

## **Datorövning 2 Matlab/Simulink**

### **i**

## **Styr- och Reglerteknik för U3/EI2**

Laborationen förutsätter en del förberedelser som genomläsning och repetering av vissa begrepp inom kursen. Det som inte hinns med vid laborationstillfället bör gås igenom på egen hand för att dels lösa inlämningsuppgifter alternativt behövas vid ett senare tillfälle. I denna laboration skall studeras olika återkopplade processer och deras stegsvar, poler och kvarstående fel. Vi ska också använda PID-reglering. Detta görs m h a Simulink och Matlab. Spara gärna m-filer och mdl-filer till exempelvis inlämningsuppgifter eller egna självstudier. Vi kommer mest använda oss av simulink

**Tid: ca 2timmar. OBSERVERA: alla uppgifter redovisas.**

**Kap 24 i kursboken sid 476-503 beskriver lite mer allmänt hur simulering av reglersystem med Matlab och Simulink går till.**

Starta matlab.

Ni har nu kommit in i kommandofönstret. Här kan ni direkt ge kommandon och få direkt respons på inmatningar. Starta simulink genom att skriva simulink i kommandofönstret. Nu har ni ett antal fönster öppna . Biblioteksfönster och ett namnlöst editeringsfönster.

Kopiera över de objekt som behövs.

Från **Simulink-> Continuous** hämtas **Transfer Function** (överföringsfunktion).

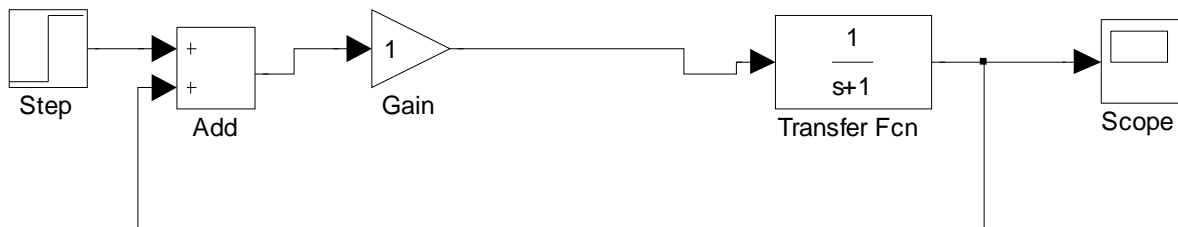
Därefter hämtas från **Simulink->Math Operations** blocken **Add** och **Gain**.

Blocken **Step** och **Scope** hämtas från: **Simulink->Sources** samt **Simulink-> Sinks**.



Figur 1

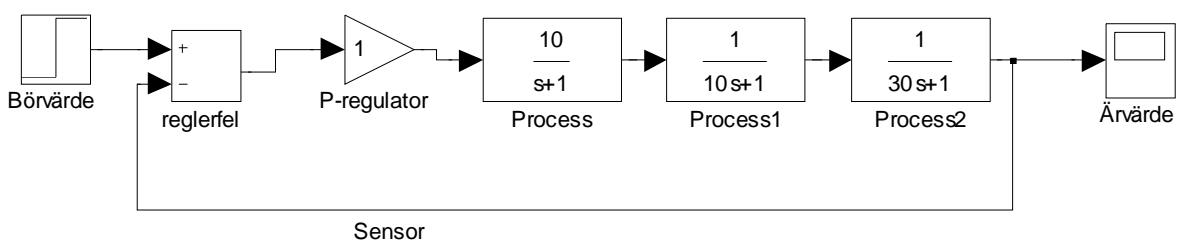
Förbind alla blocken det görs med höger musknapp när denna förs mot pilspetsen så kan ni dra förbindningar mellan blocken. Gör så att nedanstående Figur 2 erhålls.



Figur 2

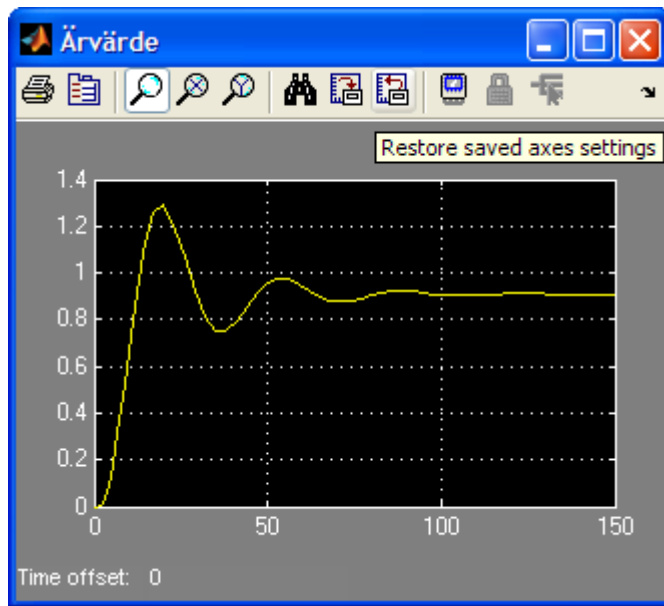
Nu återstår några detaljer att göra. Ändra i blocket Add så att vi får en negativ återkoppling.

Vi ändrar blocknamnen också. Detta görs genom att dubbelklicka med musen på befintlig text eller var ni vill i editeringsfönster. Det kan då se ut enligt nedan.



Figur 3

Simulera systemet och titta på stegsvaret. Förläng simuleringstiden till 150 sek.



Figur 4

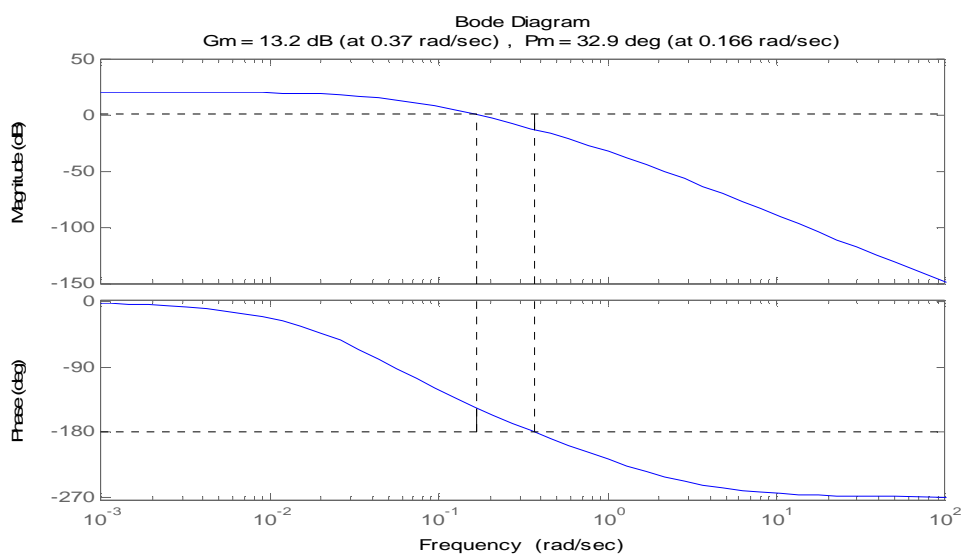
Gå nu tillbaka till Matlabs kommando fönster (command window) och definiera:  
 Det som står efter %-tecken är bara en kommentar.

```
>> Gp1=tf([10],[1 1])      % processens delöverföringsfunktion
>> Gp2=tf([1],[10 1])    % processens delöverföringsfunktion
>> Gp3=tf([1],[30 1])    % processens delöverföringsfunktion
>>
>> Gr=1                   % regulatorns överföringsfunktion
>> Gp= Gp1*Gp2*Gp3;      % Multiplicerar ihop överföringsfunktionerna hos processen.
```

Definiera kretsöverföringen och rita upp Bodediagram för avläsning av stabilitetsmarginaler

```
>> Gk=Gr*Gp              % överföringsfunktion kretsöverföring
>> margin(Gk)            % ritat Bodediagram för kretsöverföringen och visar stabilitetsmarginaler.
```

I figur visas amplitudmarginal ( Gain Margin- Gm) och fasmarginal (Phase Margin –Pm).  
 Normalt står även överkorsningsfrekvensen ( $\omega_c$ ) självsvängningsfrekvensen ( $\omega_r$ ) inom parentes.



Figur 5

Jag tror det framgår ganska väl vad som är vad i figuren. Är systemet stabilt ?  
Det skulle vara intressant att veta var reglersystemets poler ligger i vårt fall för en P-regulator  $K=1$ .  
För att göra detta så bildar vi först det totala systemets överföringsfunktion.

```
>> Gtot=feedback(Gk,1) % konstruerar det totala reglersystemets överföringsfunktion.  
% Det andra argumentet 1 står för sensorns överföringsfunktion.  
% Förutsätter negativ återkoppling.
```

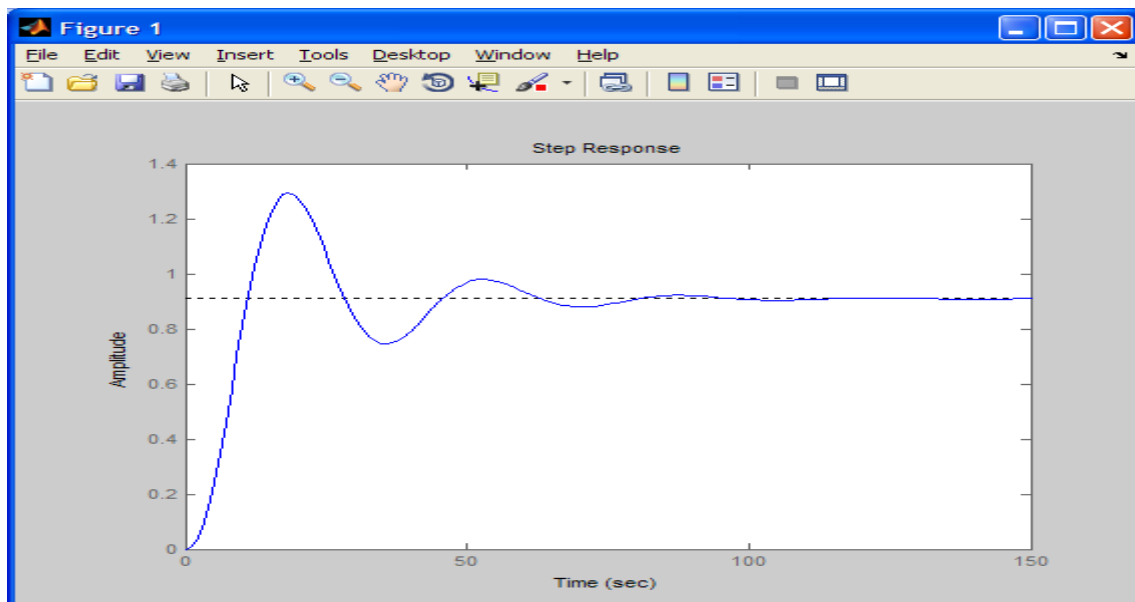
Resultatet bör bli nedanstående.

Transfer function:

$$\frac{10}{300 s^3 + 340 s^2 + 41 s + 11}$$

Vi behöver inte rita någon modell i Simulink för att simulera stegsvar. Det kan göras direkt i Matlab också.

```
>> step(Gtot, 150)
```



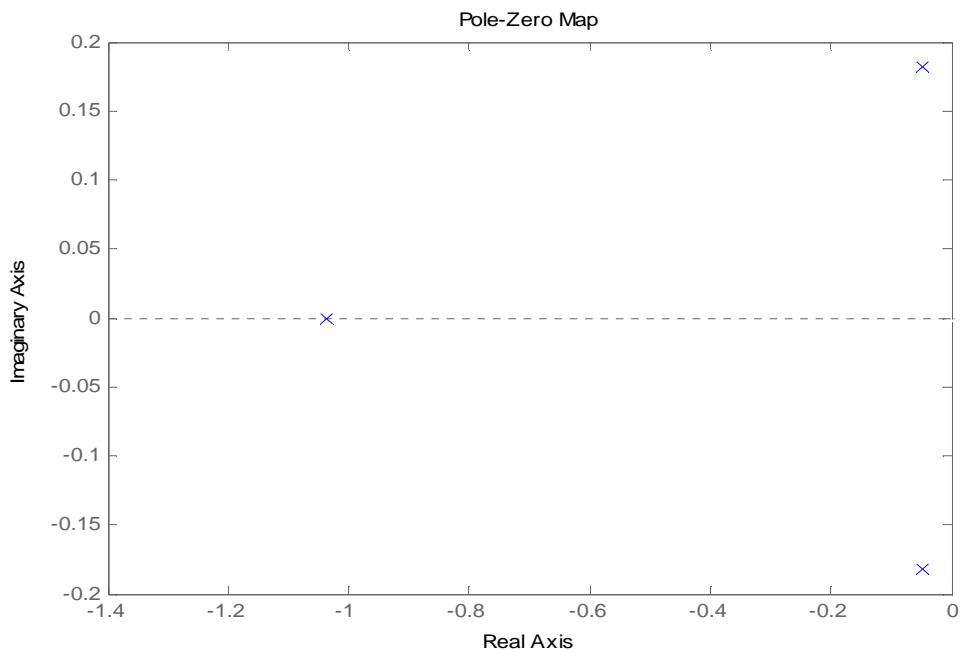
Figur 6

Om ni har gjort rätt så har ni fått samma stegsvar som i figur 4.

Nu skulle vi även ta reda på var systemets poler och nollställen går med kommandot pzmap !

Vi har bara poler i vårt fall ( 3 stycken).

```
>> pzmap(Gtot)
```



Figur 7

Är systemet stabilt ?

Om vill slippa skriva om alla kommandon för varje gång vi vill göra en ändring någonstans kan vi helt helt sätta samman en m-fil. Det är ett litet program där vi samlar alla kommandon och kör alla kommandon direkt. Öppna ett editeringsfönster med:

File-> New->M-file

Skriv in kommandon där. Se figur 8 !

Skippa alla kommentarsrader om du vill spara tid. Spara m-filen som något.

Jag gav m-filen namnet "simulering.m". För att köra programmet kan man trycka på den gröna pilen i editorn eller gå in under Debug-> Run. Alternativt kan man skriva namnet på m-filen i matlabs kommandofönster.

För att matlab skall hitta m-filen måste man välja att stå i den katalogen när man kör m-filen.

>> simulering

```

1 % M-fil skapad 081111 av Thomas Munther
2 % Filen ritat ett Bodediagram, stegsvar, poler och nollställen.
3
4 Gr=1 % P-regulatorns förstärkning
5 Gp1=tf(10,[1 1])
6 Gp2=tf([1],[10 1])
7 Gp3=tf([1],[30 1])
8 Gp=Gp1*Gp2*Gp3 % processens överföringsfunktion
9 Gk=Gr*Gp; % kretsöverföringen
10
11 figure(1) % öppnar ett figurfönster 1
12 margin(Gk) % plottar bodediagrammet med stabilitetsmarginaler. De hamnar i figurfönster 1.
13
14 Gtot=feedback(Gk,1)
15 figure(2)
16 step(Gtot,150)
17 figure(3)
18 pzmap(Gtot)
  
```

Figur 8

Resultatet bör bli samma tre figurfönster som vi fick förut när vi körde från kommandofönstret. Fast nu ligger de i en hög på varandra. Se om de stämmer !

- a) Upprepa nu och simulera reglersystemet i figur 3 med en P-regulator=2 !  
Tag upp Bodediagrammet med stabilitetsmarginaler, stegsvaret och plotta med poler och nollställen.

Notera stabilitetsmarginaler, poler, kvarstående fel samt principiella utseendet på stegsvaret !

- b) Upprepa nu och simulera reglersystemet i figur 3 med en P-regulator=3 !  
Tag upp Bodediagrammet med stabilitetsmarginaler, stegsvaret och plott med poler och nollställen.

Notera stabilitetsmarginaler, poler, kvarstående fel samt principiella utseendet på stegsvaret !

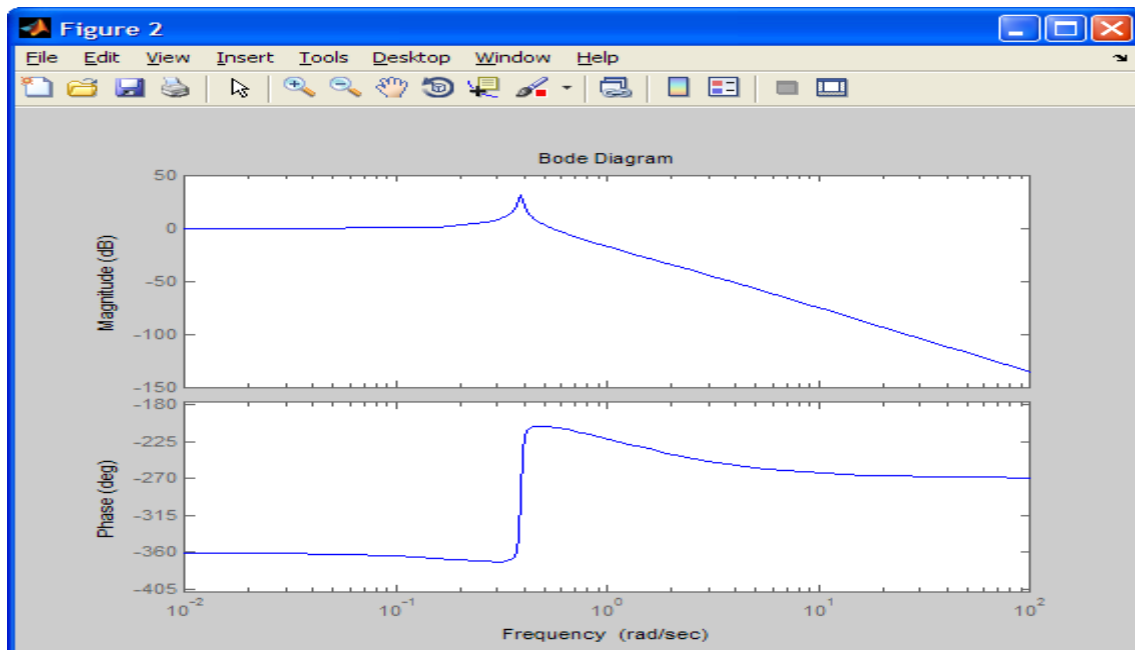
- c) Upprepa nu och simulera reglersystemet i figur 3 med en P-regulator=5 !  
Tag upp Bodediagrammet med stabilitetsmarginaler, stegsvaret och plott med poler och nollställen.

Notera stabilitetsmarginaler, poler, kvarstående fel samt principiella utseendet på stegsvaret !

- d) Kan ni räkna ut för vilket K-värde hos P-regulatorn som systemet blir instabilt ?  
Prova detta genom att köra m-filen igen !

Kommandot **margin** skall användas om vi ritar upp kretsöverföringen för det är bara om denna plottas som vi får fram stabilitetsmarginaler. Om vi bara vill rita ett Bodediagram för en allmän överföringsfunktion så använder vi kommandot **bode** istället.

```
>> bode(Gtot) % Bodediagram utan stabilitetsmarginaler.
```



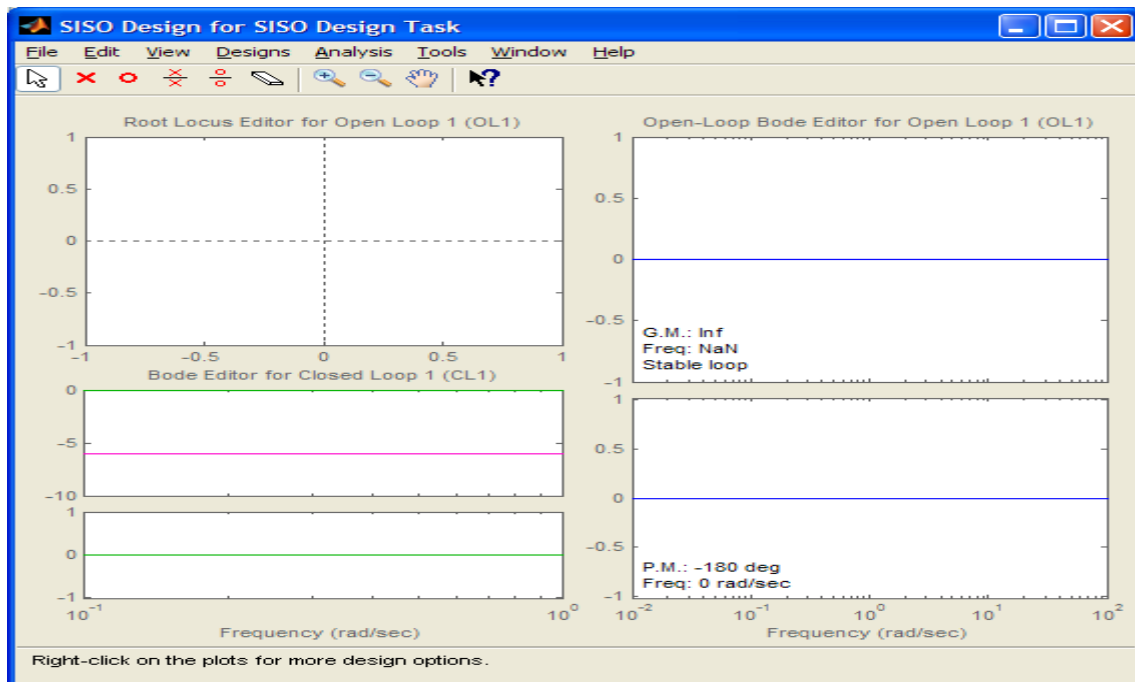
Figur 9

I figur 9 visas det totala systemets Bodediagram. Ur denna kan man inte avläsa stabilitetsmarginaller men det finns andra saker som bandbredd, resonanstopp och statisk förstärkning som ger viss information.

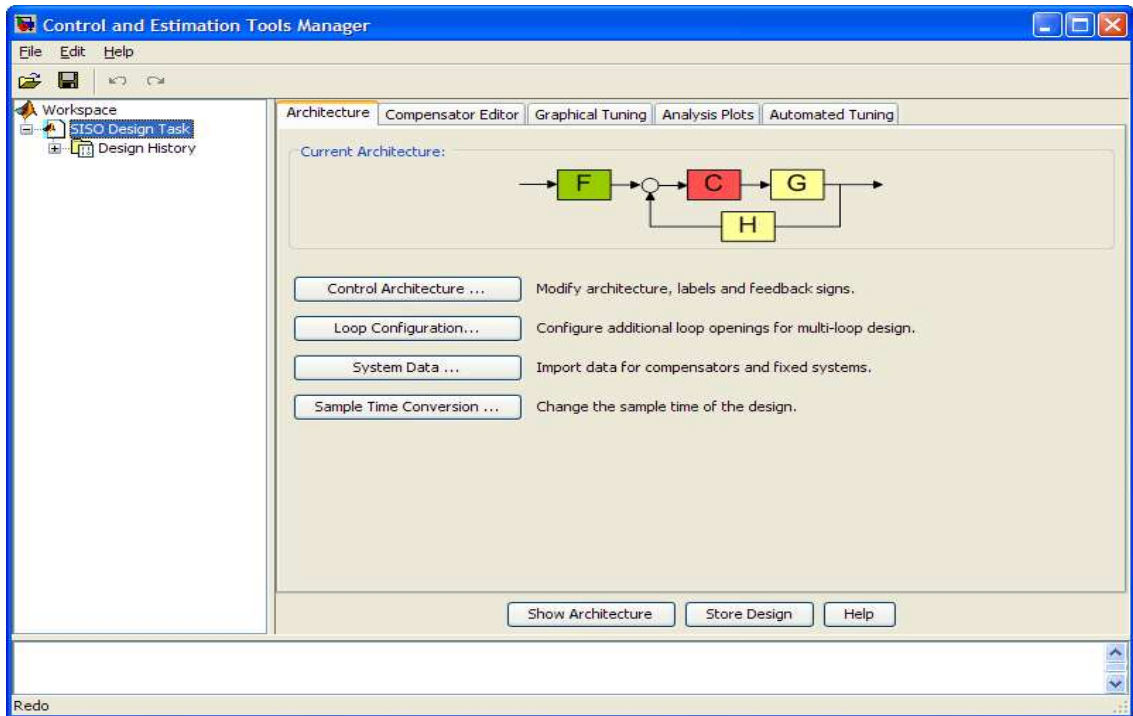
Vi skall kolla ännu ett program (m-fil) inne i matlab uppbyggt av någon med sinne för reglerteknik och GUI:s (Graphical User Interface –grafiskt användargränssnitt). Programmet startas genom att skriva sisotool i matlabs kommandofönster.

```
>> sisotool
```

Nu öppnas två nya fönster.



Figur 10

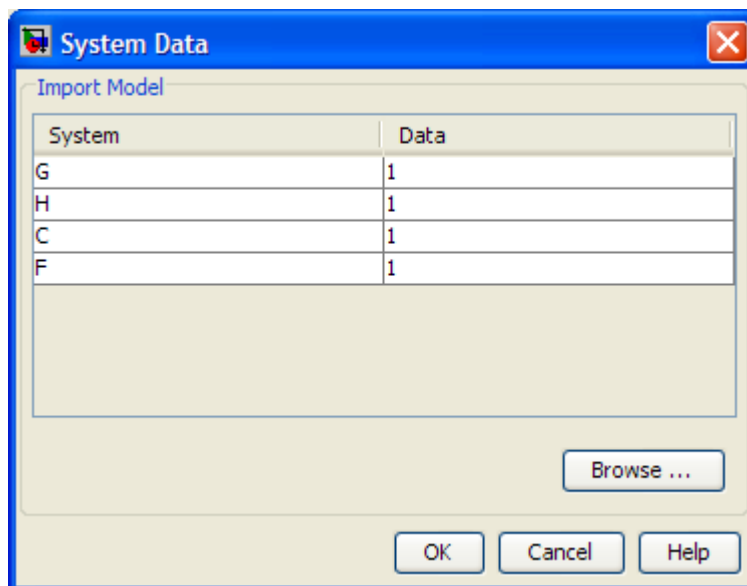


Figur 11

Gå nu in i fönster som heter Control and Estimation Tools Manager.

Välj Compensator Editor och sätt  $C=3$  (P-regulator).

Vidare så klickar du även på System Data. Där visas vilka överföringsfunktioner vi har i vårt reglersystem i figur 11. Vår process är en aning enkel. Vi skall byta ut denna mot den som vi använt under laborationen nämligen  $G_p$ . F(filter =1) och H (sensor =1) skall båda vara oförändrade.

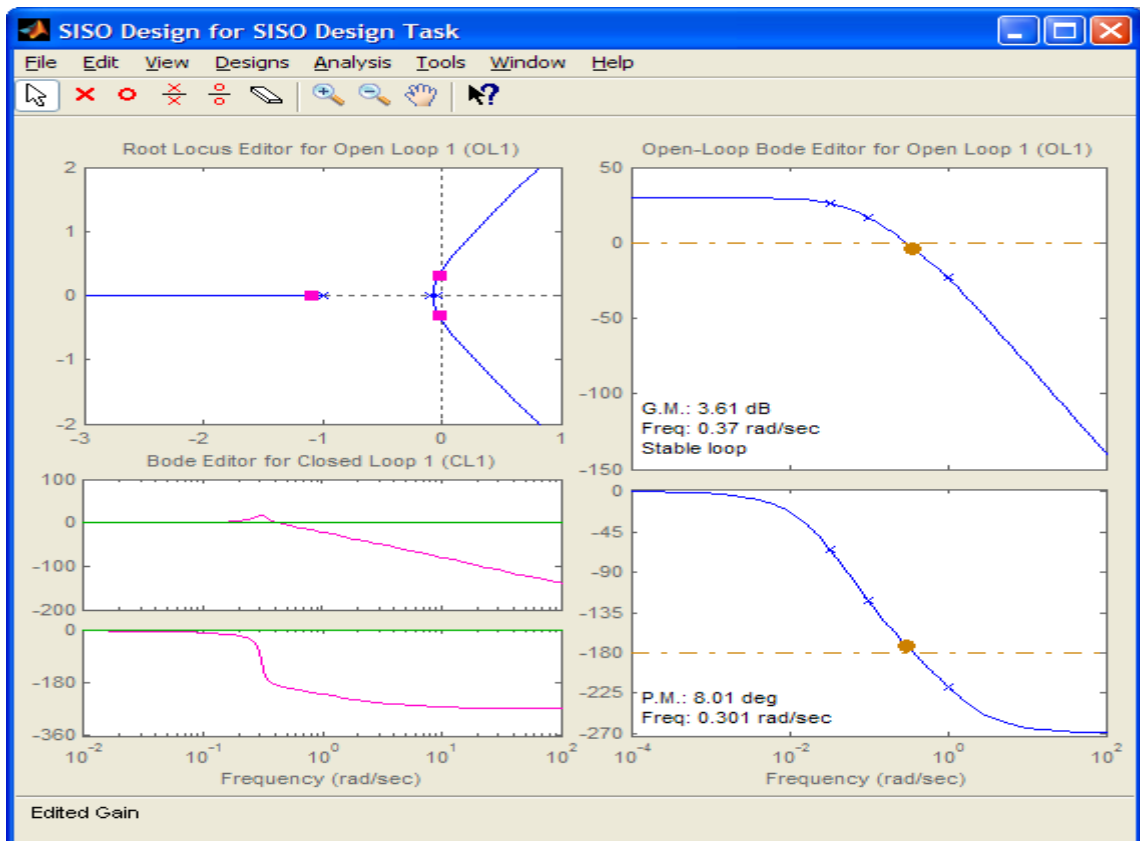


Figur 12

Markera G (process) och välj Browse för att välja in vår tidigare process  $G_p$  som den nya processen.

Gå nu tillbaka till fönstret Siso Design for Siso Design Task. Ni kan nu se att denna har förändrats här kan man t ex se. Pol- och nollställen, Bodediagram (kretsöverföring) och Bodediagram (totala systemet).

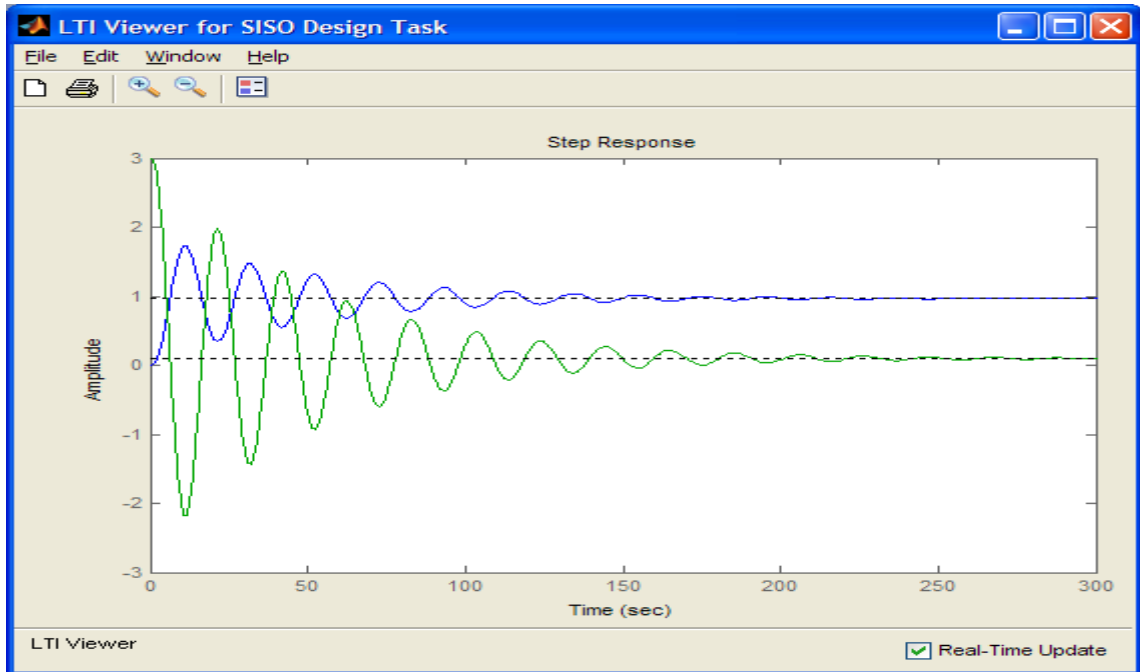




Figur 13

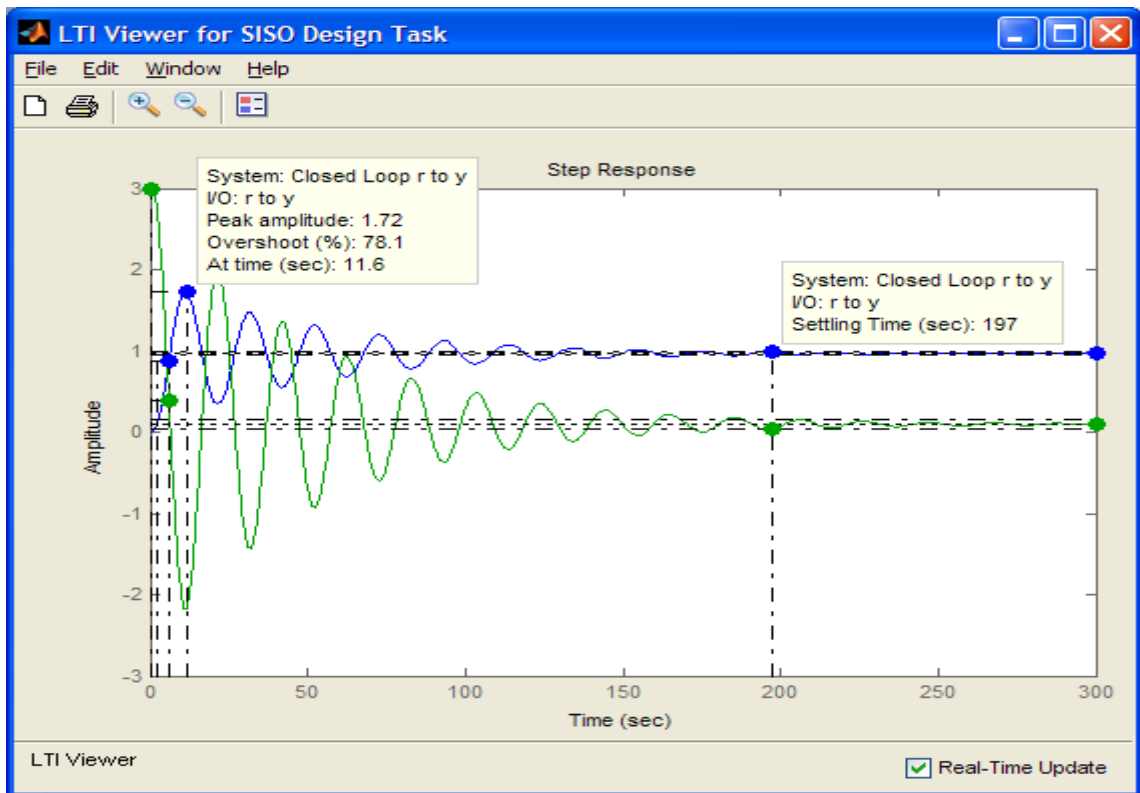
Ovanstående kurvor har vi sett tidigare när vi kört m-filen simulering för en P-reg=3 som vi skapade tidigare. Det roliga med detta att här vi riktig interaktivitet. Titta noga på figur 13 så ser ni i Bodediagrammet med kretsöverföringen 2 prickar och i den övre vänstra (poler och nollställen) tre kvadrater. Dessa går att ta tag i och då förändras momentant övriga figurer. Prova !

Det skulle vara kul att se hur andra signaler i reglersystemet beter sig när vi ändrar i dessa figurer. T ex stegsvaret för det slutna reglersystemet eller styrsignalen. Det kan vi lätt ordna genom att gå in i menyn under Analysis-> Response To Step Command. Nu öppnas ett nytt fönster . Se figur 14 ! Den innehåller både stegsvaret (blå) och styrsignalen (grön).



Figur 14

Tar man nu höger musknapp inuti plottningsytan kan man välja och få fram Characteristics. Här kan man alltså få fram direkt data ifrån stegsvaret. Figur 15 visar hur det kan se ut. Peaktid, översväng, insvängningstid visas i figur 15 nedan !



Figur 15

Om ni drar i poler, amplitud- eller faskurvan i figurfönster Siso Design for Siso Design Task så ser ni interaktivt vad som händer.