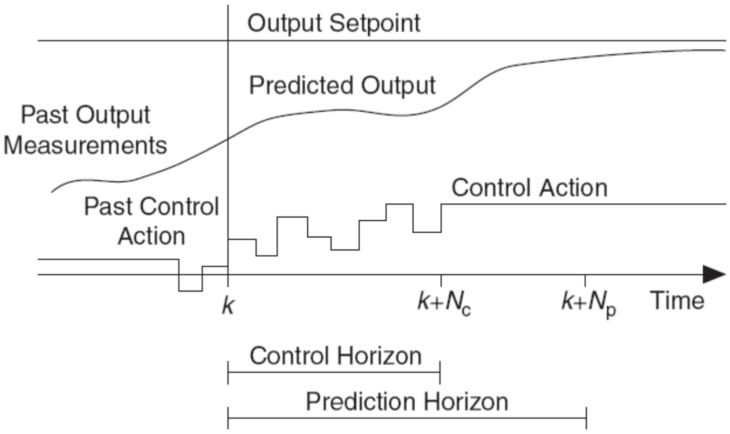


Figure 22.1: How MPC works

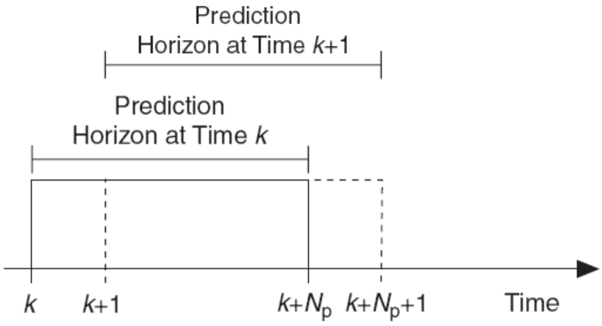


Figure 22.2: How the prediction horizon moves at each sample time k

Example 22.1 ***MPC control of a real air heater***

Figure 22.3 shows a lab process consisting of an heated air tube where the air temperature (at temperature sensor 1) has been controlled with both MPC and – for comparison – a PID controller in a number of different cases described in the following. The control system is implemented on a PC with LabVIEW. The MPC controller and the Advanced PID controller in LabVIEW Control Design and Simulation Toolkit are used. The sampling time is 0.5 s.

Mathematical modeling

The MPC controller in LabVIEW requires a process model in the form of a discrete-time state-space model. Although advanced system identification

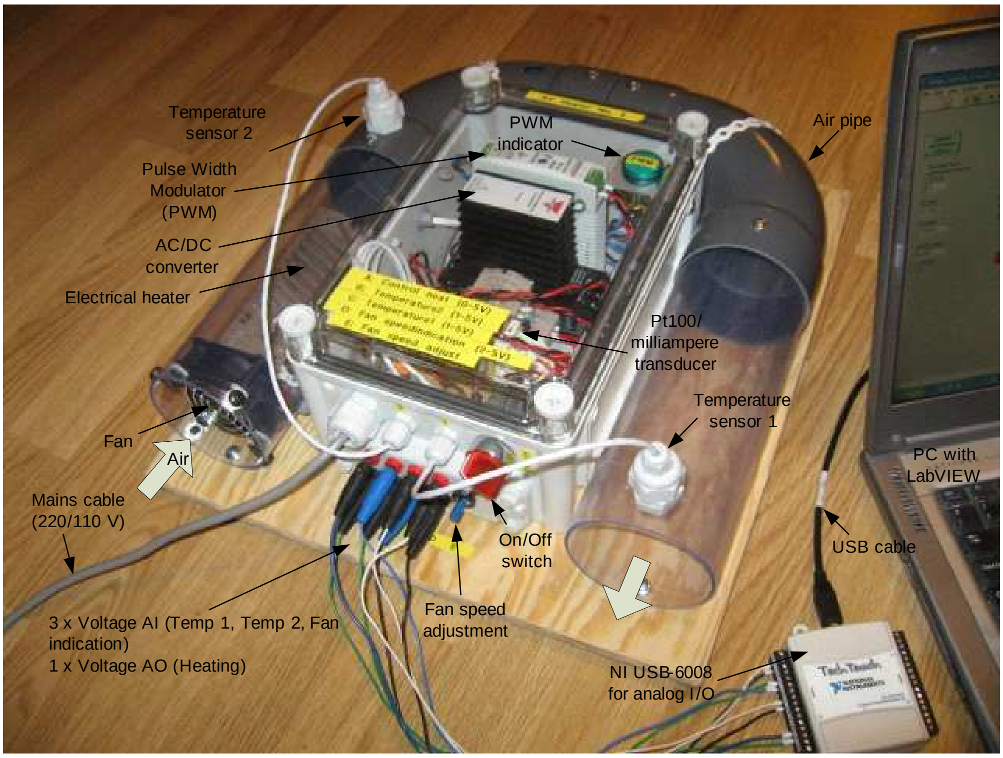


Figure 22.3: Example 22.1: A lab process consisting of a heated air tube where the air temperature will be controlled.

functions are available in LabVIEW System Identification Toolkit, a simple, manual model development is accomplished in this example: The model is estimated manually from the step response of the process.

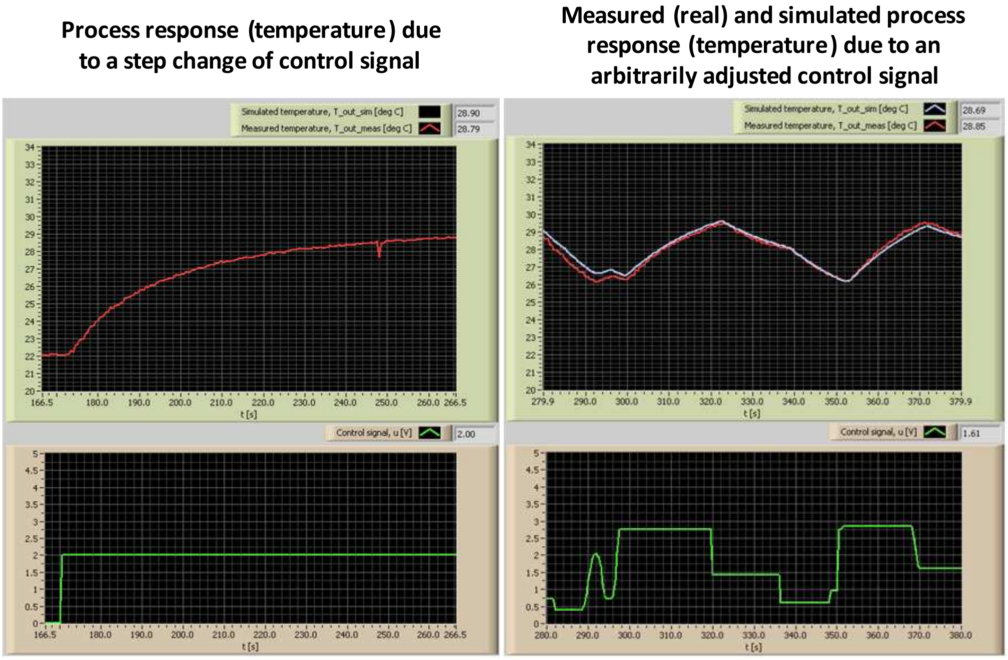
Figure 22.4 shows to the left the process step response, i.e. the response in the temperature due to a step change of the control signal to the heater. This response indicates that a proper model is a "time-constant with time-delay" transfer function with the following parameters:

$$\text{Gain: } K = 3.5\ ^{\text{o}}\text{C/V} \tag{22.8}$$

$$\text{Time-constant: } T = 22 \text{ s} \tag{22.9}$$

$$\text{Time-delay: } \tau = 2 \text{ s} \tag{22.10}$$

To validate the model, and to possibly fine-tune model parameters, a simulator based on the estimated transfer function is run in real time and in parallel with the real process. The simulator and the real process is of course excited by the same control signal, which is an arbitrarily adjusted



Process response (temperature) due to a step change of control signal

Measured (real) and simulated process response (temperature) due to an arbitrarily adjusted control signal

Figure 22.4: Example 22.1: Left: Process step response (step in control signal) used for model estimation. Right: Process response after arbitrarily adjusted control signal used for model validation

signal, see the plots to the right in Figure 22.4. It is clear that the model is quite good (accurate).

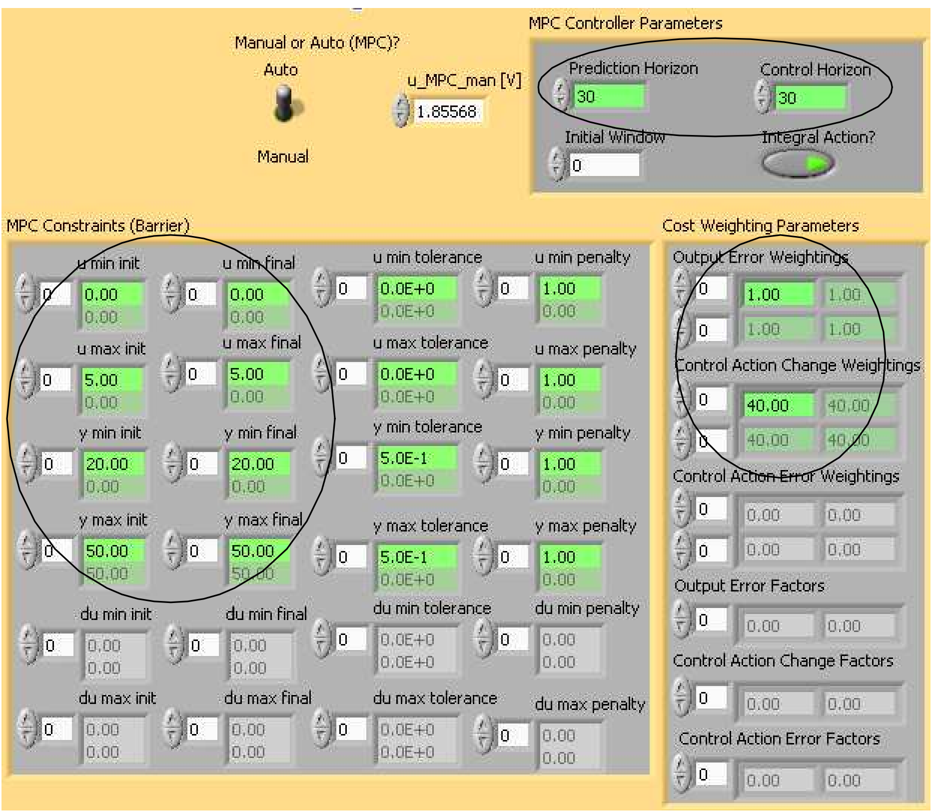
To obtain a discrete-time state-space model, model conversion functions in LabVIEW are used. (The state-space model is however not shown here.)

Settings of MPC and PID controller

MPC settings

Figure 22.5 shows the settings of various MPC parameters:

- **Horizons:** The prediction and control horizons are set not so different from process time-constant. They are set to 30, which corresponds to 15 sec since the the sampling time is 0.5 sec. (The process time-constant is 22 sec.)
- **Weightings:** The Output Error Weighting is set to 1. The Control Action Change Weighting is set by trial-and-error on real system to 40 (it can also be adjusted on the simulator, of course). Small



u_MPC_man [V] 1.85568

MPC Controller Parameters
Prediction Horizon 30
Control Horizon 30
Initial Window 0

MPC Constraints (Barrier)
u min init 0 0.00 0.00
u min final 0 0.00 0.00
u min tolerance 0 0.0E+0 0.0E+0
u min penalty 0 1.00 0.00
u max init 0 5.00 0.00
u max final 0 5.00 0.00
u max tolerance 0 0.0E+0 0.0E+0
u max penalty 0 1.00 0.00
y min init 0 20.00 0.00
y min final 0 20.00 0.00
y min tolerance 0 5.0E-1 0.0E+0
y min penalty 0 1.00 0.00
y max init 0 50.00 50.00
y max final 0 50.00 50.00
y max tolerance 0 5.0E-1 0.0E+0
y max penalty 0 1.00 0.00
du min init 0 0.00 0.00
du min final 0 0.00 0.00
du min tolerance 0 0.0E+0 0.0E+0
du min penalty 0 0.00 0.00
du max init 0 0.00 0.00
du max final 0 0.00 0.00
du max tolerance 0 0.0E+0 0.0E+0
du max penalty 0 0.00 0.00

Cost Weighting Parameters
Output Error Weightings 0 0 1.00 1.00 1.00 1.00
Control Action Change Weightings 0 0 40.00 40.00 40.00 40.00
Control Action Error Weightings 0 0 0.00 0.00 0.00 0.00
Output Error Factors 0 0.00 0.00
Control Action Change Factors 0 0.00 0.00
Control Action Error Factors 0 0.00 0.00

Figure 22.5: Example 22.1: MPC parameter settings

weightings gives fast, abrupt control. Large weighting gives sluggish control.

- **Constraints:** The constraints of the control signal u and the process output (measurement) y are set to the physical limits. The control signal range is 0 – 5 V, and the temperature measurement range is 20 – 50 $^{\mathrm{o}}$C.

PID settings

The PID controller is used as a PI controller with the following settings:

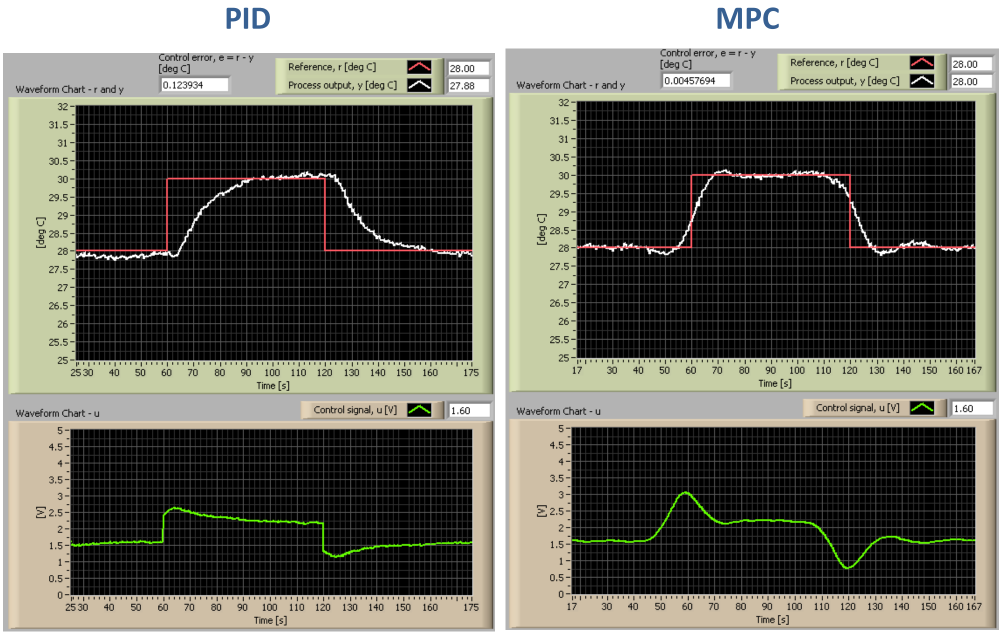
$$K_p = 0.42 \tag{22.11}$$

$$T_i = 18 \text{ s} \tag{22.12}$$

(The controller is tuned with Skogestad's method with the closed-loop time-constant set to – somewhat arbitrarily – 10 sec.)

Setpoint tracking with future setpoint step

Figure 22.6 shows the responses in the temperature and the control signal with PI control and MPC control after step changes of setpoint. MPC control is much better then PID, because the MPC controller plans the control properly by taking into account the future setpoint changes. Observe that MPC starts changing control *ahead* of the setpoint change, while the PI controller changes the control action *after* setpoint is changed.



PID:
Control error, e = r - y [deg C] 0.123934
Reference, r [deg C] 28.00
Process output, y [deg C] 27.88
Control signal, u [V] 1.60

MPC:
Control error, e = r - y [deg C] 0.00457694
Reference, r [deg C] 28.00
Process output, y [deg C] 28.00
Control signal, u [V] 1.60

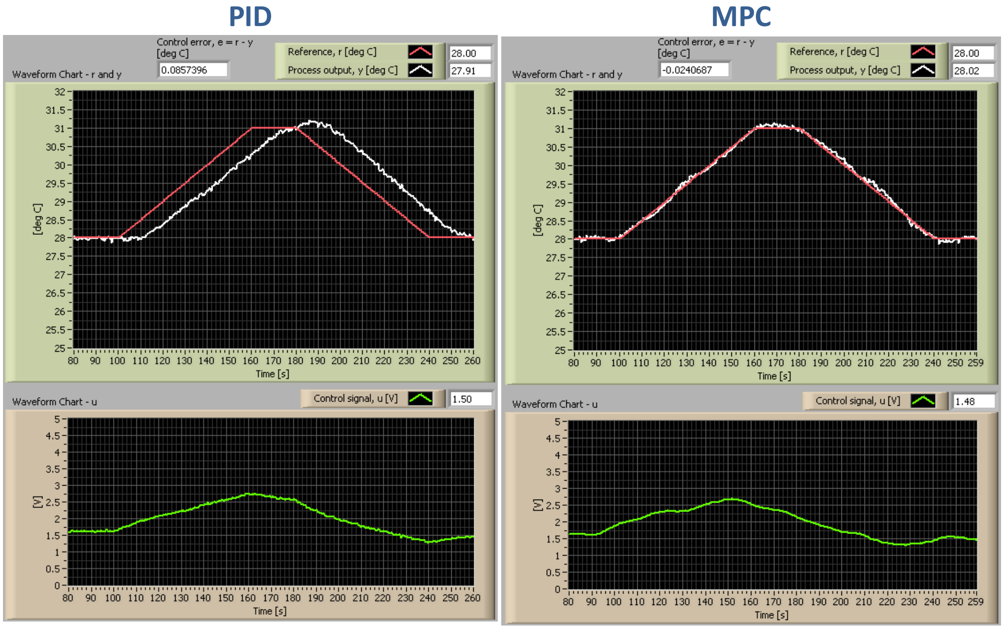
Figure 22.6: Example 22.1: Responses after step changes of the setpoint

Setpoint tracking with future setpoint ramp

Figure 22.7 shows the responses with ramped changes of the setpoint. The MPC control gives excellent control, with almost zero control error, while the PI controller gives a clear non-zero control error. With a known future setpoint trajectory, the MPC is capable of giving superior control.

Propagation of measurement noise

Figure 22.8 shows how the measurement noise is propagated through the PID controller and MPC controller. The noise is more smoothened through the MPC (less propagation of noise), which is because more samples of the measurement signal are used in calculation of the control



PID:
Control error, e = r - y [deg C]: 0.0857396
Reference, r [deg C]: 28.00
Process output, y [deg C]: 27.91
Control signal, u [V]: 1.50

MPC:
Control error, e = r - y [deg C]: -0.0240687
Reference, r [deg C]: 28.00
Process output, y [deg C]: 28.02
Control signal, u [V]: 1.48

Figure 22.7: Example 22.1: : Responses after ramped changes of the setpoint.

signal with the MPC controller than with the PID controller. In other words, there is more "averaging" with the MPC controller.

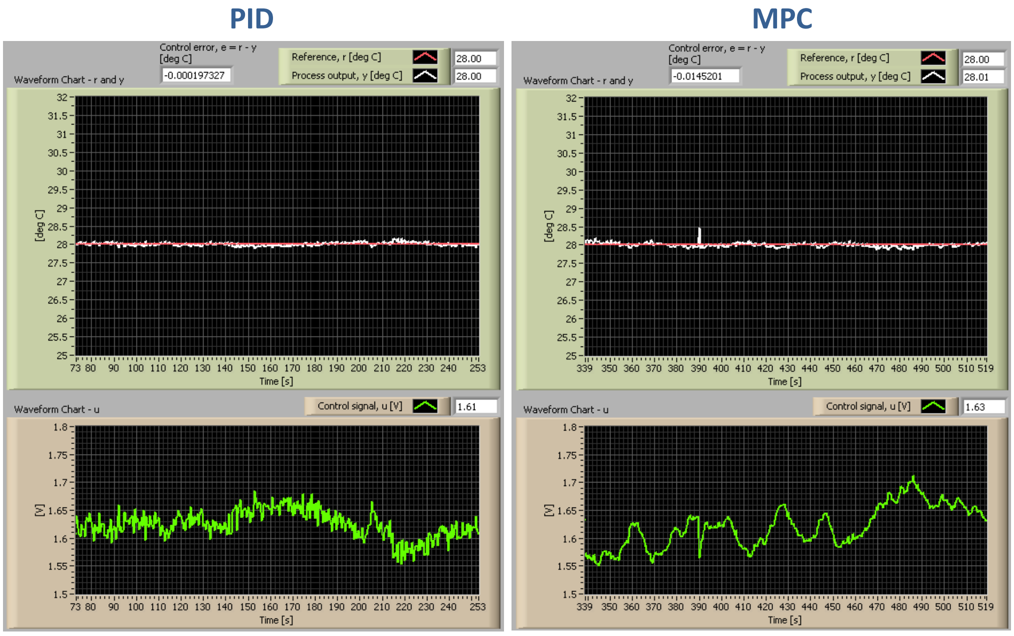
Constrained control

Figure 22.9 shows the response in the temperature with MPC control when a maximum constraint of 30 ^{o}C is set for the process output variable (temperature). This constraint is one of the MPC parameters which the user can set. Actually, a tolerance of 0.5 K is set for this constraint. The Barrier method, which is an alternative to the Dual method, is used to define the constraints in the LabVIEW MPC used in this experiment. (The Dual method may give oscillations at the constraint limit, while the Barrier method does not.)

As seen from the response in Figure 22.9 the maximum limit of 30 ^{o}C is maintained, with a margin below of about 0.5 K.

Changing the weight of incremental control signal

Figure 22.10 shows responses with MPC control with increased and decreased weight of the control signal increment. As expected, with increased weight (more "expensive" control increments) the control signal will vary less, causing the process ouput variable (temperature) to respond



PID:

Control error, e = r - y [deg C]: -0.000197327

Reference, r [deg C]: 28.00

Process output, y [deg C]: 28.00

Control signal, u [V]: 1.61

MPC:

Control error, e = r - y [deg C]: -0.0145201

Reference, r [deg C]: 28.00

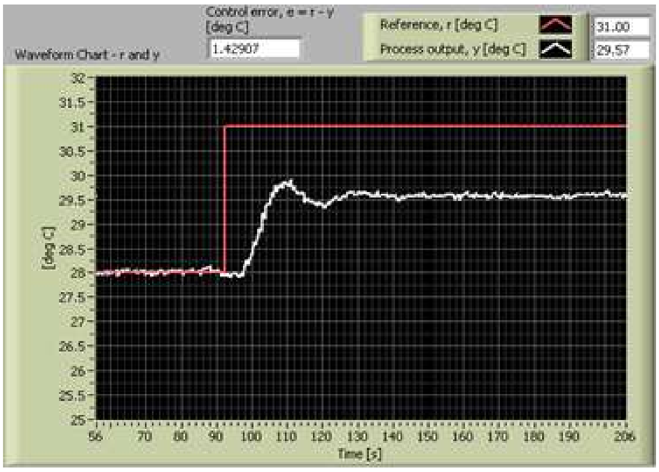
Process output, y [deg C]: 28.01

Control signal, u [V]: 1.63

Figure 22.8: Example 22.1: Measurement noise propagation through the PI controller and the MPC controller

more slowly, cf. the left plots in Figure 22.10. And with decreased weight (“cheaper” control increments) the control signal will vary more, causing the process ouput variable (temperature) to respond more quickly, cf. the right plots in Figure 22.10. If the weight is set very close to zero, the MPC controller acts almost like an On/off controller, which is denoted “dead-beat control”.

[End of Example 22.1]

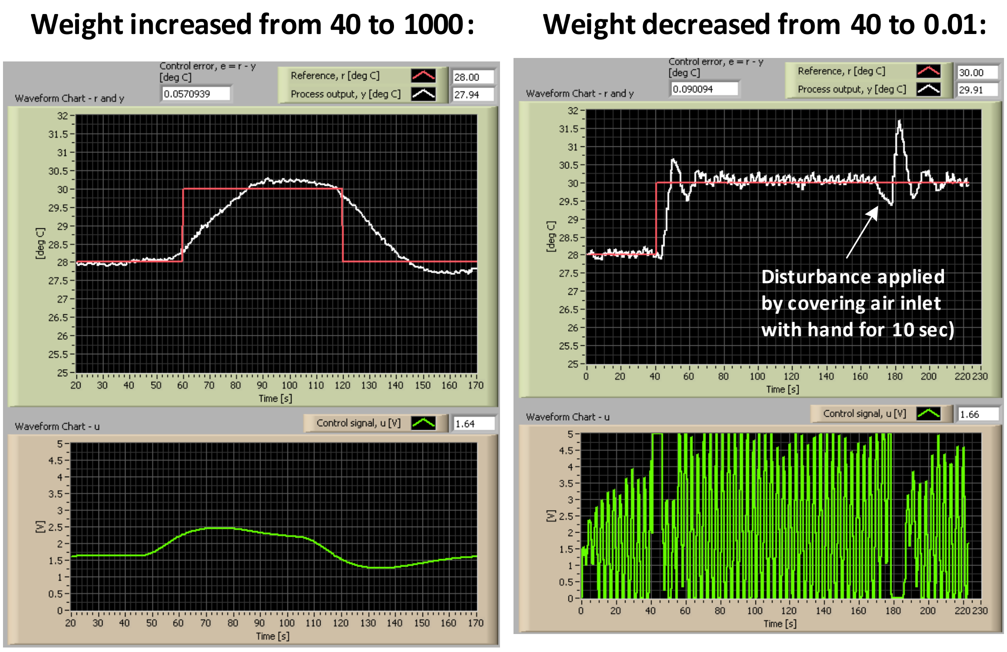


Control error, e = r - y [deg C]: 1.42907

Reference, r [deg C]: 31.00

Process output, y [deg C]: 29.57

Figure 22.9: Example 22.1: Setting the process measurement contraint to 30 °C.



Weight increased from 40 to 1000:

Control error, e = r - y [deg C]: 0.0570939

Reference, r [deg C]: 28.00

Process output, y [deg C]: 27.94

Control signal, u [V]: 1.64

Weight decreased from 40 to 0.01:

Control error, e = r - y [deg C]: 0.090094

Reference, r [deg C]: 30.00

Process output, y [deg C]: 29.91

Control signal, u [V]: 1.66

Figure 22.10: Example 22.1: Adjusting the weighting of incremental control signal to a relatively large value (left) and to a small value (right).