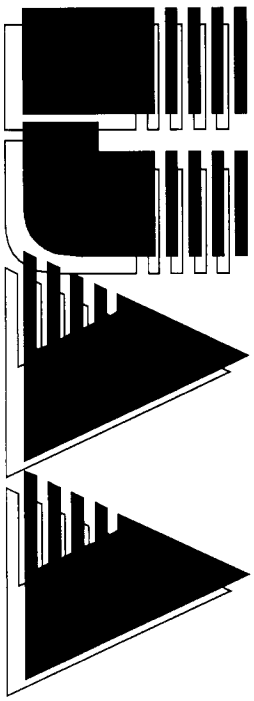
ISSN 1562 - 2703

INTERNATIONAL JOURNAL

HEFT 2 Jg. 10, (2002)

AUTOMATION AUSTRIA



INHALT

	Seite
TRAXL, R. **Optimierung von Investitions- und Make-or-Buy-Entscheidungen mit Simulated Annealing**	**89**
WEINMANN, A. **Continuous-time Linear Quadratic Regulator Subject to an Additonal Constraint on the State Controller Matrix**	**97**
PERME, T. u. NOE, D. **Erweiterte Petrinetze zur intelligenten Planung von Produktionssystemen**	**111**
FAVRE-BULLE, B.; BUSCH, Ch.; FEURSTEIN, R. **Bionic Solutions for Planar Tentacle Grippers**	**125**
HÄBERLE, St. **Sales Process for Product and Service Management for Machine Manufacturing Small and Medium Sized Enterprises**	**140**
SCHLAGLICHTER	**145**
BUCHBESPRECHUNG	**148**
DISSERTATIONSKURZFASSUNGEN	**155**
VERANSTALTUNGEN	**157**

Optimierung von Investitions- und Make-or-Buy-Entscheidungen mit Simulated Annealing

R. Traxl, Ulm

Eingelangt im Juli 2002

Kurzfassung

Die Sicherstellung einer kosteneffizienten Produktion bei turbulenten Marktbedingungen und steigender Produktkomplexität erfordert geeignete Methoden zur Analyse und Optimierung von Produktionsanlagen. Dieser Beitrag stellt eine Simulated Annealing basierte Methode zur Optimierung der Investitions- und Zukaufpolitik einer Fertigungsanlage vor. Es werden zwei Alternativen der Nachbarschaftssuche vorgestellt und die erzielte Optimierungsperformance anhand der Simulationszeit und des erreichten Fitnesswerts verglichen.

1 Einleitung

Produzierende Unternehmen stehen heute vor der Aufgabe, Produkte, Kosten, Mengen und Termine bei steigenden Qualitätsanforderungen und hohem Wettbewerbsdruck zu optimieren. Gerade in der Automobilindustrie gab es in den letzten Jahren eine enorme Erhöhung der Komplexität durch steigende Produktvielfalt, geringere Stückzahlen pro Modellreihe und kürzere Produktlebenszyklen. Bei DaimlerChrysler hat sich z.B. im Geschäftsfeld PKW die Anzahl der Baureihen im Zeitraum von 1990 bis 2002 mehr als verdoppelt. Die Automobilhersteller sind also gezwungen, eine große Kosteneffizienz auch bei hoher Produktvarianz zu gewährleisten. Dazu ist ein geeigneter, intensiver Innovationsprozess erforderlich, in welchen nicht nur die Produkte, sondern auch die Fertigungs- und Vertriebsprozesse einbezogen werden.

Eine entscheidende Unternehmensaufgabe zur Sicherstellung der kosteneffizienten Produktion in diesem turbulenten Produktionsumfeld ist eine ständige Anpassung der Investitions- und Make-or-Buy Strategie von Unternehmen an die erwartete Marktsituation. Durch den raschen Wandel des Produktionsumfeldes entsteht ein großer Bedarf, die Auswirkungen neuer Technologien und unterschiedlicher Strategien auf alternative Marktentwicklungen rasch zu quantifizieren und somit zu rechtfertigen. Zu diesem Zweck bedarf es intelligenter, computerunterstützter Planungstools, mit denen eine zeiteffiziente und flexible Planung trotz komplexer und rasch veränderlicher Informationen möglich wird.

In diesem Beitrag wird die Optimierung von Investitions- und Make-or-Buy Decisions eines Motorenwerks mit einem vom Autor entwickelten Planungstool zur Quantifizierung der Auswirkung alternativer Marktszenarien, Fertigungsstrukturen und Managementkonzepte [Tra01] beschrieben. Zur Optimierung werden zwei Simulated Annealing basierte Verfahren eingesetzt.

2 Optimierungsansatz

Den Ausgangspunkt für die Optimierung bildet ein vorgegebenes Fertigungsprogramm, das die Art und Menge der zu produzierenden Produktvarianten über den gesamten Planungshorizont festlegt. Zusätzlich müssen mögliche Alternativen für Investitionen und Teilezukäufe definiert werden. Ziel der Optimierung ist es, die Investitionsalternativen und Teilezukäufe so im Planungszeitraum zu kombinieren, dass sich ein maximaler Kapitalwert über den betrachteten Planungshorizont ergibt. Dabei steht dem Unternehmen in jedem Jahr eine definierte Investitionssumme zur Verfügung.

Als Eingangsdaten werden dem heuristischen Algorithmus der gewählte Planungshorizont, eine Liste der Investitionsalternativen, die verfügbare Investitionssumme für jedes Jahr des Planungshorizontes und Daten über mögliche Zukaufsteile zur Verfügung gestellt. Die Komponenten, die von Fremdfirmen zugekauft werden können, werden mit Angabe der Stückpreise in einer Tabelle festgehalten, in der auch mögliche Mengenrabatte berücksichtigt werden. Die Liste der Investitionsalternativen dient zur Festlegung der erforderlichen Daten über mögliche Investitionsprojekte.

Aus den genannten Inputdaten generiert der Optimierungsalgorithmus Vorschläge für die Investitions- und Make-or-Buy Politik und liefert diese an ein Planungstool zur Analyse und Optimierung von Fertigungsanlagen [Tra01]. Kernstück des Planungstools ist ein analytisches Modell einer integrierten Fertigungsanlage, bestehend aus direkten und indirekten Produktionsbereichen, mithilfe dessen eine Bewertung der generierten Lösungsvorschläge unter Berücksichtigung des vorgegebenen Fertigungsprogramms erfolgt. Aus den resultierenden Produktionskosten der einzelnen Jahre, den Investitions- und Zukaufkosten und der Restwerte der Investitionen am Ende des Planungszeitraums wird für jeden Lösungsvorschlag ein Fitnesswert bestimmt, der sich aus dem Kapitalwert des jeweiligen Investitions- und Zukaufplans ableitet. Die ermittelten Fitnesswerte werden an den heuristischen Algorithmus zurückgeliefert und dienen als Informationsquelle für die Generierung weiterer Lösungsvorschläge. Durch diese Rückkopplung wird der Informationsfluss geschlossen.

3 Optimierungsansatz mit Simulated Annealing

Problemstellungen in der Industrie lassen sich häufig auf kombinatorische Optimierungsprobleme zurückführen, die dadurch gekennzeichnet sind, dass sich die Lösungsmenge aus einer endlichen Anzahl diskreter Elemente zusammensetzt. Aufgrund der Komplexität praxisrelevanter Problemstellungen, die sich aus einer unüberschaubaren Anzahl von Einflussgrößen, gegenseitigen Abhängigkeiten der Systembereiche, unstetigen Funktionen und mehreren lokalen Optima ergibt, führen deterministische Optimierungsansätze wie lineare oder quadratische Programmierung oder Branch-and-Bound Algorithmen nur selten in angemessener Zeit zu akzeptablen Lösungen.

Zur Lösung praxisrelevanter kombinatorischer Problemstellungen wurden seit Beginn der achtziger Jahre verschiedene heuristische und stochastische Methoden entwickelt, mit denen die Lösung komplexer Optimierungsprobleme mit großen, multimodalen Suchräumen möglich wird. Ein multimodaler Suchraum ist durch eine Zielfunktion mit mehreren lokalen Optima charakterisiert. Die Aufgabe der Algorithmen ist es, in solchen Suchräumen das globale Optimum mit möglichst wenig Zeit- und Rechenaufwand zu

finden. Die wichtigsten Vertreter von Suchalgorithmen mit denen diese Art von Problemen beherrscht werden kann, sind Tabu Search [Glov97], Simulated Annealing [Kirk83] und Genetische Algorithmen [Holl75]. Die genannten Verfahren haben sich als sehr effizient bei der Lösung von NP-schweren, kombinatorischen Optimierungsproblemen herausgestellt [AaLe97, Krish99].

In diesem Beitrag wird die Optimierung von Investitions- und Make-or-Buy Entscheidungen mit zwei Simulated Annealing basierten Verfahren beschrieben. Für eine Optimierung mit Genetischen Algorithmen wird auf [Tra02] verwiesen. Simulated Annealing ist ein Optimierungsverfahren zur Lösung kombinatorischer Optimierungsprobleme, das auf Erkenntnissen über das Wachstum von Kristallen in Schmelzen beruht. Kühlt man eine Metallschmelze langsam ab, ordnen sich die Atome so an, dass sie einen Zustand möglichst geringer Energie annehmen. Ein niederer Energiezustand ist dabei gleichbedeutend mit einem stabilen Endzustand des Metalls. Jedes Teilchen in der Schmelze versucht, die für sich energetisch günstigste Position zu erreichen. Durch die thermische Energie und die daraus resultierende Teilchenbewegung werden die Teilchen aber immer wieder in energetisch ungünstigere Zustände gebracht. Die Wahrscheinlichkeit, dass ein Teilchen mit der Temperatur T in eine um ΔE schlechtere (energiereichere) Position gebracht wird ist durch

$$e^{-\left(\frac{\Delta E}{k_B T}\right)} \qquad \text{Gl. 1}$$

gegeben. Die Konstante k_B steht dabei für die Boltzmann Konstante. Bei einer Verbesserung der energetischen Position eines Teilchens wird die Energiedifferenz ΔE zwischen den beiden Zuständen frei. Durch langsames Abkühlen der Schmelze ordnen sich die Teilchen mit der Zeit so an, dass sie einen Zustand möglichst geringer Energie annehmen. Bei einer zu schnellen Abkühlung bleibt jedoch nicht genügend Zeit, das globale Energieminimum zu finden und das System bleibt in einem lokalen Optimum stecken.

Diese Erkenntnisse der statistischen Physik werden bei SA in einem Optimierungsalgorithmus umgesetzt (Abb. 1). Die Menge aller möglichen Zustände der Teilchen der Schmelze bildet die Lösungsmenge L. Die zu minimierende Zielfunktion $f(x)$ ist die Summe der energetischen Positionen aller Teilchen. Ausgehend von einer Startlösung x wird in jedem Iterationsschritt eine zufällige Lösung x' aus deren Nachbarschaft $N(x)$ generiert und die Energie bestimmt, die für den Zustandsübergang von x nach x' erforderlich ist. Die Entscheidung, ob der neue Zustand x' tatsächlich angenommen wird, wird durch das sogenannte Metropolis Kriterium festgelegt, das folgendermaßen definiert ist: Ergibt sich mit x' eine Verschlechterung der Zielfunktion, ist die Wahrscheinlichkeit für den Zustandsübergang durch Gl.1 gegeben. Bei einer Verbesserung der energetischen Situation des Gesamtsystems wird hingegen ausnahmslos auf den neuen Zustand übergegangen. Durch das Metropolis Kriterium wird also eine temporäre Verschlechterung der Lösungen zugelassen, wodurch ein Verlassen lokaler Optima möglich wird.

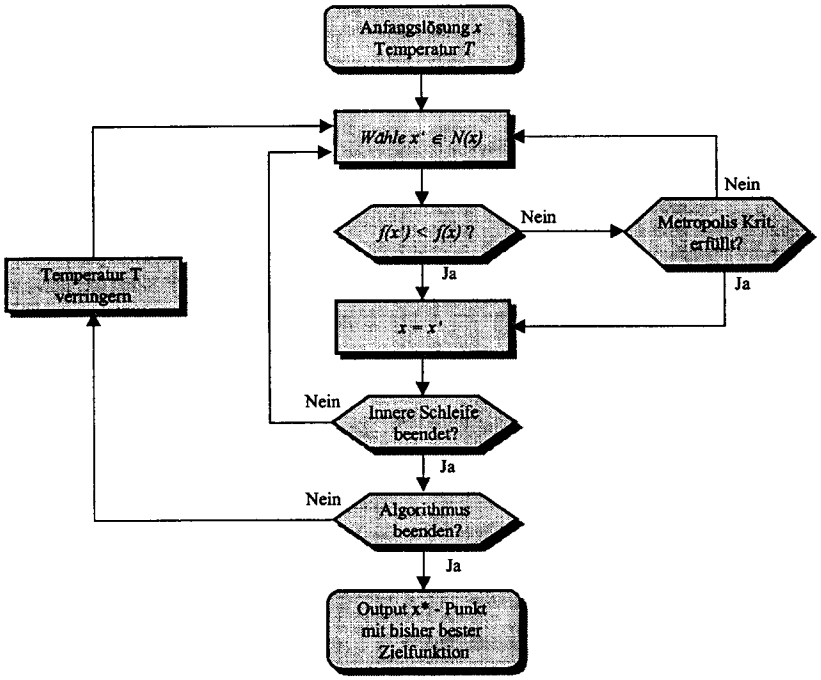


Abbildung 1: Simulated Annealing, Basisalgorithmus

Der Algorithmus wird in der Regel mit einer relativ hohen Anfangstemperatur T gestartet. Dadurch ist die Wahrscheinlichkeit für Zustandsübergänge, die eine Verschlechterung der Fitnessfunktion bewirken, gemäß Gl.1 anfangs sehr hoch. Dies ermöglicht eine großflächige Untersuchung des Suchraumes. Im Laufe der Optimierung wird die Temperatur schrittweise verringert.

Zur Optimierung der betrachteten Problemstellung wird in jedem Durchlauf der inneren Schleife ein neuer Lösungsvorschlag für das Investitions- und Zukaufproblem generiert und mit einem Fitnesswert bewertet. Dieser leitet sich aus dem Kapitalwert der vorgeschlagenen Investitions- und Make-or-Buy Strategie ab und wird anhand des Planungstools zur Analyse und Optimierung von Fertigungsanlagen ermittelt. Nach der Ermittlung des Fitnesswertes wird entschieden, ob der generierte Lösungsvorschlag den aktuellen Plan ersetzen soll oder nicht. Dazu werden die Fitnesswerte der beiden Investitions- und Zukaufspläne verglichen. Die Entscheidung, ob der neu-generierte oder der aktuelle Lösungsvorschlag im Optimierungsprozess weiterverwendet wird, wird durch das Metropolis Kriterium getroffen. Im anschließenden Schleifendurchlauf dient dann die neu ausgewählte Lösung als Vergleichsbasis für das Metropolis Kriterium.

Die innere Schleife wird beendet, wenn eine vorgegebene Schleifenanzahl überschritten wird oder die Anzahl der durch das Metropolis Kriterium akzeptierten Lösungsvorschläge in der Schleife einen definierten Wert übersteigt. Nach dem Ende der inneren Schleife kehrt der Algorithmus in die äußere Schleife zurück, in der eine Temperaturreduzierung um einen einstellbaren Faktor erfolgt. Mit dem neuen Temperaturwert wird anschließend die nächste innere Schleife gestartet. Durch die stetige Temperaturreduzierung werden mit zunehmendem Fortlauf der Optimierung in der Regel immer weniger Lösungen gefunden, die das Metropolis Kriterium erfüllen. Der Optimierungsprozess wird abgebrochen, wenn in einem Durchlauf der äußeren Schleife keine einzige neue Lösung akzeptiert wird. Tritt dieser Fall im Laufe der Optimierung nicht ein, wird der Algorithmus erst nach einer vorgegebenen äußeren Schleifenanzahl beendet.

4 Nachbarschaftssuche

Unter dem Begriff Nachbarschaftssuche versteht man die Aufgabe eines Optimierungsalgorithmus, ausgehend von einer bestimmten Lösung des Optimierungsproblems durch geringfügige Modifikationen einen neuen Lösungsvorschlag zu erzeugen.

Bei der entwickelten Methodik zur Nachbarschaftssuche wird als erster Schritt zur Generierung einer neuen Lösung eine Zufallszahl zwischen Null und Eins ermittelt, die bestimmt, ob sich die neue Lösung von der alten durch die Make-or-Buy Strategie, die Investitionspolitik oder beides unterscheiden soll. Die Wahrscheinlichkeiten für diese drei Fälle können durch den Benutzer festgelegt werden. Wird die Make-or-Buy Strategie verändert, bestimmen weitere Zufallszahlen, bei welcher Motorkomponente und in welchem Jahr des Planungszeitraums eine Änderung der Zukaufsrate erfolgen soll. Schließlich muss noch entschieden werden, ob die Zukaufsrate um eine Schrittweite erhöht oder verringert wird. Entspricht die aktuelle Zukaufsrate zkr_{akt} des betrachteten Teils der maximalen Zukaufsrate zkr_{max}, ist nur eine Verringerung möglich. Bei einer aktuellen Zukaufsrate von Null kann andererseits nur eine Erhöhung erfolgen.

Bei der Modifikation des Investitionsplans wurden zwei alternative Verfahren der Nachbarschaftssuche implementiert. Alternative 1 ist dadurch gekennzeichnet, dass Lösungsvorschläge die zu einer Überschreitung der Investitionssumme führen prinzipiell zugelassen werden. Erst bei der Berechnung eines Fitnesswertes werden solche unzulässige Lösungen mit einem Strafterm versehen. Dadurch wird deren Fitnesswert verschlechtert, was dazu führt, dass unzulässige Investitionspläne nach und nach aus dem Optimierungsprozess verschwinden. Ein großer Nachteil dieser Methode ist es, dass auch die unzulässigen Lösungen durch das Planungstool bewertet werden müssen, was den Rechenaufwand in die Höhe treibt und somit die Optimierungsperformance verschlechtert. Aus diesem Grund werden bei der zweiten Variante zur Nachbarschaftssuche unzulässige Lösungen von vornherein ausgeschlossen. Dies erfordert einen größeren Aufwand bei der Erzeugung neuer Lösungen, dafür entfällt die Bewertung unzulässiger Lösungen.

Abbildung 2 zeigt das Flussdiagramm für die Erzeugung von Investitionsplänen nach Variante 2. Als erster Schritt wird eine Zufallszahl *zz4* zwischen Null und Eins bestimmt, die festlegt, ob die Anzahl der Investitionen gleich bleibt, erhöht oder verringert wird. In den Defaulteinstellungen gelten folgende Grenzwerte für die einzelnen Fälle:

1.) $0 < zz4 \leq 1/3$ → Erhöhung der Anzahl an Investitionen
2.) $1/3 < zz4 \leq 2/3$ → Verringerung der Anzahl an Investitionen
3.) $2/3 < zz4 \leq 1$ → Anzahl an Investitionen bleibt gleich

Ist die Anzahl der Investitionen im aktuellen Investitionsplan gleich Null, kann nur Fall 1, also eine Erhöhung der Investitionszahl, eintreten. Falls alle möglichen Investitionen schon vergeben sind ($inv_{akt} = inv_{max}$), sind die Fälle 2 und 3 möglich. Tritt der Fall 1 in Kraft, wird der aktuelle Investitionsplan an einem zufällig bestimmten Jahr des Planungszeitraumes um eine neue Investition ergänzt. Eine Bedingung für die zusätzliche Investition ist, dass sie im aktuellen Investitionsplan nicht bereits vorkommen darf. Anschließend wird überprüft, ob die Erweiterung des Investitionsplans eine Überschreitung der Investitionssumme zur Folge hat. Tritt eine Überschreitung auf, wird

ein neuer Lösungsvorschlag ermittelt und erneut die Investitionssumme überprüft. Falls nach einer festgelegten Anzahl an Wiederholungen immer noch keine gültige Lösung gefunden wurde, wird der Vorgang abgebrochen und der alte Investitionsplan weiterverwendet.

Soll die Anzahl an Investitionen verringert werden (Fall 2), wird ein zufälliges Feld des Investitionsplanes selektiert, das mit einer Investition belegt ist. Anschließend wird dieses Feld gelöscht.

Schließlich ist noch Fall 3 möglich, bei dem die Anzahl an Investitionen nicht verändert wird. Der neue Investitionsplan wird dann durch eine Kombination von Fall 1 und 2 generiert. Zuerst wird eine Investition des aktuellen Investitionsplanes gelöscht und anschließend gemäß Fall 1 eine neue Investition hinzugefügt.

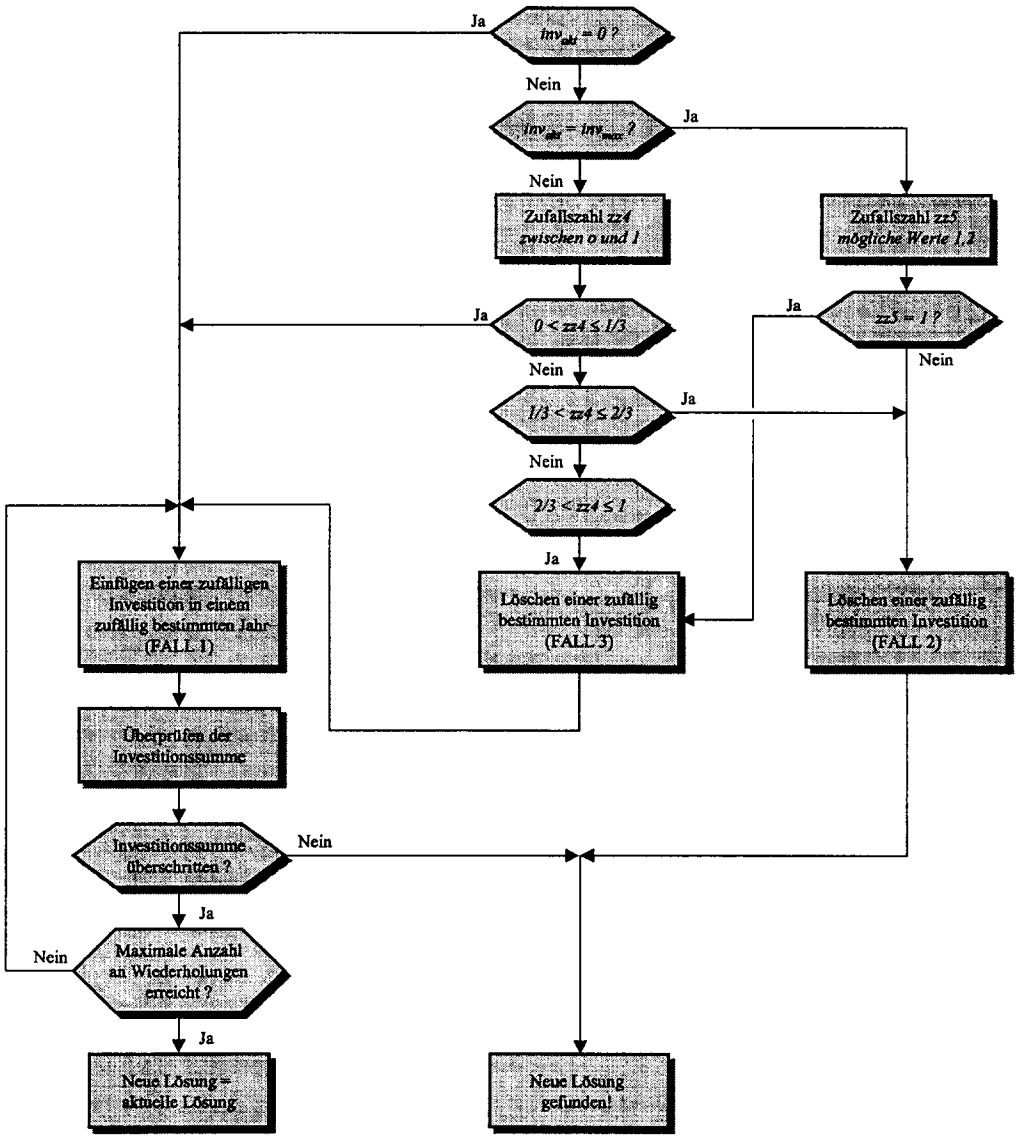


Abbildung 2: Nachbarschaftsuche mit SA: Veränderung der Investitionspolitik, inv_{max}.....maximale Anzahl an Investitionen, inv_{akt}.....aktuelle Anzahl an Investitionen

5 Ergebnisse

Zur Bewertung der Optimierungsperformance der beiden betrachteten Verfahren wurde für das betrachtete Motorenwerk ein konkretes Optimierungsbeispiel mit einem Planungshorizont von 5 Jahren und einem Kalkulationszinssatz von 10% definiert und mit den Simulated Annealing Algorithmen optimiert. Die Berechnungen wurden mit einem Intel Pentium III Prozessor mit 650 MHz Taktfrequenz durchgeführt.

Für den Vergleich der implementierten Optimierungsalgorithmen wurden für jeden Algorithmus mehrere Simulationsläufe durchgeführt und die erzielten Fitnesswerte und benötigten Zeiten festgehalten. Abbildung 3 zeigt die Ergebnisse der experimentellen Vergleichsstudie in einem zweidimensionalen Diagramm. Die Abkürzung SA1 steht dabei für das implementierte Simulated Annealing Verfahren mit Tolerierung unzulässiger Lösungen, SA2 bezeichnet jenen Algorithmus, der unzulässige Lösungen a priori vermeidet. Die Abszisse gibt die Simulationsdauer in Minuten an, auf der Ordinate können die erzielten Fitnesswerte abgelesen werden. In der Abbildung ist zusätzlich zu den Datenpunkten für jeden implementierten Algorithmus eine Trendlinie angeführt. Damit kann ein einfacher Vergleich der mit den einzelnen Algorithmen erzielten Performance erfolgen. Liegt die Trendlinie eines Algorithmus höher als jene eines anderen, erreicht der Algorithmus bei gleicher Optimierungsdauer schlechtere Fitnesswerte, bzw. benötigt eine längere Zeitdauer für die Erreichung vergleichbarer Fitnesswerte. In Abbildung 3 ist für jede Trendlinie auch der Wert des Bestimmtheitskoeffizienten R^2 angeführt, der den Grad der Übereinstimmung zwischen den Werten der Trendlinie und den Datenpunkten des Streuungsdiagramms angibt. Die Verlässlichkeit der Trendlinie ist umso größer, je näher sich der Betrag des Bestimmtheitsmaßes dem Wert 1 nähert.

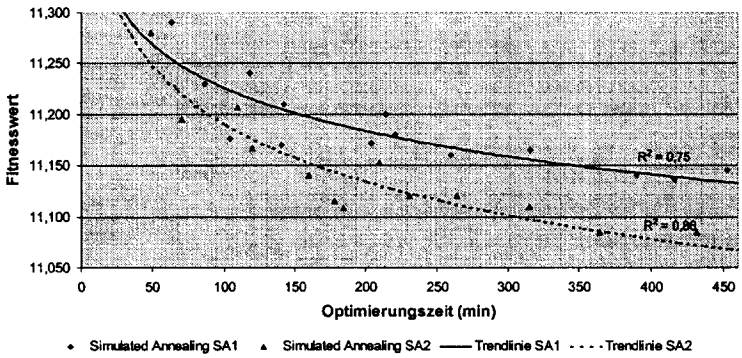


Abbildung 3: experimenteller Vergleich der implementierten Optimierungsalgorithmen anhand der Simulationsdauer und des Fitnesswertes

Anhand der Trendlinien lässt sich erkennen, dass für beide Algorithmen mit wachsender Optimierungsdauer eine Verbesserung der Fitnesswerte erzielt werden kann. Ab einer gewissen Optimierungszeit stellt sich jedoch ein Sättigungseffekt ein. Das Verfahren SA1 weist gegenüber SA2 eine höher liegende Trendlinie, d.h. eine schlechtere Optimierungsperformance auf. Eine Ursache dafür ist, das bei diesem Verfahren unzulässige Lösungsvorschläge prinzipiell zugelassen und erst bei der Berechnung eines Fitnesswertes mit einem Strafterm versehen werden. Dadurch müssen auch Lösungen, die für das Optimierungsergebnis nicht in Frage kommen, durch das Kostenmodell bewertet werden. Daraus resultiert ein gegenüber SA2 höherer Rechenaufwand und somit eine schlechtere Optimierungsperformance.

6 Zusammenfassung

Dieser Beitrag behandelte die Optimierung von Investitions- und Make-or-Buy Entscheidungen mit Simulated Annealing basierten Methoden. Die Nachbarschaftssuche wurde mit zwei alternativen Algorithmen durchgeführt. Ein Vergleich der Optimierungsperformance zeigt, dass jene Alternative, die eine Generierung unzulässiger Lösungen a priori vermeidet, zu besseren Fitnesswerten bei gleicher Optimierungszeit führt.

7 Literaturverzeichnis

[AaLe97] E. Aarts, J.-K. Lenstra, *Local Search in Combinatorial Optimization*, John Wiley & Sons, 1997.

[Glov97] F. Glover, M. Laguna, *Tabu Search*, Kluwer Academic Publishers, 1997.

[Holl75] J.-H. Holland, *Adaption in Natural and Artificial Systems*, University of Michigan Press, 1975.

[Kirk83] S. Kirkpatrick, C.-D. Gelatt jr., M.-P. Vecchi, *Optimization by Simulated Annealing*, Science, no. 220. pp. 671-680, 1983.

[Krish99] B. Krishnamachari, *Global Optimization in the design of mobile communication systems* , MS Thesis, 1999.

[Tra01] Traxl, R., Fürst, K., Zeichen G., 2001, Management support tool for complex production plant scenarios, Proceedings of the IFAC Symposium on Modeling and Control of Economic Systems - SME 2001, September 6-8, 2001, Klagenfurt, Austria.

[Tra02] Traxl, R., Pretschuh J., Fürst, K., Zeichen G., 2002, Optimization of the investment and purchase plan for complex production plants, Industrial Engineering Research Conference IERC02, May 19 - 21, 2002 Orlando, Florida.

Der Autor dieses Beitrags:

Dipl.-Ing. Dr. techn. René Traxl, geb. am 27. Okt. 1974, studierte Elektrotechnik an der Technischen Universität Wien und promovierte am Institut für Automatisierungs- und Regelungstechnik. Seit März 2002 ist er im DaimlerChrysler Forschungszentrum Ulm im Bereich Produktionstechnik tätig.

Continuous-time Linear Quadratic Regulator Subject to an Additional Constraint on the State Controller Matrix

Alexander Weinmann, ÖVE, Senior Member IEEE
Vienna University of Technology, Institute of Automatic Control
Gusshausstrasse 27-29/376
A-1040 Vienna / Austria
Phone: +43 1 58801 37600
Fax: +43 1 58801 37699
email: weinmann@acin.tuwien.ac.at

Manuscript received May 27, 2002

Abstract

Feedback control is modified by including high-frequency noise influence on the actuating variable, which is represented by an upper limit given by the controller matrix. The Frobenius norm of the feedback controller and a Lagrange multiplier are used to solve the problem as an equality condition. In addition, the neighbourhood of the optimum is analyzed by derivations based on incremental interrelations.

Keywords:
State controller restriction, sensitivity of the optimal adjustment

1 Introduction

The state feedback and output H_2 controllers are well established in control theory (*Levine, W.S., et al. 1971; Kuo, B.C., 1980; Chung, L.L., et al., 1989; Lin, C.C.,et al. 1996*). They are also the basis for H_∞ control (*Doyle, J.C., 1989*). In addition, emphasis was put on several supplements, e.g. disturbance rejection and robustness (*Doyle, J.C., 1989*).

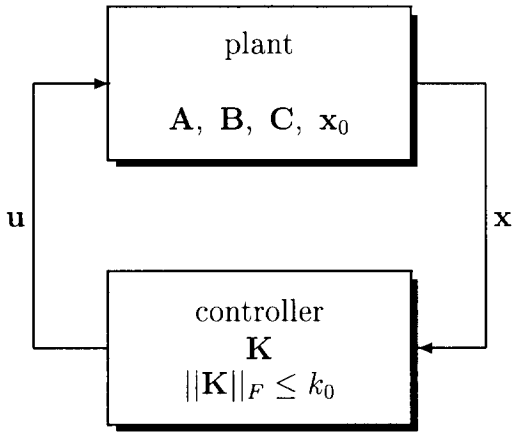


Figure 1: Controller and generalized plant

This paper addresses the problem of constraining the influence of noise on the actuating variable. In the case of high-frequency noise, this corresponds to limitation of the output controller matrix in its spectral norm (or in sufficient sense in its Frobenius norm).

In addition, fixed-point microprocessor applications should avoid large numbers in order preserve computational accuracy.

Consider the multivariable system in Fig. 1

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad \in \mathcal{R}^n \tag{1}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \quad \in \mathcal{R}^r \tag{2}$$

$$\mathbf{u} = \mathbf{K}\,\mathbf{x} \quad \in \mathcal{R}^m \tag{3}$$

$$\mathbf{A} \in \mathcal{R}^{n\times n},\ \mathbf{B} \in \mathcal{R}^{n\times m},\ \mathbf{C} \in \mathcal{R}^{r\times n},\ \mathbf{K} \in \mathcal{R}^{m\times n}\ . \tag{4}$$

Minimizing the index of performance

$$I = \int_0^\infty (\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u})dt\ , \tag{5}$$

and preassuming controllability, the well-known result is

$$\mathbf{K} = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} \tag{6}$$

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = \mathbf{0}, \tag{7}$$

or $\mathbf{K} = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}_{cl}$ in combination with Eq.(16).

Now, consider separately a linear quadratic index of performance only associated with $\mathbf{x}(t)$

$$I_x = 0,5\int_0^\infty \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t)dt = 0,5\mathbf{x}_0^T\mathbf{P}_G\mathbf{x}_0 \tag{8}$$

where

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 \ . \tag{9}$$

Then the matrix $\mathbf{P}_G$ results from the Lyapunov equation

$$\mathbf{A}^T\mathbf{P}_G + \mathbf{P}_G\mathbf{A} + \mathbf{Q} = \mathbf{0} \ , \tag{10}$$

assuming $\mathbf{A}$ stable and $\mathbf{Q} = \mathbf{Q}^T \geq 0$. The matrix $\mathbf{P}_G$ is symmetric.

In contrast to this, for the dyadic product $\mathbf{x}(t)\mathbf{x}^T(t)$ one has

$$\mathbf{J}_G = \int_0^\infty \mathbf{x}(t)\mathbf{x}^T(t)dt \tag{11}$$

where

$$\mathbf{A}\mathbf{J}_G + \mathbf{J}_G\mathbf{A}^T + \mathbf{X}_0 = \mathbf{0} \quad \text{and} \quad \mathbf{X}_0 \triangleq \mathbf{x}_0\mathbf{x}_0^T \ . \tag{12}$$

For the entire performance and for the controller $\mathbf{u}(t) = \mathbf{K}\mathbf{x}(t)$, the result is

$$I = 0,5\int_0^\infty [\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u}]\ dt = 0,5\int_0^\infty \mathbf{x}^T(\mathbf{Q}+\mathbf{K}^T\mathbf{R}\mathbf{K})\mathbf{x}\ dt \triangleq 0,5\int_0^\infty \mathbf{x}^T(t)\mathbf{Q}_{cl}\mathbf{x}(t)\ dt \ . \tag{13}$$

Using $\mathbf{u} = \mathbf{K}\mathbf{x}$, $\mathbf{A}_{cl} \triangleq \mathbf{A} + \mathbf{B}\mathbf{K}$ and positive definiteness $\mathbf{Q} + \mathbf{K}^T\mathbf{R}\mathbf{K} > 0$

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \qquad \mathbf{x}(0) = \mathbf{x}_0 \qquad \mathbf{x}(t) = e^{\mathbf{A}_{cl}t}\mathbf{x}_0 \tag{14}$$

$$I = 0,5\ \mathbf{x}_0^T\int_0^\infty e^{\mathbf{A}_{cl}^T t}\mathbf{Q}_{cl}e^{\mathbf{A}_{cl}t}dt\ \mathbf{x}_0 \triangleq \mathbf{x}_0^T\mathbf{P}_{cl}\mathbf{x}_0 \tag{15}$$

$$\mathbf{A}_{cl}^T\mathbf{P}_{cl} + \mathbf{P}_{cl}\mathbf{A}_{cl} = -\mathbf{Q}_{cl} \quad \text{where} \quad \mathbf{P}_{cl} = \mathbf{P}_{cl}^T > 0 \ . \tag{16}$$

2 Derivation and Main Result for Controller Norm Inclusion

If an expression $\mu\|\mathbf{K}\|_F^2$ is included in the setting of Eq.(13), one finds (see the entire derivation e.g. in *Weinmann, A., 1995, p. 282*)

$$\frac{\partial I}{\partial \mathbf{K}} = (\mathbf{R}\mathbf{K} + \mathbf{B}^T\mathbf{P})\mathbf{J} + 2\mu\mathbf{K} = \mathbf{0} \ . \tag{17}$$

The matrices $\mathbf{P}$ and $\mathbf{J}$ are achieved from

$$\mathbf{P}: \quad \mathbf{A}_{cl}^T\mathbf{P} + \mathbf{P}\mathbf{A}_{cl} + \mathbf{Q} + \mathbf{K}^T\mathbf{R}\mathbf{K} = \mathbf{0}, \qquad \mathbf{P} = \mathbf{P}^T \tag{18}$$

$$\mathbf{J}: \quad \mathbf{A}_{cl}\mathbf{J} + \mathbf{J}\mathbf{A}_{cl}^T + \mathbf{X}_0 = \mathbf{0} \qquad \mathbf{J} = \mathbf{J}^T \tag{19}$$

or with MATLAB

$$\mathbf{P} = \text{lyap}(\mathbf{A}_{cl}^T, \mathbf{Q} + \mathbf{K}^T\mathbf{R}\mathbf{K}) \tag{20}$$

$$\mathbf{J} = \text{lyap}(\mathbf{A}_{cl}, \mathbf{X}_0) \ . \tag{21}$$

Combining the Eqs.(17), (18) and (19) yields the result in $\mathbf{K}$, $\mathbf{J}$ and $\mathbf{P}$. For the unknown μ the fourth equation results from the equality condition $\|\mathbf{K}\|_F = k_0$. For a fixed k_0, the result μ is a fixed number, as well.

3 Incremental Change

Investigating an incremental change in the results, one of the settings has to be changed. Suppose such a change in $\Delta\mathbf{K}$, triggered by a change in k_0 or equivalently in μ. Using only small increments $\Delta\mathbf{K}$ (in norm sense), from Eq.(19)

$$\mathbf{B}\Delta\mathbf{K}\cdot\mathbf{J}+(\mathbf{A}+\mathbf{BK})\Delta\mathbf{J}+\Delta\mathbf{J}\cdot(\mathbf{A}+\mathbf{BK})^T+\mathbf{J}\Delta\mathbf{K}^T\mathbf{B}^T=\mathbf{0} \tag{22}$$

$$\text{col}\ \Delta\mathbf{K}^T \triangleq \mathbf{U}_p\ \text{col}\Delta\mathbf{K} \tag{23}$$

$$[(\mathbf{A}+\mathbf{BK})\oplus\mathbf{I}]\text{col}\Delta\mathbf{J}+[\mathbf{J}^T\otimes\mathbf{B}+(\mathbf{B}\otimes\mathbf{J})\mathbf{U}_p]\text{col}\Delta\mathbf{K}=\mathbf{0} \tag{24}$$

$$\text{col}\Delta\mathbf{J}=-\mathbf{H}_1^{-1}\mathbf{H}_2\text{col}\Delta\mathbf{K}=\mathbf{H}_8\text{col}\Delta\mathbf{K} \tag{25}$$

where colM is an operator concatenating all the columns of $\mathbf{M}$. $\mathbf{U}_p$ is a permutation matrix. Remember the correspondance

$$\text{col}(\mathbf{ABC})=(\mathbf{C}^T\otimes\mathbf{A})\text{col}\ \mathbf{B}. \tag{26}$$

Similarly to Eq.(25), from Eq.(18)

$$(\mathbf{B}\Delta\mathbf{K})^T\mathbf{P}+(\mathbf{A}+\mathbf{BK})^T\Delta\mathbf{P}+\mathbf{PB}\Delta\mathbf{K}+\Delta\mathbf{P}\cdot(\mathbf{A}+\mathbf{BK})\ + \tag{27}$$

$$+\Delta\mathbf{K}^T\mathbf{RK}+\mathbf{K}^T\mathbf{R}\Delta\mathbf{K}\ =\ \mathbf{0}\,. \tag{28}$$

Using the abbreviations of Appendix A

$$\text{col}\Delta\mathbf{P}=-\mathbf{H}_1^{-1}(\mathbf{H}_6+\mathbf{H}_7)\text{col}\Delta\mathbf{K}\,. \tag{29}$$

Considering small changes in the results Eq.(17), a small change of $\Delta\mu$ in Eq.(17) is introduced as an excitation

$$(\mathbf{RK}+\mathbf{B}^T\mathbf{P})\Delta\mathbf{J}+(\mathbf{R}\Delta\mathbf{K}+\mathbf{B}^T\Delta\mathbf{P})\mathbf{J}+\Delta\mu\cdot\mathbf{K}=\mathbf{0} \tag{30}$$

$$[\mathbf{I}\otimes(\mathbf{RK}+\mathbf{B}^T\mathbf{P})]\text{col}\Delta\mathbf{J}+(\mathbf{J}\otimes\mathbf{R})\text{col}\Delta\mathbf{K}+(\mathbf{J}\otimes\mathbf{B}^T)\text{col}\Delta\mathbf{P}+\Delta\mu\cdot\mathbf{K}=\mathbf{0}\,. \tag{31}$$

Employing the intermediate results col$\Delta\mathbf{J}$ and col$\Delta\mathbf{P}$ of Eqs.(25) and (29),

$$\text{col}\Delta\mathbf{K}=\Delta\mu\cdot\mathbf{H}_{11}\text{colK} \tag{32}$$

and

$$\mathrm{col}\Delta\mathbf{P} = -\Delta\mu \cdot \mathbf{H}_1^{-1}(\mathbf{H}_6 + \mathbf{H}_7)\mathbf{H}_{11}\mathrm{col}\mathbf{K} = \Delta\mu \cdot \mathbf{H}_9\mathbf{H}_{11}\mathrm{col}\mathbf{K} \; . \tag{33}$$

The increment of the performance is finally achieved from

$$\Delta I_x = \mathbf{x}_0^T\Delta\mathbf{P}\mathbf{x}_0 = \mathrm{col}[\mathbf{x}_0^T\Delta\mathbf{P}\;\mathbf{x}_0] = (\mathbf{x}_0^T \otimes \mathbf{x}_0^T)\mathrm{col}\Delta\mathbf{P} \tag{34}$$

$$\Delta I_x = \Delta\mu \cdot (\mathbf{x}_0^T \otimes \mathbf{x}_0^T)\mathbf{H}_9\mathbf{H}_{11}\mathrm{col}\mathbf{K} \; . \tag{35}$$

From Eqs.(32) and (35)

$$\|\mathrm{col}\Delta\mathbf{K}\|_F \leqq \Delta\mu \cdot \|\mathbf{H}_{11}\|_F \; \|\mathbf{K}\|_F \tag{36}$$

$$\Delta I_x \leq \Delta\mu \cdot \|\mathbf{x}_0^T \otimes \mathbf{x}_0^T\|_F \; \|\mathbf{H}_9\|_F \; \|\mathbf{H}_{11}\|_F \; \|\mathbf{K}\|_F \; . \tag{37}$$

Resolving by elimination of $\Delta\mu$ and using the equality signs for the sake of rough assessment, the norm increment of $\mathbf{K}$ in relation to the increment of I_x results in two profit factors, i.e.,

$$e_r \triangleq \frac{\|\Delta\mathbf{K}\|_F}{\Delta I_x} = \frac{1}{\|\mathbf{x}_0^T \otimes \mathbf{x}_0^T\|_F} \frac{1}{\|\mathbf{H}_9\|_F} = \frac{1}{\|\mathbf{x}_0\|_F^2\|\mathbf{H}_9\|_F} \tag{38}$$

and

$$e_{r,rel} \triangleq \frac{1}{\|\mathbf{H}_9\|_F} \; . \tag{39}$$

Specializing by inserting the LQR optimal result of Eq.(6), one finds $e_r \to \infty$, since $\mathbf{K} = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}$ and $\mathbf{H}_6 + \mathbf{H}_7 = \mathbf{0}$; this result could have been expected for the exact optimal point.

Example. Third-order MIMO system with two inputs: For an unstable system

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -1 & 3 \end{pmatrix} \; , \quad \mathbf{B} = \begin{pmatrix} 0 & 0.5 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \; , \quad \mathbf{C} = \mathbf{B}^T \tag{40}$$

$$\mathbf{x}_0 = (1 \;\; 0.5 \;\; 0.2)^T, \qquad \mathbf{R} = \mathbf{I}_2, \quad \mathbf{Q} = \mathbf{I}_3 \tag{41}$$

and restricting the norm of $\mathbf{K}$ as a whole, the results are illustrated in what follows.

In Fig. 2 the root contour is depicted, i.e., $\lambda[\mathbf{A} + \mathbf{BK}]$ for μ or k_0 as a parameter. For $\mu = 0$ the classical LQR result is marked with a square. For increasing μ up to 0.03, the entire root contour results. The branches at the real axis are changing their direction versus μ several times, see also Fig. 4.

The dependence of the norm $\|\mathbf{K}\|_F$ on the multiplier μ is shown in Fig. 3.

The maximum real part of the closed-loop poles of the root contour is depicted in Fig. 4.

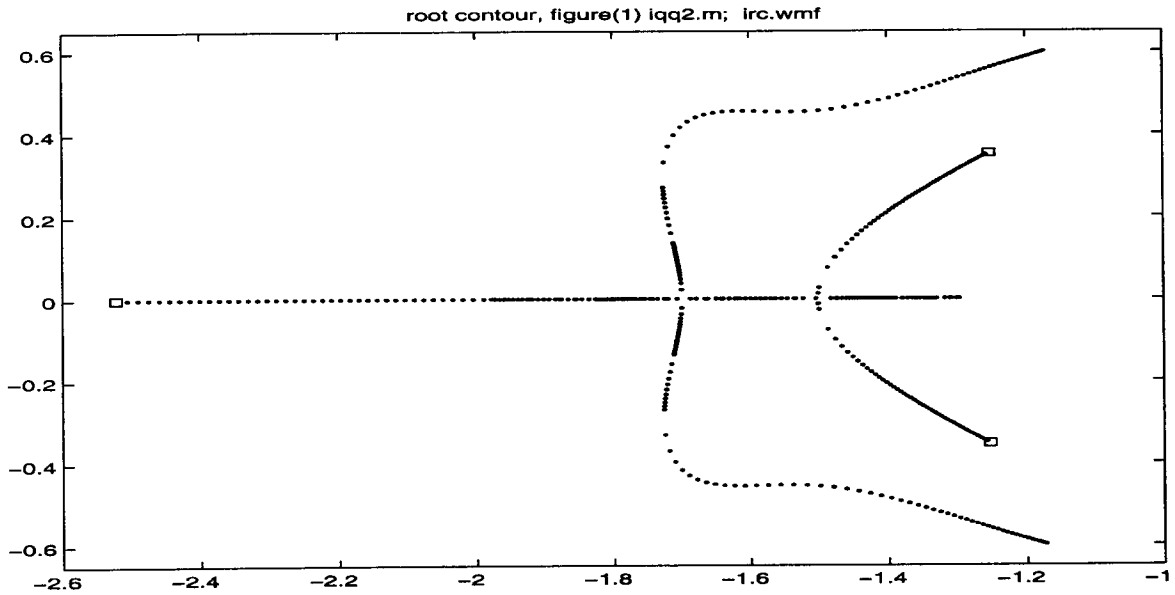


Figure 2: Root contour of the closed loop system resulting from varying the constraint on the norm of $\mathbf{K}$ where $\mu = 0$ corresponds to the square

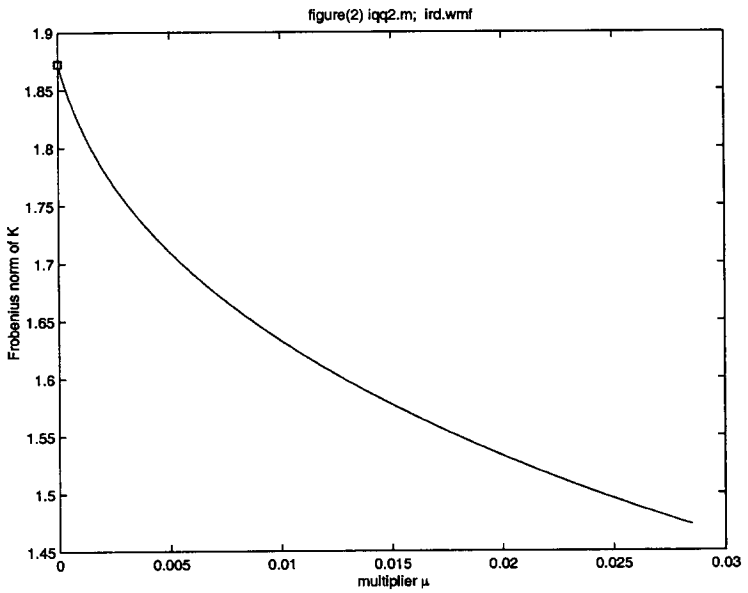


Figure 3: Norm k_0 of the controller $\mathbf{K}$ and its appropriate Lagrange multiplier μ

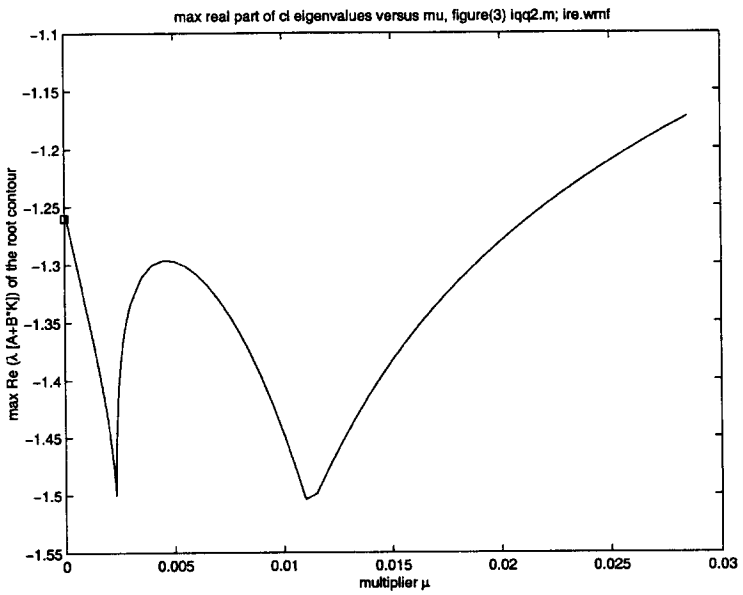


Figure 4: Maximum real part of the closed-loop eigenvalues versus Lagrange multiplier μ

In Fig. 5 one can assess the remarkable reduction of controller norm $\|\mathbf{K}\|_F$ if a slight increase (worsening) in the performance $\mathbf{x}_0^T\mathbf{P}\mathbf{x}_0$ is tolerated. 90° rotation in clockwise sense would show the maximum of performance (as the ordinate) versus controller norm (as the abscissa).

The return difference in frequency domain and its minimum singular value for some specific $\mathbf{R} = \rho\mathbf{I}$ is given (*Anderson, B.D.O., and Moore, J.B., 1989*) by

$$\sigma_{\min}[\mathbf{I} + \mathbf{K}(j\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}] \geq 1 \ . \tag{42}$$

Since LQ optimality is violated due to controller norm restriction, the relation ≥ 1 does not hold for all frequencies and big Lagrange multipliers. The characteristics are depicted for a couple of μ in Fig. 6.

The relation controller norm versus minimum singular value of the return difference is shown in Fig. 7.

Since the result disadvantageously depends on the initial state $\mathbf{x}_0$, the example is carried out for 50 arbitrary $\mathbf{x}_0$ and a fixed value $\mu = 0.005$. The result of the performance $\log I(\mathbf{x}_0)$ and controller norm $k_0(\equiv k_F)$ are portrayed in Fig. 8.

Varying the optimum result by an additional random perturbation, the minimum is made obvious in Fig. 9.

In Fig. 10 the resulting profit factors e_r and $e_{r,rel}$ are depicted for small μ.

Restriction to selected elements of K: The case of only using some entries of $\mathbf{K}$

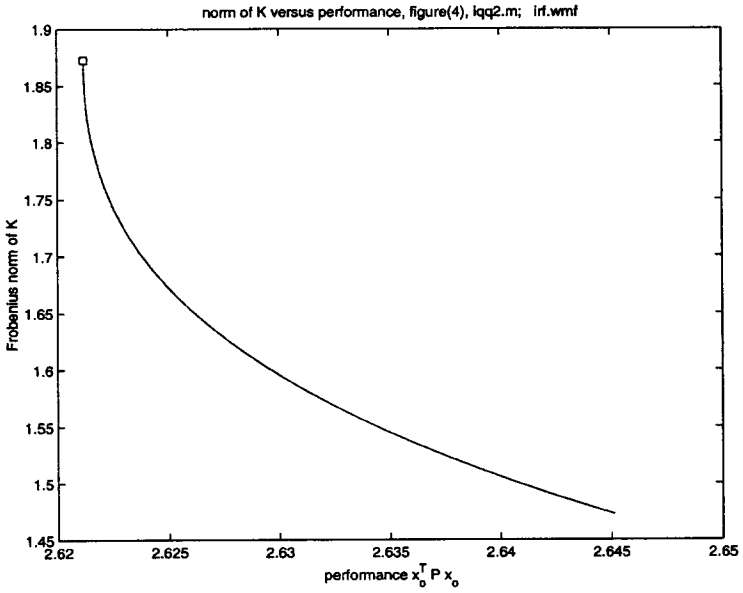


Figure 5: Controller norm versus performance

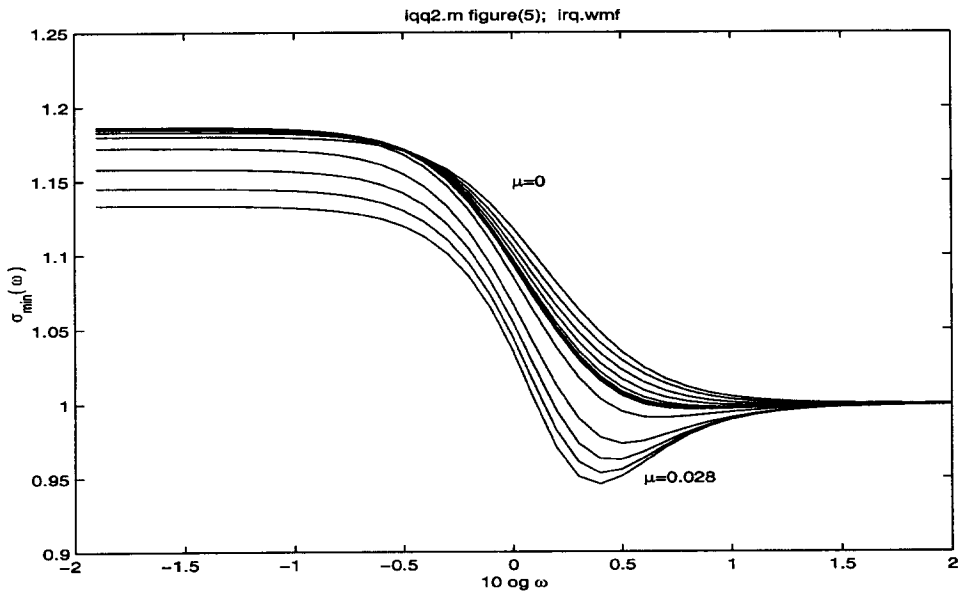


Figure 6: Minimum singular value of the return difference versus frequency

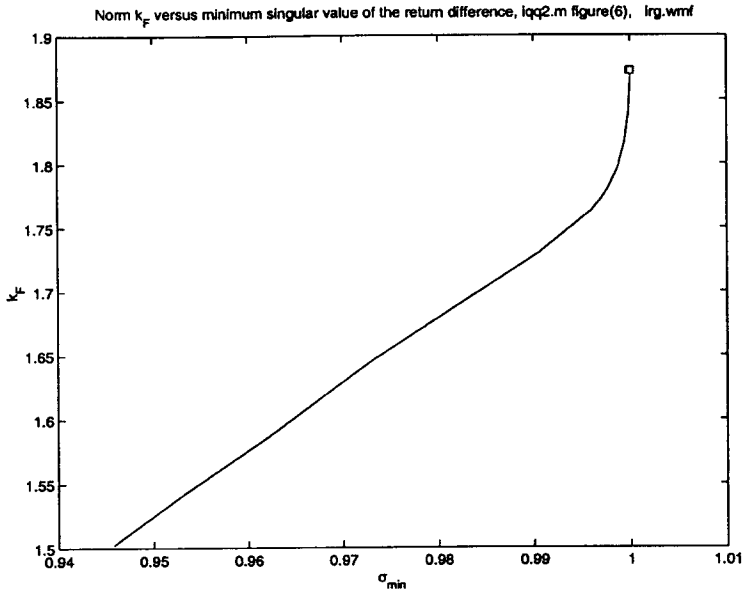


Figure 7: Controller norm $k_F(\equiv k_0)$ versus minimum singular value of the return difference where $\mu = 0$ corresponds to the square

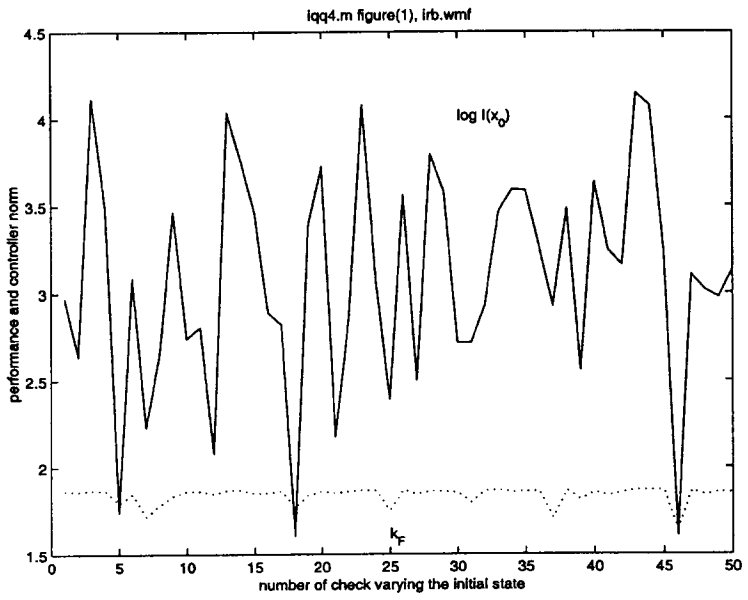


Figure 8: Depicting dependence on the inital state

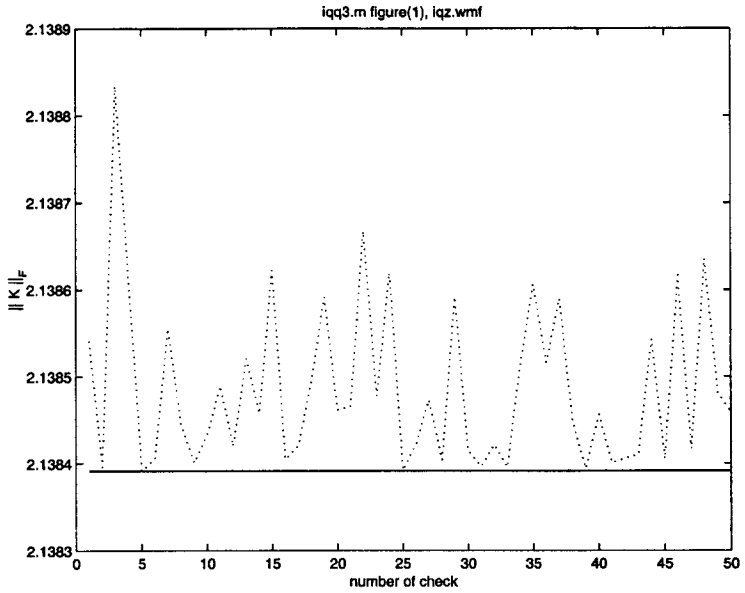


Figure 9: Randomly influenced optimum controller result ('dotted') and precise minimum ('solid') for 50 check points

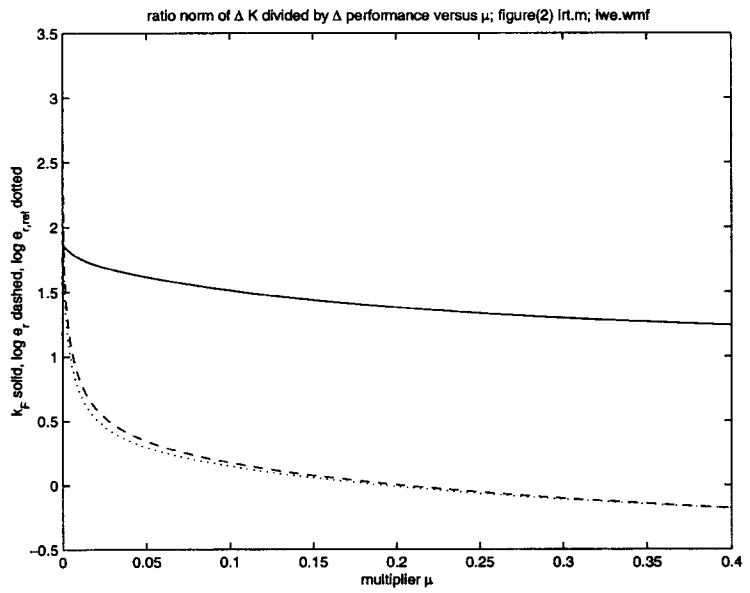


Figure 10: Profit factors e_r and $e_{r,rel}$ and controller norm k_F

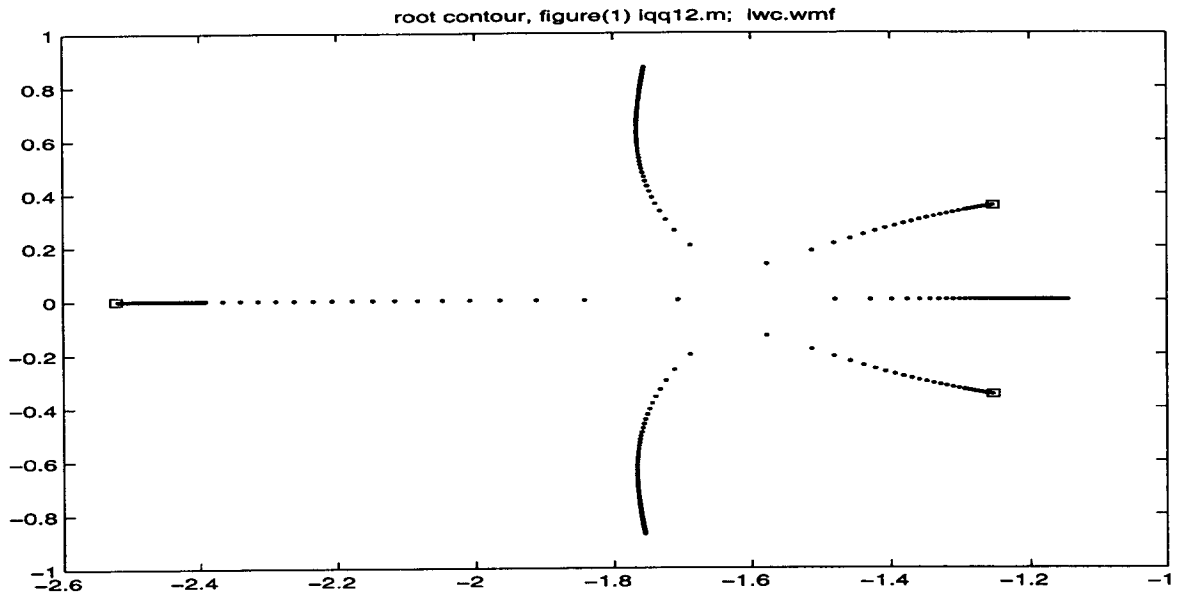


Figure 11: Root locus for partial norm restriction where $\mu = 0$ corresponds to the square

for restriction is also carried out where (see Appendix B)

$$\mathbf{D}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad \mathbf{P}_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{43}$$

Fig. 11 shows the resulting root locus, Fig. 12 presents $\|\mathbf{K}\|_F$ versus μ. There are two solutions in μ for some preselected norm of $\mathbf{K}$.

4 Alternatives

Inclusion of the term μ tr$[\mathbf{K}^T\mathbf{K}]$ could be changed to $\mu\mathbf{x}^T\mathbf{K}^T\mathbf{K}\mathbf{x}$ or $\mu\mathbf{u}^T\mathbf{u}$ with an additional term in $\mathbf{R}$ and $\mathbf{Q}$, i.e. $\mathbf{Q} + \mu\mathbf{K}^T\mathbf{K}$ and $\mathbf{R} + \mu\mathbf{I}$, respectively.

Fig. 13 is a comparison of controller gain profit when using the direct inclusion of the norm of $\mathbf{K}$ or the indirect (termed $\mathbf{K}_h$) one of this section.

5 Conclusion

Supposing similar output noise distribution, the state controller matrix $\mathbf{K}$ is considered responsible for the noise influence on actuating signals. Hence, the controller matrix is suggested being bounded by its norm. Different output noise is valuated in the norm by additional weighting matrices.

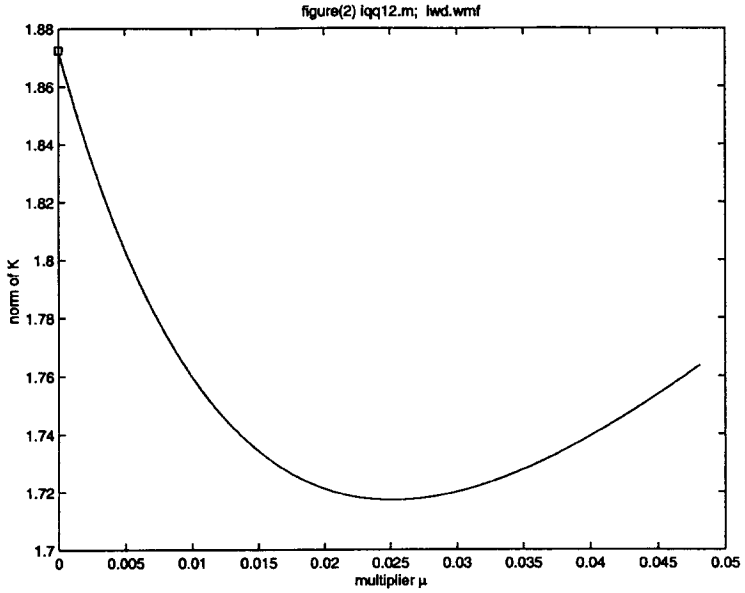


Figure 12: Norm of the controller versus Lagrange multiplier for partial norm restriction

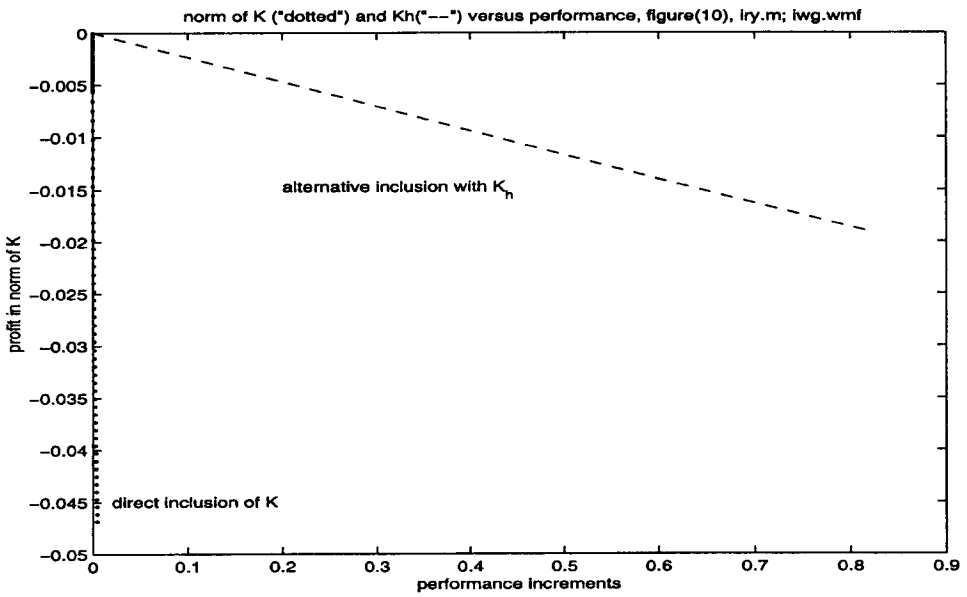


Figure 13: Norm profit comparison

The original LQ controller is modified into a suboptimal one including the restriction of $\mathbf{K}$ in Frobenius norm sense. The resulting algebraic matrix equations for the suboptimal norm-restricted controller $\mathbf{K}$ are more complicated than those for the original LQ controller but are got under control computationally by current matrix solvers.

What is left to the designer or to a supplementary computational task is the observation of the influence on the stability degree of the resulting system.

Balancing stability and noise influence on the actuating variable, both together as a combined performance sensing, is advisable because limiting the noise influence was carried out by an equality condition replacing the inequality conditon.

References

Anderson, B.D.O., and Moore, J.B., 1989, Optimal Control. Linear Quadratic Methods (Prentice-Hall).

Chan, J.T., 1999, Output feedback realization of LQ optimal systems: a general design using data-based synthesis. *Int. J. Control* **72** , pp. 1054-1064.

Chung, L.L., Lin, C.C., and Chu, S.Y., 1992, Optimal direct output feedback of structural control. *J.Engng. Mech. ASCE* **119**, pp. 2157-2173.

Doyle, J.C., Glover, K., Khargonekar, P.P., and Francis, B.A., 1989, State-space solutions to standard H_2 and H_∞ control problems. *IEEE-Trans. AC* **34**, pp. 831-847.

Kuo, B.C., 1980, Digital control systems (Holt, Rinehart and Winsten).

Levine, W.S., Johnson, T.L. and Athans, M., 1971, Optimal limited state variable feedback controllers for linear systems. *IEEE-Trans. AC* **16**, pp. 785-792.

Lin, C.C., Lu, K.H., and Chung, L.L., 1996, Optimal discrete-time structural control using direct output feedback. *Engineering Structures* **18**, pp. 472-480.

Papageorgiou, M., 1991, Optimierung (München Wien, Oldenbourg).

Weinmann, A., 1995, Regelungen, Analyse und technischer Entwurf, Vol. 2, 3rd edition (Springer Wien, NewYork).

Weinmann, A., 2001, Gradients of norms, traces and determinants for automatic control applications, *Int. J. Automation Austria* **9**, H.1/2, pp.36-50

Appendix

A Abbreviations

$$\mathbf{H}_1 \triangleq (\mathbf{A} + \mathbf{BK}) \oplus \mathbf{I} \tag{44}$$

$$\mathbf{H}_2 \triangleq \mathbf{J}^T \otimes \mathbf{B} + (\mathbf{B} \otimes \mathbf{J})\mathbf{U}_p \tag{45}$$

$$\mathbf{H}_6 \triangleq [(\mathbf{PB}) \otimes \mathbf{I}]\mathbf{U}_p + \mathbf{I} \otimes (\mathbf{PB}) \tag{46}$$

$$\mathbf{H}_7 \triangleq [(\mathbf{RK})^T \otimes \mathbf{I}]\mathbf{U}_p + \mathbf{I} \otimes (\mathbf{K}^T\mathbf{R}) \tag{47}$$

$$\mathbf{H}_8 \triangleq -\mathbf{H}_1^{-1}\mathbf{H}_2 \tag{48}$$

$$\mathbf{H}_9 = -\mathbf{H}_4^{-1}(\mathbf{H}_6 + \mathbf{H}_7) \tag{49}$$

$$\mathbf{H}_{11} \triangleq -[\mathbf{I} \otimes (\mathbf{RK} + \mathbf{B}^T\mathbf{P}) \cdot \mathbf{H}_8 + \mathbf{J} \otimes \mathbf{R} - (\mathbf{J} \otimes \mathbf{B}^T)\mathbf{H}_1^{-1}(\mathbf{H}_6 + \mathbf{H}_7)]^{-1} \tag{50}$$

B Restriction to Selected Elements of K

If only a selection of elements is to be included in restricting the norm, some pre- and postmultiplication is chosen. For a general controller, i.e., output controller $\mathbf{K}_y \in \mathcal{R}^{m\times r}$, the diagonal matrices $\mathbf{D}_\nu \in \mathcal{R}^{m\times m}$ and $\mathbf{P}_\nu \in \mathcal{R}^{r\times r}$ are defined. Selecting the (i,j)-Element of $\mathbf{K}_y$ only, $\mathbf{D}_\nu$ and $\mathbf{P}_\nu$ are zero matrices except one single number one at position (i,j), respectively. The matrix

$$\sum_{\nu=1}^{p_1} w_\nu \mathbf{D}_\nu \mathbf{K}_y \mathbf{P}_\nu \quad \in \mathcal{R}^{m\times r} \tag{51}$$

comprises the selected elements of $\mathbf{K}_y$ weighted by w_ν. Then, the squared norm of the matrix containing all the selected elements of $\mathbf{K}_y$ is

$$0.5\ \mathrm{tr}[(\sum_{1}^{p_1} w_\nu \mathbf{D}_\nu \mathbf{K}_y \mathbf{P}_\nu)^T (\sum_{1}^{p_1} w_\nu \mathbf{D}_\nu \mathbf{K}_y \mathbf{P}_\nu)] \ . \tag{52}$$

Using (*Weinmann, A., 2001*)

$$\frac{\partial}{\partial \mathbf{M}} \mathrm{tr}\ [\mathbf{AMBM}^T] = \frac{\partial}{\partial \mathbf{M}} \mathrm{tr}\ [\mathbf{BM}^T\mathbf{AM}] = \mathbf{A}^T\mathbf{MB}^T + \mathbf{AMB}, \tag{53}$$

and $\mathbf{D}_i\mathbf{D}_j = 0 \ \ \forall i \neq j$, the result is

$$0.5\ w_\nu \frac{\partial \| \ \|_F^2}{\partial \mathbf{K}_y} = \sum_{\nu=1}^{p_1} w_\nu \mathbf{D}_\nu \mathbf{K}_y \mathbf{P}_\nu \ . \tag{54}$$

Erweiterte Petrinetze zur intelligenten Planung von Produktionssystemen

T. Perme, D. Noe
Laboratorium für Handhabung und Montage,
Fakultät für Maschinenbau,
Universität Ljubljana

Kurzfassung

Der Kernpunkt eines intelligenten Konzepts für Planung von Produktionssystemen soll eine leicht verständliche und begreifliche Modellierung sein. Mit dem generischen Modell der Aktivitäten (GAM) kann man leicht einen Fertigungs- oder Montageprozess beschreiben, es ist aber nicht möglich eine Aussage über das dynamische Verhalten des Systems zu erhalten. In diesem Text ist eine detaillierte Beschreibung der mathematischen Grundlagen von GAM Modell, der Analogie zwischen GAM und Petri-Netzen, und der erarbeiteten Erweiterungen gegeben, die als eine Basis für Entwurf vom Simulationspaket LASIMCO verwendet werden.

1 Einleitung

Basierend auf der Idee eines rechnergestützten Planungssystems für den Entwurf und die Gestaltung von Produktionssystemen wird ein intelligentes Konzept entwickelt, das für das Planungssystem folgende Ziele definiert:

- Es soll dem Planer jene Daten, die für die Entscheidung wichtig sind, in entsprechender Form zur Verfügung stellen.
- Es soll Tätigkeiten, die der Planer oft wiederholen muss, automatisch durchführen.
- Es soll dem Planer eine Umgebung zur Verfügung stellen, die seine Wissen, Erfahrungen, Kreativität und Intuition mehr zur Geltung kommen lässt.

Der Kernpunkt dieses intelligenten Konzepts soll eine leicht verständliche und begreifliche Modellierung von Produktionssystemen sein, deswegen wird für die Beschreibung der Produktion ein generisches Modell der Aktivitäten (GAM - Generic Activity Model) verwendet, die zusammen mit dem Modell der Fertigungs- und Montagesysteme (SFPM – Shop Floor Production Model) als ein Referenzmodell für Standardisierung und eine Methodologie für die Identifizierung der Erfordernisse der industriellen Automatisierung definiert [1] wurde.

Mit dem GAM-Modell kann man einen Fertigungs- oder Montageprozess auch graphisch beschreiben und eine Aussage über die Struktur eines beobachten Systems bekommen. Es ist aber nicht möglich eine Aussage über das dynamische Verhalten des Systems zu erhalten, deswegen wurden auf der Basis der mathematischen Beschreibung des GAM Modells und der Analogie des GAM Modells und der Petrinetze die Erweiterungen und mathematischen Grundlagen erarbeitet.

2 GAM Modell

Das GAM Modell beschreibt einen Prozess oder ein Teil des Prozesses mit Aktivitäten. Für die Ausführung der Aktivitäten sind ferner noch die Materialien, Mittel und Informationen zu beschreiben, die sich als Eingänge und Ausgänge der Aktivitäten darstellen lassen (Abb. 1).

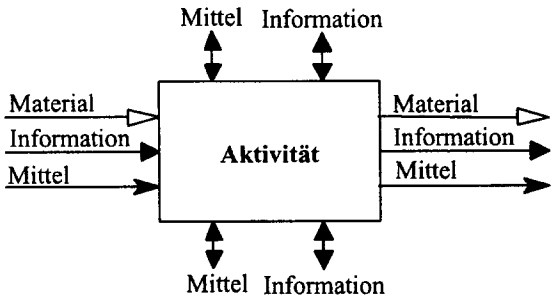


Abb. 1: Generisches Modell der Aktivitäten (GAM)

Durch eine Aktivität werden die Zustände oder Eigenschaften der Eingänge in Ausgänge verwandelt. So lassen sich die Prozesse beschreiben, die ein Teil des gesamten Fertigungsprozesses oder Montageprozesses sind. Die Aktivitäten sind Tätigkeiten, die für die Ausführung eines Prozesses notwendig sind. Vier Grundaktivitäten sind definiert: Transformieren, Transportieren, Überprüfen und Lagern.

Das Material ist ein Erzeugungsobjekt einer Produktionsaktivität. In der Fertigung und Montage sind es Komponenten, welche in einem Fertigungs- bzw. Montageprozess auftreten können, z.B. alle Erzeugnisse, alle möglichen Baugruppen als Zwischenprodukte, Einzelteile und Rohmaterial.

Mit Mittel bezeichnet man alle in einem Prozess beteiligten physischen Mittel, die kein Material sind. Sie umfassen Betriebsmittel für Transformieren, Transportieren, Überprüfen und Lagern. In der Fertigung und Montage sind das: Bearbeitungsmaschine, Werkzeuge, Roboter, Greifer, Handhabungsgeräte, Transportsysteme, Steuerungsanlagen u.a.

Informationen, die für die Ausführung der Prozesse notwendig sind, werden eingeteilt in: Steuerungsinformationen (*Befehle, Zustandsinformationen, Anfragen* und *Rückmeldungen*) und Daten (z.B. NC-Programme, Meldungen, Berichte usw.).

2.1 GAM Modell und Petri-Netze

Die Analogie zwischen GAM-Modellen und Petri-Netzen ermöglicht, dass eine Aktivität direkt in eine Transition abgebildet wird. Ein GAM-Pfeil bedeutet z.B. einen Eingang oder einen Ausgang eines Materials in bzw. aus einer Aktivität. Jeder Pfeil kann einen Eingang und ein Ausgang darstellen. Dementsprechend wird das Material aus dem GAM-Modell in eine Stelle in dem PN-Modell abgebildet. Jede Verzweigung im GAM-Modell mehrt die Pfeile im Petri-Netz-Modell. Gleich verhält es sich auch mit Informationen und Mitteln.

Bei der dynamischen Betrachtung des GAM-Modells kann ein Material in einem bestimmten Eingang warten solange es nicht in der Aktivität „bearbeitet" wird. Dadurch ist es möglich, dass an einem Eingang Material vorhanden ist – oder nicht. Dieses dynamische Verhalten des Prozesses wird mit Marken (Anwesend ja/nein) in den Stellen nachgebildet.

Abb. 2 stellt das GAM-Modell und das Petri-Netz-Modell eines bestimmten Montage-prozesses dar. In ihm werden zwei Teile mit einem Roboter und einem Greifer zusammengesetzt und eine Rückmeldung (fertig) generiert. Dieses Zusammenfügen wird nach einem Befehl (start) ausgeführt.

In dieser Weise kann man aus einer bestimmten Sequenz von Fertigungs- oder Montageoperationen, einschließlich der Handhabungs- und Kontrolloperationen, ein Prozeßmodell erstellen. Die Angaben zur Transitionen, wie die Dauer einzelner Aktivitäten (Bearbeitung, Montageoperationen) und Änderung der Eigenschaften vom Material, Information und/oder Betriebsmittel, sind für die Aktivitäten erforderlich. Diese dienen weiteres als Information zur Überprüfung ob sie mit dem verwendeten Verfahren und den verwendeten Betriebsmittel übereinstimmen. Im vorliegenden Konzept wird die Aufmerksamkeit besonders auf die Transportaktivitäten gerichtet, welche weitgehend von der Fabrikmodellierung (Layout) abhängen.

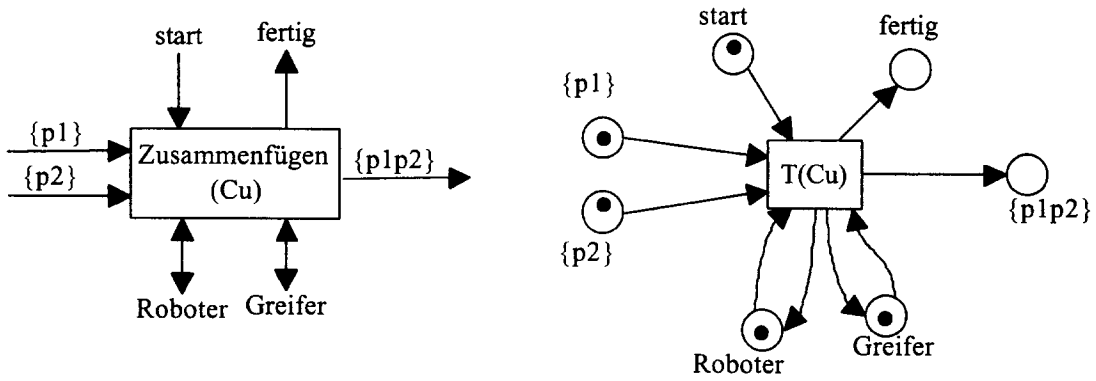


Abb. 2: Das GAM-Modell und das Petri-Netz-Modell eines Montageprozesses

2.3 Mathematische Betrachtung des Aktivitätsmodells

Mit dem GAM-Modell kann man einen Fertigungs- und Montageprozess oder Aktivitäten in einem Produktionssystem nachbilden. Zur Umwandlung dieses Modells in ein Petri-Netz-Modell wird nachfolgend erstmals eine mathematische Betrachtung des GAM-Modells entwickelt, die den Nachweis einer Analogie zwischen beiden Modellen erbringen soll.

Materialien, Informationen und Mittel kann man als eine Menge-Aktivitäten als die Funktionen aus diesen Mengen wie folgt abbilden.

Definition A.1: $M = \{m_1, m_2, \ldots, m_n\}$ ist die Menge der Materialien,
$R = \{r_1, r_2, \ldots, r_m\}$ ist die Menge der Mittel,
$D = \{d_1, d_2, \ldots, d_l\}$ ist die Menge der Informationen,

$F = \{f_1, f_2,...,f_n\}$ ist die Menge der Funktionen, die vier Teilmengen der Funktionen formen, sodaß:
$F_f = \{f \mid f \in F \wedge f = \text{Transformation}\}$
$F_t = \{f \mid f \in F \wedge f = \text{Transportieren}\}$
$F_u = \{f \mid f \in F \wedge f = \text{Überprüfen}\}$,
$F_l = \{f \mid f \in F \wedge f = \text{Lagern}\}$,
$F_f \cup F_t \cup F_u \cup F_l = F$ ... paarweise disjunkt sind.

Mit der Ausführung der Aktivitäten, die als Funktionen nachgebildet werden, verändern sich die Eigenschaften der Materialien und Zustände der Mittel. Für die Beschreibung der Funktionen sind die Zustände, Objekte und Attribute zu definieren, die den Materialien, Mitteln und Informationen zugewiesen sind.

Definition A.2: $O = \{o_1, o_2,....,o_n\}$ ist eine Menge der Objekte,
$A = \{a_1, a_2,....,a_m\}$ ist eine Menge der Attribute,
$S = \{s_1, s_2,....s_l\}$ ist eine Menge der Zustände, wo
(i) $\forall s_i \in S, i \in I_l: s_i = \{r \mid \text{Mittel } r \text{ ist im Zustand } s_i\}$.

(i) bedeutet, daß jedes Element der Menge der Zustände wieder eine Menge der Elemente ist, die sich in diesem Zustand befinden.

Ferner ist eine Eigenschaft als eine Relation auf O definiert, die zu jedem Objekt ihre veränderlichen Charakteristiken bestimmt.

Definition A.3: (i) $E: O \rightarrow A^n$ erklärt zu jedem Objekt eine Eigenschaft, die aus Attributen $a \in A$ zusammengestellt ist.

Dabei ist eine Eigenschaft e ein geordnetes n-Tupel von Attributen $a \in A$, sodaß ein Objekt $o_j \in O, j \in I_n$ mit dem Attribut $a \in A$ gekennzeichnet ist falls,

(ii) $\exists pr_i(e_j), i \in I_n: pr_i(e_j) = a$, wo
(iii) $pr_i: A^n \rightarrow A, (a_1, a_2, ...,a_n) \rightarrow a_i$,

pr_i bedeutet eine Funktion auf einem geordneten n-Tupel, die ein sich an der i-ten Stelle befindendes Element zurückgibt.

Ein Objekt kann auch spezifische Eigenschaften aufweisen, welche in einer geeigneten Form dargestellt werden müssen. So wird zu jedem Objekt sein Lageort speziell definiert:

Definition A.4: (i) $L: O \rightarrow R^{12}$ erklärt zu jedem Objekt eine Lage, die aus einem Positionsvektor und einer Rotationsmatrix gebildet wird.

Mit der folgenden Definition wird zu den Funktionen die Zeit zugeordnet.

Definition A.5: Z: $O, f \rightarrow N$, ordnet zu jedem Objekt und zu jeder Funktion die Zeit solcherart zu, daß $Z(o)$ die Zeit ist, in welchem sich ein Objekt $o \in O$ befindet und $Z(f)$ eine Dauer ist, die eine Aktivität f braucht, um deren Auftrag zu erfüllen.

Die oben angegebenen Grundlagen gestalten die Definition der Funktion der Aktivitäten. Zusätzlich ist festzuhalten, daß Eigenschaften und Lage durch: $e_i = E(o_i)$ und $l_i = L(o_i)$ auch beschrieben werden können.

Definition A.6: Transformieren bedeutet eine Veränderung der Eigenschaften des Objekts von e_i nach e_i'.

$$f_f(e_i) = e_i'; e_i, e_i' \subseteq A^n \text{ und}$$
$$Z'(o_i) = Z(o_i) + Z(f_f).$$

Definition A.7: Transportieren bedeutet die Lage des Objekte so zu verändern, daß gilt:

$$f_t(l_j) = l_j' \text{ und}$$
$$Z'(o_i) = Z(o_i) + Z(f_t).$$

Definition A.8: Überprüfen bedeutet das Durchsuchen der Eigenschaften des Objekts um festzustellen, ob ein oder mehrere Attribut(e) in der Eigenschaft der Objekte enthalten sind:

$$f_u(e,u) = \begin{cases} 1 & falls \quad \forall a \in u \wedge \exists pr_i(e), i \in I_n \Rightarrow pr_i(e) = a \\ 0 & sonst. \end{cases}$$

$$Z'(o_i) = Z(o_i) + Z(f_u).$$

u ist die zu überprüfende Menge von Attributen.

Definition A.9: Lagern bedeutet ein Objekt in einer Lage zu halten. Dabei verändert sich nur die Zeit, welche das Objekt in einem Lager oder an einem Standort bleibt.

$$Z'(o_i) = Z(o_i) + Z(f_l).$$

3 Petri-Netze

Die Grunddefinition der Petri-Netze-Theorie aus [2], [3] und [4] wird erläutert und diese Theorie wird solcherart erweitert, daß die Umwandlung vom Aktivitätmodell in ein Petri-Netze-Modell ermöglicht wird.

3.1 Grundlagen

Grundlage für Petri-Netze bilden Netzwerke, die grundsätzlich als Knoten und Kanten darstellbar sind. In Petri-Netzen gibt es zwei Arten von Knoten: die Stellen und die Transitionen (werden als Kreise und Rechtecke dargestellt). Die Kanten – als Pfeile dargestellt – verbinden die Stellen und Transitionen.

Definition 1.: Petri-Netze sind ein 6-Tupel, $N=(S, T, F, K, W, M0)$, wo gilt:

$S = \{ s_1, s_2, ...s_m\}$ ist eine gleich große Menge von Stellen,
$T = \{t_1, t_2, ...t_n\}$ ist eine gleich große Menge von Transitionen,
$F \subseteq (S \times T) \cup (T \times S)$ ist eine Menge von Verbindungen,
$K: S \rightarrow \{1,2,3, ...\}$ bestimmt die Kapazität für jede Stelle,
$W: F \rightarrow \{1,2,3, ...\}$ bestimmt zu jedem Pfeil des Netzes ein Gewicht,
$M0: S \rightarrow \{0,1,2,3 ...\}$ ist eine Anfangsmarkierung und
$S \cap T = 0$ ist eine disjunkte Menge, wo $S \cup T \neq 0$.

Definition 2.: Für $t \in N$ und für $p \in N$ ist

$\bullet s := \{t \mid (t,s) \in F\}$ eine Menge aller Eingangstransitionen der Stelle s,
$s\bullet := \{t \mid (s,t) \in F\}$ eine Menge aller Ausgangstransitionen der Stelle s,
$\bullet t := \{s \mid (s,t) \in F\}$ eine Menge aller Eingangsstellen der Transition t und,
$t\bullet := \{s \mid (t,s) \in F\}$ eine Menge aller Ausgangsstellen der Transition t.

Eine Abbildung $M: S \rightarrow (0,1,2,..)$ mit $\forall s \in S: M(s) \leq K(s)$ heißt Markierung von N.

Definition 3.: Eine Transition $t \in T$ heißt aktiviert unter M, wenn

(i) $\forall s \in \bullet t: M(s) \geq W(s,t)$
(ii) $\forall s \in t\bullet: M(s) \leq K(s) - W(t,s)$

Bedingung (i) bedeutet, daß beim Schalten genügend Marken vorhanden sind, und daß nach dem Schalten die Markierung an keiner Stelle unter Null sinkt. Bedingung (ii) sichert, daß die Kapazität an keiner Stelle überschritten wird.

Definition 4.: Eine Menge $T' \subseteq T$ heißt aktiviert unter M, falls $\forall t \in T'$ unter M aktiviert ist.

Definition 5.: Schalten von M nach M',

$$M'(s) = \begin{cases} M(s) - W(s,t) & \forall \quad s \in \bullet t \setminus t\bullet, \\ M(s) + W(s,t) & \forall \quad s \in t\bullet \setminus \bullet t, \\ M(s) - W(s,t) + W(t,s) & \forall \quad s \in t\bullet \cap \bullet t, \\ M(s) & sonst. \end{cases}$$

Schalten heißt: Entnahme von Marken aus den Eingangsstellen und Ablage von Marken auf die Ausgangsstellen der aktivierten Transition t gemäß den Kantengewichten.

Die Aktivierungsbedingungen und die oben angeführte Definition der Folgemarkierung bezeichnet man auch als „Schaltregel".

Die Netzdynamik – d.h. die Schaltung von Transitionen – kann auch durch die lineare Algebra beschrieben werden. Netze aus Stellen S und Transitionen T werden durch C^+ und C^- Matrizen beschrieben.

$$C^+_{ij} = \begin{cases} w(t_i, s_j) & \forall \quad s_i \in t_j\bullet \quad \wedge \quad (t_i, s_j) \in F \\ \varnothing & sonst. \end{cases}$$

$$C^-_{ij} = \begin{cases} w(s_j, t_i) & \forall \quad s_i \in \bullet t_j \quad \wedge \quad (s_j, t_i) \in F \\ \varnothing & sonst. \end{cases}$$

C^+_{ij} ist die Anzahl der Marken, die beim Schalten von t_i an s_j – gemäß dem Kantengewicht $w(t_i, s_j)$ – übergeben werden. C^-_{ij} ist die Anzahl der Marken, die beim Schalten von t_j aus s_i abgezogen werden. Mit der Angabe von C^+ und C^- ist ein Petri-Netz eindeutig beschrieben. C^+ und C^- können zur „Inzidenzmatrix" C zusammengefaßt werden:

$$C = C^+ - C^-$$

Die entsprechende Änderung der Markierung durch die Schaltung der aktivierte Transition t_j kann wie folgt beschrieben werden:

$$\Delta M = C_j^T = (c_{1j}, c_{2j}, \ldots, c_{mj}) . \Rightarrow M' = M + C_j^T,$$

(die Markierungen sind dabei als Zeilenvektoren beinhaltet).

3.2 Erweiterung der Grundlagen

Mit der Grundtheorie der Petri-Netze können nicht alle Eigenschaften des zu betrachteten Systems nachgebildet werden. Eine erste Erweiterung ist die Einführung der Zeitachse. Mit der Elementartheorie kann man zwar eine Zeitabhängigkeit der Transitionen feststellen - d.h. die Reihenfolge, in welcher die Transitionen schalten, ermitteln. Das entspräche jedoch nur jenem Sonderfall, daß die Aktivitäten des realen (und abgebildeten) Systems in einem unendlich kurzen Zeit ausgeführt werden.

Die zweite durchgeführte Erweiterung ist die Einführung einer Priorität des Schaltens von Transitionen, mit welcher Konfliktsituationen beseitigt werden sollten. Ein solcher Konflikt wäre zum Beispiel gegeben, wenn zwei Transitionen zum selben Zeitpunkt aktiviert sind jedoch nur eine von beiden schalten darf (wenn z.B. zwei Montageoperationen zu gleicher Zeit ein Betriebsmittel anfordern). Unter Verwendung zusätzlicher Bedingungen kann man den überwiegenden Anteil möglicher Konfliktsituationen verhindern. Manchmal

wird das aber unpraktisch oder gar unmöglich, weswegen zu jeder Transition eine Priorität zugewiesen wird. Wenn nun mehrere Transitionen aktiviert sind, schaltet zuerst jene, welche die größere Priorität hat.

Eine weitere wichtige Erweiterung ist die Einführung von Attributen und Objekten, mit denen die Materialien, Betriebsmittel, Informationen und Aktivitäten vollständig – im Sinne der Transition/Stellen Netze – modelliert werden können. Diese Erweiterungen orientieren sich an gefärbten Petri-Netzen oder an Prädikat/Ereignis-Netzen.

Ebenfalls ist die PN-Theorie für die beschriebene Anwendung durch eine Negation von Verbindungen zu erweitern – ein Zusatz, der sich in der praktischen Anwendung als unersetzlich erwiesen hat. Mit einer Negation kann zum Beispiel die Einstellung einer Montageoperation wegen Störung oder Fehler nachgebildet werden.

Alle durchgeführten Erweiterungen werden in nachfolgenden Definitionen beschrieben.
Erweiterung der Definition 3:

W: $F \rightarrow \{...-3, -2, -1, 0, 1, 2, 3, ...\}$ bestimmt zu jedem Pfeil des Netzes ein Gewicht, welches eine ganze Zahl ist.

Eine Transition $t \in T$ heißt aktiviert unter M, wenn:

$$\text{(i)}\ \forall s \in {}^\bullet t: \begin{cases} M(s) \geq W(s,t) & \forall \quad W(s,t) \geq 0 \\ M(s) < |W(s,t)| & \forall \quad W(s,t) < 0 \end{cases}$$

$$\text{(ii)}\ \forall s \in t^\bullet: \begin{cases} M(s) \leq K(s) - W(t,s) & \forall \quad W(t,s) \geq 0 \\ M(s) > |W(t,s)| & \forall \quad W(t,s) < 0 \end{cases}$$

Diese Erweiterung bedeutet, daß wenn z.B. eine Bedingung erfüllt ist ($M(s) > 0$) aber das Gewicht einer Eingangsverbindung (s,t) einen negativen Wert hat ($W(s,t) < 0$) diese Transition t nicht schalten kann.

Definition 6.: Mit der Einführung der Priorität und des Faktors „Zeit" bestehen Petri-Netze aus 8-Tupel (N, P, Z), wo

$N = (S, T, F, K, W, M_0)$
P: $T \rightarrow \{0,1,2,..,9\}$ bestimmt zu jeder Transition des Netzes eine Priorität
Z: $T \rightarrow \{0,1,2,3,..\}$ bestimmt zu jeder Transition des Netzes eine Zeit.

Definition 7.: Eine aktivierte Transition t_i darf unter M schalten und heißt dann schaltbar, wenn

(i) $\forall s \in {}^\bullet t_i: s^\bullet \cap T' = \{t_i\}$ oder
(ii) $\forall t \in T' \cap \bigcup_{s \in {}^\bullet t_i} {}^\bullet s\ ,\ t \neq t_i: P(t_i) > P(t)$.

Bedingung (i) bedeutet, daß t_i in keiner Konfliktsituation steht. Bedingung (ii) bedeutet: Wenn zwei oder mehrere Transitionen mit den selben Eingangsstellen unter M aktiviert sind, darf nur die Transition schalten, welche die größere Priorität hat.

Definition 8.: $T = \{(t_1, d_1) (t_2, d_2)....(t_n, d_n)\}$ ist eine Menge der teilgeschaltenen Transitionen mit den Elementen (t_k, d_k), $t_k \in T$ und $d_k \in \{1,2,3...\}$, wo jedes t nur teilgeschaltet (Def. 9) hat.

Eine schaltbare Transition t unter M, wenn $Z(t) = z > 0$, schaltet in zwei Teilen: Teilschalten und Endschalten.

Definition 9.: Teilschalten von der durch (i) in der erweiterten Definition 3 aktivierten und schaltbaren Transition t_i unter M:

$$M'(s) = \begin{cases} M(s) - W(s,t_i) & \forall \quad s \in \bullet t_i \\ M(s) & sonst. \end{cases}$$

$M' = M - (C_i^-)^T$ und

$T' = T \cup \{(t_i, z_i)\}$.

Teilschalten von t_i bedeutet die Entnahme von Marken aus den Eingangsstellen der Transition t_i und Zugabe eines neuen Elementes (t_i, z_i) zu den T.

Definition 10.: Endschalten von der durch (ii) in der erweiterten Definition 3 erlaubte Transition t_i unter M:

(i) $$M'(s) = \begin{cases} M(s) + W(s,t_i) & \forall \quad s \in t_i \bullet \\ M(s) & sonst. \end{cases}$$

$M' = M + (C_i^+)^T$, wo

$t_i = t_k \mid (t_k, d_k) \in T$.

(ii) $T' = T \backslash \{(t_k, d_k)\}$ und

(iii) $(t,d)' = (t, d - d_k)$ für $\forall (t,d) \in T'$ wobei für t gilt:
$\forall s \in \bullet t: M(s) < |W(s,t)|$ falls $W(s,t) < 0$

Endschalten von t_i bedeutet die Ablage von Marken auf die Ausgangsstellen der Transition t_i (i), die Entfernung des Elementes (t_k, d_k) aus T (ii) und die Subtraktion der d von allen Elementen der T um d_k, bei denen t alle negative Eingangsbedingungen erfüllt(iii).

Definition 11.: $T_s \subseteq T$ ist eine Menge von teilgeschaltenen und endschaltbaren Transitionen mit den Elementen (t,d), wenn

$\forall t \in T_s \wedge \forall s \in t\bullet: M(s) \leq K(s) - W(t,s)$.

Definition 12.: Die Transition t_k aus dem Element $(t_k\ d_k) \in T$ heißt Transition mit der kleinsten restlichen Zeit, falls:

$$\forall (t,d) \in T \mid t_k \neq t : d_k \leq d.$$

Definition 13.: Petri-Netze werden mit Attributen, Objekten und räumlichen Stellen erweitert:

$A = \{a_1, a_2, \ldots a_n\}$ ...Menge der Attribute.
$O = \{o_1, o_2, \ldots o_n\}$ ...Menge der Objekte
$E: O \rightarrow A^n$ definiert eine Eigenschaft zu jedem Objekt
$S_a \subseteq S$ ...Teilmenge der attributiven Stellen
$S_p \subseteq S$ ...Teilmenge der räumlichen Stellen
$S_a \cap S_p = \varnothing$
$O_s : O \rightarrow S_p \cup T$ definiert zu jeden Objekt eine räumliche Stelle oder Transition
$P: S_p \rightarrow N^{12}$ definiert eine Lage an räumliche Stellen
$E_a: S_a \rightarrow A$ definiert ein Attribut an attributive Stellen

Definition 14.: Eine Transition darf maximal eine räumliche Stelle in den Mengen der Eingang- und Ausgangsstellen haben:

$\forall t \in T: t\bullet \cap S_p = \{\varnothing\} \vee \{s\} \wedge \bullet t \cap S_p = \{\varnothing\} \vee \{s\}$

Definition 15.: Eine Transition ist teilschaltbar unter M, wenn

(i) erweiterte Def. 3. (i) und
(ii) $\forall s \in t\bullet \cap S_a \wedge \exists i \in \{1,2,\ldots n\}: E_a(s) = \mathrm{pr}_i(t)$, wo
$\mathrm{pr}_i: A^n \rightarrow A; (a_1, a_2, \ldots, a_n) \rightarrow a_i$.

Bedingung (ii) bedeutet, daß ein Objekt alle mit den attributiven Stellen bestimmten Attribute besitzt.

Definition 16.: Eine Transition, die normal schaltbar ist und teilschaltbar nach (ii) der Definition 15, schaltet unter M nach M' wie in Definition 5 und ändert weiter noch e_i, sodaß:

(i) $\forall s \in \bullet t \cup Sa: E'(s) = E(s) \cup Ea(s)$
(ii) $\forall s \in t\bullet \cup Sa: E'(s) = E(s) \setminus Ea(s)$

3.3 Modellierung mit erweiterten Petri-Netzen

Mit der mathematischen Beschreibung der erweiterten Petri-Netze kann man nun ein GAM-Modell (die Ähnlichkeit der mathematischen Beschreibung ist evident) übernehmen oder direkt ein Modell des beobachteten Systems erstellen. Für eine praktische Anwendung ist eine graphische Darstellung und auch eine graphische Nachbildung eines Modells unbedingt erforderlich. Tabelle 1 stellt alle Elemente der verwendeten Petri-Netze graphisch dar. Zu jedem Element wird weiters eine Datenstruktur bestimmt, mit denen ein Element ausführlich beschrieben wird.

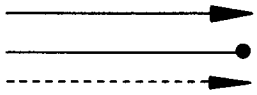
PN Elemente	Datenstruktur	graphische Darstellung
Einfache Stelle	Name Kapazität Belegung	
Attributive Stelle	Name Kapazität Belegung Attribut	
Räumliche Stelle	Name Kapazität Belegung (Objektmarke) Lage (Position, Orientierung)	
Einfache Transition	Name	
Zeitliche Transition	Name Typ Zeit Belegung*	
Verbindung	Gewicht Typ** Bedingung***	
Einfache Marke		• oder 1
Objektmarke	Attribute Zeit Typ Länge und Breite Lage (Position, Orientierung)	beliebige Form, hängt von Typ ab

Tabelle 1: Graphische Darstellung und Datenstruktur der Elemente von Petri-Netzen

Fast alle Daten der Element-Datenstruktur sind schon mittels Definitionen festgelegt. Ausgenommen davon sind lediglich die Belegung von Transitionen, der Typ und die Bedingung von Verbindungen und einige Daten von Objektmarken.

Im Prinzip kann eine zeitliche Transition auch mit Marken oder Objektmarken belegt werden. Wann immer eine Transition teilgeschaltet ist, wird sie mit mindestens einer Marke belegt (dies auch wenn die Objektmarken in das Modell eingegeben werden). Man kann so eine Objektmarke auch in eine Transition – nicht nur in eine Stelle – eintragen.

Eine Verbindung kann normal oder informatorisch sein, was durch den Verbindungstyp beschrieben wird. Eine informatorische Verbindung entspricht weitestgehend einer normalen, außer daß bei der Verbindung attributiver Stellen beim Schalten kein Attribut von einer Objektmarke genommen bzw. zu einer Objektmarke hinzugefügt wird.

Zu jeder Verbindung kann auch eine Bedingung bestimmt werden. Diese Bedingung wird aus den Attributen zusammengestellt und ist nur dann von Bedeutung, wenn die Verbindung zwischen einer räumlichen Stelle und einer zeitlichen Transition besteht ($s \in S_p$, t). Erst wenn eine Objektmarke in der räumlichen Stelle diese Bedingung erfüllt, kann die Transition teilschalten (wenn sie schon nach anderen Bedingungen aktiviert ist).

Eine Objektmarke wird auch mit den Werten Typ, Zeit, Länge und Breite beschrieben. Mit Typ wird vor allem die Darstellung festgelegt. Die Angabe für die Zeit wird verwendet, wenn eine Objektmarke in eine Transition gestellt wird. Länge und Breite definieren zu einer Objektmarke eine physische Eigenschaft, die für das räumliche Verhalten mit anderen Objekten (Objektmarken) verwendet werden.

Bei der bloßen Verwendung von Petri-Netzen kann man anhand der oben angeführten Definitionen und Elemente noch kein wirkliches „Modell" erzeugen. Erst durch eine detaillierte Analyse des nachzubildendes Systems und vor allem mit ausreichender Modellierungserfahrung mit Petri-Netzen kann man ein Modell erstellen, welches für eine spätere Simulation verwendet werden kann.

Es ist nicht Absicht dieser Arbeit eine Dokumentation oder eine Anweisung zur Verwendung von Petri-Netze zu erstellen. Es soll vielmehr ein Werkzeug entworfen werden, mit welchem der Modellaufbau unter Verwendung von Petri-Netzen für den Anwender transparent bleibt. Es werden deshalb in Tabelle 2. nur Modelle der grundlegenden Aktivitäten dargestellt.

Jede Aktivität wird mit einer zeitlichen Transition, zwei räumlichen Stellen für Objekte und einer einfachen Stelle für Betriebsmittel nachgebildet. Das Modell einer Transformation zur Abbildung einer Montageoperation hat darüber hinaus noch eine attributive Stelle (Fügeteil) und zwei einfache Stellen (I/O Informationen). Das Modell einer Überprüfung hat eine attributive Stelle mit dem Attribut „f", mit welchem überprüft wird, ob in der Baueinheit B ein Fügeteil f schon eingefügt wurde oder nicht. Das Lagern könnte zwar mit nur einer einzigen räumlichen Stelle nachgebildet werden – es würde dann aber das Betriebsmittel nicht betrachtet.

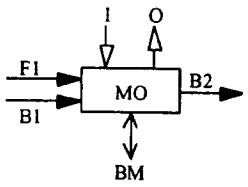
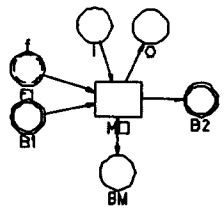
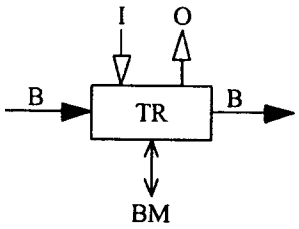
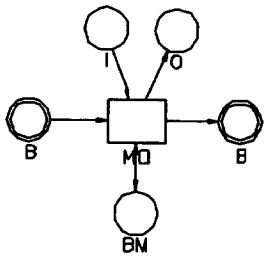
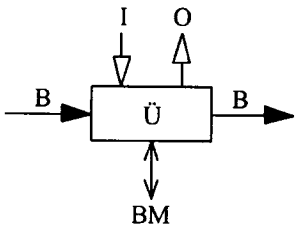
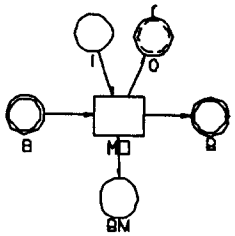
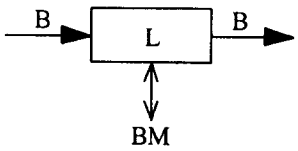
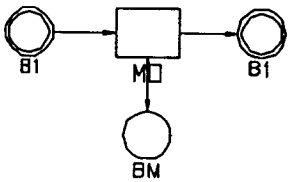
Aktivität		GAM – Darstellung	PN - Modell
Transformation	f_f		
Transport	f_t		
Überprüfung	f_u		
Lagern	f_l		

Tabelle 2: Nachbildung von GAM-Modellen mit Petri-Netzen

4 Zusammenfassung

Anhand der erweiterten Grundtheorie von Petri-Netzen wurde ein Modellierungsprogramm (PN_EDIT) und ein Simulationsprogramm (PN_EXE) entwickelt [5]. PN_EDIT – geschrieben in AutoLISP – wurde für die Verwendung als AutoCAD-Modul bestimmt. Es ermöglicht den einfachen und anwendungsfreundlichen Aufbau eines EPN-Modells. PN_EXE – ein in C und ArxADS geschriebenes

Programm – dient zur Simulation eines EPN-Modells. Es ist ebenfalls als AutoCAD-Modul konzipiert und ermöglicht die Simulation und Animation unter Verwendung von AutoCAD 3D Objekten.

Auf Basis der Konzepte: Virtuelle Produktion und mit Hilfen von EPN (PN_EDIT und PN_EXE) wurde das Programmpaket LASIMCO entworfen. Dieses besteht aus vier Hauptprogrammeinheiten für Layoutentwurf, Modellbildung, Simulation und Bewertung. Das Programmpaket LASIMCO wurde an vielen realen Beispielen verwendet [6][7] und hat gezeigt, dass es für die Planung von verschiedenen Produktionssystemen geeignet ist und dass eine leicht verständliche und begreifliche Modellierung von Produktionssystemen für die praktische Anwendung von Planungssystemen sehr bedeutend ist.

5 Literaturverzeichnis

[1] ISO 10314, Industrial automation – Shop floor production – Part 1.: *Reference model for standardisation and a methodology for identification of requirements.*

[2] Peterson, J. L.: *Petri Net Theory and the Modeling of Systems*, Prentice-Hall. Inc., New York, 1981

[3] Reisig, W.: *Petrinetze; Eine Einführung*, Springer-Verlag, 1991.

[4] Murata, T.: Petri Nets: Properties, Analysis and Applications, *In Proceedings of the IEEE*, Vol. 77, No. 4, April 1989.

[5] Perme, T.: *Ein neues Simulationskonzept zur rechnergestützten Planung von Montagesystemen*, Disertation, Wien, 1999.

[6] Perme, T., Noe, D., Kopacek, P.: *Modelling and Simulation of Assembly Systems in CAD-Environment Using Extended Petri Nets*, Ventil 5(1999)4, Ljubljana 1999.

[7] Perme, T., Noe, D.: Computer Aided Planning of Manufacturing Systems Using LASIMCO Simulation Package, *In proceeding of 2000 Internationa CIRP Design Seminar* (Editor Shpitalni, M.), Haifa, Israel 2000.

Bionic Solutions for Planar Tentacle Grippers

Bernard Favre-Bulle[1], Dipl.-Ing., Dr. techn.,
Christian Busch[2], Dipl.-Ing. (FH),
Rudi Feurstein[2], Prof. (FH), Dipl.-Ing.,
[1]Vienna University of Technology,
Institute for Handling Devices and Robotics,
TU-Wien, A-1040 Wien, Favoritenstraße 9–11 and
[2]Fachhochschule Vorarlberg, A-6850 Dornbirn
email: favre@cable.vol.at

August 8, 2002

Summary: *The manipulation of soft- or variably structured objects in automation systems often imposes difficulties due to the need of adaptive gripper devices with specialized hardware features. Conventional manipulators use end-effectors like grippers to handle objects. Hyper-redundant mechanisms have a much higher degree of kinematic freedom in joint space than in world space and permit a bionic approach of grasping. The geometry of this class of manipulators may be similar to tentacles, snakes or animal trunks. Whole-arm-grasping is a method to entwine an object with tentacle-like movements. At the final stage, the object is completely or partially surrounded by the manipulator arm. This paper discusses a new tactical grasp-planning method which uses a geometrical teaching-algorithm to train a neural net for the procedural knowledge of the grasping process. The principle of the teaching-algorithm is presented. Two different neural net structures are compared regarding their ability to learn the procedural knowledge. Finally, the performance of the neural nets in the grasping process is analyzed by means of a performance criterion. The results are discussed on the basis of simulations.*

1 Introduction and Target

Object grasping is a central function in robotic systems. Industrial manipulators have specialized grippers for the handling of solid or structurally stable objects. For delicate handling tasks, tactile sensors and intelligent actor control algorithms need to be implemented.

In the present paper, the authors discuss a different approach of object grasping. It is based on a single-arm hyper-redundant tentacle-like mechanism, which takes hold of

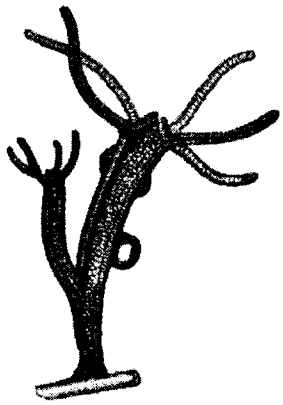


Figure 1: The sweet-water polyp *Hydra* (source: *Lenhoff H. M., and Lenhoff S. G., 1988*)

objects by wrapping around them with a large part of its arm. This method is referred to as *whole-arm-grasping*, and is based on several investigations on hyper-redundant manipulators as e. g. in *Chirikjian, G. S., 1992,* and in *Mochiyama, H., 2001*. The new approach presented here is the implementation of a neural net for arm-controlling to accomplish the grasping process. The aim of the paper is to show a bionic solution of object grasping with neural nets, including a discussion of the performance of two different neural net types for the trajectory generation.

1.1 Tentacle Grasping: An Inspiration from Nature

In natural evolution, a wide variety of organisms have evolved which exhibit organs for flexible grasping and manipulation of objects. In many cases, these organs also function as actuators for locomotion and as weapons to defend enemies or attack quarry. The flexible tentacles often carry sensor systems to accomplish secondary functions like tactile sensing, chemical sensing or even act as sound- or light sensors. Some animals spread poison via their tentacles. In some organisms, the neural system is rather centralized. In other cases, it forms a peripheral network which is very robust in function, even when a part of it is destroyed. One example of the second kind is the sweet-water polyp *Hydra* (Fig. 1). Various research has been done in the past to benefit from the concepts of nature for artificial mechanisms, as e. g. in *Hannan, M. W., and Walker, I. D., 2001* or in *Immega, G., and Antonelli, K., 1995.* For the following investigations, we will use a centralized artificial neural net with time delays for the motion coordination of a discrete planar serial-link manipulator.

2 A Hyper-Redundant Planar Manipulator

The basis for the investigations is a hyper-redundant planar manipulator with revolute joints which is fixed to the world frame with its first rotational joint (Fig. 2). The kinematics and dynamics of hyper-redundant planar mechanisms have been thoroughly analyzed

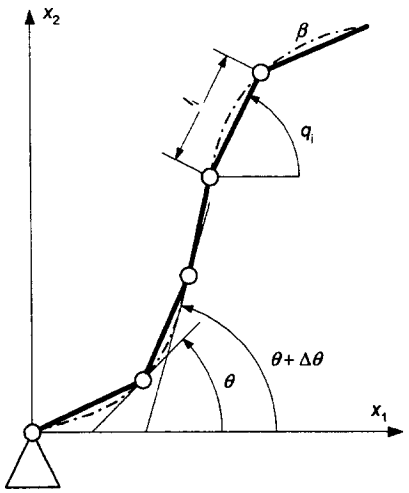


Figure 2: A planar hyper-redundant serial-chain manipulator with $\mathbf{q} = (q_1, \cdots, q_{i+1})^T$

in the past, see e. g. *Žlajpah, L., 1996* and *1998*. There are several ways for the representation of the actual manipulator configuration, from a set of joint angles to a spline formulation of the manipulator's continuous backbone curve, where the local curvature plays an important role. To derive a representation, we start with the simple parameter form of the backbone curve

$$\mathbf{x} = \mathbf{x}(s,t) = \begin{pmatrix} x_1(s,t) \\ x_2(s,t) \end{pmatrix}, \tag{1}$$

where s is the *curve parameter*, $s \in [0,1]$ and t ist the time. In order to express Eq.(1) in terms of the curvature κ, we first introduce the angle θ which is measured counterclockwise from the horizontal coordinate axis towards the tangent in a specific backbone curve point. So we get

$$x_1(s,t) = \int_0^s l(\sigma,t)\cos\theta(\sigma,t)d\sigma \tag{2}$$

$$x_2(s,t) = \int_0^s l(\sigma,t)\sin\theta(\sigma,t)d\sigma, \tag{3}$$

where l is the *extensibility-function*, which is 1 in case of an inextensible manipulator with rotational joints, as assumed here. The curvature in the planar case is $\kappa = \frac{1}{l}\frac{\partial\theta}{\partial s}$, and $\kappa = \frac{d\theta}{ds}$ for inextensible kinematics. With $\theta(s) = \int_0^s l(\sigma)\kappa(\sigma)\,d\sigma$ and Eq.(2) and Eq.(3) we get the curve representation

$$x_1(s,t) = \int_0^s l(\sigma,t)\cos\left(\int_0^\sigma l(\nu,t)\kappa(\nu,t)d\nu\right)d\sigma, \tag{4}$$

$$x_2(s,t) = \int_0^s l(\sigma,t)\sin\left(\int_0^\sigma l(\nu,t)\kappa(\nu,t)d\nu\right)d\sigma. \tag{5}$$

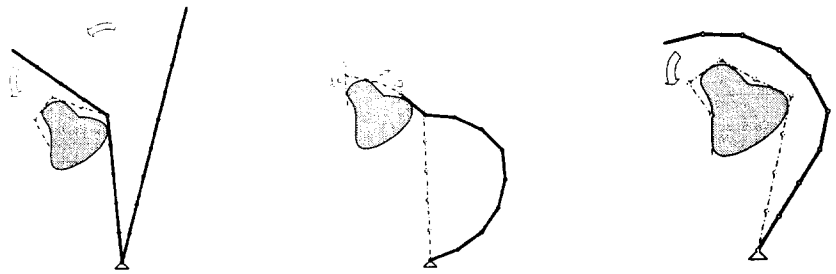


Figure 3: "Piece-wise linear wrapping", "Follow The Leader" and "Lasso-method"

This representation is important for the calculation of the *direct shape kinematic problem*, which leads to a cartesian point of the backbone curve for a given curve parameter-value s and a certain curvature function $\kappa = \kappa(s)$. The curvature function itself is an expression for the (time variable) grasping configuration of the tentacle.

G. S. Chirikjian demonstrates the usage of *modal functions* to represent the backbone of a hyper-redundant manipulator in *Chirikjian, G. S., 1992*. He is interested in solving the inverse kinematic problem for the tip of the manipulator and shows, that modal functions can have a closed form solution for this problem. However, the problem of modal singularities arises. For our present problem, we need to control the shape of the whole arm rather than the position of the tool-center-point, so modal functions are not the first choice.

As an alternative, we can represent the curve by piecewise polynomial splines (direct parameterization)

$$\mathbf{x} = \mathbf{x}(s,t) = \sum_{i=1}^{N} s^i \mathbf{v}_i(t), \tag{6}$$

where $\{\mathbf{v}_i(t)\}$ is a set of time varying vectors of spline coefficients. This permits the representation of finite curve sections, even with non-steady transients between curve segments. Thus, the manipulator can be split up into sections, where each section is described by a set of spline coefficients. All further discussions are based on a spline representation.

3 Teaching Procedural Grasping-Knowledge

We like to show that a neural net is capable of coordinating the grasping procedure for a tentacle-like manipulator. The process of grasping can be defined as a sequence of manipulator configurations, where the links of the manipulator iteratively approach the object contour with each step. In order to accomplish this, the controller needs sensor information to react to the time varying distance between the manipulator body and the object. Further, the performance of the system needs to be be optimized by teaching the net with pre-calculated data. Thus, we need to implement an adaptive neural net MIMO-system and a teaching algorithm, specialized in the grasping process. The neural net will

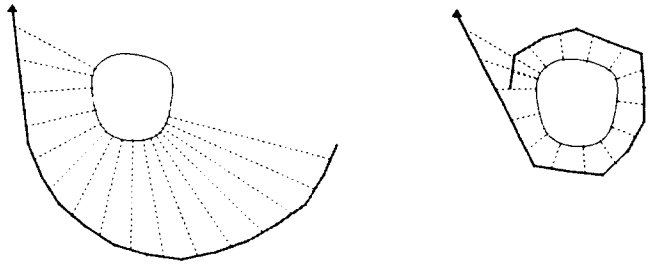


Figure 4: "Nearest-neighbour" sensorics

have to learn *procedural knowledge about grasping* from the teacher. For this dynamic task, neural architectures with time delays are required.

3.1 Whole-Arm Grasping: Problem Description

To successfully teach a neural net for the grasping-process, we need a teacher with an *optimal performance* for a given grasping method. For obstacle avoidance tasks, the authors *Reznik, D., and Lumelsky, V., 1993* and *Williams, R. L., and Mayhew, J. B., 1997* show approaches like "Follow the Leader" (a tool-center-point based approach) and a piece-wise linear wrapping method, both sketched in Fig. 3.

The piecewise linear wrapping method has obvious drawbacks what the detailed contour matching is concerned. Follow-The-Leader methods are useful for obstacle avoidance tasks, but not the first choice for grasping, since only the limbs in contact with the object contribute to the grasping at a given moment, and further a tendency towards *rotating the object* will occur.

So we consider a different method here, based on a "lasso-like" approach. The manipulator tightens its shape around the object, until finally all limbs touch the object surface at approximately the same time. This method is shown in Fig. 3, on the right.

3.2 Sensor System

The simulative approach needs an assumption for the manipulator's sensor configuration. We choose a sensor system with circular sensitivity characteristics and a finite detection range. Each limb has a distance sensor with a maximal reach of δ_{max} length-units, and returns the distance value of the object point closest to itself, located in the midpoint of the limb. We simplify this principle in choosing the distance from the sensor to the nearest object point and call it "nearest neighbour-sensing". The principle is shown in Fig. 4.

3.3 System Configuration

The system configuration is different from training-phase to playback-phase. In order to train the neural net for the grasping process, a trajectory generator (explained in Paragraph 4) produces grasping sequences of cubic spline curves $\mathbf{x}_m(s)$ for each training-object

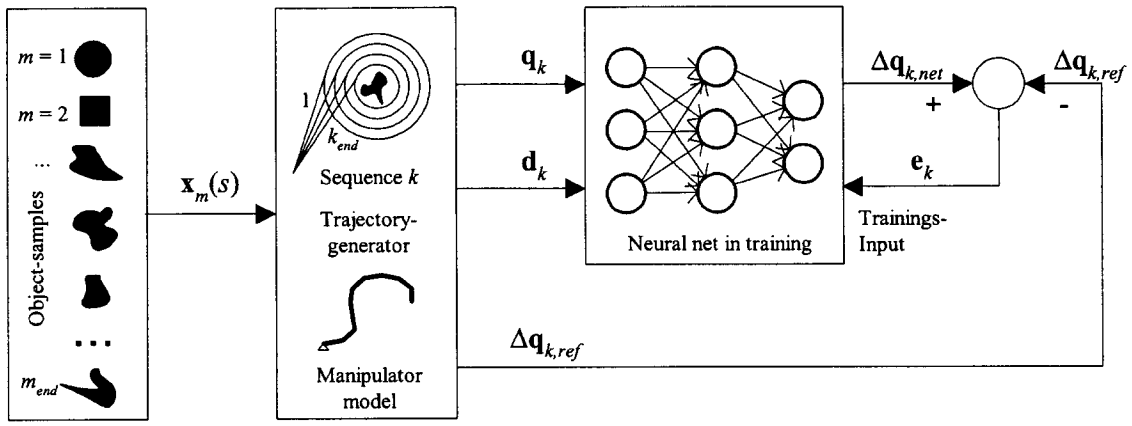


Figure 5: System in teaching-mode

ω_m out of a training-set $\Omega = \{\omega_1, \omega_2, \ldots, \omega_{m_{end}}\}$. In this context, s is the curve spline parameter. For an n-link manipulator, the output of the trajectory generator is a sequence of k_{end} joint angle vectors $\mathbf{q}_k$ of length n and a sequence of distance vectors $\mathbf{d}_k$ with the length $n-1$, for each training-object ω_m. k_{end} is the number of calculated manipulator postures. Each vector $\mathbf{d}_k$ contains distance information from the limb midpoint towards the nearest object surface point as described in Section 3.2.

The trajectory generator also determines the joint angle increments $\Delta\mathbf{q}_k = \mathbf{q}_k - \mathbf{q}_{k-1}$ of two subsequent grasping stages with the constraint $\mathbf{q}_{-1} = \mathbf{q}_0$. For a constant sampling time, the signal $\Delta\mathbf{q}_k$ is an approximative measure for the joint speed $\dot{\mathbf{q}}$ in the sequence k.

Both vectors $\mathbf{q}_k$ and $\mathbf{d}_k$ form the input of a neural net with time delays. The net is trained to react with a proper $\Delta\mathbf{q}_k$ at its output. The training is repeated with the members of the object sample set, until the mean square of the error signal $\mathbf{e}$ falls beyond a given limit ϵ. The system configuration in teaching-mode is depicted in Fig. 5.

In playback mode, we assume that the net is trained to a sufficient level, so that $\|\Delta\mathbf{q}_{k,net} - \Delta\mathbf{q}_{k,ref}\| \leq \epsilon$ for all k over all objects ω_m. Then, we can define the network to be able to reproduce grasping sequences for all objects in Ω in their nominal positions, for a certain initial manipulator configuration $\mathbf{q}_0$. Fig. 6 shows the system in playback-mode. We should note here that the playback-system can either drive the manipulator axes directly (connection b), or act as neural trajectory planner (connections a), producing a sequence of command signals for an executive level controller. In case (b), the net output $\Delta\mathbf{q}_{k,net}$ is amplified and fed forward to the axis drives. Neglecting drive dynamics, this signal is proportional to the driving torques τ_k. The direct controlling mode (b) requires a training with the real mechanical system connected to the analytical grasp-planner, so that the neural network can learn the system dynamics. This case imposes additional requirements on the dynamic capability of the network, and is outside the scope of the present paper. Here, we focus on the network as trajectory generator (configuration a).

When the manipulator changes its shape and posture, the corresponding sensor signals and joint values are fed back to the neural net, which in turn produces an output vector $\Delta\mathbf{q}_{k,net}$.

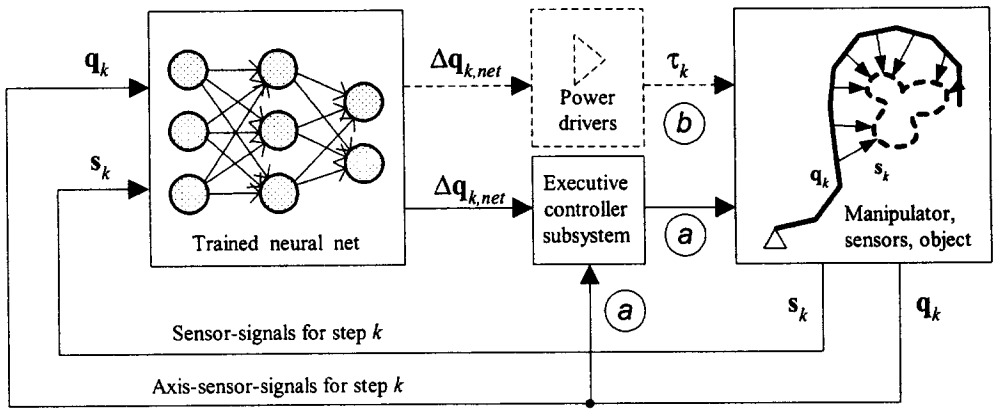


Figure 6: System in playback-mode

4 The Geometrical Grasp Planner

This section explains the principle of the trajectory generator. Given the boundary contour of an object ω_m in spline representation according to Eq.(6), we need the following steps in order to produce a sequence of grasping configurations:

1. Grow the boundary contour in discrete steps
2. Smooth the generated contours to avoid problems in concave object zones
3. Find tangents to each of the smoothed contours
4. Map the contours to a continuous manipulator
5. Find a discrete manipulator by solving the shape-inverse problem with given geometrical constraints
6. If necessary, interpolate between the contours to find additional grasping stages

Boundary growing (1.) and boundary smoothing (2.) is subsumed to the process of *boundary sequencing* and is described in the following section.

4.1 Boundary Sequencing

Boundary growing (1.) is a straightforward operation of extending points from the spline contour along a vector perpendicular to the tangent. The resulting point set is then again connected by a spline. The sequence of grown boundary contours can be either spaced equidistantly, or—which is a more favorable alternative for a smooth grasping motion—spaced with increasing contour-density in direction to the object surface. For a feasible grasping sequence, the order of grown boundary contours needs to be reversed. As a consequence, the grasping process will decelerate in close vicinity of the object, allowing a finer "grasp-adjustment".

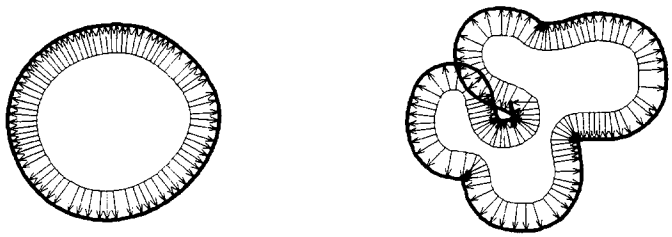


Figure 7: Boundary growing of a convex object and an object with concave zones

Concave zones in the object contour can cause problems in boundary growing due to overlapping contour segments (Fig. 7, right), which can be eliminated by means of a subsequent smoothing process (2.), regardless of the number of sample points. The boundary contour is given as vector function $\mathbf{p}(s) = (p_x(s), p_y(s))^T$, where s is the curve parameter, measured in counterclockwise direction. The smoothing is done by convolving $\mathbf{p}(s)$ with a Gaussian kernel $g(s, \sigma)$,

$$\tilde{p}_x(s,\sigma) = p_x(s) * g(s,\sigma), \tag{7}$$

$$\tilde{p}_y(s,\sigma) = p_y(s) * g(s,\sigma), \tag{8}$$

where σ is the "smoothing parameter". The filter kernel is

$$g(s,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{\frac{-s^2}{2\sigma^2}}. \tag{9}$$

The convolution is defined as

$$\tilde{f}(s,\sigma) = f(s) * g(s,\sigma) = \int_{-\infty}^{\infty} f(\nu)\, \frac{1}{\sigma\sqrt{2\pi}}\, e^{\frac{-(s-\nu)^2}{2\sigma^2}} d\nu. \tag{10}$$

Here we use the discrete approximation

$$\tilde{f}(n,\sigma) \approx \sum_{k=-3\sigma}^{3\sigma} l_\lambda f(k)\, \frac{1}{\sigma\sqrt{2\pi}}\, e^{\frac{-(n-k)^2}{2\sigma^2}}, \tag{11}$$

where n is the index of a point on the curve with the sector-length l_λ, and k runs through integer values in the range $[-3\sigma, 3\sigma]$. Other authors use Gaussian kernel smoothing for a shape representation technique called *Curvature-Scale-Space*, which is discussed in *Dudek, G., and Tsotsos J. K., 1997* and *Kindratenko, V., 1997*, mainly for the purpose of object recognition.

Up to this stage, we have generated a sequence $\Xi = [\xi_1, \xi_2, \ldots, \xi_{k,end}]$ of closed object contours, which converges towards the original object boundary. In order to match each contour to a corresponding shape of a planar manipulator, a continuous backbone curve with given finite length is to be found, with a connection from the manipulator base to the contours ξ_k. The related problems are finding a tangent to each contour (3.) and mapping the contours to a continuous manipulator (4.). Both tasks are subsumed in the term "finding a backbone curve".

4.2 Finding a Backbone Curve

Tangents $\mathbf{t}_k = (t_x, t_y)$ from the origin (manipulator basepoint) to the twice differentiable closed curve ξ: $\mathbf{p}(s) = (p_x(s), p_y(s), 0)^T$ can be found by means of the first derivative $\mathbf{p}'(s) = \frac{d}{ds}(p_x(s), p_y(s), 0)^T$, with the additional condition that $\mathbf{p}(s)$ and $\mathbf{p}'(s)$ are collinear, i. e.

$$\mathbf{p}(s) \times \mathbf{p}'(s) = \mathbf{0} \qquad \text{and} \qquad \kappa(s) \geq 0. \tag{12}$$

With positive counterclockwise variation of s, the curvature κ needs to be positive, since no tangents can be applied to concave sections of the curve. Further, tangents colliding with parts of the objects are to be eliminated. This is a process with a high computational effort. The choice of the tangent solution is dependent on q_1, in a sense that the tangent closest to this direction is to be chosen. The zeros of the left part of Eq.(12) are determined by conventional iterative algorithms.

Now, a finite length spline is mapped to the tangent section $\mathbf{t}_k$ through the origin up to the tangent point $(t_{x,k}, t_{y,k})$ and the corresponding contour ξ_k. This is accomplished with elementary spline operation functions.

4.3 Shape-Inverse Problem and Interpolation

Solving the shape-inverse problem (5.) means to map a discrete n-link planar manipulator with specified limb-lengths l_i to the finite length backbone-curve. After solving the reverse kinematics, this results in the vector of joint angles $\mathbf{q}_i$. Solutions to this problem are discussed by several authors, for example in *Chirikjian, G. S., 1992* and in *Hayashi, A., 1994*. A straightforward method is discussed in *Mochiyama, H., Shimemura, E., and Kobayashi H., 1999*, which searches for intersections of the backbone curve with a circle of radius l_i (length of i-th link). The objective is to minimize the absolute value of the distance of the actual link's joint position at s_i to the calculated joint position of the previous link at s^*_{i-1} (s_i is the parameter variable of the backbone curve)

$$\left| s_i - s^*_{i-1} \right| \rightarrow \min, \tag{13}$$

under the constraints

$$s^*_{i-1} - s_i < 0, \tag{14}$$

$$\left\| \mathbf{c}(s_i) - \mathbf{c}(s^*_{i-1}) \right\| - l_i = 0, \tag{15}$$

where $\mathbf{c}(s_i)$ and $\mathbf{c}(s^*_{i-1})$ are the cartesian position vectors for the parameters s_i and s^*_{i-1}, respectively. $\|\cdot\|$ is the euclidian vector norm. This method has a significant drawback for whole-arm-grasping, since the solution generates limbs, which intersect with the object contour.

We therefore use an alternative approach, based on a principle, that either a link lies on the backbone-curve or a neighbouring limb is a tangent to it. This method differs from the previous approaches in the fact, that not all joints are elements of the backbone-curve. There is no restriction of functionality in the case of whole-arm-grasping, since the object can be grasped by either a link or a limb.

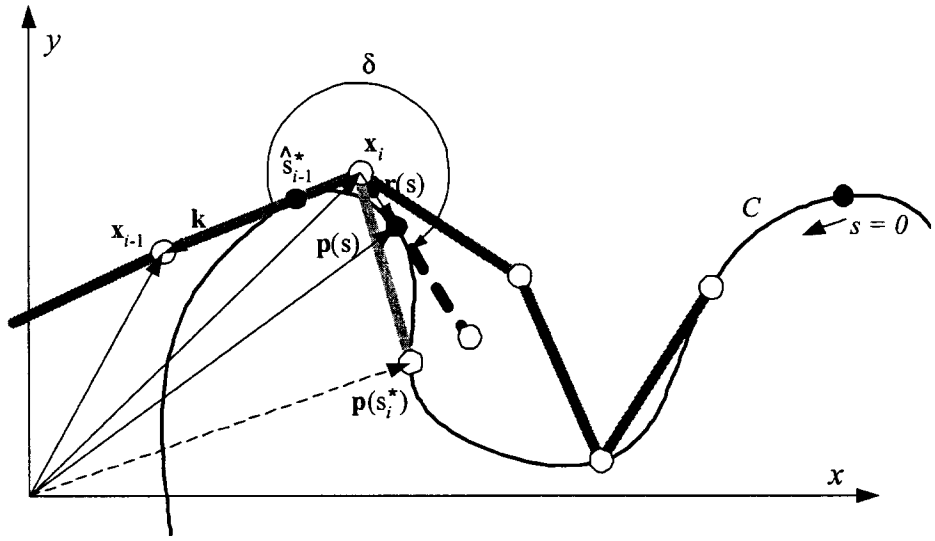


Figure 8: Alternative solution of the shape-inverse problem

The principle can be explained according to Fig. 8. When the curve-parameter s changes, the point $\mathbf{p}(s)$ moves along the curve, say, from $s = \hat{s}^*_{i-1}$ to $s = s^*_i$. The index $s = \hat{s}^*_{i-1}$ indicates the tangent point of the limb $[\mathbf{x}_{i-1}, \mathbf{x}_i]$, which is already an existing part of the shape-inverse solution. The underlying algorithm searches for a parameter s so that $\delta = \angle(\mathbf{k}, \mathbf{r}(s)) \rightarrow \min$. At this minimum, the next tangent point $p(s)$ is found. The direction of movement on the curve C is directly related to the sign of δ. This principle also uses the standard shape-inverse method Eq.(13) to Eq.(15) in order to find search-boundaries for the index s.

Now, for each object contour of the sequence $\Xi = [\xi_1, \xi_2, \ldots, \xi_{k,end}]$, a discrete planar manipulator configuration $\mathbf{q}_k$ has been determined. It may be desirable to refine the found sequence by interpolation (6.). Let us recall here that the trajectory generator works off-line in order to produce teaching-data for the neural net. Despite of this, it may make sense to use interpolation for further refinement due to calculation-time optimization.

In this paragraph, we have discussed the working principle of the trajectory generator for the production of teaching data. In the following, we summarize the properties of the neural net building blocks.

5 Neural Net Configurations

According to Fig. 6, the neural net is inside a feedback loop between the sensor signal vector $\mathbf{s}_k$ and the manipulator system. The net itself is trained to react with a proper signal vector $\Delta\mathbf{q}_k$ to sensor- and joint angle inputs $\mathbf{q}_k$, as specified by the grasp-planner during training. Let us repeat here that we assume the existence of an executive controller subsystem, which is responsible for the dynamic axis control of the manipulator system. The neural net acts as a trajectory generator, producing a sequence of reference signals.

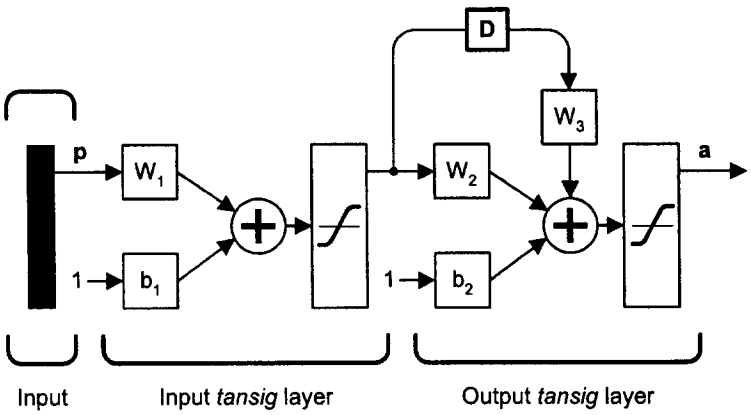


Figure 9: Dynamic feedforward-net ($\mathcal{D}$-net)

In this function, the net *recalls* a learned trajectory, a process which has already been investigated in *Araújo, A. F. R., and Barreto, G. de A., 1999 and 2000*.

5.1 Feedforward-Net with Time Delays

A straightforward approach is to use a dynamic feedforward net with time delays in the second layer as shown in Fig. 9 (we call it in short $\mathcal{D}$-network because of the time delays). The motivation of this choice is the fact, that object-grasping can be described—in a simplified view—as incrementally changing the curvature of the manipulator shape in dependence of the object distance. The curvature κ of a spatial curve $\mathbf{x}(s) = (x(s), y(s), z(s))^T$ with the unit tangent vector $\mathbf{t}_0 = \frac{\dot{x}}{\|\dot{x}\|}$ is defined as

$$\kappa = \left\| \frac{d\mathbf{t}_0}{ds} \right\|, \tag{16}$$

$\|\cdot\|$ is the euclidian vector norm. For the planar case, the curvature is $\kappa = \frac{1}{l}\frac{\partial\theta}{\partial s}$ (according to Eq.(2) and Eq.(3)), where θ is the slope of the tangent vector $\mathbf{t}_0$. From this fact we can see that the local curvature of a discrete manipulator's backbone can be approximated as being proportional to the difference of joint angles, i. e. $\kappa_{d,k}(i) = \frac{1}{l_i}(q_{i+1,k} - q_{i,k})$. The input layer of the $\mathcal{D}$-net is supposed to learn this function, which approximates the differentiation of $\mathbf{t}_0(s)$ on the backbone curve with respect to s.

Fig. 6 shows, that besides $\mathbf{q}_k$, the sensor signal vector $\mathbf{s}_k$ is fed into the fist layer of the network during playback-mode. The grasping-sequence produced by the trajectory generator changes the curvature in dependence of the local sensor distance $\mathbf{s}_k$. The $\mathcal{D}$-net is also supposed to learn this (nonlinear) relationship during training, supported by the *tansig* activation-function, in order to generate a new $\mathbf{q}_{k+1}$ corresponding to the training data set. The second layer has a time delay path, which is required to learn to calculate the joint increments $\Delta\mathbf{q}_k$.

Summarizing, this network structure has been chosen due to its capability of learning nonlinear spatial and temporal relationships between input and output signals. The simulations in Paragraph 6 show, that this configuration performs well for the given task.

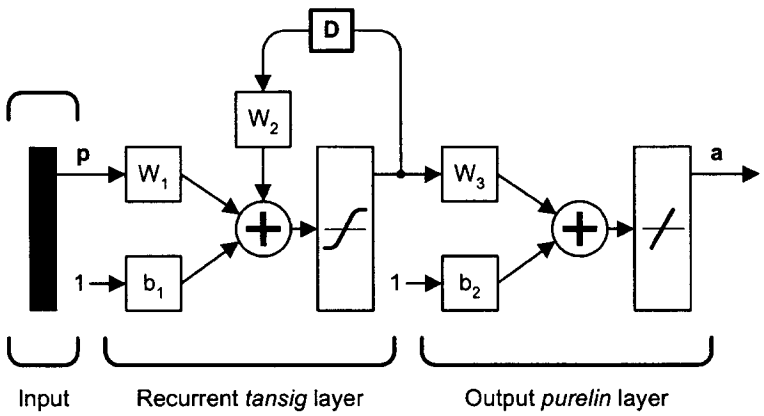


Figure 10: Elman-net

5.2 Elman-Net

The Elman-network, introduced in *Elman, J. L., 1990*, is a recurrent two-layer network with feedback from the first-layer output to the first layer input. This recurrent connection allows the network to both detect and generate time-varying patterns. These properties make the network-type suitable for the present task. The Elman-network stores the "history" of time-varying patterns up to the depth of the delay line. We use standard back-propagation for the calculation of the error gradients of weights. The simulation results are discussed in the following.

6 Simulation and Discussion

To generate a training set for both the feedforward ($\mathcal{D}$-) and Elman-network, the analytical grasp planner performs a whole-arm grasping sequence of a circular object according to the "Lasso" method. Although the circular object seems to be a trivial choice in comparison with real-world objects, it permits an easier comparison of basic performance characteristics. Further work may extend the scope to more complex object shapes.

The interpolated playback sequences are compared in Fig. 11. The $\mathcal{D}$-net has a significant better learning performance in the specified configuration than the Elman-net. This may be due to the fact that the feedforward-net supports the learning-process by its structure (compare Section 5.1), whereas the Elman-net needs much more training-data and more layers and neurons to learn the performance from the *time history* in recurrent mode.

We measure the performance by comparing the trajectory, produced by the geometrical grasp planner with the corresponding trajectory generated by the neural grasp planner. In order to compare the performance with a quantitative measure, we introduce a performance index by defining the total error ϵ_T as

$$\epsilon_T = \sqrt{\sum_{\mu=1}^{m} \sum_{\nu=1}^{\sigma} \left(\hat{d}_\nu^{(\mu)} - d_\nu^{(\mu)} \right)^2} \tag{17}$$

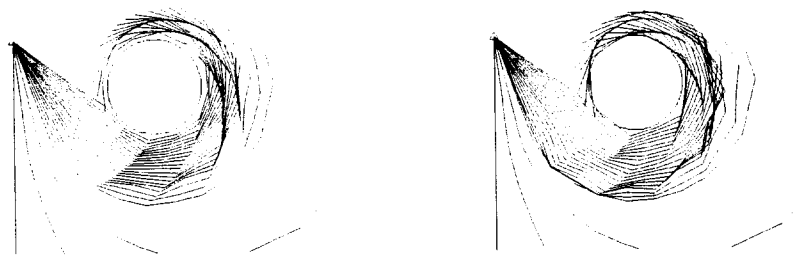


Figure 11: Results of neural "Lasso"-grasping with the feedforward $\mathcal{D}$-net (one layer, six neurons, remaining mean-square-error 10^{-8} in 200 epochs, left) and an Elman-network (three layers with three neurons each, remaining mean-square-error 10^{-4} in 4000 epochs, right) for a 16 DOF manipulator

where $\hat{d}_\nu^{(\mu)}$ is the distance of the sensor with the index ν from the object at time step μ, generated by analytical grasping, whereas $d_\nu^{(\mu)}$ is the corresponding distance value according to the neural network-based grasping sequence. Apparently, a smaller value of ϵ_T indicates a better performance. To test the ability of generalization from the learned spatial/temporal data, we train each network to grasp a circular object at an initial configuration. At playback time, the index defined in Eq.(17) is determined for different object offsets in the (x, y)-plane, where x and y are of linear spatial dimension. The results are shown in Fig. 12 and Fig. 13.

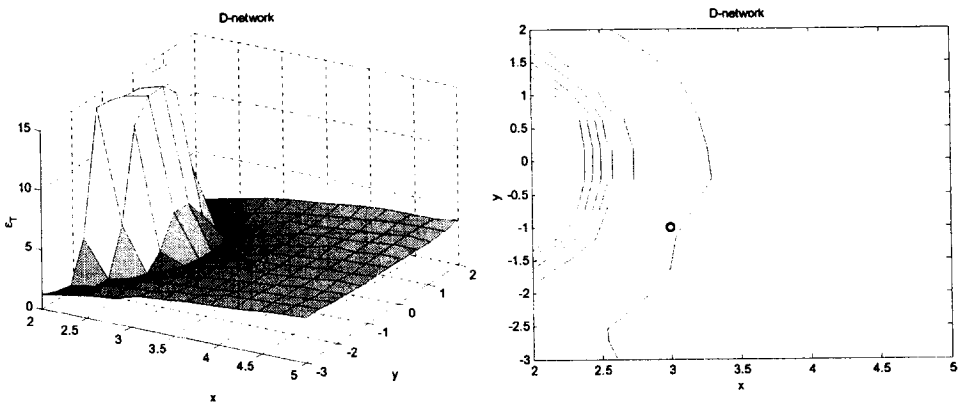


Figure 12: Overall spatial generalization performance of the feedforward-network. The diagrams show the dependence of index ϵ_T from the object-shift in the (x, y)-plane. The circle-mark in the contour plot indicates the trained position

For the circular object in the current simulation configuration, the performance of the $\mathcal{D}$-net and the Elman-net show a different behaviour, which is a particular result of the investigations. When interpreting these results, we need to see them in the context of the specific training situation and net configuration. Significantly extending the number of

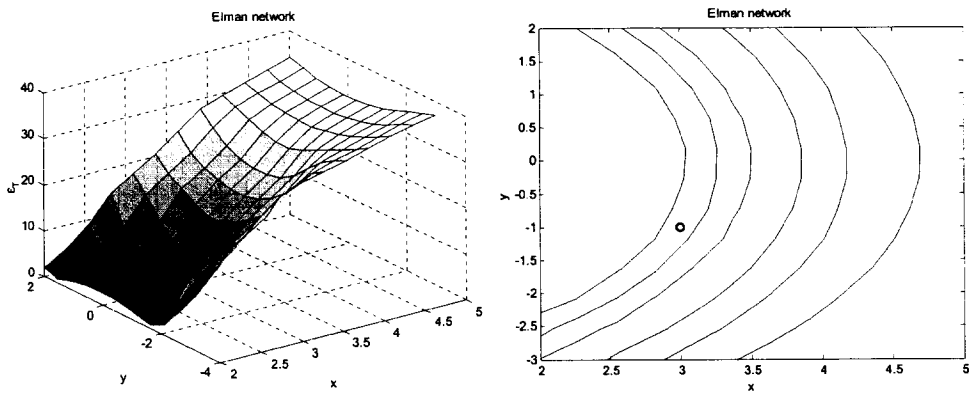


Figure 13: Overall spatial generalization performance of the Elman-network, in comparison with Fig.12

neurons in the case of the Elman-net or enlargement of the training data set may change the shown behaviour. Deeper investigations into this direction is subject to further work.

7 Conclusion

In this paper, we have discussed a new method of trajectory planning in connection with hyper-redundant planar kinematics for whole-arm grasping. A training generator provides data sets for a neural network, which in turn learns the procedural knowledge of the grasping sequence.

The proposed system is an important building-block for a whole-neural robot coordination unit. The vision is to control hyper-redundant robot arms to perform manipulation tasks including object finding, obstacle avoidance, object recognition and task planning. Neural control units have certain advantages in case of complex mechanisms with many degrees of freedom, since they eliminate the necessity of complex algorithms, and can be re-trained for new tasks. Certain drawbacks arise from the fact, that there are no closed form solutions available for the system design, and a certain effort is necessary to determine the most successful configuration, sometimes by trial-and-error. In the case of delicate human-machine interaction, neural units may be implemented e. g. in vision interfaces, tactile sensorics and coordination units. In bionics, neural trajectory controllers may play a vital part for artificial hands and other sophisticated manipulation devices.

References:

Araújo, A. F. R., and Barreto, G. de A., 1999, Unsupervised learning and recall of temporal sequences: An application to robotics. *International Journal of Neural Systems*, **9(3)**, pp. 235–242

Araújo, A. F. R., and Barreto, G. de A., 2000, A self-organizing neural network for learning and recall of complex robot trajectories. *Proceedings of the 2nd International ICSC/IEEE Symposium on Engineering of Intelligent Systems*, pp. 72–78

Chirikjian, G. S., 1992, Theory and Applications of Hyper-Redundant Robotic Manipulators. California Institute of Technology, PhD Thesis.

Dudek, G., and Tsotsos J. K., 1997, Shape representation and recognition from multiscale curvature. *Computer Vision and Image Understanding*, **68(2)**, pp. 170–189

Elman, J. L., 1990, Finding structure in time, *Cognitive Science*, **14**, pp. 179–211

Hannan, M. W., and Walker, I. D., 2001, The "Elephant Trunk" manipulator, design and implementation. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Como, Italy*

Hayashi, A., 1994, Geometric Motion Planning for Highly Redundant Manipulators Using a Continuous Model. University of Texas, 1994. PhD Thesis.

Immega, G., and Antonelli, K., 1995, The KSI Tentacle Manipulator. *1995 IEEE Conference on Robotics and Automation*, **3**, pp. 3149–3154

Kindratenko, V., 1997, Development and Application of Image Analysis Techniques for Identification and Classification of Microscopic Particles. University of Antwerp (UIA), PhD Thesis.

Lenhoff H. M., and Lenhoff S. G., 1988, How the animal nature of marine cnidarians was recognized and the nematocyst discovered. In: *The biology of nematocysts (Hessinger D. A., and Lenhoff H. M., eds.)*. Academic Press, San Diego.

Mochiyama, H., Shimemura, E., and Kobayashi H., 1999, Shape control of manipulators with hyper degrees of freedom. *The International Journal of Robotics Research*, **18(6)**, pp. 584–600

Mochiyama, H., 2001, Kinematics for the whole arm of a serial-chain manipulator. *Advanced Robotics* **15**, pp. 255–275

Reznik, D., and Lumelsky, V., 1998, Sensor-based motion planning for highly redundant kinematic structures. **ii,** the case of a snake arm manipulator. *IEEE International Conference on Robotics and Automation*

Williams, R. L., and Mayhew, J. B., 1997, Obstacle-free control of the hyperredundant nasa inspection manipulator. *Proceedings of the Fifth National Conference on Applied Mechanisms and Robotics*

Žlajpah, L., 1996, Dynamic modelling of *n-R* planar manipulators. *26th International Symposium on Industrial Robots, Singapore*, pp. 339–343

Žlajpah, L., 1998, Simulation of *n-R* planar manipulators. *Simulation Practice and Theory*, **3(6)**, pp. 305–321

Sales Process for Product and Service Management for Machine Manufacturing Small and Medium Sized Enterprises

Dipl.-Ing. Stephan Häberle (Automation and Control Institute, Vienna University of Technology, Haeberle@ACIN.tuwien.ac.at),

responsible for the Austrian Cluster within the project SALESMAN

(http://www.eu-salesman.com)

Abstract

Individualization of products and services is a challenge for sales representatives. More and more, options and variants are available to the costumer. Product configuration systems (PCS) assure that only those variants are offered, which are feasible. In the course of the EU-Project "Salesman", the sales process of five companies was analysed and restructured to provide product and service configuration opportunities for sales representatives and costumers.

1. Introduction

A study of the "European Logistics Association" considers that the future customer needs are [1]:

- Custom-made products
- A larger variety and selection of products to fit individual styles
- A stronger, more individually adapted customer-care service
- Shorter response and delivery times.

Furthermore, customers expect greater flexibility as well as better care and attention from the moment an order is placed right until the product has been delivered. Customer's wishes for modifications should also be taken into consideration whilst the product is being made, and the current status of the following of the product should be transparent at all times [2].

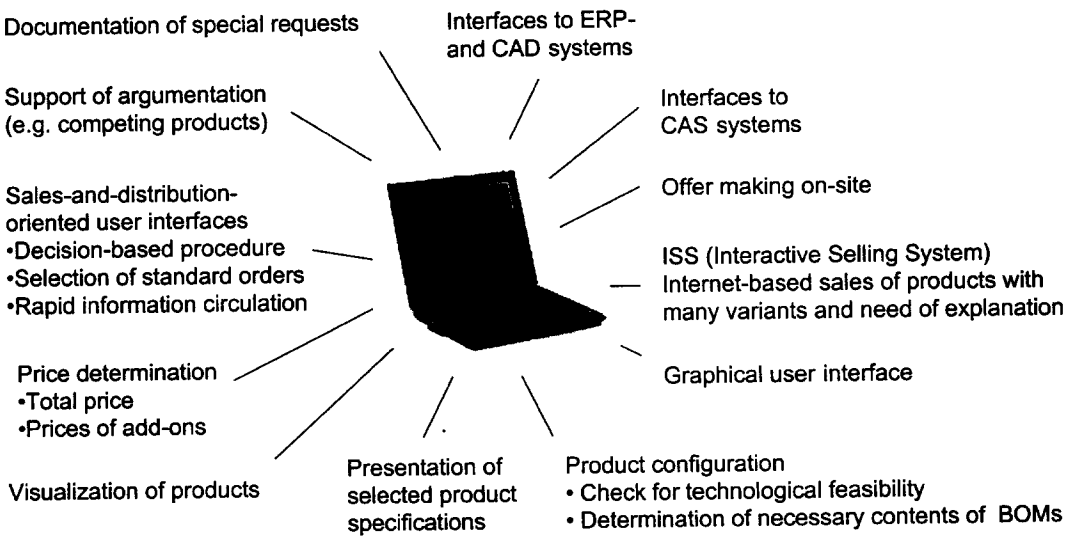


figure 1 - functionalities of a product configuration system [3]

Conventional technologies like product catalogues, price or check lists are not flexible and complex enough to describe all the product variants in a clear and easy-to-handle way. To enable to capture the sales relevant product knowledge, necessary to configure products quickly according to product rules, product configuration systems (PCSs) were developed, which combine capabilities of storing bill of materials (BOMs) and checking configured products against product rules with the aid of expert system technologies (Information about PCS on the Internet: [4]). In a market survey carried out by the Fraunhofer "Institut Arbeitwirtschaft und Organisation" [5] 26 suppliers for PCS could be found in the German speaking area as well partly in the US. Beside the key functionality of checking configuration rules, PCSs offer a wide range of other functionalities depending on the supplier (figure 1).

The most important additional features are the calculation of prices, product presentations and the integration with other systems like ERP (enterprise resource planning) systems to avoid data redundancies.

Due to these functionality, PCS have the potential to shorten and simplify the sales cycle (figure 2) by

- ensuring that offers are feasible,
- enabling costumers to configure products (self-configuration),
- reducing the amount of cycles to clarify the costumer's needs (check backs) and
- improving the information flow from engineering to sales to the costumer.

But a PCS is only an enabler for these improvements. The use of a PCS in the sales process requires the reengineering of the whole process.

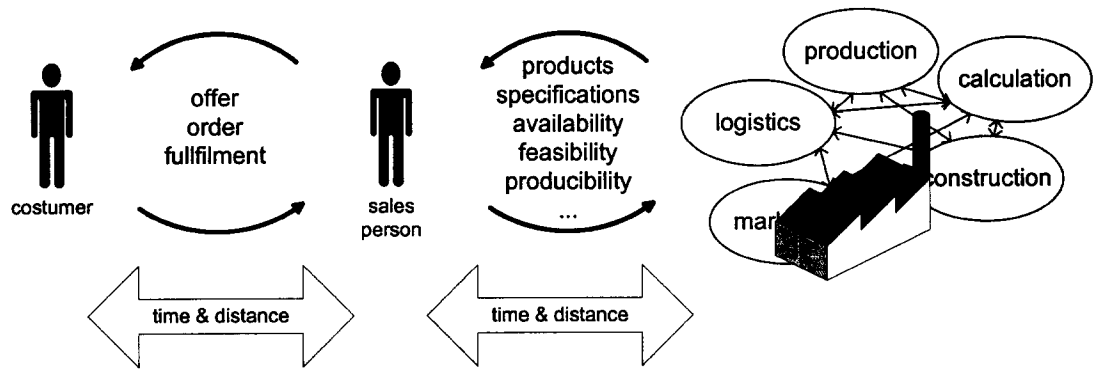


figure 2 information flow of sales

2. The project SALESMAN – introducing a PCS

To enable these functionalities for SMEs (small and medium sized enterprises) SALESMAN, a best practise project sponsored by the European Commission, was established. The project, which started in January 2001 and will be finished by October 2002, has an overall project amount of 1.2 Mio Euro.

The goal of the project is to use PCSs as an existing technology and to find and document an appropriate way of selecting and implementing a PCS.

To ensure the industrial relevance the consortium consists of five industrial partners from Norway (Moxy Trucks AS [6]), Spain (Optimus S.A. [7]), Austria (Neuson-Kramer Baumaschinen GmbH [8], Anger GmbH [9]) and Germany (Pfister GmbH

[10]). The common problems of these companies are products with a high amount of variants, which have to be configured in the right manner. The five industrial partners are supported by three research institutes: Profactor Produktionsforschungs GmbH [11] (research institute in Steyr, Austria), ACIN – Automation & Control Institute [12] (Vienna University of Technology) and IAT - Institut Arbeitswissenschaft und Technologiemanagement [13] (University of Stuttgart).

In addition to these eight partners, the project has established an industrial support group (ISG) where all interested companies can join (via the projects homepage: www.eu-salesman.com) free of charge and obligations to benefit from the results of the project and to give feedback.

The essential first step was to capture the requirements of each industrial partner. This was done by workshops at each industrial partner supported by the research institutes. These workshops were performed with the aid of a questionnaire developed in the course of the project. The questionnaire covered the most important points for the selection of a PCS to identify necessary process adaptations.

The requirements differed by the product and rule complexity, the users (sales representatives, costumer, Intra or Internet access) and by the process steps to be supported. Common requirement of all partners was a redundancy free system. Particularly for self-made legacy systems this requires a high flexibility of a PCS. Due to these reasons the selection process was much harder than expected. Although there are many suppliers for PCS, four of the industrial partners did not find a PCS fitting to their requirement – WHY?

There are two main reasons:

- First of all, the industrial partners felt the prices charged for a PCS too high for an SME. Of course prices vary substantial, but an average of 50.000 Euro for the software seems to be quite common - customization resulting in additional costs. This problem gets even worse, if the users only needs a fraction of the functionality, because the other features of the PCS are covered by other systems or not necessary.
- The second main reason, which made it hard for a SME, is the integration into legacy systems. On the one hand SME often have self-made systems, where it is – of course – not possible to get a standard interface. In these cases there are often only two options left: Continue to implement necessary functionalities into the legacy systems or replace the self-made solution by a standard-software. The decision depends on future developments and the extent of missing functionality. On the other hand PCS on the market often do not support ERP systems of small SMEs.

Beside these technical and financial problems the selection process was also hard because of the communication with the PCS supplier. The impression of one of the industrial partners is as follows:

> *"We contacted twelve software suppliers, which seemed to be capable to provide a PCS solution for us. Five of them did not even response, a very big supplier referred to the companies homepage instead of giving us information directly - but we did not find the information we needed. One supplier did not have a PCS solution.*
>
> *From the remaining five companies only three were capable to send us a demo CD or other product presentations so that we were able to get a closer*

look. We invited two of them, who offered a Web interface, to show us their PCS at our site. The supplier of the product, which we thought that would fit best, ignored our requirements completely and showed us a standard presentation. It seemed that we are too small to be a prospect for him?!"

As a result, four of the industrial partners within the consortium faced the alternative of have a tailor-made system programmed or of developing an own system based on (self-made) legacy systems.

At the end, two industrial partners decided to implement the configuration functionality needed by their own staff, two others started to look for a software-supplier able to implement a solution for them.

From the later two, one gave this task to the supplier of the ERP system they introduced at the same time. The other found a software supplier who was able to provide them a Web based solution for a PCS at a price of 10-15.000 Euro, which is much cheaper than the best offer they had from a PCS supplier. (It has to be stated, that the industrial partner in question has products with a low product complexity.)

3. Conclusion

There are many suppliers of product configuration systems on the market. The result of the EU-project SALESMAN showed, that in spite of the fact of this variety of suppliers, it is not easy for a SME to find an appropriate PCS. On the one hand prices charged for a PCS are too high to be affordable for a SME, on the other hand the integration with other systems of a SME requires substantial additional costs.

As a result of this project four of five companies decided in favour of a tailor-made or self-made solution of a configuration system instead of contracting with a PCS supplier.

Acknowledgement

This work has been partly funded by the European Commission through IST Project SALESMAN: Sales process for product- and service management for machine manufacturing SMEs (Proposal-No. IST-1999-20134).

The author wishes to acknowledge the Commission for their support and all consortium members for their input in course of the project.

References

[1] A. T. Kearney, *Insight the impact: results of the fourth quinquennial European logistics study*: ELA European Logistics Association, 1999.

[2] R. Bürscher and H. Huemer, "Existing eBusiness Technology for Manufacturing Companies," presented at E-work and E-commerce, 2001.

[3] R.-D. Kempis and J. Ringbeck, *Nutzen von Informationstechnologie in der produzierenden Industrie*: PTW/McKinsey Internationale Studie, 1998.

[4] M. Liebermann, *Get a grip on configurators*, http://www.pdmic.com/articles/artgetg1.html

[5] J. Frech, J. Schwarz, and F. Wagner, *Computer Aided Selling Systeme zur Produktkonfiguration*. Stuttgart: Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO, 2000.

[6] Moxy, *MOXY Trucks AS*, www.moxy.no

[7] Optimus, *Optimus*, www.optimus.es

[8] Neuson, *Neuson-Kramer Baumaschinen*, www.neuson.com

[9] Anger, *Anton Anger GmbH*, www.anger.at

[10] Pfister, *Pfister GmbH*, www.pfister.de

[11] Profactor, *Profactor GmbH*, www.profactor.at

[12] ACIN, *Automation Control Institute*, www.acin.tuwien.ac.at

[13] IAT, *Institut Arbeitswissenschaft und Technologiemanagement*, www.iao.fhg.de

American Control Conference

Anchorage, Alaska

8. - 10. Mai 2002

Vom 8. – 10. Mai 2002 wurde das Kernprogramm dieser Konferenz abgewickelt, am 6. und 7. Mai fanden zusätzlich ganztägige Workshops statt. Das Programm setzte sich aus je einem Plenarvortrag, gefolgt von je 17 Parallel Sessions zusammen. Insgesamt ergaben sich etwa 1000 Präsentationen aus dem Gebiet der Automatisierungs- und Regelungstechnik. Auf den Einsatz von Poster Sessions – diese Variante wurde auf der ACC'01 erstmals realisiert – wurde heuer verzichtet. So wie auch bei früheren ACCs wurde im Rahmen des Kongresses in einer Ausstellungshalle von diversen Firmen aktuelles Laborequipment, wie zum Beispiel regelungstechnische Experimente und Roboter sowie neueste Software vorgeführt.

Unser Arbeitsgebiet, die Fehlerdiagnose, war heuer schon wesentlich stärker repräsentiert als in den vergangenen Jahren. Diesem Thema war ein eigener Track gewidmet, aber auch in anderen Sessions fand man häufig Papers, die Problemstellungen der Fehlerdiagnose behandelten. Die zunehmende Bedeutung der Fehlerdiagnose spiegelte sich auch in den Workshops wieder, wo Themen wie „Process & Control System Performance Monitoring and Trend Analysis" oder „Event-Based Monitoring and Diagnosis of Industrial Systems" angeboten wurden.

Die Themen der Plenary Sessions deuten darauf hin, dass zunehmendes Interesse daran besteht, die zweifellos hoch entwickelten Methoden der Regelungstechnik so zu modifizieren, dass sie auch bei praktischen Problemen implementiert werden. Ein besonders interessanter Ansatz wurde im Vortrag von Prof. Roger Brockett angesprochen, der mit „Minimal Attention Control" eine Methode vorschlug, bei welcher der Aufwand der Implementierung in ein Optimierungskriterium mit eingeht. Der resultierende Algorithmus kann einerseits die vorhandenen (komplexen) Entwurfsmethoden nützen, andererseits sollte die Implementierung vergleichsweise einfach sein.
Da wir einen Beitrag zur nichtlinearen Systemidentifikation präsentierten, waren die entsprechenden Sessions für uns von besonderer Bedeutung. Dabei sind besonders die Beiträge von Prof. Annaswamy und Prof. Narendra zu nennen, welche beide einen Ansatz für einen generalisierten Zugang zur nichtlinearen Parameterschätzung präsentierten. Diese theoretischen Überlegungen versuchen weg von spezialisierten Stabilitäts- und Konvergenzbeweisen zu kommen und nach Möglichkeit universelle nichtlineare Kriterien für diese wichtigen Bereiche zu finden. Dabei sollte einerseits nicht so allgemein wie z.B. bei der Lyapunov-Stabilität vorgegangen werden, andererseits sollen viele wichtige nichtlineare Systemstrukturen erfasst werden.

Unser eigenes Paper behandelte die automatische Diagnose von Messfehlern an einem Motorenprüfstand. Dabei kommen neuronale Netze mit elliptischen Basisfunktionen zum Einsatz, für die ein neuer, leistungsfähiger Lernalgorithmus entwickelt wurde. Der Einsatz von genetischen

Algorithmen zur Lösung von komplexen, hochdimensionalen Problemstellungen wurde anhand einer medizintechnischen Anwendung präsentiert. Es handelte sich dabei um die Bildrekonstruktion in der Computertomographie, die typischerweise durch einen modifizierten Newton-Raphson Algorithmus realisiert wird. Der vorgestellte genetische Algorithmus konnte bei gleichen Rohdaten eine deutlich bessere Bildrekonstruktion bei jedoch höherem Rechenaufwand liefern.

Ein weiteres interessantes Paper, welches das Arbeitsgebiet der Autoren abdeckt, wurde in der Session „nonlinear identification" präsentiert: Eine Variante der nichtlinearen Identifikation ist die stückweise lineare Identifikation von Teilmodellen und deren nichtlineare Kombination. Diese Methode hat den Vorteil, dass man sich einerseits bei der Parameteridentifikation bewährter linearer Methoden bedienen kann, jedoch andererseits ein voll nichtlineares Gesamtmodell erhält. Ein aktuelles und grundlegendes Problem dieser Methode, nämlich eine optimale Aufteilung des Parameterraumes auf lineare Teilmodelle, bzw. die Bestimmung einer optimalen Anzahl an Teilmodellen wurde in dieser Session durch einen interessanten Ansatz behandelt.

Die anderen Präsentationen zur nichtlinearen Identifikation waren entweder sehr anwendungsspezifisch und daher auch sehr unterschiedlich, oder wie unsere eigene eine Mischung: Einerseits ist die Struktur der Nichtlinearität ohne all zu große Einschränkung vorgegeben, andererseits werden die Parameter dieses Modells ganz allgemein geschätzt. Unserer Meinung nach sind diese Verfahren zwar nur für bestimmte Klassen von Systemen anwendbar, in der Praxis haben sich aber bereits einige solcher Klassen etabliert und ermöglichen dem Anwender sein Problem rasch einzuordnen und in der Folge den passenden Algorithmus zu finden.

Ein anderer Interessen-Schwerpunkt war für uns die Regelung und Identifikation von chemischen Prozessen, wobei der Schwerpunkt auf Identifikation zum Zweck von prädiktiver Regelung lag. Dabei wurde ein großer Teil der Vorträge sowohl zum Thema Batch-Prozesse über Anwendungen der Halbleiter-Herstellung als auch zur chemischen Verfahrenstechnik gehalten. Diese, und die speziellen Vorträge zur modellbasierten prädiktiven Regelung (MPC) ließen erkennen, dass einerseits MPC ein Standard in der fortschrittlichen Prozessregelung geworden ist, andererseits existieren sowohl noch theoretische Fragen zur globalen Stabilität als auch praktische Probleme der Anwenderfreundlichkeit bei Entwurf, Tuning und Implementierung.

So wie auch in den letzten Jahren konnte man auf der ACC'02 nicht nur von den wissenschaftlichen Vorträgen profitieren, sondern auch von den zahlreichen Fachgesprächen mit Kollegen aus der ganzen Welt und es wurden auch neue Kontakte geknüpft.

Stefan Jakubek
Martin Kozek

1. Bericht vom 15. IFAC World Congress

Barcelona, Spanien
21. – 28. Juli 2002

Der alle drei Jahre stattfindende IFAC Weltkongress entwickelt sich immer mehr zu einem Monsterereignis. So vereinte dieser Kongress 2011 Teilnehmer aus 62 Ländern, wobei die regionale Verteilung von Spanien angeführt wurde, gefolgt von den USA. Es wurden 7 Plenary Papers vor 800 - 1000 Zuhörern, sowie 240 Technical Sessions, 12 Poster Sessions und 9 Panel Discussions gehalten. Zur Organisation waren mehr als 100 Mann im Einsatz. Eine Novität stellten sechs sogenannte Milestone Sessions dar, welche versuchten den State of the Art in bestimmten Fachgebieten umfassend darzustellen. Die meisten dieser Milestone Sessions resultierten aus den Professional Briefs, welche derzeit noch kostenlos im Internet abgerufen werden können.

Ausführliche Berichte der österreichischen Teilnehmer folgen im nächsten Heft.

P. Kopacek

ROBUST MODAL CONTROL with a Toolbox for Use with MATLAB

Jean-Francois Magni

Kluwer Academic Publishers,
New York 2002, Hardbound
ISBN 0-306-46773-9/EUR 149,50

Das vorliegende Buch und die begleitende auf CD beigelegte Toolbox ist der Eigenvektor-Vorgabe auf modaler Basis gewidmet. Schwerpunkt liegt auf Multi-Modell-Regelgesetzen und der Robustheit unter strukturierten Parameterunsicherheiten. Auch dem Einschluss von Beobachtern, der Pol-Nullstellen-Kürzung und den Mehrstufen-Entwürfen wendet sich das Werk zu.

Kapitel 1 ist als Tutorial der modalen Regelung gewidmet, insbesondere der Eigenstruktur-Zuweisung; darunter in einem ersten Teil vielen herkömmlichen Techniken, im zweiten Teil Multi-Modell-Verfahren, in einem dritten dem Feedback-Gain-Tuning.

Im Kapitel 2 werden die Verfahren auf den Entwurf ausgerichtet, u.a. Eigenwertempfindlichkeit des geschlossenen Regelkreises, Entkopplungen, separate Eigenvektor-Methoden für niedrige und hohe Frequenzen.

Im Kapitel 3 werden die begleitenden Toolboxes im einzelnen genau beschrieben.

Als Kapitel 4 rangiert eine Reihe von Beweisen zum Kapitel 1. In einem letzten Kapitel 5 ist Demonstrationsmaterial vorgelegt, insbesondere eine Flugzeugregelung in 37 verschiedenen linearisierten Modellen, enthalten auf der beigelegten CD-ROM.

Das Buch ist sowohl für Studenten als auch für Forscher und Industrieanwender ein wertvolles Nachschlagewerk und eine ergiebige Programm-Quelle, und in jeder Hinsicht sehr empfehlenswert.

A.Weinmann

Industrieroboter

Wolfgang Weber

Fachbuchverlag Leipzig, 2002
220 Seiten, 141 Bildern und 33 Übungsaufgaben
Paperback mit CD-ROM
ISBN 3-446-21604-9

Industrierobotertechnik hat bereits eine sehr große wirtschaftliche Bedeutung erlangt und ist auch Ausgangspunkt für neue Anwendungen, z.B. in der Medizintechnik und im Servicebereich. Industrieroboter – ursprünglich hauptsächlich für die klassischen Aufgaben Schweißen, Kleben und Lackieren eingesetzt - müssen heute in der Lage sein mit anderen Automatisierungseinrichtungen zusammenzuarbeiten und mit Leitsystemen zu kommunizieren; sie sind so zu einem wichtigen Rationalisierungsinstrument der Automatisierungstechnik geworden. Für Ingenieure der Automatisierungstechnik sind daher Grundkenntnisse der in Robotik unbedingt notwendig. Der Schwerpunkt dieses Buches beschäftigt sich mit den Methoden der Kinematik, Dynamik und Regelung, die es ermöglichen eine funktionsfähige Steuerung zu entwickeln und effizient einzusetzen. Dabei stehen Lagebeschreibung, Bewegungssteuerung, Programmierung, Beschreibung der Dynamik und Bewegungsregelung im Vordergrund. Kenntnisse der Programmierung und Bewegungsbeschreibung sind notwendig, um sich in Gebiete wie Sensorik, Bildverareitung, fortgeschrittene Methoden der Programmierung, kooperative Roboter, Kollisionsvermeidung, künstliche Intelligenz und autonomes Verhalten einzuarbeiten.

Die beiliegende CD-ROM enthält Lösungsbeispiele und Programme und das in Matlab geschriebene Entwicklungs- und Visualisierungswerkzeug RoCSy (Robot Control System) für einen marktgängigen Industrieroboter. Die Simulation und grafische Darstellung des Regelungsverhaltes bei konventionellen und fortgeschrittenen Regelungsmethoden ist ebenfalls möglich.

Dieses Buch kann sowohl Studenten technischer Fachrichtungen und der Informatik, die sich mit der Robotertechnik beschäftigen, als Einführung empfohlen werden; es gibt aber auch Anwendern von Industrierobotersystemen einen schnelleren Einstieg in die Programmierung und erläutert die Methoden zur Bewegungsbeschreibung von Roboterarmen und anderen Mehrachsengeräten. Entwickler finden darin einen Überblick über leistungsfähige Regelungskonzepte.

P. Kopacek, I. Nemetz

Vehicle Location and Navigation Systems

Yilin Zhao

Artech House,Inc. 1997
345 pages
ISBN 0-89006-861-5

The book is divided into two parts: Part I, Basic Modules (building blocks, Chapters 2 through 8), and Part II, Systems (integrated modules, Chapters 9 through 11). Chapter 1 is a general introduction to the vehicle location and navigation field, with a historical background and introduction to modern vehicle location and navigation systems. Chapters 2 through 8 describe the basic building modules from which systems are constructed. Chapter 2 discusses the digital map database generally used in vehicle location and navigation systems, as well as relevant reference coordinate systems and standards. Chapter 3 studies the positioning sensors and fusion methods used to locate the vehicle. Chapter 4 describes the map-matching algorithms employed for enhancing the positioning module and systems. Chapter 5 discusses various route-planning algorithms for finding the route that results in a minimum travel cost. Chapter 6 explains how to guide the vehicle along a given planned route that was derived on the basis of either the static travel cost or the dynamic travel cost. Chapter 7 discusses various human factors guidelines and technologies used for the human-machine interface, which interacts with the location and navigation computer and devices. Chapter 8 examines the different wireless communications technologies available for supporting vehicle communications.

Chapters 9 through 11 are devoted to systems. Chapter 9 discusses how to integrate the modules discussed in Part I to construct a variety of realistic autonomous vehicle location and navigation systems capable of operating without the support of a centralized remote computing and communications facility. Chapter 10 explains how a wireless communications module might be integrated with the vehicle and infrastructure to enhance in-vehicle systems using a centralized host computer. Chapter 11 uses an actual dynamic route guidance based system as a case study to shed some light on system architecture and implementation issues. The book concludes with Chapter 12, which describes past lessons learned and future directions projected. Many useful summery tables and recent references are given, and many new technologies are introduced and practical questions discussed.
This book permits a broad audience to learn about this fast-growing field without any prior knowledge. The concepts and algorithms are applicable to a broad range of location and navigation systems and form the foundation needed to understand, design, and implement advanced ITS. To help professionals with different backgrounds, the book attempts to limit the amount of mathematics used, and includes many newly developed technologies, practical examples, summary tables, and recent references. Even nonengineers should be able to understand much of the material.

This book provides a detailed description of both the principles and practices of modern vehicle location and navigation systems in a single source. It combines information scattered among many different engineering fields into a single volume, a comprehensive introduction to the highly interdisciplinary field of ITS.

P. Kopacek

Rechnergestützte Montageplanung

Montageplanung und Simultaneous Engineering

Wolfgang Holle

Carl Hanser Verlag, 2002
220 Seiten, 169 Bilder und eine CD-ROM
ISBN 3-446-21986-2

Das Buch entstand aus der langjährigen Tätigkeit des Autors in der Industrie und an der Technischen Universität Ilmenau. Zielsetzung ist es sowohl dem Studierenden als auch den in der Praxis tätigen Techniker eine Einführung in die rechnergestützte Planung von Montageaufgaben zu geben. Nach einer Einführung werden Basisinformationen und Planungsdaten, wie Produktmodellausrüstung, Prozessdaten, Mechanisierungsniveau sowie Anforderungen an Planungswerkzeuge und Planungssoftware kurz umrissen. Kapitel 3 beschäftigt sich mit Planungsschritten, Informationsbedarf und Informationsverarbeitung. Beginnend mit der Produktmodellerfassung über die Produktbewertung, der Bereitstellung von Teilen und Baugruppen wird schließlich auf die Gestaltung des Montageprozesses sowie die Gestaltung der Arbeitsstationen und Arbeitsplätze eingegangen. Es folgt ein Kapitel über die Gestaltung mechanisierter oder automatisierter Montagen. Es werden zunächst die Montagekosten diskutiert und sodann die einzelnen Komponenten für die automatisierte Montage vorgestellt. Montageschritte werden definiert, eingehend auf die Anordnung von Teilen, Vorrichtung und Werkzeugen sowie der Planungsablauf eingehend beschrieben. Hervorzuheben ist das Kapitel 5, welches sich mit dem wesentlich moderneren Gebiet der Demontage beschäftigt. Eine Einschränkung besteht darin, dass dieses Kapitel ausschließlich die Demontage zur Kreislaufwirtschaft behandelt. Die Demontage wird hier als Umkehrung der Montage vereinfacht dargestellt, was jedoch zu einigen Fehlschlüssen führen kann. Den Abschluß bilden Möglichkeiten der Projektbewertung, ein eher dürftig ausgefallenes Literaturverzeichnis sowie Beschreibung der beiliegenden CD-ROM.

Der Autor erwähnt treffend im Vorwort, dass es schwer bis aussichtslos war und auch derzeit noch ist die Gedanken der automatisierten Montage und Demontage bei den Gestaltern des Produktes zu verankern. Dieses Buch könnte dazu beitragen diesem latenten Mangel abzuhelfen. Die Planungsschritte werden anhand des Softwarepaktes ILMOPLAN (Ilmenauer Montage Planungssystem) erläutert. Die Entwicklung dieses Softwarepaketes begann in den späten siebziger und frühen achtziger Jahren an der Technischen Universität Ilmenau und war ursprünglich für Großrechner konzipiert. Nach eingehendem Studium der CD-ROM ist es dem Autor und seinen Mitarbeitern auch gelungen dieses Softwarepaket sowie Hilfssoftware zur Visualisierung von CAD und ähnlichem in – leider nur – Windows 95 zu konvertieren.

Zusammenfassend kann festgestellt werden, dass das Buch für eine erste Einführung in die rechnergestützte automatisierte Montage und Demontage empfohlen werden kann.

P. Kopacek

Wörterbuch der Elektronik, Datentechnik und Telekommunikation

Vittorio FERRETTI
2. Auflage Springer 2000

ISBN 3-540-67077-7 Deutsch – Englisch
ISBN 3-540-67078-5 Englisch – Deutsch

Dieses zweibändige wohl vollständigste Fachwörterbuch im deutsch-englischen Sprachraum enthält über 139000 Einträge aus 97 Teilgebieten der Elektronik, Informationstechnik (Informatik) und Telekommunikationstechnik sowie Hauptbegriffe der angrenzenden Fachgebiete und Grundlagenwissenschaften.

Gegenüber der ersten Auflage vor 7 Jahren wurden die Einträge um 35 % erweitert, was nicht zuletzt auf die Durchdringung der Telekommunikationstechnik mit Datentechniken zurückzuführen ist. Die Hauptstärke des vorliegenden Werkes liegt darin, dass es wohl das einzige am Markt ist, welches die Terminologie der Telekommunikation mit jener der Datentechnik verbindet.

Das lexikalische Konzept, bestehend aus der Nennung des Fachgebietes für jeden Eintrag einschließlich Zusatzinformationen wie Kurzdefinitionen, Synonyme, Quasisynonyme, Gegensatzwörter, Ober- und Unterbegriffe sowie das tabellarische Layout macht dieses Buch zu einem unentbehrlichen Hilfsmittel sowohl für den praktisch tätigen Ingenieur, dem Studierenden sowie insbesonders auch Übersetzern. Bei der Priorisierung deutscher Fachwörter wurde versucht einen Mittelweg zwischen den Extremen der unbedingten Verdeutschung und der vorbehaltslosen Übernahme englischer Fachausdrücke gewählt, was vorzügliche gelungen ist. Eine alphabetische Liste der Fachgebiete sowie eine Liste der Zusammenhänge dieser untereinander vervollständigt dieses ausgezeichnet gelungene Werk.

P. Kopacek

Projekt Engineering
Ingenieurmäßige Softwareentwicklung in Projektgruppen

Herwig Mayr
Fachbuchverlag Leipzig
im Carl Hanser Verlag, 2001
294 Seiten
ISBN 3-446-21779-7

Dieses Buch gibt eine fundierte Einführung in die ingenieurmäßige Softwareentwicklung für kleinere bis mittlere Softwarepakete mit einer Entwicklungszeit von weniger als 10 Personenjahren. Diese Struktur ist in Mitteleuropa typisch für die moderne Softwareentwicklung. Eines der Hauptanliegen ist es klar zu machen, dass moderne Softwareerstellung nicht mehr Kunst eines einzelnen, sondern eine Ingenieurdisziplin ist, für welches ein Team von Entwicklern einzusetzen ist.

Das Buch stellt daher kein Lehrbuch für Programmieren oder Algorithmenentwurf dar, es gibt viel mehr eine detaillierte Anleitung zu gutem Design und sauberer Gestaltung einer guten Implementierung. Das Hauptaugenmerk liegt jedoch auf dem gesamten Entwicklungszyklus eines Softwareprojektes von der Definitionsphase bis zur Implementierung.

Das Buch ist in 12 Kapitel unterteilt, welche den Hauptabschnitten

- Einführung in die ingenieurmäßige Projektentwicklung
- Projektphasenübergreifende Aufgabenbereiche und
- Projektphasenspezifische Aufgabenbereiche

zugeordnet sind.

Das Buch richtet sich an Studierende, die sich Wissen über die projektorientierte Softwareentwicklung selbst aneignen wollen, an Lehrende, die eine Vorlesung über die Team-orientierte Entwicklung von Softwareprojekten vorbereiten sowie an Softwareentwickler, die ihr Wissen im Bereich der ingenieurmäßigen Abwicklung von Softwareprojekten sowie des Arbeitens im Projektteam erweitern wollen. Dieses Buch kann daher, da es umso mehr von einem praxiserfahrenen technischen Softwareentwickler geschrieben wurde, den vorgenannten Zielgruppen wärmstens empfohlen werden.

P. Kopacek

Eco-Design
Effiziente Entwicklung nachhaltiger Produkte mit euroMat

Günter Fleischer (Hrsg.)

Springer Verlag, Berlin Heidelberg 2000
372 Seiten, 96 Abbildungen,
80 Tabellen und 38 Flußdiagramme

ISBN 3-540-65814-9

Im Zeitalter gravierender Umweltschäden, begrenzter Verfügbarkeit natürlicher Ressourcen und einem ständig wachsenden Müllaufkommen ist es erforderlich neue Wege zu beschreiten. euroMat'98 (entwicklungsbegleitendes Instrument für umwelt- und recyclingorientierte Materiallösungen) versucht den heutigen Anforderungen bei der Materialauswahl gerecht zu werden – es ist ein erster Schritt in Richtung des systematisch betriebenen Eco-Designs. Dieses Instrument liefert dem Leser ungewichtete produktbezogene Bewertungen für die Bereiche Gebrauchseigenschaften, Fertigung, Recycling, Arbeitsumwelt, Umwelt und Kosten über den gesamten Produktlebensweg. Der Produktentwickler hat somit die Möglichkeit sowohl auf technischer als auch kostengünstiger Basis eine optimale Materialentscheidung zu treffen.

Dies ist in der heutigen Zeit von großer Bedeutung, da die gesetzliche Entwicklung einerseits in Richtung Produkthaftung für den gesamten Lebensweg des Produktes geht und andererseits die Nachweispflicht der Umweltverträglichkeit neuer Produkte gefordert wird. Nach dem Kreislaufwirtschaftsgesetz müssen Produkte so gestaltet werden, dass die genutzten Materialien mit geringstem Aufwand wieder dem Wirtschaftskreislauf zugeführt werden können. Eine vorausschauende Materialauswahl, welche die Auswahlkriterien für den gesamten Lebensweg berücksichtigt, ist jedoch auch für die langfristige Wettbewerbsfähigkeit von großer Bedeutung. Um zu möglichst umfassenden Daten zu gelangen und eine einseitige Datenverwertung zu vermeiden, wurden die Datenbanken sowohl in firmenspezifische, vertrauliche als auch allgemein verfügbare, anonymisierte geteilt.

In diesem Buch werden zuerst die Module Technik, Arbeitsumwelt, Umwelt, Kosten und Gesamtbewertung erläutert, sodann wird anhand von Praxisbeispielen in Zusammenarbeit von 6 Instituten und 11 Unternehmen aus den verschiedensten Branchen (von der Automobil- bis zur Verpackungsbranche) diese Methode validiert. In den Kapiteln Bewertung, Fehlerbetrachtung und Geltungsbereich ist die hohe Aussagesicherheit jedes Iterationsschritts, wobei die Aussageschärfe mit den Iterationsstufen steigt, ermittelt worden.

Dieses Buch stellt für alle an einer Produktentwicklung beschäftigten Personen eine zusätzliche Hilfe zur Entscheidungsfindung dar.

P. Kopacek

Planung und Inbetriebnahme einer semiautomatischen modularen Demontagezelle für die Elektronikindustrie

Dr. Harald Zebedin

Begutachter: O.Univ.Prof.Dr.Dr.h.c.mult. P. Kopacek
O.Univ.Prof.Dr. P.H. Osanna

Gravierende Umweltschäden, die begrenzte Verfügbarkeit natürlicher Ressourcen und das ständig wachsende Müllaufkommen erfordern neue Konzepte zur Entsorgung technischer Gebrauchsgüter – anstelle von Deponierung und Verbrennung ist künftig ein größtmögliches Recycling erforderlich. Angesichts steigender Stückzahlen und hoher Personalkosten setzt dies eine Demontagetechnik voraus, die weitaus rationeller arbeitet als die herkömmlichen – überwiegend manuellen – Verfahren. Ein Problem bereitet neben der anfallenden Schrottmenge, vor allem die komplexe Zusammensetzung der Altprodukte. Daher ist es erforderlich ökonomisch und ökologisch sinnvolle Verfahren zur Demontage, Aufbereitung und Aufarbeitung von Produkten zu entwickeln, wobei die Demontage der erste Schritt für ein hochwertiges Recycling ist.

Ziel der Arbeit ist es auf Basis eines Industrieroboters eine flexible Demontagezelle zu entwickeln, um die Anforderungen, die an ein solches System gestellt werden, durch eine intelligente Systemsteuerung, unterschiedliche Module für die jeweilige Art der Demontage, sowie flexible Greif- und Demontagewerkzeuge erfüllen zu können. Im Kapitel 4 wird die grundlegende Struktur und die Forderungen an das Entstückkonzept vorgestellt. Die Hauptarbeit besteht in der Entwicklung und Konstruktion der unterschiedlichen Module und der Gesamtanlage, deren Aufgaben in Verbindung mit dem jeweiligen Demontagewerkzeug im Kapitel 5 beschrieben und analysiert werden. Dabei mussten mehrere Entwicklungsschritte durchlaufen werden. Dies führte zum Teil ein einem erheblichen Aufwand, speziell für die Programmierung des Roboters.

Um die Reaktionsfähigkeit eines automatisierten Systems auf Produkt- und Variantenwechsel zu gewährleisten, sowie auf gebrauchsbedingte Produktveränderungen (zum Beispiel Reparatur, Beschädigung, Verschmutzung) reagieren zu können, werden leistungsfähige Sensorsysteme mit entsprechender Datenauswertung integriert, damit die zu demontierenden Teile oder Komponenten unbeschädigt und mit einer hohen Effizienz der Wiederverwendung dem Markt zugeführt werden können. Dabei wurden umfangreiche Untersuchungen durchgeführt um die optimale Gestalt des Greifers zu bestimmen. Sowohl die Gestalt des Greifers als auch die dazugehörige Programmierung des Roboters wird im Kapitel 6 beschrieben.

Der Aufbau der Gesamtanlage und das Testen der unterschiedlichen Module im Betrieb bilden den Abschluss der Arbeit. Ein in Entwicklung befindliches und implementiertes Modul (Infrarot Modul) soll die Basis auch für die in Zukunft immer mehr und mehr am Markt befindliche BGA Chips bilden. Da sich die Elektro- und Elektronikindustrie laufend weiterentwickelt, wodurch sich veränderte Rahmenbedingungen ergeben, wird versucht auf ähnlichen Gebieten Erfahrungen, zum Teil auch mit der Industrie, zu sammeln um diese auch in die bestehende Anlage einfließen zu lassen. Den Abschluss der Arbeit bildet der sogenannte Anhang, in dem konstruktive Details, die Programme des Roboters, sowohl der realisierten Prototypen-Zelle als auch der Industrie-Zelle zu finden sind.

Multi Agent Systems Robot Soccer

Dr. Gregor Novak

Begutachter: O.Univ.Prof.Dr.Dr.h.c.mult. P. Kopacek
O.Univ.Prof.Dr. H. P. Jörgl

Das Ziel dieser Arbeit war die Entwicklung eines autonomen Robotersystems anhand des praxisorientierten Beispiels Roboterfußball. Roboterfußball ist geradezu prädestiniert, als Testumgebung für Multi Agent Systeme zu fungieren, beinhaltet es doch nahezu jedes Element eines autonomen Systems und vermittelt überdies eine Idee, wie ein autonomes Robotersystem realisiert werden könnte. Obwohl es faszinierend ist, kleine Roboter Fußball spielen zu sehen, gleicht Roboterfußball auf den ersten Blick einem eher einfachen Spiel, denn seine Komplexität zeigt sich erst im Detail. Hierbei ergibt sich allerdings die Möglichkeit, sich mit nahezu jeder Technologie zu befassen, die für ein autonomes System notwendig ist. Um wichtige Teile der Umgebung in eine Form zu bringen, welche die Steuerungseinheit versteht, ist ein Sensorsystem notwendig. Dieses besteht beispielsweise aus einem Bildverarbeitungssystem, Enkodern usw. Die Steuereinheit generiert Aufgaben unter Zuhilfenahme beispielsweise von neuralen Netzwerken, Fuzzy Logik oder einfachen wenn - dann Regeln. Aus diesen Aufgaben kann in weiterer Folge eine Trajektorie erzeugt werden, welche den Bewegungsablauf des Roboters vorgibt. Der komplette Kontrollmechanismus ist in ein Framework eingebettet, welches die Möglichkeit der verteilten Entwicklung und des schrittweisen Testens unterstützt. Aufgrund der Geschwindigkeit des Spiels ist die Zykluszeit des Kontrollmechanismus sehr kurz, was zu Problemen mit dem Betriebssystem führt. Theoretisch wird ein Echtzeitbetriebssystem benötigt. Schlussendlich steht und fällt das Gesamtsystem mit einem entsprechenden Miniaturroboter. Die Aufgabe eines Roboters besteht darin, einer gegebenen Trajektorie präzise zu folgen und die Sollgeschwindigkeit rasch zu erreichen. Die Konstruktion eines Miniaturroboters erfordert Wissen auf den Gebieten des Maschinenbaues, der Elektrotechnik, der Regelungstechnik sowie Informatik. Eine weitere Schwierigkeit stellte das Problem dar, alle Bauteile des Miniaturroboters in einem kleinen Würfel unterzubringen. Jeder zusätzliche eingebaute Sensor sowie jede zusätzliche auf dem Roboter untergebrachte Intelligenz beschleunigen das Gesamtsystem und erhöhen die Chance zu gewinnen.

Trotz aller gegebenen Detailprobleme und der Komplexität eines Roboterfußballspiels wurde niemals das Ziel aus den Augen verloren, ein universell anwendbares System zu entwickeln, welches auf der Grundidee eines komplett autonomen Roboters basiert. Für dieses Ziel wurde ein Schichtenmodell mit entsprechenden universellen Schnittstellen erdacht, wobei die oberste Schicht dieses Modells eine Plattform zur Entscheidungsfindung darstellt.

Um das gesetzte Ziel zu erreichen, wurden sowohl ein neuer Roboter entwickelt als auch ein objektorientiertes Programm in der Programmiersprache C++ geschrieben. Die Aufgabe des Programmteils ist es, die Aktionen des Roboters zu bestimmen. Dies führt zu einer Sollbahn, welcher der Roboter exakt zu folgen hat. Basierend auf der Idee, die menschliche Verhaltensstruktur zu reproduzieren und mit einem System zur einfachen Implementierung von Verfahren zur Entscheidungsfindung zu koppeln, wurde ein Schichtenmodell entworfen und objektorientiert abgebildet. Jeder Teil des Programms besitzt eine eindeutig definierte Schnittstelle, welche die Verbesserung von Programmteilen ermöglicht, ohne den gesamten Code zu kennen.

FIRA Robot World Congress Austria

October 1-3, 2003, Vienna, Austria

IHRT

TU WIEN — Technische Universität Wien / Vienna University of Technology

Call for Papers

Parallel to the FIRA World Cup Austria the FIRA World Congress will be organized. The congress will review and discuss the major advances in science and technology regarding robotics. Particular emphasis will be given to the applications and development in the area of entertainment/education/service/ personal robots. The contribution should focus on theories, principles and developments as well as application in the common field of robotics.

Topics include but are not limited to:
Specific topics

- Artificial Life
- Adaptive Systems
- Entertainment Robots
- Humanoid Robots
- Human Robot Interface
- Intelligent Control
- Intelligent (Smart) Sensors
- Legged Robots
- Machine Learning
- Robot Soccer System
  - HuroSot
  - KheperaSot
  - MiroSot
  - NaroSot
  - RoboSot
  - SimusoSot
- Machine Vision
- Mechatronics
- Multi-Agent Systems
- Navigation
- Path Planning
- Robot Soccer Simulation
- Sensors and Control
- Supervisory Control

Regular Paper Submission :
Authors should submit an extended abstract electronically in pdf format - 600 words – the email address : e318@ihrt.tuwien.ac.at. The first page of the paper must be contain the title, the names and affiliations of the authors, email address, and four key words/phrases. Please identify the corresponding author giving full contact information including email.

Invited Sessions:
The Program Committee also solicits proposals for invited sessions within the technical scope of the congress. Each proposal for an invited session should describe the theme and scope of the proposed session and how the papers form a cohesive and complementary in the session topic. One session typically contains six (6) papers. The proposal must contain the title and theme of the session and a list of paper's title, names and email addresses of the corresponding authors. Session proposals must be submitted through the email address : e318@ihrt.tuwien.ac.at.

Proceeding and CD Rom
A CD-ROM and a Book of Abstract will be provided at the congress. A hard copy for the proceeding will be available at an additional charge.

Important dates :

- Abstract submission ..... June 15, 2003
- Notification of Acceptance ..... July 15, 2003
- Camera-ready Manuscript ..... August 15, 2003

For further information, please contact the secretariat :
FIRA World Congress Austria
Institute for Handling Devices and Robotics
Vienna University of Technology
Favoritenstr. 9-11/318
A-1040 Vienna, Austria
Tel. : +43-1-58801 31801, Fax : +43-1-58801 31899
Email : e318@ihrt.tuwien.ac.at
Homepage : http://www.ihrt.tuwien.ac.at/FIRAWC03

General Chair
P. Kopacek (Austria)

International Program Committee
P. Albertos (Spain)
M. Cecarelli (Italy)
H. S. Cho (Korea)
R. Dillmann (Germany)
T. Dinibütün (Turkey)
G. Fürnsinn (Austria)
F. Harashima (Japan)
B.R. Hong (China)
J. H. Kim (Korea)
W. H. Kwon (Korea)
D. Matko (Slovenia)
J. Paiuk (Argentina)
C. E. Pereira (Brazil)
B. Reusch (Germany)
N. Rozsenich (Austria)
P. Robinson (UK)
I. Rudas (Hungary)
J. Sa Da Costa (Portugal)
G: Schmidt (Germany)
R.D. Schraft (Germany)
J. Sitte (Australia)

Organized by
- Institute for Handling Devices and Robotics – IHRT, Vienna University of Technology - VUT

Cooperation with
- Austrian Society for Automation and Robotics – ÖGART
- Austrian Society for Measurement and Automation – ÖGMA
- Austrian IFAC Supervisory Board – IFAC Beirat Österreich

Sponsored by
- City of Vienna
- Vienna University of Technology - VUT
- Austrian Computer Society – OCG

IFAC Workshop on "Technology and International Stability" (SWIIS '03)

July 03 - 05, 2003
Waterford,
Republic of Ireland

SCOPE

Technology development leads to dramatic changes in international stability. The scope of this IFAC/SWIIS Workshop is to offer insights into mitigating unwanted side-effects of rapid development and to share methodologies for appropriate ways of managing the introduction to technologies which will alter social stability.

TOPICS

- Social Aspects of Technology
- Managing the Introduction of Technological Change
- Ethical Aspects of Technological Proliferation
- Modeling Social Change Likely to Follow a New Technology
- Case Studies of Technology Transfer and Social Change
- Technology and Environmental Stability

MAILING ADDRESS:

Institute for Handling Devices and Robotics (E318)
Vienna University of Technology
SWIIS'03
Favoritenstraße 9-11/E318
A-1040 Vienna - Austria
Tel.: +43/1/ 58801 31801
Fax: +43/1/ 58801 31899
swiis@ihrt.tuwien.ac.at

Datum	*Veranstaltung*	*Ort*	*Weitere Informationen erhältlich bei:*
7.-11.10.2002	ISR 2002 – International Symposium on Robotics	*Stockholm Sweden*	http://isr.2002.com e-mail: isr2002@iip.kth.se
11.10.2002	17. Österreichischer Automatisierungstag	*Linz*	e-mail: nemetz@ihrt.tuwien.ac.at http://www.ihrt.tuwien.ac.at/IFAC_AUT
15.10.2002	1. Türkisch-österr. Automatisierungstag	*Istambul*	e-mail: kopacek@ihrt.tuwien.ac.at
3.-5.11.2002	Intern. Workshop on Robots for Humanitarian Demining	*Vienna Austria*	http://www.ihrt.tuwien.ac.at/HUDEM e-mail: kopacek@ihrt.tuwien.ac.at
9.-11.12.2002	2nd IFAC Conference Mechatronic Systems – in co-operation with IEEE/ASME	*Berkeley CA USA*	http://www.unex.berkeley.edu/eng/mech mechatronics2002@unx.berkeley.edu
5.-7.2.2003	IMACS/IFAC Symposium Mathematical Modelling – 4th MATHMOD	*Vienna Austria*	e-mail: inge.troch@tuwien.ac.at
12.3.2003	4. Österr.-brasilian. Automatisierungstag	*Sao Paulo Brazil*	e-mail: kopacek@ihrt.tuwien.ac.at

Datum	*Veranstaltung*	*Ort*	*Weitere Informationen erhältlich bei:*
8.-11.4.2003	IFAC Conference Intelligent Control Systems and Signal Processing – INCONS 2003	*Faro Portugal*	http://conference.ptrede.com e-mail: icons03@ualg.pt
4.-6.6.2003	American Control Conference (in Co-operation with IFAC)	*Denver, CO USA*	http://acc2003.me.berkeley.edu/ e-mail: bequeb@rpi.edu
9.-11.6.2003	5th IFAC Symposium Fault Detection, Supervision and Safety of Technical Processes – SAFEPROCESS	*Wahington, DC USA*	http://safeproc.gmu.edu e-mail: jgertler@gmu.edu
16.-18.6.2003	6th IFAC Symposium Advances in Education - ACE 2003	*Oulu Finland*	http://ntsat.oulu.fi e-mail: office@atu.fi
1.-3.10.2003	FIRA Robot World Congress Austria	*Vienna Austria*	e-mail: e318@ihrt.tuwien.ac.at http://www.ihrt.tuwien.ac.at/FIRAWC03
28.9.-3.10. 2003	FIRA Robot World Cup Austria	*Vienna Austria*	e-mail: e318@ihrt.tuwien.ac.at http://www.ihrt.tuwien.ac.at/FIRAWM03