Control theory

Identification,

State-space representation of linear systems

January 2, 2023

Foreword

This course is intended for students in the third year of Polytech Angers, option *Automation and Computer Engineering*. It is in the continuity of the "Modelling and simulation" module (3rd year), and as a preamble to the Advanced control modules (in 4th and 5th year).

It is assumed that the students have taken an introductive course about control theory in their previous years of study. In particular, the following notions on linear systems in continuous time are considered to be assimilated by the student :

- input-output representations : impulse response and transfer function ;
- dynamic and static accuracy ;
- conditions and criteria for stability.

Contents

1 Introduction **1**

1.1 Introduction 1

2 Identification of continuous systems **3**

2.1 Introduction 3

2.1.1 Position of the problem 3

2.1.2 Reminders on continuous and discrete time models of linear systems 4

2.1.3 Approach to identification 5

2.2 Characterisation 5

2.2.1 Characterisation using a step response 5

2.2.2 Characterization using a random signal: quotient of instrumental determinants (QID) test 8

2.3 Estimation using a step response 9

2.3.1 Static gain 9

2.3.2 First order system 9

2.3.3 2nd order system 10

2.3.4 Systems of order greater than 2, *Strejc* method 13

2.4 Estimation using a response to a random signal 13

2.4.1 Ordinary least squares (offline) method 15

2.4.2 Recursive least squares (online) method 16

3 State representation of linear systems **19**

3.1 Introduction and background 19

3.1.1 Stability 21

3.1.2 Precision 22

3.2 Introduction to state formalism 22

3.2.1 State representation 22

3.2.2 Variables d'état 23

3.2.3 Similar state representations 24

3.3 State representation of linear systems in continuous time 25

3.3.1 From a state representation to an input-output representation 25

3.3.2 From an input-output representation to a state representation 26

3.3.3 Simulation from the state representation 27

3.3.4 Solution of the state equations 28

3.3.5 Stability 28

3.3.6 Controllability 29

3.3.7 Observability 30

3.4 State representation of linear systems in discrete time 30

3.4.1 Solution of the state equations 31

3.4.2 Stability 31

3.4.3 Controllability 32

3.4.4 Observability 32

3.4.5 From an input-output representation to a state representation 32

3.4.6 Study of continuous systems by of a computer: sampled systems 33

3.5 State feedback control 36

3.5.1 Pole placement control 37

3.5.2 Static regime of the looped system . . . 40
3.5.3 Adding an integral effect . . . 42
3.6 State observer . . . 42
3.6.1 Principle of the state observer . . . 43
3.6.2 Synthesis of the matrix L_{obs} . . . 44
3.6.3 Separation principle . . . 45

Chapter 1

Introduction

1.1 Introduction

Control theory is a science that deals with the modelling, analysis, identification and control of dynamic systems. A common goal is to achieve control of systems: for example, to tune inputs so that outputs have pre-determined values (set point), while satisfying certain criteria (sensitivity or insensitivity to certain inputs, precision, ...). To do this, we use a *dynamic model* of the systems, i.e. a mathematical description (for example, differential equations) of the dynamic behaviour of the system.

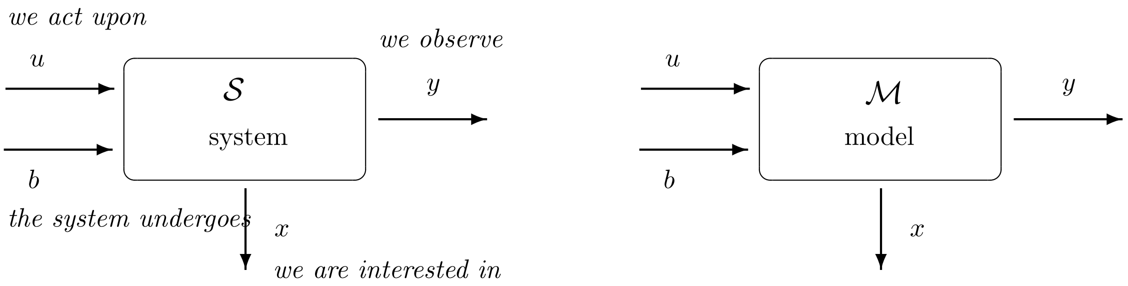


Figure 1.1: Scheme of a system and its model

There are two possible approaches to get a model :

- from the knowledge of the phenomena involved (laws of Physics, Chemistry, Biology,...) we build the model by writing conservation equations (mass, moment, energy,...), balance equations... This activity is generally called *modelling*, and one then obtains so-called ”knowledge” or ”phenomenological” models.
- From experimental data (measurements of certain quantities within the system, usually subjected to selected excitations), a mathematical model is extrapolated. This activity is rather called *identification*, and we obtain *representational models* or *behavioural models.*

The first part of this course will be dedicated to identification techniques (a course was dedicated to modelling in the first semester).

The history of automatic control began a little before 1930 with the development of the theory of so-called frequency control. The models manipulated are of the input-output type (in the diagram above, only the signals u, b and y are considered) and often take the form of transfer functions. From 1960 onwards, the notion of state emerges (by defining in the above diagram, the x signal) and many techniques have been developed using the state representation as a model of the system.

The second part of this course will be dedicated to the state representation of systems. The second part of the course will be dedicated to the state representation of systems, including the state feedback control and the state reconstruction.

- A system (and its model) is said to be *deterministic* if for each input $u(t)$, there is only one possible output $y(t)$. On the contrary, in a non-deterministic or *stochastic* model, there are several possible outputs, each of them being assigned a certain probability.
- A system (and its model) is said to be *linear*, if the superposition principle applies:

$$\mathcal{S}\left(k_1 \cdot u_1(t) + k_2 \cdot u_2(t)\right) = k_1 \cdot \mathcal{S}(u_1(t)) + k_2 \cdot \mathcal{S}(u_2(t))\ .$$

- A system (and its model) is said to be *stationary* if the relations between the input and the output are independent of time (the characteristics of the system are invariant in time).
- A system (and its model) is said to be *causal* if the value of the output at a time t_0, $y(t_0)$, does not depend on the values of the values of $u(t)$ for $t > t_0$ (all physical systems are causal).

In this course we restrict our attention to deterministic, linear, stationary and causal systems.

Chapter 2

Identification of continuous systems

2.1 Introduction

Identification consists in determining a mathematical description (a model) of a process from experimental data.

2.1.1 Position of the problem

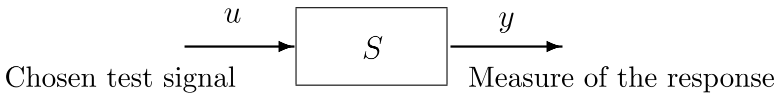
We are interested here only in linear models: this assumption is "classical" and not very restrictive (a non-linear system considered around an operating point approaches well as a linear system).
There are two classes of linear models:

- the *nonparametric models* : for example, the impulse response which establishes a sequence of points (theoretically infinite) corresponding to the response of the system to an impulse (physically impossible to generate perfectly),

- the *parametric models* : differential equations (in continuous time), difference equations (in discrete time), transfer functions with a finite number of parameters.

We restrict ourselves here to parametric models [1]. Moreover, we limit ourselves to mono-variable systems (a single input and a single output).

The problem can then be summarised as follows



$\mathcal{S}$ linear and single-input single-output

Choice of a parametric model of S?

[1] The reader can refer for example to [Borne et al.] for an account of non-parametric identification methods such as deconvolution and correlation methods

2.1.2 Reminders on continuous and discrete time models of linear systems

	DISCRETE TIME	CONTINUOUS TIME
Convolution	$y(k) = (h * u)(k)$ $= \sum_{i=0}^{k} h(i)u(k-i)$	$y(t) = (h * u)(t)$ $= \int_0^t h(\tau)u(t-\tau)d\tau$
Transfer Funct.	$H(z) = \frac{b_n z^n + b_{n-1}z^{n-1} + \ldots + b_1 z + b_0}{z^n + a_{n-1}z^{n-1} + \ldots + a_1 z + a_0}$ $\mathcal{Z}[h(k)] = H(z)$ $Y(z) = H(z)U(z)$	$H(s) = \frac{b_n s^n + b_{n-1}s^{n-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0}$ $\mathcal{L}[h(t)] = H(s)$ $Y(s) = H(s)U(s)$
Diff. Eq.	$y(k+n) + a_{n-1}y(k+n-1) + \ldots + a_0 y(k)$ $= b_n u(k+n) + b_{n-1}u(k+n-1) + \ldots + b_0 u(k)$ I.C. $y(0), y(1), \ldots, y(n-1)$	$y^{(n)} + a_{n-1}y^{(n-1)} + \ldots + a_1\dot{y} + a_0 y$ $= b_n u^{(n)} + b_{n-1}u^{(n-1)} + \ldots + b_1\dot{u} + b_0 u$ I.C. $y(0^+), \dot{y}(0^+), \ldots, y^{(n-1)}(0^+)$

Remarks 1

- *As stated in the previous subsection, only parametric models are considered, and the impulse response (recalled above in the convolution representation) is not mentioned again in the following.*

- *In the literature, it is common to distinguish in the transfer function the degree of the polynomial in the numerator (with coefficients noted b_i) from that of the polynomial in the denominator (with coefficients noted a_i). They are often called respectively the* degree *and the* order *of the system. The order is higher (often strictly with b_n=0 above) than the degree to express the causality of the system. Here, we consider the most general case where the degree is possibly equal to the order n.*

- *It is recalled that it is possible to switch from one representation to another by means of an appropriate transformation. In particular, if a model is available in the form of a continuous time transfer function, it is possible to derive a differential equation (using an inverse Laplace transform). Conversely, one can go from a differential equation to a transfer function.*

- *In many cases, a discrete time system corresponds to a sampled and blocked continuous system studied from a computer (as schematised in figure 2.1). This processing chain will be detailed in this material when we study the state representation of sampled systems.*

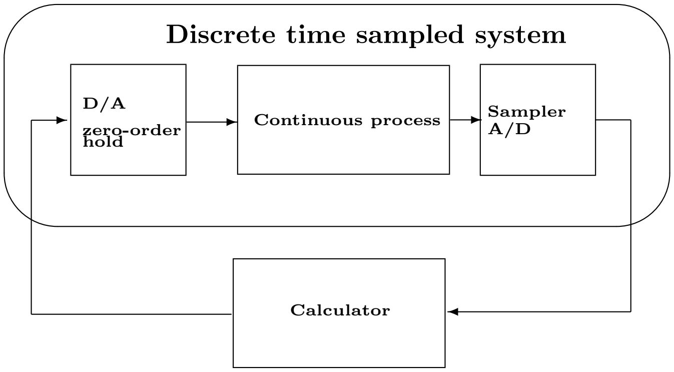


Figure 2.1: Continuous system studied as a discrete-time system by means of a calculator

2.1.3 Approach to identification

There are two main stages in the identification work:

1. the first is to fix the form of the equations: this is the *characterisation* discussed in the section 2.2,

2. the second is to find the numerical values of the coefficients involved in these equations: this is the *estimation of the parameters* discussed in the sections 2.3 and 2.4. These numerical values are determined so that the behaviour of the model is as close as possible to that of the system: this ”closeness” is measured using a ”criterion”.

2.2 Characterisation

The purpose of the characterisation is to fix the structure of the model. We are interested here only in parametric models of monovariable linear systems and we have recalled in subsection 2.1.2 the possible models. From these, it appears that the characterisation problem then boils down to choosing a value for n involved in the transfer function and the differential equations.

2.2.1 Characterisation using a step response

We recall here some elements, often well known, which allow to characterize a system from observations on its response to a step.

First order system The step response of a first-order system (example shown in Figure 2.2) is characterised by a non-zero derivative at the time the system starts to respond. There is also a lack of overshooting of the final value, but this does not identify a system of order 1 since systems of order higher than 1 can also exhibit this characteristic (see following paragraphs).

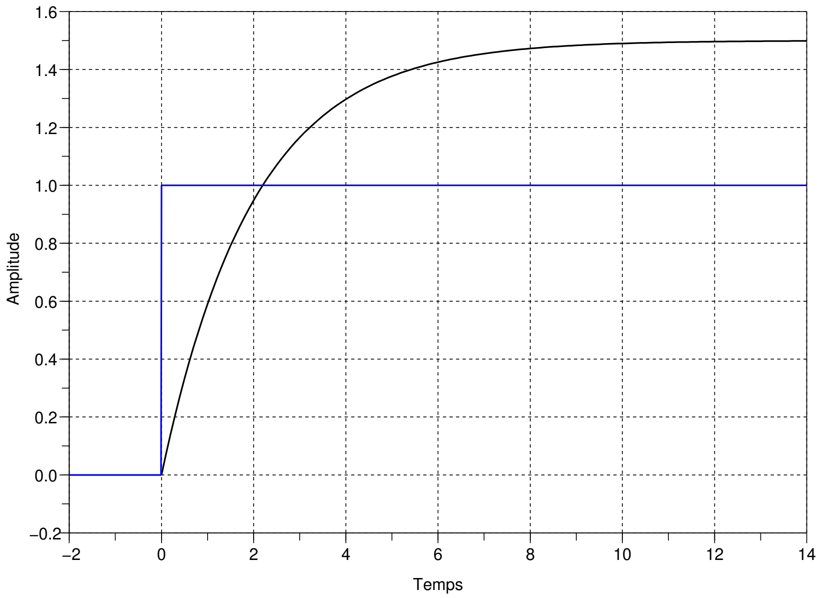


Figure 2.2: Step response of a first order system

2nd order system The step response of a second order system (two examples are shown in Figure 2.3) is characterised by zero derivative at the moment the system starts to respond. In addition, it is possible (but not necessary!) that there is an overshoot of the final value.

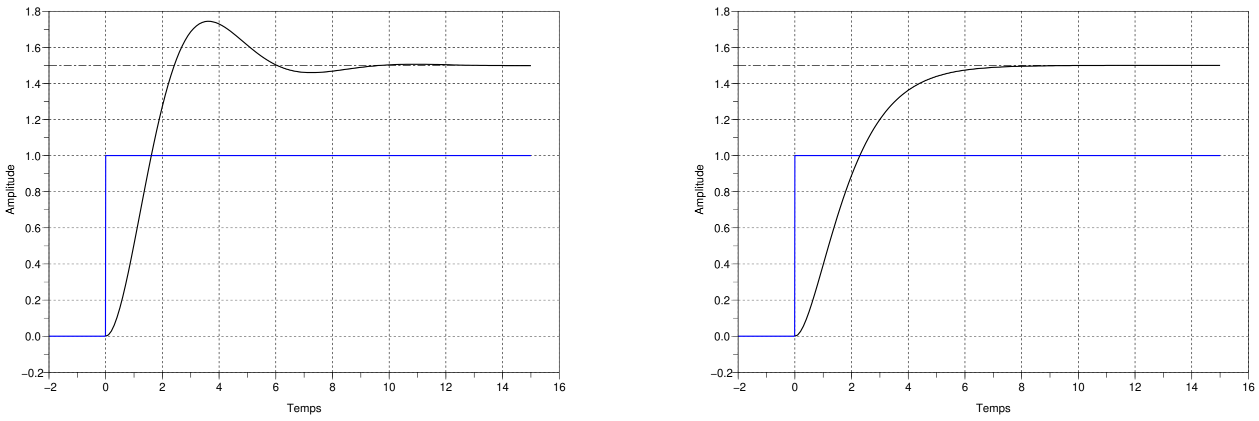


Figure 2.3: Step responses of second order systems

Systems of order greater than 2 If the order of the system is strictly greater than two, then it is rather complicated to determine the order from the step response because the shape of this curve can be very diverse. Indeed, the response of such a system can be seen as the superposition of the responses of subsystems of order one and two. The response of the system is generally marked by some of its poles which are said to be *dominant* and which correspond to high time constants or low damping (slow or very oscillatory temporal response). Two approaches can be considered:

- approach the system as a first or second order system by assimilating it to the dominant pole(s),

- use a more "elaborate" characterisation method to accurately determine the order of the system (in this manuscript, a method is presented in paragraph 2.2.2 which can be used for this purpose).

Ideal delay There are systems (usually with "slow dynamics", such as certain thermal or hydraulic systems) for which there is a delay between the input excitation time and the output reaction time of the system. We have plotted on the Figure 2.4 the step response of a delayed first order system.

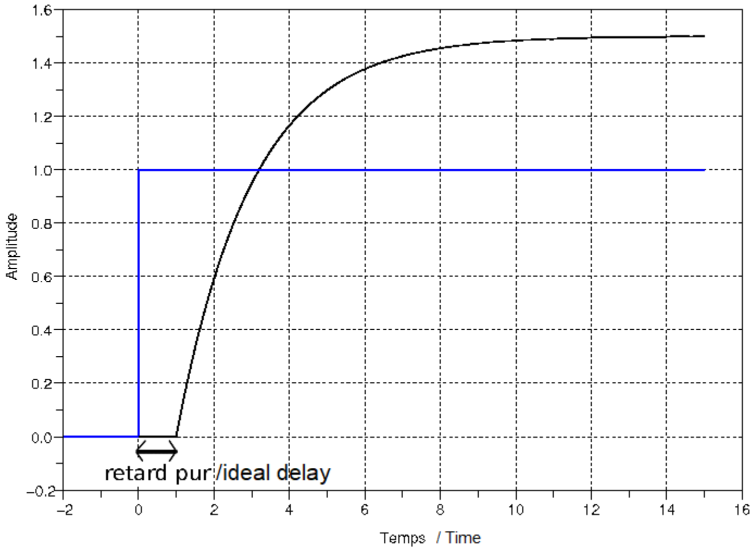


Figure 2.4: Step response of a delayed first order system

If we note τ the value of the observed ideal delay, and r is the number of discrete steps translating the pure delay τ, we obtain the following transfer functions for a system of order n:

DISCRETE TIME	CONTINUOUS TIME
$H(z) = \frac{b_n z^n + b_{n-1} z^{n-1} + \ldots + b_0}{z^n + a_{n-1} z^{n-1} \ldots + a_0} \mathbf{z^{-r}}$	$H(s) = \frac{b_n s^n + b_{n-1} s^{n-1} + \ldots + b_0}{s^n + a_{n-1} s^{n-1} + \ldots + a_0} \mathbf{e^{-s\tau}}$

For a sampled system, we take $r = \lfloor \tau/\Delta \rfloor$ where Δ denotes the sampling period and $\lfloor\cdot\rfloor$ is the default integer part. Considering the transfer function in discrete time, we can see that:

$$\begin{aligned} H(z) &= \frac{b_n z^n + b_{n-1} z^{n-1} \ldots + b_0}{z^n + a_{n-1} z^{n-1} + \ldots + a_0} \mathbf{z^{-r}} \\ &= \frac{b_n z^n + b_{n-1} z^{n-1} + \ldots + b_0}{z^{n+r} + a_{n-1} z^{n+r-1} + \ldots + a_0 z^r} \\ &= \frac{\beta_{n+r} z^{n+r} + \beta_{n+r-1} z^{n+r-1} + \ldots + \beta_0}{z^{n+r} + \alpha_{n+r-1} z^{n-1+r} + \ldots + \alpha_1 z + \alpha_0}, \end{aligned}$$

with

- $\beta_{n+r} = \ldots \beta_{n+1} = 0$ and $\beta_n = b_n$, …, $\beta_0 = b_0$,
- $\alpha_{n+r-1} = a_{n-1}$, …, $\alpha_r = a_0$ and $\alpha_{r-1} = \ldots = \alpha_0 = 0$.

From a formal point of view, a system of order n with r ideal delays can be studied as a system of order $n + r$ (with some zero coefficients).

2.2.2 Characterization using a random signal: quotient of instrumental determinants (QID) test

In the previous paragraphs, it was recalled how to recognise first and second order systems from their step response. This procedure leaves room for the "engineer's appreciation" since he has to interpret curves. On the contrary, we present a test which "mechanically" allows us to determine the value of n, i.e. the order of a system (or the sum of the order and the number of possible pure delays). Furthermore, this test does not require the use of a step as input, but rather a random signal.

The quotient of instrumental determinants (QID) test is essentially based on the rank conditions of a matrix, called *information matrix*, containing the pairs of input-output measurements. For this purpose, it is considered that N measurements of the applied input $u(i)$ and the output response $y(i)$ have been made, $i = 1, \dots N$, the information matrix Q_i at step i is written :

$$Q_i = \frac{1}{N}\sum_{k=i}^{N-i}\begin{pmatrix} u(k) \\ u(k+1) \\ u(k-1) \\ u(k+2) \\ \vdots \\ u(k-i+1) \\ u(k+i) \end{pmatrix}\begin{pmatrix} y(k+1) & u(k+1) & \dots & y(k+i) & u(k+i) \end{pmatrix}. \tag{2.1}$$

Note that Q_i is a square matrix of dimension $2i \times 2i$. The information matrix Q_{i+1} at step $i+1$ is constructed from the matrix Q_i :

$$Q_{i+1} = \frac{1}{N}\sum_{k=i+1}^{N-(i+1)}\begin{pmatrix} \mathbf{u(k)} \\ \mathbf{u(k+1)} \\ \mathbf{u(k-1)} \\ \mathbf{u(k+2)} \\ \vdots \\ \mathbf{u(k-i+1)} \\ \mathbf{u(k+i)} \\ u(k-i) \\ u(k+i+1) \end{pmatrix}\begin{pmatrix} \mathbf{y(k+1)} & \mathbf{u(k+1)} & \dots & \mathbf{y(k+i)} & \mathbf{u(k+i)} & y(k+i+1) & u(k+i+1) \end{pmatrix},$$

with Q_{i+1} a square matrix of dimension $(2i+2) \times (2i+2)$.

The structure of the information matrix is "nested", because, for example, the matrix Q_2 contains all the elements of the matrix Q_1 :

$$Q_2 = \frac{1}{N}\sum_{k=2}^{N-2}\begin{pmatrix} \mathbf{u(k)y(k+1)} & \mathbf{u(k)u(k+1)} & u(k)y(k+2) & u(k)u(k+2) \\ \mathbf{u(k+1)y(k+1)} & \mathbf{u(k+1)u(k+1)} & u(k+1)y(k+2) & u(k+1)u(k+2) \\ u(k-1)y(k+1) & u(k-1)u(k+1) & u(k-1)y(k+2) & u(k-1)u(k+2) \\ u(k+2)y(k+1) & u(k+2)u(k+1) & u(k+2)y(k+2) & u(k+2)u(k+2) \end{pmatrix}$$

The *quotient of instrumental determinants* (QID) is given by

$$QID(i) = \frac{|Q_i|}{|Q_{i+1}|} \tag{2.2}$$

For each value of i, the procedure for determining the order

- constructs the matrices Q_i and Q_{i+1} defined by (2.1),
- and evaluates the quotient of instrumental determinants defined by (2.2),
- finally, **the value of n (order of the system) is the value of i for which the absolute value of the ratio $QID(i)$ increases rapidly for the first time.**

2.3 Estimation using a step response

2.3.1 Static gain

The static gain is the quotient between the range of the response and the range of the input step (see for example Figure 2.5). The range of the response is measured as the difference between the initial value of the output y_{init} (before the system starts to respond to the step) and the final value of the response y_{end}.

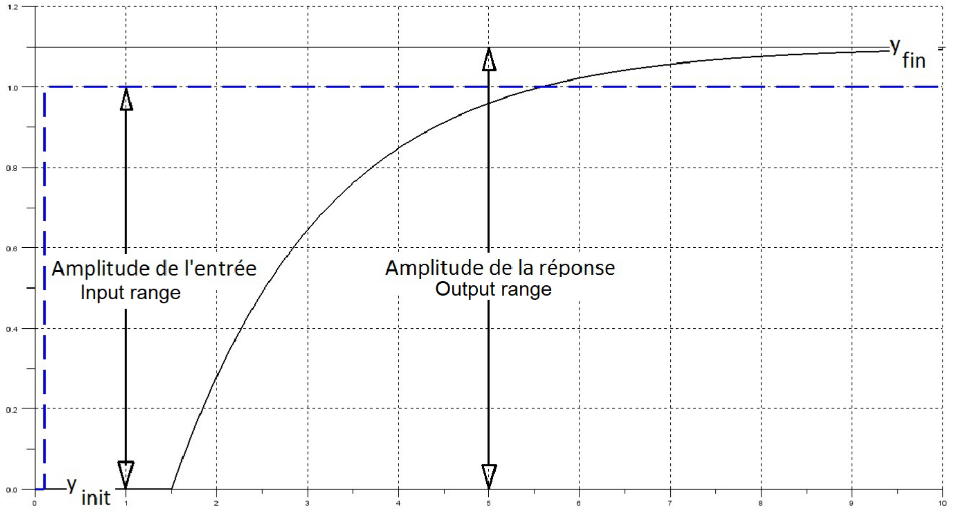


Figure 2.5: Measurement of the amplitudes for the estimation of the static gain.

2.3.2 First order system

Recall that the transfer function of a first order is typically written as follows.

- In continuous time:

$$H(s) = K\frac{1}{1+Ts}.$$

- In discrete time (sampled process):

$$\begin{array}{rcl} H(z) & = & \frac{z-1}{z}\mathcal{Z}\left[\frac{H(p)}{p}\right] \\ & = & K\frac{1-z_0}{z-z_0} \end{array}$$

with $z_0 = e^{-\frac{\Delta}{T}}$ (the sampling period Δ to be chosen with respect to: $0,25T < \Delta < 1,25T$).

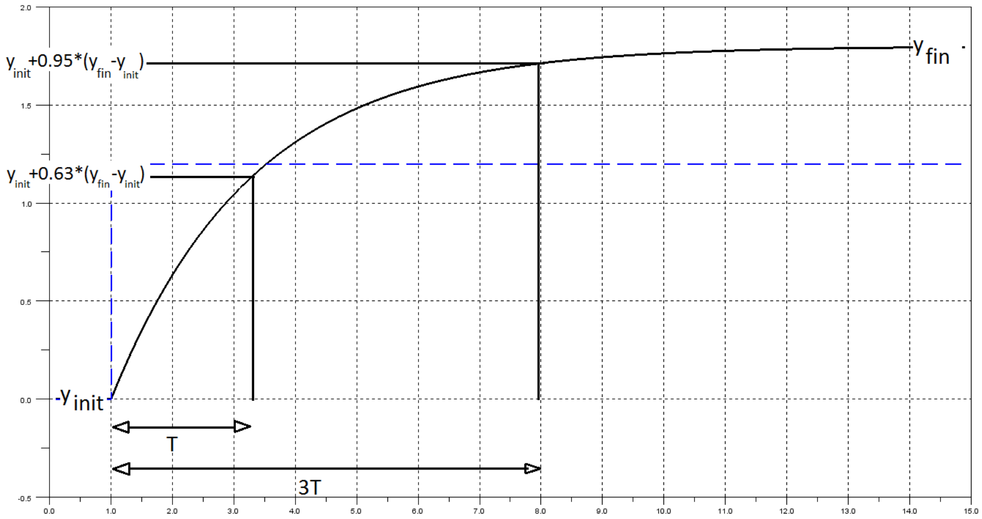


Figure 2.6: Estimation using step response

In the previous paragraph, it was specified how to estimate the static gain K. The other parameter T, called the time constant, can be estimated from readings of the index response as, for example, shown in Figure 2.6.

2.3.3 2nd order system

Recall that the transfer function of a second order system in continuous time is typically written

$$H(s) = K\frac{1}{1 + 2\xi\frac{s}{\omega_n} + (\frac{s}{\omega_n})^2}.$$

with

ξ	damping ratio
ω_n	natural frequency

For the estimation of these parameters, two cases are distinguished:

- if the system is resonant and damped ($0 < \xi < 1$), we obtain a step response of the type shown in the figure 2.7, and by noting the overshoot d and the rise time t_m (or peak time t_{pic}) on the curve, the parameters ξ and ω_n can be estimated using the following relationships:

Overshoot (as a percentage of the response amplitude)	$D_{\%} = \frac{d\times 100}{\text{response amplitude}} = 100e^{-\pi\xi/\sqrt{1-\xi^2}}$
Rise time	$t_m = \frac{1}{\omega_n\sqrt{1-\xi^2}}(\pi - \arccos\xi)$
Peak time	$t_{pic} = \frac{\pi}{\omega_n\sqrt{1-\xi^2}}$

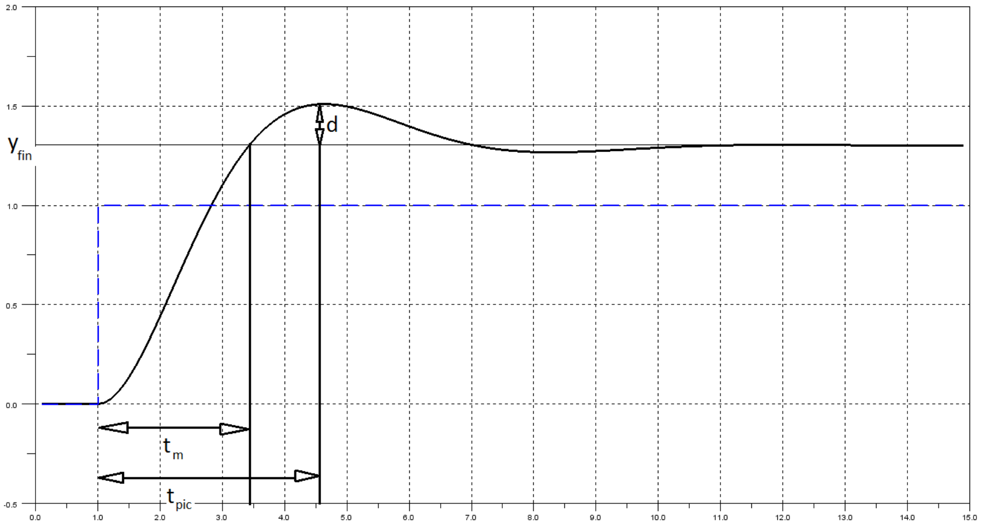


Figure 2.7: Step response of a damped second order resonant system.

- if the system is aperiodic ($\xi \geq 1$), we obtain a step response of the type represented on the figure 2.8.

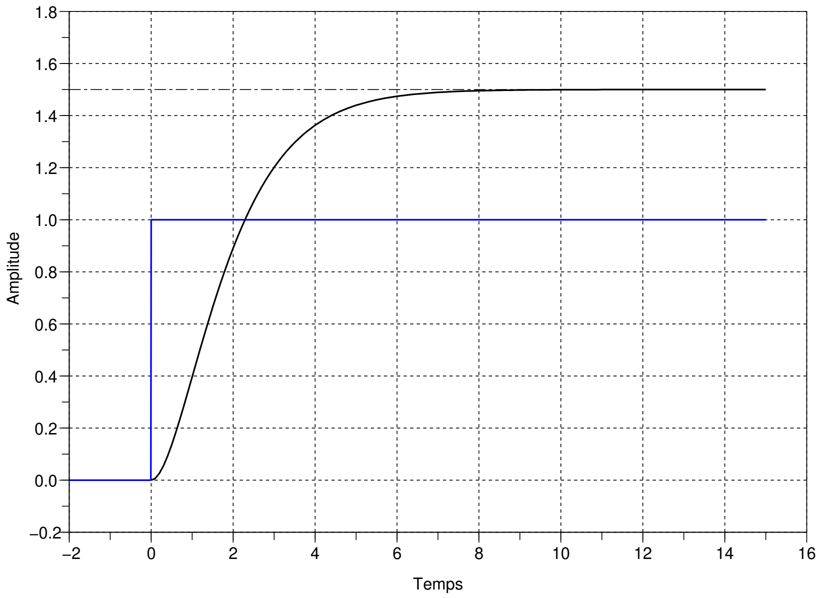


Figure 2.8: Step response of an aperiodic second order system.

The transfer function can be written as:

$$H(s) = \frac{K}{(1+T_1 s)(1+T_2 s)},$$

with

$$\xi = \frac{1}{2}\frac{T_1+T_2}{\sqrt{T_1T_2}} \text{ and } \omega_n = \frac{1}{\sqrt{T_1T_2}}$$

The tangent at the inflection point can be drawn to obtain an approximate value for T_1 and T_2 (see Figure 2.9), and thus, we can estimate ξ and ω_n.

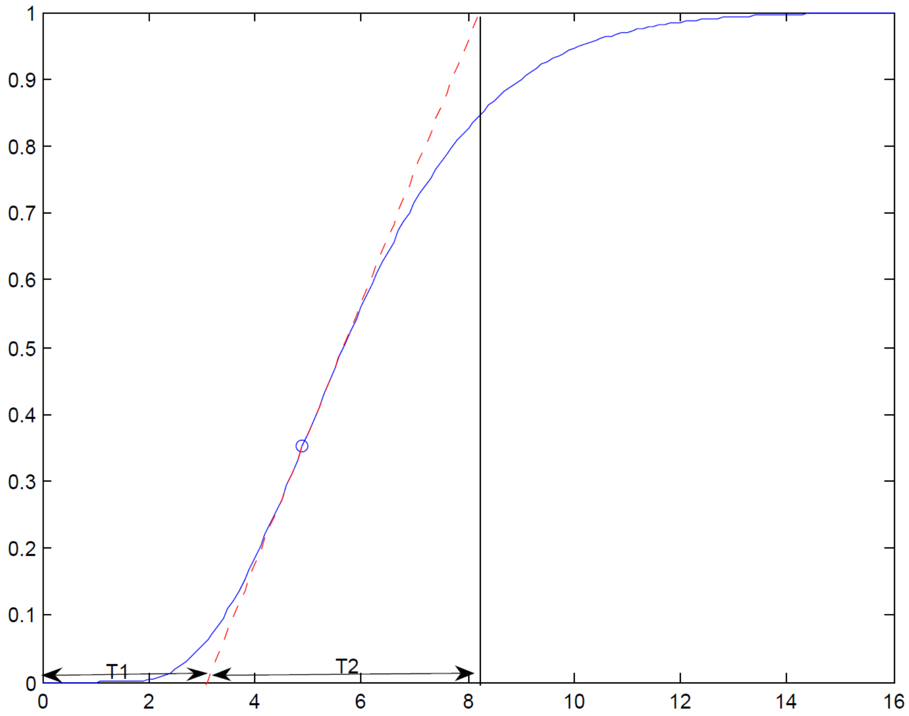


Figure 2.9: Method to obtain the values of T_1 and T_2 for an aperiodic second order.

Recall that in discrete time, the transfer function of a second order is typically written

$$H(z) = \frac{b_1 z + b_0}{z^2 + a_1 z + a_0}$$

with the sampling period chosen such that $0.25 < \omega_n\Delta < 1.25$ for a sampled process.
We have the following relationships:

if $\xi < 1$ noting $\alpha = e^{-\xi\omega_n\Delta}$, $\omega_p = \omega_n\sqrt{1-\xi^2}$	if $\xi \geq 1$ noting $z_1 = e^{-\Delta/T_1}$, $z_2 = e^{-\Delta/T_2}$,
$a_0 = \alpha^2$ $a_1 = -2\alpha\cos(\omega_p\Delta)$ $b_0 = \alpha^2 + \alpha\left[\xi\frac{\omega_n}{\omega_p}\sin(\omega_p\Delta) - \cos(\omega_p\Delta)\right]$ $b_1 = 1 - \alpha\left[\xi\frac{\omega_n}{\omega_p}\sin(\omega_p\Delta) + \cos(\omega_p\Delta)\right]$	$a_0 = z_1z_2$ $a_1 = -(z_1+z_2)$ $b_0 = K(z_1z_2 - \frac{T_1z_2 - T_2z_1}{T_1-T_2})$ $b_1 = K(\frac{1-T_1z_1-T_2z_2}{T_1-T_2})$

2.3.4 Systems of order greater than 2, *Strejc* method

As with characterisation, it can be complicated to estimate a system of order greater than 2 using only its step response (as this can take many different forms). In particular, if this step response does not show any overshoot, then there are methods such as the one of *Strejc* and the one of *Broïda* which allow to realize the estimation. They can be seen as generalisations of the method proposed above for an aperiodic second order. We explain here briefly the *Strejc* method.

This method can be applied to systems whose step response does not exhibit overshoot. They are identified with a transfer function of the form:

$$H(s) = \frac{K}{(1+Ts)^n}$$

The parameters to be estimated are therefore:

- the static gain K (referring to the paragraph above),
- the order n and the time constant T (estimated using the method below).

The method can be broken down into the following steps:

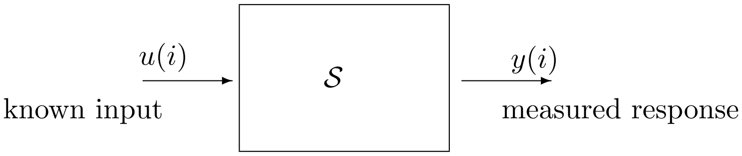
1. The tangent at the inflection point is drawn to determine two values T_1 and T_2 (as shown in Figure 2.9)
2. The order n is deduced of the value of $\frac{T_1}{T_2}$ using the table below. Between two lines of the table, we choose the smallest value of n.
3. The time constant T is determined from the value of $\frac{T_1}{T}$ or $\frac{T_2}{T}$ in the table below.

n	$\frac{T_1}{T}$	$\frac{T_2}{T}$	$\frac{T_1}{T_2}$
3	0,8	3,7	0,22
4	1,42	4,46	0,32
5	2,10	5,12	0,41
6	2,81	5,70	0,49

2.4 Estimation using a response to a random signal

In order to estimate the parameters of a model using a response to a random signal, there are two types of procedures, so-called ”off-line” and ”on-line”, which can be described as follows.

I) **Offline procedure**
The following experiment is considered:



Continuous System + ZOH + Converters

$i = 1, \dots, N$ (N measures)

u is chosen to excite the process in its full bandwidth.
Typically, a Pseudo-Random Binary Sequence (PRBS) is applied, the spectrum of which is close to that of white noise (energy equispaced over all frequencies).

We wish to build a $\mathcal{M}$ model from these data :

(a) A structure was chosen for $\mathcal{M}$ during the characterisation. This will be a difference equation (or equivalently the transfer function in $\mathcal{Z}$) :

$$y_M(i+n) = -a_{n-1}y_M(i+n-1) - \ldots - a_0 y_M(i) + b_n u(i+n) + \ldots + b_0 u(i) \tag{2.3}$$

(b) We now wish to establish the vector of parameters

$$\theta = \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \\ b_0 \\ \vdots \\ b_n \end{pmatrix} \tag{2.4}$$

such that the output of the model $y_M(i)$ is close to $y(i)$ for $i = 1, \ldots, N$, if we apply as input to the model $u(i)$.

In order to qualify the quality of the parameters chosen for the model, we will use a criterion J to minimize. It is defined from the output error ε:

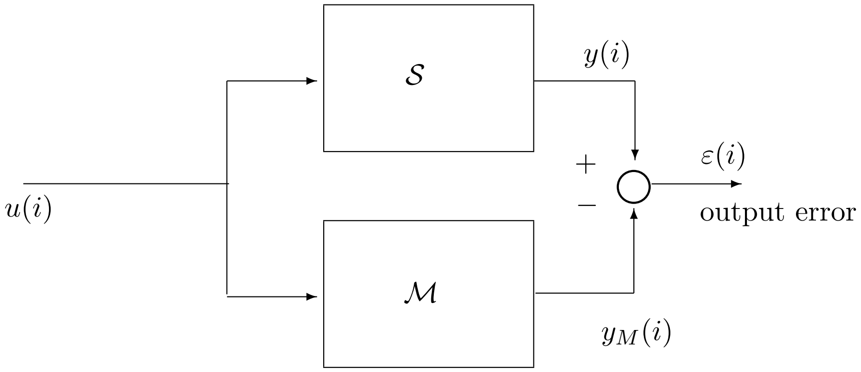


Figure 2.10: Output error

And this criterion J is therefore a function of the parameters:

$$J(\theta) = f(\varepsilon)\,.$$

In paragraph 2.4.1, we will present such an off-line estimation procedure called *ordinary least squares method* and well known for its efficiency. In the literature, we can find other off-line methods of parametric estimation, in particular the maximum likelihood method or the pointing principle method.

I) **Online procedure**
One can also implement an "on-line" procedure for estimating the parameters. More precisely, at each time i the parameters θ are estimated iteratively from the applied inputs and the output measurements up to that time. The estimation is done in a sort of "run of the mill" manner and we then have the following diagram:

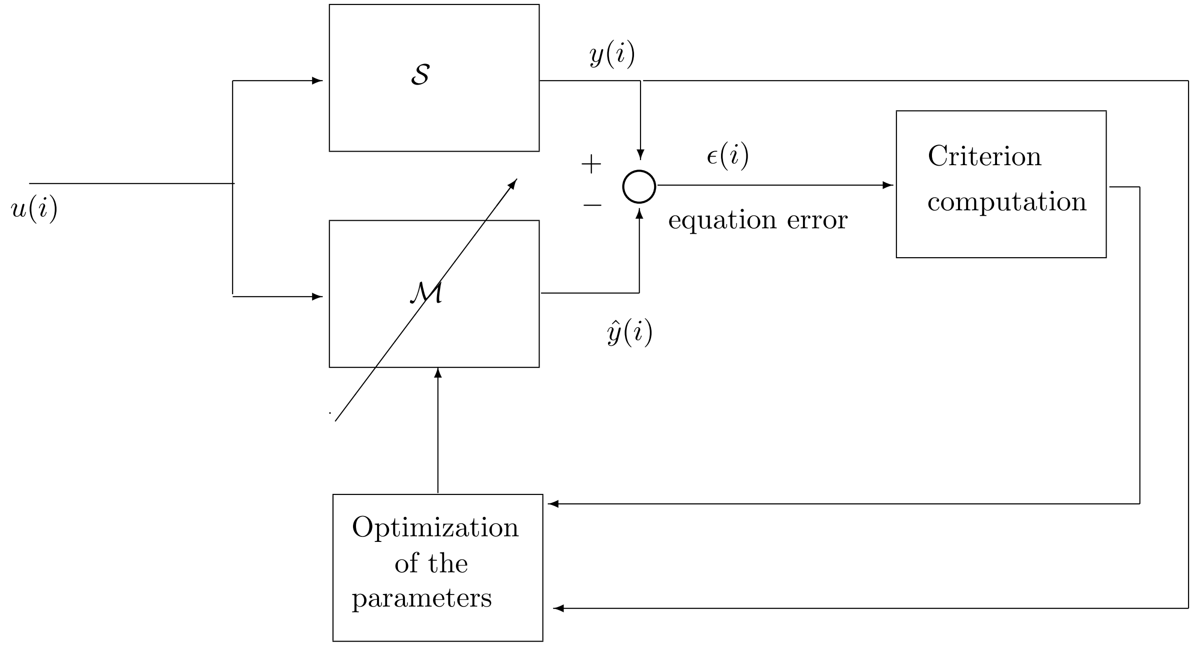


Figure 2.11: Principle for online parameter estimation

This is the only valid approach if the identification is used in adaptive control or on time-varying (non-stationary) processes: the parameter estimate can be updated as the process behaviour evolves.
In paragraph 2.4.2, the *recursive least squares method* will be presented to illustrate this approach.

2.4.1 Ordinary least squares (offline) method

Suppose we have N pairs of measurements (inputs $u(i)$, outputs $y(i)$) of the process. Ideally, these data should be such that

$$y(i) = \hat{y}(i)$$

with $\hat{y}(i)$ the output estimated by

$$\hat{y}(i+n) = -a_{n-1}y(i+n-1) - \ldots - a_0 y(i) + b_n u(i+n) + \ldots + b_0 u(i),$$

and $\theta^\top = \begin{pmatrix} a_0 & \ldots & a_{n-1} & b_0 & \ldots & b_n \end{pmatrix}$ the estimated parameters.

More realistically, we have

$$y(i) = \hat{y}(i) + \epsilon(i)$$

where $\epsilon(i) = y(i) - \hat{y}(i)$ is called the *equation error*. Note that for the output error, $\varepsilon(i)$, the output $y_M(i)$ is calculated using the previous outputs of the model $y_M(i-1)$, …, $y_M(i-n)$, while for the equation error, $\epsilon(i)$, the estimated output $\hat{y}(i)$ is calculated using the previous measurements $y(i-1)$, …, $y(i-n)$. For the set of N measurements, we can group the equations in the matrix form :

$$\underbrace{\begin{pmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ \\ \vdots \\ y(N) \end{pmatrix}}_{Y} = \underbrace{\begin{pmatrix} -y(1) & \ldots & -y(n) & u(1) & \ldots & u(n+1) \\ -y(2) & \ldots & -y(n+1) & u(2) & \ldots & u(n+2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -y(N-n) & \ldots & -y(N-1) & u(N-n) & \ldots & u(N) \end{pmatrix}}_{\Phi} \underbrace{\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \\ b_0 \\ \vdots \\ b_n \end{pmatrix}}_{\theta} + \underbrace{\begin{pmatrix} \epsilon(n+1) \\ \epsilon(n+2) \\ \vdots \\ \\ \vdots \\ \epsilon(N) \end{pmatrix}}_{\epsilon}$$

To estimate θ, we choose a quadratic criterion that represents the sum of the squares of the $\epsilon(i)$ errors:

$$\begin{aligned}
J(\theta) &= \sum_{i=n+1}^{N} \epsilon^2(i) \\
&= \epsilon^\top \cdot \epsilon \\
&= (Y - \Phi\theta)^\top \cdot (Y - \Phi\theta) \\
&= (Y^\top - \theta^\top \Phi^\top) \cdot (Y - \Phi\theta) \\
&= Y^\top Y - Y^\top \Phi\theta - \theta^\top \Phi^\top Y + \theta^\top \Phi^\top \Phi\theta
\end{aligned}$$

We assume that the matrix $\Phi^\top\Phi$ is invertible[2]. The expression of $J(\theta)$ can then be put in the form of two terms of which one is independent of θ, namely :

$$J(\theta) = \left(\theta - (\Phi^\top\Phi)^{-1}\Phi^\top Y\right)^\top \Phi^\top\Phi \left(\theta - (\Phi^\top\Phi)^{-1}\Phi^\top Y\right) + Y^\top \left(\mathbf{I} - \Phi(\Phi^\top\Phi)^{-1}\Phi^\top\right) Y$$

From this expression, we see that $J(\theta)$ is minimum if the number $\left(\theta - (\Phi^\top\Phi)^{-1}\Phi^\top Y\right)^\top \Phi^\top\Phi \left(\theta - (\Phi^\top\Phi)^{-1}\Phi^\top Y\right)^\top$ is zero, i.e.

$$\theta = \hat{\theta} = (\Phi^\top\Phi)^{-1}\Phi^\top Y \tag{2.5}$$

Methodology for applying the ordinary least squares method

The following methodology will be followed systematically:

1. We start by formulating the chosen model in the form of a difference equation identical to the equation (2.3), i.e.

   $$y_M(i+n) = -a_{n-1}y_M(i+n-1) - \ldots - a_0 y_M(i) + b_n u(i+n) + \ldots + b_0 u(i)$$

   At the cost of a possible change of variable, the difference equation is formulated so that the oldest iterate of u has index i.

2. In this equation, we identify n, and we can formulate the estimated output

   $$\hat{y}(i+n) = -a_{n-1}y(i+n-1) - \ldots - a_0 y(i) + b_n u(i+n) + \ldots + b_0 u(i).$$

3. We construct the matrices Y and Φ.

4. The parameters are estimated from the formula $\hat{\theta} = (\Phi^\top\Phi)^{-1}\Phi^\top Y$.

5. The criterion $J(\hat{\theta})$ is evaluated to validate or question the estimate.

2.4.2 Recursive least squares (online) method

The ordinary least squares identification method requires all measurements to be completed before the parameter vector $\hat{\theta}$ can be estimated.

We present here its "on-line" adaptation to re-actualise the value of the parameters after each new acquisition. In other words, we have the following problem

Having calculated a vector of parameters $\hat{\theta}_j$ from j measurements

$$\hat{\theta}_j = (\Phi_j^\top\Phi_j)^{-1}\Phi_j^\top Y_j$$

How, after a new measure $y(j+1)$,
obtain the new vector $\hat{\theta}_{j+1}$, knowing $\hat{\theta}_j$?

[2] In other words $\Phi^\top\Phi$ is of full rank and the vectors which compose its columns are not collinear. Note that this hypothesis is not verified if the values of the input $u(i)$; $i = 1, 2, \ldots, N$ are constant. This implies that the method cannot in particular be applied from a set of experimental data corresponding to an step response of the system.

This iterative method is called "recursive" because it makes it possible to calculate $\hat{\theta}_{j+1}$ without repeating the whole calculation $(\Phi_{j+1}^\top \Phi_{j+1})^{-1}\Phi_{j+1}^\top Y_{j+1}$ but expressing it using $\hat{\theta}_j$ and a correction taking into account the additional measure $y(j+1)$.

Let us note

$$\Phi_j = \begin{pmatrix} \varphi(n+1) \\ \varphi(n+2) \\ \vdots \\ \varphi(j) \end{pmatrix} = \begin{pmatrix} -y(1) & \dots & -y(n) & u(1) & \dots & u(n+1) \\ -y(2) & \dots & -y(n+1) & u(2) & \dots & u(n+2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -y(j-n) & \dots & -y(j-1) & u(j-n) & \dots & u(j) \end{pmatrix}$$

$$\Phi_{j+1} = \begin{pmatrix} \Phi_j \\ \varphi(j+1) \end{pmatrix}$$

$$Y_j = \begin{pmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(j) \end{pmatrix}$$

and

$$Y_{j+1} = \begin{pmatrix} Y_j \\ y(j+1) \end{pmatrix} .$$

The vector $\hat{\theta}_{j+1}$, taking into account the $(j+1)$-th measurement, is written:

$$\begin{aligned} \hat{\theta}_{j+1} &= (\Phi_{j+1}^\top \Phi_{j+1})^{-1}\Phi_{j+1}^\top Y_{j+1} \\ &= \left[\Phi_j^\top \Phi_j + \varphi^\top(j+1)\varphi(j+1)\right]^{-1} (\Phi_j^\top Y_j + \varphi^\top(j+1)y(j+1)) \end{aligned}$$

The idea is to make appear in this expression a relation between $\hat{\theta}_{j+1}$ and $\hat{\theta}_j$. Some (rather "heavy") manipulations allow to obtain the following result :

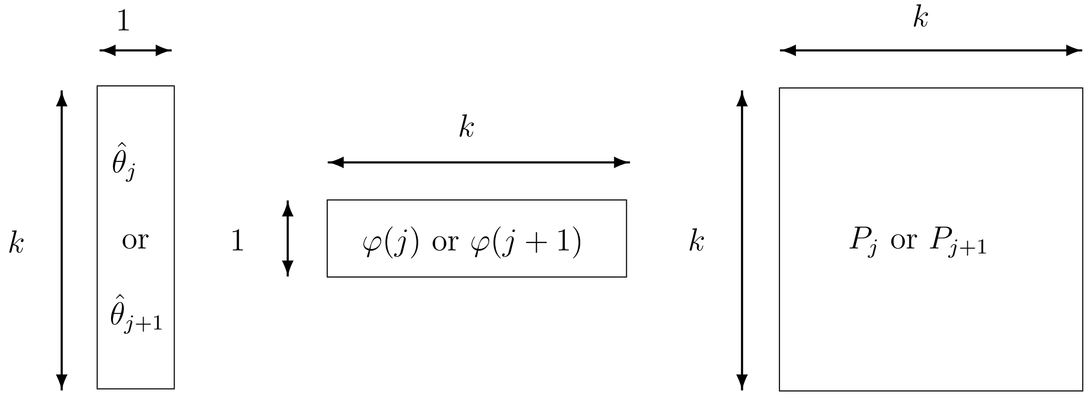
$$\hat{\theta}_{j+1} = \hat{\theta}_j + P_{j+1}\varphi^\top(j+1)\left[y(j+1) - \varphi(j+1)\hat{\theta}_j\right] \tag{2.6}$$

with

$$P_{j+1} = P_j - P_j\varphi^\top(j+1)\left[\varphi(j+1)P_j\varphi^\top(j+1) + 1\right]^{-1}\varphi(j+1)P_j \tag{2.7}$$

Remarks 2

- *Let us note $k = n + n + 1$, the dimensions of the vectors and matrices involved are :*



We can underline that $\varphi(j+1)P_j\varphi^\top(j+1)$ is a scalar, and that the algorithm does not require any matrix inversion (only products between matrices, row and column vectors).

- *In the equation (2.6), it is clear that $\hat{\theta}_{j+1}$ is deduced from the previous value $\hat{\theta}_j$ and a corrective term which takes account of the new measurement.*

- *It is possible to introduce into the equation (2.7) a forgetting factor λ which characterises the progressive forgetting of the oldest measurements :*

$$P_{j+1} = \frac{P_j - P_j\varphi^\top(j+1)\left[\varphi(j+1)P_j\varphi^\top(j+1)+\lambda\right]^{-1}\varphi(j+1)P_j}{\lambda} \tag{2.8}$$

  This factor is useful when one seeks to identify a system whose parameters vary strongly over time (a so-called non-stationary system). Typically, the value $\lambda = 0,95$ allows a fast forgetting and thus a pursuit of a strong instationnarity. Classically, $0.95 \leq \lambda \leq 0.99$.

- *As for any recurrence, it must be initialized (values of hat and P_0). In the absence of information a priori on the values of the vector of the parameters, one can take :*

$$\hat{\theta}_0 = \begin{pmatrix} 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} \textit{ and } P_0 = \alpha \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

  with α very large. If we have information on $\hat{\theta}_0$, we will take a lower value of α.

Chapter 3

State representation of linear systems

3.1 Introduction and background

In the following, concepts and results based on the state representation of linear systems are introduced. To illustrate this, the same example will be considered throughout the material.

Example 1 *We are interested in the anti-swinging system on harbour cranes such as those shown in the photo in Figure 3.1. Crane operators must manage the useful movement of the load and control unwanted load swing. This skill requires a great deal of experience. Manufacturers who use cranes extensively are looking for ways to assist in operating them, or even to automate them completely. Relieving the crane operator of the delicate task of compensating for load swinging is sure to increase productivity and allow the operator to concentrate on safety. It also means standardising movements, avoiding high mechanical stress and therefore reducing maintenance costs. Various crane manufacturers and equipment suppliers have filed patents on this issue. Port cranes are mainly targeted by these anti-swinging techniques, thanks to the good knowledge of the mechanical model (displacement with fixed cable length, 1D motion sequence, known volume and shape of the containers, ...).*

Figure 3.1: Port crane: container crane

In this example, we focus on the horizontal movement of the trolley-cable-container assembly (the lifting is considered to be stopped). More precisely, we will try to control the horizontal speed of the container while avoiding its swinging. For didactic purposes, the following simplifying assumptions are made (see figure 3.2) :

- *The trolley-cable-container assembly can be modelled as a pendulum whose attachment point moves.*
- *Consider the reference frame $(O, \overrightarrow{x}, \overrightarrow{y})$ linked to the crane frame.*
- *The trolley has a sliding connection with the crane's boom to perform the distribution movement.*
- *Given the accelerations involved, the cables, which have negligible mass, are assimilated to a single link forming an undeformable solid with the load, in pivotal connection with the carriage.*
- *The container is assimilated to a point mass placed at the centre of gravity.*
- *Friction is neglected.*

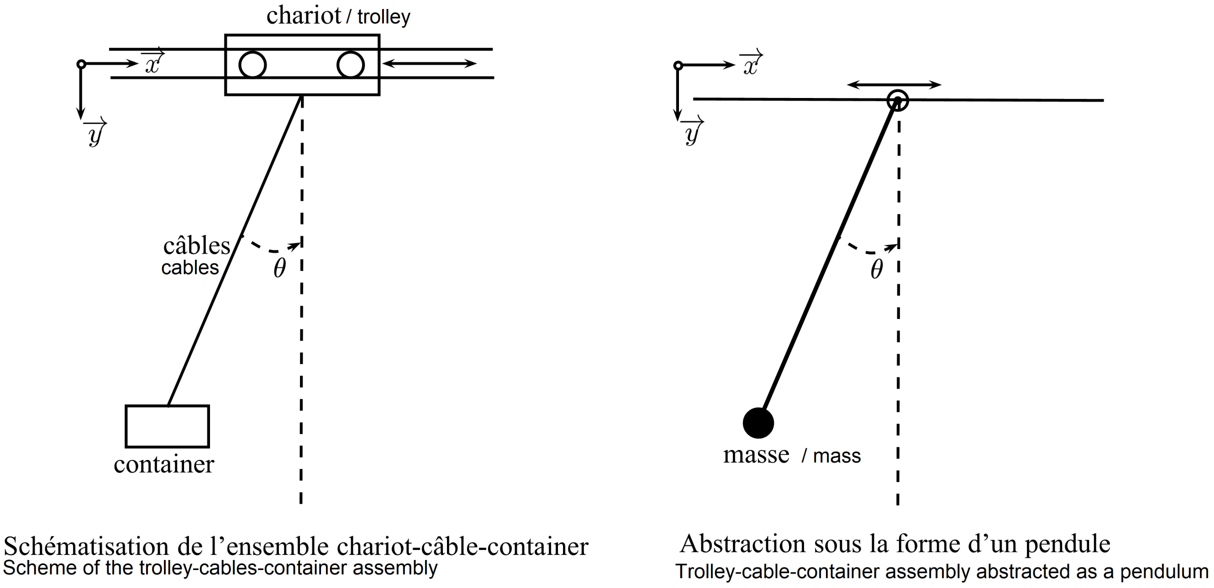

chariot / trolley
$\overrightarrow{x}$
$\overrightarrow{y}$
câbles
cables
θ
container
Schématisation de l'ensemble chariot-câble-container
Scheme of the trolley-cables-container assembly
$\overrightarrow{x}$
$\overrightarrow{y}$
θ
masse / mass
Abstraction sous la forme d'un pendule
Trolley-cable-container assembly abstracted as a pendulum

Figure 3.2: trolley-cable-container assembly designed as a pendulum

The constant quantities in the study are:

m	*container mass*
L	*length of the cables (considered as a non-deformable solid) between the point of attachment and the centre of gravity of the container*
g	*module of the acceleration of gravity*

The variables (function of time) associated with this system will be:

p_m,v_m, a_m	*respectively position, speed and acceleration of the mass (container) projected along the axis $\overrightarrow{x}$*
p_c,v_c, a_c	*respectively position, speed and acceleration of the trolley projected along the axis $\overrightarrow{x}$*
θ	*angle between the cables and the vertical (oriented as on the Figure 3.2 and counted positively in the trigonometric direction)*

We will take as our point of view that:

- *the **input** of the system is the horizontal velocity of the trolley v_c (at any time t, we can fix the value of $v_c(t)$);*

- *the **output** of the system is the horizontal speed v_m of the mass (container).*

*We neglect the **perturbations (noises)** which can intervene on the system : we could in particular consider the wind (external quantity undergone which has an influence on the sway of the container). As already mentioned above, the objective will be to control the speed of the mass while avoiding swaying.*
*This system is **deterministic**: an input v_c leads to a single possible output v_m.*
*This system is **causal**: the value of the output at a time t_0, $v_m(t_0)$, does not depend on the future of the input $v_c(t)$ for $t > t_0$.*

*We can use the laws of Physics to carry out a **modelling** of this system. In particular, we will establish the differential equations governing the evolution of the associated variables (to have a parametric model):*

a) The definition of the sine in a right-angled triangle gives us that $p_c = p_m + L\sin\theta$. We can deduce a first relation linking the speed of the trolley to that of the mass :

$$v_c = v_m + L\frac{d\sin\theta}{dt}. \tag{3.1}$$

b) The mass behaves like a pendulum. The fundamental principle of dynamics applied to the mass and projected along the $\overrightarrow{x}$ axis gives:

$$ma_m = T\sin\theta,$$

where $\overrightarrow{T}$ is the pull of the mass on the cables. Since the cables are considered to be non-deformable, the tensile force $\overrightarrow{T}$ and the component of the weight along the cable axis cancel out, i.e.:

$$T = mg\cos\theta.$$

This allows us to deduce

$$ma_m = mg\sin\theta\cos\theta \tag{3.2}$$

*The system modelled by the equations (3.1)-(3.2) is **stationary** since the values of the parameters involved in these differential equations (namely m, g and L) are assumed to be constant over time. On the other hand, it is not linear. In order to have a linear model, we will approximate the behaviour of the system (the model will be less precise but easier to manipulate because it is linear). In particular, we will consider that the variations of θ are weak and thus that $\cos\theta \approx 1$, $\sin\theta \approx \theta$. This leads to approach the equations (3.1)-(3.2) in the form of the following linear model:*

$$\frac{d\theta}{dt} = -\frac{1}{L}v_m + \frac{1}{L}v_c \tag{3.3}$$

$$\frac{dv_m}{dt}(= a_m) = g\theta \tag{3.4}$$

In the following two paragraphs, we recall the notions of stability and precision which will be useful later.

3.1.1 Stability

A linear system is *stable* if for any bounded input, the corresponding output is bounded.
Equivalently, a system is *stable* if when the input cancels, the corresponding output tends to zero when t tends to infinity.

Theorem 1 (Stability condition for a system described by a TF) *Let a dynamic system be described by its transfer function $H(s)$. It is stable if, and only if, all the poles of its transfer function $H(s)$ (i.e., the roots of the denominator of $H(s)$) have strictly negative real parts.*

Example 2 *We can guess that the system corresponding to the trolley-cable-container assembly is not stable. Indeed, if a step is applied to the input, then the input is cancelled, then the oscillations at the output will continue indefinitely (friction is neglected and the amplitude of the oscillations will not decrease) and the output will not be cancelled asymptotically. We check that the roots of the polynomial $1 + \frac{L}{g}s^2$ (denominator of the transfer function) are $\pm i\sqrt{\frac{g}{L}}$. These have not strictly negative real parts, which confirms that the system is unstable.*

3.1.2 Precision

A controlled linear system with output $y(t)$ is more accurate the smaller the difference between the desired output $y_d(t)$ and the actual output $y(t)$. The *accuracy* can be measured by :

$$\varepsilon(t) \triangleq y_d(t) - y(t)$$

The *static accuracy* ε_0 qualifies the value of this deviation $\varepsilon(t)$ when t tends towards infinity (steady state), that is :

$$\varepsilon_0 = \lim_{t\to\infty} \varepsilon(t)\ .$$

A distinction is sometimes made between the order of the stationary error. More precisely, the stationary error of order n, noted ε_n, is relative to an input of the form $U(s) = \frac{1}{s^n}$.

3.2 Introduction to state formalism

3.2.1 State representation

In this course, systems (physical, biological, economic processes,...) are described by equations of type:

$$\left\{\begin{array}{rcl} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) + Du(t) \end{array}\right. \tag{3.5}$$

under the assumption that time t is continuous (i.e. it takes its values in $\mathbb{R}$).

- The vector $u(t)$ is the input (or *command*) of the system. Its value can be chosen arbitrarily for any t.
- The vector $y(t)$ is the *output* of the system that can be measured.
- The vector $x(t)$ is called *state* of the system. As we will illustrate later, it represents the memory of the system, i.e. the set of information needed to predict the evolution of the system given the input $u(t)$.
- The matrices A, B, C and D[1] are called respectively *evolution, control, observation and direct matrices.*

- The first of the two equations of (3.5) is called *evolution equation.* It is a differential equation which allows us to know where the state $x(t)$ is going to go, knowing its value at the present time t and the command $u(t)$ which we are currently applying.
- The second equation is called *observation equation.* It allows the calculation of the output vector $y(t)$ knowing the state and the command at time t. Unlike the evolution equation, it is not a differential equation.

- The equations (3.5) constitute the *state representation* of the system[2].

- Conventionally, we note n, m and p the respective dimensions of the vectors x, u and y. The dimensions of the matrices A, B, C and D are therefore $n \times n$, $n \times m$, $p \times n$ and $p \times m$ respectively.

When the system is studied *via* a computer, we are not interested in its evolution continuously, but only in *discrete* instants of time (synchronised to the processor clock). It is then useful to consider that time takes its k values in $\mathbb{Z}$. We then use the *discrete time state representation* in the form of recurrence equations :

$$\left\{\begin{array}{rcl} x(k+1) & = & Ax(k) + Bu(k) \\ y(k) & = & Cx(k) + Du(k) \end{array}\right. \tag{3.6}$$

To be noted 1
Several general remarks on the state representation approach :

- *systems are studied in the time domain,*
- *the treatments use matrix algebra,*
- *the formalism naturally applies to multi-variable systems (several inputs and/or several outputs).*

[1] In the following, it will often be considered that the matrix D is zero.

[2] We restrict ourselves to linear systems, i.e. those for which the response to the weighted sum of several excitations is equal to the weighted sum of the responses to each of the excitations taken separately.

3.2.2 State variables

Recall that linear systems in continuous time can be represented by their impulse response[3] h and then we have the input-output relationship

$$\begin{aligned} y(t) &= (h * u)(t) \\ &= \int\limits_0^t h(\tau)u(t-\tau)d\tau\ . \end{aligned} \tag{3.7}$$

Using this representation, it appears that in order to calculate the output at a time t, one needs to know the entire past (before t) of the system input.

If we consider this time a first order linear system represented by its input-output differential equation:

$$\left\{ \begin{aligned} \dot{y}(t) - ay(t) &= bu(t) \\ y(0^+) &= y_0 \end{aligned} \right.$$

The solution to this equation is

$$y(t) = y(0^+) \cdot e^{at} + \int\limits_0^t b \cdot e^{a(t-x)} \cdot u(x)dx$$

It also appears that in order to calculate the output at a time t, one needs to know the entire past (before t) of the system input.

On the contrary, from its state representation, the output at time t of the same linear system can be directly deduced from the knowledge of the state at t. Indeed, the output equation in (3.5) gives $y(t)$ as a linear function of the components of the state vector $x(t)$. In a way, the value of the state vector summarises all the past behaviour of the system.

Physically, the notion of state can be intuited through the following examples.

Example 3 *Let us consider the example of a solid pushed on a horizontal surface. If we consider as input to the system the external force that pushes the solid. We can be convinced that in order to guess the future position of the solid, it is not necessary to know the value of the applied force from the beginning of time. The knowledge of the force at the present moment is sufficient if we also have the current values of the position and the velocity (which translates the kinetic energy of the system).*

Example 4 *Let us consider the example of an RLC circuit. If we consider the voltage $u(t)$ delivered by the power supply as the input of the system, and the current $i(t)$ in the circuit as the output. We can convince ourselves that to determine the future evolution of the current $i(t)$, it is not necessary to know the value of the applied voltage $u(t)$ from the origin of time. This information can be replaced by the value of the charge in the capacitor and the value of the flux in the inductor at the present time. These values (which determine the energy stored in the circuit) constitute the state of the system : they extract from the past history of the circuit the information necessary to determine the future.*

Example 5 *Consider the example of a mass attached to a spring and pulled by an external force. If we consider as input to the system the force pulling the mass. We can convince ourselves that in order to determine the evolution of the position of the mass, it is not necessary to know the value of the applied force from the origin of time. This information can be replaced by the value of the position of the mass and its speed at the present moment.*

Example 6 *Consider again the example of the harbour crane described in the example 1. For this example, $u = v_c$ (speed of the trolley) and $y = v_m$ (speed of the mass). One can guess that to determine the evolution of the mass velocity, one can be satisfied with knowing this velocity (which translates the kinetic energy), the angle θ (which translates the potential energy) and the future evolution of the trolley velocity (and it is not necessary to know its value since the origin of time).*
We therefore choose $x = \begin{pmatrix} v_m \\ \theta \end{pmatrix}$. From the equations (3.3) and (3.4), we deduce a state representation of the system:

$$\left\{ \begin{aligned} \dot{x}(t) &= \begin{pmatrix} 0 & g \\ -\frac{1}{L} & 0 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ \frac{1}{L} \end{pmatrix} u(t) \\ y(t) &= \begin{pmatrix} 1 & 0 \end{pmatrix} x(t) \end{aligned} \right. .$$

[3] The transfer function is the Laplace transform of the impulse response, and we then have the input-output relationship

$$Y(s) = H(s) \cdot U(s)$$

To be noted 2

- *The parameters describing the "energy reservoirs" of the system are generally good state variables of the system.*
- *We will further explain in the following that the number of state variables (the dimension of the state vector) corresponds to the order of the system.*

3.2.3 Similar state representations

Consider a system represented by its equations of state (3.5) in continuous time. Let P be an invertible square matrix, and let $\chi(t) = P^{-1}x(t)$. If we replace $x(t)$ by $P\chi(t)$ in (3.5), we obtain

$$\left\{\begin{array}{rcl} P\dot{\chi}(t) & = & AP\chi(t) + Bu(t) \\ y(t) & = & CP\chi(t) + Du(t) \end{array}\right. ,$$

that is,

$$\left\{\begin{array}{rcl} \dot{\chi}(t) & = & P^{-1}AP\chi(t) + P^{-1}Bu(t) \\ y(t) & = & CP\chi(t) + Du(t) \end{array}\right. .$$

If we now set $A' = P^{-1}AP$, $B' = P^{-1}B$, $C' = CP$ and $D' = D$, the previous system of equations is written

$$\left\{\begin{array}{rcl} \dot{\chi}(t) & = & A'\chi(t) + B'u(t) \\ y(t) & = & C'\chi(t) + D'u(t) \end{array}\right. \tag{3.8}$$

The form of the latter equations is identical to that adopted for a state representation (3.5). In other words, a second state representation for the same system was obtained.

To be noted 3

- *In a completely similar way, one can obtain an equivalent state representation for a discrete time system described by (3.6) (the manipulations used -multiplication by an invertible matrix and its inverse, change of variables- are independent of the definition set of the variables).*
- *A system has as many similar state representations as there are invertible square matrices P.*
- *The vector $\chi(t)$ is also a state vector of the system. The representations are equivalent, but the physical quantities represented by the state variables are different (except if P is equal to the identity matrix). This observation can be understood from the example 4: an equivalent state representation is obtained if the state variables of the RLC circuit are the charge and the current rather than the voltage across the capacitor (proportional to the charge) and the flux in the inductor (proportional to the current) respectively.*
- *State trasnsformations (via an invertible matrix) are often used to : facilitate the solution of certain problems or computations (e.g., evaluating A^n is easier if A is in a block diagonal form) or favour the study of properties such as stability, controllability or observability (which we will study later on)*

Remark 1 (How do we know if two state representations are equivalent?)
Suppose we have two state representations given by (3.5) and (3.8) that are supposed to represent the same system. The basis change matrix P which connects these two representations can be obtained by solving the linear system :

$$\left\{\begin{array}{rcl} PA' & = & AP \\ PB' & = & B \\ C' = CP & & \end{array}\right.$$

where the n^2 unknowns are the coefficients P_{ij} of the matrix P. Solving such a linear system is quite easy, and if no solution exists, it is clear that the two representations are not equivalent.

3.3 State representation of linear systems in continuous time

3.3.1 From a state representation to an input-output representation

The purpose of this section is to show how to obtain the transfer function of a linear system described by its state representation. For this purpose, let us apply the Laplace transform to the evolution and output matrix equations (3.5), which amounts to applying it to each of the time equations, and then writing the matrix form again. Considering that $x(0) = 0$

$$\left\{ \begin{array}{rcl} sX(s) & = & AX(s) + BU(s) \\ Y(s) & = & CX(s) + DU(s) \end{array} \right.$$

The first equation can be written as $sX(s) - AX(s) = BU(s)$, or even[4] $(s\mathbf{I} - A)X(s) = BU(s)$ with $\mathbf{I}$ the identity matrix (matrix of the dimension of A with 1 on the diagonal and 0 everywhere else). Thus, we end up with the equivalent system :

$$\left\{ \begin{array}{rcl} X(s) & = & (s\mathbf{I} - A)^{-1}BU(s) \\ Y(s) & = & CX(s) + DU(s) \end{array} \right.$$

The notation $(s\mathbf{I} - A)^{-1}$ is the inverse matrix of $s\mathbf{I} - A$. For small dimensions, it can be calculated "by hand" (see for example the reminders of linear algebra in the appendix). One can also use the Scilab software[5]. The result is

$$Y(s) = \left[C(s\mathbf{I} - A)^{-1}B + D\right] U(s) \; .$$

The matrix $C(s\mathbf{I} - A)^{-1}B + D$ is the transfer matrix of the system. In the mono-variable case (only one input and one output), $C(s\mathbf{I} - A)^{-1}B + D$ is the *transfer function*, i.e. a fraction of polynomials in s. Multiplying each member of the previous equation by the denominator of $C(s\mathbf{I} - A)^{-1}B + D$, and then performing the inverse Laplace transform gives an input-output differential equation for our system.

Example 7 *Consider again the example of the harbour crane described in the example 1 and for which a state representation was given in the example 6. The transfer function can be obtained from the state representation by applying the formula $H(s) = C(s\mathbf{I} - A)^{-1}B$. Several intermediate calculation results are given for the application of this formula :*

$$s\mathbf{I} - A = \begin{pmatrix} s & -g \\ \frac{1}{L} & s \end{pmatrix} \; , \; |s\mathbf{I} - A| = s^2 + \frac{g}{L} \; , \; com(s\mathbf{I} - A) = \begin{pmatrix} s & -\frac{1}{L} \\ g & s \end{pmatrix}$$

and

$$(s\mathbf{I} - A)^{-1} = \frac{1}{|s\mathbf{I} - A|} com(s\mathbf{I} - A)^{\top} = \frac{1}{s^2 + \frac{g}{L}} \begin{pmatrix} s & g \\ -\frac{1}{L} & s \end{pmatrix}.$$

The result is

$$H(s) = \frac{\frac{g}{L}}{s^2 + \frac{g}{L}} = \frac{1}{1 + \frac{L}{g}s^2}.$$

Example 8 *Consider a linear system described by the state representation*

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & \begin{pmatrix} 1 & 3 \\ 2 & 0 \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ 1 \end{pmatrix} u(t) \\ \\ y(t) & = & \begin{pmatrix} 0 & 1 \end{pmatrix} x(t) \end{array} \right. .$$

The matrix $(s\mathbf{I} - A)^{-1}$ can be calculated from Scilab by typing the following lines :

```
A=[1 3;2 0];
I=eye(2,2);
s=poly(0,'s');
(s*I-A)^(-1)
```

Finally, we obtain the input-output differential equation:

$$\ddot{y}(t) - \dot{y}(t) - 6y(t) = \dot{u}(t) + u(t)$$

[4] Writing $sX(s) - AX(s) = (s - A)X(s)$ is not correct because s is a scalar while A is a matrix.

[5] Free numerical computation software that incorporates the common operations of matrix computation. It also offers limited symbolic calculation functionality, which can be used to calculate $(s\mathbf{I} - A)^{-1}$, see example 8.

3.3.2 From an input-output representation to a state representation

In this section, we examine how to obtain the state representation from an input-output representation of a single-variable system (a single input and a single output). As we noticed in the section 3.2.3, a linear system admits a set of equivalent state representations. Starting from an input-output representation, there are therefore several methods for obtaining equivalent state representations, each with a particular form. In this paragraph we concentrate on obtaining two forms which will be useful later.

To explain each of these forms, consider a linear system of order 3 described equivalently by the following differential equation

$$\frac{d^3y(t)}{dt} + a_2\frac{d^2y(t)}{dt} + a_1\frac{dy(t)}{dt} + a_0y(t) \;=\; b_2\frac{d^2u(t)}{dt} + b_1\frac{du(t)}{dt} + b_0u(t) \;, \tag{3.9}$$

or the following transfer function

$$H(s) \;=\; \frac{b_2s^2+b_1s+b_0}{s^3+a_2s^2+a_1s+a_0} \;, \tag{3.10}$$

with

$$Y(s) = H(s)\cdot U(s) \;. \tag{3.11}$$

Controllable canonical form

From the transfer function given by (3.10), we can write the equation (3.11) in the form of the system of equations

$$\left\{\begin{array}{rcl} X(s) &=& \dfrac{1}{s^3+a_2s^2+a_1s+a_0}U(s) \\[2ex] Y(s) &=& \left(b_2s^2+b_1s+b_0\right)X(s) \end{array}\right.$$

or even

$$\left\{\begin{array}{l} s^3X(s)+a_2s^2X(s)+a_1sX(s)+a_0X(s) \;=\; U(s) \\ Y(s) \;=\; b_2s^2X(s)+b_1sX(s)+b_0X(s) \end{array}\right.$$

The objective is then to introduce additional variables so as to obtain a system of equations involving polynomials in s of degrees at most equal to 1. To do this, let $X_1(s) = X(s)$, $X_2(s) = sX_1(s)$, $X_3(s) = sX_2(s)$, and we obtain

$$\left\{\begin{array}{l} X_2(s) = sX_1(s) \\ X_3(s) = sX_2(s) \\ sX_3(s)+a_2sX_2(s)+a_1sX_1(s)+a_0X_1(s) \;=\; U(s) \\ Y(s) \;=\; b_2sX_2(s)+b_1sX_1(s)+b_0X_1(s) \end{array}\right.$$

or even

$$\left\{\begin{array}{l} X_2(s) = sX_1(s) \\ X_3(s) = sX_2(s) \\ sX_3(s) \;=\; -a_2X_3(s)-a_1X_2(s)-a_0X_1(s)+U(s) \\ Y(s) \;=\; b_2X_3(s)+b_1X_2(s)+b_0X_1(s) \end{array}\right. .$$

This system of equations can be written in the following matrix form

$$\left\{\begin{array}{rcl} \begin{pmatrix} sX_1(s) \\ sX_2(s) \\ sX_3(s) \end{pmatrix} &=& \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{pmatrix}\cdot\begin{pmatrix} X_1(s) \\ X_2(s) \\ X_3(s) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}\cdot U(s) \\[4ex] Y(s) &=& \begin{pmatrix} b_0 & b_1 & b_2 \end{pmatrix}\cdot\begin{pmatrix} X_1(s) \\ X_2(s) \\ X_3(s) \end{pmatrix} \end{array}\right. .$$

The inverse Laplace transform of the equations leads us to the so-called *controllable canonical form* of the state representation. Namely, by posing $x(t) = \begin{pmatrix} x_1(t) & x_2(t) & x_3(t) \end{pmatrix}^\top$,

$$\left\{\begin{array}{rcl} \dot{x}(t) &=& \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{pmatrix}x(t) + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}u(t) \\[4ex] y(t) &=& \begin{pmatrix} b_0 & b_1 & b_2 \end{pmatrix}x(t) \end{array}\right. . \tag{3.12}$$

Observable canonical form

From the transfer function given by (3.10), we can write the equation (3.11) in the form

$$s^3Y(s)+a_2s^2Y(s)+a_1sY(s)+a_0Y(s) \;=\; b_2s^2U(s)+b_1sU(s)+b_0U(s)$$

Dividing each member by s^3 and isolating $Y(s)$, we obtain

$$Y(s)=-\frac{a_2}{s}Y(s)-\frac{a_1}{s^2}Y(s)-\frac{a_0}{s^3}Y(s)+\frac{b_2}{s}U(s)+\frac{b_1}{s^2}U(s)+\frac{b_0}{s^3}U(s)\ ,$$

i.e.

$$Y(s)=\frac{1}{s}\left[(b_2U(s)-a_2Y(s))+\frac{1}{s}\left[(b_1U(s)-a_1Y(s))+\frac{1}{s}(b_0U(s)-a_0Y(s))\right]\right]\ .$$

Let's put

$$\begin{array}{rcl} X_3(s) & = & Y(s) \\ X_1(s) & = & \frac{1}{s}(b_0U(s)-a_0Y(s)) \\ & = & \frac{1}{s}(b_0U(s)-a_0X_3(s)) \\ X_2(s) & = & \frac{1}{s}\left(b_1U(s)-a_1Y(s)+X_1(s)\right) \\ & = & \frac{1}{s}\left((b_1U(s)-a_1X_3(s)+X_1(s)\right) \end{array}$$

The above equation can then be written as the following system of equations

$$\left\{\begin{array}{rcl} X_1(s) & = & \frac{1}{s}(b_0U(s)-a_0X_3(s)) \\ X_2(s) & = & \frac{1}{s}\left((b_1U(s)-a_1X_3(s))+X_1(s)\right) \\ X_3(s) & = & \frac{1}{s}\left((b_2U(s)-a_2X_3(s))+X_2(s)\right) \\ Y(s) & = & X_3(s) \end{array}\right. ,$$

i.e.

$$\left\{\begin{array}{rcl} sX_1(s) & = & b_0U(s)-a_0X_3(s) \\ sX_2(s) & = & b_1U(s)-a_1X_3(s)+X_1(s) \\ sX_3(s) & = & b_2U(s)-a_2X_3(s)+X_2(s) \\ Y(s) & = & X_3(p) \end{array}\right. .$$

In matrix form, we obtain

$$\left\{\begin{array}{rcl} \begin{pmatrix} sX_1(s) \\ sX_2(s) \\ sX_3(s) \end{pmatrix} & = & \begin{pmatrix} 0 & 0 & -a_0 \\ 1 & 0 & -a_1 \\ 0 & 1 & -a_2 \end{pmatrix}\cdot\begin{pmatrix} X_1(s) \\ X_2(s) \\ X_3(s) \end{pmatrix}+\begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}\cdot U(s) \\ \\ Y(s) & = & \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}\cdot\begin{pmatrix} X_1(s) \\ X_2(s) \\ X_3(s) \end{pmatrix} \end{array}\right. .$$

The inverse Laplace transform of the equations leads us to the so-called *observable canonical form* of the state representation. Namely, by posing $x(t)=\begin{pmatrix} x_1(t) & x_2(t) & x_3(t)\end{pmatrix}^\top$,

$$\left\{\begin{array}{rcl} \dot{x}(t) & = & \begin{pmatrix} 0 & 0 & -a_0 \\ 1 & 0 & -a_1 \\ 0 & 1 & -a_2 \end{pmatrix}x(t)+\begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}u(t) \\ \\ y(t) & = & \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}x(t) \end{array}\right. . \tag{3.13}$$

3.3.3 Simulation from the state representation

In this paragraph, we will present Euler's method for performing a computer simulation of a system described by its equations of state (3.5). This method is rather rough but simple to understand and satisfactory for describing the behaviour of most systems.

Let dt be a number very small compared to the time constants of the system and which corresponds to the sampling period of the method. The evolution equation of (3.5) can be approximated [6] by

$$\tfrac{x(t+dt)-x(t)}{dt} \quad \simeq \quad Ax(t)+Bu(t) \tag{3.14}$$

[6] This approximation can be interpreted as a Taylor expansion to order 1.

or even

$$x(t+dt) \quad \simeq \quad x(t)+Ax(t)\cdot dt+Bu(t)\cdot dt \tag{3.15}$$

The following simulation algorithm is derived:

```
x:=x0; t:=0; dt:=0.01;
repeat
    transmit (enter or assign) the value of u as input to the computer;
    y:=Cx+Du;
    exit (display or store) y;
    x:=x+A.x.dt+B.u.dt;
    wait for an interruption of the sampler;
    t=t+dt;
indefinitely
```

The sampler produces a periodic interrupt every dt seconds. Thus, if the computer is fast enough, the simulation runs at the same speed as the evolution of our physical system. We will then speak about *real time* simulation .
In some circumstances, we are interested in obtaining the result of the simulation as quickly as possible (for example, to predict how a system will behave in the future). This is known as 'time-delayed' simulation. In this case, it is not necessary to slow down the computer to synchronise it with our physical time, i.e. we omit the line waiting for an interrupt in the algorithm presented above.

3.3.4 Solution of the state equations

In this section, we focus on calculating the solution to the equations of state in analytical form (3.5).

Theorem 2 (Solution of the state equations) *Let $x(t_0)$ be the state at the initial time t_0. For a continuous-time linear system represented by (3.5), the state at time t is given by*

$$x(t)=e^{A(t-t_0)}x(t_0)+\int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau \ , \tag{3.16}$$

and the output is expressed as

$$y(t)=Ce^{A(t-t_0)}x(t_0)+\int_{t_0}^{t} Ce^{A(t-\tau)}Bu(\tau)d\tau+Du(t) \ . \tag{3.17}$$

Equation (3.16) provides the state of the system for any $t \geq t_0$ from the initial state $x(t_0)$, and the input $u(t)$ applied on the interval $[t_0, t]$.

- Function $Ce^{A(t-t_0)}x(t_0)$ is called *free homogeneous solution* (or *transitory*).
- Function $\int_{t_0}^{t} Ce^{A(t-\tau)}Bu(\tau)d\tau+Du(t)$ is called *forced solution.*

The reader is referred to the appendices of this document for a brief review of the properties of matrix exponentials, as well as for methods of calculation.

3.3.5 Stability

Definition 1 (Stability) *A linear system is (asymptotically) stable if when the input cancels, its state tends to 0 when t tends to infinity.*

Suppose a non-zero input is applied to a system represented by (3.5) from the initial time t_0 to a time t_1. The state at time t_1 can be evaluated from (3.16) :

$$x(t_1)=e^{A(t_1-t_0)}x(t_0)+\int_{t_0}^{t_1} e^{A(t_1-\tau)}Bu(\tau)d\tau \ .$$

If the input is zero after t_1, the state for $t \geq t_1$ is always written from (3.16) :

$$\begin{array}{rcl} x(t) & = & e^{A(t-t_1)}x(t_1) + \int\limits_{t_1}^{t} e^{A(t-\tau)}Bu(\tau)d\tau \\ & = & e^{A(t-t_1)}x(t_1) \ , \end{array}$$

which cancels when t tends to infinity if

$$\lim_{t\to\infty} e^{A(t-t_1)} = \mathbf{0} \ .$$

A definition equivalent to the definition 1 is therefore stated as follows.

Definition 2 *A linear system is (asymptotically) stable if after a sufficiently long time the state no longer depends on the initial conditions (whatever they may be).*

By choosing arbitrarily $t_1 = 0$, to characterize the stability we will look for the conditions so that

$$\lim_{t\to\infty} e^{At} = \mathbf{0} \ .$$

Theorem 3 (Stability criterion) *A linear system is stable if, and only if, all the eigenvalues of its evolution matrix have strictly negative real parts.*

The reader is referred to the appendices of this document for a brief review of the eigenvalues of a matrix.

Definition 3 (Characteristic polynomial) *The characteristic polynomial $P(s)$ of a linear system represented by (3.5) is defined as the characteristic polynomial of the evolution matrix A, i.e.*

$$P(s) = |s\mathbf{I} - A| \tag{3.18}$$

The roots of $P(s)$ are the eigenvalues of A.

In the mono-variable case, one can notice, referring to the paragraph 3.3.1, that $P(s)$ constitutes the denominator of the transfer function of the system. The roots of $P(s)$ then correspond to the *poles* of the transfer function.

Corollary 1 (Stability criterion) *A linear system is stable if, and only if, all roots of its characteristic polynomial have strictly negative real parts.*

Example 9 *Consider again the example of the harbour crane described in the example 1 and for which a state representation was given in the example 6, i.e.*

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & \begin{pmatrix} 0 & g \\ -\frac{1}{L} & 0 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ \frac{1}{L} \end{pmatrix} u(t) \\ y(t) & = & \begin{pmatrix} 1 & 0 \end{pmatrix} x(t) \end{array} \right. .$$

From this state representation, we can study the stability of the system by calculating the eigenvalues of A which are also the roots of its characteristic polynomial:

$$P(s) = |s\mathbf{I} - A| = \begin{vmatrix} s & -g \\ \frac{1}{L} & s \end{vmatrix} = s^2 + \frac{g}{L}.$$

Its roots are the pure imaginary $\pm i\sqrt{\frac{g}{L}}$, and the system is therefore unstable. This conclusion corroborates that of the example 2.

3.3.6 Controllability

There are several equivalent definitions for the notion of the *controllability* of a linear system. In this course, we retain the following one.

Definition 4 (Controllability) *A linear system represented by the equations (3.5) is said to be controllable if for any pair of state vectors (x_0, x_1), we can find a time t_1 and a command $u(t)$, $t \in [t_0, t_1]$, such that the system, initialised in x_0 at time t_0, reaches the state x_1 at time t_1.*

Note that in this definition, no assumption is made about the amplitude of the signals, and in particular, the u control can be as energetic as necessary. In practice, it is necessary to take into account the saturations of the actuators (which deliver the command).

Theorem 4 (Controllability criterion) *A linear system represented by the equations (3.5) is controllable if and only if,*

$$\text{rank}\,\overbrace{\left(B|AB|A^2B|\ldots|A^{n-1}B\right)}^{\Gamma_{com}} = n\ , \tag{3.19}$$

where n is the dimension of the evolution matrix A (or, equivalently, the number of state variables). In other words, it is necessary that the Γ_{com} matrix, known as the controllability matrix, obtained by juxtaposing the n matrices $B,\ AB,\ \ldots,\ A^{n-1}B$ next to each other, is of rank equal to the dimension of the A matrix.

Example 10 *Consider again the example of the harbour crane described in the example 1 with the state representation*

$$\left\{\begin{array}{rcl} \dot{x}(t) & = & \begin{pmatrix} 0 & g \\ -\frac{1}{L} & 0 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ \frac{1}{L} \end{pmatrix} u(t) \\ y(t) & = & \begin{pmatrix} 1 & 0 \end{pmatrix} x(t) \end{array}\right. .$$

We have

$$\Gamma_{com} = \begin{pmatrix} B & AB \end{pmatrix} = \begin{pmatrix} 0 & \frac{g}{L} \\ \frac{1}{L} & 0 \end{pmatrix}$$

The determinant of Γ_{com} is non-zero, the rank of Γ_{com} is thus equal to 2 (the dimension of A), and one can conclude that the system is controllable. This means that starting from any value of the state (the state variables are the velocity of the mass and the angle θ), we can "bring" the state to any desired value.

3.3.7 Observability

Definition 5 (Observability) *A linear system represented by the equations (3.5) is said to be observable if the knowledge of $y(t)$ and $u(t)$ for $t \in \mathbb{R}$ allows us to uniquely determine the state $x(t)$, for all t.*

This property is of practical importance when some of the state variables of a system are inaccessible to measurement. If the system is observable, then this guarantees that there is a way (presented later in this document) to calculate the values of these state variables from measurements of the output.

Theorem 5 (Observability criterion) *A linear system represented by the equations (3.5) is observable if and only if,*

$$\text{rank}\,\overbrace{\begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix}}^{\Gamma_{obs}} = n\ , \tag{3.20}$$

where n is the dimension of the evolution matrix A (or, equivalently, the number of state variables). In other words, it is necessary that the Γ_{obs} matrix, known as the observability matrix, obtained by stacking the n matrices $C,\ CA,\ \ldots,\ CA^{n-1}$ one below the other, is of rank equal to the dimension of the A matrix.

Example 11 *Let us consider again the example of the harbour crane. We have*

$$\Gamma_{obs} = \begin{pmatrix} C \\ CA \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & g \end{pmatrix},$$

$|\Gamma_{obs}| = g \neq 0$ and thus $\text{rang}(\Gamma_{obs}) = 2 = \dim(A)$. The system is therefore observable. This means that from the knowledge of the input and output alone, there is a way to calculate the values of the state variables.

3.4 State representation of linear systems in discrete time

In this chapter, we focus on linear systems that can be represented by the equations of state

$$\left\{\begin{array}{rcl} x(k+1) & = & Ax(k) + Bu(k) \\ y(k) & = & Cx(k) + Du(k) \end{array}\right. \tag{3.21}$$

where time, denoted k, takes discrete values (typically in $\mathbb{Z}$).

3.4.1 Solution of the state equations

The following theorem provides the analytical solution to the state equations (3.21).

Theorem 6 (Solution of the state equations) *Let $x(k_0)$ be the state at the initial time k_0. For a discrete-time linear system represented by (3.21), the state at time k is given by*

$$x(k) = A^{k-k_0}x(k_0) + \sum_{l=k_0}^{k-1} A^{k-1-l}Bu(l) \ , \tag{3.22}$$

and the output is expressed as

$$y(k) = CA^{k-k_0}x(k_0) + \sum_{l=k_0}^{k-1} CA^{k-1-l}Bu(l) + Du(k) \ . \tag{3.23}$$

Equation (3.22)provides the state of the system for any $k \geq k_0$ from the initial state $x(k_0)$, and the input $u(k)$ applied on the interval $[k_0, k]$.

- Function $CA^{k-k_0}x(k_0)$ is called *free homogeneous solution* (or *transitory*).
- Function $\sum\limits_{l=k_0}^{k-1} CA^{k-1-l}Bu(l) + Du(k)$ is called *forced solution.*

Theorem 6 is easily proven by recurrence.

3.4.2 Stability

Definition 1 applies to discrete time systems (denoting k as time instead of t). Suppose a non-zero input is applied to a system represented by (3.21) from the initial time k_0 to a time k_1. The state at time k_1 can be evaluated from (3.22) :

$$x(k_1) = A^{k_1-k_0}x(k_0) + \sum_{l=k_0}^{k_1-1} A^{k_1-1-l}Bu(l) \ .$$

If the input is zero after k_1, the state for $k \geq k_1$ is always written from (3.22) :

$$\begin{array}{rcll} x(k) & = & A^{k-k_1}x(k_1) + \sum\limits_{l=k_1}^{k-1} A^{k-1-l}Bu(l) & \\ & = & A^{k-k_1}x(k_1) & (\text{car } u(l) = 0 \text{ pour } l \geq k_1) \ , \end{array}$$

which cancels when k tends to infinity if

$$\lim_{k\to\infty} A^{k-k_1} = \mathbf{0} \ .$$

Definition 2 is therefore also equivalent for stability.
By choosing arbitrarily $k_1 = 0$, to characterize the stability we will look for conditions so that

$$\lim_{k\to\infty} A^{k} = \mathbf{0} \ .$$

Theorem 7 (Stability criterion) *A linear system is stable if, and only if, all the eigenvalues of its evolution matrix are strictly in the unit disc.*

The reader is referred to the appendices of this document for a brief review of the eigenvalues of a matrix.

Definition 6 (Characteristic polynomial) *The characteristic polynomial $P(z)$ of a linear system represented by (3.21) is defined as the characteristic polynomial of the evolution matrix A, i.e.*

$$P(z) = |z\mathbf{I} - A| \tag{3.24}$$

The roots of $P(z)$ are the eigenvalues of A.

Corollary 2 (Stability criterion) *A linear system is stable if, and only if, all roots of its characteristic polynomial are strictly in the unit disc.*

3.4.3 Controllability

The *controllability* of a linear system in discrete time is defined in the same way as for a system in continuous time (*cf.* definition 4 by denoting k the time instead of t.).
The controllability criterion is also the same: we will therefore also apply theorem 4 to characterise the controllability of a discrete-time system.

3.4.4 Observability

The *observability* of a linear system in discrete time is defined in the same way as for a system in continuous time (*cf.* definition 5 by denoting k the time instead of t.).
The observability criterion is also the same: we will therefore also apply theorem 5 to characterise the observability of a system in discrete time.

3.4.5 From an input-output representation to a state representation

While continuous-time linear systems can be described by an input-output differential equation, discrete-time systems admit an input-output relationship in the form of a difference equation. In the single-variable case, we have an equation of the type:

$$y(k+n)+a_{n-1}y(k+n-1)+\ldots+a_0y(k)=b_mu(k+m)+b_{m-1}u(k+m-1)+\ldots+b_0u(k) \tag{3.25}$$

In this section, we examine how to obtain the state representation from such an input-output representation of a single-variable system. As we noted in section 3.2.3, a linear system admits a set of equivalent state representations. Starting from an input-output representation, there are therefore several methods for obtaining equivalent state representations, each with a particular form. In this paragraph we concentrate on obtaining two forms which will be useful later.

Controllable canonical form

Let us choose a state variable $x_1(k)$ such that it checks

$$x_1(k+n)+a_{n-1}x_1(k+n-1)+\ldots+a_0x_1(k)=u(k) \tag{3.26}$$

Then we pose,

$$\left\{\begin{array}{rcl} x_2(k) & = & x_1(k+1) \\ x_3(k) & = & x_2(k+1) \\ & \vdots & \\ x_n(k) & = & x_{n-1}(k+1) \end{array}\right.$$

We have $x_n(k+1)=x_1(k+n)$, and from (3.26), we will check that then

$$x_n(k+1)=-a_0x_1(k)-a_1x_2(k)-\ldots-a_{n-1}x_n(k)+u(k)$$

Let's put

$$x(k)=\begin{pmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{pmatrix}.$$

By writing these equations in matrix form, we get the state representation:

$$\left\{\begin{array}{rcl} x(k+1) & = & \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ -a_0 & -a_1 & -a_2 & \ldots & -a_{n-1} \end{pmatrix} x(k)+\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} u(k) \\ \\ y(k) & = & \begin{pmatrix} c_1 & c_2 & \ldots & c_n \end{pmatrix} x(k)+b_nu(k) \end{array}\right. . \tag{3.27}$$

with $c_i=b_{i-1}-b_na_{i-1}$.

Observable canonical form

Let us choose as state variables

$$\begin{cases} x_1(k) &= y(k) - b_n u(k) \\ x_2(k) &= x_1(k+1) + a_{n-1} y(k) - b_{n-1} u(k) \\ \vdots \\ x_n(k) &= x_{n-1}(k+1) + a_1 y(k) - b_1 u(k) \end{cases}$$

By posing

$$x(k) = \begin{pmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{pmatrix},$$

this leads us to the state representation:

$$\begin{cases} x(k+1) &= \begin{pmatrix} -a_{n-1} & 1 & 0 & \dots & 0 \\ -a_{n-2} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_1 & 0 & 0 & \dots & 1 \\ -a_0 & 0 & 0 & \dots & 0 \end{pmatrix} x(k) + \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_{n-1} \\ h_n \end{pmatrix} u(k) \\ \\ y(k) &= \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix} x(k) + b_n u(k) \end{cases} . \tag{3.28}$$

with $h_i = b_{n-i} - b_n a_{n-i}$.

3.4.6 Study of continuous systems by of a computer: sampled systems

In this paragraph, we deal with the study of continuous processes *via* a computer. More precisely, we place ourselves in the following context :

- We consider processes whose nature implies that their state variables depend continuously on time, and which are therefore represented by equations of state (3.5). The signals involved in this representation are functions of a continuous time (typically in $\mathbb{R}$).

- These systems are studied/manipulated (identified, analysed, controlled) by means of a computer: the input u is generated by the computer, and the measurement of the output y is read by the computer. The signals manipulated by the computer are in essence in discrete time (typically in $\mathbb{Z}$).

The signal processing chain is shown in Figure 3.3. Let's detail it from the left to the right :

1. The signal $u(k\Delta)$ is provided by the computer every Δ seconds, where Δ is called the *sampling period.* It is a binary number.

2. is translated into an analogue value by a *digital-to-analogue converter* (D/A).

3. The *zero-order hold* receives as input the analogue value delivered by the D/A. At its output, it maintains this value between two sampling times (the signal $u(t)$ has a constant value between two sampling times).

4. The *continuous process* is driven by this input signal $u(t)$ (in the form of a staircase). Its output response is the measured continuous signal $y(t)$.

5. This analogue signal is sampled every Δ seconds (role of the *sampler*).

6. The sampled signal is converted into a binary value by the *A/D converter*, which is read by the computer.

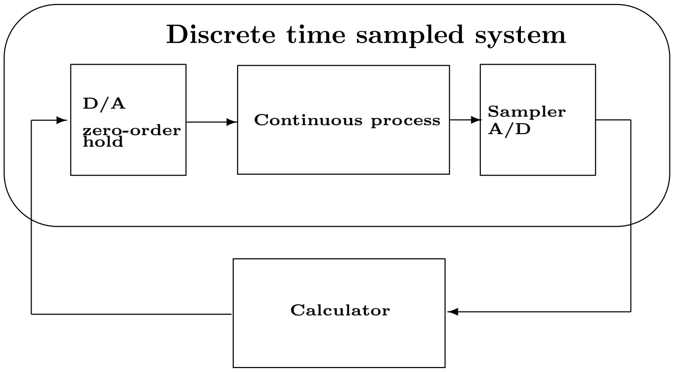


Figure 3.3: Continuous process perceived as a sampled system by the computer

Discretisation of a state equation in continuous time

We limit ourselves to describing the sampled systems at the sampling instants, they are therefore represented as discrete time systems. In this paragraph, we are therefore interested in the following problem:

- We have a continuous time state representation of the system :

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) + Du(t) \end{array} \right. \tag{3.29}$$

- The sampled system can be represented at sampling times $k\Delta$, which we note k for simplicity, by state equations:

$$\left\{ \begin{array}{rcl} x(k+1) & = & A_{ech}x(k) + B_{ech}u(k) \\ y(k) & = & C_{ech}x(k) + D_{ech}u(k) \end{array} \right. \tag{3.30}$$

- We wish to obtain the matrices A_{ech}, B_{ech}, C_{ech} and D_{ech} from the known matrices A, B, C and D.

Starting from (3.29) and knowing the state at the sampling time $t_k = k\Delta$, it is possible from (3.16) to establish the state at the next sampling time $t_{k+1} = (k+1)\Delta$:

$$x(t_{k+1}) = e^{A(t_{k+1}-t_k)}x(t_k) + \int\limits_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}Bu(\tau)d\tau \ . \tag{3.31}$$

The zero-order hold maintains $u(t)$ to the constant value $u(t_k) = u(k\Delta)$, noted $u(k)$, during interval $[t_k, t_{k+1}[$, we then have:

$$u(\tau) = u(k) \text{ pour } \tau \in [t_k, t_{k+1}[\ .$$

Let us carry out the change of variable $\nu = \tau - t_k$, and as $\Delta = t_{k+1} - t_k$, the equation (3.31) becomes:

$$x(k+1) = e^{A\Delta}x(k) + \int\limits_0^{\Delta} e^{A(\Delta-\nu)}d\nu Bu(k) \ . \tag{3.32}$$

By identification between (3.32) and (3.29), we deduce:

$$A_{ech} = e^{A\Delta} \tag{3.33}$$

$$B_{ech} = \int\limits_0^{\Delta} e^{A(\Delta-\nu)}d\nu B \tag{3.34}$$

$$C_{ech} = C \tag{3.35}$$

$$D_{ech} = D \tag{3.36}$$

To be noted 4

- *The matrices A_{ech} and B_{ech} depend on the sampling period Δ, and will therefore have to be re-evaluated each time the sampling period is modified.*
- *The computation of A_{ech} requires the evaluation of the exponential of a matrix. Elements for this computation are provided in the appendices. In practice, one can use the Mupad software for a formal or numerical calculation of A_{ech}. The software Scilab can also be used for the numerical calculation of A_{ech}.*
- *The computation of B_{ech} also requires the integration of a matrix expression. The integrand, which involves a matrix exponential, can first be calculated (notably using Mupad or Scilab). These elements (it is a matrix) can be integrated one by one still using Mupad or even Scilab.*

Choice of the sampling period

For the study of sampled systems, the choice of the sampling period Δ is crucial because:

- if Δ is too large, the computer may "miss" important information about the evolution of the system. Between two samples, the system may, for example, have oscillated and the computer will not be aware of this behaviour.
- if Δ is too small, the calculator may be solicited too often. If the system only evolves a little between two successive samples, the information received does not bring much and the computer processor is then unnecessarily solicited.

The choice of the sampling period Δ depends directly on the dynamics of the system. The sampling frequency f_e ($f_e = \frac{1}{\Delta}$) must in fact respect Shannon's theorem, namely:

$$f_e \geq 2f_h$$

where f_h is the highest frequency to be kept in the signal. In practice, the following rule is often applied in Automatic control :

$$5f_h < f_e < 25f_h \ .$$

On the one hand, we "oversample" in relation to the limit prescribed by Shannon's theorem. On the other hand, we try to limit the sampling frequency so as not to overload the computer. We then have :

	First order system	Second order system
	$H(s) = \frac{K}{1+Ts}$	$H(s) = \frac{K}{1+\frac{2\xi}{\omega_n}s+\frac{1}{\omega_n^2}s^2}$
With	$f_h \approx \frac{1}{2\pi T}$	$f_h \approx \frac{\omega_n}{2\pi}$
We obtain	$0.25T < \Delta < 1.25T$	$0.25 < \Delta\omega_n < 1.25$

Example 12 *Consider again the example of the harbour crane described in the example 1 with the state representation*

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & \begin{pmatrix} 0 & g \\ -\frac{1}{L} & 0 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ \frac{1}{L} \end{pmatrix} u(t) \\ y(t) & = & \begin{pmatrix} 1 & 0 \end{pmatrix} x(t) \end{array} \right. .$$

It has also been established that this is a second order system whose natural frequency is given by $\omega_n = \sqrt{\frac{g}{L}}$. The sampling period Δ can therefore be chosen in the interval $]0.25\sqrt{\frac{L}{g}}, 1.25\sqrt{\frac{L}{g}}[$.
We calculate the state representation of the system sampled at period Δ :

- *We use the formula (cf. appendices) $e^{At} = \mathcal{L}^{-1}\left[(s\mathbf{I}-A)^{-1}\right]$. Note that we have already computed $(s\mathbf{I}-A)^{-1}$ in the example 9. We have*[7] *:*

$$e^{At} = \mathcal{L}^{-1}\left[\begin{pmatrix} \frac{s}{s^2+g/L} & \frac{g}{s^2+g/L} \\ -\frac{1/L}{s^2+g/L} & \frac{s}{s^2+g/L} \end{pmatrix}\right] = \begin{pmatrix} \cos\left(\sqrt{\frac{g}{L}}t\right) & \sqrt{gL}\sin\left(\sqrt{\frac{g}{L}}t\right) \\ -\frac{1}{\sqrt{gL}}\sin\left(\sqrt{\frac{g}{L}}t\right) & \cos\left(\sqrt{\frac{g}{L}}t\right) \end{pmatrix}$$

[7] As a reminder, the Laplace transform of:

- $\sin(\omega t)$ is $\frac{\omega}{s^2+\omega^2}$;
- $\cos(\omega t)$ is $\frac{s}{s^2+\omega^2}$.

Hence

$$A_{ech} = \begin{pmatrix} \cos\left(\sqrt{\frac{g}{L}}\Delta\right) & \sqrt{gL}\sin\left(\sqrt{\frac{g}{L}}\Delta\right) \\ -\frac{1}{\sqrt{gL}}\sin\left(\sqrt{\frac{g}{L}}\Delta\right) & \cos\left(\sqrt{\frac{g}{L}}\Delta\right) \end{pmatrix}$$

- *We calculate*[8] $B_{ech} = \int_0^\Delta e^{A(\Delta-\nu)} d\nu B$ *:*

$$\begin{array}{rcl} B_{ech} & = & \int_0^\Delta \begin{pmatrix} \cos\left(\sqrt{\frac{g}{L}}\Delta - \sqrt{\frac{g}{L}}\nu\right) & \sqrt{gL}\sin\left(\sqrt{\frac{g}{L}}\Delta - \sqrt{\frac{g}{L}}\nu\right) \\ -1/\sqrt{gl}\sin\left(\sqrt{\frac{g}{L}}\Delta - \sqrt{\frac{g}{L}}\nu\right) & \cos\left(\sqrt{\frac{g}{L}}\Delta - \sqrt{\frac{g}{L}}\nu\right) \end{pmatrix} d\nu B \\ & = & \left[\begin{pmatrix} -\sqrt{\frac{g}{L}}\sin\left(\sqrt{\frac{g}{L}}\Delta - \sqrt{\frac{g}{L}}\nu\right) & -L\cos\left(\sqrt{\frac{g}{L}}\Delta - \sqrt{\frac{g}{L}}\nu\right) \\ \frac{1}{g}\cos\left(\sqrt{\frac{g}{L}}\Delta - \sqrt{\frac{g}{L}}\nu\right) & -\sqrt{\frac{g}{L}}\sin\left(\sqrt{\frac{g}{L}}\Delta - \sqrt{\frac{g}{L}}\nu\right) \end{pmatrix}\right]_0^\Delta B \\ & = & \begin{pmatrix} -1+\cos\left(\sqrt{\frac{g}{L}}\Delta\right) \\ \frac{1}{\sqrt{(gL)}}\sin\left(\sqrt{\frac{g}{L}}\Delta\right) \end{pmatrix} \end{array}$$

- $C_{ech} = C = \begin{pmatrix} 1 & 0 \end{pmatrix}$

3.5 State feedback control

In this chapter, we will study the design of controllers for systems represented by the state equations

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) \end{array} \right. \tag{3.37}$$

in continuous time, or

$$\left\{ \begin{array}{rcl} x(k+1) & = & Ax(k) + Bu(k) \\ y(k) & = & Cx(k) \end{array} \right. \tag{3.38}$$

in discrete time. Note that we consider here that the direct matrix D is zero, or that the output is not a direct function of the input. In practice, this situation is common.

The state feedback control method consists in the elaboration of a control signal u from the state variables x_i, initially assumed to be all measured (their values are permanently accessed). This assumption is not very realistic, but it allows us to explain the control principle directly and simply. We will see in the next chapter that if some state variables are not accessible, then, under the assumption of observability, it is possible to deduce them from the output using a state *observer* or *estimator*.
We therefore start with the case where all the state variables are accessible, and we then have the control structure shown in Figure 3.4 (we present the case in discrete time, we have the analogue in continuous time) :

- The variable $e(k)$ is the external setpoint (reference input) sent to the system.

- The control law applied to the system is given by:

$$u(k) = e(k) - L \cdot x(k)\ .$$

  It takes into account the external set point $e(k)$ and the behaviour of the system *via* $L \cdot x(k)$.

- The matrix L is the *regulation matrix* that we will try to calculate.

[8] As a reminder, the derivative of:

- $\frac{1}{\omega}\sin(\omega t + \phi)$ is $\cos(\omega t + \phi)$;
- $-\frac{1}{\omega}\cos(\omega t + \phi)$ is $\sin(\omega t + \phi)$.

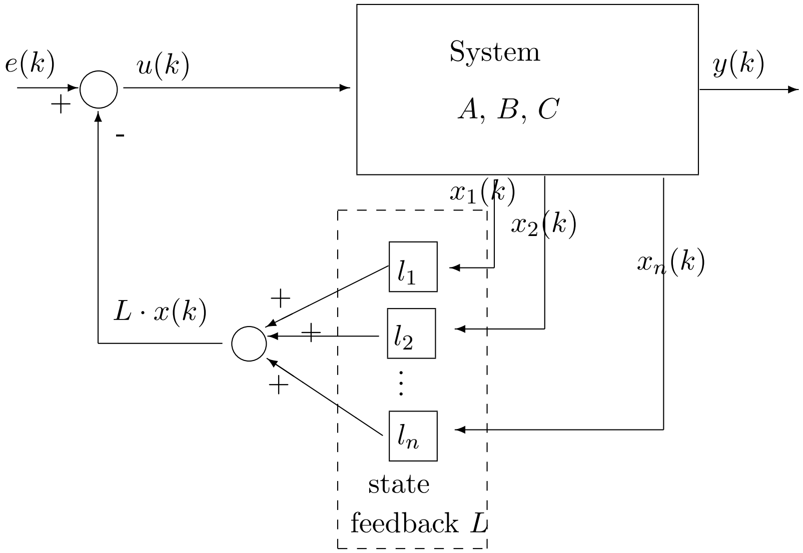


Figure 3.4: Structure of a state feedback control

3.5.1 Pole placement control

In continuous time

With state feedback, the control law applied to the system is given by:

$$u(t) = e(t) - L \cdot x(t)\,.$$

The evolution equation of the looped system is then written

$$\begin{array}{rcl} \dot{x}(t) & = & Ax(t) + B\left(e(t) - Lx(t)\right) \\ & = & (A - BL)x(t) + Be(t) \end{array} \tag{3.39}$$

The principle of pole placement adopted here is to choose the control matrix L so as to impose the poles of the looped system. In other words, the dynamics of the system are forced to ensure a chosen stability and speed.

Fixing the poles of the looped system is equivalent to imposing the characteristic polynomial of the system. Let $P_{des}(s)$ be the desired characteristic polynomial for the looped system (we choose it of degree n), we have to solve the polynomial equation

$$|s\mathbf{I} - A + BL| = P_{des}(s) \tag{3.40}$$

known as *pole placement.*
This equation can be solved directly in Scilab using the function `ppol()`.
For a more formal answer, we restrict ourselves to the case where the system has a single input[9]. Equation (3.40) can be translated into n scalar equations with n unknowns, and the problem then has a unique solution. Recall that two polynomials of degree n

$$s^n + \gamma_{n-1}s^{n-1} + \ldots + \gamma_0$$

and

$$s^n + \delta_{n-1}s^{n-1} + \ldots + \delta_0$$

are equal if, and only if, their coefficients are all equal, that is, if

$$\gamma_0 = \delta_0,\ \gamma_1 = \delta_1, \ldots,\ \gamma_{n-1} = \delta_{n-1}.$$

[9] We refer the reader to the works cited in the bibliography for the (more complex) multi-input case.

The solution of the problem is easy when the state representation of the system is in the canonical controllable form[10] (*cf.* § 3.2.3), i.e. when A and B have the following shapes

$$A = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1 \\ -a_0 & -a_1 & \dots & \dots & -a_{n-1} \end{pmatrix}, B = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} .$$

With

$$L = \begin{pmatrix} l_0 & l_1 & l_2 & \dots & l_{n-1} \end{pmatrix} ,$$

it is easy to obtain

$$|s\mathbf{I} - A + BL| = s^n + (a_{n-1} + l_{n-1})s^{n-1} + \ldots + (a_1 + l_1)s + (a_0 + l_0) .$$

The a_0, a_1, …, a_{n-1} are known coefficients of the system model. It is then sufficient to identify the l_0, l_1, …, l_{n-1} so that

$$|s\mathbf{I} - A + BL| = P_{des}(s) ,$$

that is

$$s^n + (a_{n-1} + l_{n-1})s^{n-1} + \ldots + (a_1 + l_1)s + (a_0 + l_0) = s^n + \alpha_{n-1}s^{n-1} + \ldots + \alpha_1 s + \alpha_0 ,$$

where α_0, α_1, …, α_{n-1} are the chosen coefficients of $P_{des}(s)$. These reflect the poles chosen for the looped system in order to ensure its desired performance. In other words, the control will allow the corrected system to have $P_{des}(s)$ as the denominator of its transfer function.

- In the case of a first order system, the denominator of the transfer function is written $1 + Ts$ or $1/T + s$, and therefore the choice of α_0 in the polynomial $P_{des}(s)$ will make it possible to fix the desired value for the time constant T (which expresses the "rapidity" of the system)

- In the case of a second order system, the denominator of the transfer function is written $1+\frac{2\xi}{\omega_n}s+\frac{1}{\omega_n^2}s^2$ or $\omega_n^2+2\xi\omega_n s+s^2$, and thus the choice of α_0 and α_1 in the polynomial $P_{des}(s)$ will allow to fix the desired values for the natural frequency ω_n and the damping ratio ξ.
  For the control to be physically feasible, the coefficients l_0 and l_1 must be real numbers. We must then choose α_0 and α_1 such that they are real numbers (the roots of $P_{des}(s)$ can be conjugated complexes).
  To ensure stability, ξ must be strictly positive (which amounts to choosing roots with a strictly negative real part for $P_{des}(s)$). If we want the looped system to behave like a damped second order resonant system ($0 < \xi < 1$), then the value of ξ conditions the amplitude of the overshoot.
  For a fixed value of ξ, the choice of the natural frequency will make it possible to adjust the "speed" of the system (for example to impose the value of the response time at 5%).

- For systems of orders greater than or equal to 3, we can put $P_{des}(s)$ in the form of a factorisation of polynomials of orders 1 and 2, and then reason for each polynomial as above.

Example 13 *Consider again the example of the harbour crane described in the example 1 with the state representation*

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & \begin{pmatrix} 0 & g \\ -\frac{1}{L} & 0 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ \frac{1}{L} \end{pmatrix} u(t) \\ y(t) & = & \begin{pmatrix} 1 & 0 \end{pmatrix} x(t) \end{array} \right. .$$

It was also established that this was a second-order system whose damping ratio is zero and whose natural frequency is given by $\omega_n = \sqrt{\frac{g}{L}}$. The example 9 has shown that this system is unstable.

We will apply a state feedback control in order to impose the poles of the corrected system. More precisely, we want the feedback-corrected system to have the characteristic polynomial :

$$P_{des}(s) = (s + 2)^2$$

This choice can be justified by the following explanations.

[10] Note that if the system is controllable, then it is always possible to find a similar state representation (see works cited in bibliography).

a) The roots of the characteristic polynomial of the corrected system (in this case the -2 double root) will then all have a strictly negative real part, and the system will therefore be stabilised.

b) More precisely, this amounts to imposing s^2+4s+4, or $\frac{1}{4}s^2+s+1$ as the denominator of the transfer function. By identification, this amounts to imposing :

- *$\xi = 1$ the corrected system will be free of overshoot.*
- *$\omega_n = 2$ and the rise time for a step input can be predicted using $t_m = \frac{1}{\omega_n\sqrt{1-\xi^2}}(\pi - \arccos\xi)$.*

Let us determine the control matrix L allowing to obtain $P_{des}(s)$ as the characteristic polynomial of the corrected system. We want to solve the pole placement problem :

$$|s\mathbf{I} - A + BL| = P_{des}(s).$$

$$s\mathbf{I} - A + BL = \begin{pmatrix} s & -g \\ \frac{1+l_1}{L} & s+\frac{l_2}{L} \end{pmatrix}$$

We obtain $|s\mathbf{I} - A + BL| = s^2 + s\frac{l_2}{L} + \frac{g+l_1 g}{L}$, and by identification we deduce

$$\begin{cases} l_1 &= \frac{4L}{g} - 1 \\ l_2 &= 4L \end{cases}$$

On the figure 3.5, we represented the step response of the system (trolley-cables-container assembly) corrected by this feedback (we took $\frac{L}{g} = 0,125$, $L = 1,25$).

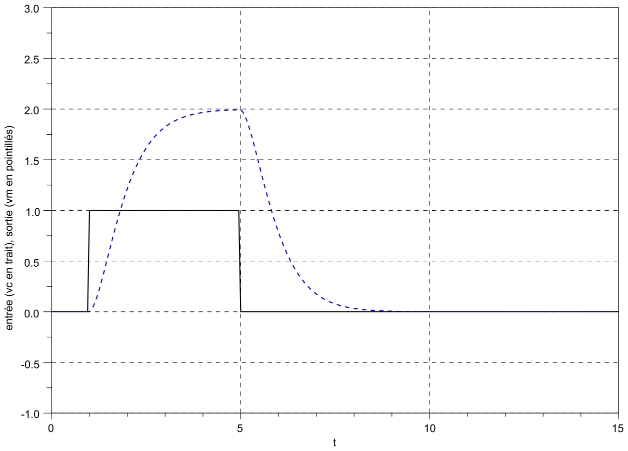


Figure 3.5: Step response of the system (trolley-cable-container assembly) corrected by a state feedback

In discrete time

The approach is the same as in continuous time. With state feedback, the control law applied to the system is given by:

$$u(k) = e(k) - L \cdot x(k)\,.$$

The evolution equation of the looped system is then written

$$\begin{aligned} x(k+1) &= Ax(k) + B\left(e(k) - Lx(k)\right) \\ &= (A-BL)x(k) + Be(k) \end{aligned} \tag{3.41}$$

The principle of pole placement adopted here is to choose the control matrix L so as to impose the poles of the looped system. This problem is equivalent to imposing the characteristic polynomial of the system. Let $P_{des}(z)$ be the desired characteristic polynomial for the looped system (we choose it of degree n), we have to solve the polynomial equation

$$|z\mathbf{I} - A + BL| = P_{des}(z) \tag{3.42}$$

said to be of *pole placement*.
This equation can be solved directly in Scilab using the function `ppol()`.
If we restrict ourselves to the case where the system has a single input[11], equation (3.42) can be translated into n scalar equations with n unknowns. There is then a unique solution, and, exactly as in continuous time, it can be made explicit directly if the state representation of the system is in the canonical controllable form (*cf.* § 3.4.5).

The choice of the coefficients of $P_{des}(z)$ will allow to fix the desired behaviour of the looped system.

- In the case of a first order system, $P_{des}(z)$ is of the form

$$P_{des}(z) = z + \alpha_0$$

where $\alpha_0 = -e^{-\frac{\Delta}{T}}$ and thus allows the desired value for the time constant T (which reflects the "speed" of the system) to be set.

- In the case of a second order system, $P_{des}(z)$ is of the form

$$P_{des}(z) = z^2 + \alpha_1 z + \alpha_0$$

where $\alpha_1 = -2e^{-\xi\omega_n\Delta}\cos(\omega_n\sqrt{1-\xi^2}\Delta$ and $\alpha_0 = e^{-2\xi\omega_n\Delta}$ for a chosen damping ratio ξ strictly less than 1.

3.5.2 Static regime of the looped system

In this section, we are interested in the behaviour of the system in *steady state*, i.e., when its state does not evolve any more, or when a constant reference input e is considered and the time (t or k) tends towards infinity. We will first deduce an expression for the static gain of the looped system. In a second step, we will be able to propose a control system that slaves the output to a non-zero value.

In continuous time

Let us denote

$$\begin{aligned} e &= \lim_{t\to\infty} e(t), \\ x &= \lim_{t\to\infty} x(t), \\ y &= \lim_{t\to\infty} y(t). \end{aligned}$$

The evolution equation of the closed system (3.39) and the static output equation can be written :

$$\left\{ \begin{array}{lcl} 0 &=& (A-BL)x + Be \\ y &=& Cx \end{array} \right. \tag{3.43}$$

In the static regime, we have $\dot{x}(t) = 0$. The first equation gives

$$x = -(A-BL)^{-1}Be.$$

By transferring this expression to the output equation, the result is

$$y \quad = \quad -C(A-BL)^{-1}Be. \tag{3.44}$$

In discrete time

The evolution equation of the discrete time looped system is written as (*cf.* §3.5.1) :

$$x(k+1) = (A-BL)x(k) + Be(k)$$

A steady state is reached as soon as for $k > K$, K corresponding to the instant of the beginning of the static regime, we have $x(k+1) = x(k)$. This gives

$$x(k) = (A-BL)x(k) + Be(k),$$

or even

$$x(k) = (\mathbf{I} - A + BL)^{-1}Be(k).$$

Transferring this expression to the output equation, and noting y and e as the signals $y(k)$ and $u(k)$ (constant in steady state), we obtain

$$y = C(\mathbf{I} - A + BL)^{-1}Be. \tag{3.45}$$

[11] We refer the reader to the works cited in the bibliography for the (more complex) multi-input case.

Achieving a unitary gain

From equations (3.44) and (3.45), the value of the static gain can be directly deduced $\frac{y}{e}$, i.e. for the looped system:

$$\begin{array}{ll} -C(A-BL)^{-1}B & \text{in continuous time,} \\ C(\mathbf{I}-A+BL)^{-1}B & \text{in discrete time.} \end{array}$$

In order to ensure a unitary gain between the reference input e and the output y, a gain K_{re} of value equal to the inverse of the latter can be added to the control structure. This is placed upstream of the state feedback as shown in figure 3.6 in discrete time.

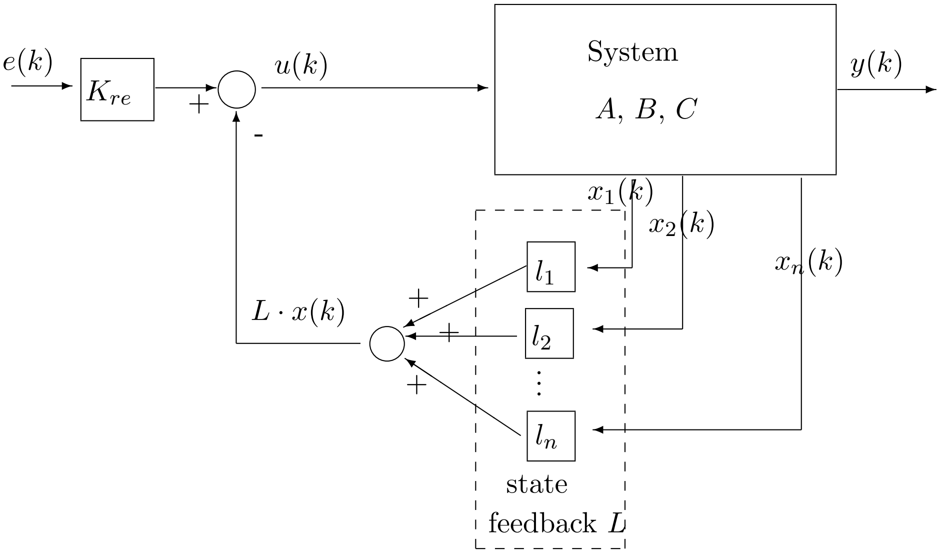


Figure 3.6: A gain is added upstream of the state feedback

Example 14 *Consider again the example of the harbour crane described in the example 1 with the state representation*

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & \begin{pmatrix} 0 & g \\ -\frac{1}{L} & 0 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ \frac{1}{L} \end{pmatrix} u(t) \\ y(t) & = & \begin{pmatrix} 1 & 0 \end{pmatrix} x(t) \end{array} \right. .$$

In example 13, we have determined the control matrix L allowing to obtain $P_{des}(s) = s^2 + 4s + 4$ as the characteristic polynomial of the corrected system, namely:

$$L = \begin{pmatrix} \frac{4L}{g} - 1 & 4L \end{pmatrix}$$

The gain (in static regime) of the looped system is $-C(A-BL)^{-1}B$. We have

$$A - BL = \begin{pmatrix} 0 & g \\ -\frac{4}{g} & -4 \end{pmatrix} \text{, d'où } (A-BL)^{-1} = \begin{pmatrix} -1 & -\frac{g}{4} \\ \frac{1}{g} & 0 \end{pmatrix}$$

and

$$-C(A-BL)^{-1}B = \frac{g}{4L}.$$

The gain to be added to obtain a unit gain for the looped system is therefore $K_{re} = \frac{1}{-C(A-BL)^{-1}B} = \frac{4L}{g}$.

Controlling the output to a non-zero constant value

In the case where the system is not subject to any external disturbance, the objective of a control system may be to bring the system (and in particular its outputs) to equilibrium points (desired output values). If the objective of the control is in particular to bring the output to a constant value y_s, the result obtained for the steady state directly provides the reference input to be applied to the system input. Indeed:

- in continuous time, we want $y = \lim_{t\to\infty} y(t) = y_s$, and we seek for $e(t) = e_s$ suitable. Equation (3.44) allows us to establish directly the expression of e_s, namely

$$e_s = [-C(A - BL)^{-1}B]^{-1}y_s,$$

- in discrete time, we obtain from (3.45)

$$e_s = [C(\mathbf{I} - A + BL)^{-1}B]^{-1}y_s.$$

3.5.3 Adding an integral effect

In conventional control, the cancellation of the static error in response to a step is carried out using an integral corrector. It is possible to implement a similar correction in state space. We consider the system $x(t) = Ax(t) + Bu(t) + p$ where p is an unknown and constant disturbance vector, supposed to represent an external disturbance which could not be taken into account in the modelling. A feedback controller with integral effect is of the form

$$u(t) = \underbrace{L_i \int_0^t (e(t) - y(t))dt}_{\text{integral effect}} - \overbrace{Lx(t)}^{\text{state feedback}},$$

assuming that the number of output(s) is equal to the number of input(s) (y and e have the same dimensions). The closed loop system can therefore be written as :

$$\left\{\begin{array}{rcl} \dot{x}(t) & = & Ax(t) + Bu(t) + p \\ \dot{z}(t) & = & e(t) - y(t) \\ u(t) & = & L_i z(t) - Lx(t) \end{array}\right. .$$

The state equations of the looped system can also be written as

$$\left\{\begin{array}{rcl} \dot{x}(t) & = & (A - BL)x(t) + BL_i z(t) + p \\ \dot{z}(t) & = & e(t) - y(t) \end{array}\right.$$

or, in matrix form:

$$\begin{pmatrix} \dot{x}(t) \\ \dot{z}(t) \end{pmatrix} = \begin{pmatrix} A - BL & BL_i \\ -C & 0 \end{pmatrix} \begin{pmatrix} x(t) \\ z(t) \end{pmatrix} + \begin{pmatrix} p \\ e(t) \end{pmatrix}.$$

Since

$$\begin{pmatrix} A - BL & BL_i \\ -C & 0 \end{pmatrix} = \begin{pmatrix} A & 0 \\ -C & 0 \end{pmatrix} - \begin{pmatrix} B \\ 0 \end{pmatrix} \begin{pmatrix} L & -L_i \end{pmatrix},$$

the evolution matrix of the looped system is still of the form $A' - B'L'$ (*cf.* eq. (3.39)), and the choice of L and L_i is then also made by poles placement (Scilab's function `ppol()` can be used).

3.6 State observer

In the previous chapter, we assumed that all the state variables of the process were measured to compute the control.
Most of the time, either because of the physical impossibility of introducing a sensor or because of cost issues, not all states can be measured.
We will see how one can, from measurements of the input and output of the process, estimate the complete state vector x, noted then $\hat{x}$. The subsystem which carries out this estimation is called *state observer.*

For the control by state feedback, we place ourselves in the context schematised in the figure 3.7. It is the estimated state $\hat{x}$ which is transmitted to the control matrix L to elaborate the state feedback, and the control law to apply is

$$u = e - L\hat{x}.$$

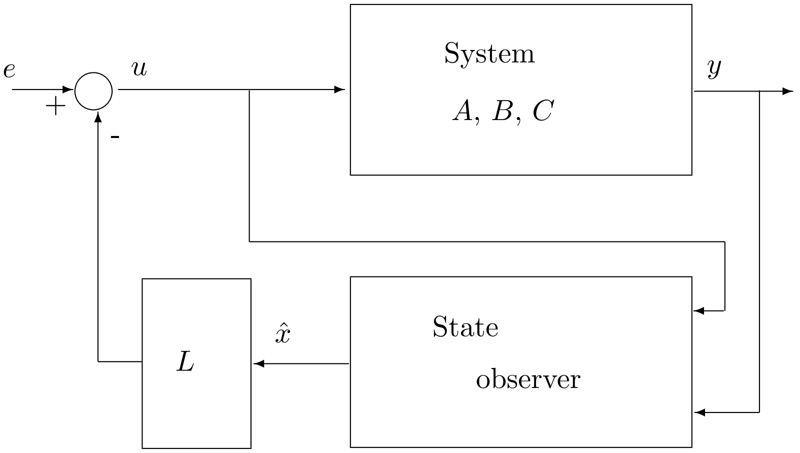


Figure 3.7: Structure of a state feedback control with a state observer

3.6.1 Principle of the state observer

Consider a process described by the matrices A, B and C of its state representation.
If the process is observable, it is possible from the input and output measurements to estimate the state of the system (*cf.* §3.3.7). The principle of the estimation is then schematized in the figure 3.8.

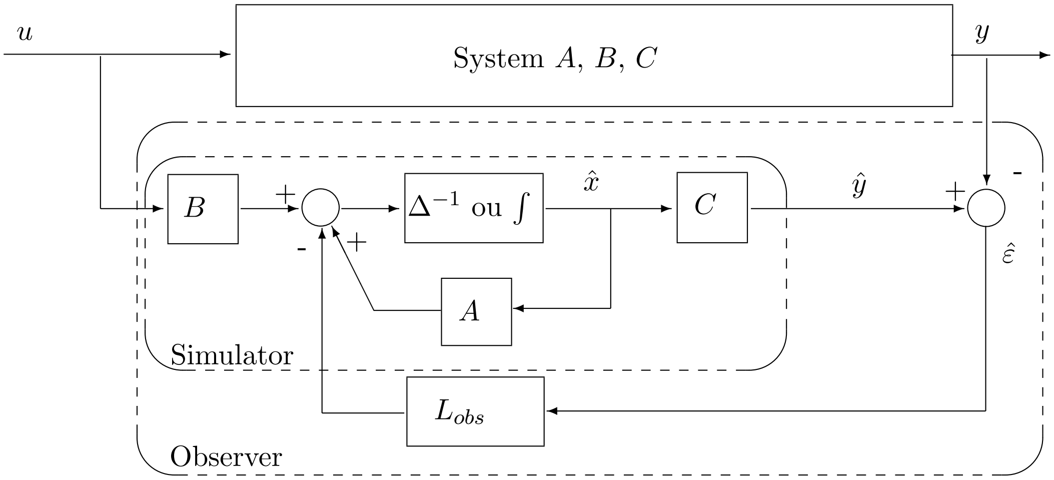


Figure 3.8: Principle of the state observer

The device integrates a *simulator* of our system. The ε error between the output of the simulator $\hat{y}$ and the output of the y system allows us to correct, by means of a correction matrix L_{obs}, the evolution of the estimated state $\hat{x}$. The corrected simulator is called *observer*. Its role is to give us a good estimate of the state vector x of the system to be transmitted to the matrix L of the state feedback regulation.

3.6.2 Synthesis of the matrix L_{obs}

In continuous time

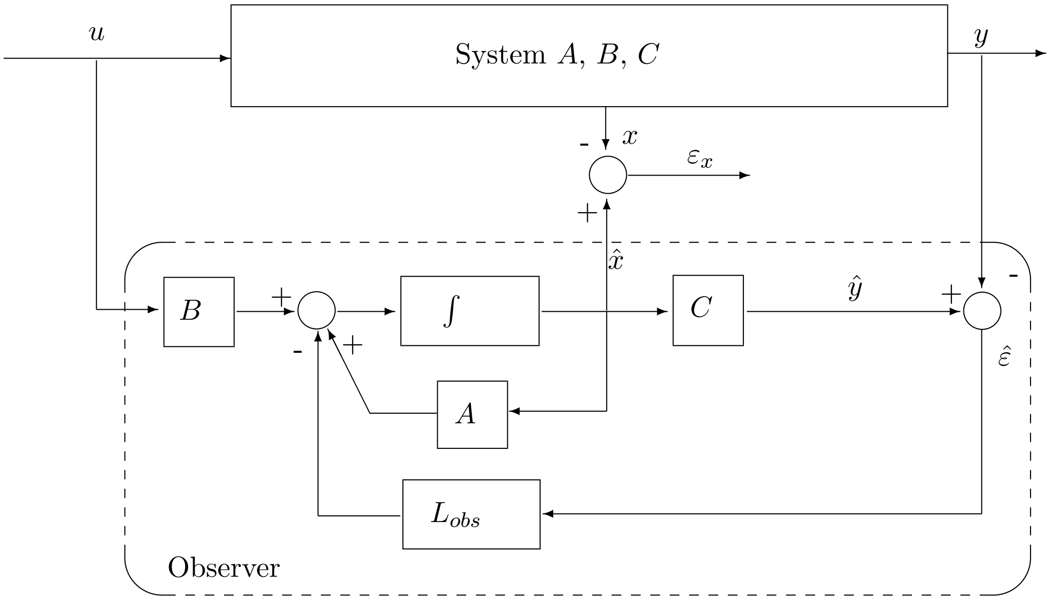


Figure 3.9:

In order to calculate L_{obs}, let us extract from the figure 3.9 the subsystem with input $x(t)$ and output $\varepsilon_x(t) = \hat{x}(t) - x(t)$. This output is not measurable because it depends on x but we abstract it for the purpose of the following developments. It will only be used to calculate L_{obs} and will not be used by the controller.

We have

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & Ax(t) + Bu(t) \\ \frac{d}{dt}(\hat{x}(t)) & = & A\hat{x}(t) + Bu(t) - L_{obs}(C\hat{x}(t) - Cx(t)) \end{array} \right. .$$

hence

$$\begin{array}{rcl} \dot{\varepsilon}_x(t) = \frac{d}{dt}(\hat{x}(t) - x(t)) & = & A\hat{x}(t) + Bu(t) - L_{obs}(C\hat{x}(t) - Cx(t)) - Ax(t) - Bu(t) \\ & = & A(\hat{x}(t) - x(t)) - L_{obs}C(\hat{x}(t) - x(t)) \end{array} ,$$

or even

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & Ax(t) + Bu(t) \\ \dot{\varepsilon}_x(t) & = & A\varepsilon_x(t) - L_{obs}C\varepsilon_x(t) \end{array} \right. .$$

By putting into matrix form, we arrive at the following state representation for the system with input $x(t)$ and output $\varepsilon_x(t) = \hat{x}(t) - x(t)$:

$$\left\{ \begin{array}{rcl} \begin{pmatrix} \dot{x}(t) \\ \dot{\varepsilon}_x(t) \end{pmatrix} & = & \begin{pmatrix} A & 0 \\ 0 & A - L_{obs}C \end{pmatrix} \begin{pmatrix} x(t) \\ \varepsilon_x(t) \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix} u(t) \\ \varepsilon_x(t) & = & \begin{pmatrix} 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} x(t) \\ \varepsilon_x(t) \end{pmatrix} \end{array} \right. . \tag{3.46}$$

The poles of this system correspond to the eigenvalues of A and $A - L_{obs}C$.
The command $u(t)$ does not intervene in the evolution of the sub-state vector $\varepsilon_x(t)$ (governed by $\dot{\varepsilon}_x(t) = (A - L_{obs}C)\varepsilon_x(t)$). The *estimation error* $\varepsilon_x(t)$ tends to zero if all eigenvalues of $A - L_{obs}C$ have strictly negative real parts (see §3.3.5). Imposing the dynamics of this error (in particular imposing the speed of convergence to zero) amounts to solving

$$|s\mathbf{I} - A + L_{obs}C| = P_{obs}(s),$$

where $P_{obs}(s)$ is chosen. Since the determinant of a matrix is equal to that of its transpose, this equation is equivalent to

$$|s\mathbf{I} - A^\top + C^\top L_{obs}^\top| = P_{obs}(s), \tag{3.47}$$

and we obtain an equation of the pole placement type (*cf.* equation (3.40)). The same method as described in paragraph 3.5.1 is therefore applied to obtain $L_{obs}^\top$ and therefore L_{obs} (e.g. using ”ppol()” in Scilab).

In discrete time

The approach is identical to that for continuous time. For the input subsystem $u(k)$ and the output subsystem $varepsilon_x(k)$, we can verify that we end up with the following state representation

$$\left\{\begin{array}{rcl}\begin{pmatrix}x(k+1)\\ \varepsilon_x(k+1)\end{pmatrix} & = & \begin{pmatrix}A & 0\\ 0 & A-L_{obs}C\end{pmatrix}\begin{pmatrix}x(k)\\ \varepsilon_x(k)\end{pmatrix}+\begin{pmatrix}B\\ 0\end{pmatrix}u(k)\\ \varepsilon_x(k) & = & \begin{pmatrix}0 & \mathbf{I}\end{pmatrix}\begin{pmatrix}x(k)\\ \varepsilon_x(k)\end{pmatrix}\end{array}\right. . \tag{3.48}$$

It can be stated that the reconstruction error $\varepsilon_x(k)$ tends to zero if all eigenvalues of $A - L_{obs}C$ are all strictly in the unit circle. Here again, we apply a pole placement technique to fix the coefficients of L_{obs}.

3.6.3 Separation principle

It can be shown that the poles of the closed-loop system with a state observer are, on the one hand, the poles chosen for the design of the controller, and on the other hand, the poles chosen for the state observer. This independence allows us to determine separately the feedback correction (L) and the estimation (L_{obs}).

Brief reminders of linear algebra

Consider a square matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

Calculation of the determinant

If A is of dimension 2×2

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

We have

$$|A| = a_{11}a_{22} - a_{21}a_{12}$$

For higher dimensions, the following properties can be applied

- a determinant is not modified when a linear combination of the other lines is added to a line,
- when two lines of a determinant are exchanged, the determinant is changed into its opposite,

to rewrite the determinant in the form:

$$|A| = \begin{vmatrix} a'_{11} & a'_{12} & \dots & a'_{1n} \\ 0 & a'_{22} & \dots & a'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a'_{n2} & \dots & a'_{nn} \end{vmatrix}$$

We then have

$$|A| = a'_{11} \cdot \begin{vmatrix} a'_{22} & \dots & a'_{2n} \\ \vdots & \ddots & \vdots \\ a'_{n2} & \dots & a'_{nn} \end{vmatrix}$$

More generally, we can apply an expansion along the j-column of A as follows:

$$|A| = \sum_{i=1}^{n} a_{ij}\mathrm{Cof}_{ij}$$

with

$$\mathrm{Cof}_{ij} = (-1)^{i+j} M_{ij} \tag{3.49}$$

where M_{ij} represents the determinant of the matrix resulting from the deletion of row i and column j in the matrix A, i.e.

$$M_{11} = \begin{vmatrix} a_{22} & \dots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n2} & \dots & a_{nn} \end{vmatrix}, \dots M_{1n} = \begin{vmatrix} a_{21} & \dots & a_{2n-1} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn-1} \end{vmatrix}$$

$$M_{n1} = \begin{vmatrix} a_{12} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n-12} & \dots & a_{n-1n-1} \end{vmatrix} \dots M_{nn} = \begin{vmatrix} a_{11} & \dots & a_{1n-1} \\ \vdots & \ddots & \vdots \\ a_{n-11} & \dots & a_{n-1n-1} \end{vmatrix}$$

Calculation of the inverse matrix

- The matrix A is invertible if and only if, $|A| \neq 0$.
- If A is invertible, then

$$A^{-1} = \frac{1}{|A|}(\mathrm{Com}A)^\top \ , \ \text{avec} \ (\mathrm{Com}A)_{ij} = \mathrm{Cof}_{ij} \ \text{(cf. equ. (3.49))}.$$

Eigenvalues of a matrix

- We call *eigenvalue* and *associated eigenvector* of a square matrix A, a scalar λ and a non zero vector v such that:

$$A \cdot v = \lambda v \ .$$

- The eigenvalues of a square matrix A are the roots of the equation:

$$|\lambda \mathbf{I} - A| = 0 \ ,$$

  called *characteristic equation* of A ; $|\lambda \mathbf{I} - A|$ is the characteristic polynomial of A.

- If A and D are two matrices such that $P^{-1}AP = D$ (then $AP = PD$) and if D is a diagonal matrix, then multiplying P on the right by D is the same as multiplying the columns of P by the diagonal coefficients of D. Let $P_{.i}$ be the ith column of the matrix P and D_{ii} the ith diagonal coefficient of D. For all $i = 1, \ldots, n$, we have:

$$AP_{.i} = D_{ii}P_{.i}.$$

  $P_{.i}$ is therefore an eigenvector of A associated with the eigenvalue D_{ii}.

Matrix exponentials

Consider square matrices M and N of dimension n. We note $\mathbf{0}$ and $\mathbf{I}$ the null and identity matrices of dimension n.

- The exponential e^M is of the same dimension as the matrix M.

- The exponential of a matrix follows the same rules of calculation as the exponential function, in particular:

$$\begin{aligned} e^{\mathbf{0}} &= \mathbf{I} && (3.50) \\ e^M \cdot e^N &= e^{M+N} && (3.51) \\ \frac{d}{dt}\left(e^{Mt}\right) &= Me^{Mt} && (3.52) \end{aligned}$$

- The exponential of a matrix can be calculated from its integer series expansion, namely:

$$e^M = \mathbf{I} + M + \frac{1}{2!}M^2 + \frac{1}{3!}M^3 + \ldots = \sum_{i=0}^{\infty} \frac{1}{i!}M^i \ . \qquad (3.53)$$

  When the powers M^i become zero or negligible, we can evaluate the expression (3.53) by a finite summation of terms. The calculation is simplified if the matrix M is diagonal, because then only the successive powers of the diagonal elements remain.

- The exponential of a matrix can also be calculated from an inverse Laplace transformation, more precisely:

$$e^{Mt} = \mathcal{L}^{-1}\left[(s\mathbf{I} - M)^{-1}\right] \ . \qquad (3.54)$$

  This method requires the calculation of the inverse of a matrix (Mupad and Scilab software can be used) and its inverse Laplace transformation. Its advantage is to provide an analytical expression.

Rank of a matrix and solving a linear system

- A square matrix is said to be of *full rank* if its determinant is non-zero.

  The *rank* of a matrix R, noted $\text{rank}(R)$, is the maximum size of the square matrices of full rank extracted from R.

- Consider the system

  $$Ax = b$$

  with A a square matrix and b a column vector with as many rows as A.
  If A is of full rank, then the system has a unique solution given by $x = A^{-1}b$.

- Consider the system

  $$Rx = b$$

  with R a rectangular matrix of p rows and q columns, and b a column vector of p rows.

Case $p < q$: if R is of rank p, then the system has infinitely many solutions.
Case $p > q$: if R is of rank p, then the system has at most one solution. If the latter exists, it is given by $x = (R^\top R)^{-1} R^\top b$.

Reminders on second order systems[12]

In continuous time

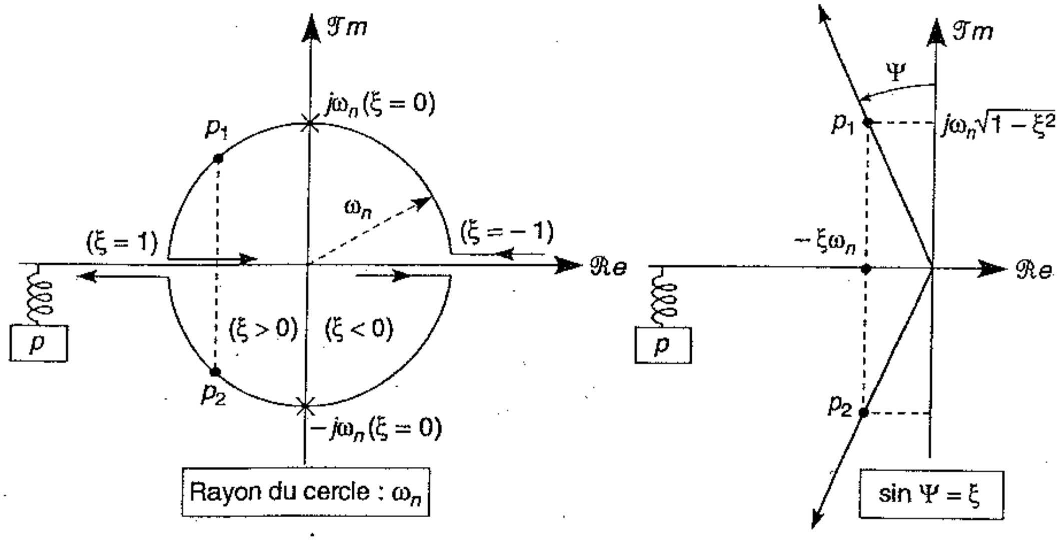


Figure 3.10: Second order system in continuous time: location of poles in the complex plane

[12] D'après [Rivoire et Ferrier]

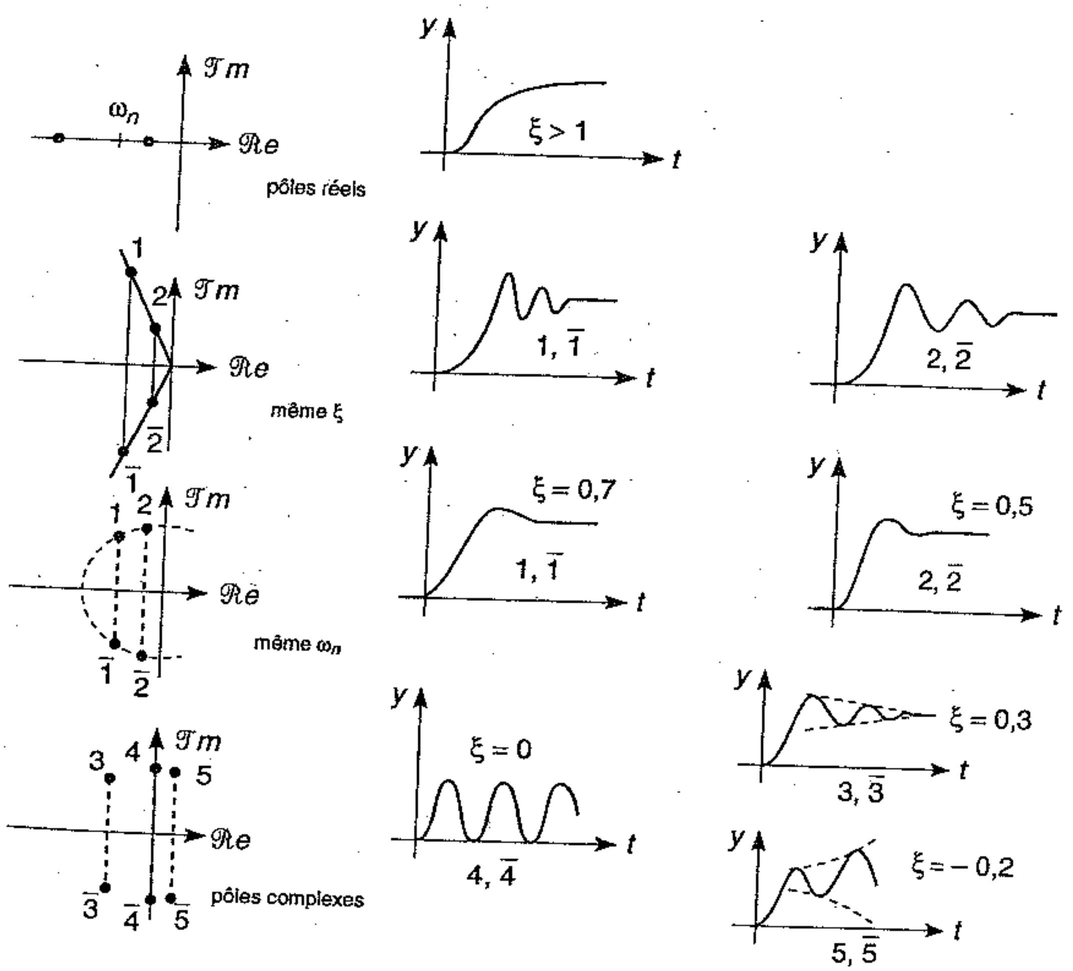


Figure 3.11: Second order system in continuous time: influence of the pole position

In discrete time

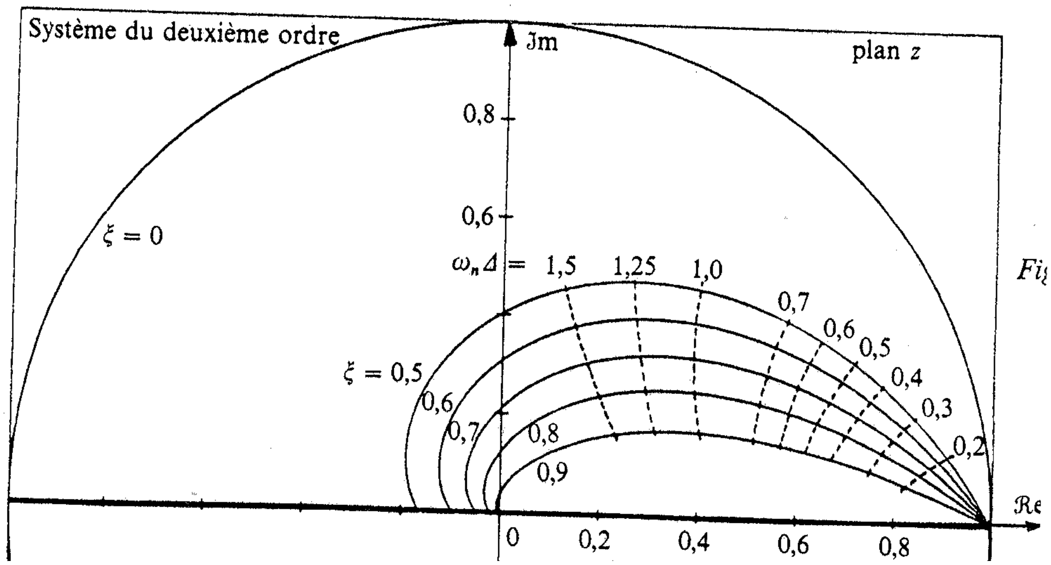


Figure 3.12: Second order system in discrete time: location of poles in the complex plane

Bibliography

[Isermann] R. Isermann, M. Munchoff, Identification of Dynamic systems, Springer, 2011.

[Friedland] B. Friedland, Control System Design: An Introduction to State-Space Methods, 1986.

[Rivoire et Ferrier] M. Rivoire et J.L. Ferrier, *Cours et exercices d'Automatique, Tomes 1, 2 et 3*, Eyrolles, Paris, 1995.

[Ferrier] J.L. Ferrier, *Modèles et systèmes : un tour d'horizon*, Notes de cours, ISTIA.

[Jaulin] L. Jaulin, *Représentation d'état pour la modélisation et la commande des systèmes*, Eyrolles, Paris, 2005.

[Jaulin] L. Jaulin, *Commande par espace d'état*, Notes de cours, Ensieta, 2007 (disponible sur le Web, `http://www.ensieta.fr/e3i2/Jaulin/polyetat.pdf`).

[Borne et al.] P. Borne, G. Dauphin-Tanguy, J.P. Richard, F. Rotella et I. Zambettakis, *Modélisation et identification des processus, Tomes 1 et 2*, Technip, Paris, 1992.

[Alazard] D. Alazard, *Introduction au filtre de Kalman*, Notes de cours, Supaero, 2005.

[Briere] Y. Briere, *Le filtre de Kalman*, Transparents de cours, ENSICA.