AutoMOOC
https://www.ensta-bretagne.fr/automooc/

Luc Jaulin

January 8, 2025

Chapter 1

Introduction

Biological, economic, and other mechanical systems surrounding us can often be described by a differential equation such as:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \end{cases}$$

under the hypothesis that the time t in which the system evolves is continuous [1]. The vector $\mathbf{u}(t)$ is the *input* (or *control*) of the system. Its value may be chosen arbitrarily for all t. The vector $\mathbf{y}(t)$ is the *output* of the system and can be measured with a certain degree of accuracy. The vector $\mathbf{x}(t)$ is called the *state* of the system. It represents the memory of the system, in other words the information needed by the system in order to predict its own future, for a known input $\mathbf{u}(t)$. The first of the two equations is called the *evolution equation*. It is a differential equation which enables us to know where the state $\mathbf{x}(t)$ is headed knowing its value at the present moment t and the control $\mathbf{u}(t)$ which we are currently exerting. The second equation is called the *observation equation*. It allows us to calculate the output vector $\mathbf{y}(t)$, knowing the state and the control at time t. Note however that, unlike the evolution equation, this equation is not a differential equation since it does not involve the derivatives of the signals. The two equations above form the *state representation* of the system. It is sometimes useful to consider a discrete time k, with $k \in \mathbb{Z}$, where $\mathbb{Z}$ is the set of integers. If, for instance, the universe being considered is a computer, it is possible to consider that the time k is discrete and synchronized to the clock of the microprocessor. Discrete-time systems often obey a recurrence equation such as:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \\ \quad\;\; \mathbf{y}(k) = \mathbf{g}(\mathbf{x}(k), \mathbf{u}(k)) \end{cases}$$

The first objective of this book is to understand the concept of state representation through numerous exercises. For this, we will consider, in Chapter 2, a large number of varied exercises and show how to reach a state representation. We will then show, in Chapter 3, how to simulate a given system on a computer using its state representation.

The second objective of this book is to propose methods to *control* the systems described by state equations. In other words, we will attempt to build *automatic* machines (in which humans are practically not involved, except to give orders, or *setpoints*), called *controllers* capable of *domesticating* (changing the behavior in a desired direction) the systems being considered. For this, the controller will have to compute the inputs $\mathbf{u}(t)$ to be applied to the system from the (more or less noisy) knowledge of the outputs $\mathbf{y}(t)$ and from the setpoints $\mathbf{w}(t)$ (see Figure 1).

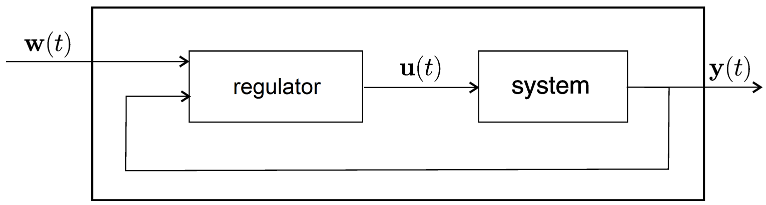


Figure 1.1: Closed loop concept illustrating the control of a system

From the point of view of the user, the system, referred to as a *closed-loop system*, with input $\mathbf{w}(t)$ and output $\mathbf{y}(t)$ will have a suitable behavior. We will say that we have *controlled* the system. With this objective of control, we will, in a first phase, only look at linear systems, in other words when the functions $\mathbf{f}$ and $\mathbf{g}$ are assumed linear. Thus, in the continuous-time case, the state equations of the system are written as:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{cases}$$

and in the discrete-time case, they become:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \end{cases}$$

The matrices $\mathbf{A}$,$\mathbf{B}$,$\mathbf{C}$,$\mathbf{D}$ are called *evolution*, *control*, *observation* and *direct matrices.* A detailed analysis of these systems will be performed in Chapter 4. We will then explain, in Chapter 5, how to stabilize these systems. Finally, we will show in Chapter 6 that around certain points, called *operating points*, non-linear systems behave like linear systems. It will then be possible to stabilize them using the same methods as those developed for the linear case.

Exercises

EXERCISE 1.1.– *Underwater robot*

See the correction video at https://youtu.be/9GCxgVcWB3k

The underwater robot *Saucisse* of the ENSTA Bretagne [2], whose photo is given in Figure 1.2, is a control system. It includes a computer, three propellers, a camera, a compass and a sonar. What does the input vector **u**, the output vector **y**, the state vector **x** and the setpoint **w** correspond to in this context?

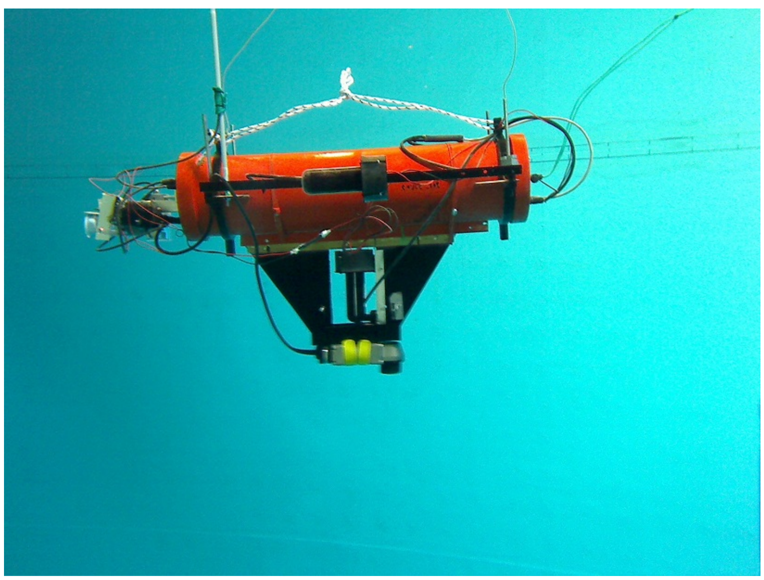
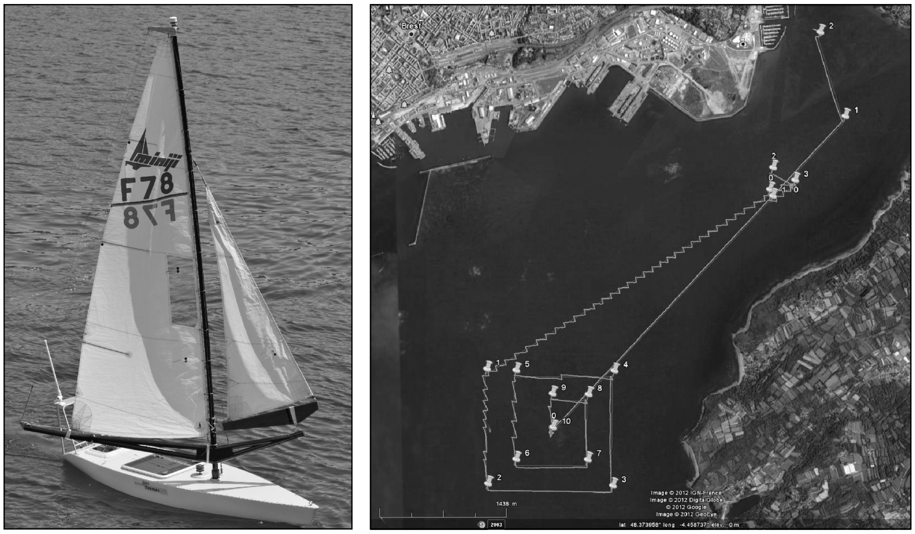


Figure 1.2: Controlled underwater robot

EXERCISE 1.2.– *Sailing robot*

See the correction video at https://youtu.be/JWUSDrIE7JY

The sailing robot VAIMOS (IFREMER and ENSTA Bretagne) in Figure 1.3 is also a control system [3, 4]. It is capable of following paths by itself, such as the one drawn on Figure 1.3. It has a rudder and a sail adjustable using a sheet. It also has an anemometer on top of the mast, a compass and a GPS. Describe what the input vector **u**, the output vector **y**, the state vector **x** and the setpoint **w** may correspond to.



a) b)

Figure 1.3: Sailing robot VAIMOS a) and a path followed by VAIMOS b). The zig-zags in the path are due to VAIMOS having to tack in order to sail against the wind

---

EXERCISE 1.3.– *Chaos*

See the correction video at https://youtu.be/JBIOOykHK9E

We consider the discrete-time state space model described by

$$x(k+1) = 4x(k) \cdot (1 - x(k)).$$

This system is an archetypal example of how complex, chaotic behavior can arise from an apparently simple non-linear dynamical equations. In this system, $x(k) \in [0, 1]$ is the population size (in million, for instance) and k is the year. The model captures two effects:

- the *growth* where the population increases exponentially when the population size is small.
- the *starvation* where the growth rate will decrease when the theoretical carrying capacity of the environment is less than the current population.

1) Give the equilibrium points of the system and specify which of them is stable.

2) How many cycles with a length 3 (i.e., $x(k+3) = x(k)$). A graphic resolution could be used for this purpose.

3) Provide a simulation of the system and illustrate the chaotic behavior of the system.

EXERCISE 1.4.– *Hénon map*

See the correction video at https://youtu.be/Tf_IHzvBnyk

Given a discrete-time state space system of the form $\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k))$. A set $\mathbb{I}$ is said to be invariant $\mathbf{f}(\mathbb{I}) = \mathbb{I}$.

1) Prove that if $\mathbb{I}_1$ and $\mathbb{I}_2$ are both invariant then their union is invariant.

2) Given a set $\mathbb{A}$, show that there exists a largest invariant set which is included in $\mathbb{A}$, denoted by $\mathrm{Inv}(\mathbb{A})$.

3) Show that the sequence of sets defined by

$$\begin{cases} \mathbb{X}(k+1) = \mathbf{f}(\mathbb{X}(k)) \cap \mathbb{A} \\ \mathbb{X}(0) = \mathbb{A} \end{cases}$$

converges toward $\mathrm{Inv}(\mathbb{A})$.

4) Assume that

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} x_2 + 1 - a x_1^2 \\ b x_1 \end{pmatrix}$$

where $a = 1.4$ and $b = 0.3$. Make a program which draws an approximation the largest invariant set of the system which is included in the box $[-2, 2] \times [-2, 2]$.

Chapter 2

Modeling

We will call *modeling* the step which consists of finding a more or less accurate state representation of the system we are looking at. In general, constant parameters appear in the state equations (such as the mass or the inertial moment of a body, the coefficient of viscous friction, the capacitance of a capacitor, etc.). In these cases, an identification step may prove to be necessary. In this book, we will assume that all the parameters are known, otherwise we invite the reader to consult Eric Walter's book [5] for a broad range of identification methods. Of course, no systematic methodology exists that can be used to model a system. The goal of this chapter and of the following exercises is to present, using several varied examples, how to obtain a state representation.

2.1 Modeling method

To model a system here means to represent the evolution of the system under the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \tag{2.1}$$

if the time t is continuous, or

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \tag{2.2}$$

if the time k is discrete. The first step of the modeling is to find what is the memory of the system. When the system is a physical mechanism, this amounts to finding where the energy is stored. The state can also be understood as all quantities we need to know at the current time on the system to be able to predict the future, from the single knowledge of the input vector $\mathbf{u}(t)$ or $\mathbf{u}(k)$. For some applications, it is also needed to find the observation equation which is often easier.

The state variables can be

- the degrees of freedom (position, angles) of a mechanical system,
- the speeds, or equivalently, the derivative of the degrees of freedom;
- the quantity of water inside tank;

- the electric current when it goes through an inductance;
- the charge of a capacitor;
- the temperature of a body.

Then, we apply basic physic rules which tell us how these state variables will change with time. These rules are often given into a simple first order differential equation. Examples are

- The kinematic relations $\dot{x} = v$ or $\dot{v} = a$, which tells us that the speed is the derivative of the position with respect to t or that the acceleration is the derivative of the speed.
- The Newton laws, such as $\dot{v} = \frac{f}{m}$, where f is the force, m is the mass and v is the speed;
- Electric rules such as the Faraday's law of induction $\frac{d}{dt}i = \frac{u}{L}$ which states that any change in flux through a circuit induces an electromotive force or the relation $\frac{dv}{dt} = \frac{i}{C}$ which states that the current $i(t)$ through a capacitor is proportional to the derivative of the voltage $v(t)$.
- The Fick's laws which postulates a time variation of a quantity (such as the temperature) is related to the spacial variations.

Combining all these equations, we have to build a single state equation of the form (2.1) or (2.2).

2.2 Linear systems

In the continuous-time case, linear systems can be described by the following state equations:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{cases}$$

Linear systems are rather rare in nature. However, they are relatively easy to manipulate using linear algebra techniques and often approximate in an acceptable way the non-linear systems around their operating point. Moreover, when we combine linear systems (for instance in line, in parallel, or a loop), we still obtain a linear system. They will we studied in details in Chapter 4.

2.3 Mechanical systems

In this section, we will focus our attention on mechanical systems which are necessary when modeling robots. The fundamental principle of dynamics allows us to easily find the state equations of mechanical systems (such as robots). The resulting calculations are relatively complicated for complex systems and the use of computer algebra systems may prove to be useful. In order to obtain the state equations of a mechanical system composed of several subsystems $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_m$, assumed to be rigid, we follow three steps.

1. *Obtaining the differential equations.* For each subsystem $\mathcal{S}_k$, with mass m and inertial matrix $\mathbf{J}$, the following relations must be applied:

$$\begin{array}{ll} \sum_i \mathbf{f}_i & = m\mathbf{a} \\ \sum_i \mathcal{M}_{\mathbf{f}_i} & = \mathbf{J}\dot{\omega} \end{array}$$

   where the $\mathbf{f}_i$ are the forces acting on the subsystem $\mathcal{S}_k$, $\mathcal{M}_{\mathbf{f}_i}$ represents the torque created by the force $\mathbf{f}_i$ on $\mathcal{S}_k$, with respect to its center. The vector $\mathbf{a}$ represents the tangential acceleration of $\mathcal{S}_k$ and the vector $\dot{\omega}$ represents the angular acceleration of $\mathcal{S}_k$. After decomposing these $2m$ vectorial equations according to their components, we obtain $6m$ scalar differential equations such that some of them might be degenerate.

2. *Removing the components of the internal forces.* In differential equations there are so-called *bonding* forces, which are internal to the global mechanical system, even though they are external to each subsystem composing it. They represent the action of a subsystem $\mathcal{S}_k$ on another subsystem $\mathcal{S}_\ell$. Following the action-reaction principle, the existence of such a force, denoted by $\mathbf{f}^{k,\ell}$, implies the existence of another force $\mathbf{f}^{\ell,k}$, representing the action of $\mathcal{S}_\ell$ on $\mathcal{S}_k$, such that $\mathbf{f}^{\ell,k} = -\mathbf{f}^{k,\ell}$. Through a formal manipulation of the differential equations and by taking into account the equations due to the action-reaction principle, it is possible to remove the internal forces. The resulting number of differential equations has to be reduced to the number n of degrees of freedom $q_1, \dots, q_n$ of the system.

3. *Obtaining the state equations.* We then have to isolate the second derivative $\ddot{q}_1, \dots, \ddot{q}_n$ from the set of n differential equations in such a way to obtain a vectorial relation such as:

$$\ddot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u})$$

   where $\mathbf{u}$ is the vector of external forces which are not derived from a potential (in other words, those which we apply to the system). The state equations are then written as:

$$\frac{d}{dt}\begin{pmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{pmatrix} = \begin{pmatrix} \dot{\mathbf{q}} \\ \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}) \end{pmatrix}.$$

2.4 Servomotors

A mechanical system is controlled by forces or torques and obeys a dynamic model which depends on many poorly known coefficients. This same mechanical system represented by a kinematic model is controlled by positions, speeds or accelerations. The kinematic model depends on well-known geometric coefficients and is a lot easier to put into equations. In practice, we move from a dynamic model to its kinematic equivalent by adding servomotors. In summary, a servomotor is a direct current motor with an electrical control circuit and a sensor (of the position, speed or acceleration). The control circuit computes the voltage u to give to the motor in order for the value measured by the sensor corresponds to the setpoint w. In practice, the signal w is generally given in the form of a square wave called PWM (*Pulse-Width Modulation*). There are three types of servomotors:

- the *position servo*. The sensor measures the position (or the angle) x of the motor and the control rule is expressed as $u = k(x - w)$. If k is large, we may conclude that $x \simeq w$;
- the *speed servo*. The sensor measures the speed (or the angular speed) $\dot{x}$ of the motor and the control rule is expressed as $u = k(\dot{x} - w)$. If k is large, we have $\dot{x} \simeq w$;
- the *acceleration servo*. The sensor measures the acceleration (tangential or angular) $\ddot{x}$ of the motor and the control rule is expressed as $u = k(\ddot{x} - w)$. If k is large, we have $\ddot{x} \simeq w$.

2.5 Non holonomic systems

A mechanical system whose dynamics can be described by the relation $\ddot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u})$ will be referred to as *holonomic*. For a holonomic system, $\mathbf{q}$ and $\dot{\mathbf{q}}$ are thus independent. If there is a so-called *non-holonomic* constraint which links the two of them (of the form $h(\mathbf{q}, \dot{\mathbf{q}}) = 0$), the system will be referred to as *non-holonomic*. Such systems may be found for instance in mobile robots with wheels [6]. Readers interested in more details on the modeling of mechanical systems may consult [7].

Non holonomic system are ubiquitous as soon as we consider vehicles in robotics. To understand how such system occur, consider the robot tank in Figure 2.1. It is composed of two parallel motorized crawlers (or wheels) whose accelerations (which form the inputs u_1 and u_2 of the system) are controlled by two independent motors. In the case where wheels are considered, the stability of the system is ensured by one or two idlers, not represented on the figure. The degrees of freedom of the robot are the x, y coordinates of its center and its heading θ.

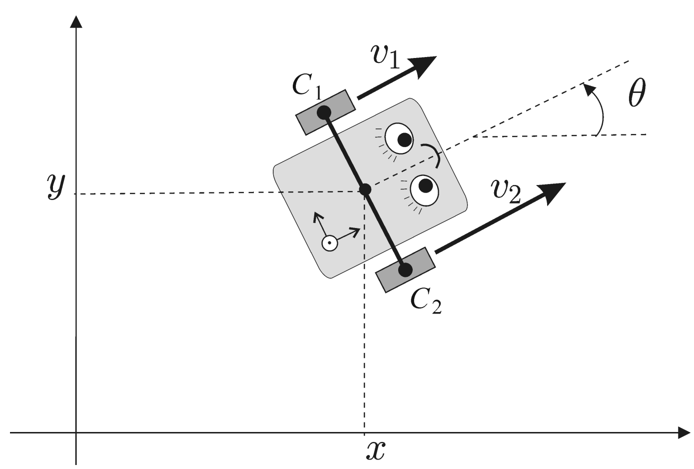


Figure 2.1: Robot tank viewed from above

The state vector cannot be chosen to be equal to $(x, y, \theta, \dot{x}, \dot{y}, \dot{\theta})^{\mathrm{T}}$, which would seem natural with respect to Lagrangian theory. Indeed, some states such as

$$\left(x = 0, y = 0, \theta = 0, \dot{x} = 1, \dot{y} = 1, \dot{\theta} = 0\right)$$

has no meaning since the tank is not allowed to skid. This phenomenon is due to the existence of wheels which creates constraints between the natural state variables. Here, we necessarily have the so-called *non-holonomic* constraint:

$$\dot{y} = \dot{x} \tan \theta$$

Mechanical systems for which there are such constraints on the equality of natural state variables (by natural state variables we mean the vector $(\mathbf{q}, \dot{\mathbf{q}})$ where $\mathbf{q}$ is the vector of the degrees of liberty of our system) are said to be *non-holonomic*. When such a situation arises, it is useful to employ these constraints in order to reduce the number of state variables and this, until no more constraints are left between the state variables.

Denote by v_1 and v_2 the center speeds of each of the motorized wheels. Let us choose as state vector the vector $\mathbf{x} = (x, y, \theta, v_1, v_2)^{\mathrm{T}}$. This choice of state variables is easily understood in the sense that these variables allow us to draw the tank (x, y, θ) and the knowledge of v_1,v_2 allows us to calculate the variables $\dot{x}, \dot{y}, \dot{\theta}$. Moreover, every arbitrary choice of vector (x, y, θ, v_1, v_2) corresponds to a physically possible situation. The state equations of the system are:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v}_1 \\ \dot{v}_2 \end{pmatrix} = \begin{pmatrix} \frac{v_1+v_2}{2} \cos \theta \\ \frac{v_1+v_2}{2} \sin \theta \\ \frac{v_2-v_1}{\ell} \\ Ru_1 \\ Ru_2 \end{pmatrix}$$

where ℓ is the distance between the two wheels. The third relation on $\dot{\theta}$ is obtained by the speed composition rule (Varignon's formula). Indeed, we have:

$$\mathbf{v}_2 = \mathbf{v}_1 + \overrightarrow{C_2C_1} \wedge \overrightarrow{\omega}$$

where $\overrightarrow{\omega}$ is the instantaneous rotation vector of the tank and $\mathbf{v}_1$ and $\mathbf{v}_2$ are the speed vectors of the centers of the wheels. Let us note that this relation is a vector relation which depends on the observer but which is independent of the frame. Let us express this in the frame of the tank, represented on the figure. We must be careful not to confuse the observer fixed on the ground with the frame in which the relation is expressed. This equation is written as:

$$\underbrace{\begin{pmatrix} v_2 \\ 0 \\ 0 \end{pmatrix}}_{\mathbf{v}_2} = \underbrace{\begin{pmatrix} v_1 \\ 0 \\ 0 \end{pmatrix}}_{\mathbf{v}_1} + \underbrace{\begin{pmatrix} 0 \\ \ell \\ 0 \end{pmatrix}}_{\overrightarrow{C_2C_1}} \wedge \underbrace{\begin{pmatrix} 0 \\ 0 \\ \dot{\theta} \end{pmatrix}}_{\overrightarrow{\omega}}$$

We thus obtain $v_2 = v_1 + \ell\dot{\theta}$ or $\dot{\theta} = \frac{v_2-v_1}{\ell}$.

Exercises

EXERCISE 2.1.– *First and second order system*

See the correction video at https://youtu.be/LPnkZJFskjY

We consider an integrator and a second-order system described by

$$\begin{array}{ll} (i) & \dot{y} = u \\ (ii) & \ddot{y} + a_1\dot{y} + a_0 y = bu \end{array}$$

where, u is the input y the output

Find a state representation in matrix form and give the characteristic polynomial for the two systems.

EXERCISE 2.2.– *Mass-spring system*

See the correction video at https://youtu.be/dVdKwrcFpOY

Let us consider a system with input u and output q_1 as shown in Figure 2.2 (u is the force applied to the second carriage, q_i is the deviation of the i^{th} carriage with respect to its equilibrium point, k_i is the stiffness of the i^{th} spring, α is the coefficient of viscous friction).

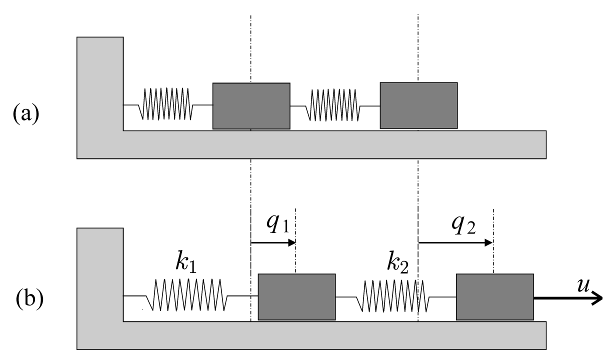


Figure 2.2: a) Mass-spring system at rest, b) system in any state

Let us take the state vector:

$$\mathbf{x} = (q_1, q_2, \dot{q}_1, \dot{q}_2)^{\mathrm{T}}$$

1) Find the state equations of the system.

2) Is this system linear?

EXERCISE 2.3.– *Model a pendulum*

See the correction video at https://youtu.be/0rkma1Jin90

Let us consider the pendulum in Figure 2.3. The input of this system is the torque u exerted on the pendulum around its axis. The output is $y(t)$, the algebraic distance between the mass m and the vertical axis:

1) Determine the state equations of this system using the Newton law.

2) Express the mechanical energy E_m in function of the state of the system. Show that the latter remains constant when the torque u is nil.

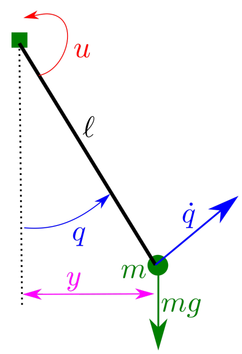


Figure 2.3: Simple pendulum with state vector $\mathbf{x} = (q, \dot{q})$

EXERCISE 2.4.– *Hamilton's method*

See the correction video at https://youtu.be/0LAGI-63-Nw

Hamilton's method allows us to obtain the state equations of a conservative mechanical system (in other words whose energy is conserved) only from the expression of a single function: its energy. For this, we define the *Hamiltonian* as the mechanical energy of the system, in other words the sum of the potential energy and the kinetic energy. The Hamiltonian can be expressed as a function $H(\mathbf{q}, \mathbf{p})$ of the degrees of freedom $\mathbf{q}$ and of the associated momentum $\mathbf{p}$. The Hamilton equations are written as:

$$\begin{cases} \dot{\mathbf{q}} = \dfrac{\partial H(\mathbf{q}, \mathbf{p})}{\partial \mathbf{p}} \\ \dot{\mathbf{p}} = -\dfrac{\partial H(\mathbf{q}, \mathbf{p})}{\partial \mathbf{q}} \end{cases}$$

1) Let us consider the simple pendulum shown in Figure 2.4. This pendulum has a length of ℓ and is composed of a single point mass m. Calculate the Hamiltonian of the system. Deduce the state equations from this.

2) Show that if a system is described by Hamilton equations, then the Hamiltonian is constant.

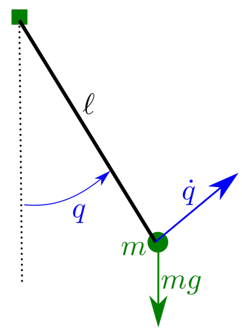


Figure 2.4: Simple pendulum as a conservative system

EXERCISE 2.5.– *Model an inverted rod pendulum*

See the correction video at https://youtu.be/ZI8EaNBDDp0

Let us consider the so-called *inverted rod pendulum* system, composed of a pendulum of length ℓ placed in an unstable equilibrium on a carriage, as represented in Figure 2.5. The value u is the force exerted on the carriage of mass m_c, s indicates the position of the carriage, θ is the angle between the pendulum and the vertical axis and $\mathbf{r}$ is the force exerted by the carriage on the pendulum. At the extremity $\mathbf{b}$ of the rod a point mass m_r is fixated. We may ignore the mass of the rod. Finally, $\mathbf{a}$ is the point of articulation between the rod and the carriage and $\boldsymbol{\omega} = \dot{\theta}\mathbf{k}$ is the rotation vector associated with the rod.

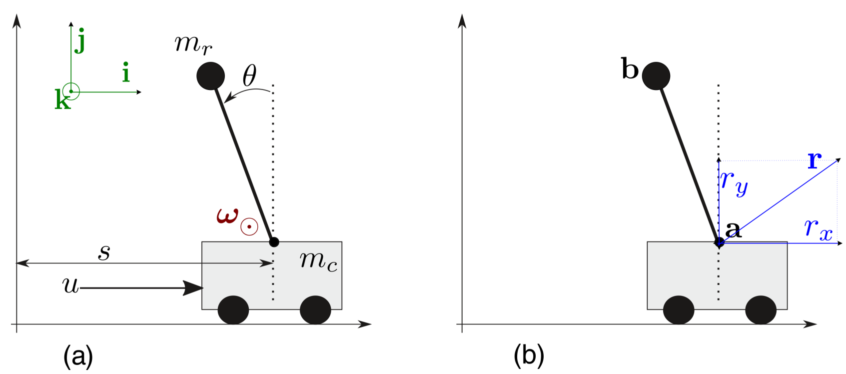


Figure 2.5: Inverted rod pendulum

1) Write the fundamental principle of dynamics as applied on the carriage and the pendulum.

2) Show that the speed vector at point $\mathbf{b}$ is expressed by the relation $\dot{\mathbf{b}} = (\dot{s} - \dot{\theta}\cos\theta) \cdot \mathbf{i} - \dot{\theta}\sin\theta \cdot \mathbf{j}$. Calculate the acceleration $\ddot{\mathbf{b}}$ of $\mathbf{b}$.

3) In order to model the inverted pendulum, we will take the state vector $\mathbf{x} = \left(s, \theta, \dot{s}, \dot{\theta}\right)$. Justify this choice.

4) Find the state equations for the inverted rod pendulum.

EXERCISE 2.6.– *Model the Segway*

See the correction video at https://youtu.be/u4cnGvygFPI

The segway represented on the left side of Figure 2.6 is a vehicle with two wheels and a single axle. It is stable since it is controlled. In the modeling step, we will of course assume that the engine is not controlled.

Its open loop behavior is very close to that of the planar unicycle represented in Figure 2.6 on the right hand side. In this figure, u represents the exerted momentum between the body and the wheel.

The link between these two elements is a pivoting pin. We will denote by $\mathbf{b}$ the center of gravity of the body and by $\mathbf{a}$ that of the wheel. $\mathbf{c}$ is a fix point on the disk. Let us denote by s the angle between the vector $\overrightarrow{\mathbf{ac}}$ and the horizontal axis and by θ the angle between the body of the unicycle and the vertical axis. This system has two degrees of freedom s and θ. The state of our system is given by the vector $\mathbf{x} = (s, \theta, \dot{s}, \dot{\theta})^{\mathrm{T}}$. The parameters are:

- for the disk: its mass M, its radius ρ, its moment of inertia J_M ;
- for the pendulum: its mass m, its moment of inertia J_p, the distance ℓ between its center of gravity and the center of the disk.

Find the state equations of the system.

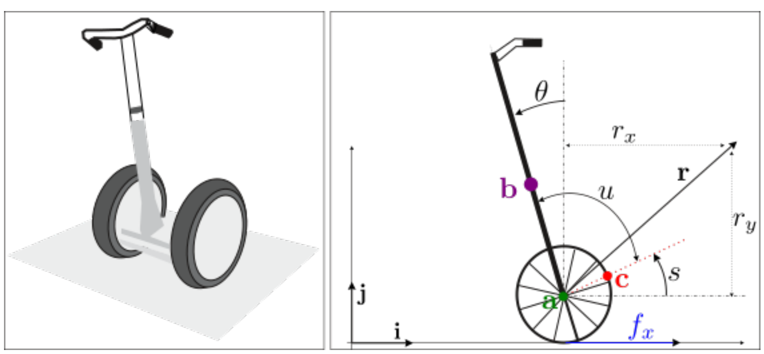


Figure 2.6: The segway has two wheels and an axle

EXERCISE 2.7.– *Modeling a car*

See the correction video at https://youtu.be/b9iBhp56OF8

Let us consider the car as shown in Figure 2.7. The driver of the car (on the left hand side on the figure) has two controls: the acceleration of the front wheels (assumed to be motorized) and the rotation speed of the steering wheel. The brakes here represent a negative acceleration. We will denote by δ the angle between the front wheels and the axis of the car, by θ the angle made by the car with respect to the horizontal axis and by (x, y) the coordinates of the middle of the rear axle. The state variables of our system are composed of:

- the position coordinates, in other words all the knowledge necessary to draw the car, more specifically the x, y coordinates of the center of the rear axle, the orientation θ of the car, and the angle δ of the front wheels;
- the kinetic coordinate v representing the speed of the center of the front axle (indeed, the sole knowledge of this value and the position coordinates allows to calculate all the speeds of all the other elements of the car).

Calculate the state equations of the system. We will assume that the two wheels have the same speed v (even though in reality, the inner wheel during a turn is slower than the outer one). Thus, as illustrated on the right hand side figure, everything happens as if there were only two virtual wheels situated at the middle of the axles.

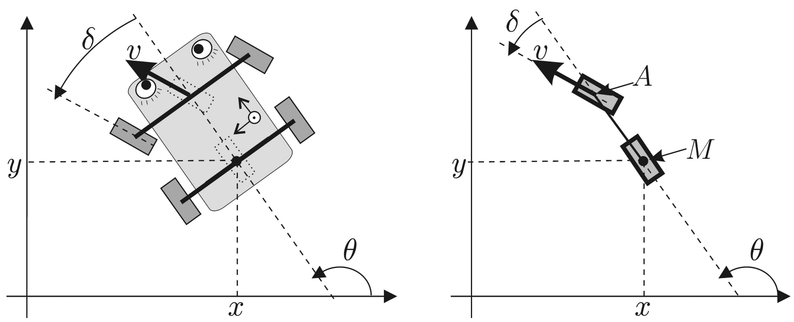


Figure 2.7: Car moving on a plane (view from above)

Exercise 2.8.– *RLC circuit*

See the correction video at https://youtu.be/yD6cwoUQBwI

The electrical circuit of Figure 2.8 has $u(t)$ as input the voltage and $y(t)$ as output the voltage. Find the state equations of the system. Is this a linear system?

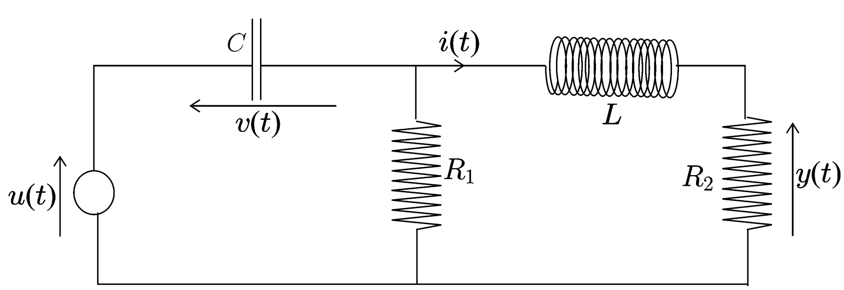


Figure 2.8: Electrical circuit to be modeled

Exercise 2.9.– *DC motor*

See the correction video at https://youtu.be/GknV51O80z8

A direct current motor can be described by Figure 2.9, in which u is the supply voltage of the motor, i is the current absorbed by the motor, R is the resistance, L is the inductance, e is the electromotive force, ρ is the coefficient of friction in the motor, ω is the angular speed of the motor and T_r is the torque exerted by the motor on the load.

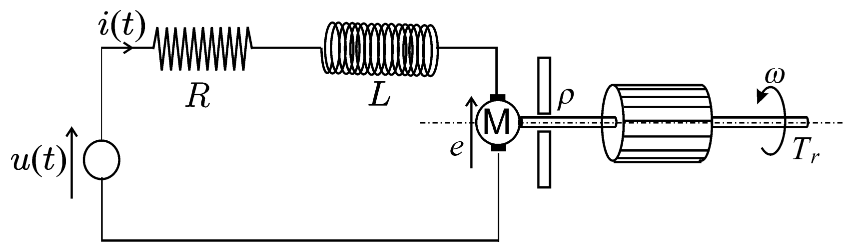


Figure 2.9: Direct current motor

Recall the equations of an ideal direct current motor: $e = K\Phi\omega$ and $T = K\Phi i$. In the case of an induction-independent motor, or a motor with permanent magnets, the flow Φ is constant. We are going to put ourselves in this situation.

1) We take as inputs of the system T_r and u. Find the state equations ;

2) We connect a fan to the output of the system with a characteristic of $T_r = \alpha\omega^2$. Give the new state equations of the motor.

Exercise 2.10.– *Water tanks*

See the correction video at https://youtu.be/Nwacp2T2MLo

Let us consider two containers placed as shown in Figure 2.10. In the left container, the water flows without friction in the direction of the right container. In the left container, the water flows in a fluid way, as opposed to the right container, where there are turbulences. These are turbulences which absorb the kinetic energy of the water and transform it into heat. Without these turbulences, we would have a perpetual back-and-forth movement of the water between the two containers.

1) If a is the cross-section of the canal, show the so-called *Torricelli*'s law which states that the water flow from the right container to the left one is equal to:

$$Q_D = a.sign\,(z_A - z_B)\sqrt{2g|z_A - z_B|}$$

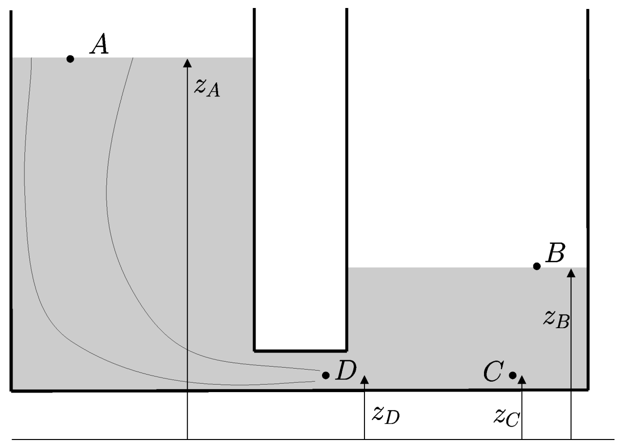


Figure 2.10: Hydraulic system composed of two containers filled with water and connected with a canal

2) Let us now consider the system composed of three containers as represented on Figure 2.11.

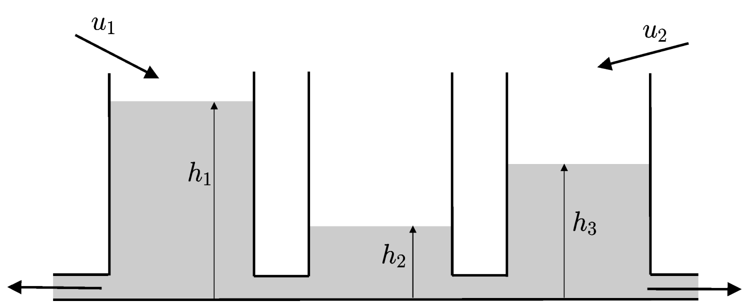


Figure 2.11: System composed of three containers filled with water and connected with two canals

The water from containers 1 and 3 can flow towards container 2, but also towards the outside with atmospheric pressure. The state variables of this system which may be considered are the heights of the containers. In order to simplify, we will assume that the surfaces of the containers are all equal to $1m^2$, thus the volume of water in a container is interlinked with its height. Find the state equations describing the dynamics of the system.

EXERCISE 2.11.– *Fibonacci sequence*

See the correction video at https://youtu.be/T3lL1Pjy2xk

We will now study the evolution of the number $y(k)$ of rabbit couples on a farm in function of the year k. At year 0, there is only a single couple of newly-born rabbits on the farm (and thus $y(0) = 1$). The rabbits only become fertile a year after their birth. It follows that at year 1, there is still a single couple of rabbits, but this couple is fertile (and thus $y(1) = 1$). A fertile couple gives birth, each year, to another couple of rabbits. Thus, at year 2, there is a fertile couple of rabbits and a newly-born couple. This evolution can be described by Table 2.1, where N means *newly-born* and A means *adult*.

$k=0$	$k=1$	$k=2$	$k=3$	$k=4$
N	A	A	A	A
		N	A	A
			N	A
				N
				N

Table 2.1: Evolution of the number of rabbits

Let us denote by $x_1(k)$ the number of newly-born couples, by $x_2(k)$ the number of fertile couples and by $y(k)$ the total number of couples.

1) Give the state equations which govern the system ;

2) Give the recurrence relation satisfied by $y(k)$.

Exercise 2.12.– *Bus network*

See the correction video at https://youtu.be/o4eEAhgxTa4

We consider a public transport system of buses with 4 lines and 4 buses. There are only two stations where travelers can change lines. This system can be represented by a Petri net (see Figure 2.12). Each token corresponds to a bus. The places p_1, p_2, p_3, p_4 represent the lines. These places are composed of a number which corresponds to the minimum amount of time that the token must remain in its place (this corresponds to the transit time). The transitions t_1, t_2 ensure synchronization. They are only crossed when each upstream place of the transition has at least one token which has waited sufficiently long. In this case, the upstream places lose a token and the downstream places gain one. This structure ensures that the correspondence will be performed systematically and that the buses will leave in pairs.

1) Let us assume that at time $t=0$, the transitions t_1 and t_2 are crossed for the first time and that we are in the configuration of Figure 2.12 (this corresponds to the initialization). Give the crossing times for each of the transitions.

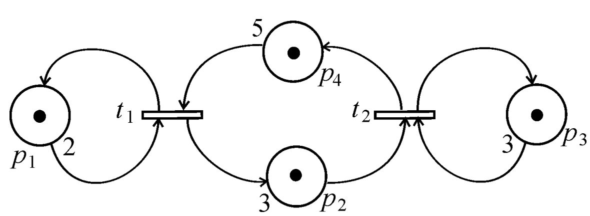


Figure 2.12: Petri net of the bus system

2) Let us denote by $x_i(k)$ the time when the transition t_i is crossed for the k^{th} time. Show that the dynamics of the model can be written using states:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k))$$

where $\mathbf{x} = (x_1, x_2)^{\mathrm{T}}$ is the state vector. Remember that here k is not the time, but an event number.

3) Let us now attempt to reformulate elementary algebra by redefining the addition and multiplication operators (see [8]) as follows:

$$\begin{cases} a \oplus b = \max(a, b) \\ a \otimes b = a + b \end{cases}$$

Thus $2 \oplus 3 = 3$ whereas $2 \otimes 3 = 5$. Show that in this new algebra (called max-plus), the previous system is linear. Discuss.

Chapter 3

Simulation

In this chapter, we will show how to perform a computer simulation of a non-linear system described by its state equations:

$$\left\{\begin{array}{lll}\dot{\mathbf{x}}(t) & = & \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t)) \\ \mathbf{y}(t) & = & \mathbf{g}(\mathbf{x}(t),\mathbf{u}(t))\end{array}\right.$$

This step is important in order to test the behavior of a system (controlled or not). Before presenting the simulation method, we will introduce the concept of vector fields. This concept will allow us to better understand the simulation method as well as certain behaviors which could appear in non-linear systems. We will also give several concepts of graphics necessary for the graphical representation of our systems.

3.1 Concept of vector field

We will now present the concept of vector fields and show the manner in which they are useful in order to better understand the various behaviors of systems. We invite the reader to consult the book of Khalil [9] for further details on this subject. A *vector field* is a continuous function $\mathbf{f}$ of $\mathbb{R}^n$ to $\mathbb{R}^n$. When $n = 2$, a graphical representation of the function $\mathbf{f}$ can be imagined. For instance, the vector field associated with the linear function:

$$\mathbf{f}: \begin{array}{ccc} \mathbb{R}^2 & \to & \mathbb{R}^2 \\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} & \to & \begin{pmatrix} x_1 + x_2 \\ x_1 - x_2 \end{pmatrix} \end{array}$$

is illustrated in Figure 3.1. In order to obtain this figure, we have taken a set of vectors from the initial set, following a grid. Then, for each grid vector $\mathbf{x}$, we have drawn its image vector $\mathbf{f}(\mathbf{x})$ by giving it the vector $\mathbf{x}$ as origin.

We may recognize on this figure the characteristic spaces (dotted lines) of the linear application. We can also see that one eigenvalue is positive and another one is negative. This can be verified by analyzing the matrix of our linear application given by:

$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

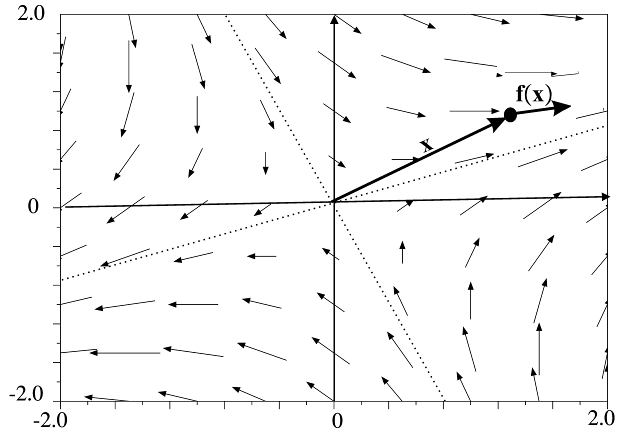


Figure 3.1: Vector field associated with a linear application

Its eigenvalues are $\sqrt{2}$ and $-\sqrt{2}$ and the associated eigenvectors are:

$$\mathbf{v}_1 = \begin{pmatrix} 0.9239 \\ 0.3827 \end{pmatrix} \text{ et } \mathbf{v}_2 = \begin{pmatrix} -0.3827 \\ 0.9239 \end{pmatrix}$$

Let us note that the vector $\mathbf{x}$ represented on the figure is not an eigenvector since $\mathbf{x}$ and $\mathbf{f}(\mathbf{x})$ are not collinear. On the other hand, all the vectors that belong to the characteristic subspaces (represented as dotted lines on the figure) are eigenvectors. Along the characteristic subspace associated with the negative eigenvalue, the field vectors tend to point towards $\mathbf{0}$ whereas these vectors point to infinity along the characteristic subspace associated with the positive eigenvalue.

For an autonomous system (i.e., without input), the evolution is given by the equation $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t))$. When $\mathbf{f}$ is a function of $\mathbb{R}^2$ to $\mathbb{R}^2$, we can obtain a graphical representation of $\mathbf{f}$ by drawing the vector field associated with $\mathbf{f}$. The graph will then allow us to better understand the behavior of our system.

3.2 Graphical representation

In this paragraph, we will give several concepts necessary for the graphical representation of systems during simulations.

3.2.1 Sketchs

A *sketch* is a matrix with two or three rows (following whether the object is in the plane or in space) and n columns which represent the n vertices of a shape-retaining polygon, meant to represent the object. It is important that the unions of all the segments formed by the two consecutive points of the sketch form the edges of the polygon that we wish to represent. For instance, the sketch $\mathbf{M}$ of the chassis (see Figure 3.2) of the car (with the rear wheels) is given by:

$$\begin{pmatrix} -1 & 4 & 5 & 5 & 4 & -1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & -1 & 1 & 0 & 0 & 3 & 3 & 3 \\ -2 & -2 & -1 & 1 & 2 & 2 & -2 & -2 & -3 & -3 & -3 & -3 & 3 & 3 & 3 & 3 & 2 & 2 & 3 & -3 \end{pmatrix}$$

It is clear on the graph of the car in movement that the front wheels can move with respect to the chassis, but also with respect to one another. They therefore cannot be incorporated into the sketch of the chassis. For the graph of the car, we will therefore have to use 3 sketches: that of the chassis, that of the left front wheel and that of the right front wheel.

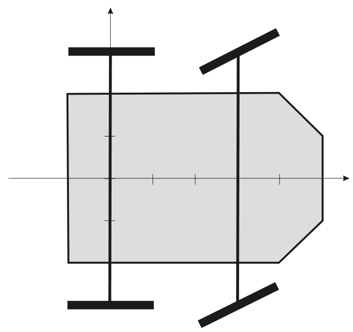


Figure 3.2: Car to be represented graphically

3.2.2 Rotation matrix

Let us recall that the j^{th} column of the matrix of a linear application of $\mathbb{R}^n \to \mathbb{R}^n$ represents the image of the j^{th} vector $\mathbf{e}_j$ of the standard basis. Thus, the expression of a rotation matrix of angle θ in the plane $\mathbb{R}^2$ is given by (see Figure 3.3):

$$\mathbf{R} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

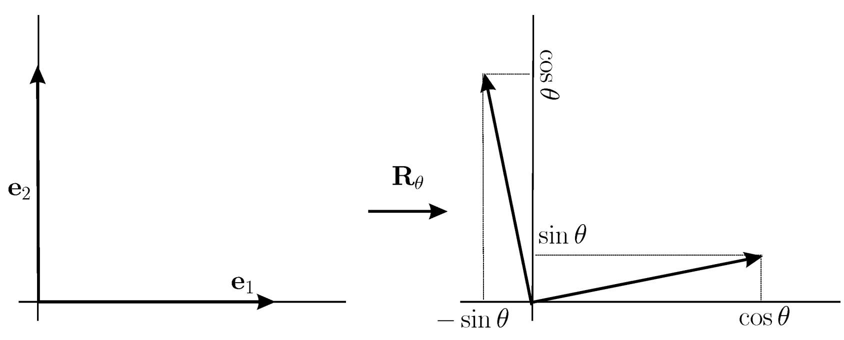


Figure 3.3: Rotation of angle θ

Concerning rotations in the space $\mathbb{R}^3$, it is important to specify the rotation axis. We can distinguish between 3 main rotations: the rotation of angle θ around the Ox axis, the one around the Oy axis and the one around the Oz axis. The associated matrices are, respectively, given by:

$$\mathbf{R}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}, \ \mathbf{R}_y = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \text{ et } \mathbf{R}_z = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

3.2.3 Homogeneous coordinates

Drawing two-dimensional or three-dimensional objects on a screen requires a series of affine transformations (rotations, translations, homotheties) of the form:

$$\mathbf{f}_i : \begin{array}{ccc} \mathbb{R}^n & \to & \mathbb{R}^n \\ \mathbf{x} & \mapsto & \mathbf{A}_i\mathbf{x} + \mathbf{b}_i \end{array}$$

with $n = 2$ or 3. However, the manipulation of compositions of affine functions is not as simple as that of linear applications. The idea of the transformation into *homogeneous coordinates* is to transform a system of affine equations into a system of linear equations. Let us note first of all that an affine equation of type $\mathbf{y} = \mathbf{Ax} + \mathbf{b}$ can be re-written in the form:

$$\begin{pmatrix} \mathbf{y} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{b} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

We will therefore define the *homogeneous transformation* of a vector as follows:

$$\mathbf{x} \mapsto \mathbf{x}_{\mathrm{h}} = \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$$

Thus, an equation such as:

$$\mathbf{y} = \mathbf{A}_3 \left(\mathbf{A}_2 \left(\mathbf{A}_1 \mathbf{x} + \mathbf{b}_1 \right) + \mathbf{b}_2 \right) + \mathbf{b}_3$$

where there is a composition of 3 affine transformations can be re-written as:

$$\mathbf{y}_{\mathrm{h}} = \begin{pmatrix} \mathbf{A}_3 & \mathbf{b}_3 \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} \mathbf{A}_2 & \mathbf{b}_2 \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} \mathbf{A}_1 & \mathbf{b}_1 \\ \mathbf{0} & 1 \end{pmatrix} \mathbf{x}_{\mathrm{h}}.$$

Two transformation matrices are useful when dealing with 3D graphics in homogeneous coordinates:

- The rotation around a vector $\boldsymbol{\omega}$ of an angle $\varphi = ||\boldsymbol{\omega}||$ is given by

$$\mathrm{Rot}(\boldsymbol{\omega}) = \begin{pmatrix} \exp \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} & 1 \end{pmatrix} \tag{3.1}$$

  which corresponds to the *Rodrigues' formula.*

- The translation matrix by a vector $\mathbf{v}$ of $\mathbb{R}^3$ is:

$$\mathrm{Tran}(\mathbf{v}) = \begin{pmatrix} 1 & 0 & 0 & v_1 \\ 0 & 1 & 0 & v_2 \\ 0 & 0 & 1 & v_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.2}$$

3.3 Simulation

In this paragraph, we will present the integration method to perform a computer simulation of a non-linear system described by its state equations:

$$\begin{cases} \dot{\mathbf{x}}(t) & = & \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) & = & \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \end{cases}$$

This method is rather approximative, but remains simple to understand and is enough in order to describe the behaviors of most of robotized systems.

3.3.1 Euler's method

Let dt be a very small number compared to the time constants of the system and which corresponds to the sampling period of the method (for example $dt = 0.01$). The evolution equation is approximated by:

$$\frac{\mathbf{x}(t+dt)-\mathbf{x}(t)}{dt} \simeq \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$$

in other words:

$$\mathbf{x}(t+dt) \simeq \mathbf{x}(t) + \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)).dt$$

This equation can be interpreted as an order 1 Taylor formula. From this we can deduce the simulation algorithm (called *Euler's method*):

Algorithm EULER(in: $\mathbf{x}_0$)
$\mathbf{x} := \mathbf{x}_0; t := 0;\ dt = 0.01;$
repeat
read the input $\mathbf{u}$;
$\mathbf{y} := \mathbf{g}(\mathbf{x}, \mathbf{u});$
return $\mathbf{y}$;
$\mathbf{x} := \mathbf{x} + \mathbf{f}(\mathbf{x}, \mathbf{u}).dt;$
wait for interrupt from timer;
$t = t + dt;$
while true

The timer creates a periodic interrupt every dt seconds. Thus, if the computer is sufficiently fast, the simulation is performed at the same speed as our physical system. We then refer to *real-time* simulation. In some circumstances, what we are interested in is obtaining the result of the simulation in the fastest possible time (for instance in order to predict how a system will behave in the future). In this case, it is not necessary to slow the computer down in order to synchronize it with our physical time.

We call *local error* the quantity:

$$e_t = ||\mathbf{x}(t+dt) - \hat{\mathbf{x}}(t+dt)|| \text{ with } \mathbf{x}(t) = \hat{\mathbf{x}}(t)$$

where $\mathbf{x}(t+dt)$ is the exact solution of the differential equation $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u})$ and $\hat{\mathbf{x}}(t+dt)$ is the estimated value of the state vector, for the integration scheme being used. For Euler's method, we can show that e_t is of order 1, i.e. $e_t = O(dt)$.

3.3.2 Runge-Kutta method

There are more efficient integration methods in which the local error is of order 2 or more. This is the case of the Runge-Kutta method of order 2, which consists of replacing the recurrence $\hat{\mathbf{x}}(t+dt) := \hat{\mathbf{x}}(t) + \mathbf{f}\left(\hat{\mathbf{x}}(t), \mathbf{u}(t)\right).dt$ with:

$$\hat{\mathbf{x}}(t+dt) = \hat{\mathbf{x}}(t) + dt.\mathbf{f}\left(\underbrace{\hat{\mathbf{x}}(t) + \frac{dt}{2}\mathbf{f}(\hat{\mathbf{x}}(t),\mathbf{u}(t))}_{\hat{\mathbf{x}}_E(t+\frac{dt}{2})}, \mathbf{u}(t+\tfrac{dt}{2})\right)$$

Let us note in this expression the value $\hat{\mathbf{x}}_E(t+\frac{dt}{2})$ which can be interpreted as the integration obtained by Euler's method at time $t+\frac{dt}{2}$. The local error e_t here is of order 2 and the integration method is therefore a lot more precise. There are Runge-Kutta methods of order higher than 2 which we will not discuss here.

3.3.3 Taylor's method

Euler's method (which is an order 1 Taylor method) can be extended to higher orders. Let us show, without loss of generality, how to extend to second order. We have:

$$\mathbf{x}\left(t+dt\right) = \mathbf{x}\left(t\right) + \dot{\mathbf{x}}(t).dt + \frac{1}{2}\ddot{\mathbf{x}}(t).dt^2 + o\left(dt^2\right)$$

But:

$$\begin{aligned}
\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t)) \\
\ddot{\mathbf{x}}(t) &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}(t),\mathbf{u}(t)).\dot{\mathbf{x}}(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{x}(t),\mathbf{u}(t)).\dot{\mathbf{u}}(t)
\end{aligned}$$

Therefore, the integration scheme becomes:

$$\begin{aligned}
\hat{\mathbf{x}}(t+dt) \;=\;& \hat{\mathbf{x}}(t) + dt.\mathbf{f}(\hat{\mathbf{x}}(t),\mathbf{u}(t)) \\
& +\tfrac{1}{2}dt^2.\left(\tfrac{\partial \mathbf{f}}{\partial \mathbf{x}}(\hat{\mathbf{x}}(t),\mathbf{u}(t)).\mathbf{f}(\hat{\mathbf{x}}(t),\mathbf{u}(t)) + \tfrac{\partial \mathbf{f}}{\partial \mathbf{u}}(\hat{\mathbf{x}}(t),\mathbf{u}(t)).\dot{\mathbf{u}}(t)\right)
\end{aligned}$$

Exercises

EXERCISE 3.1.– *Prey-predator model*

See the correction video at https://youtu.be/22OD0hPsz7w

The *predator-prey* system, also called a Lotka-Volterra system, is given by:

$$\begin{cases} \dot{x}_1(t) &=& (1 - x_2(t))\, x_1(t) \\ \dot{x}_2(t) &=& (x_1(t) - 1)\, x_2(t) \end{cases}$$

The state variables $x_1(t)$ and $x_2(t)$ represent the size of the prey and predator populations. For example, x_1 could represent the number of preys in millions whereas x_2 could be the number of predators, in thousands. Even though the number of preys and predators are integers, we will assume that x_1 and x_2 are real values. The quadratic terms of this state equation represent the interactions between the two species. Let us note that the preys grow in an exponential manner when there are no predators. Similarly, the population of predators declines when there is no prey.

1) Figure 3.4 corresponds to the vector field associated with the evolution function:

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} (1 - x_2)\, x_1 \\ (x_1 - 1)\, x_2 \end{pmatrix}$$

on the grid $[0, 2] \times [0, 2]$. Discuss the dynamic behavior of the system using this figure.

2) Also using this figure, give the equilibrium point. Verify it through calculation.

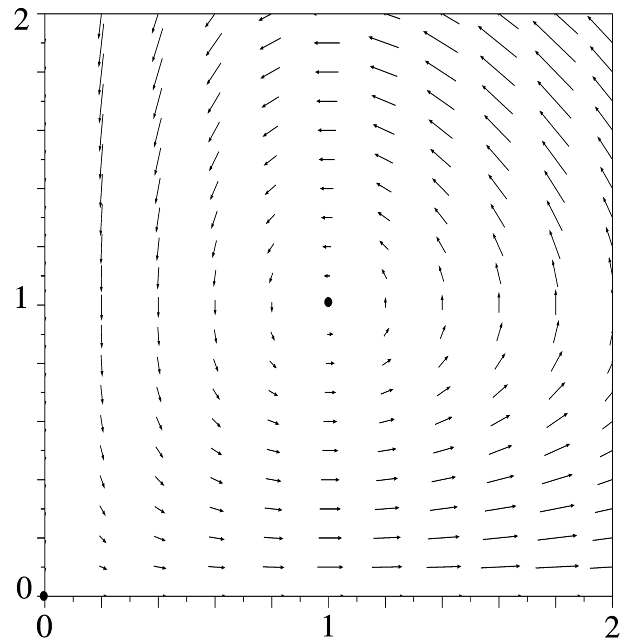


Figure 3.4: Vector field associated with the Lotka-Volterra system, in the plane (x_1, x_2)

EXERCISE 3.2.– *Simple pendulum*

See the correction video at https://youtu.be/pzNEuLJmR6A

Let us consider the simple pendulum of Figure 3.5 described by the following state equations:

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\sin x_1 \end{cases}$$

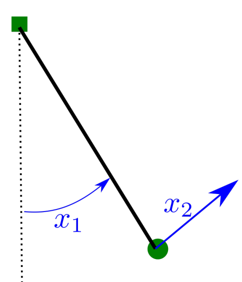


Figure 3.5: Simple pendulum with state vector $\mathbf{x} = (x_1, x_2) = (\theta, \dot{\theta})$

The vector field associated with the evolution function $\mathbf{f}(\mathbf{x})$ is drawn on Figure 3.6.

1) From the vector field, give the stable and unstable points of equilibrium.

2) Program an Euler integration and draw the trajectory on the state space.

3) Compare the Euler integration with a Runge-Kutta method.

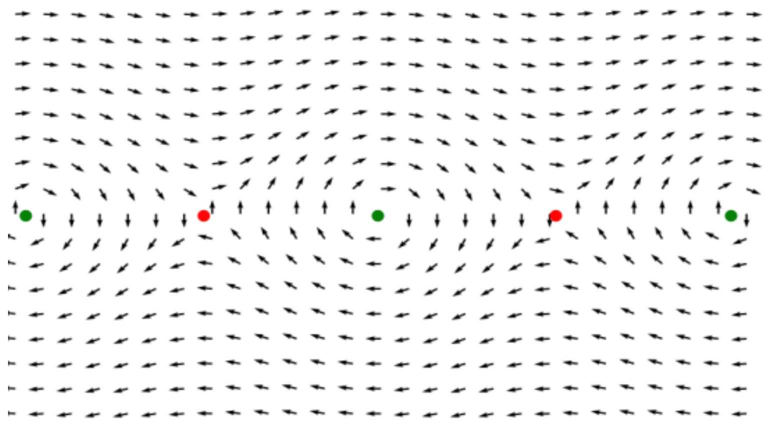


Figure 3.6: Vector field associated with the simple pendulum in the phase plane $(x_1, x_2) = (\theta, \dot{\theta})$

Exercise 3.3.– *Van der Pol system*

See the correction video at https://youtu.be/SlQrNliwHIc

Let us consider the system described by the following differential equation:

$$\ddot{y} + \left(y^2 - 1\right)\dot{y} + y = 0$$

1) Let us choose as state vector $\mathbf{x} = (y, \dot{y})$. Give the state equations of the system.

2) Linearize this system around the equilibrium point. What are the poles of the system? Is the system stable around the equilibrium point?

3) The vector field associated with this system is represented on Figure 3.7 in the state space (x_1, x_2). We initialize the system in $\mathbf{x}_0 = (0.1, 0)$. Draw on the figure the path $\mathbf{x}(t)$ of the system on the state space. Give the form of $y(t)$.

4) Can a trajectory have a loop?

5) Simulate this system, with an Euler method and also with a Runge-Kutta method. Compare both integration schemes.

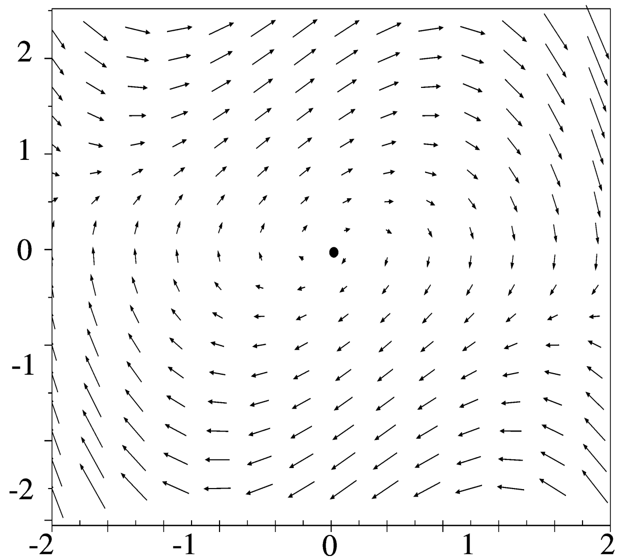


Figure 3.7: Vector field associated with the Van der Pol system

---

EXERCISE 3.4.– *Simulation of a car*

See the correction video at https://youtu.be/LV9QVSt8ll0

Let us consider the car with the following state equations:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\delta} \end{pmatrix} = \begin{pmatrix} v\cos\delta\cos\theta \\ v\cos\delta\sin\theta \\ \frac{v\sin\delta}{L} \\ u_1 \\ u_2 \end{pmatrix}$$

The state vector is given by $\mathbf{x} = (x, y, \theta, v, \delta)$, where x, y, θ corresponds to the pose of the car (in other words its position and orientation), v is the speed and δ is the angle of the front wheels. The parameter $L = 3$m is the distance between the two axles of the car.

1) Using homogeneous coordinates, design a function which draws a car in a state $\mathbf{x} = (x, y, \theta, v, \delta)^{\mathrm{T}}$

2) Propose a program which simulates the dynamic evolution of this car during 5 seconds with Euler's method and a sampling step of 0.01 sec. Take the initial state for the car as $\mathbf{x}(0) = (0, 0, 0, 50, 0)$, which means that at time $t = 0$, the car is centered around the origin, with a nil heading angle, a speed of $50ms^{-1}$ and the front wheels parallel to the axis of the car. We assume that the vectorial control $\mathbf{u}(t)$ remains constant and equal to $(0,\, 0.05)$. Which means that the car does not accelerate (since $u_1 = 0$) and that the steering wheel is turning at a constant speed of $0.05\,\mathrm{rad}\cdot\mathrm{s}^{-1}$.

Exercise 3.5.– *Integration by Taylor's method*

See the correction video at https://youtu.be/wtXp0RF8tfQ

Let us consider a robot described by the following state equations:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{pmatrix} = \begin{pmatrix} x_4 \cos x_3 \\ x_4 \sin x_3 \\ x_5 \\ u_1 \\ u_2 \end{pmatrix}$$

1) Propose a second order integration scheme using Taylor's method. The input is:

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \cos t \\ \sin t \end{pmatrix}.$$

2) Simulate the car for $t \in [0, 12]$, with the three following integration schemes: *(i)* Euler, *(ii)* Runge Kutta of order 2 and *(iii)* the Taylor of order 2. For the initial vector, we take $\mathbf{x}(0) = (0, 0, 0, 5, -\frac{1}{2})$ and the sampling time is taken as $dt = 0.05$s. Compare and discuss.

Exercise 3.6.– *Rotating cube*

See the correction video at https://youtu.be/a1AwVPjyvkk

Let us consider the three-dimensional cube $[0, 1] \times [0, 1] \times [0, 1]$.

1) Give its sketch in matrix form.

2) What matrix operation must be performed in order to rotate it with an angle θ around the Ox axis?

3) Illustrate the rotation with a program where the cube turns with time around the Ox axis.

Exercise 3.7.– *Three-dimensional simulation of a tricycle*

See the correction video at https://youtu.be/BapNhAtJYe0

The driver of the tricycle of Figure 3.8 has two controls: the acceleration of the front wheel u_1 and the rotation rate u_2 of the steering angle δ. The state variables of our system are composed of

- the degrees of freedom, *i.e.*, the x, y coordinates of the center of the rear axle, the heading θ and the steering angle δ;
- the speed v of the center of the front wheel;
- the longitudinal angles of the wheels are s_1, s_2, s_3

The parameters are

- the distance between the rear axle and the center of the front wheel: $L = 3$m;
- the distance between each rear wheel and the axis of the tricycle: $e = 2$m;
- the radius of the wheels is $r = 0.5$m.

1) Give the state equations of the three dimensional tricycle.

2) Provide a program which draws the tricycle, at a given state. For this, rotate and translate the sketch in 3D using the homogeneous coordinates and Rodrigues' formula.

3) Provide a simulation of the three-dimensional tricycle.

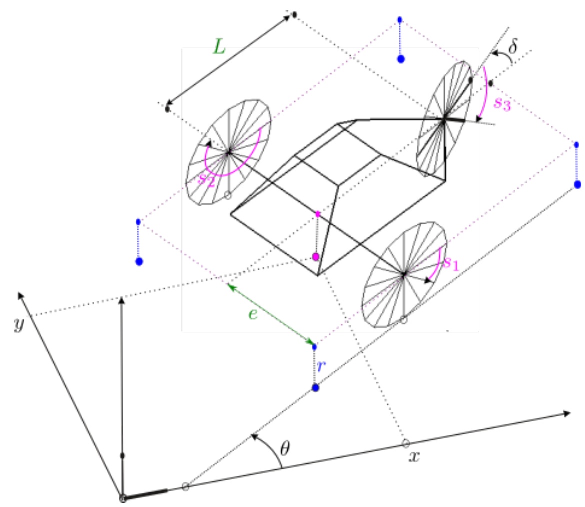


Figure 3.8: Tricycle to be simulated

Exercise 3.8.– *Manipulator robot*

See the correction video at https://youtu.be/ig3E1Lyd-Fw

The manipulator robot represented on Figure 3.9 is composed of three arms in series. The first one, of length 3, can rotate around the Oz axis. The second, of length 2, placed at the end of the first one can also rotate around the Oz axis. The third one, of length 1, placed at the end of the second, can rotate around the axis formed by the second arm. This robot has 3 degrees of freedom $\mathbf{x} = (\alpha_1, \alpha_2, \alpha_3)$, where the α_i represent the angles formed by each of the arms. The basic sketch chosen to represent each of the arms if the unit cube. Each arm is assumed to be a parallelepiped of thickness 0.3. In order to take the form of the arm, the sketch must be subjected to transformation, represented by a diagonal matrix. Then, it has to be rotated and translated in order to be correctly places. Design a program that simulates this system, with a 3D vision inspired from the figure.

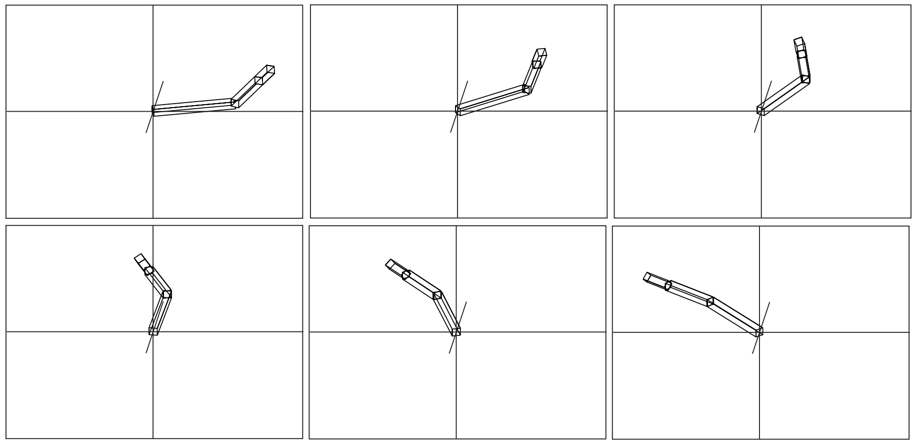


Figure 3.9: Manipulator robot composed of three arms

Exercise 3.9.– *Omni wheel robot*

See the correction video at https://youtu.be/sXmcNt5B5zQ

Let us consider the robot with three omni wheels shown on Figure 3.10. An omni wheel is equipped with a multitude of small rollers over its entire periphery which allow it to slide sideways (in other words perpendicularly to its nominal movement direction). Let us denote by $\mathbf{v}_i$ the speed vector of the contact point of the i^{th} wheel. If $\mathbf{i}_i$ is the normed direction vector indicating the nominal movement direction of the wheel, then the component of $\mathbf{v}_i$ according to $\mathbf{i}_i$ corresponds to the rotation ω_i of the wheel whereas its complementary component (perpendicular to $\mathbf{i}_i$) is linked to the rotation of the peripheral rollers. If r is the radius of the wheel, then we have the relation $r\omega_i = -\langle \mathbf{v}_i, \mathbf{i}_i \rangle$. If $\mathbf{v}_i$ and $\mathbf{u}_i$ are collinear then the wheel behaves like a classical wheel. If $\langle \mathbf{v}_i, \mathbf{i}_i \rangle = 0$, the wheel no longer turns and it is in a state of skid.

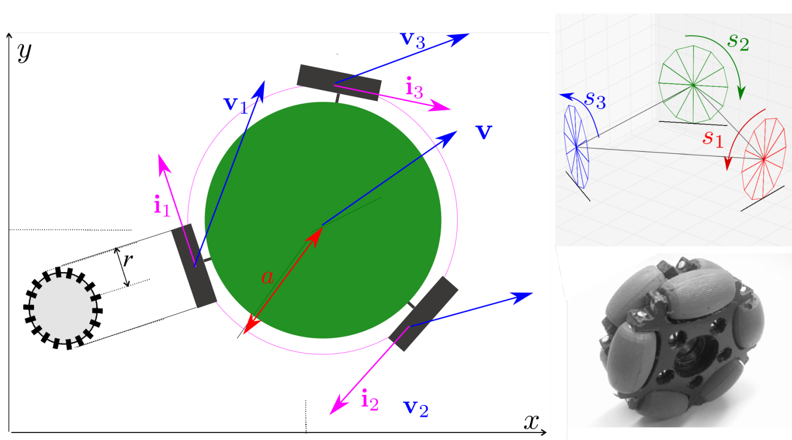


Figure 3.10: Holonomic robot with omni wheels

1) Show that

$$\begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} = \frac{1}{r} \begin{pmatrix} \sin\theta & -\cos\theta & -a \\ \sin(\theta+\frac{2\pi}{3}) & -\cos(\theta+\frac{2\pi}{3}) & -a \\ \sin(\theta-\frac{2\pi}{3}) & -\cos(\theta-\frac{2\pi}{3}) & -a \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix}$$

2) Give the state equations of the system. We will use the state vector $\mathbf{x} = (x, y, \theta, s_1, s_2, s_3)$ and the input vector $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$ where the s_i are the proper angle of the wheels. Check the behavior of the system using a 3D simulation.

3) Propose a loop which allows to obtain a model tank described by the following state equations:

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = u_1 \\ \dot{v} = u_2 \end{cases}$$

Exercise 3.10.– *Snake*

See the correction video at https://youtu.be/CwZRSnBfo8U

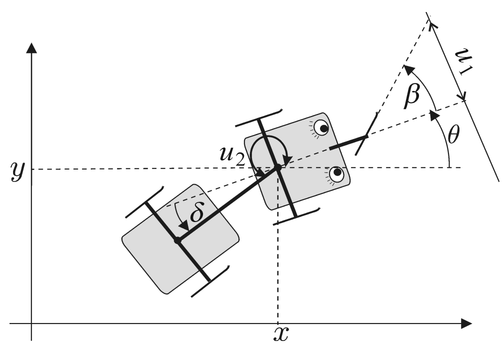


Figure 3.11: Snake or Skating robot

The *snake* is the skating vehicle represented in Figure 3.11. It is designed to move on a frozen lake and stands on 5 ice-skates [10]. This system has two inputs: the tangent u_1 of angle β of the front skate we have chosen the tangent as input in order to avoid singularities) and u_2 the torque exerted on the articulation between the two sledges and which corresponds to the angle δ. The propulsion therefore only comes from the torque u_2 and is similar to the mode of propulsion of a snake or an eel [11]. Any control over u_1 therefore doesn't give any energy to the system.

1) Show that the system can be described by the following state equations:

$$\begin{cases} \dot{x} &= & v\cos\theta \\ \dot{y} &= & v\sin\theta \\ \dot{\theta} &= & vu_1 \\ \dot{v} &= & -(u_1+\sin\delta)\,u_2 - v \\ \dot{\delta} &= & -v\,(u_1+\sin\delta) \end{cases}$$

where v is the speed of the center of the front axle. This is of course a simplified and normalized model where, for reasons of simplicity, the coefficients (masses, viscous friction, inter-axle distances, etc.) have been chosen unitary.

2) Simulate this system by using Euler's method. Discuss.

3) We will now attempt to control this system using *biomimicry*, i.e. by attempting to reproduce the propulsion of the snake. We choose u_1 to be of the form:

$$u_1 = p_1 \cos(p_2 t) + p_3$$

where p_1 is the amplitude, p_2 the pulse and p_3 the bias. We choose u_2 so that the propulsion torque is always a driving force, in other words $\dot{\delta}.u_2 \geq 0$. Indeed, the term $\dot{\delta}.u_2$ corresponds to the power brought by the robot which is transformed into kinetic energy. Program this control and make the right choice for the parameters which allow to ensure an efficient propulsion. Reproduce the two behaviors illustrated on Figure 3.12 on your computer.

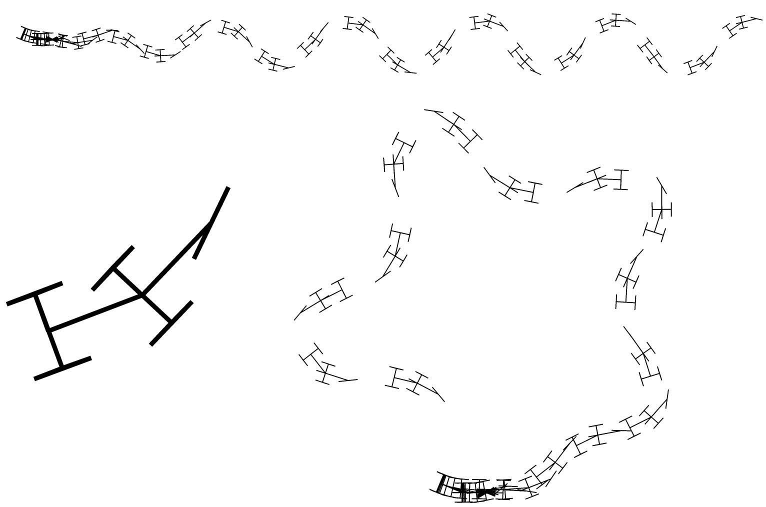


Figure 3.12: Simulations of the controlled skating robot

4) Add a second control loop which controls the parameters p_i of your controller in order for your robot to be able to follow a desired heading $\bar{\theta}$.

Exercise 3.11.– *Swimbox*

See the correction video at https://youtu.be/k37a_fM8pCs

A *swimbox* is an imaginary vehicle which swims horizontally at the bottom of the ocean. The sea floor is assumed to be flat. The swimbox has no propeller, but contains as a unique actuator a weight that can move left and right. This is illustrated by Figure 3.13 where

- u the reaction force on the weight made by the hull;
- x_1 is the position of the hull;

- $x_2 \in [-1, 1]$ is the position of the weight in the hull;
- v_1 is the speed of the hull with respect the the sea floor;
- v_2 is the speed of the weight with respect to the hull.

1) Assuming that all coefficients (mass, frictions, etc) taken as 1, show that a possible model for the system is given by

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{v}_1 \\ \dot{v}_2 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ -u - v_1 \cdot |v_1| \\ 2u + v_1 \cdot |v_1| \end{pmatrix}$$

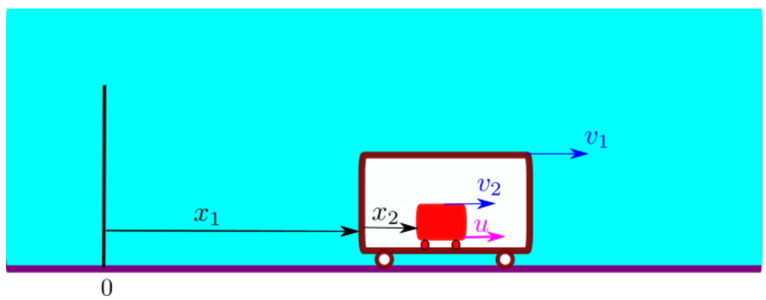


Figure 3.13: The swimbox is composed of a hull and a weight. The weight (red) can move horizontally

2) Find a controller such that the swimbox swims forward for a long distance always satisfying the constraint $x_2 \in [-1, 1]$.

EXERCISE 3.12.– *Virus*

See the correction video at https://youtu.be/Gghwa53kQNc

We consider a population of N individuals. We want to study the evolution of the virus in this population. We denote by

- $x_1(t)$, the number of individuals not yet infected with the virus.
- $x_2(t)$, the number of infected individuals. All of them are capable of spreading the disease.
- $x_3(t)$, the individuals that have been cured from the virus. Those in this category will not be infected again or transmit the virus to others.

For simplicity, we consider that a the time unit for t is in days. We assume that

- **(A1)** an individual in the population not yet infected has a probability to become infected at each unit of time which is proportional to the density of infected individuals.

- **(A2)** an individual which is infected at time t has a constant probability to become cured at time $t + dt$.

1) Find state equations for a model describing the evolution of the system. This model has two parameters related to the assumptions.

2) We assume that

- **(A3)**. If a not-yet-infected individual lives among a population which is completely infected, then the probability to be infected at time $t + dt$ is $0.3 \cdot dt$.
- **(A4)**. If one individual is infected at time t, then, it has a probability of $0.1 \cdot dt$ to be cured at time $t + dt$.

Identify the parameters of the state-space model.

3) When $t \to \infty$, can we know what is the percentage of the population that will never been infected ?

4) Assume that at time $t = 0$, we have one infected individual for a population of 10^8 persons. Using a simulation, predict at which time the peak (the maximum for x_2) occurs.

5) Draw the trajectory in the (x_1, x_2) with the corresponding vector field. Discuss.

Chapter 4

Linear systems

The study of linear systems [12] is fundamental for the proper understanding of the concepts of stability and the design of linear controllers. Let us recall that linear systems are of the form:

$$\left\{\begin{array}{rcl} \dot{\mathbf{x}}(t) & = & \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) & = & \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{array}\right.$$

for continuous-time systems and:

$$\left\{\begin{array}{rcl} \mathbf{x}(k+1) & = & \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) & = & \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \end{array}\right.$$

for discrete-time systems. This chapter introduces the notion of stability and makes the link between a state space representation and the more classical transfer approach.

4.1 Stability

A system of the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ is *stable* if, after a sufficiently long amount of time, the state no longer depends on the initial conditions, no matter what they are. The definition by Lyapunov formalizes the stability as follows. The system is *stable* if for every $\varepsilon > 0$, there exists $\delta > 0$ such that

$$\|\mathbf{x}(0)\| < \delta \Rightarrow \forall t \geq 0, \|\mathbf{x}(t)\| < \varepsilon. \tag{4.1}$$

The system is *asymptotically stable* if it is *Lyapunov stable* and there exists $\delta > 0$ such that

$$\|\mathbf{x}(0)\| < \delta \Rightarrow \lim_{t\to\infty} \|\mathbf{x}(t)\| = 0. \tag{4.2}$$

The same definitions are also valid when the time is discrete. In the case of linear systems, the asymptotic stability translates into:

$$\begin{array}{ll} \lim_{t\to\infty} e^{\mathbf{A}t} = \mathbf{0}_n & \text{if the system is continuous-time} \\ \lim_{k\to\infty} \mathbf{A}^k = \mathbf{0}_n & \text{if the system is discrete-time} \end{array}$$

In this expression we can see the concept of *matrix exponential*. The exponential of a square matrix $\mathbf{M}$ of dimension n can be defined through its power series development:

$$e^{\mathbf{M}} = \mathbf{I}_n + \mathbf{M} + \frac{1}{2!}\mathbf{M}^2 + \frac{1}{3!}\mathbf{M}^3 + \cdots = \sum_{i=0}^{\infty} \frac{1}{i!}\mathbf{M}^i$$

where $\mathbf{I}_n$ is the identity matrix of dimension n. It is clear that $e^{\mathbf{M}}$ is of the same dimension as $\mathbf{M}$. Here are some of the important properties concerning the exponentials of matrices. If $\mathbf{0}_n$ is the zero matrix of $n \times n$ and if $\mathbf{M}$ and $\mathbf{N}$ are two matrices $n \times n$, then:

$$\begin{array}{l} e^{\mathbf{0}_n} = \mathbf{I}_n \\ e^{\mathbf{M}}.e^{\mathbf{N}} = e^{\mathbf{M}+\mathbf{N}} \text{ (if the matrices commute)} \\ \frac{d}{dt}\left(e^{\mathbf{M}t}\right) = \mathbf{M}e^{\mathbf{M}t} \end{array}$$

Criterion of stability.– There is a criterion of stability which only depends on the matrix $\mathbf{A}$. A continuous-time linear system is stable if and only if all the eigenvalues of its evolution matrix $\mathbf{A}$ have strictly negative real parts. A discrete-time linear system is stable if and only if all the eigenvalues of $\mathbf{A}$ are strictly inside the unit circle. This corresponds to the criterion of stability as proven in Exercise 4.2.

Whether it is for continuous-time or discrete-time linear systems, the position of the eigenvalues of $\mathbf{A}$ is of paramount importance in the study of stability of a linear system. The *characteristic polynomial* of a linear system is defined as the characteristic polynomial of the matrix $\mathbf{A}$ which is given by the formula:

$$P(s) = \det(s\mathbf{I}_n - \mathbf{A})$$

Its roots are the eigenvalues of $\mathbf{A}$. Indeed, if s is a root of $P(s)$, then $\det(s\mathbf{I}_n - \mathbf{A}) = 0$, in other words there is a vector $\mathbf{v}$ non-nil such that $(s\mathbf{I}_n - \mathbf{A})\,\mathbf{v} = \mathbf{0}$. This means that $s\mathbf{v} - \mathbf{A}\mathbf{v} = \mathbf{0}$ or $\mathbf{A}\mathbf{v} = s\mathbf{v}$. Therefore s is an eigenvalue of $\mathbf{A}$. A corollary of the stability criterion is thus the following.

Corollary.– A continuous-time linear system is stable if and only if all the roots of its characteristic polynomial have negative real parts. A discrete-time linear system is stable if and only if all the roots of its characteristic polynomial are inside the unit circle.

4.2 Laplace transform

The *Laplace transform* is a very useful tool for the control engineer in the manipulation of systems described by differential equations. This approach is particularly utilized in the context of monovariate systems (i.e. systems with a single input and output) and may be regarded as a competitor to the state-representation approach considered in this book.

4.2.1 Laplace variable

The space of differential operators in $\frac{d}{dt}$ is a ring and has favorable properties such as associativity. For example:

$$\frac{d^4}{dt^4}\left(\frac{d^3}{dt^3}+\frac{d}{dt}\right)=\frac{d^4}{dt^4}\frac{d^3}{dt^3}+\frac{d^4}{dt^4}\frac{d}{dt}=\frac{d^7}{dt^7}+\frac{d^5}{dt^5}$$

This ring is commutative. For example:

$$\frac{d^4}{dt^4}\left(\frac{d^3}{dt^3}+\frac{d}{dt}\right)=\left(\frac{d^3}{dt^3}+\frac{d}{dt}\right)\frac{d^4}{dt^4}$$

We may associate with the operator $\frac{d}{dt}$ the symbol s called *Laplace variable*. Thus the operator $\frac{d^4}{dt^4}\left(\frac{d^3}{dt^3}+\frac{d}{dt}\right)$ will be represented by the polynomial $s^4\left(s^3+s\right)$.

4.2.2 Transfer function

Let us consider a linear system with input u and output y described by a differential relation such as:

$$y(t)=H\left(\frac{d}{dt}\right)\cdot u(t)$$

The function $H(s)$ is called the transfer function of the system. Let us take for instance the system described by the differential equation:

$$\ddot{y}+2\dot{y}+3y=4\dot{u}-5u$$

We have:

$$y(t)=\left(\frac{4\frac{d}{dt}-5}{\frac{d^2}{dt^2}+2\frac{d}{dt}+3}\right)\cdot u(t)$$

Its transfer function is therefore:

$$H(s)=\frac{4s-5}{s^2+2s+3}$$

If the transfer function $H(s)$ of a system is a rational function, its denominator $P(s)$ is called the *characteristic polynomial*.

4.2.3 Laplace transform

We call *Laplace transform* $\hat{y}(s)$ of the signal $y(t)$ the transfer function of the system which generates $y(t)$ from the Dirac delta function $\delta(t)$. We will denote it by $\hat{y}(s)=\mathcal{L}\left(y(t)\right)$. We also say that $y\left(t\right)$ is the *impulse response* of the system. The table below shows several systems together with their transfer function and impulse response.

In this table, $E\left(t\right)$ is the unit step which is equal to 1 if $t\geq 0$ and 0 otherwise. Thus, the Laplace transforms of $\delta\left(t\right), E\left(t\right), \dot{\delta}\left(t\right), \delta(t-\tau)$ are respectively $1, \frac{1}{s}, s, e^{-\tau s}$.

System	Equation	Transfer function	Impulse response
Identity	$y(t) = u(t)$	1	$\delta(t)$
Integrator	$\dot{y}(t) = u(t)$	$\frac{1}{s}$	$E(t)$ (step)
Differentiator	$y(t) = \dot{u}(t)$	s	$\dot{\delta}(t)$
Delay	$y(t) = u(t-\tau)$	$e^{-\tau s}$	$\delta(t-\tau)$

Table 4.1: Transfer function and impulse response of some elementary systems

Remark 1. The transfer function of the delay comes from the relation

$$y(t) = \sum_{i=0}^{\infty} \frac{y^{(i)}(t_0)}{i!}(t-t_0)^i$$

If we take $\tau = t_0 - t$, the relation translates into

$$y(t) = \sum_{i=0}^{\infty} \frac{y^{(i)}(\tau+t)}{i!}(-\tau)^i$$

Since $y(t) = u(t-\tau)$, or equivalently $y(t+\tau) = u(t)$, we get

$$\begin{array}{rclcl} y(t) & = & \sum_{i=0}^{\infty} \frac{u^{(i)}(t)}{i!}(-\tau)^i & = & \sum_{i=0}^{\infty} \frac{(-\tau)^i}{i!}\frac{d^i}{dt^i}u(t) \\ & = & \left(\sum_{i=0}^{\infty} \frac{1}{i!}\left(-\tau \cdot \frac{d}{dt}\right)^i\right) \cdot u(t) & = & e^{-\tau \cdot \frac{d}{dt}} \cdot u(t) \\ & = & e^{-\tau s} \cdot u(t) & & \end{array}$$

But we can go further, as is shown on the following table.

Equation	Transfer function	Impulse response
$y(t) = \alpha_1 u(t-\tau_1) + \alpha_2 u(t-\tau_2)$	$\alpha_1 e^{-\tau_1 s} + \alpha_2 e^{-\tau_2 s}$	$\alpha_1 \delta(t-\tau_1) + \alpha_2 \delta(t-\tau_2)$
$y(t) = \sum_{i=0}^{\infty} \alpha_i u(t-\tau_i)$	$\sum_{i=0}^{\infty} \alpha_i e^{-\tau_i s}$	$\sum_{i=0}^{\infty} \alpha_i \delta(t-\tau_i)$
$y(t) = \int_{-\infty}^{\infty} f(\tau) u(t-\tau)\, d\tau$	$\int_{-\infty}^{\infty} f(\tau) e^{-\tau s} d\tau$	$\int_{-\infty}^{\infty} f(\tau)\delta(t-\tau) d\tau$

Table 4.2: Transfer function and impulse response of composed systems

The operation $y(t) = \int_{-\infty}^{\infty} f(\tau) u(t-\tau)$ is called *convolution*. We may notice that the impulse response of the system described by the last row of the table is:

$$y(t) = \left.\int_{-\infty}^{\infty} f(\tau) u(t-\tau)\, d\tau\right|_{u(t)=\delta(t)} = \int_{-\infty}^{\infty} f(\tau)\delta(t-\tau)\, d\tau = f(t)$$

Therefore, the Laplace transform of a function $f(t)$ is:

$$\hat{f}(s) = \int_{-\infty}^{\infty} f(\tau) e^{-\tau s} d\tau$$

Let us note that the relation:

$$f(t) = \int_{-\infty}^{\infty} f(\tau)\delta(t-\tau) d\tau$$

illustrates the fact that $f(t)$ can be estimated by the sum of an infinity of infinitely approached Dirac distributions.

4.2.4 Input-output relation

Let us consider a system with input u, output y and transfer function $H(s)$, as in Figure 4.1.

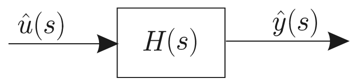


Figure 4.1: Laplace transform and transfer function

We have:

$$y(t) = H(\frac{d}{dt}).u(t) = \underbrace{H(\frac{d}{dt}).\hat{u}\left(\frac{d}{dt}\right)}_{\hat{y}(\frac{d}{dt})} \cdot \delta(t)$$

Therefore, the Laplace transform of $y(t)$ is:

$$\hat{y}(s) = H(s) \cdot \hat{u}(s)$$

4.3 Relationship between state and transfer representations

Let us consider the system described by its state equations:

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du} \end{cases}$$

The Laplace transform of the state representation is given by:

$$\begin{cases} s\hat{\mathbf{x}} &= \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\hat{\mathbf{u}} \\ \hat{\mathbf{y}} &= \mathbf{C}\hat{\mathbf{x}} + \mathbf{D}\hat{\mathbf{u}} \end{cases}$$

The first equation can be re-written as $s\hat{\mathbf{x}} - \mathbf{A}\hat{\mathbf{x}} = \mathbf{B}\hat{\mathbf{u}}$, i.e. $s\mathbf{I}\hat{\mathbf{x}} - \mathbf{A}\hat{\mathbf{x}} = \mathbf{B}\hat{\mathbf{u}}$, where $\mathbf{I}$ is the identity matrix. From this, by factoring, we get $(s\mathbf{I} - \mathbf{A})\,\hat{\mathbf{x}} = \mathbf{B}\hat{\mathbf{u}}$ (we must be careful, a notation such as $s\hat{\mathbf{x}} - \mathbf{A}\hat{\mathbf{x}} = (s - \mathbf{A})\,\hat{\mathbf{x}}$ is not permitted since s is a scalar whereas $\mathbf{A}$ is a matrix). Therefore:

$$\begin{cases} \hat{\mathbf{x}} &= (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\hat{\mathbf{u}} \\ \hat{\mathbf{y}} &= \mathbf{C}\hat{\mathbf{x}} + \mathbf{D}\hat{\mathbf{u}} \end{cases}$$

And thus:

$$\hat{\mathbf{y}} = \left(\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}\right)\hat{\mathbf{u}}$$

The matrix:

$$\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$$

is called *transfer matrix*. It is a matrix of *transfer functions* (in other words of rational functions in s) in which every denominator is a divisor of the characteristic polynomial $P_{\mathbf{A}}(s)$ of $\mathbf{A}$. By multiplying each side by $P_{\mathbf{A}}(s)$ and replacing s by $\frac{d}{dt}$ we obtain a system of input-output differential equations. The state $\mathbf{x}$ will no longer appear there.

Reciprocally, in the case of *monovariate* linear systems (in other words with a single input and a single output), we can obtain a state representation from an expression of the transfer function, as it will be illustrated in Exercises 4.12, 4.13 and 4.14.

Exercises

Exercise 4.1.– *Solution of a state equation*

See the correction video at https://youtu.be/l0accPUnKZw

1) Show that the continuous-time linear system

$$\dot{\mathbf{x}}(t) \quad = \quad \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$$

has the solution

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau.$$

The function $e^{\mathbf{A}t}\mathbf{x}(0)$ is called *homogeneous*, *free* or *transient solution*. The function $\int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau$ is called *forced solution*.

2) Show that the discrete-time linear system

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)$$

has the solution

$$\mathbf{x}(k) = \mathbf{A}^k\mathbf{x}(0) + \sum_{\ell=0}^{k-1} \mathbf{A}^{k-\ell-1}\mathbf{B}\mathbf{u}(\ell).$$

Again, $\mathbf{A}^k\mathbf{x}(0)$ is *homogeneous solution* and $\sum_{\ell=0}^{k} \mathbf{A}^{k-\ell}\mathbf{B}\mathbf{u}(\ell)$ is the *forced solution*.

Exercise 4.2.– *Stability criterion*

See the correction video at https://youtu.be/PXO2ojcMpt0

The proof of the stability criterion for linear systems is based on the theorem of correspondence of the eigenvalues which is articulated as follows. If f is a polynomial (or more generally an integer series) and if $\mathbf{A}$ is an $\mathbb{R}^{n\times n}$ matrix then the eigenvectors of $\mathbf{A}$ are also eigenvectors of $f(\mathbf{A})$. Moreover if the eigenvalues of $\mathbf{A}$ are $\{\lambda_1, \dots, \lambda_n\}$ then those of $f(\mathbf{A})$ are $\{f(\lambda_1), \dots, f(\lambda_n)\}$.

1) Prove the theorem of correspondence of the eigenvalues in the case where f is a polynomial.

2) Let $\text{spec}(\mathbf{A}) = \{\lambda_1, \dots, \lambda_n\}$ be the spectrum of $\mathbf{A}$, i.e. its eigenvalues. By using the theorem of correspondence of the eigenvalues, calculate $\text{spec}(\mathbf{I}+\mathbf{A})$, $\text{spec}\left(\mathbf{A}^k\right)$, $\text{spec}\left(e^{\mathbf{A}t}\right)$ and $\text{spec}(f(\mathbf{A}))$.

In these expressions, $\mathbf{I}$ denotes the identity matrix, k an integer and f the characteristic polynomial of $\mathbf{A}$.

3) Show that if a continuous-time linear system is stable then all the eigenvalues of its evolution matrix $\mathbf{A}$ have strictly negative real parts (in fact, the condition is necessary and sufficient, but we will limit the proof to the implication).

4) Show that if a discrete-time linear system is stable then all the eigenvalues of $\mathbf{A}$ are strictly within the unit circle. Again, even though we have the equivalence here, we will limit ourselves to the proof of the implication.

Exercise 4.3.– *Laplace variable*

See the correction video at https://youtu.be/SRoMKIjl90g

Let us consider two systems in parallel as illustrated in Figure 4.2.

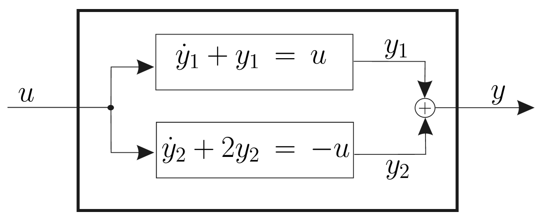


Figure 4.2: Two linear systems interconnected in parallel

1) Without employing the Laplace transform, and by only using elementary differential calculus, give a differential equation which links u with y.

2) Using algebraic manipulations involving the $\frac{d}{dt}$ operator, obtain the same differential equation again.

3) Obtain the same result by using the Laplace variable s.

Exercise 4.4.– *Transfer functions of elementary systems*

See the correction video at https://youtu.be/ibOfo-PixlY

Give the transfer function of the following systems with input u and output y:

1) a differentiator expressed by the differential equation $y = \dot{u}$;

2) an integrator which obeys the differential equation $\dot{y} = u$;

3) a delay of τ which is expressed by the input-output relation $y(t) = u(t - \tau)$.

Exercise 4.5.– *Transfer function of composite systems*

See the correction video at https://youtu.be/_L4wGYU1xhg

1) Let us consider two systems of transfer functions $H_1(s)$ and $H_2(s)$ placed in series as shown on Figure 4.3.

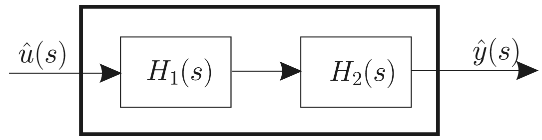


Figure 4.3: Two systems in series

Calculate, in function of $H_1(s)$ and $H_2(s)$, the transfer function of the composite system.

2) Let us place the two systems of transfer functions $H_1(s)$ and $H_2(s)$ in parallel as shown on Figure 4.4.

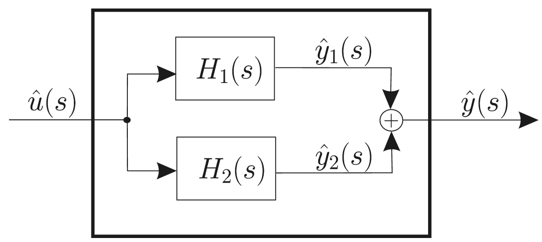


Figure 4.4: Two systems in parallel

Give, in function of $H_1(s)$ and $H_2(s)$, the transfer function of the composite system.

3) Let us loop the system $H(s)$ as shown on Figure 4.5.

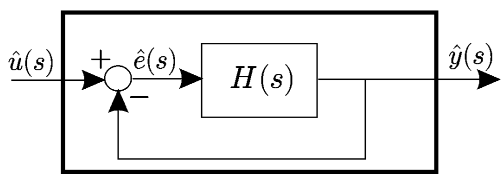


Figure 4.5: Looped system

Give, in function of $H(s)$, the transfer function of the looped system.

Exercise 4.6.– *Transfer matrix*

See the correction video at https://youtu.be/O7U57PztbGk

Let us consider the continuous-time linear system described by its state representation:

$$\begin{cases} \dot{\mathbf{x}}(t) &= \begin{pmatrix} 1 & 3 \\ 2 & 0 \end{pmatrix} \mathbf{x}(t) + \begin{pmatrix} 1 \\ 1 \end{pmatrix} u(t) \\ \mathbf{y}(t) &= \begin{pmatrix} 1 & 2 \\ 1 & 0 \end{pmatrix} \mathbf{x}(t) + \begin{pmatrix} 2 \\ 0 \end{pmatrix} u(t) \end{cases}$$

1) Calculate its transfer matrix.

2) Give a differential relation which links the input to the outputs.

Exercise 4.7.– *Change of basis*

See the correction video at https://youtu.be/c6j2TFPoiVw

Let us consider the continuous-time linear system described by its state equations:

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

Let us take the change of basis $\mathbf{v} = \mathbf{P}^{-1}\mathbf{x}$, where $\mathbf{P}$ is a transfer matrix (i.e., square and invertible).

1) What do the system equations become if $\mathbf{v}$ is the new state vector ?

2) Let us consider the system described by the following state equations:

$$\begin{cases} \dot{\mathbf{x}} &= \begin{pmatrix} 4 & -\frac{1}{2} & -\frac{1}{2} \\ 4 & 1 & -1 \\ 4 & -2 & 2 \end{pmatrix} \mathbf{x} + \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix} u \\ y &= \begin{pmatrix} 2 & 1 & 1 \end{pmatrix} \mathbf{x} \end{cases}$$

We are trying to find a simpler representation, i.e. with more zeros and ones (in order for instance to limit the number of components necessary in the design of the circuitry). We propose to take the following change of basis:

$$\mathbf{v} = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 2 \\ 2 & 1 & 0 \end{pmatrix}^{-1} \mathbf{x}$$

which brings us to a Jordan normal form. What does the new state representation become ?

3) Calculate the transfer function of this system. What is the characteristic polynomial of the system ?

Exercise 4.8.– *Controllable canonical form*

See the correction video at https://youtu.be/pZm7yAA4Npk

Let us consider a system with the input described by the evolution equation:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u$$

Let us note that the control matrix traditionally denoted by $\mathbf{B}$ becomes, in our case of a single input, a vector $\mathbf{b}$. Let us take as transfer matrix (assumed invertible):

$$\mathbf{P} = \left(\mathbf{b} \mid \mathbf{A}\mathbf{b} \mid \mathbf{A}^2\mathbf{b} \mid \dots \mid \mathbf{A}^{n-1}\mathbf{b}\right)$$

The new state vector is therefore $\mathbf{v} = \mathbf{P}^{-1}\mathbf{x}$. Show that, in this new basis, the state equations are written as:

$$\dot{\mathbf{v}} = \begin{pmatrix} 0 & 0 & 0 & -a_0 \\ 1 & 0 & 0 & -a_1 \\ 0 & \vdots & 0 & \dots \\ 0 & 0 & 1 & -a_{n-1} \end{pmatrix} \mathbf{v} + \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \mathbf{u}$$

where the a_i are the coefficients of the characteristic polynomial of the matrix $\mathbf{A}$.

EXERCISE 4.9.– *State equations of a second order system*

See the correction video at https://youtu.be/ZV-4gpKGaVY

Let us consider the system $\mathcal{S}$ described by the block diagram of Figure 4.6.

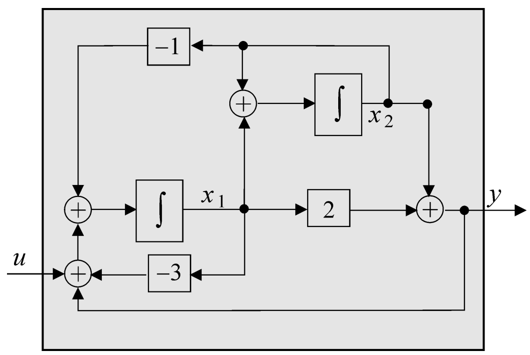


Figure 4.6: Second-order system

1) Give its state equations in matrix form.

2) Calculate the characteristic polynomial of the system. Is the system stable ?

3) Calculate the transfer function of the system.

EXERCISE 4.10.– *Combination of systems*

See the correction video at https://youtu.be/bzmd0Ue2DK8

Let us consider the systems $\mathcal{S}_1$ and $\mathcal{S}_2$ given on top of Figure 4.7.

1) Give the state equations in matrix form of the system $\mathcal{S}_a$ obtained by placing systems $\mathcal{S}_1$ and $\mathcal{S}_2$ in series. Give the transfer function and characteristic polynomial of $\mathcal{S}_a$.

2) Do the same with the system $\mathcal{S}_b$ obtained by placing systems $\mathcal{S}_1$ and $\mathcal{S}_2$ in parallel.

3) Do the same with the system $\mathcal{S}_c$ obtained by looping $\mathcal{S}_1$ by $\mathcal{S}_2$ as represented on Figure 4.7.

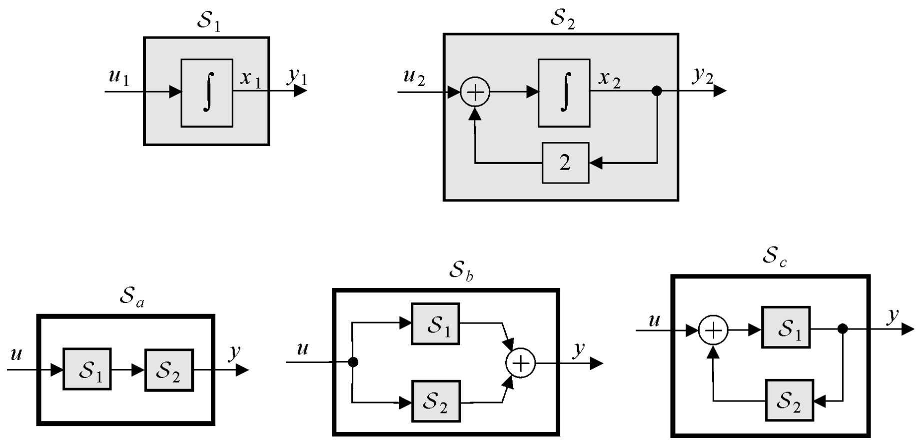


Figure 4.7: Composition of systems

EXERCISE 4.11.– *Transfer function*

See the correction video at https://youtu.be/4sA1C9vXzJQ

Let us consider the system represented by Figure 4.8. Calculate its transfer function.

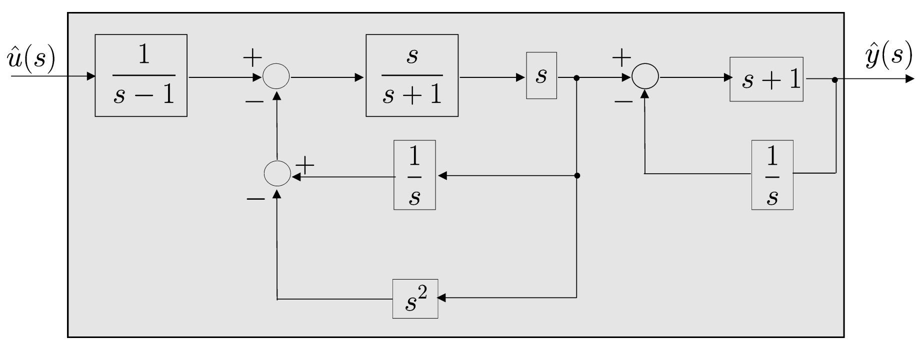


Figure 4.8: Linear system for which the transfer function must be calculated

---

EXERCISE 4.12.– *Canonical control form*

See the correction video at https://youtu.be/3vI-NVxx58w

Let us consider the order 3 linear system with a single input and a single output, described by the following differential equation:

$$\dddot{y} + a_2\ddot{y} + a_1\dot{y} + a_0 y = b_2\ddot{u} + b_1\dot{u} + b_0 u$$

1) Calculate its transfer function $G(s)$.

2) By noticing that this system can be obtained by placing the two transfer function systems in series:

$$G_1(s) = \frac{1}{s^3 + a_2 s^2 + a_1 s + a_0} \quad \text{and} \quad G_2(s) = b_2 s^2 + b_1 s + b_0$$

deduce a block diagram with only three integrators, some adders and amplifiers.

3) Give the state equations associated with this block diagram.

EXERCISE 4.13.– *Canonical observation form*

See the correction video at https://youtu.be/ZNyAHIN9PrQ

Let us consider once more the system described by the differential equation:

$$\dddot{y} + a_2\ddot{y} + a_1\dot{y} + a_0 y = b_2\ddot{u} + b_1\dot{u} + b_0 u$$

1) Show that this differential equation can be written in integral form as:

$$y = \int\int\int (b_2\ddot{u} - a_2\ddot{y} \ + \ b_1\dot{u} - a_1\dot{y} \ + \ b_0 u - a_0 y)$$

2) Deduce from this a block diagram with only three integrators, some adders and amplifiers.

3) Give the associated state equations.

4) Compare this with the results obtained in Exercise 4.12.

EXERCISE 4.14.– *Modal form*

See the correction video at https://youtu.be/Y9tlXWQrAcQ

A monovariate linear system is in *modal* form if it can be written as:

$$\begin{cases} \dot{\mathbf{x}} &= \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & \lambda_n \end{pmatrix} \mathbf{x} + \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} u \\ y &= \begin{pmatrix} c_1 & c_2 & \cdots & c_n \end{pmatrix} \mathbf{x} \ + \ d \ u \end{cases}$$

1) Draw a block diagram for this system using integrators, adders and amplifiers.

2) Calculate the transfer function associated with this system.

3) Calculate its characteristic polynomial.

EXERCISE 4.15.– *Jordan normal form*

See the correction video at https://youtu.be/79UbImbjs9Q

The system described by the state equations:

$$\dot{\mathbf{x}} = \begin{pmatrix} -2 & 1 & 0 & 0 & 0 \\ 0 & -2 & 1 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & -3 & 1 \\ 0 & 0 & 0 & 0 & -3 \end{pmatrix} \mathbf{x} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} u$$
$$y = \begin{pmatrix} -2 & -1 & 3 & -4 & 7 \end{pmatrix} \mathbf{x} + 2u$$

is in Jordan normal form since its evolution matrix $\mathbf{A}$ is a Jordan matrix. In other words, it is block diagonal and each block (here there are two) has zeros everywhere, except on the diagonal which contains equal elements and on the opposite diagonal which only contains ones. Moreover, the control matrix only contains ones and zeros which are positioned at the last row of each block.

1) Draw a block diagram for this system using integrators, adders and amplifiers.

2) Calculate its transfer function.

3) Calculate its characteristic polynomial as well as the associated eigenvalues.

Chapter 5

Linear control

In this chapter, we will study the design of controllers for systems given by linear state equations of the form:

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} \end{cases}$$

We will show in the following chapter that around very particular points of the state space, called *operating points*, many non-linear systems genuinely behave like linear systems. The techniques developed in this chapter will then be used for the control of non-linear systems. Let us denote by m, n, p the respective dimensions of the vectors $\mathbf{u}$, $\mathbf{x}$ and $\mathbf{y}$. Recall that $\mathbf{A}$ is called *evolution matrix*, $\mathbf{B}$ is the *control matrix* and $\mathbf{C}$ is the *observation matrix*.

Remark 2. We have assumed here, in the interest of simplification, that the direct matrix $\mathbf{D}$ involved in the observation equation was nil. In the case where such a direct matrix exists, we can remove it with a simple feed-forward loop $\mathbf{z} = \mathbf{y} - \mathbf{Du}$, as represented on Figure 5.1.

After having defined the fundamental concepts of controllability and observability, we will propose two approaches for the design of controllers. First of all, we will assume that the state $\mathbf{x}$ is accessible on demand. Even though this hypothesis is generally not verified, it will allow us to establish the principles of the *pole placement method*. In the second phase, we will no longer assume that the state is accessible. We will then have to develop *state estimators* capable of approximating the state vector in order to be able to employ the tools developed in the first phase. The reader should consult the book of T. Kailath [13]in order to have a wider view of the methods used for the control of linear systems.

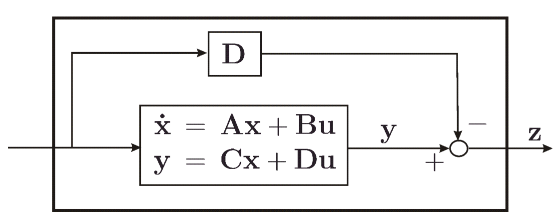


Figure 5.1: Loop allowing the removal of the direct matrix $\mathbf{D}$

5.1 Controllability and observability

There are multiple equivalent definitions for the controllability and observability of linear systems. A simple definition is the following.

Definition 3. The linear system

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} \end{cases}$$

is said to be *controllable* if, for every pair of state vectors $(\mathbf{x}_0, \mathbf{x}_1)$, we can find a time t_1 and a control $\mathbf{u}(t)$, $t \in [0, t_1]$, such that the system, initialized in $\mathbf{x}_0$, reaches the state $\mathbf{x}_1$, at time t_1. It is *observable* is the knowledge of $\mathbf{y}(t)$ and of $\mathbf{u}(t)$ for $t \in \mathbb{R}$ allows us to determine, in a unique manner, the state $\mathbf{x}(t)$, for all t.

Controllability criterion – The system is controllable if and only if:

$$\text{rank}\underbrace{\left(\mathbf{B} \mid \mathbf{AB} \mid \mathbf{A}^2\mathbf{B} \mid \ldots \mid \mathbf{A}^{n-1}\mathbf{B}\right)}_{\mathbf{\Gamma}_{\text{con}}} = n$$

where n is the dimension of $\mathbf{x}$. In other words, the matrix $\mathbf{\Gamma}_{\text{con}}$, called the *controllability matrix*, obtained by juxtaposing the n matrices $\mathbf{B}$, $\mathbf{AB}$, ..., $\mathbf{A}^{n-1}\mathbf{B}$ next to one another, has to be of full rank in order for the system to be controllable. This criterion is proved in Exercise 5.4.

Observability criterion – The linear system is *observable* if:

$$\text{rank}\underbrace{\begin{pmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{pmatrix}}_{\mathbf{\Gamma}_{\text{obs}}} = n$$

in other words the matrix, called the *observability matrix* $\mathbf{\Gamma}_{\text{obs}}$, obtained by placing the n matrices $\mathbf{C}$, $\mathbf{CA}$, ..., $\mathbf{CA}^{n-1}$ below one another, is of full rank. The proof of this criterion is discussed in Exercise 5.5.

The above definitions as well as the criteria are also valid for discrete-time linear systems (see Exercise 5.3).

5.2 State feedback control

Let us consider the system $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$ which is assumed to be controllable and we are looking to find a controller for this system of the form $\mathbf{u} = \mathbf{w} - \mathbf{Kx}$, where $\mathbf{w}$ is the new input. This leads to the assumption that $\mathbf{x}$ is accessible on demand, which is normally not the case. We will see further on how to get rid of this awkward hypothesis. The state equations of the looped system are written as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}(\mathbf{w} - \mathbf{K}\mathbf{x}) = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x} + \mathbf{B}\mathbf{w}.$$

It is reasonable to choose the control matrix $\mathbf{K}$ so as to impose the poles of the looped system. This problem is equivalent to imposing the characteristic polynomial of the system. Let $P_{\text{con}}(s)$ be the desired polynomial of degree n. We need to solve the polynomial equation:

$$\det(s\mathbf{I} - \mathbf{A} + \mathbf{B}\mathbf{K}) = P_{\text{con}}(s)$$

referred to as *pole placement*. This equation can be translated into n scalar equations. Let us indeed recall that two monic polynomials of degree n $s^n + a_{n-1}s + \cdots + a_0$ and $s^n + b_{n-1}s + \cdots + b_0$ are equal if and only if their coefficients are all equal, i.e. if $a_{n-1} = b_{n-1}, \dots, a_0 = b_0$. Our system of n equations has $m.n$ unknowns which are the coefficients k_{ij}, $i \in \{1, \dots, m\}, j \in \{1, \dots, n\}$. In fact, a single solution matrix $\mathbf{K}$ is sufficient. We may therefore fix $(m-1)$ elements of $\mathbf{K}$ so that we are left with only n unknowns. However, the obtained system is not always linear. The `ppol` instruction in SCILAB or `place` in MATLAB or PYTHON allow to solve the pole placement equation.

5.3 Output feedback control

In this paragraph, we are once again looking to stabilize the system:

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

but this time, the state $\mathbf{x}$ of the system is is no longer assumed to be accessible on demand. The controller which will be used is represented on Figure 5.2.

Only the setpoint $\mathbf{w}$ and the output of the system $\mathbf{y}$ can be used by the controller. The unknowns of the controller are the matrices $\mathbf{K}$, $\mathbf{L}$ and $\mathbf{H}$. Let us attempt to explain the structure of this controller. First of all, in order to estimate the state $\mathbf{x}$ necessary for computing the control $\mathbf{u}$, we integrate a *simulator* of our system into the controller. The simulator is a copy of the system and its state vector is denoted by $\hat{\mathbf{x}}$. The error ε_y between the output of the simulator $\hat{\mathbf{y}}$ and the output of the system $\mathbf{y}$ allows us to correct, by using a correction matrix $\mathbf{L}$, the evolution of the estimated state $\hat{\mathbf{x}}$. The corrected simulator is referred to as the *Luenberger observer*. We can then apply a state feedback technique which will be carried out using the matrix $\mathbf{K}$. For the calculation of $\mathbf{K}$ we may use the pole placement method described earlier, which consists of solving $\det(s\mathbf{I} - \mathbf{A} + \mathbf{B}\mathbf{K}) = P_{\text{con}}(s)$, where $P_{\text{con}}(s)$ is the characteristic polynomial of degree n chosen for the control dynamics. For the calculation of the correction matrix $\mathbf{L}$, we will try to guarantee a convergence of the error $\hat{\mathbf{x}} - \mathbf{x}$ to $\mathbf{0}$. The role of the matrix $\mathbf{H}$ (square matrix placed right after the setpoint vector $\mathbf{w}$), called a *precompensator* is to match the components of the setpoint $\mathbf{w}$ with certain state variables chosen beforehand. We will also discuss how to choose this matrix $\mathbf{H}$ in order to be able to perform this association.

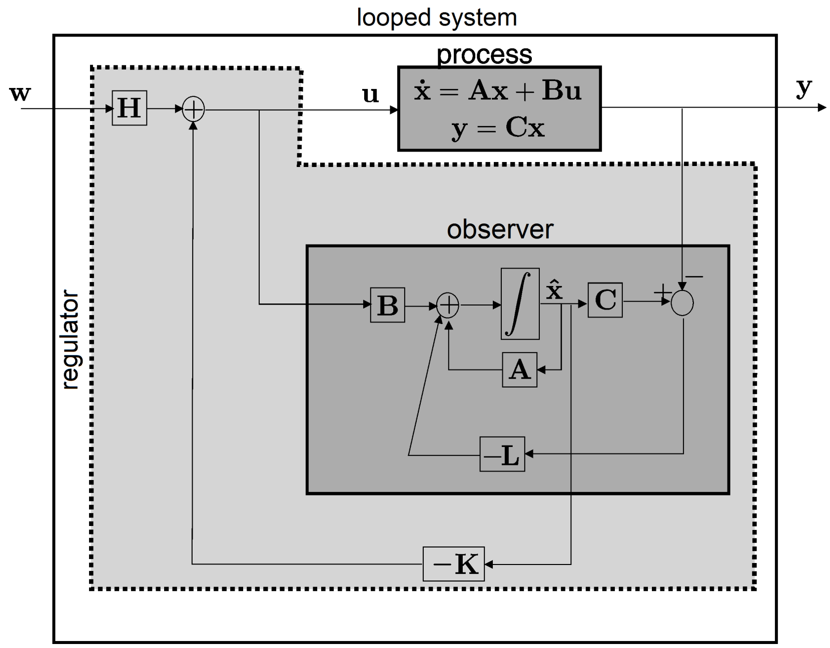


Figure 5.2: Principle of an output feedback controller

In order to calculate $\mathbf{L}$, let us extract, from the controlled system of the figure, the sub-system with input $\mathbf{u}$ formed from the system to be controlled and its observer (see Figure 5.3).

The state equations which describe the system are:

$$\begin{cases} \dot{\mathbf{x}} & = & \mathbf{Ax} + \mathbf{Bu} \\ \frac{d}{dt}\hat{\mathbf{x}} & = & \mathbf{A}\hat{\mathbf{x}} + \mathbf{Bu} - \mathbf{L}\left(\mathbf{C}\hat{\mathbf{x}} - \mathbf{Cx}\right) \end{cases}$$

where the state vector is $(\mathbf{x}, \hat{\mathbf{x}})$. Let us create, only by thought, the quantity $\varepsilon_x = \hat{\mathbf{x}} - \mathbf{x}$, as represented on Figure 5.3. By subtracting the two evolution equations, we obtain:

$$\frac{d}{dt}\left(\hat{\mathbf{x}} - \mathbf{x}\right) = \mathbf{A}\hat{\mathbf{x}} + \mathbf{Bu} - \mathbf{L}\left(\mathbf{C}\hat{\mathbf{x}} - \mathbf{Cx}\right) - \mathbf{Ax} - \mathbf{Bu} = \mathbf{A}\left(\hat{\mathbf{x}} - \mathbf{x}\right) - \mathbf{LC}\left(\hat{\mathbf{x}} - \mathbf{x}\right)$$

Thus, $\boldsymbol{\varepsilon}_x$ obeys the differential equation:

$$\dot{\boldsymbol{\varepsilon}}_x = \left(\mathbf{A} - \mathbf{LC}\right)\boldsymbol{\varepsilon}_x$$

in which the control $\mathbf{u}$ is not involved. The estimation error $\boldsymbol{\varepsilon}_x$ on the state tends towards zero if all the eigenvalues of $\mathbf{A} - \mathbf{LC}$ have negative real parts. To impose the dynamics of the error (in other words, its convergence speed) means solving:

$$\det\left(s\mathbf{I} - \mathbf{A} + \mathbf{LC}\right) = P_{\text{obs}}(s)$$

where $P_{\text{obs}}(s)$ is chosen as desired, so as to have the required poles. Since the determinant of a matrix is equal to that of its transpose, this equation is equivalent to:

$$\det\left(s\mathbf{I} - \mathbf{A}^{\mathrm{T}} + \mathbf{C}^{\mathrm{T}}\mathbf{L}^{\mathrm{T}}\right) = P_{\text{obs}}(s)$$

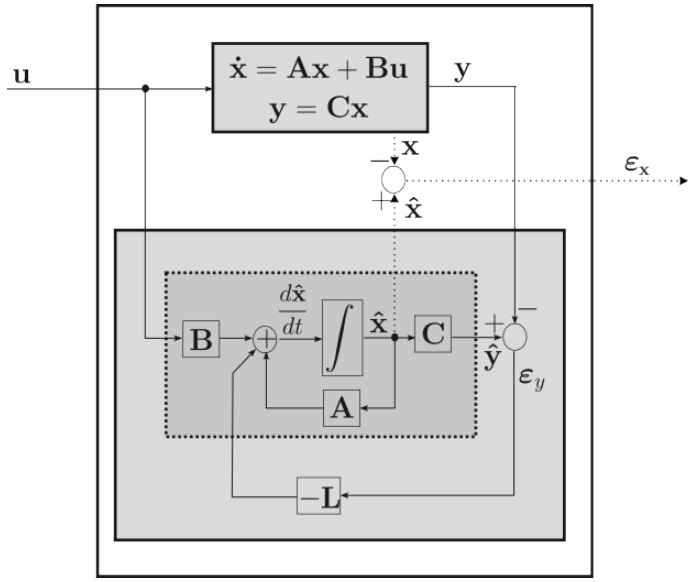


Figure 5.3: Diagram of the output error between the observer and the system

We obtain a pole placement-type equation.

PRECOMPENSATOR.– The precompensator $\mathbf{H}$ allows to associate the setpoints (components of $\mathbf{w}$) with certain values of the state variables that we are looking at. The choice of these state variables is done by using a setpoint matrix $\mathbf{E}$. Assuming that $\mathbf{w}(t) = \bar{\mathbf{w}}$ is constant, at equilibrium we have, $\hat{\mathbf{x}} = \mathbf{x} = \bar{\mathbf{x}}$ and $\dot{\mathbf{x}} = \mathbf{0}$. Thus

$$(\mathbf{A} - \mathbf{BK})\,\bar{\mathbf{x}} + \mathbf{BH}\bar{\mathbf{w}} = \mathbf{0}$$

or equivalently $\bar{\mathbf{x}} = -(\mathbf{A} - \mathbf{BK})^{-1}\mathbf{BH}\bar{\mathbf{w}}$. If we want $\bar{\mathbf{w}} = \mathbf{E}\bar{\mathbf{x}}$, we should solve

$$-\mathbf{E}\,(\mathbf{A} - \mathbf{BK})^{-1}\,\mathbf{BH} = \mathbf{I}.$$

Therefore

$$\mathbf{H} = -\left(\mathbf{E}\,(\mathbf{A} - \mathbf{BK})^{-1}\,\mathbf{B}\right)^{-1}.$$

5.4 Summary

The algorithm of the table below summarizes the method which allows to calculate an output feedback controller with precompensator.

Algorithm REGULKLH(**in:** $\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{E},\mathbf{p_{con}},\mathbf{p_{obs}}$; **out :** $\mathcal{R}$)

$\mathbf{K} :=$ PLACE $(\mathbf{A},\mathbf{B},\mathbf{p}_{\mathrm{con}})$;

$\mathbf{L} :=$ PLACE $\left(\mathbf{A}^{\mathrm{T}},\mathbf{C}^{\mathrm{T}},\mathbf{p}_{\mathrm{obs}}\right)^{\mathrm{T}}$;

$\mathbf{H} := -\left(\mathbf{E}\left(\mathbf{A}-\mathbf{B}\mathbf{K}\right)^{-1}\mathbf{B}\right)^{-1}$;

$$\mathcal{R} := \left\{ \begin{array}{ccccc} \frac{d}{dt}\hat{\mathbf{x}} & = & (\mathbf{A}-\mathbf{B}\mathbf{K}-\mathbf{L}\mathbf{C}) & \hat{\mathbf{x}} & + \begin{pmatrix} \mathbf{B}\mathbf{H} & \mathbf{L} \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ \mathbf{y} \end{pmatrix} \\ \mathbf{u} & = & -\mathbf{K} & \hat{\mathbf{x}} & + \begin{pmatrix} \mathbf{H} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ \mathbf{y} \end{pmatrix} \end{array} \right.$$

Exercises

EXERCISE 5.1.– *Non-observable states, non-controllable states*

See the correction video at https://youtu.be/Vdy3PmyV9kk

Let us consider the system:

$$\left\{\begin{array}{lcl} \dot{\mathbf{x}}(t) & = & \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} \mathbf{x}(t) + \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} u(t) \\ y(t) & = & \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix} \mathbf{x}(t) + 1 \, u(t) \end{array}\right.$$

1) Calculate its transfer function.

2) Is this system stable ?

3) Draw the associated block diagram and deduce from this the non-observable and non-controllable states.

4) The poles of the system (the eigenvalues of the evolution matrix) are composed of transmission poles (poles of the transfer function) and hidden poles. Give the hidden poles.

EXERCISE 5.2.– *Using the controllability and observability criteria*

See the correction video at https://youtu.be/n-5sA7HGH1s

Let us consider the system

$$\left\{\begin{array}{lcl} \dot{\mathbf{x}} & = & \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{x} + \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & a \end{pmatrix} \mathbf{u} \\ \mathbf{y} & = & \begin{pmatrix} 1 & 1 & b \\ 0 & 1 & 0 \end{pmatrix} \mathbf{x} \end{array}\right.$$

1) For which values of a is the system controllable ?

2) For which values of b is the system observable ?

3) Draw the block diagram of the system.

EXERCISE 5.3.– *Proof of the controllability criterion in discrete time*

See the correction video at https://youtu.be/-B3nfyBQ1U8

Prove the criterion of controllability is discrete time. This theorem states that the system $\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)$ is controllable if and only if:

$$\text{rank}\underbrace{\left(\mathbf{B} \mid \mathbf{AB} \mid \mathbf{A}^2\mathbf{B}| \dots | \mathbf{A}^{n-1}\mathbf{B}\right)}_{\mathbf{\Gamma}_{\text{con}}} = n$$

where n is the dimension of $\mathbf{x}$.

1) Show that:

$$\mathbf{x}(n) = \mathbf{A}^n\mathbf{x}(0) + \left(\mathbf{B} \mid \mathbf{AB} \mid \dots \mid \mathbf{A}^{n-1}\mathbf{B}\right)\begin{pmatrix} \mathbf{u}(n-1) \\ \vdots \\ \mathbf{u}(1) \\ \mathbf{u}(0) \end{pmatrix}$$

2) Show that if the matrix $(\mathbf{B} \mid \mathbf{AB} \mid \dots \mid \mathbf{A}^{n-1}\mathbf{B})$ is of full rank then, for every initial vector $\mathbf{x}_0$, for every target vector $\mathbf{x}_n$, we can find a control $\mathbf{u}(0), \mathbf{u}(1), \dots, \mathbf{u}(n-1)$ such that the system initialized in $\mathbf{x}_0$ reaches $\mathbf{x}_n$, at time $k = n$.

EXERCISE 5.4.– *Proof of the controllability criterion in continuous time*

See the correction video at https://youtu.be/4TWwM0HyHDw

In this exercise, we prove the controllability criterion in continuous time which states that the system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ is controllable if and only if:

$$\text{rank}\underbrace{\left(\mathbf{B} \mid \mathbf{AB} \mid \mathbf{A}^2\mathbf{B} \mid \dots \mid \mathbf{A}^{n-1}\mathbf{B}\right)}_{\mathbf{\Gamma}_{\text{con}}} = n$$

where n is the dimension of $\mathbf{x}$ and Γ_{con} is the controllability matrix.

The proof is in two steps. First, we prove in Question 1 by contradiction that if the system is controllable then the rank is equal to n. Then, we show in Question 2 that if the rank is equal to n then the system is controllable.

1) Assume that:

$$\text{rank}\underbrace{\left(\mathbf{B} \mid \mathbf{AB} \mid \mathbf{A}^2\mathbf{B} \mid \dots \mid \mathbf{A}^{n-1}\mathbf{B}\right)}_{\mathbf{\Gamma}_{\text{con}}} < n$$

Let $\mathbf{z}$ be a non-zero vector such that $\mathbf{z}^{\mathbf{T}} \cdot \mathbf{\Gamma}_{\mathbf{con}} = \mathbf{0}$. By using the solution of the state equation:

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau,$$

show that the control $\mathbf{u}$ cannot influence the value $\mathbf{z}^{\mathrm{T}}\mathbf{x}$. Deduce from this that the system is not controllable.

2) Show that if $\mathrm{rank}(\mathbf{\Gamma}_{\mathrm{con}}) = n$, for every pair $(\mathbf{x}(0), \mathbf{x}(t_1))$, $t_1 > 0$, there is a polynomial control $\mathbf{u}(t), t \in [0, t_1]$ which leads the system from state $\mathbf{x}(0)$ to state $\mathbf{x}(t_1)$. Here, we will limit ourselves to $t_1 = 1$, knowing that the main principle of the proof remains valid for any $t_1 > 0$.

EXERCISE 5.5.– *Proof of the observability criterion*

See the correction video at https://youtu.be/Y6TBDrMf8Eo

Consider the continuous-time linear system:

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

1) Show that:

$$\begin{pmatrix} \mathbf{y} \\ \dot{\mathbf{y}} \\ \ddot{\mathbf{y}} \\ \vdots \\ \mathbf{y}^{(n-1)} \end{pmatrix} = \begin{pmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{pmatrix} \mathbf{x} + \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{CB} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{CAB} & \mathbf{CB} & \mathbf{0} & \\ \vdots & & \ddots & \ddots \\ \mathbf{CA}^{n-2}\mathbf{B} & \cdots & \mathbf{CAB} & \mathbf{CB} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \dot{\mathbf{u}} \\ \ddot{\mathbf{u}} \\ \vdots \\ \mathbf{u}^{(n-2)} \end{pmatrix}$$

2) Deduce from this that if the observability matrix:

$$\mathbf{\Gamma}_{\mathrm{obs}} = \begin{pmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{pmatrix}$$

is of full rank, then we can express the state $\mathbf{x}$ as a linear function of the quantities $\mathbf{u}$, $\mathbf{y}$, $\dot{\mathbf{u}}, \dot{\mathbf{y}}$, $\dots, \mathbf{u}^{(n-2)}$, $\mathbf{y}^{(n-2)}$, $\mathbf{y}^{(n-1)}$.

EXERCISE 5.6.– *Kalman decomposition*

See the correction video at https://youtu.be/reOIIEOPz50

A linear system can always be decomposed, after a suitable change of basis, into four sub-systems $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4$ where $\mathcal{S}_1$ is controllable and observable, $\mathcal{S}_2$ is non-controllable and observable, $\mathcal{S}_3$ is controllable and non-observable and $\mathcal{S}_4$ is neither controllable nor observable. The dependencies between the sub-systems can be summarized by Figure 5.4.

Let us note that, on the figure, there is no path (respecting the direction of the arrows) leading from the input $\mathbf{u}$ to a non-controllable system. Similarly, there is no path leading from a non-

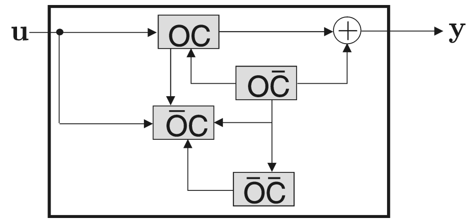


Figure 5.4: Principle of the Kalman decomposition ; C: controllable, O: observable, $\bar{\text{C}}$: non-controllable, $\bar{\text{O}}$: non-observable

observable system to $\mathbf{y}$. We consider the system described by the state equation:

$$\left\{ \begin{array}{lll} \dot{\mathbf{x}}(t) & = & \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{22} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{31} & \mathbf{A}_{32} & \mathbf{A}_{33} & \mathbf{A}_{34} \\ \mathbf{0} & \mathbf{A}_{42} & \mathbf{0} & \mathbf{A}_{44} \end{pmatrix} \mathbf{x}(t) + \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{0} \\ \mathbf{B}_3 \\ \mathbf{0} \end{pmatrix} \mathbf{u}(t) \\ \mathbf{y}(t) & = & \begin{pmatrix} \mathbf{C}_1 & \mathbf{C}_2 & \mathbf{0} & \mathbf{0} \end{pmatrix} \mathbf{x}(t) + (\mathbf{D}) \mathbf{u}(t) \end{array} \right.$$

Draw a block diagram of the system. Show a decomposition in 4 sub-systems $\mathcal{S}_i$ corresponding to the Kalman decomposition.

Exercise 5.7.– *Resolution of the pole placement equation*

See the correction video at https://youtu.be/65nmtKCWzfw

We will illustrate here the resolution of the pole placement equation when the system only has a single input. We consider the system:

$$\dot{\mathbf{x}} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \mathbf{x} + \begin{pmatrix} 1 \\ 2 \end{pmatrix} u$$

that we are looking to stabilize by state feedback of the form: $u = w - \mathbf{K}\mathbf{x}$, with: $\mathbf{K} = (k_1 \quad k_2)$. Calculate $\mathbf{K}$ so that this characteristic polynomial $P_{\text{con}}(s)$ of the closed-loop system has the roots -1 and -1.

Exercise 5.8.– *Output feedback of a scalar system*

See the correction video at https://youtu.be/6jbn4wP8IEU

Let us consider the following state equation:

$$\left\{ \begin{array}{lll} \dot{x} & = & 3x + 2u \\ y & = & 4x \end{array} \right.$$

1) Propose an output feedback controller which puts all the poles in -1 and such that the setpoint variable corresponding to x (in other words if we fix the setpoint at $\bar{w}$, we want the state x to converge towards $\bar{w}$).

2) Give the state equations of the looped system. What are the poles of the looped system ?

EXERCISE 5.9.– *Separation principle*

See the correction video at https://youtu.be/C71Yfs-mnZ4

Let us consider a system looped by a pole placement method, as represented on Figure 5.5.

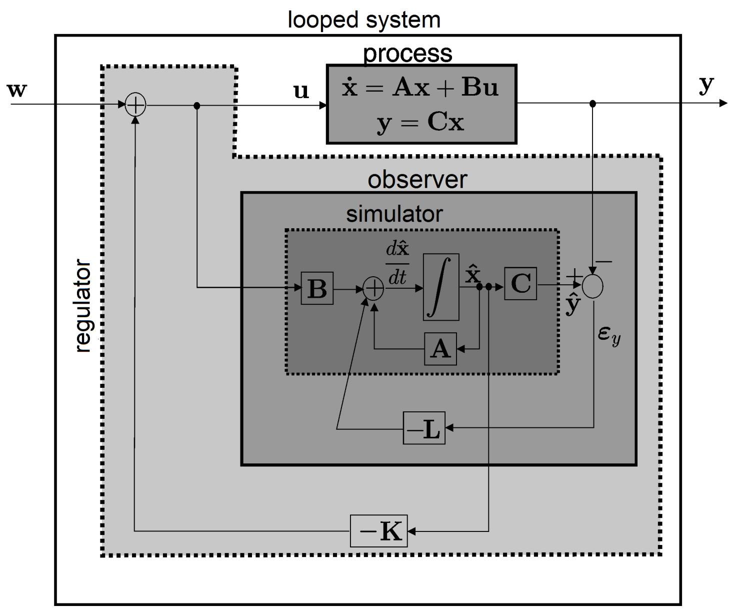


Figure 5.5: Output feedback controller

1) Give the state equations of the controller.

2) Find the state equations of the looped system. We will take as state vector the vector $(\mathbf{x}^{\mathrm{T}}\ \hat{\mathbf{x}}^{\mathrm{T}})^{\mathrm{T}}$.

3) Let us take $\boldsymbol{\varepsilon}_x = \hat{\mathbf{x}} - \mathbf{x}$ and take as new state vector $(\mathbf{x}^{\mathrm{T}}\ \boldsymbol{\varepsilon}_x^{\mathrm{T}})^{\mathrm{T}}$. Give the associated state representation. Show that $\boldsymbol{\varepsilon}_x$ is not controllable.

4) Show that the characteristic polynomial of the looped system is given by:

$$P(s) = \det\left(s\mathbf{I} - \mathbf{A} + \mathbf{BK}\right).\det\left(s\mathbf{I} - \mathbf{A} + \mathbf{LC}\right)$$

What can be deduced from this about the relationship between the poles of this system and the ones we have placed ?

EXERCISE 5.10.– *Proportional and derivative control of a pump*

See the correction video at https://youtu.be/9lFBusPAzfA

We consider the direct current motor of Figure 5.6.

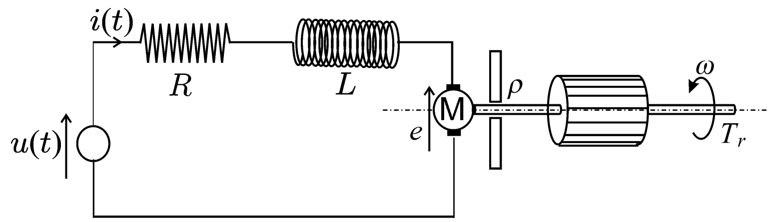


Figure 5.6: Direct current motor

Its state equations are:

$$\begin{cases} \frac{di}{dt} = -\frac{R}{L}i - \frac{\kappa}{L}\omega + \frac{u}{L} \\ \dot{\omega} = \frac{\kappa}{J}i - \frac{\rho}{J}\omega - \frac{T_r}{J} \end{cases}$$

where κ, R, L, ρ, J are constant parameters of the motor. The inputs are the voltage u and the torque T_r, the state variables are i and ω. We use this motor to pump water from a well. In this case, the torque used is $T_r = \alpha\omega$, where α is a constant parameter. We will ignore ρ before α. There is consequently a single input for the motor+pump system: the voltage u.

1) Give the state equations of the motor+pump system.

2) We choose as output $y = \omega$. Calculate the transfer function of the motor+pump system.

3) Give the differential equation associated with this transfer function.

4) We take as state vector $\mathbf{x} = (y, \dot{y})^{\mathrm{T}}$. Give a state representation of the system in matrix form.

5) A *proportional-derivative* controller is a linear combination of the output y, its derivative $\dot{y}$, and the setpoint w (be careful not to confuse the setpoint w with the speed of the motor shaft ω). This controller is written in the form:

$$u = hw - k_1 y - k_2 \dot{y}$$

Give the state equations of the motor+pump system looped by such a controller.

6) Give the values of k_1 and k_2 that we need to choose in order to have the poles of the looped system equal to -1.

7) Find h in such a way that y converges towards w when the setpoint w is constant.

Exercise 5.11.– *PID control*

See the correction video at https://youtu.be/uwcSOmcADHw

We consider a second-order system of the form described by the following differential equation:

$$\ddot{y} + a_1\dot{y} + a_0 y = u$$

1) Give the state equation of the system in matrix form. We will take as state vector $\mathbf{x} = (y \;\; \dot{y})^{\mathrm{T}}$.

2) Let w be a setpoint that we will assume constant. We would like $y(t)$ to converge towards w. We define the error by: $e(t) = w - y(t)$. We suggest controlling our system by the following PID (proportional-derivative-integral) controller:

$$u(t) = \alpha_{-1} \int_0^t e(\tau)\, d\tau \quad + \quad \alpha_0 e(t) \quad + \quad \alpha_1 \dot{e}(t)$$

where the α_i are the coefficients of the controller. This is a state feedback controller, where we assume that $\mathbf{x}(t)$ is measured. Give the state equations of this PID controller with the inputs $\mathbf{x}$,w and output u. A state variable $z(t) = \int_0^t e(\tau)\, d\tau$ will have to be created in order to take into account the integrator of the control.

3) Draw the block diagram of the looped system. This diagram will only be composed of integrators, adders and amplifiers. Encircle the controller on the one hand and the system to be controlled on the other.

4) Give the state equations of the looped system in matrix form.

5) Choose the coefficients α_i of the control (in function of the a_i) so as to have a stable looped system in which all the poles are equal to -1 ?

6) We slightly change the value of the parameters a_0 and a_1 while keeping the same controller. We assume that this modification does not destabilize our system. The new values for a_0, a_1 are denoted by a'_0, a'_1. For a value $\bar{w}$ for given w, what value $\bar{y}$ does converge y to ? Conclude.

Exercise 5.12.– *Canonical control form*

See the correction video at https://youtu.be/G6UHzx8dleA

We consider the system described by its canonical control form as in the Figure 5.7.

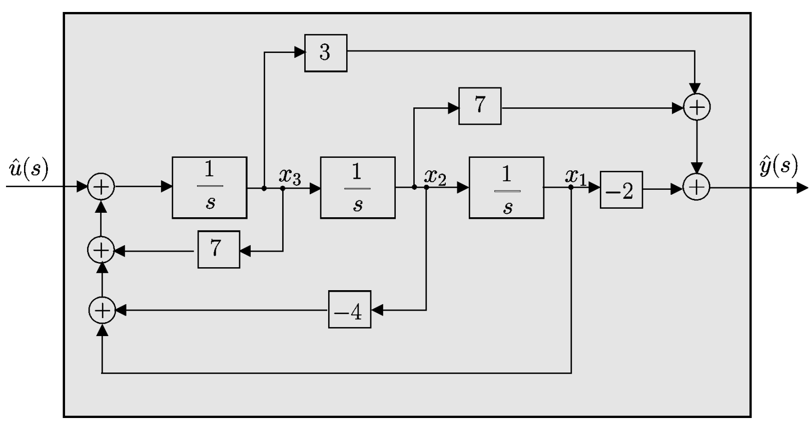


Figure 5.7: State feedback for a linear system in canonical control form

1) Give the state equations of this system. What is its characteristic polynomial ?

2) We would like to control this system by a state feedback of the form $u = -\mathbf{K}\mathbf{x} + hw$, where w is the setpoint. Calculate $\mathbf{K}$ in order for all the poles to be equal to -1.

3) We would like, at equilibrium (*i.e.*, when the setpoint and the output no longer change), to have $y = w$. Deduce from this the value to take for the precompensator h.

Exercise 5.13.– *State feedback with integral effect, monovariate case*

See the correction video at https://youtu.be/mp4tF_D5lk4

We consider the system described by the state equations:

$$\left\{ \begin{array}{lcl} \dot{\mathbf{x}} & = & \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix} \mathbf{x} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u \\ y & = & x_1 \end{array} \right.$$

where u is the input, y the output and $\mathbf{x}$ the state vector.

1) Give the characteristic polynomial of the system. Is the system stable ?

2) We loop the system by the following state feedback control:

$$u = \alpha \int_0^t (w(\tau) - y(\tau))\, d\tau \quad - \quad \mathbf{Kx}, \quad \text{avec } \mathbf{K} = (k_1 \quad k_2)$$

where w is the setpoint. Give the state equations of the controller (we will denote by z the state variable of the controller). What are the poles of the controller ?

3) Give the state equations of the looped system.

4) Calculate $\mathbf{K}$ and α in order for all the poles to be equal to -1.

5) We choose a setpoint $w = \bar{w}$ constant in time. What values $\bar{\mathbf{x}}$ and $\bar{z}$, does the state of the system $\mathbf{x}$ and state of the controller z tend to ? What value $\bar{y}$ does the output y tend to ?

6) We now replace the evolution matrix $\mathbf{A}$ by another matrix $\bar{\mathbf{A}}$ close to $\mathbf{A}$, while keeping the same controller. What value y will it converge to ?

Exercise 5.14.– *State feedback with integral effect, general case*

See the correction video at https://youtu.be/rCRQ3IjEE1w

We consider the system described by the state equation $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} + \mathbf{p}$ where $\mathbf{p}$ is an unknown and constant disturbance vector which represents an external disturbance. We will take $m = \dim \mathbf{u}$ and $n = \dim \mathbf{x}$. A state feedback controller with integral effect is of the form:

$$\mathbf{u} = \mathbf{K}_i \int_0^t (\mathbf{w} - \mathbf{y})\, dt - \mathbf{Kx}$$

where $\mathbf{w}$ is the setpoint and $\mathbf{y}$ is a vector with same dimension as $\mathbf{u}$ representing the setpoint state variables (in other words those we wish to control directly by using $\mathbf{w}$). The vector $\mathbf{y}$ is linked to the state $\mathbf{x}$ by the relation $\mathbf{y} = \mathbf{Cx}$.

1) Give the state equations of the looped system. Recall that the state of the looped system will be composed of the state $\mathbf{x}$ of the system to control and of the vector $\mathbf{z}$ of the values memorized in the integrators. Give the dimensions of $\mathbf{w}, \mathbf{y}, \mathbf{p}, \mathbf{z}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{K}_i, \mathbf{K}$.

2) Find the matrices $\mathbf{K}$ and $\mathbf{K}_i$ so that all the poles of the looped system are equal to -1.

3) Show that, for a constant setpoint $\mathbf{w}$ and disturbance $\mathbf{p}$, we necessarily have $\mathbf{y} = \mathbf{w}$, once the steady state has been reached. What is then the value of $\mathbf{p}$ (in function of $\mathbf{x}$ and $\mathbf{z}$) ?

4) Give the state equations of the controller.

Exercise 5.15.– *Luenberger observer*

See the correction video at https://youtu.be/HZmH9Rw9DVw

We consider the system with input u and output y of Figure 5.8.

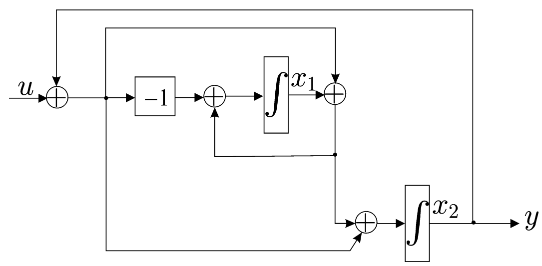


Figure 5.8: Second order system for which we need to design an observer

1) Give, in matrix form, a state representation for this system. Deduce from this a simplification for the block diagram.

2) Study the stability of this system.

3) Give the transfer function of this system.

4) Is the system controllable ? Is it observable ?

5) Find an observer which allows to generate a state $\hat{\mathbf{x}}(t)$, such that the error $||\hat{\mathbf{x}} - \mathbf{x}||$ converges towards zero at e^{-t} (which means to place all the poles at -1). Give this observer in state equation form.

Exercise 5.16.– *Demodulator*

See the correction video at https://youtu.be/Onqy3pbPeMw

Consider the sinusoidal signal with pulsation ω

$$y(t) = a\cos(\omega t + b)$$

where the parameters a and b are unknown. From the measure of $y(t)$, we need to find the amplitude a of the signal $y(t)$.

1) Find an order 2 state equation capable of generating the signal $y(t)$. We will take as state variables $x_1 = \omega y$ and $x_2 = \dot{y}$.

2) Let us assume that at time t, we know the state vector $\mathbf{x}(t)$. Deduce from this an expression of the amplitude a of the signal $y(t)$ in function of $\mathbf{x}(t)$.

3) We only measure $y(t)$. Propose a state observer (by a pole placement method) which generates an estimation $\hat{\mathbf{x}}(t)$ of the state $\mathbf{x}(t)$. We will place all the poles at -1.

4) Deduce from this the state equations of a state estimator with input y and output $\hat{a}$ which gives us an estimation $\hat{a}$ of the amplitude a of a sinusoidal signal with pulsation ω. Build a program to check the behavior of the observer for the sampled signal

$$y(t) = 2\cos t + n(t)$$

with $t \in \{0, dt, 2dt, \ldots, 10\}$ where $dt = 0.01$ is the sampling period. The signal $n(t)$ is a white Gaussian noise with standard deviation 0.1. Draw also the filtered signal $\hat{y}(t)$ as well as the estimation $\hat{a}(t)$ of the amplitude.

5) Draw the Bode diagram of the filter that generates $\hat{y}(t)$ from $y(t)$. Discuss.

Exercise 5.17.– *Output feedback of a non-strictly proper system*

See the correction video at https://youtu.be/X4WRTk4UBIg

We consider the system $\mathcal{S}$ with input u and output y described by the differential equation:

$$\dot{y} - 2y = \dot{u} + 3u$$

Since in this case the degree of differentiation of u is not strictly smaller than that of y, the system is not strictly proper.

1) Write this system in a state representation form.

2) In order to make this system strictly proper (in other words with a direct matrix $\mathbf{D}$ equal to zero), we create a new output $z = y - \alpha u$. Give the appropriate value of α. We will denote by $\mathcal{S}_z$ this new system whose input is u and output is z.

3) Find an output feedback controller for $\mathcal{S}_z$ which sets all the poles of the looped system to -1.

4) Deduce from this a controller $\mathcal{S}_r$ (in state equation form) for $\mathcal{S}$ which sets all the poles of the looped system to -1. We will denote by $\mathcal{S}_b$ the system $\mathcal{S}$ looped by the controller $\mathcal{S}_r$.

5) Give the state equations of the looped system $\mathcal{S}_b$.

6) Calculate the transfer function of the looped system $\mathcal{S}_b$.

7) Calculate the static gain of the looped system. This gain corresponds to the ratio $\frac{\bar{y}}{\bar{w}}$ in a steady state. Deduce from this the gain of the precompensator to be placed before the system which would allow to have a static gain of 1.

Exercise 5.18.– *Output feedback with integral effect*

See the correction video at https://youtu.be/0bpDeghF2Qo

We consider the system described by the state equation:

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

where $\mathbf{p}$ is a disturbance vector, representing an external disturbance which could not be taken into account in the modeling (wind, the slope of the terrain, the weight of the people in an elevator, etc.). The vector $\mathbf{p}$ is assumed to be known and constant. We will take $m = \dim \mathbf{u}$, $n = \dim \mathbf{x}$ and $p = \dim \mathbf{y}$. We need to control this system using the output feedback controller with integral effect represented on Figure 5.9.

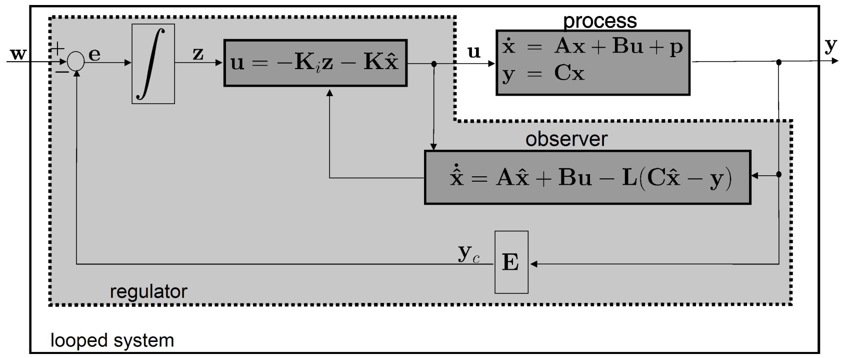


Figure 5.9: Output feedback controller with integral effect

On this figure, $\mathbf{w}$ is the setpoint and $\mathbf{y}_c$ is a vector with the same dimension as $\mathbf{u}$ representing the setpoint output variables (in other words those that we wish to control directly using $\mathbf{w}$ and thus have $\mathbf{y}_c = \mathbf{w}$, when the setpoint is constant). The vector $\mathbf{y}_c$ satisfies the relation $\mathbf{y}_c = \mathbf{E}\mathbf{y}$, where $\mathbf{E}$ is a known matrix.

1) Give the state equations of the looped system in matrix form.

2) Let us take $\boldsymbol{\varepsilon} = \hat{\mathbf{x}} - \mathbf{x}$. Express these state equations in matrix form by taking, this time, the state vectors $(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\varepsilon})$.

3) Show how we can arbitrarily set all the poles of the looped system.

4) Show that, for a constant setpoint $\mathbf{w}$ and disturbance $\mathbf{p}$, in a steady state, we necessarily have $\mathbf{y}_c = \mathbf{w}$.

5) Give the state equations of the controller, in matrix form.

Exercise 5.19.– *Poincaré map*

See the correction video at https://youtu.be/HW9XjU9hJpU

We consider the tank described by the state equation

$$\begin{cases} \dot{x}_1 &= \cos x_3 \\ \dot{x}_2 &= \sin x_3 \\ \dot{x}_3 &= u \end{cases}$$

where (x_1, x_2) corresponds to the position of the robot and x_3 is its heading. The robot has an heading control of the form $u = \sin(\bar{\psi} - x_3)$. Moreover, the desired heading $\bar{\psi}$ obeys to the automaton (or Petri net) of Figure 5.10. The variable $q \in \{0, 1, 2\}$ is a discrete and c is a clock which is initialized to 0 each time q changes.

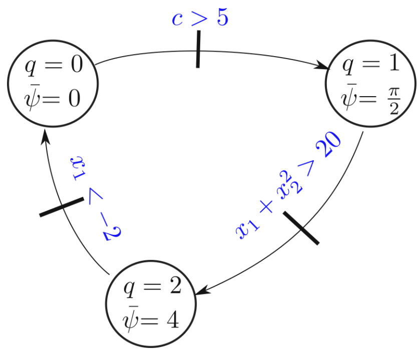


Figure 5.10: Automaton deciding the desired heading

1) Using a simulation, show that the state $\mathbf{x}(t)$ converges to a stable limit cycle. Take for the initial state $\mathbf{x}(0) = (-3, 1, -1)$.

2) Given a surface of the state space $\mathcal{S}$ defined by $g(\mathbf{x}) = 0$ which is transverse to the field $\mathbf{f}$, *i.e.*,

$$\mathbf{x} \in \mathcal{S} \Rightarrow \left(\frac{\partial g}{\partial \mathbf{x}} \cdot \mathbf{f}\right)(\mathbf{x}) \neq 0$$

almost everywhere. The surface $\mathcal{S}$ is called the *Poincaré section*. Consider a point $\mathbf{a} \in \mathcal{S}$ such that $\left(\frac{\partial g}{\partial \mathbf{x}} \cdot \mathbf{f}\right)(\mathbf{a}) > 0$. We define *Poincaré map* as the application which associates the state $\mathbf{p}(\mathbf{a})$ of the surface $\mathcal{S}$ such that the trajectory initialized at $\mathbf{a}$ comes back for the first time in $\mathcal{S}$, with $\left(\frac{\partial g}{\partial \mathbf{x}} \cdot \mathbf{f}\right)(\mathbf{p}(\mathbf{a})) > 0$. See Figure 5.11. Show that for all $q = 0$. the surface $\mathcal{S} : x_1 + 2 = 0$ is transverse to $\mathbf{f}$ where $\mathbf{f}(\mathbf{x}, q)$ is the field associated to the controlled system.

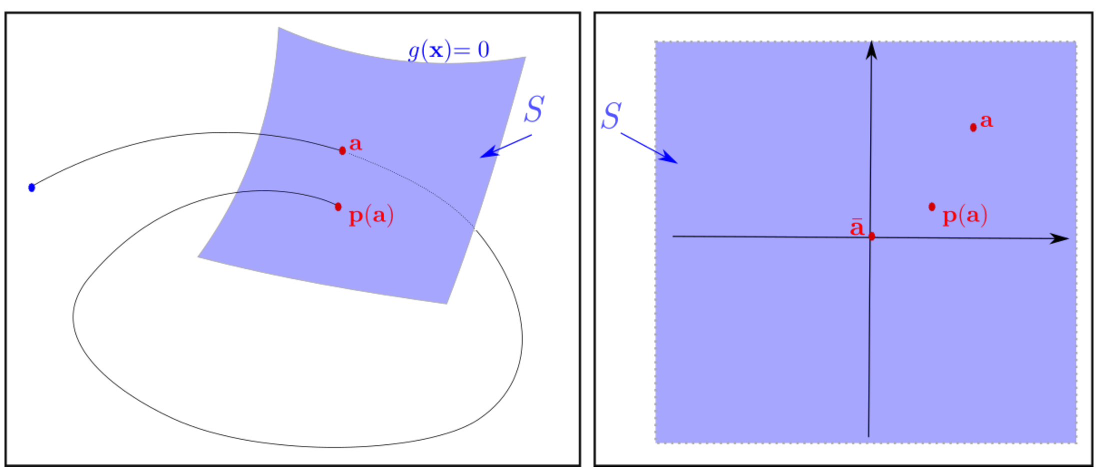


Figure 5.11: Left: the Poincaré map; Right: a view of the Poincaré map restricted to the Poincaré section $\mathcal{S}$

3) Build a program which estimates the fixed point which satisfies $\mathbf{p}(\bar{\mathbf{a}}) = \bar{\mathbf{a}}$. Note that the trajectory is in the surface $\mathcal{S}$ when q switches from $q = 2$ to $q = 0$.

4) Compute an estimate of the Jacobian matrix $\frac{d\mathbf{p}}{d\mathbf{a}}(\bar{\mathbf{a}})$. Conclude about the stability of the limit cycle.

Chapter 6

Linearized control

In Chapter 5, we have shown how to design controllers for linear systems. However, in practice, the systems are rarely linear. Nevertheless, if their state vector remains localized in a small zone of the state space, the system may be considered linear and the techniques developed in Chapter 5 can then be used. We will first show how to linearize a non-linear system around a given point of the state space. We will then discuss how to stabilize these non-linear systems.

6.1 Linearization

6.1.1 Linearization of a function

Let $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^p$ be a differentiable function. In the neighborhood of a point $\bar{\mathbf{x}} \in \mathbb{R}^n$ the first-order Taylor development of $\mathbf{f}$ around $\bar{\mathbf{x}}$ gives us:

$$\mathbf{f}(\mathbf{x}) \simeq \mathbf{f}(\bar{\mathbf{x}}) + \frac{d\mathbf{f}}{d\mathbf{x}}(\bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})$$

with

$$\frac{d\mathbf{f}}{d\mathbf{x}}(\bar{\mathbf{x}}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\bar{\mathbf{x}}) & \frac{\partial f_1}{\partial x_2}(\bar{\mathbf{x}}) & \dots & \frac{\partial f_1}{\partial x_n}(\bar{\mathbf{x}}) \\ \frac{\partial f_2}{\partial x_1}(\bar{\mathbf{x}}) & \frac{\partial f_2}{\partial x_2}(\bar{\mathbf{x}}) & \dots & \frac{\partial f_2}{\partial x_n}(\bar{\mathbf{x}}) \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_p}{\partial x_1}(\bar{\mathbf{x}}) & \frac{\partial f_p}{\partial x_2}(\bar{\mathbf{x}}) & \dots & \frac{\partial f_p}{\partial x_n}(\bar{\mathbf{x}}) \end{pmatrix}.$$

This matrix is called the *Jacobian matrix*. Very often, in order to linearize a function, we use formal calculus for calculating the Jacobian matrix, then we instantiate this matrix around $\bar{\mathbf{x}}$. When we differentiate by hand, we avoid such proceedings. We save a lot of calculations if we perform the two operations (differentiation and instantiation) simultaneously. Finally, there is a similar method which is very easy to implement: the *finite difference* method. In order to apply it to the calculation of $\frac{d\mathbf{f}}{d\mathbf{x}}$ at point $\bar{\mathbf{x}}$, we approximate the j^{th} column of the Jacobian matrix as follows:

$$\frac{\partial \mathbf{f}}{\partial x_j}(\bar{\mathbf{x}}) \simeq \frac{\mathbf{f}(\bar{\mathbf{x}}+h\mathbf{e}_j) - \mathbf{f}(\bar{\mathbf{x}})}{h}$$

where $\mathbf{e}_j$ is the j^{th} vector of the canonical basis of $\mathbb{R}^n$ and h a small real number. Thus, the Jacobian matrix is approximated by:

$$\frac{d\mathbf{f}}{d\mathbf{x}}(\bar{\mathbf{x}}) \simeq \left(\begin{array}{c|c|c|c} \frac{\mathbf{f}(\bar{\mathbf{x}}+h\mathbf{e}_1)-\mathbf{f}(\bar{\mathbf{x}})}{h} & \frac{\mathbf{f}(\bar{\mathbf{x}}+h\mathbf{e}_2)-\mathbf{f}(\bar{\mathbf{x}})}{h} & \dots & \frac{\mathbf{f}(\bar{\mathbf{x}}+h\mathbf{e}_n)-\mathbf{f}(\bar{\mathbf{x}})}{h} \end{array} \right)$$

To have a correct approximation, we need to take a very small h. However, if h is too small, numerical problems appear. This method therefore has to be considered weak.

6.1.2 Linearization of a dynamic system

Let us consider the system described by its state equations:

$$\mathcal{S} : \left\{ \begin{array}{lll} \dot{\mathbf{x}} & = & \mathbf{f}(\mathbf{x},\mathbf{u}) \\ \mathbf{y} & = & \mathbf{g}(\mathbf{x},\mathbf{u}) \end{array} \right.$$

where $\mathbf{x}$ is of dimension n, $\mathbf{u}$ is of dimension m and $\mathbf{y}$ is of dimension p. Around the point $(\bar{\mathbf{x}},\bar{\mathbf{u}})$, the behavior of $\mathcal{S}$ is therefore approximated by the following state equations:

$$\left\{ \begin{array}{lll} \dot{\mathbf{x}} & = & \mathbf{f}(\bar{\mathbf{x}},\bar{\mathbf{u}}) + \mathbf{A}(\mathbf{x}-\bar{\mathbf{x}}) + \mathbf{B}(\mathbf{u}-\bar{\mathbf{u}}) \\ \mathbf{y} & = & \mathbf{g}(\bar{\mathbf{x}},\bar{\mathbf{u}}) + \mathbf{C}(\mathbf{x}-\bar{\mathbf{x}}) + \mathbf{D}(\mathbf{u}-\bar{\mathbf{u}}) \end{array} \right.$$

with:

$$\begin{array}{ll} \mathbf{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\bar{\mathbf{x}},\bar{\mathbf{u}}), & \mathbf{B} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\bar{\mathbf{x}},\bar{\mathbf{u}}) \\ \mathbf{C} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\bar{\mathbf{x}},\bar{\mathbf{u}}), & \mathbf{D} = \frac{\partial \mathbf{g}}{\partial \mathbf{u}}(\bar{\mathbf{x}},\bar{\mathbf{u}}) \end{array}$$

This is an affine system called *tangent system* to $\mathcal{S}$ at point $(\bar{\mathbf{x}},\bar{\mathbf{u}})$.

6.1.3 Linearization around an operating point

A point $(\bar{\mathbf{x}},\bar{\mathbf{u}})$ is an *operating point* (also called *polarization point*) if $\mathbf{f}(\bar{\mathbf{x}},\bar{\mathbf{u}}) = \mathbf{0}$. If $\bar{\mathbf{u}} = \mathbf{0}$, we are in the case of an *equilibrium point*. Let us note first of all that if $\mathbf{x} = \bar{\mathbf{x}}$ and if $\mathbf{u} = \bar{\mathbf{u}}$, then $\dot{\mathbf{x}} = \mathbf{0}$, in other words the system no longer evolves if we maintain the control $\mathbf{u} = \bar{\mathbf{u}}$ when the system is in the state $\bar{\mathbf{x}}$. In this case, the output $\mathbf{y}$ has the value $\mathbf{y} = \bar{\mathbf{y}} = \mathbf{g}(\bar{\mathbf{x}},\bar{\mathbf{u}})$. Around the operating point, $(\bar{\mathbf{x}},\bar{\mathbf{u}})$, the system $\mathcal{S}$ admits the tangent system:

$$\left\{ \begin{array}{lll} \dot{\mathbf{x}} & = & \mathbf{A}(\mathbf{x}-\bar{\mathbf{x}}) + \mathbf{B}(\mathbf{u}-\bar{\mathbf{u}}) \\ \mathbf{y} & = & \bar{\mathbf{y}} + \mathbf{C}(\mathbf{x}-\bar{\mathbf{x}}) + \mathbf{D}(\mathbf{u}-\bar{\mathbf{u}}) \end{array} \right.$$

Let us take $\widetilde{\mathbf{u}} = \mathbf{u} - \bar{\mathbf{u}}$, $\widetilde{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}}$ and $\widetilde{\mathbf{y}} = \mathbf{y} - \bar{\mathbf{y}}$. These vectors are called the *variations* of $\mathbf{u}$, $\mathbf{x}$ and $\mathbf{y}$. For small variations $\widetilde{\mathbf{u}}$, $\widetilde{\mathbf{x}}$, $\widetilde{\mathbf{y}}$, we have:

$$\left\{ \begin{array}{lll} \frac{d}{dt}\widetilde{\mathbf{x}} & = & \mathbf{A}\widetilde{\mathbf{x}} + \mathbf{B}\widetilde{\mathbf{u}} \\ \widetilde{\mathbf{y}} & = & \mathbf{C}\widetilde{\mathbf{x}} + \mathbf{D}\widetilde{\mathbf{u}} \end{array} \right.$$

The system thus formed is called the *linearized system* of $\mathcal{S}$ around the operating point $(\bar{\mathbf{x}},\bar{\mathbf{u}})$.

6.2 Stabilization of a non-linear system

Let us consider the system described by its state equations:

$$\mathcal{S}:\begin{cases}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x},\mathbf{u})\\ \mathbf{y} &= \mathbf{g}(\mathbf{x})\end{cases}$$

in which $(\bar{\mathbf{x}},\bar{\mathbf{u}})$ constitutes an operating point. In order for our system to behave like a linear system around $(\bar{\mathbf{x}},\bar{\mathbf{u}})$, let us create the variables $\widetilde{\mathbf{u}}=\mathbf{u}-\bar{\mathbf{u}}$ and $\widetilde{\mathbf{y}}=\mathbf{y}-\bar{\mathbf{y}}$, as represented on Figure 6.1.

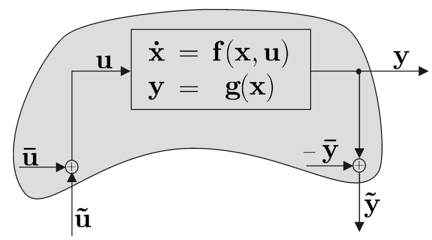


Figure 6.1: Polarization of the system: the initial system, locally affine (in the neighborhood of the operating point) is transformed into a locally linear system

The system with input $\widetilde{\mathbf{u}}$ and output $\widetilde{\mathbf{y}}$ thus designed is called a *polarized system*. The state equations of the polarized system can be approximated by:

$$\begin{cases}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x},\mathbf{u})\simeq\mathbf{A}\cdot(\mathbf{x}-\bar{\mathbf{x}})+\mathbf{B}\cdot(\mathbf{u}-\bar{\mathbf{u}})=\mathbf{A}\cdot(\mathbf{x}-\bar{\mathbf{x}})+\mathbf{B}\cdot\widetilde{\mathbf{u}}\\ \widetilde{\mathbf{y}} &= -\bar{\mathbf{y}}+\mathbf{y}=-\bar{\mathbf{y}}+\mathbf{g}(\mathbf{x})\simeq-\bar{\mathbf{y}}+\bar{\mathbf{y}}+\mathbf{C}\,(\mathbf{x}-\bar{\mathbf{x}})=\mathbf{C}\cdot(\mathbf{x}-\bar{\mathbf{x}})\end{cases}$$

where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ are obtained by the calculation of the Jacobian matrix of the functions $\mathbf{f}$ and $\mathbf{g}$ at point $(\bar{\mathbf{x}},\bar{\mathbf{u}})$. By taking now $\widetilde{\mathbf{x}}=\mathbf{x}-\bar{\mathbf{x}}$ as state vector instead of $\mathbf{x}$, we obtain:

$$\begin{cases}\frac{d}{dt}\widetilde{\mathbf{x}} &= \mathbf{A}\widetilde{\mathbf{x}}+\mathbf{B}\widetilde{\mathbf{u}}\\ \widetilde{\mathbf{y}} &= \mathbf{C}\widetilde{\mathbf{x}}\end{cases}$$

which is the linearized system of the non-linear system. Let $\mathbf{x}_c=\mathbf{E}\mathbf{x}$ be the sub-vector of the setpoint variables. We can build a controller $\mathcal{R}_{\mathrm{L}}$ for this linear system using the REGULKLH algorithm $(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{E},\mathbf{p}_{\mathrm{con}},\mathbf{p}_{\mathrm{obs}})$ described on page 61. We know that when the setpoint $\widetilde{\mathbf{w}}$ input into $\mathcal{R}_{\mathrm{L}}$ is constant, we have $\mathbf{E}\widetilde{\mathbf{x}}=\widetilde{\mathbf{w}}$. However, we would like the input of the controller $\mathbf{w}$ that we are building to satisfy $\mathbf{w}=\mathbf{E}\mathbf{x}$. We therefore have to build $\widetilde{\mathbf{w}}$ from $\mathbf{w}$ in such a way that $\mathbf{w}=\mathbf{E}\mathbf{x}$, at equilibrium. We have:

$$\widetilde{\mathbf{w}}=\mathbf{E}\widetilde{\mathbf{x}}=\mathbf{E}\,(\mathbf{x}-\bar{\mathbf{x}})=\mathbf{w}-\bar{\mathbf{w}}$$

where $\bar{\mathbf{w}}=\mathbf{E}\bar{\mathbf{x}}$. The controller thus obtained is represented on Figure 6.2, in the thick frame.

This controller stabilizes and decouples our non-linear system around its operating point. A summary of the method to calculate a controller for a non-linear system is given below.

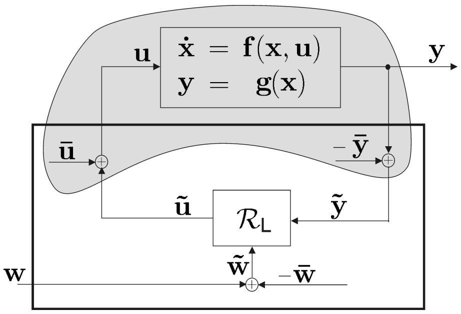


Figure 6.2: Stabilization of a non-linear system around an operating point by a linear controller

Algorithm $\text{R}\textsc{egul}\text{NL}$ **(in:** $\mathbf{f}, \mathbf{g}, \mathbf{E}, \mathbf{p_{con}}, \mathbf{p_{obs}}, \bar{\mathbf{x}}, \bar{\mathbf{u}}$ **; out :** $\mathcal{R}$**)**
Check that $\mathbf{f}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = \mathbf{0}$;
$\bar{\mathbf{y}} := \mathbf{g}(\bar{\mathbf{x}})\,;\ \bar{\mathbf{w}} := \mathbf{E}\bar{\mathbf{x}}$;
$\mathbf{A} := \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\bar{\mathbf{x}}, \bar{\mathbf{u}})\,;\ \mathbf{B} := \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\bar{\mathbf{x}}, \bar{\mathbf{u}})\,;\ \mathbf{C} := \frac{\partial \mathbf{g}}{\partial \mathbf{x}}(\bar{\mathbf{x}}, \bar{\mathbf{u}})\,;$
$\mathbf{K} := \ \textsc{place}(\mathbf{A}, \mathbf{B}, \mathbf{p}_{\text{con}})\,;$
$\mathbf{L} := \ \left(\textsc{place}\left(\mathbf{A}^{\mathrm{T}}, \mathbf{C}^{\mathrm{T}}, \mathbf{p}_{\text{obs}}\right)\right)^{\mathrm{T}};$
$\mathbf{H} := -\left(\mathbf{E}(\mathbf{A} - \mathbf{B}\mathbf{K})^{-1}\mathbf{B}\right)^{-1};$
$\mathcal{R} := \begin{cases} \frac{d}{dt}\hat{\mathbf{x}} &= (\mathbf{A} - \mathbf{B}\mathbf{K} - \mathbf{L}\mathbf{C})\,\hat{\mathbf{x}} + \mathbf{B}\mathbf{H}(\mathbf{w} - \bar{\mathbf{w}}) + \mathbf{L}(\mathbf{y} - \bar{\mathbf{y}}) \\ \mathbf{u} &= \bar{\mathbf{u}} - \mathbf{K}\hat{\mathbf{x}} + \mathbf{H}(\mathbf{w} - \bar{\mathbf{w}}) \end{cases}$

This algorithm returns our controller with inputs $\mathbf{y}, \mathbf{w}$ and output $\mathbf{u}$ in the form of its state equations.

Exercises

Exercise 6.1.– *Linearization of the predator-prey system*

See the correction video at https://youtu.be/HqmT6kBilZQ

Let us consider the Lotka-Volterra predator-prey system, given by:

$$\begin{cases} \dot{x}_1 &= (1 - x_2)\, x_1 \\ \dot{x}_2 &= (-1 + x_1)\, x_2 \end{cases}$$

1) Linearize this system around a non-zero equilibrium point $\bar{\mathbf{x}}$.

2) Among the two vector fields of Figure 6.3, which one represents the field associated with the Lotka-Volterra system and which one corresponds to its tangent system ?

3) Calculate the poles of the linearized system. Discuss.

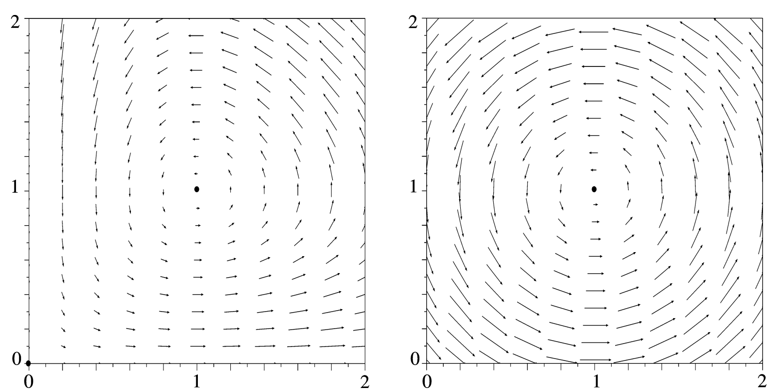


Figure 6.3: One of the two vector fields is associated with the Lotka-Volterra system. The other corresponds to its tangent system

---

Exercise 6.2.– *Water car*

See the correction video at https://youtu.be/o29wNy16XRk

Consider a car moving horizontally in the water as illustrated by Figure 6.4. Its evolution obeys the following differential equation:

$$\ddot{y} + \dot{y}.\left|\dot{y}\right| = u$$

where y is the position of the car and u is the force exerted on this car.

1) Give the state equation for this system. We will take as state vector $\mathbf{x} = (y \;\; \dot{y})^{\mathrm{T}}$.

2) Linearize around its equilibrium point. Study the stability of the system.

3) In the case of a positive initial speed $\dot{y}(0)$, give the solution of the non-linear differential equation when $u = 0$. To which value does y converge to?

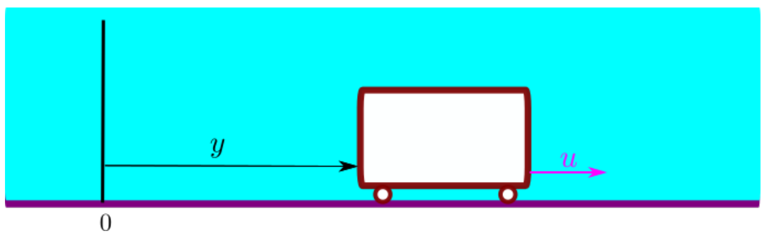


Figure 6.4: The car moves on the sea-floor

EXERCISE 6.3.– *Control of a first-order non-linear system*

See the correction video at https://youtu.be/NrwbXbThk1M

Let us consider the non-linear system given by the state equations:

$$\mathcal{S}: \begin{cases} \dot{x} & = & 2x^2 + u \\ y & = & 3x \end{cases}$$

which we wish to stabilize around the state $\bar{x} = 2$. At equilibrium, we would like y to be equal to its setpoint w. Moreover, we would like all the poles of the looped system to be equal to -1.

1) Give the state equations of the controller using pole placement which satisfies these constraints.

2) What are the state equations of the looped system?

EXERCISE 6.4.– *State feedback*

See the correction video at https://youtu.be/NifPfedkeek

Let us consider the system represented by Figure 6.5.

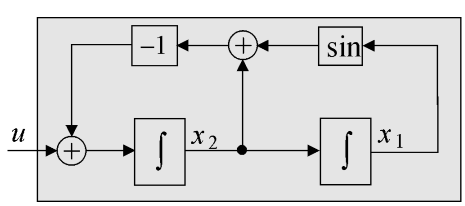


Figure 6.5: block diagram for a non-linear system

1) Give the state equations of the system.

2) Calculate its equilibrium points.

3) Linearize this system around an equilibrium point $\bar{\mathbf{x}}$ corresponding to $x_1 = \pi$. Is this a stable equilibrium point ?

4) Propose a state feedback controller of the form $u = -\mathbf{K}(\mathbf{x} - \bar{\mathbf{x}})$ which stabilizes the system around $\bar{\mathbf{x}}$. Place all poles at -1.

Exercise 6.5.– *Control of the segway*

See the correction video at https://youtu.be/xKQ9Z64Z0xY

The segway is a vehicle with two wheels and a single axle. We consider first of all that the segway moves in a straight line and therefore a two-dimensional model will suffice (see Figure 6.6).

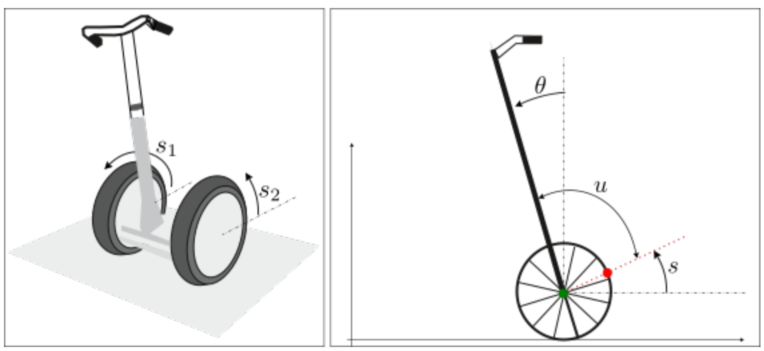


Figure 6.6: Segway which is the subject of the controllability study

This system has two degrees of freedom: the angle of the wheel s and the pitch θ. Its state vector is therefore the vector $\mathbf{x} = (s, \theta, \dot{s}, \dot{\theta})^{\mathrm{T}}$. When it is not controlled, the state equations of the segway are (refer to Exercise 2.6):

$$\left\{\begin{array}{ccc} \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} & = & \begin{pmatrix} x_3 \\ x_4 \\ \frac{\mu_3\left(\mu_2 x_4^2 - \mu_g \cos x_2\right)\sin x_2 + (\mu_2 + \mu_3 \cos x_2)u}{\mu_1\mu_2 - \mu_3^2 \cos^2 x_2} \\ \frac{\left(\mu_1\mu_g - \mu_3^2 x_4^2 \cos x_2\right)\sin x_2 - (\mu_1 + \mu_3 \cos x_2)u}{\mu_1\mu_2 - \mu_3^2 \cos^2 x_2} \end{pmatrix} \\ y & = & x_1 \end{array}\right.$$

where:

$$\mu_1 = J_M + \rho^2 (m + M), \qquad \mu_2 = J_p + m\ell^2,$$
$$\mu_3 = \rho m \ell, \qquad \mu_g = g\ell m$$

The parameters of our system are the mass M of the disk, its radius ρ, its inertial momentum J_M, the mass m of the body, its inertial momentum J_p, the distance ℓ between its center of gravity and the center of the disk. We have added the observation equation $y = x_1$ in order to assume a situation where only the angle of the wheel s is measured. We will take $m = 10$ Kg, $M = 1$ Kg, $\ell = 1$ m, $g = 10$ ms^{-2}, $\rho = 1$ m, $J_p = 10$ Kg.m^2 and $J_M = \frac{1}{2}M\rho^2$. Note that, since $\rho = 1$, the angle s also correspond to the displacement of the segway.

1) Calculate the operating points of the system.

2) Linearize the system around an upper equilibrium point, for $s = 0$.

3) By using the controllability matrix, determine whether the system is controllable.

4) Simulate this system by using Euler's method, in different situations.

5) We assume that we measure the angle s using an odometer. Calculate an output feedback controller which allows to place the segway at a given position. We will place all the poles around -2 (be careful, the pole placement algorithm usually does not allow two identical poles). Propose a simulation to validate the stability of your controller. Validate the robustness of your control by adding sensor noise to the measure of $\mathbf{x}$, which is white and Gaussian.

6) Given the model of the tank, propose a 3D model for the segway. The chosen state vector will be:

$$\mathbf{x} = (\theta, \dot{s}, \dot{\theta}, x, y, \psi, s_1, s_2)^{\mathrm{T}}$$

where s_1 is the angle of the right wheel, s_2 the angle of the left wheel, ψ is the heading of the segway and the (x, y) the coordinates of the center of the Segway's axle. The input are the body/wheels pair u_1 and the differential between the wheels u_2. Here, $\dot{s}$ represents the average angular speed between the two wheels, i.e. $\dot{s} = \frac{1}{2}(\dot{s}_1 + \dot{s}_2)$. Propose a control which allows to regulate the segway speed- and heading-wise.

Exercise 6.6.– *Controlling an inverted rod pendulum*

See the correction video at https://youtu.be/ft7lOc4r1ws

Let us consider the inverted rod pendulum represented on Figure 6.7, composed of a pendulum placed in unstable equilibrium on a carriage.

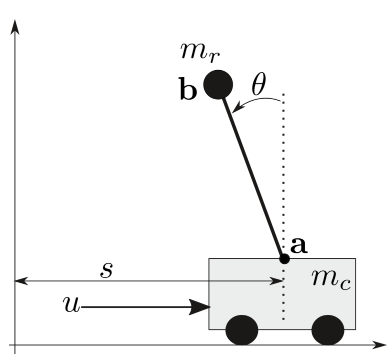


Figure 6.7: Inverted rod pendulum that we need to control

The value u is the force exerted on the carriage of mass $m_c = 5\ kg$, s indicates the position of the carriage, θ is the angle of the pendulum. At the tip $\mathbf{b}$ of the pendulum of length $\ell = 4\ m$ is a fixated mass $m_r = 1$ Kg. Finally, $\mathbf{a}$ is the point of articulation between the rod and the carriage. As seen in Exercise 2.5, the state equations of this system are:

$$\left\{ \begin{array}{ccc} \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} & = & \begin{pmatrix} x_3 \\ x_4 \\ \frac{m_r(\sin x_2)\cdot(g\cos x_2 - \ell x_4^2)+u}{m_c+m_r\sin^2 x_2} \\ \frac{(\sin x_2)\cdot((m_c+m_r)g - m_r\ell x_4^2\cos x_2)+\cos x_2 u}{\ell(m_c+m_r\sin^2 x_2)} \end{pmatrix} \\ y & = & x_1 \end{array} \right.$$

where:

$$\mathbf{x} = (x_1, x_2, x_3, x_4)^{\mathrm{T}} = (s, \theta, \dot{s}, \dot{\theta})^{\mathrm{T}}.$$

The observation equation indicates that only the position of the carriage x_1 is measured.

1) Calculate all the operating points of the system.

2) Linearize the system around the operating point $\bar{\mathbf{x}} = (0, 0, 0, 0)$ and $\bar{u} = 0$.

3) Obtain an output feedback controller which sets all the poles of the looped system to -2. We propose a control in position, which means that the setpoint w corresponds to x_1. Illustrate by a simulation the behavior of the controller.

4) Create a state feedback control which allows to raise the pendulum when it leaves its bottom position. For this, we will attempt to stabilize the mechanical energy of the pendulum before moving on to a linear control. We will assume that $u \in [-u_{\max}, u_{\max}]$. Illustrate on a simulation.

EXERCISE 6.7.– *Linear Quadratic Regulator*

See the correction video at https://youtu.be/257kfCdaKhs

Consider the state space system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

and the cost function

$$J(\mathbf{x}(\cdot), \mathbf{u}(\cdot)) = \int_0^{\infty} (\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} + \mathbf{u}^{\mathrm{T}}\mathbf{R}\mathbf{u}) \cdot dt.$$

It can be shown that a feedback control law that minimizes the value of the cost is given by $\mathbf{u} = -\mathbf{K}\mathbf{x}$ where

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{P}$$

where $\mathbf{P}$ is found by solving the continuous time algebraic Riccati equation given by

$$\mathbf{A}^{\mathrm{T}}\mathbf{P} + \mathbf{P}\mathbf{A} - (\mathbf{P}\mathbf{B})\mathbf{R}^{-1}(\mathbf{B}^{\mathrm{T}}\mathbf{P}) + \mathbf{Q} = \mathbf{0}.$$

We assume this result that can be found using optimal control theory. The corresponding controller is called LQR (Linear Quadratic Regulator). The main advantage of the LQR is that it allows us to give weights to the required dynamics. In general, the matrices are chosen as diagonal.

Consider the inverted pendulum of Exercise 6.6. We assume that the state is measured without any error.

1) With a pole-placement, propose a controller for the system. Draw the functions $s(t), \theta(t), u(t)$.

2) Find a value for $\mathbf{K}$ such that the system initialized at $\mathbf{x}(0) = (2, 0, 0, 0)$ satisfies $\forall t \geq 5, |x_1(t)| < 0.2$, $|u(t)| \leq 0.3$. Give the poles of the closed loop system. Compare with the pole placement method.

3) Find a value for $\mathbf{K}$ such that the system initialized at $\mathbf{x}(0)$ satisfies $\forall t \geq 2, |x_1(t)| < 0.2$.

4) We now have $\mathbf{x}(0) = (50, 0, 0, 0)$. How do we have to choose the controller $\mathbf{K}$ to be able to get a stabilization at $\mathbf{0}$?

Exercise 6.8.– *Autonomous car*

See the correction video at https://youtu.be/cP5x4wiaFxc

Here, we will address the case of a car which drives on an unknown road. The car is equipped with (i) a rangefinder which measures the lateral distance d between the rear axle of the car and the edge of the road, (ii) a speed sensor which measures the speed v of the front wheels, and (iii) an angle sensor which measures the angle δ of the steering wheel (for reasons of simplification, we will assume that δ corresponds also to the angle between the front wheels and the axis of the car). For our simulations, we will assume that our car is driving around a closed polygon as shown on Figure 6.8.

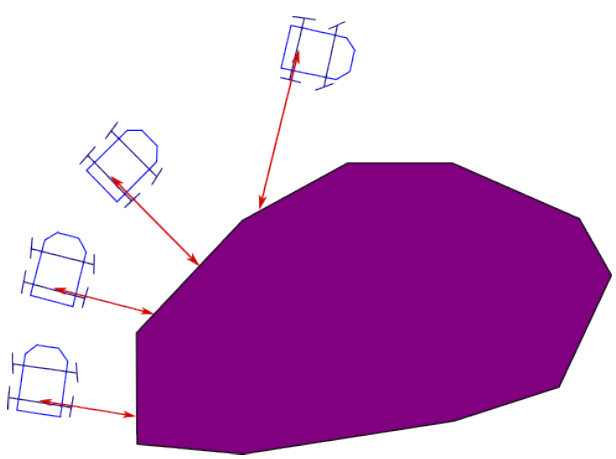


Figure 6.8: Car turning around a polygon

1) Let $\mathbf{m}$, $\mathbf{a}$, $\mathbf{b}$ be three points of $\mathbb{R}^2$ and $\overrightarrow{\mathbf{u}}$ a vector. Show that the half-line $\mathcal{E}\left(\mathbf{m}, \overrightarrow{\mathbf{u}}\right)$ with vertex $\mathbf{m}$ and direction vector $\overrightarrow{\mathbf{u}}$ intersects the segment $[\mathbf{ab}]$ if and only if:

$$\begin{cases} \det\left(\mathbf{a}-\mathbf{m}, \overrightarrow{\mathbf{u}}\right) . \det\left(\mathbf{b}-\mathbf{m}, \overrightarrow{\mathbf{u}}\right) & \leq 0 \\ \det(\mathbf{a}-\mathbf{m}, \mathbf{b}-\mathbf{a}) . \det(\overrightarrow{\mathbf{u}}, \mathbf{b}-\mathbf{a}) & \geq 0 \end{cases}$$

Moreover, show that, in this case, the distance from the point $\mathbf{m}$ to the segment $[\mathbf{ab}]$ following the vector $\overrightarrow{\mathbf{u}}$ is given by:

$$d = \frac{\det(\mathbf{a}-\mathbf{m},\mathbf{b}-\mathbf{a})}{\det(\overrightarrow{\mathbf{u}},\mathbf{b}-\mathbf{a})}.$$

2) We would like to make the car moving (without using a controller) around the polygon using Euler's method. As seen in Exercise 2.7, we could take as evolution equation:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\delta} \end{pmatrix} = \begin{pmatrix} v\cos\delta\cos\theta \\ v\cos\delta\sin\theta \\ v\sin\delta \\ u_1 \\ u_2 \end{pmatrix}$$

Here (x, y) represents the coordinates of the center of the car, θ its heading. The polygon representing the circuit will be represented by the matrix:

$$\mathbf{P} = \begin{pmatrix} -10 & -10 & 0 & 10 & 20 & 32 & 35 & 30 & 20 & 0 & -10 \\ -5 & 5 & 15 & 20 & 20 & 15 & 10 & 0 & -3 & -6 & -5 \end{pmatrix}$$

Since the polygon is closed, the first and last columns of $\mathbf{P}$ are identical. Give an expression of the observation function of the system which returns the distance measured by the rangefinder as well as v and δ.

3) We would like the car to move with a constant speed along the road. Of course, it is out of the question to involve the shape of the road in the controller, since it is unknown. Moreover, the position and orientation variables of the car are not measured (in other words we have no compass or GPS). These quantities however are often unnecessary in order to reach the given objective, which is following the road. Indeed, are we not ourselves capable of driving a car on a road without having a local map, without knowing where we are and where North is ? What we are interested in when driving a car is the relative position of the car with respect to the road. The world as seen by the controller is represented on Figure 6.9. In this world, the car moves laterally, like in a computer game, and the edge of the road remains fixed. This world has to be such that the model used by the controller has state variables describing this relative position of the car with respect to the road and that the constancy of the state variables of this model corresponds to a real situation in which the car follows the road with a constant speed. We need to imagine an ideal world for the controller in which the associated model admits an operating point which corresponds to the desired behavior of our real system. For this, we will assume that our car drives at a distance $\bar{x}$ of the edge. This model has to have the same inputs $\mathbf{u}$ and outputs $\mathbf{y}$ as the real system, more precisely two inputs (the acceleration of the front wheels $\dot{v}$ and the angular speed of the steering wheel $\dot{\delta}$) and three outputs (the distance d of the center of the rear axle to the edge of the road, the speed v of the front wheels and the angle δ of the steering wheel). Given the fact that only the relative position of the car is of interest to us, our model must only contain four state variables: x, θ, v, δ, as represented on the figure. One must be careful though, since the meaning of the variable x has changed since the

previous model. Find the state equations (in other words, the evolution equation and the observation equation) of the system imagined by the controller.

4) Let us choose as operating point

$$\bar{\mathbf{x}} = (5, \pi/2, 7, 0) \text{ et } \bar{\mathbf{u}} = (0, 0)$$

which corresponds to a speed of $7ms^{-1}$ and a distance of $5m$ between the center of the rear axle and the edge of the road. Linearize the system around its operating point.

5) We want our setpoints w_1 and w_2 to correspond to the distance of the edge of the road and the speed of the car. Design a controller by pole placement. All the poles in the interval $[-2, -1]$.

6) Test the behavior of your controller in the case where the car is turning around the polygon. Note that the system being controlled (in other words the turning car) by the controller is different than that it thinks it is controlling (i.e. a car in a straight line). You should obtain a behavior similar to that illustrated by Figure 6.8.

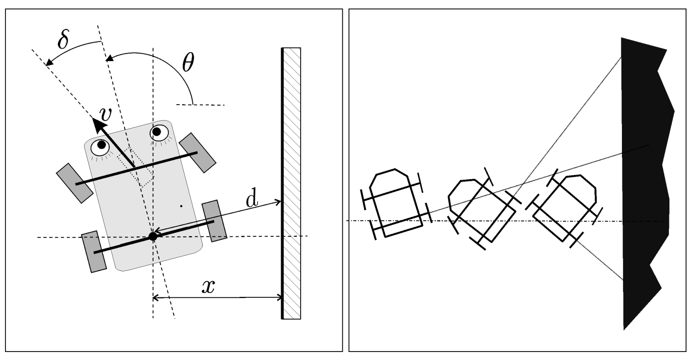


Figure 6.9: The controller imagines a simpler world than reality, with a straight vertical wall

---

Exercise 6.9.– *Follow the car*

See the correction video at tBeIozR3J8M

First part.– (proportional and derivative control). Let us consider a second-order system described by the differential equation $\ddot{y} = u$, where u is the input and y the output.

1) Give a state representation for this system. The chosen state vector is given by $\mathbf{x} = (y, \dot{y})$.

2) We would like to control this system using a proportional and derivative control of the type:

$$u = k_1(w - y) + k_2(\dot{w} - \dot{y}) + \ddot{w}$$

where w is a known function (for instance a polynomial) called *setpoint* and which can here depend on the time t. Let us note that here, we have assumed that y and $\dot{y}$ were available for the controller. Give the differential equation satisfied by the error $e = w - y$ between the setpoint and the output y.

3) Calculate k_1 and k_2, allowing to have an error e which converges towards 0 in an exponential manner, with all the poles equal to -1.

Second part.– (control of a tank-like vehicle). Let us consider the robotic vehicle on the left hand side of Figure 6.10, described by:

$$\begin{cases} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= u_1 \\ \dot{v} &= u_2 \end{cases}$$

where v is the speed of the robot, θ its orientation and (x, y) the coordinates of its center. We assume that we are capable of measuring all the state variables of our robot with great precision.

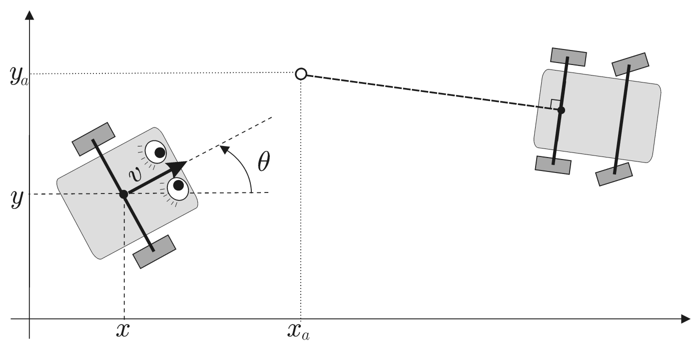


Figure 6.10: Our robot (with eyes) follows a vehicle (here a car) which has an attachment point (small white circle) to which the robot has to attach

1) Let us denote by $\mathbf{x} = (x, y, \theta, v)$ the state vector of our robot and $\mathbf{u} = (u_1, u_2)$ the vector of the inputs. Calculate $\ddot{x}$ and $\ddot{y}$ in function of $\mathbf{x}$ of $\mathbf{u}$. Show that:

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \mathbf{A}(\mathbf{x}).\mathbf{u}$$

where $\mathbf{A}(\mathbf{x})$ is a 2×2 matrix.

2) Another mobile vehicle (on the right hand side of Figure 6.10) tell us, in a wireless manner, the precise coordinates (x_a, y_a) of a virtual attachment point (in other words one that only exists mentally), fixed with respect to this vehicle, to which we need to position ourselves. This means that we want the center of the robot (x, y) to be such that $x(t) = x_a(t)$ and $y(t) = y_a(t)$. This vehicle also sends us the first two derivatives $(\dot{x}_a, \dot{y}_a, \ddot{x}_a, \ddot{y}_a)$ of the coordinates of the attachment points. Propose the expression of a control $\mathbf{u}(\mathbf{x}, x_a, y_a, \dot{x}_a, \dot{y}_a, \ddot{x}_a, \ddot{y}_a)$ which ensures us that the distance between the position (x, y) of our robot and that of the attachment point (x_a, y_a) decreases in an exponential manner. We will set the poles at -1.

Advice.– Perform a first loop as such: $\mathbf{u} = \mathbf{A}^{-1}(\mathbf{x}) \cdot \mathbf{q}$, where $\mathbf{q} = (q_1, q_2)$ is our new input. This first loop will allow you to simplify your system by making it linear. Then, proceed with a second, proportional and derivative loop.

Bibliography

[1] L. Jaulin. *Représentation d'état pour la modélisation et la commande des systèmes (Coll. Automatique de base)*. Hermès, London, 2005.

[2] L. Jaulin. Robust set membership state estimation ; application to underwater robotics. *Automatica*, 45(1):202–206, 2009.

[3] L. Jaulin and F. Le Bars. An Interval Approach for Stability Analysis; Application to Sailboat Robotics. *IEEE Transaction on Robotics*, 27(5), 2012.

[4] L. Jaulin, F. Le Bars, B. Clément, Y. Gallou, O. Ménage, O. Reynet, J. Sliwka, and B. Zerr. Suivi de route pour un robot voilier. In *CIFA 2012*, Grenoble (France), 2012.

[5] E. Walter. *Numerical Methods and Optimization ; a Consumer Guide*. Springer, London, 2014.

[6] J.P. Laumond. *La robotique mobile*. Hermès, Paris, France, 2001.

[7] E. Dombre and W. Khalil. *Robot manipulators: modeling, performance analysis and control*. ISTE Ltd., London, 2007.

[8] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. *Synchronization and Linearity. An Algebra for Discrete Event Systems*. John Wiley & Sons, New York, 1992.

[9] H.K. Khalil. *Nonlinear Systems, Third Edition*. Prentice Hall, 2002.

[10] L. Jaulin. Commande d'un skate-car par biomimétisme. In *CIFA 2010*, Nancy (France), 2010.

[11] F. Boyer, M. Porez, and W. Khalil. Macro-continuous computed torque algorithm for a three-dimensional eel-like robot. *IEEE Transactions on Robotics*, 22(4):763–775, 2006.

[12] H. Bourlès. *Systèmes linéaires ; de la modélisation à la commande*. Hermès-sciences, Paris, France, 2006.

[13] T. Kailath. *Linear Systems*. Prentice Hall, Englewood Cliffs, 1980.