

CODEN: LUTFD2/(TFRT-5447)/1-95/(1991)

Jämförelse Mellan den Adaptiva Regulatorn Novatune och en PID-Regulator

Ola Bernersson
Rikard Berglund

Institutionen för Reglerteknik
Lunds Tekniska Högskola
December 1991

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 1991	
		<i>Document Number</i> CODEN: LUTFD2/(TFRT-5447)/1-95/(1991)	
<i>Author(s)</i> Ola Bernersson and Rikard Berglund		<i>Supervisor</i> Mats Molander ABB and Karl Johan Åström	
		<i>Sponsoring organisation</i>	
<i>Title and subtitle</i> Jämförelse mellan den adaptiva regulatorn Novatune och en PID-regulator (Comparison between the adaptive controller Novatune and a PID-controller).			
<i>Abstract</i> <p>The adaptive controller Novatune, which is based on minimum-variance control, has been compared to a PID-controller based on dominant pole-placement design. Responses to set point changes, load disturbances and measurement errors have been studied. Both deterministic and stochastic disturbances have been used. The sensitivity to parameter variations is also investigated. Criteria for quantitative characterization of the responses are given. The controllers have been evaluated on a large batch of test examples representing processes with widely varying character. The comparisons are based on simulations in Simnon. A number of auxiliary routines in MATLAB have been written to evaluate the results.</p> <p>As a rule the results with Novatune have been better. This is particularly true when parameter changes are considered because Novatune can adapt to a new situation. The differences are, however, small for simple processes. The performance of the Novatune depends on choices of several parameters which have to be initialized properly. The standard procedure given in the installation manual for Novatune has been used when possible. Such procedures are not available for integrating and oscillatory processes. It has been shown that Novatune is quite sensitive for integrating processes with variable time delays. The PID controller was more robust for those processes. There are processes in the test batch where PID control gives poor results due to its restricted complexity. A double integrator with time delay is one example. For processes with integration we have also proposed commissioning rules which works on our test batch. The investigation have also given insight into the choice of controllers and their complexity. We have however not been able to capture this knowledge in simple rules of thumb.</p>			
<i>Key words</i> Adaptive control; PID-control; Dominant pole design			
<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> 95	<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.

Innehåll

1. Inledning	3
2. Tillvägagångssätt	4
2.1 Test av transienta egenskaper	4
2.2 Processerna	4
2.3 Kriterier	5
2.4 Simuleringsresultat i tabeller	6
2.5 Simulering med störningar	6
3. PID-dimensionering	8
3.1 Dominant polplacering	8
3.2 Ziegler-Nichols självsvängningsmetod	9
3.3 Polplaceringsprincipen	9
4. Den adaptiva regulatorn Novatune	10
4.1 Adaptiv reglering	10
4.2 Parametrarna i Novatune	11
4.3 Novatune- minimal variansreglering	11
5. Brusmodellering	14
5.1 Processbrus	15
5.2 Mätbrus	15
6. Simulering	17
6.1 Simuleringsmakro för PID-regulatorn	17
6.2 Simuleringsmakro för Novatune	17
6.3 Simuleringsmakro för brus	18
7. Matlabrutiner	19
8. Övergripande slutsatser	22
8.1 Processkännedom	22
8.2 Integrerande processer	23
8.3 Oscillativa processer	25
8.4 Processer som är icke-minimum-fas	25
8.5 Regulatorernas komplexitet	25
9. Process 1	27
9.1 Inställning av Novatune	27
9.2 Ändringar i tidskonstanten	27
9.3 Dödtidsändringar	28
9.4 Bruskänslighet	29
9.5 Slutsatser	30
10. Process 2	33
10.1 Inställning av Novatune	33
10.2 Ändringar i tidskonstanten	33
10.3 Dödtidsändringar	34
10.4 Bruskänslighet	35
10.5 Slutsatser	35
11. Process 3	39

11.1	Novatunes inställning	39
11.2	Variationer i tidskonstanterna	39
11.3	Dödtidsändringar	40
11.4	Slutsatser	41
12.	Process 4	43
12.1	Inställning Novatune	43
12.2	Ändringar i systemets dämpning	44
12.3	Dödtidsändringar	45
12.4	Ändringar i systemets dämpning då det finns dödtid	46
12.5	Slutsatser	47
13.	Process 5	50
13.1	Inställning av Novatune	51
13.2	Test utan dödtid	51
13.3	Test med dödtid	51
13.4	Bruskänslighet	52
13.5	Slutsatser	53
14.	Process 6	58
14.1	Inställning av Novatune	58
14.2	Ändringar i processdynamiken	58
14.3	Dödtidsändringar	59
14.4	Slutsatser	59
15.	Process 7	62
15.1	Inställning av Novatune	62
15.2	Nominellt system utan dödtid	62
15.3	Dynamikförändringar	63
15.4	Simuleringar då systemet innehåller dödtid	64
15.5	Bruskänslighet	65
15.6	Slutsatser	66
16.	Process 8	70
16.1	Inställning av Novatune	71
16.2	Nominellt system utan dödtid	71
16.3	Dynamikförändringar	71
16.4	Test med lång dödtid	72
16.5	Slutsatser	72
17.	Process 9	77
17.1	Inställning av Novatune	77
17.2	Dynamikförändringar	78
17.3	Bruskänslighet	79
18.	Generalized Predictive Control	82
18.1	Novatune-ett specialfall av GPC	82
18.2	GPC-algoritmen	82
18.3	Kommentar	84
19.	Slutord	85
	Referenser	86
A.	Programlistning	87

1. Inledning

ABB Master är ett integrerat och distribuerat styr- och övervakningssystem för processautomation. Systemet kan delas in i fyra familjer:

- ABB MasterView operatörsstationer för kommunikation mellan operatör och system.
- ABB MasterPiece processtationer som utför själva styruppgiften.
- ABB MasterAid systemhjälpmedel.
- ABB MasterNet kommunikationssystem som knyter ihop operatörs- och processtationer till system.

I ABB MasterPiece finns flera olika typer av regulatorer. Bland annat en konventionell PID och en adaptiv regulator, Novatune. Syftet med detta examensarbete är att skapa underlag för en vägledning för vilken regulator som bäst lämpar sig för en given process.

De processer vi har undersökt skiljer sig åt både vad gäller karaktär och komplexitet. Processerna kan delas in i integrerande och icke integrerande, oscillativa och processer med monotont växande stegsvar samt processer som är icke-minimum-fas.

Vi har genomgående valt att dimensionera PID-regulatorn enligt principen för dominerande polplacering. Beteendet hos regulatorerna har undersökts när processdynamik och dödtid varierar under drift. Även konsekvenserna av olika störningar har undersökts.

Huvudsakligen baserar vi våra slutsatser på simuleringar i Simnon. Våra undersökningar har visat att PID-regulatorn är att föredra när processen är integrerande och dödtiden varierar. Om processen är av högre ordning är dock Novatune ett bättre val.

I rapporten beskrivs inledningsvis hur vi gått till väga. Kapitel 3 till 7 behandlar regulatorerna, dimensioneringsmetoder samt de matlabrutiner vi skrivit för utvärdering av simuleringarna. I kapitel 8 redogör vi för de viktigaste slutsatserna och därefter behandlas varje process för sig. Ursprungligen bestod uppgiften även av en jämförelse med en ny adaptiv regulatoralgoritm, Generalized Predictive Control (GPC). På grund av tidsbrist har behandlingen av denna inskränkts till en kort genomgång av den grundläggande algoritmen. Denna redovisas i kapitel 18. Sist ges referenser och listning av programkod.

Slutligen vill vi tacka Karl-Johan Åström och Mats Molander som handlett vårt examensarbete. Tack också till institutionen för reglerteknik där vi särskilt nämner Michael Lundh, Leif Andersson, Per Persson och Tore Hägglund. Tack även till Per-Erik Modén.

2. Tillvägagångssätt

I detta kapitel ges endast en översiktlig beskrivning av tillvägagångssättet. I kapitel 6 och 7 ges mer detaljer vad gäller simuleringarna samt de Matlab-rutiner som används vid utvärdering.

2.1 Test av transienta egenskaper

Simuleringarna går i stora drag till på följande sätt

1. Typprocess, sampeltid samt ytterligare några parametrar anges.
2. PID-regulatorn dimensioneras med hjälp av funktionen `dampidz`, som beskrivs i kapitel 3. Efter dimensioneringen kontrollerar vi att produkten $\omega T_s < 0.5$. Om inte detta krav är uppfyllt provar vi med en ny sampeltid. Novatunes parametrar ställs in med hjälp av installationsanvisningen i de fall detta är möjligt. För de fall anvisningen inte ger någon utförlig rekommendation har vi provat oss fram till en bra regulatorinställning.
3. Simulering utförs (Novatune får först adaptera). Genomgående har börvärdesändringen varit ett enhetssteg som lades på från det att systemet varit i vila. Laststörningen, som var ett negativt enhetssteg, lades på vid en tidpunkt mitt emellan två sampelpunkter.
4. Genom att ändra dödtid och tidskonstanter utan att dimensionera om regulatorerna undersöks vilka konsekvenser dessa dynamikändringar får för regleringen.

Genomgående i alla simuleringar har vi låtit Novatune adaptera innan de transienta egenskaperna undersökts. Detta har gjorts genom att processen exciterats med en fyrkantvåg till dess att parametrarna svängt in sig till stationära värden. Under testerna av de transienta egenskaperna har adaptationen stängts av.

2.2 Processerna

Jämförelser mellan regulatorerna har gjorts för 9 olika typprocesser med överföringsfunktionerna

$$G_1(s) = \frac{K}{1 + sT} e^{-sL} \quad (2.1)$$

$$G_2(s) = \frac{K}{s(1 + sT)} e^{-sL} \quad (2.2)$$

$$G_3(s) = \frac{K}{(1 + sT_1)(1 + sT_2)} e^{-sL} \quad (2.3)$$

$$G_4(s) = \frac{K}{s^2 + 2\zeta\omega s + \omega^2} e^{-sL} \quad (2.4)$$

$$G_5(s) = \frac{K}{s^2} e^{-sL} \quad (2.5)$$

$$G_6(s) = \frac{K}{s(1+sT_1)(1+sT_2)} e^{-sL} \quad (2.6)$$

$$G_7(s) = \frac{K\omega_0^2}{s(s^2 + 2\zeta\omega_0s + \omega_0^2)} e^{-sL} \quad (2.7)$$

$$G_8(s) = \frac{K(1-sT_1)}{(1+sT_2)(1+sT_3)} e^{-sL} \quad (2.8)$$

$$G_9(s) = \frac{K}{(1+s)^n} \quad (2.9)$$

2.3 Kriterier

Då principen för de båda regulatorerna skiljer sig markant är det svårt att göra riktigt rättvisa jämförelser. För varje enskild egenskap har vi därför försökt åstadkomma någon form av objektiva mått. Dessa siffror presenteras i tabellform, vilket ger en möjlighet att göra en snabb jämförelse mellan de båda regulatorerna. Vidare låter vi vid direkta jämförelser först den adaptiva regulatorn adaptera. När den sedan skaffat sig en processmodell avbryts adaptationen och de transienta egenskaperna testas på samma sätt som för PID-regulatorn. De egenskaper som i första hand testas är

1. Insvängning vid börvärdesändring (stegsvar).
2. Insvängning vid laststörning.
3. Känsligheten för mätbrus.
4. Känsligheten för processbrus.
5. Förmågan att klara dynamikförändringar i processen (robusthet).

Nedan följer en uppräknig och förklaring till de storheter vi i första hand tittat på.

Stegsvar

- Översläng, M . Utsignalens maximala värde minskat med stegets storlek.
- Stigtid, Tr . Den tid det tar för utsignalen att växa från 0.1 till 0.9 gånger stegets storlek.
- ningstid, Ts . Det minsta värde sådant att utsignalen för alla $t \geq Ts$ antar värden mellan steget-p och steget+p. Vi har genomgående valt $p=0.05$.
- IAE (Integral Absolute Error). Integralen av felets absolutbelopp.
- Iq (Integral quotient). Kvoten mellan IAE och IE (Integral Error), som utgör ett mått på hur slängigt systemet är.

Laststörning

- Maxamplitud, A_{max} . Den maximala avvikelser från börvärdet.
- Maxtid, T . Tiden från det att laststörningen inträffar tills utsignalen har sin maximala avvikelse från börvärdet.
- IAE. Integralen av felets absolutbelopp.
- Iq. Se ovan.

Mät- och processbrus

- $E(y)$. Utsignalens väntevärde.
- $V(y)$. Utsignalens varians.
- $V(y_s)$. Utsignalens varians i sampelpunkterna.
- $V(\Delta u)$. Variansen för styrsignalinkrementet $u(t) - u(t - 1)$.
- Cv. Kvoten mellan standardavvikelsen för styrsignalinkrementet och sampeltiden. Detta kan ses som ett mått på hur styrsignalen ser ut. Med tanke på ställdon och dylikt är det ofta bättre med flera små ändringar än få stora.

2.4 Simuleringsresultat i tabeller

I samband med varje simulering protokollförde vi resultaten av ovanstående kriterier i tabeller. I rapporten redovisas endast en delmängd av dessa. De tabeller som redovisas har följande utseende

Tabell 2.1 Exempel på tabell.

	STEP					LOAD				
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	Not
Reg	A	B	C	D	E	F	G	H	I	1

1. Denna not anger vilken regulator simuleringen avser.

Tabellen består av två delar som avdelas med ett vertikalt streck. Den vänstra delen innehåller resultaten av ett stegsvar och den högra resultaten av en laststörning. I tur och ordning innehåller tabellen

- A. Översläng vid stegsvar, M .
- B. Stegsvarets stigtid, Tr .
- C. Stegsvarets insvängningstid, Ts (skilj på insvängningstiden Ts och sampeltiden T_s där s är ett index.)
- D. Integralen av felets absolutbelopp för stegsvaret, IAE.
- E. Kvoten mellan IAE och IE för stegsvaret, Iq .
- F. Laststörningens maxamplitud, $Amax$.
- G. Tiden då maximala avvikelser inträffar efter en laststörning, T .
- H. Integralen av felets absolutbelopp vid laststörning, IAE.
- I. Kvoten mellan IAE och IE vid laststörningen, Iq .

Längst till höger finns en siffra som hänvisar till en not under tabellen.

2.5 Simulering med störningar

Regulatorerna testas med såväl mät- som processbrus. Mätbruset är vitt med standardavvikelsen 5 % av ett enhetssteg. Processbruset är filtrerat vitt brus. Filtret ser ut som $1/(sT + 1)$, där T är av storleksordningen 20. Hur bruset genereras beskrivs i detalj i kapitel 5. Simuleringen tillgår på följande sätt

1. Filtret initieras med ett normerat slumpstal.
2. Processen får svänga in sig under cirka fem sekunder.

3. Simulering under 200 sekunder.
4. Variansen och medelvärdet för utsignalen samt variansen för styrsignalen beräknas och utvärderas.

För att undvika att statistiska avvikelser i bruset stör jämförelserna genereras exakt samma brus i alla simuleringar.

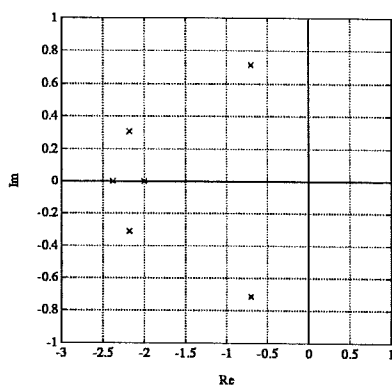
3. PID-dimensionering

3.1 Dominant polplacering

En variant av dominerande polplacering bygger på att man vill placera regulatorns poler så att det slutna systemet beter sig som

$$G(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \quad (3.1)$$

De poler som ligger närmast imaginära axeln kallas dominerande. Med en PID-regulator är det möjligt att placera tre poler. Vid dimensionering av regulator för ett system med ordning större än tre gäller det att se till att övriga poler ligger långt från de dominerande. Ett mått på dominans är polmarginalen, som definieras som kvoten mellan realdelen för den långsammaste av de snabba polerna och realdelen för de dominerande polerna. Ett exempel på dominerande polpar visas i figur 3.1



Figur 3.1 Exempel på dominerande polpar. De två högra polerna dominerar över polerna till vänster.

Kriteriet vid dimensionering är att ω_0 ska vara så stort som möjligt för en given dämpning ζ med en bibehållen dominans. Vid dimensionering av PID-regulatorn dyker det upp ett nollställe som oftast är långsamt. Det finns olika metoder att behandla detta nollställe. En metod är att söka släcka ut nollställets inverkan genom att lägga en pol mycket nära. En annan metod är att försöka flytta nollstället längre ut så att dess inverkan minskas. Detta är möjligt om man har en regulatorstruktur som ser ut som

$$u = K \left[\beta u_c - y + \frac{1}{T_i} \int_0^t (u_c - y) ds - T_d \frac{dy}{dt} \right] \quad (3.2)$$

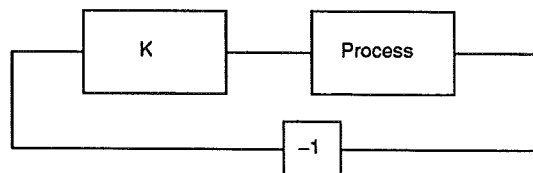
där u är styrsignalen, y processens utsignal och u_c är referenssignalen. Med hjälp av ett $\beta \leq 1$ kan man flytta ut nollstället i vänster halvplan.

Den matlabfunktion vi använder för dimensionering av PID-regulatorn heter `dompidz`. Dimensioneringen görs i diskret tid. Det dimensionerade systemets snabbhet ω_0 beror av sampeltiden. Detta innebär att vi genom att välja

en längre sampeltid kan få ett långsammare slutet system vilket i sin tur ger en robustare regulator.

3.2 Ziegler-Nichols självsvängningsmetod

Denna metod bygger på att man ur ett praktiskt experiment kan ställa in en PID-regulator. Experimentet utförs enligt en uppställning som visas i figur 3.2.



Figur 3.2 Ziegler-Nichols. Förstärkningen K ökas tills systemet självsvänger.

Ett system återkopplas negativt. Förstärkningen K ökas tills systemet självsvänger. En lämplig PID-regulator har då parametrarna $P=0.6K$, $T_i=(\text{självsvängningens periodtid})/2$ och $T_d=(\text{självsvängningens periodtid})/8$.

3.3 Polplaceringsprincipen

I en del fall tyckte vi att det var intressant att jämföra med ytterligare någon regulator. Vi dimensionerade en RST-regulator med hjälp av polplaceringsprincipen (Åström och Wittenmark, 1990). Den dimensioneringsmetod vi använt kan kort beskrivas enligt följande. Styrlagen för en RST-regulator ser ut som $Ru = Tu_c - Sy$, där u är styrsignalen, y utsignalen och u_c referenssignalen. Processen har överföringsfunktionen B/A och det önskade systemet B_m/A_m . Inför ett observerar polynom A_o med gradtal

$$\text{deg} A_o \geq 2\text{deg} A - \text{deg} A_m - \text{deg} B^+$$

där B^+ är den faktor i B som innehåller nollställen som kan förkortas med stabila poler. Dimensionering med en integrator utförs enligt följande

1. Faktorisera B och B_m som $B = B^- B^+$ och $B_m = B^- B'_m$
2. Lös den diofantiska ekvationen

$$(z-1)AR'_1 + B^-S = A_oA_m$$

där

$$\text{deg} S < 1 + \text{deg} A$$

och

$$\text{deg} R'_1 = \text{deg} A_o + \text{deg} A_m - \text{deg} A - 1$$

3. Styrlagen är

$$Ru = Tu_c - Sy$$

$$\text{där } R = B^+ R' \quad T = B'_m A_o \quad \text{och} \quad R' = (z-1)R'_1$$

4. Den adaptiva regulatorn

Novatune

4.1 Adaptiv reglering

Den grundläggande principen i all adaptiv reglering är att regulatorn själv skaffar sig en modell av den styrda processen. Med hjälp av ett antal styr- och ärvärden kan ett framtida ärvärde predikteras enligt modellen

$$\hat{y}(t+k) = \theta^T(t)\varphi(t) \quad (4.1)$$

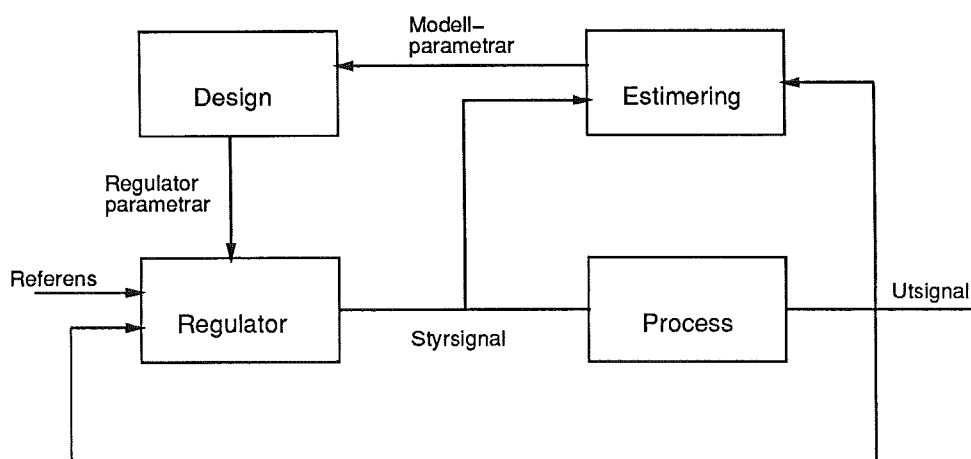
där k är det antal steg framåt i tiden där ärvärdet predikteras. θ -vektorn innehåller modellparametrarna och φ -vektorn samhörande signalvärden. Om vi nu jämför det predikterade ärvärdet med vad det verkligen blev kommer vi att få ett så kallat prediktionsfel

$$\epsilon(t) = y(t) - \hat{y}(t) \quad (4.2)$$

som dels beror på slumpmässiga störningar och dels på att inte modellen överensstämmer med verkligheten. Om man samlar ihop en mängd prediktionsfel och kvadrerar dem får man en summa

$$\sum \epsilon^2(t) \quad (4.3)$$

Genom en minstakvadratanpassning av parametrarna i modellen minimeras sedan denna summa.



Figur 4.1 Blockschema för en adaptiv regulator.

Eftersom man vill ha en regulator som är så bra anpassad som möjligt till den aktuella processen viktas termerna i kvadratsumman så att senare värden

får större betydelse än gamla. Detta görs genom att en glömskefaktor λ införs. Summan som minimeras är då

$$\sum_{t=0}^T \lambda^{T-t} \epsilon^2(t) \quad (4.4)$$

Efter att processen estimerats kan man sedan dimensionera sin regulator. I princip kan man välja vilken metod som helst. Som vi ser i figur 4.1 har ett adaptivt reglersystem två återkopplingar. Dels den långsamma adaptiva och dels den snabba där ärvärdet kopplas tillbaka.

4.2 Parametrarna i Novatune

En av nackdelarna med Novatune är att den är en mycket komplicerad regulator med ca 25 parametrar att ställa in. Vi redogör här för de viktigaste.

Parameter	Förklaring
NA	Antal parametrar i återkopplingspolynomet
NB	Antal parametrar i styrsignalpolynomet
NC	Antal parametrar i framkopplingspolynomet
KD	Prediktionshorisonten i antal sampel
T_s	Sampeltiden
PL	Pol som specificerar det önskade stegsvaret
PN	Styrsignalstraff
INT	Sätts till 1 om integralverkan önskas

Övriga parametrar är bland annat styrsignalbegränsningar, möjlighet att ange adaptionsdödband och ladda ner θ -parametrar. För utförligare beskrivningar hänvisas till installationsanvisningarna för Novatune. Vi har genomgående satt parametrarna INT=1, NC=0 och PL=0.25. Styrsignalstraffet PN har vi satt till 0 om inte annat anges. Eftersom resultatet av våra tester i stor utsträckning beror på valet av NA, NB, KD och T_s har vi varit noga med att i varje redovisad simulering ange vilken inställning som använts.

4.3 Novatune- minimal variansreglering

Den ide' som ligger till grund för Novatune kan något förenklat sammanfattas i följande två punkter

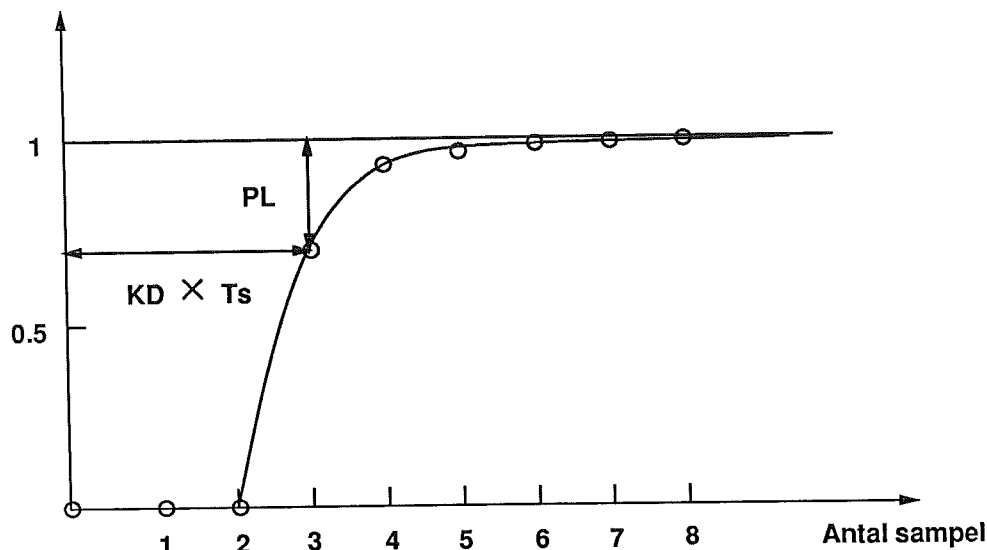
1. Bestäm $\hat{y}(t+k)$, med andra ord prediktera är-värdet k sampel framåt.
2. Kräv sedan att $\hat{y}(t+k) = u_c(t)$, där $u_c(t)$ betecknar referensvärdet vid tiden t.

Då det är önskvärt med en lugnare reglering predikteras istället ett filtrerat ärvärde

$$f(t) = (1 - pq^{-1})y(t) \quad (4.5)$$

I ovanstående ekvation är p och k två av parametrarna i Novatune. I fortsättningen av denna rapport används manualens beteckningar PL resp KD för

dessa parametrar. Med PL ges användaren möjlighet att specificera en exponentiell insvängning och med KD bestämmer man hur många sampel framåt prediktionen ska göras. Parametern KD är jämte sampeltiden den viktigaste parametern för Novatune. Innebörden av de båda parametrarna framgår av figur 4.2.



Figur 4.2 Stegsvaret som eftersträvas då $KD=3$ och $PL=0.3$.

Med hjälp av de inställda parametrarna skaffar sig Novatune en prediktionsmodell som (utan integralverkan) ser ut som

$$\hat{f}(t+k) = Bu(t) + Ay(t) + Cv(t) \quad (4.6)$$

Styrlagen blir sedan

$$\hat{f}(t+k) = (1-PL)u_c(t) \quad (4.7)$$

Genom att sätta högerledet i de båda uttrycken 4.6 och 4.7 lika kan $u(t)$ lösas ut. Polynomen i ekv 4.6 ser ut som

$$B(q^{-1}) = b_0 + b_1q^{-1} + b_2q^{-2} + \dots$$

$$A(q^{-1}) = 1 + a_0q^{-1} + a_1q^{-2} + \dots$$

$$C(q^{-1}) = c_0 + c_1q^{-1} + c_2q^{-2} + \dots$$

Det är koefficienterna i dessa polynom som estimeras i Novatune.

Den typ av adaptiv regulator som visas i figur 4.1 kallas indirekt. Detta innebär att regulatorparametrarna beräknas först efter det att processen estimerats. I Novatune estimeras koefficienterna i styrlagen direkt utan att gå vägen via skattning av processens parametrar. Denna typ av regulator kallas direkt adaptiv regulator. Ett av problemen med ovanstående strategi är att styrsignalen ofta blir onödigt stor särskilt om tiden $T_s \cdot KD$ är liten i förhållande till den styrda processens tidskonstanter. Ett sätt att åtgärda detta är, som tidigare nämnts, att specificera en lugnare insvängning enligt ekv 4.5. I Novatune används ytterligare en metod som går ut på att styrsignalen straffas. Detta görs genom att minimera förlustfunktionen

$$J = E \left\{ (\hat{f}(t+k|t) - u_c(t))^2 + \rho \Delta u(t)^2 \right\}$$

där $\hat{f}(t+k|t)$ är prediktionen av $f(t+k)$ givet information upp till tiden t . Parametern ρ ges som funktion av styrsignalstraffet PN enligt $\rho = \rho_0 2^{PN}$ där ρ_0 är av storleksordningen 10^{-6} .

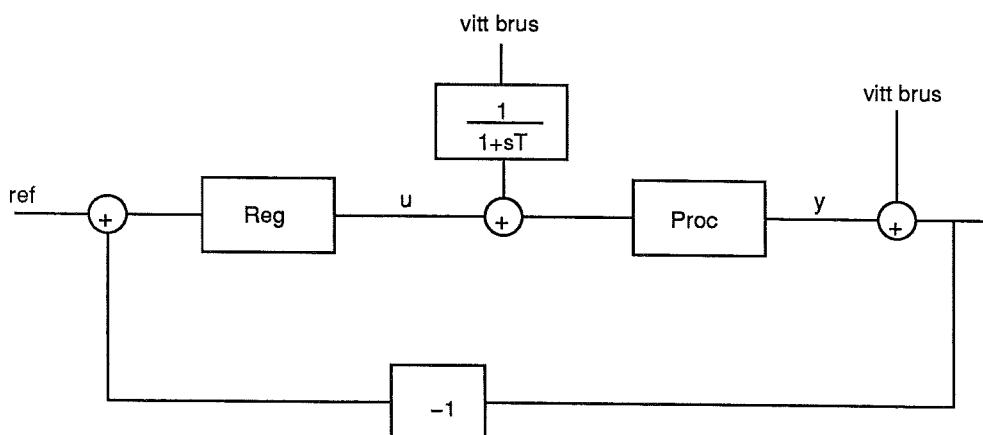
I de allra flesta fall vill man ha en regulator med integralverkan för att få bort stationära fel. Genom att uttrycka u och y som differenser $\Delta u = u(t) - u(t-1)$ och $\Delta y = y(t) - y(t-1)$ kan styrlagen med integralverkan räknas fram. Vi går inte in på detaljer utan nöjer oss med att presentera styrlagens utseende då framkopplingspolynomet NC=0

$$u_t = u_{t-1} + \frac{b_0}{b_0^2 + \rho} \left\{ (1 - PL)(u_{ct} - y_t) - b_1(u_{t-1} - u_{t-2}) - \dots - b_{NB-1}(u_{t-NB+1} - u_{t-NB}) - a_0(y_t - y_{t-1}) - \dots - a_{NA-1}(y_{t-NA+1} - y_{t-NA}) \right\}$$

5. Brusmodellering

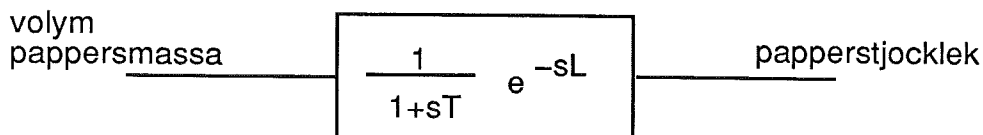
I alla industriella processer finns någon form av störning. I själva verket är detta en av de viktigaste orsakerna till att man överhuvudtaget använder regulatorer. Vi har i vårt examensarbete undersökt hur processens utsignal påverkas då systemen utsatts för följande två störningar

- Processbrus som påverkar processens tillstånd.
- Mätbrus som påverkar den återkopplade signalen.



Figur 5.1 Schematisk skiss över vår brusmodell. Mätbruset är vitt och processbruset vitt brus som filtrerats genom ett första ordningens filter.

Många industriella processer kan approximeras med första ordningens dynamik och en tidsfördröjning. Som ett konkret exempel använder vi en pappersmaskin där överföringsfunktionen från mängden pappersmassa till tjockleken på pappret utgör ett sådant system (se figur 5.2).



Figur 5.2 Överföringen från mängden pappersmassa till tjockleken på pappret i en pappersmaskin.

I ovanstående exempel kan processbruset tänkas uppstå p g a att pappersmassans egenskaper varierar med tiden. Mätbruset kan härröra från en givare som ger en brusig signal beroende på att den åldrats eller är felaktigt monterad. Även dåliga ledningar från mätobjektet till regulatorn kan ge ett bidrag. De båda brusen har oftast helt olika karakteristik. Medan processbruset varierar långsamt har mätbruset övervägande högfrekventa egenskaper. Nedan följer en beskrivning av hur vi har realiserat de båda brusen.

5.1 Processbrus

Som framgår av figur 5.1 är processbruset vitt brus som filtrerats genom ett första ordningens filter. Ett kontinuerligt filter ser ut som

$$G(s) = \frac{K}{1 + sT} \quad (5.1)$$

och dess diskreta motsvarighet blir med samplingsintervallet h och $K = 1$

$$H(q) = \frac{1 - e^{-h/T}}{q - e^{-h/T}} \quad (5.2)$$

För att få ett långsamt varierande brus valde vi en tidskonstant på 20 sekunder. Filtret kan på tillståndsform skrivas

$$\begin{cases} x(k+1) = ax(k) + be(k) \\ y(k) = x(k) \end{cases} \quad (5.3)$$

Filtrets utsignalvarians P utvecklas rekursivt enligt

$$P(k+1) = \Phi P(k) \Phi^T + R_1 \quad (5.4)$$

Vi fann det lämpligt att välja ett filter som gav oss stationära utsignalvariansen 1. Insignalen till filtret är vitt brus vilket ger $R_1 = b^2$. Med $\Phi = a$ har vi i stationärt tillstånd

$$P = a^2 P + b^2 \quad (5.5)$$

eller

$$P = \frac{b^2}{1 - a^2} \quad (5.6)$$

Ur ekvationerna 5.1 och 5.2 får vi att

$$a = e^{-h/T} \quad (5.7)$$

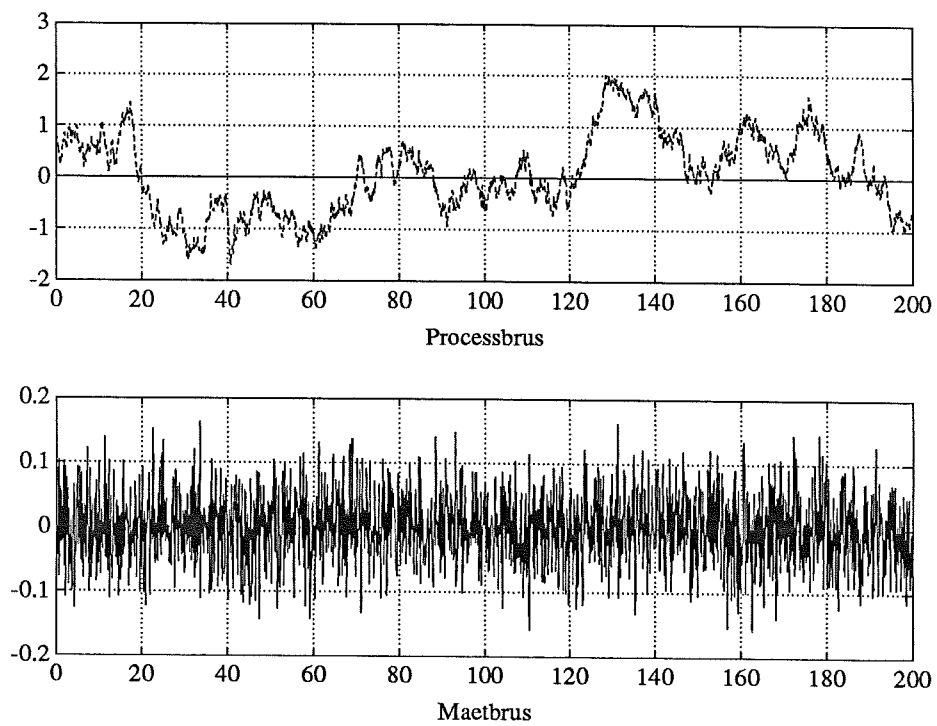
Vi satte filtrets sampeltid till 0.1 och fick $a=0.995$ med filterkonstanten $T = 20$. För att få stationära variansen 1 måste då b vara

$$b = \sqrt{1 - a^2} \quad (5.8)$$

Detta ger oss $b=0.0999$. Processbrusets utseende framgår av figur 5.3.

5.2 Mätbrus

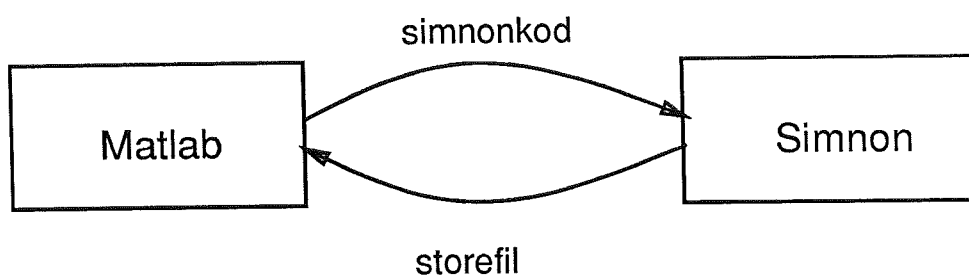
Det mer högfrekventa mätbruset modellerades med vanligt vitt brus. Standardavvikelsen sattes till 0.05. För att inte statistiska avvikelser skulle störa våra jämförelser såg vi till att alltid generera samma brus genom att initiera slumpalsgeneratorn med en konstant. Mätbruset visas i figur 5.3.



Figur 5.3 De båda brusen. Överst processbruset och i undre diagrammet mätbruset.

6. Simulering

Under detta examensarbete har vi haft tillgång till ett simuleringspaket som gett oss möjlighet att utföra själva simuleringarna i Simnon. Resultaten från simuleringen lagras i en fil som överförs till Matlab där vi skrivit egna rutiner för utvärdering av resultaten från simuleringarna. Grundstommen i våra matlabrutiner utgörs av två simuleringmakron, ett för simulering med PID-regulator och ett för simulering med Novatune. Nedan ges en beskrivning av dessa båda makron. Övriga matlabfunktioner behandlas i ett separat kapitel.



Figur 6.1 Matlab och simnon kan kommunicera. Matlab styr Simnon och läser efter avslutad simulering in storefilen.

6.1 Simuleringsmakro för PID-regulatorn

Innan man exekverar makrot för PID-simulering definierar man den process för vilken man önskar testa regulatorerna. Processens överföringsfunktion $B(s)/A(s)$ anges med vektorerna $A(s)=[a_0s^n + a_1s^{n-1} + \dots + a_n]$ och $B(s)=[b_0s^m + b_1s^{m-1} + \dots + b_m]$. Vidare anges önskad sampeltid och längd på simuleringen. Scriptet utför sedan i tur och ordning följande

- Anrop av `dompidz` som dimensionerar en PID-regulator enligt principen för dominerande polplacering. Dimensioneringen görs i diskret tid. Detta anrop är möjligt att utesluta om man vill undersöka vad som händer när processen ändrar sig utan att ny regulatordimensionering görs.
- Anrop av funktion som skapar simnonkod av ingångsvärdena, kör simuleringen och laddar över storefilen från Simnon till Matlab.
- Plottning av storefilen samt inhämtning av IAE och IE, som är kontinuerliga integraler vilka beräknas i Simnon.

Därefter låter vi Matlab beräkna maximal översläng, stigtid, lösningstid etc. Det bör påpekas att funktionen `dompidz` även ger de kontinuerliga regulatorparametrarna K , T_i och T_d som motsvarar de RST-polynom som erhölls vid dimensioneringen.

6.2 Simuleringsmakro för Novatune

På samma sätt som ovan definieras processens A- och B-polynom. Dessutom måste vi här även ange vilken regulator som ska användas. Följande parametrar skall alltså ges värden

- NA och NB.
- KD och sampeltiden, T_s .
- INT och PN.

Innebörden av dessa regulatorparametrar framgår av kapitel 4. Simulering görs sedan i två etapper. Först tillåts Novatune adaptera under en bestämd tid varefter parametrarna fryses. Efter detta fortsätter simuleringen (utan adaption) precis på samma sätt som för PID-regulatorn.

6.3 Simuleringsmakro för brus

Utöver de två ovan beskrivna makro-filerna finns ett makro som gör det möjligt att undersöka hur regulatorerna beter sig då processen utsätts för någon form av störning. Man kan här välja mellan två olika störningar.

- Ett långsamt varierande processbrus som adderas till styrsignalen (tillstånd 1 störs).
- Ett mer högfrekvent mätbrus som adderas till den återkopplade utsignalen.

I kapitel 5 om brusmodellering beskrivs hur de båda störningarna genereras.

7. Matlabrutiner

Till vår hjälp vid simulering och utvärdering skapade vi ett antal matlabrutiner. I detta kapitel beskrivs de viktigaste av dessa i stora drag. Sist i kapitlet nämner vi de funktioner som möjliggör kommunikation mellan Matlab och Simnon. Dessa funktioner är skrivna av Per Persson och Michael Lundh vid institutionen för reglerteknik. Koden för våra egna rutiner finns i appendix A.

c2rst.m

Denna funktionen omvandlar en PID-regulator med parametrarna K , T_i , T_d till diskreta R,S,T-polynom. I den dimensioneringsprocedur, dompidz, som vi använder utgår man från följande diskreta form

$$u = K(\beta u_c - y + \frac{T_s}{T_i(1-q^{-1})}(u_c - y) - \frac{T_d}{T_s}(1-q^{-1})\frac{1-\alpha}{1-\alpha q^{-1}}y) \quad (7.1)$$

där

$$\alpha = \begin{cases} 0, & \text{om } \frac{T_d}{T_s} < 5; \\ \frac{\frac{T_d}{T_s}-5}{\frac{T_d}{T_s}+5}, & \text{annars.} \end{cases}$$

Med denna regulatorbeskrivning blir återkopplingsoperatören

$$\frac{S(q^{-1})}{R(q^{-1})} = K(1 + \frac{T_s}{T_i(1-q^{-1})} + \frac{T_d}{T_s}(1-q^{-1})\frac{1-\alpha}{1-\alpha q^{-1}}) \quad (7.2)$$

vilket i framåtskift ger oss R-och S-polynomen

$$R(q) = q^2 - (1 + \alpha)q + \alpha \quad (7.3)$$

$$S(q) = K((1 + \frac{T_s}{T_i} + \frac{T_d}{T_s}(1-\alpha))q^2 - (1 + \alpha + \frac{T_s}{T_i}\alpha + \frac{2T_d}{T_s}(1-\alpha))q + \alpha + \frac{T_d}{T_s}(1-\alpha)) \quad (7.4)$$

Ur ekvation 7.1 fås sambandet

$$\frac{T(q^{-1})}{R(q^{-1})} = K(\beta + \frac{T_s}{T_i(1-q^{-1})}) \quad (7.5)$$

varur $T(q)$ löses ut till

$$T(q) = \frac{K}{T_i}((\beta T_i + T_s)q^2 - ((\beta T_i + T_s)\alpha + \beta T_i)q + \alpha \beta T_i) \quad (7.6)$$

c2dt.m

Omvandlar kontinuerlig PID-regulator till diskret RST-regulator med hjälp av Tustin-approximation.

th2rst.m

Omvandlar Novatunes θ -vektor till motsvarande R,S,T-polynom.

Omräkningen sker m h a styrlagen för Novatune (se kapitel 4). Inargument till funktionen är förutom θ -vektorn

- antal skattade parametrar NA och NB.
- styrsignalstraffet PN.
- integralverkan INT (1 eller 0).

mtr.m

Denna funktion används vid analys av en genomförd simulering. Efter att store-filen laddats över till Matlab beräknas

- Maximal översläng.
- Lösningstid.
- Stigtid för stegsvaret.
- Maximal avvikelset från börvärdet vid laststörning.
- Tidpunkten för maximala avvikelset från börvärdet vid laststörning.

stepr.m

Plottar öppna systemets stegsvar. Används som hjälp vid val av regulatorparametrar för Novatune.

noiseval.m

Används vid analys efter genomförd brussimulering. Följande storheter beräknas

- Utsignalens medelvärde.
- Utsignalens varians.
- Utsignalens varians i sampelpunkterna.
- Variansen för styrsignalinkrementet Δu .

xpl.m

Plottar referens, utsignal och styrsignal samt förser plotten med önskad bildtext.

thplot.m

En variant av förra funktionen. Här ges möjlighet att spara θ -parametrarna under en simulering.

polpd.m

För vissa av våra processer ville vi testa någon annan typ av regulator än PID och Novatune. Denna funktion dimensionerar en RST-regulator enligt principen för polplacering. Dimensioneringsmetoden är direkt hämtad ur Åström och Wittenmark 1990.

pidsim.m

Denna funktion genomför själva simuleringen för PID-regulator. I tur och ordning utförs följande.

- R,S,T omvandlas till ett diskret simnonsystem.
- Den kontinuerliga överföringsfunktionen $B(s)/A(s)$ omvandlas till ett kontinuerligt simnonsystem.

- En fil som anger kopplingar av delsystem skrivs.
- Ett startmakro skrivs där syst och store görs.
- Ett simuleringsmakro skrivs. Här görs simuleringen samt omvandling av resultatfilen till en fil som är möjlig att läsa in till Matlab.
- Startmakro och simuleringsmakro exekveras.
- Den läsbara storefilen läses in till matlab och överförs till matrisen tryu, där t=tid r=referens y=utsignal u=styrsignal.

adsim.m

Simuleringsfunktion för den adaptiva regulatorn. Utifrån ges möjlighet att sätta regulatorparametrarna, m a o koppla in den regulator som innehåller önskat antal parametrar (NA och NB). I övrigt i princip samma funktion som pidsim.

Beskrivning av funktioner för kommunikation Matlab-Simnon

- yusimnonload.m
Överför en läsbar storefil från Simnon till Matlab.
- rsimcom.m
Överför ett simnonkommando från Matlab till Simnon.
- rsimval.m
Funktion för inläsning av ett värde i Simnon till Matlab.
- simnon.m
Omvandlar ett linjärt tidsinvariant system i Matlab till dess ekvivalenta simnonssystem.

8. Övergripande slutsatser

En svårighet vid jämförelser mellan Novatune och en PID-regulator är att regulatorerna är så olika. Om man som operatör ställs inför problemet att välja regulator bör man ställa frågor av typen

- Hur mycket arbete är rimligt att lägga ner på idrifttagningen?
- Är processen enkel eller är den mer komplex?
- Ändrar sig processens egenskaper med tiden och i så fall hur mycket?
- Är processen lätt att identifiera, så att tidskonstanter och dödtid kan bestämmas med stor noggrannhet?
- Vilka krav ställs på det slutna systemet och är dessa krav överhuvudtaget möjliga att uppnå med tanke på den process som ska styras?

I detta kapitel sammanfattar vi de viktigaste iakttagelserna vi gjort under detta examensarbete. Under respektive rubrik redogör vi för de grundläggande slutsatser vi kommit fram till vad gäller val av regulatortyp. Vi diskuterar även hur regulatorerna bör ställas in i de olika fallen.

8.1 Processkännedom

En PID-regulator kan dimensioneras på många olika sätt. Vi har genomgående använt metoden med dominerande polplacering. Oavsett vilken dimensioneringsmetod som används krävs det en god kännedom om processen. I fallet med en adaptiv regulator räcker det oftast med ett stegsvar för att den ska kunna ställas in och ge acceptabel reglering även om processdynamiken inte var den vi förväntade oss. Något förenklat kan man tänka sig att man har fyra situationer.

- A. Både dynamik och dödtid är kända.
- B. Processdynamiken känd men dödtiden varierar eller svår att bestämma.
- C. Processdynamiken varierar eller är svår att bestämma medan dödtiden är känd.
- D. Varken dynamik eller dödtid är kända.

I fall A där både dödtid och dynamik är kända styr processens komplexitet valet av regulator. Det finns ingen anledning att välja en komplicerad regulator om man har en enkel process. Hög ordning på processen i kombination med dödtid ger ofta bättre resultat med en mer komplex regulator.

I fallet B har våra undersökningar visat att man bör se upp med Novatune, som lätt blir instabil när dödtiden varierar. Framför allt när dödtiden ökar visar Novatune stor känslighet. PID-regulatorn klarar dödtidsvariationer klart bättre.

När processdynamiken är okänd eller varierar men dödtiden är känd (fall C) råder det ingen tvekan om att Novatune bör väljas. Tack vare adaptationen kan processdynamiken tillåtas variera inom vida gränser.

I det sista fallet när varken dynamik eller dödtid är känd kan vi inte ge någon generell regel för vilken regulator som passar bäst. Ett sätt kan vara att använda Novatune och överskatta dödtiden. I ett sådant fall bör man dock

akta sig för att skatta allt för många parametrar eftersom detta kan leda till att Novatune får svårt att "hitta processen". Exempel på detta visas i figur 12.2.

8.2 Integrerande processer

Några av de problem som dyker upp när en process är integrerande illustrerar vi med hjälp av ett system med överföringsfunktionen

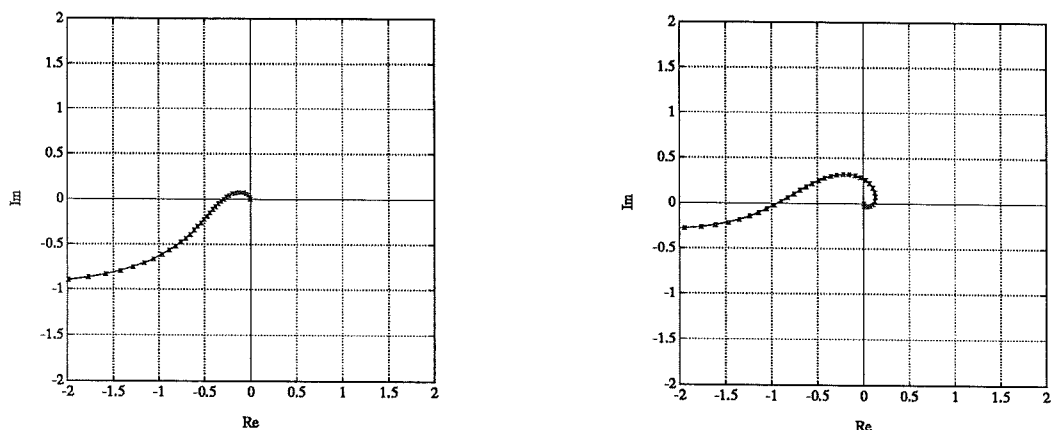
$$G(s) = \frac{1}{s(s+1)} \quad (8.1)$$

som kan tänkas beskriva ett DC-servo. Vid dimensionering av PID-regulator för denna process kan man ha en hög regulatorförstärkning innan det slutna systemet blir instabilt. Med en hög förstärkning elimineras en laststörning snabbt. Om vi istället antar att systemet innehåller dödtid måste regulatorförstärkningen sänkas för att stabiliteten ska bibehållas, vilket medför att en laststörning får stora konsekvenser. Tabell 8.1 innehåller de regulatorparametrar som erhöles med den valda metoden för dimensionering av PID-regulatorer. Resultatet ges för två olika sampeltider. Tabellen visar att förstärkningen sjunker drastiskt när en liten dödtid finns med vid dimensioneringen.

Tabell 8.1 Regulatorparametrar för PID.

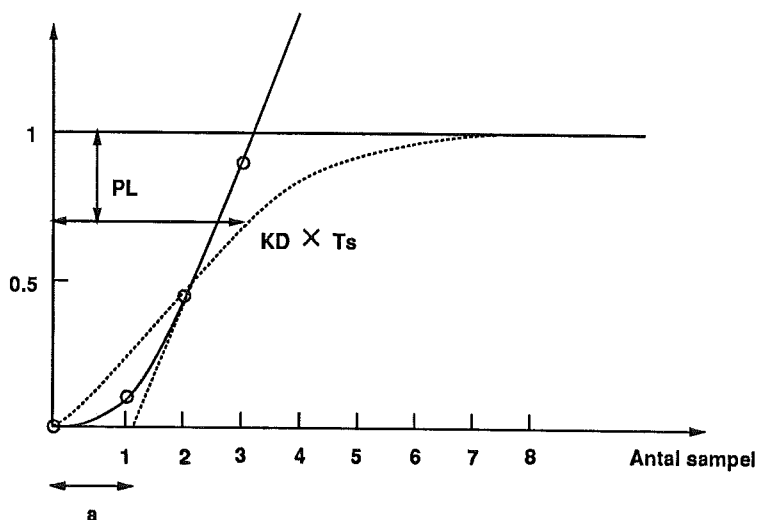
	Sampeltid	Dödtid	K	Ti	Td
PID	0.1	0.0	16.2	0.71	0.27
PID	0.1	0.1	5.5	1.4	0.43
PID	0.2	0.0	5.0	1.3	0.41
PID	0.2	0.1	2.8	1.9	0.51

Om man är helt säker på att systemet är fritt från dödtid bör man naturligtvis dimensionera sin regulator därefter. Det kan dock vara svårt att ur ett stegsvar se om det finns en liten dödtid eller ej. Om man dimensionerar sin regulator utifrån antagandet att dödtid saknas räcker det att en liten dödtid snyger sig in i systemet för att det ska bli problem med stabiliteten. Det bästa sättet att undvika dessa problem är att inte sampla alltför ofta. En längre sampeltid ger ett långsammare slutet system men vi får i gengäld en robustare regulator. I figur 8.1 visas ett extremfall där sampeltiden satts till 0.1 och en dödtid på 0.1 sekunder dykt upp i systemet.



Figur 8.1 Nyquistkurvor för slutet system med PID-regulator. Vänster figur visar system 8.1 med en PID-regulator och den högra figuren visar samma system med en dödtid på 0.1 sekunder.

Då installationsanvisningen för Novatune saknar utförliga rekommendationer för hur regulatorn ska ställas in för integrerande processer har vi provat oss fram och funnit att sampeltiden inte bör väljas för lång. Detta får till följd att prediktionshorisonten KD i de flesta fall måste väljas större än 1 även när systemet saknar dödtid. Vi illustrerar detta i figur 8.2.



Figur 8.2 Inställning av Novatune för integrerande process. Den heldragna linjen är det öppna stegsvaret för den styrda processen och den streckade är det slutna systemets stegsvar.

Genom att prova oss fram har vi kommit fram till att vi får en bra regulator om Novatune ställs in enligt den idé som visas i figur 8.2. Asymptoten till stegsvaret skär tidsaxeln i punkten a . Vår rekommendation är att välja $KD \cdot T_s \approx 1.4 \cdot a$. Kräver vi att det slutna systemet ska vara snabbare är risken stor att systemet blir instabilt.

Jämför man Novatune med en PID-regulator när båda regulatorerna är dimensionerade för en integrerande process med dödtid har våra simuleringar visat att Novatune är betydligt känsligare för variationer i dödtiden.

De slutsatser vi hittills redogjort för gäller i än högre grad för dubbelintegratorn. Finns det en dödtid i systemet måste regulatorförstärkningen vara liten för att vi ska få ett stabilt system. Vi anser att det krävs större

komplexitet hos regulatorn än man kan uppnå med en PID-regulator för att effektivt reglera denna process då det finns dödtid i systemet. Anledningen till detta är att det samplade systemet innehåller fler poler än det går att placera med en PID-regulator.

8.3 Oscillativa processer

För dessa processer gäller ett liknande resonemang som för integrerande processer. Även om systemet är fritt från dödtid kan KD väljas större än 1 för att vi ska kunna sampla oftare. Detta är särskilt viktigt om systemet är dåligt dämpat. Vi är tveksamma till om dominerande polplacering är den bästa dimensioneringsmetoden när processen har komplexa poler. Det är svårt att dimensionera en PID-regulator så att det slutna systemet beter sig som önskat om det finns en dödtid i systemet. I flera fall fick vi negativa parametrar med den dimensioneringsmetod vi använde till vår PID-regulator. Om systemets dämpning ändras är Novatune att föredra eftersom den anpassar sig till den nya processdynamiken.

8.4 Processer som är icke-minimum-fas

En process som har ett nollställe i höger halvplan kallas icke-minimum-fas och kännetecknas av att stegsvaret till en början går åt fel håll. Detta är den enda process där vi funnit att Novatune är mindre känslig för dödtidsvariationer än PID-regulatorn. Inställningen av Novatune valdes så att stegsvaret nått 70% av slutvärdet vid tiden $KD \cdot T_s$. Om man vill ha en PID-regulator som är robust mot ökningar i dödtiden gäller det att inte välja en alltför snabb regulatorinställning.

Störst problem visade det sig bli när nollstället var sämre dämpat än vad som modellerades. Novatune hade inga problem men systemet med PID-regulator blev snabbt instabilt. Vi fann att den bästa lösningen var att dimensionera PID-regulatorn efter det "värsta fallet", alltså med ett nollställe nära imaginära axeln. Om sedan nollstället skulle flytta sig ut i höger halvplan påverkar inte detta systemet nämnvärt.

8.5 Regulatorernas komplexitet

De flesta processer kan reduceras till en enklare modell. I de fall detta ej är möjligt eller då man har speciellt höga krav på regleringen kan man vinna en hel del på att prova en mer komplex regulator än PID. Vi har i några fall använt en RST-regulator av högre ordning som jämförelse. Regulatorn har dimensionerats med den tidigare nämnda polplaceringsprincipen och resultatet har varit tillfredsställande. Emellertid krävs det mer av en processingenjör vid dimensionering av en mer komplex regulator. Kanske skulle man kunna tänka sig ett användargränssnitt som ser ut som i figur 8.3.



Figur 8.3 Gränssnitt mellan operatör och regulator.

En operatör kan välja vilken typ av regulator som passar bäst. Väljs PID anger operatören K , T_i , T_d och samtidigt varefter parametrarna internt räknas om till RST-form. Är processen mer komplicerad kan operatören ges möjlighet att själv direkt ange sina R-, S- och T-polynom.

9. Process 1

Detta kapitel sammanfattar resultaten från den första processen. Den har överföringsfunktionen

$$G(s) = \frac{K}{1 + sT} e^{-sL} \quad (9.1)$$

Detta är en ofta använd modell för processer med ett monotont växande stegsvar. Här är några exempel

- Pappersmaskin där mängden pappersmassa utgör insignal och utsignalen är papperstjockleken.
- Överföringsfunktionen från ventilpådrag till nivå i en tank. Dödtiden kan till exempel uppstå på grund av långa ledningar.
- Dynamiken för en luftkonditioneringsanläggning.
- Överföringsfunktionen från spänning till varvtal i en motor. I detta fall kan man anse att dödtiden \ll tidskonstanten.

Processens karaktär varierar med kvoten L/T . I vårt fall har vi valt $L/T = 1$ som nominell process. Överföringsfunktionen ser då ut som

$$G(s) = \frac{1}{1 + s} e^{-s} \quad (9.2)$$

9.1 Inställning av Novatune

Eftersom vi anser dödtiden i processens stegsvar lätt att läsa av, samtidigt som det klart framgår att systemet är av första ordningen valde vi Novatune-parametrar enligt följande

1. $NA=1$.
2. $NB=NA+KD-1$.
3. Ur stegsvaret valdes $KD=3$ och $KD=2$.
4. I fallet då $KD=3$ valdes sampeltiden, $T_s=0.5$ (halva dödtiden), och i fallet $KD=2$ satte vi $T_s=1$.

9.2 Ändringar i tidskonstanten

Då den verkliga processen stämmer väl överens med modellen ger PID-regulatorn bra resultat. När processens tidskonstant ökar 5 gånger visar PID-regulatorn som väntat klart sämre prestanda (se tabell 9.1). Vi får en lång insvängningstid och kvoten IAE/IE är stor, vilket indikerar att vi har dålig dämpning. Görs ny dimensionering ger PID-regulatorn emellertid värden som är fullt jämförbara med Novatunes. Med $KD=3$ och $T_s=0.5$ ger Novatune en utmärkt reglering för det nominella systemet men får problem redan då tidskonstanten ökar 2-3 gånger. Om vi istället valde inställning enligt fall 2 ($KD=2$, $T_s=1$) blev visserligen regleringen av det nominella systemet något sämre men den hade inga problem att klara såväl 10 gångers ökning som minskning i tidskonstanten.

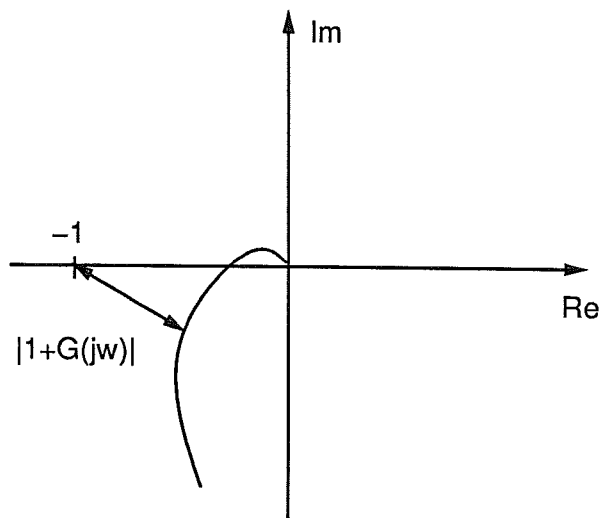
Tabell 9.1 Simuleringar då tidskonstanten ökats från 1 till 5.

	STEP					LOAD				Not
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	
PID	0.37	3.6	23	6.8	3.8	0.41	4.9	3.7	2.1	1
PID	0.04	2.1	3.6	2.3	1.0	0.26	3.0	0.85	1.0	2
Nova	0.0	1.6	4.2	2.8	1.0	0.26	2.5	0.59	1.0	3

1. PID-regulatorn dimensionerad för det nominella systemet i ekv 9.2. $T_s=0.3$.
2. PID-regulatorn dimensionerad för det verkliga systemet, alltså med en tidskonstant på 5 sekunder och döttid på 1 sekund. $T_s=0.3$.
3. Novatune med $NA=1$, $NB=2$, $KD=2$, $T_s=1$

9.3 Dödtidsändringar

I tabell 9.2 har dödtiden i det nominella systemet dubblats, vilket får stora konsekvenser för båda regulatorerna. PID-regulatorn klarar sig visserligen men vi får en maximal känslighet på 3.6, en fasmarginal på 29° och en insvängningstid på mer än 20 sekunder. Begreppet känslighet förklaras i figur 9.1.



Figur 9.1 I ett nyquistdiagram definieras känslighet som det inverterade avståndet från kurvan till -1. Den maximala känsligheten är alltså inversen av minsta avståndet till -1.

Med parameterinställning enligt ovan får Novatune stora svårigheter att hitta processen redan vid mycket små dödtidsökningar. Vid dubbla dödtiden måste ny dimensionering till. Vi satte sampeltiden = dödtiden och fick en regulator som visar lika bra prestanda som PID-regulatorn. PID-regulatorn har större tolerans mot sådana variationer, men är i gengäld långsammare. Vi provade med ett antal olika sampeltider, men resultatet blev inte bättre. Novatune kan göras robustare om man överskattar dödtiden vid inställningen. I tabellen nedan betecknar phm fasmarginalen och gm amplitudmarginalen.

Tabell 9.2 Dödtiden i det nominella systemet (ekv 9.2) har dubblats.

	STEP					LOAD							
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	S	phm	gm	Not
PID	0.6	1.4	20	6.3	3.5	0.9	4.3	5.3	2.9	3.6	29°	1.4	1
PID	0.06	2.5	3.5	7.6	1.1	0.9	4.3	3.7	1.0	1.6	64°	2.9	2
Nova	0.0	3.2	7.6	5.4	1.0	0.9	4.6	3.9	1.0	3.2	-	-	3

1. PID-regulatorn dimensionerad för det nominella systemet (ekv 9.2). $T_s=0.3$.
2. PID-regulatorn dimensionerad för det verkliga systemet. $T_s=0.3$.
3. Novatune med $NA=1$, $NB=2$, $KD=2$. Sampeltiden måste ökas för att regulatorn över huvudtaget ska fungera. Vi satte sampeltiden lika med dödtiden, $T_s = 2$. Eftersom vi fick fler än ett värde på amplitud- och fasmarginalen har vi här inte angett dessa.

Det kan i vissa fall vara av intresse att undersöka vad som händer då processen går mot en ren dödtid, det vill säga när dödtiden \gg tidskonstanten. Vi provade med en tidskonstant lika med 0.5 och en dödtid på 10 sekunder. Figur 9.2 visar denna simulering.

9.4 Bruskänslighet

Vi har inte kunnat påvisa några stora skillnader vad gäller känslighet för störningar. Figur 9.3 visar en simulering där vi lagt på ett processbrus på det nominella systemet. PID-regulatorn kommer att visa större känslighet om dynamiken ändras sig utan att ny dimensionering görs. Som väntat minskar utsignalvariansen vid snabbare sampling (se tabell 9.3 och 9.4).

Tabell 9.3 Processbrus på ekv 9.2

	E(y)	V(y)	V(ys)	V(Δu)	Cv	Not
PID	-0.0093	0.073	0.072	0.0044	0.22	1
PID	-0.0084	0.063	0.063	0.0007	0.26	2
Nova	-0.0034	0.074	0.083	0.094	0.31	3

1. PID-regulator med $T_s=0.3$.
2. PID-regulator med $T_s=0.1$.
3. Novatune med $NA=1$, $NB=2$, $KD=2$ och $T_s=1.0$.

Tabell 9.4 Mätbrus på ekv 9.2

	E(y) 10^{-3}	V(y) 10^{-3}	V(ys) 10^{-3}	V(Δu) 10^{-3}	Cv	Not
PID	-0.24	0.45	0.44	8.8	0.31	1
PID	0.39	0.20	0.20	43	2.1	2
Nova	-2.1	0.96	1.3	8.0	0.09	3

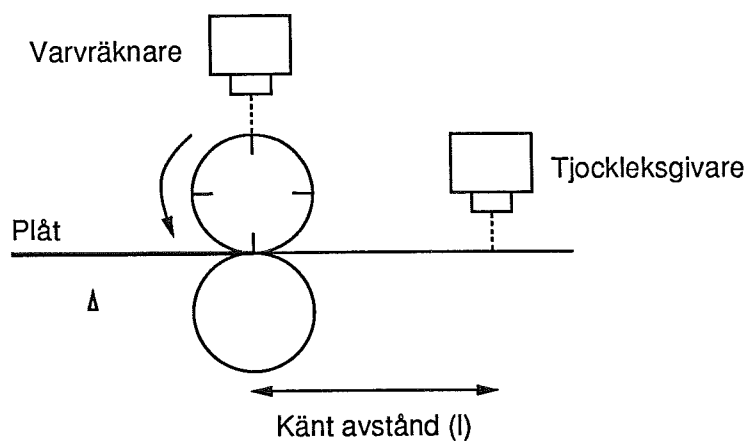
1. PID-regulator med $T_s=0.3$.
2. PID-regulator med $T_s=0.1$.
3. Novatune med $NA=1$, $NB=2$, $KD=2$ och $T_s=1.0$.

9.5 Slutsatser

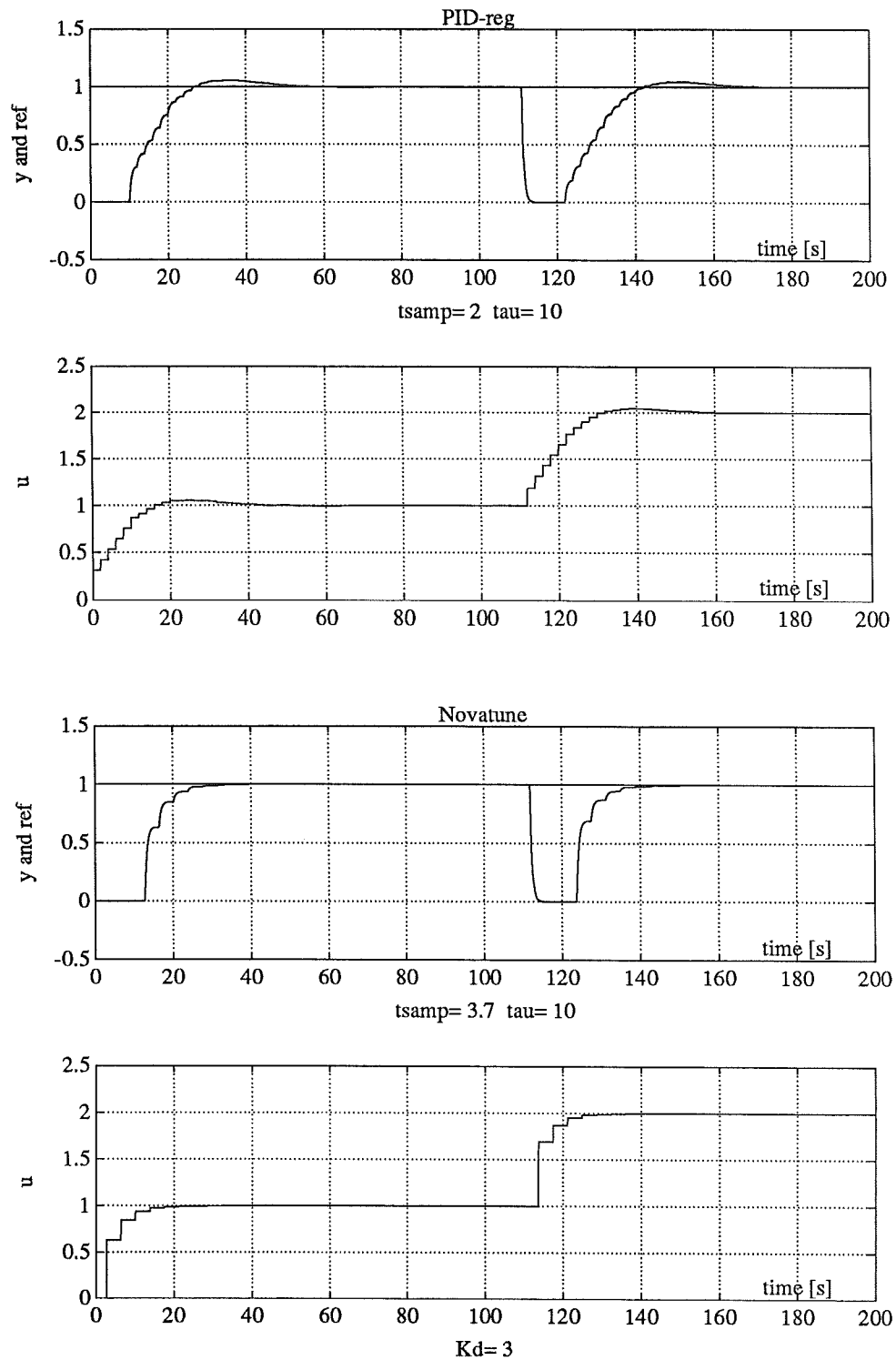
Om följande gäller för den process man önskar reglera är Novatune ett bra val

- Man har en känd dödtid som endast varierar lite.
- Dynamiken ändrar sig under drift.

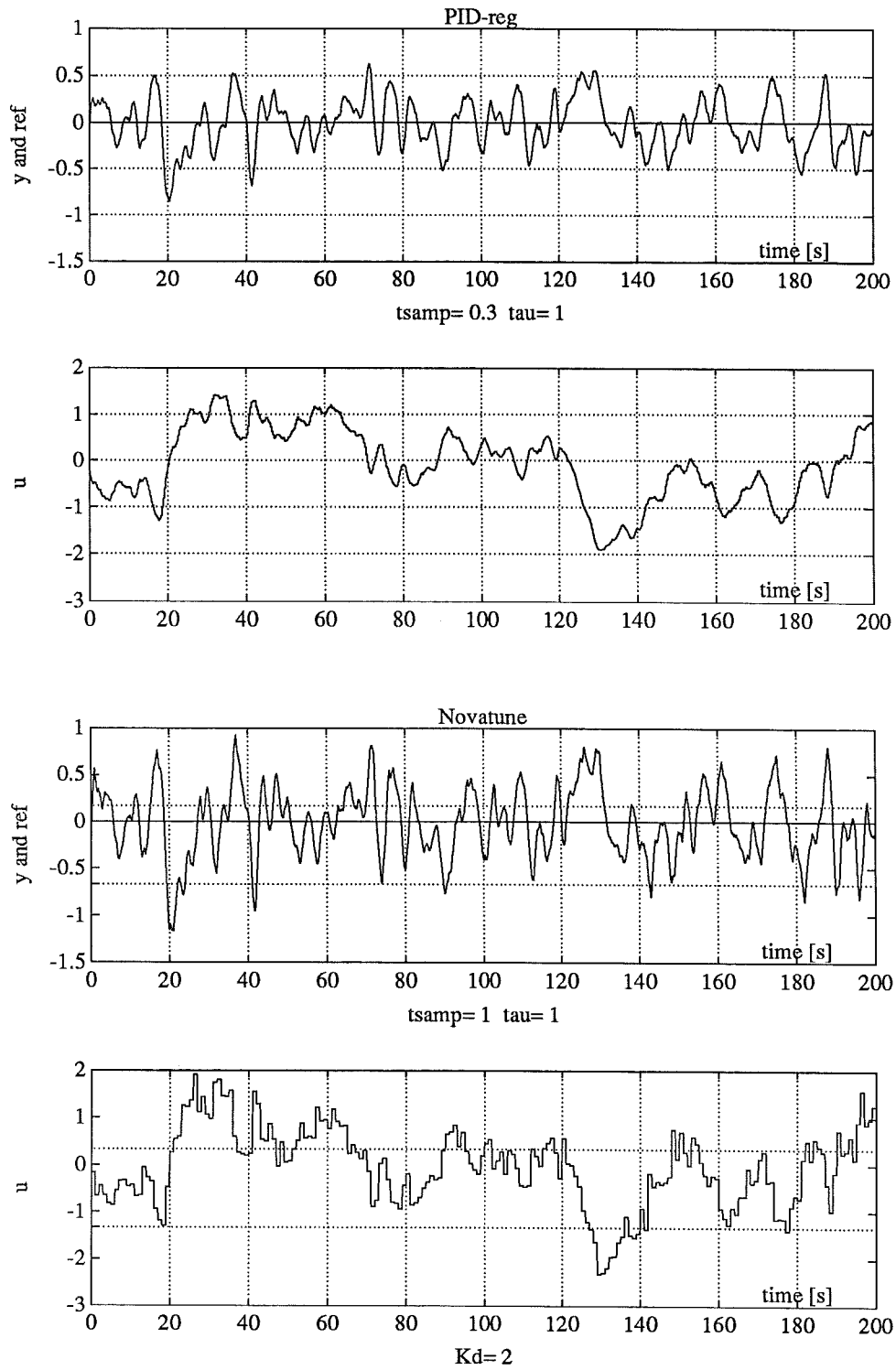
Då dödtiden varierar bör man välja PID-regulatorn. Denna reglerar visserligen sämre om processen inte är helt känd eller ändrar sig, men klarar i gengäld dödtidsökningar på upp till 3 gånger utan att bli instabil. Ett sätt att klara dödtidsvariationer med Novatune kan vara att istället för att sampla med ett fixt tidsintervall låta sampeltiden bero av dödtiden. I ett valsverk kan man till exempel låta sampeltiden vara en funktion av hur fort valsarna snurrar.



Figur 9.2 Sampeltiden beror av hur fort valsarna snurrar.



Figur 9.3 Simulering med en process med tidskonstanten 0.5 och 10 sekunders döttid. Överst PID-regulator med sampeltiden 2 sekunder och i undre figuren Novatune med $KD=3$, $T_s=3.7$, $NA=1$, $NB=3$ samt styrsignalstraff, $PN=18$. En annan variant är att sätta $PN=0$ och välja längre sampeltid. I fallet med Novatune svänger processens utsignal in sig till en stationär nivå mellan varje sampel. Detta indikerar att man skulle kunna sampla oftare. PID-regulatorn har större tolerans mot variationer i döttiden medan Novatune är svår att ställa in då vi har riktigt långa döttider.



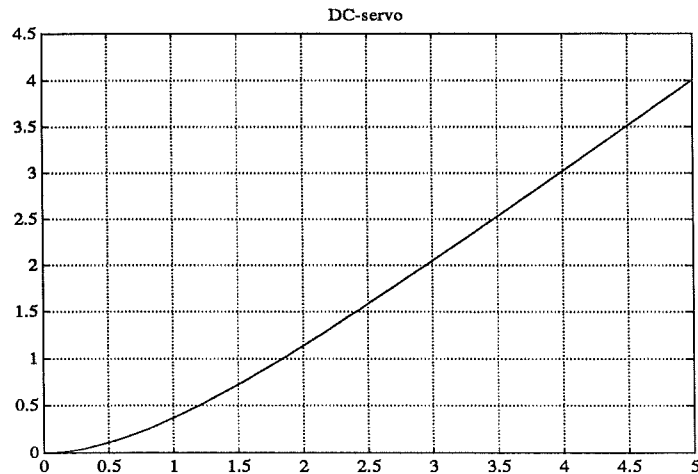
Figur 9.4 Processbrus på det nominella systemet. Processbruset är vitt brus som filterats genom första ordningens filter med tidskonstanten 20 sekunder. Båda regulatorerna är dimensionerade för det nominella systemet. Överst PID med $T_s=0.3$ och underst Novatune med grundinställning ($NA=1, NB=2, KD=2, T_s=1$).

10. Process 2

En integrerande process av andra ordningen kan beskrivas med överföringsfunktionen

$$G(s) = \frac{K}{s(1 + sT)} e^{-sL} \quad (10.1)$$

Ett exempel på denna typ av process är ett DC-servo, där $G(s)$ är överföringsfunktionen från spänning till vinkel. Ett stegsvar för det öppna systemet ser ut som i figur 10.1.



Figur 10.1 Stegsvår för DC-servo

10.1 Inställning av Novatune

I installationsanvisningen saknas en bra rekommendation för denna typ av process. Ur stegsvaret får vi dock följande information:

- Processen är integrerande.
 - Processen är av ordning två eller större eftersom stegsvaret smyger igång.
- med hjälp av dessa iakttagelser valde vi $NA = 2$ (processens ordningstal) och $KD = 2$. Slutligen valdes $NB = NA + KD - 1 = 3$ enligt installationsanvisningen. Om sampeltiden väljs enligt den metod som beskrivs i avsnitt 8.2 får vi $T_s = 0.7$. Generellt anser vi att KD bör vara större än 1 när vi har en integrerande process, även om systemet inte innehåller någon dödtid. Detta för att vi ska kunna välja en kortare sampeltid. En kortare sampeltid ger snabbare reglering av den nominella processen.

10.2 Ändringar i tidskonstanten

Vi valde det nominella systemet

$$G(s) = \frac{1}{s(1 + s)} \quad (10.2)$$

vilket kan sägas vara ett normerat DC-servo. Redan vid en tidskonstantökning på tre gånger blev PID-regulatorn instabil, medan Novatune förblev stabil vid sju till åtta gångers ökning. Denna gräns kunde ökas något med en längre sampeltid. Parameterinsvängningen gick mycket långsamt när tidskonstanten ökade. Då tidskonstanten minskas 10 ggr blir processen i princip en integrator. Detta test medförde inga problem för någon av regulatorerna.

10.3 Dödtidsändringar

När vi provade att förse DC-servot med en dödtid blev båda regulatorerna snabbt oandvändbara om inte ny dimensionering gjordes. Vid test av dödtidsvariationer valde vi i stället en nominell process med överföringsfunktionen

$$G(s) = \frac{1}{s(1+s)} e^{-s} \quad (10.3)$$

Efter att ha dimensionerat för denna process ökades dödtiden till det dubbla. PID-regulatorn klarar en dubbling av dödtiden, medan Novatune får problem redan vid en halv gångs ökning. För att få en fungerande Novatune tvingas man här vidta någon form av åtgärd. Vi testade två olika varianter

- Sampeltiden dubblades från 1 till 2 sekunder.
- Styrsignalen straffades.

Tabell 10.1 visar resultatet av denna simulering. Man ser i tabellen att om Novatune är rätt inställd eliminerar den en laststörning bättre än PID-regulatorn. Då vi gjorde en rejäl dödtidsökning (5 gånger) klarade ingen av regulatorerna av att reglera utan att ny dimensionering först gjordes. En simulering med ny dimensionering av regulatorerna redovisas i tabell 10.2 och figur 10.2. Det bör påpekas att man genom att öka sampeltiden (och minska KD) kan få en robustare Novatune till priset av en sämre reglering för den nominella processen.

Tabell 10.1 Simuleringsresultat då dödtiden dubblats.

	STEP					LOAD				Not
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	
PID	0.4	3.0	50	11	2.8	3.1	7.1	40	3.1	1
PID	0.0	6.3	11	6.9	1.1	3.7	8.5	34	1.0	2
Nova	0.07	3.0	14	5.6	1.2	2.5	5.1	13	1.1	3
Nova	0.09	1.0	8.9	4.2	1.1	2.1	5.5	6.2	1.1	4

1. PID-regulatorn dimensionerad för nominella systemet (dödtiden=tidskonstanten=1). $T_s=0.3$.
2. PID-regulatorn dimensionerad för det verkliga systemet (dödtiden=2). $T_s=0.3$.
3. Novatune med $NA=2$, $NB=3$, $KD=2$ och $T_s=2$.
4. Novatune med styrsignalstraff ($PN=15$). Övriga parametrar är $NA=2$, $NB=4$, $KD=3$ och $T_s=1$.

Tabell 10.2 Simulering då processens dödtid är 5 sekunder

	STEP					LOAD				Not
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	
PID	0.0	12	21	14	1.1	7.9	17	140	1.0	1
Nova	0.23	2.4	22	9.7	1.4	6.8	13	55	1.3	2

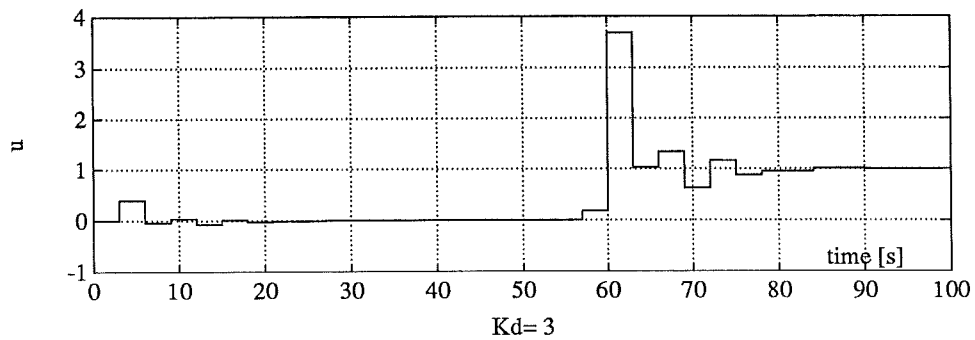
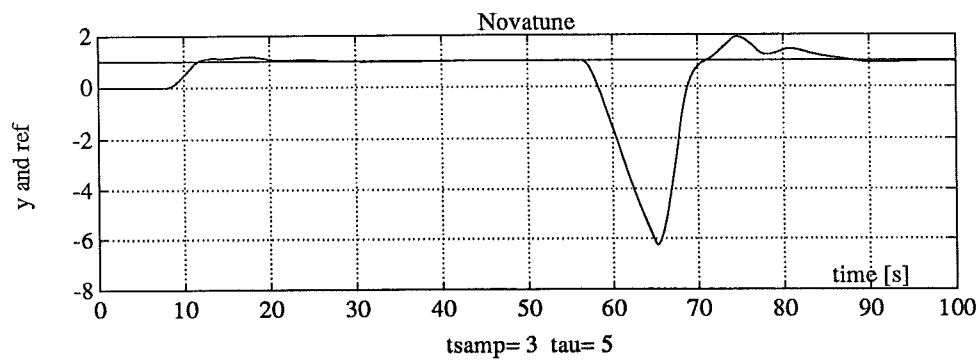
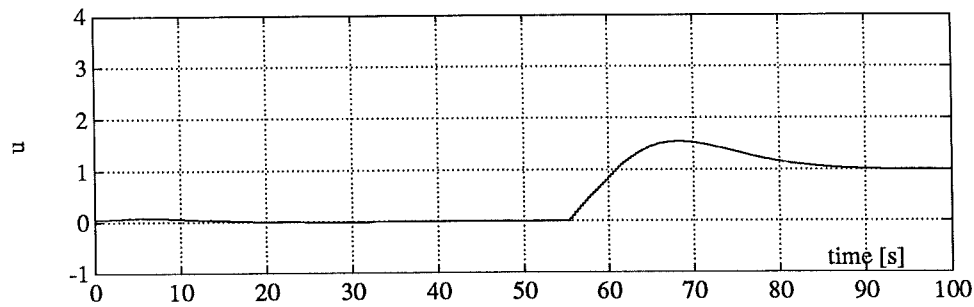
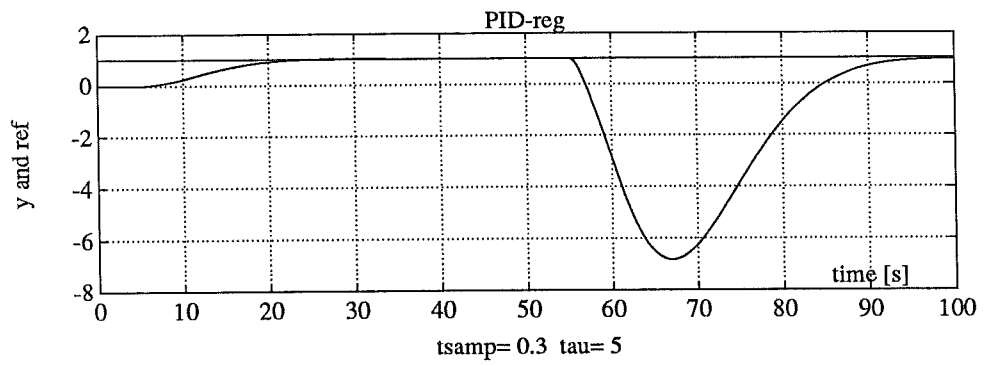
1. PID-regulator med sampeltiden $T_s=0.3$.
2. Novatune med $NA=2$, $NB=4$, $KD=3$ och $T_s=3$.

10.4 Bruskänslighet

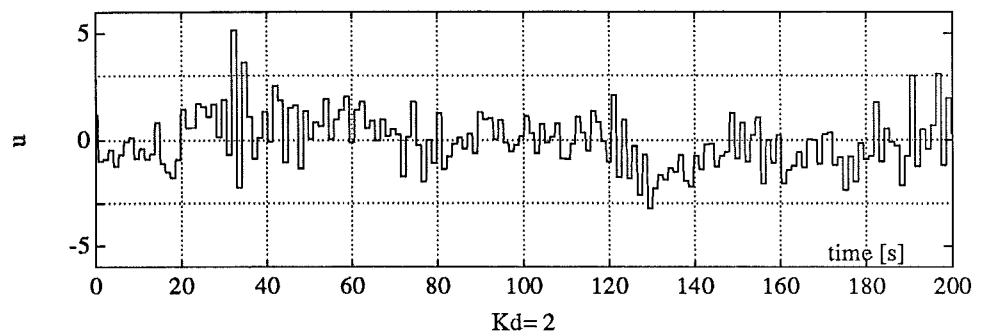
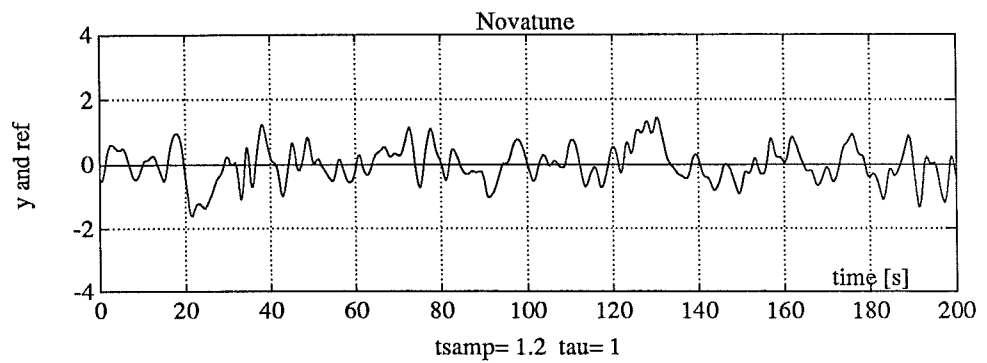
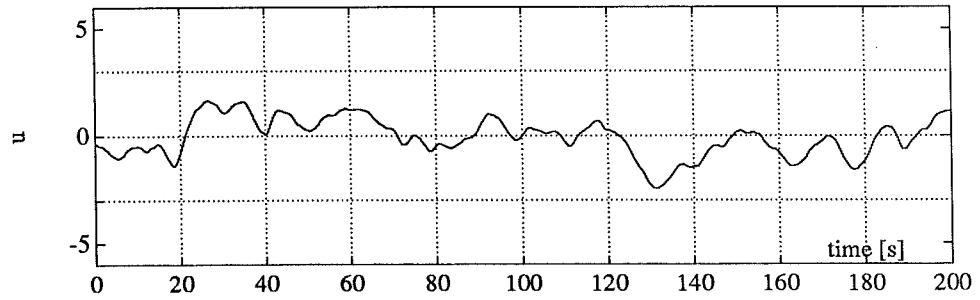
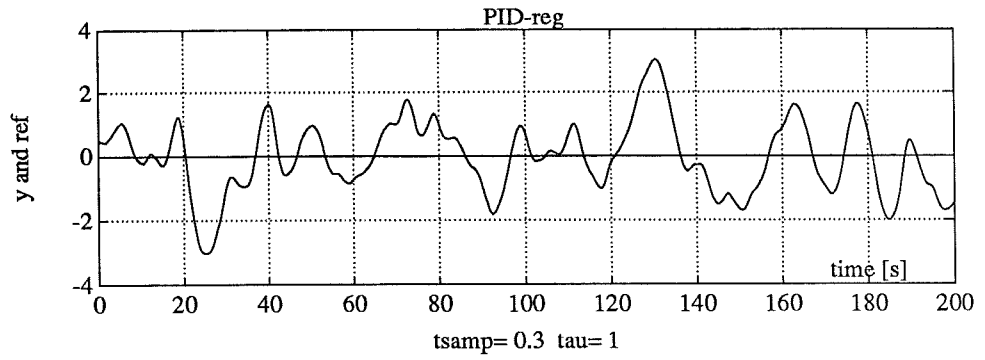
Det visade sig att för en integrerande process blir skillnaderna i bruskänslighet signifikanta (jämför med process 1). Särskilt stor skillnad var det när processerna utsattes för mätbrus. Anledningen till detta torde vara att PID-regulatorn har en relativt liten förstärkning. Även den kortare sampeltiden spelar roll. Som framgår av figur 10.3 och 10.4 har Novatune bättre prestanda vad beträffar processbrus medan motsatsen gäller för mätbruset. Det bättre resultatet för Novatune på processbruset medför dock en stor styrsignalvariation.

10.5 Slutsatser

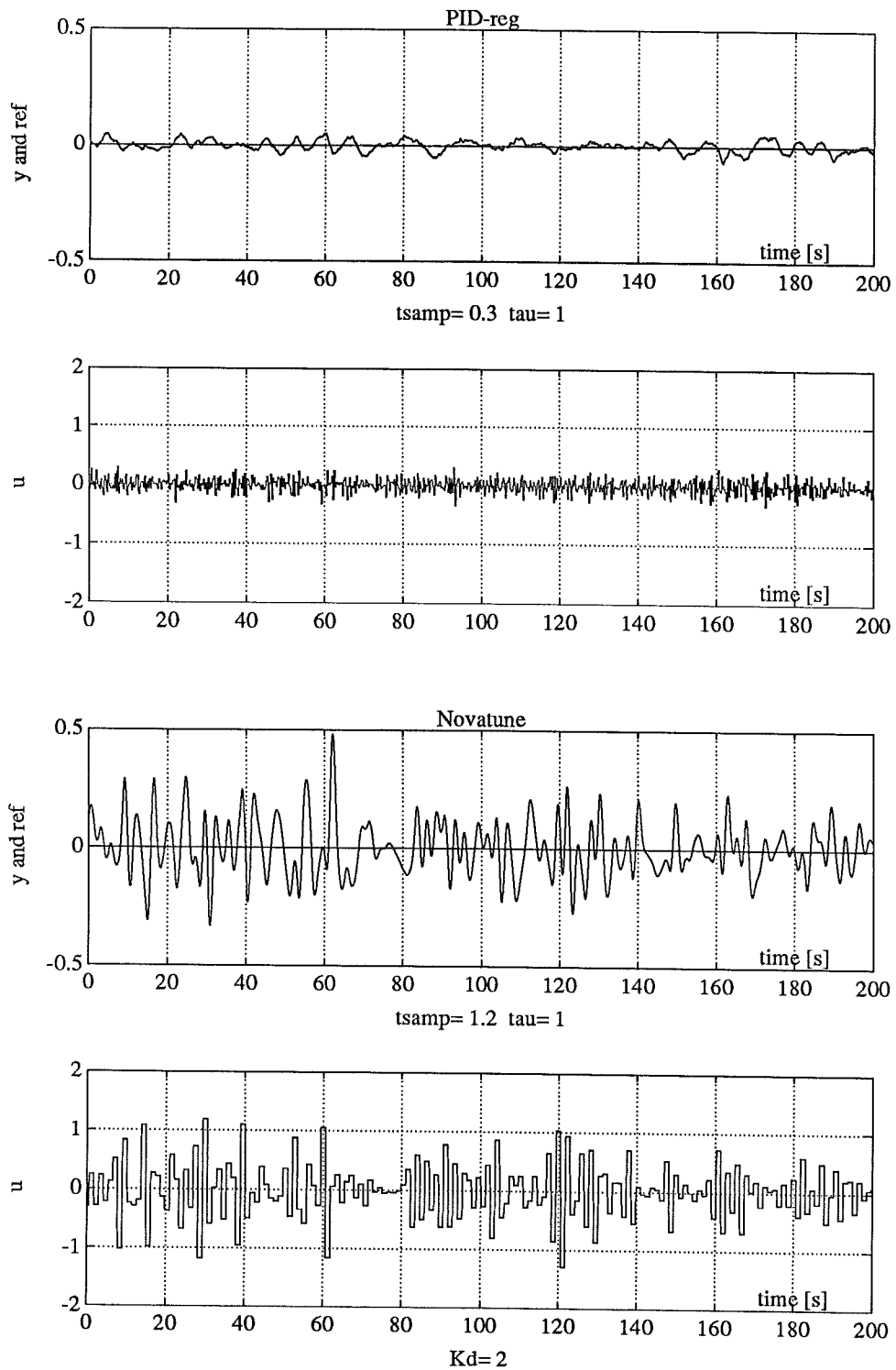
En integrerande process i kombination med variationer i dödtiden visar sig vara besvärligt för Novatune. Det fordras mycket inställningsarbete om man vill uppnå riktigt bra reglering samtidigt med en robust regulator. Har man däremot t ex ett DC-servo (utan dödtid) där polen flyttar sig då oljan i lagren blir varm är Novatune ett bra alternativ eftersom den reglerar ut en laststörning betydligt snabbare. Observera att det inte är dödtiden i sig själv som ställer till problem utan variationerna i densamma. Som nämnts ovan har Novatune svårare att klara en brusig återkopplingssignal.



Figur 10.2 Simulering med lång dödtid (5 sekunder). PID-regulatorn har sampeltiden, $T_s=0.3$ och Novatune parametrarna $NA=2$, $NB=4$, $KD=3$ och $T_s=2.8$. Sampeltiden har valts enligt avsnitt 8.2. Väljs en kortare sampeltid är risken för instabilitet stor.



Figur 10.3 Processbrus på systemet 10.3. PID-regulatorn har sampeltiden $T_s=0.3$ och Novatune parametrarna $NA=2$, $NB=3$, $KD=2$ och $T_s=1.2$. Novatune har bättre prestanda vilket dock medför en stor styrsignalvarians.



Figur 10.4 Mätbrus på systemet 10.3. PID-regulatorn har sampeltiden, $T_s=0.3$ och Novatune parametrarna $NA=2$, $NB=3$, $KD=2$ och $T_s=1.2$. PID-regulatorn är inte lika känslig för mätbrus. Troligen beror detta på att den har lägre förstärkning än Novatune. K för PID-regulatorn var i detta fallet 0.5.

11. Process 3

Vår tredje process har överföringsfunktionen

$$G(s) = \frac{K}{(1 + sT_1)(1 + sT_2)} e^{-sL} \quad (11.1)$$

Systemet kan t ex vara en beskrivning av två tankar, där den övre har sitt utlopp i den undre.

11.1 Novatunes inställning

Stegsvaret smyger igång. Ur stegsvaret för systemet 11.1 med $K = T_1 = T_2 = 1$ och $L = 0$ får vi inställningen för system utan dödtid: NA=2, NB=2, KD=1 och sampeltiden $T_s=1.5$. Vi ville sedan göra jämförelser hur dynamikförändringar inverkar när systemet innehåller dödtid. Vi valde då det nominella systemet

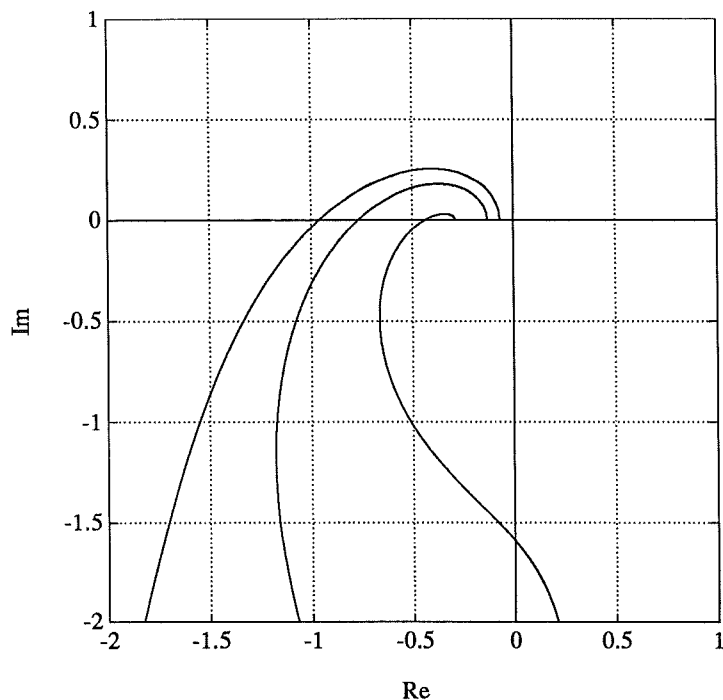
$$G(s) = \frac{1}{(1 + s)(1 + s)} e^{-s} \quad (11.2)$$

och Novatuneinställning med NA=2, NB=3, KD=2 och sampeltiden $T_s = 1.5$.

11.2 Variationer i tidskonstanterna

System utan dödtid Om systemet saknar dödtid kommer den adaptiva regulatören verkligen till sin rätt. Tester har gjorts för följande fall där PID-regulatorns sampeltid satts till 0.2 sekunder.

- En av polerna gjordes långsammare och därefter snabbare. Novatune klarar ändringar på upp mot 10 gånger, medans PID-regulatören får svårigheter då polen blir långsammare. Vi provade även att lägga en pol i höger halvplan. Till och med detta klarade Novatune.
- Båda polerna gjordes långsammare och därefter snabbare. I princip fick vi samma resultat som i föregående test, fast PID-regulatören får ännu större problem. I figur 11.1 visas nyquistdiagrammen för PID-regulatören och ett system där båda tidskonstanterna halverats. Genom att öka sampeltiden, vilket ger en långsammare dimensionering, kan robustheten förbättras.



Figur 11.1 Nyquistdiagram för PID-regulatorn och process utan dödtid när båda tidskonstanterna minskat till 0.5. Dimensionering har gjorts för $T_1 = T_2 = 1$. Från vänster till höger har följande sampeltider använts: $T_s=0.1, 0.2, 0.5$. Kurvorna går inte in till origo på grund av att det är det samplade systemet som visas.

System med dödtid. Med dimensionering för ekvation 11.2 gjorde vi simuleringar där vi flyttat en pol närmare origo. Eftersom Novatune är adaptiv ska detta inte medföra några problem, vilket det inte heller gjorde. PID-regulatorn klarade visserligen av att reglera men som visas i figur 11.2 är insvängningstiden lång på både stegsvar och laststörning.

11.3 Dödtidsändringar

Utgående från det nominella systemet 11.2 klarar PID-regulatorn en dubbling av dödtiden medan Novatune måste dimensioneras om. Vi valde att öka sampeltiden så lite som möjligt. Nya sampeltiden sattes lika med dödtiden. Med denna inställning är Novatune något sämre än den omdimensionerade PID-regulatorn. Vid en ännu längre dödtid var vi tvungna att dimensioneringa om både PID-regulatorn och Novatune. Resultaten blev likvärdiga. I tabell 11.1 ser man att PID-regulatorn visserligen klarar en dubbling av dödtiden, men kvoten IAE/IE (I_q) är stor vilket indikerar att systemet är slängigt. Låter vi dompidz dimensionera en ny regulator blir resultatet avsevärt bättre.

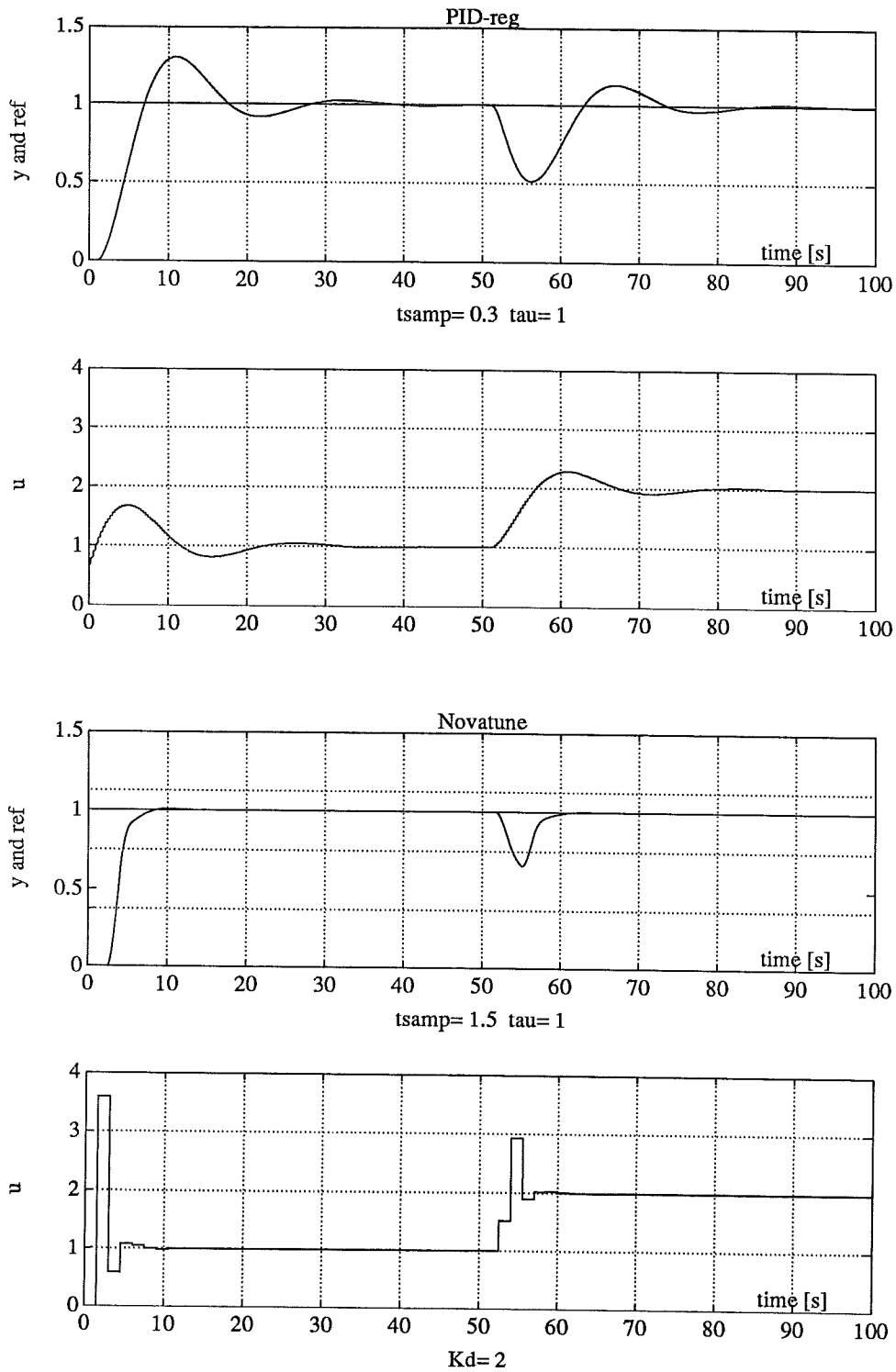
Tabell 11.1 Simuleringsresultat då dödtiden ökats från 1 till 2 sekunder.

	STEP					LOAD				Not
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	
PID	0.37	2.2	15	5.7	2.0	0.80	5.3	4.5	1.7	1
PID	0.03	3.6	6.8	4.6	1.1	0.81	5.5	4.4	1.0	2
Nova	0.006	2.0	8.0	5.6	1.0	0.83	5.0	4.0	1.0	3

1. PID-regulatorn dimensionerad för nominella systemet ekv 11.2. $T_s=0.3$.
2. PID-regulatorn dimensionerad för det verkliga systemet (dödtiden=2). $T_s=0.3$.
3. Novatune med $NA=2$, $NB=3$, $KD=2$ och $T_s=2$.

11.4 Slutsatser

I ett system med två poler kan dynamiken ändra sig drastiskt, något som Novatune visar sig klara utmärkt. Eftersom polerna är reella och processen inte är integrerande kan man i det här fallet utan tvekan välja Novatune. Som vanligt ställs man inför valet mellan bra prestanda och förmåga att klara variationer i dödtiden (jämför med slutsatserna för process 1). Som visas i figur 11.1 kan PID-regulatorns robusthet förbättras med en längre sampeltid.



Figur 11.2 Figuren illustrerar vad som händer om de båda regulatorerna dimensionerats för ett system med dubbelpol i -1 varefter en av polerna flyttat sig till -0.25. Systemet har en dödtid på en sekund. Det framgår klart att en adaptiv regulator är att föredra vid denna typ av processer där dynamiken kan ändra sig drastiskt.

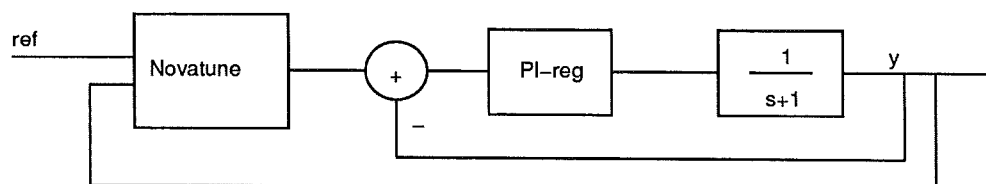
12. Process 4

Ett system med komplexa poler beskrivs ofta med överföringsfunktionen

$$G(s) = \frac{K}{s^2 + 2\zeta\omega s + \omega^2} e^{-sL} \quad (12.1)$$

Exempel på denna typ av process kan vara

- en massa upphängd i en fjäder och någon form av dämpning, där utsignalen är massans läge.
- ett DC-servo med fjädrande axlar, med insignalen en spänning och utsignalen axelns vinkelhastighet.
- reglerloop med en inre slinga, där den inre slingan består av ett första ordningens system och en regulator som tillsammans ger ett slutet system med komplexa poler.



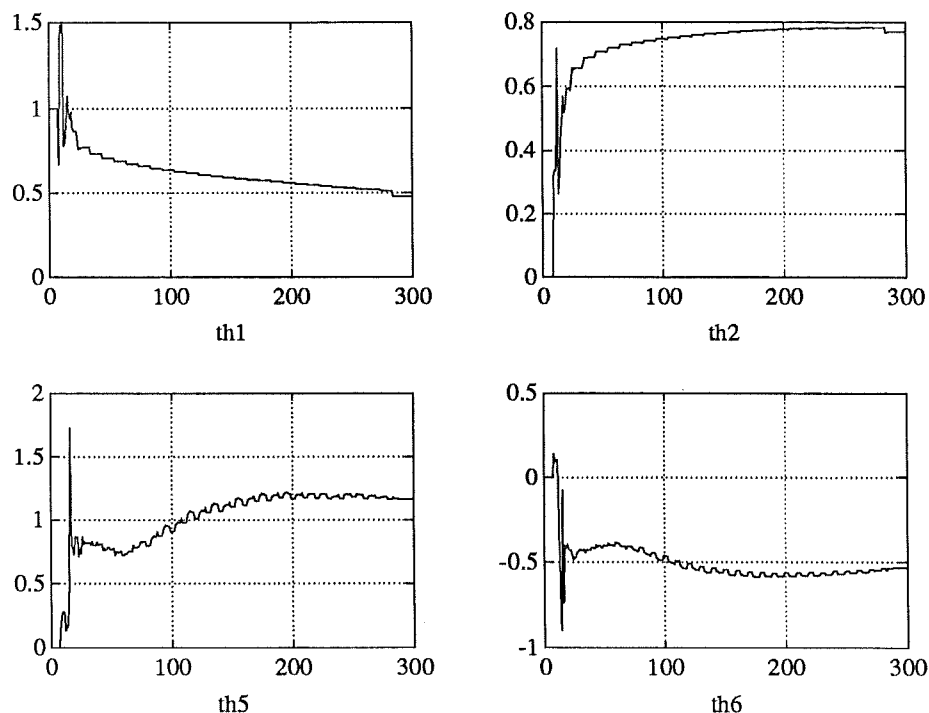
Figur 12.1 Kaskadreglering där den inre loopen kan ha oscillativa poler.

12.1 Inställning Novatune

Installationsanvisningen saknar vägledning då öppna systemets stegsvar har översläng. Emellertid föreslår installationsguiden $K_d=1$, $N_A=N_B=2$ och sam-peltiden den tid då stegsvaret uppnått 25-50% av sitt slutvärde för liknande system utan översläng och utan dödtid. Denna inställning klarar inte någon dödtid, om det trots antagandet skulle finnas en sådan. Vårt val av paramet-rar när systemet innehåller dödtid grundar sig på en rad testsimuleringar. För systemet

$$G(s) = \frac{1}{s^2 + 1.4s + 1} e^{-s} \quad (12.2)$$

fick vi en bra regulator då $N_A=2$, $N_B=4$, $K_D=3$ och $T_s=1$. Figur 12.2 vi-sar parameterinsvängningen när fyrkantvågen som vi exciterar systemet med under insvängning har en periodtid på 20 sekunder.



Figur 12.2 Parametrarnas insvängning för Novatune på process 12.2 med $NA=2$, $NB=4$, $KD=3$, $T_s=1$. Parametrarna svänger in långsamt på grund av att vi har otillräcklig excitation för så många parametrar.

12.2 Ändringar i systemets dämpning

När vi gjorde dimensioneringen för ett system utan dödtid satte vi dämpningen ζ till 0.7 och vinkelhastigheten ω till 1 så att vårt nominella system såg ut som

$$G(s) = \frac{1}{s^2 + 1.4s + 1} \quad (12.3)$$

Båda regulatorerna klarade en minskning av ζ till 0.3 utan problem. Resultatet visas i tabell 12.1. Val av sampeltid får stora konsekvenser på PID-regulatorn eftersom dompidz ger oss en regulatorinställning med hög förstärkning vid en kort sampeltid. I princip fås ett bättre resultat ju snabbare vi samplar, å andra sidan blir regulatorn samtidigt mer känslig om en dödtid smyger sig in i systemet. Vad gäller Novatune visade det sig att den lätt ger ringningar i styrsignalen då $KD=1$. Man kan i många fall få en bättre reglering om KD sätts större än 1. I figur 12.4 visas ett exempel på ringningar i styrsignalen. Med KD större än 1 kan man dessutom sampla snabbare, vilket framför allt är viktigt om processen är dåligt dämpad.

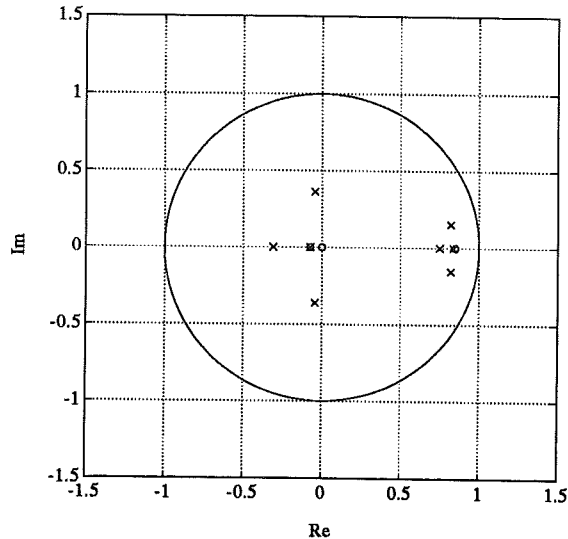
Tabell 12.1 Simulering vid ändring av ζ .

	STEP					LOAD				Not
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	
PID	0.05	0.9	1.2	0.72	1.1	0.15	1.0	0.21	1.0	1
Nova	0.02	1.4	5.6	2.8	1.0	0.66	2.1	1.6	1.0	2
PID	0.10	0.7	2.9	0.76	1.2	0.18	1.0	0.22	1.0	3
Nova	0.08	1.3	8.5	3.2	1.2	0.99	2.3	2.6	1.0	4
Nova	0.23	0.9	3.2	1.1	2.2	0.47	1.5	0.86	1.4	5

1. Stegsvär och laststörning för PID-regulator och systemet 12.3. Sampeltiden $T_s=0.2$.
2. Stegsvär och laststörning för systemet 12.3 och Novatune med $NA=NB=2$, $KD=1$ och $T_s=1.5$.
3. PID-regulator med samma dimensionering som till (1) men processens dämpning ändrad till $\zeta=0.3$. PID-regulatorn klarar ändringen bra. Diagram över simuleringen visas i figur 12.4.
4. Novatune med samma intällning som till (2) men processens dämpning ändrad till $\zeta=0.3$. Diagram över simuleringen visas i figur 12.4.
5. Novatune med $KD=2$ och $T_s=0.8$. Värdena i tabellen visar att KD bör väljas större än 1 för denna typ av processer.

12.3 Dödtidsändringar

Dimensioneringen av en PID-regulator för ett system med dödtid gav oss problem. När vi begärde ett slutet system med $\zeta=0.7$ gav oss dompidz negativa värden på K , T_i och T_d . Vid omräkning till RST-polynom fick vi trots allt fungerande regulatorer. I sammanhanget bör det påpekas att dompidz gör dimensionering i diskret tid. Problemet med negativa värden kan lösas om man av slutna systemet kräver ett ζ mindre än 0.7. Figur 12.3 visar var dompidz placerar poler och nollställen då processen har $\zeta=0.3$, $\omega=1$, dödtiden=1 och den önskade dämpningen 0.7. Om polerna mappas över till kontinuerliga s-planet får vi den önskade polmarginalen. Vi har inte fördjupat oss i detta problem utan nöjer oss med att konstatera att denna polplacering inte är realiserbar med en PID-regulator eftersom K , T_i och T_d inte kan ges negativa värden.



Figur 12.3 Pol-nollställesplot då processen har $\zeta=0.3$, $\omega=1$, dödtiden=1 och den önskade dämpningen är 0.7. När polerna mappades över till kontinuerlig tid blev polmarginalen=1.63.

Efter en dubbling av dödtiden klarar sig de båda regulatorerna fortfarande bra. Som framgår av tabell 12.2 har två simuleringar med Novatune gjorts. Den första gav en något slängig styrsignal, varför vi lade på ett styrsignalstraff (PN=15). Detta gjorde att vi fick en lugnare styrsignal samtidigt som insvängningstiden halverades.

Tabell 12.2 Dubbling av dödtiden utgående från ekv 12.2.

	STEP					LOAD				Not
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	
PID	0.3	2.5	17	5.6	1.9	0.97	5.3	5.4	1.5	1
PID	0.04	3.5	6.8	4.7	1.1	0.98	5.5	5.3	1.0	2
Nova	0.07	0.9	9.5	4.2	1.1	0.91	4.8	2.9	1.1	3
Nova	0.03	1.2	4.7	4.0	1.0	0.93	4.9	2.9	1.0	4

1. PID utan ny dimensionering, $T_s=0.3$.
2. PID med ny dimensionering, $T_s=0.3$.
3. Novatune med NA=2, NB=3, KD=2, $T_s=1.5$ och styrsignalstraffet PN=0.
4. Novatune med PN=15, för övrigt enligt (3).

12.4 Ändringar i systemets dämpning då det finns dödtid

När det finns dödtid i systemet får som väntat en minskning av dämpningen ζ från 0.7 till 0.3 större konsekvenser. PID-regulatorn tenderar att bli instabil. Resultaten från denna simulering visas i tabell 12.3 och figur 12.5. Lagg märke till kvoten IAE/IE för PID-regulatorn. Vi har här satt KD=3 för att kunna sampla oftare. Detta ger en bra regulator som är okänslig för ändringar i dämpningen. Vi ser också att styrsignalen är relativt fri från ringningar trots att vi inte har något styrsignalstraff.

Tabell 12.3 Simulering med 1 sekunds dödtid. Regulatorerna är dimensionerade för dämpningen $\zeta=0.7$ men vi har här ändrat ζ till 0.3

	STEP					LOAD				Not
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	
PID	0.15	1.9	15	4.0	1.4	1.2	3.7	5.3	1.5	1
Nova	0.01	1.2	5.2	3.0	1.0	1.2	3.7	3.3	1.0	2

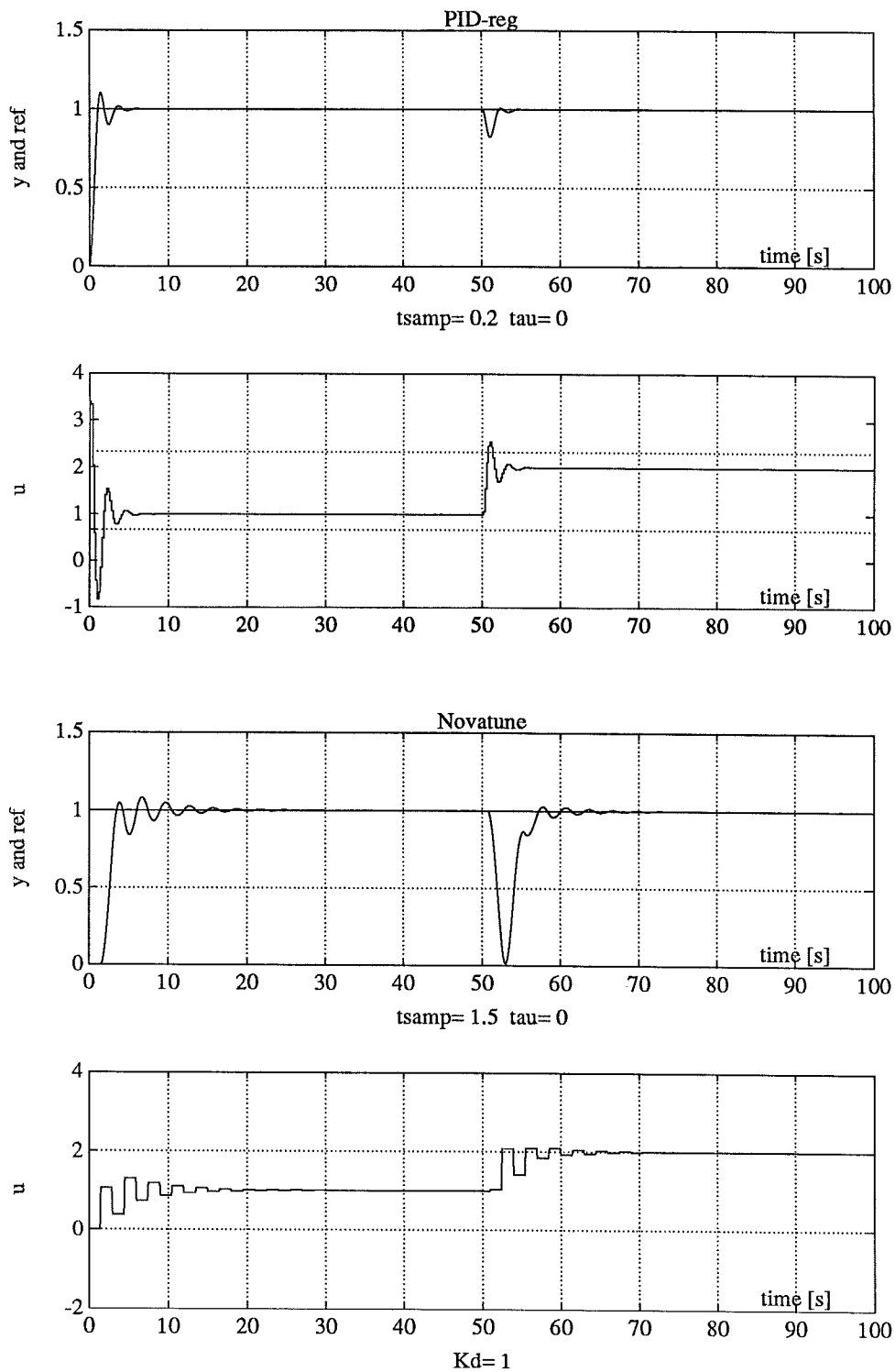
1. PID dimensionerad för ekvation 12.2. Sampeltiden $T_s=0.3$.
2. Novatune med $NA=2$, $NB=4$, $KD=3$ och $T_s=1$.

12.5 Slutsatser

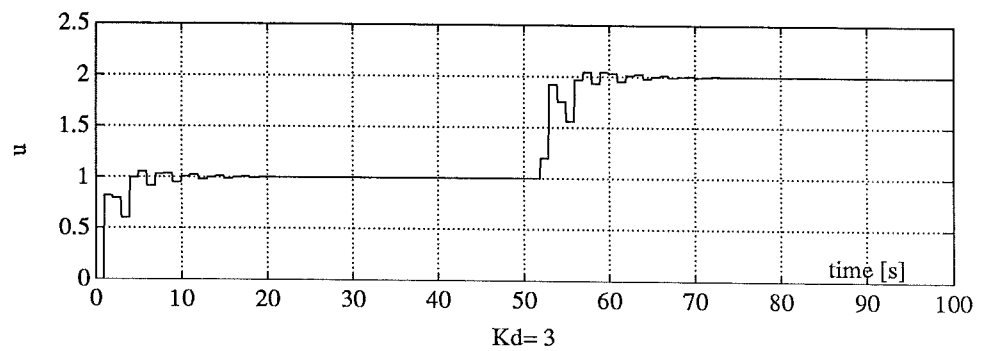
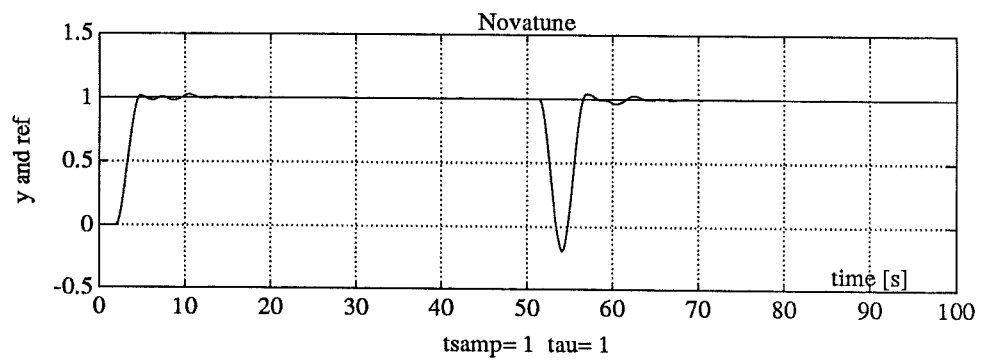
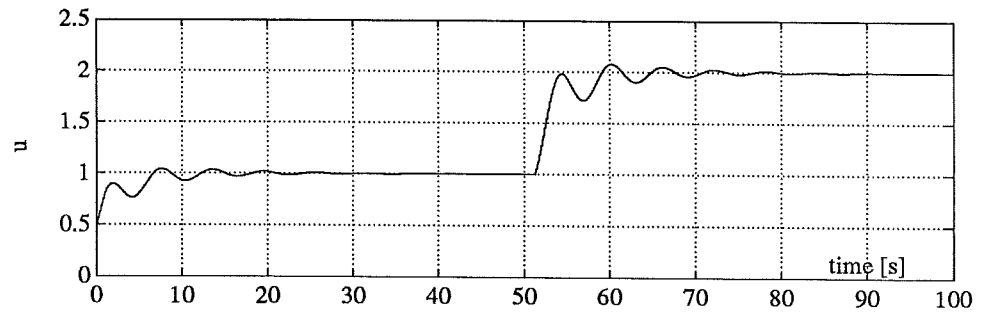
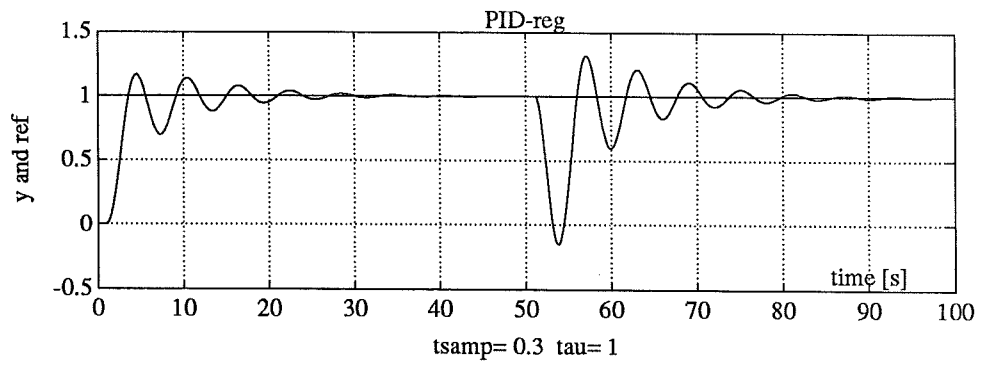
Den viktigaste slutsatsen är att valet av regulator bör grunda sig på huruvida systemet innehåller dödtid eller ej. Figurerna 12.4 och 12.5 illustrerar skillnaden mellan de två fallen. Om systemet saknar dödtid (och man är säker på detta) är PID-regulatorn nog att föredra.

Finns det dödtid måste förstärkningen minskas drastiskt för att stabiliteten ska bibehållas. Varierar dämpningen i ett system med dödtid klarar sig Novatune betydligt bättre.

Vad gäller dödtidsändringar visade sig de båda regulatorerna vara likvärdiga.



Figur 12.4 Övre figuren visar PID-regulator dimensionerad för ekv 12.3 men dämpningen ζ har ändrats till 0.3. Sampeltiden $T_s=0.2$. Den undre figuren visar Novatune för samma process med $NA=NB=2$, $KD=1$ och sampeltiden $T_s=1.5$. Simuleringen med Novatune antyder att man bör välja $KD > 1$ när vi har ett dåligt dämpat system.



Figur 12.5 PID-regulatorn och Novatune dimensionerade för ekv 12.2 men ζ har ändrats till 0.3. Novatune klarar ändringen bra medan PID-regulatorn uppvisar tendens till instabilitet.

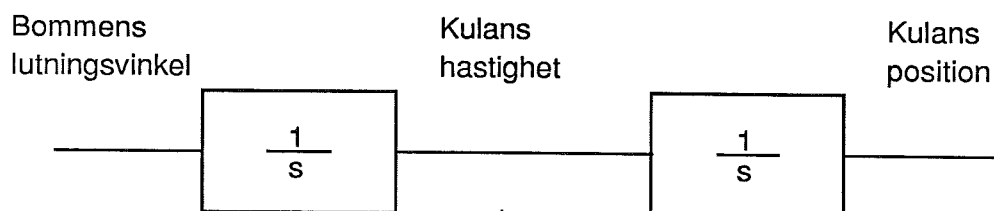
13. Process 5

Detta kapitel behandlar en process som endast består av en dubbelintegrator och en dödtid. Överföringsfunktionen är

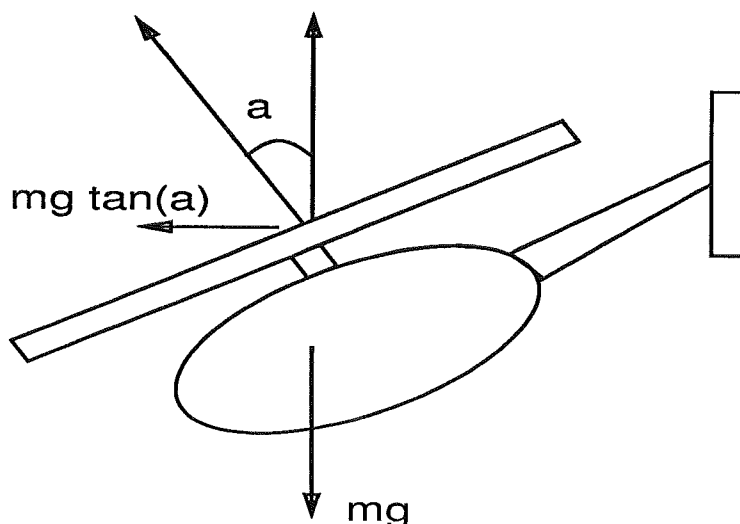
$$G(s) = \frac{K}{s^2} e^{-sL} \quad (13.1)$$

När en laststörning kommer in på en dubbelintegrator, blir felet i allmänhet ganska stort, vilket också visade sig i våra simuleringar. Ett stegsvar är lättare att reglera in. Om dödtiden är känd är det bra att sampla ofta, så att en laststörning upptäcks så tidigt som möjligt, innan processen hinner integrera upp ett stort fel. Det finns gott om fysikaliska processer som kan beskrivas som dubbelintegratorer, t ex

- Armservot i en CD-spelare (något förenklat).
- En rullande kula på en vridbar bom, där överföringsfunktionen från bommens lutningsvinkel till kulans läge kan beskrivas med hjälp av en dubbelintegrator.
- En variant på förra exemplet är överföringsfunktionen från lutningen på rotorbladen på en helikopter till dess horisontella läge.



Figur 13.1 Överföringsfunktionen från bommens lutning till kulans läge i ovan beskrivna bomprocess.



Figur 13.2 Exempel på dubbelintegrator är överföringsfunktionen från rotorbladens lutning α till helikopterns läge. Systemet kan beskrivas med $\frac{m d^2 \alpha}{dt^2} = mg \tan(\alpha) + L$, där L är luftmotstånd m m.

13.1 Inställning av Novatune

Stegsvaret säger oss att processen är integrerande. Av parabelformen kan vi sluta oss till att den är av ordning 2 eller större. Med vetskapen om att vi har att göra med en dubbelintegrator gäller det att välja sampeltiden så kort som möjligt enligt ovanstående resonemang. Vi valde $NA=2$, $NB=3$, $KD=2$ och $T_s=0.7$ som grundinställning.

13.2 Test utan dödtid

När systemet är fritt från dödtid klarar sig båda regulatorerna bra. Det räcker dock med att en dödtid på ett par tiondelar smyger sig in i systemet för att ny dimensionering ska vara nödvändig.

Tabell 13.1 Dubbelintegrator utan dödtid.

	STEP					LOAD				
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	Not
PID	0.04	1.6	2.3	1.4	1.2	0.62	1.9	1.6	1.0	1
Nova	0.05	1.2	5.2	1.4	1.2	0.64	1.6	1.1	1.2	2

1. PID-regulatorn dimensionerad med dompidz. Sampeltiden, $T_s=0.3$.
2. Novatune har parametrarna $NA=2$, $NB=3$, $KD=2$ och $T_s=0.7$.

13.3 Test med dödtid

Här ökade vi sampeltiden till 1.2 sekunder på Novatune. Vi misstänkte att den skulle ge något sämre resultat eftersom vi samplar sällan jämfört med PID-regulatorn. Tvärtom fick vi utmärkt reglering även på en laststörning. Orsaken till detta torde vara att Novatune förser oss med en instabil regulator som visserligen visar mycket bra prestanda för den specifika process för vilken den är inställd, men i gengäld är väldigt känslig. När vi räknade om θ -parametrarna till R,S,T-polynom visade det sig att R-polynomet har ett nollställe utanför enhetscirkeln. Denna simulering visas i figur 13.4 där vi ser att en laststörning regleras ut på 6-7 sampel. När dödtiden ökades till 2 sekunder gav oss dompidz en PID-regulator där laststörningar slår igenom väldigt mycket. En simulering med denna dimensionering visas i figur 13.5. Vi försökte åstadkomma bättre reglering genom att

1. ändra sampeltiden.
2. minska det önskade systemets dämpning, ζ , för att på så sätt få dompidz att ge oss en regulator med större förstärkning.

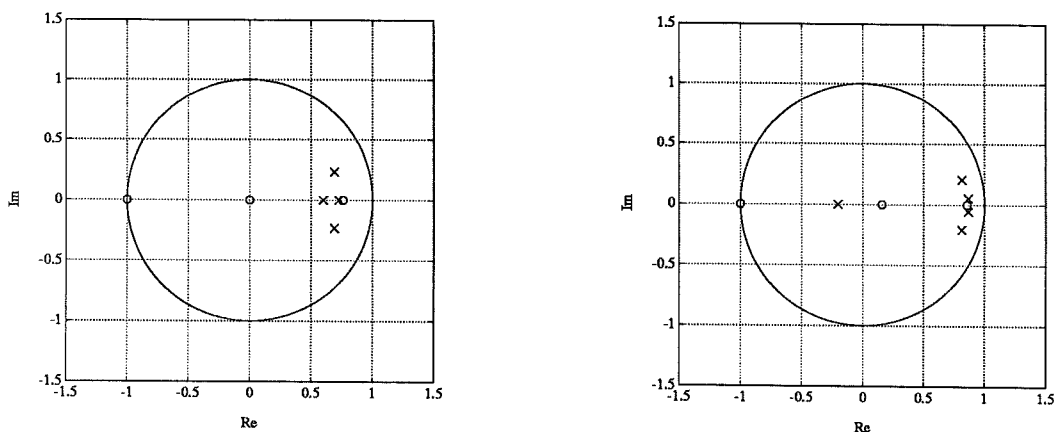
Som vi kan se i figurerna 13.6 och 13.7 fick inte någon av dessa båda åtgärder önskad effekt. Med hjälp av dompidz får vi PID-parametrarna K , T_i , T_d och de diskreta motsvarigheterna i form av R,S,T-polynom, vars ordningstal $n=2$. Vår slutsats är att detta inte är tillräckligt när den styrda processen är en dubbelintegrator med 2 sekunders dödtid. Vi dimensionerade därför en R,S,T-regulator enligt den polplaceringsalgoritm som beskrivs i Åström och Wittenmark (1990). En kort beskrivning av denna metod finns i kapitel 3. Resultatet av simuleringen visas i figur 13.8.

Tabell 13.2 Dubbelintegrator med en sekunds dödtid.

	STEP					LOAD				Not
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	
PID	0.02	7.1	12	6.8	1.0	12	8.4	118	1.0	1
PID	0.0	5.9	11	6.3	1.0	8.4	7.2	71	1.0	2
Nova	0.04	1.2	5.5	3.1	1.0	3.3	4.2	9.0	1.1	3
Nova	0.02	1.2	2.9	2.3	1.0	2.0	3.4	3.9	1.0	4

1. PID-regulatorn dimensionerad med dompidz. Sampeltiden, $T_s=0.3$. Dödtiden får stora konsekvenser framför allt vid en laststörning.
2. PID-regulatorn dimensionerad med dompidz. Sampeltiden har minskats till 0.1. Resultatet från föregående simulering har förbättrats.
3. Novatune med $NA=2$, $NB=3$, $KD=2$ och $T_s=1.2$.
4. Test med ny inställning, där vi sätter $KD=3$ för att kunna sampla oftare. Sampeltiden valdes till 0.7 sekunder. I denna simulering är $NB=4$.

I figur 13.3 ser vi förklaringen till att dompidz ger dålig PID-regulator när vi har dubbelintegrator med dödtid. Vänstra bilden visar pol-nollställesplot för slutna systemet när sampeltiden=1 och dödtiden=0. Vi får här den önskade polmarginalen 1.6. I högra figuren är dödtiden=sampeltiden (1 sekund). Dompidz har inte lyckats placera de dominanta polerna så att önskad polmarginal uppnås.



Figur 13.3 Pol-nollställesdiagram för slutna systemet med PID-regulator och dubbelintegrator utan och med dödtid.

13.4 Bruskänslighet

Tabell 13.3 och 13.4 innehåller de varianser som erhöles för ut- och styrsignalen när regulatorerna testades på en dubbelintegrator utan dödtid. Lägg särskilt märke till den höga variansen hos Novatunes Δu vid mätbrus. Se även figur 13.9 och 13.10

Tabell 13.3 Processbrus på dubbelintegrator utan dödtid

	$E(y)$	$V(y)$	$V(y_s)$	$V(\Delta u)$	C_v	Not
PID	-0.0085	0.055	0.055	0.024	0.52	1
Nova	-0.0028	0.029	0.026	8.3	4.1	2

1. PID-regulator med $T_s=0.3$.
2. Novatune med $NA=2$, $NB=3$, $KD=2$ och $T_s=0.7$.

I figur 13.9 ser vi simuleringen med Novatune då processen utsatts för processbrus och i figur 13.10 motsvarande θ -parametrar. Av någon anledning hamnar parametrarna i ett instabilt jämviktsläge. Efter cirka 100 sekunder blir styrsignalvariansen mycket stor. Detta leder till att processen exciteras med höga frekvenser vilket leder till att parametrarna får ett nytt jämviktsläge. Vi har inte fördjupat oss i fenomenet men konstaterar att detta är anledningen till den höga styrsignalvariansen i tabell 13.3.

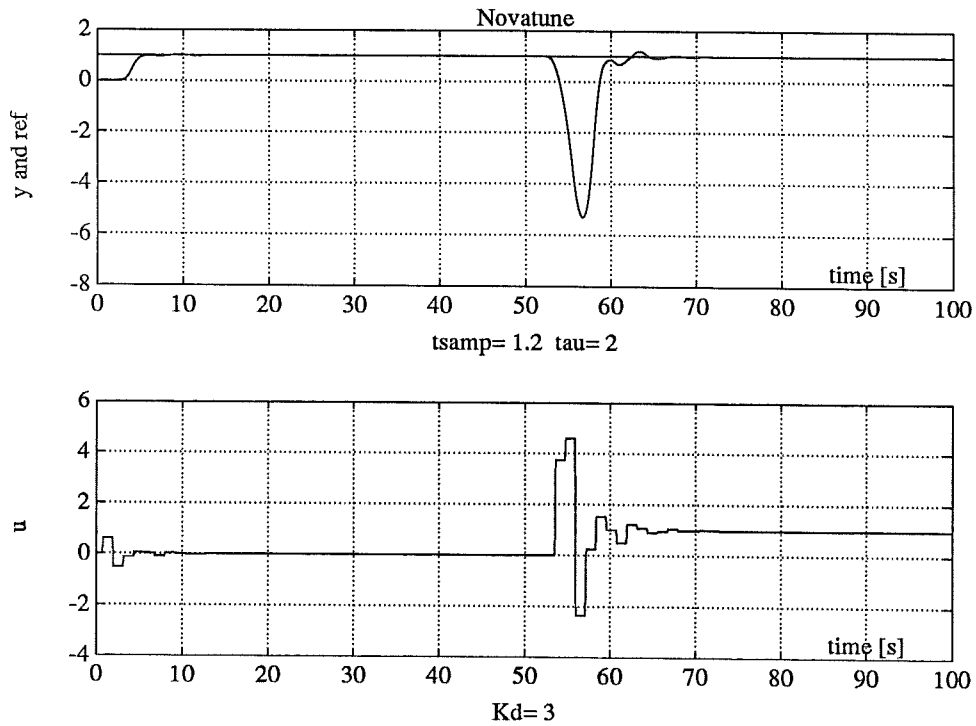
Tabell 13.4 Mätbrus på dubbelintegrator utan dödtid.

	$E(y) 10^{-3}$	$V(y) 10^{-3}$	$V(y_s) 10^{-3}$	$V(\Delta u) 10^{-3}$	C_v	Not
PID	-0.23	2.3	2.2	650	2.7	1
PID	2.4	2.7	2.8	19.2	0.20	2
Nova	-3.5	9.4	8.8	1600	1.8	3

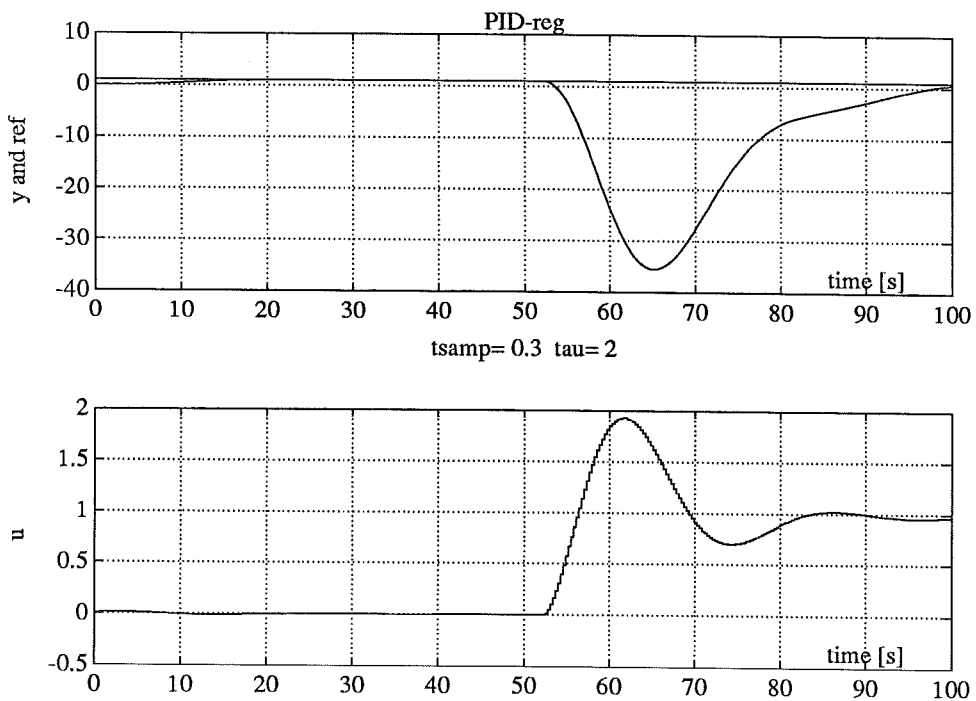
1. PID-regulator med $T_s=0.3$.
2. PID-regulator med $T_s=0.7$.
2. Novatune med $NA=2$, $NB=3$, $KD=2$ och $T_s=0.7$.

13.5 Slutsatser

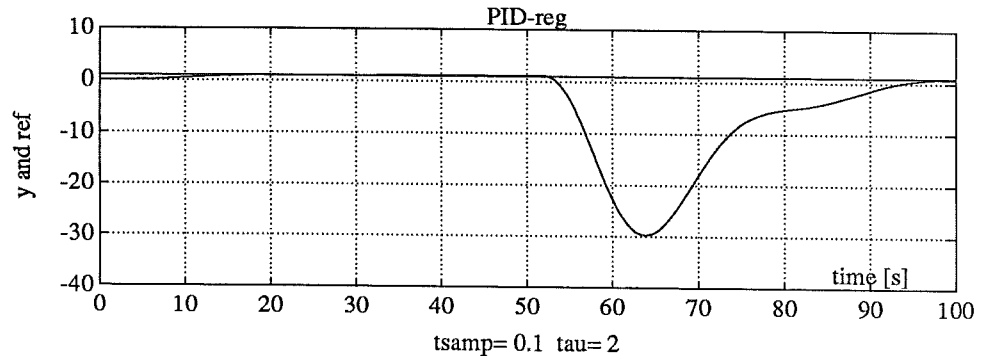
Liksom i föregående kapitel är det stor skillnad mellan regulatorerna då det finns dödtid i systemet. Det krävs en mer komplex regulator för att placera polerna så att det slutna systemet beter sig som önskat. Båda regulatorerna är känsliga för dödtidsvariationer. I valet mellan PID och Novatune bör man välja den senare. Man kan förbättra Novatunes prestanda genom att välja ett högt KD (sampla oftare).



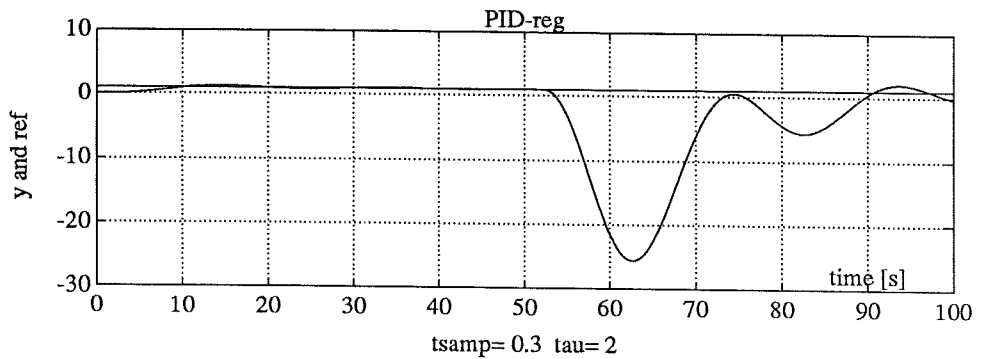
Figur 13.4 Novatune och process med två sekunders dödtid. Regulatorn är i detta fallet instabil, R-polynomet har nollställen utanför enhetscirkeln. $NA=2$, $NB=4$, $KD=3$ och sampeltiden $T_s=1.2$



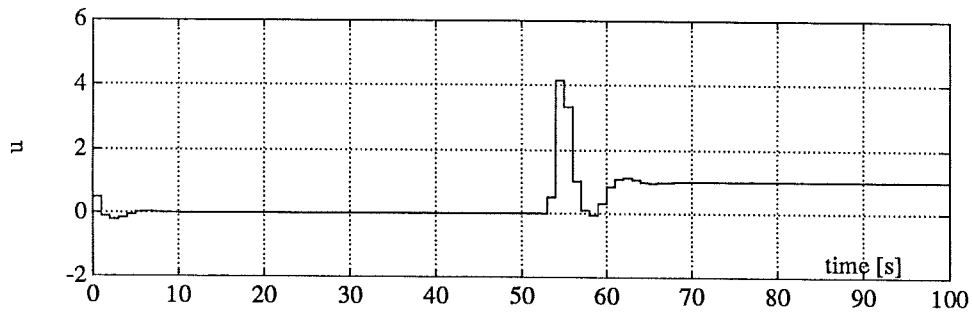
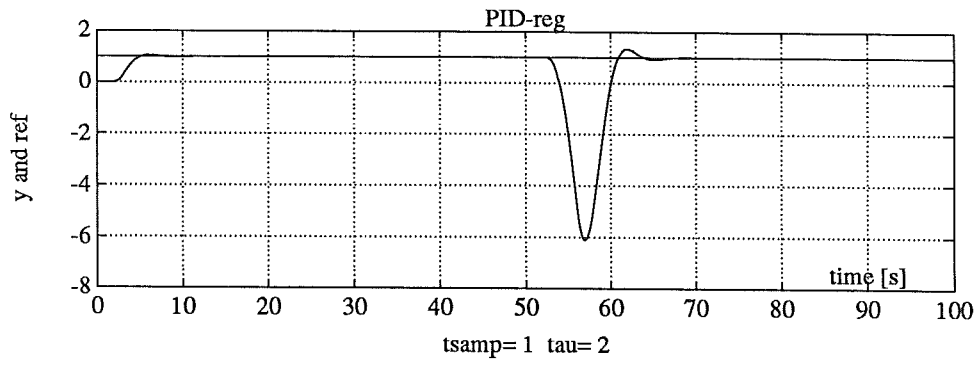
Figur 13.5 PID-regulator och dubbelintegrator med två sekunders dödtid. Regulatorn dimensionerad med dampidz . Sampeltiden $T_s=0.3$ och den önskade dämpningen $\zeta=0.7$. Regulatorn eliminerar laststörningen mycket dåligt.



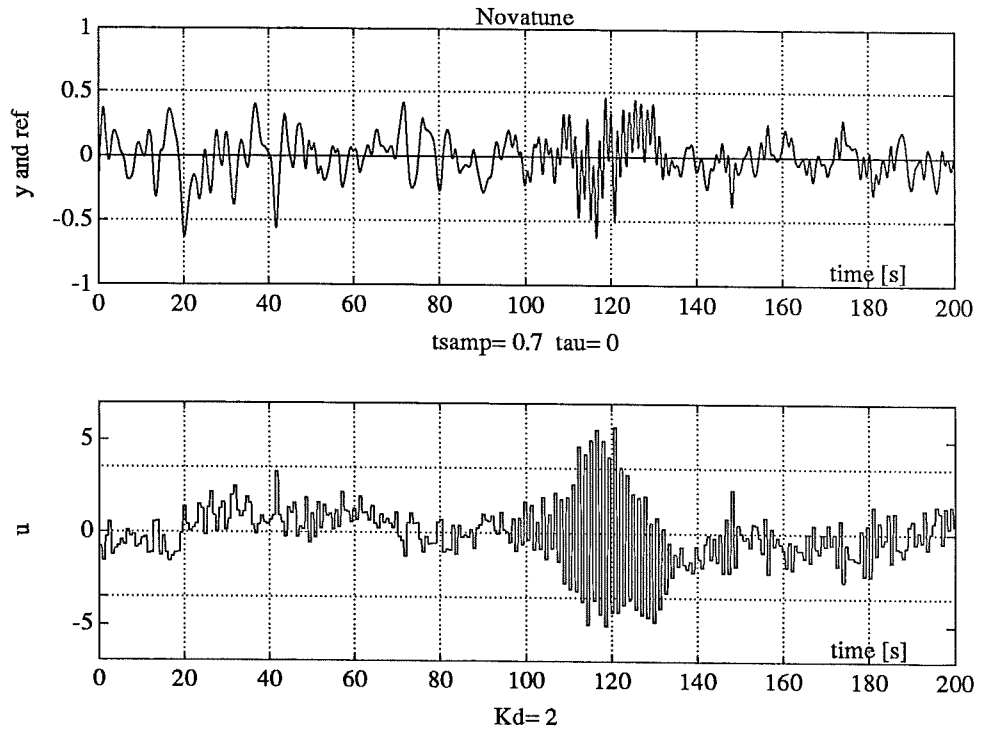
Figur 13.6 PID-regulator och dubbelintegrator med två sekunders dödtid. Regulatorn dimensionerad med dompidz. Sampeltiden har minskats till 0.1. Den kortare sampeltiden fick endast marginell effekt (jämför figur 13.5).



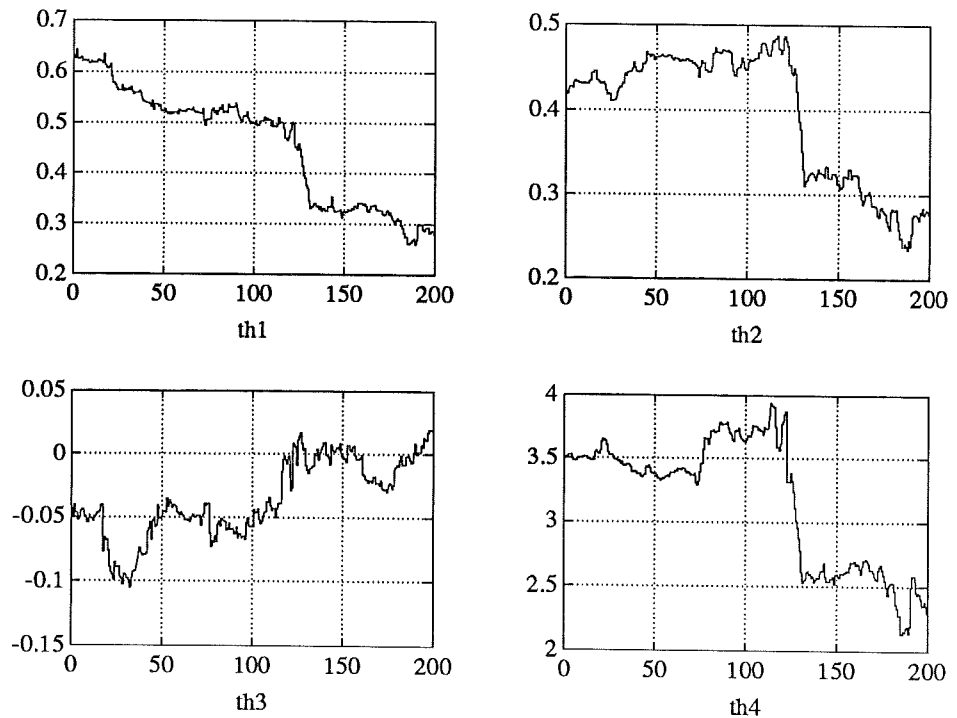
Figur 13.7 PID-regulator och dubbelintegrator med 2 sekunders dödtid. Sampeltiden $T_s=0.3$ men den önskade dämpningen har minskats till 0.3. Inte heller denna åtgärd gav önskad effekt. (jämför figurerna 13.5 och 13.6).



Figur 13.8 RST-regulator och dubbelintegrator med två sekunders dödtid. Regulatorn dimensionerad enligt polplaceringsprincipen (se texten). Observera skalorna vid jämförelse med de tidigare figurerna.



Figur 13.9 Simulering som motsvarar rad 2 i tabell 13.4. Processbrus på en dubbelintegrator och Novatune med $NA=2$, $NB=3$, $KD=2$, $T_s=0.7$. Efter halva simuleringen börjar plötsligt styrsignalen svänga.



Figur 13.10 Fyra av de fem θ -parametrarna under brus-simuleringen i figur 13.9. Parametrarna tycks inte vara stabila.

14. Process 6

Vår sjätte process är av tredje ordningen med endast reella poler. Den har överföringsfunktionen

$$G(s) = \frac{K}{s(1 + sT_1)(1 + sT_2)} e^{-sL} \quad (14.1)$$

Stegsvaret för denna överföringsfunktion liknar DC-servots. Det har dock ett skarpare knä.

14.1 Inställning av Novatune

Vid inställning av Novatune är det nödvändigt att processens stegsvar nått tillräckligt högt vid tiden $KD \cdot T_s$. Med en allt för lång sampeltid fångas inte laststörningar upp. Därför valde vi ett $KD=2$, även för system utan dödtid. Med den metod för inställning av Novatune som beskrivs i avsnitt 8.2 fås sampeltiden $T_s=1.2$ om ingen dödtid finns. Antalet parametrar NA och NB sattes båda till 3.

14.2 Ändringar i processdynamiken

Vid simuleringar på vår grundmodell

$$G(s) = \frac{1}{s(1 + s)(1 + s)} \quad (14.2)$$

uppför sig båda regulatorerna bra. Novatune eliminerar en laststörning bättre än PID-regulatorn. För att ge PID-regulatorn en chans att upptäcka laststörningen snabbare provade vi att sampla oftare ($T_s=0.1$), men resultatet blev inte nämnvärt bättre. Efter att ha dimensionerat regulatorerna för den nominella processen (ekv 14.2) testade vi överföringsfunktionen

$$G(s) = \frac{1}{s(1 + 4s)(1 + s)} \quad (14.3)$$

där en pol gjorts 4 gånger långsammare. Resultaten visas i tabell 14.1.

Tabell 14.1 Simuleringar då systemets ena pol gjorts långsammare.

	STEP					LOAD				Not
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	
PID	0.45	4.5	101	19	4.8	2.1	8.6	55	4.6	1
PID	0.02	7.7	12	6.7	1.1	1.8	8.7	20	1.0	2
Nova	0.01	2.1	4.4	2.8	1.0	0.32	3.2	0.96	1.0	3

1. Stegsvaret och laststörning för PID-regulator och systemet ekv 14.3. PID-regulatorn dimensionerad för 14.2. Sampeltiden, $T_s=0.3$. Systemet är mycket slängigt.
2. Stegsvaret och laststörning för PID-regulator och ekv 14.3. Ny dimensionering är gjord. Resultatet från föregående simulering har förbättrats väsentligt.
3. Stegsvaret och laststörning för ekv 14.3. Novatunes inställning är $NA=NB=3$, $KD=2$ och sampeltiden $T_s = 1.2$. Klart bättre prestanda än för PID-regulatorn.

Simuleringarna visas även i figurerna 14.1-14.3. Som vi kan se har PID-regulatorn längre lösningstid och större IAE även när den dimensionerats för verkliga systemet. Vi provade att låta dompidz dimensionera med önskad dämpningen $\zeta=0.3$, men resultatet blev inte bättre. Novatune å andra sidan klarar sig till och med om båda polerna samtidigt flyttas in till -0.25.

14.3 Dödtidsändringar

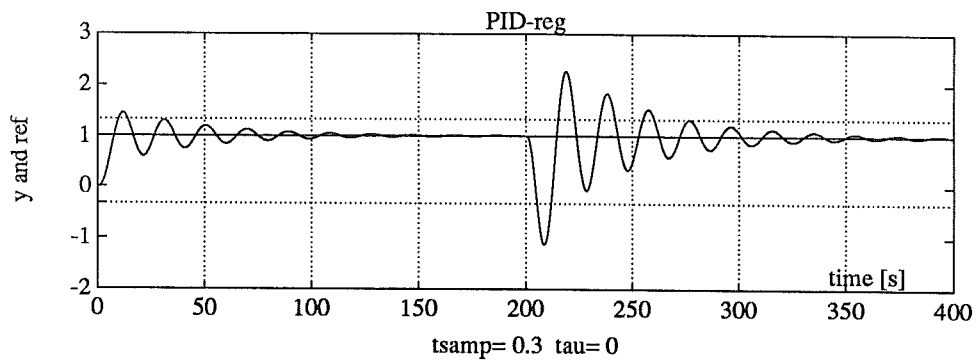
Från grunddimensioneringen för ekv 14.2 klarade båda regulatorerna en dödtidsökning till 0.5 sekunder, med längre dödtider var ny dimensionering nödvändig. Den nya dimensioneringen gjorde vi för processen

$$G(s) = \frac{1}{s(1+s)(1+s)} e^{-2s} \quad (14.4)$$

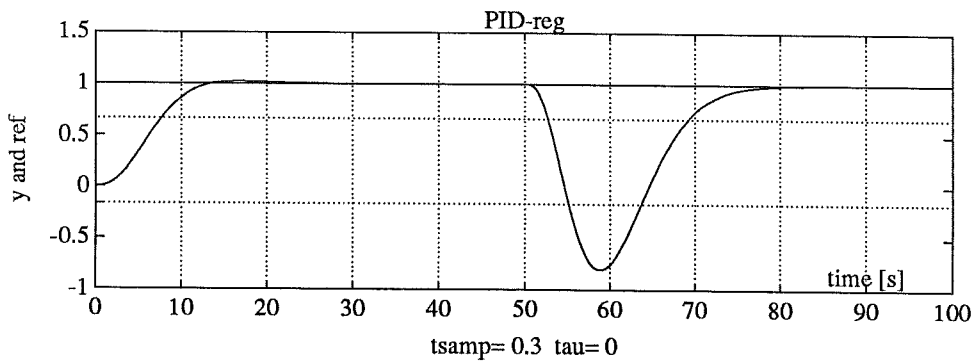
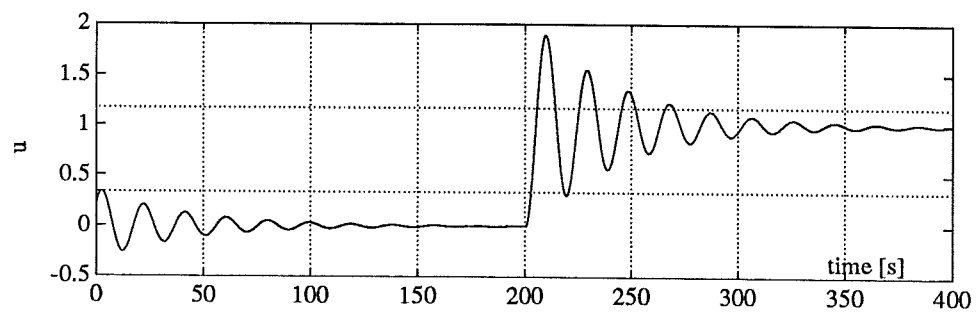
Sampeltiden för PID-regulatorn lät vi stå kvar på 0.3 sekunder och för Novatune sattes den till 1.5 med ett $KD=3$. Med denna inställning gav Novatune en stor styrsignalvarians varför styrsignalstraffet PN sattes till 18. Även här visade det sig att Novatune klarar laststörningar bättre än PID-regulatorn. Maximala amplituden på utsignalen vid en laststörning blev ungefär dubbelt så stor för PID-regulatorn som för Novatune.

14.4 Slutsatser

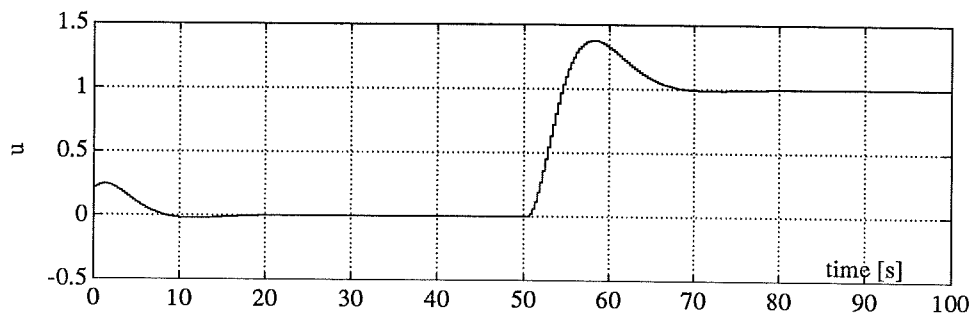
Överlag kan vi säga att denna process regleras bättre av Novatune än PID med dominant polplacering. Redan vid små ändringar i processdynamiken är det stor skillnad på resultatet. PID med dominerande polplacering ger ett mycket slängigt system och har svårt att uppfånga laststörningar på ett önskvärt sätt.

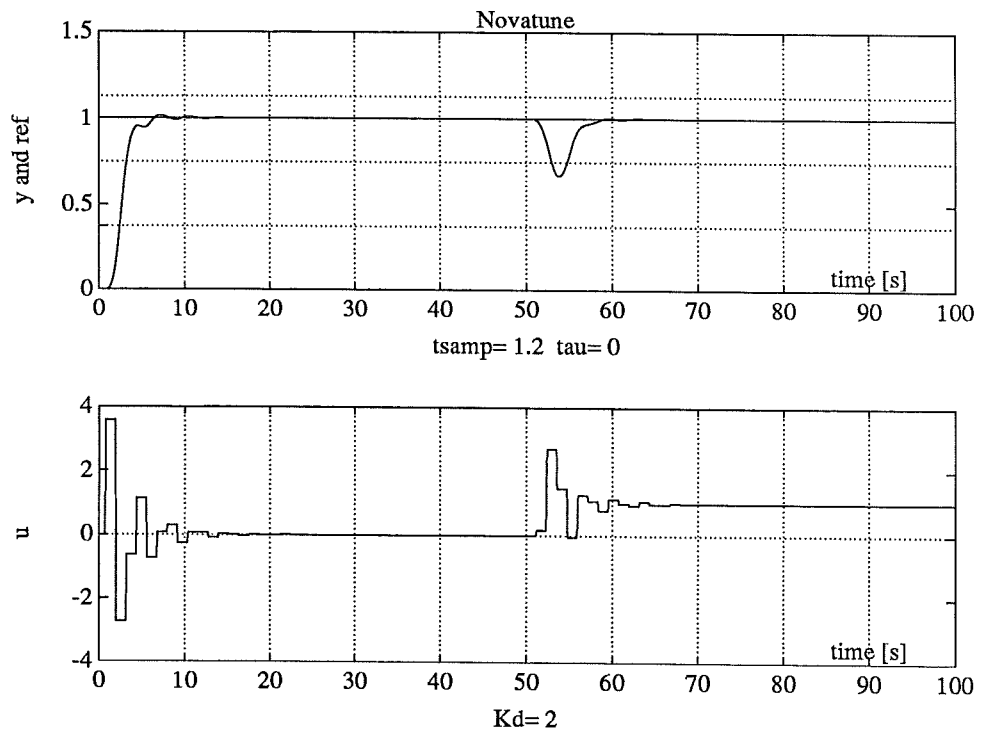


Figur 14.1 PID med dimensionering för ekv 14.2 med en verklig process ekv 14.3. Observera tidsskalan.



Figur 14.2 PID med dimensionering för "rätta" systemet 14.3. Även här blir det problem med laststörningen. Sampeltiden $T_s = 0.3$





Figur 14.3 Novatune med $NA=3$, $NB=3$, $KD=2$ och $T_s=1.2$ och systemet 14.3. I detta fall är det en klar fördel med en adaptiv regulator (jämför de båda tidigare figurerna).

15. Process 7

$$G(s) = \frac{K\omega_0^2}{s(s^2 + 2\zeta\omega_0s + \omega_0^2)} e^{-sL} \quad (15.1)$$

Detta stegsvar ser ungefär ut som stegsvaret för en överföringsfunktion med två reella poler och en i origo.

15.1 Inställning av Novatune

Enligt guiden är en standardinställning på Novatune $NA=NB=3$ och $KD=1$ om systemet saknar dödtid. Vi har tidigare konstaterat att KD bör väljas större än 1 även för system utan dödtid om

- processen är integrerande.
- processen har oscillerande poler.

I detta fall har vi både oscillerande poler och integrerande process. Som nominellt system valde vi

$$G(s) = \frac{1}{s(s^2 + 1.4s + 1)} \quad (15.2)$$

För att få en mindre känslig regulator valde vi en sampeltid som är något längre än vad vi tidigare rekommenderat för integrerande processer (se avsnitt 8.2). Sampeltiden sattes till 1.3 sekunder med $KD=2$.

15.2 Nominellt system utan dödtid

Funktionen `dompidz` ger för den nominella processen negativt värde på derivatadeln T_d . För att få bukt med problemet kan man antingen öka sampeltiden eller minska den specificerade dämpningen ζ . Vi valde att testa två varianter av dimensionering för PID-regulatorn.

- 1 Sampeltiden valdes till $T_s=0.3$ och den specificerade dämpningen ζ till 0.3. Detta gav $K = 0.51$, $T_i=7.1$ och $T_d=0.2$. I detta kapitelns tabeller benämns denna regulator PID1.
- 2 T_s sattes till 1 och den specificerade dämpningen ζ till 0.7. Detta gav $K = 0.26$, $T_i=8.3$ och $T_d=0.01$. I detta kapitelns tabeller benämns denna regulator PID2.

Resultatet från simuleringar med de båda varianterna av PID dimensionering redovisas i tabell 15.1. Variant (2) tycks vara den som är minst känslig för dynamikförändringar. Som jämförelse kan nämnas att känsligheten för denna är $S=1.9$ medan för inställningen (1) är $S=2.5$ (en definition av S ges i kapitel 9). En Nyquistkurva för (2) visas i figur 15.1. Vi provade även att ställa in PID-regulatorn med hjälp av Ziegler-Nichols självsvängningsmetod. För det återkopplade systemet med en P-regulator ges överföringsfunktionen av

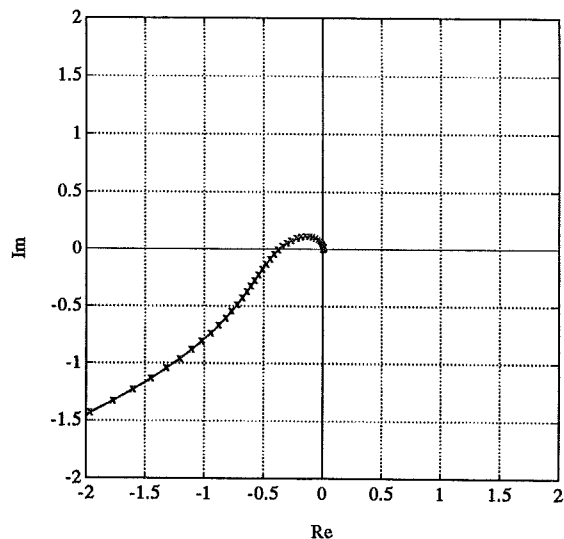
$$G(s) = \frac{K_u}{s^3 + 1.4s^2 + s + K_u}$$

som har poler med rötter i $\pm i$ om $K_u=1.4$. Periodtiden för självsvängningen är då $2\pi \approx 6.2$. Detta ger enligt Ziegler-Nichols $K = 0.84$, $T_i=3.1$ och $T_d=0.775$. Dimensioneringen med Ziegler-Nichols är den med högst förstärkning vilket resulterar i att den slår ner laststörningar bättre än dimensionering (1) och (2). Emellertid är överslängerna lite väl stora, vilket framgår av tabellen.

Tabell 15.1 Simuleringar med det nominella systemet 15.2.

	STEP					LOAD				Not
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	
PID1	0.27	2.5	13	4.0	2.0	2.2	5.0	14	1.0	1
PID2	0.04	5.7	8.4	5.1	1.1	3.5	6.9	31	1.0	2
Ziegler	0.80	1.5	13	4.8	-24	1.3	3.8	5.6	1.5	3
Nova	0.02	2.1	3.5	2.3	1.0	1.3	3.7	4.4	1.0	4

1. Stegsvär och laststörning för PID (1). PID-regulatorn är dimensionerad med dompidz. Sampeltiden, $T_s=0.3$ och den specificerade dämpningen $\zeta=0.3$.
2. Stegsvär och laststörning för PID (2). PID-regulatorn är dimensionerad med dompidz. Sampeltiden, $T_s=1$ och den specificerade dämpningen $\zeta=0.7$.
3. Stegsvär och laststörning PID. PID-regulatorn dimensionerad med hjälp av Ziegler-Nichols självsvängningsmetod som gav $K=0.84$, $T_i=3.1$ och $T_d=0.775$.
4. Stegsvär och laststörning för Novatune. Novatune har parametrarna $NA=3$, $NB=3$, $KD=2$ och $T_s=1.3$.



Figur 15.1 Nyquistkurva då PID-regulatorn dimensionerats med dompidz. Processen är den nominella (ekv 15.2), sampeltiden, $T_s=1$ och specificerad dämpning $\zeta=0.7$.

15.3 Dynamikförändringar

Vid test av hur regulatorerna beter sig vid dynamikförändringar ändrade vi processens dämpning ζ till 0.3 vilket ger systemet

$$G(s) = \frac{1}{s(s^2 + 0.6s + 1)} \quad (15.3)$$

Utan att göra ny dimensionering testade vi PID-dimensionering (1) och (2) ovan. Endast dimensionering (2) gav en acceptabel reglering. Novatune å andra sidan klarar denna dynamikförändring utan problem. Vid dimensionering av PID-regulator för ekvation 15.3 med hjälp av dompidz var vi tvungna att öka sampeltiden till 3 innan den levererade ett positivt T_d . Att minska den specificerade dämpningen ζ hjälpte inte. I tabell 15.2 har den översta PID-regulatorn dimensionerats för systemet 15.3 medan PID2 dimensionerats för 15.2. Förutom de två regulatorerna som dimensionerats med dompidz har vi testat en inställning med Ziegler-Nichols samt en RST-regulator dimensionerad enligt polplaceringsprincipen. Sist i tabellen visas resultatet för Novatune. Simuleringarna redovisas även i figurerna 15.2 till 15.5.

Tabell 15.2 Simuleringar med dåligt dämpat system ekv 15.3.

	STEP					LOAD				
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	Not
PID	0.14	3.7	12	4.1	1.5	4.2	6.1	29	1.0	1
PID2	0.03	8.6	11	5.1	1.1	3.7	5.3	32	1.0	2
Ziegler	0.52	1.9	22	5.9	-39	2.4	4.2	14	1.6	3
Polpd	0.04	2.1	3.0	2.0	1.1	0.5	2.6	1.9	1.0	4
Nova	0.01	1.8	3.5	2.4	1.0	1.8	3.5	5.7	1.0	5

1. Stegsvvar och laststörning för PID. PID-regulatorn dimensionerad med dompidz. Sampeltiden, $T_s=3$ och specificerad dämpning $\zeta=0.7$.
2. Stegsvvar och laststörning för ekv 15.3 med dimensionering för ekv 15.2. PID-regulatorn dimensionerad med dompidz. Sampeltiden, $T_s=1$ och specificerad dämpning $\zeta=0.7$.
3. Stegsvvar och laststörning för PID. Ziegler-Nichols självsvängningsmetod som gav $K=0.36$, $T_i=3.1$ och $T_d=0.775$.
4. Stegsvvar och laststörning för RST-regulator.. Polplaceringsmetoden med specificerad dämpning $\zeta=0.7$ och $\omega=1$. Sampeltiden $T_s=0.3$.
5. Stegsvvar och laststörning för Novatune. Novatune har parametrarna $NA=3$, $NB=3$, $KD=2$ och $T_s=1.3$.

Vid en jämförelse mellan de två första raderna i tabell 15.2 ser man att PID2 som är dimensionerad för process 15.2 är väl så bra som regulatorn PID, vilken är dimensionerad för rätt process, det vill säga 15.3. Observera också att RST-regulatorn är den som ger klart bäst resultat. Ziegler-Nichols inställningen är snabb men alldeles för slängig.

15.4 Simuleringar då systemet innehåller dödtid

Utgående från den nominella processen ekvation 15.2 tål Novatune att en dödtid på en halv sekund smyger sig in i sytemet. PID-regulatorn (2) klarar motsvarande dödtid på 1 sekund. Används dimensionering (1) är den mycket känslig för dödtidsökningar. Vi valde att testa regulatorerna dels på

$$G(s) = \frac{1}{s(s^2 + 1.4s + 1)} e^{-2s} \quad (15.4)$$

och dels på samma process med sämre dämpning

$$G(s) = \frac{1}{s(s^2 + 0.6s + 1)} e^{-2s} \quad (15.5)$$

Dompiz gav oss något förvånande positiva regulatorparametrar, något som vi inte lyckades få när systemet saknade dödtid. Detta har vi inte fördjupat oss i. Det visade sig också att Novatune reglerar bättre med många parametrar (NA=4, NB=4). Tyvärr uppstår problemet att man måste ha tillräckligt med excitation till systemet. Simuleringsresultaten för systemet 15.4 visas i tabell 15.3.

Tabell 15.3 Simuleringar med dödtid ekv 15.4

	STEP					LOAD				Not
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	
PID	0.04	8.2	14	8.7	1.0	5.7	11	72	1.0	1
Nova	0.08	2.3	16	5.5	1.1	3.3	7.3	16	1.1	2

1. Stegsvär och laststörning för ekv 15.4. PID-regulatorn är dimensionerad med dompiz. Sampeltiden, $T_s=1$ och den specificerade dämpningen $\zeta=0.7$.
2. Stegsvär och laststörning för ekv 15.4. Novatune med $T_s=1.5$, $KD=3$, $NA=4$, $NB=4$

Motsvarande simuleringar på ekvation 15.5 visas i tabell 15.4.

Tabell 15.4 Simuleringar med dödtid och dålig dämpning (ekv 15.5)

	STEP					LOAD				Not
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	
PID	0.04	6.0	11	7.3	1.1	5.5	9.3	52	1.0	1
Nova	0.03	1.6	4.8	4.1	1.0	4.1	7.0	16	1.0	2

1. Stegsvär och laststörning för PID. PID-regulatorn är dimensionerad med dompiz. Sampeltiden, $T_s=1$ och specificerad dämpning $\zeta=0.7$.
2. Stegsvär och laststörning för Novatune. Novatune har parametrarna $NA=4$, $NB=4$, $KD=3$ och $T_s=1.4$ och ett styrsignalstraff $PN=15$.

Som framgår av de båda tabellerna ovan slår Novatune ner laststörningar bättre än PID-regulatorn.

15.5 Bruskänslighet

Simuleringar med processbrus gav resultatet att Novatune lyckas eliminera bruset bättre än PID-regulatorn, men till priset av en större styrsignalvarians. Detta stämmer väl överens med vad vi tidigare funnit för integrerande processer. Resultaten vid simuleringar på den nominella processen 15.2 visas i tabell 15.5 samt i figurerna 15.6 och 15.7.

Tabell 15.5 Processbrus på ekv 15.2.

	E(y)	V(y)	V(ys)	V(Δu)	Cv	Not
PID1	-0.03	1.2	1.2	0.005	0.24	1
PID2	-0.03	4.7	4.7	0.027	0.16	2
Nova	-0.02	0.19	0.20	1.5	0.94	3

1. PID-regulator med $T_s=0.3$ dimensionerad med önskad dämpning $\zeta=0.3$.
1. PID-regulator med $T_s=1$ dimensionerad med önskad dämpning $\zeta=0.7$.
3. Novatune med $NA=NB=3$, $KD=2$ och $T_s=1.3$.

Samma simulering gjordes på ett system med dålig dämpning. Varianterna blev något större. Se tabell 15.6.

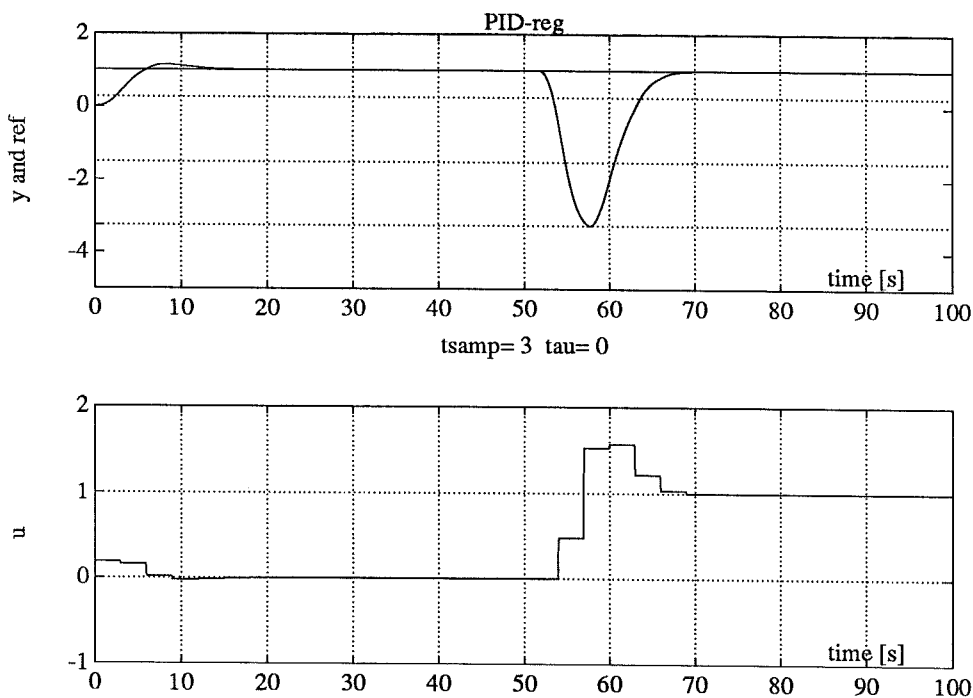
Tabell 15.6 Processbrus på ekv 15.3.

	E(y)	V(y)	V(ys)	V(Δu)	Cv	Not
PID2	-0.04	4.1	4.1	0.037	0.19	1
PID	-0.04	5.8	5.8	0.34	0.19	2
Nova	-0.029	0.46	0.46	2.8	1.3	3

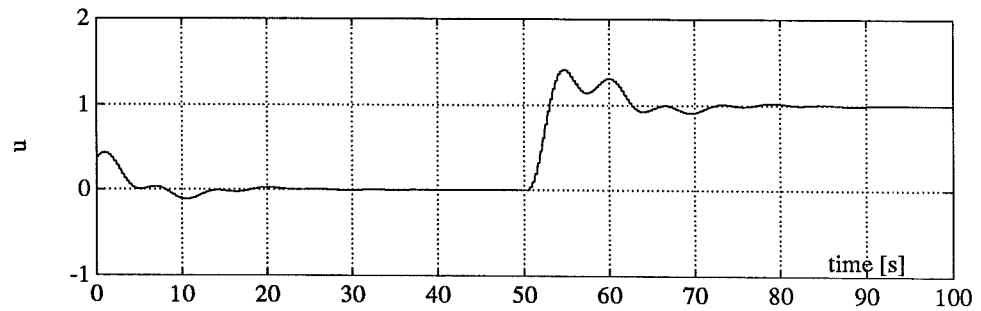
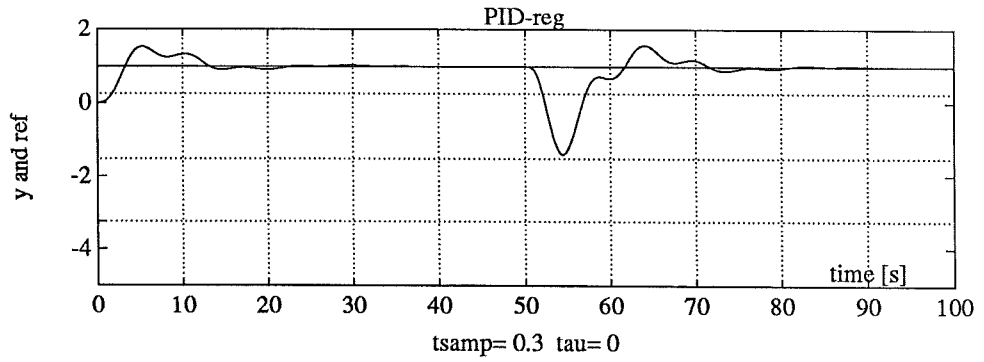
1. PID-regulator med $T_s=1$ och $\zeta=0.7$ dimensionering gjord för process ekv 15.2 med hjälp av dompidz
2. PID-regulator med $T_s=3$ och $\zeta=0.7$ dimensionering med hjälp av dompidz. Att sampeltiden är så lång beror på att med en kortare sampeltid gav dompidz negativt värde på T_d .
3. Novatune med $NA=NB=3$, $KD=2$ och $T_s=1.3$.

15.6 Slutsatser

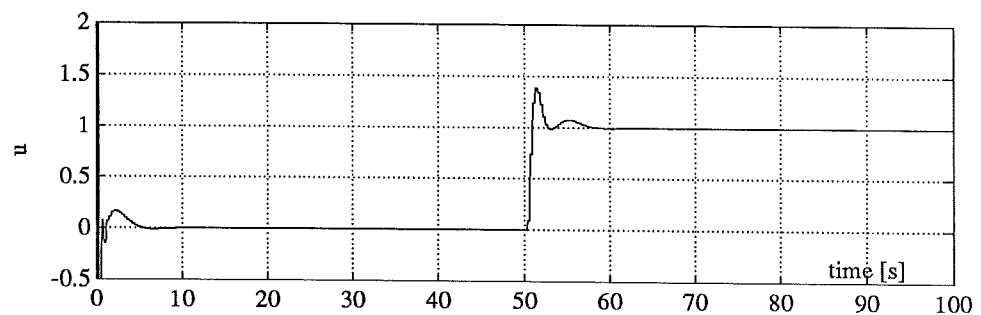
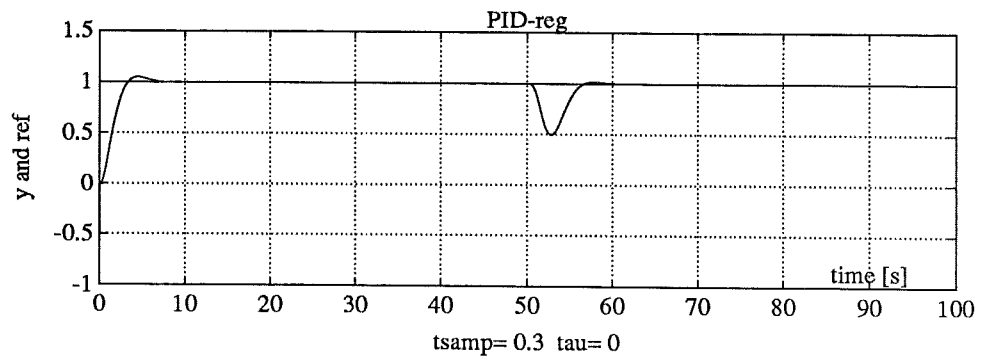
Vid reglering av denna process bör man noga tänka igenom vilken dimensioneringsmetod som är lämpligast. En lite längre sampeltid än brukligt ger visserligen en större inverkan av en laststörning, men man vinner en större stabilitet. Finns det ej dödtid i systemet är Novatune att föredra eftersom den tål dynamikförändringar bra. I annat fall välj en PID-regulator eftersom Novatune är svår att ställa in om man inte har god kännedom om processens dödtid. Eftersom processen är relativt komplicerad kan man vinna på att välja en mer komplex regulator än PID.



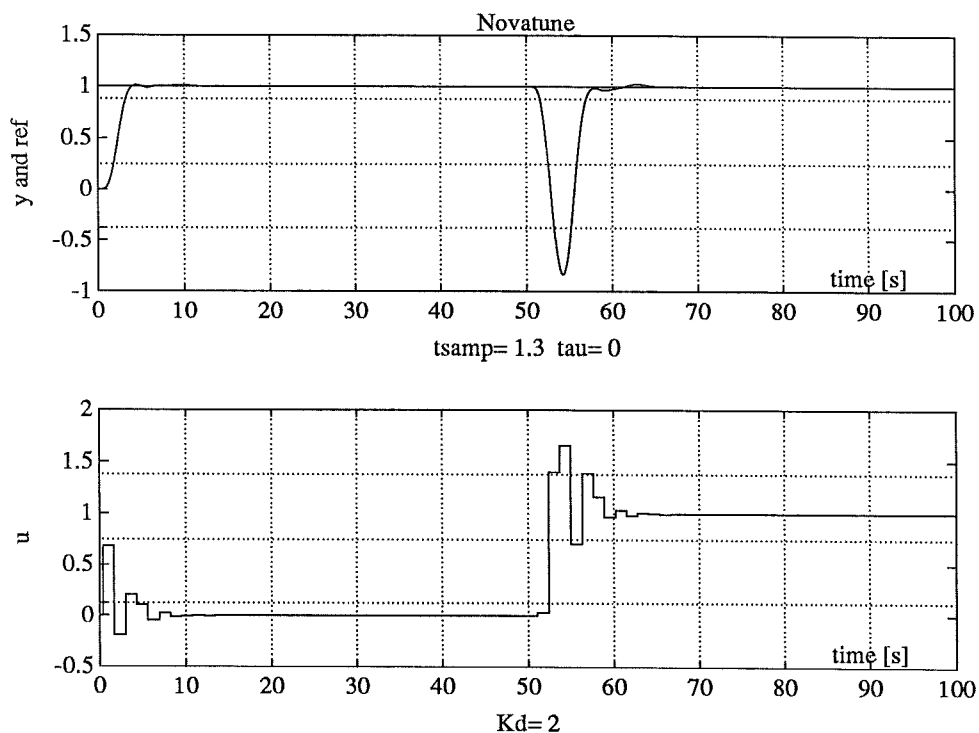
Figur 15.2 Simulering med PID-regulator på överföringsfunktion 15.3, dimensionering med dompidz $\zeta=0.7$ och $T_s=3$. Den långa sampeltiden beror på att för kortare sampeltider gav dompidz negativa värden på K , T_i och T_d även då kravet på specificerad dämpning lättades.



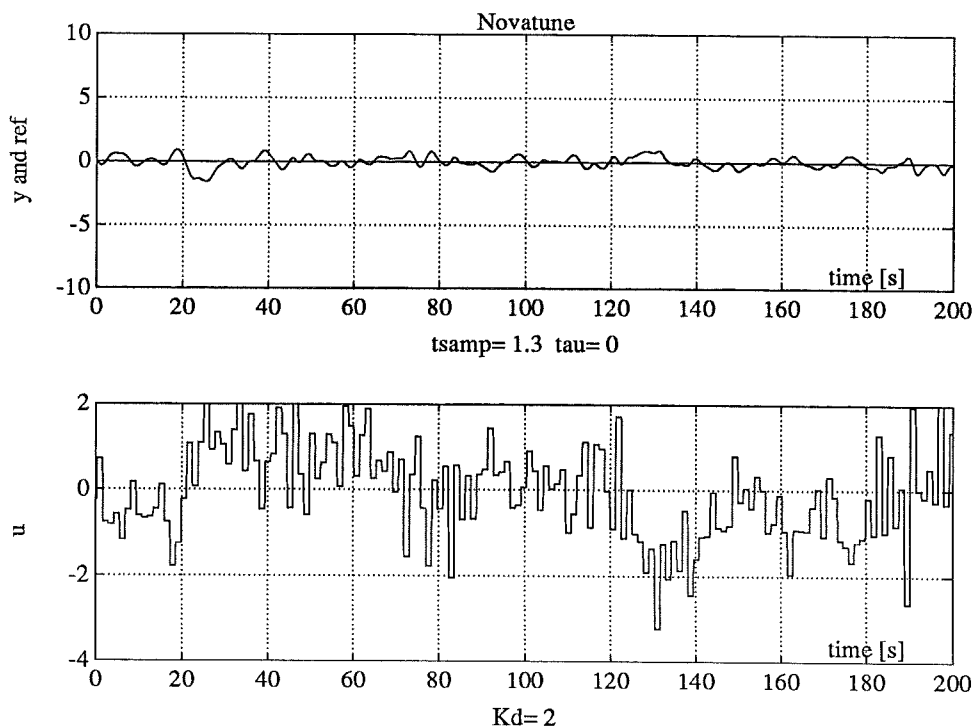
Figur 15.3 Simulering med PID-regulator på överföringsfunktion 15.3. Regulatorn är dimensionerad med hjälp av Ziegler-Nichols självsvängningsmetod som gav $K=0.36$, $T_i=3.1$ och $T_d=0.775$.



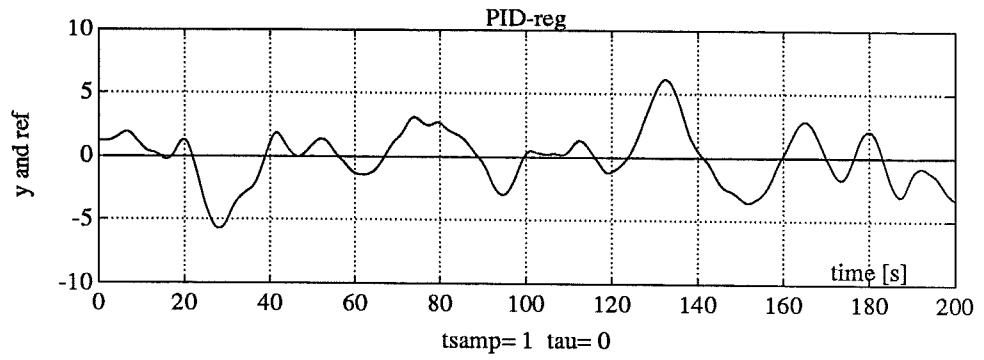
Figur 15.4 Simulering med polplaceringsmetoden på överföringsfunktion 15.3 med specificerad dämpning $\zeta=0.7$ och $\omega=1$. Sampeltiden $T_s=0.3$.



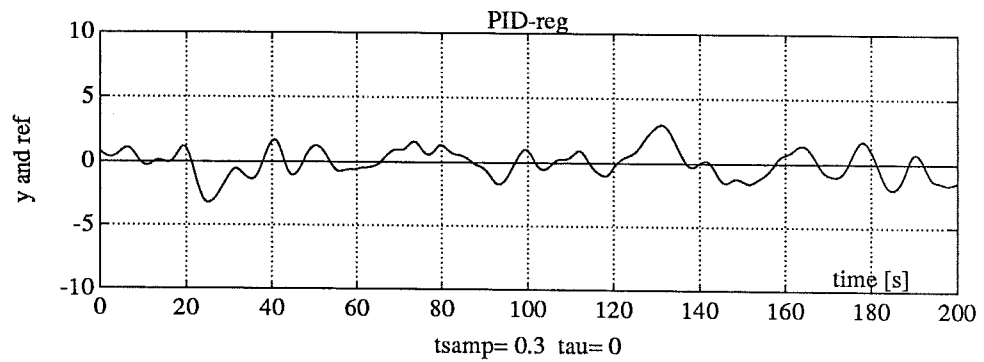
Figur 15.5 Simulering med Novatune på överföringsfunktion 15.3. $NA=NB=3$, $KD=2$ och sampeltiden $T_s=1.3$. Jämför med figurerna 15.2-4 (observera skalorna).



Figur 15.6 Processbrus på överföringsfunktionen ekv 15.2 med Novatune som har $NA=NB=3$, $KD=2$ och sampeltiden $T_s=1.3$.



Figur 15.7 Processbrus på överföringsfunktionen ekv 15.2 med PID-regulator då sampeltiden $T_s=1$ och den specificerade dämpningen $\zeta=0.7$.



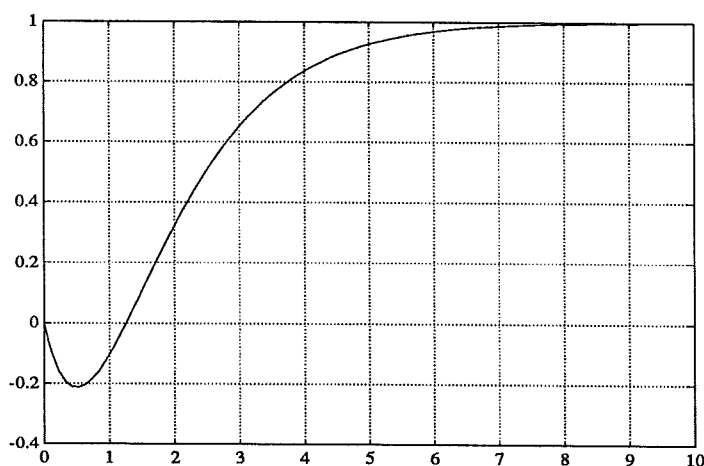
Figur 15.8 Processbrus på överföringsfunktionen ekv 15.2 med PID-regulator då sampeltiden $T_s=0.3$ och den specificerade dämpningen $\zeta=0.3$.

16. Process 8

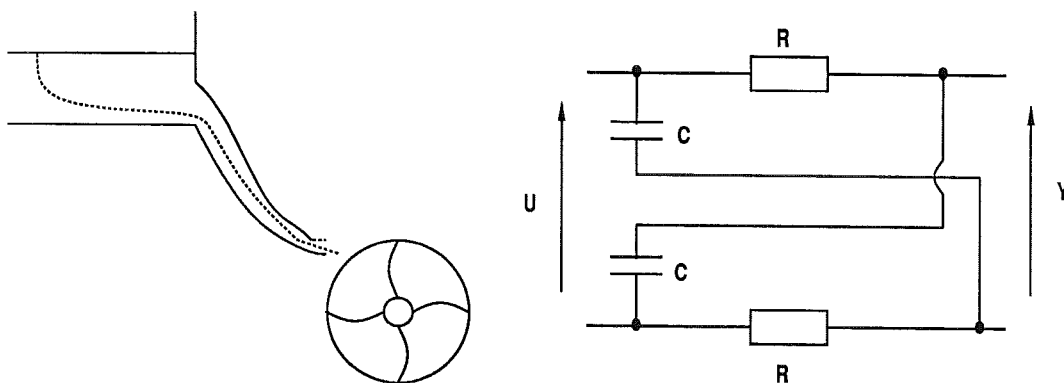
Hittills har våra processer enbart haft poler och nollställen i vänster halvplan. System med denna egenskap kallas minimum-fas-system. Process 9 är ett system av typen

$$G(s) = \frac{K(1 - sT_1)}{(1 + sT_2)(1 + sT_3)} e^{-sL} \quad (16.1)$$

med ett nollställe i höger halvplan, och benämns ofta icke minimum-fas-system. Stegsvaret framgår av figur 16.1, där vi observerar att utsignalen till en början går åt "fel håll".



Figur 16.1 Stegsvaret för ett system som inte är minimum-fas.



Figur 16.2 Två exempel på icke minimum-fas-system. Vattenturbin och en elektrisk krets.

I figur 16.2 visas två exempel på system med nollställen i höger halvplan. Dels en vattenturbin där sambandet mellan maximalt tillgänglig effekt och turbinpådrag representerar ett icke minimum-fas-system, och dels en elektrisk krets. Kretsens har överföringsfunktionen

$$G(s) = \frac{(1 - sT)}{(1 + sT)}$$

där tidskonstanten $T=RC$.

16.1 Inställning av Novatune

Vi valde till att börja med det nominella systemet

$$G(s) = \frac{(1-s)}{(1+s)(1+s)} \quad (16.2)$$

vars stegsvar visas i figur 16.1. Installationsanvisningen ger ingen riktig vägledning vad gäller icke min-fas-system, men ett rimligt krav bör vara att stegsvaret nått ca 70% av stegets storlek efter tiden $KD \cdot T_s$. Om vi följer rekommendationen med $KD=1$ för ett system utan dödtid får vi välja en sampeltid på 3 sekunder. Denna inställning fungerar tillfredsställande så länge vi kan vara säkra på att ingen dödtid smyger sig in i systemet. Minsta lilla dödtid medför att sampeltiden måste justeras. För att råda bot på detta valde vi istället en inställning med $KD=2$ och $T_s=1.5$. Parametrarna NA och NB sattes till 2 resp 3.

16.2 Nominellt system utan dödtid

Av figur 16.3 och tabell 16.1 framgår att de båda ovan nämnda inställningarna av Novatune ger liknande resultat om dödtiden är noll.

Tabell 16.1 Simuleringar med nominella systemet ekv 16.2.

	STEP					LOAD				Not
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	
PID	0.05	1.5	4.8	2.7	1.1	0.81	2.8	2.5	1.2	1
Nova	0.01	1.9	11	5.9	1.0	0.83	3.3	4.0	1.1	2
Nova	0.01	2.4	6.1	4.2	1.0	0.99	2.7	2.7	1.4	3

1. PID-regulatorn dimensionerad för ekv 16.2 (nominella systemet utan dödtid). Sampeltiden $T_s=0.3$.
2. Novatune med $NA=2$, $NB=2$, $KD=1$ och $T_s=3$.
3. Novatune med $NA=2$, $NB=3$, $KD=2$ och $T_s=1.5$.

Novatuneinställningen enligt rad 3 i tabell 16.1 (Nova2) ger en regulator som klarar sig bra även om en dödtid på en sekund skulle dyka upp i systemet. För en gångs skull kan vi dessutom konstatera att Novatune visar sig mindre känslig än PID-regulatorn vid denna typ av systemförändring. I figur 16.4 har PID-regulatorn dimensionerats för det nominella systemet 16.2 varefter en dödtid på en sekund lagts till systemet.

16.3 Dynamikförändringar

Vid test av hur dynamikförändringar påverkar systemet gjordes två olika simuleringar. I båda fallen utgick vi från det nominella systemet 16.2.

1. Den ena polen gjordes 4 gånger långsammare.
2. Nollstället flyttades från 1 till 0.5.

Det visade sig inte helt oväntat att Novatune är vida överlägsen på detta område, speciellt om nollstället är långsamt. Vid nollställesändringen blev PID-regulatorn instabil. Figur 16.5 visar denna simulering där vi också ser att Novatune reglerar ut laststörningen på 7-8 sampel. I tabell 16.3 ser vi att PID-regulatorn klarar sig utmärkt om den dimensioneras för det nya nollstället.

Tabell 16.2 Simuleringsresultat vid polförflyttning.

	STEP					LOAD				Not
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	
PID	0.39	3.2	28	7.4	3.0	0.5	5.4	4.4	2.1	1
PID	0.05	2.2	4.2	3.1	1.1	0.38	3.5	1.5	1.2	2
Nova	0.0	2.4	6.2	4.3	1.0	0.45	3.9	1.5	1.2	3

1. PID-regulator dimensionerad för ekv 16.2 (nominella systemet utan dödtid). Ena polen därefter flyttad till -0.25 . Sampeltiden $T_s=0.3$.
2. PID-regulatorn dimensionerad för det verkliga systemet.
3. Novatune med $NA=2$, $NB=3$, $KD=2$ och $T_s=1.5$.

Tabell 16.3 Simuleringsresultat då systemet har ett långsamt nollställe.

	STEP					LOAD				Not
	M	Tr	Ts	IAE	Iq	Amax	T	IAE	Iq	
PID	0.06	1.6	6.2	4.0	1.1	1.4	3.6	5.5	1.6	1
Nova	0.0	-	10	5.3	1.0	1.5	4.1	6.6	1.3	2

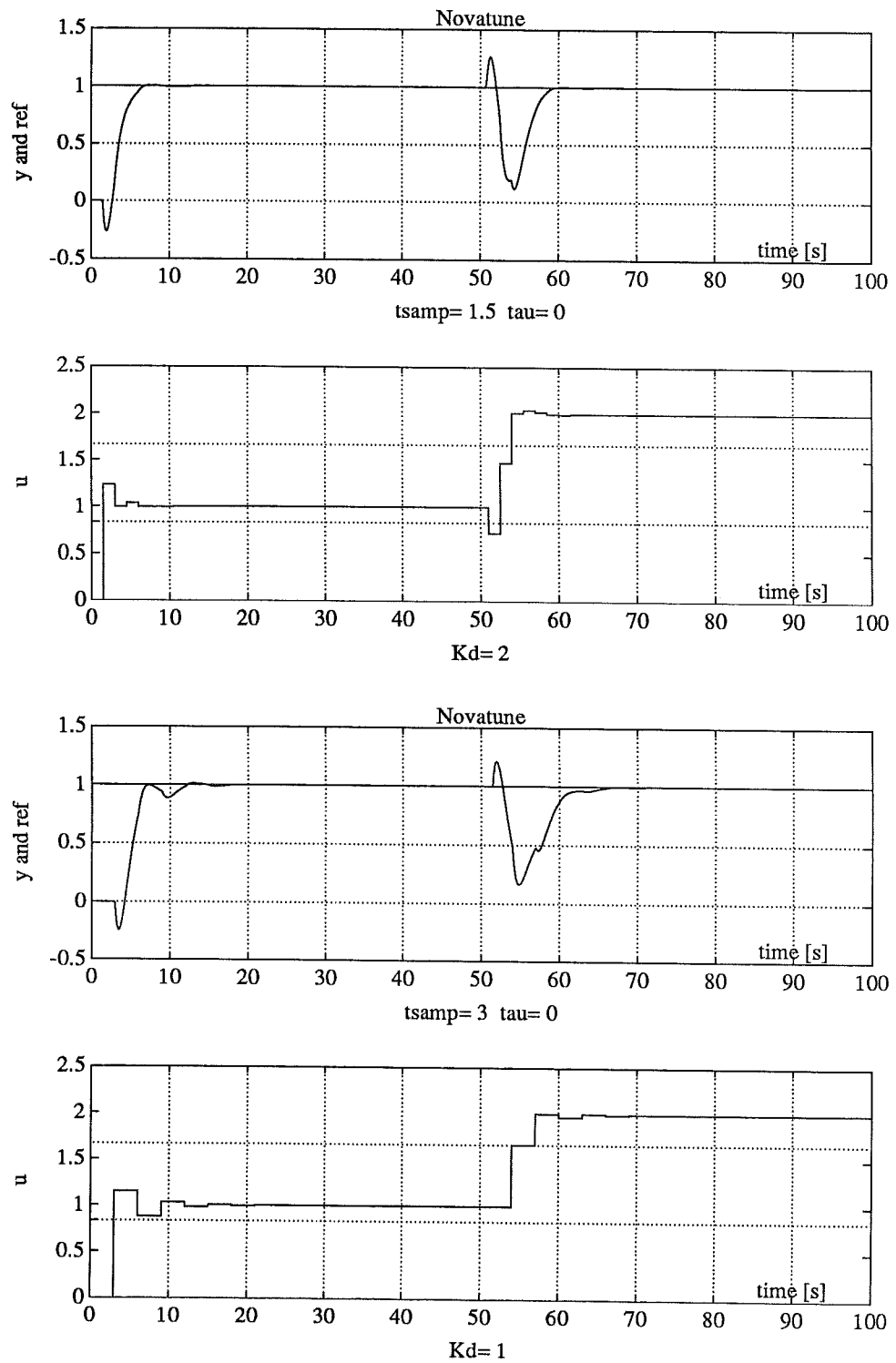
1. PID-regulatorn dimensionerad för det verkliga systemet. $T_s=0.3$.
2. Novatune med $NA=2$, $NB=3$, $KD=2$ och $T_s=1.5$.

16.4 Test med lång dödtid

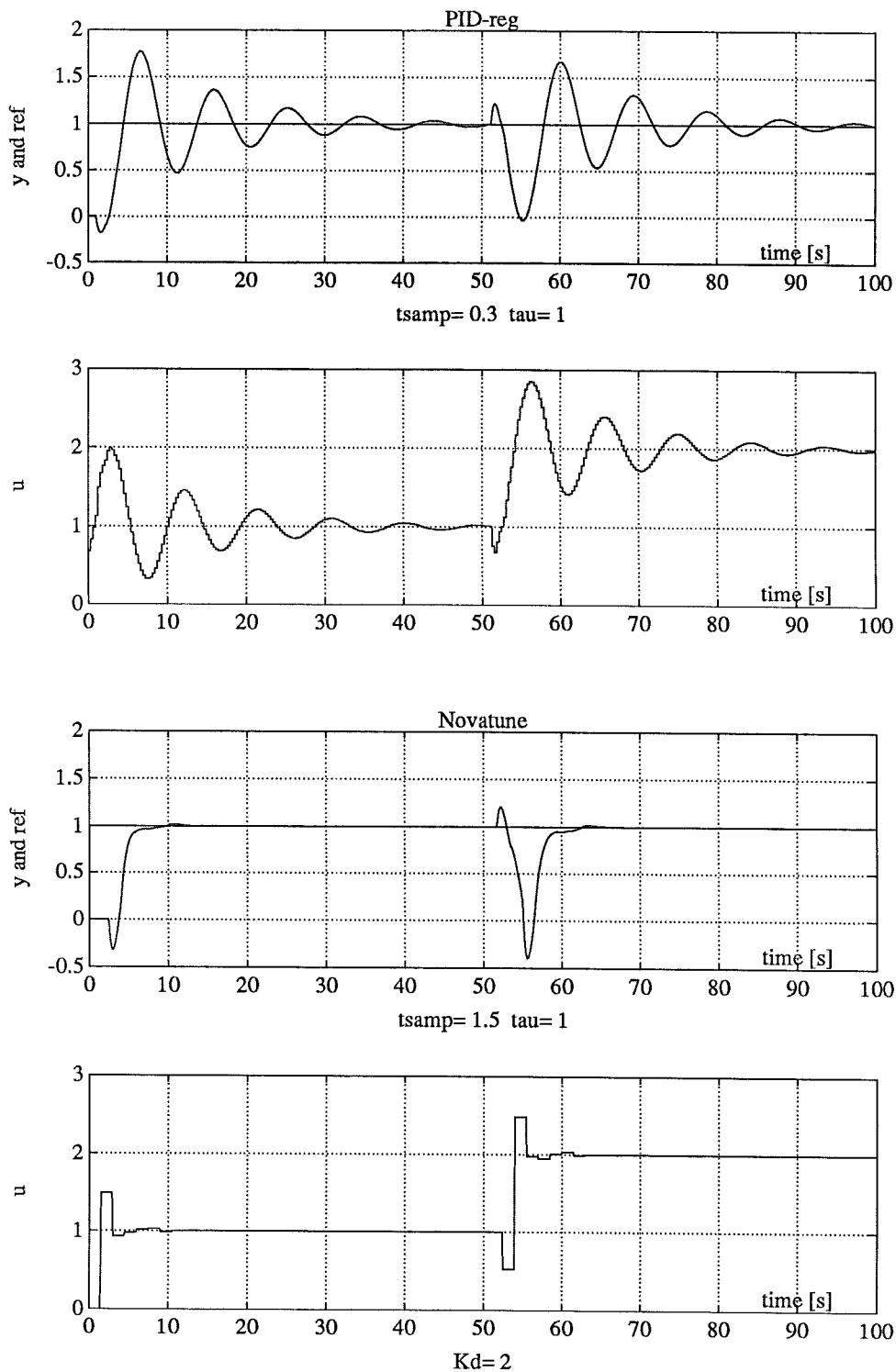
Här undersökte vi regulatorernas beteende när dödtiden sattes 5 gånger större än tidskonstanten. Figur 16.5 visar stegsvar och laststörning för de båda regulatorerna. Även här visar PID-regulatorn en större känslighet då dödtiden ökar. I simuleringen har vi valt sampeltiden på Novatune till 5 sekunder vilket ger en inställning som faktiskt tål en dödtidsökning på 1-2 sekunder.

16.5 Slutsatser

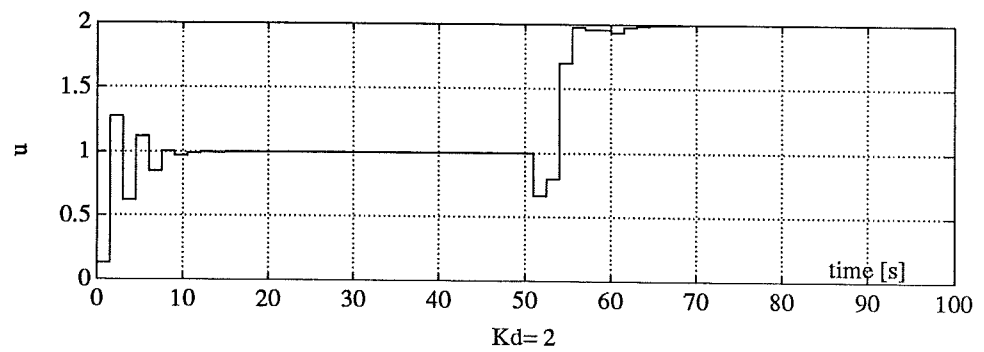
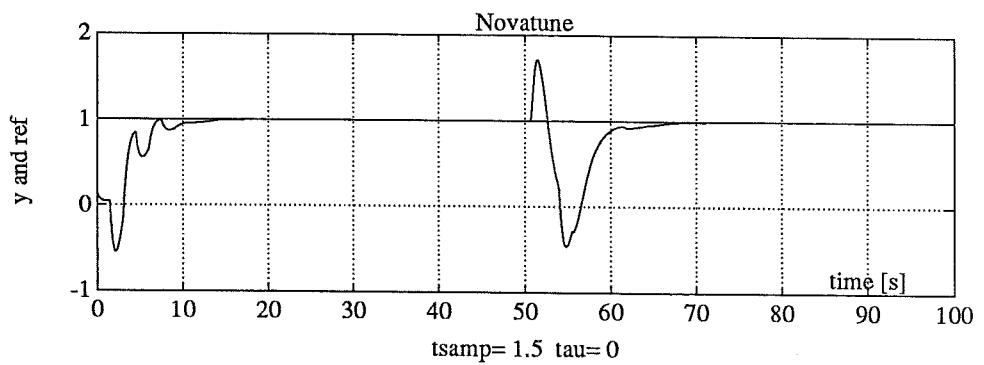
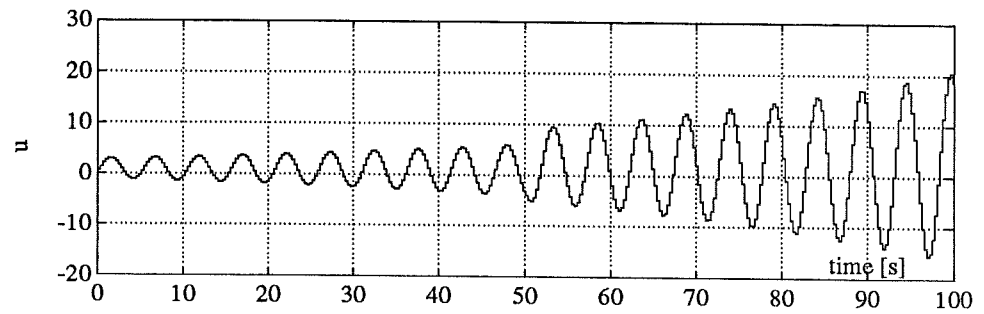
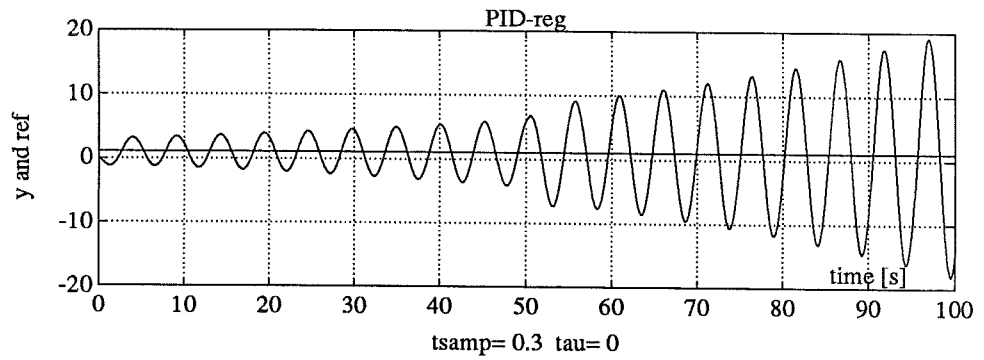
Rätt inställd visar Novatune en förvånansvärt stor robusthet mot dödtidsökningar. När tidskonstanterna varierar är Novatune som vanligt överlägsen. Samtidigt har PID-regulatorn svårt att klara ett långsamt nollställe om den inte dimensionerats för detta. Ett sätt att klara detta är att dimensionera den för "värsta fallet", alltså för ett nollställe nära imaginära axeln. Om nollstället sedan flyttar sig längre ut i höger halvplan medför detta inga problem. Sammantaget tycker vi dock att våra simuleringar visat att Novatune reglerar denna typ av process bättre än en konventionell PID-regulator.



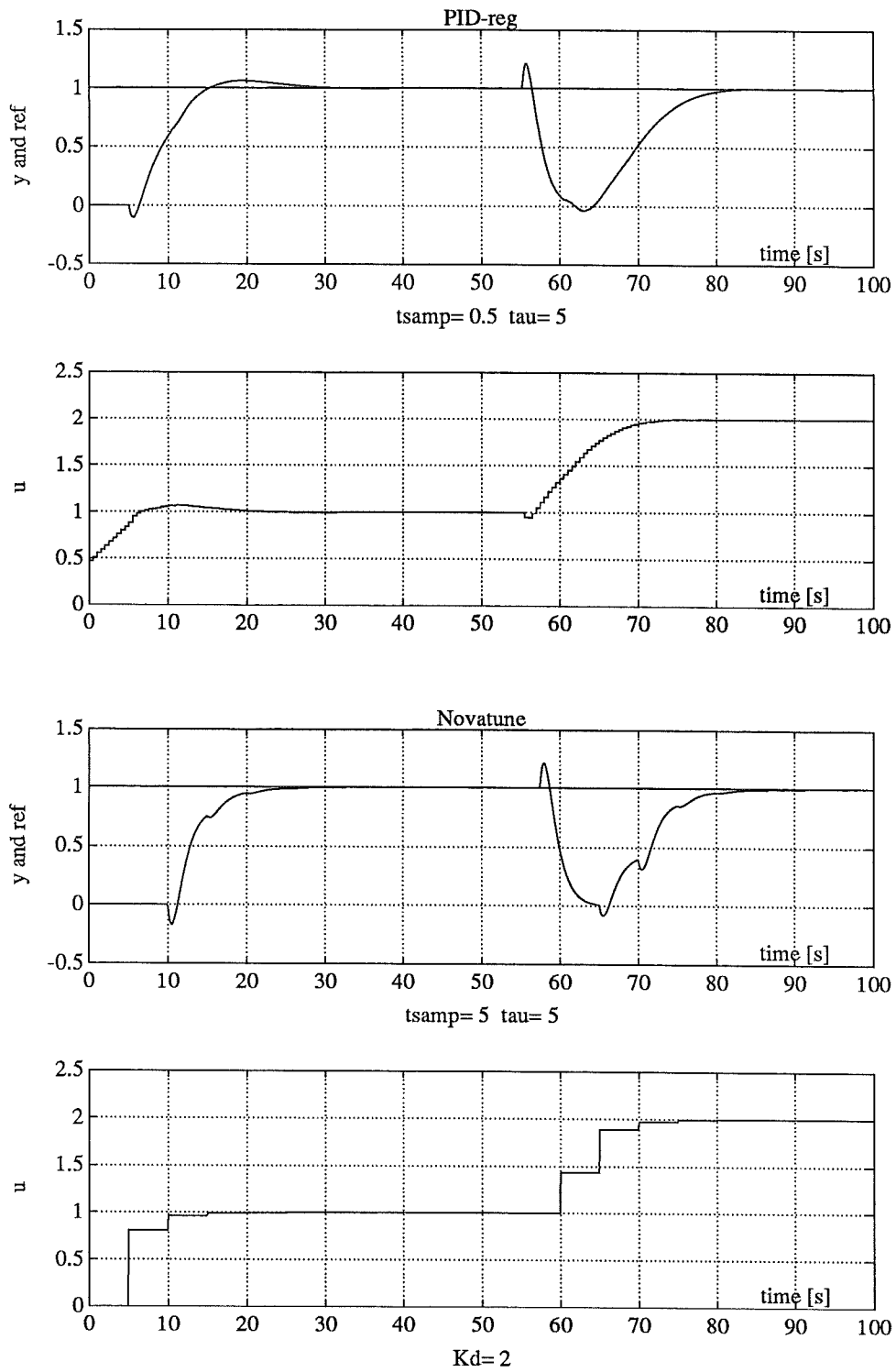
Figur 16.3 Novatune med två olika inställningar. Den styrda processen är ekv 16.2. Övre diagrammet: $NA=2$, $NB=3$, $KD=2$ och $T_s=1.5$. Undre diagrammet: $NA=2$, $NB=2$, $KD=1$ och $T_s=3$.



Figur 16.4 Regulatorerna har dimensionerats för nominella systemet 16.2. Efter dimensionering har en dödtid på en sekund smugit sig in i systemet. Tvärt emot tidigare erfarenheter visar Novatune mindre känslighet mot dödtidsändringar. PID-regulatorns sampeltid har här satts till 0.3 sekunder, men en robustare regulator kan fås med längre sampeltid.



Figur 16.5 Simulering då dimensionering gjorts med nollstället i 1, fast det i själva verket låg i 0.5. Överst PID med $T_s=0.3$ och underst Novatune med $NA=2$, $NB=3$, $KD=2$ och $T_s=1.5$.



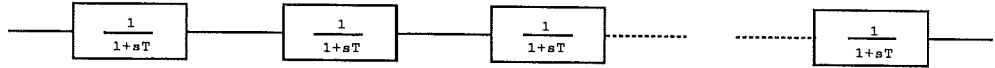
Figur 16.6 Stegsvär och laststörning för icke min-fas-system med 5 sekunders dödtid. PID-regulatorn har $T_s=0.5$.

17. Process 9

Vår sista process kan sägas vara en mängd seriekopplade första ordningens system. Överföringsfunktionen kan skrivas

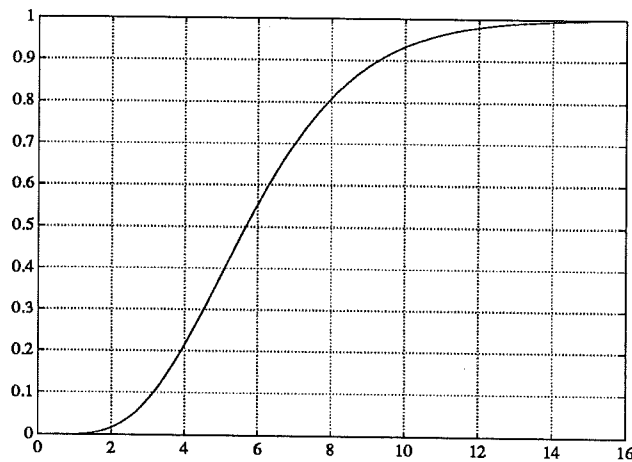
$$G(s) = \frac{K}{(1 + sT)^n} \quad (17.1)$$

där vi valt $4 \leq n \leq 8$ i våra tester.



Figur 17.1 n st seriekopplade första ordningens system.

Ett stegsvar för ett system av så hög ordning har ingen abrupt start utan smyger igång. I figur 17.2 ser vi ett exempel när $n=6$. Det är ofta mycket svårt att avgöra om systemet innehåller dödtid eller ej.



Figur 17.2 Stegsvaret för systemet 17.1 där $n=6$.

17.1 Inställning av Novatune

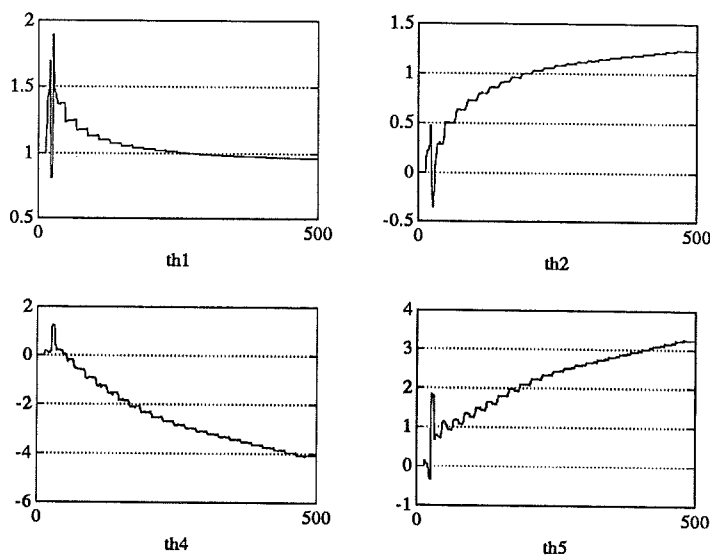
Gemensamt för alla ordningstal $4 \leq n \leq 8$ var att det var näst intill omöjligt att hitta en bra regulator om KD valdes mindre än 3. Om man utgår från ett väldefinierat stegsvar fann vi att följande strategi gav oss bra regulatorer

- KD sätts till 3.
- Sampeltiden väljs så att prediktionen sker då stegsvaret nått ca 50% av enhetssteget.

Eftersom det är vanskligt att uttala sig om ordningstal på denna typ av process valde vi vad som i installationsanvisningen benämns standardval med $NA=NB=3$.

17.2 Dynamikförändringar

Här gjorde vi två test. Dels provade vi att flytta de multipla polerna och dels ändrade vi på ordningstalet n . Vi valde att prova hur regulatorerna beter sig vid en ändring av tidskonstanten då systemet är av ordning 8. En liten ändring i tidskonstanten får stora konsekvenser eftersom 8 poler samtidigt rör sig. Vi dimensionerade regulatorerna då polerna låg i -1 och flyttade dem sedan till -0.8. I figur 17.4 visas resultatet för Novatune. Utan styrsignalstraff får vi en styrsignal som slår upp och ner i varje sampel. Genom att sätta $PN=18$ och behålla ursprunglig sampeltid får vi ett fullt acceptabelt resultat (de två nedre diagrammen). Motsvarande test av PID-regulatorn visas i figur 17.5. Eftersom det inte är lätt att uttala sig om ordningstalet provade vi hur regulatorerna klarar ett varierande eller okänt ordningstal n . Vi dimensionerade regulatorerna för en process av sjätte ordningen ($n=6$) och lät sedan n variera mellan 4 och 8. Det visade sig att PID-regulatorn klarar sig för alla $4 \leq n \leq 8$, medan Novatune får problem om inte ett styrsignalstraff införs. Vid dimensionering för ett system med $n=6$ och $T=1$ valde vi en sampeltid på 2 sekunder. När vi minskade ordningstalet n till 4 hade Novatune problem med insvängningen vilket visas i figur 17.3.



Figur 17.3 Novatunes θ -parametrar då systemet är 17.1 med $n=4$ och polerna placerade i -1. Dimensionering är gjord för ett system med ordningstal $n=6$. Novatune har svårt att "hitta processen".

Tabell 17.1 Processen 17.1 med ordningstal 8 och $T=1.25$.

	STEP					LOAD				Not
	M	T_r	T_s	IAE	I_q	Amax	T	IAE	I_q	
PID	0.18	8.4	32	13	1.4	0.79	15	12	1.3	1
Nova	0.04	8.2	17	12	1.0	0.73	13	8.0	1.1	2

1. PID-regulatorn dimensionerad för ekv 17.1 med $T=1$ och $n=8$, men vi har ändrat T till 1.25. Sampeltiden, $T_s=0.5$.
2. Novatune har parametrarna $NA=3$, $NB=3$, $KD=3$ och $T_s=2.6$ och styrsignalstraffet PN har satts till 18.

17.3 Bruskänslighet

Vi testade bruskänsligheten när processen såg ut som

$$G(s) = \frac{1}{(1+s)^8} \quad (17.2)$$

Vi har tidigare diskuterat Novatunes känslighet för mätbrus. Även här medför den höga förstärkningen att processbrus elimineras effektivt medan den uppvisar stor känslighet för mätbrus. Resultatet framgår av tabellerna nedan.

Tabell 17.2 Processbrus på ekv 17.2.

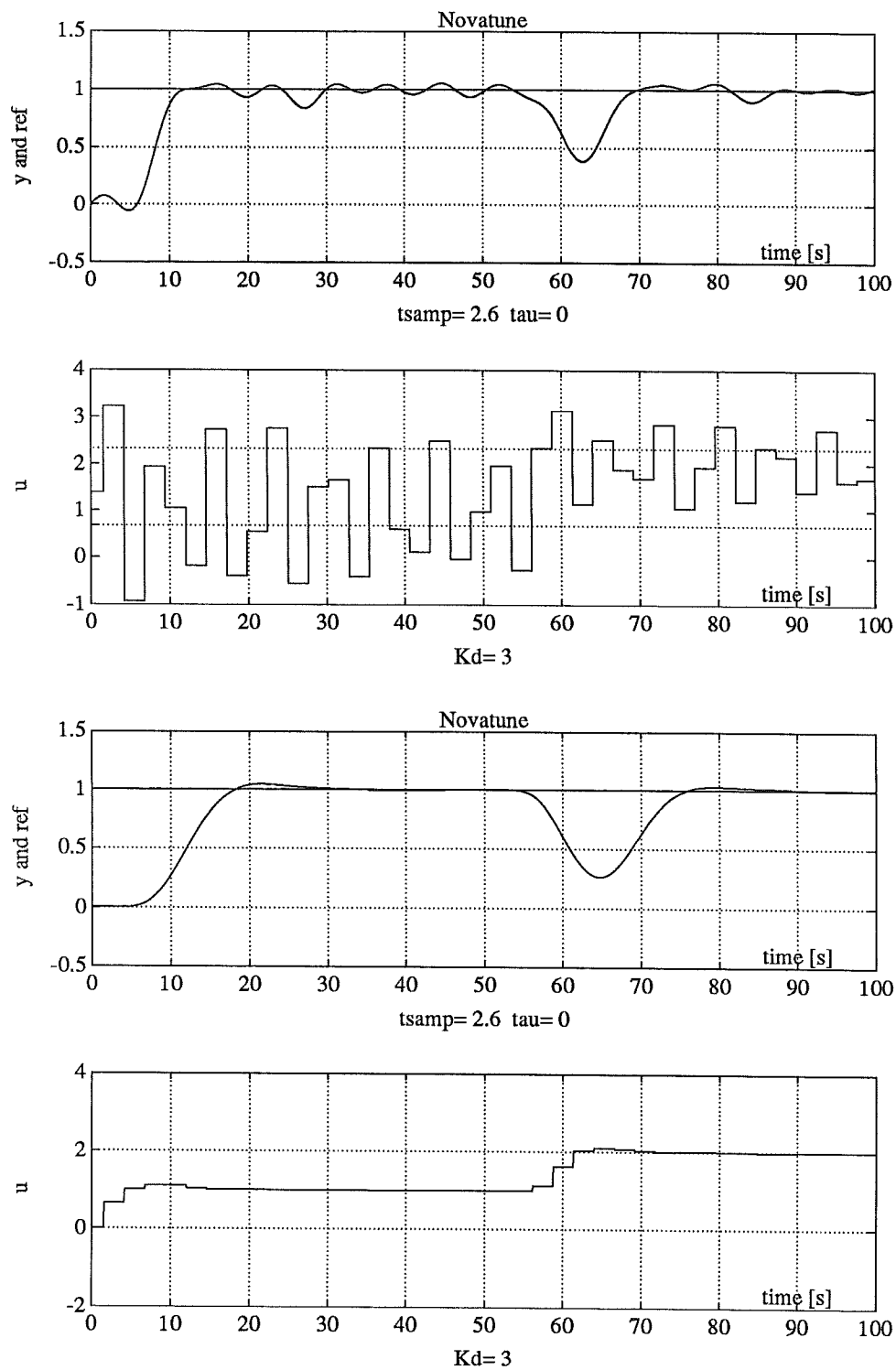
	E(y)	V(y)	V(ys)	V(Δu)	Cv	Not
PID	0.01	0.28	0.28	0.0012	0.069	1
Nova	0.014	0.15	0.15	0.29	0.21	2

1. PID-regulator med sampeltiden $T_s=0.5$.
2. Novatune med $NA=NB=3$, $KD=3$ och sampeltiden $T_s=2.6$.

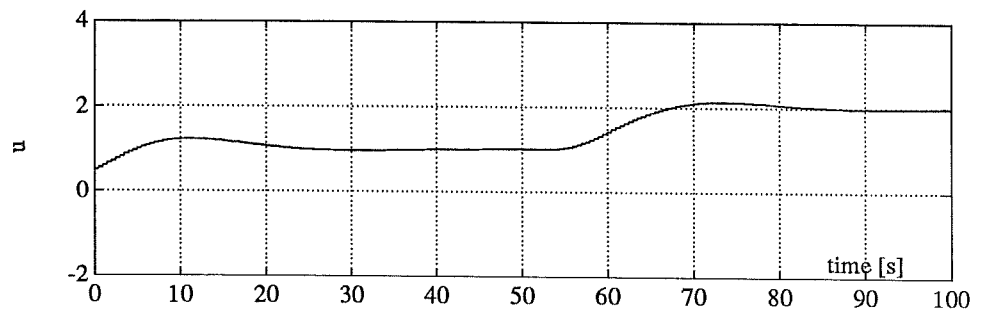
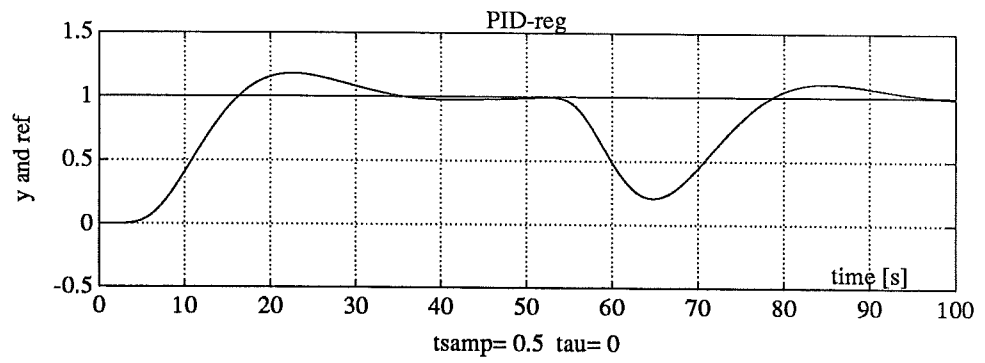
Tabell 17.3 Mätbrus på ekv 17.2.

	E(y) 10^{-3}	V(y) 10^{-3}	V(ys) 10^{-3}	V(Δu) 10^{-3}	Cv	Not
PID	1.8	0.059	0.059	22	0.30	1
PID	1.3	0.035	0.035	100	1.6	2
Nova	-4.8	2.8	2.8	280	0.20	3

1. PID-regulator med sampeltiden $T_s=0.5$.
2. PID-regulator med $T_s=0.2$. Den kortare sampeltiden medför lägre utsignalvarians men större varians i styrsignalen.
3. Novatune med $NA=NB=3$, $KD=3$ och $T_s=2.6$.



Figur 17.4 Novatune på systemet 17.2 med $n=8$ och polerna placerade i -0.8 . Dimensionering är gjord för ett system med polerna i -1 . Övre figuren visar Novatune utan styrsignalstraff. Den undre visar Novatune med styrsignalstraffet $PN=18$.



Figur 17.5 PID-regulator och systemet 17.2 med $n=8$ och polerna placerade i -0.8 . Dimensionering är gjord för ett system med polerna i -1 .

18. Generalized Predictive Control

Som vi har visat saknar Novatune robusthet mot dödtidsvariationer. I gengäld är den mindre beroende av rätt modellordning. Andra adaptiva regulatorer som till exempel polplaceraren visar motsatt beteende. I två artiklar som publicerades i Automatica 1987 beskriver Clarke, Mohtadi och Tuffs en ny metod, GPC, som visar sig vara robust mot såväl dödtidsvariationer som felaktig modellordning. Vi kommer i detta kapitel att sammanfatta den grundläggande principen samt några av de resultat som författarna kommit fram till.

18.1 Novatune-ett specialfall av GPC

Förlustfunktionen i Novatune ser ut som

$$J = E \left\{ (f(t+k) - u_c(t))^2 + \rho \Delta u(t)^2 \right\} \quad (18.1)$$

och tar alltså hänsyn till endast en term. Om man istället vid varje sampel gör flera prediktioner av ärvärdet, man också predikterar $f(t+N_1)$, $f(t+N_1+1)$, \dots , $f(t+N_2)$ blir förlustfunktionen

$$J = E \left\{ \sum_{k=N_1}^{N_2} [y(t+k) - u_c(t+k)]^2 + \sum_{k=1}^{N_2} \lambda(k) [\Delta u(t+k-1)]^2 \right\} \quad (18.2)$$

Om $N_1 = N_2 = 1$ blir förlustfunktionen exakt densamma som för Novatune.

18.2 GPC-algoritmen

I artikeln utgår man från modellen

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + C(q^{-1})\xi(t)/\Delta \quad (18.3)$$

Med hjälp av identiteten

$$1 = E_j(q^{-1})A\Delta + q^{-j}F_j(q^{-1}) \quad (18.4)$$

och ekvation 18.3 fås den optimala prediktorn för j steg framåt

$$\hat{y}(t+j|t) = G_j \Delta u(t+j-1) + F_j y(t) + E_j \xi(t+j) \quad (18.5)$$

där $G_j = E_j B$.

I Novatune görs som nämnts endast en prediktion, $\hat{y}(t+k|t)$ medan man i GPC predikterar $\hat{y}(t+j|t)$ där $N_1 \leq j \leq N_2$. För $j < k$ beror prediktionerna enbart på tillgängliga data, men för $j \geq k$ måste man göra något antagande om framtida styrsignaler. Dessa antaganden är hörnstenen i GPC. Ett sätt att göra flera prediktioner vore att ha en bank av prediktorer, en för varje

prediktionshorisont. Ett enklare och mer effektivt sätt är att rekursivt lösa de diofantiska ekvationerna så att E_{j+1} och F_{j+1} ges av E_j och F_j .

Antag att $N_1 = 1$ och $N_2 = N$. Modellen för framtida utsignaler ser då ut som:

$$\begin{aligned} y(t+1) &= G_1 \Delta u(t) + F_1 y(t) + E_1 \xi(t+1) \\ y(t+2) &= G_2 \Delta u(t+1) + F_2 y(t) + E_2 \xi(t+2) \\ &\vdots \\ y(t+N) &= G_N \Delta u(t+N-1) + F_N y(t) + E_N \xi(t+N) \end{aligned}$$

Ur formel ses att $\hat{y}(t+j)$ består av tre termer, en beror av framtida styrsignaler som ska beräknas, en beror av tidigare styrsignaler tillsammans med filtrerade utsignaler och den tredje av framtida brus. Eftersom det är svårt att säga någonting om framtida brus ignoreras brustermer. Låt $f(t+j)$ beteckna de termer som är kända vid tidpunkten t .

$$\begin{aligned} f(t+1) &= [G_1(q^{-1}) - g_{10}] \Delta u(t) + F_1 y(t) \\ f(t+2) &= q[G_2(q^{-1}) - q^{-1}g_{21} - g_{20}] \Delta u(t) + F_2 y(t) \\ &\vdots \end{aligned}$$

där $G_i(q^{-1}) = g_{i0} + g_{i1}q^{-1} + \dots$

Ekvationerna ovan kan skrivas på vektorform

$$\hat{y} = G\tilde{u} + f \quad (18.6)$$

Där vektorerna är av längden $N \times 1$

$$\begin{aligned} \hat{y} &= \begin{pmatrix} \hat{y}(t+1) & \hat{y}(t+2) & \dots & \hat{y}(t+N) \end{pmatrix}^T \\ \tilde{u} &= \begin{pmatrix} \Delta u(t) & \Delta u(t+1) & \dots & \Delta u(t+N-1) \end{pmatrix}^T \\ f &= \begin{pmatrix} f(t+1) & f(t+2) & \dots & f(t+N) \end{pmatrix}^T \end{aligned}$$

Matrisen G i ekvationen 18.6 kommer att se ut som

$$G = \begin{pmatrix} g_0 & 0 & \dots & 0 \\ g_1 & g_0 & \dots & 0 \\ \vdots & \vdots & \dots & \cdot \\ g_{N-1} & g_{N-2} & \dots & g_0 \end{pmatrix}$$

Om det finns en dödtid $k > 1$ kommer de första raderna i matrisen G att vara 0. Väljs istället $N_1 = k$ kommer dessa rader att vara skilda från noll. Problemet är att man ofta inte känner dödtiden, emellertid är regulatorn stabil även då de första raderna är noll. Med en referenssignal,

$$u_c = \begin{pmatrix} u_c(t+1) & u_c(t+2) & \dots & u_c(t+N) \end{pmatrix}^T$$

kan förlustfunktionen (18.2) nu skrivas som

$$J = E \left\{ (y - u_c)^T (y - u_c) + \lambda \tilde{u}^T \tilde{u} \right\} \quad (18.7)$$

eller

$$J = \left\{ (G\tilde{u} + f - u_c)^T (G\tilde{u} + f - u_c) + \lambda \tilde{u}^T \tilde{u} \right\}$$

Minimeras denna funktion fås styrsignalvektorn

$$\tilde{u} = (G^T G + \lambda I)^{-1} G^T (u_c - f) \quad (18.8)$$

Notera att det första elementet i \tilde{u} är $\Delta u(t)$ vilket ger styrsignalen $u(t)$ som

$$u(t) = u(t-1) + \bar{g}^T (u_c - f) \quad (18.9)$$

där \bar{g}^T är den första raden av $(G^T G + \lambda I)^{-1} G^T$

Matrisen i ekvation 18.8 är av storleken $N \times N$. För att slippa invertera en så stor matris i varje sampelpunkt antas att efter tiden $NU < N_2$ är styrsignalinkrementet noll,

$$\Delta u(t + j - 1) = 0 \quad \text{om } j > NU \quad (18.10)$$

Värdet på NU kallas styrsignalhorisonten. Ekvation 18.6 blir i stället

$$\hat{y} = G_1 \tilde{u} + f$$

där G_1 är

$$G_1 = \begin{pmatrix} g_0 & 0 & \dots & 0 \\ g_1 & g_1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & g_0 \\ \vdots & \vdots & \dots & \vdots \\ g_{N-1} & g_{N-2} & \dots & g_{N-NU} \end{pmatrix}$$

Motsvarande styrsignalvektor blir

$$\tilde{u} = (G_1^T G_1 + \lambda I)^{-1} G_1^T (u_c - f) \quad (18.11)$$

18.3 Kommentar

Vi har försökt programmera en GPC-regulator i Matlab, men har inte fått den att fungera tillfredsställande. Av prioriteringsskäl har vi inte fördjupat oss i problemet.

Vi tror dock att GPC är en intressant algoritm för framtiden. Kanske skulle den kunna utgöra underlag för ett separat examensarbete.

19. Slutord

Vi har i detta examensarbete jämfört den adaptiva regulatorn Novatune med en konventionell PID-regulator. Genom ett stort antal simuleringar har vi skapat underlag för de slutsatser som redovisats i rapporten. Genomgående har PID-regulatorn dimensionerats enligt principen för dominerande polplacering och Novatune ställts in med hjälp av installationsanvisningarna. I flera fall är det möjligt att någon annan inställningsmetod gett ett annorlunda resultat. Detta är viktigt att ha i åtanke då man beaktar våra slutsatser.

I de flesta fall har det varit näst intill omöjligt att ge ett kategoriskt svar på vilken regulator som bäst lämpar sig för en given process. Generellt tycker vi att Novatune, om den är rätt inställd, är en utmärkt regulator för de flesta processer. För att åstadkomma optimal reglering med en PID-regulator fordras att man har en relativt god processkänedom. Emellertid har man i praktiken ofta bara tillgång till ett stegsvar för processen som ska styras. I dessa fall är Novatune överlägsen eftersom den genom adaptation anpassar sig till modellen.

Novatune, liksom andra adaptiva regulatorer, har nackdelen att den är krånglig att ta i drift. Totalt ska cirka 25 parametrar ställas in vid igångkörning.

Vi kan nämna att det vid institutionen för reglerteknik pågår forskning som syftar till att flera av parametrarna genom reläexperiment ska kunna ställas in automatiskt.

Med de erfarenheter vi nu har skaffat oss anser vi att det nog hade varit bättre att koncentrera sig på något färre processer och i gengäld behandla dessa grundligare. Vi hoppas trots allt att våra synpunkter ska kunna utgöra underlag för en handledning av valet mellan PID och Novatune.

Referenser

- ABB PROCESS AUTOMATION (1990): *ABB MasterPiece 200/1 Installation, igångkörning och service 7650 043-105*, ABB Process Automation, Västerås.
- DONALD, E. KNUTH (1984): *The T_EXbook*, Addison Wesley Publishing Company, Reading, Mass.
- GUSTAFSSON, K, M. LILJA, och M. LUNDH (1990): *A Collection of Matlab Routines for Control Systems Analysis and Synthesis*, Department of Automatic Control, Lund Institute of Technology.
- JOHANSSON, R. (1991): *Process Identifiering*, Institutionen för reglerteknik, Lunds Tekniska Högskola.
- THE MATH WORKS (1990): *Pro-Matlab User's Guide..*
- MODÉN, P. E. (1990): *Den adaptiva regulatorn Novatune, KP 7336/3*, ABB Process Automation, Västerås.
- MÖHLE M. (1990): *Jämförelse mellan en PID-regulator och Novatune*, Institutionen för Reglerteknik, Tekniska Högskolan i Lund.
- SSPA (1991): *Simmon, Users' Guide for UNIX Systems*, Institutionen för reglerteknik, Lunds Tekniska Högskola, Lund.
- ÅSTRÖM, K. J. och B. WITTENMARK (1989): *Adaptive Control*, Addison Wesley, Reading, Mass.
- ÅSTRÖM, K. J (1968): *Reglerteori*, Almqvist & Wiksells/Gebbers Förlag AB, Stockholm.
- ÅSTRÖM, K. J. (1970): *Intoduction to Stochastic Control Theory*, Academic press, inc., Berkeley Square House, London.
- ÅSTRÖM, K. J. och B. WITTENMARK (1990): *Computer Controlled Systems, 2nd ed*, Prentice Hall, Englewood Cliffs, New Jersey.

A. Programlistning

Här redovisas koden för de matlabrutiner som använts vid simulering och utvärdering. Eftersom rutinerna för Novatune-simulering och PID-simulering utför samma saker nöjer vi oss av utrymmesskäl med att lista koden för Novatune. En beskrivning av rutinerna finns i kapitel 7.

Lista A.1 simu.m

```
% Simu
% Macro file for simnon simulation with the adaptive controller
% Novatune.
% Before simulation initialize:
% Bp, Ap: continuous-time plant polynomials.
% tsamp, tau: sampling period and plant deadtime.
% reg, Kd, pn, int: Controller parameters. "reg" is the number of
% A- and B-parameters to be estimated in the controller.
% Variable "int" is set to 1 if controller with integral action.
% tinsv: The time to let the controller estimate the process
% before step response simulation.

tryuth12=adsim(Bp,Ap,tau,tsamp,reg,tinsv,Kd,int,pn);
ans = input('Step response simulation? (y/n) : ','s');
ans = [ans ' '];
if ans(1)~= 'y', disp(' Abort'); return; end
ans = input('th vector? (y/n) : ','s');
ans = [ans ' '];
if ans(1)~= 'y',
    th(1)=rsimval('th1[adap]');
    th(2)=rsimval('th2[adap]');
    th(3)=rsimval('th3[adap]');
    th(4)=rsimval('th4[adap]');
    th(5)=rsimval('th5[adap]');
    th(6)=rsimval('th6[adap]');
    th(7)=rsimval('th7[adap]');
    th(8)=rsimval('th8[adap]');
end;
rsimcom('par R[zzzcon]:1');
rsimcom('disp ada[reg]');
tl=tinsv+tmax*0.5+0.5*tsamp;
rsimcom('par ts: 0');
rsimcom(sprintf('par t1: %g',tl));
rsimcom('par b0[brus]:0');
rsimcom('par tada[adap]:1000');
disp(' Step response simulation')
rsimcom(sprintf('simu 0 %g -cont',tmax));
rsimcom('export zzztryu<zzztryu /100');
dum=rsimval('tau'); % for synchronization -- wait until
                    % simulation ready

tryu=yusimnonload;
tryu=corr(tryu);
tl=tl-tinsv;
tryu2=yustairs(tryu);
xpl(tryu2,tau,Kd,tsamp);
ans = input('iaes,ises..? (y/n) : ','s');
ans = [ans ' '];
if ans(1)~= 'y',
    iaes=rsimval('iaes')
    ises=rsimval('ises')
```

```

    ies=rsimval('ies')
    iael=rsimval('iael')
    isel=rsimval('isel')
    iel=rsimval('iel')
end;

end

```

Lista A.2 adsim

```

function tryu = adsim(bp,ap,tau,h,N,tmax,Kd,int,pn)
%ADSIM Simulation using remote simnon.
%
%   TRYU = ADSIM(BP,AP,TAU,H,N,TMAX,Kd,int,pn)
%   TRYU = ADSIM(BP,AP,TAU,H,N,TMAX,Kd,int)
%   TRYU = ADSIM(BP,AP,TAU,H,N,TMAX,Kd)
%   TRYU = ADSIM(BP,AP,TAU,H,N,TMAX)
%   TRYU = ADSIM(BP,AP,TAU,H,N)
%   TRYU = ADSIM(BP,AP,TAU,H)
%
%
% Ola och Rikard original from Michael Lundh
% Copyright (c) 1991 by Department of Automatic Control,
% Lund Institute of Technology, Lund, SWEDEN

if   nargin==4, N=11;      tmax=nan; Kd=1; int=1; pn=0;
elseif nargin==5, tmax=nan; Kd=1;      int=1; pn=0;
elseif nargin==6, Kd=1;      int=1;      pn=0;
elseif nargin==7, int=1;      pn=0;
elseif nargin==8, pn=0;
end;

test =[exist('zzzproc.t') ..
       exist('zzzcon.t')  exist('zzzstart.t') exist('zzzsim.t')]==2;
if any(test)
    ans = input('File zzz*.t exists. Continue and delete? (y/n) : ','s');
    ans = [ans ' '];
    if ans(1)~= 'y', disp(' Abort'); return; end
end
if test(1), delete zzzproc.t ; end
if test(2), delete zzzcon.t  ; end
if test(3), delete zzzstart.t; end
if test(4), delete zzzsim.t  ; end
disp(' ')

%---- Process
[a,b,c,d]=tf2ss(bp,ap);
nsimnon('zzzproc',0,a,[b 0*b],c,[d 1])
disp(' ')

%---- connecting system
disp(' Connecting system')
fil='zzzcon.t';
fprintf(fil,'connecting system zzzcon \n')
fprintf(fil,' time t \n\n')
fprintf(fil,' th1[reg] = th1[adap] \n')
fprintf(fil,' th2[reg] = th2[adap] \n')
fprintf(fil,' th3[reg] = th3[adap] \n')
fprintf(fil,' th4[reg] = th4[adap] \n')

```

```

fprintf(fil,' th5[reg] = th5[adap] \n')
fprintf(fil,' th6[reg] = th6[adap] \n')
fprintf(fil,' th7[reg] = th7[adap] \n')
fprintf(fil,' th8[reg] = th8[adap] \n')
fprintf(fil,' fi1[adap] = fi1[reg] \n')
fprintf(fil,' fi2[adap] = fi2[reg] \n')
fprintf(fil,' fi3[adap] = fi3[reg] \n')
fprintf(fil,' fi4[adap] = fi4[reg] \n')
fprintf(fil,' fi5[adap] = fi5[reg] \n')
fprintf(fil,' fi6[adap] = fi6[reg] \n')
fprintf(fil,' fi7[adap] = fi7[reg] \n')
fprintf(fil,' fi8[adap] = fi8[reg] \n')
fprintf(fil,' fbt[adap] = fbt[reg] \n')
fprintf(fil,' ada[adap] = ada[reg] \n')
fprintf(fil,' u
= u[reg] \n')
fprintf(fil,' utemp
= delay(u + 1 , tau) \n')
fprintf(fil,' u1[zzzproc]= utemp+e1[brus] \n')
fprintf(fil,' u2[zzzproc]= 0 \n')
fprintf(fil,' ref[reg] = ref[input] \n')
fprintf(fil,' y[reg] = y1[zzzproc]+e2[brus] \n')
fprintf(fil,' tmax[input]= tmax \n')
fprintf(fil,' R[input] = R \n')
fprintf(fil,' r[integral]= ref[input] \n')
fprintf(fil,' y1[integral]= y1[zzzproc] \n')
fprintf(fil,' l[integral] = 1 \n')
fprintf(fil,' reftyp[integral] = R \n\n')
fprintf(fil,' l = if t<t1 then 0 else 10 \n\n')
fprintf(fil,' tau: %g \n',tau)
fprintf(fil,' t1 : 1000 \n')
fprintf(fil,' ts : 1000 \n')
fprintf(fil,' l0 : -1 \n')
fprintf(fil,' R : 0 \n')
fprintf(fil,' tmax: 100 \n')
fprintf(fil,'end \n')

%---- start macro
disp(' Start macro')
fil='zzzstart.t';
fprintf(fil,'macro zzzstart \n')
fprintf(fil,' syst zzzproc logger brus input')
if N==1, fprintf(fil,' star01 adapt1 ');
elseif N==11, fprintf(fil,' star11 adapt2 ');
elseif N==12, fprintf(fil,' star12 adapt3 ');
elseif N==13, fprintf(fil,' star13 adapt4 ');
elseif N==22, fprintf(fil,' star22 adapt4 ');
elseif N==23, fprintf(fil,' star23 adapt5 ');
elseif N==24, fprintf(fil,' star24 adapt6 ');
elseif N==33, fprintf(fil,' star33 adapt6 ');
elseif N==44, fprintf(fil,' star44 adapt8 ');
else disp(' Undefined controller'); return;
end;
fprintf(fil,'integral zzzcon \n')
fprintf(fil,' store ref[input] y1[zzzproc] u[reg] th1[adap]')
if N==1, fprintf(fil,' \n ');
elseif N==11, fprintf(fil,' th2[adap] \n ');
elseif N==12, fprintf(fil,' th3[adap] \n');
elseif N==13, fprintf(fil,' th4[adap] \n');
elseif N==22, fprintf(fil,' th3[adap] \n');
elseif N==23, fprintf(fil,' th4[adap] \n');
elseif N==24, fprintf(fil,' th5[adap] \n');
elseif N==33, fprintf(fil,' th4[adap] \n');
elseif N==44, fprintf(fil,' th5[adap] \n');
else disp(' Undefined controller'); return;

```

```

end;
ts=0.1;
fprintf(fil,sprintf(' par dt[logger] : %g \n',ts))
fprintf(fil,'end \n')

%---- simulation macro
disp(' Simulation macro')
fil='zzsim.t';
fprintf(fil,'macro zzzsim ; tid ; t10 ; fil \n')
fprintf(fil,' default tid = 30 \n')
fprintf(fil,' default t10 = tid * 0.5 \n')
fprintf(fil,' default fil = zzztryu \n\n')
fprintf(fil,' par tada : %g \n', h)
fprintf(fil,' par treg : %g \n', h)
fprintf(fil,' par INTv : %g \n',int)
fprintf(fil,' par PN[reg]: %g \n',pn)
fprintf(fil,' par Kd : %g \n', Kd)
fprintf(fil,' par step : 1 \n')
fprintf(fil,' par tmax : %g \n',tmax)
fprintf(fil,' par t1[zzzcon]: t10 \n')
fprintf(fil,' simu 0 tid / fil \n')
fprintf(fil,' export fil<fil /100 \n')
fprintf(fil,' split 2 2 \n')
fprintf(fil,' area 1 1 \n')
fprintf(fil,' ashow ref y1 \n')
fprintf(fil,' ashow u \n')
fprintf(fil,' ashow th1 \n')
if N==1, fprintf(fil,' \n ');
elseif N==11, fprintf(fil,' ashow th2[adap] \n ');
elseif N==12, fprintf(fil,' ashow th3[adap] \n ');
elseif N==13, fprintf(fil,' ashow th4[adap] \n ');
elseif N==22, fprintf(fil,' ashow th3[adap] \n ');
elseif N==23, fprintf(fil,' ashow th4[adap] \n ');
elseif N==24, fprintf(fil,' ashow th5[adap] \n ');
elseif N==33, fprintf(fil,' ashow th4[adap] \n ');
elseif N==44, fprintf(fil,' ashow th5[adap] \n ');
end;
fprintf(fil,'end \n')

disp(' ')
if isnan(tmax)
    disp(' No simulation done -- Do this by hand')
    tryu = nan;
else
    disp(' Simmon Simulation')

    rsimcom('zzzstart')
    rsimcom(sprintf('zzsim %g 1000',tmax))

    dum=rsimval('tau'); % for synchronization -- wait until
                        % simulation ready
    tryu = yusimmonload;
end

```

Lista A.3 mtr

```

function [M,Tr,Ts,TAmx]=mtr(tryu,start,stop)
%MTR
% [M,Tr,Ts,TAmx]=mtr(tryu,start,stop)
% [M,Tr,Ts,TAmx]=mtr(tryu,start)
% [M,Tr,Ts,TAmx]=mtr(tryu)
%
% Calculates overshoot M, rise time Tr,
% solution time Ts and the time for the

```

```

%         maximum deviation from set-point after
%         a load disturbance.
%         The matrix tryu should contain four columns:
%         [time reference output control]
%

if      nargin==3, t=min(find(tryu(:,1)>=start));
                    s=max(find(tryu(:,1)<=stop));
elseif nargin==2, t=min(find(tryu(:,1)>=start));
                    s=length(tryu(:,1));
elseif nargin==1, t=1; s=length(tryu(:,1));
end

ta=tryu(:,1);
tb=ta(t:s);
ra=tryu(:,2);
rb=ra(t:s);
ya=tryu(:,3);
yb=ya(t:s);
ua=tryu(:,4);
ub=ua(t:s);
tryu=[tb rb yb ub];
if tb(1)==0,
    M=max(tryu(:,3)-tryu(:,2));
    TAmx=nan;
else
    M=max(abs(tryu(:,3)-tryu(:,2)));
    for i=1:size(tryu(:,1)),
        if M==max(abs(tryu(i,3)-tryu(i,2))),
            TAmx=tryu(i,1)-start;
        end %if
    end %for
end
rmin=tryu(1,2)*0.1;
rmax=tryu(1,2)*0.9;
i=1;
while (tryu(i,3)<rmin),
    i=i+1;
    tmin=tryu(i,1);
end
while (tryu(i,3)<rmax),
    i=i+1;
    tmax=tryu(i,1);
end
Tr=tmax-tmin;
p=0.05*tryu(1,2);
for k=1:size(tb),
    if abs(tryu(k,2)-tryu(k,3))>p,
        Ts=tryu(k,1)-start;
    end
end
end

```

Lista A.4 c2rst

```

function [R,S,T]=c2rst(k,ti,td,tsamp,beta);
%c2rst
%
%         [R,S,T]=c2rst(k,ti,td,tsamp,beta)
%         [R,S,T]=c2rst(k,ti,td,tsamp)
%
%         Converts continuous-time PID-parameters
%         into discrete-time polynomials.

```

```

%
if nargin==4, beta=1;
end;

if td/tsamp<=5,
    alfa=0
else
    alfa=(td/tsamp-5)/(td/tsamp+5)
end;

s0=k*(1+tsamp/ti+td/tsamp*(1-alfa));
s1=-k*(1+alfa*(1+tsamp/ti)+2*td/tsamp*(1-alfa));
s2=k*(alfa+td/tsamp*(1-alfa));

t0=beta*ti+tsamp;
t1=-(beta*ti*(1+alfa)+tsamp*alfa);
t2=alfa*beta*ti;

R=[1 -(1+alfa) alfa];

S=[s0 s1 s2];

T=k/ti*[t0 t1 t2];

end

```

Lista A.5 th2rst

```

function [R,S,T]=th2rst(th,NA,NB,PN,intv);
%
%       [R,S,T]=th2rst(th,NA,NB,PN,intv)
%       Calculates the controller polynomials
%       R,S and T from the estimated th.
%       th is a rowvector with th1,...,th8.
%

a0=0; a1=0; a2=0; a3=0; b0=0; b1=0; b2=0; b3=0;
raa0=4*0.01^2/20^2;
raa=raa0*2^PN

if NB>=1,
b0=th(1);
end;
if NB>=2,
b1=th(2);
end;
if NB>=3,
b2=th(3);
end;
if NB>=4,
b3=th(4);
end;

if NA>=1,
a0=th(NB+1);
end;
if NA>=2,
a1=th(NB+2);
end;
if NA>=3,
a2=th(NB+3);
end;
if NA>=4,

```

```

a3=th(NB+4);
end;
% intv
x=0.75;          % x=1-PL
k=b0/(b0^2+raa);
k1=raa/(b0^2+raa);

if intv>0.5,
    Rt(1)=1; Rt(2)=k*b1-1; Rt(3)=k*(b2-b1);
    Rt(4)=k*(b3-b2); Rt(5)=-k*b3;
    % Rt
    for i=1:NB+1,
        R(i)=Rt(i);
    end;

    St(1)=k*(x+a0); St(2)=k*(a1-a0); St(3)=k*(a2-a1);
    St(4)=k*(a3-a2); St(5)=-k*a3;
    for i=1:NA+1,
        S(i)=St(i);
    end

    T=k*x;

elseif intv<0.5,
    Rt(1)=1; Rt(2)=-k1-k*b1; Rt(3)=k*b2; Rt(4)=k*b3;
    for i=1:NB,
        R(i)=Rt(i);
    end;

    St(1)=k*a0; St(2)=k*a1; St(3)=k*a2; St(4)=k*a3;
    for i=1:NA,
        S(i)=St(i);
    end

    T=k*x;

else
    disp('The variable intv should be 0 or 1')
end %if
end;

```

Lista A.6 polpd

```

function [R,S,T]=polpd(Bplus,Bminus,A,tau,tsamp,
                    alpha,deg0,omega,z)
%
%polpd
%
%[R,S,T]=polpd(Bplus,Bminus,A,tau,tsamp,alpha,
%              deg0,omega,z)
%      Blus,Bminus and A given in discrete time
%      R,S and T in discrete time
%
%      [R,S,T]=polpd(B,A,tau,tsamp)
%      B,A given in continuous time
%      R,S and T in discrete time
%

if nargin==4, Bc=Bplus; Ac=Bminus;
    tauc=A; tsamp=tau;
    [bd,ad]=sample(Bc,Ac,tauc,tsamp);
    Bplus=1;
    Bd=bd;
    if any(roots(bd)),
        rbd=roots(bd);
        v=find(abs(rbd)<0.8); %Any roots inside abs(q)<0.8
        for k=1:length(v),
            [bd,remainder]=deconv(bd,[1 -rbd(v(k))]);
            if remainder*remainder'>=eps,
                disp(sprintf('Remainder bd = %g',remainder));
            end;
            Bplus=conv(Bplus,[1 -rbd(v(k))])
        end;
    end;
    if any(roots(Bplus)<-0.1),
        disp('Warning Bplus may be not good')
    end;
    Bminus=bd;
    A=ad;
    alpha=-2; omega=1; z=0.7;
    Amc=[1 2*z*omega omega^2]; % desired system
    Bmc=omega^2;
    [Bm,Am]=sample(Bmc,Amc,tauc,tsamp);
    deg0=2*(length(ad)-1)-(length(Bplus)-1)
        -(length(Am)-1);
else
    Amc=[1 2*z*omega omega^2]; % desired system
    Bmc=omega^2;
    [Bm,Am]=sample(Bmc,Amc,tau,tsamp);
end;

if deg0~=0,
    Aot=[1 -exp(alpha*tsamp)];
    Ao=Aot;
    for i=1:(deg0-1),
        Ao=conv(Ao,Aot);
    end;
else
    Ao=1;
end;

```



```

[Bmp,remainder]=deconv(Bm,Bminus);
if remainder*remainder'>=eps,
    disp(sprintf('Remainder Bmp = %g',remainder));
end;

```

```

[R1,S]=dab(conv([1 -1],A),Bminus,conv(Am,Ao));
Rp=conv([1 -1],R1);
R=conv(Bplus,Rp);
T=sum(Am)/sum(Bm)*conv(Ao,Bmp);
S=S/R(1);
T=T/R(1);
R=R/R(1);
end;

```

Lista A.7 noisev

```

function [meany,vary,varys,vardu]=noisev(tryu,tsamp);
%
% [meany,vary,varys,vardu]=noisev(tryu,tsamp);
%
% Calculates E{y}, V{y} and
% V{y}, V{du} in the sample points.
% Noisev uses noisevec, which creates
% the vectors [ysamp y du].
% ysamp is the output in the sample-points.
% du is the control signal increments.
%

[ysamp,y,du]=noisevec(tryu,tsamp);
meany=mean(y);
vary=(std(y))^2;
varys=(std(ysamp))^2;
vardu=(std(du))^2;
end;

```

Lista A.8 noise

DISCRETE SYSTEM noise

```

output e1 e2
state x
new nx
time t
tsamp ts

```

```

nx=a*x+b*norm(t)
e1=x

```

```

e2=b0*norm(t)

```

```

ts=t+dt

```

```

a : 0 "0.995 "filter constant
b : 0 "0.0998 "filter constant
b0 : 0 "0.01 "noise standard deviation
dt : 0.1

```

```

end

```

