Pole-placement by state-space methods

Aalto University
School of Electrical
Engineering

Control Design

To be considered in controller design

* Compensate the effect of load disturbances
* Reduce the effect of measurement noise
* Setpoint following (target tracking)
* Inaccuracies in the process model and parameter variations

Items to consider in controller design

* Purpose of the control system
* Process model
* Disturbance model
* Model inaccuracies and changes
* Applicable control strategies
* Design parameters

Process to be controlled

Real continuous process

$$\begin{cases} \dot{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \end{cases}$$

Discrete equivalent

$$\begin{cases} \mathbf{x}(k+1) = \Phi\ \mathbf{x}(k) + \Gamma\ \mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}\ \mathbf{x}(k) \end{cases}$$

A linear state controller $\mathbf{u}(k) = -\mathbf{L}\ \mathbf{x}(k)$

State controller

State feedback is the most natural way to control a process, which is modelled as a state-space representation and whose states are measured or estimated. At this point it is assumed that the states can be measured directly.

$$\begin{cases} \mathbf{x}(k+1) = \Phi\ \mathbf{x}(k) + \Gamma\ \mathbf{u}(k) \\ \mathbf{u}(k) = -\mathbf{L}\ \mathbf{x}(k) \end{cases} \quad \Rightarrow \quad \mathbf{x}(k+1) = \Phi\ \mathbf{x}(k) - \Gamma \mathbf{L}\ \mathbf{x}(k)$$

$$\Rightarrow \quad \mathbf{x}(k+1) = (\Phi - \Gamma \mathbf{L})\ \mathbf{x}(k) = \Phi_{cl}\ \mathbf{x}(k)$$

Φ is the process system matrix, Φ_{cl} is the closed loop system matrix, If the process is reachable, the poles of Φ_{cl} can be set arbitrarily.

State controller

The eigenvalues of Φ_{cl} i.e. the roots of the characteristic equation of the controlled system determine the behaviour of the closed-loop system . The characteristic equation of the controlled system can be calculated as

$$\det(z\mathbf{I} - \Phi_{cl}) = \det(z\mathbf{I} - \Phi + \Gamma\mathbf{L}) = 0$$

The desired characteristic polynomial is determined by choosing the poles (z_{pi}).

$$\prod_{i=1}^{n}(z - z_{pi}) = (z - z_{p1})(z - z_{p2})\cdots(z - z_{pn}) = 0$$

By comparing the two characteristic equations, n conditions are obtained, from which the parameters of $\mathbf{L}$ can be solved.

State controller, example

The process

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

is controlled by a state feedback controller. The desired closed loop poles are $z_{p1} = z_{p2} = 0.5$. Determine the controller parameter vector **L**.

$$\det(z\mathbf{I} - \Phi_{cl}) = \det(z\mathbf{I} - \Phi + \Gamma\mathbf{L})$$

$$= \det\left(z\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} -1 & -2 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} l_1 & l_2 \end{bmatrix} \right) = \det\begin{bmatrix} z+1+l_1 & 2+l_2 \\ -1 & z \end{bmatrix}$$

$$= z^2 + (1+l_1)z + (2+l_2) = 0$$

Compare with the desired equation

State controller, example

$$\prod_{i=1}^{n}(z-z_{pi}) = (z-z_{p1})(z-z_{p2}) = (z-0.5)^2 = z^2 - z + 0.25 = 0$$

$$z^2 + (1+l_1)z + (2+l_2) \equiv z^2 - z + 0.25$$

$$\Rightarrow \quad \begin{cases} 1+l_1 = -1 \\ 2+l_2 = 0.25 \end{cases} \quad \Rightarrow \quad \begin{cases} l_1 = -2 \\ l_2 = -1.75 \end{cases}$$

The controller is then

$$u(k) = -\begin{bmatrix} -2 & -1.75 \end{bmatrix} \mathbf{x}(k)$$

Deadbeat-control

In *deadbeat* control all poles are set into the origin. The fastest possible discrete controller is obtained, which reaches the final value in n steps. By deadbeat strategy the closed loop characteristic equation becomes:

$$\prod_{i=1}^{n}(z - z_{pi}) = \prod_{i=1}^{n}(z - 0) = z^n = 0$$

State observers

It is unrealistic to assume that the states are measurable especially, when the noise states have been augmented to the state vector. Hence it is reasonable to estimate the unmeasured states. This is possible, if the system is observable.

Consider the SISO-case (SISO= single input-single output, scalars u and y)

$$\begin{cases} \mathbf{x}(k+1) = \Phi\ \mathbf{x}(k) + \Gamma\ u(k) \\ y(k) = \mathbf{C}\mathbf{x}(k) \end{cases}$$

State observers

The purpose of the estimator is to estimate ($\mathbf{x}(k)$) at time k based on present and past values of u and y , ($u(k)$, $u(k$-1), …, $y(k)$, $y(k$-1), …) The process model is however needed for this purpose. Define the following concepts

Real state	$\mathbf{x}(k)$
Estimated state	$\hat{\mathbf{x}}(k)$
Estimation error	$\tilde{\mathbf{x}}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k)$

State observers

When the process model is known, it would be logical to substitute the controls u to the model and simply calculate the state estimate at each time instant. However, this method does not consider the initial values of the state nor disturbances. The information obtained by the measured output signal y also remains unused.

The above strategy can be considerably improved . If the estimated and real output are the same, the estimator performs well. Otherwise the estimate must be corrected based on the estimation error.

State observers

Estimator based on the control signal only

Real process

$$\begin{cases} \mathbf{x}(k+1) = \Phi\,\mathbf{x}(k) + \Gamma\,\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \end{cases}$$

Estimator

$$\begin{cases} \hat{\mathbf{x}}(k+1) = \Phi\,\hat{\mathbf{x}}(k) + \Gamma\,\mathbf{u}(k) \\ \hat{\mathbf{y}}(k) = \mathbf{C}\hat{\mathbf{x}}(k) \end{cases}$$

Estimator error

$$\mathbf{y}(k) - \hat{\mathbf{y}}(k) = \mathbf{C}\mathbf{x}(k) - \mathbf{C}\hat{\mathbf{x}}(k)$$

$$= \mathbf{C}\big(\mathbf{x}(k) - \hat{\mathbf{x}}(k)\big) = \mathbf{C}\tilde{\mathbf{x}}(k)$$

State observers

Add the estimator error term

$$
\begin{aligned}
\hat{\mathbf{x}}(k+1) &= \mathbf{\Phi}\hat{\mathbf{x}}(k) + \Gamma\mathbf{u}(k) + \mathbf{K}(\mathbf{y}(k) - \hat{\mathbf{y}}(k)) \\
&= \mathbf{\Phi}\hat{\mathbf{x}}(k) + \Gamma\mathbf{u}(k) + \mathbf{K}(\mathbf{y}(k) - \mathbf{C}\hat{\mathbf{x}}(k)) \\
&= (\mathbf{\Phi} - \mathbf{KC})\hat{\mathbf{x}}(k) + \Gamma\mathbf{u}(k) + \mathbf{K}\mathbf{y}(k)
\end{aligned}
$$

The performance is studied by comparing the estimate with the real state

$$
\begin{aligned}
\tilde{\mathbf{x}}(k+1) &= \mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1) \\
&= (\mathbf{\Phi}\,\mathbf{x}(k) + \Gamma\,\mathbf{u}(k)) - (\mathbf{\Phi}\,\hat{\mathbf{x}}(k) + \Gamma\,\mathbf{u}(k) + \mathbf{K}(\mathbf{y}(k) - \hat{\mathbf{y}}(k))) \\
&= \mathbf{\Phi}(\mathbf{x}(k) - \hat{\mathbf{x}}(k)) - \mathbf{KC}(\mathbf{x}(k) - \hat{\mathbf{x}}(k)) \\
&= (\mathbf{\Phi} - \mathbf{KC})(\mathbf{x}(k) - \hat{\mathbf{x}}(k)) = (\mathbf{\Phi} - \mathbf{KC})\tilde{\mathbf{x}}(k)
\end{aligned}
$$

State observers

The model for the error dynamics is

$$\tilde{\mathbf{x}}(k+1) = (\Phi - \mathbf{KC})\tilde{\mathbf{x}}(k) = \Phi_o \tilde{\mathbf{x}}(k)$$

The matrix **K** is chosen such that the eigenvalues of Φ_o are at desired places in the complex plane. If they are inside the unit circle, the estimation error decreases, and if they are in the origin, the estimation error decreases at the maximum speed (deadbeat-strategy) vanishing totally by n steps. The procedure is dual to pole placement in controller design.

State observers, example

Estimate states of
$$\begin{cases} \mathbf{x}(k+1) = \begin{bmatrix} -1 & -2 \\ 1 & 0 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(k) \\ y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} \mathbf{x}(k) \end{cases}$$

by the dead-beat strategy. Determine the estimator parameters **K**.

$$\det(z\mathbf{I} - \Phi_o) = \det(z\mathbf{I} - \Phi + \mathbf{KC})$$

$$= \det\left(z\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} -1 & -2 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} \right) = \det\begin{bmatrix} z+1 & 2+k_1 \\ -1 & z+k_2 \end{bmatrix}$$

$$= (z+1)(z+k_2) + (2+k_1) = z^2 + (1+k_2)z + (2+k_1+k_2) = 0$$

State observers, example

Compare to the desired characteristic equation

$$\prod_{i=1}^{n}(z - z_{pi}) = (z - 0)^2 = z^2 = 0$$

$$z^2 + (1 + k_2)z + (2 + k_1 + k_2) \equiv z^2$$

$$\Rightarrow \begin{cases} 1 + k_2 = 0 \\ 2 + k_1 + k_2 = 0 \end{cases} \Rightarrow \begin{cases} k_1 = -1 \\ k_2 = -1 \end{cases}$$

State observers, example

The estimator is

$$\hat{\mathbf{x}}(k+1) = \begin{bmatrix} -1 & -2 \\ 1 & 0 \end{bmatrix} \hat{\mathbf{x}}(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \mathbf{u}(k) + \begin{bmatrix} -1 \\ -1 \end{bmatrix} (\mathbf{y}(k) - \begin{bmatrix} 0 & 1 \end{bmatrix} \hat{\mathbf{x}}(k))$$

$$\Rightarrow \quad \hat{\mathbf{x}}(k+1) = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \hat{\mathbf{x}}(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \mathbf{u}(k) + \begin{bmatrix} -1 \\ -1 \end{bmatrix} \mathbf{y}(k)$$

Let us check the performance by calculating two steps from some initial condition

$$\hat{\mathbf{x}}(0) = \begin{bmatrix} \hat{x}_{10} \\ \hat{x}_{20} \end{bmatrix}, \quad \mathbf{x}(0) = \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix}, \quad u(0) = u_0, \quad u(1) = u_1, \quad y(0) = x_{20}$$

State observers, example

Check first how the process behaves

$$\begin{cases} \mathbf{x}(1) = \begin{bmatrix} -1 & -2 \\ 1 & 0 \end{bmatrix} \mathbf{x}(0) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(0) = \begin{bmatrix} -1 & -2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_0 \\ y(1) = \begin{bmatrix} 0 & 1 \end{bmatrix} \mathbf{x}(1) = \begin{bmatrix} 0 & 1 \end{bmatrix} \mathbf{x}(1) \end{cases}$$

$$\begin{cases} \mathbf{x}(1) = \begin{bmatrix} -x_{10} - 2x_{20} + u_0 \\ x_{10} \end{bmatrix} \\ y(1) = x_{10} \end{cases}$$

State observers, example

$$\begin{cases} \mathbf{x}(2) = \begin{bmatrix} -1 & -2 \\ 1 & 0 \end{bmatrix} \mathbf{x}(1) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(1) = \begin{bmatrix} -1 & -2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -x_{10} - 2x_{20} + u_0 \\ x_{10} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_1 \\ y(2) = \begin{bmatrix} 0 & 1 \end{bmatrix} \mathbf{x}(2) = \begin{bmatrix} 0 & 1 \end{bmatrix} \mathbf{x}(2) \end{cases}$$

$$\begin{cases} \mathbf{x}(2) = \begin{bmatrix} x_{10} + 2x_{20} - u_0 - 2x_{10} + u_1 \\ -x_{10} - 2x_{20} + u_0 \end{bmatrix} = \begin{bmatrix} -x_{10} + 2x_{20} - u_0 + u_1 \\ -x_{10} - 2x_{20} + u_0 \end{bmatrix} \\ y(2) = -x_{10} - 2x_{20} + u_0 \end{cases}$$

State observers, example

The estimator can only use input and output signals u and y.

$$u(0) = u_0 \quad y(0) = x_{20}$$
$$u(1) = u_1 \quad y(1) = x_{10}$$

Based on this information the estimator should find $\mathbf{x}(2)$ without any knowledge of the real initial state (or real state at any time instant).

$$\mathbf{x}(2) = \begin{bmatrix} -x_{10} + 2x_{20} - u_0 + u_1 \\ -x_{10} - 2x_{20} + u_0 \end{bmatrix}$$

State observers, example

Let us calculate the estimate recursively from an arbitrary initial guess

$$\hat{\mathbf{x}}(0) = \begin{bmatrix} \hat{x}_{10} \\ \hat{x}_{20} \end{bmatrix}$$

$$\hat{\mathbf{x}}(1) = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \hat{\mathbf{x}}(0) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(0) + \begin{bmatrix} -1 \\ -1 \end{bmatrix} y(0)$$

$$= \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_{10} \\ \hat{x}_{20} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_0 + \begin{bmatrix} -1 \\ -1 \end{bmatrix} x_{20} = \begin{bmatrix} -\hat{x}_{10} - \hat{x}_{20} + u_0 - x_{20} \\ \hat{x}_{10} + \hat{x}_{20} - x_{20} \end{bmatrix}$$

State observers, example

$$\hat{\mathbf{x}}(2) = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \hat{\mathbf{x}}(1) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(1) + \begin{bmatrix} -1 \\ -1 \end{bmatrix} y(1)$$

$$= \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} -\hat{x}_{10} - \hat{x}_{20} + u_0 - x_{20} \\ \hat{x}_{10} + \hat{x}_{20} - x_{20} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} -1 \\ -1 \end{bmatrix} x_{10}$$

$$= \begin{bmatrix} -x_{10} + 2x_{20} - u_0 + u_1 \\ -x_{10} - 2x_{20} + u_0 \end{bmatrix} = \mathbf{x}(2)$$

The estimator converges and the real state is found in two steps according to the dead-beat strategy, even if the process itself is unstable (like in this example).

An alternative state estimator

Above the output measurement at time k is used to estimate the state $\mathbf{x}(k+1)$, (at time $k+1$). So there is a delay.

$$\hat{\mathbf{x}}(k+1) = \mathbf{\Phi}\hat{\mathbf{x}}(k) + \mathbf{\Gamma}\mathbf{u}(k) + \mathbf{K}(\mathbf{y}(k) - \hat{\mathbf{y}}(k))$$

Another possibility is to use the measurement $y(k)$ at time k to estimate $\mathbf{x}(k)$. It then follows

$$\hat{\mathbf{x}}(k+1) = \mathbf{\Phi}\hat{\mathbf{x}}(k) + \mathbf{\Gamma}\mathbf{u}(k) + \mathbf{K}(\mathbf{y}(k+1) - \hat{\mathbf{y}}(k+1))$$

$$= \mathbf{\Phi}\hat{\mathbf{x}}(k) + \mathbf{\Gamma}\mathbf{u}(k) + \mathbf{K}\left(\mathbf{y}(k+1) - \mathbf{C}\hat{\mathbf{x}}(k+1)\right)$$

$$= \mathbf{\Phi}\hat{\mathbf{x}}(k) + \mathbf{\Gamma}\mathbf{u}(k) + \mathbf{K}\left(\mathbf{y}(k+1) - \mathbf{C}\left(\mathbf{\Phi}\hat{\mathbf{x}}(k) + \mathbf{\Gamma}\mathbf{u}(k)\right)\right)$$

$$= \left(\mathbf{I} - \mathbf{KC}\right)\left(\mathbf{\Phi}\hat{\mathbf{x}}(k) + \mathbf{\Gamma}\mathbf{u}(k)\right) + \mathbf{K}\mathbf{y}(k+1)$$

An alternative state estimator, cont

The estimator error dynamics is now

$$\tilde{\mathbf{x}}(k+1) = \left(\Phi - \mathbf{KC}\Phi\right)\tilde{\mathbf{x}}(k) = \Phi_o \tilde{\mathbf{x}}(k)$$

Additionally, it holds that

$$\tilde{\mathbf{y}}(k+1) = \mathbf{C}\tilde{\mathbf{x}}(k+1) = \mathbf{C}\left(\left(\Phi - \mathbf{KC}\Phi\right)\tilde{\mathbf{x}}(k)\right) = \left(\mathbf{C}\Phi - \mathbf{CKC}\Phi\right)\tilde{\mathbf{x}}(k)$$

$$= \left(\left(\mathbf{I} - \mathbf{CK}\right)\mathbf{C}\Phi\right)\tilde{\mathbf{x}}(k)$$

By choosing **I-CK** = **0** or **CK** = **I**, the estimation error can be eliminated

$$\tilde{\mathbf{y}}(k+1) = \mathbf{0} \quad \Rightarrow \quad \mathbf{y}(k+1) - \mathbf{C}\hat{\mathbf{x}}(k+1) = 0 \quad \Rightarrow \quad \mathbf{y}(k) = \mathbf{C}\hat{\mathbf{x}}(k)$$

An alternative state estimator, cont

The order of the estimator can then be reduced, which leads to the so-called *Luenberger observer* . Use the previous example and design a Luenberger observer with deadbeat-strategy.

$$\mathbf{CK} = \mathbf{I} \quad \Rightarrow \quad \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = k_2 = 1$$

So that the second element of the matrix $\mathbf{K}$ (k_2) is fixed. Tune k_1 with deadbeat-strategy.

$$\det(z\mathbf{I} - \mathbf{\Phi}_o) = \det(z\mathbf{I} - \mathbf{\Phi} + \mathbf{KC\Phi})$$

$$= \det\left(z\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} -1 & -2 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} k_1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & -2 \\ 1 & 0 \end{bmatrix} \right)$$

An alternative state estimator, cont

$$= \det\begin{bmatrix} z+1+k_1 & 2 \\ 0 & z \end{bmatrix} = z^2 + (k_1+1)z = 0$$

By comparing to deadbeat-criterion ($z^2 = 0$) k_1 becomes -1.

The observer becomes $\mathbf{K} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$

$$\hat{\mathbf{x}}(k+1) = (\mathbf{I} - \mathbf{KC})\big(\Phi\hat{\mathbf{x}}(k) + \Gamma u(k)\big) + \mathbf{K}y(k+1)$$

$$= \begin{bmatrix} 0 & -2 \\ 0 & 0 \end{bmatrix}\hat{\mathbf{x}}(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix}u(k) + \begin{bmatrix} -1 \\ 1 \end{bmatrix}y(k+1)$$

An alternative state estimator, cont

The second state is not observed at all. It is only recognized to be the same as the output signal.

$$\hat{\mathbf{x}}(k+1) = \begin{bmatrix} 0 & -2 \\ 0 & 0 \end{bmatrix} \hat{\mathbf{x}}(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} -1 \\ 1 \end{bmatrix} y(k+1)$$

$$\Rightarrow \quad \begin{cases} \hat{x}_1(k+1) = -2\hat{x}_2(k) + u(k) - y(k+1) \\ \hat{x}_2(k+1) = y(k+1) \end{cases}$$

$$\Rightarrow \quad \begin{cases} \hat{x}_1(k+1) = -2y(k) + u(k) - y(k+1) \\ \hat{x}_2(k+1) = y(k+1) \end{cases}$$

An alternative state estimator, cont

Try the Luenberger estimator as it was done previously with the full-order observer.

$$u(0) = u_0 \quad y(0) = x_{20} \qquad \mathbf{x}(1) = \begin{bmatrix} -x_{10} - 2x_{20} + u_0 \\ x_{10} \end{bmatrix}$$
$$u(1) = u_1 \quad y(1) = x_{10}$$

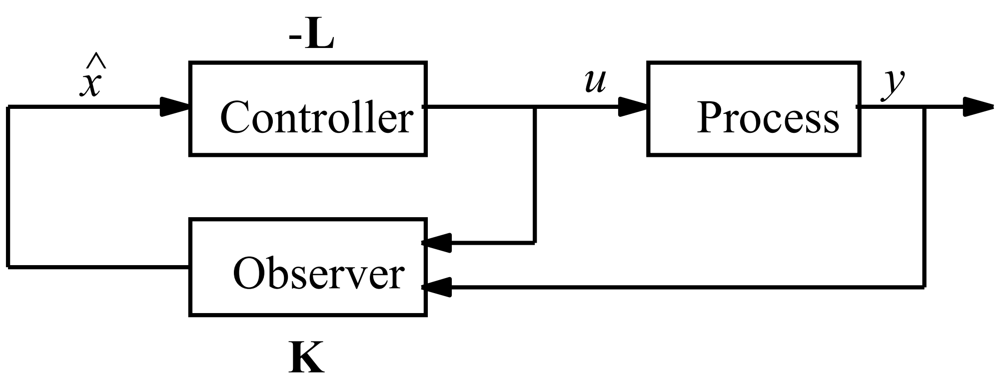
$$\begin{cases} \hat{x}_1(k+1) = -2y(k) + u(k) - y(k+1) \\ \hat{x}_2(k+1) = y(k+1) \end{cases}$$

$$\Rightarrow \quad \begin{cases} \hat{x}_1(1) = -2y(0) + u(0) - y(1) = -x_{10} - 2x_{20} + u_0 \\ \hat{x}_2(1) = x_{10} \end{cases}$$

The observer finds the correct state with one step.

Output feedback

By combining the state observer and state feedback controller it is possible to design a controller based on output measurements.



Output feedback, cont

The process
$$\begin{cases} \mathbf{x}(k+1) = \Phi\ \mathbf{x}(k) + \Gamma\ \mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}\ \mathbf{x}(k) \end{cases}$$

is controlled with state feedback, $\mathbf{u}(k) = -\mathbf{L}\ \mathbf{x}(k)$
which uses an estimated state. $\mathbf{u}(k) = -\mathbf{L}\ \hat{\mathbf{x}}(k)$

For the controlled system
$$\hat{\mathbf{x}}(k+1) = (\Phi - \mathbf{KC})\hat{\mathbf{x}}(k) + \Gamma\mathbf{u}(k) + \mathbf{K}\mathbf{y}(k)$$
$$\begin{cases} \mathbf{x}(k+1) = \Phi\ \mathbf{x}(k) - \Gamma\mathbf{L}\ \hat{\mathbf{x}}(k) \\ \mathbf{y}(k) = \mathbf{C}\ \mathbf{x}(k) \end{cases}$$

Output feedback, cont

For the state observer it holds that

$$\tilde{\mathbf{x}}(k+1) = (\Phi - \mathbf{KC})\tilde{\mathbf{x}}(k) = \Phi_o \tilde{\mathbf{x}}(k)$$

$$\tilde{\mathbf{x}}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k) \quad \Rightarrow \quad \hat{\mathbf{x}}(k) = \mathbf{x}(k) - \tilde{\mathbf{x}}(k)$$

By considering these equations it follows that

$$\begin{cases} \mathbf{x}(k+1) = \Phi\,\mathbf{x}(k) - \Gamma\mathbf{L}\left(\mathbf{x}(k) - \tilde{\mathbf{x}}(k)\right) = (\Phi - \Gamma\mathbf{L})\mathbf{x}(k) + \Gamma\mathbf{L}\,\tilde{\mathbf{x}}(k) \\ \tilde{\mathbf{x}}(k+1) = (\Phi - \mathbf{KC})\tilde{\mathbf{x}}(k) \\ \mathbf{y}(k) = \mathbf{C}\,\mathbf{x}(k) \end{cases}$$

Output feedback, cont

The state equations of the controlled system are

$$\begin{cases} \begin{bmatrix} \mathbf{x}(k+1) \\ \hline \tilde{\mathbf{x}}(k+1) \end{bmatrix} = \left[\begin{array}{c|c} \Phi - \Gamma\mathbf{L} & \Gamma\mathbf{L} \\ \hline \mathbf{0} & \Phi - \mathbf{KC} \end{array}\right] \begin{bmatrix} \mathbf{x}(k) \\ \hline \tilde{\mathbf{x}}(k) \end{bmatrix} \\ \mathbf{y}(k) = \left[\begin{array}{c|c} \mathbf{C} & \mathbf{0} \end{array}\right] \begin{bmatrix} \mathbf{x}(k) \\ \hline \tilde{\mathbf{x}}(k) \end{bmatrix} \end{cases}$$

The dimension of the controlled system is $2n$ and the characteristic equation is

$$\det\left(zI - \Phi + \Gamma\mathbf{L}\right) \cdot \det\left(zI - \Phi + \mathbf{KC}\right) = 0$$

Output feedback, cont

The eigenvalues of the closed-loop system is a combination of the eigenvalues of the controller and estimator. The important consequence of this is that the state feedback controller and state estimator can be designed separately without bothering about their influence on each other.

Servo problem

Until now only regulator problems have been discussed. If the system has to follow a changing reference signal, the servo problem follows.

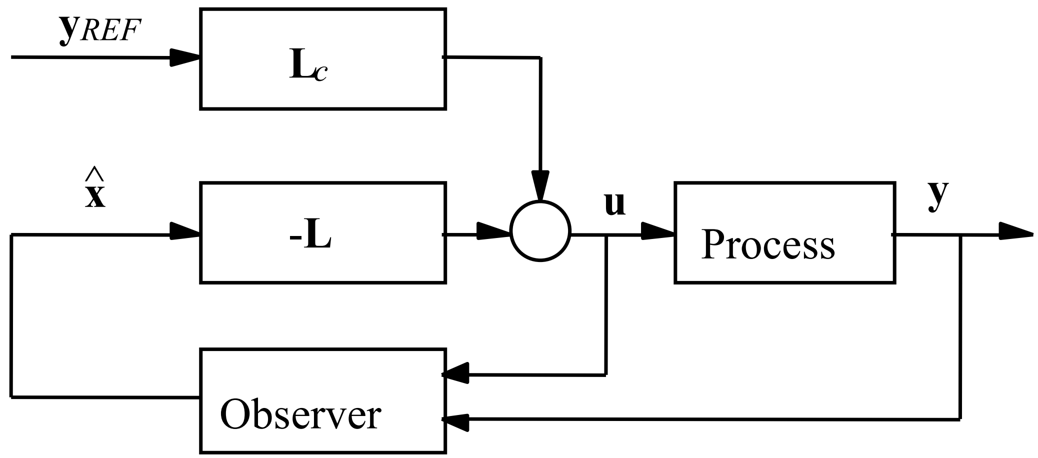
In the most simple setting the reference signal is added to the system multiplied by a constant $\mathbf{L}_c$, which is tuned to ensure the static gain 1 (in order not to have a permanent error). However this is not a good solution generally: the reference signal may change, also there might be load disturbances entering the system meaning that the permanent error can not be avoided in practice.

Servoproblem, cont

Use the control law

$$\mathbf{u}(k) = \mathbf{L}_c \mathbf{y}_{REF}(k) - \mathbf{L}\hat{\mathbf{x}}(k)$$

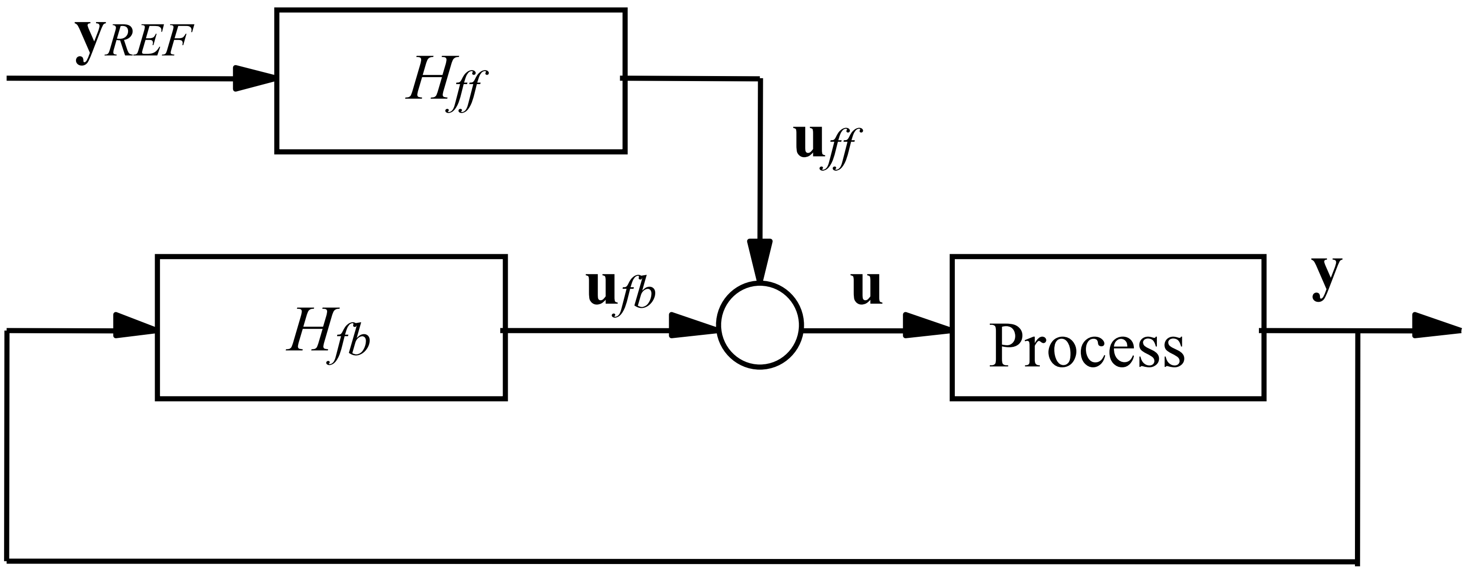
In which $\mathbf{y}_{REF}$ is the setpoint



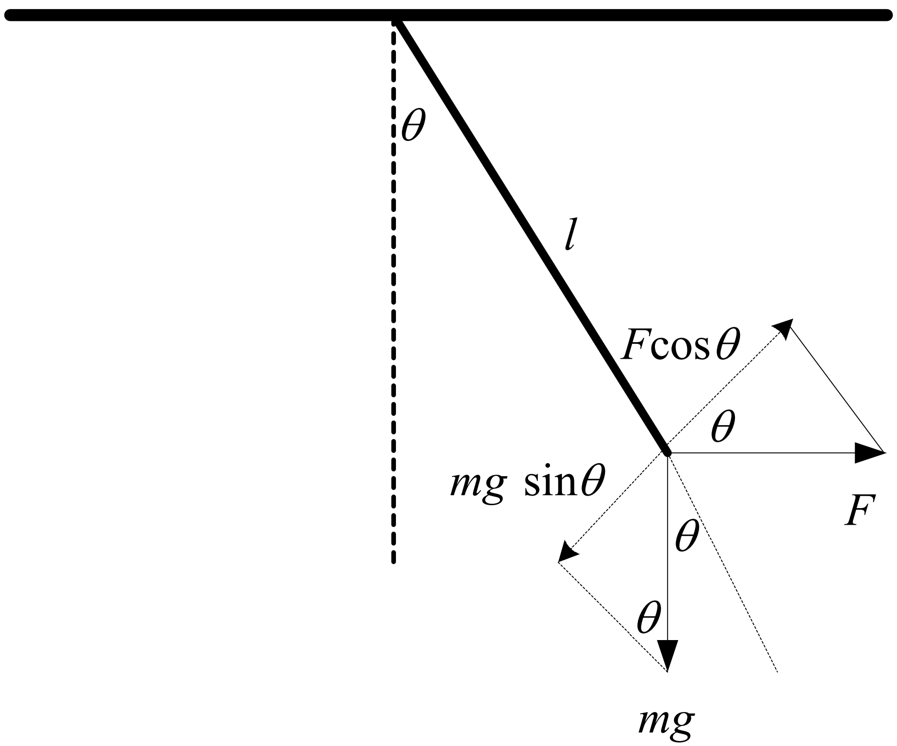
$$\begin{cases} \begin{bmatrix} \mathbf{x}(k+1) \\ \hline \tilde{\mathbf{x}}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi - \Gamma\mathbf{L} & \Gamma\mathbf{L} \\ \hline \mathbf{0} & \Phi - \mathbf{K}\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \hline \tilde{\mathbf{x}}(k) \end{bmatrix} + \begin{bmatrix} \Gamma\mathbf{L}_c \\ \hline \mathbf{0} \end{bmatrix} \mathbf{y}_{REF}(k) \\ \mathbf{y}(k) = \begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \hline \tilde{\mathbf{x}}(k) \end{bmatrix} \end{cases}$$

Two-degrees-of-freedom controller

Often servo and regulator parts are combined (feedforward H_{ff}) and (feedback H_{fb}).



Harmonic oscillator



The pendulum is controlled by a horizontal force.

Basic equations of circular motion
I (moment of inertia),
M (momentums)

$$I\ddot{\theta} = \sum M \;\Rightarrow\; ml^2\ddot{\theta} = -mg\sin\theta \cdot l + F\cos\theta \cdot l$$

$$\Rightarrow \ddot{\theta} = -\frac{g}{l}\sin\theta + \frac{F}{ml}\cos\theta$$

Use the state variables $x_1 = \theta, x_2 = \dot{\theta}, u = F$

$$y = \theta = x_1$$

which gives the nonlinear state-space representation

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\dfrac{g}{l}\sin(x_1) + \dfrac{u}{ml}\cos(x_1) = -\sin(x_1) + u\cos(x_1) \\ y = x_1 \end{cases}$$

where it has been assumed

$g = l$, $m = 1/l$, for simplicity.

Linearization around the equilibrium point

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\sin(x_1) + u\cos(x_1) \approx -x_1 + u \\ y = x_1 \end{cases}$$

$$x_1 = \theta \approx 0 \ \Rightarrow$$
$$\sin(x_1) \approx x_1,\ \cos(x_1) \approx 1$$

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$

$$G(s) = \frac{1}{s^2 + 1}$$

Generalization

$$\dot{x} = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ \omega \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$

ZOH:

$$x(kh+h) = \begin{bmatrix} \cos\ \omega h & \sin\ \omega h \\ \text{-}\sin\ \omega h & \cos\ \omega h \end{bmatrix} x(kh) + \begin{bmatrix} 1-\cos\ \omega h \\ \sin\ \omega h \end{bmatrix} u(kh)$$

$$y(kh) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(kh)$$

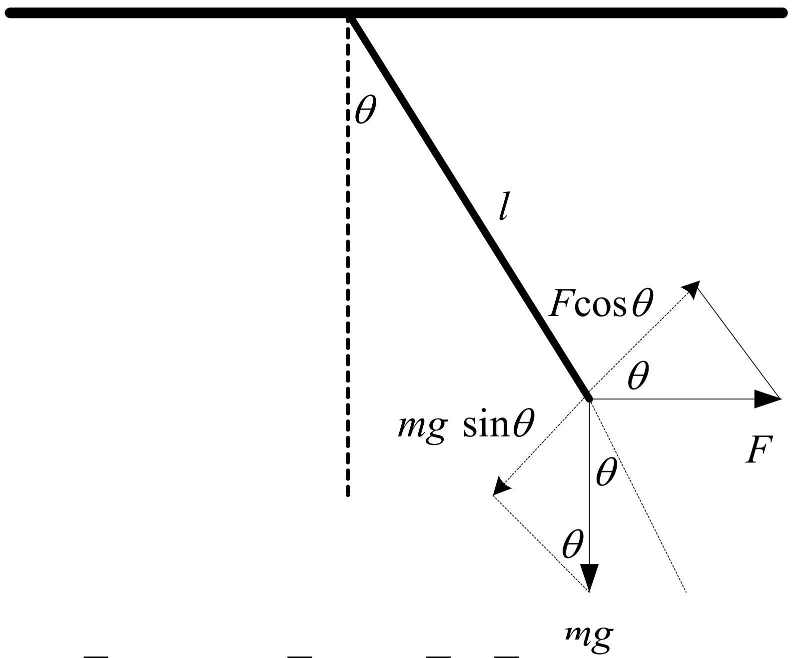
Problem to think...

Design a discrete-time controller to the pendulum.

What things must be considered?

How do you proceed?

Problem to think...



Process and its characteristics
- regulator problem?
- servo problem?

Linearity?
Linearization?

Controller design
Simulation

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$

$$G(s) = \frac{1}{s^2 + 1}$$

What about nonlinear design methods?

Problem to think...

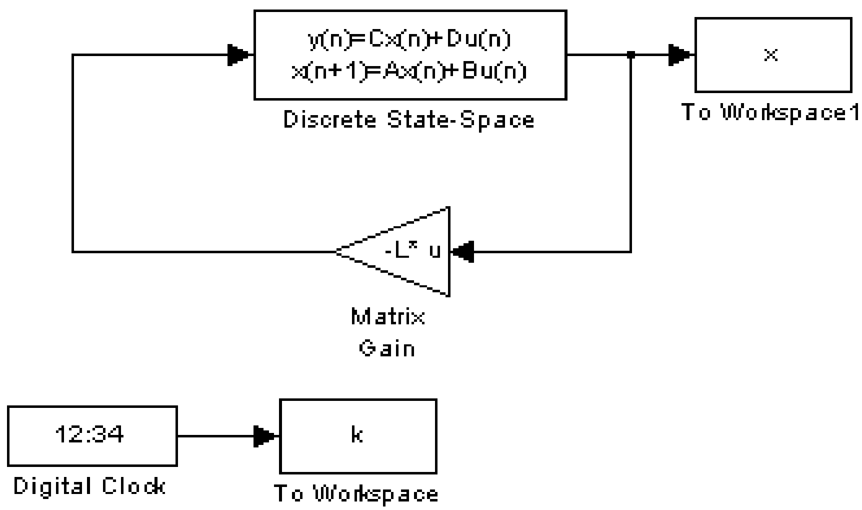
Either: design the compensator in continuous time and discretize it

or: Diskretize the process and design the compensator directly in discrete time.

Note:

- diskretization
- choosing the sampling interval, e.g. $\omega_0 h = 0.2 - 0.6$
- state feedback control
- state observer,
- servo problem, disturbances, static gain, integration in the controller

Problem to think...



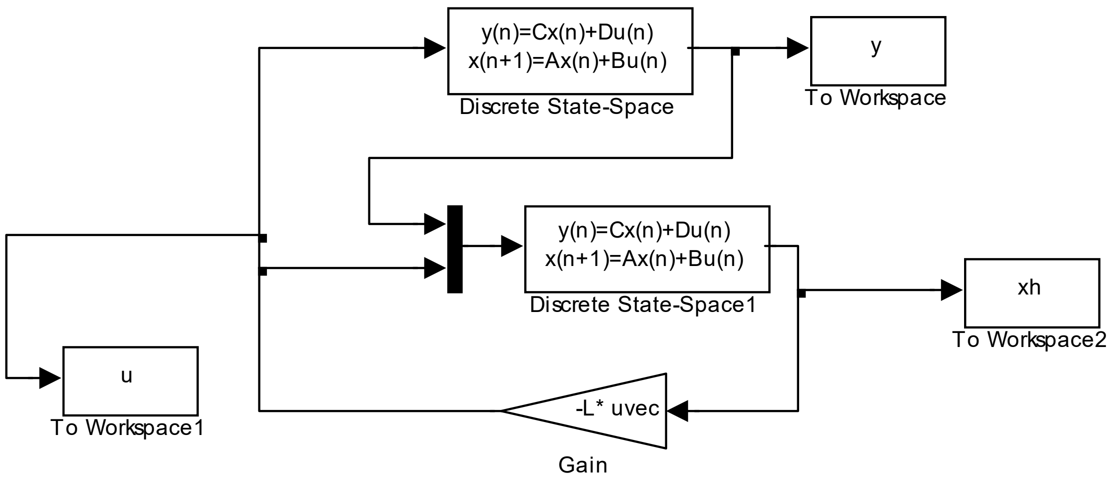
Dimensions?

Gtf=tf(1,[1 0 1]);
Gss=ss(Gtf);
h=0.3;
Htf=c2d(Gtf,h,'zoh');
Hss=ss(Htf);
[Phi,Gamma,C,D]=
ssdata(Hss);

$$p=[0.2;0.3];$$

$$L=\text{place}\,(\text{Phi},\text{Gamma},p);$$

State controller and -observer



State controller and -observer

But what are the matrices in the state observer block?

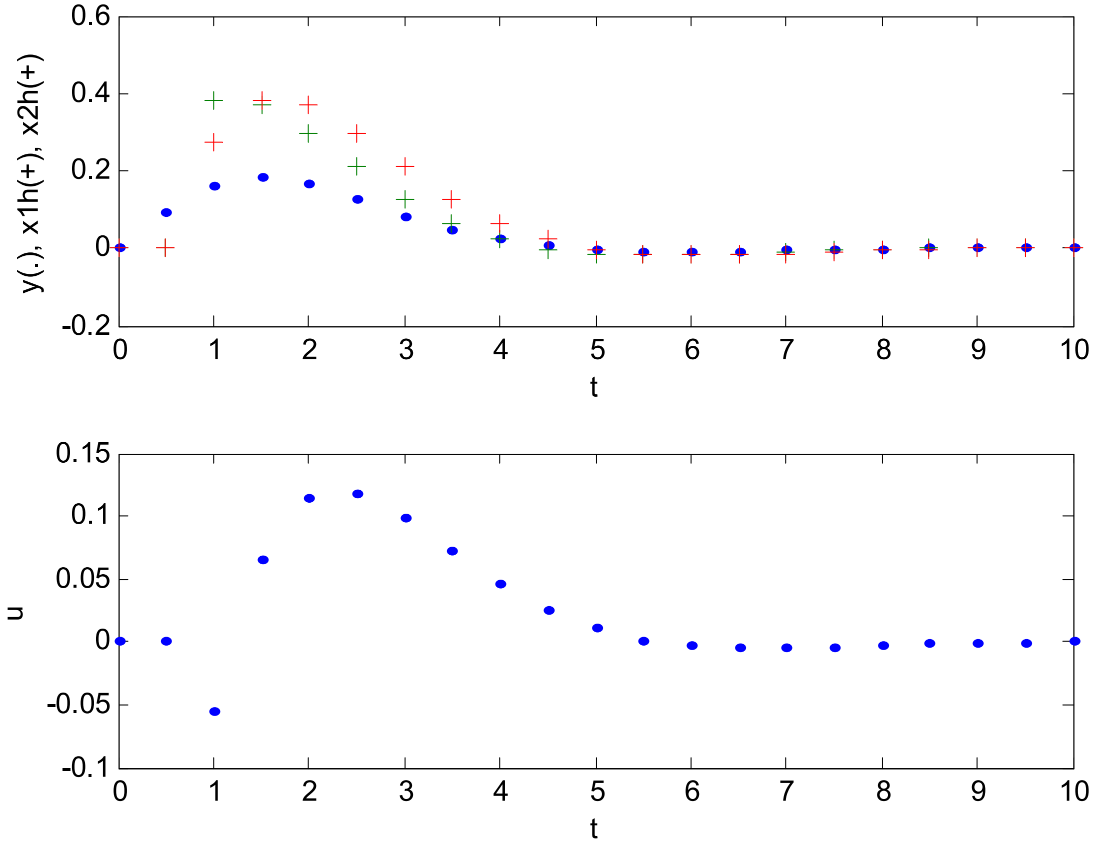
$$\hat{x}(k+1) = \Phi\hat{x}(k) + \Gamma u(k) + K\left[y(k) - C\hat{x}(k)\right]$$
$$= (\Phi - KC)\hat{x}(k) + Ky(k) + \Gamma u(k)$$
$$= (\Phi - KC)\hat{x}(k) + \begin{bmatrix} K & \Gamma \end{bmatrix}\begin{bmatrix} y(k) \\ u(k) \end{bmatrix}$$

Note. partitioned matrices

But this is an ordinary state-space representation! The corresponding Simulink block can directly be used.

$$A : \Phi - KC,\ B : \begin{bmatrix} K & \Gamma \end{bmatrix},\ C : \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ D : \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

State controller and -observer



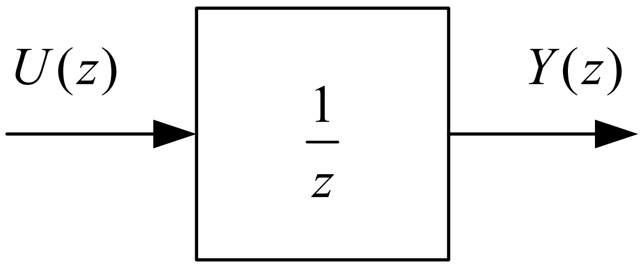
Output
Estimated states

(Initial states of the true system are non-zero initially)

Control

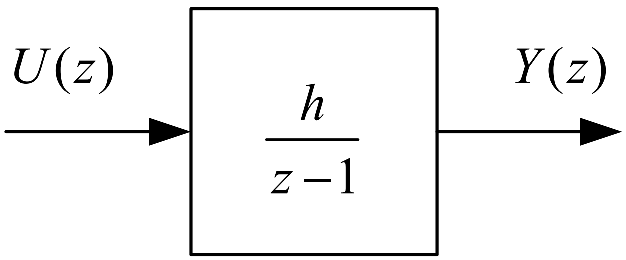
State controller and -observer

A couple things to note:



$$Y(z) = z^{-1}U(z) \Rightarrow y(k+1) = u(k)$$
$$y(k) = u(k-1)$$

Delay

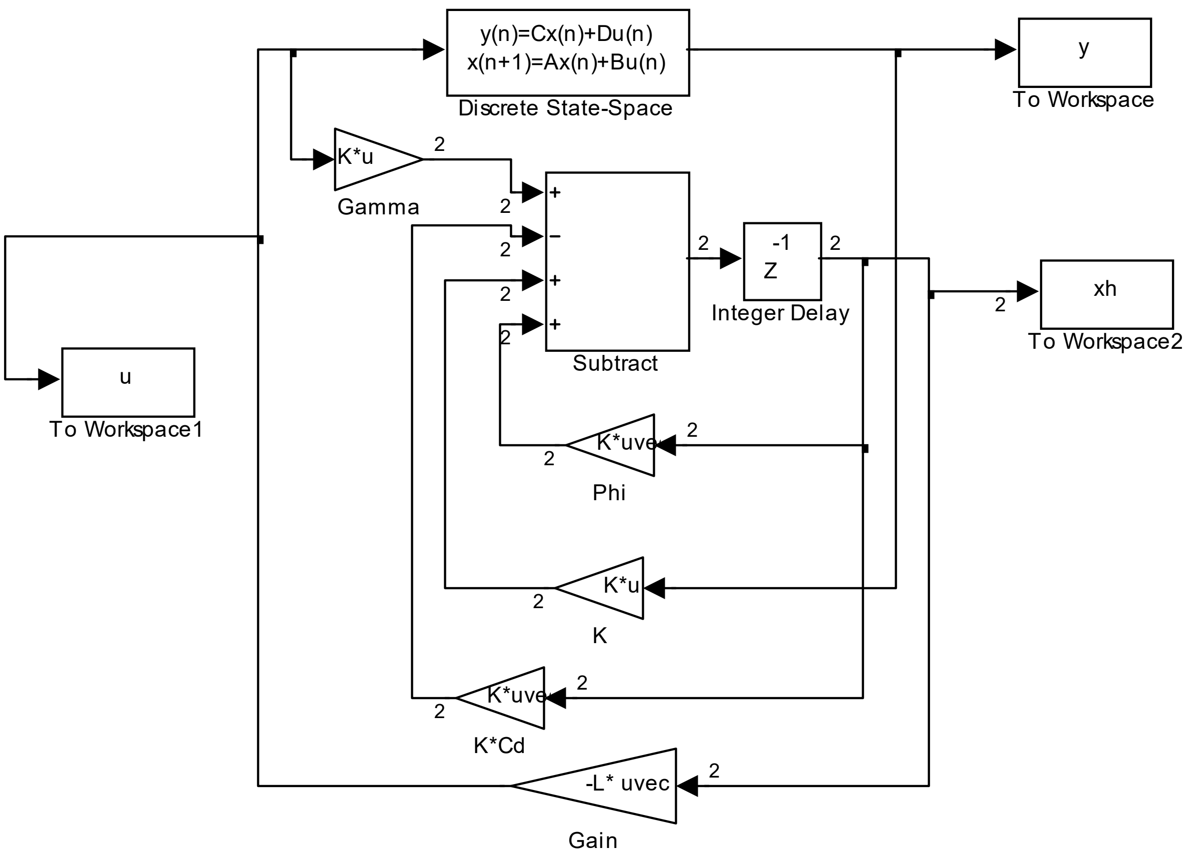


$$Y(z) = \frac{h}{z-1}U(z)$$

$$y(k+1) - y(k) = hu(k) \Rightarrow y(k) = y(k-1) + hu(k-1)$$
$$\Delta y(k) = y(k) - y(k-1) = hu(k-1)$$

Integration

State controller and -observer



You can do it this way also.

Usually you have to fight with the dimensions.

It is desired to get one compensator (pulse transfer function). Let's see

$$\hat{x}(k+1) = \Phi\hat{x}(k) + \Gamma u(k) + K\left[y(k) - C\hat{x}(k)\right]$$
$$u(k) = -L\hat{x}(k)$$

The input to the ”combined controller” is the measured signal *y*, and the output is the control *u*.

$$z\hat{X}(z) = \Phi\hat{X}(z) + \Gamma U(z) + KY(z) - KC\hat{X}(z)$$
$$U(z) = -L\hat{X}(z)$$

Taking the Z-transformation

Connecting the controller and observer

Eliminate $\hat{X}(z)$

$$\left(zI - \Phi + \Gamma L + KC\right)\hat{X}(z) = KY(z)$$

$$\hat{X}(z) = \left(zI - \Phi + \Gamma L + KC\right)^{-1} KY(z)$$

Finally

$$U(z) = -L\hat{X}(z) = -L\left(zI - \Phi + \Gamma L + KC\right)^{-1} KY(z) = H_c(z)Y(z)$$

Connecting the controller and observer

How do you realize the controller (e.g. in Matlab)?

$$H_c(z) = -L\left(zI - \Phi + \Gamma L + KC\right)^{-1} K$$

But this has the familiar form $\bar{C}\left(zI - \bar{\Phi}\right)^{-1}\bar{\Gamma}$

which corresponds to the matrices in state-space represen-tation $\left(\bar{\Phi}, \bar{\Gamma}, \bar{C}, 0\right)$

Connecting the controller and observer

Instead of calculating by hand realize the controller

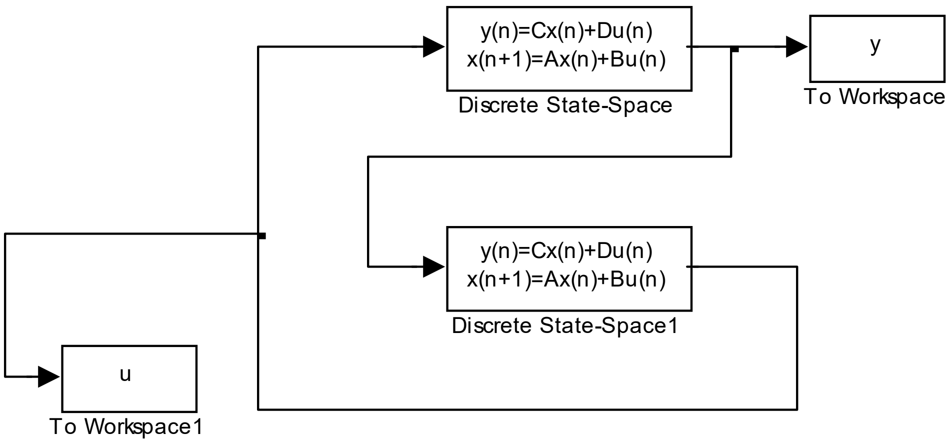
$$H_c(z) = -L\left(zI - \Phi + \Gamma L + KC\right)^{-1} K$$

by the Matlab commands

$$Hcss = \mathrm{ss}(\Phi - \Gamma L - KC, K, -L, 0);$$

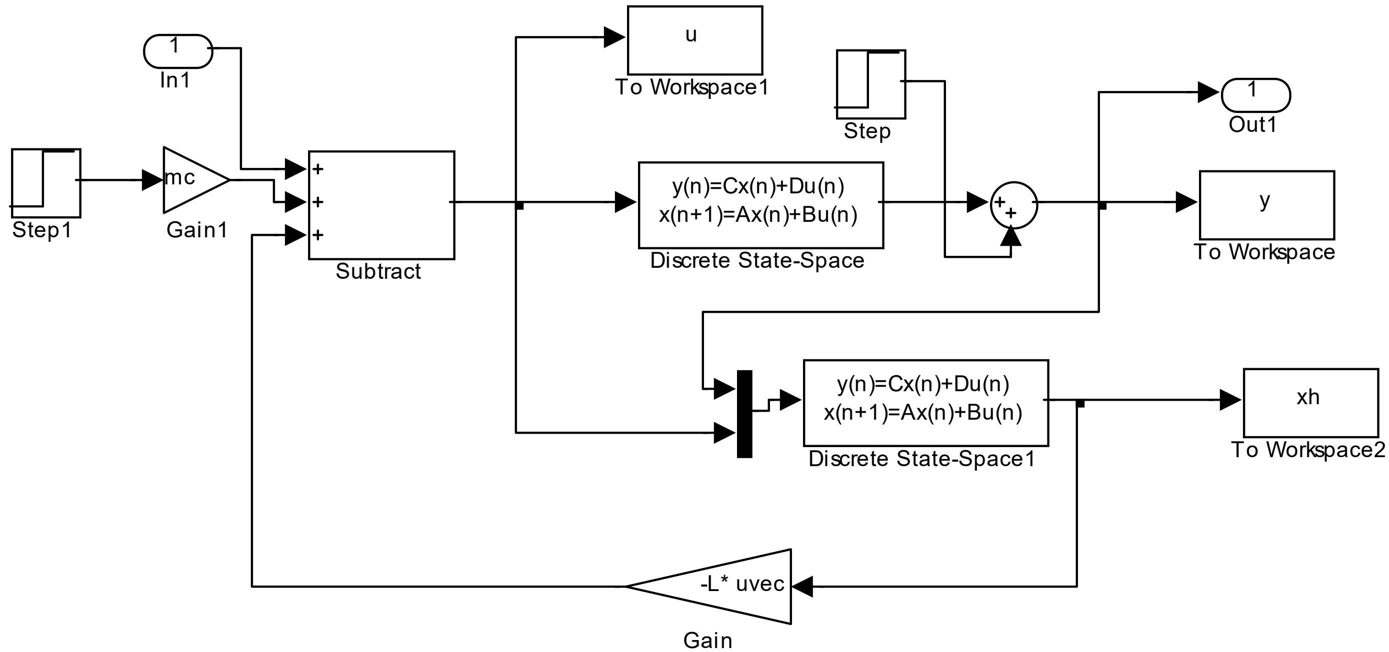
$$Hc = \mathrm{tf}(Hcss)$$

Connecting the controller and observer



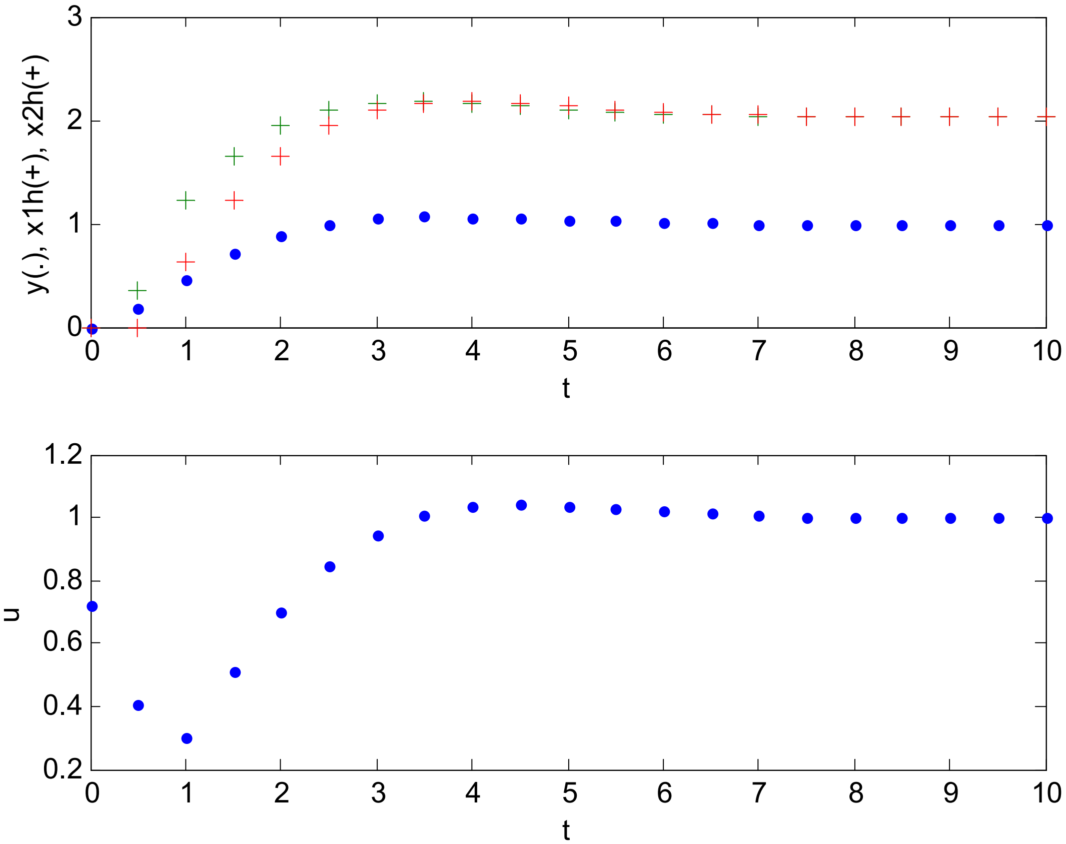
The simulation result is the same as earlier.

Reference signal and load disturbance



[Phi1,Gamma1,C1,D1]=dlinmod('heiluri_a2');
mc=1/ddcgain(Phi1,Gamma1,C1,D1);sim('heiluri_a2');

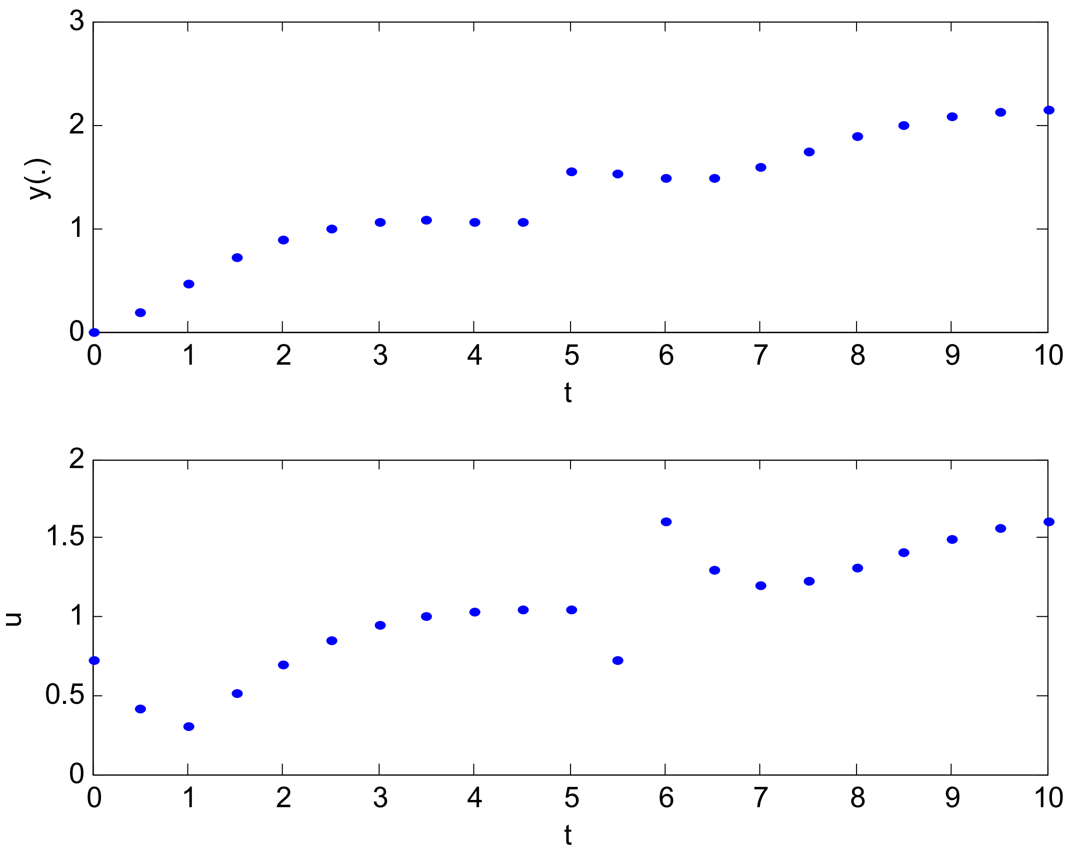
Reference signal and load disturbance



Step in reference compensated by a pre-compensator (mc).

Noise term is 0 (at the process output).

Reference signal and load disturbance



A stepwise load disturbance occurs at the time $t = 5$.

A permanent error!

Reference signal and load disturbance

Bad; load disturbances cannot be compensated by a pre-compensator. What to do?

Hmm. Integration?

Add an integratorto the controller. But how?

Look at the process representation

$$x(k+1) = \Phi x(k) + \Gamma u(k)$$
$$y(k) = Cx(k)$$

Reference signal and load disturbance

Augment the state by introducing $x_{n+1}(k)$

when n is the original state dimension. Choose

$$\Delta x_{n+1}(k) = x_{n+1}(k) - x_{n+1}(k-1) = h\left[y(k-1) - r(k-1)\right]$$

where r is the reference signal. Here the error signal is integrated.

By combining the states we obtain (note the block matrices)

Reference signal and load disturbance

$$\begin{bmatrix} x(k+1) \\ x_{n+1}(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} \Phi & 0 \\ hC & 1 \end{bmatrix}}_{\Phi_{aug}} \begin{bmatrix} x(k) \\ x_{n+1}(k) \end{bmatrix} + \underbrace{\begin{bmatrix} \Gamma \\ 0 \end{bmatrix}}_{\Gamma_{aug}} u(k) + \begin{bmatrix} 0 \\ -h \end{bmatrix} r(k)$$

Design a state feedback controller

$$u(k) = -\begin{bmatrix} L & l_{n+1} \end{bmatrix} \begin{bmatrix} x(k) \\ x_{n+1}(k) \end{bmatrix} = -L_{uusi} \begin{bmatrix} x(k) \\ x_{n+1}(k) \end{bmatrix}$$

$$L_{uusi} = \text{place}(\Phi_{aug}, \Gamma_{aug}, p_{aug});$$

Note. New L and new p contain n+1 components.

Reference signal and load disturbance

The closed loop system becomes

$$\begin{bmatrix} x(k+1) \\ x_{n+1}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi - \Gamma L & -\Gamma l_{n+1} \\ hC & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ x_{n+1}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ -h \end{bmatrix} r(k)$$

where the eigenvalues of the system matrix are inside the unit circle. If r is relatively ”slow”, the state components approach some constant values. But then

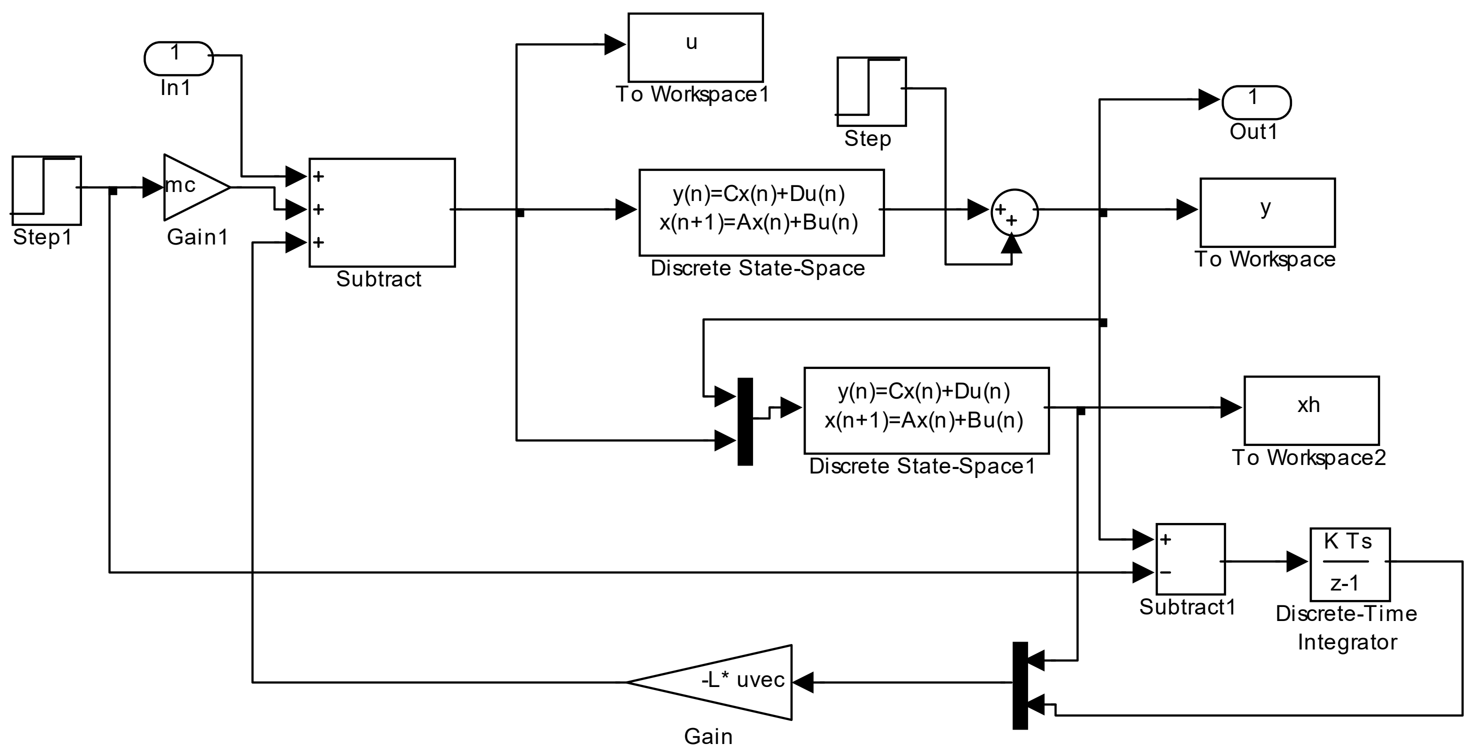
Reference signal and load disturbance

$$\Delta x_{n+1}(k) = x_{n+1}(k) - x_{n+1}(k-1) = h\left[y(k-1) - r(k-1)\right]$$

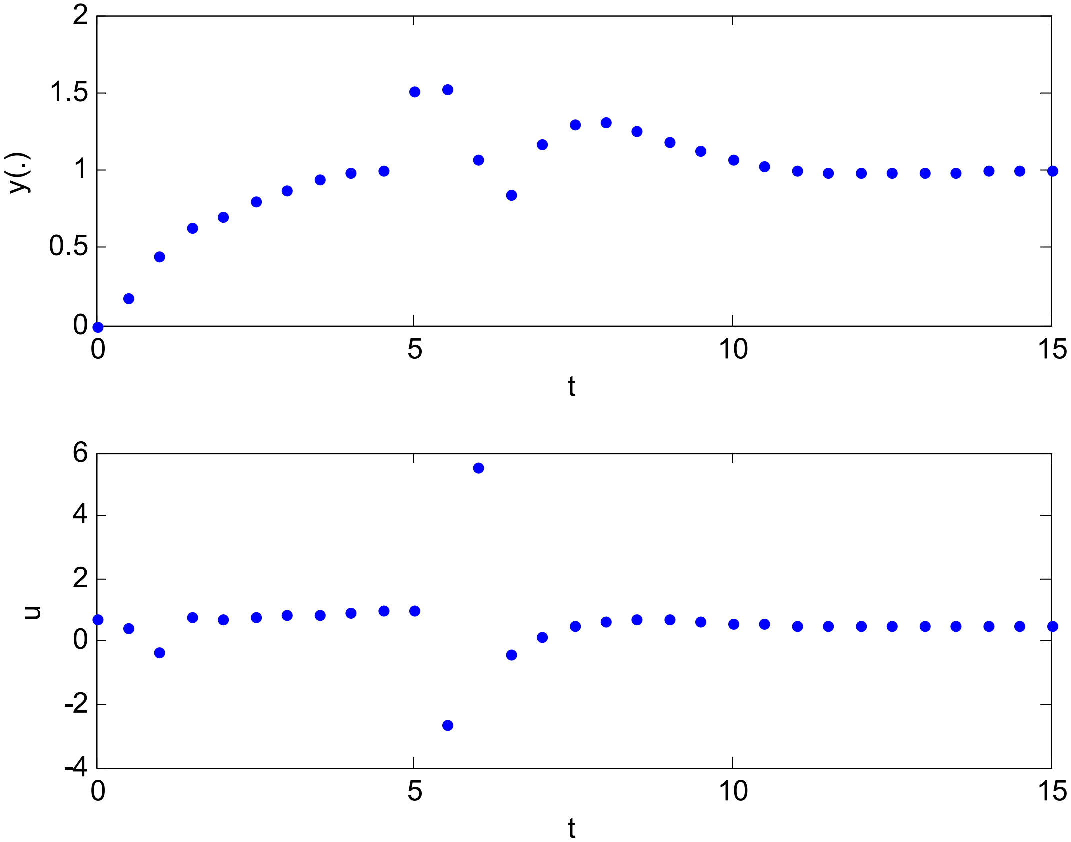
approaches zero; the output follows the reference.

Note that when adding a new state we did not change the process (that would not do). The new state is basically part of the controller in the end. But mathematically it is possible to consider all these states together.

Reference signal and load disturbance



Reference signal and load disturbance



The effect of the load disturbance is removed by the integrator.