

Projektmedlemmar

Soheil Afghani
Mehmet Cicek
Jaqub Ghairat
Hassan Mouhsen

Handledare

Gabriel Ingesson

General State Feedback Controller Using MATLAB Computer Engine

Introduktion

Projektet vi har valt att utveckla, General State Feedback Controller Using MATLAB Computer Engine, går ut på att utveckla en generell state feedback controller med en observer som skall fungera med nästan alla processer som finns tillgängliga i labbsalen, bland annat vattentanksprocessen och DC servo. Som titeln på projektet säger skall vi använda MATLAB i Java för att utföra diverse beräkningar, och för detta ändamål har vi två mjukvara att utgå ifrån, nämligen MatlabControl av Google och JMatLink. Vi kommer framförallt att fokusera på MatlabControl eftersom den är ganska stabil och enkel att använda. Vi kommer att skapa ett användarinterface där användaren kan mata in olika parametrar, och utifrån dessa värden kommer control design att skapas för processen. Som man kan ana kommer en stor del av projektet att handla om GUI-programmering, då vi på något sätt också skall representera referenssignal, mätsignal och kontrollsignal till användaren. Därav kommer vi att utnyttja plugins till Eclipse för att förenkla vårt arbete, bland annat WindowBuilder som tillåter "drag and drop" av olika Swing/AWT-komponenter.

Projektmedlemmar

Soheil Afghani
Mehmet Cicek
Jaqub Ghairat
Hassan Mouhsen

Handledare

Gabriel Ingesson

GUI

Två stycken GUI kommer att finnas i programmet:

1. "MainGUI" kommer att hantera inmatning av parametrar från användare samt kunna starta och stoppa regulatören.
 - a. Parameters
 - i. Process transfer function
 - ii. Sampling interval
 - iii. State feedback pole placement
 - iv. Observer pole placement
 - b. Actions
 - . Start - Startar regulatören med de parametrar som sparats i Update.
 - i. Stop - Stoppar regulatören
 - ii. Update parameters - Läser in angivna parametrar.

State Feedback Controller

Parameters

State matrix (A): Sampling Intervall:

Input matrix (B): State feedback pole placement:

Output matrix (C): Observer pole placement:

Feedthrough matrix (D):

Warnings

...

Figur 1 - MainGUI

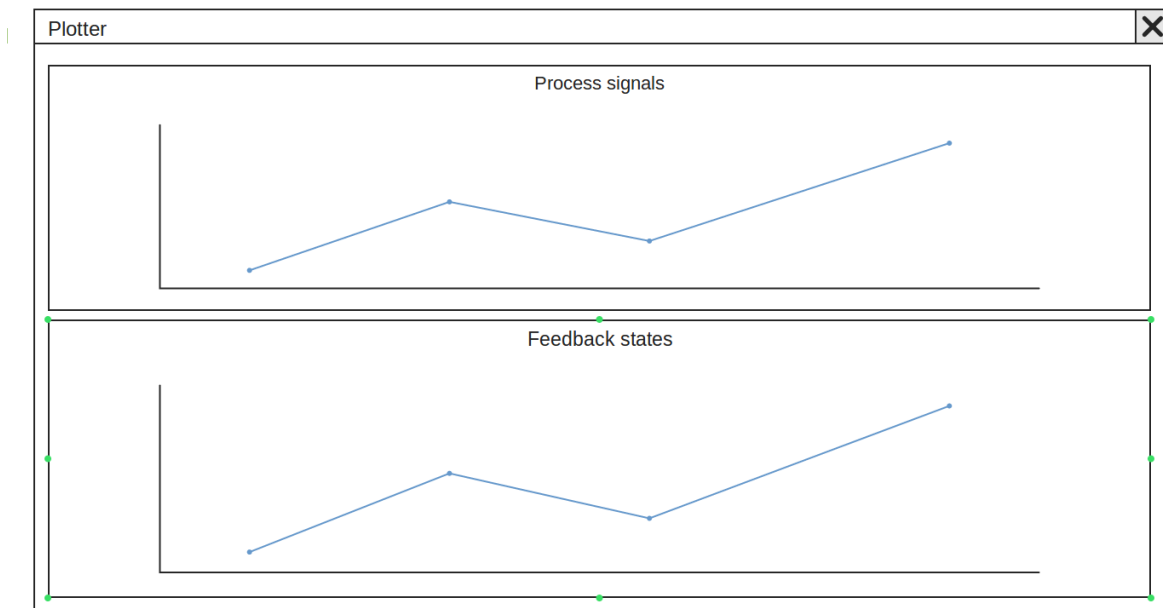
2. "PlotterGUI" kommer att hantera plottning.
 - a. Plottning av process signaler (u , y_{ref} och y)
 - b. Plottning av de olika tillstånden i state feedback controllern.

Projektmedlemmar

Soheil Afghani
Mehmet Cicek
Jaqub Ghairat
Hassan Mouhsen

Handledare

Gabriel Ingesson



Figur 2 - PlotterGUI

Förklaring av våra klasser

Refgen - Detta är klassen som kommer att skapa vår referenssignal. Vi kommer att utnyttja samma Refgen som fanns tillgänglig i laboration 1.

AnalogIn - Återanvänds från laboration 1. Används för att hämta mätsignalen.

AnalogOut - Återanvänds från laboration 1. Används för att sätta utsignalen till processen.

Regulator - Detta är vår tråd som reglerar och beräknar utsignalen för processen. Regul utnyttjar andra klasser för att hämta mätsignalen och sätta utsignalen. Innehåller även en referens till monitorn för att läsa/skriva av olika värden.

Validation - Hjälpklass till GUI för att validera input från användaren. Ifall valideringsfel skulle uppstå skriver man till varningsrutan på GUI. Innehåller därför en referens av GUI.

MatlabCommands - Den här klassen sköter kommunikation med MATLAB genom att utnyttja MatlabControl/JMatLink. Den här klassen används av Regulator.

MainGUI - Detta är vår användargränssnittet vi har presenterat tidigare i dokumentet där användaren kan mata in olika specifikationer såsom överföringsfunktion, samplingsintervall, och placering.

PlotterGUI - Används för att presentera våra signaler i ett användargränssnitt för användaren. Vi kommer att återanvända kod från laboration 1.

Monitor - Monitor som lagrar undan inmatade och beräknade värden.

Projektmedlemmar

Soheil Afghani

Mehmet Cicek

Jaqub Ghairat

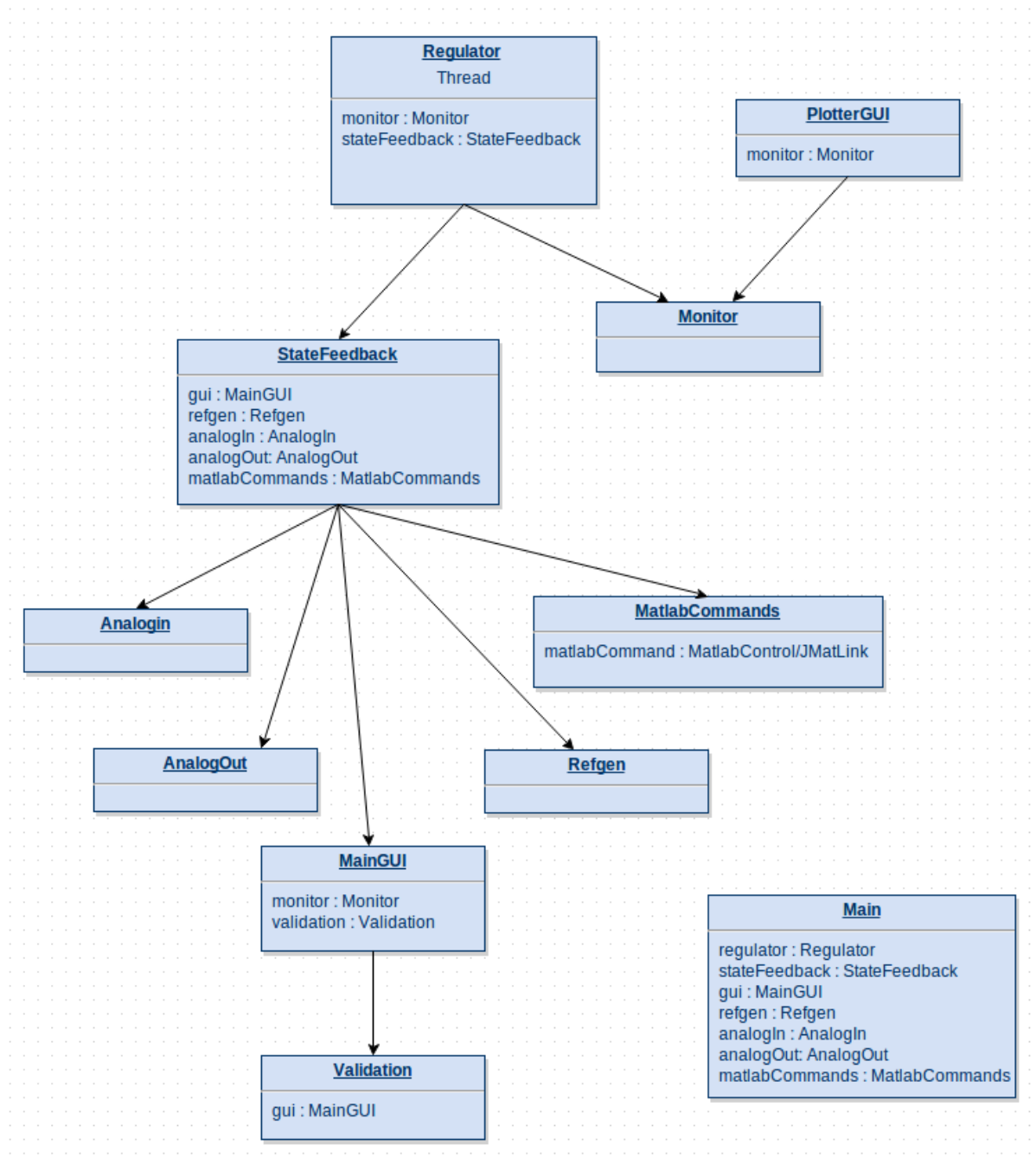
Hassan Mouhsen

Handledare

Gabriel Ingesson

StateFeedback - Klassen används för att utnyttja parametrar från underklasser till att beräkna modellen och styrsignalen för processen. Det vill säga regulatorn.

Main - Vår huvudklass som startar applikationen.



Figur 3 – UML Diagram

Control Design

State feedback and observer control

State feedback

Sampling av en tidskontinuerlig process ger oss det tidsdiskreta systemet i ekvation 1. Processens dynamik kan förändras genom att använda tillståndåterkoppling.

$$x(k+1) = \Phi x(k) + \Gamma u(k)$$

Ekvation 1 – Tidsdiskreta systemet

Målet är att det slutna systemet får en specificerad karakteristisk ekvation genom att designa en linjär regulator som ger systemet på formen.

$$u(k) = -Lx(k)$$

$$x(k+1) = [\Phi - \Gamma L]x(k)$$

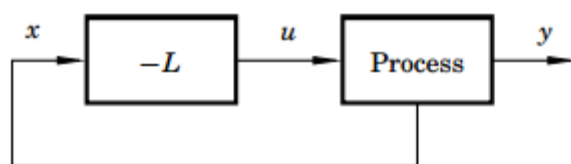
Ekvation 2 – Linjära regulatorn och det nya systemet

Om och endast om styrbarhetsmatrisen har full rank (systemet är styrbart) kan koefficienterna i L matrisen väljas genom att ansätta det karakteristiska polynomet mot det önskade.

$$\det(zI - \Phi + \Gamma L)$$

$$W_c = [\Gamma \quad \Phi\Gamma \quad \dots \quad \Phi^{n-1}\Gamma]$$

Ekvation 3 – Karakteristiskt polynom samt styrbarhetsmatrisen



Figur 4 - Tillståndåterkoppling

Observer

I praktiken är det ofta så att processens alla tillstånd inte går att mäta upp. Det man kan göra är att skatta tillstånden och om man då antar att \hat{x} approximeras av följande system.

$$\hat{x}(k+1) = \Phi \hat{x}(k) + \Gamma u(k)$$

$$\hat{y}(k) = C \hat{x}(k)$$

Projektmedlemmar

Soheil Afghani
Mehmet Cicek
Jaqub Ghairat
Hassan Mouhsen

Handledare

Gabriel Ingesson

Ekvation 4 – Tidsdiskreta systemet med skattade tillstånd

Där \hat{x} betecknar de skattade tillstånden. Det visar sig att observerarens tillstånd konvergerar mot processens tillstånd om systemet är asymptotiskt stabilt (förutsatt att systemet är observerbart). Detta kan uppnås genom att välja förstärknings-matrisen K så att systemet får önskade egenskaper genom att ansätta det karaktäristiska polynomet mot ett önskat polynom. Att systemet är observerbart innebär att observerbarhetsmatrisen har full rank.

Genom att introducera återkoppling via avvikelsen mellan modellens utsignal och processens utsignal får man följande form på systemet.

$$\hat{x}(k+1) = \Phi \hat{x}(k) + \Gamma u(k) + K(y(k) - C \hat{x}(k))$$

Ekvation 5 – Introducering av avvikelse ger följande system

Introducerar man det estimerade rekonstrueringsfelet ($\tilde{x} = x - \hat{x}$) kan systemet nu skrivas på följande form.

$$\tilde{x}(k+1) = [\Phi - KC]\tilde{x}(k)$$

Ekvation 6 – Nya systemet efter introduktion av rekonstruktionsfelet

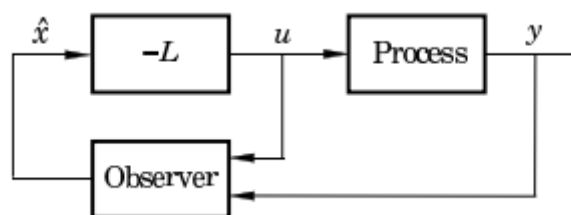
Det karaktäristiska polynomet och observerbarhetsmatrisen ges av följande formler.

$$\det(zI - \Phi + KC)$$

$$W_o = \begin{pmatrix} C \\ \vdots \\ C\Phi^{n-1} \end{pmatrix}$$

Ekvation 7 – Karaktäristiskt polynom samt observerbarhetsmatrisen

Genom att kombinera tillståndsåterkopplingen med observeraren fås följande regulator.



Figur 5 – Tillståndsåterkoppling och observerare

$$\hat{x}(k+1) = \Phi \hat{x}(k) + \Gamma u(k) + K(y(k) - C \hat{x}(k))$$

$$u(k) = -L \hat{x}(k)$$

Ekvation 8 – State feedback and observer controller

Projektmedlemmar

Soheil Afghani
Mehmet Cicek
Jaqub Ghairat
Hassan Mouhsen

Handledare

Gabriel Ingesson

Reference signals

Genom att ersätta vårt gamla u mot nedanstående kan vi på ett simpelt sätt uppnå så att processens utsignal följer ett referensvärde.

$$u(k) = L_C u_c(k) - L\hat{x}(k)$$

Ekvation 9 – Styrsignalen med adderad referensvärde och L_C

Där $u_c(k)$ är referenssignalen och L_C väljs så att den statiska förstärkningen blir 1. Då fås följande värde på L_C .

$$L_C = \frac{1}{C(I - \Phi + \Gamma L)^{-1} \Gamma}$$

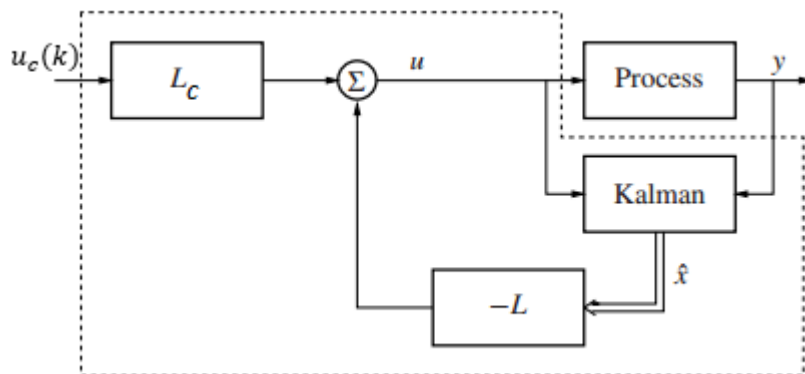
Ekvation 10 – Värdet på L_C så att statiska förstärkningen blir ett

Regulatorn blir nu på formen.

$$\hat{x}(k+1) = (\Phi - \Gamma L - KC)\hat{x}(k) + \Gamma L_C u_c(k) + Ky(k)$$

$$u(k) = L_C u_c(k) - L\hat{x}(k)$$

Ekvation 11 – Regulatorn med referensvärdeshantering



Figur 6 - Referensvärdeshantering

Projektmedlemmar

Soheil Afghani
 Mehmet Cicek
 Jaqub Ghairat
 Hassan Mouhsen

Handledare

Gabriel Ingesson

Integral Action

Introducerar vi ett extra tillstånd x_i som integralen av utsignalen kan vi eliminera stationära fel.

$$x_i(k+1) = x_i(k) + y(k)$$

Ekvation 12 – Introduktion av nytt tillstånd som integralen av utsignalen

Genom att introducera återkoppling med integralverkan blir den nya styrsignalen enligt följande.

$$u(k) = L_c u_c(k) - L \hat{x}(k) - l_i x_i(k)$$

Ekvation 13 – Styrsignalen med integralverkan

Detta medför att den slutgiltiga och fullständiga regulatorn med integralverkan samt tillståndsåterkoppling från de skattade tillstånden får följande utseende.

$$\begin{bmatrix} \hat{x}(k+1) \\ x_i(k+1) \end{bmatrix} = \begin{bmatrix} \Phi - \Gamma L - KC & -\Gamma l_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ x_i(k) \end{bmatrix} + \begin{bmatrix} K \\ 1 \end{bmatrix} y + \begin{bmatrix} \Gamma L_c \\ 0 \end{bmatrix} u_c(k)$$

$$u(k) = -[L \quad l_i] \begin{bmatrix} \hat{x}(k) \\ x_i(k) \end{bmatrix} + L_c u_c(k)$$

Ekvation 14 – State feedback and observer controller with integral action

Projektmedlemmar

Soheil Afghani
Mehmet Cicek
Jaqub Ghairat
Hassan Mouhsen

Handledare

Gabriel Ingesson

Tidsplan

Vi är fyra i vår grupp som skall utveckla projektet. Därav är vi i behov av ett versionshanteringssystem för att koordinera arbetet mellan oss. Vi kommer med största sannolikhet att använda GitHub som CMS och dela upp arbetet där det går.

Beräknad tid som kommer läggas på projektet är ca 4 veckor. Vi kommer arbeta ett par timmar varje dag (Mån-Fre). Vi kommer till största del arbeta tillsammans. Vi kommer dela upp arbetet när vi väl kan arbeta parallellt med projektet.

v2	<ul style="list-style-type: none">• Arbeta med Regul-tråden och parallellt påbörja med GUI programmering. Vi bekantar oss mer med kommunikationen mellan Java och Matlab (MatlabControl).• Träff med handledaren.
v3	<ul style="list-style-type: none">• Fortsatt utveckling kring klasser som Regul-tråden kommer att använda, bland annat monitor-klass(er) och Refgen/Validation. Dessutom börjar vi implementeringen av JavaToMatlab, där MatlabControl börjar användas.• Träff med handledaren.
v4	<ul style="list-style-type: none">• Fotsätta och avsluta GUI programmeringen. Börja utnyttja klasserna som läser av de riktiga värden från processen.• Träff med handledaren.
v5	<ul style="list-style-type: none">• Testa programmet på en eller två av processerna som finns tillgängliga i labsalen, tex double tank eller DC servo. Fixa eventuella fel.• Träff med handledaren.