Högskolan i Halmstad
Sektionen för Informationsvetenskap, Dator- och Elektroteknik
061025/ Thomas Munther

SIMULERINGS LABORATION

i

Styr- och Reglerteknik för E2/D2/Mek2

Målsättning: få en inblick i en programvara som kan användas för att simulera reglersystem/ processer. Få en repetition av vissa grundbegrepp i reglerteknik. Simulera de processer som har identifierats i laboration 1 och försöka ta fram lämpliga regulatorer.

Laborationen förutsätter en del förberedelser som genomläsning och repetering av vissa begrepp inom kursen.
I denna laboration skall studeras olika återkopplade processer och deras stegsvar, poler och kvarstående fel.
Vi ska också använda Bodediagram och 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

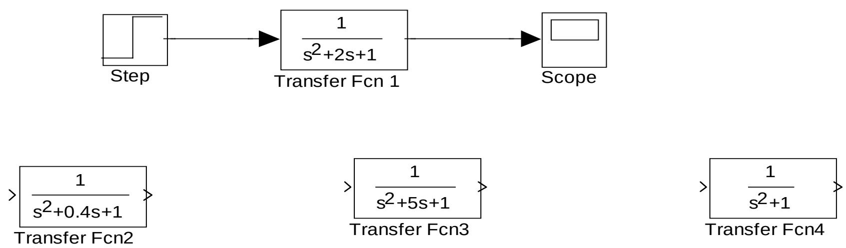
Tid: ca 4timmar. OBSERVERA: alla uppgifter redovisas.

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.
Öppna **sources, sinks, linear** och editeringsfönster. Kopiera över de objekt som behövs.

Uppgift 1

Kopiera över Scope, Step och Transfer function och koppla ihop dessa enligt nedan.
Klicka på dessa för att ändra parameterinställningar. Testa 4 olika system och avläs stigtid, insvängningstid(2%), överläng (i %) för dessa stegsvar.

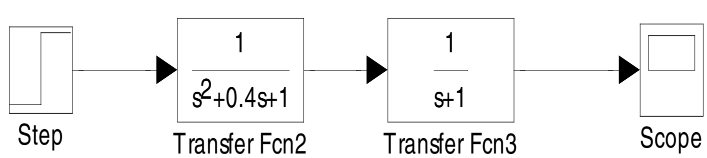


Uppgift 2:

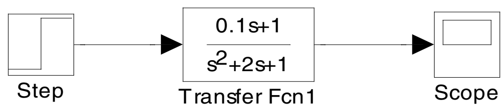
Bestäm polernas lägen för systemet i uppgift 1 och relatera dessa till stegsvaren !
Se lathunden i appendix för användbara matlabkommandon. Tips : **roots**([1 2 1]) ger rötterna till nämnaren i det översta systemet i uppgift 1.
Ger polernas explicita värden resp. **pzmap**([1],[1 2 1]) ritar ut polerna i komplexa talplanet och nollställen (om det finns några).
Notera skrivsättet först anges täljarkoefficienterna inom hakparenteser och därefter nämnarkoefficienterna allt i fallande potensordning.

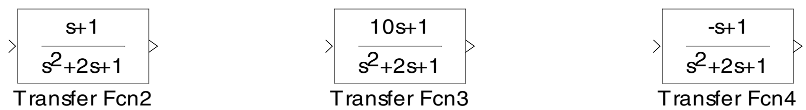
Uppgift 3:

Antag att system 2 ifrån uppgift 1 saknar en faktor i nämnaren. Testa med 2 olika faktorer (s+1) respektive (100s+1). Kan ni förklara utseendet hos stegsvaret ?



Uppgift 4: Undersök vilken påverkan eventuella nollställen har på stegsvarets utseende.
Ange orsak till utseendet. Relatera till poler och nollställen.
Testa med 4 olika nollställen för system 1 i uppgift 1.
System som har sina nollställen och poler i VHP kallas för minimum-fas system.
Ett av våra system här är exempel på ett icke-minimum-fas system. Det har ett obehagligt uppförande.





Uppgift 5:

I nedanstående figur har det lagts till block in, tid och ut som är av typen **to workspace.**
Glöm inte att ändra i dessa block och välja format array !
D v s det finns möjlighet att exportera in- och utdata som vektorer till kommandofönstret i Matlab. Testa gärna detta efter att ni har simulerat.
Gå ut i kommandofönstret och gör samma plottning som fås med Scope.
Skriv **plot**(tid,ut); **grid; zoom** i kommandofönstret.

a) Antag att vi har en process enligt nedan och den ska regleras m h a en P-regulator.
Ni hämtar en PID-regulator ifrån biblioteket **Simulink Extras**->**Additional linear->PID-controller.** I-delen och D-delen skall båda sättas till 0.
Vi kan i vårt reglersystem försumma givarens dynamik och ärvärdet är negativt återkopplat.
Undersök nu följande: för vilka regulatorförstärkningar som systemet är asymptotiskt stabilt.
Börja med K=1 och öka denna. Avläs för K=1 parametrarna : insvängningstid(2%), stigtid, översläng och kvarstående fel. Vad händer med dessa parametrar då K ökas ?

b) Vi ska nu introducera en PI-regulator. Sätt K=1 och undersök olika I-regulatorer.

b) Prova Ti=100, 10 och 1 sek. **Notera att ni skriver in I-delen i PID-regulatorn som K/Ti.** Hur stora är kvarstående felen ?
Vad händer med övriga parametrar i insvängningsförloppet ?

c) Sätt K=1 och Ti=10 sek. Komplettera nu regulatorn med en D-regulator. Prova Td=0.1, 1 och 10 sek. Vad händer med insvängningsförloppet ?
Hur påverkas kvarstående felet ?

Ge en sammanfattande beskrivning av respektive regulatordels inverkan på stegsvaret ?

6. a)Vad skulle hända om förstärkningen hos sensorn förändrades från 1 till 0.9 i ovanstående reglersystem med en PD-regulator ?
Sätt P=1 och D=0.1 ! Vilken blir skillnaden mot ett reglersystem med en PD-regulator och en vanlig sensor (=1)?

b) Vad skulle hända om förstärkningen hos sensorn förändrades från 1 till 0.9 i ovanstående reglersystem med en PI-regulator ?
Sätt P=1 och I=0.1 ! Vilken blir skillnaden mot ett reglersystem med en PI- regulator och en vanlig sensor (=1) ?

Uppgift 7:

Gå tillbaka till det ursprungliga systemet i uppgift 5a där K=1. Undersök stabiliteten m h a ett Bodediagram för detta system. Tag reda på parametrarna: självsvängningsfrekvens, överkorsningsfrekvens, fasmarginal och amplitudmarginal.
Öka K till 2.5. Vilka stabilitetsmarginaler har vi nu ?
Tips: För att ta reda på stabilitetsmarginalerna går vi till matlabs kommandofönster och använder följande funktion :**margin([0.4],[1 1.2 1.2 0.2]);** eller skriver så här :
talj=[0.4];namn=[1 1.2 1.2 0.2];
sys=**tf**(talj,namn); **margin**(sys)

Tänk nu på vad man behöver rita upp i ett Bodediagram för att bestämma stabilitetsmarginaler (Kretsöverföringen).

alternativ: se appendix bode1.m

Uppgift 8:

För att undersöka ett återkopplat system m a p stabilitet ritas det öppna systemet upp i bodediagrammet (även kallat kretsöverföringen) under antagande om negativ återkoppling, men även det slutna reglersystemet kan ritas upp i ett bodediagram fast då utläser man andra saker. Rita upp det slutna systemet i uppgift 5a i ett Bodediagram för K=1.
Bestäm för denna statisk förstärkning, bandbredd samt eventuell resonansfrekvens och resonanstopp. **Tips**: ni kan inte använda **margin** här eftersom denna förutsätter att ni använder kretsöverföringen som argument. Gör istället så här i kommandofönstret:
talj=[0.4];namn=[1 1.2 1.2 0.2];
sys=**tf**(talj,namn);
Gtot=**feedback**(sys,1)
bode(Gtot)

Upprepa det för K= 2 och 3 ! Vad händer med ovanstående uppmätta parametrar ?
Alternativ: se appendix bode1.m

Uppgift 9:

I denna uppgiften skall vi undersöka hur polerna vandrar då vi t ex ändrar förstärkningen i regulatorn. Beräkna och rita rotorten för det återkopplade systemets karakteristiska ekvation.
Detta görs för K=0 till K=10 , där K ändras i steg om 0.5 .
Denna uppgiften löses förslagsvis i Matlab,. Låt systemet i uppgift 5a styras med en P-regulator. För vilka värden på förstärkningen är det återkopplade systemet asymptotiskt stabilt (går mot ett stationärt värde efter lång tid). Skapa en vektor i matlab K=0:0.5:10;
Tips: definiera täljaren som talj=[0.4];och nämnaren som namn=[1 1.2 1.2 0.2]; vår överföringsfunktion blir sys=**tf**(talj,namn). Plotta rotorten med **rlocus**(sys,K).
Försök avgöra för vilket K-värde som polerna hamnar i HHP. Det kan vara svårt.
Använd funktionen **rlocfind**(sys)och markera på plottningen. Nu visas K-värdet.
Finns det något samband till uppgift 6 och 5a. Hur ser det ut ?

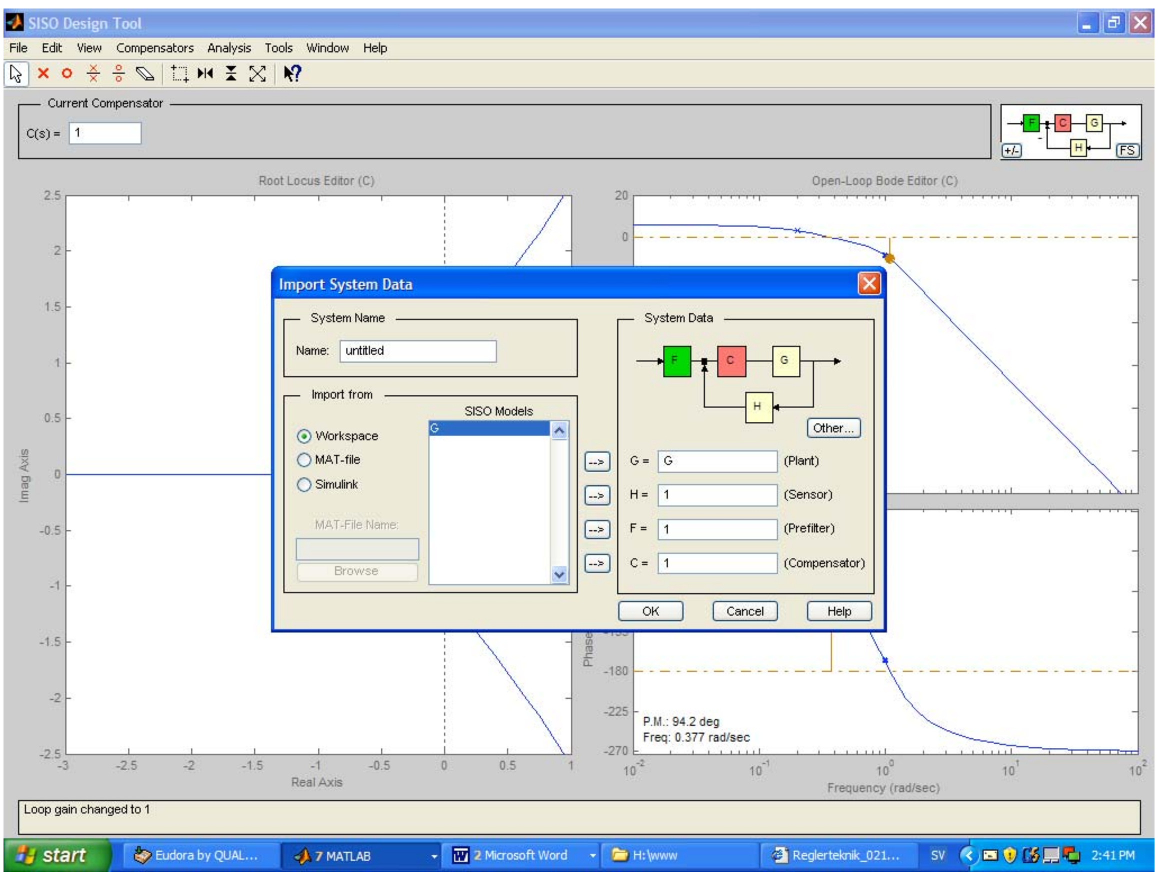
Alternativ: se appendix Rotort.m

Uppgift 10.

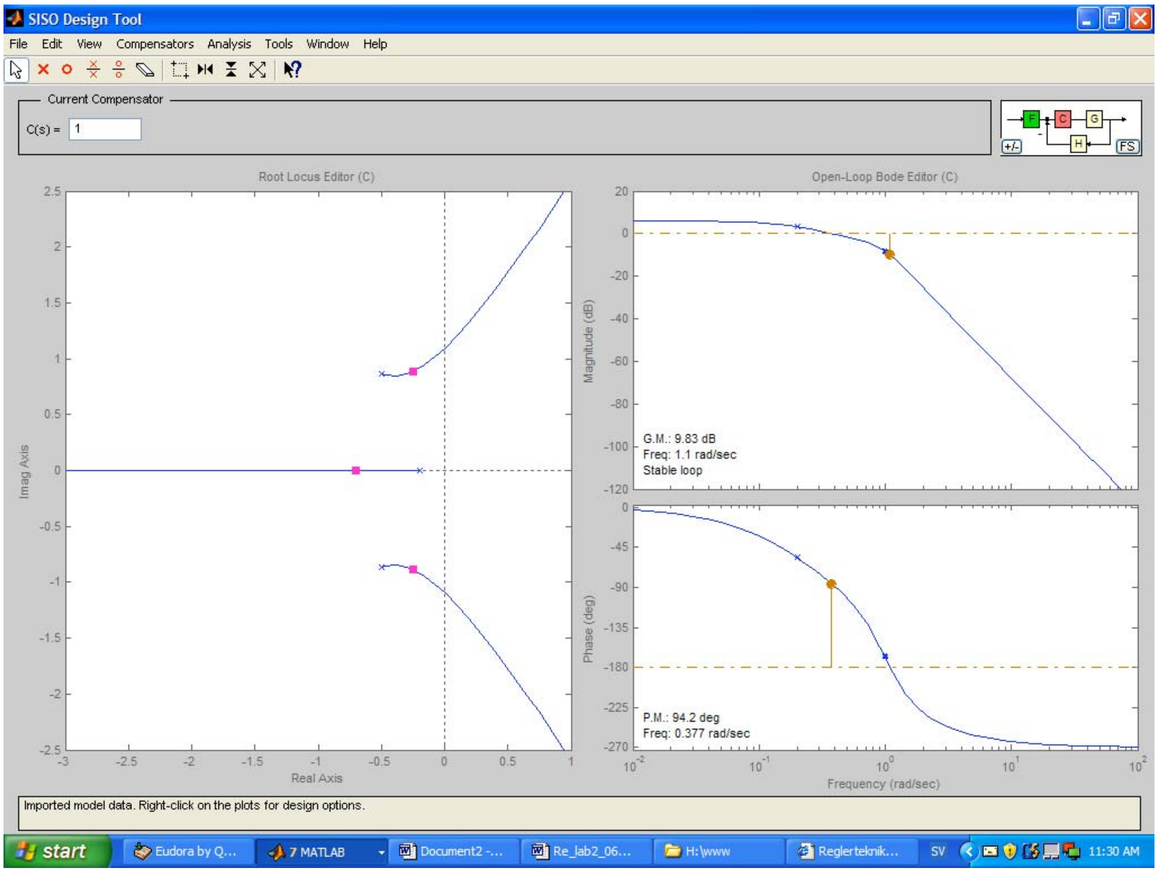
I denna uppgift skall vi titta på ett verktyg till i Matlabs värld. Gå ut i Matlabs kommandofönster och skriv **sisotool.**
Detta är ett interaktivt grafiskt användargränssnitt gjort för SISO system (1 insignal, 1 utsignal), som skall underlätta framtagning av regulatorlänkar i återkopplade reglersystem.

Uppe till höger finns ett blockschema som föreställer ett reglersystem. Där C är regulator, F är ett filter, H är sensorn och G är processen. Se till att systemet är negativt återkopplat genom att klicka +/- knappen. Regulatorn kan ligga på två ställen genom att klicka på FS-knappen får ni den på det stället ni vill ha den. Gå till **File->Import.** Markera G och klicka över med knapp -> till plant (process) därefter ok !
Under förutsättning att G finns definierad i kommandofönstret. Se figur 1 !

Se till att **Compensators-> Edit** och välj att F=1 (gain). Vi ska inte ha något filter.
Gå även in och ta bort eventuella poler och nollställen hos regulatorn , med **Compensators-> Edit** och välj C. Markera delete och sedan **ok** ! När detta är gjort visas 3 stycken fönster (Figur 2).
Det stora vänstra heter Root Locus (Rotorten på svenska – visar var slutna systemets poler ligger).
De högra fönstrena är amplitud- och fasfunktion, d v s för kretsöverföringen. Förhoppningsvis känner ni igen A_m och φ_m från uppgift 5.



Figur 1

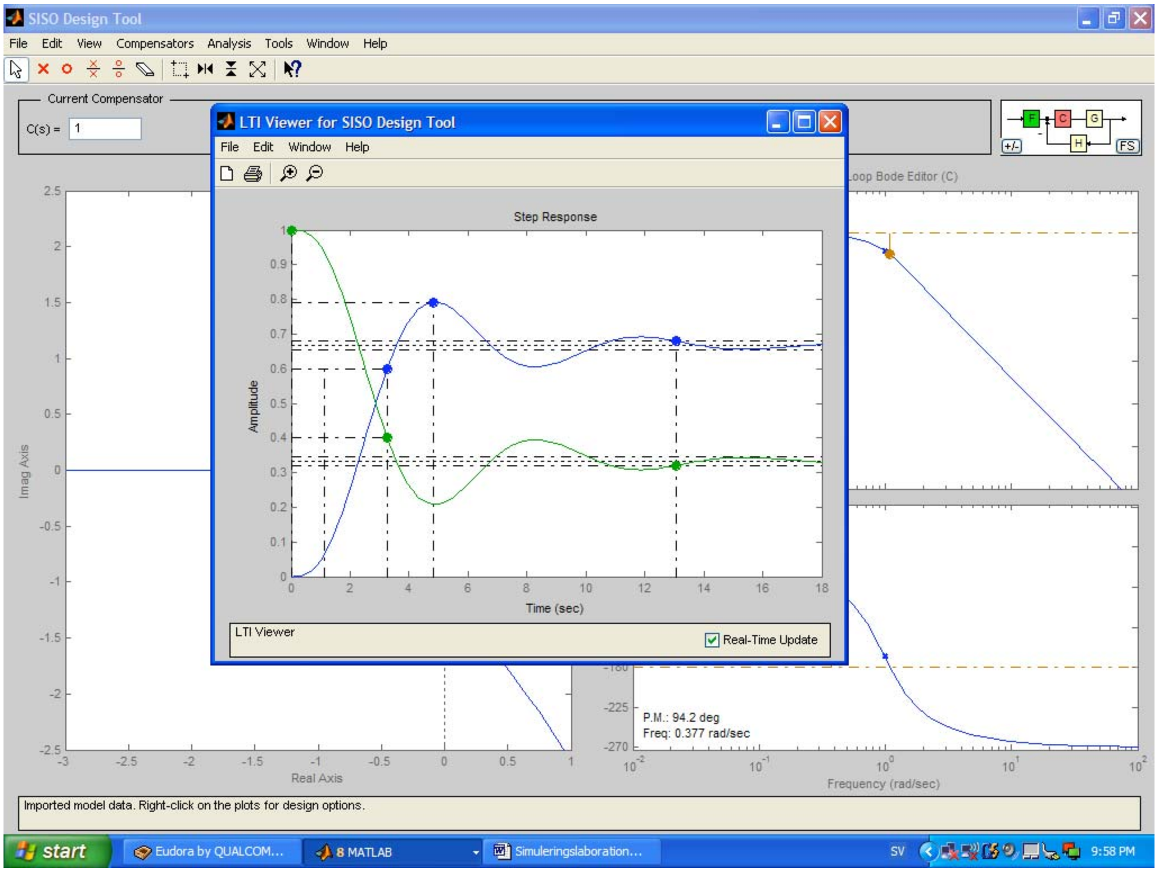


Figur 2

Aktuell regulator visas nu som **Current Compensator.**
Ni kan även klicka på H resp G för att se vilken sensor samt process vi har. Sensorn är lika med 1 medan processen är ganska komplicerad. Nu ska vi se till vår process sys hämtas in som vi skrev i matlabs kommandofönster i tidigare uppgift 7.

Nu är det dags att testa interaktiviteten. Nu ser ni tre fönster, amplitud- och faskurva för kretsöverföringen och rotorten. Ni ser några rosa kvadrater i rotortsfönstret . Det är polernas lägen i det slutna reglersystemet för just denna regulatorn. När ni drar i kvadraterna ändrar sig de andra kurvorna samt även **Current Compensator.** Se figur 2 !
Klickar ni på de rosa kvadraterna får ni information om polen.
Låt oss öppna fler fönster som också blir uppdaterade samtidigt t ex **Analysis->Response to Step command,** välj **Loops->Other Loop Responses.** Tryck på högra knappen och välj **Characteristics**.
Här finns det ett antal relevanta stegsvarsparametrar som ni känner igen. Välj t ex Peak response och lägg markören ovanför den blå punkten. Testa de andra om ni vill. Se figur 3 !
Den blåa kurvan är stegsvaret och den gröna är styrsignalen.
Jag la in settling time (insvängningstid) och rise time (stigtid).
Vi kanske vill veta exakt var vi befinner oss när vi flyttar polerna (rosa kvadrater) så öppna **View-> Closed Loop poles .**

Om längden på stegsvarsplottningen behöver förändras. Använd **Properties-> Limits** med högerknappen och ändra stopptiden.



Figur 3

a) Tag reda på för K=1, var polerna hamnar, vilka stabilitetsmarginaler vi har samt stigtid, översväng och insvängningstid (2%) !

b) Upprepa ovanstående för K=3.1 !

c) Antag att vi vill ha en amplitudmarginal på 5 ggr. Vilken fasmarginal och vilka insvängningsparametrar har
vi då i stegsvaret. Ange även polernas lägen i detta fall.

Som ni ser är det samma som uppgift 5. Kontrollera era resultat.

d) Importera en PI-regulator ifrån Matlabs kommandofönster genom att först definiera den där som GR=tf([10 1],[10 0]); % K(s*Ti +1)/ s*Ti ; här: Ti =10 och K=1.
Titta på stigtid för det slutna systemets stegsvar. Försök halvera stigtiden .
Vad händer då med insvängningstid, peaktid, regulatorförstärkning, amplitud- och fasmarginal ?

e) Vad händer med bandbredden för det slutna systemet vid denna förändring ?

f) Importera t ex en PID-regulator

```
>> Gpid=tf([1 10 1],[10 0])        % Definiera en överföringsfunktion för en PID-regulator.

Transfer function:
s^2 + 10 s + 1
--------------
    10 s
```

% K(s^2*T_d*T_i + s*T_i +1)/ s*T_i ; här: T_i =10, T_d =0.1 och K=1.

APPENDIX

Nedan följer några exempel på körningar med macro-filer eller m-filer som vi kan kalla dem. Det är egentligen små program som körs genom att skriva deras namn uti kommandofönstret. Notera att för att kunna köra dessa m-filer så måste ni först skapa och lägga dem i någon katalog t ex d.\temp. För att matlab ska kunna hitta dem måste ni ange sökvägen dit med
addpath d:\temp   **,därefter räcker det med att ange m-filens namn.**

Rotort.m

```
% Detta är en m-fil som har till uppgift att rita upp rotorten för en karakteristisk ekvation.
% I vårt fall uppgift8: q=s^3+1.2s^2+1.2s+0.2+0.4*K
% Den karakteristiska ekvationen ser ju olika ut ifrån fall  till fall och måste anges av  användaren.
% Rotorten beskriver ju hur rötterna vandrar som funktion av någon parameter t ex förstärkningen. K.
% Vilken variation K ska ha bestäms av användaren. Givetvis behöver man inte enbart titta på K.
%
% Skapad av T.Munther 001020

K=[0:0.5:10];                   % Definierar vilka förstärkningar vi vill undersöka.
for i=1:length(K)
q=[ 1 1.2 1.2 (0.2+0.4*K(i)];      % Karakteristiska ekvationens koefficienter
p(:,i)=roots(q);                    % Löser ut de olika rötterna
end

plot(real(p),imag(p),'x'),grid
xlabel('real-axel'), ylabel('imaginär –axel')
title('Rotorten plottas  för uppgift 7')
```

Nedan följer förslag på en fil för att undersöka det slutna och det öppna systemets Bodediagram, d v s amplitud- och faskurva ritas ut för uppgift 5.

Bode1.m

```
% Detta är en m-fil som plottar amplitudkurva och faskurva för det öppna och slutna systemet i uppgift 5.
%
% Skapad 001020 av T.Munther

talj=[0.4];namn=[1 1.2 1.2 0.2];              % Definierar koefficienterna för täljare respektive nämnare.
sys=tf(talj,namn);                  % Skapar  kretsöverföringens överföringsfunktion.
Gtot=feedback(sys,1)                % Skapar  det slutna systemets överföringsfunktion.
bode(Gtot)                          % Plottar bodediagram för det slutna systemet.
title('slutna systemet uppgift 5')
pause                               % Väntar på respons ifrån användaren.
margin(sys)                         % Plottar bodediagram för det öppna systemet och anger
title('öppna systemet uppgift 5')   % det slutna systemets stabilitetsmarginaler.
```

Här följer en m-fil som plottar upp ett stegsvar samt ritar ut polernas och nolställenas placering I komplexa talplanet.

Stegsvar.m

% Detta är en m-fil som plottar stegsvaret för olika system. Systemen är definierade som olika polynom.
% Ett för täljaren och ett för nämnaren. Polynomen anges bara med sin koefficienter, där koefficienterna anges
% i ordningen högst potens först.

talj=[1];namn=[1 2 1]; % Definierar koefficienterna för täljar- och nämnarpolynom.
sys=**tf**(talj,namn); % Skapar kretsöverföringens överföringsfunktion.
t=0:0.1:10; % Skapar tidsvektor, starttid, tidssteg, sluttid.
[y,t]=**step**(sys) % Beräknar stegsvaret.
plot(t,y) % plottar stegsvaret.
grid % Lägger till rutnät.
title('Stegsvar') % Anger titel.

pause % Väntar på tangentryckning.

pzmap(talj,namn) % Plottar var poler och nollställen finns.

För att hitta ett specifikt kommando, börja med att leta upp kategori genom **help**.
Du får då listat ett antal **HELP topics.**
Allmäna kommandon finns under **help general.**
Vid behov av mer reglertekniska funktioner skriv **help control.**
Vilka möjligheter ges av att använda **Matlab/Simulink**. Gå in under **demo** för att se olika exempel.

Nedan följer ett antal matlabfunktioner som kan vara bra att bekanta sig med inom ramen för kursen:
bode
c2dm
conv
d2cm
dstep

feedback
grid
hold
impulse
lsim
margin
pause
parallell
plot
pzmap
rlocfind
rlocus
roots
series
step
subplot
text
title