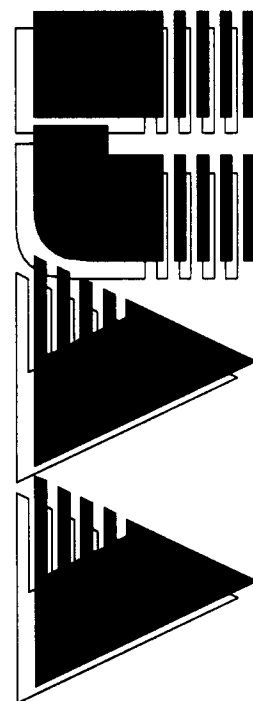


INTERNATIONAL JOURNAL

HEFT 1

Jg. 12, (2004)

AUTOMATION AUSTRIA



INHALT

Seite

PERME, T.; NOE, D.: Simulation mit Erweiterten Petri-Netzen zur Planung von Montagesystemen	1
JAMBRICH, G.: Lyapunov based approach to Adaptive Fuzzy Control	17
WEINMANN, A.: H_∞ Controllers under H_∞ Constraints	37
SCHLAGLICHTER	62
BUCHBESPRECHUNGEN	63
DISSERTATIONSKURZFASSUNGEN	66
VERANSTALTUNGEN	70

Simulation mit Erweiterten Petri-Netzen zur Planung von Montagesystemen

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

Kurzfassung

Eine Analyse derzeit verfügbarer Montageplanungssysteme zeigt, dass die Simulation einen wesentlichen Beitrag zum Entwurf dieser Systeme leisten könnte. Bisher hat die Simulation trotz ihren hohen wirtschaftlichen Nutzens in der Praxis noch keine richtige Akzeptanz gefunden. Ein Grund dafür ist dass die derzeit verfügbaren Simulationssoftwarepakete schwierig zu handhaben sind und preislich für KMU's undiskutabel sind.

Hier wird ein modernes und praxistaugliches Simulationssystem, das als Kern eines rechnergestützten Systems zur effizienten Planung von Montagesystemen dienen kann vorgestellt. Es basiert auf Petri-Netzen welche in ein bestehendes CAD System eingebunden werden können. Dadurch kann aus einer CAD Zeichnung direkt das Montagesystem geplant werden.

1 Einleitung

Planung ist eine der wichtigsten Phasen im Lebenszyklus eines Produktionssystems, da hier die wichtigsten Eigenschaften – wie auch die Kosten – des Produktes bestimmt werden. Bei der Planung geht es hauptsächlich darum, die richtigen Entscheidungen zu treffen. Bei steigender Komplexität und verringerten Lebenszyklen sind Wissen, Erfahrung, Kreativität und Intuition des Planers alleine nicht mehr ausreichend. Vielmehr wird ein objektives Bewertungsmittel benötigt, mit welchem der Planer die Ergebnisse seiner Entscheidungen realistisch einschätzen kann, um so mögliche Planungsfehler rechtzeitig zu entdecken. Eine Möglichkeit dafür wäre das Experimentieren mit wirklichen Systemen, die aber aus wirtschaftlichen Gründen kaum verwendet wird.

Genauere, realitätsnahe Ergebnisse kann man im Planungsstadium mit der Simulation ermitteln, eventuell durch Experimentieren mit einem virtuellen System. Simulation gehört schon seit Jahren zu den unentbehrlichen Werkzeugen in der Planung von Produktionssystemen, weil nur damit die tatsächlichen Eigenschaften des zu realisierenden

Systems schon in der Entwurfsphase realitätsnah dargestellt und deren Effizienz abgeschätzt werden kann. Simulation hat aber trotz des hohen wirtschaftlichen Nutzens in der Praxis noch keine richtige Akzeptanz gefunden. Die Gründe dafür liegen einerseits im großen Aufwand, der erforderlich ist, andererseits sind die existierenden Simulationsprogramme und -systeme selbst, deren Anwendung vom Benutzer – meist dem Planer selbst – vertiefte Simulationskenntnisse verlangt.

Bei der Planung von Montagesystemen ist die Entwicklung eines Layouts die letzte Stufe, in welcher auf das zu realisierende Montagesystem noch wesentlich beeinflusst werden kann. Während der Konzeptions-, Entwurfs- und Detaillierungsphase werden alle wichtigen Eigenschaften des geplanten Montagesystems, inklusive Kosten und Kapazität und davon abhängige Montagekosten des Produktes, festgelegt.

Die Layouterstellung ist in starkem Maße ein kreativer Vorgang in welche der Planer unbedingt einbezogen werden sollte. Dabei sollte dem Planer ein auf die systematische Montageplanung basierendes und rechnergestütztes Planungssystem unterstützen. Folglich kann man die Anforderungen an ein Montageplanungssystem folgendermaßen zusammenfassen:

- Layouterstellung,
- automatische Modellerstellung für die Simulation
- Simulationsdurchführung und
- Montagestückkostenberechnung.

Mit derartigen Systemen wird die Planungsdauer verkürzt, die Montagestückkosten gesenkt und Investitionen minimiert. Für diese Zielsetzungen wird ein modernes und praxistaugliches Simulationssystem benötigt, welches die folgenden Anforderungen erfüllt:

- Modellierungs- und Simulationstauglichkeit der logischen und physischen Zusammenhänge der Systemelemente
- Integration in eine CAD Umgebung
- Bedienerfreundliche Mensch-Maschine Schnittstelle für Experten und für Planer
- 3D Animation.

Anhand der Konzepte der virtuellen Produktion [1], der GAM Modelle [2] der erarbeiteten Erweiterungen von Petri-Netzen (EPN) [3] und der Integration in eine CAD Umgebung [4] wurde das Programmpaket LASIMCO entwickelt. Dieses soll den Anwender beim Entwurf, Ausarbeitung, und Steuerung von Montagesystemen unterstützen. Im Rahmen dieses Beitrages wird gezeigt, wie die Anforderungen mit EPN im CAD System AutoCAD erfüllt werden können.

2 Modellierung und Simulation mit EPN

Die mathematischen Grundlagen von GAM Modellen, die erarbeitete Erweiterungen von Petri Netzen (EPN) und die allgemeine Modellierung mit EPN sind ausführlich in [3] beschrieben. Alle hier erforderlichen Definitionen sind ausführlich in [3] beschrieben. Weiters wird der Algorithmus für die Ausführung der Simulation mittels EPN erläutert. Damit kann der Benutzer die Modellierung mit EPN begreifen und somit die Simulation erfolgreich, intuitiv durchführen.

2.1 Simulation mit EPN

Kern der Simulation mit EPN ist ein modifizierter Schaltalgorithmus, der auf den theoretischen Grundlagen der erweiterten Petri-Netze beruht [3][5]. Dieser Schaltalgorithmus setzt sich aus zwei Schleifen zusammen (Abb. 1):

- innere Schleife: Hier werden alle aktivierten normalen und zeitlichen Transitionen geschaltet bzw. teilgeschaltet, bis keine aktivierte Transition mehr vorhanden ist.
- äußere Schleife: In dieser werden alle teilgeschalteten Transitionen mit kleinster Restzeit endgeschaltet. Die Simulationszeit wird dann um den Wert der kleinsten Restzeit erhöht – die Simulation geht in der inneren Schleife weiter bis keine Transition mehr aktiviert ist oder bis die Endzeit erreicht wird.

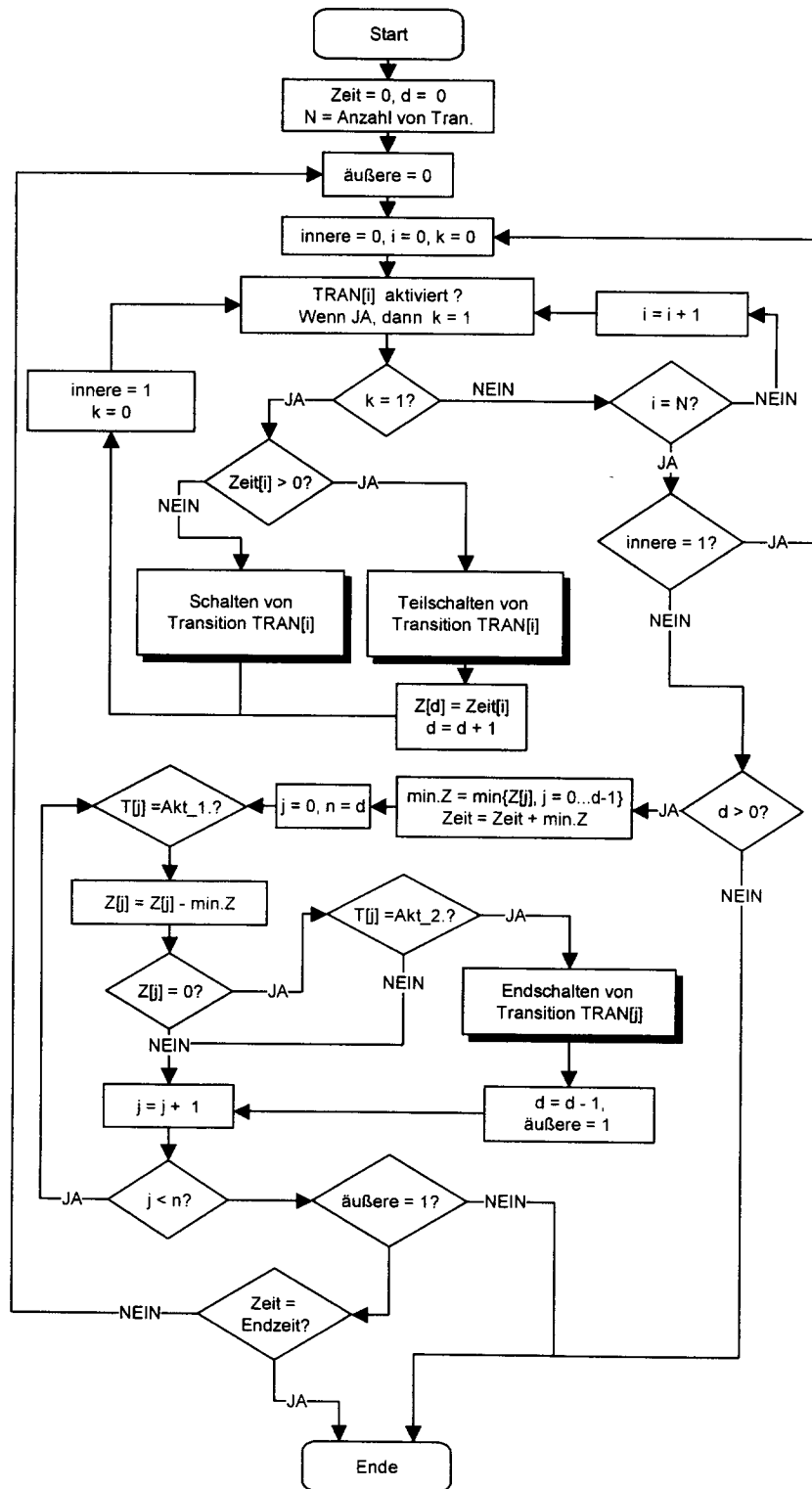


Abb. 1: Ablaufdiagramm des Algorithmus zur Simulation mit erweiterten Petri-Netze

Bei der Suche nach aktivierten Transitionen in der inneren Schleife werden die erweiterte Definition 3 [3] sowie die Definition 15 [3] verwendet. Bei „Akt_1“ wird lediglich der zweite Teil ($W(s,t) < 0$) der erweiterten Definition 3 – Bedingung (ii) – berücksichtigt. „Akt_2“ in der äußeren Schleife bedeutet, dass eine Transition Endschalten kann. Beim Schalten,

Teilschalten und Endschalten werden auch die Attribute und die Lage der Objektmarken verändert.

Für die Berücksichtigung der physischen Eigenschaften wurde ein Algorithmus entworfen, mit welchem die Objektmarken mit definierter Länge und Breite in einer Transition ihr räumliches Verhalten durch eine linearen Bewegung ($\text{Tran}(x,y,0)$) einhalten.

Teilgeschaltete Transitionen bilden eine Menge $T = \{(t_1, d_1) (t_2, d_2) \dots (t_n, d_n)\}$, welche im Programm durch eine Liste dargestellt werden. Jedes Element in dieser Liste bedeutet eine teilgeschaltete Transition und wird mit der Transition t und mit der Dauer d beschrieben (Definition 8 in [3]). Gemäß Definition 10 [3] und nach dem Simulationsalgorithmus werden in einem Schritt jene teilgeschalteten Transitionen endgeschaltet, welche die kürzeste Dauer aufweisen und gleichzeitig für das Endschalten aktiviert wurden (erweiterte Definition 3 ii [3]). Die Dauer jeder anderen teilgeschalteten und aktivierten (erweiterte Definition 3.i. [3] Fall $W(s,t) < 0$) Transition wird um die kleinste Zeitspanne verkürzt.

Dieses Verfahren ist für alle teilgeschalteten Transitionen geeignet, welche kein räumliches Verhalten direkt nachbilden – d.h. die Dauer der Transition wird eingegeben und nicht aus dem Abstand zwischen den räumlichen Ein- und Ausgangsstellen (Länge der Transition) und der Geschwindigkeit der Transition berechnet. Die Verkleinerung der restlichen Dauer einer teilgeschalteten Transition $(t_i, d_i) \in T$, mit der ein Transport eines Werkstückträgers auf einem Förderband nachgebildet wird, hängt nicht nur von der Erfüllung der Definition 3 [3] ab, sondern auch vom Abstand dieses Werkstückträgers bis zu einem in seinem Transportkurs liegenden anderen Werkstückträger. Ist dieser Abstand $G(t_i)$, dividiert durch die Geschwindigkeit $v(t_i)$ des Förderbands, kleiner als die kleinste Dauer (Dauer der endgeschalteten Transition), dann kann die restliche Dauer der Transition d_i nur um $G(t_i)/v(t_i)$ verkleinert werden (siehe Beispiel Abb. 2).

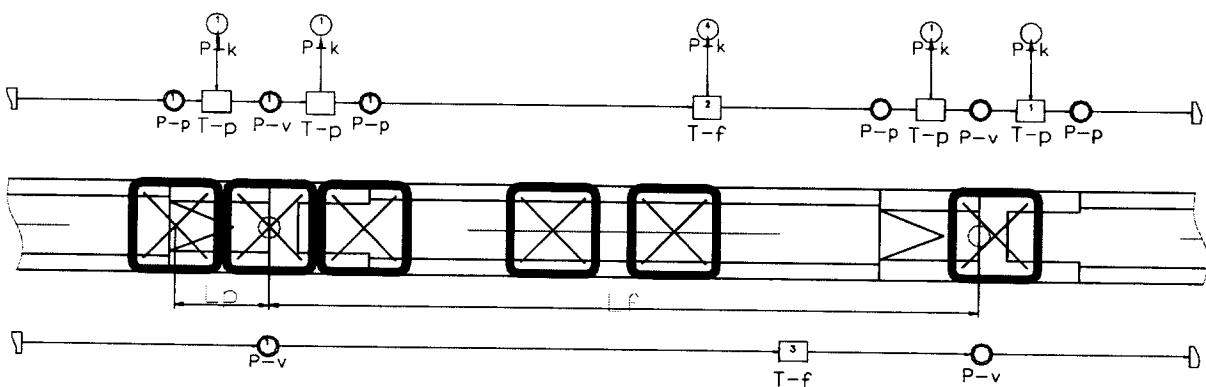


Abb. 2: Modell einer Förderbandstrecke

Eine Transportaktivität, in welcher sich ein Werkstückträger (Palette) auf einer Förderbandstrecke fortbewegt, kann für eine nicht-räumliche Simulation mit drei zeitlichen Transitionen, vier räumlichen und drei normalen Stellen nachgebildet werden (Abb. 2 oben, strichliert). Werkstückträger werden als Objekte definiert. Mit den Stellen $P-k$ wird die Kapazität der einzelnen Transitionen bestimmt – diese bestimmt die maximale Zahl an Paletten, welche sich auf einem durch die Transition ($T-f$ und $T-p$) nachgebildeten Abschnitt

der Förderbandstrecke befinden können. Jeder Transition ist eine Zeit zugeordnet, welche sich aus der Geschwindigkeit des Förderbandes und dem Abstand zwischen der Ein- und Ausgabe ergibt. Die Kapazität jeder räumlichen Stelle ($P-v$ und $P-p$) ist auf eine Palette beschränkt. Der Abstand zwischen der räumlichen Ein- und Ausgabestellen der Transition $T-p$ entspricht der Länge der Paletten. Die Transition $T-p$ sorgt dafür, daß eine Palette von einer Strecke zur nachfolgenden transportiert wird. Die Stelle $P-v$ bildet eine Verbindung von zwei Förderbandstrecken nach.

Der Ablauf der Simulation mittels eines solchen Modells kann folgendermaßen beschrieben werden. Eine Objektmarke befindet sich in einer Eingangsstelle $P-v$ der Transition $T-p$, d.h. eine Palette wird aus einer Strecke in die nachfolgende übergeleitet. Diese Transition ist aktiviert, wenn die Kapazität der Transition $T-p$ dieses ermöglicht ($P-k > 1$), d.h. wenn im Eingangsbereich (Länge einer Palette) der nächsten Strecke keine Palette mehr vorhanden ist. Nach dem Teilschalten der Transition $T-p$ wird das Objekt aus der Stelle $P-v$ und einer Marke aus der Stelle $P-k$ entfernt. Ein neues Element (t_{n+1}, d_{n+1}) wird in die Liste der teilgeschalteten Transitionen (Menge T) eingetragen. Die Dauer d_{n+1} wird aus dem Abstand zwischen $P-v$ und $P-p$ (Länge einer Palette) und der Geschwindigkeit des Transportbandes berechnet. Diese Dauer (d_i) wird bei jedem neuen Schritt (äußere Schleife der Simulation) um die kleinste Dauer der teilgeschalteten Transitionen so lange verkleinert, bis sie nicht die Kleinste oder Null ist. Wird die Transition auch für das Endschalten aktiviert (Stelle $P-p$ darf nicht belegt werden), dann endschaltet sie – d.h. das Element (t_i, d_i) wird aus der Liste entfernt, das Objekt wird in die Stelle $P-p$ eingegeben und die Stelle $P-k$ wird mit einer neuen Marke belegt.

Die Nachteile dieses Modells sind:

- Die Kapazität der Transitionen ($P-k$) muss immer eine ganze Zahl sein und kann nur für Objekte gleicher Größe genau berechnet werden. Paletten verschiedener Länge können nicht einbezogen werden.
- Ist die Dauer der Transportzeiten größer als die Bearbeitungszeiten, z.B. in Montageautomaten, kann diese Simulation ungenau sein.
- Die Geschwindigkeit der Förderbandstrecke muss konstant sein.
- Es ist keine richtige Animation. Wenn die Ausgangsstelle einer Transition für eine längere Zeit belegt ist, dann häufen sich alle Paletten am „Ende“ der Transition.

Mit einem Modell aus zwei räumlichen Stellen, einer zeitlichen Transition (Abb. 2 unten) und einer Objektmarke mit gegebener Länge und Breite wird eine Förderbandstrecke vollständig nachgebildet. Jede räumliche Stelle definiert eine Lage, sodass der Abstand zwischen zwei durch eine Transition verbundenen Stellen automatisch berechnet werden kann. Der Algorithmus für die Simulation wird so erweitert, daß bei der Verkleinerung der restlichen Dauer jeder teilgeschalteten Transition mit einer Objektmarke auch die Abstände zwischen den Objektmarken (Paletten) berücksichtigt werden. Der Abstand kann mit folgender Gleichung berechnet werden:

$$G_i = L + G_{i-1}, \quad (1)$$

G_i Abstand, mit welchem sich eine Objektmarke in einem Schritt tatsächlich fortbewegen kann.

L Abstand einer Objektmarke zu einer vorangehenden Objektmarke.

G_{i-1} Abstand, mit welchem sich eine nachfolgende Objektmarke in einem Schritt tatsächlich fortbewegen kann.

Mit dieser Gleichung können alle Abstände zwischen aneinander gereihten Objektmarken berechnet werden.

Eine Besonderheit dieser Erweiterung ist, dass der Algorithmus für die Berechnung der Abstände eine asymptotische Komplexität (n^2 ; n = Anzahl der teilgeschalteten Transitionen mit Objektmarke) aufweist.

2.2 Ergebnisse der Simulation mit EPN

Simulationsergebnisse sind Rückmeldungen über das Verhalten des simulierten Systems, z.B. analog zu Fertigungsrückmeldungen eines Betriebsdatenerfassungssystems. Sie werden je nach Modellierungssystem an definierten „Meßpunkten“ des Modellprozesses erzeugt und müssen während und nach Abschluss der Simulation in geeigneter Form aufbereitet werden, um eine Ergebnisinterpretationen zu ermöglichen [6][7]. Ergebnisse können während des Simulationslaufes (On-line-Laufbetrachtung) oder nach Beendigung des Laufes (Off-line-Laufbetrachtung) in drei Darstellungsmöglichkeiten – Statistik, Monitoring und Animation – aufbereitet werden.

Die dynamische Anzeige nutzt das Monitoring und die Animation. Mit der Animation kann der Benutzer nach dem Start des Simulationlaufes das Verhalten des Modells, d.h. zum Beispiel Ortsveränderungen von bewegten Objekten, direkt am Bildschirm beobachten.

Die Verdichtung der erzeugten Daten erfolgt mit Methoden aus der Statistik zu Kenngrößen und Parametern, die das Verhalten der abgebildeten Prozesse und Systeme anschaulich wiedergeben. Das logistische Verhalten eines Montagesystems läßt sich zum Beispiel durch Grundgrößen wie Durchsatz, Belegungszeit, Auslastung und Durchlaufzeit beschreiben.

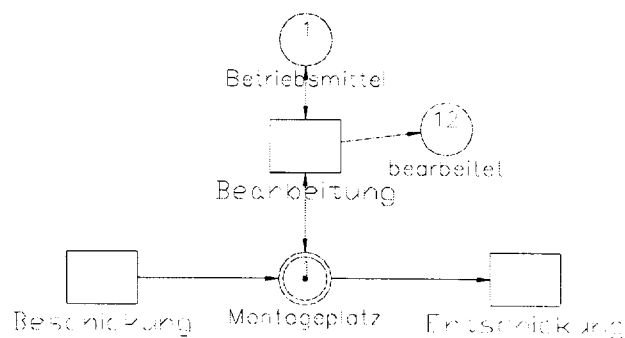


Abb. 3: Modell einer Bearbeitungsstation

Den Durchsatz, z.B. Objektdurchsatz, kann man im EPN Modell (Abb. 3) auf zwei verschiedene Arten ermitteln. Die einfachste Möglichkeit besteht darin, eine Stelle als Ausgangsstelle an eine Transition zu verbinden (Stelle „bearbeitet“, Abb. 3). Die Belegung dieser Stelle gibt direkt die Anzahl der (End-)Schaltungen der Transition – und damit auch der Durchsatz dieser Transition – an. Je nach Art der Transition kann damit der Durchsatz durch eine

Bearbeitungsstation oder des ganzen Montagesystems ermittelt werden. Eine solche Stelle dient als Zähler und kann auch für das Monitoring verwendet werden.

Eine andere Möglichkeit ist die Ermittlung der kumulativen Anzahl der Objektmarken (auch der normalen Marken) einer Stelle oder einer Transition aus ihrer Belegung ohne Verwendung einer zusätzlichen Stelle. So ergibt z.B. die kumulative Anzahl von Belegungen der Stelle „Montageplatz“ oder der Transition „Bearbeitung“ das gleiche Ergebnis wie die Belegung der Stelle „bearbeitet“ (siehe Abb. 3). Die kumulative Anzahl von Marken einer Stelle oder einer Transition wird während des Simulationslaufes berechnet, da die Speicherung der Belegung alle Simulationsschritte zu aufwendig ist.

Für eine Transition werden folgende statistische Kenndaten berechnet:

- minimale Anzahl von Teilschaltungen einer Transition,
- maximale Anzahl von Teilschaltungen einer Transition,
- mittlere Anzahl von Teilschaltungen einer Transition (L),
- kumulative Anzahl von Teilschaltungen einer Transition (N),
- minimale Dauer einer Transition,
- maximale Dauer einer Transition,
- mittlere Dauer einer Transition (t_w).

Für eine Stelle werden folgende statistische Kenndaten berechnet:

- minimale Anzahl von Marken in einer Stelle,
- maximale Anzahl von Marken in einer Stelle,
- mittlere Anzahl von Marken in einer Stelle (L),
- kumulative Anzahl von Marken in einer Stelle (N),
- mittlere Aufenthaltsdauer einer Marke in einer Stelle (t_w).

Diese statistische Kenndaten werden nur für bestimmte im Modell definierte Stellen und Transitionen berechnet. Um die Anlaufphase der Simulation zu berücksichtigen, werden die statistische Kenndaten erst nach einem bestimmten Zeitintervall ermittelt. Diese Zeit wird so definiert, daß die Anfangsbelegung keinen Einfluß auf die Simulationsergebnisse hat.

2.2 Animation mit EPN

Zum Monitoring und zur Animation dient eine graphische Darstellung der Petri-Netze. Die Änderung der Belegung von Stellen und Transitionen kann nach Ende des Simulationslaufes oder in jedem Simulationsschritt am Bildschirm angezeigt werden. So kann der Endzustand des Systems dargestellt oder ein on-line-Monitoring und/oder eine on-line-Animation durchgeführt werden.

Abb. 4 zeigt vier Ausschnitte eines Beschickungs-Bearbeitungs-Entschiebungs-Prozesses einer Montagestation. Mit der Änderung der Belegung kann auch der Zustand und damit der Ablauf des Prozesses beobachtet werden (Monitoring). Weiters bewegt sich auch die Objektmarke (Animation). Diese kann durch ein Zeichen (2D oder 3D) dargestellt werden (z.B. als Palette in Abb. 4) – es wird dann die EPN Simulation auch 2D oder 3D animiert. Die Objektmarke kann auch mit einigen wenigen Daten während des Simulationslaufes dargestellt werden (z.B. mit den Attributen und der restlichen Zeit). Diese Darstellungsform ist besonders bei der Validierung des Modells hilfreich.

Für eine realitätsnahe Animation mit EPN dient der Algorithmus zur Simulation physischer Eigenschaften. Die zeitlichen Transitionen werden durch Prozeß- und Förderbandtransitionen ergänzt, um die physischen Eigenschaften des Systems animieren zu können.

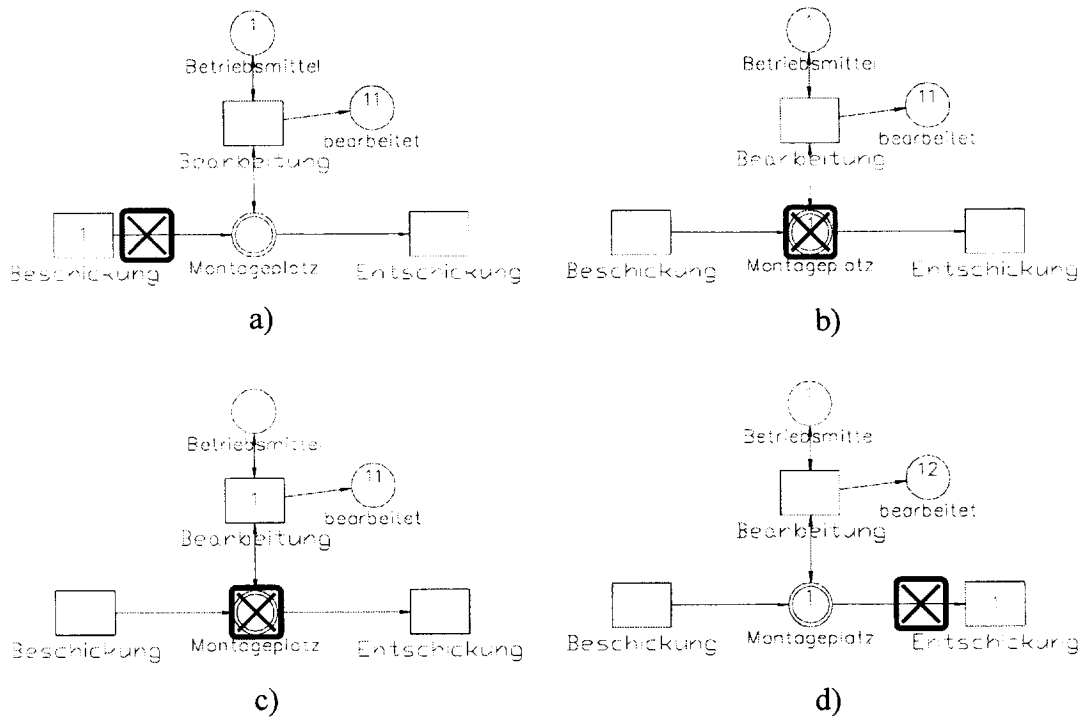


Abb. 4: Monitoring und Animation mit EPN. Die Objektmarke entspricht einer Palette mit einem eingespannten Werkstück. a) Palette wird beschickt; b) Palette wartet auf die Bearbeitung; c) Palette wird bearbeitet; d) bearbeitete Palette wird entschickt.

Eine Förderbandtransition bildet somit die Bewegung eines Objektes durch ein Fließband, Förderband, Palettentransportsystem oder durch ein anderes Betriebsmittel ab, in dem die Bewegung eines Objektes auch von anderen Objekten abhängt. Der Simulationsalgorithmus ermöglicht die Bewegung – und damit auch die Animation – aller sich in den Förderbandtransitionen befindlichen Objekte (Objektmarken). Die Änderung der Zeit aller anderen teilgeschalteten Transitionen wird normal berechnet, d.h. nur mit einer logischen Berücksichtigung der physischen Eigenschaften des Systems.

Eine Prozeßtransition hingegen dient der Nachbildung der Montageprozesse selbst. Zum Unterschied zu normalen Zeittransitionen oder Förderbandtransitionen hat die Prozeßtransition eine gemeinsame Eingangs- und Ausgangsstelle und das Objekt bleibt während der „Bearbeitung“ physisch in dieser Stelle. Die Position des Objektes ändert sich also nicht, wenn das Objekt in einer Prozeßtransition ist (Abb. 4.c).

3 Integration in eine CAD Umgebung

Die Erfahrungen zeigen, dass ein Modell oder eine Zeichnung sinnvollerweise nur einmal erstellt werden soll und diese dann im ganzen Planungszyklus (oder im gesamten

Lebenszyklus) eines Montagesystems zu verwenden. Ebenso verhält es sich mit dem Layout. Das Layout wird mit einem CAD-System erstellt und für die Simulation verwendet – die graphische Darstellung des Layouts kann dabei eventuell auch für eine Animation eingesetzt werden. Statt Schnittstellen zu definieren und spezielle Programme zu entwickeln, die zwischen den CAD- und Simulationssystemen eine Simulationsmodellerstellung und Datenübertragung ermöglichen, ist es effizienter ein Simulationssystem in ein CAD-System zu integrieren. Dadurch entsteht eine CAD-Umgebung [4][8], welche für Layouterstellung und Simulation ihre graphischen, zeichnerischen und andere Systemfunktionen zur Verfügung stellt (Abb. 5).

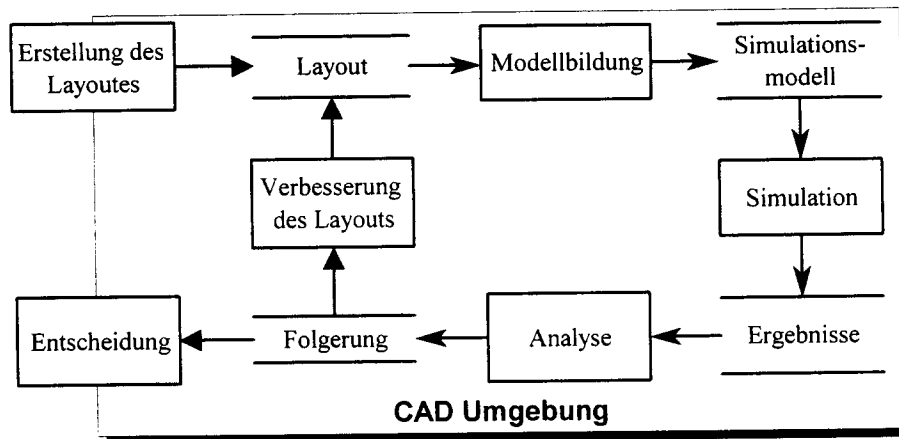


Abb. 5.: Layouterstellung und Simulation in einer CAD-Umgebung

Auf Grund der vorgeschlagenen Erweiterungen von Petri-Netzen wurde ein Modellierungsprogramm (PN_EDIT), ein Simulationsprogramm (PN_EXE) als Basis für das Montageplanungssystem LASIMCO entwickelt.

3.1 PN_EDIT

Eine Besonderheit von Petri-Netzen ist die einfache graphische Darstellung des Modells mittels Rechtecken (für Transitionen), Kreisen (für Stellen) und gerichteten Kanten (für Verbindungen). Dadurch wird die Modellstruktur transparent. Weiters erlaubt diese Darstellung ein einfaches Auffinden von Modellfehlern.

Die vorerwähnten Gründe bilden die Grundlage des Programms PN_EDIT, das praktisch ein graphisches Hilfsmittel zur Gestaltung des Simulationsmodells ist. Mit ihm werden alle für die Experimentphase notwendige Daten ermittelt. PN_EDIT ist eigentlich ein in AutoCAD integriertes AutoLISP Programm, wodurch die Zeichnungsfunktionen und das Datenbanksystem von AutoCAD ohne zusätzlichen Aufwand verwendbar sind.

Der Befehlsumfang von PN_EDIT erlaubt eine einfache Erstellung von EPN Modellen. Das System unterstützt die graphische Darstellung des EPN Modells und die Ergänzung mit notwendigen Daten. Weiters kann das Programm zur automatischen Generierung von Simulationsmodellen verwendet werden.

3.2 PN_EXE

PN_EXE ist ein rxADS Programm für AutoCAD 14, das anhand des mit dem PN_EDIT erstellten Modells und des in diesem Beitrag beschriebenen Simulationsalgorithmus die Simulationsexperimente berechnet. Es ist ebenfalls als AutoCAD-Modul konzipiert und ermöglicht die Simulation und Animation unter Verwendung von AutoCAD 3D Objekten.

Bei der Entwicklung der Software wurde besondere Aufmerksamkeit auf die Erzeugung hochwertiger Zufallszahlen gelegt, da die Stochastik der Zufallsvariablen und die Reproduzierbarkeit der Experimente wesentlich vom verwendeten Zufallszahlengenerator abhängen. Die Berechnung der notwendigen Anzahl an Experimenten, um eine mit der gewünschten Wahrscheinlichkeit genaue Aussage zu erzielen, erfolgte mittels eines speziellen Algorithmus. Damit sind alle für die Ausführung von Simulationsexperimente erforderlichen Funktionen unmittelbar in der CAD-Umgebung integriert.

3.3 LASIMCO

Das Softwarepaket LASIMCO geht davon aus, dass ein Layout mit allen notwendigen Daten für das geplante Montagesystem eine gute Basis zur Analyse mit Hilfe der Simulation darstellt. Mit dem gegebenen Layout sind die Struktur des Systems, alle Betriebsmittel und ihre räumliche Anordnung festgelegt. Erforderlich ist nur das Layout in ein Simulationsmodell umzuwandeln und die Simulation auszuführen. Der Kern des Konzeptes besteht in einer automatischen Generierung des Simulationsmodells aus einem Layout. Der Planer baut unwissentlich bei der Erstellung des Layouts bereits automatisch ein Simulationsmodell auf. Abbildung 6. zeigt das Konzept. Es besteht aus drei Einheiten: Erstellung des Layouts, Modellaufbau und Simulation.

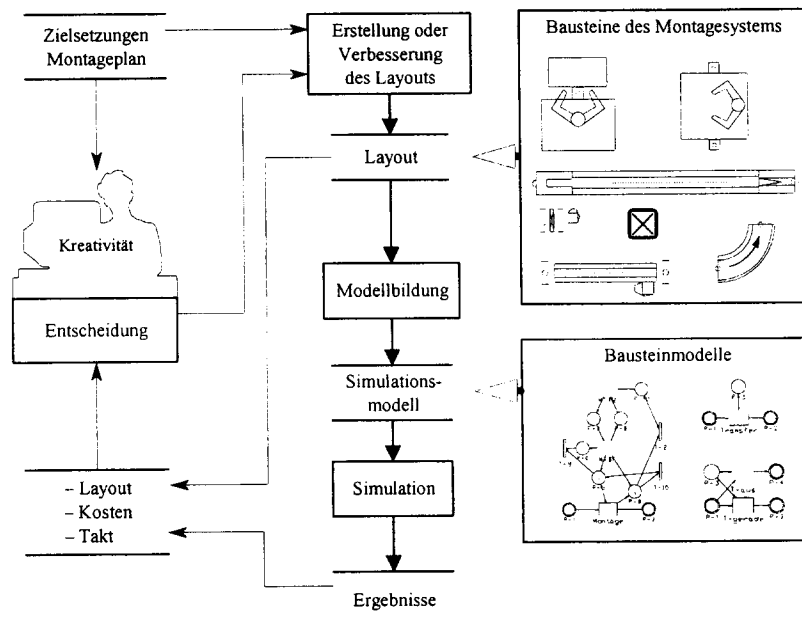


Abb. 6.: Konzept LASIMCO

Basierend auf den Anforderungen an benutzerfreundliche und effiziente Montageplanungssysteme und der Simulation mit EPN (PN-EDIT und PN_EXE) wurde LASIMCO – ein Programmpaket für den Entwurf von Montagesysteme – entworfen. Aufgrund der Typenvielfalt von Montagesystemen und der damit verbundenen Komplexität beschränkt sich LASIMCO auf Montagesysteme mit Palettentransfersystemen und auf die Bewertung alternativer Varianten von Mehrplatzmontagesystemen.

LASIMCO besteht aus einem:

- Programm zur interaktiven Layouterstellung,
- Programm zur automatischen Erstellung eines Simulationsmodells,
- Programm zur Initialisierung und zur Durchführung eines Simulationsexperimentes.

Alle Programme sind in AutoCAD integriert und alle erforderlichen Daten sind als AutoCAD Zeichnung gespeichert.

Das Layoutentwurfssystem dient zur Gestaltung des Montagesystemlayouts in einer graphischen Umgebung. Ein Layout setzt sich dabei aus Montagestationen und Transferelementen zusammen, die durch eine graphische Darstellung und Datenstruktur definiert sind. Die Objekte können beliebig in das Layout eingefügt und miteinander verbunden werden, um ein Montagesystem zu entwerfen.

Ein Layout besteht somit aus zwei Teilen:

Auswahl der Layoutelemente

Verkettung der Layoutelemente

Zur weiteren automatisierten Modellerstellung sind die für die vorhandenen Bausteine und Verbindungen notwendigen Grundmodelle mit EPN bereits vorhanden. Im folgenden Schritt werden die im Layout vorhandenen Daten durch das implementierte Modellbildungssystem im EPN Modell automatisch umgewandelt. Alle Simulationsparameter werden aus den im Layout verfügbaren Daten automatisch berechnet. Der Planer kann daher mit der Simulation sofort nach der Layouterstellung beginnen.

Wurde das Simulationsmodell entwickelt, müssen noch der Anfangszustand, die Dauer der Experimente und die Messziele bestimmt werden. Anschließend kann mit der eigentlichen Simulation begonnen werden. Für die Initialisierung und zur Durchführung eines Simulationsexperimentes wird Programm PN_EXE verwendet.

3.4 Beispiel

Als Beispiel dient die Entwicklung eines Montagesystems für die Montage der Scharniere eines Motorraumdeckels. Während der Planung sollten verschiedene Varianten des Montagesystems in verschiedenen Situationen untersucht werden um die vom Auftraggeber definierten Anforderungen zu erfüllen. Nach dem Entwurf der Montageanlage und anhand des Resultats der FMEA Analyse des Produktes und des Montagesprozesses wurde mit dem Programm LASIMCO das Layout des geplanten Montagessystems erstellt und erste Simulationen durchgeführt. Der Planer ohne Simulationskenntnisse und Erfahrungen benötigte dazu nur einige Stunden (Abb. 7).

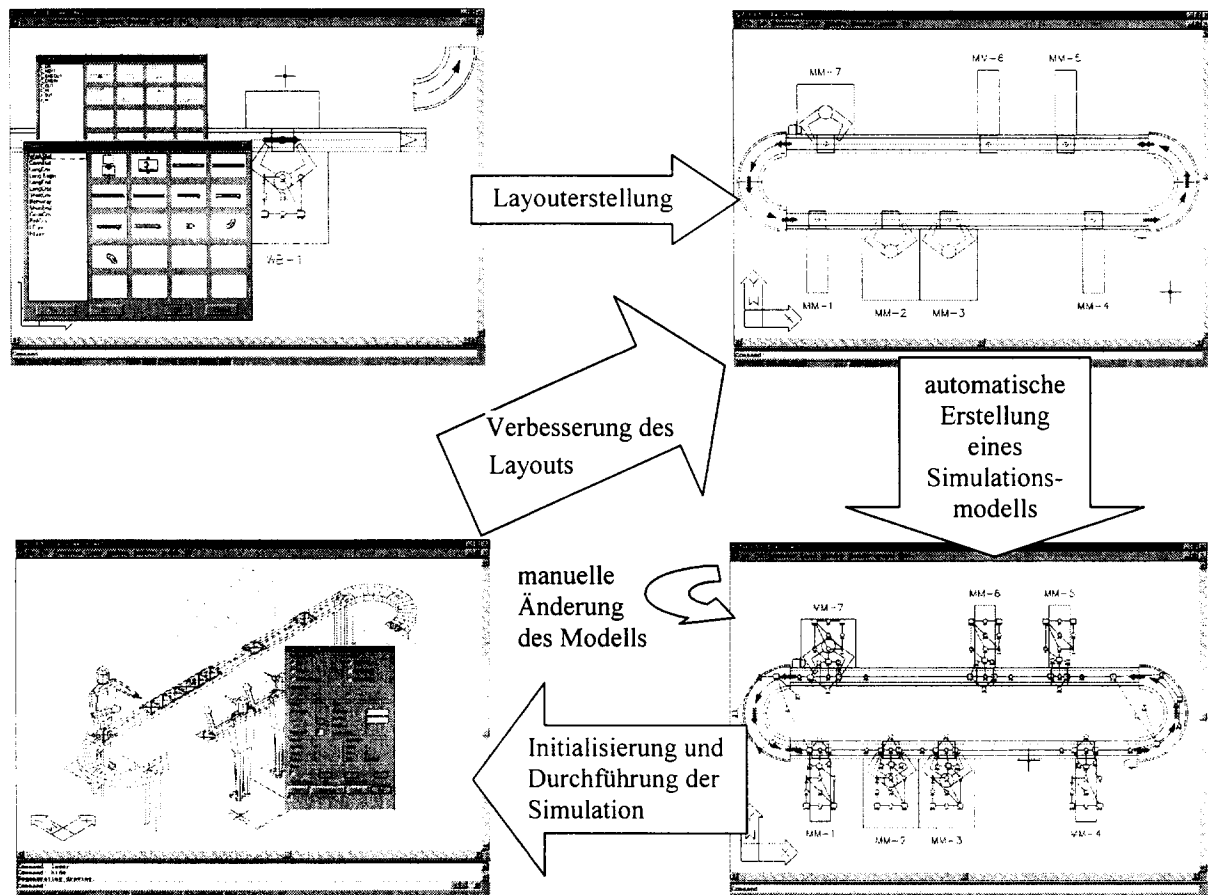


Abb. 7.: Beispiel eines Layoutentwurfs und der Simulation mit LASIMCO

Auf Grund der ersten Simulationsergebnisse wurde das Layout dementsprechend verbessert (Abb. 7). Um alle erforderliche Änderungen einzubeziehen musste der Planer manuell das Modell mit Hilfe des Programmes PN_EDIT ergänzen. Mit den derzeit verfügbaren Simulationsprogrammen wäre das für den Planer eine schwere und zeitaufwändige Aufgabe, die sich aber mit PN_EDIT schnell und effizient lösen lässt.

Auf diese Weise wurde Layout des Montagesystems vervollständigt und ein entsprechendes Simulationsmodell generiert. Der Planer führte noch verschiedene Simulationen mit unterschiedlichen Parametern aus. Die Ergebnisse jeder Simulation wurden automatisch dargestellt, ausgewertet und gespeichert.

Während der Simulation, d.h. „on-line“, kann der Planer die Ergebnisse mittels Animation laufend verfolgen und so Planungs- oder Modellierungsfehler rechtzeitig erkennen. Bei Bedarf kann auch eine „realistische“ 3D Animation generiert werden, die später „off-line“ wiedergegeben werden kann und eine bildhafte Darstellung von Planungsergebnissen bietet.

4 Zusammenfassung

Die entwickelten Programme PN_EDIT, PN_EXE und LASIMCO wurden an Hand einiger industrieller Anwendungen getestet. Die daraus gewonnenen Erkenntnisse sind:

- Die vorgeschlagenen Erweiterungen der Petri-Netz-Theorie wurden mit dem Modellierungs- und Simulationsprogrammen (PN_EDIT und PN_EXE) erfolgreich realisiert. Damit ist mittels des EPN-Modells die Modellierung und Simulation eines Montagesystems in 2D möglich. Die praktische Anwendung zeigte, dass darüber hinaus eine 3D Modellierung und Simulation vorteilhaft sind.
- Die Layouterstellung durch vordefinierte und parametrisierte Bausteine (Baueinheiten) erlaubt ein schnelles, effizientes und benutzerfreundliches Arbeiten. Mit der in Auto CAD verfügbaren 3D-Modellierung kann das Layout mit eigenen Objekten vervollständigt werden. Die Bausteinbibliothek kann jederzeit durch weitere Betriebsmittel ergänzt werden, wobei der zukünftige Benutzer seine spezifischen Anforderungen selbst definieren kann.
- Die automatische Modellerstellung erwies sich für praktische Anwendungen als unerlässlich. Dies führt zu einer erheblichen Zeitersparnis und der Planer kann sich auf konzeptionelle Aufgaben konzentrieren.
- Der Algorithmus zur Simulation der physischen Eigenschaften arbeitet effizient. Die Simulationsdauer ist akzeptabel - sie ist linear von der Anzahl der beweglichen Objekte abhängig. Der Prototyp des Programms wurde zwar nur für Bewegungen in 2D entwickelt, er kann aber einfach auch für die Simulation von 3D Bewegungen erweitert werden.
- Die Animation ist in diesem Prototyp-Simulationsprogramm die einzige Schwachstelle. Die 2D Animation wird zur Validierung von Simulationsmodellen und zur Darstellung von Montageabläufen in Auto CAD erfolgreich verwendet. Eine 3D on-line-Animation wird aber nur mit Drahtmodellen der beweglichen sowie nichtbeweglichen Objekte des Layouts – und das auch nur für eine begrenzte Anzahl von beweglichen Objekten – real-time ausgeführt. Eine realistische 3D Animation ist derzeit nur off-line möglich.

5 Zusammenfassung und Ausblick

Ein Simulationsexperte muss das verwendete Simulationssoftwarepaket kennen um ein System richtig nachbilden zu können. Das gilt besonders für die Modellierung und Simulation mit Petri-Netzen. Die vorgeschlagenen Erweiterungen der Petri-Netze [3] wurden zwar aus mathematischen Grundlagen des GAM Modells abgeleitet, manche Definitionen insbesondere über die Art der Modellierung könnten ohne „Schaltalgorithmen“ nicht vervollständigt werden. Deshalb wurde der modifizierte „Schaltalgorithmus“ mit einigen Erklärungen zu der Modellierung von Palettenmontagesysteme mit EPN beschrieben.

Die weitere Entwicklung geht in zwei Richtungen. Einerseits soll das Programm LASIMCO für eine uneingeschränkte Anwendung im Rahmen „virtuelle Montage“ vervollständigt

werden. Dabei sind insbesondere die Simulation von 3D Bewegungen und die Integration des Produktmodells mit EPN Modellen weiter zu entwickeln. Erweiterungen der Simulation mit EPN auf andere Montage-, Produktions- und Logistiksysteme sind dringend erforderlich [9].

6 Literaturverzeichnis

- [1] ISO 10314, Industrial automation – Shop floor production – Part 1.: *Reference model for standardisation and a methodology for identification of requirements*.
- [2] Iwata K., Onosato M., Teramoto K., Osaki S.: A Modelling and Simulation Architecture for Virtual Manufacturing Systems, In *Annals of the CIRP*, Vol. 44/1/ 1995, S. 399–402.
- [3] Perme, T., Noe, D.: Erweiterte Petrinetze zur intelligenten Planung von Produktionssystemen. *Int .Journal. Automation Austria*, 2002, 10/2, S. 111-124.
- [4] Kopacek P., Kronreif G. and Perme T.: Simulation within CAD-Environment, *Modeling Manufacturing Systems: From Aggregate Planning to Real-Time Control*", (P. Brandimarte, A. Villa, Eds.), Springer-Verlag, S. 115–137.
- [5] Perme, T.: *Ein neues Simulationskonzept zur rechnergestützten Planung von Montagesystemen*, Dissertation, Wien, 1999.
- [6] N.N.: *VDI-Richtlinie 3633; Simulation von Logistik-, Materialfluß- und Produktionssystemen*, VDI-Verlag, Düsseldorf, 1989
- [7] Banks J.: Output Analysis Capabilities of Simulation Software, In *Simulation*, February 1996, Seiten 23–30.
- [8] 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.
- [9] Perme, T., Noe, D.: Computer Aided Planning of Manufacturing Systems Using LASIMCO Simulation Package, In *proceeding of 2000 International CIRP Design Seminar* (Editor Shpitalni, M.), Haifa, Israel 2000.

Lyapunov based approach to Adaptive Fuzzy Control

G. Jambrich

Vienna University of Technology, Automation and Control Institute

Gusshausstrasse 27-29/376

A-1040 Vienna / Austria

Phone: +43 1 58801 37622, Fax: +43 1 58801 37699

E-mail: jambrich@acin.tuwien.ac.at

Received November 17, 2003

Abstract: *The aim of this paper is to state an approach which can be used to stabilize a closed loop Fuzzy control system by off-line tuning an adaptive fuzzy controller such, that due to applying special control Lyapunov functions a pre-specified Lyapunov derivative of the closed loop system is guaranteed. This hybrid control strategy combines the advantages of a stability criterion for global, asymptotic state stability in the sense of Lyapunov with the from a viewpoint of linguistic interpretability as well as adaptation abilities attractive Fuzzy controls. The fields of applications, which this paper is addressed to, are control of continuous time SISO and MIMO systems, with a main focus on non-linear time invariant plants under control.*

1 Introduction

These days Fuzzy systems are international by and large accepted and the applications of such systems cover a wide range of areas from fuzzy modelling and identification, over fuzzy control to computational intelligence methods based on fuzzy logic and so on. Even if the big boom of investigations in the field of fuzzy logic is actually rather abated, this theory has kept its importance in areas where e.g. implementation of human knowledge in automation systems, model uncertainties and interpretability of measured data are to be the centre of attention.

If we now look at closed loop control concepts in general, also including Fuzzy control, stability issues have a high priority concerning the design of controls and are therefore an important indicator of performance. The classical linear control theory offers a lot of practical proven tools and methods in time and frequency domain, which are relatively easy to understand even if somebody is not so familiar with the scientific background. But if we want to apply Fuzzy control, most of these classical tools become invalid because of the overall non-linear behaviour of fuzzy systems. In the control literature several general non-linear approaches can be found and they may roughly be structured and summarised as follows (Marquez, H. J., 2003): Lyapunov stability, feedback stabilization, input to output stability, input to state stability, passivity theory, dissipativity theory and feedback linearization.

From these listed main groups of non-linear stability theories, related to the Lyapunov stability theory, a special kind of control Lyapunov functions, described in (Weinmann, A., 2001), was picked up in this paper. There the rather uncomfortable search for suitable Lyapunov functions and the conservativity of the original Lyapunov theory concerning interpreting the results if the stability proof fails is relaxed by pre-specifying a certain negative Lyapunov derivative. The goal of this method is, to define a vector of control variables which can be fed into the plant to get the desired global and asymptotic stable

behaviour of the states. Nonlinearities are not treated separately and not be compensated for. The control algorithm is compact and simple from an analytical viewpoint, even it could demand a certain computational effort because of dealing with nonlinearities. The necessary conditions to apply this algorithm are the existence of mathematical differential or difference equations describing the time invariant plant, as well as the knowledge of initial conditions respectively measured values of the states at each time step of the complete control- and observable system. Furthermore the user should be informed, that there can eventually appear some problems due to a denominator inner product zero during calculation of the control variable if SISO systems are under consideration. To cope with this inconvenience for a constraint operating range a simple solution from an engineering point of view is given later by the author of the actual paper. How to successful avoid the denominator inner product zero concerning MIMO systems was already mentioned in (Weinmann, A., 2001).

By numeric calculating or simulating the closed loop control system in time domain, using the control algorithm explained above, a set of data for the fuzzy controller inputs and outputs can be created, which is used in the next step to generate and adapt an adaptive fuzzy controller. The main reasons why fuzzy techniques are applied in this paper are the inherent linguistic interpretability of the fuzzy rule structure, good local approximation properties in case of modelling non-linear functions in combination with few input dimensions and the capability to use generic learning algorithms for generating the fuzzy inference system and adjusting the parameter sets in order to archive the desired controller input-output mapping. The learning methods can in principle be subdivided in two groups: batch (off-line) learning and pattern (on-line) learning algorithms. For dealing with time variant systems, indicated by changing characteristics, the on-line methods should be preferred.

However, in the literature a lot of different approaches exist for generating and adapting fuzzy inference systems from given data. Here we investigate the hybrid combination of Lyapunov stability and adaptive fuzzy control based on two adaptation methods: "ANFIS" (Jang, J.-S. R., 1993) and "genfis2" (Chiu, S. L., 1994) which are both supported by the MATLAB® Fuzzy Logic Toolbox. The ANFIS (Adaptive- Network-based Fuzzy Inference System) approach defines a single output Takagi-Sugeno-type fuzzy inference system implemented in a five layer neuronal network. There the parameters of a fuzzy set are tuned by a hybrid learn algorithm, which combines the advantages of gradient descent and (global) linear least squares estimation techniques. To extend this method for MIMO Systems several ANFIS models could be switched in parallel at input side. The second method genfis2 represents a generic approach which generates a MIMO Sugeno-type fuzzy inference system first using fuzzy subtractive clustering to determine a minimum number of rules and the antecedent membership functions and then applying linear least squares estimation to get each rules consequent equations. The two methods ANFIS and genfis2 are compared concerning model accuracy and complexity as well as computation time, assuming a simple SISO example.

2 Lyapunov stability

2.1 Stability for multiple-input multiple-output systems

Consider a multivariable non-linear, time invariant plant which is affine in u

$$\dot{x} = f(x) + G(x)u, \quad x_0 = x(0), \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m \quad (1)$$

x state variable

x_0 initial state variable

u control variable, here equivalent to system inputs

and a Lyapunov-candidate with a pre-specified negative Lyapunov-derivative

$$V(x) = \frac{1}{2}(x - x_f)^T P (x - x_f) > 0, \quad P \stackrel{\Delta}{=} P^T > 0, \quad x_f = \text{const.} \quad (2)$$

x_f final state variable

$$\Rightarrow \dot{V}(x) = (x - x_f)^T P \dot{x} = (x - x_f)^T P (f + G u) \stackrel{\Delta}{=} - (x - x_f)^T Q (x - x_f) < 0, \quad Q \stackrel{\Delta}{>} 0 \quad (3)$$

$$b_1^T \stackrel{\Delta}{=} (x - x_f)^T P G \quad (4)$$

$$\Rightarrow b_1^T u = -q, \quad q = (x - x_f)^T [P f + Q (x - x_f)] \quad (5)$$

Fig. 2.1 visualize the closed loop control scheme and Fig. 2.2 the control law which leads to the demanded control variable u .

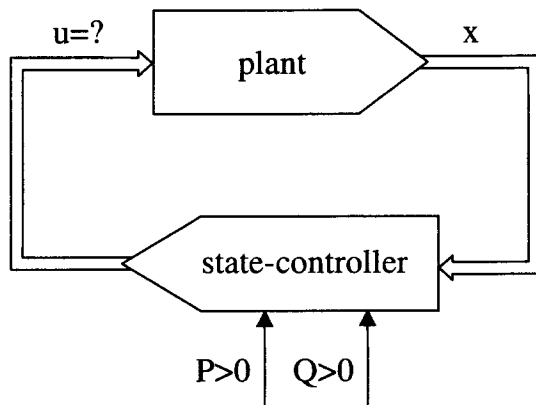


Fig. 2.1: closed loop control circuit

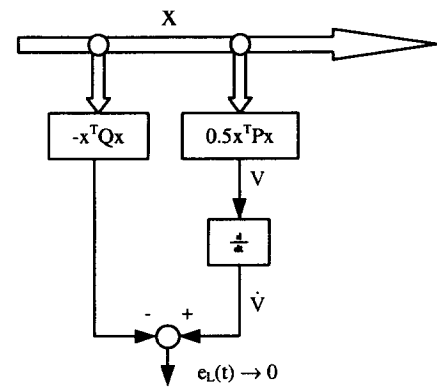


Fig. 2.2: control law for Lyapunov stability

Optimisation subject to conditions

$$\frac{1}{2} u^T R u + \mu (b_1^T u + q) \rightarrow \min_u, \quad R \stackrel{\Delta}{>} 0 \quad (6)$$

if R is not singular:

$$R u + \mu b_1 = 0 \quad \Rightarrow \quad u = -\mu R^{-1} b_1 \quad (7)$$

$$\boxed{u = -\frac{(x - x_f)^T [P f + Q(x - x_f)]}{(x - x_f)^T P G R^{-1} G^T P (x - x_f)} R^{-1} G^T P (x - x_f)}, \quad (8)$$

$$P G R^{-1} G^T P > 0$$

$$\Rightarrow (x - x_f)^T P G R^{-1} G^T P (x - x_f) > 0 \quad \forall x \neq x_f \quad (9)$$

Assessing the Norm of u

$$F_x(x - x_f) \triangleq f(x) \quad (10)$$

$$\eta = \frac{\|u\|_F}{\|x - x_f\|_F} \leq \max_x \{ \sigma_{\max}[P F_x + Q] \times \sigma_{\max}[R^{-1} G^T P] \} \times \left[\min_x \{ \sigma_{\min}[P G R^{-1} G^T P] \} \right]^{-1} \quad (11)$$

η gain factor

Concerning $x - x_f \rightarrow 0$, ($u \rightarrow \frac{0}{0}$)

$$\text{if } \|x - x_f\|_F < \varepsilon, \text{ e.g. } \varepsilon = 10^{-6}: \text{ switch to } \boxed{u \equiv u_f = \text{const.}} \quad (12)$$

u_f final control variable

Linear MIMO plants: assume $f(x) \triangleq A(x - x_f)$, $G(x) \triangleq B \Rightarrow$

$$\boxed{u = -\frac{(x - x_f)^T (P A + Q)(x - x_f)}{(x - x_f)^T P B R^{-1} B^T P (x - x_f)} R^{-1} B^T P (x - x_f)} \quad (13)$$

$$\eta = \frac{\|u\|_F}{\|x - x_f\|_F} \leq \frac{\sigma_{\max}[P A + Q] \sigma_{\max}[R^{-1} B^T P]}{\sigma_{\min}[P B R^{-1} B^T P]} \quad (14)$$

2.2 Stability for single-input single-output systems

Consider a single input, single output non-linear, time invariant plant which is affine in u

$$\dot{x} = f(x) + g(x)u, \quad x_0 = x(0), x \in \mathbb{R}^n, u \in \mathbb{R}^1 \quad (15)$$

and again a Lyapunov-candidate with a pre-specified negative Lyapunov-derivative

$$V(x) = \frac{1}{2}(x - x_f)^T P (x - x_f) > 0, \quad P = P^T > 0, x_f = \text{const.} \quad (16)$$

$$\Rightarrow \dot{V}(x) = (x - x_f)^T P \dot{x} = (x - x_f)^T P (f + g u) \stackrel{\Delta}{=} - (x - x_f)^T Q (x - x_f) < 0, \quad Q \stackrel{\Delta}{>} 0 \quad (17)$$

$$u = - \frac{(x - x_f)^T [P f + Q (x - x_f)]}{(x - x_f)^T P g} \quad (18)$$

Looking at $x - x_f \rightarrow 0$, refer to Eq. (12)

Concerning $(x - x_f)^T P g \rightarrow 0$:

- a) denominator zero does not occur
 $\Rightarrow$ this situation can often not be guaranteed for the whole range of possible initial conditions,
- b) hard limits for control variable $u: |u| \leq u_{max}$
 $\Rightarrow$ this could lead to sliding mode and quasi steady-state conditions with $x_f \neq x_{f \text{ desired}}$,
- c) assume a set-point-depending matrix $P = P(x)$ in the Lyapunov-function to avoid denominator zero
 $\Rightarrow$ a simple, analytic solution for any plant can not be stated, derivatives to x concerning the elements of P have to be considered and P has to be positive definite at each time step,
- d) method b) and an additional empirical determined time delay in the feedback of the closed loop control system: $T_V \dot{u}'(t) + u'(t) = V u(t)$
 u limited control variable
 u' delayed control variable
 T_V time constant of delay
 V gain factor
 $\Rightarrow$ easy to realise and for several plants is a stable control of the state variables in a constraint operating range possible. As reason the low pass characteristic of the time delay can be stated.

For the from an engineering viewpoint obvious methods b) and d) stability in the sense of Lyapunov can no longer be guaranteed, but in many cases stable operation is nevertheless possible.

Linear SISO plants: assume $f(x) \triangleq Ax$, $g(x) \triangleq b \Rightarrow$

$$\boxed{u = -\frac{(x - x_f)^T [PAx + Q(x - x_f)]}{(x - x_f)^T Pb}} \quad (19)$$

Including the reference input y_{ref} slightly different system equations can be considered

$$\dot{x} = Ax + bu + bv y_{ref}, \quad x_0 = x(0), \quad x \in \mathfrak{R}^n, \quad \{u, v, y_{ref}\} \in \mathfrak{R}^1 \quad (20)$$

For steady-state with $\dot{x}_f = 0$, $u = u_f$ follows x_f , if A is not singular, to

$$x_f = -A^{-1}b(u_f + v y_{ref}) \Rightarrow \quad (21)$$

$$\boxed{u = -\frac{(x - x_f)^T [P(Ax + bv y_{ref}) + Q(x - x_f)]}{(x - x_f)^T Pb}} \quad (22)$$

Concerning $(x - x_f)^T Pb \rightarrow 0$: see previous statements excluding y_{ref}

3 Design of an adaptive fuzzy controller for closed loop stability

3.1 Definition of a Takagi-Sugeno fuzzy inference system

The starting point is a Takagi-Sugeno (TS) fuzzy inference system (*Lughofer, E., 2003*) whose output is defined by

$$\hat{y} = \sum_{i=1}^M f_i \Phi_i(\hat{u}) \quad (23)$$

where

$$\Phi_i(\hat{u}) = \frac{\mu_i(\hat{u})}{\sum_{j=1}^M \mu_j(\hat{u})} \quad (24)$$

are called basis functions, which normalize the degree of rule fulfilment by using the product t-norm, i.e.

$$\mu_i(\hat{u}) = \prod_{j=1}^p \mu_{ij}(\hat{u}_j) \quad (25)$$

Where $\hat{u}_j$ is the j -th component in the current input data vector, hence reflecting the value of the j -th channel and the μ_{ij} the membership degree of $\hat{u}_j$ to the fuzzy set describing the j -th

premise part of the i -th rule. The f_i 's, don't mix up with $f(x)$ in chapter 2, are the so-called consequent functions of the M rules and are defined by

$$f_i = \omega_{i0} + \omega_{i1}\hat{u}_1 + \omega_{i2}\hat{u}_2 + \dots + \omega_{ip}\hat{u}_p \quad (26)$$

For a Sugeno controller (as a special case of a Takagi-Sugeno controller) only one constant output value per rule, i.e. ω_{i0} is applied. Arbitrary fuzzy sets can be chosen depending on the special task and behaviour of the fuzzy inference system (FIS), most common are B-splines of several orders (e.g. 2nd order or B-splines reflect triangular fuzzy sets), Gaussian functions or sigmoid functions. Fig. 3.1 shows the fuzzy reasoning mechanisms of a Takagi-Sugeno fuzzy inference system.

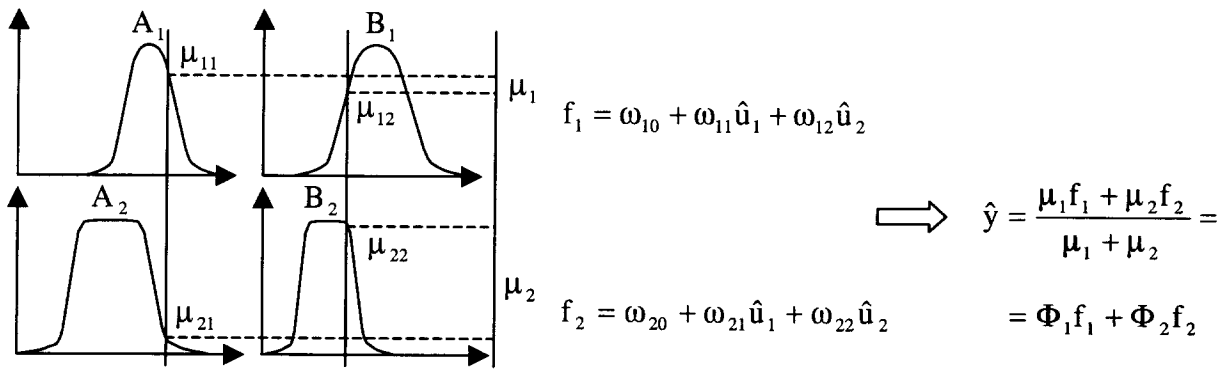


Fig. 3.1: TS-fuzzy inference system - fuzzy reasoning mechanisms

3.2 Fuzzy control concept

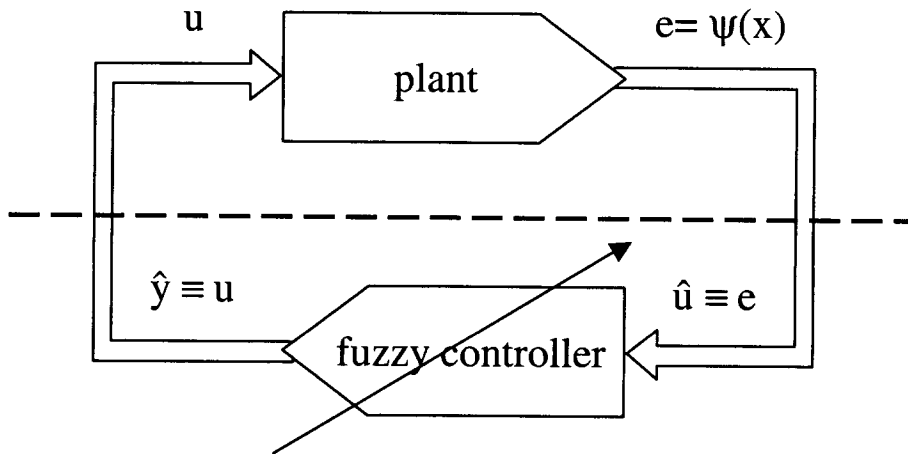


Fig. 3.2: closed loop control circuit including a tuneable fuzzy controller

Assuming the state equations of Eq. (1) for the non-linear MIMO plant and the plant output

$$y = \psi(x) = Cx, \quad y_f = Cx_f, \quad \text{and} \quad y, y_f \in \mathbb{R}^r \quad (27)$$

respectively the control deviation

$$e = y_f - y = C(x_f - x) \quad (28)$$

It is obvious that e has to vanish over time if the control law of Eq. (8) is applied to stabilize the non-linear plant ($x - x_f \rightarrow 0$) and the due to that the control deviation has to be asymptotic stable.

For non-linear SISO plants, represented by the state equations of Eq. (15), the plant output

$$y = \psi(x) = c^T x, \quad y_f = c^T x_f, \quad \text{and } y, y_f \in \mathcal{R}^1 \quad (29)$$

and the control deviation

$$e = y_f - y = c^T(x_f - x) \quad (30)$$

are assumed. It is also obvious that e has to vanish over time if the control law of Eq. (18) can be successful applied (excluding the problems with denominator zero) to stabilize the non-linear plant and the due to that the control deviation has to be asymptotic stable.

By numeric calculating or simulating the closed loop control system in time domain using the Lyapunov-based control algorithms explained in chapter 2, a set of data is created for stabilizing the closed loop control system of Fig. 2.1. This data, describing the desired state controller input (x) and output (u), is used in a next step to generate (if necessary) and adapt an adaptive fuzzy controller, input (e) and output (u) with the aim of stabilizing the closed loop control system shown in Fig. 3.2. Summarized, the main idea which is to be stated in this paper should be the successful combination of Lyapunov stability and fuzzy control with the help of control Lyapunov functions. The initial fuzzy model applied for the design of the controller based on the structure mentioned in chapter 3.1 could either be the result of an already existing or an autonomous generated (e.g. with MATLAB® ANFIS or genfis2) fuzzy model. The basic idea of tuning a fuzzy controller is in general not only restricted to TS-type fuzzy inference systems. But for the sake of presenting comprehensive results, to make use of already existing, proven software tools, which have implemented a TS-type FIS structure like the MATLAB® Fuzzy Logic Toolbox, is favourable. - Furthermore a combination of a different, already existing fuzzy control concept extended with an additional supervisory control based on the major control aspects stated above could be imaginable in order to get a higher performance in stability.

3.3 Adaptation aspects

In principle three kinds of components of a TS-FIS can be adapted

- premise parameters: they appear in the input membership functions as centres and widths,

- consequent parameters: they appear in the rule consequents as output weights,
- rule structure.

The consequent parameters are according to the definition linear, the premise parameters are non-linear ones. The consequent parameters can be estimated with recursive least squares (RLS, global adaptation, *Lughofer E. and Groissboeck TR-212/2002*) or recursive weighted least squares (RWLS, local adaptation, *Lughofer, E. and Groissboeck TR-213/2002*) methods. In order to forget older data points a forgetting factor can be incorporated to meet online-learning requirements (refer to Chapter 3.4). Seen from the mathematical point of view both groups are exact methods.

The adaptation of the premise parameters concerning an already existing FIS-structure is a little bit more sophisticated, because they are in general non-linear (e.g. in case of using Gaussian membership functions) and the applied optimisation methods which are minimizing a general loss function turn out as a so called non-linear least squares problem. There exist several approaches which can efficiently solve that problem with an appropriate gradient-based optimisation method like steepest descent, Newton's method or the quasi Newton approach. But all these approaches are iterative methods and due to that they can never be exact. As a consequence, beside the problems concerning convergence time, trapping in local minima, etc., the difficulty of choosing the right point for terminating the iterative estimation algorithm appears, which has a strong influence on the degree of model adaptation from the old to the new data blocks. The rate of forgetting old data has to be controlled by an appropriate termination criteria, otherwise the old data points will be completely forgotten, even if choosing the premise parameters of the old FIS as starting point for the new iteration.

If the number of antecedent membership functions and the rule structure is not known at the beginning of the fuzzy controller identification process or was not defined until now, generic algorithms, e.g. *genfis2*, can be applied to create a TS-FIS model. This new model can be used then as a final result or to initialise other refinement algorithms like ANFIS. Optimising the rule structure of a TS-FIS can be an important task, because it allows one to determine the optimal complexity of the FIS in order to achieve a better linguistic interpretability and by that it weakens the curse of dimensionality. Methods such as modified linear subset selections schemes (e.g. OLS) or non-linear global search techniques (e.g. genetic algorithms) exist for rule structure optimisation, for details see (*Nelles, O., 2000*). A fast online adaptation strategy for rule structures is not dealt in this paper but can be found in (*Wu, S., Er, and Gao, 2001*). And by the way, if applying *genfis2* for generating the initial model, mostly the resulting model posses a rule structure with only a handful of rules, such that rule structure optimisation is not the crucial point for adaptation.

3.4 Learning strategies

Actually there are two learning paradigms for adaptive networks. Using a batch learning (or off-line learning) algorithm, the update of the parameter vector takes place only after the whole training data set has been presented, i.e., only after each epoch or sweep. On the other hand, if the parameters are updated immediately after each input-output pair has been presented the pattern learning (or on-line learning) is resulting. The latter learning paradigm is vital to the on-line parameter identification for systems with changing characteristics. If dealing with time-varying characteristics of the incoming data we need to decay the effects of old data pairs if new data pairs become available. But on the other hand the already learned parameters related to the old data should not completely be forgotten. One simple method

concerning the adaptation of the consequent parameters of a FIS is the recursive weighted least square estimate (RWLS) with incorporation of a forgetting factor which gives higher weighting factors to more recent data pairs (*Lughofer, E. and Groissboeck TR-213/2002*).

4 Adaptation methods

In the literature a lot of methods exist for generating and adapting FIS from given data. Namely for instance with

- ANFIS: Addaptive Network-based Fuzzy Inference System,
- genfis2: cluster based generation of FIS (Fuzzy Inference Systems),
- RENO: REgularized Numerical Optimisation (*Bodendorfer, U., Burger and Haslinger, 2001*)
- LOLIMOT: Local Linear Model Tree algorithm (*Nelles, O., and Isermann, 1996*),

From these listed methods the first ones are taken for further investigation concerning the creation of a hybrid controller build up on fuzzy and Lyapunov techniques. The reason why these methods was paid more attention is because they are part of the well known MATLAB[®] Fuzzy Logic Toolbox and therefore a suitable starting point for the presentation of a new idea.

4.1 ANFIS – Adaptive Network-based Fuzzy Inference System

The ANFIS approach creates a Takagi-Sugeno-type fuzzy inference system (Fig. 3.1) through a learning method which works similar to that of neuronal networks by defining five layers (Fig. 4.1):

- Layer 1: fuzzification of the input values due to membership functions -> membership degrees,
- Layer 2: aggregation of membership degrees due to an appropriate t-norm applied in the premise parts,
- Layer 3: evaluation of the basis functions by normalisation of aggregated membership degrees,
- Layer 4: weighting of basis functions with linear (Takagi-Sugeno system) or constant (Sugeno-type) consequent functions,
- Layer 5: evaluation of output values by applying equation (23).

Constraints:

MATLAB[®] ANFIS only supports Sugeno-type systems and these must be:

- first or zeroth order Sugeno-type systems, with
- single output, obtained using weighted average defuzzification (linear or constant output membership functions) ,
- a of unity weight for each rule and
- only the provided membership and defuzzification functions can be used.

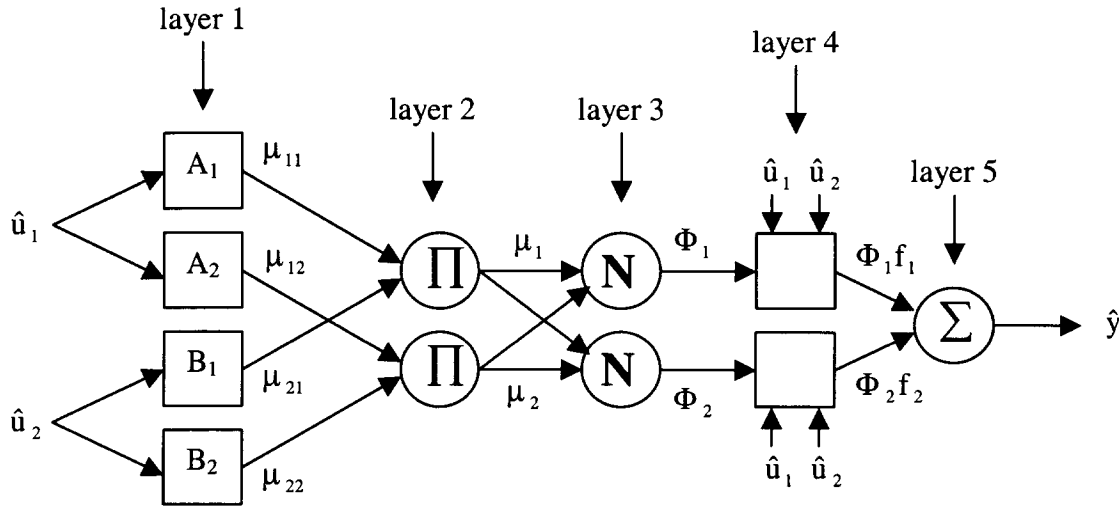


Fig. 4.1: ANFIS – 5 layer structure

There are two ways of initialise the FIS: Either only the data of input and output pairs is presented the algorithm or in addition to the data a self defined FIS is passed to ANFIS, considering the constraints above. In the first situation ANFIS uses the function “genfis1” to create a default Takagi-Sugeno-type FIS by using grid partition.

The basic approach for adapt and train the initialised FIS is denoted by a hybrid method, e.g. a combination of a gradient decent method and least square estimation. While firstly the non-linear parameters are fixed and the linear parameters in the consequent (or constant) functions are estimated with a conventional global least square method (see *Lughofer, E. and Groissboeck TR-212/2002*), afterwards the non-linear parameters are calculated by a special gradient decent approach as described in (*Jang, J.-S. R., 1993*). For further details see also (*Jang, J.-S. R., 1993*) and (*Jang, J.-S. R., 1992*).

4.2 genfis2 – Fuzzy subtractive clustering

Genfis2 performs cluster analysis in order to generate fuzzy sets due to projection of clusters (2-D: circles) in the multidimensional space onto the corresponding axis (see Fig. 4.2). For cluster analysis itself different methods can be applied, which result in different outlook and properties of the fuzzy sets. The rule extraction method of genfis2 accomplishes this by extracting a set of rule that models the data behaviour. It first uses the subtractive clustering algorithm to determine the number of rules and the antecedent membership functions and then uses global linear least square estimation to determine each rules consequent equations. It returns a Takagi-Sugeno-type fuzzy inference system structure that contains as set of fuzzy rules to cover the feature space. The subtractive clustering is managed by the function “subclust” which operates by finding the optimal data point to define a cluster centre, based on the density of surrounding data points. All data points within the pre-specified distance RAD of this point are then removed, in order to determine the next data cluster and its centre. This process is repeated until all of the data is within the distance RAD of a cluster centre. For a detailed inspection of genfis2 algorithm refer to *Chiu, S. L., 1994* respectively *Yager, R. R. and Filev, 1994*.

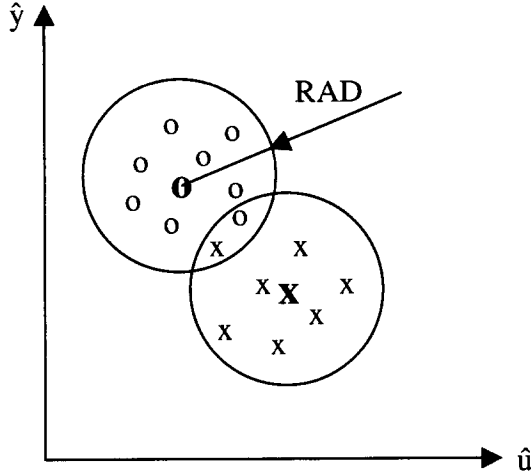


Fig. 4.2: cluster analysis scheme

5 Comparison ANFIS / genfis2

The results of this comparison do not allow to make conclusions about the performance of the investigated programs for systems with higher dimensionality or more complex rule structures, but show the experiences which were made by investigating the following SISO-example. The main task was to practise a lot of learning strategies in order to find the optimal solution for the this example applying the control strategy mentioned in chapter 3.2.

5.1 Simulation-example (non-linear continuous SISO system):

Assumptions for state equations , the matrixes P and Q

$$f(x) = \begin{pmatrix} x_2 \\ -x_1 - 0.25x_2 + 0.2x_1^2 \end{pmatrix}, \quad g(x) = \begin{pmatrix} 0.04x_1^2 + 1 \\ 0.1x_1^2 + 2 \end{pmatrix} \quad (31)$$

$$x(0) = \begin{pmatrix} 2 \\ -4 \end{pmatrix}, \quad x_f = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad P = \begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix}, \quad Q = I_2 \quad (32)$$

the system output

$$y = c^T x, \quad y_{ref} = c^T x_f = 0, \quad c = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (33)$$

and the control deviation

$$e = y_{ref} - y = -c^T x \quad (34)$$

Method d), refer to chapter 2.2 , is applied for limitation of the control variable u and to avoid sliding mode (Fig. 5.1):

- state controller block (s-function) according to Eq. (18),
- controller output limitation: $u_{\max}=\pm 1$
- PT₁-element in the feedback loop: $V=2$, $T_V=0.01s$ (located after u-limitation element)

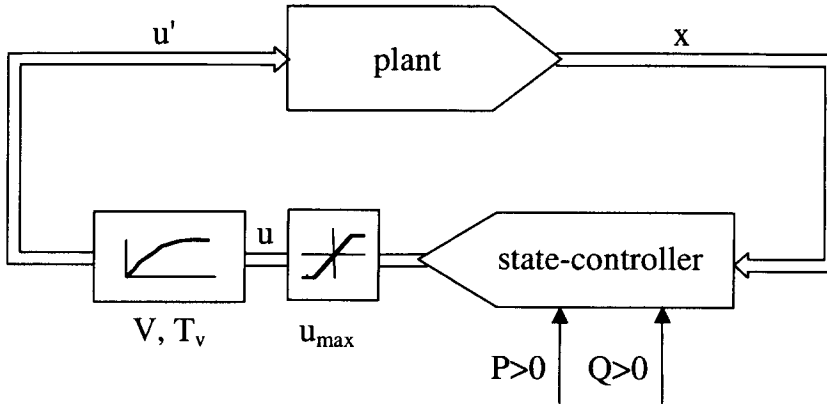


Fig. 5.1: Principle simulated closed loop control circuit

Simulation results, Fig. 5.2 - Fig. 5.4 (MATLAB® 6.1 SIMULINK):

OPTIONS = SIMSET('solver','ode45','RelTol',1e-6,'AbsTol',1e-6,'Refine',10);

We use the simulation data-matrix $Data_{SIM}$ to describe the desired fuzzy controller input/output behaviour (refer to chapter 3.2)

$$Data_{SIM} = (t \quad e \quad u'); \quad t, e, u \in \mathbb{R}^{nd \times 1} \quad (35)$$

e.....fuzzy controller input

u'.....fuzzy controller output

t.....time-vector

nd....number of data-triples (in this example nd=5651)

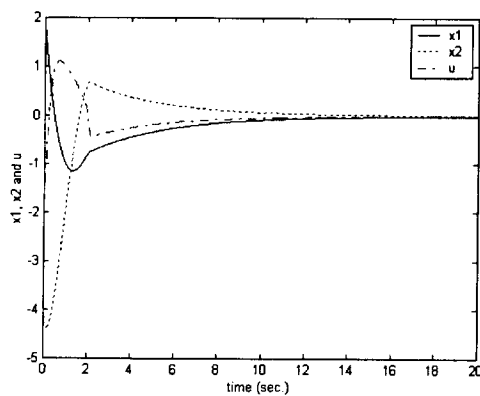


Fig. 5.2: state variables and control variable versus time

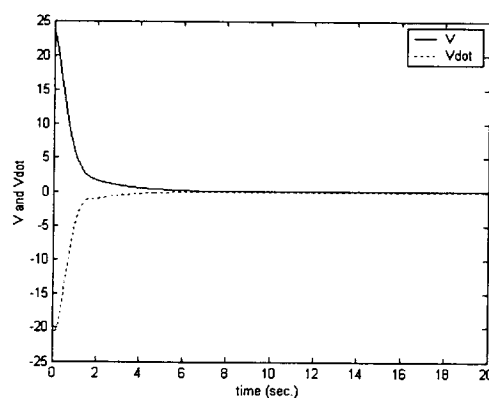


Fig. 5.3: Lyapunov-function and its derivative versus time

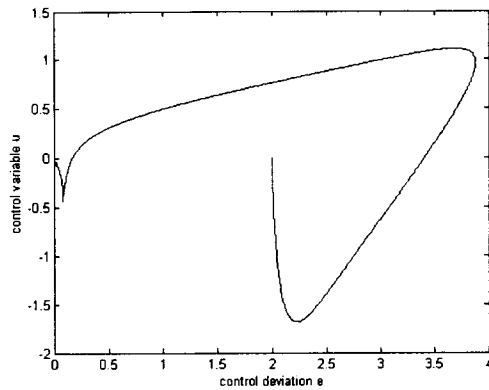


Fig. 5.4: phase plot control variable versus control deviation

5.2 Batch learning with ANFIS

For a better approximation of the simulated data first `genfis2` (cluster radius $RADII=0.075$) is used to initialise the start FIS-structure. It creates a FIS-system with 2 Gauss membership functions, two linear output membership functions and 2 rules with all together 8 tuneable parameters. The function `genfis1`, which is automatically chosen from ANFIS if no start-FIS is specified, provides in this case not such good results. In a second step ANFIS shall now refine the input-output mapping, including following specifications:

- training options: `TRNOPT=[100 0.01 0.01 0.9 1.1]`; (max. 100 epochs, training error goal 0.01,...)
- training method: `OPTMETHOD=1`; (hybrid algorithm with gradient descent and LSE)

results:

σ_1	c_1	σ_2	c_2
0.0257	0.0774	0.1326	0.0717
r_1	s_1	r_2	s_2
-8.2355	0.0038	0.1577	-0.0182

Tab. 5.1: ANFIS batch-learning, values of input membership function - and consequent function-parameters

Description Gauss membership functions (input), Tab. 5.1:

$$\mu(i) = \text{GAUSSMF}(\hat{u}, [\sigma(i), c(i)]) = \text{EXP}(-(\hat{u} - c(i))^2 / (2 * \sigma(i)^2)); \quad (36)$$

$\mu(i)$ membership degree, Gaussian function (i)

$\sigma(i)$ width of Gaussian function (i)

$c(i)$centre of Gaussian function (i)

Description linear consequent functions (output), Tab. 5.1:

$$f(i) = r(i) * \hat{u} + s(i); \quad (37)$$

$f(i)$consequent function (i)
 $r(i), s(i)$linear output parameters to $f(i)$

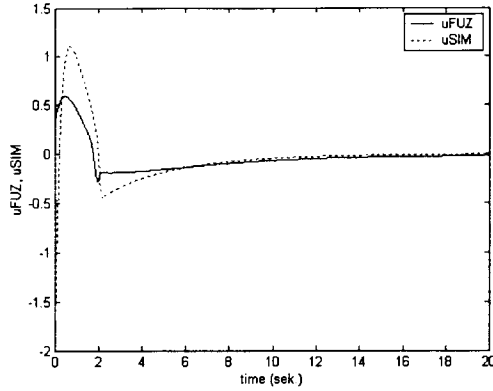


Fig. 5.5: ANFIS batch-learning, control variable:
 comparison fuzzy model (uFUZ) with simulation (uSIM)

training-model error(root-mean-square-error RMSE)

$$\text{trnRMSE} = \text{NORM}(u\text{FUZ} - u\text{SIM}) / \text{SQRT}(\text{LENGTH}(u\text{FUZ})) = 0.29883; \quad (38)$$

Fig. 5.5 confirms that the model error trnRMSE is too big for a satisfying approximation of the simulated data set. Increasing complexity due to implementation of more membership functions and rules does not help if the number of input dimensions stays the same and a good performance concerning computation time is assumed. It seems reasonable to reduce the block size of the data for training in order to get better results.

5.3 Generic approach with genfis2 (stand alone)

results:

σ_1	c_1	σ_2	c_2
0.1028	0.01474	0.1028	0.07391
r_1	s_1	r_2	s_2
-5.0790	-0.03875	0.1500	-0.00933

Tab. 5.2: genfis2 batch-learning, values of input membership function - and consequent function-parameters

model error:

$$\text{trnRMSE} = 0.30980$$

(39)

The cluster radius was again chosen to $\text{RADII}=0.075$. The resulting model error is very close to that calculated by ANFIS what leads to a similar worse performance. Varying RADII seems also not to help.

5.4 Batch learning with ANFIS and data splitting

Now the function `genfis1` is used to initialise the start FIS-structure based on the first 50 data points (block size). It creates a again a FIS-system with 2 Gauss membership functions, two linear output membership functions and 2 rules for comparison to the results without data splitting. Assuming the same training options as before, ANFIS is forced to learn in batch mode the input/output behaviour of the hole data by approximating every data block individually. At the end of one block training phase the FIS system with the updated parameter set is fed as initial FIS into the ANFIS training of the next data block and so on. It should be noticed that the MATLAB[®] ANFIS algorithm itself only takes the premise parameters of the updated FIS, the LSE-algorithm is initialised with reseted consequent parameters. After trained all data blocks of the training data set each parameter of the FIS is characterised by a parameter vector with 113 elements.

results:

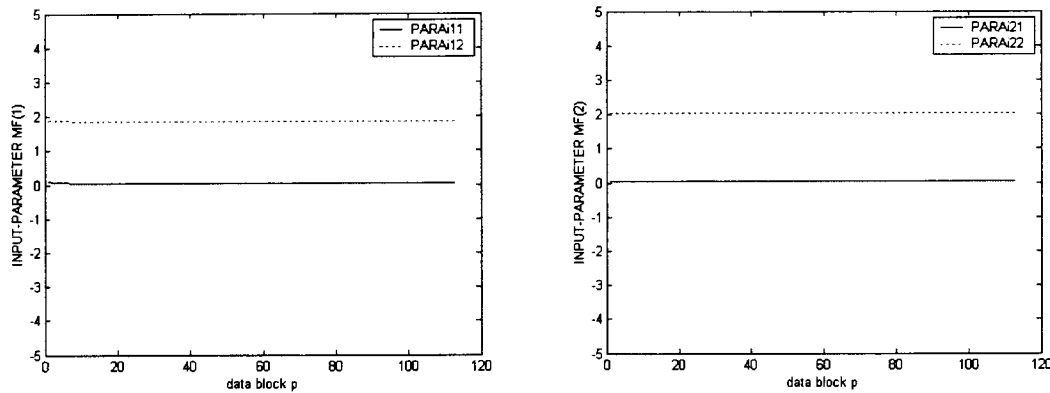


Fig. 5.6: ANFIS data-splitting, input membership-function parameters

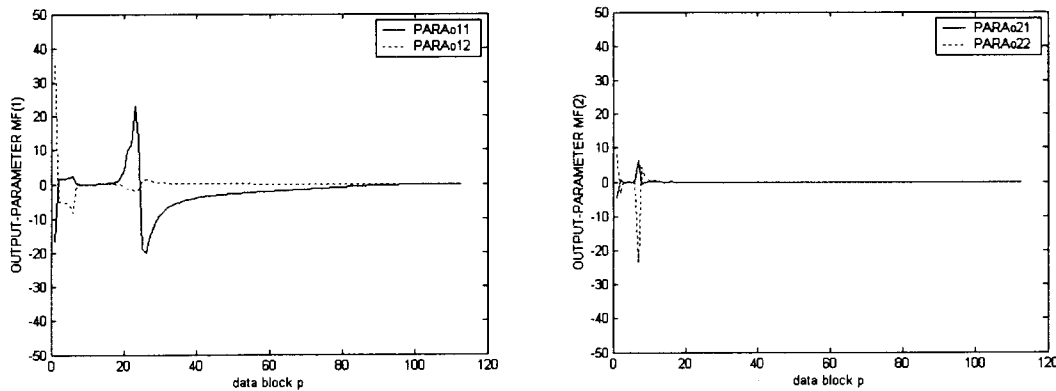


Fig. 5.7: ANFIS data-splitting, consequent-function parameters

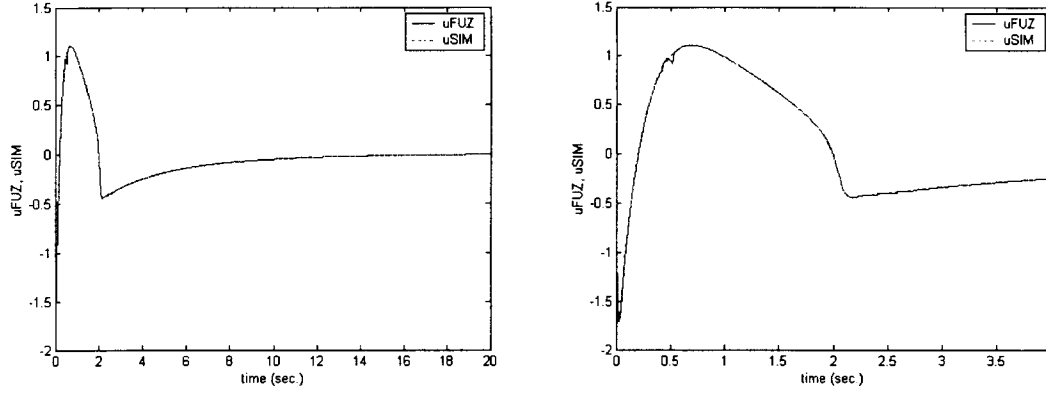


Fig. 5.8: ANFIS data-splitting, control variable:
comparison fuzzy model (uFUZ) with simulation (uSIM), right figure: zoom

Description Gauss membership functions (input), Fig. 5.6:

$$\mu(i) = \text{GAUSSMF}(\hat{u}, [\sigma(i), c(i)]) = \text{EXP}(-(\hat{u} - c(i))^2 / (2 * \sigma(i)^2)); \quad (40)$$

PARAi(i)1 $\equiv$ $\sigma(i)$

PARAi(i)2 $\equiv$ $c(i)$

Description linear consequent functions (output), Fig. 5.7:

$$f(i) = r(i) * \hat{u} + s(i); \quad (41)$$

PARAo(i)1 $\equiv$ $r(i)$

PARAo(i)2 $\equiv$ $s(i)$

model error:

$$\boxed{\text{trnRMSE} = 0.00403} \quad (42)$$

The results (Fig. 5.6 - Fig. 5.8) show that splitting or discrete the data leads to variable FIS parameters which can better approximate the simulated data. The model error was reduced about a factor 100.

5.5 Batch learning with genfis2 and data splitting

To create a FIS-system which consists of 2 Gauss membership functions, two linear output membership functions and 2 rules the cluster radius RADII has to be adapted before each data block approximation. This was realised by a loop which only accepts the new result for the FIS parameters if the FIS matches the above defined structure. If this conditions is false, the cluster radius is reduced stepwise from the value “1” to “0” until its true.

results:

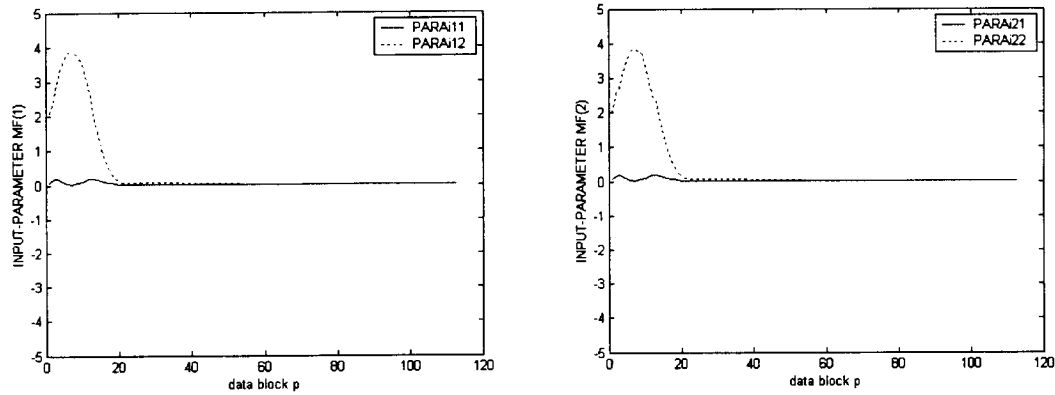


Fig. 5.9: genfis2 data-splitting, input membership-function parameters

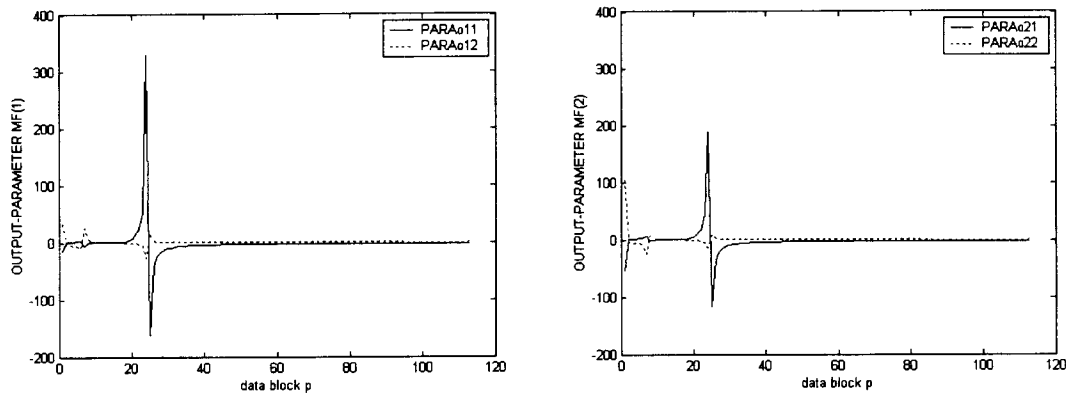


Fig. 5.10: genfis2 data-splitting, consequent-function parameters

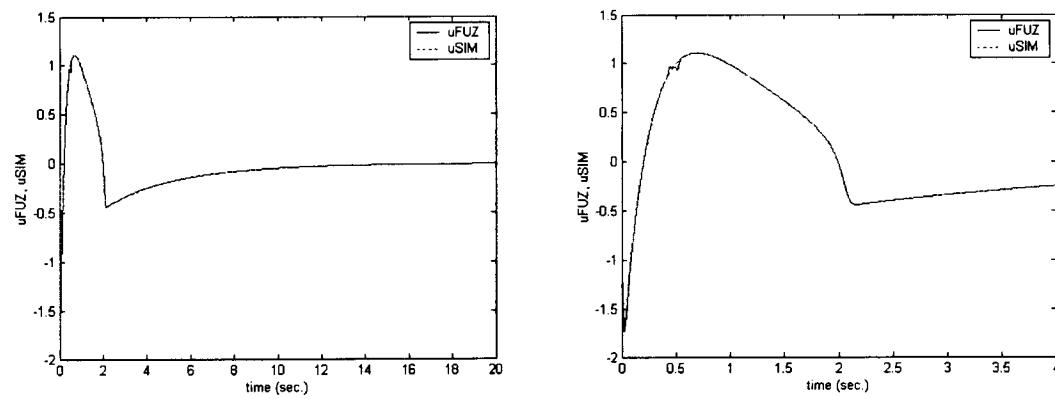


Fig. 5.11: genfis2 data-splitting, control variable:
comparison fuzzy model (uFUZ) with simulation (uSIM), right figure: zoom

model error:

$$\text{trnRMSE} = 0.00452$$

(43)

In comparison to the ANFIS training batch learning with `genfis2` (see Fig. 5.9 - Fig. 5.11) causes higher parameter variations if a pre-defined FIS structure is assumed. Especially some of the output parameters show peak values which are about a factor 10 higher than the one calculated by ANFIS. However, compared with ANFIS the model error of `genfis2` is in the same range.

5.6 Online-learning with ANFIS and `genfis2`

Both MATLAB[®] algorithms are not capable to allow an on-line learning procedure:

- ANFIS resets the initial output FIS parameters and works (structural) in batch mode with a data forgetting factor $\lambda=1$. Furthermore the gradient descent algorithm of ANFIS approximates only one data point.
- `genfis2` calculates the output parameters using an conventional global linear least squares estimation algorithm and is obviously also not able to perform online-learning.

However, both MATLAB[®] algorithms could be improved to support online-learning (*Jang, J.-S. R., 1993 and Lughofer, E., 2003*) but this should not be the aim of this paper.

5.7 Computation time and complexity of the algorithms

Concerning computation time ANFIS and `genfis2` show almost the same performance. Both need, depending on the learning-task of the above example, about 5-20 seconds if using a computer with an AMD Athlon 1.8GHz-processor inside.

ANFIS:

For decreasing computation time to a for the user acceptable delay time the MATLAB[®] ANFIS algorithm was implemented by an platform dependent MEX-file produced from a C source code. This leads to more complexity, especially developers which are not so familiar with C-programming could easily be overtaxed if they want to debug and understand this algorithm.

genfis2:

The functions “`genfis2`” as well as “`subclust`” are implemented as m-Files in MATLAB[®] and therefore relatively easy debug- and readable.

6 Conclusion and outlook

The author presented an approach which can be used to stabilize a continuous time, closed loop fuzzy control system with a non-linear plant under control by off-line tuning an adaptive fuzzy controller such, that due to applying special control Lyapunov functions a pre-specified Lyapunov derivative of the closed loop system is guaranteed. This works well if a good approximation algorithm for training the controller FIS can be found. Using the MATLAB[®] Tools Simulink, ANFIS and `genfis2` the successful step by step design of a simple fuzzy controller based on an Takagi-Sugeno-FIS was shown.

If real-time applications with time-variant plants and noisy data have to be considered, both the Lyapunov-approach and the learning strategy have to be updated by a more generalized control algorithm respectively a fast, accurate online-identification.

Acknowledgement:

The author wish to thank Mr. Edwin Lughofer, Fuzzy Logic Lab Linz, Austria who has already done a lot of useful work in the field of fuzzy identification and Professor Alexander Weinmann, for his guidance and help.

References:

- Bodendorfer, U., Burger and Haslinger, 2001, Data-Driven Construction of Sugeno Controllers, *Proc. Joint 9th IFSA World Congress and 20th NAFIPS Int. Conf.*, page(s) 239-244, 7-2001
- Chiu, S. L., 1994, A Cluster Estimation Method with Extension to Fuzzy Model Identification, *IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on*, 26-29 June 1994, Page(s): 1240 -1245 vol.2
- Jang, J.-S. R., 1992, Self Learning Fuzzy Controllers based on Temporal Backpropagation, *Neural Networks, IEEE Transactions on*, Volume: 3 Issue: 5, Sept. 1992 Page(s):714-723
- Jang, J.-S. R., 1993, ANFIS: Adaptive-Network-Based Fuzzy Inference System, *IEEE Transaction on Systems, Man., and Cybernetics*, Vol. 23, No. 3, May/June 1993
- Lughofer, E., 2003, Online Adaption of Takagi-Sugeno Fuzzy Inference Systems, *Proceedings of CESA (Computational Engineering in Systems Applications) conference*, Lille, 2003
- Lughofer E. and Groissboeck TR-212/2002, Generating Correlation and Regression Models from High Dimensional Measurement Data - Advanced Aspects, Strategies and Validation, *Technical Report FLLL-TR-0212*, Fuzzy Logic Lab Linz, Austria
- Lughofer E. and Groissboeck TR-213/2002, Online Adaptation of Correlation and Regression Models, *Technical Report FLLL-TR-0213*, Fuzzy Logic Lab Linz, Austria
- Marquez, H. J., 2003, Non-linear Control Systems – Analysis and Design (John Wiley & Sons)
- Nelles, O., 2000, Nonlinear System Identification – From Classical Approaches to Neuronal Networks and Fuzzy Models, (Springer, Berlin Heidelberg)
- Nelles, O., and Isermann, 1996, Basis function networks for interpolation of local linear models, *IEEE Conference on Decision and Control (CDC), Kobe, Japan*, pp. 470-475
- Weinmann, A., 2001, Iso-Lyapunov Dynamics – a Challenge, 12. *Steirisches Seminar über Regelungstechnik und Prozessautomatisierung, Schloss Retzhof*, in: "Tagungsband"; (2001), 3-901439-04-8; S. 61 – 96
- Wu, S., Er, and Gao, 2001, A fast Approach for Automatic Generation of Fuzzy Rules by Generalized Dynamic Fuzzy Neuronal Network, *IEEE Transactions on Fuzzy Systems*, Vol. 9, No. 4, August 2001
- Yager, R. R., and Filev, 1994, Approximate Clustering via the Mountain Method, *IEEE Transaction on Systems, Man., and Cybernetics*, Vol. 24, No. 8, August 1994

H_∞ Controllers under H_∞ Constraints

Alexander Weinmann, ÖVE, Senior Member IEEE

Vienna University of Technology, Institute of Automation and 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`

January 20, 2004

Keywords: controller infinity norm, closed-loop infinity norm, multiple eigenvalues at the imaginary axis, ramification points

Abstract

The parametrized infinity-norm minimization of the closed-loop system is extended by the infinity-norm minimization of the controller itself. The infinity norm is applied via multiple eigenvalue assignment of the Hamiltonian and its ramification points at the imaginary axis. Ramification is also considered for an extensive list of examples.

1 Introduction

Research in H_∞ is still going on whereas H_2 theory was almost finished in the seventieth. There are several similarities between H_2 and H_∞ . They are partly repeated in Appendices B and A.

This paper deals with the synthesis of H_∞ controllers based on parametrization techniques presented in the literature.

Basically, the analytical approach to problems of this kind is feasible. However, two issues have to be overcome: (i) analytic solution based on manual or computer-algebra-assisted calculation (as carried out e.g. in Weinmann, A., 2003) and (ii) the result must

be found numerically as well, since (i) yields nonlinear algebraic equations. Due to these circumstances, direct numerical minimum search is simpler and quicker, the more so as MATLAB `fmincon` and others are provided with excellent options.

For analytical evaluation, an additional drawback might be that inequalities have to be included to finish the derivation; as far as the trace is concerned or the matrix inversion of symmetric positive definite matrices (*Kerr, T.H., 1978*).

For a stable LTI system with transfer matrix $\mathbf{G}(s)$ the H_∞ norm is defined as

$$\|\mathbf{G}(s)\|_\infty \triangleq \sup_{\omega} \sigma_{\max}[\mathbf{G}(j\omega)] \quad (1)$$

where $\sigma_{\max}[\mathbf{G}] = +\sqrt{\lambda_{\max}[\mathbf{G}^T \mathbf{G}]}$ is the maximum singular value. The supremum operation requires a search versus frequency ω .

In state-space expression $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ one has

$$\mathbf{G}(s) = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array} \right] = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} . \quad (2)$$

In this context, for a system with $\mathbf{D} = \mathbf{0}$, the Hamiltonian matrix is defined as

$$\mathbf{H} \triangleq \begin{pmatrix} \mathbf{A} & \gamma^{-2}\mathbf{B}\mathbf{B}^T \\ -\mathbf{C}^T\mathbf{C} & -\mathbf{A}^T \end{pmatrix} . \quad (3)$$

Then the H_∞ norm satisfies $\|\mathbf{G}\|_\infty < \gamma$ if $\mathbf{H}$ has no eigenvalues on the imaginary axis. Usually the aforementioned inequality is easily to be satisfied for γ great.

Utilizing the relations for resolving the H_∞ norm exactly, a search is required in order to find γ such that the eigenvalues of $\mathbf{H}$ are just outside the imaginary axis. The smallest possible γ has to be found. That is,

$$\|\mathbf{H}\|_\infty = \arg \lim_{\varepsilon \rightarrow 0^+} \{ \min_{\gamma} \min_i |\Re \lambda_i[\mathbf{H}(\gamma)]| = \varepsilon \} . \quad (4)$$

In the formula in Eq.(4), the infinity norm is computed via the search over decreasing γ such that the (last) eigenvalue $\lambda_i[\mathbf{H}(\gamma)]$ just leaves the imaginary axis where the other eigenvalues are outside the imaginary axis yet. Search is performed by MATLAB `fzero`. In MATLAB, the infinity norm also results from the command `norminf`.

For the sake of completeness, the derivation of the H_∞ norm and random mean gain with the help of LMI techniques should be mentioned (*Boyd, S., et al., 1994; Gahinet, P., et al., 1995; Weinmann, A., 1997a; 1999*).

In addition, LMI techniques are frequently applied to incorporate robustness and identification problems (*Dong Hu, 1999; Weinmann, A., 1991, 1996, 1997*). Emphasizing uncertain initial conditions, see *Khargonekar, P.P., et al., 1991*.

The main result of this paper is the utilization of the freedom of the parametrization of closed-loop infinity-norm results for obtaining a minimum infinity norm of the controller.

The paper is organized as follows: The infinity norm is applied via multiple eigenvalue assignment of the Hamiltonian and its ramification points at the imaginary axis in Section 2. In Section 3, as the main part, the parametrized infinity-norm minimization of the closed-loop system is extended by the infinity-norm minimization of the controller itself. In a comprehensive appendix several examples of ramifications are presented.

2 Infinity Norm and Ramification Point

2.1 One Single Ramification Point

Consider $\mathbf{H}_n = \mathbf{H}$ as given in Eq. 3 for an n -th order system. A double eigenvalue $s_o = \lambda_i[\mathbf{H}_n]$ exists if

$$\det[s_o \mathbf{I}_n - \mathbf{H}_n(\gamma)] = 0 \quad (5)$$

$$\frac{\partial}{\partial s} \det[s_o \mathbf{I} - \mathbf{H}_n(\gamma)] = \sum_{i=1}^n \det[s_o \mathbf{I}_{n-1} - \mathbf{H}_{n \rightarrow n-1}(\gamma)]_i = 0 \quad (6)$$

in the two unknowns are resolved, i.e., in the imaginary $s_0 = j\omega_0$ and the real γ .

2.2 Multiple Imaginary Eigenvalues and Main Reduced Matrices

Consider the system $\mathbf{G}(s)$ of n -th order. Then the Hamiltonian Eq.(3) is of dimension $2n$. There are several possibilities for $\lambda_i[\mathbf{H}]$. Find γ in Eq.(3) such that no imaginary eigenvalues $\lambda_i[\mathbf{H}]$ exist, i.e., do not exist any more if γ is increased by an infinitesimally small amount. This happens if the eigenvalues just enter a multiple pole, a conjugate complex pole of multiplicity at maximum n or lower. In such a way, the search problem of the H_∞ norm is replaced by the problem of finding a single or multiple eigenvalues at the imaginary axis.

In order to detect a pole of multiplicity n in $\mathbf{H} = \mathbf{H}_n$, resolve

- the determinant

$$\det[s_o \mathbf{I} - \mathbf{H}_n(\gamma)] = 0; \quad (7)$$

- the derivative of the determinant

$$\frac{\partial}{\partial s} \det[s_o \mathbf{I} - \mathbf{H}_n(\gamma)] = 0 \quad (8)$$

equivalent to

$$\text{tr adj}[s_o \mathbf{I} - \mathbf{H}_n(\gamma)] = \sum_{i=1}^n \det[s_o \mathbf{I}_{n-1} - \mathbf{H}_{n \rightarrow n-1}]_i = 0 \quad (9)$$

where $[s_o \mathbf{I} - \mathbf{H}_{n \rightarrow n-1}]_i$ is one of the n reduced matrices of $\mathbf{H}_n$ with respect to one of the elements of the main diagonal (*Weinmann, A., 2004*); for the second derivative see Eq.(10).

For the second derivative, one finds

$$\frac{\partial^2}{\partial s^2} \det(s\mathbf{I} - \mathbf{H}_n) = \sum_{j=1}^{n-1} \sum_{i=1}^n \det[s\mathbf{I} - (\mathbf{H}_{n \rightarrow n-2})_i]_j \quad (10)$$

where

$$(\mathbf{sI} - \mathbf{H}_{n \rightarrow n-1})_i \in \mathcal{R}^{(n-1) \times (n-1)} \quad (11)$$

$$[s\mathbf{I} - (\mathbf{H}_{n \rightarrow n-2})_i]_j \in \mathcal{R}^{(n-2) \times (n-2)} . \quad (12)$$

Omitting the i -th row and column of e.g. $\mathbf{H}_3$ can be performed with the help of the matrix $\mathbf{I}_{32} \triangleq$ “eye(3) without column 2”, i.e.,

$$\mathbf{I}_{32} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}. \quad (13)$$

For $i = 2$ there results $(s\mathbf{I} - \mathbf{H}_{3 \rightarrow 2})_i = \mathbf{I}_{32}^T (s\mathbf{I} - \mathbf{H}_{3 \rightarrow 2}) \mathbf{I}_{32}$.

For illustration see some figures of Example 4 in Appendix C.

- Repeat this procedure up to n .

Remark 1: The equation $\frac{\partial}{\partial s} \det(s\mathbf{I} - \mathbf{H}_n) = 0$ has additional solutions, not corresponding to $\det(s\mathbf{I} - \mathbf{H}_n) = 0$. These solutions have to be left out of consideration. $\square$

3 H_∞ Control with H_∞ Controller Constraints

As given by *Doyle, J.C., et al., 1989*, and repeated in Appendix B, the standard state-space solutions of the H_∞ control problem of Fig. 1 are presented using a set of inequalities Eq.(48) and (49).

Studying the problem in numerical detail for some industrial example, this reveals additional opportunities for advanced considerations of design.

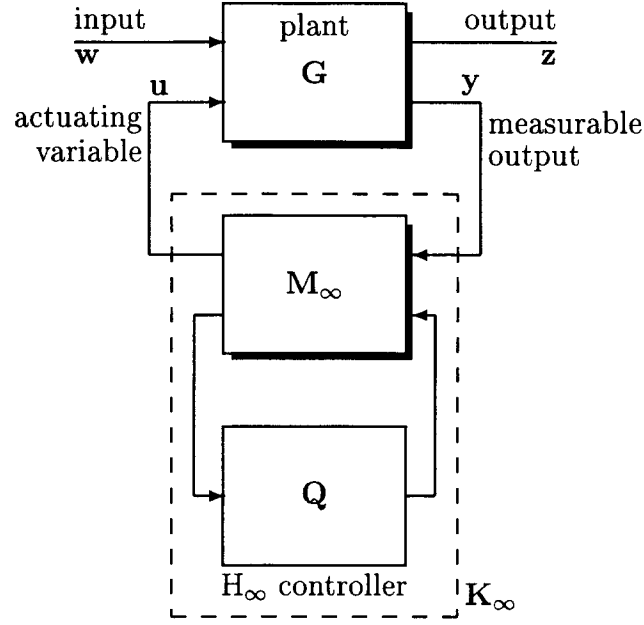


Figure 1: Generalized plant G , H_∞ controller K_∞ based on M_∞ and parameter matrix Q

The matrix-valued parameter matrix Q can be utilized for norm constraint of the H_∞ controller. In addition, selection of Q is not free of interdependence with the actual H_∞ norm of $T_{zw}(s)$.

The parametrization method of Appendix B is utilized for displaying $\sigma_{\max}[K]$ versus Q for selecting

$$\min_Q \|K\|_\infty. \quad (14)$$

The infinity norm is derived based on Eq.(4), with the help of ramification points or common MATLAB tools. The advantage when using the ramification procedure can be substantially exhausted in cases of using the infinity norm as a condition of complex problems (Weinmann, A., 2004a).

Several advantages are taken into account when the controller norm is restricted (Weinmann, A., 2004b.)

Example 1. Multivariable System: Consider the system

$$A = \begin{pmatrix} 0 & -5 \\ 1 & -2 \end{pmatrix}, \quad B = (B_1 : B_2) = \begin{pmatrix} 0 & 10 \\ 1 & 2 \end{pmatrix}, \quad (15)$$

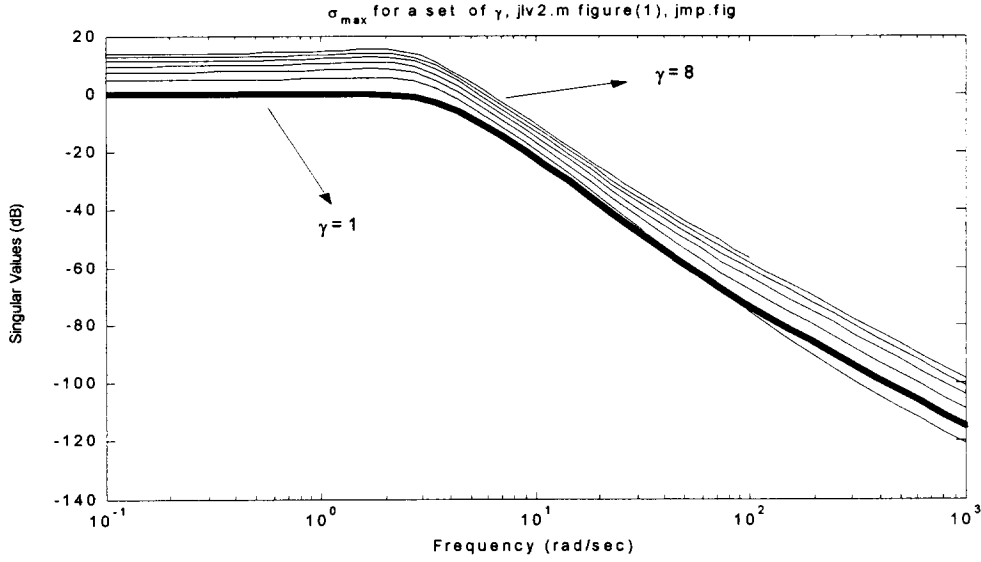


Figure 2: Singular values $\sigma_{\max}[\mathbf{T}_{zw}]$ versus frequency for a set of γ and $Q = \gamma$

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_1 \\ \dots \\ \mathbf{C}_2 \end{pmatrix} = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.3 \end{pmatrix}. \quad (16)$$

In Fig. 2, the SISO system $\mathbf{T}_{zw}$ is presented in its singular value (equivalent to the well-known Bode plot) for a set of γ and $Q \equiv \gamma$. Fig. 3 depicts a similar plot for $\gamma = 1$ and a set of scalar Q .

In Fig. 4 the singular value of $K(s)$ is shown for a set of seven different Q . The optimum choice is emphasized in line width. For the sake of illustration, the corresponding frequency plots in the complex plane are presented in Fig. 5. The resulting H_∞ norm of $\mathbf{K}$ versus Q is depicted in Fig. 6.

Fig. 7 warrants the effort for combined minimization where the combination is $\|\mathbf{T}_{zw}\|_{\infty} + 0.5\|\mathbf{K}\|_{\infty}$. The combined criterion for $\|\mathbf{T}_{zw}\|_{\infty}$ and $\|\mathbf{K}\|_{\infty}$ shows a distinct minimum versus γ and Q .

The minimum versus Q is only slightly influenced by γ . But there is no doubt that $Q = 0$ is not the minimum.

In Fig. 8, the criteria $\|\mathbf{T}_{zw}\|_{\infty}$ and $\|\mathbf{K}\|_{\infty}$ are depicted separately and in a combination, above a basis plane with scalars Q and γ as abscissa and ordinate, respectively.

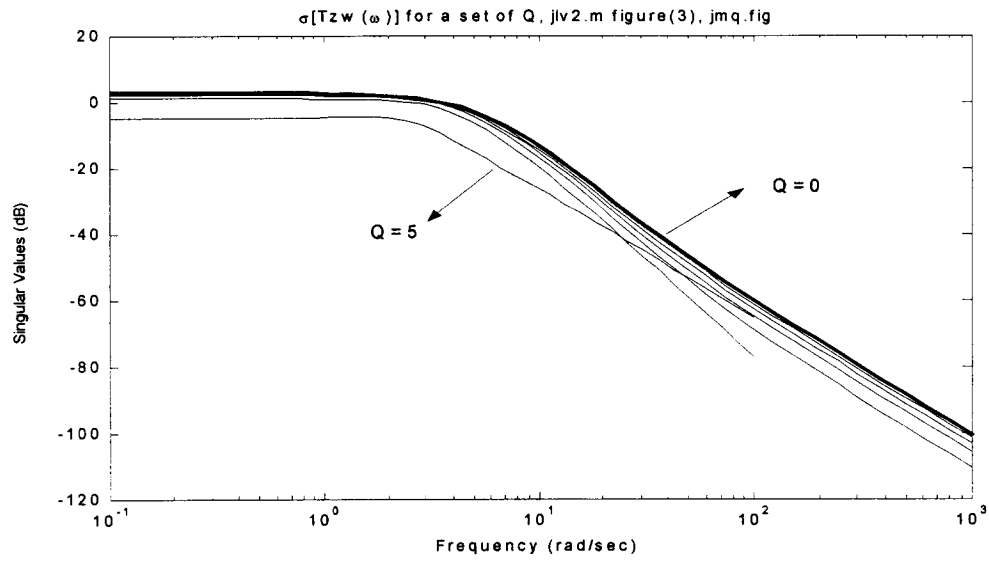


Figure 3: Singular values $\sigma_{\max}[\mathbf{T}_{zw}]$ for $\gamma = 1$ and a set of Q

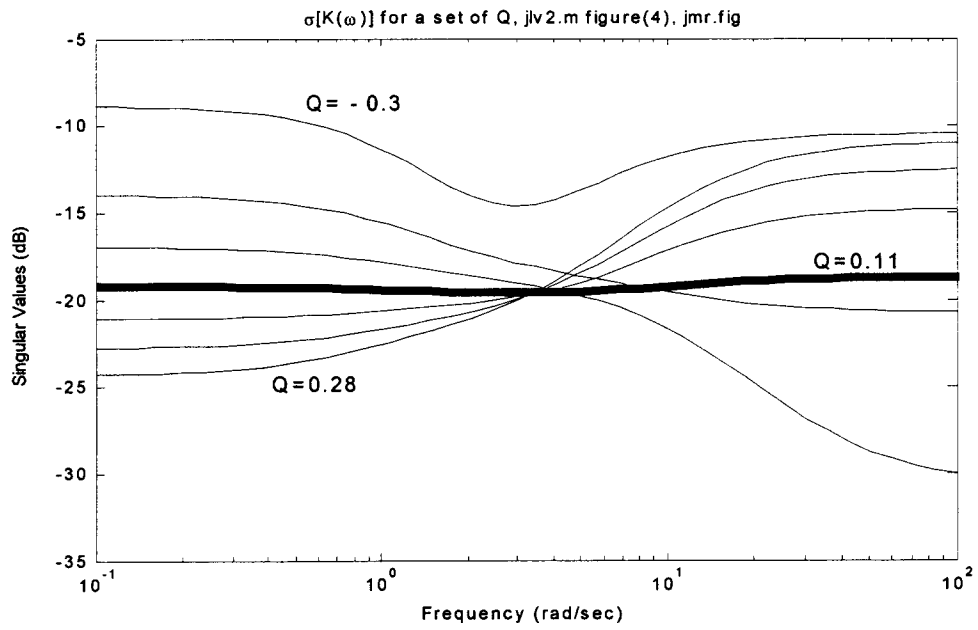
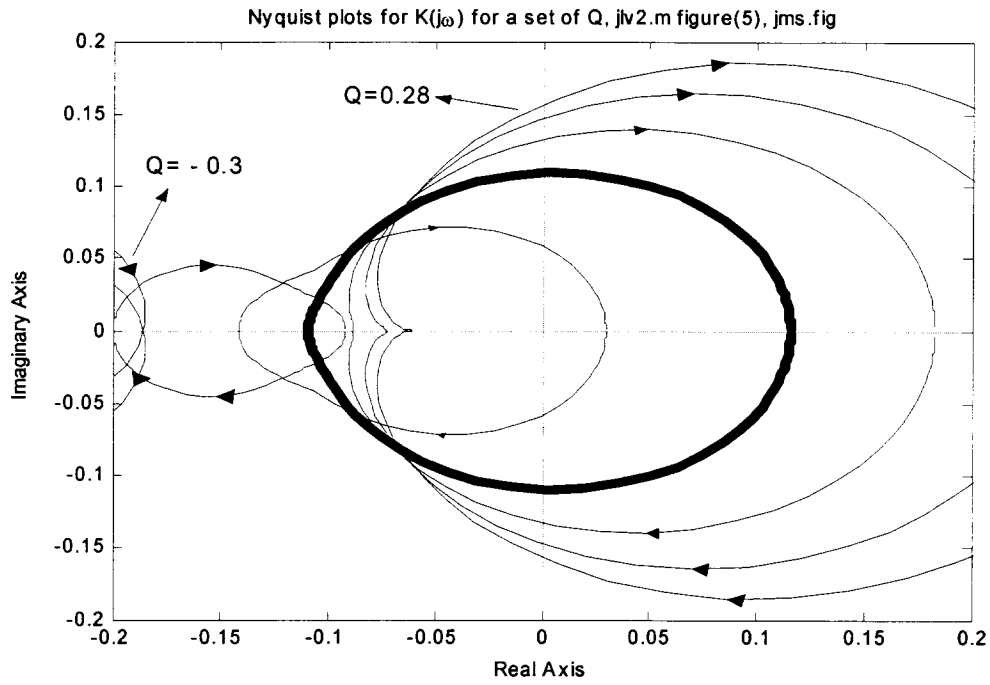
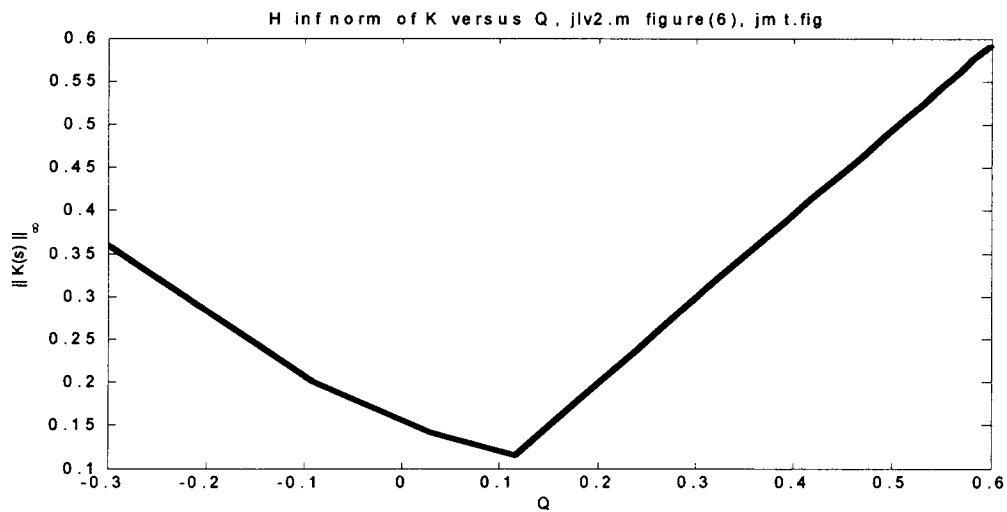


Figure 4: Singular value $\sigma_{\max}[\mathbf{K}]$ for a set of Q

Figure 5: Frequency plots of K Figure 6: $\|K\|_\infty$ versus Q

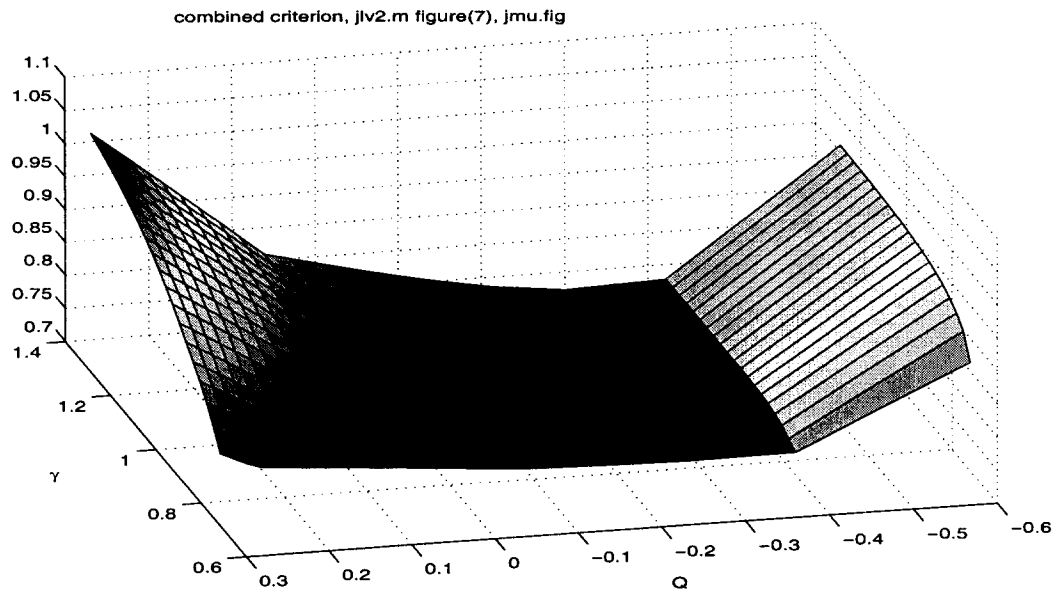


Figure 7: Combined criterion based on $\|T_{zw}\|_\infty$ and $\|K\|_\infty$ versus Q and γ

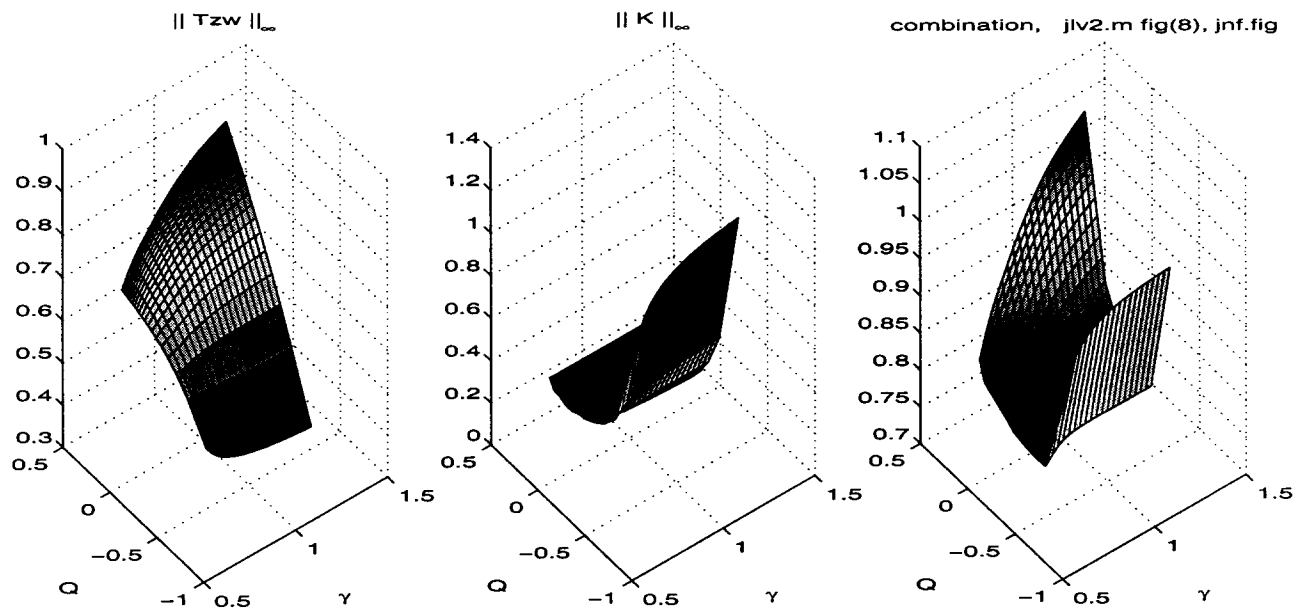


Figure 8: Separate and combined criteria versus Q and γ

4 Conclusion

The parametrization of H_∞ controllers as known in the literature is utilized for selecting the controller with least norm in order to reject measurement noise optimally and to minimize actuator signal peaks. Even for low-order systems a remarkable improvement is achieved.

Appendix

A Hamiltonian and Riccati solution

A.1 Version 1

If $\mathbf{A} \in \mathcal{R}^{n \times n}$, $\mathbf{Q}$ and $\mathbf{U} \in \mathcal{R}^{n \times n}$ are symmetric, and $(\mathbf{A}, \mathbf{U})$ are stabilizable then the Hamiltonian is defined as

$$\mathbf{H} = \begin{pmatrix} \mathbf{A} & \mathbf{U} \\ \mathbf{Q} & -\mathbf{A}^T \end{pmatrix} \in \text{dom}(\text{Ric}) . \quad (17)$$

If $\mathbf{H}$ has no eigenvalues at the imaginary axis, then

$$\mathbf{A} + \mathbf{UX} \quad (18)$$

is stable where $\mathbf{X}$ results from the Riccati equation

$$\mathbf{A}^T \mathbf{X} + \mathbf{XA} + \mathbf{XUX} - \mathbf{Q} = \mathbf{0} . \quad (19)$$

In short, we write $\mathbf{X} = \text{Ric}(\mathbf{H}) \geq 0$. Then,

$$\lambda_i(\mathbf{A} + \mathbf{UX}) \equiv \text{stable part of } \lambda_i[\mathbf{H}] , \quad (20)$$

i.e., the eigenvalues of a closed loop are equivalent to the selected part of stable eigenvalues of $\mathbf{H}$.

Example 2: Consider

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -2 & 4 \end{pmatrix}, \quad \mathbf{U} = \begin{pmatrix} -2 & -5 \\ -5 & -3 \end{pmatrix}, \quad \mathbf{Q} = 0.1 \begin{pmatrix} -3 & 4 \\ 4 & -9 \end{pmatrix}, \quad (21)$$

then

$$\mathbf{X} = \begin{pmatrix} -0.68 & 0.56 \\ 0.56 & 1.16 \end{pmatrix} \quad (22)$$

and

$$\lambda_i[\mathbf{H}] = \pm 3.32; \quad \pm 0.50; \quad \lambda_i[\mathbf{A} + \mathbf{U}\mathbf{X}] = -3.32; \quad -0.50. \quad (23)$$

A.2 Version 2

Replace $\mathbf{U} = \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T$ and change signs in the Riccati equation in order to obtain Eq.(25), then

$$\mathbf{H} = \begin{pmatrix} \mathbf{A} & \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \\ \mathbf{Q} & -\mathbf{A}^T \end{pmatrix}. \quad (24)$$

The classical LQR problem with weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ leads to the Riccati equation

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = \mathbf{0}. \quad (25)$$

Defining an (eigenvalue-preserving) similarity transformation with the specified matrix $\mathbf{T} \triangleq \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{P} & \mathbf{I} \end{pmatrix}$, then, using Eq.(25),

$$\mathbf{T}^{-1}\mathbf{H}\mathbf{T} = \begin{pmatrix} \mathbf{A} + \mathbf{B}\mathbf{K} & \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \\ \mathbf{0} & -(\mathbf{A} + \mathbf{B}\mathbf{K}) \end{pmatrix} \triangleq \mathbf{H}_1. \quad (26)$$

The characteristic equation of Eq.(26) is, see Remark 3 and Eq.(31) and *Ludyk, G., 1995*,

$$\det [\lambda\mathbf{I} - (\mathbf{A} + \mathbf{B}\mathbf{K})] \cdot \det [\lambda\mathbf{I} + (\mathbf{A} + \mathbf{B}\mathbf{K})] = 0. \quad (27)$$

Thus the set union of eigenvalues is

$$\{\lambda_i[\mathbf{H}]\} = \{\lambda_i[\mathbf{H}_1]\} = \{\lambda_i[\mathbf{A} + \mathbf{B}\mathbf{K}]\} \cup \{\lambda_i[-(\mathbf{A} + \mathbf{B}\mathbf{K})]\}. \quad (28)$$

Furthermore, selecting the eigenvectors of $\mathbf{H}$ corresponding to the eigenvalues with negative real part yields a matrix $\mathbf{V}$. After separating into $\mathbf{V}_1$ and $\mathbf{V}_2$ the interesting unknown $\mathbf{P}$ can be obtained

$$\mathbf{V} \triangleq \begin{pmatrix} \mathbf{V}_1 \\ \dots \\ \mathbf{V}_2 \end{pmatrix} \rightsquigarrow \mathbf{P} = -\mathbf{V}_2\mathbf{V}_1^{-1}. \quad (29)$$

Remark 2: A matrix

$$\mathbf{M} \triangleq \begin{pmatrix} \mathbf{A} & \mathbf{U} \\ \mathbf{N} & \mathbf{F} \end{pmatrix} \quad (30)$$

for $\mathbf{A}$ square, $\mathbf{U}$ rectangular, $\mathbf{N} = \mathbf{0}$, is characterized by $\lambda_i[\mathbf{M}]$ independent of $\mathbf{U}$. $\square$

Remark 3: The matrix

$$\mathbf{H} \triangleq \begin{pmatrix} \mathbf{A} & \mathbf{U} \\ \mathbf{N} & -\mathbf{A}^T \end{pmatrix} \quad \mathbf{N} = \mathbf{0} \quad (31)$$

and hence $\mathbf{U}$ square entails eigenvalues $\lambda_i[\mathbf{H}]$ which are symmetric to both real and imaginary axes. $\square$

Example 3: Using Eq.(15) and

$$\mathbf{Q} = \begin{pmatrix} 5 & 2 \\ 2 & 3 \end{pmatrix}, \quad \mathbf{R} = 0.1 \mathbf{I}_2 \quad (32)$$

one finds

$$\mathbf{H} = \begin{pmatrix} 0 & -5 & 1000 & 200 \\ 1 & -2 & 200 & 50 \\ 5 & 2 & 0 & -1 \\ 2 & 3 & 5 & 2 \end{pmatrix}, \quad (33)$$

$$\mathbf{V}_1 = \begin{pmatrix} -0.98 & 0.45 \\ -0.19 & -0.84 \end{pmatrix}, \quad \mathbf{V}_2 = \begin{pmatrix} 0.07 & -0.07 \\ 0.02 & 0.29 \end{pmatrix}, \quad \mathbf{P} = \begin{pmatrix} 0.077 & -0.036 \\ -0.036 & 0.33 \end{pmatrix}, \quad (34)$$

$$\mathbf{K} = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} = \begin{pmatrix} 0.36 & -3.32 \\ -7.01 & -3.04 \end{pmatrix} \quad (35)$$

$$\lambda_i[\mathbf{H}] = \pm 76.96; \pm 4.56; \quad \lambda_i[\mathbf{A} + \mathbf{BK}] = -76.96; -4.56. \quad (36)$$

B Parameterized H_∞ Controllers

In *Doyle, J.C., et al. 1989*, the following results are given and derived: Define

$$\mathbf{H}_\infty \triangleq \begin{pmatrix} \mathbf{A} & \gamma^{-2}\mathbf{B}_1\mathbf{B}_1^T - \mathbf{B}_2\mathbf{B}_2^T \\ -\mathbf{C}_1^T\mathbf{C}_1 & -\mathbf{A}^T \end{pmatrix} \quad (37)$$

$$\mathbf{J}_\infty \triangleq \begin{pmatrix} \mathbf{A}^T & \gamma^{-2}\mathbf{C}_1^T\mathbf{C}_1 - \mathbf{C}_2^T\mathbf{C}_2 \\ -\mathbf{B}_1\mathbf{B}_1^T & -\mathbf{A} \end{pmatrix}. \quad (38)$$

Iff

$$\mathbf{H}_\infty \in \text{dom}(\text{Ric}) \quad \text{and} \quad \mathbf{X}_\infty = \text{Ric}(\mathbf{H}_\infty) \geq 0 \quad (39)$$

and

$$\mathbf{J}_\infty \in \text{dom}(\text{Ric}) \quad \text{and} \quad \mathbf{Y}_\infty = \text{Ric}(\mathbf{J}_\infty) \geq 0 \quad (40)$$

and the spectral radius

$$\rho[\mathbf{X}_\infty\mathbf{Y}_\infty] < \gamma^2, \quad (41)$$

then

$$\mathbf{K}_\infty(s) = \left[\begin{array}{c|c} \mathbf{A}_\infty - \mathbf{Z}_\infty \mathbf{B}_2 \mathbf{Q} \mathbf{C}_2 & \mathbf{Z}_\infty (\mathbf{B}_2 \mathbf{Q} - \mathbf{L}_\infty) \\ \hline \mathbf{F}_\infty - \mathbf{Q} \mathbf{C}_2 & \mathbf{Q} \end{array} \right] \quad (42)$$

or

$$\mathbf{M}_\infty = \left[\begin{array}{c|cc} \mathbf{A}_\infty & \mathbf{Z}_\infty \mathbf{L}_\infty & \mathbf{Z}_\infty \mathbf{B}_2 \\ \hline \mathbf{F}_\infty & \mathbf{0} & \mathbf{I} \\ \mathbf{C}_2 & \mathbf{I} & \mathbf{0} \end{array} \right] \quad (43)$$

where

$$\mathbf{F}_\infty \triangleq -\mathbf{B}_2^T \mathbf{X}_\infty \quad (44)$$

$$\mathbf{L}_\infty \triangleq -\mathbf{Y}_\infty \mathbf{C}_2^T \quad (45)$$

$$\mathbf{Z}_\infty \triangleq (\mathbf{I} - \gamma^{-2} \mathbf{Y}_\infty \mathbf{X}_\infty)^{-1} \quad (46)$$

$$\mathbf{A}_\infty \triangleq \mathbf{A} + \gamma^{-2} \mathbf{B}_1 \mathbf{B}_1^T \mathbf{X}_\infty + \mathbf{B}_2 \mathbf{F}_\infty + \mathbf{Z}_\infty \mathbf{L}_\infty \mathbf{C}_2 \quad (47)$$

and, finally,

$$\|\mathbf{T}_{zw}\|_\infty < \gamma \quad (48)$$

and $\mathbf{Q}$ is a free parameter matrix with

$$\|\mathbf{Q}\|_\infty < \gamma. \quad (49)$$

C Infinity Norm via Minimum Search and Ramification Point

Example 4. Second-order dynamical system Consider the numerical settings of Example 1 with Eqs.(15) and (16).

Fig. 9 displays the four eigenvalues $\lambda_i[\mathbf{H}]$ for $2 < \gamma < 5$ in the complex plane whereas Fig. 10 separates the real and imaginary parts versus γ . For the sake of consistency, in Fig. 11 the four frequency plots $\mathbf{G}(j\omega)$ are depicted. Fig. 12 shows the singular values $\sigma_i[\mathbf{G}(s)]$ versus frequency including the infinity norm $\|\mathbf{G}(s)\|_\infty$ and the frequency at the multiple pole $j2.12$.

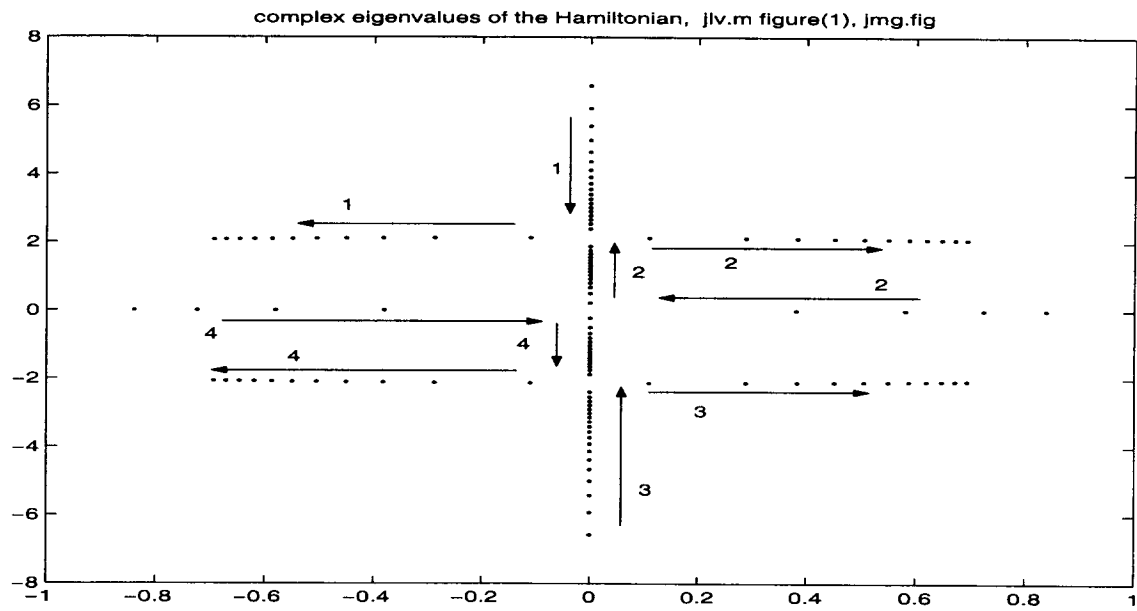
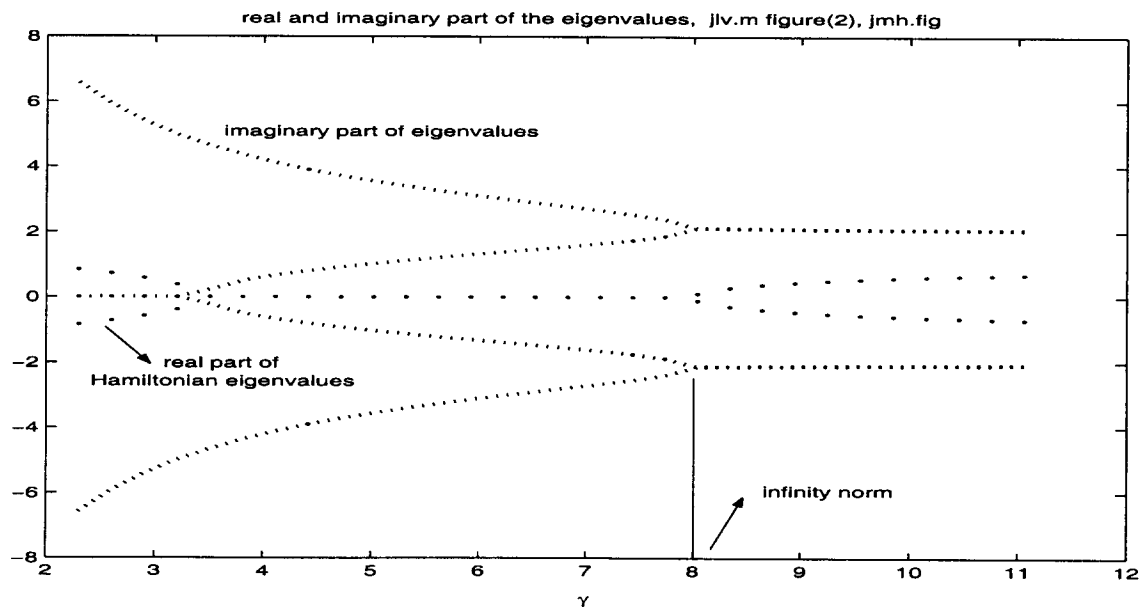
In Fig. 13 the zero of $\frac{\partial}{\partial s} \det \mathbf{H} = \text{tr } \text{adj } \mathbf{H}$ is demonstrated by displaying its absolute value versus variable γ .

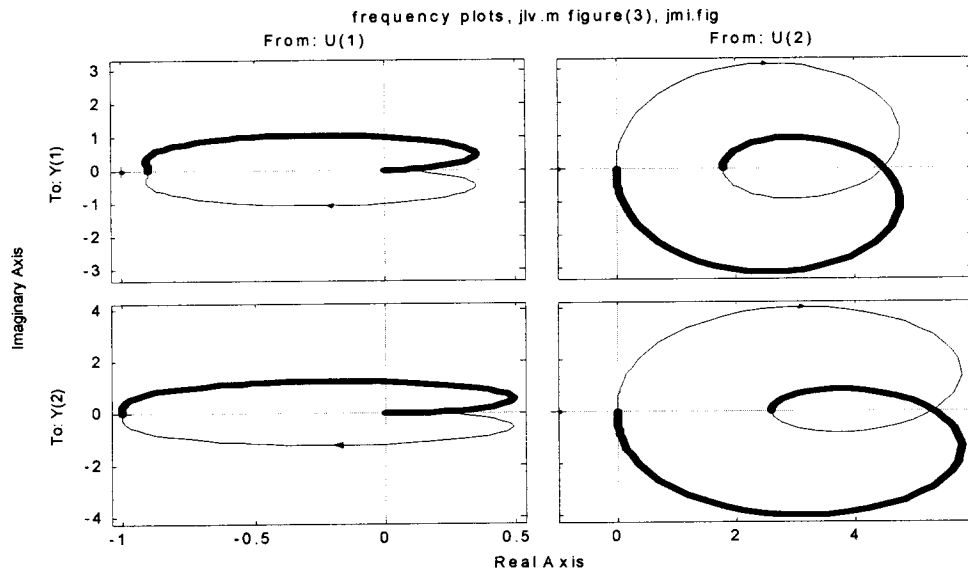
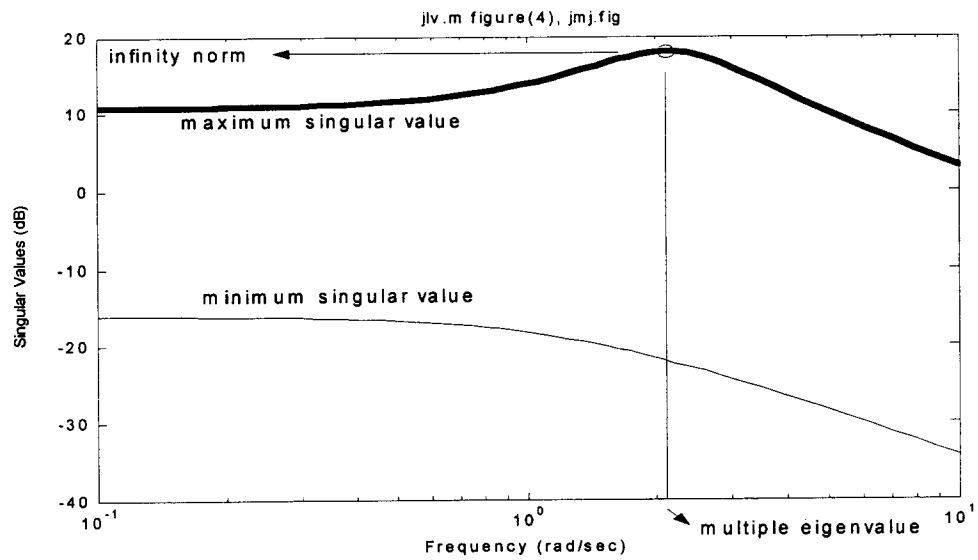
The sum

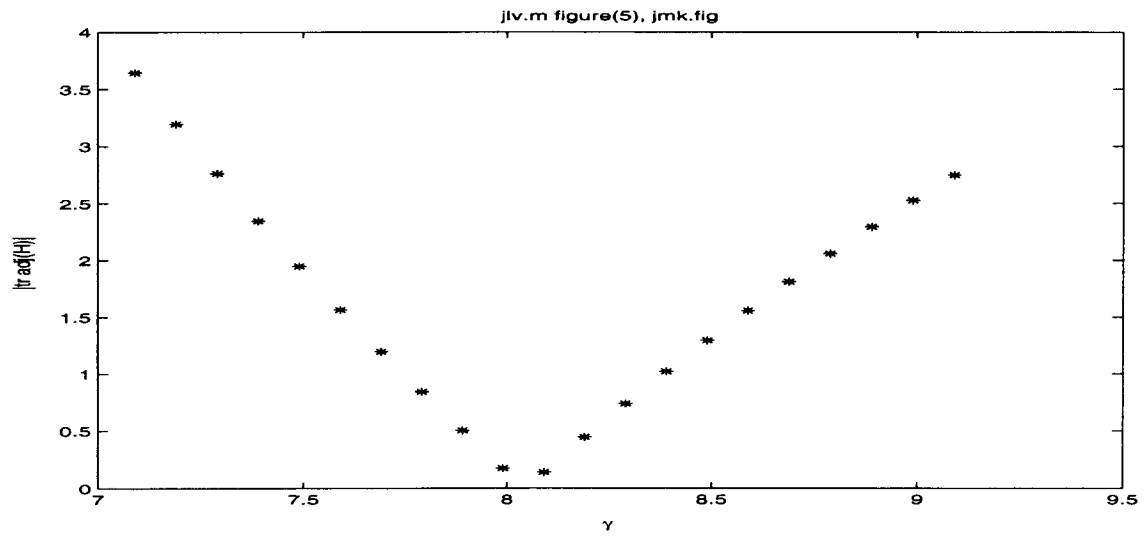
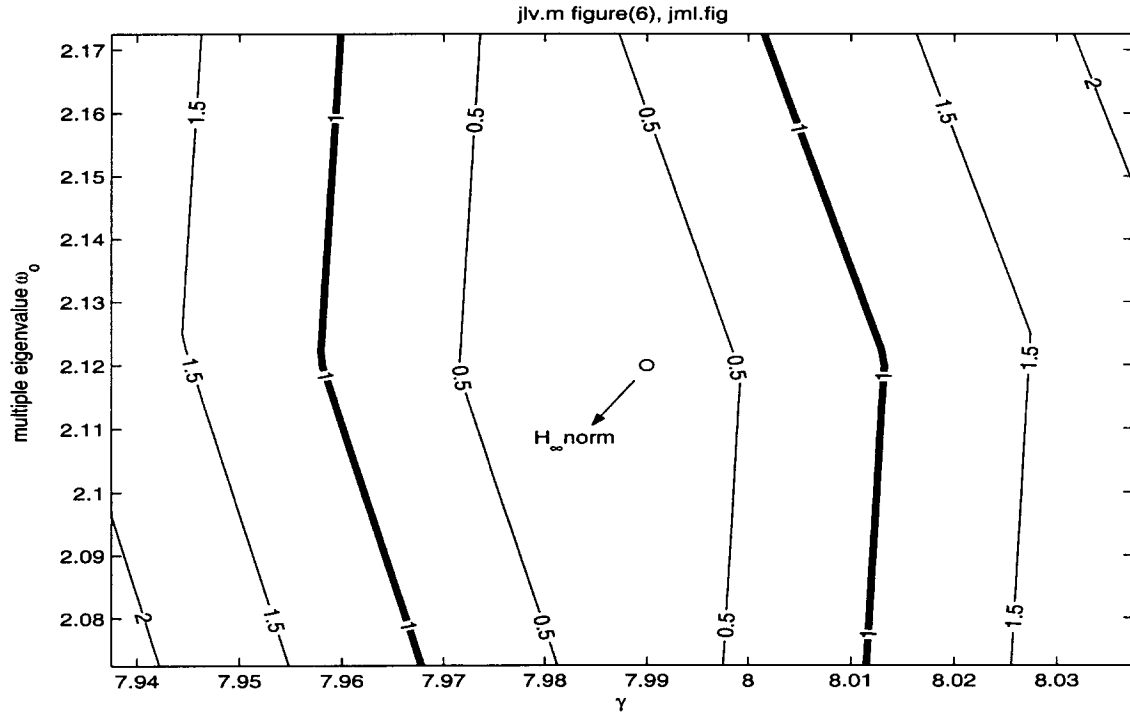
$$S = |\det \mathbf{H}| + \left| \frac{\partial}{\partial s} \det \mathbf{H} \right| \triangleq S_2 + S_1 \quad (50)$$

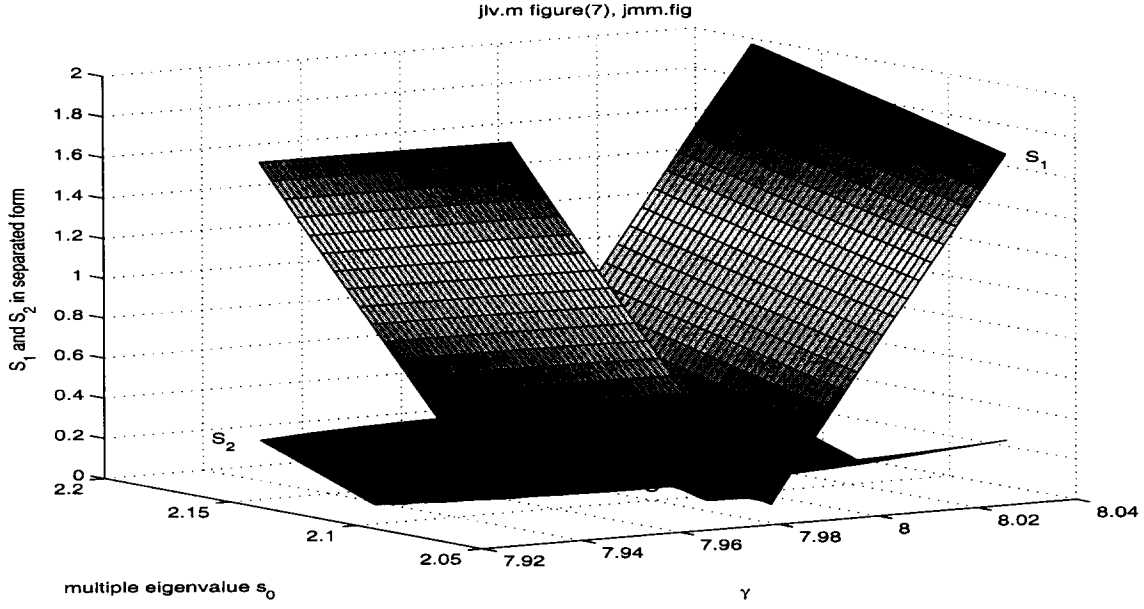
is given by a contour plot versus γ and ω_0 in Fig. 14.

The terms above are presented separately in a mesh diagram in Fig. 15.

Figure 9: Complex eigenvalues $\lambda_i[\mathbf{H}]$ Figure 10: Real and imaginary parts of $\lambda_i[\mathbf{H}]$


Figure 11: Frequency plots of $G(j\omega)$

Figure 12: Singular values $\sigma_i[G(s)]$

Figure 13: Derivative $|\text{tr adj}(H)|$ Figure 14: Contour plot of S

Figure 15: Mesh diagram for both parts S_1 and S_1 of S

In some different example, e.g. for $n = 4$, i.e. $\mathbf{H} \in R^{8 \times 8}$, the smallest γ for the last complex eigenvalue leaving the imaginary axis might also occur for one pole of multiplicity 4, 3 or 2.

D Ramification Points of Hamiltonian's Eigenvalues

Example 5. Double Zero at $\pm j2$: Consider the polynomial

$$(s^2 + 4)^2 = s^4 + 8s^2 + 16. \quad (51)$$

Select the matrix $\mathbf{H}_n$

$$n = 4 \quad \mathbf{H}_n = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -16 & 0 & -8 & 0 \end{pmatrix}, \quad \text{then } s\mathbf{I} - \mathbf{H}_n = \begin{pmatrix} s & -1 & 0 & 0 \\ 0 & s & -1 & 0 \\ 0 & 0 & s & -1 \\ 16 & 0 & 8 & s \end{pmatrix} \quad (52)$$

corresponds to

$$\det(s\mathbf{I} - \mathbf{H}_4) = (s^2 + 4)^2 = 0. \quad (53)$$

The main reduced matrices are

$$(s\mathbf{I} - \mathbf{H}_{n \rightarrow n-1})_i : \begin{pmatrix} s & -1 & 0 \\ 0 & s & -1 \\ 0 & 8 & s \end{pmatrix}, \begin{pmatrix} s & 0 & 0 \\ 0 & s & -1 \\ 16 & 8 & s \end{pmatrix}, \begin{pmatrix} s & -1 & 0 \\ 0 & s & 0 \\ 16 & 0 & s \end{pmatrix}, \begin{pmatrix} s & -1 & 0 \\ 0 & s & -1 \\ 0 & 0 & s \end{pmatrix} \quad (54)$$

The sum yields

$$\sum_{i=1}^4 \det(s\mathbf{I} - \mathbf{H}_{n \rightarrow n-1})_i = s(s^2 + 8) + s(s^2 + 8) + s^3 + s^3 \quad (55)$$

$$= 4s^3 + 16s = 0 \rightsquigarrow s = \pm j2. \quad (56)$$

The additional solution $s = 0$ is obsolete since it does not fit $\det(s\mathbf{I} - \mathbf{H}_n)$.

Example 6. Third-Order Zero at -1 :

$$s\mathbf{I} - \mathbf{H}_3 = \begin{pmatrix} s & -1 & 0 \\ 0 & s & -1 \\ 1 & 3 & s+3 \end{pmatrix}, \quad \det(s\mathbf{I} - \mathbf{H}_3) = (s+1)^3 \quad (57)$$

$$\text{adj}(s\mathbf{I} - \mathbf{H}_3) = \begin{pmatrix} \det \begin{pmatrix} s & -1 \\ 3 & s+3 \end{pmatrix} & \cdot & \cdot \\ \cdot & \det \begin{pmatrix} s & 0 \\ 1 & s+3 \end{pmatrix} & \cdot \\ \cdot & \cdot & \det \begin{pmatrix} s & -1 \\ 0 & s \end{pmatrix} \end{pmatrix} \quad (58)$$

$$\frac{\partial \det(s\mathbf{I} - \mathbf{H}_3)}{\partial s} = \text{tr adj}(s\mathbf{I} - \mathbf{H}_3) \quad (59)$$

$$= \det(s\mathbf{I} - \mathbf{H}_{3 \rightarrow 2})_1 + \det(s\mathbf{I} - \mathbf{H}_{3 \rightarrow 2})_2 + \det(s\mathbf{I} - \mathbf{H}_{3 \rightarrow 2})_3 \quad (60)$$

$$= \sum_{i=1}^3 \det(s\mathbf{I} - \mathbf{H}_{3 \rightarrow 2})_i \quad (61)$$

$$= \det \begin{pmatrix} s & -1 \\ 3 & s+3 \end{pmatrix} + \det \begin{pmatrix} s & 0 \\ 1 & s+3 \end{pmatrix} + \det \begin{pmatrix} s & -1 \\ 0 & s \end{pmatrix} \quad (62)$$

$$\begin{aligned} \frac{\partial^2 \det(s\mathbf{I} - \mathbf{H}_3)}{\partial s^2} &= \frac{\partial}{\partial s} \det(s\mathbf{I} - \mathbf{H}_{3 \rightarrow 2})_1 + \frac{\partial}{\partial s} \det(s\mathbf{I} - \mathbf{H}_{3 \rightarrow 2})_2 + \frac{\partial}{\partial s} \det(s\mathbf{I} - \mathbf{H}_{3 \rightarrow 2})_3 \\ &= \text{tr adj}(s\mathbf{I} - \mathbf{H}_{3 \rightarrow 2})_1 + \text{tr adj}(s\mathbf{I} - \mathbf{H}_{3 \rightarrow 2})_2 + \text{tr adj}(s\mathbf{I} - \mathbf{H}_{3 \rightarrow 2})_3 \\ &= \sum_{j=1}^2 \det((s\mathbf{I} - \mathbf{H}_{3 \rightarrow 1})_1)_j + \sum_{j=1}^2 \det((s\mathbf{I} - \mathbf{H}_{3 \rightarrow 1})_2)_j + \sum_{j=1}^2 \det((s\mathbf{I} - \mathbf{H}_{3 \rightarrow 1})_3)_j \\ &= \sum_{j=1}^2 \sum_{i=1}^3 \det((\mathbf{H}_{3 \rightarrow 1})_i)_j \end{aligned} \quad (63)$$

yielding

$$s+3+s+s+3+s+s+s=6s+6=0 \rightsquigarrow s=-1. \quad (64)$$

Example 7. Fourth-Order Zero at -1 : One has

$$(s+1)^4 = s^4 + 4s^3 + 6s^2 + 4s + 1 \quad (65)$$

$$s\mathbf{I} - \mathbf{H}_4 = \begin{pmatrix} s & -1 & 0 & 0 \\ 0 & s & -1 & 0 \\ 0 & 0 & s & -1 \\ 1 & 4 & 6 & s+4 \end{pmatrix}, \quad \det(s\mathbf{I} - \mathbf{H}_4) = (s+1)^4 = 0 \quad (66)$$

$$s\mathbf{I} - (\mathbf{H}_{4 \mapsto 3})_i : \begin{pmatrix} s & -1 & 0 \\ 0 & s & -1 \\ 4 & 6 & s+4 \end{pmatrix}, \begin{pmatrix} s & 0 & 0 \\ 0 & s & -1 \\ 1 & 6 & s+4 \end{pmatrix}, \begin{pmatrix} s & -1 & 0 \\ 0 & s & 0 \\ 1 & 4 & s+4 \end{pmatrix}, \begin{pmatrix} s & -1 & 0 \\ 0 & s & -1 \\ 0 & 0 & s \end{pmatrix} \quad (67)$$

$$\sum_i \det(\mathbf{H}_{4 \mapsto 3})_i = 4s^3 + 12s^2 + 12s + 4 = (s+1)^3 = 0 \quad (68)$$

$$s\mathbf{I} - ((\mathbf{H}_{4 \mapsto 2})_i)_j : \begin{pmatrix} s & -1 \\ 6 & s+4 \end{pmatrix}, \begin{pmatrix} s & 0 \\ 4 & s+4 \end{pmatrix}, \begin{pmatrix} s & -1 \\ 0 & s \end{pmatrix}, \begin{pmatrix} s & -1 \\ 6 & s+4 \end{pmatrix}, \begin{pmatrix} s & 0 \\ 1 & s+4 \end{pmatrix} \quad \text{etc.} \quad (69)$$

$$12s^2 + 24s + 12 = (s+1)^2 = 0 \quad (70)$$

$$s\mathbf{I} - (((\mathbf{H}_{4 \mapsto 1})_i)_j)_k : s+4 : s : s+4 \dots \text{etc.} \quad 24s + 24 = 0 \quad (71)$$

$$\frac{\partial \det(s\mathbf{I} - \mathbf{H}_4)}{\partial s} = \text{tr } \mathbf{adj}(s\mathbf{I} - \mathbf{H}_4) = \sum_{i=1}^4 \det(s\mathbf{I} - \mathbf{H}_{4 \mapsto 3})_i \quad (72)$$

$$\frac{\partial^2 \det(s\mathbf{I} - \mathbf{H}_4)}{\partial s^2} = \sum_{i=1}^4 \frac{\partial}{\partial s} \det(s\mathbf{I} - \mathbf{H}_{4 \mapsto 3})_i \quad (73)$$

$$= \sum_{i=1}^4 \text{tr } \mathbf{adj}(s\mathbf{I} - \mathbf{H}_{4 \mapsto 3})_i \quad (74)$$

$$= \sum_{j=1}^3 \sum_{i=1}^4 \det((s\mathbf{I} - \mathbf{H}_{4 \mapsto 2})_i)_j \quad (75)$$

$$\frac{\partial^3 \det(s\mathbf{I} - \mathbf{H}_4)}{\partial s^3} = \sum_{k=1}^2 \sum_{j=1}^3 \sum_{i=1}^4 \det(((s\mathbf{I} - \mathbf{H}_{4 \mapsto 1})_i)_j)_k. \quad (76)$$

Example 8. Ramification of a sixth- and eighth-order system:

Consider

$$\mathbf{A} = \begin{pmatrix} \text{zeros}(n_1, 1) & \beta \cdot \text{eye}(n_1) \\ & \mathbf{p}^T \end{pmatrix}, \quad (77)$$

either

$$\mathbf{p} = -(64 \ 0 \ 48 \ 0 \ 12 \ 0) \quad \text{for } n_1 = 5 \quad (78)$$

or

$$\mathbf{p} = -(256 \ 0 \ 256 \ 0 \ 96 \ 0 \ 16 \ 0) \quad \text{for } n_1 = 7 \quad (79)$$

in order to observe the eigenvalues close to $s_0 = \pm j2$ of $(s^2 + 4)^3 \doteq 0$ and $(s^2 + 4)^4 \doteq 0$, respectively.

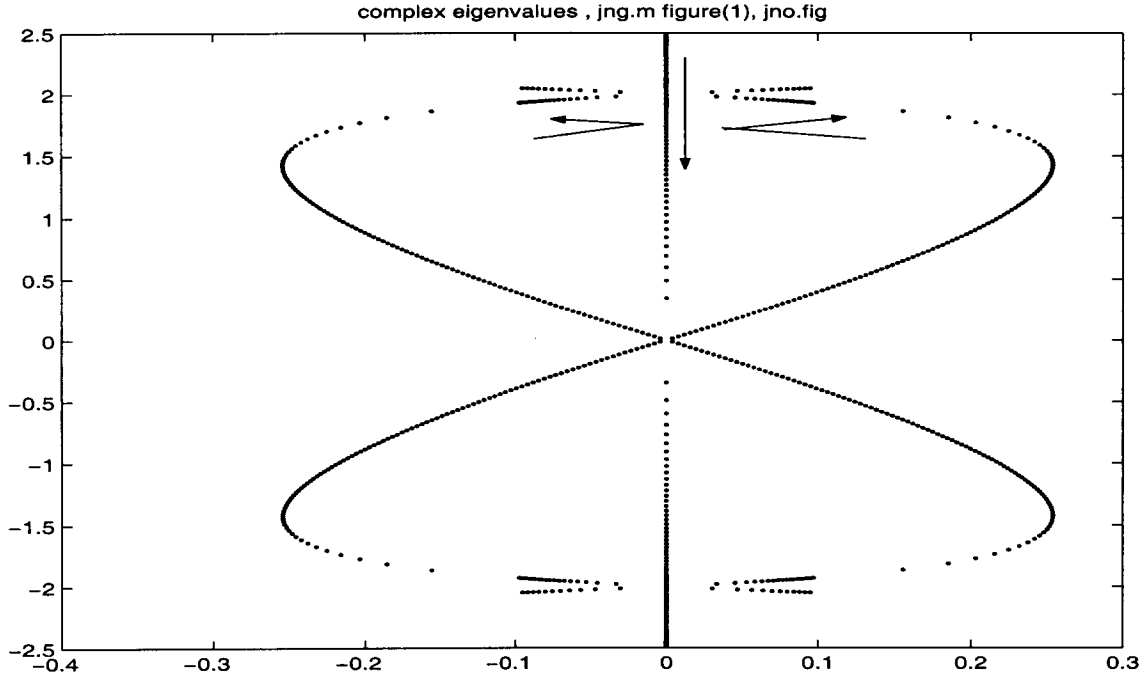


Figure 16: Eigenvalue plot $\lambda_i[\mathbf{A}]$ corresponding to solutions of $(s^2 + 4)^3 \doteq 0$

In Figs. 16 and 17, $\lambda_i[\mathbf{A}]$ are depicted for $0 < \beta < 1$. The behaviour near $\beta = 1$ is explained with the help of arrows. In Fig. 16 there is a three-fold ramification for $(s^2 + 4)^3$ at $\pm j2$ where one solution stays at the imaginary axis. In Fig. 17 there is a four-fold ramification for $(s^2 + 4)^4$. The solution is always termed by a non-zero real part.

E Hamiltonian's Eigenvalues for a Cascade of Four Second-Order Systems

Example 9. Cascade of four PT_2 systems: Consider

$$\frac{V_1 \omega_{N1}}{s^2 + 2D_1 \omega_{N1} s + \omega_{N1}^2} \quad V_1 = 2; \quad \omega_{N1} = 4; \quad D_1 = 0.5; \quad (80)$$

$$\mathbf{A}_1 = (0 \quad 1; -\omega_{N1}^2 \quad -2 D_1 \omega_{N1}); \quad \mathbf{B}_1 = (0; \quad 1); \quad \mathbf{C}_1 = (V_1 \omega_{N1}^2 \quad 0) \quad (81)$$

and the eigenvalues of the corresponding Hamiltonian with $2n = 16$.

For γ small, the sixteen eigenvalues originate from infinity to the centre of the plot. There is a ramification point R at $(0, j2.85)$, i.e., $\omega_P = 2.85$, for $\gamma = 28.4$. For γ big, the branches tend to a region around $(-2, 3.5j)$, see Fig. 18.

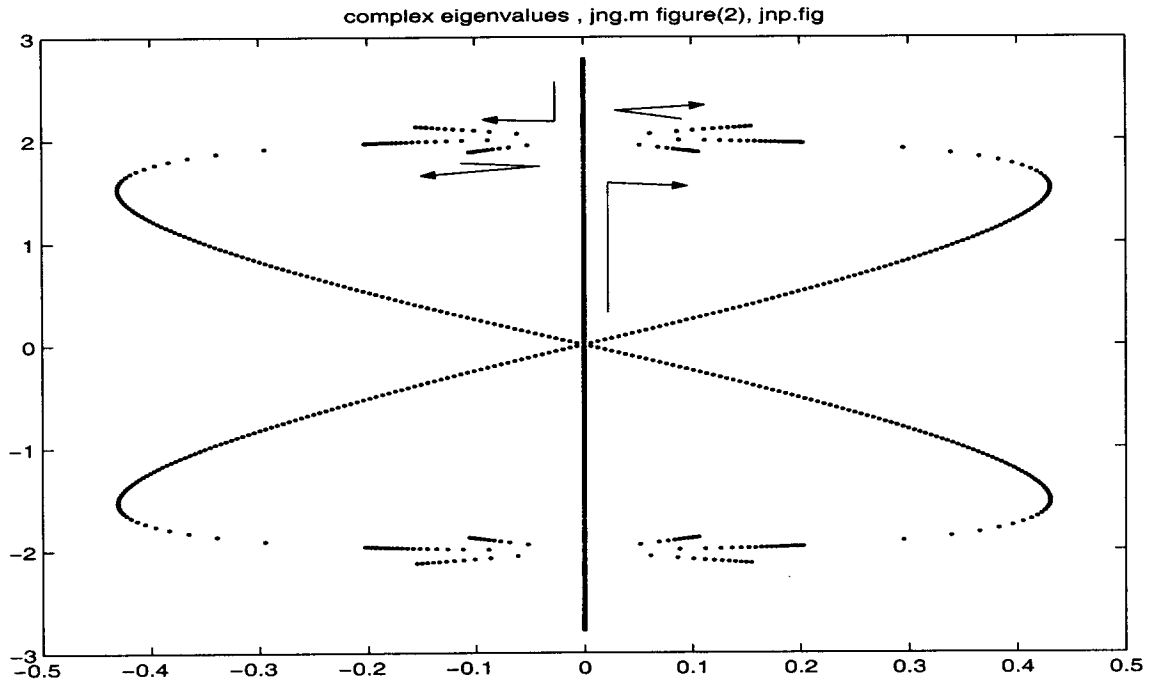


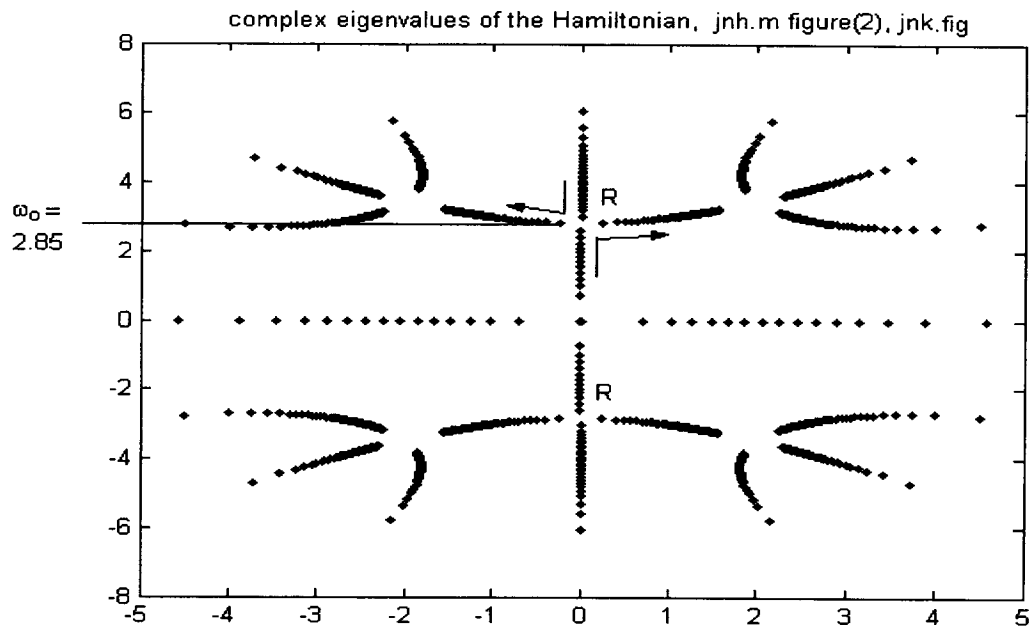
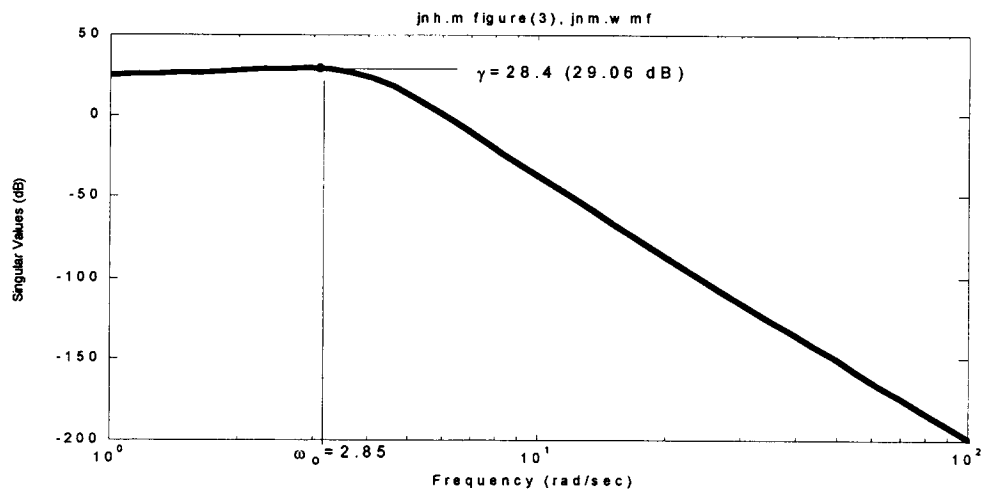
Figure 17: Eigenvalue plot $\lambda_i[\mathbf{A}]$ corresponding to solutions of $(s^2 + 4)^4 = 0$

There is also a ramification point at the origin but it is not relevant for the infinity norm.

The values $\omega_o = 2.85$ and $\gamma = 28.4$ are recalled in Fig. 19. Note that $\det(j\omega_o \mathbf{I} - \mathbf{H})$ is very sensitive to changes in ω_o , see the ordinate in Fig. 20.

Example 10: Example 9 contd., eight PT₂ systems in series:

Now we have $n = 16$ and $\mathbf{H} \in \mathcal{R}^{32 \times 32}$. There are 32 branches of $\lambda_i[\mathbf{H}]$ in Fig. 21.

Figure 18: Eigenvalues $\lambda_i[\mathbf{H}]$ for $n = 8$ Figure 19: Singular value of the cascade of four PT_2 elements

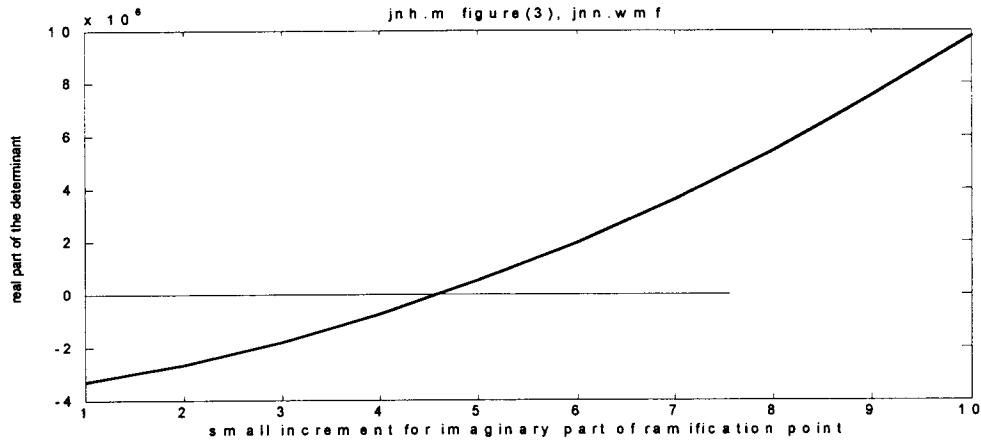


Figure 20: Determinant $\det(j\omega_o \mathbf{I} - \mathbf{H})$ for the Example 9 with $\mathbf{H} \in \mathcal{R}^{16 \times 16}$ versus small increment of ω_o

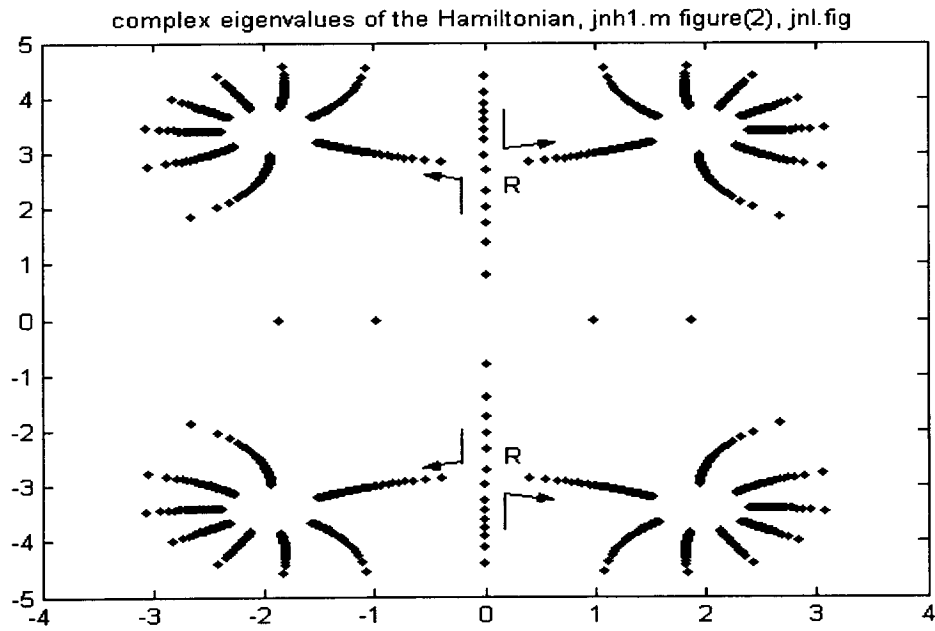


Figure 21: Thirtytwo complex eigenvalues of the Hamiltonian matrix for $n = 16$

References

- Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V., 1994, Linear Matrix Inequalities in Systems and Control Theory (Philadelphia, SIAM Books)
- Doyle, J.C., Glover, K., Khargonekar, P.P., Francis, B.A., 1989, State-space solutions to Standard H_2 and H_∞ control problems, *IEEE Trans.*, vol. AC-34, pp.831-847
- Dong Hu, 1999, Robust reliable H_∞ controllers design via LMIs, *14th Triennial IFAC World Congress, Beijing*, pp. 501-505
- Gahinet, P., Nemirovski, A., Laub, A.J., and Chilali, M., 1995, LMI Control Toolbox for Use with MATLAB (The MATHWORKS, Natick Mass.)
- Gahinet, P., Apkarian, P., and Chilali, M., 1996, Affine parameter-dependent Lyapunov functions and real parametric uncertainty, *IEEE-Trans.* **AC-41**, pp. 436-442
- Kerr, T.H., 1978, Three important matrix inequalities currently impacting control and estimation applications, *IEEE-Trans.* **AC-23**, pp. 1110-1111
- Khargonekar, P.P., Nagpal, K. M., Poolla, K.R., 1991, H_∞ control with transients, *SIAM J. Control and Optimization* **29**, pp. 1373-1393
- Ludyk, G., 1995, Theoretische Regelungstechnik 2 (Springer, Berlin)
- Weinmann, A., 1991, Uncertain Models and Robust Control (Springer, New York and Vienna)
- Weinmann, A., 1996, H_∞ Facilities in Robust Control, *Int. J. for Automation Austria* **4**, p. 73-103
- Weinmann, A., 1997, Identification based on H_2 performance and the LMI approach. 13. Österreichischer Automatisierungstag.
- Weinmann, A., 1997a, H_∞ Controllers: Hamiltonian versus LMI Approach, Tagungsband des 10. Steirisches Kolloquiums über Regelungstechnik und Prozessautomatisierung in Retzhof, S. 32-53
- Weinmann, A., 1999, Computerunterstützung für Regelungsaufgaben (Springer, Wien NewYork)
- Weinmann, A., 2001, Norm constraint regard for stable H_2 controllers, *e & i, Elek-*

- trotechnik und Informationstechnik* **118**, H. 11, pp.617-621
- Weinmann, A., 2003, Beobachter minimaler Norm, *Tagungsband des 13. Steirischen Seminars über Regelungstechnik und Prozeßautomatisierung in Retzhof*, S. 1-20
- Weinmann, A., 2004, Optimal assignment of multiple control system eigenvalues with minimum controller norm, *e & i (Elektrotechnik und Informationstechnik)* **121**, H.2, to appear
- Weinmann, A., 2004a, Gain-scheduled H_∞ controller with minimum Frobenius norm, *e & i (Elektrotechnik und Informationstechnik)* **121**, H.2, to appear
- Weinmann, A., 2004b, Multi-Lagrange-Multiplier Problems: Desensitized pole assignment with minimal controller matrix norm for networked control systems and constraints in high-order mechatronics, *Mathematical and Computer Modelling of Dynamical Systems*, **9**, to appear

**Ausschreibung
des
Fred Margulies-Preises**

für Forschungsarbeiten auf dem Gebiet der

Automatisierungstechnik

mit besonderer Berücksichtigung der sozialen Aspekte

Anlässlich des zehnjährigen Bestehens des IFAC-Beirates Österreich wurde von diesem zusammen mit der Gewerkschaft der Privatangestellten der Fred Margulies-Preis ins Leben gerufen. Dieser Preis wird jährlich (2004 zum zehnten Mal) für hervorragende österreichische Arbeiten auf dem Gebiet der Automatisierungstechnik, mit besonderer Berücksichtigung der sozialen Aspekte, verliehen. Kriterien für die Bewertung sind im wesentlichen Innovationsgehalt, Wissenschaftlichkeit, wirtschaftliche Bedeutung sowie Interdisziplinarität des Ansatzes und soziale Relevanz. Der Preis in der Höhe von

€ 4.400,-- pro Jahr

wird je zur Hälfte von den beiden Stiftern geleistet und von einer Jury vergeben, welcher neben Persönlichkeiten aus Wissenschaft und Wirtschaft auch Vertreter der Gewerkschaft angehören.

Der Fred Margulies-Preis wird am 7. Oktober 2004 im Rahmen der SMART AUTOMATION AUSTRIA - Fachmesse für industrielle Automation – im Design Center Linz, Europaplatz 1, verliehen.

Eingereicht können Studien (auch Diplomarbeiten und Dissertationen) werden, welche innerhalb der letzten 5 Jahre in Österreich abgeschlossen wurden. Diese Arbeiten sind bis spätestens 31. Juli 2004 in einfacher Ausfertigung zusammen mit einer Kurzfassung (max. 10 Seiten) beim

Generalsekretariat des IFAC-Beirates c/o Institut für Handhabungsgeräte- und Robotertechnik der Technischen Universität, Favoritenstraße 9-11, 1040 Wien
(Tel. 58801-31801; E-mail: nemetz@ihrt.tuwien.ac.at)

einzureichen.

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

Algorithmic Foundations of Robotics V

J.-D. Baisonnat et al. (Eds.)

Springer-Verlag Berlin Heidelberg, 2004

757 Seiten, 226 Abbildungen

ISBN 3-540-40476-7

Das Bild des Roboters wandelte sich in den letzten zwei Dezennien vom untelligenten stationären Industrieroboter zum mobilen intelligenten kooperativen Roboter für neue Anwendungsgebiete. Diese neue Robotergeneration dringt immer mehr in menschliche Lebensbereiche, beispielsweise in Form von Servicerobotern für Aufgaben im Haushalt, in der Gastronomie, im Hotelgewerbe, in der Landwirtschaft ... sowie darüber hinaus in spezielle Anwendungsbereiche wie Landminensuche und -entfernung, neue Weltraum- und Unterwasseranwendungen.... ein.

Klarer Weise sind für diese neue Robotergeneration neue wissenschaftlich-technische Grundlagen und Methoden erforderlich. Der vorliegende 5. Band in dieser Serie stellt eine Sammlung solcher Grundlagen, wie sie auf dem 5. Internationalen Workshop über „Algorithmic Foundations of Robotics“ vom 15.-17. Dezember 2002 in Nizza präsentiert wurden. Neben zwei Übersichtsaufsätzen „Algorithms for Motion and Navigation in Virtual Environments and Games“ sowie „Surgery Simulation: Integrating Geometry, Physics and Physiology into the Virtual Patient“ sind die 32 enthaltenen Fachbeiträge in 8 Themenbereiche zusammengefaßt. Die Beiträge stellen neue Erkenntnisse hinsichtlich der Bahnplanung, rekonfigurierbarer Roboter, Koordination von Mehrrobotersystemen, Visualisierung, sowie molekulare Modellierung dar.

Das vorliegende Sammelwerk stellt daher eine äußerst aktuelle Sammlung von neuen bzw. bekannten und bisher nicht in der Robotertechnik angewendeten algorithmischen Methoden dar. Es ist daher allen, welche sich mit diesem Fachgebiet beschäftigen, wärmstens zu empfehlen.

P. Kopacek

Regelungstechnik

Mathematische Grundlagen, Entwurfsmethoden, Beispiele

H. P. Geering

Springer-Verlag Berlin Heidelberg, 2004
6., neu bearbeitete und ergänzte Auflage
338 Seiten, 117 Abbildungen
115 Aufgaben und Lösungen
ISBN 3-540-40507-0

Dieses Buch entstand aus einem Grundlagen-Vorlesungsskript des Autors an der Abteilung für Maschinenbau und Verfahrenstechnik der ETH Zürich. Die Ziele des Buches sind die Förderung des Verständnisses für dynamische Vorgänge in Prozessen, Sensoren, Aktoren und Gesamtsystemen sowie die Befähigung zur Analyse linearer Mehrgrößensysteme im Zeit- und Frequenzbereich sowohl im deterministischen als auch stochastischen Fall. Darüber hinaus soll das Buch den Leser befähigen von wichtigen klassischen und modernen Methoden für den Entwurf von robusten Ein- bzw. Mehrgrößenreglern Gebrauch machen zu können.

Das Buch beginnt mit der Erklärung der Bausteine und Signale in einem Regelsystem anhand von Signalflussbildern. Der Analyseteil, welcher auf mehrere Kapitel verteilt ist, umfasst das dynamische Verhalten von linearen, zeitinvarianten Systemen mit Hilfe der klassischen Methode der Laplace-Transformation. Dabei werden – wie üblich – Übertragungsfunktion und Frequenzgang eingeführt. Unüblich für ein Grundlagenwerk ist die Behandlung von zeitvariablen Systemen sowie stochastischen Signalen in Form der Autokovarianzfunktion und Autokovarianzmatrix.

Der Synthese-Teil umfasst klassische Methoden des Entwurfs zeitkontinuierlicher Regler und die Analyse der Stabilität von Regelsystemen. Ebenfalls unüblich für eine Grundlagenvorlesung ist die Einführung des Luenberger-Beobachters sowie die Darstellung des Kalman-Bucy-Filters.

Jedes Kapitel schließt mit einer Sammlung von ausgewählten Aufgaben ab, wobei die Lösungen zu diesen am Ende des Textes angegeben sind. Das Buch schließt mit einem Anhang der für die Regelungstechnik wichtigsten Grundlagen, wie komplexe Zahlen, lineare Algebra, Linearisierung von Differentialgleichung und Wahrscheinlichkeitstheorie.

Das vorliegende Lehrbuch stellt sicher eine sehr weitreichende Einführung in die Grundlagen der linearen Regelungstechnik dar. Es kann daher einem etwas theoretisch orientierten Leserkreis wärmstens empfohlen werden.

P. Kopacek

Leitfaden der technischen Informatik und Kommunikationstechnik

Kopacek, P. und Zauner, M.

Springer Wien, 2004, 1. Auflage

271 Seiten

ISBN 3-211-00765-2

Dem technisch orientierten Nutzer wird eine praxisorientierte Einführung in die Informatik bzw. Informationstechnik gegeben. Die Theorie wird nur insofern gestreift, als dies für das Verständnis des Stoffes unbedingt erforderlich ist. Das Schwergewicht liegt auf der praktischen Anwendbarkeit und Beispielen aus den Ingenieurwissenschaften.

Das Buch ist didaktisch als Lehrbuch aufgebaut. Insbesondere die Heranführung an das Thema, Erklärungen, Beispiele und Kapitelzusammenfassungen erlauben ein einfaches und selbständiges Erlernen des Stoffes. Ausführliche Illustrationen sowie entsprechende Tabellen, Beispiele und Graphiken unterstützen den Leser.

Trotz des eingeschränkten Umfangs des Buches schaffen es die Autoren, über das Wissen der schier unendlich großen Fachgebiete der Informatik und der Kommunikationstechnik einen Überblick zu geben. Dabei kommen neben den klassischen Themen der Rechner- und Softwarearchitekturen, des Engineerings, der Datenbanken und der Einsatzplanung moderne Technologien wie Mechatronik, Nanotechnologie und Künstliche Intelligenz nicht zu kurz.

Das Buch ist damit eine Basis für den (zukünftigen) Manager, der über die Belange der elektronischen Datenverarbeitung und den damit verbundenen Kommunikationstechniken zu entscheiden hat.

K. Daichendt

System architecture for cost-effective automation of complex enterprise-spanning business processes

Dr. Thomas Schmidt

Begutachter: Prof. Gerfried Zeichen
Prof. Alexander Weinmann

Gesteigerter Wettbewerb im globalem Markt und beschleunigte Entwicklungen der Technologien erhöhen den Druck auf die Unternehmen, hochqualitative Produkte und Dienstleistungen mit immer kürzerer "time-to-market" und "time-to-money" zu liefern. Die Effizienz lässt sich mit der Integration und Automation von internen und unternehmensübergreifenden Geschäftsprozessen steigern. Die logische und konsequente Weiterentwicklung der Kommunikation im Bereich e-Business führt zu Extended Enterprises. Darunter versteht man unabhängige Firmen, die sich zusammenschließen und dadurch deren Wettbewerbsfähigkeit steigern.

Das Ziel dieser Dissertation ist die Entwicklung und Implementierung eines Softwaresystems, das die vollautomatische Zusammenarbeit zwischen Geschäftspartnern ermöglicht. Dafür muss das neu entwickelte System durch die Verwendung einer ganzheitlichen und flexiblen Systemarchitektur die derzeit vorhandenen isolierten Integrationsinseln innerhalb der Firmen - entstanden durch den Einsatz unterschiedlicher Unternehmenssoftware, die jeweils nur einen Teil des Produktlebenszykluses abdecken kann - hinter sich lassen.

Die Hauptkomponente des vorgestellten Ansatzes ist die Business Process Engine, die die inner-/zwischen-organisatorischen Geschäftsprozesse der Extended Enterprise abarbeitet. Bei der Ausführung einer Prozessaktivität werden Web Services verwendet, die von Unternehmenssoftwaresystemen und/oder Teilprozessen zur Verfügung gestellt werden. Ziel ist die Unterstützung des gesamten Produktlebenszykluses. Diese auf Web Services basierende Softwarearchitektur ermöglicht erstmals neben der horizontalen Integration zwischen den einzelnen Geschäftspartnern auch die durchgängige vertikale Integration durch alle Unternehmensebenen.

Die Verwendung von Open-Source Softwarekomponenten ist eine gebräuchliche Methode, um die Anforderung "kostengünstig" zu erfüllen. Damit die bestgeeignetste und ausgereifteste Workflow Engine als Basis für die Business Process Engine ausgewählt werden kann, ist im Rahmen dieser Arbeit eine ausführliche Studie durchgeführt worden. Aufgrund von aufgestellten Entscheidungskriterien fiel die Wahl auf die Workflow Engine "OFBiz", die in dieser Arbeit weiterentwickelt und in die Softwareumgebung eingebunden wird.

Der letzte Teil der Dissertation umfasst die Realisierung eines Prototypen, der im Rahmen des dreijährigen Europäischen Forschungs- und Entwicklungsprojekts "FLoCI-EE" entstanden ist. Für diesen wird ein beispielhaftes und praxisnahes Geschäftsszenario zuerst modelliert und dann implementiert. Die wichtigsten Realisierungsergebnisse und eine grobe Wirtschaftlichkeitsabschätzung des entwickelten Softwaresystems schließen die Dissertation ab.

Qualitätssicherung und die Erstellung von Qualitätssicherungssystemen an Organisationen für die universitäre Ausbildung von IngenieurInnen

Dipl.-Ing. Jorge Martin Bauer

Begutachter: O.Univ.Prof.Dr. P.H. Osanna

O.Univ.Prof. Dipl.-Ing. Dr.techn. Dr.h.c.mult. Peter Kopacek

Die Zielsetzung dieser Dissertation umfasst die Problematik der "Qualität der Universitären Lehre auf dem Gebiet der Ingenieur-Wissenschaften" und die Problematik der "Prozess-Steuerung bei der Integrierung der Kenntnisse" im Rahmen der universitären Ausbildung. Vom Blickpunkt der Qualitätssicherung und Erstellung von Qualitätssicherungssystemen an Organisationen für die universitäre Ausbildung von IngenieurInnen, wird diese Themenstellung _ insbesondere im Zusammenhang mit den Vorgaben der internationalen Normenbereiche ISO 9000 – ins-Detail gehend untersucht und in umfassender Weise diskutiert.

Die gegenwärtig aktuellen Globalisierungsprozesse und die Forderung von Regierungen, Universitäten und der menschlichen Gesellschaft im Allgemeinen nach mehr „Qualität“ in der universitären Lehre der IngenieurInnen und allgemein der AkademikerInnen, bilden einen der Ausgangspunkte für diese Arbeit. Anhand des konkreten Mit-Lebens auf dem Sektor der „universitären Lehre der IngenieurInnen“ und des Erkennens von neuen Problemen wurden Erfahrungen und theoretische Beiträge systematisch bearbeitet, und die Resultate dieser Analyse wieder auf das praktische, konkrete Aufbauen eines Qualitätssicherungssystems für die eigene Organisation als interaktiver Prozess widergespiegelt. Die theoretischen Grundideen, die praktischen Erfahrungen und das Konzept einer „Prozess-Orientierung bei der Gestaltung einer Lern- und Lehrumgebung“ werden in den ersten Kapiteln systematisch bearbeitet.

Qualitätssicherungssysteme für die „Universitäre Lehre auf dem Gebiet der Ingenieur-Wissenschaften“ sind sicher nicht „Ziel an sich“, sondern Werkzeuge zur Gestaltung einer besseren Lern- und Lehrumgebung. Erkennen und debattieren, welche Eigenschaften Lern- und Lehrprozesse in diesem Rahmen haben sollen, ist ein grundlegender Beitrag in Richtung auf die Gestaltung einer „Qualitäts-Lernumgebung“. Wenn wir zustimmen, dass das gesuchte Ziel nicht darin besteht, einen „Text-Wiederholer“ zu erzeugen, sondern kreative Fachkräfte auszubilden, welche Probleme schrittweise begrenzen und systematisch begrenzen und lösen können, dann muss der „Prozess“ zur „Integrierung der einzelnen Teilausbildungen“ führen. Hierbei wird die Meinung vertreten, dass die Integration der erworbenen Kenntnisse viel mehr ist als die algebraische Summe aller Teilvorlesungen bzw. Teilkenntnisse. Das Qualitätssicherungssystem muss sich von diesem Blickpunkt aus nicht nur auf den „Kenntnisübertragungs-Prozess“, sondern immer stärker auf den „Kenntnisintegrations-Prozess“ und auf das Schaffen einer problemorientierten Lernumgebung konzentrieren.

Da es gemäß eigener Erfahrung und auch gemäß mehreren in der Literatur gefundenen Referenzen für eine problemorientierte Lern- und Lehrumgebung an geeigneten Prüfungsinstrumenten fehlt, und da laut Deming, dem Vater der modernen Qualitätslehre, im ständigen Verbesserungskreis-Prozess, eine objektive Beurteilungs-Phase (Evaluation) benötigt wird, beschäftigt sich das grundlegende Kapitel dieser Arbeit mit der Darlegung eines konkreten Vorschlages für ein Bearbeitungsmodell. Ein explizites Verfahren für eine „systematische Steuerung und Evaluation“ des „Kenntnisintegrations-Prozesses“ mit Hilfe einer innovativen, mathematisch fundierten Methode – der Fuzzy-Logik-Technik – als Beitrag zur Anwendung des PDCA- bzw. PDSA-Kreises auf die beschriebene Problematik wurde erarbeitet.

Die beiden letzten Kapitel beschäftigen sich mit dem Einsatz von „High Technology Labors“. Der Aufbau in einer geeigneten „Lern- und Lehrumgebung“ bietet nicht nur die Möglichkeit, spezifische Technologien kennen zu lernen, sondern auch die, den „Kenntnisintegrations-Prozess“ bei der Ausbildung junger AkademikerInnen der Ingenieurwissenschaften zu fördern und systematisch zu steuern.

Spezielle Probleme der Messtechnik und Datenkommunikation bei der Qualitätssicherung in hochtechnologischen Bereichen gezeigt an Beispielen aus der Nanotechnologie und der Biomedizintechnik

Dipl.-Ing. Liangxin Si

Begutachter: O.Univ.Prof.Dr. P.H. Osanna

O.Univ.Prof. Dipl.-Ing. Dr.techn. Dr.h.c.mult. Peter Kopacek

Nach einer Themenkurzbeschreibung werden im Teil 2 „Datenkommunikation im CAD/CAQ-System“ die Problemstellungen zur Datenkommunikation und –austausch zwischen der Konstruktion und fertigungsintegrierter Qualitätssicherung, insbesondere zur messgerätefernen Programmierung von Koordinatenmessgeräten wie CAQMESS, zur DMIS-Schnittstelle, zur Low Cost-CAD/CAM/CAQ-Systemarchitektur sowie zum zukünftigen simultanen Produktdatenautauschmodell in der simultanen Produktionsumgebung behandelt.

Im Teil 3, Qualitätssicherungs-, Produktionssystem und Datenkommunikation im modernen Industriebetrieb MFIF, ist ein innovatives Unternehmensmodell bzw. –konzept MFIF für zukünftige „customer-driven“ MFPs entwickelt und präsentiert. Dabei wird eine neue Methodik für Produktqualitätssicherung und ein neues Produkt- und Unternehmensverhältnis für agile Fertigung und Produktion in der globalen Produktionsumgebung mit Unterstützung durch die Datenautobahn, DCE-Technologie und STEP-Schnittstelle dargestellt. In dieser Arbeit werden so die konzeptuelle Architektur von MFIF, die Eigenschaften und das Arbeitsprinzip von IPSs in MFIF sowie die Basistechnologien – DCE und STEP präsentiert und diskutiert. Ein gemeinsames Modell des simultanen Datenaustausches zwischen allen globalen Produktionsprozessen ist vorgestellt.

Im Teil 4 wird Qualitätssicherung in der Biomedizintechnik vorgestellt. Primär wird gezeigt, dass die Qualitätsmodellierungen – die genau mathematischen, geometrischen und kinematischen Modelle von menschlichen Kniegelenken mit Hilfe von CAD für die Herstellung der Qualitätsendoprothesenprodukte entwickelt sind.

Teil 5 beschäftigt sich mit der Qualitätssicherung in der Nanotechnologie. Hier werden die Problemstellungen in der Nanotechnologie und –metrologie vorgestellt, untersucht, analysiert und diskutiert. Der Qualitätssicherung in der Nanotechnologie wurde speziell eine Arbeit gewidmet, die sich mit der Atomkraftmikroskopie befasst. Im Rahmen dieser Spezialarbeit wurden Untersuchungen zu den Image-Topographien der Mikrostrukturen von Nano-Oberflächen mittels AFM durchgeführt, die Ergebnisse ausgewertet und auch dargestellt. Die Analysen und Diskussionen über die Ergebnisse und die AFM-Artifakte, sowie die daraus resultierenden Verbesserungsvorschläge sind gegeben. Eine praktische und kostengünstige Kalibrierungsmethode ist auch vorgestellt.

Teil 6 - Zusammenfassung – bildet den Abschluss und eine logische Zusammenstellung der Schwerpunkte für weitere zukünftige Forschungen.

Vorankündigung

19. Österreichischer Automatisierungstag

im Rahmen der SMART AUTOMATION AUSTRIA

Fachmesse für industrielle Automation

7. Oktober 2004

10.00 – 14.00 Uhr

DESIGN CENTER LINZ

Linz, Europaplatz 1

Am 19. Österreichischen Automatisierungstag werden in Übersichtsvorträgen aktuelle Gebiete der Automatisierungstechnik verständlich dargestellt. Weiters werden in Form von Kurzvorträgen zukünftig interessante Teilgebiete sowie Problemlösungen präsentiert.

Im Rahmen dieses 19. Österreichischen Automatisierungstages findet auch heuer wieder die Verleihung des Fred Margulies-Preises statt, welcher für hervorragende österreichische Arbeiten auf dem Gebiet der Automatisierungstechnik, mit besonderer Berücksichtigung der sozialen Aspekte, verliehen wird.

Die Teilnahme am Automatisierungstag ist kostenlos, eine Voranmeldung ist jedoch erforderlich.

Anmeldungen an:

Institut für Handhabungsgeräte und Robotertechnik, TU Wien

Favoritenstraße 9 – 11, 1040 Wien

Tel.: 01/58801-31801

Fax: 01/58801-31899

E-mail: nemetz@ihrt.tuwien.ac.at

Während sowie nach dem Ende des Automatisierungstages können die Teilnehmer die SMART AUTOMATION AUSTRIA kostenlos besuchen.

Es erfolgt noch eine gesonderte Aussendung mit dem detaillierten Programm.

<i>Datum</i>	<i>Veranstaltung</i>	<i>Ort</i>	<i>Weitere Informationen erhältlich bei:</i>
5.-7.4.2004	11th IFAC Symposium Information Control Problems in Manufacturing – INCOM2004	Salvador Brazil	e-mail: cpereira@eletro.ufrgs.br http://www.eletro.ufrgs.br/incom2004/
19.-23.4.2004	IFAC Symposium Advances in Automotive Control	Salerno Italy	e-mail: ifac04.unisa.it http://ifac04.unisa.it
2.-5.6.2004	13th Intern. Workshop on Robotics in Alpine-Adria-Danube Region	Brno Czech Republic	e-mail: RAAD04@uvss.fme.vutbr.cz http://www.uvssr.fme.butbr.cz/RAAD04
7.-9.6.2004	IFAC Symposium Cost Oriented Automation	Gatineau Canada	e-mail: heinz.erbe@tu-berlin.de http://www.tu-berlin/~metel/ifac-lca
16.-18.6.2004	Intern.IARP Workshop On Robotics and Mechanical assistance in Humanitarian Demining and similar risky interventions	Brussel, Leuven Belgium	e-mail: kopacek@ihrt.tuwien.ac.at http://www.ihrt.tuwien.ac.at/HUDEM04 http://mecatron.rma.ac.be
21.-23.6.2004	IFAC Symposium Telematics Applications in Automation and Robotics –TA2004	Helsinki Finland	e-mail: office@atu.fi http://www.automaatioseura.fi/TA04
30.6.-2.7.2004	American Control Conference – in co-operation with IFAC	Boston, MA USA	e-mail: speyer@seas.ucla.edu http://www.mie.uiuc.edu/acc2004/index.asp
5.-7.7.2004	IFAC Symposium Intelligent Autonomous Vehicles- IAV2004	Lisbon Portugal	e-mail: iav04@isr.ist.utl.pt http://www.iav04.isr.ist.utl.pt

20.-23.7.2004	5th Asian Control Conference – in co-operation with IFAC	<i>Melbourne Australia</i>	e-mail: ascc@ee.mu.oz.au http://ascc2004.ee.mu.oz.au
22.-27.8.2004	18th IFIP World Computer Congress	<i>Toulouse France</i>	e-mail: dervillers@wcc2004.org http://www.wcc2004.org
2.-4.9.2004	IFAC Multitrack Conference Advanced Control Strategies for Social and Economic Systems – ACS04	<i>Vienna Austria</i>	e-mail: kopacek@ihrt.tuwien.ac.at http://www.ihrt.tuwien.ac.at/acs04
27.-29.9.2004	Intern. Conference on Information Technology for Balanced Automation Systems in Manufacturing and Services – BASYS04	<i>Vienna Austria</i>	http://www.ifs.tuwien.ac.at/basys2004
29.9.-1.10.2004	8th Fuzzy Days – International Conference on Computational Intelligence	<i>Dortmund Germany</i>	e-mail: fd8@lsl.cs.uni-dortmund.de
3.-5.10.2004	IFAC DECOM TT WS Automatic Systems for Building the Infrastructure in Developing Countries	<i>Bansko Bulgaria</i>	e-mail: decom04@hsi.iccs.bas.bg
7.10.2004	19. Österreichischer Automatisierungstag	<i>Linz Austria</i>	e-mail: nemetz@ihrt.tuwien.ac.at http://ihrt.tuwien.ac.at/IFAC_AUT/default.htm
19.-20.11.2004	IFAC Workshop on Intelligent Assembly and Disassembly – IAD04	<i>Vienna Austria</i>	e-mail: kopacek@ihrt.tuwien.ac.at http://ihrt.tuwien.ac.at/iad04
2.-4.12.2004	CLAWAR – Robots in Entertainment, Leisure and Hobby	<i>Vienna Austria</i>	e-mail: kopacek@ihrt.tuwien.ac.at http://www.ihrt.tuwien.ac.at/HUDEM