CODEN: LUTFD2/(TFRT-5408)/1–36/(1989)

Användande av à priori-kunskap i adaptiva regulatorer

Jonas Eneberg

Institutionen for Reglerteknik
Lunds Tekniska Högskola
September 1989

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name
Master Thesis

Date of issue
September 1989

Document Number
CODEN: LUTFD2/(TFRT-5408)/1-36/(1989)

Author(s)
Jonas Eneberg

Supervisor
Björn Wittenmark

Sponsoring organisation

Title and subtitle
Användande av a priori-kunskap i adaptiva regulator. (A priori knowledge in adaptive controllers.)

Abstract

A priori knowledge about the process is often available when designing an adaptive controller. This report describes a method to use this knowledge.

The first step is to design a nominal controller. In the following steps the nominal controller is modified adaptively. The principal advantage of this method over conventional methods is that the calculations are minimized.

In this report the method has been investigated through two examples. The adaptation has been carried out using approximations of increasing complexibility. Simulations show that for reasonable deviations of the process parameters from the nominal values, considerable improvement in performance is achieved with the adaptive controllers compared to the nominal controller. However, the nominal controller has a larger stability region.

The primary conclusion of the report is that a priori knowledge should always be used in the design of adaptive controllers.

Key words
A priori knowledge, adaptive control

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
Swedish

Number of pages
36

Recipient's notes

Security classification

Förord

Denna rapport behandlar ett examensarbete, som blivit utfört under sommaren 1989 vid institutionen för Reglerteknik på Lunds Tekniska Högskola, under handledning av professor Björn Wittenmark. Examensarbetet, som omfattar 15 poäng, är gjort som avslutning på studier vid Elektrotekniklinjen på LTH.

Arbetet har skett genom simuleringar på institutionens dator Bode, med hjälp av dels simuleringsprogrammet SIMNON och dels matrisräkningsprogrammet MATLAB.

Jag tackar också härmed professor Björn Wittenmark för trevligt samarbete och ovärderlig hjälp.

Lund augusti 1989

J.E.

Innehållsförteckning

1. Inledning

Det här kapitlet presenterar idén bakom examensarbetet och uppläggningen av rapporten.

Reglering

I en vanlig regulator utgår man från en modell av systemet, där man antar sig känna systemet perfekt. Utgående från denna modell dimensionerar man sedan en regulator för att få ett önskat uppförande hos det slutna systemet.

Adaptiv reglering

Många gånger känner man inte sitt system så perfekt. Man kan då använda sig av en adaptiv regulator. Det traditionella sättet att göra detta på är att anse alla systemets parametrar okända och sedan uppskatta dessa parametrar (identifiera systemet) vid exempelvis varje sampeltidpunkt. Ett vanligt sätt att göra detta på är genom rekursiv minsta-kvadrat-skattning. Baserat på skattningarna dimensionerar man en regulator, dvs man uppdaterar sin regulator succesivt.

A priori-kunskap

Gör man på ovan beskrivna sätt, tar man dock inte hänsyn till den kunskap man eventuellt har om systemet à priori (i förväg). Vad man istället kan göra är att sätta upp en modell av systemet, där man anser vissa parametrar kända och andra okända (osäkra). Sedan dimensionerar man en regulator utgående från de kända parametrarna och antagna, nominella värden på de okända, en *nominell regulator*. Nästa steg är att uppskatta avvikelserna från de nominella värdena och modifiera den nominella regulatorn med hjälp av dessa skattningar. På så sätt får man (förhoppningsvis) färre parametrar att identifiera, vilket avsevärt kan förenkla beräkningsarbetet samt förbättra överskådligheten och eventuellt robustheten hos systemet.

I detta examensarbete studeras dimensionering av en adaptiv regulator, när man har à priori-kunskap om processen som skall regleras. Då adaptiva regulatorer i de flesta fall görs på tidsdiskret form, görs även i denna rapport alla beräkningar på tidsdiskret form. Huvuddelen av arbetet ägnas åt att undersöka två exempel med hjälp av SIMNON och MATLAB, för att se hur metoden fungerar i praktiken.

I kapitel 2 ges den allmänna teorin, medan kapitel 3 och 4 redovisar de två exemplen. Kapitel 5 ger en sammanfattning med de kommentarer och slutsatser, som arbetet givit anledning till.

2. Design av adaptiv regulator med användande av à priori-kunskap

Detta kapitel beskriver arbetsgången i det allmänna fallet. Vi skall i tur och ordning titta på process, nominell regulator, identifiering och beräkning av regulatorparametrar.

2.1 Process

I de följande två kapitlen kommer vi att behandla två exempel, där vi vill reglera en process med hjälp av den i föregående kapitel beskrivna metoden. Vi vill skriva processens överföringsfunktion på formen

$$\begin{aligned} H(q) &= \frac{B_0(q) + \Delta B(q)}{A_0(q) + \Delta A(q)} \\ &= \frac{b_1 q^{n-1} + b_2 q^{n-2} + \ldots + b_n + \sum \beta_i B_i(q)}{a_0 q^n + a_1 q^{n-1} + \ldots + a_n + \sum \alpha_i A_i(q)}, \end{aligned}$$

där A_0 och B_0 representerar den nominella processen och ΔA och ΔB bestående av $A_i(q)$ och $B_i(q)$, som är kända funktioner i skiftoperatorn q, och α_i och β_i, som är okända parametrar, representerar avvikelserna från den nominella processen.

2.2 Nominell regulator

Vid designen av en RST-regulator till det nominell systemet

$$H_0(q) = \frac{B_0(q)}{A_0(q)}$$

användes här den metod för polplacering med polynomdesign, som beskrivs på sidan 231 i Åström-Wittenmark (1984). Enligt denna väljes först den önskade överföringsfunktionen

$$H_m(q) = \frac{B_m(q)}{A_m(q)}$$

Det är dessutom nödvändigt att välja ett observerarpolynom $A_O(q)$. Detta väljes lämpligen så att dynamiken hos observeraren är något snabbare än hos det önskade systemet.

Man löser sedan en sk diofantisk ekvation och får på så sätt fram regulatorpolynomen $R_0(q)$, $S_0(q)$ och $T_0(q)$ i regulatorn

$$R_0(q)u(k) = T_0(q)u_c(k) - S_0(q)y(k), \tag{2.1}$$

där $u_c(k)$ är referenssignalen.

2.3 Identifiering

Nästa steg på vägen till vår kompletta regulator är att identifiera (skatta) den osäkra delen av processen $H(q)$. Detta skall vi göra med minsta-kvadrat-metoden, som beskrivs utförligt i Åström-Wittenmark (1984). Vi kan skriva

$$[A_0(q) + \Delta A(q)]y(k) = [B_0(q) + \Delta B(q)]u(k),$$

vilket kan skrivas om som

$$A_0(q)y(k) - B_0(q)u(k) = -\Delta A(q)y(k) + \Delta B(q)u(k).$$

Vi inför de nya funktionerna

$$\begin{aligned} y_i(k) &= A_i(q)y(k), \\ u_i(k) &= B_i(q)u(k), \end{aligned}$$

och kan då skriva

$$y_0(k) - u_0(k) = -\sum \alpha_i y_i(k) + \sum \beta_i u_i(k) = \phi(k)\Theta,$$

där

$$\begin{aligned} \phi(k) &= \begin{pmatrix} -y_1(k) & -y_2(k) & \dots & u_1(k) & u_2(k) & \dots \end{pmatrix} \text{ och} \\ \Theta^T &= \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \beta_1 & \beta_2 & \dots \end{pmatrix}. \end{aligned}$$

Enligt minsta-kvadrat-metoden bestämmes då skattningen $\hat{\Theta}(k)$ av ekvationerna

$$\begin{aligned} \hat{\Theta}(k+1) &= \hat{\Theta}(k) + K(k)[z(k+1) - \phi(k+1)\hat{\Theta}(k)] \\ K(k) &= P(k)\phi^T(k+1)[\lambda + \phi(k+1)P(k)\phi^T(k+1)]^{-1} \\ P(k+1) &= [I - K(k)\phi(k+1)]P(k)/\lambda, \end{aligned} \tag{2.2}$$

där λ är en sk glömske-faktor, och $z(k) = y_0(k) - u_0(k)$.

2.4 Beräkning av regulatorparametrar

De skattade parametrarna skall nu användas för att uppdatera regulatorn. Inför nya parametrar enligt

$$\begin{aligned} R(q) &= R_0(q) + \Delta R(q) \\ S(q) &= S_0(q) + \Delta S(q) \\ T(q) &= K_m T_0(q), \end{aligned}$$

där $grad(\Delta R) = grad(R) - 1$ och $grad(\Delta S) = grad(S)$. (Eftersom $R(q)$ skall vara monisk blir alltid koefficienten framför högsta potensen i $R(q)$ lika med ett.) K_m väljes så att den stationära förstärkningen blir ett, dvs $H_m(1) = 1$.

Vi får följande designekvation:

$$(A_0 + \Delta A)(R_0 + \Delta R) + (B_0 + \Delta B)(S_0 + \Delta S) = A_m(q)A_O(q)$$

Vi skriver om ovanstående uttryck och får

$$\begin{aligned}(A_0 + \Delta A)\Delta R + (B_0 + \Delta B)\Delta S &= A_m A_O - (A_0 + \Delta A)R_0 - (B_0 + \Delta B)S_0 \\ &= -\Delta A R_0 - \Delta B S_0\end{aligned}$$

eftersom $A_0 R_0 + B_0 S_0 = A_m A_O$, om vi har samma specifikation på det slutna systemet som när vi designade den nominella regulatorn. Denna ekvation kan skrivas på matrisform enligt

$$(\mathbf{S} + \Delta\mathbf{S}) \begin{pmatrix} \Delta R \\ \Delta S \end{pmatrix} = l$$

där $\mathbf{S}$ är konstant, $\Delta\mathbf{S}$ och l beror på α_i och β_i, samt $\begin{pmatrix} \Delta R \\ \Delta S \end{pmatrix}$ innehåller de sökta parametrarna. Härav följer att om man kan välja ΔR och ΔS enligt

$$\begin{pmatrix} \Delta R \\ \Delta S \end{pmatrix} = (\mathbf{S} + \Delta\mathbf{S})^{-1} l$$

så får man den önskade karakteristiska ekvationen. Serieutveckling av $(\mathbf{S} + \Delta\mathbf{S})^{-1}$ ger

$$(\mathbf{S} + \Delta\mathbf{S})^{-1} = (\mathbf{I} + \mathbf{S}^{-1}\Delta\mathbf{S})^{-1} \approx (\mathbf{I} - \mathbf{S}^{-1}\Delta\mathbf{S} + (\mathbf{S}^{-1}\Delta\mathbf{S})^2 - \ldots)\mathbf{S}^{-1}.$$

Det är lämpligt att använda sig av serieutvecklingen vid implementering av regulatorn i ett SIMNON-program, eftersom man då minskar ner betydligt på beräkningsarbetet. Man inverterar $\mathbf{S}$-matrisen en gång för alla och får sedan bara några matrismultiplikationer att utföra vid varje sampeltidpunkt, jämfört med att invertera $(\mathbf{S} + \Delta\mathbf{S})$. Genom att använda tillräckligt många termer i serieutvecklingen kan man komma godtyckligt nära de exakta värdena.

3. Exempel 1

I detta kapitel skall vi närmare studera ett exempel. Sektion 1 går igenom design-beräkningarna enligt samma mönster som i kapitel 2, och i sektion 2 ser vi på resultatet av simuleringarna.

3.1 Design

Process

Processen vi skall undersöka beskrivs av överföringsfunktionen

$$H(s) = \frac{K}{s(1+Ts)},$$

där parametrarna K och T är osäkra och har de nominella värdena $K = 1$ och $T = 1$.

Sampling av den nominella processen med samplingsintervallet $h = 0.5$ s ger (val av samplingsintervall diskuteras i nästa avsnitt)

$$H_0(q) = \frac{0.107q + 0.09}{q^2 + 1.607q + 0.607} = \frac{b_{10}q + b_{20}}{(q-1)(q-a_{20})},$$

där b_{10}, b_{20} och a_{20} beror på K och T. Här har vi alltså tre parametrar, som är osäkra. Jämför detta med att vi har totalt fyra parametrar i den samplade överföringsfunktionen, vilka vi hade fått skatta om vi gjort en "vanlig" adaptiv regulator. Vi kan skriva den verkliga processen som

$$H(q) = \frac{b_{10}q + b_{20} + \beta_1 q + \beta_2}{(q-1)(q-a_{20}) + \alpha_1(q-1)} = \frac{B_0(q) + \Delta B(q)}{A_0(q) + \Delta A(q)},$$

dvs att vi har

$$\Delta A(q) = \alpha_1(q-1),$$
$$\Delta B(q) = \beta_1 q + \beta_2.$$

Vid implementeringen av processen i SIMNON, måste vi först skriva den på tillståndsform. På observerbar form ser det ut så här:

$$x(k+1) = \begin{pmatrix} -a_1 & 1 \\ -a_2 & 0 \end{pmatrix} x(k) + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} u(k)$$
$$y(k) = \begin{pmatrix} 1 & 0 \end{pmatrix} x(k),$$

där

$$a_1 = a_{10} + \alpha_1 = -1.607 + \alpha_1$$
$$a_2 = a_{20} - \alpha_1 = 0.607 - \alpha_1$$
$$b_1 = b_{10} + \beta_1 = 0.107 + \beta_1$$
$$b_2 = b_{20} + \beta_2 = 0.09 + \beta_2$$

En listning av SIMNON-programmet cproc finns i Appendix 1.

Design av nominell regulator

Den önskade karakteristiska ekvationen uttryckt i kontinuerliga termer har valts till

$$A_c(s) = s^2 + 2\zeta\omega s + \omega^2,$$

med dämpningen $\zeta = 0.7$ och egenfrekvensen $\omega = 1$. Med utgångspunkt från denna specifikation kan vi välja samplingsintervall. Detta diskuteras i Åström-Wittenmark (1984). Jag har här valt $h = 0.5$ s. Med detta värde får vi den diskreta specifikationen

$$A_m(q) = q^2 - 1.32q + 0.497.$$

Om vi väljer att behålla det öppna systemets nollställe, får vi dessutom

$$B_m(q) = K_m(0.107q + 0.09),$$

där K_m väljes så att $H_m(1) = B_m(1)/A_m(1) = 1$, dvs den stationära förstärkningen är ett. Återstår nu att välja ett observerarpolynom. Detta skall vara av första ordningen, och vi väljer det till $A_O(q) = q - 0.5$, vilket motsvarar en egenfrekvens på ungefär 1.4. Med dessa specifikationer ställer vi upp den diofantiska ekvationen

$$A(q)R(q) + B(q)S(q) = A_O(q)A_m(q).$$

Denna löses med hjälp av MATLAB och ger

$$\begin{aligned} R_0(q) &= q - 0.323 = q + r_{10} \\ S_0(q) &= 1.028q - 0.583 = s_{00}q + s_{10} \\ T_0(q) &= K_m(q - 0.5) = t_{00}q + t_{10} \end{aligned}$$

För att implementera regulatorn i SIMNON, måste vi återigen ställa upp en tillståndsbeskrivning. Ekvation (2.1) ger

$$u = -\frac{S_0}{R_0}y + \frac{T_0}{R_0}u_c = -(s_{00} + \frac{s_{10} - s_{00}r_{10}}{q + r_{10}})y + (t_{00} + \frac{t_{10} - t_{00}r_{10}}{q + r_{10}})u_c.$$

En observerbar realisation av regulatorn blir

$$\begin{aligned} x_s(k+1) &= -r_{10}x_s(k) + (s_{00}r_{10} - s_{10})y(k) \\ x_t(k+1) &= -r_{10}x_t(k) - (t_{00}r_{10} - t_{10})u_c(k) \\ u(k) &= x_s(k) + x_t(k) - s_{00}y(k) + t_{00}u_c(k) \end{aligned}$$

SIMNON-programmet `reg` implementerar regulatorn (se Appendix 1).

Identifiering

Vi skall nu försöka skatta parametrarna α_1, β_1 och β_2 med hjälp av minstakvadratmetoden. Enligt ovan gäller

$$y(k)[q^2 - 1.607q + 0.607 + \alpha_1(q - 1)] = u(k)[0.107q + 0.09 + \beta_1 q + \beta_2].$$

Vi förskjuter denna ekvation två sampelintervall och får

$$\begin{aligned} &y(k)[1 - 1.607q^{-1} + 0.607q^{-2} + \alpha_1(q^{-1} - q^{-2})] = \\ =&u(k)[0.107q^{-1} + 0.09q^{-2} + \beta_1 q^{-1} + \beta_2 q^{-2}]. \end{aligned}$$

Inför

$$
\begin{aligned}
y_0(k) &= (1 - 1.607q^{-1} + 0.607q^{-2})y(k),\\
u_0(k) &= (0.107q^{-1} + 0.09q^{-2})u(k),\\
y_1(k) &= (q^{-1} - q^{-2})y(k),\\
u_1(k) &= q^{-1}u(k),\\
u_2(k) &= q^{-2}u(k).
\end{aligned}
$$

Då kan vi skriva

$$y_0 - u_0 = -\alpha_1 y_1 + \beta_1 u_1 + \beta_2 u_2 = \phi(k)\Theta,$$

där

$$\phi(k) = \begin{pmatrix} -y_1(k) & u_1(k) & u_2(k) \end{pmatrix} \textit{ och}$$

$$\Theta = \begin{pmatrix} \alpha_1 \\ \beta_1 \\ \beta_2 \end{pmatrix}$$

Ekvationerna (2.2) finns implementerade i SIMNON-programmet `est`, som är listat i Appendix 1.

Beräkning av regulatorparametrar

I det här exemplet får vi

$$
\begin{aligned}
\Delta R(q) &= \delta r_1\\
\Delta S(q) &= \delta s_0 q + \delta s_1
\end{aligned}
$$

Vi följer räkningarna från sektion 2.4, vilket ger

$$\mathbf{S} = \begin{pmatrix} 1 & 0.107 & 0 \\ -1.607 & 0.09 & 0.107 \\ 0.607 & 0 & 0.09 \end{pmatrix}$$

$$\mathbf{\Delta S} = \begin{pmatrix} 0 & \beta_1 & 0 \\ \alpha_1 & \beta_2 & \beta_1 \\ -\alpha_1 & 0 & \beta_2 \end{pmatrix}$$

$$\begin{pmatrix} \Delta R \\ \Delta S \end{pmatrix} = \begin{pmatrix} \delta r_1 \\ \delta s_0 \\ \delta s_1 \end{pmatrix}$$

$$l = \begin{pmatrix} -\alpha_1 - 1.028\beta_1 \\ 1.323\alpha_1 + 0.583\beta_1 - 1.028\beta_2 \\ -0.323\alpha_1 + 0.583\beta_2 \end{pmatrix}$$

Första ordningens approximation

Om vi endast tar med en term i serieutvecklingen av $(\mathbf{S}+\mathbf{\Delta S})^{-1}$, får vi följande designekvation:

$$\begin{pmatrix} \delta r_1 \\ \delta s_0 \\ \delta s_1 \end{pmatrix} = \mathbf{S}^{-1} l = \begin{pmatrix} 0.2654 & -0.3155 & 0.3751 \\ 6.8654 & 2.9484 & -3.5053 \\ -1.7897 & 2.1277 & 8.5815 \end{pmatrix} l.$$

Denna finns implementerad i SIMNON-programmet `appr1`, som är listat i Appendix 1.

Andra ordningens approximation

Med två termer i serieutvecklingen blir designekvationen

$$\begin{pmatrix} \delta r_1 \\ \delta s_0 \\ \delta s_1 \end{pmatrix} = (\mathbf{I} - \mathbf{S}^{-1}\Delta\mathbf{S})\mathbf{S}^{-1}l,$$

vilken finns implementerad i SIMNON-programmet appr2.

Tredje ordningens approximation

Designekvationen blir

$$\begin{pmatrix} \delta r_1 \\ \delta s_0 \\ \delta s_1 \end{pmatrix} = [\mathbf{I} - \mathbf{S}^{-1}\Delta\mathbf{S} + (\mathbf{S}^{-1}\Delta\mathbf{S})^2]\mathbf{S}^{-1}l.$$

Se appr3 i Appendix 1.

3.2 Simulering

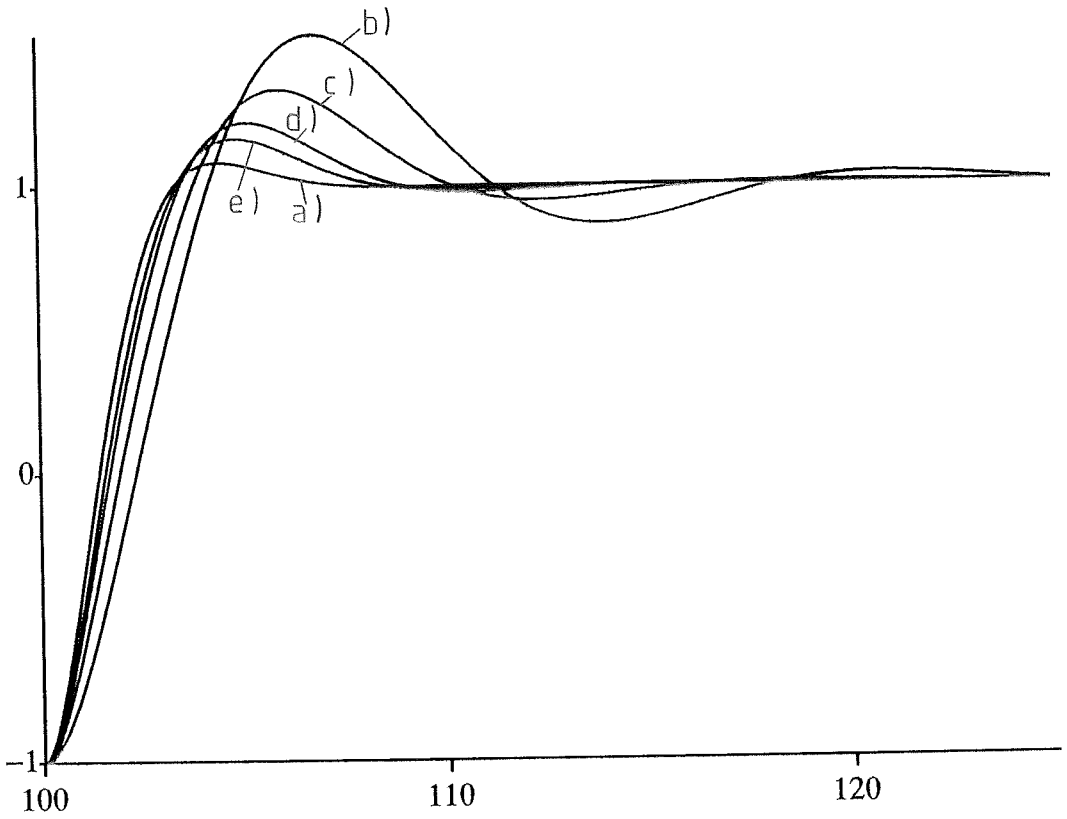
Förutsättningar

Vi skall nu undersöka vad som händer när värdena på K och T avviker från de nominella och jämföra de olika approximationerna med det önskade svaret. Som referenssignal användes en fyrkantvåg, som finns implementerad i SIMNON-programmet square, se Appendix 1. Denna har amplituden 1 och perioden 100 s, vilket befanns vara lämpliga värden för att få identifieringen att fungera tillfredsställande. Identifieraren ställer in sig mycket snabbt (på ett fåtal samplingsintervall), och vi studerar utsignalen efter en period av insignalen, vilket innebär att det är den stationära regulatorn vi ser på.

Resultat

Låt oss börja med att se vad som händer när värdet på T är större än det nominella. Detta innebär ju att polen i $s = -1$ flyttas närmare origo; eller motsvarande, att den diskreta polen i $z = 0.607$ flyttas ut mot enhetscirkeln och alltså närmar sig stabilitetsgränsen. Ett exempel på detta kan studeras i figur 3.1, där vi ser hur regleringen fungerar då $T = 3$ och $K = 1$. Polen i $s = -1$ är alltså flyttad till $s = -1/3$. Man ser här tydligt hur svaret närmar sig det önskade mer och mer ju bättre approximation som används. Med $T = 2$ (polen i $s = -0.5$) och $K = 2$ uppför sig systemet på liknande sätt, se figur 3.2. Här sammanfaller i det närmaste 3:e ordningens approximation med det önskade svaret. Vi låter nu polen flytta sig till vänster på negativa, reella axeln, dvs vi låter T bli mindre än 1. Fallet $T = 0.5$, $K = 1$ kan betraktas i figur 3.3, och vi ser att resultatet är liknande de tidigare fallen, dvs svaret närmar sig det önskade mer och mer ju bättre approximation vi använder.

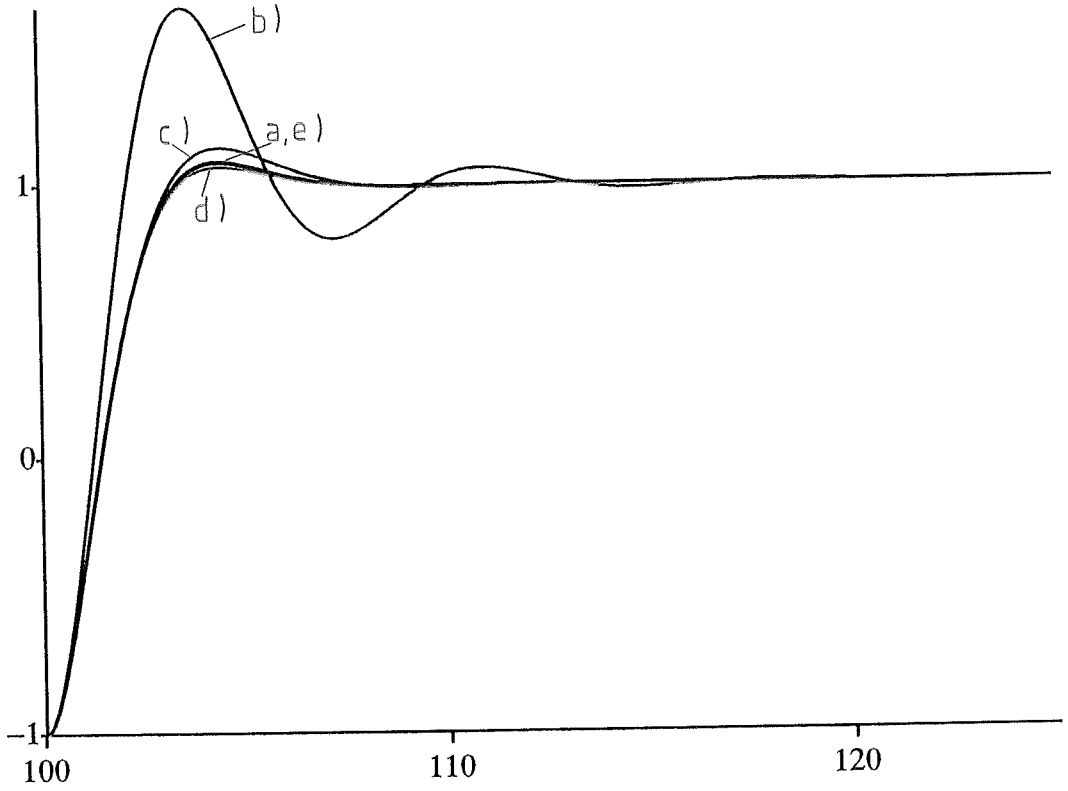
Som nämndes tidigare fungerar identifieringen mycket bra. Detta innebär att de olika approximativa designalgoritmerna har i det närmaste exakta värden på parametrarna α_1, β_1 och β_2 att utgå ifrån. Hur bra är då approximationerna? Ja, som exempel kan vi se hur nära de kommer i de ovan nämnda fallen. Tabell 3.1 visar detta. Som väntat närmar sig värdena de exakta mer och mer ju fler termer vi tar med i serieutvecklingen av $(\mathbf{S} + \Delta\mathbf{S})^{-1}$.



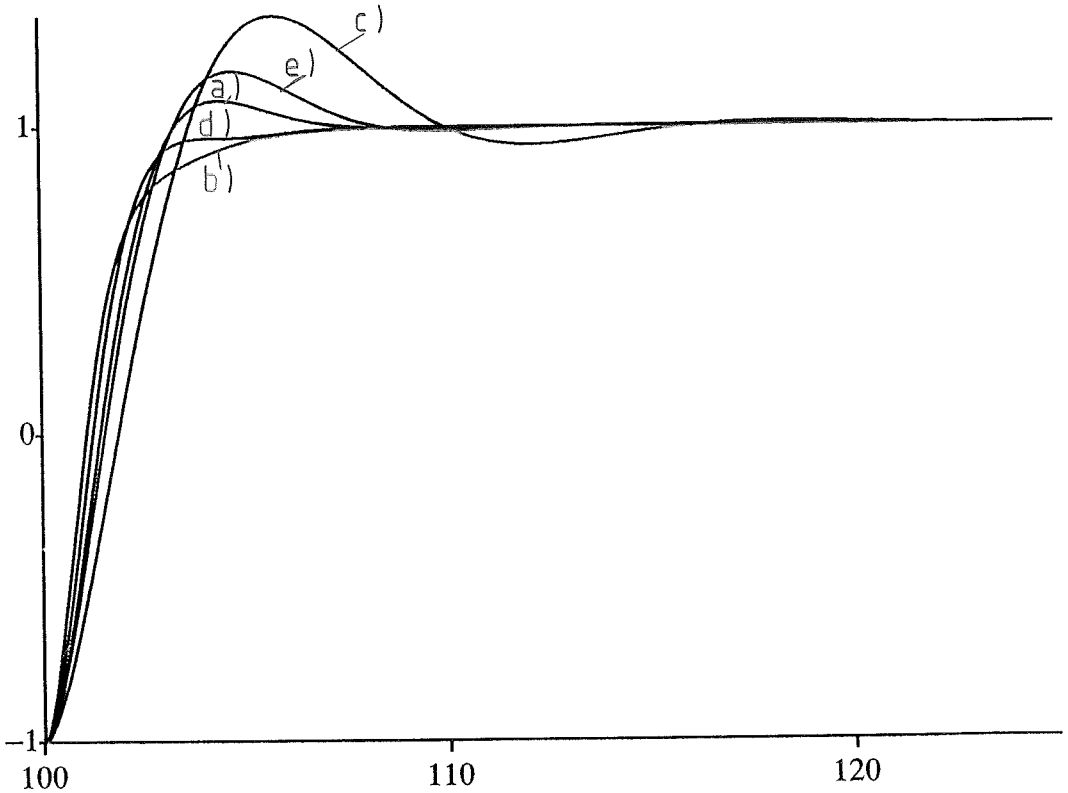
Figur 3.1 Utsignal från processen, då $T = 3$ och $K = 1$ (polen i $s = -1$ är flyttad till $s = -1/3$). a) önskat svar, b) nominell regulator, c) 1:a ordningens approximation, d) 2:a ordningens approximation och e) 3:e ordningens approximation.

Stabilitet

Det slutna systemet blir instabilt när karakteristiska ekvationen $AR + BS$ har något nollställe utanför enhetscirkeln. Ju större avvikelser vi har från de nominella värdena på T och K, desto mer fel blir det i approximationen, och desto mer avviker naturligtvis regulatorns parametrar från de önskade. Vi uppnår till slut en stabilitetsgräns. Stabilitetsgränserna i de olika fallen kan studeras i figur 3.4. Intressant är att notera dels att den nominella regulatorn är stabil för betydligt högre värden på K vid låga värden på T än de adaptiva regulatorerna, och dels att andra ordningens approximation har ett större stabilitetsområde än tredje ordningens. Det senare beror förmodligen på serieutvecklingens karaktär, med omväxlande positiva och negativa termer. Jämför också med kurvorna i figur 3.3. Det kan också noteras att stabilitetsgränserna för första och tredje ordningens approximationer sammanfaller för värden på T, som är större än ungefär 0.4. Vid användande av tredje ordningens approximation inträffar också ett annat fenomen när T blir tillräckligt liten, nämligen att regulatorn i sig själv blir instabil, medan det slutna systemet är stabilt. Med andra ord har polynomet $R(q)$ sitt nollställe utanför enhetscirkeln, men $AR + BS$ har sina nollställen innanför enhetscirkeln. Detta medför att tillståndet i regulatorn växer, tills datorn inte klarar av högre tal. Innan detta inträffar blir dock systemet instabilt av andra numeriska orsaker.



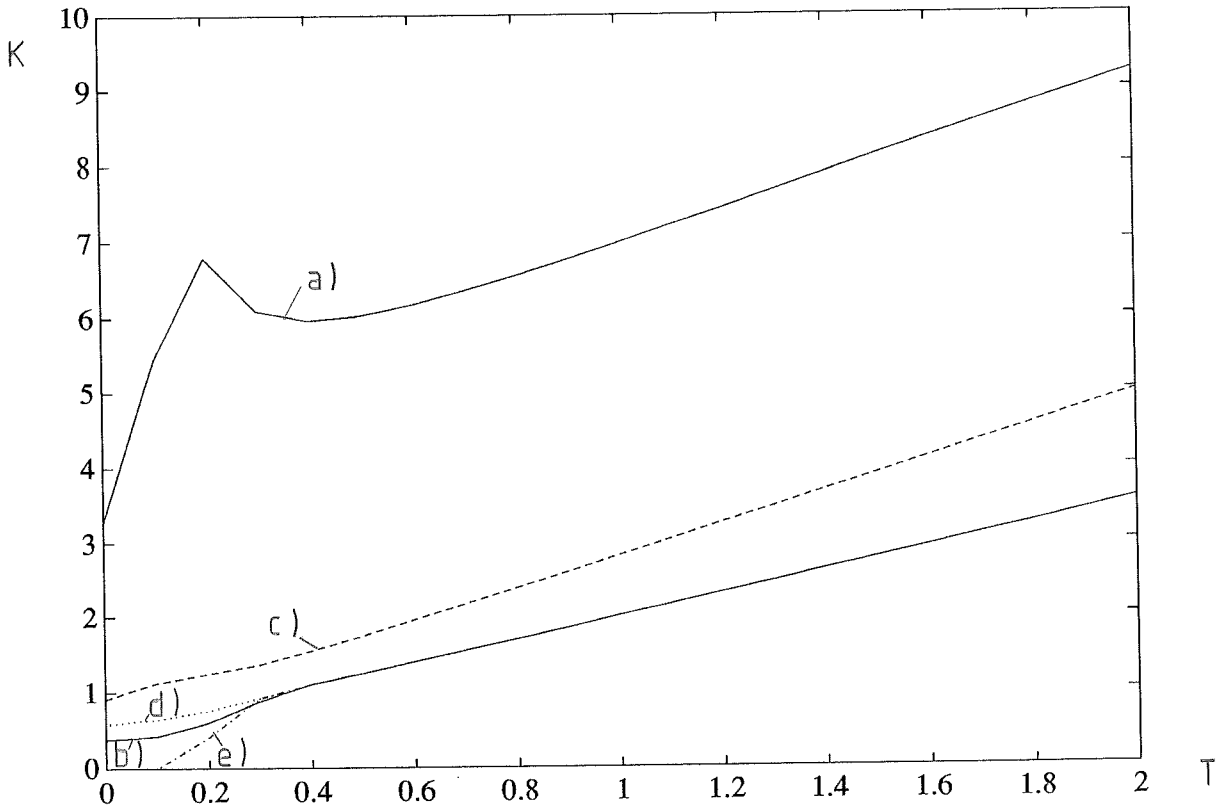
Figur 3.2 Utsignal i de olika fallen då $T = 2$ och $K = 2$ (polen i $s = -0.5$).



Figur 3.3 Utsignal i de olika fallen då $T = 0.5$ och $K = 1$ (polen i $s = -2$).

T	K	Appr1	Appr2	Appr3	Exakt
3	1	0.1947	0.1651	0.1725	0.1734
		1.0665	2.0158	2.5466	3.3940
		-0.7950	-1.5786	-2.0082	-2.7009
2	2	0.1430	0.1274	0.1301	0.1296
		0.1897	0.3308	0.2860	0.2922
		-0.2444	-0.3687	-0.3348	-0.3413
0.5	1	-0.2043	-0.2236	-0.2402	-0.2382
		-1.0615	-0.1177	-0.6411	-0.4339
		0.7928	0.0112	0.4367	0.2661

Tabell 3.1 Uträknade värden på dr_1, ds_0 respektive ds_1 för de olika approximationerna i de genomgångna fallen.



Figur 3.4 Stabilitetsgränser för det slutna systemet i de olika fallen i exempel 1. Systemet är stabilt för värden på T och K som ligger under gränserna. Gränserna gäller a) nominell regulator, b) 1:a ordningens approximation, c) 2:a ordningens approximation, d) 3:e ordningens approximation utan hänsyn tagen till regulatorns stabilitet och e) 3:e ordningens approximation med hänsyn tagen till regulatorns stabilitet.

Sammanfattning

Av ovanstående resultat kan vi dra den slutsatsen, att metoden fungerar mycket bra i det här exemplet, om avvikelserna från de nominella värdena i processens överföringsfunktion inte är alltför stora. Man får avsevärda förbättringar redan med första ordningens approximation, och använder man tre termer i serieutvecklingen blir regleringen mycket nära optimal för många värden på parametrarna T och K.

4. Exempel 2

I detta kapitel studerar vi ytterligare ett exempel enligt samma mönster som tidigare.

4.1 Design

Process

Vi skall nu undersöka en process med överföringsfunktionen

$$H(s) = \frac{K}{s(1+s)(1+Ts)},$$

med de nominella värdena $K = 1$ och $T = 0.1$.

Vi samplar processen med $h = 0.5$ s, vilket ger

$$H(q) = \frac{0.0738q^2 + 0.1158q + 0.0058 + \beta_1 q^2 + \beta_2 q + \beta_3}{q^3 - 1.6133q^2 + 0.6174q - 0.0041 + \alpha_1(q^2 - 1.607q + 0.607)}.$$

Vi har här som synes fyra osäkra parametrar och har alltså "sparat in" två stycken, vilket avsevärt förenklar beräkningsarbetet. Med stöd från kapitel 2 kan vi nu införa

$$\Delta A(q) = \alpha_1(q^2 - 1.607q + 0.607),$$
$$\Delta B(q) = \beta_1 q^2 + \beta_2 q + \beta_3.$$

För att kunna skriva ett SIMNON-program för processen inför vi också tillstånd enligt

$$x(k+1) = \begin{pmatrix} -a_1 & 1 & 0 \\ -a_2 & 0 & 1 \\ -a_3 & 0 & 0 \end{pmatrix} x(k) + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} u(k)$$
$$y(k) = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} x(k),$$

där

$$a_1 = -1.6133 + \alpha_1,$$
$$a_2 = 0.6174 - 1.607\alpha_1,$$
$$a_3 = -0.0041 + 0.607\alpha_1,$$
$$b_1 = 0.0738 + \beta_1,$$
$$b_2 = 0.1158 + \beta_2,$$
$$b_3 = 0.0058 + \beta_3.$$

Programmet `proc2` finns listat i Appendix 1.

Design av nominell regulator

Vi väljer här den önskade karakteristiska ekvationen till

$$A_m(q) = (q^2 - 1.195q + 0.432)(q - 0.135).$$

Detta motsvarar i kontinuerliga termer en pol i $s = -4$ och två poler med $\omega = 1.2$ och $\zeta = 0.7$. Vi behåller även här öppna systemets nollställen och får alltså

$$B_m(q) = K_m(0.0738q^2 + 0.1158q + 0.0058).$$

För observeraren väljer vi polerna i $\omega = 2$ med $\zeta = 0.7$, vilket motsvarar

$$A_O(q) = q^2 - 0.75q + 0.247.$$

Design-proceduren ger sedan

$$R_0(q) = q^2 - 0.3005q + 0.6418 = q^2 + r_{10}q + r_{20}$$
$$S_0(q) = -2.2515q^2 + 4.8028q - 2.0299 = s_{00}q^2 + s_{10}q + s_{20}$$
$$T_0(q) = K_m A_O(q) = t_{00}q^2 + t_{10}q + t_{20}$$

Vi vill nu skriva dessa ekvationer på tillståndsform. Ekvation (2.1) ger

$$u = -(s_{00} + \frac{(s_{10} - s_{00}r_{10})q + s_{20} - s_{00}r_{20}}{q^2 + r_{10}q + r_{20}})y$$
$$+ (t_{00} + \frac{(t_{10} - t_{00}r_{10})q + t_{20} - t_{00}r_{20}}{q^2 + r_{10}q + r_{20}})u_c$$

Tillståndsbeskrivningen blir

$$x_s(k+1) = \begin{pmatrix} -r_{10} & 1 \\ -r_{20} & 0 \end{pmatrix} x_s(k) + \begin{pmatrix} s_{00}r_{10} - s_{10} \\ s_{00}r_{20} - s_{20} \end{pmatrix} y(k)$$
$$x_t(k+1) = \begin{pmatrix} -r_{10} & 1 \\ -r_{20} & 0 \end{pmatrix} x_t(k) - \begin{pmatrix} t_{00}r_{10} - t_{10} \\ t_{00}r_{20} - t_{20} \end{pmatrix} u_c(k)$$
$$u(k) = \begin{pmatrix} 1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_s(k) \\ x_t(k) \end{pmatrix} - s_{00}y(k) + t_{00}u_c(k)$$

Regulatorn är implementerad i SIMNON-programmet `reg2`.

Identifiering

Vi följer räkningarna från tidigare och får i det här fallet

$$y_0(k) = (1 - 1.6133q^{-1} + 0.6174q^{-2} - 0.0041q^{-3})y(k),$$
$$u_0(k) = (0.0738q^{-1} + 0.1158q^{-2} + 0.0058q^{-3})u(k),$$
$$y_1(k) = (q^{-1} - 1.607q^{-2} + 0.607q^{-3})y(k),$$
$$u_1(k) = q^{-1}u(k),$$
$$u_2(k) = q^{-2}u(k),$$
$$u_3(k) = q^{-3}u(k),$$
$$y_0 - u_0 = -\alpha_1 y_1 + \beta_1 u_1 + \beta_2 u_2 + \beta_3 u_3 = \phi(k)\Theta,$$
$$\phi(k) = \begin{pmatrix} -y_1(k) & u_1(k) & u_2(k) & u_3(k) \end{pmatrix},$$
$$\Theta^T = \begin{pmatrix} \alpha_1 & \beta_1 & \beta_2 & \beta_3 \end{pmatrix}.$$

Ekvationerna (2.2) är implementerade i programmet `est2`, som finns listat i Appendix 1.

Beräkning av regulatorparametrar

Analogt med tidigare får vi

$$\Delta R(q) = \delta r_1 q + \delta r_2,$$
$$\Delta S(q) = \delta s_0 q^2 + \delta s_1 q + \delta s_2,$$

$$\mathbf{S} = \begin{pmatrix} 1 & 0 & 0.0738 & 0 & 0 \\ -1.6133 & 1 & 0.1158 & 0.0738 & 0 \\ 0.6174 & -1.6133 & 0.0058 & 0.1158 & 0.0738 \\ -0.0041 & 0.6174 & 0 & 0.0058 & 0.1158 \\ 0 & -0.0041 & 0 & 0 & 0.0058 \end{pmatrix},$$

$$\Delta\mathbf{S} = \begin{pmatrix} 0 & 0 & \beta_1 & 0 & 0 \\ \alpha_1 & 0 & \beta_2 & \beta_1 & 0 \\ -1.607\alpha_1 & \alpha_1 & \beta_3 & \beta_2 & \beta_1 \\ 0.607\alpha_1 & -1.607\alpha_1 & 0 & \beta_3 & \beta_2 \\ 0 & 0.607\alpha_1 & 0 & 0 & \beta_3 \end{pmatrix},$$

$$\begin{pmatrix} \Delta R \\ \Delta S \end{pmatrix} = \begin{pmatrix} \delta r_1 \\ \delta r_2 \\ \delta s_0 \\ \delta s_1 \\ \delta s_2 \end{pmatrix} \textit{ och}$$

$$l = \begin{pmatrix} -\alpha_1 + 2.2515\beta_1 \\ 1.3318\alpha_1 - 4.8028\beta_1 + 2.2515\beta_2 \\ -0.7659\alpha_1 + 2.0299\beta_1 - 4.8028\beta_2 + 2.2515\beta_3 \\ 1.2138\alpha_1 + 2.0299\beta_2 - 4.8028\beta_3 \\ -0.3896\alpha_1 + 2.0299\beta_3 \end{pmatrix}.$$

Första, andra och tredje ordningens approximationer finns implementerade i SIMNON-programmen `appr12`, `appr22` och `appr32`, vilka finns listade i Appendix 1.

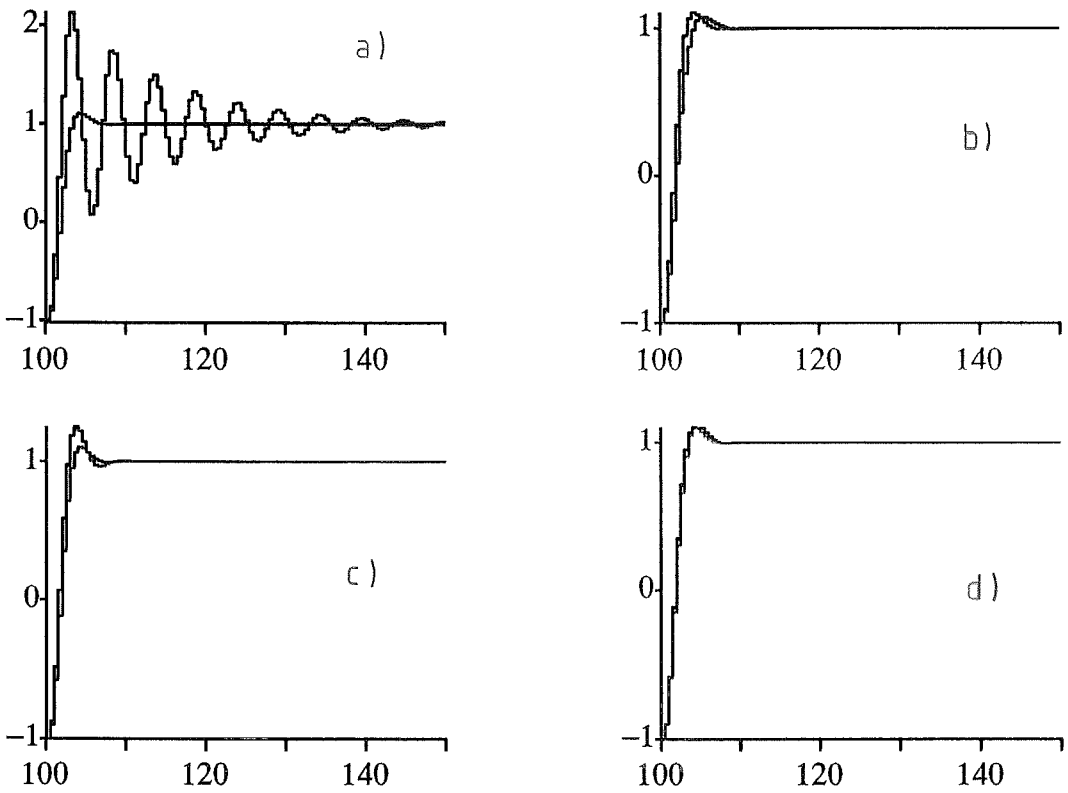
4.2 Simulering

Förutsättningar

Förutsättningarna är desamma som exempel 1, dvs vi använder samma referenssignal, och vi studerar även här den stationära regulatorn. En skillnad föreligger i det här exemplet, nämligen att processen är implementerad i diskret form. Detta gjordes för att undvika en del numeriska problem, som uppkom vid användande av en tidskontinuerlig implementering.

Resultat

I figur 4.1 kan vi studera stegsvaren för de olika regulatorerna, då $T = 0.08$ och $K = 1.5$. Polen i $s = -10$ är alltså flyttad till $s = -12.5$. Som synes är skillnaden mycket liten mellan det önskade svaret och tredje ordningens approximation, medan den nominella regulatorn inte kan sägas fungera tillfredsställande. För $T = 0.14$ och $K = 1$, polen i $s \approx -7.14$, blir resultatet



Figur 4.1 Stegsvar då $T = 0.08$ och $K = 1.5$. a) Nominell regulator, b) 1:a ordningens approxomation, c) 2:a ordningens approximation och d) 3:e ordningens approximation, alla jämförda med det önskade stegsvaret.

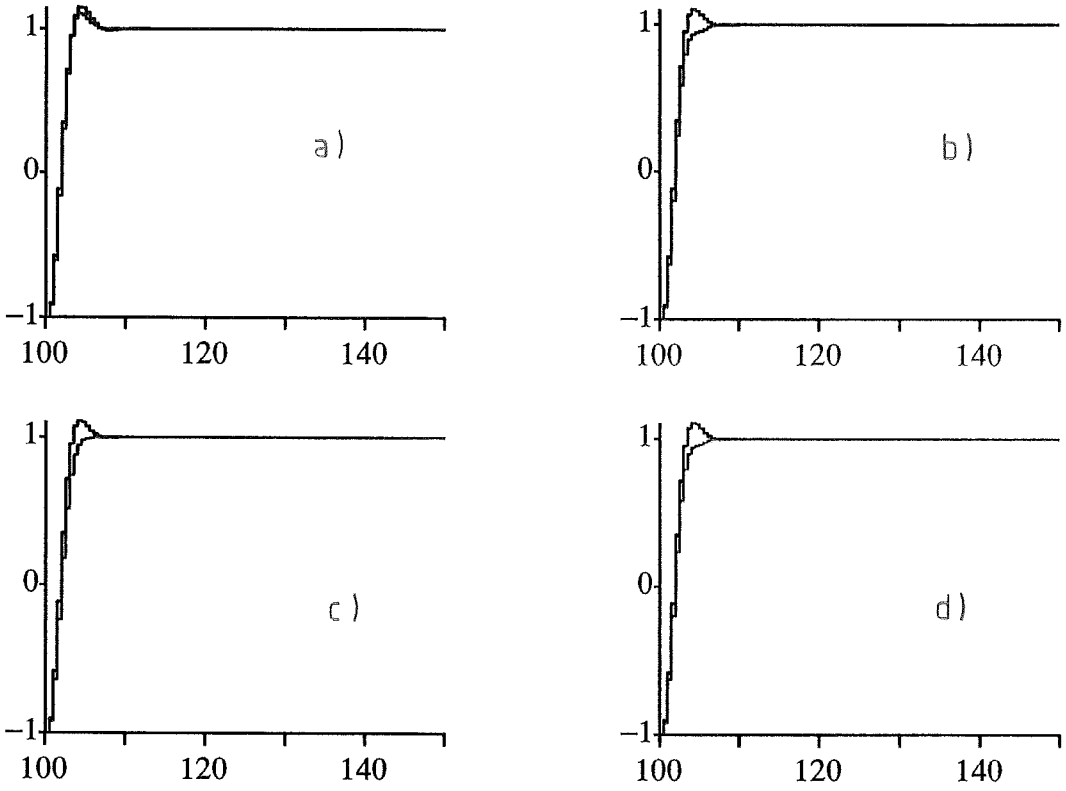
annorlunda, vilket kan beskådas i figur 4.2. Här är den nominella regulatorns stegsvar mycket nära det önskade, medan de adaptiva regulatorerna har blivit för kraftigt dämpade.

I tabell 4.1 presenteras liksom i exempel 1 de beräknade approximationerna. Räkningarna är gjorda i MATLAB. Se Appendix 2 för använda funktioner.

Stabilitet

Figurerna 4.3, 4.4 och 4.5 visar stabilitetsgränserna för de olika approximationerna. Två stora skillnader mot exempel 1 kan noteras. För det första är stabiliteten starkt begränsad för högre värden på T. Detta var inte fallet i exempel 1, där K som funktion av T kunde beskrivas som en rak linje med positiv riktningskoefficient. För det andra råkar vi här ut för det tråkiga att stabiliteten hos regulatorn kraftigt inverkar på den praktiska stabilitetsgränsen. Detta påverkas naturligtvis av hur man väljer att specificera sitt önskade system. Det kan säkerligen i svåra fall (liknande det här exemplet) löna sig att lägga ner lite extra möda på avvägningen mellan snabbhet och stabilitet hos det önskade systemet.

Den nominella regulatorns stabilitetsgräns finns inte medtagen i figurerna. Denna stabilitetsgräns bildar för de undersökta värdena på T och K en i stort sett rak linje för $K \approx 1.57$.



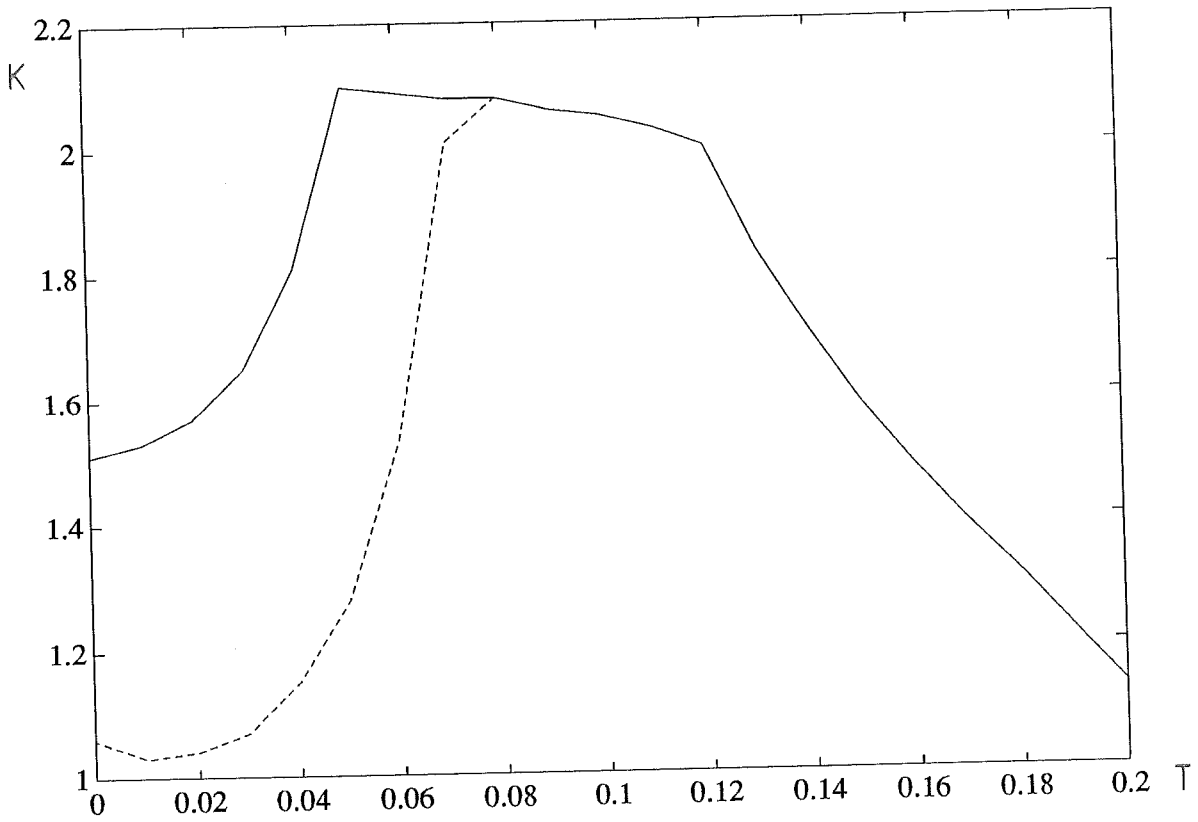
Figur 4.2 Stegsvar då $T = 0.14$ och $K = 1$.

T	K	Appr1	Appr2	Appr3	Exakt
0.08	1.5	0.1543	0.1773	0.1818	0.1910
		0.2032	0.2301	0.2375	0.2503
		-0.7823	-0.6167	-0.7782	-0.9271
		0.6489	0.6923	0.7944	0.8212
		-0.1215	-0.2013	-0.2073	-0.1782
0.14	1	-0.2863	-0.0055	-0.2495	-0.1318
		-0.4345	-0.0605	-0.3853	-0.2316
		3.8878	0.5692	3.4600	2.1013
		-6.2051	-0.8765	-5.5207	-3.3279
		2.2859	0.2752	2.0286	1.2443

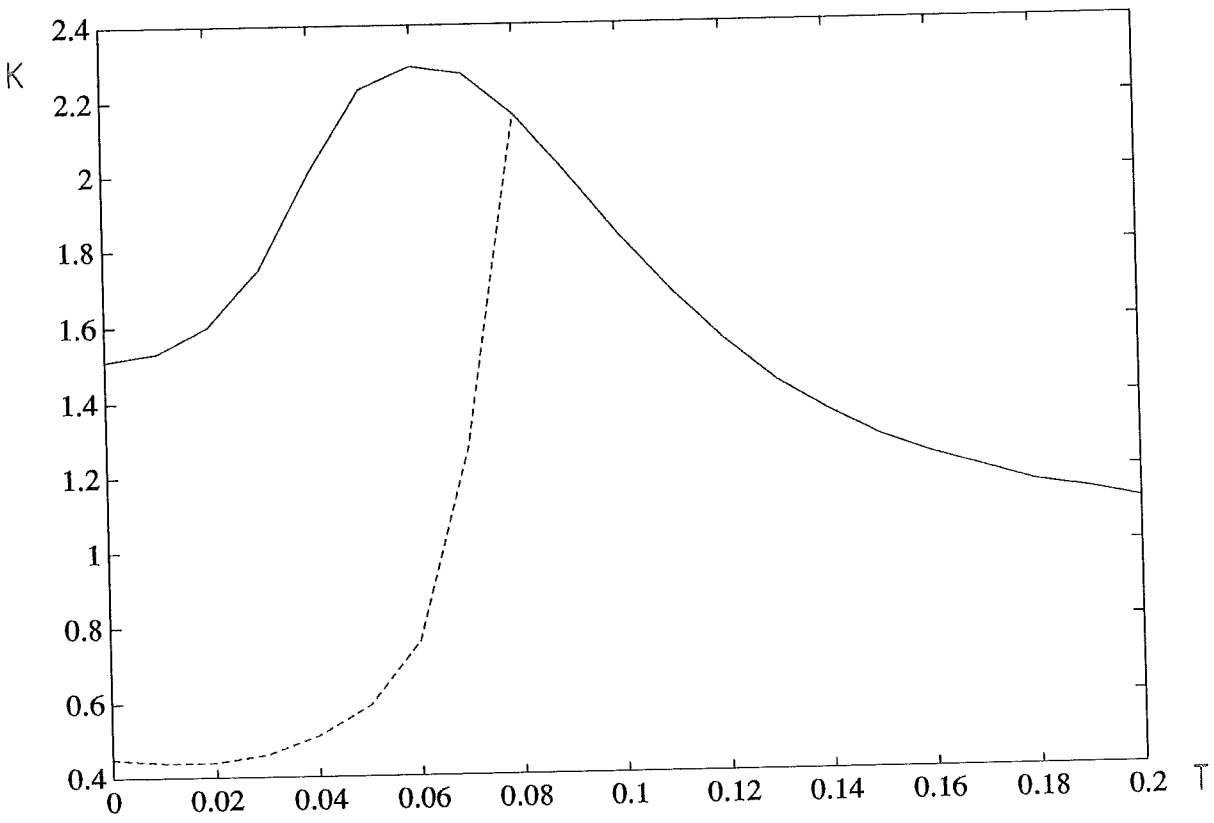
Tabell 4.1 Uträknade värden på dr_1, dr_2, ds_0, ds_1 respektive ds_2 för de olika approximationerna i de genomgångna fallen.

Sammanfattning

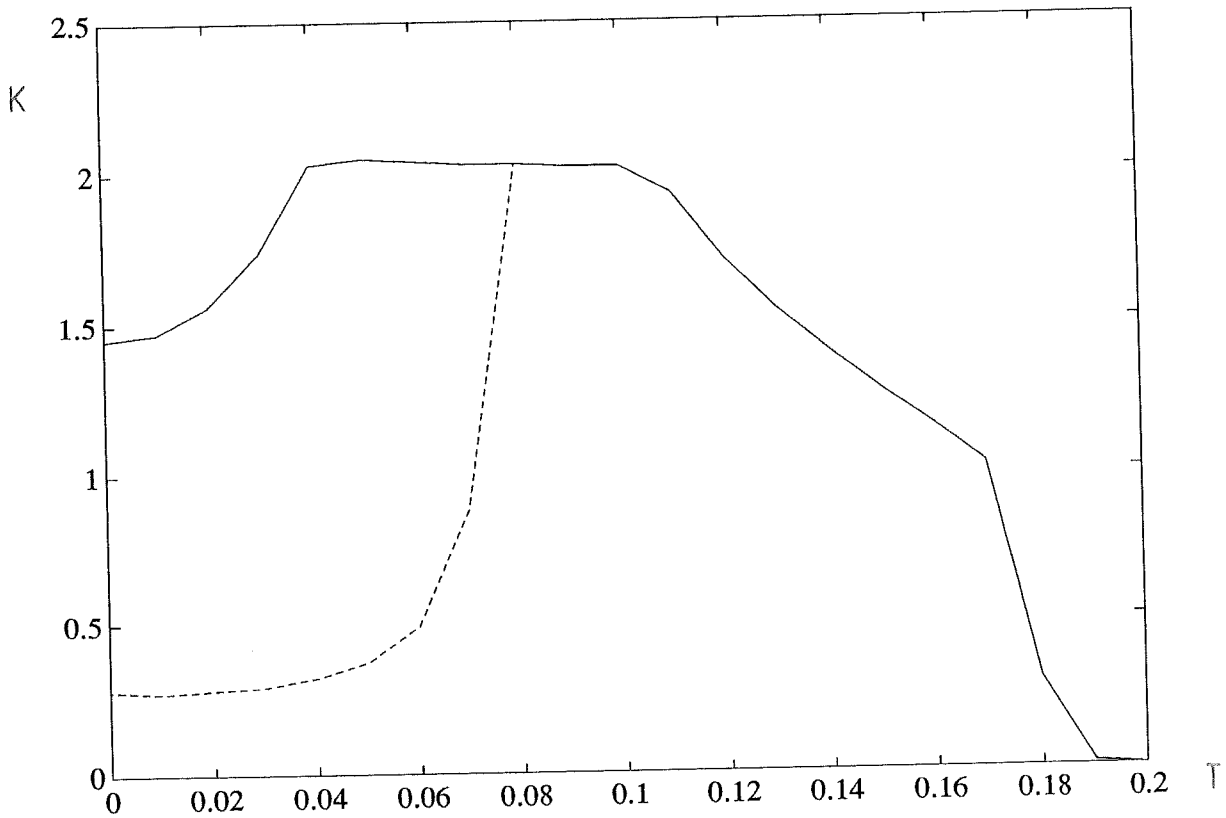
Metoden fungerar även i det här exemplet tillfredsställande. Ett problem var att regulatorns stabilitet (polynomet $R(q)$) starkt begränsade det totala stabilitetsområdet.



Figur 4.3 Stabilitetsgränser för 1:a ordningens approximation. Den heldragna linjen markerar gränsen för stabilitet hos det slutna systemet $AR + BS$, och den streckade visar stabiliteten hos regulatorpolynomet $R(q)$.



Figur 4.4 Stabilitetsgränser för 2:a ordningens approximation.



Figur 4.5 Stabilitetsgränser för 3:e ordningens approximation.

5. Sammanfattning

När man vill reglera en process adaptivt och har à priori-kunskap om processen, kan man använda den design-metod, som är beskriven i denna rapport. Metoden går ut på att man först dimensionerar en nominell regulator och sedan adaptivt modifierar denna. Metoden har den grundläggande fördelen gentemot en traditionell adaptiv regulator, att man väsentligt minskar ner beräkningsarbetet.

Adapteringen har utförts med hjälp av approximationer av olika noggrannhet. Simuleringarna visar att för rimliga avvikelser hos processparametrarna från de nominella värdena får man tydliga förbättringar jämfört med den nominella regulatorn. För större avvikelser visade det sig dock att instabilitet inträdde tidigare för de adaptiva regulatorerna än för den nominella.

Som avslutning kan sägas, att det torde i de fall då adaptiv reglering över huvud taget är tillämpbar vara av intresse att utnyttja den à priori-kunskap som föreligger om processen. Exemplen i rapporten visar att detta ger goda resultat.

6. Referenser

ANDERSSON, LEIF (1987): "Compact TEX," TFRT-7353.

ANDERSSON, MATS (1988): "Datorstöd för reglerteknik," Reglerteknik, LTH.

ELMQVIST, HILDING, ÅSTRÖM, KARL JOHAN och SCHÖNTHAL, TOMAS (1986): *SIMNON User's Guide for MS-DOS Computers.*

KNUTH, DONALD E (1987): *The TEXbook*, Addison Wesley, Reading, MA.

WITTENMARK, BJÖRN (1987): "Adaptive stability augmentation or How to use à priori knowledge," Rapport EE8713 University of Newcastle Australien.

ÅSTRÖM, KARL JOHAN och WITTENMARK, BJÖRN (1984): *Computer Controlled Systems: Theory and Design*, Prentice Hall Inc, Engelwood Cliffs, NJ.

ÅSTRÖM, KARL JOHAN och WITTENMARK, BJÖRN (1989): *Adaptive Control*, Addison Wesley, Reading, MA.

Appendix 1: SIMNON-program

Detta appendix innehåller alla de SIMNON-program, som användes i examensarbetet.

1. Exempel 1

Vi börjar med att lista de program som hör till exempel 1. Första programmet implementerar processen:

```
CONTINUOUS SYSTEM cproc
"Process
INPUT u
OUTPUT y
STATE x1 x2
DER dx1 dx2
INITIAL
a1= 1/T
a2= 0
b1= 0
b2= K/T
SORT
dx1 = -a1*x1+x2+b1*u
dx2 = -a2*x1+b2*u
y = x1
T:1
K:1
END
```

Nästa program är regulatorn:

```
DISCRETE SYSTEM reg
" Regulator
INPUT y uc dr ds0 ds1
OUTPUT u
STATE xs xt
NEW nxs nxt
TIME t
TSAMP ts
r1= -0.323+dr
s0= 1.028+ds0
s1= -0.583+ds1
k=2*(s0+s1)
t0= k
t1= -0.5*k
nxs=IF abs(xs)<10000 THEN -r1*xs+(s0*r1-s1)*y ELSE 0
nxt=IF abs(xt)<10000 THEN -r1*xt+(t1-t0*r1)*uc ELSE 0
v = xs+xt-s0*y+t0*uc
u =if v<umin then umin else if v<umax then v else umax
ts = t+h
h: 0.5
umin:-5
umax:5
END
```

Härnäst listas identifieraren:

```
DISCRETE SYSTEM est
INPUT u y
OUTPUT alfa1 beta1 beta2
STATE p11 p12 p13 p22 p23 p33 teta1 teta2 teta3 x1 x2 x3 x4
NEW np11 np12 np13 np22 np23 np33 nteta1 nteta2 nteta3 nx1 nx2 nx3 nx4
TIME t
TSAMP ts
INITIAL
p11=p0
p22=p0
p33=p0
SORT
y1=x1-x2
u1=x3
u2=x4
fi1=-y1
fi2=u1
fi3=u2
z=y-1.607*x1+0.607*x2-0.107*x3-0.09*x4
k11=p11*fi1+p12*fi2+p13*fi3
k22=p12*fi1+p22*fi2+p23*fi3
k33=p13*fi1+p23*fi2+p33*fi3
kn=lambda+fi1*k11+fi2*k22+fi3*k33
k1=k11/kn
k2=k22/kn
k3=k33/kn
e=z-(fi1*teta1+fi2*teta2+fi3*teta3)
nteta1=teta1+k1*e
nteta2=teta2+k2*e
nteta3=teta3+k3*e
np11=((1-k1*fi1)*p11-k1*fi2*p12-k1*fi3*p13)/lambda
np12=((1-k1*fi1)*p12-k1*fi2*p22-k1*fi3*p23)/lambda
np13=((1-k1*fi1)*p13-k1*fi2*p23-k1*fi3*p33)/lambda
np22=(-k2*fi1*p12+(1-k2*fi2)*p22-k2*fi3*p23)/lambda
np23=(-k2*fi1*p13+(1-k2*fi2)*p23-k2*fi3*p33)/lambda
np33=(-k3*fi1*p13-k3*fi2*p23+(1-k3*fi3)*p33)/lambda
alfa1=teta1
beta1=teta2
beta2=teta3
nx1=y
nx2=x1
nx3=u
nx4=x3
ts=t+h
h:0.5
p0:10000
lambda:0.99
END
```

Beräkning av regulatorparametrar sker i de tre följande programmen, som implementerar de olika approximationerna:

```
DISCRETE SYSTEM appr1
INPUT alfa1 beta1 beta2
OUTPUT dr ds0 ds1
TIME t
TSAMP ts
```

```
l1=-alfa1-1.028*beta1
l2=1.323*alfa1+0.583*beta1-1.028*beta2
l3=-0.323*alfa1+0.583*beta2
dr=0.2654*l1-0.3155*l2+0.3751*l3
ds0=6.8658*l1+2.9484*l2-3.5053*l3
ds1=-1.7897*l1+2.1277*l2+8.5815*l3
ts=t+h
h:0.5
END

DISCRETE SYSTEM appr2
INPUT alfa1 beta1 beta2
OUTPUT dr ds0 ds1
TIME t
TSAMP ts
l1=-alfa1-1.028*beta1
l2=1.323*alfa1+0.583*beta1-1.028*beta2
l3=-0.323*alfa1+0.583*beta2
sil1=0.2654*l1-0.3155*l2+0.3751*l3
sil2=6.8658*l1+2.9484*l2-3.5053*l3
sil3=-1.7897*l1+2.1277*l2+8.5815*l3
dr1=1+0.6906*alfa1
dr2=0.2654*beta1-0.3155*beta2
dr3=0.3155*beta1-0.3751*beta2
dr=dr1*sil1-dr2*sil2+dr3*sil3
ds01=-6.4537*alfa1
ds02=1-6.8654*beta1-2.9484*beta2
ds03=2.9484*beta1-3.5053*beta2
ds0=ds01*sil1+ds02*sil2-ds03*sil3
ds11=6.4538*alfa1
ds12=1.7897*beta1-2.1277*beta2
ds13=1-2.1277*beta1-8.5815*beta2
ds1=ds11*sil1+ds12*sil2+ds13*sil3
ts=t+h
h:0.5
END

DISCRETE SYSTEM appr3
INPUT alfa1 beta1 beta2
OUTPUT dr ds0 ds1
TIME t
TSAMP ts
l1=-alfa1-1.028*beta1
l2=1.323*alfa1+0.583*beta1-1.028*beta2
l3=-0.323*alfa1+0.583*beta2
sil1=0.2654*l1-0.3155*l2+0.3751*l3
sil2=6.8658*l1+2.9484*l2-3.5053*l3
sil3=-1.7897*l1+2.1277*l2+8.5815*l3
x11=-0.6906*alfa1
x12=0.2654*beta1-0.3155*beta2
x13=-0.3155*beta1+0.3751*beta2
x21=6.4537*alfa1
x22=6.8654*beta1+2.9484*beta2
x23=2.9484*beta1-3.5053*beta2
x31=-6.4538*alfa1
x32=-1.7897*beta1+2.1277*beta2
x33=2.1277*beta1+8.5815*beta2
xx11=x11*x11+x12*x21+x13*x31
```

```
xx12=x11*x12+x12*x22+x13*x32
xx13=x11*x13+x12*x23+x13*x33
xx21=x21*x11+x22*x21+x23*x31
xx22=x21*x12+x22*x22+x23*x32
xx23=x21*x13+x22*x23+x23*x33
xx31=x31*x11+x32*x21+x33*x31
xx32=x31*x12+x32*x22+x33*x32
xx33=x31*x13+x32*x23+x33*x33
dr=(1-x11+xx11)*sil1+(-x12+xx12)*sil2+(-x13+xx13)*sil3
ds0=(-x21+xx21)*sil1+(1-x22+xx22)*sil2+(-x23+xx23)*sil3
ds1=(-x31+xx31)*sil1+(-x32+xx32)*sil2+(1-x33+xx33)*sil3
ts=t+h
h:0.5
END
```

Det sista programmet för exempel 1 är

```
CONNECTING SYSTEM consys
TIME t
u[cproc] = u[reg]
y[reg] = y[cproc]
uc[reg] = u[square]
dr[reg]=if p<1.5 then dr[appr1] else if p<2.5 then dr[appr2] else dr[appr3]
ds0[reg]=if p<1.5 then ds0[appr1] else if p<2.5 then ds0[appr2] else ds0[appr3]
ds1[reg]=if p<1.5 then ds1[appr1] else if p<2.5 then ds1[appr2] else ds1[appr3]
alfa1[appr1]=alfa1[est]
beta1[appr1]=beta1[est]
beta2[appr1]=beta2[est]
alfa1[appr2]=alfa1[est]
beta1[appr2]=beta1[est]
beta2[appr2]=beta2[est]
alfa1[appr3]=alfa1[est]
beta1[appr3]=beta1[est]
beta2[appr3]=beta2[est]
u[est]=u[reg]
y[est]=y[cproc]
p:1
END
```

2. Exempel 2

Nu listas de program som gäller exempel 2 i samma ordning som ovan.

```
DISCRETE SYSTEM proc2
INPUT u
OUTPUT y
STATE x1 x2 x3
NEW nx1 nx2 nx3
TIME t
TSAMP ts
INITIAL
a1=-1.6133+alfa
a2=0.6174-1.607*alfa
a3=-0.0041+0.607*alfa
b1=0.0738+beta1
b2=0.1158+beta2
b3=0.0058+beta3
```

```
SORT
nx1=-a1*x1+x2+b1*u
nx2=-a2*x1+x3+b2*u
nx3=-a3*x1+b3*u
y=x1
ts=t+h
alfa:0
beta1:0
beta2:0
beta3:0
h:0.5
END

DISCRETE SYSTEM reg2
" regulator till K/s(1+Ts)(1+T1S)
INPUT y uc dr1 dr2 ds0 ds1 ds2
OUTPUT u
STATE xs1 xs2 xt1 xt2
NEW nxs1 nxs2 nxt1 nxt2
TIME t
TSAMP ts
r1= -0.3005+dr1
r2= 0.6418+dr2
s0= -2.2515+ds0
s1= 4.8028+ds1
s2= -2.0299+ds2
k= (s0+s1+s2)/0.787
t0= k
t1= -0.335*k
t2= 0.122*k
nxs1 = -r1*xs1+xs2+(s0*r1-s1)*y
nxs2 = -r2*xs1+(s0*r2-s2)*y
nxt1 = -r1*xt1+xt2+(t1-t0*r1)*uc
nxt2 = -r2*xt1+(t2-t0*r2)*uc
v = xs1+xt1-s0*y+t0*uc
u= if v<vmin then vmin else if v<vmax then v else vmax
ts = t+h
h: 0.5
vmin:-5
vmax:5
END

DISCRETE SYSTEM est2
INPUT u y
OUTPUT alfa beta1 beta2 beta3
STATE p11 p12 p13 p14 p22 p23 p24 p33 p34 p44 t1 t2 t3 t4
NEW np11 np12 np13 np14 np22 np23 np24 np33 np34 np44 nt1 nt2 nt3 nt4
STATE x1 x2 x3 x4 x5 x6
NEW nx1 nx2 nx3 nx4 nx5 nx6
TIME t
TSAMP ts
INITIAL
p11=p0
p22=p0
p33=p0
p44=p0
SORT
y1=x1-1.607*x2+0.607*x3
```

```
u1=x4
u2=x5
u3=x6
fi1=-y1
fi2=u1
fi3=u2
fi4=u3
a1=-1.6133
a2=0.6174
a3=-0.0041
b1=0.0738
b2=0.1158
b3=0.0058
z=y+a1*x1+a2*x2+a3*x3-b1*x4-b2*x5-b3*x6
k11=p11*fi1+p12*fi2+p13*fi3+p14*fi4
k22=p12*fi1+p22*fi2+p23*fi3+p24*fi4
k33=p13*fi1+p23*fi2+p33*fi3+p34*fi4
k44=p14*fi1+p24*fi2+p34*fi3+p44*fi4
kn=lambda+fi1*k11+fi2*k22+fi3*k33+fi4*k44
k1=k11/kn
k2=k22/kn
k3=k33/kn
k4=k44/kn
e=z-(fi1*t1+fi2*t2+fi3*t3+fi4*t4)
nt1=t1+k1*e
nt2=t2+k2*e
nt3=t3+k3*e
nt4=t4+k4*e
np11=((1-k1*fi1)*p11-k1*fi2*p12-k1*fi3*p13-k1*fi4*p14)/lambda
np12=((1-k1*fi1)*p12-k1*fi2*p22-k1*fi3*p23-k1*fi4*p24)/lambda
np13=((1-k1*fi1)*p13-k1*fi2*p23-k1*fi3*p33-k1*fi4*p34)/lambda
np14=((1-k1*fi1)*p14-k1*fi2*p24-k1*fi3*p34-k1*fi4*p44)/lambda
np22=(-k2*fi1*p12+(1-k2*fi2)*p22-k2*fi3*p23-k2*fi4*p24)/lambda
np23=(-k2*fi1*p13+(1-k2*fi2)*p23-k2*fi3*p33-k2*fi4*p34)/lambda
np24=(-k2*fi1*p14+(1-k2*fi2)*p24-k2*fi3*p34-k2*fi4*p44)/lambda
np33=(-k3*fi1*p13-k3*fi2*p23+(1-k3*fi3)*p33-k3*fi4*p34)/lambda
np34=(-k3*fi1*p14-k3*fi2*p24+(1-k3*fi3)*p34-k3*fi4*p44)/lambda
np44=(-k4*fi1*p14-k4*fi2*p24-k4*fi3*p34+(1-k4*fi4)*p44)/lambda
alfa=t1
beta1=t2
beta2=t3
beta3=t4
nx1=y
nx2=x1
nx3=x2
nx4=u
nx5=x4
nx6=x5
ts=t+h
h:0.5
p0:10000
lambda:0.99
END

DISCRETE SYSTEM appr12
INPUT alfa beta1 beta2 beta3
OUTPUT dr1 dr2 ds0 ds1 ds2
```

```
TIME t
TSAMP ts
l1=-alfa+2.2515*beta1
l2=1.3318*alfa-4.8028*beta1+2.2515*beta2
l3=-0.7659*alfa+2.0299*beta1-4.8028*beta2+2.2515*beta3
l4=1.2138*alfa+2.0299*beta2-4.8028*beta3
l5=-0.3896*alfa+2.0299*beta3
dr1=0.4437*l1-0.2902*l2+0.1476*l3+0.7446*l4-16.7454*l5
dr2=0.0228*l1-0.0116*l2-0.0585*l3+1.316*l4-25.5306*l5
ds0=7.5375*l1+3.9321*l2-2.0006*l3-10.0897*l4+226.9018*l5
ds1=-2.4359*l1+1.1938*l2+7.1597*l3+14.2771*l4-376.1504*l5
ds2=0.0161*l1-0.0082*l2-0.0414*l3+0.9303*l4+154.3663*l5
ts=t+h
h:0.5
END

DISCRETE SYSTEM appr22
INPUT alfa beta1 beta2 beta3
OUTPUT dr1 dr2 ds0 ds1 ds2
TIME t
TSAMP ts
l1=-alfa+2.2515*beta1
l2=1.3318*alfa-4.8028*beta1+2.2515*beta2
l3=-0.7659*alfa+2.2099*beta1-4.8028*beta2+2.2515*beta3
l4=1.2138*alfa+2.0299*beta2-4.8028*beta3
l5=-0.3896*alfa+2.0299*beta3
sil1=0.4437*l1-0.2902*l2+0.1476*l3+0.7446*l4-16.7454*l5
sil2=0.0228*l1-0.0116*l2-0.0585*l3+1.316*l4-25.5306*l5
sil3=7.5375*l1+3.9321*l2-2.0006*l3-10.0897*l4+226.9018*l5
sil4=-2.4359*l1+1.1938*l2+7.1597*l3+14.2771*l4-376.1504*l5
sil5=0.0161*l1-0.0082*l2-0.0414*l3+0.9303*l4+154.3663*l5
sds11=-0.0754*alfa
sds12=-11.2134*alfa
sds13=0.4437*beta1-0.2902*beta2+0.1476*beta3
sds14=-0.2902*beta1+0.1476*beta2+0.7446*beta3
sds15=0.1476*beta1+0.7446*beta2-16.7454*beta3
sds21=0.8812*alfa
sds22=-17.6704*alfa
sds23=0.0228*beta1-0.0116*beta2-0.0585*beta3
sds24=-0.0116*beta1-0.0585*beta2+1.316*beta3
sds25=-0.0585*beta1+1.316*beta2-25.5306*beta3
sds31=1.0226*alfa
sds32=151.9429*alfa
sds33=7.5375*beta1+3.9321*beta2-2.0006*beta3
sds34=3.9321*beta1-2.0006*beta2-10.0897*beta3
sds35=-2.0006*beta1-10.0897*beta2+226.9018*beta3
sds41=-1.6456*alfa
sds42=-244.1069*alfa
sds43=-2.43159*beta1+1.1938*beta2+7.1597*beta3
sds44=1.1938*beta1+7.1597*beta2+14.2771*beta3
sds45=7.1597*beta1+14.2771*beta2-376.1504*beta3
sds51=0.623*alfa
sds52=92.164*alfa
sds53=0.0161*beta1-0.0082*beta2-0.0414*beta3
sds54=-0.0082*beta1-0.0414*beta2+0.9303*beta3
sds55=-0.0414*beta1+0.9303*beta2+154.3663*beta3
dr1=(1-sds11)*sil1-sds12*sil2-sds13*sil3-sds14*sil4-sds15*sil5
```

```
dr2=-sds21*sil1+(1-sds22)*sil2-sds23*sil3-sds24*sil4-sds25*sil5
ds0=-sds31*sil1-sds32*sil2+(1-sds33)*sil3-sds34*sil4-sds35*sil5
ds1=-sds41*sil1-sds42*sil2-sds43*sil3+(1-sds44)*sil4-sds45*sil5
ds2=-sds51*sil1-sds52*sil2-sds53*sil3-sds54*sil4+(1-sds55)*sil5
ts=t+h
h:0.5
END

DISCRETE SYSTEM appr32
INPUT alfa beta1 beta2 beta3
OUTPUT dr1 dr2 ds0 ds1 ds2
TIME t
TSAMP ts
l1=-alfa+2.2515*beta1
l2=1.3318*alfa-4.8028*beta1+2.2515*beta2
l3=-0.7659*alfa+2.0299*beta1-4.8028*beta2+2.2515*beta3
l4=1.2138*alfa+2.0299*beta2-4.8028*beta3
l5=-0.3896*alfa+2.0299*beta3
sil1=0.4437*l1-0.2902*l2+0.1476*l3+0.7446*l4-16.7454*l5
sil2=0.0228*l1-0.0116*l2-0.0585*l3+1.316*l4-25.5306*l5
sil3=7.5375*l1+3.9321*l2-2.0006*l3-10.0897*l4+226.9018*l5
sil4=-2.4359*l1+1.1938*l2+7.1597*l3+14.2771*l4-376.1504*l5
sil5=0.0161*l1-0.0082*l2-0.0414*l3+0.9303*l4+154.3663*l5
sds11=-0.0754*alfa
sds12=-11.2134*alfa
sds13=0.4437*beta1-0.2902*beta2+0.1476*beta3
sds14=-0.2902*beta1+0.1476*beta2+0.7446*beta3
sds15=0.1476*beta1+0.7446*beta2-16.7454*beta3
sds21=0.8812*alfa
sds22=-17.6704*alfa
sds23=0.0228*beta1-0.0116*beta2-0.0585*beta3
sds24=-0.0116*beta1-0.0585*beta2+1.316*beta3
sds25=-0.0585*beta1+1.316*beta2-25.5306*beta3
sds31=1.0226*alfa
sds32=151.9429*alfa
sds33=7.5375*beta1+3.9321*beta2-2.0006*beta3
sds34=3.9321*beta1-2.0006*beta2-10.0897*beta3
sds35=-2.0006*beta1-10.0897*beta2+226.9018*beta3
sds41=-1.6456*alfa
sds42=-244.1069*alfa
sds43=-2.43159*beta1+1.1938*beta2+7.1597*beta3
sds44=1.1938*beta1+7.1597*beta2+14.2771*beta3
sds45=7.1597*beta1+14.2771*beta2-376.1504*beta3
sds51=0.623*alfa
sds52=92.164*alfa
sds53=0.0161*beta1-0.0082*beta2-0.0414*beta3
sds54=-0.0082*beta1-0.0414*beta2+0.9303*beta3
sds55=-0.0414*beta1+0.9303*beta2+154.3663*beta3
m11=sds11-(sds11*sds11+sds12*sds21+sds13*sds31+sds14*sds41+sds15*sds51)
m12=sds12-(sds11*sds12+sds12*sds22+sds13*sds32+sds14*sds42+sds15*sds52)
m13=sds13-(sds11*sds13+sds12*sds23+sds13*sds33+sds14*sds43+sds15*sds53)
m14=sds14-(sds11*sds14+sds12*sds24+sds13*sds34+sds14*sds44+sds15*sds54)
m15=sds15-(sds11*sds15+sds12*sds25+sds13*sds35+sds14*sds45+sds15*sds55)
m21=sds21-(sds21*sds11+sds22*sds21+sds23*sds31+sds24*sds41+sds25*sds51)
m22=sds22-(sds21*sds12+sds22*sds22+sds23*sds32+sds24*sds42+sds25*sds52)
m23=sds23-(sds21*sds13+sds22*sds23+sds23*sds33+sds24*sds43+sds25*sds53)
m24=sds24-(sds21*sds14+sds22*sds24+sds23*sds34+sds24*sds44+sds25*sds54)
```

```
m25=sds25-(sds21*sds15+sds22*sds25+sds23*sds35+sds24*sds45+sds25*sds55)
m31=sds31-(sds31*sds11+sds32*sds21+sds33*sds31+sds34*sds41+sds35*sds51)
m32=sds32-(sds31*sds12+sds32*sds22+sds33*sds32+sds34*sds42+sds35*sds52)
m33=sds33-(sds31*sds13+sds32*sds23+sds33*sds33+sds34*sds43+sds35*sds53)
m34=sds34-(sds31*sds14+sds32*sds24+sds33*sds34+sds34*sds44+sds35*sds54)
m35=sds35-(sds31*sds15+sds32*sds25+sds33*sds35+sds34*sds45+sds35*sds55)
m41=sds41-(sds41*sds11+sds42*sds21+sds43*sds31+sds44*sds41+sds45*sds51)
m42=sds42-(sds41*sds12+sds42*sds22+sds43*sds32+sds44*sds42+sds45*sds52)
m43=sds43-(sds41*sds13+sds42*sds23+sds43*sds33+sds44*sds43+sds45*sds53)
m44=sds44-(sds41*sds14+sds42*sds24+sds43*sds34+sds44*sds44+sds45*sds54)
m45=sds45-(sds41*sds15+sds42*sds25+sds43*sds35+sds44*sds45+sds45*sds55)
m51=sds51-(sds51*sds11+sds52*sds21+sds53*sds31+sds54*sds41+sds55*sds51)
m52=sds52-(sds51*sds12+sds52*sds22+sds53*sds32+sds54*sds42+sds55*sds52)
m53=sds53-(sds51*sds13+sds52*sds23+sds53*sds33+sds54*sds43+sds55*sds53)
m54=sds54-(sds51*sds14+sds52*sds24+sds53*sds34+sds54*sds44+sds55*sds54)
m55=sds55-(sds51*sds15+sds52*sds25+sds53*sds35+sds54*sds45+sds55*sds55)
dr1=(1-m11)*sil1-m12*sil2-m13*sil3-m14*sil4-m15*sil5
dr2=-m21*sil1+(1-m22)*sil2-m23*sil3-m24*sil4-m25*sil5
ds0=-m31*sil1-m32*sil2+(1-m33)*sil3-m34*sil4-m35*sil5
ds1=-m41*sil1-m42*sil2-m43*sil3+(1-m44)*sil4-m45*sil5
ds2=-m51*sil1-m52*sil2-m53*sil3-m54*sil4+(1-m55)*sil5
ts=t+h
h:0.5
END
CONNECTING SYSTEM consys2d
TIME t
u[proc2] = u[reg2]
y[reg2] = y[proc2]
uc[reg2] = u[square]
dr1=dr1[appr12]
dr2=dr2[appr12]
ds0=ds0[appr12]
ds1=ds1[appr12]
ds2=ds2[appr12]
dr1[reg2]=if p<1.5 then dr1 else if p<2.5 then dr1[appr22] else dr1[appr32]
dr2[reg2]=if p<1.5 then dr2 else if p<2.5 then dr2[appr22] else dr2[appr32]
ds0[reg2]=if p<1.5 then ds0 else if p<2.5 then ds0[appr22] else ds0[appr32]
ds1[reg2]=if p<1.5 then ds1 else if p<2.5 then ds1[appr22] else ds1[appr32]
ds2[reg2]=if p<1.5 then ds2 else if p<2.5 then ds2[appr22] else ds2[appr32]
alfa[appr12]=alfa[est2]
beta1[appr12]=beta1[est2]
beta2[appr12]=beta2[est2]
beta3[appr12]=beta3[est2]
alfa[appr22]=alfa[est2]
beta1[appr22]=beta1[est2]
beta2[appr22]=beta2[est2]
beta3[appr22]=beta3[est2]
alfa[appr32]=alfa[est2]
beta1[appr32]=beta1[est2]
beta2[appr32]=beta2[est2]
beta3[appr32]=beta3[est2]
u[est2]=u[reg2]
y[est2]=y[proc2]
p:1
END
```

3. Referenssignal

Ett program återstår, nämligen referenssignalen:

```
DISCRETE SYSTEM square
"Fyrkantv}g
OUTPUT u
TIME t
TSAMP ts
u= IF mod(t,per)<per/2 THEN umax ELSE umin
ts=t+h
h:0.5
umax:1
umin:-1
per:100
END
```

Appendix 2: MATLAB-funktioner

Detta appendix innehåller de viktigaste av de MATLAB-funktioner, som blev skrivna för examensarbetet.

1. Exempel 1

För omvandling av parametrarna T och K till α_1, β_1 och β_2, t ex för koll av identifieringen användes

```
function [alfa1,beta1,beta2]=tkomv(T,K)
%[alfa1,beta1,beta2]=tomv(T,K)
b1=K*T*(0.5/T-1+exp(-0.5/T))
b2=K*T*(1-exp(-0.5/T)-(0.5/T)*exp(-0.5/T))
a1=-(1+exp(-0.5/T))
a2=exp(-0.5/T)
beta1=b1-(0.5-1+exp(-0.5))
beta2=b2-(1-exp(-0.5)-0.5*exp(-0.5))
alfa11=a1+(1+exp(-0.5))
alfa1=exp(-0.5)-a2
```

De olika approximativa regulatorerna kan beräknas med följande funktioner:

```
function [a,b,r,s,t]=appdes(a1,b1,b2)
%[a,b,r,s,t]=appdes(alfa1,beta1,beta2)
a=[1 -1.607+a1 0.607-a1]
b=[0.107+b1 0.09+b2]
ao=[1 -.5]
S=[1 0.107 0;-1.607 0.09 0.107;0.607 0 0.09]
l=[-a1-1.028*b1;1.323*a1+0.583*b1-1.028*b2;-0.323*a1+0.583*b2]
x=inv(S)*l
r=[1 -0.323+x(1)]
s=[1.028+x(2) -0.583+x(3)]
k=(s(1)+s(2))*2
t=k*ao
```

```
function [a,b,r,s,t] = appr2(a1,b1,b2)
%design, anv tv} termer i serieutv av inv(S+dS)
a=[1 -1.607+a1 0.607-a1]
b=[0.107+b1 0.09+b2]
Z=[1 0.107 0;-1.607 0.09 0.107;0.607 0 0.09]
dZ=[0 b1 0;a1 b2 b1;-a1 0 b2]
L=[-a1-1.028*b1;1.323*a1+0.583*b1-1.028*b2;-0.323*a1+0.583*b2]
drds=(eye(3)-inv(Z)*dZ)*inv(Z)*L
r=[1 -0.323+drds(1)]
s=[1.028+drds(2) -0.583+drds(3)]
k=2*(s(1)+s(2))
t=k*[1 -0.5]
```

```
function [a,b,r,s,t] = appr3(a1,b1,b2)
%design, anv tv} termer i serieutv av inv(S+dS)
a=[1 -1.607+a1 0.607-a1]
b=[0.107+b1 0.09+b2]
Z=[1 0.107 0;-1.607 0.09 0.107;0.607 0 0.09]
dZ=[0 b1 0;a1 b2 b1;-a1 0 b2]
L=[-a1-1.028*b1;1.323*a1+0.583*b1-1.028*b2;-0.323*a1+0.583*b2]
```

```
x=inv(Z)*dZ
drds=(eye(3)-x+x*x)*inv(Z)*L
r=[1 -0.323+drds(1)]
s=[1.028+drds(2) -0.583+drds(3)]
k=2*(s(1)+s(2))
t=k*[1 -0.5]
```

Den optimala regulatorn beräknas med

```
function [a,b,r,s,t] = des(a1,b1,b2)
a=[1 -1.607+a1 0.607-a1]
b=[0.107+b1 0.09+b2]
ao=[1 -0.5]
S=[1 0.107+b1 0;a1-1.607 0.09+b2 0.107+b1;0.607-a1 0 0.09+b2]
L=[-a1-1.028*b1;1.323*a1+0.583*b1-1.028*b2;-0.323*a1+0.583*b2]
drds=inv(S)*L
r=[1 -0.323+drds(1)]
s=[1.028+drds(2) -0.583+drds(3)]
k=2*(s(1)+s(2))
t=k*ao
```

2. Exempel 2

Motsvarande funktioner som ovan för exempel 2 följer här:

```
function [alfa,beta1,beta2,beta3]=tkomv2(t,k)
c=exp(-.5)
e=exp(-.5/t)
b1=k*(.5+(1+c+e)*(1+t)-1/(1-t)*(2+e)-t*t/(t-1)*(2+c))
b2=k*(-(1+t)*(c+(1+c)*e)-.5*(c+e)+1/(1-t)*(2*e+1)+t*t/(t-1)*2.214)
b3=k*((1+t)*c*e+c/2*e-1/(1-t)*e-t*t/(t-1)*c)
a1=-1-c-e
a10=-1-c-exp(-5)
b10=.5+(1+c+exp(-5))*1.1-1/.9*(2+exp(-5))+.01/.9*(2+c)
b20=-1.1*(c+(1+c)*exp(-5))-.5*(c+exp(-5))+1/.9*(2*exp(-5)+1)-.01/.9*2.214
b30=1.1*c*exp(-5)+c/2*exp(-5)-1/.9*exp(-5)+.01/.9*c
alfa=a1-a10
beta1=b1-b10
beta2=b2-b20
beta3=b3-b30
```

```
function [a,b,r,s,t] = appr12(a1,b1,b2,b3)
%design, anv en term i serieutv av inv(S+dS)
a=[1 -1.6133+a1 0.6174-1.607*a1 0.0041+0.607*a1]
b=[0.0738+b1 0.1158+b2 0.0058+b3]
Z=[1 0 .0738 0 0
  -1.6133 1 .1158 .0738 0
  .6174 -1.6133 .0058 .1158 .0738
  -.0041  .6174 0 .0058 .1158
  0 -.0041 0 0 .0058]
L=[-a1+2.2515*b1;1.3318*a1-4.8028*b1+2.2515*b2
  -0.7659*a1+2.0299*b1-4.8028*b2+2.2515*b3
  1.2138*a1+2.0299*b2-4.8028*b3;-0.3896*a1+2.0299*b3]
drds=inv(Z)*L
r=[1 -0.3005+drds(1) 0.6418+drds(2)]
s=[-2.2515+drds(3) 4.8028+drds(4) -2.0299+drds(5)]
k=(s(1)+s(2)+s(3))/0.497
t=k*[1 -0.75 0.247]
```

```
function [a,b,r,s,t] = appr22(a1,b1,b2,b3)
%design, anv en term i serieutv av inv(S+dS)
a=[1 -1.6133+a1 0.6174-1.607*a1 0.0041+0.607*a1]
b=[0.0738+b1 0.1158+b2 0.0058+b3]
Z=[1 0 .0738 0 0
  -1.6133 1 .1158 .0738 0
  .6174 -1.6133 .0058 .1158 .0738
  -.0041  .6174 0 .0058 .1158
  0 -.0041 0 0 .0058]
dZ=[0 0 b1 0 0
a1 0 b2 b1 0
-1.607*a1 a1 b3 b2 b1
0.607*a1 -1.607*a1 0 b3 b2
0 0.607*a1 0 0 b3]
L=[-a1+2.2515*b1;1.3318*a1-4.8028*b1+2.2515*b2
  -0.7659*a1+2.0299*b1-4.8028*b2+2.2515*b3
  1.2138*a1+2.0299*b2-4.8028*b3;-0.3896*a1+2.0299*b3]
drds=(eye(5)-inv(Z)*dZ)*inv(Z)*L
r=[1 -0.3005+drds(1) 0.6418+drds(2)]
s=[-2.2515+drds(3) 4.8028+drds(4) -2.0299+drds(5)]
k=(s(1)+s(2)+s(3))/0.497
t=k*[1 -0.75 0.247]

function [a,b,r,s,t] = appr32(a1,b1,b2,b3)
a=[1 -1.6133+a1 0.6174-1.607*a1 0.0041+0.607*a1]
b=[0.0738+b1 0.1158+b2 0.0058+b3]
Z=[1 0 .0738 0 0
  -1.6133 1 .1158 .0738 0
  .6174 -1.6133 .0058 .1158 .0738
  -.0041  .6174 0 .0058 .1158
  0 -.0041 0 0 .0058]
dZ=[0 0 b1 0 0
a1 0 b2 b1 0
-1.607*a1 a1 b3 b2 b1
0.607*a1 -1.607*a1 0 b3 b2
0 0.607*a1 0 0 b3]
L=[-a1+2.2515*b1;1.3318*a1-4.8028*b1+2.2515*b2
  -0.7659*a1+2.0299*b1-4.8028*b2+2.2515*b3
  1.2138*a1+2.0299*b2-4.8028*b3;-0.3896*a1+2.0299*b3]
x=inv(Z)*dZ
drds=(eye(5)-x+x*x)*inv(Z)*L
r=[1 -0.3005+drds(1) 0.6418+drds(2)]
s=[-2.2515+drds(3) 4.8028+drds(4) -2.0299+drds(5)]
k=(s(1)+s(2)+s(3))/0.497
t=k*[1 -0.75 0.247]

function [a,b,r,s,t] = des2(a1,b1,b2,b3)
a=[1 -1.6133+a1 0.6174-1.607*a1 -0.0041+0.607*a1]
b=[0.0738+b1 0.1158+b2 0.0058+b3]
ao=[1 -0.335 0.122]
S=[1 0 0.0738+b1 0 0
-1.6138+a1 1 0.1158+b2 0.0738+b1 0
0.6174-1.607*a1 -1.6138+a1 0.0058+b3 0.1158+b2 0.0738+b1
-0.0041+0.607*a1 0.6174-1.607*a1 0 0.0058+b3 0.1158+b2
0 -0.0041+0.607*a1 0 0 0.0058+b3]
L=[-a1+1.9005*b1
1.5738*a1-5.0347*b1+1.9005*b2
-1.1486*a1+2.2878*b1-5.0347*b2+1.9005*b3
```

```
0.936*a1+2.2878*b2-5.0347*b3
-0.3612*a1+2.2878*b3]
drds=inv(S)*L
r=[1 0.0332+drds(1) 0.595+drds(2)]
s=[-1.9005+drds(3) 5.0347+drds(4) -2.2878+drds(5)]
k=(s(1)+s(2)+s(3))/0.787
t=k*ao
```

3. Stabilitet

En användbar liten funktion kommer här:

```
function r=stab(a,b,r,s)
n1=conv(a,r)
n2=conv(b,s)
n=n1+[0 n2]
r=abs(roots(n))
```