

how it varies: Both these assumptions are expressed with the following differential equations describing the augmentative state variable x_a :

$$\dot{x}_a(t) = 0 \quad (17.51)$$

This is the most common way to model the augmentative state.

- **Augmentative state has (almost) constant rate:** The corresponding differential equation is

$$\ddot{x}_a = 0 \quad (17.52)$$

or, in state space form, with $x_{a1} \equiv x_a$,

$$\dot{x}_{a1} = x_{a2} \quad (17.53)$$

$$\dot{x}_{a2} = 0 \quad (17.54)$$

where x_{a2} is another augmentative state variable.

Once you have defined the augmented model, you can design and implement the observer in the usual way. The observer estimates both the original states and the augmentative states.

The following example shows how the state augmentation can be done in a practical (simulated) application.

Example 17.3 *Observer for estimating level and flow*

Figure 17.7 shows a liquid tank with a level control system (PI controller) and an observer. (This system is also be used in Example 18.2 where a Kalman Filter is used in stead of an observer.) We will design an observer to estimate the outflow F_{out} . The level h is measured.

Mass balance of the liquid in the tank is (mass is ρAh)

$$\rho A_{tank} \dot{h}(t) = \rho K_p u - \rho F_{out}(t) \quad (17.55)$$

$$= \rho K_p u - \rho F_{out}(t) \quad (17.56)$$

After cancelling the density ρ the model is

$$\dot{h}(t) = \frac{1}{A_{tank}} [K_p u - F_{out}(t)] \quad (17.57)$$

We assume that we do not know how the outflow is actually varying, so we use the following augmentative model describing its behaviour:

$$\dot{F}_{out}(t) = 0 \quad (17.58)$$

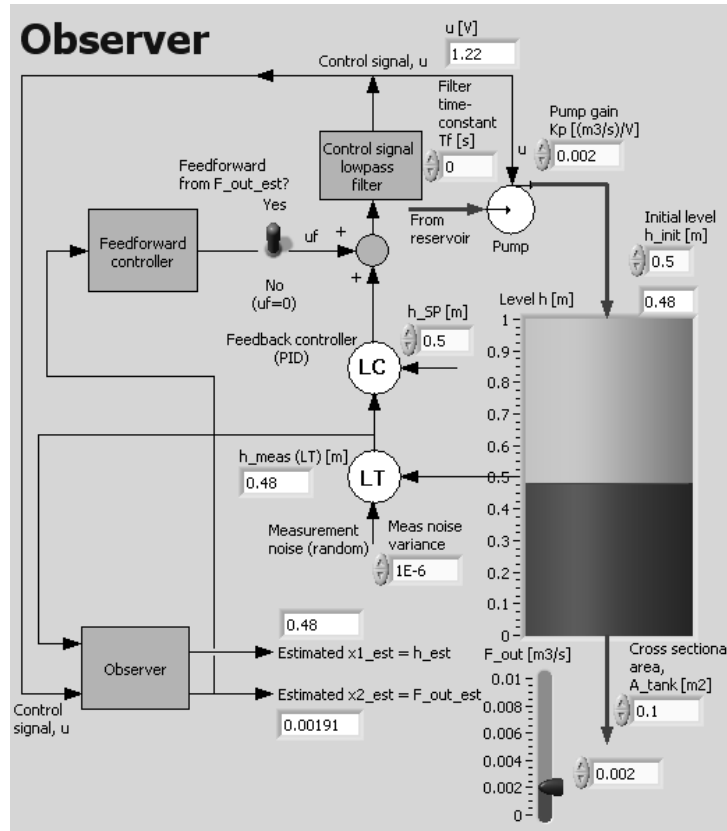


Figure 17.7: Example 17.3: Liquid tank with level control system and observer for estimation of outflow

The model of the system is given by (17.57) – (17.58). The parameter values of the tank are displayed (and can be adjusted) at the front panel, see Figure 17.7. The sampling time is

$$T_s = 0.1 \text{ s} \quad (17.59)$$

Although it is not strictly necessary, it is convenient to rename the state variables using standard names. So we define

$$x_1 = h \quad (17.60)$$

$$x_2 = F_{out} \quad (17.61)$$

The model (17.57) – (17.58) is now

$$\dot{x}_1(t) = \frac{1}{A_{tank}} [K_p u(t) - x_2(t)] \equiv f_1(\cdot) \quad (17.62)$$

$$\dot{x}_2(t) = 0 \equiv f_2(\cdot) \quad (17.63)$$

The measurement equation is

$$y = x_1 \quad (17.64)$$

The initial estimates are as follows:

$$x_{1p}(0) = x_1(0) = y(0) \text{ (from the sensor)} \quad (17.65)$$

$$x_{2p}(0) = 0 \text{ (assuming no information about initial value)} \quad (17.66)$$

The observer algorithm is, according to (17.50),

$$x_{1e}(t_{k+1}) = x_{1e}(t_k) + T_s [f_1(\cdot, t_k) + K_1 e] \quad (17.67)$$

$$= x_{1e}(t_k) + T_s \left\{ \frac{1}{A_{\text{tank}}} [K_p u(t_k) - x_{2e}(t_k)] + K_1 e \right\} \quad (17.68)$$

$$x_{2e}(t_{k+1}) = x_{2e}(t_k) + T_s [f_2(\cdot, t_k) + K_2 e] \quad (17.69)$$

$$x_{2e}(t_k) + T_s K_2 e \quad (17.70)$$

To calculate observer gain K we need a linearized process model on the form

$$\Delta \dot{x} = A_c \Delta x + B_c \Delta u \quad (17.71)$$

$$\Delta y = C \Delta x + D \Delta u \quad (17.72)$$

Here:

$$A_c = \left[\begin{array}{cc} \frac{\partial f_1}{\partial x_1} = 0 & \frac{\partial f_1}{\partial x_2} = -\frac{1}{A_{\text{tank}}} \\ \frac{\partial f_2}{\partial x_1} = 0 & \frac{\partial f_2}{\partial x_2} = 0 \end{array} \right] \Bigg|_{x_e(k), u(k)} \quad (17.73)$$

$$= \left[\begin{array}{cc} 0 & -\frac{1}{A_{\text{tank}}} \\ 0 & 0 \end{array} \right] \quad (17.74)$$

$$B_c = \left[\begin{array}{c} \frac{\partial f_1}{\partial u} = -\frac{K_p}{A_{\text{tank}}} \\ \frac{\partial f_2}{\partial u} = 0 \end{array} \right] \Bigg|_{x_e(k), u(k)} \quad (17.75)$$

$$= \left[\begin{array}{cc} 0 & -\frac{1}{A_{\text{tank}}} \\ 0 & 0 \end{array} \right] \quad (17.76)$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{17.77}$$

$$D = \begin{bmatrix} 0 \end{bmatrix} \tag{17.78}$$

The Butterworth polynomial is (17.26) which is repeated here:

$$B_2(s) = (Ts)^2 + 1.4142(Ts) + 1 \tag{17.79}$$

where T is given by (17.33) which is repeated here:

$$T \approx \frac{T_r}{n} \tag{17.80}$$

where $n = 2$ (the number of states). I specify the observer response time T_r to be

$$T_r = 2 \text{ s} \tag{17.81}$$

The observer gain K is calculated using function blocks in LabVIEW, see Figure 17.5. The result is

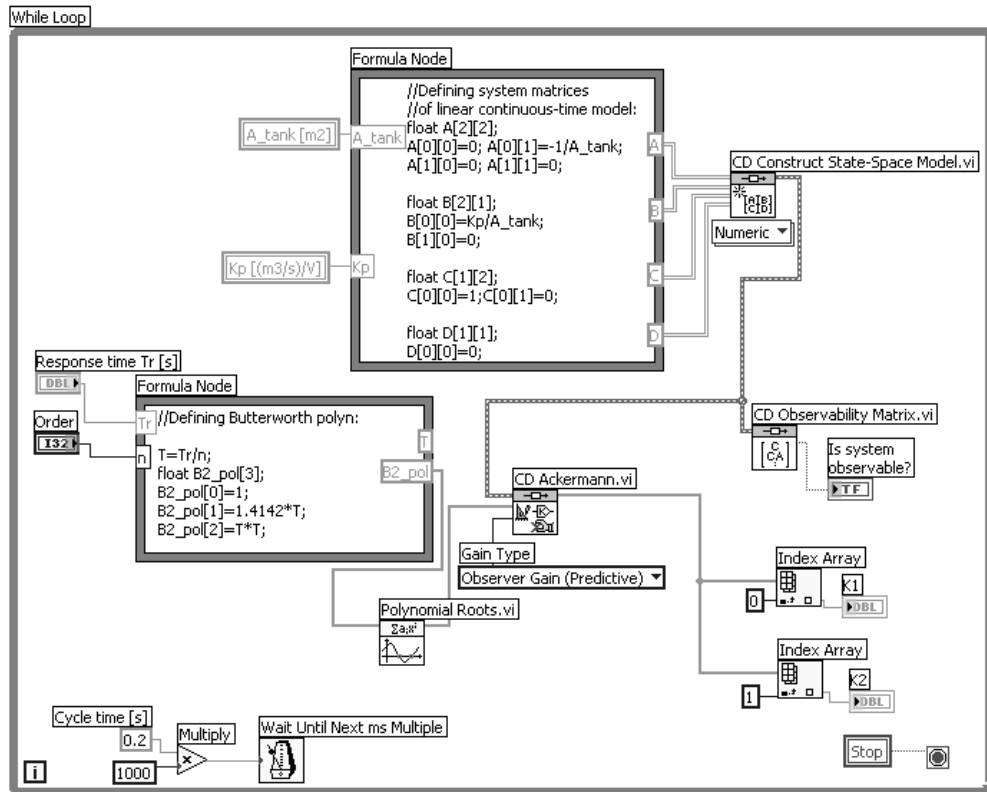


Figure 17.8: Example 17.3: While-loop for calculating the observer gain K

$$K = \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} = \begin{bmatrix} 1.414 \\ -0.1 \end{bmatrix} \tag{17.82}$$

Figure 17.9 shows the responses after a stepwise change of the outflow. (The level is controlled with a PI controller with settings $K_c = 10$ and $T_i = 10$ s.) The figure shows the “real” (simulated) and estimated level and outflow. We see from the lower chart in the figure that the Kalman Filter seems to estimate the outflow well, with response time approximately 2 sec, as specified, and with zero error in steady state.

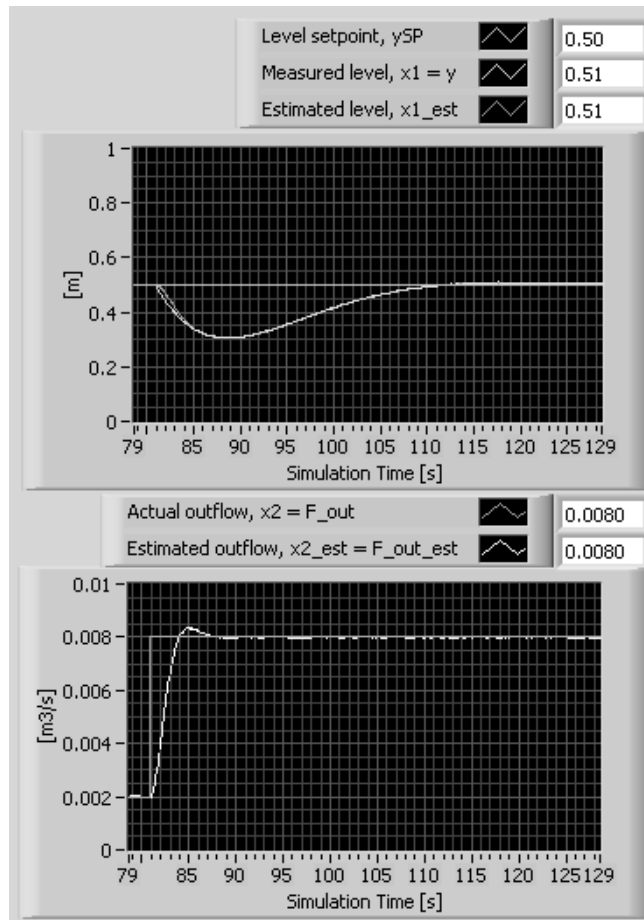


Figure 17.9: Example 17.3: The responses after a stepwise change of the outflow.

Figure 17.10 shows the implementation of the observer with C-code in a Formula Node. (The Formula Node is just one part of the block diagram. The total block diagram consists of one While loop where the observer gains are calculated, and one Simulation loop containing the Formula Node, PID controller, and the tank simulator.) Limitation of the estimated states to maximum and minimum values is included in the code. The

input a is used to force the observer to run just as a simulator which is very useful at sensor failure, cf. Section 17.8.

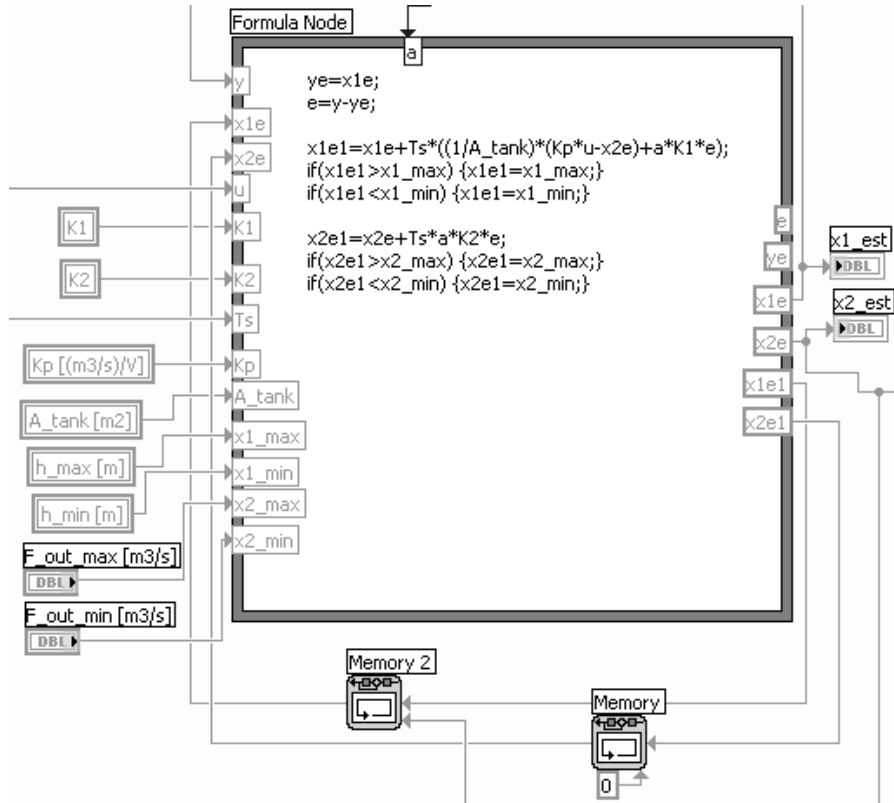


Figure 17.10: Example 17.3: Implementation of the observer in a Formula Node. (The observer gain K is fetched from the While loop in the Block diagram, see Figure 17.8, using local variables.)

[End of Example 17.3]

17.7 Using observer estimates in controllers

In the introduction to this chapter are listed several control functions which basically assumes that measurements of states and/or disturbances (loads) are available. If measurements from “hard-sensors” for some reason are not available, you can try using an estimate as provided by a soft-sensor as an observer (or Kalman Filter) in stead. One such control function is feedforward control. Figure 17.11 shows feedforward from

estimated disturbance.

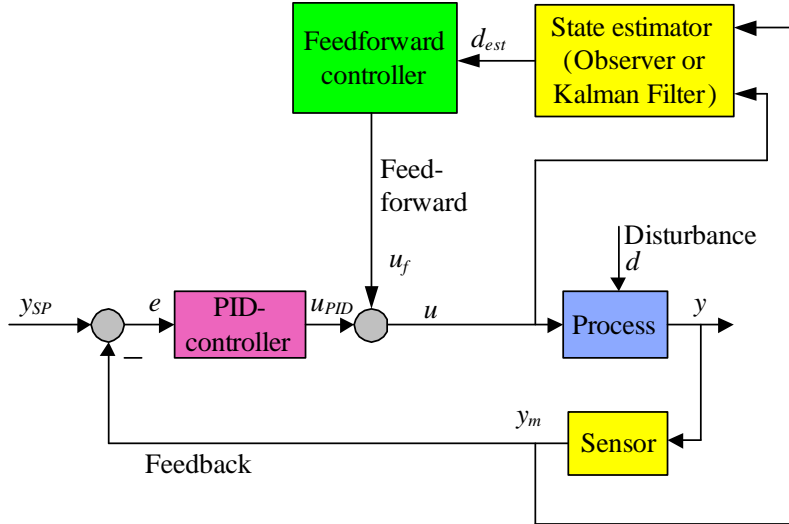


Figure 17.11: Control system including feedforward control from estimated disturbance (with observer or Kalman Filter)

Example 17.4 *Level control with feedforward from estimated disturbance (load)*

Figure 17.7 in Example 17.3 shows the front panel of a LabVIEW program of a simulated level control system. On the front panel is a switch which can be used to activate feedforward from estimated outflow, $F_{out_{est}}$. The estimator for $F_{out_{est}}$ based on observer was derived in that example. Let us now derive the feedforward controller, and then look at simulations of the control system.

The feedforward controller is derived from a mathematical model of the process. The model is given by (17.57), which is repeated here:

$$\dot{h}(t) = \frac{1}{A_{tank}} [K_p u - F_{out}(t)] \quad (17.83)$$

Solving for the control variable u , and substituting process output variable h by its setpoint h_{SP} gives the feedforward controller:

$$u_f(t) = \underbrace{\frac{A_{tank} \dot{h}_{SP}(t)}{K_p}}_{u_{f_{SP}}} + \underbrace{\frac{F_{out}(t)}{K_p}}_{u_{f_d}} \quad (17.84)$$

Let us assume that the level setpoint h_{SP} is constant. Then, $\dot{h}_{SP}(t) = 0$, and the feedforward controller becomes

$$u_f(t) = \frac{F_{out}(t)}{K_p} \quad (17.85)$$

Assuming that the estimate $F_{out_{est}}(t)$ is used in stead of F_{out} , the feedforward controller becomes

$$u_f(t) = \frac{F_{out_{est}}(t)}{K_p} \quad (17.86)$$

Let us look at a simulation where the outflow has been changed as a step from 0.002 to $0.008 \text{ m}^3/\text{s}$. Figure 17.12 shows the level response with feedforward. Compare with Figure 17.9 which shows the response *without*

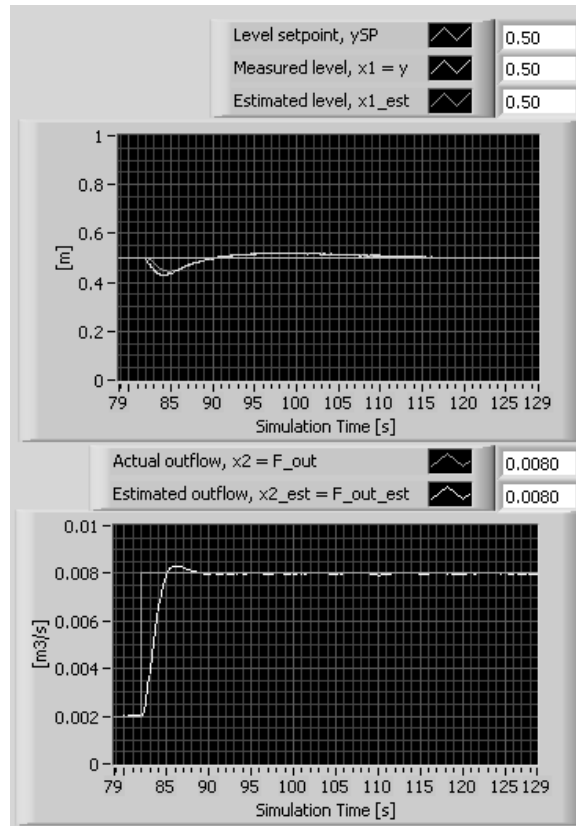


Figure 17.12: Example 17.4: Level response with feedforward from estimated outflow

feedforward. There is a substantial improvement by using feedforward from outflow— even if the outflow was not measured (only estimated)!