

Lineær analyse i SIMULINK

Av Finn Haugen (finn@techteach.no)
TechTeach (<http://techteach.no>)

20.12 2002

Innhold

1	Innledning	7
2	Kommandobasert linearisering av modeller	9
2.1	Beregning av arbeidspunkt	9
2.2	Utvikling av lineær modell	13
3	Lineær analyse med LTI Viewer	19
3.1	Analyse omkring arbeidspunkt	19
3.2	Utvikling av LTI-modell	20
4	Lineær analyse av tilbakekoplede systemer	23

Forord

Dette dokumentet supplerer boka **Lær SIMULINK trinn for trinn** [1]. Dokumentet beskriver metoder som er tilgjengelige i SIMULINK for analyse av en lokal, lineær modell omkring et spesifisert arbeidspunkt for det opprinnelige systemet, som kan være lineært eller ulineært. Analysemetodene som er tilgjengelige, er bl.a. sprangrespons, frekvensrespons og pol-nullpunkt-beregninger.

MATLAB-skriptene (m-filene) og SIMULINK-modellene (mdl-filene) som er benyttet i dette dokumentet er fritt tilgjengelig fra hjemmesiden for <http://techteach.no>.

Skien, desember 2002

Finn Haugen

Kapittel 1

Innledning

Det fins mange analyse- og designmetoder for *lineære* systemer (modeller). Metodene baseres gjerne på

- lineær tilstandsrommodell
- transferfunksjon
- poler og nullpunkter
- egenverdier
- frekvensrespons

SIMULINK har to muligheter for lineær analyse:

- SIMULINK har funksjonene `linmod` og `dlinmod` for å finne disse størrelsene fra en gitt blokkdiagrammodell — både for lineære modeller og for ulineære modeller. Dersom modellen er ulineær, vil den lineære modellen utgjøre en lokal modell omkring et arbeidspunkt. SIMULINK har funksjonen `trim` for å finne dette arbeidspunktet, som må bestemmes før lineariseringen utføres.
- Dersom Control System Toolbox er installert, kan en få utført linearisering og forskjellige slags lineære analyser vha. menyvalget Tools / Linear analysis i SIMULINK, som bruker LTI Viewer til visning av f.eks. poler og nullpunkter, frekvensrespons og sprangrespons. En kan også få generert en modell i form av et LTI-objekt.

Begge de to ovenfor nevnte mulighetene for lineær analyse beskrives i dette dokumentet.

Kapittel 2

Kommandobasert linearisering av modeller

2.1 Beregning av arbeidspunkt

`trim` beregner verdien av variablene for et system som befinner seg i et spesifisert arbeidspunkt, som typisk vil være et likevektspunkt der alle variablene har konstante verdier, dvs. et statisk arbeidspunkt. `trim` kan være første trinn i utvikling av en lokal linear modell omkring et statisk arbeidspunkt.

Før du bruker `trim`, må du angi hva som er systemets innganger og utganger. Det gjøres vha. `Inport`-blokker hhv. `Outport`-blokker som fins i hhv. `Sources`- hhv. `Sinks`-biblioteket i SIMULINK.

Syntaksen for en typisk anvendelse av `trim` er

```
[xop,uop,yop] = trim('sys',xop0,uop0,yop0,ix,iu,iy)
```

(Du kan utføre kommandoen `help trim` for å se videre muligheter, for eksempel hvordan du kan beregne et ikke-statisk arbeidspunkt, der tilstandsvariablenes deriverte er forskjellig fra null.) I uttrykket ovenfor representerer `sys` SIMULINK-modellen med navn `sys.mdl`. Videre er `xop`, `uop` og `yop` returnerte verdier for hhv. tilstanden, inngangen og utgangen i arbeidspunktet. `xop0`, `uop0` og `yop0` er enten anslåtte verdier eller faste (ikke justerbare) verdier for `xop`, `uop` og `yop`, slik: Hvis du for eksempel

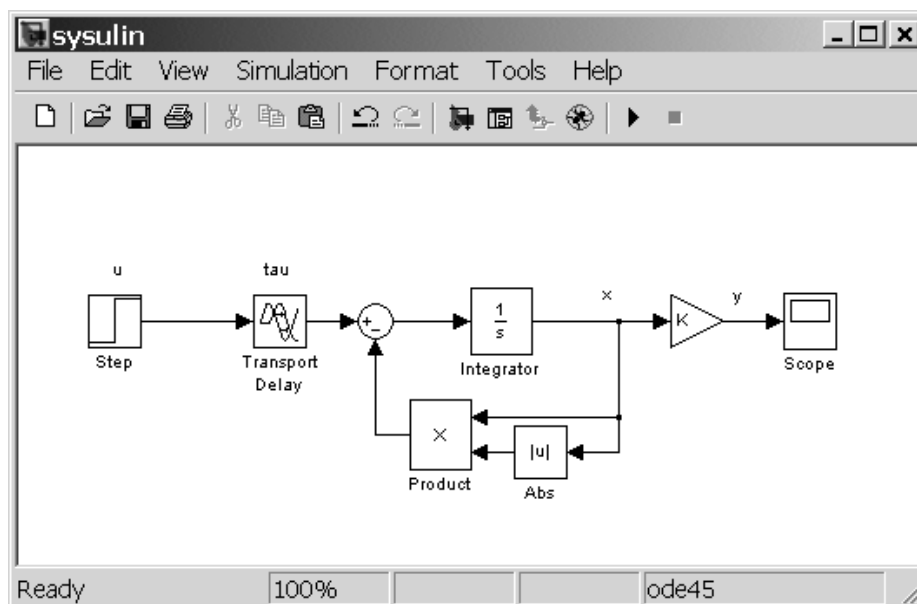
setter `iu = []` (tom matrise), blir `uop0` betraktet av `trim` til å være bare en *omtrentlig* verdi. Hvis du i stedet spesifiserer la oss si `iu = [1;2]`, vil de to første elementene i den spesifiserte vektoren `uop0` betraktes som faste verdier. Tilsvarende gjelder for `x` og `y`.

La oss se på et *eksempel*. Gitt følgende ulineære modell:

$$\dot{x}(t) = -x|x| + u(t - \tau) \quad \text{der } \tau = 1 \quad (2.1)$$

$$y = Kx \quad \text{der } K = 3 \quad (2.2)$$

Initialtilstanden er $x_0 = 0$. Figur 2.1 viser et SIMULINK-blokkdiagram for denne modellen. Vi skal nå først for hånd beregne verdien, x_s , av tilstanden



Figur 2.1: SIMULINK-blokkdiagram for modellen $\dot{x}(t) = -x|x| + u(t - \tau)$, $y = Kx$.

x i det stasjonære arbeidspunktet svarende til at u er en konstant med verdi

$$U = 4 \quad (2.3)$$

og deretter skal vi bruke `trim`-funksjonen til å finne denne tilstandsverdien (`trim` skal da anvendes på SIMULINK-modellen). Vi antar at x faktisk vil få en slik stasjonær konstant verdi¹. Den tidsderiverte av x_s er null, og

¹Vi antar med andre ord at det arbeidspunktet er asymptotisk stabilt.

betydningen av tidsforsinkelsen faller bort, så (2.1) blir

$$0 = -x_s(t) |x_s(t)| + U \quad (2.4)$$

som gir (som eneste reelle løsning)

$$x_s = \sqrt{U} = 2 \quad (2.5)$$

som gir

$$y_s = Kx_s = 3 \cdot 2 = 6 \quad (2.6)$$

La oss se om en simulering bekrefter at $y_s = 6$ når $U = 4$.
MATLAB-skriptet `skript_sim_sysulin.m` kjører simuleringen:

`skript_sim_sysulin.m`

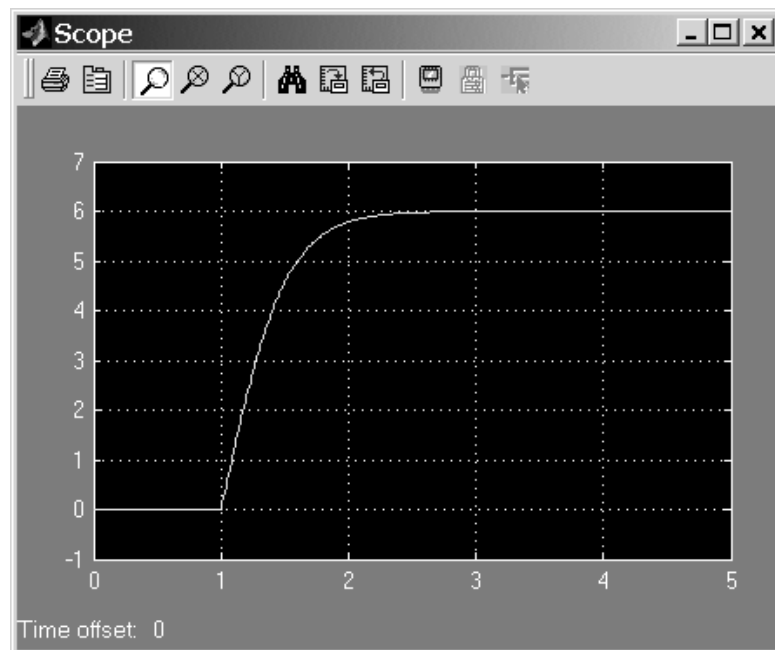
```
tu=0; %Sprangtidspunkt for u
u0=0; %Verdi av u før spranget
u1=4; %Verdi av u etter spranget
tau=2; %Tidsforsinkelse
K=3;
x0=0; %Initialverdi av x

%Setter simuleringsparametre:
tidsskritt=0.01;
options=simset('solver','ode5','fixedstep',tidsskritt);
tstopp=5;

%Simulerer systemet sysulin:
sim('sysulin',tstopp,options)
```

Figur 2.2 viser responsen i y etter et sprang i u fra 0 til 4 ved tidspunktet $t = 0$. Vi ser at den stasjonære responsen ganske riktig er $y_s = 6$.

Vi skal nå prøve å finne arbeidspunktet som vi fant ovenfor (med håndregning og med simulering) ved å bruke SIMULINKs `trim`-funksjon. Vi definerer innganger og utganger vha. `Inport`-blokker hhv. `Outport`-blokker (disse fins SIMULINKs Source- hhv. Sinks-bibliotek). Vi erstatter signalkilder med `Inport`-blokker (`Inport`-blokken fins i) og hekter `Outport`-blokker på de signallinjene som utgjør systemets utganger. Det viser seg at `trim`-funksjonen ikke fungerer tilfredsstillende, dvs. at arbeidspunktet ikke beregnes korrekt av `trim`-funksjonen, dersom det er en



Figur 2.2: Simulering av det ulineære systemet (sysulin.mdl)

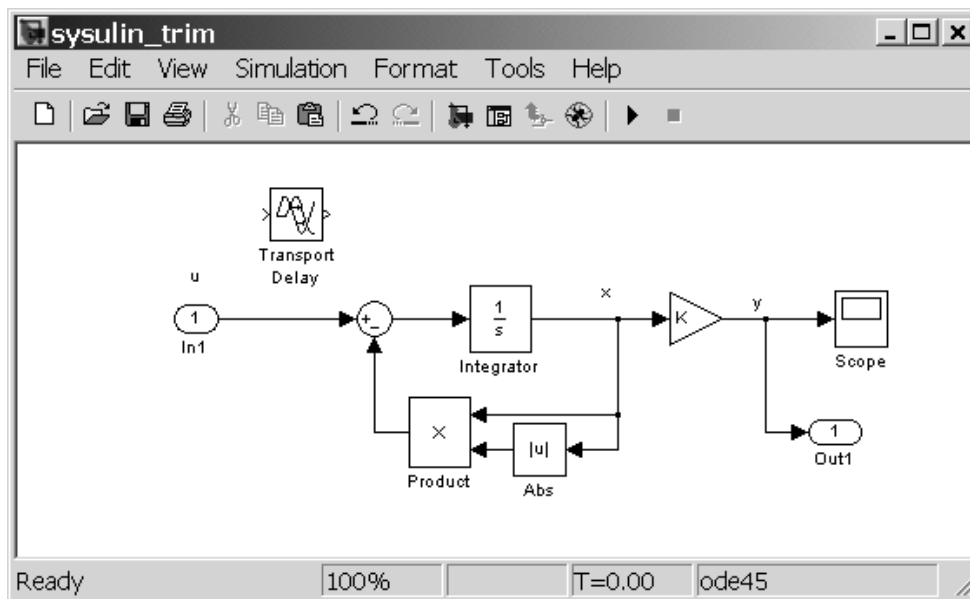
Transport Delay-blokk i systemet. Jeg har derfor fjernet denne fra blokkdiagrammet for vårt system (det gjør ikke så mye — tidsforsinkelser har jo ingen betydning i statiske arbeidspunkter). Figur 2.3 viser blokkdiagrammet med Inport- og Outport-blokk innsatt (og Transport Delay-blokken fjernet eller satt utenfor systemet).

Følgende MATLAB-skript beregner arbeidspunktet:

skript_trim_sysulin.m:

```
xop0=[3]; %Gjettet verdi for x
uop0=[4]; %Spesifisert verdi for u
yop0=[5]; %Gjettet verdi for y
ix=[]; %x er ikke fast (men gjettet)
iu=1; %u er spesifisert
iy=[]; %y er ikke fast (men gjettet)
[xop,uop,yop] = trim('sysulin_trim',xop0,uop0,yop0,ix,iu,iy)
```

MATLAB svarer (ganske riktig)



Figur 2.3: Systemets blokkdiagram med Inport- og Outport-blokk innsatt (og Transport Delay-blokken fjernet eller satt utenfor systemet).

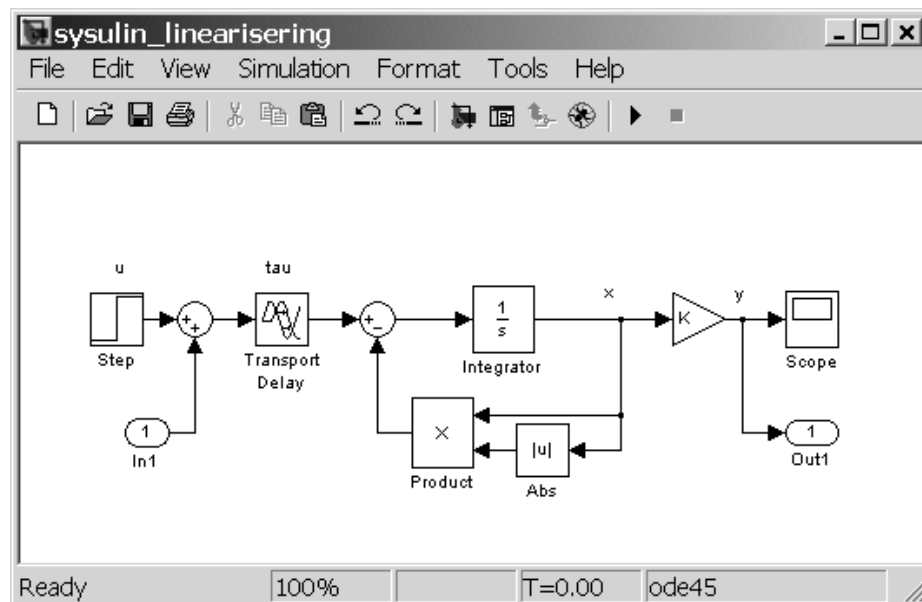
```
xop =
2.0000
uop =
4
yop =
6.0000
```

2.2 Utvikling av lineær modell

SIMULINK har funksjonene `linmod` for å utvikle en tidskontinuerlig lokal linear modell og `dlinmod` for å utvikle en tidsdiskret lokal lineær modell for et ulineært system (blokkdiagram) omkring et spesifisert arbeidspunkt. Modellen som `linmod` og `dlinmod` gir, er tilstandsrommodeller i form av tilstandsrommodellens standard systemmatriser (4 stk. matriser). `linmod` og `dlinmod` kan brukes også til utvikle tilstandsrommodeller for SIMULINK-modeller som i utgangspunktet er lineære men ikke på tilstandsromform. Fra den lineære tilstandsrommodellen som `linmod` eller `dlinmod` gir, kan vi så finne f.eks. en transferfunksjonsmodell i form av et

LTI-objekt vha. `tf`-funksjonen og beregne poler med `pzmap` og foreta frekvensresponsanalyse med f.eks. `bode` eller `nichols` eller `nyquist` (alle disse funksjonene er tilgjengelig i Control System Toolbox).

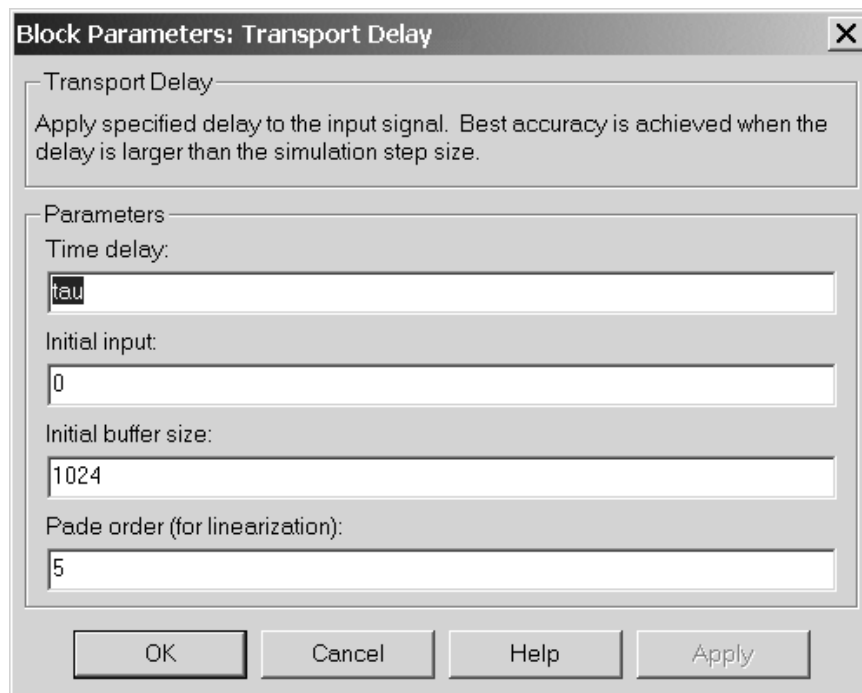
Vi skal nå finne en lokal lineær modell for systemet `sysulin.mdl`, se figur 2.1, omkring arbeidspunktet gitt ved $u = 4$, $x = 2$ og $y = 6$, som vi fant i kap. 2.1 bl.a. vha. `trim`-funksjonen. Den lineære modellen skal finnes for systemet mellom inngang u og utgang y . Vi må sette inn en **Inport**-blokk og en **Outport**-blokk for å definere det lineære systemets inngang og utgang, se figur 2.4.



Figur 2.4: En Inport-blokk og en Outport-blokk settes inn for å definere det lineære systemets inngang og utgang

Systemet inneholder en tidsforsinkelse. Den lineære modellen skal da (selvsagt) inkludere en tidsforsinkelse. `linmod` vil da benytte en Padé-approksimasjon for tidsforsinkelsen. Approksimasjonens orden kan velges i parametervinduet for **Transport Delay**-blokka, se figur 2.5. Vi velger orden 5 i vårt eksempel. MATLAB-skriptet `skript_linearisering_sysulin.m` vist nedenfor viser hvordan lineariseringen kan utføres. (Det er tilstrekkelig å angi arbeidspunktverdiene for x og u — altså ikke for y .)

skript_linearisering_sysulin.m



Figur 2.5: Valg av Pad é-approksimasjonens orden for tidsforsinkelsen gjøres i parametervinduet for Transport Delay-blokka

```
xop=2;
uop=4;
[A,B,C,D] = linmod('sysulin_linearisering',xop,uop)
sysulin_lti=ss(A,B,C,D); %Lager LTI-objekt
sysulin_tf=tf(sysulin_lti)%Lager transferfunksjon
```

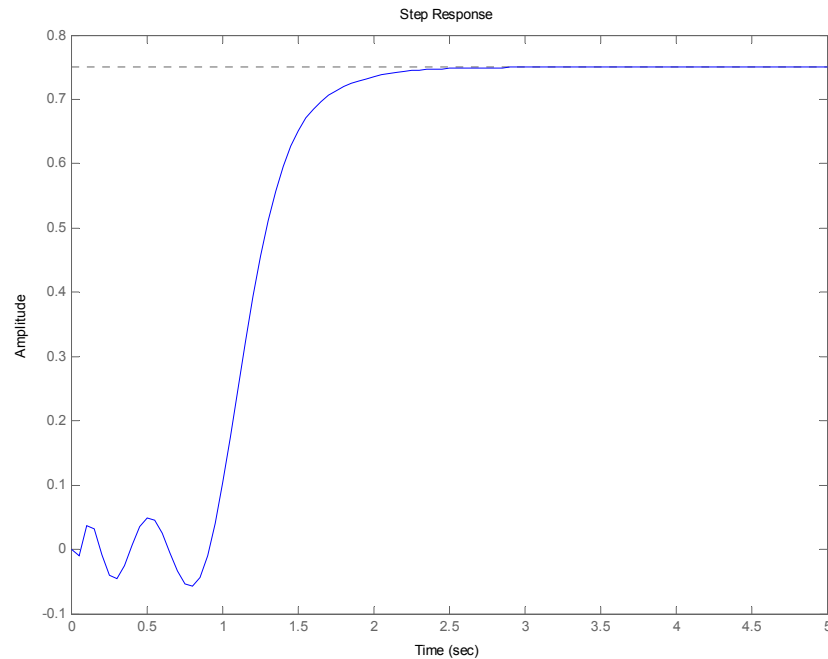
Den resulterende (6. ordens) transferfunksjonen er som vist nedenfor:

```
Transfer function:
-3 s^5 + 90 s^4 - 1260 s^3 + 1.008e004 s^2 - 4.536e004 s + 9.072e004
-----
s^6 + 34 s^5 + 540 s^4 + 5040 s^3 + 2.856e004 s^2 + 9.072e004 s + 1.21e005
```

Figur 2.6 viser sprangresponsen for denne transferfunksjonen. Sprangresponsen er generert med

```
step(sysulin_tf,5)
```

(Oscillasjonene stammer fra Padé-approksimasjonen.)



Figur 2.6: Sprangresponsen for den lineære modellen

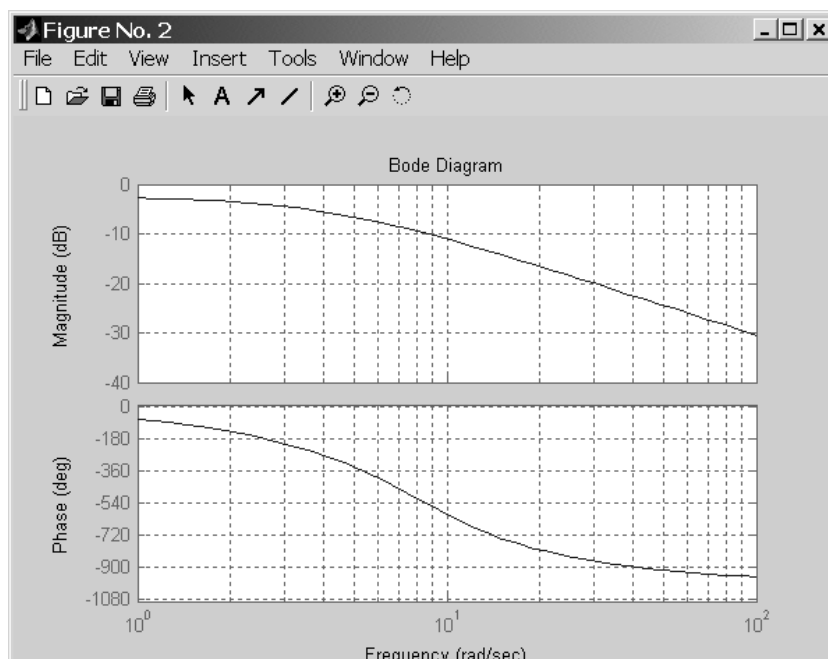
Og vi kan f.eks. tegne Bode-plott:

```
bode(sysulin_tf)
```

Bode-plottet er vist i figur 2.7.

Betydningen av tidsforsinkelsen (Transport Delay-blokka)

Ovenfor var utgangspunktet for lineariseringen en ulineær modell som inneholder en tidsforsinkelse. Den tilsvarende lineære modellen ble da en modell av orden 6, selv om det opprinnelige systemet kun inneholder én tilstandsvariabel, nemlig x (de 5 øvrige tilstandsvariablene stammer fra padé-approksimasjonen). Det viser seg at dersom det opprinnelige systemet (i vårt eksempel) *ikke* inneholder noen tidsforsinkelse, gir `linmod` en lineær modell som blir av 1. orden. Mer spesifikt: `linmod` sammen med funksjonene `ss` og `tf`, jf. skriptet `skript_linearing_sysulin.m` vist ovenfor, gir



Figur 2.7: Bode-plott for den lineære modellen

følgende 1. ordens modell omkring arbeidspunktet ($u = 4$, $x = 2$):

$$\frac{dy(s)}{du(s)} = \frac{3}{s + 4} \quad (2.7)$$

Dette er faktisk precis den transferfunksjonen som vi vil finne ved å linearisere (2.1), (2.2) analytisk omkring arbeidspunktet $u = 4$, $x = 2$. Den analytisk lineariserte modellen blir nemlig

$$d\dot{x}(t) = -2|x_a|dx + du \quad (2.8)$$

$$dy = Kdx \quad (2.9)$$

der x_a er verdien av x i arbeidspunktet. Fra denne modellen, med $x_a = 2$, får vi (2.7).

Kapittel 3

Lineær analyse med LTI Viewer

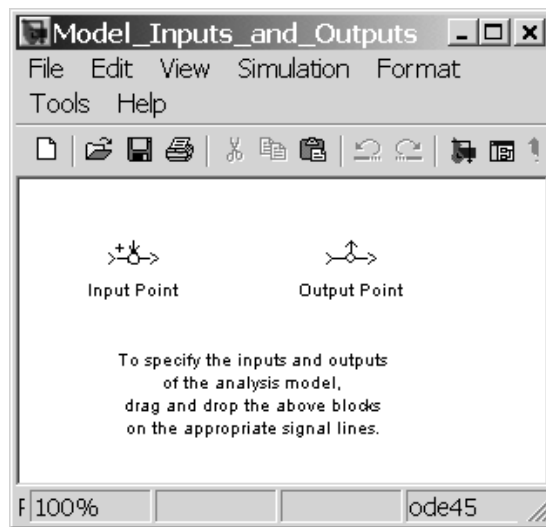
3.1 Analyse omkring arbeidspunkt

Som alternativ til bruk av `trim` og `linmod` beskrevet ovenfor, kan vi utføre linearisering med menyvalg og knappevalg via menyen **Linear Analysis** i SIMULINK-vinduet. Merk at dette ikke gir noen eksplisitt lineær modell, men du får utført lineær analyse, som beregning av poler, plotting av frekvensrespons og plotting av sprangrespons omkring et spesifisert arbeidspunkt.

Menyvalget **Linear Analysis** åpner i første omgang vinduet **Model_Inputs_and_Outputs** vist i figur 3.1. Blokkene der, som er **Input Point-blokka** og **Output Point-blokka**, brukes til å definere innganger og utganger for det lineariserte systemet. Du trekker ganske enkelt disse blokkene inn på de aktuelle signallinjene i systemets blokkdiagram, og de faller automatisk på plass på signallinjene. Figur 3.2 viser det resulterende blokkdiagrammet for vårt eksempel.

Når **Input Point-blokka** og **Output Point-blokka** er på plass i systemets blokkdiagram, bør du definere det aktuelle arbeidspunktet som den lineære analysen skal foretas ut fra. Arbeidspunktet defineres med menyvalget **Simulink / Set Operating Point** i vinduet **LTI Viewer**, se figur 3.3, som åpnes automatisk etter menyvalget **Linear Analysis** i SIMULINK-vinduet.

Når arbeidspunktet er valgt, kan du angi hva slags lineær analyse (bl.a. impulse, step, bode, nichols, pole-zero) som skal utføres og vises. Dette



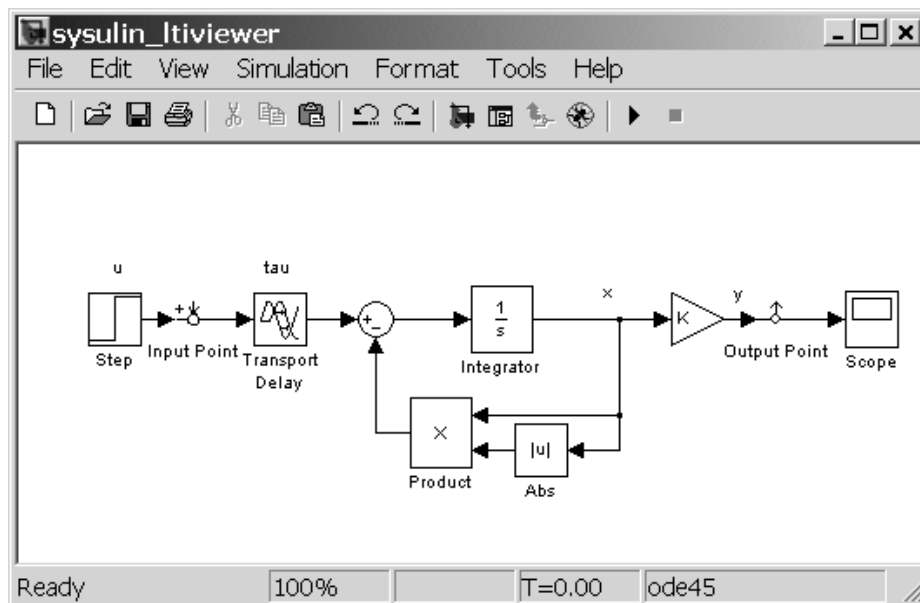
Figur 3.1: Menyvalget Linear Analysis i SIMULINK-vinduet åpner i første omgang vinduet Model_Inputs_and_Outputs. Blokkene der brukes til å definere innganger og utganger for det lineariserte systemet.

gjøres via menyvalget Edit / Plot Configurations i LTI Viewer-vinduet, hvilket åpner dialogvinduet Plot Configurations vist i figur 3.4.

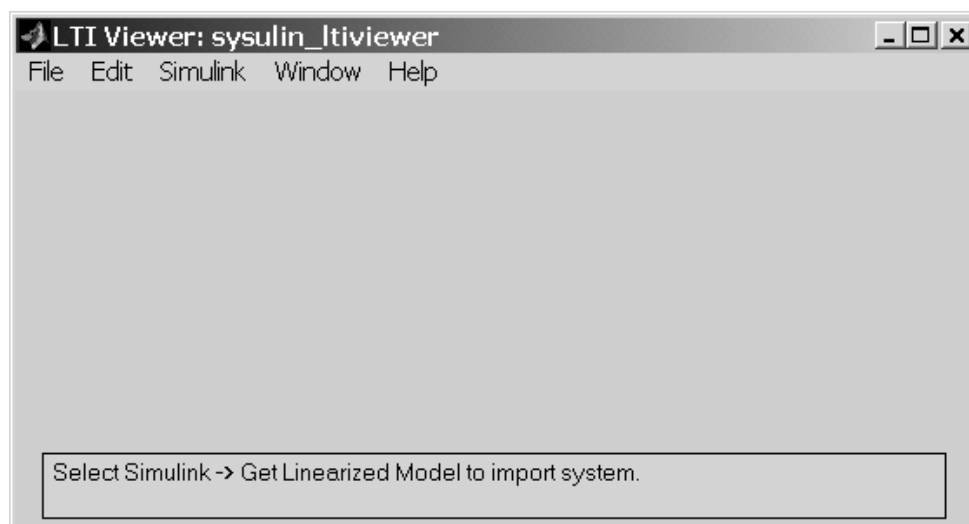
Når du har utført ovenstående (definert inngang og utgang, angitt arbeidspunkt og bestemt type lineær analyse), er du endelig klar for å utføre den lineære analysen. Dette gjøres med menyvalget Simulink / Get Linearized Model i LTI Viewer-vinduet. Figur 3.5 viser sprangresponsen for den lineære, lokale modellen omkring arbeidspunktet tilsvarende $x = 2$. (Sprangresponsen er lik den som ble funnet etter `trim` og `linmod` i kap. 2, se figur 2.6.)

3.2 Utvikling av LTI-modell

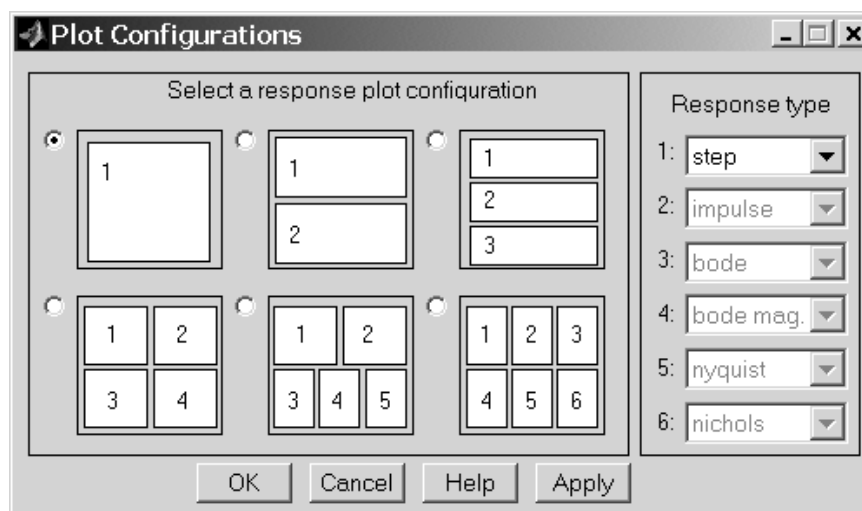
Du får generert en LTI-modell i form av en tilstandsrommodell med menyvalget File / Export i LTI Viewer-vinduet. Modellen er tilgjengelig i MATLABs arbeidsrom.



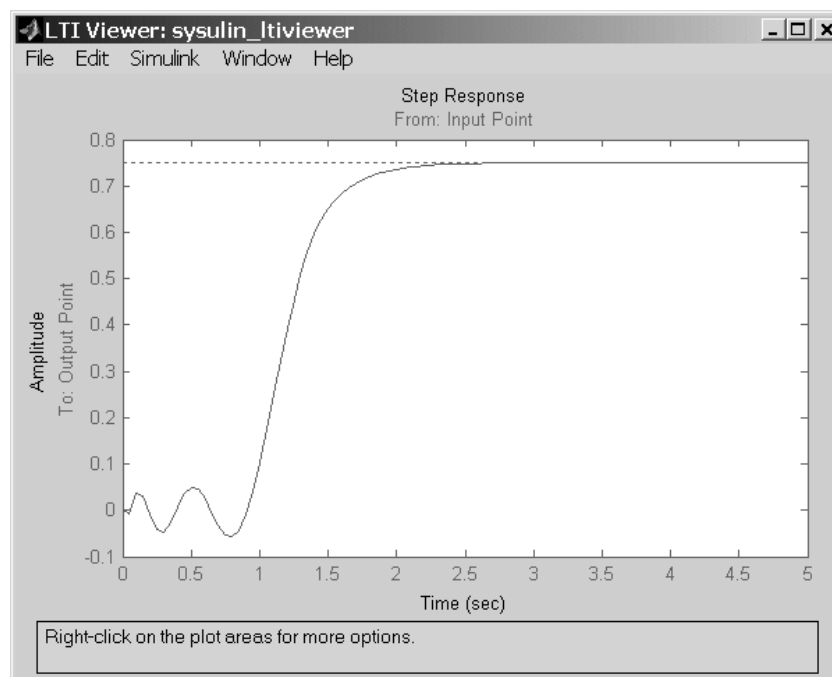
Figur 3.2: Det resulterende blokkdiagrammet etter at Input Point- og Output Point-blokker er lagt inn p å de aktuelle signallinjene i systemets blokkdiagram



Figur 3.3: Vinduet LTI Viewer åpnes automatisk etter menyvalget Linear Analysis i SIMULINK-vinduet.



Figur 3.4: Dialogvinduet Plot Configurations som åpnes via menyvalget Edit / Plot Configurations i LTI Viewer-vinduet.



Figur 3.5: Sprangresponsen for den lokale lineære modellen funnet i arbeidspunktet $x = 2$

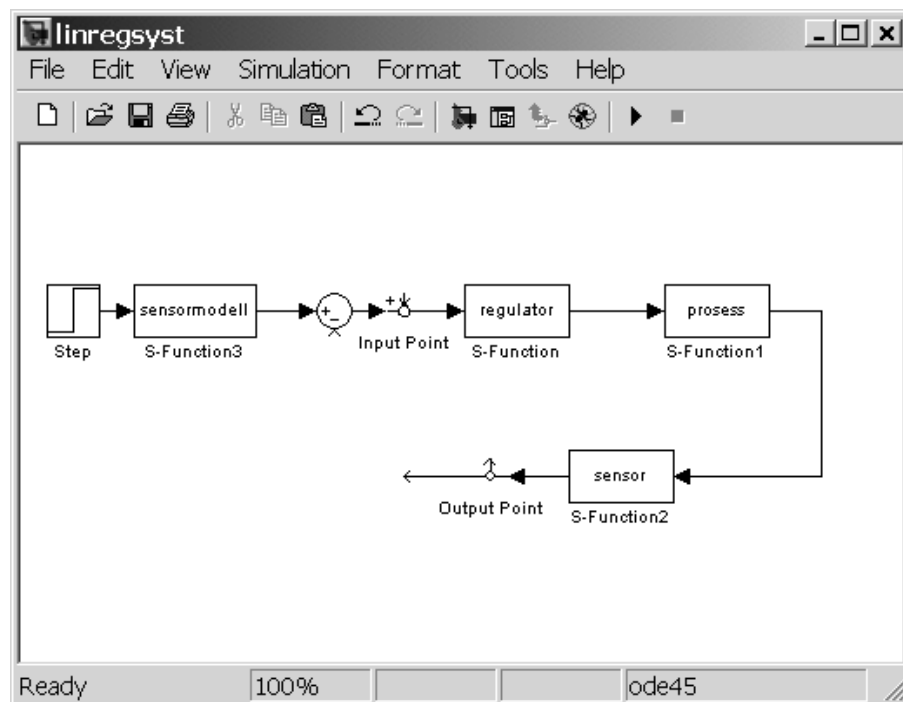
Kapittel 4

Lineær analyse av tilbakekoblede systemer

Dersom du skal analysere et tilbakekoplet system, f.eks. et reguleringsystem, ut fra en lineær åpen-sløyfe-modell for systemet¹, må du huske å *åpne sløyfa* før du setter inn blokker som definerer inngang og utgang. Dette er illustrert i figur 4.1.

På den annen side: Dersom du skal utføre en analyse av det lukkede (tilbakekoblede) system ut fra systemets lukket-sløyfe-modell, f.eks. følgeforholdet eller sensitivitetsfunksjonen, skal du selvsagt beholde sløyfa lukket under analysen.

¹En slik analyse kan gjelde f.eks. frekvensanalyse av et reguleringsystems sløyfetransferfunksjon.



Figur 4.1: Analyse av et tilbakekoplet systems åpen-sløyfe-modell må foretas når sløyfa er *åpen*.

Bibliografi

- [1] Finn Haugen: **Lær SIMULINK trinn for trinn**, TechTeach og Tapir Akademisk Forlag, 2003