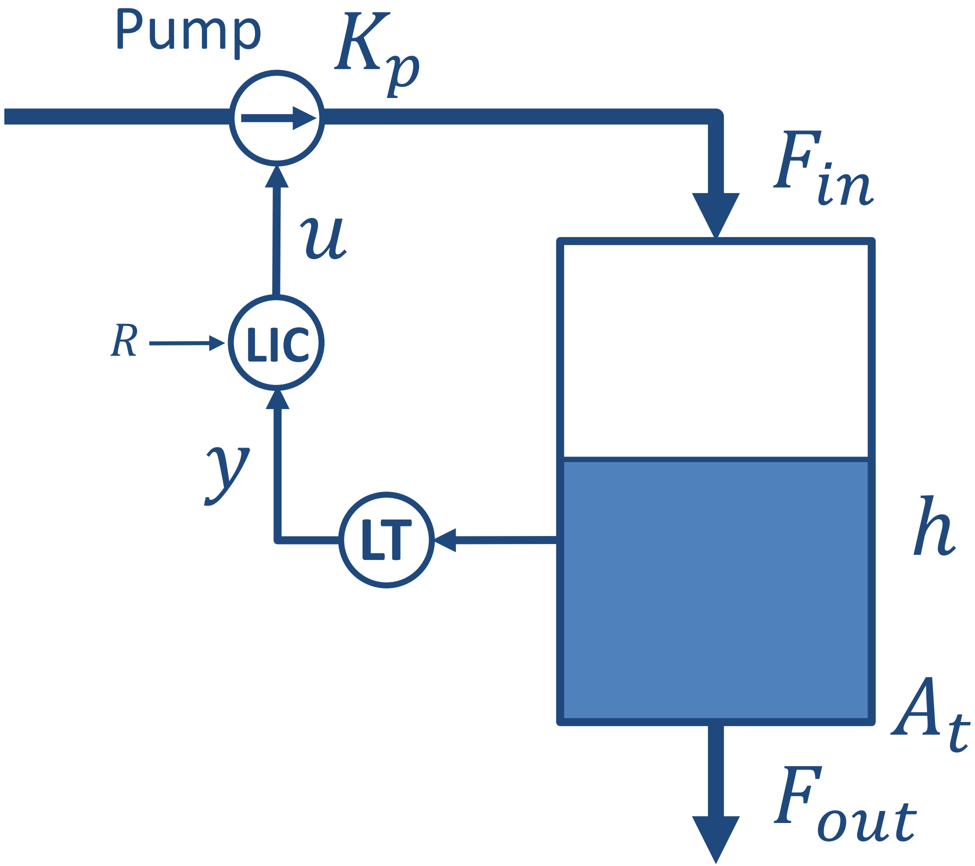
https://www.halvorsen.blog

# Level Tank System

Hans-Petter Halvorsen

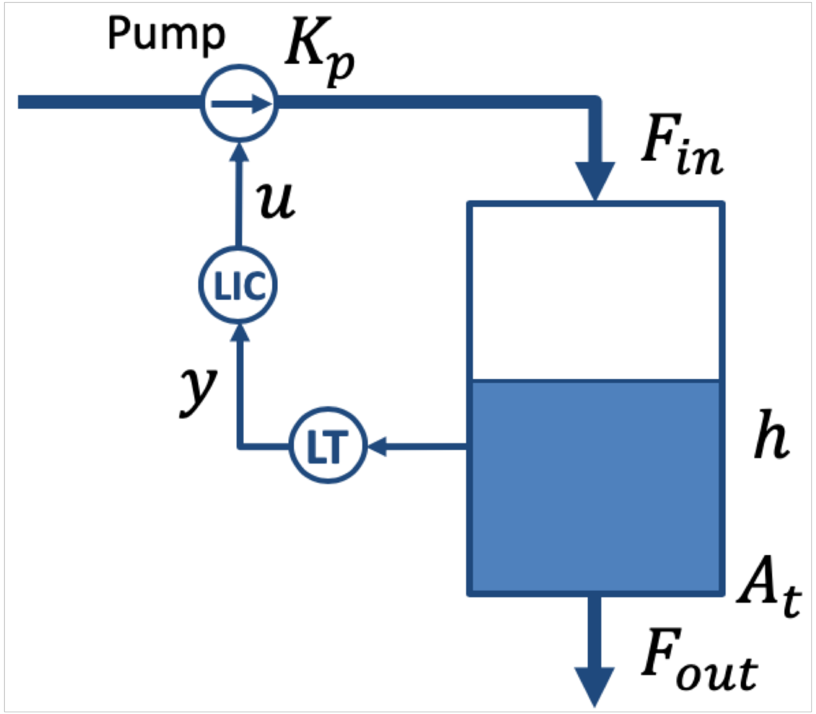
# Level Tank



Aim:

Control the Level in the Tank ($h$)

# Level Tank



$$A_t \frac{dh}{dt} = F_{in} - F_{out}$$

or:

$$\dot{h} = \frac{1}{A_t}(K_p u - F_{out})$$

Where:

- $F_{in}$ - flow into the tank , $F_{in} = K_p u$
- $F_{out}$ - flow out of the tank
- $A_t$ is the cross-sectional area of the tank

# Model Values

- $h\ [cm]$ is the level in the water tank. $0cm \leq h \leq 20cm$
- $u\ [V]$ is the pump control signal to the pump. $0V\ \leq u \leq 5V$
- $A\ [cm2]$ is the cross-sectional area in the tank
- $K_p\ [(cm3/s)/V]$ is the pump gain. The flow into the tank is $F_{in} = K_p u$, i.e. we control the flow into the tank using a pump.
- $F_{out}\ [cm3/s]$ is the outflow through the valve. The outflow may be manually adjusted with a handle.

# Model Values

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

You can assume the following values in your simulations:

$$A_t = 78.5\ cm$$

$$K_p = 16.5\ cm^3/s$$

$F_{out}$ should be adjustable from your Front Panel

The range for $F_{out}$ could, e.g., be $0 \leq F_{out} \leq 40 cm^3/s$

#1

# Level Tank model – Integrator Model

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

- $K_p$ $[cm^3/s)/V]$ is the pump gain
- $F_{out}[cm^3/s]$ is is the outflow through the valve
- $A_t$ $[cm^2]$ is the cross-sectional area of the tank
- $u$ $[V]$ is the control signal to the pump

# Level Tank model - 1.order linear system

A more accurate model may, e.g., be:

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

where $K_v$ is the valve gain on the outflow.

It is more normal to put it like this:

$$\dot{h} = -\frac{K_v}{A_t}h + \frac{K_p}{A_t}u$$

(The general term is $\dot{x} = ax + bu$)

You may find $K_p$ and $K_v$ using, e.g., the Least Square method based on logged data from the real system

The model above is a so-called Time-constant system (1.order linear system).

# Level Tank model - 1.order Nonlinear Model

The following model is even more accurate:

$$\dot{h} = \frac{1}{A_t}\left[K_p(u - u_0) - K_v\sqrt{\rho g h}\right]$$

This is a so-called 1.order nonlinear model

- $h\ [cm]$ is the level
- $u\ [V]$ is the pump control signal to the pump
- $u_0$ is the bias voltage needed to get any flow (with $u$ less than $u_0$ there is no flow into the tank)
- $A_t[cm^2]$ is the cross-sectional area of the tank
- $K_p[(cm3/s)/V]$ is the pump gain
- $K_v$ is the valve constant. It depends on the opening of the valve, but if the opening is constant, $K_v$ is constant
- $\rho$ is the is the density of the liquid (water: $1\ kg/m^3$)
- $g$ is the is the gravity constant, $9.81\ \mathrm{m}/s^2$

You may find $K_p$ and $K_v$ using, e.g., the Least Square method based on logged data from the real system

https://www.halvorsen.blog

# Level Tank in LabVIEW

Hans-Petter Halvorsen

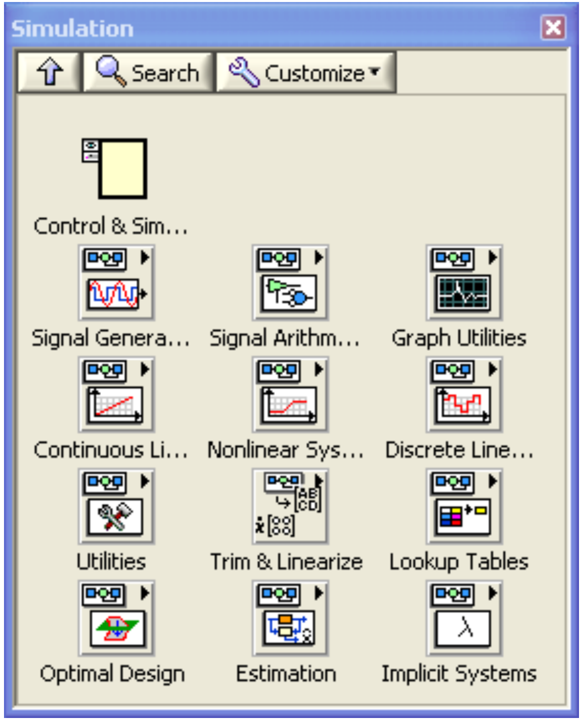
# Level Tank in LabVIEW

You can implement the Level Tank in LabVIEW in different ways:

- The model can be implemented using the blocks (Integrator, Summation, Multiplication, etc.) from the Simulation palette in LabVIEW (LabVIEW Control Design and Simulation Module)
- Create a Discrete version of the differential equation (use e.g., Euler Forward). Then use the Formula Node, MathScript Node or MATLAB Node inside LabVIEW

You should test both these alternatives

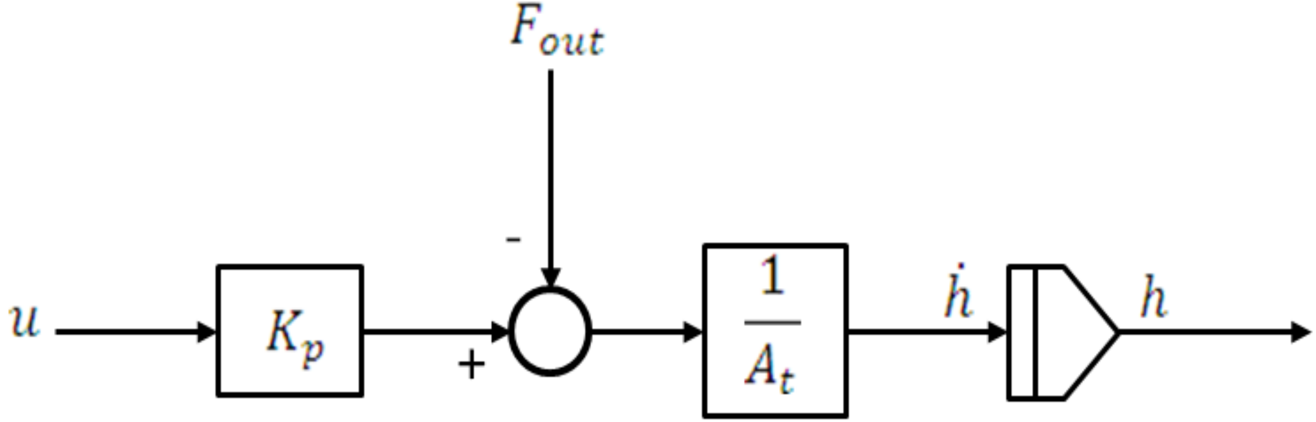
# LabVIEW Simulation Palette



# Block Diagram of Level Tank

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

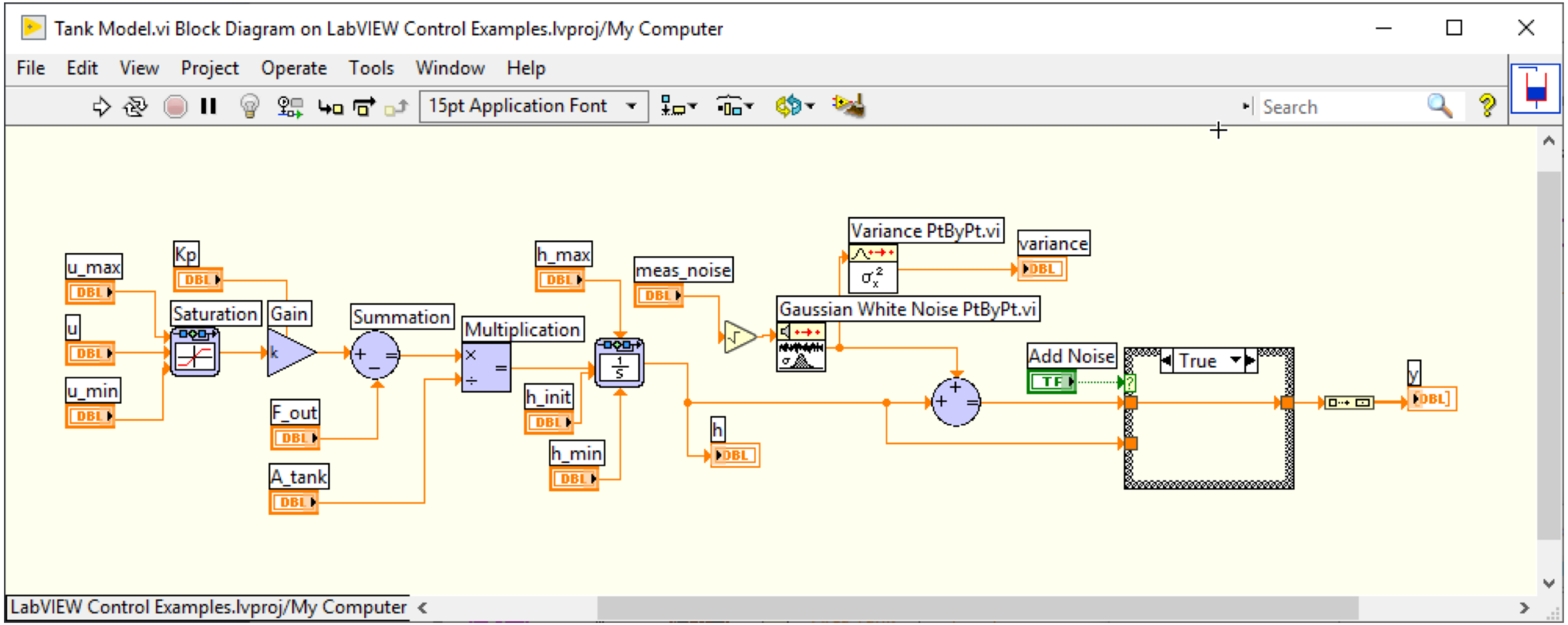
Here you see a “Pen and Paper” version of the block diagram for the Level Tank



This block diagram can easily be implemented in LabVIEW using the LabVIEW Control Design and Simulation Module

# Level Tank Model in LabVIEW

LabVIEW Control Design and Simulation Module



**Note!** This model is implemented in a so-called “**Simulation Subsystem**” (which is recommended!!!)

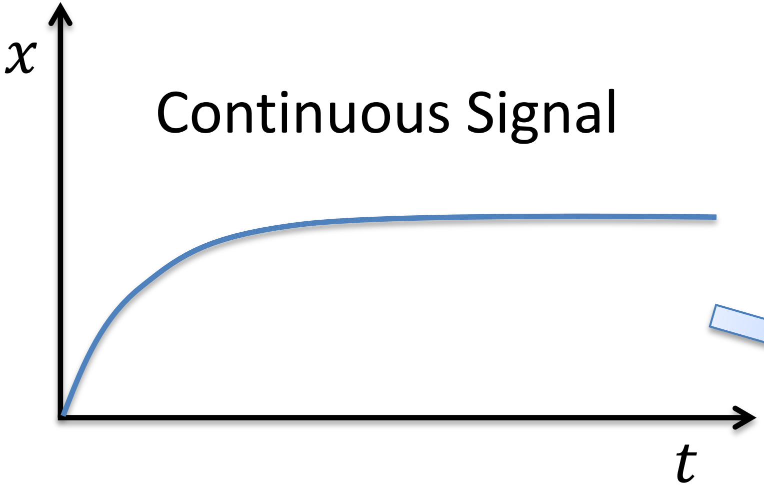
https://www.halvorsen.blog

# Discretization

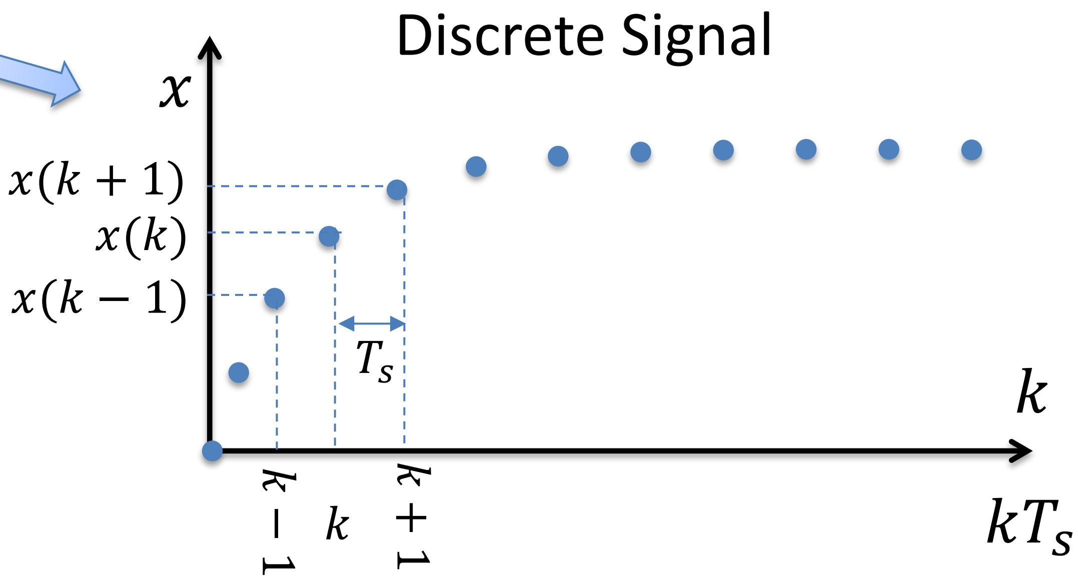
Hans-Petter Halvorsen

# Continuous vs. Discrete Systems

A computer can only deal with discrete signals



Continuous Signal



Discrete Signal

$T_s$ - Sampling Interval

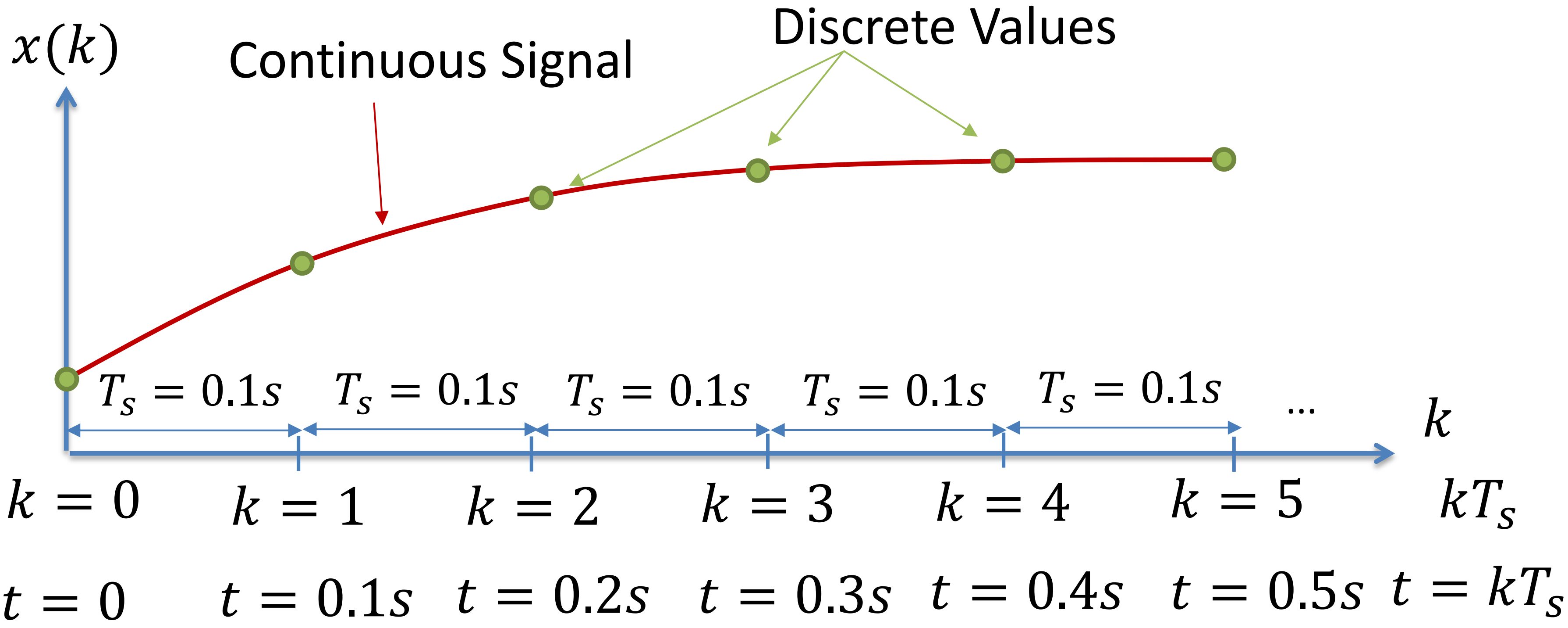
$x(k-1)$ - Previous Value

$x(k)$ - Current Value

$x(k-1)$ - Next Value

# Continuous vs. Discrete Systems - Example

In this Example we have used Sampling Interval $T_s = 0.1s$



# Discretization

Continuous Model:

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

We can use e.g., the Euler Approximation in order to find the discrete Model:

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

$T_s$ - Sampling Time

$x(k)$ - Present value

$x(k+1)$ - Next (future) value

The discrete Model will then be on the form:

$$x(k+1) = x(k) + \ \dots$$

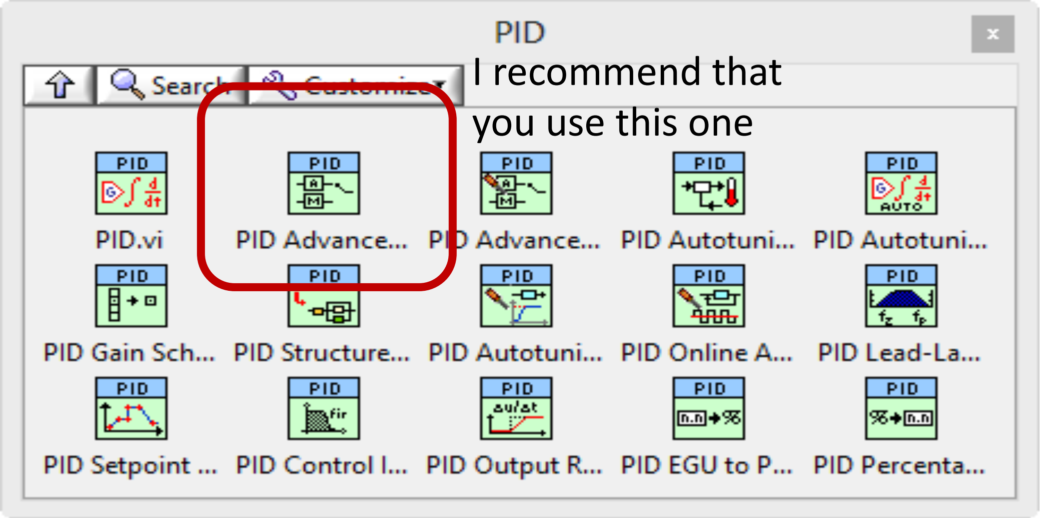
We can then implement the discrete model in any programming language

https://www.halvorsen.blog

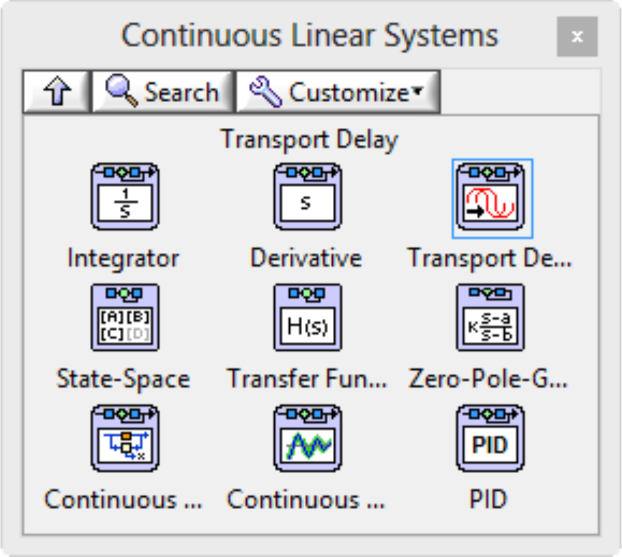
# Control System in LabVIEW

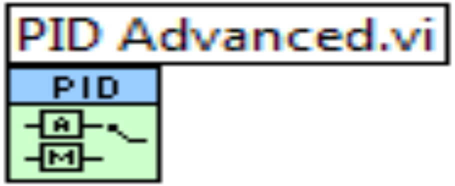
Hans-Petter Halvorsen

# Built-in PID in LabVIEW



I recommend that you use this one

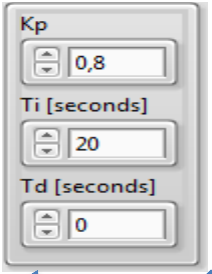




# PID in LabVIEW

Example

Front Panel

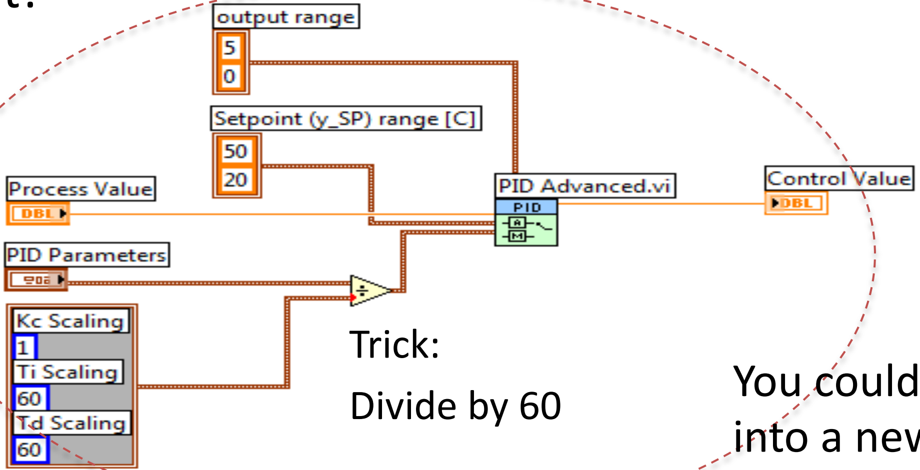

Kp
0,8
Ti [seconds]
20
Td [seconds]
0

Cluster

Normally we use <u>seconds</u> as unit for Ti and Td (which is recommended!)

But the built-in PID algorithm in LabVIEW uses <u>minutes</u> as unit!

Block Diagram:

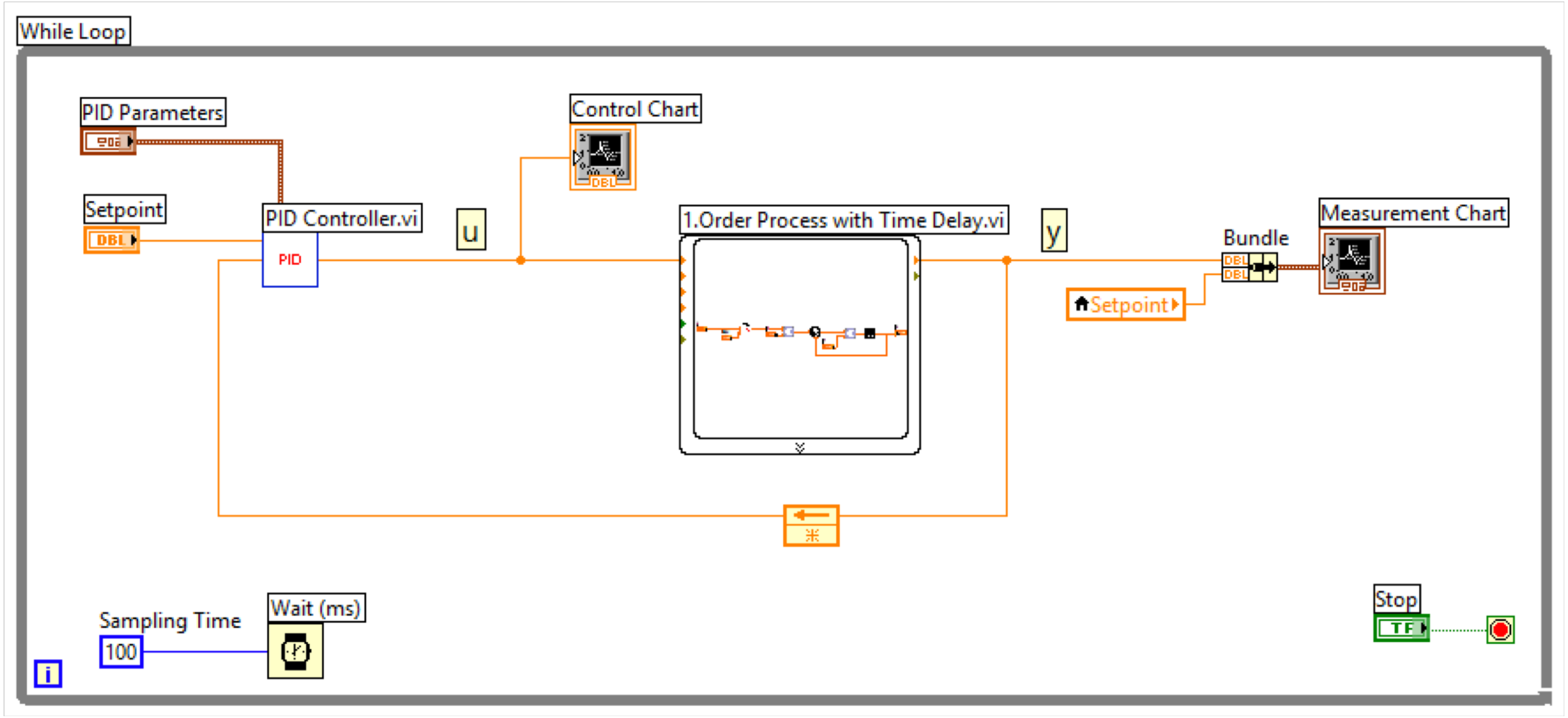


Trick:

Divide by 60

You could also put this code into a new SubVI

# Example of Control System in LabVIEW



# The PID Algorithm

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

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

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

$r$ is the Reference Signal or Set-point

$y$ is the Process value, i.e., the Measured value

Tuning Parameters:

| | |
|---|---|
| $K_p$ | Proportional Gain |
| $T_i$ | Integral Time [sec.] |
| $T_d$ | Derivative Time [sec.] |

# PID Parameters

You may use the following Parameters as a starting point:

$$K_p = 3$$

$$T_i = 15\ s$$

$$T_d = 0$$

Then you can "fine-tune" them by using "Trial and Error", i.e., run the simulations with different values for the parameters and observe if the results are good or not

# Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

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

Web: https://www.halvorsen.blog