufgrund von Untersuchungen zur Sensibilität und Wahl der Fuzzy Sets wurden als Eingangsgrößen die Druckänderung und die Druckänderungsgeschwindigkeit festgelegt. Der Druck als Absolutwert konnte nicht als Eingangsgröße dienen, da der Enddruck für die Experimente jeweils $p_e = 0.7$ MPa beträgt und damit eine eindeutige Zuordnung erschwert wird. Dieses Problem wird mit dem Einsatz der Größe Druckänderung umgangen. Während des Druckentlastungsvorganges werden gleiche Parameterbereiche der Eingangsgrößen des Fuzzy Modells mehrfach durchlaufen (Bild 2.2 und 2.3). Die Ausgangsgröße des Fuzzy Modells bildet der Wärmestrom.

4.2 Klassisches Fuzzy Modell

4.2.1 Struktur

Zunächst wurde eine Modellierung des nichtlinearen Prozesses mittels eines einfachen klassischen 2D-Fuzzy Modells durchgeführt (Bild 4.1).

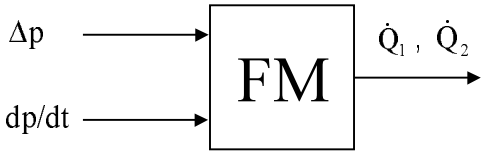


Bild 4.1: Struktur des klassischen Fuzzy Modells

4.2.2 Basisregel

Die Basisregel des klassischen Fuzzy Modells lautet:

IF 'Δp' AND 'dp/dt' THEN '$\dot{Q}_1$' (7)

IF 'Δp' AND 'dp/dt' THEN '$\dot{Q}_2$' (8)

Die Definition der linguistischen Variablen, der Sets und die Generierung der Regelbasis erfolgt auf der Grundlage von drei Referenzdatensätzen mit den Anfangsdrücken ($p_0 = 2.4$ MPa , $p_0 = 2.0$ MPa , $p_0 = 1.4$MPa).

4.2.3 Fuzzy Sets

In den Bildern 4.2 und 4.3 sind die Definition der linguistischen Variablen, die Anzahl und die Verteilung der Fuzzy Sets für die Eingangsgrößen des Fuzzy Modells dargestellt.

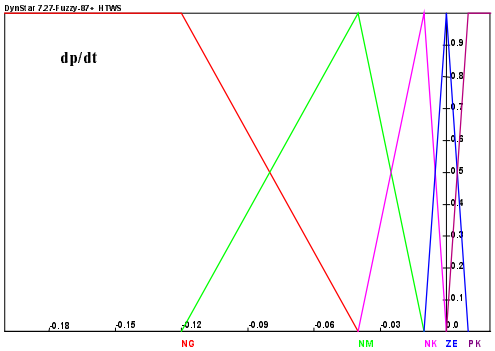


Bild 4.2: Fuzzy Set für die ling. Variable Druckänderungsgeschwindigkeit

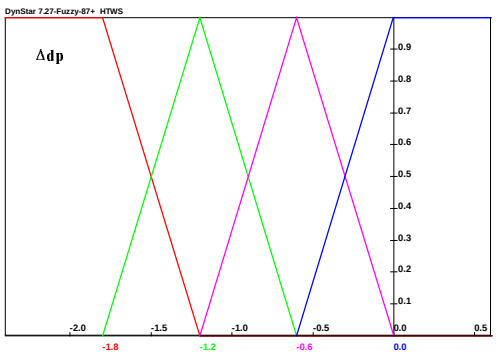


Bild 4.3: Fuzzy Set für die ling. Variable Druckänderung

Aufgrund der nichtlinearen Charakteristik des Prozesses wird für die Druckänderungsgeschwindigkeit eine unsymmetrische Verteilung der Sets erforderlich. Bild 4.4 zeigt die Definition und Verteilung der Sets für die Ausgangsgröße Wärmestrom beispielhaft für die Zone 2 ($\dot{Q}_2$).

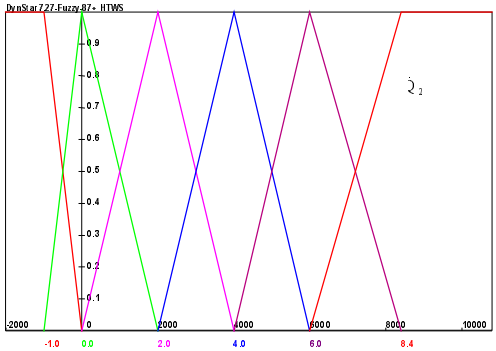


Bild 4.4: Fuzzy Set für die linguistische Variable Wärmestrom Zone 2 $\dot{Q}_2$

4.2.4 Regelbasis

Die Beziehungen zwischen den linguistischen Variablen werden mit Hilfe der Regelbasis definiert. Die Regelbasis enthält die Information über die Form des Fuzzy Kennfeldes. Nachfolgend werden beispielhaft einige ausgewählte Einzelregeln der Regelbasis für den Wärmestrom der Zone 2 dargestellt:

```
IF 'Δp'=  '-0.6' AND  'dp/dt'=  'NK' THEN   'Q2'=  '8.4'
IF 'Δp'=  '-0.6' AND  'dp/dt'=  'PK' THEN   'Q2'=  '0.0'
IF 'Δp'=  '-1.2' AND  'dp/dt'=  'ZE' THEN   'Q2'=  '0.0'
IF 'Δp'=  '-1.8' AND  'dp/dt'=  'ZE' THEN   'Q2'=  '0.0'
IF 'Δp'=  '0.0'  AND  'dp/dt'=  'NM' THEN   'Q2'=  '6.0'
IF 'Δp'=  '0.0'  AND  'dp/dt'=  'NG' THEN   'Q2'=  '6.0'
```

Mit dem klassischen statischen Fuzzy Modell ist die hinreichend genaue Rekonstruktion der Wärmeströme in beiden Zonen für die Datensätze im gesamten angenommenen Parameterbereich nicht erreichbar (vgl. Abschnitt 4.4). Die Güte der Nachbildung wurde auch nicht durch eine Erhöhung der Anzahl der Sets verbessert. Im Besonderen wurde die Dynamik des Prozesses nicht exakt widergespiegelt. Das Passieren gleicher Parameterbereiche der Ein- und Ausgangsgrößen bei der Druckentlastung erschwert die Unterscheidung der Regeln.

4.3 Fuzzy Modell mit externer Dynamik

4.3.1 Struktur

Aufgrund von Erfahrungen zur Simulation nichtlinearer Prozesse [6] und Strukturuntersuchungen wird eine neue Form des Fuzzy Modells zur Anwendung gebracht, ein Fuzzy Modell mit externer Dynamik. Das klassische Fuzzy Modell wird in ein dynamisches Fuzzy Modell überführt, indem es um eine dynamische Struktur erweitert wird. Die Eingangs- und Ausgangsgrößen des klassischen Fuzzy Modells bleiben auch beim dynamischen Fuzzy Modell erhalten. Das dynamische Modell umfaßt den eigentlichen Modellalgorithmus, die Integration der direkten Ausgangsgröße des Modells sowie die Rückführung des integrierten Ausgangssignals. Die direkte Ausgangsgröße des Modellalgorithmus' ist nunmehr der Gradient des Wärmestromes. Diese Größe wird integriert und als weitere Eingangsgröße auf das Fuzzy Modell zurückgeführt [3]. Bild 4.5 veranschaulicht die Struktur des dynamischen Fuzzy Modells und die Definition seiner Eingangs- und Ausgangsgrößen.

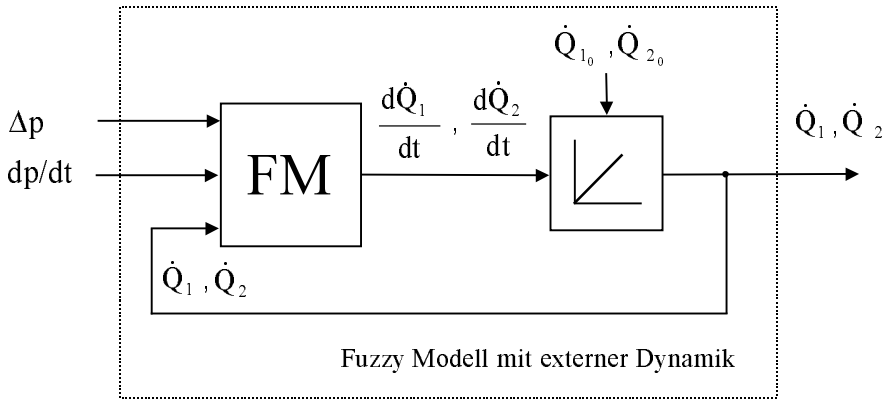


Bild 4.5: Struktur des Fuzzy Modells mit externer Dynamik

Die nichtlineare Statik des Fuzzy Kennfeldes wird um eine externe Dynamik erweitert. Für die Verifikation des Algorithmus‘ ist von Vorteil, daß die Ausgangsgröße des Fuzzy Modells (zeitliche Ableitung des Wärmestromes) eine physikalisch interpretierbare Größe ist. Der Stabilitätsnachweis für das dynamische Fuzzy Modell wird aus der Prozeßkenntnis abgeleitet. Möglichkeiten für einen mathematischen Stabilitätsnachweis für nichtlineare Modelle wie Fuzzy Controller werden in [5] dargestellt. Prinzipiell lautet die Definition so, daß ein geschlossenes System stabil genannt wird, wenn das Fuzzy Modell tolerierbare und sichere Werte für die Ausgangsgrößen für begrenzte interne und externe Störungen garantiert.

4.3.2 Basisregel

Die Basisregeln für das Fuzzy Modell zur Nachbildung der Wärmeströme der Zone 1 und 2 lauten:

IF 'Δp' AND 'dp/dt' AND '$\dot{Q}_1$' THEN '$d\dot{Q}_1/dt$' (9)

IF 'Δp' AND 'dp/dt' AND '$\dot{Q}_2$' THEN '$d\dot{Q}_2/dt$' (10)

Die Definition der linguistischen Variablen, der Sets und die Generierung der Regelbasis erfolgt auf der gleichen Grundlage wie bei dem klassischen Fuzzy Modell.

4.3.3 Fuzzy Sets

Die Definition der linguistischen Variablen für die Eingangs- und Ausgangsgrößen bleibt entsprechend dem klassischen Fuzzy Modell erhalten. Hinzu kommt die Definition der Zugehörigkeitsfunktionen für den Gradienten des Wärmestromes $d\dot{Q}/dt$ (Bild 4.6).

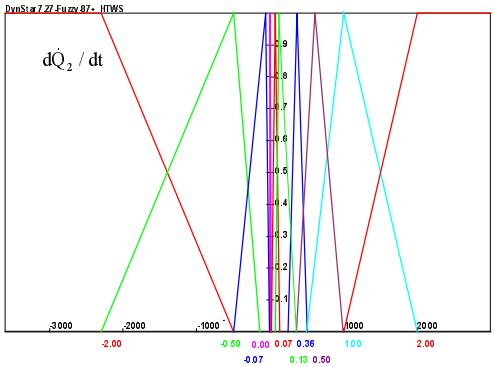


Bild 4.6: Fuzzy Set der linguistischen Variable Gradient des Wärmestromes $d\dot{Q}_2/dt$

Infolge der hohen Dynamik des Prozesses und der Sensibilität des Gradienten ist eine hohe Anzahl und eine unsymmetrische Verteilung von Sets zur Beschreibung der nichtlinearen Charakteristik notwendig, beispielhaft dargestellt für das Fuzzy Set $d\dot{Q}_2/dt$.

4.3.4 Regelbasis

Die Beziehungen zwischen den linguistischen Variablen werden mit Hilfe der Regelbasis definiert. Die Regelbasis enthält die Information über die Form des Fuzzy Kennfeldes. Es werden einige ausgewählte Einzelregeln der Regelbasis für den Wärmestrom der Zone 2 dargestellt:

```
IF 'Δp'= '-0.6'   AND 'dp/dt'='NK' AND  'Q2f'= '6.0'   THEN 'dQ2'= '-0.07'
IF 'Δp'= '-1.2'   AND 'dp/dt'='NG' AND  'Q2f'= '8.4'   THEN 'dQ2'= '-0.01'
IF 'Δp'= '-1.2'   AND 'dp/dt'='PK' AND  'Q2f'= '4.0'   THEN 'dQ2'= '-2.00'
IF 'Δp'= '-1.8'   AND 'dp/dt'='ZE' AND  'Q2f'= '4.0'   THEN 'dQ2'= '-0.15'
IF 'Δp'= '0.0'    AND 'dp/dt'='NM' AND  'Q2f'= '0.0'   THEN 'dQ2'= '0.50'
IF 'Δp'= '0.0'    AND 'dp/dt'='NK' AND  'Q2f'= '4.0'   THEN 'dQ2'= '0.50'
```

4.4 Ergebnisse

4.4.1 Nachweis der Funktionsfähigkeit

Nachfolgend werden die Ergebnisse der Rekonstruktion des Wärmestromes mit dem klassischen und dynamischen Fuzzy Modell gegenübergestellt. Dazu wird der Vergleich für ein Druckentlastungsexperiment gezogen, das als Referenzdatensatz für die Erstellung der Fuzzy Algorithmen verwendet wurde, um die Funktionsfähigkeit der Fuzzy Modelle zu demonstrieren. Für ein Druckentlastungsexperiment mit dem Anfangsdruck von $p_0 = 2.4\,\mathrm{MPa}$ wird im Bild 4.7 der zeitliche Verlauf des Wärmestromes für die Zone 2 im Vergleich gezeigt.

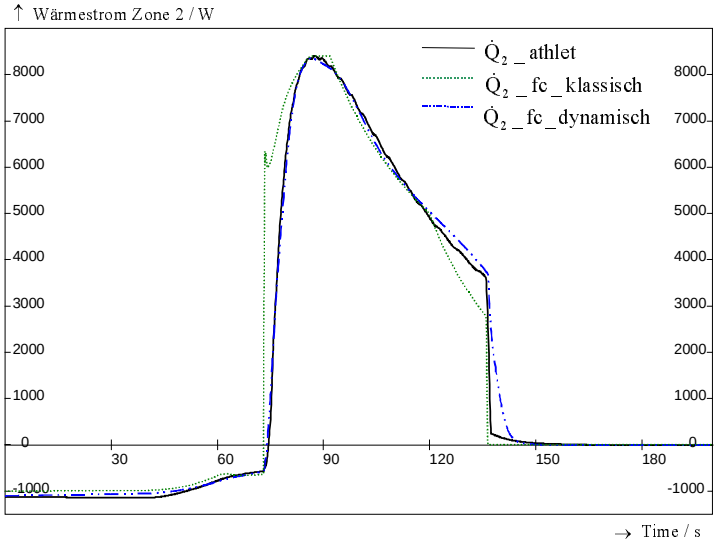


Bild 4.7: Zeitlicher Verlauf des Wärmestromes Zone 2, berechnet mit dem ATHLET (Prozeß) und dem klassischen und dynamischen Fuzzy Modell, $p_0 = 2.4\,\mathrm{MPa}$

Dargestellt ist die durch den ATHLET-Code bereitgestellte Größe (nichtmeßbares Prozeßsignal) und der durch die klassischen und dynamischen Fuzzy Modelle nachgebildete Wärmestrom der Zone 2. Es ist zu erkennen, daß die Wärmeströme durch

das dynamische Fuzzy Modell mit hinreichend großer Genauigkeit nachgebildet werden, währenddessen diese Güte mit dem klassischen Modell nicht erreicht wird.

4.4.2 Allgemeingültigkeit

Mit einem Datensatz innerhalb des gewählten Parameterbereiches, der nicht dem Modellentwurf zu Grunde lag, wird der Nachweis der Allgemeingültigkeit geführt. Damit werden die guten Interpolationseigenschaften und die Robustheit der Fuzzy Modelle demonstriert. Für ein Druckentlastungsexperiment mit Anfangsbedingungen (hier $p_0 = 1.6$ MPa) wird im Bild 4.8 die Rekonstruktionsgüte für den Wärmestrom Zone 2 für das klassische und dynamische Modell verglichen. Das dynamische Modell bildet den Wärmestrom mit großer Genauigkeit nach. Diese Aussage gilt ebenfalls für Datensätze, deren Anfangsdruck im gesamten Parameterbereich von $p_0 = 2.4 \ldots 1.4$ MPa variiert. Das klassische Modell ist nicht in der Lage, den Verlauf des Wärmestroms mit hinreichender Genauigkeit zu reproduzieren.

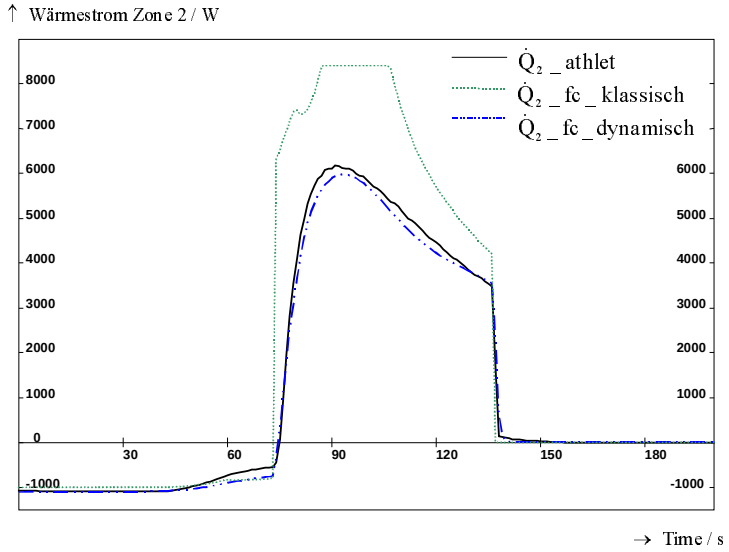


Bild 4.8: Zeitlicher Verlauf des Wärmestromes Zone 2, berechnet mit dem ATHLET-Code (Prozeß) und dem klassischen und dynamischen Fuzzy Modell, $p_0 = 1.6$ MPa

Die hohe Güte der fuzzy-basierten Nachbildung des Wärmestromes hat zu der im Bild 3.2 beschriebenen Verbesserung der Rekonstruktion der nichtmeßbaren Zustandsgröße geführt. Im Mittelpunkt der Untersuchungen stand der Nachweis der Realisierbarkeit einer einfachen fuzzy-basierten Modellierung des Wärmestromes, nicht die Optimierung des Fuzzy Modells.

5 Zusammenfassung

Die Kombination der Verfahren der analytischen Redundanz mit Methoden der Verarbeitung unscharfer Informationen (Fuzzy-Set Theorie) führt zu einer neuen Qualität von fortgeschrittenen modellgestützten Meßverfahren. Dies wurde am Beispiel des Hybriden nichtlinearen Arbeitspunkt-Beobachters demonstriert.

Die Vorteile des Hybridverfahrens bestehen in der Nutzung bereits vorhandener analytischer Modellstrukturen und in der Verbesserung der Rekonstruktionsgüte der Zustandsgrößen stark nichtlinearer Prozesse.

Wissensbasierte Algorithmen sind immer dann einsetzbar, wenn Erfahrungswissen über den Prozeßverlauf, experimentelle Daten bzw. Simulationsdaten vorliegen.

Die exakte Nachbildung transienter Parameterverläufe mit Hilfe von Fuzzy Algorithmen erfordert den Einsatz dynamischer Fuzzy Modelle. Besondere Transparenz erhält die Struktur durch Verwendung von real meßbaren Größen als Eingangs- und Zwischengrößen der Modelle.

Die Entwicklung des Hybriden nichtlinearen Arbeitspunkt-Beobachters stellt einen Beitrag zur nichtlinearen Modellierung nichtlinearer Prozesse mit Methoden des Soft Computing dar.

Literatur

[1] *DynStar - Ein Simulationsprogramm für Automatisierungstechniker*, Version 7.23 Fuzzy +, 1999

[2] Fenske (Traichel), A.: *Bestimmung nichtmeßbarer Zustandsgrößen in stark nichtlinearen Systemen*. IPM Bericht, 1997

[3] Fenske (Traichel), A.; Hampel, R.; Kästner, W.: *Model-based and knowledge-based measuring methods for the observation of non-linear processes.* Proceeding of the 5th Zittau Fuzzy Colloquium, 1997

[4] Schäfers, E.; Krebs, V.: *Dynamic Fuzzy Systems for Qualitative Process Modeling: Principles of a New Systems Theory*. at 47 (1999) 8, S. 383-389

[5] Mikut, R.: *Modellgestützte on-line Stabilitätsüberwachung komplexer Systeme auf der Basis unscharfer Ljapunow Funktionen*. Dissertation. Universität Karlsruhe, 1999

[6] Hampel, R.; Chaker, N.: *Application of Fuzzy Logic in Control and Limitation Systems using Industrial Hardware*, 3rd International Mendel Conference, Brno, Czech Republic, June 1997

[7] Babuška, R: *Fuzzy Modeling for Control*. Kluwer Academic Publishers, Dordrecht, 1998

[8] Nelles, O.; Ernst, S.; Isermann, R.: *Neuronale Netze zur Identifikation nichtlinearer, dynamischer Systeme: Ein Überblick*. Übersichtsaufsatz, at 45 (1997) 6, S. 251-262

[9] Popp, H. E.: *Hybride wissensbasierte Systeme zur Datenanalyse und Eigenschaftsvorausberechnung physikalisch-chemischer Systeme dargestellt an Elektrolytlösungen.* Dissertation, Universität Regensburg, 1992

[10] Wey, T.; Spielmann, M.: *Analytische und Fuzzy-Regelungskonzepte am Beispiel eines aufechtstehenden Pendels*. at 47 (1999) 1, S. 20-28

[11] Simutis, R.; Havlik, I.; Lübbert, A.: *A fuzzy-supported Extended Kalman Filter: a new approach to state estimation and prediction exemplified by alcohol formation in beer brewing.* Journal of Biotechnology, 24, 1992, S. 211-234

[12] Wollert, J.: *Ein Beitrag zur kognitiven Modellierung regelungstechnischer Systeme unter Berücksichtigung heuristischer und analytischer Merkmale.* VDI-Fortschrittsbericht Reihe 8, Nr. 450, VDI-Verlag, Düsseldorf, 1994

[13] Hampel, R.; u. a.: *Meß- und Automatisierungstechnik zur Störfallbeherrschung - Methoden der Signalverarbeitung, Simulation und Verifikation*, Abschlußbericht BMBF-Projekt 150 10 15, HTWS Zittau/Görlitz (FH), 1999

[14] Adjallah, K.: *Non-linear Observers using Fuzzy Gain Adaptation*. International Workshop on Fuzzy Technologies in Automation and Control, Duisburg 1994, S.73-85

[15] Berger, M.;Jelali, M.: *Robust Model-Based Fuzzy Observer for an Inverted Pendulum*. IEEE Transactions 1996, S. 118-122

[16] Leiviskä, K.: *Industrial Applications of Intelligent Methods*. EUFIT'98, Aachen, 7.-10. September 1998

[17] Kiendl, H.: *Fuzzy Control methodenorientiert*. Oldenbourg-Verlag, München, 1997

Untersuchung zur Anwendbarkeit von Künstlichen Neuronalen Netzen (KNN) zur Steuerung /Regelung komplexer, nichtlinearer Systeme.

Dipl. Ing. / LL.M Pat. Ing D. Karimanzira
Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
Postfach 10 0565, 98684 Ilmenau
e-mail : divas.karimanzira@systemtechnik.tu-ilmenau.de

Kurzfassung

Insbesondere für komplexe nichtlineare Systeme haben sich in den letzten Jahren in verstärktem Maße Regelungskonzepte mit Künstlichen Neuronalen Netzen durchgesetzt. Inzwischen existiert eine Vielzahl unterschiedlicher Strukturen. Ziel des vorliegenden Beitrages ist es, vergleichende Untersuchungen zur Eignung dieser Regelungskonzepte für nichtlineare Systeme vorzunehmen, damit Aussagen über ihre Vor- und Nachteile sowie die sich daraus ergebenden Einsatzmöglichkeiten getroffen werden können. Die Untersuchungen konzentrieren sich für unterschiedliche Streckentypen bei stationären und instationären Systemeigenschaften auf folgende Schwerpunkte:

- Führungsverhalten,
- Störverhalten,
- Stabilitätseigenschaften sowie
- Robustheit.

1.0 Zielsetzung

Ziel ist es festzustellen, inwieweit neuronale Verabeitungsprinzipien zur Steuerung und Regelung nichtlinearer Systeme technisch einsetzbar sind, unter welchen Voraussetzung sowie mit welchem Trainingsaufwand praxisrelevante Aufgabestellungen gelöst werden können und welche Systemleistungen zu erwarten sind. Es wird gezeigt,

- was neuronale Steuer- und Regelungen hinsichtlich ihre Echtzeit- und Adaptionsfähigkeit sowie Genauigkeit und Robustheit leisten können,
- wie sich neuronale Verarbeitungsprinzipien zur Steuer- und Regelung realer nichtlinearen Systemen einsetzen lassen,

2.0 Hintergrund

Für die technische Reglerauslegung kommt es darauf an, einen geeigneten Kompromiß zwischen hoher Regelgüte und geringem Komplexitätsgrad unter Beachtung von Stabilität, Genauigkeit und Übergangsverhalten zu finden. Entsprechend der unterschiedlichen Komplexität der zu regelnden Systeme existieren bereits zahlreiche verfahren zur Reglerauslegung. Allerdings zeigt sich hier, daß diese Verfahren auf lineare Regelungssysteme beschränkt sind, während für nichtlineare Systeme selbst bei bekannten Übertragungsfunktionen kaum industriell verwertbare

Ansätze existieren. Es gibt einige adaptive Reglungsstrategien wie Korrelationstechniken, Zufallsmethoden, Gradientenverfahren, stochstischen und Fuzzy-Automaten sowie globale Suchverfahren. Ihnen ist gemein, daß sie anwendungsspezifisch sind und von dem verwendeten Optimierungskriterium abhängen. Zudem sind sie nur sehr eingeschränkt fähig, sich zeitlich veränderlichen Systemparametern bzw. neuen Optimierungskriterien automatisch anzupassen.

3.0 Analyse neuronaler Steuer- und Regelungskonzepte, Klassifizierung neuronaler Steuer- und Regelungsverfahren

Vor diesem Hintergrund entsteht der Wunsch nach einem autonomen Regelunskonzept, das sich in einem interaktiven Prozeß an veränderliche Systembedingungen anpaßt und das Regelverhalten erlernt. In [Hunt,92] werden unter anderem Eigenschaften neuronaler Netze herausgestellt, die für einen echtzeitfähigen, multivariablen, nichtlinearen, autonomen und fehlertoleranten Regler nötig sind. Je nach Trainingsanordnung stellen sie eine Vorsteuerung zur Kompensation von Nichtlinearitäten dar oder sie sind als arbeitspunktabhängige neuronale Regler ausgelegt. Eine Möglichkeit, neuronale Regelungskonzepte zu klassifizieren, ist in Tabelle 1 wiedergegeben.

Kategorie	Regelungskonzept		Referenzen
Direkte neuronale Regelung	▪ Indirektes, generalisiertes und spezialisiertes Lernen[DIC]		PSAL,87
	▪ Inverse Transfermatrix Regelung	*	JORD,89a/89b
	▪ Feedback Error Regelung [FEC]		KAWA,87/87a
	▪ Hierarchiche Neuronale Netzwerk Regelung [HAC]		KAWA,87
Indirekte neuronale Regelung	▪ Indirekte Inverse Regelung		JORD,89a/89b
	▪ Indirekte Feedback Error Regelung	*	
	▪ Interne Modellregelung [IMC]		GARC,82; BHAT,89
	▪ Model Reference Adaptive Control [MRAC]		NARE,90a
	▪ Self-Tuning Control [STC]		TZIR,91
	▪ Cerebellar Model Arithmetic Computer [CMAC]	*	MILL,87
Optimierende Regelung	▪ Prädiktive Regelung [APC], [NPC]		SAIN,91
	▪ Backpropagation-Through-Time	*	WERB,90
	▪ Method of Temporal Differences	*	NGUY,90a/90b
Unüberwachte Steuer-/Regelung	▪ Erweiterte Self-Organizing Feature Maps [SOM]	*	KOTTO,91
	▪ Locally Linear Maps	*	RITT,89
Überwachte	▪ Kopieren einer Steuerung		SIG,95

Tabelle 1: Eine mögliche Klassifikation neuronaler Regelungskonzepte

Im folgenden werden einige ausgewählte neuronale Regelungskonzepte im Detail analysiert.

3.1 Analyse direkter neuronaler Steuer- und Regelungskonzepte Trainingsverfahren zur neuronalen Vorsteuerung

Im Grunde genommen gibt es zwei alternative Ansätze zum Trainieren von Reglern, die auf neuronalen Verabeitungsprinzipien basieren. Dies sind die von WIDROW entwickelte direkte inverse Modellierung [WIDR,85] und der von JORDAN vorgeschlagene Einsatz eines differenzierbaren Modells [JORD,89a/89b]. Für direkte inverse Regelung sind schon mehrere neuronale Lernschemata entwickelt worden[PSAL,87]: indirektes, generalisiertes und spezialisiertes Lernen. Neben der Problematik, daß die inverse Abbildung meistens nicht existiert, hat das Konzept der direkten inversen Vorsteuerung den Nachteil, nicht zielgerichtet zu sein. So liegt nach ausreichendem Training trotz einer Stellgrößendifferenz *(e_u=0)* immer noch eine Regeldifferenz *($e_y \neq 0$)* vor. Von Vorteil ist, daß das indirekte Lernschema online trainierbar und auf den gewünschten Arbeitspunkt beschränkt ist. Die Simulationsergebnisse zeigen sehr deutlich die möglichen negativen Auswirkung des direkten, nicht zielgerichteten Trainierens. Da die Stellgrößenabweichung infinitesimal klein wird, erfolgt kein Anpassen der Netzwerkparameter mehr, womit sich schließlich ein Systemverhalten ergeben kann, das weit ab von dem Führungsgrößenverlauf liegt. Der Erfolg des generalisiertes Lernschema ist eng mit der Generalisierungsfähigkeit neuronaler Netzwerke verbunden, da ein stabiles Verhalten auch für nicht im Trainingset enthaltene Eingangsgrößen gefordert ist. Aus diesem Grunde setzt diese Vorgehensweise voraus, daß Stellgrößensignale vorliegen (diese können zum Beispiel von einem Experten generiert werden) und daß diese den gesamten Arbeitsbereich des Systems abdecken. Allerdings wird damit ein Netzwerk möglicherweise viel zu weit jenseits des eigentlichen Arbeitspunktes trainiert. Um die Sensitivität für den Arbeitspunkt zu erhöhen, kann eine geeignete nichtlineare Verzerrung bzw. Reskalierung der Signale vorgenommen werden.

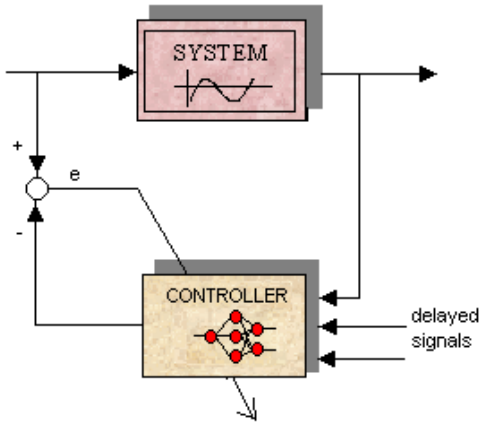


Fig 3.1a: Blockbild eines generalisierten Lernschema.

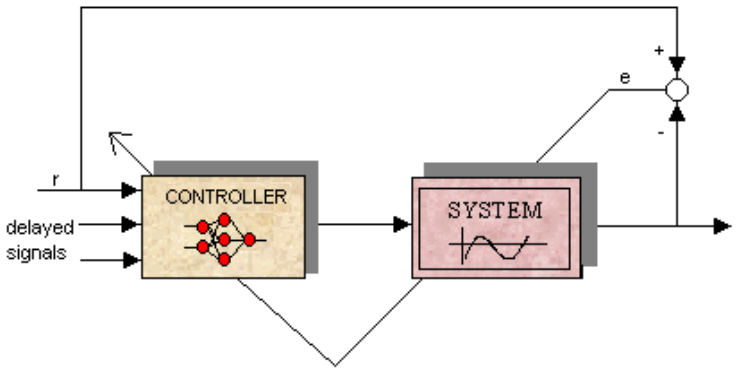


Fig 3.1b: Blockbild eines spezielisierten Lernschema.

Die spezialisierte Lernarchitektur eignet sich für ein Online-Adaptieren um den Arbeitspunkt und erfüllt die Stabilitätsanforderung bezüglich einer begrenzten Ein- und Ausgabe (BIBO-stabilität) [PSAL,88]. Dieses Lernschema wurde entwickelt, um die von der generalisierten Lernarchitektur vorgenommene Approximation zu vermeiden. Nachteilig beim spezialisierten Lernen ist das schlechte Konvergenzverhalten und die direkte Beeinflussung des differentiellen Verstärkungsfaktors auf die Netzwerkadaption. Hinsichtlich des direkten Einflusses der Verstärkung auf die Netzwerkdynamik ist festzuhalten, daß stark schwankende I/O-Relationen ein entsprechendes Schwanken der Netzwerkparameter nach sich ziehen.

Bei den Feedback Error Regelungskonzept dient der konventionelle Regler insbesondere der Regelkreisstabilisierung und Unterdrückung von Störungen, ohne

daß dieser optimal eingestellt sein muß während der Feedforward Regler für die schnellere Verfolgung des Referenzsignals sorgt. Das auf der Basis der rückgekoppelten Regeldifferenz e_y vom PID-Regler um den Faktor Kp verstäkter Stellsignal u_{FB} dient dabei als direktes Maß für die Gewichtsadaption des neuronalen Netzes. Es bedarf folglich keiner Rückpropagierung der Regeldifferenz durch das System bzw. eines Systemmodells. Zudem kann das neuronale Netz online trainiert werden; Die Trainingsmethode ist zielgerichtet, daß heißt u_{FB} wird mit einem Verschwinden der Regeldifferenz zu null. In diesem Fall liegt eine Identitätsabbildung vor. Das Feedback-Error-Regelungskonzept zeichnet sich aufgrund des direkten Einflusses des Regeldifferenz durch eine hohe Konvergenzgeschwindigkeit aus. Allerdings ist insbesondere für nichtlineare zeitvariante Systeme der Verstärkungsfaktor Kp genau abzuschätzen. Deshalb ist für ein stabiles Verhalten eine entsprechend kleine Lernrate „η“ einzustellen, was allerdings die Konvergenzrate herabsetzt. Ferner wird eine optimale Regelung erst dann erzielt, wenn das neuronale Netz praktisch die Systeminverse korrekt identifiziert.

3.2 Analyse indirekter neuronaler Steuer- und Regelungskonzept

Ein wesentliche Unterschied des indirekten inversen Lernschemas gegenüber der direkten inversen Modellierung besteht darin, daß anstelle einer Interpolation im Suchraum nur spezielle Lösungen gefunden werden. Zudem ist das Verfahren zielgerichtet. Bei diesem Verfahren muß unter Voraussetzung, daß das Systemmodell perfekt ist und das verwendete Lernverfahren das globale Optimum findet, beim Hintereinanderschalten des Reglers mit dem Modell auch die systemzustandabhängige I/O-Abbildung korrekt sein. Liegt dagegen ein unvollkommenes Systemmodell vor, wird unter der Maßgabe der Minimierung des Prädiktionsfehler auch die gewünschte I/O-Abbildung falsch sein. Deshalb ist es wichtig als Fehlermaß für das indirekte Training eines neuronalen Reglers die Regeldifferenz $(r-y)$ statt $(r-\hat{y})$ zu verwenden. Auf diese weise läßt sich das gewünschte Gesamtübertragungsvehalten ohne ein perfektes neuronales Streckenmodell abbilden. Der Abbildungsfehler manifestiert sich als Bias-Term während des Trainingsprozesses für den neuronalen Regler [JORD,93]. Allerdings kann es unter bestimmten Bedingungen angebrachte sein, den Prädiktionsfehler zu verwenden, wie etwa bei dem in [SUTT,90] beschriebenen internen Modell Regelungskonzept.

Wegen den Schwierigkeiten, die mit der Stabilität nichtlinearer, rückgekoppleter Regelkreise verbunden sind, schlagen GARCIA und MORARI [GARC,82] das Konzept der Internal Modell Regelung[IMC] vor. Dieses verwendet ein Model der Strecke und ein inverse Abbildung des Streckenmodells und nicht etwa der Strecke. Da unter praktischen Gesichtpunkten eine perfekte Modellabbildung nicht realisierbar ist, wird beim Aufstellen der Gesamtübertragungsfunktion idealisiert. Ungenügende Modellabbildungen verlangen nach einem unendlich großen Verstärkungsfaktor, was zu Sensitivitätsproblemen führt. Um diese zu beheben, wird ein Tiefpaß Filter in den Regelkreis integriert, der eine Begrenzung der Regelverstärkung bewirkt. Ein attraktive Eigenschaft dieses Regelungskonzeptes ist es, daß es auch wenn das System durch konstante Laststörungen beeinflußt wird, ein offsetfreies Antwortverhalten erzeugt.

3.3 Analyse optimierender neuronaler Steuer- und Regelungskonzept

Die Reglerbemessung für einen PID-Regler erfolgt stets für einen im Arbeitspunkt linearisierten Prozeß. Vielfach reicht diese Vorgehensweise aus, um geeignete

Reglerparameter zu finden. Insbesondere wenn die Nichtlinearitäten des zu regelnden Prozesses nur geringfügig und somit eine Linearisierung zulässig ist, kann auf diese Weise ein brauchbarer Regler gefunden werden. Wenn aber die Nichtlinearität des Prozesses eine Linearisierung nicht zulässt wird kein zufriedenstellendes Reglerverhalten im gesamten Arbeitsbereich erzielt. In diesem Fall ist eine Möglichkeit zur Situationsabhängigen Parametervariation wünschenswert. Auf diese Weise arbeitet das INL-Regelungskonzept. Die Besonderheit der INL-Technik ist, daß anstatt ein lineares Streckenmodell an jeder Iteration rekursiv abzuschätzen, wird dies von einem nichtlinearen neuronalen Netzwerkmodell extrahiert. Diese Methode funktioniert zufriedenstellend bei glatte Nichtlinearitäten. Da die Nichtlinearitäten im allgemeinen einen stetigen Verlauf haben, sollte die Lösung der automatischen Parametervariation auch stetig, also weich erfolgen. Somit bietet sich die Verwendung der ‚Nonlinear Prediktive Control' geradezu an. Hier werden die zukünftige Ausgabewerte nicht wie oben durch Linearisierung sondern durch sukzessive rekursive Benutzung des nichtlinearen NNARX-Modells bestimmt Dadurch ist diese Optimierungsproblem etwas kompliziert und benötigt eine iterative Suchmethode, z.B. Quasi-Newton Verfahren. Wie beim NPC Regelungskonzept wird auch beim ‚approximate generalised predictive ' Regelungskonzept in jeder Iteration die folgende Funktion minimiert:

$$J(t,U(t)) = \sum_{i=N_1}^{N_2} \left[r(t+i) - \hat{y}(t+i) \right]^2 + \rho \sum_{i=1}^{N_u} \left[\Delta u(t+i-1) \right]^2 \qquad (1)$$

Das Optimierungsproblem (On-line gelöst, da ein neues lineares Modell in jede Iteration erzeugt wird) erzeugt eine Reihe von Zukunftswerten *u(t)* bis *u(t+Nu-1)* und von dieser Reihe wird das erste Element angewandt. Nachteilig bei diesen optimierenden Regelungskonzepten ist, daß sie optimierungsverfahrenabhängig sind und daß in manche Fälle wie beim NPC die Optimierung langsam erfolgt. Das ‚optimal control' Regelungskonzept benötigt wie beim spezialisiertem Lernschema zwei neuronale Netzwerke, eines für das Streckenmodell und ein zweites für den optimierenden Regler. Zur Erhöhung der Konvegenzrate wird der Regler zur Näherung der Inverse der Strecke trainiert. Im On-Line Betrieb wird bei dieser Methode statt die Funktion nach Gleichung (2) zu minimieren wie beim spezialisierten Lernschema die Funktion nach Gleichung (3) minimiert.

$$J(\Theta) = \sum_t \left(r(t) - y(t) \right)^2 \qquad (2)$$

$$J(\Theta) = \sum_t \left(r(t) - y(t) \right)^2 + \rho \left(u(t) \right)^2, \rho \geq 0 \qquad (3)$$

4.0 Simulationen

Anstelle von praktischen Systemen wurden zur besseren Vergleichbarkeit neuronaler Regelungskonzepte zunächst nichtlineare, dynamische Systeme in Form von Differenzengleichungen verwendet. Die gemäß des nichtlinearen Anteils in vier Klassen unterteilt werden:

- nichtlineare Abhängigkeit der Stellgrößen, lineare Abhängigkeit der Regelgrößen,

$$y(k+1) = u^3(k) + 0.3u^2(k) - 0.4u(k) + 0.225y(k) + 0.45y(k-1) \qquad (4)$$

- lineare Abhängigkeit der Stellgrößen, nichtlineare Abhängigkeit der Regelgrößen,

$$y(k+1)=\frac{y(k)y(k-1)[y(k)+2.5]}{1+y^2(k)+y^2(k-1)}+u(k) \qquad (5)$$

- getrennte nichtlineare Abhängigkeit der Stell- und Regelgrößen,

$$y(k+1)=\frac{y(k)}{1+y^2(k)}+u^3(k) \qquad (6)$$

- nicht getrennte nichtlineare Abhängigkeit der Stell- und Regelgrößen

$$y(k+1)=\frac{y(k)y(k-1)y(k-2)(u(k-1)-1.0)+u(k)}{1+y^2(k-1)+y^2(k-2)} \qquad (7)$$

Als zweites wurden die neuronale Regelungskonzepte zur Steuer- und Regelung eines mechanischen Systems mit Nichtlinearitäten verschiedener Ordnungen eingesetzt und verglichen.

$$\frac{\partial^2 y(t)}{\partial^2(t)}+2\zeta\omega_n\frac{\partial y(t)}{\partial(t)}+\omega_n^2 y(t)+y^3(t)=\omega_n^2 u(t) \quad \omega_n=2\pi*2.5\,rad/\sec \text{ und } \zeta=0.1 \qquad (8)$$

Die Untersuchungen konzetrieren sich für die oben dargestellten Streckentypen bei stationären und instationären Systemeigenschaften auf folgende Schwerpunkte:

- Führungsverhalten,
- Störverhalten,
- Stabilitätseigenschaften,
- Robustheit,
- Rechenzeit und
- Adaptionsfähigkeit.

Wegen des begrenzten Umfangs dieser Arbeit werden hier nur ein paar Simulationsergebnisse in der Tabelle 2 und Bilder(1-5) gezeigt.

5.0 Bewertung

In diesem Abschnitt werden die Simulationsergebnisse für die ausgewählte neuronale Regelungskonzepte bewertet und verglichen.

Ein mit einer Sprungfolge trainierter neuronaler Regler erzeugt kein zufriedenstellender Antwortverhalten bei Anregung durch alternative Referenzsignale. Bild19 zeigt das Antwortverhalten des Regelungssystems, wenn der Regler mit entweder einer Sprungfolge, Sinusoidal oder gemischter Signalen trainiert wurde. Eine Verbesserung bei den durch das gemischte Signal trainierten neuronalen Regler ist deutlich zu sehen.

Regler	PID	FEC	AIC	IMC	HAC	MRA	APC	NPC	INL1	INL2	FBL	OTC	DIC
Zeit für 600 Cyc /s	24.0	36.3	108	40.8	38.7	36.0	40.4	91.3	37.3	41.9	30.2	24.9	24.2

Tabelle 2 : Quantitative Auswertung der Regelungskonzepte.

5.1 Direkt Inverse Control [DIC]:

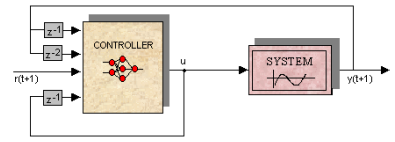


Fig1: Direkte Inverse Regelungsstruktur.

VORTEILE:
- Einfach zu implementieren,
- Optimierungsmöglichkeit für ein bestimmte Referenzsignal durch spezialisierten Training,
- Es ist einfach spezialisierter Training an Zeitvarianten Systemen anzuwenden,

NACHTEILE:
- Nicht anwendbar an Systemen mit instabilen Inversen,
- Probleme bei Systemen mit fehlender Eindeutigkeit,
- Keine Tuningmöglichkeit,
- Zeigt höhere Sensitivität zu Störungen,
- Probleme mit untergedämpften Inversemodellen (Nullstellen in der Nähe des Einheitskreises).

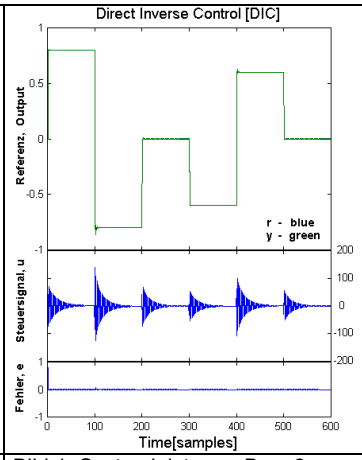


Bild 1: Systemleistung->Bsp. 2

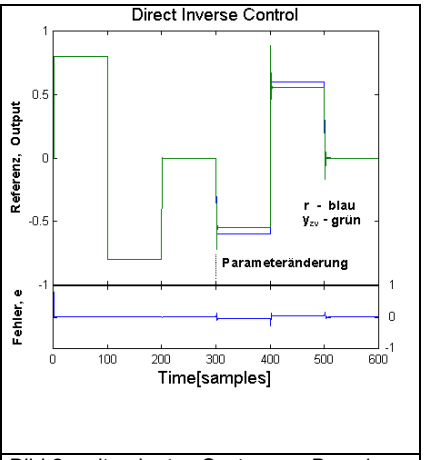


Bild 2: zeitvariantes System - > Bsp. 1

Das DIC-Regelungskonzept zur Steuerung des mechanischen Systems erzeugt sehr gute Ergebnisse [Bild 1], aber es erzeugt eine kleine Überschwingung beim Referenzsignalwechsel. Wie Bild 2 zeigt, eignet sich dieses System nicht zur Steuerung von Systemen mit Störungen oder veränderlicher Systemstruktur. Wie der PID-Regler ist dieser neuronale Regler einfach und schnell. Aus der Darstellung des Steuersignals (nicht hier gezeigt) wird deutlich, daß das Inversemodell außerhalb des Trainingsbereichs benutzt wird.

5.2 Model Reference Adaptive Control [MRAC]:

Deutlich zu sehen in den Ergebnissen des MRAC-Regelungskonzepts (nicht hier gezeigt) ist die Verringerung des Einschwingweite im Vergleich zu dem Ergebnis des DIC-Reglers. Der Nachteil dieses Verfahrens liegt in der verhältnismäßig längere Rechenzeit.

5.3 Internal Model Control [IMC]:

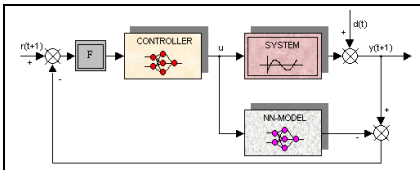


Fig 2: Interne Modellregelungsstruktur.

Die meisten Eigeschaften der interne Modellregelung entspricht der des direkten Inverseregelung. Zusätzlich weist die interne Modellregelung folgenden Eigenschaften noch:

VORTEILE:
- Offsetfreie Führungsverhalten für Systeme mit konstanter Laststörung.

NACHTEILE:
- Das System muß „open-loop" stabil sein,
- Es ist nicht einfach sicherzustellen, daß das Inversemodell an eine realistische „Data Set" trainiert wird wegen des Struktures des IMC,
- Notwendigkeit beide Vorwärts- und Inversemodell.

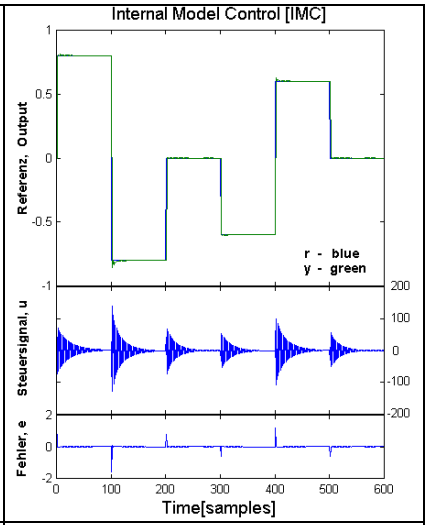


Bild 3: Systemleistung – Bsp. 2

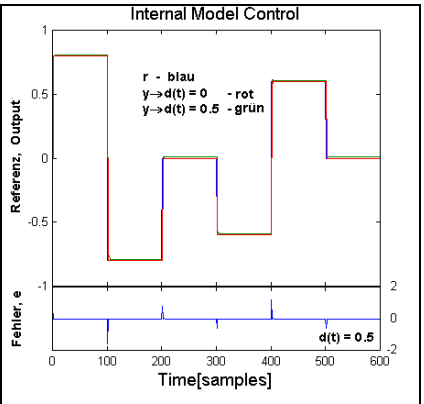


Bild 4:Verhalten bei Laststörungen

Basierend auf den Ergebnissen im Bild 3, erzeugt das IMC Regelungskonzept ein Führungsverhalten mit noch geringerer Einstellzeit als bei dem DIC- oder PID-Regelungskonzept. Bild 4 zeigt wie vermutet, daß dieses Regelungskonzept weniger empfindlich für Störungen ist, aber es gibt Probleme mit der Regelung von Systemen mit veränderlicher Struktur. Die Rechenzeit ist auch verhältnismäßig kurz.

5.4 Feedback-Error-Regelung [FEC]:

Das Konzept der Feedback-Error-Regelung [FEC] ist hervorzuheben. Es zeichnet sich durch hohe Regelgüte, verhältnismäßig kurze Rechenzeit, ein gutes Sprungantwortverhalten [Bild 5] sowie Robustheit gegenüber den untersuchten zeitvarianten Systemen in Bild 6 und unterdrückt auch Störungen. Das HAC-Regelungskonzept erzeugt ungefähr die gleichen Ergebnisse wie das FEC-Regelungskonzept, aber es ist verhältnismäßig zeitaufwendig (siehe Tabelle2) und verursacht Instabilitäten im System, wenn der PID-Regler nicht gut genug eingestellt ist.

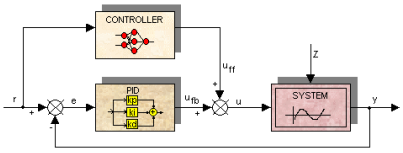


Fig 3: Feedback- Error-Regelungsstruktur.

VORTEILE:
- Einfach zu implementieren,
- verbessert das Führungsverhalten des Systems ohne dessen Sensitivität zu erhöhen,
- Einfach „Training Data" zu erstellen, da das System durch das Vorhandensein eines Feedback-Reglers stabil ist.

NACHTEILE:
- Braucht ein linear controller,
- kann nicht Störungen die auf das System einwirken kompensieren,
- Probleme, falls das Inversemodell des Systems instabil oder schlecht gedämpft ist,
- Ein schlechtes Inversemodell kann sorge das System stören statt es zu verbessern.

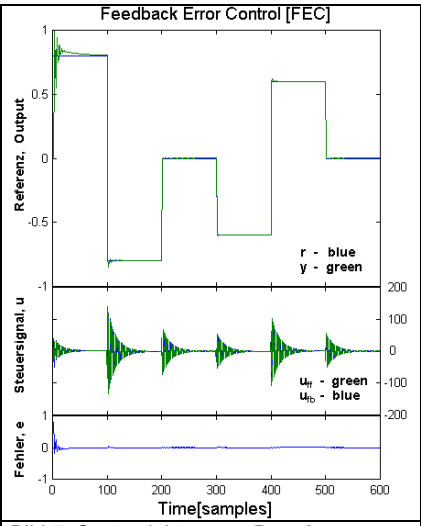


Bild 5: Systemleistung -> Bsp. 2

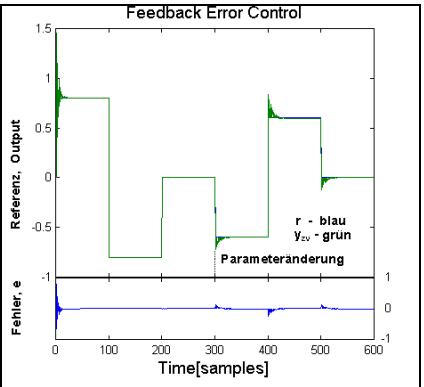


Bild 6: zeitvariantes System -> Bsp. 1

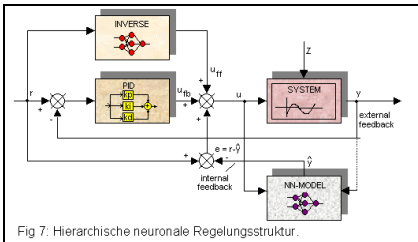


Fig 7: Hierarchische neuronale Regelungsstruktur.

VORTEILE:
s.h. Das Feedback-Error-Regelungskonzept.

NACHTEILE:
- Verursacht instabilitäten im System, falls der Feedback-Regler nicht gut genug eingestellt ist,
- Braucht beide Vorwärts- und Inversemodelle,
- Kann nicht Störungen die auf das System einwirken kompensieren.

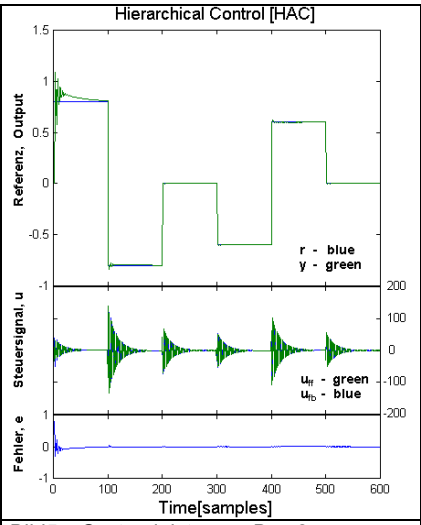


Bild5a: Systemleistung ->Bsp 2

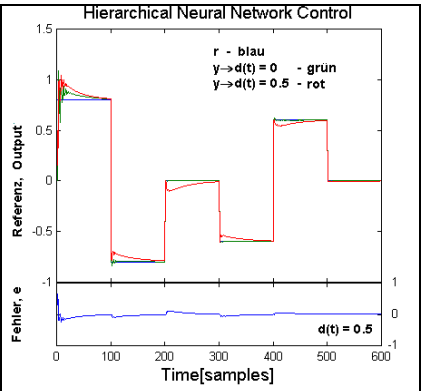


Bild6a: Verhalten bei Laststörungen

5.5 Optimal Control [OTC]:

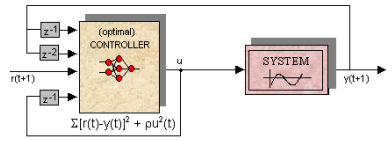


Fig 7: Optimal Control Struktur.

VORTEILE:

- Anwendbar für eine größere Klassen von Systemen im Vergleich zu DIC,
- zielgerichtet,
- Geeignet für Reglerentwurf für ein bestimmte
- Referenztrajektorie,
- Direkt anwendbar an zeitvarianten Systemen,

NACHTEILE:

- Controller-Netzwerk wird online trainiert,
- die Initialisierung des Netzwerkes erstellt sich als schwierig,
- das Netzwerk muß jedesmal trainiert werden, falls den Penalty Faktor modifiziert wird.

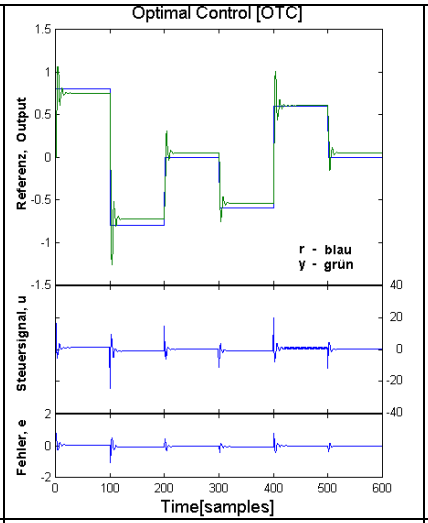


Bild 7: Systemleistung ->Bsp. 2

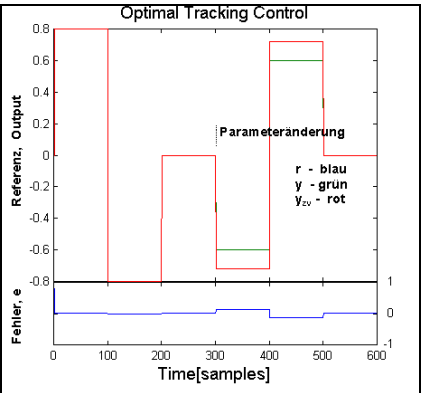


Bild 8: zeitvariantes System-> Bsp. 1

Das OTC-Regelungsprinzip zeigt zufriedenstellende Ergebnisse mit einem weniger aktiven Steuersignal wie in der Direkt-Inverse-Regelung. Leider ist festzustellen, daß ein Steady-State-Error bleibt. Weitere Untersuchungen (nicht hier gezeigt) haben ergeben, daß dieser von der Penalty-Faktor ρ abhängig ist. Dies kann vermieden werden durch Berücksichtigung der Differnz u(t)-u(t-1) statt nur u(t). Dieses neuronale Regelungskonzept ist ebenso wie das DIC, nicht zur Steuerung von systemen mit Laststörungen oder veränderlichen Struktur geeignet.

5.6 Nonlinear Predictive Control [NPC]:

Diese Methode ‚Nonlinear Predictive Control [NPC]' erzeugt recht gute offsetfreie Ergebnisse für alle Sprünge wie Bild 9 zeigt. Deutlich zu sehen ist auch die prädiktive Natur des Reglers, daß diesen schon Referenzsignaländerungen erwartet. Dieses Konzept erzeugt eine sehr glatte Steuersignal (nicht gezeigt hier). Leider, die Simulation ist verhältnismäßig zeitaufwendig (siehe Tabelle 2). Dies berührt auf dem Minimierungsalgorithmus, der in jeder Iteration laufen muß. Wahlweise man kann die Methode der arbeitspunktabhängige Linearisierung anwenden, um die Rechenzeit zu Reduzieren. Das APC-Regelungsprinzip ist einfach, verhältnismäßig schnell und erzeugt auch gute Ergebnisse wie Bild11 zeigt.

Der kleine Unterschied im Systemverhalten der beiden neuronale Regelungsprinzipien ist nicht nur durch die Linearisierung zu erklären sondern auch durch die Benutzung einer völlig andere Methode zur Berechnung der Vorhersagewerte. Der Unterschied zeigt sich deutlich in der Steuerung von Systemen mit Störungen und veränderlichen Struktur (Bild 12).

Für das APC-Regelungskonzept gilt das gleiche wie für die Internal-Model-Regelung. Es ist nicht empfindlich für Laststörungen, wie beim FEC-Konzept werden Parameterveränderungen im System unterdrückt.

Bild12 verdeutlicht, daß die Störungen beim APC-Regelungskonzept die Dämpfung des Systems beeinflussen.

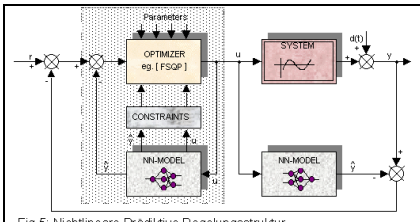


Fig 5: Nichtlineare Prädiktive Regelungsstruktur.

VORTEILE

- Anwendbar für eine größere Klassen von Systemen,
- Geeignet für Systeme mit Zeitverzögerungen,
- kann instabile oder Systeme mit instabile Inverse stabilisieren,
- Kompensiert bekannte Störungen,
- Möglichkeit die Beschrankungen in I/O Größen zu berücksichtigen,

NACHTEILE

- Probleme wegen lokale Minima,
- Rechenaufwendig,
- Ad-hocs können Konflikte mit Echtzeitanforderungen auslösen.

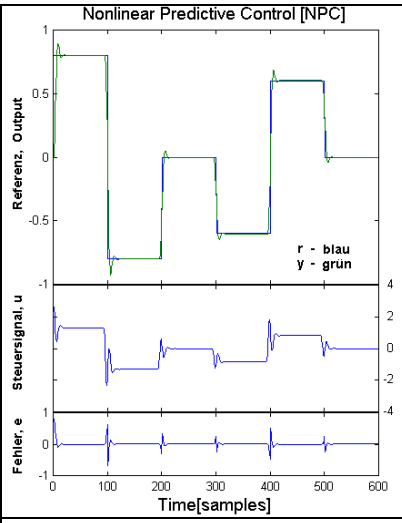


Bild 9: Systemleistung -> Bsp. 2

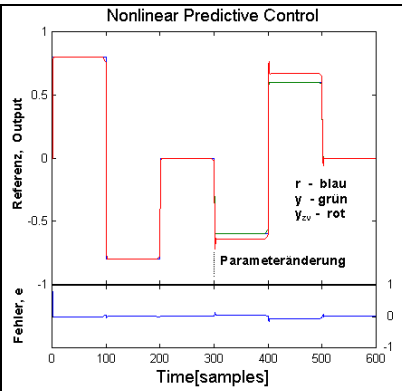


Bild10: zeitvariantes System-> Bsp. 1

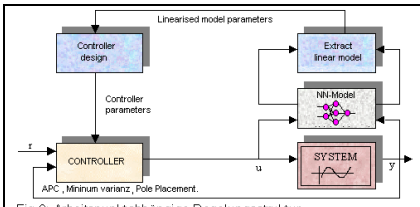


Fig 6: Arbeitspunktabhängige Regelungsstruktur.

VORTEILE

- Möglichkeit hochetabilierte linearer Reglerentwurfsmethoden anzuwenden,
- Implementierung zweckmäßig einfach,
- geeignet für echtzeitanwendungen,
- anwendbar an Systemen mit instabilen Inversen,
- Analyse des Verhaltens des geschlossenen Kreises und Regler ist einfach,
- Erzeugt wichtige Information über die Dynamik des Systems.

NACHTEILE

- Das linear Modell kann manchmal nur in einem engen Bereich um den Arbeitspunkt gültig sein.

APC Regelungsstruktur.

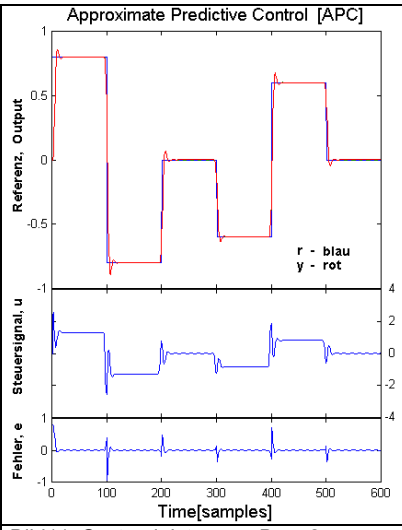


Bild11: Systemleistung -> Bsp. 2

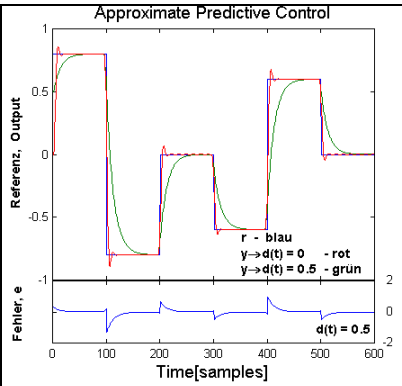


Bild12: Verhalten bei Laststörungen

5.7 Instantaneous Linearisation [INL]

Das neuronale INL-Regelungskonzept erzeugt auch sehr gute Ergebnisse, siehe Bild13-14. Leider ist festzustellen, daß die Methode der ‚Pole placement with zeros cancelled' ein sehr aktives Steuersignal erzeugt.

Dies ist damit zu erklären, daß die eliminierte Nullstelle in der Nähe von –1 ist. Und im Fall der ‚Pole placement without zeros cancelled' verhält sich das Steuersignal ruhig und das Antwortverhalten hat sich nicht viel verändert.

Da die Extrahierung des neuronalen Modells und die Linearisierung innerhalb eines Simulationsschritt erfolgt, ist dieses Konzept ist verhältnismaßig schnell und ist besonderes geeignet zur Anwendung in Echtzeitsystemen. Es verhält sich besser als

der PID-Regler wegen der Fahigkeit nächste Wertfolgen vorherzusagen. Deshalb ergeben auch die weichen Übergänge beim Referenzsignalwechsel.

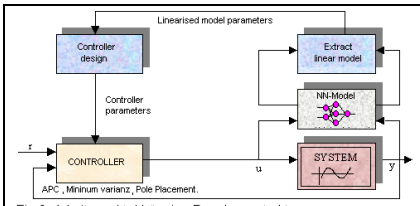


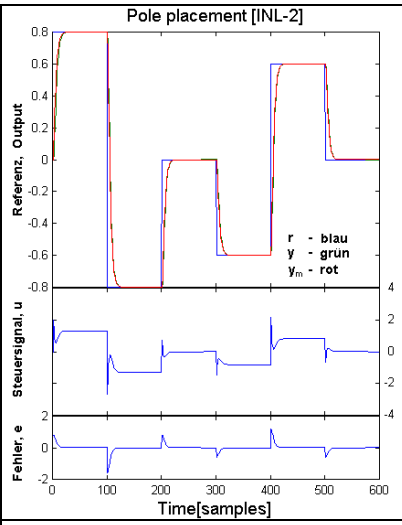
Fig 6: Arbeitspunktabhängige Regelungsstruktur.

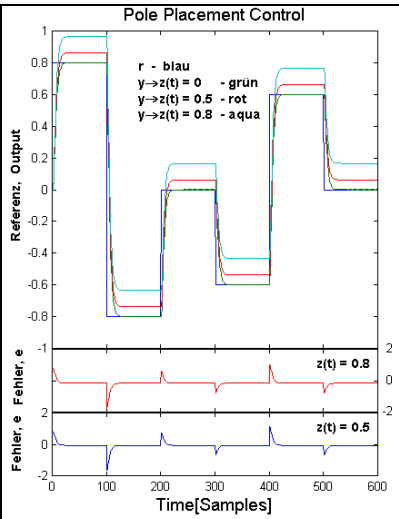
VORTEILE

- Möglichkeit hochetabilierte linearer Reglerentwurfsmethoden anzuwenden,
- Implementierung zweckmäßig einfach,
- geeignet für echtzeitanwendungen,
- anwendbar an Systemen mit instabilen Inversen,
- Analyse des Verhaltens des geschlossenen Kreises und Regler ist einfach,
- Erzeugt wichtige Information über die Dynamik des Systems.

NACHTEILE

- Das linear Modell kann manchmal nur in einem engen Bereich um den Arbeitspunkt gültig sein.





,Poleplacement' Regler | Bild13: Systemleistung-> Bsp. 2 | Bild14: Verhalten bei Eingangsstörungen

5.8 Feedback Linearisation Control [FBL]:

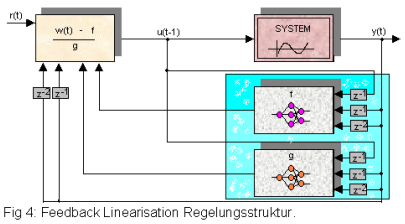


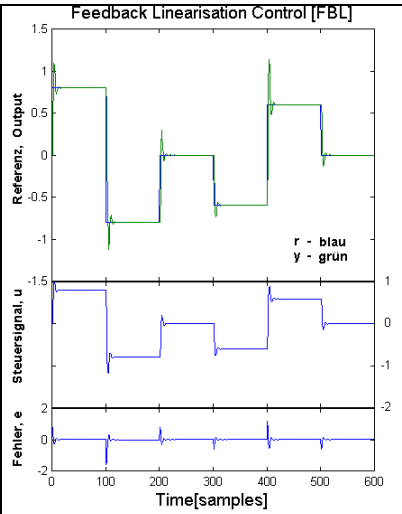
Fig 4: Feedback Linearisation Regelungsstruktur.

VORTEILE

- Einfach zu implementieren,
- Braucht nur ein Vorwartsmodell des Systems,
- Tuningmöglichkeit ohne das Modell neu zu trainieren,
- Möglichkeit ein ‚pole placement Regler' in einer externe Rückköpplung einzusetzen,

NACHTEILE

- Probleme, falls die inverse des Systems instabil oder in der Nähe der Stabilitätsgrenze liegt,
- Nicht einfach ein passende Modellstruktur auszuwählen, da zwei neuronale netzwerke ausgewählt werden müssen,
- beschränkt auf einem Typ von Systemen.



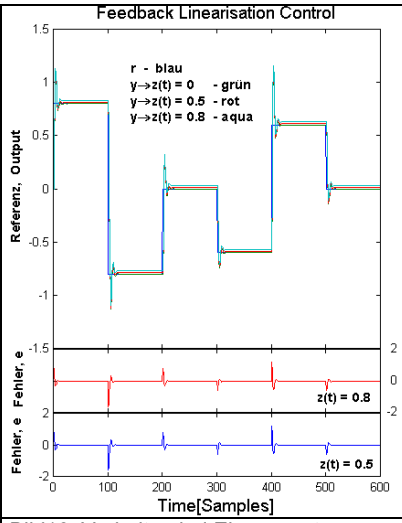


	Bild15: Systemleistung -> Bsp. 2	Bild16: Verhalten bei Eingangsstörungen

Das ‚Feedback Linearisation Control [FBL]' Regelungskonzept zeichnet sich durch schnellere Trainingszeiten der Modelle aus. Leider hat ihr Antwortvehalten (Bild15) eine längere Einstellzeit und mehr Überschwingungen als die restlichen neuronalen Regelungskonzepte. Was interessant ist, daß sogar diese Methode in unserem Fall bessere Ergebnisse als der des PID-Reglers erzeugt.

Wie Bild17 zeigt ist, die Einstellzeit des PID-Reglers ist sehr lang und die Übershwingweite groß. Alle Reglungskonzepte, die auf neuronalen Netzen basiere sind, erzeugen fine Steuersignale, was zur Schonung der Stellantriebe führ.

Dies ist durch die nichtlineare Eigenschaften der neuronalen Netzen zu erklären. Wegen des linearen Verhalten des PID-Reglers, sind die so erzeugten Steuersignale sehr abrupt und können deshalb zur Schädigung der Stellantriebe führen.

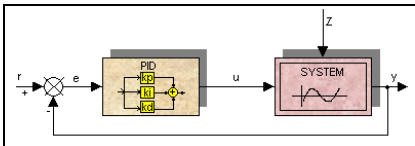


Fig 9: P-I-D Regelungskonzept.

VORTEILE:

- Einfach zu implementieren,
- Sehr bekannte Technik,
- braucht kein Streckenmodell.

NACHTEILE

- Probleme bei komplexerer nichtlinearer Systeme,
- Parametereinstellung ist nicht trivial,
- nicht Robust.

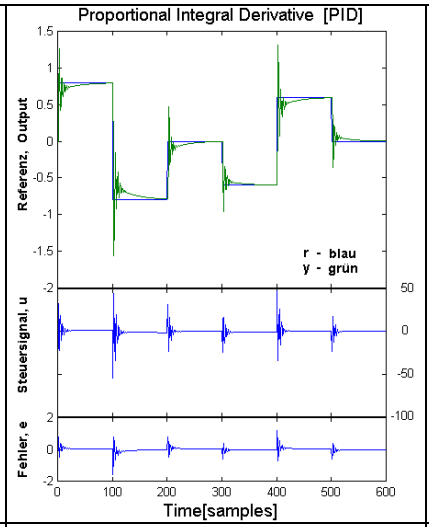


Bild17: Systemleistung→Bsp. 2

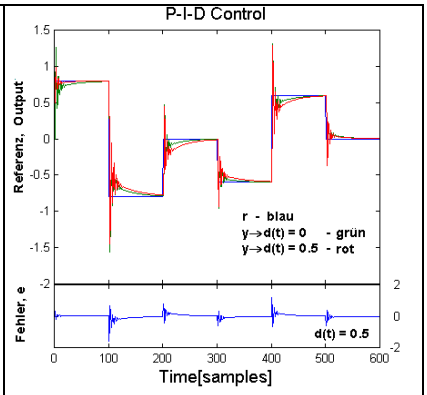


Bild18: Verhalten Bei Laststörungen

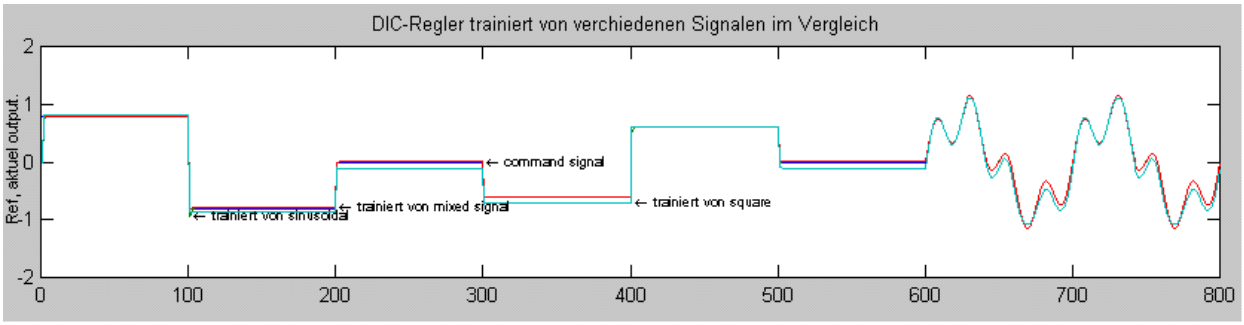


Bild19: DIC-Regler trainiert von verschiedener Signalen im Vergleich.

6.0 Zusammenfassung

Es wurden Konzepte für den Einsatz neuronaler Netzwerke zur Steuer- und Regelung dynamischer Systeme präsentiert und analysiert. Die Untersuchungen liefern Aussagen zur Adaptionsfahigkeit, Genauigkeit, Echtzeitfähigkeit und Robustheit. Eine quantitative Auswertung des systematischen Vergleichs der verschiedenen neuronalen Regelungskonzepten ist in Tabelle 2 dargestellt und die Untersuchungen des Sprungantwortverhaltens ist in Bild1-18 wiedergegeben. Daraus geht eine deutliche Verbesserung des Antwortverhaltens gegenüber konventionellen Methoden hervor. Hier profitiert die Regelungsgüte von der Adaptivität der Regler im jeweiligen Arbeitsbereich. Zudem tragen insbesondere die indirekten Regelungskonzepte mit rückgekoppelten Reglern zur Stabilisierung bei. Von den verschiedenen neuronalen Regelungskonzepten ist insbesondere das Feedback-Error-Regelungskonzept hervor zuheben, das sich durch eine hohe Regelgüte, ein gutes Sprungantwortverhalten sowie Robustheit gegenüber den untersuchten zeitvarianten Systemen auszeichnet. Neuronale Regelungskonzepte erzeugen glattere Stellsignale gegenüber dem PID-Regler, was zur Verlängerung der Lebensdauer der Stellantriebe führen könnte. Es zeigt sich, daß für bestimmte Zwecke bereits sehr kleine neuronalen Netzwerke zur Systemidentifikation und Regelung genügen, was eine echtzeitfähige Implementierung ermöglicht. Da theoretische Aussagen über die Stabilitätseigenschaften neuronale

Netzwerke kaum möglich sind, wird diese wichtige Eigenschaft auf der Grundlage empirischer Beobachtungen überprüft. Die Ergebnisse sind zufriedenstellend. Im Allgemeinen kann aus den Untersuchungen gefolgert werden, daß neuronale Netzwerke für nichtlineare Systeme niedrige Ordnung sehr effizient einsetzbar sind. Dabei werden für die untersuchten Systeme zahlreiche Annahmen getroffen, so etwa ein begrenztes Ausgabeverhalten und die Existenz inverser Systemoperatoren. Es konnte gezeigt werden, daß es prinzipiell möglich ist, auf der Basis der neuronalen Verarbeitungsprinzipien das typische Verhalten klassischer Reglerkonzepte zu erreichen. Von dieser Erkenntnis ausgehend, liegen die Vorteile des neuronalen Konzepts vor allem in:

- der nichtlinearen Gestaltung der Kennlinien und Kennflächen,
- den Möglichkeiten des Entwurfs von Mehrgrößenregelungskonzepten,
- der Nutzung effektiver arbeitspunktabhängiger Regelungen,
- der Nutzung nichtlinearer Methoden der dynamischen Optimierung,
- der Beeinflußbarkeit des Trainingsprozesses und demzufolge der Qualität des Reglers.

7. Referenzen:

[PSAL,87] PSALTIS, D. / SIDERIS, A. / YAMAMURA, A. : Neural Controllers. In: IEEE International Joint Conference on Neural Networks, Vol. IV (1987), S.551-558.

[JORD,89b] JORDAN M.I.: generic Constraints on unspecified Target Trajectories. In : International Joint Conference on Neural Networks, Vol. I(1989),S. 217-225.

[KAWA,87] KAWATO, M. / FURUKAWA K. / SUZUKI, R.: A Hierarchical Neural Network Model for Control and Learning of Voluntary Movement. In: Biological Cybernetics 57 (1987), S.169-185.

[MIYA,88] MIYAMOTO, H. / KAWATO, M. /SETOMAYA, T. / SUZUKI, R.: Feedback-error-Learning Neural Network for trajectory Control of Robotic Manipulators. In : Neural Networks, Vol.1 (1988), S.225-265.

[GARC,82] GARCIA, C.E. /MORARI, M. : Internal Model Control: 1. A Unifying Review and some New Results. In : Ind. Eng. Chem. Process Des. Dev., 21(1982), S.308-232.

[NARE,90a] NARENDRA. K.S/PARTHASARATHY,K. : indentification and Control of Dynamical Systems Using Neural Networks. In: IEEE Transactions on Neural Networks, Vol.1, No.1, March 1990, S.4-27.

[TZIR,91] TZIKEL-HANCOCK, E. /FALLSIDE, F. : Stable Control of nonlinear Systems Using Neural Networks. Technical Report, CUED/F-INFENG/TR.81, Cambridge University Engineering Department, Cambridge, UK(1991).

[WERB,90] Werbos, P.J.: Overview of Designs and Capabilites. In MILLER, W.T. /SUTTON, R.S. / WERBOS, P.J.(Hrsg.), Neural Networks for Control, Cambrige, USA (1990), S.59-65

[SAIN,91] SAINT-DONAT, J. /BHAT, N/ MCVOY, T.J. ; Neural Net Based Model Predictive Control. In: International Journal of Control 54(6) 1991, S.1453-1468.

[NGU,90a] NGUYEN, D / WIDROW, B. : Neural Networks for self-Learning Control Systems. In. IEEE Control Systems Magazine, 10(3) 1990.

[WIDR,87a] WIDROW, B./WINTER, R.G. / BAXTER, R.A.: Learning Phenomena in Layered Neural Networks. In: IJCNN International Joint Conference On Neural Networks, Vol. II (1987), S.145-157.

[BART,90] BARTO, A.G. : Connectionist Learning for Control. In: MILLER, W.T. / SUTTON, R.S. WERBOS, P.J. (Hrsg), Neural Networks for Control, MIT_Press, Cambridge, MA (1990), S.5-58.

[KOHO,91] KOHONEN, T. ET.AL. (hrsg): Artificial Neural Networks, Vol.1 &2, North Holland,

1991.

[RITT,89b] RITTER, H. /MARTINETZ, T./SCHLUTEN, K.: Ein Gehirn für Roboter – Wie neuronale Netzwerke Roboter steuern können. In: MC-Mikrocomp., Februar 1989.

[HORN,89] Hornik,K./Stinchcombe,M. /White, H.: Multilayered Feedforward Networks are Univaersal Approximators. In: Neural Networks, 2(5) (1989), S.359-366.

[NARE90b] Narenda, K.S.: Adaptive Control Using Neural Networks. In Miller, W.T. /Sutton,R.S. / Werbos, P.J. (hrsg), Neural Networks for Control, MIT –Press, Cambridge, MA (1990), S.115-142.

[JORD,89a] JORDAN, M.I./ROSENBAUM, D.A.: Action. In: Posner, M.I. (Hrsg) , Foundations of Cognitive Science, Cambridge, MA: MIT-Press (1989).

[WIDR,85] WIDROW, B. /STEARNS, S.D.: Adaptive Signal Processing. Englewood Cliffs, NJ: Prentice-Hall (1985).

[WIDR,85] WIDROW, B. /STEARNS, S.D.: Adaptive Signal Processing. Englewood Cliffs, NJ: Prentice-Hall (1985).

Modellierung eines virtuellen Kraftsensors mit neuronalen Netzen

Carsten Otto

Mess-, Steuer- und Regelungstechnik (Prof. Dr.-Ing. H. Schwarz)
Gerhard-Mercator-Universität Duisburg, D-47048 Duisburg
Tel.: +49 203 379-3423 ; Fax: +49 203 379-3027
E-Mail: co@uni-duisburg.de

Kurzfassung

Der Beitrag stellt die Modellierung eines virtuellen Kraftsensors mittels neuronaler Netze vor. Ausgangspunkt ist ein hydraulisch angetriebener zweiachsiger Roboter mit elastischem Arm, dessen Strukturschwingungen durch Regelung des Aktuators gedämpft werden. Dieses Regelungskonzept erfordert jedoch den Einsatz eines aufwendigen Sensors zur Messung der auf den Aktuator wirkenden Kraft. Auf Basis des mehrschichtigen Perzeptrons wird ein Black-Box-Modell identifiziert, dass im Eingang nur Prozessgrößen verwendet, die von ohnehin vorhandener Sensorik gemessen werden, und die gesuchte Kraft im Sinne eines virtuellen Sensors schätzt. Die Validierung des Modells erfolgt am realen System und schließt einen Vergleich mit bereits bestehenden Ansätzen ein.

1 Einleitung

Die Modellierung technischer Systeme mit neuronalen Netzen hat sich in den letzten Jahren auf dem Gebiet der Identifikation nichtlinearer dynamischer Systeme einen festen Platz erobert. Dieser parametrische Ansatz lässt sich in den Bereich der Black-Box-Modellbildung einordnen, die im Gegensatz zur klassischen Modellbildung kein oder nur sehr wenig physikalisches Wissen zur Erstellung eines Modells erfordert. Insbesondere für Anwendungen, bei denen bestimmte Prozessgrößen nur unter erheblichem Aufwand gemessen werden können oder deren Messung kostenintensiv ist, stellen neuronale Netze als virtuelle Sensoren oft eine Möglichkeit dar, diese Prozessgrößen zu schätzen.

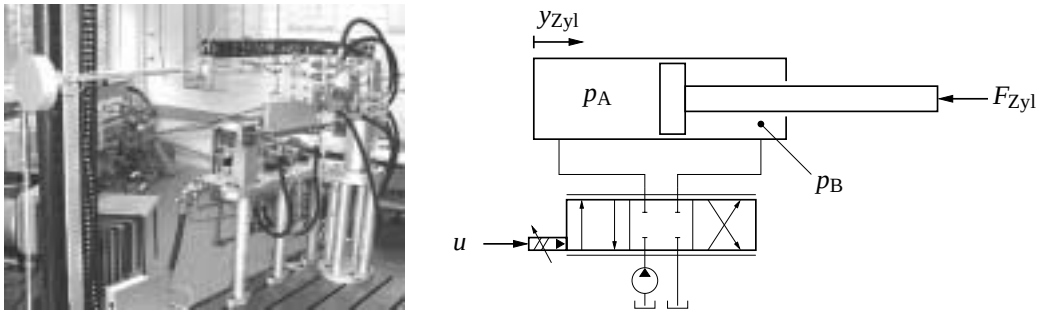
Der Beitrag stellt die Modellierung eines virtuellen Sensors zur Schätzung der auf einen Aktuator eines elastischen Roboters wirkenden Kraft vor. Als Black-Box-Modell werden neuronale Netze in Form des mehrschichtigen Perzeptrons verwendet, dass die Identifikation nichtlinearer dynamischer Ein-Ausgangs-Modelle erlaubt. Ausgangspunkt ist ein hydraulisch betriebener zweiachsiger Roboter mit elastischem Arm. Das zugrunde gelegte Regelungskonzept des Roboters prägt dem Aktuator das Ein-Ausgangs-Verhalten eines Feder-Dämpfer-Elementes auf, mit dem Ziel, die durch die elastische Verformung des Arms enstehenden Strukturschwingungen zu dämpfen. Dieses Konzept erfordert jedoch den mit erheblichem konstruktiven Aufwand verbundenen Einsatz eines kostenintensiven Sensors zur Messung der auf den Aktuator wirkenden resultierenden Kraft. Daher wird auf Basis des mehrschichtigen Perzeptrons ein Black-Box-Modell identifiziert, dass im Eingang nur Prozessgrößen verwendet, die von ohnehin vorhandener Sensorik gemessen werden. Als Modellausgang wird die geschätzte Kraft im Sinne eines virtuellen Sensors zur Verfügung gestellt. Die Validierung des Modells erfolgt am realen System

und beinhaltet zum einen den Vergleich zwischen gemessener und geschätzter Kraft und zum anderen wird der Aktuator unter Verwendung der geschätzten Kraft geregelt, um den Einfluss der Modellgüte auf die Regelung zu beurteilen.

Der Abschnitt 2 beschreibt den elastischen Roboter sowie das der Schwingungsdämpfung zugrunde liegende Regelungskonzept. Abschnitt 3 stellt die Methode der Systemmodellierung mit neuronalen Netzen vor und Abschnitt 4 enthält die Ergebnisse der Modellidentifikation und Validierung und diskutiert diese im Vergleich zu einem bestehenden Ansatz. Eine Zusammenfassung und ein Ausblick schließen diesen Beitrag ab.

2 Elastischer Roboter

Bei den in diesem Beitrag betrachteten Versuchsträger handelt sich um einen zweiachsigen Roboter, der aufgrund seiner schlanken Armsegmente eine ausgeprägte Elastizität besitzt und zur Nachbildung der Eigenschaften von Schwerlasthandhabungssystemen im Labormaßstab dient. In Bild 1(a) ist der Roboter in der hier zugrunde gelegten Referenzlage dargestellt, bei der sich der Arm in der horizontalen Ebene befindet, in der auch die Bewegung des Systems durch Drehung um die Hochachse stattfindet. Der Antrieb erfolgt



(a) Elastischer Roboter (b) Prinzipskizze des Aktuators

Bild 1: Versuchsstand elastischer Roboter

durch einen in Bild 1(b) skizzierten hydraulisch betriebenen Differentialzylinder mit den beiden Kammerdrücken p_A und p_B, der über ein Servoventil mit der Steuerspannung u angesteuert wird. Am Zylinderkolben greift die Kraft F_{Zyl} an. Weiterführende Informationen zum Versuchsstand und insbesondere zur Modellbildung des Differentialzylinders können [1] entnommen werden.

2.1 Schwingungsdämpfung

Bedingt durch die ausgeprägte Elastizität der Armsegmente kann es bei der Bewegung des Roboterarms zu erheblichen Strukturschwingungen kommen, die ein wesentliches regelungstechnisches Problem darstellen. Daher wird in [1] ein Regelungskonzept zur passiven Schwingungsdämpfung mittels virtueller Feder-Dämpfer-Elemente vorgestellt. Grundlage dieses Konzeptes in Bild 2 ist ein nichtlinearer Regler der dem Aktuator das

Verhalten eines Feder-Dämpfer-Elementes mit der Steifigkeit c und der Dämpfung d aufprägt. Diese Parameter sind frei wählbar und daher kann die Schwingungstilgung den jeweiligen Gegebenheiten angepasst werden. In diesem Beitrag betragen $c = 12\,000$ Nm^{-1} und $d = 5\,500$ Nsm^{-1} [2]. Die Bewegung des Roboters kann über die Verschiebung y_F

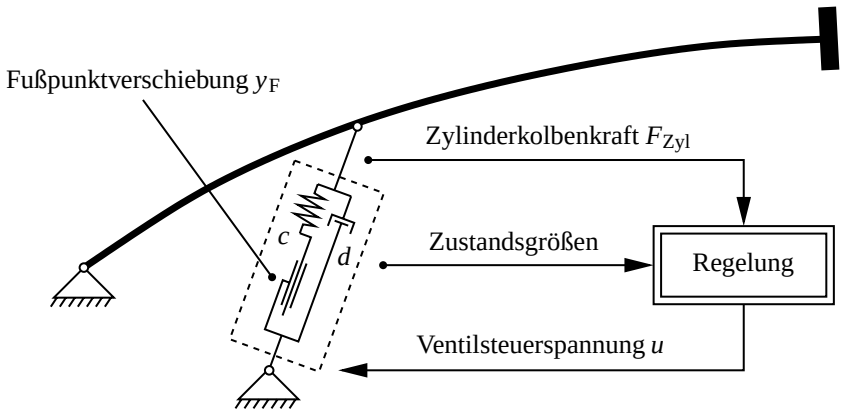


Bild 2: Elastischer Roboterarm mit virtuellem Feder-Dämpfer-Element

des Federfußpunktes erfolgen. Der wesentliche Vorteil dieses Konzeptes besteht darin, dass es aufgrund seiner Passivität ein System nicht destabilisieren kann und es daher auch robust gegenüber Modellungenauigkeiten ist.

2.2 Bestimmung der Zylinderkolbenkraft

Entscheidend für die Funktion des Regelungskonzeptes ist die Kenntnis der resultierenden Zylinderkolbenkraft F_{Zyl}, die im einfachen Fall mittels auf der Kolbenstange aufgebrachter Dehnungsmessstreifen bestimmt wird. Die robuste Erfassung der Kraft kann in der industriellen Praxis jedoch nur über einen in Bild 3 dargestellten Kraftsensor erfolgen, der zwischen der Kolbenstange und dem Armsegment eingebaut ist. Der Einsatz dieses Sensors ist zum einen mit intensiven Kosten verbunden und erfordert zum anderen einen erheblichen konstruktiven Aufwand, so dass eine Nachrüstung vorhandener Aktuatoren nicht wirtschaftlich ist. Aus diesen Gründen macht es Sinn, über alternative Möglichkeiten zur Schätzung der Zylinderkolbenkraft nachzudenken.

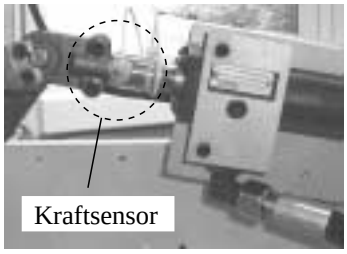


Bild 3: Kraftsensor

In [2] wird das Kräftegleichgewicht am Kolben, dass von der Trägheitskraft $m_{ges}\ddot{y}_{Zyl}$, der Druckdifferenzkraft $F_{\Delta p}$, der Reibkraft F_R sowie der Zylinderkolbenkraft F_{Zyl} abhängt, betrachtet:

$$m_{ges}\ddot{y}_{Zyl} = F_{\Delta p} - F_R - F_{Zyl} \quad . \tag{1}$$

Unter der Voraussetzung kleiner Beschleunigungen $\ddot{y}_{Zyl}$ wird die Trägheitskraft vernachlässigt, so dass sich als Approximation

$$\tilde{F}_{Zyl} = F_{\Delta p} - F_R \tag{2}$$

ergibt. Die Druckdifferenzkraft ist unter Messung der Zylinderkammerdrücke p_A und p_B und mittels der Zylinderkolbenfläche A_K sowie des Flächenverhältnisses φ durch

$$F_{\Delta p} = A_K \left(p_A - \frac{p_B}{\varphi} \right) \tag{3}$$

zu berechnen. Die Reibkraft ergibt sich in [2] über die im allgemeinen als Stribeck-Kurve bekannte Funktion zur Approximation geschwindigkeitsabhängiger Reibung [3]. In den folgenden Abschnitten wird nun die Möglichkeit aufgezeigt, die Zylinderkolbenkraft mittels neuronaler Netze zu schätzen. Hierbei dient Gleichung (2) zum Vergleich der Ergebnisse untereinander.

3 Systemmodellierung mit neuronalen Netzen

Künstliche neuronale Netze, oft auch als neuronale Netze bezeichnet, lassen sich als informationsverarbeitende Systeme beschreiben, die aus einfachen Einheiten, den Zellen oder Neuronen bestehen, welche über gewichtete Verbindungen miteinander kommunizieren [4]. Die Anfänge neuronaler Netze liegen zu Beginn der vierziger Jahre und intensive Forschungen reichten bis zum Ende der sechziger Jahre. Nachdem das Interesse in den siebziger Jahren nahezu erlosch, erlebt das Gebiet der neuronalen Netze eine Renaissance, die eng mit der Arbeit [5] verbunden ist. Heutzutage haben neuronale Netze einen breiten Anwendungsbereich, beispielsweise in der Mustererkennung, der Klimaprognose oder in der Medizintechnik [6]. Insbesondere zeigen jüngste Arbeiten [7], [8], [9] und [10], dass neuronale Netze auf dem Gebiet der Modellbildung und Regelung ebenfalls stark vertreten sind.

3.1 Das mehrschichtige Perzeptron

Das mehrschichtige Perzeptron (MLP, *engl.* multilayer perceptron) ist einer der bekanntesten Vertreter neuronaler Netze. Diese Netzarchitektur stellt eine nichtlineare Abbildung aus einem Eingangsraum $\mathbb{R}^n$ in einen Ausgangsraum $\mathbb{R}^m$ dar und eignet sich als universeller Approximator funktionaler Zusammenhänge [11]. Die einzelnen Zellen des MLP-Netzes sind in einer oder mehreren verdeckten Schichten und einer Ausgangsschicht angeordnet und mit den Zellen benachbarter Schichten über gewichtete Verbindungen vernetzt. Die Informationsweitergabe ist vom Eingang über die verdeckten Schichten hin zur Ausgangsschicht gerichtet.

In Bild 4 ist die Struktur einer Zelle j in der Schicht s des MLP-Netzes dargestellt. Diese erhält die Ausgangssignale $y_{s-1,i}$ der n Zellen der vorhergehenden Schicht und berechnet mit Hilfe der gewichteten Verbindungen w_{ij} das Aktivierungssignal

$$v_{s,j} = w_{0j} y_{s-1,0} + \sum_{i=1}^{n} y_{s-1,i} w_{ij} \quad , \tag{4}$$

sowie die Aktivierung $a_{s,j}$ über die Aktivierungsfunktion $g(v_{s,j})$, die als Ausgang $y_{s,j}$ an die Zellen der nachfolgenden Schicht weitergeleitet wird. Die Zellen des MLP-Netzes besitzen einen Schwellenwert, der als Signal $y_{s-1,0}$ mit konstantem Wert $y_{s-1,0} = 1$ und dem zugehörigen Gewicht w_{0j} realisiert ist [6].

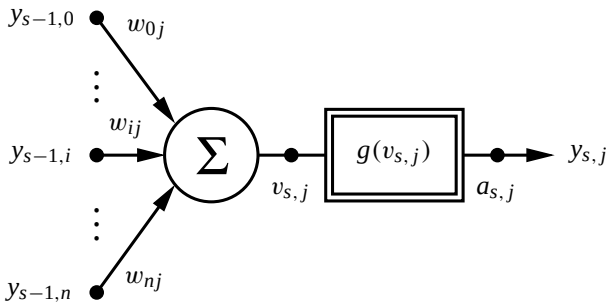


Bild 4: Struktur einer Zelle

Dieser Beitrag verwendet MLP-Netze mit einer verdeckten Schicht und einer einzelnen Zelle in der Ausgangsschicht (Bild 5). Die Zellen der verdeckten Schicht besitzen den hyperbolischen Tangens

$$g(v_{s,j}) = \tanh v_{s,j} \quad , \tag{5}$$

und die Ausgangszelle besitzt die Identität als Aktivierungsfunktion. Der Ausgang $\hat{y}$ des Netzes ergibt sich damit als Superposition der einzelnen Ausgänge der Zellen der verdeckten Schicht.

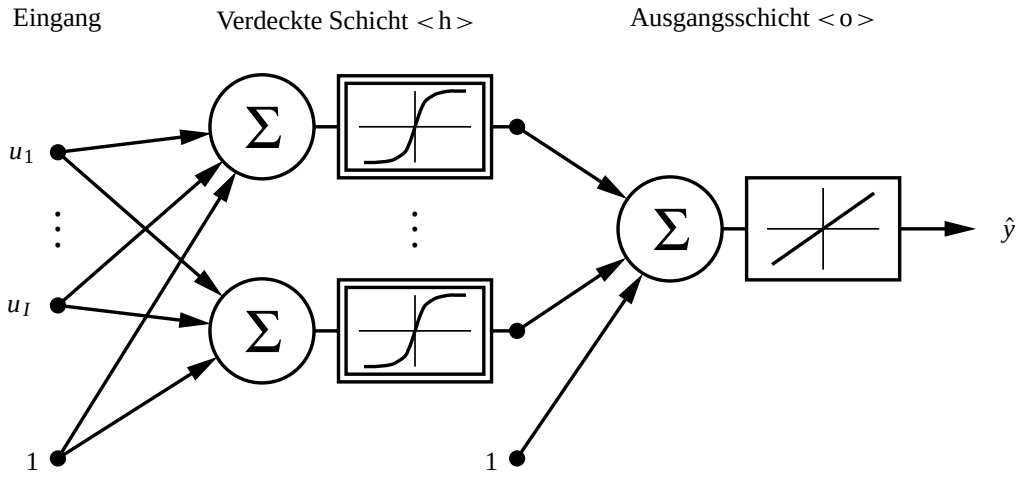


Bild 5: Zweischichtiges MLP-Netz

Der funktionale Zusammenhang zwischen Ausgang $\hat{y}$ und Eingang $\boldsymbol{u} = [u_1, \ldots, u_I]$ in Abhängigkeit der Gewichte $\boldsymbol{w}$ ergibt sich zu

$$\hat{y}(\boldsymbol{u}, \boldsymbol{w}) = w_{01}^{<\mathrm{o}>} + \sum_{j=1}^{H} w_{j1}^{<\mathrm{o}>} \tanh\left(w_{0j}^{<\mathrm{h}>} + \sum_{i=1}^{I} w_{ij}^{<\mathrm{h}>} u_i \right) \quad , \tag{6}$$

wobei I die Anzahl der Eingänge und H die Anzahl der Zellen der verdeckten Schicht beschreiben. Die hochgestellten Indizes $<\mathrm{h}>$ und $<\mathrm{o}>$ deuten die Zugehörigkeit des jeweiligen Gewichtes zur verdeckten bzw. zur Ausgangsschicht an.

Während einer Trainingsphase erlernt das MLP-Netz den funktionalen Zusammenhang zwischen Ein- und Ausgang anhand eines Messdatensatzes. Zur Bewertung des Lern-

erfolges wird die Kostenfunktion

$$V = \frac{1}{N} \sum_{p=1}^{N} \left(y_p - \hat{y}_p \right)^2 \tag{7}$$

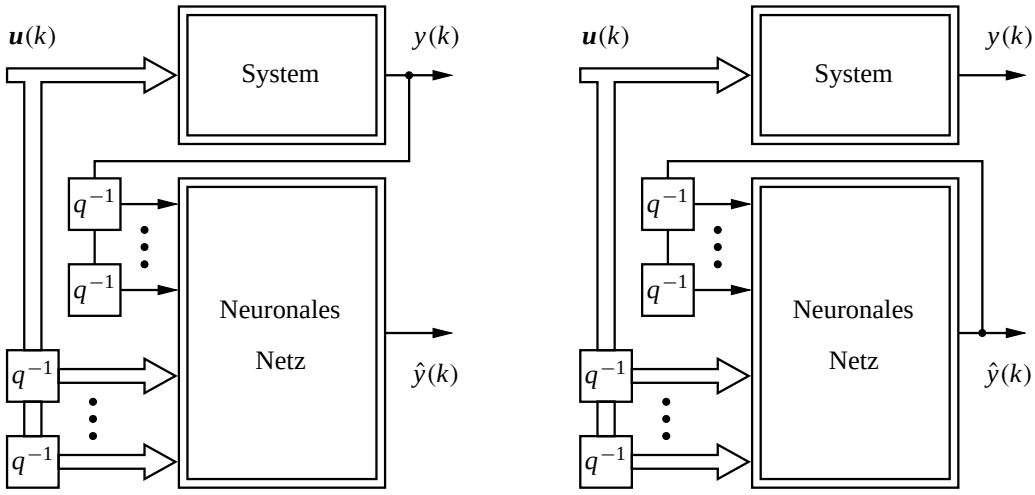
verwendet, die den quadratischen Fehler zwischen dem gemessenen Ausgang y_p und dem geschätzten Ausgang des MLP-Netzes $\hat{y}_p$ berechnet und über alle N Messdaten aufsummiert. Das Ziel von Lernverfahren für das MLP-Netz besteht somit in der Minimierung der Kostenfunktion durch Adaption der Gewichte $\boldsymbol{w}$. Hierzu wird in diesem Beitrag das Levenberg-Marquardt-Verfahren [7] benutzt.

3.2 Modellierung nichtlinearer dynamischer Systeme

Die Modellierung nichtlinearer dynamischer Systeme erfordert die Erweiterung der zuvor beschriebenen statischen MLP-Netze mit Speicherelementen. Die in diesem Beitrag angewandte Methode besteht darin, zum einen die Eingänge $\boldsymbol{u}(k)$ des zugrunde liegenden Systems zeitlich zu verzögern und zum anderen den Ausgang extern zurückzuführen. Die dadurch entstehenden Ein-/Ausgangsmodelle können in der Form

$$\hat{y}(k) = f(\boldsymbol{\varphi}(k), \boldsymbol{w}) \tag{8}$$

beschrieben werden, wobei der Regressionsvektor $\boldsymbol{\varphi}(k)$ die zeitlich verzögerten Ein- und Ausgänge enthält und $f(\cdot)$ den durch das MLP-Netz repräsentierten funktionalen Zusammenhang (6) darstellt. Das dynamische Verhalten des Modells wird dabei wesentlich durch die Art der Rückkopplung des Ausgangs auf den Modelleingang bestimmt.



(a) Seriell-paralleles Modell

(b) Paralleles Modell

Bild 6: Black-Box-Modelle

Prinzipiell können der am System gemessene Ausgang y oder der vom Modell prädizierte Ausgang $\hat{y}$ genutzt werden. Im letztgenannten Fall besteht vollständige Parallelität zwischen System und Modell (Paralleles Modell in Bild 6(b)), wohingegen im erstgenannten Fall nur bezüglich des Eingangs $\boldsymbol{u}$ Parallelität herrscht (Seriell-paralleles Modell

in Bild 6(a)). Darüberhinaus ist die Unterscheidung wichtig, in welcher Konfiguration ein Modell identifiziert bzw. zum Einsatz kommt, denn ein parallel verwendetes Modell kann durchaus in seriell-paralleler Konfiguration identifiziert werden und umgekehrt.

Vor diesem Hintergrund finden sich in der Literatur zur Systemidentifikation nichtlinearer dynamischer Ein-/Ausgangsmodelle [12] zahlreiche Klassen, die eine Erweiterung der Klassen für lineare Systeme [13] darstellen. Zwei häufig anzutreffende Klassen, die sich direkt dem seriell-parallelen bzw. dem parallelen Ansatz zuordnen lassen, sind das NARX-Modell mit

$$\boldsymbol{\varphi}(k) = [\boldsymbol{u}(k-1) \ \cdots \ \boldsymbol{u}(k-m) \ \cdots \ y(k-1) \ \cdots \ y(k-m)]^{\mathrm{T}} \tag{9}$$

sowie das NOE-Modell mit

$$\boldsymbol{\varphi}(k) = [\boldsymbol{u}(k-1) \ \cdots \ \boldsymbol{u}(k-m) \ \cdots \ \hat{y}(k-1) \ \cdots \ \hat{y}(k-m)]^{\mathrm{T}} \tag{10}$$

als Regressionsvektor, wobei hier nur zur Vereinfachung der Notation für alle Ein- und Ausgänge die gleiche Ordnung m verwendet wird.

Im Hinblick auf den Einsatz des neuronalen Netzes als virtueller Kraftsensor muss das parallele NOE-Modell (10) verwendet werden. Denn das seriell-parallele NARX-Modell hängt von der Messung des Systemausgangs ab und erlaubt lediglich eine Einzelschrittprädiktion. Aufgrund der rekurrenten Struktur gestaltet sich jedoch die Identifikation eines NOE-Modells und die damit einhergehende Gradientenberechnung der Optimierung wesentlich aufwendiger, wohingegen beim NARX-Modell keine dynamische Optimierung notwendig ist und im Allgemeinen Modelle von höherer Güte identifiziert werden. Dennoch hat sich in [14] gezeigt, dass es für Einsatz eines NOE-Modells sinnvoll ist, auch zur Identifikation bereits den parallelen Ansatz zu verwenden.

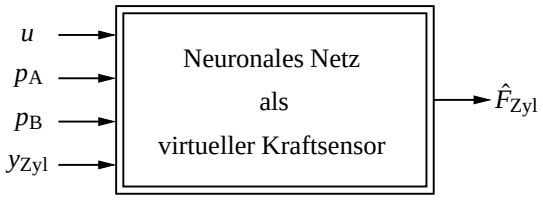
4 Modellierung des Kraftsensors

Im Folgenden wird die Modellierung des virtuellen Kraftsensors für den in Abschnitt 2 beschriebenen elastischen Roboter vorgestellt. Zunächst werden Signale zur Anregung des Federfußpunktes erzeugt, die zur Identifikation bzw. Validierung des Modells zur Schätzung der Zylinderkolbenkraft dienen. Im Anschluss daran werden die Ergebnisse der Identifikation und Validierung des Modells präsentiert, wobei ein Vergleich mit einem bereits bestehenden Ansatz eingeschlossen ist. Abschließend wird die vom virtuellen Sensor geschätzte Zylinderkolbenkraft in die Regelung mit einbezogen, um den Einfluss auf die Güte der Schwingungsdämpfung zu beurteilen.

4.1 Modellansatz

Zur Modellierung eines virtuellen Sensors für die Zylinderkolbenkraft stellt sich zunächst die Frage nach der Wahl geeigneter Eingangsgrößen. Diese müssen zum einen im Ursache-Wirkungs-Zusammenhang mit der Zylinderkolbenkraft stehen und zum anderen sollte die Messung dieser Größen durch ohnehin vorhandene Sensorik abgedeckt sein. Die Ergebnisse zur analytischen Modellbildung des hydraulischen Differentialzylinders in [1] zeigen, dass die Steuerspannung u des Servoventils, die Drücke p_{A} und p_{B} in den beiden Zylinderkolbenkammern sowie die Position y_{Zyl} des Zylinderkolbens diese beiden

Voraussetzungen erfüllen. Damit ergibt sich der in Bild 7 dargestellt Modellansatz. Die Black-Box-Modellstruktur, die ebenfalls in Bild 7 angegeben ist, wurde auf empirische Weise ermittelt und führt aufgrund des einfachen Ansatzes zur einem Modell mit geringer Parameterzahl. Das neuronale Netz besitzt $H = 3$ Zellen innerhalb der verdeckten Schicht, so dass für den hier gewählten Ansatz ein Modell mit 22 Parametern entsteht.

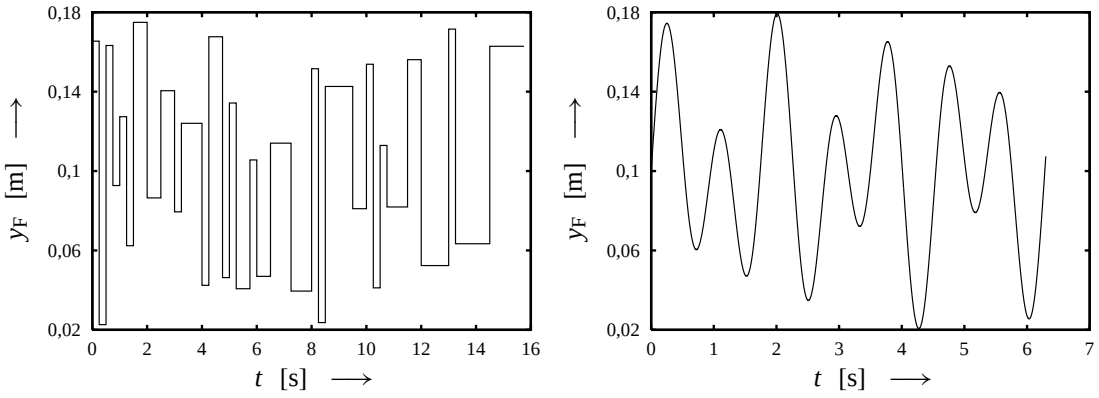


$$\hat{F}_{\mathrm{Zyl}}(k) = f\left(\hat{F}_{\mathrm{Zyl}}(k-1), u(k-1), p_{\mathrm{A}}(k-1), p_{\mathrm{B}}(k-1), y_{\mathrm{Zyl}}(k-1)\right)$$

Bild 7: Modellansatz des virtuellen Kraftsensors

4.2 Erzeugung von Testsignalen

Bei der Identifikation dynamischer Systeme hat die Art der Anregung des zugrunde liegenden Prozesses wesentlichen Einfluss auf die Güte der entstehenden Modelle, wobei sich Testsignale bewährt haben, die die Eigenschaften von weißem Rauschen annähern. Für lineare Systeme finden häufig pseudostochastische binäre Testsignale Verwendung, die mittels rückgekoppelter Schieberegister erzeugt werden und zwei Amplitudenwerte annehmen können [15]. Zur Identifikation nichtlinearer dynamischer Systeme bietet es sich an, ein binäres Signal mit einem normalverteilten Signal zu überlagern, so dass ein amplitudenmoduliertes pseudostochastisches binäres Testsignal mit verschiedenen Amplitudenwerten entsteht.



(a) Identifikation (b) Validierung

Bild 8: Anregung des Federfußpunktes

Die Messung der Ein-/Ausgangsdaten zur Identifikation des Kraftsensormodells nach Bild 7 erfolgt unter Anregung des Federfußpunktes des virtuellen Feder-Dämpfer-Elementes mit dem in Bild 8(a) dargestellten Testsignal. Hierbei liegen diesem pseudostochastischen Signal ein sechsstufiges Register sowie eine Taktzeit von $T = 250$ ms zu-

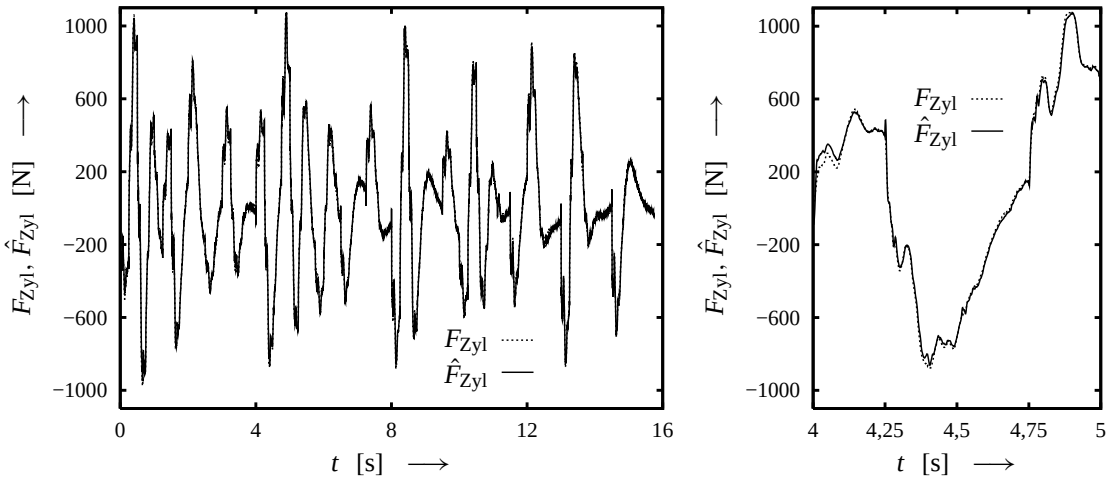
grunde. Zur Validierung des Modells wird der Federfußpunkt durch ein harmonisches Signal angeregt, welches aus zwei überlagerten Sinus-Schwingungen gebildet wird:

$$y_{\mathrm{F}}(t) = 0{,}1 + 0{,}03 \cdot \sin(0{,}4 \cdot t) + 0{,}05 \cdot \sin(0{,}7 \cdot t) \tag{11}$$

In Bild 8(b) ist der Verlauf dieses Signals dargestellt.

4.3 Identifikationsergebnisse

Im Folgenden werden die Ergebnisse der Identifikation des Kraftsensormodells vorgestellt. In Bild 9 sind der gemessene sowie der mit dem Modell simulierte Verlauf der Zylinderkolbenkraft dargestellt. Anhand des Bildes 9(a) ist zu erkennen, dass das Mo-



(a) Gesamter Messdatensatz

(b) Vergrößerung

Bild 9: Schätzung der Zylinderkolbenkraft

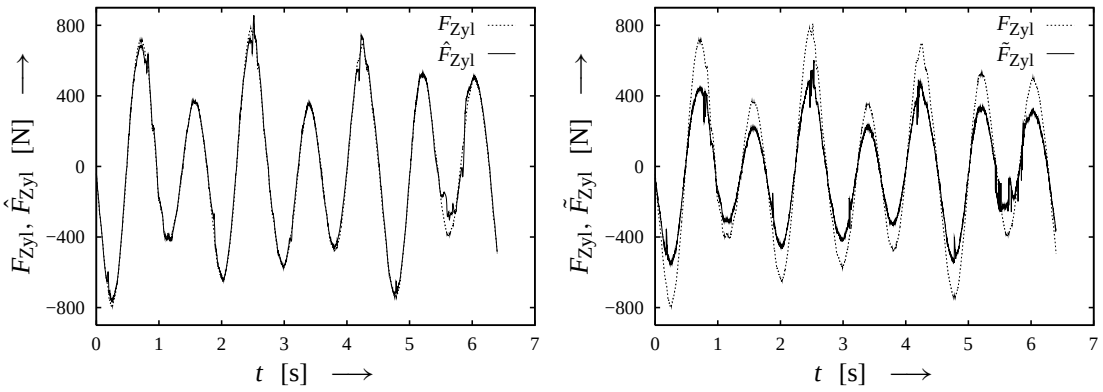
dell die Kraft im gesamten Verlauf sehr gut schätzt. Der zeitliche Ausschnitt in Bild 9(b) zeigt darüberhinaus die hohe Güte der Schätzung im Detail. Der mittlere Absolutfehler zwischen Messung und Schätzung beträgt für die Identifikation $\bar{E} = 14{,}01$ N bei einer Standardabweichung von $\sigma_E = 14{,}52$ N.

4.4 Validierungsergebnisse

Die Validierung des Kraftsensors erfolgt nicht in der Rechnersimulation, sondern wird real am Versuchsstand unter Anregung des Federfußpunktes mit dem in Bild 8(b) dargestellten harmonischen Signal durchgeführt. Allerdings wird die geschätzte Zylinderkolbenkraft zunächst nicht im Regelungskonzept zur Schwingungsdämpfung benutzt, sondern weiterhin die mit dem realen Kraftsensor gemessene Kraft.

In Bild 10(a) ist der Verlauf der gemessenen sowie der geschätzten Zylinderkolbenkraft für die Validierung dargestellt. Es ist zu erkennen, dass das neuronale Netz auch in der Validierung sehr gute Ergebnisse erzielt. Dies wird durch den mittleren Absolutfehler von

$\bar{E} = 21{,}22\,\mathrm{N}$ bei einer Standardabweichung von $\sigma_E = 24{,}96$ N bestätigt. Der globale Verlauf der Kraft wird ausgezeichnet wiedergegeben, lediglich in der Umkehrpunkten des Kraftverlaufes sind Abweichungen zu entdecken. Diese Abweichungen resultieren aus dem sogenannten Stick-Slip-Effekt im Bereich kleiner Zylinderkolbengeschwindigkeiten, die insbesondere in den Umkehrpunkten der schwingenden Roboterarmbewegung auftreten.



(a) Neuronales Netz (b) Approximation

Bild 10: Kraftverläufe bei Regelung mit gemessener Kraft

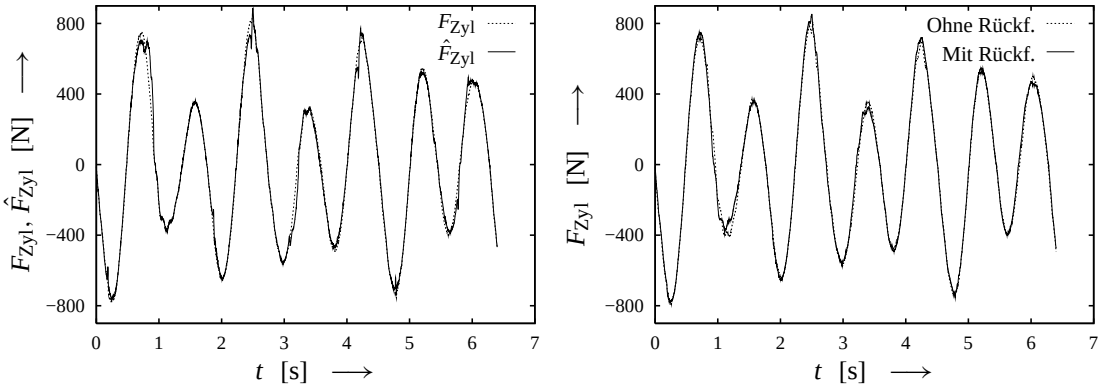
Zum Vergleich der Schätzergebnisse des neuronalen Netzes mit den Ergebnissen der Vorgehensweise in [2] ist in Bild 10(b) die gemessene Zylinderkolbenkraft der Approximation nach Gleichung (2) gegenübergestellt. Hierbei ist die Approximation mit einem mittleren Absolutfehler von $\bar{E} = 120{,}1\,\mathrm{N}$ bei einer Standardabweichung von $\sigma_E = 72{,}43$ N behaftet. Es zeigt sich, dass die Modellierung mit neuronalen Netzen eine wesentlich höhere Güte erreicht. Dies ist dadurch zu erklären, dass in der Approximation der Anteil der Trägheitskräfte vernachlässigt wird, dieser bei der Identifikation eines Modells aber implizit berücksichtigt wird. Zudem sind Ungenauigkeiten in der Modellierung der geschwindigkeitsabhängigen Reibkraft vorhanden.

4.5 Rückführung der geschätzten Kraft

Abschließend wird der Aktuator des elastischen Roboterarms unter Verwendung der vom neuronalen Netz geschätzten Kraft geregelt, um den Einfluss der bei der Schätzung verursachten Fehler auf die Funktion der Regelung zu beurteilen. Hierzu wird wiederum der Federfußpunkt des virtuellen Feder-Dämpfer-Elementes mit dem in Bild 8(b) dargestellten Testsignal angeregt. Als Kriterium zur Beurteilung der Modellgüte bietet es sich an, die Verläufe der gemessenen Kraft mit und ohne Rückführung der geschätzten Kraft in das Regelungskonzept zur vergleichen. Der Unterschied zwischen diesen beiden Verläufen muss dabei möglichst gering sein. Je weniger dieses Kriterium erfüllt ist, um so mehr verschlechtert sich auch die Güte der Schwingungsdämpfung, da diese maßgeblich von der Qualität des Kraftsignals abhängt.

Betrachtet man die Verläufe von gemessener und geschätzter Kraft in Bild 11(a) unter Verwendung der geschätzten Kraft im Regelungskonzept, so zeigen sich wiederum nur

geringfügige Unterschiede. Dies verdeutlicht sich auch im mittleren Absolutfehler von $\bar{E} = 21{,}37\,\mathrm{N}$ bei einer Standardabweichung von $\sigma_E = 33{,}42$ N. Darüberhinaus sind in Bild 11(b) die Verläufe der gemessenen Zylinderkolbenkraft mit und ohne Rückführung der geschätzten Kraft in das Regelungskonzept dargestellt. Es zeigt sich, dass beide Verläufe sehr gut übereinstimmen und die geringen Abweichungen, die durch einen mittleren Absolutfehler von $\bar{E} = 20{,}06\,\mathrm{N}$ bei einer Standardabweichung von $\sigma_E = 19{,}29$ N gekennzeichnet sind, auf die Güte der Regelung zur Schwingungsdämpfung keinen spürbaren Einfluss haben.



(a) Neuronales Netz

(b) Vergleich der gemessenen Kräfte

Bild 11: Kraftverläufe bei Regelung mit geschätzter Kraft

5 Zusammenfassung

Der Beitrag demonstriert die Modellierung eines virtuellen Sensors zur Schätzung der auf einen Aktuator eines elastischen Roboters wirkenden Kraft mittels neuronaler Netze. Aufgrund der ausgeprägten Elastizität der Armsegmente enstehen infolge der Bewegung Strukturschwingungen, die durch Regelung des hydraulischen Aktuators gedämpft werden. Dieses Konzept erfordert jedoch den mit erheblichem konstruktiven Aufwand verbundenen Einsatz eines kostenintensiven Sensors zur Messung der auf den Aktuator wirkenden resultierenden Kraft. Daher wird auf Basis des mehrschichtigen Perzeptrons ein Black-Box-Modell identifiziert, dass im Eingang nur Prozessgrößen verwendet, die von ohnehin vorhandener Sensorik gemessen werden. Als Modellausgang wird die geschätzte Kraft im Sinne eines virtuellen Sensors zur Verfügung gestellt.

Die Identifikationsergebnisse zeigen, dass das Modell sehr gut in Lage ist die Zylinderkolbenkraft zu schätzen. Darüberhinaus bestätigt die Validierung des Modells am realen System seine Eignung als virtueller Sensor. Im Vergleich zu einer bestehenden Arbeit, in der die Zylinderkolbenkraft über einen auf physikalischem Wissen basierenden Ansatz approximiert wird, erzielt das in diesem Beitrag vorgestellte Modell deutlich bessere Ergebnisse. Die Gründe liegen in den komplexen physikalischen Zusammenhängen, beispielsweise bei der Bestimmung der Reibkräfte, die in der Approximation nicht ausreichend genau modelliert werden können. Insbesondere wird die Güte des auf neuronalen Netzen basierenden Modells deutlich, wenn die geschätzte Kraft im Regelungskon-

zept verwendet wird: Weder die angestrebte Schwingungsdämpfung noch der gemessene Kraftverlauf weichen wesentlich von der Vorgabe mit einem realen Kraftsensor ab. Als Nachteil der in diesem Beitrag vorgestellten Methode ist zu nennen, dass zur Identifikation eines Modells zunächst immer die Messung der Zylinderkolbenkraft notwendig ist. Dieser Nachteil ließe sich aber durch die Verwendung von Dehnungsmessstreifen abschwächen, weil diese geringere Kosten und wenig konstruktiven Aufwand erfordern, aber zu Identifikationszwecken ausreichend geeignet sind.

Als Ausgangspunkt für weiterführende Arbeiten ist die Sicherstellung der Zuverlässigkeit des Modells mittels anderer Testsignale zu betrachten. Insbesondere steht die Robustheit des Modells bei Änderung der Steifigkeit und Dämpfung des virtuellen Feder-Dämpfer-Elementes im Mittelpunkt des Interesses. Zudem kann versucht werden, das Modell im Bereich kleiner Zylinderkolbengeschwindigkeiten, in denen die Haftreibung einen wesentlichen Einfluss hat, zu verbessern.

Literatur

[1] Bernzen, Werner: *Zur Regelung elastischer Roboter mit hydrostatischen Antrieben*. Düsseldorf : VDI Verlag, 1999 (Fortschritt-Berichte VDI Reihe 8 Nr. 788)

[2] Polzer, Jan ; Nissing, Dirk: Mechatronic design using flatness-based control to compensate for a lack of sensors. In: *Preprints of the 1st IFAC Conference on Mechatronic Systems*. Darmstadt, 2000

[3] Schwarz, Helmut: *Einführung in die Systemtheorie nichtlinearer Regelungen*. Aachen : Shaker Verlag, 1999

[4] Zell, Andreas: *Simulation neuronaler Netze*. München : R. Oldenbourg Verlag, 1997

[5] Rumelhart, David E. ; Hinton, Geoffrey E. ; Williams, Ronald J.: Learning representations by back-propagating errors. In: *Nature* 323 (1986), Nr. 9, S. 533–536

[6] Haykin, Simon: *Neural Networks*. Upper Saddle River, New Jersey, USA : Prentice Hall, 1999

[7] Nørgaard, Magnus: *System Identification and Control with Neural Networks*. Lyngby, Dänemark, Department of Automation, Technical University of Denmark, Dissertation, 1996

[8] Bunke, Jörg: *Künstliche Neuronale Netze zur Systemidentifikation aus gestörten Meßwerten*. Düsseldorf : VDI Verlag, 1997 (Fortschritt-Berichte VDI Reihe 8 Nr. 667)

[9] Schultz, Jörg: *Identifikation nichtlinearer dynamischer Systeme mit Künstlichen Neuronalen Netzen*. Düsseldorf : VDI Verlag, 1998 (Fortschritt-Berichte VDI Reihe 8 Nr. 721)

[10] Junge, Thomas F.: *„On-line"-Identifikation und lernende Regelung nichtlinearer Regelstrecken mittels neuronaler Netze*. Düsseldorf : VDI Verlag, 1999 (Fortschritt-Berichte VDI Reihe 8 Nr. 807)

[11] Hornik, Kurt ; Stinchcombe, Maxwell ; White, Halbert: Multilayer feedforward networks are universal approximators. In: *Neural Networks* 2 (1989), S. 359–366

[12] Sjöberg, Jonas [u. a.]: Nonlinear black-box modeling in system identification: A unified overview. In: *Automatica* 31 (1995), Nr. 12, S. 1691–1724

[13] Ljung, Lennart: *System Identification: Theory for the User*. Upper Saddle River, New Jersey, USA : Prentice Hall, 1999

[14] Otto, Carsten: Modeling a hydraulic drive using neural networks. In: *Proceedings of the 3rd IMACS Symposium on Mathematical Modelling*. Wien, Österreich, 2000, S. 551–555

[15] Wernstedt, Jürgen: *Experimentelle Prozeßanalyse*. Berlin : VEB Verlag Technik, 1989

Mehrpunktregelungen mit Neuro-Fuzzy-Systemen am Beispiel einer adaptiven elektronischen Endlagendämpfung von Pneumatikzylindern

Werner Brockmann, Jens Köhne
Medizinische Universität zu Lübeck
Ratzeburger Allee 160
23538 Lübeck
brockman@iti.mu-luebeck.de

Kurzfassung

Fuzzy- und Neuro-Fuzzy-Systeme werden gern für die Approximation nichtlinearer Funktionen eingesetzt, die mit formalen Methoden nicht oder nur schwer zu beschreiben sind. Aufgrund ihrer internen Arbeitsweise sind sie für kontinuierliche Abbildungen prädestiniert. Sie werden daher selten für Anwendungen eingesetzt, bei denen diskrete Ausgaben gefordert werden, wie z.B. Mehrpunktregelungen. Beim Einsatz von Klassifikatoren wie Neuronale Netzen oder im einfachsten Fall eines Fuzzy-Systems mit nachgeschalteter Schwelle ist das Systemverhalten wegen eines fehlenden oder schlechten intuitiven Zugangs ist das Systemverhalten schwierig handzuhaben und ein sicheres Verhalten des Automatisierungssystems schwierig oder gar nicht zu garantieren. Hinzu kommen Probleme beim Lernen hinsichtlich einer fein abgestuften und schnellen Konvergenz in adaptiven oder selbsteinstellenden Systemen. In diesem Beitrag wird eine erste Machbarkeitsstudie vorgestellt, die am Beispiel der elektronischen Endlagendämpfung von Pneumatikzylindern zeigt, daß auch kontinuierlich arbeitende Fuzzy- und Neuro-Fuzzy-Systeme einfach und zielgerichtet in Anwendungen mit diskreten Stellgrößen eingesetzt werden können. Bei dem vorgestellten System zur Endlagendämpfung wird großer Wert auf eine flexible Einsetzbarkeit in unterschiedlichsten Systemkonfigurationen und auf ein sicheres Systemverhalten auch und gerade während der Adaption gelegt.

1 Einleitung

1.1 Problemstellung

Eine ganze Anwendungsklasse fand bisher im Bereich der Fuzzy-Regelungen wenig Beachtung, nämlich Prozesse, die mit diskret arbeitenden Stellgliedern angesteuert werden, wie z.B. Zwei-, Dreipunktregelungen. Dabei ist es egal, ob es sich dabei um geregelte oder gesteuerte Prozesse handelt. Sie werden hier allgemein unter dem Begriff „Mehrpunktregelungen“ zusammengefaßt. Standardentwurfsverfahren für Regelungen, wie z.B. für PID-Regler, sind nicht für Mehrpunktregelungen anwendbar. Es werden vielmehr spezielle Entwurfsverfahren benötigt. Sie sind allerdings i.d.R. so kompliziert, daß sie nur von Spezialisten angewendet werden können. Wie bei anderen formalen Entwurfsverfahren, z.B. für kontinuierliche Regelungen, stößt aber auch der formale Entwurf von Mehrpunktregelungen bei Prozessen, die schlecht zu modellieren sind, an seine Grenzen. Vorhandenes Expertenwissen von Anlagenfahrern oder Prozeßexperten (i.F. Experte genannt) kann dann nicht genutzt werden, weil die Verfahren für ihn nicht einsichtig sind.

Wie bei Fuzzy-Regelungen könnten modellfreie Methoden der Computational Intelligence genutzt werden, um in einem nicht formalen, wissensbasierten Entwurf das Erfahrungswissen zu nutzen und in ein Fuzzy-System zu übertragen. So ließe sich Vorwissen nutzen und ein sicheres, d.h. kontrollierbares Systemverhalten erzielen, wenn die Eigenschaften eines Fuzzy-Systems dem nicht entgegenstünden. Aufgrund ihrer internen Arbeitsweise sind sie nämlich für die Approximation kontinuierlicher nichtlinearer Abbildungen prädestiniert. Es lassen sich daher nicht direkt diskrete Ausgangsgrößen erzeugen, wie sie zur Ansteuerung von diskret arbeitenden Stellgliedern benötigt werden. Folglich steht dem Ziel eines modellfreier Entwurfs und einer modellfreien Adaption ein Dilemma gegenüber, denn Fuzzy- (und Neuro-Fuzzy-) Methoden sind die einzige Möglichkeit, mit nicht-formalen Methoden ein sicheres Systemverhalten für Anwendungen zu garantieren, bei denen formale Methoden nicht oder nur schwer angewendet werden können.

Dieser Problemkreis wird im folgenden noch näher erläutert und dann ein Beispiel vorgestellt, mit dem gezeigt wird, daß das Dilemma mit Hilfe eines Fuzzy-Systems gelöst werden kann und daß sich eine Mehrpunktregelung sogar mit Neuro-Fuzzy-Methoden flexibel an ein zeitvariantes Prozeßverhalten anpassen kann. Dadurch ist ein solches System durch Online-Lernen auch für die Adaption bzw. Selbsteinstellung selbst für sicherheitskritische Prozesse geeignet. In diesem Beitrag kann leider nur sehr allgemein auf die Lösung eingegangen werden, weil zur Zeit noch Verhandlungen mit einem Interessenten aus der Industrie laufen.

1.2 Diskussion bekannter Verfahren

Prinzipiell handelt es sich bei einer Mehrpunktregelung um eine Abbildung von kontinuierlichen Eingangsgrößen auf eine (oder mehrere) diskrete Ausgangsgröße[1], die k diskrete Schaltzustände (k = 2, 3, ...) annehmen kann. Grundsätzlich müssen also von der Mehrpunktregelung Entscheidungen gefällt werden, in welchen Situationen welche diskrete Ausgangsgröße auszugeben ist. Aus Sicht der geforderten Systemeigenschaften ist dies eine Klassifikations- und keine Funktionsapproximationsaufgabe, wobei jeder der k Schaltzustände einer Klasse entspricht. Bei der Klassifikation erfolgt eine Partitionierung des Eingangsraums und eine Zuordnung eines diskreten (Schalt-) Zustands der Ausgangsgröße (Klasse, Cluster) zu den einzelnen Partitionen. Die Kontrolle dynamischer Prozesse erfordert i.allg. eine Berücksichtigung des dynamischen Verhaltens. Daher sind bei dynamischen Systemen oft viele Eingangsvariablen zu verarbeiten. Dadurch entsteht ein hochdimensionaler Eingangsraum, in dem die Partitionen ebenfalls hochdimensional sind und i.d.R. eine entsprechend komplexe Form haben.

Methoden aus dem großen Gebiet der Datenanalyse können i.d.R. bei Regelungen nicht angewendet werden, weil die Aufgabenstellung eine grundsätzlich andere ist. Es geht nicht um die Extraktion von Wissen durch die Analyse gegebener Datenbestände, sondern umgekehrt um die Spezifikation einer Regelung für dynamisches System, das meist noch nicht oder nur unzureichend vorher geregelt wurde. Deshalb stehen keine

[1] Im folgenden wird ohne Beschränkung der Allgemeinheit bei der Beschreibung von nur einer Ausgangsgröße ausgegangen. Die Erläuterungen sind aber direkt auf Systeme mit mehreren Ausgangsgrößen übertragbar, was auch bei dem betrachteten Beispiel geschieht.

(sinnvollen) Daten zum Erlernen einer Regelung zur Verfügung. Es muß vielmehr Wissen aus ganz anderen Quellen eingebracht werden, im hier diskutierten Fall Expertenwissen. Das Nachbilden des Verhaltens eines Experten ist nur sehr eingeschränkt dafür geeignet, weil viele Zusammenhänge zwischen den Prozeßgrößen und seinen Stelleingriffen implizit sind und gerade bei diskret arbeitenden Systemen, die für eine Mehrpunktregelung eingesetzt werden, systembedingt Schwierigkeiten beim Lernen auftreten. Daher ist die explizite Spezifikation über Regeln zu bevorzugen.

Würde man Mehrpunktregelungen durch Expertenwissen mittels regelbasierter Systeme spezifizieren, werden in den Prämissen die Partitionsgrenzen beschrieben und in den Konklusionen die jeweilige Klassenzuordnung. Die einzelnen Partitionen sind dann, selbst wenn sie zusammenhängend sind, durch ihr komplexe, hochdimensionale Form nicht mehr geschlossen in der Prämisse einer einzelnen Regel zu beschreiben. Eine Partition muß deshalb aus mehreren Regeln zusammengesetzt werden. Bei hochdimensionalen Problemen entstehen so sehr viele Regeln. Der explodierende Regelraum macht es auch schon bei kleineren Problemstellungen schwierig, die einzelnen Klassengrenzen gezielt in den Prämissen der Regeln anzugeben. Deshalb werden oft lernende Klassifikatoren eingesetzt.

Die klassischen Varianten lernender Klassifikatoren sind statistische Verfahren und Neuronale Netze. Beide Verfahren haben in automatisierungstechnischen Anwendungen eine Reihe von Grenzen und Nachteilen, insbesondere wenn sie in einem geschlossenen Regelkreis eingesetzt werden sollen. Da kein Vorwissen explizit eingebracht werden kann, müssen Lerndaten zur Verfügung stehen. Das Lernergebnis kann dann aber nur hinsichtlich dieser Lerndaten und ggf. bezüglich eines vorgegebenen Testdatensatzes überprüft werden. Das erlaubt generell nur Aussagen über die Reproduktionsfähigkeiten und nur eingeschränkt über die Verallgemeinerungsfähigkeiten. Eine absolute Aussage über die Verallgemeinerungsfähigkeiten ist nicht möglich, weil keine Interpretierbarkeit des Systemverhaltens gegeben ist. Dadurch ist es schwierig, die Stabilität zu garantieren, wodurch der Einsatz beider klassischen Verfahren zumindest in sicherheitskritischen Anwendungen fraglich ist.

Lernende Fuzzy- bzw. Neuro-Fuzzy-Verfahren zur Klassifikation (Übersicht in [1]) würden zwar das Einbringen von Vorwissen in Mehrpunktregelungen erlauben, so daß sich das System (von Anfang an) annähernd wie gewünscht verhält. Doch durch die hohe Dimensionalität der Eingangsräume entsteht wie oben beschrieben nach wie vor ein Komplexitäts- und damit auch ein Engineering-Problem bei der Handhabbarkeit der Regelbasen. Dieser Knowledge-Engineering-Bottleneck betrifft nicht nur das Ad hoc-Wissen, sondern auch das Feintuning und ggf. die Interpretierbarkeit gelernten Wissens. Dadurch ist ebenfalls die handhabbare Komplexität beschränkt, insbesondere wenn die Sicherheit des Systems kritisch ist.

Selbst wenn sich das Engineering-Problem lösen ließe, kommen bei lernenden (adaptiven oder selbsteinstellenden) Mehrpunktregelungen weitere Schwierigkeiten hinzu, die eine schnelle Konvergenz (Anzahl erforderlicher Lerndaten, Adaptionsgeschwindigkeit) und die Sicherheit beim Lernen (insbes. beim Online-Lernen im geschlossenen Regelkreis) betreffen. Zum einen muß i.d.R. gleich zu Beginn des Betriebs zumindest ein stabiles Verhalten sichergestellt sein. Das geht nur, wenn ausreichend Vorwissen im System vorhanden ist. Zum anderen wird durch das Online-Lernen zur Adaption das Systemverhalten ständig dynamisch geändert. Das darf nicht dazu führen, daß während des Lernens unsichere Zustände auftreten. Das

Systemverhalten muß also wie bei kontinuierlichen Regelungen auch während des Lernens kontrolliert werden können. Im Gegensatz zu kontinuierlichen Systemen kann bei Mehrpunktregelungen aber schon eine kleine Änderung durch das Lernen die Zuordnung zu einer anderen Klasse zur Folge haben. Das kann das Systemverhalten drastisch ändern, bis hin zur Instabilität. Deshalb müssen die Lernvorgaben für die Adaption wohl bedacht sein. Das betrifft beispielsweise Fragen wie: wann wird eine Änderung vorgenommen und zu welcher neuen Klasse wird gewechselt.

Dabei sind für kontinuierlich arbeitende Neuro-Fuzzy-Systeme Verfahren bekannt, die online auch an komplexen, sicherheitskritischen Prozessen für eine adaptive Regelung eingesetzt werden können (z.B. [2,3,4]). Sie würden sich daher als Ausgangsbasis auch für Mehrpunktregelungen anbieten. Der Kernpunkt einer wissensbasierten und möglichst auch lernfähigen Abbildung kontinuierlicher Eingangsgrößen auf diskrete Ausgangsgröße erfordert aber eine andere Vorgehensweise. Das es auch in der Praxis funktionieren kann, zeigt das folgende Beispiel.

2 Anwendungsbeispiel

2.1 Pneumatisches Positionieren

Als Anwendungsklasse, um Mehrpunktregelungen in der Praxis anzuwenden und wie hier insbesondere als adaptive Regelung zu untersuchen, bietet sich der Bereich des pneumatischen Positionierens mit Schaltventilen an, denn Pneumatik ist mit klassischen Methoden nur sehr aufwendig zu modellieren und zu beherrschen. Das betrifft insbesondere das dynamische Verhalten der Luft beim Strömungsaufbau und -abbau in den Schläuchen und den Ventilen sowie das Verhalten der Ventile und Zylinder. Bei Schaltventile treten zusätzliche Probleme durch Totzeiten der Ventile auf, die abhängig von den aktuellen Druck- und Strömungsverhältnissen variieren. Außerdem bewirkt die Kompressibilität der Luft eine arbeitspunktabhängige Steifigkeit des Systems. Gerade im Zusammenspiel mit dem Stick-Slip-Effekt, der aus der ausgeprägten Haftreibung des Kolbens resultiert, macht das das Systemverhalten schwierig zu kontrollieren.

Die meisten der genannten Effekte sind stark nichtlinear und extrem zeitvariant. Das bereitet große Schwierigkeiten bei der Modellierung und beim Entwurf geeigneter Regelungen. So ist i.d.R. eine permanente Adaption an das Systemverhalten erforderlich, die bei klassischen Methoden mit einem sehr hohen Rechenaufwand verbunden ist und eine teure Mikroelektronik (Signalprozessor) nach sich zieht. Da die pneumatische Antriebstechnik aber sehr kostensensitiv ist, besteht ein Bedarf an alternativen Lösungen. Eine solche soll nachfolgend exemplarisch an einem einfachen Beispiel vorgestellt werden.

Die konventionelle Art, mit Pneumatik zu positionieren, verwendet Proportionalventile als Stellglieder. Oft ist bereits eine Druckregelung integriert, so daß die beiden Kammerdrücke bzw. der Differenzdruck als Stellgröße fungieren. In anderen Fällen wird zur Einstellung des Luftstroms die Lage des Kolbenschiebers im Proportionalventil vorgegeben und teilweise von einem integrierten System geregelt. Jedoch haben Proportionalventile den Nachteil, relativ teuer zu sein. Durch ihren 3/2-Wege- bzw. 5/3-Wege-Charakter weisen sie außerdem eine Querströmung auf, die zu einem unnötig erhöhten Luftverbrauch führt. Deshalb ist der Einsatz von Schaltventilen

beim pneumatischen Positionieren erstrebenswert und trotz der geringeren zu erwartenden Genauigkeit für viele Anwendungen ausreichend.

2.2 Endlagendämpfung von Pneumatikzylindern

Um grundsätzliche Methoden zum Einsatz von (Neuro-)Fuzzy-Systemen in Mehrpunktregelungen an einer konkreten, realen Anwendung aus dem Bereich des pneumatischen Positionierens zu erarbeiten, wurde vor diesem Hintergrund die elektronische Endlagendämpfung von Pneumatikzylindern als erstes einfaches Beispiel ausgewählt. Das Ziel dabei ist eine möglichst schnelle Bewegung von einer Endlage zur anderen. Dabei ist die kinetische Energie der bewegten Masse weitestgehend abzubauen, bevor der Kolben auf einen Endanschlag trifft. Durch die schonendere Bewegung werden Stöße und dadurch die mechanischen Belastungen und Verschleiß am Zylinder, am angetriebenen System selbst sowie am Transportgut minimiert.

Die Endlagendämpfung wird konventionell mechanisch realisiert. Die eine Variante besteht aus kleinen Luftpolstern mit variabler Auslaßöffnung, die an beiden Enden im Zylinder integriert sind und durch den bewegten Kolben kurz vor Erreichen der Endposition zusammengedrückt werden. Durch den Widerstand, der der ausströmenden Luft entgegengesetzt wird, wird die kinetische Energie der bewegten Masse abgebaut. Die Auslaßöffnung läßt sich allerdings nur für eine bewegte Masse optimal einstellen, und das auch nur durch (aufwendiges) Probieren. Ist die bewegte Masse größer, entstehen zu große, belastende Stöße im System. Ist die bewegte Masse kleiner, dauert das Erreichen der Endposition länger als nötig. Dadurch wird in vielen Anwendungen die erreichbare Taktrate und damit der Output z.B. von Verpackungsmaschinen unzulässig reduziert. Das gleiche gilt für separate Dämpferelemente, die außerhalb des Zylinders am angetriebenen System abgebracht sind. Bei ihnen kommt nachteilig hinzu, daß ein größerer Bauraum erforderlich wird und daß die Dämpferelemente einem Verschleiß unterworfen sind.

Die genannten Nachteile zu vermeiden und gleichzeitig eine schnellere Bewegung zu ermöglichen, um z.B. die erreichbare Taktrate zu steigern, ist das Anliegen einer elektronischen Endlagendämpfung der Firma Festo [5]. Dieses „Soft Stop“ genannte System arbeitet mit einer Erfassung der Position des Kolbens und einem 5/3-Wege-Proportionalventil, das den Luftstrom zu beiden Kammern stellt. Für die Bewegung von einem Endanschlag zum anderen wird eine feste Geschwindigkeitsrampe abgefahren und über einen PD-Regler geregelt. Die Parametrierung des Reglers erfolgt über ein Tabellenwerk, aus dem abhängig von den verwendeten Komponenten (Zylinder, bewegte Masse, Ventil) und der Einbaulage des Antriebssystems (horizontal, vertikal) die Stellung von DIP-Schaltern abgelesen und am Steuergerät eingestellt wird. Zur Kompensation der zeitvarianten Effekte erfolgt eine ständige Adaption im laufenden Betrieb.

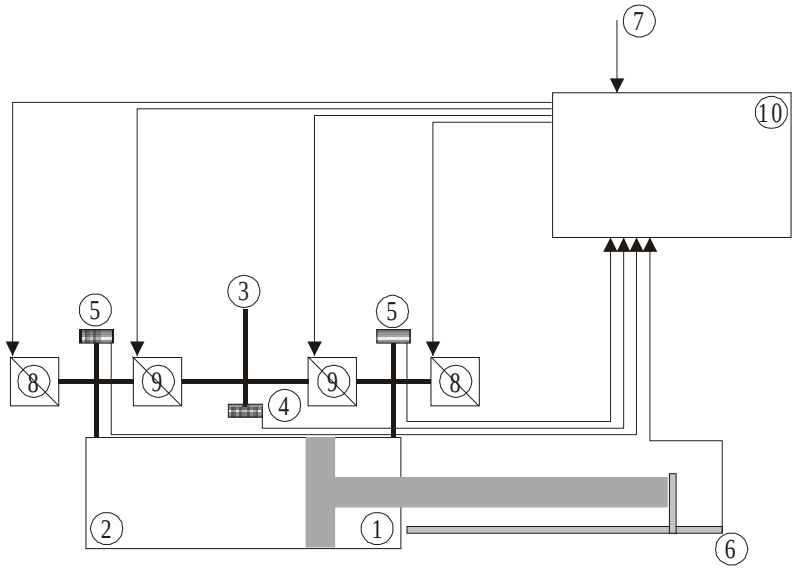
Durch diese Vorgehensweise ist allerdings nur eine eingeschränkte Auswahl an Konfigurationsmöglichkeiten für das Antriebssystem (Zylinder, Ventile usw.) möglich. Außerdem wird eine symmetrische Verschlauchung vorausgesetzt. Eine wesentliche Grenze des Soft Stop-Systems stellt die eingeschränkte Variation der bewegten Masse dar, denn es sind nur Änderungen um +/- 30% zulässig, wahrscheinlich aus Stabilitätsgründen. Bei vielen Anwendungen, z.B. Pusher, kann die bewegte Masse aber von einer Bewegung zur anderen um bis zu 100 % variieren, so daß das Soft Stop-

System nicht anwendbar ist. Die elektronische Endlagendämpfung von Pneumatikzylindern ist also eine vielversprechende Anwendung für alternative Ansätze.

2.3 Idealbild einer elektronischen Endlagendämpfung

Wegen der vielfältigen Modellierungsprobleme bei pneumatischen Antrieben ist es sinnvoll, einen modellfreien Ansatz für den Entwurf und die Adaption zu verwenden, zumal die bewegte Masse, die Reibkräfte etc. nicht bekannt sind und sehr stark variieren können. Soll eine Endlagendämpfung quasi aus dem Regal verkauft werden, aber flexibel für eine möglichst beliebige Palette von Systemkonfigurationen einsetzbar sein, sind noch wesentlich härtere Anforderungen zu stellen. Es ist nämlich von einem beliebigen Mix aus Komponenten (insbes. Zylindertyp, -hub, -durchmesser, Ventile, Schlauchlängen), beliebigen bewegten Massen und einer beliebigen Einbaulage auszugehen. Trotzdem soll aus Kostengründen der Aufwand für die Inbetriebnahme beim Aufbau oder Änderungen einer Anlage, also das Einstellen von systemabhängigen Parametern, ohne Spezialkenntnisse beim Anwender auch für Laien möglich sein, d.h. auch ohne Schulung des Inbetriebnahme- oder Bedienpersonals.

Vom Verhalten wird natürlich eine möglichst schnelle Bewegung von einer Endlage zur anderen und eine flexible Anpassung an eine große Variation der bewegten Masse (bis zu 100 % von Bewegung zu Bewegung) gefordert. Aus Kostengründen sollen Schaltventile anstelle von Proportionalventilen eingesetzt werden. Um eine große Ventillebensdauer zu erreichen, wird außerdem eine möglichst kleine Anzahl von Schaltspielen gefordert. Schließlich sollte auch der Luftverbrauch minimal sein. Nach einer kurzen Beschreibung des Lösungsansatzes werden die gewonnenen Ergebnisse vorgestellt.



1. Vordere Zylinderkammer
2. Hintere Zylinderkammer
3. Leitung für den Versorgungsdruck
4. Drucksensor für den Versorgungsdruck
5. Drucksensoren für die Kammern
6. Positionserfassung
7. Benutzerereingaben
8. Ventile zum Entlüften der jeweiligen Kammern
9. Ventile zum Belüften der jeweiligen Kammern
10. Steuereinheit

Abb. 1: Schematischer Aufbau eines NEED-Systems

3 Lösungsansatz für eine Elektronische Endlagendämpfung

3.1 Vorbemerkungen

Den Aufbau eines Systems zur elektronischen Endlagendämpfung zeigt schematisch Abb. 1. Dabei werden sowohl die Position des Aktors (Pneumatikzylinder) als auch der Druck in beiden Zylinderkammern und der Versorgungsdruck erfaßt. Vier Schaltventile fungieren als Stellglieder, und zwar jeweils ein 2/2-Wege-Ventil für das Be- und Entlüften jeder Kammer. Alle Sensoren und Stellglieder sind an eine Steuereinheit angeschlossen, die im derzeitigen prototypischen Stadium, in dem die Analyse des Verhaltens unterstützt werden muß, ein PC ist. Ein Standard-16 Bit-Mikrocontroller ist aber für den praktischen Einsatz ausreichend.

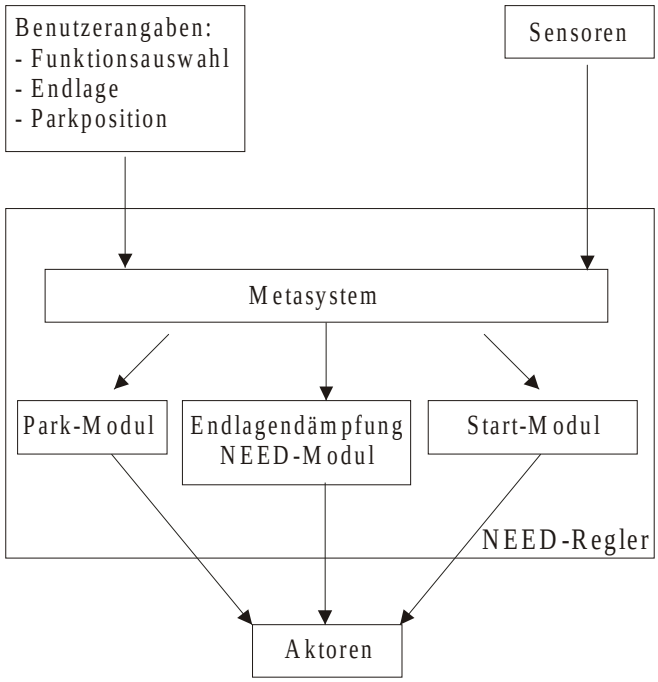


Abb. 2: Softwarearchitektur eines NEED-Systems

Das Ziel bei der Entwicklung des NEED-Systems (Neuro-fuzzy Electronic End-position Damping) war es, dem Ideal einer elektronischen Endlagendämpfung möglichst nahe zu kommen. Zusätzlich zur elektronischen Endlagendämpfung soll das NEED-System neben dem aktiven Halten der Endposition auch das Anfahren von Parkpositionen erlauben, z.B. für Arbeiten an der Anlage, für einen Werkzeugwechsel u.ä. Für jede der Funktionalitäten existiert ein eigenes Software-Modul (Abb. 2). Der eigentliche Kern des NEED-Systems ist das NEED-Modul. Es basiert auf einem (Neuro-)Fuzzy-Ansatz, um die Anforderungen nach Modellfreiheit, einer einfachen, schnellen Anwendbarkeit und einem sicheren Einsatz trotz Adaption zu erfüllen.

3.2 Lösungsansatz

Für das NEED-Modul sind zwei aufeinander aufbauende Varianten implementiert worden, nämlich zunächst eine rein wissensbasierte Lösung, um die grundsätzlichen Zusammenhänge zu studieren, und eine adaptive Variante, bei der im laufenden Betrieb eine permanente Adaption erfolgt und die eine Selbstkalibrierung unterstützt. Diese

Vorparameterierung durch Selbstkalibrierung wurde ergänzt, damit die Inbetriebnahme ohne Spezialwissen und möglichst einfach und schnell vonstatten geht.

Das NEED-Modul muß letztlich die oben diskutierte Abbildung von kontinuierlichen Prozeßgrößen auf diskrete Stellgrößen vornehmen. Es basiert darauf, daß eine kontinuierliche Zwischengröße eingeführt wird, die ein (Neuro-)Fuzzy-System generiert. Sie verdichtet eine Reihe von Prozeßgrößen auf eine einzige stetige und kontinuierliche Größe, aus der die erforderlichen Stellaktionen eindeutig abgeleitet werden können. Die dabei angewendete Vorgehensweise läßt sich wie folgt zusammenfassen. In einer Analysephase werden zunächst mehrere Phasen, die bei einer Bewegung von Endlage zu Endlage auftreten können, identifiziert und diskrete Übergangsbedingungen festgelegt. Ausgangspunkt dazu ist die Analyse, welche Schaltzustände es gibt bzw. welche Untermenge möglicher Schaltzustände für die Anwendung sinnvoll ist. Danach ist die Frage zu klären, welcher Effekt auf das Prozeßverhalten in einem gegebenen Prozeßzustand erreicht werden soll und durch welchen Schaltzustand dieser Effekt erzielt wird. Dazu wird aus den Sensorwerten (teilweise erst in der laufenden Bewegung) eine Reihe von Größen abgeleitet, die den Prozeßzustand widerspiegeln, wie Kräfte und Geschwindigkeit.

Die Menge der Schaltzustände ist zwar klein, aber der Zustandsraum kann abhängig von der Anzahl zu verarbeitender Prozeßgrößen beliebig groß werden. Beim Entwurf werden daher zunächst nur grobe Strategien festlegt, nach denen der Prozeß geregelt werden soll. Auf dieser Basis erfolgt dann eine hierarchische Strukturierung. So ist jeweils nur eine Untermenge der Eingangs- und Zwischenvariablen für die Übergangsbedingungen nötig. Der Schritt von kontinuierlichen Größen auf die diskreten Stellgrößen erfolgt durch den Wechsel zwischen den Schaltzuständen über die Übergangsbedingungen abhängig vom Vergleich der durch das Fuzzy-System generierten kontinuierlichen Zwischengröße mit einer festen Schwelle bzw. mit einer Prozeßgröße. Dieses Vorgehen erlaubt es, bis zum Vergleich mit kontinuierlichen Größen zu arbeiten. Das hält die Intuition des Experten aufrecht und erlaubt auch den Einsatz kontinuierlich arbeitender lernfähiger Systeme und Lernverfahren. Beim NEED-Modul mußte nur für eine Übergangsbedingung so vorgegangen werden. Alle anderen Übergangsbedingungen leiten sich direkt aus den Prozeßgrößen ab. Dadurch wird der Entwurf und die Handhabung des NEED-Moduls übersichtlicher und stark vereinfacht.

Grundsätzlich kann die Bestimmung der kontinuierlichen Zwischengröße(n) durch ein beliebiges Subsystem erfolgen. Zu bevorzugen sind an dieser Stelle wegen der Modellfreiheit, der Interpretierbarkeit, der Möglichkeit, Vorwissen einzubringen und die Sicherheit beim Lernen kontrollieren zu können, Fuzzy- bzw. Neuro-Fuzzy-Systeme. Sie fassen alle nicht offensichtlichen und zeitvarianten Effekte, die sich auch einer Modellierung entziehen, mehr oder weniger indirekt zu einer kontinuierlichen Zwischengröße zusammen. Die so generierte Zwischengröße sollte aus Anwendungssicht sinntragend sein, damit das Engineering zielgerichtet erfolgen kann und ein sicheres Verhalten erreicht wird. Wird sie durch ein Neuro-Fuzzy-System generiert, ermöglicht das auch eine Selbsteinstellung der Regelung und/oder die modellfreie Adaption an ein zeitvariantes Prozeßverhalten.

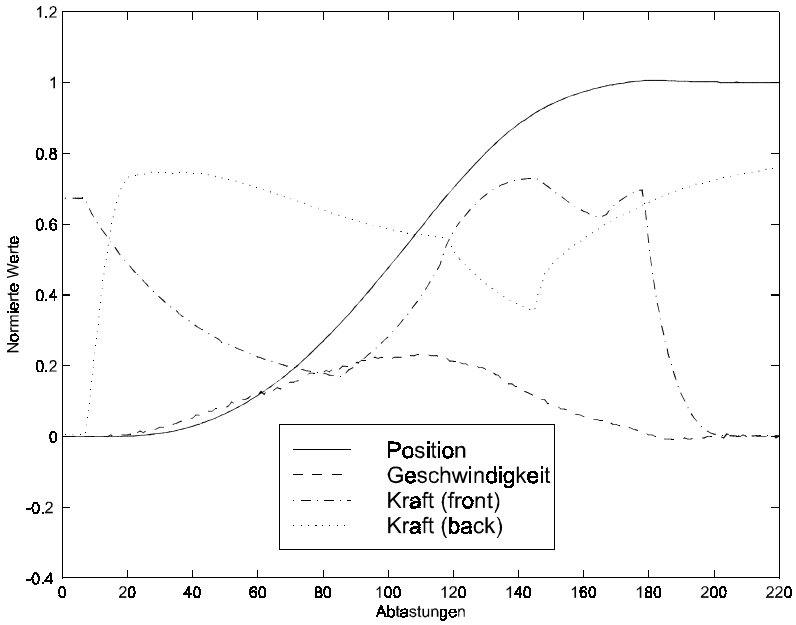
Beides wurde im Rahmen des NEED-Systems zur elektronischen Endlagendämpfung von Pneumatikzylindern untersucht. Bei der adaptiven Version wurde großer Wert auf eine schnelle Konvergenz und ein sicheres Systemverhalten auch und gerade während

der Adaption gelegt. Dabei erwies es sich als vorteilhaft, daß bei Anwendung der obigen Vorgehensweise keine Veränderungen im Eingangsraum vorgenommen werden müssen. Das erleichtert das Finden eines modellfreien Adaptionsgesetzes und beschleunigt die Adaption.

Beim Lernen in der adaptiven Variante mußte berücksichtigt werden, daß im Gegensatz zu adaptiven, kontinuierlich arbeitenden Regelungen keine ständig mitlaufende Adaption während einer Fahrt möglich ist. Vielmehr kann erst am Ende einer Bewegung beurteilt werden kann, wie groß die Auftreffenergie ist, also wie gut das Verhalten angepaßt ist. Dadurch stehen deutlich weniger Lerneingriffe zur Verfügung. Für eine schnelle Konvergenz bei der Adaption ist daher wichtig, daß möglichst wenig Parameter eingestellt werden müssen. Dadurch, daß die verhaltensbestimmenden Effekte in einer einzigen Zwischengröße zusammen gefaßt sind, die durch ein einzelnes, vergleichsweise einfaches Neuro-Fuzzy-System generiert wird, ist das hier erfüllt, und es wird trotzdem eine schnelle Konvergenz erreicht.

Die adaptive Variante des NEED-Systems wurde schließlich noch um eine Selbstkalibrierung ergänzt. Sie bewirkt eine sinnvolle Vorparametrierung und Skalierung interner Größen abhängig von der Systemkonfiguration. Dazu sind nur jeweils wenige Fahrten mit der minimalen und der maximalen bewegten Masse vorzunehmen. Die Selbsteinstellung geschieht dabei über eine Ermittlung spezifischer systeminterner Größen (Teach-In) und einer anschließenden Normierung. Die Generalisierung auf andere Massen wird im (Neuro-)Fuzzy-System bei dessen Initialisierung vor dem eigentlichen Betrieb vorgenommen.

4 Ergebnisse



Skalierung:	Position:	0 : eingefahren,	1 : ausgefahren
	Geschwindigkeit:	1 : -10 mm pro Abtastung	1 : 10 mm pro Abtastung
	Kraft:	0 : keine Kraft	1 : max. Zylinderkraft

Abb. 3: Typischer Verlauf der Prozeßgrößen während einer Fahrt

Das Verhalten des NEED-Systems wurde für verschiedene Massen an folgendem Versuchsaufbau untersucht:

- doppeltwirkender Zylinder nach DIN ISO 6432 CETOP mit 25 mm Durchmesser und 200 mm Hub
- vier 2/2-Wege-Schaltventile Kuhnke Typ 67 mit Nennweite 2
- Versorgungsdruck 4 bar, horizontaler Betrieb, Abtastzeit 3 ms

Einen typischen Verlauf der Prozeßgrößen zeigt Abb. 3 für einen Ausfahrvorgang mit 16 kg Nutzlast, der von der regelbasierten Version geregelt wird. Der Ausfahrvorgang wird zum Zeitpunkt 0 gestartet. Zunächst verstreichen ca. 10 Abtastungen (Ventiltotzeit, Einsetzen der Luftströmung), bis die Drücke anfangen, sich zu ändern. In der schiebenden Kammer wird dann wegen des kleinen Kammervolumens sehr schnell eine große Kraft aufgebaut, die zu einer Umkehr der Kraftrichtung führt und den Kolben nach ca. 20 Abstastungen zu beschleunigen beginnt. Die maximale Geschwindigkeit ist nach ca. 110 Abtastungen erreicht. Ab hier beginnt ein mehrstufiger Bremsvorgang, bei dem die kinetische Energie dosiert auf unterschiedliche Weise abgebaut wird, so daß die Endposition nach ca. 180 Abtastungen mit einer Geschwindigkeit nahe Null erreicht wird. Ab diesem Zeitpunkt wird der Zylinder fest an die Endposition gedrückt.

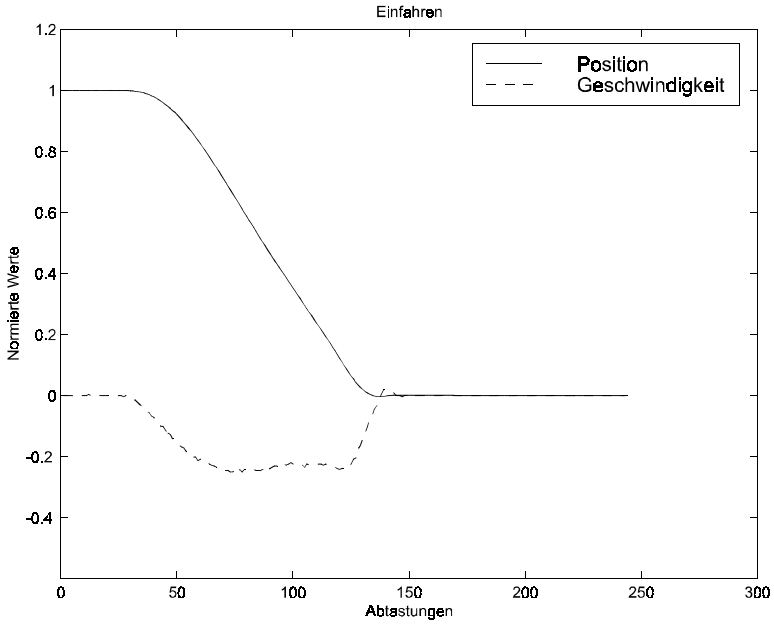


Abb. 4: Einfahren ohne Nutzlast

Prinzipiell sind die Verläufe für verschiedene Massen und für das Ein- und Ausfahren ähnlich. Letztere unterscheiden sich bei gleicher Masse in ihrem dynamischen Verhalten, weil in beiden Fällen unterschiedlich große Kolbenflächen wirksam sind. Es prägt sich aber immer eine glockenförmige Geschwindigkeitskurve aus. Lediglich bei kleinen bewegten Massen flacht sie wie in Abb. 4 oben ab, weil die Geschwindigkeit so groß werden kann, daß Strömungswiderstände die Menge der nachströmende Luft und damit die Geschwindigkeit nach oben begrenzen.

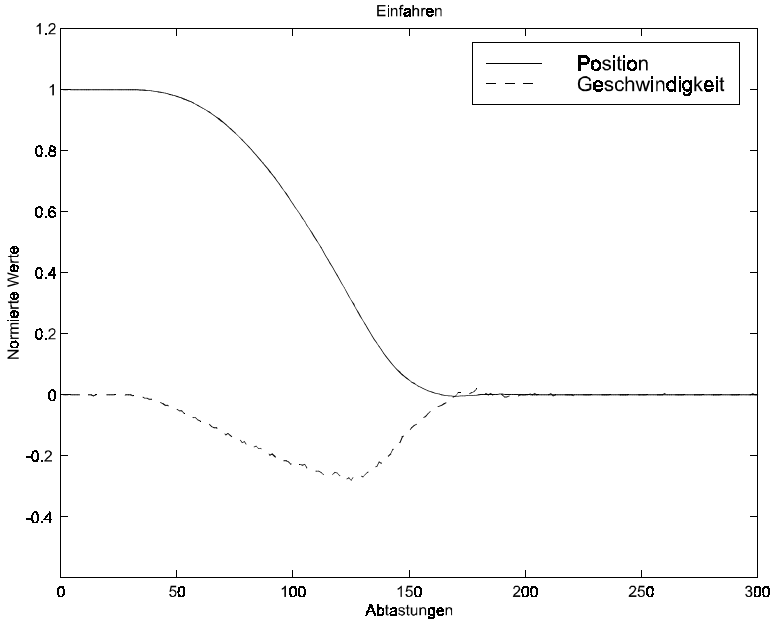


Abb. 5: Einfahren mit 8 kg Nutzlast

Bei einer größeren bewegten Masse wird keine so große Geschwindigkeit erreicht, daß die Luftströmung als Begrenzung wirkt, wie Abb. 5 für das Einfahren mit 8 kg Nutzlast zeigt. Auch hier wird die Endlage mit einer sehr kleinen Geschwindigkeit, also einer guten Endlagendämpfung, erreicht.

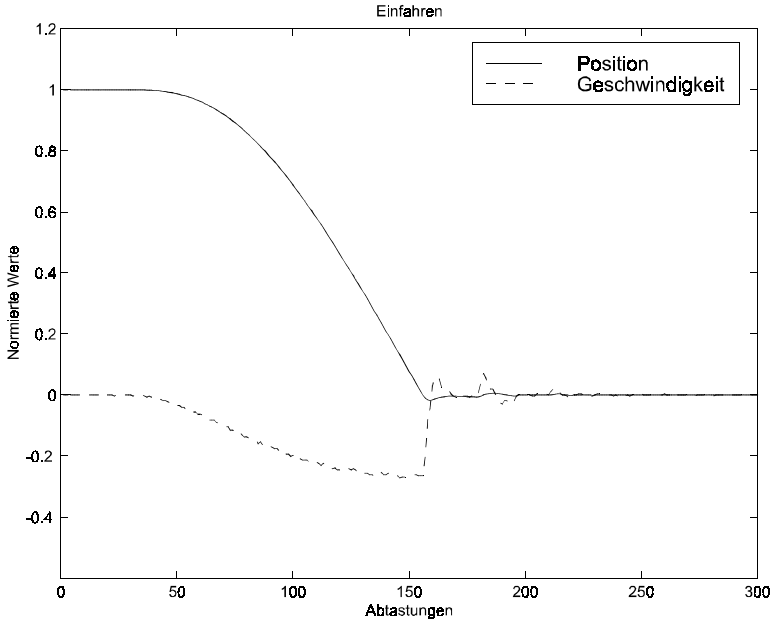


Abb. 6: Einfahren mit 8 kg Nutzlast ohne Endlagendämpfung

Im Vergleich zur Bewegung in Abb. 5 zeigt Abb. 6 eine Bewegung ohne Endlagendämpfung. Es wird also über den ganzen Verfahrweg maximal beschleunigt. Die Endposition wird hier mit einer so großen kinetischen Energie erreicht, daß die Wucht den Versuchsaufbau so stark verwindet, daß die sogar Positionserfassung Werte außerhalb des Bewegungsbereichs anzeigt. Trotz alledem ist die Verfahrzeit nicht wesentlich kürzer als bei einer Bewegung mit elektronischer Endlagendämpfung.

Tab. 1: Verfahrzeiten

Richtung	Nutzlast	Ohne NEED	Mit NEED
Ausfahren	0 kg	330 ms	360 ms
Einfahren	0 kg	390 ms	399 ms
Ausfahren	4 kg	360 ms	405 ms
Einfahren	4 kg	420 ms	450 ms
Ausfahren	8 kg	390 ms	450 ms
Einfahren	8 kg	459 ms	480 ms
Ausfahren	16 kg	438 ms	570 ms
Einfahren	16 kg	531 ms	570 ms

Tab. 1 stellt die Verfahrzeiten quantitativ für verschiedene Nutzlasten und beide Bewegungsrichtungen mit und ohne NEED-System gegenübergestellt. Der Beginn der Bewegungsphase ist in beiden Fällen solange gleich, bis das NEED-System vom Beschleunigen zum Bremsen übergeht. Mit zunehmender Nutzlast nimmt dadurch der Unterschied der Verfahrzeiten zu, weil zunehmend früher gebremst werden muß, um die bewegte Masse zielgenau abzubremsen. Dieser Effekt wirkt gleich zweifach, denn zum einen steht dadurch weniger Zeit und Stecke für die Beschleunigung zur Verfügung und zum anderen dauert der Bremsvorgang länger.

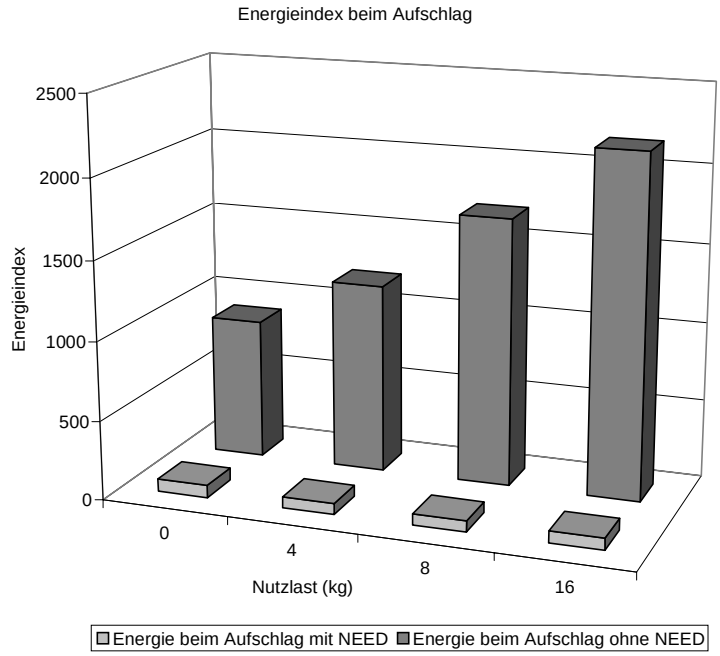


Abb. 7: Energieindex mit und ohne Endlagendämpfung

Tab. 1 verdeutlicht auch, daß durch die elektronische Endlagendämpfung mit dem NEED-System die Verfahrzeit beim Ausfahren um weniger als 10 % und beim Einfahren um weniger als 20 % erhöht wird. Wie stark im Vergleich dazu die Auftreffenergie am Ende einer Bewegung durch ein NEED-System abgebaut wird, zeigt Abb. 7. Sie stellt einen Energieindex, der proportional zur kinetischen Energie beim

Auftreffen auf die Endlage ist, für verschiedene Massen mit und ohne Endlagendämpfung gegenüber.

Dadurch, daß auf ein pulsierendes Schalten der Ventile für eine feinere Dosierung des Luftstroms verzichtet wurde, und durch eine angemessene Wahl der Phasen und eine geschickte Ausnutzung des dynamischen Verhaltens des pneumatischen Antriebssystems kommt die Endlagendämpfung zum einen mit einer sehr geringen Anzahl von Schaltspielen aus. Bei einer normalen Bewegung schaltet jedes Ventil höchstens zweimal. Nur in Ausnahmefällen, z.B. wenn die Adaption auf die bewegte Masse noch nicht ausreichend ist, werden zwei Ventile häufiger geschaltet. Dadurch wird eine hohe Lebensdauer der Ventile erreicht. Zum anderen ist der Luftverbrauch minimiert, weil keine Querströmung auftritt und während eines Bewegungsvorgangs keine Kammer erst belüftet und dann wieder entlüftet wird.

Die Variante des NEED-Systems, die mit Adaption arbeitet, verhält sich grundsätzlich genauso, wie die durch den Experten per Hand optimal eingestellte Version. Selbst bei der ersten Fahrt eines nur vorinitialisierten Systems wird immer eine ausreichende Reduktion der Auftreffenergie erreicht. Die Adaption benötigt pro bewegter Masse höchstens 3 bis 5 Fahrten, um vergleichbar gute Ergebnisse zu erreichen, obwohl die aus dem Prozeß abgeleiteten Informationen teilweise stark verrauscht sind. Das ist unabhängig davon, ob die bewegte Masse kurz vor der Endlage zum Stillstand kommt oder mit zu hoher Geschwindigkeit auf diese trifft. Ein sicheres Systemverhalten beim Adaptieren bzw. Lernen wird dadurch erreicht, daß eine feste Strategie verwendet wird, bei der nur Parameter, aber nicht das grundsätzliche Verhalten geändert werden. Wichtig ist dabei, daß dies Parameter einer stetigen, kontinuierlichen Beziehung sind.

Insgesamt hat das NEED-System Eigenschaften, die deutlich über die der klassischen, mechanischen Endlagendämpfung und über die der Festo-Lösung hinaus gehen. Das betrifft insbesondere die flexible, dynamische Anpassung an die bewegte Masse und eine schnellere Adaption mit 3 bis 5 statt 20 Fahrten. Sie erlaubt Änderungen der Nutzlast um 100 % von Bewegung zu Bewegung, ohne daß das System umparametriert werden müßte. Der Kostenersparnis durch den Einsatz von Schaltventilen anstelle von Proportionalventilen stehen die Kosten für die Druckerfassung gegenüber. Außerdem hat das NEED-System weniger Einschränkungen, was die Auswahl an Pneumatik-komponenten betrifft. Untersuchungen mit anderen Ventilen und Schlauchlängen bestätigen, daß die erarbeiteten Lösungen (Endlagendämpfung, Adaption und Selbst-kalibrierung) auch für andere Systemkonfigurationen einsetzbar sind. Aussagen über die Grenzen des Systems erfordern allerdings noch weitergehende Untersuchungen.

5 Zusammenfassung

Das Beispiel der elektronischen Endlagendämpfung von Pneumatikzylindern diente als erste Studie, um den Einsatz von Fuzzy- und insbesondere Neuro-Fuzzy-Systemen bei der Regelung von Prozessen zu untersuchen, die mit diskreten Stellgrößen angesteuert werden. Es konnte gezeigt werden, daß kontinuierlich arbeitende Fuzzy- und Neuro-Fuzzy-Systeme auch für solche Prozesse sinnvoll und zielgerichtet eingesetzt werden können. Dabei war auch eine Adaption möglich, bei der sichergestellt werden kann, daß das geregelte System während des Adaptionsvorgangs nicht instabil wird. Kernpunkt dabei ist, daß mit Hilfe eines (Neuro-)Fuzzy-Systems eine (möglichst sinntragende) Zwischengröße gebildet wird, aus der die diskreten Zustände der Stellgröße abgeleitet

werden. Beim Entwurf und bei der Adaption wirkte sich vorteilhaft aus, daß das Tuning der Parameter nicht durch Verändern von Parametern im Eingangsraum, sondern im Ausgangsraum des (Neuro-)Fuzzy-Systems erfolgte. Dadurch blieb die Vorstellung, die der Experte beim Entwurf hatte, erhalten. Außerdem ergab sich dadurch eine schnelle Konvergenz beim der Adaption.

Im beschriebenen Fall konnte ein Ablauf aus mehreren Phasen pro Bewegungsvorgang identifiziert werden, der zu dem gewünschten Verhalten führt. Dabei war nur für eine Phase der Einsatz eines (Neuro-)Fuzzy-Systems nötig, um alle relevanten Prozeßgrößen auf eine Zwischengröße zurückzuführen, die auch noch die Selbsteinstellung und Adaption an das zeitvariante Prozeßverhalten erlaubte. Durch die entsprechende Strukturierung des Systems konnte eine gute Verständlichkeit erreicht werden, so daß es einfach und zielgerichtet zu entwerfen war. Durch die rein verhaltensbasierte Vorgehensweise war der Entwurf einschließlich des Adaptionsgesetzes zielgerichtet und modellfrei möglich.

Wie weit sich die angewendete Methodik verallgemeinern läßt und wie sie sich mit alternativen Verfahren vergleicht, müssen weitere Arbeiten zeigen. Unter Laborbedingungen kann das Verhalten des NEED-Systems völlig zufriedenstellen. Vor dem Einsatz in realen Anwendungen sollten aber noch weitergehende Untersuchungen durchgeführt werden, insbesondere was die Robustheit und die Grenzen möglicher Systemkonfigurationen betrifft.

6 Literatur

[1] Bezdek, J.C.; Pal, S.K.: *Fuzzy Models for Pattern Recognition – Methods That Search for Structures in Data*. IEEE Press, Piscataway, USA; 1992

[2] Brockmann, W.; Huwendiek, O.: *Adaptive Fuzzy Control of a Batch Process*. Proc. 3rd European Congress on Intelligent Techniques and Soft Computing - EUFIT ´95, Verlag Mainz, Aachen; 1995, 883-888

[3] Brockmann, W.: *Engineering von Neuro-Fuzzy-Regelungen mit decide!*. 7. GMA-Workshop „Fuzzy Control“, Forschungsbericht Nr. 0397, Universität Dortmund, ISSN 0941-4169; 1997, 191-203

[4] Huwendiek, O.; Brockmann, W.: *Multi-staged Neuro Fuzzy Control of a Pneumatically Actuated Robot Arm*. In: P.P. Wang (Ed.): Proc. 4th Joint Conf. On Information Sciences - JCIS´98, Vol. 1, Durham, North Carolina; 1998, 68-72

[5] *Schneller Soft Stop*. Firmenschrift Info 183 der Firma Festo AG & Co., Esslingen; 2000

Generierung von Takagi–Sugeno–Fuzzy–Systemen aus relevanten Fuzzy–Regeln

Krause, P.

Lehrstuhl für Elektrische Steuerung und Regelung
Universität Dortmund, 44221 Dortmund
Tel.: 0231 755-2496 Fax: 0231 755-2752
E-Mail: Krause@esr.e-technik.uni-dortmund.de

Zusammenfassung

Bei der Verwendung von Fuzzy–Systemen in den Bereichen des Data–Minings und der Modellierung kommen zum einen Mamdani–Systeme und zum anderen Takagi–Sugeno–Kang(TSK)–Systeme zum Einsatz. Für die automatische Generierung der beiden Systemetypen haben sich verschieden Verfahren etabliert. Diese Verfahren haben Vor- und Nachteile bezogen auf das generierte System. Die Mamdani–Systeme sind einfach zu interpretieren, wohingegen TSK–Systeme eine höhere Approximationsgüte aufweisen.

Eine der Methoden, die ursprünglich für die datenbasierte Generierung von Mamdani–Systemen entwickelt worden ist, ist das Fuzzy–ROSA–Verfahren. In diesem Beitrag wird ein neues Verfahren vorgestellt, das ausgehend von, mit dem Fuzzy–ROSA–Verfahren generierten Mamdani–Systemen, TSK–Systeme erzeugt. Dabei wird die Approximationsgüte erhöht ohne dass die Interpretierbarkeit verloren geht. Bei anderen Verfahren, die TSK–Systeme generieren, kann dies meistens nicht gewährleistet werden.

Für das neue Verfahren werden verschieden Möglichkeiten für den Übergang von Mamdani–Systemen zu TSK–Systemen diskutiert. Die Anwendbarkeit des Verfahrens und die Ergebnisse werden anhand von Beispielen verdeutlicht.

1 Einführung

In den Bereichen des Data–Minings und der datenbasierten Modellierung werden verstärkt Modelle und Verfahren aus dem Bereich der Computational Intelligence eingesetzt. Für viele Aufgaben haben sich hierbei Fuzzy–Systeme als sehr gut geeignet herausgestellt. Daher haben sich in diesem Bereich verschiedene Methoden zur automatischen und datenbasierten Generierung von Fuzzy–Systemen etabliert. Diese Methoden werden zunächst grob in solche unterteilt, die TSK–Systeme [1] erzeugen und solche, die Mamdani–Systeme [2] generieren. Der Vorteil der generierten Mamdani–Systeme liegt in ihrer guten Interpretierbarkeit, wohingegen die TSK–Systeme in der Regel eine höhere Approximationsgüte erzielen. In den meisten Fällen sind beide Eigenschaften erwünscht, aber bei Anwendung der bekannten Verfahren muss ebenso oft auf einen der Vorteile zu Gunsten des anderen verzichtet werden.

Für die datenbasierte Generierung von Mamdani–Systemen ist das Fuzzy–ROSA–Verfahren [3, 4, 5, 6] entwickelt worden. Dieses Verfahren basiert auf der statistischen Relevanzanalyse von Hypothesen zur Generierung von Fuzzy–Regeln. Wird eine Hypothese ausreichend durch die vorliegenden Daten gestützt, repräsentiert diese Regel einen relevanten Teilaspekte des zu modellierenden Systems. Ein so generiertes Modell bietet eine große Einsicht in das betrachtete System, da die Fuzzy–Regeln hier leicht lesbar sind. Im Folgenden wird gezeigt, dass es trotz guter Modellierungsergebnisse ein bisher nicht ausgeschöpftes Potenzial zu weiteren gibt.

Dieses Potenzial kann durch den Einsatz von TSK–Systemen genutzt werden. Ausgehend von den zuvor generierten Mamdani–Fuzzy–Regeln, werden diese in TSK–Fuzzy–Regeln umgewandelt. Hierzu werden die Datenpunkte, die die Aussage einer relevanten Regel stützen, für eine Approximation der Ausgangsgröße herangezogen. Auf diese Weise wird die Lesbarkeit der Konklusionen der Regeln weiter gewährleistet. Das resultierenden Systeme bietet dann eine höhere Approximationsgüte und ist weiterhin gut interpretierbar.

Im Folgenden wird zunächst eine kurze Übersicht über Mamdani– und TSK–Systeme gegeben und danach über das Fuzzy–ROSA–Verfahren. Im Anschluss wird auf die Umwandlung der Regeln eingegangen und die Leistungsfähigkeit der hier vorgeschlagenen Vorgehensweise wird anhand von Beispielen verdeutlicht.

1.1 Fuzzy–Systeme

Die am häufigsten Typen von verwendeten Fuzzy–Systeme sind das Mamdani–System und das TSK–System. Beide Systemtypen verwenden die gleichen Verfahren für die Auswertung der Prämissen der Regeln, unterscheiden sich aber in der Art der Konklusion und der Berechnung des Ausgangsgößenwertes.

1.1.1 Mamdani–Systeme

Jede Regel r_i eines Mamdani–System hat die Form

$$\textbf{Wenn} \;\; P_i \;\; \textbf{Dann} \;\; C_i \quad , \tag{1}$$

wobei P_i eine Prämisse der Form

$$P_i = \bigwedge_{j=1}^{c} e_{i,j} \tag{2}$$

ist. Die Anzahl der c Elementaraussagen $e_{i,j}$ ist dabei kleiner oder gleich der Anzahl V der Eingangsvariablen. Jede Elementaraussage $e_{i,j}$ ist ein Ausdruck der Form „linguistische Variable=linguistischer Term“. Die Konklusion C_i ist ebenfalls eine Elementaraussage bezogen auf die Ausgangsgröße. Für die Berechnung der Ausgangsgröße y des Fuzzy–Systems zu einer bestimmten Eingangssituation, die durch den Eingangsvektor $\mathbf{x}$ gegeben ist, wird zunächst die ausgangsseitige Zugehörigkeitsfunktion $\mu(\mathbf{x}, y)$ nach der Vorschrift

$$\mu(\mathbf{x}, y) = \bigvee_{i=1}^{R} [p_i(\mathbf{x}) \wedge \mu_{Y,c_i}(y)] \tag{3}$$

gebildet. Dabei ist p_i der Erfülltheitsgrad der Prämisse P_i, $s_{Y,i}$ ist der Index des linguistischen Terms in der Konklusion der i-ten Regel und $\mu_{Y,c_i}(y)$ ist die Zugehörigkeitsfunktion des linguistischen Terms $s_{Y,i}$ der Ausgangsgröße. Aus der Zugehörigkeitsfunktion $\mu(\mathbf{x}, y)$ wird dann mit einer Defuzzifizierungmethode der Ausgangsgrößenwert berechnet. Für die Schwerpunkt–Defuzzifizierung (COG) ergibt sich der Ausgangsgrößenwert zu

$$y_D^{COG} = \frac{\int_{y_{min}}^{y_{max}} \mu(\mathbf{x}, y) \cdot y \, dy}{\int_{y_{min}}^{y_{max}} \mu(\mathbf{x}, y) \, du} \tag{4}$$

und für die Maximum–Defuzzifizierung (MOM)

$$y_D^{MOM} \quad \text{mit} \quad \mu(y_D^{MOM}) = \max(\mu(\mathbf{x}, y)) \quad . \tag{5}$$

Wenn mehrere Punkt y_D^{MOM} die Bedingung erfüllen, wird der mittlere Wert dieser Punkte als Ausgangsgrößenwert verwendet. Diese beiden Defuzzifizierungsmethoden sind die gebräuchlichsten.

1.1.2 TSK–Systeme

Jede Regel r_i eines TSK–Systems hat die Form

$$\textbf{Wenn} \;\; P_i \;\; \textbf{Dann} \;\; f_i(\mathbf{x}) \quad , \tag{6}$$

wobei für P_i das gleiche gilt wie in Abschnitt 1.1.1. Die Funktion $f_i(\mathbf{x})$ ist eine beliebig wählbare Funktion der Eingangsgrößen sein. Am häufigsten wird für $f_i(\mathbf{x})$ eine Linearkombination

$$f_i(\mathbf{x}) = a_{i,0} + \sum_{j=1}^{V} a_{i,j} \cdot x_j \quad . \tag{7}$$

der V Eingangsgrößen verwendet. Daraus wird der Ausgangsgrößenwert durch

$$y_D = \frac{\sum_{i=1}^{R} p_i \cdot f_i(\mathbf{x})}{\sum_{i=1}^{R} p_i} \tag{8}$$

definiert.

1.2 Das Fuzzy–ROSA–Verfahren

Die grundlegende Idee des Fuzzy–ROSA–Verfahrens besteht darin, einzelne Regeln daraufhin zu testen, ob sie einen signifikanten Teilaspekt des betrachteten Systems beschreiben. Die dafür verfügbaren statistischen Regeltest- und Bewertungsverfahren sind in [7, 8, 4] beschrieben. Mit diesem Ansatz wird das Problem, gute Regelbasen zu generieren, auf das Problem reduziert, einzelne relevante Regeln zu finden. Unter der Voraussetzung, dass relevante Regeln einen signifikanten Teilaspekt beschreiben und damit einzeln nachvollziehbar sind, können transparente und interpretierbare Regelbasen erzeugt werden.

Das Fuzzy–ROSA–Verfahren berücksichtigt grundsätzlich auch generalisierende (unvollständige) Regeln mit unterschiedlicher Prämissenlänge c (Kombinationstiefe). Wenn eine Regel aus weniger Teilprämissen (linguistischen Aussagen) als die gesamte Anzahl linguistischer Eingangsvariablen besteht, dann deckt diese Regel mehrere linguistische Eingangssituationen ab. Insbesondere in hoch dimensionalen Suchräumen mit vielen Eingangsgrößen sind Regelbasen aus generalisierenden Regeln meist kleiner als solche, die aus vollständigen Regeln bestehen.

Im Fuzzy–ROSA–Verfahren kann auch mit Zeittiefen gearbeitet werden, d. h. die linguistischen Aussagen können sich auch auf vorherige Zeitpunkte beziehen. Dies entspricht der Einführung einer neuen linguistischen Variablen (z. B. Temperatur vor zwei Zeitschritten).

Der Generierungsprozess ist in vier Hauptschritte unterteilt. Für jeden sind alternative Strategien verfügbar, so dass das Verfahren an die unterschiedlichen Anwendungsanforderungen (Modellierung, Klassifikation, etc.) und Problemgrößen (Anzahl ling. Variablen/Terme, Datensätzen etc.) angepasst werden kann.

Projektdefinition: Vor der eigentlichen Regelgenerierung müssen die Zugehörigkeitsfunktionen für die Eingangs-/Ausgangsvariablen des betrachteten Systems festgelegt werden. Dies kann wissensbasiert, datenbasiert oder heuristisch erfolgen [9]. Außerdem kann die maximale Kombinationstiefe c_{max} für die Prämisse und gegebenenfalls eine maximale Zeittiefe t_{max} festgelegt werden, um den Rechenaufwand zu beschränken

Regelgenerierung: Abhängig von der Suchraumgröße kann eine komplette Suche, eine evolutionäre Suche [10, 6] oder eine Kombination dieser beiden Suchstrategien gewählt werden. Der Regelsatz wird sukzessive aus allen relevanten, nicht redundanten Regeln aufgebaut.

Regelreduktion: Die Anzahl der Regeln kann anschließend durch Offline–Regelreduktionsverfahren verringert werden [11]. Dabei können verschiedene Anforderungen, wie z. B. komplette Überdeckung aller Eingangssituationen, gleichmäßige Ausnutzung der Daten, Verringerung des Modellierungsfehlers oder der Regelanzahl, berücksichtigt werden.

Regelsatzanalyse und -optimierung: Durch die abschließende Analyse des Regelsatzes kann der Modellierungsprozess und das Modellierungsergebnis bewertet und gegebenenfalls Feedback für die Problemformulierung erhalten werden. Zusätzlich kann das Eingangs-/Ausgangsverhalten des erhaltenen Fuzzy–Systems durch Anpassung der verbleibenden freien Parameter optimiert werden. Ein Verfahren, die optimierende Konfliktreduktion [12], kann als Kombination aus Regelreduktion und E/A-Optimierung angewendet werden.

2 Transformation von Regeln

Die Regeln, die in der Form von Gl. (1) mit dem Fuzzy–ROSA–Verfahren generiert werden, erlauben Systemzusammenhänge einfach zu erkennen und zu verstehen. Dies

ist gerade für industrielle Anwendungen von hoher Bedeutung, um die Akzeptanz von Fuzzy–Systemen zu fördern. Zur Verdeutlichung der hier vorgestellten Methoden wird im folgenden ein Beispielsystem mit den zwei Eingangsgrößen „*Druck*“ und „*Temperatur*“ und der Ausgangsgröße „*Zufuhr*“ verwendet. Für dieses System sei folgende relevante Regel r_1

$$\textbf{Wenn}\ \ Temperatur = mittel \wedge Druck = normal\ \ \textbf{Dann}\ \ Zufuhr = mittel\,, \quad (9)$$

gefunden worden. Der Systemzusammenhang, der sich daraus ergibt, ist klar erkennbar. Trotzdem kann bei einer Validierung dieser Regel ein unerwartet hoher Fehler auftreten. Wird nur die Regel r_1 für die Auswertung herangezogen, so ergibt sich für die Situation ein konstanter Ausgangsgrößenwert. In Abbildung 2(links) zeigt die Ebene, die dem berechneten Ausgangsgrößenwert entspricht. Die grauen Balken repräsentieren den Fehler der einzelnen Datenpunkte zu dem empfohlenen Wert. Der mittlere Fehler, der sich so ergibt ist relativ hoch, obwohl die Regel einen tatsächlich vorhandenen Zusammenhang in dem zu modellierenden System beschreibt. Wird nun anstatt des linguistischen Wertes in der Konklusion eine Funktion verwendet, die eine Hyperebene repräsentiert (Gl. (7)), dann kann der Fehler durch Anpassung (Regression) der Parameterwerte der Funktion drastisch verkleinert werden (Abbildung 2(rechts)).

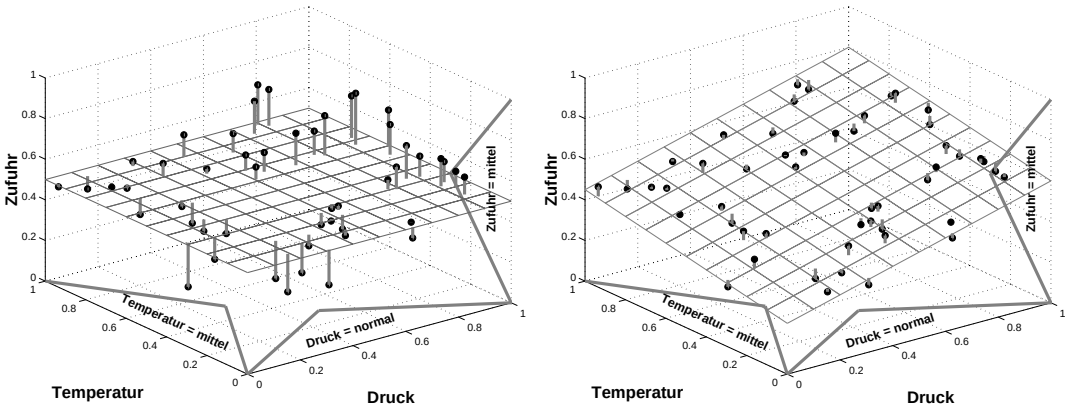


Abbildung 1: Modellierungsfehler auf Validierungsdaten für ein Mamdani– und ein TSK–System (links bzw. rechts)

Auf diese Weise bleibt die linguistische Aussage erhalten: Die Hyperebene ist so gelegt ist, dass für jeden Datenpunkt in der Prämisse ein Wert berechnet wird, der innerhalb der Konklusion liegt. Gleichzeitig wird die Approximationsgüte erhöht, ohne die Lesbarkeit der Regeln zu beeinträchtigen.

2.1 Singleton Systeme

Für die Wahl der Form der ausgangsseitigen Zugehörigkeitsfunktion für Mamdani–Systeme sind verschieden Alternativen bekannt. Die gebräuchlichsten sind trapezförmige oder Gauß'sche Zugehörigkeitsfunktionen. Als Sonderfall werden häufig auch Singletons als ausgangsseitigen Zugehörigkeitsfunktionen verwendet. Aus programmiertechnischen Überlegungen, sind für die hier betrachteten Verfahren trapezförmige Zugehörigkeitsfunktionen verwendet worden. D. h. eine Zugehörigkeitsfunktion

wird durch vier Stützstellen $y_1 \leq y_2 \leq y_3 \leq y_4$ definiert und die folgende Funktion

$$\mu_Y(y) = \begin{cases} 0 & : \quad y < y_1 \\ \frac{y-y_1}{y_2-y_1} & : \quad y_1 \leq y < y_2 \\ 1 & : \quad y_2 \leq y < y_3 \\ \frac{y_4-y}{y_4-y_3} & : \quad y_3 \leq y < y_4 \\ 0 & : \quad y \geq y_4 \end{cases} \tag{10}$$

beschrieben. Diese Darstellung beinhaltet als Sonderfälle Rechtecke, Dreiecke und Singletons. Für die datenbasierte Generierung von Regeln mit dem Fuzzy–ROSA–Verfahren für kontinuierliche Ausgangsgrößen werden zumindest Funktionen benötigt, für die gilt $y_1 < y_4$. Für den Einsatz des Fuzzy–Systems können die Zugehörigkeitsfunktionen in Singletons überführt werden, da sich dadurch die Inferenz und die Defuzzifizierung erheblich vereinfacht und beschleunigt. Die Überführung geschieht durch Bildung des Flächenschwerpunktes f_S der Zugehörigkeitsfunktion[1]. Durch eine solche Umwandlung verändert sich das Verhalten des Systems nur sehr geringfügig und die Aussage der Regeln bleibt erhalten.

Ein Regel, die mit der Konklusion ein Singleton empfiehlt, kann einfach in eine TSK–Regel umgewandelt werden, in dem die Funktion $f_i(\mathbf{x})$ als Konstante mit dem zuvor bestimmt Flächenschwerpunkt angesetzt wird. Die Regel r_i hat dann die Form

$$\textbf{Wenn} \;\; P_i \;\; \textbf{Dann} \;\; f_S(\mu_{Y,c_i}) \quad . \tag{11}$$

Auch hier bleibt die Lesbarkeit der Regel bestehen, da jedem f_S eindeutig ein linguistischer Term der Ausgangsgröße zugeordnet ist.

Eine weitere Anwendung findet diese Umwandlung bei dem mittelwertbasierten Bewertungsindex[13]. Dieser beruht auf dem Vergleich des allgemeinen Mittelwerts der Ausgangsgröße und dem Mittelwert der Ausgangsgröße in einer bestimmten Situation. Wird eine relevante Regel gefunden, so wird der Mittelwert der Ausgangsgröße in dieser Situation empfohlen. Die bisherige Implementierung beruhte darauf, dass derjenige ausgangsseitige Term empfohlen wird, der dem Mittelwert am nächsten ist. Unter Verwendung eines TSK–Systems kann der Mittelwert direkt empfohlen werden[2].

2.2 Verwendung von Hyperebenen

Wie Abbildung 2(rechts) zeigt, können die Daten noch einen Trend aufweisen, den eine Mamdani–Regel nicht berücksichtigten kann. Um die Mamdani–Regel zu transformieren, wird einen TSK–Regel r_i der Form

$$\textbf{Wenn} \;\; P_i \;\; \textbf{Dann} \;\; \overbrace{f_S(\mu_{Y,c_i}) + b_i}^{a_{i,0}} + \sum_{j=1}^{n} a_{i,j} \cdot x_j \tag{12}$$

[1]Wegen der häufigen Verwendung von symmetrischen Zugehörigkeitsfunktionen wird in dem Programm WINROSA das Singleton durch $\frac{y_3-y_2}{2}$ bestimmt

[2]Die bisherige Implementierung der Software lies keine TSK–Systeme zu, so dass dies nicht früher berücksichtigt werden konnte

aufgestellt. Von den gegebenen D Daten fallen D_P Datenpunkte in die Prämisse P_i und davon haben $D_{P\wedge C}$ Datenpunkte eine Ausgangsgröße, die die Konklusion C_i aktivieren, d. h. $\mu_{Y,c_i}(y) > 0$. Mit Hilfe eines Least–Squares–Algorithmus[14] basierend auf den $D_{P\wedge C}$ Datenpunkte werden die Koeffizienten $a_{i,j}$, $0 \leq j \leq V$ bestimmt. Bei einer gleichmäßigen Verteilung der $D_{P\wedge C}$ Datenpunkte in der Prämisse P_i ist für eine Eingangssituation $\mathbf{x}$, $p_i(\mathbf{x}) > 0$ die Beziehung $\mu_{Y,c_i}(y) \geq 0$ erfüllt. Dies bedeutet, dass trotz einer Anpassung an die Daten, die zuvor bestimmte linguistische Aussage weiter bestand hat. Wird die Regel nun wie Gl. (12) notiert, mit $b_0 = a_{i,0} - f_S(\mu_{Y,c_i})$, kann über $f_S(\mu_{Y,c_i})$ die linguistische Aussage Regel weiter verstanden werden.

Die ausschließliche Berücksichtigung nur der Datenpunkte mit $p_i(\mathbf{x}) > 0 \wedge \mu_{Y,c_i}(y) > 0$ statt aller Datenpunkte $p_i(\mathbf{x}) > 0$ hat folgenden Grund: Würden auch Datenpunkte mit $p_i(\mathbf{x}) > 0 \wedge \mu_{Y,c_i}(y) = 0$ für die Bestimmung der Hyperebene verwendet werden, könnte eine ungewollte Verschiebung der Hyperebene aus dem Bereich der Konklusion auftreten. Um dies zu vermeiden, werden diese Punkte nicht berücksichtigt. Dies gilt insbesondere, da aufbauend auf dem Relevanzkonzept des Fuzzy–ROSA–Verfahrens nicht zwangsläufig $D_P \approx D_{P\wedge C}$ gelten muss. Es kann sogar durchaus der Fall $D_{P\wedge C} < \frac{D_P}{2}$ auftreten, in dem die Approximation dann nicht mehr mit der ursprünglich intendierten Aussage der Regel in Zusammenhang stehen würde. Dieses Verhalten konnte anhand verschiedener Tests beobachtet werden, und es hat sich gezeigt, dass die Ergebnisse mit den Regelsätzen, die $D_{P\wedge C}$ Punkte zur Approximation verwenden, besser sind, als die, die D_P Punkte verwenden.

Eine Ausnahme bildet die Transformation eines Regelsatzes, der mit dem mittelwertbasierten Bewertungsindex erstellt worden ist. Hierbei soll der richtige Mittelwert in einer bestimmten Eingangssituation vorhergesagt werden. Daher werden hier alle D_P Punkte für die Approximation verwendet, um so den mittleren Fehler zu reduzieren.

Die Voraussetzung für die Anwendung eines Least–Squares–Algorithmus, sind $V+1$ unabhängige Datenpunkte. Sind nicht genügend unabhängige Datenpunkte für eine relevante Regel vorhanden, so werden alle $a_{i,j}$, $1 \leq j \leq V$ gleich Null gesetzt und $a_0 = f_S(\mu_{Y,c_i})$. Dies gilt auch für alle weiteren Ansätze, wie zum Beispiel in 2.3.

Das Fuzzy–ROSA–Verfahren generiert nicht nur Regeln in deren Prämissen alle Eingangsvariablen vorkommen, sondern auch generalisierende Regeln[15]. Es wird nun die generalisierende Regel r_i

$$\textbf{Wenn} \;\; Druck = normal \;\; \textbf{Dann} \;\; Zufuhr = mittel \tag{13}$$

betrachtet. Jetzt könnte der Ansatz verfolgt werden, auch nur die Eingangsgrößen, die in der Prämisse vorkommen, für die Berechnung der Hyperebene zu verwenden. D. h. alle $a_{i,j}$ aus Gl(12) werden fest auf 0 gesetzt, die sich auf eine Eingangsgröße beziehen, die nicht in der Prämisse vorkommt. Dieser Ansatz vereinfacht die Aufstellung und Lösung der Regression deutlich, insbesondere für große V. Abbildung 2 zeigt die Auswirkungen dieser Vorgehensweise: Unabhängig von der Temperatur ist die Zufuhr mittel, wenn der Druck normal ist. Wird nur der Druck für die Bestimmung einer Hyperebene herangezogen (Abbildung 2(links)), so verbessert sich die Approximation, aber, der auch in der Temperatur vorhandene Trend wird nicht berücksichtigt (Abbildung 2(rechts)).

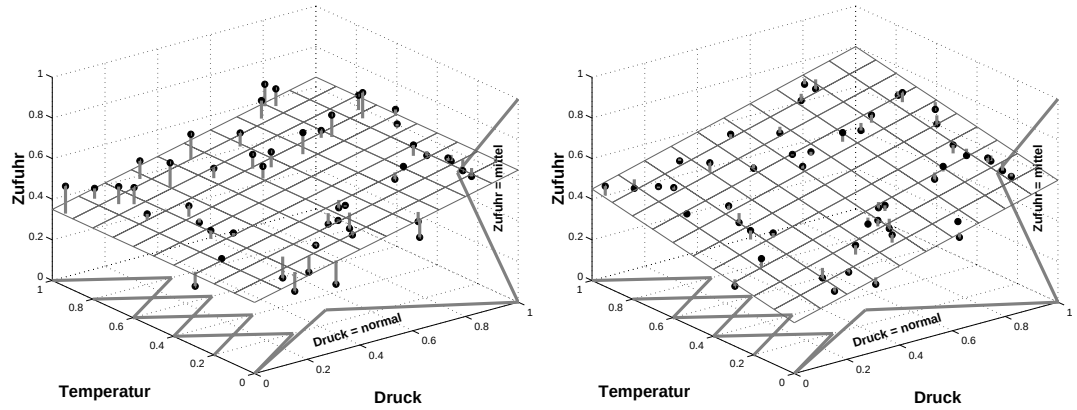


Abbildung 2: Fehler der Validierung bei unterschiedlichen Ansätzen für die Hyperebenen. Nur generalisierende Eingangsgrößen (links), alle Eingangsgrößen (rechts)

Offensichtlich werden mit beiden Ansätzen Verbesserungen erzielt, jedoch ist die Verwendung stets aller Eingangsgrößen für die Regression deutlich überlegen. Dies wiegt den erhöhten Aufwand bei der Berechnung der Hyperebenen bei weitem auf, so dass immer alle Eingangsgrößen für die Regression verwendet werden sollten.

2.3 Verwendung von allgemeinen Funktionsansätzen

Die zuvor gezeigte Verwendung von Hyperebenen für die Konklusion in einer TSK–Regel ist zwar der gebräuchlichste, aber sie ist auch zugleich nur ein Spezialfall von einem allgemeinen Funktionsansatz. Allgemeiner kann eine TSK–Regel r_i wie folgt angesetzt werden:

$$\textbf{Wenn} \;\; P_i \;\; \textbf{Dann} \;\; \overbrace{f_S(\mu_{Y,c_i}) + b_i}^{a_{i,0}} + \sum_{j=1}^{m} a_{i,j} \cdot g_j(\mathbf{x}) \quad . \tag{14}$$

Die Funktionen $g_j(\mathbf{x})$ sind dabei eine beliebige Anzahl m von linearen oder nichtlinearen Funktionen. Die Hyperebene bildet den Spezialfall, dass $m = V$ und $g_j = x_j$. Für eine bestimmte Prämisse P_i lassen sich die a_j mit Hilfe eines Least–Squares–Algorithmus genauso wie in Abschnitt 2.2 exakt bestimmen. Eine Erweiterung des Hyperebenenansatzes ist die Verwendung von multilinearen Funktionen der Form

$$f_i = a_{i,0} + a_{i,1} \cdot x_1 + \ldots + a_{i,V} \cdot x_V + a_{i,n+1} \cdot x_1 \cdot x_2 + \ldots + a_{i,2^V} \cdot \prod_{l=1}^{n} x_l \quad . \tag{15}$$

Durch die Verwendung von multilinearen Funktionen können mehr Freiheitsgrade genutzt werden. Dadurch können weitere nichtlineare Tendenzen entdeckt und approximiert werden (Abbildung 3).

Der Vorteil eines multilinearen Ansatz liegt in der noch besseren Approximationsgüte. Dies kann allerdings auch zu einem Overfitting führen (3). Des weiteren wächst die Anzahl der Koeffizienten mit 2^V, so dass bei höher dimensionalen Problemen zum einen die Berechnung der Koeffizienten sehr zeitintensiv wird, und

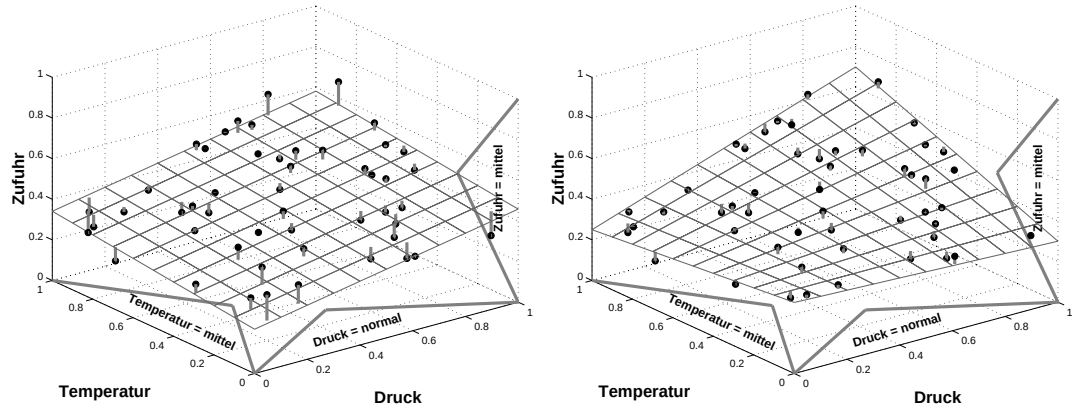


Abbildung 3: Fehler der Validierung bei linearem (links) und multilinearem Ansatz (rechts)

zum anderen sind entsprechend viele Datenpunkte nötig, um eine Approximation durchführen zu können.

Allgemein bietet die Verwendung allgemeiner nichtlinearer Ansätze den Vorteil Vorwissen in den Modellierungsprozess einzubringen. Werden jedoch Ansätze mit sehr vielen freien Parametern gewählt, kann dies leicht zu einem Overfitting führen, oder es kann keine Approximationen mehr durchgeführt werden, da nicht genügend unabhängige Datenpunkte für eine Regel vorhanden sind.

3 Beispiele

3.1 Mackey–Glass Zeitfolge

Die chaotische Mackey–Glass–Zeitfolge [16] wird in der Literatur häufig zum Testen von Lern- und Modellierungsverfahren genutzt. Die diskrete Version der Mackey–Glass–Zeitfolge lässt sich beschreiben durch

$$x(t+1) = (1-a)x(t) + \frac{bx(t-\tau)}{1+x^{10}(t-\tau)} \qquad . \tag{16}$$

Als Parameter werden $a = 0.1$, $b = 0.2$ und $\tau = 17$ gewählt. Für $t < 0$ ist $x(t) = 0$, und für $t = 0$ ist $x(0) = 1.2$. Die Aufgabe besteht darin, auf Basis der Werte x(t-18), x(t-12), x(t-6) und x(t) den Wert x(t+6) vorherzusagen. Die Ausgangsgröße kann dabei Werte zwischen 0.4 und 1.3 annehmen. Für das hier vorgestellte Verfahren wurden 1000 vorliegenden Datenpunkte zufällig in je zwei Hälften zu 500 Datenpunkten geteilt. Tabelle 1 zeigt den mittleren absoluten Fehler (MAE) für die verschiedenen Durchläufe. Der Basisregelsatz ist mit dem Relevanzindex (J_{RI} mit $\alpha = 0.1$) generiert worden. Es ist zu sehen, dass mit der Verwendung von einem TSK–System die Modellierungsgüte deutlich zunimmt. Werden die ersten drei Systeme mit Hilfe der Optimierenden Konfliktreduktion (OCR) optimiert, können in allen drei Fällen die Ergebnisse weiter verbessert werden. Allerdings ist auch zu sehen, dass die Transformation mit Hilfe eines multilinearen Ansatz ohne Optimierung schon besser ist als

der optimierte Basisregelsatz. Für dieses Beispiel führt die Verwendung eines multilinearen Ansatz nicht zu einem Overfitting, sondern die in den Daten vorhandenen Tendenzen sind multilinear und werden hier entsprechend gut modelliert.

	Lerndaten	Validierungsdaten	Gesamtdaten
Basisregelsatz	0.06389	0.06639	0.06551
TSK linear	0.03143	0.03404	0.03313
TSK multilinear	0.02071	0.02258	0.02193
Basisregelsatz OCR	0.03098	0.03255	0.03200
TSK linear OCR	0.01208	0.01483	0.01387
TSK multilinear OCR	0.00850	0.01035	0.00970

Tabelle 1: MAE für Gesamt-, Lern- und Validierungsdaten

Tabelle 2 zeigt eine Einordnung der Ergebnisse mit anderen Verfahren (Vergleich [5]). Die neben dem MAE verwendeten Bewertungen sind der mittlere quadratische Fehler (MSE), die Standardabweichung (RMSE) und der mittlere quadratische Fehler geteilt durch die Standardabweichung der Originaldaten (NDEI).

Verfahren	Regelanzahl	MAE	MSE	RMSE	NDEI
[17]	4096	–	–	–	0.0042
	20736	–	–	–	0.0011
[18]	81	–	0.0057	–	–
	256	–	0.0023	–	–
[19]	129	–	–	0.0332	–
[20]	23	–	–	0.0114	–
Basisregelsatz	92	0.0655	0.0080	0.0895	0.0343
TSK linear	92	0.0331	0.0021	0.0454	0.0088
TSK multilinear	92	0.0219	0.0016	0.0405	0.0070
Basisregelsatz OCR	31	0.0320	0.0017	0.0415	0.0074
TSK OCR	21	0.0139	$3.73 \cdot 10^{-4}$	0.0193	0.0016
TSK multilinear OCR	24	0.0097	$2.09 \cdot 10^{-4}$	0.0145	$8.96 \cdot 10^{-4}$

Tabelle 2: Ergebnisse für verschiedene Verfahren

In [17] werden die Zugehörigkeitsfunktionen und Regeln schrittweise im Sinne des Modellierungsfehlers verbessert. In [18] wird eine ähnliche schrittweise Vorgehensweise vorgeschlagen. In [19] wird eine Neuro–Fuzzy–Lernmethode vorgestellt, die auf möglichst einfache und schnelle Weise ein gängiges Fuzzy–System vom Mamdani–Typ erzeugt, bei dem die Zugehörigkeitsfunktionen eine feste Reihenfolge haben und

sich nicht beliebig überlappen dürfen. In [20] wird eine Neuro–Fuzzy–Lernmethode vorgestellt, die kleine Regelbasen anstrebt. Die Zugehörigkeitsfunktionen dürfen eine beliebige Reihenfolge annehmen und können sich beliebig stark überlappen.

Der Vergleich der Methoden zeigt, dass die Ergebnisse für das TSK–System mit multilinearem Ansatz und OCR in den meisten Fällen die besten sind. Das einzige System, dass eine bessere Modellierungsgüte mit einer vergleichbaren Anzahl von Regeln aufweist ist [20]. Der Nachteil dieses Systems ist allerdings die oben beschriebene Bestimmung der Zugehörigkeitsfunktionen, die die Interpretierbarkeit der generierten Regeln sehr erschwert.

3.2 Boston Housing Problem

Das Boston Housing Problem[3][21] ist ebenfalls ein Benchmarkproblem. Die Daten enthalten Informationen über Grundstückspreise in dem Gebiet von Boston, Mass., die von dem U.S Census Service gesammelt worden sind. Zu jedem Grundstückspreis gehören 13 Merkmale, die die Lage des Grundstücks beschreiben. Die Werte der Grundstücke schwanken zwischen 10000$ und 50000$. Insgesamt sind 506 Datenpunkte verfügbar, die zu gleichen Teilen in Lern- und Validierungsdaten geteilt worden sind. Der Basisregelsatz ist mit dem Relevanzindex (J_{RI} mit $\alpha = 0.1$) generiert worden. Für die Bewertung ist wieder der mittlere absolute Fehler (MAE) verwendet worden, wobei der Fehler hier in 1000$ angegeben ist.

	Regeln	Lerndaten	Validierungsdaten	Gesamtdaten
Basisregelsatz	160	3.240	3.613	3.427
TSK linear	160	2.679	2.973	2.826
Basisregelsatz OCR	32	2.210	3.033	2.622
TSK linear OCR	27	1.798	2.678	2.238

Tabelle 3: MAE für Gesamt-, Lern- und Validierungsdaten

In Tabelle 3 ist zu erkennen, dass die Transformation auch für ein höher dimensionales Problem anwendbar ist. In beiden Fällen, mit und ohne OCR, ist eine deutliche Verbesserung zu erkennen. Ein vollständiger multilinearer Ansatz ist hier allerdings nicht mehr praktikabel, da 2^{13} Parameter für jede relevante Regel zu bestimmen wären.

3.3 Kurzfristige Lastprognose

Ziel dieses Modellierungsproblems ist es, eine möglichst genaue Prognose der in einem Versorgungsgebiet nachgefragten elektrischen Leistung zu erstellen. In [22] ist das Problem ausführlich beschrieben. In diesem Fall wird die Vorhersage der

[3]Data for Evaluation Learning in Valid Experiments:
http://www .cs.toronto.edu/ delve/data/boston/desc.html

Laständerung auf der Basis von der Tageszeit, dem Tagestyp und der Dämmerung getroffen. Die vorliegenden Daten sind in einem 15-Minuten Raster über ein aufgenommen worden. Das 1. Halbjahr ist zum Lernen und das 2. Halbjahr zum Validieren verwendet worden. Für die Erstellung des Basisregelsatzes ist der mittelwertbasierte Bewertungsindex (J_{MVB} mit $\alpha = 0.01$) verwendet worden. Die Laständerung schwankt zwischen -350 MW und 350 MW. Wenn immer eine Laständerung von 0 MW vorgeschlagen würde, würde ein mittlerer absoluter Fehler von 53.11 MW gemacht werden. Dies bedeutet, das dieser Wert durch die Modellierung verbessert werden muss. Die Ergebnisse sind in Tabelle 4 zusammengefasst.

	Lerndaten	Validierungsdaten	Gesamtdaten
Basisregelsatz 101 Sets	40.18	40.47	40.33
Basisregelsatz exakt	40.14	40.46	40.30
TSK linear	38.74	39.89	39.31
TSK multilinear	38.40	40.12	39.26
Neuronales Netz	44.00	45.10	44.55

Tabelle 4: MAE für Gesamt-, Lern- und Validierungsdaten

Der Basisregelsatz mit den 101 ausgangsseitigen linguistischen Termen ist, wie in Abschnitt 2.1 beschrieben, erstellt worden. Es ist zu sehen, dass durch die Verwendung des exakten Mittelwerts keine große Verbesserung erzielt wird. Die Transformation mit einem linearen Ansatz verbessert die Prognose jedoch um 1 MW im Mittel. Für die gegeben Problemstellung ist dies eine signifikante Verbesserung, mit der Einsparungen vorgenommen werden können. Der multilinear Ansatz führt hier zu einem Overfitting. Das Ergebnis auf den Lerndaten kann weiter verbessert werden, aber das Gesamtergebnis wird schlechter. Im Rahmen eines Methodenvergleichs wurde von einem anderen Projekt des SFB CI der Universität Dortmund versucht, die Problemstellung mit Hilfe eines neuronalen Netz zu lösen. Zum jetzigen Stand zeigt sich, dass die Ergebnisse weniger befriedigend sind, als die, die mit einem Fuzzy–Systemen erzielt wurden.

4 Zusammenfassung und Ausblick

In diesem Beitrag ist gezeigt worden, wie aufbauend auf ein datenbasiert generiertes Mamdani–Systems, dieses unter Zuhilfenahme eines TSK–Systems verbessert werden kann. Die Verbesserung bezieht sich hierbei auf eine höhere Approximationsgüte.

Mamdani–Systeme haben als herausragende Eigenschaft ihre gute Interpretierbarkeit. Dies geht allerdings zumeist auf Lasten der Approximationsgüte. TSK–Systeme sind in der Regel sehr schwer interpretierbar können aber die Daten besser approximieren. Durch die Transformation von Mamdani–Regeln in TSK–Regeln auf die hier beschrieben Weise kann die Approximationgüte verbessert werden, ohne dass die Interpretierbarkeit verloren geht. Der Ansatz für die Transformation ist ein Least–Squares–Algorithmus, der Hyperebenen, multilineare Funktionen oder allgemeine

Funktionen an die Daten, die eine Regel stützen, anpasst. Hierzu sind verschiedene Ansätze diskutiert worden.

Derzeit wird daran gearbeitet dieses Verfahren auf weitere Aufgabenstellungen anzuwenden. Dabei ist besonders Einsatz von nichtlinearen Funktionsansätzen von Interesse.

Danksagungen

Diese Arbeit wurde von der Deutschen Forschungsgemeinschaft im Rahmen des Sonderforschungsbereiches *„Computational Intelligence“ (531)* gefördert.

Ein Teil der **Referenzen** kann auf folgender Webseite geladen werden:
http://esr.e-technik.uni-dortmund.de/winrosa/winrosa.htm.

Literatur

[1] Takagi, T. ; Sugeno, M.: Fuzzy Identification of Systems and its Application to Modeling and Control. In: *IEEE Transactions on Systems, Man, and Cybernetics* 15 (1985), Nr. 1, S. 116–132

[2] Mamdani, E. H. ; Gaines, B. R.: *Fuzzy Reasoning and its Applications.* London : Academic Press, 1981

[3] Krabs, M.: *Das ROSA–Verfahren zur Modellierung dynamischer Systeme durch Regeln mit statistischer Relevanzbewertung.* Fortschritt–Berichte VDI, Reihe 8, Nr. 404. Düsseldorf : VDI, 1994

[4] Kiendl, H.: *Fuzzy–Control methodenorientiert.* München : Oldenbourg Verlag, 1997

[5] Krone, A.: *Datenbasierte Generierung von relevanten Fuzzy–Regeln zur Modellierung von Prozesszusammenhängen und Bedienstrategien.* Düsseldorf : VDI, 1999 (Fortschritt–Berichte VDI, Reihe 10, Nr. 615)

[6] Slawinski, T. ; Krone, A. ; Hammel, U. ; Wiesmann, D. ; Krause, P.: A Hybrid Evolutionary Search Concept for Data-based Generation of Relevant Fuzzy Rules in High Dimensional Spaces. In: *Proceedings of IEEE International Conference on Fuzzy Systems, FUZZ–IEEE ’99* Bd. 3. Seoul, Korea, 1999, S. 1432–1437

[7] Krone, A. ; Taeger, H.: Relevance Test for Fuzzy Rules. In: *Reihe Computational Intelligence*, CI–40/98, Sonderforschungsbereich 531, Universität Dortmund, 1998

[8] Jessen, H. ; Slawinski, T.: Test- and Rating Strategies for Data–based Rule Generation. In: *Reihe Computational Intelligence*, CI–39/98, Sonderforschungsbereich 531, Universität Dortmund, 1998

[9] Krone, A. ; Slawinski, T.: Data–based extraction of unidimensional fuzzy sets for fuzzy rule generation. In: *Proceedings of the 1998 IEEE World Congress on Computational Intelligence, WCCI ’98.* Bd. 2. Anchorage, Alaska, USA, 1998, S. 1032–1037

[10] Krone, A. ; Kiendl, H.: Evolutionary Concept for Generating Relevant Fuzzy Rules from Data. In: *International Journal of Knowledge–based Intelligent Engineering Systems* 1 (1997), Nr. 4, S. 207–213

[11] Krone, A.: Advanced Rule Reduction Concepts for Optimising Efficiency of Knowledge Extraction. In: *Proceedings of the Fourth European Congress on Intelligent Techniques and Soft Computing, EUFIT '96* Bd. 2. Aachen, 1996, S. 919–923

[12] Krone, A. ; Krause, P. ; Slawinski, T.: A New Rule Reduction Method for Finding Interpretable and Small Rule Bases in High Dimensional Search Spaces. In: *Proceedings of the Ninth IEEE International Conference on Fuzzy Systems, FUZZ–IEEE '00* Bd. 2. San Antonio, USA, 2000, S. 696–699

[13] Jessen, H.: *Test- und Bewertungsverfahren zur regelbasierten Modellierung und Anwendung in der Lastprognose.* Düsseldorf : VDI, 2000 (Fortschritt–Berichte VDI, Reihe 8, Nr. 836)

[14] Press, W. H. ; Teukolsky, S. A. ; Vetterling, W. T. ; Flannery, B. P.: *Numerical Recipes in C.* Cambridge, England : Cambridge University Press, 1999

[15] Krone, A. ; Slawinski, T. ; Krause, P.: Search Space Structuring as a Key to cope with different Problem Sizes in the Field of Fuzzy Modeling. In: *Proceedings of World Automation Congress, WAC '00.* Maui, Hawaii, USA, 2000

[16] Mackey, M. ; Glass, L.: Oscillation and chaos in physiological control systems. In: *Science* 197 (1977), S. 287–289

[17] Nakoula, Y. ; Galichet, S. ; Foulloy, L.: Simultaneous Learning of Rules and Linguistic Terms. In: *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, FUZZ–IEEE '96* Bd. 3. New Orleans, USA, 1996, S. 1743–1749

[18] Chen, C.-L. ; Hsu, S.-H. ; Hsieh, C.-T. ; Lin, W.-K.: Generating Crisp–type Fuzzy Models from Operating Data. In: *Proceedings of the Seventh IEEE International Conference on Fuzzy Systems, FUZZ–IEEE '98* Bd. 1. Anchorage, Alaska, USA, 1998, S. 686–691

[19] Nauck, D. ; Kruse, R.: A Neuro–fuzzy Approach to Obtain Interpretable Fuzzy Systems for Function Approximation. In: *Proceedings of the Seventh IEEE International Conference on Fuzzy Systems, FUZZ–IEEE '98* Bd. 2. Anchorage, Alaska, USA, 1998, S. 1106–1111

[20] Cho, K.B. ; Wang, B.H.: Radial Basis Function based Adaptive Fuzzy Systems and their Applications to System Identification and Prediction. In: *Fuzzy Sets and Systems* 83 (1996), S. 325–339

[21] Harrison, D. ; Rubinfeld, D. L.: Hedonic Prices and the Demand for Clean Air. In: *Economics & Managment* 5 (1978), S. 81–102

[22] Jessen, H.: New Frontiers in Computational Intelligence and its Applications. IOS Press, 2000, Kapitel Test and Rating Strategies for Automatic Fuzzy Rule Generation and Application to Load Prediction, S. 11–21

Datenaufbereitung mittels Wavelet-Methoden

Sam Ellis

Mess-, Steuer-, und Regelungstechnik (Prof. Dr.-Ing. H. Schwarz)
Gerhard-Mercator-Universität Duisburg, D-47048 Duisburg
Tel: 0203 / 379 1585 Fax: 0203 / 379 3027
E-Mail: sellis@uni-duisburg.de

Kurzfassung

Dieser Beitrag gibt einen Überblick über die Wavelet-Transformation und ihre Anwendung in der Reduktion eines Messsignals durch Auswahl geeigneter lokaler Eigenschaften. Die Reduktion eines Messsignals auf seinen wesentlichen Informationsgehalt ist ein wichtiger Schritt in der Systemanalyse und -identifikation. Ausgehend von einem kurzen Umriss der Wavelet-Theorie wird in diesem Beitrag über die Eigenschaften unterschiedlicher Wavelet-Funktionen und die Gestaltung einer geeigneten Transformation diskutiert. Auf Grund der überwiegenden Zahl digitalisierter Messsignale in der technischen Praxis, gilt der Großteil dieser Diskussion der diskreten Wavelet-Transformation. Eine Wavelet basierte wird Datensatzreduktion anhand des Raddrehzahlsignals eines ABS-Sensors demonstriert. Hierzu findet eine Darstellung der Methoden für die Auswahl geeigneter Signalelemente aus den Wavelet-Transformationskoeffizienten statt, mit der eine Datenkomprimierung einhergeht.

1 Einleitung

Ein Messsignal stellt den Verlauf einer physikalischen Größe dar und gibt Auskunft über den Zustand des gemessenen Prozesses. Die darin enthaltenen Informationen bieten einerseits wichtige Einsichten in das Verhalten des betrachteten Systems, anderseits ermöglichten sie den gezielten Eingriff in das Systemverhalten mit einem Regler. In der technischen Praxis ist die wesentliche Information im Nutzsignal oftmals mit Störsignalen überlagert, so dass eine Verarbeitung des Signals notwendig ist, um Nutz- und Störanteil voneinander unterscheiden zu können.

Eine der wesentlichen Methoden der Signalverarbeitung ist die Spektralanalyse, die ein Messsignal in seine harmonischen Komponenten aufteilt [1]. Für viele technische Systeme existiert bereits ein Fundus an Erfahrung und Wissen über ihre Frequenzeigenschaften und ihr Verhalten. Eine spektralbasierte Signalverarbeitung ermöglicht die direkte Anwendung dieses Wissens und erleichtert die Kommunikation und Interpretation der Ergebnisse mit und unter Ingenieuren. Ein geeignetes und weit verbreitetes Werkzeug für die Erstellung der Frequenzspektren periodischer und stationärer Signale ist die Fourier-Transformation [2], die Dank des Fast-Fourier-Transformationsalgorithmus und moderner Rechenleistung sehr schnell und effizient berechnet werden kann.

Wegen der grundlegenden physikalischen Gesetze, sowie stochastischer Anregungen und der bereits erwähnten Störungen besitzen technische Messsignale meistens einen nichtstationären Charakter. Eine geeignete Positionierung der Sensorik kann den Einfluß störender Prozesse auf ein Minimum reduzieren. Jedoch ist eine optimale Positionierung

der Messgliede am System aus konstruktionstechnischen und/oder finanziellen Gründen nicht immer möglich. Die Problematik der Signalanalyse liegt dann in der Auffindung der wesentlichen Informationen aus einem stark verrauschten, stochastischen Signal. Die klassischen Fourier-basierten Filtermethoden setzen die Stationarität und Linearität des Signals voraus, und liefern bei stochastischen Signalen nur eine (teilweise unzufriedenstellende) Approximation der spektralen Eigenschaften.

Das Frequenzspektrum eines nichtstationären Prozesses kann durch ihre Aufteilung in *lokal* stationären Prozessen und die Untersuchung dieser Teilprozessen erfolgen [3]. Ein relativ neues Analyseverfahren, das seine Wurzel zum Teil in den seismischen Untersuchungen der Geologie hat, ist die Wavelet-Transformation (Abschnitt 2), die lokale Basisfunktionen anwendet, um eine orthogonale Darstellung eines Signal und eine bessere Zeit-Frequenz-Auflösung zu erzielen.

Die diskrete Wavelet-Transformation (Abschnitt 3) teilt die Signaleigenschaften in Skalen immer enger werdender Bandbreite. Hier findet auch eine Diskussion über geeignete Transformationsbasen zur Untersuchung der Frequenzeigenschaften eines Signals statt. Aus den Transformationskoeffizienten ist es dann möglich, bestimmte Ereignisse im Prozess zeitlich festzulegen und ihre Frequenzeigenschaften, bzw. ihre Auswirkung auf das Gesamtsystem festzustellen. Die Vorgehensweise wird anhand des Raddrehzahlsignals eines Fahrzeuges demonstriert (Abschnitt 4). Dieser Abschnitt führt auch eine Reduktion der Koeffizienten und eine Minimaldarstellung des Signals durch. Die bleibenden Koeffizienten heben die wesentliche Informationen im Signal hervor und filtern unwichtige Frequenzen aus dem Spektrum heraus.

2 Grundlagen der Wavelet-Theorie

Ein Messsignal im Zeitbereich $f(t)$ liefert eine beschränkte Beschreibung der Prozesse und Zustände, die in einem gemessenen Signal enthalten sind. Die Übertragung des Signals in einen anderen Darstellungsbereich erlaubt dem Betrachter, weitere Signaleigenschaften zu beobachten. Die Fourier-Transformation $\widehat{f}(\omega)$ ist die bekannteste und meistverwendete Methode für die Beschreibung der Signalinformation in einem anderen Bereich. In diesem Fall wird das Signal $f(t)$ mit einer Basis harmonischer Funktionen verschiedener Frequenz verglichen. Mathematisch geschieht dies über das innere Produkt zwischen dem Signal $f(t)$ und den Basisfunktionen $g(t) = \mathrm{e}^{j\omega t}$:

$$\widehat{f}(\omega) = \langle f(t), g(t) \rangle = \int_{-\infty}^{\infty} f(t)\mathrm{e}^{-j\omega t}\mathrm{d}t \quad . \tag{1}$$

Die Ähnlichkeit zwischen dem Signal und den einzelnen harmonischen Funktionen wird jeweils mit einem Koeffizient bewertet. Gemäß der Theorie von Parseval [2] entspricht der Verlauf dieser Koeffizienten über das Frequenzspektrum der Verteilung der Leistung im Signal. Ein hoher Koeffizient bedeutet eine hohe Leistung der harmonischen Schwingung mit entsprechender Frequenz.

Ein Nachteil der harmonischen Basisfunktionen der Fourier-Transformation ist ihre unendliche Länge. Die Fourier-Transformation (1) bildet das komplette Signal ab und wandelt alle *lokale* Eigenschaften im Zeitbereich in *globale* Eigenschaften im Frequenzbe-

reich um. Selbst bei Verwendung einer Kurz-Zeit-Fourier-Transformation führt die Abbildung kurzer Signalelemente mit unendlich langen Basisfunktionen zu mangelhaften Ergebnissen, besonders bei der Transformation von Transienten und Unstetigkeiten [4].

2.1 Bildung einer Wavelet-Transformationsbasis

Ein Wavelet $\psi(t) \in L^2(\mathbb{R})$ ist eine rasch abklingende, oszillierende Funktion mit endlicher Leistung. Das heißt, es ist nur innerhalb eines kurzen Zeitbereiches ungleich Null. Die mathematischen Grundlagen dieser Eigenschaften lassen sich mit Hilfe seiner Fourier-Transformation beschreiben:

$$\widehat{\psi}(0) = \int_{-\infty}^{\infty} \psi(t)\mathrm{d}t = 0 \quad , \tag{2}$$

was bedeutet, dass die Funktion oszilliert und sein Mittelwert gegen Null verschwindet. Ist Gl. (2) erfüllt und $\widehat{\psi}(\omega)$ ist stetig differenzierbar, so ist auch die Wavelet *Zulässigkeitsbedingung*

$$C_\psi = \int_{0}^{+\infty} \frac{|\widehat{\psi}(\omega)|^2}{\omega}\mathrm{d}\omega < +\infty \tag{3}$$

erfüllt. Die Fourier-Transformierte $\widehat{\psi}(\omega)$ ist dann stetig differenzierbar, wenn $\psi(t)$ hinreichend schnell abklingt: $\int_{-\infty}^{+\infty}(1+|t|)|\psi(t)|\mathrm{d}t < +\infty$. Weitere notwendige Bedingungen für die Existenz eines Wavelets sind in der Literatur enthalten [3, 5].

Eine Transformationsbasis wird gebildet in dem die Wavelet-Funktion ψ über dem Parameter s skaliert (gedehnt bzw. gestaucht) und über dem Parameter u versetzt wird:

$$\psi_{u,s}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-u}{s}\right) \quad . \tag{4}$$

Gleichung (4) stellt eine Schar von oszillierenden Funktionen mit unterschiedlichen Frequenzeigenschaften und zeitlichen Positionen. An dieser Stelle sei ausdrücklich betont, dass nur eine Wavelet-Funktion die komplette Wavelet-Transformationsbasis erzeugt. Damit haben alle Funktion der Transformationsbasis die gleiche Form, wenn auch eine unterschiedliche Skalierung.

In Analogie zur Fourier-Transformation (1), bildet die Wavelet-Transformation (5) das Signal zur verschieden Skalen ab und erfasst damit die oszillierenden Eigenschaften des Signals. Die Transformation erfolgt über das innere Produkt zwischen dem Signal $f(t)$ und einer Basis von Wavelet-Funktionen $g(t) = \psi_{u,s}(t)$:

$$W(u,s) = \langle f(t), g(t)\rangle = \int_{-\infty}^{\infty} f(t)\overline{\psi}_{u,s}\mathrm{d}t \quad , \quad u > 0, s \in \mathbb{R}. \tag{5}$$

Wavelet-Funktionen mit einer hoher Skalierung haben eine lange Periode und bilden die entsprechende niederfrequente Anteile im Signal ab. Umgekehrt erfassen die kurzen, klein skalierten Wavelet-Funktionen die hochfrequenten Signalanteile. Aufgrund der schmaleren Breite besitzen die klein skalierten Wavelet-Funktionen eine feinere zeitliche Auflösung als die groben (hohen) Skalen. In Abhängigkeit der Abtastfrequenz des Signals ist es möglich, die Wavelet-Skalen in Frequenzen umzurechnen (siehe Abschnitt 3). Die zeitliche Abbildung des Signals erfolgt durch die Verschiebung der endlich lange Wavelet-Basisfunktionen über der Länge des Signals. Das Signal wird schrittweise mit den Basisfunktionen verglichen und Koeffizienten, die (ebenfalls mit Hilfe der Parseval'schen Theorie) die Leistung des Signals zu den verschiedenen Skalen beschreiben, werden jeweils zu den einzelnen Positionen entlang des Zeitvektors berechnet, so dass ein zeitlicher Verlauf des Signalschwingungsverhaltens entsteht. Bild 1 zeigt die Eigenschaften eines Signals aufgeteilt in dyadischen Skalen, d.h. die Auflösung verdoppelt sich mit jedem Skalierungsschritt.

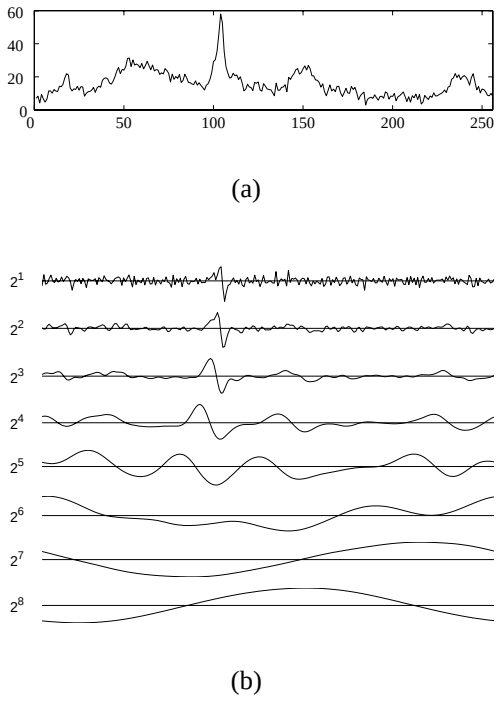


Bild 1: (a) Nichtstationäres Signal. (b) Dyadische Wavelet-Transformation für die Skalen bis 2^8.

3 Diskrete Wavelet-Transformation

Die kontinuierliche Wavelet-Transformation in (5) wird diskretisiert in dem der Skalierungsparameter s durch ganzzahlige Potenzen eines festen Auflösungswertes $s_0 > 1$ ersetzt wird, d. h. $s = s_0^j,\ j \in \mathbb{Z}$. Die Exponenten j bestimmen dann die Skalierung des Wavelets. Eine Diskretisierung des Verschiebungsparameters u hängt von der Breite des Wavelets und damit vom Parameter j ab. Enge Wavelets (niedrige Skalierung) werden mit kleineren Schritten verschoben, um das ganze Signal zu transformieren; breitere Wavelets (hohe Skalierung) werden mit größeren Schritten verschoben. Die Breite eines Wavelets ist zur Skalierung s_0^j proportional, d. h. $k\, s_0^j,\ k \in \mathbb{Z}$. Für einen festen Mindestschrittwert $u_0 > 0$ ist die Diskretisierung des Verschiebungsparameters durch $k u_0 s_0^j$ beschrieben. Die Verwendung dieser diskretisierten Parameter in Gl. (5) ergibt die diskrete Version der Dilatations- und Translationsgleichung:

$$\psi_{j,k}(t) = s_0^{-j/2} \psi \left(s_0^{-j} t - k\, u_0 \right) \quad . \tag{6}$$

Bei der Diskretisierung der Transformation darf keine Information verloren gehen; d. h. die Folge, die aus dem inneren Produkt $\langle f, \psi \rangle$ entsteht, muss das Signal $f(t)$ komplett charakterisieren und das Signal muss aus dieser Folge exakt rekonstruierbar sein. Eine Funktionsschar $\{e_n\}_{n \in \mathbb{Z}}$ kann ein Signal $f \in \boldsymbol{H}$ stabil und komplett darstellen wenn

sie eine *Riesz-Basis* des Signalraumes bildet [3, 6]. Dies ist der Fall wenn $\{e_n\}$ eine orthonormale Basis ist. Die Rekonstruktion eines Signals f erfolgt dann in einfacher Weise durch eine lineare Erweiterung mit den orthonormalen Basisfunktionen:

$$f = \sum_{n\in\mathbb{Z}} \langle f, e_n\rangle e_n \quad . \tag{7}$$

Das Einsetzen von $s_0 = 2$ und $u_0 = 1$ bildet aus den Wavelet-Funktionen in Gl. (6) eine orthonormale Basis des Signalraumes [7, 8]

$$\psi_{j,k}(t) = 2^{-j/2}\psi\left(2^{-j}t - k\right), \qquad j, k \in \mathbb{Z}\,. \tag{8}$$

Mit Hilfe dieser Basisfunktionen beschreibt die diskrete Wavelet-Transformation die Eigenschaften eines Signals zu diskreten Punkten in der Zeit-Skala-Ebene.

Ein Messsignal $f(t)$ aus der technischen Praxis existiert nicht als analytische Funktion, sondern als Vektor von Abtastschritten $f[n] = \{x_n; n = 0, 1, \ldots, N\}$. Ein mathematisch und numerisch effizientes Verfahren zur Berechnung der diskreten Wavelet-Transformation einer diskreten Signalfolge ist Mallats *Mehrfachauflösungs-Analyse* (engl. Multiresolution Analysis, MRA) [7].

3.1 Frequenzermittlung aus diskreter Wavelet-Skalen

Durch die Faltung des Signals $f(t)$ mit dem skalierten Wavelet $\psi_{j,k}(t)$ werden Änderungen der gleichen Periode wie das Wavelet selbst mit hohen Koeffizienten bewertet. Da der Skalierungsparameter j die Periode des Wavelets $\psi(t)$ bestimmt, in dem er sie dehnt oder staucht, existiert eine Verbindung zwischen den Wavelet-Skalen und der Frequenzen im Signal.

Im praktischen Fall ist der Wertebereich des Skalierungsparameters durch die Länge N des Signalvektors beschränkt. Eine Wavelet-Transformation kann nur Signaleigenschaften mit einer Periode zwischen 1 und N Datenpunkten erfassen. Ferner, in Analogie zur Shannon'schen Abtasttheorie, können nur die Signaleigenschaften, die eine maximale Frequenz von $\frac{F_\mathrm{T}}{2}$ besitzen, als solche erkannt werden, wobei F_T die Abtastfrequenz ist. Auf Grund der dyadischen Werte der Waveletskalierung, die notwendig für die Existenz einer orthonormalen Transformationsbasis (Gl. (8)) ist, setzen viele Software-Analysepakete eine Signallänge, die ebenfalls Teil einer dyadischen Zahlenfolge ist, d. h. $\log_2(N) \in \mathbb{Z}$, voraus, um schnelle Algorithmen anwenden zu können. Aus diesem Grund geht dieser Bericht, wie auch viele andere Beiträge über Wavelets, von einer dyadischen Signallänge aus. Diese zum Teil technischen, zum Teil physikalischen Begrenzungen der Skalierungsparameter führen für den Skalierungsindex zum folgenden gültigen Wertebereich:

$$0 \le j \le \log_2\left(\frac{N}{2}\right) \quad .$$

Eine dyadische Skala eines diskreten Signals entspricht einem Frequenzband einer ebenfalls dyadisch aufgeteilten Signal-Abtastfrequenz. Die erste Skalierung, 2^0, hat eine Bandbreite von $\frac{F_\mathrm{T}}{2}$, die nächste Skalierung 2^1 enthält alle Signaleigenschaften der Frequenzen zwischen $\frac{F_\mathrm{T}}{2}$ und $\frac{F_\mathrm{T}}{4}$, die Skalierung 2^2, schließt alle Frequenzen zwischen $\frac{F_\mathrm{T}}{4}$

und $\frac{F_\mathrm{T}}{8}$ ein, bis schließlich die gröbste Skalierung, 2^J, das Frequenzband zwischen $\frac{F_\mathrm{T}}{2^J}$ und $\frac{F_\mathrm{T}}{2^{J+1}}$ darstellt.

Unter Angabe der Signalabtastfrequenz F_T ist das entsprechende Frequenzband für eine beliebige diskrete Skala 2^j angegeben als

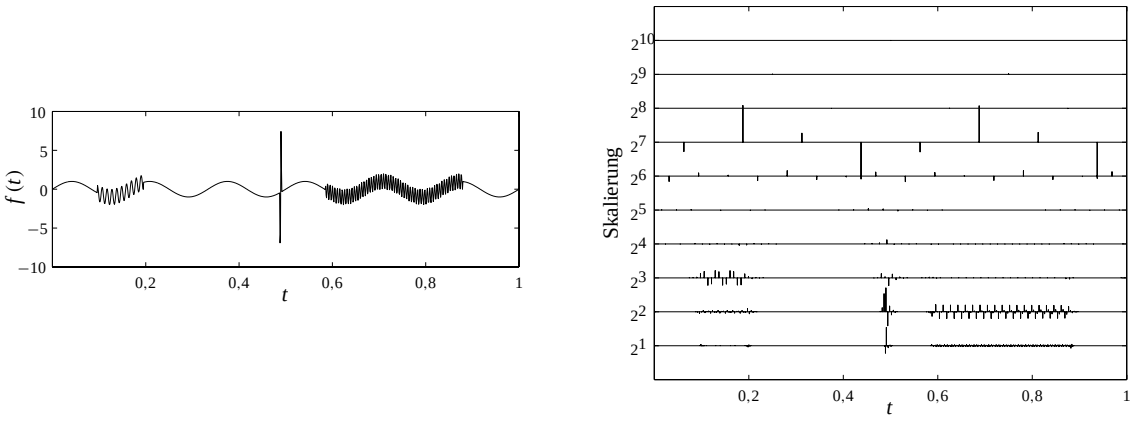
$$\Omega_\mathrm{o} = \frac{F_\mathrm{T}}{2^j}, \qquad \Omega_\mathrm{u} = \frac{F_\mathrm{T}}{2^{j+1}} \quad , \tag{9}$$

wobei Ω_o der obere und Ω_u der untere Frequenzwert ist. Diese Beziehung zwischen Wavelet-Skala und Frequenz ist in Tabelle 1 erläutert.

In Bild 2(a) ist eine 6 Hz Dauerschwingung mit zwei überlagerten kurzlebigen Schwingungen von 96 Hz bzw. 192 Hz dargestellt. Die Zerlegung der Signaleigenschaften in Wavelet-Skalen mittels DWT ist in Bild 2(b) gezeigt. Mit Hilfe der Tabelle 1 sind insbesondere drei Resonanzfrequenzen zu erkennen: Eine Dauerschwingung im Bereich zwischen 4 und 8 Hz, eine kurzzeitige Schwingung im Bereich zwischen 64 und 128 Hz und eine etwas längere Schwingung im Bereich zwischen 128 und 256 Hz. Eine schmale Gruppe von senkrecht übereinanderstehender Koeffizienten (zum Zeitpunkt $t = 0{,}5$) deuten auf eine hochfrequente, impulsartige Änderung hin.

Skala	**Frequenz [Hz]**
2^{10}	0,5 – 1
2^9	1 – 2
2^8	2 – 4
2^7	4 – 8
2^6	8 – 16
2^5	16 – 32
2^4	32 – 64
2^3	64 – 128
2^2	128 – 256
2^1	256 – 512
(2^0	512 – 1024)

Tabelle 1: Korrespondenz zw. Skala und Frequenz für $F_\mathrm{T} = N = 1024$.



(a) (b)

Bild 2: (a) Beispielsignal der Länge $N = 1024$ und (b) die zugehörigen Wavelet-Koeffizienten einer DWT, verteilt über alle Skalen

Das Beispiel in Bild 2(b) zeigt die wesentlichen Vor- und Nachteile einer diskreten Darstellung der Signalfrequenzeigenschaften mittels Wavelet-Transformation gegenüber z. B.

der Fourier-Transformation. Der deutliche Vorteil ist die schon erwähnte Erhaltung zeitlicher Information, so dass nicht nur die Frequenzen, sondern auch die Frequenzdynamik untersucht werden kann. Eine Beschränkung der diskreten Wavelet-Transformation ist die exponential steigende Breite der Frequenzbände bei zunehmender Auflösung. Kleine Änderungen der Frequenz im höheren Frequenzbereich können somit nicht erfasst bzw. nicht voneinander unterschieden werden. Ein weiterer Nachteil der diskreten Wavelet-Transformation ist die zum Teil schlechte Frequenzlokalisierung der Koeffizienten, selbst bei eindeutigen harmonischen Frequenzen wie in Bild 2(b). Eine Diskussion über die Ursachen für dieser Ungenauigkeit findet weiter unten im Abschnitt 3.2 statt. Eine Möglichkeit, Einfluss auf die Verteilung der Koeffizienten, sowie die Korrespondenz zwischen Skala und Frequenz, zu nehmen ist durch geeignete Wahl der Signalabtastfrequenz, so dass die Bandbreite einer Skala den gewünschten Resonanzbereich abdeckt. Dieser Ansatz ist besonders sinnvoll, wenn der Resonanzbereich eines Prozesses im Voraus bekannt ist.

3.2 Geeignete Eigenschaften für eine Spektralanalyse

Welches Wavelet für eine Signalanalyse am besten geeignet ist, hängt von der Art des Signals und der gesuchten Information ab. Die Suche nach dem richtigen Wavelet erfolgt letztendlich durch empirische Untersuchungen, kann aber unter Berücksichtigung einiger Eigenschaften wie *Kompaktheit, Regularität, Frequenzselektivität* und *Symmetrie* in gezielter Weise stattfinden.

Die ersten drei oben erwähnten Eigenschaften werden von der Anzahl der *verschwindenden Momente* des Wavelets (auch die *Ordnung* eines Wavelets genannt) beeinflusst. Ein Wavelet mit p verschwindenden Momenten ist zu allen Polynomen $(p-1)$-ten Grades und kleiner orthogonal; d. h. das innere Produkt aus dem Polynom und dem Wavelet ergibt den Koeffizient Null. Die Anzahl der verschwindenden Momente beschreibt die Schwingungseigenschaften eines Wavelets: Je höher der Parameter p desto mehr oszilliert das Wavelet und desto mehr Zeit nimmt es in Anspruch. Folglich sind Wavelets hoher Ordnung breiter als die niedriger Ordnung. Daubechies [6] hat bewiesen, dass ein Wavelet mit Ordnung p einen *Träger* (engl. support) mit einer Mindestbreite von $2p-1$ besitzt.

Kompaktheit, oder einen (zeitlich) schmalen Träger, ist eines der grundlegende Merkmale der Wavelet-Funktionen, die gegenüber den unendlich langen Basis-Funktionen der Fourier-Transformation besondere Vorteile bei der Untersuchung nichtstationärer und stochastischer Prozesse bietet. Auf Grund ihrer kompakten Länge können Wavelets die im Signal enthaltene Information innerhalb eines kurzen Zeitraums ermitteln. Wie schmal dieser Zeitraum ist, hängt unter anderem von der Ordnung des Wavelets ab.

Für Informationen über kurzzeitige Änderungen im Signal (z. B. eine impulsartige Störung in der Drehzahl einer Welle mit defektem Lager) kann ein kompaktes Wavelet die besten Ergebnisse liefern. Das rasche Abklingen der Störung, die oftmals einer Singularität im Messsignal gleicht, erfordert ein Wavelet, das den Zeitpunkt der Störung genau erfassen kann. Dies ist besonders der Fall, wenn das Signal viele Singularitäten enthält; denn ein breites Wavelet, das während der Transformation mehrere Singularitäten gleichzeitig abdeckt und/oder überlappt, führt zur überhöhten Koeffizienten und kann die einzelnen Störungen nicht voneinander unterscheiden.

Bild 3 zeigt ein singularitätbehaftetes Signal und die Koeffizienten zweier diskreten Wavelet-Transformationen (DWT) mit dyadischer Skalierung. Die Transformation in Bild 3(b) verwendet ein schmales *Daubechies* Wavelet der Ordnung 2. Bild 3(c) stellt die Koeffizienten einer Transformation mit dem Daubechies Wavelet 10. Ordnung. In beiden Transformationen sorgt die impulsartige Singularität für eine starke Anregung aller Skalen, zu erkennen an die Dichte und Höhe der Koeffizienten zum Zeitpunkt der Singularität am Ende des Signals. Die Koeffizienten des kompakteren Daubechies Wavelets 2. Ordnung sind im Bereich der Singularität jedoch nicht so zerstreut als die des breiteren Daubechies Wavelets 20 Ordnung. Das kompaktere Wavelet lässt besonders in den niedrigen Skalen (z.b. 2^8 oder 2^7) den Zeitpunkt der Singularität aus der Position eines deutlich hervorragenden Koeffizients genau bestimmen. Wenn zwei solche Störungen im Signal kurz hintereinander auftreten, ist es eventuell möglich, dass sie aus er zerstreuten Koeffizienten einer Wavelet Transformation mit einem Wavelet hoher Ordnung nicht getrennt analysiert werden können.

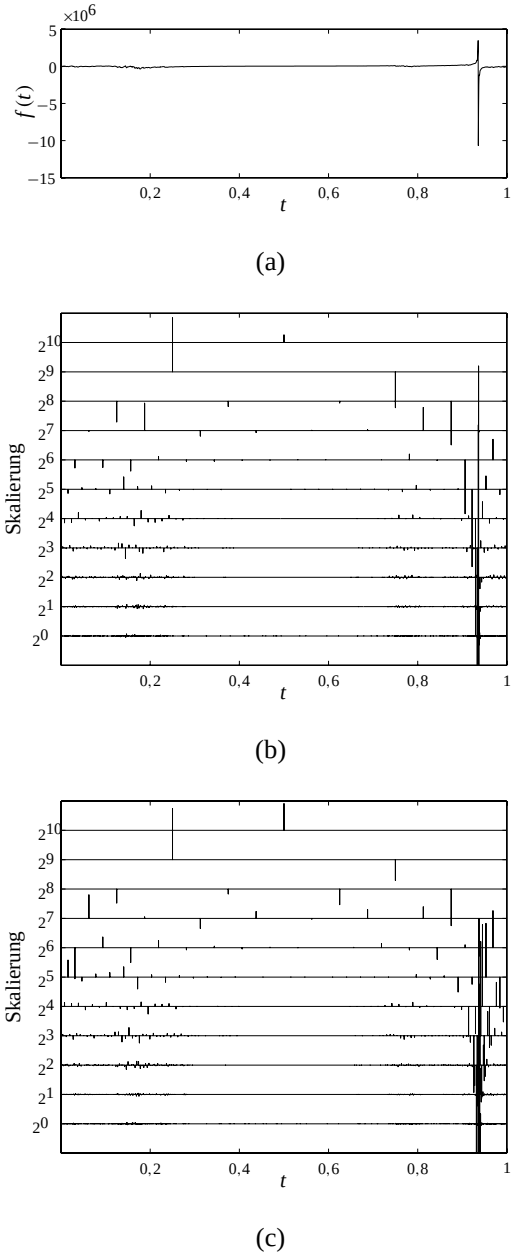
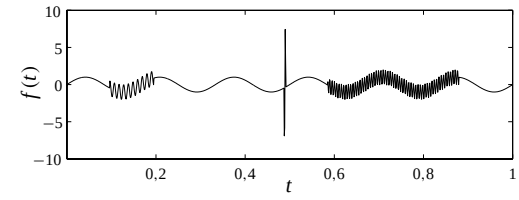
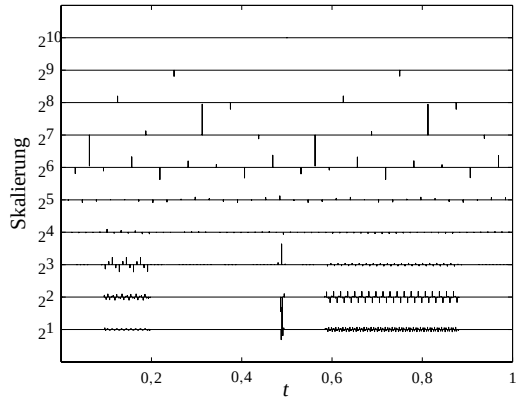


Bild 3: (a) Singulariätbehaftetes Signal und die Koeffizienten einer DWT mit einem Daubechies Wavelet (b) 2. Ordnung und (c) 10. Ordnung.

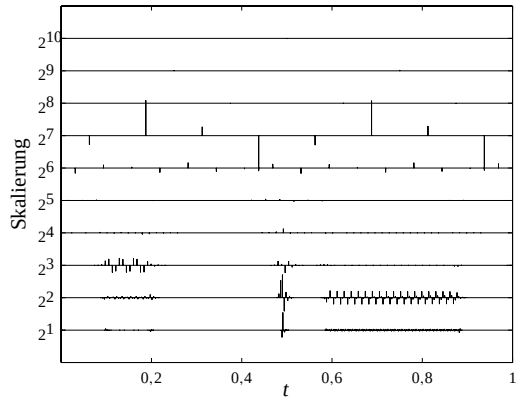
Für viele Anwendungen der Wavelet-Transformation, wie etwa Datenkomprimierung oder Bildverarbeitung, ist die Fähigkeit, ein Signal mit möglichst wenig Koeffizienten ungleich Null darzustellen, ein wichtiger Aspekt. Eine minimale Signalbeschreibung ist von Vorteil, wenn große Datenmengen in kurzer Zeit zu bearbeiten und/oder zu übertragen sind. Obwohl eine Signaltransformation mit einem Wavelet hoher Ordnung generell weniger Koeffizienten generiert, können diese Koeffizienten sehr groß sein (wie oben erwähnt). Wenn auch kleine Änderungen im Signal von Interesse sind, besteht dann die Gefahr, dass sie in der Menge der Koeffizienten nicht wieder gefunden werden können bzw. nicht wieder rekonstruierbar sind. Ein zusätzliche Aspekt einer Wavelet-Transformation mit einem Wavelet höherer Ordnung ist der damit verbundene höhere Rechenaufwand. Die Vorteile von Wavelets mit vielen verschwindenden Momenten zeigen sich besonders bei der Analyse eines regulären Signals, das nur wenige Singularitäten enthält, und bei der gezielten Untersuchung der Frequenzdynamik.



(a)



(b)



(c)

Bild 4: (a) Reguläres Signal und die Koeffizienten einer DWT mit einem Daubechies Wavelet (b) 2. Ordnung und (c) 10. Ordnung.

In Bild 4(a) ist das Beispielsignal aus Abschnitt 3.1 nochmals dargestellt. Das Signal enthält nur fünf Singularitäten – am Anfang und am Ende der kurzen Schwingungen und einen Impuls. Die Bilder 4(b) und 4(c) zeigen die Koeffizienten einer diskreten Wavelet-Transformation dyadischer Skalierung mit einem kompaktem bzw. breitem Wavelet. In beiden Transformationen sind die Eigenschaften der niederfrequenten Dauerschwingung sind in den Koeffizienten der groben Skalen (beispielsweise Skala 2^7) erfasst. Der hochfrequente Impuls und die kurzlebigen Schwingungen ist ebenfalls in beiden Koeffizientendarstellungen in den feinen Skalen (2^3 bis 2^1) im Mittelpunkt der Zeitachse zu erkennen. Jedoch ist die Frequenzlokalisierung des breiteren Daubechies Wavelets 10. Ordnung besser als die des kompakteren Wavelet. Dies ist an der klarerer Zuordnung der Koeffizienten einer Schwingung zu einer bestimmten Skala zu sehen: Beispielsweise sind die 6 Hz Dauerschwingung darstellende Koeffizienten der Transformation mit breitem Wavelet lediglich auf die Skalen 2^7 und 2^6 verteilt. Die 6 Hz Koeffizienten des kompakten Wavelets sind dagegen über fünf Skalen verteilt. Diese Tatbestand geht auf die Frequenzselektivität des breiteren Wavelets zurück.

Wavelets höherer Ordnung besitzen eine bessere Frequenzselektivität als solche mit einem kompakteren Träger. Mehrere Oszillationen und eine breitere Zeitspanne erlauben, mit solchen Wavelets eine genauere Aussage über die Frequenzeigenschaften des Signals zu treffen. Diese Eigenschaft ist basiert auf der Heisenberg'sche Unschärferelation, die aussagt, dass die Varianz um einen Frequenzmittelwert umgekehrt proportional zu der Varianz um einen zeitlichen Mittelwert ist. Mit anderen Worten: Eine gute Frequenzlokalisierung wird mit einer schlechten zeitlichen Lokalisierung erkauft.

Eine letzte Eigenschaft, die in diesem Abschnitt über die Auswahl eines geeigneten Wavelets für die Frequenzanalyse erwähnt werden soll, ist die *Symmetrie*. Diese Eigenschaft bezieht sich auf das Wavelet, oder die Wavelet-Transformation, als Signalfilter. Bei der Filterung eines Signals ist eine minimale oder möglichst gleichmäßige Phasenverschiebung oftmals eine wichtige Bedingung. Eine Phasenverschiebung über einen Teil des Frequenzspektrums führt zu einer Änderung des Signals wenn es rekonstruiert wird. Ein

symmetrisches oder antisymmetrisches Filter mit linearer Phase vermeidet eine zu große Verzerrung des Signals, hat aber den Nachteil, dass auch hier möglich ein großen Rechenaufwand in Kauf genommen werden muss. Ein orthogonales Wavelet mit kompaktem Träger kann kein symmetrisches/antisymmetrisches Filter bilden[1]. Jedoch wurden die *Symmlets* als näherungsweise antisymmetrische Wavelets entwickelt, um die Vorteile einer Filterung mit linearer Phasenverschiebung möglichst ausnutzen zu können.

Eine Vielzahl von Standard-Wavelets stehen in den meisten Wavelet-Software-Paketen zur Verfügung. Unter Berücksichtigung der oben erwähnten Eigenschaften, kann ein am besten geeignetes Wavelet gewählt werden, wobei auch die Implementation, bzw. die Handhabung der zugehörigen Wavelet-Transformation sicherlich eine entscheidende Rolle spielt.

4 Anwendung der diskreten Wavelet-Transformation

Auf Grund ihrer bandpassartigen Zerlegung der Signaleigenschaften eignet sich die diskrete Wavelet-Transformation als Werkzeug der Signalverarbeitung. Die Skalen einer diskreten Wavelet-Transformation existieren entweder als Koeffizientenvektor oder, nach partieller Rücktransformation, als Vektor der Signaleigenschaften zu den entsprechenden Skalen. Diese Verteilung der Signaleigenschaften auf mehrere Vektoren unterschiedlicher Frequenzeigenschaften ermöglicht die gezielte Untersuchung spezifischer Signalelementen.

Einer der Vorteile der diskreten Wavelet-Transformation ist die Möglichkeit, Information in einem langen Datensatz durch weniger Koeffizienten darzustellen. Jedoch existieren Methoden, die die Anzahl der Koeffizienten noch weiter kürzen, so dass eine minimale Repräsentation eines Signals entsteht. Donoho [9] beschreibt die Anwendung von harten und weichen Schwellwerten auf die Wavelet-Koeffizienten für die Unterdrückung von Rauschen in Messsignalen und eine Minimierung des Speicherbedarfs für die erhaltene Information. Die Reduktion eines Signals auf seine wesentlichen Informationen bewirkt eine Hervorhebung besonders bedeutsamer Ereignisse aus dem Signal [10]. Diese Methode der Signalvorverarbeitung läßt eine Manipulation einzelner Koeffizienten zu, zur Unterstützung der Signalparameteridentifikation. Die Vorgehensweise der Koeffizientenreduktion wird nun anhand eines ABS-Raddrehzahlsignals (Bild 5(a)), gemessen an der Hinterachse eines Produktionfahrzeug, vorgestellt.

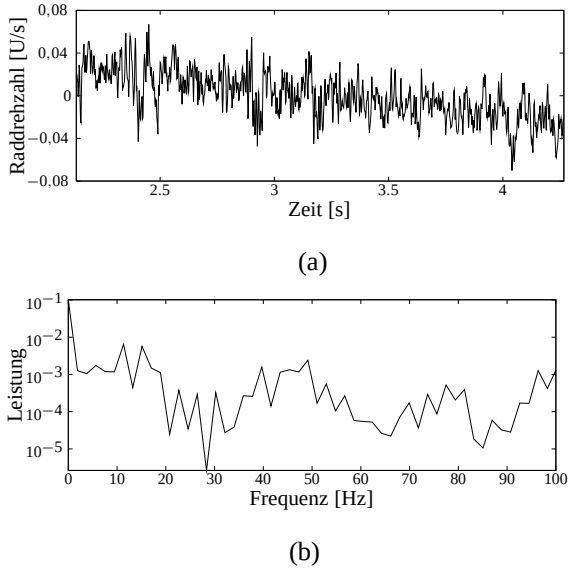


Bild 5: Raddrehzahlsignal (a) und (b) Leistungsspektrum (bis 100 Hz)

Das Signal in Bild 5(a) stellt eine $N = 1024$ Tastschritte langen Ausschnitt eines Mittelwert befreiten Messsignals dar. Die Messung stellt eine Fahrt mit 80 km/h auf Betonpiste

[1] Ausnahmen sind das einfachste „Haar“ Wavelet und Biorthogonal Wavelets [3]

dar und wurde mit einer Taktfrequenz von $F_T = 480$ Hz aufgenommen, was einer Zeitspanne von $2{,}13$ s entspricht. Neben der niederfrequenten Tendenz weist das Leistungsspektrum des Signals (Bild 5(b)) drei Resonanzen bei ca. 12 Hz, 45 Hz und 80 Hz auf, die den Eigenbewegungen des Rades entsprechen [11]. Die erste Resonanz entstammt der vertikalen Bewegung zwischen Reifen und Felge, die nun in dieser Darstellung näher betrachtet werden soll.

4.1 Frequenzbasierte Auswahl der Wavelet-Koeffizienten

Eine Änderung im Resonanzverhalten einer Eigenbewegung gibt Auskunft über den Zustand des Systems. Unebenheiten in der Fahrbahn bewirken eine stärkere Anregung der 12 Hz Eigenschwingung und geben die Möglichkeit, solche Änderungen besser erfassen zu können. Die klassische Trennung der 12 Hz Resonanzeigenschaften vom Spektrum erfolgt durch Anwendung eines Bandpassfilters, jedoch nimmt diese Methode keinerlei Rücksicht auf die zeitliche Lokalisierung der Anregungen (Bild 6).

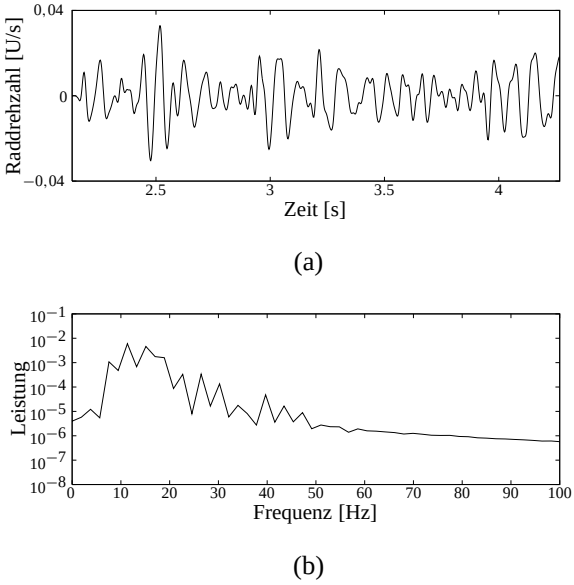


Bild 6: (a) Bandpassgefiltertes ($\omega_e = [7{,}5\text{ Hz},\ 30\text{ Hz}]$) Raddrehzahlsignal und (b) resultierendes Leistungsspektrum.

Die Wavelet-Transformation dagegen erlaubt die Auswahl und die Zusammensetzung bestimmter Koeffizienten mit geeigneten Zeit- und Frequenzeigenschaften, die sowohl den Zeitpunkt der Anregung als auch die ausgelöste Dynamik im rekonstruierten Signal erkennen lassen.

Ein hoher Transformationskoeffizient deutet auf eine starke Anregung, die meistens als Folge impulsartiger Änderungen auftritt z. B. durch Getriebeschäden, Lastwechsel, Stößen oder Gelenkspiel. Für viele technische Anwendungen sind solche Ereignisse häufig mit einer Abweichung vom Normal- oder Sollzustand verbunden und deren Auffindung bzw. Identifikation in der Signalanalyse dient zur Maschinenüberwachung oder Fehlerdiagnose. Die Auswahl einiger der höchsten Transformationskoeffizienten führt zu einem reduzierten Koeffizientenvektor, der, nach einer inversen Wavelet-Transformation, ein Signal ergibt, das die impulsartigen Änderungen deutlich hervorhebt. Benutzt man dieses reduzierte Signal als Eingang zu einem Identifikationsalgorithmus, so können bessere Ergebnisse oder eine frühere Fehlerdetektion erreicht werden, als bei dem entsprechenden Rohsignal, und oftmals mit deutlich weniger Rechenaufwand [12, 13].

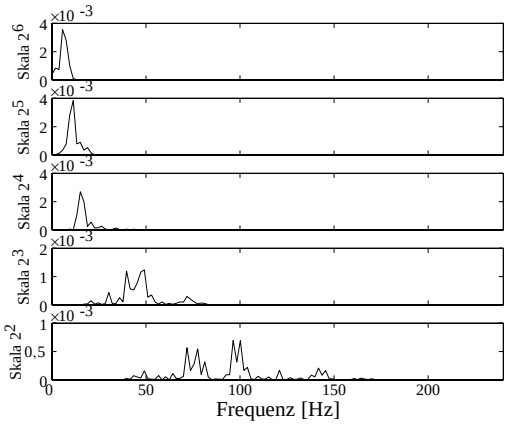
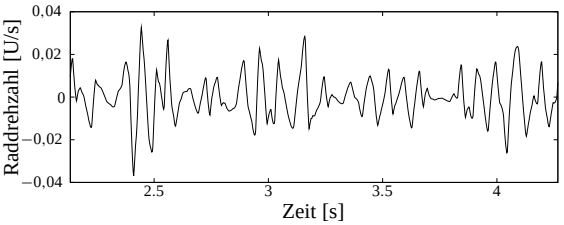


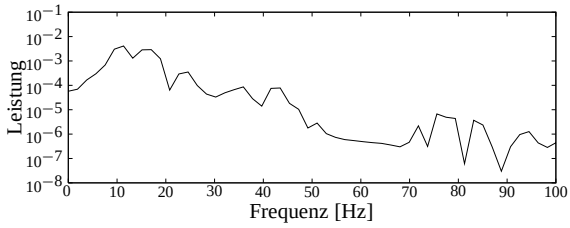
Bild 7: Leistungsspektren der Wavelet-Skalen 2^2 bis 2^6

Zwei Methoden für die Auswahl geeigneter Frequenzeigenschaften aus den Wavelet-Transformationskoeffizienten eines Signals werden hier dargestellt, die zum einen die Frequenzeigenschaften zum anderen die zeitliche Position der Koeffizienten als Reduktionskriterium verwenden.

Eine Spektralanalyse der einzelnen Wavelet-Skalen (Bild 7) verdeutlicht die Aufteilung der Signalfrequenzen. Das Bandpassverhalten der Wavelet-Transformation ist in den Wavelet-Spektren deutlich zu erkennen. Die Resonanz um 12 Hz spiegelt sich in den Skalen 2^5 und 2^4 wider. Die partielle Signalrekonstruktion zu den einzelnen Wavelet-Skalen führt zu der merkbaren Überlappung der Frequenzbänden, denn die Frequenzeigenschaften einer Wavelet-Funktion gleichen nicht einer reinen harmonischen Bandbreite. Es ist trotzdem möglich, wie in Bild 7, mittels Fourier-Transformation das Frequenzspektrum einer Wavelet-Skala eindeutig festzustellen.



(a)



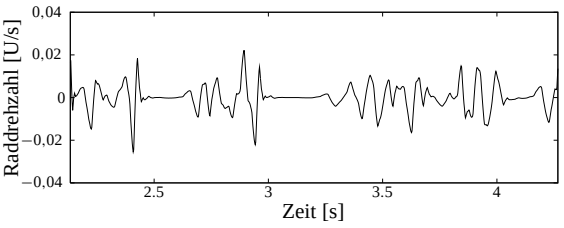
(b)

Bild 8: (a) Frequenzbasierte Kompression des Raddrehzahlsignals und (b) resultierendes Leistungsspektrum.

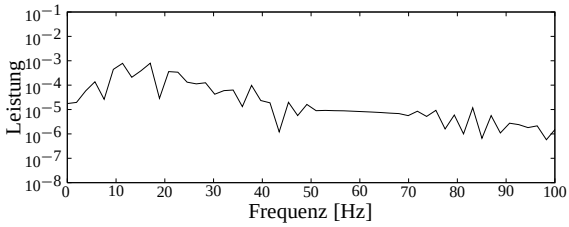
Die Auswahl der wichtigsten Koeffizienten aus den Skalen 2^5 und 2^4 stellt die wesentlichen Eigenschaften der 12 Hz Resonanz dar. Methoden zur Koeffizientenselektion sind beispielsweise Thresholding [9] oder genetische Algorithmen [10]. Ein sehr einfacher Ansatz, der auch eine Art Schwellwert verwendet, nimmt die höchsten Koeffizienten der beiden Skalen für die Rekonstruktion des komprimierten Signals. Bild 8 zeigt das Ergebnis eine Signalrekonstruktion aus den 50 höchsten Koeffizienten, die eine Bandbreite von $7{,}5$ Hz bis 30 Hz darstellen. Die regelmäßigen Abständen der Anregung von den Fahrbahnteerfugen (bei 80 km/h ca. $0{,}6$ s) sind als periodisch erhöhte Amplitude jetzt im Signal sichtbar. Dazu sind die Frequenzen um 12 Hz im Spektrum deutlich hervorgehoben worden, so dass eine Änderung im Vergleich zu anderen Fahrsituationen besser festgestellt werden kann.

4.2 Zeitbasierte Auswahl der Wavelet-Koeffizienten

Wie bereits in Abschnitt 3.1 erklärt, führt die Erfassung der Frequenzen durch Wavelet-Basisfunktionen zu einer Zerlegung des Zeitbereiches, die feiner wird je höher das Frequenzband. Eine Wavelet-Transformation eines Signal der Länge N ergibt einen $N \times 1$ Vektor der Koeffizienten, die nach Skala und zeitlicher Position geordnet sind. Jede Skala 2^j ist durch 2^j Koeffizienten im Vektor vertreten, die gleichmäßig über den Zeitbereich der Transformation verteilt sind. Es ist dadurch möglich, jedem Wavelet-Koeffizient einen Zeitpunkt zuzuordnen. Wavelet-Koeffizienten, die zu einem bestimmten Zeitpunkt gehören, können damit für eine Rekonstruktion der Signaleigenschaften zu diesem Zeitpunkt gezielt ausgesucht und zusammengestellt werden.



(a)



(b)

Bild 9: (a) Zeitbasierte Kompression des Raddrehzahlsignals und (b) resultierendes Leistungsspektrum.

Bild 9(a) zeigt das Ergebnisse eine Signalrekonstruktion aus 48 zeitlich ausgesuchten Koeffizienten. Wie in Abschnitt 4.1 bereits erwähnt stellen die höheren Koeffizienten im Skala 2^5 eine besonders hohe vertikale Anregung des Rades dar. Diese Skala ist durch die Koeffizientenvektorelementen 33 bis 64 dargestellt und der höchste dieser Koeffizienten hat die Vektorposition 38. Skala 2^4 enthält ebenfalls Eigenschaften der 12 Hz Resonanz (Bild 7), und die gleiche Zeitposition wie Koeffizient 38 ist in Skala 2^4 durch den Koeffizienten 76 und 77 dargestellt. Der Rekonstruktionsvektor wurde in diesem Beispiel aus den 16 höchsten Koeffizienten der Skala 2^5 und die 32 entsprechen Koeffizienten der Skala 2^4 zusammengestellt.

Das rekonstruierte Signal (Bild 9(a)) zeigt sehr deutlich vier Gruppen von Schwingungen im Rad, angeregt von den Teerfugen in der Fahrbahn. Tatsächlich sind die 16 höchsten Koeffizienten der Skala 2^5 paarweise gruppiert, was auf die Anregung von der Fahrbahn und die Systemantwort des Radträgers zurückzuführen ist. Die Amplitude dieses Nachschwingens ist größer als die Amplitude der ursprünglichen Anregung (siehe z. B. die Zeitspanne zwischen 2,6 und 3 Sekunden) und deutet auf eine ausgelöste Resonanz im Fahrwerk. Wie auch bei dem Verfahren der frequenzbasierten Koeffizientenauswahl ist der gezielte Resonanzbereich im resultierenden Spektrum deutlich hervorgehoben (Bild 9(b)).

5 Zusammenfassung und Ausblick

Dieser Bericht erklärt und demonstriert zwei Anwendungen der Wavelet-Transformation für die Verarbeitung von Messdaten. Durch die Auswahl geeigneter Transformationskoeffizienten können bestimmte Eigenschaften eines Signals hervorgehoben und näher untersucht werden. Andere Forscher haben berichtet, dass die Verwendung eines reduzierten Signals als Eingang eines Identifikationsalgorithmus eine höhere Genauigkeit des Identifikationsverfahrens liefert [12, 13]. Dieser Bericht stellt zwei Methoden vor, die die Spektrale Eigenschaften des Signals als Selektionskriterium des Reduktionsverfahren verwendet. Die Wavelet-Methoden werden anhand eines ABS-Raddrehzahlsignals präsentiert. Zukünftig soll in diesem Bereich auch waveletbasierte Methoden entwickelt werden, die den physikalischen Zustand einzelner Systemelemente aus den vorhandenen Messsignalen direkt bestimmen lassen.

Literatur

[1] G.M. Jenkins and D.G. Watts. *Spectral Analysis and its Applications*. Holden–Day, San Fransisco, 1968.

[2] A. Papoulis. *The Fourier Integral and its Applications*. Mc Graw-Hill, New York, 1962.

[3] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, 1998.

[4] P. Terwiesch. Zeit-Frequenz-Analyse und Wavelets: Eine Einführung. *Automatisierungstechnik*, 46(1):3–14, 1998.

[5] S. Ellis. Zur Zeit–Frequenz–Analyse von Meßsignalen mittels Wavelet–Verfahren. Diplomarbeit, Fachgebiet Mess-, Steuer- und Regelungstechnik, Gerhard–Mercator–Universität–GH Duisburg, 1998.

[6] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, 1992.

[7] S. Mallat. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674 – 693, 1989.

[8] A. Cohen and J. Kovačević. Wavelets: The Mathematical Background. *Proceedings of the IEEE*, 84(4):514–522, 1996.

[9] D. L. Donoho. Nonlinear Wavelet Methods for Recovery of Signals, Densities and Spectra from Indirect and Noisy Data. In *Symposia in Applied Mathematics*. American Mathematical Society, 1993.

[10] W. J. Staszewski. Wavelet Based Compression and Feature Selection for Vibration Analysis. *Journal of Sound and Vibration*, 211(5):735–760, 1998.

[11] P.W.A. Zegelaar. *The Dynamic Response Of Tyres To Brake Torque Variations And Road Unevennesses*. PhD thesis, Delft University of Technology, 1998.

[12] K. Englehart. *Signal Representation for the Classification of the Transient Myoelectric Signal*. PhD thesis, University of New Brunswick, 1998.

[13] W. J. Staszewski. Structural and Mechanical Damage Detection Using Wavelets. *The Shock and Vibration Digest*, 30(6):457–472, 1998.

Fuzzy-Regelgenerierung und multivariate statistische Verfahren zur Schrittphasenerkennung in der Instrumentellen Ganganalyse

Ralf Mikut, Norbert Peter, Georg Bretthauer

Forschungszentrum Karlsruhe GmbH, Institut für Angewandte Informatik,
D-76021 Karlsruhe, Postfach 3640,
Telefon: (07247) 82-5731, Fax: (07247) 82-5785, E-Mail: mikut@iai.fzk.de

Rüdiger Rupp, Rainer Abel, Andrea Siebel, Hans Jürgen Gerner, Leonhard Döderlein

Orthopädische Universitätsklinik Heidelberg,
D-69118 Heidelberg, Schlierbacher Landstr. 200a,
Tel.: (06221) 96-6322, Fax: (06221) 96-6345, E-Mail:
Ruediger.Rupp@ok.uni-heidelberg.de

1 Motivation

Die Instrumentelle Ganganalyse stellt ein Verfahren zur quantitativen Untersuchung und Dokumentation des menschlichen oder tierischen Ganges dar [1]. Dabei werden die Videodaten von Bewegungsabläufen, die gemessenen Bodenreaktionskräfte sowie die Muskelaktivität simultan aufgezeichnet. Die Methodik erlaubt die Erfassung von dynamischen Bewegungsvorgängen, die in ihrer Komplexität durch einfache optische Kontrolle nicht in ausreichendem Umfang ermittelt werden können. Die so gewonnenen Daten werden zur Diagnose- und Therapieplanung bei Bewegungsstörungen [1–6], zur Untersuchung von Bewegungsmustern bei der funktionellen Elektrostimulation [7, 8], zur Regelung von Prothesen [9] und zur Übertragung der Ergebnisse auf zwei- bzw. mehrbeinige Roboter [10] verwendet. Gegenwärtig erfolgt die Auswertung der Daten noch manuell, was ein hohes Maß an jeweiligem Expertenwissen erfordert. Deswegen gibt es zunehmend Bestrebungen, mit Hilfe klassischer Verfahren der multivariaten Statistik (z. B. Diskriminanzanalyse [11]) oder mit Verfahren der Computational Intelligence (CI - Fuzzy-Logik [12], Künstliche Neuronale Netze [13]) auch die nachfolgenden Auswerteschritte zumindest teilweise zu automatisieren.

Die Vor- und Nachteile von statistischen und Fuzzy-Ansätzen sollen in diesem Beitrag anhand der Schrittphasenerkennung in der Instrumentellen Ganganalyse diskutiert werden. Die Schrittphasenerkennung ist dabei ein Beispiel für eine Klassifikationsaufgabe in der Instrumentellen Ganganalyse [14]. Sie ist eine Voraussetzung für nachfolgende Auswerteverfahren, mit denen langfristig eine Entscheidungsunterstützung und Objektivierung bei Diagnose- und Therapieentscheidungen angestrebt wird.

Ziel dieses Beitrags ist es,

- in die Problemstellungen bei der Instrumentellen Ganganalyse einzuführen,
- die verwendeten Klassifikationsverfahren vorzustellen und
- die Vor- und Nachteile dieser Verfahren anhand der Schrittphasenerkennung zu erläutern.

2 Instrumentelle Ganganalyse

In der Orthopädischen Universitätsklinik Heidelberg besteht das Ziel der Instrumentellen Ganganalyse darin, die Diagnosestellung und Therapieplanung bei neurogenen Gangstörungen (z. B. frühkindliche Hirnschädigung, Querschnittlähmung) zu objektivieren. Beispielsweise ermöglicht ein Laufband (Bild 1) mit einem Gewichtsentlastungssystem, ähnlich einem Fallschirmspringergurt, die Untersuchung von Patienten, die ihr volles Körpergewicht nicht selbst tragen können [15]. Für diagnostische Zwecke können die Patienten genau dosiert belastet werden. Die Analyse der Kräfte in den einzelnen Bewegungsabschnitten der unteren Extremitäten gibt Aussagen darüber, ob Hilfsmittel wie Orthesen, Schuhzurichtungen und zukünftig Neuroprothesen sinnvoll sind.

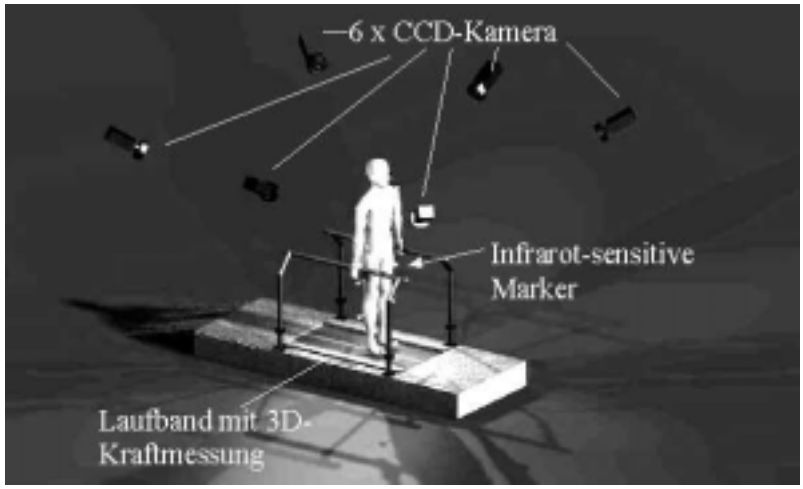


Bild 1: Schematische Darstellung der Datenaufnahme mit Laufband

Das bislang allgemein übliche untersucherabhängige Vorgehen setzt ein hohes Maß an klinischer Erfahrung voraus. Zur Festlegung und Überprüfung von Therapiestrategien ist aber eine objektive Diagnosestellung notwendig. Das mittelfristige Ziel des hier beschriebenen Gemeinschaftsprojektes zwischen der Orthopädischen Universitätsklinik Heidelberg und dem Forschungszentrum Karlsruhe besteht darin, aus der Gesamtheit der gemessenen Bewegungsgrößen und den individuellen Diagnoseentscheidungen nachvollziehbare, qualitätsstandardisierte Entscheidungsregeln automatisch zu generieren.

Eine wichtige Voraussetzung aus diagnostischer Sicht ist dabei, Schrittphasen im Bewegungsmuster eines Patienten zu erkennen und eventuell individuelle Besonderheiten innerhalb der einzelnen Schrittphasen zu formulieren. Diese Information dient einer späteren Bildung neuer Merkmale, die aus einer zusammenfassenden Beschreibung der Bewegung innerhalb einer Schrittphase entstehen.

Innerhalb eines Schrittzyklusses werden nach [1] sieben Schrittphasen unterschieden ($m_y = 7$ Klassen, Bild 2). Die erste Phase (*Loading Response*) beginnt mit dem Aufsetzen des betrachteten Fußes auf dem Boden (*Initial Contact*). Diese Phase umfasst etwa 0-10 % des Schrittzyklusses (SZ) und endet, wenn die Zehe des anderen Fußes vom Boden abhebt. Die zweite Phase (*Midstance*) enthält den Zeitraum, bis sich der Schwerpunkt des Körpers über dem betrachtetem Vorderfuß befindet (10-30% SZ). In der dritten Phase (*Terminal Stance*) ist nach wie vor nur der betrachtete

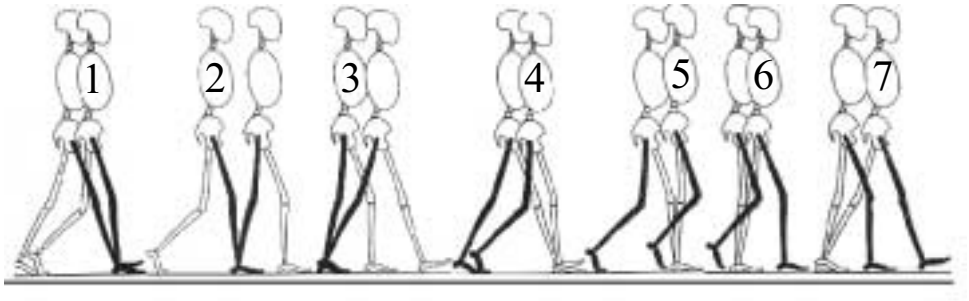


Bild 2: Schrittphasen - rechte (R) Körperseite (1: Initial Contact und Loading Response (LRE), 2: Mid Stance (MST), 3: Terminal Stance (TST), 4: Pre Swing (PSW), 5: Initial Swing (ISW), 6: Mid Swing (MSW), 7: Terminal Swing (TSW); nach [1])

Fuß am Boden (30-50% des SZ). Das Aufsetzen des anderen Fußes leitet zur vierten Phase über, wobei der betrachtete Fuß mit der Ferse abhebt (*Pre Swing*, 50-60% SZ). Der andere Fuß befindet sich zu dieser Zeit in der ersten Phase. Wenn die Zehe des betrachteten Fußes abhebt, beginnt die fünfte Phase (*Initial Swing*). Diese endet, wenn sich das Schwungbein auf Höhe des Standbeins befindet (60-73% SZ). Die sechste Phase (*Mid Swing*, 73-87% SZ) umfasst den Zeitraum, bis das Schienbein senkrecht zum Boden steht. In der abschließenden siebenten Phase (*Terminal Swing*, 87-100% SZ) wird die Schwungphase bis hin zum Aufsetzen der Ferse beendet.

Für die Erhebung der ganganalytischen Daten kommt ein markergestütztes Infrarot-Bewegungsanalysesystem der Firma Motion Analysis mit sechs Kameras und ein zum größten Teil in Heidelberg entwickeltes Diagnostiklaufband [2] mit der Möglichkeit zur dreidimensionalen Bodenreaktionskraftmessung zum Einsatz (Bild 1). Die Daten werden in einer ersten Auswertestufe mit Hilfe des Programmpakets OrthoTrak 4.1 der Firma MotionAnalysis verarbeitet [16]. Die Videokameras nehmen die Positionen der reflektierenden Marker in einem festen Raumkoordinatensystem auf. Die Auswertung der Daten beruht auf einem stark vereinfachten Körpermodell. Dieses Körpermodell enthält Annahmen über die Bewegungsmöglichkeiten zwischen verschiedenen, in sich starren Körpersegmenten (z. B. Fuß als Platte) sowie deren Abmessungen, Massen und Trägheitsmomente. Aus den gemessenen Markerpositionen können so zunächst Gelenkwinkel berechnet werden. Aus den Gelenkwinkeln und den ebenfalls gemessenen Bodenreaktionskräften ergeben sich dann Kräfte, Momente und die mechanische Arbeit an den einzelnen Gelenken.

Die Datensätze enthalten somit für verschiedene Gelenke (Fußgelenk, Knie, Hüfte, Becken, Hals, Schulter usw.) jeweils Schätzungen für die Gelenkwinkel, die Kräfte, die Momente und die mechanische Arbeit in drei verschiedenen Projektionsebenen (frontal - von vorn; sagittal - von der Seite, transversal - von oben). Daraus entsteht zunächst ein Datensatz mit 87 Merkmalen, aus denen noch zusätzliche Merkmale berechnet werden, wie z. B. die zeitliche Änderung eines Merkmals:

$$\Delta x_l[k] \;=\; \frac{1}{2}(x_l[k+1] - x_l[k-1]). \tag{1}$$

Dieser Wert ist bei äquidistanter Abtastung proportional der Änderungsgeschwindigkeit des jeweiligen Merkmals. Somit ergeben sich nun 174 Merkmale. Diese Daten lassen sich als Matrix der Eingangsgrößen $\boldsymbol{X}$ mit $k = 1, \cdots, N$ Messungen und $l = 1, \cdots, s$ Merkmalen darstellen.

Für den zur Verfügung stehenden Datensatz wurden alle 174 Merkmale verwendet. Der Datensatz umfasst jeweils ca. zehn Schritte von acht gesunden Probanden bei unterschiedlichen Geschwindigkeiten (selbst gewählte Normalgeschwindigkeit, langsames Gehen, schnelles Gehen). Der Datensatz wurde durch zufällige Auswahl in einen Lerndatensatz und einen Testdatensatz zur Validierung aufgespalten.

Weil eine manuelle Klassifikation der Ausgangsklassen für den Lerndatensatz nicht zur Verfügung stand, wurde diese Klassifikation über die in [1] angegebene prozentuale Aufteilung angesetzt. Außerdem wurde ein datenbasierter Ansatz zur Erkennung von Schritten verwendet. Diese Ansätze bewirken allerdings systematische Fehler im Lerndatensatz. Aufgrund der Ungenauigkeiten in der Positionierung der Marker am Körper, der Messfehler bei der Positionsbestimmung der Marker und der vereinfachten Modelle kommt es ebenfalls zu Abweichungen der gemessenen Merkmale von der realen Bewegung. Folglich sind einige Merkmale (z. B. Knierotation usw.) wegen schlechter Stör-Nutz-Signalverhältnisse wenig aussagekräftig.

Weder die einzelnen Schrittphasen noch die Übergangsereignisse zwischen den Phasen lassen sich manuell problemlos aus den Messdaten erkennen. Damit ist dieses Problem repräsentativ für ähnliche Probleme in der Instrumentellen Ganganalyse, wie z. B. die Diagnose unterschiedlicher Erkrankungen oder den Vergleich pro- und postoperativer Daten des gleichen Patienten zur Beurteilung von Operationsergebnissen usw. Im Folgenden soll gezeigt werden, wie mit Hilfe statistischer und CI-Verfahren eine teilautomatisierte Auswertung erfolgen kann.

3 Schrittphasenerkennung mit multivariater Statistik

In einem ersten Arbeitsschritt sind zunächst wichtige Merkmale für die Klassifikation herauszufinden. Das Ziel besteht dabei darin, eine kleine Anzahl s_m (z. B. $s_m = 6$) aus den potentiellen $s = 174$ Merkmalen auszuwählen und dabei möglichst wenig Information zu verlieren. Dazu wird mit dem MANOVA-Verfahren (Multivariate ANalysis Of VAriance [17]) nach einer Gruppe von s_m Merkmalen gesucht, die *in ihrem Zusammenwirken* besonders wichtig sind. Dieses Vorgehen liefert bessere Ergebnisse als das Heraussuchen der wichtigsten Merkmale ohne die Analyse ihres Zusammenwirkens, weil Redundanzen (korrelierte Merkmale) mit betrachtet werden. Ein Experte kann zwar verbale Hinweise geben (Im Beispiel: Betrachtung von der Seite wichtiger als von vorn, Gelenkwinkel am Knie wichtiger als Gelenkwinkel am Fußgelenk usw.), eine quantitative Betrachtung oder gar eine Analyse von Redundanzen überfordert jedoch den medizinischen Experten. Auf der Basis dieser Merkmale werden dann mit Hilfe der Diskriminanzanalyse durch Linearkombinationen neue Merkmale erzeugt. Die Erfolgschancen einer manuellen Suche nach solchen Merkmalen durch Experten sind ebenfalls sehr gering.

Das MANOVA-Verfahren und die Diskriminanzanalyse bauen auf der Untersuchung der Kovarianzmatrizen der Eingangsgrößen auf [17]. Sie beruhen auf der Annahme, dass jede Klasse der Ausgangsgröße (Schrittphase) $y = B_j$, $j = 1, \ldots, m_y$, näherungsweise durch eine mehrdimensionale Normalverteilung der l reellwertigen Eingangsgrößen $\boldsymbol{x}$ beschrieben werden kann. Im Unterschied zu anderen Verfahren (z. B. Hauptkomponentenanalyse) gehen sowohl die Eingangsgrößen als auch die bekannten Ausgangsgrößen (Klassen - hier: Schrittphasen) des Lerndatensatzes ein.

Eine Gruppe von Merkmalen $\mathcal{I}$ ist umso aussagekräftiger, je dichter die Messwerte der gleichen Klasse zusammen liegen und je weiter Messwerte unterschiedlicher Klassen auseinander liegen. Um ein Maß für diese Forderung zu gewinnen, werden aus der Matrix der Eingangsgrößen $\boldsymbol{X}$ diejenigen s_m Merkmale ausgesucht, die in der Indexmenge $\mathcal{I}$ enthalten sind. Daraus ergibt sich eine neue Matrix $\tilde{\boldsymbol{X}}$ mit N Messwerten und s_m Merkmalen. Die Schätzung für deren Kovarianzmatrix berechnet sich aus

$$\hat{\boldsymbol{S}} = \frac{1}{N} \cdot (\tilde{\boldsymbol{X}} - \frac{1}{N}\mathbf{1}_{N,N} \cdot \tilde{\boldsymbol{X}})^T \cdot (\tilde{\boldsymbol{X}} - \frac{1}{N}\mathbf{1}_{N,N} \cdot \tilde{\boldsymbol{X}}) \tag{2a}$$

$$= \frac{1}{N} \cdot \tilde{\boldsymbol{X}}^T \cdot \boldsymbol{Z}_{N,N} \cdot \tilde{\boldsymbol{X}} \qquad \boldsymbol{Z}_{N,N} = \boldsymbol{I}_{N,N} - \frac{1}{N}\mathbf{1}_{N,N}. \tag{2b}$$

Dabei bezeichnen $\boldsymbol{I}$ die Einheitsmatrix und $\mathbf{1}$ Matrizen, die aus Eins-Elementen bestehen. Für jede der m_y Klassen der Ausgangsgröße (im Beispiel $m_y = 7$) wird außerdem eine Klassenkovarianzmatrix der Matrix $\boldsymbol{X}_j$ geschätzt, für die jeweils diejenigen N_j Datensätze verwendet werden, die zur j-ten Klasse gehören:

$$\hat{\boldsymbol{S}}_j = \frac{1}{N_j} \cdot \tilde{\boldsymbol{X}}_j^T \cdot \boldsymbol{Z}_{N_j,N_j} \cdot \tilde{\boldsymbol{X}}_j. \tag{3}$$

Aus diesen Schätzungen lassen sich die Gesamtvariationsmatrix $\boldsymbol{T}$ (Total Variance) als Maß für die Verteilung des gesamten Datenmaterials, die Innerklassenvariationsmatrix $\boldsymbol{W}$ (Within-Groups-Variance) für die Verteilung innerhalb der einzelnen Klassen und die Zwischenklassenvariationsmatrix $\boldsymbol{B}$ (Between-Groups-Variance) für die Verteilung zwischen den Klassen gewinnen:

$$\boldsymbol{T} = N \cdot \hat{\boldsymbol{S}} \qquad \boldsymbol{W} = \sum_{j=1}^{m_y} N_j \cdot \hat{\boldsymbol{S}}_j \tag{4a}$$

$$\boldsymbol{B} = \sum_{j=1}^{m_y} N_j \cdot (\frac{1}{N_j}\mathbf{1}_{1,N_j} \cdot \tilde{\boldsymbol{X}}_j - \frac{1}{N}\mathbf{1}_{1,N} \cdot \tilde{\boldsymbol{X}})^T \cdot (\frac{1}{N_j}\mathbf{1}_{1,N_j} \cdot \tilde{\boldsymbol{X}}_j - \frac{1}{N}\mathbf{1}_{1,N} \cdot \tilde{\boldsymbol{X}}). \tag{4b}$$

Dabei gilt der Streuungszerlegungssatz $\boldsymbol{T} = \boldsymbol{B} + \boldsymbol{W}$. Je größer $\boldsymbol{B}$ im Verhältnis zu $\boldsymbol{W}$ ist, desto besser eignen sich die Merkmale und erfüllen die oben genannte Forderung zur Lage von Messungen gleicher und unterschiedlicher Klassen. Die Lösung des Problems führt auf ein verallgemeinertes Eigenwertproblem, das sich bei Invertierbarkeit von $\boldsymbol{W}$ in ein klassisches Eigenwertproblem umwandeln lässt:

$$(\boldsymbol{B} - \lambda\boldsymbol{W})\boldsymbol{v} = \mathbf{0} \qquad (\boldsymbol{W}^{-1}\boldsymbol{B} - \lambda\boldsymbol{I})\boldsymbol{v} = \mathbf{0}. \tag{5}$$

Auf den sortierten Eigenwerten ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{s_m} \geq 0$) von (5) bauen nun verschiedene Maße $M_{\mathcal{I}}$ zur Beurteilung der Merkmale in $\mathcal{I}$ auf. Das Ziel aller Maße ist es, große Eigenwerte zu favorisieren, wobei es dabei insbesondere auf den oder die ersten Eigenwerte ankommt. Im Sonderfall $s_m = 1$ entsprechen alle Tests dem t-Test. Ein häufig verwendetes Maß ist das Likelihood-Quotienten-Kriterium:

$$M_{\mathcal{I}} = 1 - \prod_{i=1}^{s_m} \frac{1}{1 + \lambda_i}. \tag{6}$$

Mit dem Kriterium können nun verschiedene Hypothesen sinnvoller Merkmalskombinationen $\mathcal{I}$ geprüft werden. Ein suboptimales Verfahren mit geringem Rechenaufwand sucht zunächst das beste Einzelmerkmal ($s_m = 1$) heraus. Anschließend wird

Merkmal	Bezeichnung	Güte
x_3	R-HIP Flex ANG	0.726
x_6	R-KNEE Flex ANG	0.721
x_{90}	R-HIP Flex ANG V	0.718
x_{102}	L-HIP Flex ANG V	0.700
x_{15}	L-HIP Flex ANG	0.680
x_{93}	R-KNEE Flex ANG V	0.639

Merkmal	Bezeichnung	Güte
x_3	R-HIP Flex ANG	0.726
$+x_6$	R-KNEE Flex ANG	0.939
$+x_{102}$	L-HIP Flex ANG V	0.977
$+x_{105}$	L-KNEE Flex ANG V	0.991
$+x_{93}$	R-KNEE Flex ANG V	0.995
$+x_{15}$	L-HIP Flex ANG	0.997

Tabelle 1: Beste Merkmalsrelevanzen der Einzelmerkmale (links) und von Merkmalsgruppen (rechts) mit dem Likelihood-Quotienten-Test - Bezeichnungen: Knie (KNEE), Hüfte (HIP), links (L), rechts (R), Flexion in der Sagittalebene (Flex), Winkel (ANG) und Änderungsgeschwindigkeit (V)

iterativ zur bisher gefundenen besten Merkmalskombination immer das Merkmal gesucht, das den Wert von (6) maximiert. Im Beispiel führt diese Auswahl dazu, dass unter den sechs wichtigsten Einzelmerkmalen und Merkmalskombinationen nur Gelenkwinkel und ihre Änderungsgeschwindigkeiten bei einer Betrachtung von der Seite (sagittal) im Knie- und Hüftbereich zu finden sind (Tabelle 1). Das relevanteste Einzelmerkmal ist der Beugungswinkel des Hüftgelenks auf der rechten Seite (x_3, Winkel zwischen rechtem Oberschenkel und Oberkörper). Das dazu am besten passende Merkmal ist der Beugungswinkel des rechten Kniegelenks (x_6). Das dritte Merkmal ist nicht das drittbeste Einzelmerkmal (x_{90} - Änderungsgeschwindigkeit des rechten Hüftwinkels), weil dieses wiederum sehr stark mit Merkmal x_6 korreliert und somit kaum neue Informationen liefert. Stattdessen werden nacheinander die Merkmale x_{102} (Winkelgeschwindigkeit der linken Hüftseite - 4. Einzelmerkmal), x_{105} (Winkelgeschwindigkeit des linken Knies, 7. Einzelmerkmal), x_{93} (Winkelgeschwindigkeit des rechten Knies, 6. Einzelmerkmal) und x_{15} (Winkel der linken Hüftseite, 5. Einzelmerkmal) hinzugefügt. Die erreichten Werte der Merkmalsrelevanzen auf einer Skale zwischen 0 und 1 zeigen, dass diese Kombination aus sechs Merkmalen relevante Ergebnisse liefert.

Andere Einzelmerkmale (Betrachtung von vorn und oben, Kräfte usw.) werden drastisch schlechter bewertet. Damit liefert diese Merkmalsauswahl auch einen Beitrag zur Interpretierbarkeit der Lösungen, indem sie über eine sortierte Merkmalsliste Auskunft über wesentliche, redundante und irrelevante Merkmale gibt.

Die Matrix der ersten s_d Eigenvektoren $\boldsymbol{V}$ des Problems (5), die zu den s_d größten Eigenwerten gehören, ist die Transformationsmatrix für die Diskriminanzanalyse. Damit entsteht aus einer Linearkombination der ursprünglichen Merkmale ein niederdimensionaler ($s_d \leq s_m$) Raum neuer Merkmale

$$\tilde{\boldsymbol{X}}_D = \tilde{\boldsymbol{X}} \cdot \boldsymbol{V}. \tag{7}$$

Auf diese Art werden zunächst aus den ursprünglichen s Merkmalen s_m Merkmale ausgewählt, aus denen mittels einer Linearkombination s_d neue Merkmale resultieren. Dieses Vorgehen lässt sich auch als eine Einschritt-Transformation

$$\tilde{\boldsymbol{X}}_D = \boldsymbol{X} \cdot \boldsymbol{V}_{ges} = \boldsymbol{X} \cdot \tilde{\boldsymbol{V}} \cdot \boldsymbol{V} \tag{8}$$

darstellen, bei dem sich die Gesamt-Transformationsmatrix $\boldsymbol{V}_{ges}$ aus der Transformations-Matrix zur Merkmalsauswahl $\tilde{\boldsymbol{V}}$ (Dimension $s \times s_m$ mit je einer Eins pro

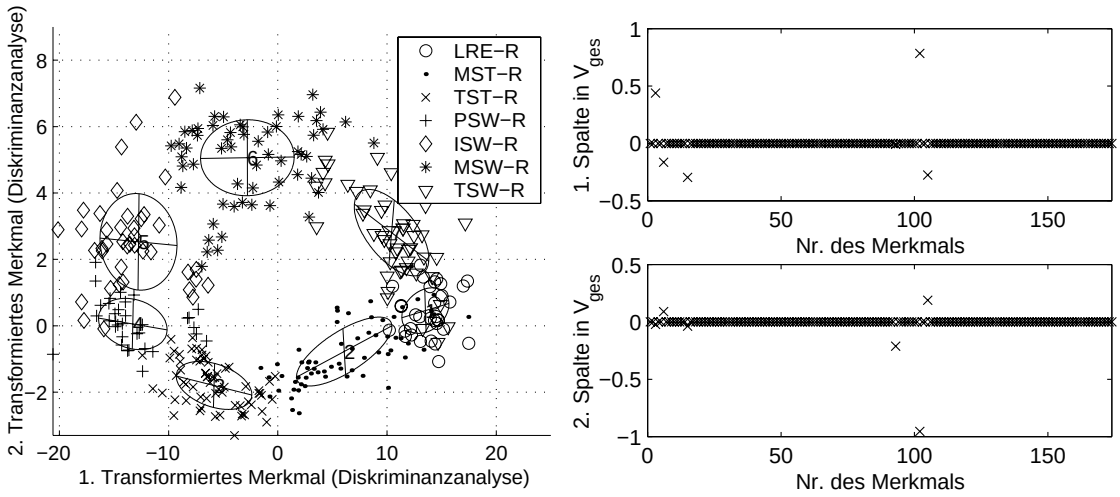


Bild 3: Klassifikation der Schrittphasen in einem zweidimensionalen Merkmalsraum ($\tilde{x}_{D,1}$,$\tilde{x}_{D,2}$) nach einer Diskriminanzanalyse mit vorgeschalteter Merkmalsreduktion auf sechs Merkmale (links) und zugehörige Gesamt-Transformationsmatrix V_{ges} gemäß (8)

Spalte in den Zeilen der ausgewählten Merkmale und sonst Nullen) und der Transformationsmatrix der Diskriminanzanalyse $\boldsymbol{V}$ ergibt. Auf den so transformierten, niederdimensionalen Daten setzt dann ein Bayes-Klassifikator auf, der eine Metrik auf der Basis der ausgangsklassenspezifischen Kovarianzmatrizen verwendet [18].

Die Ergebnisse für das Beispiel zeigt Bild 3 (links). Der Schrittzyklus ist deutlich durch die Kreisform der Datenprojektion erkennbar, beginnt unten rechts mit dem Zustand LRE-R und wird im Uhrzeigersinn durchlaufen. Die benachbarten Phasen sind qualitativ voneinander trennbar. Die Kovarianzmatrizen der einzelnen Schrittphasen sind zur Kennzeichnung der Metrik der Bayes-Klassifikatoren ebenfalls eingezeichnet. Die auftretenden Abweichungen zwischen Lerndatensatz und Klassifikatorergebnissen sind zum Teil durch die in Abschnitt 2 diskutierten Probleme erklärbar (Fehler Lerndatensatz 16.3%, Testdatensatz 18.2%) und betreffen bis auf 0.5% Fehler nur Abweichungen zur benachbarten Phase. Dennoch ergibt sich eine gute Klassifikation der Schrittphasen (Bild 6 in Abschnitt 5).

Im rechten Teilbild von Bild 3 sind die Werte der Transformationsmatrix $\boldsymbol{V}_{ges}$ dargestellt (Transformationsvorschrift für das erste Merkmal $\tilde{x}_{D,1}$ (oben) und das zweite Merkmal $\tilde{x}_{D,2}$ - unten). Sie hat nur für die ausgewählten Merkmale Werte ungleich Null und erlaubt noch Interpretationsversuche. So ist beispielsweise die frühe Schwungphase (Schrittphasen 4 und 5) durch kleine Werte des transformierten Merkmals $\tilde{x}_{D,1}$ charakterisiert, in das mit positivem Vorzeichen die Merkmale x_3 und x_{102} bzw. mit negativem Vorzeichen die Merkmale x_6, x_{15} und x_{105} eingehen. Die Stellung der Gelenke ist in diesen Phasen tendenziell durch einen gestreckten rechten Oberschenkel (rechtes Hüftgelenk, x_3), einen eher gebeugten linken Oberschenkel (linkes Hüftgelenk, x_{15}) und ein gebeugtes rechtes Knie (x_6) gekennzeichnet. Die Dynamik der Bewegung kommt hauptsächlich durch eine Streckung im linken Hüftgelenk (x_{102}) zum Ausdruck.

Die so gefundene abstrakte Darstellung ist als Vorverarbeitungsstufe zur Informationsreduktion für den Vergleich zwischen verschiedenen Patientengruppen ebenso einsetzbar wie zum Vergleich der Auswirkungen unterschiedlicher Geschwindigkeiten im Gangbild (engerer Kreis in Bild 3 entspricht geringerer Geschwindigkeit).

4 Schrittphasenerkennung mit Fuzzy-Systemen

Für den Entwurf eines Fuzzy-Systems kommt ebenfalls ein Verfahren zum Einsatz, das zunächst eine Merkmalsreduktion vornimmt [19, 20]. In Ergänzung dazu erfolgt eine automatische Generierung der Eingangs-Zugehörigkeitsfunktionen [19] und der Fuzzy-Regeln [21, 22]. Da alle Verfahren bereits ausführlich vorgestellt wurden, soll hier nur der prinzipielle Ablauf skizziert und auf die Besonderheiten im Vergleich zu anderen Verfahren hingewiesen werden.

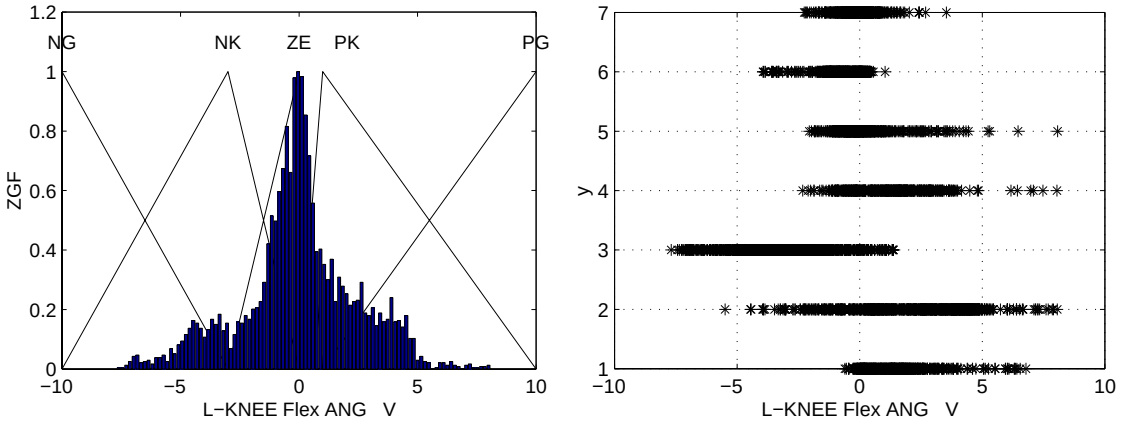


Bild 4: Automatisch entworfene Zugehörigkeitsfunktionen für das Merkmal x_{105} mit Histogramm (links) und Werte des Merkmals x_{105} für die sieben Schrittphasen (rechts)

Die methodische Basis für den Entwurf der Zugehörigkeitsfunktionen (ZGF) und die Merkmalsauswahl sind informationstheoretische Maße, wie Eingangs- und Ausgangsentropie sowie die Transinformation. Diese Maße setzen ursprünglich diskrete Verteilungen voraus, weshalb bei Fuzzy-Systemen spezielle Modifikationen zur Bestimmung der Häufigkeiten diskretisierter Klassen (linguistische Terme des Fuzzy-Systems) und ihrer Verbundverteilungen erforderlich sind. Damit können auf einem relativ hohen Abstraktionsniveau vermutete Abhängigkeiten zwischen den verschiedenen Merkmalen und der Ausgangsgröße untersucht werden, ohne bereits nach konkreten Fuzzy-Regeln suchen zu müssen.

Im Folgenden werden dreieckförmige Zugehörigkeitsfunktionen in der Mitte und trapezförmige Zugehörigkeitsfunktionen am Rand eingesetzt. Wenn die Zugehörigkeitsfunktionen jeweils Zugehörigkeitswerte zwischen Null und Eins aufweisen, sich zu Eins ergänzen und nur einfach überlappen, reicht pro linguistischem Term ein skalarer Parameter zur Beschreibung der jeweiligen Zugehörigkeitsfunktion aus (Punkt mit Funktionswert Eins der Zugehörigkeitsfunktion). Im Beispiel aus Bild 4 sind das folglich die Parameter -10, -3, 0, 1 und 10.

Ausgehend von einer Startverteilung der Parameter, die eine gleichmäßige Verteilung der Daten auf die linguistischen Terme garantiert, wird jeder Parameter nacheinander gelöscht und durch einen neuen Parameter ersetzt. Dieser Parameter wird so gewählt, dass er ein Kompromisskriterium aus gleichmäßiger Verteilung der Daten und nicht zu geringen Parameterabständen (Maximierung der Eingangsentropie), eine möglichst starke Abhängigkeit von den Ausgangsklassen (Maximierung Transinformation) bei guter Interpretierbarkeit (spezielles Maß, das interpretierbare Werte wie 0.00 gegenüber anderen Werten wie -0.0147 bevorzugt) sichert. Die linguistischen Terme werden dabei automatisch mit Namen wie Null (ZE), Positiv Klein

(PK), Positiv Mittel (PM), Positiv Groß (PG), Positiv Sehr Groß (PSG), Negativ (N) usw. versehen.

Die Problematik dieses Kompromisses verdeutlicht Bild 4 anhand der automatischen Festlegung der Änderungsgeschwindigkeit der linken Kniebeugung (Merkmal x_{105}). Zu berücksichtigen ist dabei insbesondere, dass

- im Bereich um Null eine große Datendichte auftritt,
- die Platzierung der Parameter von ZE bei 0 und von PK bei 1 dazu führt, die Schrittphase $y = 6$ zu separieren,
- die Platzierung der Parameter von ZE bei 0 und von NK bei -3 dazu führt, die Schrittphasen $y = 4, y = 5$ und $y = 7$ gut von $y = 6$, $y = 2$ und $y = 3$ zu separieren usw.

Nach erfolgter Fuzzifizierung werden Entscheidungsbäume [23] generiert. Ein Entscheidungsbaum ist ein gerichteter Graph. Er besteht aus Knoten und gerichteten Zweigen, die die Knoten miteinander verbinden. In der obersten Hierarchieebene existiert nur ein Knoten (Wurzelknoten). Zweige gibt es nur von Knoten einer höheren zur nächsttieferen Hierarchieebene. Jeder Knoten legt für die Ausgangsgröße y einen linguistischen Term (Klasse) fest, der für die zugehörige Beispielmenge die beste Entscheidung darstellt. Bei Bedarf wird ein Merkmal x_l ausgewählt, dessen Wert entscheidet, welcher Zweig weiterverfolgt wird. Da auch hier die Merkmale bereits durch relative Häufigkeiten diskreter Merkmale charakterisiert sind, muss der Entwurf von Zugehörigkeitsfunktionen vor der Merkmalsreduktion erfolgen.

Ein spezielles Verfahren ist das ID3-Verfahren, das die Auswahl des Merkmals x_l über eine Maximierung der Transinformation vornimmt. Dabei entsteht als Nebenprodukt beim Entwurf des Entscheidungsbaums in jedem Knoten eine Merkmalsrelevanz für die aktuelle Teil-Klassifikationsaufgabe. Die Merkmalsrelevanz im Wurzelknoten weist Parallelen zur Auswahl eines Einzelmerkmals mit univariaten statistischen Verfahren auf. Diese Relevanz berücksichtigt die spezielle Eignung des Merkmals für den Entwurf von Fuzzy-Systemen, weil sie Unterschiede in der Verteilung von Ausgangsklassen bei linguistischen Termen der Eingangsklasse bewertet. Der linke Teil der Tabelle 2 zeigt die Merkmalsrelevanzen im Wurzelknoten des Entscheidungsbaums. Auch diese Maße nehmen wieder Werte zwischen Null (irrelevant) und Eins (relevant) an. Das Verfahren kommt zu ähnlichen Ergebnissen wie der $t - Test$ in Tabelle 1, links - wobei die konkreten Zahlenwerte keinen direkten Vergleich zwischen beiden Verfahren zulassen. Wichtig sind auch hier Merkmale, die in der sagittalen Ebene Knie- und Hüftgelenkwinkel und ihre Änderungsgeschwindigkeiten bewerten, wobei sich die Reihenfolge der Merkmale gegenüber dem t-Test geringfügig unterscheidet. Dabei werden im Folgenden 10 Merkmale verwendet, weil Fuzzy-Verfahren für eine gute Klassifikationsgüte erfahrungsgemäß eine größere Anzahl von Merkmalen im Vergleich zu statistischen Verfahren benötigen.

Die Merkmalsrelevanzen in Knoten tieferer Hierarchieebenen berücksichtigen hingegen Redundanzen zwischen mehreren Merkmalen. Eine gewichtete Summe von Merkmalsrelevanzen aller Knoten eines Entscheidungsbaums ist somit ein Maß für die multivariaten Relevanzen verschiedener Merkmale. Das Verfahren kann noch dadurch verfeinert werden, dass mehrere Entscheidungsbäume generiert und bezüglich

Merkmal	Bezeichnung	Güte
x_6	R-KNEE Flex ANG	0.276
x_{90}	R-HIP Flex ANG V	0.273
x_3	R-HIP Flex ANG	0.261
x_{102}	L-HIP Flex ANG V	0.260
x_{93}	R-KNEE Flex ANG V	0.246
x_{15}	L-HIP Flex ANG	0.245
x_{21}	L-ANK Flex ANG	0.239
x_{105}	L-KNEE Flex ANG V	0.233
x_{18}	L-KNEE Flex ANG	0.212
x_{14}	L-HIP Abd ANG	0.211

Merkmal	Bezeichnung	Güte
x_6	R-KNEE Flex ANG	0.175
x_3	R-HIP Flex ANG	0.161
x_{102}	L-HIP Flex ANG V	0.158
x_{93}	R-KNEE Flex ANG V	0.156
x_{15}	L-HIP Flex ANG	0.150
x_{18}	L-KNEE Flex ANG	0.150
x_{105}	L-KNEE Flex ANG V	0.147
x_{90}	R-HIP Flex ANG V	0.140
x_{21}	L-ANK Flex ANG	0.135
x_{14}	L-HIP Abd ANG	0.122

Tabelle 2: Beste Merkmalsrelevanzen der Einzelmerkmale (links, Wurzelknoten ID3) und durchschnittliche Merkmalsrelevanz unter Berücksichtigung von Redundanzen (rechts, klassenspezifische ID3) - Bezeichnungen: wie Tabelle 1, zusätzlich ANK (Fußgelenk), Abd (Abduktion - Gelenkwinkel in der Frontalebene)

Merkmalsrelevanzen ausgewertet werden. Das können einerseits Entscheidungsbäume mit den nächstbesten Merkmalen im Wurzelknoten oder klassenspezifische Entscheidungsbäume sein, die immer die Unterschiede zwischen einer Ausgangsklasse und deren Negation auswerten. Die so entstehenden Merkmalsrelevanzen sind in der rechten Teiltabelle von Tabelle 2 angegeben.

Sie unterscheiden sich weder signifikant von den Relevanzen der Einzelmerkmale noch von den Ergebnissen der statistischen multivariaten Auswahl mit dem MANOVA-Verfahren. Allerdings geht die Wertigkeit des Merkmals x_{90} im Vergleich zur Einzelrelevanz zurück. Im Unterschied zu MANOVA beziehen sich die Relevanzen nicht auf eine Gruppe von Merkmalen, sondern auf eine durchschnittliche Relevanz des Merkmals unter Berücksichtigung der Auswahlentscheidungen in höheren Hierarchieebenen des Entscheidungsbaums. Deswegen ist auch hier weder ein Zahlenvergleich der Merkmalsrelevanzen zum MANOVA-Verfahren noch zu den Einzelrelevanzen im Wurzelknoten aussagekräftig.

Aus den Entscheidungsbäumen können dann Regeln generiert werden. Jeder Endknoten (Knoten ohne weitere Verzweigung) liefert eine Regel, deren Konklusion die Entscheidung über die Ausgangsgröße im Knoten ist. Die Prämisse folgt aus der UND-Verknüpfung aller spezifizierten Merkmale x_l auf dem Rückweg zum Wurzelknoten. Nachfolgend werden die Regeln noch durch ODER-Verknüpfungen mit weiteren linguistischen Termen spezifizierter Merkmale oder durch Streichen von Merkmalen generalisiert.

Im Beispiel werden acht verschiedene Entscheidungsbäume generiert: Ein Entscheidungsbaum spezifiziert dabei alle sieben Ausgangsklassen gegeneinander und die anderen sieben klassenspezifischen Entscheidungsbäume unterscheiden jeweils eine Ausgangsklasse gegen ihre Negation (ODER-Verknüpfung aller anderer Ausgangsklassen). Aus diesen Entscheidungsbäumen werden 220 Regeln extrahiert, von denen nach dem Generalisieren noch 107 Regeln übrig bleiben. Diese Regeln können einerseits direkt zur Klassifikation der Schrittphasen verwendet werden, indem jeder Regel eine bestimmte Relevanz zur Auflösung von Konflikten konkurrierender Regeln zugewiesen wird. Besser interpretierbar ist eine Variante, die 10 besonders gut kooperierende Regeln auswählt:

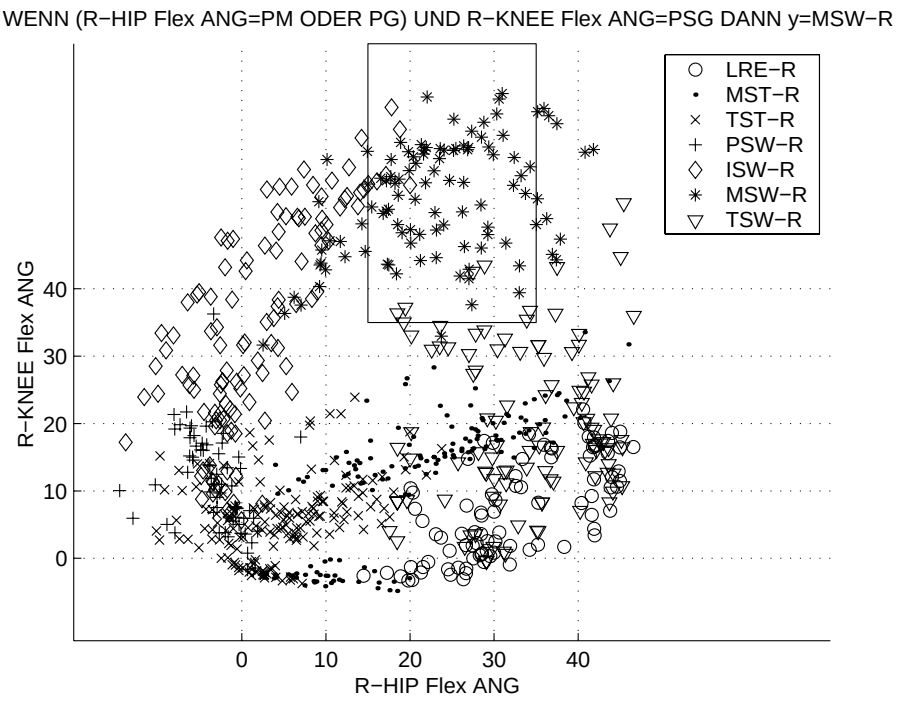


Bild 5: Visualisierung der Fuzzy-Regel R_8 zur Bestimmung der Schrittphase Mid Swing mit den Merkmalen x_3 (R-HIP Flex Ang) und x_6 (R-KNEE Flex Ang)

R_1:	WENN x_{15}=ZE $\cap$ x_{21}=PG $\cap$ (x_{93}=ZE $\cup$ PK $\cup$ PG)	DANN y=LRE-R
R_2:	WENN (x_3=PM $\cup$ PG $\cup$ PSG) $\cap$ x_{90}=ZE $\cap$ (x_{93}=ZE $\cup$ PK $\cup$ PG)	DANN y=LRE-R
R_3:	WENN (x_3=PK $\cup$ PM $\cup$ PG) $\cap$ (x_{14}=NG $\cup$ NK) $\cap$ x_{105}=PK	DANN y=MST-R
R_4:	WENN x_{90}=NEG $\cap$ (x_{102}=PK $\cup$ PG) $\cap$ (x_{105}=ZE $\cup$ PK $\cup$ PG)	DANN y=MST-R
R_5:	WENN (x_{90}=NEG $\cup$ ZE) $\cap$ (x_{105}=NG $\cup$ NK)	DANN y=TST-R
R_6:	WENN (x_{15}=PM $\cup$ PG $\cup$ PSG) $\cap$ (x_{102}=ZE $\cup$ PK) $\cap$ (x_{105}=NG $\cup$ NK $\cup$ ZE)	DANN y=TST-R
R_7:	WENN x_3=ZE $\cap$ (x_{18}=ZE $\cup$ PK $\cup$ PM) $\cap$ (x_{90}=NEG $\cup$ ZE) $\cap$ (x_{105}=ZE $\cup$ PK)	DANN y=PSW-R
R_8:	WENN (x_3=PM $\cup$ PG) $\cap$ x_6=PSG	DANN y=MSW-R
R_9:	WENN (x_{15}=NICHT ZE) $\cap$ (x_{90}=PK $\cup$ PM $\cup$ PG) $\cap$ x_{93}=PK	DANN y=ISW-R
R_{10}:	WENN (x_{90}=NEG $\cup$ ZE $\cup$ PK) $\cap$ x_{93}=NK $\cap$(x_{102}=NG $\cup$ NK $\cup$ ZE)	DANN y=TSW-R

Damit wird jede Schrittphase durch ein bis zwei Regeln linguistisch beschrieben. Auffällig ist dabei, dass jeweils unterschiedliche Merkmale verwendet werden.

Eine der einfacheren Regeln, die Regel R_8 zur Beschreibung der mittleren Schwungphase, ist in Bild 5 dargestellt. Sie baut auf den Merkmalen x_3 (Winkel zwischen rechtem Oberschenkel und Oberkörper) und x_6 (Beugung rechtes Knie) auf. In der mittleren Schwungphase ist dabei das Knie maximal angewinkelt (Term PSG) und der Winkel zwischen rechtem Oberschenkel und Oberkörper ist mittel (PM) bis groß (PG). Die genannte Phase ist in Bild 5 durch * markiert, das Gebiet der aktivierten Regel (Wahrheitswert Prämisse > 0.5) ist eingerahmt. Die Regel erfasst einen Großteil der Messungen, die zur genannten Phase gehören. Die Parameter der Zugehörigkeitsfunktionen sind durch gepunktete Linien dargestellt. Während einige Parameter (10 und 20 für x_3) aus Sicht der Regel gut gewählt sind, lassen andere noch Reserven offen: Eine Verschiebung des Parameters 40 auf 50 für x_3 eröffnet die

Möglichkeit, bisher nicht erfasste Datensätze mit in die Regel einzubeziehen. Eine Verschiebung des Parameters 40 auf ca. 45 für x_6 reduziert hingegen die Fehleranzahl zur Klasse TSW-R. Bei beiden Parametern ist aber zu berücksichtigen, dass sie einen Kompromiss für die Abgrenzung aller möglicher Klassen auf der Grundlage des Merkmals, nicht der Regel darstellen.

Die auftretenden Abweichungen zwischen Lerndatensatz und Klassifikatorergebnissen sind geringfügig besser im Vergleich zum statistischen Klassifikator (Fehler Lerndatensatz 16.3%, Testdatensatz 17.5%) und betreffen bis auf 1.0% Fehler nur Abweichungen zur benachbarten Phase (Bild 6 in Abschnitt 5).

5 Vergleich der methodischen Ansätze

Beide Ansätze, das statistische Verfahren auf der Basis des MANOVA-Verfahrens, der Diskriminanzanalyse und von Bayes-Klassifikatoren sowie das Verfahren zur Generierung von Fuzzy-Systemen, führen im hier genannten Problem auf eine gute Klassifikationsgüte unter Berücksichtigung der in Abschnitt 2 ausgeführten Umstände der Entstehung des Lerndatensatzes. Bild 6 vergleicht die Ergebnisse des Testdatensatzes und beider Verfahren anhand von fünf Schritten mit zwei unterschiedlichen Geschwindigkeiten (links - schnell, rechts -langsam), wobei jeweils noch eine logische Nachkorrektur zur Beseitigung nichtplausibler Übergänge erfolgte. Es fällt auf, dass

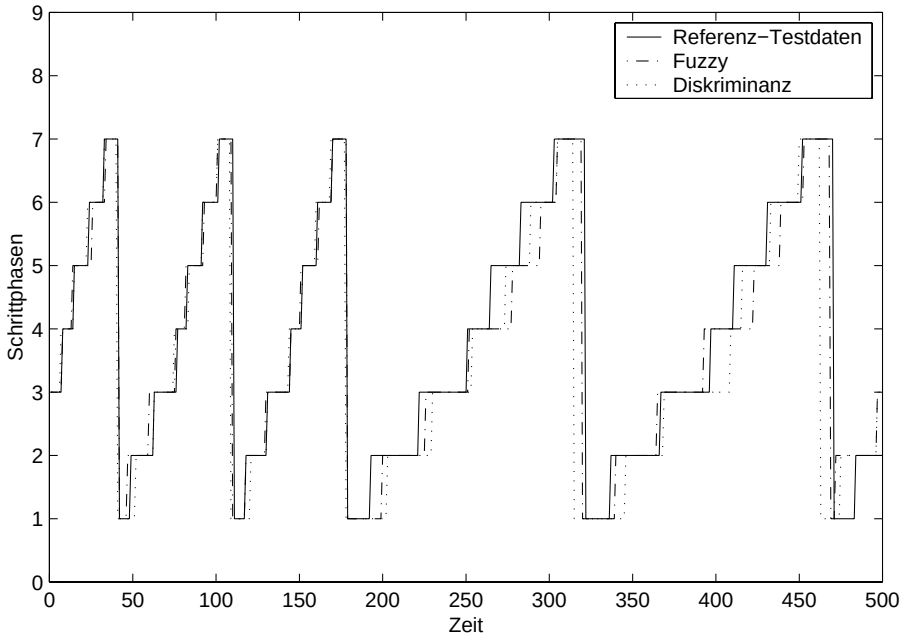


Bild 6: Vergleich zwischen den Schrittphasen im Testdatensatz (Referenzwerte aus Codierung entsprechend Abschnitt 2), den Ergebnissen des Bayes-Klassifikators nach der Diskriminanzanalyse (Abschnitt 3) und dem Fuzzy-Klassifikator (Abschnitt 4)

die Ergebnisse bei der schnelleren Geschwindigkeit weniger voneinander abweichen. Dafür gibt es drei Ursachen, erstens sind die Unterschiede zwischen den einzelnen Schrittphasen bei größeren Geschwindigkeiten stärker ausgeprägt, zweitens sind die für die Klassifikation der Lerndaten verwendeten Prozentangaben auf normale Gehgeschwindigkeiten bezogen und drittens ist die schnellere Geschwindigkeit auch im

Lerndatensatz stärker vertreten und folglich detaillierter angelernt. Dennoch gelingt es beiden Ansätzen, die qualitativen Abläufe richtig zu erfassen.

Der Vorteil der statistischen Verfahren besteht darin, meist eine geschlossene optimale Lösung zu bieten. Das Vorgehen über die Kovarianzmatrizen findet die markantesten Merkmale heraus und vermeidet starke Korrelationen zwischen diesen Merkmalen. Damit berücksichtigt es Abhängigkeiten und Redundanzen von Merkmalen. Nachteilig sind hier allerdings parameterlineare Ansätze, die die Freiheitsgrade einschränken, und die ebenfalls reduzierte Interpretierbarkeit der Ergebnisse.

Beim vorgestellten datenbasierten Entwurf von Fuzzy-Systemen sind die Ergebnisse relativ gut interpretierbar, weil alle Regeln mit einer überschaubaren Anzahl von Merkmalen und linguistischen Termen arbeiten. Die geringe Anzahl der Regeln garantiert auch noch die Überschaubarkeit des Gesamtsystems. Da keine Rücksicht auf den logischen Zusammenhang der Merkmale genommen wird, muss sich der Experte u. U. um die Interpretation von Regeln mit heterogenen Bestandteilen (R_5: linkes Knie und rechte Hüfte) bemühen. Allerdings findet sich für jedes dieser Merkmale auch ein korreliertes Ersatzmerkmal mit logischem Zusammenhang, das ohne wesentliche Verluste zu einer ähnlichen Regelgüte führt. Aufgrund der nichtlinearen Struktur ist der Klassifikator an viele Probleme besser anpassbar. Allerdings entstehen bei Erhalt der ursprünglichen Merkmale nur achsenparallele Klassifikatoren, was u.U. die Klassifikationsgüte begrenzt, und der Entwurfsaufwand steigt an.

Aus der Ähnlichkeit der Ergebnisse bei der Merkmalsauswahl und der Klassifikationsgüte darf aber nicht geschlossen werden, dass beide Verfahren immer zu ähnlichen Ergebnissen führen. Bei anderen Aufgabenstellungen fallen die Vor- oder Nachteile der jeweiligen Verfahren stärker ins Gewicht, weswegen sich deutliche Unterschiede ergeben können. Insbesondere bei der Merkmalsauswahl existieren Fälle, die zu vollkommen verschiedenen Merkmalskombinationen führen, was unterschiedlichen Lösungswegen entspricht.

6 Zusammenfassung

In diesem Beitrag werden zwei Verfahren zur Schrittphasenerkennung in der Instrumentellen Ganganalyse vorgestellt und verglichen. Das erste Verfahren stammt aus der multivariaten Statistik und basiert auf einer Merkmalsauswahl mit dem MANOVA-Verfahren, einer Diskriminanzanalyse und dem Einsatz von Bayes-Klassifikatoren. Das andere Verfahren umfasst den datenbasierten Entwurf eines Fuzzy-Systems inklusive der Merkmalsauswahl, der Generierung von Zugehörigkeitsfunktionen und der Regeln. Beide Verfahren liefern gute Resultate und bieten Zugänge zur Interpretierbarkeit der Lösungen.

Die Autoren danken an dieser Stelle Herrn Dr. Jens Jäkel und Herrn Dr. Lutz Gröll vom Forschungszentrum Karlsruhe, die wesentlichen Anteil an der Erarbeitung der methodischen Grundlagen für dieses Forschungsprojekt haben, und Herrn Dipl.-Ing. Matthias Schablowski und Herrn Dipl.-Inf. Joachim Schweidler von der Orthopädischen Universitätsklinik Heidelberg, die stets als Diskussionspartner zu allen Fragen der Datenerfassung am Laufband zur Verfügung standen.

Literatur

[1] Perry, J.: *Gait Analysis. Normal and Pathological Function*. Thorofare: Slack Inc. 1992.

[2] Rupp, R.; Schablowski, M.; Gerner, H.: Entwicklung und Evaluierung eines Diagnostiklaufbandes zur dreidimensionalen Erfassung der Gangdynamik. *Biomed Tech* (1998), S. 192–93.

[3] Siebel, A.; Berghof, R.; Döderlein, L.: Modification of the Walking Pattern in Patients with Operated Anterior Cruciate Ligament Rupture Measured With and Without a Brace. *Gait and Posture* 3(2) (1995).

[4] Schumann, N.: Grundlagen und Einsatz oberflächenelektromyographischer und biomechanischer Methoden im Rahmen der Diagnostik und Therapieverlaufskontrolle von Schlaganfall-Patienten. In: *Konzepte der Bewegungstherapie nach Schlaganfall* (Seidel, E. e. a., Hg.), S. 97–100. Reihe Praktische Physiotherapie / Sporttherapie, GFBB Bad Kösen. 1995.

[5] Bauer, H.; Schöllhorn, W.: Self-Organizing Maps for the Analysis of Complex Movement Patterns. *Neur. Proc. Let.* 5 (1997), S. 193–199.

[6] Mainka, C.; Boenick, U.: Integrated gait analysis for future routine clinical use. *Biomed Tech* 38 (1993) 12, S. 325–31.

[7] Riener, R.; Fuhr, T.; Schmidt, G.; Quintern, J.: Entwurf von geregelten Neuroprothesen unter Berücksichtigung der intakten Willkürmotorik am Beispiel des Aufstehens. *Automatisierungstechnik* 46 (1998), S. 507–515.

[8] Daunicht, W.; Steiner, R.; Hömberg, V.: A Metaattractor Controller for a Stance-Gait-Neuroprothesis. In: *Proc. EUFIT'98*, S. 1775 – 1779. Aachen. 1998.

[9] Zlatnik, D.: Intelligently Controlled Above Knee (A/K) Prosthesis. In: *Proc. 4th Int. Conf. On Motion and Vibration Control (MOVIC'98)*. Zürich. 1998.

[10] Dorgan, S.; Schmidt, G.: Human Walking as a Model for Bipedal Walking Robot Design. In: *Proc. Euromech 375: Biology and Technology of Walking*, S. 260–267. 1998.

[11] Huang, P.; Harris, C.; Nixon, M. S.: Recognising humans by gait via parametric canonical space. *Artificial Intelligence in Engineering* 13 (1999), S. 359–366.

[12] Ng, S. G.; Chizeck, H. J.: Fuzzy Model Identification for Classification of Gait Events in Paraplegics. *IEEE Transactions on Fuzzy Systems* 5 (4) (1997), S. 536–544.

[13] Barton, G.: Interpretation of Gait Data Using Kohonen Neural Networks. *Gait & Posture* 10 (1999), S. 85–86.

[14] Abel, R.; Rupp, R.; Siebel, A.; Döderlein, L.; Gerner, H. J.; Mikut, R.; Peter, N.; Bretthauer, G.: Classification of Gait Analysis Data Using Fuzzy Logic Based Rule Generation and Multivariate Statistical Analysis. In: *ESMAC'00*. Lund. 2000.

[15] Wernig, A.; Nanassy, A.; Müller, S.: Laufband (Treadmill) Therapy in Incomplete Paraplegia and Tetraplegia. *Journal of Neurotrauma* 16 (1999) 8, S. 719–726.

[16] N.N.: *OrthoTrack 4.1- Gait Analysis Software, Reference Manual*. MotionAnalysis. 1999.

[17] Ahrens, H.; Läuter, J.: *Mehrdimensionale Varianzanalyse: Hypothesenprüfung, Dimensionserniedrigung, Diskrimination bei multivariaten Beobachtungen*. Berlin: Akademie-Verlag. 1974.

[18] Tatsuoka, M. M.: *Multivariate Analysis*. New York: Macmillan. 1988.

[19] Mikut, R.; Jäkel, J.; Gröll, L.: Informationstheoretische Maße zur Merkmalsauswahl, Generierung von Zugehörigkeitsfunktionen und Regeln für Fuzzy-Klassifikatoren. In: *Proc. Computational Intelligence und Industrielle Anwendungen*, VDI-Berichte 1526, S. 125–130. Düsseldorf: VDI-Verlag. 2000.

[20] Mikut, R.; Jäkel, J.; Gröll, L.: Automatic Design of Interpretable Membership Functions. In: *Proc. 8th Fuzzy Colloquium Zittau*, S. 103–111. 2000.

[21] Jäkel, J.; Mikut, R.; Malberg, H.; Bretthauer, G.: Datenbasierte Regelsuche für Fuzzy-Systeme mittels baumorientierter Verfahren. In: *Berichtsband 9. Workshop* Fuzzy Control *d. GMA-FA 5.22*, S. 1–15. Dortmund. 1999.

[22] Mikut, R.; Jäkel, J.; Malberg, H.; Bretthauer, G.: Datenbasierter Entwurf von Fuzzy-Systemen für medizinische Diagnoseaufgaben. *Automatisierungstechnik* 48 (2000) 7, S. 317–326.

[23] Quinlan, J. R.: Induction of Decision Trees. *Machine Learning* 1 (1986), S. 81–106.

Modellierung des Verhaltens Dynamischer Systeme mit erweiterten Fuzzyregeln

Andreas Fick, Hubert B. Keller

Institut für Angewandte Informatik, Forschungszentrum Karlsruhe
Hermann-von-Helmholtz-Platz 1, D-76344 Leopoldshafen
Tel.: 07247/82-5782 Fax: 07247/82-5730 E-Mail: fick@iai.fzk.de

Abstract

Der Artikel beschäftigt sich mit der symbolischen, regelbasierten Modellierung des Verhaltens dynamischer Systeme. Neben der Erzeugung eines solchen Modells aus Prozeßdaten (Zeitreihen) besteht zunächst das Problem, überhaupt eine geeignete Modellstruktur zu definieren. Ein solches Modell sollte sowohl unscharfe Daten verarbeiten können als auch die explizite Repräsentation zeitlicher Beziehungen ermöglichen. Als Lösung wird eine auf dem Fuzzy Control-Ansatz von Mamdani basierende Modellstruktur vorgeschlagen und beschrieben.

1 Einleitung

Steht für ein dynamisches System kein analytisches Modell zur Verfügung, weil eine mathematische Modellierung, beispielsweise aufgrund der hohen Anzahl von Prozeßgrößen, weil nicht alle Größen meßtechnisch erfaßbar sind oder weil viele Systemzusammenhänge nicht bekannt sind, gescheitert ist, so kann der Einsatz datengetriebener Verfahren eine Möglichkeit darstellen, dennoch ein brauchbares Verhaltensmodell zu erhalten [7].

Zur Erzeugung von Verhaltensmodellen aus Prozeßdaten (Zeitreihen) werden beispielsweise künstliche Neuronale Netze erfolgreich eingestzt [12]. Ein solches Modell besitzt jedoch den Nachteil, eine „Black Box“ darzustellen, die vom Menschen nur sehr schwer verstanden werden kann. Insbesondere ist auch die Beurteilung des Modells schwierig, was verständlicherweise auch zu Akzeptanzproblemen beim Anwender führt, der einem Modell vertrauen soll, das keinerlei Begründung für sein Verhalten liefert. Deshalb besteht das Ziel, verständliche, regelbasierte Modelle zu verwenden.

Zur Erzeugung solcher symbolischen Modelle steht prinzipiell eine große Zahl maschineller Lernverfahren zur Verfügung [6], die jedoch nicht direkt verwendet werden können [16].

Die Probleme beginnen bereits damit, daß nicht klar ist, welches die zu betrachtenden Basismerkmale sind, d.h. auf welcher Grundlage überhaupt nach Zusammenhängen gesucht werden soll. Geht man von einzelnen Meßwerten, von Meßintervallen, von ganzen Meßwertreihen oder geeigneten, aggregierten Größen als Attributen aus? Damit besteht die erste Aufgabe eines automatischen Modellierungssystems in der Erzeugung einer geeigneten symbolischen Repräsentationsbasis meßbarer Größen (Definition der Merkmale), die dann die Grundlage für die gewünschte Analyse und Regelgenerierung darstellt. Dieser Übergangsschritt von der numerischen zur symbolischen Form stellt einen wesentlichen Unterschied z.B. gegenüber der Suche nach implizitem Wissen in Datenbanken (data mining, knowledge discovery in databases) dar.

Des weiteren ist bei der Modellierung eines dynamischen Systems die Zeit als wichtiger Parameter zu berücksichtigen. Dies bedeutet, daß Beziehungen zwischen verschiedenen

Systemgrößen und damit entsprechend auch zwischen den daraus abgeleiteten Merkmalen nicht statisch sind, sondern verschiedenartige, zustandsabhängige Beeinflussungen mit gewissen Zeitverzögerungen anzunehmen sind. Deshalb müssen Lernbeispiele neben der aktuellen Situation auch alle relevanten Situationen der Vergangenheit enthalten, was aufgrund der Größe des zu durchsuchenden Hypothesenraums (kombinatorische Explosion) zwangsläufig zum Scheitern direkt angewandter etablierter maschinellen Lernverfahren wie z.B. ID3/CN2/C4.5 führt.

Als weiterer Punkt soll die Forderung nach dem angemessenen Umgang mit unpräziser, unscharfer Information genannt werden. Dies betrifft insbesondere auch die Ausgabe des Modells, bei der es nicht genügt, einen bestimmten Zustand aus einer Menge vorgegebener Prozeßzustände im Sinne einer Klassifikation vorherzusagen, sondern bei der die Werte möglichst aller Größen zu prognostizieren sind.

2 Bezeichnungen

Gegeben seien die Systemgrößen X_i. Wird eine eindeutige oder beliebige Systemgröße betrachtet, so wird im folgenden der Index weggelassen. Zur einfacheren Unterscheidung von den Größen der Prämisse werden in den Konklusionen von Fuzzyregeln auftretende Größen mit Y beziehungsweise Y_i bezeichnet.

$x_i(t)$ bezeichnet den Wert der Systemgröße X_i zum Zeitpunkt t. Schätzwerte werden durch ein „ˆ“ gekennzeichnet, z.B. $\hat{x}_i(t)$.

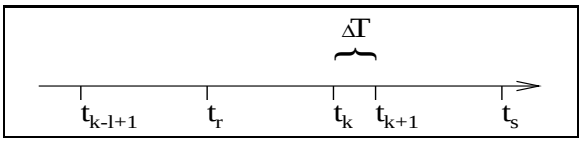


Abbildung 1: Bezeichnungen auf der Zeitachse.

Die Folge $\{t_n\}$ äquidistanter, diskreter Zeitpunkte mit dem Abstand ΔT, d.h. $t_{n+1} = t_n + \Delta T$ bezeichnet die Folge der Zeitpunkte, zu denen die Werte des gegebenen Systems betrachtet werden, bei Meßgrößen also die Abtastzeitpunkte.

A und B sind Fuzzymengen mit Zugehörigkeitsfunktion $\mu_A(x)$ bzw. $\mu_B(x)$. T bezeichnet eine Fuzzymenge für eine (relative) Zeitangabe und wird analog definiert.

$R = R_1, \ldots, R_N$ bezeichnet die Menge der Regeln eines Fuzzyreglers, die sich im folgenden zur Vereinfachung der Schreibweise implizit alle auf dieselbe linguistische Variable bzw. Größe Y beziehen sollen.

Im Zusammenhang mit Fuzzyregeln wird im folgenden der Zeitpunkt des Zutreffens der Prämisse, bzw. des betrachteten Fuzzy-Terms der Prämisse, mit t_r und der Zeitpunkt des Zutreffens der Konklusion mit t_s bezeichnet.

Die zeitliche Differenz zwischen beiden Zeitpunkten ist $\Delta t := t_s - t_r = t_d$ bzw. $\Delta t = t_d$ mit $d = s - r$ bzw. $\Delta t = d * \Delta T$.

Zur Erläuterung konkreter Berechnungen im Zusammenhang mit Prognosen ist wichtig, welche Daten zum Zeitpunkt der Prognoseerstellung bekannt sind. Hierzu werden im folgenden die bekannten Zeitpunkte, d.h. das (endliche) „Gedächtnis“ der Länge t_l, mit

$t_{k-l+1} \ldots t_k$ bezeichnet, die betrachtete „Zukunft“ wird durch die Zeitpunkte $t_{k+1} \ldots t_{k+m}$ repräsentiert.

3 Ausgangspunkt

Die in der Einleitung angeführten Forderungen an die Modellstruktur lassen sich mit den Stichworten „regelbasiertes Modell“, „Verarbeitung unscharfer Daten“ und „Behandlung von Zeitabhängigkeiten“ zusammenfassen. Ausgangspunkt der folgenden Überlegungen ist daher eine Fuzzy-Regelbasis nach Mamdani [10], die zwar eigentlich zum Zweck der Regelung (Fuzzy Control) konzipiert worden ist, sich aber aufgrund der universellen Struktur auch zur Modellierung des Verhaltens dynamischer Systeme anbietet. Sie basiert auf Regeln der Art

Wenn [x ist A] Dann [y ist B].

Zu einer Menge solcher Regeln ist zum einen die Prämissenauswertung nach den Regeln der Fuzzylogik [17] mit den bekannten Wahlmöglichkeiten hinsichtlich der logischen Verknüpfungen (t-Normen und t-Conormen) und zum anderen das Inferenzprinzip definiert. Bei diesem wird im wesentlichen die Prämisse jeder einzelnen Regel auf einen Wert reduziert und anschließend in verschiedenen Ausprägungen, z.B. durch Minimum- oder Produktbildung, mit der jeweiligen unscharfen Konklusionsmenge verknüpft. Anschließend folgen „Komposition“ zur Zusammenfassung der zu einer Größe gehörenden Konklusionsmengen und „Defuzzifizierung“ zur Reduktion der Ergebnismenge auf einen scharfen Wert.

Der im folgenden beschriebene Ansatz verfolgt das Ziel, einen möglichst großen Teil dieser etablierten Methode beizubehalten. Dadurch können ihre Vorteile wie Bekanntheit, Transparenz oder Effizienz genutzt werden, und es kann allgemein auf den damit gemachten weitgehend positiven Erfahrungen und den erzielten Forschungsergebnisse aufgebaut werden. Außerdem ist so eine einfache Realisierung (Implementierung) auf Basis existierender Software-Tools, wie z.B. Matlab, möglich.

4 Beobachtung

Im Gegensatz zu Differentialgleichungsmodellen, in die zeitliche Bezüge direkt eingehen, beschreibt eine Regel einer Fuzzyregelbasis im wesentlichen einen statischen Zusammenhang zwischen den Fuzzymengen A und B. Dynamische Aspekte werden in Regelbasen nach Mamdani lediglich implizit über den Auswertezeitpunkt der Prämissen und den „Takt“ des Reglers, der zyklisch in bestimmten Abständen Werte erfaßt, auswertet und anschließend Stellgrößen ausgibt, berücksichtigt. Desweiteren können sie bei Regelbasen nach Sugeno-Takagi [15], die als Abwandlung des Mamdani-Ansatzes angesehen werden können, prinzipiell in die Funktionsausdrücke der Konklusionen eingehen.

Trotz der Repräsentation ohne Zeitbezug muß bei der Verwendung einer Regel im Rahmen von Fuzzy Control natürlich eine zeitliche Zuordnung zwischen Prämisse bzw. Konklusion der Regel und Werten der Systemgrößen X bzw. Y vorgenommen werden, und

der durch obige Regel festgelegte Zusammenhang läßt sich genauer beschreiben als Beziehung zwischen $\mu_A(x(t_r))$ und $\mu_B(y(t_s))$ mit dem „Meßzeitpunkt“ t_r und dem „Stellzeitpunkt“ t_s.

5 Regeln mit expliziter Zeitdarstellung

Eine explizite Angabe der Zeitpunkte innerhalb der Regel würde den zeitlichen Zusammenhang zwischen Prämisse und Konklusion offensichtlich machen. Dies entspräche einer am menschlichen Denken in Ursache-Wirkungsbeziehungen orientierten Beschreibung, die die Verstehbarkeit solcher Regelbasen für den Menschen erleichtern würde (vgl. [3] und [14] S.77ff). Außerdem würde sie eine Behandlung unterschiedlicher Verzögerungszeiten auf der Ebene des Fuzzy Control ermöglichen. Unterschiedliche Totzeiten im betrachteten dynamischen System könnten in der Regelbasis explizit angegeben werden und müßten nicht in unterlagerte Schichten, z.B. in die Definition der Meßgrößen, „eingebaut“ werden.

Natürlich wäre eine absolute Angabe der Zeitpunkte, im obigen Beispiel also von t_r und t_s, nicht sinnvoll. Vielmehr ist nur die Verzögerungszeit (Totzeit) Δt zwischen Ursache und Wirkung als zeitliche Differenz von t_s und t_r wesentlich.

Desweiteren sollte die Angabe ungefährer Zeitverzögerungen durch die Verwendung unscharfer Mengen für relative Zeitangaben möglich sein. Der Grund hierfür liegt zum einen im Wunsch nach Verständlichkeit des resultierenden Modells, die durch Verwendung linguistischer Werte für Zeitangaben erreicht werden soll. Zum anderen erlaubt die Zusammenfassung (Clusterung) zeitlich ähnlicher, nicht aber unbedingt identischer Beziehungen im Rahmen der maschinellen Modellerstellung aus Zeitreihen lediglich ungefähre Zeitangaben (vgl. [16]). Auch das Einbringen unscharfen (Experten-) Wissens wird so ermöglicht.

„Scharfe“ Zeitangaben ergeben sich als Sonderfälle durch die Wahl geeigneter Zugehörigkeitsfunktionen (Singletons).

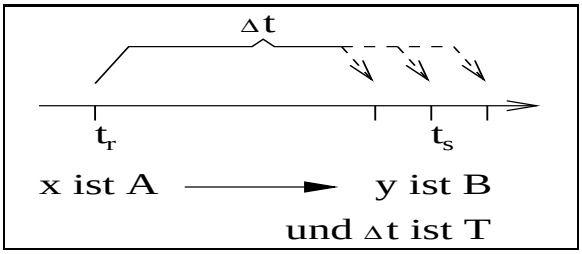


Abbildung 2: Regel Typ 1.

Unter diesen Voraussetzungen und Beibehaltung der Struktur des Fuzzy Control-Ansatzes nach Mamdani bieten sich zwei äquivalente Darstellungsformen an. Zum einen

$$\text{Wenn } [x \text{ ist } A] \text{ Dann } [y \text{ ist } B \text{ und } \Delta t \text{ ist } T]. \tag{1}$$

mit Δt als der Zeitverzögerung, nach der „y ist B“ gelten soll [1], angegeben relativ zum implizit angenommenen Bezugszeitpunkt t_r der Auswertung von „x ist A“.

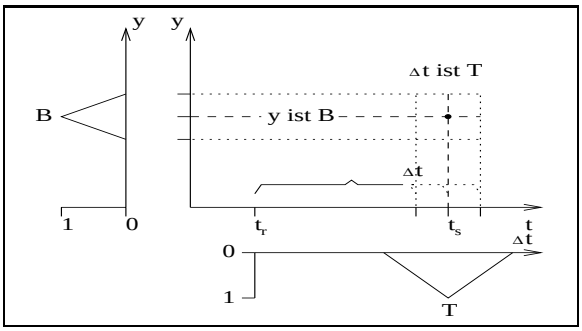


Abbildung 3: [y ist B und ΔT ist T].

Hier wurde eine „Zeitverzögerung von T“ in Fuzzy-Schreibweise als „Δt ist T“ übersetzt und zur Verdeutlichung der zeitlichen Zuordnung (vgl. Abbildung 3) „und“ geschrieben. Dieses „und“, das so etwas wie „nach“ oder „zum (relativen) Zeitpunkt“ ausdrückt, *kann* als übliche Fuzzy-Und-Verknüpfung (t-Norm) umgesetzt werden (siehe Abschnitt 6), und die Entscheidung für dieselbe t-Norm wie bei der Konjunktion kann gegebenenfalls zu einem besonderes effizienten Verarbeitungsschema führen. Im Prinzip ist die Wahl der entsprechenden zweidimensionalen Fuzzyrelation jedoch unabhängig von den sonst gewählten Verknüpfungsoperatoren wie z.B. Minimumbildung oder Multiplikation. Selbstverständlich sollten dabei sinnvolle Nebenbedingungen, beispielsweise Monotonie in den Zugehörigkeitsgeraden, berücksichtigt werden.

Im folgenden wird ein mit einer unscharfen (relativen) Zeitangabe T versehener Fuzzy-Term, d.h. beispielsweise die zweidimensionale Fuzzymenge (Fuzzyrelation) $B\tilde{\wedge}T$ als „zeitbehafteter Fuzzy-Term“, kurz „**t-Fuzzy-Term**“, bezeichnet.

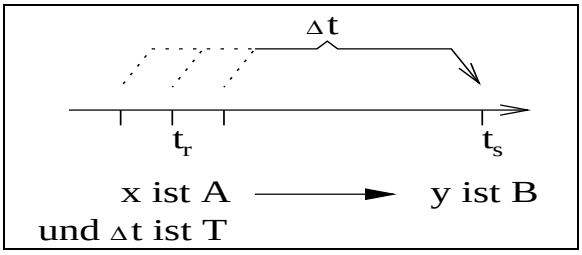


Abbildung 4: Regel Typ 2.

Die zweite Möglichkeit besteht in der Angabe der Zeitverzögerung von „x ist A“ relativ zum Zeitpunkt t_s von „y ist B“ (Abbildung 4):

$$\text{Wenn } [x \text{ ist } A \text{ und } \Delta t \text{ ist } T] \text{ Dann } [y \text{ ist } B]. \tag{2}$$

Da im vorliegenden Fall keine allgemeine logische, sondern eine kausale Beziehung, bei der „x ist A“ immer vor „y ist B“ liegt, angegeben wird, genügt es, sich auf positive

[1] Die Zusammengehörigkeit von B und ΔT ist durch „[...]“ angedeutet. Auf eine explizite Zuordnung, z.B. mittels Indizes wurde aus Gründen der Übersichtlichkeit verzichtet.

Zeitverzögerungen (Beträge) zu beschränken, und für beide Darstellungsformen können dieselben Fuzzymengen verwendet werden.

Zu beiden Darstellungsmöglichkeiten ist zu bemerken, daß bei diesen einfachen Regeln eindeutig erkennbar ist, auf welche Zeitpunkte sich die Zeitverzögerung bezieht. Bei Regeln mit mehreren Fuzzy-Termen in der Prämisse und einem Term in der Konklusion ist die jeweilige Zeitverzögerung für jeden einzelnen Fuzzy-Term der Prämisse anzugeben, falls nicht (in Ausnahmefällen) alle Fuzzy-Terme zum selben (relativen) Zeitpunkt ausgewertet werden sollen, etwa

$$\text{Wenn } [x_1 \text{ ist } A_1 \text{ und } \Delta t_1 \text{ ist } T_1] \text{ Und } [x_2 \text{ ist } A_2 \text{ und } \Delta t_2 \text{ ist } T_2] \text{ Dann } [y \text{ ist } B]. \qquad (3)$$

Im folgenden wird ausschließlich der Ansatz nach Regel 2 weiterverfolgt, da dieser, wie gerade angedeutet, eine naheliegende Erweiterung auf Prämissen mit mehreren Bedingungen und damit auch mehreren Zeitverzögerungen, die sich auf dieselbe Konklusionsgröße auswirken, ermöglicht, während z.B. mehrere verknüpfte Fuzzy-Terme in der Konklusion nicht üblich sind und eine Vermischung beider Ansätze unübersichtlich würde.

An dieser Stelle soll darauf hingewiesen werden, daß beide Arten von Regeln entgegen der intuitiven Vorstellung nur bei geeigneter Wahl der Fuzzy-Operatoren und des Inferenzschemas logisch äquivalent sind (siehe Abschnitt 7).

6 Semantik und Inferenzschema

Im folgenden wird die Angabe der Gleichzeitigkeit als Konjunktion „$\tilde{\wedge}$“ interpretiert (siehe Erläuterungen oben) und als Verknüpfungsoperation das Minimum gewählt. Weiter wird von Singleton-Fuzzifizierung ausgegangen. Als Aggregationsoperator für die „normale“ Fuzzy-Konjunktion wird die Multiplikation verwendet, für die Inferenz ebenfalls. Auch hier ist jeweils die Auswahl unter den üblichen Methoden möglich.

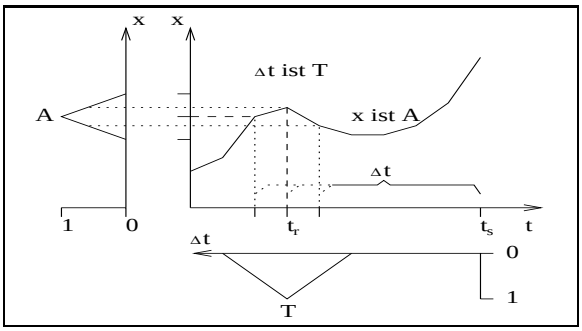


Abbildung 5: [x ist A und ΔT ist T].

Damit führt die Analyse der linken Seite einer Regel nach Art von 2 im Rahmen der Fuzzylogik zu

$$\mu_{A\tilde{\wedge}T}(x, \Delta t) = \min\{\mu_A(x), \mu_T(\Delta t)\}.$$

Die Konjunktionsrelation „$A\tilde{\wedge}T$“ beschreibt eine zweidimensionale Fuzzymenge, was bedeutet, daß sie nicht nur eine Aussage für einen bestimmten Zeitpunkt, sondern für eine Zeitspanne macht. Zur Beibehaltung des prinzipiellen Vorgehens nach Mamdani ist diese Menge für die Inferenz auf einen Wert zu reduzieren.

Am einfachsten ist es, hierzu als Ergebnis der Auswertung des zeitbehafteten Fuzzy-Terms den Wert mit der maximalen Zugehörigkeit zu verwenden, d.h. nach dem besten Kompromiß aus x und Δt zu suchen. Diese Definition des Zugehörigkeitsgrades basiert auf einer Interpretation eines t-Fuzzy-Terms dahingegend, daß er dann als gut erfüllt gilt, wenn er dieses zu irgendeinem Zeitpunkt ist. In Abbildung 5 würde also z.B. zwischen der Auswertung des t-Fuzzy-Terms $A\tilde{\wedge}T$ zum Zeitpunkt t_{r-1} mit $\mu_A(x(t_{r-1})) = 1$, aber nicht passender Zeit, sowie zum perfekt passenden Zeitpunkt t_r ($\mu_T t_r = 1$), aber zu großen Wert von $x(t_r)$, abgewogen werden. Als Ergebnis zum Bezugszeitpunkt t_s ergibt sich

$$\mu_P(t_s) = \max_{\Delta t}\{\mu_{A\tilde{\wedge}T}(x(t_s - \Delta t), \Delta t)\}. \tag{4}$$

Man erhält als Ergebnis die Zugehörigkeitsfunktion der Konklusion zum Zeitpunkt t_s

$$\mu_C(t_s, y) := \mu_{A\tilde{\wedge}T \to B}(y) = \mu_P(t_s) * \mu_B(y). \tag{5}$$

Bei mehreren t-Fuzzy-Termen in der Prämisse ergibt sich beispielsweise (vergleiche Regel 3):

$$\mu_{(A_1\tilde{\wedge}T_1, A_2\tilde{\wedge}T_2)\to B}(y) = (\mu_{P_1}(t_s) * \mu_{P_2}(t_s)) * \mu_B(y). \tag{6}$$

7 Alternativen

Zur expliziten Repräsentation zeitlicher Zusammenhänge ist eine bisher nicht überschaubare Menge verschiedener Ansätze denkbar. Als Ergänzung zum oben erläuterten sollen kurz zwei weitere Möglichkeiten erwähnt werden.

7.1 Zeitangaben in der Konklusion

Geht man von Regeln der Art (1) aus, so erhält man zunächst als zweidimensionale Zugehörigkeitsfunktion der Konklusion

$$\mu_{B\tilde{\wedge}T}(y, \Delta t) = \min\{\mu_B(y), \mu_T(\Delta t)\}.$$

Mit Produktinferenz ergibt sich

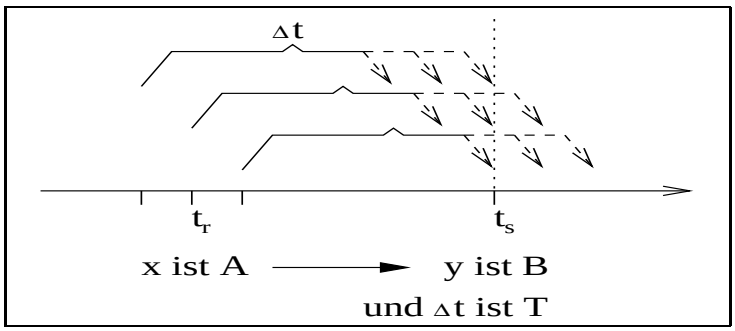


Abbildung 6: Alternative auf Basis von Regel Typ 2.

$$\mu_{A\to(B\tilde{\wedge}T)}(y,\Delta t) = \mu_A(x(t_r)) * \mu_{B\tilde{\wedge}T}(y,\Delta t).$$

Hierdurch werden ausgehend vom Wert von x zum Bezugszeitpunkt t_r Aussagen über Zugehörigkeitsfunktionen der Konklusionsmengen für verschiedene Zeitpunkte $t_s = t_r + \Delta t$ gemacht. Möchte man alle Aussagen, die einen bestimmten Zeitpunkt t_s betreffen, zu einer Fuzzymenge zusammenfassen (vgl. Abbildung 6), um später sukzessive für jeden Zeitpunkt t_s die Konklusionsmengen verschiedener Regeln mittels Komposition auf übliche Weise weiterverarbeiten zu können, so erhält man, falls man analog zur Herleitung von Gleichung 4 vorgeht,

$$\mu_{C'}(t_s,y) := \max_{\Delta t}\{\mu_A(x(t_s-\Delta t)) * \mu_{B\tilde{\wedge}T}(y,\Delta t)\}. \tag{7}$$

Man erkennt in der ausgeschriebenen Version, daß sich dieses Ergebnis im allgemeinen von Gleichung 5 unterscheidet:

$$\begin{aligned}
\mu_{C'}(t_s,y) &= \max_{\Delta t}\{\mu_A(x(t_s-\Delta t)) * \mu_{B\tilde{\wedge}T}(y,\Delta t)\} \\
&= \max_{\Delta t}\{\mu_A(x(t_s-\Delta t)) * \min\{\mu_B(y),\mu_T(\Delta t)\}\} \\
&= \max_{\Delta t}\{\min\{\mu_A(x(t_s-\Delta t)) * \mu_B(y), \mu_A(x(t_s-\Delta t)) * \mu_T(\Delta t)\}\} \\
&= \max_{\Delta t}\{\min\{\mu_A(x(t_s-\Delta t)), \mu_A(x(t_s-\Delta t)) * \frac{\mu_T(\Delta t)}{\mu_B(y)}\} * \mu_B(y)\} \\
&\overset{i.a.}{\neq} \max_{\Delta t}\{\min\{\mu_A(x(t_s-\Delta t)), \mu_T(\Delta t)\} * \mu_B(y)\} \\
&= \max_{\Delta t}\{\min\{\mu_A(x(t_s-\Delta t)), \mu_T(\Delta t)\}\} * \mu_B(y) \\
&= \max_{\Delta t}\{\mu_{A\tilde{\wedge}T}(x(t_s-\Delta t),\Delta t)\} * \mu_B(y) \\
&= \mu_P(t_s) * \mu_B(y) \\
&= \mu_C(t_s,y)
\end{aligned}$$

Die Gleichungen 5 und 7 wären beispielsweise dann äquivalent, wenn man statt des Minimums für „$\tilde{\wedge}$“ das Produkt wählen würde. An dieser Stelle sollen nicht Vor- und Nachteile unterschiedlicher Operatoren untersucht werden, sondern es soll herausgestellt werden,

daß qualitativ gleichbedeutende (Fuzzy-)Regeln auf verschiedene Arten interpretiert werden können, so daß sich eine im Detail unterschiedliche Semantik ergeben kann.

7.2 Regeln als zusammenfassende Regeln

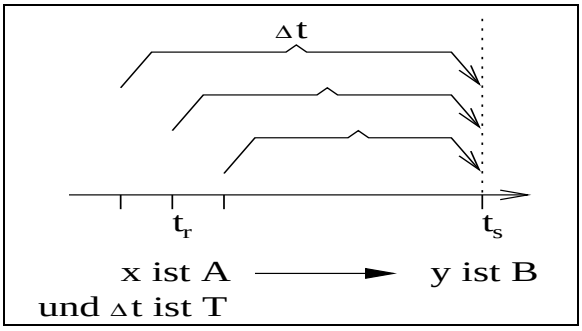


Abbildung 7: Regeln als zusammenfasende Regeln.

Man kann eine Regel nach (2) auch dahingehend interpretieren, daß es sich um die Kurzschreibweise für eine Menge von Regeln mit jeweils präzisen Zeitangaben handelt, wobei jede Regel gemäß dem Zutreffen der Zeitangabe gewichtet würde (siehe Abbildung 7). Eine Regel der Art von (2) entspräche z.B. den „normalen“ gewichteten Fuzzyregeln

Wenn $[x(t_r - \Delta T)$ ist $A]$ Dann $[y$ ist $B]$ Mit Gewicht $\mu_T(\Delta T)$.
Wenn $[x(t_r - 2 * \Delta T)$ ist $A]$ Dann $[y$ ist $B]$ Mit Gewicht $\mu_T(2 * \Delta T)$.
Wenn $[x(t_r - 3 * \Delta T)$ ist $A]$ Dann $[y$ ist $B]$ Mit Gewicht $\mu_T(3 * \Delta T)$.
...
Wenn $[x(t_r - k * \Delta T)$ ist $A]$ Dann $[y$ ist $B]$ Mit Gewicht $\mu_T(k * \Delta T)$.

Für jede dieser Regeln ergäbe sich eine normale Konklusionsberechnung, und die Konklusionen aller dieser Regeln könnte man beispielsweise wie bei „normalen“ Einzelregeln mittels Komposition zusammenfügen.

7.3 Ergänzung

Für jeden der beiden skizzierten Alternativansätze ergeben sich durch die Wahlmöglichkeiten bei den Fuzzy-Operatoren (Mengenoperatoren bzw. t-Normen, Inferenzprinzip, Defuzzifizierungsstrategie, Gewichtung usw.) wiederum verschiedene Möglichkeiten. Je weniger man sich an vorhandenen Ansätzen (z.B. Mamdani) orientiert, desto mehr Freiheitsgrade kommen hinzu.

8 Komposition, Defuzzifizierung und Ergänzungen

Legt man Regeln der Struktur (2) zugrunde und berechnet das Ergebnis jeder Regel nach Gleichung 5, so ändert sich nach der zeitbezogenen Auswertung der Prämissenterme an

Komposition und Defuzzifizierung durch die explizite Repräsentation der Zeitbezüge gegenüber dem Konzept von Mamdani nichts.

Als Komposition der Ergebnisse der Regeln $R_1 \ldots R_N$ ergibt sich für jeden Zeitpunkt t_s

$$\mu_K(t_s,y) = \max\{\mu_{C_{R_1}}(t_s,y) \ldots \mu_{C_{R_N}}(t_s,y)\}. \tag{8}$$

Die Defuzzifizierung geschieht anhand von $\mu_C(t_s,y)$ für den gewählten Zeitpunkt t_s. Im folgenden steht $\hat{y}(t)$ für den Schätzwert der Größe Y zur Zeit t, der aus $\mu_K(t_s,y)$ durch Defuzzifizierung berechnet wird. Entsprechend wird im folgenden die Schreibweise $\hat{x}(t)$ für einen Schätzwert von $x(t)$ verwendet.

8.1 Alternativansätze

Interessant im Zusammenhang mit dem ersten der vorgestellten Alternativansätze ist die Möglichkeit, auf Basis von Gleichung 7 nicht die Fuzzymengen zu einem bestimmten Zeitpunkt t_s für Komposition und anschließende Defuzzifizierung nach y zu verwenden, sondern direkt mit der durch die Gleichung definierten zweidimensionalen Fuzzymenge zu arbeiten.

Nach einer zweidimensionalen Überlagerung (Komposition) könnte man z.B. einen Wert vorgeben und nach der Zeit defuzzifizieren, was man als Schätzung des Auftrittszeitpunktes eines erwarteten Wertes deuten könnte. Beispielsweise könnte so der Augenblick des Überschreitens eines Schwellwertes bestimmt werden.

Bei Definition eines geeigneten zweidimensionalen Defuzzifizierungsverfahrens, das sowohl einen Zeitpunkt als auch einen Wert liefern würde, wäre es auch denkbar, einen allgemein möglichst verläßlichen Schätzwert zu erhalten. D.h. zu dem gefundenen Zeitpunkt wäre die dafür abgegebene Prognose wahrscheinlich ziemlich gut.

9 Wohldefiniertheit (Schätzung unbekannter Werte)

Die Verwendung einer Regelbasis mit Regeln der oben angegebenen Struktur zur Berechnung der Größe Y zu einem Zeitpunkt t_s ist, sofern die Werte aller Größen bis t_s bekannt sind, abgesehen von Effizienzgesichtspunkten (siehe unten) unproblematisch, da der Definitionsbereich von T auf den positiven Bereich beschränkt und damit die Prämisse $\mu_P(t_s)$ in Gleichung 5 jeweils für jede Regel wohldefiniert ist, so daß sich $\mu_K(t_s,y)$ berechnen läßt.

Anders sieht es aus, wenn die Regelbasis z.B. als Prozeßmodell (Verhaltensmodell) verwendet werden soll, um zu einem gegebenen Zeitpunkt t_r Voraussagen (Prognosen) über den Wert bestimmter Größen zu einem bestimmten Zeitpunkt t_s in der Zukunft zu treffen.

Problematisch ist in der Definition der Prämissenzugehörigkeit in Gleichung 4 der Term

$$\mu_{A\hat{\wedge}T}(x(t_s-\Delta t),\Delta t) = \min\{\mu_A(x(t_s-\Delta t)),\, \mu_T(\Delta t)\},$$

falls $x(t_s - \Delta t)$ nicht bekannt ist. Der Term kann aber geschätzt werden, beispielsweise mit folgender Definition:

$$\hat{\mu}_{A\tilde{\wedge}T}(x(t_s-\Delta t),\Delta t)) := \begin{cases} \min\{\mu_A(x(t_s-\Delta t)),\, \mu_T(\Delta t)\}, & (a) \\ \quad \text{falls } x(t_s-\Delta t_i) \text{ bekannt}, & \\ 0, & (b) \\ \quad \text{falls } \mu_T(\Delta t) = 0, & \\ \min\{\mu_A(\hat{x}(t_s-\Delta t)), \mu_T(\Delta t)\}, & (c) \\ \quad \text{falls } \hat{x}(t_s-\Delta t) \text{ bekannt}, & \\ \text{undefiniert}, & (d) \\ \quad \text{sonst}. & \end{cases} \tag{9}$$

(b) ist sinnvoll, da unter der Voraussetzung $\mu_T(\Delta t) = 0$ gilt $min\{\mu_A(x), \mu_T(\Delta t)\} = 0$ unabhängig von x.
(c) berücksichtigt die beste verfügbare Schätzung von x und ist deshalb sinnvoll, obwohl so nicht unbedingt die beste Schätzung für $\mu_P(t_s)$ bzw. $\mu_C(t_s, y)$ sichergestellt werden kann. Dies liegt daran, daß nur ein Schätzwert berücksichtigt wird und außerdem keinerlei Information über die Güte dieser Schätzung eingeht.
(d) bedeutet, daß keine Schätzung möglich ist. Auf diesen Fall wird im nachfolgenden Abschnitt 10 näher eingegangen.

Gegebenenfalls ist auch folgender einfacher Ansatz sinnvoll, der nicht auflösbare Beziehungen in Gleichung 4 einfach ausblendet, indem er nur bekannte Werte verwendet, so daß Regeln mit nicht definierten Prämissen gar nicht berücksichtigt werden:

$$\hat{\mu}_{A\tilde{\wedge}T}(x(t_s-\Delta t),\Delta t)) := \begin{cases} \min\{\mu_A(x(t_s-\Delta t)),\, \mu_T(\Delta t)\}, \text{ falls } x(t_s-\Delta t_i) \text{ bekannt}, \\ 0, \text{ sonst}. \end{cases}$$

Auch der Rückgriff auf Defaultwerte, z.B. den beobachteten Mittelwert einer Größe wäre denkbar.

Entsprechende Definitionen sind bei der Verwendung anderer Operatoren als der Konjunktion für $\tilde{\wedge}$ ebenfalls möglich.

10 Umsetzung unter Effizienzgesichtspunkten

Die folgenden Betrachtungen gehen davon aus, daß ausgehend von einem Zeitpunkt t_k innerhalb eines vorgegebenen Zeithorizontes $t_k \ldots t_{k+m}$ eine Prognose für eine gegebene Größe Y zu erstellen ist. Als Information stehen die zwischengespeicherten Werte der Größen aus dem Bereich von t_{k-l+1} bis t_k, dem „Gedächtnis“ des Systems, zur Verfügung. Die Prämissenberechnung soll auf Basis von Definition 9 erfolgen.

Zunächst ist festzustellen, daß effektive (nicht nur effiziente!) Berechnungen überhaupt nur dann möglich sind, wenn der bei jeder Prämisse bzw. Konklusion betrachtete Zeitbereich endlich ist, d.h. wenn $\mu_T(t)$ nur zu endlich vielen Zeitpunkten von Null abweicht. Eine Berechnung von Konklusion bzw. Prämisse nach Gleichung 5 bzw. 4 kann sich dann auf diesen endlichen Zeitbereich beschränken.

Außerdem müssen alle direkt oder indirekt zur Berechnung oder Schätzung von Werten notwendigen Informationen innerhalb des „Gedächtnisses“ des Systems vorhanden sein. Das bedeutet, daß für jeden in einer Regel auftretenden zeitbehafteten Fuzzy-Term gelten muß, daß für jedes Δt entweder $\mu_T(\Delta t) = 0$ oder $t_{k-l+1} \leq t_k - \Delta t \leq t_k$.

Desweiteren ist offensichtlich, daß sich eine „direkte“ Umsetzung mittels eines rekursiven Ansatzes, der gemäß Gleichung 9 gegebenenfalls auf Schätzwerte $\hat{x}(t_s - \Delta t)$ zurückgreift und diese bei Bedarf berechnet, so daß Fall (d) in Definition 9 nicht auftritt, aus Gründen des Berechnungsaufwandes verbietet. Ein Lösungsansatz besteht in der Vermeidung von Mehrfachberechnungen von Schätzwerten, beispielsweise indem diese Werte in der richtigen zeitlichen Reihenfolge berechnet und im Sinne des dynamischen Programmierens (zwischen-) gespeichert werden, so daß bei Bedarf auf sie zurückgegriffen werden kann.

Ein weitergehender Weg zur Steigerung der Effizienz könnte in einer prinzipiell inkrementellen Berechnung von auf Schätzungen basierenden Werten bestehen. Im Vordergrund stände die Vermeidung einer kompletten Neuberechnung von $\mu_P(t_s)$ bzw. $\mu_C(t_s, y)$ nach Gleichung 4 bzw. 5 bei Eintreffen neuer Informationen, d.h. nach jedem Takt, durch Ausnutzung der Eigenschaften des Maximum-Operators. Allerdings ist anzumerken, daß eine solche effiziente inkrementelle Berechnung je nach Wahl der Operatoren nicht einfach ist und entscheidend von den spezifischen Eigenschaften der gewählten Operatoren abhängt.

11 Einordnung

Es existiert eine Reihe von Arbeiten zur Erzeugung von Fuzzyregelbasen aus Daten und zu ihrer Anwendung zur Vorhersage von Größen dynamischer Systeme (z.B. [1], [11] oder [9]). Jedoch verwendet keiner der bekannten Ansätze (vgl. z.B. [18]) eine mit dem vorliegenden Ansatz vergleichbare explizite Repräsentation zeitlicher Zusammenhänge.

Einen interessanten Ansatz zur Rückkopplung der Ausgänge einer Fuzzyregelbasis *ohne* Defuzzifizierung findet man in [13]. Durch den Verzicht auf eine Defuzzifizierung kann man gegenüber obigem Ansatz prinzipiell den bei der Reduktion der Fuzzymengen auf Schätzwerte auftretenden Informationsverlust vermeiden, jedoch ergeben sich daraus sehr harte Anforderungen sowohl an das Inferenzverfahren als auch an die verwendeten Fuzzymengen.

Der Ansatz sieht keine explizite Repräsentation zeitlicher Bezüge, insbesondere von Verzögerungszeiten, vor, sondern arbeitet streng taktweise, d.h. in jedem Schritt werden die aktuellen Eingangs-Fuzzymengen auf Fuzzymengen am Ausgang abgebildet, die wiederum einen Teil der Eingangs-Fuzzymengen für den nächsten Zeitschritt darstellen. Ein Rückgriff auf ältere Eingangs-Fuzzymengen findet nicht statt.

Ein Verfahren zur automatischen Modellierung des Verhaltens dynamischer Systeme, das auf klassischen Fuzzy-Regelbasen nach Mamdani beruht, ist das ROSA-Verfahren (vgl. Fuzzy-ROSA in [9]). Dieses beruht auf der Definition relevanter Ereignisse, der Vorgabe der potentiellen Auftretenszeitpunkte der Ereignisse und behandelt das Auftreten eines Ereignisses zu einem bestimmten, diskreten Zeitpunkt der Vergangenheit jeweils als eigenständige Fuzzymenge.

Ein anderes, auf Entscheidungsbäumen basierendes Verfahren zur Erzeugung von Fuzzyregelbasen aus Daten wird in [11] vorgestellt. Eine für die maschinelle Modellierung in-

teressante Weiterentwicklung eines bekannten Verfahrens des Maschinellen Lernens wird z.B. in [5] (Fuzzy Decision Trees) beschrieben. [2] definiert geeignete Merkmale von Zeitreihen, um automatisch generierte Entscheidungsbäume zur Prognose verwenden zu können. In [16] wird ein auf Maschinellen Lernverfahren basierender Ansatz zur Modellierung dynamischer Systeme beschrieben, der das Systemverhalten beschreibende, d.h. deskriptive Regeln generiert und so die in diesem Beitrag untersuchte Frage nach einer anwendbaren, d.h. effektiven Regelstruktur motivierte.

12 Zusammenfassung und Ausblick

Die vorgeschlagene Modellstruktur ermöglicht den Umgang mit unscharfen Daten und die explizite Repräsentation zeitlicher Beziehungen im Rahmen eines verständlichen, regelbasierten Modells. Sie ist motiviert durch die Forderung nach einer kognitionspsychologisch adäquaten Repräsentation von mittels maschineller Lernverfahren abgeleiteten Zusammenhängen im Verhalten dynamischer Systeme.

Bei der Analyse prinzipieller Eigenschaften des Modells liegt der Schwerpunkt der Arbeit insbesondere auch auf einem Vergleich mit intuitiven Erwartungen an das Verhalten des Modells, die aufgrund der verbalen Formulierung der Regeln geweckt werden. Gleichzeitig werden Alternativansätze betrachtet, wie in Abschnitt 7 angedeutet wurde.

Wichtigstes langfristiges Forschungsthema ist die Entwicklung eines Systems zur maschinellen Generierung von Regelbasen aus Zeitreihen, das sich an der menschlichen Informationsverarbeitung orientiert [4] und auf dem in [8] beschriebenen Ansatz zur Heuristischen Modellierung beruht.

Literatur

[1] Kurt Dirk Bettenhausen. *Automatische Struktursuche für Regler und Strecke*. TH Darmstadt, 1996.

[2] Michael Boronowsky. Effiziente Interpretation multipler Meßdaten mittels Entscheidungsbauminduktion. In Andreas Fick and Hubert B. Keller, editors, *Workshop Wissensbasierte/Intelligente Systeme in Umweltanwendungen*, pages 39–47. Forschungszentrum Karlsriuhe GmbH, 1999.

[3] Dietrich Dörner and Ute Reichert. Heurismen beim Umgang mit einem "einfachen" dynamischen System. In M. Amelang, editor, *Bericht über den 35. Kongreß der deutschen Gesellschaft für Psychologie*, Heidelberg, 1986. Hogrefe.

[4] Andreas Fick and Hubert B. Keller. Konzeptbildung zur automatischen Modellierung dynamischer Systeme. In Ute Schmid, editor, *Workshop Maschinelles Lernen und Konzepterwerb, Bremen, 15.-17.9.1998*, pages 28–42. TU Berlin, 1998.

[5] Cezary Z. Janikow. Fuzzy decision trees: Issues and methods. *IEEE Transactions on Systems, Man, and Cybernetics - Part b: Cybernetics*, 28(1):1–14, February 1998.

[6] Hubert B. Keller. *Maschinelle Intelligenz*. Vieweg, 2000.

[7] Hubert B. Keller and Andreas Fick. Maschinelle Lernverfahren am Beispiel der thermischen Abfallbehandlung. *KI*, 1998.

[8] Hubert B. Keller and Thomas Weinberger. Heuristische Modellierung - ein Arbeiten mit Hypothesen. In *Workshop Modellierung und Simulation im Umweltbereich*, Rostock, 1992. Universität Rostock, Fb. Informatik.

[9] M. Krabs. *Das ROSA-Verfahren zur Modellierung dynamischer Systeme mit statistischer Relevanzbewertung*. VDI Verlag, 1994.

[10] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy controller. *International Journal of Man Machine Studies*, 7:1–13, 1975.

[11] Ralf Mikut, Jens Jäkel, Hagen Malberg, and Georg Bretthauer. Datenbasierter Entwurf von Fuzzy-Systemen für medizinische Diagnoseaufgaben. *at*, 48(7):317–326, 2000.

[12] Bernd Müller and Hubert B. Keller. Neural networks for combustion process modelling. In *International Conference on Engeneering Applications of Neural Networks (EANN 96)*, 1996.

[13] Elmar Schäfers, Volker Krebs, and Klaus Schmid. Inferenzverfahren für dynamische Fuzzy-Systeme. In *7. Workshop Fuzzy Control des GMA-UA 1.4.2*, pages 165–178, 1997.

[14] Roland Soltysiak. *Wissensbasierte Prozeßregelung*. Oldenbourg, München, Wien, 1989.

[15] M. Sugeno. An introductory survey of fuzzy control. *Information Sciences*, 36:59–83, 1985.

[16] Thomas Weinberger. *Ein Ansatz zur Modellierung dynamischer Systeme*. VDI Verlag, 1995.

[17] Lofti A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[18] Hans-Jürgen Zimmermann, editor. *Proceedings / EUFIT '99 : 7th European Congress on Intelligent Techniques & Soft Computing ; Aachen, Germany, September 13-16, 1999*, Aachen , Mainz, 1999.

Genetische Algorithmen für die Strukturvereinfachung nichtlinearer, ordnungsreduzierter Systeme

Maik Buttelmann, Boris Lohmann

Institut für Automatisierungstechnik, Universität Bremen
Kufsteiner Straße, D – 28359 Bremen
Tel.: +49 (0) 421-218 3906, Fax: +49 (0) 421-218 4707
E-Mail: maik.buttelmann@iat.uni-bremen.de

Kurzfassung
Aufbauend auf einem bekannten Ordnungsreduktionsverfahren für nichtlineare Systeme wird eine Lösung vorgestellt, die auftretenden hohen Systemkomplexitäten der berechneten ordnungsreduzierten Modelle mit Hilfe geeigneter Nebenbedingungen zu minimieren. Die Funktionsweise des hierfür verwendeten Genetische Algorithmus wird anhand eines Beispielsystems demonstriert und die Eigenheiten dieses Optimierungsproblems in Hinblick auf die Verwendung des Genetischen Algorithmus' untersucht.

1 Ordnungsreduktion

Betrachtet werden nichtlineare, zeitinvarante Systeme in der allgemeinen Darstellung

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \tag{1.1}$$

wobei $\boldsymbol{x}(t)$ den n-dimensionalen Zustandsvektor und $\boldsymbol{u}(t)$ den p-dimensionalen Stellgrößenvektor bezeichnen. Bei solchen Systemen hoher Ordnung können Simulation, Analyse und der Reglerentwurf von technischen Systemen sehr rechenintensiv werden. Abhilfe schafft hier die Ordnungsreduktion nach B. Lohmann [1], die ein System niedrigerer Ordnung berechnet, mit dessen Hilfe die Zustände des Originalsystems nachgebildet werden können. Um die Ordnungsreduktion durchführen zu können, werden die nichtlinearen Summanden des Originalsystems (1.1) in dem neuen Vektor $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})$ zusammengefaßt. Somit läßt sich das System in der Form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{F}\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) \tag{1.2}$$

darstellen, wobei jedes Element des Vektor $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ aufgebrochen wird in die Zeitfunktionen $\boldsymbol{x}(t)$, $\boldsymbol{u}(t)$ und $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})$ und in deren Konstanten zusammengefaßt werden in den Matrizen $\boldsymbol{A}$, $\boldsymbol{B}$ und $\boldsymbol{F}$. Ziel des Ordnungsreduktionsverfahrens ist es nun, ein Modell niedrigerer Ordnung

$$\dot{\boldsymbol{x}}_R(t) = \boldsymbol{A}_R\boldsymbol{x}_R(t) + \boldsymbol{B}_R\boldsymbol{u}(t) + \boldsymbol{F}_R\boldsymbol{g}(\boldsymbol{W}\boldsymbol{x}_R, \boldsymbol{u}), \tag{1.3}$$

zu berechnen, das das Verhalten des Originals approximiert, d. h. die wesentlichen oder *dominanten* Zustandsgrößen gut nachbildet. Diese dominanten Zustände werden vom Nutzer vorgegeben und sind über eine Reduktionsmatrix $\boldsymbol{R}$ gemäß

$$\boldsymbol{x}_{do} = \boldsymbol{R} \cdot \boldsymbol{x}(t) \tag{1.4}$$

aus den Originalzuständen zu gewinnen. Das ordnungsreduzierte System bildet also die dominanten Zustände des Originals nach, die übrigen Zustände können mit Hilfe der

ebenfalls vom Reduktionsverfahren berechneten Matrix $\boldsymbol{W}$ aus dem reduzierten System gewonnen werden:

$$\hat{\boldsymbol{x}} = \boldsymbol{W} \cdot \boldsymbol{x}_R \,. \tag{1.5}$$

Die Berechnung der Matrizen $\boldsymbol{A}_R$, $\boldsymbol{B}_R$, $\boldsymbol{F}_R$ und $\boldsymbol{W}$ erfolgt mit Hilfe von Simulationsdaten des Originalsystems und ist in [1] und [3] ausführlich dargestellt.

1.1 Nachteil der Ordnungsreduktion

Es hat sich gezeigt, daß die berechneten Systemmatrizen des ordnungsreduzierten Systems $\boldsymbol{E} = [\boldsymbol{A}_R, \boldsymbol{B}_R, \boldsymbol{F}_R]$ und die Matrix $\boldsymbol{W}$ in der Regel voll besetzt sind. Dies führt zu einer hohen Systemkomplexität, die den Reglerentwurf erschwert.

Um die Ordnung *und* die Komplexität zu reduzieren, können bei der Berechnung des ordnungsreduzierten Modells Nebenbedingungen so aufgestellt werden, daß an bestimmten Stellen Nullelemente in die neu berechneten Matrizen einfügt werden und somit die Komplexität vermindert wird:

$$\boldsymbol{L}_E = \boldsymbol{K}_E \boldsymbol{E} \boldsymbol{H}_E \,, \qquad \boldsymbol{L}_W = \boldsymbol{K}_W \boldsymbol{W} \boldsymbol{H}_W \tag{1.6}$$

In diesen Nebenbedingungen ist definiert, wie viele Nullelemente an welcher Stelle in den Matrizen stehen sollen. Für das Blockschaltbild des Systems bedeutet das Einfügen dieser Nullen, daß Verbindungen zwischen den einzelnen Blöcken gekappt werden. Zum Beispiel führen die Matrizen $\boldsymbol{H}_E = [1, 0, \ldots, 0]^T$, $\boldsymbol{K}_E = [0, 1, 0, \ldots, 0]$ und $\boldsymbol{L}_E = [0]$ dazu, daß in der Matrix $\boldsymbol{E}$ das erste Element der zweiten Reihe zu null wird, und somit in der Matrix $\boldsymbol{A}_R$ das Element $a_{21} = 0$ ist.

Aufgrund der hohen Anzahl an Möglichkeiten die Nebenbedingungen aufzustellen, werden Genetische Algorithmen (GA) eingesetzt, um ein Optimum bezüglich einer einfachen Modellstruktur und einer guten Nachbildung des Originalsystems zu finden [2]. So ergeben sich bei dem hier betrachteten Beispielsystem mit einer relativ niedrigen Ordnung bereits eine Anzahl von $2^{n_{red} \cdot (n_{red} + p + q)} = 2^{3 \cdot (3 + 2 + 2)} = 2^{21} \approx 10^6$ möglichen Lösungen (Ordnung des reduzierten Systems: $n_{red} = 3$, Anzahl der Eingänge $p = 2$ und Anzahl der Nichtlinearitäten $q = 2$) .

Die Berechnung des reduzierten Systems erfolgt somit in zwei Schritten:

1. Aufstellen der Nebenbedingungen mit Hilfe des GA.
2. Berechnung der Matrizen $\boldsymbol{E} = [\boldsymbol{A}_R, \boldsymbol{B}_R, \boldsymbol{F}_R]$ und $\boldsymbol{W}$ mit der herkömmlichen Methode der Ordnungsreduktion unter Berücksichtigung der Nebenbedingungen aus 1.

2 Der Genetische Algorithmus

Das Ablaufschema des eingesetzten GAs ist in der folgenden Abbildung dargestellt:

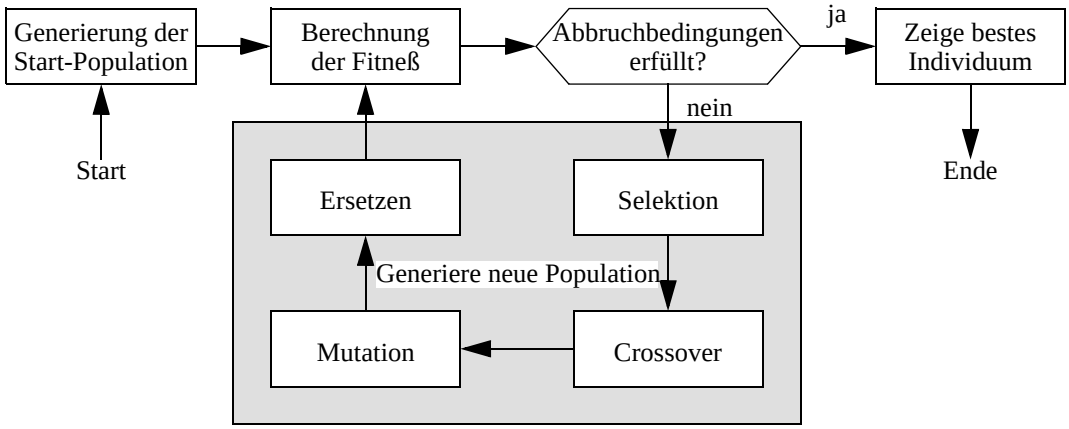


Bild 2.1 Ablaufschema des eingesetzten GAs

2.1 Die Individuen

Genetische Algorithmen imitieren evolutionäre Prozesse unter besonderer Betonung genetischer Mechanismen [5]. In dieser Arbeit wird der GA für die Lösung des Optimierungsprozesses eingesetzt. Dabei arbeitet ein GA mit einer Menge (Population) von künstlichen Individuen. Jedes Individuum ist ein String bestehend aus L Bits, wobei L ein anwendungsabhängiger Wert ist, und stellt eine mögliche Lösung des Optimierungsproblems dar. Jeder String gliedert sich außerdem in m Segmente ($m \leq L$). Jedes Segment korrespondiert zu einer Variablen des betrachteten Optimierungsproblems. Segment j (j = 1, 2, ... , L) enthält in binär codierter Form einen Wert für die Entscheidungsvariable j des Optimierungsproblems. Segmente können gleichlange oder verschieden lange Bitfolgen enthalten. Bei dem hier betrachteten Problem ist ein Segment lediglich ein Bit lang. Es enthält die Information, ob an der entsprechenden Stelle der Matrix $\boldsymbol{E}$ bzw. $\boldsymbol{W}$ eine Null eingefügt wird: Eine Null in dem Individuum bedeutet *nein*, füge kein Nullelement in die Matrix ein, eine Eins bedeutet *ja*, füge eine Null an der entsprechenden Stelle in die Matrix ein. Die Anzahl der Segmente m entspricht also der Anzahl der Elemente in der Matrix $\boldsymbol{E}$ bzw. $\boldsymbol{W}$.

Ein Beispiel:

Die Matrix $\boldsymbol{E}$ soll folgendes Aussehen haben:

$$\boldsymbol{E} = \begin{bmatrix} 0 & x & 0 \\ x & 0 & x \end{bmatrix}, \text{ mit } x\text{: Nicht-Nullelement}$$

Das entsprechende Individuum lautet dann:

$$I = [\underbrace{1\ 0\ 1}_{\text{1. Zeile von } \boldsymbol{E}}\ \underbrace{0\ 1\ 0}_{\text{2. Zeile von } \boldsymbol{E}}]$$

2.2 Die Fitneßfunktion

Um die Güte der einzelnen Individuen zu bestimmen, wird eine dem Problem angepaßte Fitneßfunktion F verwendet:

$$F = \underbrace{\frac{\sum_{i=1}^{n} \int_0^{\infty} (x_i(t) - \boldsymbol{w}_i x_{R,i}(t))^2 dt}{\sum_{i=1}^{n} \int_0^{\infty} x_i^2(t)}}_{\text{Modellnachbildung}} \underbrace{- k \cdot \text{sum}(I)}_{\text{Modellkomplexität}}. \qquad (2.1)$$

Der Teil *Modellnachbildung* der Fitneßfunktion berechnet die Fläche zwischen den einzelnen Zuständen des Originalsystems und des reduzierten Systems. Der Teil *Modellkomplexität* besteht lediglich aus der Anzahl der vorhandenen Einsen in einem Individuum (also der Anzahl der Nullelemente in $\boldsymbol{E}$ bzw. $\boldsymbol{W}$) multipliziert mit einem beliebigem Faktor $k \geq 0$. Da die Modellbildung demnach die Differenz zwischen Originalsystem und reduziertem System ist, sind kleine Fitneßwerte besser als große, es handelt sich hier also um ein Minimierungsproblem mit dem Optimum $F = 0$, wenn $k = 0$ ist.

2.3 Die genetischen Operationen

Im folgenden soll kurz auf die verwendeten genetischen Operationen *Selektion*, *Crossover*, *Mutation* und *Ersetzen* eingegangen werden.

Selektion

Angewendet wird die Wettkampfselektion [5], bei der aus der Population ξ Individuen ($2 \leq \xi < \mu$, μ: Populationsgröße) mit gleicher Selektionswahrscheinlichkeit $p_S = 1/\mu$ gezogen werden und das Beste unter ihnen für die Erzeugung von Nachkommen ausgewählt wird. So werden μ Individuen ausgewählt und in einen Pool eingefügt, wobei durch die Veränderung des Parameters ξ der Selektionsdruck verändert werden kann.

Crossover

Es werden zwei Eltern aus dem Pool ausgewählt und mit der Wahrscheinlichkeit $p_C = 0{,}7$ ein 2-Punkt-Crossover durchgeführt, um zwei Nachkommen zu erzeugen:

```
Eltern:                    Nachkommen:
0 0 1 | 1 1 | 0 0 →  0 0 1 | 0 1 | 0 0
1 0 0 | 0 1 | 1 0 →  1 0 0 | 1 1 | 1 0
```

Die Crossover-Stellen werden dabei zufällig bestimmt.

Mutation

Die erzeugten Nachkommen werden dann mit der Wahrscheinlichkeit $p_M = 1 / L$ (L: Länge der Individuen, vorgeschlagen von Mühlenbein in [6]) mutiert, wobei ein stochastisch gewähltes Bit gekippt wird:

```
Nachkomme:                 Nachkomme, mutiert
0 0 1 0 1 0 0      →       0 0 1 0 1 1 0
```

Ersetzen

Die erzeugten Nachkommen ersetzen ihre Eltern vollständig in der neuen Population. Alternativ kann das beste oder die besten Eltern mit in die neue Population übernommen werden, wobei sie die schlechtesten Nachkommen ersetzen.

2.4 Abbruchbedingung

Da bei diesem Optimierungsproblem der bestmögliche Fitneßwert bekannt ist, nämlich null, können hier leicht zwei Abbruchkriterien formuliert werden:

1. Die Fitneß des besten Individuums unterschreitet einen bestimmten Fitneßwert.
2. Die Höchstanzahl an berechneten Generationen wurde erreicht.

Solange die Abbruchbedingung nicht erfüllt ist, wird der GA durchlaufen und es werden immer neue Generationen von Individuen generiert.

3 Anwendung an einem Beispielsystem

Um die Funktionsweise zu überprüfen wurde ein einfaches Beispielsystem gewählt, dessen Ergebnisse sich gut interpretieren lassen. Es handelt sich dabei um einen Prüfstand, bestehend aus einem Verbrennungsmotor, der durch eine flexiblen Welle mit einer federnd gelagerten Wirbelstrombremse verbunden ist [1]. Eingangsgrößen des Systems sind die Drosselklappenstellung u_1 des Verbrennungsmotors und die Steuergröße u_2 der Wirbelstrombremse.

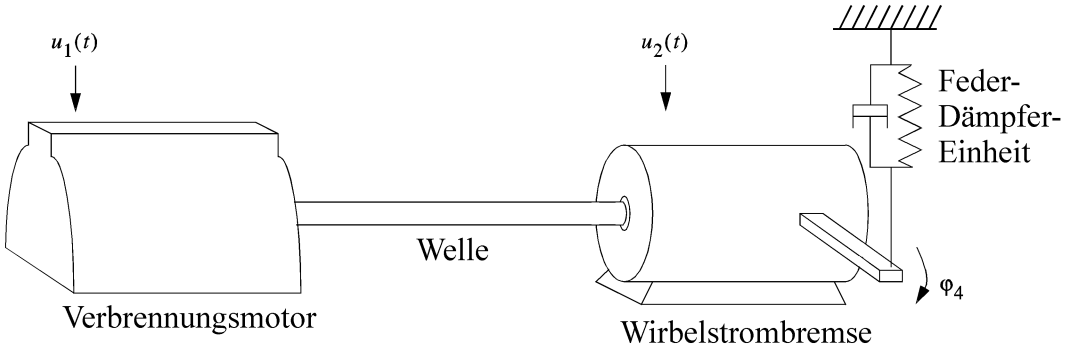


Bild 3.1 Aufbau des Prüfstandes

Der schematische Aufbau ist im nächsten Bild dargestellt. Die flexible Welle wird dabei recht aufwendig durch drei Schwungmassen, zwei Federn und zwei schwach wirkenden Dämpfern nachgebildet.

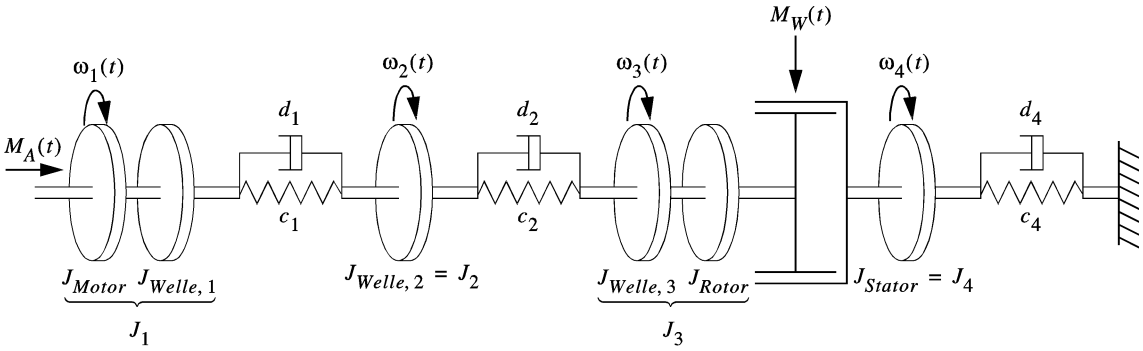


Bild 3.2 Schematischer Aufbau des Prüfstandes, $n = 7$

Dieses System hat eine Ordnung von $n = 7$, wobei die Zustände die vier Drehzahlen ω_1 bis ω_4 und die drei Momente M_1 (an J_1 und J_2 angreifendes Moment), M_2 (an J_2 und J_3 angreifendes Moment) und M_4 (an J_4 angreifendes Moment) sind. Die Systemmatrizen und der Vektor $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})$ lauten dann wie folgt:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 20/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 2000 & -100 & 0 & 0 & 0 \\ 5000 & 0 & 0 & 0 & -1020 & -5000 & 1000 \\ 0 & 0 & 0 & 0 & 333{,}3 & 0 & -333{,}3 \\ 0 & -5000 & 0 & 0 & 1000 & 5000 & -1300 \end{bmatrix}, \boldsymbol{F} = \begin{bmatrix} 20/3 & 0 \\ 0 & -100 \\ 0 & 2 \\ 0 & 100 \\ 20 & 0 \\ 0 & 0 \\ 0 & 300 \end{bmatrix},$$

$$\boldsymbol{B} = \boldsymbol{0}, \qquad \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} (50 + 50 \cdot u_1) \cdot e^{-\left(\frac{x_1 + 200 - 600 \cdot u_1}{200 + 300 \cdot u_1}\right)} \\ u_2^2 \cdot (x_2 - x_3) \end{bmatrix},$$

mit den Zustandsgrößen

$$\boldsymbol{x}^T = \begin{bmatrix} \omega_1 & \omega_3 & \omega_4 & M_4 & M_1 & \omega_2 & M_2 \end{bmatrix}.$$

Um eine Ordnungsreduktion *von Hand* durchzuführen, werden die Zustände $x_1 = \omega_1$, $x_3 = \omega_4$ und $x_4 = M_4$ als dominant angesehen. Es wird also die Systemordnung von 7 auf 3 reduziert. Physikalisch hat das die Bedeutung, daß die Welle nunmehr als starr angesehen und nur noch durch eine Schwungmasse modelliert wird, wie im nächsten Bild zu sehen ist.

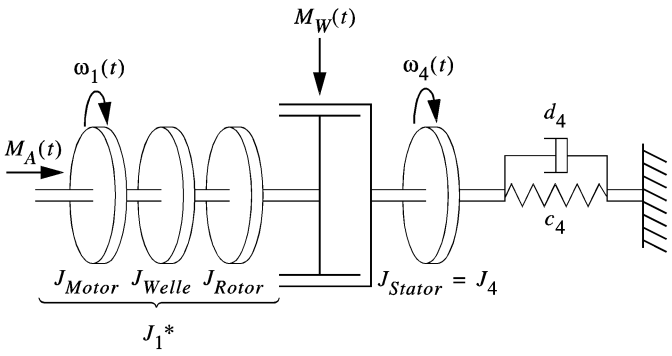


Bild 3.3 Schematischer Aufbau mit starrer Welle, $n = 3$

Die Systemmatrizen des von Hand reduzierten Systems dritter Ordnung lauten dann wie folgt:

$$\boldsymbol{A}_R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 2000 & -100 \end{bmatrix}, \quad \boldsymbol{B}_R = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \boldsymbol{F}_R = \begin{bmatrix} 6{,}135 & -6{,}135 \\ 0 & 2 \\ 0 & 100 \end{bmatrix}. \quad (3.1)$$

Dieses Modell gibt das Verhalten der drei Zustandsgrößen ω_1, ω_4 und M_4 sehr gut wieder, wie in Bild 3.4 zu sehen ist. Mit Hilfe der oben angegebenen Fitneßfunktion wird hier ein Wert von 289 ermittelt.

Für die nun folgenden *Berechnungen* des reduzierten Systems werden dem Algorithmus ebenfalls die drei Zustandsgrößen $x_1 = \omega_1$, $x_3 = \omega_4$ und $x_4 = M_4$ als dominant vorgegeben. Der Algorithmus *ohne* Nebenbedingung liefert dann die Systemmatrizen

$$\boldsymbol{A}_R = \begin{bmatrix} 0 & -44 & -3{,}2 \\ 0 & -0{,}1 & -2 \\ 0 & 2000 & -100 \end{bmatrix}, \quad \boldsymbol{B}_R = \begin{bmatrix} -2{,}5 & -22 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \boldsymbol{F}_R = \begin{bmatrix} 6 & -2{,}6 \\ 0 & 2 \\ 0 & 100 \end{bmatrix}. \quad (3.2)$$

Auch dieses System bildet die drei Zustandsgrößen ω_1, ω_4 und M_4 sehr gut wieder, allerdings ist zu erkennen, daß die Systemkomplexität im Vergleich zu dem von Hand reduzierten Modell zugenommen hat. Der Fitneßwert dieses Modells beträgt 205.

Werden nun mit Hilfe des GA Nebenbedingungen erzeugt, so erhält man nach einigen Generationen als beste Lösung folgende Systemmatrizen:

$$\boldsymbol{A}_R = \begin{bmatrix} 0 & 0 & -33{,}3 \\ 0 & 0 & 0 \\ 0 & 0 & -100 \end{bmatrix}, \quad \boldsymbol{B}_R = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \boldsymbol{F}_R = \begin{bmatrix} 6 & -2{,}7 \\ 0 & 0 \\ 0 & 100 \end{bmatrix}. \quad (3.3)$$

Wie zu erkennen ist, nimmt nicht nur die System*komplexität* ab, sondern auch die System*ordnung*, die jetzt nur noch 2 beträgt. Der Fitneßwert beträgt bei diesem System 82. Physikalisch macht die Reduzierung auf Ordnung $n_R = 2$ durchaus Sinn. Das System „Prüfstand“ besteht aus zwei Subsystemen: zum einen dem rotieren System aus der Schwungmasse des Verbrennungsmotors, der Welle und dem Anker der Wirbelstrombremse, zum anderen aus der Masse des federnd gelagerten Stators der Wirbelstrom-

bremse. Diese beiden Subsysteme sind lediglich über das Moment gekoppelt, das vom elektrischen Feld in der Wirbelstrombremse erzeugt wird. Die Zustandsgröße x_2, die Drehzahl ω_4 des Stators der Wirbelstrombremse, wird demnach nur sehr gering bei sprunghaften Stellgrößenänderungen angeregt, der Optimierungsalgorithmus setzt daher diese Größe zu null.

Die folgende Abbildung zeigt die Verläufe von sämtlichen sieben Zustandsgrößen, wie sie im Originalsystem vorkommen. Die fehlenden vier Größen werden bei den ordnungsreduzierten Systemen mit Hilfe der Matrix $\boldsymbol{W}$ berechnet. Dargestellt sind die Verläufe der Zustandsgrößen des Originalsystems, des berechneten Modells *ohne* Nebenbedingungen (linker Fitneßwert in Bild 3.4), des von Hand reduzierten Modells (mittlerer Fitneßwert) und des berechneten Modells *mit* Nebenbedingungen (rechter Fitneßwert). Bei allen sieben Zustandsgrößen ist zu erkennen, daß es keine signifikanten Unterschiede in den Zeitverläufen gibt und die drei hier präsentierten Modelle das Originalsystem ausreichend genau nachbilden.

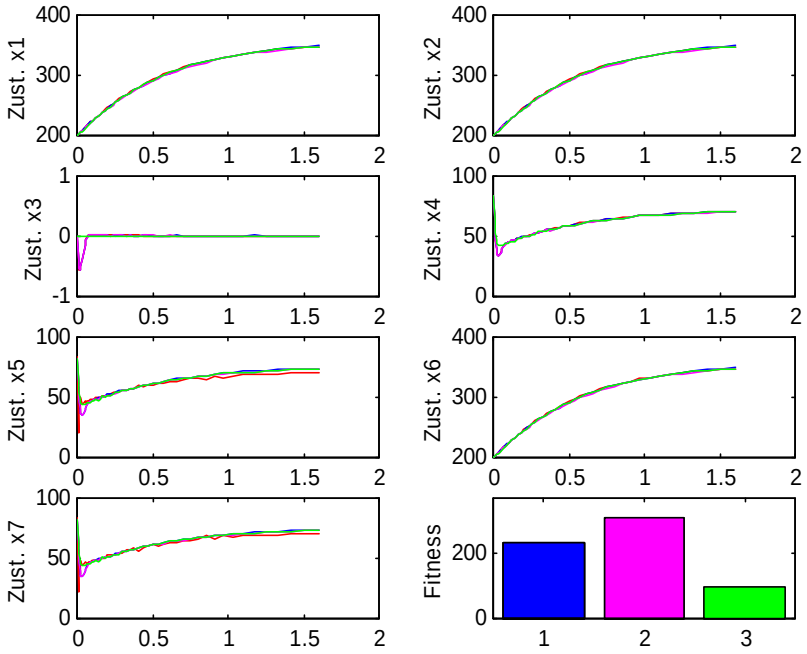


Bild 3.4 Zeitverläufe der Zustandsgrößen und Fitneßwerte

4 Interpretation der Ergebnisse

Der Einsatz von Genetischen Algorithmen macht nur dann Sinn, wenn das Optimierungsproblem eine gewisse innere Ordnung hat, so daß der Algorithmus nicht zu einem stochastischen Suchprozeß wird. Wird zum Beispiel das Prinzip „kleine Ursache, kleine Wirkung“ verletzt, erschwert dieses den Optimierungsprozeß [5]. Ein wichtiges Kriterium ist dabei die Fitneßänderung (Wirkung), wenn ein Bit in einem Individuum gekippt wird (Ursache). Dafür wird eine Art „Suchraum“ oder „Fitneßlandschaft“ erstellt, mit der eine Aussage über die Eigenschaften des Optimierungsproblems getroffen werden soll. Bei diesem Problem ist es allerdings nicht möglich, die Fitneßfunktion F als Funktion zweier Variablen darzustellen, um so einen dreidimensionalen Suchraum zu plotten, wie z. B. von vielen Benchmark-Funktionen her bekannt. Um trotzdem eine grafische Darstellung des Problems zu erhalten, wird von dem besten Individuum ausgegangen, das in Formel (3.2) dargestellt ist. Zunächst werden die Fitneßwerte sämtlicher Individuen mit

der Hamming-Distanz[1] eins berechnet und hiervon die Besten ausgewählt. Für diese werden jeweils wiederum die Fitneßwerte der Individuen mit der Hamming-Distanz eins berechnet und die Besten ausgewählt. So werden die Fitneßwerte von Individuen bis zu einer beliebigen „Tiefe“ ermittelt. Die Individuen der Tiefe k erfüllen dabei folgende Kriterien: Sie haben die Hamming-Distanz von eins zu dem korrespondierenden Individuum der Tiefe $k - 1$ und die Hamming-Distanz k zu dem besten Individuum, dem Ausgangspunkt der Berechnungen.

Die grafische Darstellung der ermittelten Werte erfolgt in der nächsten Abbildung. Wiederum ausgehend vom absolut besten Individuum mit den Koordinaten (0, 0, 82) werden auf einem Kreis mit dem Radius eins die Fitneßwerte der fünf besten Individuen mit der Hamming-Distanz eins geplottet. Auf einem Kreis mit dem Radius zwei wird dann jeweils nur noch der Fitneßwert des bestes Individuums mit der Hamming-Distanz zwei zum absolut besten und der Hamming-Distanz eins zum korrespondierenden Individuum der Tiefe eins geplottet usw. Hier dargestellt sind die Fitneßwerte bis zur Tiefe $k = 15$.

Es ist zu erkennen, daß bis zu einer Tiefe von $k = 10$ die Änderung der Fitneßwerte nur sehr gering ist und dann stark ansteigt. Es gibt demnach eine recht große Anzahl von guten Individuen in der Nähe der besten (hier gefundenen) Lösung, was in der rechten Grafik in Bild 4.1 noch einmal deutlicher dargestellt ist. Geplottet wurden jetzt jeweils die drei besten Individuen mit der Hamming-Distanz eins eines jeden Individuums. Auch hier liegen die Fitneßwerte der Individuen in einem recht kleinen Intervall [82; 95], womit zumindest hier die Forderung „kleine Ursache, kleine Wirkung“ erfüllt ist.

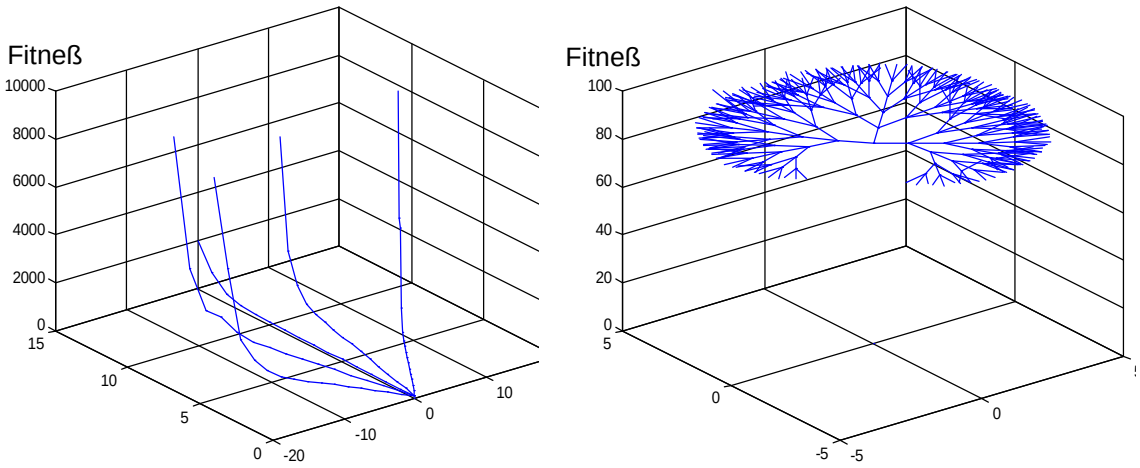


Bild 4.1 Darstellung des Suchraumes

Für das hier betrachtete Optimierungsproblem bedeutet dies, daß es eine relativ große Anzahl an Individuen mit guter Fitneß gibt, die sich alle in der Nähe der besten Lösung befinden und eine sehr große Anzahl an Individuen mit sehr schlechter Fitneß, die eine große Hamming-Distanz zur besten Lösung aufweisen. Bei der Durchführung des GA macht sich dieses dahingehend bemerkbar, daß es nach nur wenigen Generationen eine hohe Anzahl von Individuen in der Population gibt, die bereits eine gute Fitneß aufweisen. Wird dann der Selektionsdruck, der zu Beginn der Suche einen Minimalwert von $\xi = 2$ aufweist *nicht* erhöht, stagniert meistens die Fitneß der bis dahin gefundenen besten Lösung. Erst bei Erhöhung des Selektionsdrucks werden wieder bessere Lösungen

[1] Die Hamming-Distanz beschreibt die Anzahl der Bits, die zwei Strings voneinander unterscheidet. Sind z. B. die beiden Strings bis auf ein Bit identisch, so beträgt die Hamming-Distanz eins.

gefunden. Das liegt auch daran, daß bei dem Crossover von zwei guten Lösungen, die sich bereits in der Nähe des optimalen Individuums befinden, kein Individuum entstehen kann, daß eine schlechte Fitneß aufweist. Die beiden guten Lösungen enthalten bereits physikalisch sinnvolle Nebenbedingungen für die Ordnungsreduktion, die aufgrund der Segmentlänge von eins nicht mehr verworfen werden können.

Eine Erhöhung der Mutationsrate anstatt des Selektionsdruckes hingegen würde nur zu einer zufälligen Suche führen. Gerade bei der Mutation können durch Kippen eines ungünstigen Bits Individuen entstehen, die wieder eine sehr schlechte Fitneß aufweisen. Hier ist demnach nicht immer die Forderung „kleine Ursache, kleine Wirkung“ erfüllt.

5 Zusammenfassung und Ausblick

Es wurde ausgehend von dem bekannten Ordnungsreduktionsverfahren nach B. Lohmann eine Lösung vorgeschlagen, die auftretende hohe Systemkomplexität der reduzierten Modelle zu reduzieren, indem mit Hilfe eines Genetischen Algorithmus' geeignete Nebenbedingungen für die Berechnung der ordnungsreduzierten Systeme generiert werden. Anhand eines Beispielsystems wurde die Funktionsweise demonstriert und auf die Besonderheiten dieses Optimierungsproblemes eingegangen.

In der Zukunft wird der Algorithmus an einem weiteren System getestet, das eine wesentlich höhere Ordnung aufweist als das hier vorgestellt. Es bleibt dabei abzuwarten, ob der Suchraum die gleichen Charakteristika aufweisen wird: schlechte Fitneß für Individuen mit einer großen Hamming-Distanz zum Optimum, gute Fitneßwerte für Individuen in der Nähe des Optimums. Ferner könnte untersucht werden, ob es sinnvoll ist, die Bits zu markieren, bei deren Mutation eine signifikante Fitneßänderung auftritt. So ließen sich für diese Bits schnell die geeigneten Werte finden, was den Suchraum stark einschränken würde.

6 Literatur:

[1] Lohmann, B.: *Ordnungsreduktion und Dominanzanalyse nichtlinearer Systeme*, VDI-Fortschrittsberichte, Reihe 8, VDI-Verlag, Düsseldorf, 1994.

[2] Buttelmann, M., Lohmann, B.: *Model Simplification And Order Reduction of Nonlinear Systems With Genetic Algorithms.* Proceedings of the IMACS Symposium on Mathematical Modellung, 3rd MATHMOD, Vienna 2000, p. 777 – 781.

[3] Lohmann, B.: *Order Reduction and Determination of Dominant State Variables of Non-linear Systems*. In: Proc. of the IMACS Symposium on Mathematical Modelling, 1st MATHMOD, Vienna 1994, 239 – 243 and Mathematical Modelling of Systems, Vol. 1, (1995), 77 – 90, (extended version).

[4] Schöneburg, E, et al.: *Genetische Algorithmen und Evolutionsstrategien*. Addison-Wesley, 1994.

[5] Nissen, V.: *Einführung in Evolutionäre Algorithmen.* Vieweg, 1997.

[6] Mühlenbein, H.: *How Genetic Algorithm Really Work I. Mutation and Hillclimbing*. In: Männer, R., Manderick, B. (Hrsg.): *Parallel Problem Solving from Nature, Proceedings of the Second Conference on Parallel Problem solving from Nature*. Amsterdam, North-Holland, 1992.

Einsatz und Entwurf wissensbasierter analytischer Regler mit der Engineering- und Informationsverarbeitungssoftware MaxXControl®

Jens- Uwe Müller, Christian Rähder

Hochschule Zittau/Görlitz
Fachbereich Elektrotechnik
Theodor Körner Allee 16
02763 Zittau
Tel. (03583) 611292
Fax. (03583) 611293
E-Mail j.mueller@hs-zigr.de

1 Notwendigkeit für den Einsatz wissensbasierter Systeme

Für viele technologische Prozesse ist die Erstellung eines mathematischen Modells aus Aufwandsgründen kaum vertretbar. Zusätzlich erschwert der Einfluß nicht meßbarer Störgrößen Aussagen über die aktuelle Prozeßsituation.

Da Steuer- und Regelverfahren weitgehend auf der Auswertung von Meßgrößen bzw. über Modelle rekonstruierter Prozeßgrößen beruhen, wird vielfach nur ein Minimum der möglichen Informationen über den Prozeß ausgewertet. Dies gilt besonders für die Verfahrenstechnik und insbesondere für Prozesse bei der Herstellung von Nahrungs- und Genußmitteln. Unterschiedliche Eigenschaften von natürlichen Rohstoffen oder die Aktivitätsvarianzen von Enzymen oder Hefen führen zu veränderten Prozeßeigenschaften, die ein optimal arbeitender Steuer– bzw. Regelalgorithmus verarbeiten muß.

In der Praxis werden solche Einflüsse im allgemeinen durch empirisch begründete Strukturen- und Parametermodifikationen berücksichtigt. Häufig zeigt sich auch, daß kritische Situationen vom Bedienpersonal per Handbedienung gefahren werden. Der erfahrene Operator ist dabei vielfach „erfolgreicher“ als eine Automatik, obwohl ihm keine anderen Meßgrößen zur Verfügung stehen. Der Unterschied liegt in der erfahrungsgestützten Bewertung der Prozeßsituation mit Hilfe der Prozeßhistorie und der impliziten koordinierten Verknüpfung aller Informationen aus seinem Erfahrungsschatz.

Eine Automatisierungseinrichtung mit diesen menschlichen Fähigkeiten muß sich demnach eines großen Datenspeichers bedienen sowie Informationen komprimieren und in geeigneter Weise verknüpfen können. Eine Selbstanpassung an sich verändernde Situationen emuliert den Lernvorgang eines menschlichen Bedieners.

Konventionelle bzw. modellbasierte Verfahren bieten hierfür nur eingeschränkte Möglichkeiten und führen zu Strukturen, die oftmals für Dritte schwer verständlich sind. Die Pflege und Wartung solcher Regler erweisen sich dann als problematisch.

Demgegenüber bieten wissensbasierte Systeme geeignete Verfahren, wie z.B. Fuzzy-Control, Neuro-Fuzzy oder das wissensbasiert analytische Regelverfahren (WAR) [3], die erforderlichen Eigenschaften für die Implementierung sowohl heuristischen als auch algorithmierbaren Prozeßwissens:

- linguistische Bewertung von Prozeßgrößen und damit eine Komprimierung des Informationsgehalts der Signale
- Darstellung des Prozeßwissens in Steuerregeln auf Basis bewerteter Signale
- Generierung eines Steuer- und Regelgesetzes für einen festgelegten Arbeitsbereich aus den einzeln formulierten Steuerregeln (Wissensbasis)
- Methoden zur Selbstanpassung der Wissensbasis

Angeregt durch die im Einsatz sehr praxisnahen Möglichkeiten der Fuzzy-Set Theorie, in der Automatisierungstechnik als Fuzzy-Control bezeichnet, wurde 1991 an der Technischen Hochschule Zittau das WAR-Verfahren entwickelt.

2 Das wissensbasiert analytische Regelverfahren (WAR)

Kernstück des WAR-Verfahrens (Bild 1) ist ein n-dimensionaler Operatorbaustein, der die Verknüpfung einer Ausgangsgröße u mit n-1 Prozeßgrößen x_i erlaubt.

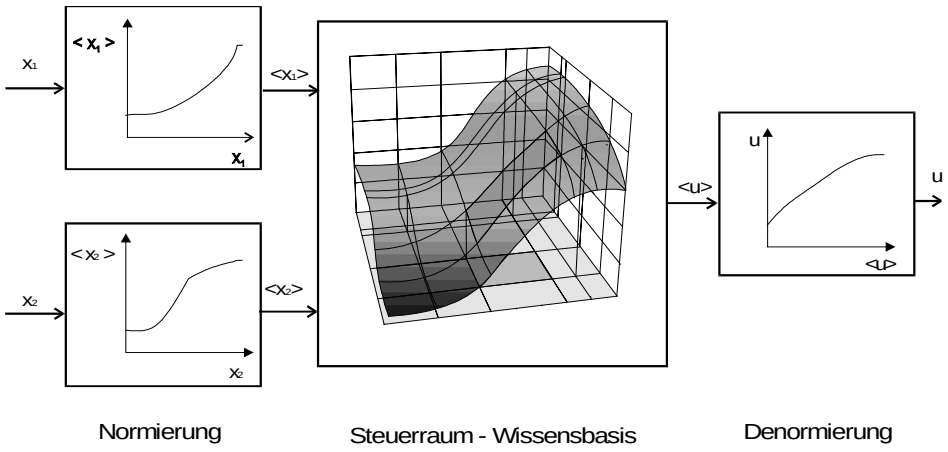


Bild 1: WAR-Grundmodul

Jeder wissensbasierte Regler kann mit einem Satz von Steuerregeln (punktuell bekannten Steuerregeln) eine Zielgröße (z.B. Stellgröße) in Abhängigkeit der Prozeßgrößen x_i beschreiben. Dazu ist es notwendig, eine Diskretisierung (Normierung) der einzelnen Eingangsgrößen x_i in $1...m_i$ Merkmale vorzunehmen. In einer Normierungsfunktion (Bild 1) werden diese primär durch ganze Zahlen ($1...m_i$) beschrieben und physikalischen Eingangsgrößenwerten zugeordnet. Auf Anforderung ist eine Verknüpfung mit linguistischen Bezeichnern (Aliasnamen) möglich. Somit steht für die Beschreibung (Adressierung) des Steuerraumes ein einheitliches auf Integerwerten basierendes Datenformat zur Verfügung.

Die Steuerregeln werden in einem regelmäßigen Gitterraster im Steuerraum (Bild 2), welcher durch die normierten Eingangsgrößen aufgespannt wird, angeordnet. Ein solches Rasterfeld wird als Teilsteuerraum bezeichnet, in welchem eine lokale Interpolation zwischen den Steuerregeln für die Ergebnisbildung erfolgt. Die Gesamtheit aller Teilsteuerräume definiert ein zusammenhängendes Steuer- bzw. Regelgesetz.

Da auf den normierten Steuerraum direkt zugegriffen werden kann, können beliebige punktuelle Zusammenhänge in diesem verarbeitet werden. Dies bedeutet, daß neben linguistischem Wissen auch andere Informationsquellen (z.B. Meßdaten, analytische Regelgesetze, Ventilkennlinien) direkt in den Entwurf einbezogen werden können.

Diese hybriden Wissensbasen sind für einen effektiven Einsatz des Verfahrens sehr vorteilhaft.

Die dritte Verarbeitungsstufe, die Denormierung (Bild 1), wird nur bei einem linguistisch geprägten Entwurf notwendig und kann als Umkehrung der Normierung verstanden werden.

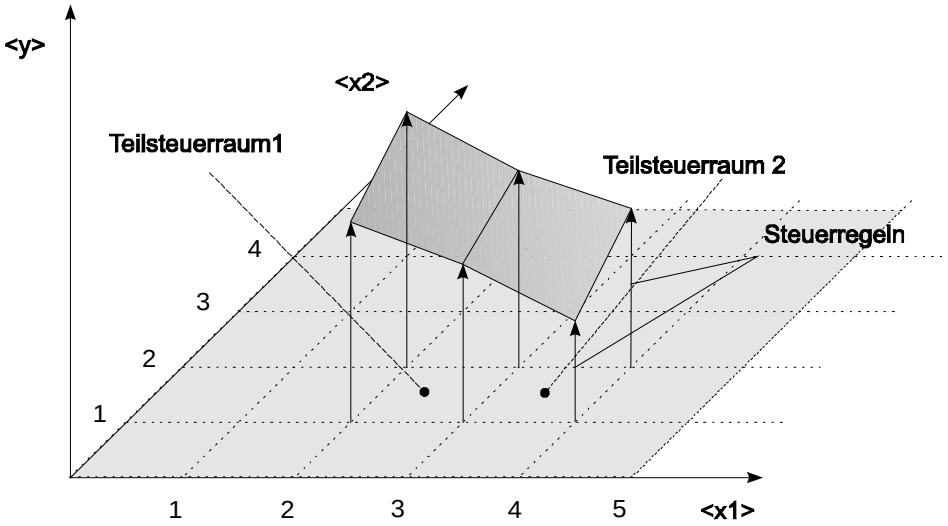


Bild 2: Teilsteuerraum

Im Softwarepaket MaxXControl ist das WAR-Modul nach Bild 1 als Baustein verfügbar und kann beliebig mit anderen Bausteinen verknüpft werden. Mit dieser Strukturierung ist auch der praktikable Aufbau größerer Wissensbasen möglich. Ein Vorschlag für eine universell nutzbare Reglerstruktur mit Hilfe verschiedener WAR-Module und deren Anpassung an sich verändernde Prozeßeigenschaften wird nachfolgend beschrieben.

3 Entwurf und Optimierung WAR-basierter Regler

3.1 Standardstruktur

Vorteilhaft für einen praxisgerechten Einsatz und die Akzeptanz eines Regelverfahrens ist die Angabe einer einfachen Struktur, z.B. einer Grundschaltung mit P, I und D-Wirkungen, welche vom Nutzer nur noch parametriert werden muß.

In vielen Anwendungen des WAR-Verfahrens in der Verfahrens- und Kraftwerkstechnik [4][5] hat sich die in Bild 3 dargestellte Standardstruktur bewährt. Die Kombination aus Vorsteuerung, PID-Reglerkomponenten und einem speziellen Modul für besondere Situationen ist für Praktiker einfach nachzuvollziehen und bietet dabei die Möglichkeit einer optimalen Anpassung des Reglers an komplexe Prozesse. Diese Struktur ermöglicht

- eine Bilanzierung des Prozesses auf Basis der meßbaren Stör- und Führungsgrößen,
- eine situationsbewertete Ausregelung von nichtmeßbaren Störeinflüssen,
- eine Zusammenfassung von heuristischem Prozeßwissen, hier Situationsregelwerk genannt,

- günstige Schnittstellen für die Regleroptimierung, da dynamische und statische Anteile getrennt betrachtet werden,
- die Wirkungen einzelner Anteile bzw. Steuerregeln bei der Inbetriebnahme bzw. Adaption des Reglers besser zu erfassen.

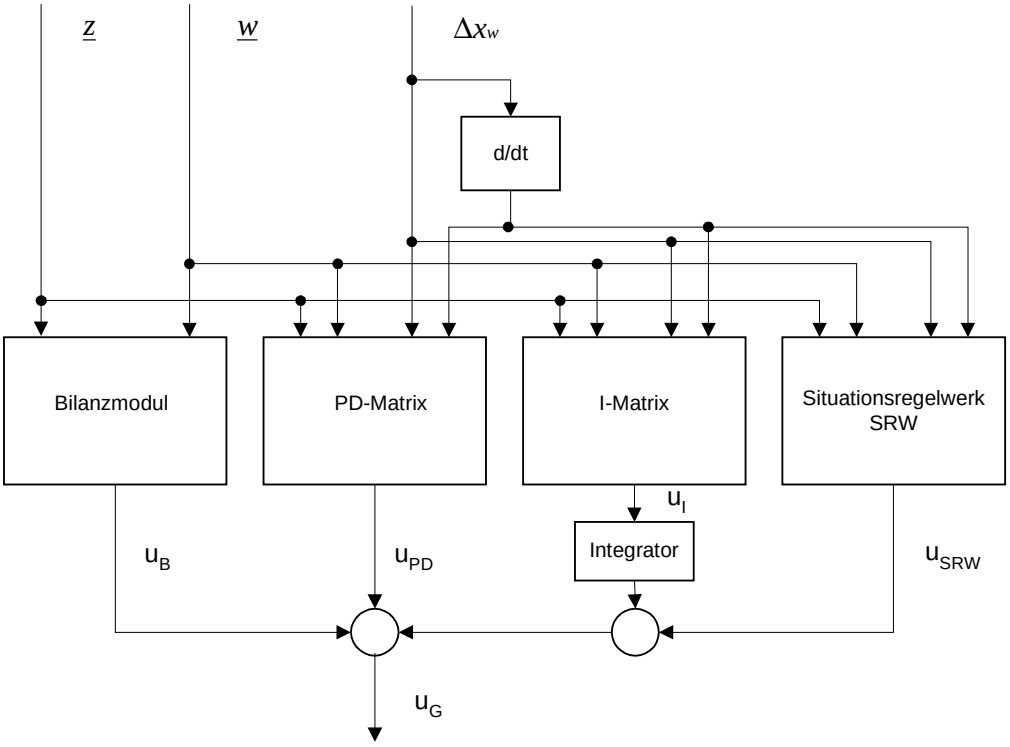


Bild 3: WAR-Standardstruktur

3.2 Statische Optimierung

Beim Entwurf eines Reglers ist es kaum zu erwarten, daß die Bilanz und alle Reglerparameter optimal an den Prozeß angepaßt sind. Treten während des Betriebes zusätzliche Parameterdriften wie z.B. durch Verschmutzungen oder Alterungsvorgänge auf, wird der Bedarf eines einfachen und robusten Hilfsmittels für die Anpassung der Wissensbasis deutlich.

Da i.d.R. nichtlineare Prozesse betrachtet werden, ist es praktikabel, die erforderliche Anpassung der Wissensbasis arbeitspunktbezogen vorzunehmen. Es wird demnach eine Methode benötigt, die eine gezielte Anpassung bestimmter Steuerregeln vornimmt.

Durch lokale Interpolation im Steuerraum werden arbeitspunktbezogen nur die Regeln eines Teilsteuerraumes geändert. Dafür wurde eine Formel (1) abgeleitet, die eine Steuerregel in Abhängigkeit ihres Einflusses auf den Ausgangswert u_B im zu optimierenden Arbeitspunkt anpaßt.

$$\Delta S_i = \alpha \cdot \frac{\Delta u_B \cdot b_i^2}{\sum_{j=1}^{n} b_j^2} \qquad (1)$$

Δu_B Bilanzfehler
b_j Einflußfaktor einer Steuerregel j $b \in [0...1]$
α Lernfaktor $\alpha \in [0{,}1...1]$

Der exakten Bilanzierung technologischer Prozesse kommt eine besondere Bedeutung zu. Das Anfahren neuer Arbeitspunkte nach Führungsgrößenänderungen oder eine Störgrößenkompensation kann ohne Rückkopplung über die Zustandsgrößen des Systems dynamisch günstig ausgeführt werden. Der Stellgrößenanteil des Bilanzmoduls in der Standardstruktur ist deshalb auch sehr groß (>90%). Kommt es jedoch zu Fehlbilanzierungen, müssen diese zusätzlich vom Regler kompensiert werden. Eine Verminderung der Regelgüte ist die Konsequenz einer fehlerhaften Bilanz.

Bild 4 zeigt die Struktur der statischen Bilanzoptimierung im Rahmen der WAR-Standardstruktur [1]. Das Bilanzmodul ist mit dem Optimisator gekoppelt. Dieser erhält seine Informationen aus den meßbaren Störgrößen und dem I-Anteil des Reglers. Befindet sich der Prozeß in Ruhe, d.h. sind die Ausgleichs- und Übergangsvorgänge abgeschlossen, kann der Wert des Integrators als Bilanzfehler interpretiert werden. Der Optimisator bestimmt über eine statistische Auswertung der Prozeßhistorie, welche Teilsteuerräume für diesen „Fehler“ verantwortlich sind und korrigiert die Wissensbasis.

Im praktischen Einsatz hat sich das vorgestellte Modul vielfach bewährt und ist als Baustein im Softwaresystem MaxXControl verfügbar. Beim Einsatz des Optimisators gibt es verschiedene Randbedingungen, wie z.B.:

- Prozesse, in denen sich durch Störeinflüsse praktisch kein statischer Zustand einstellt,
- eine zu grobe Steuerraumrasterung (zu wenig Stützstellen),
- nichtmeßbare Störungen, die keine eindeutigen Rückschlüsse von Stellgrößen und den im Bilanzmodul zugeordneten Arbeitspunkten zulassen,

die eine statische Optimierung erschweren, was sich in schlechten Konvergenz-eigenschaften oder fehlerhaften stark zerklüfteten Kennfeldern widerspiegelt.

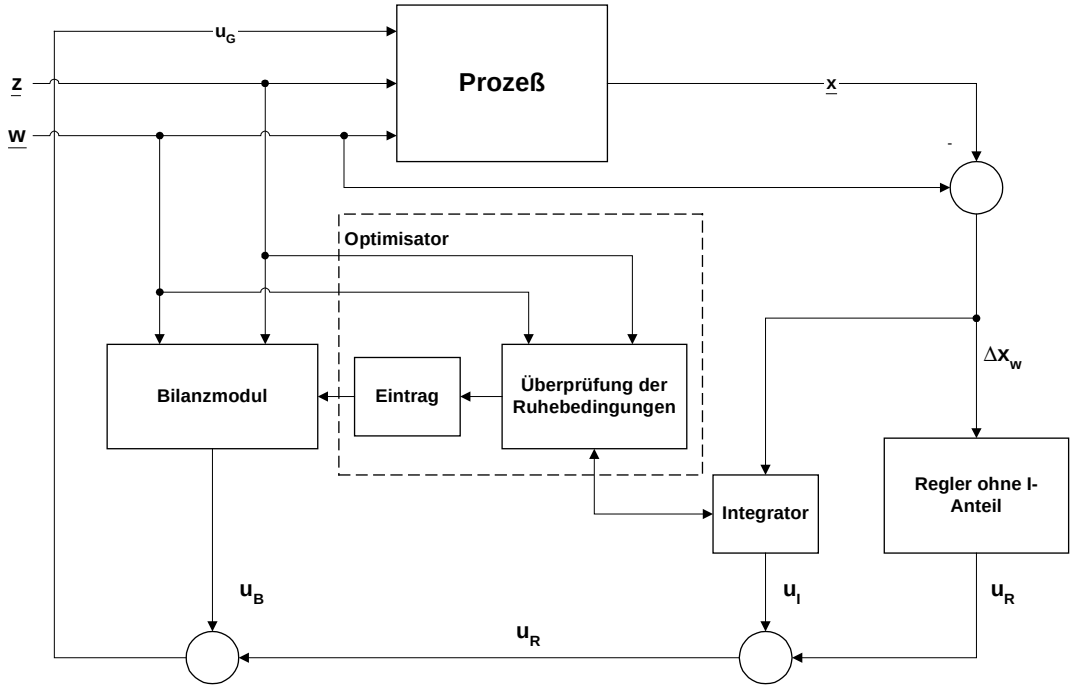


Bild 4: Blockschaltbild des statischen Optimisators

4 Softwaresystem MaxXControl®

Vor der Umsetzung eigener Applikationen in die Praxis stand die Frage, kann eine Software gekauft werden oder ist eine Eigenentwicklung sinnvoller. Die Entscheidung fiel zugunsten einer Eigenentwicklung mit dem Hintergrund, daß

- zum fraglichen Zeitpunkt keine Software zur Verfügung stand, die eine unkomplizierte und robuste Umsetzung unserer Entwicklungen ermöglicht hätte,
- die Schnittstellenproblematik wesentlich vereinfacht wird (Unabhängigkeit) und
- erworbenes Know-how bei dem Einsatz der Software direkt in dessen Weiterentwicklung fließen kann.

Für die IPC-basierte realisierte Umsetzung von Steuer- und Regelalgorithmen wurde ein Runtimesystem (Abtastzeiten>2ms) auf der Basis von Windows NT entwickelt.

Neben dem Simulationskern wurde Wert auf das zentrale Verwalten

- von Meßdaten,
- der Ein- und Ausgangssignale und
- der Reglerstrukturen

gelegt. Bild 5 vermittelt eine Übersicht über Leistungsumfang und Oberfläche des Engineeringsystems. Hervorzuhebende Eigenschaften von MaxXControl® sind:

- Echtzeitsimulationskern mit Abtastzeiten>2ms
- Runtimesystem mit Paßwortschutz und Hintergrundaktivität
- Automatischer und konventioneller Reglerentwurf für PID-Strukturen, sowohl ein- als auch mehrschleifig
- Objektorientierte übersichtliche Projektverwaltung (Signale, Meßdateien, Funktionspläne, wissensbasierte Reglermodule)
- Flexible Meßsignaldefinition, -erfassung, - verarbeitung und -darstellung
- Vollständig unterstützter Entwurf für WAR-Reglersysteme
    - Einfache Verbindung analytischer Regelgesetze (z.B. automatisch Übernommen aus arbeitspunktsensitivem PID-Reglerentwurf) mit wissensbasierten Regelungsstrategien
    - Hohe Transparenz durch einheitlich aufgebauten WAR-Baustein, sowie durch Unterstützung linguistischer Variablen
    - Automatische Online Optimierung durch kennfeldadaptierende Algorithmen
- Komfortabler graphischer Funktionsplaneditor mit umfangreicher Bausteinbibliothek als Basis für Simulationsrechnungen (ohne Echtzeit) als auch für die Ausführung entworfener Reglerstrukturen am Prozeß
- Erstellung übersichtlicher Dokumentationen in Anlehnung an bestehende Dokumentationsstandards
- verschiedene Prozeßschnittstellen für die Einbindung einer PC-basierten Lösung in bestehende Systeme (z.B. OPC, Datenbankkopplung, PROCONTROL P, EXCEL (Honeywell))

Auf der Basis des WAR-Verfahrens sind mit dem System MaxXControl verschiedene Applikationen erfolgreich ausgeführt worden:

- Kraftwerksregelkreise
- Regelung von Umweltschutzanlagen
- Regelung von Klimaanlagen für Bürogebäude und Produktionshallen
- Regelung von Heizungszentralen
- Regelung des Läutervorgangs in einem Brauerei Sudhaus
- Feuerleistungsregelung von Müllverbrennungsanlagen
- Regelung des Bierfiltrationsvorgangs (Kerzenfilter)

Bild 5: Oberfläche und Leistungsumfang des Softwaresystems MaxXControl

5 Applikationen

5.1 Erfahrungen

Während der langjährigen Arbeit mit dem WAR-Verfahren und dem Engineering-System MaxXControl in den Bereichen Kraftwerks- und Verfahrenstechnik (Müllverbrennungsregelung, Entstickungsregelungen), der Gebäudeautomatisierung und der Lebensmittelherstellung (Brauprozeß) konnten zahlreiche Rückschlüsse auf die Leistungsfähigkeit, die Einsatzschwerpunkte und die zu beachtenden Randbedingungen beim Einsatz PC-gestützter wissensbasierter Regler gezogen werden. Die betrachteten Prozesse zeichnen sich durch ein dynamisch problematisches (z.B. erhebliche Streckentotzeiten) nichtlineares Verhalten sowie durch eine über stetige Funktionen beschreibbare Prozeßbilanz aus.

Als vorteilhaft hat sich die einfache Synthese von steuerregelbasiertem (heuristischem) und algorithmierbarem (z.B. Kennlinien) Wissen in einem universell einsetzbaren Baustein erwiesen. Es ist damit möglich, die meßbaren Prozeßgrößen beliebig über Steuerregeln zu verknüpfen und somit verschiedene Prozeßsituationen zu erfassen. Nichtmeßbare Störungen sind dabei oftmals aus der Prozeßhistorie der meßbaren Größen rekonstruierbar. So ist es einem erfahrenen Operator möglich, einen gestörten Prozeß in vielen Fällen wieder in den gewünschten Zustand zu bringen. Eine Eigenschaft die auch durch wissensbasierte Verfahren emulierbar ist.

Ein weiterer Schwerpunkt ist die Software für den Reglerentwurf und -umsetzung an der Anlage. Neben der Formulierung der Wissensbasis (Steuerregeln) sind erfahrungsgemäß eine Reihe von weiteren Funktionen notwendig, wie z.B.

- Meßsignalfilterung,
- Bildung von Merkmalen aus Meßsignalen,
- Prozeßkopplung oder
- Ableitung dynamischer Größen aus den Meßsignalen.

Folgende Probleme sind eng mit der Leistungsfähigkeit des Regelverfahrens und der zur Verfügung stehenden Engineering-Software verbunden:

- **Zeitproblematik**
  Je einfacher die Umsetzung und die Änderung von Wissensbasis und Randbeschaltung ausfällt, desto mehr Zeit steht für die Prozeßbeobachtung und Regleranpassung zur Verfügung.

- **Komplexität**
  Je komplizierter eine heuristische Aussage umsetzbar ist, desto schwieriger wird ihre Validierung und Verifizierung.

- **Strukturierung**
  Je mehr Informationen über den Prozeß gezielt zusammenfließen können, desto höher ist die zu erwartende Regelgüte. Dies kann z.B. einfach über die WAR-Standardstruktur erfolgen.

Mit dem System MaxXControl konnten in dieser Hinsicht positive Erfahrungen gesammelt werden. Als Beispiel für die Anwendung des Verfahrens soll an dieser Stelle die Regelung des Läutervorgangs beim Brauprozeß angeführt werden [2][5].

5.2 Regelung des Läutervorgangs im Brauereiprozeß

5.2.1 Anlagenbeschreibung

Im Brauereiprozeß besteht die Aufgabe, nach dem Maischen die festen Bestandteile (Spelzen, Stärkegriese usw.) von der zucker- bzw. extrakthaltigen Würze zu trennen. Dafür wird häufig ein Läuterbottich genutzt (Bild 6).

Nach dem Umpumpen der Maische in den Läuterbottich setzen sich auf dem Siebboden die festen Bestandteile langsam ab und bilden einen Filterkuchen. Darunter befindet sich eine Absaugvorrichtung, die die Würze durch den Filterkuchen und den Siebboden zieht und in den Würzekocher pumpt.

Der Prozeß des Läuterns ist in zwei Schritte unterteilt. Im ersten wird die eingelassene Maische abgepumpt, bis die Würze in den Filterkuchen einzieht. Im zweiten Schritt wird in zeitlicher Folge immer wieder Wasser auf den Kuchen gegeben (angeschwänzt), um diesen optimal auszulaugen.

Während des Filtervorgangs wird der Filterkuchen komprimiert, was bei konstantem Volumenstrom einen Soganstieg (Unterdruck) zur Folge hat, da die Läuterpumpe gegen einen höheren Widerstand arbeiten muß. Eine Lockerung des Filterkuchens durch ein höhenverstellbares Hackwerk kompensiert diesen Effekt.

Mit den Stellgrößen der Hackwerkshöhe, der Schneidgeschwindigkeit und dem Volumenstrom des Filtrats (Abläutergeschwindigkeit) kann der Filtervorgang beeinflußt werden.

Bei der Steuerung bzw. Regelung des Filtervorgangs wird Wert auf eine geringe Trübung des Filtrats bei maximaler Geschwindigkeit und höchster Ausbeute (geringer Extraktanteil im Filterkuchen) gelegt.

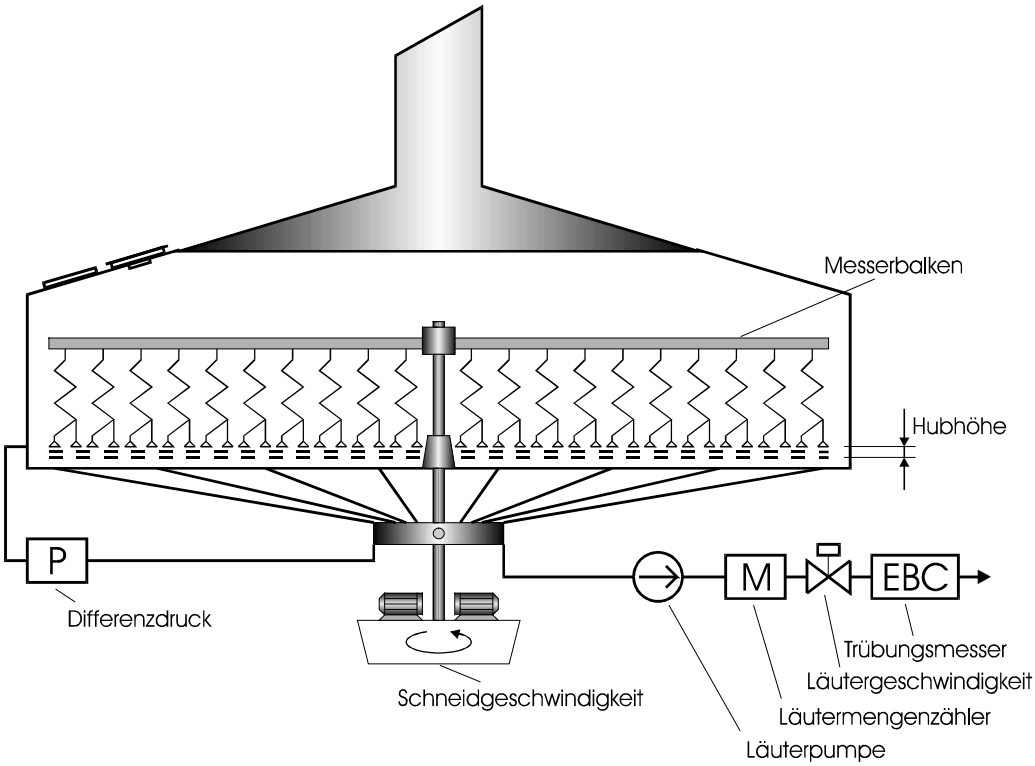


Bild 6: Schematische Darstellung einer Läuteranlage

5.2.2 Prozeßeigenschaften

Für den Brauprozeß ist eine hohe Qualität des Produktes bei gleichzeitigem Minimieren der Läuterzeit und dem Erreichen einer maximalen Ausbeute[1] Ziel einer optimalen Prozeßregelung, womit folgende technologischen Zielstellungen verbunden werden können:

- Einhalten der vorgegebenen Läuterzeit (Läutern ist Teil eines Batchprozesses)
- Maximales Auslaugen des Filterkuchens (Restextrakt unter 1%)
- Minimierung der Trübung (1...5 EBC)

Dabei sind mit den Stellgrößen Pumpenleistung, Schneidgeschwindigkeit und Hubwerkshöhe, die Zielgrößen Läuterzeit, Trübung[2] und Ausbeute zu koordinieren.

Für den Filtervorgang sind besonders die Zusammenhänge zwischen Filterwirkung, Filterkuchenkonsistenz und Pumpenleistung (bzw. dem Sog unter dem Filter) interessant. Durch den Sog wird der Filterkuchen verdichtet und ändert seine Eigenschaften, was teilweise durch die Auflockerungen mit dem Schneidwerk kompensiert werden kann. Die Wirkungen sind nichtlinear und zeitvariant. Hinzu kommt der Einfluß der Rohstoffeigenschaften auf den Filterkuchen bzw. die -wirkung. Schon die Entnahmehöhe aus dem Malzsilo und die damit verbundene Variation der Feinstoffanteile (Staub) im Malz ändern die Filtereigenschaften u.U. erheblich.

Beachtet werden muß, daß die technologischen Ziele z.T. konträre Stellreaktionen erfordern. Für die Ausbeute und die Trübung wäre z.B. ein möglichst langsames Abläutern von Vorteil. Durch die Verlängerung der Kontaktzeit des Filterkuchens mit der flüssigen Würze wird dieser stärker ausgelaugt.

5.2.3 Reglerstruktur

Ausgehend von der vorhandenen läutermengenabhängigen Steuerung der Stellgrößen wurde der WAR-Läuterregler (Bild 7) auf Basis der Standardstruktur entwickelt [2][5].

Jede Stellgröße setzt sich aus einem arbeitspunktabhängigen PI-Regler und einer Bilanzgröße zusammen. In die Bilanzierung werden die Stellgrößen zusätzlich über die Koordinationsmodule der drei technologischen Ziele Zeit, Ausbeute und Trübung beeinflußt. Die Hauptwirkung des Reglers wird über die Bilanzmodule erzeugt. Die darin eingestellten mengenabhängigen Profile (Kennlinien) werden durch die technologischen Zielstellungen variiert. Der Regler arbeitet abhängig vom Widerstand des Filterkuchen und paßt die Stellwerte damit den vorhandenen Malz- bzw. Filtereigenschaften an.

---

[1] Minimierung des Extraktanteils im Filterkuchen (Treber)

[2] Minimierung des Trubstoffanteils in der gefilterten Würze

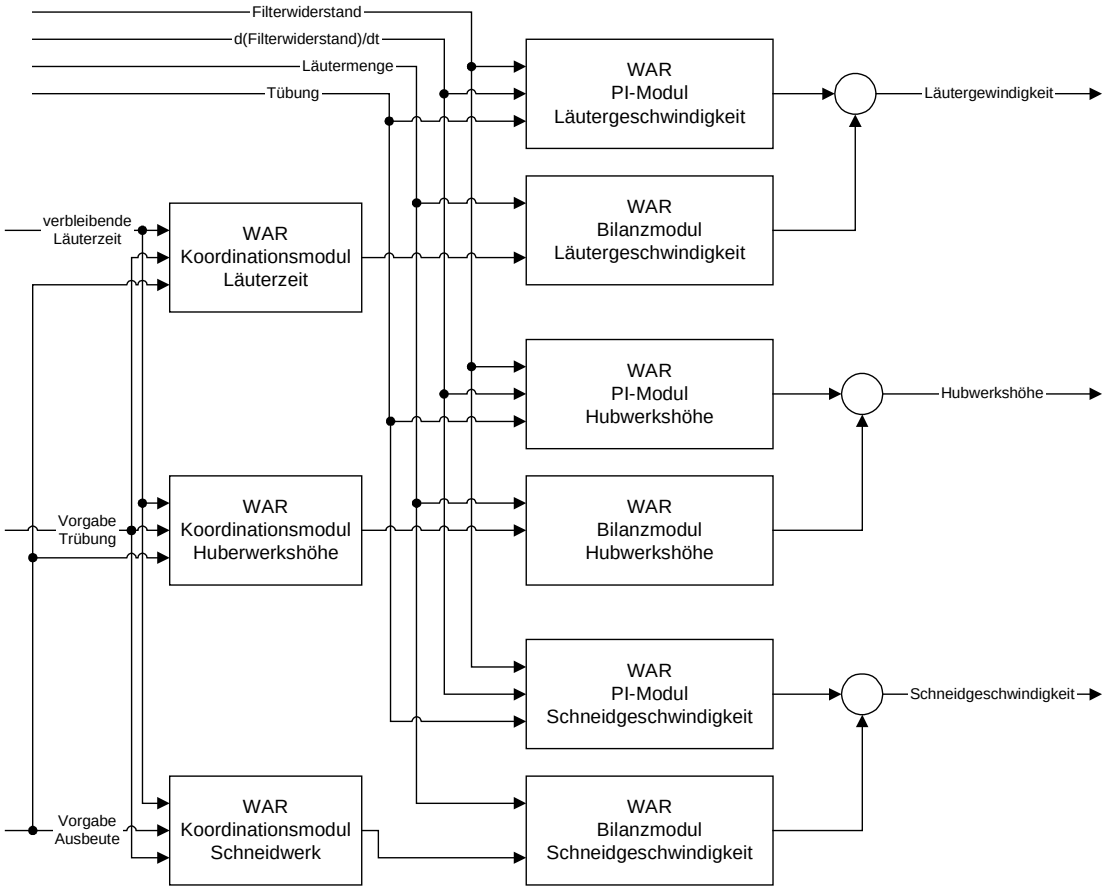


Bild 7: Struktur des WAR-Läuterreglers

5.2.4 Ergebnisse

Die vorgestellte Läuterbottichregelung ist in mehreren Brauereien installiert worden. Hauptanliegen beim Einsatz in bestehende Anlagen war eine Verringerung der Läuterzeiten, um Spitzenbelastungen besser abfangen zu können oder um die Sudwoche zu verkürzen. Mit dem vorgestellten Läuterregler konnten Einsparungen von 10...30% erreicht werden. Der Vergleich der beiden Läutervorgänge nach Bild 8 zeigt eine Verringerung der Läuterzeit von ca. 30%.

Neben der Aufgabe den Läutervorgang zu beschleunigen, besteht zusätzlich die Möglichkeit der Änderung technologischer Ziele und damit der flexibleren Anpassung an die Vorgaben des Marktes (z.B. Sommer-Winterbetrieb).

Bild 8 zeigt den Vergleich der WAR-Regelung mit einer konventionellen Lösung. An der Größe Differenzdruck tritt die unterschiedliche Arbeitsweise der beiden Regelungen am deutlichsten hervor. Der WAR-Regler ist bestrebt, das System unter dem maximal zulässigen Unterdruck zu halten, damit Tiefschnitte und irreversible Filterkuchen-kompressionen zu vermeiden und somit eine Läuterzeitverkürzung von 10...30% zu erreichen.

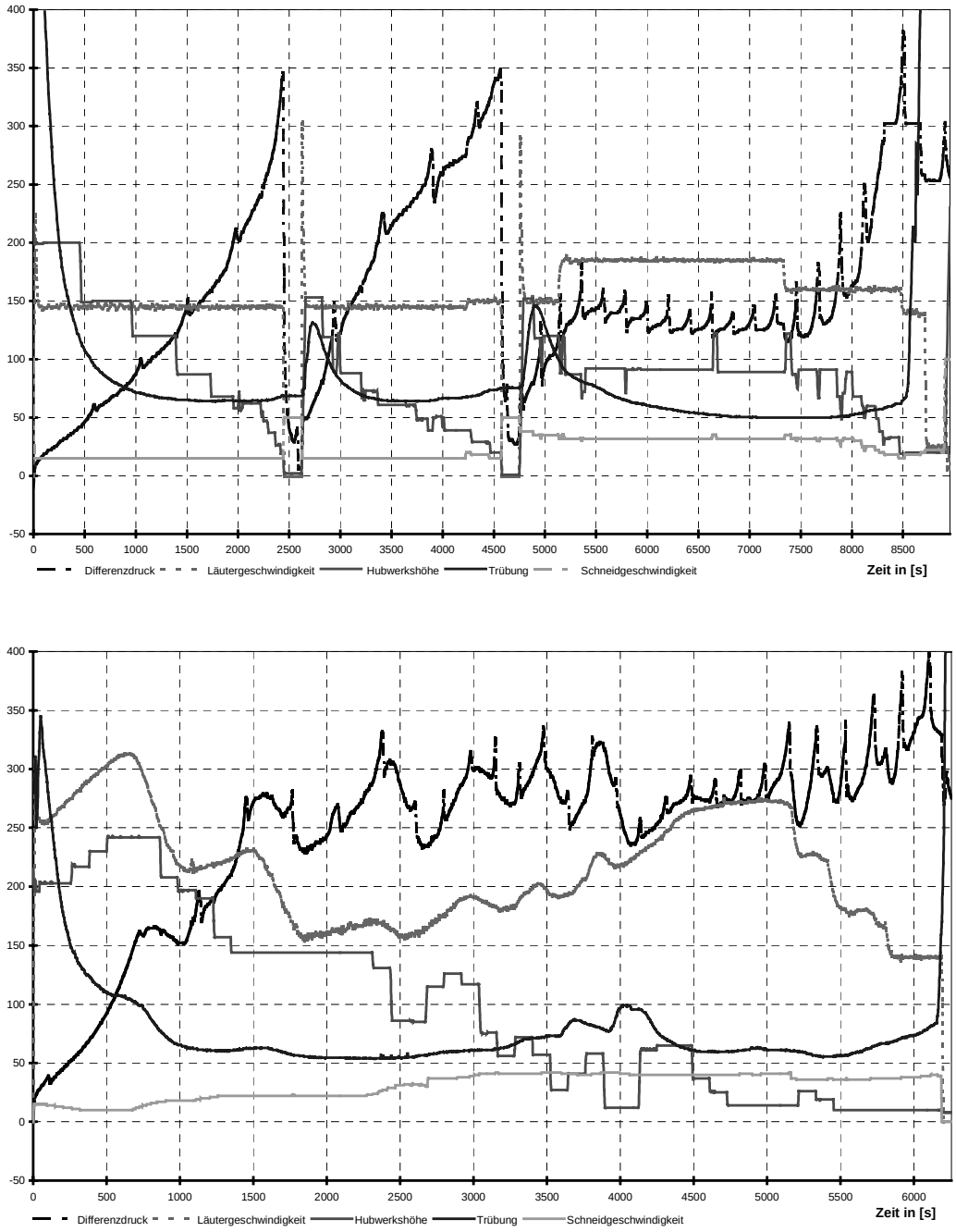


Bild 8*:* Konventionell (oben) und mit WAR (unten) geregelter Läutervorgang

6 Zusammenfassung

Wissensbasierte Systeme wurden speziell in den letzten zehn Jahren vielfach für Aufgaben in der Prozeßreglung eingesetzt (z.B. [4][5]). Die flexible Anpassung solcher Systeme an den Prozeß und der damit verbundenen umfangreichen Informationsverarbeitung ermöglicht in vielen Fällen eine erhebliche Verbesserung der Regelgüte gegenüber konventionellen Lösungen, wie z.B. in Müllheizkraftwerken.

Sehr wichtig für die Anpassung des Reglers bzw. der Wissensbasis sind geeignete Entwurfs- und Optimiermethoden. Eine schnelle und vorgezeichnete Methodik für den Entwurf solcher Systeme erhöht deren Praxisakzeptanz erheblich. Die robuste Opti-

mierung eines wissensbasierten Systems kann die Regelgüte auch bei Parameterdriften oder in Arbeitspunkten, die in der Entwurfs- und Erprobungsphase nicht explizit berücksichtigt werden konnten, erheblich verbessern.

Als Baustein ist im Softwaresystem MaxXControl u.a. das WAR-Modul für die Verknüpfung von Informationen und ein statischer Optimisator verfügbar, welcher in verschiedensten Applikationen getestet wurde. Komplizierter gestaltet sich erfahrungsgemäß die Anpassung dynamischer Parameter. Die Randbedingungen für eine dynamische Optimierung sind wesentlich schwieriger zu berücksichtigen als für statische Situationen. Eine sichere Methode, die das selbständige Arbeiten eines dynamischen Optimisators im Bereich stochastisch gestörter Prozesse der Verfahrenstechnik zuläßt, konnte bisher nicht gefunden werden.

Beim Entwurf von Wissensbasen hat sich gezeigt, daß für die Aufbereitung von Informationen, wie z.B. verschiedener gemittelter Vergangenheitswerte, Gradienten bzw. andere bewertete Größen (z.B. Strömungswiderstände) oftmals ein erheblicher Aufwand betrieben werden muß, was bei komplexen Prozessen zu umfangreichen Strukturen führt. Der übersichtlichen Strukturierung und Verwaltung von Informationen kommt daher eine große Bedeutung zu.

Die PC-basierte Automation mit wissensbasierten Verfahren besitzt erhebliches Potential beim Entwurf- und bei der Umsetzung eines Reglers für komplexe Prozesse. Es ist einfach und unkompliziert möglich, Änderungen vorzunehmen, benötigte Hilfsgrößen zu berechnen und diese mit anderen Größen zu einen Stellsignal zu verknüpfen. An dieser Stelle sei jedoch betont, daß damit keine Freistellung von einer gründlichen Prozeßanalyse verbunden ist. Nur wenn die Eigenschaften und Einflußgrößen eines Prozesses hinreichend aufgeklärt sind, kann eine optimale Konfiguration von Wissensbasis und Regler erfolgen.

7 Literatur

[1] Dauscha, A.
Entwicklung eines statischen Optimisators für WAR-Regelsysteme
Diplomarbeit, HTWS Zittau/Görlitz 1992

[2] Hagen, R.
Die neue Dimension des Abläuterns
Huppmann Post Ausgabe Nr. 19 November 1999 S. 6-8

[3] Müller, J.-U.; Rähder, Ch.
Wissensbasierte Steuer- und Regeleinrichtung
Internationale Patentanmeldung Mp-Nr. 91/640

[4] Müller, J.-U.; Rähder, Ch.
Ergebnisse der Erprobung eines wissensbasierten analytischen Regelverfahrens an verschiedenen Regelkreisen
GMA Kongreß 1996 Mess- und Automatisierungstechnik, VDI Berichte 1282 S.163-181

[5] Rähder, Ch; Müller, J-U; Hummel, M.
Wissensbasierte Regelung steigert die Leistung von Anlagen
Maschinenmarkt, Würzburg 103 (1997) S. 26-29

Nonlinear System Identification with Global and Local Soft Computing Methods

Thomas A. Runkler

Siemens AG, Zentralabteilung Technik
Information und Kommunikation
81730 München
Tel.: (089) 636–45372
Fax: (089) 636–49767
E-Mail: Thomas.Runkler@mchp.siemens.de

Abstract

An important step in the design of control systems is system identification. Data driven system identification finds functional models for the system's input output behavior. Regression methods are simple and effective, but may cause overshoots for complicated characteristics. Neural network approaches such as the multilayer perceptron yield very accurate models, but are black box approaches which leads to problems in system and stability analysis. In contrast to these global modeling methods crisp and fuzzy rule bases represent local models that can be extracted from data by clustering methods. Depending on the type and number of models different degrees of model accuracy can be achieved.

Key words: regression, multilayer perceptron, rule based system, singleton, Takagi–Sugeno model, clustering, fuzzy c–means, elliptotypes

1 Introduction

Design of control systems requires information about the underlying processes. Input, output and intermediate signals have to be defined, structural and topological information about the system blocks and their interconnections has to be provided, and the input–output behavior of individual blocks has to be estimated. The estimation of the input–output behavior from experimental data is called *identification* or *modeling*. *Linear* systems can be easily modeled using the data from step response experiments. For *nonlinear* systems these step response data are often not sufficient. Instead, data from many different standard as well as unusual operation modes are collected and used to build system models. We denote the measured data of the p system inputs as $X = \{x_1, \ldots, x_n\} \in \mathbb{R}^p$ and the corresponding data of the q system outputs as $Y = \{y_1, \ldots, y_n\} \in \mathbb{R}^q$, so that each input vector x_k, $k = 1, \ldots, n$, corresponds to the output vector y_k. Notice that this formally reflects a static relation between input x and output y. However, if input and output components are defined as derivatives or integrals of the actual signals, arbitrary dynamics can be incorporated into this scheme. A PID system with input u and output w, for example, can be realized by defining

$$x = (u, \dot{u}, \int_0^t u \, dt) \quad \text{and} \quad y = w. \tag{1}$$

Also notice that this scheme does not say anything about the time when the signals x and y were sampled. The n input output data tuples may or may not be obtained by equidistant sampling

$$x_k = x(T + k \cdot \Delta t), \quad y_k = y(T + k \cdot \Delta t), \quad k = 1, \ldots, n. \tag{2}$$

If the data are sampled in this equidistant way, the computation of derivatives, for example, becomes very simple:

$$\dot{x}_k = (x_k - x_{k-1})/\Delta t. \tag{3}$$

Given the data sets X and Y modeling can be viewed as the following problem: Find a function or an algorithm that (at least approximately) reproduces the output data Y from the input data X. If we drop the distinction between functions and algorithms we can write $f : X \to Y$ for both, and thus require

$$f(x_k) \approx y_k, \quad k = 1, \ldots, n. \tag{4}$$

We distinguish *global* and *local* modeling approaches. *Global modeling* tries to identify explicit functions f that fit all the measured data with a certain accuracy. We illustrate this for various regression methods in section 2. Neural network approaches use a multitude of models, but none of those is visible as a local model, so we discuss these approaches in the framework of global modeling in section 3. *Local modeling* tries to find an appropriate partition of the input and/or output space and identify local functions f_i for each local subspace that fit the data well. A very simple approach to local modeling using clustering and crisp rule bases with constant models is presented in section 4. We extend this approach to fuzzy rules in section 5. In section 6 we finally show how clustering and fuzzy rule based approaches can be used to extract arbitrary local models, and illustrate in particular the case of local *linear* models.

2 Modeling by regression

A popular method to build global models is *regression*. Assume for simplicity a single input single output (SISO) system, i.e. $p = q = 1$. *Linear regression* finds a linear function f

$$y \approx f(x) = a + b(x - \bar{x}) \tag{5}$$

specified by the parameters $a, b \in \mathbb{R}$, where

$$\bar{x} = \frac{1}{n} \sum_{k=1}^{n} x_k \tag{6}$$

is the average of all the input data. With the linear *function prototype* (5) the regression problem (and thus also the modeling problem) reduces to the problem of finding *good* parameters a and b. A popular way of specifying *good* models is the minimization of the square error criterion: The best model is defined as the minimum of the mean square error

$$E = \frac{1}{n} \sum_{k=1}^{n} (y_k - f(x_k))^2. \tag{7}$$

For the linear regression model (5) in particular we obtain

$$E = \frac{1}{n}\sum_{k=1}^{n}\left(y_k - a - b(x_k - \bar{x})\right)^2. \tag{8}$$

Necessary conditions for (local) extrema of E are

$$\frac{\partial E}{\partial a} = -\frac{2}{n}\sum_{k=1}^{n}\left(y_k - a - b(x_k - \bar{x})\right) = 0, \tag{9}$$

$$\frac{\partial E}{\partial b} = -\frac{2}{n}\sum_{k=1}^{n}(x_k - \bar{x})\left(y_k - a - b(x_k - \bar{x})\right) = 0. \tag{10}$$

Solving these equations yields the solutions for the optimal parameters a and b:

$$a = \bar{x}, \tag{11}$$

$$b = \frac{\sum_{k=1}^{n}(x_k - \bar{x})(y_k - \bar{y})}{\sum_{k=1}^{n}(x_k - \bar{x})^2}. \tag{12}$$

If we denote the variance of x as

$$v_{xx} = \sum_{k=1}^{n}(x_k - \bar{x})^2 \tag{13}$$

and the covariance beween x and y as

$$v_{xy} = \sum_{k=1}^{n}(x_k - \bar{x})(y_k - \bar{y}), \tag{14}$$

then (12) becomes

$$b = v_{xy}/v_{xx}. \tag{15}$$

This means that optimal regression parameters can be immediately obtained from the statistical measures mean, variance, and covariance without running any optimization algorithm. This is a big advantage of linear regression. The disadvantage is that only linear functions (and non–parametric nonlinear functions) can be identified this way.

One way to build nonlinear models is the use of *polynomial function prototypes*. In a similar way as in (5) we define f as a polynomial of degree $r = 2, 3, \ldots$

$$y \approx f(x) = \sum_{m=0}^{r} a_m x^m \tag{16}$$

Inserting (16) into the square error functional (7), setting the derivatives to zero, and solving the resulting equation system yields the values of the optimal polynomial parameters $a_0, \ldots, a_r$. We don't want to go into the details of this method, but just illustrate the results by the two examples shown in Figure 1. The points ($\times$) in the left graph show the

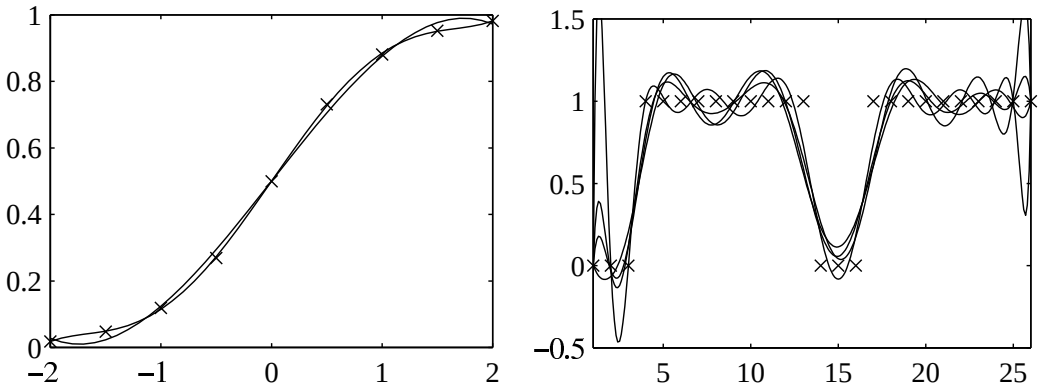


Figure 1: Polynomial fit of two data sets.

data set

$$X \times Y = \{(x, y) \mid x \in \{-2, -1.5, \ldots, 2\},\ y = \frac{1}{2}(1 + \tanh(x)) = \frac{e^x}{e^x + e^{-x}}\}. \quad (17)$$

The hyperbolic tangent is not a polynomial, but a very smooth function (in a mathematical, but also in a colloquial sense — notice the difference between colloquial and mathematical smoothness!). The two curves in the left view of Figure 1 show the polynomial approximation functions f with the degrees $r = 3$ and $r = 5$. Apparently the polynomials are good global models of $X \times Y$. The right view in Figure 1 shows the data set

$$X \times Y = \{(1, 0) \ldots (3, 0), (4, 1) \ldots (13, 1), (14, 0) \ldots (16, 0), (17, 1) \ldots (26, 1)\}. \quad (18)$$

These samples might belong to a (in the mathematical sense) smooth function, but it skips between different local structures ($y = 0$ and $y = 1$), and the number of samples has to be much higher than in the example before to represent the underlying function. This requires an approximation with higher polynomial degrees. The curves in the right view of Figure 1 show the polynomial approximations of $X \times Y$ for the degrees $r = 12, \ldots, 15$. Some points are matched quite well by these approximations, but all of them show high overshoots caused by forcing a global polynomial through various locally distinct models. Similar effects occur when the prototype functions for the regression are sinusoidal functions (the so–called harmonic regression). Unfortunately, systems with locally distinct behavior occur often in real world applications, so these regression methods are often inappropriate for non–linear system modeling.

3 Modeling with perceptrons

The regression approaches described in the previous section need a specification of function prototypes like linear functions or polynomials. If the function prototypes match the type of function underlying the data set, good models are obtained. If they don't match, undesirable effects like those illustrated in the right view of Figure 1 might occur leading to bad models. One popular way of modeling without function prototypes is modeling with *multilayer perceptrons*. A multilayer perceptron (MLP) [6, 11] is a directed graph like the one shown in Figure 2. The MLP nodes (called neurons) are usually arranged in

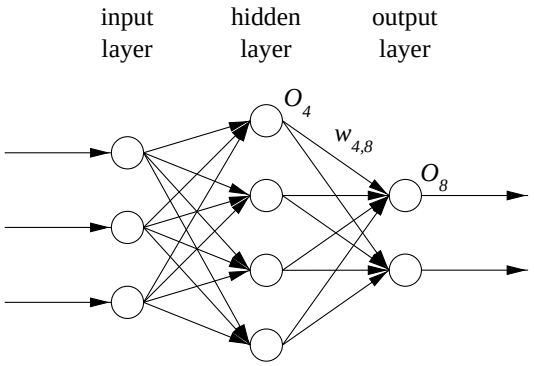


Figure 2: A multilayer perceptron.

layers like the input, hidden, and output layer shown in Figure 2. We number the neurons consecutively from left to right and from top to bottom. Each MLP edge, say from node number i to node number j, is assigned a weight $w_{ij} \in \mathbb{R}$. If we denote the neuron outputs as $O_j \in \mathbb{R}$, we define the effective neuron input as

$$I_j = \sum_i w_{ij} O_i, \tag{19}$$

and compute the neuron output as

$$O_j = f(I_j). \tag{20}$$

Notice that missing edges are represented in this scheme by zero edge weights. A common choice for the neuron transfer function f in (20) is the logistic function

$$f(x) = \frac{1}{1 + e^{-x}}. \tag{21}$$

The overall input output behavior of the MLP is completely determined by (19), (20) and (21) given the weight matrix $W = (w_{ij})$. The only free system parameters are the weights in W. Training an MLP means adjusting the parameters W so that each MLP input from X approximately leads to the corresponding output in Y (4). In the following we derive the update equations for W given X and Y. For simplicity assume only a single hidden layer, and denote $p, q, r \geq 1$ as the numbers of neurons in the input, hidden, and output layer. With this notation the MLP input is $x = (I_1, \ldots, I_p)$, and the output is $y = (O_{p+q+1}, \ldots, O_{p+q+r})$. In analogy to (5) and (16) we therefore require

$$y \approx f(x) = (O_{p+q+1}, \ldots, O_{p+q+r}) \mid_{x=(I_1,\ldots,I_p)} . \tag{22}$$

To minimize the mean square error functional (7) with f from (22) we can apply a gradient descent method: In each update step each weight w_{ij} is added the term

$$\Delta w_{ij} = -\alpha(t) \cdot \frac{\partial E}{\partial w_{ij}}, \tag{23}$$

where $\alpha : \mathbb{N} \to \mathbb{R}$ is a monotonously decreasing function specifying the learning rate of the neurons. The error gradient in (23) is computed from (19), (20) as

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial O_j} \cdot \frac{\partial O_j}{\partial I_j} \cdot \frac{\partial I_j}{\partial w_{ij}} \sim (O_j - x^{(j-p-q)}) \cdot f'(I_j) \cdot O_i \tag{24}$$

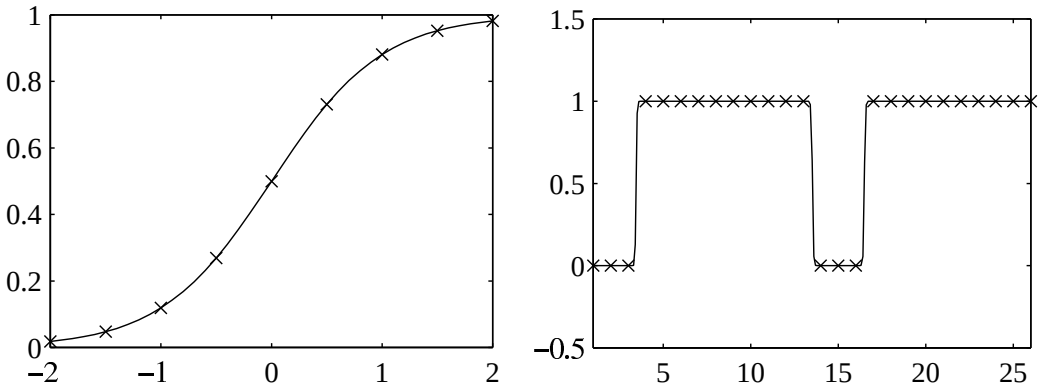


Figure 3: Modeling two data sets with multilayer perceptrons.

for the output weights ($i = p+1, \ldots, p+q$, $j = p+q+1, \ldots, p+q+r$), and

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}} &= \sum_{l=p+q+1}^{p+q+r} \frac{\partial E}{\partial O_l} \cdot \frac{\partial O_l}{\partial I_l} \cdot \frac{\partial I_l}{\partial O_j} \cdot \frac{\partial O_j}{\partial I_j} \cdot \frac{\partial I_j}{\partial w_{ij}} \\ &\sim \sum_{l=p+q+1}^{p+q+r} (O_l - x^{(l-p-q)}) \cdot f'(I_l) \cdot \; w_{jl} \; \cdot f'(I_j) \cdot \; O_i \end{aligned} \tag{25}$$

for the input weights ($i = 1, \ldots, p$, $j = p+1, \ldots, p+q$). Both update equations (24) and (25) can be summarized as the so–called *generalized delta rule*. We do not want to go into the details but mention that this rule allows to easily define weight update mechanisms in arbitrary perceptron architectures. We applied MLPs to model the two data sets from (17) and (18). The left view in Figure 3 shows the samples from (17) and the model obtained with an MLP with two neurons in the hidden layer. The right view in Figure 3 shows the samples from (18) and the model obtained with an MLP with four neurons in the hidden layer. For the second data set more hidden neurons were necessary, because it contains more local models. Both MLP models fit the data very well. The disadvantage of this approach, however, is that the MLP output is difficult to comprehend. For each input, the output is generated by interaction of all the neurons and all the net weights together. Therefore is difficult to make clear statements about the model behavior. Because of this disadvantage MLP models are also called *black box models*. Especially when system stability has to be examined, black box models are not acceptable.

4 Modeling with crisp singleton rules from clustering

Each component in regression models as well as in perceptron models represents a global function $f_i : \mathbb{R}^{p_i} \to \mathbb{R}^{q_i}$. In many applications, and also in the example data set from (18) a model structure might be better suited that uses local functions which are responsible only for certain sub–spaces of the input space. To approximate the samples (18), for

example, we might use the local functions

$$\begin{array}{lrcll} f_1: & [1,3] & \to & \mathbb{R}, & f_1(x)=0, \\ f_2: & [4,13] & \to & \mathbb{R}, & f_2(x)=1, \\ f_3: & [14,16] & \to & \mathbb{R}, & f_3(x)=0, \\ f_4: & [17,26] & \to & \mathbb{R}, & f_4(x)=1. \end{array} \tag{26}$$

Notice that the union of the local sub–spaces not necessarily equals the whole input space, $[1,3]\cup[4,13]\cup[14,16]\cup[17,26] \neq \mathbb{R}$. Inputs for which no valid (sub–)model exist might lead to undefined outputs, or the outputs might be generated by interpolation or extrapolation from valid model values for other inputs. The equations of the local functions in (26) can also be written as a *rule based system*

$$\begin{array}{lllll} R_1: & \text{If} & x \in [1,3] & \text{then} & y = f_1(x) = 0, \\ R_2: & \text{If} & x \in [4,13] & \text{then} & y = f_2(x) = 1, \\ R_3: & \text{If} & x \in [14,16] & \text{then} & y = f_3(x) = 0, \\ R_4: & \text{If} & x \in [17,26] & \text{then} & y = f_4(x) = 1. \end{array} \tag{27}$$

Notice that this rule based system approximates the data set (18) with zero error. Each rule premise $x \in C_i$, $i = 1, \ldots, 4$, can be written as the so–called *characteristic function* μ_i of the set C_i in the universe $\mathbb{R}$:

$$[\![x \in C_i]\!] = \mu_i(x) = \begin{cases} 1, & \text{if } x \in C_i \\ 0, & \text{if } x \notin C_i, \end{cases} \tag{28}$$

where the brackets $[\![\varepsilon]\!]$ denote the numerical truth value of an expression ε, where the numerical values 1 and 0 are used to represent the truth values *true* and *false*, respectively. With the characteristic function, each rule from (27) can be written in the form

$$R_i: \quad \text{If} \quad \mu_i(x) \quad \text{then} \quad y = c_i. \tag{29}$$

Now let's assume a rule base with characteristic functions chosen so that for any $x \in \mathbb{R}$ there is at least one rule that "fires", i.e. $\max\{\mu_i(x), i = 1, \ldots, c\} = 1$, where $c \in \mathbb{N}$ is the number of rules. If for a given input $x \in \mathbb{R}$ more than one rule fires, we require the output to be the average output of all firing rules. Generally, the output of such a rule base can then be computed as

$$y(x) = \frac{\sum_{i=1}^{c} \mu_i(x) \cdot f_i(x)}{\sum_{i=1}^{c} \mu_i(x)}. \tag{30}$$

Notice that this equation also includes the trivial case when only one rule fires, say rule number j, because then the output becomes $y(x) = f_j(x)$ as expected.

The local subspace of each rule R_i, $i = 1, \ldots, c$, (29) is specified by the set C_i. Instead, the system of local subspaces might be specified by the individual points $v_i \in \mathbb{R}^p$, $i, \ldots, c$, representing "centers" of the local subspaces. The rule R_i is then fired by an input x if $x \in C_i$, i.e. if

$$d_{ik} = \min_{j=1,\ldots,c} d_{jk}, \tag{31}$$

where d_{ik} is a distance measure

$$d_{ik} = \|x_k - v_i\|, \tag{32}$$

for example the Euclidean distance that is used thoughout this paper. The input x fires the rule whose center is the nearest neighbor to x (in case of multiple nearest neighbors all the corresponding rules are fired). Equation (31) is therefore also called the *nearest neighbor rule*. The corresponding characteristic function is

$$\mu_i(x) = \begin{cases} 1, & \text{if } d_{ik} = \min_{j=1,\dots,c} d_{jk}, \\ 0, & \text{otherwise.} \end{cases} \tag{33}$$

If we think of the centers $V = \{v_1, \dots, v_c\}$ as the centers of clusters in the input data set X, i.e. as regions containing a relatively high data density, we also call V the set of *cluster centers*. We can use the characteristic function (33) to set up a so–called *characteristic matrix* $U \in [0,1]^{c \times n}$ for X and V defined by $u_{ik} = \mu_i(x_k)$, $i = 1, \dots, c$, $k = 1, \dots, n$. To determine V and the charcteristic matrix U from the data X we can use the so–called *c–means* (CM) model [1] defined as the following problem: minimize the sum of squared distances between cluster centers and the respective data points

$$J_{CM}(U, V; X) = \sum_{i=1}^{c} \sum_{x_k \in C_i} d_{ik}^2 = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}\, d_{ik}^2. \tag{34}$$

From the necessary condition for local extrema of (34)

$$\frac{\partial J_{CM}(U, V; X)}{\partial v_i} = 0, \quad i = 1, \dots, c, \tag{35}$$

we obtain the cluster centers

$$v_i = \frac{1}{\|C_i\|} \sum_{x_k \in C_i} x_k = \frac{\sum_{k=1}^{n} u_{ik}\, x_k}{\sum_{k=1}^{n} u_{ik}}, \tag{36}$$

which is the first moment of C_i. Both the set V of cluster centers and the characteristic matrix U can be computed from a data set X by alternatingly computing $V(U, X)$ using (36) and $U(V, X)$ using (33). This scheme is also called *alternating optimization* (AO).

5 Modeling with fuzzy singleton rules from clustering

Rule bases like (27) perform crisp switches between local models, which typically lead to unsteady and unsmooth global characteristics. Instead, we typically desire a smooth interpolation between adjacent models. This can be implemented by fuzzifying the premise parts of the rules. To do this we formally extend the range of values of the characteristic function (28) from the crisp numerical truth values $\{0, 1\}$ to the unit interval $[0, 1]$,

$$[\![x \in C_i]\!] = \mu_i(x) \in [0, 1]. \tag{37}$$

We maintain the semantics of the crisp case and define $\mu_i(x) = 1$ if and only if x is *completely* included in C_i, and $\mu_i(x) = 0$ if and only if x is *not at all* included in C_i. The non–crisp truth values $\mu_i(x) \in (0, 1)$ are interpreted as the degree of partial inclusion of x in C_i. Fuzzy characteristic functions are also called *membership functions*, since they define degrees of membership in a (fuzzy) set [21]. Rules of the form (29) with (fuzzy) membership functions (37) are called *singleton* (or *Sugeno–Yasukawa*) rules [19]. The

term *singleton* rule stems from the fact that it maps a fuzzy set to a crisp set containing only a *single element* $\{c_i\}$. In the notion of fuzzy sets these single element sets can be written with the singleton membership function

$$\mu_{c_i}(x) = \begin{cases} 1, & \text{if } x = c_i, \\ 0, & \text{otherwise.} \end{cases} \tag{38}$$

In the more general case, fuzzy rules can be used to map fuzzy sets to fuzzy sets. Rules of this kind are called *Mamdani–Assilian* rules [8]. For a more detailed discussion of Mamdani–Assilian and other types of fuzzy rule based systems (and how to extract them by clustering) we refer to [7].

In the previous section we have shown how crisp singleton rules like (27) can be extracted from data using the c–means model (34). In a similar way, fuzzy singleton rules can be extracted from data with the *fuzzy c–means* model (FCM) [2] defined using the objective function

$$J_{FCM}(U, V; X) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^m \, d_{ik}^2, \tag{39}$$

where $m > 1$ is a fuzziness parameter, and the constraints

$$\sum_{k=1}^{n} u_{ik} > 0, \qquad i = 1, \ldots, c, \tag{40}$$

$$\sum_{i=1}^{c} u_{ik} = 1, \qquad k = 1, \ldots, n. \tag{41}$$

Condition (40) requires non–empty clusters, and condition (41) requires the sum of the memberships of each element in all clusters to be equal to one. If both conditions (40) and (41) hold, we talk abou a *fuzzy partition* and call the membership matrix a *(fuzzy) partition matrix*. For the optimization of J_{FCM} (39) with the constraint (41) we use the Lagrange function

$$F_{FCM}(U, V, \lambda; X) = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^m \, d_{ik}^2 - \lambda \cdot \left(\sum_{i=1}^{c} u_{ik} - 1 \right). \tag{42}$$

The necessary conditions for local extrema of F_{FCM} yield, after some conversions, the equations for the computation of the cluster centers V and the partition matrix U in an AO algorithm.

$$\left.\begin{array}{l} \dfrac{\partial F_{FCM}}{\partial \lambda} = 0 \\[2ex] \dfrac{\partial F_{FCM}}{\partial u_{ik}} = 0 \end{array}\right\} \Rightarrow \quad u_{ik} = 1 \Big/ \sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{m-1}}, \tag{43}$$

$$\frac{\partial J_{FCM}}{\partial v_i} = 0 \quad \Rightarrow \quad v_i = \frac{\sum\limits_{k=1}^{n} u_{ik}^m \, x_k}{\sum\limits_{k=1}^{n} u_{ik}^m}. \tag{44}$$

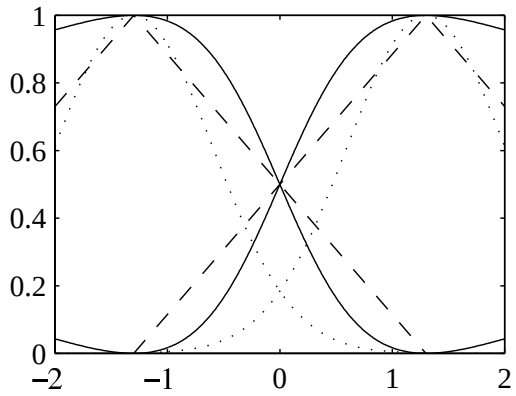


Figure 4: Membership functions from FCM–AO (solid), Gaussian membership functions (dotted), and triangular membership functions (dashed).

The FCM–AO membership function shapes $\mu(x)$ generated by extending equation (43) with (32) to $\mathbb{R}$,

$$\mu_i(x) = 1 \Bigg/ \sum_{j=1}^{c} \left(\frac{\|x - v_i\|}{\|x - v_j\|} \right)^{\frac{2}{m-1}} , \tag{45}$$

are inconvex and highly nonlinear. In fuzzy control systems, however, often triangular or trapezoidal membership functions are used; and *radial basis function* (RBF) networks [9, 10], which can be viewed as neural network implementations of fuzzy singleton rule bases that are trained by clustering algorithms, often use Gaussian membership functions

$$\mu_i(x) = e^{\|x - v_i\|^\alpha}, \quad \alpha > 0. \tag{46}$$

Figure 4 compares FCM–AO membership functions from (45) with $m = 2$ (solid), Gaussian membership functions from (46) with $\alpha = 2$ (dotted), and triangular membership functions (dashed). Clusters with any of these (and other) membership functions can be extracted from data using an extension of FCM–AO called *alternating cluster estimation* (ACE) [15].

We applied FCM–AO to the data sets (17) and (18). First, we used FCM–AO to compute V and U by (43) and (44) for the data sets $X \times Y$ (clustering in the product space). Then we computed the membership functions $\mu_i(x)$, $i = 1, \ldots, c$, by (45) for the *projections* $v_i^{(x)}$ of the cluster centers to the input space (input space projection). Finally, we computed the model outputs using (30) with the local models $f_i(x) = c_i = v_i^{(y)}$. For (17) with $m = 2$, $c = 2$ we obtained the solid membership functions from Figure 4. The resulting model is shown in the left view of Figure 5. The two circles ($\circ$) show the positions of the two cluster centers. With two local models a quite good approximation is achieved. For the data set (18) we did not achieve a good model with $c = 2$. Therefore, we increased the number of local models. The right view of Figure 5 shows the results for $c = 8$. Although the model output oscillates in the intervals $x \in [4, 13]$ and $[17, 26]$, we consider this a quite good model.

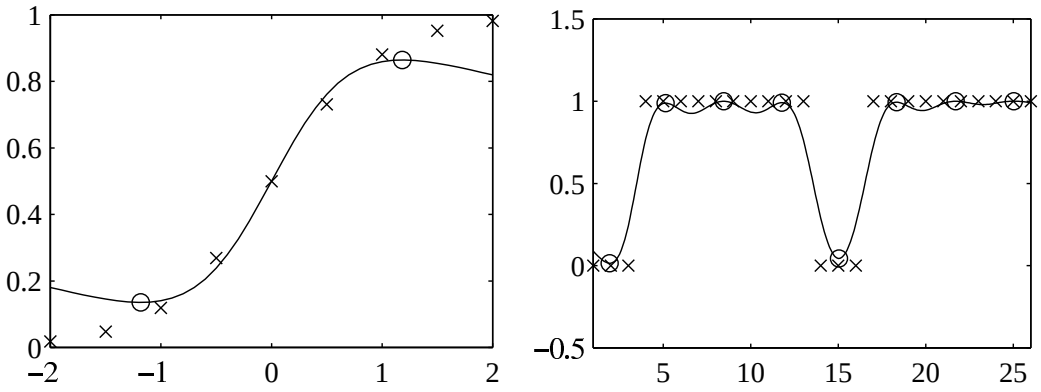


Figure 5: Modeling two data sets with Sugeno–Yasukawa systems generated with the fuzzy c–means model.

6 Takagi–Sugeno modeling with fuzzy c–elliptotypes clustering

In the previous section we only considered fuzzy rules with constant conclusions c_i such as (29). In (27), however, we had already suggested to use local model functions $f_i(x)$ in the rule conclusions. This leads to so–called *Takagi–Sugeno* (TS) rules [20]

$$R_i: \quad \text{If} \quad \mu_i(x) \quad \text{then} \quad y = f_i(x). \tag{47}$$

In control engineering often local linear models are used since for linear systems we can apply the superposition principle which allows closed–form system analysis and stability considerations. If all the functions $f_i(x)$ are *linear* functions (5) we talk about *first–order TS systems* [12]. To find local linear models in data, we have to extend the FCM model from spherical cluster prototypes specified by the centers V to r–dimensional, $r \in \{1, 2, \ldots, p+q-1\}$, linear cluster prototypes specified by cluster centers V and a tensor of direction vectors $S = (s_{ij} \mid i = 1, \ldots, c, j = 1, \ldots, r)$. The distance between cluster i and the data point x_k can then be computed as

$$d_{ik} = \sqrt{\|x_k - v_i\|^2 - \sum_{j=1}^{r} \left((x_k - v_i)^T s_{ij}\right)^2}. \tag{48}$$

If we insert this distance into (39) we obtain the objective function of the *fuzzy c–varieties* (FCV) model [3] that can be optimized in the same way as the FCM model. In particular, we can keep (43) and (44) for the computation of U and V, but we have to use (48) instead of (32) for the distances d_{ik}. The direction vectors are computed as the eigenvectors of the r largest eigenvalues of the within cluster fuzzy scatter matrices

$$B_i = \sum_{k=1}^{n} u_{ik}^m \, (x_k - v_i)(x_k - v_i)^T. \tag{49}$$

FCV clusters have infite extension. The α cuts of the i^{th} FCV cluster are hypercylinders with a common main axis going through v_i. To build local models, however, we require the clusters to be local structures, each concentrated around the cluster center v_i. This

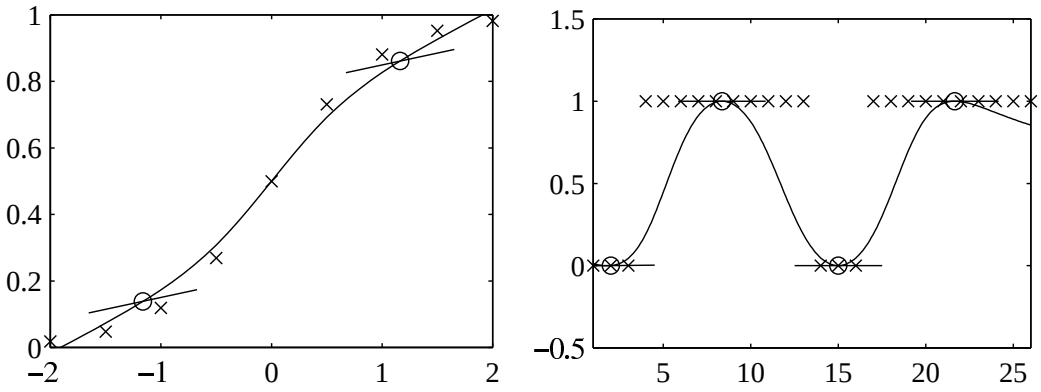


Figure 6: Modeling two data sets with Takagi–Sugeno systems generated with the fuzzy c–elliptotypes model.

property can be achieved with the distance measure for the so–called *fuzzy c–elliptotypes* (FCE) model [4]

$$d_{ik} = \sqrt{\|x_k - v_i\|^2 - a \cdot \sum_{j=1}^{r} ((x_k - v_i)^T s_{ij})^2}, \tag{50}$$

where $a \in [0, 1]$ is a locality parameter. For $a = 1$ we obtain the FCV distance, for $a = 0$ the FCM distance. For $0 < a < 1$ the α cuts of the i^{th} FCV cluster are ellipsoidals with the center v_i. For the sake of completeness we also mention the *Gustafson–Kessel* algorithm [5] that finds similar cluster shapes using local Mahalanobis distances.

We applied FCE–AO to the data sets (17) and (18). FCE–AO was used to compute V, U, and S by (43), (44), (49), and (50) for the data sets $X \times Y$ (product space). Then we computed the membership functions $\mu_i(x)$, $i = 1 \ldots, c$, from the projections $v_i^{(x)}$ using (45). The local model functions were computed as

$$f_i(x) = v_i^{(y)} + \frac{s_i^{(y)}}{s_i^{(x)}}(x - v_i^{(x)}), \quad i = 1, \ldots, c, \tag{51}$$

and summarized in a global model using (30). Figure 6 shows the resulting model for (17), $a = 0.99$, $r = 1$, $m = 2$, $c = 2$. The two cluster centers V are plotted as circles ($\circ$), and the two lines through the centers show the corresponding direction vectors S. With the local linear models a better model accuracy is achieved than with the FCM model (compare Figure 5 left). For the data set (18) FCE needs only $c = 4$ clusters instead of $c = 8$ for FCM (Figures 5 and 6 left). The four model centers ($\circ$) and the horizontal slope of the local models (lines through $\circ$) perfectly match the data.

7 Conclusions

We have presented various approaches for extracting system models from data sets. Among the global modeling methods, regression is a very simple and computationally efficient method. For complex systems that contain many local sub–structures, prototype functions

of a high order, e.g. higher order polynomials, are required that bring along undesirable properties like overshoots. Neural network approaches such as the multilayer perceptron are also considered as global modeling methods here, since their inherent local structures are not visible but packaged in a black box. Among all the modeling methods described in this paper the multilayer perceptron yielded the best model accuracy. Due to its black box behavior, however, it hardly seems acceptable for basic automation in control engineering problems.

Besides these global modeling approaches we presented several local modeling methods based on crisp or fuzzy rule based systems. Each rule in these rule based systems represents an individual local model that might be constant (singleton model), linear (first oder Takagi–Sugeno model), or of a higher order. To extract the local models from data we presented various clustering methods: The c–means model for crisp singleton systems, the fuzzy c–means model for fuzzy singleton systems, and the fuzzy c–elliptotypes model for first order Takagi–Sugeno systems. For the example data, satisfactory results could be obtained with singleton systems, but for complicated local structures many local models are necessary. First order Takagi–Sugeno systems yielded a higher accuracy than singleton systems and also provided local linear models that are useful for system analysis and stability considerations.

All the modeling methods described in this article were applied to numerous artificial as well as real–world problems. For brevity we just mention the author's work on modeling and optimization in the pulp and paper industry [16], in steel mills [17], in road traffic systems [18], and in medical systems [14].

More detailed information about data analysis methods and their industrial applications can be found in the book [13] (in German).

References

[1] G. H. Ball and D. J. Hall. Isodata, an iterative method of multivariate analysis and pattern classification. In *IFIPS Congress*, 1965.

[2] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[3] J. C. Bezdek, C. Coray, R. Gunderson, and J. Watson. Detection and characterization of cluster substructure, I. Linear structure: Fuzzy c–lines. *SIAM Journal on Applied Mathematics*, 40(2):339–357, April 1981.

[4] J. C. Bezdek, C. Coray, R. Gunderson, and J. Watson. Detection and characterization of cluster substructure, II. Fuzzy c–varieties and convex combinations thereof. *SIAM Journal on Applied Mathematics*, 40(2):358–372, April 1981.

[5] E. E. Gustafson and W. C. Kessel. Fuzzy clustering with a covariance matrix. In *IEEE Conference on Decision and Control, San Diego*, pages 761–766, 1979.

[6] R. Hecht-Nielsen. *Neurocomputing*. Addison-Wesley, 1990.

[7] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis — Methods for Image Recognition, Classification, and Data Analysis*. Wiley, 1999.

[8] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man–Machine Studies*, 7(1):1–13, 1975.

[9] M. J. D. Powell. Radial basis functions for multi–variable interpolation: a review. In *IMA Conference on Algorithms for Approximation of Functions and Data*, pages 143–167, Shrivenham, 1985.

[10] M. J. D. Powell and A. Iserles. *Approximation Theory and Optimization*. Cambridge University Press, 1998.

[11] H. J. Ritter, T. M. Martinetz, and K. J. Schulten. *Neuronale Netze*. Addison–Wesley, München, 1991.

[12] T. A. Runkler. Automatic generation of first order Takagi–Sugeno systems using fuzzy c–elliptotypes clustering. *Journal of Intelligent and Fuzzy Systems*, 6(4):435–445, 1998.

[13] T. A. Runkler. *Information Mining — Methoden, Algorithmen und Anwendungen intelligenter Datenanalyse*. Computational Intelligence. Vieweg, Wiesbaden, 2000.

[14] T. A. Runkler and J. C. Bezdek. Image segmentation using fuzzy clustering with fractal features. In *IEEE International Conference on Fuzzy Systems*, volume 3, pages 1393–1398, Barcelona, July 1997.

[15] T. A. Runkler and J. C. Bezdek. Alternating cluster estimation: A new tool for clustering and function approximation. *IEEE Transactions on Fuzzy Systems*, 7(4):377–393, August 1999.

[16] T. A. Runkler, E. Gerstorfer, M. Schlang, E. Jünnemann, and K. Villforth. Data compression and soft sensors in the pulp and paper industry. In *European Control Conference '99, Karlsruhe*, volume CM–2, pages 1–5, August 1999.

[17] M. Schlang, B. Feldkeller, B. Lang, T. Poppe, and T. Runkler. Neural computation in steel industry. In *European Control Conference '99, Karlsruhe*, volume BP–1, pages 1–6, August 1999.

[18] C. Stutz and T. A. Runkler. Fuzzy c–mixed prototype clustering. In W. Brauer, editor, *Fuzzy–Neuro–Systems '98, München*, volume 7 of *Proceedings in Artificial Intelligence*, pages 122–129, March 1998.

[19] M. Sugeno and T. Yasukawa. A fuzzy–logic–based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems*, 1(1):7–31, February 1993.

[20] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1):116–132, 1985.

[21] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

Merkmalgenerierung und Klassifikation

Christian Kuhn
Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
Institut für Automatisierungs- und Systemtechnik
D-98684 Ilmenau, PF: 100565
Tel.: 03677 691467, Fax: 03677 691434
email: christian.kuhn@rz.tu-ilmenau.de

1 Einführung

Methoden der automatischen Klassifikation haben sich in vielen Bereichen der Technik etabliert. Die Anwendungsgebiete erstrecken sich u. a. auf die Sprach- und Bildverarbeitung, auch in der Mustererkennung werden Klassifikationsalgorithmen benötigt. In der Automatisierungstechnik liegen Anwendungsgebiete vor allem bei der Fehlerdiagnose, dem Prozeßmonitoring, aber auch bei Prognoseproblemen. Klassifikatoren werden auch in Klassensteuerungen eingesetzt, um nichtlineare Prozesse durch approximierte linearisierte Teilprozesse zu beschreiben und für diese linearisierten Teilprozesse entsprechende Steuerungskonzepte auszuwählen [1]. Dem Klassifikator obliegt hier die Aufgabe der Prozeßsituationserkennung. In vielen Fällen sind traditionell drei Teilprobleme zu lösen (Abbildung 1), vgl. [2].

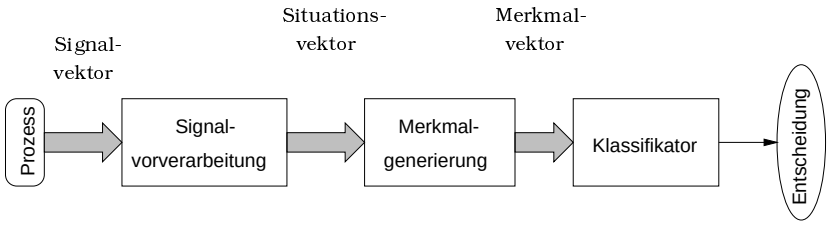


Abbildung 1: Komponenten eines Klassifikationssystems

Signalvorverarbeitung. Die Signalvorverarbeitung dient der Vorbereitung der Daten für die weitere numerische Verarbeitung in einem Rechnersystem. Sie ist Teil des Klassifikationssystems. Zur Signalvorverarbeitung zählt u. a. die A/D-Wandlung der Meßgrößen, oder aber, speziell für das Training eines Klassifikators, das Einlesen und die Aufbereitung archivierter Daten. Hinzu kommt die Aufgabe, Meßwertausfälle geeignet zu markieren und die Daten für die weitere Auswertung einer zweckmäßigen Normierung bzw. Skalierung zu unterziehen. Aus der Signalvorverarbeitung resultiert der Situationsvektor, welcher Ausgangsbasis für die nachfolgende Merkmalgenerierung ist.

Die Signalvorverarbeitung hat einen bedeutenden Anteil an der Klassifikationsgüte, der oftmals unterschätzt wird. Ihr obliegt die Aufgabe, den zumeist kontinuierlichen Prozeß der diskreten Weiterverarbeitung zu erschließen. Dabei können durch eine ungeeignete Quantisierung bzw. Diskretisierung (SHANNONsches Abtasttheorem) der analogen Prozeßsignale wichtige Informationen verloren gehen.

Merkmalgenerierung. Die Merkmalgenerierung schließt sich an die Signalvorverarbeitung an. Die Merkmalgenerierung umfaßt zwei wesentliche Komplexe:

- Merkmaltransformation
- Merkmalextraktion

Während durch die Merkmaltransformation der Vektor mit den abgetasteten Meßwerten einer aktuellen Situation in einen Merkmalvektor überführt und damit in einen geeigneten Merkmalraum transformiert wird, dient die Merkmalextraktion der Untersuchung der Merkmalräume und der Selektion geeigneter Merkmale. Dabei hängt die Güte eines Klassifikators und seine Struktur vor allem von der Merkmalauswahl ab. Durch die Merkmalgenerierung werden klassentrennende Eigenschaften der gemessenen Signale hervorgehoben und für die Klassifikation geeignet vorbereitet.

Klassifikation. Der Klassifikator übernimmt die Zuordnung eines Objektes bzw. einer aktuellen Prozeßsituation zu einer Klasse. Die Klassifikation erfolgt auf der Grundlage der bereitgestellten Merkmale. Eine hohe Güte des Klassifikators setzt deshalb eine sorgfältige Merkmalauswahl voraus.

2 Methoden der Merkmaltransformation

In [3] wurden vorhandene Methoden der Merkmaltransformation strukturiert und zu drei wesentlichen Gruppen zusammengefaßt. Die Methoden wurden unter dem Aspekt der Fehlerdiagnose gruppiert, sie besitzen jedoch für die Klassifikation eine übergreifende Bedeutung. Durch sie werden gemessene Signale in geeignete Merkmalräume transformiert, wobei die Merkmale durch die Transformationen bestimmt werden.

Die *signalgestützten* Merkmaltransformationen umfassen die Gesamtheit der Signalanalysemethoden. Der Vorteil dieser Analysemethoden ist ihre weite Verbreitung, die sich auf universelle Einsatzmöglichkeiten stützt. Gewisse Nachteile müssen beim Einsatz zur Fehlerdiagnose in Kauf genommen werden, die meist bei der Fehlerfrüherkennung oder bei der Erkennung von Fehlern, die sich auf die Prozeßdynamik auswirken, spürbar werden. Für die Untersuchungsmethoden stehen der Amplitudenbereich, Zeitbereich, Spektralbereich und der Cepstralbereich zur Verfügung. Die Zielbereiche hängen von der Aufgabenstellung und der Art der vorliegenden Signale ab. Darüberhinaus gibt es auch neuere Analysemethoden, die nicht zu den traditionellen Signalanalysemethoden gezählt werden. Beispielsweise wird mitunter zur Klassifikation von Schlafphasen die *fraktale Dimension* des EEG-Signals eines Patienten benutzt [4], [5]. Damit erfolgt eine drastische Informationskompression. Durch eine Maßzahl läßt sich damit eine Aussage zum stochastischen Verlauf einer Zeitreihe treffen.

Leistungsfähiger als die signalgestützen Merkmaltransformationen sind *modellgestützte* Transformationen. Vom Prozeß wird ein zumeist analytisches Modell gebildet, das den Prozeß bei einer bestimmten Situation nachbildet. Weichen nun die nominalen Werte des Modells von den Prozeßmeßwerten ab, so ist dies ein Hinweis für das Vorliegen einer anderen Prozeßsituation. Die modellgestützte Transformation ist sehr leistungsfähig, da auch Fehlerfrüherkennung und die Detektion dynamischer Fehler möglich ist. Jedoch ist die Bildung eines analytischen Modells oftmals sehr schwierig, und es ist meist nicht universell einsetzbar. Oftmals ist es an den zu überwachenden Prozeß gebunden. Zu Vertretern dieser Verfahren zählen Signalmodelle, Ausgangsbeobachter, Fehlerdetektionsfilter und Beobachterbänke. Nicht separat erwähnt, jedoch wichtig erscheinen auch Modelle, die eine Aussage über den *Zustand* des Prozesses liefern. Die hier gewonnenen Informationen lassen Untersuchungen zur *Stabilität* einer Prozeßsituation, dem dynamischen Verhalten des Prozesses bzw. dessen Prognose zu.

Wissensgestützte Transformationen kommen dem Anwender bei schwierigen und komplexen Prozessen entgegen, bei denen ein analytisches Modell nicht oder nur mit unverhältnismäßig hohem Aufwand erstellt werden kann. Dies können beispielsweie *qualitiative Prozeßmodelle* [6] oder auch *neuronale Netze* sein. Mitunter erschweren ein hoher Trainingsaufwand und eine schwierige Interpretation die Anwendung wissensgestützter Modelle.

Es ist zweckmäßig, den Vorrat an Transformationsalgorithmen nach bestimmten Kriterien zu klassifizieren, um den Zugriff zu erleichtern. Mit Hilfe derartiger Klassifizierungsmerkmale läßt sich aus dem Pool aus Merkmaltransformationen eine strukturierte Wissensbasis erstellen, wobei die Klassifizierungsmerkmale als Kriterien zur Strukturierung der Wissensbasis herangezogen werden. Die Ausprägungen der Strukturierungskriterien stellen einen linguistischen Wertevorrat dar, der sich zur gezielten Suche nach Transformationen verwenden läßt. Es ergeben sich zwei Zugriffsarten auf das Wissen:

- die lexikalische Suche über den Namen der Transformation
- die Indizierung (Suche bestimmter Transformationen) aufgrund vorgegebener linguistischer Attribute der Klassifizierungsmerkmale (Suchkriterien).

Während die lexikalische Suche eine eindeutige Zugriffsart darstellt, da ein Name eindeutig mit einer Routine verknüpft ist, ist die Indizierung über die linguistischen Attribute eine Suche, die mehrdeutig ausfallen kann. Man erhält eine Ergebnismenge mit Routinen, die vorgegebene Kriterien erfüllen. Falls keiner der verfügbaren Algorithmen den Suchkriterien entspricht, kann die Ergebnismenge auch leer sein.

3 Merkmalextraktion

Die Merkmalextraktion zielt auf die Suche nach signifikanten Objekthäufungen ab und dient der Untersuchung und Bewertung der trennenden Eigenschaften der Merkmale. Die Objekthäufungen müssen — um eine Trennung zu ermöglichen — charakteristisch für eine Klasse sein. Sie unterscheiden sich damit durch ihre Lage im Merkmalraum.

Für die Suche von Objekthäufungen gibt es eine ganze Reihe populärer Clusteranalyseverfahren [7], denen die Berechnung der Distanzen zwischen den transformierten Meßwerten

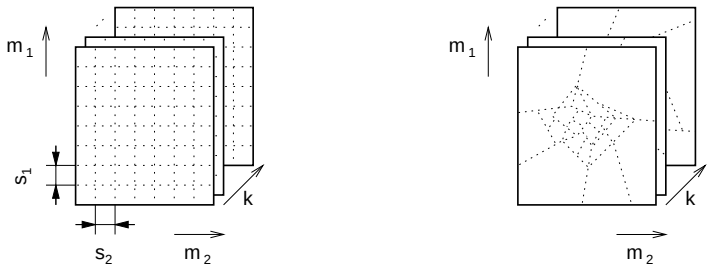
— im folgenden auch als *Objekte* bezeichnet — zugrunde gelegt wird. Voraussetzung für Diskriminanzuntersuchungen sind oftmals normalverteilte Datensätze [8]. In vielen Fällen sind die Prozesse jedoch nichtlinear, so daß hier die Annahme von normalverteilten Daten eine fehlerbehaftete Approximation darstellen kann.

3.1 Suche nach signifikanten Objekthäufungen

Die hier vorgestellte Untersuchungsmethode fußt deshalb auf der Dichteanalyse des Merkmalraumes. Voraussetzung ist eine klassifizierte Lernstichprobe, die zuvor durch eine gegebene Transformation (bzw. durch Merkmalkombination) in einen Merkmalraum transformiert wurde.

3.1.1 Unterteilung des Merkmalraumes in Subräume

Der gesamte Merkmalraum $\mathcal{M}$ wird in Subräume $\mathcal{R}$ unterteilt. Jeder Subraum umfaßt Objekte mit genügend ähnlichen Eigenschaften, wobei deren Ähnlichkeit durch einen entsprechend der Merkmalanzahl n-dimensionalen Ausdehnungsvektor $\boldsymbol{s}$ beeinflußt wird. Durch die Aufteilung des Merkmalraumes in Subräume entsteht eine *Map*. Um die Objektdichten der verschiedenen Klassen an den unterschiedlichen Positionen untersuchen zu können, muß für jede Klasse eine separate Map angelegt werden, es werden bei K Klassen also K Maps benötigt. Eine Map spiegelt damit die Dichte im Merkmalraum einer Klasse wider. Wir wollen hier zur Vereinfachung der Berechnungsalgorithmen eine äquidistante Unterteilung des Merkmalraumes annehmen, so daß alle Subräume die gleiche Ausdehnung besitzen, vgl. Abbildung 2(a). Die Nutzung der Dichte zur Auswertung erlaubt aber prinzipiell auch unregelmäßige Konstellationen, wie in Abbildung 2(b) zu sehen ist. Damit ist eine Subraumaufteilung entsprechend den Dichteverhältnissen im Merkmalraum möglich. Dies kann sinnvoll sein, wenn Maßnahmen zur Minimierung der Granulation in schwach besetzten Subräumen getroffen werden sollen oder der Dichtegradient innerhalb eines Subraumes vernachlässigbar klein ist.



(a) Äquidistante Raumaufteilung

(b) Unregelmäßige Raumaufteilung

Abbildung 2: Aufteilung des Merkmalraumes in klassenabhängige Maps aus Subräumen

3.1.2 Berechnung der Objektdichten

Für die Berechnung der klassenbezogenen *Objektdichten* ist die Kenntnis der klassenbezogenen relativen Objekthäufigkeiten erforderlich, die mit Hilfe der klassifizierten Lernprobe ermittelt werden können. Zur Vereinfachung der Berechnung wird angenommen, daß die Objekte innerhalb eines Subraumes gleichverteilt vorliegen. Über die relativen Häufigkeiten

$$H_r = \frac{N_{\mathcal{R}}}{N_k} \tag{1}$$

mit $N_{\mathcal{R}}$ als der (klassenbezogenen) Objektanzahl im Subraum und N_k als der Anzahl der Objekte einer Klasse k bei gleichen Klassenwahrscheinlichkeiten läßt sich die Dichte r der Objekte einer Klasse k im Subraum $\mathcal{R}$ berechnen

$$r = \frac{H_r}{V}. \tag{2}$$

Als *Hypervolumen* V gilt der Rauminhalt des Subraumes unabhängig von seiner Dimension. Das Hypervolumen für einen Hyperquader mit den Abmessungen $\boldsymbol{s}$ beträgt

$$V = \prod_{i=1}^{n} s_i. \tag{3}$$

Aus der Tatsache

$$\sum_{\mathcal{R}\in\mathcal{M}} H_r^{\mathcal{R}} = 1 \tag{4}$$

für gleiche Klassenwahrscheinlichkeiten folgt für die Objektdichte, daß deren Summe über den gesamten Merkmalraum multipliziert mit dem Volumen eines Subraumes gleich eins ist

$$\sum_{\mathcal{R}\in\mathcal{M}} r^{\mathcal{R}} \cdot V^{\mathcal{R}} = 1. \tag{5}$$

Für einen unendlich großen Datensatz konvergiert die relative Häufigkeit gegen die Wahrscheinlichkeit [9]

$$\lim_{N\to\infty} \frac{N_k}{N} = P_k, \tag{6}$$

so daß unter dieser Voraussetzung und infinitesimal kleinen Subraumabmessungen s_i die Dichte r gegen die Wahrscheinlichkeitsdichte p konvergiert.

3.1.3 Das Problem der Granularität

Ein Effekt, der in der diskreten Repräsentation der Objekte und dem endlichen Umfang der Lernstichprobe zu suchen ist, ist die *Granularität*. Sie macht sich durch ein überlagertes Rauschen bzw. eine Rauhigkeit der Dichten benachbarter Subräume bemerkbar. Sie wird um so größer, je kleiner der Stichprobenumfang und je größer die Dimension des Merkmalraumes ist. Sie entspricht den in [8] diskutierten Instabilitäten multivariater Verfahren. Eine starke Granularität kann den Eindruck einer nicht vorhandenen Heterogenität im Merkmalraum erwecken. Abhilfe kann in einer Vergrößerung der Subraumabmessungen gefunden werden,

allerdings stellt diese Möglichkeit nur einen Kompromiß dar, der auf Kosten der Auflösung vorgenommen werden muß. Auch können die Subräume entsprechend der Dichtegradienten aufgeteilt werden. Gebiete mit konstanten Dichten lassen sich zu gemeinsamen Subräumen zusammenfassen, siehe auch Abbildung 2(b). Eine wirksamere Abhilfe bieten jedoch veränderte Berechnungsgrundlagen zur Ermittlung der Objektdichte.

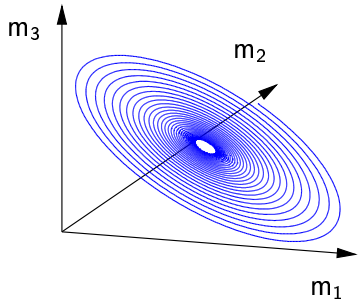


Abbildung 3: Trajektorie eines Prozeßobjektes eines abklingenden Schwingungsvorganges im dreidimensionalen Merkmalraum

Die Annahme diskreter Objekte läßt sich auf den „Stroboskopeffekt“ während des Abtastvorganges der Signalvorverarbeitung zurückführen. Kontinuierliche Signalverläufe werden durch die Abtastung „zerhackt“, Informationen über den Prozeßverlauf stehen danach nur noch zu diskreten Zeiten ΔT zur Verfügung, die als diskrete Objekte im Merkmalraum beobachtet werden können. Ein kontinuierlicher Prozeß hat jedoch — eine geeignete Merkmaltransformation vorausgesetzt — ein sich im Merkmalraum bewegendes Prozeßobjekt zur Folge, dessen Bahn eine Trajektorie der Form

$$m_1 = f_1(t) \tag{7}$$
$$m_2 = f_2(t) \tag{8}$$
$$\vdots$$
$$m_n = f_n(t) \tag{9}$$

beschreibt, vgl. Abbildung 3.

Eine Möglichkeit zur Glättung der Dichten besteht in der Berechnung der Bogenlänge l des Trajektorienabschnitts zwischen den Abtastpunkten ΔT und deren Subraumanteilen $l_{\mathcal{R}}$. Zur Berechnung glatter Kurvensegmente werden Splines bzw. adaptive Kurzzeitmodelle vorgeschlagen, die Berechnung der Kurvenlängen in den einzelnen Subräumen stellt allerdings gerade in höherdimensionalen Merkmalräumen ein schwieriges Unterfangen dar (Abbildung 4).

Ein weiterer Ansatz für die Berechnung eines glatten Dichteverlaufs ergibt sich durch die Interpretation der Klassenschwerpunkte als *Attraktoren*, die eine gewisse Anziehungskraft auf die Objekte ausüben.[1] Attraktoren können Stabilitätszentren eines Prozesses sein.

[1]In einem Gedankenexperiment entsprechen die Attraktoren Magnetpolen, über die ein Blatt Papier gelegt wurde. Die Attraktoren werden vom Papier verdeckt und sind nun nicht mehr sichtbar. Streut man jedoch Eisenspäne auf das Papier — hier die *Beobachtungen* verschiedener Meßversuchsreihen — verraten sich die Attraktoren durch eine Häufung der Späne an den Stellen, unter denen sich die Magnetpole befinden. Obwohl die Eisenspäne wie auch die Beobachtungen diskreten Charakter haben, durchzieht das Feld den Merkmalraum bzw. bestimmte Gebiete *kontinuierlich*.

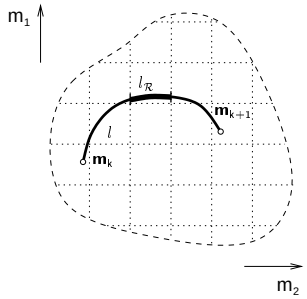


Abbildung 4: Aufteilung der Trajektorie in Kurvensegmente

Bei Anwesenheit von Attraktoren breitet sich ein kontinuierliches *Attraktorfeld* im Merkmalraum aus. Ein erster Ansatz, dieses Paradigma für die Informationsverarbeitung nutzbar zu machen, stellt das in [10] so bezeichnete *Informationskontinuum* dar. Die Berechnung des Feldverlaufs ist aufwendig, Möglichkeiten eröffnen sich prinzipiell durch Fuzzy-Clustering, finite Elementemethoden und Interpolation des Feldverlaufs zwischen Stützstellen entsprechend dem Subraummodell. Falls sich Funktionen aufstellen lassen, die den Dichteverlauf exakt nachbilden, kann auf die Bildung von Subräumen verzichtet werden. Subräume sind jedoch vorteilhaft bei unregelmäßigen Dichteverläufen und nichtkonvexen Klassengebieten einsetzbar.

3.1.4 Untersuchung des Merkmalraumes

Eine Randbedingung der Merkmalgewinnung ist, daß die Dichte im Merkmalraum einen gewissen Schwellwert überschreiten muß. Auf diese Weise lassen sich Ausreißer und kurzzeitige Meßausfälle eliminieren. Bei einer stetigen Dichteverteilung werden damit nicht alle Gebiete im Merkmalraum zu Klassengebieten deklariert, sondern schwach besetzte Klassengebiete aus den nachfolgenden Betrachtungen ausgenommen. Im zweidimensionalen Merkmalraum sind diese Kerngebiete als Inseln sichtbar, die ein durch den Schwellwert $r_{\min}$ gegebene Mindestniveau überschreiten, vgl. Abbildung 5. Die Vorgabe eines Mindestniveaus hat auf diese Weise eine Maskierung des Merkmalraumes zur Folge.

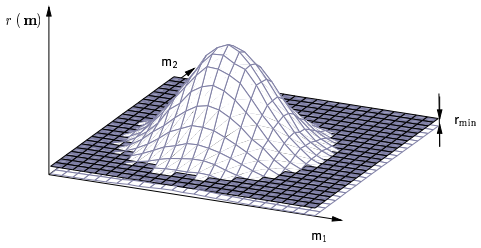


Abbildung 5: Ein Klassengebiet als Insel im zweidimensionalen Merkmalraum

Für die erfolgreiche Klassentrennung müssen sich die Dichten verschiedener Klassen in den Häufungsgebieten signifikant unterscheiden. Dazu muß sich eine *Heterogenität* im Merkmalraum feststellen lassen. Liegen in den Kerngebieten die Dichten mehrerer Klassen über einem Schwellwert, so *durchdringen* sich die Klassengebiete. Diese Bereiche sind für die Klassifikation ungeeignet, vgl. Abbildung 6. Die Heterogenität wird aus der Differenz der an

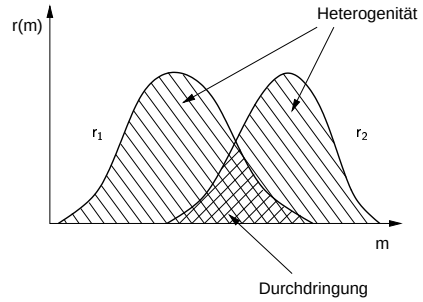


Abbildung 6: Heterogenität und Durchdringung im Merkmalraum

einer Position $\boldsymbol{m}$ dominierenden Dichte r_k zur nächstkleineren Dichte r_l gebildet, wobei für die beiden Dichten gilt

$$r_k = \max_{i \in \{1 \ldots K\}} r_i \tag{10}$$

$$r_l = \max_{j \in \{1 \ldots K\} \setminus k} r_j. \tag{11}$$

Die Heterogenität an einer Position des Merkmalraumes berechnet sich nach

$$h\left(\boldsymbol{m}\right) = r_k\left(\boldsymbol{m}\right) - r_l\left(\boldsymbol{m}\right), \tag{12}$$

aus der sich mit Gleichung 5 die Gesamtheterogenität für das Modell äquidistanter Subräume nach

$$h = \frac{V^{\mathcal{R}}}{K} \sum_{\mathcal{R} \in \mathcal{M}} r_k^{\mathcal{R}} - r_l^{\mathcal{R}} \tag{13}$$

ableiten läßt. Über die Heterogenität läßt sich eine Aussage über die Eignung des gesamten Merkmalraumes für die Merkmalextraktion treffen; dies erlaubt Rückschlüsse über die Eignung der Transformation zur Merkmalgewinnung. Die *Durchdringung* d wird aus den Dichten gebildet, die nicht zu Heterogenität beitragen. Für die Durchdringung im Merkmalraum gilt damit

$$d = \frac{V^{\mathcal{R}}}{K} \sum_{\mathcal{R} \in \mathcal{M}} \left(r_l^{\mathcal{R}} - r_k^{\mathcal{R}} + \sum_{i \in \{1 \ldots K\}} r_i^{\mathcal{R}} \right). \tag{14}$$

Für beide Größen gilt die Beziehung

$$h + d = 1. \tag{15}$$

Für einen ideal glatten Dichteverlauf ohne Granularitätserscheinungen gilt, daß alle Klassen vollständig trennbar sind, falls die Heterogenität des betrachteten Merkmalraumes gleich (oder im praktischen Fall: nahe) eins ist.

Zur Bildung einer Entscheidungsregel eignet sich die Berechnung der lokalen Heterogenität weniger, da sie von den Dichten abhängig ist. Deshalb wird eine Größe vorgeschlagen, die die Ausrichtung bzw. Polarisation einer Stelle des Merkmalraumes zu einer dominierenden Klasse angibt und unabhängig von den vorherrschenden Objektdichten ist. Diese Größe wird als *Polarisation* λ bezeichnet. Sie berechnet sich nach

$$\lambda = \frac{r_k - r_l}{r_k} \quad \text{mit } r_k > 0. \tag{16}$$

Sie kann nur für die Gebiete im Merkmalraum ermittelt werden, die Dichten größer als null aufweisen. Die Polarisation gibt die *Sicherheit* einer Entscheidung an und ist für die maximale Sicherheit gleich eins.

4 Klassifikation im Merkmalraum

4.1 Ableitung einer Entscheidungsregel

Um eine Situation in einem Merkmalraum zu klassifizieren, müssen drei Bedingungen erfüllt sein. Die Hauptbedingung für die Zuordnung einer Situation zur Klasse k ist Gleichung 10, d. h. die mit der Klasse k korrespondierende Dichte r_k muß die dominierende Dichte im Vergleich zu den Dichten der anderen Klassen sein. Neben dieser Hauptbedingung lassen sich Nebenbedingungen formulieren, mit denen das Entscheidungsverhalten an die Dichteverhältnisse angepaßt und Anforderungen an die Entscheidungssicherheit gestellt werden. Die Maskierung des Merkmalraumes wird durch

$$r \geq r_{\min} \tag{17}$$

erreicht. Ist die Sicherheit der Entscheidung von Bedeutung, läßt sich eine Sicherheitsschwelle vorgeben

$$\lambda \geq \lambda_{\min}. \tag{18}$$

Eine Entscheidung zugunsten einer Klasse wird in diesem Fall nur getroffen, wenn die Polarisation an der zu untersuchenden Stelle des Merkmalraumes größer oder gleich einem vorgegebenen Mindestwert ist. Ist eine Bedingung nicht erfüllt, erfolgt die *Rückweisung* des zu klassifizierenden Merkmalvektors. Die Entscheidungsregel bekommt damit die folgende Gestalt

$$\boldsymbol{e} = \begin{cases} [k, \lambda] & \text{falls} \quad r_k = \max_{i \in \{1 \ldots K\}} r_i \ \wedge \ r_k \geq r_{\min} \ \wedge \ \lambda \geq \lambda_{\min} \\ [0, 1-\lambda] & \text{sonst; } \wedge \ \lambda \text{ definiert} \\ [0, 1] & \text{sonst; } \wedge \ \lambda \text{ nicht definiert} \end{cases} . \tag{19}$$

Über die Polarisation λ läßt sich die Sicherheit einer Entscheidung beurteilen und damit eine Entscheidung *wichten*. Diese Bewertung kann für die Ermittlung einer Gesamtentscheidung von Bedeutung sein, wenn eine Vielzahl von Einzelentscheidungen (z. B. bei der Klassifikation von Situations- bzw. Objekt*arrays* wie Zeit- oder Frequenzreihen) zu einer Gesamtentscheidung aggregiert werden soll. Man erhält somit anstelle einer skalaren Größe einen Entscheidungsvektor $\boldsymbol{e}$, der neben dem Klassenindex k als erstem Eintrag auch die Sicherheit der getroffenen Entscheidung enthält.

Für die Rückweisung in Gleichung 19 ist eine Fallunterscheidung denkbar, die eine Behandlung der konkreten Situation (leerer Merkmalraum / unsichere Bereiche) ermöglicht.

4.2 Klassifikation des Merkmalraumes

Nach der Klassifizierung *aller* Subräume des Merkmalraumes ist die Suche nach zusammenhängenden Klassengebieten möglich. Dabei wird die in Abschnitt 4.1 vorgestellte Entscheidungsregel benutzt. Die Suche entspricht einer Clusteranalyse im Merkmalraum, in die alle Subräume einbezogen werden. Über die Vorgabe einer Distanz kann entschieden werden, bis zu welchem Abstand Subräume noch zum Cluster zugehörig erkannt werden oder neue Cluster bilden. Bei einem äquidistanten Subraumgitter nach Abbildung 2(a) läßt sich ein rekursiver Suchalgorithmus implementieren, der die regelmäßigen Nachbarschaftsbeziehungen der Subräume vorteilhaft ausnutzt.

Mit einer Subraumclusterung lassen sich nicht nur Klassengebiete aus dem Merkmalraum bei Nutzung einer klassifizierten Lernprobe extrahieren; es kann auch eine Prästrukturierung einer unklassifizierten Lernstichprobe vorgenommen werden, da über die Vorgabe einer Untersuchungsdistanz alle Subräume einem neuen Cluster zugeordnet werden, falls der Abstand zwischen ihnen eine Distanz überschreitet.

4.3 Bemerkungen zur Komplexität

Werden Maps mit äquidistanten Subraumanordnungen untersucht, ergeben sich neben den Problemen der Granularität auch Schwierigkeiten durch eine von der Dimension des Merkmalraumes abhängige Komplexität. Für k Klassen, n Merkmale und $\boldsymbol{v}$ äquidistante Unterteilungen des Merkmalraumes in den entsprechenden Merkmalrichtungen läßt sich eine (Speicher-)Komplexität O wie folgt formulieren

$$O = k \cdot \prod_{i=1}^{n} v_i. \tag{20}$$

Gleichung 20 verdeutlicht, daß man bei höheren Dimensionen sehr schnell auf Grenzen bei der Durchführung der Berechnungsalgorithmen stoßen wird. Eine Reduzierung der Komplexität erreicht man — neben einer ggf. durchgeführten Neuorganisation der Subraumaufteilung — durch *Projektionen* des hochdimensionalen Merkmalraumes auf Merkmalräume kleinerer Dimension. Dies entspricht einer Eliminierung von Merkmalen aus dem Merkmalvektor und ist mit einer Reduzierung der Vektorlänge verbunden. Allerdings besteht die Gefahr eines mit der Projektion verbundenen *Informationsverlustes*, da es zu Durchdringungen von Klassengebieten kommen kann, die dann nicht mehr für die Klassifikation geeignet sind, vgl. Abbildung 7. Eine Analyse des Merkmalraumes kann auch durch die Hinzunahme von Merkmalen mit fehlendem Informationsgehalt erschwert werden. In der Literatur sind Verfahren zur Extraktion von relevanten Merkmalen als *Variablenselektionsverfahren* bekannt [8]. In Anlehnung an das dort erwähnte Aufbauverfahren wird ein Verfahren vorgeschlagen, das mit einer Untersuchung niedrigdimensionaler Merkmalräume beginnt und durch Kombination der Merkmale untereinander die Komplexität der untersuchten Merkmalräume steigert. Die anschließende Bewertung der Merkmalräume gibt Auskunft über die Eignung der Merkmalkombination für die Klassifikation. In Algorithmus 1 wird die Vorgehensweise verdeutlicht.

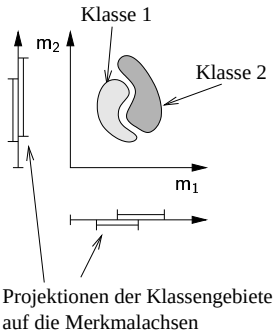


Abbildung 7: Informationsverlust durch Projektion des höherdimensionalen Merkmalraumes auf die entsprechenden Merkmale

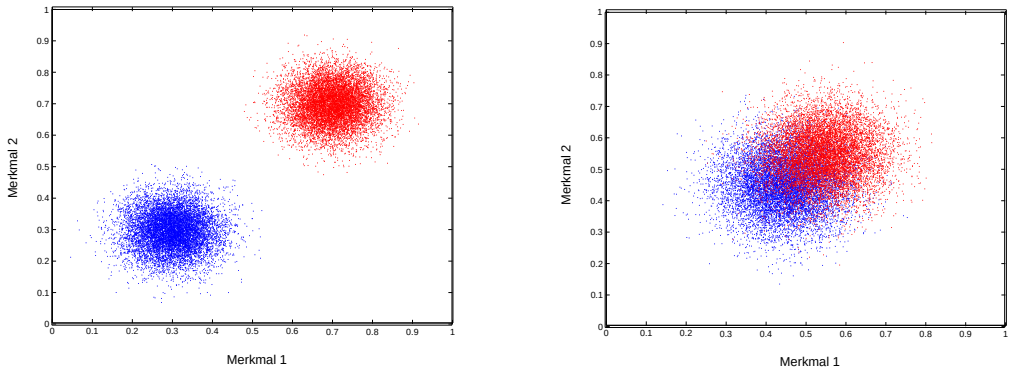
1. Mit eindimensionalen Merkmalräumen beginnen
2. Klassengebiete suchen und Polarisationen testen
3. Sind Gebiete mit $\lambda^{\mathcal{R}} \geq \lambda_{\min}$ für alle $\mathcal{R}$ dabei?
   (a) ja: Klassengebiete auf Redundanz testen; diese Klassen sind klassifizierbar
4. Sind noch Klassen übrig?
   (a) ja: Dimension der Merkmalräume durch Hinzunahme eines neuen Merkmals erhöhen, Komplexität noch ok?
      i. ja: weiter bei 2.
      ii. nein: Klassifikationsaufgabe nicht lösbar
   (b) nein: ermittelte Dichten mit Testprobe vergleichen
   (c) widersprechen die Dichten denen der Lernprobe (d. h. keine Teilmenge)?
      i. ja: Merkmale gefunden
      ii. nein: Lern- und Testprobe nicht repräsentativ

Algorithmus 1: Merkmalselektion durch Steigerung der Komplexität

5 Simulation

Das Vorgehen soll an einem Simulationsbeispiel verdeutlicht werden. Die Meßwerte von zwei diskret abgetasteten Signalverläufen werden im zweidimensionalen Merkmalraum abgebildet. Um eine übermäßige Granularität zu vermeiden, wurden je Klasse 10000 diskrete Daten verwendet. Damit war eine relativ hohe Auflösung von 50 Unterteilungen je Merkmal möglich. Erkennbar sind die Häufungen der Meßwerte, wobei sich in Abbildung 8(b) die

Objekthäufungen der beiden Klassen im Gegensatz zu Abbildung 8(a) überlagern.

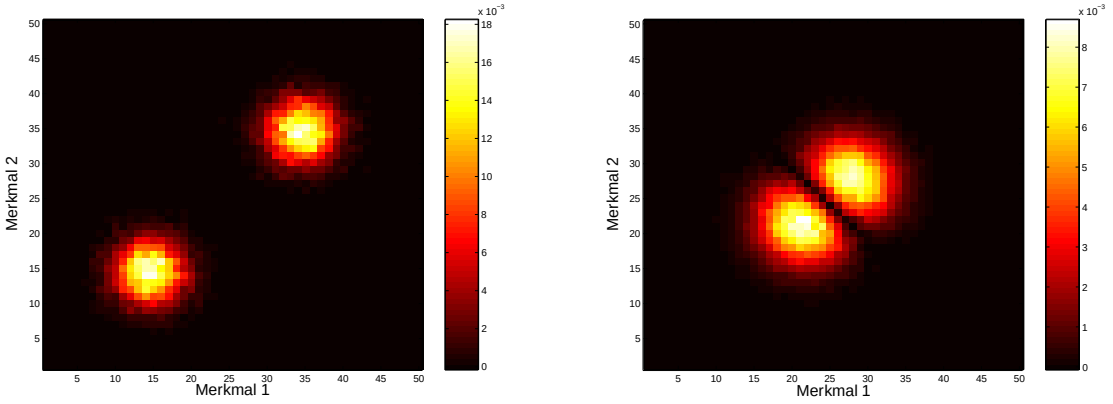


(a) Zwei sich nicht durchdringende Objekthaufen

(b) Zwei sich durchdringende Objekthaufen

Abbildung 8: Objekthaufen im zweidimensionalen Merkmalraum

Ein Vergleich der Heterogenitätsmaps zeigt deutlich die diskriminierenden Klassengebiete (Abbildung 9). Im Bereich der Durchdringung der beiden Objekthaufen wird trotz hoher Dichte die Heterogenität klein, vgl. Abbildung 9(b). Die Isolinie mit dem Niveau null trennt beide Klassengebiete voneinander ab. Betrachtet man die Durchdringungsmaps, stellt man



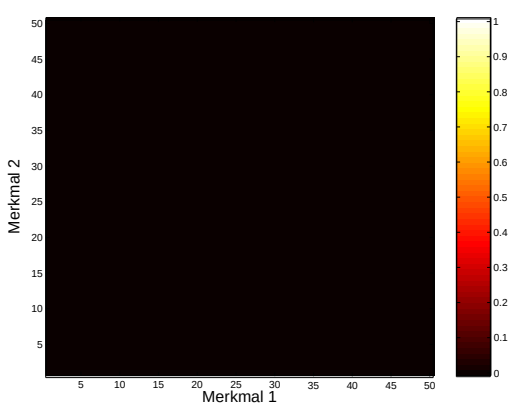
(a) Heterogenitätsmap für das Beispiel nach Abbildung 8(a)

(b) Heterogenitätsmap für das Beispiel nach Abbildung 8(b)

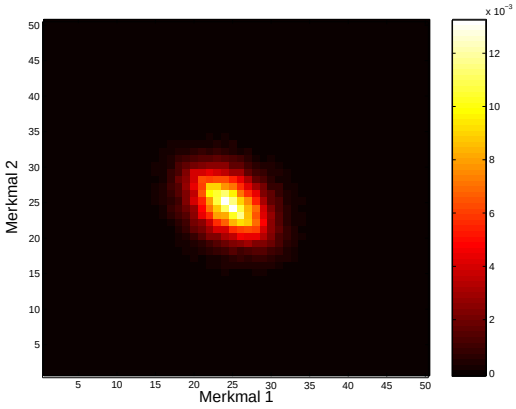
Abbildung 9: Heterogenitätsmaps

erwartungsgemäß für das Beispiel nach Abbildung 8(a) keine Durchdringung fest. Hingegen ist in Abbildung 10(b) ein Durchdringungsgebiet zu erkennen.

Eine Bewertung der Klassenkonstellation ohne Durchdringungsgebiet ergibt die maximale Heterogenität, alle Klassen können vollständig voneinander unterschieden werden. Da



(a) Durchdringungsmap für das Beispiel nach Abbildung 8(a)



(b) Durchdringungsmap für das Beispiel nach Abbildung 8(b)

Abbildung 10: Durchdringungsmaps

	Heterogenität	Durchdringung
Beispiel nach Abb. 8(a)	1.0	0.0
Beispiel nach Abb. 8(b)	0.62	0.38

Tabelle 1: Heterogenitäts- und Durchdringungswerte für die simulierten Beispiele

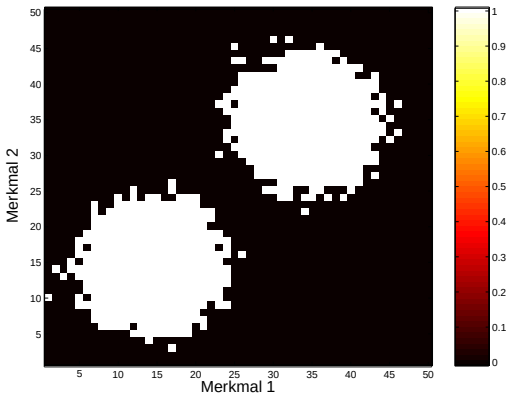
im Beispiel nach Abbildung 8(b) ein Durchdringungsgebiet existiert, kann dieser maximale Wert nicht erreicht werden (Tabelle 1).

Der Vergleich der Polarisationsmaps (Abbildung 11) ergibt für das erste Beispiel Klassengebiete, in denen sich sichere Entscheidungen $\lambda = 1$ treffen lassen. Für das zweite Beispiel ist diese Sicherheit nicht überall gleichermaßen gegeben. Im Bereich der Überlappung nimmt diese Sicherheit ab und strebt für gleich große Dichten gegen null.

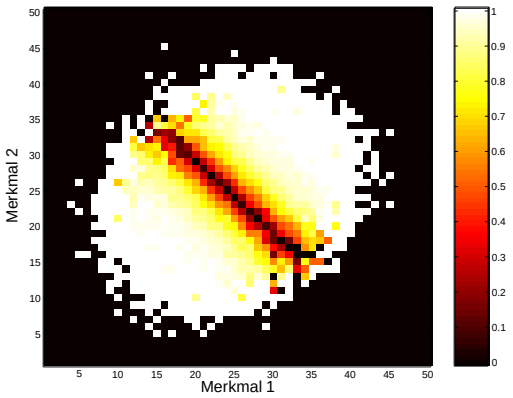
Im klassifizierten Merkmalraum (Abbildung 12) werden diese Gebiete zurückgewiesen und nur die Gebiete klassifiziert, in denen die vorgegebene Mindestsicherheit erfüllt wird.

6 Zusammenfassung

In diesem Beitrag wird eine Möglichkeit zur Untersuchung des Merkmalraumes vorgestellt, die auf dem Vergleich von Objekt- bzw. Aktivitätsdichten basiert. Es werden globale Kriterien zur Bewertung des Gesamtmerkmalraumes, als auch lokale Kriterien zur Bewertung der Dichteverältnisse an bestimmten Positionen des Merkmalraumes vorgestellt. Mit den lokalen Kriterien läßt sich eine Entscheidungsregel formulieren, mit der auf Randbedingungen eingegangen werden kann. Ein Merkmalselektionsverfahren untersucht die Möglichkeit der Klassentrennung bei möglichst kleiner Komplexität. Mit einem abschließenden Simulationsbeispiel sollen die Aussagen über die Dichteverhältnisse und das Subraummodell unterstrichen werden.

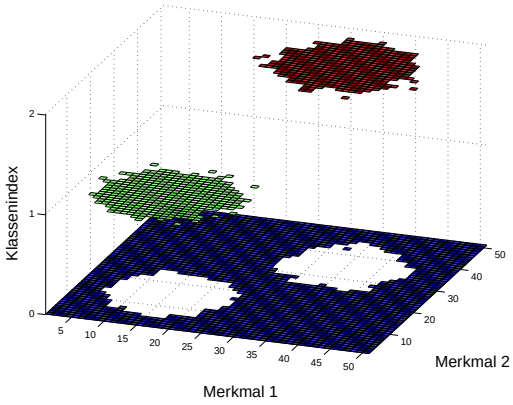


(a) Polarisationsmap für das Beispiel nach Abbildung 8(a)

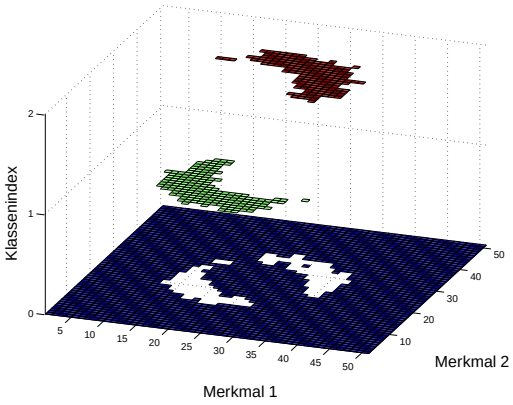


(b) Polarisationsmap für das Beispiel nach Abbildung 8(b)

Abbildung 11: Polarisationsmaps



(a) Klassifizierter Merkmalraum für das Beispiel nach Abbildung 8(a)



(b) Klassifizierter Merkmalraum für das Beispiel nach Abbildung 8(b)

Abbildung 12: Klassifizierung des Merkmalraumes

Literatur

[1] KOCH, M., T. KUHN und J. WERNSTEDT: *Fuzzy Control: optimale Nachbildung und Entwurf optimaler Entscheidungen*. Oldenbourg Verlag, München Wien, 1996.

[2] JÄPEL, D.: *Klassifikatorbezogene Merkmalauswahl*. Technischer Bericht Friedrich Alexander Universität Erlangen Nürnberg, Juli 1980.

[3] FRANK, P. M.: *Diagnoseverfahren in der Automatisierungstechnik*. at – Automatisierungstechnik, 2 1994.

[4] BARNSLEY, M. F.: *Fraktale: Theorie und Praxis der deterministischen Geometrie*. Spektrum Akademischer Verlag, Heidelberg · Berlin · Oxford, 1995.

[5] SCHOLL, R. und O. PFEIFFER: *Natur als fraktale Grafik*. Markt&Technik Verlag, Haar, 1991.

[6] LUNZE, J.: *Qualitative Modellierung dynamischer Systeme durch stochastische Automaten*. at – Automatisierungstechnik, 6 1998.

[7] KLEMM, E.: *Das Problem der Distanzbindungen in der hierarchischen Clusteranalyse*. Europäischer Verlag der Wissenschaften, Frankfurt am Main, 1995.

[8] LÄUTER, J.: *Stabile multivariate Verfahren: Diskriminanzanalyse – Regressionsanalyse – Faktoranalyse*. Akademie Verlag, Berlin, 1992.

[9] WERNSTEDT, J.: *Experimentelle Prozeßanalyse*. Verlag Technik, Berlin, 1989.

[10] LORENZ, K.: *Neuronales Netz oder Informationskontinuum?* it + ti, 6 1999.

[11] DEICHSEL, G. und H. J. TRAMPISCH: *Clusteranalyse und Diskriminanzanalyse*. Gustav Fischer Verlag, Stuttgart, New York, 1985.

Abstrakte Verhaltensmodellierung und -prognose auf der Basis räumlich verteilter Sensornetze mit Kohonen-Karten und Markov-Ketten

Jörg Matthes, Hubert B. Keller, Ralf Mikut

Forschungszentrum Karlsruhe GmbH
Institut für Angewandte Informatik (IAI)
PF 3640, D-76021 Karlsruhe
Telefon: 07247/82-5741, Telefax: 07247/82-5785, E-Mail: matthes@iai.fzk.de

1 Einführung

Für die Überwachung dynamischer Prozesse, deren physikalischen Größen räumlich verteilt auftreten, werden oft Netze räumlich verteilter Sensoren eingesetzt. Umweltprozesse, wie z. B. die örtliche Luftqualitätsänderung in Ballungsräumen, Gebäuden und Produktionsanlagen, aber auch verschiedene verfahrenstechnische Prozesse sind Vertreter dieser Prozessklasse. Solchen Prozessen hinterliegen komplexe Systeme, die durch eine hohe Anzahl an Freiheitsgraden und durch stochastische, schwer messbare Einflüsse gekennzeichnet sind.

Für die Beschreibung und Prognose des Verhaltens solcher stochastischer dynamischer Systeme bieten sich Markov-Modelle [1–3] an. Insbesondere Markov-Modelle mit diskreten Zuständen (Markov-Ketten), werden für die Modellierung stochastischer Systeme genutzt. Dies erfolgt jedoch bisher meist nur für Systeme, bei denen sich die diskreten Markov-Zustände direkt aus den nur qualitativ vorliegenden Messinformationen (z. B. Grenzwertüberschreitungen oder binäre Messgrößen) ergeben [4–6].

Die hier betrachteten räumlich verteilten Sensoren liefern i. Allg. wertekontinuierliche Messdaten. Diese sind aber substituive Ausprägungen nicht direkt messbarer Größen, also abstrakte Größen (z. B. Geruch beim Einsatz elektronischer Nasen [7, 8]), die sich nicht direkt in ein mathematisches Modell integrieren lassen. Es muss daher ein abstraktes Modell [9] beispielsweise auf der Basis diskreter Prozesszustände gefunden werden. Das formale Diskretisieren des Messbereichs jedes Einzelsensors zum Erzeugen diskreter Zustände liefert bei der meist hohen Anzahl von Sensoren eine große Anzahl diskreter Zustände. Die Verteilung dieser Zustände ist dabei völlig unabhängig von der tatsächlichen Verteilung der im Prozess auftretenden bzw. in den Lerndaten enthaltenen relevanten Zuständen. Das Bestimmen der Zustandsübergangswahrscheinlichkeiten für jeden dieser Zustände erfordert eine sehr große Messdatenmenge. Insbesondere für Zustände, die in den Messdaten selten oder gar nicht erreicht werden, ist das Bestimmen der Übergangswahrscheinlichkeiten problematisch.

Ziel dieses Vortrages ist es,

- eine kurze Einführung zu Markov-Ketten zu gegeben (Abschnitt 2),
- ein datengestütztes Verfahren vorzuschlagen, das nach einer Schätzung abstrakter Prozesszustände mit Hilfe von Kohonen-Karten [10–12] Markov-Ketten als Prognosemodell generiert (Abschnitt 3), und

- dieses Verfahren anhand eines Simulationsbeispiels zur Luftqualitätsüberwachung zu demonstrieren (Abschnitt 4).

2 Markov-Ketten

Betrachtet wird ein diskreter stochastischer Prozess $\{X_D(k), k = 0, 1, ...\}$. Der Wert $x_D(k)$ der Zufallsvariable $X_D(k)$ wird als Zustand bezeichnet und kann zum Zeitpunkt k einen der N möglichen diskreten Zustände des Systems annehmen $(X_D(k) = i, i = 1, ..., N)$. Eine Sequenz von Zufallsvariablen $X_D(0), X_D(1), ...,$ $X_D(k), X_D(k+1)$ bildet eine Markov-Kette, wenn die Wahrscheinlichkeit dafür, dass sich das System zum Zeitpunkt $k+1$ im Zustand $x_D(k+1)$ befindet, ausschließlich von der Wahrscheinlichkeit abhängt, dass sich das System zum Zeitpunkt k im Zustand $x_D(k)$ befindet. Das heißt, es muss gelten:

$$\begin{aligned} &P\big(X_D(k+1) = x_D(k+1)|X_D(k) = x_D(k), ..., X_D(0) = x_D(0)\big) = \\ &P\big(X_D(k+1) = x_D(k+1)|X_D(k) = x_D(k)\big). \end{aligned} \tag{1}$$

Bei den hier betrachteten zeitdiskreten Markov-Ketten finden Zustandsübergänge in einem festen Takt statt. Das Beibehalten des aktuellen Zustands wird als Zustandsübergang auf denselben Zustand aufgefasst. Befindet sich das System zum Zeitpunkt k im Zustand i, dann erfolgt ein Übergang zum Zeitpunkt $k+1$ in den Zustand j mit der Wahrscheinlichkeit p_{ij}. Ist p_{ij} (wie hier vorausgesetzt) konstant, liegt eine homogene Markov-Kette vor. Da p_{ij} Wahrscheinlichkeiten sind, gilt:

$$0 \leq p_{ij} \leq 1. \tag{2}$$

Das System muss nach jedem Zustandsübergang in irgendeinem Zustand sein, das heißt es muss für alle i gelten:

$$\sum_{j=1}^{N} p_{ij} = 1. \tag{3}$$

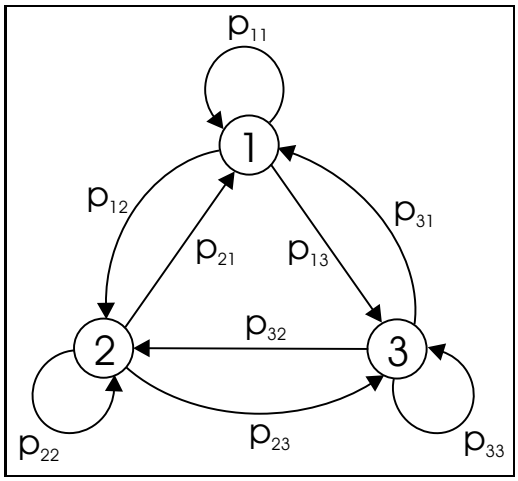


Abbildung 1: Markov Kette mit drei Zuständen

Abb. 1 zeigt die grafische Veranschaulichung einer Markov-Kette mit $N = 3$ Zuständen. Die Kreise symbolisieren die Zustände; die Übergangswahrscheinlichkeiten p_{ij} sind an die jeweiligen Pfeile zwischen den Zuständen angetragen.

Die Übergangswahrscheinlichkeiten lassen sich in Form einer Matrix

$$\mathbf{P} = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N1} & p_{N2} & \cdots & p_{NN} \end{pmatrix} \tag{4}$$

zusammenfassen. Aus (3) folgt, dass die Zeilensumme für jede Zeile von $\mathbf{P}$ eins ist.

Weiter wird für jeden Zustand i eine Zustandswahrscheinlichkeit $\pi_i(k)$ eingeführt, die die Wahrscheinlichkeit dafür angibt, dass sich das System zum Zeitpunkt k im Zustand i befindet.

Durch Definieren des Zeilenvektors $\boldsymbol{\pi}(k)$, der die Komponenten $\pi_i(k)$ enthält, folgt:

$$\boldsymbol{\pi}(k+1) = \boldsymbol{\pi}(k)\mathbf{P} \tag{5}$$

bzw.

$$\boldsymbol{\pi}(k) = \boldsymbol{\pi}(0)\mathbf{P}^k. \tag{6}$$

Eine allgemeinere Form der Markov-Ketten stellen die **Semi-Markov-Ketten** dar [1]. Während bei den "klassischen", zeitdiskreten Markov-Ketten in jedem Takt ein Zustandsübergang (evtl. nur in denselben Zustand) stattfindet, ist bei zeitdiskreten Semi-Markov-Ketten die Anzahl der Takte zwischen zwei Zustandsübergängen ebenfalls eine Zufallsvariable, deren Verteilung vom aktuellen und vom zukünftigen Zustand abhängt.

Bei **Hidden-Markov-Modellen** (HMM) [13] ist nicht nur der Übergang von einem Zustand in einen anderen probabilistisch, sondern auch die generierte Ausgabe des Zustands [3]. Für jeden Zustand eines HMM existieren daher Wahrscheinlichkeiten für alle Ausgabesymbole.

3 Markov-Kette als Prognosemodell

3.1 Prinzip

Im hier vorgestellten Ansatz zum Generieren eines Prognosemodells auf der Basis von Markov-Ketten werden mittels selbstorganisierenden Kohonen-Karten **diskrete Prozesszustände** gefunden. Bei der beschriebenen Methode repräsentieren die Knoten der Kohonen-Karte nach der Lernphase die diskreten Zustände, für die die Annahme getroffen wird, dass sie die Markov-Eigenschaft besitzen. Damit aktiviert jeder gemessene Prozesszustand Knoten in der Kohonen-Karte, wobei entweder eine scharfe Zuordnung (Gewinnerneuron) oder eine unscharfe Zuordnung (Zugehörigkeiten) erfolgt.

Die Übergangswahrscheinlichkeiten der Zustandsübergänge werden anhand der Trainingsdaten geschätzt. Dafür wird hier eine Methode auf Basis einer unscharfen Klassifikation vorgeschlagen, die auf das Lösen eines restringierten Optimierungsproblems führt. Für eine qualitative Überwachung des Prozesses lassen sich den gefundenen diskreten Prozesszuständen der Markov-Kette **benennbare Prozesssituationen** zuordnen. Nach einer scharfen oder unscharfen Klassifikation kann mit der Markov-Kette die Prognose der Wahrscheinlichkeit (des Eintretens) zukünftiger Prozesszustände und Prozesssituationen erfolgen.

Läuft der betrachtete Prozess in verschiedenen **Prozessmodi** ab, in denen trotzdem ähnliche Zustände durchlaufen werden, kann es vorkommen, dass solche Zustände die Forderung nach Markov-Eigenschaft bereits für die Lerndaten nicht ausreichend erfüllen.

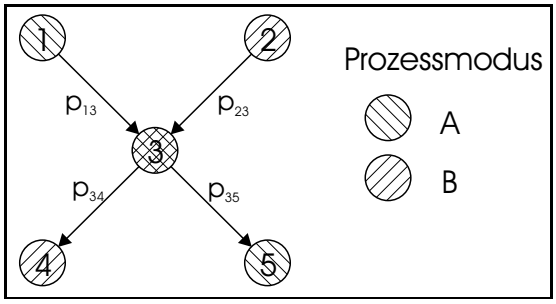


Abbildung 2: Verletzung der Markov-Bedingung bei verschiedenen Prozessmodi

Dies verdeutlicht Abb. 2. Im Prozessmodus A werden vorzugsweise nacheinander die Zustände 1,3,5 aktiviert, im Prozessmodus B die Zustände 2,3,4. Die Wahrscheinlichkeiten für den Übergang in die Zustände 4 oder 5 hängen daher im Zustand 3 stark vom vorhergehenden Zustand ab. Damit ist die Markov-Bedingung verletzt. Die geschätzten Übergangswahrscheinlichkeiten stellen also nur ein Kompromiss über alle Prozessmodi dar. Für den aktuellen Prozessmodus sind sie deshalb nicht optimal. Lassen sich die Prozessmodi Teilen der Lerndaten zuordnen, so kann für jeden Modus eine separate Markov-Kette (Multi-Modell-Ansatz) generiert werden, indem die Lerndaten für das Bestimmen der Übergangswahrscheinlichkeiten nach den Prozessmodi aufgespalten werden. Während der Überwachungsphase erfolgt die Auswahl der relevanten Markov-Kette automatisch. Dafür wird anhand der bereits durchlaufenen Sequenz von Prozesszuständen diejenige Markov-Kette als Prognosemodell ausgewählt, die diese Sequenz von allen anderen Markov-Ketten am wahrscheinlichsten durchläuft.

Ähnliche Ansätze werden in der Spracherkennung [13,14] bereits erfolgreich genutzt. Dabei werden zunächst aus kurzen Stücken des Sprachsignals mittels verschiedener Signalanalyseverfahren Merkmale generiert. Durch Nutzung von Kohonen-Karten erfolgt eine Dimensionsreduktion der Merkmalsvektoren auf eine zweidimensionale Karte. Die Knoten der Kohonen-Karte repräsentieren nach der Lernphase die typischen Phoneme bzw. Silben der Sprache (Prozesssituationen). Ein Wort der Sprache ist eine Sequenz der erlernten Phoneme. Jedem Wort wird ein diskretes Hidden-Markov-Modell zugeordnet. Dessen diskrete Zustände sind die Phoneme, die im Wort enthalten sind. Durch das Lernen der Übergangswahrscheinlichkeiten mit Hilfe des Baum-Welch-Algorithmus [13, 15] wird die Dynamik der Aussprache eines Wortes ermittelt. Die Ausgabeschicht eines solchen Hidden-Markov-Modells besitzt nicht nur die erwarteten Phoneme des Wortes, sondern auch andere, in der Trainingsphase erkannte Phoneme. Bei der Spracherkennung wird mit Hilfe des Viterbi-Algorithmus [13, 16] für jedes gelernte Hidden-Markov-Modell die optimale Zustandssequenz bestimmt, die zur Ausgabe der aktuellen Sequenz erkannter Phoneme führt. Das Modell, in dem diese optimale Zustandssequenz mit der höchsten Wahrscheinlichkeit durchlaufen wird, repräsentiert das zu erkennende Wort.

Vergleichbare Verfahren werden auch zur Erkennung menschlicher Gesten eingesetzt [17]. Im Forschungsstadium befinden sich weiterhin Systeme, mit denen eine Klassifikation des aktuellen Umgebungszustandes (Kontext) tragbarer Geräte (z. B. Mobiltelefone) auf der Basis von low-level Sensoren ermöglicht werden soll [18].

Im Gegensatz zum hier vorgestellten Ansatz zur Nutzung von Markov-Ketten für die Verhaltensprognose liegt der Schwerpunkt der eben beschriebenen Anwendungen in der Klassifikation (z. B. von Phonemen oder Wörtern). Diese Klassifikation wird dort durch übergeordnete Markov-Modelle überwacht bzw. unterstützt.

3.2 Bestimmen diskreter Zustände mittels Kohonen-Karte

3.2.1 Datenvorverarbeitung

Wird ein Prozess betrachtet, der durch ein Sensornetz mit S Sensoren überwacht wird, dann liegen zu jedem Zeitpunkt k S Messwerte über den Prozess vor. In der Praxis messen die Sensoren meist nicht im gleichen Takt. Daher ist es notwendig, die über einen bestimmten Zeitraum gemessenen Lerndaten einem Resampling zu unterwerfen. Ergebnis des Resamplings sind Lerndaten, die die Messwerte für alle Sensoren zu festen, äquidistanten Zeitpunkten k mit der Abtastzeit T enthalten. Diese Messwerte werden für jeden Zeitpunkt k zu einem Messvektor $\mathbf{y}(k)$ zusammengefasst.

Da hier dynamische Systeme betrachtet werden, wird der aktuelle Zustand des Systems nicht nur durch die aktuellen Messwerte, sondern zusätzlich auch durch dessen zeitliche Ableitungen beschrieben. Durch welche Messgrößen (und deren Ableitungen) der Zustand des Systems hinreichend gut beschrieben wird, ist für die betrachteten Systeme a priori nicht bekannt. Daher wird als Repräsentant des aktuellen Zustands des Systems ein Vektor $\hat{\mathbf{x}}(k)$ eingeführt, der die aktuellen Messwerte (also den Messvektor $\mathbf{y}(k)$) und deren zeitliche Ableitungen bis zur Ordnung n zum Zeitpunkt k enthält. Für das Ermitteln der Ableitungen sind je nach Beschaffenheit der Messsignale entsprechende Filter zu verwenden. Im einfachsten Fall kann die Ableitung erster Ordnung nach Tiefpassfilterung des Messsignals zur Rauschunterdrückung durch den Differenzenquotienten approximiert werden [19].

Anschließend werden die Elemente aller Vektoren $\hat{\mathbf{x}}(k)$ der Lerndaten so skaliert, dass der kleinste Wert eines jeden Elements von $\hat{\mathbf{x}}(k)$ in den Lerndaten nach dem Skalieren den Wert null und der größte Messwert den Wert eins annimmt.

Ergebnis der Datenvorverarbeitung ist ein Lerndatensatz mit dem Umfang L von Vektoren $\hat{\mathbf{x}}_N(k) \in [0,1]^{(n+1)S}$ mit $k = 1, ..., L$.

3.2.2 Training der Kohonen-Karte

Das grundsätzliche Ziel von selbstorganisierenden Karten (sog. Kohonen-Karten) [10–12] ist es, ein Eingangssignalmuster beliebiger Dimension durch eine nichtlineare Transformation auf eine niederdimensionale (meist zweidimensionale) diskrete Karte abzubilden [3]. Dabei wird Topologieerhaltung angestrebt.

Hier wird eine zweidimensionale Kohonen-Karte verwendet. Damit ist eine Abbildung vom Eingangsraum der Dimension $(n+1)S$ auf die zweidimensionale Kohonen-Karte notwendig. Die Wahl einer zweidimensionalen Karte ist insbesondere für eine spätere Visualisierung sinnvoll.

Zu Beginn muss die Größe, also die Anzahl der Knoten N, der Kohonen-Karte festgelegt werden. Die Wahl einer großen Anzahl von Knoten führt zu einer verbesserten Repräsentation der Lerndaten. Allerdings stellt jeder Knoten der Kohonen-Karte später einen diskreten Zustand in der Markov-Kette dar. Für jeden dieser Zustände müssen aus den Lerndaten die Übergangswahrscheinlichkeiten für Übergänge in andere Zustände ermittelt werden. Ist die Anzahl der Zustände zu groß, ist die

statistische Absicherung für die berechneten Übergangswahrscheinlichkeiten nicht gewährleistet. Die Wahl der Größe des Kohonen-Netzes stellt damit einen Kompromiss zwischen Datenrepräsentationsgüte und Güte der statistischen Absicherung der Übergangswahrscheinlichkeiten dar.

Nach der Lernphase repräsentieren die N Knoten der Kohonen-Karte die Lerndaten. Jedem Knoten $i, i = 1, .., N$ ist ein Referenzvektor $\mathbf{r}_i \in [0,1]^{(n+1)S}$ im Eingangsraum zugeordnet.

3.2.3 Benennen der diskreten Prozesszustände

Um den diskreten Prozesszuständen benennbare Prozesssituationen zuordnen zu können, ist es sinnvoll, die zugehörigen Referenzvektoren $\mathbf{r}_i$ zu betrachten [11], und diese entsprechend der Normierung bei der Datenvorverarbeitung wieder zurück zu skalieren. Insbesondere der Teil eines Referenzvektors, der einem zugehörigen, imaginären Messvektor $\mathbf{y}_i$ entspricht, ist dazu heranzuziehen.

3.2.4 Prozesszustandsklassifikation

Für die Klassifikation eines Eingabevektors $\hat{\mathbf{x}}_N(k)$ bestehen prinzipiell zwei Möglichkeiten:

Bei einer **scharfen Klassifikation** des Eingabevektors wird derjenige Knoten i der Kohonen-Karte ausgewählt, dessen Referenzvektor $\mathbf{r}_i$ den kleinsten euklidischen Abstand $d_i(k)$ zum Eingabevektor $\hat{\mathbf{x}}_N(k)$ besitzt:

$$d_i(k) = \min_{j=1,\ldots,N} \{d_j(k)\} \tag{7}$$

mit

$$d_j(k) = \|\hat{\mathbf{x}}_N(k) - \mathbf{r}_j\|_2 = \sqrt{(\hat{\mathbf{x}}_N(k) - \mathbf{r}_j)^T (\hat{\mathbf{x}}_N(k) - \mathbf{r}_j)}. \tag{8}$$

Die scharfe Klassifikation liefert einen Zustandszugehörigkeitsvektor $\boldsymbol{\mu}(k) \in [0,1]^N$, dessen Element i den Wert eins und alle anderen Elemente den Wert null annehmen.

Durch eine **unscharfe Klassifikation** soll der Tatsache Rechnung getragen werden, dass die Eingangsvektoren i. Allg. wertekontinuierliche Elemente besitzen und damit eine Abbildung auf den nächstgelegenen diskreten Zustand (Knoten) einen Informationsverlust darstellt. Liegt beispielsweise ein Eingabevektor zwischen den Referenzvektoren zweier Zustände, so ist es sinnvoll, diesem Eingabevektor sowohl eine Zugehörigkeit zu dem einen als auch zu dem anderen Zustand zuzuordnen. Die Gesamtzugehörigkeit eines Eingabevektors zu verschiedenen Zuständen (Summe eins) soll also in Abhängigkeit von den Abständen des Eingabevektors zu den Referenzvektoren auf die umliegenden (hier vier) diskreten Zustände aufgeteilt werden. Dazu lassen sich die vier Elemente $\mu_i(k)$ des Zugehörigkeitsvektors $\boldsymbol{\mu}(k) \in [0,1]^N$, deren zugeordnete Referenzvektoren $\mathbf{r}_i$ die vier kleinsten Abstände $d_K(k)$ zum aktuellen

Eingabevektor $\hat{\mathbf{x}}_N(k)$ besitzen, beispielsweise mit

$$\boldsymbol{\mu}_i(k) = \frac{\frac{1}{d_i(k)}}{\sum\limits_K \frac{1}{d_K(k)}} \tag{9}$$

berechnen. Die übrigen Elemente von $\boldsymbol{\mu}(k)$ erhalten den Wert null. Durch diese Berechnungsweise ist sichergestellt, dass alle Elemente von $\boldsymbol{\mu}(k)$ zwischen null und eins liegen und deren Summe gleich eins ist:

$$\sum_{i=1}^{N} \mu_i = 1. \tag{10}$$

3.3 Schätzen der Übergangswahrscheinlichkeiten

3.3.1 Schätzen der Übergangswahrscheinlichkeiten nach scharfer Klassifikation

Anhand der Trainingsdaten muss eine Schätzung $\hat{\mathbf{P}} \in [0,1]^{N\times N}$ für die Matrix $\mathbf{P}$, die die Übergangswahrschweinlichkeiten für alle diskreten Zustände enthält, erfolgen. Als Schätzung für die Elemente von $\mathbf{P}$ dienen die relativen Häufigkeiten der jeweiligen Zustandsübergänge:

$$\hat{p}_{ij} = \frac{z_{ij}}{Z_i}. \tag{11}$$

Dabei ist Z_i die Anzahl der Trainingseingangsvektoren, die durch die scharfe Klassifikation dem diskreten Zustand i zugeordnet wurden. (Ausgenommen ist dabei der letzte Eingangsvektor, da für diesen der Folgezustand nicht bekannt ist.) z_{ij} ist die Anzahl der Zustandsübergänge vom diskreten Zustand i in den diskreten Zustand j in den scharf klassifizierten Lerndaten.

Für das Schätzen der Übergangswahrscheinlichkeiten ist sowohl bei der scharfen als auch der unscharfen Klassifikation für jeden diskreten Zustand eine Mindestbeispielzahl von Aktivierungen in den Trainingsdaten erforderlich. Wenn diese Mindestbeispielzahl unterschritten wird, werden die entsprechnenden diskreten Zustände gelöscht. In diesem Fall werden die Trainingsdaten den verbliebenen diskreten Nachbarzuständen zugeordnet.

3.3.2 Schätzen der Übergangswahrscheinlichkeiten nach unscharfer Klassifikation

Das Schätzen der Übergangswahrscheinlichkeiten $\hat{p}_{ij}$ auf der Basis der scharfen Klassifikation nimmt den in Abschnitt 3.2.4 erwähnten Informationsverlust bei der Abbildung der Eingangsvektoren auf diskrete Zustände in Kauf. Um diesen Nachteil zu vermeiden, wird ein Verfahren auf der Basis der unscharfen Klassifikation eingesetzt. Die Lösung für das dabei auftretende restringierte Optimierungsproblem ist [20] bzw. [21] entnommen, wobei dort strukturell identische Probleme für die Fuzzy-Regelgenerierung gelöst werden.

Ergebnis der unscharfen Klassifizierung der L Trainingsvektoren sind L Zustandszugehörigkeitsvektoren $\boldsymbol{\mu}(k) \in [0,1]^N$ mit $(k = 1, ..., L)$. Die Vektoren $\boldsymbol{\mu}(1)$ bis $\boldsymbol{\mu}(L-1)$ werden nun zur Matrix der Ausgangszustände

$$\boldsymbol{\Psi}_0 = \begin{pmatrix} \boldsymbol{\mu}^T(1) & \boldsymbol{\mu}^T(2) & \cdots & \boldsymbol{\mu}^T(L-1) \end{pmatrix} \tag{12}$$

und die Vektoren $\boldsymbol{\mu}(2)$ bis $\boldsymbol{\mu}(L)$ zur Matrix der Folgezustände

$$\boldsymbol{\Psi}_1 = \begin{pmatrix} \boldsymbol{\mu}^T(2) & \boldsymbol{\mu}^T(3) & \cdots & \boldsymbol{\mu}^T(L) \end{pmatrix} \tag{13}$$

zusammengefasst $\left(\boldsymbol{\Psi}_0, \boldsymbol{\Psi}_1 \in [0,1]^{N \times (L-1)}\right)$. Aus (10) folgt, dass die Spaltensummen von $\boldsymbol{\Psi}_0$ und $\boldsymbol{\Psi}_1$ jeweils eins sind.

Ziel der Optimierung ist es, eine solche Matrix $\hat{\mathbf{P}}$ zu ermitteln, dass die auf $\hat{\mathbf{P}}$ und $\boldsymbol{\Psi}_0$ basierende Prognose $\hat{\mathbf{P}}^T\boldsymbol{\Psi}_0$ für $\boldsymbol{\Psi}_1$ einen minimalen Fehler besitzt. Das zu lösende restringierte Optimierungsproblem lässt sich damit mit

$$Q(\hat{\mathbf{P}}) = \tfrac{1}{2}\|\hat{\mathbf{P}}^T\boldsymbol{\Psi_0} - \boldsymbol{\Psi_1}\|_F^2 \rightarrow \min_{\hat{\mathbf{P}}} \tag{14}$$

angeben, das den Restriktionen

$$\hat{\mathbf{P}} \geq \mathbf{0}_{N \times N} \tag{15}$$

$$\hat{\mathbf{P}} \cdot \mathbf{1}_N = \mathbf{1}_N. \tag{16}$$

unterliegt. $\|\cdot\|_F$ bezeichnet die Frobenius-Norm. Die Restriktion (15) stellt sicher, dass $\hat{\mathbf{P}}$ nur nicht-negative Elemente besitzt (vgl. (2)), und (16) gewährleistet, dass die Zeilensummen von $\hat{\mathbf{P}}$ jeweils eins sind (vgl. (3)).

Die Lösung lautet:

$$\hat{\mathbf{P}}_{opt}^T = \underbrace{\left(\mathbf{I}_N - \tfrac{1}{N}\mathbf{1}_{N\times N}\right)}_{\mathbf{Z}_N} \underbrace{\boldsymbol{\Psi}_1\boldsymbol{\Psi}_0^T\left(\boldsymbol{\Psi}_0\boldsymbol{\Psi}_0^T\right)^{-1}}_{\hat{\mathbf{P}}_{LS}^T} + \tfrac{1}{N}\mathbf{1}_{N\times N} + \underbrace{\left(\mathbf{I}_N - \tfrac{1}{N}\mathbf{1}_{N\times N}\right)}_{\mathbf{Z}_N}\mathbf{L}^*\left(\boldsymbol{\Psi}_0\boldsymbol{\Psi}_0^T\right)^{-1} \tag{17}$$

mit der Langrange-Matrix $\mathbf{L}^*$, der nichtrestringierten Least-Square-Lösung $\hat{\mathbf{P}}_{LS}$, einer Zentriermatrix $\mathbf{Z}_N$, einer Eins-Matrix $\mathbf{1}$ und der Einheitsmatrix $\mathbf{I}$. Einfachere Ansätze (Summierung der Zugehörigkeitswahrscheinlichkeiten) führen u. U. zu irreführenden Ergebnissen [20].

Das Bestimmen der Übergangswahrscheinlichkeiten auf der Basis der scharfen Klassifikation stellt einen Spezialfall des eben gezeigten Verfahrens dar. Die Elemente von $\boldsymbol{\Psi}_0$ und $\boldsymbol{\Psi}_1$ besitzen dabei entsprechend nur die Werte null oder eins.

3.4 Verhaltensprognose

Ausgangspunkt für die Online-Verhaltensprognose bildet die Datenvorverarbeitung der aktuellen Messwerte $\mathbf{y}(k)$ (Abb. 3). Der daraus resultierende Zustandsrepräsentant $\hat{\mathbf{x}}(k)$ wird der Prozesszustandsklassifikation (vgl. 3.2.4) zugeführt. Ergebnis dieser Klassifikation ist der Vektor $\boldsymbol{\mu}(k) \in [0,1]^N$. Ein Element $\mu_i(k)$ gibt die Zugehörigkeit des Prozesszustands zum Zeitpunkt k zum diskreten Zustand i an. Bei Verwendung der scharfen Klassifikation enthält $\boldsymbol{\mu}(k)$ nur eine Eins und sonst Nullen. Bei der unscharfen Klassifikation können die Elemente $\mu_i(k)$ Werte zwischen null und eins annehmen.

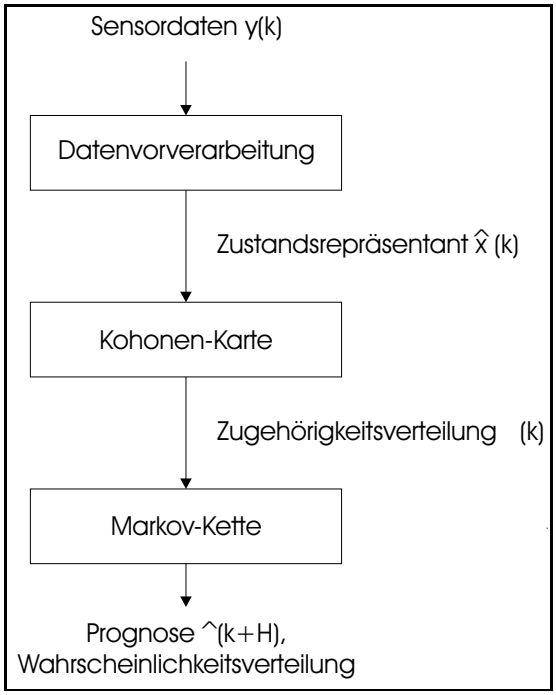


Abbildung 3: Ablauf der Online-Prognose

Die Zustandszugehörigkeit $\mu_i(k)$ wird nun direkt als Schätzwert für die Zustandswahrscheinlichkeit $\pi_i(k)$ (vgl. Abschnitt (2)) verwendet. Das bedeutet:

$$\hat{\boldsymbol{\pi}}(k) = \boldsymbol{\mu}(k). \tag{18}$$

Unter Verwendung von (5) lässt sich nun eine Einschrittprognose

$$\hat{\boldsymbol{\pi}}(k+1) = \hat{\boldsymbol{\pi}}(k)\hat{\mathbf{P}} \tag{19}$$

für $\hat{\boldsymbol{\pi}}(k+1)$ angeben. Unter Nutzung von (6) kann eine Mehrschrittprognose mit dem Prognosehorizont H erfolgen durch:

$$\hat{\boldsymbol{\pi}}(k+H) = \hat{\boldsymbol{\pi}}(k)\hat{\mathbf{P}}^H. \tag{20}$$

Bei Verwendung mehrerer Modelle (Multi-Modell-Ansatz, vgl. Abschnitt 3.1) muss zunächst das relevante Modell ausgewählt, d. h. der aktuelle Prozessmodus erkannt werden. Alle M Modelle besitzen die gleichen diskreten Zustände, jedoch verschiedene Übergangswahrscheinlichkeitsmatrizen $\hat{\mathbf{P}}_m$ $(m = 1, ..., M)$.

Zunächst werden die Zugehörigkeitsvektoren $\boldsymbol{\mu}(k+1-R), ..., \boldsymbol{\mu}(k)$, die durch die letzten R erfassten Messvektoren ermittelt wurden, wie in Abschnitt (3.3) beschrieben zu den Matrizen $\boldsymbol{\Psi}_0$ und $\boldsymbol{\Psi}_1$ zusammengefasst. Anschließend wird mit diesen Matrizen der Prognosefehler

$$Q_m(\hat{\mathbf{P}}_m) = \|\hat{\mathbf{P}}_m^T\boldsymbol{\Psi}_0 - \boldsymbol{\Psi}_1\|_F^2 \tag{21}$$

für alle M Modelle $\hat{\mathbf{P}}_1, ..., \hat{\mathbf{P}}_M$ bestimmt (vgl. (14)). Dasjenige Modell $\hat{\mathbf{P}}_m$, dass den kleinsten Prognosefehler aufweist, wird für die Prognose der zukünftigen Prozesszustände verwendet.

Für die **Prognose von Prozesssituationen** sind die prognostizierten Zustandswahrscheinlichkeiten der Prozesszustände, die jeweils derselben Prozesssituation zugeordnet sind, zu addieren.

4 Beispiel: Luftqualitätsüberwachung

Die Anwendung des dargestellten Verfahrens wird nun am Beispiel der Luftqualitätsüberwachung mit verteilten Sensoren skizziert. Ausgangspunkt ist eine Strömungssimulation, mit der sich die zeitliche und örtliche Konzentration von Schadstoffen nachbilden lässt. Im hier dargestellten Beispiel wird eine Landschaft betrachtet, die durch feste Hindernisse (z. B. Berge) und verschiedene, typische Strömungsverhältnisse (z. B. Windrichtungen) gekennzeichnet ist. Die dabei ablaufenden unterschiedlichen Szenarien ergeben im Raum der Zustandsrepräsentanten verschiedene Trajektorien, die in der zweidimensionalen Projektion der Kohonen-Karte jeweils durch mehrere, benachbarte Knoten repräsentiert werden.

Abb. 4 (links) zeigt eine Momentaufnahme der örtlichen Konzentrationsverteilung zum Zeitpunkt k. Diese ist hier stark durch eine Hindernis (Berg) geprägt. In Abb. 4 (links) sind weiterhin die Standorte der Sensoren des Sensornetzes eingezeichnet.

Nach der Trainingsphase der Kohonen-Karte liefert die unscharfe Klassifikation der Momentaufnahme (Abb. 4 (links)) die in Abb. 4 (rechts) dargestellte Zugehörigkeitsverteilung auf die 6×6 diskreten Zustände ($N = 36$).

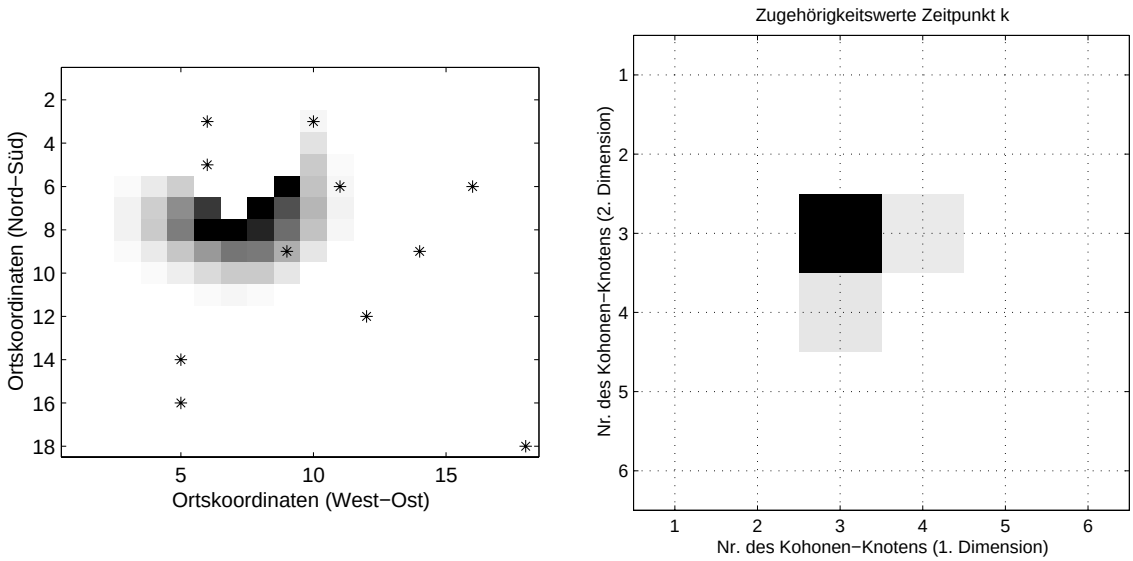


Abbildung 4: Beispiel für örtliche Emissionsverteilung (Grauwerte zwischen 0% - weiß und 100% - schwarz) mit Sensorpositionen * (linkes Teilbild) und Repräsentation dieser Beispielsituation in der Kohonen-Karte (rechtes Teilbild)

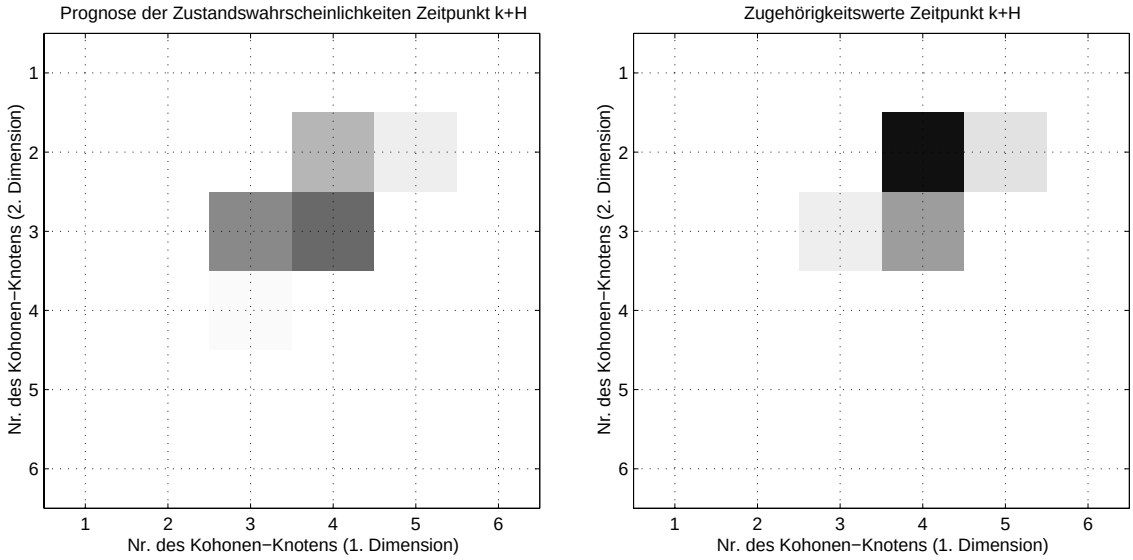


Abbildung 5: Prognose der Zustandswahrscheinlichkeiten für den Zeitpunkt $k + H$ (linkes Teilbild) und Zugehörigkeitsverteilung des zum Zeitpunkt $k+H$ eingetretenen Zustands (rechtes Teilbild)

Ausgehend von dieser Zugehörigkeitsverteilung erfolgt mit Hilfe der gelernten Markov-Kette eine Prognose der Wahrscheinlichkeitsverteilung des Eintretens bestimmter diskreter Zustände (Abb. 5 (links)) für den Zeitpunkt $k + H$, die im Zeitpunkt k erstellt wurde. Abb. 5 (rechts) zeigt die zum Zeitpunkt $k + H$ (hier: $H = 10$) eingetretene Zugehörigkeitsverteilung. Für die Zustände, die zum Zeitpunkt $k + H$ aktiviert werden, wurde eine hohe Wahrscheinlichkeit des Eintretens prognostiziert.

5 Zusammenfassung und Ausblick

Im vorliegenden Beitrag wird ein Verfahren vorgestellt, dass datengestützt eine Markov-Kette als Prognosemodell für komplexe Prozesse (z. B. Umweltprozesse) generiert. Die Beobachtung des Prozesses findet dabei mit Hilfe von räumlich verteilten Sensornetzen statt, die teilweise auch abstrakte Messgrößen (z. B. Geruch) liefern. Für das Bestimmen diskreter Prozesszustände wird eine Kohonen-Karte verwendet. Die Lernphase umfasst die Ermittlung einer geeigneten Kohonen-Karte und das Schätzen der Übergangswahrscheinlichkeiten der Markov-Kette. Die Klassifikation der Zustände kann dabei sowohl scharf als auch unscharf erfolgen. Es wird aufgezeigt, wie eine oder wie mehrere Markov-Ketten (Multi-Modell-Ansatz) als Prognosemodell eingesetzt werden können. Anhand des Beispiels der Luftqualitätsüberwachung wird der Einsatz einer Markov-Kette als Prognosemodell verdeutlicht.

Zukünftig sind Verfahren zu entwickeln, die die Entscheidung über die Anzahl diskreter Zustände und Markov-Ketten (bei Multi-Modell-Ansätzen) automatisch treffen. Einen Ansatz dafür bietet das Bestimmen der Abweichung der gefundenen diskreten Zustände von der Markov-Bedingung für die Trainingsdaten. Außerdem ist zu untersuchen, ob zum einen die Rückinterpretation der Prognose in den Eingangsraum der Kohonen-Karte und zum anderen die Analyse einzelner Pfade des Prozesses in der Markov-Kette nützliche Informationen über den Prozess liefern. Weiterhin ist zu prüfen, ob eine Verbesserung des Prognosemodells durch den Einsatz von Semi-Markov-Ketten möglich ist.

Literatur

[1] Howard, R. A.: *Dynamic probabilistic systems*, Bd. 1, 2. New York, London, Sydney, Toronto: John Wiley and Sons. 1971.

[2] Howard, R. A.: *Dynamische Programmierung und Markov-Prozesse*. Zürich: The Massachusetts Institute of Technology, Verlag industrielle Organisation. 1965.

[3] Haykin, S.: *Neural networks - A comprehensive foundation*. New Jersey: Prentice Hall Inc., 2. Aufl. 1999.

[4] Lunze, J.: Qualitative modelling of linear dynamical systems with quantised state measurements. *Automatica* 30 (1994), S. 417–431.

[5] Lunze, J.; Nixdorf, B.; Schröder, J.: A unified approach to the representation of discrete-time and discrete-event quantised systems. In: *European Control Conference*. Karlsruhe. 1999.

[6] Lunze, J.; Nixdorf, B.; Schröder, J.: Deterministic discrete-event representation of linear continious-variable systems. *Automatica* 35 (1999), S. 395–406.

[7] Rapp, M.; Reibel, J.: Gasanalytik mit Sensorsystemen: Ein Weg zur elektronischen Nase? *Nachr. Chem. Tech. Lab.* 44 (1996).

[8] Jerger, A.; Kohler, H.; Becker, F.; Keller, H. B.; Seifert, R.: Intelligent Sensor System for Reliable Monitoring of Ammonia Leakages. In: *8th International Meeting on Chemical Sensors*. Basel. 2000.

[9] Keller, H. B.: *Maschinelle Intelligenz*. Braunschweig, Wiesbaden: Vieweg Verlag. 2000.

[10] Kohonen, T.: *Self-organizing maps. Springer Series in Information Sciences*, Bd. 30. Berlin: SpringerVerlag. 1995.

[11] Hafner, S.: *Einsatz von Künstlichen Neuronalen Netzen zur Signalverarbeitung im Kraftfahrzeug am Beispiel spezifischer Motorsteuerungsprobleme*. Düsseldorf: Dissertation, VDI-Fortschritt-Bericht Nr. 349, Reihe 12, VDI-Verlag. 1998.

[12] Hafner, S.: Ein spezielles Neuronales Netz zur Merkmalsbildung für Klassifikatoren. *at - Automatisierungstechnik* 47(9) (1999), S. 421–428.

[13] Fallside, F.; Woods, W.: *Computer Speech Processing*. New Jersey: Prentice Hall Inc. 1985.

[14] Rabiner, L. R.: *Fundamentals of Speech Recognition*. New Jersey: Prentice Hall Inc. 1993.

[15] Baum, L. E.; Sell, G.: Growth Transformations for Functions on Manifolds. *Inequalities* 3,1-8 (1972).

[16] Viterbi, A. J.: Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Trans. Information Theory* 17-13, 260-9 (1967).

[17] Stiefelhagen, R.; Finke, M.; Yang, J.; Waibel, A.: From Gaze to Focus of Attention. In: *Visual Information and Information Systems, VISUAL '99* (Huijsmans, D. P.; Smeulders, A., Hg.), S. 761–768. Berlin: Springer Verlag. 1999.

[18] Laerhoven, K. V.: *On-line Adaptive Context Awareness starting from low-level Sensors*. Free University of Brussels: Dissertation. 1999.

[19] Isermann, R.: *Digitale Regelsysteme*. Berlin: Springer. 1987.

[20] Mikut, R.; Jäkel, J.; Gröll, L.: Automatic Design of Interpretable Membership Functions. In: *Proc., 8th Fuzzy Colloquium Zittau*, S. 103–111. 2000.

[21] Jäkel, J.; Gröll, L.; Mikut, R.: Bewertungsmaße zum Generieren von Fuzzy-Regeln unter Beachtung linguistisch motivierter Restriktionen. Berichtsband 8. Workshop *Fuzzy Control* d. GMA-FA 5.22, S. 15–28. 1998.

Formelzeichen

S: Anzahl der Sensoren

$\hat{\mathbf{x}}$: Zustandsvektor (Elemente wertekontinuierlich), entsteht aus $\mathbf{y}$ und dessen Ableitungen

$\hat{\mathbf{x}}_\mathbf{N}$: normierter Zustandsvektor

X_D: Zufallsgröße

x_D: diskreter Zustand (Werte: 1,2,...), Realisierung von X_D

$\mathbf{y}$: Messvektor

n: Ordnung der höchsten Ableitung des Messvektors, die in $\hat{\mathbf{x}}_\mathbf{N}$ enthalten ist

k: Abtastzeitpunkt

T: Abtastzeit

L: Größe des Lerndatensatzes

N: Anzahl der Knoten der Kohonen-Karte und damit Anzahl der diskreten Zustände der Markov-Kette

$\mathbf{r}_i$: Referenzvektor zum Knoten i im Eingangsraum

d_j: euklidischer Abstand des aktuellen normierten Zustandsvektors $\mathbf{x}_\mathbf{N}$ zum Referenzvektor $\mathbf{r}_j$ des Knotens j der Kohonen-Karte

H: Prognosehorizont

Kaskadierung und Anwendung hochdimensionaler Fuzzy-Controller

Nasredin Chaker, Rainer Hampel

Institut für Prozeßtechnik, Prozeßautomatisierung und Meßtechnik (IPM)
Hochschule Zittau/Görlitz
D-02763 Zittau, Theodor-Körner-Allee 16
Tel.: +49 (0)3583 61 1383
Fax: +49 (0)3583 61 1288
E-Mail: n.chaker@hs-zigr.de

1 Einführung

Die Anwendung der Fuzzy-Logik in der Prozeßautomatisierung und Prozeßdiagnose hat sich in den zurückliegenden Jahren fest etabliert. Während kleine Anwendungen in der Konsumgüterbranche (z. B. Kamera, Waschmaschinen usw.) ohne Vorbehalte akzeptiert werden, sind aus der Energie- und Verfahrenstechnik weniger Anwendungen bekannt. Ursachen dafür könnten sein:

- Die Anforderungen an Überwachung, Automatisierung und die Diagnose sind sehr komplex, anlagen- und prozeßspezifisch, so daß ein geringer Wiederholungsgrad für eine technische Realisierung gegeben ist.
- Es bestehen nach wie vor Vorbehalte gegen die Anwendung gehobener Methoden der Prozeßautomatisierung bei den Verfahrensträgern und Anwendern.
- Echtzeitanforderungen
- Software-Zuverlässigkeit

Die genannten Ursachen sind nicht vollständig und von gleicher Relevanz. Wie immer bei der Einführung neuer Techniken gibt es eine Phase der Euphorie sowie objektive und subjektive Hinderungsgründe.

Begriffe und Formulierungen wie

- linguistische Werte der Eingangsvariablen
- unvollständige Wissensbasis
- unbekannte physikalische und/oder analytische Modelle

führen dazu, daß Ingenieure die Fuzzy-Logik als

unscharfe Informationsverarbeitung

interpretieren und den reproduzierbaren Zusammenhang zwischen Eingangs- und Ausgangsvariablen in Frage stellen. Deshalb ist es notwendig zu betonen, die Fuzzy-Logik ist ein

Verfahren zur Verarbeitung unscharfer Signale und Relationen

mit reproduzierbaren Algorithmen, so daß für einen gleichen (ähnlichen) Eingangsdatensatz, daß die gleichen (ähnlichen) Ausgangsdatensätze ermittelt werden.

In dieser Beziehung wie auch im Rechenaufwand, unterscheidet sich die Fuzzy-Logik nicht von anderen gehobenen Methoden der Signalverarbeitung in der Prozeßautomatisierung.

Voraussetzung für die Erschließung weiterer Anwendungsgebiete sind die Bereitstellung von effizienten Methoden und Verfahren zur

Strukturauswahl

Parametrierung

Optimierung

für die Anwendung der Fuzzy-Logik in der Prozeßautomatisierung und Diagnose.

Mit dieser Zielstellung wurden am IPM an der Hochschule Zittau/Görlitz in den zurückliegenden Jahren Arbeiten zu folgenden Themen durchgeführt:

- Beschreibung der Fuzzy-Controller durch Kennfelder
- Minimierung der Freiheitsgrade für die Parametrierung und Optimierung
- Kaskadierung hochdimensionaler Fuzzy-Controller

Der folgende Bericht gibt die Ergebnisse der Untersuchungen in zusammenfassender Form wieder. Erfahrung mit der Anwendung dieser Ergebnisse liegen vor.

Fuzzy-Controller für

- Dampfturbinenregelung
- Positionsregelung für magnetisch gelagerte rotierende Wellen
- Neutralisation von Abwässern
- Optimale Fahrweise von Rauchgasentschwefelungsanlagen

Modellgestützte Meßverfahren für die Füllstandsmessung

- Fuzzy Modelling of Multidimensional Non-Linear Process - Design and Analysis of Structure [20]
- Fuzzy Modelling of Dynamic Non-Linear Processes Applied for Water Level Measurement [19]

Fuzzy-unterstützte Diagnosesysteme für

- Füllstandsmessung nach dem hydrostatischen Meßprinzip
- Zustandsdiagnose für magnetgelagerte rotierende Maschinen

2 Beschreibung des Fuzzy-Controllers durch Kennfelder

Klassische PID-Regler mit verschiedenen Ergänzungen (Adaption, Stör- und Führungsgrößenaufschaltung) haben eine hohe Akzeptanz in der Prozeßautomatisierung. Daraus resultiert das Bestreben, die Güte von Fuzzy-Controllern mit denen von klassischen PID-Reglern zu vergleichen [1], [4], [7], [11], [12], [13], [14], [17].

Bild 1 zeigt, daß der Fuzzy-Controller im Prozeß die gleichen Schnittstellen aufweist wie der klassische PID-Regler. Unabhängig von der gewählten Struktur wird die Dynamik durch die Differentation und Integration der Regelabweichung außerhalb des Fuzzy-Controller-Moduls bestimmt. Damit hat der Fuzzy-PID-Controller bereits drei Eingangsgrößen. Die Dimension erhöht sich noch, wenn zusätzlich Stör- und Führungsgrößen aufgeschaltet werden [18]. Damit wird das System unübersichtlich.

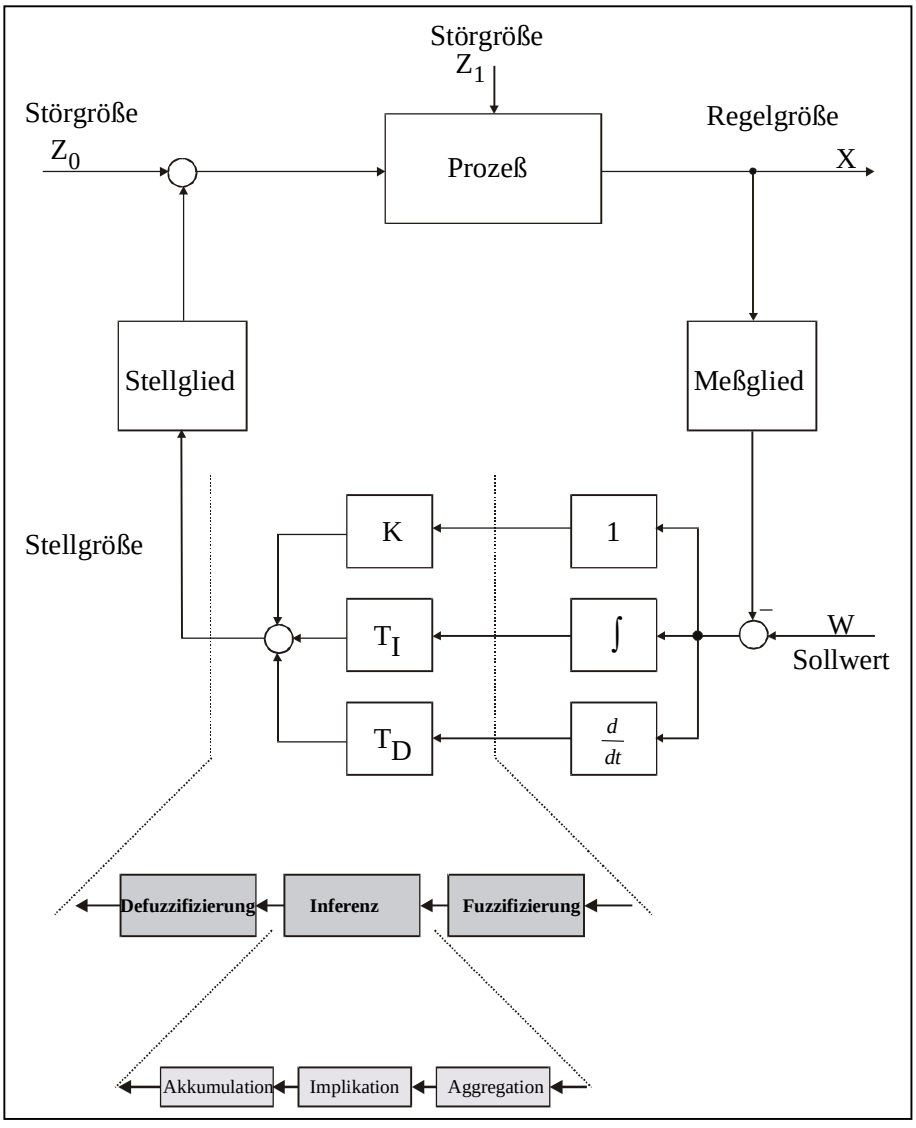


Bild 1: Einschleifiger Regelkreis mit Fuzzy-Controller

Die am IPM entwickelte Strategie basiert darauf, Multi Input-Singl Output (MISO) Systeme durch Strukturen mit zweidimensionalen Fuzzy-Controllern zu beschreiben, die einzeln parametriert und optimiert werden können.

Bild 2 zeigt beispielhaft einen Fuzzy-PI-Controller, für den die Basis-Regel gilt:

IF X1 AND X2 THEN Y (1)

X1 Regelabweichungsintegral

X2 Regelabweichung

Y Stellgröße

Das Kennfeld für den PI-Regler zeigt zwei Gebiete

- nicht deformiertes Kennfeld entsprechend dem klassischen festeingestellten PI-Regler,
- deformiertes Kennfeld entsprechend dem nichtlinearen Fuzzy-PI-Controller.

Für die Parametrierung sind die Werte

$Y1_{max}$, $Y2_{max}$ und Y_{max}

festzulegen. Die Optimierung erfolgt durch die Deformation des Kennfeldes. Mit Hilfe eines einfachen Beispieles für eine Durchflußmengenregelung mit einem adaptiven PI-Regler (kompressibles Medium) wird gezeigt, daß die erforderliche Deformation des Kennfeldes in engen Grenzen liegt und sehr gut aus Erfahrungen approximiert werden kann.

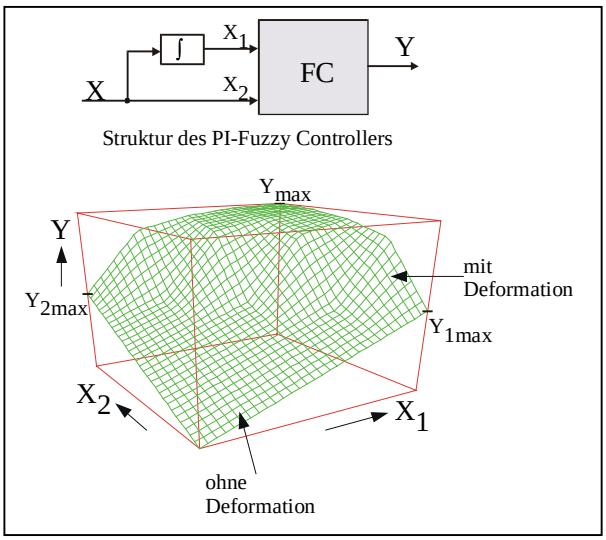
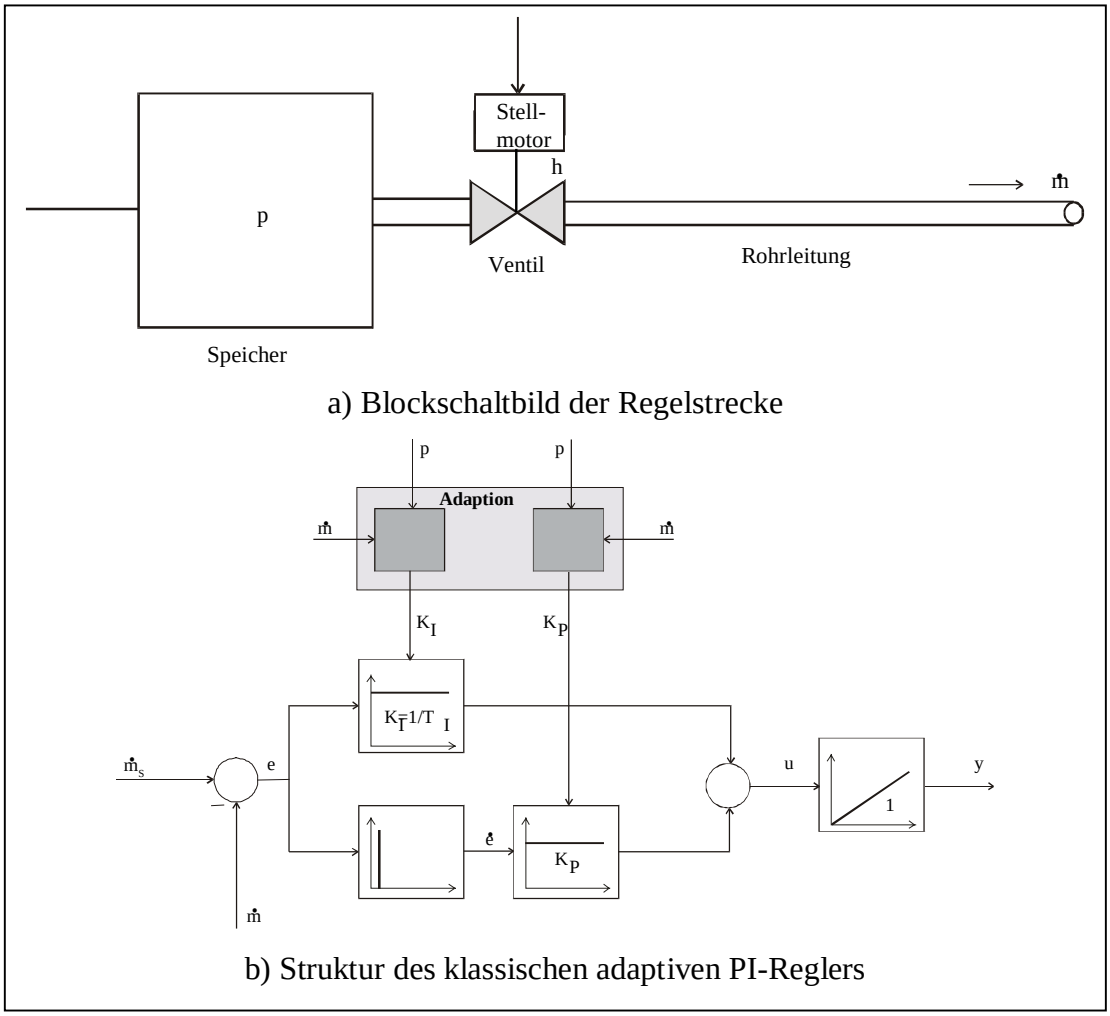


Bild 2 : Kennfeld eines nichtlinearen Fuzzy-PI-Reglers



a) Blockschaltbild der Regelstrecke

b) Struktur des klassischen adaptiven PI-Reglers

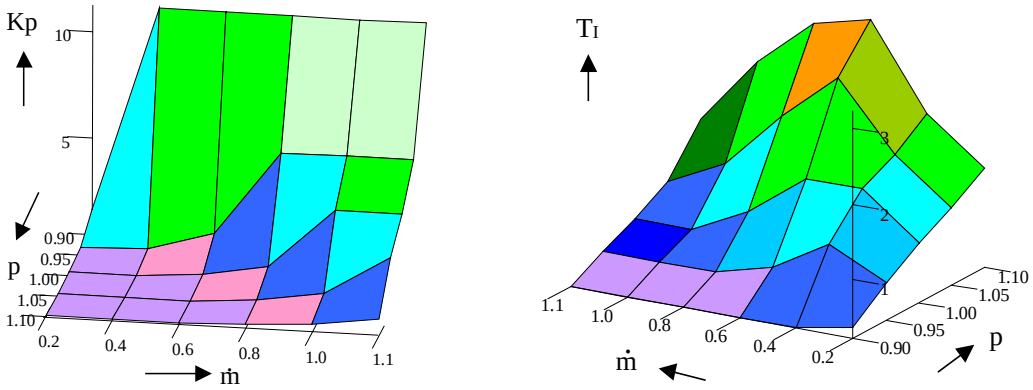
Bild 3: Struktur der Regelstrecke und des Reglers

Geht man davon aus, daß durch Adaption der Einstellwerte K_I und K_P entsprechend der klassischen Regeltheorie ein optimales Verhalten erreicht werden kann, müssen die Kennfelder

$$T_I = f_1(p, \dot{m}) \quad (2)$$

$$K_P = f_2(p, \dot{m}) \quad (3)$$

im Bild 4 eine Deformation aufweisen, die dieser Adaption entspricht. Für die Berechnung der Werte wurde die Methode des Betragsoptimums verwendet. Alle Eigenschaften des Controllers sind reproduzierbar im Kennfeld abgebildet.



a) Kennfeld der Verstärkung K_P b) Kennfeld der Integrationszeitkonstante T_I

Bild 4: Optimale Reglereinstellwerte (gemäß Betragsoptimum)

3 Freiheitsgrade für die Parametrierung und Optimierung zweidimensionaler Fuzzy-Controller

Für die Parametrierung und Optimierung des Fuzzy-Controller-Kennfeldes stehen zahlreiche Freiheitsgrade zur Verfügung (Tabelle 1). Für den Anwender ist diese Vielfalt eher verunsichernd als nützlich. Ergebnisse einer umfangreichen Sensibilitätsanalyse, die von Chaker [16] durchgeführt wurde, zeigen, daß durch Festlegung einiger Randbedingungen für viele Anwendungen die Zahl der freien Parameter auf den Wert „2“ reduziert werden kann.

Folgende Gütekriterien für die Form und Deformation des Kennfeldes werden verwendet:

- Differenzierbarkeit (gleitende Übergänge bei veränderlichen Übertragungsfaktoren)
- geringe Welligkeit
- Deformierbarkeit (maximal erforderliche Abweichung vom linearen Kennfeld).

Tabelle 1: Freiheitsgrade für die Optimierung von Fuzzy-Controllern

Fuzzifizierung (ZGF)	**Inferenz (Operator)**	**Defuzzifizierung**
• Anzahl der Sets • Form der Sets • Verteilung symmetrisch unsymmetrisch äquidistant nicht-äquidistant • Überlappung • Spreizung linke Spreizung rechte Spreizung	• T-Norm Min Prod Bounded Difference • S-Norm Max Sum Bounded Sum	• Center-Of-Gravity • Singleton • Maximum • Mean of Maximum

ZGF - Zugehörigkeitsfunktion

Differenzierbarkeit und Welligkeit wird von der Kombination Operator-Zugehörigkeitsfunktion wesentlich bestimmt. Gut geeignet sind:

Operator: Sum-Prod für λ-Sets

Operator: Max-Min für Gauß'sche Sets

Für diese Fälle ist es ausreichend, die Zahl und die Verteilung der Fuzzy-Sets für die Deformation des Kennfeldes zu variieren.

Das optimale Verhalten des Fuzzy-Controllers ist abgebildet in der Deformation des Kennfeldes.

Die Bilder 5 bis 7 demonstrieren diese Ergebnisse. Bild 5 zeigt als Vergleichsbasis ein lineares Kennfeld, das durch λ-Sets, den Operator Sum-Prod und die Singleton-Methode für die Defuzzifizierung erzeugt wurde. Dieses Kennfeld erfüllt auch alle o. g. Gütekriterien.

Durch Veränderung der Verteilung des Sets (Bild 6) bei unveränderlichem Überlappungsgrad wird eine Deformation des Kennfeldes in der Art erreicht, daß im Zentrum der Übertragungsfaktor geringer ist als am Rand. So können Toleranzbänder erzeugt werden. In umgekehrter Weise kann durch Spreizung der Sets für Y eine Erhöhung des Übertragungsfaktors realisiert werden.

Bild 7 zeigt eine Variante, die für eine optimale Regelung ungeeignet ist infolge einer hohen Welligkeit. Im ungünstigsten Fall treten zahlreiche lokale Maxima und Minima auf und es kann eine Vorzeichenumkehr für den Übertragungsfaktor auftreten. Damit treten lokale bandbegrenzte Instabilitäten auf. Das Gütekriterium der Welligkeit ist nicht erfüllt.

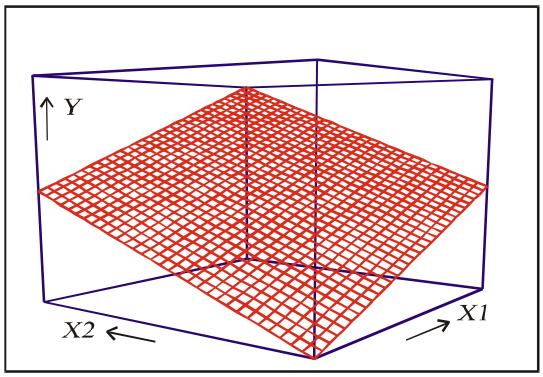


Bild 5: Lineares Kennfeld eines Fuzzy-Controllers

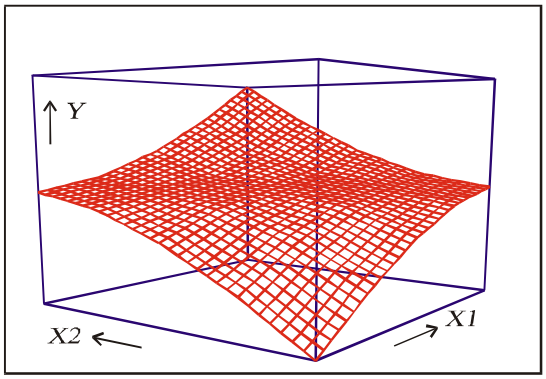


Bild 6: Deformiertes Kennfeld eines Fuzzy-Controllers

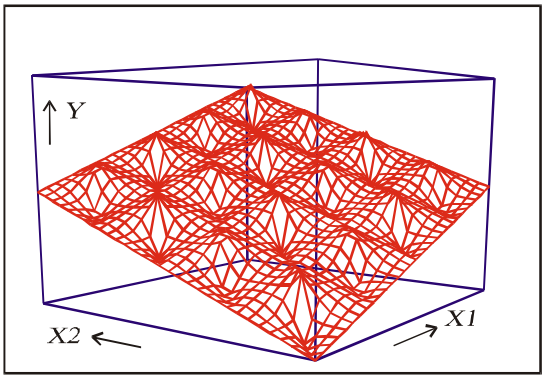


Bild 7: Nicht-optimale deformation des Kennfeldes eines Fuzzy-Controllers

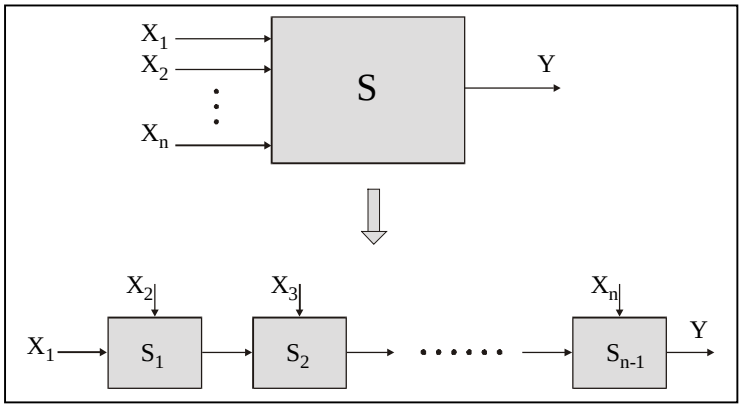
4 Kaskadierung hochdimensionaler Fuzzy-Controller

Die Strukturtransformation für hochdimensionale Fuzzy-Controller kann in zwei Richtungen erfolgen:

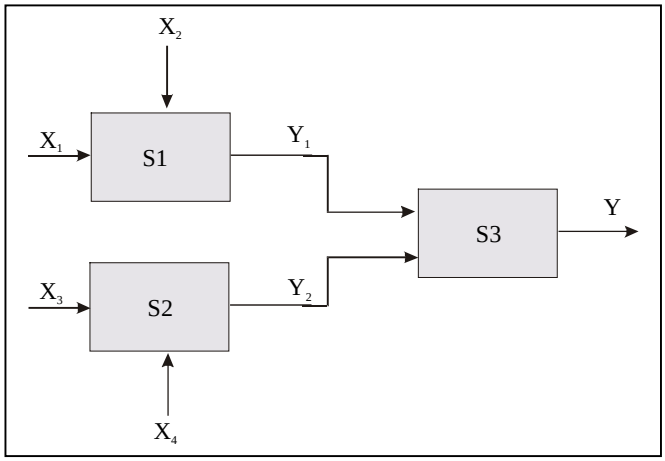
- kaskadierte Fuzzy-Controller,
- hierarchische Fuzzy-Controller.

Wie aus Bild 8 ersichtlich, bestehen beide Strukturen aus zweidimensionalen Fuzzy-Controllern. Der hierarchische Controller ist ein Sonderfall des kaskadierten Controllers, deshalb wird im weiteren nur der kaskadierte Controller behandelt.

a) Hochdimensionaler Fuzzy-Controller



b) Kaskadierter Fuzzy-Controller



c) Hierarchischer Fuzzy-Controller

Bild 8: Strukturen von Fuzzy-Controllern

Für einen mehrdimensionalen Controller gilt

Basis-Regel

IF [OP X_i] THEN Y $i \rightarrow 1, n$ (4)

Einzel-Regel

IF [OP X_{ij}] THEN Y_k $i \rightarrow 1, n$ (5)

$j \rightarrow 1, m$

$k \rightarrow 1, r$

Die Zahl der Einzelregeln R_m für die vollständige Beschreibung des Controllers beträgt

$$R_m = \prod_{i=1}^{n} m_i \qquad (6)$$

n Zahl der Eingangsvariablen

m_i Zahl der Sets der i-ten Eingangsvariable

r Zahl der Sets der Ausgangsvariable Y

Für einen 4-dimensionalen Fuzzy-Controller mit 5 Sets je Eingangsgröße ergeben sich 625 Einzelregeln. Diese Zahl ist sehr groß, die Regelmatrix wird unübersichtlich. Daraus leiten sich die Forderungen ab für

- Reduzierung der Zahl der Einzelregeln,
- Erhöhung der Übersichtlichkeit.

Diese Forderungen werden durch die Kaskadierung erfüllt.

Für den o.g. 4-dimensionalen Fuzzy-Controller ergibt sich eine Struktur nach Bild 8b mit drei hintereinander geschalteten 2-dimensionalen Fuzzy-Controllern. Die Zahl der notwendigen Regeln beträgt in diesem Fall nur noch 75.

Die Voraussetzung für die Zulässigkeit der Transformation ist die Erfüllung des Assoziativ-Gesetzes.

$$X1 \wedge X2 \wedge X3 \wedge X4 = ((X1 \wedge X2) \wedge X3) \wedge X4 \qquad (7)$$

Das bedeutet auch, daß die Einzelregeln des mehrdimensionalen Controllers mit denen des kaskadierten Controllers übereinstimmen müssen.

Als Grundlage der physikalisch-technisch begründeten Kaskadierung sind folgende Vereinbarungen notwendig:

Typ der Basis-Regeln

Typ 1 Basis-Regeln zum Beschreiben des Übertragungsverhaltens des Controllers

Typ 2 Basis-Regeln zur Adaption des Übertragungsverhaltens des Controllers durch Veränderung der Regelbasis und/oder freier Optimierungsparameter

Typ der Eingangsvariablen

Dominante Eingangsvariable

Eingangsvariable in starkem Einfluß auf die Form (Deformation) des Kennfeldes

Nichtdominante Eingangsvariable

Eingangsgrößen mit geringem Einfluß auf die Deformation des Kennfeldes

Der Demonstration der Ergebnisse dient Bild 9. X1 und X2 sind dominierende Eingangsgrößen des Typs 1. X3 ist eine nichtdominierende Eingangsgröße. Ihr Einfluß auf den Controller wird durch die Adaptionsregeln

IF	X3 = L	Verschiebung von Y1 in L-Richtung
IF	X3 = H	Verschiebung von Y in H-Richtung

Der Vergleich der Regelmatrix für den vollständigen dreidimensionalen Controller (Bild 9a) mit dem kaskadierten Controller (Bild 9b) zeigt Nichtübereinstimmungen an zwei Stellen (markiert in Bild 9a und 9b). Durch Hinzufügen von zwei Sets zur virtuellen Zwischengröße Y_{v1} kann vollständige Übereinstimmung erreicht werden (Bild 9c).

Die unter diesen Bedingungen mögliche Reduzierung der Zahl der Einzelregeln und der Zahl der Rechenoperationen UND und ODER zeigt Bild 10.

5 Zusammenfassung

Regelbasierte Systeme für Fuzzy Control, Fuzzy Diagnose und Überwachung sind nichtlineare Mehrgrößensysteme. Das Ergebnis der Signalverarbeitung innerhalb des Systems ist ein hochdimensionales Kennfeld. Für die Optimierung des Systemverhaltens gibt es eine Vielzahl von Freiheitsgraden. Die Rekonstruktion der Wissensbasis vom Kennfeld als Ausgangspunkt ist unmöglich.
Mit diesem Hintergrund wurde im Beitrag die Güte des Kennfeldes eines zweidimensionalen Fuzzy-Controllers im Zusammenhang mit der erforderlichen Deformation des Kennfeldes zur Kompensation von nichtlinearen Effekten beschrieben. Für die Optmierung des Controller-Verhaltens mit Hilfe der Deformation des Kennfeldes sind nur zwei Freiheitsgrade erforderlich.
Mit Hilfe dieser Erfahrung wurde der hochdimensionale Controller kaskadiert. Eine subjektive Entscheidung dafür ist die Unterscheidung des Typs der Eingangsgrößen in dominante, nicht-dominante und Optimierungsvariablen erforderlich. Die Güte der Kaskade ist vom endgültigen Kennfeld und der Vollständigkeit der Regelbasis abhängig. Anhand eines anschaulischen Beispiels wurden die Effekte der Kaskadierung demonstriert.

Y_M				X_1		
				L	N	H
X_3	L	X_2	L	L	L	L
			N	L	L	N
			H	L	N	H
	N		L	L	L	N
			N	L	N	H
			H	N	H	H
	H		L	L	N	H
			N	N	H	H
			H	H	H	H

a) Vollständige Regelmatrix für einen 3-dimensionalen Fuzzy-Controller

Y_{V1}		X_1		
		L	N	H
X_2	L	L	L	N
	N	L	N	H
	H	N	H	H

Y_C		X_3		
		L	N	H
Y_{V1}	L	L	L	N
	N	L	N	H
	H	N	H	H

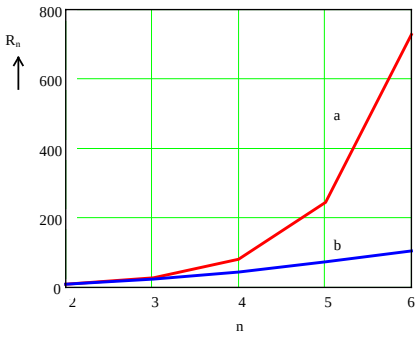
b) Vollständige Regelmatrizen für den kaskadierten Fuzzy-Controller

Y_{V1}		X_1		
		L	N	H
X_2	L	VL	L	N
	N	L	N	H
	H	N	H	VH

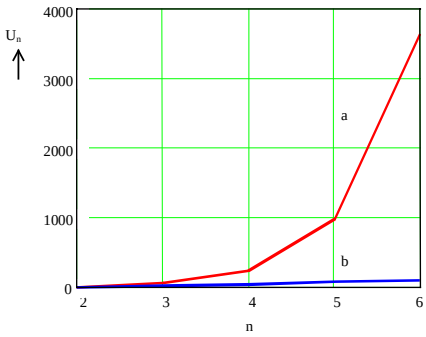
Y_C		X_3		
		L	N	H
Y_{V1}	VL	L	L	L
	L	L	L	N
	N	L	N	H
	H	N	H	H
	VH	H	H	H

c) Regelmatrizen für den verbesserten 3-dimensionalen kaskadierten Fuzzy-Controller

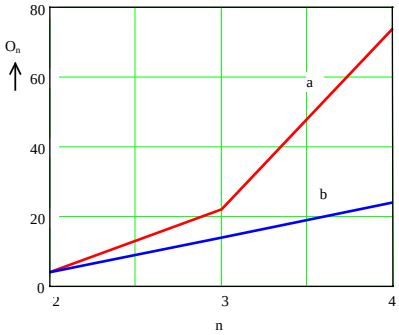
Bild 9: Kaskadierung eines 3-dimensionalen Fuzzy-Controllers



Anzahl der Regeln



Anzahl der UND-Operationen



Anzahl der ODER-Operationen

Bild 10: Vergleich der Anzahl der Regeln und Operator-Operationen zwischen dem a) hochdimensionalen und dem b) kaskadierten Fuzzy-Controller

n : Anzahl der Eingangsvariablen m : Anzahl der Sets einer Eingangsvariable

6 Literatur

[1] Kuccera, T.: *Hierarchical Fuzzy Controllers (Conventional PID controller and Fuzzy Logic Controllers FLC).* Conference on Fuzzy Logic in Engineering and Natural Sciences, Zittau, September 25 - 27, 1996, Proceedings pp. 80 - 82

[2] Hampel, R.; Chaker, N.: *Structure Analysis for Fuzzy-Controller.* Conference on Fuzzy Logic in Engineering and Natural Sciences, Zittau, September 25 - 27, 1996, Proceedings pp. 83 - 90

[3] Steinkogler, A.; Koch, J.: *Genetic programming designs hierarchie Fuzzy Logic Controller.* Conference on Fuzzy Logic in Engineering and Natural Sciences, Zittau, September 25 - 27, 1996, Proceedings pp. 150 - 159

[4] Pivonka, P.; Breijl, M.: *Use of PID Controllers in Fuzzy Control of coal power plants.* Conference on Fuzzy Logic in Engineering and Natural Sciences, Zittau, September 25 - 27, 1996, Proceedings pp. 441 - 448

[5] Czogala, E.; Leski, J.: *On Destructive Fuzzy Logic Controllers.* 5th Zittau Fuzzy-Colloquium, September 4 - 5, 1997, pp. 8 - 12

[6] Hampel, R.; Chaker, N.: *Cascading of Multi-Dimensional Fuzzy Controllers.* 5th Zittau Fuzzy-Colloquium, September 4 - 5, 1997, pp. 17 - 31

[7] Vogrin, P.; Halang, W. A.: *Approximation of Conventional Controllers by Fuzzy Controllers with Equal Discribing Functions.* 5th Zittau Fuzzy-Colloquium, September 4 - 5, 1997, pp. 51 - 65

[8] Czogala, E.; Henzel, N.; Leski, J.*: The Equality of Inference Results Using Fuzzy Implication and Conjunctive Interpretations of the IF-THEN Rules under Defuzzification.* 6th Zittau Fuzzy-Colloquium, September 3 - 4, 1998, pp. 1 - 6

[9] Halang, W. A; Colnaric, M.; Vogrin, P.: *Safety Licensable Inference Controller.* 6th Zittau Fuzzy-Colloquium, September 3 - 4, 1998, pp. 18 - 23

[10] Wagenknecht, M.; Chaker, N.: *Towards an Algorithmic Cascading of Fuzzy Rules.* 6th Zittau Fuzzy-Colloquium, September 3 - 4, 1998, pp. 56 - 61

[11] Pivonka, P.; Sidlo, M.: *Fuzzy PI + PD Controller with a Normalised Universe - The Exact Solution for Setting of Parameters.* 6th Zittau Fuzzy-Colloquium, September 3 - 4, 1998, pp. 62 - 67

[12] Arakeljan, E.; Panko, M.; Usenko, V.: *Comperative Analysis of Classical and Fuzzy PID Algorithms.* 6th Zittau Fuzzy-Colloquium, September 3 - 4, 1998, pp. 68 - 73

[13] Pacyna, K.; Pieczynski, A.: *Influence of Changes of Membership Function on PID Fuzzy Logic Control.* 6th Zittau Fuzzy-Colloquium, September 3 - 4, 1998, pp. 80 - 85

[14] Rotach, V.: *On Connection Between Traditional and Fuzzy PID Regulators.* 6th Zittau Fuzzy-Colloquium, September 3 - 4, 1998, pp. 86 - 90

[15] Hampel, R.; Keil, A.; Gierth, L.: *Fuzzy Drehzahlregelung.* atp 3/99, S. 37 - 42

[16] Hampel, R.; Chaker, N.: *Minimizing the number of variable parameters for optimizing the Fuzzy Controller.* Fuzzy Sets and Systems 100(1998), pp. 131 - 142

[17] Hampel, R.; Chaker, N.; Gierth, L.: *Adaptive Dampfturbinenregelung mit Fuzzy Logik zur Beherrschung von Lastabwürfen in Heizkraftwerken.* atp 2/98, S. 42 - 49

[18] Hampel, R.; Chaker, N.: *Application of Fuzzy Logic in Control and Limitation Systems Using Industrial Hardware.* Mendel 97, Proceedings pp. 291 - 298, ISBN 80-214-0884-7

[19] Traichel, A.; Kästner, W.; Hampel, R.: *Fuzzy Modeling of Dynamic Non-Linear Processes - Applied for Water Level Measurement.* 7th Zittau Fuzzy Colloquium, September 8 - 10, 1999, pp. 119 - 134

[20] Pieczynski, A.; Kästner, W.; Hampel, R.: *Fuzzy Modeling of Multidimensional Non-Linear Processes - Design and Analysis of Structures.* 7th Zittau Fuzzy Colloquium, September 8 - 10, 1999, pp. 85 - 101

[21] Hampel, R.; Chaker, N.; Stegemann, H.*: High Speed Matrix Controller for Safety Related Applications.* Mendel '99, 5th International Conference on Soft Computing, June 9 - 12, 1999, Brno, Czech Republic, pp. 243 - 248

[22] Wagenknecht, M.; Hampel, R.; Stemberk, P.: *On Fuzzy Arithmetic Operations.* Mendel '99, 5th International Conference on Soft Computing, June 9 - 12, 1999, Brno, Czech Republic, pp. 299 - 304

Trainierbarer Neuro-PID-Regler für hohe Regelgüte

Ulrich Lehmann, Stefan Dormeier, Manfred Büchel, Dietrich Peters, Udo Reitz, Egon Weiner

Märkische Fachhochschule Iserlohn, FH Bielefeld, FH Gelsenkirchen, EUREGIO Neuro-Fuzzy-Centrum

Tel.: (0049) -(0)2371/566- (0) -180, Fax.: (0049) -(0)2371/36564,

Kontaktaufnahme per E-Mail: nfl@wwwfbp.mfh-iserlohn.de

http://www.mfh-iserlohn.de/Verbunde/NFL

Kurzfassung

Der entwickelte adaptive Neuro-PID-Regler kann wie ein Sportler auf hohe Regelgüte trainiert werden. Dies ist ein Ergebnis des vom Ministerium für Schule, Weiterbildung, Wissenschaft und Forschung (MSWWF) geförderten Forschungsverbundes "Neuronale Fuzzy-Logik NRW", in dem 6 NRW-Fachhochschulen zusammenarbeiten. Die Sprecherfunktion für den Forschungsverbund hat die MFH Iserlohn. Weitere Arbeitsgebiete sind die Mustererkennung und Data Mining mittels Neuro-Fuzzy-Logik (NFL).

In enger Kooperation zwischen den Fachhochschulen Bielefeld, Gelsenkirchen, Iserlohn und Münster wurde der adaptive, trainierbare Regler entwickelt. Anwendungsbereiche sind Temperaturregelungen bei verfahrenstechnischen Prozessen, z.B. in der Kunststoffverarbeitung und in der Gebäudeautomatisierung. In diesem Beitrag soll näher auf die Anwendung in Kunststoffverarbeitungsmaschinen eingegangen werden.

Temperaturregelungen an Kunststoffverarbeitungsmaschinen gehören zu jener Klasse von Prozessen, die sich aufgrund ihrer komplexen inneren Struktur nur bedingt mathematisch-physikalisch beschreiben lassen, so dass Reglereinstellungen und -optimierung zu einem erheblichen Teil auf antrainiertem Erfahrungswissen der Bediener beruhen. Oftmals treten

während der Produktion Störeinflüsse und Veränderungen der Regelstrecke auf, die zu inakzeptablen Verschlechterungen der Produktgüte führen können [1].

Der neu entwickelte adaptive Neuro-PID-Regler ist in der Lage, sich selbsttätig auf unterschiedliche Situation im Produktionsprozess einzustellen und sein Verhalten im Sinne der Erzielung einer optimalen Prozessgüte zielgerichtet zu trainieren. Er arbeitet als "Hochleistungssportler", der seine Ergebnisse durch ständiges Training verbessert. Dabei ist er achtsamer und zuverlässiger als ein menschlicher Experte.

Zur Modellierung der Adaptionsstrategie wurde ein mehrschichtiges Künstlich Neuronales Netz (KNN) mit überwachtem Lernen eingesetzt; als Regelalgorithmus ein standardmäßiger PID-Algorithmus.

Die Regler-Software wurde mit einer standardisierten, regelungstechnischen CAE-Umgebung erstellt. Aus dem Makrocode wurde C-Code generiert. So kann der Neuro-PID-Regler mit geringem Aufwand auf verschiedene Hardwareplattformen wie beispielsweise Industrie-PCs oder Industrieregler auf Mikrocontroller-Basis portiert werden. Darüber hinaus wurde eine Portierung auf SPS vorgenommen.

Die Adaption bewährt sich auch bei Parametervariation der Regelstrecke über den Lernbereich des KNN hinaus.

1 Modellbildung der Regelstrecke

Die Modellierung der Regelstrecke erfolgte mit WinFACT98. Die Regelstrecke ist nichtlinear und zeigt P-T_4-Verhalten. Der Einfluss der Nichtlinearitäten der Strecke, gegeben durch die Stellgrößenbegrenzung und die Quantisierung von Stell- und Regelgröße ist aus folgender Sprungantwort abzuschätzen.

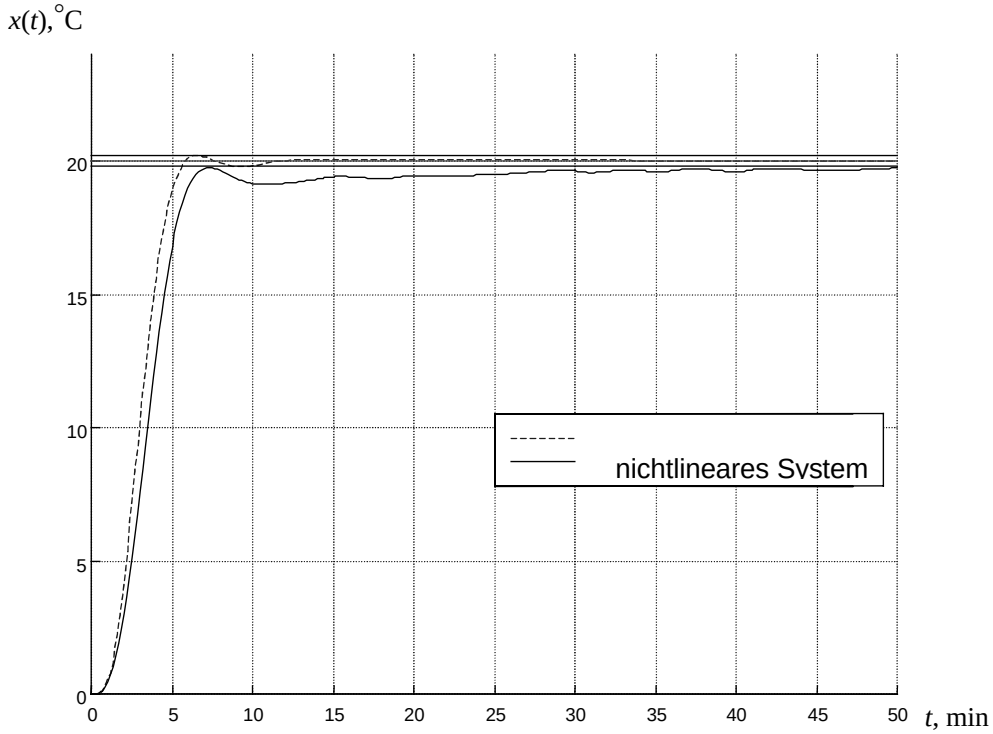


Abb 1: Einfluss der Nichtlinearitäten der Regelstrecke auf die Regelgüte des geschlossenen Kreises mit konstanten, für den linearen Fall dimensionierten, Reglerparametern

In [1] sind Regelstrecke und Regelkreis bereits beschrieben worden. Hier eine Zusammenfassung der wichtigsten Daten:

- die Extruderheizzone als Regelstrecke kann durch ein PT_4 - Glied simuliert werden, wobei der Proportionalbeiwert K_s = 0,057 °C/W und die Zeitkonstanten T_1 =17,5 min, T_2 = 1,6 min, T_3 = 0,65 min, T_4 = 0,25 min betragen,
- während des Betriebes können sich die Streckenparameter K_s und T_1 ändern, der Variationsbereich der Parameter beträgt für K_s ± 40 %, für T_1 ± 20%,
- Führungsgrößenänderungen liegen in einem Bereich von ±1°C bis ±20 °C, ausgehend von einem Arbeitspunktwert von 200 °C,
- die eingesetzten Industrie - Temperaturregler haben eine Stellgrößenbegrenzung von 0 V bis 10 V,

- die Abtastzeit des Reglers beträgt 10 Sekunden,
- Wertediskretisierung erfolgt in AD- und DA-Umsetzern mit einer effektiven Auflösung von 9 Bit.

Die oben aufgeführten technischen Daten der Strecke, insbesondere der Streubereich der Parameter K_s und T_1, haben einen großen Einfluss auf die zeitlichen Verläufe der Regelgröße. Abb.1 zeigt die Sprungantworten eines linearen Regelkreises mit optimalen Reglereinstellungen K_p = 12,2; T_n = 36,1 min; T_v = 1,51 min und des nichtlinearen mit denselben Einstellungen. Der Arbeitspunkt der Regelung liegt bei X_{AP} = 200°C. Es wurde ein Führungsgrößensprung mit w_0 = 20°C auf das System gegeben. Die Sprungantwort zeigt nur die relative Änderung der Regelgröße x(t) um den Arbeitspunkt. Das zulässige Toleranzband für x(t) beträgt ±0,2°C und ist damit relativ schmal. Die Einschwingzeit hat sich erheblich vergrößert von 337 s beim linearen System auf 3400 s beim nichtlinearen. Wesentlich in diesem Zusammenhang ist auch die Zeitvarianz der Streckenverstärkung K_s und der größten Zeitkonstanten T_1.

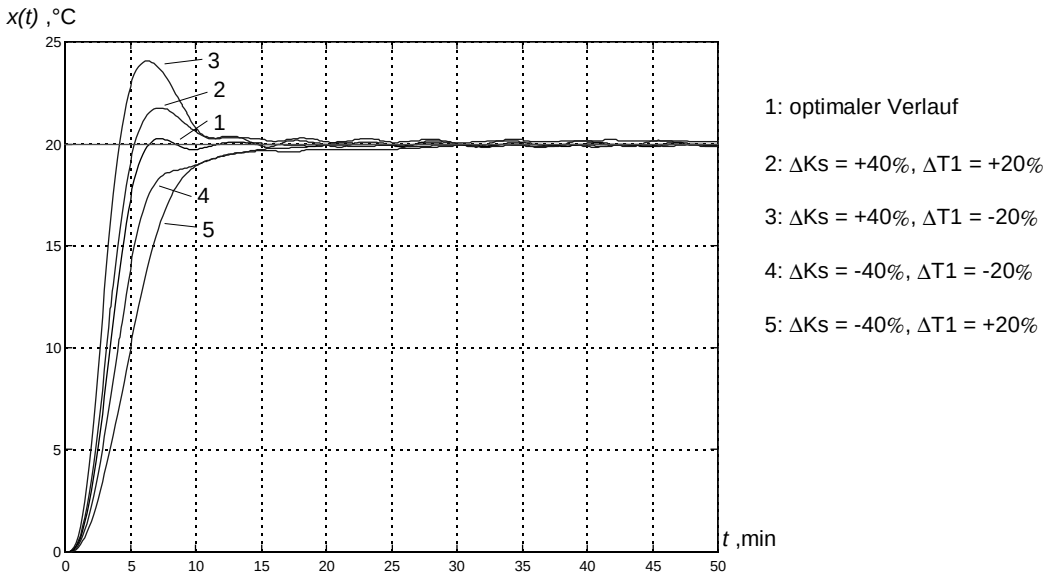


Abb 2: Einfluss der Zeitvarianz von K_s und T_1 auf die Regelgüte für Führungsverhalten

Die Änderungen in den Sprungantworten bei Streckenverstimmung sind in Abb.2 gezeigt. Das Toleranzband ±0,2°C wird durch die große Überschwingweite Δx für den Fall 2 und 3 erheblich überschritten.

Der Optimale Verlauf im nichtlinearen Regelkreis mit Reglereinstellungen: K_p = 17,3; T_n = 21,0 min; T_v = 1,71 min hat eine Überschwingweite x_m = 0,25°C, Anregelzeit t_{an} = 376 s, Einschwingzeit t_{aus} = 642 s. Durch die Streckenverstimmung erreicht das maximale Überschwingen den Wert 4°C bei K_s = 0,0798, T_1 = 14 min. Die Anregelzeit beträgt im ungünstigsten Fall 1650 s bei K_s = 0,0342, T_1 = 14 min, die Einschwingzeit 1710 s bei K_s = 0,0798, T_1 = 21 min.

Konstante Reglerparameter eignen sich auch schlecht bei Führungsgrößenänderungen, wie man in Abb. 3 sehen kann. Die Überschwingweite für die Führungsgrößenänderung w_o = 5 °C beträgt z.B. 1,9 °C oder 38%, die Einschwingzeit 1190 s. Für w_o = 2 °C ist das maximale Überschwingen 1,5 °C, oder 75 %, dabei entsteht eine Arbeitsschwingung, deren Amplitude deutlich größer als der zulässige Toleranzbereich von ±0,2 °C ist.

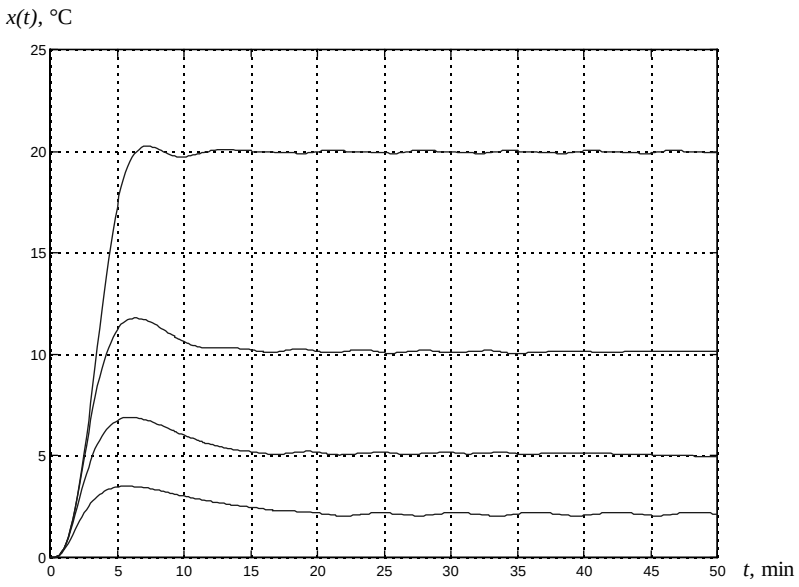


Abb 3: Einfluss der Nichtlinearität und der Führungsgrößenänderung auf die Regelgüte für Führungsverhalten

Diese Beschreibung soll zur Charakterisierung der Regelstrecke genügen. Es ist verständlich, dass in diesem Fall mit einem digitalen Standard-PID-Regler nicht die geforderte Regelgüte

hinsichtlich der Überschwingweite bzw. des schnellen Einschwingens der Temperatur auf die Führungsgröße erreichbar ist.

2 Neuro-PID-Architektur

Nach einigen Voruntersuchungen wurde von der MFH Iserlohn eine Struktur gewählt, die zur Adaptation des Reglers einen Identifikationsblock und ein Künstlich Neuronales Netz (KNN) enthält [2].

Im Identifikationsteil sollen anhand der im laufenden Betrieb gemessenen Ein- und Ausgangssignale der Regelstrecke die Parameter Streckenverstärkung K_s, maximale Überschwingweite Δx_m, Anregelzeit T_{AN} und Ausregelzeit T_{aus} bestimmt werden. Auf eine Identifikation von T_1, der maximalen Streckenzeitkonstanten, zur Laufzeit wurde bei diesem Ansatz, im Gegensatz zu [1], bewusst verzichtet. Zur Erinnerung sei erwähnt, dass eine leichte Portierung auf SPS eine Forderung im Projekt war. Das zeitliche Verhalten der Strecke wird Indirekt durch die Anregelzeit T_{AN} und Ausregelzeit T_{aus} für das KNN erfassbar.

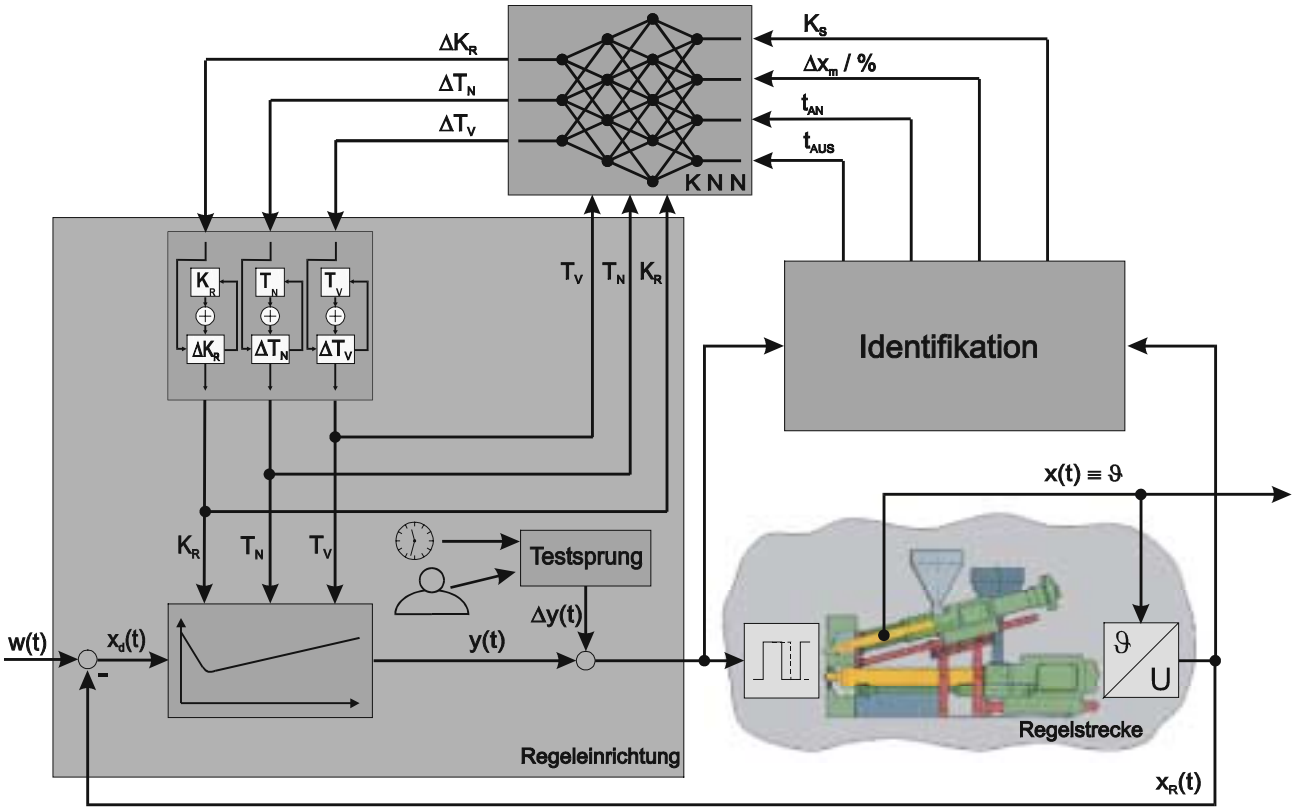


Abb. 4: Architektur des adaptiven Neuro-PID-Reglers

Die genannten Werte, zusammen mit dem Führungsgrößensprung w_0 und der aktuellen Reglereinstellung K_p, T_n, T_v werden dann während der Produktionsphase in das KNN eingelesen, welches Korrekturwerte ΔK_p, ΔT_n und ΔT_v für die Adaption der Reglerparameter K_{pneu}, T_{nneu}, T_{vneu} liefert. Das KNN liefert also nur die Δ-Werte für die Regelparameter. Dies ist für die Funktion der Adaption außerhalb des Bereiches des Trainingsdatensatzes wesentlich. Abb. 4 zeigt die Architektur des adaptiven Neuro-PID-Reglers Reglers.

3 Identifikation und Adaption

Für die Bestimmung der Korrekturwerte der Reglerparameter ($\Delta K_R, \Delta T_N, \Delta T_V$) kommt ein Künstlich Neuronales Netz (KNN) vom Typ Multilayer-Perceptron (MLP) zum Einsatz. Die Eingangswerte für das KNN werden aus den Güteparameter (Δx_m, T_{AN}, T_{AUS}) errechnet. In einem unter WinFACT98 entwickelten Identifikationsblock (Abb. 5) werden die Güteparameter aus einer Sprungaufschaltung ermittelt und in das KNN eingelesen.

Die Trainingsdaten wurden aus etwa 40 Simulationsversuchen berechnet und mit Hilfe von Microsoft Excel für das KNN aufbereitet.

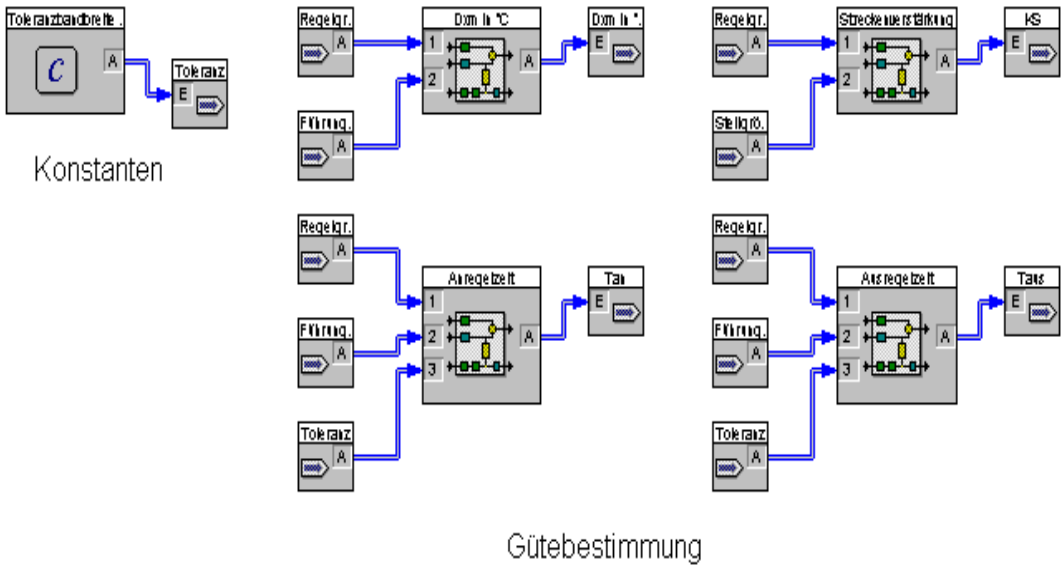


Abb. 5: Blockschaltbild der Identifikation

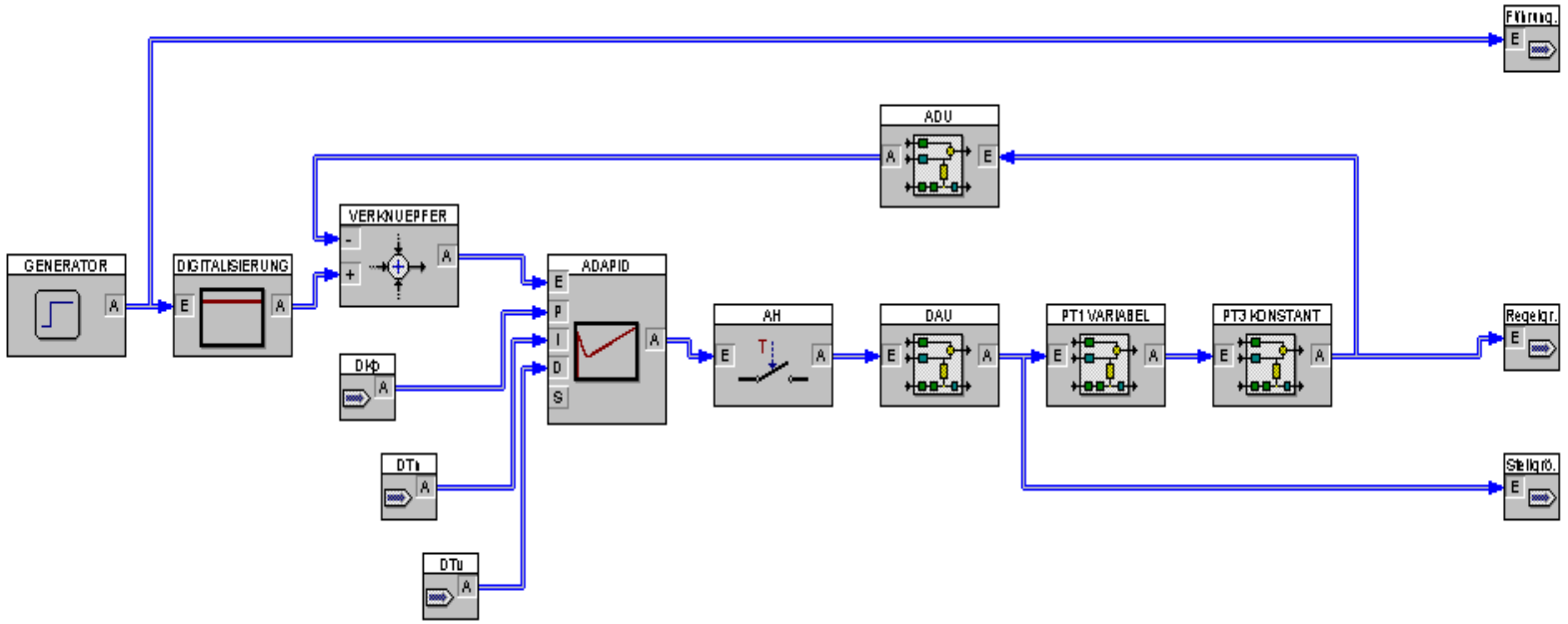


Abb. 6: Modell der Regelstrecke mit Regler

In Abb. 6 die Strecke mit dem Adaptiven-PID-Regler abgebildet. Es ist lediglich die Schnittstelle für die Übergabe der neuen Reglerparameter erkennbar. In Abb. 7 ist das KNN mit seinen 7 Eingängen und 3 Ausgängen gezeigt. Auf der linken Seite des KNN werden die errechneten Güteparameter aus den oben abgebildeten Blöcken eingelesen. Als weitere Eingangsgröße wird die Streckenverstärkung K_s in das KNN eingegeben. Weitere drei Eingänge werden für die Reglerparameter K_p, T_n und T_v des jeweiligen Arbeitspunktes (aktuelle Reglereinstellung) benötigt. Die Ausgabeblöcke (rechts) geben die neuen Reglerparameter in Form von Delta-Werten aus. Die Anbindung an den PID-Regler erfolgt so, dass die vom KNN errechneten Reglerparameter-Korrekturwerte (reelle Gleitpunktzahlen) zu den aktuellen Reglerparametern hinzu addiert werden.

Die Erfassung und Identifizierung der veränderlichen Streckenparameter im laufenden Betrieb erfolgt für Tan und Taus sowie über die Streckenverstärkung. Das KNN wurde mit etwa 160 Trainingsdatensätzen trainiert.

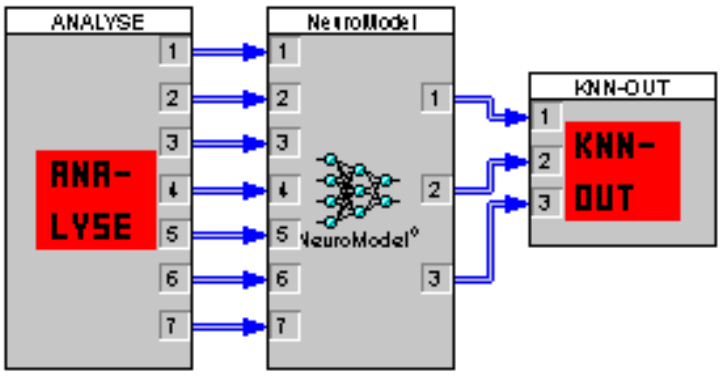


Abb. 7a: KNN mit den Eingängen aus der Identifikation und den Ausgängen für die Parameterkorrektur des Reglers

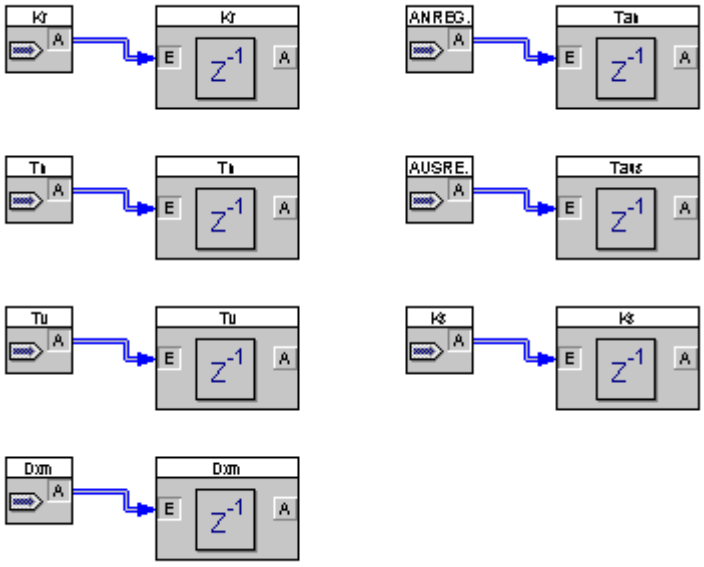


Abb. 7b: KNN mit den Eingängen aus der Identifikation und den Ausgängen für die Parameterkorrektur des Reglers

4 Optimierung der Reglerparameter / Trainingsdaten

Zur Optimierung der Reglereinstellung wurden Evolutionsstrategien [7;10] genutzt. Anders als in [1] wurden hier die Optimierungsverfahren eingesetzt, die von der Simulationsumgebung WinFACT bereitgestellt werden [3,4]. So wurde eine numerische Optimierung der Regelparameter vorgenommen. Mit hoher Zuverlässigkeit lässt sich auch bei sehr zerklüfteten Topologien der Zielfunktion das globale Optimum bestimmen. Zur Anwendung kam die Optimierung nach dem ITAE-Kriterium (zeitgewichtete betragslineare Regelfläche)[6]. Diese Aussagen werden von den praktischen Ergebnissen bestätigt.

Das Optimierungsprogramm, in dem das oben beschriebene Optimierungsverfahren eingesetzt wurde, optimiert die Reglerparameter Kp, Tn und Tv für die jeweils eingestellte Kombination der Streckenparameter und Sollwerte. Die erreichte Qualität der Sprungantworten wird jeweils in einem neuen Simulationslauf mit den geänderten Reglerparametern verifiziert. Das Ergebnis ist in Form von Sprungantworten für den geschlossenen Regelkreis mit zeitvarianter Strecke für jeweils optimale Reglerparameter im Sinne der Anforderungen in den Abb. 8 und 9 gezeigt.

Zur Erstellung der Trainingsdaten für das neuronale Netze sind folgende Kombinationen untersucht worden: Führungsgrößenänderungen w_0 = 20°C, 10°C und 5°C. Für die Streckenparameter-Variationen wurden jeweils neun Betriebspunkte gewählt: K_s= [0.0342; 0,0399; 0.0456; 0,0513; 0.057; 0,0627; 0.0684; 0,0741; 0.0798] sowie T_1= [840s; 892,5s; 945s; 997,5s; 1050s; 1102,5s; 1155; 1207,5s; 1260s]. Dadurch ergibt sich eine Tabelle für die betrachteten Betriebspunkte der Regelstrecke. Von dieser ist weiter unten ein Ausschnitt dargestellt.

Wichtiger für die Qualität der Regelergebnisse sind die validierten Sprungantworten des geschlossenen Regelkreises für die verschiedenen Betriebspunkte. Eine Auswahl der Ergebnisse der Optimierung ist in den nachfolgenden Abbildungen dargestellt.

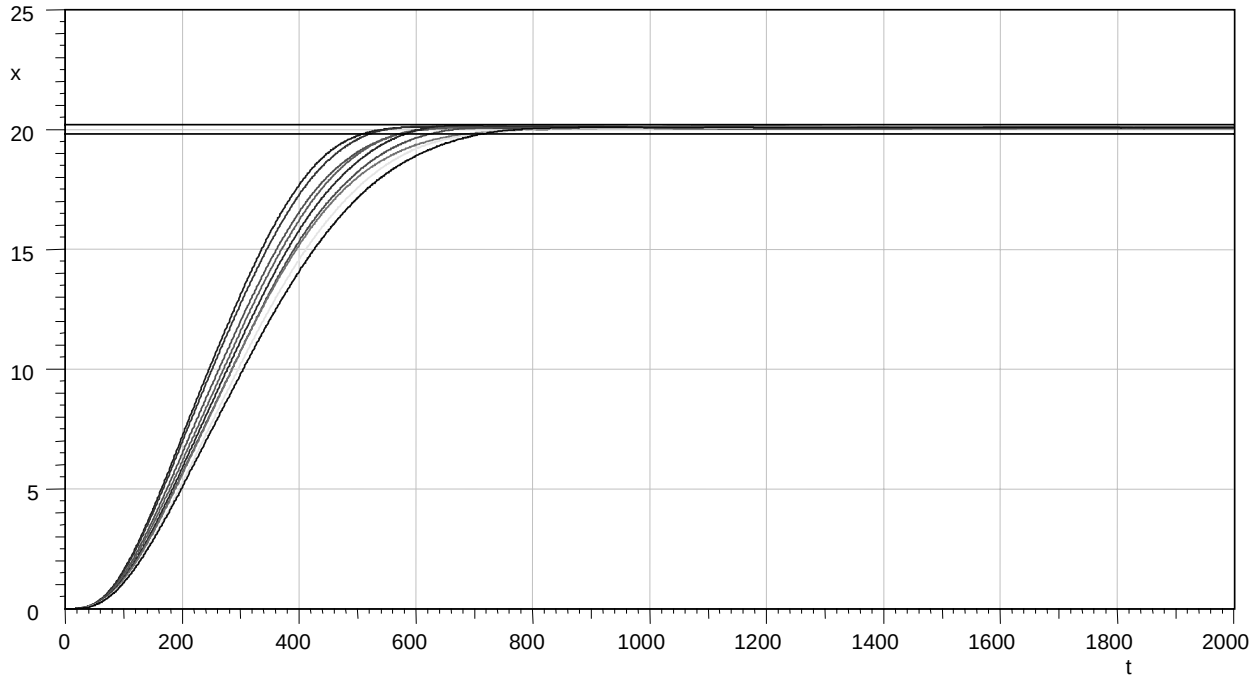


Abb. 8: Regleroptimierung mit Evolutionären Algorithmen für $w_0 = 20^oC$

Auf Angabe der Betriebspunkte wurde in den Abb. 8 und 9 aus Gründen einer besseren Übersicht bewusst verzichtet. Es ist gut erkennbar, das die Streckendynamik mit den Streckenparametern verändert wird. Die Ausregelzeit variiert ebenfalls.

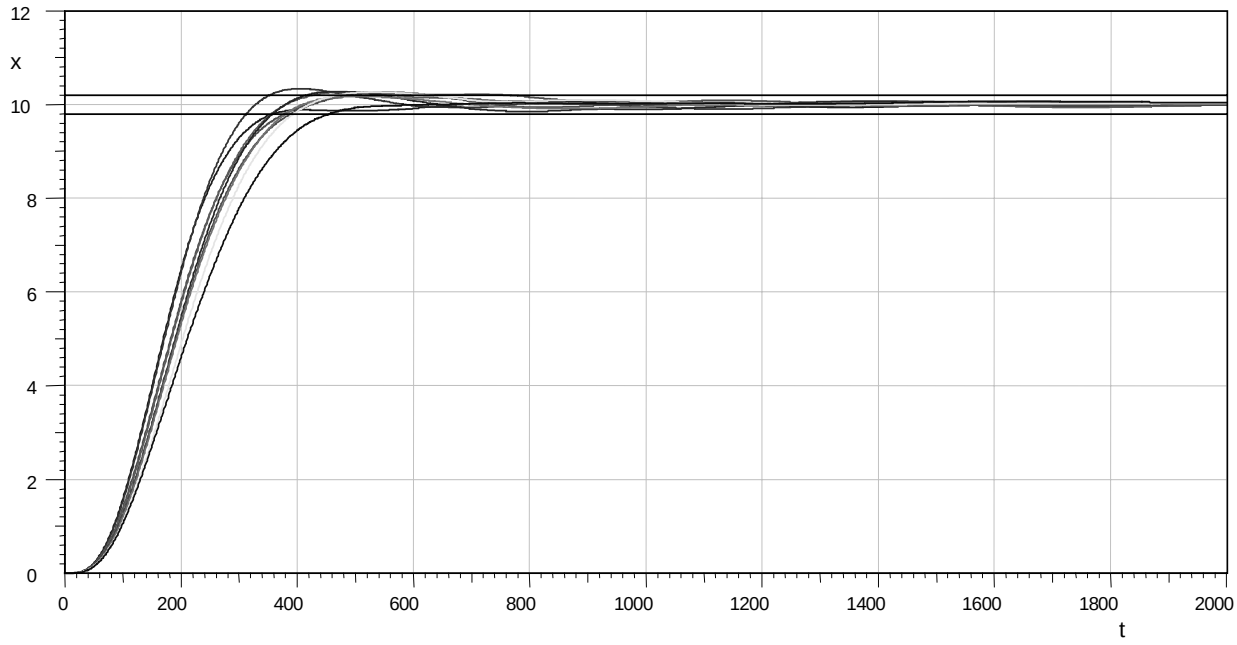


Abb. 9: Regleroptimierung mit Evolutionären Algorithmen für $w_0 = 10^oC$

Die ermittelten Reglereinstellungen und die sich dadurch ergebenden Regelgüten für den geschlossenen Kreis sind in einer Excel-Tabelle abgelegt. In der Praxis können solche Daten bei funktionsfähiger Identifikation im laufenden Betrieb durch das Leitsystem ermittelt werden. Die Excel-Tabelle enthält die Daten für das Training des Neuronalen Netzes.

Tabelle: Auszug aus den Trainingsdaten für das Künstlich Neuronale Netz

w	Ks	Kr	Tn	Tv	□xm	Tan	Taus	Ks	□Kr	□Tn	□Tv
20	0,0342	4,00	1000,00	50,00	-2,5148	0,0	0,0	0,03327	6,57	737,97	21,29
20	0,0342	4,00	1000,00	125,50	-2,7323	0,0	0,0	0,03137	6,57	737,97	-54,21
20	0,0342	4,00	1000,00	200,00	-3,8539	0,0	0,0	0,02983	6,57	737,97	-128,71
20	0,0342	4,00	1800,00	50,00	-12,8721	0,0	0,0	0,03244	6,57	-62,03	21,29
20	0,0342	4,00	1800,00	125,50	-13,6644	0,0	0,0	0,03075	6,57	-62,03	-54,21
20	0,0342	4,00	1800,00	200,00	-14,8222	0,0	0,0	0,03008	6,57	-62,03	-128,71
20	0,0342	4,00	2600,00	50,00	-17,4907	0,0	0,0	0,03219	6,57	-862,03	21,29
20	0,0342	4,00	2600,00	125,50	-18,3089	0,0	0,0	0,03157	6,57	-862,03	-54,21
20	0,0342	4,00	2600,00	200,00	-19,7887	0,0	0,0	0,02987	6,57	-862,03	-128,71
20	0,0342	11,50	1000,00	50,00	15,2109	392,0	392,0	0,03449	-0,93	737,97	21,29
20	0,0342	11,50	1000,00	125,50	8,2511	523,4	523,4	0,03516	-0,93	737,97	-54,21
20	0,0342	11,50	1000,00	200,00	9,9469	661,7	661,7	0,03491	-0,93	737,97	-128,71

20	0,0342	11,50	1800,00	50,00	6,2007	416,3	693,3	0,03435	-0,93	-62,03	21,29
20	0,0342	11,50	1800,00	125,50	0,7748	791,3	791,3	0,03383	-0,93	-62,03	-54,21
20	0,0342	11,50	1800,00	200,00	1,8491	941,1	941,1	0,03160	-0,93	-62,03	-128,71
20	0,0342	11,50	2600,00	50,00	2,7064	435,1	595,1	0,03366	-0,93	-862,03	21,29
20	0,0342	11,50	2600,00	125,50	-2,6254	0,0	0,0	0,03352	-0,93	-862,03	-54,21
20	0,0342	11,50	2600,00	200,00	-2,3656	0,0	0,0	0,03149	-0,93	-862,03	-128,71
20	0,0342	19,00	1000,00	50,00	23,5292	375,1	375,1	0,03536	-8,43	737,97	21,29
20	0,0342	19,00	1000,00	125,50	11,9928	437,3	437,3	0,03930	-8,43	737,97	-54,21

...

Als Testdaten für die Verifikation des trainierten KNN wurden einige Parametersätze für optimale Betriebspunkte herangezogen. Sie wurden natürlich vor dem Training aus der Trainingsdatenmenge entnommen und nicht für das Training benutzt.

5 Verifikation der Adaption

Die Adaption der Reglerparameter des PID-Reglers durch das Künstlich Neuronale Netz erfolgt in mehreren Adaptionsschritten. Dabei ist es nicht erforderlich, dass die Anfangswerte der Reglerparameter aus dem Zahlenbereich der Trainingsdaten stammen.

Funktionsweise: Die Reglerparameter des Neuro-PID-Reglers werden so eingestellt, dass sie für jeweils gültige Streckenparameter K_S und T_1 immer eine optimale Regelgüte gewährleisten. Verändern sich die Streckenparameter bei Verwendung eines gewöhnlichen PID-Reglers, so verschlechtert sich die Regelgüte. Im schlechtesten Fall kann der Regelkreis instabil und damit unbrauchbar werden. Durch die Neuro-Adaption werden die Änderungen der Streckenparameter detektiert und die Reglerparameter so angepasst, dass die Regelgüte auf dem gewünschten Niveau bleibt. Auch während der Inbetriebnahme der Anlage können grob verstellte Regler durch die Adaption optimiert werden.

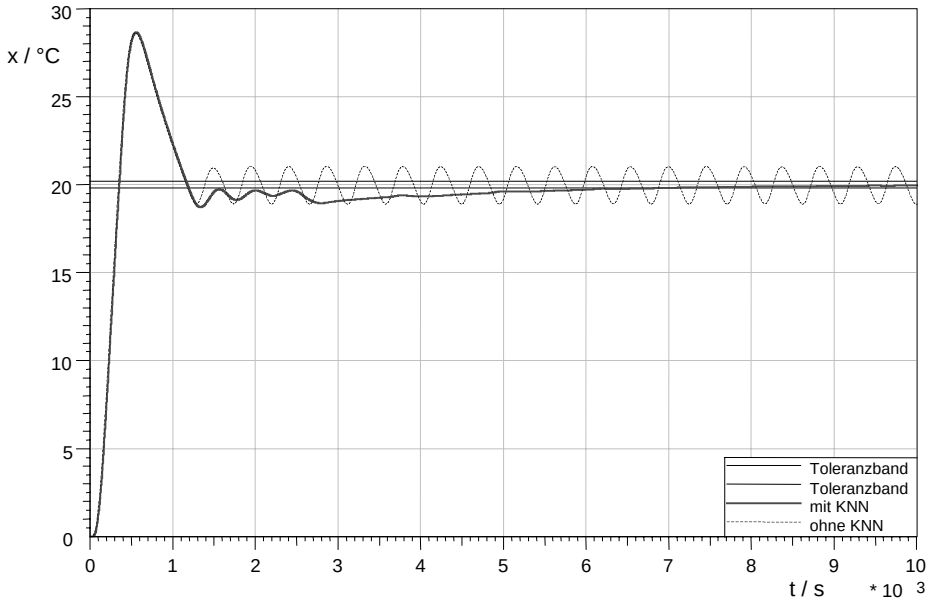


Abb. 10: Vergleich der Sprungantwort eines fehlparametrierten PID-Reglers ohne KNN gegen den Neuro-PID-Regler

Abbildung 10 zeigt die Sprungantworten des geschossenen Regelkreises mit adaptivem Neuro-PID-Regler im Vergleich zu einem konventionellen PID-Regler (gestrichelte Linie) mit nicht optimalen Reglerparametern. Die Reglerparameter wurden zu Beginn auf die Werte K_R=50; T_N=1000 und T_V=10 grob verstellt. Die Streckenparameter betragen konstant K_S=0,057$^{°C}/_W$ und T_1=1050s.

6 Bewertung

Der Neuro-PID-Regler benötigt drei Adaptionsschritte, um die Reglerparameter auf optimale Werte einzustellen. Weitere drei Adaptionsschritte werden durch die automatische Testsprungauslösung erzeugt, weil die Regelgröße das Toleranzband (dünne Linien um x = 20°C) nach einer max. Ausregelzeit T_{aus} noch nicht erreicht hat. Dies ist eine Folge der Adaptionsstrategie, d.h. die Adaption wird aktiviert, wenn die Ausregelzeit einen vorgegebenen Grenzwert überschreitet. Die Reglerparameter werden durch die weiteren Adaptionsschritten nur schwach verändert.

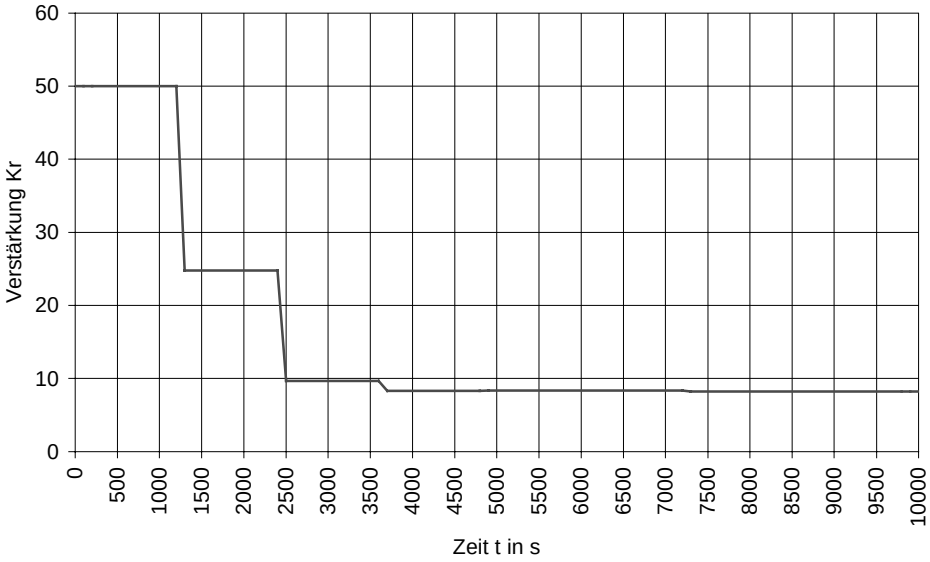


Abb. 11: Adaptionsschritte für K_R für die weiter oben gezeigte Sprungantwort

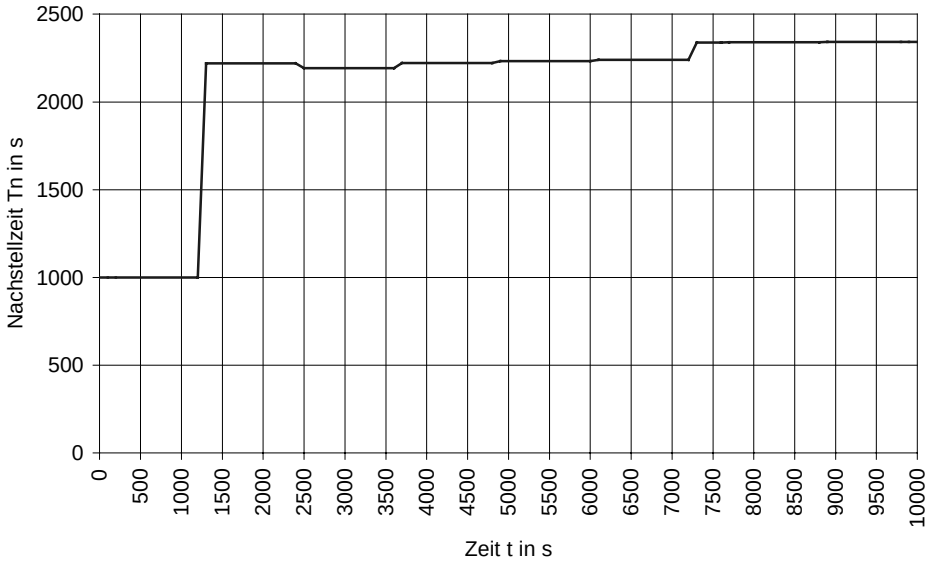


Abb. 12: Adaptionsschritte für T_n für die weiter oben gezeigte Sprungantwort

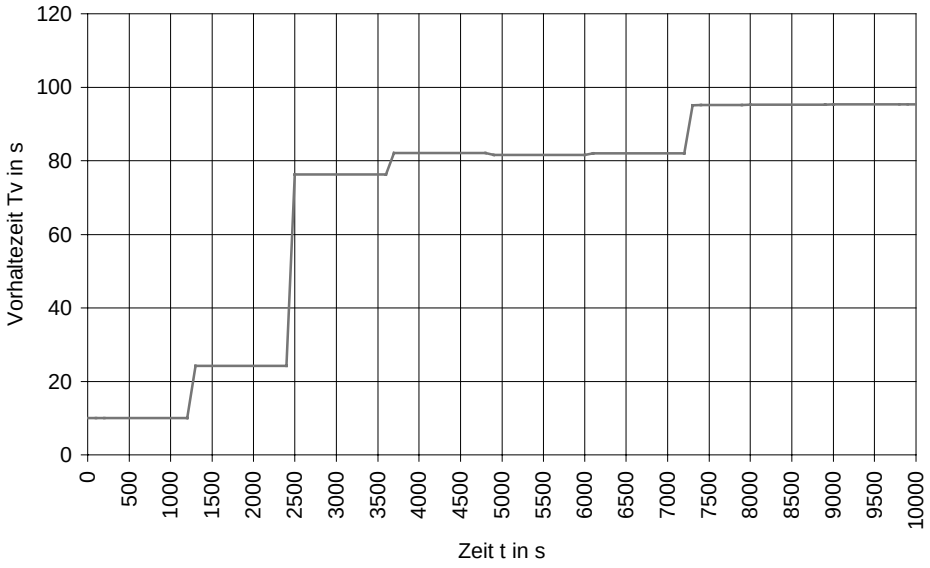


Abb. 13: Adaptionsschritte für T_V für die weiter oben gezeigte Sprungantwort

Die Zeitvarianz der Streckenparameter K_s und T_1 wirkt ähnlich wie eine Störgröße. Insbesondere dann, wenn die Veränderung schnell erfolgt. Die Adaption wird aktiviert, wenn der Ausregelvorgang (T_{aus}) dieser Störung nicht schnell genug verläuft. Ist die Veränderung der Streckenparameter dagegen von langsamer Natur, dann hat sie unter Umständen keine Auswirkungen auf die Regelgröße. In diesem Fall wird die Adaption erst nach einer Führungsgrößenänderung oder durch ein zyklisches Testsignal (siehe Uhrsymbol in Abb. 4) aktiviert. Eine Verstellung der Reglerparameter wird durch diese Strategie nur dann durchgeführt, wenn die vorgegebene Regelgüte nicht erreicht wird.

7 Zusammenfassung

Es wurde ein Konzept zur Adaption von Temperaturreglern an zeitvariante Regelstrecken mit Hilfe von Künstlich Neuronalen Netzen vorgestellt. Die hierfür entwickelte Strategie nach dem Grundsatz **„nur soviel Adaption wie nötig“** kann sowohl für die Inbetriebnahme von einfachen Regelstrecken, als auch zur Nachführung der Reglerparameter bei zeitvarianten Regelstrecken im laufenden Betrieb herangezogen werden.

Der sogenannte adaptive Neuro-PID-Regler wurde auf Industrie-PC und SPS S7-400 portiert. Im Falle der SPS war es eine aufwendige Reimplementierung. Die Systemumgebungen sind, was die Performance anbelangt, vollkommen ausreichend. In einem nächsten Schritt soll die Portierung auf einen Kompaktregler erfolgen.

In der jetzigen Version wird das KNN vor dem eigentlichen Regelbetrieb trainiert. Dazu kann auf Daten aus einem vorhandenen Leitsystem bzw. aus einer historischen Datenbank für die Erzeugung der Trainingsdaten zurückgegriffen werden.
Eine interessante, noch zu lösende Fragestellung ist das Nachtrainieren des KNN im laufenden Betrieb des Neuro-PID-Reglers.

Hinweis: Die Arbeiten wurden vom Ministerium für Schule, Wissenschaft und Forschung des Landes NRW im Rahmen der Forschungsverbundes „Neuro-Fuzzy-Logik an Fachhochschulen in NRW“ an der MFH Iserlohn gefördert.

Literatur

[1] Dormeier, S; Peters, D.: Adaptive Temperaturregelung mit Künstlichem Neuronalen Netz

[2] Dormeier, S; Büchel, M.; Lehmann, U.; Peters, D.; Weiner, E.: Adaptive Temperaturregelung mit Künstlich Neuronalem Netz. Exponat auf der INTERKAMA 1999 in Düsseldorf

[3] WinFACT98 Benutzerhandbuch Release 1.0. Ingenieurbüro Dr. Kahlert, Hamm 1991-98

[4] WinFACT98 Kurz-Dokumentation. Ingenieurbüro Dr. Kahlert, Hamm 1998

[5] WinFACT98 AutoCode-Generator Benutzerhandbuch Release 1.0, Ingenieurbüro Dr. Kahlert, Hamm 1991-99

[6] Kahlert, J.: Fuzzy Control für Ingenieure. Analyse, Synthese und Optimierung von Fuzzy-Regelungssystemen. Vieweg Verlag Wiesbaden 1995

[7] Kahlert, J.: Globale vektorielle Optimierung mit Evolutionsstrategien. Automatisierungstechnik at 3/95

[8] NeuroModel Benutzerhandbuch Release 1.2. Atlan-tec KG Viersen 1995

[9] Lehmann, U. et. al.: Arbeitsberichte des Forschungsverbundes Neuronale Fuzzy-Logik der Fachhochschulen in NRW. Iserlohn April 1999

[10] Kruse, R.; Klawon, F.; Nauck, D.: Neuronale Netze und Fuzzy-Systeme. Vieweg Verlag, Braunschweig/Wiesbaden 1996

Ein Beitrag zur Didaktik – simulierbare Applikationsbeispiele zu FuzzyControl++ für SIMATIC S7

B-M. Pfeiffer
Siemens AG

A&D GT 5 (Vorfeldentwicklung Automatisierungsfunktionen)
D-76181 Karlsruhe
Tel.: +49-721-595-5973 - Fax: +49-721-595-6728
e-mail: Bernd-Markus.Pfeiffer@khe.siemens.de
http://www.ad.siemens.de

Kurzfassung: Anhand von sechs einfachen, überschaubaren Anwendungbeispielen mit einer entsprechenden Prozess-Simulation wird ein Überblick verschiedener Anwendungsmöglichkeiten der Fuzzy-Logik in den Bereichen Regelungstechnik, Prozessführung, Klassifikation, Mustererkennung und Fehlerdiagnose gegeben. Besonderer Wert wird dabei auf die Vermittlung von allgemeinen, für zukünftige Anwender nützlichen Aspekten der Fuzzy-Logik gelegt.

1 Einführung

Der heutige Entwicklungsgrad der Fuzzy-Tools und der darin verwendeten Methoden erlaubt es, diese uneingeschränkt für industrielle Anwendungen einzusetzen. Da es sich bei den erfolgreichen Anwendungen von Fuzzy Logik jedoch meist nicht mehr um einfache Fuzzy Controller als Ersatz für klassische Regler handelt, sondern um mehr oder weniger komplexe hybride Strukturen, ist es für die Anwender von großem Nutzen, wenn sie zusammen mit dem Fuzzy-Tool bereits eine Sammlung von ausprogrammierten Applikationsbeispielen geliefert bekommen.

Diese Beispiele können zur Einarbeitung im Sinne eines Tutorials dienen, oder bei Vorführungen (z.B. auf Messen), bei Kursen und Schulungen oder in der Lehre eingesetzt werden. Darüber hinaus bilden sie auch eine „Baustelle“ für Ansätze, Strukturen und Programm-Module, die von Anwendern als Basis für eigene Applikationen genützt werden können. Auf diesem Wege findet ein Anwender schneller zu einer passenden Lösung für seine Problemstellung, und kann sie ggf. auch leichter implementieren.

Ein Ziel bei der Auswahl von Beispielen ist es, das Spektrum der Fuzzy-Logik Anwendungen möglichst breit abzudecken, d.h. neben Beispielen aus der Regelungstechnik im engeren Sinne auch Anwendungen aus den Bereichen Prozessführung, Klassifikation, Mustererkennung und Fehlerdiagnose anzubieten. Allgemeine Aspekte beim Einsatz von Fuzzy-Logik wie die Darstellung von Kennfeldern, die logische Verknüpfung von Information unterschiedlicher Natur (analoge Messwerte, binäre Variablen und linguistische Eingaben) sowie die Organisation von Wissen in hierarchischen Regelbasen sollen illustriert werden.

Dennoch sollten die Beispiele möglichst einfach und überschaubar bleiben, und mit einer entsprechenden Prozess-Simulation ausgestattet sein, damit sie sofort auf dem Zielsystem, z.B. einer SPS, im vollständigen Ablauf betrachtet werden können.

Ein wichtiger Bereich von Anwendungen der Fuzzy-Logik muss leider ausgespart werden: Lern- bzw. Identifikationsverfahren, d.h. Verfahren zur automatischen Regelgenerierung. Diese Verfahren lassen sich nicht aus einfachen Fuzzy-Logik-Blöcken aufbauen, sondern werden meist als eigenständige, umfangreiche Tools wie z.B. Winrosa [1.] realisiert. Im Umfeld von SIMATIC S7 steht dazu das mit dem Fuzzy-Tool gekoppelte Neuro-Tool „NeuroSystems“ zur Verfügung, das ein spezielles Verfahren zum Training von „Neuro=Fuzzy“-Systemen enthält, die aus drei Teilnetzen bestehen, mit denen die Teilschritte Fuzzifizierung, Inferenz und Defuzzifizierung nachgebildet werden.

In einem ersten Schritt sind folgende sechs Fuzzy-Applikationsbeispiele realisiert worden:

- Imitation eines linearen PI-Reglers, als Basis für weitere Manipulationen
- Gesteuerte Parameteradaption eines PID-Reglers in Abhängigkeit vom Arbeitspunkt eines nichtlinearen Prozesses
- Selbsteinstellung klassischer PI-Regler entsprechend der Strategie menschlicher Bediener
- Mustererkennung mit Fuzzy-Automaten am Bsp. einer Stranggießanlage
- Entscheidungslogik zur Optimierung einer Kühlwasseraufbereitung
- Fehlerdiagnose aus Symptomen an einem Gleichstrommotor

In den folgenden Kapiteln zu den einzelnen Beispielen sind die allgemeinen (d.h. nicht auf den Einzelfall beschränkten) didaktischen Aspekte jeweils durch einen Kasten hervorgehoben.

2 Imitation eines linearen PI-Reglers, als Basis für weitere Manipulationen

Bei einigen Anwendungen besteht die Aufgabenstellung darin, einen vorhandenen, funktionstüchtigen aber nicht ausreichend leistungsfähigen linearen PI(D)-Regler durch einen besseren Fuzzy-Regler zu ersetzen. In anderen Fällen erfolgt ein erster, linearer Reglerentwurf auf Basis eines einfachen, linearisierten Prozessmodells, bevor mit Hilfe der vielen Freiheitsgrade eines Fuzzy-Reglers gezielte Anpassungen an das nichtlineare Prozessverhalten vorgenommen werden. In diesen beiden Situationen muss zunächst ein gegebener, linearer PI-Regler zu einem äquivalenten Fuzzy-Regler umgesetzt werden. Nach der Transformation von der analytischen Form auf eine linguistische Darstellung kann die weitere Feinanpassung auf linguistischer Ebene, d.h. mit den Mitteln der Fuzzy-Logik erfolgen.

Die Umsetzung von der analytischen auf die linguistische Darstellung lässt sich nach einem in [3.] , S. 96ff vorgeschlagenen Verfahren formalisieren, d.h. automatisch durchführen. Der zeitkontinuierliche PI-Regler

$$G_{PI}(s) = K_R\left(1+\frac{1}{T_I s}\right) \tag{1}$$

wird in seine zeitdiskrete Form

$$G_{PI}(z^{-1}) = K_R\left(z^{-1} + c_{ir}\cdot\frac{z^{-1}}{1-z^{-1}}\right) \tag{2}$$

mit der Abtastzeit T_A und dem relativen I-Anteil $c_{ir} = T_A/T_I$ ([5.] , S. 116) transformiert. Die zusätzliche Verzögerung von einem Abtastschritt im P-Kanal entsteht wegen der Differentiation und anschließenden Integration beim Fuzzy-PI-Regler.

Der zeitdiskrete PI-Regler wird auf analytischem Wege in eine Fuzzy-Reglerstruktur mit dynamischen Ein- und Ausgängen übersetzt (Bild 1).

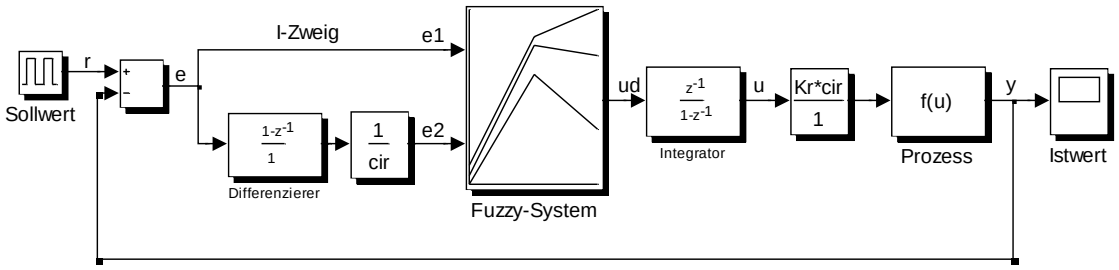


Bild 1: Regelkreis mit Fuzzy-PI-Regler, der mit dynamischen Vorschaltgliedern realisiert ist

Dabei werden als Eingangsgrößen e_1 und e_2 des Fuzzy-Systems die Größen Regelabweichung (Regeldifferenz) $e_1 = e = r - y$ und deren zeitliche Änderung e_2 gewählt. Die Ausgangsgröße u_d entspricht der zeitlichen Änderung der Stellgröße u. Die im Bild bereits dargestellten Vorfaktoren werden im folgenden hergeleitet.

Die Fuzzy-Regeln müssen so bestimmt werden, dass das Verhalten des Fuzzy-Systems mit dem Verhalten des konventionellen PI-Reglers mit den frei wählbaren Parametern K_R und c_{ir} übereinstimmt. Ein möglicher Lösungsweg besteht in der Aufstellung der Systemgleichungen und deren analytischer Lösung [3.] . Der Nachteil dieser Methode liegt darin, dass die Lösung von der Anzahl und Lage der Fuzzy-Zugehörigkeitsfunktionen abhängt und jedes Mal erneut berechnet werden muss.

Aus diesem Grund wird hier eine andere Methode vorgestellt, die unabhängig von der Wahl der Zugehörigkeitsfunktionen ist und wesentlich schneller implementiert werden kann. Dafür wird ein normiertes lineares Fuzzy-Kennfeld mit der Verknüpfung der Eingangsvariablen

$$u_d(e_1, e_2) = e_1 + e_2 \tag{3}$$

betrachtet, das in beiden Koordinatenrichtungen die Verstärkung 1 hat. Um das Verhalten eines PI-Regler nachzubilden werden die Eingangsgrößen mit den Faktoren k_1 und k_2 vormultipliziert. Für den Gesamtregler gilt dann:

$$u(z) = \left[k_1 e + k_2 (1 - z^{-1}) e\right] \frac{z^{-1}}{1 - z^{-1}} = \left[k_1 \frac{z^{-1}}{1 - z^{-1}} + k_2 z^{-1} \right] e \,. \tag{4}$$

Die Faktoren k_1 und k_2 müssen so gewählt werden, dass der reale PI-Regler nachgebildet wird. Durch einen Koeffizientenvergleich der Gleichungen (2) und (4) erhält man den Zusammenhang:

$$k_1 = K_R c_{ir} \text{ und } k_2 = K_R \,. \tag{5}$$

Beide Faktoren wirken als Skalierungsfaktoren der beiden Eingänge e_1 und e_2 des Fuzzy-Systems. Damit man bei der späteren Manipulation des Fuzzy-PI-Reglers mit möglichst anschaulichen Eingangsgrößen arbeiten kann, wird der Vorfaktor k_1 auf den Reglerausgang verlegt. Zwar bleibt dabei der Eingang e_2 weiterhin vorskaliert; er ist aber als zeitliche Änderung von e_1 ohnehin nicht besonders anschaulich. Nach Ausklammern des Vorfaktors k_1 in Gleichung (4) erhält man die Rechenvorschrift:

$$u(z) = \left[e + \frac{1}{c_{ir}} (1 - z^{-1}) e \right] \frac{z^{-1}}{1 - z^{-1}} \cdot K_R c_{ir} \tag{6}$$

Die Parameter des Fuzzy-PI-Reglers können also, wie in Bild 1 bereits dargestellt, von dem zu imitierenden PI-Regler direkt übernommen werden.

An dieser Stelle sollte die Begrenzung des Fuzzy-PI-Reglers erwähnt werden, die sich von einer konventionellen Stellgrößenbegrenzung unterscheidet: Übersteigt eine der Eingangsgrößen das entsprechende Definitionsintervall, so kann die Fuzzy-Ausgangsgröße nicht entsprechend ansteigen und bleibt konstant. Um das Problem zu vermeiden sollte man bereits beim Entwurf der Zugehörigkeitsfunktionen des Fuzzy-Reglers das Definitionsintervall entsprechend groß wählen.

Für die Berechnung der Ausgangs-Singletons des Fuzzy-Reglers wird die Eigenschaft der Einheitssteigung des internen Kennfelds benutzt. Werden etwa die dreieckförmigen Zugehörigkeitsfunktionen $A_1, A_2, \ldots, A_n$ bzw. $B_1, B_2, \ldots, B_n$ der Eingangsvariablen e_1 und e_2 mit den Mittelpunkten $a_1, a_2, \ldots, a_n$ bzw. $b_1, b_2, \ldots b_m$ betrachtet, so können die jeweiligen Singletons S_{ij} zur Fuzzy-Regel „*IF* $e_1 = A_i$ *AND* $e_2 = B_j$ *THEN* $ud = S_{ij}$ “ direkt berechnet werden:

$$S_{ij}(e_1 = a_i,\ e_2 = b_j) = a_i + b_j \,. \tag{7}$$

Im vorliegenden Beispiel ergeben sich zunächst folgende Ausgangs-Singletons als Matrix (Tabellendarstellung der Regelkonklusionen):

$$S = \begin{bmatrix} \mathbf{-6766} & -2322 & 2122 & \mathbf{6566} \\ -6699 & \mathbf{-2255} & \mathbf{2188} & 6632 \\ -6632 & \mathbf{-2188} & \mathbf{2255} & 6699 \\ \mathbf{-6566} & -2122 & 2322 & \mathbf{6766} \end{bmatrix}. \tag{8}$$

Da FuzzyControl++ nur 9 verschiedene Ausgabe-Singletons verarbeiten kann, müssen die 16 berechneten Werte reduziert werden. Dazu werden die in Gleichung (8) fett markierten Werte auf den Diagonalen auch für die benachbarten Elemente übernommen. Man sollte bei der Auswahl der Stützwerte darauf achten, dass das Kennfeld symmetrisch zum Ursprung bleibt und das ursprüngliche Kennfeld, zumindest

in der Umgebung des Punktes $e_1 = e_2 = 0$, genügend gut approximiert wird, da dieser Bereich für die stationäre Genauigkeit des Regelkreises von großer Bedeutung ist.

Bei der in Bild 2 gezeigten Realisierung auf der SPS erkennt man, das der Fuzzy-PI-Regler aus zwei Funktionsbausteinen besteht: dem eigentlichen Fuzzy-Kennfeld, sowie einem Rahmenbaustein, in dem die dynamischen Vorschaltglieder enthalten sind. Am Rahmenbaustein lassen sich direkt die Parameter Verstärkung und Nachstellzeit des linearen Vorbild-PI-Reglers parametrieren, während am Fuzzy-Kennfeld mit Hilfe des Fuzzy-Tools Manipulationen auf linguistischer Ebene vorgenommen werden können.

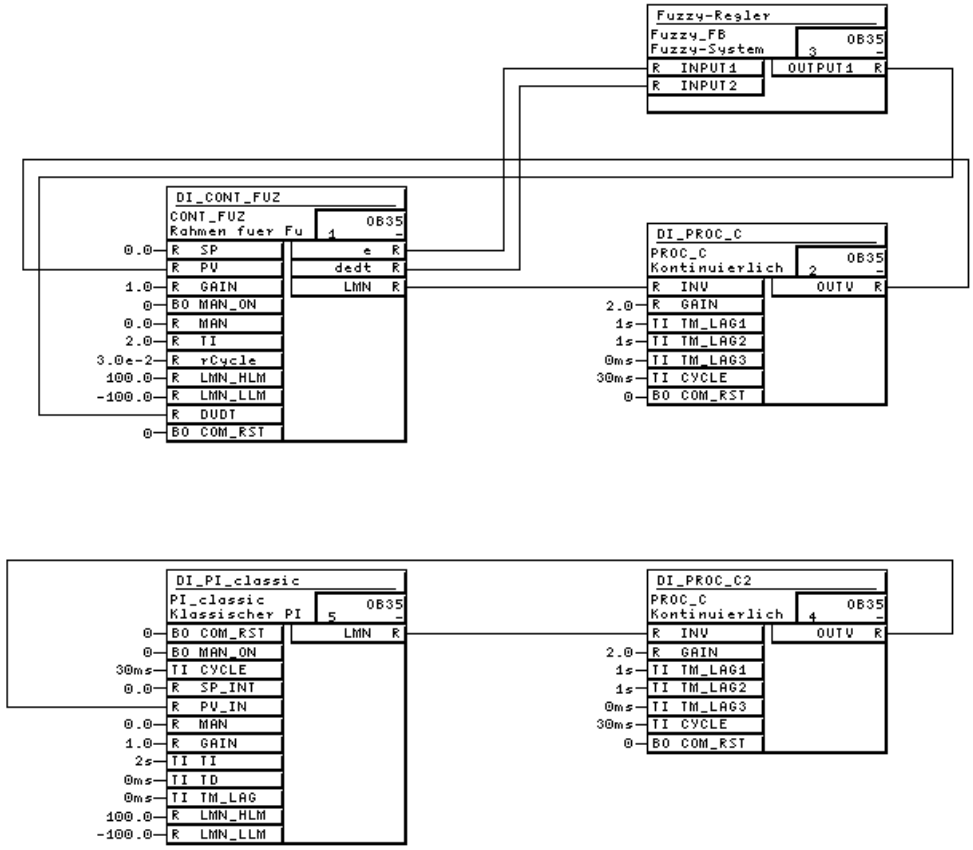


Bild 2 CFC-Plan (***C***ontinuous ***F***unction ***C***hart) zur Simulation eines Fuzzy-PI-Reglers, mit folgenden Funktionsbausteinen. Fuzzy_FB: Fuzzy-Kennfeld, CONT_FUZ: Regler-Rahmen mit dynamischen Vorschaltgliedern, PROC_C: Prozesssimulation als Verzögerungsglied dritter Ordnung, PI_classic: linearer PI-Regler, entspricht CONT_C bei Simatic Step 7

3 Gesteuerte Parameteradaption eines PID-Reglers in Abhängigkeit vom Arbeitspunkt eines nichtlinearen Prozesses

Diese Struktur repräsentiert vielleicht den am weitesten verbreiteten Einsatz von Fuzzy-Logik zur Regelung im engeren Sinne. Ein nichtlinearer Prozess wird in mehreren Punkten linearisiert und für jeden Arbeitspunkt wird ein entsprechender Regler entworfen. Anschließend wird in einer übergeordneten Fuzzy-Steuerungsebene eine „weiche“ Umschaltung zwischen den einzelnen Reglerparametern in Abhängigkeit der Regelgröße realisiert. Im Vergleich zum „crisp“ gain scheduling hat diese Lösung den Vorteil, dass die Fuzzy-Logik zwischen den verschiedene Parametersätzen interpoliert, anstatt sie schlagartig umzuschalten.

Das vorliegende Simulations-Beispiel ist einem realen technischen Prozesses nachempfunden: der **Temperaturregelung einer Glühbirne**, wie sie von der Siemens A&D SH als Messe-Vorführmodell verwendet wird.

Zur Umsetzung der Regelstrecke auf einer SIMATIC-S7 wurde eine Prozessidentifikation durchgeführt. Demnach kann die Regelstrecke als ein lineares Übertragungsglied zweiter Ordnung

$$G_S(s) = \frac{K_g}{(1+T_1 s)(1+T_2 s)} \tag{9}$$

mit den vom Arbeitspunkt (Temperatur T) abhängigen Parametern K_g, T_1 und T_2 angesehen werden. Es liegt also nahe, den nichtlinearen Prozess gemäß Bild 3 als lineares System mit überlagerter Fuzzy-Parametersteuerung zu implementieren.

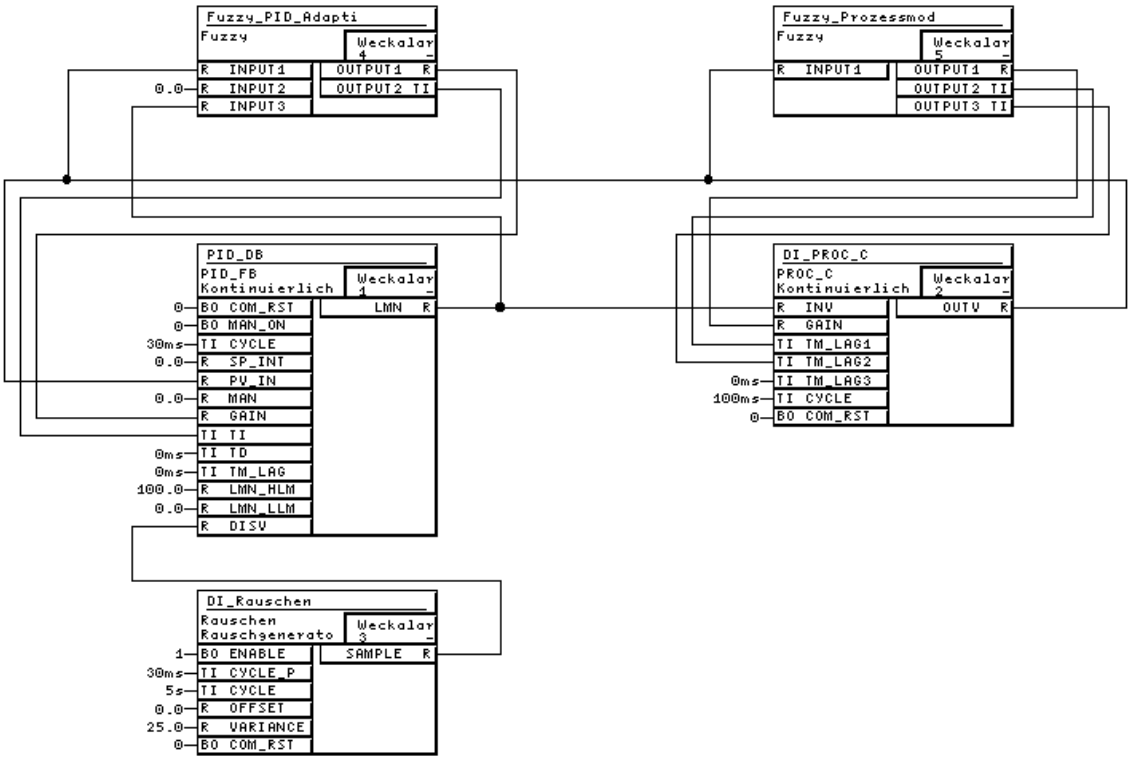


Bild 3 Fuzzy-Gain-Scheduling PID-Regler PID_FB an einer Fuzzy-adaptierten Prozess-Simulation PROC_C

Als Hauptproblem ist das unruhige Verhalten des reinen PI-Reglers bei tiefen Temperaturen zu nennen. Der Regler ist für hohe Temperaturen ausgelegt und wegen der Nichtlinearitäten der Regelstrecke für tiefe Temperaturen nicht so gut geeignet. Der fuzzy-adaptierte Regler dagegen weist in allen Temperaturbereichen ein in etwa gleich gutes Regelverhalten auf und ist daher flexibler einsetzbar. Durch abwechselndes Ein-/Ausschalten der Fuzzy-Adaption bei gleichen Sollwertsprüngen kann das bessere Regelverhalten des fuzzy-adaptierten PI-Reglers im Vergleich zum reinen PI-Regler veranschaulicht werden.

4 Selbsteinstellung klassischer PI-Regler entsprechend der Strategie menschlicher Bediener

In der Praxis wird die Inbetriebnahme und Einstellung von Reglern vielfach nicht als wissenschaftliche Aufgabe betrachtet, die mit Hilfe von Modellbildung und mathematischen Verfahren zu lösen ist, sondern als eine „Kunst“ (oder gar „Kunsthandwerk“), die mit Erfahrung, Fingerspitzengefühl und systematischem Probieren angegangen wird. Die Fuzzy-Logik ist ein geeignetes Hilfsmittel, um solche heuristischen Strategien zu erfassen, darzustellen und damit letztlich zu automatisieren.

Das von [2.] vorgeschlagene und in [3.] ausführlich dokumentierte Verfahren zur Selbsteinstellung klassischer PI-Regler entsprechend der Strategie menschlicher Bediener wird hier auf einer SPS implementiert. Ähnliche Ansätze sind in den vergangenen Jahren immer wieder aufgegriffen worden, zuletzt von [4.].

Die Anforderungen an die Regelgüte werden anhand der fuzzifizierten Merkmale Überschwinger und Einregelverhältnis einer Sprungantwort mit Hilfe von Zugehörigkeitsfunktionen definiert. Mit Hilfe von zwei Regelbasen mit jeweils 12 Fuzzy-Regeln werden aus den beobachteten Sprungantworten Rückschlüsse auf sinnvolle Änderungen der Reglerparameter gezogen. Ausgehend von einer sehr vorsichtigen Grundeinstellung wird auf diesem Weg entsprechend der Strategie menschlicher Bediener nach einigen Iterationen eine brauchbare Parameterkombination für den Regler gefunden.

Das Verfahren ist sehr einfach und anschaulich. Es kommt ohne ein explizites Prozessmodell aus und ist daher für ein großes Spektrum unterschiedlicher (stabiler!) Streckentypen geeignet: aperiodische Strecken, aber auch Strecken mit leicht oszillierendem oder nicht phasenminimalem Verhalten sowie mit kleineren Totzeiten. Folgende Voraussetzungen sind jedoch für den praktischen Einsatz relevant:

- Der Prozess muss mit einem PI-Regler zufriedenstellend beherrschbar sein, eine Erweiterung des Ansatzes auf PID-Regler hat sich als schwierig erwiesen.
- Der Prozess muss (kleine) Überschwinger (z.B. 5-10%) erlauben, da eine Spezifikation „null Überschwinger“ wenig signifikante Merkmale bietet und daher mit diesem Verfahren nicht direkt erreicht werden kann.
- Der Prozess muss so schnell sein, dass die Auswertung mehrerer (3 bis 10) Sprungantworten vom Zeitaufwand her zumutbar ist.

Aus Speicherplatzgründen wird bei der SPS-Realisierung darauf verzichtet, vollständige Sprungantworten als Datensätze abzulegen, und stattdessen ein kleine Signalverarbeitungsroutine zur online-Bestimmung der Merkmale Anschwingzeit, Überschwinger und Einschwingzeit einer Sprungantwort realisiert.

Ansonsten ergeben sich jedoch keine neuen Gesichtspunkte gegenüber den bekannten Literaturstellen, so dass auf eine ausführliche Darstellung an dieser Stelle verzichtet werden kann.

5 Mustererkennung mit Fuzzy-Automaten am Bsp. einer Stranggießanlage

Bei verfahrenstechnischen Anlagen werden sehr viele Prozessgrößen (z.B. Temperaturen, Drücke usw.) fortlaufend gemessen. Aus den zeitlichen Verläufen kann oftmals auch auf Abweichungen vom Normalbetrieb, sich anbahnende Fehler oder Beeinträchtigungen der Produktqualität geschlossen werden. Dabei weisen die Prozessgrößenverläufe im Fehlerfall bestimmte Muster auf, anhand deren die einzelnen Fehler erkannt werden können. Im Gegensatz zu einer offline-Mustererkennung, mit der komplette, historische Zeitverläufe ausgewertet und klassifiziert werden können (blockweise Datenverarbeitung) ist hier oft eine sequentielle (online-) Mustererkennung gefordert. Diese kann mit Zustands-Automaten und sog. synataktischen Verfahren durchgeführt werden. Für die Erkennung von Mustern in kontinuierlichen Signalen eignen sich *Fuzzy-Automaten* [6.] besonders gut, bei denen die Messwerte in linguistische Werte transformiert werden. Im Gegensatz zu klassischen Automaten sind bei den Fuzzy-Automaten die Übergänge zwischen den Zuständen *unscharf*. Der Automat kann sich also z.B. zu 20% im Zustand 1 und zu 80% im Zustand 2 befinden.

Ein Mustererkennungsautomat bestimmt fortlaufend die Wahrscheinlichkeit *P* dafür, dass sich im Meßgrößenverlauf ein Muster entwickelt. Als Eingangsgrößen des Automaten werden nur der im Zeitschritt *i* erfasste Messwert x_i und seine zeitliche Änderung $\Delta x_i = x_i - x_{i-1}$ verwendet. Ein einzelner Kurvenpunkt bildet jedoch noch kein Muster. Informationen über den vergangenen Messwertverlauf sind also unbedingt erforderlich, weshalb der Automat die Erkennungswahrscheinlichkeit P_i als innere Zustandsgröße verwendet. P_i enthält in extrahierter Form die Information über den bisherigen Messwertverlauf, so dass zusammen mit den aktuellen Messwerten x_i und Δx_i der aktuelle Wahrscheinlichkeitswert $P_{i+1} = F(x_i, \Delta x_i, P_i)$ ermittelt werden kann. Diese wird zwischengespeichert und im nächsten Schritt auf den Systemeingang geschaltet. Die Rückkopplung der Zustandsgröße auf den Eingang des Fuzzy-Systems erlaubt somit eine zustandsgerechte Behandlung der Messwerte.

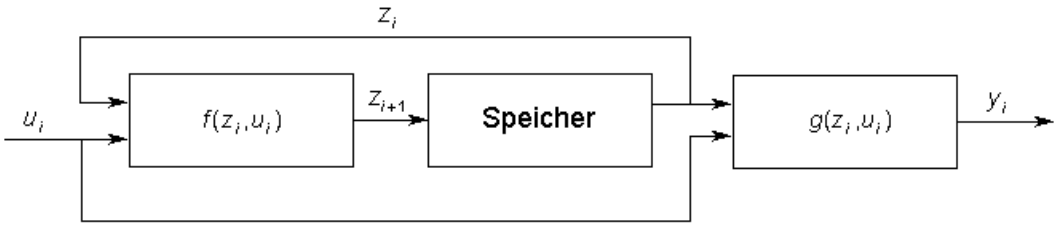


Bild 4 Prinzipielle Struktur eines Fuzzy-Mealy-Automaten

Dazu wird ausgehend von boolschen Automaten die allgemeinere Klasse der Fuzzy-Automaten definiert, indem man in den Beschreibungsgleichungen boolscher Automaten (hier Mealy-Automaten)

$$z_{i+1} = f(z_i, u_i) \text{ und } y_i = g(z_i, u_i) \qquad (10)$$

mit der Eingangsgröße *u*, der Zustandsgröße *z* und dem Ausgangsvektor *y* die *boolschen* Funktionen *f* und *g*, die nur diskrete Werte annehmen können, durch „*Fuzzy*-Funktionen“ ersetzt., d.h. durch vollständige Fuzzy-Systeme mit Fuzzifizierung und Defuzzifizierung von Ein- und Ausgangsgrößen (Bild 4). Die Größen *u*, *y* und insbesondere der gespeicherte Zustandsvektor *z* sind dennoch scharfe Größen, werden aber intern unscharf verarbeitet.

Um einen solchen Fuzzy-Automaten zu entwerfen, müssen zunächst die Automatenzustände festgelegt werden. Dies erfolgt durch Extraktion einiger für das betrachtete Muster markanter Punkte mit deren Ableitungen.

Der im vorliegenden Beispiel entworfene Fuzzy-Automat erkennt Muster von der Form einer Glockenkurve. Ein ähnlicher Automat wird beim **Stranggießen von Stahl** eingesetzt, um sog. Kleber, d.h. Schwachstellen in der Strangschale, die nach Verlassen der Kokille aufbrechen und zu schweren Schäden führen können, anhand von bestimmten Mustern im Temperaturverlauf an der Kokille erkennen zu können.

Für diese Aufgabe wurden fünf Merkmalspunkte nach Bild 5 mit den entsprechenden Steigungen betrachtet, denen die 5 Zustände (Erkennungswahrscheinlichkeiten) 0.2, 0.4, 0.6, 0.8, 1 zugeordnet werden.

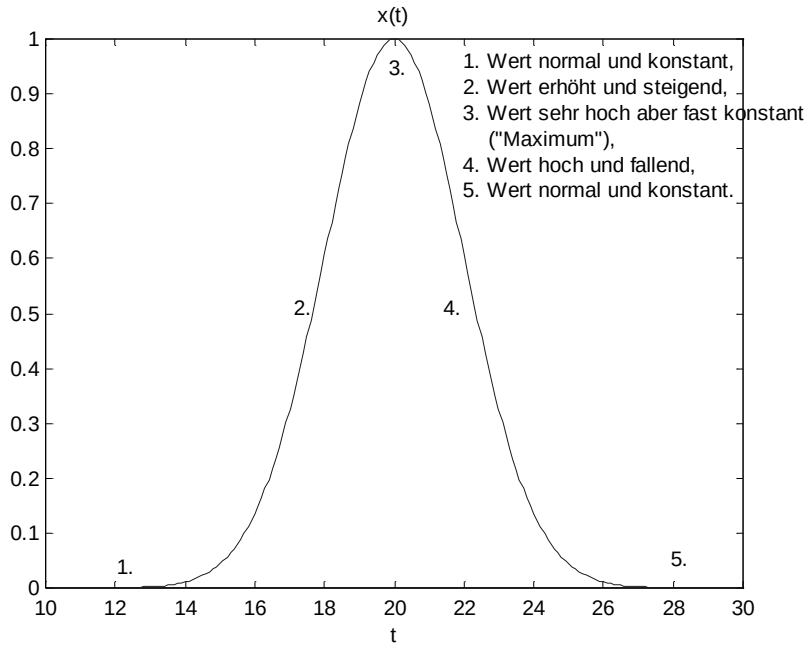


Bild 5: Glockenkurve mit 5 sukzessiven Merkmalen

Wird das erste Merkmal eines Musters erkannt, schaltet der Automat zum nächsten Zustand weiter. Wenn daraufhin das nächste Merkmal erkannt wird, wechselt der Automat zum nächsten Zustand und zeigt damit eine höhere Erkennungswahrscheinlichkeit an. Erst wenn alle Merkmale in der spezifizierten Reihenfolge erkannt worden sind, wird die maximale Erkennungswahrscheinlichkeit erreicht.

Mathematisch gesehen wird das Muster durch die ausgewählten Merkmalspunkte nur ungefähr approximiert, auch weil die Parameter unscharf sind. Somit werden alle ähnlichen Signalverläufe als dieses Muster erkannt. Ähnlich heißt hier, dass die Signalform durch die Abfolge der gewählten Punkte mit den jeweiligen Steigungen

fuzzy-approximierbar ist. Dabei kann sowohl die tatsächliche Signalform als auch die Signalamplitude vom vorgegebenen Muster abweichen. Die Definition der Toleranzbereiche für Form und Amplitude der zu erkennenden Signale erfolgt durch die Festlegung der Zugehörigkeitsfunktionen für die Eingangsvariablen Wert und Steigung.

6 Entscheidungslogik zur Optimierung einer Kühlwasseraufbereitung

Die Fuzzy-Logik wird nicht nur für Regelungszwecke verwendet. Viel öfter wird sie auf höheren Hierarchieebenen eingesetzt, um unterlagerte Regelkreise zu steuern und Prozessabläufe zu koordinieren. Auch binäre logische Steuerentscheidungen, die unter Berücksichtigung einer Vielzahl kontinuierlicher Messwerte zu treffen sind, lassen sich vorteilhaft mit Fuzzy-Logik realisieren, wenn der defuzzifizierte Ausgang des Fuzzy-Systems mit Hilfe von Grenzwertschaltern oder Hysteresegliedern ausgewertet wird. Das Beispiel demonstriert darüber hinaus, dass es bei der Verarbeitung von sehr vielen Eingangsgrößen sinnvoll ist, das Problem in kleinere Teil-Einheiten zu zerlegen und dann mehrere Fuzzy-Systeme hintereinander zu kaskadieren.

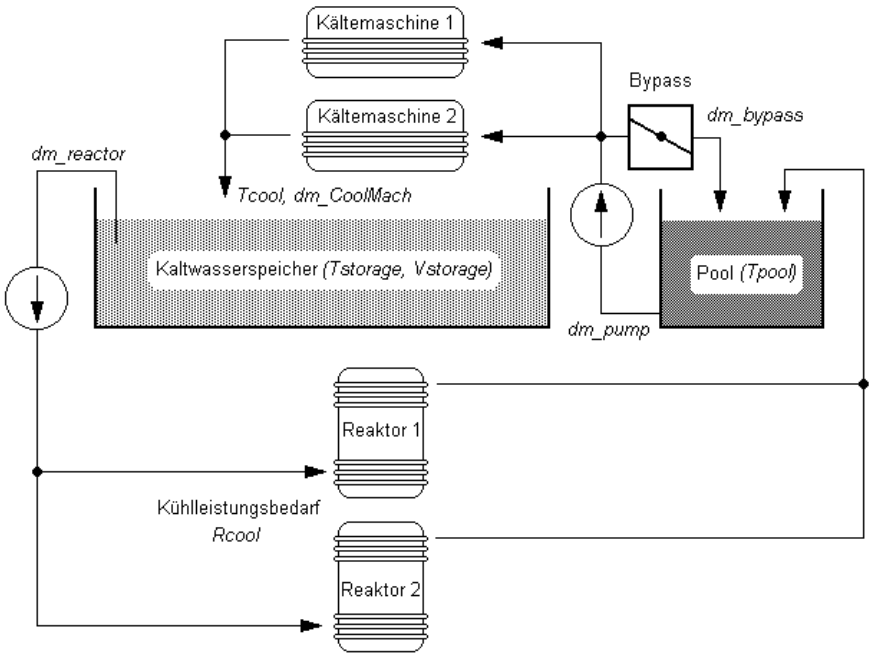


Bild 6 Anlagenschema zur Kühlwasseraufbereitung

Als Beispiel soll eine reale Kühlwasseraufbereitungsanlage für zwei chemische Reaktoren betrachtet werden [7.] . Durch Einsatz der Fuzzy-Logik konnte nicht nur das Operatorteam von der bisherigen manuellen Prozessführung entlastet werden, es wurden auch gesteigerte Produktionsraten und darüber hinaus eine Verringerung der Energiekosten erreicht.

Die Anlage besteht aus zwei Reaktoren, zwei Kältemaschinen und einem Kühlwasserspeicher wie in Bild 6 dargestellt. Bisher wurden die Kältemaschinen im wesentlichen durch manuell eingegebene Kälteanforderungen so gesteuert, dass bei einem laufenden Reaktor nur eine Kältemaschine und bei zwei Reaktoren beide Kältemaschinen eingeschaltet wurden. Das Kühlmittel muss immer in ausreichendem Maß zur Verfügung stehen , wobei jedoch die Art der erzeugten Produkte sowie die Anzahl und Größe der in der Anlage arbeitenden Reaktoren veränderlich sind. Es wird in einem entsprechend großen Behälter gespeichert, so dass dann bei Kühlmittelbedarf neben der direkten Kälteerzeugung auf den Speicherinhalt zurückgegriffen werden kann.

Bei der bisherigen Strategie der Kältemaschinensteuerung wurden jedoch bei weitem nicht alle Möglichkeiten zum wirtschaftlichen Anlagenbetrieb genutzt. Insbesondere blieb die Speicherfähigkeit des Kühlwasserbehälters unberücksichtigt. Der in der Niedrigproduktionsphasen oder während Produktionspausen extrem geringe Gesamtstromverbrauch des Werkes bietet dabei grundsätzlich die Möglichkeit, Kältemaschinen einzuschalten und den Kühlwasserbehälter bis zu seiner Kapazitätsgrenze zu laden. Dadurch lässt sich dann nicht nur Kühlwasservorrat für zukünftigen Bedarf erzeugen, sondern es wird auch der Stromtarifrahmen bei der Steuerung der Kühlwasseraufbereitung mit einbezogen. Andererseits erlaubt ein großer Kühlwasservorrat es auch, Kältemaschinen bei Werkstromspitzen trotz laufender Produktion abzuschalten. Hierdurch lassen sich Energiekosten vermeiden, die durch Verletzung von Stromtarifgrenzen entstehen können.

Unter Verwendung von Fuzzy Control soll die Kühlwassererzeugung so gesteuert werden, dass Kältemaschinen eingeschaltet werden, wenn der Kühlwasserbehälter z.B. nur teilweise gefüllt ist bzw. wenn er sich produktionsbedingt leert oder auch wenn ein bevorstehender Bedarf an Kühlleistung abzusehen ist. Dabei müssen die zur Verfügung stehenden Kältemaschinen je nach Größe des voraussichtlichen Kühlleistungsbedarfs und des Kühlwasservorrats beeinflusst werden, dies aber unter zusätzlicher Berücksichtigung des jeweiligen Werkstroms. Dabei sind die produktionstechnischen Belange besonders zu beachten; eine Werkstromgrenze, bei der Stromverbraucher abgeschaltet werden müssten, muss unter Umständen ignoriert werden, wenn die Produktion oder die Anlagensicherheit durch ungenügende Kühlung gefährdet wäre. Wenn die Kältemaschinen in einen Arbeitsbereich gelangen können, bei dem die erforderliche Kühlwassertemperatur nicht mehr gewährleistet ist, soll dieses durch Minderung des Durchflusses mit Hilfe des Bypasses vermieden werden.

Als Eingangsgrößen des Fuzzy-Systems werden die Prozessgrößen Kühlwasservorrat, Kühlenergiebedarf, Vorlauftemperatur und Werkstrom benutzt, auf deren Grundlage die Entscheidung über Ein-/Ausschalten der einzelnen Kältemaschinen und des Bypasses getroffen werden kann. Werden etwa zur Fuzzifizierung und Defuzzifizierung der Ein-/Ausgangsvariablen jeweils 5 linguistische Werte verwendet, so benötigt man für eine komplette Beschreibung $5^4 = 625$ Regeln bei einer Realisierung mit einer einzigen Regelbasis. In diesem Fall erweist es sich als sinnvoll, das Problem in separierbare Teilprobleme aufzuspalten: In einer ersten Verarbeitungsstufe (Regelbasis) wird aus den ersten drei Eingangsgrößen eine Zwischengröße berechnet, in diesem Fall ein auch anschaulich verständlicher „Kühlleistungssollwert“, bevor in einer zweiten, nachgeschalteten Verarbeitungsstufe (Regelbasis) unter zusätzlicher Beachtung der Wasservorlauftemperatur die eigentlichen Stellgrößen für das Ein-/Ausschalten der Kältemaschinen und des Bypasses bestimmt werden. Durch die (partiell) hierarchische Fuzzy-Logik-Struktur reduziert sich die Anzahl der Regeln auf $5^3 + 5^2 = 150$.

Mit Fuzzy-Anwendungen wird oft die Hoffnung auf eine transparente, gut verständliche und nachvollziehbare Lösung verbunden. Daher ist es wichtig, auf eine klare Strukturierung des Problems und eine möglichst kompakte Darstellung der Regelbasis zu achten. Hilfreich dazu sind auch verallgemeinernde Regeln, die bei mehr als nur einer bestimmten Kombination von linguistischen Eingangswerten gültig sind.

7 Fehlerdiagnose aus Symptomen an einem Gleichstrommotor

Technische Systeme werden immer größer und komplexer. Gleichzeitig sind die Ansprüche an die Sicherheit, Zuverlässigkeit und Verfügbarkeit der technischen Anlagen gestiegen. Daher wird auch die Prozessüberwachung immer wichtiger, wobei nicht nur Grenzwerte und Trends von Signalen überwacht, sondern zunehmend auch modellbasierte mathematische Verfahren eingesetzt werden. Die jeweiligen Schwellenwerte müssen so gewählt werden, dass möglichst wenige Fehlalarme ausgelöst werden. Es ist somit stets ein Kompromiss zwischen hoher Fehlerempfindlichkeit und großer Robustheit gegenüber Störungen (z.B. Rauschen) zu definieren. Solche Kompromisse lassen sich mit Hilfe von Fuzzy-Logik leichter definieren, da keine harten ja/nein-Entscheidungen getroffen werden müssen, sondern auch unscharfe Aussagen der Form „Folgender Fehler ist mit hoher Wahrscheinlichkeit (Möglichkeit im Sinne der Fuzzy-Logik) aufgetreten“ nützlich sind. Auch weil zur Diagnose, d.h. Klassifikation eines Fehlers meist mehrere Symptome verwendet werden, deren Korrespondenzen oft nur unscharf bekannt sind, bietet sich zu Fehlerdiagnose der Einsatz von Fuzzy-Logik besonders an.

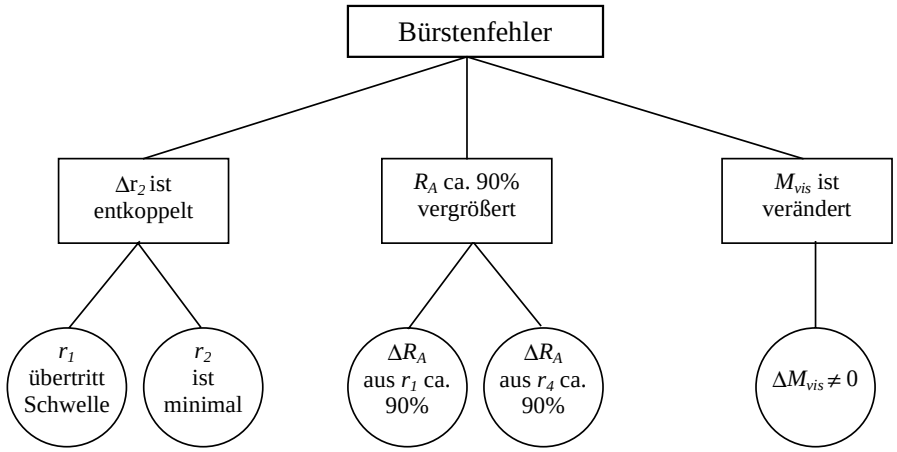


Bild 7 Fehler-Symptom-Baum für den Bürstenfehler an einem Gleichstrommotor

In diesem Beispiel wird anhand einer Simulation eines Gleichstrommotors und einiger Fehler eine einfache Fuzzy-Fehlerdiagnose demonstriert. Dabei wird die Methode der Paritätsgleichungen zur Residuenerzeugung, d.h. zur modellbasierten Generierung von Symptomen benutzt [8.], [9.] . Die Differenz zwischen gemessen und geschätzten („beobachteten“) Signalen wird als Residuum (lat. residuum = Rest) bezeichnet. Im fehlerfreien Fall sollten die Residuen sehr kleine Werte annehmen. Bei einem Fehler muss man dagegen an einem oder mehreren Residuen Werte erkennen können, die

deutlich von Null abweichen. Zu Erkennung solcher Abweichungen wird für jedes Residuum ein Toleranzband festgelegt, dessen Überschreitung einen Fehler signalisiert.

Zur Generierung solcher Residuen r_i für einen gegebenen Prozess müssen Gleichungen gefunden werden, die die Messgrößen auf entsprechende Weise miteinander verknüpfen. Diese Gleichungen werden in Anlehnung an den Begriff Paritätsbit, welches in der Nachrichtentechnik als Prüfsumme mehrere Bits eines Datenwortes miteinander verknüpft, als Paritätsgleichungen bezeichnet. Zur Aufstellung der Paritätsgleichungen ist ein dynamisches, physikalisches Prozessmodell erforderlich. Zusätzlich kommen auch Parameterschätzverfahren für einzelne Prozessparameter zum Einsatz.

In diesem Beispiel wird ein System vorgestellt, welches drei Fehler eines Gleichstrommotors mit Permanentmagneterregung erkennen kann: Bürstenfehler, Veränderung des Ankerwiderstandes R_A und Messfehler (Sensorfehler) bei der Ankerspannung U_A.

Die logischen Zusammenhänge zwischen beobachteten Symptomen und erkannten Fehlern sind in einem sog. Fehler-Symptom-Baum Bild 7 dargestellt, der von unten nach oben ausgewertet wird. Man beachte, dass die Richtung des logischen Schließens dabei der physikalischen Kausalität genau entgegengesetzt ist. Das physikalische Modell besagt: „Wenn ein bestimmter Fehler auftritt, dann entstehen folgende Symptome“. Bei der Fehlerdiagnose sollen dagegen Schlüsse der Form „Wenn bestimmte Symptome zu beobachten sind, dann liegt wahrscheinlich folgender Fehler zugrunde“ gezogen werden.

8 Literatur

[1.] Krone, A.: *Generierung von Fuzzy-Regeln mit WINROSA.* Kassel, VDI Berichte 1381, September 1997.

[2.] Pfeiffer, B-M.: *Selbsteinstellende klassische Regler mit Fuzzy-Logik.* 2. Workshop "Fuzzy Control" des GMA-UA 1.4.2., Dortmund, November 1992, S. 285-298

[3.] Pfeiffer, B-M.: *Einsatz von Fuzzy-Logik in lernfähigen digitalen Regelsystemen.* Fortschrittbericht VDI, Reihe 8, Nr. 500, VDI-Verlag, Düsseldorf, 1995.

[4.] Schädel, H.M., Ctistis, Ch., Nikolai, D.: *Fuzzy-Adaption von PI-Reglern im geschlossenen Regelkreis ohne Prozesskenntnis*. 9. Workshop "Fuzzy Control" des GMA-FA 5.2.2., Dortmund, November 1999, S. 270-283

[5.] Isermann R. *Digitale Regelsysteme I.* Springer-Verlag Berlin, 1988

[6.] Adamy, J. *„Breakout Prediction for Continous Casting by Fuzzy Mealy Automata“*, Proc. of the 3rd European Congress on Intelligent Techniques and Soft Computing EUFIT, Aachen, 1995, S. 754-759

[7.] Bork P. *„Fuzzy Control zur Optimierung der Kühlwasseraufbereitung an einer Chemie-Reaktoranlage“*, Automatisierungstechnische Praxis atp 35 (1993) Heft 5

[8.] Höfling T. *„Methoden zur Fehlererkennung mit Parameterschätzung und Paritätsgleichungen“*, VDI-Verlag Reihe 8, Nr. 546, Düsseldorf 1996

[9.] Füssel, D., Ballé, P., Moseler, O., Willimowski, M., Höfling, Th. (1998) "Residuenbasierte Fehlererkennung und Diagnose an komplexen Prozessen", at-Automatisierungstechnik, Vol. 46, No. 9, S. 435-443