TP1 cours CAVAN: Bouclage, robustesse et incertitudes

E. Kamal and B. Marinescu

Exercise 1

Let

$$G(s) = \frac{1}{s^2 + 0.2683s + 5}, \quad (1)$$

Let be a second order LTI transfer function. The goal is to design a controller K (s) for (1), based on the Loop-Shaping control strategy, in order to achieve satisfactory performances and to minimize the control effort as well as the effect of the measurement noise. The performances are specified in the frequency domain in terms of the steady-state error ε, the frequency bandwidth w_b and a maximum overshoot M. For this, let

$$\begin{cases} S = \dfrac{1}{1+L} \\ T = \dfrac{L}{1+L} \end{cases}$$

be the sensitivity and the complementary sensitivity functions with L = K (s)G(s).

I. What is the link between S and T?
II. Is it possible to independently impose constraints on S and T?
III. Now, to design the controller K (s), such that the objectives mentioned above can be taken into account, let us define the following frequency weighting function

$$W_1(s) = \frac{\frac{1}{M}s + w_b}{s + w_b * \varepsilon} \quad \textbf{(2)}$$

To achieve the following performances (for the closed-loop system)

- A frequency bandwidth of w_b =1.5 rad/s,
- A steady-state error of $\varepsilon = 10^{-4}$ within the specified frequency bandwidth.
- Maximum overshoot M lower or equal to 2,

Let $W_2 = 0.1$ and $W_3 = 0$.

IV. Plot the Bode Diagram of $W_1(s)$.
   1) Find the H_∞ controller (use augw and hinfsyn functions of Matlab) K(s) which can satisfy the above specifications

2) Fix the reference signal r to r = 1 and add an output-step disturbance dy of a measurement white noise η of power 10^{-8} and sample time 10^{-3}. Test the performances of the closed-loop system in this case and compare with the open-loop.
3) With dy = 0, increase the frequency bandwidth w_b to 5 rad/s.
4) In order to minimise the magnitude of the input u and the effect of the measurement noise η, let W_2 (s) = A, with A = 10^{-2}.
5) Find the H_∞ controller K(s)
6) Increase the value of A to 1 and look at the values of resulting from the optimization function hinfsyn of Matlab.

Replace the controller H_∞ controller by H_2 controller found in steps (1) and (6) and (use augw and h2syn functions of Matlab) and compare the performances of the resulting closed-loop system with the ones of the previous closed-loop for all the above values.

Exercise 2 : SIMULATION EXAMPLE: AN AIRCRAFT OBSERVER DESIGNS

The corresponding block diagram of the system-observer configuration, also known as the observer based controller configuration, is presented in Figure 1 with the feedback loop closed using the feedback controller as defined as

$$u(t)=-G\hat{x}(t) \tag{3}$$

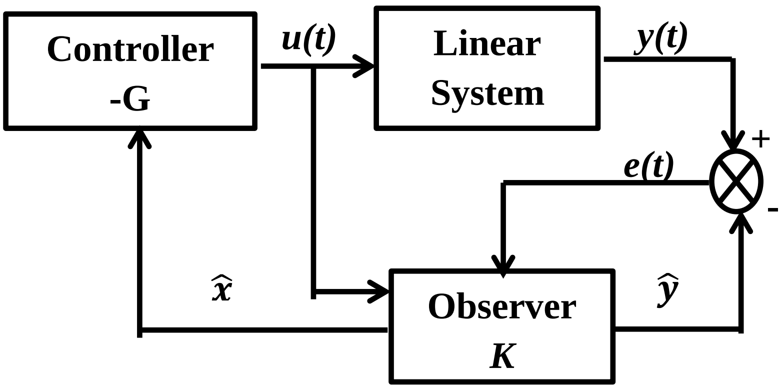

Figure 1

In this TP we consider an aircraft mathematical model taken from [1], whose state space equations are given by

$$\dot{x}(t) = Ax(t) + Bu(t) \ , \qquad y(t) = Cx(t) \tag{4}$$

Where,

$$A = \begin{bmatrix} -0.01357 & -32.2 & -46.3 & 0 \\ 0.00012 & 0 & 1.214 & 0 \\ -0.0001212 & 0 & -1.214 & 1 \\ 0.00057 & 0 & -0.1 & -0.6696 \end{bmatrix},$$

$$B = \begin{bmatrix} -0.433 \\ 0.1394 \\ -0.1394 \\ -0.1577 \end{bmatrix} \quad , C = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$

The obtained differences between the actual state trajectories (*x(t)*) and the estimated state trajectories ($\hat{x}(t)$) is given by,

$$e(t) = x(t) - \hat{x}(t) \tag{5}$$

with $\hat{x}(t)$ obtained from

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x}(t) + Bu(t) + K(y(t) - \hat{y}(t)) \\ \hat{y}(t) &= C\hat{x}(t) \end{aligned} \tag{6}$$

1. Design the optimal controller gains ***(G)*** (3) and the optimal observer gains (*K*) (6) based on LMI strategy for the system (4), which are set to $G=MZ^{-1}$ and $K=P_2^{-1}N$ with matrices *Z*, *M* and *N* satisfying the following LMIs,

$$AZ+ZA^T-BM-(BM)^T<0 \tag{7}$$

$$P_2A+A^TP_2-NC-(NC)^T<0 \tag{8}$$

Where $Z=P_1^{-1}$, $M=GZ$ and $N=P_2K$

2. Write the YALMIP code to solve the obtained LMIs (7) and (8) **(to download and install YALMIP see Appendix: YALMIP)**

3. In Simulink Editor of Matlab, construct and implement the closed-loop system of Fig. 1

Exercise 3

Consider a system represented by the following linear continuous-time state equations:

$$\dot{x} = Ax + Bu, \quad x(0) = x_0,$$

where $x \epsilon\ R^n$ is the state vector, $u \epsilon\ R^m$ is the input vector, A and B are known matrices of appropriate dimensions and given by ,

$$A = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix},\ B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

The objective function is to

$$J = \int_0^\infty (x^T Qx + u^T Ru)dt,$$

where $Q \epsilon\ R^{n \times n}$ is a real symmetric positive semi-definite matrix and $R \epsilon\ R^{p \times p}$ is a real positive definite matrix. The optimal control input which minimizes J is given by

$$u(t) = R^{-1}B^T Px(t) = Kx(t), \quad K = R^{-1}B^T P,$$

where the matrix P is obtained by solving the following Riccati equation:

$$A^T P + PA + PBR^{-1}B^T P + Q < 0,\ P > 0,\ R > 0. \tag{3}$$

Note that the Riccati equation, in contrast to Lyapunov equations, is a nonlinear equation in P, this is because the quadratic term $PBR^{-1}B^T P$ appears in the inequality is known NonLinear Matrix Inequality (NLMI).

1. Using the Schur Complement, represent the inequality (3) as LMIs.
2. Also solving LMI based on YALMIP, to calculate the controller gain K.

Appendix: YALMIP

How to download and install YALMIP?

1. Make sure you are running Matlab R2009a or later
2. Download YALMIP from http://www.control.isy.liu.se/~johanl/YALMIP.zip
3. Create a directory `tbxmanager`
4. Go to that directory in Matlab
5. Run the following:
```
urlwrite('http://www.tbxmanager.com/tbxmanager.m', 'tbxmanager.m');
tbxmanager
savepath
```
6. Edit/create `startup.m` in your Matlab startup folder and put the following line there:
```
tbxmanager restorepath
```
Alternatively, run this command manually every time you start Matlab.
7. Write this in command wind and then enter
```
  tbxmanager install yalmip
tbxmanager install sedumi
```

SDT3

http://www.math.nus.edu.sg/~mattohkc/sdpt3.html

- Copyright: This version of SDPT3 is distributed under the GNU General Public License 2.0. For commercial applications that may be incompatible with this license, please contact the authors to discuss alternatives.
- SDPT3 is currently used as one of the main computational engines in optimization modeling languages such as CVX and YALMIP.
- **Download** SDPT3-4.0.zip
  Please read. Welcome to SDPT3-4.0! The software is built for MATLAB version 7.4 or later releases, it may not work for earlier versions. The software requires a few Mex files for execution. You can generate the Mex files as follows:
  - Firstly, unpack the software:
    unzip SDPT3-4.0.zip;
  - Run Matlab in the directory SDPT3-4.0
  - In Matlab command window, type:
    >> Installmex(1)
  - After that, to see whether you have installed SDPT3 correctly, type:
    >> startup
    >> sqlpdemo
  - By now, SDPT3 is ready for you to use.
- **User's guide** (pdf) (Draft)

https://yalmip.github.io/tutorial/maxdetprogramming/

[1] Z. Gajic and M. Lelic, *Modern Control Systems Engineering*, Prentice Hall International, London, 1996.