

CODEN: LUTFD2/(TFRT-5419)/1-63/(1989)

Adaptiv parameterstyrning av förbränningsmotorer

Gunnar Klinghult

Institutionen for Reglerteknik
Lunds Tekniska Högskola
December 1989

Department of Automatic Control Lund Institute of Technology P.O. Box 118 S-221 00 Lund Sweden		<i>Document name</i> MASTER THESIS	
		<i>Date of issue</i> December 1989	
		<i>Document Number</i> CODEN: LUTFD2/(TFRT-5419)/1-63/(1989)	
<i>Author(s)</i> Gunnar Klinghult		<i>Supervisor</i> Anders Höglund, Björn Wittenmark	
		<i>Sponsoring organisation</i>	
<i>Title and subtitle</i> Adaptiv parameterstyrning av förbränningsmotorer (Adaptive gain scheduling of a combustion engine.)			
<i>Abstract</i> <p>The project deals with developing of software in order to simulate adaptive gain scheduling of a combustion engine. The control algorithms will be used on a real engine. The purpose has been to obtain a system, which at a given load will find the control parameters at which the engine gives the lowest possible consumption of fuel and values of emission. This means that the engine will optimize itself. The optimized adjustment of the parameters of the engine is stored. The more the engine is run, the more the system will learn about the engine and the better the engine will run. This also makes the system able to correct itself for such changes that are difficult to measure. Examples of this is wearing of the engine or variations of fuel quality.</p>			
<i>Key words</i>			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i>			<i>ISBN</i>
<i>Language</i> Swedish	<i>Number of pages</i> 63	<i>Recipient's notes</i>	
<i>Security classification</i>			

The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

**ADAPTIV
PARAMETERSTYRNING
AV
FÖRBRÄNNINGSMOTORER**

**EXAMENSARBETE AV GUNNAR KLINGHULT M84
SOMMAREN-HÖSTEN 1989**

HANDLEDARE:

ANDERS HÖGLUND

VOLVO TEKNISK UTVECKLING

PROFESSOR BJÖRN WITTENMARK

INSTITUTIONEN FÖR REGLERTEKNIK
LUNDS TEKNISKA HÖGSKOLA

INNEHÅLLSFÖRTECKNING

1	Inledning	3
2	Översiktlig beskrivning	
2:1	Motorstyrning	4
2:2	Problemformulering	6
2:3	Begränsningar	7
3	Beskrivning av simuleringsprogrammet	
3:1	Översiktlig beskrivning av styrmodulen	8
3:2	Optimeringsrutinen	10
3:3	Beskrivning av motormodellen	13
3:4	Flödesschema över programmet	16
4	Resultat	18
5	Användarmanual till Engopt	22
6	Användarmanual till Engplot	25
6.1	Framtagning av motorkarakteristika med hjälp av Engplot	27
7	Slutsatser	28
	Referenser	30
	Appendix	
A1	Beskrivning av använda beteckningar	
A2	Programlistning av Engopt	
A3	Programlistning av Engplot	

1 INLEDNING

Med dagens allt större belastning på miljön i kombination med stigande bränslepriser kommer kraven att öka på förbränningsmotorn. En självklarhet i detta sammanhang är att motorns driftsförhållande ska vara optimalt. Med optimalt menas här, att för ett givet effektuttag ska bränsleförbrukning och emissioner vara så låga som möjligt. Ett sätt som detta kan lösas på är genom en effektiv reglering av motorns driftsparametrar.

De reglersystem som finns i bruk på dagens bilar kan beskrivas som en sorts parameterstyrning. Syftet med detta examensarbete har varit att prova en förbättring av de befintliga systemen. Förbättringen gäller speciellt systemets förmåga att uppnå optimalt driftsförhållande. Min handledare på Volvo för detta examensarbete, Anders Höglund, hade ett förslag på hur ett sådant system skulle kunna vara uppbyggt. Detta förslag kan i princip beskrivas som parameterstyrning i kombination med extremalsökning.

För att prova om denna princip kan fås att fungera valdes att simulera ett tänkt system bestående av en förbränningsmotor och ett styrsystem. Detta har resulterat i att programvara har utvecklats för att simulera systemet motor-styrsystem.

2 ÖVERSIKTLIG BESKRIVNING

2:1 MOTORSTYRNING

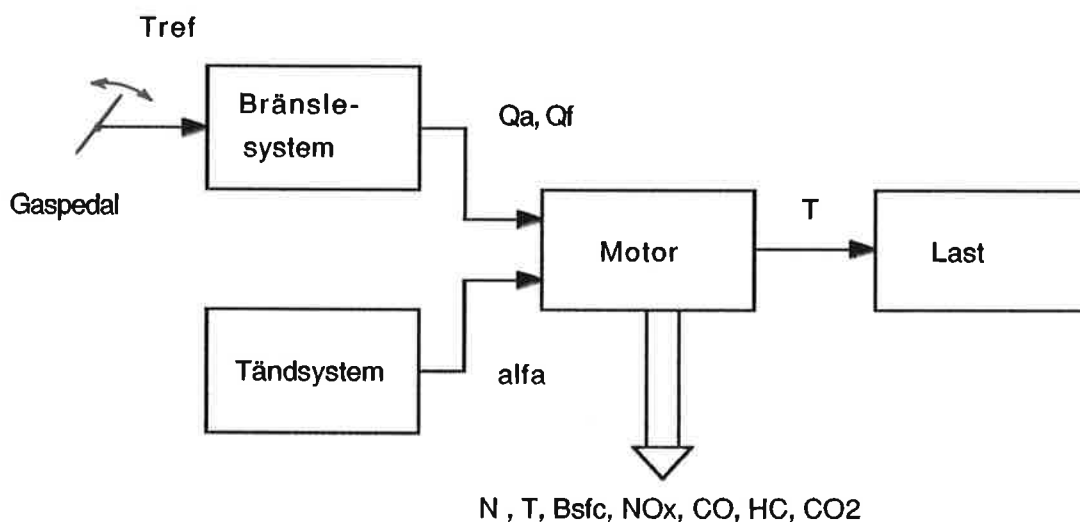
Motorn som ska styras är en Ottomotor. Det går givetvis även att tillämpa idén på en dieselmotor.

En förbränningsmotor är en process, som är förhållandevis komplicerad att reglera. Den som har studerats, har sju utsignaler och tre insignaler. Processen, som ska regleras, är kraftigt olinjär, och kännedomen om den är dålig.

De parametrar som man enkelt kan styra i en förbränningsmotor är bränsleflödet (Q_f), luftflödet (Q_a) och tändförställningen (alfa). Dessa är insignaler till motorn.

Tändsystemet ger den gnista som tänder bränsle-luft blandningen. Vevvinkeln när denna gnista ges kallas för tändförställningen. Bränslesystemet sörjer för att en homogen bränsle-luftblandning åstadkommes. Luftflödet ställs in med ett spjäll. Bränslet tillförs motorn med ett insprutningsaggregat. I dagens läge är bränsleinsprutning det enda sätt där man kan styra bränsleflödet så exakt att det kan utnyttjas i ett reglersystem.

De utsignaler från motorn som mäts upp i detta fall är varvtalet (N), momentet (T), bränsleförbrukningen (B_{sfc}) och emissioner (NO_x , HC , CO , CO_2). Se figur 2.1.



Figur 2.1 Schematisk beskrivning av en förbränningsmotor.

Referensmomentet (Tref) ställs i en bil in med hjälp av gaspedalen. Referensmomentet svarar då mot en given vinkel. Varvtalet är beroende av den last som bromsar motorn.

De system som används för att styra motorerna i dagens bilar är i regel en form av parameterstyrning. Parameterstyrning innebär att regulatorparametrarna ändras i förhållande till driftsförhållandet på ett förprogramerat sätt [1]. Denna förprogramering kan bestå av en tabell där parametrarna lagras.

En kort beskrivning av hur ett kommersiellt parameterstyrt system (Bosch) fungerar ges nedan. Detta mikroprocessorbaserade system fungerar i stort sett så att tillverkaren av systemet bestämmer sig för ett antal värden på styrparametrarna till motorn som ska gälla för ett antal givna driftsfall. Dessa lagras i en tabell. Vid drift går systemet in i tabellen och interpolerar fram ett värde. Detta värde justeras sedan med hänsyn till bland annat omgivande temperatur och skickas som styrsignal till motorn. Den katalytiska avgasreningen kräver att lambda är lika med ett för att fungera. Därför används som återkoppling avkänning av en lambdasond. Lambda är förhållandet mellan den av motorn insugna luftmängden, och den teoretiskt behövliga luftmängden för att förbränna den insprutade bränslemängden. Att ligga runt lambda lika med ett är ej nödvändigtvis optimalt varken ifråga om bränsleförbrukning eller emissioner.

En förbättring av ett parametersstyrt system fås genom att införa möjligheten att optimera systemet. Tabellen där parametrarna lagras skall då uppdateras kontinuerligt. Reglersystemet måste fungera så att momentet ut från motorn ska ställas in efter referensmomentet. Därefter ska bränsleförbrukning och emissioner minimeras. Här finns inget referensvärde att ställa in efter utan det gäller att söka ett minimum hos dessa utsignaler genom optimering med hjälp av insignalerna till motorn. Medan systemet söker detta minimum får inte momentet variera så mycket att det ger upphov till en ryckig gång hos motorn

Ett system som detta skulle hela tiden kunna kompenseras för förändringar av driftbetingelserna även sådana som är mycket svåra att mäta, till exempel förslitning av motorkomponenter och variationer i bränslets sammansättning.

2:2 PROBLEMFÖRMULERING

Detta examensarbete har gått ut på att skapa programvara för att möjliggöra simulering av motorstyrning på dator. Detta för att prova om principen är riktig. Simulering gör det möjligt att snabbt och enkelt utföra ändringar av systemet. Det är meningen att styralgoritmerna senare ska användas till att styra en verklig motor.

För ändamålet har följande programvara utvecklats:

- Engopt
Ett program som simulerar systemet motor-styrprogram.
- Engplot
Ett program som grafiskt åskådliggör motorkarakteristika i form av så kallade musseldiagram. Med hjälp av mätdata från en verklig motor kan de konstanter som ingår i den motormodell som används av simuleringsprogrammet bestämmas.

Framtagningen av programmen har skett på en IBM PC kompatibel dator.

För att köra programmen krävs en EGA färgskärm, och helst en matematikprocessor. Det tar annars mycket lång tid att rita upp kurvorna.

Programmen har skrivits i ett språk som heter Quick-Basic. Detta är ett interpreterande språk med den stora fördelen att man snabbt kan göra ändringar och köra program. Det kan även separat-kompileras, vilket ger en programversion som kan köras med maximal hastighet. Detta språk är en vidareutveckling av det vanliga Basic och det tillåter en programstruktur liknande den Pascal har. Språket har även gott om olika funktioner, bland annat åtskilliga för grafik.

2:3 BEGRÄNSNINGAR

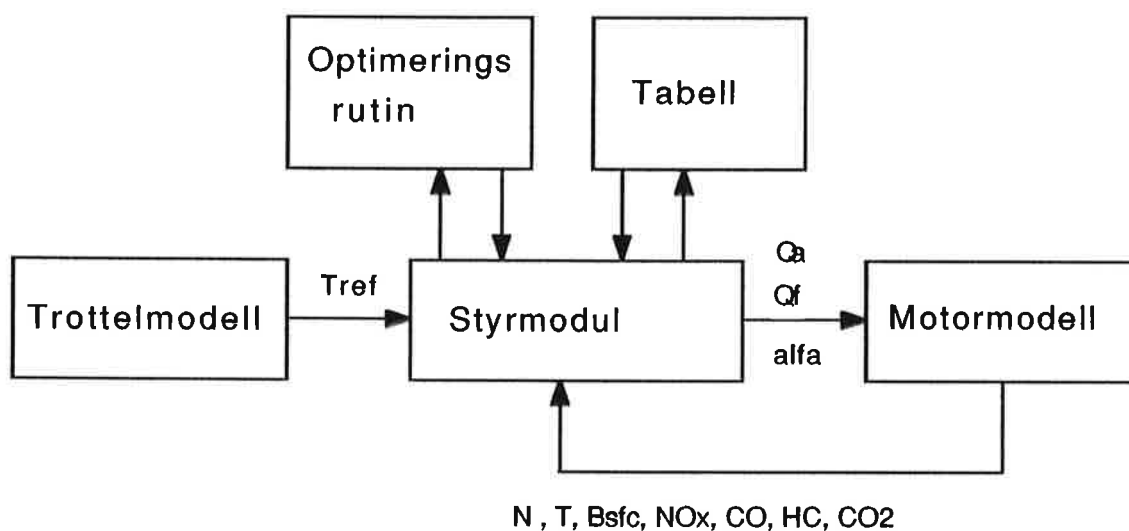
Systemet som beskrivs i detta examensarbete klarar ej av att behandla de transienta förlopp som uppkommer vid gaspådrag eller kraftigare belastningsförändringar, till exempel en uppförsbacke. En anledning till detta är att tabellens data ej är giltigt under transienter. För att kringgå detta problem kan en separat tabell avsedd för transienta förlopp kopplas in i stället för den ordinarie tabellen. En annan anledning är att motormodellen inte längre är giltig under en transient. Vad som händer i en motor under en transient är ej till fullo utforskat.

3 BESKRIVNING AV SIMULERINGSPROGRAMMET.

3:1 ÖVERSIKTLIG BESKRIVNING AV STYRMODULEN

Idén med att simulera systemet motor-styrprogram går ut på att plotta utsignalerna från motorn som funktion av tiden. De ut signaler som vi har valt att använda är varvtal (N), moment (T), emissionsvärdena (NO_x, CO, HC, CO₂) och bränsleförbrukning per utträttat arbete (Bsfc).

Simuleringsprogrammet består dels av en styrmodul som i sin tur anropar en tabell och en optimeringsrutin, dels av två olika modeller som tillsammans ska simulera en verklig motor. Se figur 3.1 som visar ett blockschema över simuleringsprogrammet.



Figur 3.1 Simuleringsprogramets uppbyggnad

Trottelmodellen ger som utsignal till styrmodulen önskat moment som funktion av tiden. För varje gång rutinen anropas så räknas tiden upp med en enhet. Vid plottningen avsätts tiden på x-axeln och datorns snabbhet får bestämma hur lång tid det tar att plotta kurvan.

I tabellen finns lagrat värden på Q_a, Q_f och alfa som funktion av N och Tref. Tabellen är på femtio gånger femtio punkter vilket är tillräckligt många för att interpolering inte ska behövas. En punkt innehåller en uppsättning värden på Q_a, Q_f och alfa. Dessa värden skickas till motormodellen som svarar med att skicka tillbaka värden på emissioner, moment, varvtal och bränsleförbrukning. Dessa mätsignaler stoppas in i något som vi kallar målfunktion.

Optimeringsrutinen söker minimera denna funktion. I målfunktionen multipliceras de olika mätsignalerna med viktfactorer. De optimala utsignalerna beror av viktfactorernas storlek.

När antingen N eller Tref ändras över ett visst tröskelvärde så avbryts optimeringen. Det framoptimerade värdet lagras i tabellen och ett nytt startvärde till optimeringsrutinen hämtas ur tabellen.

Eftersom optimeringen går till så att den provar ett värde, mäter upp motorns svar, provar ett nytt värde osv, så blir optimeringen långsam i förhållande till motorns varvtal. Om de värden optimeringsrutinen provar ligger långt ifrån de optimala så blir motorns gång ryckig. För att undvika detta så används en tabell där de startvärden som optimeringsrutinen utgår ifrån lagras. Dessa startvärden kommer att ligga nära de rätta värdena och optimeringen kommer snabbt att hamna rätt. För att tabellens värden alltid ska ligga så nära rätt värde som möjligt, oberoende av eventuella förändringar, så uppdateras tabellen kontinuerligt under körningen.

Utgångsläget för tabellen är att den är fylld med värden som ger ett lambda värde lika med ett. I början, innan samtliga driftsfall hunnit att köras och tabellen blivit uppdaterad så aviker optimeringsrutinens startvärden kraftigt från de värden tabellen har när ett optimum har nåtts. Detta medför att samtliga parametrar svänger fram och tillbaka i början tills ett optimum har nåtts. Därefter kommer systemet att ligga och finjustera tabellens värden hela tiden.

När systemet ska provas på en verklig motor så går det att först köra en simulering för just denna motor så att tabellvärdena står någorlunda rätt. Sedan flyttas tabellen över till den verkliga motorn. Då kommer startvärdena i tabellen att vara någorlunda bra, givetvis beroende på hur bra den av simuleringensprogrammet använda motormodellen är. Om det finns tillförlitliga mätdata från motorn som ska styras så används dessa givetvis.

3:2 OPTIMERINGSRUTINEN

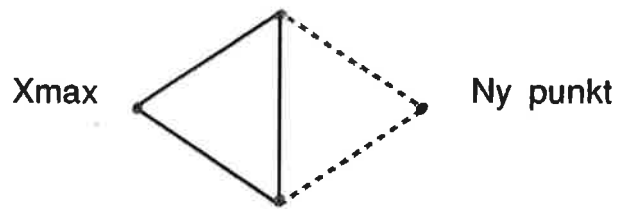
Optimeringsrutinen är skriven av Anders Höglund, Volvo Teknisk Utveckling.

Optimeringsproblemet går ut på att minimera den så kallade målfunktionen f_m .

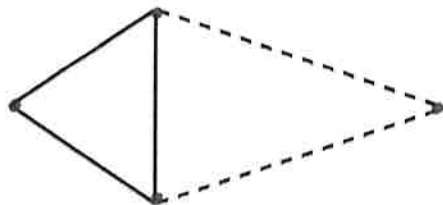
$$f_m = k_1 * Bsfc(Q_a, Q_f, \alpha) + k_2 * NO_x(Q_a, Q_f, \alpha) + k_3 * CO(Q_a, Q_f, \alpha) + k_4 * CO_2(Q_a, Q_f, \alpha) + k_5 * HC(Q_a, Q_f, \alpha) + k_6 * \frac{|T - T_{ref}|}{T + 1}$$

där $k_1, k_2, k_3, k_4, k_5, k_6$ är konstanter och kallas för viktfaktorer. För att önskat moment ska hållas någorlunda konstant så söker rutinen minimera felet mellan inställt och önskat moment. Om k_6 väljs för liten i förhållande till de övriga viktfaktorerna så kommer momentet att svänga fram och tillbaka medan optimeringsrutinen provar sej fram. För att optimeringen ska fungera måste N och T_{ref} vara konstanta. Så fort någon av dem ändras så avbryts optimeringen, de framoptimerade värdena sparas i tabellen och nya startvärden till optimeringsrutinen hämtas ur tabellen.

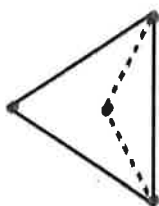
Optimeringen sker enligt simplexmetoden [2]. Om den optimerar i n dimensioner, så krävs $n+1$ hörnpunkter. Målfunktionens värde i dessa hörnpunkter jämförs. Utgående från dessa jämförelser sker en förflyttning mot minimipunkten. Förflyttningen sker med operationerna reflexion, expansion, krympning eller kontraktion. I det tvådimensionella fallet ser förflyttningarna ut som i figur 3.2 Vid krympning byts alla punkter utom den minsta ut.



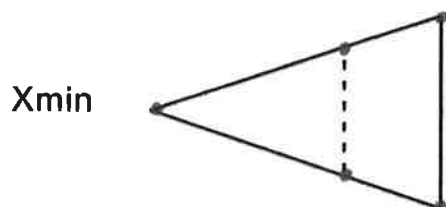
Reflexion



Expansion

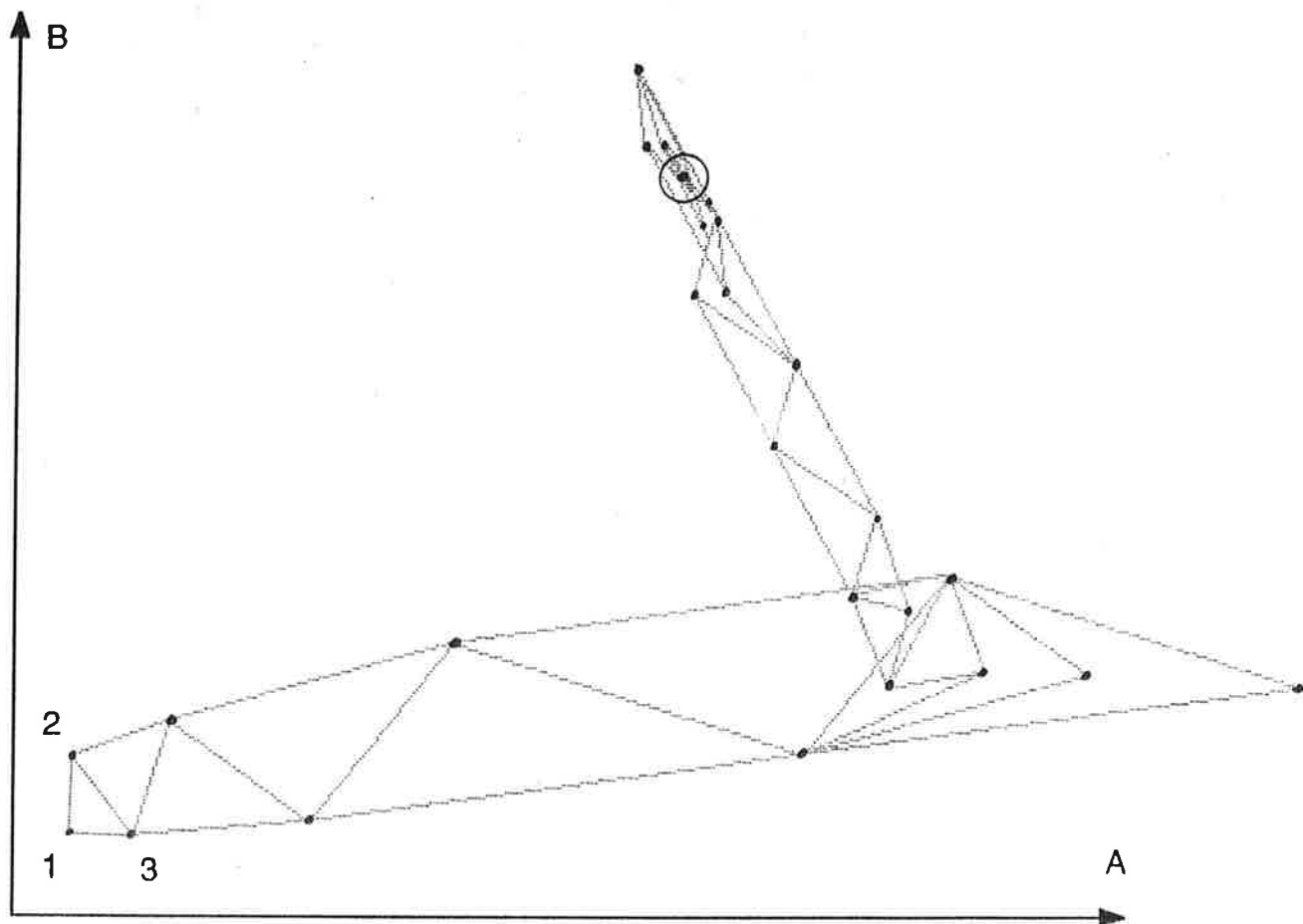


Kontraktion



Krympning

Figur 3.2 Förflyttningar i simplexmetoden

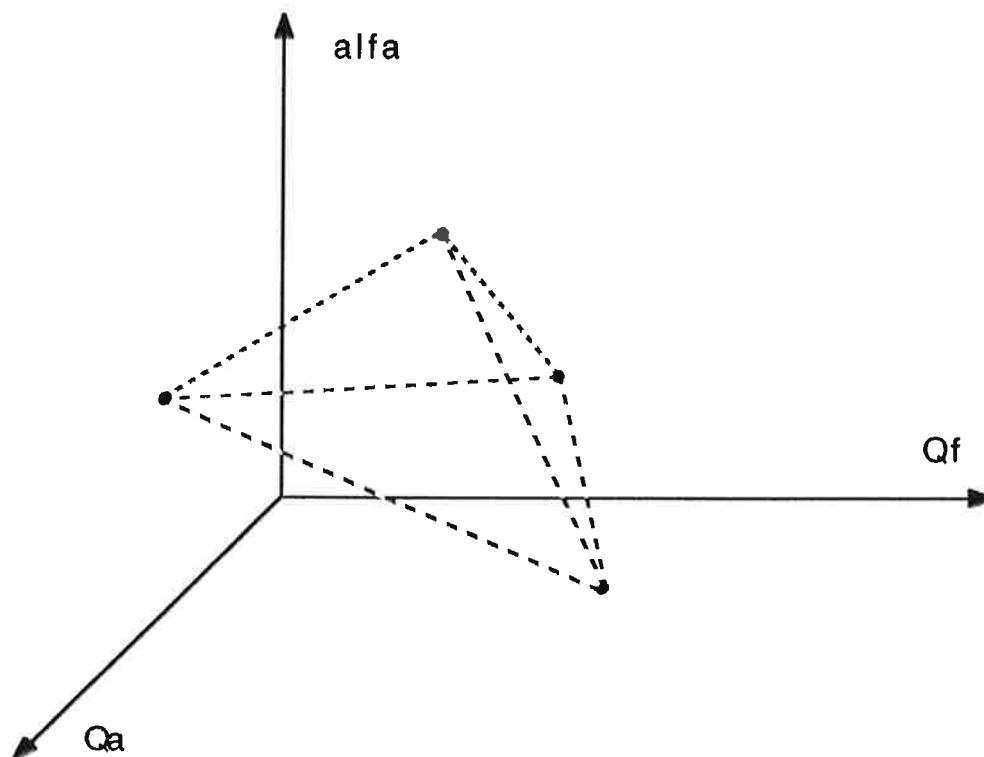


Figur 3.3 Exempel på sökning av min i två dimensioner.

Figur 3.3 ger ett exempel på en sökning av minimum i två dimensioner. Punkterna 1, 2 och 3 är de startpunkter som optimeringsrutinen utgår ifrån. Vid den inringade punkten har funktionens minimum nåtts.

I vårt fall är antalet dimensioner tre (Q_a , Q_f , α).

Optimeringsrutinen behöver då fyra hörnpunkter eftersom den optimerar i tre dimensioner. Man kan tänka sej en tetraeder placerad i rummet. Se figur 3.4.



Figur 3.4 Optimering i tre dimensioner.

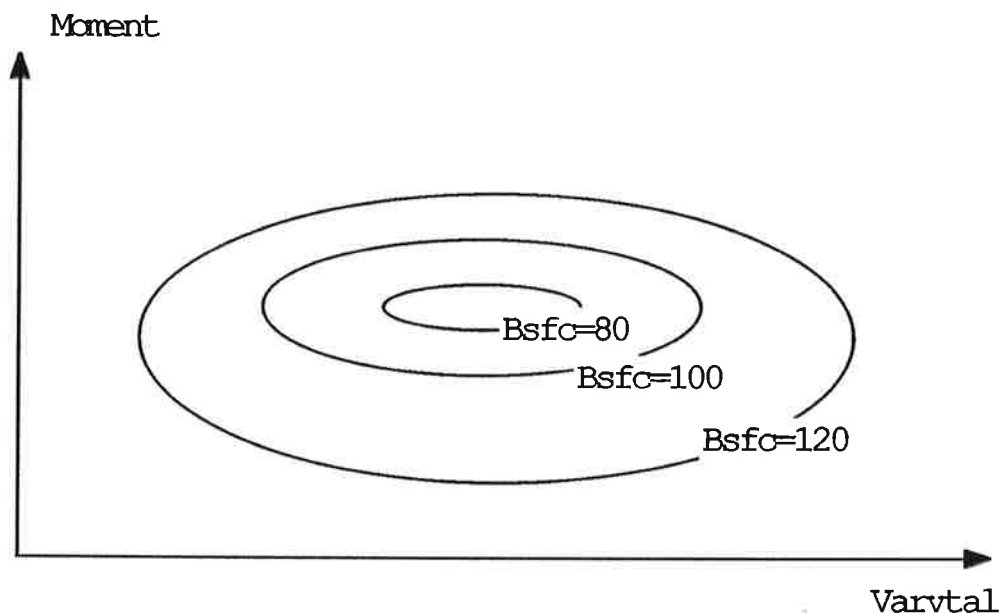
Varje hörnpunkt definieras av de tre parametrarna Q_a , Q_f , α samt målfunktionens (f_m) värde för dessa parametrar som motormodellen ger.

Viktfaktorerna har den betydelsen att de påverkar de enskilda termernas jämviktstillstånd. Detta kan exempelvis utnyttjas så att om det är extra viktigt att nedbringa CO utsläppen så sätts viktfaktorn k_3 större än de övriga.

Eftersom optimeringsrutinen hela tiden provar nya värden så kommer moment och varvtal att oscillera svagt. En variation på upp till en procent är godtagbart och kommer ej att märkas.

3:3 BESKRIVNING AV MOTORMODELLEN

Ett sätt att beskriva en motors karakteristik är med hjälp av musseldiagram. I musseldiagrammet ritas en parameter som funktion av varvtal och moment. Detta ser ut som nivåkurvor i diagrammet. Figur 3.5 visar exempel på ett musseldiagram.



Figur 3.5 Musseldiagram.

Detta ger en överskådlig bild av motorns prestanda för den aktuella parametern, och den visar även de mest fördelaktiga driftsförhållandena. I motormodellen har vi försökt att efterlikna detta genom att ansätta ekvationen för en ellips.

$$f(x, y) = \frac{(x-x_0)^2}{a} + \frac{(y-y_0)^2}{b}$$

Sedan har ekvationerna justerats med hänsyn till bland annat tändförställningen och bränsle-luft-förhållandet för att få en trolig överensstämmelse med en verklig motor.

Motormodellen som används i programet beskrivs i figur 3.6. F_i är bränsle-luft förhållandet. De tre första beräkningarna som ger F_i , F_{id} och d_d används som justeringskoefficienter för att musseldiagrammen ska stämma överens med en verklig motor. Sedan sker en iterativ sökning av momentet (T) och bränsleförbrukningen ($Bsfc$). Om inte det föreskrivna felet uppnås inom tio iterationer så avslutas loopen och de framtagna värdena på moment och bränsleförbrukning används. Slutligen beräknas emissionerna.

Insignaler till motormodellen är Q_a , Q_f och α . Utsignaler från modellen är T , N , $Bsfc$, NO_x , HC , CO och CO_2 .

Bokstavsbezeichnungarna a_1, b_1 osv, är de konstanter som specificerar just den motor som modellen försöker att efterlikna. Dessa konstanter kan bestämmas med programet Engplott.

$$Fi = 15 * \frac{Qf}{Qa}$$

$$Fid = f1 * (Fi - 0.8)^2 + 1$$

$$dd = \frac{e1}{e2^2} * \left[\alpha - e2 * \left(1 + (20 * (Fi - 1.1)^2 + 1) * \frac{N}{100} \right) \right]^2 + 1$$

do until felet < 0.001

$$Told = T$$

$$Bsfc = Fid * dd * \left(a1 * \left[\frac{|Tref - To|}{Tref + b1} + \frac{(N - No)^2}{c1} \right] + d1 \right)$$

$$T = \frac{d1 * To * No * Qf}{Bsfc * N}$$

$$felet = |Told - T|$$

loop

$$NOx = \frac{a2 * Bsfc}{b2} * \frac{(Fi - 0.9)^2 + 1}{c2 * (Fi - 0.9)^2 + 1} + \left(d2 * \left(\frac{\alpha}{c2 + 1} \right)^2 + 1 \right) * \frac{3 * Tref + To}{4 * To}$$

$$HC = \frac{a3 * Bsfc}{b3} * \left(1 + 15 * c3 * \frac{Qf}{Qa} \right)$$

$$CO = \frac{15 * a4 * Bsfc}{b4} * \frac{Qf}{Qa}$$

$$CO_2 = \frac{44 * a5 * Bsfc}{12 * a5 + b5}$$

Figur 3.6 Motormodellen

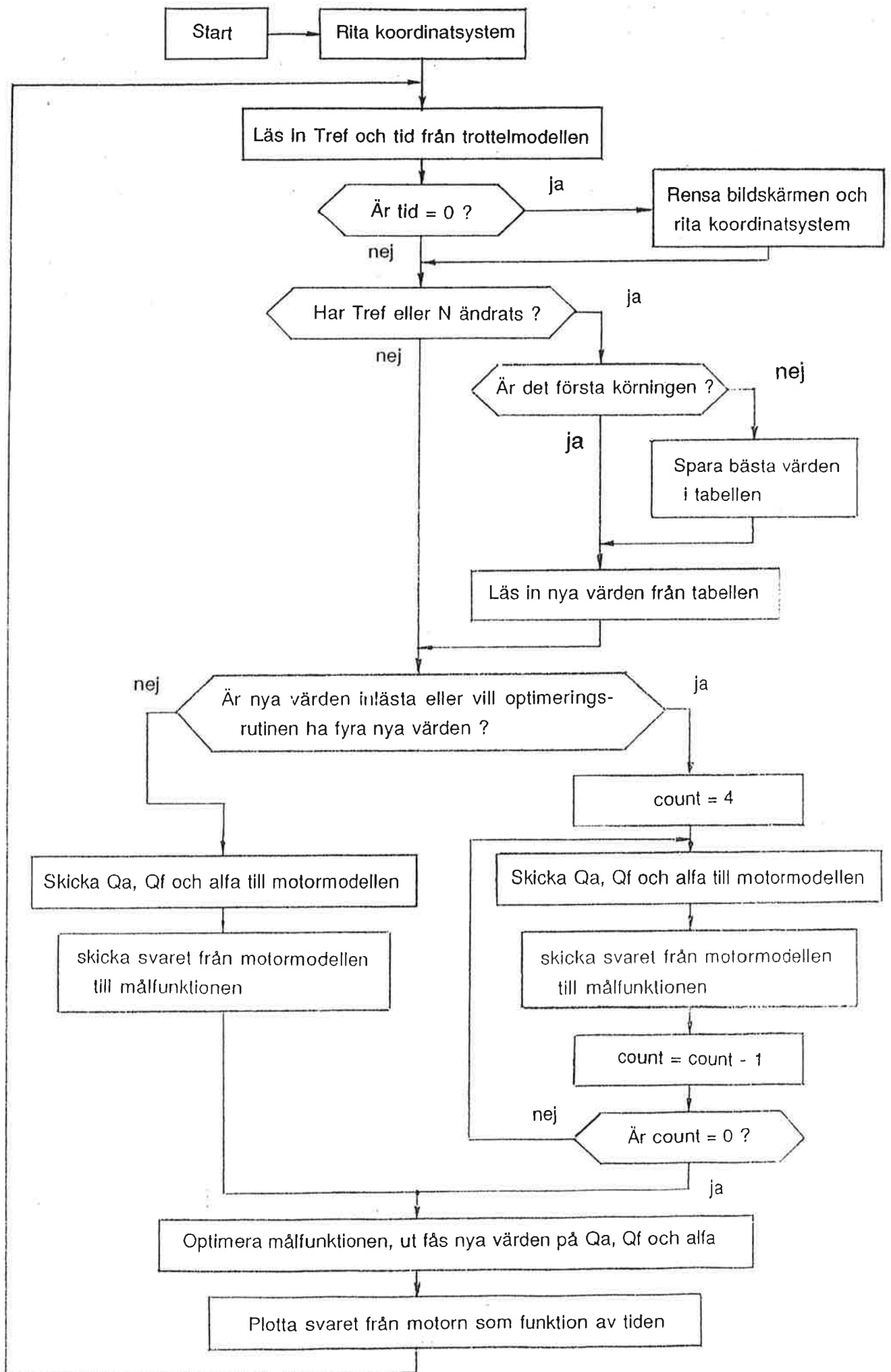
3:4 FLÖDESSCHEMA ÖVER PROGRAMMET

Figur 3.7 visar ett flödesschema över simuleringsprogrammet. Här har endast de funktioner som har med styrsystemet att göra medtagits.

En körning inleds med att ett koordinatsystem ritas upp. Tiden räknas upp med en enhet för varje gång loopen körs igenom. Detta upprepas tills bildskärmen är fylld i x-led. Då rensas bildskärmen, koordinatsystemet ritas upp och tiden nollställs. Q_a , Q_f och alfa skickas till motormodellen. Svaret från motormodellen läses in i målfunktionen, samt ritas upp i koordinatsystemet. Målfunktionen optimeras och ut fås nya värden på Q_a , Q_f och alfa att prova.

Därefter börjar loopen om från början.

Om Tref eller N har ändrats eller om det är första gången systemet körs, så görs följande: De värden på Q_a , Q_f och alfa som har optimerats fram sparas i tabellen. Nya värden på Q_a , Q_f och alfa hämtas ur tabellen för dom nya värden på Tref och N som föreligger. Ur dessa startvärden skapas fyra hörnpunkter. Detta sker genom att adera till ett givet steg i en given riktning så att fyra hörn fås. För vart och ett av dessa hörn behövs sedan ett värde på målfunktionen. Detta fås genom att i tur och ordning skicka värdena i hörnpunkterna till motormodellen och sedan läsa in svaret i målfunktionen.



Figur 3.7 Flödesschema över programet

4 RESULTAT

För att åskådliggöra resultatet har ett antal diagram ritats upp. Avsikten är att studera hur snabbt systemet ställer in sig efter en förändring.

Referensmomentet har här specificerats att motsvara en fyrkantvåg som pendlar mellan värdena tio respektive tjugo Newtonmeter. Detta för att visa hur stegsvaret ändras allteftersom systemet når optimum och tabellen blir uppdaterad. I figuren 4.1 kan man se hur insignalerna till motormodellen, (Q_a , Q_f , α), varierar medan sökningen av optimum pågår.

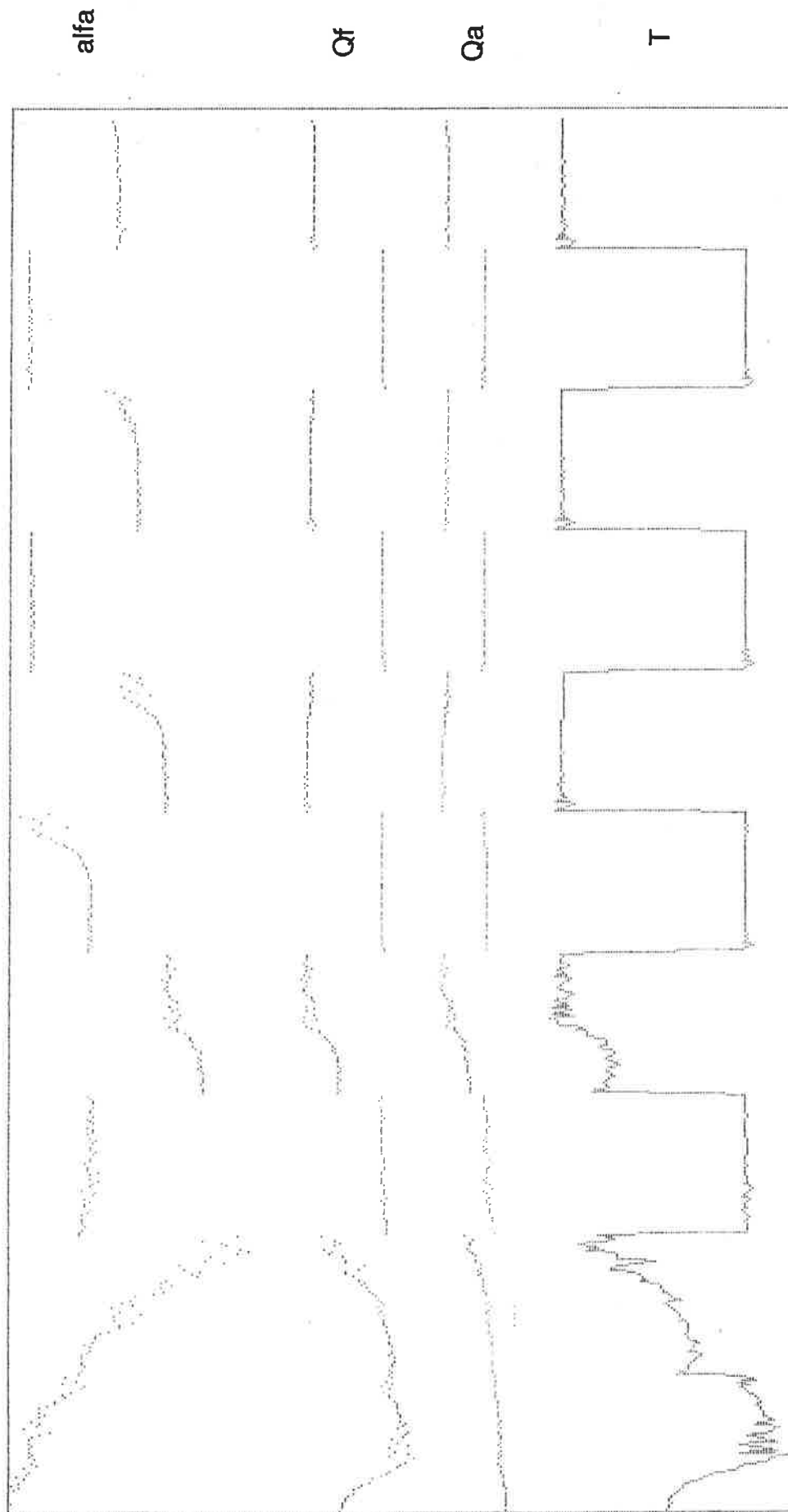
Utsignalerna för dessa insignaler visas i figur 4.2.

Det är stor skillnad mellan stegsvaret då systemet startar från noll och det inte finns några värden i tabellen och ett senare stegsvar då parametrarna i tabellen ligger approximativt rätt.

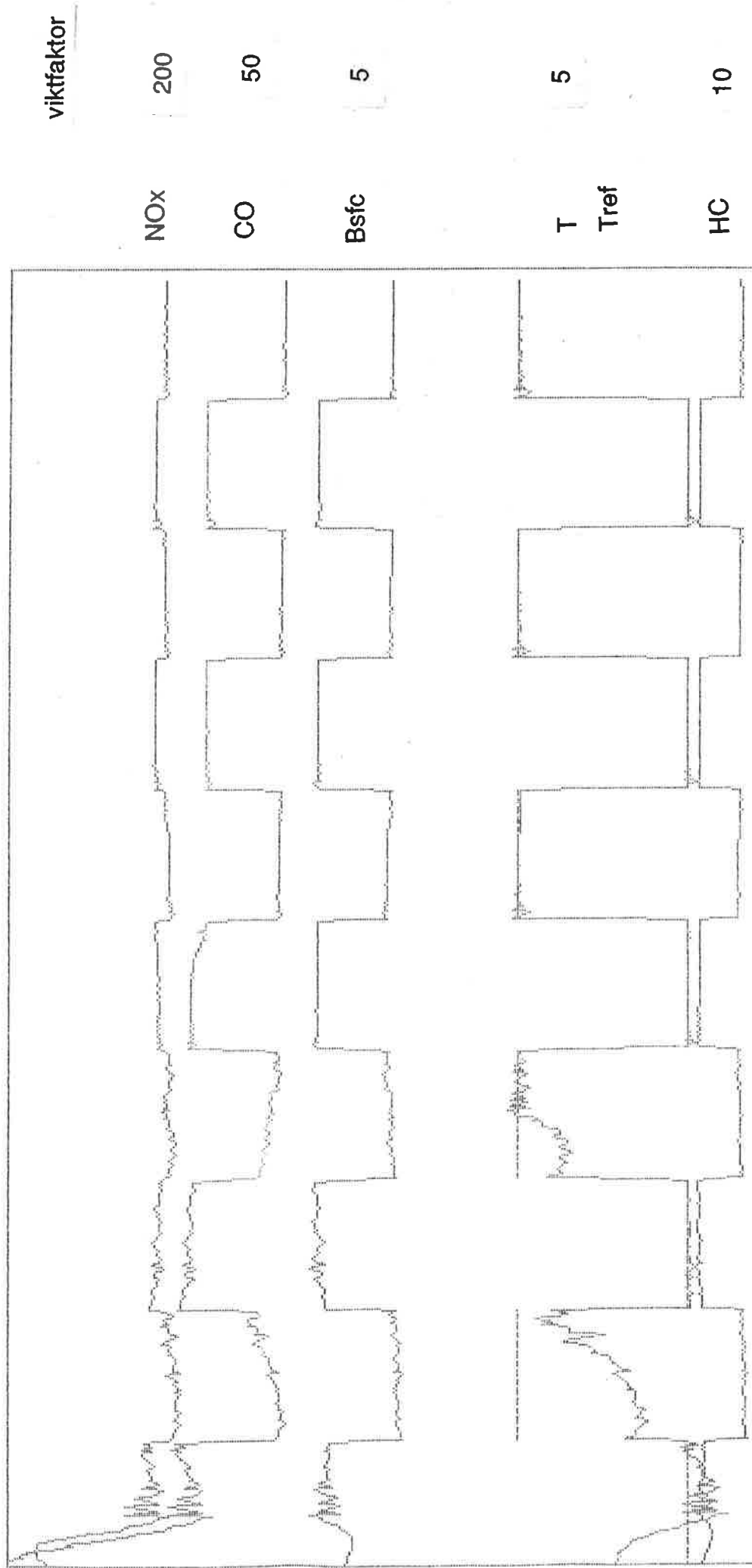
Det är även intressant att se resultatet då viktfaktorerna ändras. En kraftig viktning av en parameter innebär att den snabbare svänger in sig, detta på bekostnad av de andra parametrarnas insvängningstid.

Figur 4:2 och 4:3 visar två körningar där det enda som skiljer är viktfaktorerna. I figur 4.2 har viktfaktorerna för NO_x och CO satts väsentligt högre än i figur 4.3. Detta medför att värdena på NO_x och CO ställer in sig snabbare i figur 4.2 än i figur 4.3. Detta sker på bekostnad av insvängningstiden för de övriga utsignalerna.

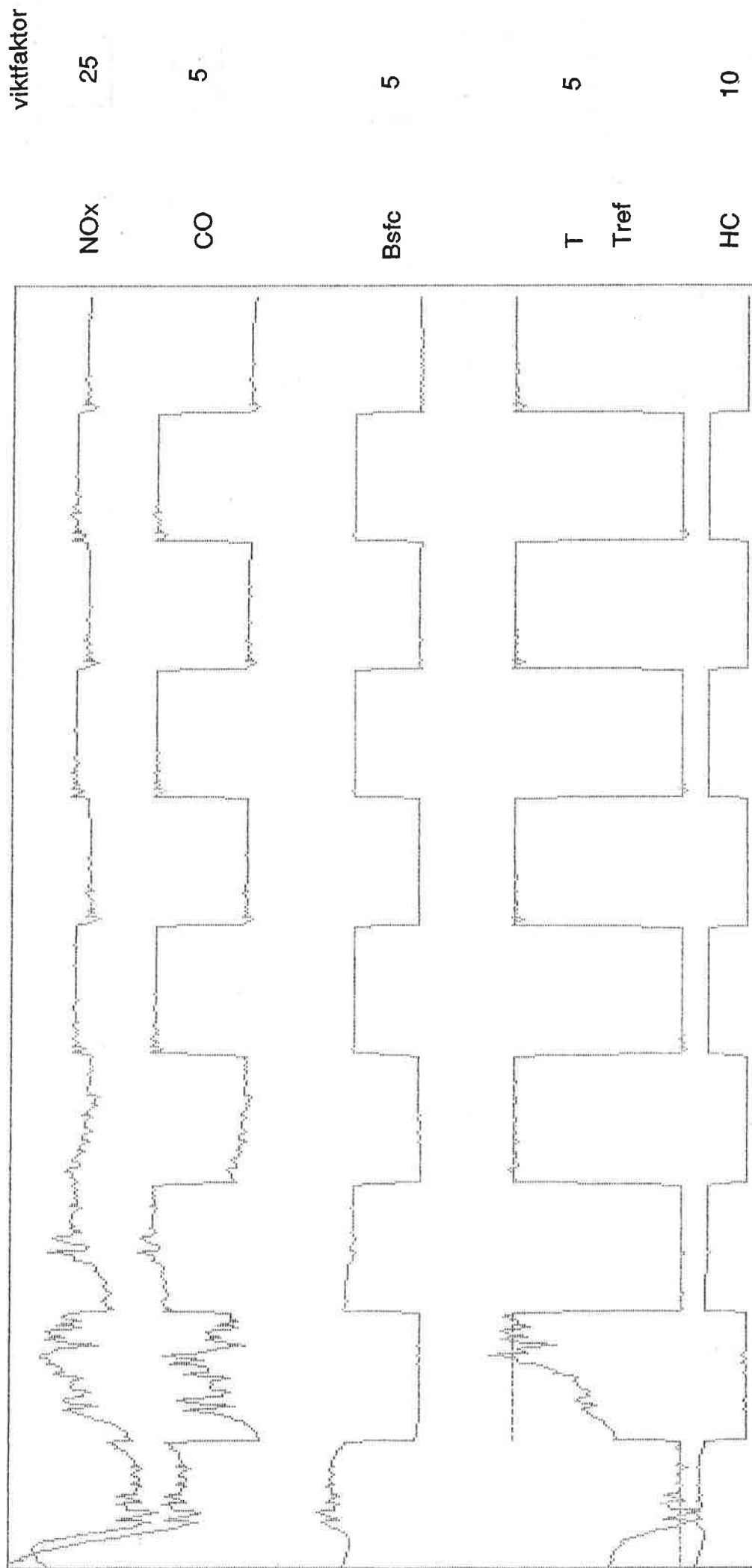
Figur 4.1 Insignaler (Qa, Qf, alfa) till motormodellen. En utsignal från motormodellen.



Figur 4.2 Utsignaler från motormodellen



Figur 4.3 Utsignaler från motormodellen



5 ANVÄNDARMANUAL TILL ENGOPT

Engopt är ett program som simulerar systemet motorstyrprogram. Ut och insignaler kan åskådliggöras grafiskt.

Man kan välja följande funktioner ur huvudmenyn:

- set plot parameters
Här kan man bestämma vilka parametrar som man vill plotta, alternativt plotta alla på en gång. Om man vill kan man få parametrarnas siffervärden utskrivna på skärmen under körningen. Man kan välja skalfaktorer på kurvorna, bestämma koordinatsystemets axlar och välja vilken färg respektive kurva ska ha.
- set throttle parameters
Här väljer man Tref eller N.
- set engine parameters
Här ändrar man konstanterna i motormodellen.
- set weight factors
Här ändrar man viktfaktorerna i målfunktionen
- save all parameters
Här kan man lagra alla använda parametrar från en körning i en extern fil. Den lagras i filen par nummer.dat .
- recall all parameters
Hämtar parametrarna från en tidigare körning
- save kennfeld
Tabellen lagras på samma sätt som parametrarna i filen kf nummer.dat
- recall kennfeld
Hämtar tabellen från en tidigare körning

- run engine

Kör simuleringsprogrammet. Om inte tabell eller parametrar specificeras så används default värden. Körningen avbryts med enter, och sedan kan man antingen fortsätta med samma körning igen eller lagra undan parametrar och tabell och börja en ny körning.

Figur 5:1 visar hur skärmen kan se ut när man skriver ut samtliga värden. Eftersom skärmen visas i färg så blir överskådligheten i verkligheten bättre än vad figuren ger sken av.

6 ANVÄNDARMANUAL TILL ENGPLOT

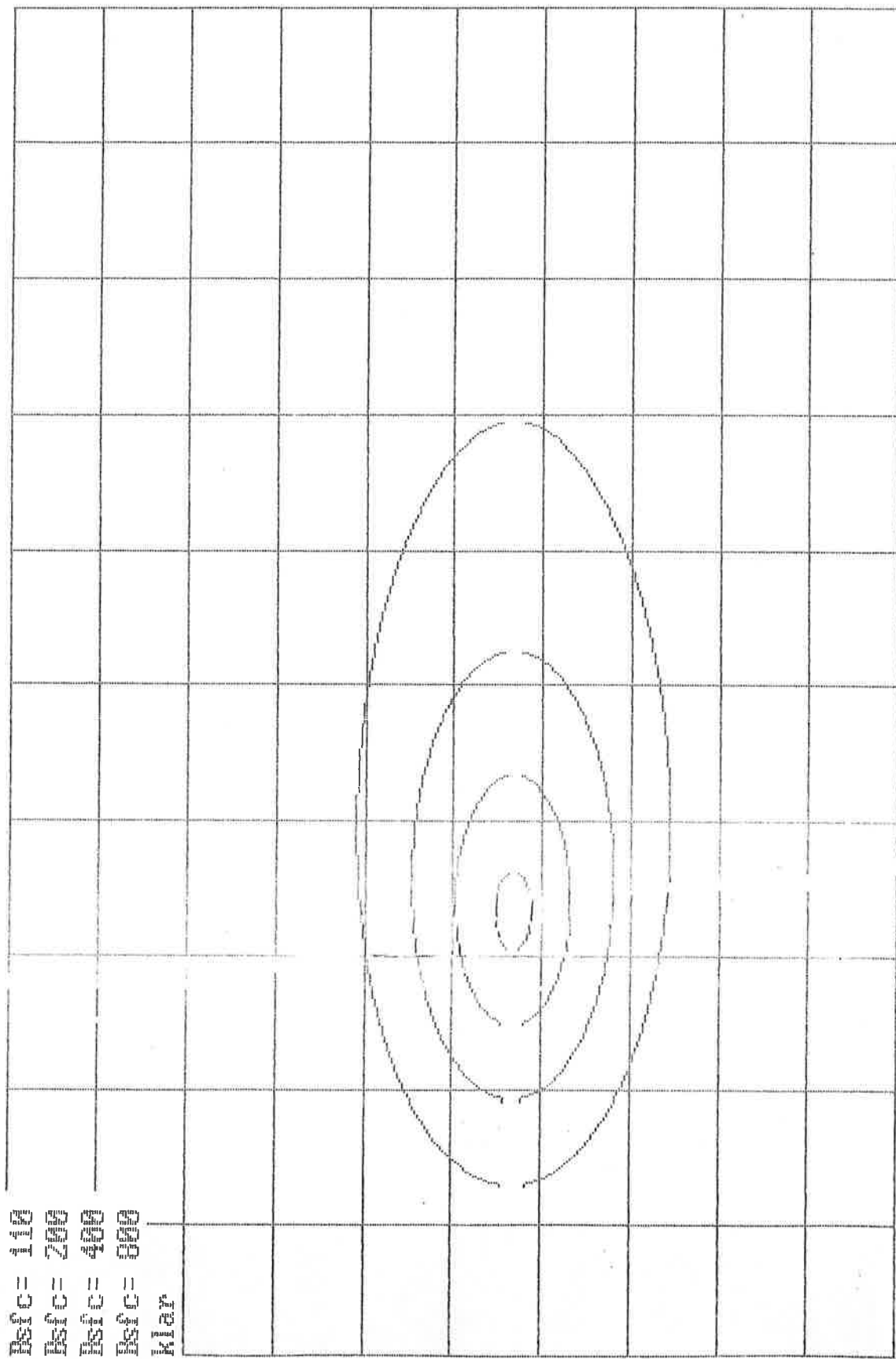
Engplot är ett program som grafiskt åskådliggör motor-karakteristika i form av musseldiagram. Syftet är att bestämma de konstanter som ingår i den motormodell som används av Engopt.

Man kan välja följande funktioner ur huvudmenyn:

- set plot parameters
Definiera koordinatsystemet
- set input parameters
Först väljer man värden på alfa, Qa, Qf, dessa hålls konstanta för ett diagram.
Sedan väljer man antalet nivåkurvor och värden på dessa. Detta görs för varje parameter som ska plottas.
- set engine parameters
Här ändrar man de konstanter som ingår i motormodellen.
- set output parameters
Här väljer man vilken parameter som ska plottas. Detta måste göras innan man plottar.
- run
Kurvorna ritas upp. "klar" skrivs ut när den är färdig. Man återvänder till menyn med enter.
- save parameters
Alla konstanter sparas i en extern fil som heter motor2.dat.
- quit
Programmet stoppas och alla konstanter sparas, se ovan.

Figur 6.1 visar hur bildskärmen ser ut när programmet har ritat ut musseldiagram under en körning.

Figur 6.1 Engplot



6:1 FRAMTAGNING AV MOTORKARAKTERISTIKA MED HJÄLP AV ENGPLOT

Utgående från mätdata från den motor som ska styras, ritas musseldiagram upp. För varje diagram hålls Q_a , Q_f och alfa konstant. Sedan ska de konstanter som ingår i motormodellen bestämmas. Detta görs genom att köra engplott och prova sej fram med olika värden på konstanterna tills de plottade diagrammen approximativt stämmer överens. Då kan värdena på konstanterna flyttas över till engopt och motorn kan simuleras.

7 SLUTSATSER

Simuleringen visar att systemet fungerar bra. En fortsättning av projektet bör ske genom att prova på en verklig motor.

Ett system som detta borde ha framtiden för sig.

Implementeringen i mikroelektronik torde inte bjuda på några problem med dagens allt snabbare processorer.

Tack vare parameterstyrningen ger systemet ett snabbt stegsvar som ligger nära den optimala driftspunkten. Optimeringen finjusterar sedan hela tiden systemet.

Eftersom en kontinuerlig uppdatering av tabellen sker så kommer systemet att kunna korrigera sig för sådana förändringar som är svåra att mäta, till exempel förslitning av motorn eller variationer i bränslekvaliteten. Detta borde leda till att serviceintervallerna på motorn kan bli längre eftersom den blir okänslig för små feljusteringar. Systemet bör givetvis kompletteras med utskrift av felmeddelanden så att inte onormala driftsförhållanden uppträder utan att åtgärdas.

Vårt system bygger på att, till skillnad från dagens bilar, ej använda en lambdasond, utan att mäta upp de olika emissionshalterna var för sig. I dagens läge finns inga enkla och billiga sätt att mäta detta kontinuerligt på en bil. En förenkling av sättet att optimera emissionerna kan bli nödvändig i en kommersiellt anpassad version. Dock borde nya typer av sensorer, till exempel biosensorer, möjliggöra en allmän användning av ett sådant system som har beskrivits i denna rapport.

Ett sätt att kringgå detta problem kan vara att, genom laboratiemätningar på motorn, skapa en modell för emissionerna. Denna modell implementeras i styrsystemet och beräknar emissionerna som funktion av N , T , Q_f , Q_a och α . Dessa framräknade värden på emissionerna används sedan av styrsystemet på samma sätt som om de hade mätts upp.

En utveckling av programmet är att införa en möjlighet att under pågående simulering kunna ändra motorkonstanter. Detta för att simulera till exempel förslitning av motorn.

Systemet bör vara extra lämpligt för att styra stationära motorer. Eftersom dessa körs på ett konstant varvtal så undviks problemet med transienter.

Även andra typer av motorer respektive tillämpningar på motorer bör kunna styras på detta sätt. Vid en annan tillämpning kan en optimering för att få maximal effekt vid en minimal bränsleförbrukning vara aktuell.

REFERENSER

- [1] Åström, Wittenmark, *Adaptive Control*
Addison-Wesley publishing company (1989)

- [2] Ramsin, *Ickelinjär optimering*, LiberLäromedel
Lund (1976)

A1 BESKRIVNING AV ANVÄNDA BETECKNINGAR

Q_a = luftflöde m^3/s

Q_f = bränsleflöde m^3/s

α = tändförställning vevinkelgrader

B_{sfc} = Brake specific fuel consumption g/MJ

NO_x = Kväveoxider g/MJ

HC = Oförbrända kolväten g/MJ

CO = Koloxid g/MJ

CO_2 = Koldioxid g/MJ

N = Varvtalet på motorn $varv/s$

T = Momentet ut från motorn Nm

T_{ref} = Referensmomentet från trottelmодellen Nm

f_m = målfunktionen

A2 PROGRAMLISTNING AV ENGOPT

```
DECLARE SUB RESPONSE (I!, S!(), Tref!, T!, N!, Aign!, Qa!, Qf!, BSFC!, NOx!, HC!, CO!, C
DECLARE SUB THROTTLE (N!, Tref!, TIME!)
DECLARE SUB OPTIMIZE (R!(), S!(), B!, W!, P!, NOP!, SS!)
DECLARE SUB ENGINE (Tref!, T!, N!, Aign!, Qa!, Qf!, BSFC!, NOx!, HC!, CO!, CO2!)
```

```
COMMON SHARED XMAX, YMAX, XS, YS
COMMON SHARED NOP, INU, TIME, RFLAG, WFLAG, SS
COMMON SHARED T01, T02, T03, T04, T05, N01, N02, N03, N04, N05
COMMON SHARED A1, A2, A3, A4, A5
COMMON SHARED B1, B2, B3, B4, B5
COMMON SHARED C1, C2, C3, C4, C5
COMMON SHARED D1, D2, D3, D4, D5
COMMON SHARED E1, E2, F1, F3
COMMON SHARED WT, WBSFC, WNOx, WHC, WCO, WCO2
COMMON SHARED CT, CBSFC, CNOx, CHC, CCO, CCO2
COMMON SHARED ST, SBSFC, SNOx, SHC, SCO, SCO2, STIME
COMMON SHARED Nstart, Nstop, Nstep, Trefstart, Trefstop, Trefstep, Tstep, TIMEstep
COMMON SHARED R(), S(), C(), KF(), B, W, T, Tref, Trefold, N, Nold
COMMON SHARED Aign, Qa, Qf
COMMON SHARED PT$, PBSFC$, PNOx$, PHC$, PCO$, PCO2$, DISP$, LPAR$
```

```
NOP = 3 'number of control variables (Aign, Qa, Qf)
```

```
DIM R(NOP + 4, NOP) 'corner vector
DIM S(NOP + 2) 'response vector
DIM C(NOP)
```

```
DIM PT$(2)
DIM PBSFC$(2)
DIM PNOx$(2)
DIM PHC$(2)
DIM PCO$(2)
DIM PCO2$(2)
DIM DISP$(2)
```

```
DEFINT K
DIM KF(100, 50, 3) 'kennfeldt
```

```
WFLAG = 0
RFLAG = 0
INA = 1
```

```
GOSUB CLEARSCREEN
```

```
PRINT
PRINT " PROGRAM NAME: ENGOPT (ENGINE OPTimizer)"
PRINT
PRINT " VERSION NUMBER: 2.0"
PRINT
PRINT " VERSION DATE: 1989-09-11"
```

```
GOSUB WAITING
```

```
ON ERROR GOTO DEFAULT
```

```
GOSUB INITIALIZE
```

```
MAINMENU:
```

```
GOSUB CLEARSCREEN
COLOR 15
PRINT " MAIN MENU"
PRINT
```

```

PRINT " 1 SET PLOT PARAMETERS"
PRINT " 2 SET THROTTLE PARAMETERS"
PRINT " 3 SET ENGINE PARAMETERS"
PRINT " 4 SET WEIGHT FACTORS"
PRINT " 5 SAVE ALL PARAMETERS"
PRINT " 6 RECALL ALL PARAMETERS"
PRINT " 7 SAVE KENNFELDT"
PRINT " 8 RECALL KENNFELDT"
PRINT " 9 RUN ENGINE"
PRINT
INPUT " ", N$
IF N$ = "" THEN
  LOCATE 14, 1
  INPUT " Q TO QUIT ", N$
  IF N$ = "Q" THEN
    GOSUB SAVEINI
    STOP
  END IF
END IF

IF N$ = "1" OR N$ = "P" OR N$ = "p" THEN GOTO PLOTPAR
IF N$ = "2" OR N$ = "T" OR N$ = "t" THEN GOTO THROTTLEPAR
IF N$ = "3" OR N$ = "E" OR N$ = "e" THEN GOTO ENGINEPAR
IF N$ = "4" OR N$ = "W" OR N$ = "w" THEN GOTO WEIGHTFAC
IF N$ = "5" OR N$ = "S" OR N$ = "s" THEN GOSUB SAVEPAR
IF N$ = "6" OR N$ = "R" OR N$ = "r" THEN GOSUB RECALLPAR
IF N$ = "7" OR N$ = "A" OR N$ = "a" THEN GOSUB SAVEKENN
IF N$ = "8" OR N$ = "C" OR N$ = "c" THEN GOSUB RECALLKENN
IF N$ = "9" OR N$ = "U" OR N$ = "u" THEN GOTO RUNENGINE

```

GOTO MAINMENU

PLOTPAR:

GOSUB CLEARSCREEN

```

PRINT "      PLOT PARAMETERS"
PRINT
PRINT " 1 TORQUE           "; PT$
PRINT " 2 FUEL CONSUMPTION "; PBSFC$
PRINT " 3 NOx EMISSIONS    "; PNOx$
PRINT " 4 HC EMISSIONS     "; PHC$
PRINT " 5 CO EMISSIONS     "; PCO$
PRINT " 6 CO2 EMISSIONS    "; PCO2$
PRINT
PRINT " 7 DATA DISPLAY    "; DISP$
PRINT " 8 LINE PARAMETER   "; LPAR$
PRINT
PRINT " 9 SET SCALE FACTORS"
PRINT " 10 SET GRID PARAMETERS"
PRINT " 11 SET COLORS"

```

LOCATE 18, 3: INPUT " ", N\$

```

IF N$ = "1" AND PT$ = "ON" THEN
  PT$ = "OFF"
ELSEIF N$ = "1" THEN
  PT$ = "ON"
END IF
IF N$ = "2" AND PBSFC$ = "ON" THEN
  PBSFC$ = "OFF"
ELSEIF N$ = "2" THEN
  PBSFC$ = "ON"
END IF

```

```

IF N$ = "3" AND PNOx$ = "ON" THEN
  PNOx$ = "OFF"
ELSEIF N$ = "3" THEN
  PNOx$ = "ON"
END IF
IF N$ = "4" AND PHC$ = "ON" THEN
  PHC$ = "OFF"
ELSEIF N$ = "4" THEN
  PHC$ = "ON"
END IF
IF N$ = "5" AND PCO$ = "ON" THEN
  PCO$ = "OFF"
ELSEIF N$ = "5" THEN
  PCO$ = "ON"
END IF
IF N$ = "6" AND PCO2$ = "ON" THEN
  PCO2$ = "OFF"
ELSEIF N$ = "6" THEN
  PCO2$ = "ON"
END IF
IF N$ = "7" AND DISP$ = "ON" THEN
  DISP$ = "OFF"
ELSEIF N$ = "7" THEN
  DISP$ = "ON"
END IF
IF N$ = "8" AND LPAR$ = "ON" THEN
  LPAR$ = "OFF"
ELSEIF N$ = "8" THEN
  LPAR$ = "ON"
END IF

IF N$ = "9" THEN GOTO SCALEPAR
IF N$ = "10" THEN GOTO GRIDPAR
IF N$ = "11" THEN GOTO COLORPAR

IF N$ = "" THEN GOTO MAINMENU

```

```
GOTO PLOTPAR
```

```
SCALEPAR:
```

```
GOSUB CLEARSCREEN
```

```

PRINT "      SCALE PARAMETERS"
PRINT
PRINT " 1 ST      = "; USING "#####"; ST
PRINT " 2 SBSFC = "; USING "#####"; SBSFC
PRINT " 3 SNOx    = "; USING "#####"; SNOx
PRINT " 4 SHC     = "; USING "#####"; SHC
PRINT " 5 SCO     = "; USING "#####"; SCO
PRINT " 6 SCO2    = "; USING "#####"; SCO2
PRINT " 7 STIME   = "; USING "#####"; STIME
PRINT

```

```

LOCATE 12, 3; INPUT "", N$
LOCATE 12, 6

```

```

IF N$ = "" THEN GOTO PLOTPAR
IF N$ = "1" THEN INPUT "ST      = ", ST
IF N$ = "2" THEN INPUT "SBSFC = ", SBSFC
IF N$ = "3" THEN INPUT "SNOx  = ", SNOx
IF N$ = "4" THEN INPUT "SHC   = ", SHC
IF N$ = "5" THEN INPUT "SCO   = ", SCO
IF N$ = "6" THEN INPUT "SCO2  = ", SCO2

```

```
IF N$ = "7" THEN INPUT "STIME = ", STIME
```

```
GOTO SCALEPAR
```

```
GRIDPAR:
```

```
GOSUB CLEARSCREEN
```

```
PRINT "      GRID PARAMETERS"
```

```
PRINT
```

```
PRINT "  1  XMAX = "; USING "#####"; XMAX
```

```
PRINT "  2  YMAX = "; USING "#####"; YMAX
```

```
PRINT "  3  XS   = "; USING "#####"; XS
```

```
PRINT "  4  YS   = "; USING "#####"; YS
```

```
PRINT
```

```
LOCATE 9, 3: INPUT "", N$
```

```
LOCATE 9, 6
```

```
IF N$ = "" THEN GOTO PLOTPAR
```

```
IF N$ = "1" THEN INPUT "XMAX = ", XMAX
```

```
IF N$ = "2" THEN INPUT "YMAX = ", YMAX
```

```
IF N$ = "3" THEN INPUT "XS   = ", XS
```

```
IF N$ = "4" THEN INPUT "YS   = ", YS
```

```
GOTO GRIDPAR
```

```
COLORPAR:
```

```
GOSUB CLEARSCREEN
```

```
PRINT "      COLORS"
```

```
PRINT
```

```
PRINT "  1  CT     = "; USING "#####"; CT
```

```
PRINT "  2  CBSFC = "; USING "#####"; CBSFC
```

```
PRINT "  3  CNOx  = "; USING "#####"; CNOx
```

```
PRINT "  4  CHC   = "; USING "#####"; CHC
```

```
PRINT "  5  CCO   = "; USING "#####"; CCO
```

```
PRINT "  6  CCO2  = "; USING "#####"; CCO2
```

```
PRINT "  7  CGRID = "; USING "#####"; CGRID
```

```
PRINT
```

```
LOCATE 12, 3: INPUT "", N$
```

```
LOCATE 12, 6
```

```
IF N$ = "" THEN GOTO PLOTPAR
```

```
IF N$ = "1" THEN INPUT "CT     = ", CT
```

```
IF N$ = "2" THEN INPUT "CBSFC = ", CBSFC
```

```
IF N$ = "3" THEN INPUT "CNOx  = ", CNOx
```

```
IF N$ = "4" THEN INPUT "CHC   = ", CHC
```

```
IF N$ = "5" THEN INPUT "CCO   = ", CCO
```

```
IF N$ = "6" THEN INPUT "CCO2  = ", CCO2
```

```
IF N$ = "7" THEN INPUT "CGRID = ", CGRID
```

```
GOTO COLORPAR
```

```
THROTTLEPAR:
```

```
GOSUB CLEARSCREEN
```

```
PRINT "      THROTTLE PARAMETERS"
```

```
PRINT
PRINT " 1 Nstart      ="; USING "#####.##"; Nstart: LOCATE 4, 24: PRINT " r/s"
PRINT " 2 Nstop       ="; USING "#####.##"; Nstop: LOCATE 5, 24: PRINT " r/s"
PRINT " 3 Nstep        ="; USING "#####.##"; Nstep: LOCATE 6, 24: PRINT " r/s"
PRINT " 4 Trefstart    ="; USING "#####.##"; Trefstart: LOCATE 7, 24: PRINT " Nm"
PRINT " 5 Trefstop     ="; USING "#####.##"; Trefstop: LOCATE 8, 24: PRINT " Nm"
PRINT " 6 Trefstep     ="; USING "#####.##"; Trefstep: LOCATE 9, 24: PRINT " Nm"
PRINT " 7 TIMEstep     ="; USING "#####.##"; TIMEstep: LOCATE 10, 24: PRINT " s"
```

```
LOCATE 12, 3: INPUT "", N$
LOCATE 12, 6
```

```
IF N$ = "1" THEN INPUT "Nstart      = ", Nstart
IF N$ = "2" THEN INPUT "Nstop       = ", Nstop
IF N$ = "3" THEN INPUT "Nstep        = ", Nstep
IF N$ = "4" THEN INPUT "Trefstart    = ", Trefstart
IF N$ = "5" THEN INPUT "Trefstop     = ", Trefstop
IF N$ = "6" THEN INPUT "Trefstep     = ", Trefstep
IF N$ = "7" THEN INPUT "TIMEstep     = ", TIMEstep
IF N$ = "" THEN GOTO MAINMENU
GOTO THROTTLEPAR
```

ENGINEPAR:

GOSUB CLEARSCREEN

```
PRINT "          ENGINEPARAMETERS"
PRINT
PRINT " 1 BSFC"
PRINT " 2 NOx"
PRINT " 3 HC"
PRINT " 4 CO"
PRINT " 5 CO2"
```

```
LOCATE 10, 3: INPUT "", N$
```

```
IF N$ = "" THEN GOTO MAINMENU
IF N$ = "1" THEN GOTO BSFCPAR
IF N$ = "2" THEN GOTO NOxPAR
IF N$ = "3" THEN GOTO HCPAR
IF N$ = "4" THEN GOTO COPAR
IF N$ = "5" THEN GOTO CO2PAR
```

```
GOTO ENGINEPAR
```

WEIGHTFAC:

GOSUB CLEARSCREEN

```
PRINT "          WEIGHT FACTORS"
PRINT
PRINT " 1 WT              ="; USING "#####.##"; WT
PRINT " 2 WBSFC           ="; USING "#####.##"; WBSFC
PRINT " 3 WNOx            ="; USING "#####.##"; WNOx
PRINT " 4 WHC             ="; USING "#####.##"; WHC
PRINT " 5 WCO             ="; USING "#####.##"; WCO
PRINT " 6 WCO2           ="; USING "#####.##"; WCO2
PRINT
PRINT " 7 STEPSIZE        ="; USING "#####.##"; SS
PRINT
```

```
LOCATE 13, 3: INPUT "", N$
```

LOCATE 13, 6

```
IF N$ = "" THEN GOTO MAINMENU
IF N$ = "1" THEN INPUT "WT      = ", WT
IF N$ = "2" THEN INPUT "WBSFC   = ", WBSFC
IF N$ = "3" THEN INPUT "WNOx    = ", WNOx
IF N$ = "4" THEN INPUT "WHC     = ", WHC
IF N$ = "5" THEN INPUT "WCO     = ", WCO
IF N$ = "6" THEN INPUT "WCO2    = ", WCO2
IF N$ = "7" THEN INPUT "STEPSIZE = ", SS
```

GOTO WEIGHTFAC

RETURN

SAVEINI:

GOSUB CLEARSCREEN

FILENAME\$ = "PAR.INI"

LOCATE 4, 1

PRINT " STORING PARAMETERS IN FILE: "; FILENAME\$

OPEN "O", 1, FILENAME\$, 2048

```
PRINT #1, XMAX, YMAX, XS, YS
PRINT #1, A1, A2, A3, A4, A5
PRINT #1, B1, B2, B3, B4, B5
PRINT #1, C1, C2, C3, C4, C5
PRINT #1, D1, D2, D3, D4, D5
PRINT #1, E1, E2, F1, F2, F3
PRINT #1, T0, T01, T02, T03, T04, T05, N01, N02, N03, N04, N05
PRINT #1, NOP, INU, TIME, SS
PRINT #1, WT, WBSFC, WNOx, WHC, WCO, WCO2
PRINT #1, CT, CBSFC, CNOx, CHC, CCO, CCO2, CGRID
PRINT #1, ST, SBSFC, SNOx, SHC, SCO, SCO2, STIME
PRINT #1, Nstart, Nstop, Nstep, Trefstart, Trefstop, Trefstep, Tstep, TIMEstep
PRINT #1, B, W, T, Tref, Trefold, N, Nold
PRINT #1, Aign, Qa, Qf
WRITE #1, PT$, PBSFC$, PNOx$, PHC$, PCO$, PCO2$, DISP$, LPAR$
CLOSE
```

RETURN

SAVEPAR:

GOSUB CLEARSCREEN

INPUT " PARAMETER FILE NUMBER = ", PFN\$

IF PFN\$ = "" THEN RETURN

FILENAME\$ = "PAR" + PFN\$ + ".DAT"

LOCATE 4, 1

PRINT " STORING PARAMETERS IN FILE: "; FILENAME\$

OPEN "O", 1, FILENAME\$, 2048

PRINT #1, XMAX, YMAX, XS, YS

```
PRINT #1, A1, A2, A3, A4, A5
PRINT #1, B1, B2, B3, B4, B5
PRINT #1, C1, C2, C3, C4, C5
PRINT #1, D1, D2, D3, D4, D5
PRINT #1, E1, E2, F1, F2, F3
PRINT #1, T0, T01, T02, T03, T04, T05, N01, N02, N03, N04, N05
PRINT #1, NOP, INU, TIME, SS
PRINT #1, WT, WBSFC, WNOx, WHC, WCO, WCO2
PRINT #1, CT, CBSFC, CNOx, CHC, CCO, CCO2, CGRID
PRINT #1, ST, SBSFC, SNOx, SHC, SCO, SCO2, STIME
PRINT #1, Nstart, Nstop, Nstep, Trefstart, Trefstop, Trefstep, Tstep, TIMEstep
PRINT #1, B, W, T, Tref, Trefold, N, Nold
PRINT #1, Aign, Qa, Qf
WRITE #1, PT$, PBSFC$, PNOx$, PHC$, PCO$, PCO2$, DISP$, LPAR$
CLOSE
```

RETURN

RECALLPAR:

GOSUB CLEARSCREEN

```
INPUT "    PARAMETER FILE NUMBER = ", PFN$
IF PFN$ = "" THEN RETURN
FILENAME$ = "PAR" + PFN$ + ".DAT"
LOCATE 4, 1
PRINT "    LOADING PARAMETERS FROM FILE: "; FILENAME$
```

OPEN "I", 1, FILENAME\$, 2048

```
INPUT #1, XMAX, YMAX, XS, YS
INPUT #1, A1, A2, A3, A4, A5
INPUT #1, B1, B2, B3, B4, B5
INPUT #1, C1, C2, C3, C4, C5
INPUT #1, D1, D2, D3, D4, D5
INPUT #1, E1, E2, F1, F2, F3
INPUT #1, T0, T01, T02, T03, T04, T05, N01, N02, N03, N04, N05
INPUT #1, NOP, INU, TIME, SS
INPUT #1, WT, WBSFC, WNOx, WHC, WCO, WCO2
INPUT #1, CT, CBSFC, CNOx, CHC, CCO, CCO2, CGRID
INPUT #1, ST, SBSFC, SNOx, SHC, SCO, SCO2, STIME
INPUT #1, Nstart, Nstop, Nstep, Trefstart, Trefstop, Trefstep, Tstep, TIMEstep
INPUT #1, B, W, T, Tref, Trefold, N, Nold
INPUT #1, Aign, Qa, Qf
INPUT #1, PT$, PBSFC$, PNOx$, PHC$, PCO$, PCO2$, DISP$, LPAR$
CLOSE
```

RETURN

SAVEKENN:

GOSUB CLEARSCREEN

```
INPUT "    KENNFELDT NUMBER = ", KFN$
IF KFN$ = "" THEN RETURN
FILENAME$ = "KF" + KFN$ + ".DAT"
LOCATE 4, 1
PRINT "    STORING KENNFELDT IN FILE: "; FILENAME$
```

OPEN "B", 1, FILENAME\$
FOR NN = 0 TO 50


```
FOR TT = 0 TO 50
  FOR J = 1 TO 3
    PUT #1, , KF(NN, TT, J)
  NEXT
NEXT
NEXT
CLOSE
```

```
RETURN
```

```
RECALLKENN:
```

```
GOSUB CLEARSCREEN
```

```
INPUT "    KENNFELDT NUMBER = ", KFN$
IF KFN$ = "" THEN RETURN
FILENAME$ = "KF" + KFN$ + ".DAT"
LOCATE 4, 1
PRINT "    LOADING KENNFELDT FROM FILE: "; FILENAME$
```

```
OPEN "B", 1, FILENAME$
FOR NN = 0 TO 50
  FOR TT = 0 TO 50
    FOR J = 1 TO 3
      GET #1, , KF(NN, TT, J)
    NEXT
  NEXT
NEXT
CLOSE
```

```
RETURN
```

```
BSFCPAR:
```

```
GOSUB CLEARSCREEN
```

```
PRINT "    BSFC PARAMETERS"
PRINT
PRINT "  1  A1  = "; A1
PRINT "  2  B1  = "; B1
PRINT "  3  C1  = "; C1
PRINT "  4  D1  = "; D1
PRINT "  5  E1  = "; E1
PRINT "  6  E2  = "; E2
PRINT "  7  F1  = "; F1
PRINT "  8  F2  = "; F2
PRINT "  9  T01 = "; T01
PRINT " 10  N01 = "; N01
PRINT
```

```
LOCATE 15, 3: INPUT "", N$
LOCATE 15, 6
```

```
IF N$ = "" THEN GOTO ENGINEPAR
IF N$ = "1" THEN INPUT "A1 = ", A1
IF N$ = "2" THEN INPUT "B1 = ", B1
IF N$ = "3" THEN INPUT "C1 = ", C1
IF N$ = "4" THEN INPUT "D1 = ", D1
IF N$ = "5" THEN INPUT "E1 = ", E1
IF N$ = "6" THEN INPUT "E2 = ", E2
IF N$ = "7" THEN INPUT "F1 = ", F1
IF N$ = "8" THEN INPUT "F2 = ", F2
```

```
IF N$ = "9" THEN INPUT "T01 = ", T01
IF N$ = "10" THEN INPUT "N01 = ", N01
```

```
GOTO BSFCPAR
```

```
NOxPAR:
```

```
GOSUB CLEARSCREEN
```

```
PRINT "      NOx PARAMETERS"
PRINT
PRINT "  1  A2  = "; A2
PRINT "  2  B2  = "; B2
PRINT "  3  C2  = "; C2
PRINT "  4  D2  = "; D2
PRINT "  5  E1  = "; E1
PRINT "  6  E2  = "; E2
PRINT "  7  T02 = "; T02
PRINT "  8  N02 = "; N02
PRINT
```

```
LOCATE 13, 3: INPUT "", N$
LOCATE 13, 6
```

```
IF N$ = "" THEN GOTO ENGINEPAR
IF N$ = "1" THEN INPUT "A2 = ", A2
IF N$ = "2" THEN INPUT "B2 = ", B2
IF N$ = "3" THEN INPUT "C2 = ", C2
IF N$ = "4" THEN INPUT "D2 = ", D2
IF N$ = "5" THEN INPUT "E1 = ", E1
IF N$ = "6" THEN INPUT "E2 = ", E2
IF N$ = "7" THEN INPUT "T02 = ", T02
IF N$ = "8" THEN INPUT "N02 = ", N02
```

```
GOTO NOxPAR
```

```
HCPAR:
```

```
GOSUB CLEARSCREEN
```

```
PRINT "      HC PARAMETERS"
PRINT
PRINT "  1  A3  = "; A3
PRINT "  2  B3  = "; B3
PRINT "  3  C3  = "; C3
PRINT "  4  D3  = "; D3
PRINT "  5  E1  = "; E1
PRINT "  6  E2  = "; E2
PRINT "  7  T03 = "; T03
PRINT "  8  N03 = "; N03
PRINT
```

```
LOCATE 13, 3: INPUT "", N$
LOCATE 13, 6
```

```
IF N$ = "" THEN GOTO ENGINEPAR
IF N$ = "1" THEN INPUT "A3 = ", A3
IF N$ = "2" THEN INPUT "B3 = ", B3
IF N$ = "3" THEN INPUT "C3 = ", C3
IF N$ = "4" THEN INPUT "D3 = ", D3
IF N$ = "5" THEN INPUT "E1 = ", E1
IF N$ = "6" THEN INPUT "E2 = ", E2
```

```
IF N$ = "7" THEN INPUT "T03 = ", T03
IF N$ = "8" THEN INPUT "N03 = ", N03
```

```
GOTO HCPAR
```

```
COPAR:
```

```
GOSUB CLEARSCREEN
```

```
PRINT "      CO PARAMETERS"
PRINT
PRINT "  1  A4 = "; A4
PRINT "  2  B4 = "; B4
PRINT "  3  C4 = "; C4
PRINT "  4  D4 = "; D4
PRINT "  5  E1 = "; E1
PRINT "  6  E2 = "; E2
PRINT "  7  T04 = "; T04
PRINT "  8  N04 = "; N04
PRINT
```

```
LOCATE 13, 3: INPUT "", N$
LOCATE 13, 6
```

```
IF N$ = "" THEN GOTO ENGINEPAR
IF N$ = "1" THEN INPUT "A4 = ", A4
IF N$ = "2" THEN INPUT "B4 = ", B4
IF N$ = "3" THEN INPUT "C4 = ", C4
IF N$ = "4" THEN INPUT "D4 = ", D4
IF N$ = "5" THEN INPUT "E1 = ", E1
IF N$ = "6" THEN INPUT "E2 = ", E2
IF N$ = "7" THEN INPUT "T04 = ", T04
IF N$ = "8" THEN INPUT "N04 = ", N04
```

```
GOTO COPAR
```

```
CO2PAR:
```

```
GOSUB CLEARSCREEN
```

```
PRINT "      CO2 PARAMETERS"
PRINT
PRINT "  1  A5 = "; A5
PRINT "  2  B5 = "; B5
PRINT "  3  C5 = "; C5
PRINT "  4  D5 = "; D5
PRINT "  5  E1 = "; E1
PRINT "  6  E2 = "; E2
PRINT "  7  T05 = "; T05
PRINT "  8  N05 = "; N05
PRINT
```

```
LOCATE 13, 3: INPUT "", N$
LOCATE 13, 6
```

```
IF N$ = "" THEN GOTO ENGINEPAR
IF N$ = "1" THEN INPUT "A5 = ", A5
IF N$ = "2" THEN INPUT "B5 = ", B5
IF N$ = "3" THEN INPUT "C5 = ", C5
IF N$ = "4" THEN INPUT "D5 = ", D5
IF N$ = "5" THEN INPUT "E1 = ", E1
IF N$ = "6" THEN INPUT "E2 = ", E2
```

```
IF N$ = "7" THEN INPUT "T05 = ", T05
IF N$ = "8" THEN INPUT "N05 = ", N05
```

```
GOTO CO2PAR
```

```
-----
RUNENGINE:
```

```
TIME = 0
Tstep = 0
Nold = 0
N = Nstart
Tref = Trefstart
WFLAG = 0
```

```
GOSUB GRID
```

```
DO UNTIL NOT INKEY$ = ""
```

```
CALL THROTTLE(N, Tref, TIME)
```

```
IF TIME = 0 THEN GOSUB GRID
```

```
IF NOT INT(N + .5) = INT(Nold + .5) OR NOT INT(Tref + .5) = INT(Trefold + .5) THEN
```

```
IF WFLAG = 1 THEN GOSUB WRITEKENN
```

```
GOSUB READKENN
```

```
END IF
```

```
IF P = 4 OR RFLAG = 1 THEN
```

```
FOR I = 1 TO 4
```

```
CALL ENGINE(Tref, T, N, R(I, 1), R(I, 2), R(I, 3), BSFC, NOx, HC, CO, CO2)
```

```
CALL RESPONSE(I, S(), Tref, T, N, Aign, Qa, Qf, BSFC, NOx, HC, CO, CO2, WT, WB
```

```
NEXT
```

```
ELSE
```

```
I = 5
```

```
CALL ENGINE(Tref, T, N, R(I, 1), R(I, 2), R(I, 3), BSFC, NOx, HC, CO, CO2)
```

```
CALL RESPONSE(I, S(), Tref, T, N, R(I, 1), R(I, 2), R(I, 3), BSFC, NOx, HC, CO,
```

```
END IF
```

```
RFLAG = 0
```

```
CALL OPTIMIZE(R(), S(), B, W, P, NOP, SS)
```

```
Aign = R(I, 1)
```

```
Qa = R(I, 2)
```

```
Qf = R(I, 3)
```

```
GOSUB PLOT
```

```
GOSUB DISPLAY
```

```
Trefold = Tref
```

```
Nold = N
```

LOOP

GOSUB WRITEKENN

GOTO MAINMENU

WRITEKENN:

```
KF(INT(Nold + .5), INT(Trefold + .5), 1) = 200 * R(B, 1) 'Aign (ignition angle)
KF(INT(Nold + .5), INT(Trefold + .5), 2) = 200 * R(B, 2) 'Qa (air mass flow)
KF(INT(Nold + .5), INT(Trefold + .5), 3) = 5000 * R(B, 3) 'Qf (fuel mass flow)
```

RETURN

READKENN:

FOR I = 1 TO 4

```
R(I, 1) = .005 * KF(INT(N + .5), INT(Tref + .5), 1) 'Aign (ignition angle)
R(I, 2) = .005 * KF(INT(N + .5), INT(Tref + .5), 2) 'Qa (air mass flow)
R(I, 3) = .0002 * KF(INT(N + .5), INT(Tref + .5), 3) 'Qf (fuel mass flow)
```

```
IF R(I, 1) <= 0 THEN R(I, 1) = .005 * KF(INT(Nold + .5), INT(Trefold + .5), 1)
IF R(I, 2) <= 0 THEN R(I, 2) = .005 * KF(INT(Nold + .5), INT(Trefold + .5), 2)
IF R(I, 3) <= 0 THEN R(I, 3) = .0002 * KF(INT(Nold + .5), INT(Trefold + .5), 3)
```

```
IF R(I, 1) <= 0 THEN R(I, 1) = 20
IF R(I, 2) <= 0 THEN R(I, 2) = 3
IF R(I, 3) <= 0 THEN R(I, 3) = .2
```

NEXT

FOR J = 1 TO 3

```
R(J + 1, J) = (1 + SS / 500) * R(1, J)
```

NEXT

```
'create three new
'corners by using
'SS (step size)
```

P = 0

RFLAG = 1

WFLAG = 1

```
'set optimizing parameter
'set read flag
'set write flag
```

RETURN

DEFAULT:

IF ERR = 53 AND INA = 1 THEN

GOSUB CLEARSCREEN

LOCATE 2, 1

PRINT " COULD NOT FIND FILE: PAR.INI"

PRINT

PRINT " LOADING DEFAULT VALUES INSTEAD"

```
'set default values
'if no PAR.INI file
'is found
```

INA = 0

SS = 1

XMAX = 1000

YMAX = 50

XS = 10

YS = 10

PT\$ = "ON"

PBSFC\$ = "ON"
PNOx\$ = "ON"
PHC\$ = "ON"
PCO\$ = "ON"
PCO2\$ = "ON"
DISP\$ = "ON"
LPAR\$ = "ON"

CT = 10
CBSFC = 15
CNOx = 11
CHC = 13
CCO = 14
CCO2 = 9
CGRID = 4

WT = 10
WBSFC = 1
WNOx = 0
WHC = 0
WCO = 0
WCO2 = 0

ST = 5
SBSFC = 50
SNOx = 2
SHC = 2
SCO = 2
SCO2 = 100
STIME = 100

Nstart = 25
Nstop = 25
Nstep = 0

Trefstart = 5
Trefstop = 50
Trefstep = 5

TIMEstep = 100

T01 = 45
T02 = 45
T03 = 45
T04 = 45
T05 = 45

N01 = 30
N02 = 30
N03 = 30
N04 = 30
N05 = 30

A1 = 20
A2 = 3.2
A3 = 1.6
A4 = 6.4
A5 = 8

B1 = 1
B2 = 80
B3 = 80
B4 = 80
B5 = 18

C1 = 10000

```
C2 = 10
C3 = .2
C4 = 0
C5 = 0
```

```
D1 = 80
D2 = .1
D3 = 0
D4 = 0
D5 = 0
```

```
E1 = .1
E2 = 25
```

```
F1 = 1
F2 = 0
```

```
FOR I = 1 TO 200
LOCATE 2, 1
PRINT "    COULD NOT FIND FILE:  PAR.INI"
PRINT
PRINT "    LOADING DEFAULT VALUES INSTEAD"
NEXT
```

```
ELSEIF INA = 0 THEN STOP
END IF
```

```
GOTO MAINMENU
```

```
INITIALIZE:
```

```
GOSUB CLEARSCREEN
```

```
LOCATE 2, 1
PRINT "    INITIALIZING FROM FILE:  PAR.INI"
```

```
OPEN "I", 1, "PAR.INI", 2048
INPUT #1, XMAX, YMAX, XS, YS
INPUT #1, A1, A2, A3, A4, A5
INPUT #1, B1, B2, B3, B4, B5
INPUT #1, C1, C2, C3, C4, C5
INPUT #1, D1, D2, D3, D4, D5
INPUT #1, E1, E2, F1, F2, F3
INPUT #1, T0, T01, T02, T03, T04, T05, N01, N02, N03, N04, N05
INPUT #1, NOP, INU, TIME, SS
INPUT #1, WT, WBSFC, WNOx, WHC, WCO, WCO2
INPUT #1, CT, CBSFC, CNOx, CHC, CCO, CCO2, CGRID
INPUT #1, ST, SBSFC, SNOx, SHC, SCO, SCO2, STIME
INPUT #1, Nstart, Nstop, Nstep, Trefstart, Trefstop, Trefstep, Tstep, TIMEstep
INPUT #1, B, W, T, Tref, Trefold, N, Nold
INPUT #1, Aign, Qa, Qf
INPUT #1, PT$, PBSFC$, PNOx$, PHC$, PCO$, PCO2$, DISP$, LPAR$
CLOSE
```

```
FOR I = 1 TO 200
LOCATE 2, 1
PRINT "    INITIALIZING FROM FILE:  PAR.INI"
NEXT
```

```
RETURN
```

```
DISPLAY:
```

COLOR 15

LOCATE 2, 2: PRINT "Nref =" ; USING "###.##"; N: LOCATE 2, 19: PRINT " r/s"
LOCATE 3, 2: PRINT "Tref =" ; USING "###.##"; Tref: LOCATE 3, 19: PRINT " Nm"

IF DISP\$ = "OFF" THEN RETURN

LOCATE 5, 2: PRINT "SPEED =" ; USING "###.##"; N: LOCATE 5, 19: PRINT " r/s"
LOCATE 6, 2: PRINT "TORQUE =" ; USING "###.##"; T: LOCATE 6, 19: PRINT " Nm"

LOCATE 8, 2: PRINT "IGNITION =" ; USING "###.##"; Aign: LOCATE 8, 19: PRINT " deg"
LOCATE 9, 2: PRINT "AIR FLOW =" ; USING "###.##"; Qa: LOCATE 9, 19: PRINT " g/s"
LOCATE 10, 2: PRINT "FUEL FLOW =" ; USING "###.##"; Qf: LOCATE 10, 19: PRINT " g/s"

LOCATE 12, 2: PRINT "LAMBDA =" ; USING "###.##"; Qa / Qf / 15

LOCATE 2, 25: PRINT " MASS/ENERGY"
LOCATE 3, 25: PRINT " (g/MJ) "
LOCATE 4, 25: PRINT "BSFC:" ; USING "#####.##"; BSFC
LOCATE 5, 25: PRINT "NOx: " ; USING "#####.##"; NOx
LOCATE 6, 25: PRINT "HC: " ; USING "#####.##"; HC
LOCATE 7, 25: PRINT "CO: " ; USING "#####.##"; CO
LOCATE 8, 25: PRINT "CO2: " ; USING "#####.##"; CO2

LOCATE 2, 51: PRINT " COST"
LOCATE 3, 51: PRINT " (kr/MJ) "
LOCATE 4, 51: PRINT USING "#####.##"; .001 * WBSFC * BSFC
LOCATE 5, 51: PRINT USING "#####.##"; .001 * WNOx * NOx
LOCATE 6, 51: PRINT USING "#####.##"; .001 * WHC * HC
LOCATE 7, 51: PRINT USING "#####.##"; .001 * WCO * CO
LOCATE 8, 51: PRINT USING "#####.##"; .001 * WCO2 * CO2
LOCATE 10, 45: PRINT "TOTAL:"
LOCATE 10, 51: PRINT USING "#####.##"; .001 * (WBSFC * BSFC + WNOx * NOx + WHC * HC + WCO

RETURN

CLEARSCREEN:
CLS
PRINT
RETURN

WAITING:

DO WHILE INKEY\$ = ""
LOOP
RETURN

GRID:

SCREEN 9
COLOR CGRID
CLS

WINDOW (0, 0)-(XMAX, YMAX)

FOR I = 0 TO XMAX STEP XMAX / XS
LINE (I, 0)-(I, YMAX)
NEXT

FOR I = 0 TO YMAX STEP YMAX / YS
LINE (0, I)-(XMAX, I)

NEXT

COLOR 2

LINE (0, 0)-(XMAX, 0)
LINE (XMAX, 0)-(XMAX, YMAX)
LINE (XMAX, YMAX)-(0, YMAX)
LINE (0, YMAX)-(0, 0)

COLOR 15

IF DISP\$ = "ON" THEN
LOCATE 2, 40: PRINT " PRICE"
LOCATE 3, 40: PRINT " (kr/kg) "
LOCATE 4, 40: PRINT USING "#####.##"; WBSFC
LOCATE 5, 40: PRINT USING "#####.##"; WNOx
LOCATE 6, 40: PRINT USING "#####.##"; WHC
LOCATE 7, 40: PRINT USING "#####.##"; WCO
LOCATE 8, 40: PRINT USING "#####.##"; WCO2
END IF

LOCATE 2, 65: PRINT "SCALES " "
COLOR CT
LOCATE 3, 65: PRINT "T: "; USING "###.#"; ST: LOCATE 3, 75: PRINT " Nm "
COLOR CBSFC
LOCATE 4, 65: PRINT "BSFC:"; USING "###.#"; SBSFC: LOCATE 4, 75: PRINT " g/MJ"
COLOR CNOx
LOCATE 5, 65: PRINT "NOx: "; USING "###.#"; SNOx: LOCATE 5, 75: PRINT " g/MJ"
COLOR CHC
LOCATE 6, 65: PRINT "HC: "; USING "###.#"; SHC: LOCATE 6, 75: PRINT " g/MJ"
COLOR CCO
LOCATE 7, 65: PRINT "CO: "; USING "###.#"; SCO: LOCATE 7, 75: PRINT " g/MJ"
COLOR CCO2
LOCATE 8, 65: PRINT "CO2: "; USING "###.#"; SCO2: LOCATE 8, 75: PRINT " g/MJ"
COLOR CGRID
LOCATE 10, 65: PRINT "TIME:"; USING "#####"; STIME: LOCATE 10, 75: PRINT " s "

TIME = 0
Tstep = 0

RETURN

PLOT:

XTIME = TIME / STIME * XMAX / XS
YT = T / ST * YMAX / YS
YBSFC = BSFC / SBSFC * YMAX / YS
YNOx = NOx / SNOx * YMAX / YS
YHC = HC / SHC * YMAX / YS
YCO = CO / SCO * YMAX / YS
YCO2 = CO2 / SCO2 * YMAX / YS

IF TIME <= 1 THEN GOSUB SETPLOT

IF PT\$ = "ON" THEN
COLOR CT
PSET (XTIME, YT)
IF LPAR\$ = "ON" THEN LINE -(XOTIME, YOT)
END IF
IF PBSFC\$ = "ON" THEN
COLOR CBSFC
PSET (XTIME, YBSFC)
IF LPAR\$ = "ON" THEN LINE -(XOTIME, YOBSFC)
END IF

```

IF PNOx$ = "ON" THEN
COLOR CNOx
PSET (XTIME, YNOx)
IF LPAR$ = "ON" THEN LINE -(XOTIME, YONOx)
END IF
IF PHC$ = "ON" THEN
COLOR CHC
PSET (XTIME, YHC)
IF LPAR$ = "ON" THEN LINE -(XOTIME, YOHC)
END IF
IF PCO$ = "ON" THEN
COLOR CCO
PSET (XTIME, YCO)
IF LPAR$ = "ON" THEN LINE -(XOTIME, YOCO)
END IF
IF PCO2$ = "ON" THEN
COLOR CCO2
PSET (XTIME, YCO2)
IF LPAR$ = "ON" THEN LINE -(XOTIME, YOCO2)
END IF

```

```
GOSUB SETPLOT
```

```
RETURN
```

```
-----
SETPLOT:
```

```

XOTIME = XTIME
YOT = YT
YOBSTC = YBSTC
YONOx = YNOx
YOHC = YHC
YOCO = YCO
YOCO2 = YCO2

```

```
RETURN
```

```
DEFSNG K
```

```
SUB ENGINE (Tref, T, N, Aign, Qa, Qf, BSFC, NOx, HC, CO, CO2)
```

```
'input: Tref, N, Aign, Qa, Qf
```

```
'output: T, Bsf, NOx, HC, CO, CO2
```

```
IF Qa = 0 THEN Qa = .15
```

```
IF Qf = 0 THEN Qf = .01
```

```
FI = 15 * Qf / Qa
```

```
FID = F1 * (FI - .8) ^ 2 + 1
```

```
DD = E1 * (Aign - E2 * (1 + (20 * (FI - 1.1) ^ 2 + 1) * N / 100)) * (Aign - E2 * (1 + (2
```

```
E = 1
```

```
DO UNTIL E < .001
```

```
  J = J + 1
```

```
  Told = T
```

```
  BSFC = (A1 * (ABS(Tref - T01) / (Tref + B1) + (N - N01) ^ 2 / C1) + D1) * FID * DD
```

```
  T = D1 * T01 * N01 * Qf / BSFC / N
```

```
  E = ABS(Told - T)
```

```
  IF J > 10 THEN E = 0
```

```
LOOP
```

```

NOx = A2 * BSFC / B2 * ((FI - .9) ^ 2 + 1) / (C2 * (FI - .9) ^ 2 + 1) * (D2 * (Aign / C2
HC = A3 * BSFC / B3 * (1 + C3 * 15 * Qf / Qa)
CO = A4 * BSFC / B4 * 15 * Qf / Qa
CO2 = (A5 * 12) / (A5 * 12 + B5 * 1) * 44 / 12 * BSFC 'calculated from fuel carbon cont
END SUB

```

```

SUB OPTIMIZE (R(), S(), B, W, P, NOP, SS)

```

```

'R(): Parameter vector (Aign,Qa,Qf R(corner,parameter))
'S(): Response vector (input Bsf, NOx, HC, CO S(response))
'B: Best parameter combination (index 1..4) (best corner)
'W: Worst parameter combination (index 1..4) (worst corner)
'P: Calculation parameter
'NOP: Number of parameters (3 parameters: Aign, Qa, Qf)
'SS: Stepsize in percent

```

```

IF P = 0 THEN GOTO P0
IF P = 1 THEN GOTO P1
IF P = 2 THEN GOTO P2
IF P = 3 THEN GOTO P3

```

```

P0:
GOSUB RANKRES
GOSUB MEAN
GOSUB REFLECT
P = 1
GOTO SUBEXIT

```

```

P1:
IF S(NOP + 2) < S(W) THEN
  GOSUB SWITCH
  GOSUB EXPAND
  P = 2
  GOTO SUBEXIT

```

```

P2:
  IF S(NOP + 2) < S(W) THEN GOSUB SWITCH
  GOTO P0
END IF
GOSUB CONTRACT
P = 3
GOTO SUBEXIT

```

```

P3:
IF S(NOP + 2) < S(W) THEN
  GOSUB SWITCH
  GOTO P0
END IF
GOSUB SHRINK
P = 4
GOTO SUBEXIT

```

```

RANKRES:

```

```

'-----
B = 1
W = 1
FOR I = 2 TO NOP + 1
  IF S(I) < S(B) THEN B = I
  IF S(I) > S(W) THEN W = I
NEXT I
FOR J = 1 TO NOP

```

```
    R(NOP + 3, J) = R(W, J)
NEXT J
RETURN
```

MEAN:

```
' ----
FOR J = 1 TO NOP
  R(NOP + 4, J) = 0
  FOR I = 1 TO NOP + 1
    R(NOP + 4, J) = R(NOP + 4, J) + R(I, J)
  NEXT I
NEXT J
FOR J = 1 TO NOP
  R(NOP + 4, J) = R(NOP + 4, J) - R(NOP + 3, J)
  R(NOP + 4, J) = R(NOP + 4, J) / NOP
NEXT J
RETURN
```

REFLECT:

```
' -----
FOR J = 1 TO NOP
  R(NOP + 2, J) = R(NOP + 3, J) + 2 * (R(NOP + 4, J) - R(NOP + 3, J))
NEXT J
RETURN
```

EXPAND:

```
' -----
FOR J = 1 TO NOP
  IF ABS(R(NOP + 4, J) - R(NOP + 3, J)) > SS / 100 * ABS(R(NOP + 3, J)) THEN GOTO P0
NEXT
FOR J = 1 TO NOP
  R(NOP + 2, J) = R(NOP + 3, J) + 3 * (R(NOP + 4, J) - R(NOP + 3, J))
NEXT J
RETURN
```

CONTRACT:

```
' -----
FOR J = 1 TO NOP
  R(NOP + 2, J) = R(NOP + 3, J) + .5 * (R(NOP + 4, J) - R(NOP + 3, J))
NEXT J
RETURN
```

SHRINK:

```
' -----
FOR I = 1 TO NOP + 1
  FOR J = 1 TO NOP
    R(I, J) = .5 * (R(I, J) + R(B, J))
  NEXT J
NEXT I
RETURN
```

SWITCH:

```
' -----
S(W) = S(NOP + 2)
FOR J = 1 TO NOP
  R(W, J) = R(NOP + 2, J)
NEXT J
RETURN
```

SUBEXIT:

'-----

END SUB

SUB RESPONSE (I, S(), Tref, T, N, Aign, Qa, Qf, BSFC, NOx, HC, CO, CO2, WT, WBSFC, WNOx,
S(I) = WT * ABS(T - Tref) / (Tref + 1) * 100 + WBSFC * BSFC + WNOx * NOx + WHC * HC + WC

END SUB

SUB THROTTLE (N, Tref, TIME)

'output: N, Tref, TIME

TIME = TIME + 1

IF TIME > XMAX THEN TIME = 0

Tstep = Tstep + 1

IF Tstep >= TIMEstep THEN

 Tstep = 0

 Tref = Tref + Trefstep - .000001

 IF Tref > Trefstop THEN

 Tref = Trefstart

 N = N + Nstep - .000001

 IF N > Nstop THEN N = Nstart

 END IF

END IF

END SUB

A3 PROGRAMLISTNING AV ENGPLOT

```
80/11/20 11:23:42 SUB valjskala (Nmax!, Tmax!, xd!, yd!)
11:23:42 SUB valjindata (alfa!, Qa!, Qf!, engplot.bas NOx!(), HC!(), CO!(), antalkurvorB!, a
DECLARE SUB valjutdata (valdutdata!)
DECLARE SUB plotta (valdutdata!, startN!, Nmax!, Tmax!, xd!, yd!, deltaN!, alfa!, Qf!, Q
, N02!, N03!, N04!)
DECLARE SUB beraknaBsfc (alfa!, n!, t1!, t2!, xx!, Bsfc!, Qf!, Qa!, A1!, A2!, A3!, A4!,
DECLARE SUB beraknaNOx (alfa!, n!, t1!, t2!, xx!, NOx!, A1!, A2!, A3!, A4!, B1!, B2!, B3
DECLARE SUB beraknaHC (alfa!, n!, t1!, t2!, xx!, HC!, A1!, A2!, A3!, A4!, B1!, B2!, B3!,
DECLARE SUB beraknaCO (alfa!, n!, t1!, t2!, xx!, CO!, A1!, A2!, A3!, A4!, B1!, B2!, B3!,
DECLARE SUB motorkonstanter (A1!, A2!, A3!, A4!, B1!, B2!, B3!, B4!, C1!, C2!, C3!, C4!,
DECLARE SUB koordinatsystem (Nmax!, Tmax!, xd!, yd!)
DECLARE SUB plotta2 (Bsfc!(), NOx!(), HC!(), CO!(), valdutdata!, antalkurvorB!, antalkur
C1!, C2!, C3!, C4!, D1!, D2!, D3!, D4!, E1!, E2!, F1!, F2!, F3!, T01!, T02!, T03!, T04!,
ina = 1
antalkurvorB = 20
antalkurvorN = 20
antalkurvorH = 20
antalkurvorC = 20
DIM Bsfc(1 TO antalkurvorB)
DIM NOx(1 TO antalkurvorN)
DIM HC(1 TO antalkurvorH)
DIM CO(1 TO antalkurvorC)

ON ERROR GOTO default
GOSUB initialize
start:
startN = 0 'startvarvtal
deltaN = .5 'steglangnd vid berkning

meny:
SCREEN 0
WIDTH 40
COLOR 15, 9
CLS
PRINT " engplot "
PRINT " 1 set plot parameters "
PRINT " 2 set input parameters"
PRINT " 3 set engine parameters"
PRINT " 4 set output parameters"
PRINT " 5 run "
PRINT " 6 save parameters "
PRINT " 9 quit "
DO
  INPUT "main menu ", valdmeny
  SELECT CASE valdmeny
    CASE 1
      CALL valjskala(Nmax, Tmax, xd, yd)
    CASE 2
      CALL valjindata(alfa, Qa, Qf, Bsfc(), NOx(), HC(), CO(), antalkurvorB, antalkurvor
    CASE 3
      CALL motorkonstanter(A1, A2, A3, A4, B1, B2, B3, B4, C1, C2, C3, C4, D1, D2, D3, D
    CASE 4
      CALL valjutdata(valdutdata)
    CASE 5
      CALL plotta2(Bsfc(), NOx(), HC(), CO(), valdutdata, antalkurvorB, antalkurvorN, an
, E2, F1, F2, F3, T01, T02, T03, T04, N01, N02, N03, N04)
    CASE 6
      GOSUB save
    CASE 9
      PRINT " quit "
```

89/11/30 GOSUB save
11:23:42 STOP

engplot.bas

2

```
CASE ELSE  
END SELECT  
GOTO meny  
LOOP
```

```
default:  
IF ERR = 53 AND ina = 1 THEN  
  ina = 0  
ELSE  
  IF ina = 0 THEN STOP  
END IF  
GOTO start
```

```
initialize:  
OPEN "I", 1, "motor2.dat"  
INPUT #1, Nmax, Tmax, xd, yd  
INPUT #1, alfa, Qa, Qf  
INPUT #1, A1, A2, A3, A4, B1, B2, B3, B4, C1, C2, C3, C4, D1, D2, D3, D4, E1, E2, F1, F2  
INPUT #1, antalkurvorB, antalkurvorN, antalkurvorH, antalkurvorC  
FOR x = 1 TO antalkurvorB  
  INPUT #1, Bsfc(x)  
NEXT x  
FOR x = 1 TO antalkurvorN  
  INPUT #1, NOx(x)  
NEXT x  
FOR x = 1 TO antalkurvorH  
  INPUT #1, HC(x)  
NEXT x  
FOR x = 1 TO antalkurvorC  
  INPUT #1, CO(x)  
NEXT x  
CLOSE #1  
  
RETURN
```

```
save:  
CLOSE #1  
OPEN "o", 1, "motor2.dat"  
PRINT #1, Nmax, Tmax, xd, yd  
PRINT #1, alfa, Qa, Qf  
PRINT #1, A1, A2, A3, A4, B1, B2, B3, B4, C1, C2, C3, C4, D1, D2, D3, D4, E1, E2, F1, F2  
PRINT #1, antalkurvorB, antalkurvorN, antalkurvorH, antalkurvorC  
FOR x = 1 TO antalkurvorB  
  PRINT #1, Bsfc(x)  
NEXT x  
FOR x = 1 TO antalkurvorN  
  PRINT #1, NOx(x)  
NEXT x  
FOR x = 1 TO antalkurvorH  
  PRINT #1, HC(x)  
NEXT x  
FOR x = 1 TO antalkurvorC  
  PRINT #1, CO(x)  
NEXT x  
CLOSE #1
```

8/11/20
11:23:42

engplot.bas

3

```
SUB beraknaBsfc (alfa, n, t1, t2, xx, Bsfc, Qf, Qa, A1, A2, A3, A4, B1, B2, B3, B4, C1,
fi = F3 * Qf / Qa
fiberoende = F2 * (fi - .8) ^ 2 + F1
dd = (alfa - E1 * n) ^ 2 + E2
xx = A1 * ((Bsfc - D1) / (C1 * dd * fiberoende) - (n - N01) ^ 2 / B1)
IF xx < 0 THEN
  EXIT SUB
END IF
t1 = T01 + SQR(xx)
t2 = T01 - SQR(xx)
END SUB
SUB beraknaCO (alfa, n, t1, t2, xx, CO, A1, A2, A3, A4, B1, B2, B3, B4, C1, C2, C3, C4,
dd = (alfa - E1 * n) ^ 2 + E2
xx = A4 * ((CO - D4) / (C4 * dd) - (n - N04) ^ 2 / B4)
IF xx < 0 THEN
  EXIT SUB
END IF
t1 = T04 + SQR(xx)
t2 = T04 - SQR(xx)
END SUB
SUB beraknaHC (alfa, n, t1, t2, xx, HC, A1, A2, A3, A4, B1, B2, B3, B4, C1, C2, C3, C4,
dd = (alfa - E1 * n) ^ 2 + E2
xx = A3 * ((HC - D3) / (C3 * dd) - (n - N03) ^ 2 / B3)
IF xx < 0 THEN
  EXIT SUB
END IF
t1 = T03 + SQR(xx)
t2 = T03 - SQR(xx)
END SUB
SUB beraknaNOx (alfa, n, t1, t2, xx, NOx, A1, A2, A3, A4, B1, B2, B3, B4, C1, C2, C3, C4
dd = (alfa - E1 * n) ^ 2 + E2
```


89/11/30 * ((NOx - D2) / (C2 * dd) - (n - N02) ^ 2 / B2)
11:23:42

engplot.bas

4

```
IF xx < 0 THEN  
  EXIT SUB  
END IF
```

```
t1 = T02 + SQR(xx)
```

```
t2 = T02 - SQR(xx)
```

```
END SUB
```

```
SUB grid (xs, ys, xd, yd)
```

```
col = 4  
co2 = 2
```

```
SCREEN 9  
'COLOR col
```

```
FOR i = 0 TO xs STEP xs / xd  
  PSET (i, 0)  
  LINE -(i, ys)  
NEXT
```

```
FOR i = 0 TO ys STEP ys / yd  
  PSET (0, i)  
  LINE -(xs, i)  
NEXT
```

```
'COLOR co2
```

```
PSET (0, 0)  
LINE -(xs, 0)  
LINE -(xs, ys)  
LINE -(0, ys)  
LINE -(0, 0)
```

```
END SUB
```

```
SUB koordinatsystem (Nmax, Tmax, xd, yd)
```

```
SCREEN 9  
COLOR 4
```

```
WINDOW (0, 0)-(Nmax, Tmax)
```

```
LINE (0, 0)-(Nmax, 0)
```

```
LINE (0, 0)-(0, Tmax)
```

```
FOR i = 0 TO Nmax STEP Nmax / xd
```

```
  LINE (i, 0)-(i, Tmax)
```

```
NEXT
```

```
FOR i = 0 TO Tmax STEP Tmax / yd
```

```
  LINE (0, i)-(Nmax, i)
```

```
NEXT
```

```
COLOR 15
```

```
END SUB
```

```
SUB motorkonstanter (A1, A2, A3, A4, B1, B2, B3, B4, C1, C2, C3, C4, D1, D2, D3, D4, E1,
```

```
MENY0:
```

```
PRINT " set data "  
PRINT  
PRINT " 1 Bsfc "  
PRINT " 2 NOx "  
PRINT " 3 HC "  
PRINT " 4 CO "  
  
PRINT  
INPUT n$  
  
IF n$ = "" THEN GOTO EXIT0  
IF n$ = "1" THEN GOTO MENY1  
IF n$ = "2" THEN GOTO MENY2  
IF n$ = "3" THEN GOTO MENY3  
IF n$ = "4" THEN GOTO MENY4  
GOTO MENY0  
  
MENY1:  
'SCREEN 0  
'WIDTH 40  
'COLOR 15, 9  
CLS  
PRINT "Bsfc"  
PRINT  
PRINT " 1 a1 = "; A1  
PRINT " 2 b1 = "; B1  
PRINT " 3 c1 = "; C1  
PRINT " 4 d1 = "; D1  
PRINT " 5 e1 = "; E1  
PRINT " 6 e2 = "; E2  
PRINT " 7 f1 = "; F1  
PRINT " 8 f2 = "; F2  
PRINT " 9 f3 = "; F3  
PRINT "10 T01= "; T01  
PRINT "11 N01= "; N01  
  
PRINT  
INPUT n$  
  
IF n$ = "" THEN GOTO MENY0  
IF n$ = "1" THEN INPUT " a1 = "; A1  
IF n$ = "2" THEN INPUT " b1 = "; B1  
IF n$ = "3" THEN INPUT " c1 = "; C1  
IF n$ = "4" THEN INPUT " d1 = "; D1  
IF n$ = "5" THEN INPUT " e1 = "; E1  
IF n$ = "6" THEN INPUT " e2 = "; E2  
IF n$ = "7" THEN INPUT " f1 = "; F1  
IF n$ = "8" THEN INPUT " f2 = "; F2  
IF n$ = "9" THEN INPUT " f3 = "; F3  
IF n$ = "10" THEN INPUT " T01 = "; T01  
IF n$ = "11" THEN INPUT " N01 = "; N01  
  
GOTO MENY1  
  
MENY2:  
'SCREEN 0
```

```
CLS
PRINT "NOx"
PRINT
PRINT " 1 a2 = "; A2
PRINT " 2 b2 = "; B2
PRINT " 3 c2 = "; C2
PRINT " 4 d2 = "; D2
PRINT " 5 e1 = "; E1
PRINT " 6 e2 = "; E2
PRINT " 7 T02= "; T02
PRINT " 8 N02= "; N02

PRINT
INPUT n$

IF n$ = "" THEN GOTO MENY0
IF n$ = "1" THEN INPUT " a2 = "; A2
IF n$ = "2" THEN INPUT " b2 = "; B2
IF n$ = "3" THEN INPUT " c2 = "; C2
IF n$ = "4" THEN INPUT " d2 = "; D2
IF n$ = "5" THEN INPUT " e1 = "; E1
IF n$ = "6" THEN INPUT " e2 = "; E2
IF n$ = "7" THEN INPUT " T02 = "; T02
IF n$ = "8" THEN INPUT " N02 = "; N02

GOTO MENY2

MENY3:
'SCREEN 0
'WIDTH 40
'COLOR 15, 9
CLS
PRINT "HC"
PRINT
PRINT " 1 a3 = "; A3
PRINT " 2 b3 = "; B3
PRINT " 3 c3 = "; C3
PRINT " 4 d3 = "; D3
PRINT " 5 e1 = "; E1
PRINT " 6 e2 = "; E2
PRINT " 7 T03= "; T03
PRINT " 8 N03= "; N03

PRINT
INPUT n$

IF n$ = "" THEN GOTO MENY0
IF n$ = "1" THEN INPUT " a3 = "; A3
IF n$ = "2" THEN INPUT " b3 = "; B3
IF n$ = "3" THEN INPUT " c3 = "; C3
IF n$ = "4" THEN INPUT " d3 = "; D3
IF n$ = "5" THEN INPUT " e1 = "; E1
IF n$ = "6" THEN INPUT " e2 = "; E2
IF n$ = "7" THEN INPUT " T03 = "; T03
IF n$ = "8" THEN INPUT " N03 = "; N03

GOTO MENY3
```

89/11/30
11:23:42

engplot.bas

7

```
MENY4:
'SCREEN 0
'WIDTH 40
'COLOR 15, 9
CLS
PRINT "CO"
PRINT
PRINT " 1 a4 = "; A4
PRINT " 2 b4 = "; B4
PRINT " 3 c4 = "; C4
PRINT " 4 d4 = "; D4
PRINT " 5 e1 = "; E1
PRINT " 6 e2 = "; E2
PRINT " 7 T04= "; T04
PRINT " 8 N04= "; N04

PRINT
INPUT n$

IF n$ = "" THEN GOTO MENYO
IF n$ = "1" THEN INPUT " a4 = "; A4
IF n$ = "2" THEN INPUT " b4 = "; B4
IF n$ = "3" THEN INPUT " c4 = "; C4
IF n$ = "4" THEN INPUT " d4 = "; D4
IF n$ = "5" THEN INPUT " e1 = "; E1
IF n$ = "6" THEN INPUT " e2 = "; E2
IF n$ = "7" THEN INPUT " T04 = "; T04
IF n$ = "8" THEN INPUT " N04 = "; N04

GOTO MENY4

EXIT0:
END SUB

SUB plotta (valdutdata, startN, Nmax, Tmax, xd, yd, deltaN, alfa, Qf, Qa, Bsfc, NOx, CO,
n = startN
flag = 1
DO UNTIL n >= Nmax
  nold = n
  n = n + deltaN
  nnew = n
  SELECT CASE valdutdata
    CASE 1
      CALL beraknaBsfc(alfa, n, t1, t2, xx, Bsfc, Qf, Qa, A1, A2, A3, A4, B1, B2, B3, B4
    CASE 2
      CALL beraknaNOx(alfa, n, t1, t2, xx, NOx, A1, A2, A3, A4, B1, B2, B3, B4, C1, C2,
    CASE 3
      CALL beraknaHC(alfa, n, t1, t2, xx, HC, A1, A2, A3, A4, B1, B2, B3, B4, C1, C2, C3
    CASE 4
      CALL beraknaCO(alfa, n, t1, t2, xx, CO, A1, A2, A3, A4, B1, B2, B3, B4, C1, C2, C3
    CASE ELSE
      n = Nmax
  END SELECT
  IF xx > 0 AND flag = 1 THEN
    t1old = t1
    t1new = t1
    t2old = t2
```

89/11/30
11:23:42

engplot.bas

8

```
t2new = t2
flag = 0
LINE (nold, t1old)-(nnew, t1new)
LINE (nold, t2old)-(nnew, t2new)
END IF
IF xx > 0 THEN
  t1old = t1new
  t1new = t1
  t2old = t2new
  t2new = t2
  LINE (nold, t1old)-(nnew, t1new)
  LINE (nold, t2old)-(nnew, t2new)
END IF
LOOP

END SUB

SUB plotta2 (Bsfc(), NOx(), HC(), CO(), valdutdata, antalkurvorB, antalkurvorN, antalkur
F1, F2, F3, T01, T02, T03, T04, N01, N02, N03, N04)

CLS
CALL koordinatsystem(Nmax, Tmax, xd, yd)
'CALL grid(Nmax, Tmax, xd, yd)

x = 0
SELECT CASE valdutdata
CASE 1
  DO UNTIL x >= antalkurvorB
    x = x + 1
    Bsfc = Bsfc(x)
    CALL plotta(valdutdata, startN, Nmax, Tmax, xd, yd, deltaN, alfa, Qf, Qa, Bsfc, NOx)
    PRINT "Bsfc="; Bsfc
  LOOP
CASE 2
  DO UNTIL x >= antalkurvorN
    x = x + 1
    NOx = NOx(x)
    CALL plotta(valdutdata, startN, Nmax, Tmax, xd, yd, deltaN, alfa, Qf, Qa, Bsfc, NOx)
    PRINT "NOx="; NOx
  LOOP
CASE 3
  DO UNTIL x >= antalkurvorH
    x = x + 1
    HC = HC(x)
    CALL plotta(valdutdata, startN, Nmax, Tmax, xd, yd, deltaN, alfa, Qf, Qa, Bsfc, NOx)
    PRINT "HC="; HC
  LOOP
CASE 4
  DO UNTIL x >= antalkurvorC
    x = x + 1
    CO = CO(x)
    CALL plotta(valdutdata, startN, Nmax, Tmax, xd, yd, deltaN, alfa, Qf, Qa, Bsfc, NOx)
    PRINT "CO="; CO
  LOOP
CASE ELSE
  PRINT
END SELECT
PRINT "klar"

DO WHILE INKEY$ = ""
```

89/11/30
11:23:42

engplot.bas

9

END SUB

```
SUB valjindata (alfa, Qa, Qf, Bsfc(), NOx(), HC(), CO(), antalkurvorb, antalkurvorn, ant
PRINT " ign.angel (alfa)="; alfa, "airflow (Qa)="; Qa, "fuelflow (Qf)="; Qf
PRINT " change input data? y / n "
INPUT n$
IF n$ = "y" THEN
  INPUT " ign.angel (alfa)="; alfa
  INPUT "airflow (Qa)="; Qa
  INPUT "fuelflow (Qf)="; Qf
END IF
MENY00:
CLS
PRINT

PRINT " set input data "
PRINT
PRINT " 1 Bsfc "
PRINT " 2 NOx "
PRINT " 3 HC "
PRINT " 4 CO "

PRINT
INPUT n$

IF n$ = "" THEN GOTO exit00
IF n$ = "1" THEN GOTO MENY10
IF n$ = "2" THEN GOTO MENY20
IF n$ = "3" THEN GOTO MENY30
IF n$ = "4" THEN GOTO MENY40
GOTO MENY00

MENY10:
CLS
x = 0
DO UNTIL x = antalkurvorb
  x = x + 1
  PRINT "Bsfc="; Bsfc(x)
LOOP
PRINT "continue? y/n"
INPUT n$
IF n$ = "" OR n$ = "n" THEN GOTO exit00

PRINT " set number of curves "
INPUT antalkurvorb
PRINT "set data"
x = 0
DO UNTIL x = antalkurvorb
  x = x + 1
  INPUT "Bsfc="; Bsfc(x)
LOOP
GOTO exit00

MENY20:
CLS
x = 0
DO UNTIL x = antalkurvorn
  x = x + 1
```

89/11/20
11:23:42

engplot.bas

10

```
PRINT "NOx="; NOx(x)
PRINT "continue? y/n"
INPUT n$
IF n$ = "" OR n$ = "n" THEN GOTO exit00
```

```
PRINT " set number of curves "
INPUT antalkurvorN
PRINT "set data"
x = 0
DO UNTIL x = antalkurvorN
  x = x + 1
  INPUT "NOx="; NOx(x)
LOOP
GOTO exit00
```

```
MENY30:
CLS
x = 0
DO UNTIL x = antalkurvorH
  x = x + 1
  PRINT "HC="; HC(x)
LOOP
PRINT "continue? y/n"
INPUT n$
IF n$ = "" OR n$ = "n" THEN GOTO exit00
```

```
PRINT " set number of curves "
INPUT antalkurvorH
PRINT "set data"
x = 0
DO UNTIL x = antalkurvorH
  x = x + 1
  INPUT "HC="; HC(x)
LOOP
GOTO exit00
```

```
MENY40:
CLS
x = 0
DO UNTIL x = antalkurvorC
  x = x + 1
  PRINT "CO="; CO(x)
LOOP
PRINT "continue? y/n"
INPUT n$
IF n$ = "" OR n$ = "n" THEN GOTO exit00
```

```
PRINT " set number of curves "
INPUT antalkurvorC
PRINT "set data"
x = 0
DO UNTIL x = antalkurvorC
  x = x + 1
  INPUT "CO="; CO(x)
LOOP
GOTO exit00
```

```
SUB valjskala (Nmax, Tmax, xd, yd)
PRINT " Nmax="; Nmax, "Tmax="; Tmax
PRINT " number of steps in x axis ="; xd, "number of steps in y axis ="; yd
PRINT " set scale? y / n "
INPUT n$
IF n$ = "y" THEN
  INPUT " Nmax=", Nmax
  INPUT " Tmax=", Tmax
  INPUT " number of steps in x axis =", xd
  INPUT " number of steps in y axis =", yd
END IF
END SUB

SUB valjutdata (valdutdata)

PRINT " set output data "
PRINT " 1 Bsfc "
PRINT " 2 NOx "
PRINT " 3 HC "
PRINT " 4 CO "
INPUT valdutdata

END SUB
```