

# Introduction to Computer Control Systems

## Computer exercise 2

### Observer-based state feedback control

**Reading instructions:** Glad-Ljung, Chapters 3 and 5.

|                  |              |                      |
|------------------|--------------|----------------------|
| Name             |              | Assistant's comments |
| Program          | Year of reg. |                      |
| Date             |              |                      |
| Passed prep. ex. | Sign         |                      |
| Passed comp. ex. | Sign         |                      |

## 1 Introduction

This computer simulation exercise provides an introduction to the process lab exercise 3. In computer exercise 1 and process lab 2, the characteristics and performance of the PID controller has been discussed. Here another kind of controller will be addressed, *i.e.* state feedback controller designed by pole placement. To implement a closed-loop state feedback, information about system states has to be available. In the case when only some of the states are measured, a state observer is required.

## 2 Pole placement: State feedback with an observer

One way to design a controller is through pole placement, *i.e.* by assigning the roots of the characteristic polynomial of the closed-loop system to pre-defined locations in the complex plane. An advantage of this approach is that stability of the closed-loop system is guaranteed by placing the poles in the stable region, *i.e.* in the left half plane for continuous time system and inside the unit circle for discrete time system. Pole placement may be implemented using state feedback. If the states are not readily available, it is instead possible to use an estimated state vector in the feedback. This estimate is obtained by the use of an observer. Thus, the stability of the closed-loop system would also depend on the model utilized in the observer. Modeling error might significantly degrade the observer performance and even lead to instability. In the following, a state feedback of observer states will be used for pole placement of the system.

### 2.1 Controllability and observability

From computer exercise 1, the linearized model of the vehicle is given by

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

where

$$\begin{aligned}A &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1/T & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ K/T \\ 0 \end{bmatrix} \\ C &= [L \quad 0 \quad v], \quad D = 0\end{aligned}$$

with  $T = 0.15$ ,  $K = 0.0262$ ,  $L = 0.075$ , and  $v = 0.1$

To implement state feedback controller, the system has to fulfill the controllability condition, *i.e.* the controllability matrix  $\mathcal{S}$  has full rank, where

$$\mathcal{S} = (B \quad AB \quad \dots \quad A^{n-1}B)$$

**Exercise 2.1:** Check the controllability condition of the system. Is the system controllable?

**Answer:**

As it has been mentioned before, to implement state-feedback control, one needs information about system states. Typically, only some of the plant states are measurable. Thus, a state observer is required. To design an observer, the first thing to do is to check whether the system is observable or not, *i.e.* the observability matrix  $\mathcal{O}$

$$\mathcal{O} = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix}$$

has full rank.

**Exercise 2.2:** Check the observability condition of the system. Is the system observable?

**Answer:**

## 2.2 State feedback with pole placement

Consider a system with process disturbance  $v(t)$  and measurement disturbance  $w(t)$

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Nv(t) \\ y(t) &= Cx(t) + w(t). \end{aligned} \tag{1}$$

Let  $r(t)$  is the reference value for output signal and  $K$  is the control feedback gain, the state feedback is then defined as

$$u(t) = -Kx(t) + mr(t).$$

Here,  $m$  is a constant gain which is needed to ensure that the static gain of the controlled system, from the reference signal to the output signal, is equal to the desired value. Thus the closed-loop system is given by

$$\begin{aligned}\dot{x}(t) &= (A - BK)x(t) + Bmr(t) + Nv(t) \\ y(t) &= Cx(t) + w(t)\end{aligned}$$

By implementing a pole placement method, the closed-loop system poles are assigned to the stable region thus rendering the closed-loop system asymptotically stable. Pole placement for single-input single-output systems is easily implemented by means of Ackermann's formula and performed in Matlab with the command `acker(A,B,p_s)`, where `p_s` represents the desired closed-loop system pole(s). However, the state feedback control approach demands that all the states are measurable.

### 2.3 Observer design

The idea of state observer is to obtain the estimated states  $\hat{x}(t)$  given information about the output signal  $y(t)$ , the input signal  $u(t)$  and a system model. With observer gain  $L$ , the state equation of an observer based on (1) is given by

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - C\hat{x}(t)).$$

Define the state estimation error as  $\tilde{x}(t) = x(t) - \hat{x}(t)$ . Then the dynamics of estimation error is described by

$$\dot{\tilde{x}}(t) = (A - LC)\tilde{x}(t) + Nv(t) - Lw(t)$$

The poles of the observer are then given by the eigenvalues of  $(A - LC)$ . To obtain a faster observer, the poles should be placed further from the origin in the left half complex plane. A faster observer would respond faster to the difference between  $x(t)$  and  $\hat{x}(t)$ , however it would also be more sensitive to measurement noise  $w(t)$  because it would also get amplified by the observer gain  $L$ . The rule of thumb for choosing the poles of the observer is that the observer's dynamics should be faster than the state feedback controller's dynamics to compensate the estimation error on the controller performance.

Quite similar with the state feedback design, an observer could be designed based on pole placement with the Matlab command `acker()`. Since the command follows a generic algorithm for a given pair of matrices  $A, B$ , small adjustments is needed for observer design. (Hint:  $(A - LC)^T = A^T - C^T L^T$ )

## 2.4 Closed-loop system representation of state feedback with observer

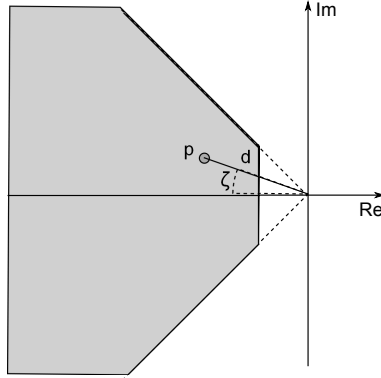
The closed-loop system with the state observer and state feedback controller is then defined by the following state space model

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \end{bmatrix} &= \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + \begin{bmatrix} Bm \\ 0 \end{bmatrix} r + \begin{bmatrix} N \\ N \end{bmatrix} v + \begin{bmatrix} 0 \\ -L \end{bmatrix} w \\ y &= [C \ 0] \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + w \end{aligned} \quad (2)$$

Fig. 1 hints on how to place the poles to obtain asymptotically stable closed-loop system. Generally speaking, the poles have to be inside left half planes of the complex planes, *i.e.* all poles must have negative real part. The distance between the poles and the origin determines how fast the dynamics of the closed-loop system are. To provide fast response, pole should be placed far away from the origin, however would require very large input signals. Thus, a trade-off between fast response and input signal magnitude should be considered in the design procedure.

If the absolute value of one pole is small compared to the others, meaning that this pole is the closest to the origin, the decay rate corresponding to this pole will be the slowest one. Since the influence of the other poles will decay faster, the closest to the origin pole will dominate the behavior of the system. Such a pole is referred to as a dominating pole.

The angle  $\zeta$  in Fig. 1 decides the oscillatory behavior of the system. Higher angle would lead to a more oscillatory response.



**Figure 1:** A pole  $\mathbf{p}$  with distance  $\mathbf{d}$  to the origin and angle  $\zeta$ . The shaded area shows where it is usually safe to place poles in continuous time systems.

The static gain from reference signal to output signal is given by  $G(0) = m(C(-A + BK)^{-1}B)$ .

**Exercise 2.3:** Define  $m$  in terms of  $A, B, C$ , and  $K$  so that the static gain of the closed loop system is equal to one.

**Answer:**

For the following task, the file `sim_lab2.m` could be used as a template.

**Task:** Construct an LTI object for the closed-loop system using the `ss()` command. Place the state feedback poles  $\mathbf{p}_s$  on the real axis so that they are single and lie in the interval  $[-15, -1]$ . Choose the observer poles  $\mathbf{p}_o$  to be 10% further away from the origin than the state feedback poles. Run `run_sim_lab2()` to see the step response of the closed-loop system. Try different values of the poles.

**Exercise 2.4:** How do the poles influence the response of the system?

**Answer:**

**Task:** Try also different values of the observer poles and study how they influence the states estimates.

**Exercise 2.5:** *How do the observer poles influence the state estimates?*

**Answer:**

**Task:** Now place the state feedback poles  $\mathbf{p}_s$  at  $p_{1,2} = -5 \pm \beta i$ ,  $p_3 = -6$ . Vary the value of  $\beta$  and simulate the step response.

**Exercise 2.6:** *How does the value of  $\beta$  influence the step response?*

**Answer:**

## A List of matlab commands

`rank(M)` calculates the rank of matrix M.

`ss(A,B,C,D)` creates a continuous system in state space form.

`acker(A,B,p)` pole placement design based on Ackermann's formula.

`run_sim_lab2(Ac1,Bc1,Cc1,Dc1,K,L,m)` simulate the step response of closed-loop system