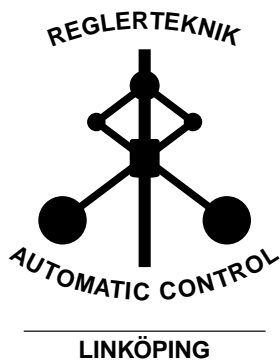
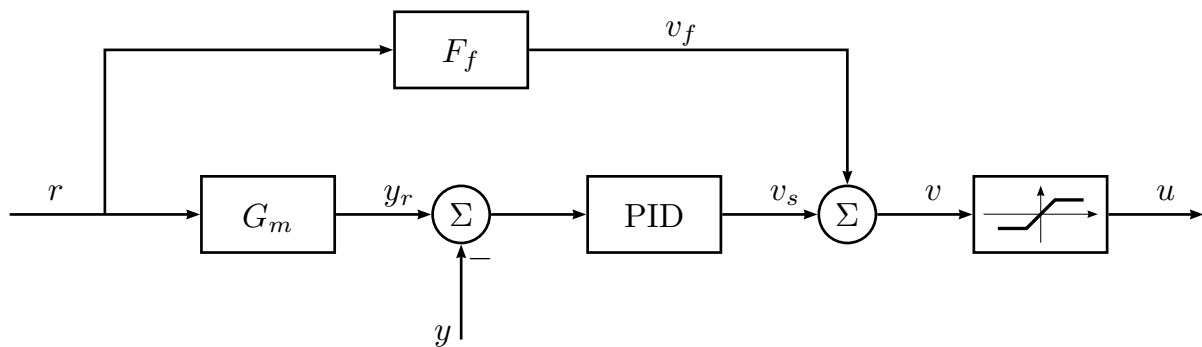


Industriell reglerteknik — Lab 2

Inställning och implementering av PID-regulatorer med framkoppling

Denna version: 6 februari 2020



Namn: _____

Personnr: _____

Datum: _____

Godkänd: _____

1 Inledning

PID-regulatorn är den överlägset vanligaste regulatorn i industriella tillämpningar. Grundprincipen för en PID-regulator är relativt enkel och lärs ut i alla grundkurser i reglerteknik men det finns många sätt att modifiera den. Vidare finns det mycket att säga om inställning av PID-regulatorns parametrar samt om hur man på bästa sätt implementerar en PID-regulator i programkod.

Denna laboration handlar om tidsdiskret implementering och inställning av PID-regulatorer och hur man kan kombinera en sådan med en framkoppling från referenssignalen. Den resulterande regulatorn ska utvärderas på *en* av följande laborationsprocesser:

- Vattentank (system 2a)
- Likströmsmotor (system 2b)
- Vindflöjel (system 2c)

Laborationens syfte

Syftet med laborationen är att ni ska kunna

- implementera en PID-regulator i programkod
- införa hantering av integratoruppvridning i regulatorkoden
- skriva regulatorkoden så att man får stötfria övergångar vid mod- och parameterbyten
- ställa in regulatorparametrarna så att ett verkligt system kan regleras väl
- komplettera PID-regulatorn med en framkoppling från referenssignalen
- välja framkopplingen så att referensföljningen förbättras

2 Laborationens upplägg

Laborationen genomförs i grupper om två studenter. Det praktiska upplägget är följande:

1. **Förberedelser:** Läs igenom detta laborationshäfte och avsnitten om framkoppling från referenssignalen samt inställning och implementering av PID-regulatorer i kurskompendiet. Lös lektionsuppgifterna som handlar om PID-implementering.
2. **Introduktionstillfälle (2h):** Det obligatoriska introduktionstillfallet genomförs i en av ISY:s datorsalar och tanken är att ni där ska kunna komma igång med laborationsuppgifterna och kunna diskutera eventuella oklarheter med en assistent.
3. **Eget arbete:** Laborationsuppgifterna ska lösas på egen hand av varje laborationsgrupp vid tillfällena som ni själva bokar in under en vecka efter introduktionstillfallet. Många av uppgifterna går att lösa bara med tillgång till en dator med Matlab, men för att göra den slutliga inställningen av regulatorparametrarna och testa er regulator på den verkliga laborationsprocessen behöver ni boka tid i Laboteket. Bokningspärmar kommer att finnas vid arbetsplatserna i RT3. Skriv namn och mejladress på listorna. Eftersom det kommer att vara flera grupper som arbetar parallellt med samma laborationsprocess är det bara tillåtet att boka **max 2 timmar** per laborationsgrupp åt gången så att alla ska få en likvärdig chans att använda arbetsplatserna. Detta gäller även helgtiderna, men så snart en grupp har avverkat sina bokade timmar kan de förstås boka två nya timmar o.s.v.
4. **Resurstillfällen:** Under tiden som det egna arbetet pågår kommer det att finnas resurstillfällen i Laboteket kl. 12.30 de dagar som anges på kurshemsidan. Vid dessa tillfällen kommer en assistent att finnas tillgänglig för att svara på frågor.
5. **Examinationstillfälle (1h):** Det obligatoriska examinationstillfallet genomförs i RT3 i Laboteket och äger rum cirka en vecka efter introduktionstillfallet. Fyra laborationsgrupper kommer att examineras vid varje tillfälle och dessa grupper kommer att ha arbetat med varsin laborationsprocess. Examinationen går till så att ni berättar hur ni har löst laborationsuppgifterna och vad som var speciellt med just er laborationsprocess samt demonstrerar er regulator för de andra laborationsgrupperna och för assistenten. Tanken är alltså att ni ska hålla en kort muntlig genomgång på cirka fem minuter samt därefter kunna svara på frågor under ytterligare fem minuter. **Observera att alla laborationsuppgifter måste vara helt lösta när ni kommer till redovisningstillfallet eftersom det inte finns någon tid över där!**

Att tänka på

Eftersom arbetet med laborationsuppgifterna till största delen kommer att ske på egen hand finns det några saker som ni bör tänka på:

- Det är inte tillåtet att äta mat i Laboteket eller att ta med några drycker till laborationsplatserna. Det främsta motivet för denna regel är er egen säkerhet (laborationsprocesserna kan kortslutas av utspillda vätskor) men också för att öka livslängden på utrustningen och att se till att laborationsplatserna är fräscha.
- Visa hänsyn till era medstudenter. Följ regeln som nämndes ovan om hur mycket tid man får boka. Tänk också på att det kan pågå annan undervisning i Laboteket samtidigt som ni är där för att arbeta med laborationsuppgifterna och se till att laborationsplatsen är ren och städad när ni lämnar den.

3 Kravspecifikation för examination

Nedan följer de krav som måste vara uppfyllda för att ni ska bli godkända på laborationen. Dessutom krävs närvaro vid redovisningstillfället. Om ni vill kan ni examineras på krav 1 och 2 innan redovisningstillfället för att ge er mer tid för redovisningen av krav 3, 4 och 5 där. En variant kan vara att sikta på att bli klar med dessa krav redan på introduktionstillfället.

1. Ni ska ha implementerat en PID-regulator som hanterar integratoruppvridning i matlabkod. Vid redovisningen ska ni visa plottar som visar ut- och styrsignaler med och utan hantering av integratoruppvridning när er regulator används i simulinkmodellen `simmodel.mdl` med systemet och parametrarna i `benchmark.m`. Vidare ska ni kunna förklara hur ni har löst problemet med integratoruppvridning.
2. Ni ska ha implementerat en PID-regulator som har stötfria övergångar mellan manuell och automatisk mod i matlabkod. Vid redovisningen ska ni visa plottar som visar ut- och styrsignaler med och utan stötfria övergångar när er regulator används i simulinkmodellen `simmodel.mdl` med systemet och parametrarna i `benchmark.m`. Vidare ska ni kunna förklara hur ni har sett till att övergångarna blir stötfria.
3. Ni ska redogöra för vad som karakteriserar er laborationsprocess (system 2a, 2b eller 2c) och vad det har för implikationer för inställning

av PID-parametrar och framkopplingslänkar. (Vilka inställningsmetoder kan användas för er typ av process?, Kan man välja godtyckliga framkopplingar från referenssignalen?, etc.)

4. Ni ska ha implementerat en PID-regulator i matlabkod och ställt in dess parametrar så att regleringen av er laborationsprocess fungerar väl både med avseende på referensföljning och störningsundertryckning. Vid redovisningen ska ni förklara hur ni har gjort regulatorinställningen samt demonstrera hur väl er regulator fungerar på det verkliga systemet. Denna regulator ska även ha stötfria övergångar vid parameterbyten och ni ska demonstrera att så är fallet. Dessutom ska regulatorn hantera integratoruppvridding men det behöver inte demonstreras. Arbetet med att hitta en lämplig inställning kan med fördel göras på ett simulerat system men den slutliga inställningen ska vara för det verkliga systemet.
5. Ni ska ha implementerat en PID-regulator med framkoppling från referenssignalen i matlabkod och ställt in både regulatorparametrarna och framkopplingen så att regleringen av er laborationsprocess fungerar väl både med avseende på referensföljning och störningsundertryckning. Vid redovisningen ska ni förklara hur ni har gjort samt demonstrera hur väl er regulator fungerar på det verkliga systemet när det kommer ett steg i referenssignalen. Ni ska också kunna redogöra för om framkopplingen var användbar i det aktuella fallet. Regulatorn ska även hantera integratoruppvridding men det behöver inte demonstreras.

OBS: Alla laborationsuppgifter ska lösas på egen hand av varje laborationsgrupp och det är inte tillåtet att kopiera andra gruppers lösningar. Däremot är det tillåtet att *diskutera* lösningarna med andra laborationsgrupper. Vid redovisningen måste båda medlemmarna av laborationsgruppen kunna förklara alla delar av er lösning.

Tips

- Det är *inte* något krav att implementera en generell PID-regulator som hanterar alla krav i kravspecifikationen utan det går att göra olika implementeringar som löser de olika deluppgifterna.
- Det kan vara lämpligt att först boka ett tvåtimmarspass i Laboteket i början av laborationsveckan så att ni kan testa laborationsutrustningen i lugn och ro och få lite känsla för hur er laborationsprocess fungerar. Efter det kan ni troligen lösa de flesta laborationsuppgifterna vid någon dator med Matlab för att mot slutet av laborationsveckan åter boka ett tvåtimmarspass i Laboteket där ni kan verifiera era lösningar,

justera parameterinställningarna samt förbereda demonstrationen vid redovisningen.

- Vissa kombinationer av inställningsmetoder och system är mer naturliga än andra (och en del fungerar inte alls). Testa ett par olika som verkar lämpliga för just ert system.
- God referensföljning betyder inte nödvändigtvis snabb referensföljning. I många tillämpningar vill man få det slutna systemet långsammare än det öppna, och ni kan själva välja att prioritera just detta. Här kan framkoppling vara intressant eftersom det möjliggör olika referensföljningsegenskaper utan att man ändrar på återkopplingsregulatorns hantering av störningar.

4 Processbeskrivningar

4.1 Benchmarksystemet (system 1)

I laborationen används ett simulerat benchmarksystem som är samma för alla grupper för att utvärdera hur er regulator hanterar integratoruppvridning och stötfria övergångar vid modomkopplingar. Benchmarksystemet är

$$G(s) = \frac{2}{s^3 + 7s^2 + 16s + 10}$$

och har även studerats i t.ex. lektionsuppgift 5.3. PID-parametrarna som ska användas för detta system är $K = 17.85$, $T_i = 1.1932$ och $T_d = 0.298$. Dessa finns redan definierade i filen `benchmark.m`. Genom att köra denna fil utförs ett benchmarkexperiment (se krav 1 och 2). I detta experiment har signalen `amn` värdet 1 om regulatorn ska vara i automatisk mod och 0 om den ska vara i manuell mod.

4.2 Vattentank (system 2a)

Detta system består av en vattentank och en pump. Målet är att styra spänningen in till pumpen så att den pumpar in vatten i tanken till en viss referensnivå och håller nivån konstant. Detta är ett olinjärt system och den modell som ni får given är en stabil linjär modell som är framtagen genom att systemet har linjäriserats runt nivåerna $u_0 = 4$ och $y_0 = 5.5$. Insignalen till systemet får ligga i intervallet $[0, 10]$ V. Systemet är

$$G(s) = \frac{3.47}{26.8s + 1}$$

Modellen är framtagen för det fall då hålet vattnet strömmar ut ur är 3 mm.

Uppkoppling

På IO-kortet fås styrsignalen u från utgången som heter A00. Denna skall kopplas till kontakten på vattentanken som heter STYRSIGNAL, dvs. den översta kontakten på vattentanken. Ingången AI1 på IO-kortet ger mätsignalen y . Denna skall kopplas till kontakten som heter NIVÅ. Både styrsignalen och mätsignalen skall jordas genom att den svarta kontakten i utgången på koaxialkabeln kopplas in på ingången som är märkt med jordsymbolen, dvs den nedersta kontakten på vattentanken. Dessutom måste man se till att processen är strömsatt.

Om man vill koppla in en extern referenssignal kan man göra detta genom att koppla in en styrspak på AI0 på IO-kortet.

IO-kortet som används är det som är märkt Connector 0.

4.3 Likströmsmotor (system 2b)

Detta system består av en elmotor. Målet är att styra spänningen till elmotorn så att man får ut rätt vinkel på vinkelskivan på motorn. Detta system är ett linjärt system med integrator och $u_0 = 0$ och $y_0 = 0$. Insignalen till systemet får ligga i intervallet $[-10, 10]$ V. Systemet är

$$G(s) = \frac{39.1}{s(0.277s + 2)}$$

Uppkoppling

På IO-kortet fås styrsignalen u från utgången som heter A00. Denna skall kopplas till **Pre-amplifier Unit** där jord (svart kontakt) skall kopplas till 0 V och den andra kontakten skall kopplas till **Signal 1**. Ingången AI1 på IO-kortet ger mätsignalen y . Denna skall kopplas till **Output Potentiometer**, där jord (svart kontakt) skall kopplas till ingången märkt 2 och den andra kontakten skall kopplas till ingången märkt 3. Dessutom måste man se till att processen är strömsatt.

Om man vill koppla in en extern referenssignal kan man göra detta genom att koppla AI0 till **Input Potentiometer** på motsvarande sätt som mätsignalen kopplades in.

IO-kortet som används är det som är märkt Connector 0.

4.4 Vindflöjel (system 2c)

Detta system består av en fläkt och en vindflöjel. Målet är att styra spänningen till fläkten så att vindflöjeln får en given vinkel. Detta är ett olinjärt system och den modell som ni får given är en stabil linjär modell som är framtagen genom att systemet har linjäriserats runt nivån $u_0 = 5$. Då det finns två vindflöjlar och dessa har olika nollnivå på sensorerna, måste ett experiment utföras för att ta reda på korrekt y_0 för den vindflöjeln som ni använder. Detta görs enklast genom att modifiera modellen i `xpcpid.mdl` så att en spänning på 5V skickas ut och sedan kolla vilken nivå y hamnar på. Insignalen till systemet får ligga i intervallet $[0, 10]$ V. Systemet är

$$G(s) = \frac{2.434}{4.69 \cdot 10^{-3}s^3 + 1.81 \cdot 10^{-2}s^2 + 4.61 \cdot 10^{-1}s + 1}$$

När modellen togs fram satt den justerbara vikten på flöjeln längst in mot axeln.

Uppkoppling

På IO-kortet fås styrsignalen u från utgången som heter A00. Denna skall kopplas till U STYR, där den svarta kontakten skall kopplas till - och den andra kontakten till +. Ingången AI1 på IO-kortet ger mätsignalen y . Denna skall kopplas till U MÅT, genom att den svarta kontakten kopplas till - och den andra kontakten till +. Dessutom måste man se till att processen är strömsatt.

Om man vill koppla in en extern referenssignal kan man göra detta genom att koppla in en styrspak på AI0 på IO-kortet.

IO-kortet som används är det som är märkt Connector 0.

5 Utvecklingsverktyg

5.1 Matlab och Simulink

De främsta verktygen vid lösningen av laborationsuppgifterna är Matlab och Simulink, som ni är bekanta med från lektionerna i kursen. För att regulatorerna ska fungera i Laboteket (med xPC Target) måste ni implementera dem i blocket `Embedded MATLAB Function` i Simulink. Detta är en skillnad jämfört med lektionerna där ni gör det i en m-fil. Blocket `Embedded MATLAB Function` finns redan inlagt med ett programskelett för en PID-regulator

i simulinkfilen `simmodel.mdl`. Regulatorkoden har samma syntax som på lektionerna.

Filen `simmodel.mdl` kan användas för att testa er regulator på ett simulerat system. För att ställa in vilket system ni har sätter ni variabeln `labprocess` till `elmotor`, `vattentank` eller `vindflojel` i filen `setpidparam.m` och kör filen. Sedan laddar ni in rätt process genom att köra filen `initsimparam.m`. Filen `simmodel.mdl` används även när ni ska testa er regulator på benchmarksystemet.

Innan ni börjar

När ni kommer till en av ISY:s datorsalar behöver ni kopiera laborationsfilerna till ert konto. Laborationsfilerna finns i katalogen `pidtuneimp` som ni i Linux kan kopiera med kommandot

```
cp -r /courses/tsrt07/labs/pidtuneimp .    (Observera punkten!)
```

när ni står i er hemkatalog. I Windows kommer ni åt dem via sökvägen `K:\TSRT07\labs\`.

5.2 xPC Target

Verktyget xPC Target är framtaget av MathWorks och kan användas för att snabbt och enkelt kunna göra om simulinkscheman till programkod som kan exekveras i realtid, till exempel för att reglera ett fysiskt system. När man arbetar med xPC Target designar man först sin regulator i Simulink på en värddator (eng. *host PC*) och därefter automatgenererar man kod från detta simulinkschema med hjälp av Real-Time Workshop. Koden kan därefter laddas ner till en måldator (eng. *target PC*) och där exekveras i realtid.

Innan ni börjar

När ni kommer till Laboteket behöver ni göra följande saker för att kunna använda xPC Target.

- Starta måldatorn (den lilla gråsvarta datorn som står på bordet).
- Logga in på värddatorn med ditt vanliga användarnamn, gå till katalogen `K:\TSRT07\labs\` och kopiera därefter katalogen `pidtuneimp` till ert konto (om ni inte har gjort detta tidigare).

- Starta Matlab 2016b och gå till katalogen med laborationsfilerna.
- Skriv `xpcsetup` i matlabfönstret.
- Skriv `xpcexplr` för att starta xPC Target Explorer, högerklicka på `TargetPC1` i fönstret som kommer upp och välj `Connect`. Lämna sedan fönstret öppet.

Problem med anslutning till xPC Target

Det finns ett känt fel som ibland dyker upp när man försöker ansluta till xPC, med följande felmeddelande: `TargetPC1: The IP address value specified is already used by another environment.`

När det inträffar så har det på något vis skapats två xPC Target-instanser, och det enklaste sättet att lösa problemet är att öppna xPC Explorer enl. ovan och

- Högerklicka på `TargetPC2` och välj `connect`
- Högerklicka på `TargetPC1` och välj `delete`
- Högerklicka på `TargetPC2` och välj `disconnect`
- Högerklicka på `TargetPC2` och välj `rename`
- Byt namn till `TargetPC1`.
- Högerklicka på `TargetPC1` och välj `connect`

Att arbeta med xPC Target

Den typiska arbetsgången när man arbetar med xPC Target i denna laboration är följande.

- Använd simulinkfilen `xcpid.mdl` och modifiera den vid behov. Denna fil innehåller de speciella in- och utblock som sköter kommunikationen med måldatorns IO-kort samt utportar för de signaler som man kan vara intresserad av att kunna studera efteråt och scopeblock för de signaler som man kan vilja följa i realtid. I första hand kommer ni att behöva modifiera matlabkoden som finns i blocket `Embedded MATLAB Function`.

- Skriv `setpidparam` i matlabfönstret för att definiera regulatorparametrarna och samplingstiden samt vilket system ni arbetar med. (Denna fil kommer ni förstås att behöva ändra i.)
- Skriv `initsimparam` i matlabfönstret för att läsa in några systemspecifika parametrar.
- Kompilera er modell med kommandot `slbuild('xpcpid')`.
- Ladda ner programmet med kommandot `tg.load('xpcpid')`
- Programmet kan startas på måldatorn med kommandot `tg.start` och stoppas med `tg.stop`.
- Under körning kan man titta på styr-, referens- och utsignalerna genom att välja `TargetPC1 - xpcpid - xPC Scopes` i `xPC Target Explorer` och högerklicka på `Scope: 1` och välja `View Scope`. Alternativt kan man byta skärmbild och titta på signalplottarna direkt på måldatorn.
- Om man vill ändra ett parametervärde under körningen kan man göra det genom att ändra värdet i filen `setpidparam.m` och köra filen igen. Ett alternativ är kommandot `tg.setparam(parameterno,value)`. En lista över parametrarna kan fås med `tg.showParameters='on'` och det aktuella värdet på en parameter med `tg.getparam(parameterno)`. Följande exempel visar hur man kan ändra regulatorförstärkningen (första elementet i `ctrlparam`-vektorn) i `xpcpid.mdl` under en körning:


```
tg.getparam(3)
tg.setparam(3, [0.1;10;0.1;0.1;10])
```
- Efter en körning finns data tillgängligt i objektet `tg.outputLog`. Skriv `plotpiddata` i matlabfönstret om ni vill se en plott av styr-, referens- och utsignalerna.

6 Laborationsfiler

Nedanstående filer behövs för att genomföra laborationen.

- xPC-filer:
 - `xpcsetup.m` – Används för att göra de nödvändiga inställningarna i `xPC Target`.
 - `xpcpid.mdl` – En simulinkfil som kan användas för att testa PID-reglering med `xPC Target`.

- `plotpiddata.m` – Används för att plotta styr-, referens- och ut-signalerna efter en xpcörning med `xpcpid.mdl`.
- Simuleringsfiler:
 - `simmodel.mdl` – En simulinkfil som kan användas för att testa PID-reglering med simulerad modell.
 - `benchmark.m` – Körs för att utvärdera en PID-regulator på ett givet system med en given signal.
 - `benchmarksignals.mat` – I den här filen finns signalerna som används för benchmarktestet.
 - `paramel.mat` – I den här filen finns parametervärden för elmotorn.
 - `paramvind.mat` – I den här filen finns parametervärden för vindflöjeln.
 - `paramvatten.mat` – I den här filen finns parametervärden för vattentanken.
- Generella filer:
 - `setpidparam.m` – Används för att definiera regulatorparametrarna och samplingstiden i `xpcpid.mdl` och `simmodel.mdl`. Används även för att ändra parametrar under körning.
 - `changeparam.m` – Används av `setpidparam.m` för att ändra parametrar under körning.
 - `initsimparam.m` – Körs för att definiera modeller och parametervärden för `simmodel.mdl` och `xpcpid.mdl`.

I första hand behöver ni bara göra ändringar i de tre filerna `setpidparam.m`, `xpcpid.mdl` och `simmodel.mdl`. I `setpidparam.m` anger ni era parametervärden samt vilket system ni har och i de två simulinkmodellerna ska ni lägga in koden för er regulator.