



LUND UNIVERSITY

The CACE Project -- Steering Committee Meeting 2, 1985-09-06

Mattsson, Sven Erik; Åström, Karl Johan

1986

Document Version:

Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):

Mattsson, S. E., & Åström, K. J. (1986). *The CACE Project -- Steering Committee Meeting 2, 1985-09-06*. (Technical Reports TFRT-7321). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:

2

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

CODEN: LUTFD2/(TFRT-7321)/1-098/(1986)

The CACE Project - Steering Committee Meeting 2

Sven Erik Mattsson
Karl Johan Åström

Department of Automatic Control
Lund Institute of Technology
April 1986

Department of Automatic Control Lund Institute of Technology P.O. Box 118 S-221 00 Lund Sweden		Document name Report	
		Date of issue 1986-04-28	
		Document Number CODEN:LUTFD2/(TFRT-7321)/1-098/(1986)	
Author(s) Sven Erik Mattsson Karl Johan Åström		Supervisor	
		Sponsoring organisation The Swedish Board of Technical Development	
Title and subtitle The CACE project - Steering committee meeting 2, 1985-09-06			
Abstract <p>This report contains documentation handed out to the participants of the 2nd steering committee meeting of the STU Computer Aided Control Engineering Programme (CACE) on September 6, 1985. The minutes of the meeting are also included.</p>			
Key words Computer Aided Control Engineering			
Classification system and/or index terms (if any)			
Supplementary bibliographical information			
ISSN and key title			ISBN
Language English and Swedish	Number of pages 98	Recipient's notes	
Security classification			

PREFACE

This report contains documentation handed out to the participants of the 2nd steering committee meeting of the STU Computer Aided Control Engineering Programme (CACE) on September 6, 1985. The minutes of the meeting are also included.

Table of Contents

AGENDA	5
PROJECT OVERVIEW	
View graphs (in Swedish), Karl Johan Åström	6
REVIEW OF CURRENT CACE PROJECTS	
New Forms of Man-Machine Interaction	
View graphs (in Swedish), Sven Erik Mattsson	13
Experiment with Knowledge Based Man-Machine Interfaces	
View graphs, Jan Eric Larsson and Per Persson	25
NEW CACE PROJECTS	
Combination of Formula Manipulation and Numerics	
View graphs (in Swedish), Sven Erik Mattsson	38
High Level Problem Solving Languages	
View graphs (in Swedish), Karl Johan Åström	41
Expert Control	
View graphs, Karl-Erik Årzén	45
Expert Control (in Swedish) Karl-Erik Årzén and Karl Johan Åström	56
INTERNATIONAL CONTACTS	
View graphs (in Swedish), Karl Johan Åström	68
Some Interesting International CACE Software Efforts Sven Erik Mattsson and Karl Johan Åström	73
MINUTES OF THE MEETING	97

CACE PROJEKTET - STYRGRUPPSSAMMANTRÄDE 2

6 september 1985

1. Projektläge översikt

Tid och kostnadsläge

Personalläge

Svårt men ej omöjligt

Allmänna intryck

Bra team

Börjar fungera

Mer Lisp än vi trodde från början

Diskussion om ändrad tidsplan

2. Läge delprojekt 1 - Nya former för MMC

3. Läge delprojekt 2 - Expertsysteminterface

4. Planerade projekt

Formelbehandling

Beslut om ansökan

Högnivåspråk för problemlösning

Beslut om ansökan

Implementeringsspråk

Expertreglering

Beslut om ansökan

Lispmaskin?

5. Internationella kontakter

Policy för samarbete och programbyte

6. Demonstrationer

HÖGNIVÅSPRÅK

SYMBOLISKA BÄR

EXPERT CONTROL

EXPERT SYSTEM
INTERFACE

NYA FORMER
AV MAN-MASKIN
KOMMUNIKATION

↓

↓ Ulf Holmberg

↓ Knut Erik Årzén

↓ Jan Eric Larsson
Per Persson

↓ IRIS
Konsult
Dag Brück
Sven Erik Mattson

↓

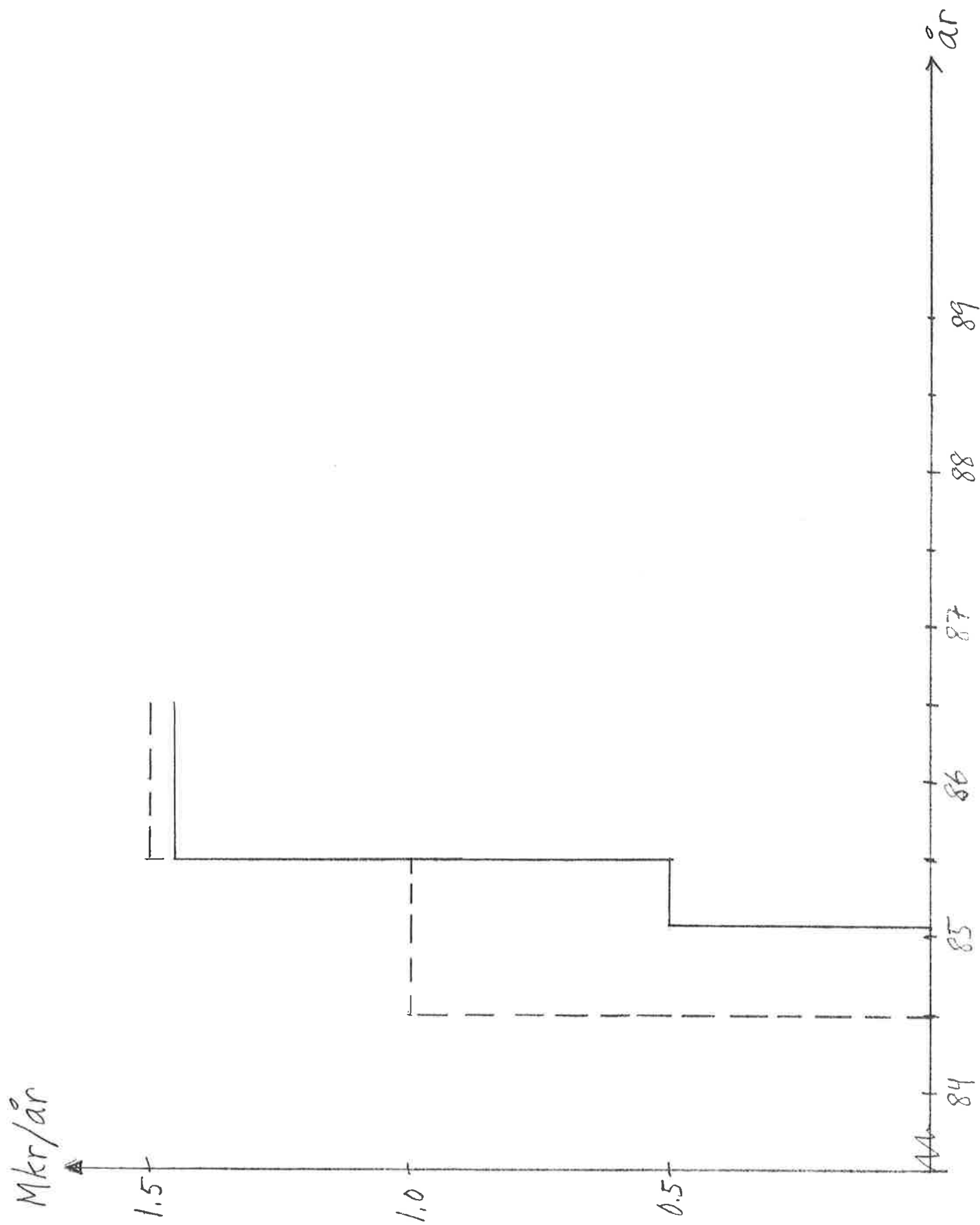
STU BESLUT
↓

Bakgrundsarbete Institutionspersonal

J A S O N D | J F M A M J J A S O N D

1984

1985



PERSONAL

STU Finansierad

Beviljat

Sven Erik Mattsson	100% STU
Per Persson	75% STU
Jan Eric Larsson	75% STU
Dag Brück	30% STU

Ansökt

Karl Erik Årzén	100% STU
Ulf Holmberg	50% STU
Mats Lilja	25% STU

LTH Finansierad

Karl Johan Åström	20%
Per Hagander	20%
Leif Andersson	20%
Tomas Schönthal	40%

Gästforskare och konsulter

Wolfgang Kreutzer
Hilding Elmqvist

BUDGET

Enligt liggande ramprogram:

År	Utrustning	Löner material resor
1984/85	800 kkr	1000 kkr
1985/86	500 kkr	1500 kkr
1986/87	100 kkr	1500 kkr
1987/88	50 kkr	1500 kkr
1988/89		500 kkr
	<hr/>	<hr/>
Summa:	1450 kkr	6000 kkr

Nytt förslag:

År	Utrustning	Löner material resor
1984/85	800 kkr	200 kkr
1985/86	500 kkr	1400 kkr
1986/87	100 kkr	1500 kkr
1987/88	50 kkr	1500 kkr
1988/89		1400 kkr
	<hr/>	<hr/>
Summa:	1450 kkr	6000 kkr

KOPPLING TILL UTBILDNING

Doktorandkurs

Ideer

Gå igenom existerande paket

Tänkt efter

Examensarbeten

Doktorandprojekt

Flerbetygsuppsatser

Liz projekt

Dr projekt

TRE BESLUT

- ✿ KOMBINATION AV FORMEL-BEHANDLING OCH NUMERIK
- ✿ HÖGNIVÅSPRÅK FÖR LÖSNING AV REGLERPROBLEM
- ✿ EXPERTREGLERING

VÄGLEDNING OM

- ✿ HANTERING AV PROGRAMUAGA
- ✿ SAMARBETE MED ANDRA GRUPPER
- ✿ NÄSTA ARBETSSTATION

NYA FORMER FÖR MAN-MASKINKOMMUNIKATION

Man-maskinsnittet är en mycket viktig del av ett CACE-system.

De nya arbetsstationerna med realtidsgrafik med hög prestanda ger nya möjligheter.

Avsikten med detta projekt är att sätta upp ett prototypsystem så att ideer kan testas och så att vi kan få erfarenhet av att använda grafik.

PILOTPROJEKT

Simulator för dynamiska system

Grafisk beskrivning av modellstrukturen

Hierarkiska blockdiagram

Informationszoomning

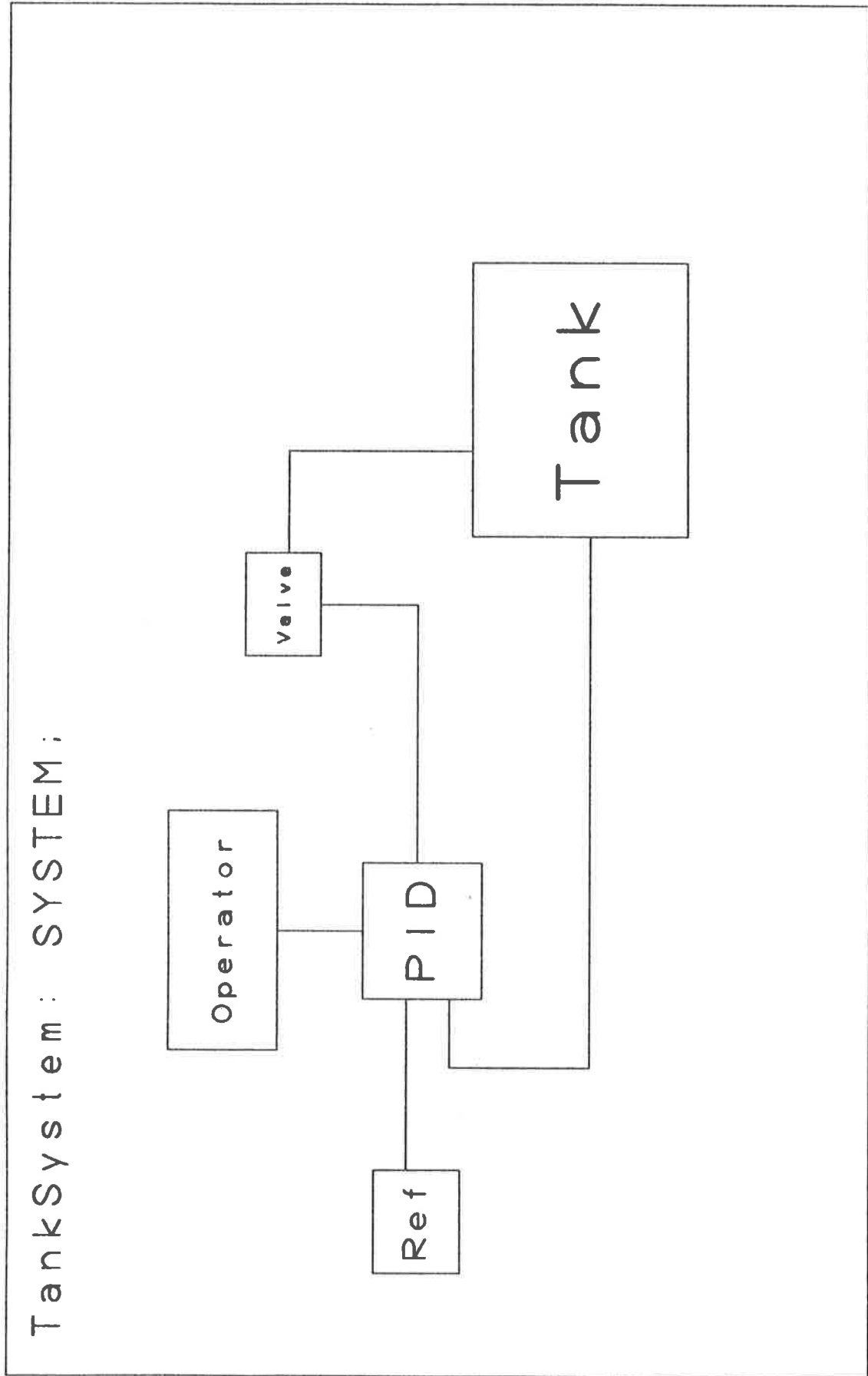
Ekvationsbaserade delmodeller

Symbolisk formelmanipulering

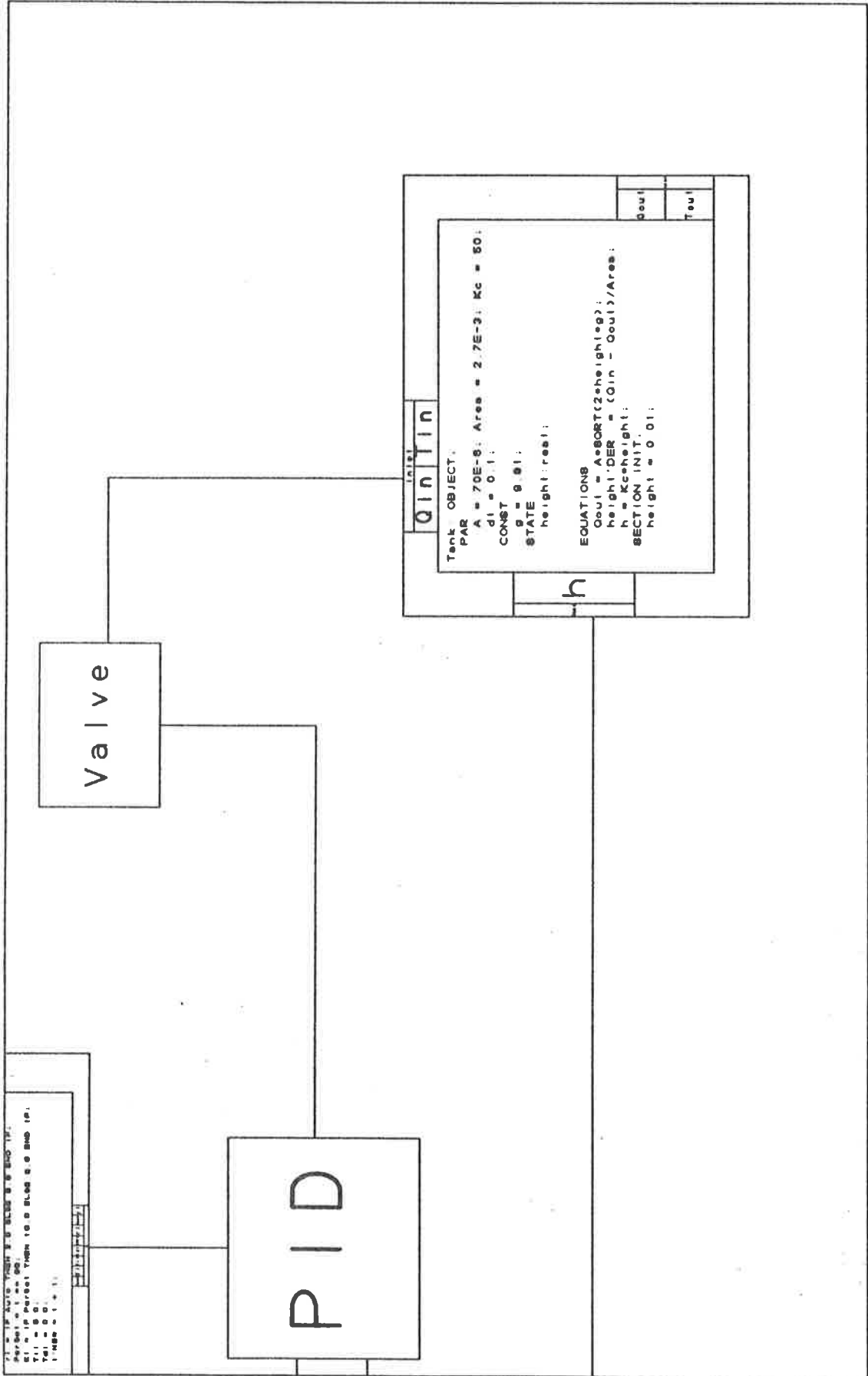
Numerisk integration av

differential-algebraiska modeller

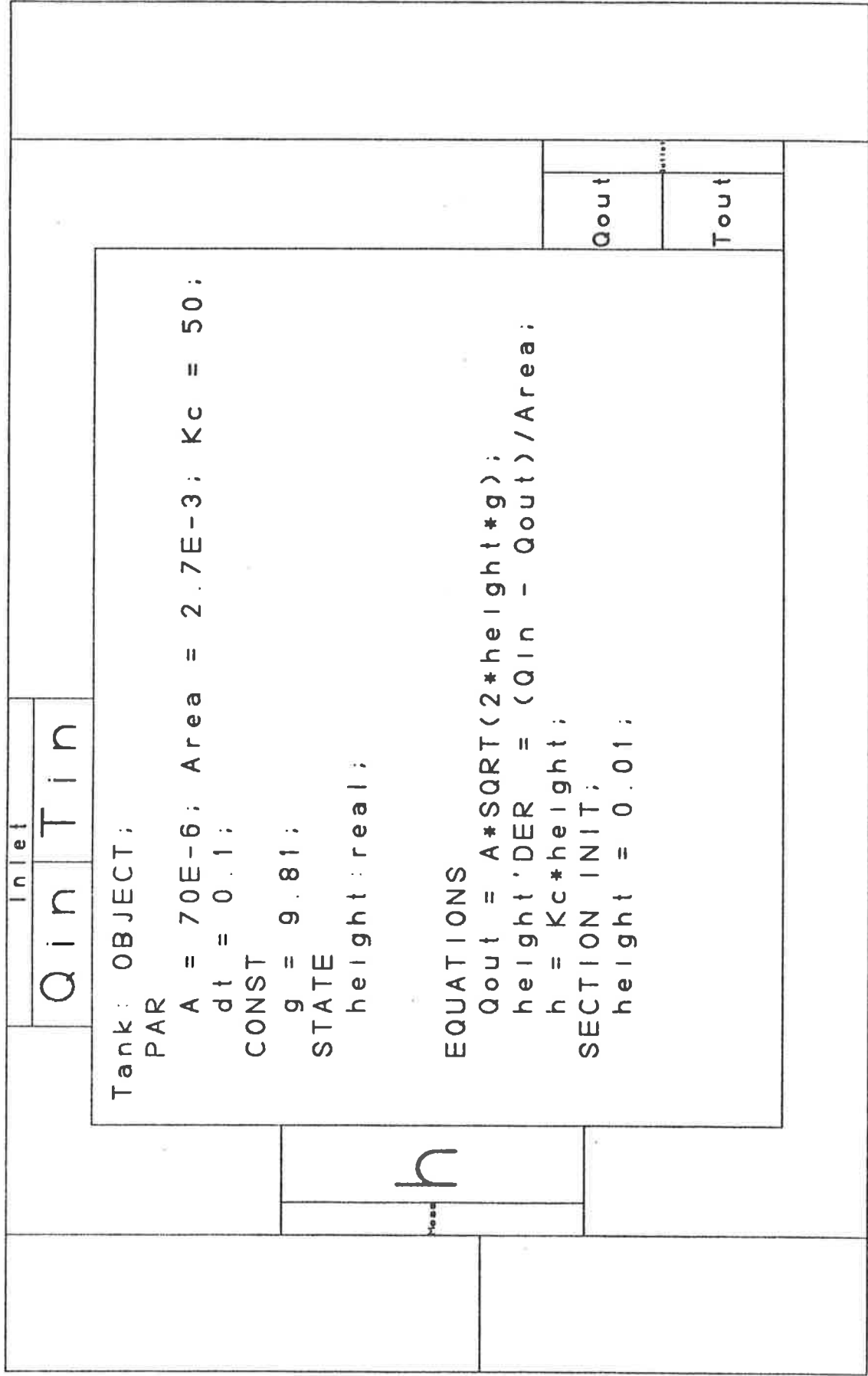
HIERARKISKA BLOCKDIAGRAM - 1



HIERARKISKA BLOCKDIAGRAM - 2



HIERARKISKA BLOCKDIAGRAM - 3



Inlet: (Qin: IN real, Tin: real)
 Outlet: (Qout: OUT real, Tout: real)

VARFÖR EKVATIONER?

Nödvändigt för 'generella' modellbibliotek

Enklare att göra modeller

Säkrare

Bättre dokumentation

IRIS 2400

**Operativsystem: UNIX System V
med Berkeley 4.2 utvidgningar
C, Fortran, Pascal**

CPU: MC68010 med flyttalsaccelerator

CPU-Minne: 2 MB

Skärm: 19 tum, RGB, 1024*768 pixel, 60 Hz

3-knapps-mus

Winchester-disk, 70 MB

Kvartstum-kasettband, 60 MB

Ethernet-interface med XNS protokoll

IRIS 2400 - Grafikhårdvara

Grafikprocessorer: 12 Geometry Engine system

> 6 Mflops (32 bit)

transformering: 65 000 koordinater/s

150 000 rastertecken/s

linjer (med textur): 3 Mpixel/s

polygoner (med textur): 44 Mpixel/s

krökta kurvor och ytor: 100 000 punkter/s

Teckenminne (fonter, texturer, markörer): 64 kB

Bildminne: 24 bildplan (1024*1024 bit)

Skärm: 19 tum, RGB, 1024*768 pixel, 60 Hz

IRIS GRAPHICS LIBRARY

Anropbart från C, Fortran och Pascal

Man arbetar i (lokala) världskoordinatsystem

Primitiver för linjer, polygoner, cirkelbågar

”Kompileerade” grafiska objekt (segment)

I/O-hantering för tangentbord och mus

Identifiering av vad markören pekar på

Primitiver för krökta linjer och ytor

Primitiver för skuggning och depth-cueing

LÄGE - NYA FORMER FÖR MMC

Personal

Sven Erik Mattsson

Dag Brück

Konsult

Hilding Elmqvist

Datorbiten

IRIS 2400

anlände 9 juli 1985

Ethernet: XNS till VAX ej levererat

kan använda Kermit till/från VAX

mindre Pascalproblem; grafikinterfacet

hårdvarufel på kassettbandstationen

Modell- och språkbiten

Generera ekvationer från ihopkopplingar

Ekvationssortering

Enklare formelbehandling

Kodgenerering

Interpretator

Integrationsrutin

Robusthet, felåterhämtning

Grafik: Interfacedesign och Implementering

Effektivitet - Portabilitet

Segment, grafiska objekt

Rita om hela bilden eller uppdatera?

Lagra information eller räkna?

Texthantering

Generera grafiska fonter

Generera raster fonter

Trend kurvor

Primärt mål:

Körbar simulatorprototyp 31 mars 1986

Hierarkiska blockdiagram

Differential-algebraiska delmodeller

Möjliga utvidgningar:

Modelltyper

Mer formelbehandling

Tidsdiskreta system

Symboler för delsystemen

Animering av processen

Visa resultat som på mätinstrument

Kunna editera axlar och grafer

An Expert System Interface
for Idpac

by
Jan Eric Larsson
and
Per Persson

Why use an Expert System?

- Systems identification is a well defined problem
- Experts are available
- Idpac is well fitted for the problem
- The problem is not trivial
- The problem is not too large
- An expert system solution seems to be possible

- Program packages

**IDPAC, SYNPAAC, MODPAC,
SIMNON, ...**

- Interactive, command driven

- Based on INTRAC

- Written in FORTRAN, ca. 1973

- IDPAC knows 40 commands

- Complex command syntax

- Brief

Very efficient

Cryptic

- E.g.

CONV outfile < infile (1 3) 4 1

RESID res < syst data 1 4 1

IDPAC commands

1. UTILITIES

CONV
DELET
EDIT
FHEAD
FORMAT
FTEST
LIST
MOVE
TURN

2. GRAPHIC OUTPUT

BODE
HCOPY
PLMAG
PLOT

3. FREQUENCY RESPONSE OPERATIONS

ASPEC
CSPEC
DFT
FROP
IDFT

4. TIME SERIES OPERATIONS

ACOF
CCOF
CONC
CUT
INSI
PICK
SCLOP
STAT
TREND
VCOP

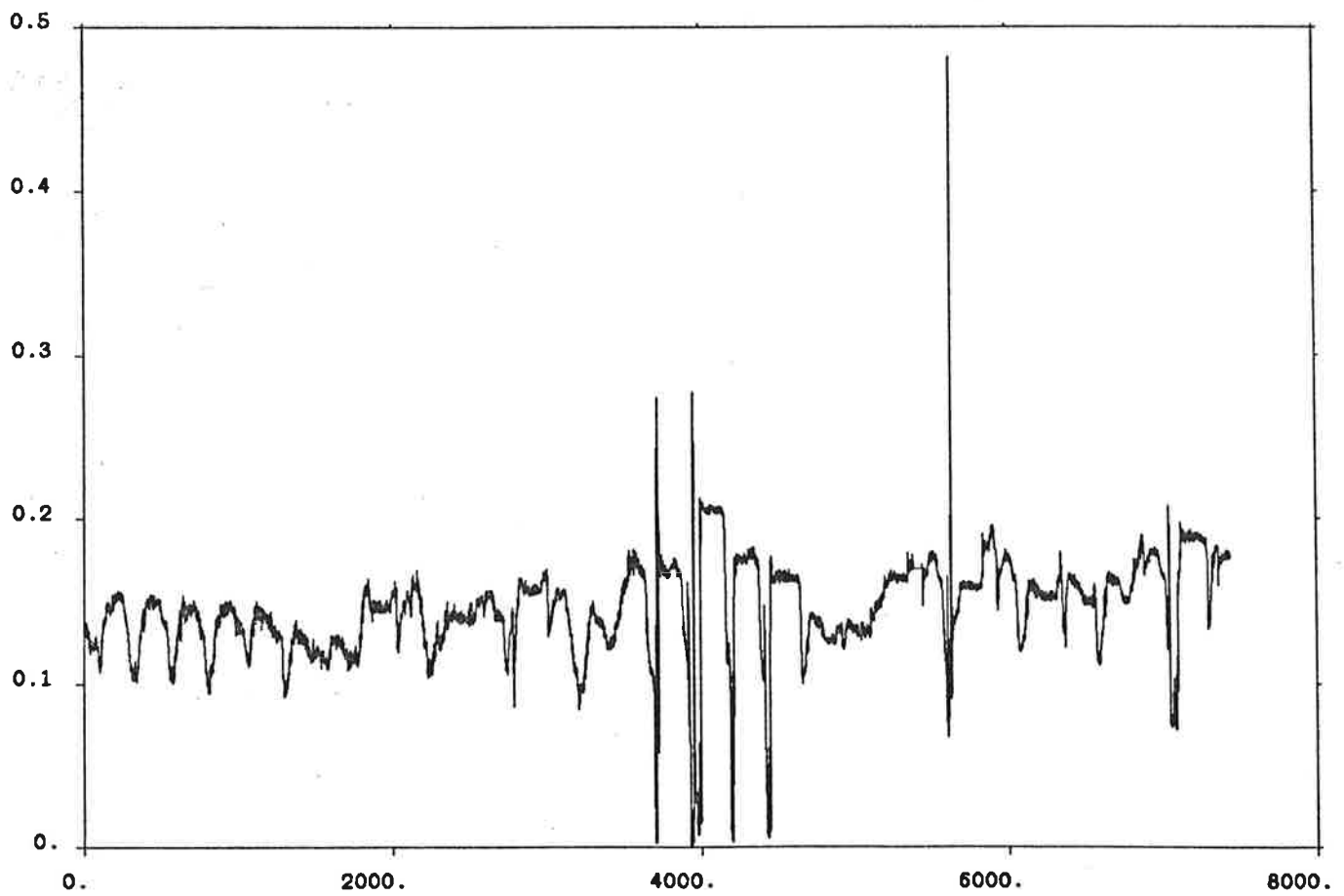
5. SIMULATION AND MODEL ANALYSIS

DETER
DSIM
FILT
RANPA
RESID
SPTRF

6. IDENTIFICATION

LS
ML
SQR
STRUC

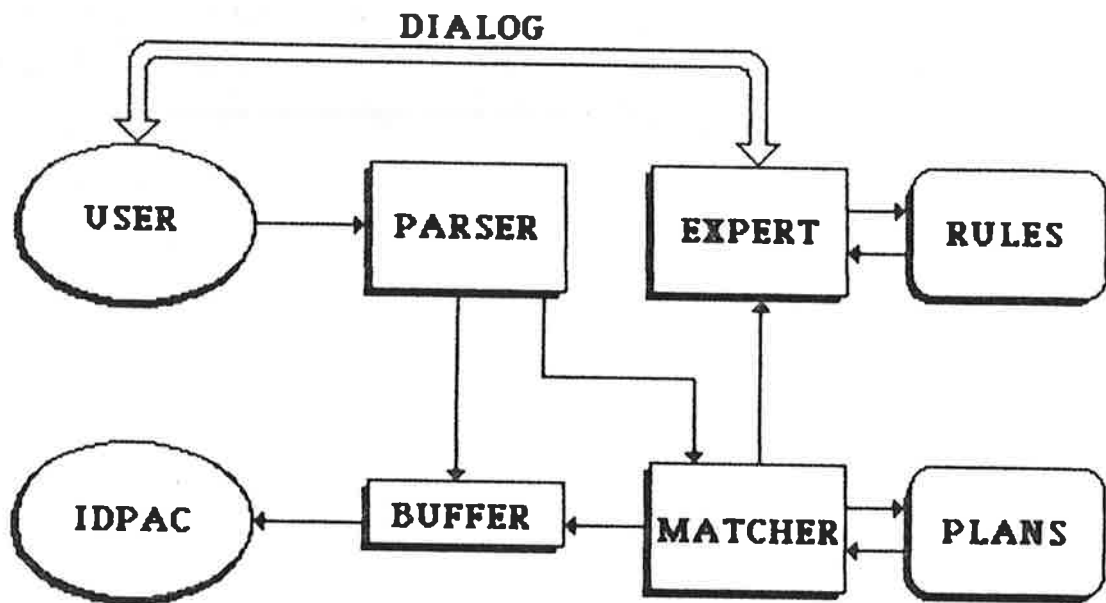
plot(8000) k1h(3)
09.03 - 09:34:33



- IDPAC's commands are hard to remember
- Help - gives only name and syntax
- Improvements of the help system
- Intelligent defaults
- Query mode

- The result strongly depends on the user's knowledge
- Demands on a good help system
- When to use a command
- How to combine commands
- A goal related help function,
help on methods,
a dynamic (= fancy) help function

- Question and answer dialog
- Command style dialog
- We want something in between
- Normal command dialog,
sometimes the help system
comes into action
- "The commmand spy"



- Franz Lisp

- VMS processes and mailboxes

(plot x)	Command sent
(ml1 ml2 ml3 ml4 corana data-analysis)	
(plmag (plot trend) acof stat)	
<p>Rule ml1-1 deduces: (write: "If there are any outliers, remove them with PLMAG")</p> <p>Rule ml1-4 deduces: (write: "If there are any trends, remove them with TREND")</p> <p>Rule corana-1 deduces: (write: "In analysis use ACOF-PLOT, ASPEC-BODE and DFT-BODE")</p>	
<pre>IDPAC> conv x < y 1 IDPAC> plot x IDPAC></pre>	

Lägesrapport september 1985

- Litet experimentsystem
 - Parser
 - Plan matcher
 - Rule system

Fortsättning

- Realistiska planer och regler
- Filhanteringsystem
- Automatisk dokumentation
- Statistik
- Query mode / Intelligent defaults
- Kopplat till doktorandkurs i identifiering

KOMBINATION AV FORMELBEHANDLING OCH NUMERIK

Motiv för formelbehandling

Strukturell information är mycket viktig

Analytiska uttryck kan ge bra insikt

Kan eliminera numeriska problem

Automatgenerering av kod

PILOTPROJEKT

Analys och syntes av flervariabla system
med polynommatrismetoder

Teori finns

Numeriskt svårt

Vad kan göras med formelbehandling?

Analys och syntes innebär

transformering till standardformer

beräkning av poler och nollställen

test av stabilitet, styr- och observerbarhet

beräkning av impuls-, steg- och rampsvar

ihopkoppling av delsystem

lösning av diofantiska ekvationer

Vilket i sin tur innebär beräkning av

$$A + B, \quad A * B, \quad \det A, \quad A^{-1}$$

Detta är i princip är möjligt att göra analytiskt

kan ta lång tid

koefficienterna kan bli komplexa

Man vill också lösa polynomekvationer

$$p(\lambda) = 0$$

vilket bara säkert går om $\deg p(\lambda) < 5$

HÖGNIVÅSPRÅK FÖR LÖSNING AV REGLERPROBLEM

Vokabulär

Data, Operatorer

Syntax

Semantik

Språket skall vara

Tillräckligt rikt för att lösa många problem

Enkelt och lätt att lära

Utvidgbart

SYSTEM REPRESENTATION ETT CENTRALT PROBLEM

Dagens CACE-system har otillräckliga datastrukturer för representation av system.

En förutsättningslös diskussion av hur system, signaler och sammankopplingar skall representeras har inte gjorts.

Koppling till formelmanipulation.

Både kvantitativa och kvalitativa egenskaper skall kunna representeras och manipuleras.

PILOTPROJEKT

Avsikt

Kartlägga existerande system som högnivåspråk för problemlösning.

Undersöka olika sätt att representera system och signaler.

Strategi

Undersöka och experimentera med olika existerande CACE-system.

För att få tillgång till andra system använder vi Simnon och Idpac som bytesobjekt.

Det ger oss en unik samling av CACE-system.

Gästforskare

Prof. W. Kreutzer, Nya Zeeland

Intressanta system är

Ctrl-C

PC-Matlab

Sandys (ASEA)

Impact (ETH, Schweiz)

Blaise (INRIA, Frankrike)

CC (Caltech, USA)

Delight (UCB, USA)

Parasol (UCB, USA)

EAGLES, M (LLNL, USA)

De engelska paketen

(UMIST, Imperial College etc.)

EXPERT CONTROL

- Feedback control with an expert system.
- Orchestrate numerical algorithms for control, identification and supervision.
- Successively increase the knowledge about the plant.

Different approaches

Using the expert system as an operator tool for alarm and performance analysis, compare Three Mile Island accident. Used on top of large, existing control systems and in open loop. Much work being done e.g. LMI's PICON, Teknowledge, Carnegie Group ...

Using the expert system in the feedback loop to improve the control at the single loop level. The expert system is used in closed loop for a relatively small control problem. Compare with adaptive control ideas. Work being done: This project, Tokyo Inst. of Tech.

Motivation

- A large part of an ordinary controller consists of a "safety jacket" or logic network needed for safe industrial operation.
- The combination of different algorithms increases the amount of logic.
- The knowledge of an experienced process operator can often be expressed in terms of rules.
- A natural division between logic and numerical algorithms is achieved.
- The interactive expert system environment provides a convenient workbench for experiments with new control structures.

Languages

Most expert systems written in Lisp or Prolog.

- The interactive programming environment.
- Symbolical languages.
- Tradition.

Numerical algorithms preferably written in a traditional language like Pascal, ADA or Fortran.

⇒ Implementation divided in two parts.

Realtime problem.

- Expert systems – slow.
- Control – time demands.

⇒ Parallel tasks.

Requirements on the expert system

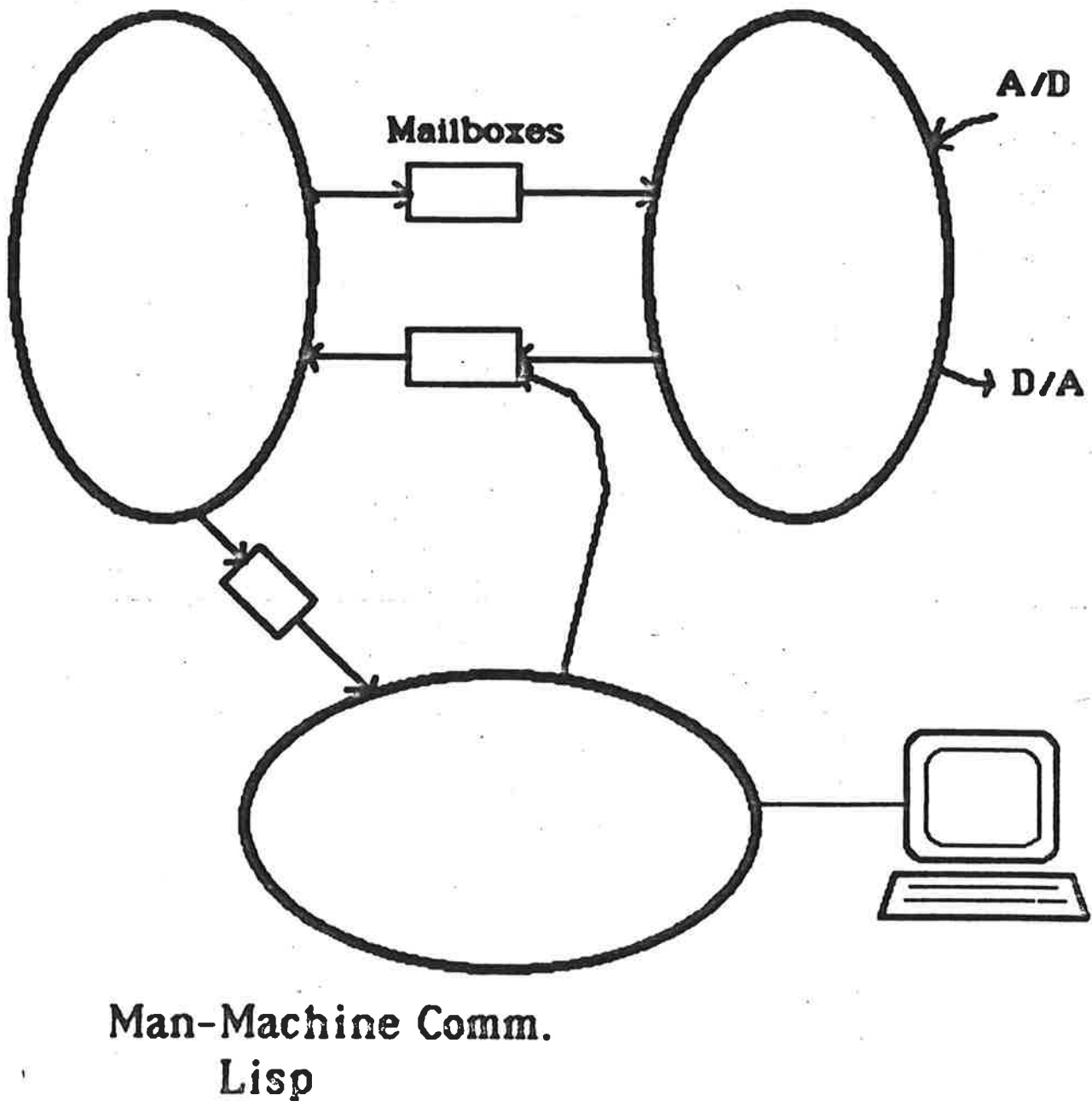
- The application consists of both planning and monitoring.
- Real time expert system. Most existing expert systems are off-line systems.
- The system should be able to take time into account in its reasoning.

Existing framework e.g. OPS5, Rosie, Loops, ART, KEE do not fulfill these requirements.

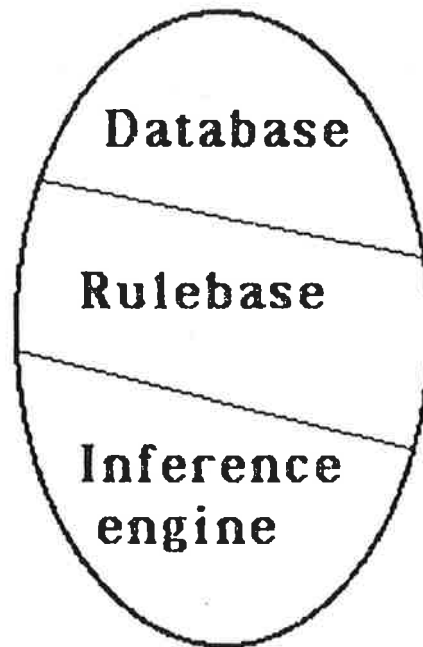
SYSTEM 1

Expert system
Lisp/OPS4

Algorithms
Pascal



OPS4



Forward chaining system

<pattern>

-->

<actions>

OPS4 is a forward chaining production system consisting of database, rulebase and inference engine.

A typical rule looks like

```
R26 ((State is upstart)
      (Goal is pidcontrol)
      (Pid parameters available)
```

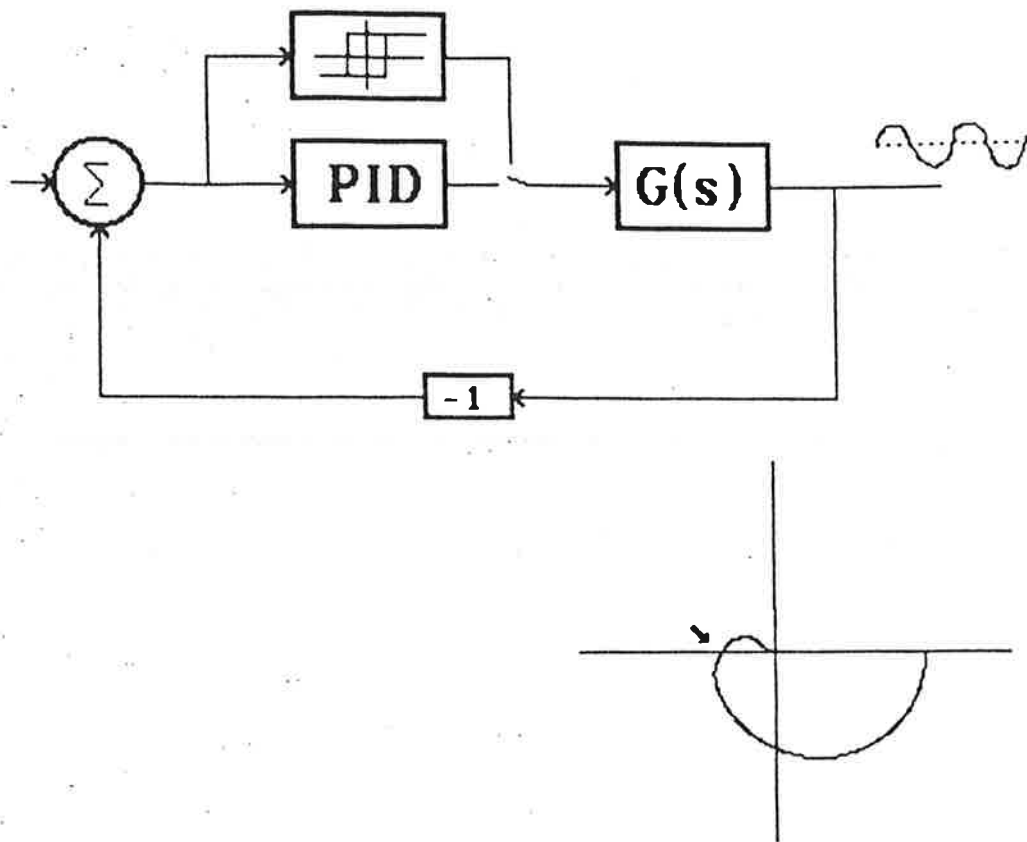
⇒

```
(DELETE (State is upstart))
(ADD (State is pidcontrol))
(ADD (State PID))
```

The messages sent to the algorithms are of the type start, stop or change parameters in an algorithm.

The message received from the algorithms are either results from an identification algorithm or an alarm from a supervision algorithm.

The user can add or delete items in the database on-line or add, edit and delete rules in the rulebase.



Used for experiments with a Ziegler-Nichols auto-tuner with gainscheduling.

Algorithms

- PID
- Relay
- Oscillation analyzer
- Noise estimator

≈ 70 rules.

SYSTEM 2

The expert system part will be specially designed for expert control. It will contain planning and take time into account in its reasoning.

It will be used for experiments with an "Intelligent" PID controller. The controller should be able to determine if

- the process can be controlled by a PI controller.
- the process can be controlled by a PID controller.
- the process needs a more complex controller.

It will also be used for initialization and monitoring of adaptive algorithms.

Possible results

- Controller implementation workbench.
- Explicit knowledgebase of practical control issues.
- Algorithm development.
- Man-machine interface.
- Combination of symbolic and numeric computations in real time.

Expertreglering

1 INLEDNING

Idén bakom detta projekt är att använda ett expertsystem som en del i en regulator. Expertsystemets uppgift är att handha logikdelen i regulatorn. Expertsystemet gör det också möjligt att inkludera heuristik och tumregler och på så sätt utnyttja den expertis som innehas av en erfaren processoperatör. Motivering för användningen av expertsystemmetodik ges i detta kapitel. I kap. 2 diskuteras vilka konsekvenser detta får för reglersystemet. En utförligare beskrivning ges i Åström och Anton: Expert Control vilken bifogas. Implementeringen av systemet diskuteras i kap. 3 och i kap. 4 ges en projektbeskrivning. Den användning som projektet kan få diskuteras i kap. 5.

Forskningen inom reglerteknik har under de sista 30 åren främst fokuserats på den teoretiska delen. Stora framsteg har skett vad det gäller identifiering, analys och design. Komplexa algoritmer för t.ex. adaptiv reglering har tagits fram bl.a. här på institutionen, (STU projekt 82-3430). En stor del i de framsteg som gjorts har den snabba utvecklingen på datorsidan. Denna utveckling har emellertid inte nått ut på industrigolvet i någon högre grad. I huvudsak är det bara i implementeringen, där digitala regulatorer ersätter de analoga. De regleralgoritmer som används är till största del av PID typ. Tittar man på en industriell PID regulator ser man att den består av långt mer än bara själva PID algoritmen. Den måste också kunna hantera saker som byte mellan olika reglermoder, integratoruppvridning, olinjära styrdon etc. Dessa faktorer är av extrem betydelse för regulatorns funktion men det finns endast begränsat stöd från teorin om hur de skall lösas. Istället löses de med heuristiska metoder och lösningen ser oftast ut som ett logiskt nätverk av if-then-else eller case satser. Detta logiska nätverk eller säkerhetsnät är ofta den dominerande delen av

regulatorkoden. De är ofta svåra att modifiera och gör koden svår genomtränglig. Dessa säkerhetsnät blir än mer betydelsefulla när en mer komplex regleralgoritm skall implementeras. Storleken på dem växer också kraftigt. Exempel är adaptiva regulatorer och multivariabla regulatorer.

Expertsystem är en gren av artificiell intelligens området som har tilldragit sig mycket uppmärksamhet på sistone (Davis och Lenat, 1982; Hayes-Roth m.fl. 1983). Expertsystemets syfte är att modellera en mänsklig experts kunskap inom ett område. Logik, heuristik och resonemang under osäkerhet är saker som kan representeras. Några exempel på tillämpningar är medicinsk diagnostik, geologiska undersökningar, datorsystemkonfigurering och VLSI design. Gemensamt för områden lämpade för expertsystem är att komplexitetsgraden är hög d.v.s. data eller kunskapsmängden är stor, och att mänsklig expertis existerar och är enig. De flesta expertsystem är avsedda för off-line bruk. Användaren svarar på frågor och matar in data i dialog med systemet. Det finns dock undantag, där expertsystemet används on-line. Dessa finns främst inom signalbehandling där expertsystem används för att ge en intelligent tolkning åt stora mängder störd och motsägelsefull information, Nii m.fl. (1982). Metoderna att representera kunskap varierar mellan olika system. De s.k. produktionssystemen bygger på regler av typen if <conditions> then <actions>. Expertsystem bygger på symbolisk informationsbehandling till skillnad från den hittills helt dominerande numeriska informationsbehandlingen. Denna dominans är emellertid på väg att minskas. Exempel på detta är Japans 5'th Generation Project, Moto-oka (1982), EG's ESPRIT och i Usa, MCC och DARPA's SCS. De flesta expertsystem är implementerade i symboliska, interaktiva programmeringsspråk som Lisp eller Prolog. De senaste åren har det kommit fram personliga arbetsstationer med speciell hårdvara för Lisp och Prolog, t.ex. Xerox, Symbolics, LMI och PSI. Dessa är direkt avsedda för AI tillämpningar med priser runt 80 000 \$.

2 EXPERTREGLERING

Att använda ett expertsystem som en komponent i ett reglersystem har många intressanta aspekter. En uppenbar fördel är den uppdelning mellan numeriska algoritmer och logik som fås. Algoritmerna kan kodas i så ren form som möjligt i ett programmeringsspråk som är väl lämpat för numeriska problem och logiken beskrivs av regler i ett expertsystem. Detta ökar läsbarheten hos programmet

och gör det lättare för personer utan programmeringserfarenhet att sätta sig in i koden. Programutveckling och underhåll underlättas också. Jämför Åström (1983b).

En annan intressant aspekt är att utnyttja ett expertsystem för att implementera stora logiska säkerhetsnät. Detta medför att implementeringen av komplexa regleralgoritmer, t.ex. adaptiva algoritmer, förenklas betydligt. Det gör det också möjligt utnyttja logik och heuristik i en högre grad än vad som görs nu. Det är ofta önskvärt att använda sig av mer än en regleralgoritm vid reglering av en process. Exempel är en algoritm för reglering under normala förhållanden och en robust algoritm under uppstart, eller olika identifieringsalgoritmer i en självinställande regulator. Genom en kombination av algoritmer, var och en väl lämpad för sitt område, kan en intressant, generell expertregulator skapas. Expertreglering innebär att låta ett regelbaserat expertsystem dirigera appliceringen av olika numeriska algoritmer på den process som skall regleras. De numeriska algoritmerna som behövs består huvudsakligen av tre olika grupper; regler-, identifierings- och övervakningsalgoritmer. Komplexiteten hos algoritmerna varierar från enkla robusta algoritmer till mera avancerade. Regleralgoritmerna sträcker sig från enkla PI och reläregulatorer till mer avancerade som polplaceringsregulatorer. Identifieringsalgoritmerna består dels av algoritmer som ger grova skattningar av okända parametrar som t.ex. estimering av kritisk period och kritisk förstärkning hos processen, och dels av algoritmer som ger en noggrannare modell som t.ex. minsta-kvadrat identifierare. Övervakningsalgoritmerna skall detektera sådana saker som överskridning av alarmnivåer, instabilitet, ringning, statistiskt fel etc. Expertsystemet skall bestämma i vilken ordning dessa algoritmer skall appliceras och parametrarna till dem. Appliceringen av en algoritm ökar kunskapsmängden om processen och påverkar på så sätt expertsystemets fortsatta beslut. Kunskapsmängden om processen byggs på detta sätt successivt upp. Denna kunskap är skiftande och osäker till sin karaktär. Den består av olika parameterskattningar och processmodeller och av de slutsatser som dragits från denna information. Expertregulatorn kan ses som en "intelligent" on-line kombination av modellbygge, identifiering, design och övervakning.

Kombinationen av flera algoritmer ställer än större krav på att logiken är korrekt. Det vanliga sättet att implementera detta är att använda en kombination av ett PC språk och ett blockorienterat språk speciellt avsett för reglertekniska

applikationer. De flesta blockorienterade reglersystem har ett användarinterface av relativt låg nivå. Användaren arbetar med logiska och analoga signaler och funktionsblock med ingångar och utgångar. Syntaxen är fix vilket gör det svårt att arbeta hierarkiskt eller med abstraktioner. Det är dessutom svårt att ändra regulatorstrukturen on-line. Istället måste processen stannas och systemet kompileras om. En expertsystem implementation skulle förenkla detta betydligt. Om ett interpreterande språk som Lisp eller Prolog används för expertsystemet kan t.ex. reglerna ändras on-line utan några begränsningar.

Ett expertsystem får stora konsekvenser även vad det gäller användarinterface. Dagens regulatorer kan inte till fullo utnyttja den a priori information som alltid finns tillgänglig om den process som skall regleras. De parametrar som operatören kan ändra på en regulator är för det mesta relaterade till regulatorns inre struktur som t.ex. förstärkning, integrations- och derivationstid i en PID regulator, och har ingen naturlig koppling till processen. Endast på ett fåtal regulatorer finns det möjlighet att specificera slutna systemets utseende t.ex. bandbredd och översläng. En erfaren process operatör har alltid a priori information av kvalitativ eller vag karaktär om den process som skall regleras. Exempel är storleksordning på dominerande tidskonstant och processförstärkning, om process har integralverkan eller ej, grad av olinjäritet etc. Ett expertsystem lämpar sig väl för att representera denna typ av information och att utnyttja den i sin beslutsfattning. Det är också möjligt att representera a priori information av en mer procedurartad karaktär. Processoperatören har kanske ett särskilt tillvägagångssätt eller särskilda tumregler var gång en större produktionsomställning skall ske. Detta kan ofta representeras som regler och regulatorn kan på så sätt fås att efterlikna operatörens beteende.

Användarinterfacet får också en ny dimension. Förutom de operatörskommandon som finns på traditionella regulatorer, som t.ex. byte mellan olika reglermoder, ändring av regulatorparametrar, finns det på en expertregulator möjlighet att ge kommandon till expertsystemet. Exempel på sådana kommandon kan vara att inspektera innehållet i kunskapsbasen, fråga vilket mål expertsystemet för närvarande arbetar på att uppfylla och vad skälet är till det och att ändra innehållet i kunskapsbasen och reglerna on-line.

3 IMPLEMENTERING

Implementeringen av en expertregulator görs lämpligen i olika programspråk. De numeriska algoritmerna bör skrivas i ett språk som lämpar sig väl för beräkningar som t.ex. Pascal eller ADA. Det skall också vara möjligt att utnyttja standardbibliotek skrivna i Fortran. Dessa språk lämpar sig emellertid inte för den symboliska informationsbehandling som krävs i ett expertsystem. De språk som i huvudsak används här är Lisp eller Prolog även om undantag finns. Dessa olika delar måste kunna kommunicera med varandra. Expertsystemet måste kunna skicka kommandon till algoritmdelen att en ny algoritm skall startas eller att parametrarna i någon algoritm skall ändras och identifierings- och övervakningsalgoritmerna måste kunna skicka resultat respektive alarm till expertsystemet. En programmeringsomgivning för experiment med expertreglering har tagits fram och beskrivs i K-E Årzén: Experiments with Expert Control vilken bifogas. I denna omgivning är algoritmdelen och expertsystem delen implementerade som separata VAX/VMS processer som kommunicerar med hjälp av VMS mailbox system. Expertsystemet är skrivet i Franz Lisp och algoritmerna i Pascal.

Ett regelbaserat expertsystem har tre huvudkomponenter:

1. Databas
2. Regelbas
3. Slutledningsmekanism

Databasen innehåller typiskt sådana saker som fakta, hypoteser, dvs. fakta som ännu ej verifierats, och mål som expertsystemet skall försöka uppfylla. I en expertregulator motsvarar detta de olika typer av statisk och dynamisk processinformation som t.ex. processmodeller, parametervärden etc., som expertsystemet arbetar med. Databasen kan internt vara organiserad på olika sätt. Regelbasen innehåller de regler som behövs för att ta fram nya fakta, verifiera hypoteser och uppfylla mål. Reglerna är ofta av typen if <condition> then <assertion>. Slutledningsmekanismen applicerar reglerna på databasen. Principerna för hur detta sker varierar mycket mellan olika system. Ofta innehåller ett expertsystem också en förklaringsfunktion dvs. det förklarar för användaren hur och varför det använder sina regler. Inom AI området har olika expertsystemramverk tagits fram. De flesta av dessa är skrivna i Lisp. Dessa

ramverk kan ses som verktyg för utveckling av expertsystem och innehåller ovan nämnda byggbitar plus hjälpmedel för regelinmatning och resultatutmatning. Dessa ramverk har stora inbördes skillnader och lämpar sig olika väl för olika tillämpningar. Några av de mest kända beskrivs här kort.

EMYCIN togs fram som ett medicinskt diagnoshjälpmedel under namnet MYCIN på Stanford University och utvecklades sedan till ett generellt system. Slutledningsmekanismen bygger på backward chaining. Detta innebär att systemet utgår från en hypotes och försöker bekräfta denna genom att hitta de regler som leder till den och därefter verifiera de fakta som dessa regler har som villkor. Denna procedur sker eventuellt rekursivt. Logiken i EMYCIN är inte tvåvärd utan till varje utsaga är kopplad en osäkerhetsfaktor. På så sätt kan osäkra och inkompleta data representeras. Det finns flera andra ramverk som bygger på EMYCIN och gemensamt för dem är att de lämpar sig väl för diagnosproblem av olika slag.

OPS5 är ett system vars slutledning helt bygger på forward chaining. Detta innebär att systemet tar alla regler vars villkor är uppfyllda, dvs. finns som fakta i databasen, och evaluerar dessa. På så sätt adderas nya fakta till databasen och det hela fortsätter tills inga fler regler är uppfyllda. Om en regel är uppfylls eller inte bestäms genom en avancerad mönster matchning av regelns villkor mot innehållet i databasen. Databasen består av objekt med attribut som reglerna kan referera till.

ROSIE har en syntax som påminner om ett naturligt språk vilket gör systemet lätthanterligt. Det har olika slutledningsstrategier och en kraftfull mönster matchning.

LOOPS är ett av de modernaste systemen. Det har utvecklats för att användas på XEROX Lisp maskiner. Det är ett komplext system som bl.a. tillåter objektorienterade expertsystem.

Ett problem med att använda något av de existerande ramverken för expertreglering är att de alla främst är avsedda för off-line tillämpningar där expertsystemet arbetar i dialog med användaren. Expertreglering kräver att expertsystemet används i reell tid och att det arbetar mot en fysikalisk process. Speciellt tidsbegreppet är svårt att uttrycka på ett naturligt sätt i existerande

system. Det går att lösa det, men inte på ett tillfredställande sätt. En implementation är gjord på institutionen med hjälp av OPS4, en föregångare till OPS5. Detta är beskrivet i Årzén (1985). Alternativet till att använda ett existerande ramverk är att skriva ett eget. Detta har fördelen att expertsystemet kan anpassas till de speciella problem som expertreglering innebär. Att skriva ett expertsystemramverk för realtidsproblem är emellertid ingen trivial uppgift, särskilt om krav finns på att det skall vara effektivt och klara av stora regelbaser. I nuläget är det för tidigt att bestämma vilket som skall göras inom projektet; att skriva ett eget ramverk eller använda ett existerande, med de begränsningar det får. Det är också osäkert vilket språk som skall användas om ett ramverk konstrueras. De flesta är skrivna i Lisp men Prolog, som direkt bygger på logik, har många fördelar. Språket bygger på sökning och mönstermatchning och backward chaining finns inbyggt. En nackdel är emellertid att det är långsamt.

Det finns vissa speciella egenskaper hos expertreglering som ställer särskilda krav på ett passande ramverk. Expertreglering kan ses som ett planeringsproblem där det gäller att finna en plan som överför systemet från ett begynnelsestillstånd till ett önskat sluttillstånd. Ett sluttillstånd eller mål kan t.ex. vara god reglering där god t.ex. definieras som att som att variansen hos reglerfelet skall var mindre än ett visst värde eller så liten den kan göras under förhåanden varande omständigheter. En plan kan bestå av de algoritmer som skall appliceras på processen. Algoritmerna kan ses som operatorer med ett specificerat syfte, villkor för att de skall kunna användas och en beskrivning av de sidoeffekter som kan orsakas av dem. Syftet med en operator är att uppfylla något av de delmål som krävs för att nå sluttillståndet. Systemet måste också ha möjlighet att revidera sin plan under exekveringen beroende på utfallet av operatorerna. Expertsystem för planeringsproblem har tagits fram bl.a för robot rörelse (Nilsson, 1980; Sacerdoti, 1977) och design av genetiska experiment, Stefik, (1981a,b). Dessa system tar emellertid inte hänsyn till den osäkerhet i tillstånd i resultat av en operator som finns i expertreglering. Sökandet efter en plan som tillfredställer ett visst slutmål är väl lämpat för backward chaining. Systemet arbetar från målet bakåt mot de operationer som behövs för att nå dit. Evaluering av en färdig plan lämpar sig väl för forward chaining. Ett system med båda dessa slutledningsmekanismer är alltså önskvärt. Någon metod för att hantera resonemang under osäkerhet bör också finnas med.

Nuvarande expertsystem är väl lämpade för tillämpningar vars problem kan karakteriseras av statiska logiska samband. Realtidsproblem är emellertid tidsberoende och därmed är sekvenser ett viktigt begrepp som måste finnas med i systemet. En AI motsvarighet till sekvenser finns i begreppet scripts, Schank och Abelson (1977).

En annan viktig del i expertreglering är att systemet skall lära sig processens karaktäristik under regleringens gång. En metod att lösa detta på är att utnyttja tabeller för lagring av information om processen. Ett enkelt exempel är lagring av regulatorparametrar för olika arbetspunkter. Systemet kan t.ex. lära sig graden av olinjäritet och tidsvariabilitet ur sådana tabeller. Möjlighet att lagra information i tabeller och dra slutsatser ur dessa är alltså också önskvärt.

4 PROJEKTBEKRIVNING

Projektet är uppdelat i två delmål. Det första är en "intelligent" självinställande PID regulator som skall kunna avgöra följande.

- * Processen kan regleras med enbart PI verkan.
- * Processen kan regleras tillfredställande med en PID regulator.
- * Processen har så komplicerad dynamik att en PID regulator inte duger.

En sådan regulator skulle kunna bygga på den Ziegler-Nichols auto-tuner som tagits fram på institutionen, Åström (1982) och Åström och Hägglund (1983). Denna metod utnyttjar ett relä för att finna den punkt där Nyquistkurvan skär negativa reella axeln. Denna punkt ger kritiska perioden och kritiska förstärkningen för processen och de utnyttjas sen i en modifierad version av Ziegler-Nichols inställningsregler för PID regulatorer. Regulatorn lagrar regulatorparametrar för olika arbetspunkter och utnyttjar dessa för parameterstyrning. Arbetet med denna är redan på gång och en prototyp finns i drift.

Det andra delen i projektet rör uppstart och övervakning av adaptiva regulatorer och då speciellt minimalvarians självinställare, Åström och Wittenmark (1973). Typiskt för adaptiva algoritmer är att de fungerar enligt lokal gradient metoden. En översikt över adaptiva regulatorer finns i Åström (1983a). De är mycket kraftfulla lokalt men kräver relativt mycket a priori information om processen.

Startas regulatorn utanför sitt arbetsområde kan den t.o.m. resultera i ett instabilt system. Varianter på Åström och Hägglunds inställningsmetoder kan användas till att plocka fram en del av den information som behövs till den adaptiva regulatorn. Prediktionshorisont och samplingsintervall är parametrar som är möjliga att plocka fram. Detta projekt finns utförligare beskrivet i Åström och Anton (1984). Några av de algoritmer som behövs är en ringnings detektor som övervakar systemet och detekterar närvaron av dåligt dämpade process nollställena, en minimalvarians övervakare som kontrollerar variansen genom att beräkna korrelationsfunktioner etc. Komplexiteten hos detta experiment kommer att bli betydligt större än hos experiment 1. Antal algoritmer som behövs blir betydligt fler och storleken på logiknätet växer betydligt.

Expertsystem angreppssättet på reglerproblem blir mer motiverat ju mer komplext problemet är. Multivariabla problem och reglerproblem med flera samverkande loopar är fall där mycket finns att vinna. En mycket intressant utveckling är att göra system som utnyttjar den ofta stora processkunskap som skickliga operatörer utnyttjar för att styra stora anläggningar. Trots detta kommer projektet huvudsakligen att inrikta sig på SISO fallet. Detta är ingen allvarlig begränsning eftersom de problemställningar som måste lösas när ett expertsystem används för reglering i reell tid är de samma. Strukturen på den expertsystemlösning som fås bör också gå att utvidga till flera loopar. Målsättningen för projektet är att konstruera ett faktiskt expertreglersystem och inte bara en papperskonstruktion. Genom att börja med ett enklare problem ökas förutsättningarna för detta betydligt.

Den miljö som tas fram kan ses som en arbetsbänk för utveckling med komplexa regulatorer. Det skall snabbt vara möjligt att införa nya algoritmer och att ändra och testa det logiknät som byggs upp i reglerna. Detta är möjligt genom den snabba interaktion som fås med ett expertsystem och genom uppdelningen i logik och algoritmer. När ett fungerande system väl är utvecklat är det en öppen fråga hur det skall implementeras. Om det inte är nödvändigt att bibehålla den snabba interaktionen kan implementeringen mycket väl ske med traditionell hård- och mjukvara.

5 ANVÄNDNING

De resultat som detta projektet leder till bör vara intressanta dels för företag som producerar reglerutrustning och dels för processindustrin som helhet. Det är tre olika aspekter av projektet som har speciellt intresse. Det första är angreppssättet som används med ett expertsystem för symboliskt resonemang och beslutsfattande och numeriska algoritmer för identifiering och reglering. Detta ligger väl i linje med det ökade intresse som finns för symbolisk informationsbehandling och expertsystem och bör vara intressant även utanför rena reglerteknik kretsar. Expertsystem som arbetar i reell tid är en annan aspekt av detta projekt som likaledes borde vara av allmännt intresse.

Den andra aspekten är den kunskapsbas som byggs upp under konstruktionen av expertsystemet. En allmän iakttagelse vid arbete med expertsystem är att man får verkligen klart för sig vad det är man egentligen vet om tillämpningsområdet. Den kunskap som finns måste struktureras och formaliseras och detta i sig har ett egenvärde. Den kunskap som tas fram om uppstart och övervakning av adaptiva regulatorer är av stort potentiellt intresse för svensk industri.

Angreppssättet med ett expertsystem i kombination med algoritmer ställer också nya teoretiska krav. Speciellt gäller detta den utveckling av numeriska algoritmer som måste ske framförallt för processövervakning. Instabilitet, ringning och om systemet befinner sig i stationaritet eller ej är några saker som det är önskvärt att kunna detektera. Förutom dessa tre saker så kan främst den betydelse expertsystem får för operatörsinterfacet att vara av intresse och då speciellt möjligheten för systemet att efterlikna en erfaren processoperatörs beteende.

Projektet bör också ha ett betydande nyhetsvärde. Användning av ett regelbaserat expertsystem för reglering är vad vi vet ny, även om en del angränsande projekt finns. Foxboro har en självinställande PID regulator vars inställningsregler bygger på mönstermatchning, Kraus och Myron (1984). Detta koncept kallar de för Expert Adaptive Controller Tuning. Inom CAD system för reglerteknik har expertsystem använts bl.a. för identifiering, Larsson (1984), och linjär kvadratisk design, Trankle och Markosian (1985). Det bedrivs också mycket arbete vad det gäller övervakning av stora reglersystem. Detta arbete inspirerades av Harrisburg olyckan och en typisk tillämpning är tolkning av alarmsignaler för att utreda det ursprungliga felet och ge förslag på åtgärd. Detta arbete skiljer sig

från vårt projekt på två sätt. För det första används expertsystemet endast för övervakning, inte för reglering och identifiering, och för det andra används det ovanpå ett existerande, traditionellt reglersystem. Lisp Machine Inc. är en tillverkare av Lispmaskiner som driver projekt inom den här typen av processreglering.

Institutionen har kontakter med flera olika industrier och högskolor inom områden som gränsar till projektet. Inom Sverige kan nämnas Inst. för Tillämpad Elektronik på Chalmers som sysslar med Knowledge Based Signal Processing främst för medicinska tillämpningar, Alfa Laval, NAF Controls, SCA, Processdata, ASEA, First Control Systems och Erik Sandevalls AI grupp i Linköping. Utomlands har institutionen kontakter med Systems Control Technology och Reasoning Systems Inc. båda AI företag i Palo Alto, Kalifornien och MIT's och University of Maryland's AI grupper.

Vad det gäller ren AI forskning är Sverige väl representerat främst genom grupperna i Linköping, Uppsala, KTH och Chalmers. Vår målsättning är att tillämpa AI resultat inom reglerteknik.

Resultaten av projektet kommer att dokumenteras i rapporter och presenteras på konferenser och i tekniska tidskrifter.

6 REFERENSER

- Davis, R. och D. Lenat (1982): Knowledge-Based Systems in Artificial Intelligence. McGraw-Hill, New York.
- Hayes-Roth, F., D. Waterman och D. Lenat (1983): Building expert systems. Addison-Wesley, Reading, Mass.
- Kraus, T.W. och T.J. Myron (1984): Self-Tuning PID Controller Uses Pattern Recognition Approach. Control Engineering. June 1984, pg 106-111.
- Larsson, J.E. (1984): An expert system interface for IDPAC. Rapport TFRT-5310, Inst. för Reglerteknik, Lunds Tekniska Högskola.
- Moto-Oka, T. (1982): Fifth Generation Computer Systems. North-Holland.
- Nii, H., E. Feigenbaum, J.J. Anton och A.J. Rockmoore (1982): Signal-to-Symbol transformation: HASP/SIAP case study. The AI Mag. Vol. 3 No. 2 1982.
- Nilsson, N. (1980): Principles of Artificial Intelligence. Toga Press Palo Alto, Ca.

- Sacerdoti, E. (1977): *The structure of Plans and Behavior*. Elsevier North Holland, New York.
- Schank, R.C. och R.P. Abelson (1977): *Scripts, plans, goals and understanding*. Lawrence Erlbaum Associates, Hillsdale, N.J.
- Stefik, M. (1981a): Planning with constraints (MOLGEN: Part 1). *Artificial Intelligence* 16, No. 2, 111-139.
- Stefik, M. (1981b): Planning and meta-planning (MOLGEN: Part 2). *Artificial Intelligence* 16, No. 2, 141-169.
- Trankle, T.L. och L. Z. Markosian (1985): *Control Design Using an Expert Planning System*, to appear at IEEE CACSD '85, Santa Barbara, Ca 1985.
- Årzén, K-E. (1985): *Experiments with Expert Control*. To appear.
- Åström, K.J. (1982): *A Ziegler-Nichols Auto-tuner Rapport TFRT-3167*. Inst. för Reglerteknik, Lunds Tekniska Högskola.
- Åström, K.J. (1983a): *Theory and applications of adaptive control*. *Automatica* 19, 471-486.
- Åström, K.J. (1983b): *Implementation of an auto-tuner using expert system ideas*. Rapport TFRT 7256, Inst. för Reglerteknik, Lunds Tekniska Högskola.
- Åström, K.J. och J.J. Anton (1984): *Expert Control*, Proc. IFAC 9'th World Congress, Budapest, Ungern 1984.
- Åström, K.J. och T. Hägglund (1983): *Automatic tuning of simple regulators*. Preprints IFAC workshop on Adaptive systems in control and signal processing, San Francisco, CA.
- Åström, K.J. och B. Wittenmark (1973): *On self-tuning regulators*. *Automatica* 20, 645-651.

IEEE**Symposier**

CACSD'83, Boston, 28 - 30 sep 1983

CACSD'85, Santa Barbara, 13 - 15 mars 1985

CACSD'86, Washington DC, oktober 1986

Technical Committee on CACSD

Charles Herget, LLNL, USA

Arbetsgrupper

Algorithms

AI

Benchmarks

Design and Applications

Simulation

Workstations

IFAC

Symposier

Zurich, 1979

West Lafayette, 1982

CADCE'85, Lyngby, 31/7 - 2/8 1985

Kommer troligen att bilda en arbetsgrupp för standardisering av CACE-mjukvara.

KATALOGISERING AV CACE-MJUKVARA

ELCS, ETH, Zurich

SYCOT, WGS, Benelux

prof. Frederick, RPI, USA

Dr. Herget, LLNL, USA

PROJEKT

England

SERC, 2.8M GBP, 5 år

Cambridge, Maciejowski, datatyper för CACE

UMIST, Munro, stand. användarsnitt

Univ. of Salford, Gray, grafik, MMI

Kingston Polytechnic, Denham, numerik

Frankrike

INRIA, Delebecque och Steer, Blaise

INRIA, Gomez, AI, formelbeh., olinj. filter
tillsammans med Maryland, USA

Schweiz

ETH, Rinvall, IMPACT

USA

LLNL, Herget, Eagles och M

**Univ. of Tennessee, Birdwell, AI för design
ingår i DOE:s CASCADE-projekt**

UCB, Polak, DELIGHT

DELIGHT projekt även på Univ. of Maryland

**Univ. of Maryland, Blankenship,
formelbeh. olinj. filter**

tillsammans med INRIA, Frankrike

RPI, Frederick, AI för design

tillsammans med GE (Taylor)

och West Point (James)

UCSB, Laub, numerik

SOME INTERESTING INTERNATIONAL CACE SOFTWARE EFFORTS

Sven Erik Mattsson and Karl Johan Åström

**Department of Automatic Control
Lund Institute of Technology
Lund, Sweden**

1985-08-30

1. INTRODUCTION

As stated in the general program for the Swedish, STU supported CACE project, an important part of the project is to set up international contacts. The purpose of this report is to identify and describe some foreign CACE software efforts, which are of special interest for the Swedish CACE project. These projects may give us inspiration and possibilities to exchange ideas, experiences and software. These contacts also help us avoiding redoing things and mistakes already done by others. Special stress has been laid on the computer scientific parts like man-machine interaction, numerics, high level language design etc., whereas more traditional automatic control like design of algorithms almost has been neglected.

Many packages for computer aided modelling, analysis and design of control systems have been developed and more are to come. Every university with education and research in automatic control seems to have developed software in this area. The software is often the outcome of MS thesis projects. The quality of the packages varies. As stated above it is not our purpose to give a list of control software. This task has been undertaken by other groups. The Institute of Automatic Control at ETH, Zürich collects information on existing CACSD software efforts around the world and plans to issue this information regularly (probably quarterly). The list is called ELCS (the ETH List of Control Software) and the first number was issued in June 1985. It contains 40 software descriptions. This effort is coordinated by similar efforts in Benelux (the WGS Newsletter, see Benelux below) and in the USA (prof. Frederick, RPI and Dr. Herget, Lawrence Livermore National Laboratory).

International organizations like IEEE and IFAC play also an important role. They organize conferences, symposia, workshops etc. which make it possible to find and establish new contacts as well as in an easy way maintain old ones. The IEEE Control Systems Society (CSS) has arranged two symposia on computer-aided control system design: CACSD'83 in Boston, September 28 - 30, 1983 and CACSD'85 in Santa Barbara, March 13 - 15, 1985. The third symposium is planned to be held in Washington DC in October 1986. In June 1981 IEEE CSS formed the Technical Committee on Computer-Aided Control Systems Design. It is chaired by Dr. Charles Herget. In December 1984 IEEE CSS also formed six working groups: Algorithms, AI, Benchmarks, Design and Applications, Simulation and Workstations (IEEE CSS, 1985). The working group for benchmark problem

has started with preparing problems for evaluating the analytic capabilities of packages (Frederick, 1985, IEEE CSS, 1985) and will continue with procedures for evaluating the computer-aided engineering capabilities.

IFAC has organized three symposia on computer aided design in control and engineering systems. The first one was held in Zürich, Switzerland, 1979 and the second one in West Lafayette, 1982. The third one CADCE '85 was held in Lyngby, Denmark, July 31 - August 2, 1985. One session there was devoted to standardization of software for CACSD. One outcome of the discussion will probably be the forming of a working group.

We will now start considering a number of CACE projects in different countries and our relation to them. A list of visits to the department and visits of the staff to companies and other universities as well as participation in conferences, symposia, workshops, courses are given in Appendix A.

2. PROJECTS

United Kingdom

Current work in computer aided design of control systems in the United Kingdom is surveyed by Jacobs (1985).

Over the past decade UK control engineering developments have been concentrated on research into the fundamental theory of multivariable systems and the development of new control design techniques for such systems. This work has resulted in such design techniques as the complex variable design methods (MacFarlane, Cambridge), the inverse Nyquist array design method (Rosenbrock, UMIST), optimization techniques (Mayne, Imperial College). Many of these developments have been embodied in CAD packages.

Unfortunately, this work does not fulfil the desires and demands from aerospace, marine and production industries. The design methods are not suitable in their present form for systems with widely varying parameters or significant nonlinearities. Furthermore, the software has been produced in a fragmented and an unstandardized manner. To remedy these weak points the Science and

Engineering Research Council (SERC) promotes a programme in computing and design techniques for control engineering. It was started in 1984 and the budget over the first 5 year period is 2.8M GBP. Groups participate from amongst others the University of Manchester (UMIST), Cambridge University, National Physics Laboratory and Imperial College.

Their idea is to produce a tool kit of general purpose modules and to avoid to build large software packages. To be able to do that without a lot of interfaces a common data model must be used. Maciejowski (1984, 1985a,b) at Cambridge has got a grant to develop a data model which could be accepted as a standard by the control community. At UMIST Goodfellow and Munro (1985a,b,c) have got a grant to consider the integration environment for CACE. Their attention is now focused on the man-machine interface. A software tool for input translation named INTEGRA (INput of TExt and GRAPhics) has been designed. It is a general compiler-compiler (maybe too general) that can generate a translator for a stream of mixed graphical and textural input. The acceptable inputs are defined by graphs with attached action information. These two projects are the only ones granted by SERC. The continuation is open. It was planned to use the DELIGHT environment (Nye, 1983) as a starting point for the software development. Unfortunately, it was not possible for them to get an agreement with prof. Polak, University of California, Berkeley.

Another interesting project in the United Kingdom is the development of a general purpose graphics based man-machine interface at the University of Salford (King and Gray, 1985). Work on a Control Engineering Subroutine Library has been centered at Kingston Polytechnic.

We have also established relations with a group at CEGB who is doing very elaborate simulation of power system components. This group also has a wide experience of integrating large systems of stiff equations.

The contacts with the research groups in the United Kingdom are well-established. SERC has made a formal request to STU for collaboration.

Benelux

In March 1983 a working group named "Working Group on Software (WGS)" with members from the academic society and industry in the Netherlands and Belgium was formed (WGS, 1985a).

Its first aim is the production and maintenance of a basic software library for CACSD. They name it SYCOT (SYstem and COntrol Tools). Implementation and documentation standards have been established. Now contacts are made with persons and institutions all over the world to start cooperation and to get routines and information about existing packages. The result of this work will be an inventory of existing, available software which possibly can be used as a basis for SYCOT. The first list has already been published (WGS, 1985b) and a first release of SYCOT is expected in 1985. Then the gaps in SYCOT will be filled in by own or external improvements of existing routines or by development of new routines. The long term aim is an interactive, user-friendly and flexible design software.

France

In France at INRIA Delebecque and Steer (1985) develop an interactive system for control engineering called Blaise. The starting material has been MATLAB. Blaise has both real and complex matrices. Data structures for strings and polynomial matrices have been introduced. Rational function matrices are not supported due to the lack of good numerical algorithms. A linear system can be represented as a triple of matrices or as a pair of polynomial matrices. Function macros make it possible to define parameter and time dependent matrices and nonlinear systems. Facilities for analysis and design in the time domain as well as in the frequency domain are available. It is also possible to simulate ordinary differential equations and solve optimization problems. Macros can be used to interactively define new algorithms. An interesting feature of Blaise is that it has an interface to MACSYMA to allow symbolic formula manipulation. They plan to implement an automatic generation of Blaise macros from MACSYMA expressions. The desire to have a portable system and the fact that most of the available numerical packages (Linpack and Eispack) are written in Fortran have imposed them to choose Fortran for the implementation.

We have a version of Blaise on our VAX and we have common interests in graphics, symbolic calculation and design of high level languages.

We have also contact with another project at INRIA. It is a joint project with the Electrical Engineering Department, University of Maryland, USA. They develop a prototype expert system for the treatment of stochastic control and nonlinear filtering (Gomez et al., 1984). It accepts user input in natural language or symbolic form and then carries out the basic analysis of the problem in symbolic form and outputs Fortran code for the numerical reduction of the problem. The system itself is written in Macsyma and Lisp.

Switzerland

In Switzerland at the Institute of Automatic Control at ETH, Zürich there is a project to develop a CAD-program for control systems. The program is named IMPACT (Interactive Program for Automatic Control Theory) (Rimvall and Cellier, 1984). An interesting fact about IMPACT is that it is implemented in Ada. However, Fortran packages like EISPACK and LINPACK will be used until appropriate packages are available in Ada.

IMPACT is meant to be a general purpose CACSD package. It is inspired of MATLAB. However, unlike MATLAB it also supports other data structures than complex matrices like polynomials and rational functions and matrices thereof. There is also a data structure for aggregating the A, B, C and D matrices of a linear state space description. Function macros make it possible define parameter and time dependent matrices and nonlinear systems. Facilities for analysis and design in the time domain as well as in the frequency domain are available. Macros can be used to interactively define new algorithms. The standard command language is syntactically similar to Ada. A facility of concurrent sessions using window techniques are suggested to improve the user interface (Rimvall and Bomholt, 1985). The user can then treat different aspects of the problem in different windows. He can also easily create scratch pads without cluttering the rest of the calculations. Macros are normally run in command mode, but a facility for running macros in question/answer mode is suggested. The system then outputs information supplied by the macro and prompts for the parameters. A first subset of IMPACT will be available during 1985 at low cost.

USA

The Research and Development Center, General Electric Company, Schenectady, NY has played an important role. A workshop held in Schenectady and Troy, NY sponsored by GE and RPI initiated a number of IEEE CSS activities as described above.

GE has developed what they call the Federated Computer-Aided Control Design System (Spang, 1984). It consists of several separately developed subsystems tied together by a unified database. The Federated System includes Idpac for modelling, CLADP (Cambridge Linear Analysis and Design Program) and SSDP (GE State Space Design Package) for analysis and design and Simnon for nonlinear simulation. Matlab has also been added to provide flexible means of handling matrix equations. The Federated System also includes programs to go from one subsystem to another. To simplify the use of the Federated System there is a joint project with the United States Military Academy, West Point and RPI to develop an expert system (Taylor and Frederick, 1984 and James, Taylor and Frederick, 1985).

The general purpose, interactive, optimization-based CAD-system DELIGHT (Polak et al., 1982 and Nye 1982, 1983), which has been developed by professor Polak's group at University of California, Berkeley is another source of inspiration. DELIGHT has some interesting graphical representations for gradients, constraints and criteria while solving a multi criteria optimization problem.

DELIGHT is actually a successor to Intrac. We provided prof. Polak with Intrac in source code and programming assistance. Our programmer Tommy Essebo was invited to Berkeley for two three month periods to get the DELIGHT project started. We have also been kept informed about the progress with DELIGHT continuously and we have access to the software.

The Lawrence Livermore National Laboratory has also been involved in the DELIGHT project and their experiences are used in the EAGLES project (Lawver, 1985) and the M language project (Gavel, 1985). EAGLES is the Environment for Advanced Graphical (LLNL) Engineering Systems on workstation computers. It is meant to support a user friendly interface to engineering tools with an emphasis on graphical interfaces. The EAGLES interactive environment interprets M language constructs. The M language handles general expressions involving

complex variables, polynomials, transfer functions and matrices of these types. The part of the M language dealing with matrix manipulation is quite similar to Matlab. String manipulation is also supported. The language includes also ordinary if-then-else, for, and while statements. Functions may be matrix-valued. M also has Simmon-like constructs for defining ordinary differential and difference equations for dynamic simulation. The release of version 1.0 is scheduled to October 1986 and it will be possible to get a system for a nominal fee, since it is public domain software.

We will of course continue to have contact with Dr. Herget and his research group at Lawrence Livermore National Laboratory. They have most types of workstations and we are invited to test different workstations, and learn about their experiences.

We have been in very close contact with the Systems Research Center at University of Maryland from its conception. We are also listed as one of their international collaboration partners in their application to NSF (Baras, 1984). We strongly believe that a liason with University of Maryland can be extremely valuable.

In USA the Department of Energy (DOE) funds a large project to develop CACSD software named CASCADE (Computer-Aided Systems and Control Analysis and Design Environment).

The first phase of this project has been the specification of the requirements (Birdwell, 1984). Several organizations participated. The Oak Ridge National Laboratory acted as the principal contractor and provided expertise in energy systems, numerical analysis and optimization theory. The University of Tennessee provided program support services and expertise in digital systems, computer science and expert systems. The Lawrence Livermore Laboratory provided expertise in computer graphics, systems theory and CAD. The Charles Stark Draper Laboratory provided expertise in CAD and large systems. Professor Alan Laub of the University of California, Santa Barbara provided expertise in numerical analysis. Professor Michael Athans, MIT provided expertise in estimation and Control Theory. Dr. Gunter Stein, Honeywell and MIT and Dr. Austin Spang, General Electric and MIT have also been advisors.

Most of the recommendations are generally agreed upon like the need of reliable

numerical software, interactive input and output graphics support. However, there are also recommendations which are open to discussion. It is recommended to use the Unix environment as a host for CASCADE and to use C and Fortran for implementation. It is recommended that the design procedures should be based on optimization rather than depend on the user's intuition to choose. It is stated: "The converse to this recommendation is that the CASCADE cannot do all the design work; the engineer must remain in control in an interactive design approach". From our point of view this is an empty statement, because this assumes a fixed criterion and fixed constraints. This is usually not the case, but the criterion is just another way to parameterize the design problem and the designer changes the parameters of the criterion until he is satisfied. This is also the recommended way of using the interactive optimization-based system DELIGHT, where the user can trade off the constraints against the performance objectives.

The second phase, which is under way, will produce a prototype CASCADE for linear systems analysis and design. The time for this phase is 1.5 years. Five features are emphasized: expert system interface (Birdwell et al., 1985a,b), standardized device-independent graphics, reliable and robust numerical software, distributed computing, large scale system modeling and design. The objective of the third phase is to design and implement a production grade CASCADE. This effort is expected to require a 25 to 30 man-year effort and four years.

LMI Process Systems, Cambridge, Massachusetts has an very interesting project for development of a real-time expert system for process control called PICON (More et al. 1985). Systems Control Technology and Reasonings Systems in Palo Alto have a joint project for development of an expert system for process control (Trankle and Markosian 1985a,b). They develop one system called RIFCS, Reconfigurable Flight Control System. Reasoning Systems is also working on a very advanced concept for a knowledge-based environment. It is based on ideas developed by Cordell Green at the Kestrel Institute (Smith et al, 1985, Rockmore, 1985). The idea is based on giving formal executable specifications. The code is then generated by knowledge based code generation system. There is a variety of tools for validation. The code generation can also be made in many different way depending on the purpose, rapid prototyping, memory efficient code etc. We have good personal relation with the group at Reasoning Systems.

Dr. Gary Copec, Fairchild Laboratory for Artificial Intelligence, Palo Alto, California has developed some interesting system for signal processing. First,

there is the integrated signal processing system ISP (Kopec, 1984a) with the signal representation language SRL (Kopec, 1983). It provides an unified environment for signal data processing and development of signal processing algorithms. ISP provides a signal stack, display windows, signal pictures which are used to create and view selected signals from the universe defined by SRL. Second, there is the signal database system SDB (Kopec, 1984b), which is a set of facilities for managing a corpus of signal data and the results of operations performed on them. These two systems are interesting because they may give us ideas on how to use graphics, how to represent data and how to design databases, query systems and high level languages. These systems are written in Zetalisp and run on Symbolics 3600. It would probably be possible to get the systems for a nominal fee for non-commercial use. They have been released for use at a number of universities in the USA. Dr. Kopec would probably also be willing to visit us and to give some seminars. Then it would of course be nice to have a Symbolics 3600 available.

Professor Eldeib, George Mason University, Virginia is a new acquaintance. His interests are AI, decision and automatic control. He has started research in expert systems for computer aided design of control system (Eldeib, 1985).

We also have well-established collaboration with prof. Alan Laub of the University of California, Santa Barbara. He works with numerical linear algebra and Riccati equation solvers.

3. CONCLUSIONS

Several trends can be recognized. First, there are efforts to make extensions to MATLAB by introducing new data types besides complex matrices. Examples of such efforts are Blaise (Delebecque and Steer, 1985), IMPACT (Rimvall and Cellier, 1984) and the M language (Gavel, 1985). The need to consider standardization to make it possible to switch between different software is also recognized by international organizations like IEEE and IFAC. There is also a desire to view software as tools and have possibilities to aggregate these tools in an open and extensible toolbox and to avoid as closed packages. This aspect plays an important role in the programme supported by SERC in the United Kingdom. Second, there are efforts to introduce expert knowledge in CACE systems. Most of these efforts are currently done in USA; General Electric, the United States Military Academy, West Point and RPI (Taylor and Frederick, 1984 and James, Taylor and Frederick, 1985), the CASCADE project funded by the DOE (Birdwell et al., 1985a,b), LMI Process Systems, Cambridge, Massachusetts (More et al.1985), Systems Control Technology and Reasonings Systems (Trankle and Markosian 1985a,b) and George Mason University, Virginia (Eldeib, 1985). The use of expert system techniques for CACE is still in its infancy and our project to develop an expert system interface for Idpac is certainly in the front line. Third, everybody talks about graphics, but not too much new is done within the control community. People who we have talked with and who attended CACSD'83 in Boston remember still Hilding Elmqvist's presentation of LICS and they encourage us to continue this work. They think that our ideas to design a simulator based on this type of graphics are exciting. Fourth, numerics is as always an important area and symbolic manipulation has been introduced in CACE, for example Gomez et al. (1984).

The international exchange program is running well. Contacts are established with most of the important groups. We have benefitted from long-established contacts and the reputation of the department gained from older projects. The outcomes of the older projects have been valuable goods. Most people we have had contacts with has found our new CACE project to be both extensive and exciting and will therefore cooperate with us.

References

- Baras J.S. (ed) (1984): Proposal for the establishment of a Systems Research Center submitted to National Science Foundation. College of Engineering, University of Maryland, Maryland, USA.
- Birdwell, J.D. (1984): Issues in the Design of a Computer-Aided Systems and Control Analysis and Design Environment (CASCADE). ORNL/TM-9038. Oak Ridge National Laboratory, Oak Ridge, Tennessee, USA.
- Birdwell, J.D. et al. (1985a): CASCADE: Experiments in the Development of Knowledge-Based Computer-Aided Systems and Control Analysis and Design Environments. Proceedings of the 2nd IEEE Control Systems Society Symposium on Computer-Aided Control System Design (CACSD), Santa Barbara, California, March, 13-15, 1985.
- Birdwell, J.D., Cockett, J.R.B., Heller, R., Rochelle, R.W. Laub A.J., Athans, M. (1985b): Expert Systems Techniques in a Computer Based Control System and Design Environment. CADCE '85, 3rd IFAC/IFIP Symposium on Computer Aided Design in Control and Engineering Systems, Lyngby, Denmark, July 31 - August 2, 1985, pp. 1 - 8.
- Delebecque, F. and Steer, S. (1985): The Interactive System Blaise for Control Engineering. CADCE '85, 3rd IFAC/IFIP Symposium on Computer Aided Design in Control and Engineering Systems, Lyngby, Denmark, July 31 - August 2, 1985, pp. 44 - 46.
- DOD (1983): Reference Manual for the Ada Programming Language. ANSI/MIL-STD-1815A-1983, United States Department of Defence, Washington, DC.
- Dongarra, J.J., Moler, C.B., Bunch, J.R., and Stewart, G.W. (1979): LINPACK - Users' guide. SIAM, Philadelphia.
- Eldeib, H.K. (1985): Outline of a New Approach to Computer Aided Design of Control Systems. CADCE '85, 3rd IFAC/IFIP Symposium on Computer Aided Design in Control and Engineering Systems, Lyngby, Denmark, July 31 - August 2, 1985, pp. 1 - 8.
- Frederick, D.K. and Rimer, M. (1985): Benchmark Problems for CACSD Packages. Proceedings of the 2nd IEEE Control Systems Society Symposium on Computer-Aided Control System Design (CACSD), Santa Barbara, California, March, 13-15, 1985.
- Gavel, D.T., Herget, C.J., Bennet, T.M. and Wang, J.L. (1985): The M Language - An Interactive Tool for Manipulating Matrix Ordinary Differential Equations. Dynamics and Controls Group, Engineering Research Division, Lawrence Livermore National Laboratory, Livermore, California, July 1985.
- Garbow, B.S., et al. (1977): Matrix eigensystem routines - Eispack Guide Extension. Lecture Notes in Computer Science, Vol. 51, Springer-Verlag, New York.

- Gomez, C., Quadrat. J.P., Sulem, A. (INRIA, France), Blankenship, G.L., Kumar, P., LaVigna, A., Mac Enany, D.C., Paul, P. and Yan, I. (University of Maryland) (1984): An Expert System for Control and Signal Processing with Automatic Fortran Code Generation. Proc. IEEE CDC, Las Vegas, Dec 1984, pp. 716-723.
- Goodfellow, S.D. and Munro, N. (1985a): Control System Computer Aided Engineering and Unix. Proceedings of the 2nd IEEE Control Systems Society Symposium on Computer-Aided Control System Design (CACSD), Santa Barbara, California, March, 13-15, 1985.
- Goodfellow, S.D. and Munro, N. (1985b): An Integrated Environment for Computer Aided Control Systems Engineering. CADCE '85, 3rd IFAC/IFIP Symposium on Computer Aided Design in Control and Engineering Systems, Lyngby, Denmark, July 31 - August 2, 1985. pp. 34 - 37.
- Goodfellow, S.D. and Munro, N. (1985c): INTEGRA, An Input Translator Facility for Computer Aided Control Systems Engineering. CADCE '85, 3rd IFAC/IFIP Symposium on Computer Aided Design in Control and Engineering Systems, Lyngby, Denmark, July 31 - August 2, 1985. pp. 104 - 109.
- IEEE CSS (1985): News, Technical Committee on Computer-Aided Control Systems Design, IEEE Control Systems Society, Vol. 1, No. 2, Summer 1985.
- Jacobs O.L.R. (1985): Computer Aided Design of Control System in the United Kingdom. CADCE '85, 3rd IFAC/IFIP Symposium on Computer Aided Design in Control and Engineering Systems, Lyngby, Denmark, July 31 - August 2, 1985. pp. 62 - 67.
- James, J.R., Taylor, J.H. and Frederick, D.K. (1985): An Expert System Architecture for Coping with Complexity in Computer-Aided Control Engineering. CADCE '85, 3rd IFAC/IFIP Symposium on Computer Aided Design in Control and Engineering Systems, Lyngby, Denmark, July 31 - August 2, 1985. pp. 47 - 52.
- Kernighan, B.W., and Pike, R. (1983): The Unix Programming Environment. Prentice-Hall, Englewood Cliffs, NJ.
- King, R.A. and Gray, J.O. (1985): A Flexible Data Interpreter for Computer Aided Design & Simulation of Dynamic Systems. CADCE '85, 3rd IFAC/IFIP Symposium on Computer Aided Design in Control and Engineering Systems, Lyngby, Denmark, July 31 - August 2, 1985. pp. 87 - 91.
- Kopec G. (1983): The Signal Representation Language SRL. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, Boston, MA, 1983.
- Kopec G. (1984a): The Integrated Signal Processing System ISP. Proceedings ICASSP 84 - IEEE International Conference on Acoustics, Speech, and Signal Processing, March 19 - 21, 1984 or IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-32, No. 4, August 1984, pp. 842 - 851.
- Kopec G. (1984b): The Signal Data Base System SDB. Proceedings ICASSP 84 - IEEE International Conference on Acoustics, Speech, and Signal Processing, March 19 - 21, 1984.

- Lawver, B. and Poggio, P. (1985): EAGLES Requirements. Computer Systems Research Group, Engineering Research Division, Lawrence Livermore National Laboratory, Livermore, California, July 1985.
- Little, J.N., Emami-Noeini, A., and Bangert, S.N. (1984): CTRL-C and matrix environments for the computer-aided design of control systems. In Bensoussan and Lions (Eds.): Analysis and Optimization of Systems. Springer Lecture Notes in Control and Information Sciences, Springer, Berlin.
- Maciejowski, J.M. (1984): Data structures for control system design. EUROCON 84, Brighton UK, September 1984.
- Maciejowski, J.M. (1985a): Data structures and software tools in CACSD. Proceedings of the 2nd IEEE Control Systems Society Symposium on Computer-Aided Control System Design (CACSD), Santa Barbara, California, March, 13-15, 1985.
- Maciejowski, J.M. (1985b): A Core Data Model for Computer-Aided Control Engineering. Draft. Engineering Department, University of Cambridge, UK, June 7, 1985.
- Moler, C.B. (1980): MATLAB - User's Guide. Department of Computer Science, University of New Mexico, Albuquerque, USA.
- Moore, R.L., Hawkinson, L.B., Knickebocker, C.G. and Churchman, L.M. (1984): Expert systems applications in industry. Proc. ISA Int. Conf. Houston, Texas.
- Nye, W.T. and Tits, A. (1982): DELIGHT for Beginners. Memorandum No. UCB/ERL M82/55, Electronics Research Laboratory, College of Engineering, University of California, Berkeley.
- Nye, W.T. (1983): DELIGHT. An Interactive System for Optimization-Based Engineering. Memorandum No. UCB/ERL M83/33, Electronics Research Laboratory, College of Engineering, University of California, Berkeley.
- Polak E., Siegel P., Wu, T., Nye W.T. and Mayne D.Q. (1982): DELIGHT.MIMO: An Interactive Optimization-Based Multivariable Control System Design Package. IEEE Control Systems Magazine, Vol.2, No. 4, Dec 1982, pp. 9 - 14.
- Rimvall, M., and Cellier, F. (1984): IMPACT Interactive mathematical program for automatic control theory. In Bensoussan and Lions (Eds.): Analysis and Optimization of Systems. Springer Lecture Notes in Control and Information Sciences, Springer, Berlin.
- Rimvall, M. and Bomholt, L. (1985): A Flexible Man-Machine Interface for CACSD Application. CADCE '85, 3rd IFAC/IFIP Symposium on Computer Aided Design in Control and Engineering Systems, Lyngby, Denmark, July 31 - August 2, 1985. pp. 98 - 103.
- Rockmore, A.J. (1985): The knowledge-based design and synthesis approach to software development. Reasoning Systems Inc., Palo Alto, California, USA. Also in Electronic Design Magazine, July 1985.

- Smith, B.T., et al. (1976): Matrix eigensystem routines - Eispack guide. 2nd ed., Lecture Notes in Computer Science, Vol. 6, Springer-Verlag, New York.
- Smith, D.R., Kotik, G.B. and Westfold, S.J. (1985): Research on Knowledge-Based Environments at Kestrel Institute. Kestrel Institute, Palo Alto, California, USA.
- Spang, H., III (1984): The Federated Computer-Aided Control Design System. Proc. of the IEEE, Vol. 72, No. 12, Dec 1984, pp. 1724 - 1731.
- Taylor, J.H. and Frederick, D.K. (1984): An Expert System Architecture for Computer Aided Control Engineering. Proc. of the IEEE, Vol. 72, No. 12, Dec 1984, pp. 1795 - 1805.
- Trankle, T.L. and Markosian, L.Z. (1985a): Control Design Using an Expert Planning System. Proceedings of the 2nd IEEE Control Systems Society Symposium on Computer-Aided Control System Design (CACSD), Santa Barbara, California, March, 13-15, 1985.
- Trankle, T.L. and Markosian, L.Z (1985b): An Expert System for Control Design. IEE Control 85, Conference Publication Number 252, Vol. 2, pp. 495 - 499.
- WGS (1985a): WGS Newsletter, No. 1, Jan 1985. Benelux Working Group on Software, Dept. of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven, The Netherlands.
- WGS (1985b): An Inventory of Basic Software for Computer Aided Control System Design (CACSD). Benelux Working Group on Software, WGS-Report-85-1. Dept. of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven, The Netherlands, May 1985.

APPENDIX A - EXTERNAL CONTACTS DURING 1985

This is a list of external contacts the Department of Automatic Control, Lund Institute of Technology has had during 1985 that are of interest for the CACE project. The list includes visits to the department and visits of the staff to companies and other universities as well as participation in conferences, symposia, workshops, courses etc.

Our visitors normally get a presentation of our department and our research as well as live demonstrations of our packages for CACE (Simnon, Idpac etc.) so this is not explicitly mentioned in the list below.

January 8

Sven Erik Mattsson visited STFI, Stockholm and got informed of how they use CACE programs like Simnon and Idpac. The steering committee of the CACE project had a meeting in the evening.

January 14 - 25

Karl-Erik Årzén and Per Persson attended the course Structure and Interpretation of Computer programs: SCHEME at Department of Computer and Information Science, Linköping University.

January 21 - 22

Prof. Wolfgang Kreutzer, Dept. of Computer Science, University of Canterbury, Christchurch 1, New Zealand visited the Department of Computer Science and Computer Engineering and gave two seminars titled "Making Smalltalk" and "Simulation Programming: Styles & Languages". Prof. Kreutzer who has a wide experience in discrete simulation has expressed an interest in our project. We have invited him to serve as a consultant to our project. He will initially spend two months with us starting November 15, 1985. We believe that it is extremely valuable to get input from such a competent computer scientist.

January 28 - February 2

Dr. Herbert Hanselman, University of Paderborn, West-Germany gave a course on implementing fast digital control systems using signal processors.

February 6

Conny Björkwall, Bengt Convadi and Jonas Rehnberg from Telelogic visited our department and presented their Clerc system.

February 7

Karl Johan Åström, Karl-Erik Årzén, Per Persson and Leif Andersson visited Gunnar Bohlin at SCA in Sundsvall. AI applications in process and production control were discussed.

March 1

Karl-Erik Årzén visited a seminar by Dr. Robert L. Moore, LMI Process Systems about the LMI Lambda Lisp machine and about PICON, an expert system for process control. AI in process control were discussed together with Robert Moore and Karl Johan Åström.

March 13 - 15

Per Hagander, Sven Erik Mattsson and Tomas Schönthal participated in CACSD'85, The 2nd IEEE Control Systems Society Symposium on Computer-Aided Control System Design (CACSD), Sheraton Santa Barbara Hotel & Spa, Santa Barbara, California. Sven Erik Mattsson presented his paper "Uses of CACSD Tools in Analysis and Design of Control for Windmills". Per Hagander presented the paper "An Expert System Interface for Idpac" by Jan Eric Larsson and Karl Johan Åström.

March 14

Karl Johan Åström was invited to participate in a meeting with the steering committee for the SERC project at Oxford. At that meeting he also presented the CACE project. The steering group gave a very positive reaction to our project

and expressed an interest to be kept informed about our progress. Possible collaboration between our projects was also discussed.

March 18 - April 19

Tomas Schönthal spent the period with Alan Laub in Santa Barbara to learn about software for numerical linear algebra and Riccati equation solvers. He also implemented these in our local version of Synpac and he also made Synpac available to Prof. Laub.

March 18

Sven Erik Mattsson and Per Hagander visited (together with Sten Bergman, ASEA Relays) Dr. Tom Trankle and Phil Chu, Systems Control Technology, Palo Alto and got a presentation and demonstration of their expert planning system for control.

March 19 - 22

Sven Erik Mattsson visited together with Sten Bergman, ASEA Relays the following companies in the San Francisco area.

At Fairchild Laboratory for Artificial Intelligence, Palo Alto Dr. Gary Kopeck presented and demonstrated first the signal processing system ISP and its signal representation language IPL and then the signal database system SDB.

At Analog Design Tools, Inc., Menlo Park Rich Davies and Andrew Hughes demonstrated the Analog Workbench. It is a microcomputer-based workstation (Sun-2) for analog circuit design. The designer creates a circuit using graphics. He can select components from a list and specify their values. The mouse is used to arrange the components on the display and to and draw wires to interconnect them. The mouse is also used to insert voltage and current probes which in turn can be attached to simulated test instrumentation and analysis results are presented in windows just as if you were viewing them on real instrumentation.

At Silicon Graphics, Inc., Mountain View we got informed about the status of the export licence for our purchase of the workstation IRIS 2400. We also got a demonstration of its new facilities. We discussed graphics standard, but they meant that they did not have the resources to participate in the standardization

committees. Silicon Graphics has no plans at the moment to support Lisp on their workstations, but if demand is high enough they will reconsider it.

At Xidak Inc., Menlo Park Gregory Jirak demonstrated Mainsail. It is a portable programming language. It is Algol-like and has modules. The modules can be compiled separately and replaced on-line. The system including editor and debugger gives the programmer a Lisp-like environment while maintaining the advantages of a typed, procedural language.

March 25

Karl-Johan Åström and Sven Erik Mattsson (together with Sten Bergman, ASEA Relays) visited Symbolics, Cambridge, Massachusetts. Symbolics designs, develops and manufactures Lisp machines, but it is no software house. However, Harry Schrobe demonstrated how they used the Symbolics machines to develop internal software for circuit design. They had a model verifier that could compare the circuit diagrams and the circuit layout. It used an ad hoc method which for each transistor calculated a hash number describing the neighbours of the transistor. He stated that simple AI for simple problems for example synthesizing chips is useful. He also thought that design of chips for fast digital control is a nice research project.

March 26 - 28

Asea participates in the Industrial Liaison Program at Massachusetts Institute of Technology (MIT), Cambridge. Sven Erik Mattsson accompanied Sten Bergman, Asea Relays. Sten Bergman had arranged a number of visits dealing with different topics. One interesting visit was that to Webster Dove, Department of Electrical Engineering & Computer Science. He worked on a package for signal processing. Dr. G. Kopec (see March 19 - 22) received his Ph.D at this department. Dr. Kopec's packages are more oriented to calculation, whereas Mr. Dove wanted to duplicate the theoretical properties of signals (infinite, periodic etc)). Mr. Dove hoped to finish his Ph.D this fall and then documentation and program will be available for a nominal fee (the package runs on Symbolics machines).

Sven Erik Mattsson and Sten Bergman also visited Lisp Machines Inc. (LMI). They got a demonstration of PICON, an expert system for process control. They also

learned that LMI will have a landscape screen instead of that small standing A4 screen. Both their Lambda machine and the IRIS systems (Silicon Graphics, Inc.) has a multibus. Mr. Grandfield had used this fact and set up a system with a Lambda machine and an IRIS system. This system gave him fast graphics and a Lisp environment.

April 1 - 3

Professor D.P. Atherton, The University of Sussex, UK gave four lectures on Stability and limit cycles in nonlinear control systems. SUNS (Sussex University Nonlinear Software) was installed on our VAX.

April 22 - May 10

Tomas Schönthal visited the MathWorks Inc., California to learn about development of CACE software for personal computers.

April 26

Sven Erik Mattsson and Karl-Erik Årzén attended a half day seminar on AI arranged by Tektronix. An overview of AI, Lisp, Prolog and Smalltalk was given. The workstation TEK 4404 and Smalltalk were demonstrated.

May 9

Karl-Erik Årzén and Per Persson visited a seminar by Carnegie Group Inc. Their AI tools Knowledge Craft and Language Craft were presented.

May 13

Karl Johan Åström visited Cambridge University to discuss the activity in computer aided design. Cambridge University has obtained a large grant from IBM. Under the direction of prof. Mac Farlane they have a gigantic project aimed at introducing computer design university wide.

May 15

Karl Johan Åström participated in a meeting on "New Developments and

Applications of CAD packages to Control System Design" organized by the Computing and Control Division of the IEE where he gave a talk entitled "The new CACSD project at Lund Institute of Technology".

May 24

Lars Odén, Bo Holmgren, Petr Janecek and Leif Sehlström from ASEA Research and Innovation visited us. Lars Odén presented ASEA Research and Innovation.

May 28

Erik Elgeskog and Kjell Karlsson, Volvo Personvagnar AB visited us and discussed identification of process models for cars.

May 29 - 30

Poster presentations of the new CACE project (Sven Erik Mattsson), the Expert Interface project (Jan Eric Larsson and Per Persson) and the Expert Control project (Karl-Erik Årzén) at Reglermöte 85 (Control Meeting 85), Lund Institute of Technology arranged by the Swedish IFAC committee, Instrumenttekniska föreningen and the Department of Automatic Control, Lund Institute of Technology.

June 11

Karl-Erik Årzén attended a meeting by Svenska Mekanförbundet's supporting committee for Knowledge based systems and Expert systems. Karl-Erik Årzén joined the committee.

June 12

SIGRAD (The Swedish Interest Group on Computer Graphics) and LDC (Lund University Computing Center) arranged a study tour to Lund and visited the Department of Automatic Control, the Department of Electrical Measurements and Lund Observatory. About 30 persons from university and industry participated. Sven Erik Mattsson presented the new CACE project and demonstrated LICS.

June 14

Professor G. Blankenship, Electrical Engineering Department, University of Maryland gave a seminar "Design of nonlinear filters using real and artificial intelligence"

June 18 - 20

Per Persson attended LUND '85, the first Apple University Consortium conference in Europe.

June 17 - 19

Professor Alan Laub, University of California, Santa Barbara gave a course on "Computational Methods in Control Theory".

July 2 - 7

Karl Johan Åström, Björn Wittenmark, Sven Erik Mattsson, Rolf Johansson and Tore Hägglund participated and presented papers at the 7th IFAC/IFORS Symposium on Identification and System Parameter Estimation, University of York, United Kingdom. Karl Johan Åström and Sven Erik Mattsson went also (July 4) to Manchester to discuss CACE projects with Professor Neil Munro and Dr. Simon Goodfellow, UMIST and Professor Dean Frederick, Rensselaer Polytechnic Inst. (RPI), Troy, New York.

July 9

Karl Johan Åström was invited to participate in an informal meeting of the steering group for the SERC project. The desire for a collaboration with our CACE project was reinforced and SERC had made a formal request to STU for collaboration.

July 24 - 29

Dr. Francois Delebecque and Dr. Serge Steer from INRIA, France visited us. Dr. Delebecque gave a seminar on July 26 and presented the interactive system Blaise for control engineering. After the seminar a demonstration of Blaise was given.

We exchanged Simnon for Blaise, so Blaise is now available on our VAX.

July 29 - 30

Magnus Rimvall, ETH, Zurich, Switzerland visited us.

July 30

Dr. Jan Maciejowski, Cambridge University, UK, Dr. Charles Herget, Lawrence Livermore National Laboratory, Livermore, California and Dr. Richard Klabunde, Naval Weapons Center, China Lake, California visited our department. Dr. Maciejowski gave a seminar on data structures for use in CACE. Dr. Charles Herget gave a seminar and presented the Eagles project and the M language.

July 31 - August 2

Sven Erik Mattsson, Per Persson, Jan Eric Larsson and Ulf Holmberg attended the 3rd IFAC/IFIP Symposium CADCE'85, Computer Aided Design in Control and Engineering Systems (Advanced tools for modern technology) at The Technical University of Denmark, Lyngby, Denmark.

August 1

Professor John Gray, Department of Electronic and Electrical Engineering, University of Salford, UK visited our department.

August 2

Professor Neil Munro and Dr. Simon Goodfellow UMIST, Manchester, UK and Dr. Dean Frederick, RPI, Troy, New York visited our department.

August 5

Professor Hany K. Eldeib, George Mason University, Virginia visited our department.

August 27 - 28

14 persons from the department made a study tour to SAAB, Linköping. The assembly factory for the passenger airliner SAAB/Fairchild 340 which takes 34 passengers and simulators for design and evaluation were shown. They also presented some projects within the control field. Sven Erik Mattsson, Per Persson and Karl Erik Årzén presented our CACE project.

Protokoll från mötet med STUs styrgrupp för CACE-programmet, Institutionen för Reglerteknik, Lund, 6/9 1985, kl 10-15,30.

Närvarande: Sven-Gunnar Edlund, Karl Eklund, Claes Källström, Gustaf Söderlind, Karl-Johan Åström, Arne Otteblad (i st f Ove Berkefelt, Göran Bryntse, Thomas Liljemark, STU) samt ca 10 personer från Inst för Reglerteknik i Lund.
Erik Sandewall hade anmält förhinder.

- § 1 Den av Karl-Johan Åström i förväg utskickade dagordningen godkändes. Till sekreterare för mötet valdes Göran Bryntse, STU. Till justeringsman valdes Sven-Erik Mattson, Inst för Reglerteknik, Lund. Föregående mötes protokoll godkändes. En pärm utdelades till mötesdeltagarna med kopior på under mötet visade OH-bilder.
- § 2 Karl-Johan Åström gav en överblick över budgetläget och tidtabellen för projekten, se ovannämnda pärm. Han redovisade också personalsituationen. På g a tidpunkten för STUs beslut kunde IRIS-utrustningen inte beställas förrän okt -84. Leverans av hårdvaran försenad, delvis beroende på administrativt krångel i USA, och kom först i juli -85. Det har också varit svårt att få tag på en kvalificerad datalog. Förseningarna motiverar en revidering av den ursprungliga tidtabellen för CACE-programmets genomförande.

Styrgruppen rekommenderar STU att ta stor hänsyn till de förslag till modifieringar av tidtabellen för CACE-programmets genomförande som föreslås av Inst för Reglerteknik i Lund.

- § 3 Sven-Erik Mattson föredrog läget för projektet "Nya former för man-maskin-kommunikation", se OH-kopiorna i pärmen.


Styrgruppen finner att de förseningar som projektet drabbats av är rimliga och naturliga mot bakgrund av de yttre omständigheter som påverkat projektet. Styrgruppen vill påpeka vikten av att få synpunkter från tänkbara användare på ett tidigt stadium. Ett sätt att göra detta är att arrangera en workshop i Lund med brukare av interaktiv programvara. Vid ett sådant tillfälle skulle man dels presentera projekten, dels samla erfarenheter från existerande paket och dels spekulera över nya möjligheter. En bra tidpunkt för en sådan workshop är någon gång under våren.

- § 4 Jan-Erik Larsson föredrog projektet "Expertsystem interface för Idpac". Styrgruppen godkände lägesrapporten (se OH-kopiorna i pärmen).
- § 5 Planerade projekt. Sven Erik Mattson redogjorde för projektet "Kombination av formelbehandling och numerik", se OH-kopior i pärmen. Karl Johan Åström redogjorde för projektet "Högnivåspråk för lösning av reglerproblem", se OH-kopior i pärmen. Karl-Erik Åhrzén redogjorde för projektet "Expertreglering".
- Styrgruppen anser att projektet "Expertreglering" är ett bra projekt och att det väl passar in i CACE-programmets syften. Styrgruppen stöder därför STUs förslag att projektet ingår i CACE-programmet. Styrgruppen stöder också den föreslagna inriktningen och omfattningen av de två andra redovisade projekten, "Kombination av formelbehandling och numerik" samt "Högnivåspråk för lösning av reglerproblem".
- § 6 Diskuterades inköp av utrustning inom ramen för CACE-programmet. Inst för Reglerteknik begär att inköp av utrustningen ska kunna senareläggas i förhållande till den tidigare uppgjorda budgetplanen. Skälet är att man behöver mer betänketid innan man väljer utrustning. Styrgruppen ansåg att en sådan förskjutning av tidtabellen var OK.
- § 7 Karl Johan Åström redovisade de internationella kontakter som man har inom ramen för CACE-programmet, se OH-kopior i pärmen. Styrgruppen godkände redovisningen.
- § 8 Nästa möte bestämdes att vara tisdag 15/4 1986 på Institutionen för Reglerteknik, Lund. Därefter avslutades mötet och styrgruppen fick en demonstration av de nya arbetsstationerna med realtidsgrafik med höga prestanda från Silicon Graphics.

Vid protokollet


Göran Bryntse
STU

Justeras


Sven Erik Mattsson
LTH