# Under Control: A Control Theory Introduction for Computer Scientists

Quentin Guilloteau[4] Sophie Cerf[2*] Raphaël Bleuse[1] Bogdan Robu[3] Eric Rutten[1]

***Abstract*—This hands-on tutorial aims at introducing the basis of Control Theory, applied to the runtime management of computing systems, through a practical example. This tutorial is composed of two parts. In the first part, we present the motivation and classical tools of Control Theory: formulation, modeling, and controllers. During the second part, the participants will experiment with the concepts presented in the first part through *interactive* computational documents, via a Jupyter Notebook. Attendees will need a computer with an internet connection and a "recent" web browser, and only basic programming knowledge. The tutorial is available at the following URL: https://control-for-computing.gitlabpages.inria.fr/tutorial/**

***Index Terms*—autonomic computing, control of computing systems, control theory, feedback management**

## I. General Information

This hands-on tutorial is an introduction, addressed to beginners in control theory. Its duration may vary between participants, between 3 and 4 hours. It is aimed for students, academics, and industry. Table I summarizes the general information about this tutorial.

This tutorial does not have any strong prerequisite for the participants. On the technical side, participants are required to have access to a computer with an internet connection and a "recent" browser (Firefox 90+, Chromium 89+) which supports WebAssembly. On the knowledge side, participants need basic programming knowledge. Some mathematical analysis knowledge can be useful to fully understand the section explaining the principles behind control theory guarantees, however it is not necessary to understand the key principles.

## II. Control for Computing

Computing systems are getting more and more complex. The software stacks are growing, and are executed on top of complex hardware. The behavior of applications is thus becoming extremely difficult to predict or model correctly in the design phase. We need regulation to provide Quality-of-Service guarantees.

[1]: Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, Grenoble, F-38000, France; [2]: Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France; [3]: Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, Grenoble, F-38000, France; [4]: University of Basel, Switzerland. [*]: Presenter.

| Characteristics | Details |
|---|---|
| Content level | Beginner |
| Duration | 3 hours, up to 4 hours |
| Target Audience | Industry & Academia |
| Tutorial format | Hands-on |
| Prerequisites | Computer with internet |

TABLE I: General Information

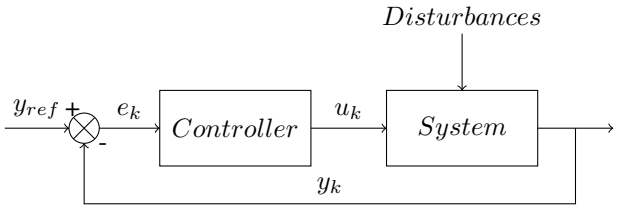


Fig. 1: Control Loop representation. $u_k$ is the control action, $y_k$ the sensed measure, $y_{ref}$ its desired reference value, and $e_k$ is the corresponding error.

One approach consists in building a theoretical model of the system, and design a complex configuration algorithm based on this model. Such an approach is limited by the quality of the model, and its ability to capture all runtime events. Modelling all the possible cases, and behaviors is tedious and error-prone.

A different approach to regulate the behavior of a computing system is to periodically take measurements of relevant metrics, and adapt the configuration based on these measurements. This runtime approach is called closed-loop, and this is the interest of the Autonomic Computing community [2].

The Autonomic Computing formulation relies on the *MAPE-K* loop, called after the different phases of the loop: **M**onitor, **A**nalyze, **P**lan, **E**xecute, and **K**nowledge. There are several ways to perform this runtime adaptation. One can use ad-hoc solutions based on arbitrary rules or IA for example [4]. These solutions, however, do not often come with guarantees on the closed-loop system. On the contrary, the Control Theory field has been applying math-proven methods to closed-loop physical systems for decades. But the application of Control Theory to computing systems is only quite recent. Figure 1 represents a feedback loop of Control Theory.

As an example of application of Control Theoryto HPC, a regulation problem arises where the objective is to harvest unused computing resources at cluster level. It is approached by injecting low-priority jobs in order to maximize the utilization of CPUs. However, this can cause bottlenecks on e.g., memory access. Therefore, the more specific objective is to control the load of the distributed file-system of a computing cluster by adapting the number of low-priority jobs, using various controllers, as well as coordination with the scheduler [1].

## III. Motivation and impact of this tutorial

### A. Motivation

Despite its promising results and guarantees, the use of Control Theory as a tool for autonomic systems is only limited, possibly due to very few interactions between the Control Theory and Computer Science fields, in universities and industry. This tutorial aims at bringing accessible Control Theory concepts to computer scientists. While a theoretical approach can be given on the topic of control for computing systems [3], we advocate that a complementary hands-on, interactive experience with computational notebooks is a suitable tool for computer scientists to favor the actual usage of Control Theory in their systems.

### B. Impact

This tutorial aims at introducing Control Theory, a method particularly suited for autonomic management of dynamic systems. It explains Control Theory to computer scientists in a simple and (almost) mathematical-free way. It allows understanding the principles of the guarantees that Control Theory provides. After this tutorial, attendees will be able to apply basic Control Theory tools and practices to their own systems, but also will have the knowledge and understanding to discuss and collaborate with experts in Control Theory.

## IV. Planned Content

The tutorial begins with a 15 to 20-minute introductory presentation by the speaker, followed by the hands-on session.

At the end of the tutorial, the attendees should have:

- understood the motivations, methodology, and basic tools (PID Controller) of Control Theory
- performed an "identification" of a system
- designed a Proportional (P) Controller for this system, and understood the limits of the controller
- designed a Proportional-Integral-Derivative (PID) Controller for the same system
- apprehend advanced control techniques
- formalize a control closed-loop of their own system.

## V. Previous Sessions

As of June 2024, there have been **7** sessions of this tutorial, for a total number of attendees around **110**. The tutorial has been given to a number of events such as research seminars and conferences in parallel and distributed computing. The participants of ACSOS, familiar with the challenges of autonomic computing, are a novel audience that could benefit from a hands-on introduction to Control Theory.

## VI. Authors Biographies

### A. Quentin Guilloteau

Quentin Guilloteau holds a Ph.D. in Computer Science from Univ. Grenoble Alpes in France (2023) on the topic of applying Control Theory approaches to the regulation of High Performance Computing systems. He is currently a Post-doc at the University of Basel in Switzerland, working on multi-level scheduling. He is also interested in reproducible research and autonomic computing in HPC.

### B. Sophie Cerf

Sophie Cerf is a Research Scientist in the Spirals team at Inria center of the University of Lille. She got her Ph.D. from the Univ. Grenoble Alpes in 2019. Her research interests are Control Theory for Distributed Systems. She focuses her research on sustainable and social computing.

### C. Raphaël Bleuse

Raphaël Bleuse is an Associate Professor at Univ. Grenoble Alpes, France since 2019. He received is Ph.D. in 2017 from Univ. Grenoble Alpes. His research focuses on the combination of control theory with scheduling theory, aiming at designing pragmatic solutions for the allocation of resources on HPC systems and alike. His research ranges from modeling computing systems, designing controllers and scheduling algorithms to conducting reproducible experiments by simulation or on real hardware.

### D. Bogdan Robu

Bogdan Robu is an Associate Professor in the GIPSA-lab research laboratory of the Univ. Grenoble Alpes. He received his Ph.D. in 2010 from the University of Toulouse. His research focus is on applying Control Theory techniques (discrete, continuous, hybrid) to Distributed Systems for the runtime dynamic management of learning algorithms and neural networks, Cloud/Fog software, parallel computing systems as well as IoT systems.

### E. Eric Rutten

Éric Rutten (Ph.D. 90, Habil. 99 at U. Rennes, France) is researcher at Inria in Grenoble, France, heading the Ctrl-A team. After working on embedded, real-time and robotic systems, contributing methods and tools to support computing for control systems, he then reversed the perspective by considering control for computing systems, exploring the potentials of Control Theory (discrete and continuous) in the runtime management of self-adaptive Cloud/Edge or High Performance Computing systems, with objectives of self-(re)configuration, self-optimization, self-healing, and self-protection.

## References


[1] Q. Guilloteau. *Control-based runtime management of HPC systems with support for reproducible experiments*. Theses, Université Grenoble Alpes, Dec. 2023. URL: https://hal.science/tel-04389290.

[2] J. O. Kephart and D. M. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.

[3] A. V. Papadopoulos. Designing self-adaptive software systems with control theory: an overview. In *2022 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*, 2022. DOI: 10.1109/ACSOSC56246.2022.00027.

[4] B. Porter, R. R. Filho, and P. Dean. A survey of methodology in self-adaptive systems research. In *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*, pages 168–177, 2020. DOI: 10.1109/ACSOS49614.2020.00039.