

José Luis Guzmán, Ramon Costa-Castelló,  
Manuel Berenguel, Sebastián Dormido

# Automatic Control with Interactive Tools

– Monograph –

March 2, 2022

Springer



**Part I**  
**About interactivity and main objectives of**  
**the book**

Many engineering curricula have only an introductory course in automatic control, which usually takes one semester. It contains a large number of new concepts to be assimilated by students, who generally find it very difficult to a thorough understanding of the subject. This also poses a challenge for teachers.

One objective of this text is to promote the continued study and to encourage individual student work through a set of interactive materials, which are useful for both teacher and students. Rather than the concepts to be introduced, the book has focused on how to present each of them. Once the ideas that are relevant have been identified, consideration has been given to how to visualize and formulate them through example-based learning.

The methodology of study proposed in this book also allows students to solve exercises and provide the solution in the form of short videos, so that the teacher can check whether the students are simply “playing” with the tool looking for a solution or whether they are understanding the concept through the use of the tool, as the main aim is to promote “what if” analysis. We agree with the recent survey [95] in the fact that “while heuristics are useful and an easy way in, we must encourage students to learn the value of systematic and rigorous analysis which is more generalizable”.

The role of this new interactive learning experience in automatic control curricula is twofold [44]:

- To provide a new method for generating learning material that allows introducing the concepts of control systems engineering through interactive applications.
- To provide a new opportunity to innovate in laboratory work where students can analyze, design and modify control engineering systems through interactive tools.

In Chapter 1 a conceptual diagram of the book is explained helping as guide on how-to-use the book by the readers.

# Chapter 1

## Introduction

### Introduction

In recent years, active participation of students in their own learning process has been sought, changing the culture from that of “information receivers” to that of “knowledge seekers”, an ideal scenario in which teachers and students share the exciting adventure of discovery. From this point of view, teaching can mean two completely different things. Firstly, it can simply mean presenting information, so that when a class is taught on a certain topic, it can be said that it has been taught, regardless of whether anyone has learned. The second meaning of teaching is to help someone to learn. According to this meaning, which the authors of this text personally accept, when a concept is taught and the students have not learned, we can conclude that the students have not been taught. The traditional approach of “teaching a course” implicitly uses the first meaning. A syllabus is prepared, making explicit the topics that are planned and are to be covered, the topics are presented in class and for doing this task teachers earn their salaries. It does not matter how many students learn; if the syllabus has been explained, the work has been done. The alternative approach is sometimes called outcomes-based education. Instead of defining a course by simply writing a syllabus, it attempts to specify in as much detail as possible the knowledge, skills, attitudes and competencies that students should acquire by the end of the course.

In the *teaching-learning* process, teachers should present and explain the specified knowledge, provide practical exercises, give timely feedback throughout the course, and offer aids and models for the attitudes that are considered important for students to adopt. Students must use all means at their disposal to meet the learning goals and acquire the competencies and skills related to the subject under study. Perhaps, the most direct way to analyze whether the learning process has been successful is to observe how students do something that demonstrates their knowledge or understanding (give a few “reference points” or “benchmarks” against which they can verify their understanding). The more explicit these objectives are, especially

those requiring high levels of critical and creative thinking, the more likely it is that students will achieve them.

*Learning styles* are the different ways that students characteristically adopt to process new information. Students operate with a panoply of strategies in learning situations. Some prefer to deal with specific information (facts, observations or experimental data) and others are more comfortable with abstract concepts and mathematical models. Some retain more visual information (figures, diagrams, graphs) than strictly verbal information (spoken or written words), and some progress better in the opposite way. Most science and engineering students learn in an active way and gain their highest degree of assimilation when they do something (solve problems or exchange ideas with others), preferring courses based on exercises, case studies and laboratories (which are notable exceptions today), to those that follow a master class format. They are uncomfortable if they cannot see connections between what they are learning and the “real world”. They can be considered “visual learners”.

## 1.1 What is meant by visualization

The term *visualization* is employed to describe the process of producing or using geometric or graphical representations of control concepts, principles or problems, whether represented by hand or generated by computer. Science, engineering and in a somewhat more limited sense, mathematics, are enjoying a renaissance of interest in visualization. To some extent, this revival is being driven by technological developments. Computer graphics are expanding the scope and potential of visualization in all fields. Visualization transforms the symbolic into the geometric, allowing simulations and calculations to be observed in a much clearer way and without requiring highly specialized computer skills. Visualization offers a method of making visible the invisible.

One of the reasons why the teaching of automatic control has a certain technical complication is because the concepts that control experts consider “intuitive”, are not so for students. The reason is quite simple. Intuition is a global resonance in the brain, which is a function of each individual’s cognitive structure, which in turn is dependent on one’s prior experience. There is no reason at all to assume that the newcomer to a field will have the same intuitions as the expert, even when considering seemingly simple visual situations. Automatic control has elements that are spatial, elements that are arithmetic or algebraic, elements that are verbal or programmatic, elements that are logical or didactic, and elements that are intuitive or even counter-intuitive.

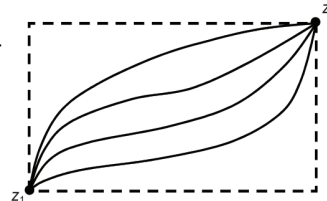
Our use of the term *visualization* differs somewhat from that used in everyday language and psychology, where the meaning of visualization is closer to its fundamental meaning of forming a *mental image*. Vision is not visualization; seeing is not necessarily understanding. The kind of visualization in which we are interested in automatic control is not an end in itself, but a means to an end which is its learning.

Visualization in control is the process of forming images and diagrams (mentally, with pencil and paper or with the aid of technology) and using them effectively for the discovery and understanding of the concept to be conveyed and learned. It provides depth and meaning to understanding, and it serves as a reliable guide to problem solving and inspires creative discovery, but without isolating itself from other forms of numerical or symbolic representation.

Visualization also has its distinctive disadvantages. The problem is that images can often suggest theorems or conjectures that are false, so the strengths and weaknesses of visual imagery need to be considered and weighed against the notions of intuition and rigour, which should never be opposed but rather complementary. From a formal and theoretical point of view, in control theory the “proof of a theorem” is the last stage of the abstraction process and synthesizes knowledge in a particular field. However, before a proof exists, there must be an idea of what the theorem is trying to prove or what theorems might be true. This exploratory phase of abstract thinking about the objects over which control operates can benefit from correct visualization. It is not accidental that when we think we understand something, we say “oh, I see!” The following anecdote attributed to Norbert Wiener, the father of cybernetics, illustrates better than anything this fact. Wiener was teaching at Massachusetts Institute of Technology (MIT) developing a complicated mathematical proof, with the blackboard full of formulas, when he suddenly stopped and stared, for a long time, at the last mathematical expression he had written down. All his students, in stony silence, thought that their teacher was at a dead end. Wiener, however, without saying anything, went to a corner of the blackboard that was still free of his formulae and drew a few figures that his broad shoulders hid from his students. He suddenly changed his face and with a sense of relief, erased those enigmatic figures, returned to the point of the proof where he had stopped and continued flawlessly and smoothly to the end.

A good example of the value of visualization is the famous Cauchy theorem in complex analysis, which is essential in the proof of the Nyquist stability criterion. This theorem states that the integral of an analytical function around a closed curve with no singularities inside is zero. When Cauchy stated the first version of this theorem, he thought of a complex number  $z = x + jy$  analytically in terms of its real and imaginary parts. By analogy with the real case, he defined the contour integral  $\int_{z_1}^{z_2} f(z) dz$  between two points  $z_1$  and  $z_2$  along a curve whose real and imaginary parts are both either monotonic increasing or decreasing.

As a formal generalization of the real case, this restriction of the type of curve is natural. However, if we open our eyes and look at a picture, we see that such graphs (for increasing  $x$  and  $y$ ) are a restricted set of curves that are contained in a rectangle with opposite vertices in  $z_1$  and  $z_2$ .



Cauchy had to visualize the situation for a more general curve in the complex plane in order to formulate his theorem in the form in which is known today. If great

mathematicians need to think “visually”, why keep such a thought process away from students?

## 1.2 The role of visualization in automatic control teaching and learning

Visualization, understood as the way of paying explicit attention to graphic representations that allow explaining abstract contents, is highly developed in the field of computer science, where it is possible to find objects in three dimensions with excellent visual quality.

The fundamentals, ideas, concepts and procedures of automatic control present a great wealth of visual contents that supports an intuitive geometric representation. Clear examples are the time response, the frequency response diagrams (Bode, Nyquist and Nichols diagrams), the poles and zeros plane ( $s$ -plane), the root locus, the phase plane, etc. Their use is very useful in the presentation and handling of the related concepts and procedures, as well as their manipulation in a graphical and interactive way for solving problems. Experts have visual images that act as intuitive mechanisms to understand the fundamentals of their discipline and that at the same time are of great value and effectiveness in the development of their creative work. From these images, they are able to relate and project in a very versatile way a huge amount of facts that allow them to select, in a natural way and without apparent effort, which is the most effective line of attack to solve the complex problems they face. This way of acting, with explicit attention to the possible concrete representations that reveal the abstract relationships of interest to the control engineering specialist, constitutes what we call *visualization in control*. The fact that visualization is a particularly important aspect of the control expert's activity comes naturally when one takes into account the applied mathematics characteristics of control theory. Visualization thus appears as something deeply inherent in the transmission and communication of our knowledge. One of the important tasks that we teachers have is to transmit to our students not only the logical and formal structure of our own discipline, but also, and with much more emphasis, the most intuitive and motivating facets of the subject we are addressing, so facilitating their learning. These are precisely the aspects that are much more difficult to explain and to be assimilated because very often they are in the less conscious substratum of our activity as teachers.

Given the nature of visualization, there will be many highly subjective elements. The ways to visualize and make the ideas of automatic control more approachable and intuitive, in order to implement them in certain situations and apply them to specific problems, largely depend on the mental structure of each person. The degree of visual support certainly varies considerably from one analysis to another, and what may be useful for one person may be a complication for another. However, these differences should not be an impediment to offering those instruments that



have been useful in the work on visualization and without which this task would have been much more difficult, obscure and tedious.

The technical language used by automatic control specialists is a mixture of natural language and language formalized in mathematical terms, a strange jargon consisting of elements of natural language, more or less esoteric words and mathematical and logical symbols. In this context, explicit or not so explicit references are made to scientific conventions that have been established over time and that are interspersed with visual and intuitive connotations.

### 1.3 What is meant by interactivity

There are two essential aspects in the study of dynamical systems that today's computers allow to address and to which much attention has not been paid. One is the visualization referred to in the previous sections and the other is the interactivity of the user with the dynamical system, in its different forms of representation, through the visualization itself. Interactivity is a concept based on the well-known cause-effect relationship. It is about providing the dynamical system with powerful mechanisms so that the user can answer any imaginable question of the types "what would happen if ...?" or "could you do that ...?" simply by acting on the visual elements that constitute the graphical expression that represents it. More often than one might think, ideas in engineering arise from specific and visual situations. The perception of reality is essentially visual and it is not surprising that this support is so deeply rooted in our everyday way of working.

Symbolic treatments, visual diagrams and other forms of imaginative processes are often used to acquire an intuition of what, in its most formal aspect, has a much more abstract content. It is precisely here where it makes sense to have powerful and user-friendly environments that, using the current state of the art computer technology, allow the creation of interactive dynamic simulations with high visual content. The idea of changing properties and immediately being able to see the effects of the changes is very helpful for both analysis and design. The dynamics of the changes provide additional information that is not available in a static graph.

Bricklin and Frankston made a pioneering effort in the use of interactivity in the late 1970s when they developed VisiCalc<sup>1</sup>, which was based on the metaphor of an electronic sheet. VisiCalc contained a grid of rows and columns of data accessible for financial calculations. Its implementation in an Apple II was one of the reasons why personal computers started to be used in companies. VisiCalc transformed the spread sheet from a calculation application to a modeling and optimization tool. Excel (which is derived from these ideas) is now standard software in all offices and companies. The spreadsheets were easy to implement because they only operated on numbers; however, in automatic control much richer graphical objects are present. The possibility of having them all interconnected so that they can be di-

---

<sup>1</sup> <http://bricklin.com/visicalc.htm>

rectly manipulated and updated interactively and concurrently has great potential to enhance learning. The VisiDyn program was a pioneering implementation in this direction [53]. Another attempt was made by Blomdell [21], who developed a program for an Apple Macintosh computer. His system had some novel features such as a graphical diagram representing the pole-zero structure of a transfer function. It also had an original way of interacting with Bode diagrams. These early tools were very useful, but their implementation required a great deal of effort, which limited their use as educational tools.

Therefore, the goal of interactivity as discussed in this text is to enhance learning, by exploiting the advantages of immediately seeing the effects of changes that can never be shown by images of a static nature, e.g., on a blackboard. The use of interactivity in education provides a wide range of possibilities to both teachers and students. Teachers can make use of interactive presentations where not only the meaning of a certain concept is shown, but also how that concept may be related to others or how it may be influenced by certain external factors (e.g. some inputs modifications). For the teaching staff, interactivity provides a way to assess the main ideas on a given topic and the level of difficulty involved in assimilating these ideas by the students. On the other hand, students can make use of interactive websites or computer-based interactive tools to study theoretical concepts abstracted through interactive objects.

From a software implementation point of view, interactivity can be defined as a *fourth dimension*. In this way, the saying “a picture is worth a thousand words” could be modified to “interactivity is worth a thousand images” [59].

*Interactive tools* can be used to investigate precisely what needs to be done to make students aware of theoretical concepts. In this way, the virtue of simplicity becomes a fact in learning research on the design and use of this type of tools. The main reasons why interactive tools are useful for teaching are [49]:

- They facilitate student-focused learning, allowing choice in the pathways for learning and the rate at which new material is introduced.
- They can address several learning styles and modalities, providing a rich variety of instructional approaches which can teach in most of the ways what students learn best.
- They motivate students’ interaction, experimentation and cooperative learning.
- Students often work together on computer projects as they never did on paper and-pencil projects.
- They facilitate storylines or thematic learning, where a pathway for exploration can easily be woven around a particular concept dynamics.
- They promote the constructivist view of learning.

Interactivity also has drawbacks. Students can try to learn and solve problems simply by manipulating the interactive applications, but without understanding what they are doing. Interactive tools should challenge students by making them relate theory to practice, as interactive objects are abstractions from theoretical concepts. Neither should teachers develop modules of this type without correctly managing the different levels of abstraction (from the idea to be shown using interactivity and

taking into account the student's perspective to the creation of the final product). It can be said that interactivity can be very useful in the teaching-learning process, but it must be used with control.

## 1.4 The role of interactivity in automatic control teaching and learning

The concepts of visualization and interactivity are becoming increasingly relevant in the teaching of automatic control, where the underlying ideas, concepts and methods are very rich in visual content, which can be represented intuitively and geometrically. Calculations and representations include responses in the time and frequency domains, poles and zeros in the complex plane, Bode/Nyquist/Nichols diagrams, phase plane, etc. Often, such magnitudes are strongly related and represent different views of the same reality. The understanding of these relationships is one of the keys to achieving good learning of the basic concepts and allows the student to be able to accurately design automatic control systems [44].

Traditionally, systems design is carried out following an iterative process. Specifications of the problem are not usually used to calculate the value of the system parameters, because there is no explicit formula that relates them directly. This is the reason for dividing each iteration into two phases. The first, known as *synthesis*, consists of calculating the unknown parameters of the system based on a set of design variables, which are related to the specifications. During the second phase, called *analysis*, the performance of the system is evaluated and compared to the specifications. If they do not match, the design variables are modified and a new iteration is performed. It is possible, however, to merge both phases into one, so that when a parameter affecting the design is modified, the analysis results are updated and observed immediately, or vice versa, so that it is possible to visually modify representative elements of the analysis phase and obtain the design parameters needed to achieve these objectives. In this way, the design process becomes truly dynamic and the students perceive the gradient of change in the performance criteria for the elements they are manipulating. This interactive capability makes it much easier to identify compromises that can be achieved.

The computing resources and programming environments available today have enabled the emergence of a new generation of environments for interactive learning of automatic control. These tools are based on objects that support direct graphic manipulation. During these manipulations, objects are updated immediately, so that the relationships between the objects are maintained at all times. Ictools and CCS-demo, developed at the Department of Automatic Control at the University of Lund [5, 75] and Sysquake<sup>®</sup> at the Institute of Automatic Control of the Federal Institute of Technology in Lausanne [91] are, among others [54, 74, 82], excellent examples of this new educational philosophy of automatic control.

For newcomers in the field of automatic control, many of the concepts are not very intuitive when first approached, since their properties are expressed in two

different domains: The time domain and the frequency domain. Transient behavior (e.g. settling time and peak overshoot) and saturation risk are typically analyzed in the time domain; while concepts such as stability, noise rejection and robustness are more easily expressed in the frequency domain. The basic mechanisms that relate them and other phenomena, such as the inclusion of nonlinearities, can be illustrated very effectively through interactivity, as is the case with the applications included with this text.

## 1.5 Objectives and organization of the contents

As pointed out by [99], the process of control system design often follow a step by step procedure:

1. Study the system (process or plant) to be controlled to get initial information about control objectives.
2. Model the system with a degree of detail depending on the final use (simulation, control, optimization). When necessary, simplify the model.
3. Analyze and determine the properties of the resulting model.
4. Decide which variables are to be controlled (controlled outputs).
5. Decide what sensors are required to measure the controlled outputs and disturbances and what actuators have to be used to implement the manipulated variables (and where sensors and actuators will be placed).
6. Select the control configuration.
7. Take decisions on the controller structure to be used.
8. Set performance specifications based on control objectives.
9. Design a controller using the most adequate method according to the selected model, control objectives and control architecture.
10. Analyze the obtained closed-loop system to verify that specifications are fulfilled. If they are not satisfied, modify the type of controller or the specifications.
11. Simulate the resulting controlled closed-loop system.
12. Repeat from step 2, if necessary.
13. Choose hardware and software to implement the controller.
14. Test and validate the control system on the real plant and re-tune the controller online if necessary.

From all these steps there are several that are related to implementation issues in the real plant (5 in part, 13 and 14). With the aim of facilitating the understanding of basic concepts related to the steps in the control system design that do not require the real plant, a set of *interactive tools* has been designed as a support to this book, which are made up of:

- A graphical application, called *interactive tool*, that visualizes each concept and includes numerous interactive objects in order to enhance interaction with the application.

- The application incorporates predefined examples that illustrate a specific phenomenon or particularity of the concept introduced.
- Each application is accompanied by a theoretical summary that presents the concept formally and analytically. It is not intended to replace the textbook. Therefore, the theoretical contents are reduced to the minimum necessary to understand the specific application. It contains references to specific books for users to broaden their knowledge. The package consisting of the interactive tool and the related documentation is what we have called a *learning card* (other authors refer to learning pills).
- The learning cards have been designed to be self-contained, so the summary includes a small manual of the interactive tool describing all its functionalities. Interactive symbols are explained in the Terminology introduction of the book, while common aspects of all interactive tools are explained at the end of this chapter to avoid the text being somewhat repetitive for the reader. We have tried to find a tradeoff so that the understanding of each interactive tool does not require previous analysis of others.
- The documentation of each learning card contains different exercises to be solved using the interactive tools. The purpose of these examples is to initially guide the student's steps to reduce the time required to learn each of the concepts and to emphasize the important points. The objectives follow a *learning-by-discovering* approach and aim to combine *qualitative knowledge* obtained through interactivity with *quantitative knowledge* obtained through the different examples. Once the exercises have been completed, the users will be able to create their own scenarios and self-assess the knowledge acquired. Each part of the book (open-loop analysis and closed-loop analysis and design) includes solved problems that make use of the different tools analyzed. Moreover, the website of the book will be updated with new exercises and solutions.

As a complement to the previous information, a series of data related to the way in which the tools have been developed are briefly included below:

- Sysquake<sup>®</sup> [91] has been used as a development tool due to its programming language, very close to the *de facto standard*, MATLAB<sup>®</sup>. Interactivity is fully integrated into the language. This provides a high-level development framework that is very convenient for non-professional programmers.
- The applications contain a large number of common components, such as the input signal or the pole-zero editors, which have been included in a library used by all applications. This simplifies the development and maintenance of the toolkit. These components are described at the end of this chapter. We encourage users to read it.
- The applications are lightweight and distributed via executables that do not require pre-installation for Mac OS and Windows.

The contents have been designed to cover a basic course on classical control techniques based on external description (transfer function), with a strong focus on those having graphical visualisation features that can be interacted with. In the preface we have provided the arguments to focus the text on the use of transfer functions which,

on the other hand, are the natural description in process control [97]. A brief introduction to state-space analysis and design (constrained to 2-dimensional systems) can be found on the website of the book. Figure 1.1 shows a conceptual diagram of the book, showing all the topics covered, that are those most important for any engineer to know about dynamical systems, such as dynamic behavior, physical and empirical modeling, computer simulation, feedback and stability, basic control approaches, and analysis and design tools. The applications cover the following fields:

- **From nonlinear physical models to linear models.** This chapter explains the concept of dynamical system, the linearization of the mathematical representations of nonlinear systems around an operating point and the analysis of their behavior. Five processes are described that will be used as examples throughout the text: a tank with water and discharge to the atmosphere (two versions are included: constant area and variable area), a ball & beam system, an inverted pendulum on a cart system (simplified version), and a direct current (DC) motor. As only the tank level systems are open-loop stable, they will have three related interactive tools each one: one devoted to linearizing the nonlinear representative model, another one to relate the model parameters to the representative transfer function, and the last one to feedback control of the fluid level. The other processes have one interactive tool described in Chapter 8 dedicated to closed-loop control, although the summary of theory and exercises will be included in the appropriate chapters.
- **Time Response.** This group contains seven interactive tools designed to illustrate the time response (step response) of continuous linear time invariant (LTI) systems. An interactive tool has also been incorporated to explain the fitting of linear models from experimental data.
- **Frequency Response.** This group includes eight interactive tools aimed at illustrating the concept of frequency response and its analysis for the same systems treated in the previous group. The main graphical representations associated with this concept are also introduced, as well as an interactive tool for fitting models from Bode diagrams obtained experimentally.
- **Relationship of model parameters to physical models.** This chapter illustrates the relationship between physical models and the main elements of models used in controller design (transfer functions), such as poles, zeros and time response. The parameters of the nonlinear physical models discussed in Chapter 2 are related to those of the linearized models.
- **Closed-loop systems and stability.** The main techniques for analyzing the stability of closed-loop linear systems are introduced: Root locus, Nyquist criterion and phase and gain margins. An interactive tool is also included to illustrate the limitations imposed by the time delay in a feedback loop.
- **Control system design.** This chapter brings together the analysis of steady-state errors in systems with unit feedback and the main controller design techniques used in introductory courses, both in the time and frequency domains, covered in eleven tools. It covers aspects of analytical design of simple PID-type controllers for systems without delay (basically pole cancellation and pole assignment), heuristic or approximate designs for systems with delay (Ziegler-Nichols

and lambda method), the inclusion of filters in the reference to decouple the reference tracking problem and disturbance rejection, the classical frequency design of phase-lead and phase-lag controllers in the Bode diagram, and also the analytical design of these kind of compensators based on typical frequency response specifications. Finally, a more complex tool aimed at explaining how to design controllers by the *loop shaping* technique is included. Although to a certain extent this tool visually breaks with the simplicity of the previous ones, being the last controller design tool, it encompasses practically all the concepts analyzed in the previous ones. We believe that the reader will find this last tool very interesting for combining all the concepts acquired in this chapter.

- **Control of physical systems.** The design of controllers and the simulation of the closed-loop system are studied for the processes explained in Chapter 2. Although the interactive tools include selected control techniques, the reader can solve exercises on these processes using the different interactive tools included in the text.

It should again be noted that the text is not restricted and oriented to sequential reading by chapters, as is usual in a conventional book. The users can locate the concept they want to analyze in the table of contents and go to the self-contained card where the concept is treated. In fact, the book can be used as a dictionary of basic control concepts, under the support of interactive tools. The conceptual map showed in Figure 1.1 also helps to locate the main concepts (state-space modeling and design are not treated but can be found on the web page of the book).

Chapters 2 and 5 cover the main concepts related to Modeling approaches indicated in the conceptual diagram, including linearization and the explanation of the use of steady-state models for evaluating the operating point. Linear models are dealt with in detail in Chapter 3 (time domain) and 4 (frequency domain). The main tools for representing transfer functions are also treated in these chapters (time response plots, pole-zero map and frequency-domain diagrams). Some hints about experimental identification are included in the last interactive tool of both chapters. Basic closed-loop stability analysis tools are introduced in Chapter 6, mainly root locus, Nyquist criterion and stability margins, also considering the negative effects of time delays in the feedback loop. Chapter 7 deals with steady-state performance with unit feedback and closed-loop transient specifications, both in time and frequency domains. Control design is based on 1- and 2-degree-of-freedom (2-DoF) control schema for PID controllers and lead-lag controllers using both classical methods and loop shaping, as already explained. Finally, Chapter 8 relates to closed-loop simulation and analysis using the five representative processes introduced in Chapter 2.

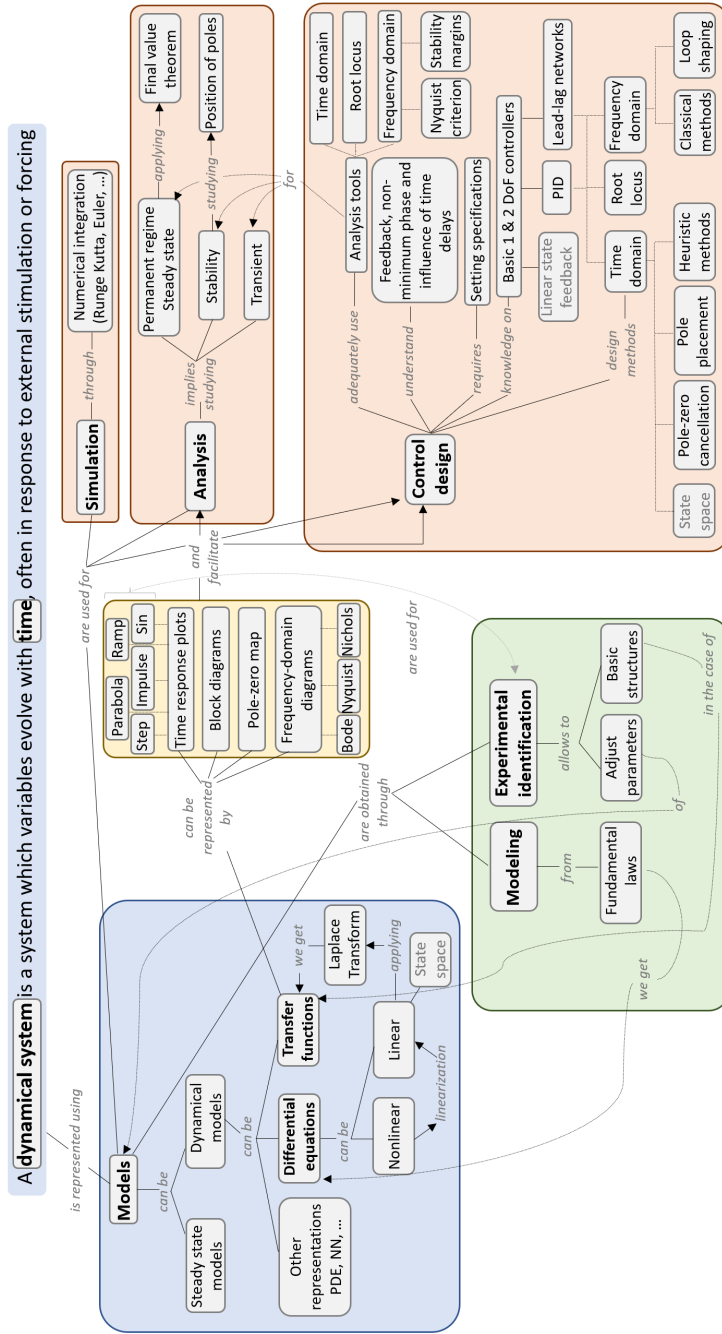



Fig. 1.1 Conceptual map of the book



## 1.6 Components and structures of the interactive tools

### 1.6.1 *Distribution of elements and interactive objects in the tools*

This section briefly explains the types of interactive objects, graphic and text elements that can be found in the interactive tools, as well as their spatial distribution. Notice that the size of the tools is scalable to a maximum. Interactive objects are characterized by the fact that when the mouse pointer is placed over them, their form of representation changes to a hand , indicating that this element is interactive. For instance, placing the mouse pointer over a certain point of the time or frequency response curves, activates a label that shows the properties associated with that point.

In addition to the interactive elements, other non-interactive elements will be used, such as symbolic representations of formulas or transfer functions, legends on graphs, etc., which help to understand the concepts that are intended to be analyzed with the tools. In any case, the values associated with these elements can be modified using the available interactive elements.

#### 1.6.1.1 Distribution of graphic windows in the tools

Most interactive tools used in this text have been designed following certain patterns for the distribution of objects on the screen (patterns which in turn have many similarities between them). Table 1.1 shows the main patterns used for the different interactive tools, which are described in the following:

**Pattern 1. Interactive tools related to physical systems.** The organization in this pattern by quadrants is:

Interactive parameters and equations	Time response of the system
Process representation	Input signal to the system

Table 1.1 includes as example of this pattern the tool `tank_level.lin`.

**Pattern 2. Interactive tools related to the analysis of the time response, design in time domain, and root locus.** The organization of the interactive tools associated with this pattern follow the same structure (with small differences in the size of the objects), being shown as an example in the Table 1.1 those corresponding to the tools `t_second_order` (Pattern 2.a) and `PI_lambda` (Pattern 2.b):

Interactive parameters, characteristics and equations	Time response of the system
Pole-zero diagram	Input signal to the system

**Pattern 3. Interactive tools related to the analysis of the frequency response (without including the open-loop time response).** The layout of the interactive

tools associated with this pattern (the one corresponding to the interactive tool `f.second_order` is included in the Table 1.1) follows the same structure:

Interactive parameters, characteristics and equations	Bode magnitude plot
Pole-zero diagram	Bode phase plot

This group has a uniform distribution in which the main working graph (Bode plots) occupies the entire right side of the tool, so its distribution is optimal for the user to analyze the associated concepts.

**Pattern 4. Interactive tools for analysis of the frequency response (including the open-loop time response), stability analysis, and design in the frequency domain.** Table 1.1 presents as examples of this group the tools `f.concept` and `f.design_lead`. The layout of all these tools is the same:

Interactive parameters, characteristics and equations	Selectable frequency diagrams
Pole-zero diagram	Time response (hideable)

The fundamental elements in this group are the representations in the frequency domain, but it may be of interest for the user to also analyze the open-loop (Pattern 4.a) or closed-loop (Pattern 4.b) system response, these representations occupying a different area on the screen (smaller in the case of open-loop ones). By default, when the tool is started, the time response curve will appear in the lower-right quadrant in all displays, but there will be an entry in the Options menu (explained later) that allows to hide them to increase the size of the frequency plot (Bode, Nyquist, Nichols or all four at once). In pattern 4.b., the option in the Options menu to show the time response enables (on user demand) the evolution of the system output or input.

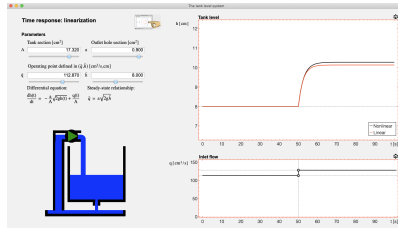
**Pattern 5. Interactive tools devoted to explaining the concepts associated with PID controllers and the influence of time delays.** The layout of all the interactive tools associated with this group (Table 1.1 includes as example the tool `PID.concept`) is the same, the main difference being the substitution of the pole-zero diagram by the Nyquist plot.

Interactive parameters, characteristics and equations	Time response of the system
Nyquist plot	Input signal to the system

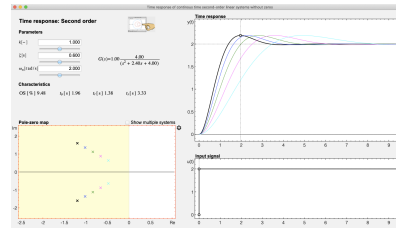
As commented before, the interactive tool `loop.shaping` used to explain the loop shaping design approach is more complex, as this method is difficult to explain with a basic layout. It will be explained in more detail in the corresponding section.

### 1.6.1.2 Color codes

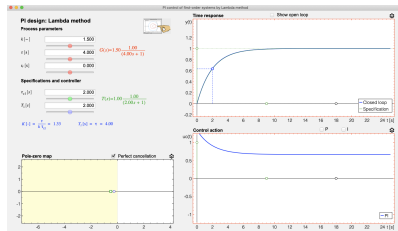
The predefined order of the color code in the library is red, blue, green, magenta and cyan.



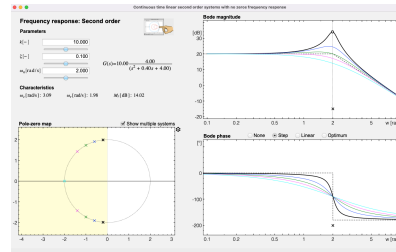
Pattern 1. Cards related to physical systems



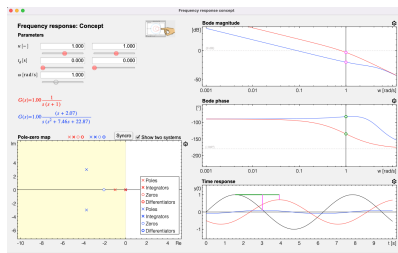
Pattern 2.a. Open-loop analysis



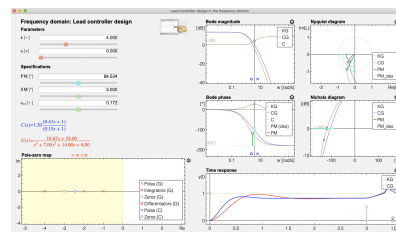
Pattern 2.b. Closed-loop analysis



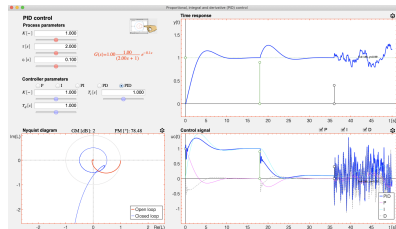
Pattern 3. Cards related to frequency response analysis



Pattern 4.a. Open-loop analysis



Pattern 4.b. Closed-loop responses



Pattern 5. Cards related to PID controllers

Table 1.1 Typical examples of object layout in interactive tools

In the time and frequency response analysis interactive tools, by default, the color used for the representation of text and graphics associated with a given system is black. When the representation of two systems is carried out simultaneously, the representative colors are red color and blue color. When multiple systems are displayed, the predefined order is used and black color is used to highlight the active system, i.e. the one that represents the current parameter values. In the case of the interactive tools treating the controllers' design the color codes used are:

- Red to represent the transfer function of the open-loop system, its associated time response (dashed line) and its frequency response (solid line).
- Blue to represent the transfer function of the controller, the closed-loop time response (solid line), and the frequency response of the loop transfer function in frequency-domain plots (thin line).
- Green to represent the frequency response of the controller, the closed-loop transfer function defining the specifications and its associated time response (solid line). The reference (setpoint) is drawn in dashed line.
- Magenta is employed to represent other lines, for instance, the root locus.

These colors are also used to represent some performance and stability metrics such as phase lags, relative stability margins, etc. (details are provided in each card).


### 1.6.1.3 Menu bar

The menu bar contains three menus in all tools:

- **Session:** This menu includes the so-called **Reset** option, allowing to restart the tool to its initial settings. Moreover, there are two interesting features: **Save session** and **Restore session**, providing the possibility of saving and restoring scenarios.
- **Options:** A drop-down menu with different options appears on the top toolbar. An attempt has been made to standardize as much as possible the information represented in this menu, the most common options being those presented in Table 1.2, which are explained in detail in each card.
- **Info:** Includes information about the tools in two sub-menus: **About** and **support**.





The last interactive tool in Chapter 7 does not follow this basic layout, as it practically summarizes the contents of the book, requiring more specific options which are explained in the corresponding card.

### 1.6.1.4 Interactive elements associated with the selection of values and options

- **Textboxes** , allow changing the value of a parameter by entering it directly via the keyboard. By clicking the box with the mouse, the text is highlighted. After introducing the new text, the *enter* key must be pressed and the new value will be refreshed throughout the tool. The point (not the comma) is used as a decimal separator.


Options		Tools including the option
Gain effect		t_first_order
Time constant effect		f_first_order
Damping factor effect	Constant imaginary part	t_second_order
Natural frequency effect	Constant real part	f_second_order
$P1(s)=k/s^n$	$P_1(s) = \frac{k}{s^n}$	t_generic*
$P2(s)=k/(Ts+1)$ ( $P_2, P_4, P_{11}$ )	$P_2(s) = \frac{k}{(\tau s + 1)}$	f_concept*
$P3(s)=k(bs+1)/s^n$ ( $P_3, P_{11}, P_{12}$ )	$P_3(s) = \frac{k(\beta s + 1)}{s^n}$	f_generic*
$P4(s)=k(bs+1)/(Ts+1)$	$P_4(s) = \frac{k(\beta s + 1)}{(\tau s + 1)}$	f_nonminimum_phase*
$P5(s)=k/(s(Ts+1))$ ( $P_8, P_2 P_6, P_2 P_{11}, P_6 P_{12}$ )	$P_5(s) = \frac{k}{s(\tau s + 1)}$	root_locus*
$P6(s)=k(bs+1)/(s(Ts+1))$	$P_6(s) = \frac{k(\beta s + 1)}{s(\tau s + 1)}$	Nyquist_criterion**
$P7(s)=k(bs+1)/((T1s+1)(T2s+1))$	$P_7(s) = \frac{k(\beta s + 1)}{(\tau_1 s + 1)(\tau_2 s + 1)}$	stability_margins
$P8(s)=k/(s(T1s+1)(T2s+1))$	$P_8(s) = \frac{k}{s(\tau_1 s + 1)(\tau_2 s + 1)}$	steady_state
$P9(s)=k/((T1s+1)(T2s+1)(T3s+1))$	$P_9(s) = \frac{k}{(\tau_1 s + 1)(\tau_2 s + 1)(\tau_3 s + 1)}$	PID_concept*
$P10(s)=k wn^2/(s(s^2+2 z wn s+wn^2))$	$P_{10}(s) = \frac{k\omega_n^2}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)}$	f_design_lag*
$P11(s)=k wn^2/((Ts+1)(s^2+2 z wn s+wn^2))$	$P_{11}(s) = \frac{k\omega_n^2}{(\tau s + 1)(s^2 + 2\zeta\omega_n s + \omega_n^2)}$	f_design_lead*
$P12(s)=k wn^2 (bs+1)/(s^2+2 z wn s+wn^2)$	$P_{12}(s) = \frac{k\omega_n^2(\beta s + 1)}{(s^2 + 2\zeta\omega_n s + \omega_n^2)}$	f_design_lead_lag*
<b>Note:</b> The tools with (*) only incorporate as examples some of the $P_i(s)$ . Those with (**) include more examples		
Introduce plant (NUM,DEN)		t_generic
Introduce plant (ZPK)		t_dominance
Allow defining examples of plants in MATLAB format		f_concept
(NUM,DEN): Numerator and denominator in polynomial format		f_generic
$G(s) = \frac{b_m s^m + \dots + b_1 s + b_0}{a_n s^n + \dots + a_1 s + a_0} \rightarrow [b_m, \dots, b_1, b_0], [a_n, \dots, a_1, a_0]$		f_nonminimum_phase root_locus
ZPK: Zero, pole, gain formats		stability_margins
$G(s) = \kappa \frac{(s - z_1)(s - z_2)}{(s - p_1)(s - p_2)(s - p_3)} \rightarrow [z1;z2],[p1;p2;p3],k$		steady_state
Complex poles or zeros are introduced in the format [x+yj;-x-yj]		
Bode diagram		f_concept
Nyquist diagram		f_nonminimum_phase
Nichols diagram		lead_lag_concept
Complete view		f_design_lag
Graphical representations of the frequency response		f_design_lead f_design_lead_lag
Experiment $i, i = 1, \dots, 4$		t_model_fitting
Analyze dominance	Dominance example 1	t_dominance
Dominance ratio	Dominance example 2	
Lag compensator	Several lead compensators	lead_lag_concept
Lead compensator	Lag vs lead compensators	f_design_lead_lag
Several lag compensators		
<b>Note:</b> lead_lag_concept also includes an option to switch the Nichols diagram to a time response plot.		

**Table 1.2** Options menu

- **Sliders or scrollbars**  are used to change the value of a certain parameter. Moving to the right increases the value of the parameter and moving to the left decreases its value.
- **Radio buttons** , are used to activate a certain option (alternative or exclusive to the one being used) by clicking on them.
- **Check boxes** , are used to activate a certain option by clicking on them.
- **Activation buttons** , serve to perform a certain action.

### 1.6.1.5 Interactive elements associated with graphical representations

#### Common to all representations

- **Scaling in graphs** : By clicking the gearwheel symbol, two possibilities arise: Locked scale, allowing the user to modify the scale settings on the abscissa and ordinate axes of the corresponding plot  $(x_{min}, x_{max}, y_{min}, y_{max})$ , both introducing the values or moving the slides; Variable scale, where the scale is automatically updated.
- **Interactive lines**: In the plots of physical systems, the boundary lines that compose the objects have a greater thickness (–), which usually indicates that they are interactive objects (in addition to the change in cursor representation) and that therefore the dimensions of the graphic object can be modified directly by clicking and dragging the mouse over them<sup>2</sup>.

#### Associated with the representation in the $s$ -plane and the geometrical root locus

- **Poles**: The poles of a dynamical system are described using a cross:  $\times$ .
- **Zeros**: The zeros of a dynamical system are described using a circle:  $\circ$ .
- **Integrators**: They are symbolized by a bold cross:  $\mathbf{\times}$ .
- **Differentiator**: They are represented using a bold circle:  $\mathbf{\circ}$ .
- **Closed-loop poles**: They are represented by a blue triangle  $\triangle$ .
- **Desired closed-loop poles** (obtained from the specifications): They are represented using a green square ( $\square$ ).
- **Centroid** of root locus: The intersection of the linear asymptotes is represented by the cyan diamond symbol  $\diamond$ .
- **Entry or exit points of the real axis of the root locus**, where the root locus leaves or reaches the real axis (breakaway point, saddle points, bifurcation points) are represented by a magenta square symbol  $\square$ .

#### Associated with time representation

- **Characteristic point of the response**: There may exist characteristic points of the response to which an interactive behavior can be associated, such as the time

<sup>2</sup> *Clicking and dragging* is a way to move certain objects on the screen. To move an object, place the mouse cursor over it, press and hold down the left mouse button, then move the mouse while still holding down the left mouse button. When you have *dragged* the object to the location you want, let go of the mouse button.

constant in first-order systems, overshoot in second-order systems, etc. These points are usually represented by a circle (○) that allows to modify the shape of the response dragging it.

- **Exogenous input** (⊕): Dashed vertical and horizontal lines with circles may be displayed in the time response of systems (output and input plots), that allow modifying the amplitude of the step used as input in open-loop or that of the closed-loop reference (○···), as well as the step-shaped load disturbance (○). These signals are measured on the axes of the plots on which they are represented, and their associated values (activation instant and amplitude) are displayed in the lower-left corner of the tool. The variance of the noise signal is also represented with the same symbol (○). When dragged vertically, the value of said variance is shown in the lower-left corner of the tool, the axes of the graphs not being valid in this case because they would provide a noise value without physical sense. The formula implemented in the tool for the noise generator in the form of a difference equation is given by [57]:

$$x(t_i) = 0.7x(t_{i-1}) + \sqrt{0.5\sigma^2} \text{randn},$$

where subscript  $i$  represents discrete time instants,  $\sigma$  is the noise variance and “randn” is a random number generator normally distributed.

When the circles determining the amplitude of the signal are separated from the time axis, the circles which remain on said axis are used to change the time at which the disturbance or noise is introduced.

#### Associated with frequency representation

- **Characteristic point of the response:** There may exist characteristic points of the response to which an interactive behavior can be associated.
  - In the case of low-frequency gain, the interactive element that appears is a dash line (–) associated with the log-magnitude axis.
  - The frequency linked to the position of a pole in the Bode plots (corner or break frequency of the representative asymptotes) is usually represented by a cross (×) placed on the frequency axis, while that associated with a zero uses a circle (○).
  - Some characteristic points of the response, such as a resonant peak or a relative stability margin, can also be represented. To make them interactive, a black circle is used (○), linked to that point. These characteristic points are those that facilitate the modification of the shape of the frequency response in the corresponding graphs.
  - The maximum or minimum magnitude and phase design points of phase-lead or phase-lag controllers are symbolized by a green square (□). The design frequency and the safety margin is drawn using a green and white diamond respectively (◇, ◇).

### ***1.6.2 A remark on numerical implementation***

The time responses in closed-loop have been implemented using a discrete numerical approach, as it is the way to consider the presence of time delays and to allow introducing load disturbances and noise at any time. A tradeoff solution between numerical performance and avoiding loss of interactivity has been found using a sampling time of 0.05 seconds in the discrete-time approximation, as a lower value has a negative impact in interactivity. This means that in systems with very fast responses (e.g. using derivative action in closed-loop control), numerical errors may appear in the time responses in special cases. Where this may occur, a clarification has been added in the associated text.