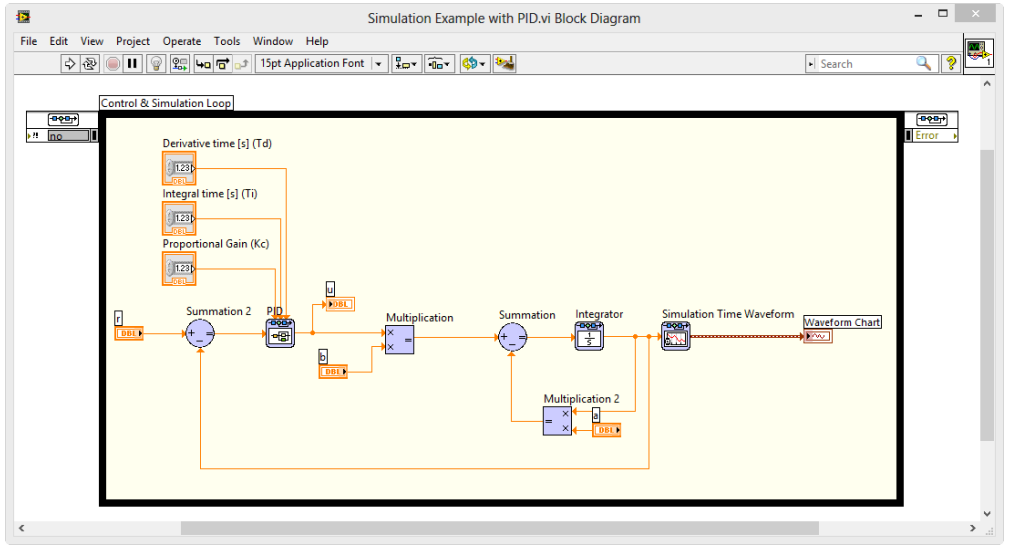
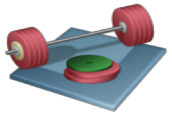
https://www.halvorsen.blog

Simulation in LabVIEW





with Self-paced Step-by-Step Exercises

Hans-Petter Halvorsen

LabVIEW Installation

You need the following Software

- LabVIEW (LabVIEW Professional Development System 32-Bit: English)
- NI-DAQmx (Hardware Driver for NI USB-6008, NI TC-01, etc.)
- LabVIEW Control Design and Simulation Module
- LabVIEW MathScript RT Module

Note! These packages are separate downloads!

All LabVIEW Software can be downloaded here: www.ni.com/download

Contents

- Introduction to LabVIEW
- Installation
- **Block Diagram Simulation based on differential Equations**
  - **Simulation Loop**
- **PID Control** with built-in PID blocks/functions
- Creating and using **Simulation Subsystems**
- Simulations using a While Loop with Subsystems inside
- **Discrete** Simulation
  - **Formula Node**
- MathScript

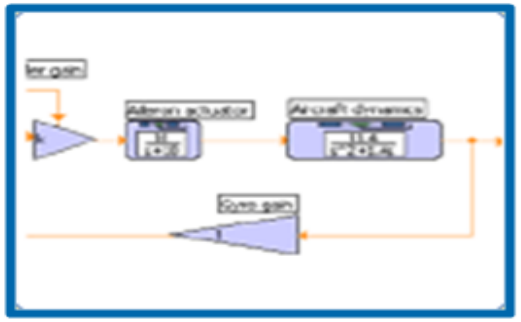
High-Level Design Tools

Configuration

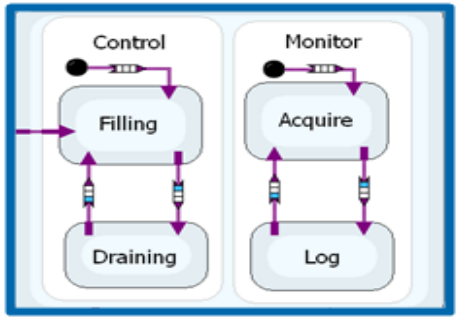
Textual Math

```
1 c = 0.285 + 0.013i;
2 [X Y] = meshgrid(x, y);
3 z = X + i*Y;
4 for k=1:30
5     z = z.^2 + c;
6 end
```

Simulation



Statechart



LabVIEW

Graphical Programming

National Instruments is the vendor of LabVIEW

National Instruments creates both **Hardware** and **Software**

Desktop Platform

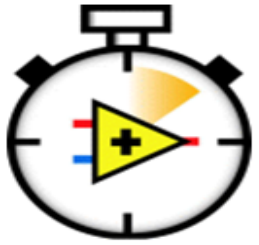
Linux®

Macintosh

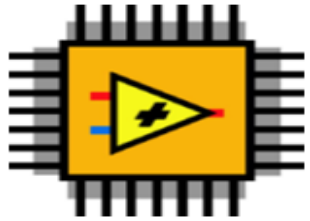
Windows

Embedded Platform

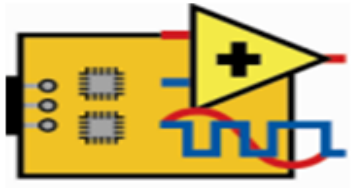
Real-Time



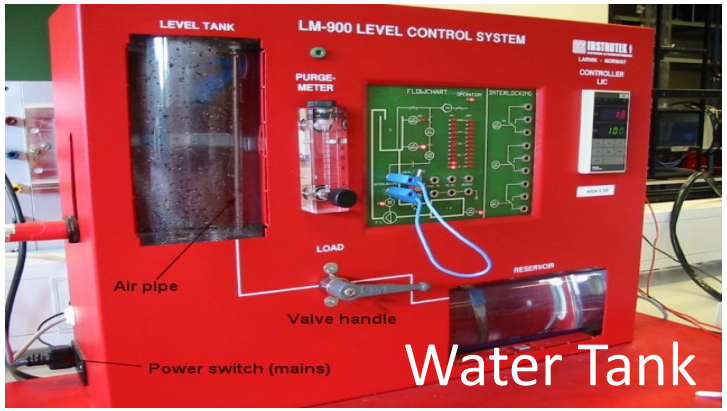
FPGA



MPU



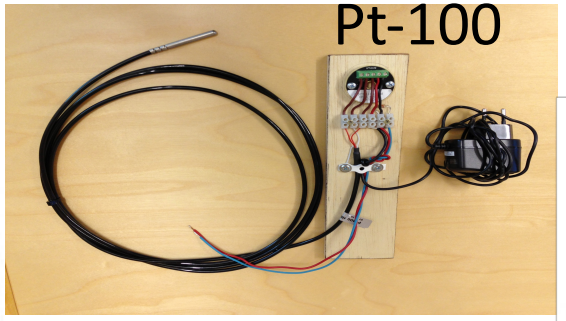
Hardware Examples



Water Tank

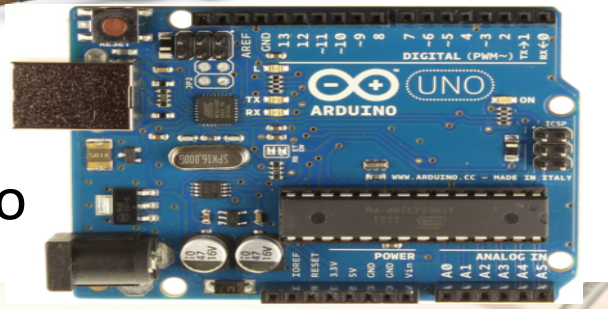
USB-6008

Wi-Fi DAQ


Pt-100


ZigBee


TC-01



Arduino

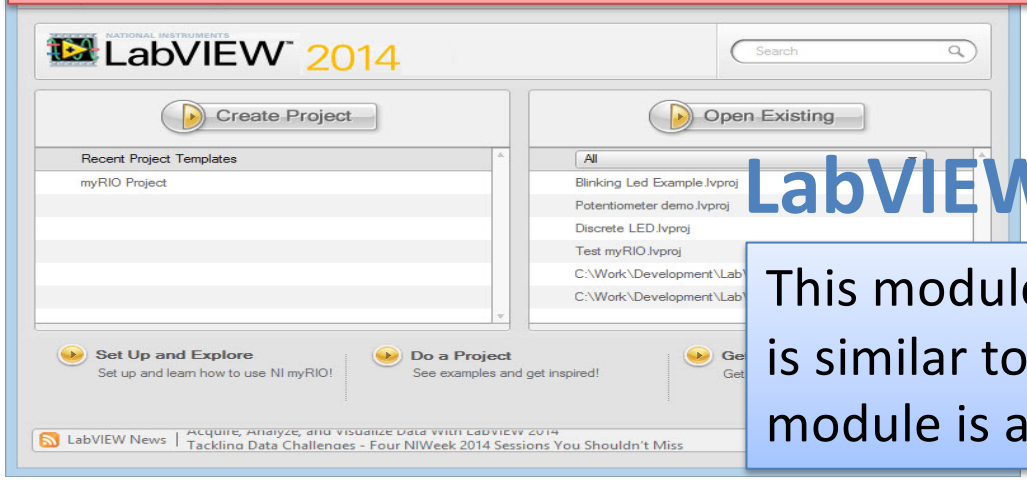

cRIO

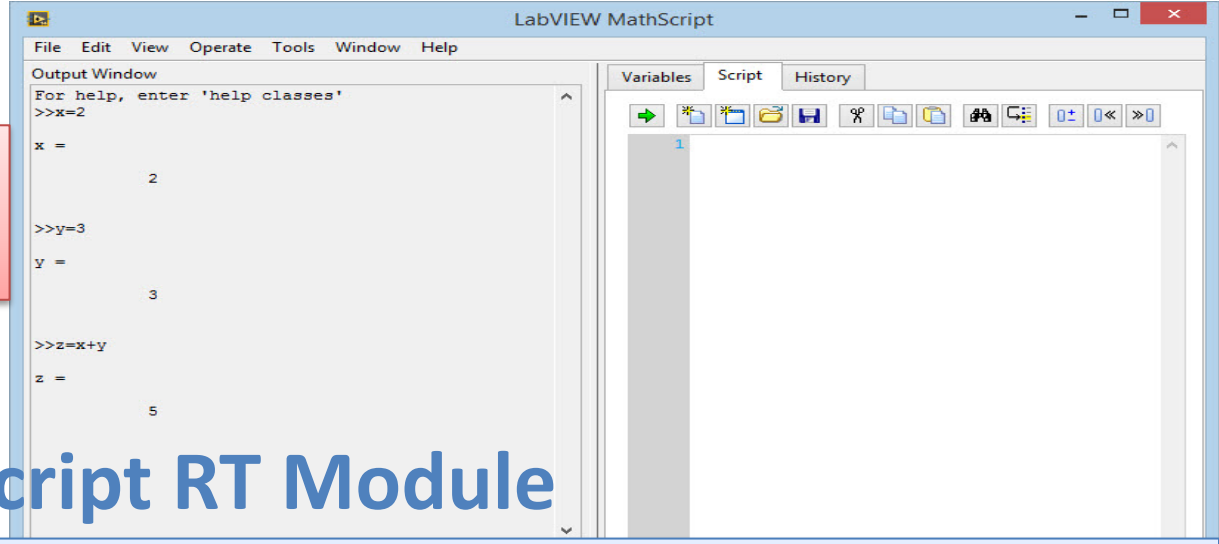

NOx Sensor

Vision System

LabVIEW

This is the core LabVIEW installation that installs the LabVIEW Programming Environment.



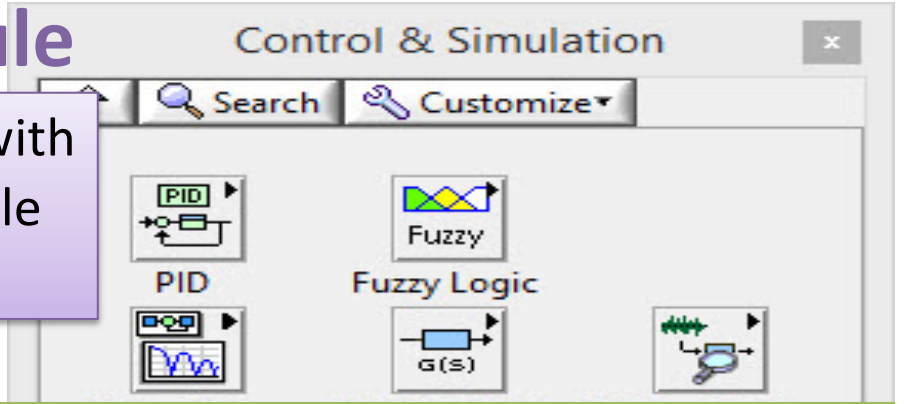


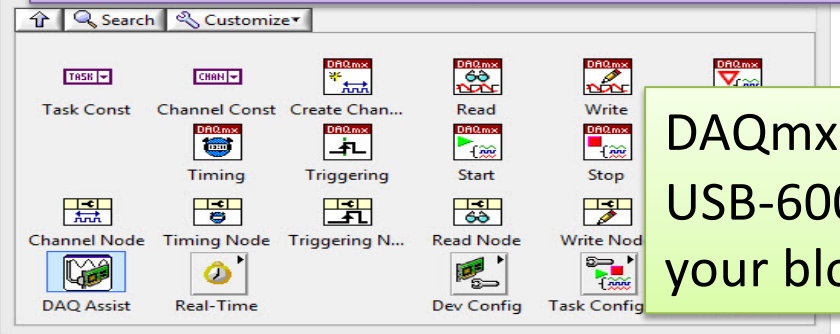
LabVIEW MathScript RT Module

This module is a text-based tool that is very similar to MATLAB. The syntax is similar to MATLAB, you can create and run so-called m files, etc. The module is available from the Tools menu inside LabVIEW.

LabVIEW Control Design and Simulation Module

This module is used for creating Control and Simulation applications with LabVIEW. Here you will find PID controllers, etc. The module is available as a palette on your block diagram.





NI-DAQmx

DAQmx is the Hardware Driver needed in order to use hardware devices like NI USB-6008, NI TC-01, etc. inside LabVIEW. The module is available as a palette on your block diagram.

http://www.ni.com/pdf/manuals/376039a.pdf

NATIONAL INSTRUMENTS

LabVIEW™ Quick Reference Guide

Keyboard Shortcuts

File

Ctrl-N	Create new VI
Ctrl-S	Save VI
Ctrl-P	Print

Edit

Ctrl-V	Paste object
Ctrl-U	Clean up diagram
Ctrl-Space	Activate quick drop
Ctrl-B	Remove broken wires
Ctrl-C	Copy an object
Ctrl-X	Cut object
Ctrl-Z	Undo last action
Ctrl-Shift-Z	Redo last action

Operate

Ctrl-R	Run VI
Ctrl-.	Abort VI

Window

Ctrl-E	Display block diagram/ front panel
Right-Click	Display controls/ functions palette
Shift-Right-Click	Display tools palette
Ctrl-T	Tile block diagram and front panel windows

Help

Ctrl-H	Display context help

Editing Tools

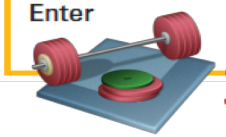
Tool	Icon	Description
Show Context Help		Display the context help window
Text Settings 15pt Application Font		Change the font setting for the VI, including size, style, and color
Align Objects		Align selected objects
Distribute Objects		Space objects evenly
Resize Objects		Resize multiple front panel objects to the same size
Reorder		Reorder the layers of the objects
Clean Up Diagram		Rearrange wires and objects on the block diagram
Enter		Appears when a new value is available to replace an old value

Debugging Tools

Tool	Icon	Description
Run		Execute the VI
List Errors		List errors that prevent the VI from running
Run Continuously		Execute the VI continuously until abort or pause is pressed
Abort Execution		Stop VI execution immediately
Execution Highlighting		Animate data movement on the block diagram wires
Pause		Temporarily stop execution to debug a portion of the VI
Step Into		Single-step into a subVI or structure to debug it
Step Over		Execute a subVI or structure and pause at the next one
Step Out		Execute a subVI or structure and resume single-stepping

Tools Palette

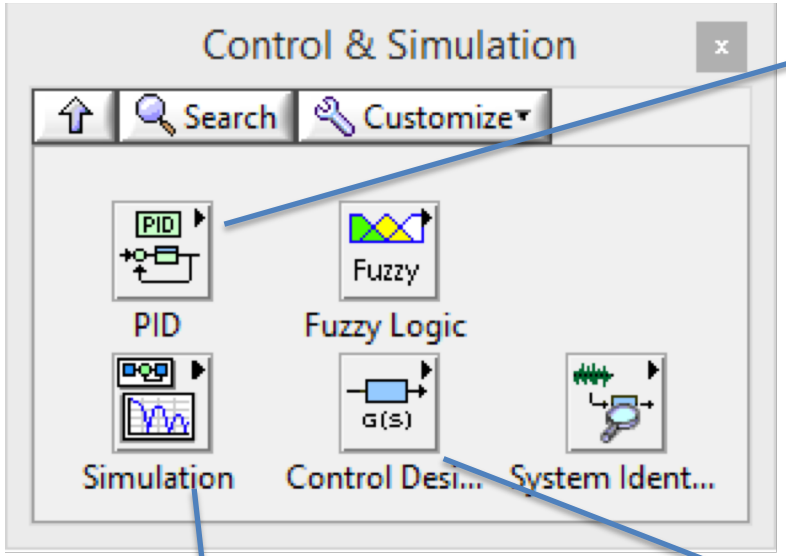
Tool	Icon	Description
Automatic Tool Selection		Automatically choose the appropriate tool
Operating Tool		Change the value of a control or select the text within a control
Positioning Tool		Position, resize, and select objects
Labeling Tool		Edit text and create free labels
Wiring Tool		Wire objects together on a block diagram
Scrolling Tool		Scroll the window without using the scroll bars
Breakpoint Tool (Used for debugging)		Set breakpoints on VIs, functions, wires, loops, sequences, and cases
Probe Tool (Used for debugging)		Create probes on wires and display intermediate values on a wire in a running VI
Get Color Tool		Copy colors for pasting with the Color Tool
Coloring Tool		Set the foreground and background colors



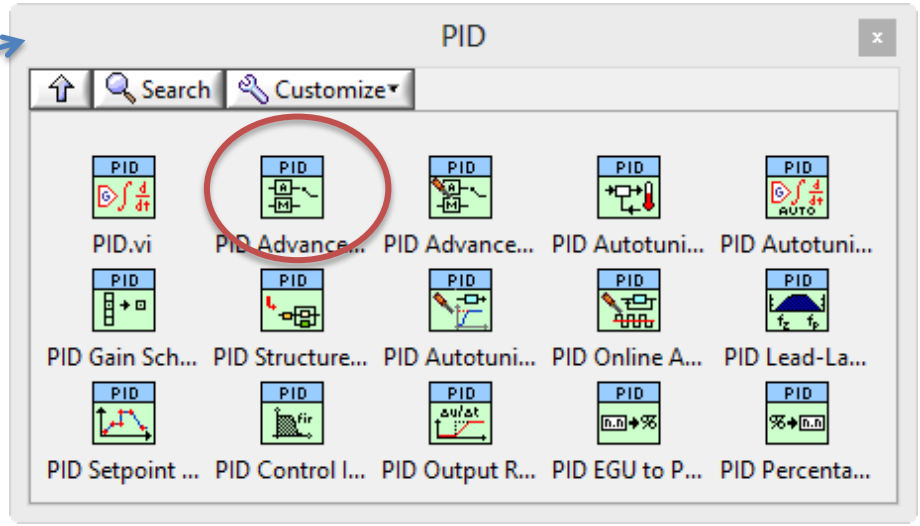
Students: Try some of these Shortcuts and Tools

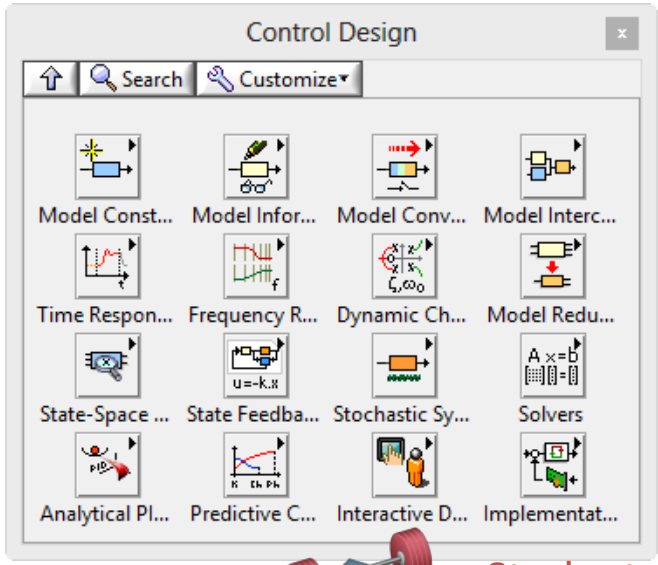
Control and Simulation in LabVIEW

Control Design & Simulation Palette in LabVIEW

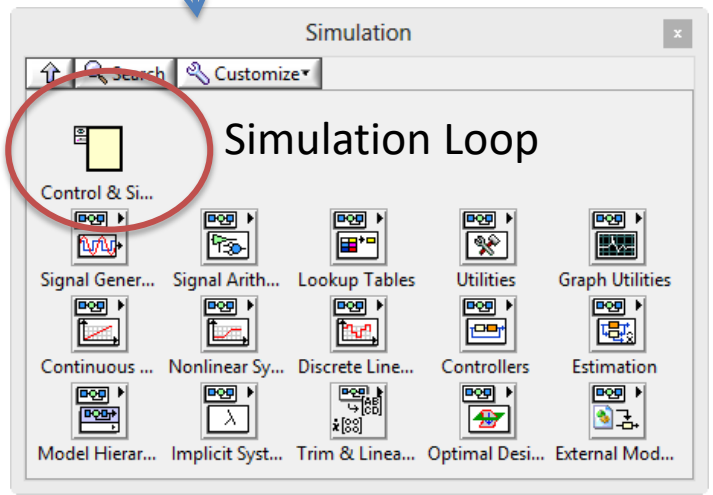


PID Palette in LabVIEW

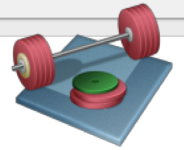




Control Design Palette in LabVIEW

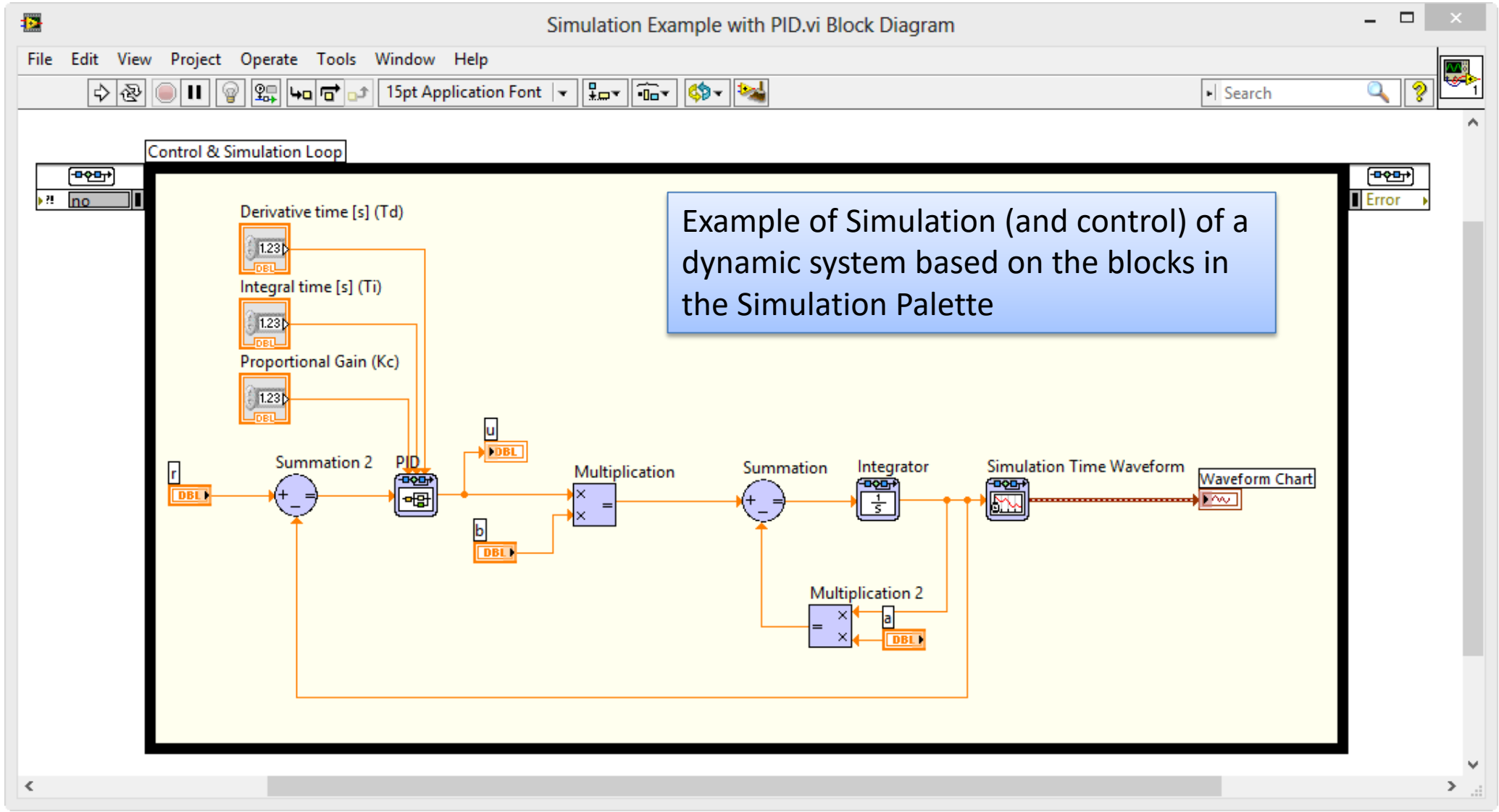


Simulation Palette in LabVIEW



Students: Check that you have all these palettes. Open the different subpalettes, etc.

LabVIEW Control and Simulation Example



We are going to learn to create such a system (and much more)!

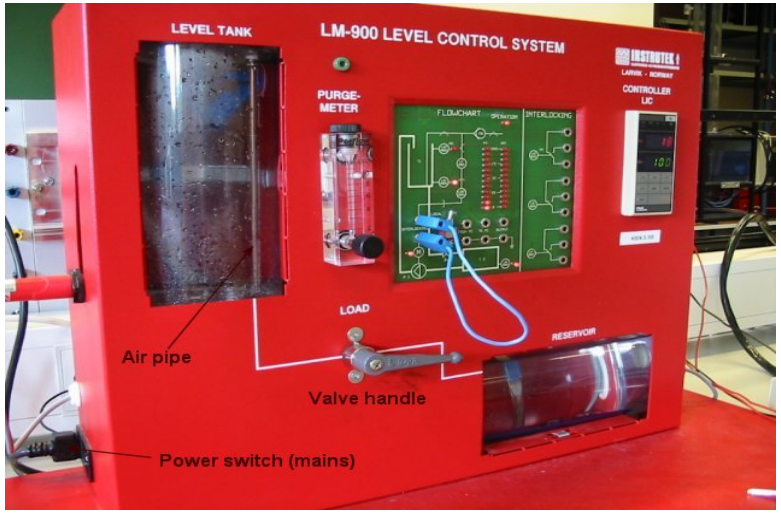
https://www.halvorsen.blog

Modelling of Dynamic Systems

Hans-Petter Halvorsen

Dynamic Systems Examples

Water Tank:



h – Level in the tank

Mathematical Models (differential equations):

Alt 1 (Integrator):

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

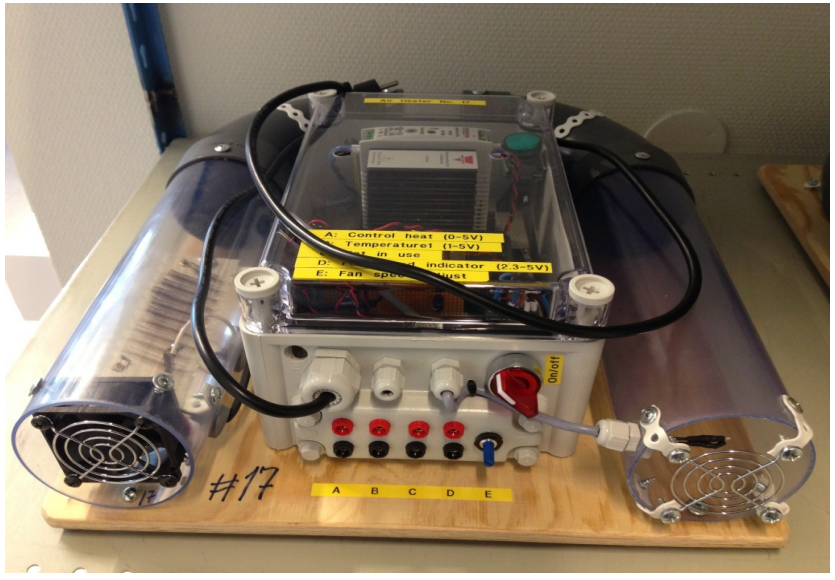
Alt 2 (Time constant/1.order):

$$\dot{h} = \frac{1}{A}\left[K_p u - K_v h\right]$$

Alt 3 (Nonlinear):

$$\dot{h} = \frac{1}{A}\left[K_p u - K_v \sqrt{\frac{\rho g h}{G}}\right]$$

Air Heater:

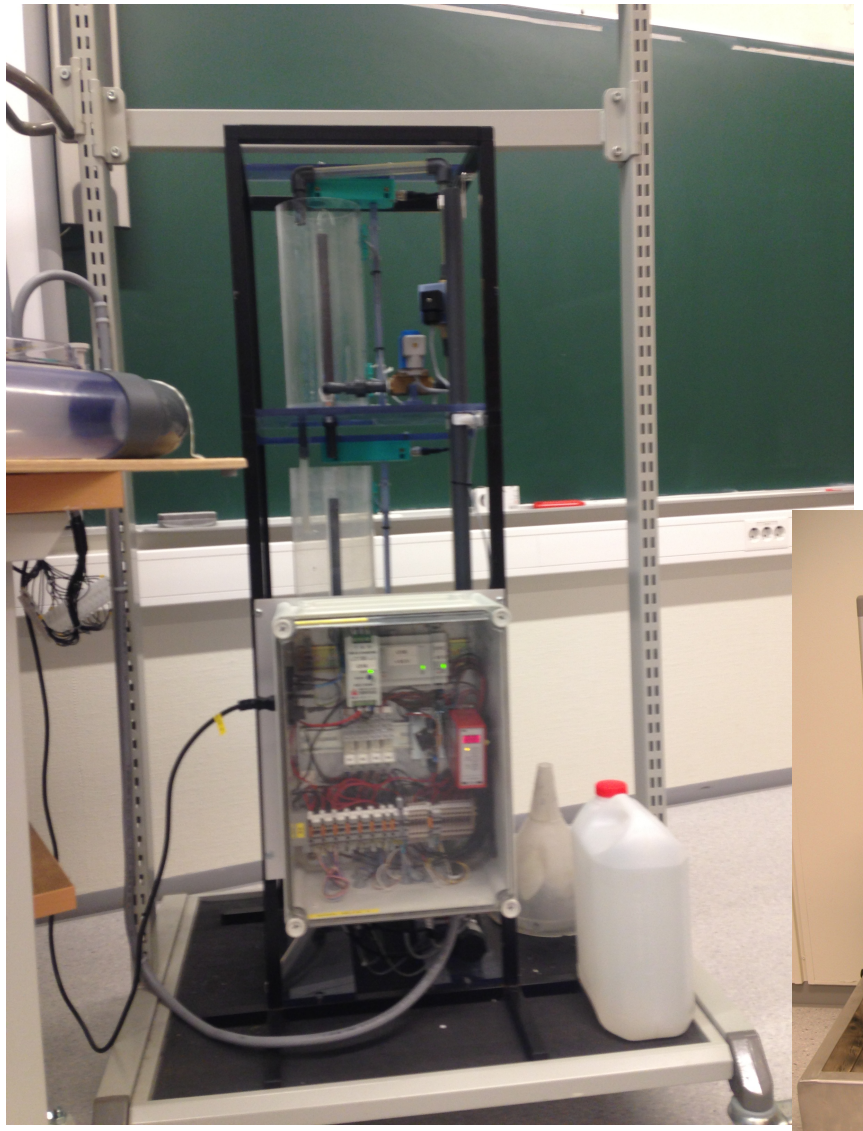


$$\dot{T}_{out} = \frac{1}{\theta_t}\{-T_{out} + [K_h u(t-\theta_d) + T_{env}]\}$$

T – Temperature in the tube

Dynamic Systems Examples

2 Tank:



Mathematical Models (differential equations):

$$A_1 \frac{dy_1}{dt} = Q_p - Q_1 - Q_2$$

$$A_2 \frac{dy_2}{dt} = Q_1 - Q_3 - Q_4$$

4 Tank:

$$\frac{dh_1}{dt} = -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1} v_1$$

$$\frac{dh_2}{dt} = -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2} v_2$$

$$\frac{dh_3}{dt} = -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_2) k_2}{A_3} v_2$$

$$\frac{dh_4}{dt} = -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_1) k_1}{A_4} v_1$$

Dynamic Systems

Dynamic system represented as a differential equation

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

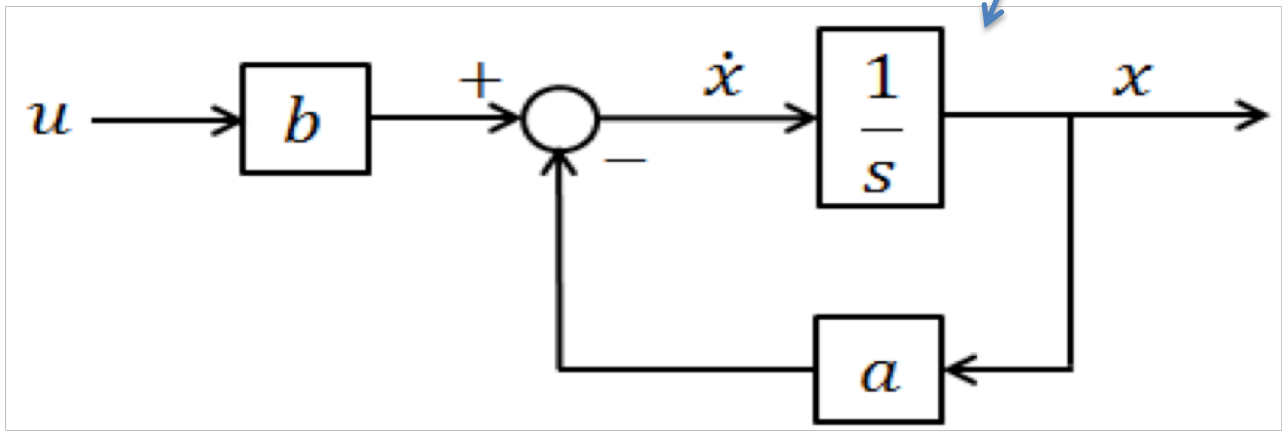
$\frac{dx}{dt}$

Note

We can "easily" create a block diagram from the differential equation(s)

Dynamic system represented as a block diagram

Integrator symbol



When we have the block diagram for the system, we can easily implement it in LabVIEW

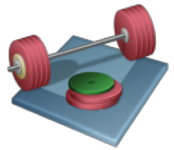
Block Diagram Examples

Example 1:

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

Example 2:

$$\dot{y} = -\frac{1}{4}y + \frac{1}{2}u$$

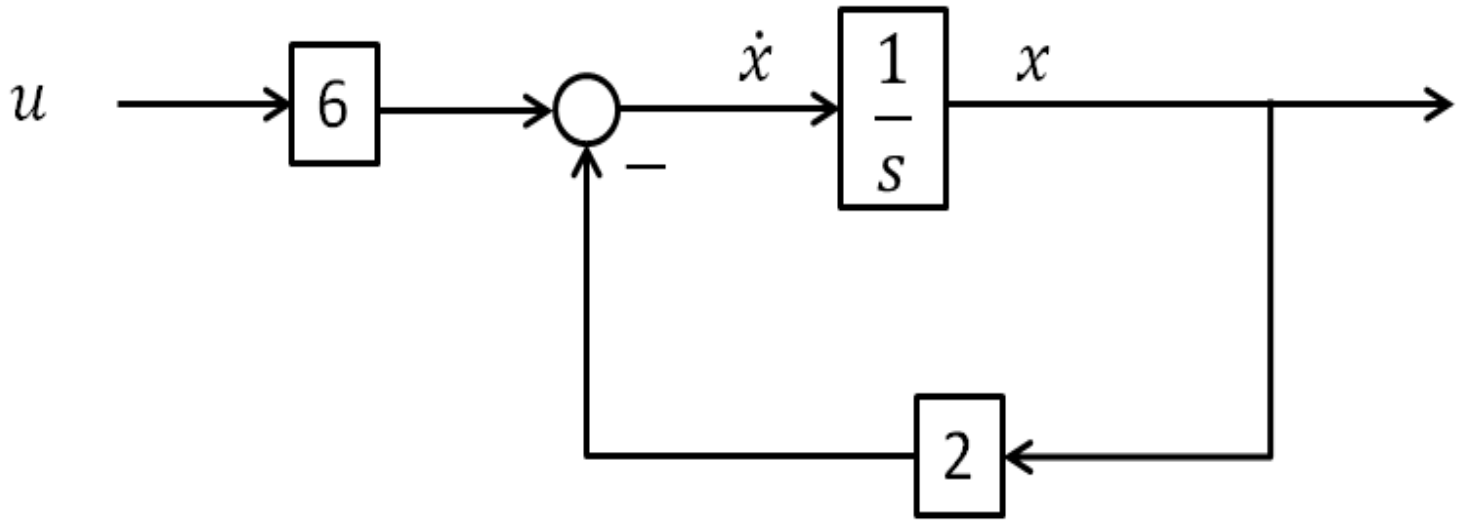


Students: Find the block diagrams for the differential equations above (pen & paper)

Block Diagrams - Solutions

Example 1:

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

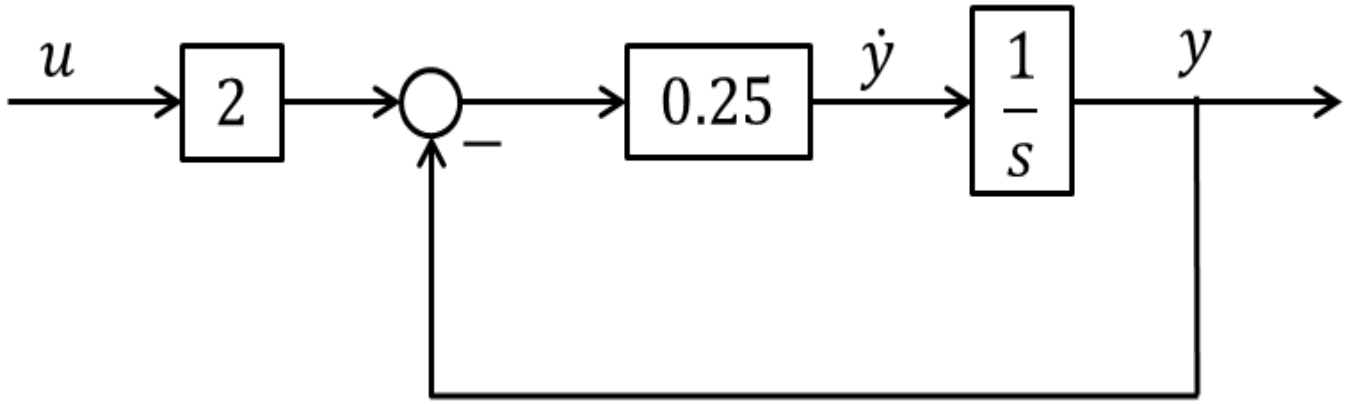


Example 2:

$$\dot{y} = -\frac{1}{4}y + \frac{1}{2}u$$

or:

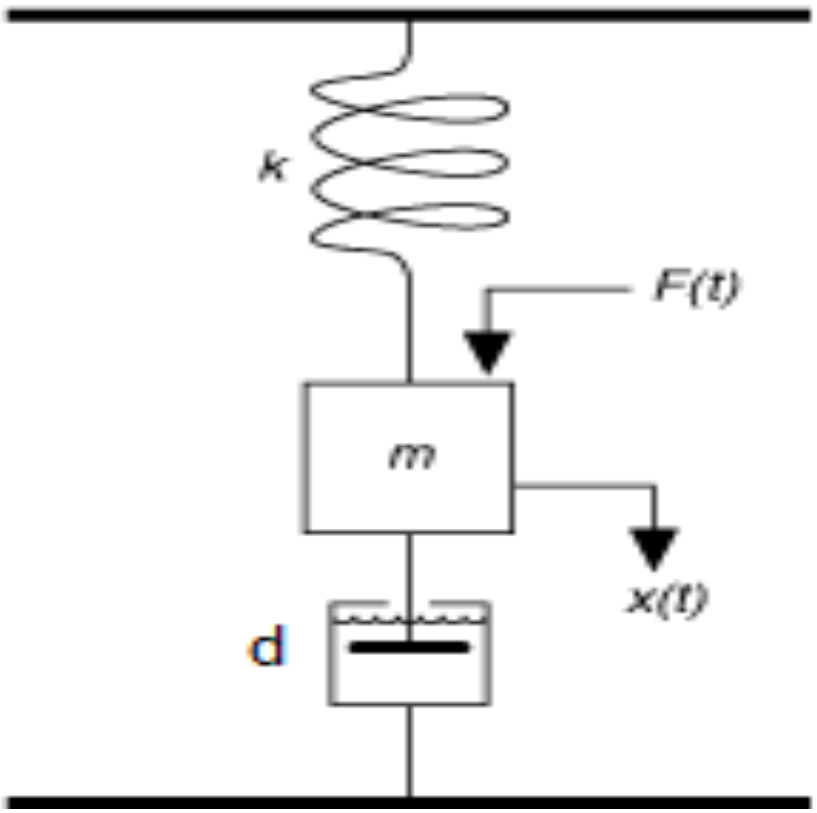
$$\dot{y} = \frac{1}{4}(-y + 2u)$$



Block Diagram Examples

Higher order differential equations

Mass-Spring-Damper Example:



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

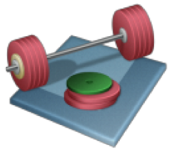
x is the position

$\dot{x}$ is the speed/velocity

$\ddot{x}$ is the acceleration

F is the Force (control signal, u)

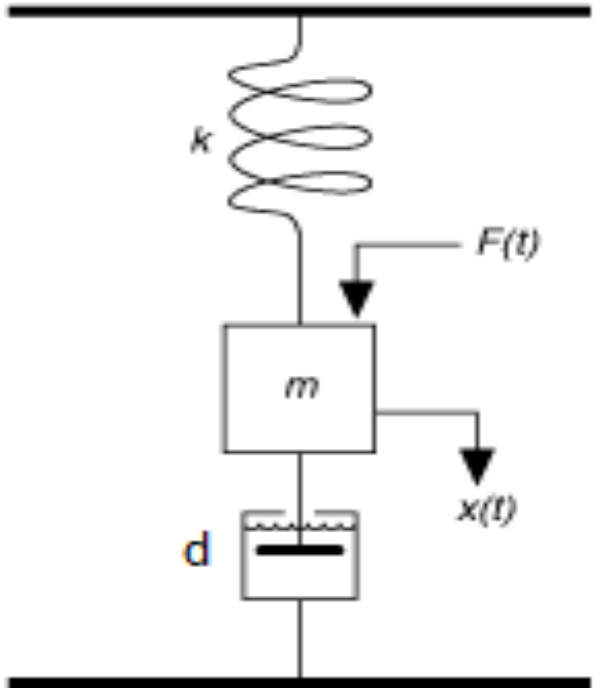
d and k are constants



Students: Find the block diagram for the differential equation above (pen & paper)

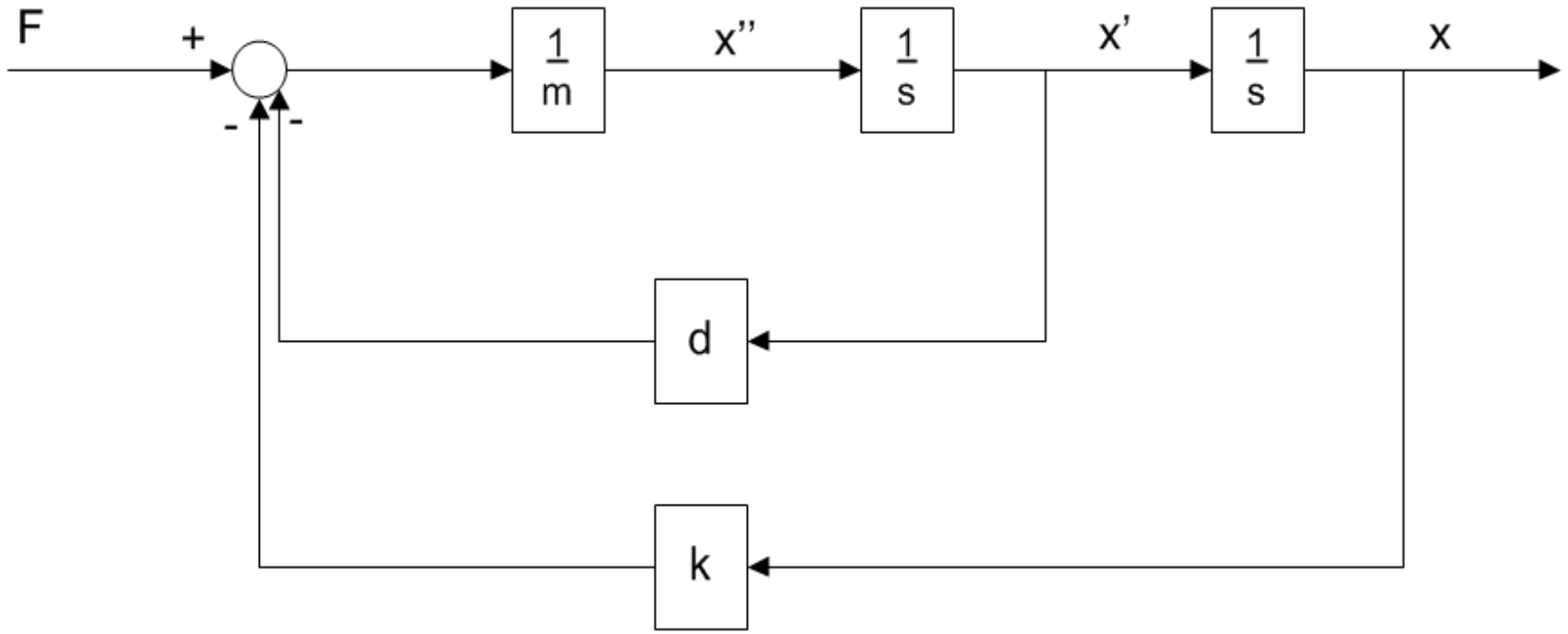
Block Diagram - Solutions

Mass-Spring-Damper Example:



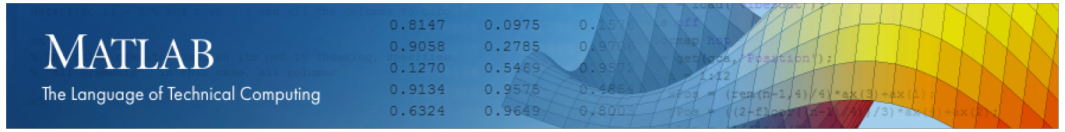
Higher order differential equations

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



Simulation Tools

- MATLAB
  - Text-based Programming Tool
  - www.mathworks.com



- Simulink
  - Block diagram-based Simulation, Integrated with MATLAB
- **LabVIEW**
- **MathScript**
  - Uses MATLAB syntax, Integrated with LabVIEW

- Modelica
  - https://www.modelica.org

- HYSYS
  - http://www.aspentech.com/products/aspen-hysys.aspx
- …

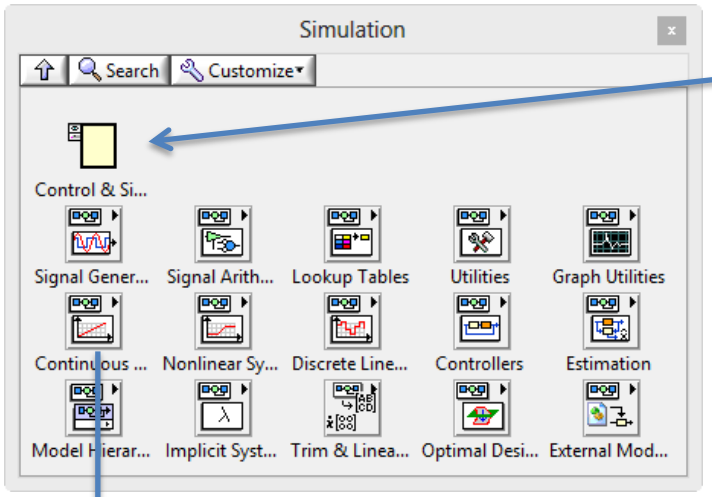
https://www.halvorsen.blog

Simulation in LabVIEW

Hans-Petter Halvorsen

Simulation in LabVIEW

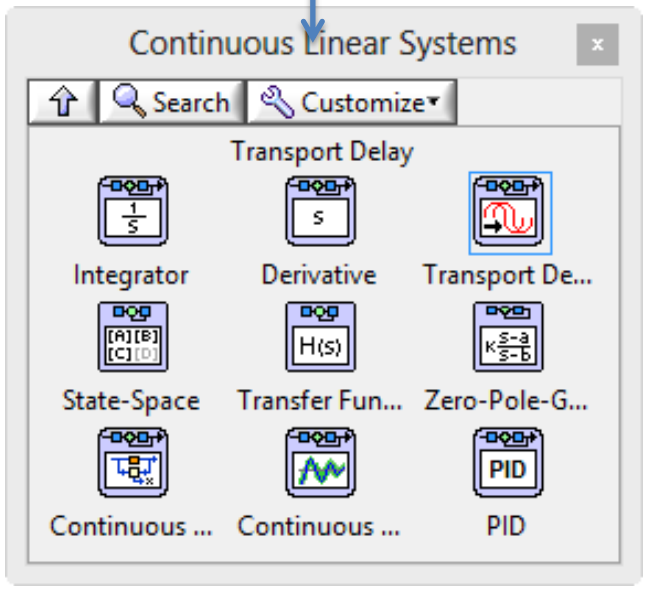
Simulation Palette in LabVIEW



Simulation Loop: Similar to a While Loop, but cusomized for used together with the Simulation Blocks available in LabVIEW

Different Simulation Blocks by Category

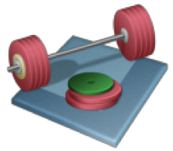
- Continuous Systems
- Discrete Systems
- Nonlinear Systems
- etc.



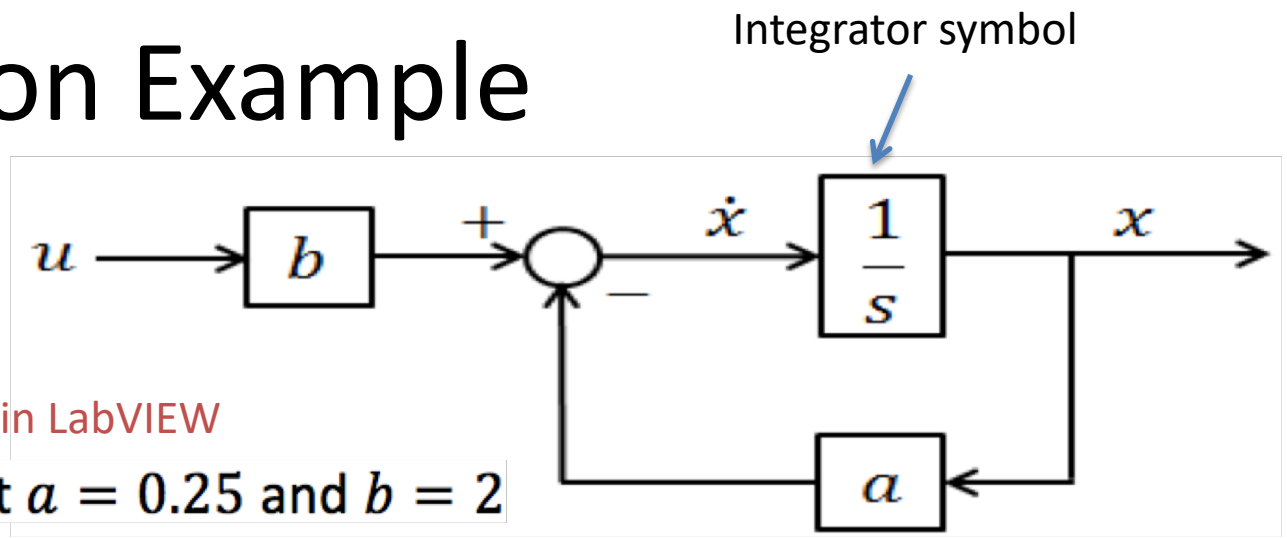
Simulation Example

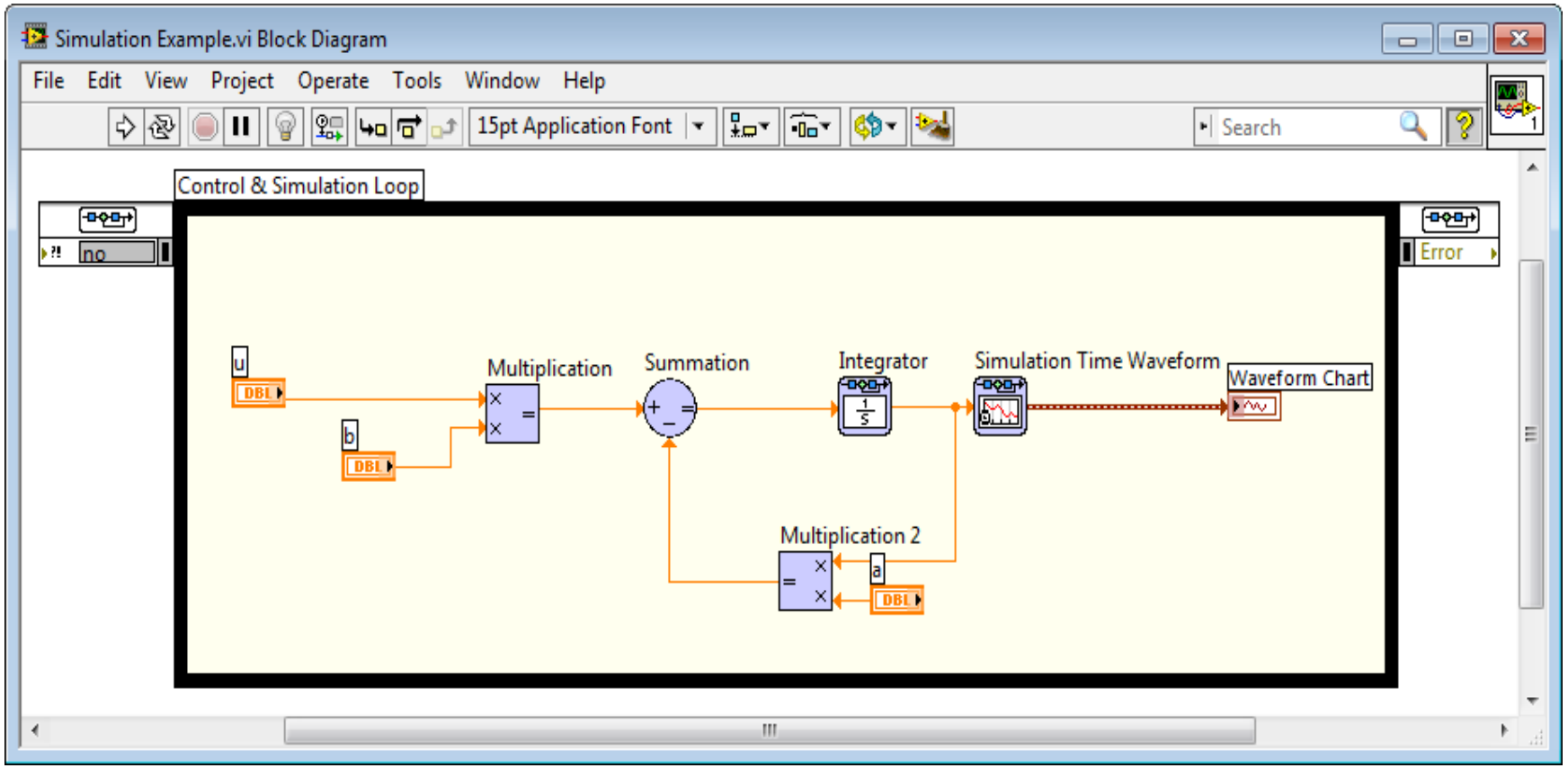
Dynamic system represented as a differential equation

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



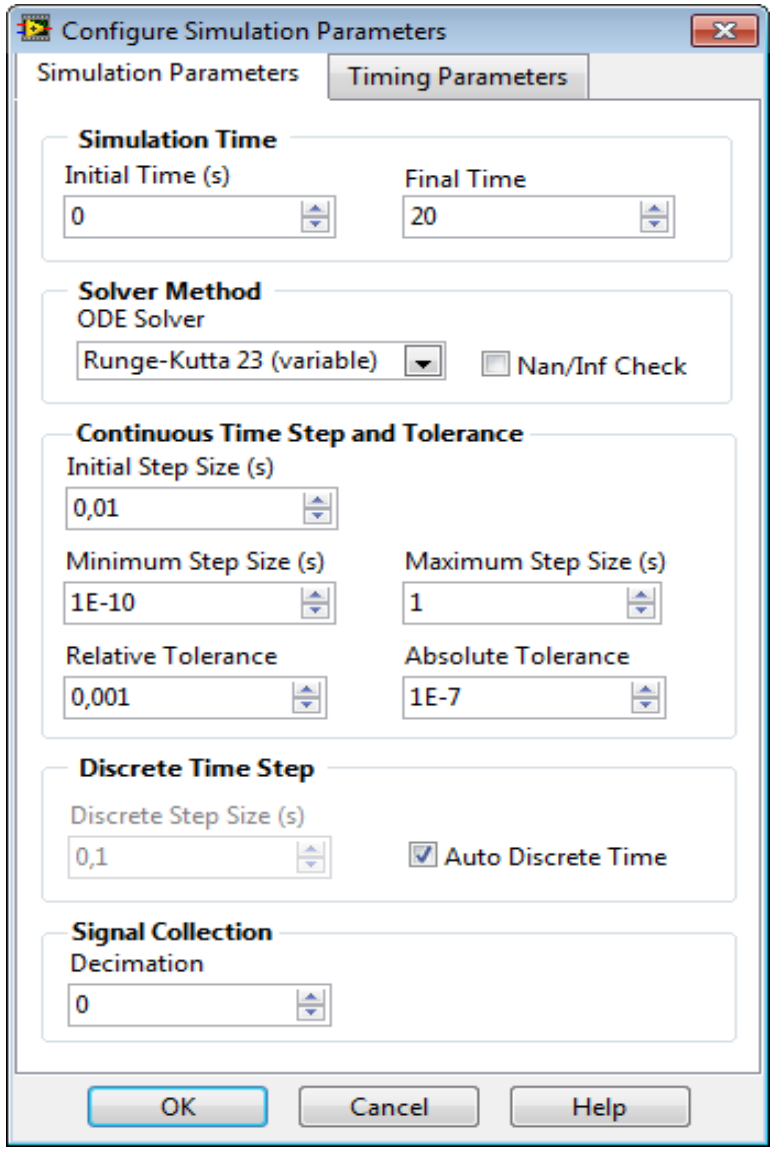
Students: Implement and Simulate this system in LabVIEW

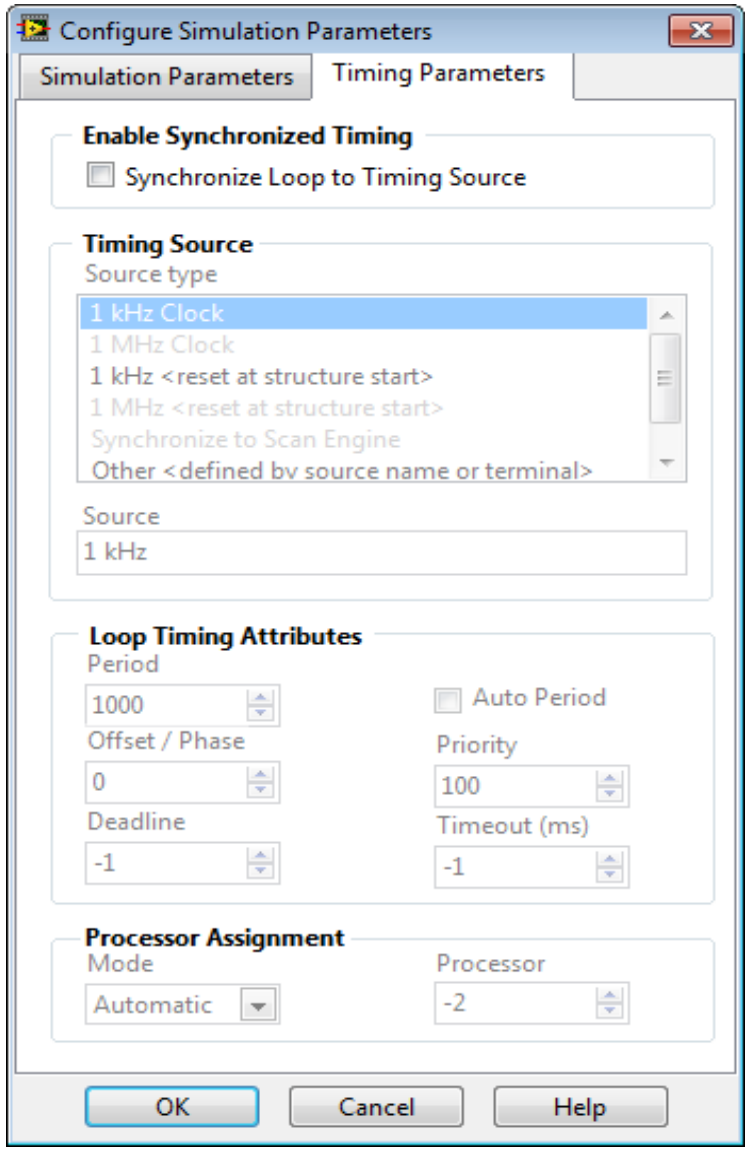




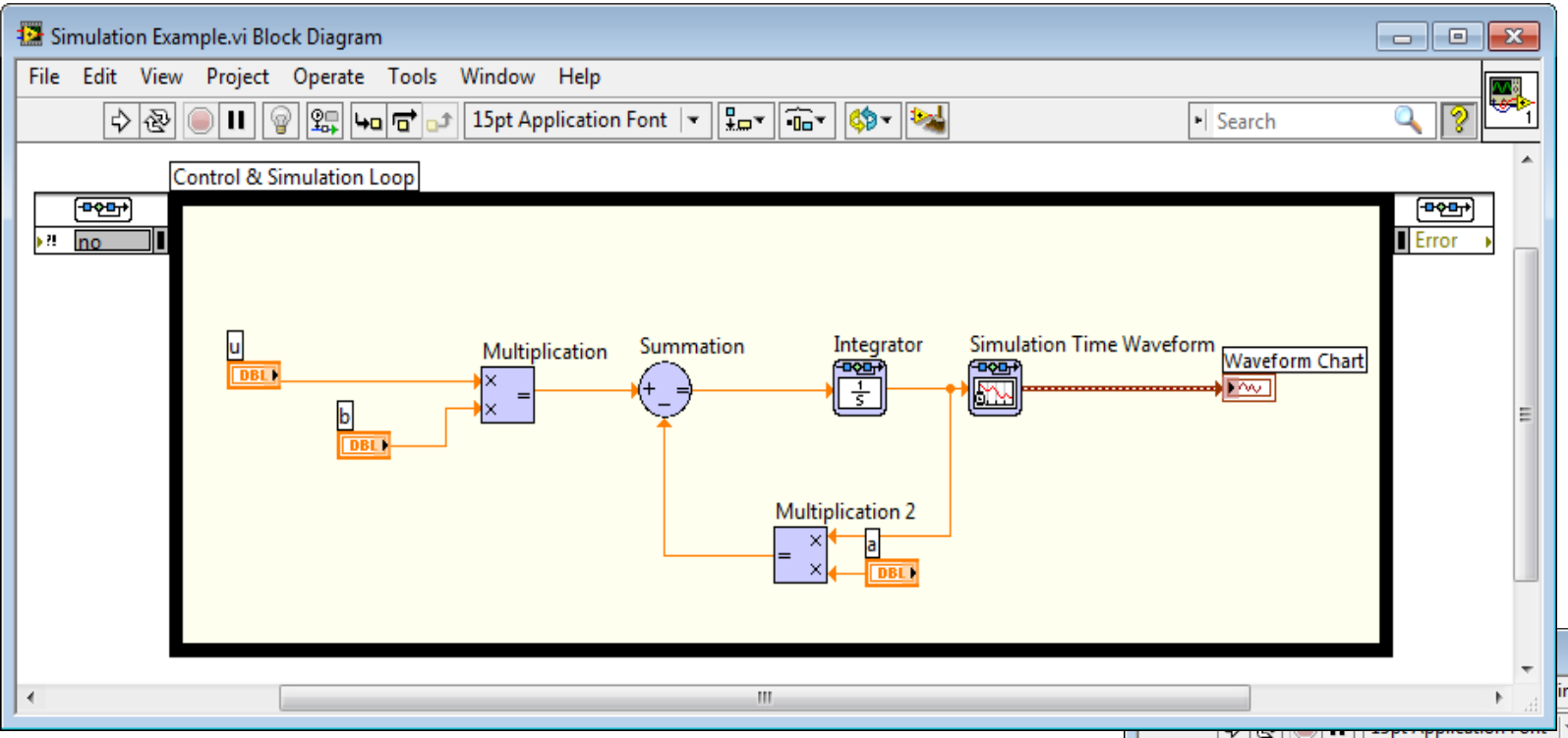
Simulation Example - Configuration

In the example the following simulation parameters could be used (right-click on the Simulation Loop border and select “Configure Simulation parameters…”):

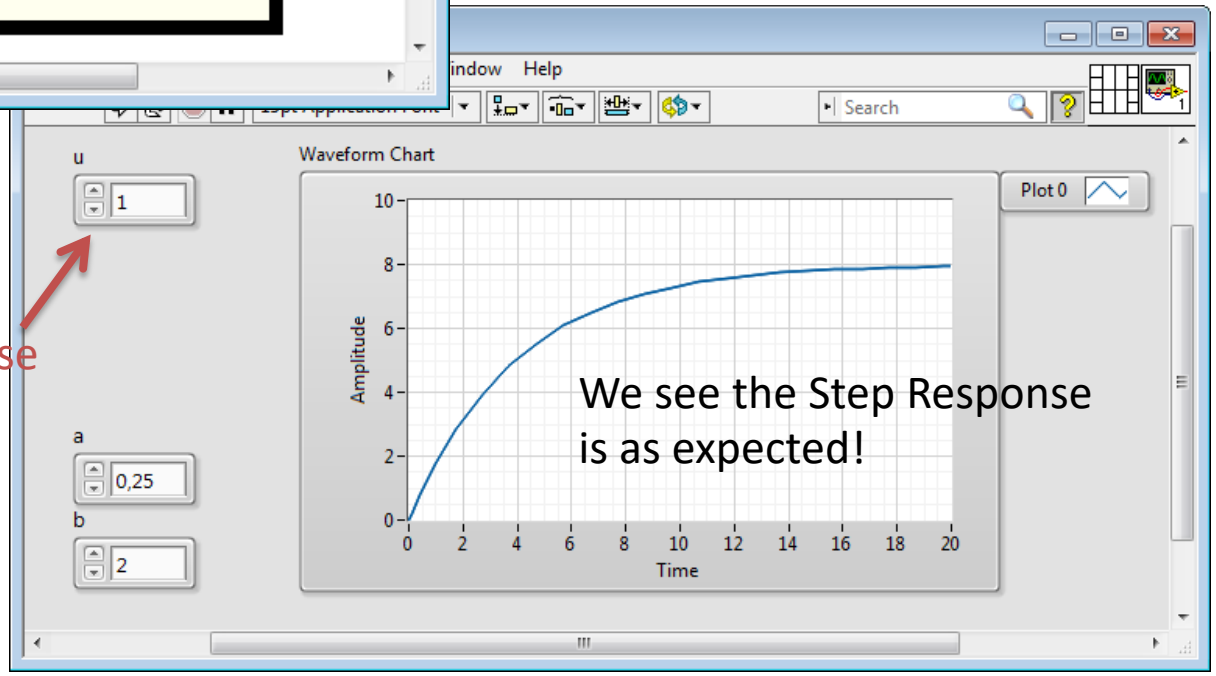




Simulation Example - Solutions



Try with different values for u



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

set $a = 0.25$ and $b = 2$

Step Response

Correct results? – Check static response:

$$0 = -ax_s + bu_s$$

$$x_s = \frac{b}{a}u_s = \frac{2}{0.25}1 = 8$$

https://www.halvorsen.blog

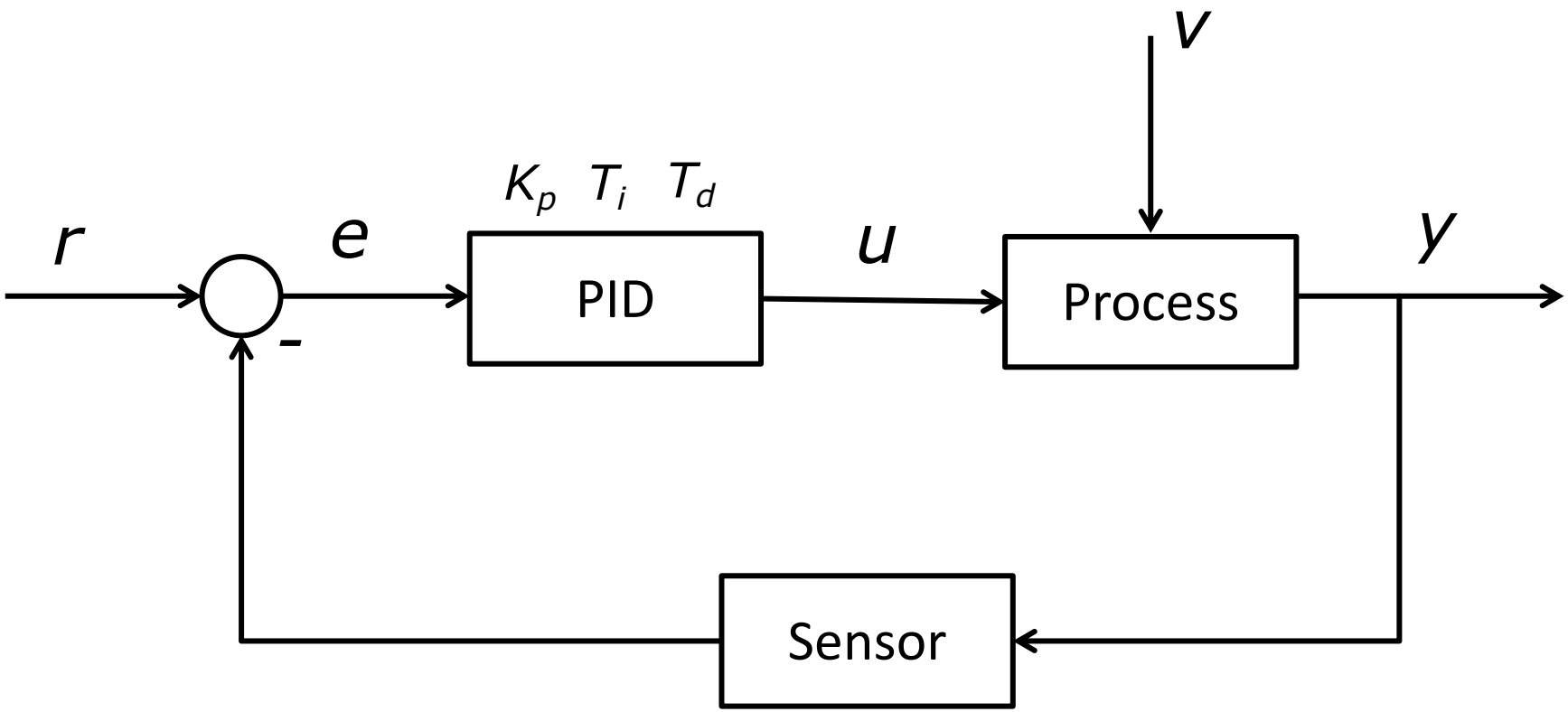
PID Control in LabVIEW

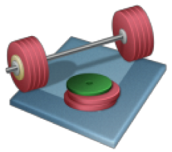
Hans-Petter Halvorsen

Control Systems

Example of Industrial Controllers

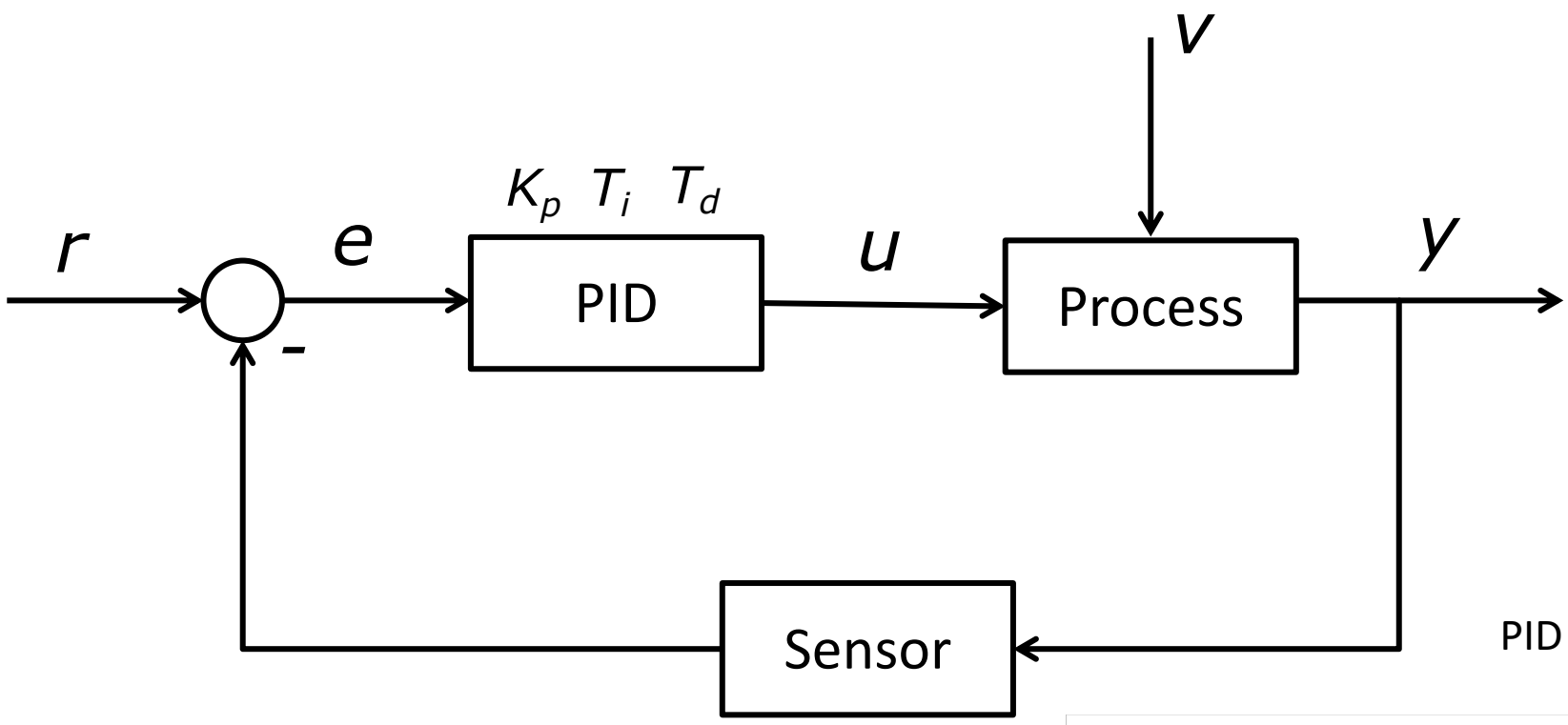
Typical Block Diagram:





Students: What is *r, e, u, v, y, Kp, Ti, Td*?

Control System

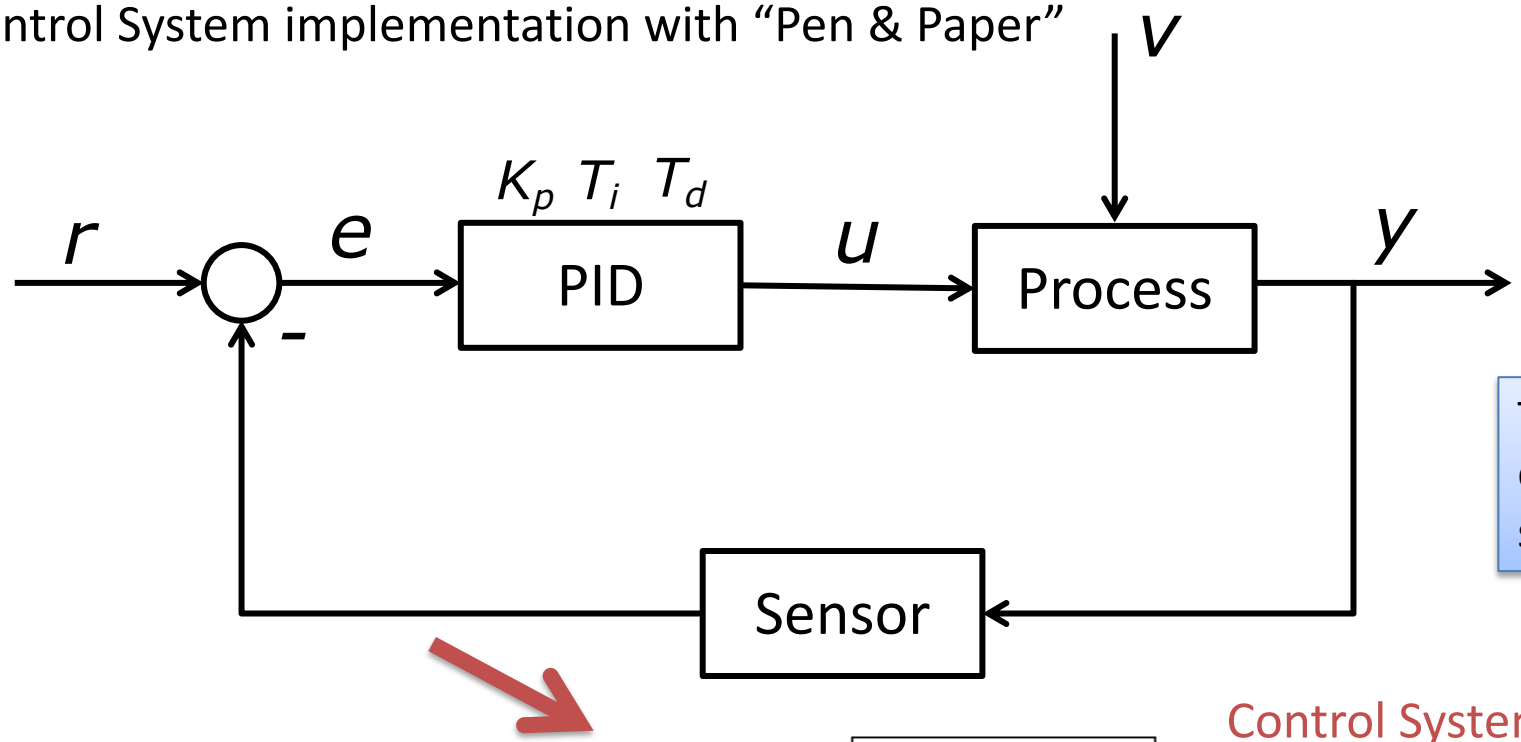


PID Algorithm:

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

r – Reference Value, SP (Setpoint), SV (Set Value)
y – Measurement Value (MV), Process Value (PV)
e – Error between the reference value and the measurement value (e = r – y)
v – Disturbance, makes it more complicated to control the process
Kp, Ti, Td – PID parameters

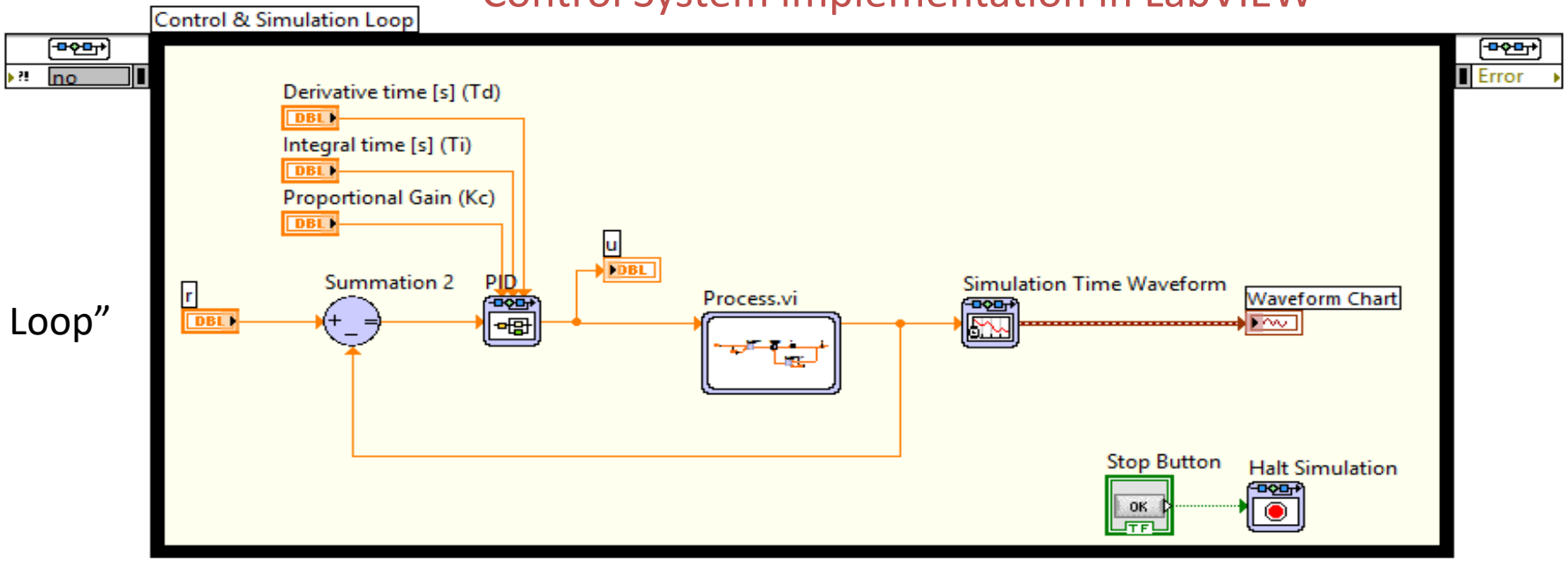
Control System implementation with “Pen & Paper”



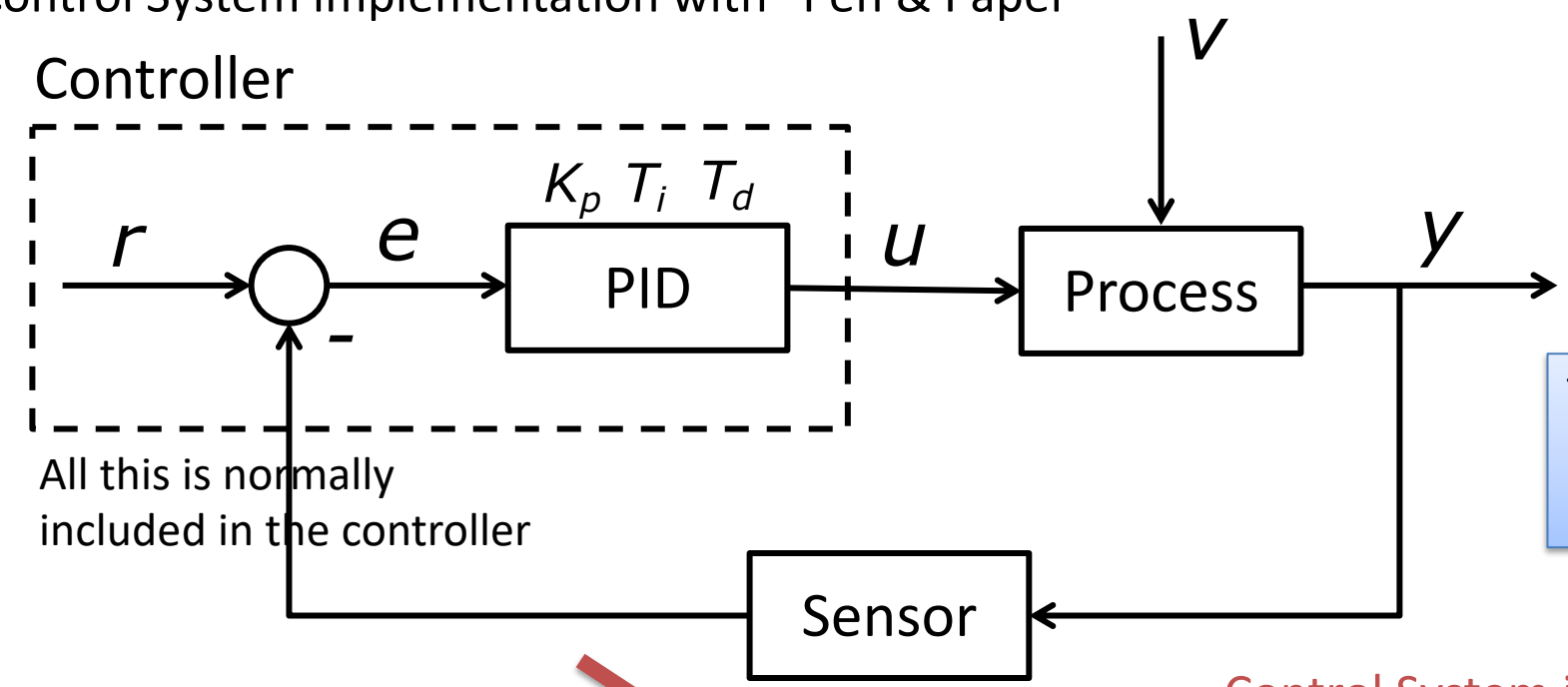
The transition from “paper” to LabVIEW is easy, because the implementation is very similar to the “paper” version

Here we have used the “Simulation Loop”

Control System implementation in LabVIEW



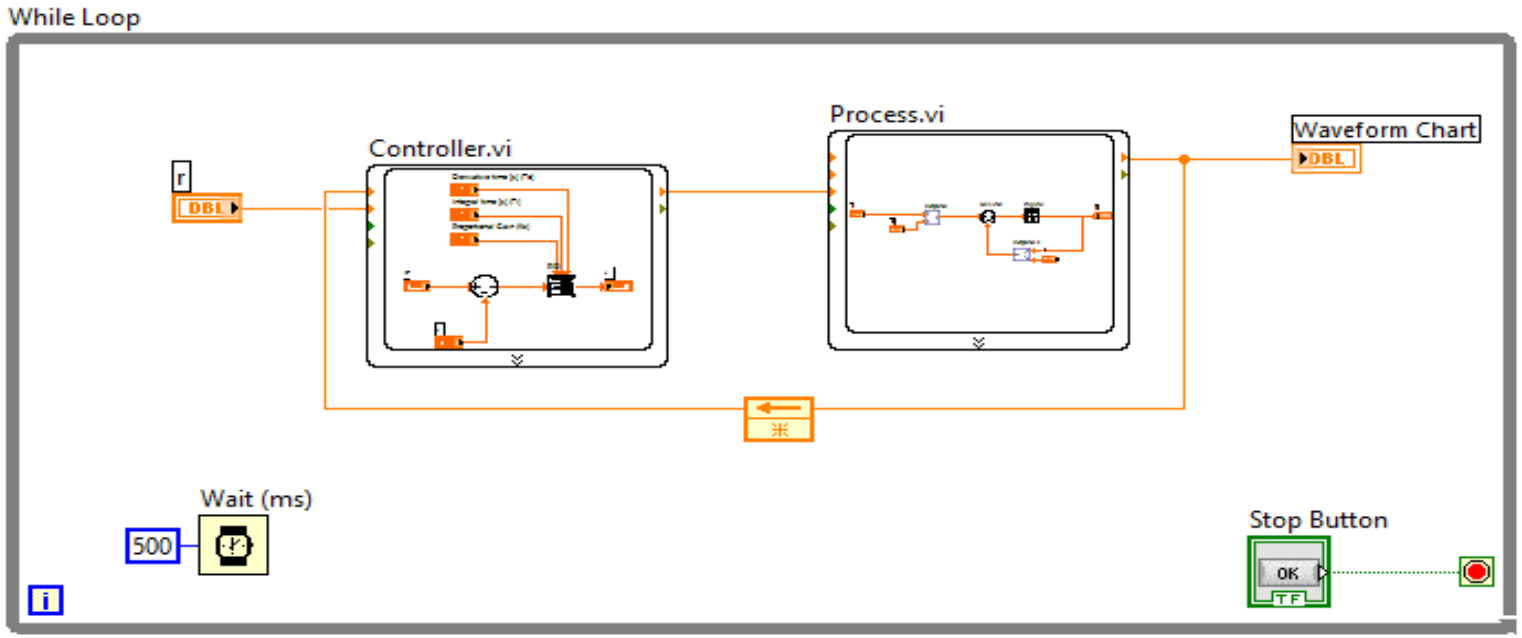
Control System implementation with “Pen & Paper”



All this is normally
included in the controller

The transition from “paper” to LabVIEW is easy, because the implementation is very similar to the “paper” version

Control System implementation in LabVIEW

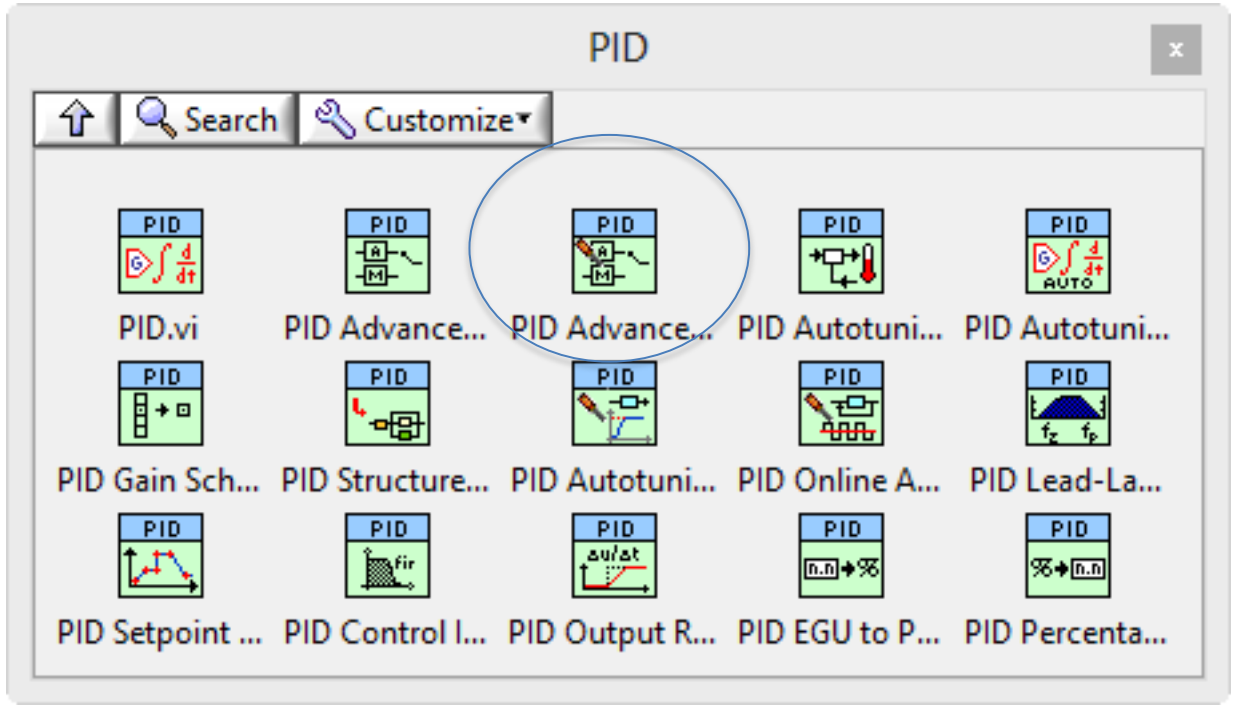


Here we have used an ordinary While Loop
(which is recommended!)

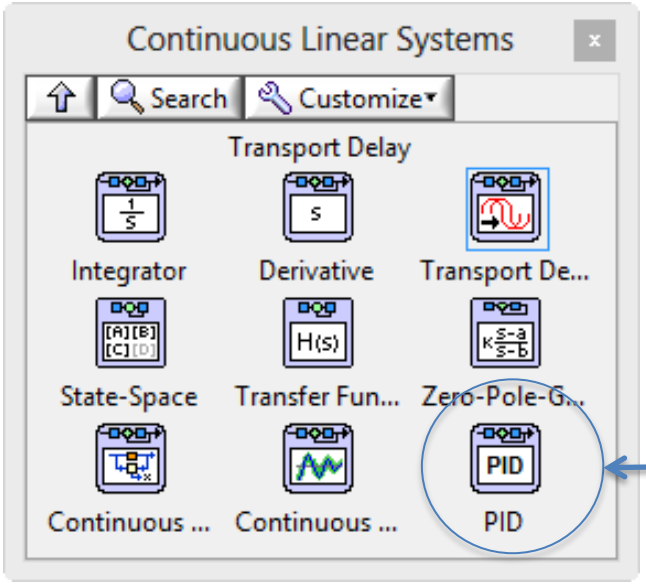
PID Control in LabVIEW

Alternative 1:

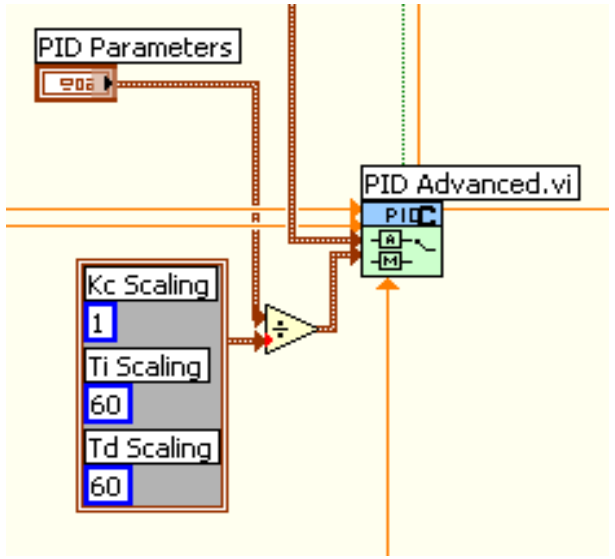
PID Palette in LabVIEW (PID Toolkit)



Alternative 2:



This alternative uses seconds!



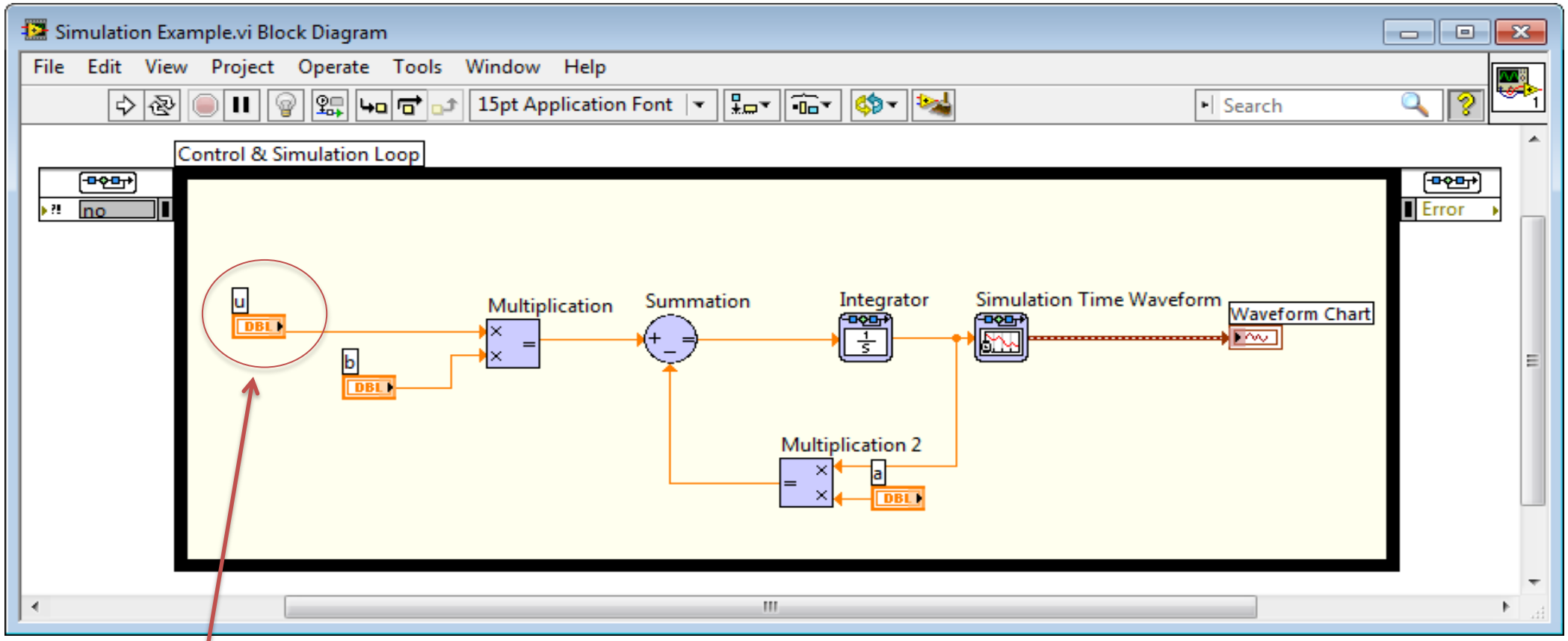
Note! The functions “PID.vi” and “PID Advanced.vi” requires that Ti and Td are in minutes, while it’s normal to use seconds as the unit for these parameters. You can use the following piece of code in order to transform them:

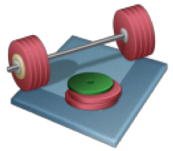
This means we enter values for Ti and Td in seconds on the Front Panel and the values are converted to minutes in the code.

LabVIEW PID Example

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

set $a = 0.25$ and $b = 2$





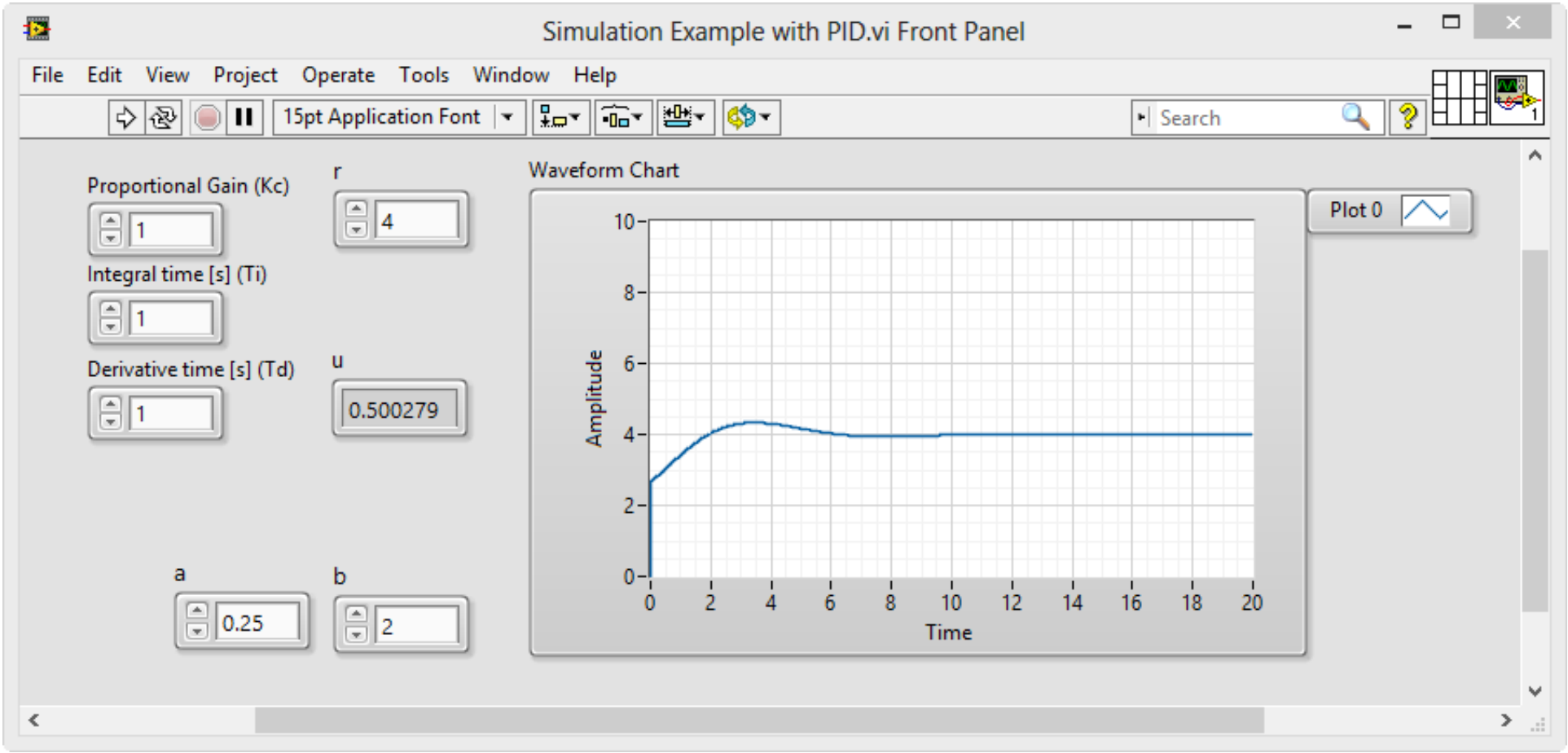
Students: Replace u in the previous example with the built-in PID Controller (use Alternative 2)

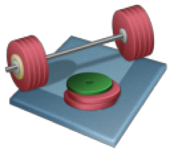
PID Example - Solutions

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

set $a = 0.25$ and $b = 2$

Front Panel:



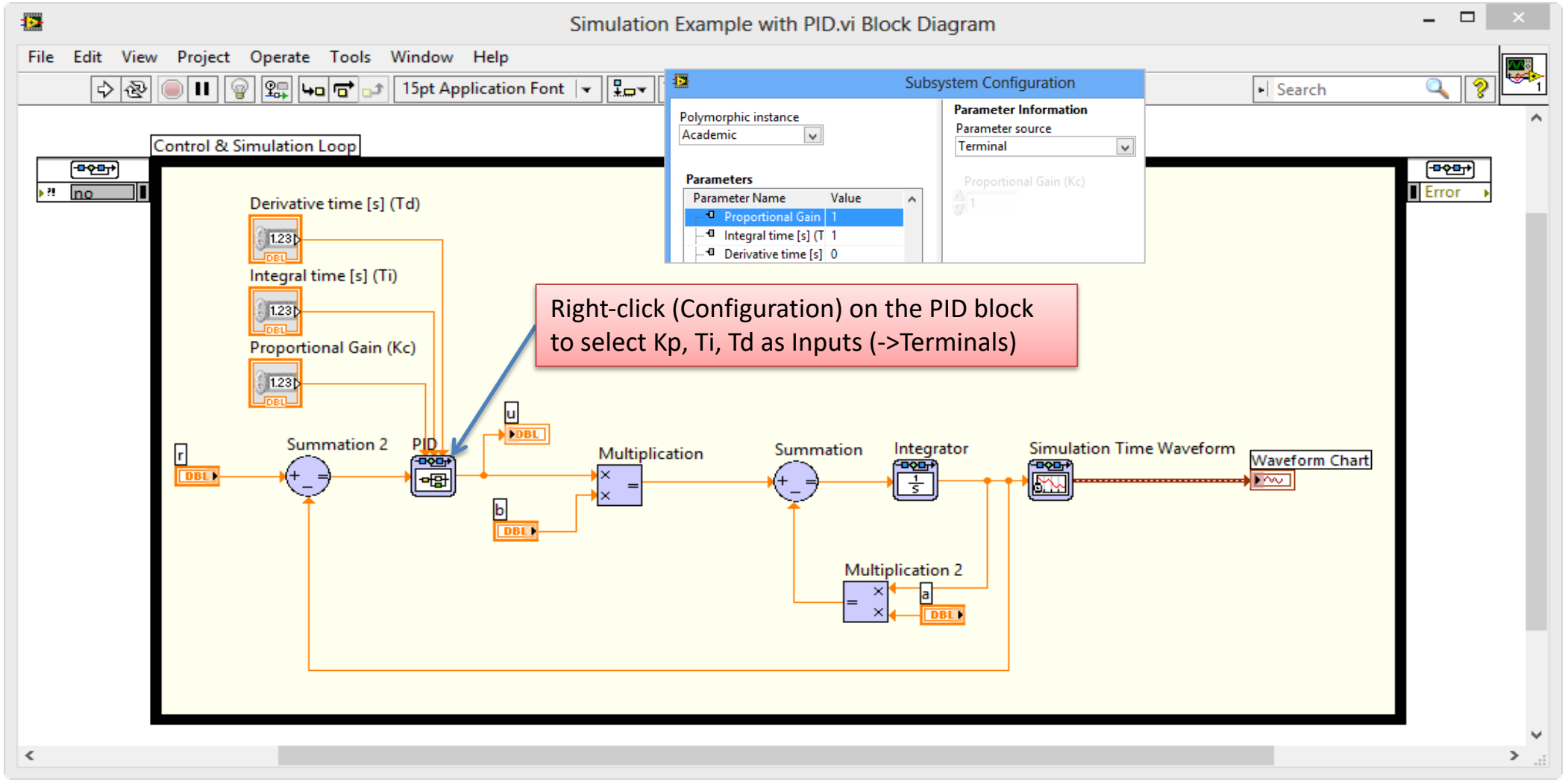


Students: Find proper K_p, T_i, T_d Parameters for this system. Use "Trial and Error" (or a more systematic approach)

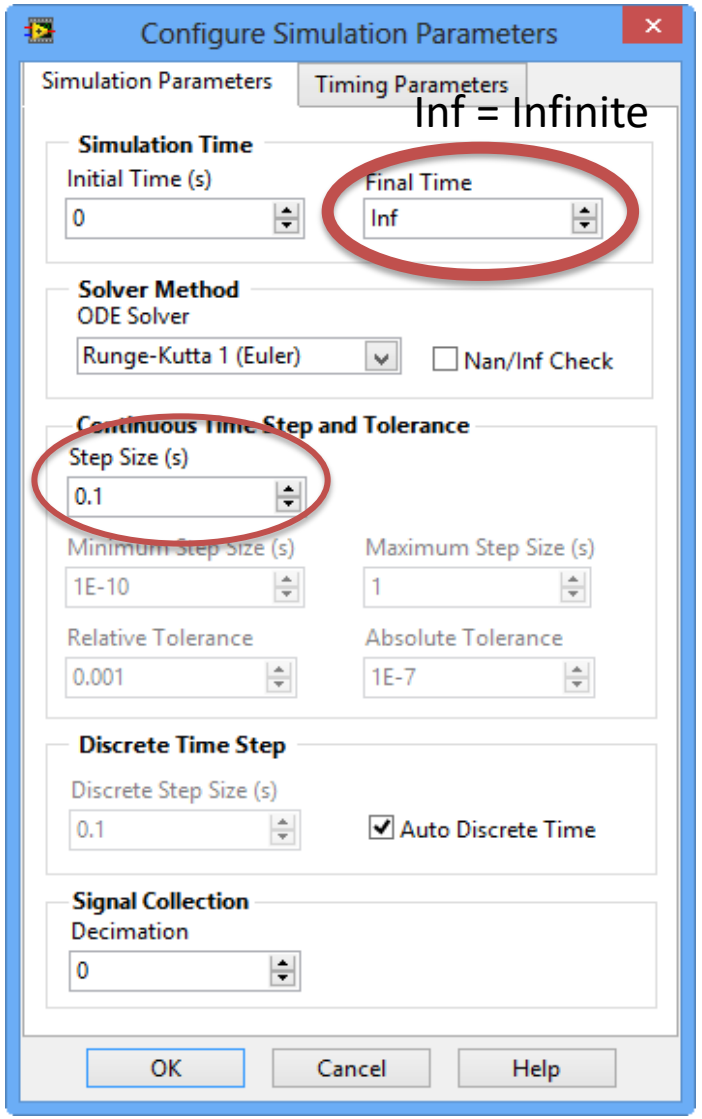
PID Example - Solutions

$$\dot{x} = -ax + bu$$ set $a = 0.25$ and $b = 2$

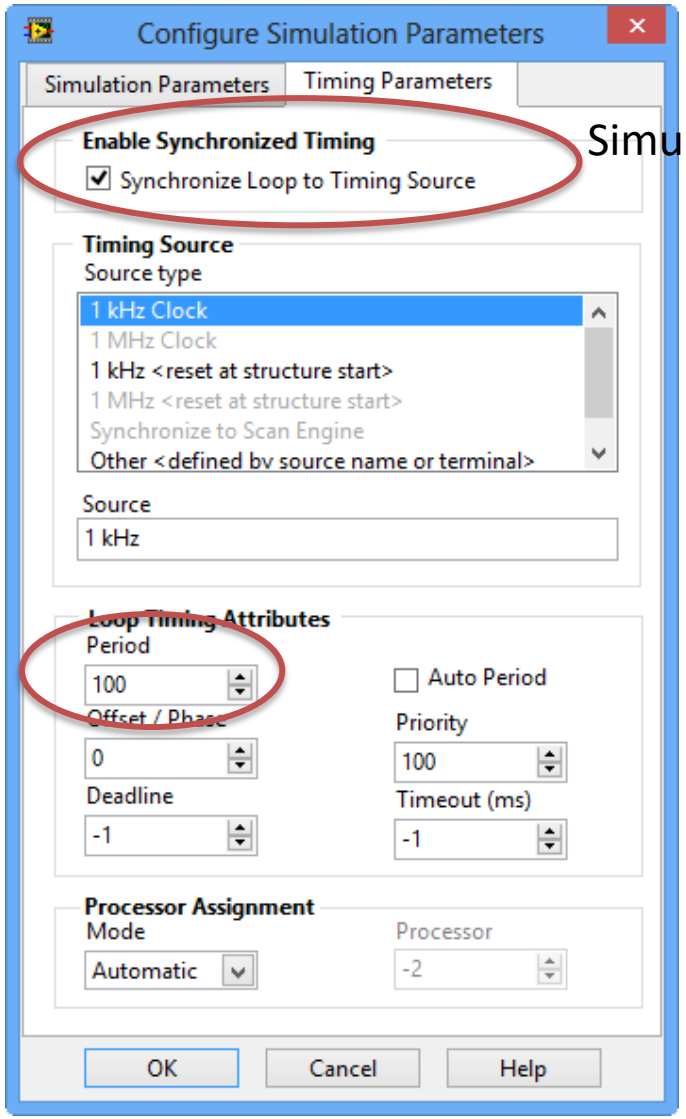
Block Diagram:



Next Step: Continuous Simulation



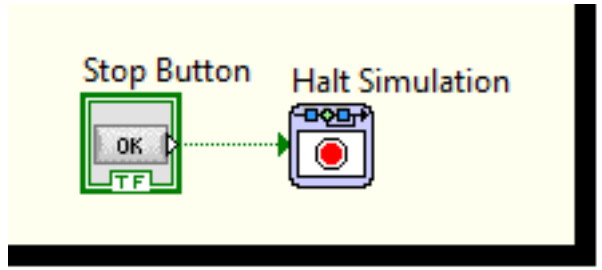
Inf = Infinite

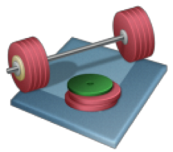


Simulation in “Real Time”

Right-click on the Simulation Loop border and select “Configure Simulation Parameters…”

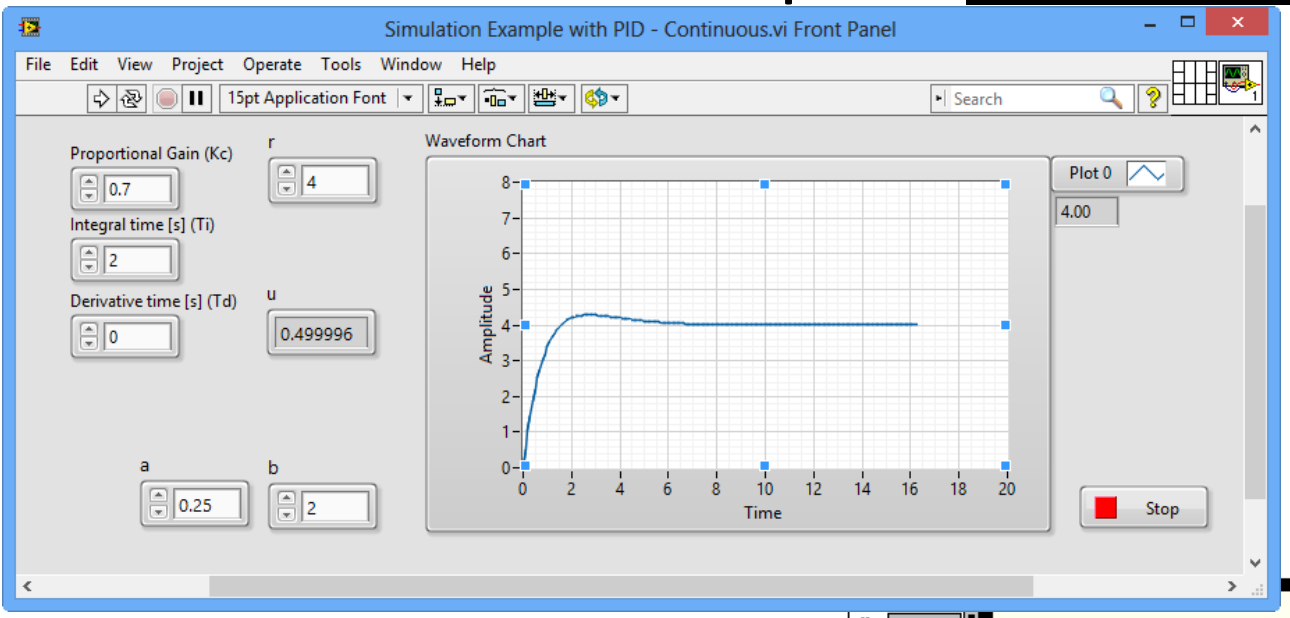
Add a Stop Button
and a “Halt Simulation” block

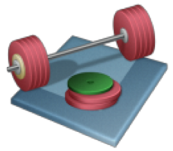




Students: Change your Simulation Settings and Run your Simulation with these changes

PID Example – Continuous Simulation - Solution

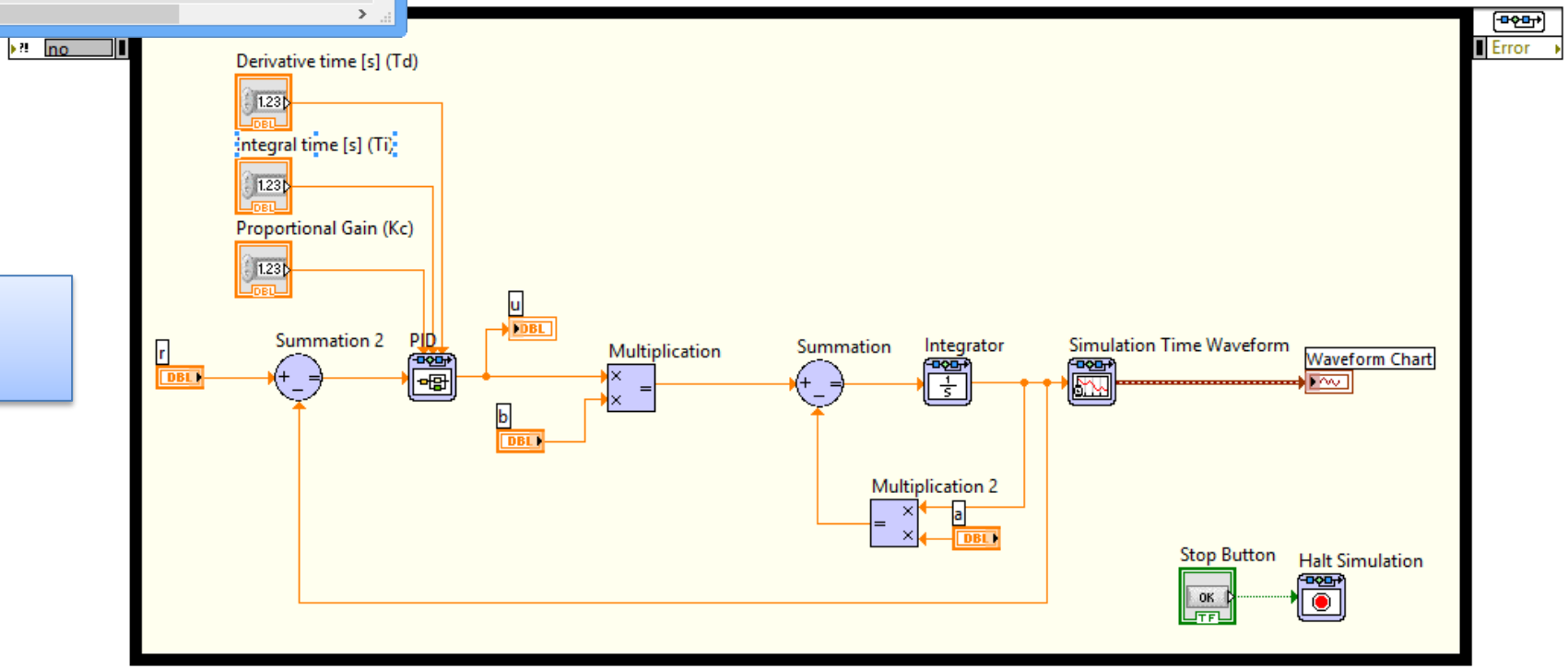




Students:

Extend the example with a new Chart of the Control Signal (u) and include the reference signal (r) in the existing Chart, so you can easily see the changes in r and how the system handle it.

The Simulation now runs until you press the Stop button

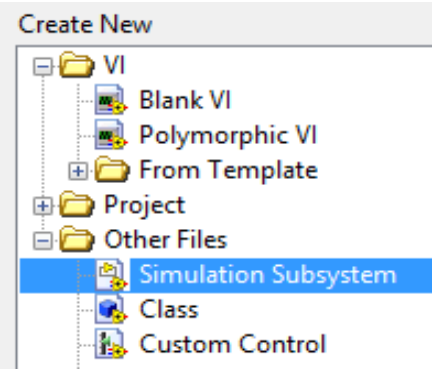


https://www.halvorsen.blog

Simulation Subsystems

Hans-Petter Halvorsen

Simulation Subsystem

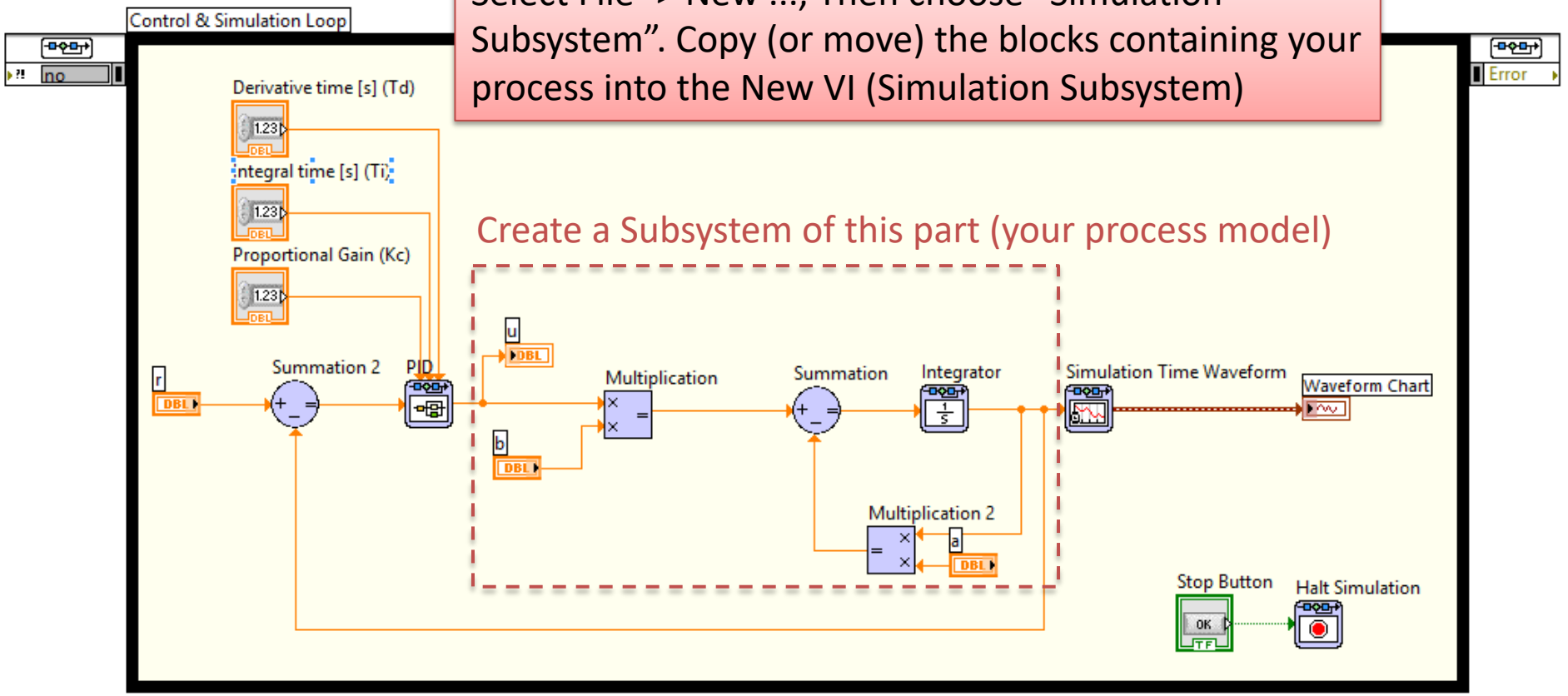


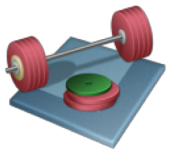
File -> New…

A Way to structure your code, similar to SubVIs

This is the recommended way to do it! – You can easly reuse your Subsystems in different VIs and your code becomes more structured!

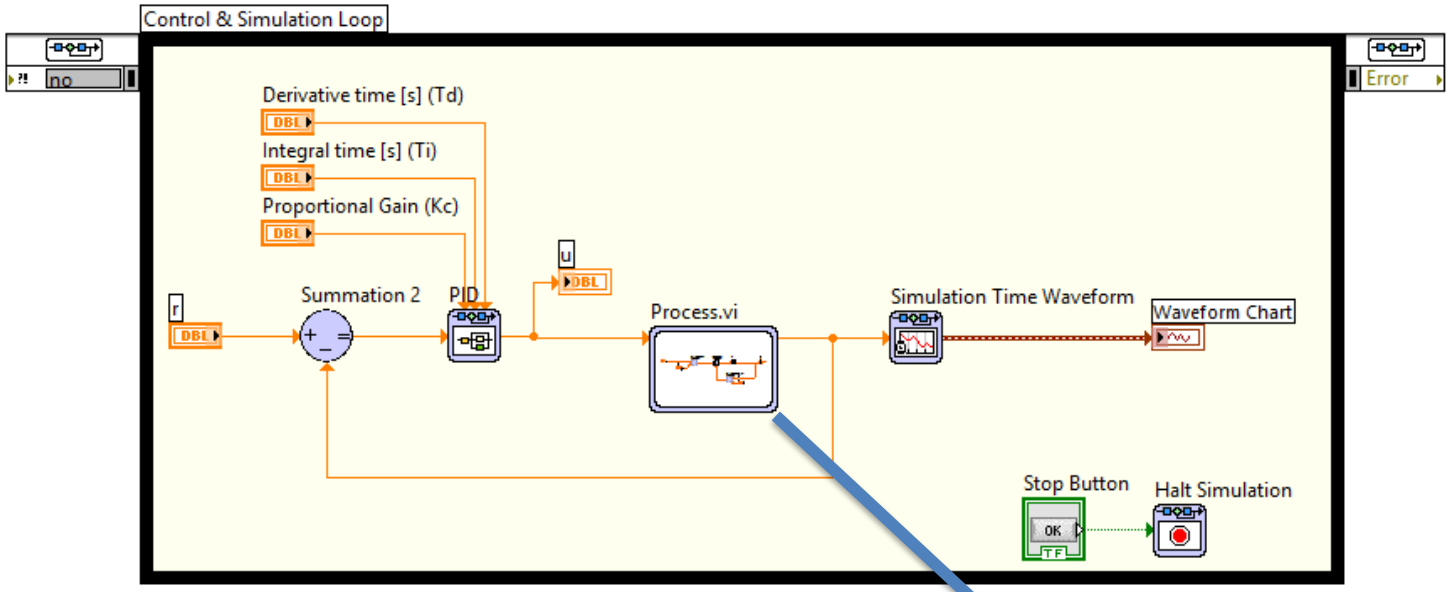
Select File -> New ..., Then choose “Simulation Subsystem”. Copy (or move) the blocks containing your process into the New VI (Simulation Subsystem)



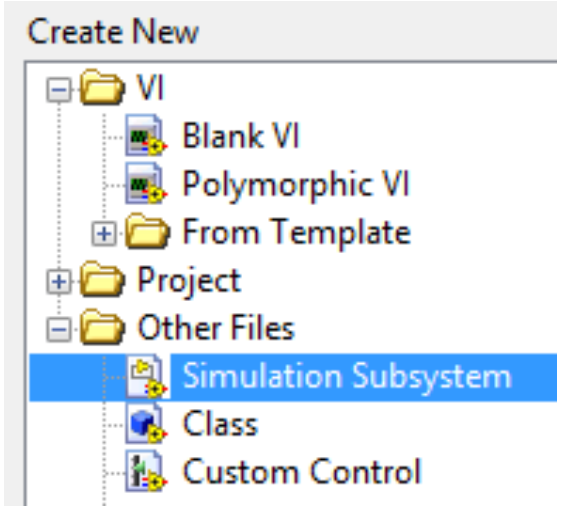


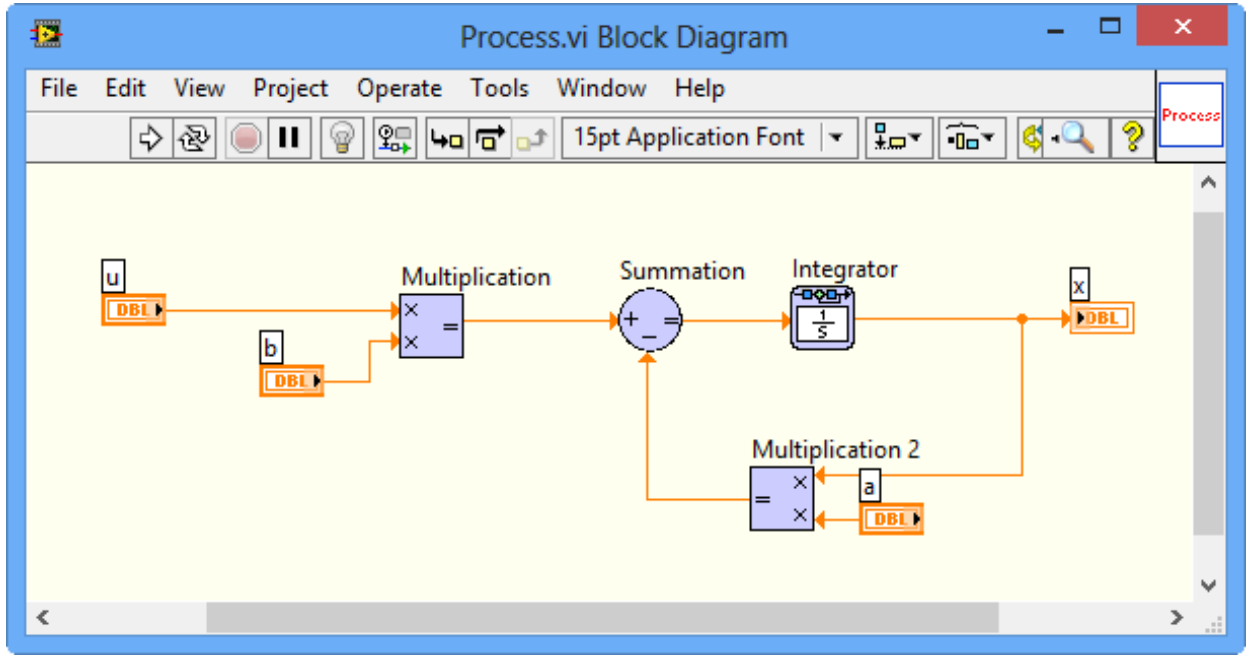
Students: Change your code above where you create a Simulation Sub System for your Process

Simulation Subsystem - Solutions

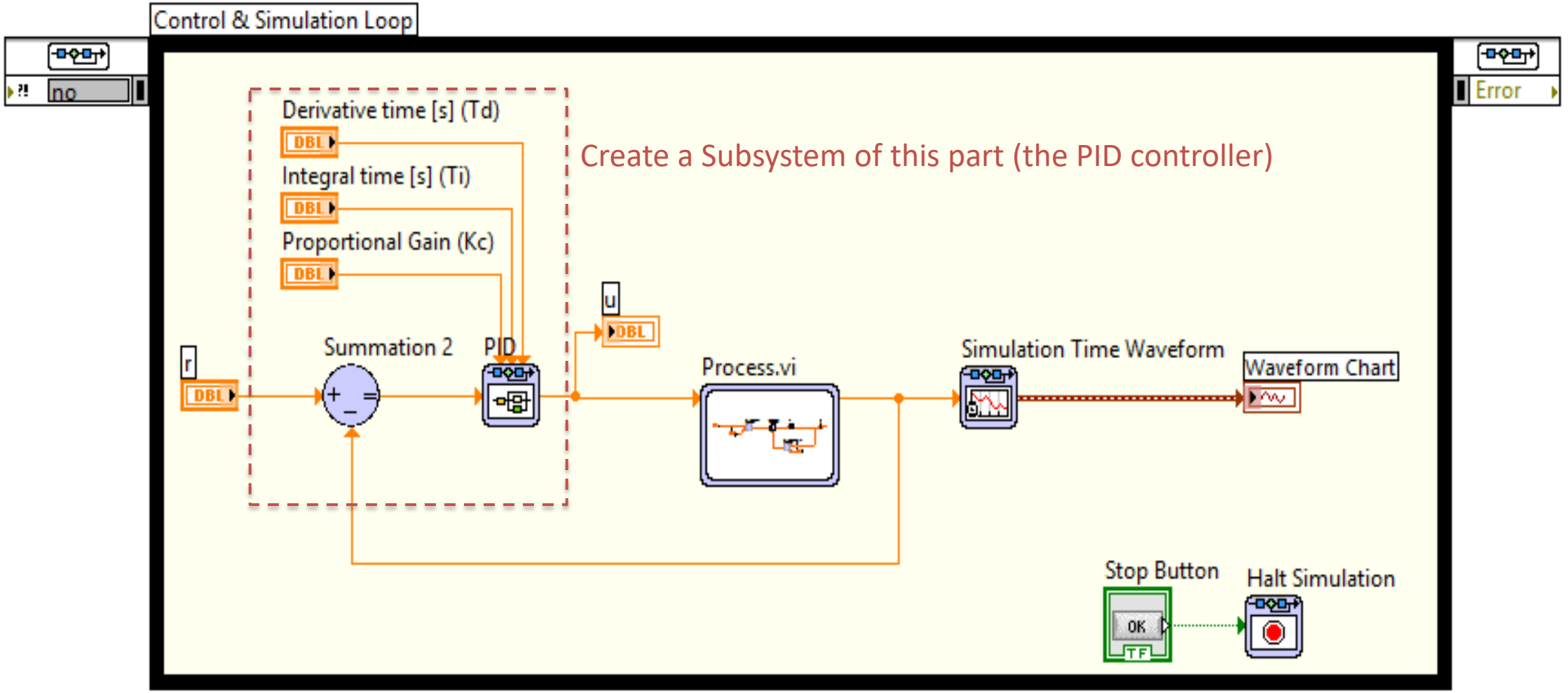


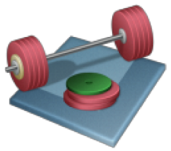
File -> New…





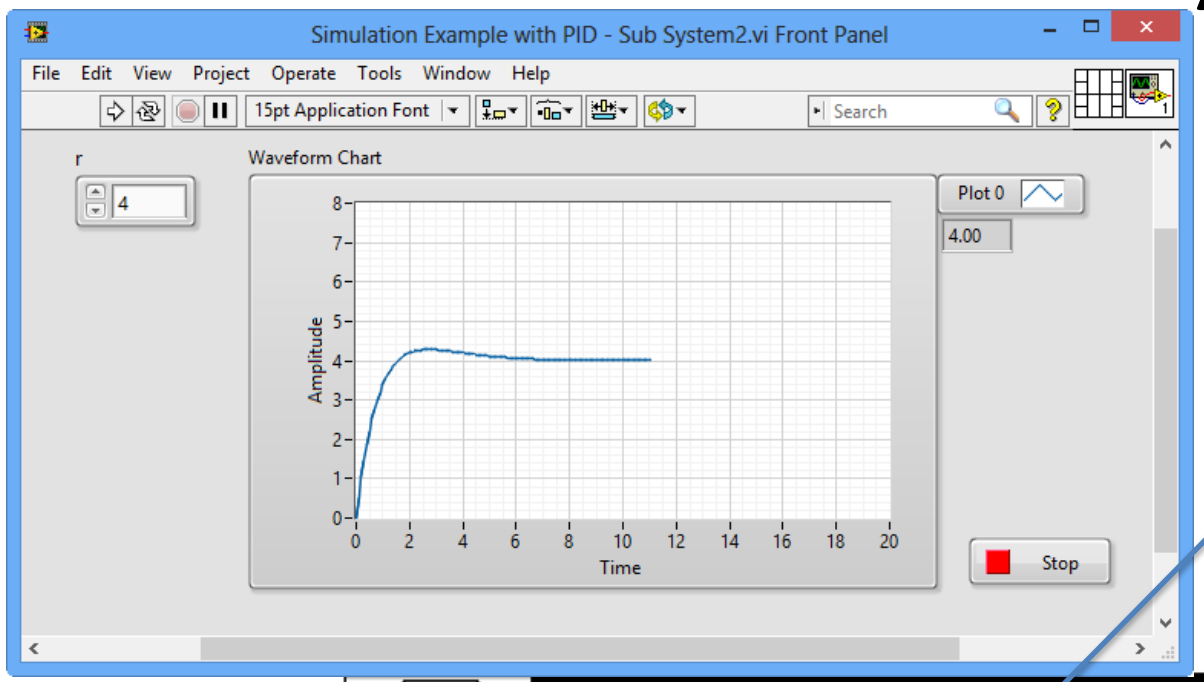
Simulation Subsystem 2 (PID Controller)

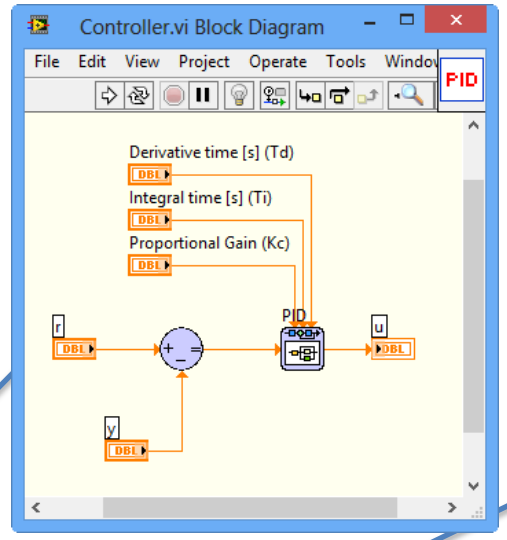


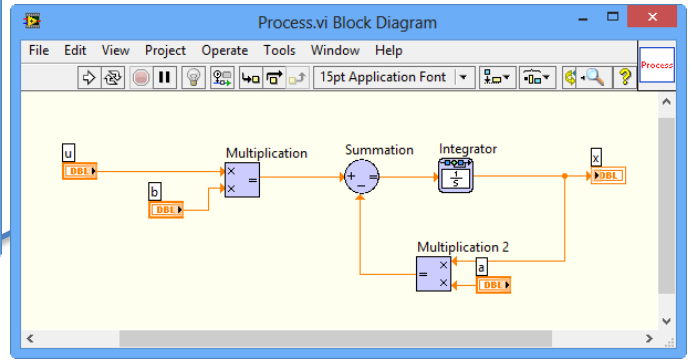


Students: Change your code above where you create a Simulation Sub System for the PID Controller as well.

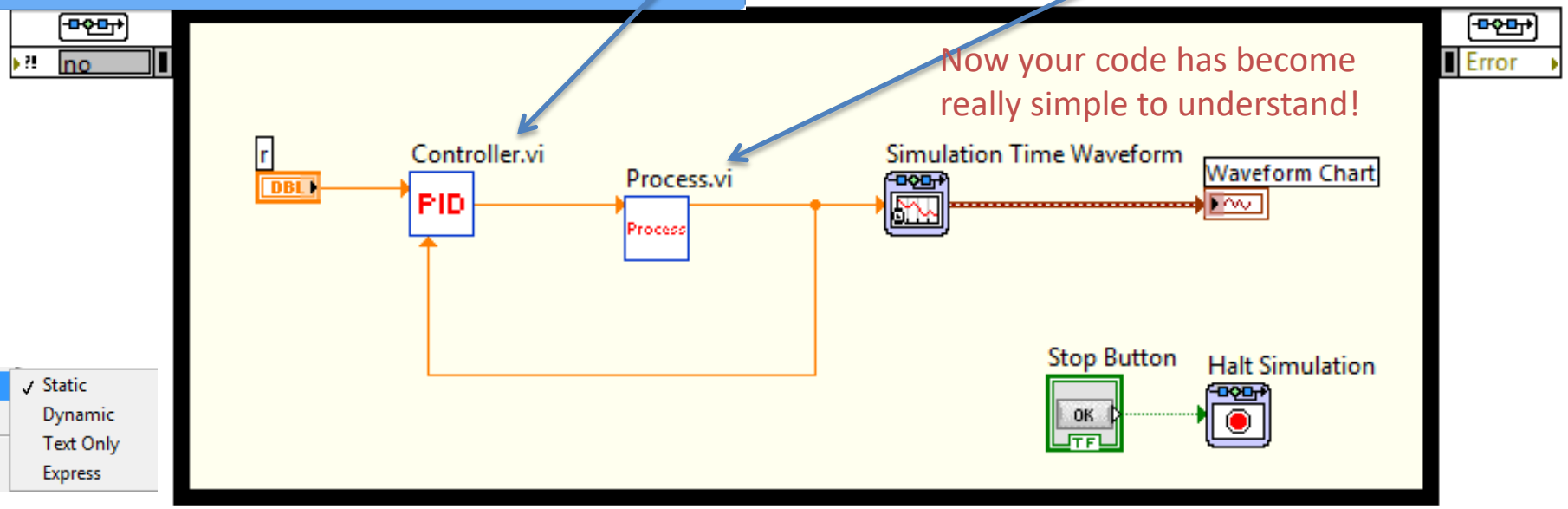
Simulation Subsystem – Solutions2



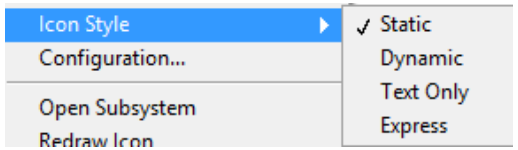




Simulation Sub Systems

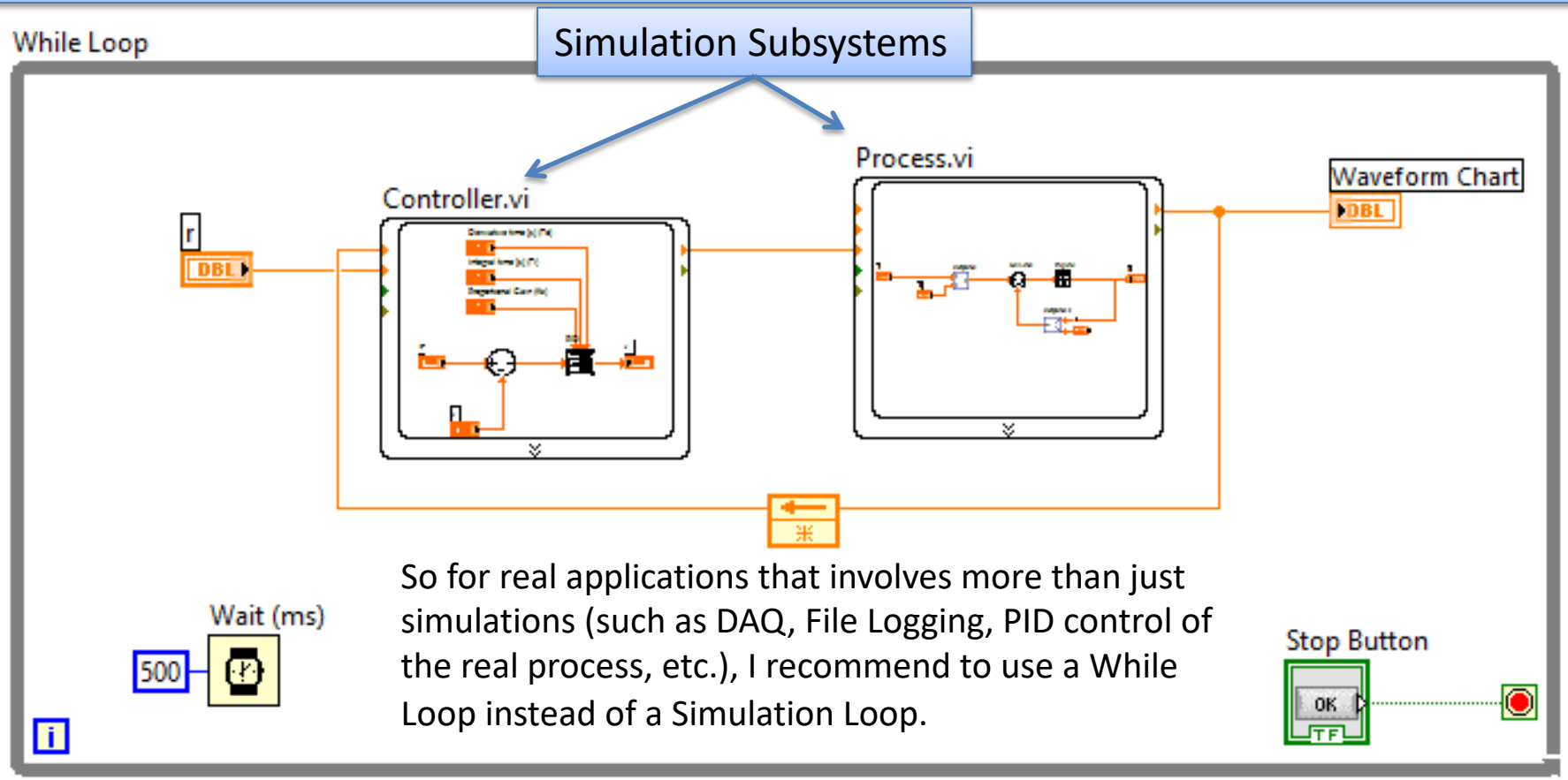


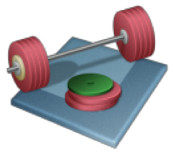
Note! You may select
different icon style



Simulations using a While Loop

Note! The Simulation Loop has some drawbacks/is more complicated to use than an ordinary While Loop. If we use Simulation Subsystems, we can use them inside a While Loop instead! - which becomes very handy!





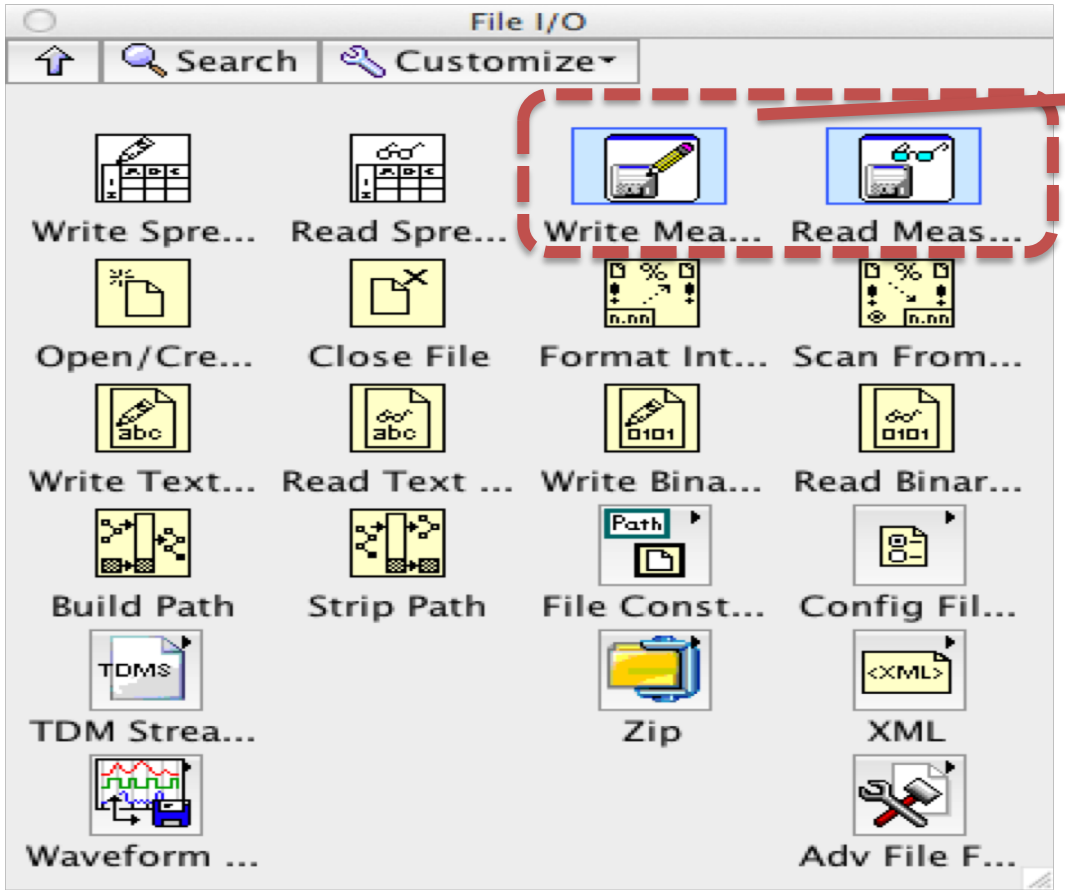
Students: Add your Controller and Process Subsystems inside a While loop as shown above. Simulate the system. Do you get the same results?

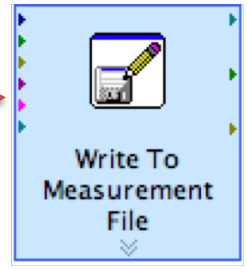
https://www.halvorsen.blog

Datalogging with LabVIEW

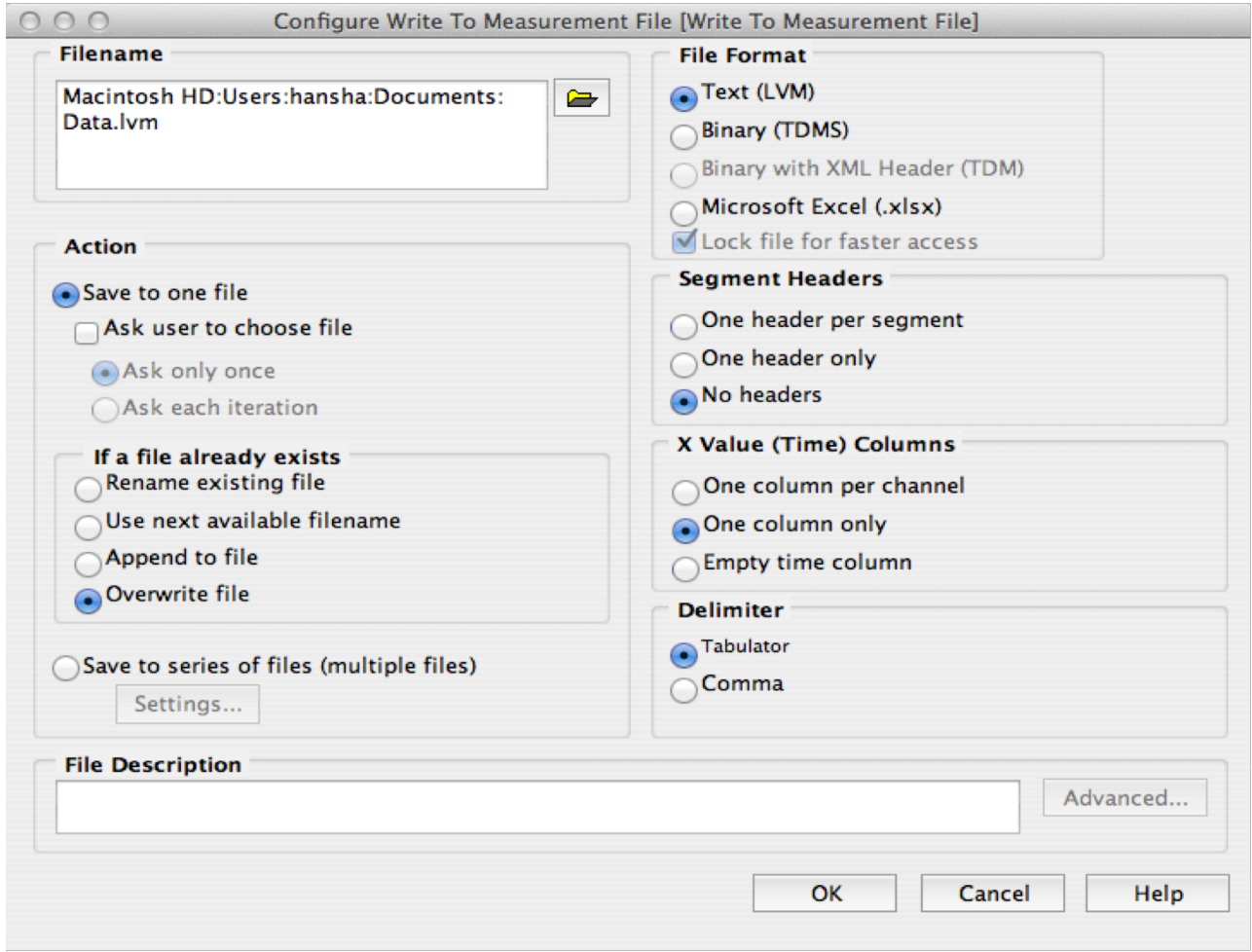
Hans-Petter Halvorsen

Save Data to File (Datalogging)



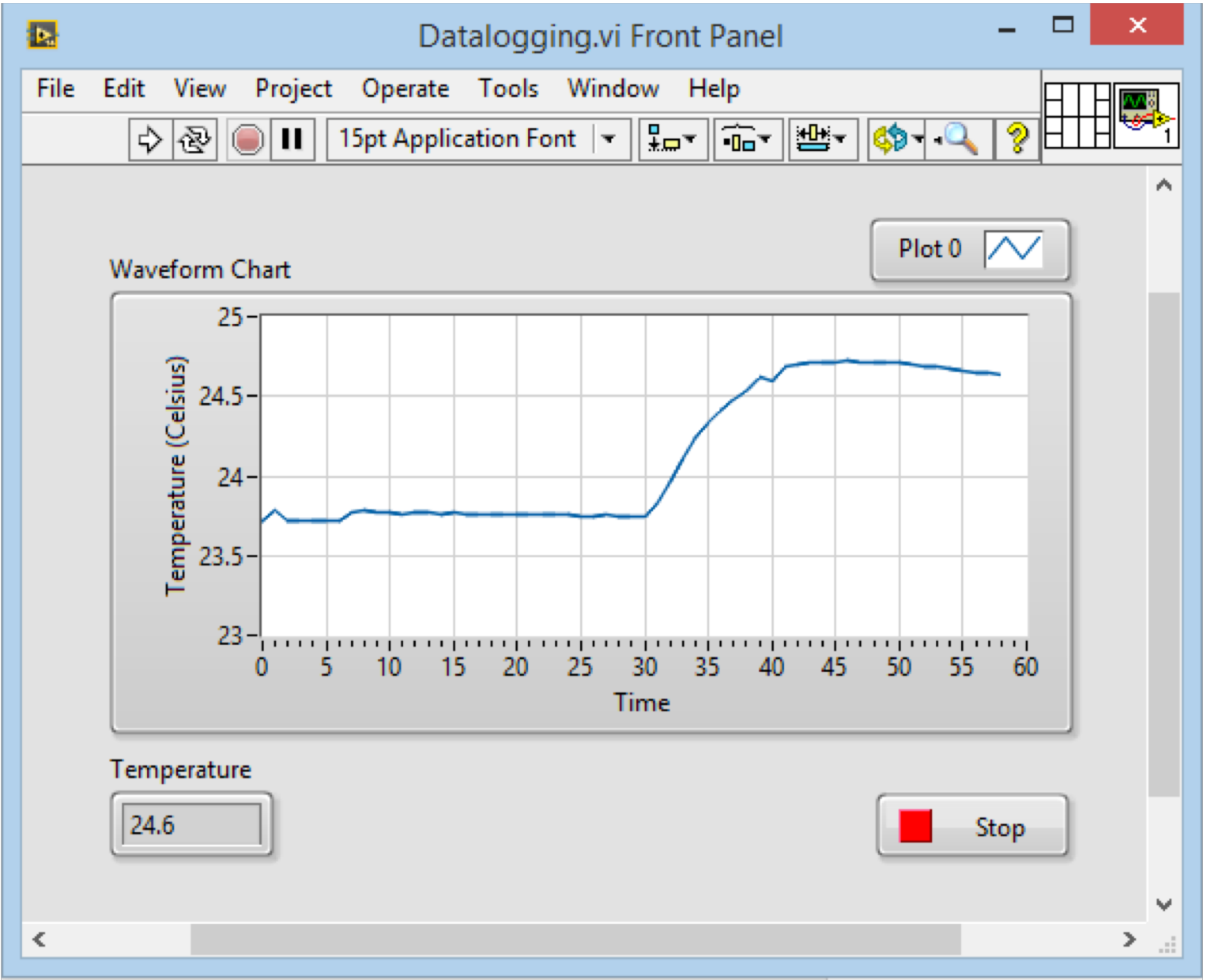


Right-click-Properties



Recommended Settings

Datalogging Example

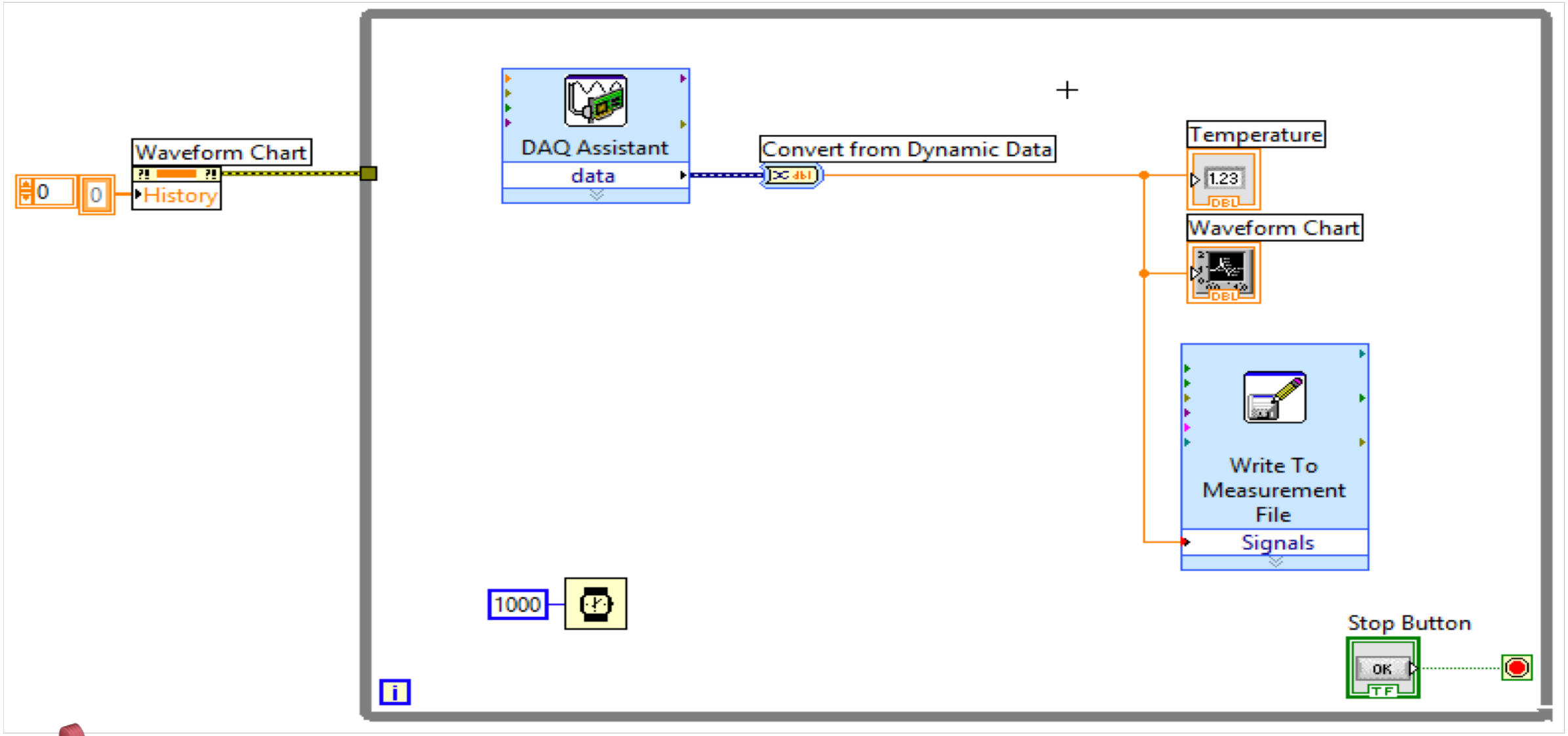


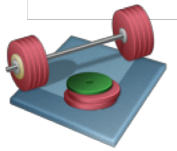
Data.lvm - Notepad

File Edit Format View Help

0.000000	23.722386
0.975883	23.782507
1.973000	23.714294
2.977028	23.719689
3.975200	23.719689
4.976168	23.716991
5.974145	23.714294
6.977184	23.774415
7.977247	23.779810
8.976395	23.777113
9.976493	23.771718
10.980489	23.763626
11.976687	23.771718
12.980719	23.766323
13.982748	23.763626
14.983700	23.766323
15.979765	23.763626
16.977789	23.760928
17.979809	23.760928
18.977904	23.760928
19.976963	23.758231
20.977973	23.755534
21.979071	23.755534
22.980054	23.752836
23.979137	23.752836
24.978214	23.750139
25.978157	23.747441
26.978513	23.752836

Datalogging Example – Block Diagram



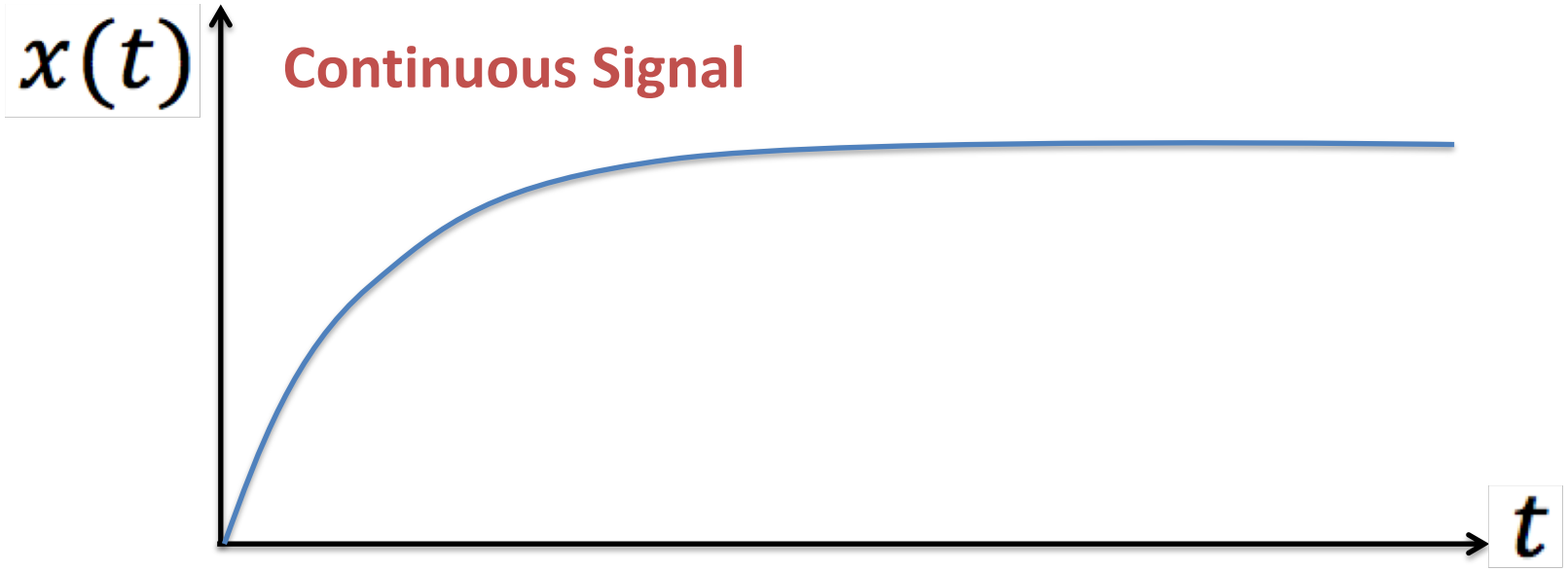


Students: **(1)** Log Data from your system, i.e., the Control Signal (u) and the output of the process (y) to a "Measurement File". **(2)** Then Plot the Data from the File in Excel.

https://www.halvorsen.blog

Discrete Systems

Hans-Petter Halvorsen

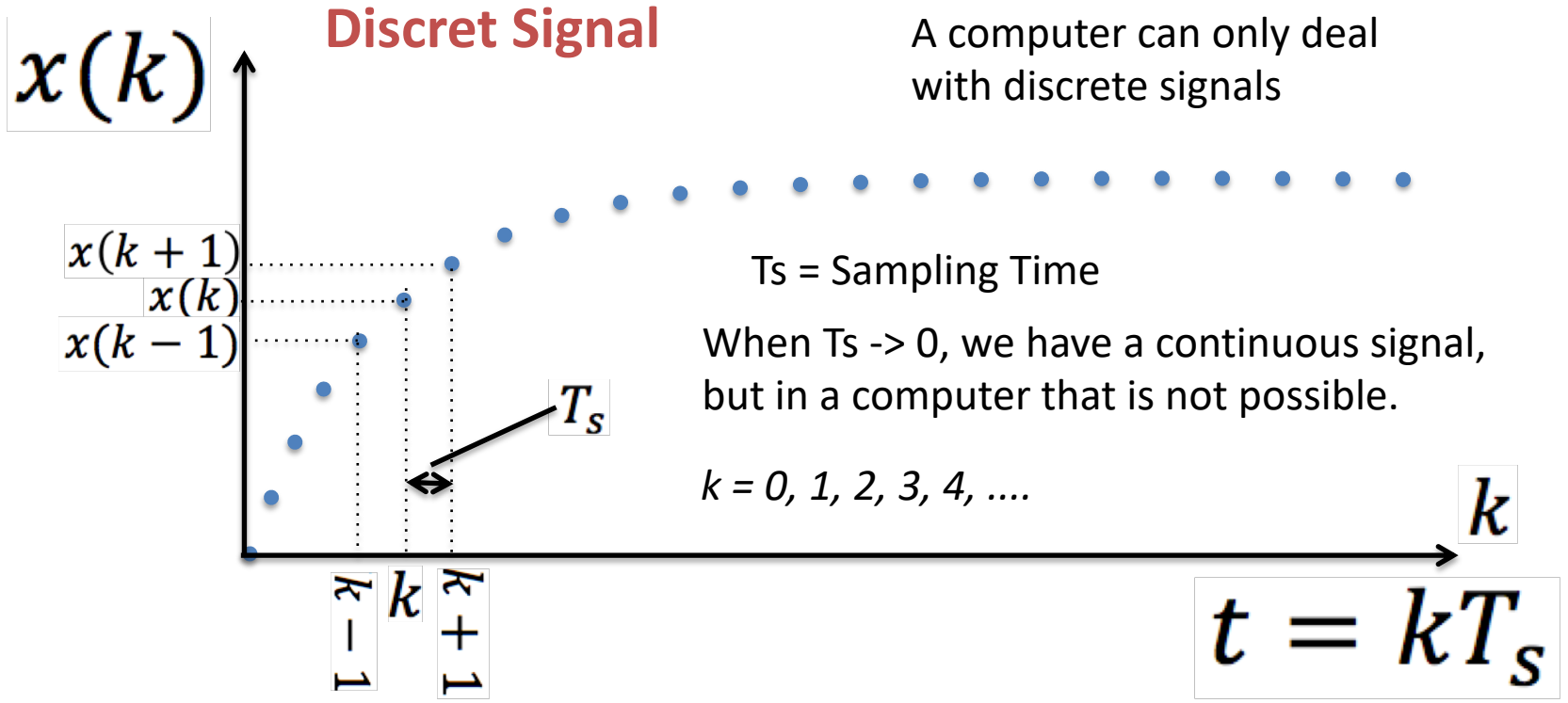


Note!
Different books use different notations

$x(k)$

$x(t_k)$

x_k



A computer can only deal with discrete signals

Ts = Sampling Time

When Ts -> 0, we have a continuous signal, but in a computer that is not possible.

k = 0, 1, 2, 3, 4, ...

Example

Discretization

Given the following differential equation:

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

In order to simulate this system in LabVIEW using the Formula Node we need to find the discrete differential equation.

We can use e.g., the **Euler Approximation**:

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

Ts – Sampling Time

Then we get:

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

This gives the following discrete differential equation:

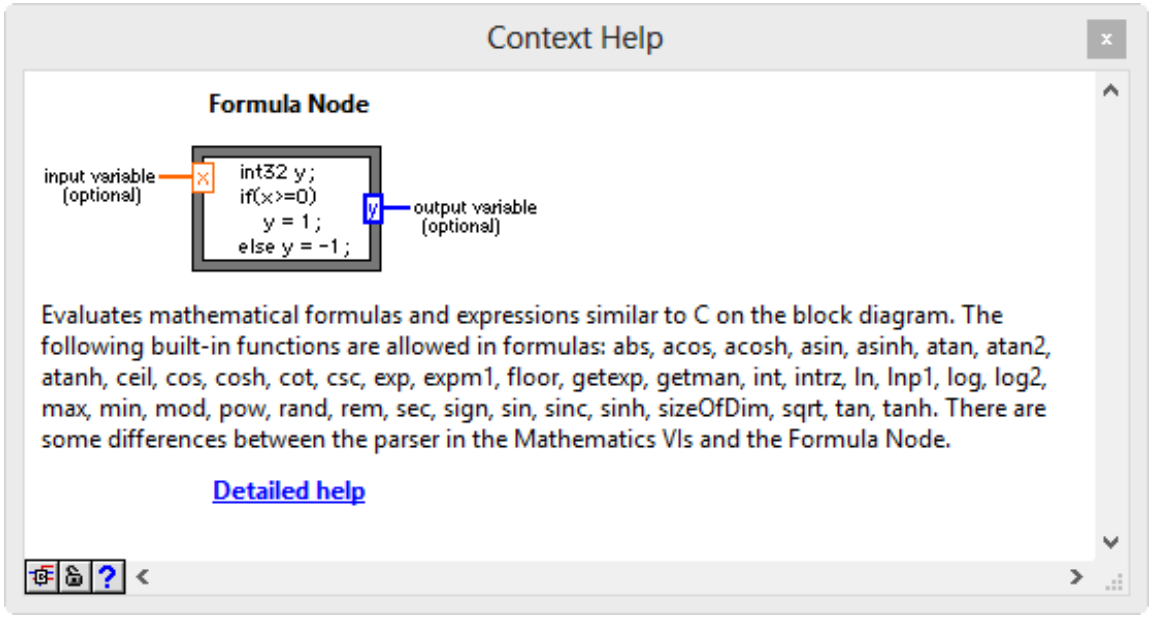
$$x(k+1) = (1 - T_s a)x(k) + T_s bu(k)$$

https://www.halvorsen.blog

Formula Node/MathScript

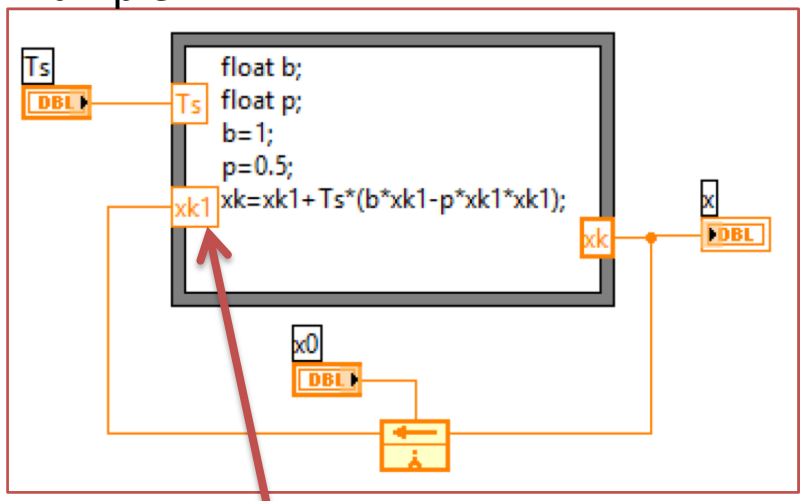
Hans-Petter Halvorsen

Formula Node & MathScript Node



Formula Node: Create and use C code within LabVIEW

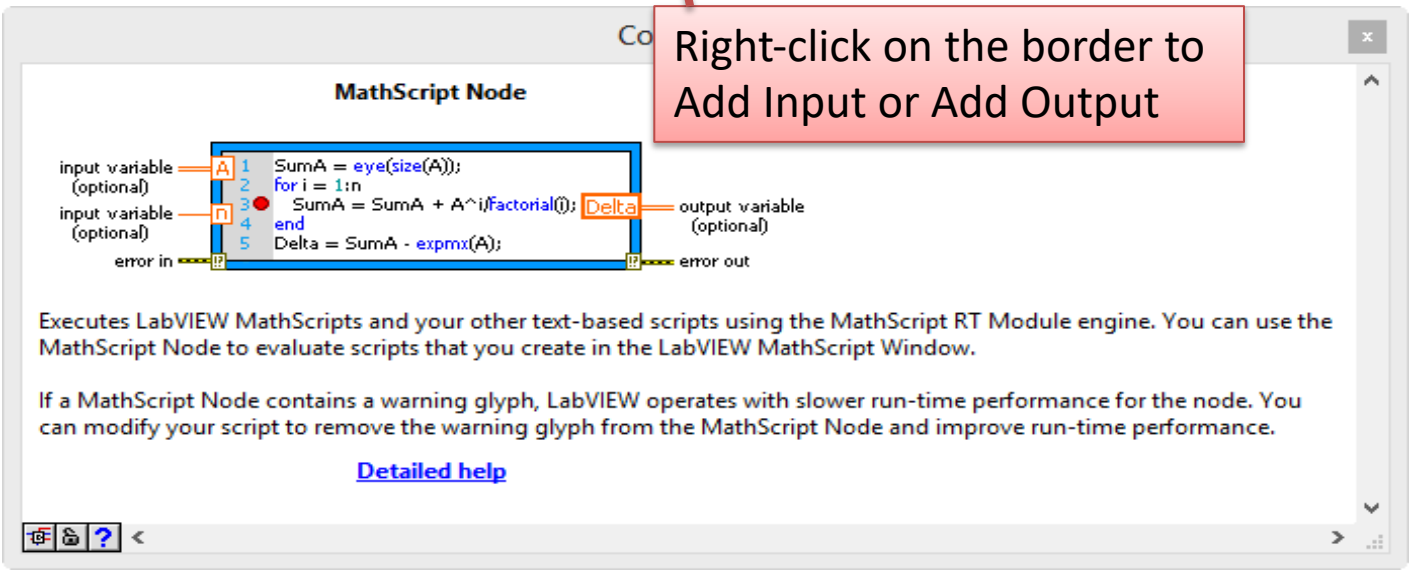
Example:



Right-click on the border to Add Input or Add Output

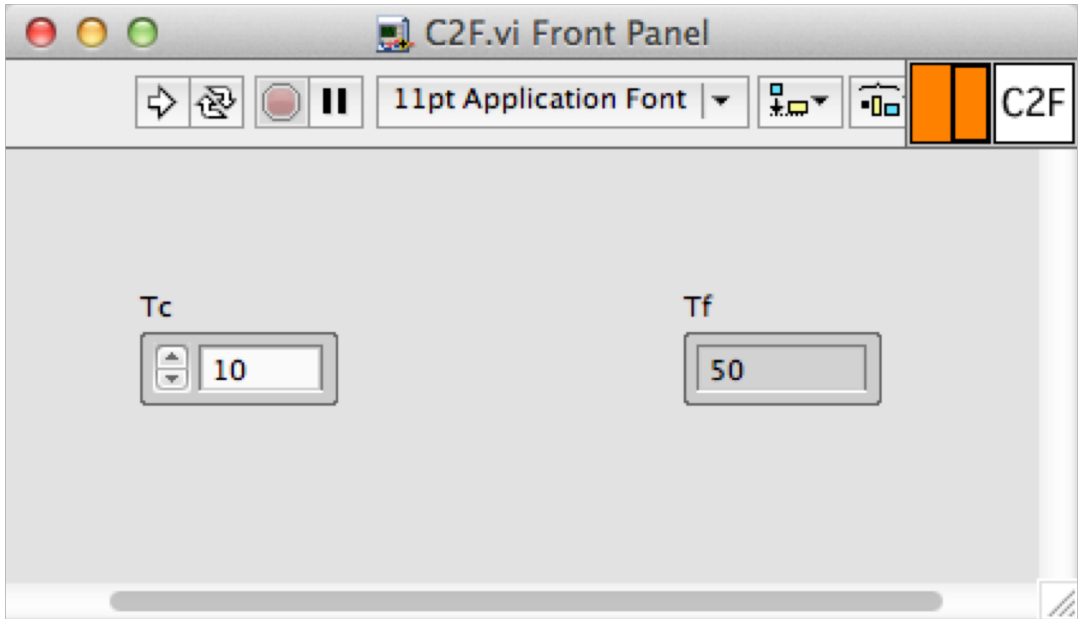
Very useful for mathematical expressions and simulations!

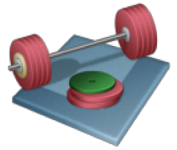
MathScript Node: Create and use MathScript/MATLAB code within LabVIEW



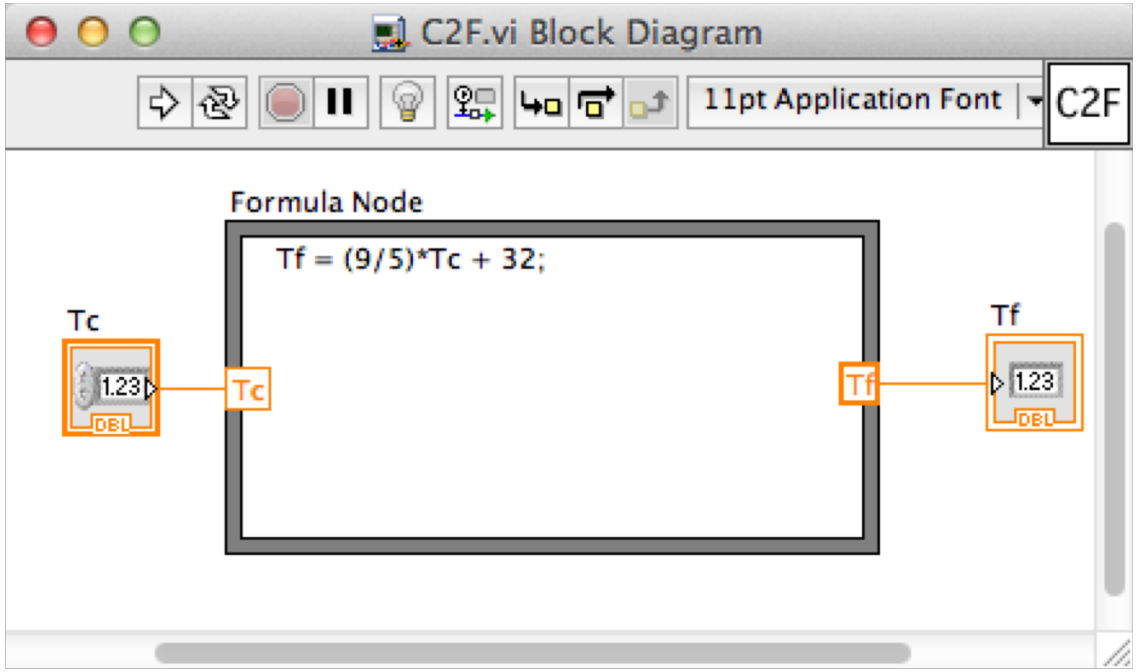
Formula Node/MathScript Node

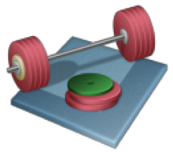
$$T_F = \frac{9}{5} T_C + 32$$





Students: Create this Example both in **Formula Node** and in **MathScript Node**





Students: Try this Formula as well (both in Formula Node and in MathScript Node):

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

Example

Simulate Discrete Systems using the Formula Node in LabVIEW

Given the following differential equation:

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

In order to simulate this system in LabVIEW using the Formula Node we need to find the discrete differential equation.

We can use the **Euler Approximation**:

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

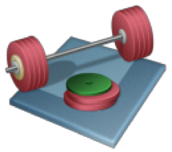
Ts – Sampling Time

Then we get:

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

This gives the following discrete differential equation:

$$x(k+1) = (1 - T_s a)x(k) + T_s bu(k)$$

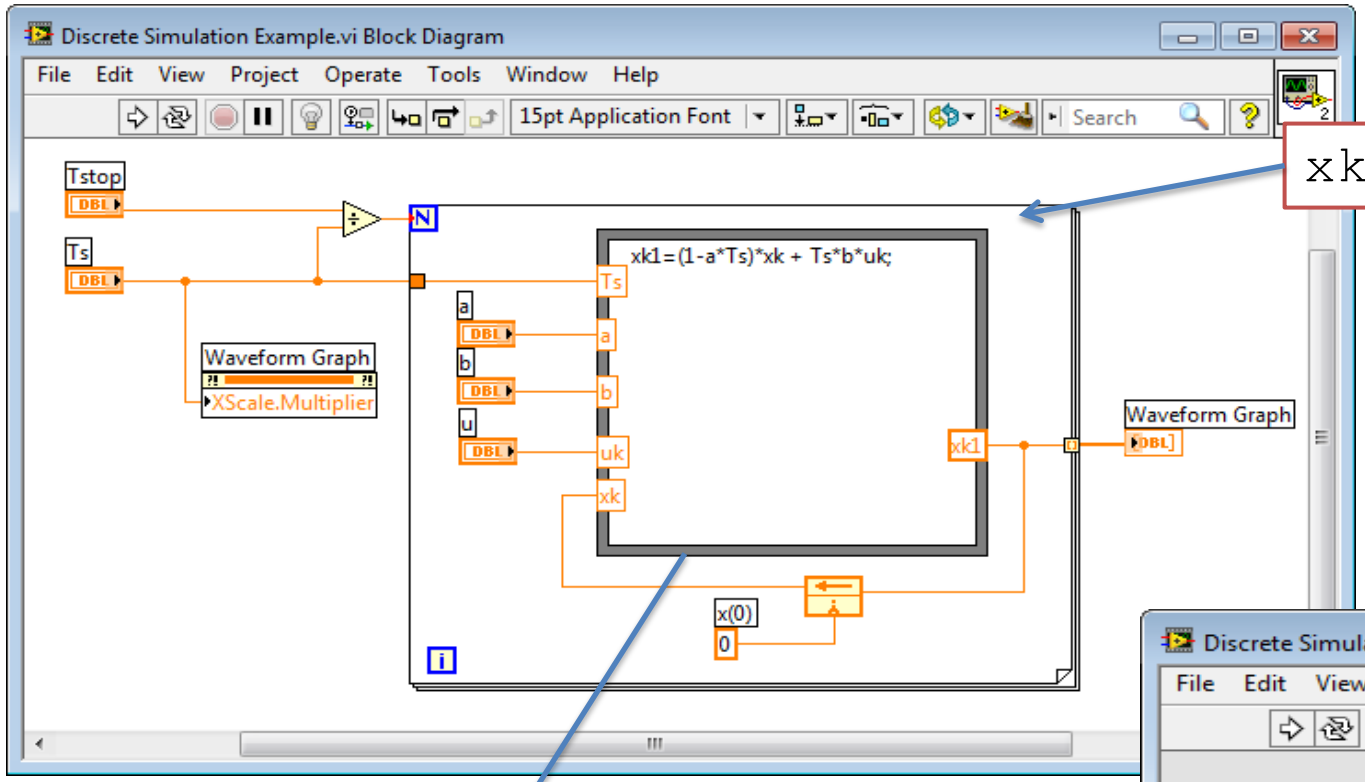


Students: Simulate and Plot the discrete system above using a **Formula Node** and a **For Loop** in LabVIEW

set $a = 0.25$ and $b = 2$

Ts = 0.1 s

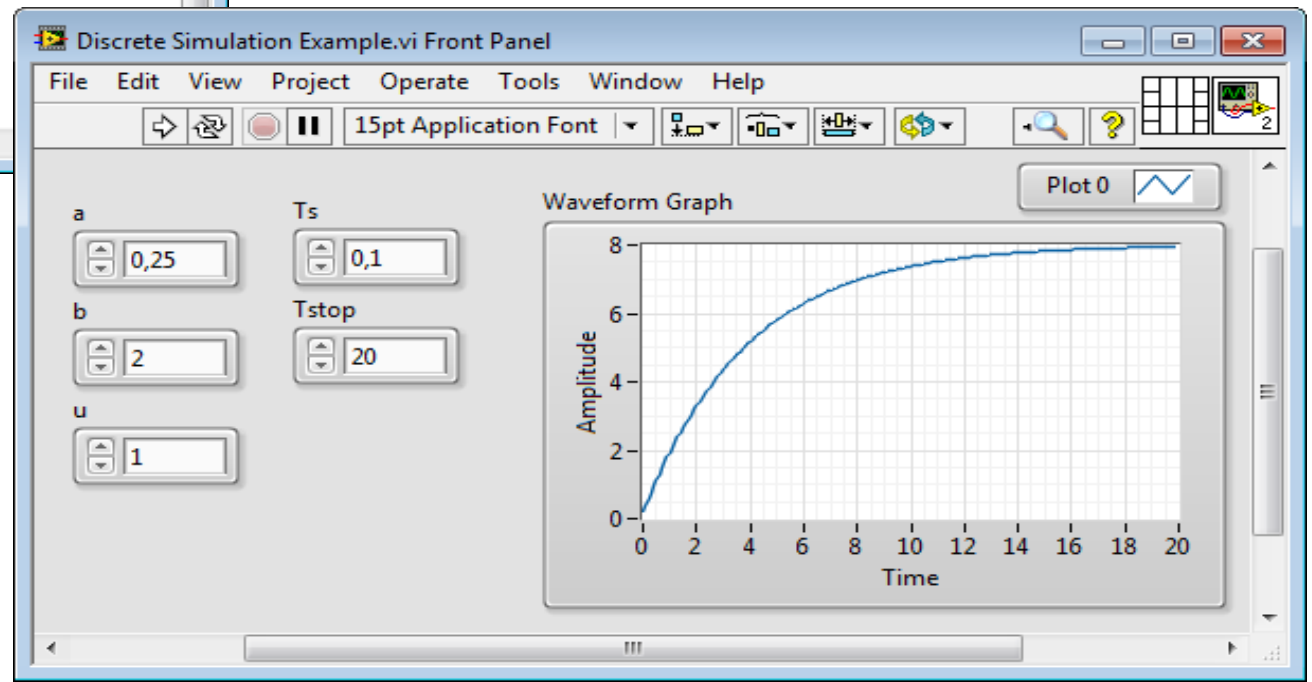
Formula Node in LabVIEW - Solutions

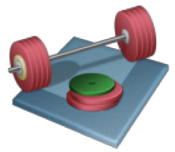


Discrete model:

```
xk1=(1-a*Ts)*xk + Ts*b*uk;
```

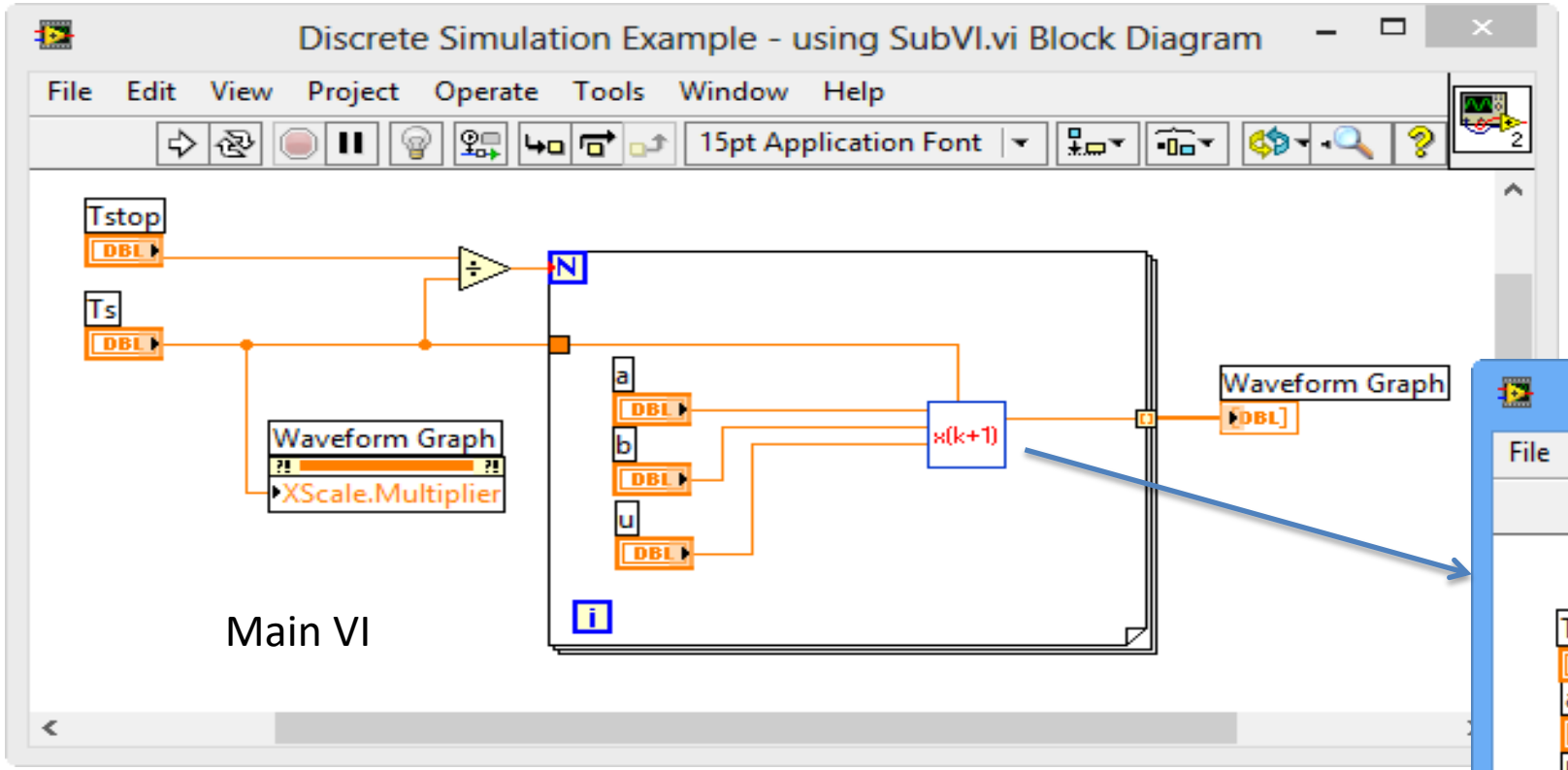
As expected, we get the same results as in the previous example using a block diagram simulation



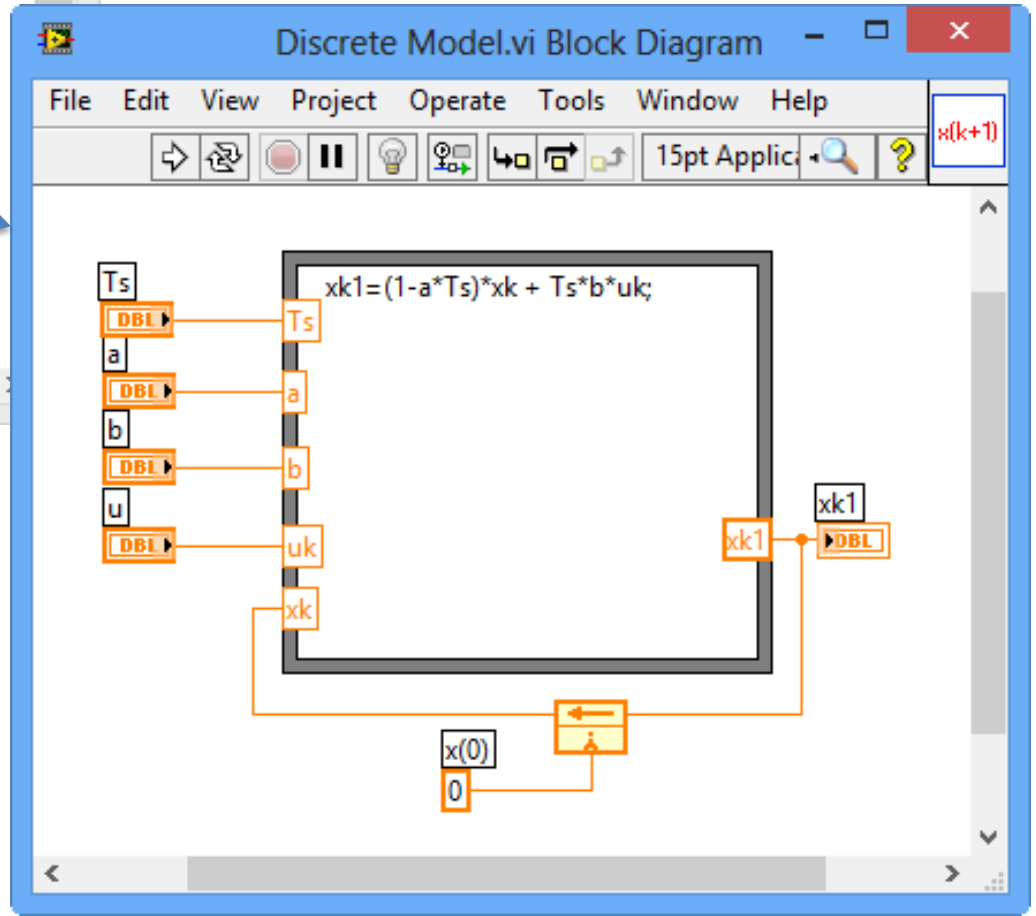


Students: Improve your solution by creating a **SubVI** of your model

Formula Node in LabVIEW – Solutions 2



SubVI

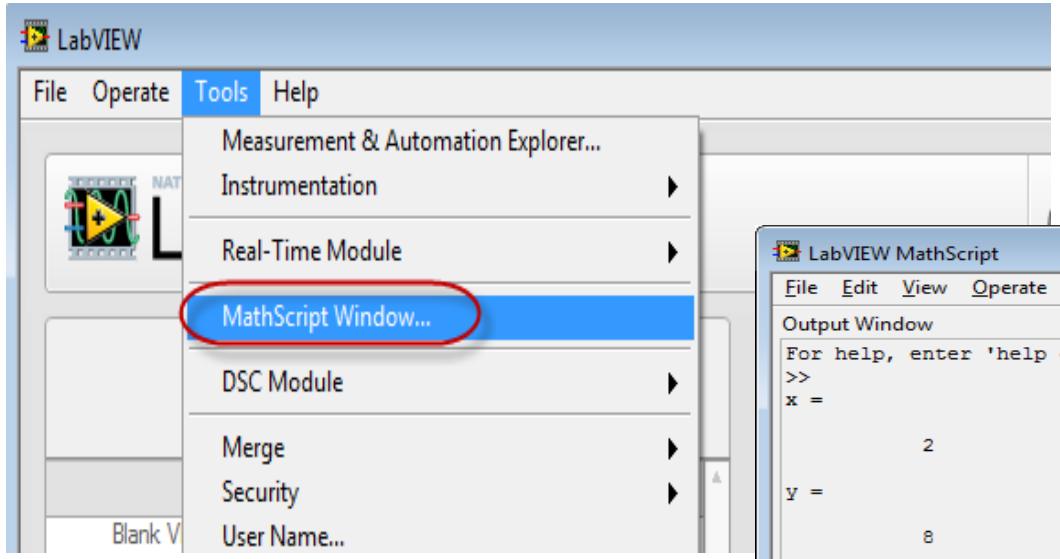


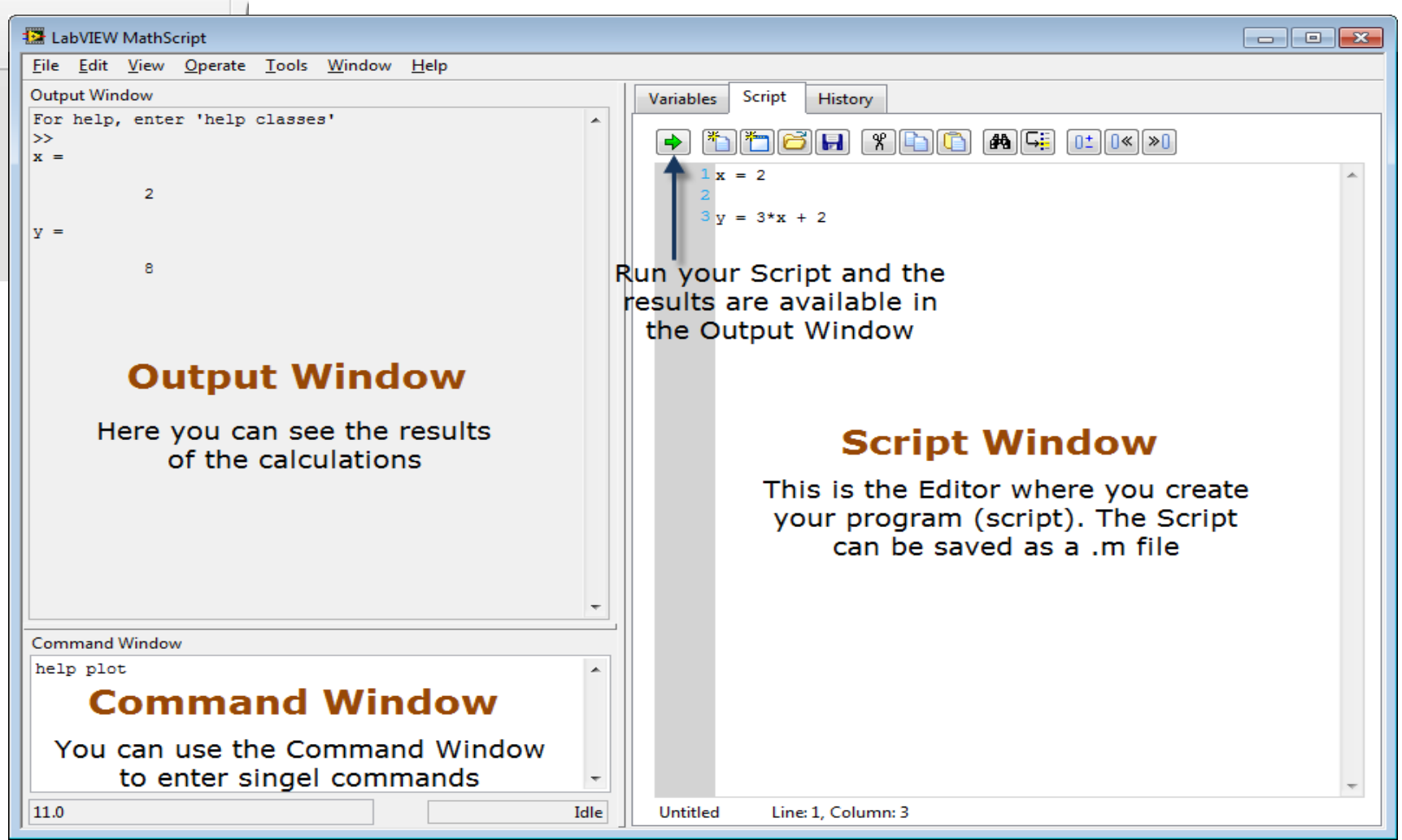
Advantages using SubVI:

- Our Main program has become simpler and easier to understand and maintain
- We can reuse our model in other applications
- If we need to do changes in the model, we only do it once and in one place!

MathScript

LabVIEW MathScript RT Module





Add-on Module for LabVIEW where we can do text-based programming and simulations
– very powerful!

MathScript Simulation Example

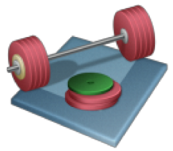
$$\dot{x} = -ax + bu$$

set $a = 0.25$ and $b = 2$

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

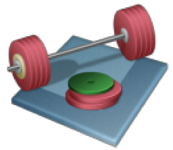
Create the following code in the MathScript "Script Editor"

```
% Simulation of discrete model
clear, clc

% Model Parameters
a = 0.25;b = 2;

% Simulation Parameters
Ts = 0.1; %s
Tstop = 20; %s
uk = 1; % Step Response
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
```

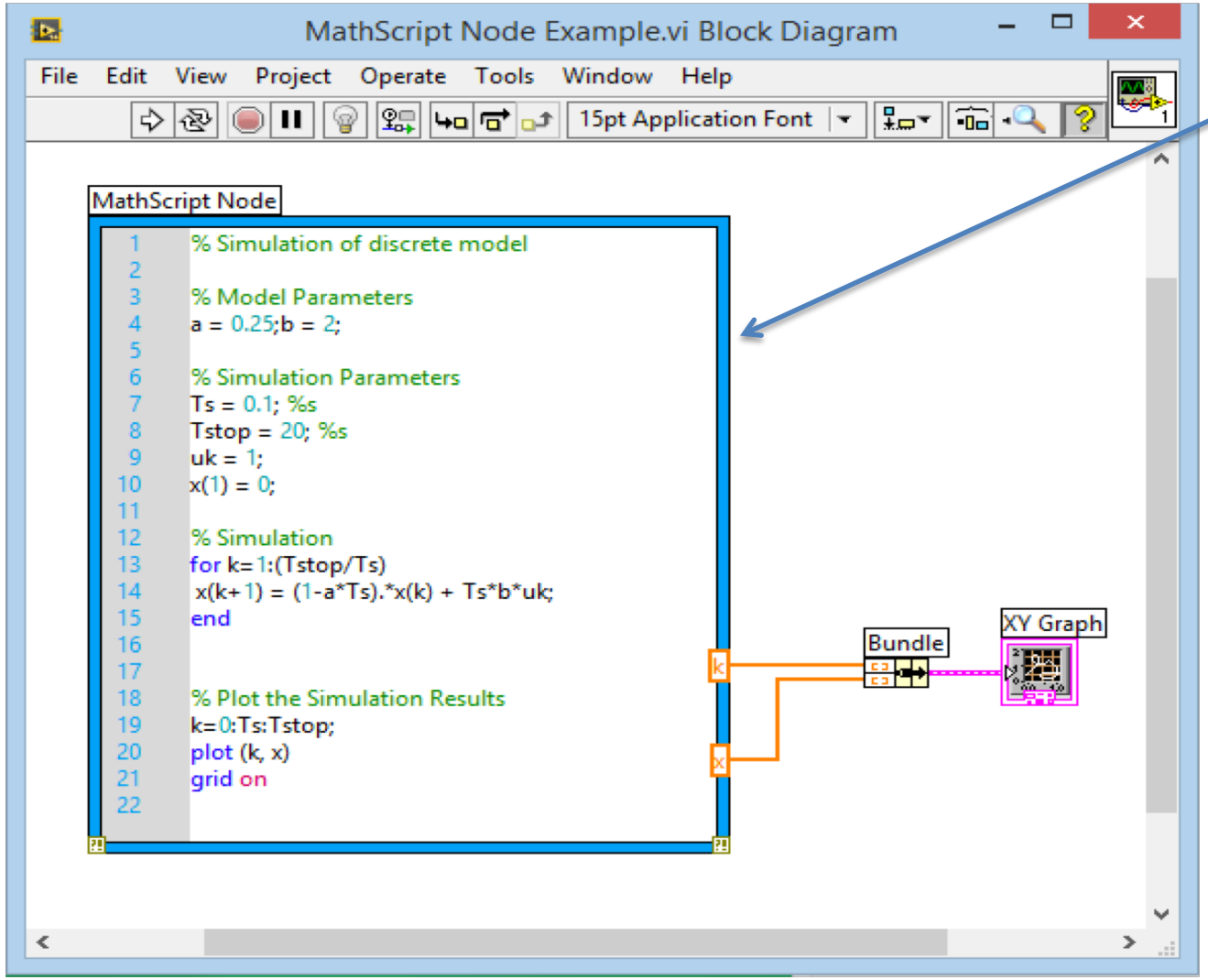


Students: Create and test the MathScript code. You should get the same results as in the LabVIEW Examples

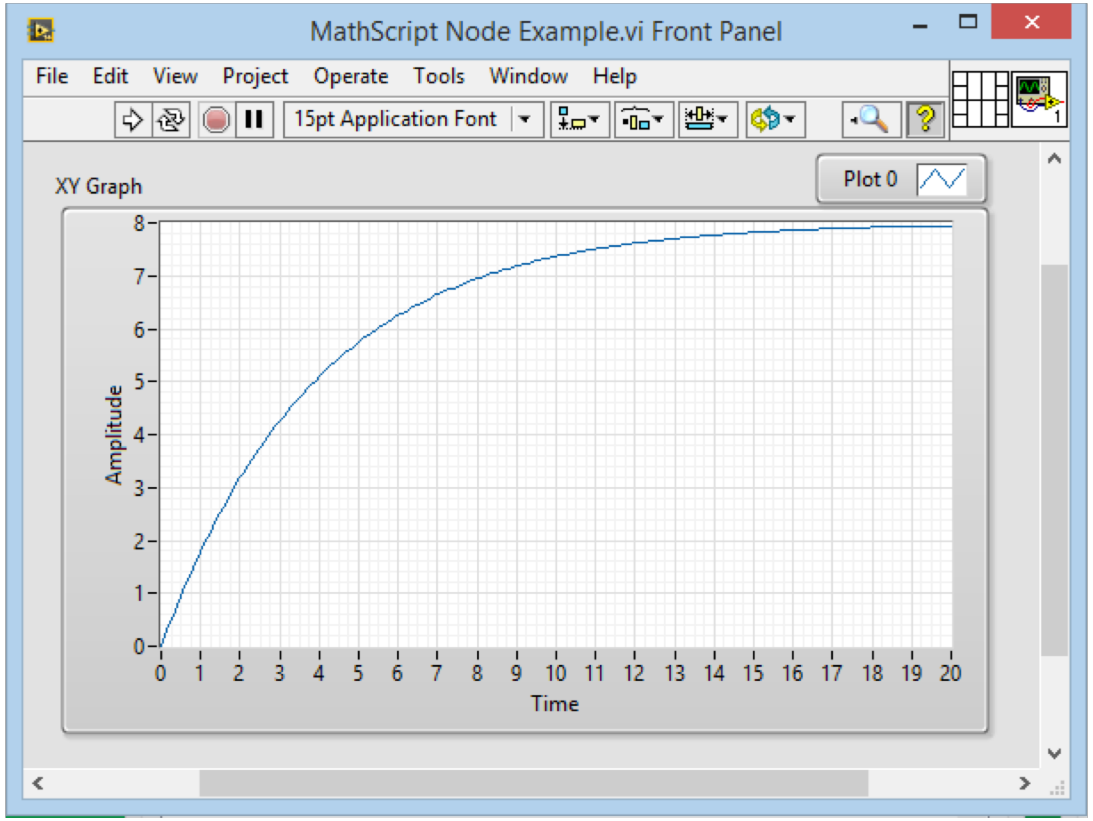
MathScript Node



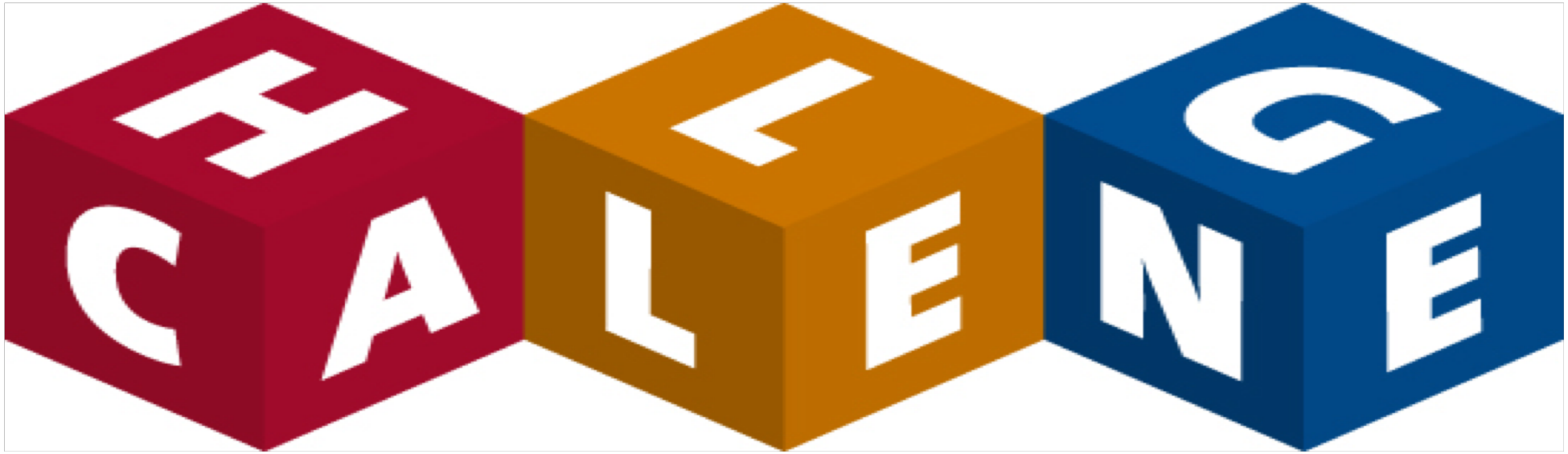
Students: Try the same example inside LabVIEW using the MathScript Node



Just copy and paste the code from the previous example



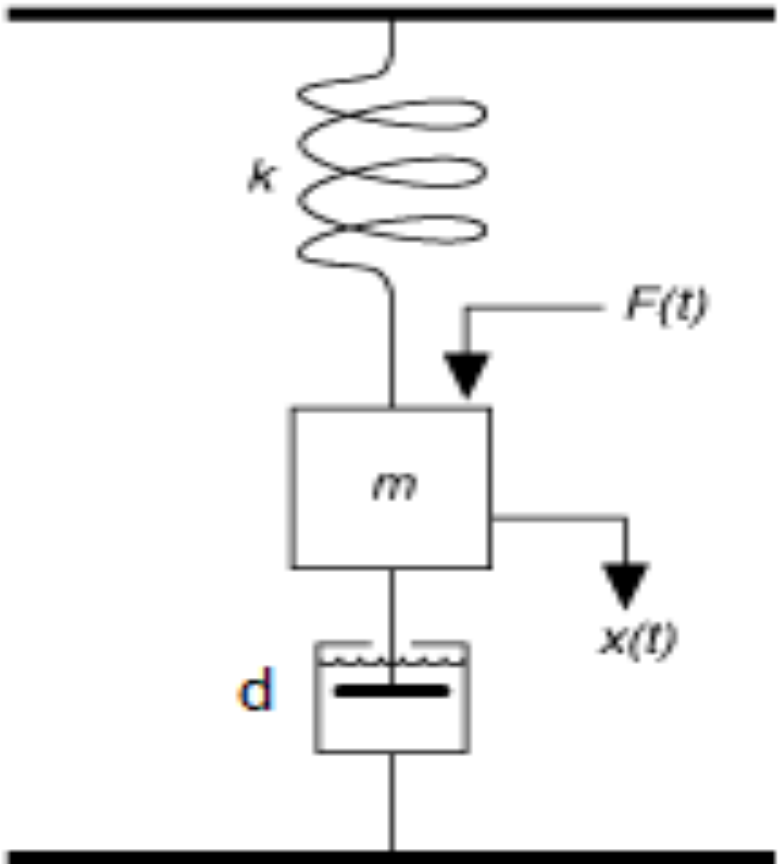
https://www.halvorsen.blog



Do you need more Practice? - Select a Challenge

Hans-Petter Halvorsen

Mass-Spring-Damper System



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

x is the position

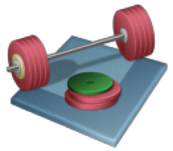
$\dot{x}$ is the speed/velocity

$\ddot{x}$ is the acceleration

F is the Force (control signal, u)

d and k are constants

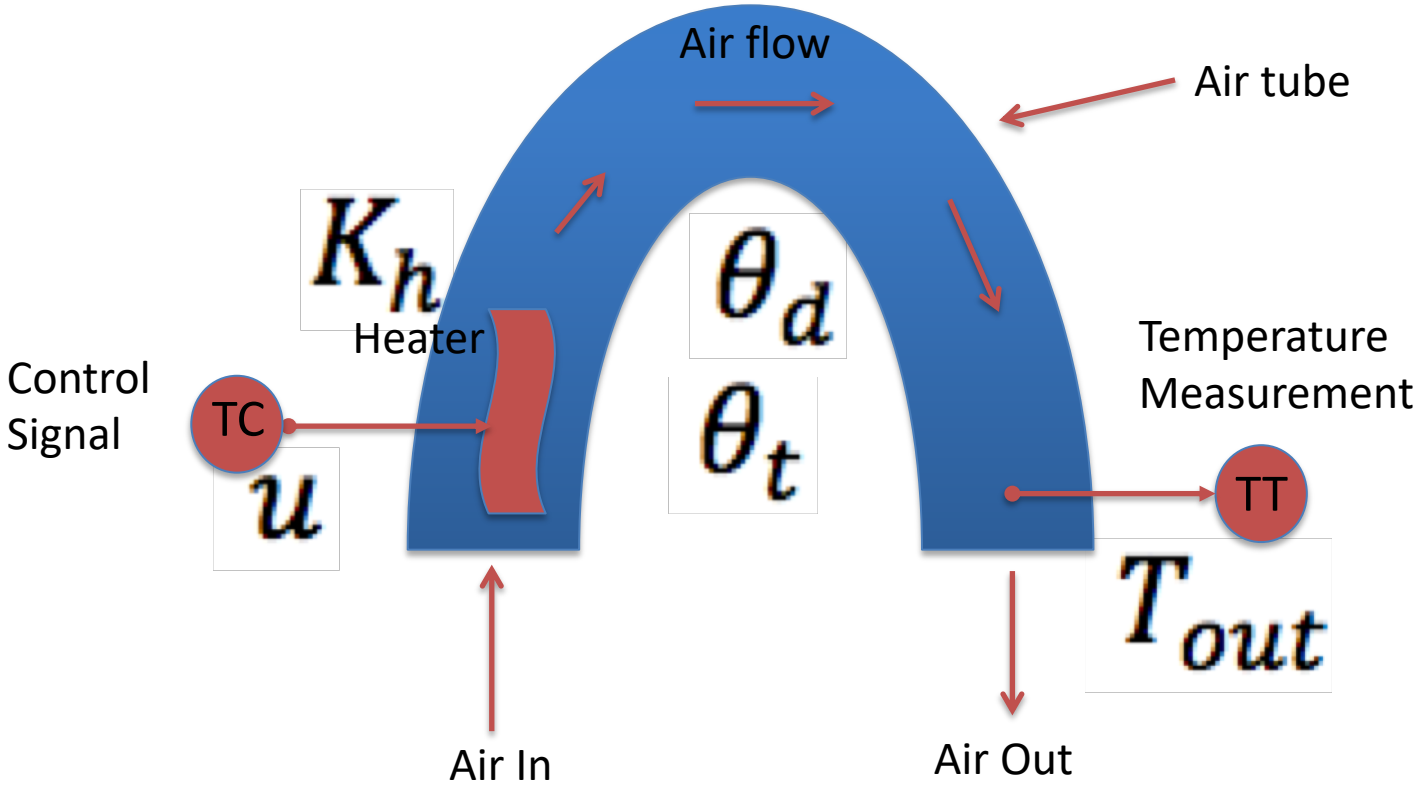
http://www.techteach.no/simview/mass_spring_damper/index.php



Students: Simulate this system using LabVIEW. Plot the position, speed and the accelleration. Test with different values on m, k and d.

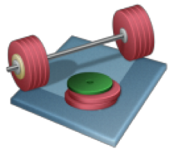
Air Heater Overview

A sketch of the Air Heater System:



Real Air Heater

Air Heater Simulation



In this assignment we are going to simulate and control a mathematical model of an Air Heater system.

A mathematical model of the system could be:

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

Where:

- T_{out}[°C] is the air temperature at the tube outlet (between 20-50°C)
- u [V] is the control signal to the heater
- θ_t [s] is the time-constant
- K_h [$deg\ C\ /\ V$] is the heater gain
- θ_d [s] is the time-delay representing air transportation and sluggishness in the heater
- T_{env}[°C] is the environmental (room) temperature.

Use the following values in the simulations:

$$\theta_t = 22\ sec$$

$$\theta_d = 2\ sec$$

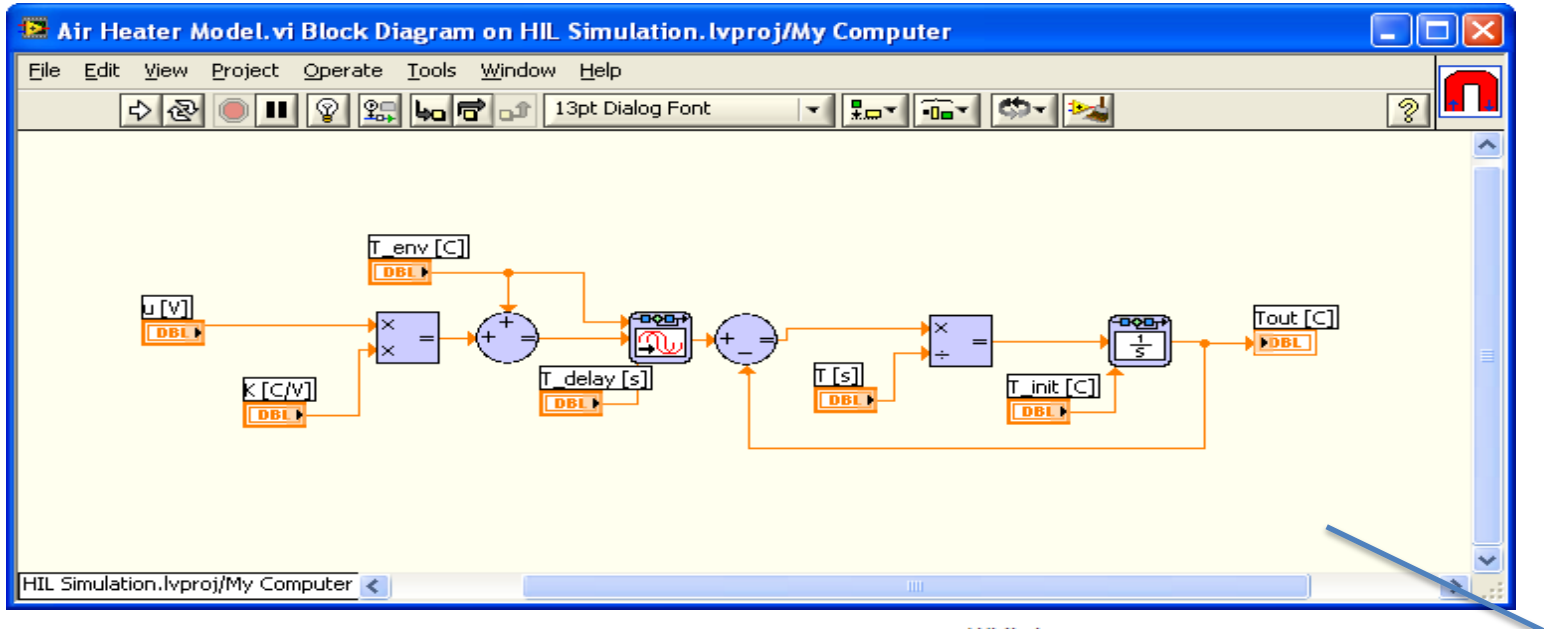
$$K_h = 3.5\ \frac{°C}{V}$$

$$T_{env} = 21.5\ °C$$

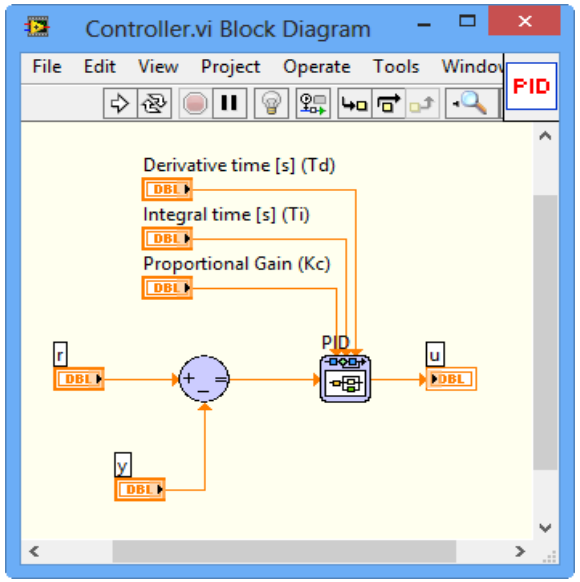
Note! You find an example of how to do it in LabVIEW on the next slides – but try to solve it by yourself first!

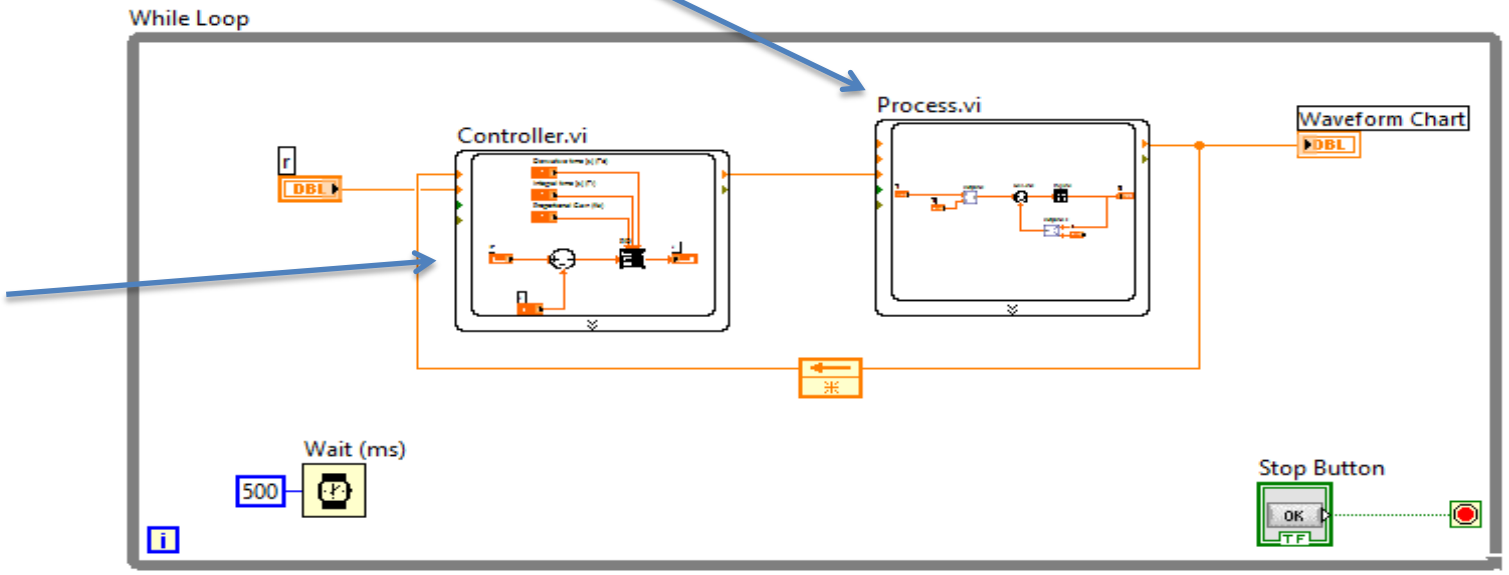
Air Heater Simulation Example

Implement the Heater model in a Simulation Sub System, as shown below:



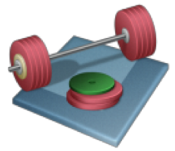
See next slide for Front Panel example



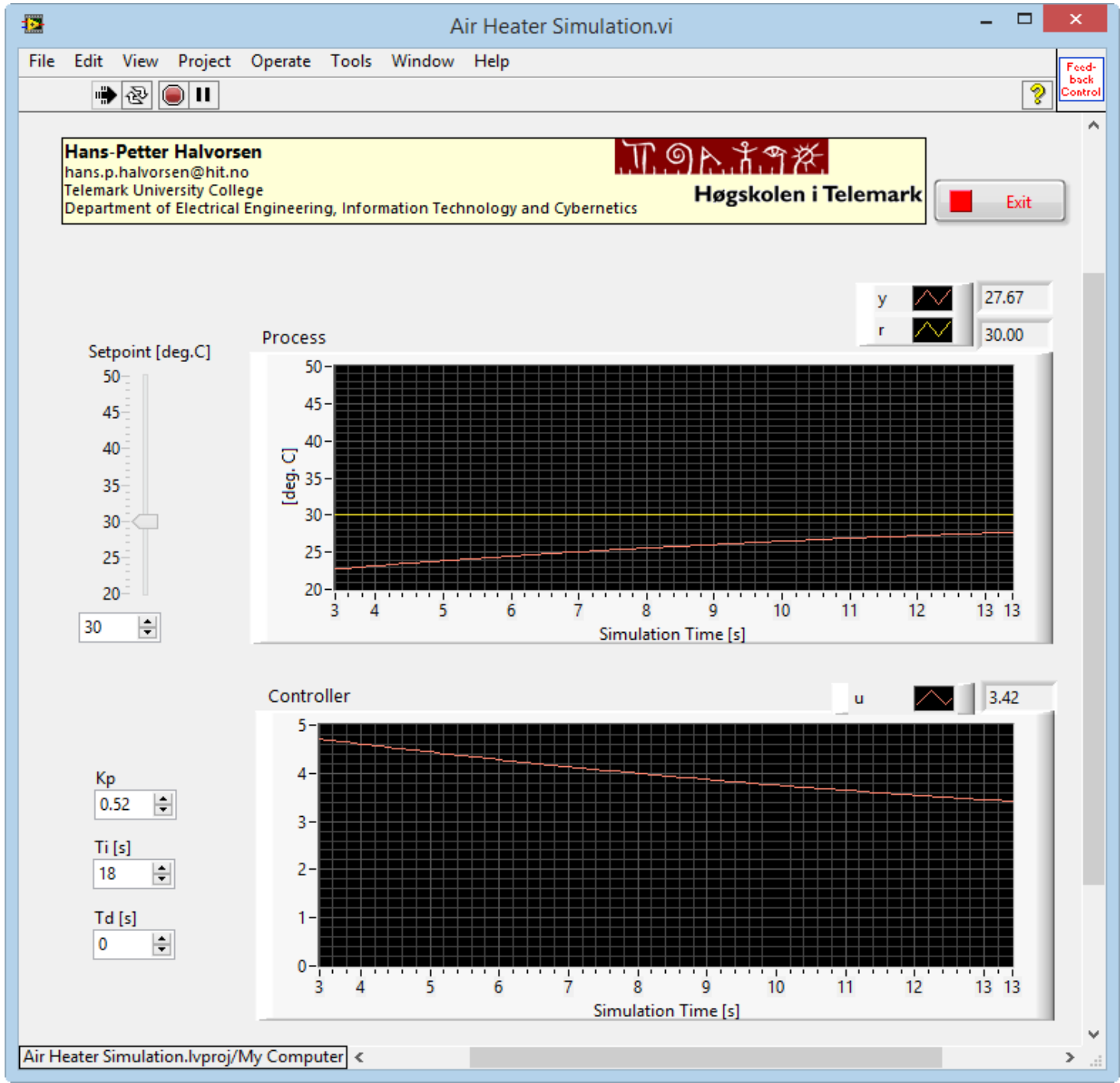


Air Heater Simulation

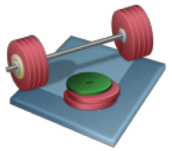
Front Panel Example:



Find Proper PI(D) Parameters



Air Heater Simulation with MathScript



Are you able to simulate the Air Heater model using MathScript or/and MathScript Node?

MathScript Simulation Example

```
% 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
```

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

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

Web: https://www.halvorsen.blog