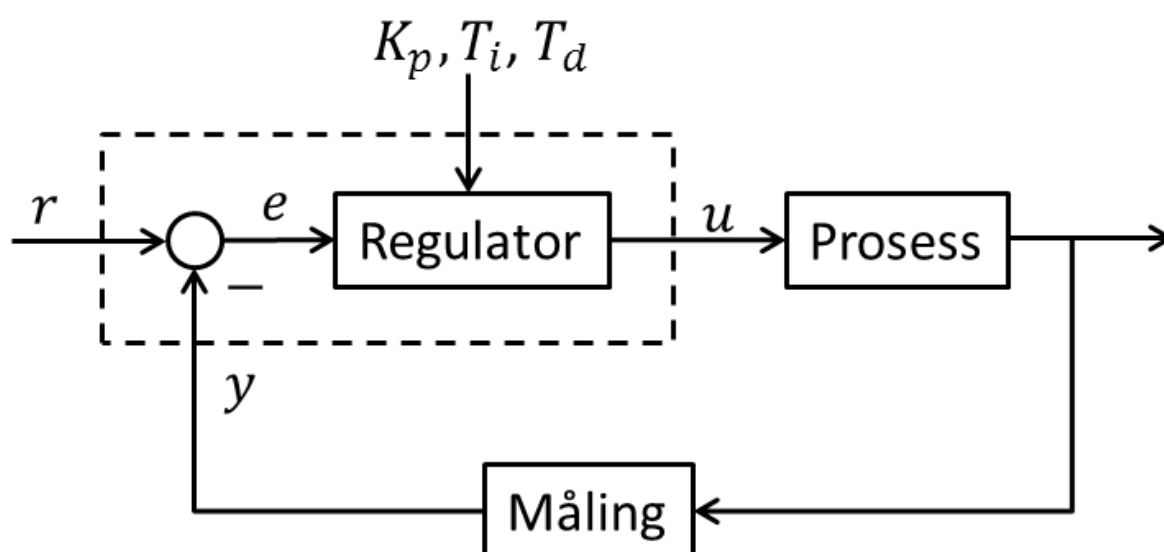


Reguleringsteknikk med LabVIEW og MathScript eksempler

Hans-Petter Halvorsen, 2016.10.26



Innholdsfortegnelse

Innholdsfortegnelse	ii
1 Introduksjon	8
1.1 Regulator	9
1.1.1 PID regulering	9
1.2 Eksempel.....	10
1.3 I/O (DAQ) enhet.....	11
1.4 Dataverktøy	12
2 Hva er LabVIEW?	13
2.1 Frontpanel og Blokkdiagram.....	13
2.2 Controls Palett og Functions Palett	14
2.3 Controls og Indicators.....	14
2.4 Wiring: Lag logikken.....	15
3 Hva er MathScript?.....	18
3.1 Innledning	18
3.2 Plotting	19
4 Introduksjon til Transferfunksjoner.....	20
4.1 Transferfunksjoner	20
4.2 MathScript	20
4.3 Differensiallikninger.....	22
4.4 Sprangrespons	24
4.5 Sluttverditeoremet	27

5	Integrator	29
5.1	Sprangrespons	29
6	1.ordens systemer	34
6.1	Sprangrespons	34
6.2	MathScript	35
6.3	1.ordens system med tidsforsinkelse	36
6.3.1	MathScript	36
6.3.2	Eksempler	37
7	2.ordens systemer	43
7.1	Responstid	43
7.2	MathScript	43
7.3	Sprangrespons og stabilitet	44
7.4	Poler	44
7.5	Spesialtilfelle	45
7.6	Eksempel: Masse-Fjær-demper system	45
8	Blokkdiagrammer	48
8.1	Serie	49
8.2	Parallell	49
8.3	Tilbakekobling (Feedback)	50
8.4	1.ordens system	50
8.4.1	1.ordens prosess med dødtid	51
8.5	2.ordens system	51
9	Tilstandsrom- modeller	56
10	Tidsforsinkelse og Pade' Approksimasjon	59

10.1	Pade' Approsimasjon.....	60
10.2	MathScript.....	60
10.2.1	Metode 1.....	61
10.2.2	Metode 2.....	61
10.2.3	Metode 3.....	62
11	Stabilitets -analyse	63
11.1	Impulsrespons	63
11.1.1	Asymptotisk stabilt system.....	63
11.1.2	Marginalt stabilt system.....	64
11.1.3	Ustabilt system.....	64
11.1.4	MathScript.....	64
11.2	Poler	66
11.2.1	Asymptotisk stabilt system.....	66
11.2.2	Marginalt stabilt system.....	66
11.2.3	Ustabilt system.....	67
11.2.4	MathScript.....	67
11.3	Tilbakekoblede systemer.....	71
11.3.1	Sløyfetransferfunksjonen	71
11.3.2	Følgeforholdet.....	71
11.3.3	Sensitivitetsfunksjonen/Avviksforholdet.....	72
11.3.4	MathScript.....	72
12	PID regulering.....	74
12.1	PI regulator.....	75
12.2	Sprangrespons.....	76
13	Frekvensrespons.....	77

13.1	Bodediagram	78
13.1.1	MathScript.....	78
13.2	Frekvensresponsen fra transferfunksjonen	80
13.2.1	Komplekse tall	81
13.2.2	Eksempler	84
13.2.3	Knekkfrekvenser	85
13.2.4	MathScript.....	86
13.3	Praktiske forsøk.....	88
13.4	Frekvensrespons for standardfunksjoner	94
13.4.1	Forsterker	94
13.4.2	Integrator	95
13.4.3	Derivator.....	96
13.4.4	1.ordens system	97
13.4.5	2.ordens system	98
13.4.6	Nullpunktsledd	99
13.4.7	Tidsforsinkelse	99
13.5	Filtere	100
13.5.1	Lavpassfilter.....	101
13.5.2	Høypassfilter.....	101
13.5.3	Båndstopp	102
13.5.4	Båndpass	102
13.5.5	MathScript.....	102
13.6	Definisjoner	105
13.6.1	Periode – T.....	105
13.6.2	Amplitude - A.....	105
13.6.3	Frekvens - f	105

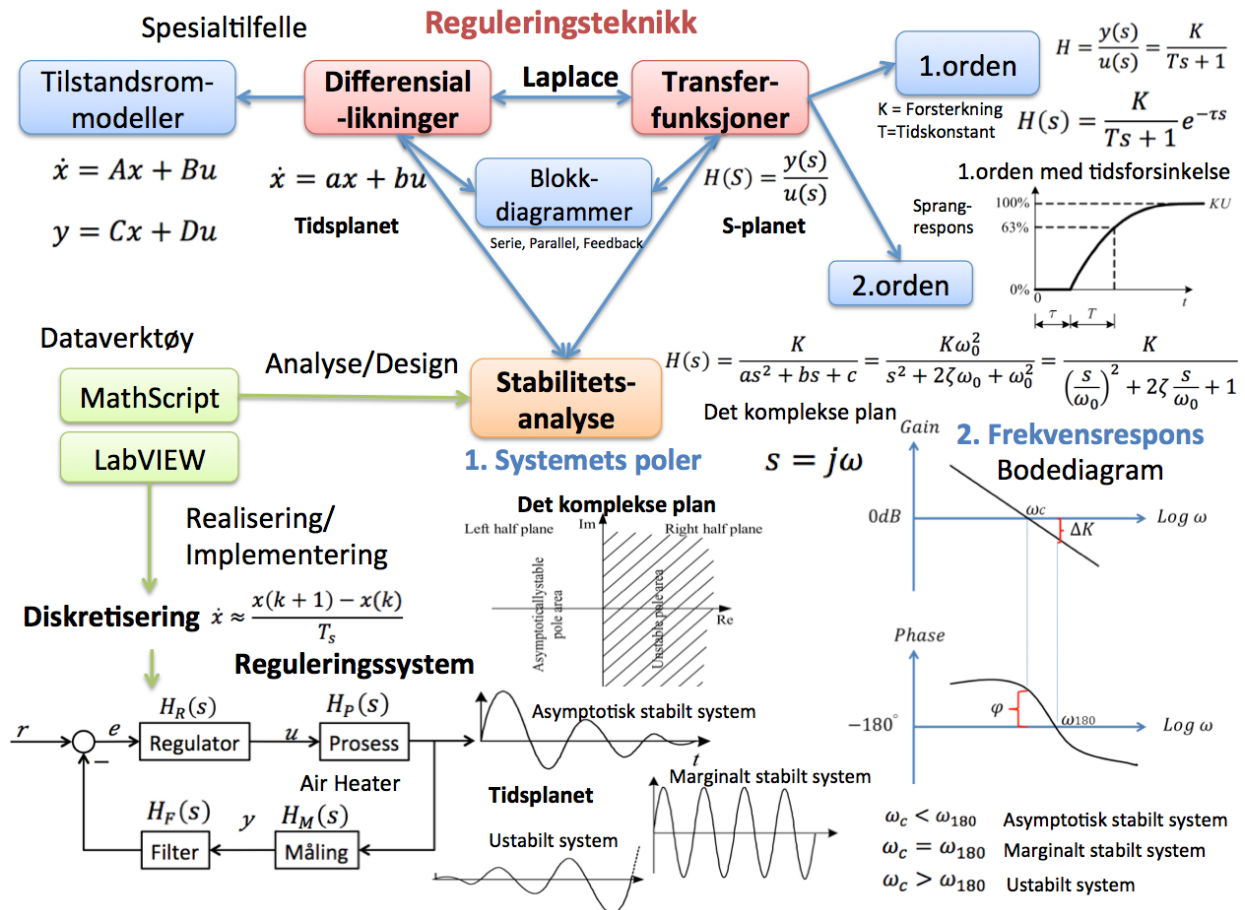
14	Frekvensrespons- analyse	107
14.1	Innledning	107
14.2	Følgeegenskaper	108
14.3	Båndbredde.....	109
14.4	MathScript.....	111
15	Stabilitetsanalyse i Frekvensplanet	113
15.1	Stabilitetsmarginer	113
15.2	Stabilitetsanalyse	114
15.2.1	MathScript.....	114
16	MathScript Tips & Triks	118
16.1	Innledning	118
16.2	Plotting.....	119
16.3	Tips & Triks	120
17	MathScript- funksjoner	124
18	DAQ i LabVIEW	126
18.1	USB-6008.....	126
18.2	DAQ	126
18.3	Bruk av NI USB-6008 i LabVIEW	127
18.3.1	Analog Input	128
18.3.2	Analog Output	132
19	Simulering og regulering i LabVIEW	137
19.1	Simulation Subsystem	141
19.2	PID	142

20	Diskretisering.....	144
20.1	Innledning	144
20.2	Diskretisering i MathScript.....	146
20.2.1	Vanntank	148
20.3	Diskretisering i LabVIEW	151

1 Introduksjon

Kybernetikk (eller reguleringsteknikk) er læren om styremekanismer og styringsproblemer. I dag blir begrepet benyttet for å beskrive styring og regulering i maskiner og levende systemer. Kybernetikk er referanseramme for moderne datavitenskap.

Nedenfor ser vi en skisse av temaer som gjennomgås i dette dokumentet.



Styring og regulering skjer ofte ved hjelp av tilbakekobling. Ved å måle eller estimere systemets nåtilstand kan det bli regnet ut hvilke endringer som er nødvendig for at systemet skal holde seg stabilt rundt en referanseverdi. Ett eksempel på dette kan være et skip man ønsker skal ligge i ro (for eksempel ved siden av en oljeinstallasjon for lossing av olje) selv om det er både vind og bølger. Man måler da posisjon og eventuelt vind, bølger og strøm, for å så beregne hva slags pådrag som må gis til motorene for at skipet skal ligge i ro i forhold til oljeplattformen.

Kybernetikk/Reguleringsteknikk beskriver hvordan man kan regulere og stabilisere et dynamisk system.

Et dynamisk system kan reguleres eller styres på flere måter. Et system som involverer en person som styrer en maskin, f.eks. en person som kjører bil, kalles manuell styring.

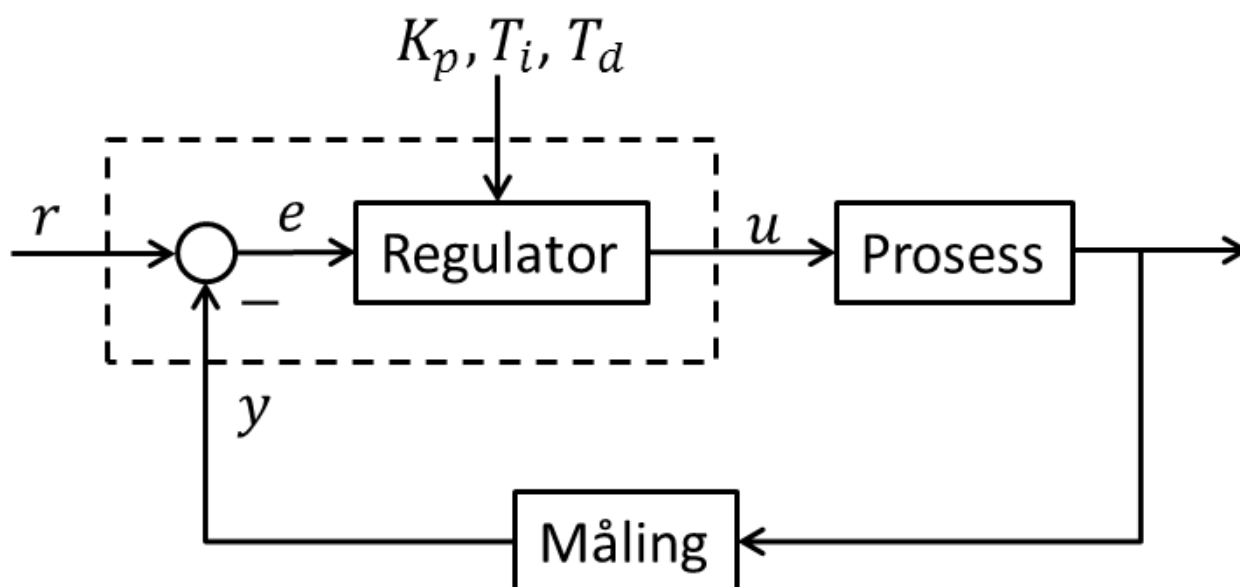
Regulering av vannivå i en tank, regulering av temperatur, eller regulering av gjennomstrømning, regulering av trykk, fart, osv.

Det er ubegrenset hva du kan regulere, og alle disse kan bli regulert på samme måte; med en **regulator**.

Dette dokumentet har fokus på praktiske eksempler ved bruk av LabVIEW og MathScript. Det er veldig viktig å kunne beherske slike verktøy, da man ikke bruker "penn og papir" i industrien. På den ene siden vil praktiske eksempler vha disse verktøyene hjelpe deg til å forstå og kunne anvende den reguleringsstekniske teorien i praksis, mens på den andre siden for å få full glede av disse verktøyene må man selvsagt ha den grunnleggende reguleringsstekniske forståelsen på plass.

1.1 Regulator

Oppgaven til en regulator er å endre pådraget i forhold til måleverdien fra prosessen, det gjør den med noen parameterer som kalles P, I og D (PID regulator). Disse variablene varierer fra prosess til prosess. Formålet til disse variablene er å skape den optimale reguleringsalgoritmen til en viss prosess, for å få en mest mulig stabil reguleringen, dvs at utgangen y følger referansen r best mulig.



1.1.1 PID regulering

En PID regulator er gitt ved:

$$u(t) = K_p e + \frac{K_p}{T_i} \int_0^t e d\tau + K_p T_d \dot{e}$$

Der e er avviket mellom referansen r og utgangen y ($e = r - y$), mens u er pådraget.

P-ledd (Proporsjonal):

$$u_p(t) = K_p e$$

Der K_p er (proporsjonal) forsterkningen

I-ledd (Integral):

$$u_i(t) = \frac{K_p}{T_i} \int_0^t e d\tau$$

Der T_i er integraltiden

→ I-leddet sørger for at regulatoren gir null avvik (stasjonært) (Statisk ytelse)

D-ledd (Deriverte):

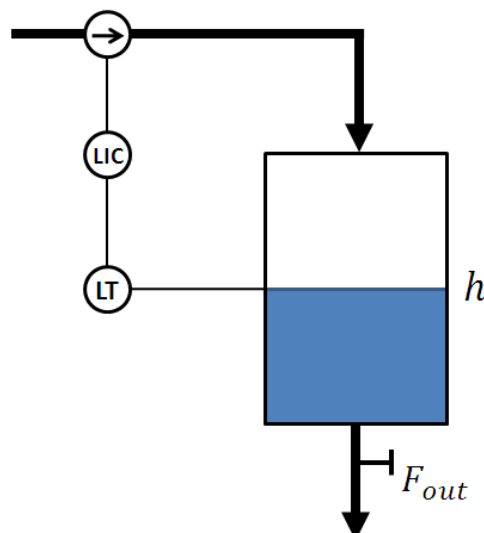
$$u_D(t) = K_p T_d \dot{e}$$

Der T_d er derivattiden

→ D-leddet sørger for at regulatoren reagerer raskt (Dynamisk ytelse)

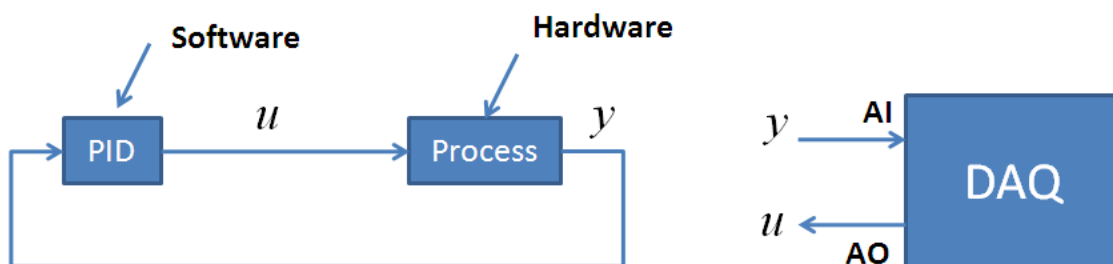
1.2 Eksempel

Nedenfor ser vi reguleringsystemet for en vanntank. Formålet er å regulere nivået i tanken (h) på et gitt nivå (referanseverdien). Dette gjøres i dette tilfellet ved en PID regulator som styrer en pumpe på tankens innløp.



1.3 I/O (DAQ) enhet

Vanligvis blir regulatoren implementert i software, dvs. vha. en datamaskin eller liknende, mens prosessen er en fysisk enhet (f.eks. en vanntank eller likende). Vi må da sende signaler mellom regulatoren og prosessen. Får å få til dette må vi bruke en DAQ (I/O) enhet. En del av DAQ enhets oppgave vil da være å konvertere mellom analoge og digitale signaler.



Et eksempel på en DAQ enhet kan være NI USB-6008:

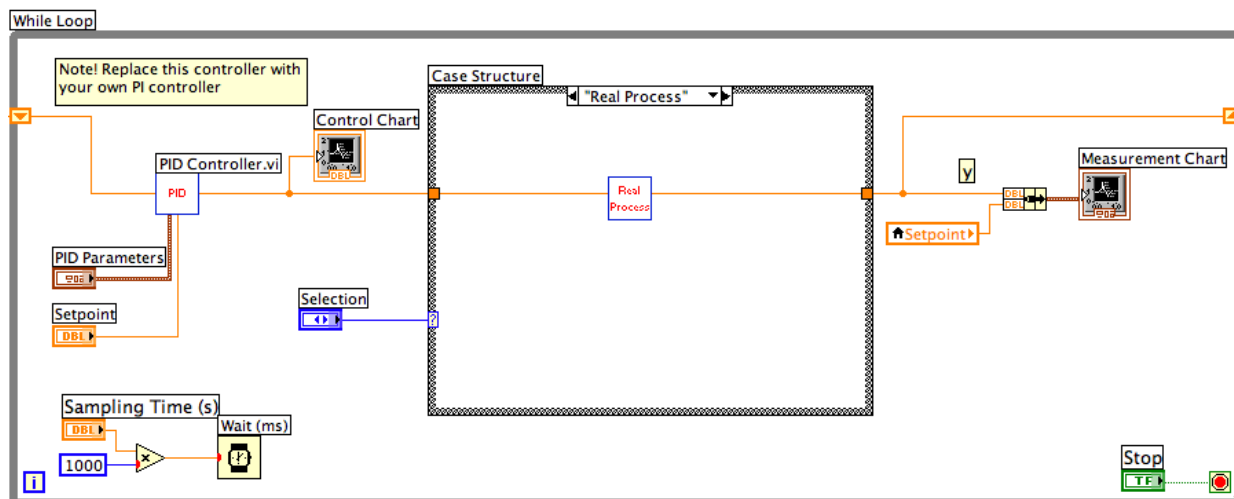


Denne enheten har 8 analoge innganger og 2 analoge utganger, i tillegg til 12 digitale inn/utganger.

Denne enheten blir brukt i mange fag ved Høgskolen i Telemark, Fakultet for teknologiske fag.

Denne kan enkelt knyttes opp mot PC og brukes ifm blant andre LabVIEW.

I LabVIEW kan et reguleringsystem se f.eks slik ut:



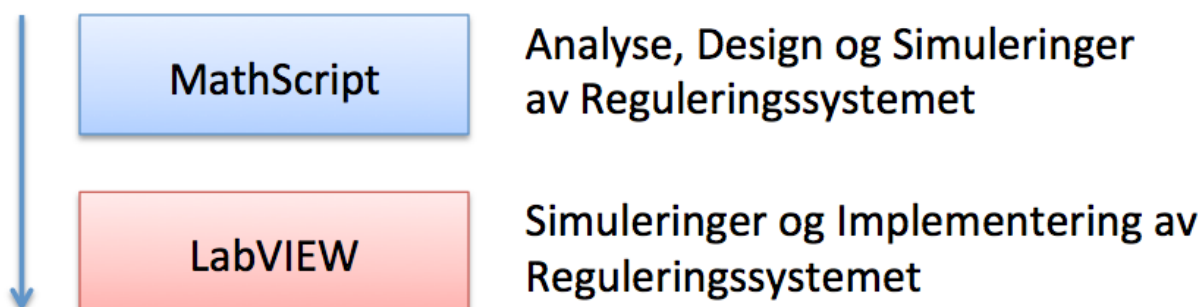
1.4 Dataverktøy

Vi vil bruke LabVIEW og MathScript som dataverktøy for å analysere, designe, simulere og implementere reguleringsystemet vårt.

LabVIEW er nyttig ifm simulering og regulering av dynamiske systemer, datainnsamling, m.m. MathScript er en tilleggsmodul til LabVIEW.

LabVIEW er et grafisk programmeringsspråk, mens MathScript er et tekstbasert programmeringsspråk.

Du må ha LabVIEW installert for at du skal kunne bruke MathScript, da MathScript startes fra LabVIEW IDE.

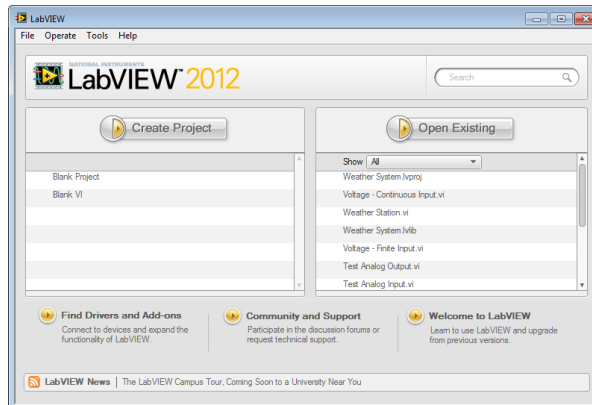


MathScript er best egnet i Analyse- og Designfasen, mens MathScript er best egnet til den praktiske implementeringen av reguleringsystemet med DAQ moduler, kontinuerlig kjøring av systemet, osv.

2 Hva er LabVIEW?

LabVIEW er nyttig ifm simulering og regulering av dynamiske systemer, datainnsamling, m.m.

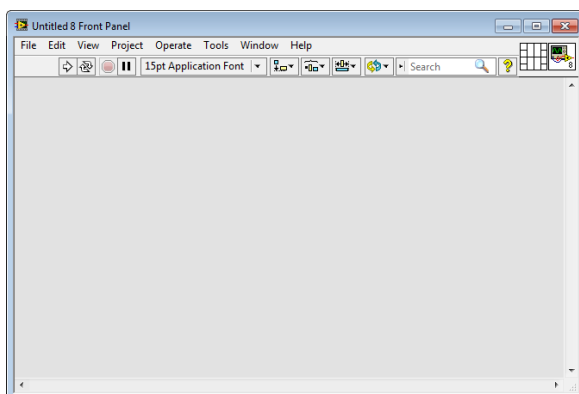
LabVIEW programmer kalles ”**Virtual Instruments**”, eller VIs, fordi de kan blant annet brukes til å lage virtuelle instrumenter i software, som f.eks oscilloscope og multimeter. Når du åpner LabVIEW, kommer du til ”Getting Started” vinduet. Velg da ”File → New VI (Ctrl + N)” for å starte opp selve LabVIEW.



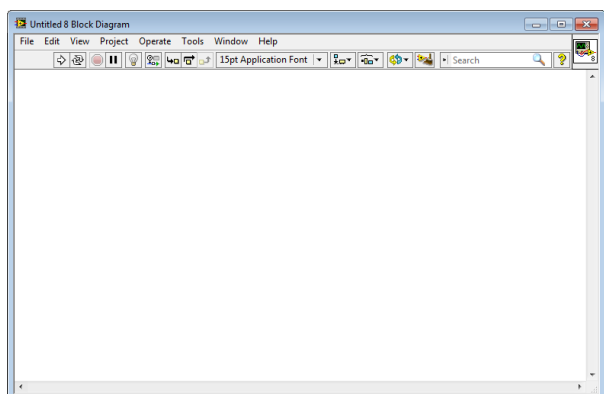
2.1 Frontpanel og Blokkdiagram

2 forskjellige vinduer kommer til syne: ”**Front Panel**” og ”**Block Diagram**”. Du bruker Frontpanlet til å lage det grafiske brukergrensesnittet (GUI/HMI), mens Blokkdiagrammet brukes til å utvikle selve koden, dvs. Logikken som skal til for å få programmet til å virke. Både Frontpanelet og Blokkdiagrammet blir lagret i en og samme fil (.vi).

Du kan enkelt bytte mellom ”Front Panel” og ”Block Diagram” ved å bruke ”**Ctrl + E**”.



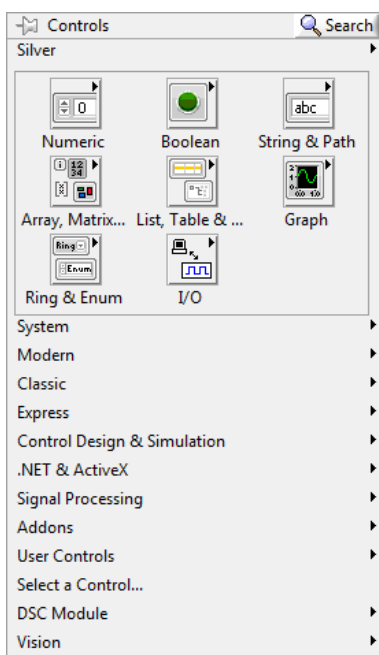
Front Panel → Brukergrensesnitt



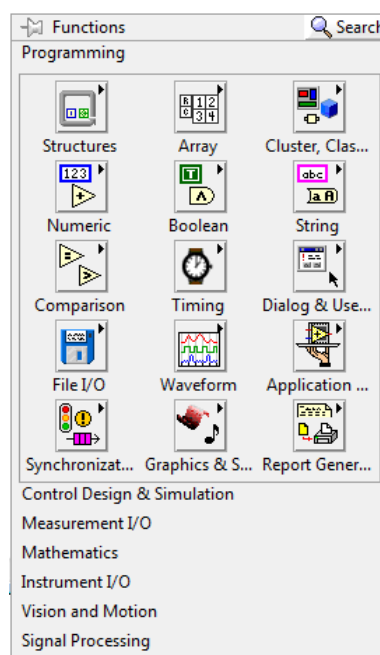
Block Diagram → Koding

2.2 Controls Palett og Functions Palett

Når du høyreklikker på “Front Panel”, dukker “**Controls Palette**” opp. Her finner du alle nødvendige elementer/objekter for å lage ditt brukergrensesnitt. Når du høyreklikker på “Block Diagram”, dukker “**Functions Palette**” opp. Her finner du funksjoner og logikk for å lage koden din.



Controls Palett (høyreklikk på frontpanelet)



Functions Palett (høyreklikk på blokkdiagrammet)

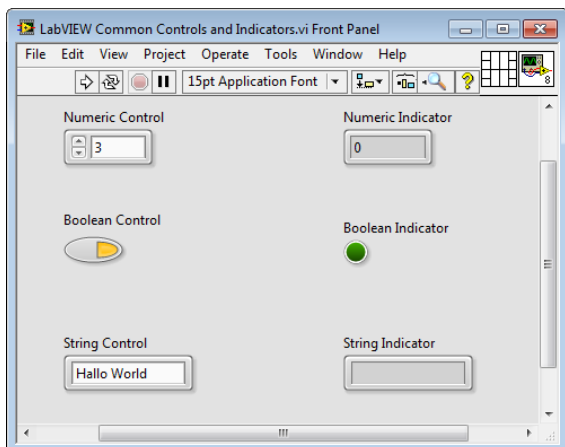
Merk! Disse kan se litt forskjellig ut avhengig av hvilke moduler du har installert på din PC, osv.

2.3 Controls og Indicators

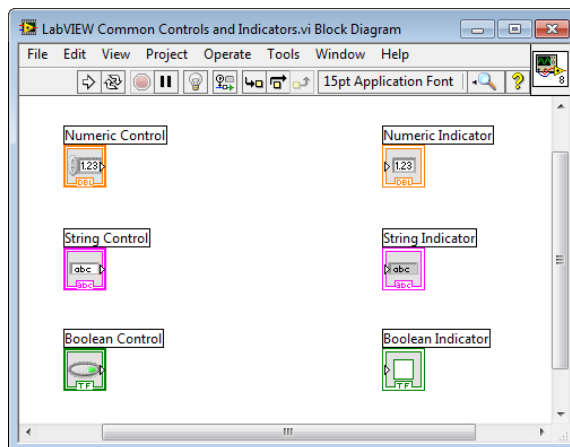
I LabVIEW skiller vi mellom “**Controls**” og “**Indicators**”. En “Control” er et element på Frontpanelet hvor brukeren kan forandre på verdier, osv., mens en “Indicator” er et element som ikke kan forandres, dvs man kan kun lese ut informasjon.

Vi har ulike typer for “Controls” og “Indicators”; de mest vanlige er **Numeric**, **Boolean** og **String**.

“Numerics” (orange farge) blir brukt til tallverdier, “Booleans” (grønn farge) blir brukt til True/False verdier, mens “Strings” (rød/rosa farge) blir brukt til tekst og bokstaver.



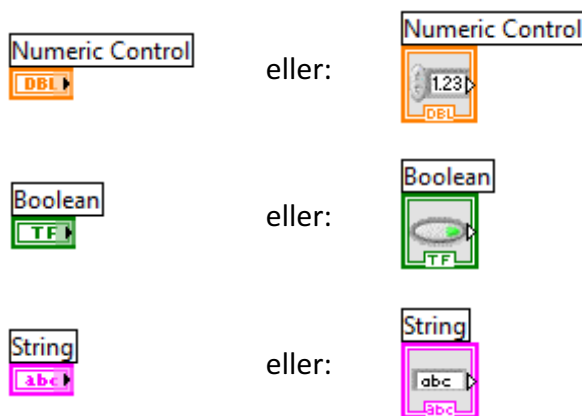
Controls og Indicators på Frontpanelet



Controls og Indicators på Blokkdiagrammet

Merk! LabVIEW har ulike layout for Controls og Indicators ("Silver", "Modern", "System" og "Classic"). Disse kan brukes om hverandre og det er en smakssak hvem du foretrekker.

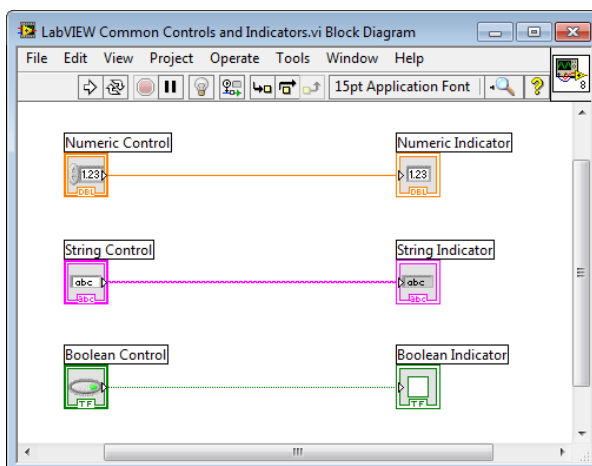
Ikonene på Blokkdiagrammet kan også variere avhengig av dine innstillinger:




Du kan enkelt bytte mellom disse ikonene ved å høyreklikke på ikonene og velge "View As Icon".

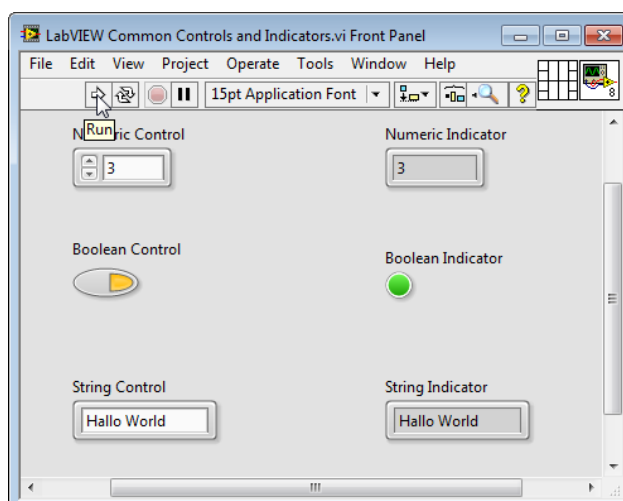
2.4 Wiring: Lag logikken

For å lage logikken i koden din må du trekke tråder (wiring) mellom de ulike ikonene på blokkdiagrammet ditt:



Kjør Programmet

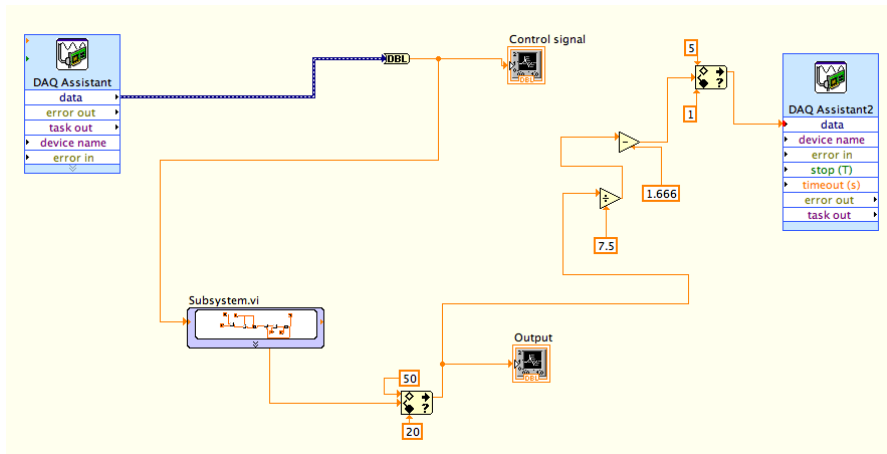
Vi er nå klare til å teste eller kjøre programmet. Når du er klar, trykk på **“Run”** knappen  i verktøylinja (eller **Ctrl + R**).



Det var alt!

LabVIEW er veldig enkelt å ta i bruk, men pass på! Koden kan fort bli rotete og uoversiktlig (“spagettikode”) hvis du ikke jobber strukturert. Bruk kommentarer i koden, samt pass på at alle “ledningene” (wires) mellom de ulike ikonene er “rette og fine”.

Nedenfor ser du et typisk eksempel på såkalt spagettikode – dvs rotete og ustrukturert kode.



Dette er bare et lite program – tenk dere hvor vanskelig det vil være å forstå og vedlikeholde denne koden etter hvert som programmet blir større.

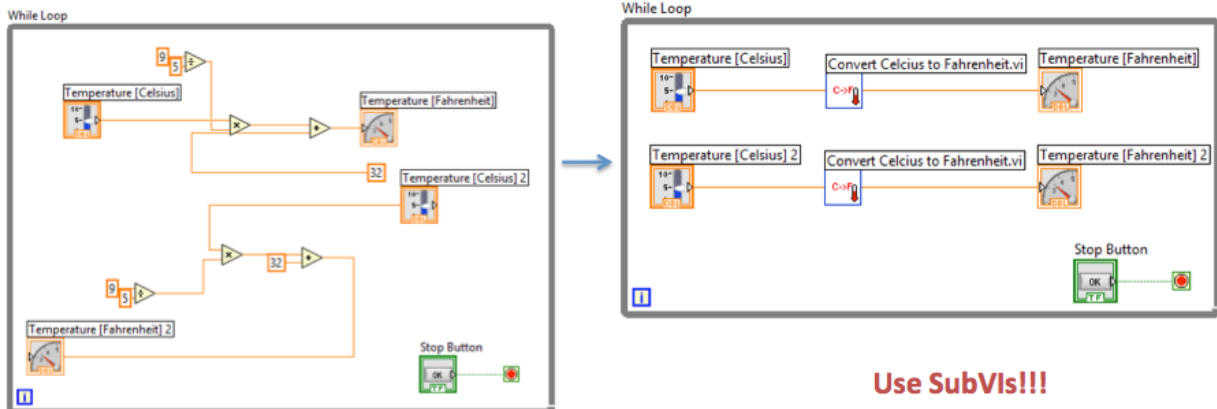
Good vs. Bad Code

The Flow should go from left to right



Avoid Spaghetti Code!

Make your code readable for others!



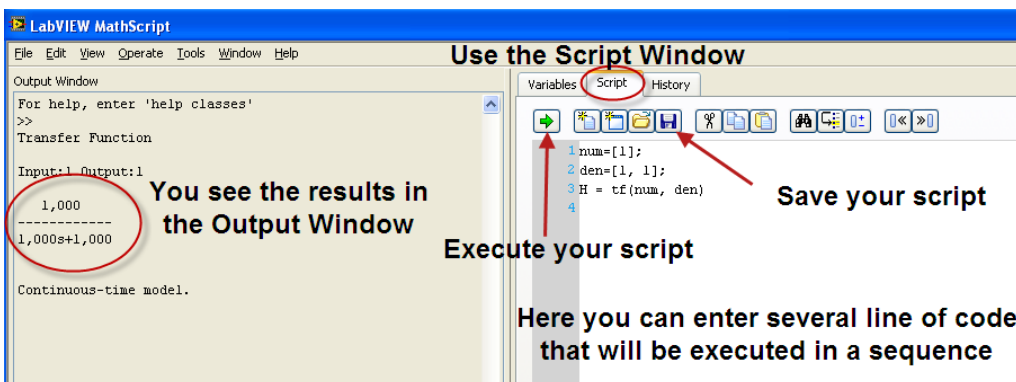
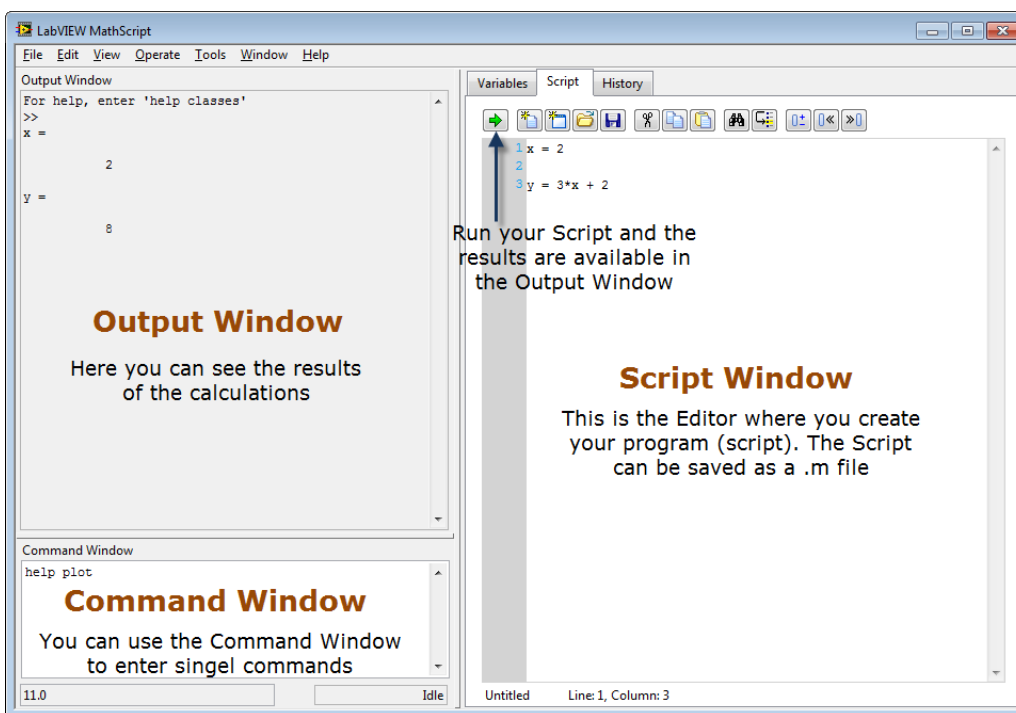
Use SubVis!!!

3 Hva er MathScript?

3.1 Innledning

MathScript er en tilleggspakke til LabVIEW. Syntaksen er identisk som MATLAB. MathScript er velegnet ifm. design, analyse og simulering av reguleringsystemer da det finnes mange innebygde funksjoner for dette.

Vi åpner MathScript fra menyen i LabVIEW: Tools → MathScript Window.



Exempel:

Gitt følgende funksjon:

$$y(x) = 3x + 2$$

Vi ønsker å finne $y(2)$

Først må vi definere x :

```
x = 2
```

Deretter kan vi definere funksjonen:

```
y = 3*x + 2
```

MathScript gir da følgende svar:

```
y =
```

```
8
```

Merk! Maple kan utføre symbolsk matematikk, mens MathScript (og VB/C#) er et numerisk verktøy. Dette betyr at du alltid må definere verdier for variablene dine før du bruker dem i et matematisk uttrykk.

Merk! Syntaksen i MathScript er ganske lik syntaksen i VB/C#, men MathScript er mye enklere på den måten at du ikke trenger å deklarere variablene og bestemme en datatype før du bruker dem. Dette blir automatisk håndtert av kompilatoren.

[Slutt på eksempel]

3.2 Plotting

MathScript er veldig anvendelig til å plote verdier, funksjoner, m.m.

Eksempel:

Vi ønsker å plote følgende uttrykk vha. MathScript:

$$y(x) = 3x + 2$$

Vi bruker da **plot()** funksjonen i MathScript.

```
x = 0:5;  
y = 3*x + 2;  
plot(x, y)
```

[Slutt på eksempel]

4 Introduksjon til Transferfunksjoner

4.1 Transferfunksjoner

Transferfunksjoner er nyttige å bruk ved analyse og design av dynamiske systemer.

Transferfunksjoner er modeller basert på Laplace-transformasjonen. Transferfunksjonen gir altså en modellbeskrivelse i s-planet (der s er Laplace-operatoren).

En transferfunksjon kan skrives på følgende generelle form:

$$H(s) = \frac{y(s)}{u(s)}$$

Hvor $y(s)$ er utgangen mens $u(s)$ er inngangen.

Transferfunksjonen er altså forholdet mellom inngangen og utgangen når alle andre inngangsvariable, samt alle initialbetingelser er satt lik null.

Merk! Transferfunksjoner gjelder bare for lineære systemer.

En transferfunksjon kan skrives på følgende generelle polynomform:

$$H(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}$$

Der telleren til transferfunksjonen beskriver nullpunktene til systemet, mens nevneren beskriver polene til systemet.

4.2 MathScript

I MathScript definerer vi transferfunksjoner vha. den innebygde **tf** funksjonen:

```
num=[bm, bm_1, bm_2, ... , b1, b0];  
den=[an, an_1, an_2, ... , a1, a0];  
H = tf(num, den)
```

Eksempel:

1. Gitt følgende transferfunksjon:

$$H(s) = \frac{2s^2 + 3s + 4}{5s + 9}$$

MathScript kode:

```
num=[2, 3, 4];  
den=[5, 9];  
H = tf(num, den)
```

2. Gitt følgende transferfunksjon:

$$H(s) = \frac{4s^4 + 3s + 4}{5s^2 + 9}$$

Merk! Hvis noen ledd mangler, setter vi disse lik 0. Transferfunksjonen kan bli omskrevet som:

$$H(s) = \frac{4s^4 + 0 \cdot s^3 + 0 \cdot s^2 + 3s + 4}{5s^2 + 0 \cdot s + 9}$$

MathScript kode:

```
num=[4, 0, 0, 3, 4];  
den=[5, 0, 9];  
H = tf(num, den)
```

3. Gitt følgende transferfunksjon:

$$H(s) = \frac{7 + 3s + 2s^2}{5s + 6s^2}$$

Merk! Hvis noen ledd mangler, setter vi disse lik 0, samt passer på at de kommer i riktig rekkefølge. Transferfunksjonen kan bli omskrevet som:

$$H(s) = \frac{2s^2 + 3s + 7}{6s^2 + 5s}$$

MathScript kode:

```
num=[2, 3, 7];  
den=[6, 5, 0];  
H = tf(num, den)
```

[Slutt på eksempel]

4.3 Differensiallikninger

Vi kan si at transferfunksjonen gir en ekstern inn-ut representasjon av et system. Differensiallikningene derimot gir en intern representasjon av systemet fordi differensiallikningene beskriver systemets indre liv.

Vi kan finne transferfunksjonen fra differensiallikning(e) vha. Laplace transformasjon.

Vi har følgende Transformasjonspar:

Derivasjon:

$$\dot{x} \Leftrightarrow sx(s)$$

Integrasjon:

$$\int x \Leftrightarrow \frac{1}{s}x(s)$$

Tidsforsinkelse:

$$u(t - \tau) \Leftrightarrow u(s)e^{-\tau s}$$

Eksempel:

Gitt følgende differensiallikning:

$$\dot{x} = -0.5x + 2u$$

Vi ønsker å finne transferfunksjonen:

$$H(s) = \frac{x(s)}{u(s)}$$

Laplace gir:

$$sx(s) = -0.5x(s) + 2u(s)$$

Videre:

$$sx(s) + 0.5x(s) = 2u(s)$$

Videre:

$$x(s)(s + 0.5) = 2u(s)$$

Videre:

$$\frac{x(s)}{u(s)} = \frac{2}{s + 0.5} = \frac{4}{2s + 1}$$

Dette gir:

$$H(s) = \frac{x(s)}{u(s)} = \frac{4}{2s + 1}$$

[Slutt på eksempel]

Eksempel:

Gitt følgende differensiallikning (med tidsforsinkelse):

$$\dot{x} = -\frac{1}{T}[x + Ku(t - \tau)]$$

Vi ønsker å finne transferfunksjonen:

$$H(s) = \frac{x(s)}{u(s)}$$

Laplace gir:

$$sX(s) = -\frac{1}{T}X(s) + \frac{K}{T}U(s)e^{-\tau s}$$

Videre:

$$TsX(s) + X(s) = Ku(s)e^{-\tau s}$$

Videre:

$$X(s)(Ts + 1) = Ku(s)e^{-\tau s}$$

Videre:

$$\frac{X(s)}{U(s)} = \frac{K}{Ts + 1} e^{-\tau s}$$

Som gir:

$$H(s) = \frac{X(s)}{U(s)} = \frac{K}{Ts + 1} e^{-\tau s}$$

[Slutt på eksempel]

Motsatt kan vi finne differensiallikningen(e) fra transferfunksjonen.

Eksempel:

Gitt følgende transferfunksjon:

$$H(s) = \frac{x(s)}{u(s)} = \frac{3}{0.5s + 1}$$

Vi ønsker å finne differensiallikningen.

Vi gjør følgende:

$$x(s)(0.5s + 1) = 3u(s)$$

Videre:

$$0.5sx(s) + x(s) = 3u(s)$$

Inverse Laplace:

$$0.5\dot{x} + x = 3u$$

Dette gir følgende differensiallikning:

$$\dot{x} = -2x + 6u$$

[Slutt på eksempel]

4.4 Sprangrespons

Vi ønsker ofte å finne et systems sprangrespons, dvs. systemets dynamiske oppførsel over tid etter et sprang i pådraget.

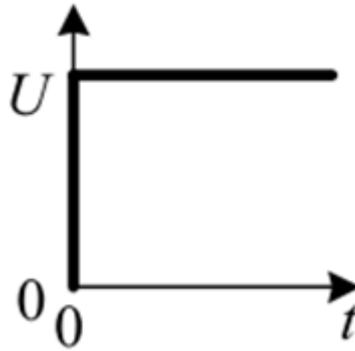
For en gitt transferfunksjon:

$$H(s) = \frac{y(s)}{u(s)}$$

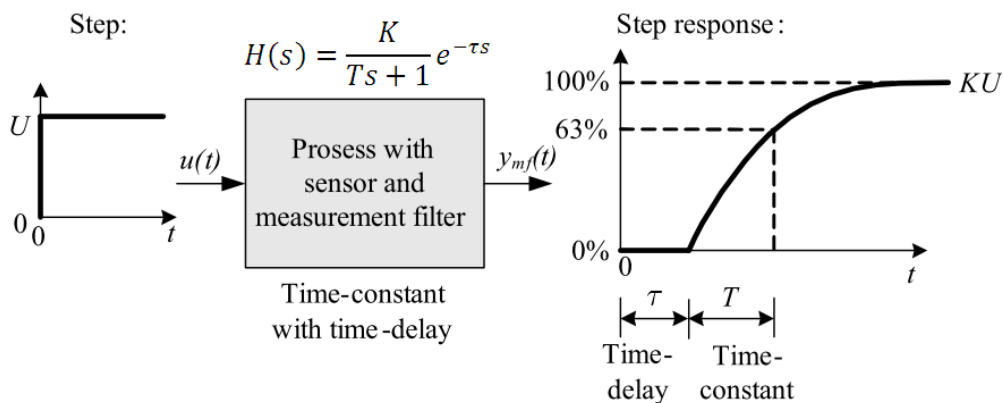
Får vi:

$$y(s) = H(s)u(s)$$

der vi vanligvis bruker et enhetssprang på inngangen, i dette tilfellet pådraget u , dvs pådraget u øker fra 0 til 1 ved $t = 0$.



Dvs. vi ønsker å plote $y(t)$ etter et sprang på inngangen. Nedenfor ser vi et typisk eksempel:



[Figure: F. Haugen, Advanced Dynamics and Control: TechTeach, 2010]

For et enhetsprang har vi følgende Laplacetransformasjon:

$$\boxed{\frac{1}{s} \Leftrightarrow 1}$$

For et sprang med størrelse k (amplitude k) har vi følgende Laplacetransformasjon:

$$\frac{k}{s} \Leftrightarrow k$$

Vi finner dermed sprangresponsen for et gitt system på følgende måte:

Gitt transferfunksjonen:

$$H(s) = \frac{y(s)}{u(s)}$$

1. Vi ordner det slik at utgangen står på høyre side:

$$y(s) = H(s)u(s)$$

2. Deretter finner vi inngangsvariabelens transformasjonspar og setter denne inn i uttrykket.

For et sprang med størrelse k (amplitude k) får vi da følgende:

$$u(s) = \frac{k}{s}$$

Dvs.:

$$y(s) = H(s) \cdot \frac{k}{s}$$

3. Tilslutt tidsresponsen ved å ta inverse laplace av uttrykket over.

Eksempel:

Gitt følgende system:

$$H = \frac{y(s)}{u(s)} = \frac{3}{2s + 1}$$

Vi får da:

$$y(s) = H(s)u(s)$$

Hvor u er et enhetssprang:

$$u(s) = \frac{1}{s}$$

Dvs.:

$$y(s) = \frac{3}{2s + 1} \cdot \frac{1}{s} = \frac{3}{(2s + 1)s}$$

Vi bruker følgende Laplace-transformasjonspar (som vi finner i en tabell):

$$\boxed{\frac{K}{(Ts + 1)s} \Leftrightarrow K(1 - e^{-t/T})}$$

Vi får da:

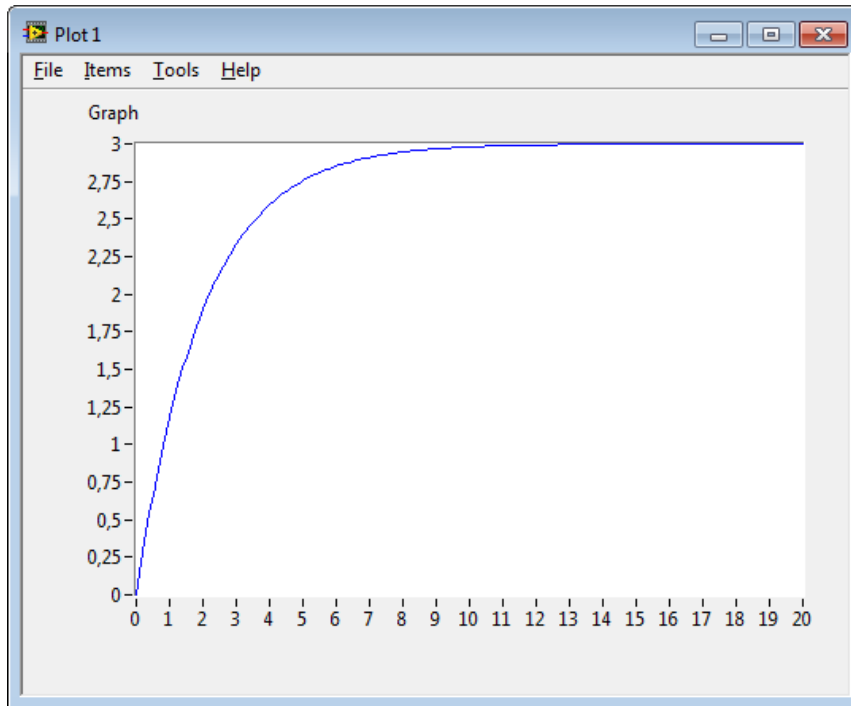
$$\underline{y(t) = 3(1 - e^{-t/2})}$$

MathScript:

Vi kan da plote denne sprangresponsen vha. MathScript:

```
t=0:0.1:20;
y = 3*(1-exp(-t/2));
plot(t,y)
```

Som gir følgende sprangrespons:



Vi ser også at den stasjonære verdien i dette tilfellet er $y_s = 3$. Den stasjonære verdien er den verdien vi får når $t \rightarrow \infty$.

Dette er jo litt tungvint siden vi må gjøre en del manuelle beregninger, men vha. MathScript kan vi gjøre det mye enklere:

```
num=[3];
den=[2, 1];
H = tf(num, den);

step(H, t)
```

Dvs. vi kan definere transfer funksjonen (vha. **tf** funksjonen) direkte i MathScript og bruke den innebygde **step** funksjonen til å beregne og eller plote sprangresponsen. Resultatet blir det samme.

[Slutt på eksempel]

4.5 Sluttverditeoremet

I mange tilfeller er det den stasjonære responstiden man er interessert i, det vil si tidsresponsen når $t \rightarrow \infty$.

Da kan vi bruke sluttverditeoremet:

$$y_s = \lim_{t \rightarrow \infty} y(t) = \lim_{s \rightarrow 0} s \cdot y(s)$$

Eksempel:

Gitt følgende system:

$$H = \frac{y(s)}{u(s)} = \frac{3}{2s + 1}$$

Vi får:

$$y_s = \lim_{t \rightarrow \infty} y(t) = \lim_{s \rightarrow 0} s \cdot y(s) = \lim_{s \rightarrow 0} s \frac{3}{(2s + 1)s} = \frac{3}{2 \cdot 0 + 1} = \underline{3}$$

Som du ser får vi samme resultat som vi fikk i forrige eksempel.

[Slutt på eksempel]

5 Integrator

Transferfunksjonen for en integrator:

$$H(s) = \frac{y(s)}{u(s)} = \frac{K}{s}$$

5.1 Sprangrespons

Vi ønsker å finne $y(t)$:

Laplace transformasjonen for et sprang er som følger:

$$\frac{1}{s} \Leftrightarrow 1$$

Sprangresponsen for en integrator blir da:

$$y(s) = H(s)u(s) = \frac{K}{s} \cdot \frac{U}{s} = KU \frac{1}{s^2}$$

Vi har at:

$$\frac{1}{s^2} \Leftrightarrow t$$

Dette gir:

$$\underline{y(t) = KUt}$$

→ Vi ser at sprangresponsen til en integrator er en rampe.

Konklusjon Jo større K , jo større stigningstall.

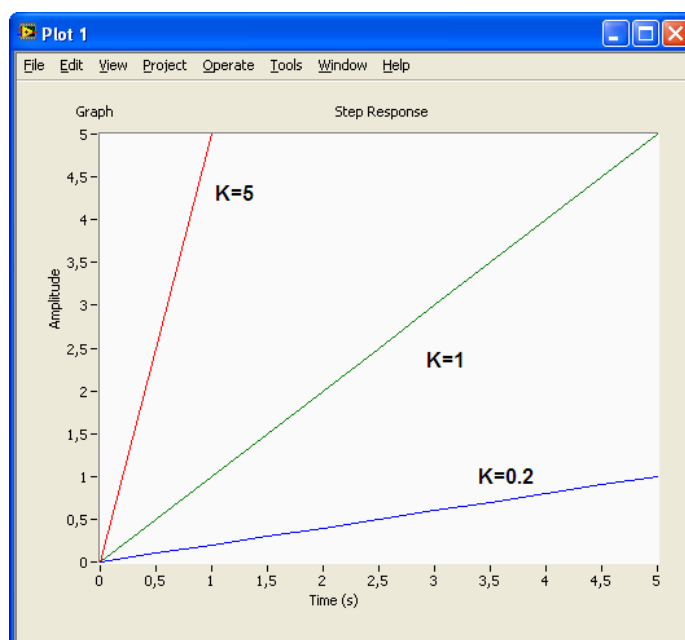
Eksempel

I MathScript kan vi bruke **step** funksjonen for å simulere sprangresponsen. Vi setter f.eks $K = 0.2, 1, 5$.

MathScript kode blir som følger:

```
t=[0:0.5:5];  
K=0.2  
num=[K];  
den=[1 0];  
H1=tf(num,den);  
  
K=1  
num=[K];  
den=[1 0];  
H2=tf(num,den);  
  
K=5  
num=[K];  
den=[1 0];  
H3=tf(num,den);  
  
step(H1,H2,H3,t)  
axis([0, 5, 0, 5])
```

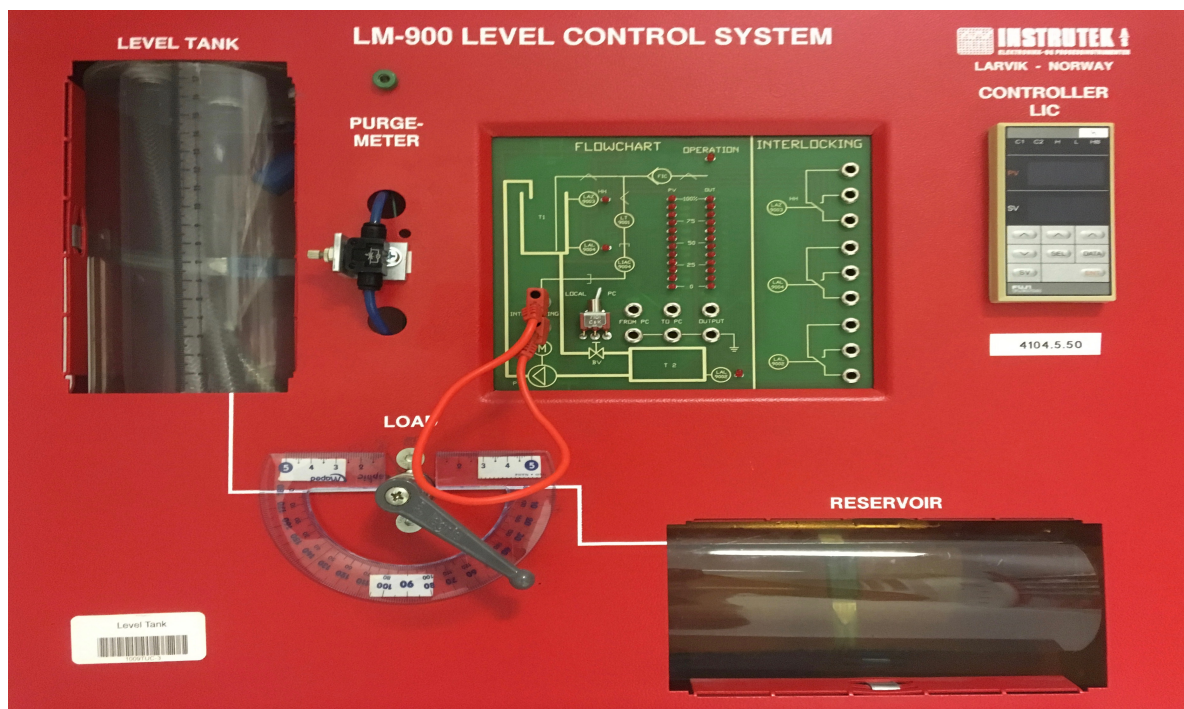
Merk! Å bruke en For Loop ville kanskje vært bedre rent programmeringsteknisk.



[Slutt på eksempel]

Eksempel: Vanntank

Nedenfor ser vi en enkel vanntankmodell:



Differensiallikningen for systemet er som følger:

$$A_t \dot{h} = K_p u - F_{out}$$

eller:

$$\dot{h} = \frac{1}{A_t} [K_p u - F_{out}]$$

hvor:

- h [cm] er nivået i vanntanken
- u [V] er pådragssignalet til pumpa
- A_t [cm²] arealet til bunnen i vanntanken
- K_p [(cm³/s)/V] er pumpeforsterkningen
- F_{out} [cm³/s] er utstrømningen gjennom ventilen i bunnen av tanken.

Vha. MathScript kan vi simulere systemet:

```
clear, clc
% Model

% Model parameters
Kp = 15;
At = 80;
Fout = 10;

h(1) = 0; %Initial value
Ts = 0.1; %Sampling Time
```

```

uk = 1; %Step

k = 1:100;
N = length(k);

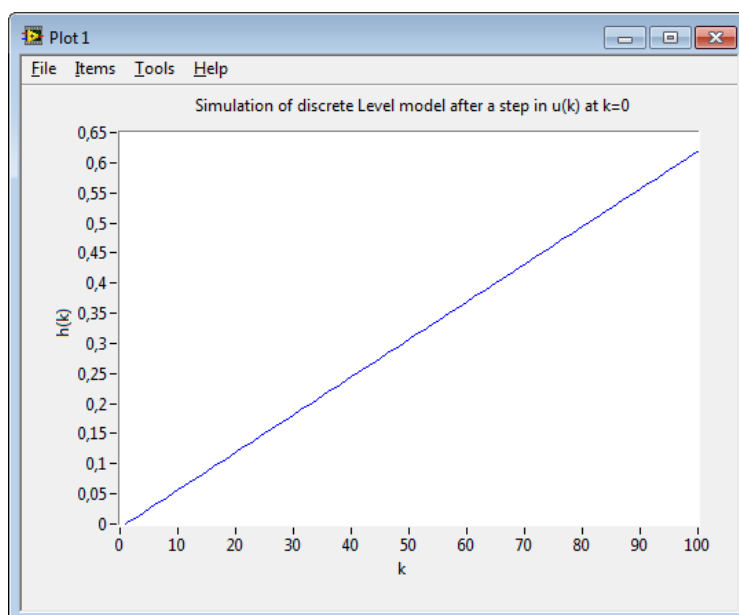
for i=2:N
    h(i) = h(i-1) + (Ts/At)*(Kp*uk-Fout);
end

plot(k, h)
title('Simulation of discrete Level model after a step in u(k)
at k=0')
xlabel('k')
ylabel('h(k)')

```

Merk! Her har vi diskretisert modellen og simulert vha en For Loop. Vi vil lære mer om diskretisering senere.

Dette gir følgende resultat:



Som vi ser så er dette en typisk integrator.

Utrekning:

Vi kan finne transferfunksjonen $H(s) = \frac{h(s)}{u(s)}$

Vi starter med følgende differensiallikning:

$$\dot{h} = \frac{1}{A_t} [K_p u - F_{out}]$$

Laplace gir:

$$sh(s) = \frac{K_p}{A_t} u(s)$$

Vi setter $F_{out} = 0$

Dette gir:

$$H(s) = \frac{h(s)}{u(s)} = \frac{K_p}{A_t} \frac{1}{s} u(s)$$

Vi introduserer $\frac{K_p}{A_t} \equiv K$, da får vi:

$$\underline{H(s) = \frac{h(s)}{u(s)} = \frac{K}{s} u(s)}$$

→ vi ser at prosessen er en integrator.

Vi ønsker å finne de matematiske uttrykket for sprangresponsen $h(t)$.

Laplaceformasjonsparet for et sprang er som følger:

$$\boxed{\frac{1}{s} \Leftrightarrow 1}$$

Sprangresponsen for en integrator blir da:

$$h(s) = H(s)u(s) = \frac{K}{s} \cdot \frac{U}{s} = KU \frac{1}{s^2}$$

Vi bruker følgende Laplaceformasjonspar for å finne $h(t)$:

$$\boxed{\frac{1}{s^2} \Leftrightarrow t}$$

Dette gir til slutt:

$$\underline{h(t) = KUt}$$

→ Vi ser at sprangresponsen for en integrator er en rampe.

En større K vil gi et større stigningstall og integrasjonen vil gå fortere. Simulering i MathScript vil vise dette.

[Slutt på eksempel]

6 1.ordens systemer

1.ordens transferfunksjon:

$$H(s) = \frac{K}{Ts + 1}$$

K er forsterkning

T er tidskonstant

6.1 Sprangrespons

For et 1.ordens system har vi:

$$H = \frac{y(s)}{u(s)} = \frac{K}{Ts + 1}$$

Vi får da:

$$y(s) = H(s)u(s)$$

Hvor:

$$u(s) = \frac{U}{s}$$

Dvs.:

$$y(s) = \frac{K}{Ts + 1} \cdot \frac{U}{s}$$

Vi bruker følgende Laplace-transformasjonspar (som vi finner i en tabell):

$$\frac{K}{(Ts + 1)s} \Leftrightarrow K(1 - e^{-t/T})$$

Tilslutt får vi da følgende:

$$\underline{y(t) = KU(1 - e^{-t/T})}$$

6.2 MathScript

Eksempel:

$$H(s) = \frac{K}{Ts + 1}$$

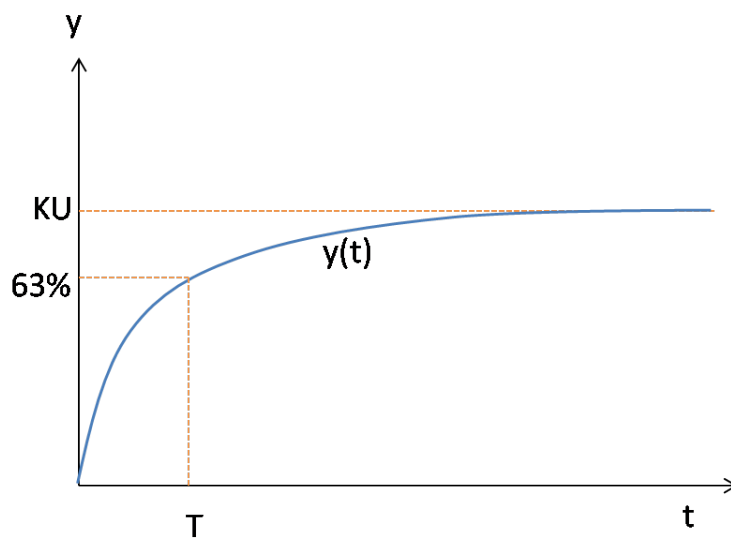
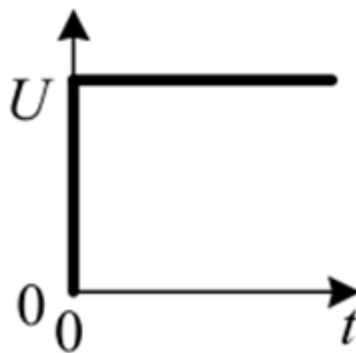
$$K = 1, T = 2$$

MathScript:

```
K = 1;  
T = 2;  
num = [K];  
den = [T, 1];  
H = tf(num, den)
```

Sprangrespons:

$$y(s) = H(s)u(s)$$



Vanligvis er $U = 1$, dvs en enhetsprang

Mathscript:

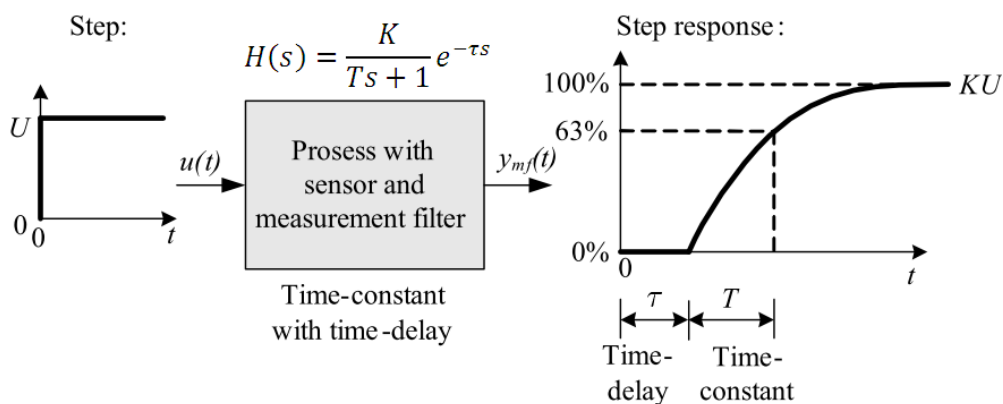
`step(H)`

6.3 1.ordens system med tidsforsinkelse

En 1.ordens transferfunksjon med tidsforsinkelse:

$$H(s) = \frac{K}{Ts + 1} e^{-\tau s}$$

En sprangrespons for en slik transferfunksjon har følgende karakteristik:



[Figure: F. Hagen, Advanced Dynamics and Control: TechTeach, 2010]

Fra sprangresponsen kan vi enkelt finne K , T and τ .

6.3.1 MathScript

Eksempel:

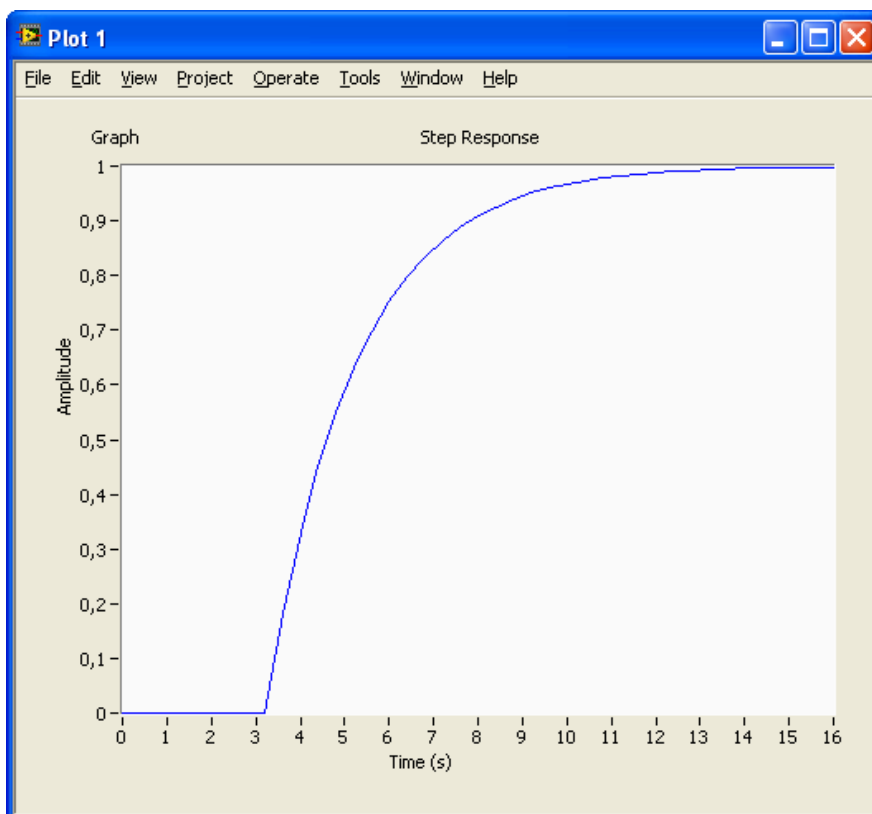
$$H(s) = \frac{K}{Ts + 1} e^{-\tau s}$$

$K = 1, T = 2, \tau = 3$

MathScript:

```
K = 1;
T = 2;
delay=3;
H = sys_order1(K, T, delay)
step(H)
```

Dette gir følgende plot:



6.3.2 Eksempler

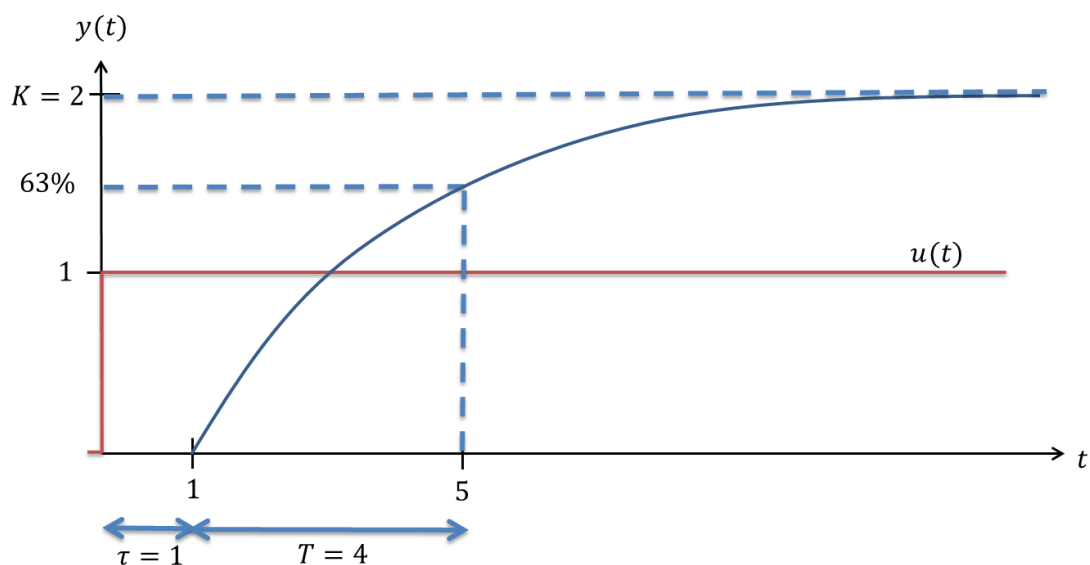
Eksempel:

Gitt følgende system:

$$H = \frac{y(s)}{u(s)} = \frac{2}{4s + 1} e^{-s}$$

Vi ønsker å finne sprangresponsen til systemet.

Vi kan illustrere sprangresponsen slik:



MathScript:

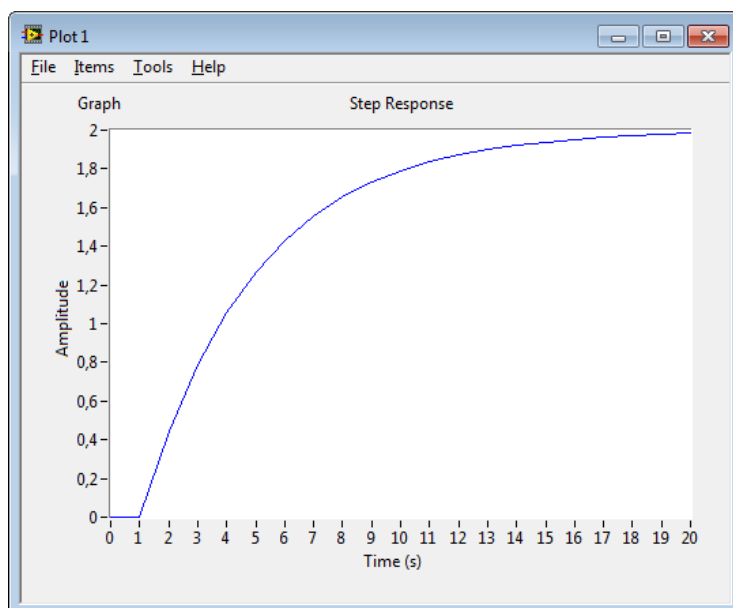
```
K=2;
T=4;
delay=1;

H = sys_order1(K,T,delay)

t=0:20;

step(H, t)
```

Sprangresponsen blir:



[Slutt på eksempel]

Eksempel – Air Heater

Nenedfor ser vi en såkalt Air Heater prosess (varmluftsprosess):



Vi kan definere følgende matematiske modell for denne prosessen:

$$\dot{T}_{out} = \frac{1}{\theta_t} \{-T_{out} + [K_h u(t - \theta_d) + T_{env}]\}$$

hvor:

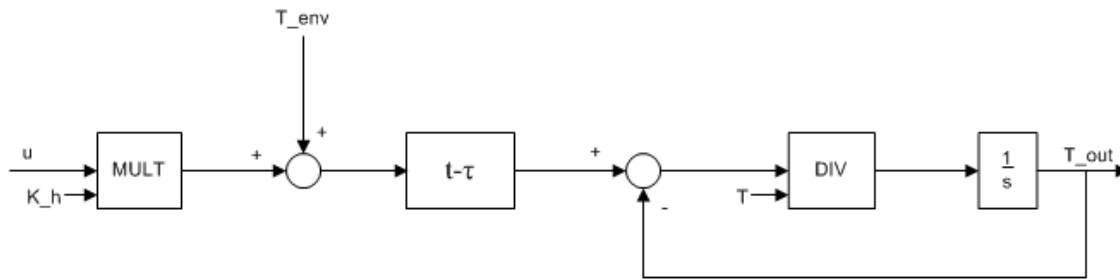
- $u [V]$ er pådragssignalet til varmeelementet.
- $\theta_t [s]$ er prosessens tidskonstant.
- $K_h [^{\circ}C / V]$ er forsterkningen til varmeelementet.
- $\theta_d [s]$ er tidsforsinkelen til prosessen representert ved lufttransport gjennom røret.
- $T_{env} [^{\circ}C]$ er romtemperaturen. Dette er temperaturen ved utgangen av røret når pådragssignalet er satt til null for en relativt lang periode (noen minutter).

Varmeelement: Lufta blir varmet opp av et varmeelement. Spenningssignalet er $0 - 5 V$ (min effekt, max effekt).

Vifte: En vifte blåser luft gjennom røret. Vifta kan manuelt justeres.

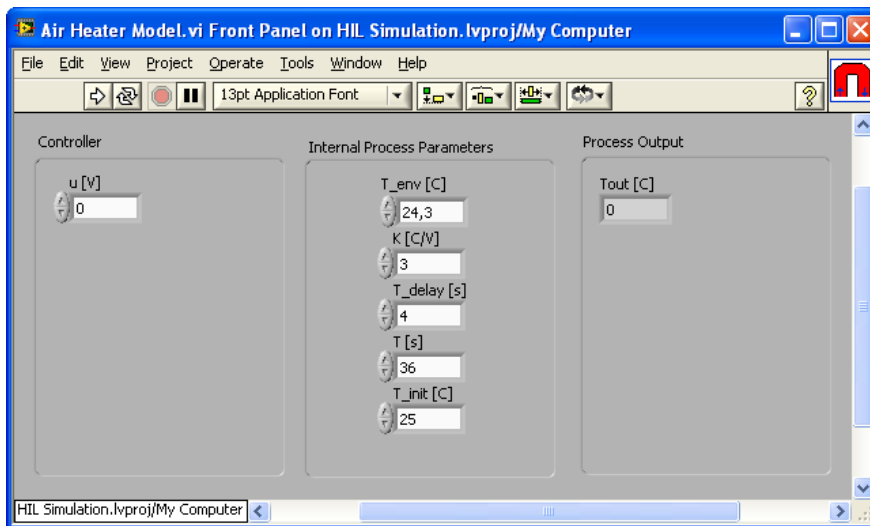
Temperatur sensorer: 2 Pt100 temperaturelementer er tilgjengelige. Området er mellom $1 - 5 V$, som tilsvarer en temperatur mellom $20 - 50^{\circ}C$ (lineær relasjon).

Et typisk blokkdiagram for denne prosessen ser ut som følger:

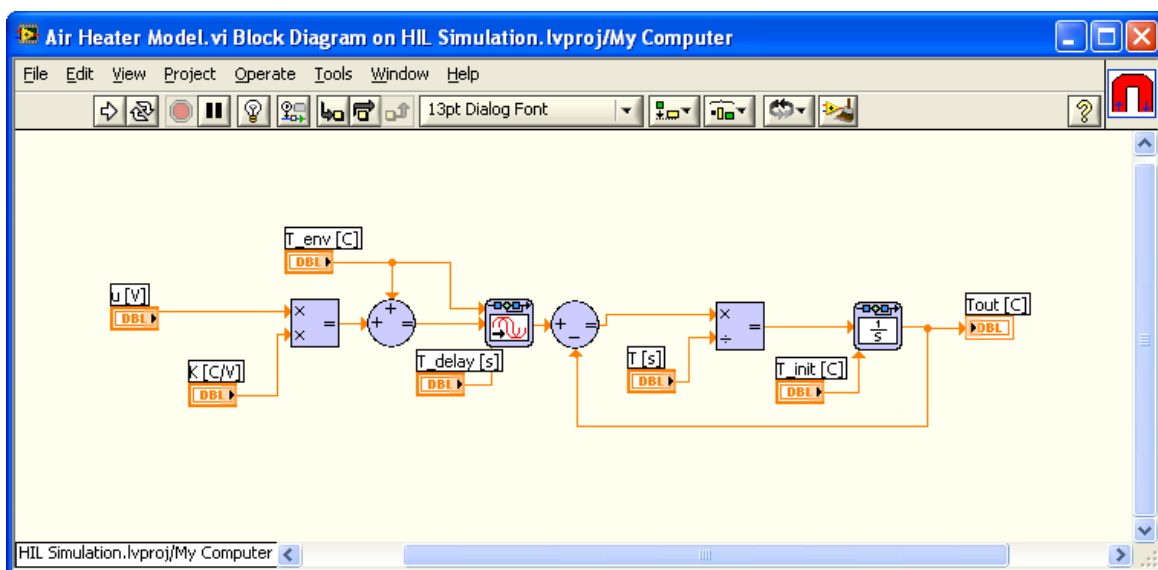


Vi implementerer blokkdiagrammet i LabVIEW vha “**LabVIEW Control Design and Simulation Module**” og “**Control and Simulation Loop**”. Vi kan også bruke en ordinær While Loop. Da må vi lage blokkdiagrammet i et såkalt “**Simulation Subsystem**” – dette er anbefalt!

Frontpanelet:



Blokkdiagrammet (et Simulation Subsystem er brukt):



Bruk f.eks følgende verdier:

$$\theta_t = 22 \text{ sek}$$

$$\theta_d = 2 \text{ sek}$$

$$K_h = 3.5 \frac{^\circ\text{C}}{\text{V}}$$

$$T_{env} = 21.5 \text{ }^\circ\text{C}$$

Vi finner transferfunksjonen for systemet:

Vi har følgende differensiallikning:

$$\dot{T}_{out} = \frac{1}{\theta_t} \{-T_{out} + [K_h u(t - \theta_d) + T_{env}]\}$$

Vi bruker Laplace og finner transferfunksjonen fra $u(s)$ til $T_{out}(s)$, dvs.:

$$H(s) = \frac{T_{out}(s)}{u(s)}$$

Dette gir:

$$\dot{T}_{out} = -\frac{T_{out}}{\theta_t} + \frac{K_h}{\theta_t} u(t - \theta_d)$$

Laplace:

$$sT_{out}(s) = -\frac{T_{out}}{\theta_t} + \frac{K_h}{\theta_t} e^{-\theta_d s} u(s)$$

$$T_{out}(s) \left(s + \frac{1}{\theta_t} \right) = \frac{K_h}{\theta_t} e^{-\theta_d s} u(s)$$

$$\frac{T_{out}(s)}{u(s)} = \frac{\frac{K_h}{\theta_t}}{s + \frac{1}{\theta_t}} e^{-\theta_d s}$$

Tilslutt får vi:

$$\frac{T_{out}(s)}{u(s)} = \frac{K_h}{\theta_t s + 1} e^{-\theta_d s} = \frac{K}{Ts + 1} e^{-\tau s}$$

Vi ser at dette er et 1.ordens system med tidsforsinkelse/dødtid.

MathScript:

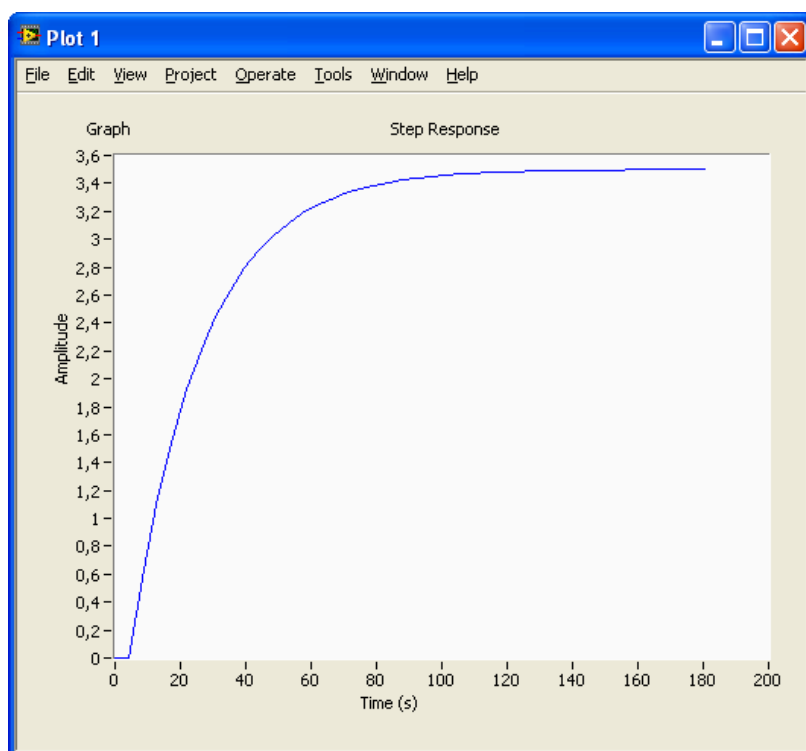
```
clear
clc
close all

% Process Transfer function
K = 3.5;
T = 22;
delay = 2;

H = sys_order1(K, T, delay)

step(H)
```

Som gir følgende resultat:



[Slutt på eksempel]

7 2.ordens systemer

2.ordens transferfunksjon:

$$H(s) = \frac{K}{as^2 + bs + c} = \frac{K\omega_0^2}{s^2 + 2\zeta\omega_0s + \omega_0^2} = \frac{K}{\left(\frac{s}{\omega_0}\right)^2 + 2\zeta\frac{s}{\omega_0} + 1}$$

K er forsterkning

ζ (zeta) er relativ dempingsfaktor

ω_0 [rad/s] er udempet resonansfrekvens

7.1 Responstid

Tidskonstanten for et 2.ordens system er tilnærmet lik:

$$T_r \approx \frac{1.5}{\omega_0}$$

7.2 MathScript

Eksempel:

$$H(s) = \frac{1}{s^2 + s + 1}$$

Dvs: $K = 1, \omega_0 = 1, \zeta = 1$

MathScript kode:

```
K = 1;  
w0 = 1;  
zeta = 1;  
  
a = 1/w0^2;  
b = 2*zeta/w0;  
c = 1;  
  
num = [K];
```

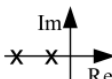
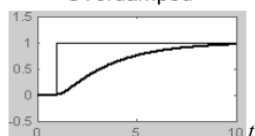
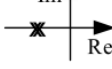
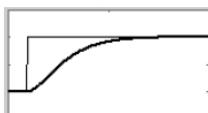
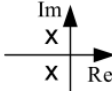
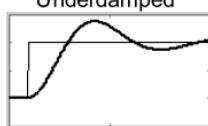
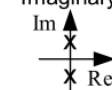
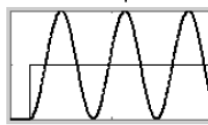
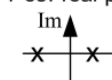
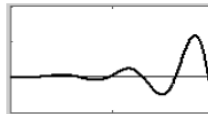
```
den = [a, b, c]
H = tf(num, den)
```

Sprangrespons:

```
step(H)
```

7.3 Sprangrespons og stabilitet

Verdien på ζ er avgjørende for systems sprangrespons og stabilitetsegenskaper:

Value of ζ	Poles p_1 and p_2	Type of step response $y(t)$
$\zeta > 1$	Real and distinct 	Overdamped 
$\zeta = 1$	Real and multiple 	Critically damped 
$0 < \zeta < 1$	Complex conj. 	Underdamped 
$\zeta = 0$	Imaginary 	Undamped 
$\zeta < 0$	Pos. real part 	Unstable 

[Figure: F. Haugen, Advanced Dynamics and Control: TechTeach, 2010]

7.4 Poler

Systemets stabilitetsegenskaper er definert av polens plassering i det komplekse plan.

Gitt følgende generelle 2.ordens system:

$$H(s) = \frac{K}{as^2 + bs + c}$$

Polene finner vi ved å sette nevneren lik null:

$$as^2 + bs + c = 0$$

Dvs. vi har en standard 2.ordens likning:

$$ax^2 + bx + c = 0$$

Løsningen på denne er som kjent:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

7.5 Spesialtilfelle

Når polene er reelle og forskjellige ($\zeta > 0$) har vi følgende viktige spesialtilfelle:

$$H(s) = \frac{Kp_1p_2}{(s-p_1)(s-p_2)} = \frac{K}{(T_1s+1)(T_2s+1)}$$

Der p_1 og p_2 er systemets poler og T_1 og T_2 er systemets tidskonstanter.

Dette kan sees på som to 1.ordens systemer i serie:

$$H(s) = H_1(s)H_2(s) = \frac{K}{(T_1s+1)} \cdot \frac{1}{(T_2s+1)} = \frac{K}{(T_1s+1)(T_2s+1)}$$

7.6 Eksempel: Masse-Fjær-demper system

Gitt følgende system:

$$m\ddot{x} = F - d\dot{x} - kx$$

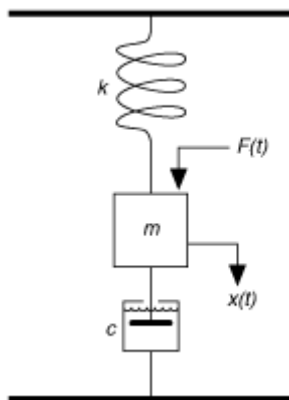
x er posisjonen

\dot{x} er hastigheten

\ddot{x} er akselerasjonen

F er kraften (pådragssignalet, u)

d og k er konstanter



Transferfunksjonen blir:

$$H(s) = \frac{x(s)}{u(s)} = \frac{\frac{1}{m}}{s^2 + \frac{d}{m}s + \frac{k}{m}}$$

(Bruk Laplace for å utlede denne)

MathScript kode:

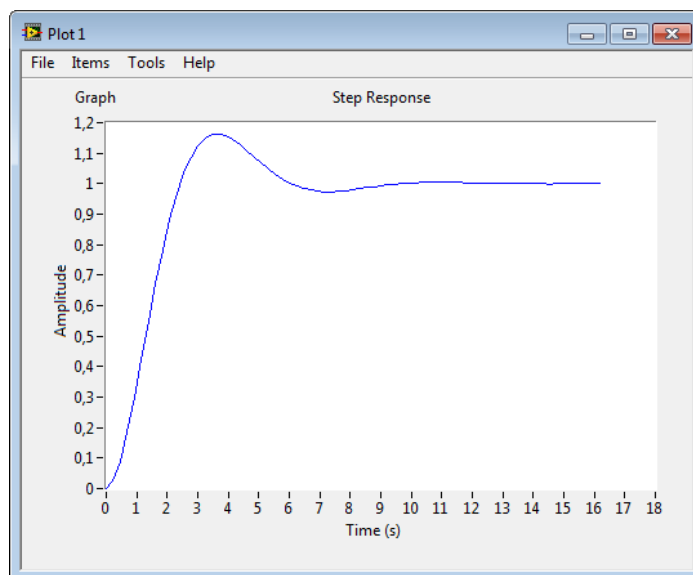
```
% Mass-spring-damper system
clear
clc

% Define variables
m = 1;
d = 1;
k = 1;

% Define Transfer function
num = 1/m ;
den = [1, (d/m), (k/m)];
H = tf(num, den);

% Step Response
step(H)
```

Dette gir følgende resultat:

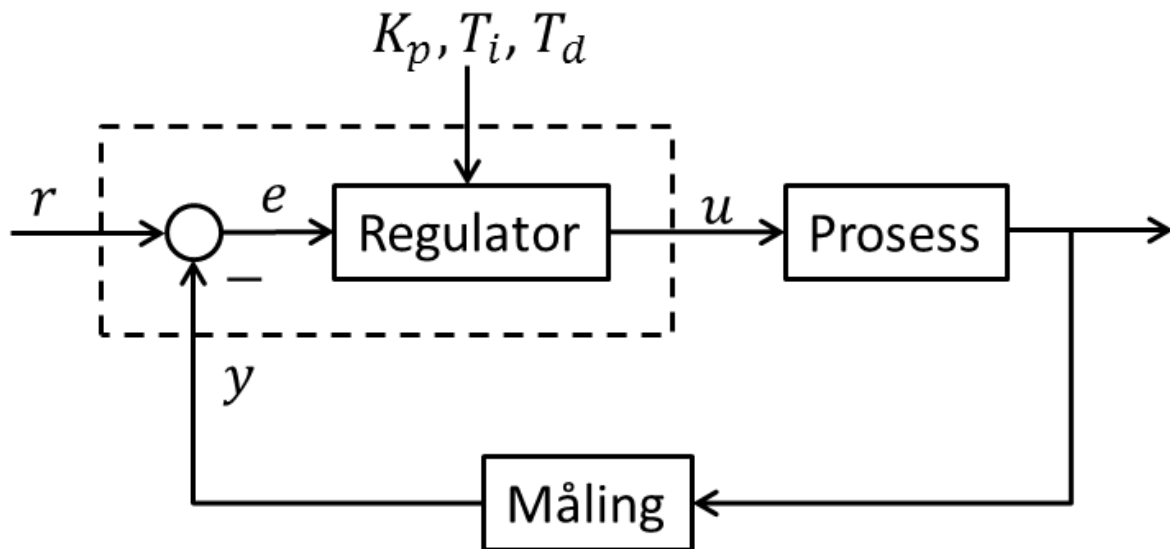


Prøv med forskjellige verdier på m , d og k .

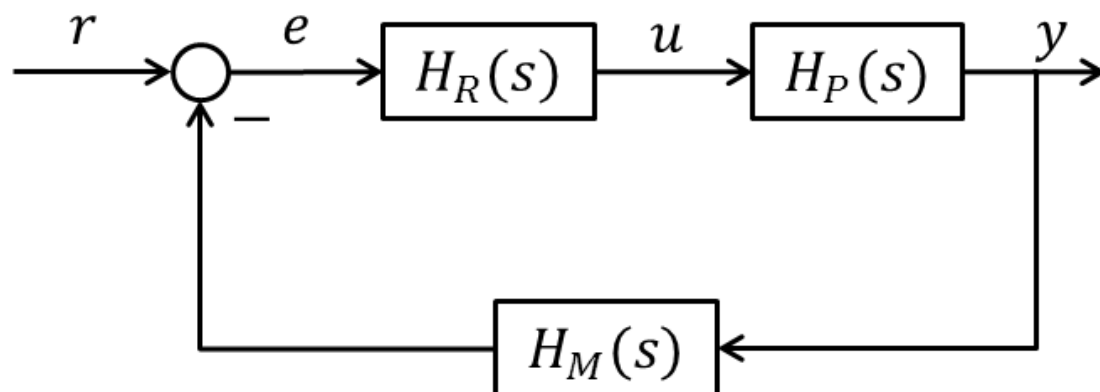
8 Blokkdiagrammer

Et reguleringsystem består av ulike blokker (transferfunksjoner), disse er enten koblet i serie eller i parallell. I reguleringsystemer er det også vanlig å bruke tilbakekoblinger i form av at regulatoren bruker målingen til å beregne pådragssignalet.

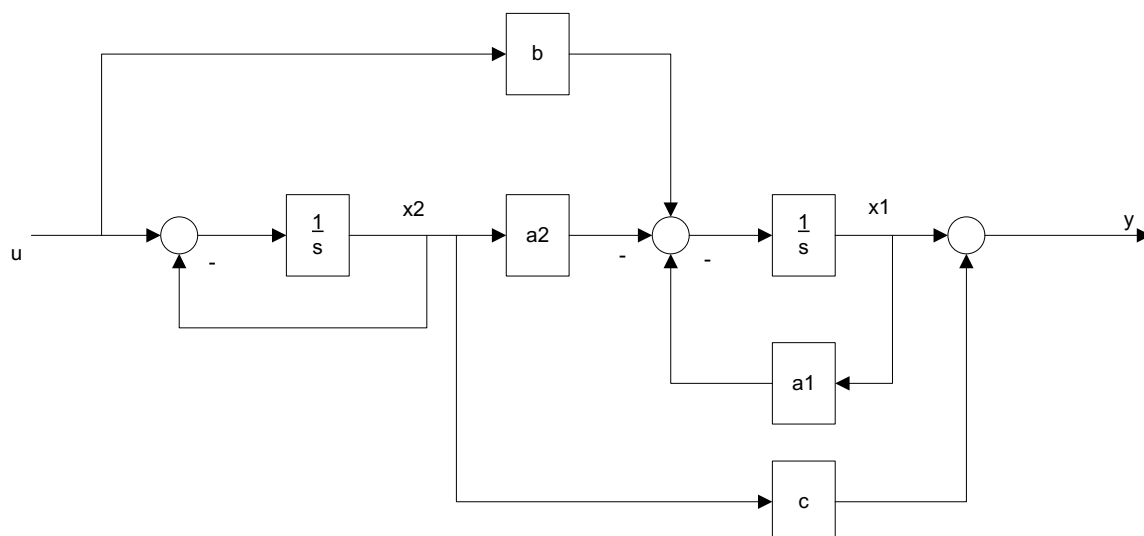
Nedenfor ser vi et typisk blokkdiagram for et reguleringsystem:



Hver av disse blokkene (Regulator, Prosess og Måling) vil ha sin egen transferfunksjon.

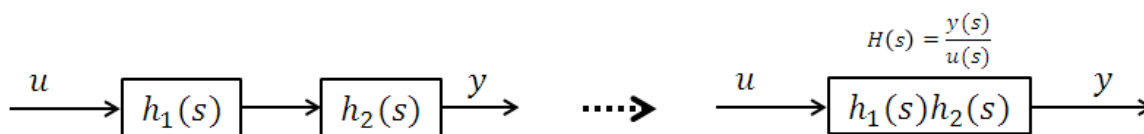


Nedenfor ser vi et eksempel på et typisk blokkdiagram som er basert på differensiallikninger:



8.1 Serie

Seriekobling:



MathScript:

I MathScript kan vi bruke **series** funksjonen:

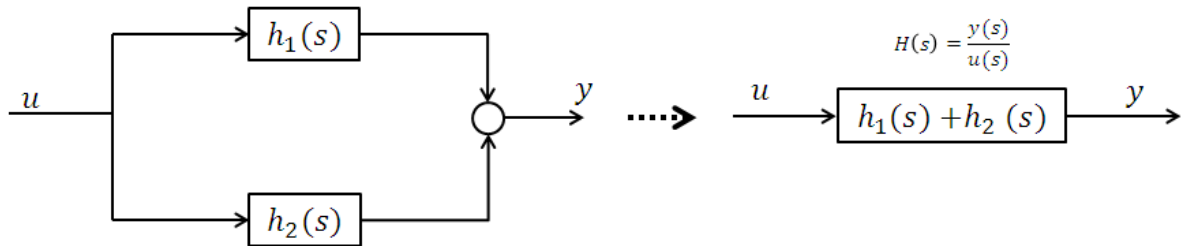
```
...
H = series(h1, h2)
```

Hvis vi har mer enn 2 kan vi gjøre følgende:

```
...
H = series(h1, series(h2, h3))
```

8.2 Parallell

Parallellkobling:

**MathScript:**

I MathScript kan vi bruke **parallel** funksjonen:

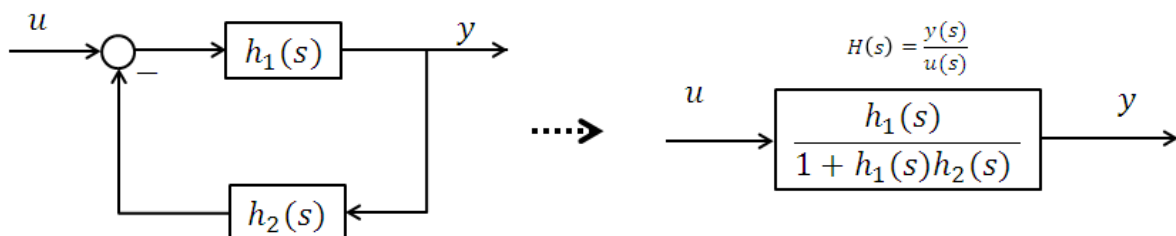
```
...
H = parallel(h1, h2)
```

Hvis vi har mer enn 2 kan vi gjøre følgende:

```
...
H = parallel(h1, parallel(h2, h3))
```

8.3 Tilbakekobling (Feedback)

Negativ tilbakekobling:

**MathScript:**

I MathScript kan vi bruke **feedback** funksjonen:

```
...
H = feedback(h1, h2)
```

8.4 1.ordens system

Transferfunksjon for 1.ordens system:

$$H(s) = \frac{K}{Ts + 1}$$

Hvor

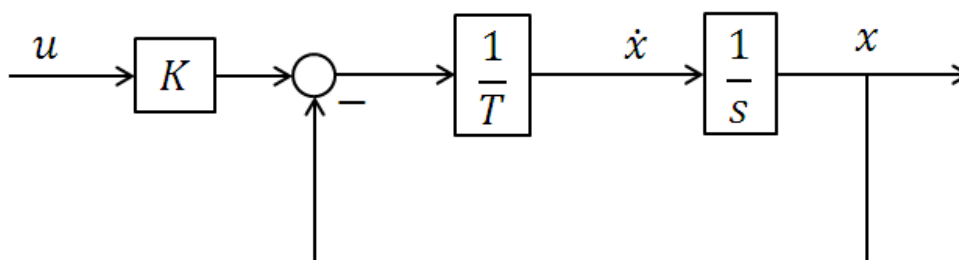
K er forsterkning

T er tidskonstant

Ved bruk av Invers Laplace får vi følgende differensiallikning:

$$\dot{x} = \frac{1}{T}(-x + Ku)$$

Blokkdiagram for systemet kan tegnes som følger:



8.4.1 1.ordens prosess med dødtid

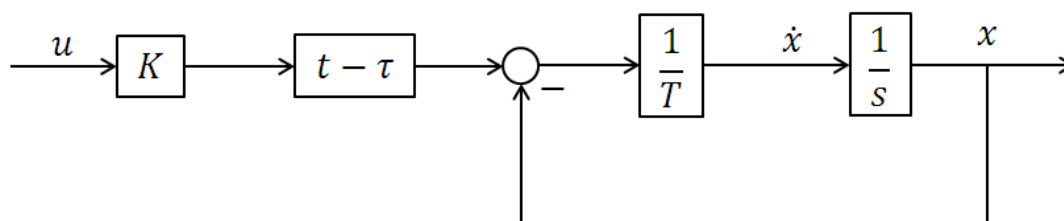
Transferfunksjon for 1.ordens system med dødtid:

$$H(s) = \frac{K}{Ts + 1} e^{-\tau s}$$

Ved bruk av Invers Laplace får vi følgende differensiallikning:

$$\dot{x} = \frac{1}{T}(-x + Ku(t - \tau))$$

Blokkdiagram for systemet kan tegnes som følger:



8.5 2.ordens system

Eksempel: Masse-Fjær-demper system

Gitt følgende system:

$$m\ddot{x} = F - d\dot{x} - kx$$

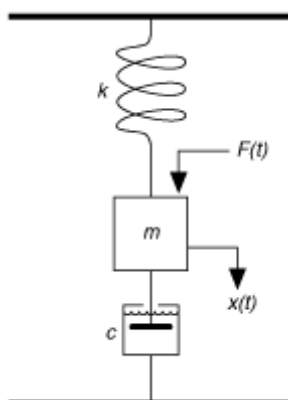
x er posisjonen

\dot{x} er hastigheten

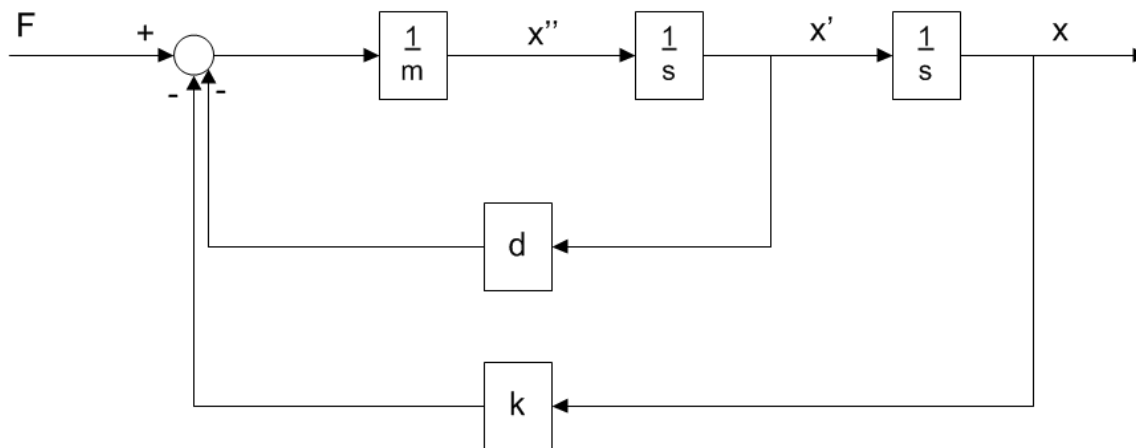
\ddot{x} er akselerasjonen

F er kraften (pådragssignalet, u)

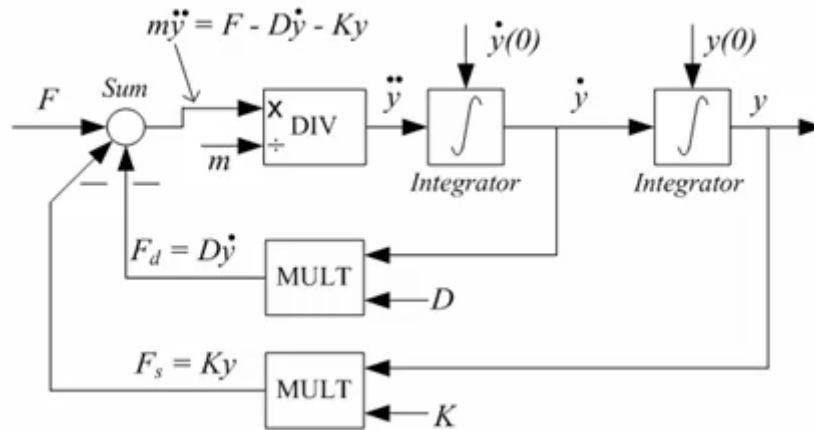
d and k er konstanter



Blokkdiagrammet blir som følger:



Her er en litt annen notasjon brukt:



[Figure: F. Haugen, Advanced Dynamics and Control: TechTeach, 2010]

Vi finner transferfunksjonen vha blokkdiagrammet:

Vi bruker serie og feedback reglene.

Vi begynner med serie-regelen:

$$h_1(s) = \frac{1}{m} \cdot \frac{1}{s} = \frac{1}{ms}$$

Deretter feedback-regelen:

$$h_2(s) = \frac{\frac{1}{ms}}{1 + \frac{1}{ms} \cdot d} = \frac{1}{ms + d}$$

Deretter serie-regelen igjen:

$$h_3(s) = \frac{1}{ms + d} \cdot \frac{1}{s} = \frac{1}{s(ms + d)}$$

Tilslutt bruker vi feedback-regelen:

$$H(s) = \frac{x(s)}{u(s)} = \frac{\frac{1}{s(ms + d)}}{1 + \frac{1}{s(ms + d)} \cdot k} = \frac{1}{s(ms + d) + k} = \frac{1}{ms^2 + ds + k}$$

På standardform får vi følgende:

$$H(s) = \frac{K}{\left(\frac{s}{\omega_0}\right)^2 + 2\zeta \frac{s}{\omega_0} + 1} = \frac{K\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$$

Transferfunksjonen blir:

$$H(s) = \frac{x(s)}{u(s)} = \frac{\frac{1}{m}}{s^2 + \frac{d}{m}s + \frac{k}{m}}$$

Det betyr:

$$\omega_0^2 = \frac{k}{m} \rightarrow \omega_0 = \sqrt{\frac{k}{m}}$$

$$K\omega_0^2 = \frac{1}{m} \rightarrow K \frac{k}{m} = \frac{1}{m} \rightarrow K = \frac{1}{k}$$

$$2\zeta\omega_0 = \frac{d}{m} \rightarrow \zeta = \frac{d}{2\omega_0 m} \rightarrow \zeta = \frac{d}{2\sqrt{mk}}$$

MathScript kode:

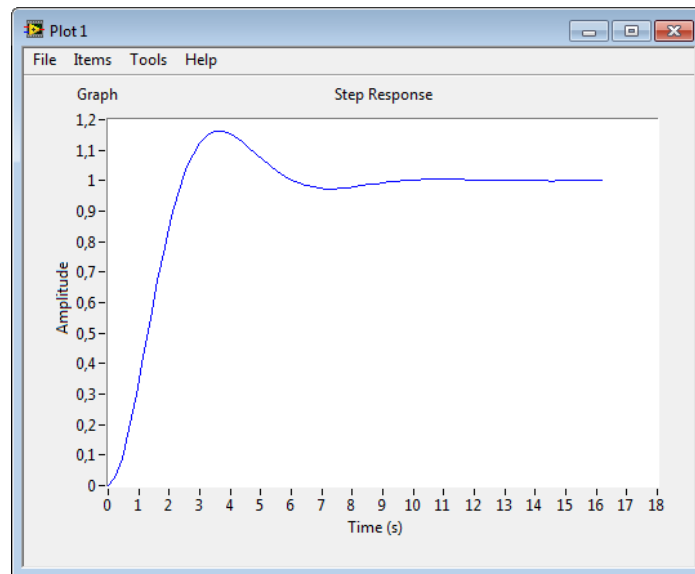
```
% Mass-spring-damper system
clear
clc

% Define variables
m = 1;
d = 1;
k = 1;

% Define Transfer function
num = 1/m ;
den = [1, (d/m), (k/m)];
H = tf(num, den);

% Step Response
step(H)
```

Dette gir følgende resultat:



Prøv med forskjellige verdier på m , d og k .

9 Tilstandsrom- modeller

En tilstandsrommodell er en strukturert måte å representere et sett med differensiallikninger på. Tilstandsmodeller er veldig nyttige i forbindelse med reguleringsteori. Differensiallikningene blir satt på matrise-vektor form, som jo er basis-elementene i MathScript.

Anta følgende differensiallikninger:

$$\begin{aligned} \dot{x}_1 &= a_{11}x_1 + a_{21}x_2 + \dots + a_{n1}x_n + b_{11}u_1 + b_{21}u_2 + \dots + b_{n1}u_n \\ &\vdots \\ \dot{x}_n &= a_{1n}x_1 + a_{2n}x_2 + \dots + a_{nn}x_n + b_{1n}u_1 + b_{2n}u_2 + \dots + b_{nn}u_n \\ &\vdots \end{aligned}$$

Disse kan skrives på følgende generelle form (tilstandsrommodell):

$$\begin{aligned} \underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix}}_{\dot{x}} &= \underbrace{\begin{bmatrix} a_{11} & \dots & a_{n1} \\ \vdots & \ddots & \vdots \\ a_{1m} & \dots & a_{nm} \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_x + \underbrace{\begin{bmatrix} b_{11} & \dots & b_{n1} \\ \vdots & \ddots & \vdots \\ b_{1m} & \dots & b_{nm} \end{bmatrix}}_B \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}}_u \\ \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_y &= \underbrace{\begin{bmatrix} c_{11} & \dots & c_{n1} \\ \vdots & \ddots & \vdots \\ c_{1m} & \dots & c_{nm} \end{bmatrix}}_C \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_x + \underbrace{\begin{bmatrix} d_{11} & \dots & d_{n1} \\ \vdots & \ddots & \vdots \\ d_{1m} & \dots & d_{nm} \end{bmatrix}}_D \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}}_u \end{aligned}$$

Eller på følgende enkle form:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

Der x – systemets interne tilstander, u – pådrag, y - måling

Eksempel:

Gitt følgende differensiallikninger:

$$\begin{aligned} \dot{x}_1 &= x_1 + 2x_2 \\ \dot{x}_2 &= 3x_1 + 4x_2 + u \end{aligned}$$

$$y = x_1$$

Dette gir:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}}_A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_B u$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_C \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Merk! D matrisa er lik null i dette tilfellet, så dermed tar vi ikke med den.

I MathScript gjør vi følgende:

```
A = [1 2; 3 4];
B = [0; 1];
C = [1, 0];
D = [0];
model = ss(A, B, C, D)
```

Sprangrespons:

```
step(model)
```

Implementer dette i MathScript og se hva resultatet blir.

[Slutt på eksempel]

Eksempel:

Gitt følgende system:

$$\dot{x}_1 = x_1 + u$$

$$\dot{x}_2 = x_2$$

$$\dot{x}_3 = x_3$$

$$y_1 = x_3$$

$$y_2 = x_1 + 2x_2$$

Vi ønsker å sette systemet på følgende form:

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

Tilstandsmodellen blir som følger:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_A \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}}_B u$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 1 & 2 & 0 \end{bmatrix}}_C \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

[Slutt på eksempel]

10 Tidsforsinkelse og Pade' Approximasjon

Eksempler på dødtid/tidsforsinkelser i prosess -sammenheng:

- Transportbånd
- Rør, f.eks Luftrør (f.eks Air Heateren som vi har brukt som eksempel tidligere)



Transferfunksjon for tidsforsinkelse/dødtid:

$$H(s) = \frac{y(s)}{u(s)} = e^{-\tau s}$$

→ Transferfunksjonen $e^{-\tau s}$ er såkalt ikke-rasjonal, dvs. den har ikke teller og nevnerpolynom med potenser av s .

→ Ulike metoder for analyse og design av reguleringsystemer krever at systemet er en rasjonal transferfunksjon

Tidsplan:

$$y(t) = u(t - \tau)$$

10.1 Pade' Approsimasjon

Følgende approksimasjon brukes:

$$e^{-\tau s} \approx \frac{1 - k_1 s + k_2 s^2 + \dots \pm k_n s^n}{1 + k_1 s + k_2 s^2 + \dots + k_n s^n}$$

Hvor n er approksimasjonens orden.

k_1, k_2, \dots er konstanter

→ Jo høyere orden, jo bedre tilnærming til virkeligheten, men ulempen er at transferfunksjonen blir veldig kompleks! Så en gylden middelvei må velges

→ For 1.ordens og 2.ordens approksimasjoner er disse konstantene gitt ved:

$n = 1$	$n = 2$
$k_1 = \frac{\tau}{2}, \text{ other } k_i = 0$	$k_1 = \frac{\tau}{2}, k_2 = \frac{\tau^2}{12}, \text{ other } k_i = 0$

1.ordens Pade approksimasjon:

$$e^{-\tau s} \approx \frac{1 - k_1 s}{1 + k_1 s}$$

Hvor

$$k_1 = \frac{\tau}{2}$$

2.ordens Pade approksimasjon:

$$e^{-\tau s} \approx \frac{1 - k_1 s + k_2 s^2}{1 + k_1 s + k_2 s^2}$$

Hvor:

$$k_1 = \frac{\tau}{2}$$

$$k_2 = \frac{\tau^2}{12}$$

10.2 MathScript

Implementering av dødtid/Tidsforsinkelse.

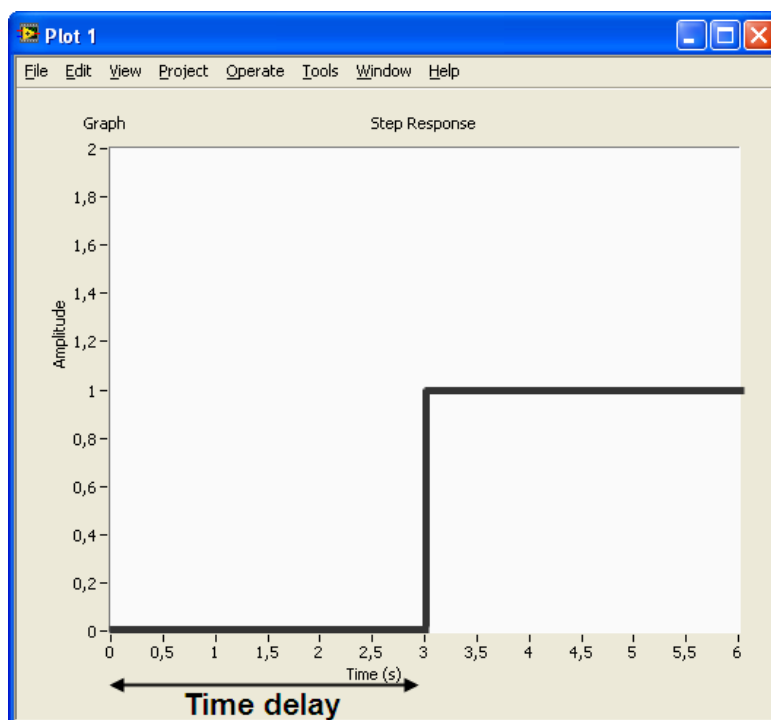
Eksempel:

$$H(s) = e^{-3s}$$

Dvs. tidsforsinkelsen er på 3 sekunder.

Sprangrespons:

I dette eksemplet viser vi ulike måter å implementere tidsforsinkelse på. For en tidsforsinkelse på 3 sekunder får vi følgende sprangrespons:



10.2.1 Metode 1

→ Vi bruker den innebygde funksjonen `sys_order1`

```
K = 1;  
T = 0;  
delay = 3  
H = sys_order1(K, T, delay)  
step(H)
```

10.2.2 Metode 2

→ Vi bruker den innebygde `pade` funksjonen

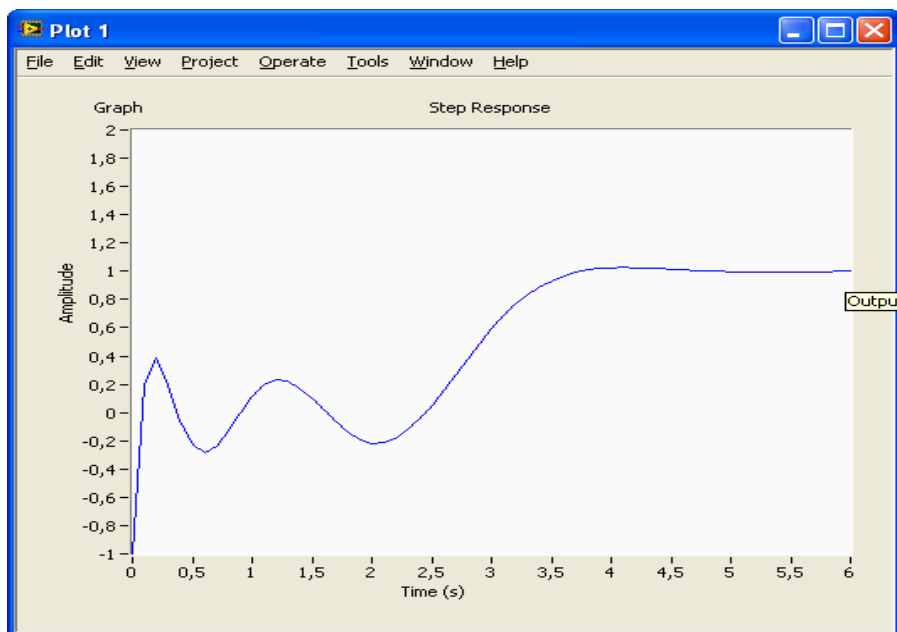
Eksempel på en 2.ordens approksimasjon:

```

delay = 3;
order = 2;
H = pade(delay,order)
step(H)

```

Nedenfor ser vi sprangresponsen til en 5.ordens approksimasjon:



10.2.3 Metode 3

→ Vi bruker uttrykkene for Pade appr. og bruker **tf** funksjonen

Eksempel på en 1.ordens approksimasjon:

$$e^{-\tau s} \approx \frac{1 - k_1 s}{1 + k_1 s}$$

```

k1 = delay/2;
num = [-k1, 1];
den = [k1, 1];

H = tf(num,den)
step(H)

```

→ Det er slik den innebygde funksjonen **pade** er laget!

Merk! Pass på rekkefølgen på koeffisientene i telleren og nevneren (høyeste orden til venstre og deretter i synkende rekkefølge)!

$$e^{-\tau s} \approx \frac{1 - k_1 s}{1 + k_1 s} = \frac{-k_1 s + 1}{k_1 s + 1}$$

11 Stabilitets -analyse

3 typer systemer (stabilitetsegenskaper):

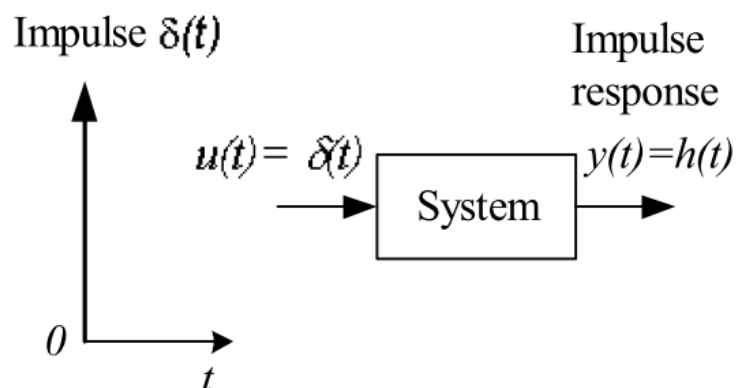
- Asymptotisk stabilt system
- Marginalt stabilt system
- Ustabilt system

Metoder for å finne stabilitetsegenskapene:

- Impulsrespons
- Polplassering

11.1 Impulsrespons

Vi kan definere et systems stabilitetsegenskaper utfra impulsresponsen [F. Haugen, Advanced Dynamics and Control: TechTeach, 2010]:



11.1.1 Asymptotisk stabilt system

Impulsrespons:

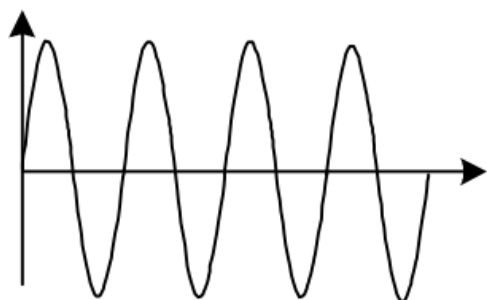


Stasjonær Impulsrespons:

$$\lim_{t \rightarrow \infty} h(t) = 0$$

11.1.2 Marginalt stabilt system

Impulsrespons:

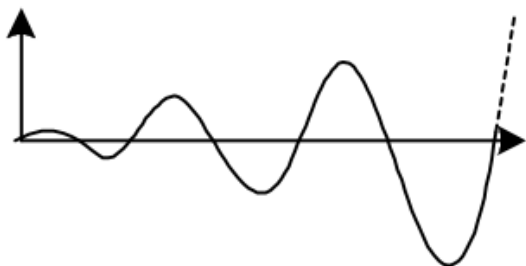


Stasjonær Impulsrespons:

$$0 < \lim_{t \rightarrow \infty} h(t) < \infty$$

11.1.3 Ustabilt system

Impulsrespons:



Stasjonær Impulsrespons:

$$\lim_{t \rightarrow \infty} h(t) = \infty$$

11.1.4 MathScript

Eksempel:

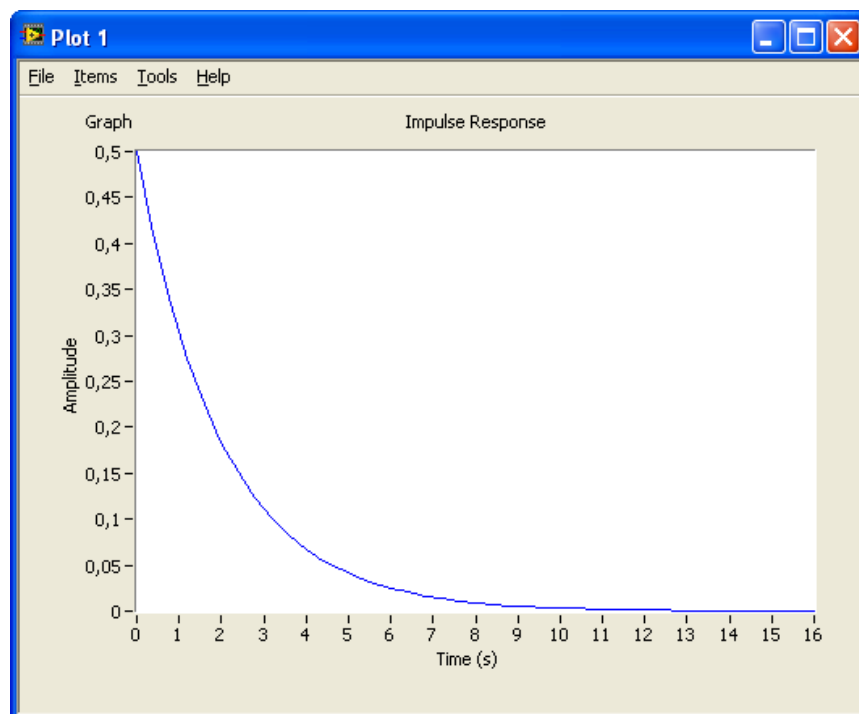
$$H(s) = \frac{1}{2s + 1}$$

Impulsrespons:

```
K=1;
T=2;
num = [K];
den = [T, 1];
H = tf(num, den)
```

`impulse` (H)

Dette gir følgende impulsrespons:



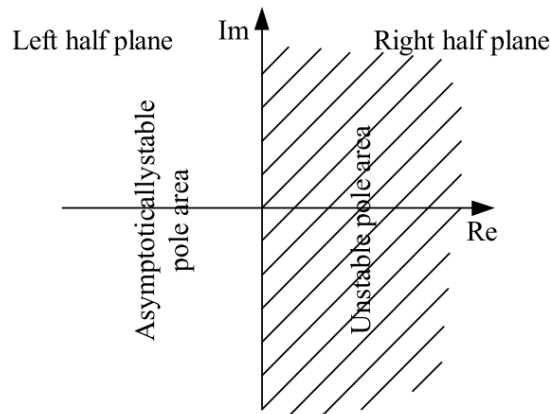
→ Asymptotisk stabilt system

11.2 Poler

Det kan være upraktisk å måtte jobbe med impulsresponser ved stabilitetsundersøkelser, så en bedre metode vil være å bruke systemets poler.

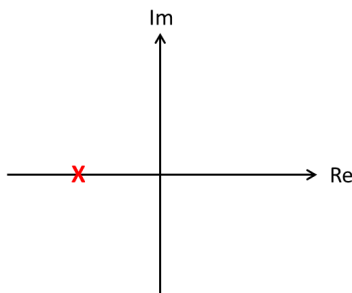
Polene til et gitt system finnes fra nevnerpolynomet i transferfunksjonen.

Det komplekse plan:



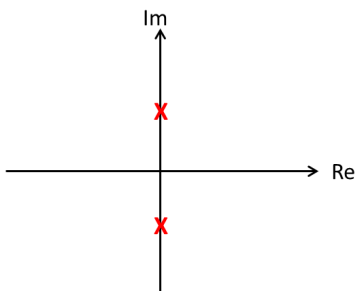
[Figure: F. Haugen, Advanced Dynamics and Control: TechTeach, 2010]

11.2.1 Asymptotisk stabilt system



Alle polene ligger i venstre halvplan (negativ realdel).
Ingen poler på den imaginære akse.

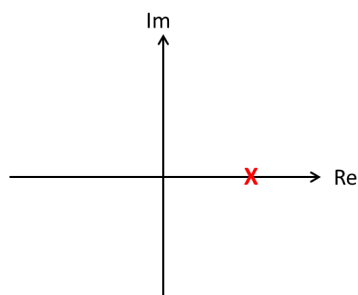
11.2.2 Marginalt stabilt system



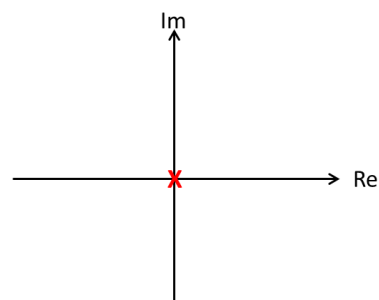
En eller flere poler ligger på den imaginære akse (har realdelen lik 0), og alle polene er forskjellige/ikke sammenfallende.

Dessuten, ingen poler i høyre halvplan

11.2.3 Ustabilt system



En eller flere poler ligger i høyre halvplan (har realdel større enn 0).



Eller: Det er multiple/sammenfallende poler på den imaginære akse.

F.eks: Dobbeltintegrator $H = \frac{1}{s^2}$

11.2.4 MathScript

Eksempel:

$$H(s) = \frac{1}{2s + 1}$$

Poler:

```
K=1;
T=2;
num = [K];
den = [T, 1];
H = tf(num, den)
P = poles(H)

pzgraph(H)
```

→ $p = -0.5$ → Asymptotisk stabilt system

[Slutt på eksempel]

Eksempel: Masse-fjær-demper system

Gitt følgende system:

$$m\ddot{x} = F - d\dot{x} - kx$$

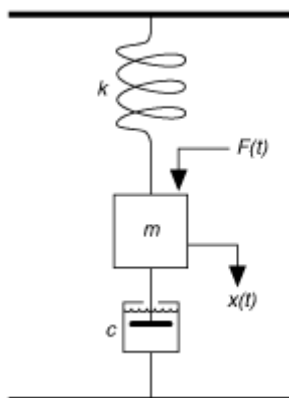
x er posisjonen

\dot{x} er hastigheten

\ddot{x} er akselerasjonen

F er kraften (pådragssignalet, u)

d and k er konstanter



Transferfunksjonen blir:

$$H(s) = \frac{x(s)}{u(s)} = \frac{1}{s^2 + \frac{d}{m}s + \frac{k}{m}}$$

Som gir følgende:

$$\omega_0^2 = \frac{k}{m} \rightarrow \omega_0 = \sqrt{\frac{k}{m}}$$

$$K\omega_0^2 = \frac{1}{m} \rightarrow K \frac{k}{m} = \frac{1}{m} \rightarrow K = \frac{1}{k}$$

$$2\zeta\omega_0 = \frac{d}{m} \rightarrow \zeta = \frac{d}{2\omega_0 m} \rightarrow \zeta = \frac{d}{2\sqrt{mk}}$$

MathScript kode:

```
% Mass-spring-damper system
clear
clc

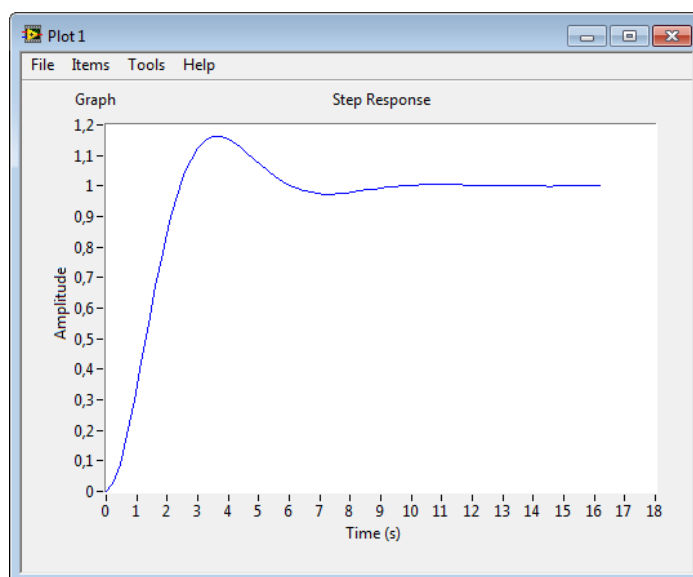
% Define variables
m = 1;
d = 1;
k = 1;
```

```
% Define Transfer function
num = 1/m ;
den = [1, (d/m), (k/m)];
H = tf(num, den);

% Step Response
step(H)


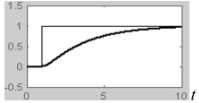

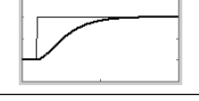



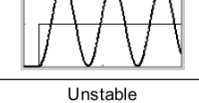

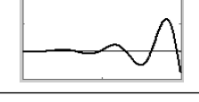
% Stability Analysis
p = poles(H)
z = zero(H)
pzmap(H)
```

Dette gir følgende resultat:



Prøv med forskjellige verdier på m , d og k og sjekk stabilitetsegenskapene.

For 2.ordens systemer har vi som kjent følgende:

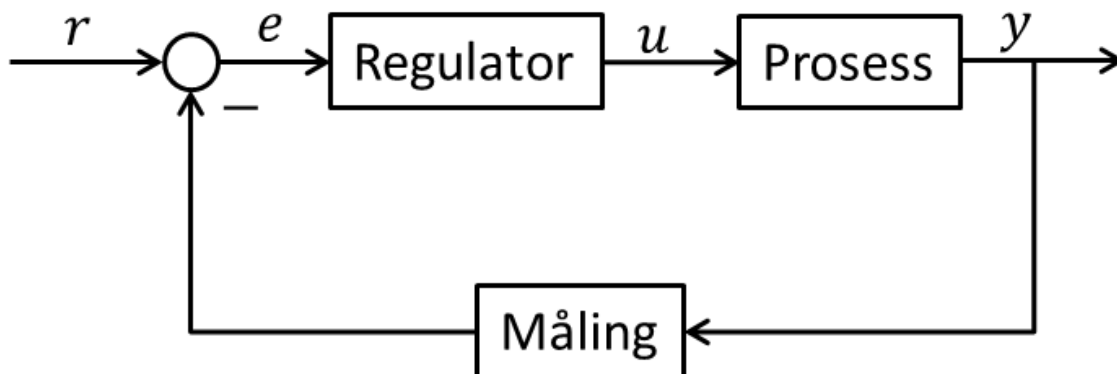
Value of ζ	Poles p_1 and p_2	Type of step response $y(t)$
$\zeta > 1$	Real and distinct 	Overdamped 
$\zeta = 1$	Real and multiple 	Critically damped 
$0 < \zeta < 1$	Complex conj. 	Underdamped 
$\zeta = 0$	Imaginary 	Undamped 
$\zeta < 0$	Pos. real part 	Unstable 

[Figure: F. Haugen, Advanced Dynamics and Control: TechTeach, 2010]

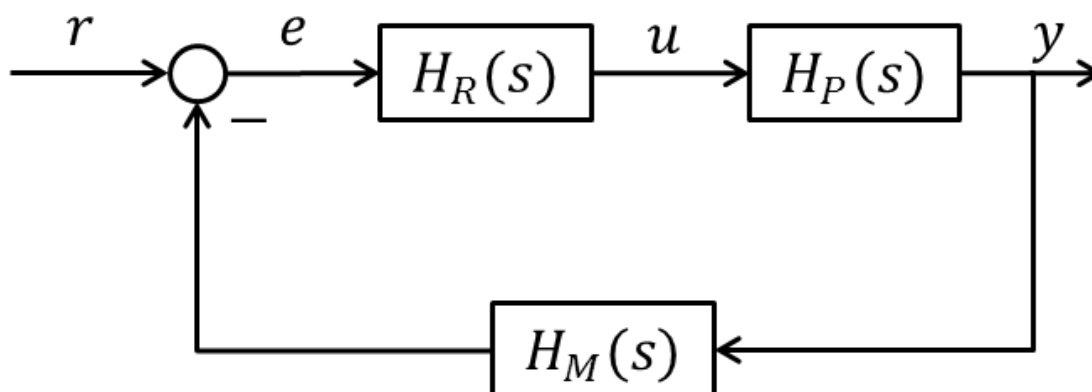
Klarer du å gjenskape disse?

11.3 Tilbakekoblede systemer

Skisse av et tilbakekoblede system:



Hver av disse kan beskrives vha. en transfer funksjon:



11.3.1 Sløyfetransferfunksjonen

(Engelsk: "Loop transfer function")

$$L(s) = H_R H_P H_M$$

11.3.2 Følgeforholdet

(Engelsk: "Tracking transfer function")

$$T(s) = \frac{y(s)}{r(s)} = \frac{H_R H_P H_M}{1 + H_R H_P H_M} = \frac{L(s)}{1 + L(s)}$$

Denne uttrykker hvor godt systemet følger referansen. Systemet har gode følgeegenskaper hvis $y \approx r$, dvs.:

$$|T| \approx 1$$

11.3.3 Sensitivitetsfunksjonen/Avviksforholdet

(Engelsk: "Sensitivity transfer function")

$$S(s) = \frac{e(s)}{r(s)} = \frac{1}{1 + L(s)} = 1 - T(s)$$

Denne uttrykker hvor "sensitivt" avviket er overfor referansen og denne bør derfor være "liten", dvs:

$$|S| \approx 0 \text{ or } |S| \ll 1$$

Merk!

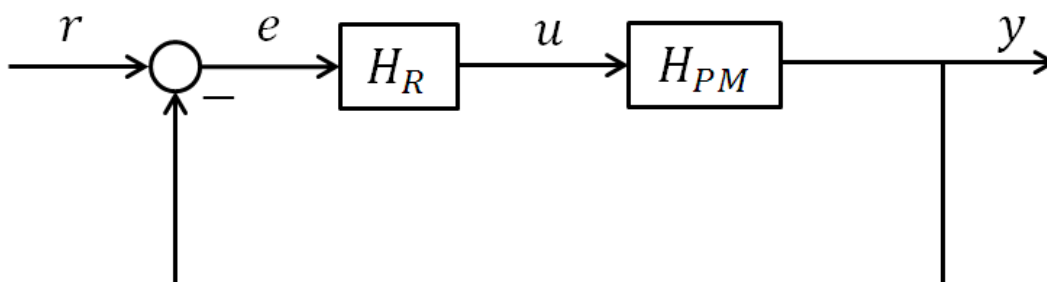
$$S(s) = 1 - T(s) \leftrightarrow T(s) = 1 - S(s)$$

og

$$T(s) + S(s) = \frac{L(s)}{1 + L(s)} + \frac{1}{1 + L(s)} \equiv 1$$

11.3.4 MathScript

Eksempel:



Sløyfetransferfunksjonen:

`L = series(Hr, Hpm)`

Hvis flere enn 2: `M=series(H1, series(H2, H3))`

Følgeforholdet:

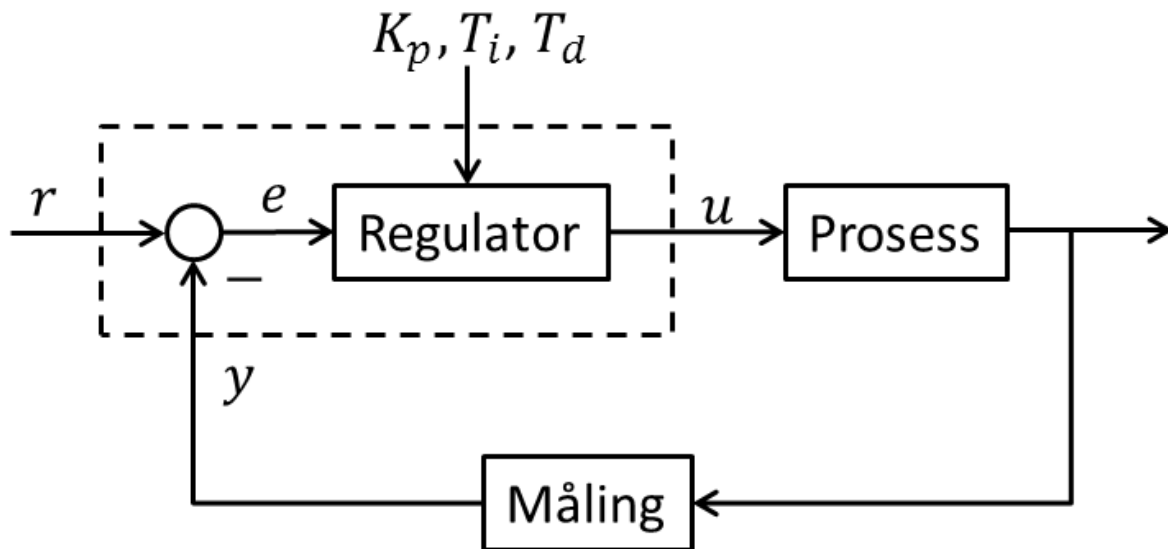
$$T = \text{feedback}(L, 1)$$

Sensitivitetsfunksjonen/Avviksforholdet

$$S = 1 - T$$

12 PID regulering

Oppgaven til en regulator er å endre pådraget i forhold til måleverdien fra prosessen, det gjør den med noen parameterer som kalles P, I og D (PID regulator). Disse variablene varierer fra prosess til prosess. Formålet til disse variablene er å skape den optimale reguleringsalgoritmen til en viss prosess, for å få en mest mulig stabil reguleringen, dvs at utgangen y følger referansen r best mulig.



En PID regulator er gitt ved:

$$u(t) = K_p e + \frac{K_p}{T_i} \int_0^t e d\tau + K_p T_d \dot{e}$$

Der e er avviket mellom referansen r og utgangen y ($e = r - y$), mens u er pådraget.

P-ledd (Proporsjonal):

$$u_p(t) = K_p e$$

Der K_p er (proporsjonal) forsterkningen

I-ledd (Integral):

$$u_i(t) = \frac{K_p}{T_i} \int_0^t e d\tau$$

Der T_i er integraltiden

→ I-leddet sørger for at regulatoren gir null avvik (stasjonært) (Statisk ytelse)

D-ledd (Deriverte):

$$u_D(t) = K_p T_d \dot{e}$$

Der T_d er derivattiden

→ D-leddet sørger for at regulatoren reagerer raskt (Dynamisk ytelse)

Transferfunksjon:

Transferfunksjonen for en PID regulator:

$$H_{pid}(s) = \frac{u(s)}{e(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right)$$

eller:

$$H_{pid}(s) = \frac{u(s)}{e(s)} = K_p + \frac{K_p}{T_i s} + K_p T_d s$$

Eller på felles brøk:

$$H_{pid}(s) = \frac{u(s)}{e(s)} = \frac{K_p (T_i T_d s^2 + T_i s + 1)}{T_i s}$$

Merk! Følgene laplace transformasjoner er brukt:

$$\dot{x} \Leftrightarrow s x(s)$$

$$\int x \Leftrightarrow \frac{1}{s} x(s)$$

12.1 PI regulator

Veldig ofte bruker man bare en PI regulator. En PI regulator er gitt ved:

$$u(t) = K_p e + \frac{K_p}{T_i} \int_0^t e d\tau$$

Så hva er transferfunksjonen til en PI regulator?

$$H_{pi}(s) = \frac{u(s)}{e(s)}$$

Vi bruker Laplace:

$$u(s) = K_p e(s) + \frac{K_p}{T_i} \cdot \frac{1}{s} e(s)$$

eller:

$$u(s) = K_p e(s) + \frac{K_p}{T_i s} e(s)$$

Transferfunksjonen blir da:

$$H_{pi}(s) = \frac{u(s)}{e(s)} = K_p + \frac{K_p}{T_i s} = \frac{K_p T_i s}{T_i s} + \frac{K_p}{T_i s} = \frac{K_p T_i s + K_p}{T_i s} = \frac{K_p (T_i s + 1)}{T_i s}$$

Dette gir:

$$H_{pi}(s) = \frac{u(s)}{e(s)} = \frac{K_p (T_i s + 1)}{T_i s}$$

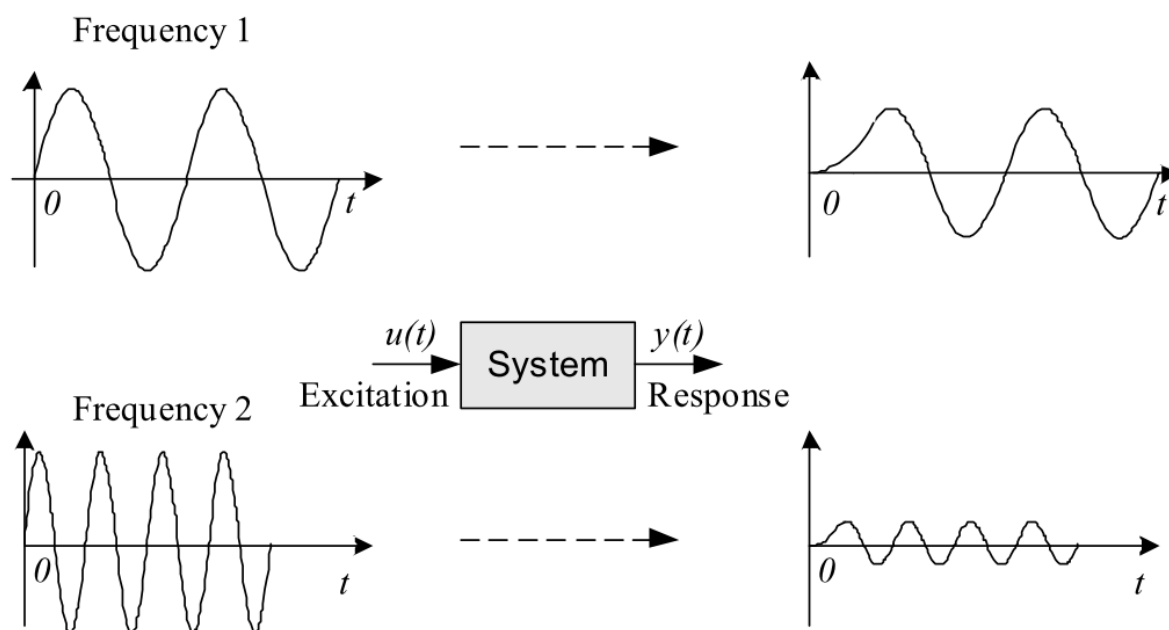
12.2 Sprangrespons

Implementer sprangresponsen (enhetssprang i e) til en P-regulator, PI-regulator, PD-regulator og en PID-regulator vha MathScript.

Er responsen som forventet?

13 Frekvensrespons

Et systems frekvensrespons uttrykker hvordan sinussignaler på systemets inngang endres gjennom systemet. Som vi ser av figuren under så vil både amplituden forandres, samt at signalet blir faseforskyvet.



[Figure: F. Haugen, Advanced Dynamics and Control: TechTeach, 2010]

Frekvensresponsen forteller hvilken amplitudeforsterkning og faseforskyvning den enkelte frekvenskomponent får gjennom systemet.

Frekvensresponsen til et system er definert som steady-state responsen til systemet hvor inngangssignalet er et sinus signal.

I steady-state vil utgangssignalet være forskjellig mtp amplitude/forsterkning (A) samt være faseforskyvet (ϕ).

Vi kan definere **inngangssignalet** som:

$$u(t) = U \sin \omega t$$

Utgangssignalet i steady-state vil da være:

$$y(t) = \underbrace{UA}_Y \sin(\omega t + \phi)$$

Hvor $A = \frac{Y}{U}$ er forholdet mellom utgangssignalets og inngangssignalets amplitude (i steady-state).

A og ϕ er funksjoner av frekvensen ω , så vi kan skrive $A = A(\omega)$, $\phi = \phi(\omega)$

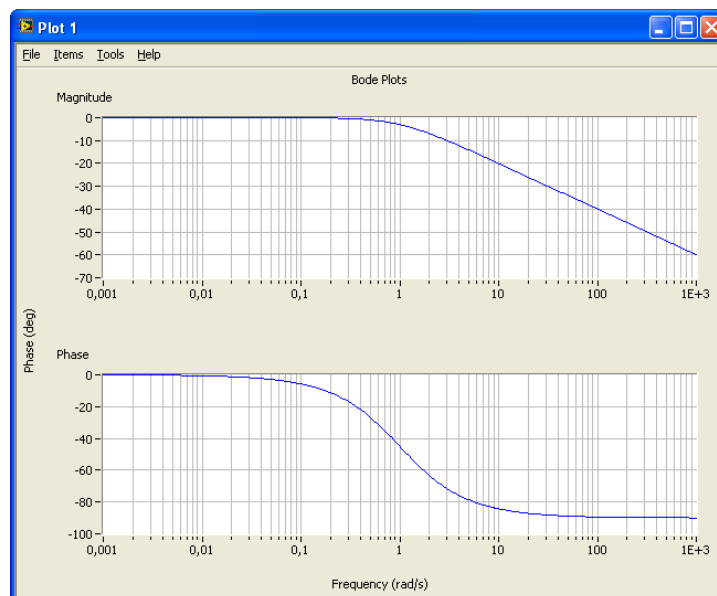
- $A(\omega)$ - Amplitdefunksjonen
- $\phi(\omega)$ – Fasefunksjonen

13.1 Bodediagram

Frekvensresponsen kan presenteres grafisk i et Bodediagram.

Bodediagrammet består av 2 deler:

- Et Amplitudediagram (Forsterkning [dB] som funksjon av frekvens [ω])
- Et Fasediagram (Faseforskyvning [$grader$] som funksjon av frekvens [ω])



Merk! Frekvensaksen er logaritmisk med 10-logaritmen for frekvensen som enhet

Vanligvis er enheten for frekvens Hertz [Hz], mens i frekvensrespons brukes radianer ω [rad]. Sammenhengen mellom disse er:

$$\omega = 2\pi f$$

13.1.1 MathScript

I MathScript kan vi enkelt tegne Bodediagrammer vha den innebygde funksjonen `bode`.

Eksempel:

Gitt transferfunksjon:

$$H(s) = \frac{y(s)}{u(s)} = \frac{1}{s + 1}$$

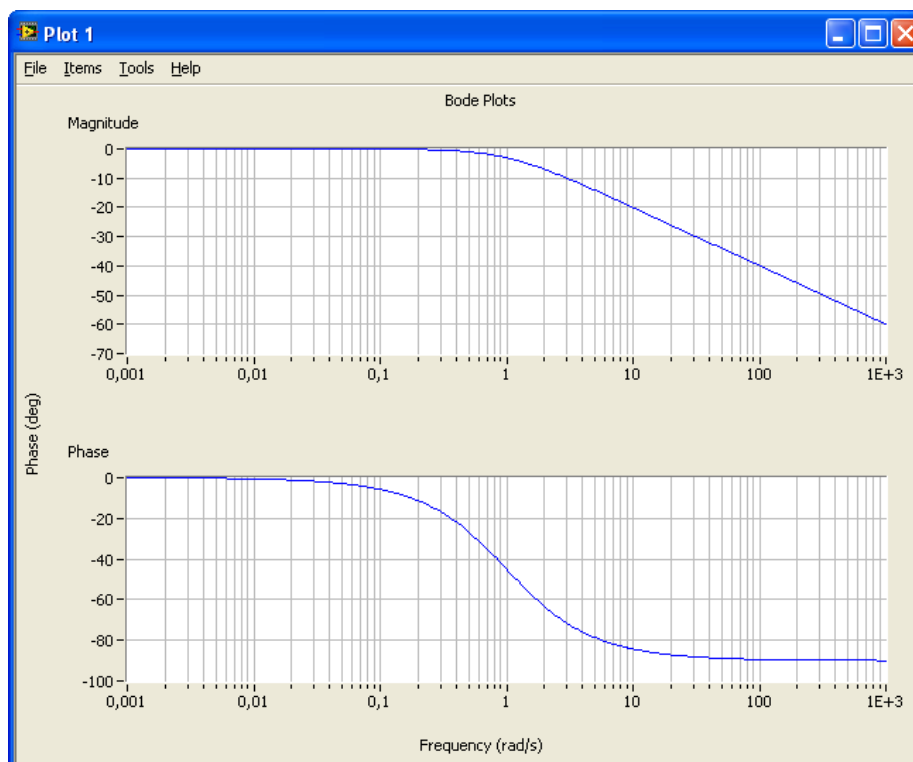
MathScript koden blir som følger:

```
K=1;
T=1;
num=[K];
den=[T, 1];
H = tf(num, den)
bode (H);
```

Det er ønskelig å ha grid på plottet:

```
subplot(2,1,1)
grid
subplot(2,1,2)
grid
```

Bode-diagrammet blir som følger:



[Slutt på eksempel]

13.2 Frekvensresponsen fra transferfunksjonen

For en generell transferfunksjon:

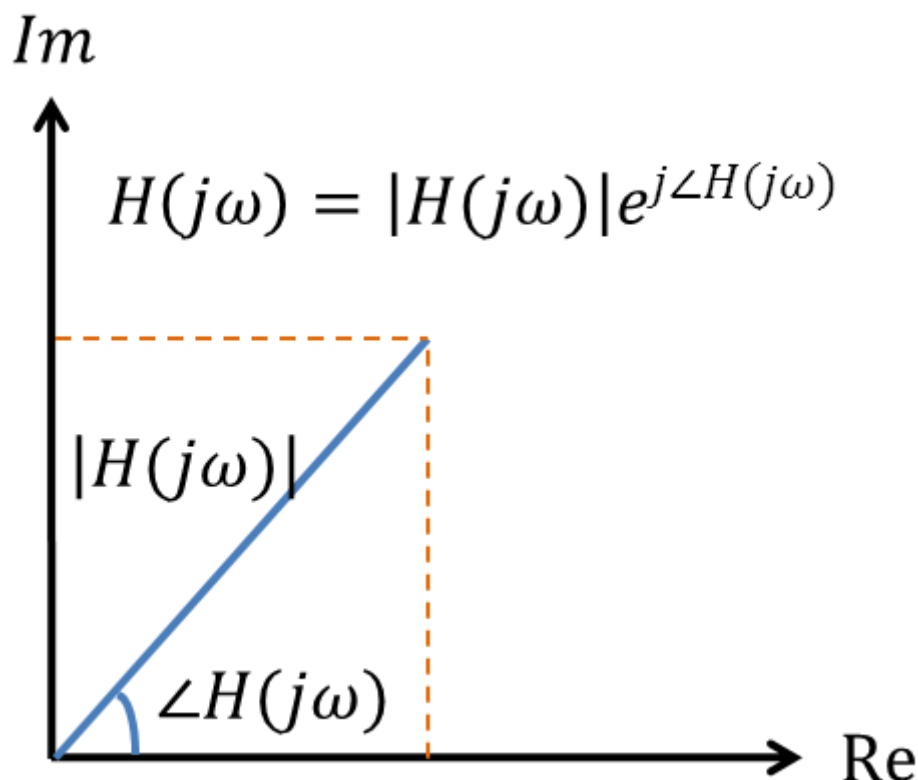
$$H(S) = \frac{y(s)}{u(s)}$$

Har vi følgende (setter $s = j\omega$):

$$H(j\omega) = |H(j\omega)|e^{j\angle H(j\omega)}$$

Hvor $H(j\omega)$ er frekvensresponsen til systemet, dvs. **Vi finner frekvensresponsen ved å sette $s = j\omega$ i transferfunksjonen.**

Dette kan illustreres i det komplekse planet slik:



Et Bodediagram viser en grafisk presentasjon av frekvensresponsen, og er nyttig ved analyse og design av reguleringsystemet. Bodediagrammet består av 2 forskjellige plot, Amplitudediagram, $A(\omega)$ and fasediagram, $\phi(\omega)$.

Amplitudedefunksjonen:

$$A(\omega) = |H(j\omega)|$$

Fasefunksjonen:

$$\phi(\omega) = \angle H(j\omega)$$

Merk! $A(\omega)$ -aksen er i desibel (dB), hvor desibelverdien av x beregnes som følger:

$$x[dB] = 20 \log_{10} x$$

Merk! $\phi(\omega)$ -aksen er i grader (ikke i radianer!)

13.2.1 Komplekse tall

Siden dette er basert på komplekse tall tar vi en liten oppsummering ifm dette.

Et kompleks tall er definert som følger:

$$z = a + jb$$

Den imaginære delen j er definert slik:

$$j = \sqrt{-1}$$

Hvor a er den reelle delen av z og b er den imaginære delen av z

$$\operatorname{Re}(z) = a, \operatorname{Im}(z) = b$$

Vi kan også bruke exponential/polar form:

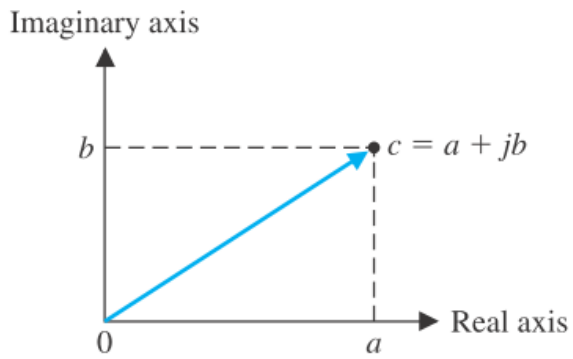
$$z = r e^{j\theta}$$

Hvor:

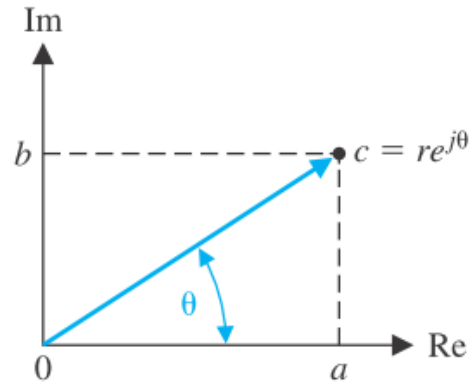
$$r = |z| = \sqrt{a^2 + b^2}$$

$$\theta = \operatorname{atan} \frac{b}{a}$$

Merk! $a = r \cos \theta$ og $b = r \sin \theta$



Komplekse tall på rektangulær form



Komplekse tall på eksponensial/polar form

[Figure: R. C. Dorf and R. H. Bishop, Modern Control Systems vol. Eleventh Edition: Pearson Prentice Hall.]

Gitt følgende komplekse tall:

$$z_1 = r_1 e^{j\theta_1} \text{ and } z_2 = r_2 e^{j\theta_2}$$

Multiplikasjon:

$$z_3 = z_1 z_2 = r_1 r_2 e^{j(\theta_1 + \theta_2)}$$

Divisjon:

$$z_3 = \frac{z_1}{z_2} = \frac{r_1 e^{j\theta_1}}{r_2 e^{j\theta_2}} = \frac{r_1}{r_2} e^{j(\theta_1 - \theta_2)}$$

Frekvensresponsen er definert som:

$$H(j\omega) = |H(j\omega)| e^{j\angle H(j\omega)}$$

Forsterkningsfunksjonen er definert som:

$$A(\omega) = |H(j\omega)|$$

Fasefunksjonen er definert som:

$$\phi(\omega) = \angle H(j\omega)$$

Eksempel

Gitt følgende 1.ordens transferfunksjon:

$$H(s) = \frac{y(s)}{u(s)} = \frac{K}{Ts + 1}$$

Frekvensresponsen (we replace s with $j\omega$) for vårt system blir da:

$$H(j\omega) = \frac{K}{Tj\omega + 1} = \frac{K}{\underbrace{1}_{Re} + j\underbrace{T\omega}_{Im}}$$

Polar form:

$$\begin{aligned} H(j\omega) &= \frac{K}{\sqrt{1^2 + (T\omega)^2} e^{j \arctan\left(\frac{T\omega}{1}\right)}} \\ &= \frac{K}{\sqrt{1 + (T\omega)^2}} e^{j[-\arctan(T\omega)]} \end{aligned}$$

Dvs.:

$$H(j\omega) = \frac{K}{\sqrt{1 + (T\omega)^2}} e^{j[-\arctan(T\omega)]}$$

Forsterkningsfunksjonen blir da:

$$A(\omega) = |H(j\omega)| = \frac{K}{\sqrt{1 + (T\omega)^2}}$$

Eller i [dB] (siden dette brukes i Bodediagrammet):

$$A(\omega)_{dB} = |H(j\omega)|_{dB} = 20 \log K - 20 \log \sqrt{1 + (T\omega)^2}$$

Fasefunksjonen blir da ([rad]):

$$\phi(\omega) = \angle H(j\omega) = \arg H(j\omega) = -\arctan(T\omega)$$

Eller i grader [°] (siden dette brukes i Bodediagrammet):

$$\phi(\omega) = \angle H(j\omega) = -\arctan(T\omega) \cdot \frac{180}{\pi}$$

Eksempel

Gitt følgende 2.ordens transferfunksjon:

$$H(s) = \frac{y(s)}{u(s)} = \frac{K}{(T_1s + 1)(T_2s + 1)}$$

Frekvensresponsen (we replace s with $j\omega$) for vårt system blir da:

$$H(j\omega) = \frac{K}{(T_1 j\omega + 1)(T_2 j\omega + 1)}$$

Polar form:

$$\begin{aligned} H(j\omega) &= \frac{K}{\left[\sqrt{1^2 + (T_1\omega)^2} e^{j \arctan\left(\frac{T_1\omega}{1}\right)} \right] \left[\sqrt{1^2 + (T_2\omega)^2} e^{j \arctan\left(\frac{T_2\omega}{1}\right)} \right]} \\ &= \frac{K}{\sqrt{1 + (T_1\omega)^2} \sqrt{1 + (T_2\omega)^2}} e^{j[-\arctan(T_1\omega) - \arctan(T_2\omega)]} \end{aligned}$$

Dvs.:

$$H(j\omega) = \frac{K}{\sqrt{1 + (T_1\omega)^2} \sqrt{1 + (T_2\omega)^2}} e^{j[-\arctan(T_1\omega) - \arctan(T_2\omega)]}$$

Forsterkningsfunksjonen blir da:

$$A(\omega) = |H(j\omega)| = \frac{K}{\sqrt{1 + (T_1\omega)^2} \sqrt{1 + (T_2\omega)^2}}$$

Eller i [dB] (siden dette brukes i Bodediagrammet):

$$A(\omega)_{dB} = |H(j\omega)|_{dB} = 20 \log K - 20 \log \sqrt{1 + (T_1\omega)^2} - 20 \log \sqrt{1 + (T_2\omega)^2}$$

Fasefunksjonen blir da ([rad]):

$$\phi(\omega) = \angle H(j\omega) = \arg H(j\omega) = -\arctan(T_1\omega) - \arctan(T_2\omega)$$

Eller i grader [°] (siden dette brukes i Bodediagrammet):

$$\phi(\omega) = \angle H(j\omega) = [-\arctan(T_1\omega) - \arctan(T_2\omega)] \cdot \frac{180}{\pi}$$

13.2.2 Eksempler

Vi finner uttrykkene for $A(\omega)$ [dB] og $\phi(\omega)$ for ulike transferfunksjoner:

Transferfunksjon:	$A(\omega)$ og $\phi(\omega)$:
$H(s) = \frac{y(s)}{u(s)} = \frac{1}{s + 1}$	$ H(j\omega) _{dB} = \underline{20 \log 1 - 20 \log \sqrt{(\omega)^2 + 1}}$

	$\angle H(j\omega) = -\arctan(\omega)$
$H(s) = \frac{4}{2s+1}$	$ H(j\omega) _{dB} = 20\log 4 - 20\log\sqrt{(2\omega)^2+1}$ $\angle H(j\omega) = -\arctan(2\omega)$
$H(s) = \frac{5}{(s+1)(10s+1)}$	$ H(j\omega) _{dB} = 20\log 5 - 20\log\sqrt{(\omega)^2+1} - 20\log\sqrt{(10\omega)^2+1}$ $\angle H(j\omega) = -\arctan(\omega) - \arctan(10\omega)$
$H(s) = \frac{1}{s(s+1)^2}$	$ H(j\omega) _{dB} = -20\log\sqrt{(\omega)^2} - 2 \times 20\log\sqrt{(\omega)^2+1}$ $= 20\log\omega - 40\log\sqrt{(\omega)^2+1}$ $\angle H(j\omega) = -90 - 2\arctan(\omega)$
$H(s) = \frac{3.2e^{-2s}}{3s+1}$	$ H(j\omega) _{dB} = 20\log 3.2 - 20\log\sqrt{(3\omega)^2+1}$ $\angle H(j\omega) = -2\omega - \arctan(3\omega)$
$H(s) = \frac{5s+1}{(2s+1)(10s+1)}$	$ H(j\omega) _{dB} = 20\log\sqrt{(5\omega)^2+1} - 20\log\sqrt{(2\omega)^2+1} - 20\log\sqrt{(10\omega)^2+1}$ $\angle H(j\omega) = \arctan(5\omega) - \arctan(2\omega) - \arctan(10\omega)$

Merk! For å finne fasen i grader må vi multiplisere med: $\frac{180}{\pi}$

Fasekurven:

→ Ledd i telleren: +90 grader for hvert ledd (ledd av typen s eller $(Ts+1)$)

→ Ledd i nevneren: -90 for hvert ledd (ledd av typen s eller $(Ts+1)$)

[Slutt på eksempel]

13.2.3 Knekkfrekvenser

Knekkfrekvensen(e) er der hvor kurven skifter retning, enten oppover eller nedover. Disse kan enkelt finnes fra transferfunksjonen og gir et godt bilde av hvordan kurven vil bli.

Eksempel:

Her er noen eksempler på knekkfrekvenser for ulike transferfunksjoner:

Transferfunksjon:	Knekkfrekvenser:
$H(s) = \frac{y(s)}{u(s)} = \frac{1}{s+1}$	$\omega = \frac{1}{T} = \frac{1}{1} = \underline{1}$
$H(s) = \frac{4}{2s+1}$	$\omega = \frac{1}{T} = \frac{1}{2} = \underline{0.5}$

$H(S) = \frac{5}{(s+1)(10s+1)}$	$\omega_1 = \frac{1}{T_1} = \frac{1}{1} = \underline{1}$ $\omega_2 = \frac{1}{T_2} = \frac{1}{10} = \underline{0.1}$
$H(S) = \frac{1}{s(s+1)^2}$	$\omega = \frac{1}{T} = \frac{1}{1} = \underline{1}$
$H(s) = \frac{3.2e^{-2s}}{3s+1}$	$\omega = \frac{1}{T} = \frac{1}{3} = \underline{0.33}$
$H(S) = \frac{5s+1}{(2s+1)(10s+1)}$	$\omega_1 = \frac{1}{T_1} = \frac{1}{5} = \underline{0.2}$ $\omega_2 = \frac{1}{T_2} = \frac{1}{2} = \underline{0.5}$ $\omega_3 = \frac{1}{T_3} = \frac{1}{10} = \underline{0.1}$

[Slutt på eksempel]

13.2.4 MathScript

Selv om MathScript kan finne frekvensresponsen og tegne et Bodediagram direkte fra transferfunksjonen vha. den innebygde funksjonen **bode** kan det også være av interesse å vite hvordan dette gjøres "manuelt", dvs. ta utgangspunkt i de matematiske uttrykkene for $A(\omega)$ og $\phi(\omega)$.

Eksempel:

Gitt følgende transferfunksjon

$$H(s) = \frac{y(s)}{u(s)} = \frac{1}{s+1}$$

Matematiske uttrykk for $A(\omega)$ og $\phi(\omega)$ er gitt ved:

$$|H(j\omega)|_{dB} = \underline{20\log 1 - 20\log\sqrt{(\omega)^2 + 1}}$$

$$\angle H(j\omega) = \underline{-\arctan(\omega)}$$

MathScript:

Vi finner $A(\omega)$ og $\phi(\omega)$ for $\omega = 1$:

```
w=1;
```

```
gain = 20*log10(1) - 20*log10(sqrt(w^2+1))
phase = -atan(w);
phasedeg = phase * 180/pi %convert to degrees
```

Svaret blir:

```
gain =
    -3.0103

phasedeg =
    -45
```

Hvis vi ønsker å finne $A(\omega)$ og $\phi(\omega)$ for flere frekvenser kan vi f.eks. bruke en For loop:

```
w = [0.01, 0.1, 1, 10, 100];
N = length(w);

for i=1:N
    gain(i) = 20*log10(1) - 20*log10(sqrt(w(i)^2+1));
    phase(i) = -atan(w(i));
    phasedeg(i) = phase(i) * 180/pi; %convert to degrees
end
```

Alternativt kan vi gjøre det slik:

```
gain = 20*log10(1) - 20*log10(sqrt(w.^2+1))
phase = -atan(w);
phasedeg = phase * 180/pi %convert to degrees
```

MathScript er et kraftig verktøy som kan jobbe direkte med vektorer ved å bruke følgende notasjon:

```
.^
.*
./
```

Svaret blir i begge tilfellene:

ω	$A(\omega)[dB]$	$\phi(\omega)(degrees)$
0.01	0	-0.6
0.1	0.04	-5.7
1	-3	-45
10	-20	-84

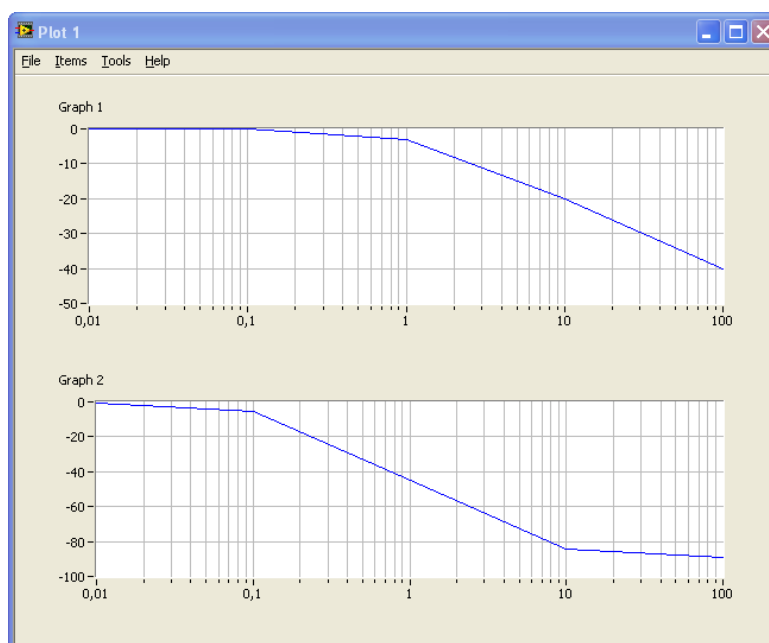
100	-40	-89
-----	-----	-----

Disse kan vi plotte:

```
...
%Gain Plot
subplot(2,1,1)
semilogx(w, gain)
grid

%Phase Plot
subplot(2,1,2)
semilogx(w, phasedeg)
grid
```

Som gir:



[Slutt på eksempel]

13.3 Praktiske forsøk

Hvordan finne frekvensresponsen fra sinuskurver på inngangen og utgangen?

Vi kan finne frekvensresponsen for et gitt system ved å logge inngangs og utgangsdata for forskjellige frekvenser, og utfra dette beregne oss til verdier for $A(\omega)$ og $\phi(\omega)$ for utvalgte frekvenser.

Inngangssignalet er gitt som:

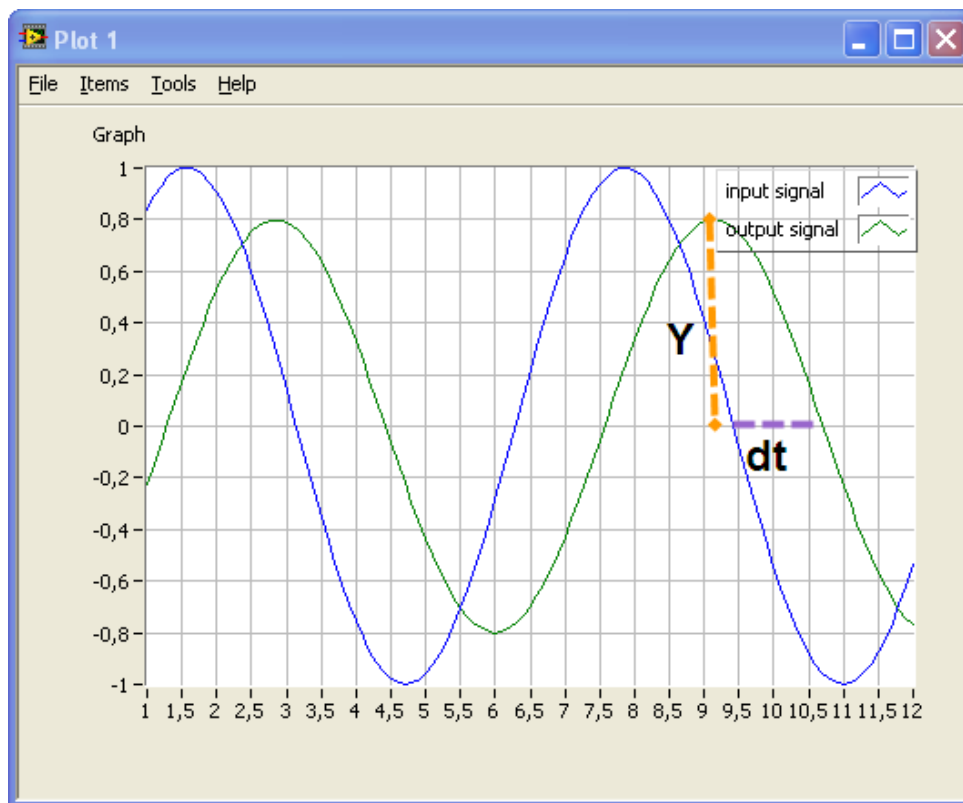
$$u(t) = U \sin \omega t$$

Steady-state utgangssignalet blir da:

$$y(t) = \underbrace{UA}_Y \sin(\omega t + \phi)$$

(dvs. Det er kun amplituden/forsterkningen som er forskjellig, samt at de er faseforskyvet i forhold til hverandre)

Vi vil da få et plot slik som dette for hver frekvens:



Forsterkningen er gitt ved:

$$A = \frac{Y}{U}$$

Der U er gitt av inngangssignalet, mens Y leses av fra plottet.

Faseforskyvningen er gitt ved:

$$\phi = -\omega \Delta t \text{ [rad]}$$

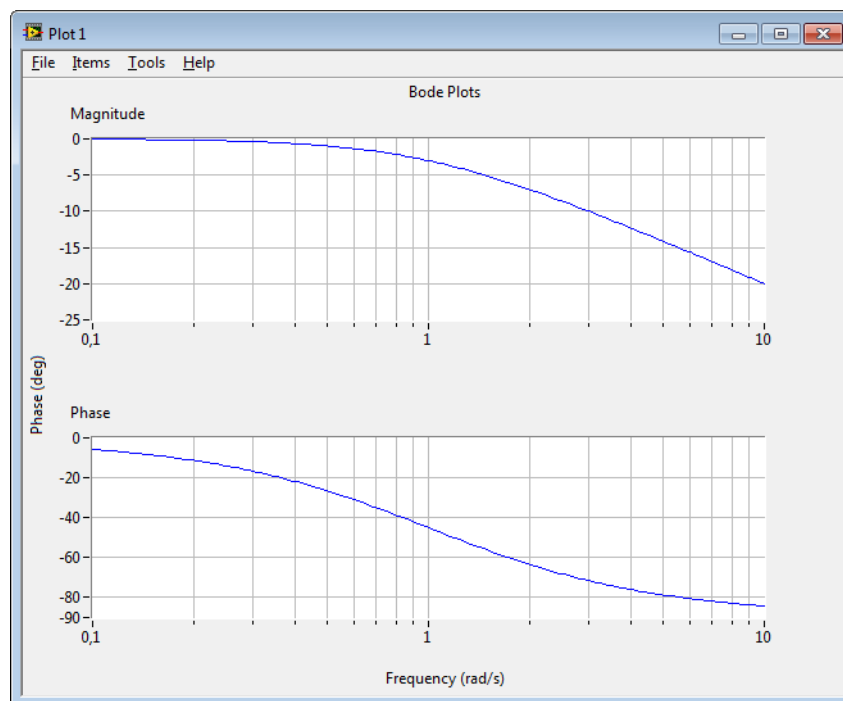
Der frekvensen ω er gitt av inngangssignalet, mens Δt leses av fra plottet.

Eksempel:

Anta følgende system:

$$H(s) = \frac{y(s)}{u(s)} = \frac{1}{s + 1}$$

Frekvensresponsen er som følger:



(Du kan bruke MathScript for å se om du får det samme)

Anta at inngangssignalet u er et sinussignal med amplitude $U = 0.8$ og frekvens $\omega = 1.0 \text{ rad/s}$.

Hva er steady-state (stasjonære) utgangssignalet.

Det Steady-state (stasjonære) utgangssignalet er definert som følger:

$$y(t) = UA \sin(\omega t + \phi)$$

hvor $U = 0.8$

Vi finner A for frekvensen $\omega = 1.0 \text{ rad/s}$ fra Bodediagrammet:

$$A(1.0) = -3 \text{ dB}$$

eller:

$$A(1.0) = 10^{-3/20} = \underline{0.71}$$

(Merk! $x[\text{dB}] = 20 \log_{10} x$)

Vi finner $\phi(1.0)$ fra Bodediagrammet:

$$\phi(1.0) = -45^\circ = -45 \frac{\pi}{180} \text{rad} = \underline{\underline{-0.79 \text{ rad}}}$$

(Merk! 2π [radianer] = 360 [grader])

Dette gir følgende steady-state utgangssignal:

$$y(t) = UA \sin(\omega t + \phi)$$

Med verdier innsatt:

$$y(t) = 0.8 \cdot 0.71 \cdot \sin(1.0 \cdot t - 0.79)$$

Dvs., steady-state utgangssignalet blir:

$$\underline{\underline{y(t) = 0.57 \sin(t - 0.79)}}$$

MathScript koden som ble brukt til å tegne frekvensresponsen er som følger:

```
% We define the transfer function:
K = 1;
T = 1;
num = [K];
den = [T, 1];
H = tf(num, den)

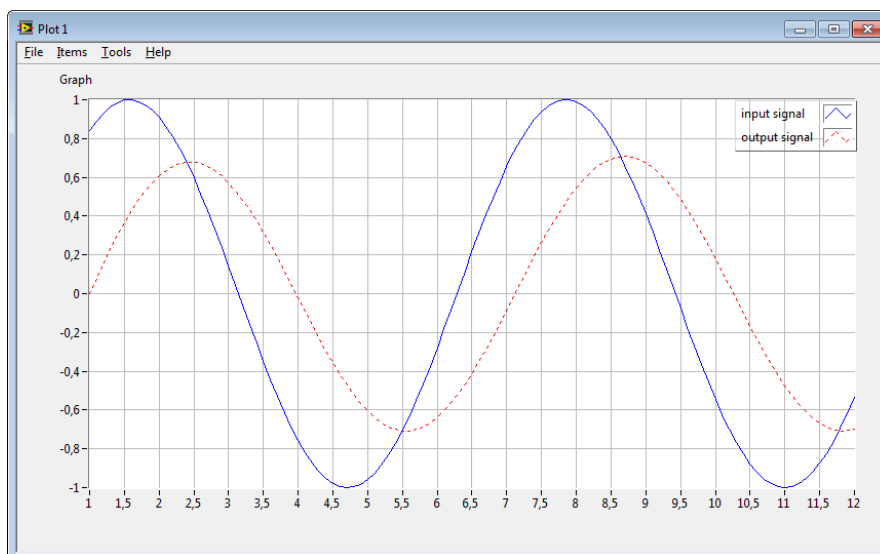
% We plot the Bode diagram:
bode(H);

% We add grid to the plot:
subplot(2,1,1)
grid on
subplot(2,1,2)
grid on
```

[Slutt på eksempel]

Eksempel:

Gitt følgende plott av et sinussignal på inngangen og et utgangssignal for et gitt system:



Inngangssignalet er definert som:

$$u(t) = U \sin \omega t$$

Steady-state utgangssignalet blir da:

$$y(t) = \underbrace{UA}_Y \sin(\omega t + \phi)$$

Forsterkningen er gitt av:

$$A = \frac{Y}{U}$$

Faseforskyvningen er gitt av:

$$\phi = -\omega \Delta t \text{ [rad]}$$

Hva blir frekvensen på signalet i Hz og i rad/s?

Fra figuren ser vi at perioden på inngangssignalet blir som følger:

$$T_p = 6.2 \text{ sec}$$

(dvs., $7.8 \text{ sec} - 1.6 \text{ sec} = 6.2 \text{ sec}$)

Dette gir følgende frekvens:

$$f = \frac{1}{T_p} = \frac{1}{6.2} = 0.16 \text{ Hz}$$

eller:

$$\omega = 2\pi f = 2\pi \cdot 0.16 = 1 \text{ rad/s}$$

Hva blir amplitudeforsterkningen i dB? Hva blir faseforskyvningen i grader?

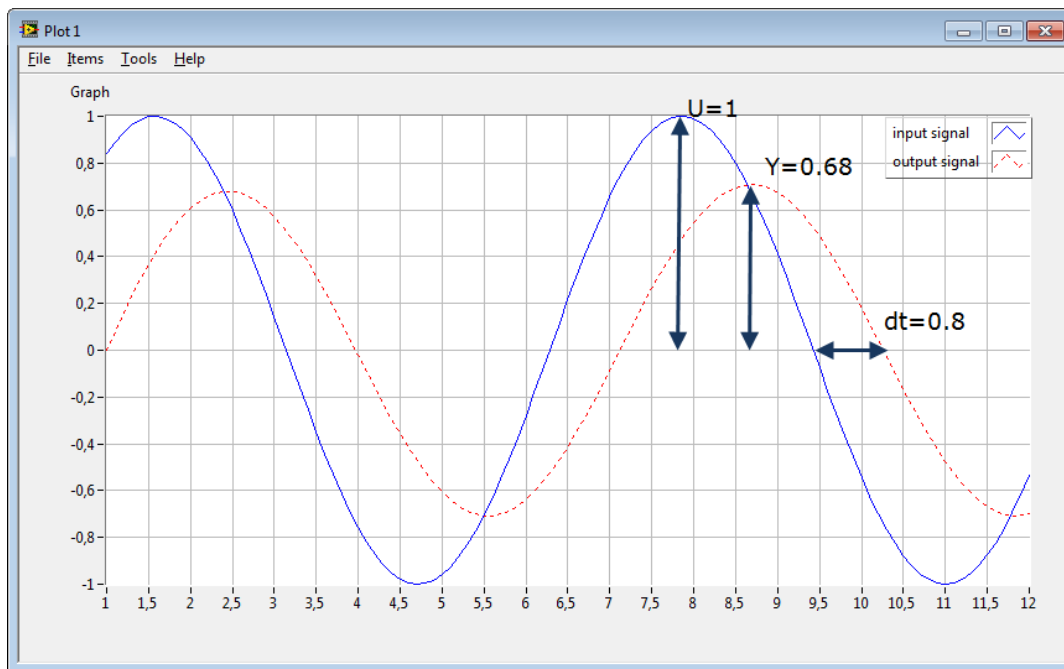
Forsterkningen er gitt av:

$$A = \frac{Y}{U}$$

Faseforskyvningen er gitt av:

$$\phi = -\omega\Delta t \text{ [rad]}$$

Fra plottet får vi følgende:



$$U = 1$$

$$Y = 0.68$$

$$\Delta t = 0.8$$

Dette gir følgende:

Amplituden:

$$A = \frac{Y}{U} = \frac{0.68}{1} = \underline{0.68}$$

eller i dB:

$$A \text{ [dB]} = 20 \log 0.68 = \underline{-3.35 \text{ dB}}$$

Faseforskyvningen:

$$\phi = -\omega\Delta t = -1 \cdot 0.8 = \underline{-0.8 \text{ rad}}$$

eller i grader ($2\pi \text{ [rad]} = 360^\circ$):

$$\phi [\text{degrees}] = \frac{180}{\pi} \cdot (-0.8) = \underline{-45.9^\circ}$$

Hva blir steady-state utgangssignalet?

Steady-state utgangssignalet:

$$y(t) = UA \sin(\omega t + \phi)$$

Med innsatte verdier:

$$y(t) = 1 \cdot 0.68 \sin(1 \cdot t - 0.8)$$

Dvs.:

$$\underline{y(t) = 0.68 \sin(t - 0.8)}$$

[Slutt på eksempel]

13.4 Frekvensrespons for standardfunksjoner

Vi vil se nærmere på frekvensresponsen for en del standardfunksjoner.

- Forsterker
- Integrator
- Derivator
- 1.ordens system
- 2.ordens system
- Nullpunkt
- Tidsforsinkelse

13.4.1 Forsterker

Transferfunksjon:

$$H(s) = K$$

Hvor K er forsterkningen

Matematiske uttrykk for $A(\omega)$ and $\phi(\omega)$ er som følger:

$A(\omega)$:

$$A(\omega) = |H(j\omega)| = K$$

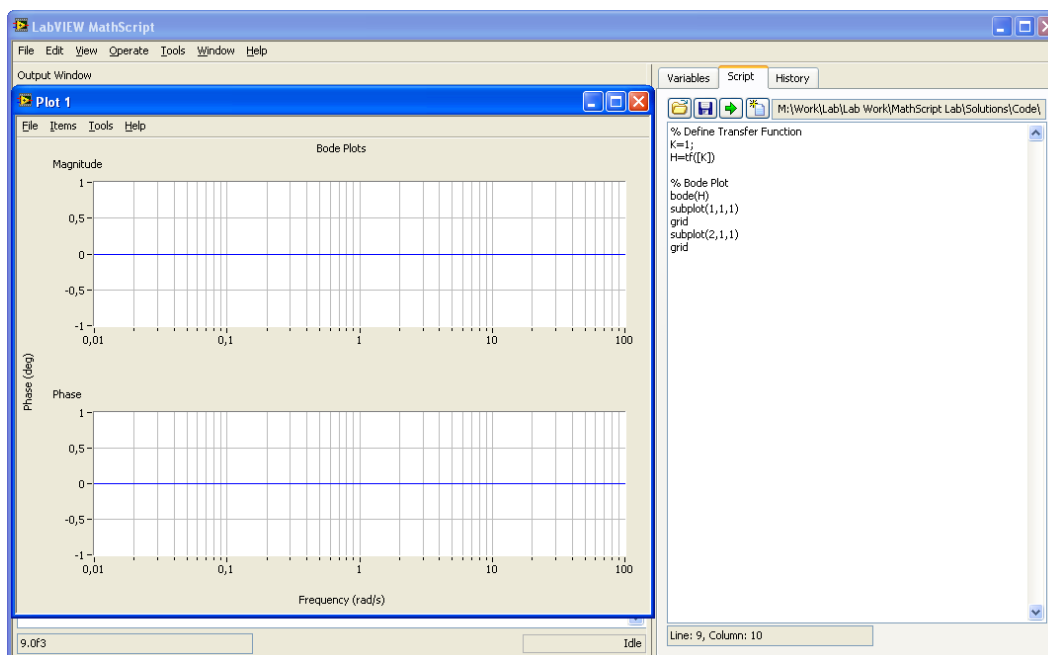
Eller i dB:

$$|H(j\omega)|_{dB} = 20\log K$$

$\phi(\omega)$:

$$\phi(\omega) = \angle H(j\omega) = 0$$

Bode plot (K=1):



→ Vi ser at både $A(\omega)$ and $\phi(\omega)$ er uavhengig av frekvensen ω .

13.4.2 Integrator

Transferfunksjon:

$$H(s) = \frac{K}{s}$$

Hvor K er forsterkningen

Matematiske uttrykk for $A(\omega)$ and $\phi(\omega)$ er som følger:

$A(\omega)$:

$$A(\omega) = |H(j\omega)| = \frac{K}{\omega}$$

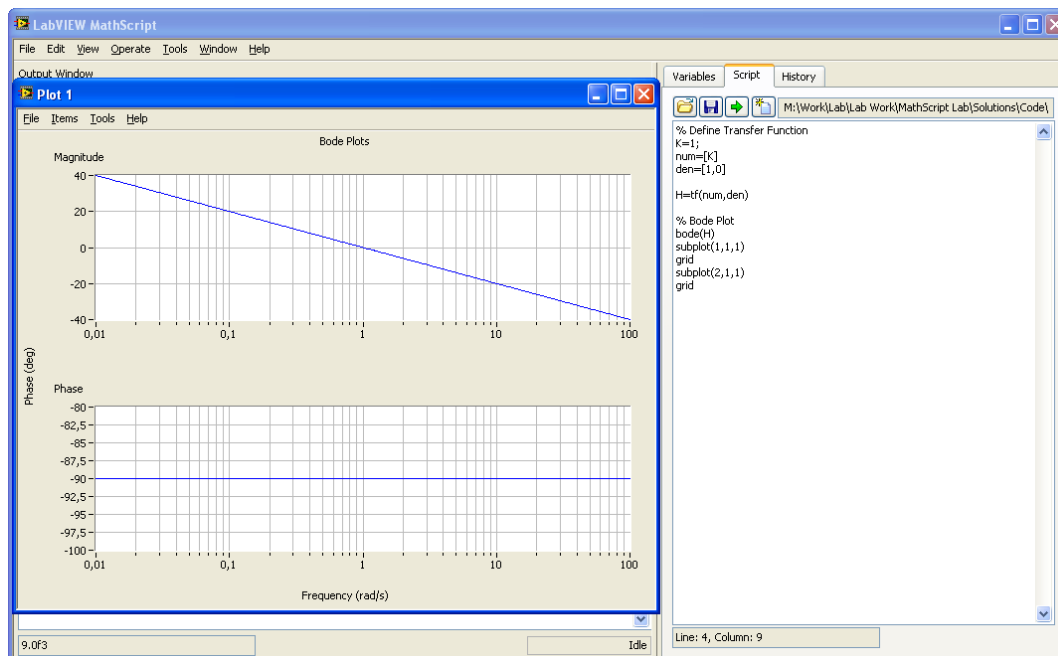
eller i dB:

$$|H(j\omega)|_{dB} = 20\log \frac{K}{\omega}$$

$\phi(\omega)$:

$$\phi(\omega) = \angle H(j\omega) = -\frac{\pi}{2} \text{ rad} = -90^\circ$$

Bode plot:



13.4.3 Derivator

Transferfunksjon:

$$H(s) = Ks$$

Hvor K er forsterkningen

Matematiske uttrykk for $A(\omega)$ and $\phi(\omega)$ er som følger:

$A(\omega)$:

$$A(\omega) = |H(j\omega)| = K\omega$$

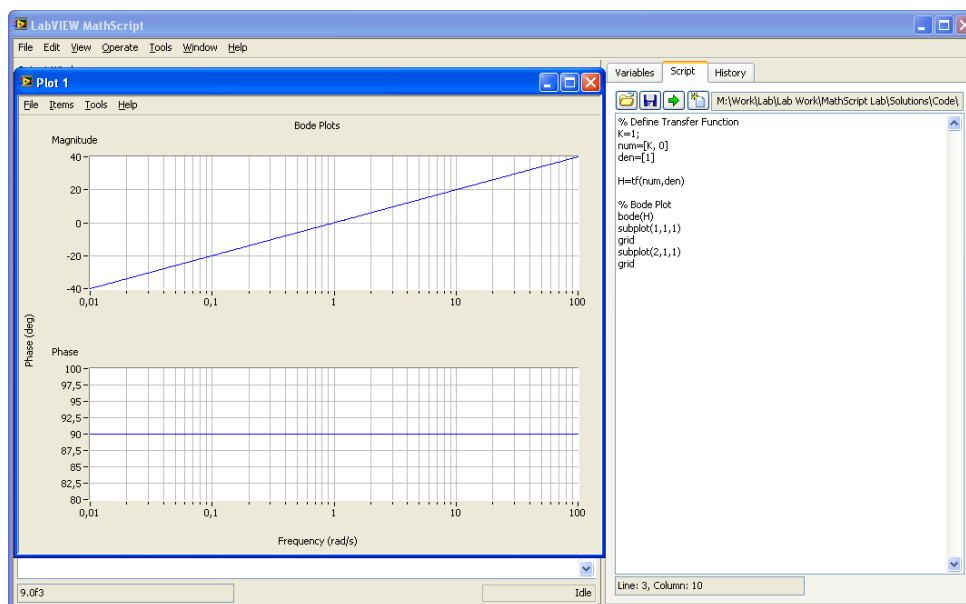
eller i dB:

$$|H(j\omega)|_{dB} = 20 \log K\omega$$

$\phi(\omega)$:

$$\phi(\omega) = \angle H(j\omega) = +\frac{\pi}{2} \text{ rad} = +90^\circ$$

Bode plot:



13.4.4 1.ordens system

Transferfunksjon:

$$H(s) = \frac{K}{Ts + 1}$$

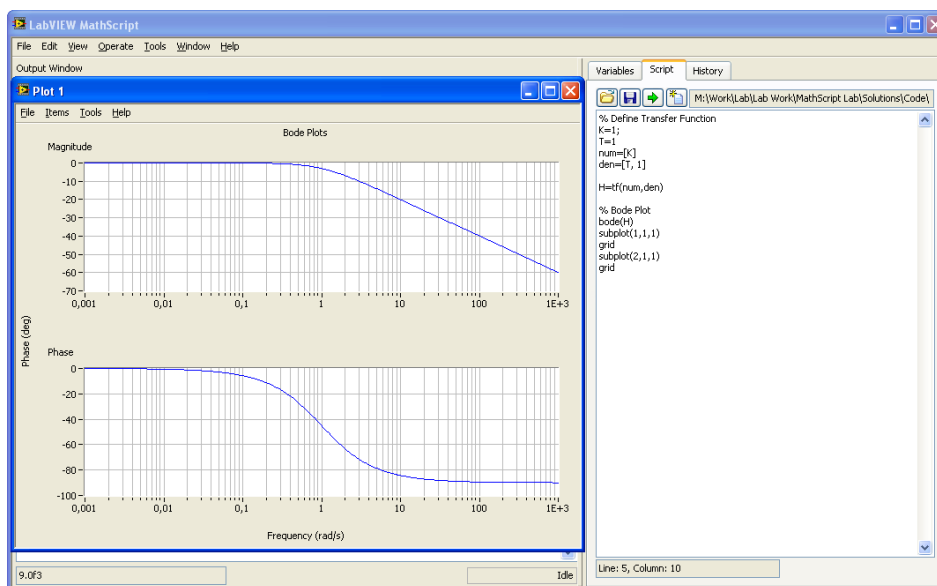
Hvor K er forsterkningen og T er tidskonstanten.

Matematiske uttrykk for $A(\omega)$ and $\phi(\omega)$ er som følger:

$$A(\omega) = |H(j\omega)| = \frac{K}{\omega^2 T^2 + 1}$$

$$\phi(\omega) = \angle H(j\omega) = -\arctan(\omega T)$$

Bode plot:



13.4.5 2.ordens system

Transferfunksjon:

$$H(s) = \frac{K\omega_0^2}{s^2 + 2\zeta\omega_0s + \omega_0^2} = \frac{K}{\left(\frac{s}{\omega_0}\right)^2 + 2\zeta\frac{s}{\omega_0} + 1}$$

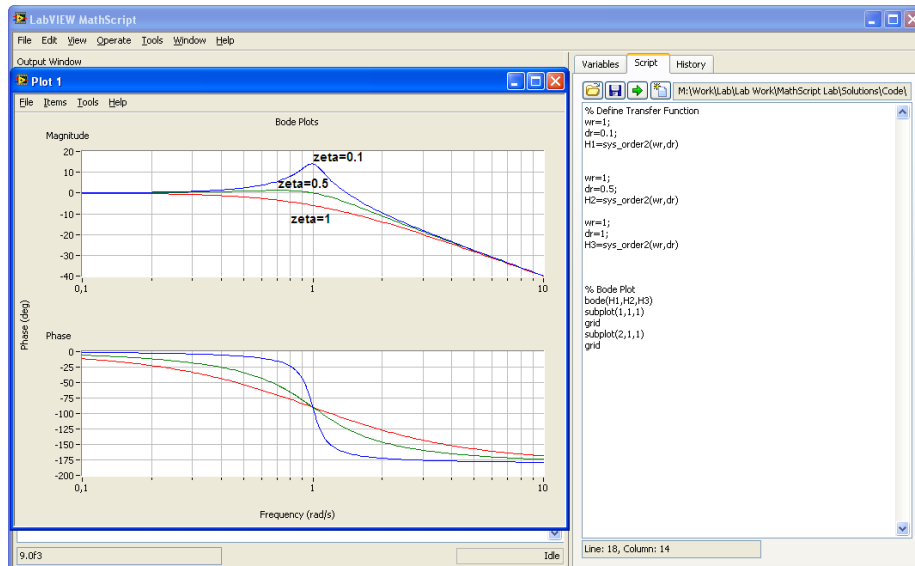
Hvor

K er forsterkningen

ζ (zeta) er relativ dempingsfaktor

ω_0 [rad/s] er udempet resonansfrekvens

Bode plot:



13.4.6 Nullpunktsledd

Transferfunksjon:

$$H(s) = K(Ts + 1)$$

Hvor K er forsterkningen og T er tidskonstanten.

Matematiske uttrykk for $A(\omega)$ and $\phi(\omega)$ er som følger:

$A(\omega)$:

$$A(\omega) = |H(j\omega)| = K\sqrt{(\omega T)^2 + 1}$$

eller i dB:

$$|H(j\omega)|_{dB} = 20\log K\sqrt{(\omega T)^2 + 1}$$

$\phi(\omega)$:

$$\phi(\omega) = \angle H(j\omega) = +\arctan(\omega T)$$

13.4.7 Tidsforsinkelse

Transferfunksjon:

$$H(s) = Ke^{-\tau s}$$

Hvor K er forsterkningen og τ er tidsforsinkelsen

Matematiske uttrykk for $A(\omega)$ and $\phi(\omega)$ er som følger:

$A(\omega)$:

$$A(\omega) = |H(j\omega)| = K$$

$\phi(\omega)$:

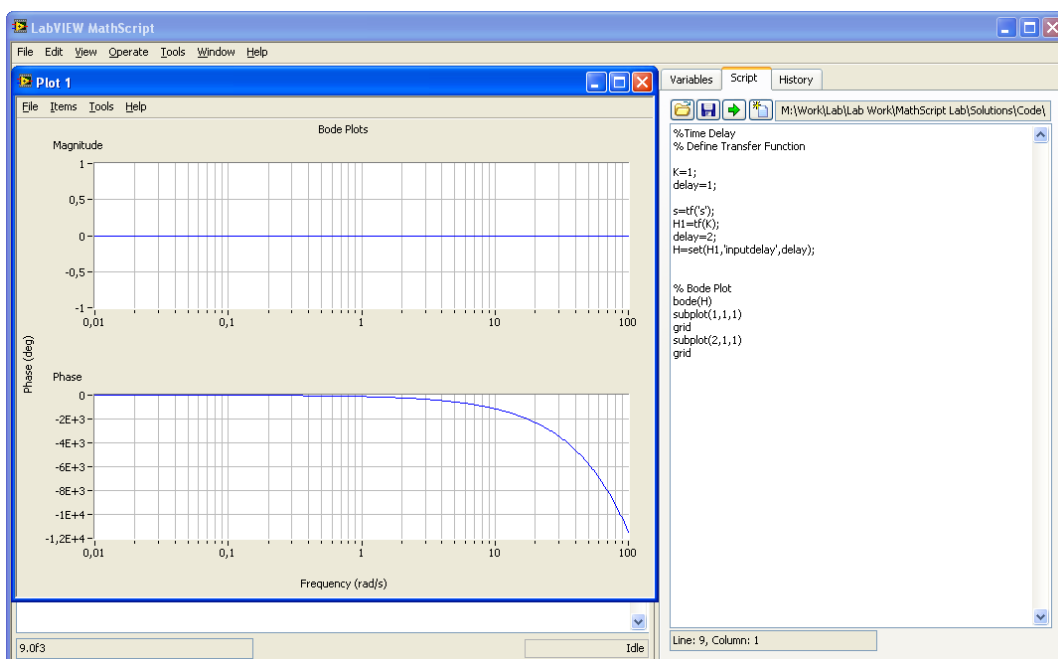
$$\phi(\omega) = \angle H(j\omega) = -\omega\tau \text{ rad} = -\omega\tau \frac{180}{\pi} \text{ degrees}$$

Merk!

$$2\pi \text{ rad} = 360^\circ$$

$$\pi \text{ rad} = 180^\circ$$

Bode plot:



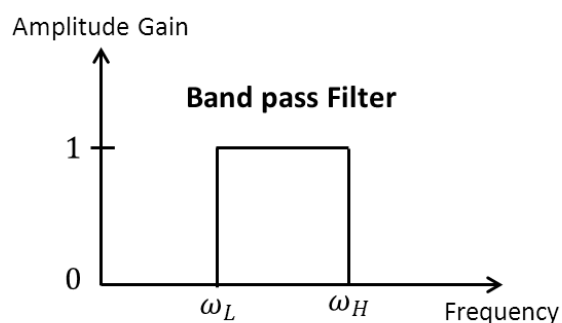
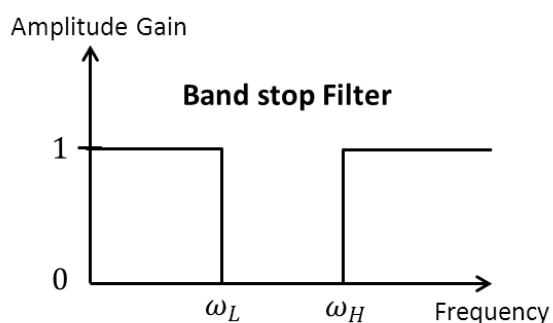
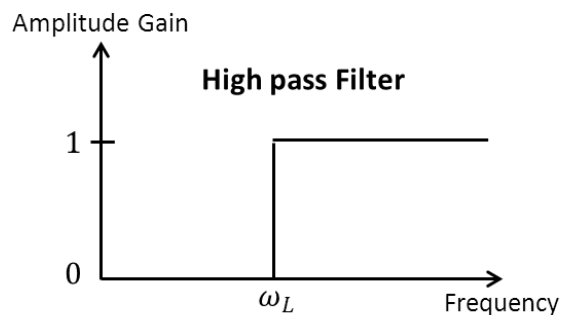
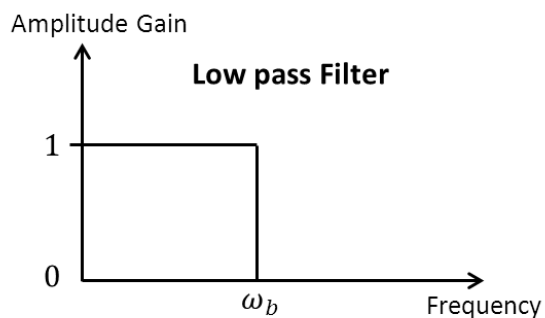
13.5 Filtre

En viktig anvendelse av frekvensrespons er ifm. filtre. Et filter brukes til å fjerne gitte frekvenser eller et intervall med frekvenser fra et signal. En slik anvendelse vil typisk være å fjerne støy fra et signal. Det mest vanlige er Lavpassfilteret.

Vi har 4 typer filtre:

- Lavpassfilter
- Høypassfilter
- Båndstoppfilter
- Båndpassfilter

Nedenfor ser vi de ideelle filterkarakteristikkene for disse 4 typene:



Filtrene vil slippe gjennom signaler for de frekvensene hvor amplitudeforsterkingen er 1 på figuren (eller $0dB$). For virkelige filtere vil ikke overgangen være så skarpe som på bildet over (ideelle).

13.5.1 Lavpassfilter

Transferfunksjon for et Lavpassfilter:

$$H(s) = \frac{1}{Ts + 1} = \frac{1}{\frac{1}{\omega_b}s + 1}$$

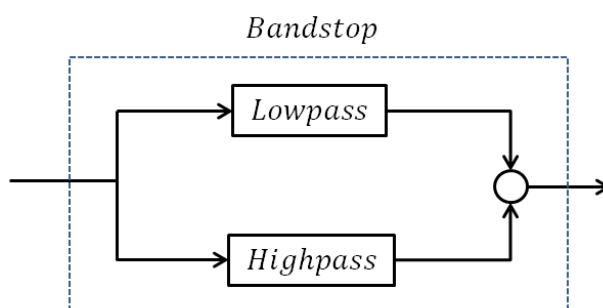
Der ω_b er definert som filterets båndbredde.

13.5.2 Høypassfilter

Transferfunksjon for et Høypassfilter:

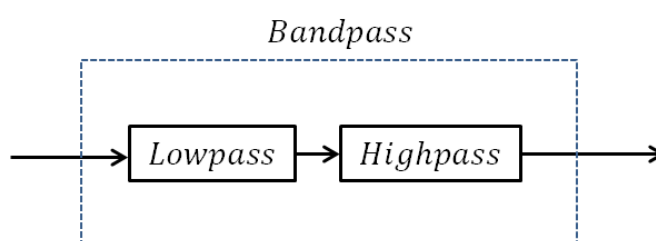
$$H(s) = \frac{Ts}{Ts + 1} = \frac{\frac{1}{\omega_b}s}{\frac{1}{\omega_b}s + 1}$$

13.5.3 Båndstopp



For dette filteret definerer vi en nedre frekvens ω_L og en øvre frekvens ω_H .

13.5.4 Båndpass



For dette filteret definerer vi en nedre frekvens ω_L og en øvre frekvens ω_H .

13.5.5 MathScript

Eksempel:

MathScript kode:

Lavpassfilter:

```
wl = 1;
Tl = 1/wl;

num = [1];
den = [Tl, 1];

H_lowpass = tf(num, den)

figure(1)
bodemag(H_lowpass)
grid
title('Lowpass Filter')
```

Høypassfilter:

```
wh = 10;
Th = 1/wh;

num = [Th, 0];
den = [Th, 1];

H_highpass = tf(num, den)

figure(2)
bodemag(H_highpass)
grid
title('Highpass Filter')
```

Båndstopfilter:

Vi parallellkobler et Lavpassfilter og et Høypassfilter:

```
H_bandstop = parallel(H_lowpass, H_highpass)

figure(3)
bodemag(H_bandstop)
grid
title('Bandstop Filter')
```

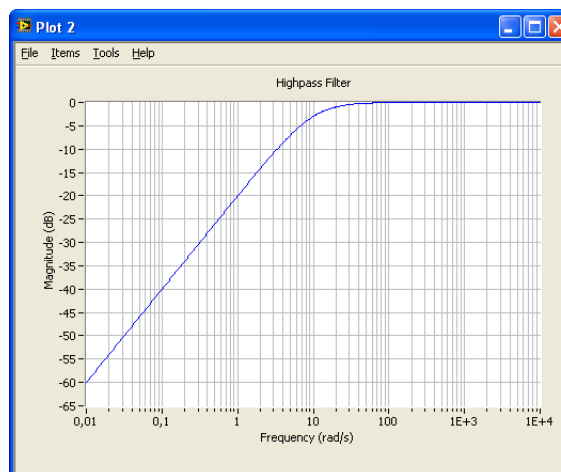
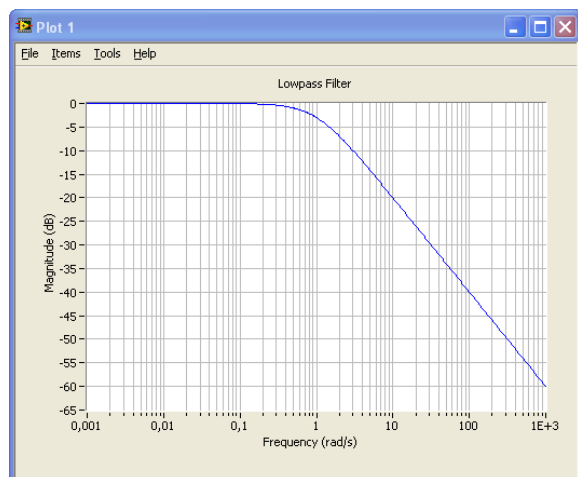
Båndpassfilter:

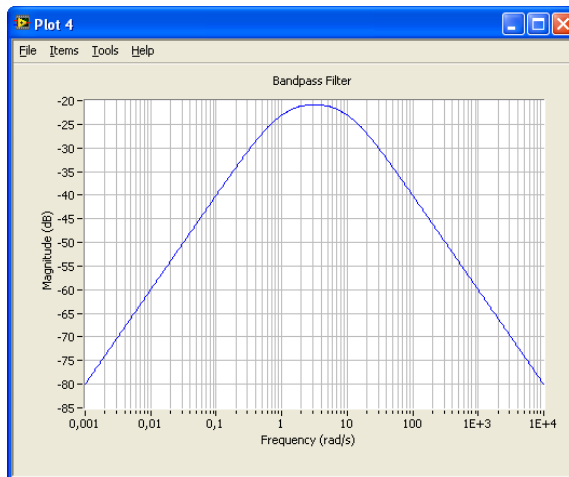
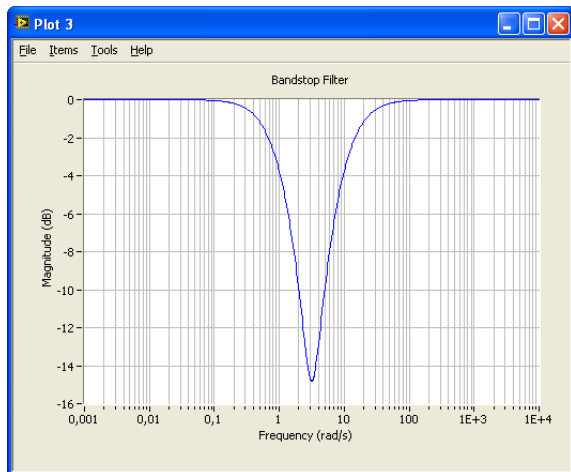
Vi seriekobler et Lavpassfilter og et Høypassfilter:

```
H_bandpass = series(H_lowpass, H_highpass)

figure(4)
bodemag(H_bandpass)
grid
title('Bandpass Filter')
```

Plot:



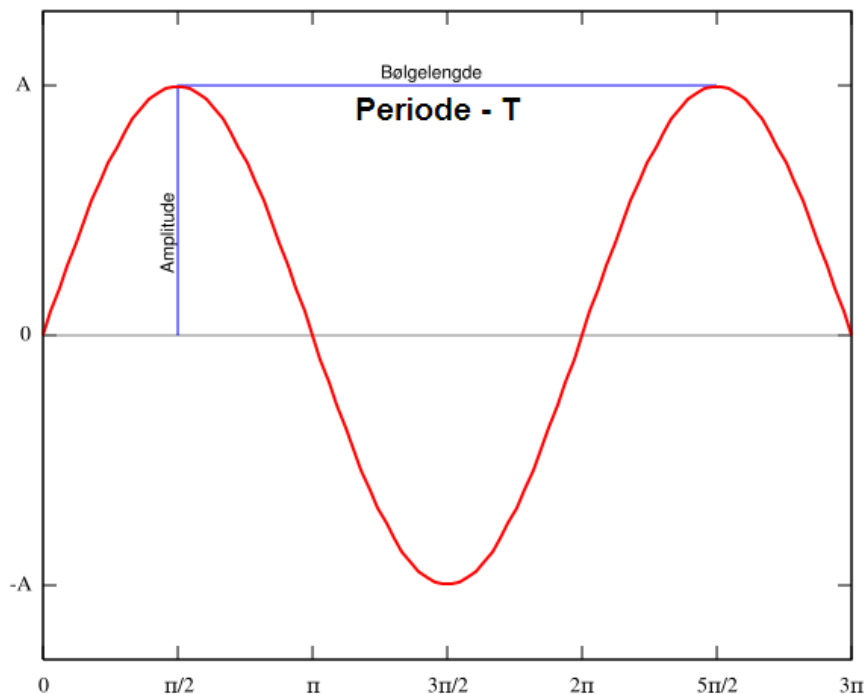


Vi ser at resultatene blir som forventet.

[Slutt på Eksempel]

13.6 Definisjoner

Gitt følgende sinuskurve:



13.6.1 Periode – T

Avstanden mellom 2 topper.

13.6.2 Amplitude - A

Amplitude er avstanden fra en bølges største utslag (toppunkt eller bunnpunkt) til likevektstilstanden (midten). Amplituden er altså en positiv verdi. For sinus- og cosinusbølger er amplituden halvparten av høyden mellom toppunkt og bunnpunkt.

13.6.3 Frekvens - f

Frekvens er et mål på antallet ganger en hendelse gjentar seg i løpet av en enhetstid. For å beregne frekvens, setter man et fast tidsintervall, teller antall ganger en hendelse inntreffer og dividerer på tidsintervallets lengde.

Så lenge hendelsene inntreffer regelmessig er en enkel metode for å beregne frekvens å måle tiden mellom to ganger hendelsen inntreffer (perioden) og så beregne frekvensen f som den inverse av denne tiden:

$$f = \frac{1}{T}$$

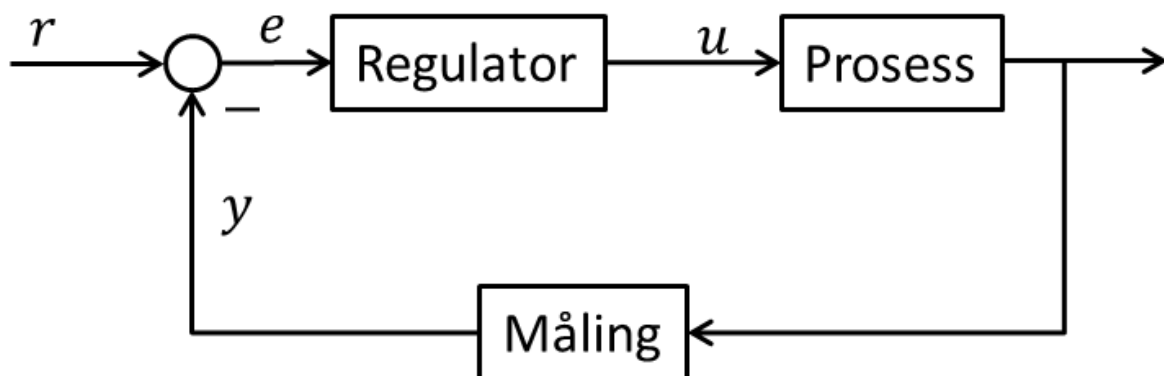
Vanligvis er enheten for frekvens Hertz [*Hz*], mens i frekvensrespons brukes radianer ω [*rad*]. Sammenhengen mellom disse er:

$$\omega = 2\pi f$$

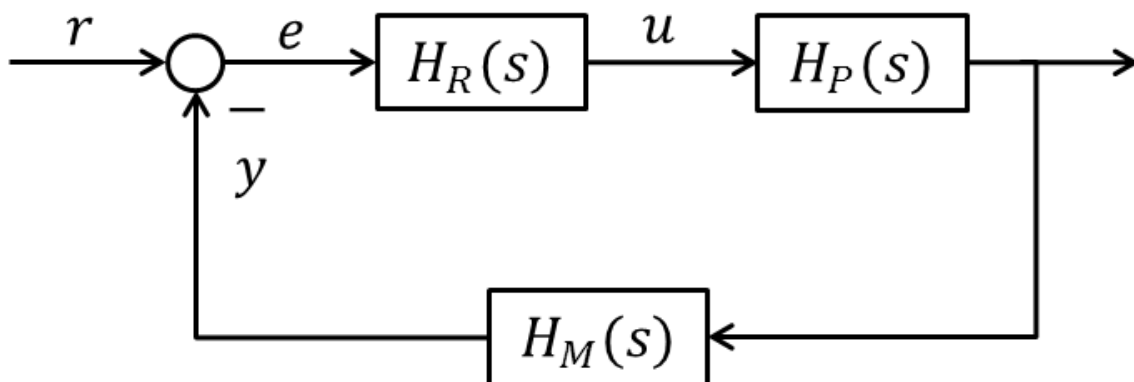
14 Frekvensrespons- analyse

14.1 Innledning

Skisse av et tilbakekoblet system:



Der Regulatoren, Prosessen og Måleinstrumentet er transferfunksjoner:



Vi har da følgende:

Sløyfetransferfunksjonen (Engelsk: "Loop transfer function"):

$$L(s) = H_R H_P H_M$$

Følgeforholdet (Engelsk: "Tracking transfer function"):

$$T(s) = \frac{y(s)}{r(s)} = \frac{H_R H_P H_M}{1 + H_R H_P H_M} = \frac{L(s)}{1 + L(s)}$$

Denne uttrykker hvor godt systemet følger referansen. Systemet har gode følgeegenskaper hvis $y \approx r$, dvs.:

$$|T| \approx 1$$

Sensitivitetsfunksjonen/Avviksforholdet (Engelsk: "Sensitivity transfer function"):

$$S(s) = \frac{e(s)}{r(s)} = \frac{1}{1 + L(s)} = 1 - T(s)$$

Denne uttrykker hvor "sensitivt" avviket er overfor referansen og denne bør derfor være "liten", dvs.:

$$|S| \approx 0 \text{ eller } |S| \ll 1$$

Merk!

$$S(s) = 1 - T(s) \leftrightarrow T(s) = 1 - S(s)$$

og

$$T(s) + S(s) = \frac{L(s)}{1 + L(s)} + \frac{1}{1 + L(s)} \equiv 1$$

14.2 Følgeegenskaper

I frekvensplanet har vi:

Gode følgeegenskaper:

$$|T(j\omega)| \approx 1 \text{ eller } |S(j\omega)| \ll 1$$

Eller:

$$|L(j\omega)| \gg 1$$

Dårlige følgeegenskaper:

$$|T(j\omega)| \ll 1 \text{ eller } |S(j\omega)| \approx 1$$

Eller:

$$|L(j\omega)| \ll 1$$

Kryssfrequensen:

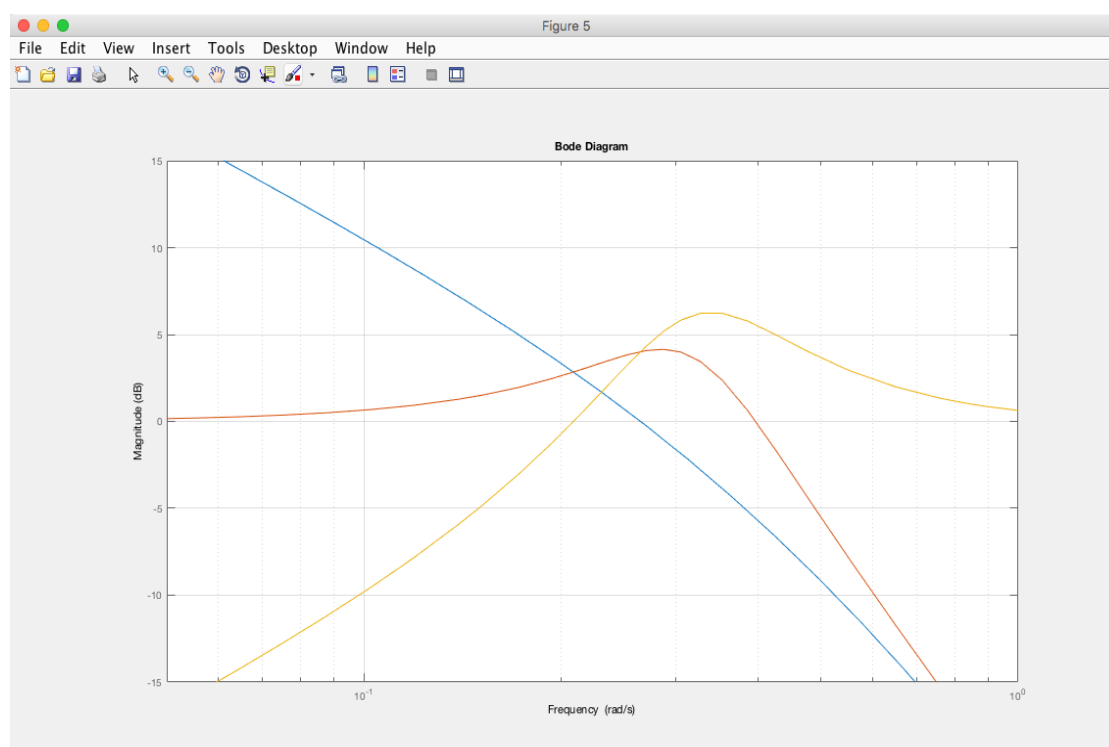
På bakgrunn av dette kan vi definere kryssfrequensen ω_c :

$$|L(j\omega_c)| = 1 = 0dB$$

→ Et reguleringsystems følgeegenskaper er *gode* for frekvenser *under* kryssfrequensen, mens de er *dårlige* for frekvenser *over* kryssfrequensen.

14.3 Båndbredde

Når vi plotter L , T og S vil vi typisk få et Bodeplot som dette (Kun amplitudeplottet er interessant i denne sammenhengen):



Ut fra dette definerer vi følgende båndbredder:

ω_c – Kryssfrequensen – Frekvensen der sløyfetransferfunksjonen $L(j\omega)$ har følgende verdi:

$$1 = \underline{0dB}$$

ω_t – Frekvensen der forsterkningen til Følgefunksjonen $T(j\omega)$ har følgende verdi:

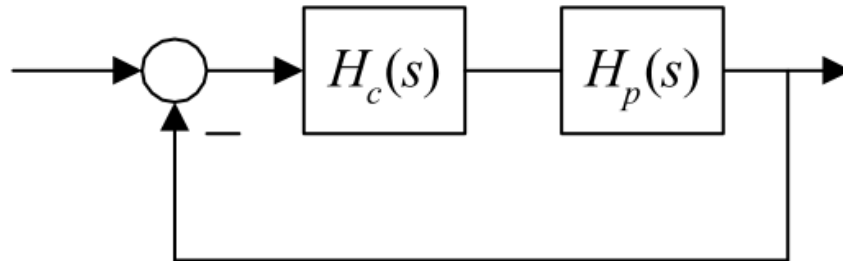
$$\frac{1}{\sqrt{2}} \approx 0.71 = \underline{-3dB}$$

ω_s – Frekvensen hvor forsterkningen til sensitivitetsfunksjonen $S(j\omega)$ har følgende verdi:

$$1 - \frac{1}{\sqrt{2}} \approx 0.29 = \underline{\underline{-11dB}}$$

14.4 MathScript

Gitt følgende reguleringsystem:



Sløyfetransferfunksjonen:

$$L = \text{series}(H_c, H_p)$$

Hvis flere enn 2: $M = \text{series}(H_1, \text{series}(H_2, H_3))$

Følgforholdet:

$$T = \text{feedback}(L, 1)$$

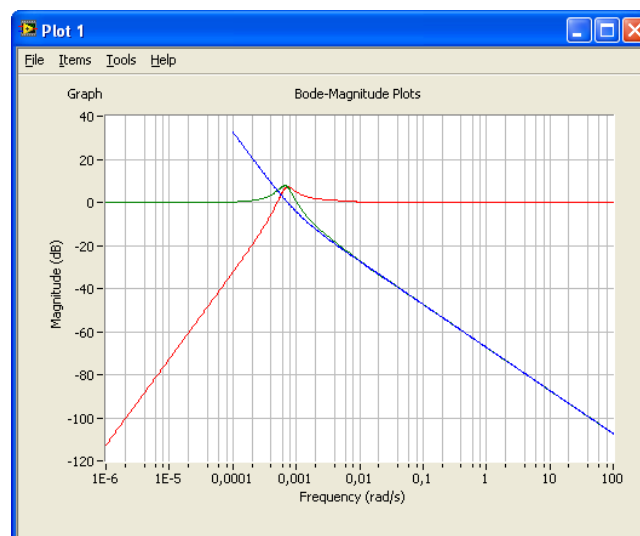
Sensitivitetsfunksjonen/Avviksforholdet:

$$S = 1 - T$$

Tilslutt tegner vi disse tre funksjonene i et Bodediagram (kun amplitudeplottet):

$$\text{bodemag}(L, T, S)$$

I MathScript vil plottet typisk se slik ut:



Ut fra dette plottet kan vi lese av båndbreddene ω_c , ω_t og ω_s .

I MathScript kan vi enkelt forandre på aksene slik at det blir enklere å lese av verdiene for kurven.

15 Stabilitetsanalyse i Frekvensplanet

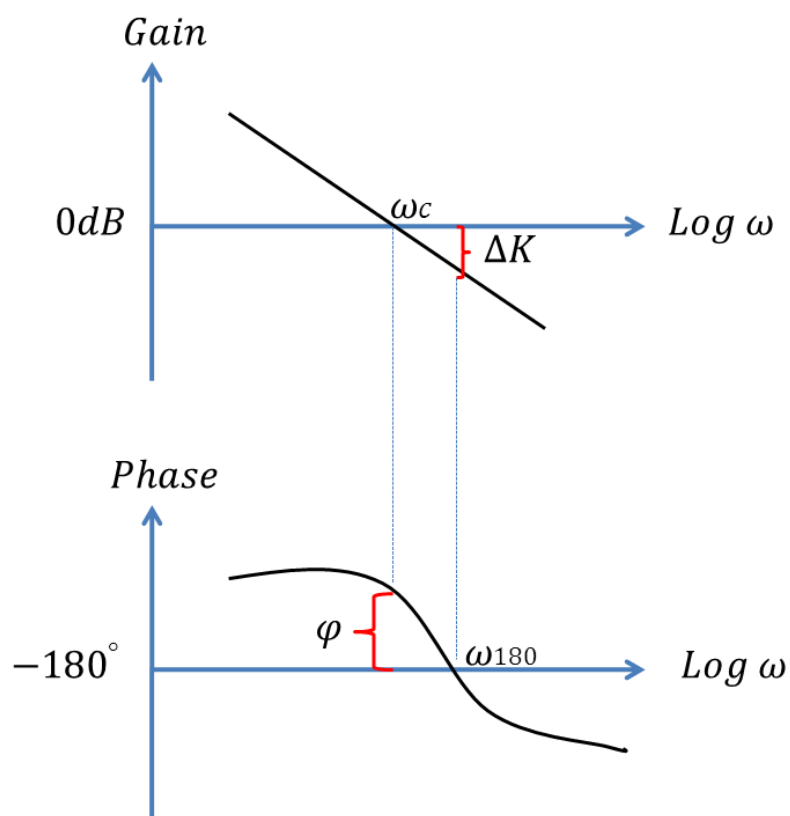
Et system kan ha en av følgende stabilitetsegenskaper:

- Asymptotisk stabilt system
- Marginalt stabilt system
- Ustabilt system

15.1 Stabilitetsmarginer

Forsterkningsmargin ($GM/\Delta K$) og Fasemarginen (PM/φ) er viktige mtp. analyse av stabilitetsegenskaper for tilbakekoblede systemer.

Vi tegner følgende Bodediagram:



Ut fra dette definerer vi følgende:

Kryssfrequensen:Kryssfrequensen ω_c :

$$|L(j\omega_c)| = 1 = 0dB$$

Fasekryssfrequensen ω_{180} :

$$\angle L(j\omega_{180}) = -180^\circ$$

Forsterkningsmargin:Forsterkningsmargin $GM/\Delta K$:

$$\Delta K = \frac{1}{|L(j\omega_{180})|}$$

Eller:

$$GM [dB] = -|L(j\omega_{180})| [dB]$$

Forsterkningsmargin ($GM/\Delta K$) sier hvor mye sløyfeforsterkningen kan øke før systemet blir ustabil.

Fasemargin:Fasemarginen PM/φ :

$$PM = 180^\circ + \angle L(j\omega_c)$$

Fasemarginen (PM/φ) sier hvor mye faseforskyvningen kan reduseres før systemet blir ustabil.

15.2 Stabilitetsanalyse

Ut fra dette definerer vi følgende stabilitetsegenskaper:

Asymptotisk stabilt system:	$\omega_c < \omega_{180}$
Marginalt stabilt system:	$\omega_c = \omega_{180}$
Ustabil system:	$\omega_c > \omega_{180}$

15.2.1 MathScript

I MathScript kan vi lese av ω_c , ω_{180} , ΔK , φ fra Bodediagrammet som vi finner ved å bruke **bode** funksjonen i MathScript.

En annen metode er å bruke funksjonen **margin**.

Eksempel:

Sløyfetransferfunksjonen er gitt som følger:

$$L(S) = \frac{K_p}{s(2s + 1)(5s + 1)}$$

Vi setter $K_p = 0.1$

Vi starter med å definere transferfunksjonen:

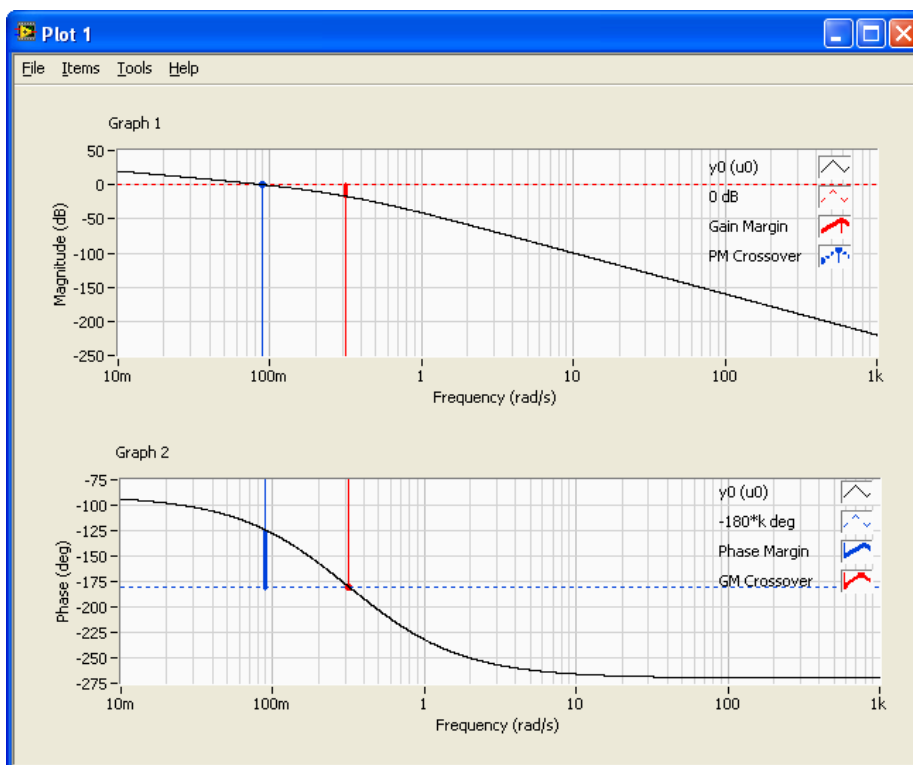
```
Kp = 0.1;  
Num = [Kp];  
den1 = [1, 0];  
den2 = [2, 1];  
den3 = [5, 1];  
den = conv(den1, conv(den2, den3));  
L = tf(num, den)
```

Deretter finner vi ω_c , ω_{180} , ΔK , φ :

```
margin(L)
```

der $L(s)$ er sløyfetransferfunksjonen.

Når vi bruker margin funksjonen på denne måten, får vi et Bodediagram med ω_c , ω_{180} , ΔK , φ ferdig tegnet inn:



Vi kan også få tallverdier for ω_c , ω_{180} , ΔK (GM), φ (PM) direkte ved å bruke **margin** funksjonen slik:

```
[gm, pm, w180, wc] = margin(L)
```

Merk! Det er feil i hjelpeteksten som dukker opp når du skriver “help margin” i Command window i MathScript. Det riktige er rekkefølgen som er illustrert i eksemplet over!

Hvis vi ønsker GM i dB gjør vi følgende:

```
gmdB = 20*log10(gm)
```

Dette gir følgende resultat:

$$\omega_c = 0.09 \text{ rad}$$

$$\omega_{180} = 0.3 \text{ rad}$$

$$\Delta K \text{ (GM)} = 6.99 \text{ eller } 16.9 \text{ dB}$$

$$\varphi \text{ (PM)} = 55 \text{ grader}$$

Vi ser at $\omega_c < \omega_{180} \rightarrow$ Asymptotisk stabilt system

Vi kan øke sløyfeforsterkningen med 6.99 før systemet blir ustabil, dvs. for $K_p = 0.7$ og større vil systemet være ustabil.

Hvor store bør stabilitetsmarginene være?

2 egenskaper som for et system som stabilitetsmarginene kan gi uttrykk for:

- Innsvingningsforløpet i reguleringsystemets dynamiske responser
- Reguleringsystemets robusthet overfor visse parameterendringer i sløyfen. Med robusthet menes reguleringsystemets evne til å opprettholde visse egenskaper til tross for parameterendringer.

Gylden regel:

Stabilitetsmarginene bør ligge i området:

$$2 (6dB) < \Delta K < 4 (12dB)$$

$$30grader < \varphi < 60 grader$$

Så i eksemplet vårt kan K_p økes noe, f.eks. $K_p = 0.2$

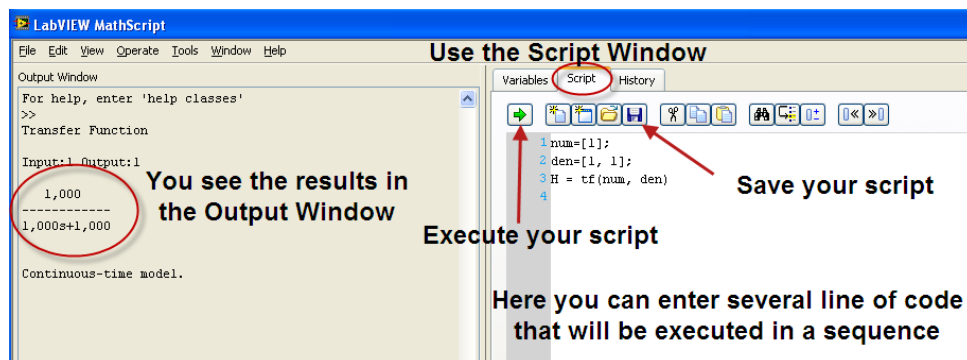
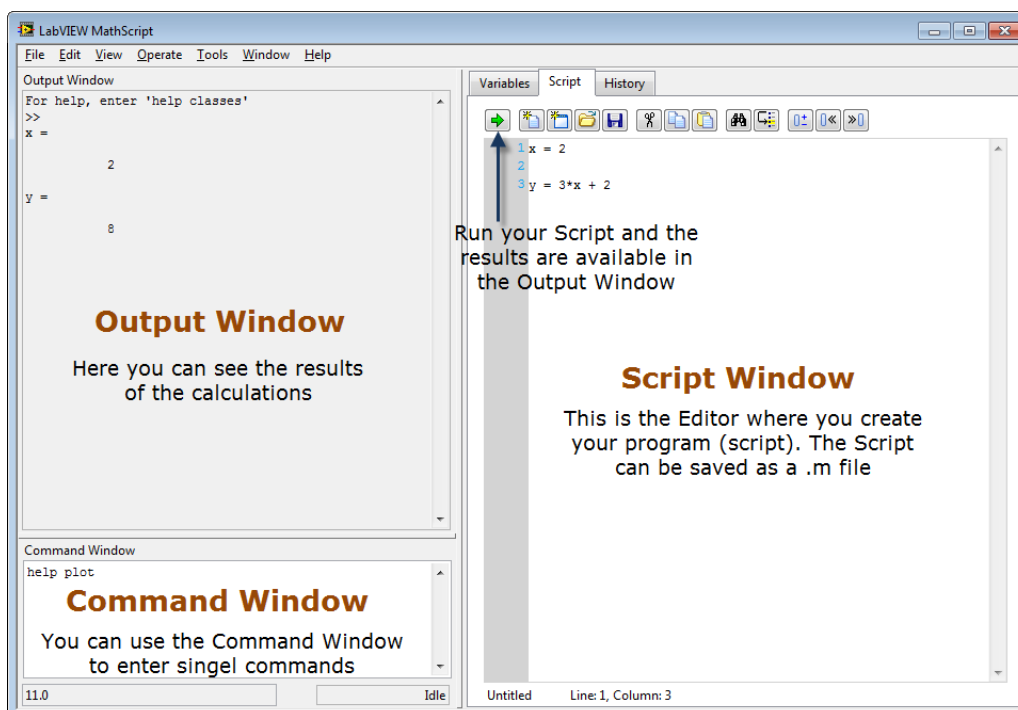
[Slutt på eksempel]

16 MathScript Tips & Triks

16.1 Innledning

MathScript er en tilleggspakke til LabVIEW. Syntaksen er identisk som MATLAB. MathScript er velegnet ifm. design, analyse og simulering av reguleringsystemer da det finnes mange innebygde funksjoner for dette.

Vi åpner MathScript fra menyen i LabVIEW: Tools → MathScript Window



Exempel:

Gitt følgende funksjon:

$$y(x) = 3x + 2$$

Vi ønsker å finne $y(2)$

Først må vi definere x :

```
x = 2
```

Deretter kan vi definere funksjonen:

```
y = 3*x + 2
```

MathScript gir da følgende svar:

```
y =
```

```
8
```

Merk! Maple kan utføre symbolsk matematikk, mens MathScript (og VB/C#) er et numerisk verktøy. Dette betyr at du alltid må definere verdier for variablene dine før du bruker dem i et matematisk uttrykk.

Merk! Syntaksen i MathScript er lik syntaksen i VB/C#, men MathScript er mye enklere på den måten at du ikke trenger å deklarere variablene og bestemme en datatype før du bruker dem. Dette blir automatisk håndtert av kompilatoren.

16.2 Plotting

MATLAB er veldig anvendelig til å plote verdier, funksjoner, m.m.

Eksempel:

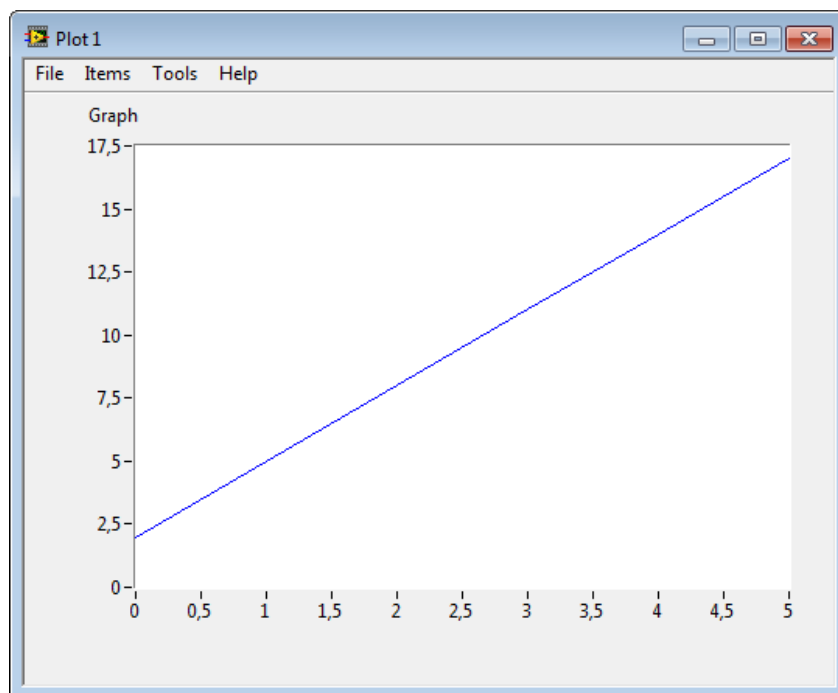
Vi ønsker å plote følgende uttrykk vha. MathScript:

$$y(x) = 3x + 2$$

Vi bruker da **plot()** funksjonen i MathScript.

```
x = 0:5;  
y = 3*x + 2;  
plot(x,y)
```

I eksemplet over har vi plottet funksjonen i intervallet $[0, 5]$. Resultatet blir da som følger:



Følgene funksjoner blir brukt mye ifm plotting:

Function	Description	Example
plot	Generates a plot. <code>plot(y)</code> plots the columns of <code>y</code> against the indexes of the columns.	<pre>>X = [0:0.01:1]; >Y = X.*X; >plot(X, Y)</pre>
figure	Create a new figure window	<pre>>>figure >>figure(1)</pre>
subplot	Create subplots in a Figure. <code>subplot(m,n,p)</code> or <code>subplot(mnp)</code> , breaks the Figure window into an m-by-n matrix of small axes, selects the p-th axes for the current plot. The axes are counted along the top row of the Figure window, then the second row, etc.	<pre>>>subplot(2,2,1)</pre>
grid	Creates grid lines in a plot. "grid on" adds major grid lines to the current plot. "grid off" removes major and minor grid lines from the current plot.	<pre>>>grid >>grid on >>grid off</pre>
axis	Control axis scaling and appearance. "axis([xmin xmax ymin ymax])" sets the limits for the x- and y-axis of the current axes.	<pre>>>axis([xmin xmax ymin ymax]) >>axis off >>axis on</pre>
title	Add title to current plot <code>title('string')</code>	<pre>>>title('this is a title')</pre>
xlabel	Add xlabel to current plot <code>xlabel('string')</code>	<pre>>> xlabel('time')</pre>
ylabel	Add ylabel to current plot <code>ylabel('string')</code>	<pre>>> ylabel('temperature')</pre>
legend	Creates a legend in the corner (or at a specified position) of the plot	<pre>>> legend('temperature')</pre>
hold	Freezes the current plot, so that additional plots can be overlaid	<pre>>>hold on >>hold off</pre>

16.3 Tips & Triks

→ Bruk **Script** window – ikke Command window. Du kan bruke Command window til enkle ting, for eksempel til `help <function>`. I Script window kan du lagre koden i såkalte .m filer.

I Command window kan du bare skrive en og en kommando av gangen, mens i Script window kan du skrive flere kommandoer adskilt med linjeskift.

→ **Kommentarer** - %

```
% Dette er en kommentar  
x=2;  
y=3*x
```

Bruk kommentarer aktivt for å øke lesbarheten i programmet ditt!

→ **Desimaltegn**: Bruk punktum – ikke komma! Dvs. $y=3.2$ – ikke $y=3,2$

→ Ikke bruk mellomrom (space) i filnavn eller navn på funksjoner!

→ Bruk piltaster (Pil opp og Pil Ned) for å bla i tidligere brukte kommandoer i Command Window

→ En grei regel: En oppgave – en fil, dvs. ikke putt alle oppgavene i en fil!!

→ Funksjoner:

- **Kun en funksjon i hver fil!**

- **Filnavnet (.m) og navnet på funksjonen må være det samme!**

→ Bruk semikolon ";" etter kommandoer/funksjoner hvis du ikke trenger å vise svaret på skjermen

```
a=2;  
b=4;  
y=a+b
```

→ Bruk engelske navn på variable, funksjoner, filer, m.m. Dette er vanlig praksis i programmering!

Bruk alltid variable - ikke sett inn tall direkte i uttrykkene

```
a=2;
```

```
b=4;
y=a+b
```

Ikke:

```
y=2+4
```

→ **Bruk hjelp** for å finne ut mer om de funksjonene du skal bruke

For å få hjelp om tf funksjonen skriver du følgende i Command window:

```
help tf
```

→ I toppen av Scriptet ditt, legg alltid til følgende kommandoer:

```
clear
clc
close all
...
```

Dette sørger for at du ikke får problemer med gamle variable, m.m.

→ Greske bokstaver: I matematikken og i reguleringsteknikk er det vanlige å bruke greske

Bokstaver i formler, m.m. Disse kan ikke brukes direkte i MathScript, så finn på gode variabelnavn for disse. Eksempler:

ω_0 – w0

ζ – zeta eller eventuelt bare z

osv.

→ Matematiske uttrykk: Bruk følgende i MathScript:

x^2	x^2
\sqrt{x}	sqrt(x)
$\ln(x)$	log(x)
$\log(x)$	Log10(x)
e^x	exp(x)
π	pi

Eksempel:

Anta følgende matematiske funksjon:

$$z = 3x^2 + \sqrt{x^2 + y^2} + e^{\ln(x)}$$

Hvor z er en funksjon av x og y , dvs. $z(x, y)$.

Vi ønsker å bruke MathScript til å finne $z(2,2)$.

MathScript koden blir som følger:

```
x = 2;  
y = 2;  
z = 3*x^2 + sqrt(x^2 + y^2) + exp(log(x))
```

MathScript gir følgende svar:

$z = 16.8284$

Merk! Vi må alltid definere verdier for variablene (x og y) før vi bruker dem i et uttrykk.

17 MathScript- funksjoner

Her er en kort oversikt over MathScript funksjoner som vi har benyttet.

Funksjon	Beskrivelse	Eksempler
atan	Beregner arctangens til x	<code>>atan(x)</code>
bode	Lager et Bodediagram for en gitt transferfunksjon. Eller den kan returnere forsterkning og faseforskyvning for gitte frekvenser. Hvis du ikke spesifiserer noen utganger fra funksjonen vil funksjonen tegne et Bodediagram.	<code>>num = [4]; >den = [2, 1]; >H = tf(num, den) >bode(H)</code>
bodemag	Nesten same som bode funksjonen, men denne lager kun forsterkningsplottet – ikke faseforskyvningsplottet.	<code>>[mag, wout] = bodemag(SysIn) >[mag, wout] = bodemag(SysIn, [wmin wmax]) >[mag, wout] = bodemag(SysIn, wlist)</code>
conv	Kan brukes når du har komplekse transferfunksjoner som du må slå sammen.	<code>>den1 = [1, 2, 3]; >den2 = [3, 4]; >den = conv(C1, C2)</code>
feedback	Lager transferfunksjonen for et tilbakekoblet system.	<code>>T = feedback(H1, H2)</code>
log10	Beregner 10'er logaritmen.	<code>>log(x)</code>
lsim	Brukes til å simulere et lineært system.	<code>>t = [0:0.1:10] >u = sin(0.1*pi*t)' >lsim(H, u, t)</code>
margin	Tegner et Bodediagram med forsterkningsmargin og fasemargin tegnet inn.	<code>>num = [1] >den = [1, 5, 6] >H = tf(num, den) margin(H)</code>
margin	Denne funksjonen kan også beregne kryssfrekvens, fasekryssfrekvensen, forsterkningsmargin og fasemargin	<code>>[gmF, gm, pmF, pm] = margin(H)</code>
pade	Brukes til å finne transferfunksjonen til en Pade' aproksimasjon.	<code>>[num, den] = pade(delay, order) >[A, B, C, D] = pade(delay, order)</code>
pid	Brukes til å lage transferfunksjonen til en P, PI, PD eller PID regulator.	<code>>Kc = 0.5; >Ti = 0.25; >H = pid(Kc, Ti, 'academic');</code>
plot	Brukes til å generere et plot.	<code>>x = [0:0.01:1]; >y = sin(x) >plot(x, y)</code>
poles	Finner polene for en transferfunksjon	<code>>num = [1] >den = [1,1] >H = tf(num,den) >poles(H)</code>
semilogx	Kan brukes til å tegne Bodediagram "manuelt" basert på de matematiske uttrykkene for $A(\omega)$ og $\phi(\omega)$	<code>>semilogx(w, gain)</code>
series	Sla sammen 2 eller flere transferfunksjoner som er i serie	<code>>H = series(H1,H2)</code>
ss	Lager en tilstandsrommodell fra A, B, C og D matrisene.	<code>>A = [1, 2; 3; 4] >B = [0; 1] >C = [1, 0] >model = ss(A, B, C)</code>
ssinfo	Brukes til å returnere A, B, C og D matrisene i tilstandsrommodellen.	<code>>A = [1, 1; -1, 2] >B = [1, 2]' >C = [2, 1] >D = 0 >model = ss(A, B, C, D) >[A, B, C, D, Ts] = ssinfo(model)</code>
step	Sprangrespons	<code>>num=[1,1]; >den=[1,-1,3]; >H=tf(num,den); >t=[0:0.01:10]; >step(H,t);</code>
Sys_order1	Lager en 1.ordens transferfunksjon, med eller uten tidsforsinkelse.	<code>>K = 1; >T = 2; >H = sys_order1(K, T)</code>
Sys_order2	Lager en 2.ordens transferfunksjon.	<code>>delay = 3; >H = sys_order1(K, T, delay) >z = 0.5 >w = 20 >[num, den] = sys_order2(w, z) >H = tf(num, den) >[A, B, C, D] = sys_order2(wn, dr)</code>

tf	Brukes til å lage en transferfunksjon. Kan også brukes til å transformere en tilstandsrommodell til en transferfunksjon.	<pre>>SysSS = ss(A, B, C, D) >num=[1]; >den=[1, 1, 1]; >H = tf(num, den)</pre>
tfinfo	Brukes til å returnere teler, nevner og eventuelt dødtid fra en transferfunksjon.	<pre>>[num, den, delay, Ts] = tfinfo(H)</pre>

Skriv “**help <funksjonsnavn>**” i Command window for å finne ut hvordan den enkelte funksjon virker i detalj.

18 DAQ i LabVIEW

LabVIEW er veldig anvendelig ifm I/O og datainnsamling.

18.1 USB-6008

Et eksempel på en DAQ enhet kan være NI USB-6008:



Enheten har følgende spesifikasjoner:

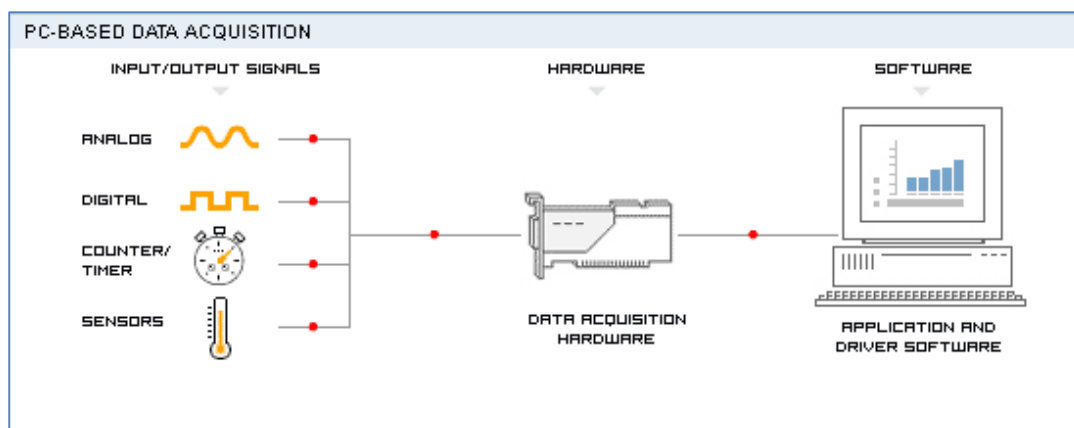
- 8 analoge innganger (12-bit, 10 kS/s)
- 2 analoge utganger (12-bit, 150 S/s)
- 12 digitale I/O
- USB tilkobling, ikke behov for ekstra strømforsyning
- Kompatibel med LabVIEW og Visual Studio
- Bruker NI-DAQmx driver software

Denne blir brukt i mange fag ved Høgskolen i Telemark.

Denne kan enkelt knyttes opp mot PC og brukes ifm blant andre LabVIEW.

18.2 DAQ

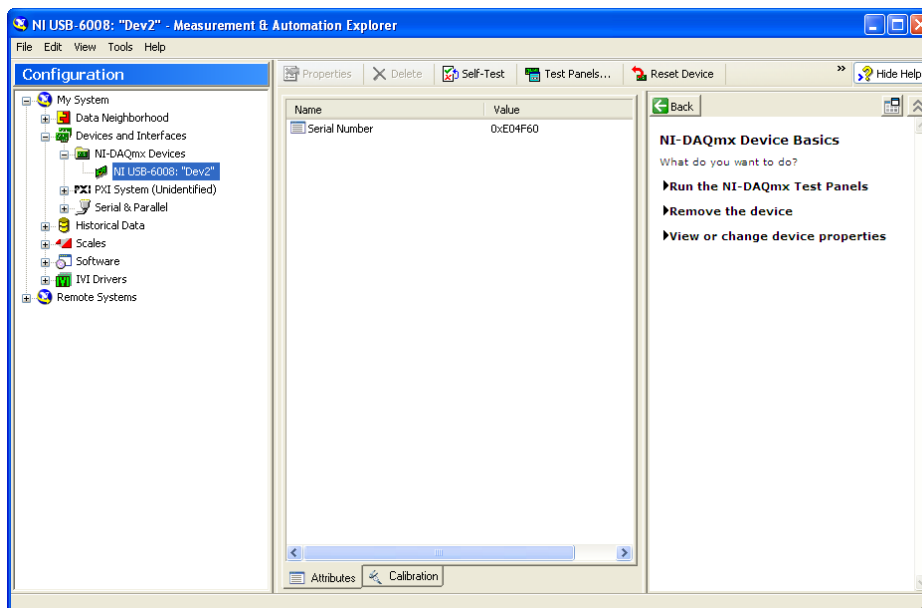
Grunnleggende DAQ prinsipp:



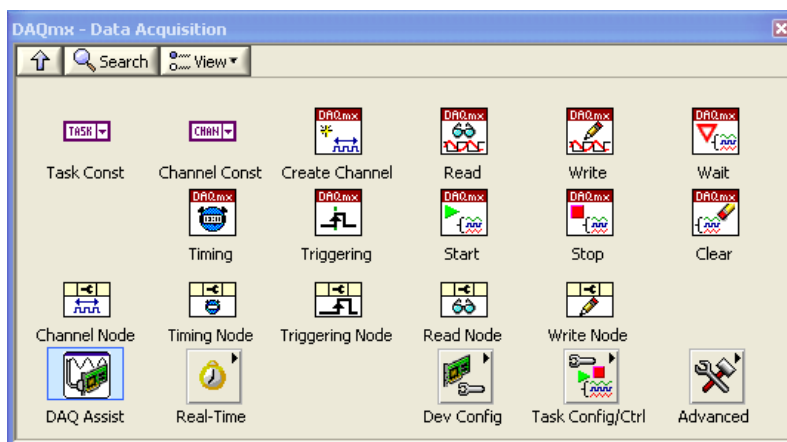
18.3 Bruk av NI USB-6008 i LabVIEW

NI-DAQmx er software-driveren som du må installere.

MAX: Measurement & Automation Explorer kan brukes til å konfigurere DAQ enhetene fra National Instruments.

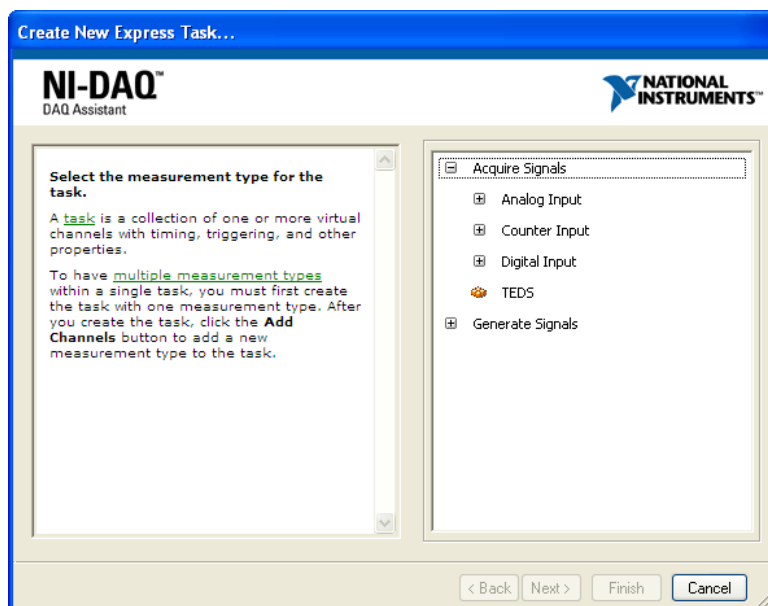


DAQmx paletten samt DAQ Assistant i LabVIEW:



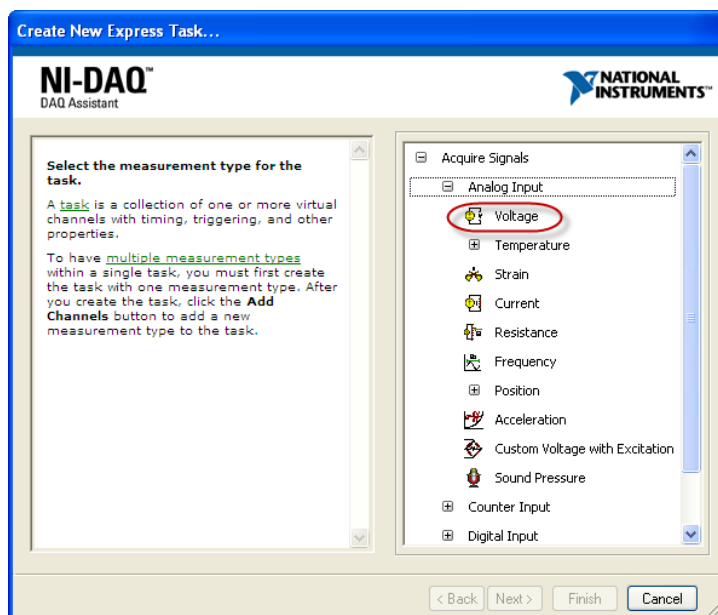
18.3.1 Analog Input

Dra og slipp "DAQ Assistant" ikonet på blokkdiagrammet ditt og følgende vindu dukker opp:

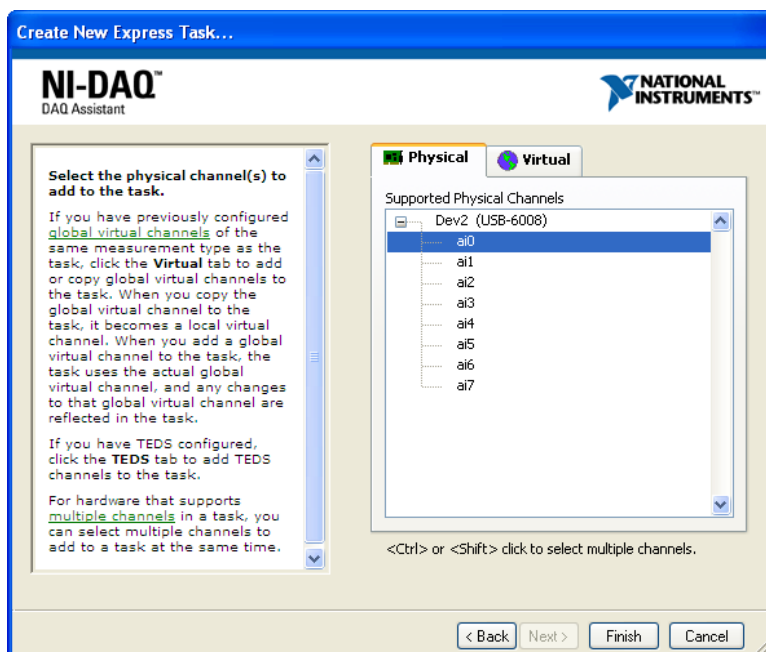


Her kan du velge enten "Acquire Signals" (inngangssignaler) eller "Generate Signals" (utgangssignaler).

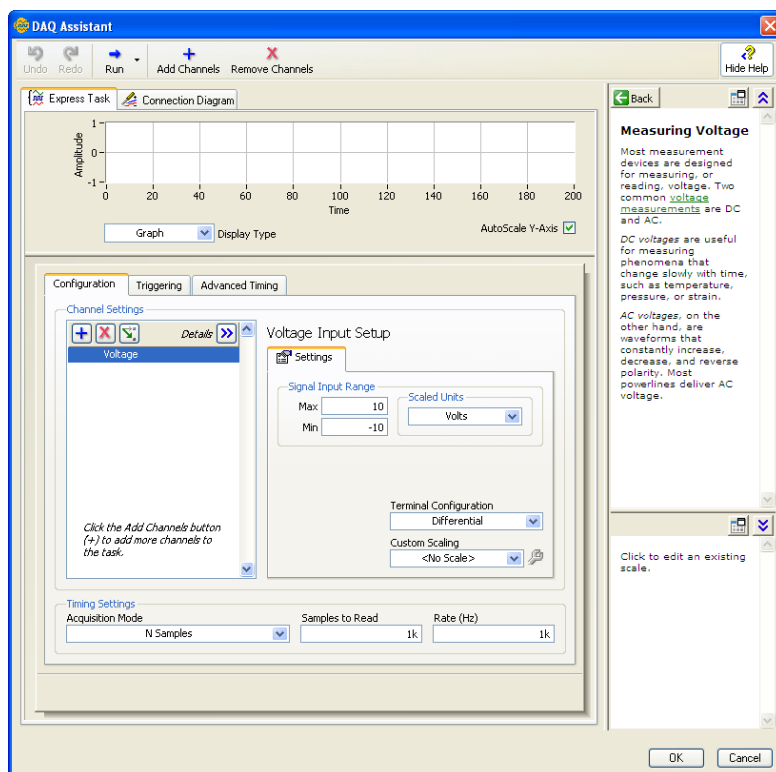
Velg "Acquire Signals" → "Analog Input" → "Voltage".



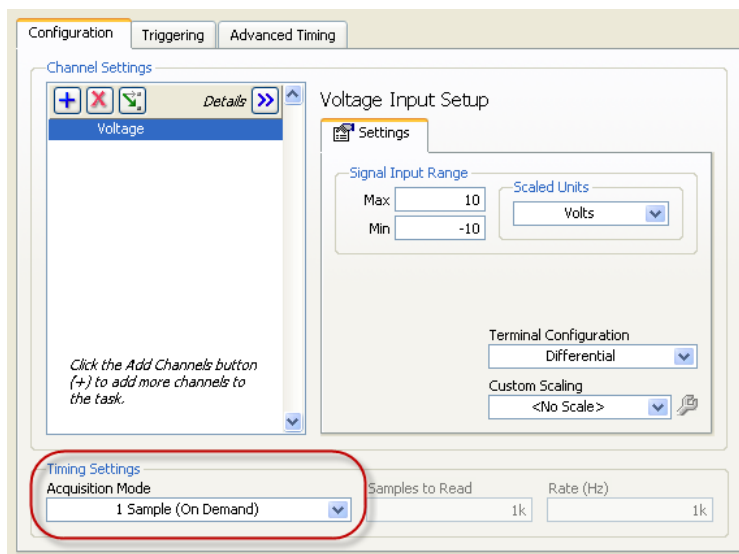
Deretter velger du riktig kanal. Typisk velger du "ai0" (Analog Input channel 0). Trykk "Finish".



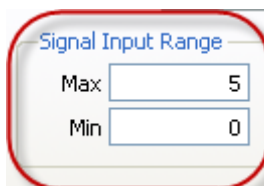
Følgende vindu dukker opp:



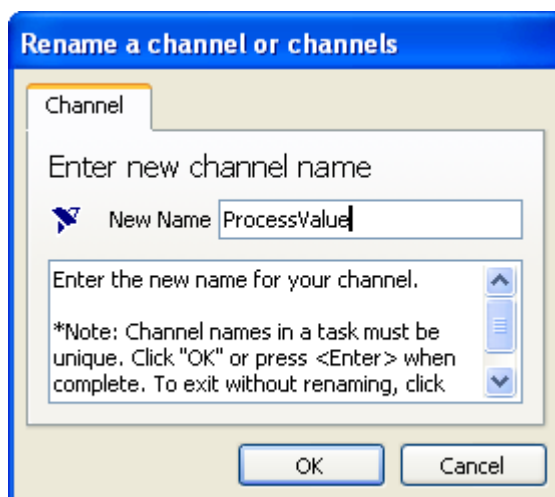
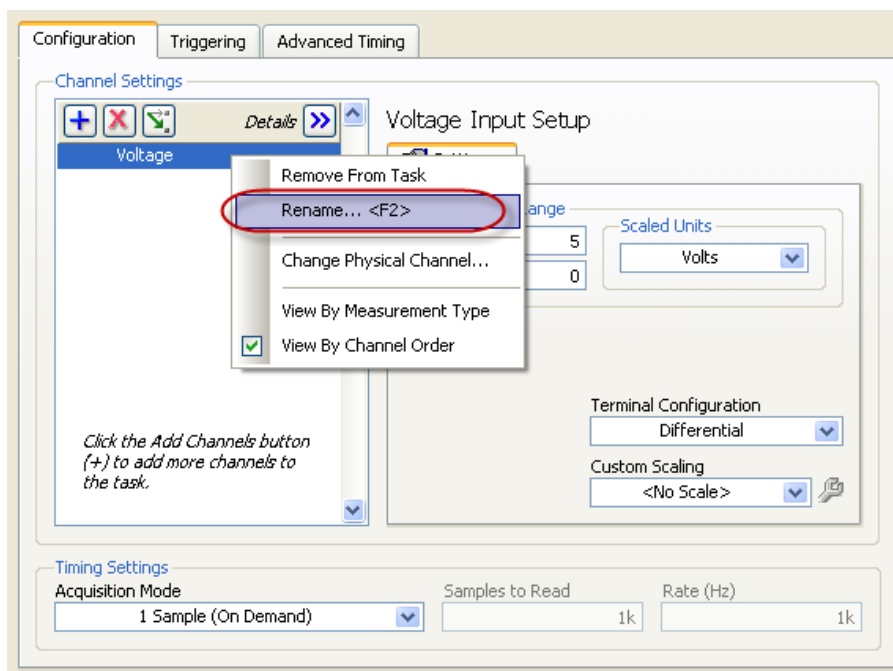
Merk! Sørg for a velge "Timing Settings" -> "1 Sample (On Demand)".



Velg deretter "Signal Input Range". Typisk velger du 0-5V.



Du kan også forandre navnet på kanalen (høyreklikk og velg "Rename..."):



Du er nå ferdig med konfigureringen. Trykk OK i "DAQ Assistant" vinduet og følgende ikon dukker opp på blokkdiagrammet ditt:



Du er nå klar til å bruke USB-6008 DAQ enheten i ditt LabVIEW program.

Eksempel:

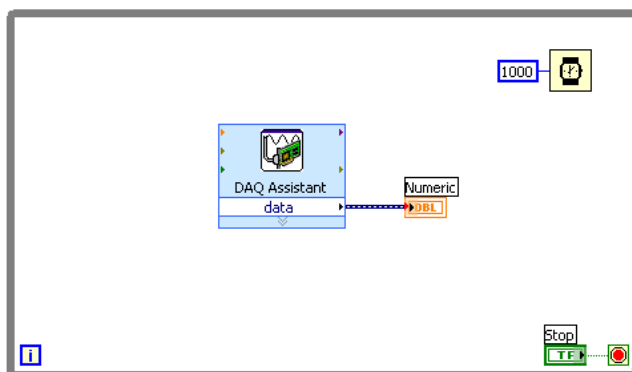
Koble "data" utgangen til en numerisk indiaktor som vist nedenfor og trykk på "Run" knappen:



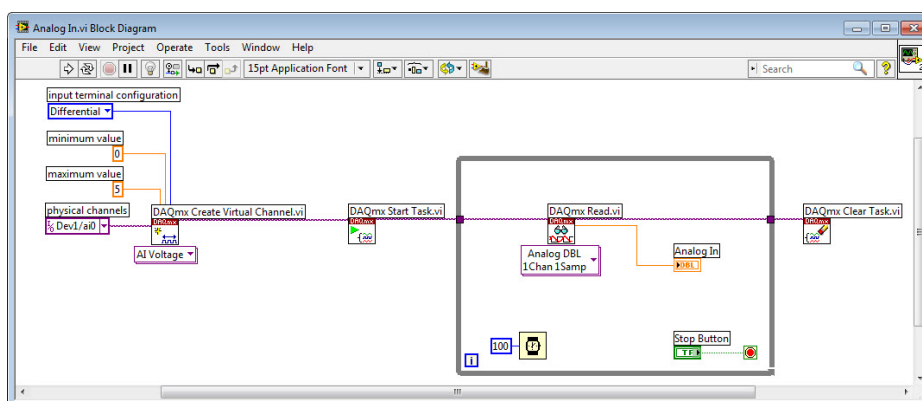
Den numeriske indiaktoren vil vise spenningssignalet som evt er koblet til DAQ enheten, for eksempel:



Hvis datainnsamlingen skal foregå "kontinuerlig", bruk en While loop rundt DAQ Assistant som vist nedenfor:



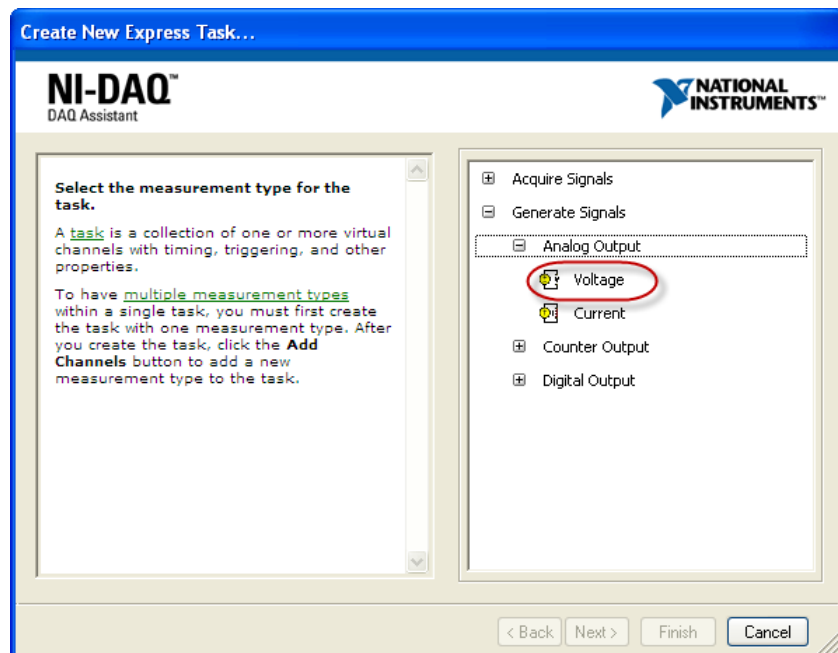
For mer avanserte programmer med høyere krav til ytelse kan vi bruke såkalte "lavnivå" funksjoner:

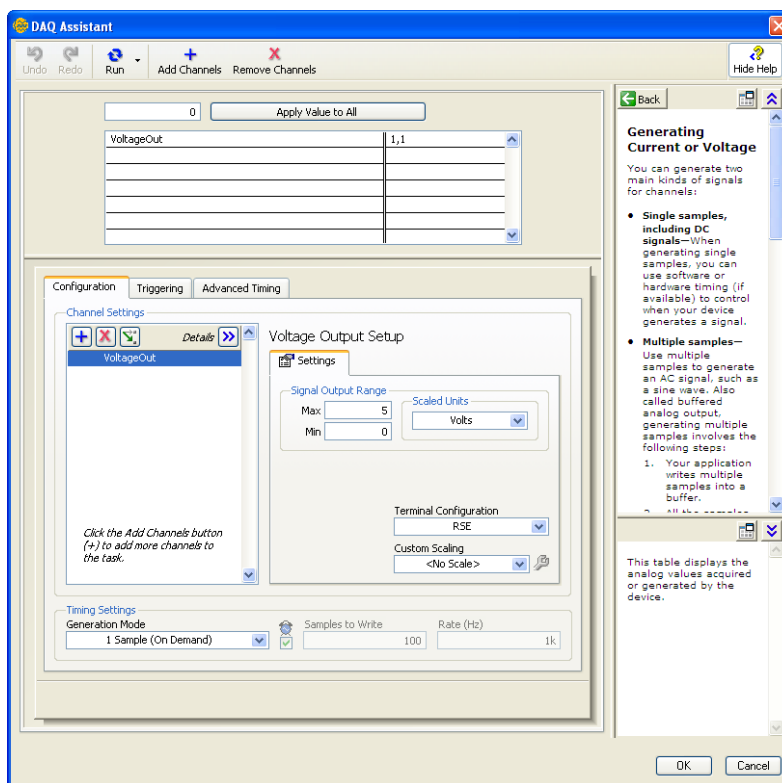


[Slutt på eksempel]

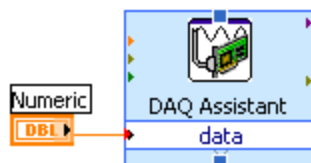
18.3.2 Analog Output

Analog ut er tilsvarende:

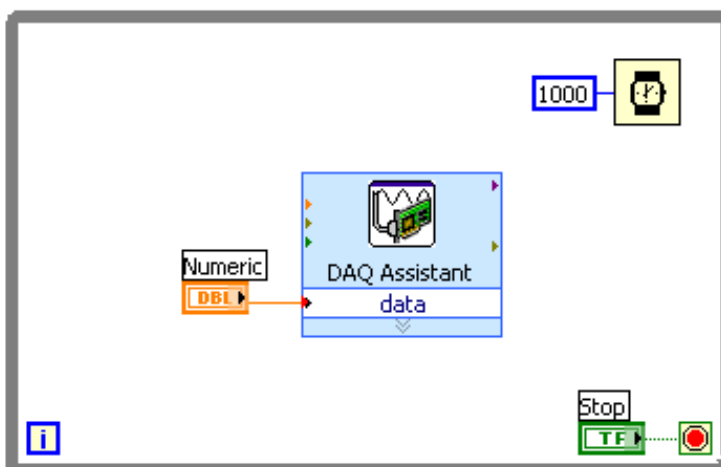




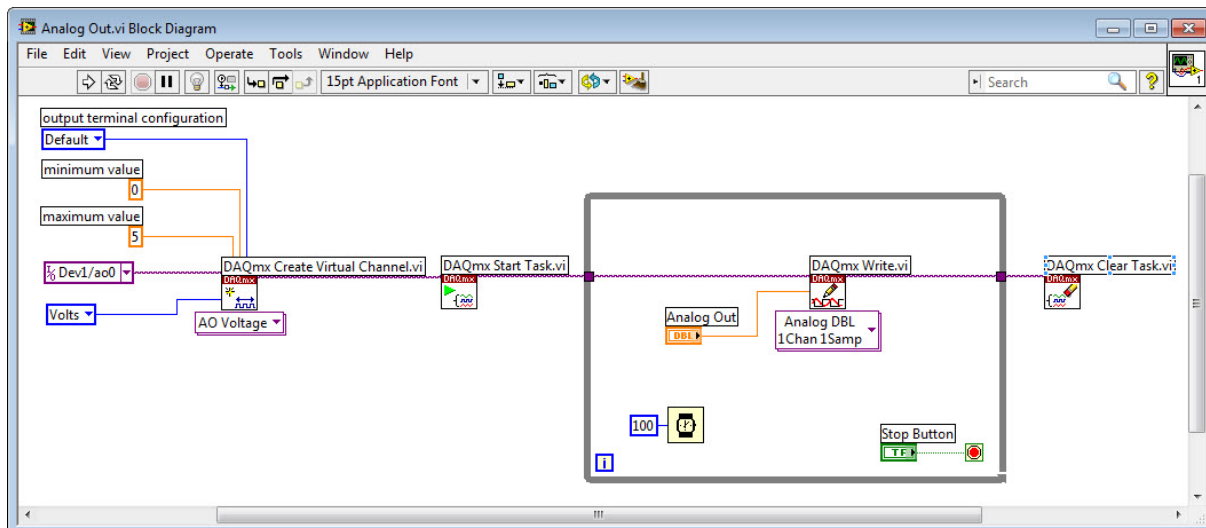
Eksempel:



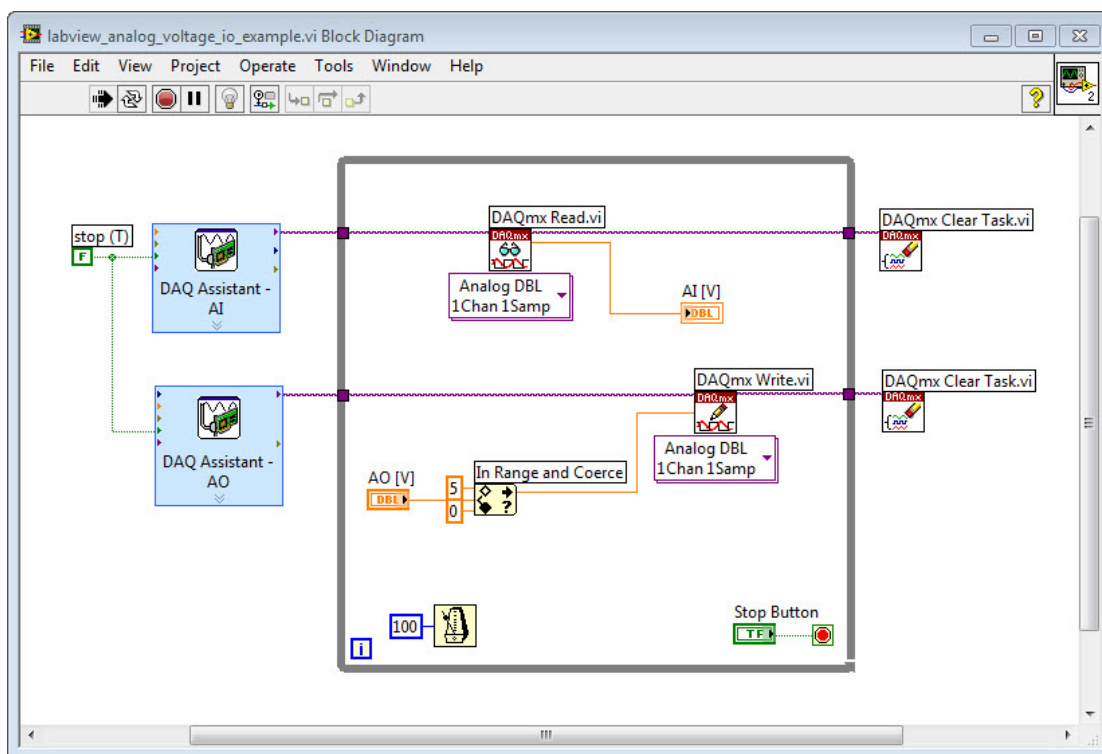
Eller inni en While Loop:



Eller mer avansert bruk:

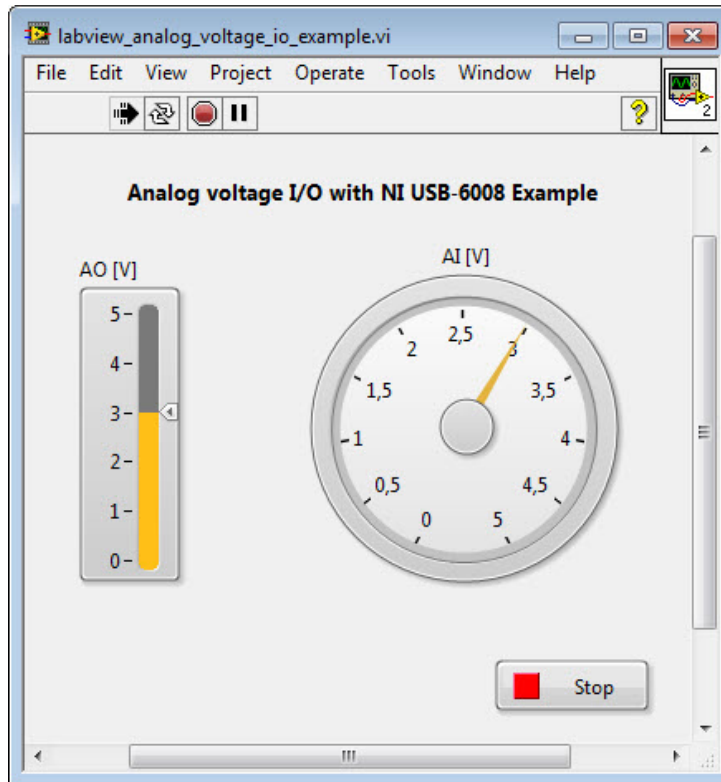


Nedenfor ser vi et eksempel hvor DAQ Assistant er plassert på utsiden av While løkken, samt at både Analog ut og Analog inn kjører samtidig:



Ved at DAQ Assistant er plassert på utsiden av While løkka slik som dette kan gi noe bedre ytelse.

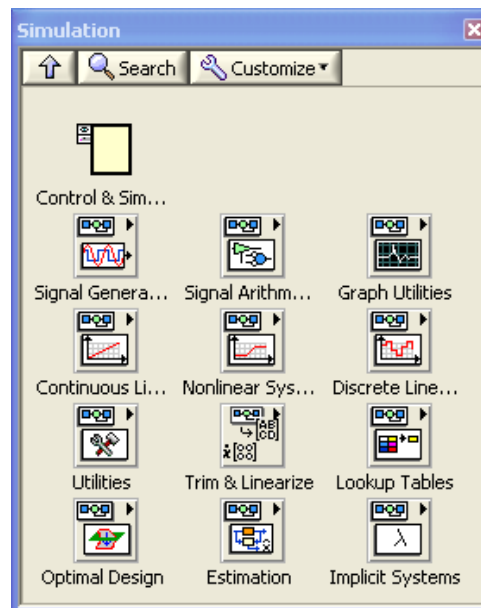
Frontpanel:



19 Simulering og regulering i LabVIEW

LabVIEW egner seg veldig godt til simulering, enten på bakgrunn av differensiallikninger, blokkdiagrammer eller transferfunksjoner.

Nedenfor ser vi "Simulation" paletten i LabVIEW:



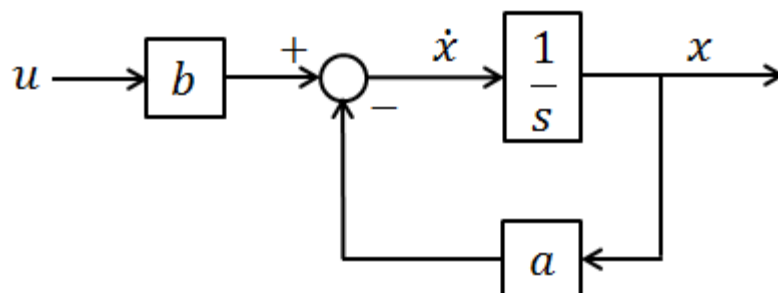
Fundamentet for simuleringene er en såkalt "**Control & Simulation Loop**", som er lokalisert i øvre venstre hjørne i "Simulation" paletten. Control & Simulation Loop er veldig lik en ordinær "While loop", men den har noen finessersom kan være nyttige ifm simuleringer. Forøverig kan en vanlig While Loop også benyttes, men da må vi lage et såkalt "Simulation Sub System" som vi plasserer inni While Loopen.

Eksempel:

Gitt følgende system

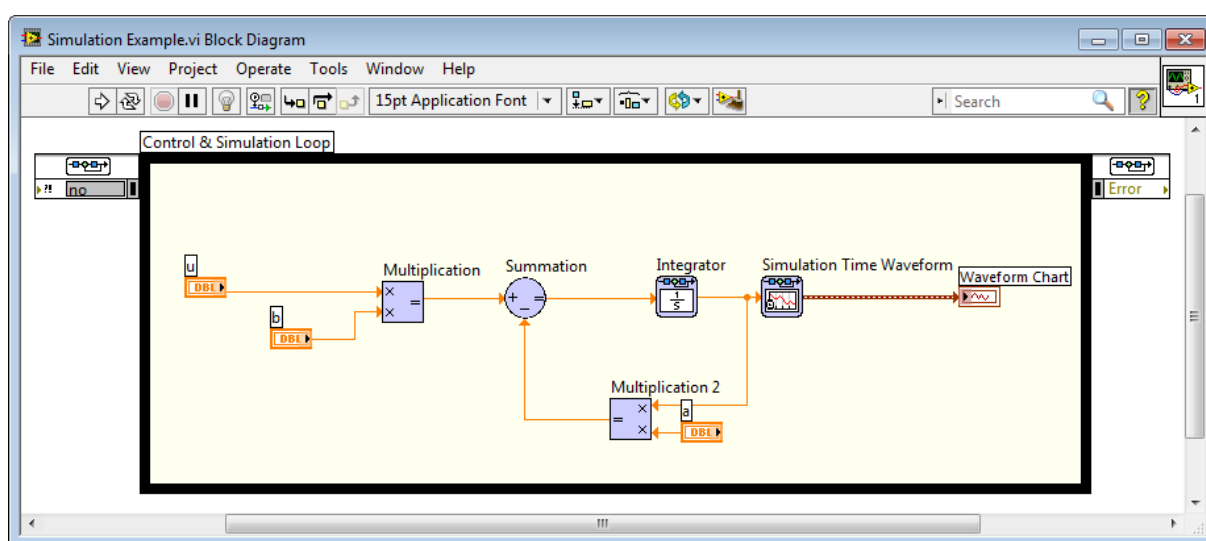
$$\dot{x} = -ax + bu$$

Blokkdiagrammet blir som følger:

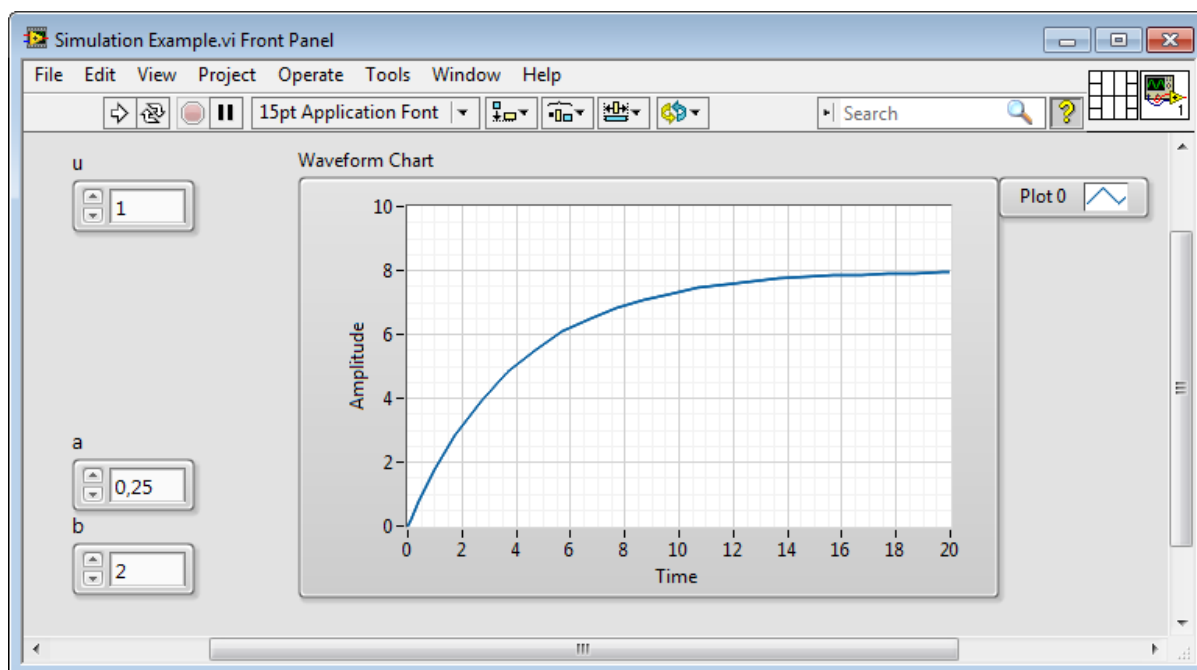


LabVIEW Implementering:

I LabVIEW kan vi gjøre det slik:

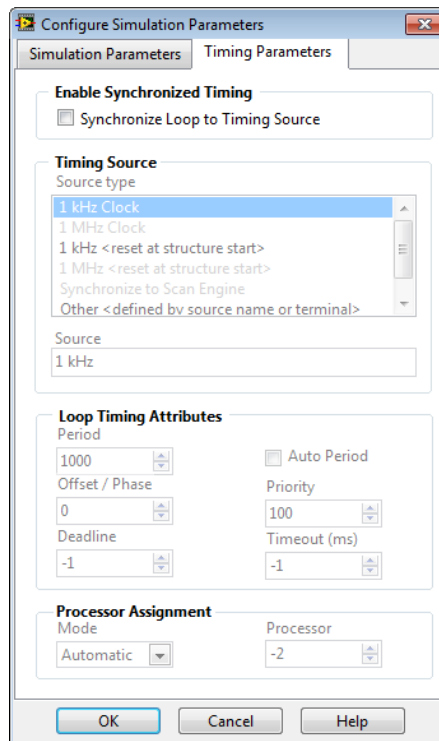
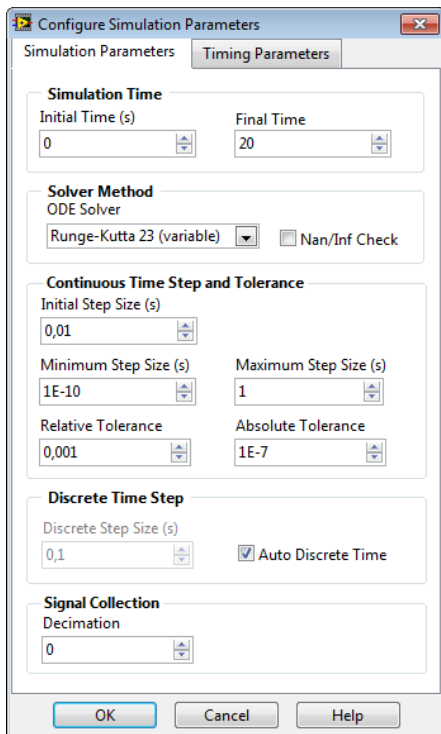


Sett $a = 0.25$ og $b = 2$. Da får vi følgende:



Simulation Parametere:

I eksempelet er følgende parametre brukt (høyreklikk på Simulation Loop kanten og velg "Configure Simulation parameters..."):



Transferfunksjon:

Vi kan finne transferfunksjonen vha Laplace:

$$sx(s) = -ax(s) + bu(s)$$

Som gir følgende transferfunksjon:

$$H(s) = \frac{x(s)}{u(s)} = \frac{b}{s+a} = \frac{b/a}{\frac{1}{a}s+1} = \frac{K}{Ts+1}$$

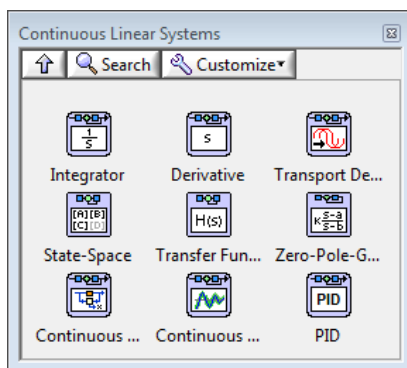
Dvs. $K = b/a$ and $T = 1/a$

Med $a = 0.25$ and $b = 2$ får vi følgende:

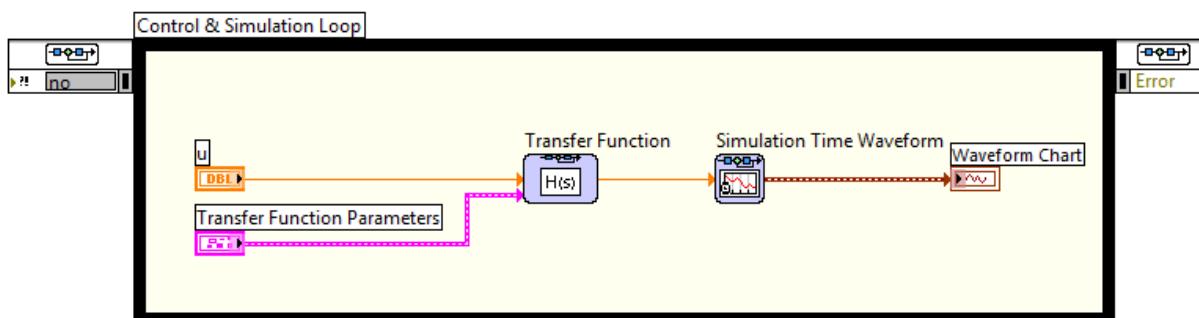
$$H(s) = \frac{x(s)}{u(s)} = \frac{8}{4s+1}$$

Vi ser fra plottet over at dette er korrekt ($K = 8$ og $T = T_{63} = 4s$).

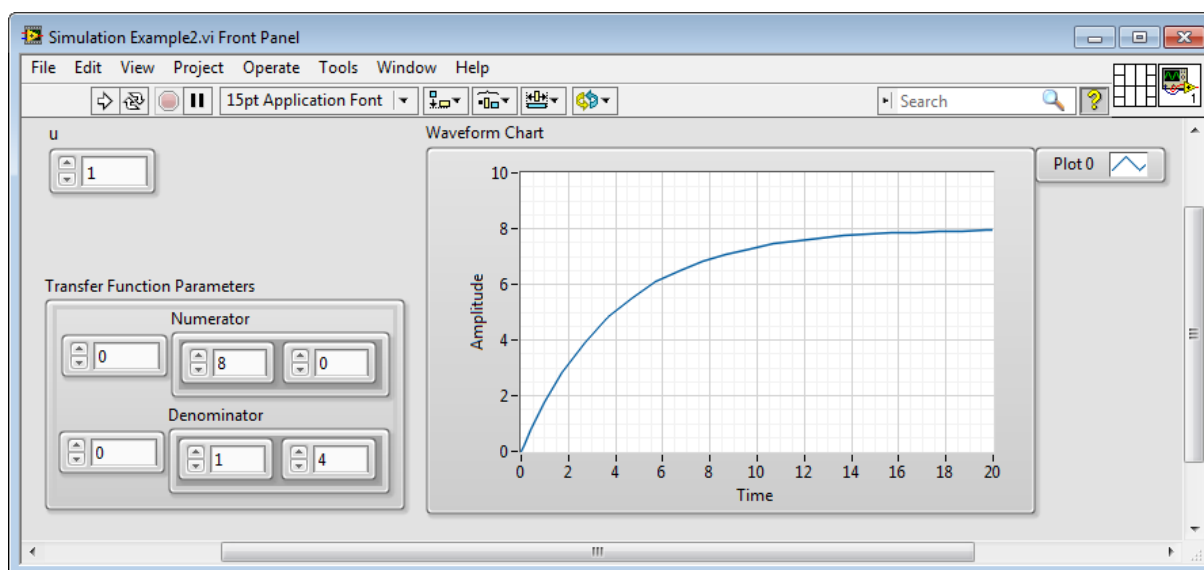
I "Continuous Linear Systems" paletten i LabVIEW har en funksjon for å definere en transferfunksjon:



I LabVIEW ser det slik ut:



Detet gir samme resultat:



Merk! Rekkefølgen på koeffisientene er forskjellig i LabVIEW og MathScript.

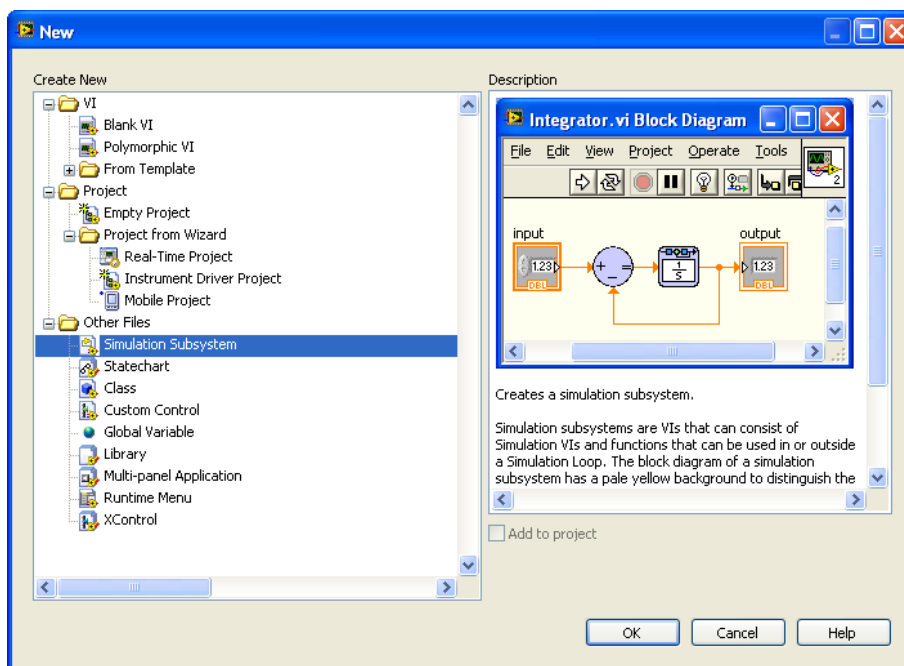
I LabVIEW gjør vi følgende:

$$H(s) = \frac{b_0 + b_1s + \dots + b_{m-1}s^{m-1} + b_m s^m}{a_0 + a_1s + \dots + a_{n-1}s^{n-1} + a_n s^n}$$

Prøv med forskjellige verdier for a and b og observer hva som skjer – er resultatene som forventet ihht teorien for 1.ordens systemer?

19.1 Simulation Subsystem

Vi kan lage et såkalt **Simulation Subsystem** (File → New...):

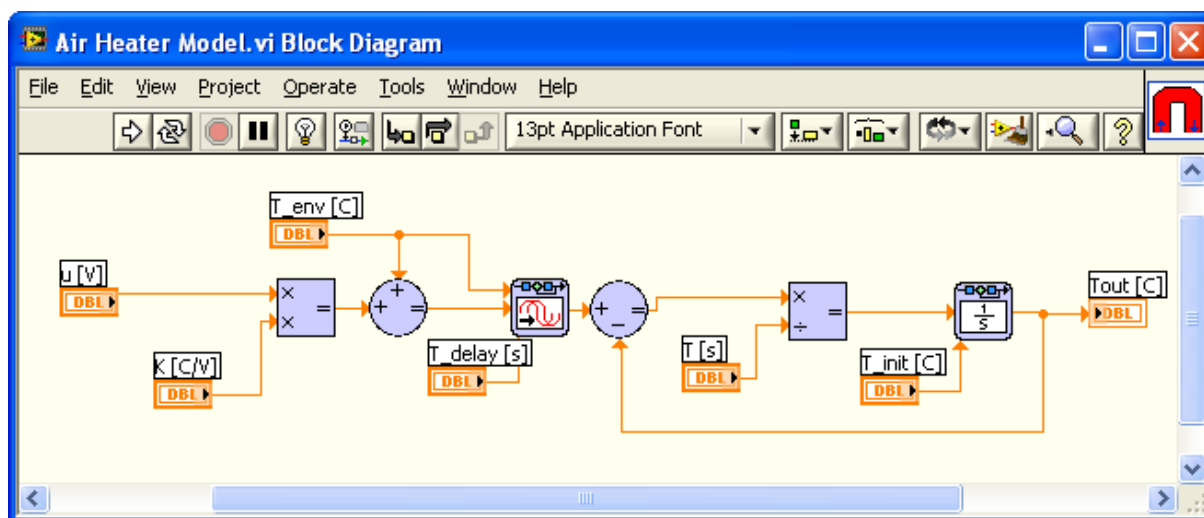


Når vi lager større programmer eller ønsker å gjenbruke kode er dette en god ide.

Simulation Subsystem er nesten identisk som et normalt blokkdiagram, men bakgrunnsfargen er noe mørkere (akkurat som bakgrunnsfargen inni en Simulation Loop).

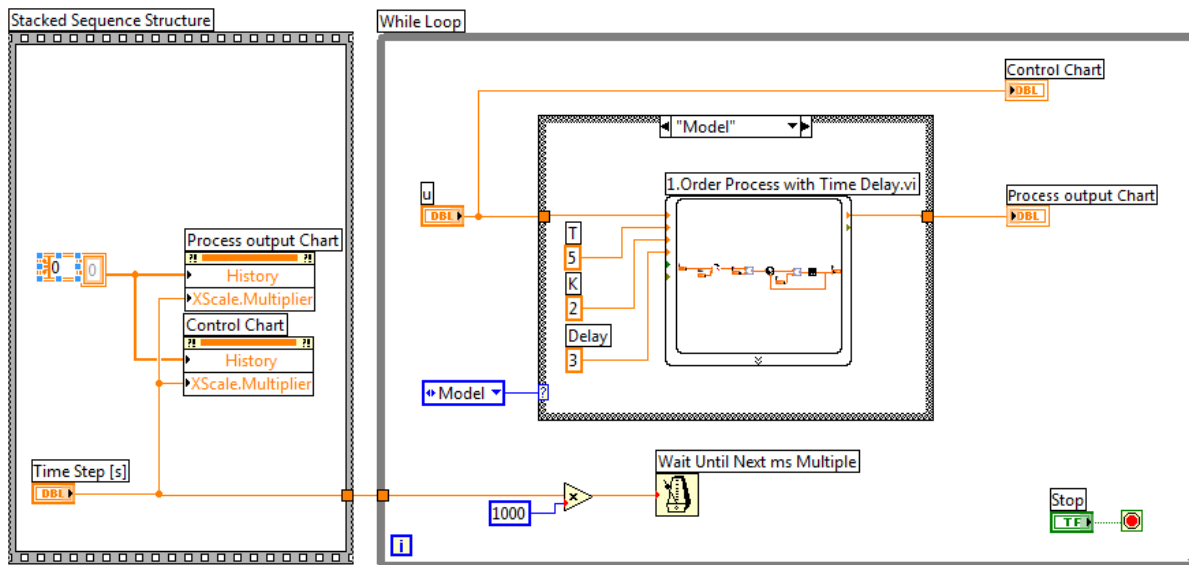
Merk! For å åpne et Simulation Subsystem, høyreklikk og velg "Open Subsystem".

Nedenfor ser vi et eksempel på et Simulation Subsystem:



Som du ser trenger vi ikke å bruke en Simulation Loop når vi bruker et Simulation Subsystem.

Nedenfor der vi et eksempel på et Simulation Subsystem som brukes inni en vanlig While løkke.

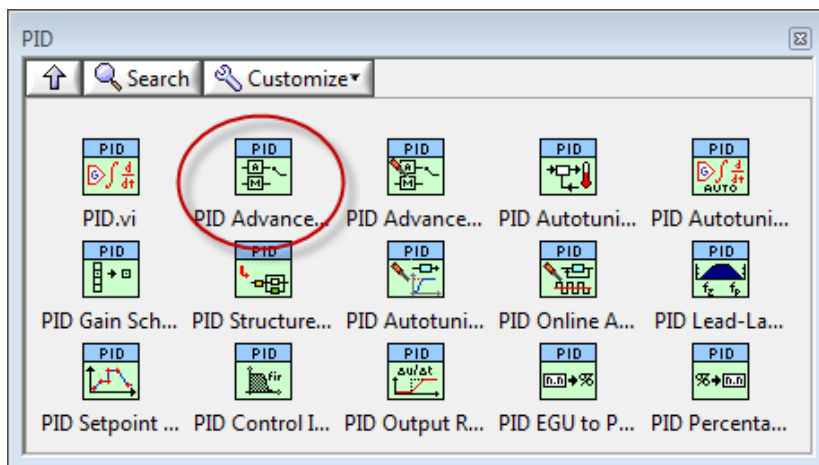


Fordelen er at man enkelt kan gjenbruke den samme kode i mange. En annen fordel er at vi kan bruke en vanlig While løkke istedenfor en Simulation Loop.

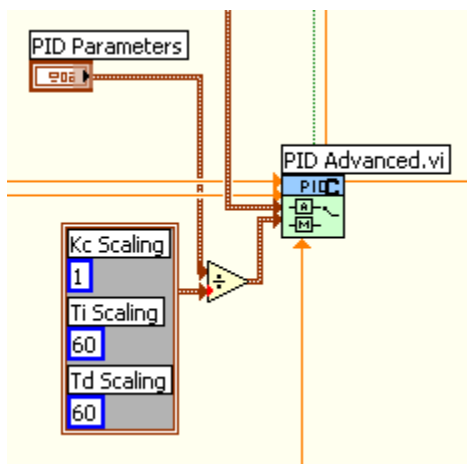
Denne fremgangsmåten vil jeg anbefale å bruke i de fleste tilfellene.

19.2 PID

Du finner de nødvendige PID funksjonene i PID paletten:



Merk! Funksjonen "PID.vi" og "PID Advanced.vi" krever at T_i and T_d er i minutter, mens det er mer normalt å bruke sendunder på disse parametrene. Følgende kodesnutt kan brukes for å løse dette:



20 Diskretisering

20.1 Innledning

Det finnes mange forskjellige diskretiseringsmetoder, vi vil fokusere på de som er mest brukt/enklest å bruke.

Euler:

Vi har 2 forskjellige varianter:

Euler forward metoden:

$$\dot{x} \approx \frac{x(k+1) - x(k)}{T_s}$$

Euler backward metoden:

$$\dot{x} \approx \frac{x(k) - x(k-1)}{T_s}$$

Andre metoder er Zero Order Hold (ZOH), Tustin's method, osv.

Hvor T_s er smaplingstiden og $x(k+1)$, $x(k)$ and $x(k-1)$ er diskrete verdier.

Merk! Ulik ntasjon blir brukt i ulik litteratur: $x(k) = x_k = x(t_k)$, $x(k+1) = x_{k+1} = x(t_{k+1})$, $x(k-1) = x_{k-1} = x(t_{k-1})$, osv.

Eksempel:

Gitt følgende system:

$$\dot{x} = -ax + bu$$

Vi bruker Euler forward metoden for å lage en diskret versjon av systemet.

Vi får da:

$$\frac{x(k+1) - x(k)}{T_s} = -ax(k) + bu(k)$$

Videre:

$$x(k+1) = x(k) + T_s[-ax(k) + bu(k)]$$

Dette gir:

$$x(k+1) = x(k) - T_s a x(k) + T_s b u(k)]$$

Til slutt får vi da:

$$\underline{\underline{x(k+1) = (1 - T_s a)x(k) + T_s b u(k)}}$$

Hvis du bruker $a = 0.25$, $b = 2$ and $T_s = 0.1s$, får vi følgende:

$$x(k+1) = -0.975x(k) + 0.2u(k)$$

[Slutt på eksempel]

Eksempel:

Gitt følgende system:

$$\dot{x}_1 = K_p u - x_2$$

$$\dot{x}_2 = 0$$

Den diskrete versjonen blir da:

$$\frac{x_1(k+1) - x_1(k)}{T_s} = K_p u(k) - x_2(k)$$

$$\frac{x_2(k+1) - x_2(k)}{T_s} = 0$$

$$y(k) = x_1(k)$$

Dette gir følgende:

$$x_1(k+1) = x_1(k) + T_s K_p u(k) - T_s x_2(k)$$

$$x_2(k+1) = x_2(k)$$

$$y(k) = x_1(k)$$

Vi kan også sette det på tilstandsromform:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & -T_s \\ 0 & 1 \end{bmatrix}}_A \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \underbrace{\begin{bmatrix} T_s K_p \\ 0 \end{bmatrix}}_B u(k)$$

$$y = x_1(k)$$

[Slutt på eksempel]

20.2 Diskretisering i MathScript

Vha MathScript kan vi simulere diskrete systemer ved å bruke en For Loop eller en While Loop.

Vi kan også bruke de innebygde funksjonene for å konvertere kontinuerlige systemer til diskrete systemer. I MathScript kan vi bruke funksjonen `c2d()` til å konvertere fra et kontinuerlig system til et diskret system.

Eksempel:

Vi skal simulere følgende system i MathScript ved bruk av en For Loop:

$$x(k + 1) = (1 - T_s a)x(k) + T_s b u(k)$$

Vi setter $a = 0.25$ og $b = 2$

MathScript kode:

```
% Simulation of discrete model
clear
clc

% Model Parameters
a = 0.25;
b = 2;

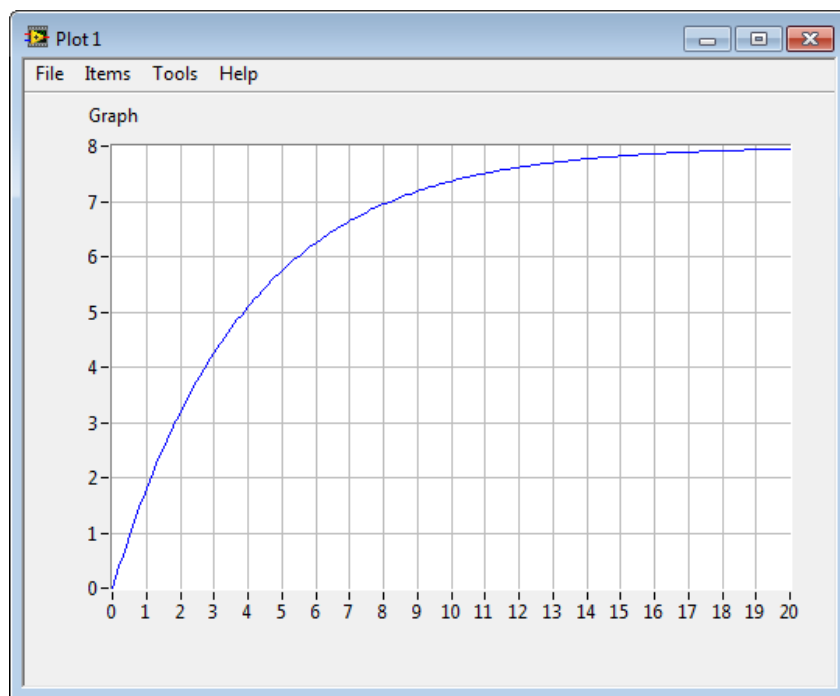
% Simulation Parameters
Ts = 0.1; %s
Tstop = 20; %s
uk = 1;
x(1) = 0;

% Simulation
for k=1:(Tstop/Ts)
    x(k+1) = (1-a*Ts).*x(k) + Ts*b*uk;
end

% Plot the Simulation Results
k=0:Ts:Tstop;
plot (k, x)
grid on
```

Vi bruker $T_s = 0.1s$ (samplingstiden).

Dette gir følgende resultat:



[Slutt på eksempel]

Eksempel:

Gitt følgende system:

$$\dot{x} = -ax + bu$$

Vi setter $a = 0.25$ og $b = 2$

Vi bruker den innebygde funksjonen `c2d()` for å finne det diskrete systemet:

```
% Find Discrete model
clear
clc

% Model Parameters
a = 0.25;
b = 2;
Ts = 0.1; %s

% State-space model
A = [-a];
B = [b];
C = [1];
D = [0];

model = ss(A,B,C,D)
model_discrete = c2d(model, Ts, 'forward')
```

Dette gir følgende resultat:

a	b	c	d
0.975	0.2	1	0

eller:

$$x(k+1) = -0.975x(k) + 0.2u(k)$$

$$y(k) = x(k)$$

Som du ser får vi samme resultat.

[Slutt på eksempel]

20.2.1 Vanntank

Nedenfor ser vi en enkel vanntankmodell:



Differensiallikningen for systemet er som følger:

$$A_t \dot{h} = K_p u - F_{out}$$

eller:

$$\dot{h} = \frac{1}{A_t} [K_p u - F_{out}]$$

hvor:

- h [cm] er nivået i vanntanken
- u [V] er pådragssignalet til pumpa

- A_t [cm^2] arealet til bunnen i vanntanken
- K_p [$(cm^3/s)/V$] er pumpeforsterkningen
- F_{out} [cm^3/s] er utstrømningen gjennom ventilen i bunnen av tanken.

Diskretisering:

Vi bruker Euler Forward diskretiseringsmetoden:

$$\dot{x} \approx \frac{x_{k+1} - x_k}{T_s}$$

hvor T_s er samplingstiden i sekunder.

Dette gir følgende diskrete system:

$$\frac{h_{k+1} - h_k}{T_s} = \frac{1}{A_t} [K_p u_k - F_{out}]$$

Dette gir

$$h_{k+1} = h_k + T_s \frac{1}{A_t} [K_p u_k - F_{out}]$$

Eller:

$$h_{k+1} = h_k - \frac{T_s}{A_t} F_{out} + \frac{T_s K_p}{A_t} u_k$$

Vi ønsker å implementere modellen i MathScript og simulere den, dvs finne sprangresponsen for systemet.

Vi kan bruke følgende verdier:

$$K_p = 15(cm^3/s)/V$$

$$A_t = 80cm^2$$

$$F_{out} = 10cm^3/s$$

$$T_s = 0.1s$$

Vha. MathScript kan vi simulere systemet:

```
clear, clc
% Model

% Model parameters
Kp = 15;
At = 80;
Fout = 10;
```

```

h(1) = 0; %Initial value
Ts = 0.1; %Sampleing Time
uk = 1; %Step

k = 1:100;
N = length(k);

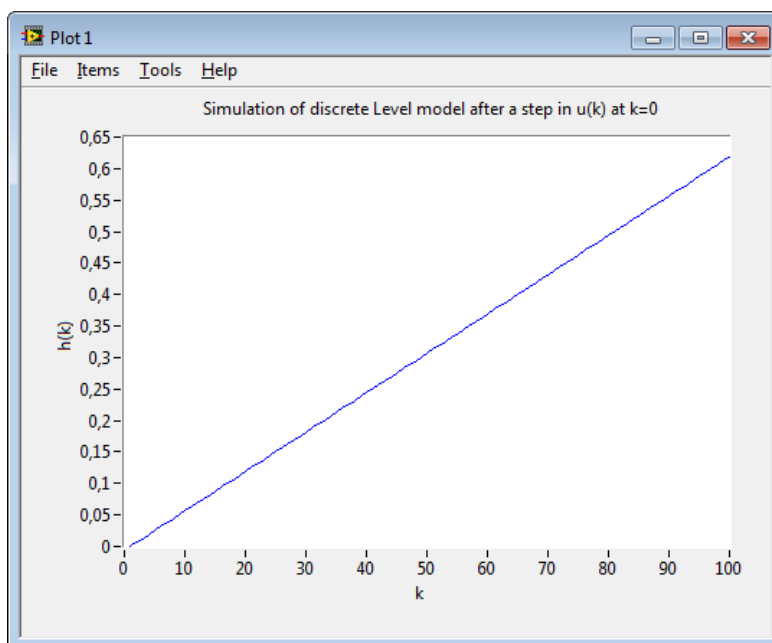
for i=2:N
    h(i) = h(i-1) + (Ts/At)*(Kp*uk-Fout);
end

plot(k, h)
title('Simulation of discrete Level model after a step in u(k)
at k=0')
xlabel('k')
ylabel('h(k)')

```

Merk! Her har vi diskretisert modellen og simulert vha en For Loop. Vi vil lære mer om diskretisering senere.

Dette gir følgende resultat:



Som vi ser så er dette en typisk integrator.

Diskusjon:

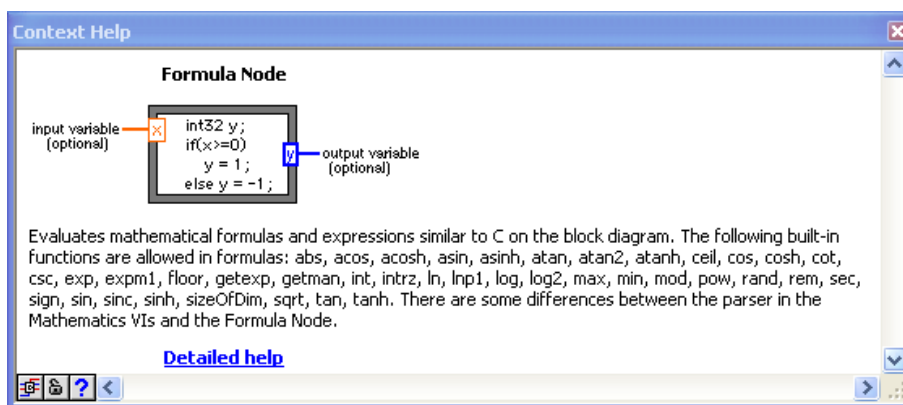
Merk! k (t_k) er den diskrete tidsindeksen. T_s er tiden mellom 2 samplinger t_k and t_{k+1} . F.eks hvis $k = 10$, da blir $t = T_s \times k = 0.10s \times 10 = 1s$

Merk! Du burde alltid vise enheter på x-aksen og på y-aksen i alle plot. Vi kan bruke de innebygde funksjone *title()*, *xlabel()* and *ylabel()* for å få til dette.

[Slutt på eksempel]

20.3 Diskretisering i LabVIEW

I LabVIEW har vi også ulike muligheter for å simulere både kontinuerlige og diskrete systemer. I dette eksempelet vil vi bruke "Formula Node". "Formula Node" i LabVIEW evaluerer matematiske formler og uttrykk på lik linje med vanlig C kode. På denne måten kan vi bruke eksisterende C kode (eller lage ny) direkte inni LabVIEW. Dette kan være spesielt nyttig hvis du har komplekse matematiske uttrykk.



Eksempel:

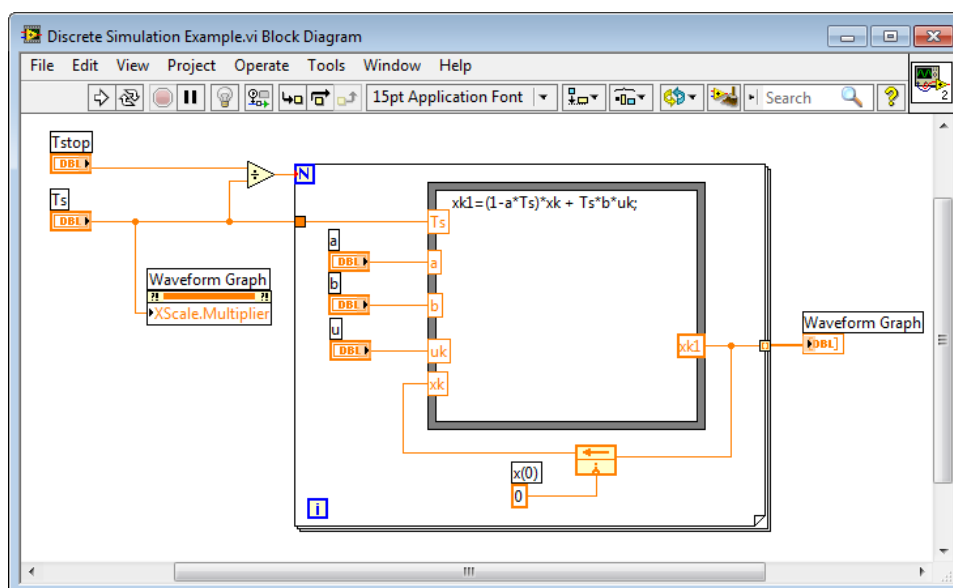
Vi bruker følgende diskrete system:

$$x(k + 1) = (1 - T_s a)x(k) + T_s b u(k)$$

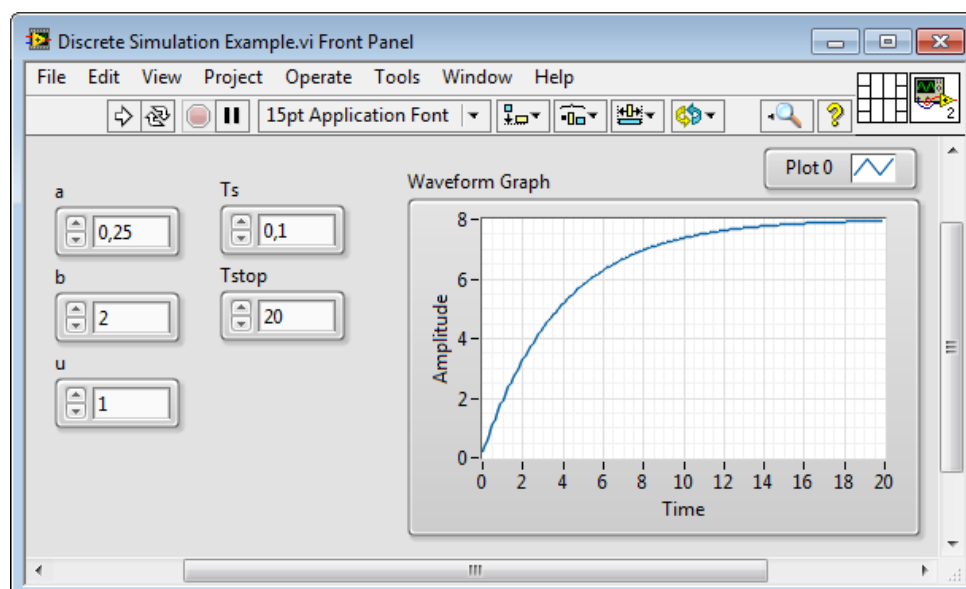
Hvor vi setter $a = 0.25$ og $b = 2$

Vi bruker da "Formula Node" og en For Løkke i LabVIEW for å simulere det diskrete systemet.

LabVIEW koden blir som følger:



Simuleringsresultater vises da på frontpanelet:



Som du ser får vi samme resultat som i tidligere eksempler.

[Slutt på eksempel]



Hans-Petter Halvorsen, M.Sc.

E-mail: hans.p.halvorsen@hit.no

Blog: <http://home.hit.no/~hansha/>



University College of Southeast Norway

www.usn.no
