Multivariable Control (ME-422) - Exercise session 6 SOLUTIONS

Prof. G. Ferrari Trecate

1. Consider the MIMO DT LTI system

$$x^+ = \begin{bmatrix} 0 & 2 & 3 \\ 0 & -2 & 4 \\ 0 & 0 & 3 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

(a) Is it possible to assign the eigenvalues using a scalar channel? If yes, design a state-feedback controller assigning the eigenvalues in $\frac{1}{2}$, $\frac{1}{3}$, and $\frac{1}{4}$.
Hint: Relevant MATLAB commands are `ctrb` and `place`.

(b) Use the probabilistic approach seen in the lectures for assigning the eigenvalues.

Solution:

(a) Using only the first channel corresponds to setting $B_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$. One has

$$M_r = \texttt{ctrb}(A, B_1) = \begin{bmatrix} 1 & 2 & -4 \\ 1 & -2 & 4 \\ 0 & 0 & 0 \end{bmatrix}$$

which has *rank* $2 < 3$. The system is not reachable from the first channel.

Using only the second control input corresponds to setting $B_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$. One has

$$M_r = \texttt{ctrb}(A, B_2) = \begin{bmatrix} 0 & 3 & 17 \\ 0 & 4 & 4 \\ 1 & 3 & 9 \end{bmatrix}$$

which has *rank* 3. The system is reachable from the second channel. Now, it is possible to set the feedback control law to $u = Kx = K_1 K_2 x$ where we set $K_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ such that the system corresponds to

$$x^+ = (A + \underbrace{BK_1}_{\triangleq B_2} K_2)x = (A + B_2 K_2)x \qquad A \in \mathbb{R}^{3\times 3}, B_2 \in \mathbb{R}^{3\times 1}.$$

Now, it is possible to use Ackermann's formula for the design of K_2. Having already found M_r above, let us find the desired characteristic polynomical of the system as

$$p^D(\lambda) = (\lambda - \frac{1}{2})(\lambda - \frac{1}{3})(\lambda - \frac{1}{4}) = \lambda^3 + \underbrace{\left(-\frac{26}{24}\right)}_{\tilde{a}_2} \lambda^2 + \underbrace{\left(\frac{9}{24}\right)}_{\tilde{a}_1} \lambda + \underbrace{\left(-\frac{1}{24}\right)}_{\tilde{a}_0}. \tag{1}$$

Then, we evaluate this polynomical for the matrix A as

$$p^D(A) = A^3 + \tilde{a}_2 A^2 + \tilde{a}_1 A + \tilde{a}_0 I = \begin{bmatrix} -0.0417 & 13.0833 & 17.7083 \\ 0 & -13.1250 & 25.1667 \\ 0 & 0 & 18.333 \end{bmatrix}.$$

We can now calculate K_2 using Ackermann's formula as

$$K_2 = -\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} M_r^{-1} p^D(A) = \begin{bmatrix} 0.0030 & -1.6376 & 0.0833 \end{bmatrix}.$$

Therefore, the feedback control law is given by

$$Kx = K_1 K_2 x = \begin{bmatrix} 0 & 0 & 0 \\ 0.0030 & -1.6376 & 0.0833 \end{bmatrix}.$$

Checking the eigenvalues of the closed-loop system gives

$$\texttt{eig}(A + BK_1K_2) = \begin{bmatrix} 0.25 \\ 0.33 \\ 0.5 \end{bmatrix}.$$

(b) For the probabilistic approach, we will first select random matrices $K_1 \in \mathbb{R}^{2\times 3}$ and $K_2 \in \mathbb{R}^{2\times 1}$ using MATLAB's `rand` command. One possible choice of K_1 and K_2 matrices is

$$K_1 = \begin{bmatrix} 0.2785 & 0.9575 & 0.1576 \\ 0.5469 & 0.9649 & 0.9706 \end{bmatrix} \qquad K_2 = \begin{bmatrix} 0.9572 \\ 0.4854 \end{bmatrix}$$

for which, the feedback control law is $u = K_1 x + K_2 v, v \in \mathbb{R}$. The new augmented system has the dynamics

$$x^+ = \underbrace{(A + BK_1)}_{A_1} x + \underbrace{BK_2}_{B_1} v.$$

We then calculate the reachability matrix as

$$M_r = \begin{bmatrix} B_1 & A_1 B_1 & A_1^2 B_1 \end{bmatrix} = \begin{bmatrix} 0.9572 & 4.6300 & 15.7495 \\ 0.9572 & 1.2867 & 13.9768 \\ 0.4854 & 3.3742 & 17.1714 \end{bmatrix}.$$

The pair (A_1, B_1) is reachable because $\texttt{rank}(M_r) = 3$. Then, one can use Ackermann's formula to design matrix $K_3 \in \mathbb{R}^{1\times 3}$ such that $v = K_3 x$ as

$$K_3 = -\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} M_r^{-1} p^D(A_1)$$

where $p^D(A_1)$ is the calculation of characteristic equation in (1) with matrix $\lambda = A_1$. We have

$$K_3 = \begin{bmatrix} -0.3913 & 0.5777 & -4.7421 \end{bmatrix}$$

and the feedback control law is defined as $u = (BK_1 + BK_2K_3)x$. The eigenvalues of the closed loop system

$$x^+ = (A + BK_1 + BK_2K_3)x$$

are now placed at $\frac{1}{2}$, $\frac{1}{3}$, and $\frac{1}{4}$.

2. **Inverted Pendulum on Cart**
In this exercise, you will discretize, simulate, and control a cart with an inverted pendulum on top. This exercise will be mainly based on MATLAB and Simulink. A folder containing all the necessary files for simulating the system is provided in Moodle; you will only have to familiarize with the simulation file and implement into it the controllers that you design.

   (a) Download the file *Inverted+pendulum+2018-1+no+solutions.zip* and unzip it. Read the document *Inverted+Pendulum+description.pdf*, which describes the dynamics of the system.

(b) Run the initialization file *pendulum_sys_init.m*.

- Open the file *pendulum_openloop_anim.mdl*. If you have the "Simulink 3D Animation" toolbox installed, you will see the pendulum animation. Otherwise, comment the block named "VR_visualization" to run a simulation. Values of variables will be still visible in the scope blocks.
  Hint: To install, under the "APPS" tab in the MATLAB windows, select "Get More Apps" and search for the toolbox.
- Linearize the system around the obtained equilibrium. To do that, run the Simulink file *pendulum_openloop_anim.mdl*. Thanks to the "Time-Based Linearization" block you will find, in the workspace, the structure "pendulum_openloop_anim_Time_Based_Linearization" with the matrices of the linearized system.
- Run a simulation, add a perturbation $V = 0.0001$ to the input and observe the result. Then, run a second simulation with $V = 0$ in order to store the correct linearized model in the workspace, before you proceed with the rest of the exercise.
- Explore the Simulink blocks. In particular, zooming into the subsystem block, the initial state for a simulation is stored in the "integrator" block.

(c) Using MATLAB, discretize the system with sampling time $T = 0.1s$ and exact discretization.
Hint: Use the MATLAB function `c2d`.

(d) Assume that the state is fully accessible. After checking if the discretized system is reachable, design a controller that assigns its eigenvalues in e^{-2T}, $e^{-2.5T}$, e^{-3T}, and e^{-4T}.

Hint: To compute the reachability matrix, use the MATLAB function `ctrb(A,B)`. To calculate the pole placement gain, you can use the MATLAB function `place(A,B,P)`, where P is a vector containing the values of the desired poles.

The simulation file *pole_placement_pendulum_anim.slx* simulates how the closed-loop system reacts to pulse disturbances (of amplitude 5) on the nominal voltage $V = 0$, for a given choice of feedback gain K. Run this model with the K matrix you calculated to check the performance of the controller against disturbance. Note that the pendulum nonlinear model is present in the loop.

- Increase the pulse amplitude to 20 and repeat the simulation.

(e) Design a new controller assigning the eigenvalues in e^{-10T}, e^{-3T}, $e^{-6.5T}$, and e^{-6T}.
Run a simulation and compare with the previous regulator. Keep increasing the disturbance amplitude. Which controller provides better performance? Can you say intuitively? Why?

Solution: See the MATLAB file `Ex6.m`.