Introduction to computer based control systems
Process lab 3

Name:
Program:
Date:
Approved: Sign

UPPSALA UNIVERSITY
DEPARTMENT OF INFORMATION TECHNOLOGY
SYSTEMS AND CONTROL

Preparation: Read these instructions carefully. Read up on observers and state feedback, see Glad-Ljung chapters 3 and 5.

1 Introduction

In this lab you will design an observer and also a controller for the robot using pole placement. First an observer will be designed that performs estimation of the systems states.

The poles will then be placed at different location using state feedback, and you will study how the system behaves for these different placements of the poles.

2 Start

Open Matlab and get to the lab directory (it will be specified on the lab). Type **addLabPath** in Matlabs command prompt and hit enter to get access to all required m-files.

Calibrate the light sensor. In this lab model-based control will be implemented and it is important that the model agrees with the actual system. The model assumes that negative y direction is to the left of the vehicle and your calibration must be consistent with this definition.

- Execute **[y l] = calibrateLS(1E-3*[-10 -5 0 5 10])** in response to Matlab's command prompt. Make sure that negative values of track deviation are to the left side of the vehicle, looking in the direction of travel.
- Place the sensor at the specified values of y and hit enter.
- Determine a_0 and a_1 by calling $polyfit(y, l, 1)$. Write down these values for later use.

3 The system

The system model derived in process lab 2 and was found to be

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u \\ y &= \mathbf{C}\mathbf{x} + \mathbf{D}u \end{aligned}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1/T & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ K/T \\ 0 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} L & 0 & v \end{bmatrix}$$

$$\mathbf{D} = \mathbf{0}$$

The parameters v, L, K and T were determined by using system identification techniques. Since these parameters differ somewhat from robot to robot, and it is not guaranteed that you will have the same robot for this session as for process lab 2, the parameter values for each robot have been determined beforehand and written in a note attached to it.

4 Controllability and Observability

In computer lab 2 , the system was found to be both controllable and observable. The latter property allows an observer to estimate of all system states. Also the controllability of the system assures that the poles can be placed at arbitrary locations using state feedback.

5 Observer

The system model has three state variables, but actually none of them is measured and therefore all of them have to be estimated. An observer takes in the measurements of y and u and gives as output an estimate $\hat{\mathbf{x}}$ of the system state $\mathbf{x}$ as shown in Fig 1.

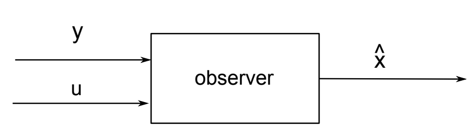

Figure 1: Block diagram for observer system

A linear observer for the system model at hand is given by

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}u + \mathbf{L}(y - \mathbf{C}\hat{\mathbf{x}}).$$

In terms of the state estimation error $\tilde{\mathbf{x}} = \hat{\mathbf{x}} - \mathbf{x}$, the dynamics of the observer will be given by (see Glad-Ljung, page 124 chapter 5.7)

$$\dot{\tilde{\mathbf{x}}} = (\mathbf{A} - \mathbf{CL})\tilde{\mathbf{x}} + \mathbf{N}v_1 + \mathbf{L}v_2$$

where v_1 is the process noise and v_2 is the measurement noise entering the system according to

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u + v_1 \\ y &= \mathbf{C}\mathbf{x} + v_2. \end{aligned}$$

The poles of the system describing the state estimation error are thus given by the eigenvalues of $(\mathbf{A} - \mathbf{CL})$. By placing these poles far into the l.h.s. plane will give an observer that responds quickly when $\hat{\mathbf{x}}$ differs from $\mathbf{x}$. However, it will also amplify the measurement noise v_2. There is thus a trade off between making the observer fast and insensitive to measurement noise.

Task: Open the m-file **lineTrackerPole.m** by typing **open lineTracker-Pole** in Matlabs command prompt and hit enter. In this file a state feedback controller and an observer with the poles $\mathbf{p}_0$ is implemented. During execution, the m-file will run the state feedback controller that keeps the vehicle on the track, and also plot the estimated states $\hat{\mathbf{x}}$. Your task is as follows:

- Look at the plot and determine whether the observer is too slow or too noise sensitive.
- According to your conclusion about the observer speed and noise sensitivity, adjust the placement of the poles by scaling them with a constant $\alpha > 0$.
- Iterate until you find a value of α that you are satisfied with.

Show the plot to the lab assistant when you have found a satisfactory α.

Question: What value of α have you arrived to?

6 Pole placement with state feedback

Provided that the system is controllable, the poles of the closed loop system can be placed at arbitrarily locations. In this part you will place the poles of the system using state feedback. But first of all you will look at where the poles of your system were placed with the PID regulator designed in process lab 2. The poles of the closed loop system are given by the poles of $G_c(s) = \frac{F(s)G(s)}{1+F(s)G(s)}$. The function $pidPoles(sys, K_P, K_I, K_D)$ (sys is given in lineTrackerPole by sys_c) will return the poles for the system with state space matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ and the PID regulator with parameters K_P,K_I and K_D .

Task:

Run the Matlab function pidPoles to see where the poles for the PID regulator were placed. If you do not remember your parameters, use $K_P = 3000$, $K_I = 8000$, $K_D = 600$.

Question: Where were the poles for the closed loop system placed with the PID controller?

Under state feedback, the control signal is given by the law

$$u = -\mathbf{Kx}$$

The block diagram for such a feedback controller is shown in Fig. 2.

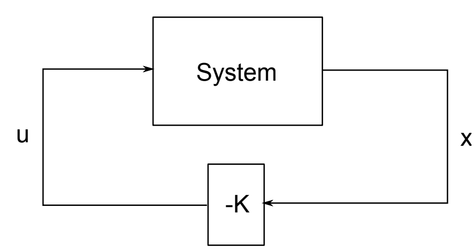

Figure 2: Block diagram for state feedback.

The closed loop system is then given by

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{B}u = \mathbf{Ax} - \mathbf{BKx} = (\mathbf{A} - \mathbf{BK})\mathbf{x} \\ y &= \mathbf{Cx} \end{aligned}$$

and the poles of the close- loop system are thus the eigenvalues of $(\mathbf{A}-\mathbf{BK})$. To place the poles at the desired locations, $\mathbf{K}$ should be chosen so that $(\mathbf{A}-\mathbf{BK})$ has the corresponding eigenvalues.

6.1 Where to place the poles

Fig. 3 shows where the poles should usually be located in order to obtain a reasonably good behavior of the regulated system.

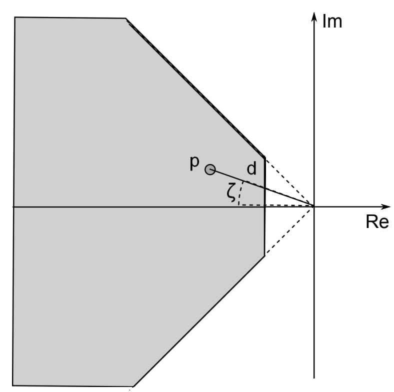

Figure 3: A pole p with distance d to the origin, and angle ζ. The shaded area shows where it is usually good to place the poles.

To get a stable closed-loop system, which is the primal objective when a regulator is designed, the poles must be located in the l.h.s. of the complex plane, i.e. all poles must have negative real part.

How fast the modes of the system will respond is determined by the distance from the origin to the poles d (See Fig 3). Poles far away from the origin give a fast response, but such a placement requires large control signals. There is therefore a trade off between making the system fast and not saturating the control signal. The behavior of the system will be dominated by the poles that are closest to the origin. This means that if there would be a pole very close to the origin, the behavior of the system can be expected to be slow.

The larger the angle ζ is (See Fig 3) the more oscillatory can the system response will be.

6.2 State feedback with observer

Since the state variables are not measured, the observer is required to find an estimate of $\mathbf{x}$. The block diagram in Fig. 4 shows the connection of the feedback matrix $\mathbf{K}$ and the observer.

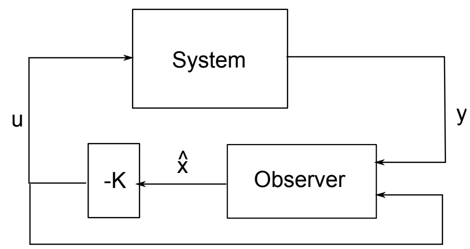

Figure 4: State feedback with observer.

Task: Change the poles of the closed-loop system in **lineTrackerPole.m**. Place the poles on the real axis at different distances from the origin in the interval $[-15, -1]$ and test the obtained regulator on the track. Place the observer poles 1.1 to 3 times further away from the origin than the poles of the closed-loop system.

Question: Describe the behavior of the system and magnitude for the control signal for your pole placements alternatives.

Task: Place the poles at $p_{1,2} = -5 \pm \beta i$, $p_3 = -6$. Note that the imaginary unit is given by **1i** in Matlab. Vary β and try out the obtained regulator on the track. Describe the behavior of the system for different magnitudes of the imaginary part of the poles for the system you have tested.

Question: What is the stability criterion for the placement of the poles?

Task: Now place poles $p_1 = -3$, $p_2 = -4$. Place the third pole somewhere in the interval $p_3 \in [-1, 1]$ and investigate where instability of the system occurs.

Question: Why is the border of stability not located at $p_3 = 0$?

Question: Have you noticed that there is a static error in the control loop in the previous task?

6.3 Introducing integration

When using pole placement with state feedback, there is no integral action provided by the regulator, which means that there will be a static error in the control signal. To handle this, a fictitious state $\epsilon(t) = \int_0^t e(\tau)d\tau = \int_0^t -y(\tau)d\tau$ can be introduced in the state space model which gives the augmented system

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\epsilon} \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & 0 \\ -\mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \epsilon \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} u \\ y &= \begin{bmatrix} \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \epsilon \end{bmatrix} \end{aligned}$$

Since a state feedback will try to force the states to zero, the integrated error will become zero if a regulator is designed for the augmented system. The resulting controller will thus have integral action that eliminates the static error.

Task: Open **lineTrackerPoleInt** by typing **open lineTrackerPoleInt** in Matlabs command prompt and hit enter. In this file, the augmented system will be provided. Place the poles on the real axis somewhere in the interval $[-10, -4]$. Note that the augmented system has 4 states and hence 4 poles should be taken care of. Try the regulator on the track.

Question: Is there still a static error?