



Model Predictive Control (MPC) in LabVIEW

Hans-Petter Halvorsen, M.Sc.

Examples of Industrial Control Systems (ICS)

Industrial Control Systems are computer controlled systems that monitor and control industrial processes that exist in the physical world



National Instruments
cRIO

Programmable Automation Controller (PAC) **4**



LabVIEW **1**
Industrial PID Controller



PC based Control System/SCADA System (Supervisory Control And Data Acquisition) **5**



I/O Module

2 Distributed Control Systems (DCS)



Controller I/O Modules

DeltaV from Emerson

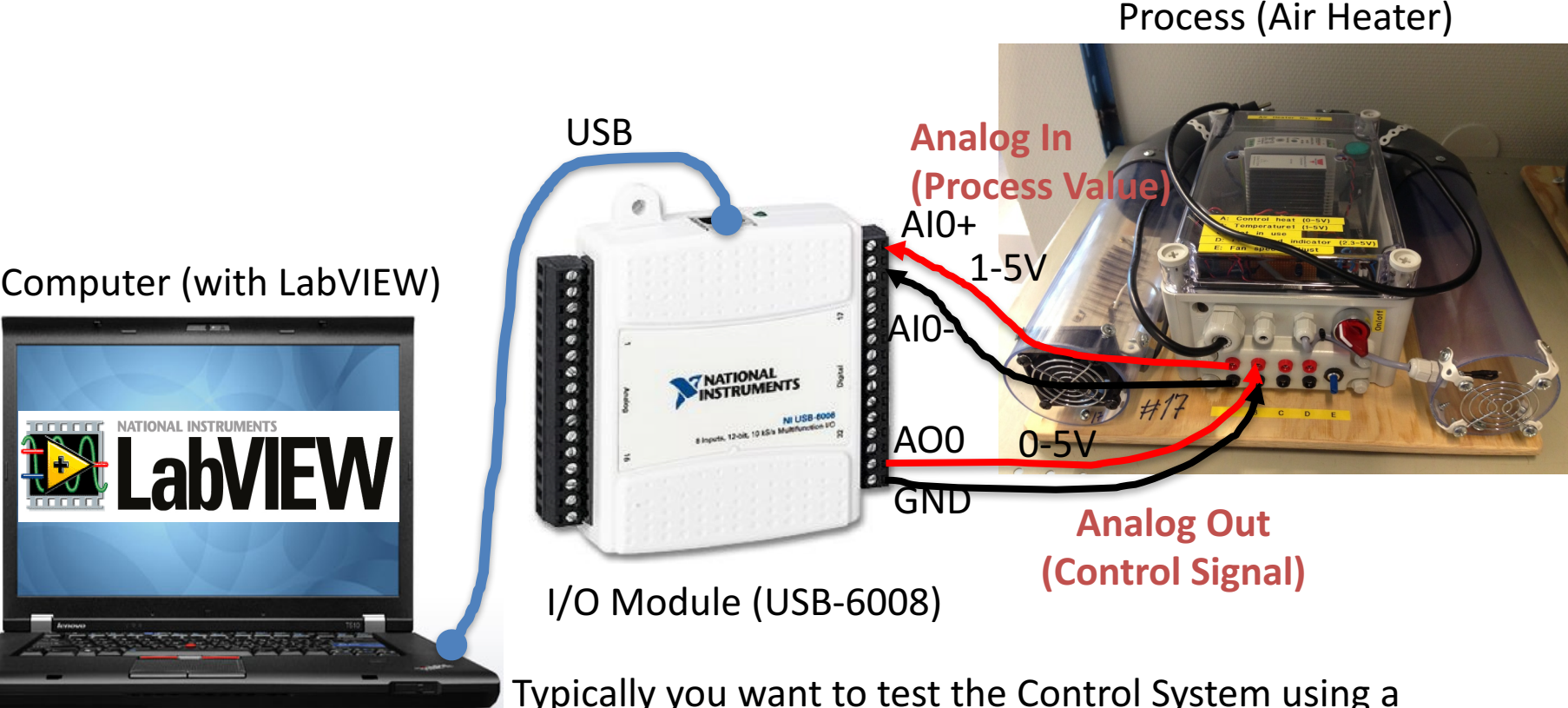


3 PLC (Programmable Logic Controller)



Siemens PLC

Example of PC-based Control System



Computer (with LabVIEW)

Process (Air Heater)

USB

Analog In
(Process Value)

AI0+

1-5V

AI0-

AO0

0-5V

GND

Analog Out
(Control Signal)

I/O Module (USB-6008)

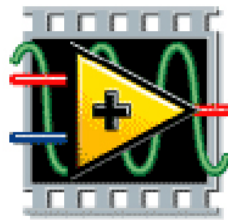
PID Control and Monitoring

Typically you want to test the Control System using a Mathematical Model before you use it on the Real System

Necessary Software



- LabVIEW
- LabVIEW Control & Simulation Module
- DAQmx Driver Software



NATIONAL INSTRUMENTS

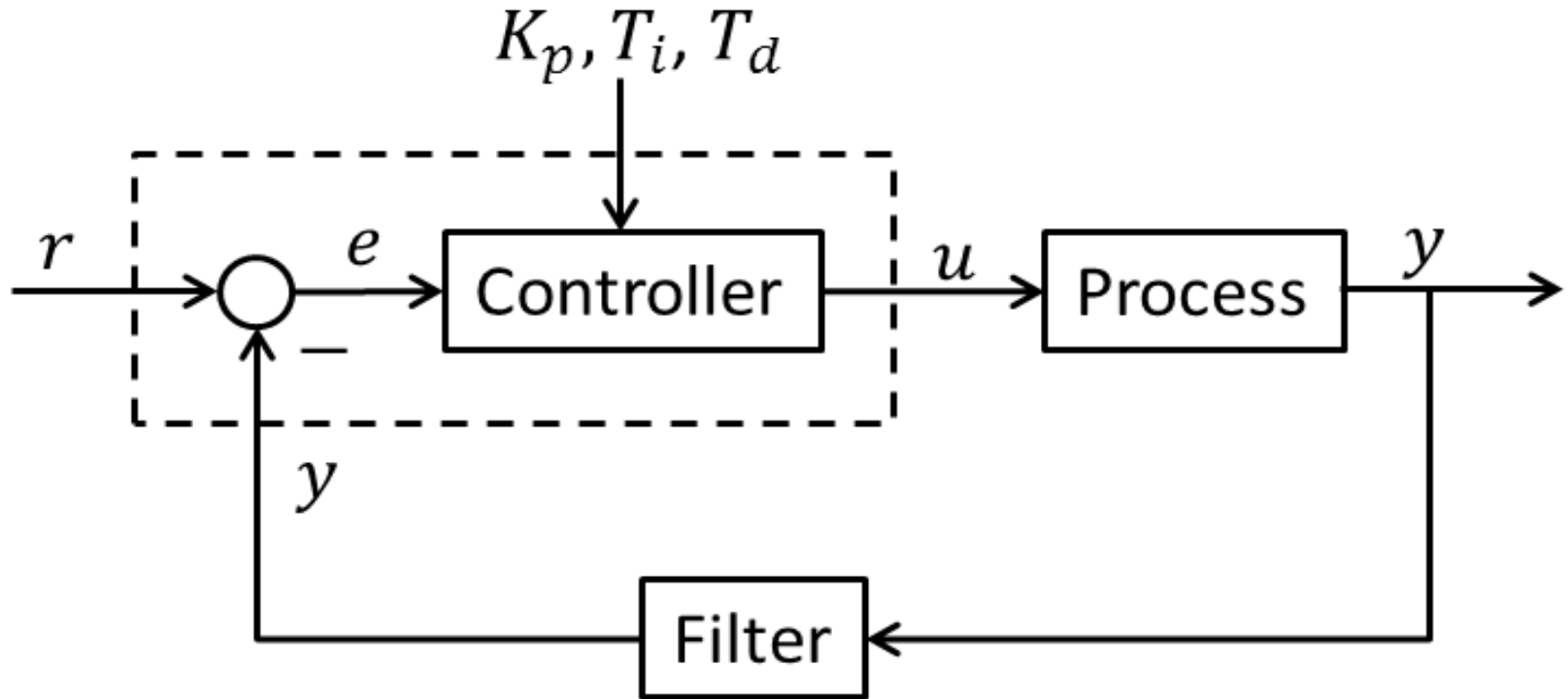
LabVIEW



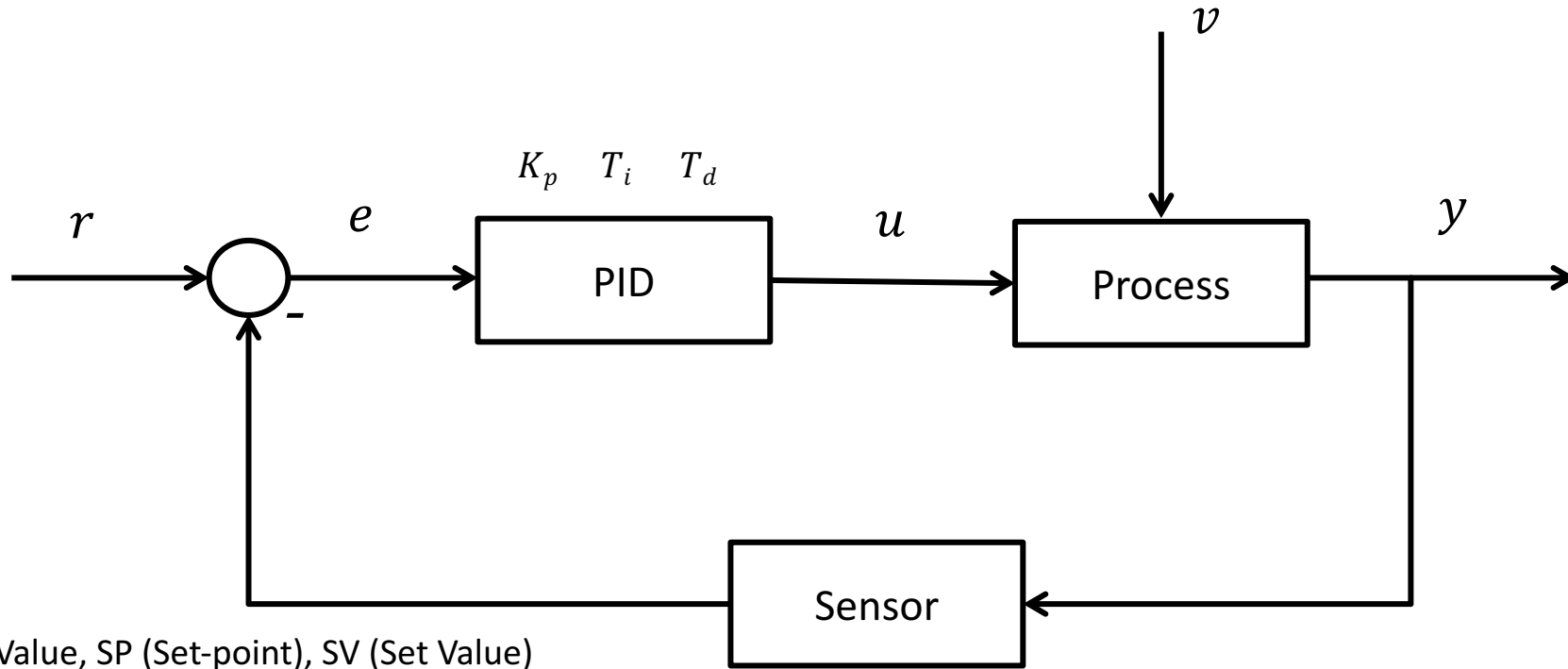
PID Control

Hans-Petter Halvorsen, M.Sc.

Feedback (PID) Control



Feedback (PID) Control System



r – Reference Value, SP (Set-point), 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

K_p, T_i, T_d – PID parameters

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

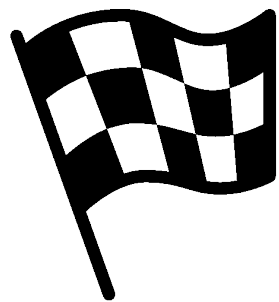
y is the Process value, i.e., the Measured value

Tuning Parameters:

K_p Proportional Gain

T_i Integral Time [sec.]

T_d Derivative Time [sec.]





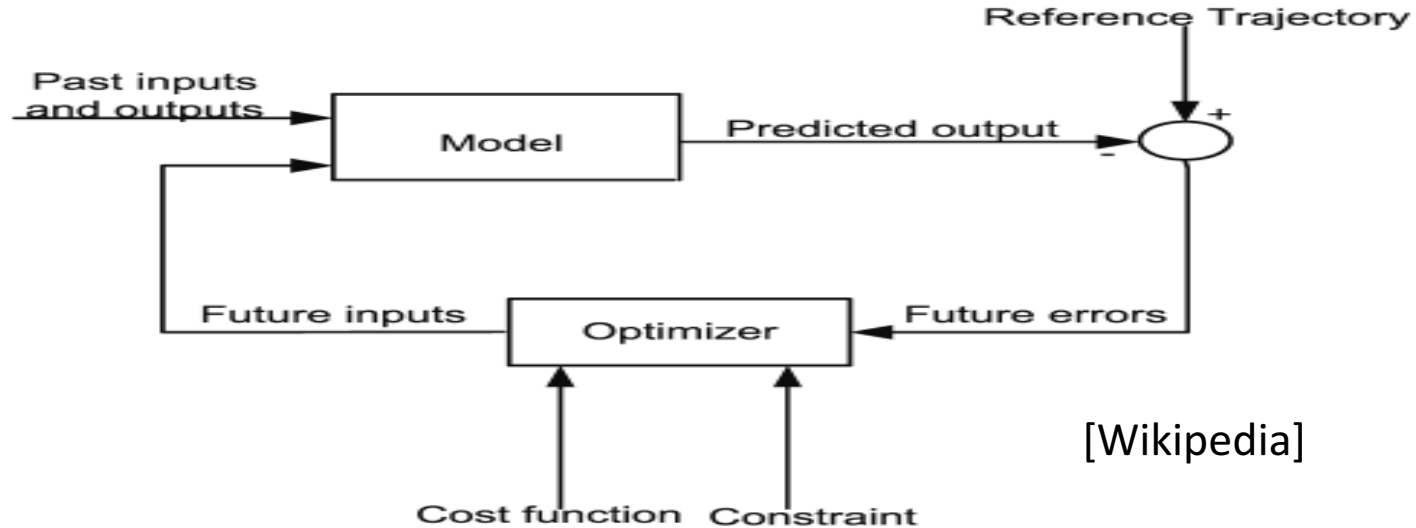
Model Predictive Control

Hans-Petter Halvorsen, M.Sc.

Model Predictive Control (MPC)

- Model predictive control (MPC) is an advanced method of process control that has been in use in the process industries since the 1980s.
- Model Predictive Control (MPC) is a multivariable control algorithm.
- Model predictive controllers rely on dynamic models of the process, most often linear empirical models obtained by system identification.
- MPC is based on iterative, finite-horizon optimization of a plant model.
- This is achieved by optimizing a finite time-horizon, but only implementing the current timeslot. MPC has the ability to anticipate future events and can take control actions accordingly.

Model Predictive Control - MPC

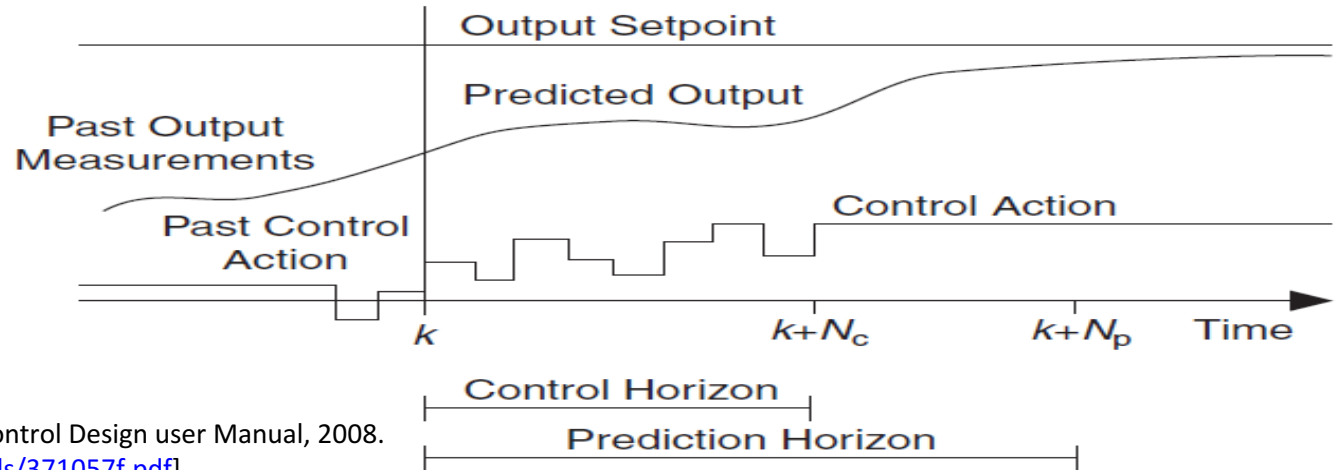


- Traditional feedback controllers operate by adjusting control action in response to a change in the output set-point of a system.
- Model predictive control (MPC) is a technique that focuses on constructing controllers that can adjust the control action before a change in the output set-point actually occurs.
- This predictive ability, when combined with traditional feedback operation, enables a controller to make adjustments that are smoother and closer to the optimal control action values.

Prediction

Prediction horizon (N_p) - The number of samples in the future the MPC controller predicts the plant output.

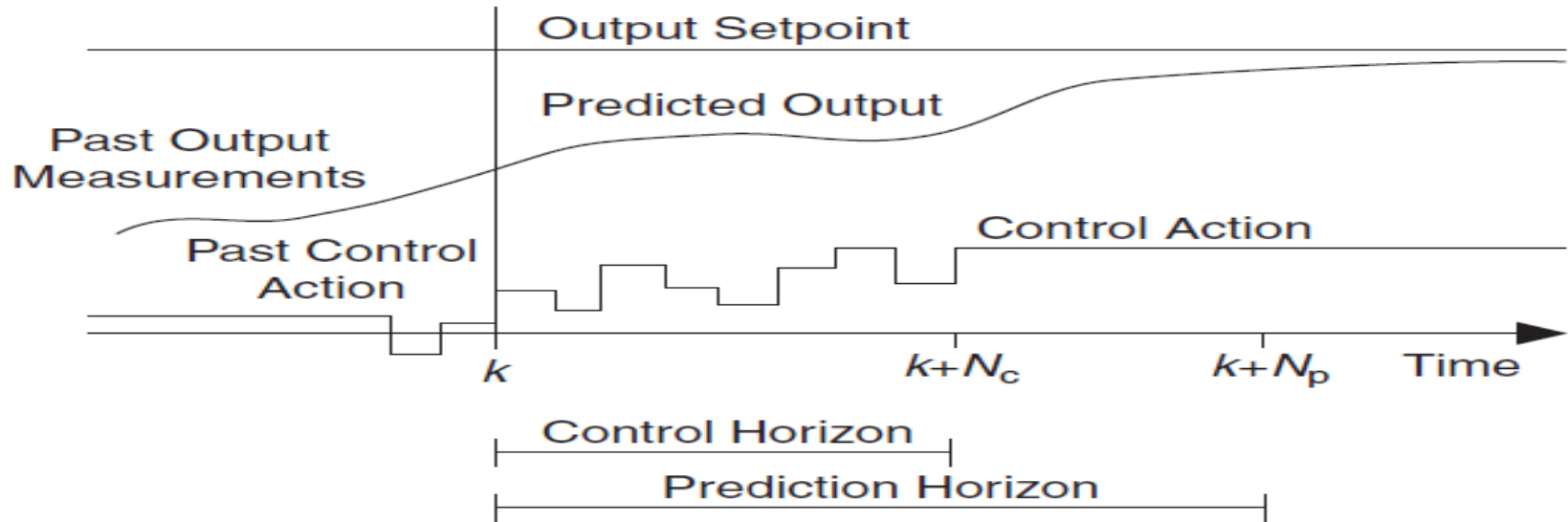
Control horizon (N_c) – The number of samples within the prediction horizon where the MPC controller can affect the control action.



Note!

$$N_c \leq N_p$$

Model Predictive Control - MPC



[Figure: National Instruments, LabVIEW Control Design user Manual, 2008.]

- For time k the MPC controller predicts the plant output for time $k + N_p$.
- The first input in the optimal sequence is then sent into the plant, and the entire calculation is repeated at subsequent control intervals. For each iteration the prediction horizon is moving forward in time and the MPC controller again predicts the plant output.

MPC

An MPC algorithm consists of:

- A Model of the process
- A Cost function
- Constraints

The Model

- The main drawback with MPC is that a model for the process, i.e., a model which describes the input to output behavior of the process, is needed.
- Mechanistic models derived from conservation laws can be used.
- Usually, however in practice simply data-driven linear models are used.

In MPC it is assumed that the model is a discrete state-space model of the form:

$$x_{k+1} = Ax_k + Bu_k$$

$$y_k = Cx_k + Du_k$$

The Cost Function

The cost function often used in MPC is like this (a linear quadratic function):

$$J = \sum_{k=0}^{N_p} (\hat{y} - r)^T Q (\hat{y} - r) + \sum_{k=0}^{N_p} \Delta u^T R \Delta u$$

Where:

N_p – Prediction horizon

r – Set-point

\hat{y} – Predicted process output

Δu – Predicted change in control value, $\Delta u_k = u_k - u_{k-1}$

Q – Output error weight matrix

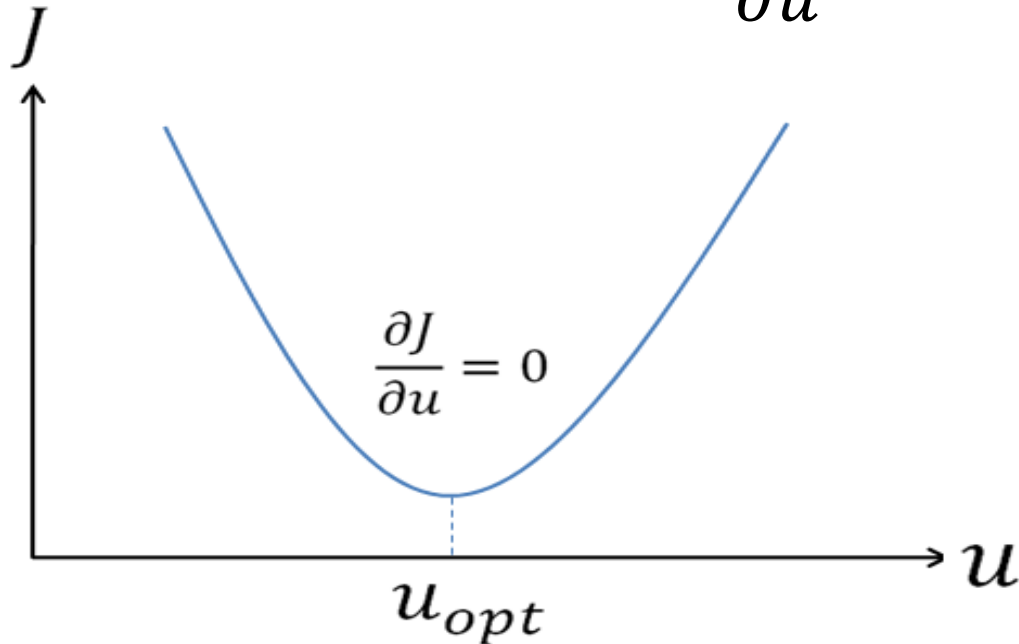
R – Control weight matrix

So the basic problem is to solve:

$$\frac{\partial J}{\partial u} = 0$$

The Cost Function

So the basic problem is to solve: $\frac{\partial J}{\partial u} = 0$



By solving this we get the future optimal control.

Solving $\frac{\partial J}{\partial u} = 0$ is quite complex and will not be part of this tutorial, but in the figure below we see an illustration of the problem.

E.g, LabVIEW, MATLAB have built-in functions and algorithms

Constraints

- All physical systems have constraints.
- We have physical constraints like actuator limits, etc. and we have safety constraints like temperature and pressure limits.
- Finally we have performance constraints like overshoot, etc.

In MPC you normally define these constraints:

Constraints in the outputs:

$$y_{min} \leq y \leq y_{max}$$

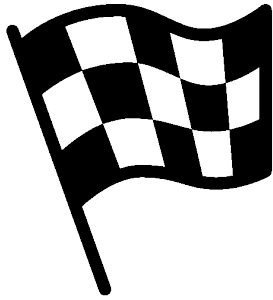
Constraints in the inputs:

$$\Delta u_{min} \leq \Delta u \leq \Delta u_{max}$$

$$u_{min} \leq u \leq u_{max}$$

Note! $\Delta u_k = u_k - u_{k-1}$

The MPC controller takes all these constraints into consideration when calculating the future controls.





MPC in LabVIEW

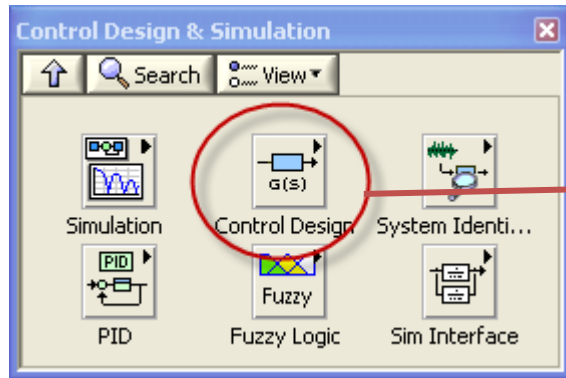
Hans-Petter Halvorsen, M.Sc.

MPC in LabVIEW

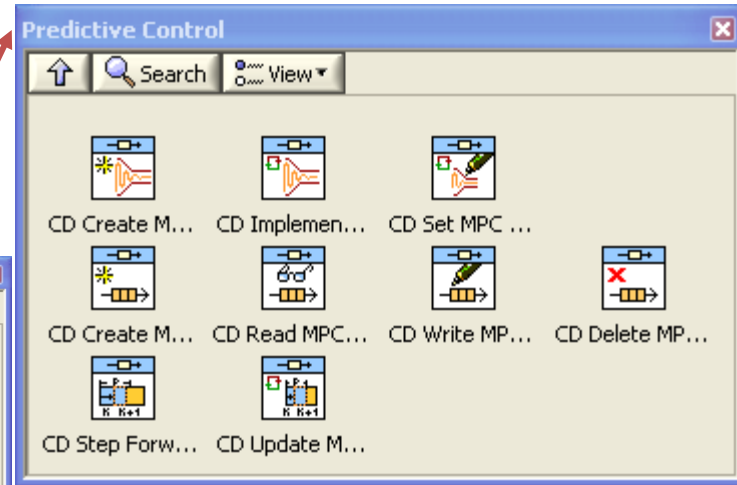
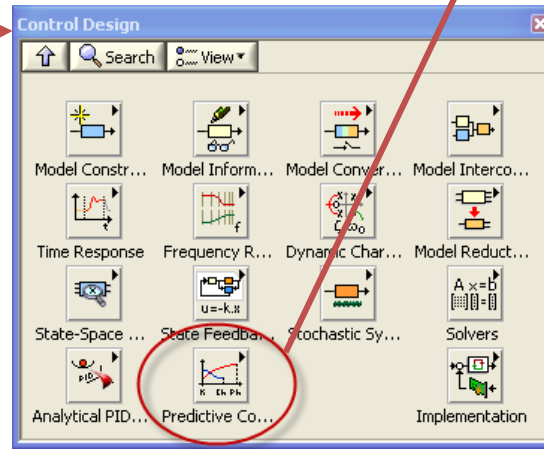
The MPC functionality in LabVIEW is part of the “Control Design and Simulation Module”

The “Predictive Control” palette in LabVIEW:

The “Control Design and Simulation” palette in LabVIEW:

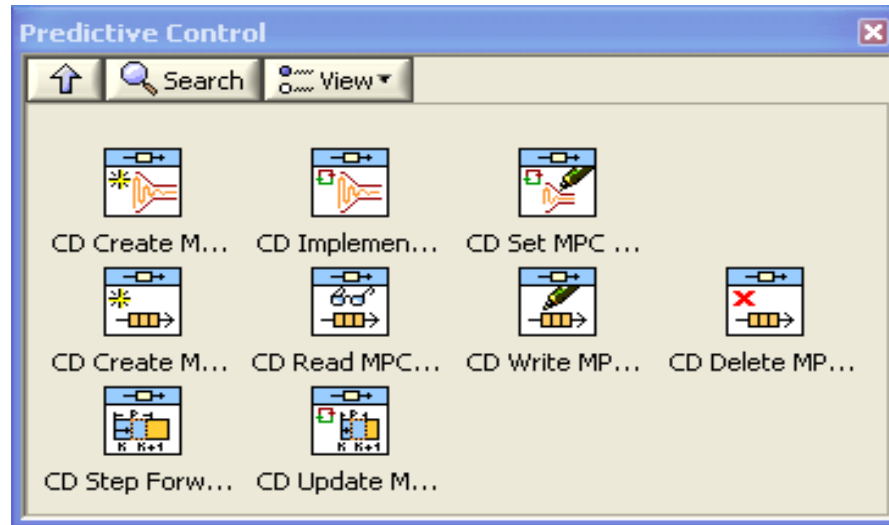


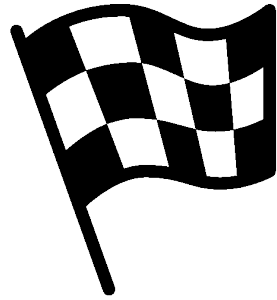
The “Control Design” palette in LabVIEW:



MPC in LabVIEW

- You use the “**CD Create MPC Controller**” VI to create an MPC controller. This VI bases the MPC controller on a state-space model of the plant that you provide.
- The “**CD Implement MPC Controller**” is used to calculate the control values for each sampling time and is normally implemented in a loop, e.g., a While Loop.







MPC Example 1

Hans-Petter Halvorsen, M.Sc.



1. order System

Given the following system:

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

Where

T is the time constant for the system

K is, e.g., the pump gain

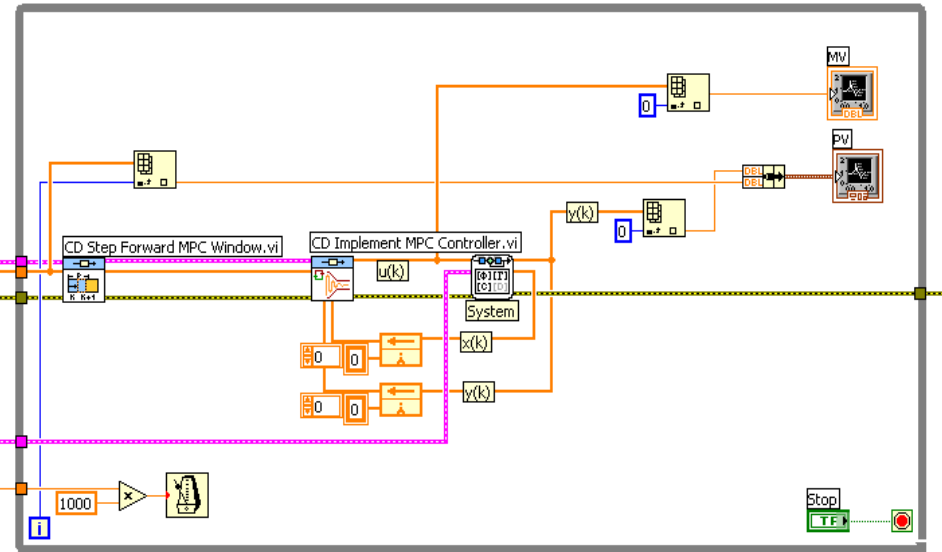
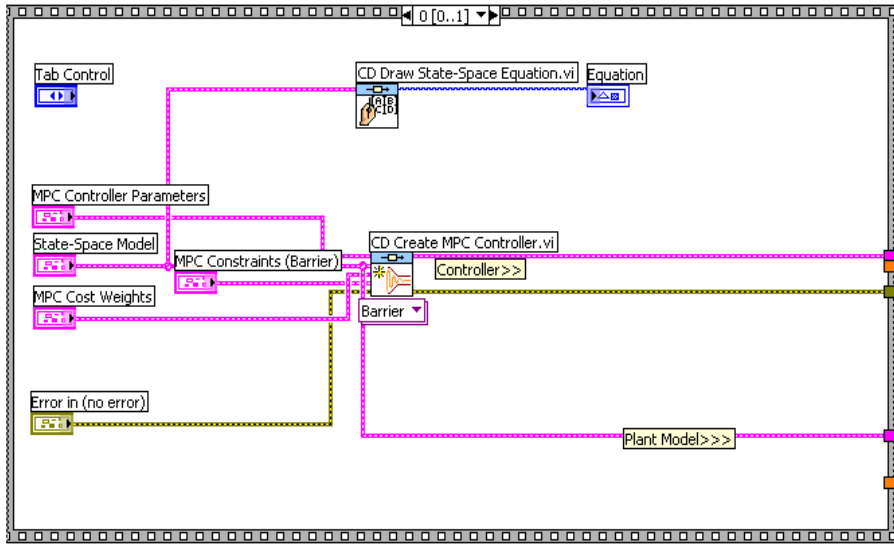
We set $T = 5s$ and $K = 2$ in this Example

MPC Example - Simple Model.vi Block Diagram

File Edit View Project Operate Tools Window Help

13pt Application Font

Search



System Profiles Constraints and Weighting

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

T = 5

K = 2

State-Space Model

Model name:

Sampling Time: 500

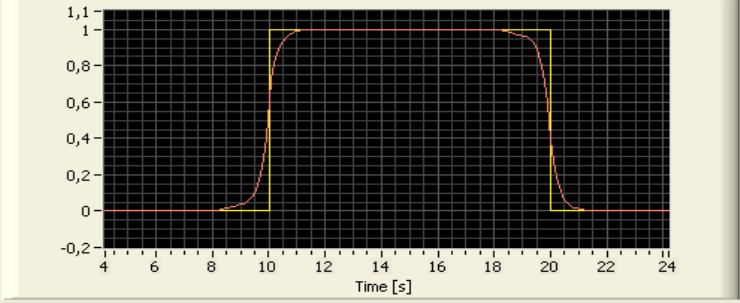
A				B			
0	-0,2	0	0	0	2	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
C				D			
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Equation

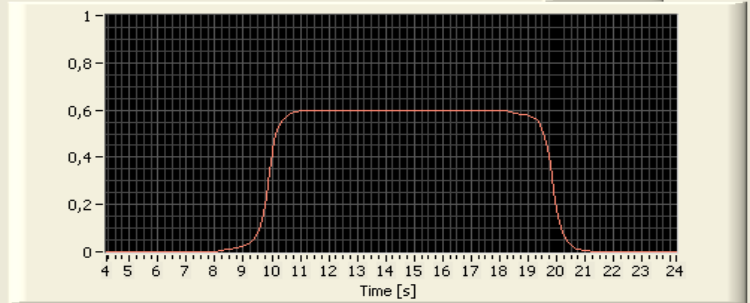
$$x(k+1) = \begin{bmatrix} -0,2 \\ 1 \end{bmatrix} x(k) + \begin{bmatrix} 2 \\ 0 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \end{bmatrix} u(k)$$

y 0,00
r 0,00

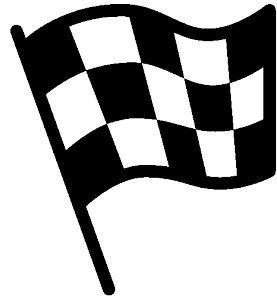


u 0,00



Stop

DEMO





MPC Example 2

Hans-Petter Halvorsen, M.Sc.

1. order System with Time Delay



Given the following system:

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

Where

T is the time constant for the system

K is, e.g., the pump gain

τ is the time delay

We set $T = 5s$, $K = 2$ and $\tau = 3s$ in this example

Transfer Function System Profiles Constraints and Weighting

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

$T = 5s, K = 2$ and $\tau = 3s$

$$H(s) = \frac{x(s)}{u(s)} = \frac{10}{5s + 1} e^{-3s}$$

Static Gain

10,0000

Time Constant

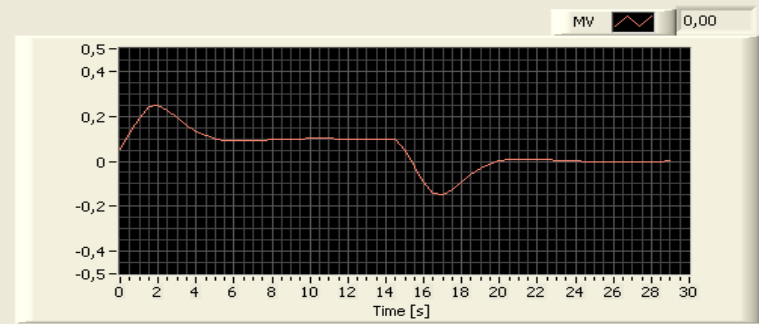
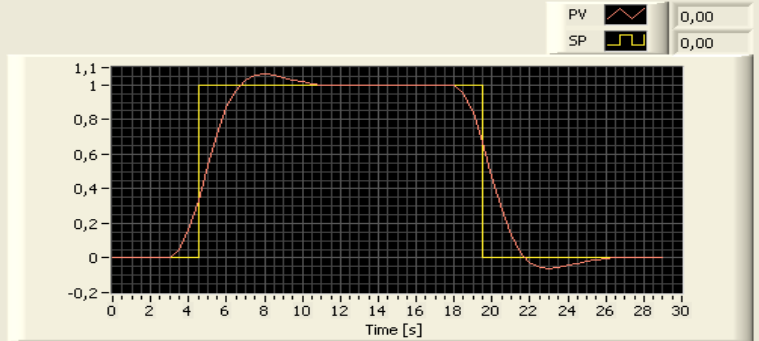
5,0000

Delay

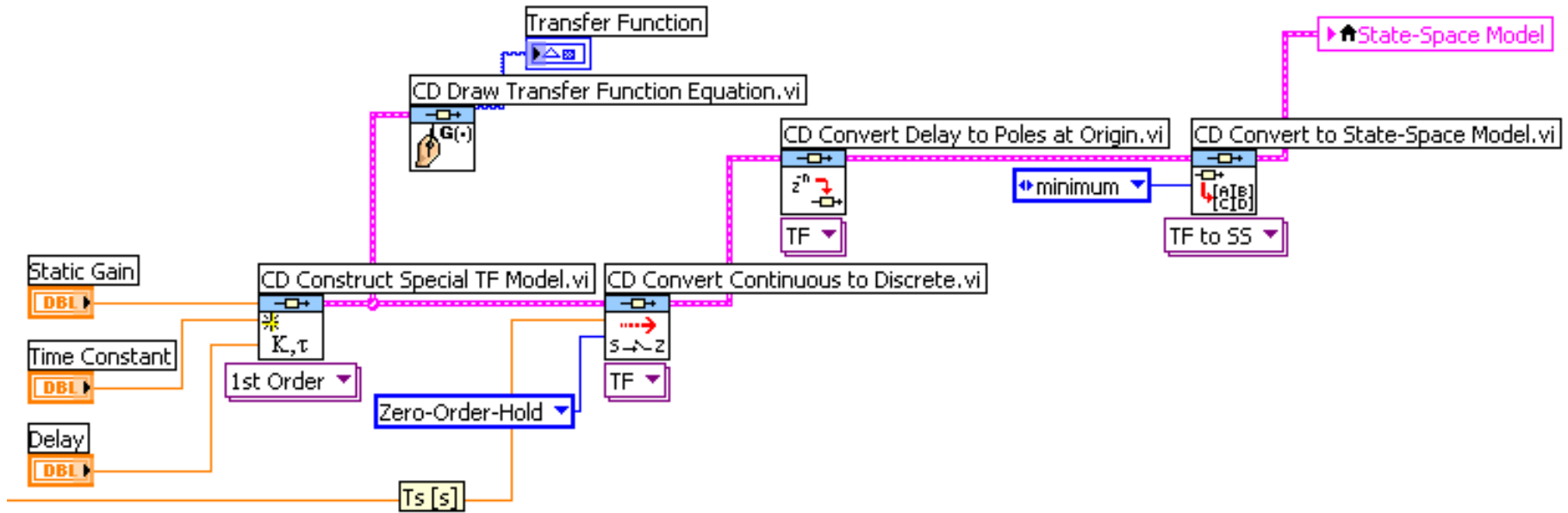
3,00

Transfer Function

$$e^{-3s} \frac{10}{5s + 1}$$

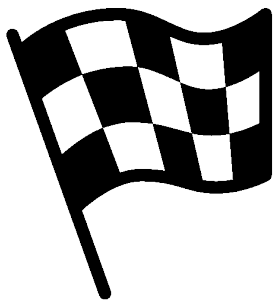


Stop



- We can use the “CD Construct Special TF Model.vi” in order to create the transfer function.
- Then we use miscellaneous functions in order to end up with a discrete state-space model that handles the time delay (additional states are added).

DEMO

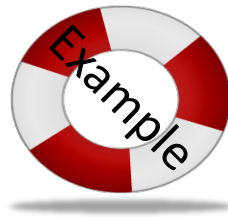




MPC Example 3

Air Heater System

Hans-Petter Halvorsen, M.Sc.



Control Air Heater Process using MPC

- The Air Heater is a small-scale laboratory process available at the university
- We will implement a MPC control system that controls the temperature for this system
- and compare to traditional PID control

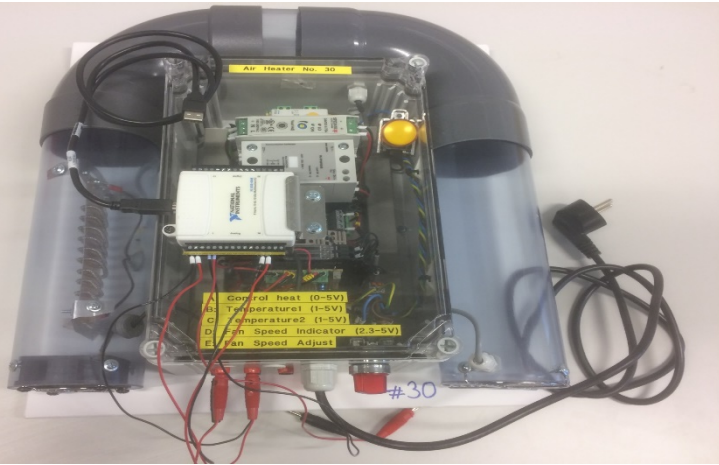
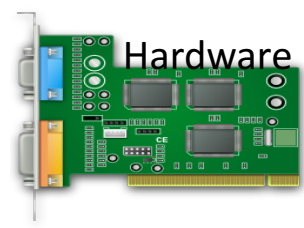


Air Heater No. 30

- A: Control heat (0-5V)
- B: Temperature1 (1-5V)
- C: Temperature2 (1-5V)
- D: Fan Speed Indicator (2.3-5V)
- E: Fan Speed Adjust

#30

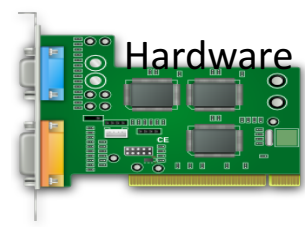
Air Heater



- **Heater:** The air is heated by an electrical heater. The supplied power is controlled by an external voltage signal in the range **0 - 5 V** (min power, max power).
- **Temperature sensors:** A Pt100 temperature sensor. The range is **1 - 5 V**, and this voltage range corresponds to the temperature range **20 - 50°C** (with a linear relation).



Air Heater Mathematical Model



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

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

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

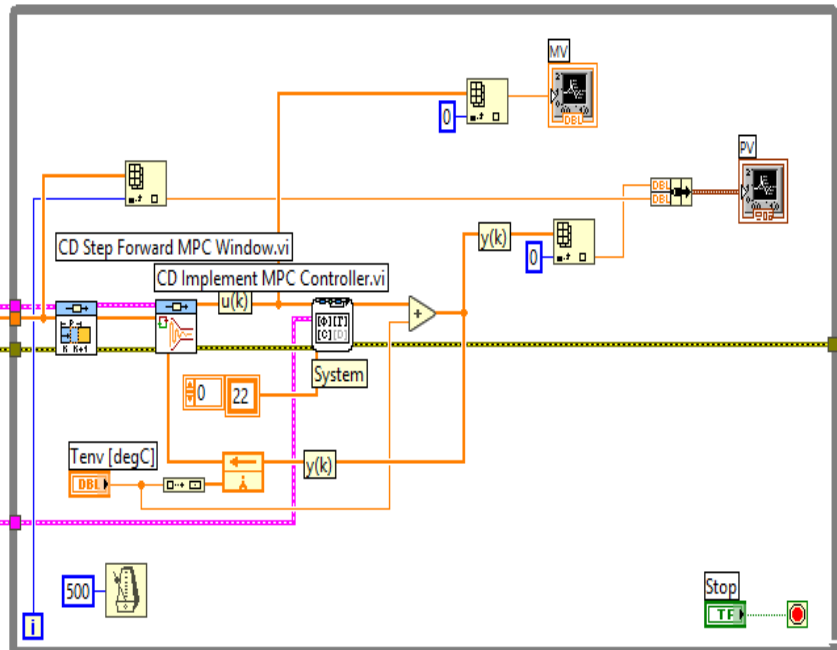
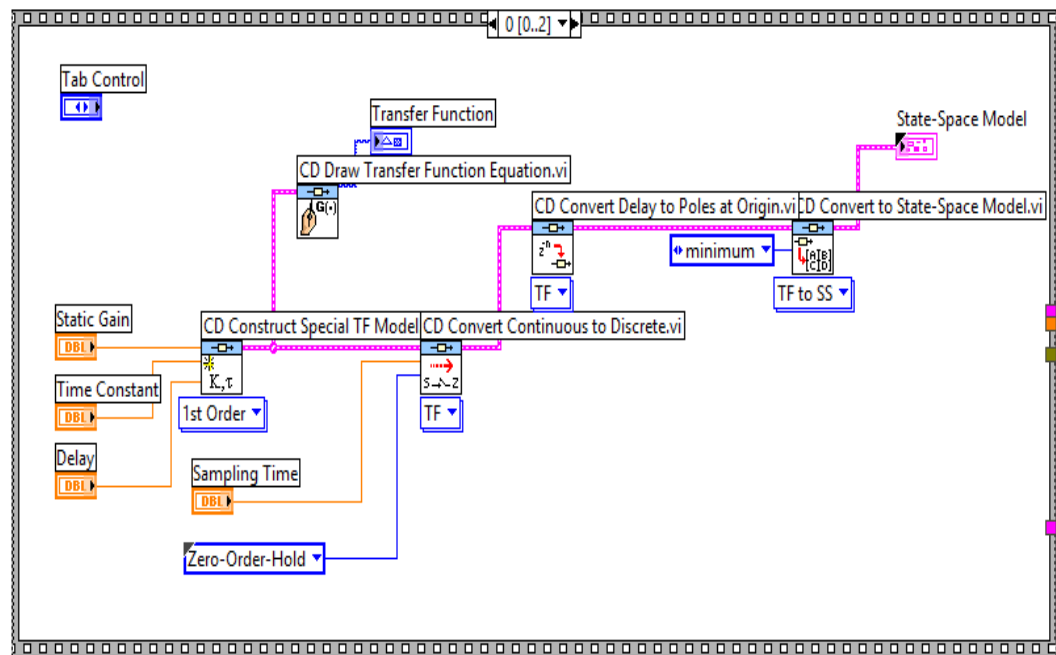
Where:

- T_{out} is the air temperature at the tube outlet
- $u [V]$ is the control signal to the heater
- $\theta_t [s]$ is the time-constant
- $K_h [deg C / V]$ is the heater gain
- $\theta_d [s]$ is the time-delay representing air transportation and sluggishness in the heater
- T_{env} is the environmental (room) temperature. It is the temperature in the outlet air of the air tube when the control signal to the heater has been set to zero for relatively long time (some minutes)

We will use the following values in the Simulations:

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

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



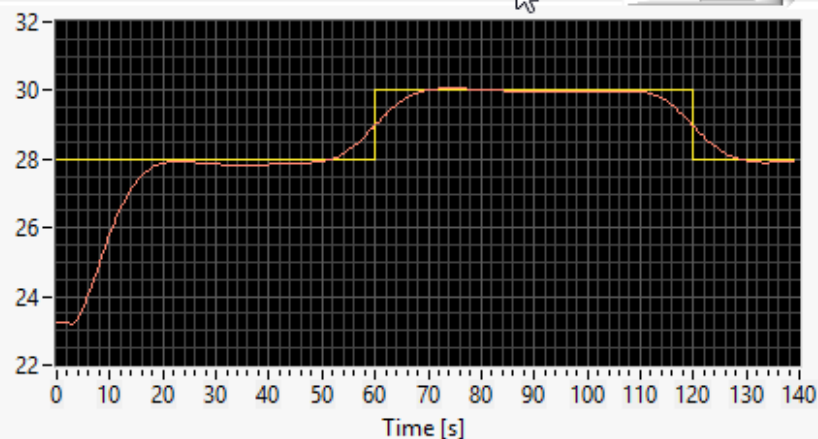


Set-Point Profile

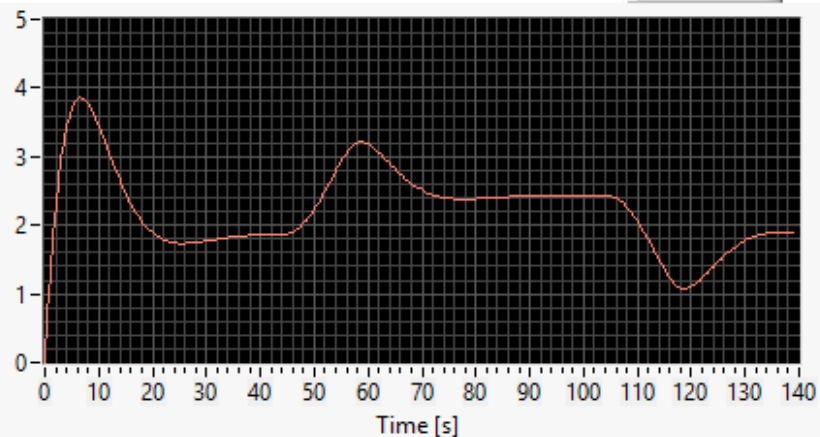
time (s)	Reference
0,000	28,00
60,000	30,00
120,000	28,00

y 27,94

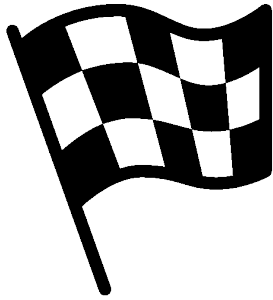
r 28,00



u 1,90



DEMO





PID vs. MPC

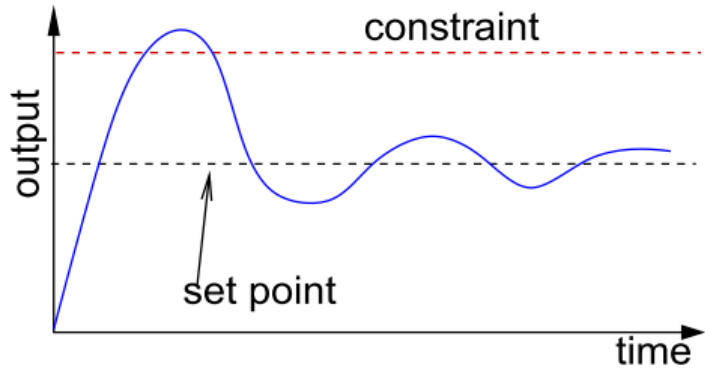
Hans-Petter Halvorsen, M.Sc.

PID vs. MPC

- MPC is often used in addition to traditional control like PID – not as a replacement.
- In large plants MPC is not a replacement for traditional PID, but used in addition to PID controllers.
- PID controllers are used as single-loop controllers, while MPC is used as an overall system.
- PID handles only a single input and a single output (SISO systems), while MPC is a more advanced method of process control used for MIMO systems (Multiple Inputs, multiple Outputs).

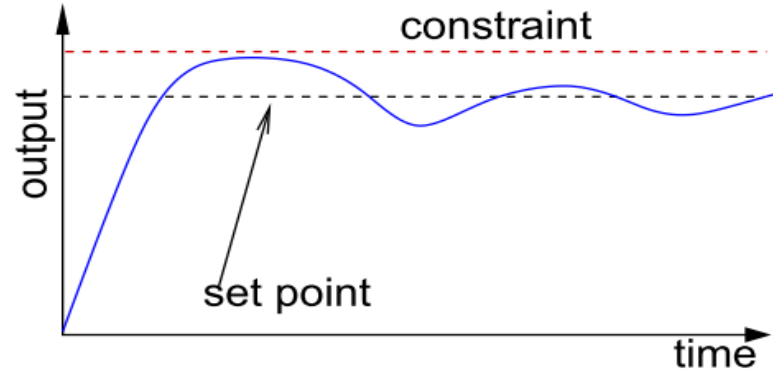
PID vs. MPC

Traditional Control (PID)



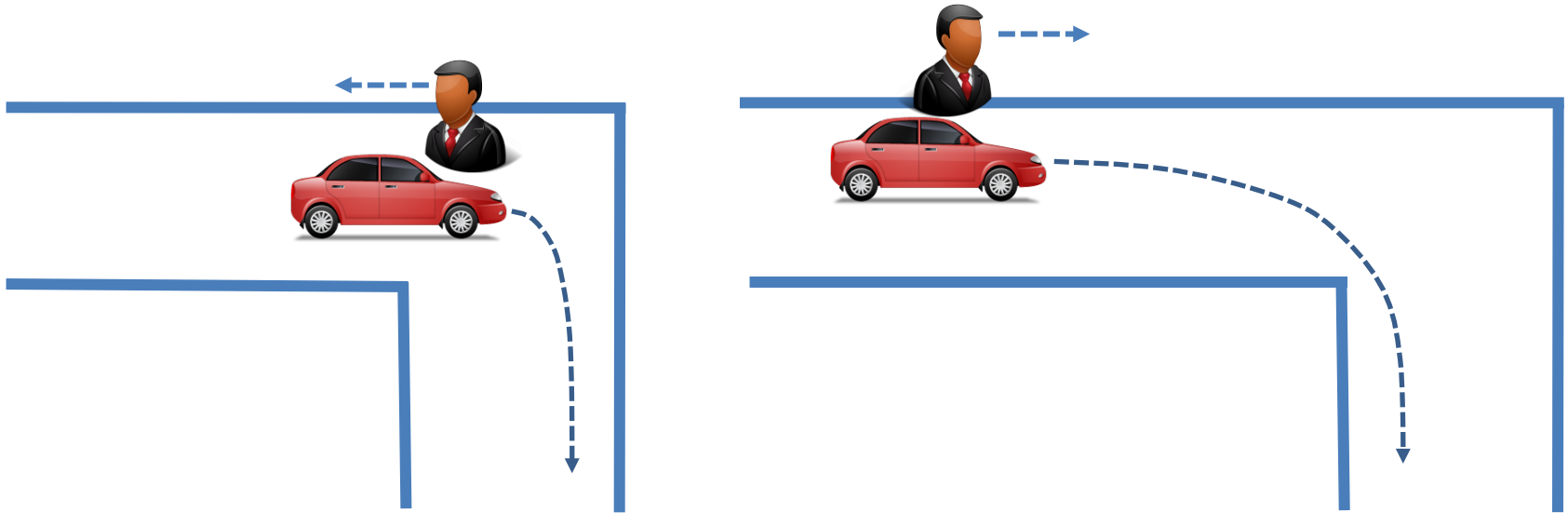
- No knowledge about constraints
- Set-point far from constraints
- Not optimal process operation
- SISO systems
- A mathematical model is not needed

MPC



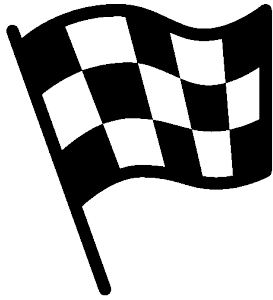
- Constraints included in the design
- Set-point can be closer to constraints
- Improved process operation
- MIMO systems
- A mathematical model is needed

PID vs. MPC



PID can be compared to driving a car where you sit the wrong way looking backwards only. You don't start to turn to right before you are inside the curve

MPC: When you sit the correct way, you are able to see the curve ahead in good time. Then you can start to plan and react to the curve before you reach it.



Hans-Petter Halvorsen, M.Sc.



University College of Southeast Norway

www.usn.no

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

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

