Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 48-29 (2015) 265–276

Computing systems and the network as a control education arena

Alberto Leva *

* *Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy (e-mail: alberto.leva@polimi.it)*

Abstract: Computers and the network are widely used and studied as a means to provide control education. However, they also offer a wide variety of problems to teach control. This paper concentrates on the second aspect, presenting some didactic activities that are interesting for both control and computer science/engineering students. After motivating the proposed pedagogy from a cultural point of view, also as a means to foster the necessary convergence of the two mentioned communities, some examples of activities are presented and discussed. The purpose is twofold. On one hand, the reader can see how many concepts – ranging from control structures through cyber-physical systems to process/control co-design and more – can be exemplified and experimented with, both in simulation and on real systems that virtually any student already possesses. On the other hand, and maybe most important, the usefulness of those concepts for the *design* – not just the control – of computing systems, can be appreciated, and constitutes a very promising research and engineering field. To this end, a brief sketch of future perspectives concludes the paper.

Keywords: Control education, computing systems, networks, feedback control, simulation.

1. INTRODUCTION

Over the last years, the impressive development of computing systems and the network has deeply changed control education. The literature – of which a minimal and largely incomplete set of examples is here quoted – reports a variety of experiences, ranging from remote/virtual laboratories (Casini et al., 2003; Martín-Villalba et al., 2012; Jouaneh and Palm, 2013; Soares et al., 2014) and networks among them (Vargas et al., 2011) to computer- and/or Internet- and web-based learning tools (Michau et al., 2001; Gillet et al., 2005; Gonzalez et al., 2012), interactive tools (Johansson et al., 1998; Guzmán et al., 2014), Massive Open Online Courses or MOOCs (Egerstedt, 2013; de la Croix and Egerstedt, 2014), and much more.

In all those experiences, however, the relationship between computing systems and control is essentially unidirectional: computers and the network are used to provide control education and the necessary material, in the broadest sense of the term, but control is not applied to computing systems and the network. The research presented in this paper reverses the perspective, viewing computers and the network as a source of problems to teach control.

A first motivation for doing this is that computing systems allow to devise several nice didactic experiences, that can be carried out with a minimal equipment. But there is much more to the proposed perspective reversal.

In fact, a key point of this work is that the presented activities are suited – and in the author's opinion should be proposed – to both control and computer science students, provided the latter audience know just a bare minimum of systems theory.

For a computer scientist, a system- and control-theoretical mentality is very useful, but to the best of the author's knowledge, this mentality is not fostered as strongly as it should by the typical computer science curriculum. This has two undesired consequences. First, a huge number of management and design problems in that domain, that are control ones in nature, are not treated as such in the computer science literature. Second, and maybe worse, several attempts to add control to computing systems and the Internet, are carried out viewing the control theory as hardly anything more than a source of algorithms, which often leads to inadequate solutions, to a questionable use of techniques, and in general to a poor comprehension of what the systems and control theory really is—i.e., to put it simply for this introduction, a *corpus* of theoretical and technological knowledge made consistent by a systemic *forma mentis*, that has to be put to work at almost any stage of any design.

For a control scientist/engineer, conversely, not taking the operation of the computing system he/she uses as just a matter of fact, can open interesting perspectives. It is already well known that several computing-related issues that are often treated just as implementation-related accidentals, can in fact be modelled and considered in the design of a computer-based control system. However, there is more to this idea. In some cases, the same issues can be handled by devoting to them *specific* controllers, independent of what use (control-oriented or not, for example) is to be made of the computing system. If control is the goal, this allows to design the rest of the system on top of a computing machinery that gives improved operational certainties, to the advantage of simplicity and assessment possibilities. However, doing so requires to know and model some internals of computers and networks, that are normally out of scope for the typical control curriculum.

To enhance the integration and the cooperation between the control and the computer science/engineering communities, the author believes that education plays a fundamental role. The

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2015.11.247

rest of this paper is devoted to first elaborate on this idea from a general standpoint, which is done in Section 2. Then, Sections 3 through 5 present some examples of didactic activities to put the previously reported considerations to work. Each of these sections starts from an informal problem statement, then formalises the same problem with a control-centric approach, models the controlled system, synthesises the controller, shows an application example, and then discusses some possible student activities, ending with a round-up of the applied concepts and of the pedagogical outcome.

In each activity section, references are given to papers where the reader can find the details here omitted. It is worth noticing that these are *recent research* papers. This fact may look peculiar at a first glance, as normally the subject of control education activities is established well enough to be found in textbooks. However, the same fact witnesses that the control-centric approach here shown is really beneficial for the present and future development of computing systems, not to say of great importance for an effective treatment of their steadily increasing complexity. Comparisons with non control-centric approaches to the addressed problems are not reported here, but can be found in the papers just mentioned, so as to provide support to the statement on a "correct use of control" made above.

Finally, Section 6 ends the paper with some concluding remarks, and sketches out future perspectives.

2. FILLING A CULTURAL GAP

We start from the control community side. Basically, and brutalising the matter a bit for space reasons, a control engineer views computers as a means to realise a control strategy, and takes their operation as a matter of fact. There have been attempts to co-design a controller and the provisioning of the computational resources to run it (Xia and Sun, 2008; Al-Areqi et al., 2015), and the matter has been studied also in the real-time community (Marti et al., 2002). However many other aspects, notably the possibility of re-designing network protocols – rediscussing the entire stack if needed – to make delays more deterministic, have received far less attention.

Moreover, the management of computational resources – when addressed – is almost invariantly viewed as a part of the *overall* problem, and designed jointly with the control part in the strict sense of the term. Adopting the cyber-physical paradigm, we could thus say that the physical part is entirely outside the computing system.

Let us conversely adopt a *physical-cyber-physical* point of view, and dream – but not that much – for a while. Software (the cyber part) stands between an *outside* and an *inside* physics, and is split (roughly speaking) in two layers. The outside-looking one controls the outside physics (the plant) and relies on certainties on the available resources. Providing those certainties – maybe with convenient confidence bounds, but this is not the key point – is the task of the inside-looking layer, that is designed to ensure them by governing the inside physics (CPU, memory, network interface, clock, and so forth) *independently* of the reasons why the mentioned certainties are necessary for the outside-looking layer. And most important, also the inside-looking layer is conceived and realised as a set of controllers, based on dynamic models, and formally assessed.

No doubt this is an interesting *scenario* to investigate, but doing so requires control people to know about the internals of computing systems and networks more than in general (*absit injuria*) they do.

We now move to the computer science/engineering side. Many attempts were made to apply control in order to make systems operate correctly in the presence of unforeseen external conditions; just a couple of examples are Hellerstein et al. (2004); Brun et al. (2009). However, the idea is almost invariantly to take an *already functional* system, model its dynamics (frequently based on measured data) and then close loops *around it*: in this respect, it is illuminating to notice that in the survey by Patikirikorala et al. (2012), one of the taxonomy axes (RQ2, page 34) is "what are the methods used to model the dynamics of the software system". As noticed in Leva et al. (2013a), this is a quite partial use of control. It would be advisable to move, whenever possible, from the mere control to a *control-based design* of systems.

Again, this has to do with evidencing an *inside* physics, but pushes the idea to identifying what is *real* physics, and what is added by the software layers of the existing system. Anticipating the task scheduling case of Section 3, which is paradigmatic in this respect, the real, core physics is extremely simple: the CPU time used by a task after its k-th activation it that up to the $(k-1)$-th, plus the time allotted by the scheduler at the k-th, plus a disturbance (discussed later on in due course). A discrete integrator; that's it.

The physics added by a functioning scheduler, should one decide to close a loop around it, may on the contrary include priorities, multilevel queues, deadlines, ant their management. To close a loop around a functional system, one has thus in general to describe as dynamic systems things that were conceived as algorithms, and as such, complex and articulated combinations of various formalisms (discrete-event systems, aggregates of queue networks, and more) are required. Controlling such models is correspondingly complex, not easy to implement in real systems, potentially inefficient, and cumbersome to assess. Indeed, better stop modelling at the boundaries of *real* physics: the rest must be control.

The cost of re-designing part of a system can be significant, of course. But when such a re-design is possible and convenient, the payback in terms of efficiency and simplicity is relevant as well. The research papers quoted in the following sections provide some examples, and a discussion on the importance of "taking the right measurements" and "using the right actuators", which is another way to indicate that controllers need positioning correctly with respect to the controlled system physics, can be found in the challenge paper by Papadopoulos (2015).

Pursuing this research path up to its technological outcomes cannot be done without the competences of computer science/engineering people, but symmetrically to the discussion above, this requires them to know about systems and control not just as a source of algorithms, but as a discipline in the widest sense of the term—i.e., *absit injuria* again, more than they most often do.

Summarising, we have evidenced a cultural gap, and the need for the two mentioned communities to converge. In the opinion of the author, correctly targeted and structured education activities, suitable to target – possibly even jointly – the two