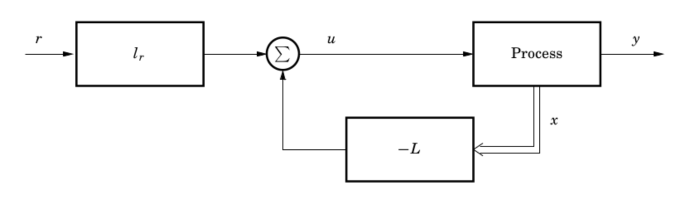
Lec 7: Observers, Observability, Output Feedback, Pole/Zero Cancellation

Charlotta Johnsson

Nov 28, 2017

Lund University, Department of Automatic Control

Previous lecture: State feedback



Feedback signal as a linear combination of the states:

$$u = -l_1 x_1 - l_2 x_2 - \cdots - l_n x_n = -Lx$$

Gives closed loop system matrix

$$A_{cl} = A - BL$$

Previous lecture: State feedback

Feedback signal as a linear combination of the states:

$$u = -l_1 x_1 - l_2 x_2 - \cdots - l_n x_n = -Lx$$

Gives closed loop system matrix

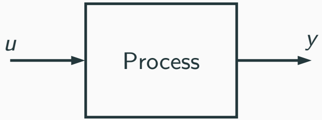
$$A_{cl} = A - BL$$

If the system is *controllable*, we can place closed loop poles (eigenvalues of $A - BL$) arbitrarily

One big problem with this approach. . . typically not all states x_i are measured

This lecture: Observers

Key idea:
System model + output signal y + control signal $u \rightarrow$ Estimate $\hat{x}$ of x



The Kalman filter

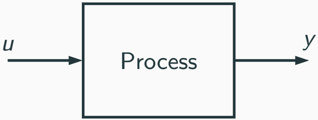
developed c. 1960 by
Rudolf Kalman (1930-2016)

Used in the Apollo navigation computer

Applications: automatic control, radar tracking, medical imaging, seismology, battery charge estimation, economics, online parameter estimation etc, etc

This lecture: Observers

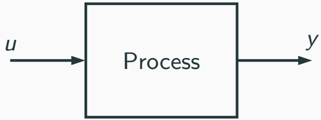
Key idea:
System model + output signal y + control signal $u \rightarrow$ Estimate $\hat{x}$ of x



This lecture: Observers

Key idea:

System model + output signal y + control signal $u \rightarrow$ Estimate $\hat{x}$ of x



We will design observers using pole placements.

Similar to what we did for state feedback.
Dual problems, i.e. "Same, same, but different"

Observability

Is it always possible to estimate the state of a system from u and y?

Yes — if the system is observable

Definition: A state vector $x_0 \neq 0$ is not observable if the output is $y(t) = 0$ when the initial state vector is $x(0) = x_0$ and the input is given by $u(t) = 0$. A system is *observable* if it lacks non-observable states.

Test for observability

Test for observability: The *observability matrix*

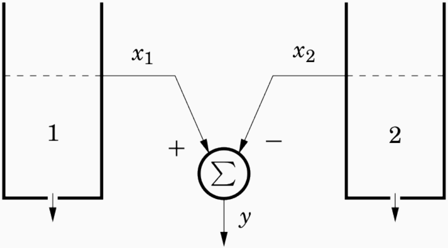
$$W_o = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

has n (= number of states), linearly independent columns

Note that:

- Observability only depends on A and C
- Non-observable states x_0 satisfy the equation $W_o x_0 = 0$

Example: Observability of water tanks (1/2)

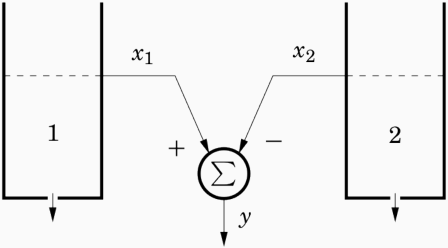


State-space model:

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

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

Example: Observability of water tanks (1/2)



State-space model:

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

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

Observability matrix:

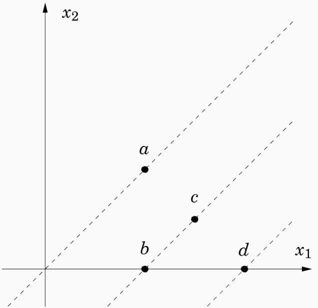
$$W_o = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

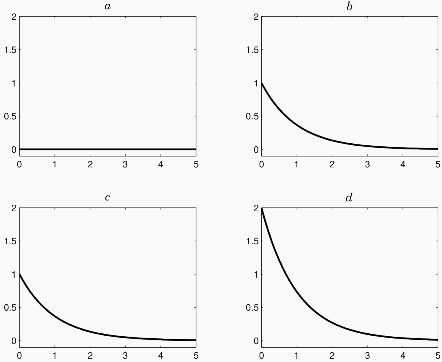
W_o has rank 1 $\Rightarrow$ system is not observable

Example: Observability of water tanks (2/2)

The non-observable states satisfy $W_o x_0 = 0$, i.e. the non-observable states are given by

$$x_0 = \begin{bmatrix} a \\ a \end{bmatrix}$$





State-estimation: Notation

Want to estimate the state x of system

$$\frac{d}{dt}x = Ax + Bu$$

Introduce

- $\hat{x}$ - estimated state vector
- $\tilde{x} = x - \hat{x}$ - estimation error

State-estimation: Via simulation

Let the state estimate evolve according to

$$\frac{d}{dt}\hat{x} = A\hat{x} + Bu$$

The estimation error evolves according to

$$\begin{aligned}\frac{d}{dt}\tilde{x} &= \frac{d}{dt}\left(x - \hat{x}\right)\\ &= Ax + Bu - (A\hat{x} + Bu)\\ &= A\left(x - \hat{x}\right) = A\tilde{x}\end{aligned}$$

- Estimation error converges to 0 if A is stable
- Convergence rate depends on eigenvalues of A
- Requires perfect model and no load disturbances
- Information in measured signal y is not used

State-estimation: Via observer (1/2)

Let the state estimate take y into account

$$\begin{aligned}\frac{d}{dt}\hat{x} &= A\hat{x} + Bu + K(y - \hat{y})\\ \hat{y} &= C\hat{x}\end{aligned}$$

or

$$\frac{d}{dt}\hat{x} = (A - KC)\hat{x} + Bu + Ky$$

State-estimation: Via observer (1/2)

Let the state estimate take y into account

$$\frac{d}{dt}\hat{x} = A\hat{x} + Bu + K(y - \hat{y})$$
$$\hat{y} = C\hat{x}$$

or

$$\frac{d}{dt}\hat{x} = (A - KC)\hat{x} + Bu + Ky$$

The estimation error evolves according to

$$\begin{aligned}\frac{d}{dt}\tilde{x} &= \frac{d}{dt}(x - \hat{x})\\ &= Ax + Bu - ((A - KC)\hat{x} + Bu + KCx)\\ &= (A - KC)(x - \hat{x}) = (A - KC)\tilde{x}\end{aligned}$$

State-estimation: Via observer (1/2)

Let the state estimate take y into account

$$\frac{d}{dt}\hat{x} = A\hat{x} + Bu + K(y - \hat{y})$$
$$\hat{y} = C\hat{x}$$

or

$$\frac{d}{dt}\hat{x} = (A - KC)\hat{x} + Bu + Ky$$

The estimation error evolves according to

$$\begin{aligned}\frac{d}{dt}\tilde{x} &= \frac{d}{dt}(x - \hat{x}) \\ &= Ax + Bu - ((A - KC)\hat{x} + Bu + KCx) \\ &= (A - KC)(x - \hat{x}) = (A - KC)\tilde{x}\end{aligned}$$

By choosing K we can affect convergence speed of the state estimate

State-estimation: Via observer (2/2)

$$\frac{d}{dt}\tilde{x} = (A - KC)\tilde{x}$$

Poles are placed by choosing K, same as for state-feedback

Large K, fast poles of $A - KC$

- Fast convergence of state estimation
- Sensitive to measurement noise

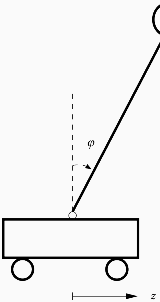
Small K, slow poles of $A - KC$

- Slow convergence of state estimate
- sensitive to load disturbances and modeling errors

As always: Trade-off between robustness and performance

Inverted pendulum example (1/2)

Process:

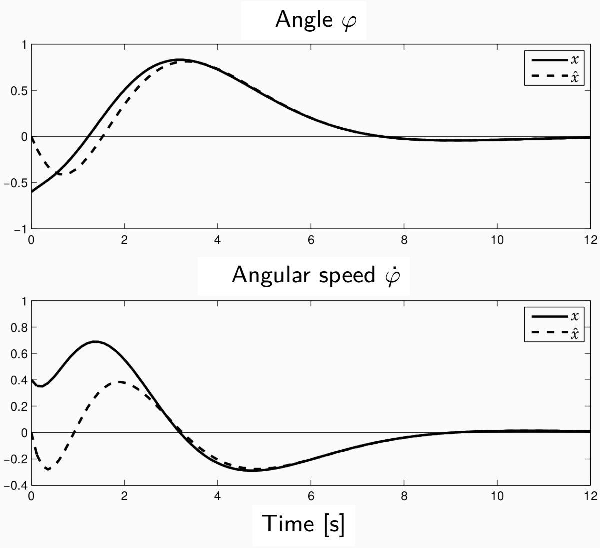


$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ -1 \end{bmatrix} u = Ax + Bu$$

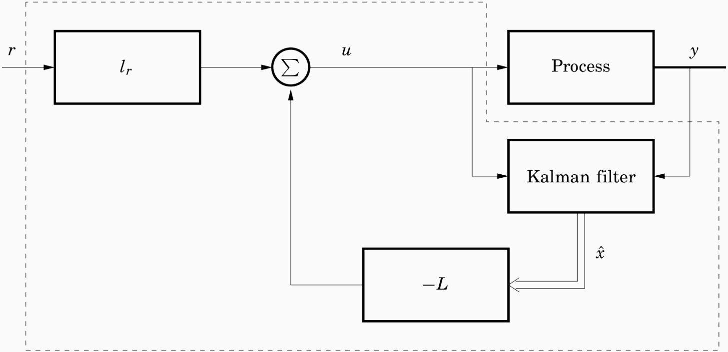
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x = Cx$$

Inverted pendulum example (2/2)

Simulation from initial state $\varphi(0) = -0.6$, $\dot{\varphi}(0) = 0.4$



Output Feedback



Process:

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

Kalman filter + Controller:

$$\dot{\hat{x}} = A\hat{x} + Bu + K(y - \hat{y})$$

$$\hat{y} = C\hat{x}$$

$$u = l_r r - L\hat{x}$$

Output Feedback (2/2)

Process:

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

Kalman filter + Controller:

$$\dot{\hat{x}} = A\hat{x} + Bu + K(y - \hat{y})$$

$$\hat{y} = C\hat{x}$$

$$u = l_r r - L\hat{x}$$

Introduce state-vector extended with estimation errors $x_e = \begin{bmatrix} x \\ \tilde{x} \end{bmatrix}$

Closed loop state-space equations become:

$$\begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} A - BL & BL \\ 0 & A - KC \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + \begin{bmatrix} Bl_r \\ 0 \end{bmatrix} r = A_e \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + B_e r$$

$$y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} = C_e \begin{bmatrix} x \\ \tilde{x} \end{bmatrix}$$

Ouput Feedback: Closed loop dynamics

Characteristic polynomial of closed loop system:

$$\det\left(\begin{bmatrix} A - BL & BL \\ 0 & A - KC \end{bmatrix}\right) = \det(sI - (A - BL)) \cdot \det(sI - (A - KC))$$

Possible to place poles for state feedback and the observer independently!!

Ouput Feedback: Closed loop dynamics

Characteristic polynomial of closed loop system:

$$\det\left(\begin{bmatrix} A - BL & BL \\ 0 & A - KC \end{bmatrix}\right) = \det(sI - (A - BL)) \cdot \det(sI - (A - KC))$$

Possible to place poles for state feedback and the observer independently!!

Can show that the transfer function $r \to y$ is

$$G_{r\to y}(s) = C(sI - (A - BL))^{-1} B l_r$$

I.e. same as for state feedback!
Reason: After convergence of the Kalman filter, estimated state equals true state. (Problems with load disturbances and modeling errors)

Ouput Feedback: Closed loop dynamics

Characteristic polynomial of closed loop system:

$$\det\left(\begin{bmatrix} A - BL & BL \\ 0 & A - KC \end{bmatrix}\right) = \det(sI - (A - BL)) \cdot \det(sI - (A - KC))$$

Possible to place poles for state feedback and the observer independently!!

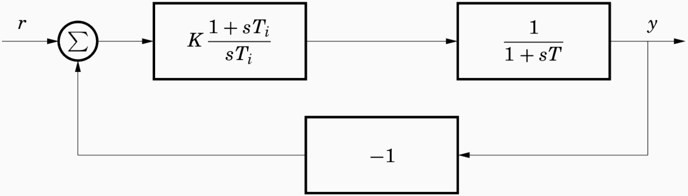
Can show that the transfer function $r \to y$ is

$$G_{r \to y}(s) = C(sI - (A - BL))^{-1} Bl_r$$

I.e. same as for state feedback!
Reason: After convergence of the Kalman filter, estimated state equals true state. (Problems with load disturbances and modeling errors)

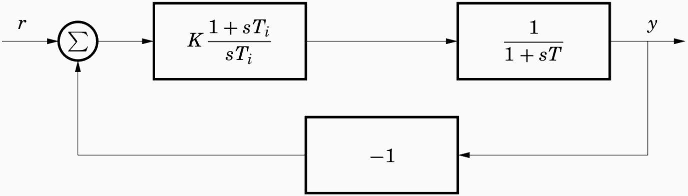
Rule of thumb: Observer poles twice as fast as state feedback poles

Cancellation of Poles and Zeros



Process $G_P(s) = \frac{1}{1+sT}$, PI-controller $G_R(s) = K\left(1 + \frac{1}{sT_i}\right)$

Cancellation of Poles and Zeros

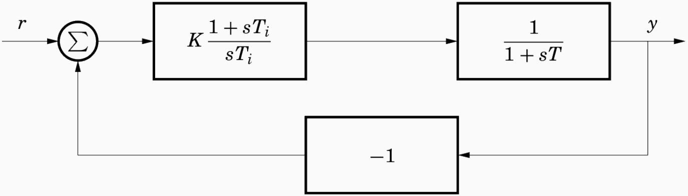


Process $G_P(s) = \dfrac{1}{1+sT}$, PI-controller $G_R(s) = K\left(1 + \dfrac{1}{sT_i}\right)$

Many tuning rules for PI-control specify $T_i = T$, resulting in

open loop system $G_0(s) = \dfrac{K(1+sT)}{sT}\dfrac{1}{(1+sT)} = \dfrac{K}{sT}$

Cancellation of Poles and Zeros



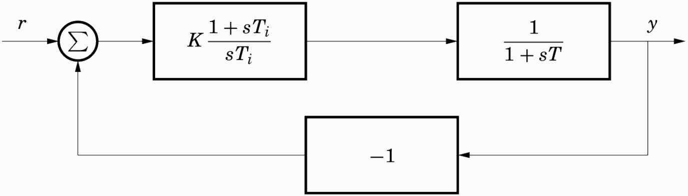
Process $G_P(s) = \dfrac{1}{1+sT}$, PI-controller $G_R(s) = K\left(1 + \dfrac{1}{sT_i}\right)$

Many tuning rules for PI-control specify $T_i = T$, resulting in

open loop system $G_0(s) = \dfrac{K(1+sT)}{sT}\dfrac{1}{(1+sT)} = \dfrac{K}{sT}$

closed loop system $G(s) = \dfrac{K}{K+sT}$

Cancellation of Poles and Zeros



Process $G_P(s) = \dfrac{1}{1+sT}$, PI-controller $G_R(s) = K\left(1 + \dfrac{1}{sT_i}\right)$

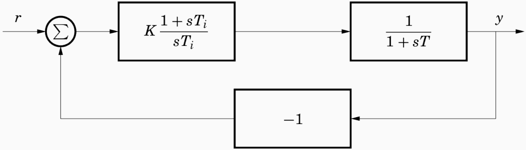
Many tuning rules for PI-control specify $T_i = T$, resulting in

open loop system $G_0(s) = \dfrac{K(1+sT)}{sT}\dfrac{1}{(1+sT)} = \dfrac{K}{sT}$

closed loop system $G(s) = \dfrac{K}{K+sT}$

NOTE: pole/zero cancellation in $G_0(s)$

Cancellation of Poles and Zeros



open loop system $G_0(s) = \frac{K}{sT}$ closed loop system $G(s) = \frac{K}{K+sT}$

Resulting in the transfer functions

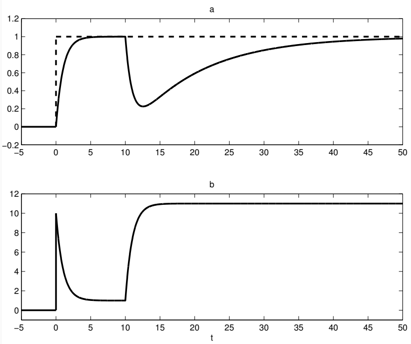
$$Y(s) = \frac{K}{K+sT}R(s) + \frac{K}{(K+sT)(1+sT)}L(s)$$

NOTE: the pole/zero cancellation shows up in the load-disturbance

Cancellation of Poles and Zeros

Transfer functions

$$Y(s) = \frac{K}{K + sT} R(s) + \frac{K}{(K + sT)(1 + sT)} L(s)$$



NOTE: the pole/zero cancellation shows up in the load-disturbance.