Høgskolen i Telemark

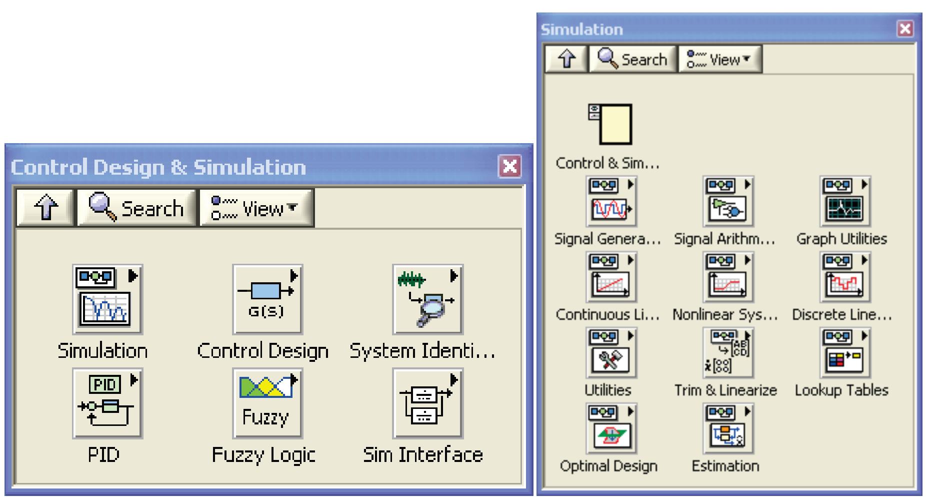
Telemark University College

Department of Electrical Engineering, Information Technology and Cybernetics

→ **Solutions**

Control and Simulation in LabVIEW

HANS-PETTER HALVORSEN, 2011.08.11



Faculty of Technology, Postboks 203, Kjølnes ring 56, N-3901 Porsgrunn, Norway. Tel: +47 35 57 50 00 Fax: +47 35 57 54 01

Table of Contents

1 Differential Equations and Block Diagrams

1.1 1. Order systems

Task 1: 1.order system

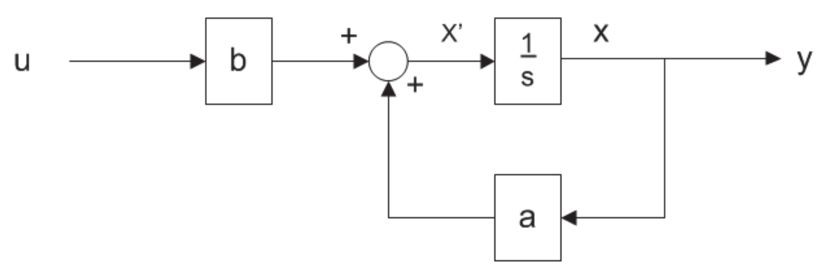
Given the following system:

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

→ Draw a block diagram for the system using pen and paper.

Solutions:

The block diagram becomes:



1.order system with noise

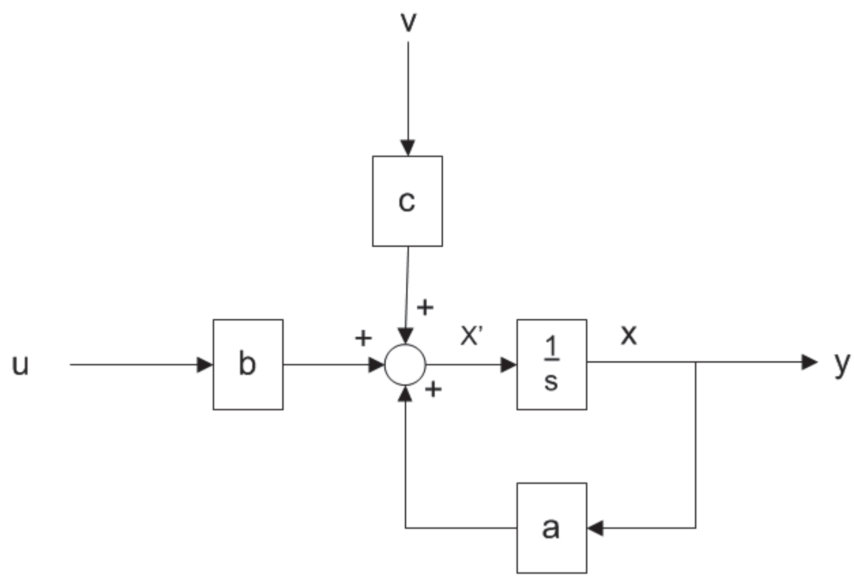
Given the following system:

$$\dot{x} \quad = ax + bu + cv$$

→ Draw a block diagram for the system using pen and paper

Solutions:

The block diagram becomes:



[End of Task]

1.2 2. Order systems

Task 2: 2.order system

Given the following system:

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

→ Draw a block diagram for the system using pen and paper.

x is the position

$\dot{x}$ is the speed

$\ddot{x}$ is the acceleration

F is the Force (control signal, u)

d and k are constants

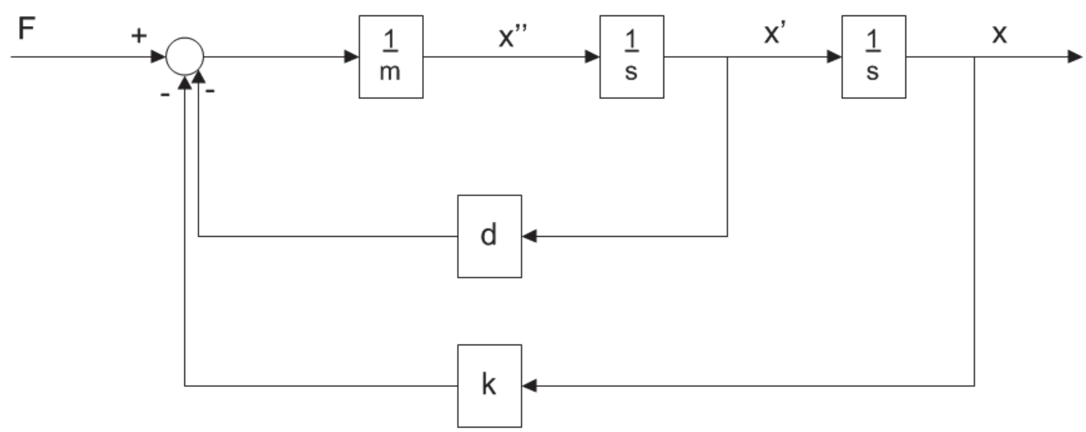
[End of Task]

Solutions:

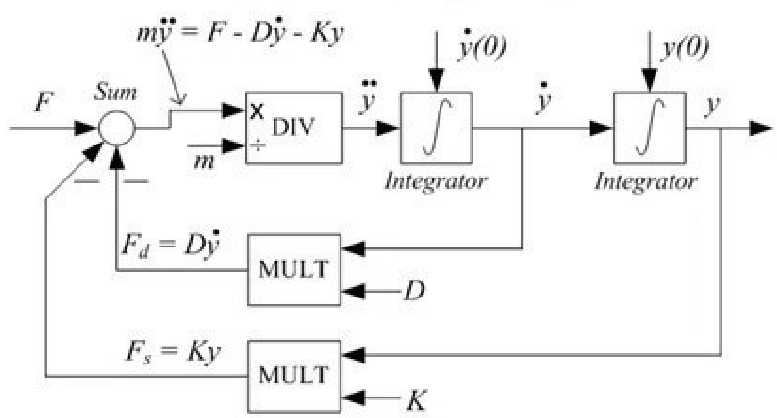
We do the following:

$$\ddot{x} = \frac{1}{m}[F - d\dot{x} - kx]$$

The block diagram becomes:



You may also use this notation:



1.3 State space models

Task 3: State Space model

Given the following system:

$$\dot{x}_1 = -a_1x_1 - a_2x_2 + bu$$

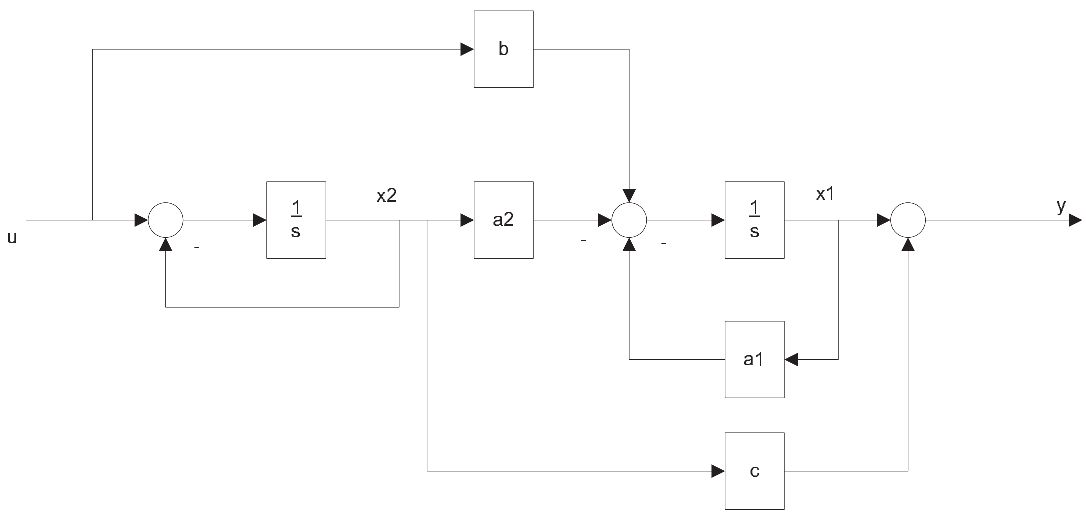
$$\dot{x}_2 = -x_2 + u$$

$$y = x_1 + cx_2$$

→ Draw a block diagram for the system using pen and paper

Solutions:

The block diagram becomes:



Given the following state-space model:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -2 & 0 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 6 \\ 0 \end{bmatrix} u$$

→ Draw the block diagram for this model using pen and paper.

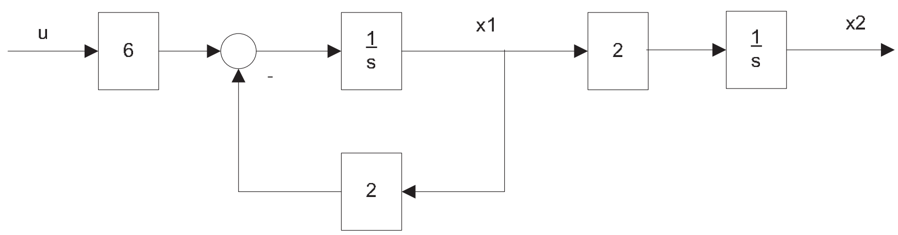
Solutions:

The differential equations becomes:

$$\dot{x}_1 = -2x_1 + 6u$$

$$\dot{x}_2 = 2x_1$$

The block diagram becomes:



Given the following system:

$$\dot{x}_1 = -a_1 x_1 - a_2 x_2 + bu$$

$$\dot{x}_2 = -x_2 + u$$

$$y = x_1 + cx_2$$

→ Find the state-space model on the form:

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

$$y = Cx$$

Draw the block diagram for this simplified model using pen and paper

Solutions:

We get the following state-space model:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b \\ 1 \end{bmatrix} u$$

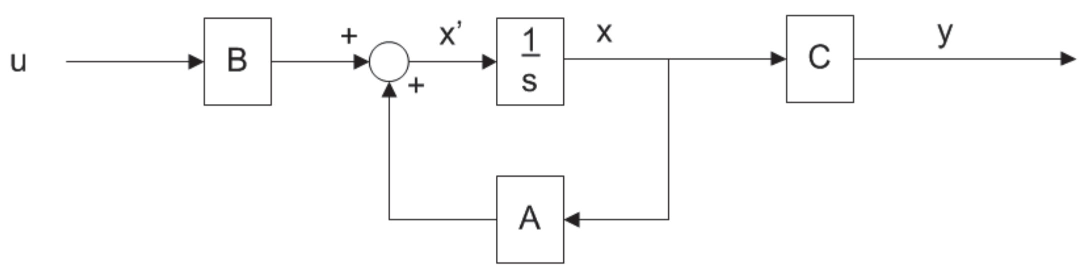
$$y = \begin{bmatrix} 1 & c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Block diagram for

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

$$y = Cx$$

becomes:



[End of Task]

2 Simulation in LabVIEW

Task 4: Bacteria Population

In this task we will use LabVIEW and the LabVIEW Control Design and Simulation Module to simulate a simple model of a bacteria population in a jar.

The **model** is as follows:

$$birth\ rate = bx$$

$$death\ rate = px^2$$

Then the total rate of change of bacteria population is:

$$\boxed{\dot{x} = bx - px^2}$$

Set $b=1$/hour and $p=0.5$ bacteria-hour

We will simulate the number of bacteria in the jar after **1 hour**, assuming that initially there are **100 bacteria** present.

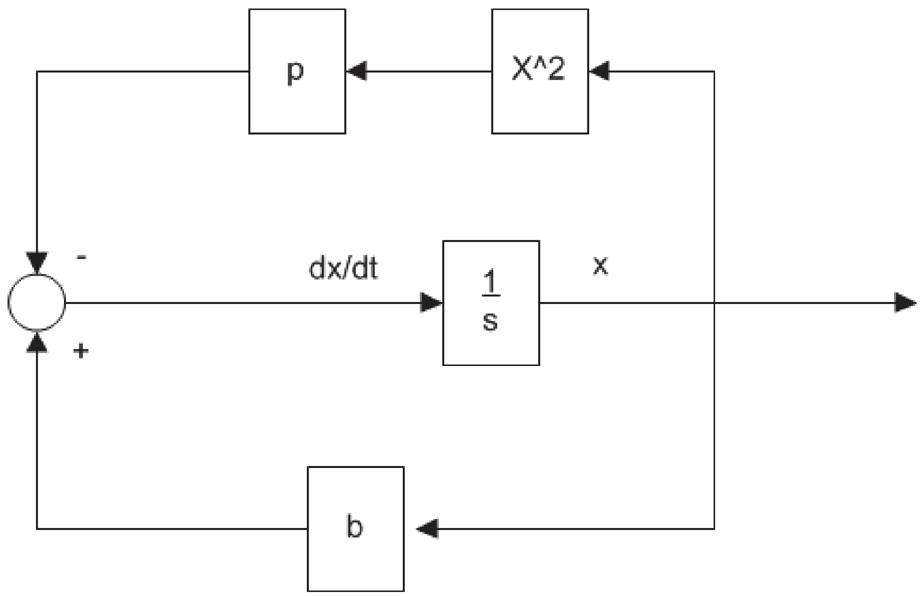
You should follow these steps:

1. Draw Block Diagram using "pen and paper".
2. Start LabVIEW and use the **Control and Simulation Loop** from Control Design and Simulation Palette in LabVIEW
3. Drag in the necessary Blocks from the palette.
4. Use the "**Connection Wire**" from the **Tools palette** and draw the necessary wires.
5. **Configure Simulation Parameters** (right-click on the Control and Simulation Loop border)
6. Start the Simulation. The Simulation result should be present in a plot.

[End of Task]

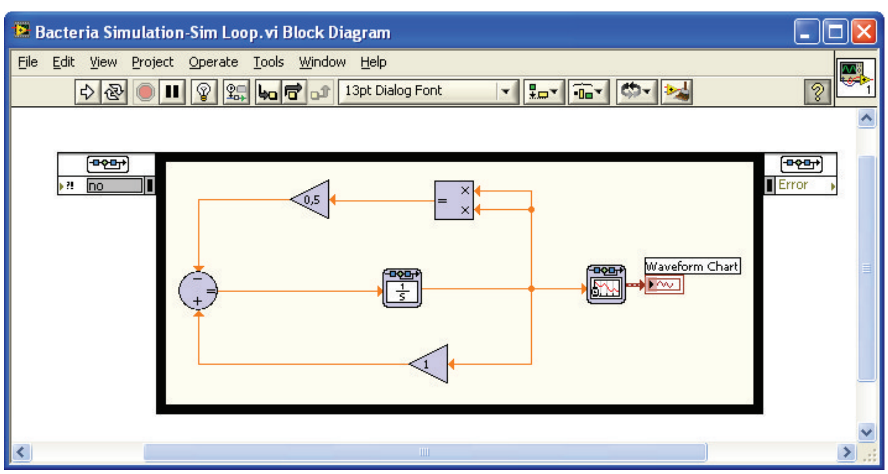
Solutions:

The Block Diagram becomes:

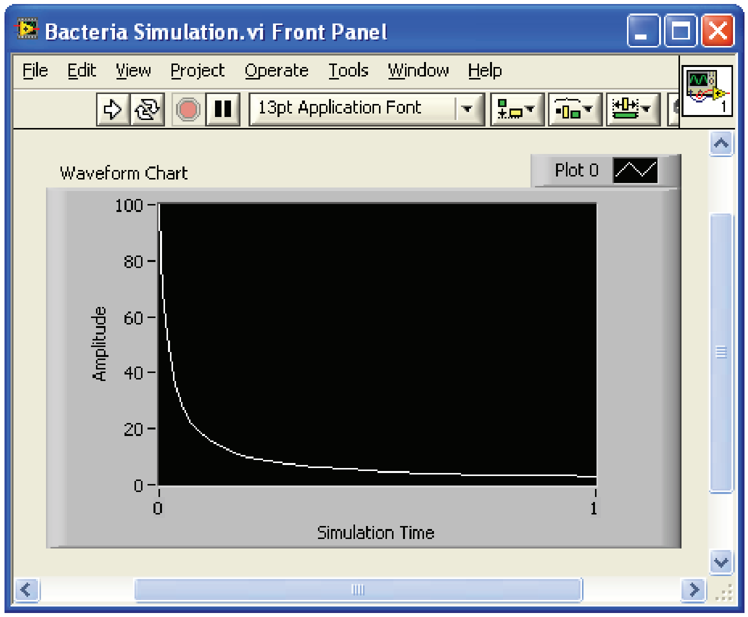


See "**Differential equations represented as block diagrams**" for more information about creating block Diagrams.

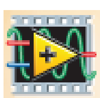
Below you see the "Control and Simulation Loop" and different blocks that may be used in this simulation:



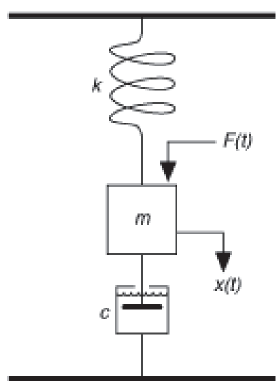
The Simulation Result should look like this:



Task 5: Simulation of a "spring-mass damper" system in LabVIEW



Use "LabVIEW Control Design and Simulation Module" and the "**Control and Simulation Loop**" in order to create a simulation of a spring-mass damper system.



The differential equation for the system is as follows:

$$\boxed{\ddot{x} = \frac{1}{m}(F - c\dot{x} - kx)}$$

Where.

F is an external force applied to the system

c is the damping constant of the spring

k is the stiffness of the spring

m is a mass

x is the position of the mass

$\dot{x}$ is the first derivative of the position, which equals the velocity of the mass

$\ddot{x}$ is the second derivative of the position, which equals the acceleration of the mass

Note! Draw a block diagram of the system using pen and paper before you implement the system in LabVIEW.

In the simulations you may set $F = -9.8, m = 1, d = 1 \; and \; k = 100$

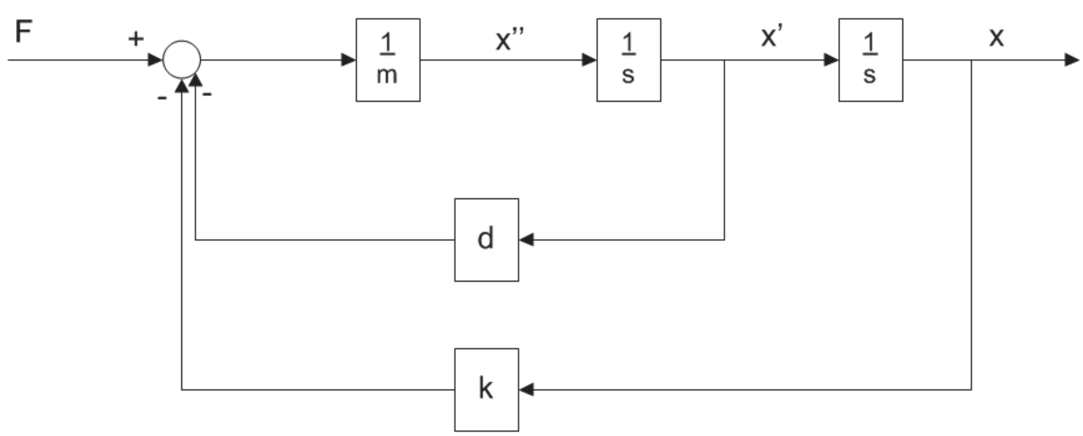
Then try to set $m = 10$ and see the difference.

[End of Task]

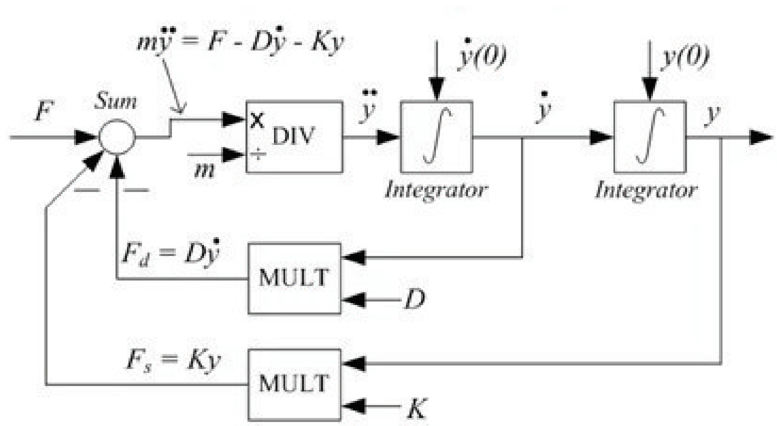
Solutions:

See the video "Simulation Palette Overview" by Finn Haugen for a step-by-step instructions or read the Tutorial "Control and Simulation in LabVIEW" for step-by-step instructions.

Block Diagram:

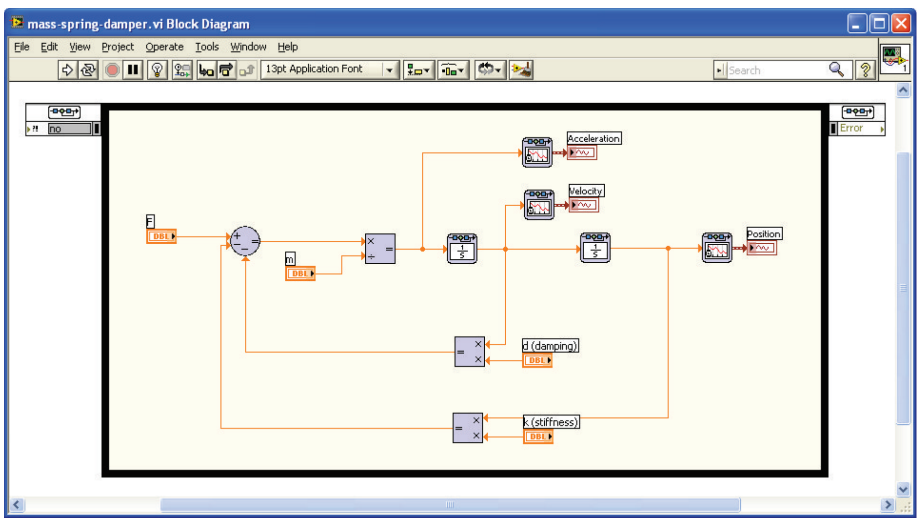


You may also use this notation:

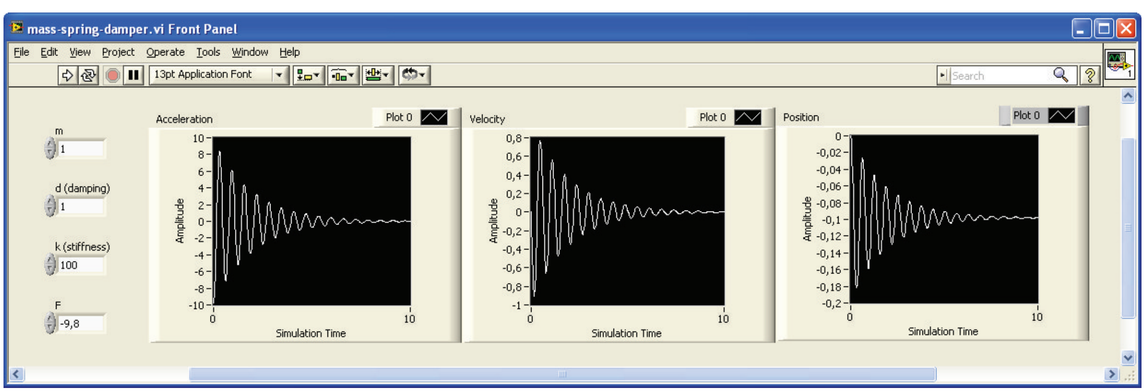


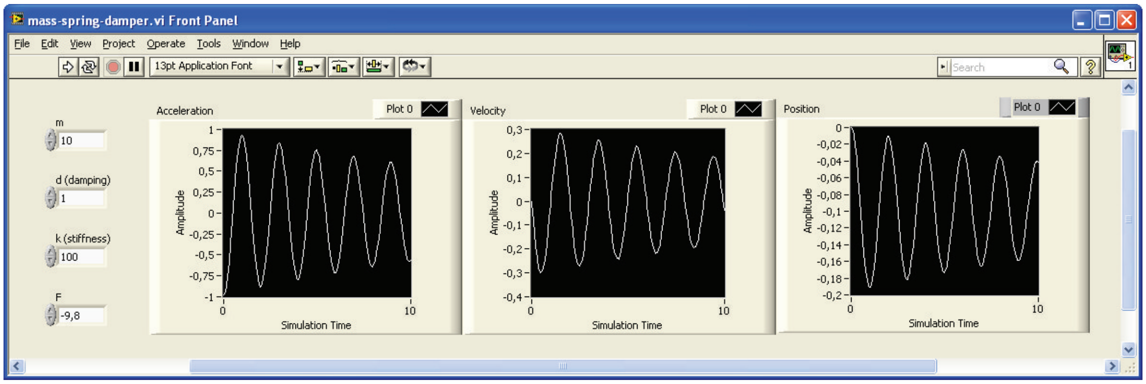
LabVIEW:

Block Diagram:



Front Panel:



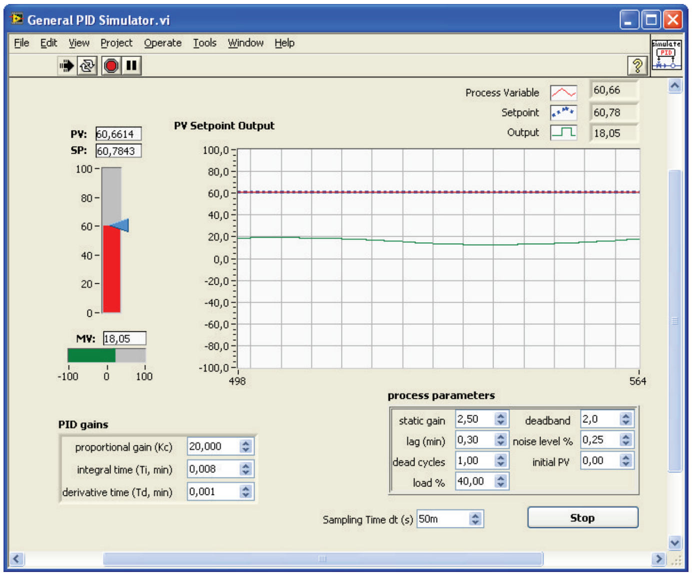

mass-spring-damper.vi Front Panel
File Edit View Project Operate Tools Window Help
13pt Application Font
Search
m
10
d (damping)
1
k (stiffness)
100
F
-9,8
Acceleration
Plot 0
Amplitude
Simulation Time
Velocity
Plot 0
Amplitude
Simulation Time
Position
Plot 0
Amplitude
Simulation Time

3 PID Control in LabVIEW

Task 6: Built-in PID Controller in LabVIEW

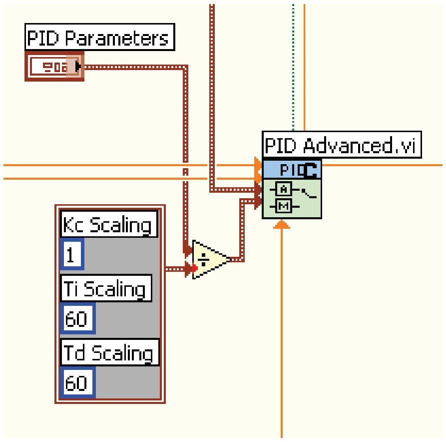
In this task you will use the example "**General PID Simulator.vi**" as a base for your simulation. Use the "NI Example Finder" (Help → Find Examples…) in order to find the VI in LabVIEW.

Run the example and so how it is implemented and how it works.



Make changes to the program (make sure to save it with another name) and use the "**PID Advance.vi**" function instead.

Note that T_i and T_d is in minutes, while it's normal to use seconds as the unit for these parameters. So it is recommended that you do like this in your code:

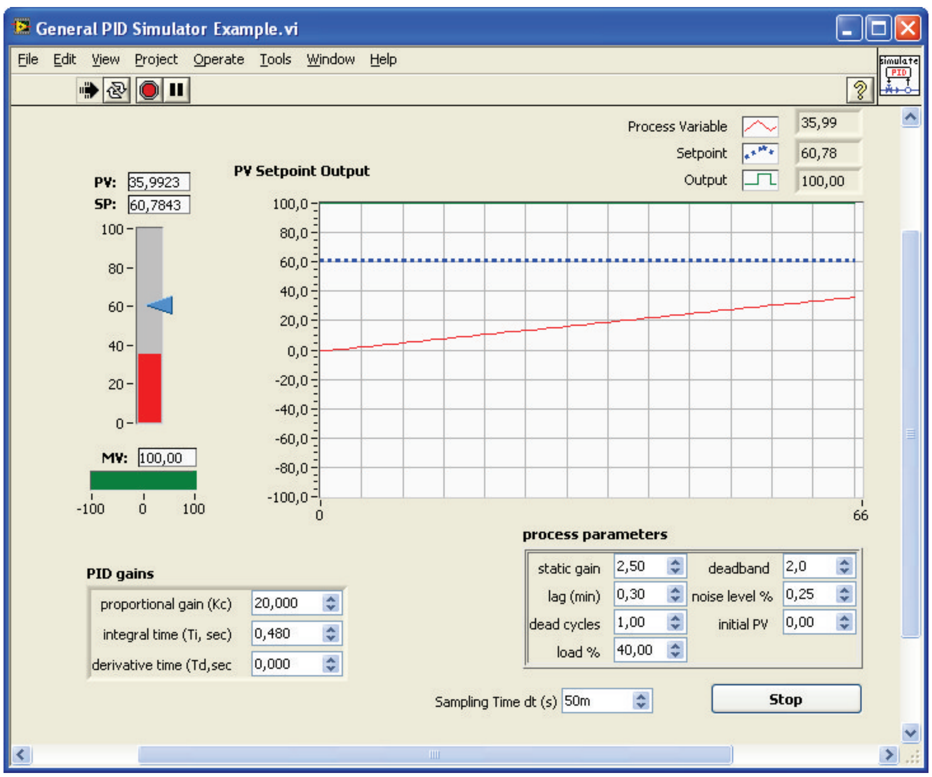


Try with different values for K_p, T_i and T_d and see what happens.

[End of Task]

Solutions:

Front Panel:



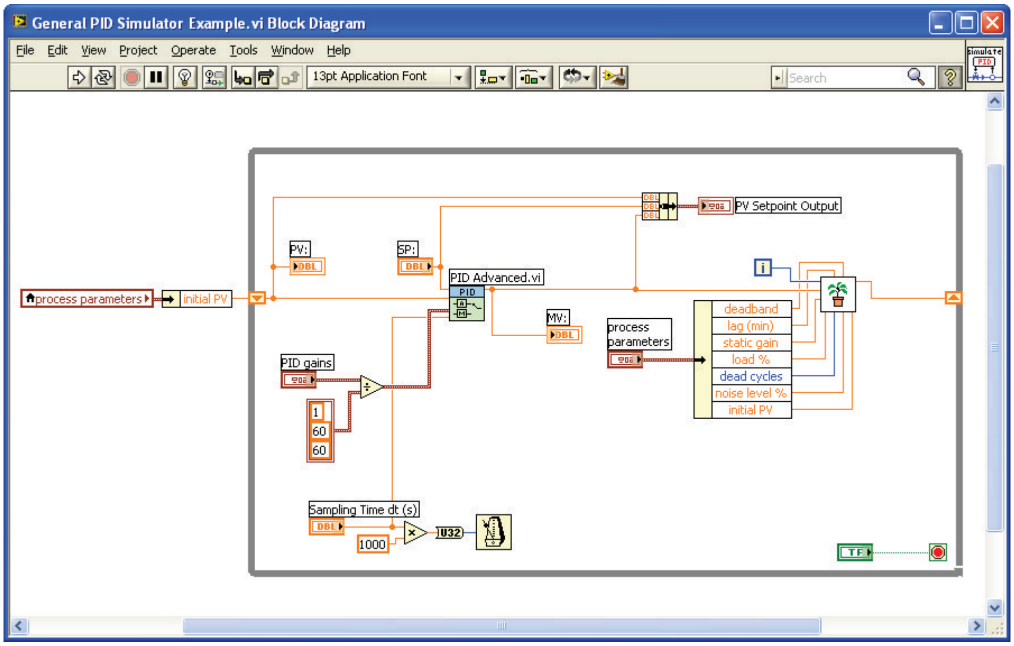
PID Parameters:

$$K_p = 20$$

$$T_i = 0.48$$

$$T_d = 0$$

Block Diagram:



4 Additional Tasks

Task 7: Simulation Loop

Given the following (nonlinear) system of a liquid tank:

$$\boxed{\dot{x} = -K_1\sqrt{x} + K_2 u}$$

where u is the input (control value), x is the output (level), K_1 is the valve outflow parameter, and K_2 is the pump inflow parameter.

Find proper values for K_1, K_2 and x_0. Apply a Step signal for u.

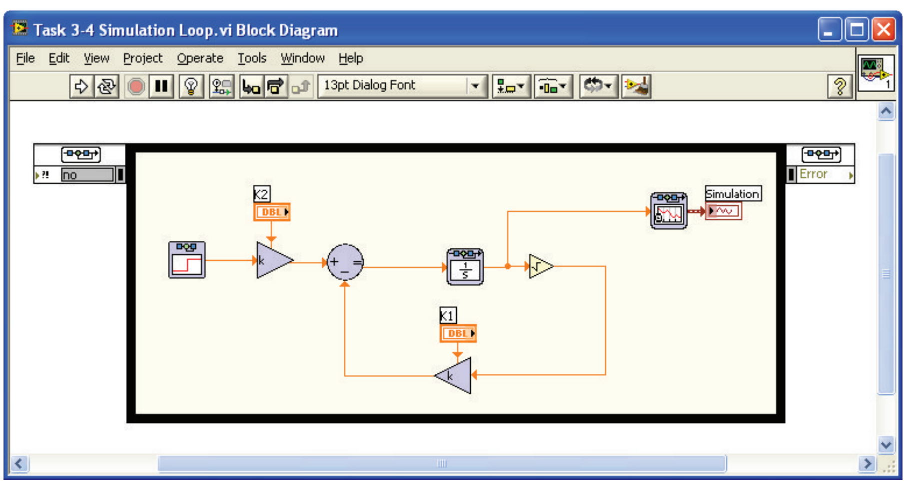
→ Create an application where you use "**LabVIEW Control Design and Simulation Module**" and the "**Control and Simulation Loop**" in order to simulate the system. Here you will use the simulation blocks from the Simulation palette in LabVIEW to create a continuous model.

Plot the results.

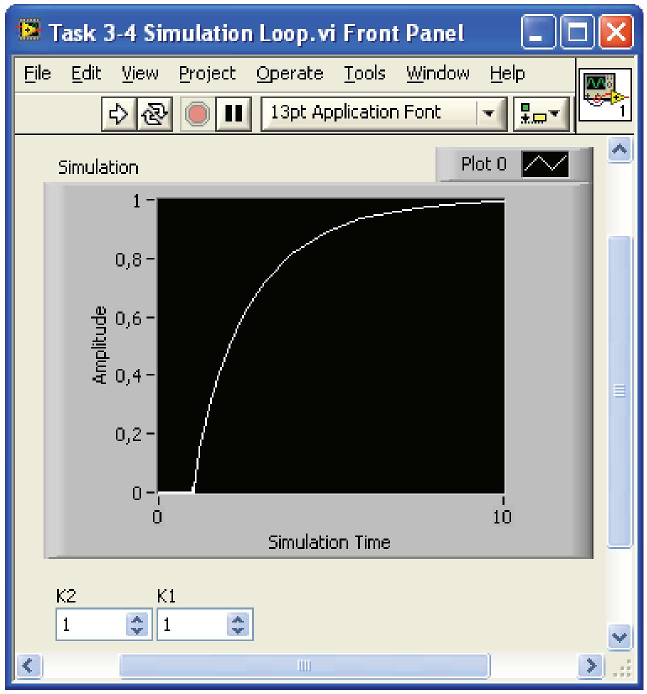
[End of Task]

Solutions:

Block Diagram:



Front Panel:



Task 8: Discretization

Given the following (nonlinear) system of a liquid tank:

$$\boxed{\dot{x} = -K_1\sqrt{x} + K_2 u}$$

→ Create a new application in LabVIEW where you **Simulate** the model using a **Formula Node** to implement the discrete model.

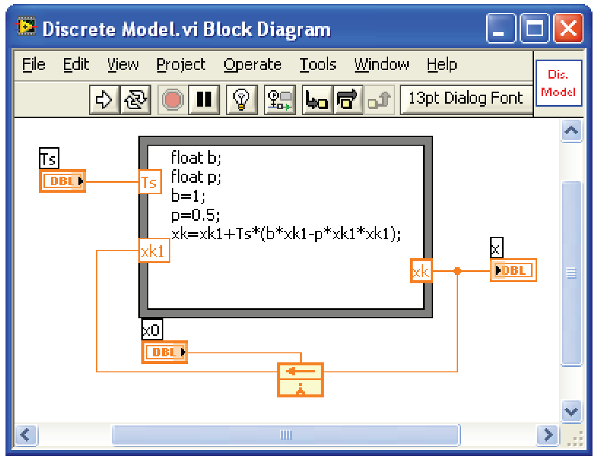
Use one of **Eulers** Discretization methods in order to create a **discrete model** of the system. Which of Eulers methods did you use? Why?

Use a While Loop or a For Loop and select a proper time-step T_s for the simulation. Use the same settings and values as in the previous case.

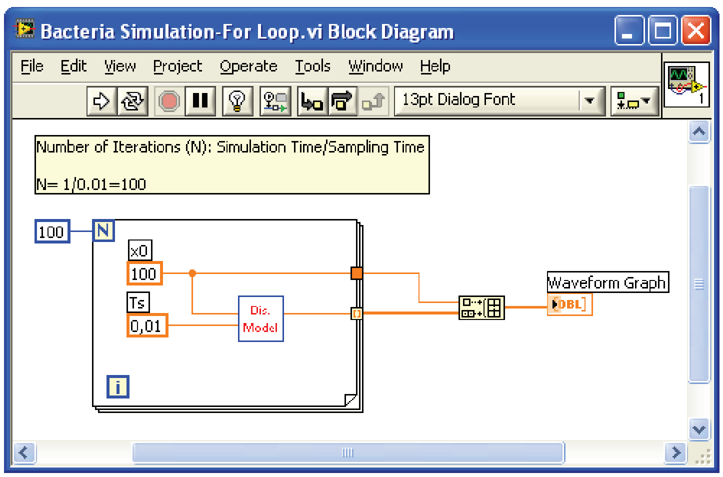
→ Compare the results from the 2 different methods used above. Do you get the same results?

Example:

Implementing the model as a SubVI using the Formula Node:



Use the model in a For Loop:



[End of Task]

Solutions:

Discretization:

$$\boxed{\dot{x} \quad = -K_1\sqrt{x} + K_2 u}$$

We use Euler Forward (because this is a nonlinear equation):

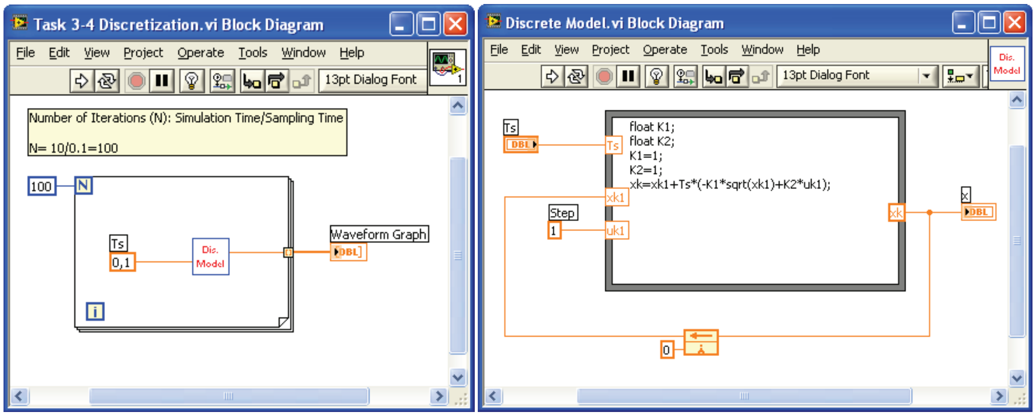
$$\dot{x} \quad \approx \frac{x_{k+1} - x_k}{T_s}$$

This gives:

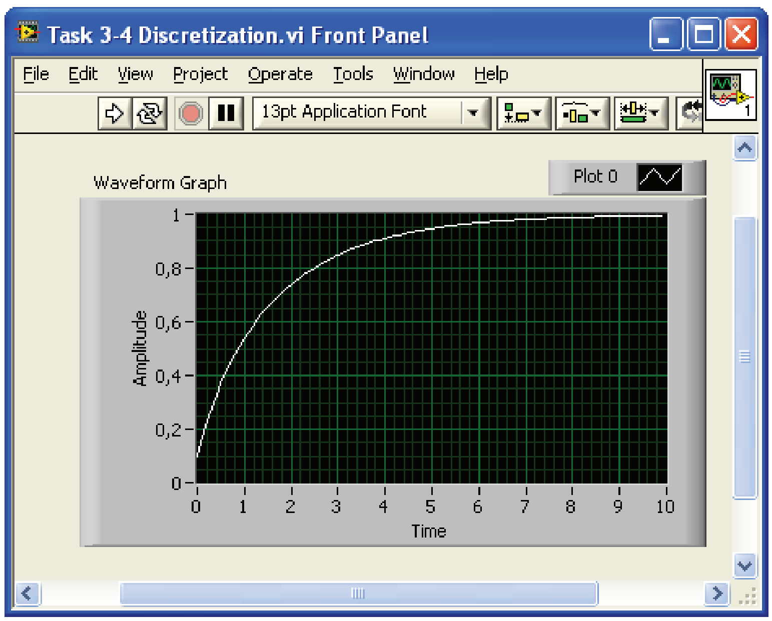
$$\frac{x_{k+1} - x_k}{T_s} = -K_1\sqrt{x} + K_2 u$$

$$\underline{\underline{x_k \; = x_{k-1} + T_s\left[-K_1\sqrt{x_{k-1}} + K_2 u_{k-1}\right]}}$$

Block Diagram:



Front Panel:



Task 9: Simulation and Control in LabVIEW

Control the system created above using a **discrete PI controller** that you create yourself.

A PI controller may be written:

$$\boxed{u(t) = u_0 + K_p e(t) + \frac{K_p}{T_i}\int_0^t e d\tau}$$

Where u is the controller output and e is the control error:

$$e(t) = r(t) - y(t)$$

Compare the results from the previous task.

[End of Task]

Solutions:

Discretization:

Given:

$$u(t) = u_0 + K_p e(t) + \frac{K_p}{T_i}\int_0^t e\,d\tau + K_p T_d \dot{e}(t) \tag{19}$$

Differentiating both sides of (19) gives[2]

$$\dot{u}(t) = \dot{u}_0 + K_p \dot{e}(t) + \frac{K_p}{T_i} e(t) + K_p T_d \ddot{e}(t) \tag{21}$$

Applying the Backward differentiation method (2) to $\dot{u}$, $\dot{e}$, and $\ddot{e}$ gives

$$\begin{aligned}
\frac{u(t_k) - u(t_{k-1})}{T_s} &= \frac{u_0(t_k) - u_0(t_{k-1})}{T_s} && (22)\\
&\quad + K_p \frac{e(t_k) - e(t_{k-1})}{T_s} && (23)\\
&\quad + \frac{K_p}{T_i} e(t_k) && (24)\\
&\quad + K_p T_d \frac{\dot{e}(t_k) - \dot{e}(t_{k-1})}{T_s} && (25)
\end{aligned}$$

Applying the Backward differentiation method on $\dot{e}_f(t_k)$ and $\dot{e}_f(t_{k-1})$ in (22) gives

$$
\begin{aligned}
\frac{u(t_k) - u(t_{k-1})}{T_s} &= \frac{u_0(t_k) - u_0(t_{k-1})}{T_s} && (26)\\
&+K_p \frac{e(t_k) - e(t_{k-1})}{T_s} && (27)\\
&+\frac{K_p}{T_i} e(t_k) && (28)\\
&+K_p T_d \frac{\frac{e(t_k)-e(t_{k-1})}{T_s} - \frac{e(t_{k-1})-e(t_{k-2})}{T_s}}{T_s} && (29)
\end{aligned}
$$

Solving for $u(t_k)$ finally gives the discrete-time PID controller:

$$
\begin{aligned}
u(t_k) &= u(t_{k-1}) + [u_0(t_k) - u_0(t_{k-1})] && (30)\\
&+K_p [e(t_k) - e(t_{k-1})] && (31)\\
&+\frac{K_p T_s}{T_i} e(t_k) && (32)\\
&+\frac{K_p T_d}{T_s} [e(t_k) - 2e(t_{k-1}) + e(t_{k-2})] && (33)
\end{aligned}
$$

1. First the *incremental control value* $\Delta u(t_k)$ is calculated:

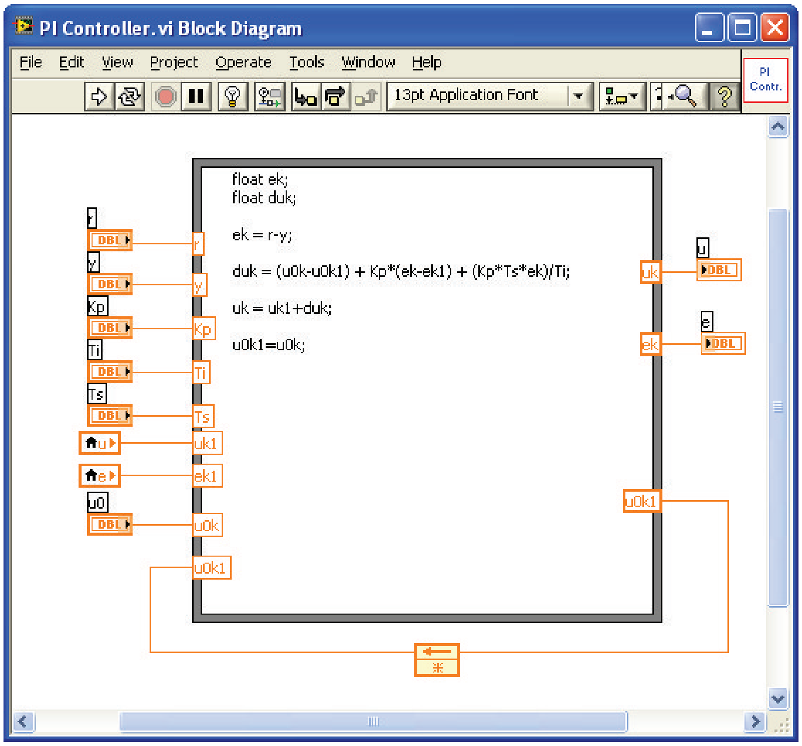
$$
\begin{aligned}
\Delta u(t_k) &= [u_0(t_k) - u_0(t_{k-1})] && (34)\\
&+K_p [e(t_k) - e(t_{k-1})] && (35)\\
&+\frac{K_p T_s}{T_i} e(t_k) && (36)\\
&+\frac{K_p T_d}{T_s} [e(t_k) - 2e(t_{k-1}) + e(t_{k-2})] && (37)
\end{aligned}
$$

2. Then the *total or absolute control value* is calculated with

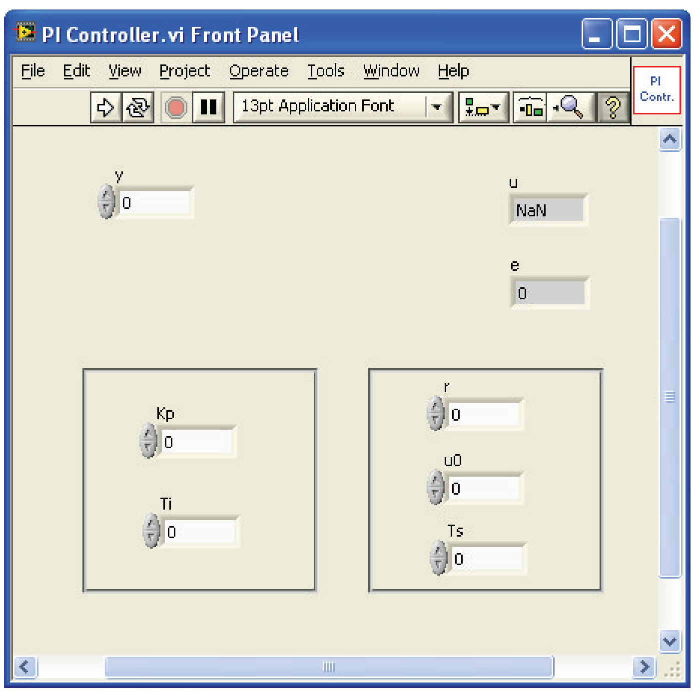
$$
u(t_k) = u(t_{k-1}) + \Delta u(t_k) \qquad (38)
$$

Discrete PI Controller in LabVIEW (implemented in SubVI using a Formula Node):

Block Diagram:

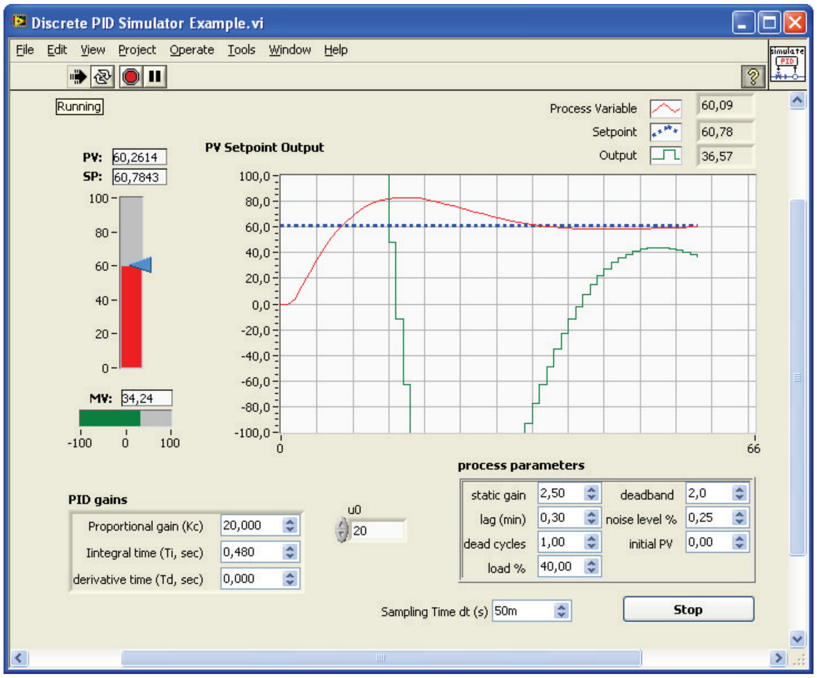


Front Panel:

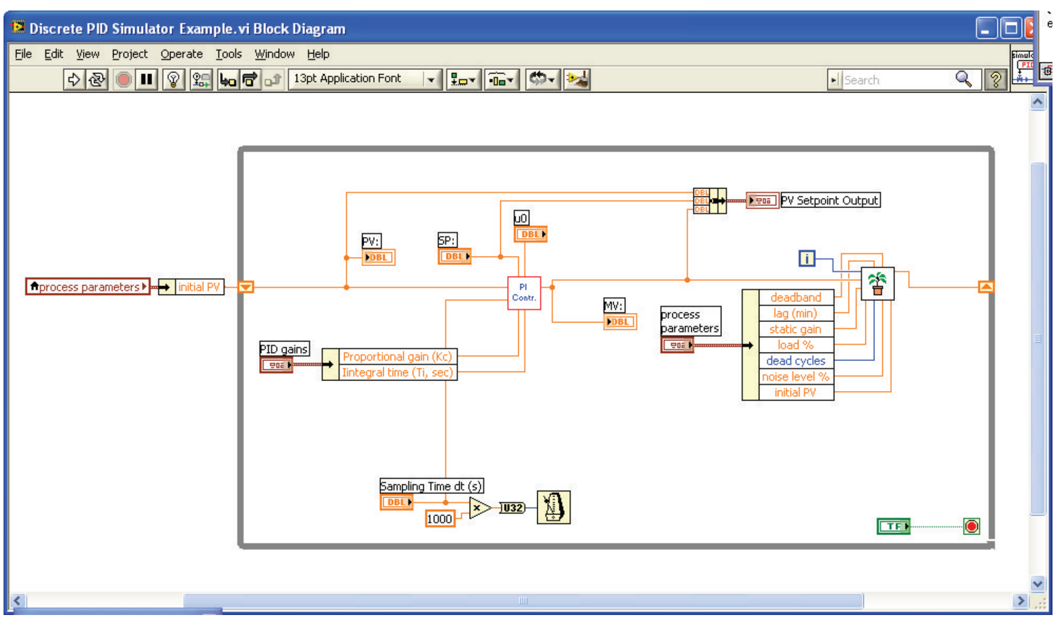


Test of the PI Controller:

Front Panel:



Block Diagram:

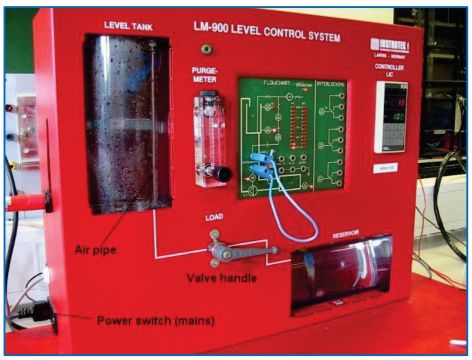


5 PID Control on real process

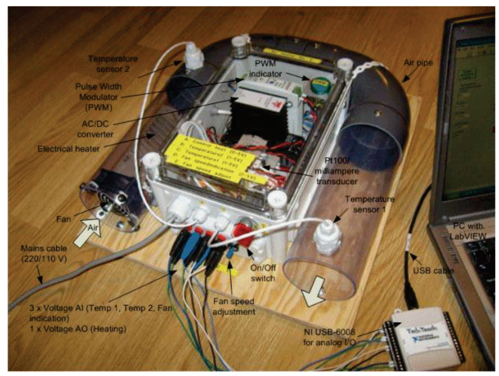
Create a PID Control system for a real process.

Below we see the Lab Equipment available for this assignment:

Level Tank



Air Heater



Documents of how to use the Level Tank/Air Heater and the USB-6008 DAQ device is available from http://home.hit.no/~hansha.

- **Level Tank**: http://home.hit.no/~hansha/?equipment=leveltank
- **Air Heater:** http://home.hit.no/~hansha/?equipment=airheater
- **USB-6008**: http://home.hit.no/~hansha/?equipment=usb6008

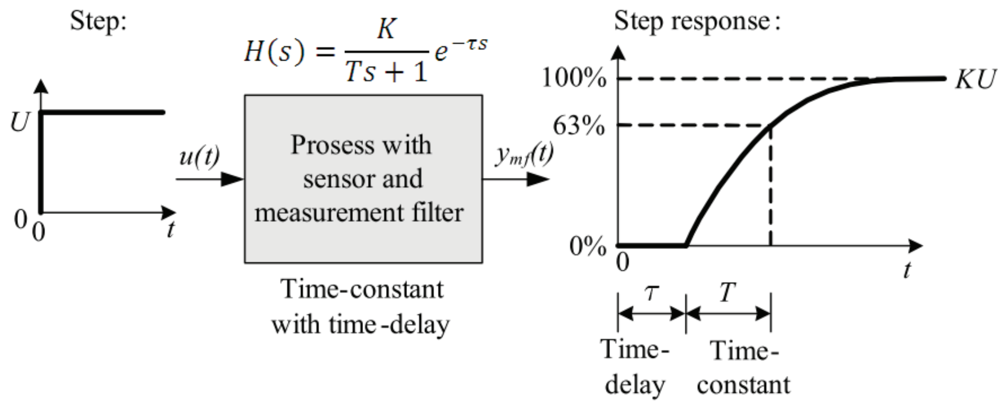
Find proper PID parameters. Use, e.g., Skogestad's method in order to find PID parameters.

Skogestad's method:

In this task we assume the following process:

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

You need to apply a step on the input (u) and then observe the response and the output, as shown below:



Here are the Skogestad's formulas for finding the PID parameters:

Process type	$H_{psf}(s)$ (process)	K_p	T_i	T_d
Integrator + delay	$\frac{K}{s}e^{-\tau s}$	$\frac{1}{K(T_C+\tau)}$	$c\,(T_C+\tau)$	0
Time-constant + delay	$\frac{K}{Ts+1}e^{-\tau s}$	$\frac{T}{K(T_C+\tau)}$	$\min\,[T,\,c\,(T_C+\tau)]$	0
Integr + time-const + del.	$\frac{K}{(Ts+1)s}e^{-\tau s}$	$\frac{1}{K(T_C+\tau)}$	$c\,(T_C+\tau)$	T
Two time-const + delay	$\frac{K}{(T_1s+1)(T_2s+1)}e^{-\tau s}$	$\frac{T_1}{K(T_C+\tau)}$	$\min\,[T_1,\,c\,(T_C+\tau)]$	T_2
Double integrator + delay	$\frac{K}{s^2}e^{-\tau s}$	$\frac{1}{4K(T_C+\tau)^2}$	$4\,(T_C+\tau)$	$4\,(T_C+\tau)$

Table 1: Skogestad's formulas for PI(D) tuning.

The Skogestad's formulas for this system are:

$$K_p = \frac{T}{K(T_c+\tau)}$$

$$T_i = min[T, c(T_c+\tau)]$$

$$T_d = 0$$

Air Heater: Set $T_C = \tau$ sec and $c = 1.5$.

Water Tank: Set $T_C = 10$ sec and $c = 1.5$.

For more details about the Skogestads method, please read this article: "**Model-based PID tuning with Skogestad's method**".

[End of Task]

Solutions:

Using Skogestads method on the Air Heater:

In this task we can set $T_C = \tau$ sec and $c = 1.5$.

We assume the following transfer function:

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

The Skogestad's formulas for this system are:

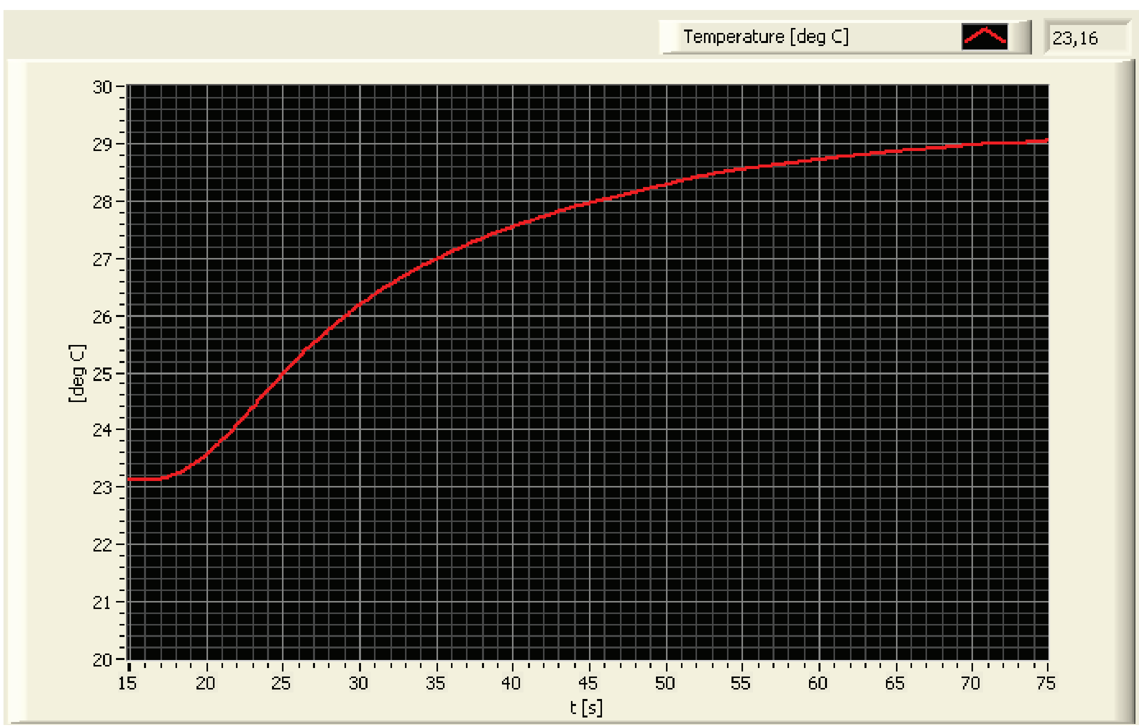
$$K_p = \frac{T}{K(T_c + \tau)}$$

$$T_i = min[T, c(T_c + \tau)]$$

$$T_d = 0$$

From a previous task we have:

The control signal is a step with $U = 1$ at $t = 15$.



From the plot above we can find:

Gain:

$$KU = 5.8$$

With $U = 1$ we get:

$$K = K_h = 5.8$$

Time-delay:

$$\tau = \theta_d = 2\,sec$$

Time constant:

$$T = \theta_t = T_{63} = 34 - 17 = 17sec$$

Comments: T is too small?? Have the response above reached steady-state?

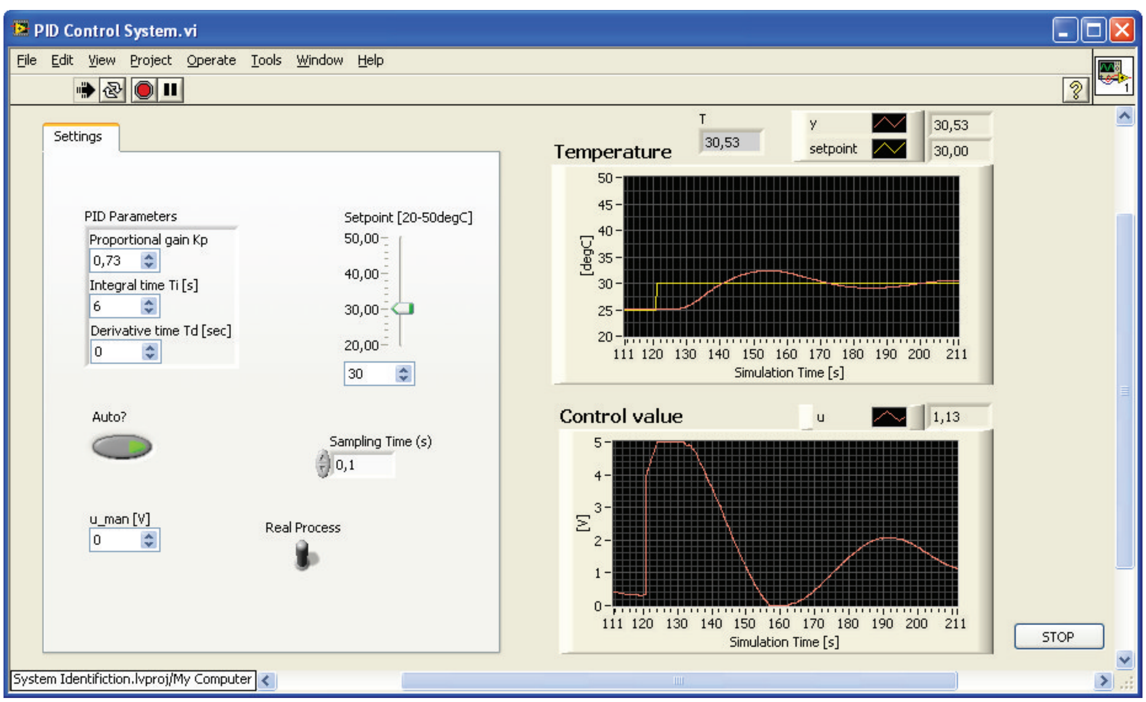
This gives the following PID parameters from Skogestad's formulas:

$$K_p = \frac{T}{K(T_c + \tau)} = \frac{17}{5.8(2+2)} = 0.73$$

$$T_i = min[T, c(T_c + \tau)] = min[17, 1.5(2+2)] = 6$$

$$T_d = 0$$

Trying out the PID parameters found above:

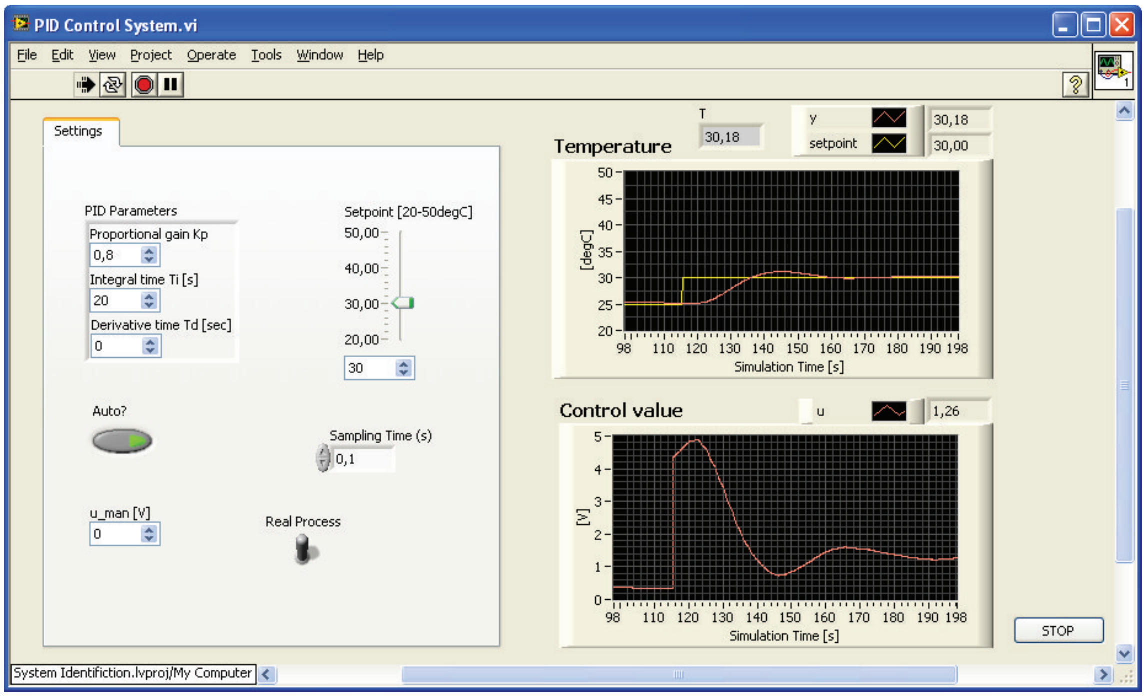


Trying other PID Parameters:

Here are some values that worked fine for me from another assignment:

$$\underline{\underline{K_p = 0.8, T_i = 20, T_d = 0}}$$

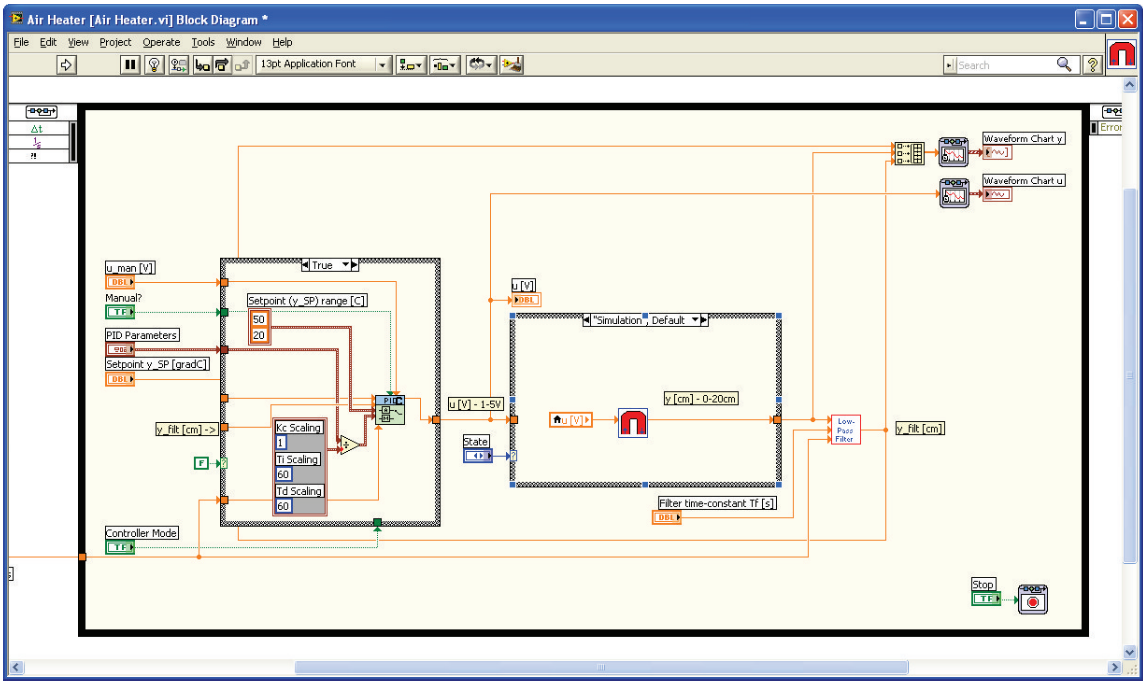
These PID parameters gave the following results:



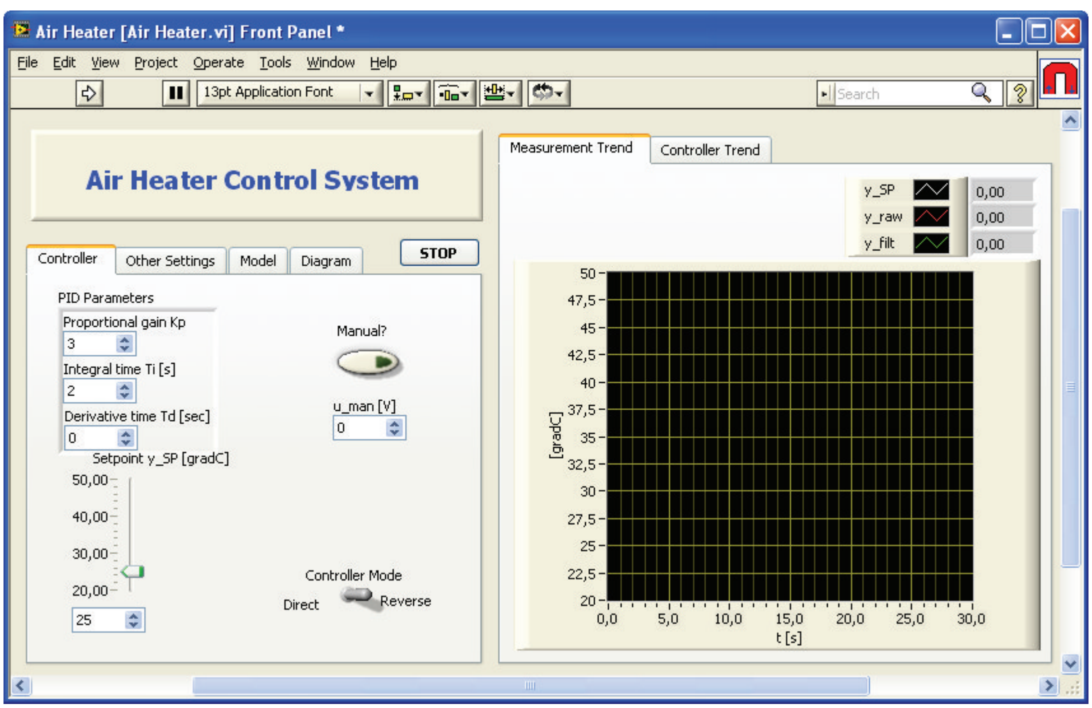
These PID parameters gives a better result.

<u>LabVIEW Program for Air Heater – Example II:</u>

Block Diagram:


Air Heater [Air Heater.vi] Block Diagram *
u_man [V]
Manual?
PID Parameters
Setpoint y_SP [gradC]
Setpoint (y_SP) range [C]
50
20
Kc Scaling
Ti Scaling
Td Scaling
y_filt [cm] ->
Controller Mode
u [V]
u [V] - 1-5V
State
"Simulation", Default
y [cm] - 0-20cm
Filter time-constant Tf [s]
y_filt [cm]
Waveform Chart y
Waveform Chart u
Stop

Front Panel:


Air Heater [Air Heater.vi] Front Panel *
Air Heater Control System
Controller
Other Settings
Model
Diagram
STOP
PID Parameters
Proportional gain Kp
Integral time Ti [s]
Derivative time Td [sec]
Setpoint y_SP [gradC]
Manual?
u_man [V]
Controller Mode
Direct
Reverse
Measurement Trend
Controller Trend
y_SP
y_raw
y_filt
[gradC]
t [s]

Høgskolen i Telemark

Telemark University College

Faculty of Technology

Kjølnes Ring 56

N-3914 Porsgrunn, Norway

www.hit.no

Hans-Petter Halvorsen, M.Sc.

Telemark University College

Department of Electrical Engineering, Information Technology and Cybernetics

Phone: +47 3557 5158

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

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

Room: B-237a
