

Soft Computing in der Automatisierungstechnik

Publikation aus der ELEKTRONIK Nr. 24 und 26 1998
Fachzeitschrift für industrielle Anwender und Entwickler

Dipl.-Ing. (FH) Peter Auer

Rathenastr.41

33102 Paderborn

Peter.Auer@hotmail.com

Abstract : Die steigende Komplexität der Prozesse und die hohen Anforderungen an die Algorithmen in der Automatisierungstechnik fordern neue Konzepte und Verfahren. Dabei greifen zusehends Verfahren, die menschliche bzw. an der Natur orientierte Vorgehensweisen nachbilden - das Schlagwort dieses Trends heißt Soft Computing. Soft Computing besteht aus drei großen Themenblöcken, der Fuzzy Logic, den Neuronalen Netzen und den Evolutionären Algorithmen.

Stichwörter : Soft Computing, Fuzzy Logic, Neuronale Netze, Evolutionäre Algorithmen

1. Soft Computing

Seit der Erfindung des Rechners herrscht die Meinung vor, jedes Problem lösen zu können, wenn es mathematisch präzise und vollständig beschrieben ist. Mit zunehmender Komplexität wird es jedoch immer schwieriger ein System mathematisch zu beschreiben.

" In dem Maße, wie die Komplexität eines Systems zunimmt, nimmt die Fähigkeit ab, präzise und doch signifikante Aussagen über sein Verhalten zu machen. "

Lotfi Zadeh 1973

Bild 1 Unvereinbarkeit von Komplexität und Präzision

Hierbei macht sich das Prinzip der Unvereinbarkeit von hoher Komplexität und Präzision bemerkbar, das von Lotfi Zadeh 1973 aufgestellt wurde (Bild 1).

Die Gebiete *Fuzzy Logic*, *neuronale Netze*, *evolutionäre Algorithmen* sowie *Möglichkeitstheorie* werden immer mehr unter dem Begriff *Soft Computing* zusammengefaßt. Die Bezeichnung Soft Computing wurde von Lotfi Zadeh auf der International Conference on Fuzzy Logic and Neural Nets in Iizuka, Japan 1992 eingeführt. Lotfi Zadeh sagt : Soft Computing (weiches Rechnen) unterscheidet sich von dem konventionellen Hard Computing (hartes Rechnen), indem Soft Computing toleranter in bezug auf Ungenauigkeiten und Unsicherheiten ist.

- **Fuzzy Logic** kann unscharfes Wissen verarbeiten und Erfahrungswissen in technischen Systemen repräsentieren .
- **Möglichkeitstheorie** kann unsicheres Wissen verarbeiten.
- **Evolutionäre Algorithmen** optimieren Parameter und Prozesse.
- **Neuronale Netzwerke** sind lernfähig und können aus Daten Wissen extrahieren.

Hybridsysteme aus den einzelnen Komponenten ergeben neue Systeme wie "neurofuzzy" oder "neurogenetic". Es erscheint sinnvoll, Soft Computing mit konventionellen Methoden zu kombinieren und je nach Aufgabenstellung mit unterschiedlichen Schwerpunkten gemeinsam einzusetzen.

2. Fuzzy Logic

Der Grundgedanke der Fuzzy Logic ist, daß sich nicht alle Ereignisse und Beobachtungen in wahr und nicht wahr klassifizieren lassen. Lotfi Zadeh benannte 1965 seine neue Idee mit dem Namen Fuzzy Logic, was in etwa mit "unscharfe Logic" zu übersetzen ist. Hinter der Fuzzy Logic verbirgt sich eine Verallgemeinerung der klassischen zweiwertigen Logik. Der Begriff Fuzzy Logic bezeichnet nicht eine bestimmte mathematische Logik, sondern eine Theorie der "unscharfen Mengen".

Eine wichtige Grundlage der Logik ist die Mengenlehre, die auf Georg Cantor zurückgeht. Nach der klassischen Theorie der Mengenlehre gehört ein Element zu einer Menge oder nicht. Etwas dazwischen erlaubt die klassische Theorie nicht. Weiterhin kann ein Element nicht gleichzeitig wahr und falsch sein. Die Logik beschäftigt sich mit Methoden die eine Schlußfolgerung auf der Basis von logischen Aussagen und Mengen erzeugen. Nehmen wir als beispiel die Aussage "Wenn Temperatur $> 30^{\circ}\text{C}$ und Feuchte $> 60\%$ dann Ventilator ein". Diese Aussage besitzt einen Wahrheitsgehalt, der wahr oder falsch ist. Die Schlußfolgerung "Ventilator ein" kann nur wahr sein, wenn die beiden Prämissen wahr sind.

Was ist aber mit den alltäglichen logischen Aussagen, wie : "Wenn Temperatur warm und Feuchte hoch, dann Ventilatorgeschwindigkeit hoch". Der Begriff "warm" ist ein "unscharfer Begriff", er gibt nicht eine bestimmte Temperatur, sondern einen unscharfen Bereich an. Die Umsetzung der menschlichen Ausdrucksweise in eine mathematische Beschreibung, die in dieser Form auch einem Computer zugänglich gemacht werden kann, ist mit der Fuzzy Logic möglich. Im Gegensatz zur klassischen Mengenlehre, bei der jedes Element entweder zu einer bestimmten Menge gehört oder nicht, können die einzelnen Elemente der unscharfen Menge dieser auch zu einem bestimmten Grad im Intervall zwischen 0 und 1 angehören. (Bild 2) Ob es warm oder heiß ist, muß nicht mehr mit einer scharfen Grenze definiert werden. Ein Temperaturwert kann zu einem gewissen Grad zu der Menge "warm" und ebenso zu "heiß" gehören. Die klassische Logik erlaubt es, die scharfen Mengen in Verknüpfung wie UND, ODER bzw. Negation zu setzen. Für die unscharfen Mengen gibt es ebenfalls Verknüpfungsoperatoren, die auch für die klassischen scharfen Mengen gelten. Die klassischen Logik kann somit als ein Sonderfall der Fuzzy Logic angesehen werden bzw. Fuzzy Logic ist die Verallgemeinerung der klassischen Logik.

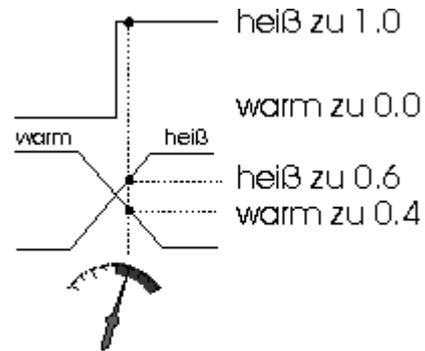


Bild 2 Die Zugehörigkeitsfunktion

Die Fuzzy Regelung

Fuzzy Regelung ist die Anwendung der Fuzzy Logic zur Automatisierung von technischen Systemen. Das besondere ist, die unmittelbare Verwendung von verbal formuliertem Erfahrungswissen in Form von **WENN-DANN-Regeln**. Beim Einparken beispielsweise sagt man nicht "30° nach links", sondern "ein bißchen nach links". Und trotz dem können wir aus den vagen und unsicheren Daten präzise und vernünftige Entscheidungen treffen. Diese Methode entspricht dem intuitiven menschlichen Denken und unterscheidet sich wesentlich von der herkömmlichen Entwurfsmethodik, die von einem mathematischen Modelle ausgeht. Im Unterschied zum klassischen Regler bildet der Fuzzy Regler das Stellsignal durch Verarbeitung unscharfer Signale (Bild 3). Dazu sind die Ein- und Ausgagssignale des FuzzyReglers in linguistische Variablen zu transformieren. Hierbei wird mit der sprachlichen Unschärfe von Wörtern wie "warm, heiß, klein, groß, ein bißchen warm" usw. gearbeitet.

Dies ermöglicht es, das Wissen mit wenigen Klassen ohne Informationsverlust zu verarbeiten. Die Signalverarbeitung innerhalb des Fuzzy Systems erfolgt dann mit Regeln der Art : "Wenn Temperatur warm und Feuchte hoch, dann Kühlung stark". Alle solche Regeln sind in der Regelbasis gespeichert und werden dort ausgewertet (Inferenz). Mit Hilfe der Regelbasis formuliert der Entwickler seine Strategie. Das Ergebnis der Inferenz wird dann durch Defuzzifizierung in eine scharfe Stellgröße transformiert. Von außen gesehen zeigt das Fuzzy System keine "Unschärfe", nur innerhalb des Systems kommt die "Unschärfe" zum Tragen. Die Regeln entsprechen der menschlichen Denkweise und sind leicht verständlich. Bereits mit wenigen Regeln lassen sich komplexe Strategien realisieren. Durch Hinzufügen von einzelnen Regeln läßt sich die Funktion lokal einfach erweitern, ohne das gesamte System zu verändern. Durch die lokale Wirkung der Regeln und die Überlappung der unscharfen Mengen steigen die Robustheit und die Unempfindlichkeit.

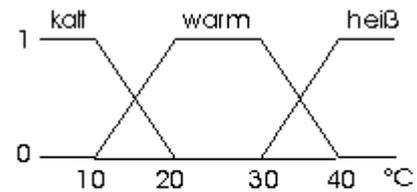


Bild 3 Linguistische Variablen : Die sprachliche Unschärfe wie "warm" oder "kalt" sind an keine scharfen Grenzen gebunden.

Zwischen den Ein- und Ausgängen des Fuzzy-Systems entsteht eine statische, nichtlineare Übertragungsfunktion in Form eines Kennfeldes (Bild 4). Das bedeutet : Für einen gegebenen Eingangsvektor liefert das System immer den gleichen Ausgangsvektor. Dem statischen Fuzzy-System wird - wenn notwendig - am Ein- und/oder Ausgang eine externe Dynamik in Form von Verzögerungsgliedern vorgeschaltet (Differenzieren, Integrieren).

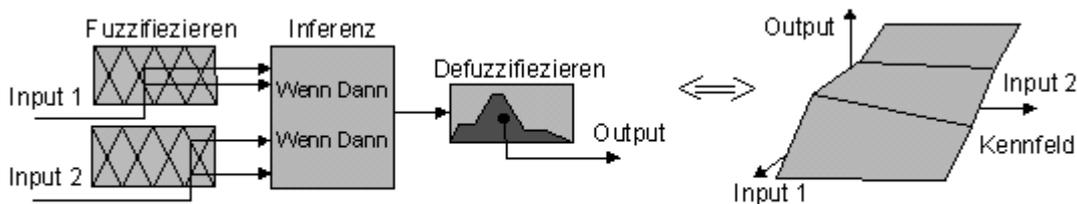


Bild 4 Ein Fuzzy-System mit zwei Eingängen und einem Ausgang sowie einem äquivalenten Kennfeld.

Folgende Vorteile der Fuzzy Regelung sind hervorzuheben :

- Verarbeitung von unscharfen Daten in technischen Systemen;
- schnelle Realisierung von Regelstrategien ohne mathematische Modelle;
- das Erfahrungswissen über den Prozeß kann direkt in die Regelstrategie einfließen;
- gut geeignet für eine Mehrgrößenregelung;
- effizient Behandlung von Nichtlinearitäten;
- gute Wartbarkeit der Systeme.

Dazu sind die folgenden Nachteile zu nennen :

- Der Stabilitätsnachweis erfordert einen hohen mathematischen Aufwand;
- keine systematische Entwurfsmethode(trial-and-error);
- große Zahl an Freiheitsgraden erschwert eine gezielte Optimierung;
- die Fuzzy-Regelung ist nicht lernfähig.

Die Lernfähigkeit und die Optimierungsmethoden lassen sich mit Neuronalen Netzwerken bzw. Evolutionären Algorithmen verbessern. Da die meisten Fuzzy Komponenten in der Automatisierungstechnik reine Softwarelösungen sind, liefern die meisten Fuzzy Entwicklungstools einen hochoptimierten C-Code. Durch die reinen Softwarelösungen und den steigenden Anteil von Fuzzy-Komponenten in der industriellen Automatisierungstechnik bekam die Forderung nach genormten Entwurfs-, Programmier- und Inbetrieb-

nahmewerkzeugen. Aus diesen Gründen wird Fuzzy Control in der IEC 1131-3 integriert. Die ersten Programmiersprachen, die dieser Norm entsprechen, sind bereits auf dem Markt.

Strukturen von Fuzzy-Reglern

Für den Einsatz der Fuzzy-Regelung stellt sich die Frage nach der geeigneten Einbindung der Fuzzy-Komponenten in den Regelkreis und nach der geeigneten Einstellung der Fuzzy-Parameter. Bei der Parameter-Optimierung können die Evolutionsstrategien (werden später beschrieben) gut angewendet werden. Die Wahl der Reglerstruktur ist keineswegs eine leichte Aufgabe. In den meisten Fällen denkt man beim Fuzzy-Regler an den Ersatz des konventionellen Reglers. Fuzzy Systeme können aber auch adaptiv oder in Kombination mit konventionellen Reglern eingesetzt werden. Deshalb nun ein kleiner Überblick über Reglerstrukturen und Anwendungsgebiete in der Automatisierungstechnik.

- Abbildung eines Experten auf einen Fuzzy Regler : Ist die direkte Umsetzung einer Strategie in ein Fuzzy-Regler; dies wird meist bei sehr komplexen Systemen angewendet, die bisher nur vom Menschen geregelt wurden.
- PID-Regler : Durch die Nachbildung konventioneller Regler mit Fuzzy-Regler versucht man, bessere Regeleigenschaften zu erreichen. Hierzu gehört vor allem PID-Regler, der durch die Fuzzy-Komponente nichtlineares Verhalten bekommt.
- Fuzzy-Zustandsregler : Beim klassischen Zustandsregler werden die Zustandsgrößen mit festen Faktoren gewichtet und auf die Stellgröße addiert. Werden diese Zustandsgrößen über Fuzzy-Entscheidungsregeln auf die Stellgröße addiert, so ergibt sich ein nichtlinearer Fuzzy-Zustandsregler.
- Variables Führungs- und Störverhalten : Wird der Sollwert als Eingangssignal des Fuzzy Reglers benutzt, so kann man entscheiden, ob es sich um eine Führungs- oder Störgrößen-Änderung handelt, und dementsprechend reagieren.
- Multi-Arbeitspunkt-Fuzzy-Regler : Werden weitere Signale vom Prozeß als Eingang des Fuzzy-Reglers benutzt, so können diese Informationen in der Regelbasis berücksichtigt werden.
- Adaption der Reglerparameter : Ziel der Adaption ist es, die Reglerparameter des konventionellen Reglers mittels Fuzzy-Komponente zu adaptieren. Die Vorteile eines konventionellen Reglers bleiben hierbei erhalten. Speziell die Interpolation des Fuzzy Systems zwischen den Regeln ist von großer Bedeutung für Adaptive Regelsysteme.
- Umschalten zwischen konventionellen und Fuzzy-Regler : Erfolgt in Abhängigkeit von geeigneten Informationen zwischen Fuzzy und konventionellem System.
- Sollwert-Generierung mit Fuzzy-System : Aus verschiedenen Informationen kann mittels eines Fuzzy-Systems eine geeignete Führungsgröße für einen Regelkreis generiert werden.
- Fuzzy-Modellbildung : Der Grundgedanke der Fuzzy-Modellbildung besteht darin, mittels linguistischer Regeln das Ein- und Ausgangsverhalten eines nichtlinearen Systems zu beschreiben. Solche Fuzzy-Prozeßmodelle können für die modellbasierte Regelung verwendet werden.
- Fuzzy Logic und Sensorik : Einige Sensoren werden mit Fuzzy Logic ausgestattet, um eine zentrale Intelligenz zu erlangen. Einfache und preiswerte Sensoren können mit Fuzzy Logic für komplexe Aufgaben benutzt werden. Aus den Signalen lassen sich beispielsweise nichtmeßbare Größen abgeleitet oder aus mehreren Signalen neue Informationen extrahieren.

Weitere Einsatzgebiete der Fuzzy Logic sind sehr breit, sie reichen von der Datenanalyse über Bildverarbeitung, Fehlerdiagnose, Überwachungssysteme, Qualitätskontrolle, Prozeßleittechnik, Expertensysteme, unscharfe Petri-Netze, Steuerungssysteme mit SPS bis

zur Robotik. Fuzzy Logic findet ebenfalls Anwendungen in nicht-technischen Bereichen. Einige der Gebiete werden bei den Neuronalen Netzwerken genauer beschrieben.

3. Möglichkeitstheorie

Lange Zeit war das Rechnen mit Wahrscheinlichkeit die einzige mathematisch formulierte Theorie, die sich mit unvollständigen Informationen befaßte. Bei industriellen Anwendungen ist es aber nicht immer angebracht, von Wahrscheinlichkeiten zu sprechen, sondern man möchte von "Möglichkeiten" sprechen. 1978 entwickelt Lotfi Zadeh die Theorie der Möglichkeit (eng. Possibility). Demzufolge läßt sich die Möglichkeit eines Ereignisses - ebenso wie die Wahrscheinlichkeit - als eine Zahl zwischen 0 und 1 definieren. Der Unterschied besteht nur darin, daß die Möglichkeit anderen Gesetzen gehorcht. Um den Unterschied zwischen Möglichkeit und Wahrscheinlichkeit zu zeigen, nehmen wir das anschauliche Beispiel von Lotfi Zadeh :

Karola nimmt jeden Tag in ihrem Auto ihre Kolleginnen mit zur Arbeit. Wie viele Kolleginnen nimmt sie mit? Um die Möglichkeit zu bestimmen, müssen wir Karola einem Test unterziehen. Wie viele Kolleginnen kann sie mitnehmen? Nehmen wir einen herkömmlichen PKW, so darf sie höchstens vier Personen im Auto befördern. Bei allen weiteren Personen nimmt die Möglichkeit ab. (Bild 5) Fünf Personen sind ja noch möglich, aber 8 Personen sind unmöglich. Zur Bestimmung der

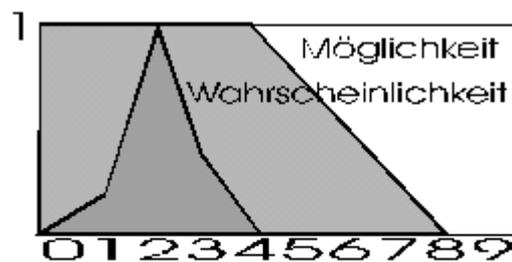


Bild 5 Die Möglichkeitstheorie:
Was völlig unwahrscheinlich ist,
kann dennoch möglich sein.

Wahrscheinlichkeit dagegen, müssen wir Karola mehrere Tage auf der Fahrt zur Arbeit beobachten und die Anzahl der Personen im Auto aufschreiben. Nach einem Monat z.B. kennen wir die Häufigkeit, wie oft Karola keine, eine, zwei usw. Personen mitgenommen hat. Die Ergebnisse der Möglichkeit und Wahrscheinlichkeit stellen wir in einem gemeinsamen Diagramm dar. (Bild 5) Die Kurve der Möglichkeit erinnert sehr stark an Zugehörigkeitsfunktionen. Die Kurve besagt, daß es Karola zum Grad 1 möglich ist keine, eine, zwei, drei oder vier Personen mitzunehmen. Ab vier Personen nimmt der Grad der Möglichkeit ab, bis bei acht Personen der Grad null wird, was unmöglich bedeutet. Aus der Wahrscheinlichkeit kann man lediglich ablesen : Wenn man Karola an irgendeinem Tag bei der Fahrt zur Arbeit trifft, wird sie mit hoher Wahrscheinlichkeit 2 Kolleginnen im Auto mitnehmen. Was ist aber, wenn Karola allein zur Arbeit fährt. Die Wahrscheinlichkeit versagt hier absolut, denn sie sagt, daß es völlig unwahrscheinlich ist. Die Möglichkeitstheorie sagt, daß es möglich ist. Bei sieben Personen versagt die Wahrscheinlichkeit ebenfalls, und die Möglichkeit besagt, daß es noch zu 30% möglich ist. Aus den Kurven kann man folgendes ablesen "**Was wahrscheinlich ist, ist auch möglich, aber was völlig unwahrscheinlich ist, kann trotzdem möglich sein.**"

Mit der Möglichkeitstheorie sind Zuverlässigkeitsaussagen über technische Systeme möglich, die bislang nur mit der Wahrscheinlichkeitstheorie gemacht wurden. Man spricht dann z.B. nicht mehr von wahrscheinlichen Fehlerauftritt, sondern von einem möglichen Fehlerauftritt. Die Möglichkeit ersetzt nicht die Wahrscheinlichkeit, sondern ergänzt sie.

4. Neuronale Netzwerke

Die ersten Ideen für Neuronale Netze reichen bis in die 60er Jahren zurück, gerieten damals aber in Vergessenheit, da der Ausbau des "*von Neumann Konzeptes*" vielversprechender

erschien. Die Motivation und die Grundlagen für die Neuronale Netze wurden dem menschlichen Gehirn abgeschaut. In den letzten Jahren haben die Neuronale Netze ein großes Interesse in der Technik erfahren.

Künstliche neuronale Netzwerke

In Analogie zum menschlichen Gehirn besteht ein künstliches Neuronales Netzwerk aus einer Vielzahl von Künstlichen Neuronen, die in Schichten angeordnet sind und Informationen parallel verarbeiten. Das Künstliche Neuron (Bild 6) besteht aus mehreren Eingängen und einem Ausgang. Die Umformung der Aktivitäten im Neuron geschieht in drei Stufen. Zuerst multipliziert das Neuron die eingehenden Aktivitäten mit den Gewichten der zugehörigen Verbindungen und summiert diese Eingaben zu einer Gesamteingabe. Dazu kann auch ein Offset aufsummiert. Auf diese Gesamteingabe wendet das Neuron die sogenannte Transferfunktion an, woraus sich die ausgehende Aktivität ergibt.

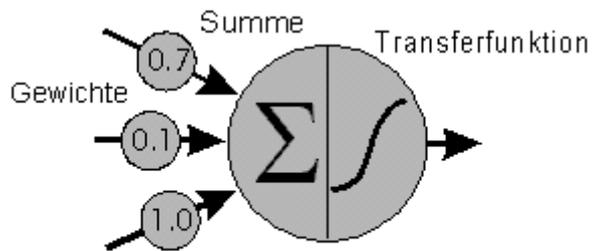


Bild 6. Das Künstliche Neuron: dem menschlichen Gehirn abgeschaut.

Die bekannteste Struktur eines Neuronalen Netzes ist die dreischichtige "feed forward"-Struktur (Bild 7). Die erste Schicht empfängt die ankommenden Input.-Signale und verteilt sie an die zweite Schicht. Die zweite Schicht wird Hidden Layer (versteckte Schicht) genannt, weil sie keine Verbindung nach außen hat. Die dritte Schicht gibt die Ergebnisse an die Außenwelt. Die Netzform heißt "feed forward", weil alle Signale nur von der Eingabe- zu der Ausgabeschicht geleitet werden. Es gibt keine internen Rückkopplungen. Bei den "feedback" Netzwerken werden Signale rückwärts innerhalb des Netzwerkes weitergeleitet. Zwischen den Ein- und Ausgangsneuronen eines neuronalen Netzes entsteht eine multidimensionale Übertragungsfunktion, deren Komplexität von der Anzahl der Neuronen abhängig ist. Um eine bestimmte Übertragungsfunktion dem Netzwerk beizubringen, muß es mit einem Lernverfahren trainiert werden.

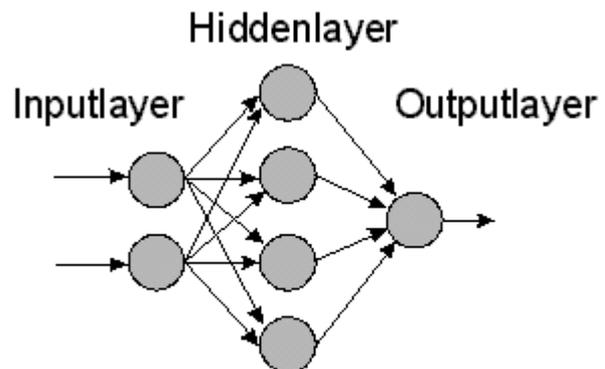


Bild 7. Das Neuronale Netzwerk in dreischichtiger "feed forward" Struktur.

Die unterschiedlichen Lernverfahren

Die Aufgabe des Lernverfahren liegt in der Einstellung der Gewichte. Die Lernregel sorgt dafür, daß sich während des Trainings schrittweise sinnvolle Werte der Gewichte herausbilden. Dies entspricht dem Lernverhalten des Menschen. Wenn ein Kind laufen lernt, so macht es anfangs noch viele Fehler, die sich aber nach und nach vermindern. Als Beispiel soll das Modell eines Prozesses erlernt werden (Bild 8). Zuerst präsentieren wir dem Netze einige Input- und Output-Signale von dem realen Prozeß. Das Neuronale Netz generiert gleichzeitig aus den Input-Signalen dessen Output-Signale. Zwischen jedem Output-Signal vom Prozeß und dem des Neuronalen Netzes wird eine Differenz gebildet, um zu sehen, wie die Ausgabe des Netzes mit dem Prozeß übereinstimmt. Dann werden die Gewichte aller

Verbindungen derart verändert, daß der Fehler kleiner wird und das Netz eine bessere Annäherung an das gewünschte Ergebnis erzielt. Diese Trainingsprozedur wird so oft mit den Daten wiederholt, bis der Fehler Null ist bzw. eine definierte Schranke erreicht hat. Danach zeigt man dem Neuronalen Netzwerk Eingangssignale aus dem Prozeß, die es beim Lernen nie gesehen hat. Antwortet das Netzwerk richtig, dann ist das Lernverfahren abgeschlossen. Das Netzwerk bildet den Prozeß nach.

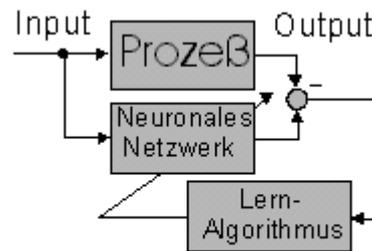


Bild 8. Die Lernmethode: schrittweise Annäherung.

Das Verändern der Gewichte wird z.B. mit dem **Backpropagation-Algorithmus** durchgeführt. Man reicht den Fehler von der Ausgangschiene an die vorhergehenden Schichten zurück, bis die Eingangschiene erreicht ist, daher der Name "Rückwärts-Vorhersage". Gleichzeitig werden die Gewichte in allen Schichten angepaßt. Diese Lernmethode gehört zu der "**Supervised Learning**"-Methode, weil dem Netzwerk die gewünschte Reaktion beim Lernen gezeigt wird. Man kann es mit einem Lehrer-Schüler-Lernverfahren vergleichen. Der Lehrer (Prozeß) zeigt dem Schüler (Netzwerk) die Input- und die Outputsignale, und der Schüler lernt das. Bei der **Reinforcement**-Lernmethode teilt der Lehrer dem Schüler lediglich ein skalares Gütemaß der Leistung und nicht den gewünschten Zustand am Ausgang mit. Beim "**Unsupervised Learning**" macht der Lehrer gar keine Angaben zu den Ausgangsaktivitäten (Kohonen-Netzwerk).

Neuronales Netz und herkömmliche Algorithmen

Konventionelle Algorithmen sind extrem anfällig gegenüber Fehlern, ein einziges falsches Signal, eine einzige 1 statt einer 0, also ein einziges Bit, kann ein ganzes Programm zunichte machen. Neuronale Netze sind dagegen fehlertolerant; fehlen einige Impulse oder sind sie verwechselt, so zeigt das Neuronale Netz eine größere Robustheit. Herkömmliche Algorithmen benötigen Programme, die ihnen präzise vorschreiben, was sie zu tun haben. Neuronale Netze dagegen lernen aus Daten und versuchen, immer besser zu werden. Das Erstellen von komplexen nichtlinearen Algorithmen benötigt einen hohen mathematischen und meßtechnischen Aufwand. Neuronale Netze lernen aus Daten und erzeugen komplexe nichtlineare Systeme. Neuronale Netze brauchen teilweise lange zum Erlernen der Aufgabe, aber benötigen wenig mathematischen Aufwand und können somit die Entwicklungszeit verkürzen.

Neuronale Netze in der Regelungstechnik

Um die Leistung und Dynamik moderner Regelung optimal ausnutzen zu können, ist der Einsatz nichtlinearer Regelungsverfahren vorteilhaft. Der Einsatz Neuronaler Netzwerke als einen universellen Ansatz zur Repräsentation des nichtlinearen Verhaltens eines Reglers bietet sich hier an.

Lernen vom Experten

Neuronale Netzwerke sowie Fuzzy-Systeme eignen sich besonders zu der "Nachahmung" des menschlichen Experten. Der Mensch kann hoch komplexe Systeme, die analytisch schwierig zu beschreiben sind, noch zufriedenstellend steuern. Das Rückwärtsfahren eines LKWs mit einem Anhänger bereitet den LKW-Fahrer weniger Probleme. Für einen konventionellen

Regelalgorithmus ist das eine Herausforderung. Mit einem Neuronales Netzwerk läßt sich diese Aufgabe effizienter lösen. (Bild 9) Wir nehmen alle relevanten Stelleingriffe des LKW-Fahrers, wie z.B. Auslenkung, Gas, Bremse usw. auf einigen Rückwärtsfahrten auf und trainieren mit diesen Daten ein Neuronales Netzwerk. Konvergiert das Netzwerk, so haben wir die Strategie und die Erfahrung des LKW Fahrers in einem Algorithmus. Diese Vorgehensweise bietet sich immer dann, wenn ein Prozeß bereits vom Menschen hinreichend gut geregelt wird und man ihn automatisieren möchte.

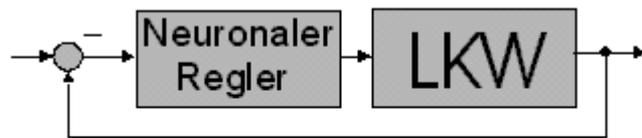


Bild 9. Lernen vom Experten: Die Strategie und Erfahrung des LKW-Fahrers stecken in dem Neuronalen Regler.

Die Modell-Identifikation

Für die Erstellung von linearen statischen oder dynamischen Modellen gibt es hinreichend gute Methoden und Softwaretools. Für nichtlineare Modelle ergeben sich dennoch große Schwierigkeiten. Hier lassen sich die Neuronales Netzwerke effizient eingesetzt. Die Identifikationsaufgabe besteht in der Approximation einer unbekanntenen Funktion. Dem statischen Neuronales Netzwerk wird eine externe Dynamik vorgeschaltet. (Bild 10) Hierfür werden Vergangenheitswerte der Prozeß-meßgrößen über Verzögerungsglieder dem statischen Netz zur Verfügung gestellt. Das Neuronales Netz selbst ist statisch, d.h. seine Neuronen enthalten keine Gedächtniselemente. Die Anzahl der Verzögerungsglieder pro Ein- und Ausgang kann aus der Differenzgleichung, die den Prozeß beschreibt, ermittelt werden. Zwischen den Ein- und Ausgängen des Netzes entsteht eine multidimensionale Übertragungsfunktion. Die Identifikation nichtlinearer, dynamischer Systeme hat große Bedeutung in den Bereichen Regelung, Simulation, Prädiktion, Fehlererkennung und Fehlerdiagnose. Ein gutes Modell ist dabei die Grundlage all dieser Methoden. Der Vorteil der neuronalen Modellbildung ist, daß die Neuronales Netze von Haus aus auf Systeme mit mehreren Ein- und Ausgängen und mit nichtlinearem Übertragungsverhalten zugeschnitten sind und daß dabei keine detaillierten Kenntnisse über die inneren Prozeßvorgänge nötig sind.

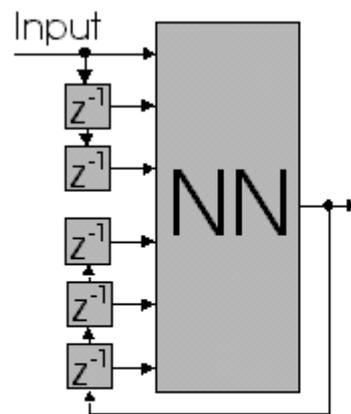


Bild 10. Die Modellidentifikation: Die dynamischen Größen sind dem Neuronales Netz vorgeschaltet.

Die Kompensation der Nichtlinearitäten

Es ist möglich, einem Neuronales Netzwerk die inverse Übertragungsfunktion eines Prozesses beizubringen. Es läßt sich dann zur Kompensation der Nichtlinearitäten im Prozeß nutzen. Das inverse Modell wird



Bild 11. Das Inverse der Strecke: Starke Nichtlinearitäten lassen sich so beseitigen.

in Reihe vor die reelle Strecke im Regelkreis eingebaut (Bild 11). Auf derartige Weise lassen sich stark nichtlineare Systeme linearisieren und somit mit einem linearen Regler steuern.

Die modellbasierte Regelung

Neuronale Netzwerke eignen sich auch für die modellbasierte Regelung. Ein nichtlineares Neuronales Modell der Regelstrecke wird der realen Regelstrecke parallelgeschaltet und bekommt dieselbe Stellgröße (Bild 12). Aus der Differenz zwischen der realen Regelstrecke und dem Neuronalen Modell ergeben sich Rückschlüsse auf Störgrößen. Dementsprechend läßt sich dann auf den Regler einwirken.

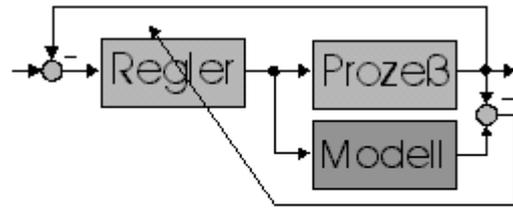


Bild 12. Die modellbasierte Regelung lebt von der Parallelschaltung der realen Regelstrecke und dem Modell.

Die Fehlerdiagnose und Überwachung

Ziel der Fehlerdiagnose ist, aus der Differenz zwischen simulierten und gemessenen Prozeßsignalen mögliche Fehler im Prozeß festzustellen (Bild 13). Das Modell erhält die nötigen Meßdaten vom Prozeß. Wenn die Differenz zwischen den Ausgängen von Modell und Prozeß 0 ist, dann ist der Prozeß fehlerfrei.

Andernfalls wird aus der Differenz der Fehler lokalisiert. Die Leistungsfähigkeit der Fehlerdiagnose hängt hauptsächlich von der Güte des Modells ab. Bei komplexen und stark nichtlinearen Prozessen ist das mathematische Modell sehr schwierig zu erlangen. Hier bieten sich die Neuronale Netzwerke an. Aus Meßdaten des Prozesses muß lediglich das Modell identifiziert werden. Die Fehlerdiagnose mit konventionellen Modellen in Form von Differentialgleichungen erfordert einen hohen mathematischen Aufwand und muß bei jeder Änderung des Prozesses neu aufgestellt werden. Neuronale Modelle bieten hier eine schnellere und flexiblere Anwendung in industriellen Anwendung.

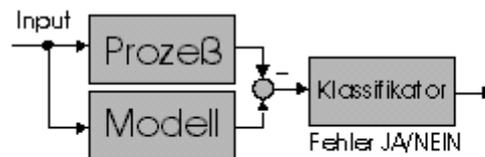


Bild 13. Die Fehlerdiagnose: Das Modell bestimmt die Güte der Diagnose.

Die Qualitätskontrolle

Bei der Qualitätskontrolle in der Produktion müssen Teile auf gut/schlecht untersucht werden. Prüfungen, die akustische bzw. optische Untersuchungen verlangen, werden oft noch von Menschen durchgeführt. Die Automatisierung solcher Verfahren liefert enorme Mengen an Informatinsdaten. Die Klassifikation der Daten auf gut/schlecht erfordert sehr komplexe Algorithmen. Hier bieten Neuronale Netzwerke eine leistungsstarke Hilfe, indem sie die Klasifikation vornehmen (Bild 14). Anhand von Meßdaten von schlechten und guten Teilen wird das Neuronale Netzwerk trainiert. In der Produktion erkennt dann das Netzwerk die Teile nach Güte.

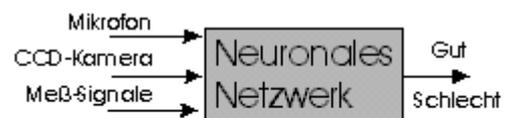


Bild 14. Die Qualitätskontrolle: Das Neuronale Netz wird mit guten und schlechten Meßdaten gefüttert.

Die Datenanalyse

Unter Datenanalyse versteht man, aus vorhandenen Daten neue Informationen zu gewinnen. Mit den "Unsupervised Learning"-Methode lassen sich aus Meßdaten Zusammenhänge herausfinden, die mit klassischen statistischen Methoden nicht möglich sind. Aus den Meßsignalen werden strukturelle Ähnlichkeiten in Klassen zusammengefaßt (Bild 15). Die Klassenbildung teilt ähnliche Objekte in die gleiche Klasse und entsprechend unähnliche Objekte in andere Klassen.

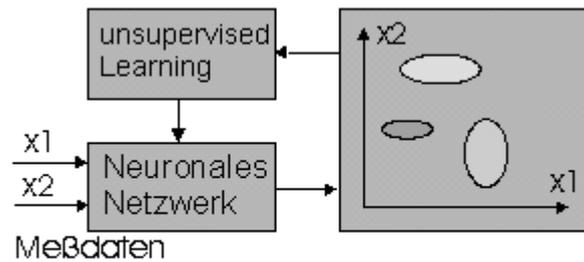


Bild 15. Die Datenanalyse: Es erfolgt eine Klassenbildung der Meßdaten.

Neuronale Netzwerke werden ebenfalls in den folgenden Bereichen eingesetzt : Prognose, Risikoanalyse, Expertensysteme, Bildverarbeitung, Entscheidungsunterstützung, Signalanalyse, Robotik, Simulationstechnik und in anderen Bereichen (nichttechnischen).

5. Neuro Fuzzy

Neuronale Netzwerke sind in der Lage komplexe System zu regeln. Ein Nachteil besteht darin, daß ihre Handlungsweise, bzw. der Algorithmus im Netz, nicht transparent, sondern verteilt in den Gewichten gespeichert ist. Das Wissen läßt sich nicht extrahieren. Diesen Mangel der Transparenz zeigen die Fuzzy-Systeme nicht. Im Gegenteil : Das Wissen läßt sich aus den WENN-DANN-Regeln gut identifizieren. Fuzzy-Systemen fehlt aber die Lernfähigkeit, was die Neuronale Netzwerke wiederum hervorragend beherrschen. Eine Möglichkeit, die jeweiligen Nachteile zu vermeiden bzw. die Vorteile der System auszunutzen, ist die Neuro-Fuzzy-Methode.

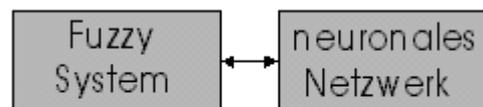


Bild 16. Neuro-Fuzzy-Methode

Das Ziel besteht darin, ein Fuzzy-System in ein äquivalentes Neuronales Netzwerk zu übersetzen und vice versa (Bild 16). Alle Fuzzy-Funktionen wie Fuzzifizieren, Regelbasis oder Defuzzifizieren werden in ein äquivalentes Neuronales Netz transformiert, um die Fuzzy-Parameter mit Meßdaten zu optimieren. Nach der Optimierung wird das Neuronale Netz in das Fuzzy System zurücktransformiert. Es wird zunächst ein Fuzzy System aufgestellt, das grob die gewünschten Anforderungen erfüllt und anschließend in ein äquivalentes Neuronales Netz transformiert. Nun beginnt die Feinoptimierung mit Meßdaten. Nach Abschluß der Optimierung ist die lernfähige Komponente übrig, und das Neuronale Netz wird in das Fuzzy-System rücktransformiert. Die Betriebsphase erfolgt mit dem optimierten Fuzzy-System.

6. Evolutionäre Algorithmen

Evolutionäre Algorithmen sind Such- und Optimierungsalgorithmen, die grundlegende Prinzipien der natürlichen Evolution nachbilden. Während in der Vergangenheit nur physikalische Erkenntnisse aus der Natur, wie etwa der Vogelflug, zum Vorbild für technische Entwicklungen genommen wurde, interessiert heute die Forscher nicht das Ergebnis der Evolution, sondern der Weg dahin. Man versucht, die biologische Evolution für technische Aufgaben nachzuahmen und damit zu besseren Ergebnissen zu kommen. Zu den evolutionären Algorithmen gehören die folgenden Verfahren :

- Genetische Algorithmen **GA**
- Evolutionsstrategien **ES**
- Genetische Programmierung **GP**

Die Genetische Algorithmen (GA)

Das Prinzip von Charles Darwin (1809-1882) „*The survival of the fittest*“ besagt, daß besser angepaßte Individuen größere Chance in der Natur haben, zu überleben und sich fortzupflanzen. Genetische Algorithmen verfolgen eben diese These als Grundlage für den Optimierungsalgorithmus. Bei den Genetischen Algorithmen werden üblicherweise die zu optimierenden Parameter in eine binärer Form codiert (Bild 17). Die codierten Parameter heißen **Gen**. Alle Parameter zusammen ergeben einen binären Vektor bzw. ein **Chromosom**. Mehrere solcher

P	I	D	Code
0.2	3.5	0.2	1001010101
1.1	4.1	0.5	0110100101
0.7	3.9	0.3	1010110101

Bild 17. Die Codierung des PID-Reglers in einem Chromosomen.

Chromosomen oder auch Individuen bilden eine **Population**. Zu Beginn der Optimierung erzeugt man zufällig eine Anfangspopulation. Ein bereits vorhandenes Wissen über die Parameter kann in die Anfangspopulation einfließen. Die Population wird mit einer Bewertungsfunktion bewertet. Das kann z.B. die Regelgüte aus der Sprungantwort sein. Aus der bewerteten Population werden nach einem Zufallsprinzip Individuen für die Fortpflanzung selektiert. Die besten Individuen bekommen auch die besten Chancen zum Fortzupflanzen und zum Überleben (Selektion). Die erzeugten Nachkommen werden dem Operator Crossover unterworfen (Bild 18). Bei dem Crossover Operator tauschen zwei Nachkommen an einer zufälligen Stelle im Chromosom Teile der Bits aus. Hierdurch wird - wie in der Natur - genetisches Material ausgetauscht und vermischt. Der Mutation Operator (Bild 19) sorgt für kleine Änderungen in den Chromosomen. Dazu werden sehr wenige Bits

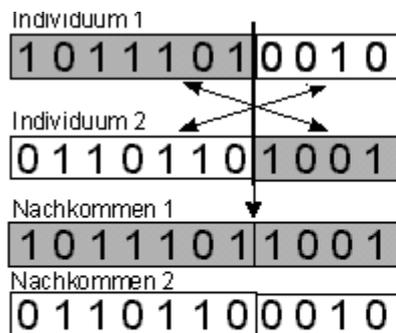


Bild 18. Das Crossover: An einer zufälligen Stelle im Chromosom werden Teile der Bits ausgetauscht.

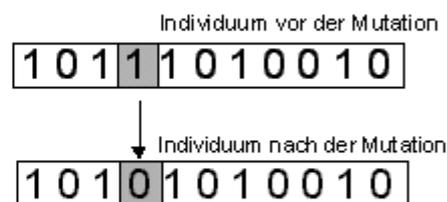


Bild 19. Mutation: Der Mutation Operator invertiert an zufälligen Stellen ein Bit.

an zufälligen Stellen invertiert. Die neue Generation von Chromosomen wird in die alte Generation eingegliedert. Die alte und die neue Generation werden wieder mit der Bewertungsfunktion bewertet. Aus den bewerteten Individuen werden wieder neue Nachkommen gebildet und so weiter (Bild 20). Es werden so lange weitere Nachkommen gebildet, bis das beste Individuum bestimmten Bedingungen genügt oder eine vorgegebene Anzahl von Optimierungen erreicht ist.

Im Vergleich zu den herkömmlichen Optimierungsalgorithmen liegt hier das Wissen über die Problemlösung fast ausschließlich in der Bewertungsfunktion. Das Funktionieren Genetischer Algorithmen hängt daher maßgeblich vom Finden problem-adäquater Bewertungsfunktionen

ab. In die Bewertungsfunktion können Beschränkungen eingebaut werden, wie beispielsweise der Faktor "minimaler Energieaufwand". Der Algorithmus braucht keine Informationen über den Gradienten der Zielfunktion. Die Suche erfolgt nicht nach festen Regeln, sondern wird

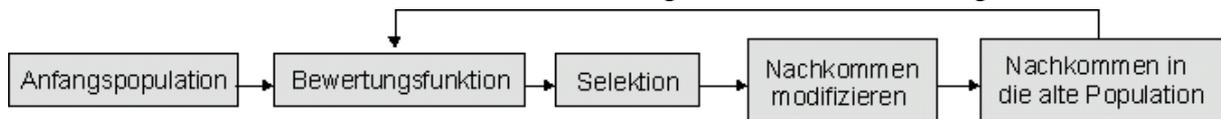


Bild 20. Der Genetische Algorithmus: Neue Nachkommen entstehen zufällig und ungerichtet. über Zufallsprozesse gesteuert und geht von einer Population und nicht von einer Lösung aus. Das Erzeugen neuer Nachkommen erfolgt zufällig und ungerichtet. Durch die Suche mit vielen Individuen im Suchraum steigt die Wahrscheinlichkeit, das globale Optimum zu finden. Der Implementierungsaufwand von Genetischen Algorithmen ist gering und ist unabhängig von der Größe des Suchraumes.

Genetische Algorithmen und Fuzzy Systeme

Zur Optimierung der Fuzzy-Regler kann man die Genetischen Algorithmen einsetzen (Bild 21). Man codiert zunächst alle Parameter des Fuzzy-Reglers wie Zugehörigkeitsfunktionen, Skalierungsfaktoren, und Regelbasis in eine binäre Form und wendet darauf den Genetischen Algorithmus an. Die Bewertungsfunktion läßt sich anhand der Sprungantwort des geschlossenen

Fuzzy-Regelkreises bestimmen. Die Optimierung erfolgt in einem Simulationsmodell, das ein gutes Modell der Regelstrecke voraussetzt. Hier eignen sich auch Neuronale Modelle, die die Nichtlinearitäten gut darstellen können. Wenn möglich, sollte der Fuzzy-Regler an der realen Regelstrecke optimiert werden.

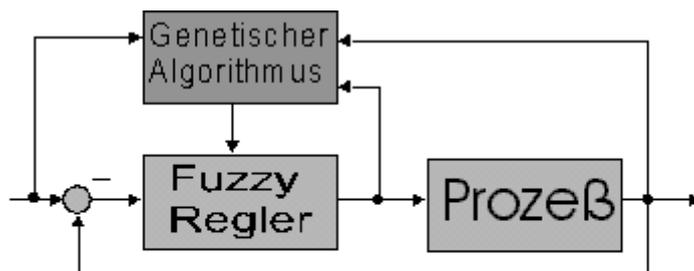


Bild 21. Genetische Algorithmen und Fuzzy Regler: Die Algorithmen sind ein Mittel, um die Regler zu optimieren.

Genetische Algorithmen und Neuronale Netze

Was den Aufbau eines neuronalen Netzes für eine konkrete Anwendung betrifft, gibt es bis heute noch keine theoretischen Vorgehensweisen, die die Wahl der Netzstruktur bzw. des Lernalgorithmus vorgeben. Während der Entwicklung bleibt dem Ingenieur nur die "Trial and error"-Methode, die einer Optimierung entspricht. In umfangreichen Versuchsreihen erfolgt die Suche nach einer optimalen Lösung.

Eine Alternative zu dieser kostenintensiven Methode stellt die Optimierung durch Genetische Algorithmen dar (Bild 22). Grundsätzlich werden entweder die Gewichte der Neuronen oder die Netzwerk-Struktur optimiert. Bei der Optimierung der Gewichte müssen lediglich die Gewichte in binärer Form dargestellt werden und können dem Genetischen

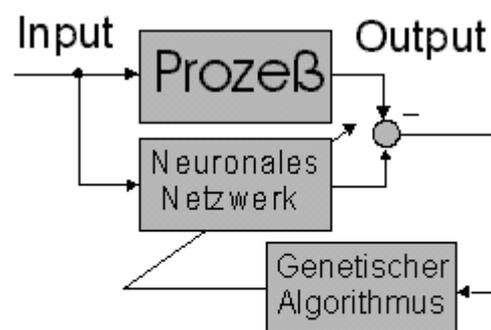


Bild 22. Genetische Algorithmen und Neuronale Netze: Eine Alternative zur herkömmlichen Lern-Methoden.

Algorithmus übergeben werden. Vorteil der Genetischen Optimierung ist, dass sie fähig ist, die lokalen Minima zu überwinden, was die Backpropagation-Lernmethode nicht kann. Die Genetischen Algorithmen eignen sich ebenfalls gut für die Optimierung der Netzwerkstruktur. Dazu muß man lediglich die Struktur und die Anzahl der Neuronen pro Schicht in einem geeigneten Chromosom codieren. Das große Problem bei der Optimierung der Neuronalen Netzwerke ist die relativ aufwendige Beurteilung der Bewertungsfunktion.

Die genetischen Algorithmen lassen sich ebenfalls zur Optimierung von Parametern in Simulationsmodellen nutzen. Simulationsmodelle lassen sich z.B. auch auf Energie- oder Kosten-Minimum optimieren. Solche Optimierungen führten bislang Experten durch, was sehr kostenintensiv ist. Genetische Algorithmen eignen sich grundsätzlich für alle Optimierungsaufgaben in der Automatisierungstechnik, wie beispielsweise bei der Suche nach der besten Maschinenauslastung und Belegung.

Die Evolutionsstrategien (ES)

Die Evolutionsstrategien arbeiten nach dem gleichen Prinzip wie die Genetischen Algorithmen, mit dem Unterschied, daß die Parameter nicht in binärer Form codiert werden. Es wird direkt mit den reellwertigen Parameter gearbeitet. Durch die reellwertige Darstellung ändert sich die Funktionsweise der Operatoren Mutation und Selektion. Bei der Mutation erfolgt eine Addition einer normal verteilte Zufallsgröße auf den reellwertigen Parameter. Der Schwerpunkt der Evolutionsstrategie liegt auf der Mutation. Sie erlaubt eine schrittweisenregulierung, die die Individuen ständig verbessert. Eine Rekombination durch Crossover findet üblicherweise nicht statt.

Die Genetische Programmierung (GP)

Die Genetischen Algorithmen und die Evolutionsstrategien optimieren eine Anzahl von Parametern. Ersetzt man die Parameter durch Variablen, logische und mathematische Operationen und andere Funktionen, so spricht man von Genetischer Programmierung. Die Population besteht hierbei nicht aus Parametern, sondern aus einer Anweisungsliste von Operationen und Variablen, die das Problem lösen könnten. Nehmen wir z.B. eine einfache Anweisung ($a+b*c$), die eine mögliche Lösung der Aufgabe darstellt. Die Repräsentation dieser Anweisung in GP erfolgt in Form einer Baumstruktur (Bild 23). Eine Population besteht aus einer endlichen Anzahl solcher Baumstrukturen, in denen die Anweisung gespeichert ist. Mit Hilfe von Crossover und Mutation werden neue mögliche Lösungen erzeugt. Konvergiert die Optimierung, so ergibt sich eine geschlossene mathematische Lösung für die Aufgabe.

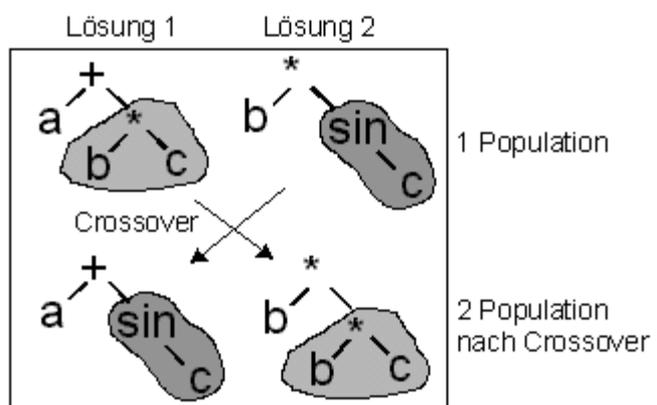


Bild 23. Genetische Programmierung und Crossover: Mittels Crossover und Mutation entstehen neue Lösungen.

7. Schlußbemerkung

Geht die Entwicklung von Soft Computing weiter so schnell voran, so werden in ein paar Jahren schon die Maschinen und Technischen Systemen einen „Maschinen-IQ“ erhalten, prophezeit Lotfi Zadeh. Diese Definition des "Maschine Intelligence Quotient" (MIQ) soll Aussagen über den Grad der Intelligenz eines Systems im Vergleich zu anderen veranschaulichen. Anwendungen von Soft Computing stärkt die Wettbewerbsfähigkeit, denn sie liefert intelligente Algorithmen, die eine flexible und rasche Einsatzbarkeit liefern und neue Einsatzmöglichkeiten erschließen. Dem Einsatz von Soft Computing steht - dank der ausgereiften Software und Hardware auf dem Markt - nichts mehr im Wege.

Literatur :

- [1] C. v. Altrock "Fuzzy Logic" Band 1 Technologie, Oldenbourg
- [2] H.J.Zimmerman C.v.Altrock "Fuzzy Logic" Band 2 Anwendungen, Oldenbourg
- [3] J. Kahlert H. Frank "Fuzzy-Logic und Fuzzy-Control", Viewg
- [4] H.J. Zimmermann "Fuzzy Technologien" VDI Verlag
- [5] H.J. Zimmermann "Neuro+Fuzzy Technologie - Anwendung" VDI Verlag
- [6] C. Pöppr "Mathematische Unterhaltung" Spektrum der Wissenschaft
- [7] E. Schöneburg F. Heinzmann S. Feddersen "Genetische Algorithmen und Evolutionsstrategien" Addison-Wesley
- [8] D.Dasgupta Z.Michalewicz "Evolutionary Algorithms in Engineering Applications" Springer Verlag
- [9] H.C.Schulz, S.Hilgenfeldt "Experimente zum Chaos", Spektrum der Wissenschaft, Januar 1994
- [10] C.J.Harris,C.G.Moore,M.Brown "Intelligent control, Aspects of Fuzzy Logic and Neural Nets", World Scientific, Vol. 6
- [11] A.Kandel, G.Langholz "Fuzzy control systems", CRC Press
- [12] T.Furuhashi y.Uchikawa "Fuzzy Logic, Neural Networks, and Evolutionary Computation" Springer Verlag, Selected Papers, November 1995