SY32: Practical Works on Linear State Space Control

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

2021/2022

Contents

1 TP1 – Modeling and Study of State-Space Systems

The goal of this TP is to improve the understanding of the basic conceptual tools for the modeling of physical systems in the state-space framework.

1.1 Case Study 1: Vehicle Suspension

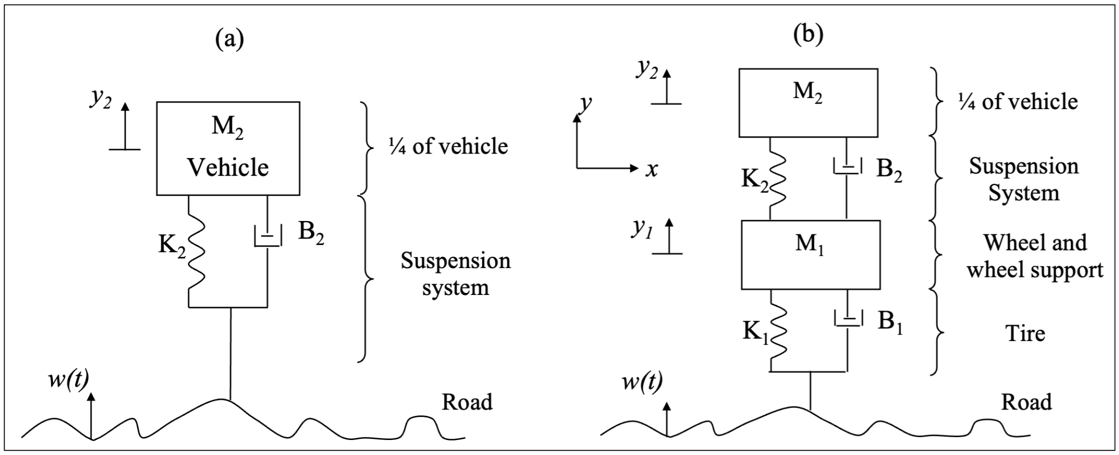

Figure 1: Quarter-car Suspension System

First, in Figure 1 (a), we assume the wheel to be stiffer than the vehicle suspension and its mass to be very small compared to the one of the vehicle. Therefore, the dynamics of the wheel is neglected.

Then, in Figure 1 (b), the mass of the wheel is no longer neglected, and its dynamics are modeled as an additional mass-spring-damper system.

In both cases, the vehicle is supposed to move within the horizontal direction. Thus, the vertical variation of the pavement is considered as an ideal source of speed denoted $w(t)$, and the corresponding output is the vertical component of the vehicle mass velocity denoted $y(t) = v_2(t)$.

1.1.1 *Exercises*

For both systems:

1) Give the movement equations that describe each of the system elements.
2) Choose an appropriated set of states variables.
3) Give the state equations and the output.

In MATLAB:

1) Create a new MATLAB script and declare the parameters of the system as:
   M_1=25 kg, M_2=250 kg, K_1=210 kN m^{-1}, K_2=29.5 kN m^{-1}, B_1=13.1 kN s m^{-1}, B_2=1850 N s m^{-1}.
2) Declare the A, B, C and D matrices for both state-space representations obtained in part 1.
3) Create a Simulink and implement the system representation then simulate its response for the input:
   $w(t) = 10\sin\ (\omega t)$, where $\omega = 2\pi\mathrm{rad\,s}^{-1}$.
4) Observe the evolution of the input, the states, and the system output. Compare the results of both configurations depicted in Figure 1.
5) Simulate the system behavior for both configurations with the following input signals:
   $\omega_1 = 5\sqrt{\frac{K_1}{M_1}}$, $\omega_2 = \sqrt{\frac{K_1}{M_1}}$, $\omega_3 = 0.5\sqrt{\frac{K_1}{M_1}}$, $\omega_4 = \sqrt{\frac{K_2}{M_2}}$, $\omega_5 = 0.1\sqrt{\frac{K_2}{M_2}}$,
6) What can we say about the signals ω_2 and ω_4?
7) For both configurations, make a simulation that physically correspond to the suspension responses when the vehicle passes a 10 cm high curb.

1.2 Case Study 2: Transfer Function and State-Space representation

Consider the following state-space representation:

$$\dot{x} = Ax + Bu, \tag{1}$$
$$y = Cx, \tag{2}$$

with

$$A = \begin{bmatrix} 1 & 2 & 6 \\ 0 & 2 & 3 \\ 1 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}. \tag{3}$$

1.2.1 Exercises

1) Generate the transfer function using the functions `tf` and `ss`.
2) Let W be an invertible matrix given by:

$$W = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 2 & 3 \\ 4 & 0 & 1 \end{bmatrix}, \tag{4}$$

   Given the change of variable $x(t) = Wz(t)$, give the state-space model and the corresponding transfer function.
3) Compare these results (before and after the change of variable) and give some explanations to conclude.

1.3 Case Study 3: Controllability and Observability

Consider the following linear system:

$$\dot{x}(t) = \begin{bmatrix} -\frac{7}{2} & 1 & -2 & 0 \\ \frac{3}{2} & 0 & 2 & 0 \\ \frac{7}{2} & -1 & 2 & 0 \\ -1 & 2 & -3 & -2 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{5}{2} \\ 1 \\ 1 \end{bmatrix} u(t), \tag{5}$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} x(t). \tag{6}$$

1.3.1 Exercises

1) Is this system stable?
2) Calculate the system observability matrix using the `obsv` function.
3) Calculate the rank of the observability matrix using the `rank` function. Is this system fully observable?
4) Consider the fully observable part of the system:

$$\dot{x}(t) = \begin{bmatrix} -\frac{7}{2} & 1 & -2 \\ \frac{3}{2} & 0 & 2 \\ \frac{7}{2} & -1 & 2 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{5}{2} \\ 1 \end{bmatrix} u(t), \tag{7}$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x(t). \tag{8}$$

   Show that the corresponding transfer function of both systems is (and explain why):

$$F(s) = \frac{0.5s + 2}{s^3 + 1.5s^2 + 0.5s} \tag{9}$$

5) Give a representation of this system in the observable companion form. You may use the `canon` function, even though it's broken.
6) Using the state-space system representation in the observable companion form, give the system controllability matrix using the `ctrb` function.
7) Calculate the rank of this matrix. Is this system entirely controllable?

2 TP2 – State Feedback Controller Design

The goal of this TP is to analyze the different characteristics of controllability and observability of a system and to design an adapted state feedback controllers.

2.1 Pole placement and closed-loop dynamic characteristics

Let us consider the system defined by the following transfer function:

$$F_{BO}(s) = \frac{5}{s(s+1)(s+2)} \tag{10}$$

When the controller is only proportional, the equivalent block-diagram scheme is given by (with a unity feedback loop):

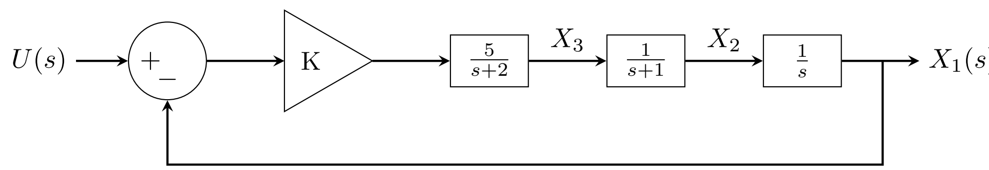

Figure 2: Closed-loop system with a proportional state feedback

2.1.1 Exercises

1) With MATLAB, plot the root locus of the system with various $K > 0$. Choose the roots' location by taking particular values K_b (Value of K for which the place leaves the real axis) and K_{ins} (Value of K which the system is in its instability limit).
2) Record on the same diagram the unit step response of the system for $K = 2K_b, K_b, \frac{K_b}{2}, K_{ins}$ and $1.1K_{ins}$.
3) Which is the influence of K over the system dynamics?
4) Give the state-space equations of the system choosing the state variables x_1, x_2 and x_3 specified on Figure 2 (Model 2).
5) Give the state space equations of the system in the controllable form (Model 3).
6) In the same Simulink file, simulate the previous 3 systems' responses with $K = 2K_b$
7) Analyze these simulation results and conclude.

2.2 Theoretical analysis and state feedback control of a system

In this part, we consider the linear system represented by the following transfer function:

$$G(s) = \frac{5s+1}{s^3 - 4s} \tag{11}$$

2.2.1 Exercises

1) Compute the poles of $G(s)$. What can you say about the system stability?
2) Give a state-space representation for this system.
3) Is it possible to construct a state feedback controller to stabilize this system? Justify your answer.
4) Is this system observable?
5) Propose a state feedback controller for this system to have a closed-loop response corresponding to the following transfer function:

$$G_{reg}(s) = \frac{0.1}{s^3 + 1.1s^2 + 0.2s + 0.1} \tag{12}$$

6) Using the state-space model obtained earlier, construct a Simulink model of the system $G(s)$.
7) Impose to the system a unit-step input and observe the output response. What can you conclude? Justify your answer about the obtained results.
8) In the same Simulink file, add the block scheme $G_{reg}(s)$ to simulate the expected behavior for the system $G_s(s)$ controlled by a state feedback scheme.

9) Observe the response of $G_{reg}(s)$ and comment your result. Does the choice of $G_{reg}(s)$ make sense for you? Justify your answer.
10) Make a MATLAB program to design the state feedback controller gains. Compare your results with the theoretical part.
11) Add to your Simulink model the state feedback control blocks for $G(s)$. Observe the output signals of $G(s)$ with a state feedback and $G_{reg}(s)$ in the same plot. Comment the results.
12) Observe the evolution of the different state variables of $G(s)$. Considering that $G(s)$ represent a physical system, does this control strategy seem realistic?
13) Can you improve this closed-loop response (controller design)? Propose a solution. Justify your choices and the obtained results.

3 TP3 – Feedback Controller from Estimated States

In state-space control we usually use the state of the system to derive the control signal. However, the state is not always fully available, as it may not be possible, or it may be too costly to insert sensors for them. Nonetheless, the control signal ($u(t)$) is always known and at least the system output ($y(t)$) is usually known, so we can use state observers to estimate the state vector ($\hat{x}(t)$) given the input, output and the system's dynamics, using the schematics shown in Figure 3.

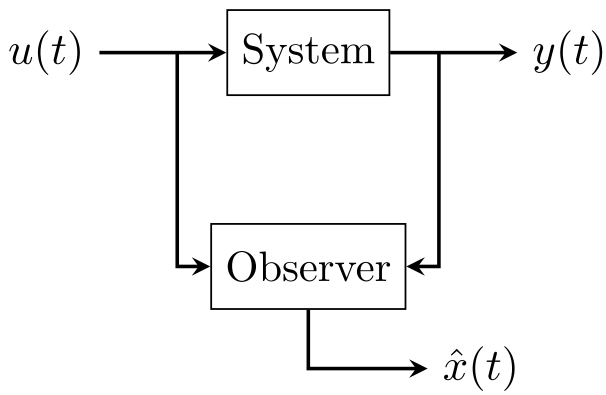

Figure 3: State observer block diagram.

3.1 Luenberger Observer

Based on the linear state-space representation of a system, and assuming that the system is observable, then we can design a so-called Luenberger Observer to estimate the continuous state $x(t)$. The equations

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - C\hat{x}(t)), \tag{13}$$
$$\hat{y}(t) = C\hat{x}(t), \tag{14}$$

define such observers where $\hat{x}(t)$ is the observed state and L is the observer gain vector.

The error dynamics of such system is

$$\dot{\epsilon}(t) = (A - LC)\epsilon(t) \tag{15}$$

and its stability depends on the Hurwitz criterion (if and only if the eigenvalues of $A - LC$ have strictly negative real parts).

3.2 State estimation and feedback control from estimated states

The objective of this part is to design an output feedback control scheme, i.e. using a Luenberger Observer associated with an estimated state feedback controller. For this purpose, we will consider the system defined by the transfer function

$$G(s) = \frac{5s + 1}{s^3 - 4s^2 + 2s + 1}. \tag{16}$$

3.2.1 Exercises

1) From the transfer function of the system, give a state-space representation.
2) From the open-loop systems' state representation, make the stability analysis and verify the controllability and the observability.
3) Make a Simulink simulation of the open-loop system's unit step response.
4) Make a MATLAB script to synthesize the observer gain (L) such that the poles of the estimator error dynamics are all located at -0.5.
5) In the same MATLAB script, design the controller gain (K) by choosing appropriate poles (with different locations) for the closed loop system matrix ($A - BK$). Remark: if at least two desired poles have the same location, can we use the place function?
6) Now, in the previous Simulink file, implement the designed Luenberger Observer and estimated state feedback controller.

7) Observe and analyze the system response's characteristics:
   - first with $x_0 \neq 0$, $\hat{x}_0 = 0$ and $r = 0$;
   - then with $x_0 \neq 0$, $\hat{x}_0 = 0$ and $r = 1$;

   For both the above given cases, show on plots the system's responses with the Peak Time, Settling Time, Overshoot and plot the estimation error. Conclude on the effectiveness of the proposed output feedback control scheme. Does the result match the expectations? Make some adjustments on the observer and controller by tuning their gains to obtain a closed-loop settling time of about 1 s.
8) Based on the simulations' results, explain your choices to achieve the performance target provided at the end of the previous question.
9) Repeat the steps 1 through 8, for the system described by the following transfer function:

$$F(s) = \frac{s+1}{s^2+s+1} \tag{17}$$

4 TP4 – 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.

4.1 Horizontal mass-spring-damper system

Consider the mass-spring-damper system depicted in Figure 4, which dynamic equations are:

$$M\ddot{q}(t) = -kq(t) - k_d\dot{q}(t) + F(t) \tag{18}$$

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.

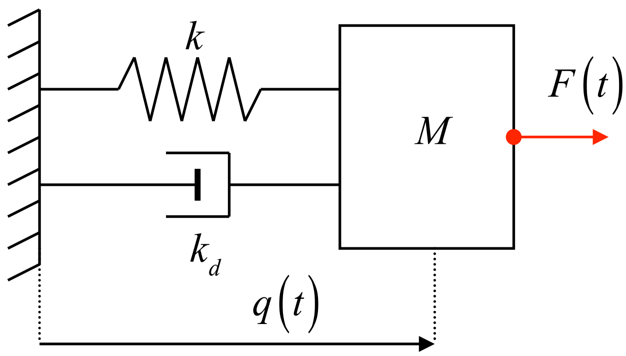

Figure 4: Horizontal mass-spring-damper system.

4.1.1 *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.

4.2 Simple inverted pendulum

Consider the robot arm with 1 DOF (Degree of Freedom) represented by the simple inverted pendulum depicted in Figure 5.

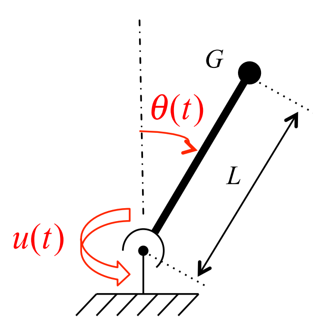

Figure 5: Inverted Pendulum

The system's dynamic equation is

$$mgL\sin(\theta(t)) - kL\dot{\theta}(t) + u(t) = mL^2\ddot{\theta}(t), \tag{19}$$

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.

4.2.1 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 (19) 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 $x(0) = \begin{bmatrix}\theta \\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\frac{\pi}{10} \\ 0\end{bmatrix}$ and $x(0) = \begin{bmatrix}\pi \\ 0\end{bmatrix}$ (Pendulum in the low position). Discuss the results and conclude.

4.3 Academic Examples

4.3.1 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.

5 TP5 – Inverted Pendulum: Simulation, State Estimation and Control

In this TP the goal is to design and implement an observer and a controller for a simulated system which represents the "inverted pendulum mounted on a cart" depicted in Figure 6.

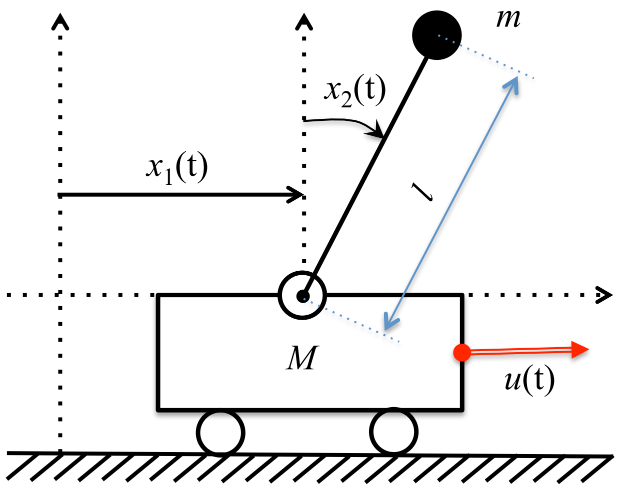

Figure 6: Horizontal mass-spring-damper system.

The dynamic equations of the systems are

$$\begin{aligned} k_1\dot{x}_1(t) + (M+m)\ddot{x}_1(t) + ml\ddot{x}_2(t)\cos(x_2(t)) - ml\dot{x}_2^2(t)\sin(x_2(t)) &= u(t) \\ \frac{k_2}{m}\dot{x}_2(t) + l\ddot{x}_2(t) - g\sin(x_2(t)) + \ddot{x}_1(t)\cos(x_2(t)) &= 0 \end{aligned} \tag{20}$$

where $M = 2\,\text{kg}$ represent the cart mass, $m = 0.1\,\text{kg}$ the pendulum mass (we will neglect the pendulum inertia), $l = 0.5\,\text{m}$ the pendulum length, $x_1(t)$ and $x_2(t)$ represent, respectively, the horizontal position of the cart and the angular position of the inverted pendulum relative to the vertical axis, $u(t)$ is the horizontal force applied over the cart (horizontally motorized) and k_1 and k_2 are the friction coefficient of the horizontal translation and rotational DOF respectively.

5.1 Exercises

1) The system (20) can be rewritten as the following non-linear state-space:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_3 \\ x_4 \\ \Pi^{-1}(x_2) \cdot f(x,u) \end{bmatrix}. \tag{21}$$

   Find the matrix $\Pi(x_2)$, function $f(x,u)$ and state vector x that makes (21) equivalent to (20). Hint: use the fact that a system of equations can be rewritten as matricial operations to isolate both $\ddot{x}_1$ and $\ddot{x}_2$ at the same time.
2) Determine the set of the equilibrium points of this system. What can we conclude about the equilibrium of this system?
3) The objective is to stabilize the pendulum around the erect position. Therefore, give a linearized model of (20) around the equilibrium point $x_e = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top$.

5.2 Non-linear system simulation

5.2.1 *Exercises*

1) In a MATLAB script, declare the system (20) parameters with $k_1 = k_2 = 0.1$.
2) In Simulink, implement in a sub model block the non-linear model (20). Tip: this system is a bit complex, so take a look at the MATLAB function `matlabFunctionBlock`.

3) With the initial conditions $x(0) = \begin{bmatrix} 0 & 0.01 & 0 & 0 \end{bmatrix}^\top$, make simulations of the non-linear system with the following parameters $(k_1 = 0, k_2 = 0)$, $(k_1 = 0.1, k_2 = 0)$, $(k_1 = 0, k_2 = 0.1)$ and $(k_1 = 0.1, k_2 = 0.1)$. What do you observe?
4) Then, consider $(k_1 = 0.1, k_2 = 0.1)$. With Simulink tools for linear system analyze (Analysis/Control Design/Linear Analysis) linearize the system around $x_e = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top$. Store the linearization results in the A, B, C and D matrices. Compare the results with the other obtained in the Item 3 of Section 5.1.
5) What are the eigenvalues of the linearized model? What can you conclude about the given equilibrium point?
6) Using the function `ctrb`, study the controllability of the obtained linear model.
7) Consider the state feedback control law $u = -Fx(t)$. Synthesize the gain matrix for the control law so that the closed-loop poles of the linearized system are $\{-2, -2.5, -9, -10\}$.
8) In Simulink, feedback the obtained control law to the nonlinear system then make a simulation of the closed-loop plant, considering $x(0) = \begin{bmatrix} 0 & \frac{\pi}{12} & 0 & 0 \end{bmatrix}^\top$. What can you observe?
9) Simulate the system with successively increasing values of $x_2(0) \in [0, \pi]$ to estimate the validity domain of the control law synthesized around the point $x_e = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top$. Consider $\bar{x}_2$ the maximal value of $x_2(0)$ for which the closed-loop system remains stable.
10) Visualize the response of each state variable for $x_2(0) = \bar{x}_2$. If the system is stable in simulation, what can you conclude regarding to the practical application of this approach? What do you propose?

5.3 State estimator design and output feedback controller

5.3.1 Exercises

1) Now, we consider that only the positions $x_1(t)$ and $x_2(t)$ are measured.
2) Considering the linear model obtained around $x_e = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top$, is it observable?
3) Consider a Luenberger observer given by

$$\hat{x} = Ax + Bu + K(y - \hat{y}) \tag{22}$$
$$\hat{y} = C\hat{x} \tag{23}$$

4) Regarding the output feedback control approach, which values do you choose for the poles of the estimation dynamics? Synthesize the gain matrices for the observer with these poles.
5) Implement the state estimator in your Simulink file and simulate the whole non-linear system (system with observer) in open-loop for $x(0) = \begin{bmatrix} 0 & \frac{\pi}{12} & 0 & 0 \end{bmatrix}^\top$. Does the observer converge?
6) Now, feedback the controller with the estimated state to the nonlinear system, i.e. by considering the control law $u = -F\hat{x}(t)$, with the controller gain matrix F synthesized in Item 7 of Section 5.2.1
7) Simulate the closed-loop system with the initial condition $x(0) = \begin{bmatrix} 0 & \bar{x}_2 & 0 & 0 \end{bmatrix}^\top$. What do you notice? If required, proceed with a new estimation of $\bar{x}_2$ and conclude.
8) Apply a disturbance signal on the output $y_2(t)$ having the shape of a pulse at $t = 5\,\mathrm{s}$, with length $0.1\,\mathrm{s}$ and amplitude $0.1\,\mathrm{rad}$. What can you notice?

6 TP6 – 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 7.

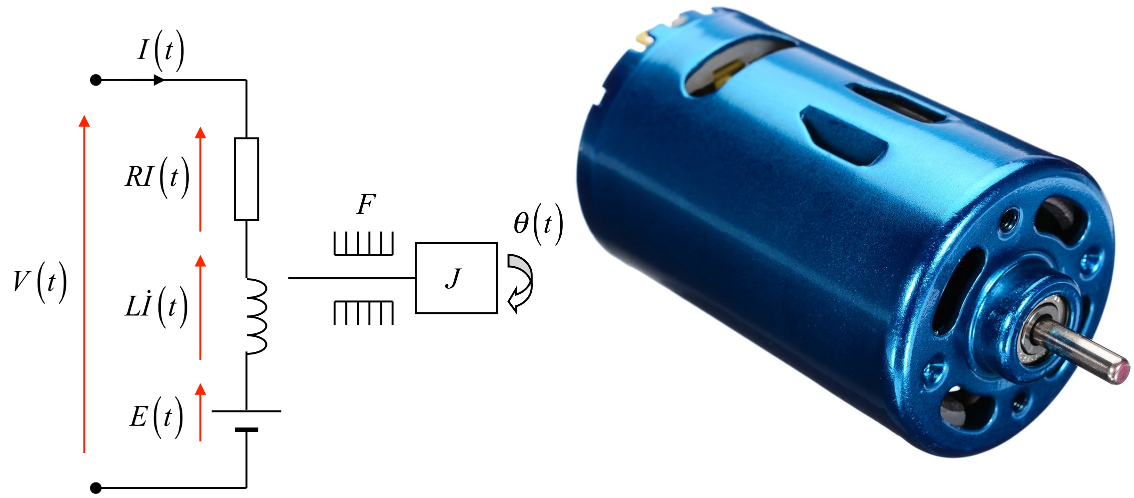

Figure 7: DC motor electrical scheme and illustrative picture.

The dynamical equations of this motor are:

$$\begin{aligned} V(t) &= R_m I(t) + L\dot{I}(t) + K_e\dot{\theta}(t), \\ J\ddot{\theta}(t) &= K_c I(t) - F_r\dot{\theta}(t) - \tau(t), \end{aligned} \tag{24}$$

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);

6.1 State-space representation

6.1.1 Exercises

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

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + H\tau(t), \\ y(t) &= Cx(t). \end{aligned} \tag{25}$$

   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?

6.2 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

$$u(t) = -Kx(t) + R\theta_c(t), \tag{26}$$

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

6.2.1 Exercises

1) Using the function `place`, synthesize the gain matrix K for the control law (26) with desired closed-loop eigenvalues as $\lambda_{1,2} = -0.5 \pm \sqrt{\frac{3}{2}}$ 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 (26) 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).

6.3 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:

$$u(t) = -K_P x(t) - K_I \int_0^t (\theta(\tau) - \theta_c(t)) d\tau \tag{27}$$

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

6.3.1 Exercises

Repeat Section 6.2's exercises with the control law (27). Use $\lambda_5 = -5$. There's no matrix R this time.

6.4 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:

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t)), \\ \hat{y}(t) &= C\hat{x}(t) \end{aligned} \tag{28}$$

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

$$u(t) = -K_P \hat{x}(t) - K_I \int_0^t (\hat{\theta}(\tau) - \theta_c(t)) d\tau \tag{29}$$

6.4.1 Exercises

1) Design the matrix L.
2) Repeat Section 6.3's exercises with the control law (29).

6.5 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:

$$u(t) = -K_P x(t) - K_I \int_0^t (\dot{\theta}(\tau) - \dot{\theta}_c(t)) d\tau \tag{30}$$

6.5.1 Exercises

1) Do you think that the state representation (25) 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 6.3's exercises with the control law (30). This time, disconsider the pole λ_5, as it is no longer necessary.

6.6 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

$$u(t) = -K_P \hat{x}(t) - K_I \int_0^t (\hat{\dot{\theta}}(\tau) - \dot{\theta}_c(t))d\tau \tag{31}$$

6.6.1 Exercises

1) Repeat Section 6.4's exercises with the control law (31).