

## Kapittel 3

# Introduksjon til LabVIEW-programmering

### 3.1 Innledning

Vi skal i dette kapitlet programmere en VI – som skal kalles `sigtrans.vi`<sup>1</sup> – som realiserer følgende funksjonssammenheng:

$$y = Ku + b \tag{3.1}$$

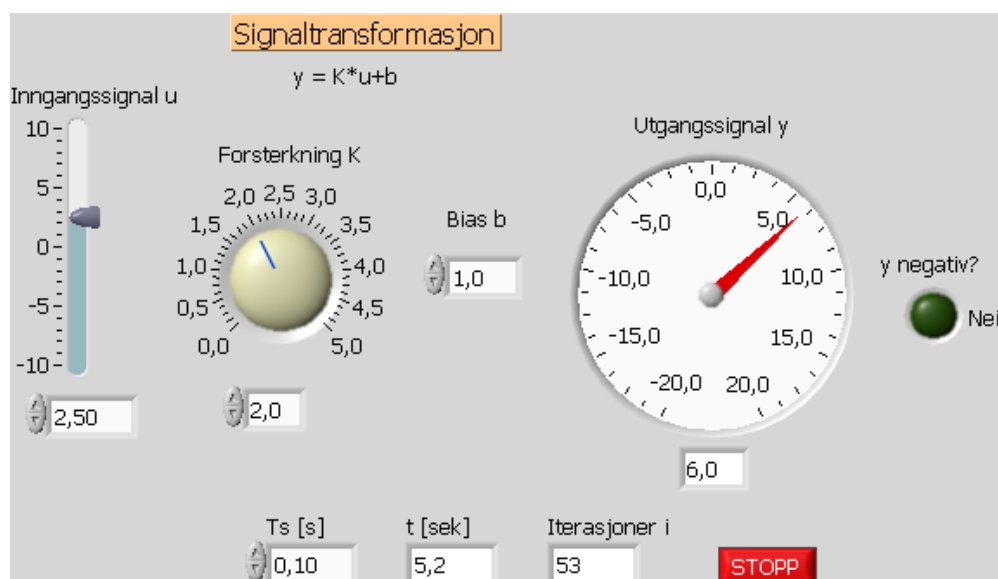
Signalet eller variabelen  $y$  beregnes altså som variabelen  $u$  multiplisert med forsterkningsfaktoren  $K$ , pluss  $b$ . VI'en utfører altså en signaltransformasjon av  $u$ , som i praksis kan være et innlest målesignal. I tillegg skal VI'en indikere med en lampe på frontpanelet dersom  $y$  får en negativ verdi.

Figur 3.1 viser VI'ens frontpanel, og figur 3.2 viser VI'ens diagram etter at VI'en er programmert. Det er i diagrammet funksjonssammenhengen (3.1), samt tenningen av lampen, er realisert. Som forklart i kapittel 2.3, er de små firkantene med påskrift “DBL” terminaler av datatype flyttall med dobbel presisjon, mens “TF” er en boolsk terminal.

(Du kan gjerne nå åpne en ny VI via LabVIEWs oppstartsvindu, se figur 2.1.)

---

<sup>1</sup>Denne VI'en, som alle andre filer beskrevet i boka, er tilgjengelig via bokas hjemmeside på <http://techteach.no> (brukernavn og passord for tilgang til filene er oppgitt i bokas forord).



Figur 3.1: Frontpanelet for sigtrans.vi

## 3.2 Brukergrensesnittet under programmeringen

Programmeringen av en VI går ut på å plassere elementer hentet fra de aktuelle *palettene* på frontpanelet og diagrammet. Det fins 3 paletter:

- **Tools Palette** (eller verktøypaletten)
- **Controls Palette** (eller frontpanelpaletten)
- **Functions Palette** (eller funksjonspaletten)

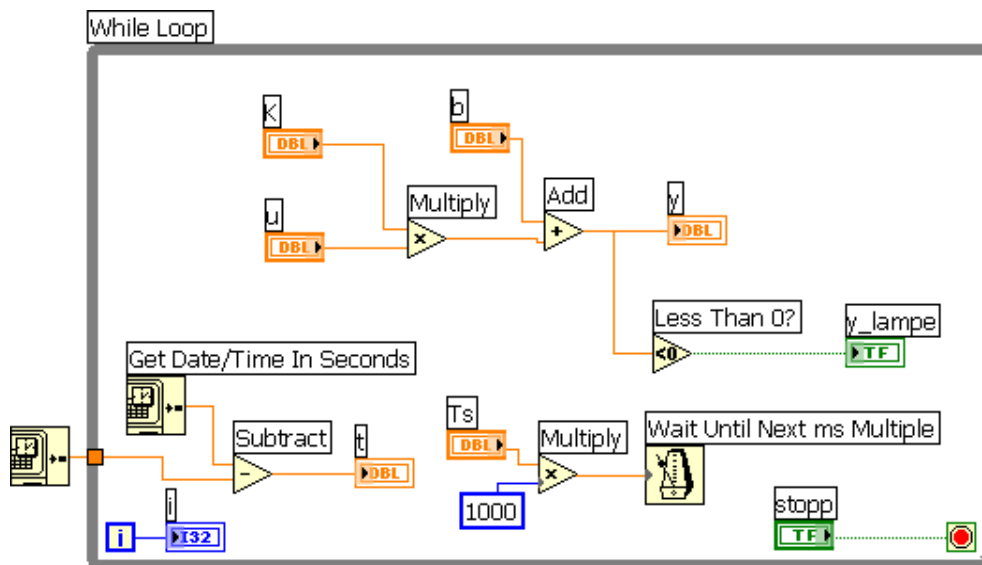
Disse palettene er nærmere omtalt nedenfor:

### 3.2.1 Tools Palette

Tools Palette inneholder verktøyknapper som når de klikkes på, *setter musmarkøren (cursor) i ulike modi*. Tools Palette åpnes automatisk når du programmerer i LabVIEW, men hvis den av en eller annen grunn er blitt lukket (du kan selv lukke den), kan den åpnes med menyvalget *Windows / Show Tools Palette*. Figur 3.3 viser Tools Palette.

Tools Palette har følgende knapper:

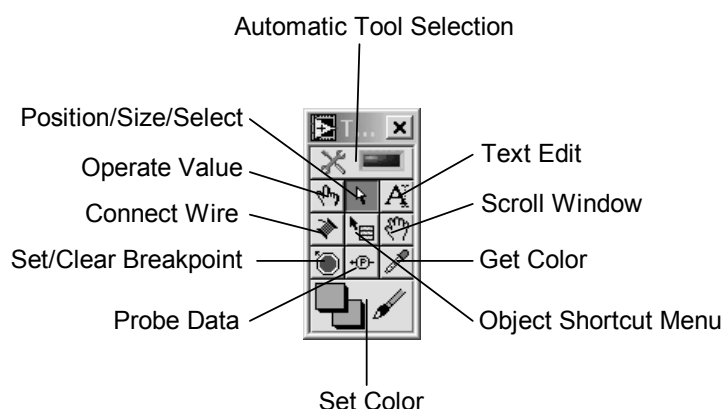
- **Operate Value**, som benyttes når en skal *endre verdien av*



Figur 3.2: Diagrammet for sigtrans.vi

elementer på frontpanelet eller i diagrammet, f.eks. for å skrive inn ny verdi i et tallfelt (digital display)

- **Position/Size/Select**, som benyttes når en skal
  - *posisjonere* eller
  - *endre størrelsen av* eller
  - *velge*
- elementer på frontpanelet eller i diagrammet.
- **Edit Text**, som velges før en skal *skrive tekst* i forskjellige sammenhenger, f.eks. skrive tekst på frontpanelet eller i diagrammet, endre skalaers maksimal- eller minimalverdi (ved å skrive på skalaen direkte), skrive inn ny elementtittel (eng.: label), m.m.
- **Scroll Window**, som gir mulighet til å *skyve* (ikke flytte) visningen av hele innholdet i frontpanelvinduet eller diagramvinduet.
- **Object Shortcut Menu**, som vil åpne en *hurtigmeny* når en klikker på et element. Dette er ekvivalent med høyreklikk på elementet.
- **Probe Data**, som gir mulighet til å sette inn en *probe* i tilknytning til det valgte (klikkede) elementet i diagrammet. En probe er i form av et lite vindu der den nåværende tallverdien av det valgte elementet vises. Du kan legge inn så mange prober som du ønsker i diagrammet.

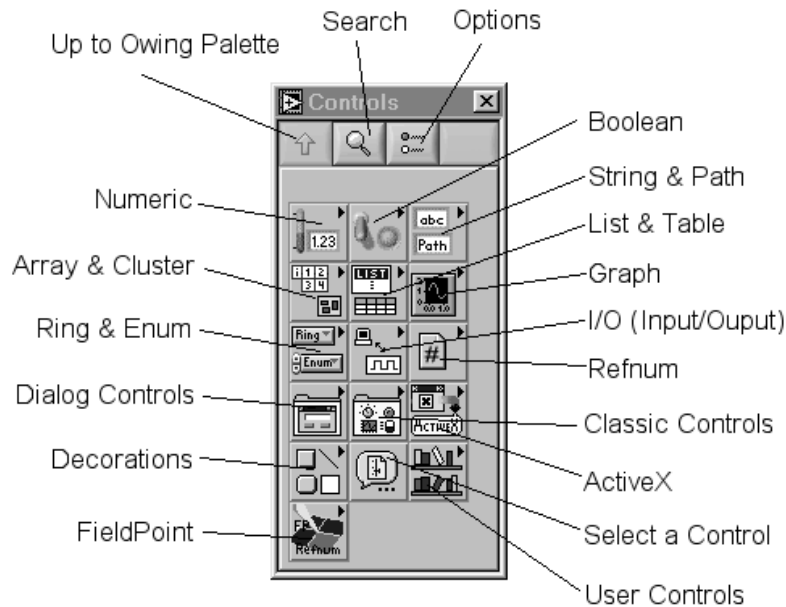


Figur 3.3: Tools Palette eller verktøypaletten i LabVIEW.

- **Get Color**, som *fanger fargen* på det stedet der du har klikket. Denne fargen vil så legge seg inn i feltet for sist valgte farger (nederst) i fargekartet som åpnes når du velger verktøyknappen Set Color, se nedenfor.
  - **Set Color**, som benyttes når du skal *fargelegge* elementer på frontpanelet eller i diagrammet. Framgangsmåten er først å velge Set Color-knappen, og så høyreklikke på det aktuelle elementet frontpanelet eller i diagrammet og deretter velge farge fra fargekartet som åpnes. Du kan fargelegge nær sagt et hvilket som helst element og en hvilken som helst del av et element!
- Merk
- **Set/Clear Breakpoint**, som benyttes for å legge inn og fjerne *stoppunkter* i diagrammet. Dette kan være aktuelt i forbindelse med feilsøking.
  - **Connect Wire** eller **trådsnella**, som benyttes for å *trekke forbindelseslinjer* eller *ledninger* mellom elementer i *diagrammet*.
  - **Automatic Tool Selection**, som gjør at markørens funksjon velges automatisk blant de ovenfor nevnte markørfunksjonene, avhengig av kontekst.

### 3.2.2 Controls Palette

Controls Palette benyttes for å legge inn elementer på *frontpanelet*. Figur 3.4 viser Controls Palette og dens subpaletter. Denne paletten åpnes



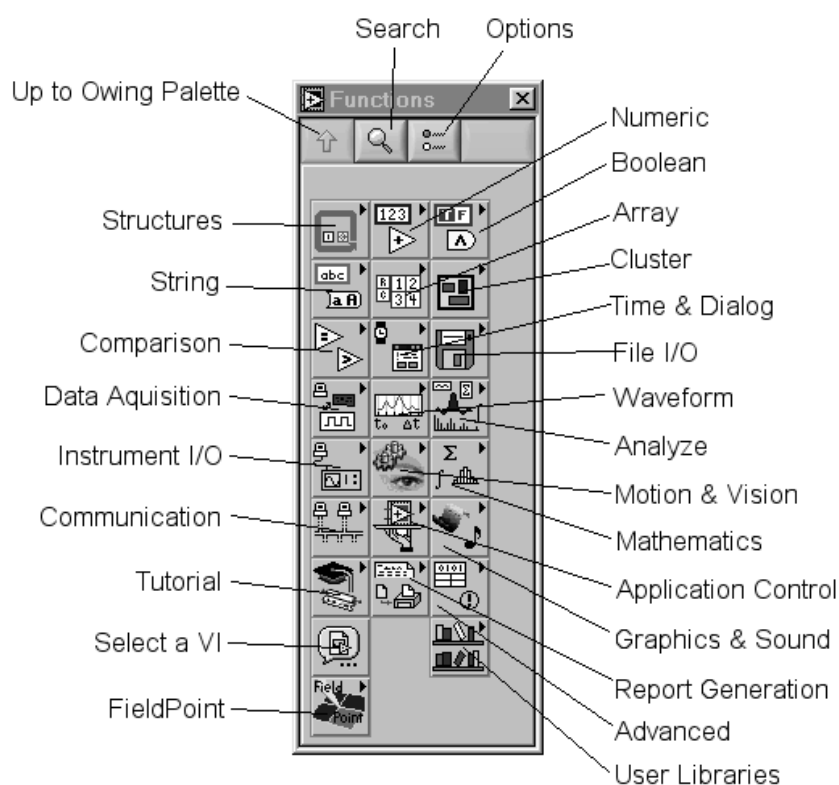
Figur 3.4: Controls Palette hvorfra elementer som skal plasseres på frontpanelet, hentes.

automatisk når du har framme frontpanelet for VI'en du programmerer, hvis den av en eller annen grunn er blitt lukket (du kan selv lukke den), kan den åpnes med menyvalget *Windows / Show Controls Palette*.

### 3.2.3 Functions Palette

*Functions Palette* eller funksjonspaletten benyttes for å legge inn elementer i *diagrammet*. Denne paletten åpnes automatisk når du har framme diagrammet for VI'en du programmerer. Hvis den av en eller annen grunn er blitt lukket (du kan selv lukke den), kan den åpnes med menyvalget *Windows / Show Functions Palette*. Figur 3.5 viser *Functions Palette* og dens subpaletter. Når tilleggspakker (toolkits) til LabVIEW installeres, utvides *Functions Palette* med tilhørende subpaletter.

I tillegg til *Tools Palette* og *Functions Palette* vil *menyer fra menylinja* og *knapper på verktøylinja* i frontpanelvinduet og diagramvinduet være aktuelle under programmeringen. Disse menyene og knappene blir ikke gjennomgått her. Vi gjør oss heller kjent med dem i forbindelse med de konkrete programmeringseksemplene utover i boka.



Figur 3.5: Functions Palette eller funksjonspaletten hvorfra elementer som skal plasser i VI'ens diagram, hentes.

### 3.3 Trinn i programutviklingen

Programutviklingen består gjerne av følgende trinn, i denne rekkefølgen:

1. **Konstruksjon av frontpanelet med de ønskede elementene**, dvs. med styrelementer (controls) og indikatorer (indicators) hentet fra Controls Palette. Skriv eventuell fritekst på frontpanelet.
2. **Konfigurering av frontpanelementene** (oftest via høyreklikk på elementet).
3. **Programmering av diagrammet slik at VI'en får ønsket funksjonalitet**. Dette gjøres ved å hente funksjonselementer fra Functions Palette inn til diagrammet. Terminalene (som jo svarer til elementer på frontpanelet) og funksjonselementene koples sammen ved å trekke forbindelseslinjer mellom dem etter at du har valgt

Connect Wire-knappen på Tools Palette.

4. **Lagring av VI'en.** Dette gjøres med menyvalget File /Save. Du kan selvsagt lagre VI'en når som helst, dvs. også underveis i programutviklingen. (Det er lurt stadig å lagre VI'en mens du programmerer, slik at du ikke mister arbeid hvis PC'en henger seg.) Husk at det fins en rekke opsjoner for fillagring, jf. underkap. 3.9.

### 3.4 Nyttige håndgrep

Underveis i programmeringen kan følgende håndgrep være nyttige (beskrivelsen nedenfor blir klarere når du har kommet i gang med selve programmeringen, og det kan da være lurt å kikke tilbake på denne lista):

- **Svitsje mellom frontpanelvinduet og diagramvinduet:**  
Menyvalg: Window /Show Diagram hhv. Window / Show Panel.
- **Angre og gjenta:** Menyvalget Edit / Undo hhv. Redo.
- **Finne sammenhørende elementer på frontpanel og i diagram:** Høyreklikk / Find Terminal, som utføres fra frontpanelet, og Høyreklikk / Find Control eller Indicator, som utføres fra diagrammet.
- **Merke ett eller flere elementer på frontpanelet eller i diagrammet:** Først velger du Tools Palette / Position. For å merke *ett element* klikker du én gang på elementet. For å merke *flere elementer*, kan du enten klikke på de aktuelle elementene mens du holder skift-tasten på tastaturet nede, eller du kan bruke musa til å trekke en tenkt firkant rundt elementene som skal merkes.
- **Slette ett eller flere elementer på frontpanelet eller i diagrammet:** Først må de aktuelle elementene merkes, se ovenfor. Så kan de slettes med slettetasten på tastaturet.

I utgangspunktet kan du ikke slette et frontpanelement ved å slette den tilsvarende terminalen i diagrammet, men du kan angi at dette allikevel skal være mulig via følgende menyvalg: Tools / Options / Block Diagram / Delete front panel elements from diagram.

- **Flytte elementer på frontpanelet eller i diagrammet:** Først må de aktuelle elementene merkes, se ovenfor, og så kan du bruke musa til å flytte elementene.

Du kan, som alternativ til å bruke musa, finposisjonere elementer med *piltastene på tastaturet*.

- **Hvordan distribuere eller posisjonere elementer på frontpanelet og i diagrammet:** Kanskje du ønsker å posisjonere elementer slik at elementene blir venstrejustert mot en tenkt felles venstre kantlinje, eller slik at avstanden mellom elementene blir lik? Dette kan gjøres slik: Først markeres elementene. Så velges knappen Distribute Objects eller Align Objects i verktøylinja i frontpanelvinduet eller i diagramvinduet.
- **Hvordan erstatte et elementet med et annet:** Høyreklikk på elementet / Replace / Velg nytt element fra Controls Palette eller fra Functions Palette.
- **Vise navnet (label) på en funksjon i diagrammet:** Høyreklikk på funksjonens blokk, og velg Visible / Label.
- **Skjule frontpanelement:** Dette kan gjøres fra frontpanelet eller fra diagrammet.
  - Fra frontpanelet: Høyreklikk på elementet / Advanced / Hide.
  - Fra diagrammet: Høyreklikk på terminalen / Hide.
- **Gjøre et frontpanelement inaktivt:** Høyreklikk på elementet / Advanced / Enabled State / Disable (evt. Disabled & Greyed).
- **Endre fra control til indicator og motsatt:** Høyreklikk på elementet / Change to Control eller Indicator.
- **Automatisk opprette et frontpanelement fra diagrammet:** Høyreklikk på den aktuelle terminalen eller funksjonen i diagrammet / Create / Control eller Indicator. Dette er en hurtigmåte til f.eks. å opprette (lage) en indikator (på frontpanelet) som skal vise verdien av en funksjonsutgang (i diagrammet).
- **Slette et frontpanelement fra diagrammet:** I utgangspunktet kan du ikke slette frontpanelementer fra diagrammet. Dette blir allikevel mulig etter følgende menyvalg i VI'ens menylinje: Tools / Options / Block Diagram / Delete Front Panel Elements from Block Diagram.
- **Fjerne alle løse signalledninger i diagrammet:** Menyvalg Edit / Remove Broken Wires eller tastekombinasjon Ctrl B.

### 3.5 Eksempel: Programmering av sigtrans.vi

Vi skal nå i detalj utvikle sigtrans.vi vist i figurene 3.1 og 3.2. Beskrivelsen nedenfor tar utgangspunkt i trinnene beskrevet i kap. 3.3.

Hvis du ikke allerede har gjort det, kan du nå åpne en ny VI f.eks. gjennom New-knappen i LabVIEWs oppstartsvindu, se figur 2.1. (Det er lurt å lagre VI'en jevnlig.)

### 3.5.1 Konstruksjon av frontpanelet med de ønskede elementene

1. Legg inn skyveren for  $u$  på frontpanelet: Controls Palette / Numeric / Vertical Pointer Slide.
2. Legg inn skruknappen (pot-meteret) for  $K$ : Controls Palette / Numeric / Dial.
3. Legg inn tallfeltet for  $b$ : Controls Palette / Numeric / Digital Control.
4. Legg inn viserindikatoren for  $y$ : Controls Palette / Numeric / Meter.
5. Legg inn tallfeltet for angivelse av tidsskrittet  $T_s$  som VI'en skal kjøres med: Controls Palette / Numeric / Digital Control.
6. Legg inn tallfeltet for visning av tiden  $t$  fra VI'en er startet: Controls Palette / Numeric / Digital Indicator.
7. Legg inn stoppknappen for stopp av VI'en: Controls Palette / Boolean / Stop Button.
8. Legg inn en lampeindikator for visning av fortegnet av  $y$ : Controls Palette / Boolean / Round LED<sup>2</sup>.
9. Skriv inn friteksten "Signaltransformasjon" og " $y = K*u+b$ " på frontpanelet. Dette gjøres med Tools Palette / Edit Text.
10. Legg inn tallfelt for antall iterasjoner (som programmets while-løkke har gått).

### 3.5.2 Konfigurerer av frontpanelementene

Vi tar for oss de enkelte frontpanelementene vist i figur 3.1 etter tur:

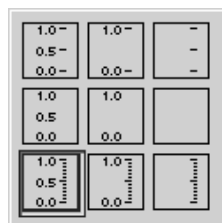
#### Vertikal skyver for $u$ :

1. Angi Label (elementets navn): Høyreklikk / Visible Items / Label. Før du kan begynne å skrive, må du ha valgt Tools Palette / Edit Text. La Label bli  $u$ . Elementets Label er elementets (eller variabelens) *navn*, som vil bli vist i VI'ens diagram.

---

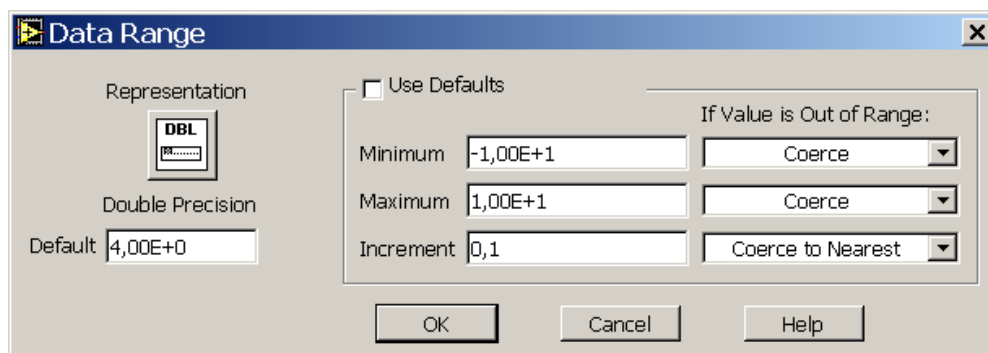
<sup>2</sup>LED = Light Emitting Diode

2. Ta fram elementets Caption, som utgjør en elementbeskrivelse: Høyreklikk / Visible Items / Caption, hvilket plasserer elementets Label til venstre og setter Caption på Labels opprinnelige plass. Skriv **Inngangssignal** u i Caption-feltet.
3. Skjul Label: Høyreklikk / Visible Items / Label.
4. Få fram Digital Display (tallfelt): Høyreklikk / Visible items / Digital display.
5. Flytt Digital Display til posisjon under skruknappen ved først å velge Tools Palette / Position og deretter flytte elementet med musa.
6. Øk størrelsen av skyverens (etter eget valg) ved først å velge Tools Palette / Size og deretter trekke i hendlene som vises over skyverens når musmarkøren føres over knappen.
7. Angi antall desimaler som skal vises på skalaen og i Digital Display: Høyreklikk / Formats & Precicion / Digits of Precision. Angi 1 desimal.
8. Angi at det skal være fingraderte skalastreker: Høyreklikk / Scale / Style, se figur 3.6.



Figur 3.6: Angivelse av skalastreker: Høyreklikk / Scale / Style.

9. Angi inkrementet 0,1 på skyverens: Høyreklikk på elementet / Data Range / Increment, se figur 3.7.
10. Angi skala fra  $-10$  til  $+10$  ved å skrive direkte på skalaens minimal- hhv. maksimalverdi etter at Tools Palette / Edit Text er valgt.
11. Angi at  $-10$  er minste verdi som elementet kan stilles inn på: Høyreklikk på elementet / Data Range / Minimum. (Generelt kan en i tallfeltet knyttet til elementet stille inn verdier utenfor skalaområdet for elementet.)
12. Angi at  $10$  er største verdi som elementet kan stilles inn på: Høyreklikk på elementet / Data Range / Maximum.

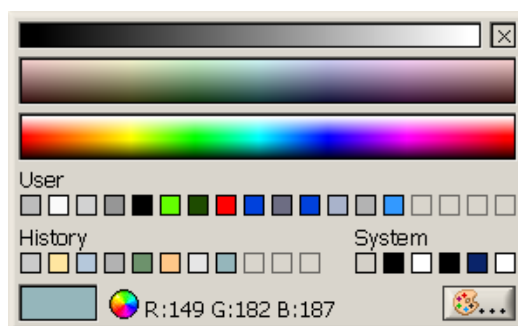


Figur 3.7: Angivelse av inkrementet 0,1 p å skyverer: Høyreklikk på elementet / Data Range / Increment

13. Angi forhåndsverdi (Default value) lik 4,0: Høyreklikk / Data operations / Make current value default.

Hvis du vil definere forhåndsverdier for *alle* elementene på frontpanelet på én gang, velges menyen (i VI'ens menylinje) Operate / Make current values default etter at ønskede verdier er angitt for de enkelte elementene.

14. Velg farge for selve skruknappen: Tools Palette / Set Color / høyreklikk på det aktuelle elementet / velg farge fra fargekartet som åpnes, se figur 3.8.



Figur 3.8: Velg farge for en del av et element: Tools Palette / Set Color / høyreklikk på den aktuelle elementdelen / velg farge fra fargekartet som åpnes.

Du kan på denne måten velge farge for hvilket som helst element, og hvilken som *del* av et element. Merk det lille feltet oppe til høyre på fargekartet. Dersom feltet inneholder en T (eng: Transparent), kan du velge transparent som farge, dvs. at fargen blir som for elementet

i bakgrunnen.

Mange av menyvalgene i punktene som følger, er de samme som i punktene for u-skyveren (ovenfor), og slike like menyvalg gjentas derfor ikke.

### **Skruknappen for $K$ :**

1. Angi Label  $K$ .
2. Angi Caption Forsterkning  $K$ .
3. Skjul Label.
4. Vis Digital Display (tallfelt).
5. Flytt Digital Display til posisjon under skruknappen.
6. Øk størrelsen av skyveren etter eget ønske.
7. Angi at antall desimaler som skal vises på skalaen og i Digital Display skal være 1.
8. Angi at det skal være finglydte skalastreker.
9. Angi inkrement 0,1.
10. Angi skala fra 0 til 5.
11. Angi at 0 er minste verdi. (Det skal ikke være begrensning på øvre verdi.)
12. Angi forhåndsverdi (Default value) lik 2,0.
13. Fargelegg deler av elementet etter eget ønske.

### **Tallfeltet for $b$ :**

1. Angi Label  $b$ .
2. Angi Caption Bias  $b$ .
3. Skjul Label.
4. Angi at antall desimaler som skal vises er 1.
5. Angi inkrement 0,1.
6. Angi at  $-2$  er minste verdi og 2 er maksimal verdi.
7. Angi forhåndsverdi (Default value) 0.
8. Fargelegg deler av elementet etter eget ønske.

**Viserindikatoren for y:**

1. Angi Label y.
2. Angi Caption **Utgangssignal** y.
3. Skjul Label.
4. Vis Digital Display (tallfelt).
5. Flytt Digital Display til posisjon under skruknappen.
6. Juster størrelsen av viseren etter eget ønske.
7. Angi at antall desimaler som skal vises på skalaen og i Digital Display, er 1.
8. Angi at det skal være fingraderte skalastreker.
9. Angi viserskala  $-20$ ,  $+20$ .
10. Fargelegg deler av elementet etter eget ønske.

**Lampe- (LED-)indikatoren:**

1. Angi Label **y\_lampe**.
2. Angi Caption **y negativ?**.
3. Skjul Label.
4. Juster størrelsen av lampen etter eget ønske.
5. Vis Boolean Text: **Høyreklikk / Visible Items / Boolean Text**. Boolean Text har verdi ON hhv. OFF.
6. Flytt Boolean Text f.eks. til høyre for lampeindikatoren (slik at teksten ses bedre) ved først å velge **Tools Palette / Position**.
7. Endre Boolean Text for lampeindikatoren fra ON og OFF til hhv. **Ja** og **Nei**. Dette gjøres ved først å velge **Tools Palette / Edit Text** før du skriver inn ny tekst oppå den gamle. For å svitsje mellom de to boolske tilstandene for lampeindikatoren, kan du velge **Tools Palette / Operate Value** og deretter klikke på selve lampen.

**Tallfeltet for  $T_s$ :**

1. Angi Label  $T_s$ .
2. Angi Caption  $T_s$  [sek].
3. Skjul Label.
4. Antall desimaler som skal vises, er 2.
5. Angi inkrement 0,01.
6. Angi at 0 er minste verdi.
7. Angi forhåndsverdi (Default value) 0,1.
8. Juster størrelsen etter eget ønske.
9. Fargelegg deler av elementet etter eget ønske.

**Tallfeltet for  $t$ :**

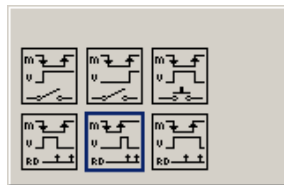
1. Angi Label  $t$ .
2. Angi Caption  $t$  [sek].
3. Skjul Label.
4. Antall desimaler som skal vises, er 1.
5. Juster størrelsen etter eget ønske.
6. Fargelegg deler av elementet etter eget ønske.

**Tallfeltet for  $i$ :**

1. Angi Label  $i$ .
2. Angi Caption Iterasjon nr.
3. Skjul Label.
4. Angi at datatypen er heltall (ikke desimaltall): Høyreklikk på elementet / Representation / Long.
5. Juster størrelsen etter eget ønske.
6. Fargelegg deler av elementet etter eget ønske.

**Stopp-knappen:**

1. Angi Label **stopp**.
2. Skjul Label.
3. Skriv teksten **STOPP** på knappen (dette er egentlig knappens Boolean Text), i passende farge, f.eks. hvit. Tekstfarge velges fra Text Settings-knappen i VI'ens verktøylinje.
4. Angi passende knappefarge, f.eks. rød.
5. Angi knappens "mekaniske" virkemåte: **Høyreklikk på knappen / Mechanical Action / Latch when Pressed**, se figur 3.9.



Figur 3.9: Knappens "mekaniske" virkemåte angis med Høyreklikk på knappen / Mechanical Action / Latch when Pressed.

### 3.5.3 Programmering av diagrammet ihht. ønsket funksjonalitet

#### Innledning om while-løkker

VI'en sigtrans.vi, som vi utvikler, skal kjøre kontinuerlig, og brukeren skal kunne stoppe VI'en ved å trykke på stoppknappen på frontpanelet. (Generelt i applikasjoner som omfatter datainnsamling, styring og simulering, skal programmet (VI'en) gå kontinuerlig, f.eks. med innsamling av måledata hvert millisekund eller utskrivning av et nytt styresignal hvert halve sekund.) Dette realiseres med en såkalt *while-løkke* (eng.: while loop). Begrepet while-løkke stammer fra tradisjonelle programmeringsspråk, der while-løkkene kan se ut som følger, skrevet i pseudokode:

```
while (not stoppknapp)
{
y = K*u + b;
wait(10);
}
```

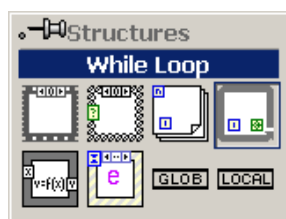
der det antas at **stoppknapp** er en logisk (boolsk) variabel hvis verdi blir **true** (sann) når en viss stoppknapp er trykket og **false** ellers. **not** betyr logisk negasjon eller invertering, dvs. at **false** er lik **not true**. **wait()** er en tenkt funksjon som holder (låser) programutførelsen i et spesifisert antall millisekunder (her 10), dvs. at den realiserer en slags tidsforsinkelse. While-løkke vil utføre programkoden som står inne i while-løkke, dvs. mellom begynnelsemarkøren { og stoppmarkøren }, om igjen og om igjen så lenge while-betingelsen, som her er **not stoppknapp**, har logisk verdi **true**, eller med andre ord: så lenge stoppknappen ikke er trykket. (Eksempelvis vil programuttrykkene for innsamling av data i datainnsamlingsapplikasjoner stå inne i while-løkke, og i styringsapplikasjoner vil programuttrykkene for beregning og utskrivning av styresignalet til prosessen stå inne i while-løkke.)

I LabVIEW gis while-løkker en grafisk representasjon i form av en ramme (frame), og programkoden som skal utføres i løkke, skal angis inne i denne ramma.

### Programmering av diagrammet i sigtrans.vi

Punktene nedenfor skal resultere i diagrammet vist i figur 3.2. Diagrammet i figuren inneholder Label-felter for alle funksjonene. I utgangspunktet er ikke Label for funksjoner vist, men de kan vises slik: Høyreklikk på funksjonen / Visible Items / Label. I underkap. 3.4 er mange nyttige programmeringshåndgrep beskrevet. Du kan gjerne prøve noen av dem mens du programmerer diagrammet.

1. Legg inn en while-løkke fra Functions Palette / Structures / While-loop, se figur 3.10, og plasser alle terminalene inne i

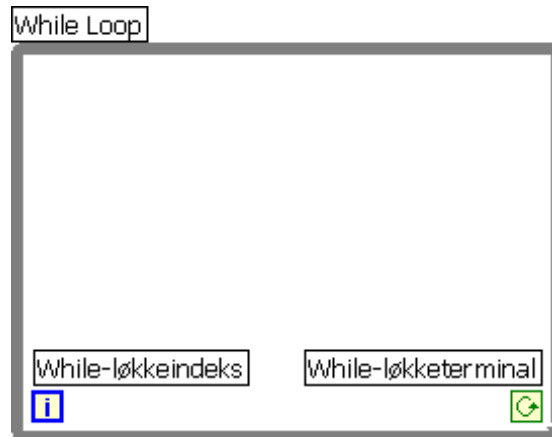


Figur 3.10: En while-løkke legges inn i diagrammet med Functions Palette / Structures / While-loop.

while-løkke ihht. figur 3.2.

Kommentarer:

- Nederst til høyre inne i ramma er en grønnfarget firkant – den såkalte *løkketerminalen*, som er en logisk variabel. Se også figur 3.11. Løkketerminalen virker normal slik at while-løkka vil



Figur 3.11: While-løkkeindeksen og while-løkketerminalen

fortsette å gå så lenge løkketerminalen har logisk verdi *true*, dvs. at den stopper hvis løkketerminalen har logisk verdi *false*. Det er imidlertid mulig å konfigurere løkketerminalen slik at while-løkka vil stoppe hvis løkketerminalen får logisk verdi *true*. Denne omkonfigureringen foretas via høyreklikk på løkketerminalen. Denne omkonfigureringen *er foretatt* i diagrammet i figur 3.2. En grunn til å foreta en slik omkonfigurering er at stoppknappen som en gjerne legger inn på frontpanelet – og hvis terminal tilkoples løkketerminalen, gir logisk verdi *true* når den er trykket.

- Nederst til venstre inne i while-løkka er en firkant med en blå “i” inni. Dette er *løkkeindeksen*, som er en variabel hvis verdi øker med én (fra null) for hver gang løkka utføres. Se også figur 3.11. Ved å kople en indikator til løkkeindeksen, kan du få vist antall ganger løkka til en hver tid har blitt utført (vi skal gjøre dette for sigtrans.vi).
- Hvis du ønsker å utvide while-løkka, kan du ganske enkelt trekke i ett av hjørnene i while-løkke-ramma. En mulig ulempe ved å utvide ramma på denne måten, er at elementer utenfor ramma kan bli skjult (men de blir ikke fjernet av den grunn) bak ramma. An alternativ måte å utvide ramma på, er som følger: Trykk Ctrl-tasten på tastaturet og hold den inne / bruk musa til å utvide området inne i ramma. Med denne måten vil evt. elementer utenfor ramma automatisk bli flyttet, uten å bli

skjult av ramma når den utvides.

2. Legg inn komparatoren `Less Than 0?` via `Functions Palette / Comparison / Less Than 0?`.
3. Legg inn ventefunksjonen `Wait Until Next ms Multiple` (som ser ut som en metronom) fra `Functions Palette / Time and Dialog`. Denne ventefunksjonen holder eller låser programutførelsen i så mange millisekunder som angis av verdien av ventefunksjonens inngangsterminal, og på denne måten realiseres VI'ens tidsskritt eller samplingsintervall. Brukeren kan angi verdien av tidsskrittet i et eget tallfelt på frontpanelet, se figur 3.1, og siden verdien der forutsettes angitt i sekunder, må sekundverdien multipliseres med 1000 (for å få antall millisekunder).
4. Legg inn klokkefunksjonene `Get Date/Time in Seconds` fra `Functions Palette / Time and Dialog`. Denne funksjonen returnerer antall sekunder fra 1. januar 1904.<sup>3</sup> Den funksjonen som skal plasseres utenfor while-løkkka, kalles bare én gang. Den funksjonen som skal legges inne i while-løkkka, kalles hver gang while-løkkka utføres. Differansen mellom disse to funksjonene vil derfor være antall sekunder som er gått siden while-løkkka gikk for første gang, dvs. antall sekunder siden denne VI'en ble startet.
5. Legg inn en konstant med verdi 1000: `Functions Palette / Numeric / Numeric Constant`, hvoretter du skriver verdi 1000 i konstanten (du kan også angi heltall som datatype via høyreklikk / `Representation / Long`).
6. Legg inn følgende numeriske funksjoner via `Functions Palette / Numeric`, og plasser dem som vist i figur 3.2: 2 stk. multiplikatorer, 1 subtraktor og 1 adderer.
7. Kople terminalene og funksjonene sammen ihht. figur 3.2. Før dette kan gjøres, må du ha valgt `Tools Palette / Connect Wire`.

Litt informasjon om ledningstrekkingen:

- Ledningen (forbindelseslinja) kan trekkes *hvilken som helst vei* ved å venstreklikke og tegne en ny vei etter at du har begynt ledningstrekkingen.
- En ledningsdel kan *merkes* ved å klikke én gang på delen, og jo flere ganger du klikker, jo større del av ledningen merkes.
- Hvis du vil *fjerne* en merket ledning, kan du trykke `Delete-knappen` (på tastaturet).

---

<sup>3</sup>Merkelig, men nyttig allikevel.

- Hvis du vil *fjerne en brutt ledning*, som vil vises med svart farge og stiplet, kan du fjerne den som beskrevet i punktet ovenfor. Hvis du vil fjerne *samtlig*e brutte ledninger i diagrammet, kan du foreta menyvalget **Edit / Remove Broken Wires**.

Dersom VI'en ikke inneholder syntaksfeil, som er "tekniske" programmeringsfeil, skal Run-knappen i VI'ens verktøylinje være hel og hvit. Hvis det er syntaksfeil, vil Run-knappen være gråfarget og brutt. (Vi kommer inn på feilsøking i kap. 3.7.)

Hvis *sigtrans.vi* ikke inneholder syntaksfeil, kan du nå kjøre VI'en for å se om den fungerer korrekt.

### 3.6 Testing av en VI

En VI kan testes ved at den kjøres, selvsagt. *sigtrans.vi* kjører kontinuerlig (pga. while-løkke), så å kjøre den innebærer en kontinuerlig testing (endre parametre etc. og se virkningen umiddelbart).

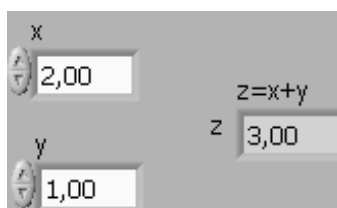
Det hender vi lager VI'er som ikke skal gå kontinuerlig i seg selv, f.eks. en VI'er som utgjør en subVI som skal kalles av en overordnet VI når denne overordnede VI'en bestemmer det. For å teste slike VI'er kontinuerlig, kan vi bruke *Run Continuously*-knappen rett til høyre for Run-knappen, se f.eks. figur 3.1. Bruk av denne knappen er ekvivalent med gjentatte trykk på Run-knappen. Merk: Run Continuously-knappen er kun for testformål, ikke for realisering av normal kontinuerlig kjøring (sistnevnte skal i stedet realiseres med en while-løkke).

La oss som eksempel se på testing av en enkel VI som *ikke* kjører kontinuerlig (den inneholder ingen while-løkke). Figur 3.12 viser frontpanelet og figur 3.13 diagrammet for en enkel VI, *ikke\_kont.vi*, som kun utfører addisjon av to tall. For å teste denne VI'en kontinuerlig, må vi kjøre den vha. Run Continuously-knappen. Hvis vi starter den med Run-knappen, kjører VI'en bare én gang og stopper deretter umiddelbart (her: etter noen mikrosekunder).

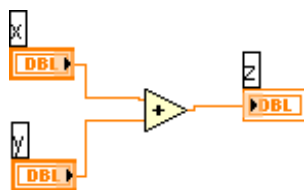
### 3.7 Feilsøking

Feil i programmet er av to typer:

- **Syntaksfeil**, som er feil bruk av språkets syntaks eller regler.
- **Funksjonsfeil**, som er feil i programmets funksjon, selv om programmet rent teknisk fungerer.



Figur 3.12: Frontpanelet for en VI som ikke kjører kontinuerlig. (Fil: ikke\_kont.vi)



Figur 3.13: Diagrammet for en VI som ikke kjører kontinuerlig. (Fil: ikke\_kont.vi)

LabVIEW har gode verktøy for å detektere begge typer feil. Verktøyene demonstreres nedenfor.

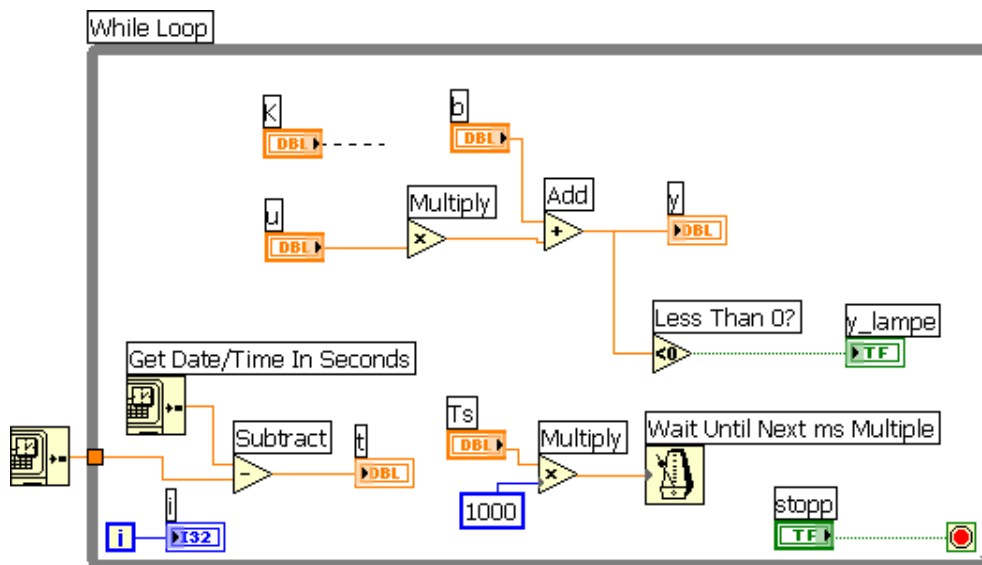
### Syntaksfeil

Vi tar utgangspunkt i *sigtrans.vi*, jf. figurene 3.1 og 3.2. Vi legger inn en feil ved å fjerne ledningen fra K-terminalen til multiplikatoren (ledningen fjernes med klikk og slett på vanlig måte etter at Tools Palette / Select er valgt). Run-knappen blir da vist *brutt og med grå farge*, se figur 3.14. Du kan allikevel starte VI'en via Run-knappen, men programmet stopper umiddelbart, og et vindu kalt *Error List* åpnes, der det angis hva som er feil, se figur 3.15. Dersom du markerer den aktuelle feilbeskrivelsen i dette vinduet og deretter klikker på Find-knappen i det samme vinduet, vil feilen bli markert i diagrammet.

### Funksjonsfeil

Dersom programmet ikke inneholder programmeringsfeil, kan det allikevel være at det ikke fungerer slik du hadde tenkt det skulle, dvs. at det inneholder *funksjonsfeil*. Du har tilgjengelig flere hjelpemidler som kan hjelpe deg å spore opp funksjonsfeil:

- **Sett inn prober på kritiske steder i diagrammet.** En probe er



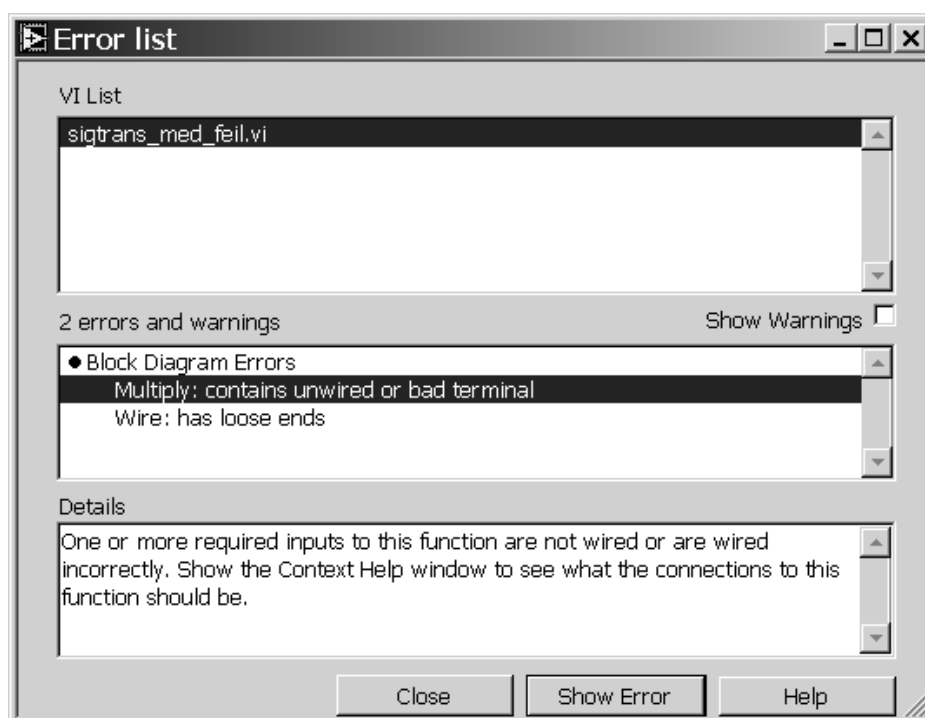
Figur 3.14: Det er programfeil i VI'en ved at den ene av inngangene til multiplikatoren mangler tilkopling. Run-knappen vises grå og brutt. (Fil: sigtrans\_med\_feil.vi)

et lite vindu der den nåværende tallverdien av elementet vises. Du kan knytte en probe til hvilket som helst element i diagrammet. Prober settes inn slik: Tools Palette / Probe Data. Du kan sette inn prober mens VI'en kjører.

- **Indikering av nåværende sted i programmet (under kjøring):** Ved å klikke knappen Highlight Execution i diagrammets verktøylinje (lyspæra), blir det vist hvor i diagrammet programutførelsen foregår akkurat i øyeblikket. Med Highlight Execution aktivert vil programmet gjennomløpes langsommere enn normalt slik at du skal få en sjans til å følge med i diagrammet.
- **Trinnvis programutførelse:** Med dette menes at du vha. knappen Start Single Stepping i diagrammets verktøylinje, går igjennom programmet (diagrammet) trinn for trinn.

### 3.8 Hvordan få hjelp?

Du kan få åpnet et hjelp-vindu med informasjon/dokumentasjon for de fleste elementer i VI-diagrammet med Høyreklikk / Help på elementet. Figur 3.16 viser informasjonen for multiplikatoren i diagrammet for sigtrans.vi, jf. figur 3.2.



Figur 3.15: Et feilvindu (med navn Error List) åpnes dersom du starter en VI som inneholder programmeringsfeil.

Du kan også søke etter informasjon på basis av bl.a. stikkord via menyvalget **Help / Contents and Index** menylinja i frontpanelvinduet eller i diagramvinduet.

### 3.9 Valgmuligheter ved lagring av VI

Du kan når som helst lagre VI'en under programmeringen. Selv om VI'en presenteres i to vinduer – ett for fronpanelet og ett for diagrammet – lagres VI'en som én fil. (På enkelte PC'er kan det være at filtypen "vi" ikke blir lagt til filnavnet automatisk. I så fall må du selv skrive filnavnet på formen \*.vi.)

Menyene for fillagring ligger under hovedmenyen **Edit** (i frontpanelvinduet og i diagramvinduet). Det fins flere valgmuligheter ved fillagringen:

- **Save**, som lagrer fila (VI'en) med sist valgte filnavn. Dette er den vanlige (normale) lagringsfunksjonen.
- **Save As**, som gir mulighet til å lagre med nytt filnavn.

## Multiply

Returns the product of the inputs. The connector pane displays the default data types for this polymorphic function.

---

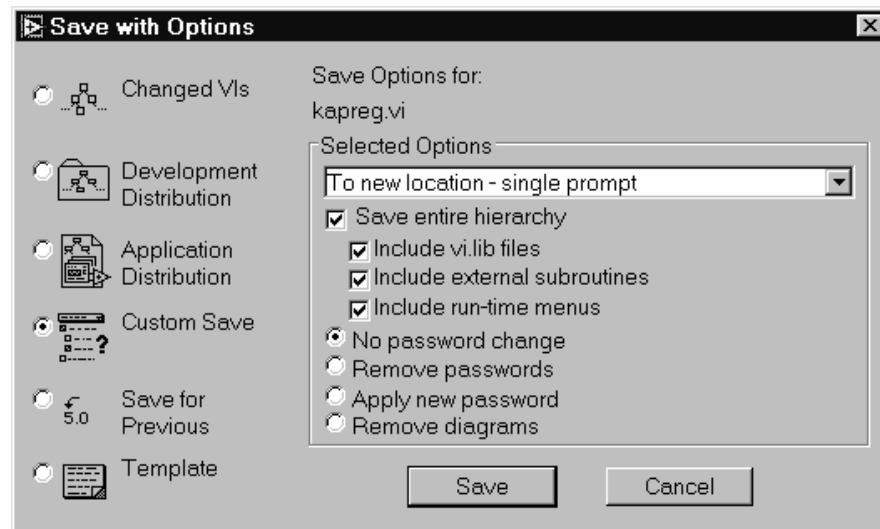
DBL **x** can be a scalar number, array or cluster of numbers, array of clusters of numbers, and so on.

DBL **y** can be a scalar number, array or cluster of numbers, array of clusters of numbers, and so on.

DBL **x\*y** is the product of **x** multiplied by **y**.

Figur 3.16: Informasjon for multiplikatoren i diagrammet for sigtrans.vi. Informasjonen vises etter høyreklikk / Help på multiplikatoren.

- **Save Copy As**, som virker som Save As, men nå slik at den opprinnelige fila fremdeles er åpen.
- **Save With Options**, som gir en rekke muligheter, se figur 3.17. De mest aktuelle mulighetene er:
  - **Development Distribution** kan benyttes når du senere skal programmere videre på VI'en. Med Development Distribution kan du lagre VI'en med *alle eventuelle subVI'er* (en subVI er en VI som er benyttet i diagrammet til en overordnet VI, jf. kap. 10) i en LLB-fil, men allikevel ikke inkludert slike VI'er som følger med LabVIEW og som er såkalte vi.lib-VI'er. LLB står for LabVIEW Library. En LLB-fil er som et filarkiv bestående av flere filer. Du får muligheten til å velge filnavnet for LLB-fila etter at du har trykket Save-knappen, se figur 3.17.
  - **Application Distribution** som kan benyttes når du skal lagre VI'er *uten diagram*, hvilket er nyttig dersom du vil hindre brukerne av VI'en i å se eller modifisere diagrammet. *Obs, obs:* Du hindrer deg selv i å få tilgang til diagrammet også, så det er lurt å velge et nytt filnavn i det du lagrer med Application Distribution.
  - **Custom Save** gir, som Development Distribution, mulighet til å velge helt fritt hvilke VI'er som skal lagres i LLB-fila. Med Custom Save kan du velge å inkludere vi.lib-filer i llb-fila. Dette



Figur 3.17: Vinduet som åpnes ved menyvalget Save With Options.

sikrer at alle VI'er er tilgjengelige dersom du flytter din LLB-fil til en annen datamaskin som kanskje ikke inneholder de samme vi.lib-filene.<sup>4</sup>

---

<sup>4</sup>Jeg pleier å bruke Custom Save ved fillagring.