1

Field Elements of Classic Control Systems

After reading this chapter, you should be able to understand that:

- The chapter mainly compares different classical theories regarding process control.
- We will focus on systems described in terms of ordinary differential equations for linear and nonlinear processes.
- In addition, it must be emphasized that most ideas, methods, and results presented here do extend to this more general setting, which leads to very important technical developments.

1.1 The Principles of Control (Industry 5.0)

Industry 5.0 is a continuation of Industry 4.0 with the objective of introducing humans (human intelligence) as the main axis of industrial production processes. Furthermore, innovation in production processes is human-oriented and highly customizable based on technological advances and high productivity of systems. However, the concept of Industry 5.0 is not accepted so far by corporations and industries but is promoted by researchers because, today, the industrial situation challenges are still congenital to Industry 4.0 and the era of digitalization. Industry 4.0 encourages high manufacturing efficiency and quality, and focuses on near novelty, techno-economic development, and industrial technology progress [1, 2]. Thus, Industry 5.0 is a prolongation and chronological extension of Industry 4.0 [3]. Industry 4.0 has restrictions with regard to industrial sustainability security, as it emphasizes on the productivity and flexibility of manufacturing through digitalization and technologies and integration of data from operations and business activity. The present manufacturing toward evolution of Industry 4.0 operations allows for better-quality productivity through information-driven automation, not only by infrastructure, but also by introducing more advanced monitoring, modeling, sensors, measurements, and control strategies in real time [4, 5]. One of the main advantages of Industry 4.0 is big data, which relates to large sets of processes and manufacturing data collected by sensors and greater visibility of

Advanced Control Methods for Industrial Processes: Modeling, Design, and Simulation of Emerging Systems in Real Time, First Edition. Edited by Pablo A. López-Pérez et al.

process analytical technologies in the manufacturing operations. The improvements obtained from such a proactive, predictive feed-forward control approach can exceed the incremental yield progresses that corporations seek [6–9]. Furthermore, these data can be used for optimization purposes by applying innovative big data analytics. Machine learning (ML), a branch of artificial intelligence, is one of the ways to accomplish this [10] (see Figure 1.1).

Technological processes consist of handling, working, refining, combining, and manipulating materials and fluids to produce cost-effective end products. These processes can be precise, demanding, and potentially hazardous. Small changes in a process can have a large impact on the end result [14, 15]. Variations in proportions, temperature, flow, turbulence, and many other parameters are to be carefully and consistently controlled to consistently produce the end product of the desired quality with a minimum of raw materials and energy. Instrumentation provides various indications used to operate a technological process [16–18]. In some cases, the operator records these indications for use in the operation of the process. The information recorded helps the operator evaluate the current condition of the process and take action if the conditions are not as expected. Requiring the operator to take all of the necessary corrective actions is impractical, or sometimes impossible, especially if a large number of indications are to be monitored. For this reason, most technological processes are controlled automatically once they are operating under normal conditions [19, 20]. The main role of process control was to contribute to safety, minimize external perturbation influence, and optimize processes by preserving process variables near the desired values. As the processes become larger in scale-up or behavior complex, the role of process automation has become more important.

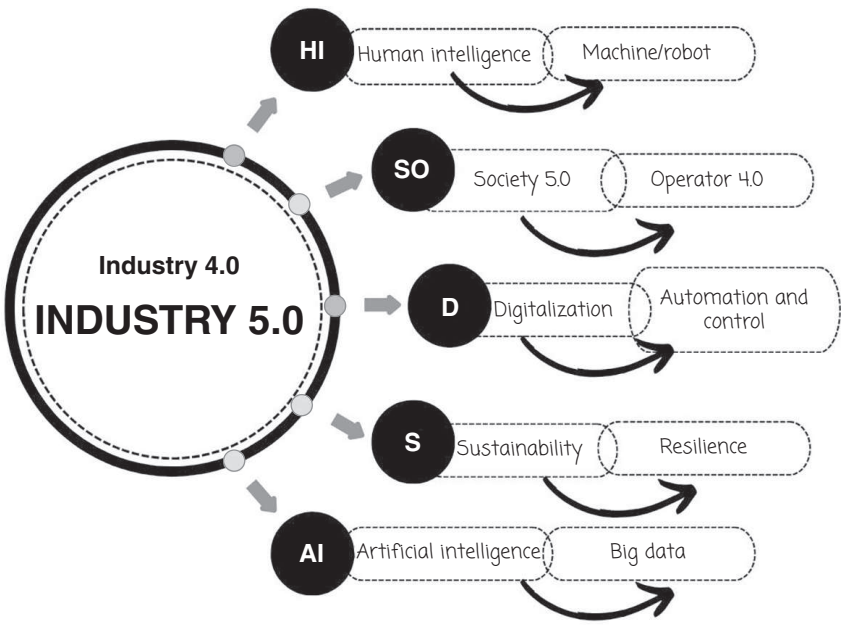

Figure 1.1 The concept of Industry 5.0. Adapted from Borchardt et al. [11].

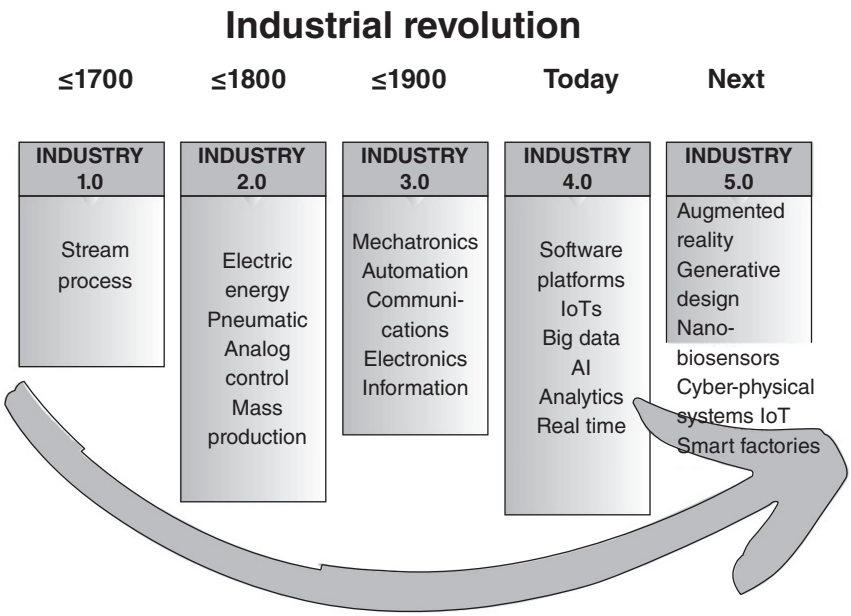

Figure 1.2 The timeline of industrial revolutions. Adapted from Madsen and Berg [12] & Cohen and Singer [13].

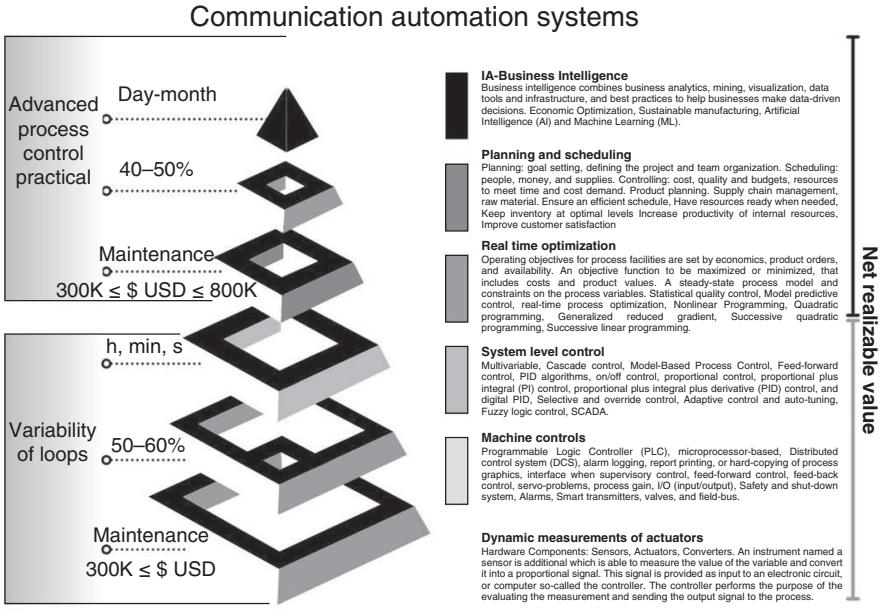

Figure 1.3 The timeline of industrial revolution's pyramid of automation. Adapted from López-Pérez et al. [7], Lucizano et al. [23], Wollschlaeger et al. [24], Martinez et al. [25].

Today automation has taken over process control purposes, which means that operatives are assisted by a distributed control system, which communicates with the instruments in the real process. Process control is a combination of the statistics and engineering areas that deal with the sensors, designs, and algorithms for controlling a process. The aim of process control is to have it behave in a desired value. This includes the processes that are appealing, more accurate, more reliable, or more economical [21, 22] (see Figure 1.2).

In different manufacturing industries, advanced process control systems (APC) have become a nonnegotiable necessity for any manufacturing operation, which allows progress in the automation, understanding, and use of complex systems. APC incorporates a variety of model-based software system technologies, as well as stochastic and metaheuristic systems (see Figure 1.3). Currently, APCs provide supervisory control, bridging the gap between basic controls and overall process improvement, allowing the process to be cost-effective and of sustainable quality and operational safety aligned with Industry 4.0 and 5.0 [26–28].

1.2 Field Elements of Classic and Modern Control Systems

Modern advanced control techniques are model-based in data and look to apply mathematical optimization tools to optimize the performance based on future predictions and conditions. The necessary components and fields for this background are as follows:

- **i.** A dynamic model.
- **ii.** Estimator that converts measured process variables into estimates of unmeasured states and/or parameters.
- **iii.** An algorithm that computes the optimal control action based on model predictions (multi-objective optimization function or Pareto and constraint set).
- **iv.** Methods that restrict the model to be linear.
- **v.** Methods that require a very large amount of data to provide any statistical guarantees.
- **vi.** Methods for uncertainty descriptions are not necessarily accurate or related to physical quantities.
- **vii.** Improving system performance in terms of functionality, security, energy efficiency, environmental impacts, and costs.
- **viii.** Virtual sensors in the industry manages to optimize the operational performance, safety, functionality, and reliability of the bioprocess.
- **ix.** Monitoring, diagnosis, and control could be provided more reliably and robustly using physical sensors.
- **x.** Embedded system is any device that is made up of a programmable computer (microprocessor or microcontroller).
- **xi.** Computer hardware.
- **xii.** Operating system in real time.
- **xiii.** Efficiency and quality in manufacturing.
- **xiv.** Hyper-competitive manufacturing sector.
- **xv.** Internet of Things (IoT).

xvi. Human–machine interface and supervisory control and data acquisition (SCADA).

xvii. Basic regulatory control, advanced regulatory control, multivariable, model-based control, constrained economic optimization, multi-unit constrained economic optimization, first principle economic optimization (RTO), steady-state process model, and economic information (e.g., prices and costs) performance Index to be maximized (e.g., profit) or minimized (e.g., cost) [29, 30].

1.2.1 Advantages

- Reducing operational costs to secure tribal knowledge.
- Easy maintenance as it is not a compact system. In the case of breakdowns, the affected component is easily replaced without completely replacing the entire system.
- They have small size, so they easily adapt to any industrial application without requiring a large workspace.
- It is adaptable. Any necessary module can be integrated.
- They consume minimum energy, which causes the battery life to be extended.
- They give high performance in processing data at high speed and in real time.

1.2.2 Disadvantages

- They are specific operating systems.
- The software of an embedded system presents some restrictions such as small amounts of memory (generally, in the order of kB).
- Limited processing capabilities (generally, the speed of the processors does not exceed the order of MHz).
- Limits the consumption of instant energy whether in execution state or not.
- They may present cybersecurity risks because they feature weak encryption.
- Data shared between two devices can be easily intercepted and decrypted.

For example: data acquisition systems and system monitor parameters such as O_2, CO_2, temperature, flow, humidity, and pH based on the sensors that will collect data and the controllers to correspond to the set points of the variables. The measured data have the potential to practice another improved software tool for the estimation of variables and parameters from process data [31–33].

1.2.3 Why Control and Monitor?

The measurement of variables in processes is a necessary requirement to overcome following concerns:

i. To know internal behaviors.

ii. Fault diagnosis.

iii. Monitoring.

iv. Visualization of critical variables.

v. Processing is subject to disturbances.

vi. Nonlinear systems.
vii. Unstability.
viii. Maintain productivity.
ix. Quality standards.

In addition:

i. Absence of trusted devices.
ii. Time delays.
iii. Errors in the measurement system.
iv. High device costs.
v. Measurement conditions.
vi. Unavailability of sensors.

Nowadays, process and system control theory can be divided into two areas: classical methods and modern control theory methods. The main differences between these groups focus on the representation, design, and operation of dynamic systems or data management in real time, at-line, offline, online, and in-line. Figure 1.4 briefly summarizes the differences between classical and modern control theory methods [7].

General characteristics of classical control methods: Classical control methods are not able to incorporate constraints logically ascending from industrial control problems and have optimization missing complete performance. The most common

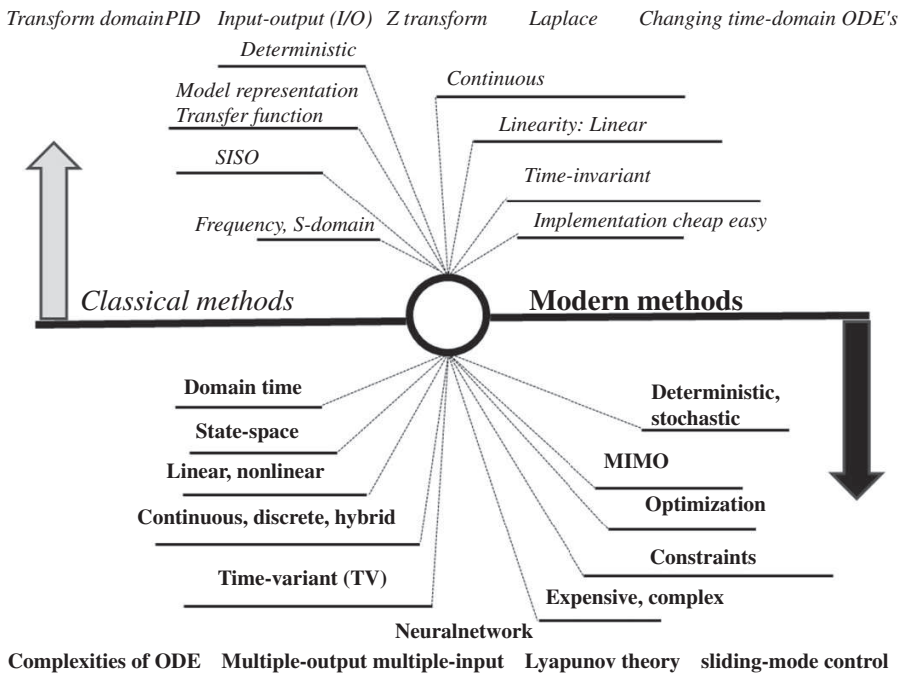

Figure 1.4 Properties' comparison of classical and modern control methods. Adapted from Jha et al. [34], Drgoˇna et al. [35], Holaza et al. [36].

example of classical control methods is the proportional integral derivative (PID) controller, which accounts for more than 95% of the control and automation applications today, mainly thanks to its simple implementation with relative efficiency [37]. Main advantage of state–space representation is the preservation of the time-domain attractiveness, where the response of a dynamical system is a function of many inputs, previous system states, and time, presently

$$y = f(x, u, t). \tag{1.1}$$

Industrial processes often experience nonlinear behaviors that may include output multiplicity, bifurcations, chaos, unstable dynamic response to disturbances, and changes in system parameters; all these phenomena can lead to instability and, ultimately, affect the yield of production. For this reason, the application of traditional linear controllers is limited since they are not able to cope with the high nonlinear behavior of industrial processes. Aside from this, incidental external and internal disturbances in a manufacturing process lead to disappointment and can be a costly and annoying problem for any engineer. Regulatory control is the main strategy for attaining the basic operational requirements in manufacturing processes. The regulatory control has been performed through classical PID feedback controllers by considering the easiness of its practical implementation and satisfactory performance within common industrial practice [38]. Controllers employed range from simple on–off type to proportional (P), integral (I), derivative (D), and PID controls and expert systems. A scheme of typical control is shown in Eq. (1.2).

$$u(t) = \underbrace{K_p e(t)}_{\text{Proportional}} + \underbrace{K_i \int e(t)\,dt}_{\text{Integral}} + \underbrace{K_d \frac{d}{dt} e(t)}_{\text{Derivative}} \tag{1.2}$$

The K_P, K_d, and K_i are the tuning parameters of the controller that can be adjusted by varying the dynamics of the control loop. Feed rates can also be adjusted based on an optimal objective function derived online or offline. The objective function targets to increase productivity or maximize operating profits [39].

In relation, other feedback controllers have been proposed for improving the dynamic performance of the manufacturing processes: among them, the adaptive controllers can modify some parameters in the control's structure to maintain a satisfactory process operation [40]. Controllers designed to combat input disturbances and noisy measurements have been presented in the literature for several years within the following frameworks: sliding-mode theory, observer-based I/O linearizing controllers, and optimal controllers [41]. On the other hand, most of the controllers designed for the aforementioned existing outputs are complicated; therefore, they are difficult to perform in real applications. In fact, how to design a simple and physical controller to perform the control problems of complex chaotic systems is also important both in theory and in application. One of the most important drawbacks of the advanced control designs is their complexity for practical applications and the complete understanding by the plant engineers. Thus, alternative control structures must be simple enough to avoid the abovementioned drawback. Alternatively, most of the controllers designed in the aforementioned existing outputs are

complicated; therefore, they are difficult to perform in real applications. Therefore, the alternative control structures must be simple enough to avoid the abovementioned drawback [42, 43].

Once the information is in the supervision and monitoring equipment, not only is interaction with the user achieved through graphics, alarm signals, report generation, and global analysis of the plant, but it can also directly influence the dynamic behavior of the system variables through programming of observers, filters, virtual sensors, among other components that are incorporated into the plant [44]. Supervision achieves the reconfiguration of system parameters through the controller, executing a set of actions to bring the process to its normal operation using model-based self-tuning methods or without mathematical knowledge of the process, among others. This constitutes a crucial difference in relation to monitoring that only covers detection and diagnosis and sometimes those systems that undertake only surveillance tasks and have been mistakenly called supervision systems. Finally, the last *n* levels of the pyramid, manufacturing execution system planning, and enterprise resource planning are manufacturing execution systems that organize the resources necessary to execute the plant's production plan that covers raw materials, order of priorities, change of production instructions, the controllers, and measurement interval of the sensors, among others. Therefore, there is a strong financial inspiration to develop the finest control scheme that would facilitate rapid startup and stabilization of manufacturing processes subject to redundant disturbances. In the control literature, regardless of the considerable progress in APC proposals such as sliding-mode control, model predictive control, and internal model control, PID controllers are still widely employed in industrial control systems because of their structural simplicity, reputation, robust behavior, and easy implementation (see, Figure 1.5). Along with the system's stability, it also satisfies chief performance such as smooth reference tracking, efficient disturbance rejection, and measurement of noise attenuation criteria [45–47].

For a process which is operating satisfactorily, the variation of product quality falls within acceptable limits. These limits normally correspond to the minimum and maximum values of a specified property. Normal operating data can be used to compute the mean deviation and the standard deviations of a given process variable from a series of observations. The standard deviation is a measure for how the values of the variable spread around the mean. A large value indicates wide variations in the variable. Assuming the process variable follows a normal probability distribution, then 99.7% of all observations is to lie within an upper limit and a lower limit. This can be used to determine the quality of the control. If all data from a process lie within the limits, then it can be concluded that nothing unusual has happened during the recorded period, the process environment is relatively unchanged, and the product quality lies within specifications. On the other hand, if repeated violations of the limits occur, then the conclusion can be drawn that the process is out of control and that the process environment has changed. Once this has been determined, the process operator can take action to adjust operating conditions to counteract undesired changes that have occurred in the process conditions [48–51].

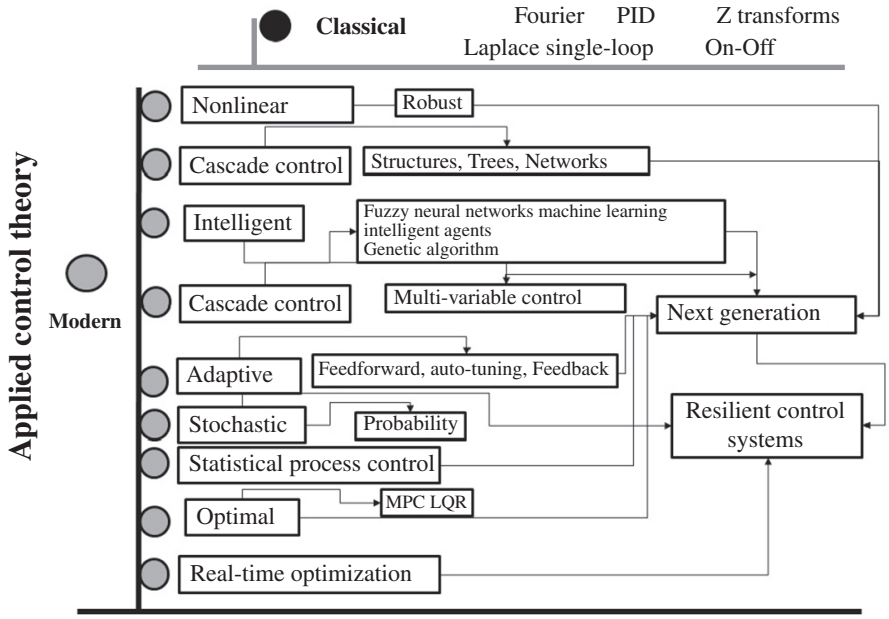

Figure 1.5 Classical and modern control theory.

1.3 Process Modeling in Control Systems Design

Mathematical model of a dynamic System is a set of equations that represent with a certain degree of accuracy the dynamics of the physical system. The model is generally described as an operator between the inputs and outputs of the system, or as a set of differentials (continuous case) and/or difference (discrete case) equations. Generally, when working with dynamic systems that are modeled by a finite number of coupled first-order ordinary differential equations, the state variables represent the "memory" that the dynamic system has of its past. Vector notation is usually used to write these equations compactly; the n first-order differential equations can be defined and rewritten as a vector differential equation of dimension n [52–55]. A thorough understanding of the time-dependent behavior of the technological processes is required to instrument and control the process. This, in turn, requires an appreciation of how mathematical tools can be employed in the analysis and design of process control systems. There are several mathematical principles that are utilized for automatic control. These are as follows:

- Physical, chemical, biological models, and empirical models.
- Simulation of dynamic models.
- Laplace transforms.
- Fitting dynamic models to experimental data [56–58].

Dynamic system: it is the one that generates data that change with the passage of time, that is, they possess certain dynamics. Dynamic systems are systems whose

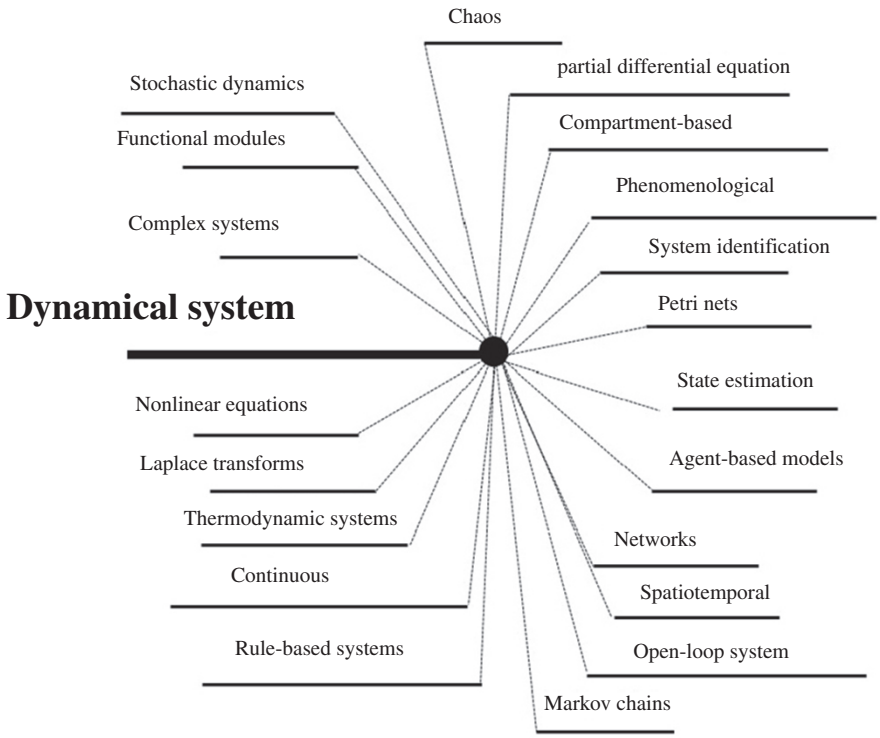

Figure 1.6 Dynamic system modules. Adapted from López-Pérez et al. [59, 60], Jongeneel and Moulay [61].

internal variables (state variables) follow a series of temporal rules (see Figure 1.6). They are called systems because they are described by a set of equations that are time-dependent, either implicitly or explicitly. Dynamic system classes:

- Isolated: they do not interact with their environment.
- Not insulated: interact with your environment.
- Natural: unaffected by human intervention.
- Artificial: created by man.
- Physics: they involve matter and energy.
- Not physical: thoughts.

Black Box: Black box is a system in terms of inputs and outputs. These models allow a global characterization of the model by disturbing its input to observe the variation or effect of said input on the states and parameters of the system at the output, that is to say: an identification, parametric sensitivity, and confidence interval of operation and parameters. Examples of some tools for the development of black-box models are as follows:

- Support vector machines.
- Partial least squares.
- Artificial neural networks.
- Fuzzy inference system.

White Boxes:
Detailed mechanistic models (white box) require:

- Large sets of well-identified parameters.
- Large uncertainty in parameters.
- Internals being completely exposed to the user.
- Complexity of the implementation.

Gray Boxes:

- These models do not study the source code and can lean on the results obtained by testing the user interface (possibly gathered purely by experimentation).
- Knowing the internal structure of the program can create more varied and smarter scenario.
- Tests cannot hope to provide a complete coverage of the program.
- View allows only partial exposure of the system behavior [14].

1.4 Ordinary Differential Equations and Laplace

A general form of an ordinary differential equation (ODE) containing one independent and one dependent variable is $f(x, y, y', y'', \ldots, y^n) = 0$, where is an arbitrary function of $x, y, y', y'', \ldots, y^n$, here x is the independent variable, while y is the dependent variable, and $y^n = \frac{d''y}{d''x}$. The order of an ODE is the order n of the highest derivative appearing in it.

A common class of mathematical models for dynamical systems is *ODEs* written as.

$$\frac{dx}{dt} = f(x) \tag{1.3}$$

Here $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ is a vector of real numbers that describes the current state of the system, and Eq. (1.3) describes the rate of change of the state as a function of the state itself. Note that we do not bother to write the vector x any differently than a scalar variable. It will generally be clear from the context whether a variable is a vector or scalar quantity.

The state of a system is the minimal set of numbers $\{x_i(t), \forall\, i = 1, 2, 3, \ldots, n\}$ needed together with the input $u(t)$ with t in the interval $[t_0, t_i)$ to uniquely determine the behavior of the system in the interval. The number n is known as the order of the system. As t increases, the state of the system evolves and each of the numbers $x_i(t)$ becomes a time variable. These variables are known as state variables. In *vector* notation, the set of state variables form the state vector.

The differential equation (1.3) is called an autonomous system because there are no external influences. In many examples, it is useful to model the effects of external disturbances or controlled forces on the system. One way to capture this is to replace Eq. (1.3) by

$$\frac{dx}{dt} = f(x, u), \tag{1.4}$$

where u represents the effect of external influences. The model (1.4) is called a forced or controlled differential equation. The model implies that the rate of change of the state can be influenced by the input $u(t)$.

A description of systems that are linear and time-invariant (LTI) is observed in Eqs. (1.5) and (1.6), that is, systems described by linear differential equations with constant coefficients [8]. If a system is LTI and described by n state variables, r input variables, and m output variables, Eq. (1.4), then the state equation will have the form:

$$\begin{aligned} x_1 &= a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + b_{11}u_1 + b_{12}u_2 + \ldots + b_{1r}u_r \\ x_2 &= a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n + b_{21}u_1 + b_{22}u_2 + \ldots + b_{2r}u_r \\ &\vdots \\ x_m &= a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n + b_{n1}u_1 + b_{n2}u_2 + \ldots + b_{nr}u_r \end{aligned} \tag{1.5}$$

An important property of the linear state equation description is that all system variables may be represented by a linear combination of the state variables and the system inputs, and the output equation will have the form:

$$\begin{aligned} y_1 &= c_{11}x_1 + c_{12}x_2 + \ldots + c_{1n}x_n + d_{11}u_1 + d_{12}u_2 + \ldots + d_{1r}u_r, \\ y_2 &= c_{21}x_1 + c_{22}x_2 + \ldots + c_{2n}x_n + d_{21}u_1 + d_{22}u_2 + \ldots + d_{2r}u_r, \\ &\vdots \\ y_m &= c_{m1}x_1 + c_{m2}x_2 + \ldots + c_{mn}x_n + d_{m1}u_1 + d_{m2}u_2 + \ldots + d_{mr}u_r, \end{aligned} \tag{1.6}$$

where the coefficients a_{ij}, b_{ij}, c_{ij}, and d_{ij} are constants. If we use vector–matrix expressions, these equations can be written as follows:

State equation

$$x = Ax + Bu \tag{1.7}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix}, B = \begin{bmatrix} b_{11} & b_{12} & \ldots & b_{1r} \\ b_{21} & b_{22} & \ldots & b_{2r} \\ \vdots & \vdots & \vdots & \vdots \\ b_{n1} & b_{n2} & \ldots & b_{nr} \end{bmatrix}, u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_r \end{bmatrix}. \tag{1.8}$$

Dynamical relation, output equation

$$y = Cx + Du \tag{1.9}$$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, C = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix}, D = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1r} \\ d_{21} & d_{22} & \dots & d_{2r} \\ \vdots & \vdots & \vdots & \vdots \\ d_{m1} & d_{m2} & \dots & d_{mr} \end{bmatrix}. \tag{1.10}$$

Remarks

- **i.** Matrices A, B, C, and D are called the state matrix, input matrix, output matrix, and direct transmission matrix, respectively.
- **ii.** Vectors x, u, and y are the state vector, input vector, and output vector, respectively.
- **iii.** The elements of the state vector are the state variables.

When the dynamical system does not have any control input, i.e., it runs autonomously, then we refer to it as an autonomous system. Mathematically, this is written as follows:

$$\dot{x} = f(x,\, t)\,. \tag{1.11}$$

An important concept in dynamical systems is equilibrium. Roughly, the equilibrium of a dynamical system is the point in the state–space where the states remain constant, i.e., steady state. It is defined mathematically as follows (equilibrium):

$$f(x^{eq},\, t) = 0, \;\; x^{eq}: \text{ State equilibrium (continuous systems at steady state)} \tag{1.12}$$

Remarks

- **i.** Systematic analysis and synthesis of higher-order systems without truncation of system dynamics.
- **ii.** Convenient tool for Multiple Input, Multiple Output (MIMO) systems.
- **iii.** Uniform platform for representing time-invariant systems, time-varying systems, linear systems, as well as nonlinear systems.
- **iv.** Can describe the dynamics in almost all systems (mechanical systems, electrical systems, biological systems, economic systems, social systems, etc.)

Laplace Transforms

The Laplace transform is a linear operator that switched a function $f(t)$ to $F(s)$.

- The technique of Laplace transform (and its inverse) facilitates the solution of ODE.
- Transformation is from the time domain to the frequency domain.
- Functions are complex, often described in terms of magnitude and phase. Laplace Transform for ODEs:
  - Equation with initial conditions.
  - Laplace transform is linear.

- Apply derivative formula.
- Rearrange.
- Take the inverse.

Case Study Research 1

The most important mathematical tools used in this chapter are shown in the following text. We study dynamical systems modeled by a finite number of ODE coupled together in the following compact form:

$$\begin{aligned} \dot{x} &= f(x) + g(x)\,u + \xi_p(t), \\ y &= h(x), \end{aligned} \tag{1.13}$$

where $x = x(t)$, $y = y(t)$, $x(t_0) = x_0 \forall t \geq t_0$.

$$f : \mathbb{R}^n \to \mathbb{R}^n,\ g : \mathbb{R}^n \to \mathbb{R}^{n\times m},\ h : \mathbb{R}^n \to \mathbb{R}^p,$$

and $x \in \mathcal{D} \subset \mathbb{R}^n$ is the state vector, $u \in \mathcal{U} \subset \mathbb{R}^m$ is the vector of control inputs, and $\xi_p(t) \subset \mathbb{R}^n$ is the term of external disturbances to the system. Also, it is considered that y is the output of the system, e.g., physically measurable variables, and stabilization of the output at a desired value is sought.

Case Study Research 2

The following is an example of using the Laplace transform for a convergence test.

In this study, the following equations represent the core of such modeling:

$$\begin{aligned} \frac{dx_1}{dt} &= r_X(x_1)\,x_1 = k_1\left(1 - \left(\frac{x_1}{k_2}\right)^{k_3}\right)x_1, \\ \frac{dx_2}{dt} &= r_{cl}(x_2)\,\frac{1}{Y_1}x_1 = \frac{1}{Y_1}k_4\left(1 - \left(\frac{x_2}{k_5}\right)^{k_8}\right)x_1, \\ \frac{dx_3}{dt} &= r_S(x_3)\cdot\frac{1}{Y_2}x_1 = \frac{1}{Y_2}k_6\left(1 - \left(\frac{x_3}{k_7}\right)^{k_9}\right)x_1. \end{aligned} \tag{1.14}$$

These equations are fundamental in capturing the dynamics of microalgae growth in photobioreactors, as they describe the rate of change of biomass, chlorophyll, and substrate concentrations over time. The parameters within these equations, such as ψ, δ, β, λ, η, α, and the γ exponents, are crucial in defining the specific growth conditions and responses of the microalgae. In the sophisticated domain of microalgae growth modeling, particularly in state–space nonlinear systems, the concept of state variables becomes integral. These variables, denoted as C_X, C_{Cl}, and C_S, represent the concentrations of biomass, chlorophyll, and substrate, respectively. To streamline the mathematical representation, these variables are aggregated into a state vector, x_t, defined as follows:

$$x_t = \begin{bmatrix} x_{t,1} \\ x_{t,2} \\ x_{t,3} \end{bmatrix} = \begin{bmatrix} C_X \\ C_{Cl} \\ C_S \end{bmatrix}, \tag{1.15}$$

where $x_{t,1}$, $x_{t,2}$, and $x_{t,3}$ correspond to the concentrations of biomass, chlorophyll, and substrate at any given time t. This vector notation allows for a more compact and

efficient description of the system dynamics. The evolution of these state variables over time is governed by a set of differential equations, collectively forming a state–space nonlinear system. This system can be succinctly expressed as follows:

$$\dot{x}_t = f(x_t, u_t), \tag{1.16}$$

$$y = g(x_t, u_t). \tag{1.17}$$

Here, $\dot{x}_t$ represents the derivative of the state vector with respect to time, indicating the rate of change of the state variables. The function $f(\cdot) : \mathbb{R}^{n+q} \to \mathbb{R}^n$ is a nonlinear, smooth vector function that is Lipschitz continuous in x_t and uniformly bounded in u_t, the control input vector. The term Δf signifies the additive modeling error inherent in the system. The output vector $y_t \in \mathbb{R}^m$ comprises the measured states, providing a link between the model and empirical observations. This framework of state–space modeling is pivotal in capturing the complex dynamics of microalgae growth, enabling the prediction and control of the system under varying conditions. It is determined that the system is observable since the rank of O is 3.

Case Study Research 3

In the proposed observer design, the system represented by Eq. (1.17) and system (1.16) is expressed as follows:

$$\dot{x}_t = f(x_t),\ x_{t=0} = 0,\ y = g(x_t) = Cx_t. \tag{1.18}$$

An auxiliary system, serving as an observer for system (1.14), is defined as follows:

$$\begin{aligned} \frac{d}{dt}\hat{x}_t &= f(\hat{x}) - l_1\left(e_t + \int e_t dt + \int\left(\int e_t dt\right)dt + l_2\right), \\ \hat{y}_t &= h(x_t) = Cx_t. \end{aligned} \tag{1.19}$$

The following conditions are established:

- The initial condition $\hat{x}_{t=0} = x_{t=0} = 0$ implies $\hat{x}_t = x_t$. 2. The norm $\|f(\hat{x}_t) - f(x_t)\|$ is bounded by L.
- The term $l_1\left(e_t + \int e_t dt + \int\left(\int e_t dt\right)dt + l_2\right)$ in (1.19) aims to achieve asymptotic estimation error. By selecting l_1 and l_2 adequately
- $l_1 l_2 \approx L < \infty$, with $l_1 > 0$ and $l_2 > 0$.
- Considering the error $e_t = x_t - \hat{x}_t$ (see Figure 1.7).
- $\|e_t\| = \|x_t\| - \|\hat{x}_t\| \to 0$ as $t \to \infty$.

Therefore:

$$\frac{d}{dt}\|e_t\| \le -l_1\left(\|e_t\| + \int \|e_t\| dt + \int\left(\int \|e_t\| dt\right)dt\right) \tag{1.20}$$

Applying the Laplace transformation to inequality (1.20) results in:

$$\begin{aligned} &s\|x_s\| - \|x_s(0)\| - \{s\|\hat{x}_s\| - \|\hat{x}_s(0)\|\} \\ &\le -l_1\left(\|x_s\| - \|\hat{x}_s\| + s^{-1}\left(\|x_s\| - \|\hat{x}_s\|\right) + s^{-2}\left(\|x_s\| - \|\hat{x}_s\|\right)\right). \end{aligned} \tag{1.21}$$

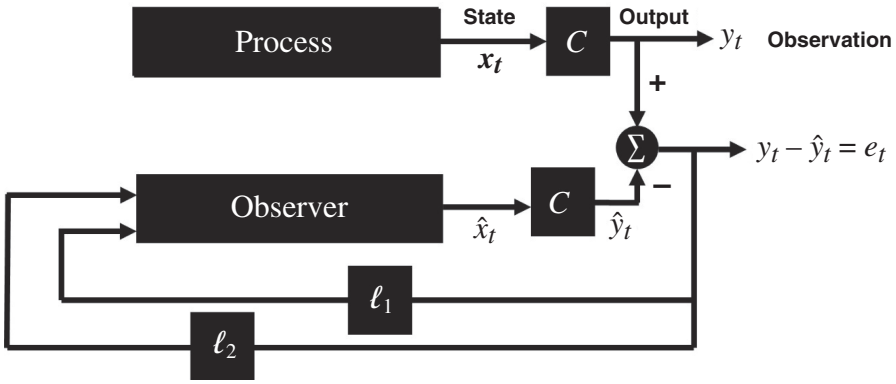

Figure 1.7 Block diagram for the proposed observer.

Assuming homogeneous initial conditions for inequality (1.21), i.e., $x_s(0) = 0$ and $\hat{x}_s(0) = 0$, and rearranging algebraically, we obtain:

$$||x_s|| \left(s + l_1 + l_1 s^{-1} + l_1 s^{-2}\right) \leq ||\hat{x}_s|| \left(s + l_1 + l_1 s^{-1} + l_1 s^{-2}\right) \tag{1.22}$$

or equivalently:

$$\frac{||x_s||}{||\hat{x}_s||} \leq \frac{\left(s + l_1 + l_1 s^{-1} + l_1 s^{-2}\right)}{\left(s + l_1 + l_1 s^{-1} + l_1 s^{-2}\right)}. \tag{1.23}$$

Applying the final value theorem to inequality (1.23) for asymptotic behavior:

$$\lim_{s \to 0} \frac{||x_s||}{||\hat{x}_s||} \leq \lim_{s \to 0} \frac{\left(s + l_1 + l_1 s^{-1} + l_1 s^{-2}\right)}{\left(s + l_1 + l_1 s^{-1} + l_1 s^{-2}\right)}, \tag{1.24}$$

which implies:

$$\lim_{s \to 0} \frac{\left(s + l_1 + l_1 s^{-1} + l_1 s^{-2}\right)}{\left(s + l_1 + l_1 s^{-1} + l_1 s^{-2}\right)} = \frac{l_1}{l_1} = 1 \tag{1.25}$$

and

$$\lim_{s \to 0} \frac{||x_s||}{||\hat{x}_s||} \leq 1 \tag{1.26}$$

or in the time domain:

$$\lim_{t \to \infty} \frac{||x_t||}{||\hat{x}_t||} \leq 1. \tag{1.27}$$

1.5 Linear Systems

Linear systems: A system is called linear if the superposition principle is applied. This principle states that the response produced by the simultaneous application of two functions of different inputs is the sum of the two individual responses. Thus, for

the linear system, the response to several inputs is calculated by treating one input at a time and adding the results. To map a model to frequency space,

- system must be linear – output proportional to input.
- Given system P – input signals: x_1 and x_2 – output signals (response): y_1 and y_2.

Theorem 1.1 Existence and uniqueness

Let f be continuous in sections in t satisfying the Lipschitz condition

$$\|f(x) - f(y)\| \leq L\|x - y\|, \forall x, y \in B = \{x \in \mathbb{R}^n | \|x - y\| \leq r\} \forall\, t \geq t_0.$$

Then there exists a $\delta > 0$ such that

$$\dot{x} = f(x),\ x(t_0) = x_0$$

has only one solution [62].

Definition 1.1 Lie or directional derivative [63]

Let $V : D \to R, f : D \to R^n$. The Lie derivative of V with respect to f along f, is given by $L_f V$ and is defined by

$$L_f V(x) = \frac{\partial V}{\partial x} f(x).$$

This is the familiar notion of the derivative of V along the trajectory of the autonomous system of the form $\dot{x} = f(x)$.

Definition 1.2 Equilibrium point [62]

A point $x = x^*$ in state–space is an equilibrium point of the system $\dot{x} = f(x) + g(x)u$, with $u = u^*$, if it has the property that the initial condition of the state x^*, remains at x^* for all time t, i.e., x^* is the equilibrium point of the system $\dot{x} = f(x^*) + g(x^*)\,u^*$.

Lemma 1.1 Gronwall–Bellman inequality [62]

Let $\lambda : [a, b] \to \mathbb{R}$ be continuous nonnegative. If a function $y : [a, b] \to \mathbb{R}$ satisfies

$$y(t) \leq \lambda(t) + \int_a^t \mu(s) y(s) ds,$$

with $a \leq t \leq b$ within the same interval

$$y(t) \leq \lambda(t) + \int_a^t \lambda(s)\mu(s) e^{\int_s^t \mu(\tau) d\tau} ds,$$

in particular, if $\lambda(t) = \lambda$ is a constant, then

$$y(t) \leq \lambda e^{\int_a^t \mu(\tau) d\tau},$$

if, in addition $\mu(t) = \mu \geq 0$ is a constant, then

$$y(t) \leq \lambda e^{\mu(t-a)}.$$

Lemma 1.2 Lemma of Barbalat [62]

If $f(t)$ is uniformly continuous, such that $lim_{t\to\infty} \int_0^t \|f(\tau)\| d\tau$ exists and is finite. So

$$f(t) \to 0 \text{ when } t \to \infty.$$

Corollary: If $G, \dot{G} \in L_\infty$, in addition $G(t) \in L_{\mathbf{p}}$, for $p = [1, \infty)$, so $\lim_{t\to\infty} G(t) = 0$.

Lemma 1.3 Lemma of comparison [62]

Consider the scalar differential equation

$$\dot{u} = L(u),\ u = u(t),\ u(t_0) = u_0,$$

where $L(u)$ is continuous at t and locally Lipschitz at u, for all $t \geq 0$ and all $u \in J \subset R$. Let $[t_0, T)$ (T can be infinite) the maximum interval of the existence of the solution u, and assume $u \in J$ for all $t \in [t_0, T)$. Let $v = v(t)$ a continuously differentiable function whose derivative on the right-hand side D^+v satisfies the differential inequality

$$D^+v \leq L(v, t),\ v(t_0) \leq u_0,$$

with $v \in J$ for all $t \in [t_0, T)$. So $v < u$ for all $t \in [t_0, T)$.

1.6 Nonlinear Dynamical Systems

Nonlinear systems: A system is nonlinear if the superposition principle is not applied. Therefore, for a nonlinear system, the response to two inputs cannot be calculated by treating each one at a time and summing the results, represented by the system of nonlinear differential equations

$$\dot{x}(t) = f(x(t),\ u(t)), x(t_0) = x_0. \tag{1.28}$$

Consider an input–output system whose state trajectory, $x : \mathbb{R} \to \mathbb{R}^n$, system output $y : \mathbb{R} \to \mathbb{R}^m$, and control input $u : \mathbb{R} \to \mathbb{R}^p$ satisfy the following set of equations,

$$\begin{aligned} \dot{x}(t) &= f(x,\ u), \\ y &= h(x,\ u), \\ u &= k(x). \end{aligned} \tag{1.29}$$

We assume that the functions $f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$ and $h : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^m$ are known.

Given a desired output signal, $y^* : \mathbb{R} \to \mathbb{R}^m$, the objective is to find the state feedback controller $k : \mathbb{R}^n \to \mathbb{R}^p$ such that the output signal $y(t) \to y^*(t)$ as $t \to \infty$.

Case Study Research 4
Consider the following nonlinear dynamic system with control affine input and linear measurements:

$$\Sigma = \begin{cases} \dot{x} = f(x) + g(x)\,u \\ \qquad y = Cx \end{cases}. \tag{1.30}$$

Here $x \in \mathbb{R}^n$ represents the state vector, using values in X as a connected manifold of dimension n, and $u \in \mathbb{R}^q$ denotes the vector of known external inputs, taking values in some open subset $\mathbb{U}$. Furthermore, $\in \mathbb{R}^m$ denotes the vector of measured outputs, taking values in some open subset $\mathbb{Y}$. Function f will generally be assumed to be C^∞ of their arguments and input functions $u(^\circ)$ assumed to be locally essentially bounded and measurable functions in a set $\mathbb{U}$.

Now let us assume the following.

A1. $f(0) = 0$ and f is globally Lipschitz bounded: $\|f(x) - f(x_{sp})\| \leq \mathcal{L}\|e\|; \forall x \in \mathbb{R}^n$, $e(t) = x(t) - x_{sp}(t)$ is the named regulation error, where x_{sp} is the required set point and $\dot{e}(t) = \dot{x}(t) - \dot{x}_{sp}(t)$ and

$$\mathcal{L} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}. \tag{1.31}$$

A2. The vector field $g(x)$ is bounded: for $\forall x \in \mathbb{R}^n$, $\|g(x) - g(x_{sp})\| \leq \wp < \infty$. Both assumptions **A1** and **A2** can be physically performed for whatever process.

The following control input u can regulate the nonlinear system:

For example,

$$u = (k_1(e) - \iota_1)\,e + k_2 sign(e)^{\frac{1}{n}}. \tag{1.32}$$

1.7 Stability Theory

Definition 1.3 Stability [62]

The equilibrium point $x^* = 0$ of the system (1.13) is stable, if for every $\varepsilon > 0$, there exists a $\delta = \delta(\varepsilon, t_0) > 0$, such that

$$\|x(t_0)\| < \delta \Rightarrow \|x(t)\| < \varepsilon, \quad \forall\, t \geq t_0 \geq 0.$$

Definition 1.4 Uniformly stable [62]

The equilibrium point $x^* = 0$ of the system (1.13) is uniformly stable, if for every $\varepsilon > 0$, $\exists \delta = \delta(\varepsilon) > 0$, independent of t_0 such that

$$\|x(t_0)\| < \delta \Rightarrow \|x(t)\| < \varepsilon, \quad \forall\, t \geq t_0 \geq 0.$$

Definition 1.5 Asymptotically stable [62]

The equilibrium point $x^* = 0$ is asymptotically stable, if it is stable; moreover, if it exists $c = c(t_0) > 0$ such that $x(t) \to 0$ in $t \to \infty$, for all $||x(t_0)|| < c$.

Definition 1.6 Exponentially stable [62]

The equilibrium point x^* is exponentially stable, if there exists a $\alpha > 0, \beta \in R$ such that

$$x_0 \in B_r \Rightarrow ||x(t)|| < \beta e^{-\alpha(t-t_0)} x_0.$$

Definition 1.7 Stability Uniformly Lately Bounded, Stability UUA [62]

The solution of the system (1.13) is said to be uniformly ultimately bounded stable, if it is uniformly stable and there exists a $c > 0$ with ultimate boundary $b > 0$ independent of t_0, i.e., for $a \in (0, c)$ exists $T = T(a, b) > 0$, such that

$$||x(t_0)|| \leq a \quad \Rightarrow ||x(t)|| \leq b, \quad \forall\, t \geq t_0 + T.$$

Theorem 1.2 Lyapunov's Theorem [62]

Let x^* be an equilibrium point of the autonomous system, $D \in \mathbb{R}^n$ the domain containing the origin. $x = 0$. Let $V : D \to \mathbb{R}$ a continuously differentiable function such that

$$V(0) = 0, V(x) > 0, \text{ on } D - \{0\}.$$

$$\dot{V}(x) \leq 0 \quad en \quad D.$$

So, x^* is stable. In addition, if

$$\dot{V}(x) < 0 \quad en \quad D - \{0\},$$

then x^* is asymptotically stable.[1]

Definition 1.8 Attractive set [62]

Let x^* be an equilibrium point of the system (1.13) and $V(x)$ be an energy function. The attractive set R_A is defined by

$$R_A = \left\{x \in D | \dot{V}(x) < c_1\right\}. \tag{1.33}$$

Theorem 1.3 [64] Let $V : \mathbb{R}^n \to \mathbb{R}$ be a real function such that

$$\frac{d}{dt} V(x) \leq -\alpha V(x) + \beta \tag{1.34}$$

is satisfied, with α, β positive scalars. Then, $V(x)$ is an attractive set; moreover, the property

$$\limsup_{t \to \infty} V(x) \leq \frac{\beta}{\alpha}, \tag{1.35}$$

is fulfilled.

---

1 In Lie derivative notation, $\frac{dV}{dt} = \frac{\partial V}{\partial x}\frac{dx}{dt} = \frac{\partial V}{\partial x}\dot{x} = \frac{\partial V}{\partial x}\{f(x) + g(x)\,u\} = L_f V(x) + L_g V(x)$.

Theorem 1.4 Taylor's theorem [65]

Let $r \geq 1$ be a positive integer and the function $f : \mathbb{R} \to \mathbb{R}$ differentiable r times at the point $a \in \mathbb{R}$. Then there is a function $h_r : \mathbb{R} \to \mathbb{R}$ such that

$$f(x) = f(a) + \dot{f}(a)(x-a) + \frac{\ddot{f}(a)}{2!}(x-a)^2 + \cdots + \frac{f^{(r)}(a)}{r}(x-a)^r + h_r(x)(x-a)^r, \tag{1.36}$$

where $\lim_{x\to a} h_r(x) = 0$. These are the so-called higher order terms. For the case $x \in \mathbb{R}^n$, the derivatives of the function f are obtained by means of the Jacobian matrix defined by $\boldsymbol{J}$

$$\dot{f}(a) = \frac{\partial f_i(x)}{\partial x_j}|_{x=a} = \boldsymbol{J}(a)_{ij}.$$

1.8 Systems' Identification

Identification is the experimental determination of the temporal behavior of a process or system using measured signals and determining the temporal behavior within a class of models [66]. The identification algorithm used in the thermal plant, in the coupled tanks, and the tomato dehydrator plant is presented in the following text. It should be noted that this identification provides an approximate model used for the design of different controls.

1.8.1 Recursive Least Squares Method (Applied to Chapter 8)

The identification algorithm by the least squares method is based on the minimization of a quadratic form. For processes, in general, of (1.13), we note that

$$\begin{aligned}
\dot{x}_1 &= f_1(x) + g_{11}(x)u_1 + g_{12}(x)u_2 \cdots g_{1m}(x)u_m, \\
\dot{x}_2 &= f_2(x) + g_{21}(x)u_1 + g_{22}(x)u_2 \cdots g_{2m}(x)u_m, \\
&\vdots \qquad\qquad\qquad \vdots \\
\dot{x}_n &= f_n(x) + g_{n1}(x)u_1 + g_{n2}(x)u_2 \cdots g_{nm}(x)u_m.
\end{aligned} \tag{1.37}$$

A requirement for applying the recursive least squares method is that the linear or nonlinear functions $f_1(x) \cdots f_n(x), g_{11}(x) \cdots g_{nm}(x)$ contain constant parameters and also the system (1.37) is linear with respect to its parameters. Then the system can be parameterized as follows:

$$\dot{x} = \varphi^\intercal \theta, \varphi \in \mathbb{R}^{k\times n}, \theta \in \mathbb{R}^k,$$

where $\varphi = \varphi(x_1, x_2, \ldots x_n, u_1, u_2 \ldots u_m)$ contains the dynamics, so there are k unknown parameters θ.

The least squares method leads to a minimization in the quadratic error as a performance index. See, for example [67], where the estimation error is defined by

$$e = Z(\tau) - \varphi^\intercal(\tau)\hat{\theta}(t), \quad Z = \dot{x}. \tag{1.38}$$

The performance index to be minimized is defined by

$$J(\theta) = \frac{1}{2}\int_{t_0}^{t} \left|Z(\tau) - \varphi^{\intercal}(\tau)\hat{\theta}(t)\right|^2 d\tau.$$

The performance index penalizes all past errors by $\tau = 0$ at t due to $\theta(t) = \hat{\theta}(t)$. Since $J(\theta)$ is a convex function on $\mathbb{R}$ at each time instant t, its minimum satisfies

$$\nabla J(\theta) = \int_{t_0}^{t} Z(\tau)\varphi(\tau)\, d\tau + \hat{\theta}(t)\int_{t_0}^{t} \varphi^2(\tau)\, d\tau = 0.$$

Obtaining the estimate of the form

$$\hat{\theta}(t) = \left(\int_{t_0}^{t} \varphi^2(\tau)\, d\tau\right)^{-1} \int_{t_0}^{t} Z(\tau)\varphi(\tau)\, d\tau,$$

or in vector form, i.e., when more than one parameter needs to be estimated, is described by

$$\hat{\theta} = \left[\int_{t_0}^{t} \varphi(x,u)\varphi^{\intercal}(x,u)d\tau\right]^{-1} \int_{t_0}^{t} \varphi(x,u)Zd\tau = \mathbf{P}\int_{t_0}^{t} \varphi(x,u)Zd\tau.$$

Defining $\mathbf{P}^{-1}$ as

$$\mathbf{P}^{-1} = \int_{t_0}^{t} \varphi(x,u)\varphi^{\intercal}(x,u)\, d\tau, \quad \mathbf{P} \in \mathbb{R}^{k\times k},$$

where $\mathbf{P}$ may not exist. Therefore, it is established that $\mathbf{P}\mathbf{P}^{-1} = \mathbf{I}_k$ is satisfied $\frac{d}{dt}\left(\mathbf{P}\mathbf{P}^{-1}\right) = 0_{k\times k}$. Then it is evident that the following property is satisfied

$$\dot{\mathbf{P}}^{-1} = \varphi(x,u)\varphi^{\intercal}(x,u),$$

$$\frac{d}{dt}\left(\mathbf{P}\mathbf{P}^{-1}\right) = \dot{\mathbf{P}}\mathbf{P}^{-1} + \mathbf{P}\dot{\mathbf{P}}^{-1},$$

where

$$\dot{\mathbf{P}} = -\mathbf{P}\varphi(x,u)\varphi^{\intercal}(x,u)\mathbf{P}.$$

Accordingly,

$$\dot{\hat{\theta}} = \mathbf{P}\frac{d}{dt}\int_{t_0}^{t} \varphi(x,u)Zd\tau + \frac{d}{dt}\mathbf{P}\int_{t_0}^{t} \varphi(x,u)Zd\tau,$$

$$\dot{\hat{\theta}} = \mathbf{P}\varphi(x,u)Z + \dot{\mathbf{P}}\int_{t_0}^{t} \varphi(x,u)Zd\tau,$$

$$\dot{\hat{\theta}} = \mathbf{P}\varphi(x,u)Z - \left(\mathbf{P}\varphi(x,u)\varphi^{\intercal}(x,u)\mathbf{P}\right)\int_{t_0}^{t} \varphi(x,u)Zd\tau.$$

Grouping terms by left-hand side and substituting $\hat{\theta}$

$$\dot{\hat{\theta}} = \mathbf{P}\varphi(x,u)\left(Z - \varphi^{\intercal}(x,u)\,\mathbf{P}\int_{t_0}^{t}\varphi(x,u)\,Z d\tau\right),$$

$$\dot{\hat{\theta}} = \mathbf{P}\varphi(x,u)\left(Z - \varphi^{\intercal}(x,u)\,\hat{\theta}\right).$$

Note that the estimation error term appears (1.38), so that

$$\dot{\hat{\theta}} = \mathbf{P}\varphi(x,u)\,e. \tag{1.39}$$

1.8.2 Parameter Identification

The foundations of the least squares algorithm (LSA) were laid in 1974 by the German mathematician Johann Friedrich Gauss (1777–1855). This problem consists of finding the parameters of the mathematical model that represents the behavior of a physical plant through measurable or estimated input and output values.

1.8.3 Ordinary Least Squares

In [68], a detailed explanation of the LSA is presented, which is taken up in a simplified form, in the following text. Consider that the model

$$Ay(t) = Bu(t-1) + Cv(t), \tag{1.40}$$

which represents a plant with input $u(t)$ and output $y(t)$ and which is subject to known disturbances $v(t)$. The coefficients of Eq. (1.40) are defined, by using the backward shift interpretation z^{-1}, as follows:

$$\begin{aligned} A &= 1 + a_1 z^{-1} + \cdots + a_{n_a} z^{-n_a}, \\ B &= b_0 + b_1 z^{-1} + \cdots + b_{n_b} z^{-n_b}, \\ C &= c_0 + c_1 z^{-1} + \cdots + c_{n_c} z^{-n_c}. \end{aligned} \tag{1.41}$$

In order to estimate these parameters, it is convenient that the system Eq. (1.40) is presented with the available information and that which is to be estimated in a factored form as follows:

$$y(t) = \chi^{T}(t)\theta + e(t), \tag{1.42}$$

where θ is the vector with unknown parameters, defined as

$$\theta^{T} = \left[-a_1, \ldots, -a_{n_a}, b_0, \ldots, b_{n_b}, c_0, \ldots, c_{n_c}\right], \tag{1.43}$$

$\chi(t)$ is the regression vector (or data vector), formed by the measured input and output variables in a previous instant, defined as

$$\chi^{T}(t) = \left[y(t-1), \ldots, y(t-n_a), u(t-1), \ldots, u(t-n_b-1), v(t), \ldots, v(t-n_d)\right], \tag{1.44}$$

and $e(t)$ is a random disturbance (usually white noise). Finally, in order to estimate the real parameter vector θ from the available information, it is necessary to consider the system with the structure

$$y(t) = \chi^T(t)\,\hat{\theta} + \hat{e}(t), \tag{1.45}$$

where $\hat{\theta}$ is the adjustable parameter vector, $\hat{e}(t)$ is the corresponding adjustment error in time t, and $\hat{\theta}$ is selected as the vector that minimizes the performance index

$$J = \sum_{t=1}^{N} \hat{e}^2(t) = \hat{e}^T(t)\,\hat{e}(t). \tag{1.46}$$

From (1.45)

$$\hat{e}(t) = y(t) - \chi^T(t)\,\hat{\theta} = Y - X\hat{\theta}, \tag{1.47}$$

and (1.46) is expressed as

$$J = \left(Y - X\hat{\theta}\right)^T \left(Y - X\hat{\theta}\right) = Y^TY - \hat{\theta}^TX^TY - Y^TX\hat{\theta} + \hat{\theta}^TX^TX\hat{\theta}. \tag{1.48}$$

Computing the first partial derivative of J with respect to $\hat{\theta}$ and matching zero, it yields to

$$\frac{\partial J}{\partial \hat{\theta}} = -2X^TY + 2X^TX\hat{\theta} = 0, \tag{1.49}$$

then $X^TX\hat{\theta} = X^TY$. The second partial derivative is positively defined as

$$\frac{\partial J^2}{\partial \hat{\theta}^2} = 2\left(X^TX\right) > 0, \tag{1.50}$$

i.e., J has a minimum. Therefore, the system parameter vector, from the least squares estimator is

$$\hat{\theta} = \left[X^TX\right]^{-1}\left[X^TY\right]. \tag{1.51}$$

1.8.4 Recursive Least Squares (Applied to Chapter 6)

One of the disadvantages of the OLS method is the need to work with a set of past values, which implies that it must be executed offline. To address this, the recursive version of this method allows the estimation of the model parameters at each sample interval when new data are available. It is outlined slightly below, according to [68]. In the recursive schema, input/output data are available at each sampling interval. From the model based on past information (summarized in $\hat{\theta}(t-1)$), an estimate $\hat{y}(t)$ of the current output is obtained. This is compared to the measured output $y(t)$ to generate an error $\epsilon(t)$ and update the model that corrects $\hat{\theta}(t-1)$ to

the new value $\hat{\theta}(t)$. It is effective to simply store the calculation of the previous estimate at time t, indicated by $\hat{\theta}(t)$, and get the new estimates $\hat{\theta}(t+1)$ by an update step involving only the new measurement. The estimate in step $t+1$ is given by the expression

$$\hat{\theta}(t+1) = \left[\chi^T(t+1)\,\chi(t+1)\right]^{-1}\chi^T(t+1)\,\mathbf{y}(t+1), \tag{1.52}$$

where

$$\chi(t+1) = \begin{bmatrix} \chi^T(1) \\ \chi^T(2) \\ \vdots \\ \chi^T(t) \\ \cdots \\ \chi^T(t+1) \end{bmatrix} = \begin{bmatrix} \chi(t) \\ \cdots \\ \chi^T(t+1) \end{bmatrix} \tag{1.53}$$

and

$$\mathbf{y}(k+1) = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(k) \\ \cdots \\ y(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{y}(k) \\ \cdots \\ y(k+1) \end{bmatrix}. \tag{1.54}$$

We observe that

$$\chi^T(t+1)\,\chi(t+1) = \chi^T(t)\,\chi(t) + \chi(t+1)\,\chi^T(t+1), \tag{1.55}$$

$$\chi^T(t+1)\,\mathbf{y}(t+1) = \chi^T(t)\,\mathbf{y}(t) + \chi(t+1)\,y(k+1), \tag{1.56}$$

and defining

$$\begin{aligned} P(t) &= \left[\chi^T(t)\,\chi(t)\right]^{-1} \\ B(t) &= \chi^T(t)\,\mathbf{y}(t), \end{aligned}, \tag{1.57}$$

the new estimate is given by

$$\begin{aligned} \hat{\theta}(t+1) &= P(t+1)\,B(t+1), \\ \hat{\theta}(t) &= P(t)\,B(t). \end{aligned} \tag{1.58}$$

The update from $B(t)$ to $B(t+1)$ is given by

$$B(t+1) = B(t) + \chi(t+1)\, y(t+1) \tag{1.59}$$

and the update from $P(t)$ to $P(t+1)$ is obtained by using the matrix inversion lemma $(A+BCD)^{-1} = A^{-1} - A^{-1}B\left(C^{-1} + DA^{-1}B\right)^{-1}DA^{-1}$. For

$$P^{-1}(t+1) = P^{-1}(t) + \chi(t+1)\,\chi^T(t+1), \tag{1.60}$$

it yields to

$$P(t+1) = P(t)\left[I_m - \chi(t+1)\left(1 + \chi^T(t+1)\,P(t)\,\chi(t+1)\right)^{-1}\chi^T(t+1)\,P(t)\right]. \tag{1.61}$$

The error variable is

$$\epsilon(t+1) = y(t+1) - \chi^T(t+1)\,\hat{\theta}(t) \tag{1.62}$$

and replacing $y(t+1)$ into Eq. (1.59)

$$B(t+1) = B(t) + \chi(t+1)\,\chi^T(t+1)\,\hat{\theta}(t) + \chi(t+1)\,\epsilon(t+1). \tag{1.63}$$

The new estimates are obtained by substituting $B(t)$, $B(t+1)$ into Eq. (1.58):

$$\hat{\theta}(t+1) = \hat{\theta}(t) + P(t+1)\,\chi(t+1)\,\epsilon(t+1). \tag{1.64}$$

1.9 General Methodology Based on Recursive Least Squares for Nonlinear Systems

Consider the generalized system (see [69])

$$Ay(t) = Bu(t-1) + Du(t) + \mathcal{D}(t) + Ce(t), \tag{1.65}$$

where

$$A = 1 + a_1 z^{-1} + \cdots + a_{n_a} z^{n_a},$$
$$B = b_0 + b_1 z^{-1} + \cdots + b_{n_b} z^{n_b},$$
$$D = d_0 + d_1 z^{-1} + \cdots + d_{n_d} z^{n_d},$$
$$\mathcal{D} = \mathbf{d}_0 + \mathbf{d}t + \cdots + \mathbf{d}_{n_\mathbf{d}} t^{n_\mathbf{d}},$$
$$C = 1 + c_1 z^{-1} + \cdots + c_{n_c} z^{n_c}.$$

When these coefficients are unknown, then they are taken as parameters to be determined, either through measurements or estimates. The system is represented in Figure 1.8, where $u(t)$ is the control input, $v(t)$ is the measurable disturbance, $\underline{d}(t) = \mathbf{d}(t)$ is the nonmeasurable disturbance, and $e(t)$ is the random disturbance.

For estimation purposes, this model can be rewritten as follows:

$$Y(t) = \mathcal{X}^T(t)\,\theta + e(t), \tag{1.66}$$

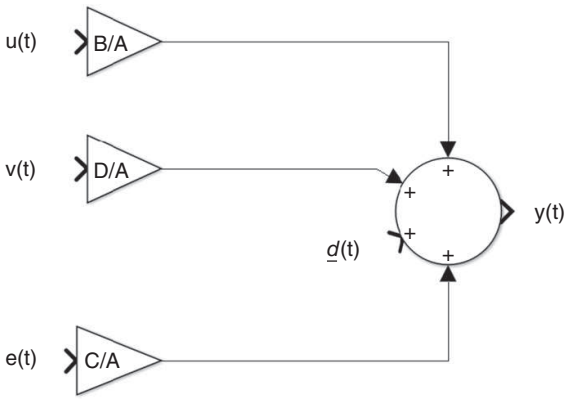

Figure 1.8 Simulink system's representation.

where $\theta^T = [-a_1, \cdots, -a_{n_a}, b_0, b_1, \cdots, b_{n_b}, d_0, d_1, \cdots, d_{n_d}, \mathbf{d}_0, \mathbf{d}_1, \cdots, \mathbf{d}_{n_\mathbf{d}}, c_1, \cdots, c_{n_c}]$ and $\mathcal{X}^T(t)$ is known as the regression vector and is constructed from the measurements of the input and output variables, defined by $\mathcal{X}^T(t) = [y(t-1), \cdots, y(t-n_a), u(t-1), \cdots, u(t-n_b-1), v(t), \cdots, v(t-n_d), 1, t, \cdots, t^{n_\mathbf{d}} e(t), \cdots, e(t-n_c)]$. This vector contains the past noise measurements, which are difficult to obtain, therefore they are assumed to be zero, so

$$\theta^T = [-a_1, \cdots, -a_{n_a}, b_0, b_1, \cdots, b_{n_b}, d_0, d_1, \cdots, d_{n_d}, \mathbf{d}_0, \mathbf{d}_1, \cdots, \mathbf{d}_{n_\mathbf{d}}]$$

and

$$\mathcal{X}^T(t) = [y(t-1), \cdots, y(t-n_a), u(t-1), \cdots, u(t-n_b-1), v(t), \cdots, v(t-n_d), 1, t, \cdots, t^{n_\mathrm{d}}].$$

It is assumed that

$$Y(t) = \mathcal{X}^T(t)\theta \tag{1.67}$$

is an exact description of the system and we want to determine θ, that is, the vector of true parameters of the system, then the model

$$Y(t) = \mathcal{X}^T(t)\hat{\theta} + \hat{e}(t) \tag{1.68}$$

also gets the system output, where $\hat{\theta}$ is an adjustable parameter vector and $\hat{e}(t)$ the measurement error at time t. We wish to select $\hat{\theta}$ such that the modulation error is minimized in some sense. Note that

$$(Y(t) = \mathcal{X}^T(t)\theta - \left(Y(t) = \mathcal{X}^T(t)\hat{\theta} + \hat{e}(t)\right)$$
$$\Rightarrow\ 0 = \mathcal{X}^T(t)\theta - \mathcal{X}^T(t)\hat{\theta} + \hat{e}(t),$$

so

$$\hat{e}(t) = \mathcal{X}^T(t)\left(\theta - \hat{\theta}\right); \tag{1.69}$$

so the modeling error depends on $\hat{\theta}$ values and in some cases on the measurement noise of $\mathcal{X}^T(t)$, modifying (1.69) to

$$\hat{e}(t) = \mathcal{X}^T(t)\left(\theta - \hat{\theta}\right) + e(t). \tag{1.70}$$

Assuming that the system (1.67) has been active for some time and N data have been taken, using the model (1.68) the measured data are expressed as follows:

$$\begin{bmatrix} Y(1) \\ Y(2) \\ \vdots \\ Y(N) \end{bmatrix} = \begin{bmatrix} \mathcal{X}^T(1) \\ \mathcal{X}^T(2) \\ \vdots \\ \mathcal{X}^T(N) \end{bmatrix} \hat{\theta} + \begin{bmatrix} \hat{e}(1) \\ \hat{e}(2) \\ \vdots \\ \hat{e}(N) \end{bmatrix}. \tag{1.71}$$

To be able to estimate the parameters unequivocally, the number of measurements N must not be less than the number of unknown parameters in the vector θ, m. In the noise-free case, $e(t) = 0$, this equation can be solved as a system of linear equations with $N = m$ unknowns; however, in practice, $N >> m$.

We represent Eq. (1.71) as

$$\bar{Y} = \bar{X}\hat{\theta} + \bar{e}, \tag{1.72}$$

in the same way, $\bar{e} = \bar{Y} - \bar{X}\hat{\theta}$. We select the parameter vector $\hat{\theta}$ such that it minimizes

$$J = \sum_{t=1}^{N} \bar{e}^2(t) = \bar{e}^T\bar{e} = \bar{Y}^T\bar{Y} - \hat{\theta}^T\bar{X}^T\bar{Y} - \bar{Y}^T\bar{X}\hat{\theta} + \hat{\theta}\bar{X}^T\bar{X}\hat{\theta}. \tag{1.73}$$

According to the fundamental theorem of the calculus of variations, [70], the first variation, the partial derivative of J with respect to $\hat{\theta}$, is equal to zero, that is

$$0 = -2\bar{X}^T\bar{Y} + 2\bar{X}^T\bar{X}\hat{\theta},$$

and from the second variation, we have

$$\frac{\partial^2 J}{\partial\hat{\theta}^2} = 2\left(\bar{X}^T\bar{X}\right),$$

which, being positive definite, allows us to obtain the solution for $\hat{\theta}$ of the form

$$\hat{\theta} = \left(\bar{X}^T\bar{X}\right)^{-1}\bar{X}^T\bar{Y}. \tag{1.74}$$

The dimension of $\left(\bar{X}^T\bar{X}\right)$ depends on the number of parameters, not the number of measurements. In general, for m parameters, the matrix will be of dimension $m \times m$. If the matrix is singular, then this term cannot be obtained; however, this is easily solved by not setting $u(t)$ a constant, thus avoiding that the determinant of the matrix is 0.

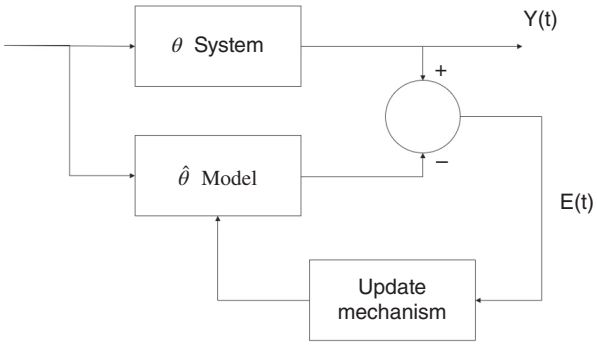

Figure 1.9 Recursion scheme.

If the system is contaminated with noise, then this feedback may be sufficient excitation. Another way to get enough excitation is to vary the reference signal.

We consider the scheme shown in Figure 1.9.

At time t, we have

$$\hat{\theta}(t) = (\bar{X}^T(t)\bar{X}(t))^{-1}\bar{X}^T(t)\bar{Y}(t), \tag{1.75}$$

where t indicates the number of the measurement in discrete time t, that is

$$\bar{X}(t) = \begin{bmatrix} \mathcal{X}^T(1) \\ \mathcal{X}^T(2) \\ \vdots \\ \mathcal{X}^T(t) \end{bmatrix}, \bar{Y}(t) = \begin{bmatrix} Y(1) \\ Y(2) \\ \vdots \\ Y(t) \end{bmatrix}.$$

So, in time $(t+1)$, we have

$$\bar{X}(t+1) = \begin{bmatrix} \mathcal{X}^T(1) \\ \mathcal{X}^T(2) \\ \vdots \\ \mathcal{X}^T(t+1) \end{bmatrix} = \begin{bmatrix} \bar{X}^T(t) \\ \mathcal{X}^T(t+1) \end{bmatrix}, \quad \bar{Y}(t+1) = \begin{bmatrix} Y(1) \\ Y(2) \\ \vdots \\ Y(t+1) \end{bmatrix} = \begin{bmatrix} \bar{Y}(t) \\ Y(t+1) \end{bmatrix}, \tag{1.76}$$

and the estimated parameters vector in time $(t+1)$ is

$$\hat{\theta}(t+1) = \left(\bar{X}^T(t+1)\bar{X}(t+1)\right)^{-1}\bar{X}^T(t+1)\bar{Y}(t+1). \tag{1.77}$$

Therefore, we have

$$\bar{X}^T(t+1)\bar{X}(t+1) = \begin{bmatrix} \bar{X}^T(t) & \mathcal{X}(t+1) \end{bmatrix} \begin{bmatrix} \bar{X}(t) \\ \mathcal{X}^T(\mathrm{t}+1) \end{bmatrix}$$
$$= \bar{X}^T(t)\,\bar{X}(t) + \mathcal{X}(t+1)\mathcal{X}^T(t+1)$$

and

$$\bar{X}^T(t+1)\,\bar{Y}(t+1) = \begin{bmatrix} \bar{X}^T(t) & \mathcal{X}(t+1) \end{bmatrix} \begin{bmatrix} \bar{Y}(t) \\ \bar{Y}(t+1) \end{bmatrix}$$
$$= \bar{X}^T(t)\,\bar{Y}(t) + \mathcal{X}(t+1)\,\bar{Y}(t+1).$$

Considering the following definitions

$$P(t) = \left(\bar{X}^T\bar{X}\right)^{-1} \text{ and} \tag{1.78}$$

$$B(t) = \bar{X}\,\bar{Y}, \tag{1.79}$$

we can rewrite (1.77) as

$$\hat{\theta}(t+1) = P(t+1)\,B(t+1) \text{ and} \tag{1.80}$$

$$\hat{\theta}(t) = P(t)\,B(t). \tag{1.81}$$

This implies

$$P^{-1}(t+1) = P^{-1}(t) + \mathcal{X}(t+1)\mathcal{X}^T(t+1) \text{ and} \tag{1.82}$$

$$B(t+1) = B(t) + \mathcal{X}(t+1)\bar{Y}(t+1). \tag{1.83}$$

Applying the matrix inversion lemma to $P^{-1}(t+1)$, we have

$$P(t+1) = P(t) - P(t)\,\mathcal{X}(t+1)\left(1 + \mathcal{X}^T(t+1)\,P(t)\,\mathcal{X}(t+1)\right)^{-1}\mathcal{X}^T(t+1)\,P(t).$$

Since the reverse term is actually a scalar, then without any problem it can be rewritten as follows:

$$P(t+1) = P(t)\left[I_m - \mathcal{X}(t+1)\,\frac{\mathcal{X}^T(t+1)\,P(t)}{1 + \mathcal{X}^T(t+1)\,P(t)\,\mathcal{X}(t+1)}\right]. \tag{1.84}$$

We define the estimation error as follows:

$$E(t+1) = \bar{Y}(t+1) - \mathcal{X}^T(t+1)\,\hat{\theta}. \tag{1.85}$$

Then (1.83) is rewritten as

$$B(t+1) = B(t) + \mathcal{X}(t+1)\,E(t+1) + \mathcal{X}(t+1)\,\mathcal{X}^T(t+1)\,\hat{\theta}, \tag{1.86}$$

and the Eq. (1.80) is possible to perform the pertinent operations and substitutions as follows:

$$\begin{aligned}
\hat{\theta}(t+1) &= P(t+1)\left[B(t) + \mathcal{X}(t+1)E(t+1) + \mathcal{X}(t+1)\mathcal{X}^T(t+1)\hat{\theta}(t)\right], \\
&= P(t+1)\left[B(t) + \mathcal{X}(t+1)\mathcal{X}^T(t+1)\hat{\theta}(t)\right] \\
&\quad + P(t+1)\mathcal{X}(t+1)E(t+1), \\
&= P(t+1)\left[P^{-1}(t)\hat{\theta}(t) + \mathcal{X}(t+1)\mathcal{X}^T(t+1)\hat{\theta}(t)\right] \\
&\quad + P(t+1)\mathcal{X}(t+1)E(t+1), \\
&= P(t+1)\left[P^{-1}(t) + \mathcal{X}(t+1)\mathcal{X}^T(t+1)\right]\hat{\theta}(t) \\
&\quad + P(t+1)\mathcal{X}(t+1)E(t+1), \\
&= P(t+1)P^{-1}(t+1)\hat{\theta}(t) + P(t+1)\mathcal{X}(t+1)E(t+1).
\end{aligned}$$

Finally,

$$\hat{\theta}(t+1) = \hat{\theta}(t) + P(t+1)\mathcal{X}(t+1)E(t+1). \tag{1.87}$$

Therefore, the recursive least squares algorithm is performed as follows:

- **i.** Using new data taken from the system measurement form $\mathcal{X}(t+1)$.
- **ii.** Create $E(t+1)$ using (1.85).
- **iii.** With (1.84) create $P(t+1)$ the first iteration starts the vector with initial conditions of P big enough, a diagonal matrix of 100000 works fine.
- **iv.** Calculate $\hat{\theta}(t+1)$ from (1.87).
- **v.** Wait for the next time and go back to step 1.

1.10 Optimal Controllers

1.10.1 Linear Quadratic Regulator

The linear quadratic controller (LQR) is defined as a dynamic programming algorithm that is conducive to finding optimal controllers [70–72].

The nonlinear control system is represented as follows:

$$\dot{x}(t) = f(x(t), u(t)),\ x(t_0) = x_0, \tag{1.88}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ $yf(.,.) \in \mathbb{R}^n$ satisfies the Lipschitz condition and the control $u(.) \in \mathbb{U} \subset \mathbb{R}^m$.

The control $u \in \mathbb{U}$ once set, the system (1.88) determines the trajectory with initial condition x_0 at the instant t_0.

1.10.1.1 The Infinite Horizon Optimal Control Problem

To find a control $u \in \mathbb{U}$, such that in a closed loop of the system (1.88), there is a trajectory that converges to the origin in a minimum time or properly with the minimum of energy and its convergence is as fast as possible. Having said earlier, the

problem of optimal control is posed as the problem of optimal control, so it establishes a performance index, which penalizes the state of the system and control, also called cost functional [70–72].

$$J^{u(\cdot)}(t_0, x_0) = \int_{t0}^{\infty} L(x(t), u(t))\, dt, \tag{1.89}$$

where L is a positive definite scalar function. L penalizes the state $x(.)$ and the energy $u(.)$. We consider the case of only minimizing the energy $L(x, u) = u^T u$. If the control u^* minimizes the cost functional $J^{u(\cdot)}(t_0, x_0)$, it is as follows:

$$J^{u^*}(.)(t_0, x_0) \leq J^{u(\cdot)}(t_0, x_0), \forall\, u(.) \in \mathbb{U},$$

where u^* is called optimal control. In 1975, Richard Bellman proposed dynamic programming to solve optimization problems for differential equations with constraints. Optimal control with constraints is interpreted as the equation of state. The method consists of replacing Eqs. (1.88) and (1.89) and contains a minimization in the space $\mathbb{U}$, by the matrix differential equation in partial derivatives, which is called the Hamilton–Jacobi–Bellman equation:

$$0 = \min_{u \in \mathbb{U}} \{L(x, u) + \nabla_x V(x) \cdot f(x, u)\}, \quad t\,[t_0, \infty], \quad x \in \mathbb{R}^n, \tag{1.90}$$

where $\frac{dV(x)}{dt} = \nabla_x V(x) \cdot f(x, u)$. The Hamilton–Jacobi–Bellman equation satisfies the Bellman function $V(x)$, defined as follows:

$$V(x_0) = \min_{u \in \mathbb{U}} J^u(\cdot)(t_0, x_0). \tag{1.91}$$

Bellman's function (1.91) must satisfy Eq. (1.90) and it is necessary that the function of $V(.)$ be continuously differentiable along the trajectories of (1.88).

The following is a brief explanation of the application of the aforementioned equations to solve the infinite horizon optimal control problem in the case of linear systems LQR problem.

Consider the linear system in the form of state–space representation:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \tag{1.92}$$

such that $A \in \mathbb{R}^{n \times n}$ y $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and define a quadratic performance index

$$J = \int_{t0}^{tf} \left\{x^T(t) Q x(t) + u^T(t) R u(t)\right\} dt, \tag{1.93}$$

where $Q \in \mathbb{R}^{n \times n}$ y $R \in \mathbb{R}^{m \times m}$ are positive semidefinite and positive definite matrices, respectively. The set of admissible controls $\mathbb{U}$ and the system (1.92) is set closed loop with u.

Thus $u \in \mathbb{U}$, such that it is the linear function of the state $x(t)$, that said, $u(t) = u(x(t))$. It is assumed that there are admissible controls for the system (1.92) and that the performance index (1.93) reaches a minimum for:

$$u(x(t)) = u^*(x(t)). \tag{1.94}$$

Now, a positive definite function $V(x(t))$ is sought so that its derivative on all trajectories of the system (1.92) is as follows:

$$\frac{dV(x(t))}{dt} = -L^*(x(t), \quad u^*(t)), \tag{1.95}$$

where

$$L^*(x(t), \quad u^*(t)) = x^T(t)\,Qx(t) + u^{*T}(x(t))\,Ru^*(x(t))$$

is integrated from 0 to ∞ on both sides of (1.94) and is obtained as follows:

$$\lim_{t\to\infty} V(x)(t) - V(x_0) = -\int_0^\infty L^*(x(t,x_0), u^*(t))dt.$$

So, the system is stable (u^*) and is an admissible control; we have the following equation:

$$V(x_0) = \int_0^\infty L^*(x(t,x_0), u^*(t))dt.$$

It is now known that $x(t,x_0)$ denotes the solution to the system (1.92) in closed loop with (1.94). Although $L^*(x(t,x_0), u^*(t))$ is positive definite, for obviousness $V(x(t))$ is a Lyapunov function for the system (1.92). The expression (1.95) can be seen as follows:

$$\frac{dV(x(t))}{dt} + L^*(x(t), \quad u^*(t)) = 0 \tag{1.96}$$

Rewriting is as follows:

$$\min_{u\in\mathbb{U}} \left(\frac{dV(x(t))}{dt}\Big|_{1.92} + x^T(t)\,Qx(t) + u^T(t)\,Ru(t) \right) = 0. \tag{1.97}$$

So, Eq. (1.97) is known as the Hamilton–Jacobi–Bellman equation for linear systems, and the function satisfying this equation is called the Bellman function for systems (1.92). When the Bellman function $V(x(t))$ is known, its solution is (1.97) with respect to u, since it gives the optimal control of $u^*(t)$. The function of $V(x(t))$ can be considered as a Lyapunov function for the system (1.92) in a closed loop with $u^*(t)$. Of course, for a linear system and $u^*(t) = u^*(x(t))$, a closed-loop Lyapunov function can be proposed as follows:

$$V(x(t)) = x^T(t)\,Px(t), \tag{1.98}$$

where $P \in \mathbb{R}^{n\times n}$ is defined as positive and the derivative of $V(x(t))$ is calculated for all system trajectories (1.92),

$$\frac{dV(x(t))}{dt}\Big|_{(1.92)} = 2x^T(t)\,P\,(Ax(t) + Bu(t))\,. \tag{1.99}$$

The derivative (1.99) is substituted into (1.97) to have the form:

$$\min_{u\in\mathbb{U}} \left(Ax^TPAx(t) + 2x^T(t)\,PBu(t) + \left(x^T(t)\,Qx(t) + u^T(t)\,Ru(t)\right)\right). \tag{1.100}$$

From Eq. (1.100), it is desired to find the optimal control law u^*. It can be seen that in (1.100), a strongly convex quadratic function is present with respect to u.

Given the following, the existence of a global minimum for (1.100) is guaranteed. Consequently, in the results of the calculus of variations, the first variation of the function to be optimized with respect to u should be zero.

$$\frac{\partial}{\partial u}\left(2Ax^{T}PAx(t)+2x^{T}(t)PBu(t)+\left(x^{T}(t)Qx(t)+u^{T}(t)Ru(t)\right)\right)=0.$$

Therefore,

$$2B^{T}Px(t)+2Ru(t)=0. \tag{1.101}$$

Therefore, the optimum control law for the system (1.92) is

$$u^{*}(t)=-R^{-1}B^{T}Px(t). \tag{1.102}$$

The expression (1.102) is said to be optimal given that,

$$\frac{\partial^{2}}{\partial u^{2}}\left(2Ax^{T}PAx(t)+2x^{T}(t)PBu(t)+\left(x^{T}(t)Qx(t)+u^{T}(t)Ru(t)\right)\right)=2R>0.$$

It is assumed that the matrix R is strictly positive definite and according to the sufficient optimality condition $u^{*}(t)$ given (1.92) is optimal.

Now, we proceed to formulate the matrix P of the Bellman function $V(x(t))$. To know it, it is necessary to calculate the time derivative of (1.99) and then evaluate the trajectories of the system (1.92) in a closed loop with the optimal control law (1.102):

$$\frac{dV(x(t))}{dt}=x(t)^{T}\left(A^{T}P+PA\right)x(t)+u^{*T}(t)B^{T}Px(t)+x(t)^{T}PBu^{*}(t). \tag{1.103}$$

Substituting Eqs. (1.102) and (1.103) into (1.99), we obtain that for every state $x(t)$

$$x(t)^{T}\left(A^{T}P+PA+Q-PBR^{-1}B^{T}P\right)x(t)=0.$$

The procedure is terminated as follows:

$$A^{T}P+PA+Q-PBR^{-1}B^{T}P=0. \tag{1.104}$$

Equation (1.104) is called the algebraic Riccati equation. Given the matrices Q positive semidefinite and R strictly positive, this equation can be solved numerically with respect to P, which defines the optimal control u^{*}.

1.10.2 Optimal PI

Next, the process to tune a Proportional Integral (PI) controller optimally by designing the optimal quadratic regulator (LQR) will be described, detailing the [73] procedure for a first-order process (see Figure 1.10). A large number of processes are modeled as first order as well, as controlled by a proportional–integral–derivative (PID) control; regularly, this model is of the form:

$$G(s)=\frac{be^{-hs}}{s+a}. \tag{1.105}$$

Although this model is not the exact one, it is used for design purposes.

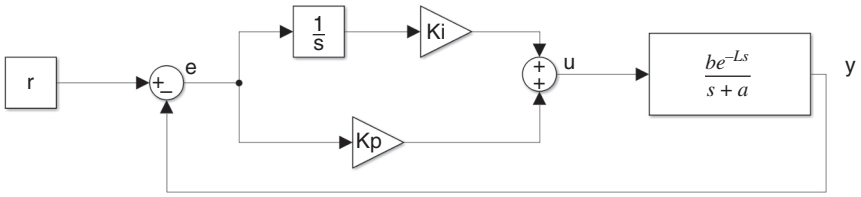

Figure 1.10 PI control diagram in Simulink.

Now consider a PI control:

$$u(t) = Kp\left[e(t) + \frac{1}{T_i}\int_0^t e(\tau)\,d\tau\right], \tag{1.106}$$

which is a control law that is sufficient to control a first-order plant.

Since the system is linear, it is possible to set $r = 0$, which would imply $e = -y$; therefore, we have $(s + a)\,e = be^{-hs}u$, and in the time domain

$$\dot{e} = -ae(t) - bu(t - h), \tag{1.107}$$

if $r \neq 0$, the result is unmodified.

By defining

$$x_1 = \int_0^t e(\tau)\,d\tau, \quad x_2 = e(t),$$

we can determine the extended state as

$$x(t) = [x_1\, x_2]^T.$$

We derivate the state as

$$\frac{d}{dt}\int_0^t e(\tau)\,d\tau = e(t),$$

obtaining

$$\dot{x} = \begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}x_2\\ \dot{e}(t)\end{bmatrix}$$

Now in state–space, we have

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & -a\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ -b\end{bmatrix}u(t - h). \tag{1.108}$$

In this system, all the state variables are available and it is completely controllable,

$$C = \begin{bmatrix}B & AB\end{bmatrix} = \begin{bmatrix}0 & -b\\ -b & ab\end{bmatrix},$$

since the controllability matrix rank is 2 or full rank. Therefore, the state feedback control is

$$u(t) = \begin{bmatrix} k_i & k_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = k_1 \int_0^t e(\tau)\, d\tau + k_2 e(t), \tag{1.109}$$

a PI control. The parameters of this PI will be obtained using the optimal quadratic regulator methodology.

Therefore, we proceed to find k_1 and k_2; substituting in the Riccati algebraic equation,

$$A^T P + PA - PBR^{-1}B^T P + Q = 0, \tag{1.110}$$

the known values are

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix} \text{ and } B = \begin{bmatrix} 0 \\ -b \end{bmatrix}$$

as well as

$$Q = \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix} \text{ and } P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$$

and

$$\begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix}^T \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} + \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix} - \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} 0 \\ -b \end{bmatrix} R^{-1} \begin{bmatrix} 0 & -b \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} -q_1 & 0 \\ 0 & -q_2 \end{bmatrix}$$

which, performing the procedure, is

$$\begin{bmatrix} 0 & P_{11} - aP_{12} \\ P_{11} - aP_{21} & P_{12} + P_{21} - 2aP_{22} \end{bmatrix} - \begin{bmatrix} \frac{1}{R} b^2 P_{12} P_{21} & \frac{1}{R} b^2 P_{12} P_{22} \\ \frac{1}{R} b^2 P_{21} P_{22} & \frac{1}{R} b^2 P_{22}^2 \end{bmatrix} = \begin{bmatrix} -q_1 & 0 \\ 0 & -q_2 \end{bmatrix}$$

$$\begin{bmatrix} -\frac{1}{R} b^2 P_{12} P_{21} & P_{11} - aP_{12} - \frac{1}{R} b^2 P_{12} P_{22} \\ P_{11} - aP_{21} - \frac{1}{R} b^2 P_{21} P_{22} & P_{12} + P_{21} - 2aP_{22} - \frac{1}{R} b^2 P_{22}^2 \end{bmatrix} = \begin{bmatrix} -q_1 & 0 \\ 0 & -q_2 \end{bmatrix}.$$

If we consider that P is symmetric, then $P_{12} = P_{21}$, which simplifies

$$\begin{bmatrix} -\frac{1}{R}b^2P_{12}^2 & P_{11} - P_{12}(a + \frac{1}{R}b^2P_{22}) \\ P_{11} - P_{12}(a + \frac{1}{R}b^2P_{22}) & 2P_{12} - 2aP_{22} - \frac{1}{R}b^2P_{22}^2 \end{bmatrix} = \begin{bmatrix} -q_1 & 0 \\ 0 & -q_2 \end{bmatrix}.$$

This brings us to the equation system

$$-q_1 = -\frac{1}{R}b^2P_{12}^2, \tag{1.111}$$

$$0 = P_{11} - P_{12}\left(a + \frac{1}{R}b^2P_{22}\right), \tag{1.112}$$

$$-q_2 = 2P_{12} - 2aP_{22} - \frac{1}{R}b^2P_{22}^2 \tag{1.113}$$

From (1.111), we directly obtain

$$P_{12} = \frac{\sqrt{Rq_1}}{b}. \tag{1.114}$$

Using the general formula for second-order equations in (1.113) and having the analytical positive value,

$$\begin{aligned} P_{22} &= \frac{2a - \sqrt{4a^2 - 4\left(-\frac{1}{R}b^2\right)\left(\frac{2\sqrt{Rq_1}}{b} + q_2\right)}}{2(\frac{-1}{R}b^2)} \\ &= \frac{a - \sqrt{a^2 - \left(-\frac{1}{R}b^2\right)(2P_{12} + q_2)}}{\frac{-1}{R}b^2} \\ &= \frac{-Ra + R\sqrt{a^2 - \left(-\frac{1}{R}b^2\right)(2P_{12} + q_2)}}{b^2}, \end{aligned}$$

we have

$$P_{22} = \frac{-Ra + \sqrt{R^2a^2 + Rb^2\,(2P_{12} + q_2)}}{b^2}. \tag{1.115}$$

Now, for Eq. (1.112), we obtain

$$P_{11} = aP_{12} + R^{-1}b^2P_{22}P12. \tag{1.116}$$

If

$$F = R^{-1}B^TP = R^{-1}\begin{bmatrix} 0 & -b \end{bmatrix}\begin{bmatrix} P_{11} & P_{12} \\ P_{12} & P_{22} \end{bmatrix} = -R^{-1}b\begin{bmatrix} P_{12} & P_{22} \end{bmatrix},$$

then in the closed-loop matrix

$$A_c = A - BF = \begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix} + \begin{bmatrix} 0 \\ -b \end{bmatrix} R^{-1} b \begin{bmatrix} P_{12} & P_{22} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -R^{-1}b^2 P_{12} & -\sqrt{a^2 + R^{-1}b^2(2P_{12} + q_2)} \end{bmatrix}. \tag{1.117}$$

So, we have a linear system of the form

$$\dot{x}(t) = Ax(t) + Bu(t-h). \tag{1.118}$$

The performance index of this control is given by

$$J = \int_0^\infty \left(x^T(t) Q x(t) + u^T(t) R u(t) \right) dt \tag{1.119}$$

using Ricatti's Eq. (1.110) and factoring Q into $Q = H^T H$ the LQR solution, [74], in the following theorem.

Theorem 1.5 [73] For the linear process with time delay (1.118), if the pair (A, B) is controllable and the pair (H, A) is observable, then the optimal control that minimizes the performance index (1.119) is given by

$$u(t) = -R^{-1}B^T P e^{\left(A - BR^{-1}B^T P\right)t} e^{A(h-t)} x(t), \quad 0 \le t < h$$

and

$$u(t) = -R^{-1}B^T P e^{\left(A - BR^{-1}B^T P\right)t} x(t), \quad t \ge h,$$

where P is the positive definite solution of (1.110). The resulting system also is stable,

which is reduced as

$$u(t) = \begin{cases} -Fe^{A_c t} e^{A(t-h)} x(t), & 0 \le t < h \\ -Fe^{A_c t} x(t), & t \ge h \end{cases}. \tag{1.120}$$

If the general form is replaced in Q, that is,

$$Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{12} & q_{22}, \end{bmatrix},$$

then

$$-q_{11} = -\frac{1}{R} b^2 P_{12}^2,$$

$$-q_{12} = P_{11} - P_{12}\left(a + \frac{1}{R} b^2 P_{22}\right),$$

$$-q_{22} = 2P_{12} - 2aP_{22} - \frac{1}{R} b^2 P_{22}^2,$$

which, in turn, implies a slight change in the form of P_{11}, but P_{12} and P_{22} remain the same; the change occurs from (1.116) to

$$P_{11} = aP_{12} + R^{-1}b^2P_{22}P12 - q_{12}; \tag{1.121}$$

because of this, it is possible to consider Q with its diagonal shape.

To obtain $e^{A(h-t)}$ and e^{A_ct}, an inverse Laplace transform needs to be done: starting with $e^{A(h-t)}$,

we have

$$e^{A(h-t)} = \mathcal{L}^{-1}\left\{(sI - A)^{-1}\right\}|_{t=h-t}\,.$$

Recalling the equality $(sI - A)^{-1}$, we have

$$(sI - A)^{-1} = \begin{bmatrix} s & -1 \\ 0 & s+a \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{s} & \frac{1}{s^2+as} \\ 0 & \frac{1}{s+a} \end{bmatrix}.$$

Applying the inverse Laplace transform, we have

$$\mathcal{L}^{-1}\left\{\begin{bmatrix} \frac{1}{s} & \frac{1}{s^2+as} \\ 0 & \frac{1}{s+a} \end{bmatrix}\right\}\Bigg|_{t=h-t} = \begin{bmatrix} 1 & \frac{1-e^{-at}}{a} \\ 0 & e^{-at} \end{bmatrix}\Bigg|_{t=h-t} = \begin{bmatrix} 1 & \frac{1-e^{-a(h-t)}}{a} \\ 0 & e^{-a(h-t)} \end{bmatrix}.$$

Following the same procedure for e^{A_ct},

$$e^{A_ct} = \mathcal{L}^{-1}\{(sI - A_c)^{-1}\},$$

we have

$$(sI - A_c)^{-1} = \begin{bmatrix} s & -1 \\ R^{-1}b^2P_{12} & s + \sqrt{a^2 + R^{-1}b^2(2P_{12} + q_2)} \end{bmatrix}^{-1}.$$

Defining $\hat{a}_1 = \sqrt{a^2 + R^{-1}b^2\,(2P_{12} + q_2)}$ and $\hat{a}_2 = R^{-1}b^2P_{12}$, then we have

$$(sI - A_c)^{-1} = \frac{1}{s^2 + \hat{a}_1 s + \hat{a}_2}\begin{bmatrix} s + \hat{a}_1 & 1 \\ -\hat{a}_2 & s \end{bmatrix}. \tag{1.122}$$

Assuming that the denominator roots are:

$$\alpha_1 = \frac{-\hat{a}_1 + \sqrt{\hat{a}_1^2 - 4\hat{a}_2}}{2}$$

$$\alpha_2 = \frac{-\hat{a}_1 - \sqrt{\hat{a}_1^2 - 4\hat{a}_2}}{2},$$

we can rewrite (1.122) as

$$(sI - A_c)^{-1} = \frac{1}{(s-\alpha_1)(s-\alpha_2)} \begin{bmatrix} s+\hat{a}_1 & 1 \\ -\hat{a}_2 & s \end{bmatrix}.$$

Applying the inverse Laplace transform, we obtain

$$\begin{aligned} \mathcal{L}^{-1}\{(sI - A_c)^{-1}\} &= \frac{1}{\alpha_1 - \alpha_2} \begin{bmatrix} e^{t\alpha_1}(\hat{a}_1+\alpha_1) - e^{t\alpha_2}(\hat{a}_1+\alpha_2) & e^{t\alpha_1} - e^{t\alpha_2} \\ -\hat{a}_2(e^{t\alpha_1} - e^{t\alpha_2}) & \alpha_1 e^{t\alpha_1} - \alpha_2 e^{t\alpha_2} \end{bmatrix} \\ &= \begin{bmatrix} f_{11}(t) & f_{12}(t) \\ f_{21}(t) & f_{22}(t) \end{bmatrix} \end{aligned} \tag{1.123}$$

According to the control form (1.120), from $0 \le t < h$

$$\begin{aligned} u(t) &= -R^{-1}b \begin{bmatrix} P_{12} & P_{22} \end{bmatrix} \begin{bmatrix} f_{11}(t) & f_{12}(t) \\ f_{21}(t) & f_{22}(t) \end{bmatrix} \begin{bmatrix} 1 & \dfrac{1-e^{-a(h-t)}}{a} \\ 0 & e^{-a(h-t)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= -R^{-1}b \begin{bmatrix} f_{11}(t)P_{12} + f_{21}(t)P_{22} & e^{-ah-t}\Big[f_{12}(t)P_{12} + f_{22}(t)P_{22} \\ & -\dfrac{1}{a}(f_{11}(t)P_{12} + f_{21}(t)P_{22})\Big] \\ & +\dfrac{1}{a}(f_{11}(t)P_{12} + f_{21}(t)P_{22}) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \end{aligned}$$

and also for $t \ge h$

$$\begin{aligned} u(t) &= -R^{-1}b \begin{bmatrix} P_{12} & P_{22} \end{bmatrix} \begin{bmatrix} f_{11}(t) & f_{12}(t) \\ f_{21}(t) & f_{22}(t) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= -R^{-1}b \begin{bmatrix} f_{11}(h)P_{12} + f_{21}(h)P_{22} & f_{12}(h)P_{12} + f_{22}(h)P_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \end{aligned}$$

We have the following theorem.

Theorem 1.6 [73] The optimal LQR control for the process (1.105) and state (1.108) is given in the form of a PI controller (1.106) or (1.109), where, for $0 \le t < h$,

$$\begin{aligned} K_i(t) &= R^{-1}b\,(f_{11}(t)\,P_{12} + f_{21}(t)\,P_{22}), \\ K_p(t) &= R^{-1}b\Big\{e^{-ah-t}\Big[f_{12}(t)\,P_{12} + f_{22}(t)\,P_{22} - \frac{1}{a}\,(f_{11}(t)\,P_{12} + f_{21}(t)\,P_2 2)\Big], \\ &\quad + \frac{1}{a}\,(f_{11}(t)\,P_{12} + f_{21}(t)\,\mathrm{P}_{22})\Big\} \end{aligned}$$

and in the same way for $t \geq h$,

$$K_i(t) = R^{-1}b\left(f_{11}(h)P_{12} + f_{21}(h)P_{22}\right),$$

$$K_p(t) = R^{-1}b\left(f_{12}(h)P_{12} + f_{22}(h)P_{22}\right);$$

q_1, q_2, and R are the tuning parameters.

In the case where $t \geq h$, the optimal controller ζ, the damping coefficient, and ω_n, the undamped natural frequency, can be obtained from the characteristic equation of A_c, (1.117):

$$\Delta = s^2 + \sqrt{\left(a^2 + R^{-1}b\left(\sqrt{q_1R} + q_2b\right)\right)}s + R^{-1}b\sqrt{q_1R}. \tag{1.124}$$

When it is compared to the general form of a second-order system:

$$\Delta_g = s^2 + 2\zeta\omega_n s + \omega_n^2,$$

then the following is obtained

$$\omega_n^2 = R^{-1}b\sqrt{q_1R}$$

$$2\zeta\omega_n = \sqrt{(a^2 + R^{-1}b(\sqrt{q_1R} + q_2b))},$$

which implies

$$\omega_n = \sqrt{R^{-1}b\sqrt{q_1R}}, \tag{1.125}$$

$$\zeta = \frac{\sqrt{(a^2 + R^{-1}b(\sqrt{q_1R} + q_2b))}}{2\sqrt{R^{-1}b\sqrt{q_1R}}}, \tag{1.126}$$

or, in the same way, obtaining the desired values of ζ and ω_n, the q_1 and q_2 values are given as

$$q_1 = \frac{\omega_n^4 R}{b^2}, \tag{1.127}$$

$$q_2 = \frac{[(4\zeta^4 - 2)\omega_n^2 - a^2]R}{b^2}. \tag{1.128}$$

Thus, the performance index is

$$J = \int_0^\infty \left\{ \left[\int_0^t e(\tau)d\tau \quad e(t) \right] \begin{bmatrix} \frac{\omega_n^4 R}{b^2} & 0 \\ 0 & \frac{[(4\zeta^4 - 2)\omega_n^2 - a^2]R}{b^2} \end{bmatrix} \begin{bmatrix} \int_0^t e(\tau)d\tau \\ e(t) \end{bmatrix} + Ru^2(t) \right\},$$

$$= R\left\{ \int_0^\infty \left[\frac{\omega_n^4}{b^2}\left(\int_0^t e(\tau)d\tau \right)^2 + \frac{(4\zeta^2 - 2)\omega_n^2 - a^2}{b^2} e^2(t) + u^2(t) \right] dt \right\}.$$

This indicates that the performance index J is proportional to R, so it is possible to choose $R = 1$; this implies that energy consumption and convergence are affected by R.

To conclude, the direct identities of a second-order system are exposed [75]; the maximum overshoot M_p and the establishment time t_s, for $\zeta < 1$, are as follows:

$$M_p = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}},$$

$$t_s = \frac{4}{\zeta\omega_n}.$$

1.10.3 Pontryagin Maximum Principle

The idea of exploiting optimization at process is not novel; the novelty here is how the optimization is formulated and solved together with the experimental proof-of-concept. Pontryagin maximum principle and current with comparatively new developments, such as methods of inverse dynamic problems, decomposition, and embedded systems, are used. In addition, Pontryagin maximum principle is used for establishing positive properties of optimal controls with a minimum of mathematical operation. Pontryagin maximum principle is applied for solving the variational problem of optimal open-loop control. Pontryagin maximum principle was used for solving applied problems of control and other problems of dynamic optimization. That is, commonly, the optimal control actions are saturation-limited, which means that the optimal control consists generally of both singular and bang-bang arcs; moreover, the optimized feeding command relies on accurate model knowledge and high-quality measurement of the main state variables for its real-time implementation, which could be a serious drawback for realistic processes implementations [76]. Other fact is that the structure of the corresponding objective function to be optimized might include nonlinear terms of the state variables and, therefore, the cost of the optimization rises, as the planning of the feeding, based on those control law, is large [77]. The Pontryagin's maximum principle is a relevant mathematical formalism and is widely employed to search for possible optimal conditions for controlled processes [78]. However, such optimization approaches might not have the necessary ability to ensure either feasibility or optimization or even, in many cases, to point out how near optimization a specific feasible solution is [79].

Case Study Research 4

First, in search of a compact notation, let us consider the following generic representation of any batch reactor modeled from the mass balance:

$$\dot{x} := \frac{dx}{dt} = f(x) + g(x)\,u, \tag{1.129}$$

where $x \in \mathbb{R}^n$ is the state variable vector that represents the concentration variables within the reactor. The term $f(x)$, $f : \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear vector field, whose components can be derived from the right side of each differential equation of the mass balance, where $f(x) \subset \Sigma \in C^\infty$ and Σ is a compact set; $g(x) \in \mathbb{R}^{n\times m}$ is an invertible bounded matrix, whose entries are also derived from right side of the mass

balance, but taking into account the terms multiplying a variable defined by the feeding rate $u = F_{in}$, where $u \in \mathbb{R}^m$, with $m \leq n$ is known as the control input, which is computed by the instrumentation on the experimental real-time implementation. Now, let us consider a function given by:

$$\Psi(\varrho) = \int_0^T \varrho(x, \dot{x}, u)\, dt. \tag{1.130}$$

Then, if such a function has an extreme (maximum or minimum), the extreme can be computed from the derivative:

$$\xi\Psi(\varrho) = \int_0^T \xi\varrho(x, \dot{x}, u)\, dt. \tag{1.131}$$

Now, from the general form of the model $\dot{x} = f(x) + g(x)u$, we have:

$$f(x) = \dot{x} - g(x)u. \tag{1.132}$$

The aforementioned computation suggests that functions $\Psi(\varrho)$ and $\varrho(x, \dot{x}, u)$ can be respectively taken as follows:

$$\Psi(\varrho) \equiv f(x) \quad \text{and} \quad \varrho(x, \dot{x}, u) \equiv \dot{x} - g(x)u. \tag{1.133}$$

Henceforth, since by definition $\Psi(\varrho) = \int_0^T \varrho(x, \dot{x}, u)\, dt$, we have

$$\Psi(\varrho) = \int_0^T (\dot{x} - g(x)u)\, dt = 0. \tag{1.134}$$

The equation for computing the feeding rate can be obtained through the calculation of u, which stabilizes the model around an extreme point as follows.

1.11 Observer-based Controllers

In order to use full-state feedback, it is necessary that all states of a given system are measured. In practice, this is not always possible. In some cases, sensors are simply not available or cannot be made for the states that one would desire to measure. If the process to be controlled is observable and a reasonably accurate model is available for it, then it is possible to use a modified model to estimate the states of the process that are not measured. The main model modification is to add a term that corrects model errors and internal disturbances. Under proper conditions, the state estimator can be used instead of direct measurements of full system states. Such a modified model used in a feedback system is called a State Observer. In the case of full-order observers, all the states of the control object are estimated whether or not this is necessary. In some cases, control objects have states that can be effectively measured without noise, while other states have large noise components. Such model reductions may be advantageous because they make it possible to simplify the control system and thus reduce system costs. An observer-based control structure could be useful in the case of anti-windup bump-less transfer, e.g., switching between manual (open loop) and automatic (closed loop) control modes. A more complex control system often uses several control modes, e.g., failsafe or limp home mode. The control strategies and variables may vary, while the observer is the same [80–82].

1.12 Examples of Modeling, Simulation, and Practical Platforms for Industrial Processes

An embedded system is said to be a system that monitors processes and in some cases controls them. One of the main characteristics of a real-time system is to correct calculations that depend not only on the accuracy of the result but also on the time at which it is produced, avoiding delays if this is the case. A real-time system must respond to changes in the configurations that were established at the beginning within the established time to be performed, otherwise, this can jeopardize the system that is being monitored at that time [83].

According to [84], an embedded system is an engineering-developed artifact that is directly related to computation and is subject to physical constraints that arise through the two ways in which computational processes interact with the physical world [85]:

i. Reaction to a physical environment.
ii. Execution on a physical platform.

The main common reaction constraints specify deadlines, performance, and jitter, originating from behavioral requirements. The common execution constraints are limited by the available processor speeds, processor power, and hardware failure rates, giving way to the different implementation options. It is important to emphasize that control theory is concerned, in this case, with constraining system reactions, while computer engineering is in charge of execution.

Some of the main characteristics that an embedded system should possess, according to [84], are the following:

- Multiple operations can be performed by the same chip (processor).
- Within the period of the process to be monitored or controlled, it must be fully automatic.
- It is considered that it must be compact, as well as have a quick response regarding the process in which it is operating.

Some possible configurations that can be found in an embedded system are as follows:

- **Real-time system**: One of the main characteristics is that it produces a result within a defined timeframe; working outside these timeframes results in system failure. They are capable of preserving as much of the system's data and capabilities as possible in the face of disturbances. [86].
- **Online system**: They are characterized by being semi-permanently connected to a computer, terminal, cash register, or other device with a certain processing capacity. They are capable of encoding and decoding information immediately (see Figure 1.11).

Considering this section, we present the proposed embedded system for the development applied to a bioreactor:

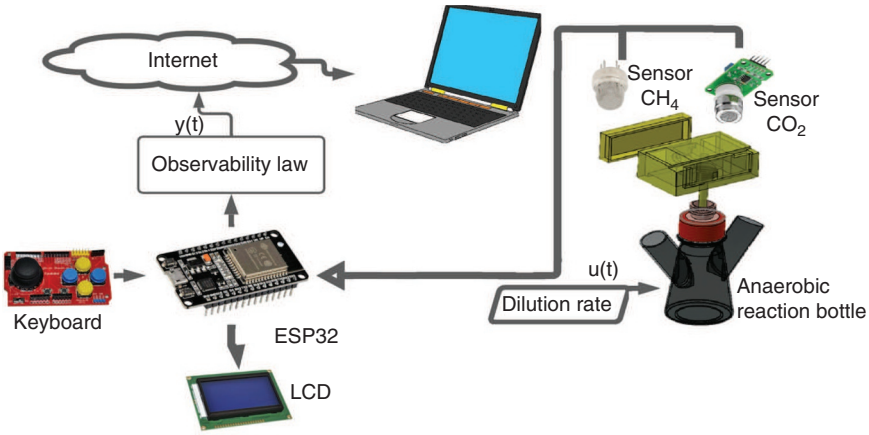

Figure 1.11 Embedded bioreactor system.

1.12.1 LabVIEW

LabVIEW is the acronym of Laboratory Virtual Instrument Engineering Workbench. It is a graphical programming environment that engineers use to develop automated testing systems for research, validation, and production. In LabVIEW, it is possible to acquire data and process signals, create automated test bench and validation systems, and control industrial environments, as well as prototypes.

It is a high-level language programming environment in which virtual instruments can be created. These are software modules that simulate the front panel of a physical instrument with buttons, Light-Emitting Diodes (LEDs), controls, and display screens.

The hybrid programming allows us to combine textual and graphical programming to increase the efficiency. Text-based mathematical formulas can directly enter on the LabVIEW block diagram with the *MathScript node*. MATLAB [87] functions are available, for example, for signal and data processing, vector and matrix operations, among others.

The foundations of graphical programming can be found directly in the website of National Instruments, which is the developer and its owner [88, 89]. A real-time data acquisition and control system based on virtual instrumentation should allow:

i. Define a test procedure.
ii. Selection of the instruments.
iii. Provide initial values.
iv. Analyze results using a user interface.

The components of a program in LabVIEW consist of:

i. The front panel, which is the user interface.
ii. Block diagram containing the graphical source code that defines the functionality.
iii. Icon and connectors that identify each virtual instrument and allow us to use it.

In this book, we will only mention the basic elements and functions to create virtual instruments for implementing the parameter identification algorithms and for developing advanced control techniques, creating a user interface to monitor the process variables.

The used LabVIEW blocks and structures are listed as follows:

- *Timed loop*: It is an iterative structure, widely used in real-time operations, that executes one or more sub-diagrams, or frames, sequentially each iteration of the loop at a specific period. It is possible to configure the clock source, the execution period, and the phase.
- *Sequence structure*: Statements are executed in the order of appearance when all data are available.
- *MathScript node*: It is a structure in the LabVIEW block diagram that provides the ability to put text-based MathScript code in line with graphical LabVIEW code.
- *Shift register*: It is a local variable that allows us to transfer the values from the end of an iteration to the beginning of the next. After the loop code executes, data enter the right shift register and are passed to the left shift register on the next iteration of the loop.
- *Sample Compression Express VI*: It acquires a large number of data points and compresses those data points into a smaller number of points.
- *Build Array Function*: It concatenates multiple arrays or appends elements to an *n*-dimensional array.
- *Index Array Function*: It returns the element or subarray of an *n*-dimension array at index.
- *XY Graph*: It expects a cluster input that contains the waveforms' X and Y arrays formed using a *Bundle Function*.
- *Write to Measurement File Express VI*: It can write data to a text-based measurement file (.lvm), a binary measurement file with headers (.tdm), a binary measurement file without headers (.tdms), or a Microsoft Excel file (.xlsx).

One of the commonly used devices that provide the computer with the ability to acquire general signals is the data acquisition card, which can be used with the Data Acquisition (DAQ) *Assistant*. It is a graphical interface for configuring measurement tasks, channels, and scales.

1.13 Sensors

1.13.1 ESP32

The ESP32 (datasheet) card is designed for the IoT and improves the integration of wireless technology, such as Wi-Fi and Bluetooth, into a chip. It manages to make the board more practical due to added advantage of small-size chip (equivalent to an Arduino nano), which makes it a more practical option for the implementation of projects that require a small and aesthetic space.

The ESP32 is a development board containing a System-on-Chip (SoC) that consists of all or most of the elements of a computer or any other electronic system on a

single integrated circuit or chip, such as Random-Access Memory (RAM) memory, analog and digital inputs, and outputs. These systems have the advantage of low-power consumption. The specifications of the board are shown in Section 1.13.2.

In the specific case of this work, the digital to analog converter (DAC) inputs will be used since they are in charge of receiving the analog signal from the Methane and Natural Gas Concentration (MQ-4) sensor module. As can be seen in the list of specifications, the resolution of this module is 12 (bits), in addition to the 32 (bits) architecture. One of the most attractive features of the aforementioned board is the fact that the board has a Bluetooth module and an integrated Wi-Fi module, which improves the transmission of data to other devices, such as a computer or a smartphone.

1.13.2 Specifications

The technical specifications of the ESP32 board are as follows:

- Dual-core Xtensa^{TM} processor LX6 of 32 bits
- Clock speed: between 160 and 240 MHz
- 520 kB of RAM memory
- Wi-Fi integrated
- Bluetooth 4.2–2.4 GHz
- 36 GPIO pins
- 16 × Analog to Digital Converters with 12-bit resolution and can be programmed with input limit at 1 V, 2 V, and 4 V
- 2 × digital to analog converter DAC of 8 bits
- Up to 16 pulse width modulation channels can be defined
- 2 × universal asynchronous receiver transmitter or serial ports
- 2 × channels I^2C
- 2 × channels I^2S
- 4 × channels serial peripheral interface

1.13.3 Sensor Infrastructure

For the construction of the sensor, we will use a prototyping board (Protoboard), the ESP32 board, and the MQ-4 sensor module (this can be replaced by any module of the sensor family).

Both the sensor and the board need to be powered by 5 V direct current. The sensor module can send two types of signals: digital and analog. The digital signal indicates the presence or absence of a gas with a certain concentration, and the analog signal gives us, as a result, a voltage variation depending on how much gas is in the environment. This is what is required for this case study. Therefore, the analog input of the board is used to receive the data provided by the sensor module. These variations are received by the board in a range of 0–3.3 V with a resolution of 12 bit (4096 samples). This indicates how much gas is there with a concentration ranging from 200 to 10 000 parts per million (ppm). A model was created to contain the sensors and make their implementation easier for the user, so that in case of having more than one sensor, they can be contained in a single structure.

It is intended to implement the estimator that responds in the best way to make a virtual sensor. This leads to using the material that is accessible to students or researchers. As mentioned earlier, the virtual sensor or state estimator needs to at least have the physical measurement of the process. This can be, for example, the methane produced, carbon dioxide, or hydrogen sulfide, so it takes into account the materials that will be seen in the following subsections. Also mentioned is the method to calibrate the sensor, which depends on the effectiveness of the state estimator.

1.14 Module MQ

The MQ-4 module has an alumina ceramic microtube (AL_2O_3) and a tin dioxide sensitive layer ($Sn\ O_2$). In other models of the MQ family, the material of the sensitive layer changes to give sensitivity to other types of gases. The measuring electrode and heater provide the necessary conditions for the sensitive components to work on. The measurement occurs when the gas is trapped inside the grid and reacts with the ceramic tube, which causes the resistance of the circuit to increase, thus giving the value of the gas concentration. These sensors have the particularity of being very economical with a fast response time.

1.15 Sensor Operation

In order to create the sensor assembly, the ESP32 board and the MQ modules are used as gas sensors. The methane concentration measurement is adjusted by means of a load resistor R_l. The heater has a 5 V connection so that it can operate. It is powered separately to have the best possible measurement.

For the connection, the analog pins A0 GPIO 33 are used to measure the potential difference on the output pins of the sensor module and the programming has been done in the software (LabVIEW). For the creation of the user interface, the Arduino Integrated Development Environment (IDE) for data processing is used on the board, which along with Bluetooth module facilitates the communication between the PC and the board.

To obtain the methane concentration measurement, the following steps are followed in the programming.

- The analog values of pin A0 are converted to a voltage measurement.
- The resistance R_s is calculated, which indicates the resistance at the sensor in the presence of methane gas.
- The ratio of $\frac{R_s}{R_{s\ \text{of the air}}}$ (resistance at the sensor in the presence of methane gas/ resistance at the sensor in the presence of air).
- The ppm value is obtained on a linear scale according to the relationship calculated in the previous point.
- The value (ppm) is converted to a logarithmic scale.
- The value (ppm) is converted to the percentage of CH_4 in air.

For all this, we used as a guide the Arduino programming for the MQ-4 sensor obtained from examples in the Arduino community.

With these measurements performed by our sensor, we intend to perform the estimation of parameters for which a sensor is not available, using a nonlinear estimator observer that will be fed with the input of the low-cost sensor.

1.15.1 Sensor Calibration

For the calibration process, different samples taken by a CO_2 probe were used and related to the measurements displayed by the MQ module with the LSA. The process is described in the following text.

The least squares method is used for sensor parameterization resulting in a polynomial describing the sensor behavior and is defined as follows for a Single-Input Single-Output (SISO) system:

Let there be a SISO measurement system of the form:

$$y_i = Mx_i + N. \tag{1.135}$$

For $i = 1, 2, \ldots, n$ samples, the estimation of the coefficients M and N is defined by:

$$\begin{bmatrix} M \\ N \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & n \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{n} y_i x_i \\ \sum_{i=1}^{n} y_i \end{bmatrix},$$

where y is the output and x is the input. This is the simplest case that approximates a straight line. If we want to improve the parameterization, then we can use the higher degree algorithm given for n measurements and defined by a polynomial approximation:

$$y_i = a_m x_i^m + a_{m-1} x_i^{m-1} + \cdots + a_2 x_i^2 + a_1 x_i + a_0.$$

Coefficient of regression, $a_k\ k = 1, 2, 3, a, \ldots, m$, is defined by

$$\underbrace{\begin{bmatrix} a_m \\ \vdots \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}}_{X} = \underbrace{\begin{bmatrix} \sum_{i=1}^{n} x_i^{2m} & \cdots & \sum_{i=1}^{n} x_i^{m+2} & \sum_{i=1}^{n} x_i^{m+1} & \sum_{i=1}^{n} x_i^{m} \\ \vdots & \ddots & \cdots & \cdots & \cdots \\ \sum_{i=1}^{n} x_i^{m+2} & \cdots & \sum_{i=1}^{n} x_i^{4} & \sum_{i=1}^{n} x_i^{3} & \sum_{i=1}^{n} x_i^{2} \\ \sum_{i=1}^{n} x_i^{m+1} & \cdots & \sum_{i=1}^{n} x_i^{3} & \sum_{i=1}^{n} x_i^{2} & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i^{m} & \cdots & \sum_{i=1}^{n} x_i^{2} & \sum_{i=1}^{n} x_i & n \end{bmatrix}}_{S^{-1}} \underbrace{\begin{bmatrix} \sum_{i=1}^{n} y_i x_i^m \\ \vdots \\ \sum_{i=1}^{n} y_i x_i^2 \\ \sum_{i=1}^{n} y_i x_i \\ \sum_{i=1}^{n} y_i \end{bmatrix}}_{Y} \tag{1.136}$$

with S^{-1} nonsingular for $n = 2m$ minimum number of measurements.

Table 1.1 Test data.

Feature	Value
Concentration (ppm)	Resistance R_s/R_0
200	1.8
300	1.6
500	1.4
800	1.1
1000	1
2000	0.79
3000	0.68
5000	0.59
7000	0.5
10 000	0.35

Table 1.2 Parameters obtained from LSA.

Feature	Value
a_0	8.863152555487748 e^{+03}
a_1	4.696522688472391 e^{+04}
a_2	–1.944975271307188 e^{+05}
a_3	2.414530821540368 e^{+05}
a_4	–1.256669450302607 e^{+05}
a_5	2.371184194478921 e^{+04}

As a first calibration, we used the data provided by the sensor datasheet. These data give us an idea of how the sensor should behave. The data are shown in Table 1.1 as follows:

With these data, the LSA is applied to characterize the sensor. In this case, the first-degree and fifth-degree algorithms were used to select the one that best fits the values provided (Table 1.2).

These data result in the following polynomial:

$$Y = 23711.8419x^5 + -125666.9450x^4 + 241453.0821x^3 + 46965.2268x^2 + x^1 + 8863.1525$$

1.15.2 Methane Sensor Programming Codes

For the sensor, we took advantage of the compatibility of the ESP32 board with the Arduino IDE to be able to program it in this development environment, which

provides us with some in-built libraries for Arduino such as BluetoothSerial.h. This allows us to take advantage of the integrated Bluetooth module on the board. The code used for receiving data on the board is as follows:

```
//Sensor MQ-4 Methane
\#include "BluetoothSerial.h" //bookstore Bluetooth
int gas_sensor = 34; //Sensor pin
float m = -0.368; //pending
float b = 1.104; //Y-interception
float R0 = 11.820; //sensor resistance in pure air
BluetoothSerial SerialBT;
void setup() {
Serial.begin(115200); //Baud rate
SerialBT.begin("Sensor CH4");
pinMode(gas_sensor, INPUT); // pin as input
}
void loop() {
float sensor_volt;
float RS_gas;
float ratio;
float sensorValue = analogRead(gas_sensor);
sensor_volt = 0;
ratio = 0;
RS_gas = 0;
sensor_volt = sensorValue*(3.3/4096);
RS_gas = ((3.3*10.0)/sensor_volt)-1.14;
ratio = RS_gas/R0; // Gets the relationship RS_gas/RS_air
double ppm_log = (log10(ratio)-b)/m;
double ppm = pow(10, ppm_log);
double porcentaje = ppm/10000; //converts to percentage
Serial.print("ppm = "); //shows the data
Serial.println(ppm);
SerialBT.print(ppm);
SerialBT.print(" ppm");
SerialBT.print(" ; ");
SerialBT.print(percentage);
SerialBT.print(" % ");
SerialBT.print("; ");
SerialBT.print(sensorValue);
SerialBT.println(" mV ");
delay(1000);
}
```

For this, we first declare the libraries and variables to be used, followed by configuring the input pins and the serial port that allows us to send the data obtained to a Bluetooth device. In the main part of the code, the value read to a voltage scale is converted to a maximum of 3.3 V, which is what the board can get. This value is

divided by 4096 corresponding to 12 bits of resolution in the analog inputs of the board. Once we have this value, a relationship is calculated between the values of the internal resistance of the sensor in the presence of gas and pure air. The sensor does not have a linear behavior, so a linear approximation is made using the calibration table provided in the datasheet of the device. For that reason, the calculation is made in a logarithmic scale, which is then returned to a linear scale to display the value in ppm. In the end, all the final data are displayed in the serial port that will be responsible for sending them through Bluetooth to any device that can read serial Bluetooth. The ppm value is used as the input for the extended Luenberger observer. Therefore, the calibration and validation of the sensor is a very important step because if the sensor sends erroneous measurements, the observer will not be able to estimate the desired variables correctly.

1.15.3 Carbon Dioxide Sensor Programming

```
#define         MG_PIN                      (36)
#define         BOOL_PIN                    (39)
#define         DC_GAIN                     (8.5)



#define         READ_SAMPLE_INTERVAL        (50)
#define         READ_SAMPLE_TIMES           (5)



#define         ZERO_POINT_VOLTAGE          (0.322)
#define         REACTION_VOLTGAE            (0.263)



float CO2Curve[3]={2.602,ZERO_POINT_VOLTAGE,
                    (REACTION_VOLTGAE/(2.602-4))};

void setup()
{
    Serial.begin(115200);
    pinMode(BOOL_PIN, INPUT);
    digitalWrite(BOOL_PIN, HIGH);

    Serial.print("MG-811 Demostration\n");
}

void loop()
{
    int percentage;
```

```
    float volts;

    volts = MGRead(MG_PIN);
    Serial.print( "SEN-00007:" );
    Serial.print(volts);
    Serial.print( "V        " );

    percentage = MGGetPercentage(volts,CO2Curve);
    Serial.print("CO2:");
    if (percentage == -1) {
        Serial.print( "<400" );
    } else {
        Serial.print(percentage);
    }

    Serial.print( "ppm" );
    Serial.print("\n");

    if (digitalRead(BOOL_PIN) ){
        Serial.print( "=====BOOL is HIGH======" );

    } else {
        Serial.print( "=====BOOL is LOW======" );

    }
        Serial.print("\n");

        delay(200);
}



float MGRead(int mg_pin)
{
    int i;
    float v=0;

    for (i=0;i<READ_SAMPLE_TIMES;i++) {
        v += analogRead(mg_pin);
        delay(READ_SAMPLE_INTERVAL);
    }
    v = (v/READ_SAMPLE_TIMES) *3.3/4095 ;
    return v;
}
```

```
int MGGetPercentage(float volts, float *pcurve)
{
    if ((volts/DC_GAIN )>=ZERO_POINT_VOLTAGE) {
       return -1;
    } else {
       return pow(10, ((volts/DC_GAIN)-pcurve[1])
                     / pcurve[2]+pcurve[0]);
    }
}
```

1.15.4 Carbon Dioxide Vernier Probe Programming

```
#include "VernierLib.h"
VernierLib Vernier;

float sensorReading;

void setup() {
  Serial.begin(9600);
  Serial.println("CLEARSHEET");
  Serial.println("LABEL,Date,Time,CO2");
  Serial.println("RESETTIMER");
  Vernier.autoID();
}

void loop() {
  sensorReading = Vernier.readSensor(); /
  Serial.print("DATA,DATE,TIME,");
  Serial.print(sensorReading);
  Serial.print(" ");
  Serial.println(Vernier.sensorUnits());
  delay(1000);
}
```

1.15.5 MATLAB Function

```
function dy = bio4(t,y)
dy=zeros(4,1);


Ks = 150; %251;
Ki = 50; %3506;
Kmax = 2.9; %4355;
Umax = (Kmax*(y(1)/(Ks+y(1)+Ki))*y(2))^1.9
Fd = 0.0009;
S0 = 30;
```

```
Yxs = 3;
Ysx = 0.426;
D = 0.001;
alpha = 0.29;
beta = 0.20;
Yxco2 = 0.67;
Yxch4 = 0.73; %performance
% plant
dy(1) = -(Umax/Ysx) + D*(S0-y(1));
dy(2) = (Umax/Yxs) - Fd*y(2)-y(2)*D;
dy(3) = Umax*Yxco2*(y(3)^alpha)-D*(y(3));
dy(4) = Umax*Yxch4*(y(4)^beta)*y(2)- D*(y(4));
end
```

References

1 J. Alves, T. M. Lima, and P. D. Gaspar. Is Industry 5.0 a human-centred approach? A systematic review. *Processes*, 11(1):193, 2023.

2 S. Huang, B. Wang, X. Li, P. Zheng, D. Mourtzis, and L. Wang. Industry 5.0 and society 5.0 – comparison, complementation and co-evolution. *Journal of Manufacturing Systems*, 64:424–428, 2022.

3 S. Grabowska, S. Saniuk, and B. Gajdzik. Industry 5.0: improving humanization and sustainability of industry 4.0. *Scientometrics*, 127(6):3117–3144, 2022.

4 S. Saniuk, S. Grabowska, and M. Straka. Identification of social and economic expectations: Contextual reasons for the transformation process of Industry 4.0 into the Industry 5.0 concept. *Sustainability*, 14(3):1391, 2022.

5 D. Ivanov. The Industry 5.0 framework: viability-based integration of the resilience, sustainability, and human-centricity perspectives. *International Journal of Production Research*, 61(5):1683–1695, 2023.

6 P. A. López-Pérez, D. J. Hernández Melchor, L. V. Hernández Quijano, M. I. Sánchez Contreras, and R. I. Beltrán Hernández. Process integration via a sustainable biorefinery approach using agro-industrial residues and photosynthetic consortia. In *Biotechnological Processes for Green Energy, and High Value Bioproducts by Microalgae, and Cyanobacteria Cultures* (ed. Martínez-Roldán, A.d.J.), pages 213–228. Cham: Springer, 2024.

7 P. A. López-Pérez, R. Aguilar Lopez, and R. Femat. *Control in bioprocessing: Modeling, estimation and the use of soft sensors*. Hoboken, NJ: John Wiley & Sons, 2020.

8 R. Aguilar López, E. Alvarado Santos, F. Thalasso, and P. A. López-Pérez. Monitoring ethanol fermentation in real time by a robust state observer for uncertainties. *Chemical Engineering & Technology*, 47(5):779–790, 2024.

9 P. A. López-Pérez, M. López López, C. A. Núñez Colín, H. Mukhtar, R. Aguilar López, and V.e Peña Caballero. A novel nonlinear sliding mode observer to estimate biomass for lactic acid production. *Chemical Product and Process Modeling*, 18(4):565–580, 2023.

10 A. Hassan, M. G. A. Elrahman Sayed, S. A. Ali, N. S. Abdulkhaleq Mohammed, M. Dahlan, and G. Shaker. Using artificial intelligence (AI) in the management process. In *Proceedings of the International Conference on Business and Technology*, pages 549–557. Springer, 2022.

11 M. Borchardt, G. M. Pereira, G. S. Milan, A. R. Scavarda, E. O. Nogueira, and L. C. Poltosi. Industry 5.0 beyond technology: an analysis through the lens of business and operations management literature. *Organizacija*, 55(4):305–321, 2022.

12 D. O. Madsen and T. Berg. An exploratory bibliometric analysis of the birth and emergence of industry 5.0. *Applied System Innovation*, 4(4):87, 2021.

13 Y. Cohen and G. Singer. A smart process controller framework for industry 4.0 settings. *Journal of Intelligent Manufacturing*, 32(7):1975–1995, 2021.

14 P. A. López-Pérez, M. I. Neria González, and R. Aguilar López. Increasing the bio-hydrogen production in a continuous bioreactor via nonlinear feedback controller. *International Journal of Hydrogen Energy*, 40(48):17224–17230, 2015.

15 P. A López-Pérez, R. Aguilar López, and M. I. Neria González. Cadmium removal at high concentration in aqueous medium: mediated by *Desulfovibrio alaskensis*. *International Journal of Environmental Science and Technology*, 12:1975–1986, 2015.

16 V. Peña Caballero, P. A. López-Pérez, O. G. García Salas, and A. T. Morales Vargas. Experimental validation off-line a nonlinear controller for removal of chromium using non-living cells of *Yarrowia lipolytica*. *Preparative Biochemistry & Biotechnology*, pages 1–10, 2024.

17 R. Aguilar López, P. A. López-Pérez, and R. Femat. Unit intensification for exothermic chemical reactors via minimum stabilization time. *Chemical Engineering Science*, 289:119869, 2024.

18 L. E. Amabilis Sosa, A. E. Rodríguez Mata, R. Baray Arana, I. Robles Vega, V. A. Gonzalez Huitrón, and P. A. López-Pérez. Robust fractional control based on high gain observers design (RNFC) for a *Spirulina* maxima culture interfaced with an advanced oxidation process. *Open Chemistry*, 21(1):20220214, 2023.

19 P. A. Lopéz-Peréz, A. E. Rodriguez Mata, O. Hernández González, L. E. Amabilis Sosa, R. Baray Arana, and J. Leon Borges. Design of a robust sliding mode controller for bioreactor cultures in overflow metabolism via an interdisciplinary approach. *Open Chemistry*, 20(1):120–129, 2022.

20 R. Aguilar López, B. Ruiz Camacho, M. I. Neria González, E. Rangel, O. Santos, and P. A. López-Pérez. State estimation based on nonlinear observer for hydrogen production in a photocatalytic anaerobic bioreactor. *International Journal of Chemical Reactor Engineering*, 15(5):20170004, 2017.

21 C. Peng, D. Yue, and Q. L. Han. *Communication and control for networked complex systems*. Springer, 2015.

22 E. Alvarado Santos, J. L. Mata Machuca, P. A. López-Pérez, R. A. Garrido Moctezuma, F. Pérez Guevara, and R. Aguilar López. Comparative analysis of a family of sliding mode observers under real-time conditions for the monitoring in the bioethanol production. *Fermentation*, 8(9):446, 2022.

23 C. Lucizano, A. A. de, Andrade, J. F. Facó Blumetti, and A. de Freitas Gomes. Revisiting the automation pyramid for the Industry 4.0. In *Proceedings of the 2023 15th IEEE International Conference on Industry Applications (INDUSCON)*, pages 1195–1198. IEEE, 2023.

24 M. Wollschlaeger, T. Sauter, and J. Jasperneite. The future of industrial communication: Automation networks in the era of the Internet of things and Industry 4.0. *IEEE Industrial Electronics Magazine*, 11(1):17–27, 2017.

25 E. M. Martinez, P. Ponce, I. Macias, and A. Molina. Automation pyramid as constructor for a complete digital twin, case study: a didactic manufacturing system. *Sensors*, 21(14):4656, 2021.

26 S. Garrido, J. Muniz Jr, and V. B. Ribeiro. Operations management, sustainability & industry 5.0: a critical analysis and future agenda. *Cleaner Logistics and Supply Chain*, page 100141, 2024.

27 P. A. López-Pérez, E. Alvarado Santos, D. J. Hernández Melchor, F. J. Martínez Farias, and E. Jiménez Muñoz. *Chapter 4. A Generic Mathematical Model for Pest Control with Time Delay Dynamics Based on Silico Approaches*, pages 95–128. Nova, 12 2023. ISBN 979-8-89113-292-4.

28 R. O. Cañizares Villanueva, C. Ramírez López, P. A. López-Pérez, and D. J. Hernández Melchor. Biofertilization by nitrogen-fixing cyanobacteria, nutrient supplementation, and growth promotion. In *Biotechnological Processes for Green Energy, and High Value Bioproducts by Microalgae, and Cyanobacteria Cultures* (eds. Martínez-Roldán, A.d.J.), pages 69–82. Cham: Springer, 2024.

29 R. Aguilar López, P. A. López-Pérez, M. I. Neria González, and A. R. Domínguez Bocanegra. Observer based adaptive model for a class of aerobic batch bioreactor. *Revista Mexicana de Ingeniería Química*, 9(1):29–35, 2010.

30 G. Miles, J. Wiles, T. Claypoole, P. Drake, P. A. Henry, L. J. Johnson, S. Lowther, M. W. Tobias, and J. H. Windle. *Techno Security's Guide to Securing SCADA: A Comprehensive Handbook on Protecting the Critical Infrastructure*. Syngress, 2008.

31 P. A. López-Pérez, O. S. Castillo Baltazar, R. Aguilar López, V. Peña Caballero, and E. Castañeda Vallejo. Virtual sensors for biofuels production: a brief mathematical description for synthesis of algorithms. *Comptes Rendus de l'Academie Bulgare des Sciences*, 72(10):1383–1392, 2019.

32 F. Grijalva Hernández, V. Peña Caballero, P. A. López-Pérez, and R. Aguilar López. Estimation of plasmid concentration in batch culture of *Escherichia coli* dh5 via simple state observer. *Chemical Papers*, 72:2589–2598, 2018.

33 P. A. López-Pérez and R. Aguilar López. Dynamic nonlinear feedback for temperature control of continuous stirred reactor with complex behavior. *Journal of Applied Research and Technology*, 7(2):202–217, 2009.

34 Pushpakant and S. K. Jha. Comparative study of different classical and modern control techniques for the position control of sophisticated mechatronic system. *Procedia Computer Science*, 93:1038–1045, 2016.

35 J. Drgoňa, M. Klaučo, F. Janeček, and M. Kvasnica. Optimal control of a laboratory binary distillation column via regionless explicit MPC. *Computers & Chemical Engineering*, 96:139–148, 2017.

36 J. Holaza, M. Klaučo, J. Drgoňa, J. Oravec, M. Kvasnica, and M. Fikar. MPC-based reference governor control of a continuous stirred-tank reactor. *Computers & Chemical Engineering*, 108:289–299, 2018.

37 P. A. López-Pérez, M. I. Neria González, and R. Aguilar López. Cadmium concentration stabilization in a continuous sulfate reducing bioreactor via sulfide concentration control. *Chemical Papers*, 67:326–335, 2013.

38 R. Matušů, B. Şenol, and L. Pekař. Robust PI control of interval plants with gain and phase margin specifications: Application to a continuous stirred tank reactor. *IEEE Access*, 8:145372–145380, 2020.

39 H. I. Velázquez Sánchez and J. C. Figueroa Estrada. Uncertainty observer-based I/O linearizing control for the regulation of a continuous wastewater bioreactor for Cd removal. *International Journal of Mathematics, Game Theory, and Algebra*, 26(4):259–281, 2017.

40 B. R. Reddy, K. T. Ram, and S. Pranavanand. Adaptive pi controller design and deployment for chemical reactor. In *Proceedings of the 2020 International Conference on Smart Electronics and Communication (ICOSEC)*, pages 1210–1215. IEEE, 2020.

41 S. R. Mahapatro, B. Subudhi, and S. Ghosh. Design and experimental realization of a robust decentralized PI controller for a coupled tank system. *ISA Transactions*, 89:158–168, 2019.

42 R. Villafuerte Segura, B. A. Itzá Ortiz, P. A. López-Pérez, and E. Alvarado Santos. Mathematical model with time-delay and delayed controller for a bioreactor. *Mathematical Methods in the Applied Sciences*, 46(1):248–266, 2023.

43 S. Manabe, Y. C. Kim, S. Manabe, and Y. C. Kim. Advanced controller design: case studies. *Coefficient Diagram Method for Control System Design*, 133–166, 2021.

44 A. E. Rodriguez Mata, Y. Bustos Terrones, V. Gonzalez Huitrón, P. A. Lopéz-Peréz, O. Hernández González, and L. E. Amabilis Sosa. A fractional high-gain nonlinear observer design—application for rivers environmental monitoring model. *Mathematical and Computational Applications*, 25(3):44, 2020.

45 M. Schwenzer, M. Ay, T. Bergs, and D. Abel. Review on model predictive control: an engineering perspective. *The International Journal of Advanced Manufacturing Technology*, 117(5):1327–1349, 2021.

46 J. Drewnowski. Advanced supervisory control system implemented at full-scale WWTP – a case study of optimization and energy balance improvement. *Water*, 11(6):1218, 2019.

47 P. A. Lopez-Perez, M. I. Neria Gonzalez, and R. Aguilar López. Nonlinear controller design with application to a continuous bioreactor. *Theoretical Foundations of Chemical Engineering*, 47(5):585–592, 2013.

48 D. Krishnamoorthy and S. Skogestad. Real-time optimization as a feedback control problem – a review. *Computers & Chemical Engineering*, 161:107723, 2022.

49 B. Srinivasan. Real-time optimization of dynamic systems using multiple units. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 17(13):1183–1193, 2007.

50 J. O. Trierweiler. Real-time optimization of industrial processes. In *Encyclopedia of Systems and Control*, pages 1827–1836. London: Springer, 2021.

51 C. Rieger, R. Boring, B. Johnson, and T. McJunkin. *Resilient Control Architectures and Power Systems*. Hoboken, NJ: John Wiley & Sons, 2022.

52 P. A. López-Pérez, M. I. Neria González, and R. Aguilar López. A mathematical model for cadmium removal using a sulfate reducing bacterium: *Desulfovibrio alaskensis* 6SR. *International Journal of Environmental Research*, 7:501–512, 05 2013.

53 R. Aguilar López, J. Mata Machuca, R. Martinez Guerra, and P. A. López-Pérez. Uniformly bounded error estimator for bioprocess with unstructured cell growth models. *Chemical Product and Process Modeling*, 4(5), 2009.

54 P. A. López-Pérez, H. Puebla, H. I. Velázquez Sánchez, and R. Aguilar López. Comparison tools for parametric identification of kinetic model for ethanol production using evolutionary optimization approach. *International Journal of Chemical Reactor Engineering*, 14(6):1201–1209, 2016.

55 D. J. Hernández Melchor, P. A. López-Pérez, S. Carrillo Vargas, A. Alberto Murrieta, E. González Gómez, and B. Camacho Pérez. Experimental and kinetic study for lead removal via photosynthetic consortia using genetic algorithms to parameter estimation. *Environmental Science and Pollution Research*, 25:21286–21295, 2018.

56 T. Romero Cortes, J. A. Cuervo Parra, V. J. Robles Olvera, E. Rangel Cortes, and P. A. López-Pérez. Experimental and kinetic production of ethanol using mucilage juice residues from cocoa processing. *International Journal of Chemical Reactor Engineering*, 16(11):20170262, 2018.

57 D. J. Hernández Melchor, B. Camacho Pérez, E. Ríos Leal, J. Alarcón Bonilla, and P. A. López-Pérez. Modelling and multi-objective optimization for simulation of hydrogen production using a photosynthetic consortium. *International Journal of Chemical Reactor Engineering*, 18(7):20200019, 2020.

58 V. Peña Caballero, R. Aguilar López, P. A. López-Pérez, and M. I. Neria González. Reduction of Cr (VI) utilizing biogenic sulfide: an experimental and mathematical modeling approach. *Desalination and Water Treatment*, 57(28):13056–13065, 2016.

59 P. A. López-Pérez, M. I. Neria González, M. R. Pérez García, and R. Aguilar Lopez. Concentrations monitoring via software sensor for bioreactors under model parametric uncertainty: application to cadmium removal in an anaerobic process. *Alexandria Engineering Journal*, 55(2):1893–1902, 2016.

60 P. A. López-Pérez, F. A. Cuevas Ortiz, R. V. Gómez Acata, and R. Aguilar López. Improving bioethanol production via nonlinear controller with noisy measurements. *Chemical Engineering Communications*, 202(11):1438–1445, 2015.

61 W. Jongeneel and E. Moulay. *Topological Obstructions to Stability and Stabilization: History, Recent Advances and Open Problems*. Berlin: Springer Nature, 2023.

62 K H. Khalil. *Nonlinear systems 2/e*. Upper Saddle River, NJ: Prentice-Hall Inc., 1996. ISBN 0132280248.

63 L. Leithold. *El cálculo*. Oxford: Oxford University Press, 1982.

64 A. Poznyak, A. Polyakov, and V. Azhmyakov. *Attractive Ellipsoids in Robust Control*. Springer, 2014.

65 A. Genocchi. *Differential Calculus and Principles of Integral Calculus*, volume 1. Bocca, 1884.

66 R. Isermann and M. Münchhof. *Identification of Dynamic Systems: An Introduction with Applications*. Springer Science & Business Media, 2010.

67 F. Rodríguez Rubio and M. J. Sánchez López. *Control Adaptativo y Robusto*, volume 9. Universidad de Sevilla, 1996.

68 P. E. Wellstead and M. B. Zarrop. *Self-Tuning Systems: Control and Signal Processing*. Wiley, 1991. ISBN 9780471928836.

69 K. J. Åström and B. Wittenmark. *Adaptive Control*. Dover Books on Electrical Engineering. Dover Publications, 2008. ISBN 9780486462783.

70 D. E Kirk. *Optimal Control Theory: An Introduction*. Courier Corporation, 2004.

71 N. F. Santos Sánchez, R. Salas Coronado, O. J. Santos Sánchez, H. Romero, and E. Garrido Aranda. On the effects of the temperature control at the performance of a

dehydration process: energy optimization and nutrients retention. *The International Journal of Advanced Manufacturing Technology*, 86(9):3157–3171, 2016.

72 N. Cervantes Escorcia. Control clásico avanzado para un proceso de temperatura., 2017. URL http://dgsa.uaeh.edu.mx:8080/bibliotecadigital/handle/231104/2620.

73 J. B. He, Q. G. Wang, and T. H. Lee. PI/PID controller tuning via LQR approach. *Chemical Engineering Science*, 55(13):2429–2439, 2000.

74 J. E. Marshall. *Control of Time-Delay Systems*. P. Peregrinus, 1979. ISBN 9780906048122. https://typeset.io/journals/iee-control-engineering-series-11hcn7lm

75 K. Ogata. *Modern Control Engineering* (*Instrumentation and Controls Series*). Prentice Hall, 2010. ISBN 9780136156734. URL https://books.google.com.mx/books?id=Wu5GpNAelzkC.

76 R. Vinter. Optimal control and Pontryagin's Maximum Principle. In *Encyclopedia of Systems and Control*, pages 1578–1584. Springer, 2021.

77 R. Luus. Application of dynamic programming to singular optimal control problems. In *Proceedings of the 1990 American Control Conference*, pages 2932–2937. IEEE, 1990.

78 R. Paulen, M. Fikar, and M. A. Latifi. Dynamic optimization of a polymerization reactor. In *Proceedings of the 18th Mediterranean Conference on Control and Automation, MED'10*, pages 733–738. IEEE, 2010.

79 M. J. Mardanov and Y. A. Sharifov. Pontryagin's maximum principle for the optimal control problems with multipoint boundary conditions. In *Abstract and Applied Analysis*, page 428042. Wiley Online Library, 2015.

80 J. S. Meditch. The Pontryagin Maximum Principle and some of its applications. In *Advances in Control Systems*, volume 1, pages 55–74. Elsevier, 1964.

81 L. Wang, H. Wu, and J. Cao. An observer-based topology identification and synchronization in finite time for fractional singularly perturbed complex networks via dynamic event-triggered control. *Neural Processing Letters*, 56(3):1–24, 2024.

82 M. Diehl. Optimization algorithms for model predictive control. In *Encyclopedia of Systems and Control*, pages 1619–1626. Springer, 2021.

83 V. Ramya and B. Palaniappan. Embedded system for hazardous gas detection and alerting. *International Journal of Distributed and Parallel Systems (IJDPS)*, 3(3):287–300, 2012.

84 T. A. Henzinger and J. Sifakis. The discipline of embedded systems design. *Computer*, 40(10):32–40, 2007.

85 J. S. Valdez Martínez, J. Y. Rumbo Morales, K. Santiago Santiago, H. R. Azcaray Rivera, and A. M. Beltrán Escobar. Sistema teleoperado de relajación podal basado en sistemas embebidos de bajo costo. *Revista Internacional de Investigación e Innovación Tecnológica*, 6(36):0–0, 2019.

86 I. Sommerville. *Ingeniería del software*. Pearson Educación, 2005.

87 Mathworks. What is MATLAB?, 2023. URL https://www.mathworks.com/discovery/what-is-matlab.html.

88 National Instruments. What is NI LabVIEW?, 2023. URL https://www.ni.com/es/shop/labview.html.

89 National Instruments. LabVIEW tutorial, 2023. URL https://learn.ni.com/learn/article/labview-tutorial.