GER: Practical Works on Linear State Space Control

Álan e Sousa, Kevin Guelton
{alan.e-sousa, kevin.guelton}@univ-reims.fr

2022/2023

Contents

1 TP1 – A Library of Systems Modeling

To use the techniques studied we need a system model. The control technique can only be as good as your model, so having a poor model will surely lead to a poor performance in the real system, even if everything checks out on paper. Thus, the objective of this TP is to provide some real, practical modeling problems in MATLAB and Simulink related to the examples studied during this course.

1.1 Horizontal mass-spring-damper system

Consider the mass-spring-damper system depicted in Figure 1, which dynamic equations are:

$$M\ddot{q}(t) = -kq(t) - k_d\dot{q}(t) + F(t) \tag{1}$$

where $M = 0.1\,\mathrm{kg}$ is the weight of the horizontally displaced mass, $k = 10\,\mathrm{N\,m^{-1}}$ is the spring constant (stiffness) and k_d is the damping constant (viscous friction), whose influence on the system behavior we will study.

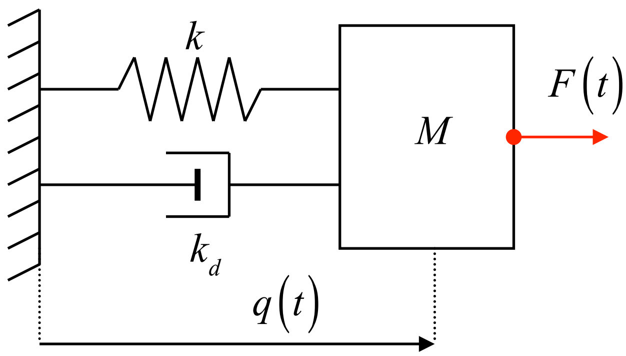

Figure 1: Horizontal mass-spring-damper system.

1.1.1 *Exercises*

1) Give a state-space representation of this system.
2) Create a MATLAB file with the system parameters.
3) Study the open-loop system stability with $k_d = 0$, $k_d = 1$, and $k_d = 20$. For these different values compute the eigenvalues of the obtained systems using the MATLAB function `eig`.
4) Assume that only the position is measured, plot the impulse response of the open-loop system for all above given values of k_d, with an initial condition $x(0) = \begin{bmatrix} \dot{q} \\ q \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, using the function `initial`.
5) For $k_d = 1$, plot the unit step response of the system.

1.2 Simple inverted pendulum

Consider the robot arm with 1 DOF (Degree of Freedom) represented by the simple inverted pendulum depicted in Figure 2.

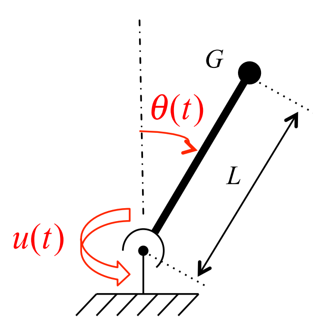

Figure 2: Inverted Pendulum

The system's dynamic equation is

$$mgL\sin(\theta(t)) - kL\dot{\theta}(t) + u(t) = mL^2\ddot{\theta}(t), \tag{2}$$

where $m = 0.4\,\mathrm{kg}$ is the pendulum mass, $L = 0.3\,\mathrm{m}$ is the pendulum length, $k = 0.1$ is the friction coefficient and $g = 9.81\,\mathrm{m\,s^{-2}}$ is the gravitational acceleration.

1.2.1 Exercises

1) In MATLAB, declare the model parameters.
2) In Simulink, implement the non-linear model of the system (tip: create a subsystem).
3) Simulate this model in an open loop for $\theta(0) = 0$ and $\dot{\theta}(0) = 0$, then for $\theta(0) = 0.001$ and $\dot{\theta}(0) = 0$. What do you notice?
4) Using Simulink analysis tools (Analysis/Control Design/Linear Analysis), linearize the model (2) around $\theta = 0$, then $\theta = \pi/2$ and finally around $\theta = \pi$. Compare these results with the ones obtained in the course (Slide 26). Remark: Under Simulink, we must choose the initial conditions of the Integrator's block according to the points around which we wish to obtain the linearization (made by default for $\theta = 0$).
5) Using the function `eig`, analyze the stability of each of the three obtained linear models.
6) Based on the linear model and considering the erected position, with the function `place`, considering a state feedback input ($u(t) = -Fx(t)$), compute the gain matrix for the following desired closed-loop poles $\lambda_1 = -0.1$, $\lambda_2 = -0.2$, $\lambda_1 = -10$, $\lambda_2 = -20$. and $\lambda_1 = -1 + 0.5i$, $\lambda_2 = -1 - 0.5i$. For these three cases, plot the closed-loop system's impulsive response with the initial conditions $x(0) = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{\pi}{2} \\ 0 \end{bmatrix}$ and $x(0) = \begin{bmatrix} \pi \\ 0 \end{bmatrix}$ (Pendulum in the low position). Discuss the results and conclude.

1.3 Academic Examples

1.3.1 Exercises

1) Program three MATLAB scripts to simulate the examples 1-3, presented in Mr. Guelton's support slides 33-35. (use the functions `ss` and `initial`).
2) Program a MATLAB script to simulate the stabilization example presented in slides 41-43 from Mr. Guelton's support slides. (The phase plan should be plot with the functions `meshgrid` and `quiver`).
3) Prepare a MATLAB script to simulate the stabilization example by pole placement presented on slides 48-51 from Mr. Guelton's support slides.
4) Make a MATLAB script to compute the observer gains for the example presented on slides 80-82. Implement the state-space model and the related observer in Simulink, then plot the states and estimated states (like in slide 82) with an input $u(t) = \sin\ (2t)$ applied to the system.

2 TP2 – Direct Current (D.C.) motor position and velocity control

In this last TP, you will implement the position control then the speed control of a DC motor, represented in Figure 3.

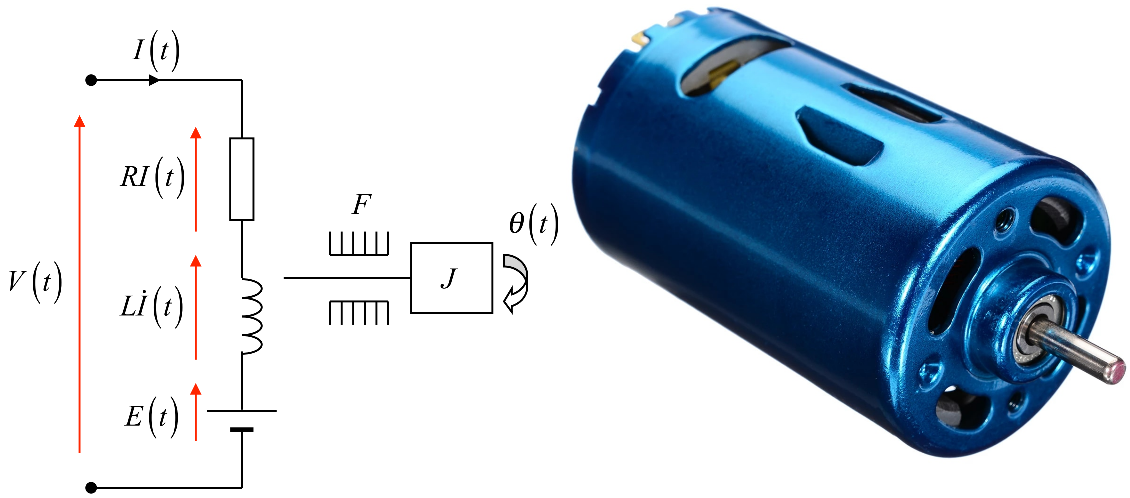

Figure 3: DC motor electrical scheme and illustrative picture.

The dynamical equations of this motor are:

$$\begin{aligned} V(t) &= R_m I(t) + L\dot{I}(t) + K_e\dot{\theta}(t), \\ J\ddot{\theta}(t) &= K_c I(t) - F_r\dot{\theta}(t) - \tau(t), \end{aligned} \tag{3}$$

where $V(t)$ and $I(t)$ are, respectively, the motor input voltage and the current flowing through its winding (inductor). $\theta(t)$ is the angular position of the axis (rotor) and $\tau(t)$ is the resistive torque (supposed piece-wise constant), representing an external load applied to the rotor of the motor (motor reduction, gear box...). The constants are:

$R_m = 0.1\,\Omega$	motor's internal electrical resistance;
$L = 0.5 \times 10^{-3}\,\mathrm{H}$	armature's inductance (motor winding);
$J = 0.01\,\mathrm{kg\,m^2}$	rotor's inertia;
$K_c = 0.1\,\mathrm{N\,m\,A^{-1}}$	torque constant;
$F_r = 0.1\,\mathrm{N\,m\,s\,rad^{-1}}$	viscous friction coefficient;
$K_e = 0.1\,\mathrm{V\,rad^{-1}\,s^{-1}}$	electromotive force connecting the rotational velocity to the counter-electromotive force (CEMF);

2.1 State-space representation

2.1.1 Exercises

1) Give the state-space representation's matrices in the form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + H\tau(t), \\ y(t) &= Cx(t). \end{aligned} \tag{4}$$

   Add the position $\theta(t)$ as a state, as it will be necessary next.
2) Create a MATLAB script with the system's parameters.
3) Calculate the system open-loop eigenvalues. What can you conclude?
4) Assuming that only the position is measurable, is this model controllable and observable?
5) In Simulink, create a sub model block of the motor's model with $y(t) = x(t)$.
6) Make an open-loop simulation of the motor for $u = 1$. What can you observe?

2.2 Rotor position control with pre-compensation gain

In this part we will assume that all the state variables' measurements are available. Let the state feedback control law with the pre-compensation gain be

$$u(t) = -Kx(t) + R\theta_c(t), \tag{5}$$

where $\theta_c(t)$ is the position set-point.

2.2.1 Exercises

1) Using the function `place`, synthesize the gain matrix K for the control law (5) with desired closed-loop eigenvalues as $\lambda_{1,2} = -0.5 \pm \frac{\sqrt{3}}{2}i$ and $\lambda_3 = -2$. Compare the obtained result with the theoretical one (obtained in TD course).
2) Compute the pre-compensation gain R which ensures the set-point tracking in the absence of the resistivity coupling ($\tau(t = 0)$). Compare the obtained results with the obtained in TD class.
3) In Simulink, implement the control law (5) and run a 100 s closed-loop simulation:
   - To track a piece-wise constant signal generated by the source `Repeating Sequence Stair` with the following parameters: `vector of output values` $= \begin{bmatrix} 3 & 1 & 4 & 2 & 1 \end{bmatrix}$ and `sample time` $= 20$.
   - Simulate the load $\tau(t)$ using the `unitary step` source, starting in $t = 0$ s, to $t = 50$ s.
4) Discuss the obtained closed-loop simulation results and, if necessary, modify the eigenvalues of the closed-loop (apply a multiplying factor on them) in order to increase the speed of the transient regime (chose the parameters to obtain a settling time of about 1 s).

2.3 Rotor's position control with integral action

We suppose in this part that all the state variables are available for the controller. Lets consider the state feedback control law with integral action given by:

$$u(t) = -K_P x(t) - K_I \int_0^t (\theta(\tau) - \theta_c(t))d\tau \tag{6}$$

where $\theta_c(t)$ represents the desired position.

2.3.1 Exercises

Repeat Section 2.2's exercises with the control law (6). Use $\lambda_4 = -5$. There's no matrix R this time.

2.4 Position Control with state feedback from estimator and Integral action

In this part, we assume that only the position is available for the controller. Therefore, we propose the implementation of a Luenberger type observer to estimate the state vector. The introduction of this system into the regulation loop gives the following dynamics:

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t)), \\ \hat{y}(t) &= C\hat{x}(t) \end{aligned} \tag{7}$$

Thus, assuming the observer convergence, the following control law can be applied:

$$u(t) = -K_P \hat{x}(t) - K_I \int_0^t (\hat{\theta}(\tau) - \theta_c(t))d\tau \tag{8}$$

2.4.1 Exercises

1) Design the matrix L.
2) Repeat Section 2.3's exercises with the control law (8).

2.5 Speed control with integral action

In this part we suppose that all the states are available for the controller and we assume that the output is the measured velocity $\dot{\theta}(t)$. So, we want to design a control scheme in order to guarantee that the speed $\dot{\theta}(t)$ tracks a desired speed value $\dot{\theta}_c(t)$. In this context we consider the following control law with an integral action:

$$u(t) = -K_P x(t) - K_I \int_0^t (\dot{\theta}(\tau) - \dot{\theta}_c(t))d\tau \tag{9}$$

2.5.1 Exercises

1) Do you think that the state representation (4) with $x(t) = \begin{bmatrix}\theta(t) & \dot{\theta}(t) & I(t)\end{bmatrix}^\top$ is appropriated to make the controller gain synthesis for this tracking problem? (Justify your answer).
2) Considering $x(t) = \begin{bmatrix}\dot{\theta} & I\end{bmatrix}^\top$, repeat Section 2.3's exercises with the control law (9). This time, disconsider the pole λ_5, as it is no longer necessary.

2.6 Speed tracking using state estimation and integral action

Now, we want to perform a speed tracking control of the motor with an integral action. Nevertheless, it turns out that only the **position** of the motor is measured. So the control law is

$$u(t) = -K_P\hat{x}(t) - K_I \int_0^t (\hat{\dot{\theta}}(\tau) - \dot{\theta}_c(t))d\tau \tag{10}$$

2.6.1 Exercises

1) Repeat Section 2.4's exercises with the control law (10).