Politecnico di Milano
Facoltà di Ingegneria Industriale e dell'Informazione
Corso di Laurea in Ingegneria dell'Automazione

Event-Based Realisation of Industrial Controllers: Stability and Robustness

Realizzazione a Eventi di Controllori Industriali: Stabilità e Robustezza

Relatore: Prof. Alberto LEVA

Tesi di Laurea di:
Giovanni Maria GERLI Matr. 783218

Anno Accademico 2012-2013

Contents

List of Figures

Abstract

This thesis deals with the stability and robustness analysis of an Event-Based (EB) realisation of an industrial controller. After a short introduction in which the main concepts are presented, the Event-Based Control paradigm is explained, and the main technological and application-related motivations that support the development of event-based control systems are given. A brief literature review is in order to present the main topics on which research has been concentrating to date, and the progresses achieved. Motivations for EB control are then reconsidered and further explained; EB realisations are then compared to their counterparts in continuous time and in discrete time; this evidences some open problems in EB control theory and allows to trace some possible research directions.

The first topic to be addressed in this thesis is the stability analysis. The mathematical preliminaries are presented, and a set of hypotheses is given which lead the EB realisation to be very close to its fixed-rate counterpart, still preserving all of the advantages illustrated so far. After some preliminary results, the EB realisation reveals its (induced) switching nature, allowing to prove a sufficient stability criterion under arbitrary switching; the analysis is concluded with a corollary and some remarks.

The robustness analysis is the second topic addressed in this thesis; in particular, we will study the robustness of the stability with respect to parametric uncertainties in the process model. A short description of the uncertainty (depicting its causes, how it is modeled and how –and even if– can be counteracted) is given; robustness is given in terms of regions in which a controller tuned on the nominal process ensures stability even with a perturbed process. The analysis is concluded with the choice of a triggering rule for the event-generation mechanism.

To strengthen the theoretical results, simulation examples are provided that prove that the EB loop is stable and has some advantages over its fixed-rate counterpart.

In the end, besides the already mentioned research directions, some possible improvements to this work are proposed.

Sommario

Questa tesi tratta l'analisi di stabilità e di robustezza di un controllore industriale realizzato a eventi. Dopo una breve introduzione in cui vengono presentati i concetti principali, viene spiegato il paradigma del Controllo a Eventi, e vengono fornite le principali motivazioni tecnologiche e applicative a supporto dello sviluppo di sistemi di controllo a eventi. Una breve ricerca in letteratura è d'obbligo per presentare i principali argomenti sui quali ad oggi si è concentrata la ricerca, e i progressi ottenuti sinora. Le motivazioni del controllo a eventi vengono quindi riprese e ulteriormente spiegate; la realizzazione a eventi è poi messa a confronto con le controparti a tempo continuo e a tempo discreto; questo mette in evidenza alcuni problemi aperti nella teoria del controllo a eventi e consente di tracciare alcune possibile direzioni di ricerca.

Il primo argomento ad essere affrontato in questa tesi è l'analisi di stabilità. Vengono presentati i preliminari matematici, e viene fornito un insieme di ipotesi che porta la realizzazione a eventi ad essere molto simile alla sua controparte a passo fisso, pur mantenendo tutti i vantaggi illustrati sinora. Dopo alcuni risultati preliminari, la realizzazione a eventi rivela la sua natura switching (indotta), permettendo di dimostrare un criterio sufficiente di stabilità sotto switching arbitrario; l'analisi si conclude con un corollario e alcune osservazioni.

L'analisi di robustezza è il secondo argomento affrontato in questa tesi; in particolare, studieremo la robustezza della stabilità rispetto a incertezze parametriche nel modello del processo. Viene data una breve descrizione dell'incertezza (le sue cause, come viene modellata e come –e se– può essere contrastata); la robustezza è espressa in termini di regioni entro le quali un controllore tarato sul processo nominale assicura la stabilità anche con un processo perturbato. L'analisi si conclude con la scelta di una regola di triggering per il meccanismo di generazione degli eventi.

Per rinforzare i risultati teorici, vengono forniti esempi di simulazioni che provano come l'anello di controllo a eventi sia stabile e possieda alcuni vantaggi rispetto alla sua controparte a passo fisso.

Infine, oltre alle già menzionate direzioni di ricerca, vengono proposti alcuni possibili miglioramenti a questo lavoro.

Introduction

Nowadays, digital technology is an indispensable component in many control systems. Digital feedback controllers sample, transmit measurements and compute control action periodically and at constant step; in literature, this control paradigm has been labeled in various manners, such as ‘time-triggered’, ‘fixed-rate’ or ‘sampled-data’ control. The major reason for the widespread of this paradigm is the existence of a powerful and well established theory which allows to directly design the controller, both in a linear and a non-linear context. However, the theory of sampled-data systems hides some practical –and not negligible– issues.

Limits of time-triggered control emerge, for example, when considering a networked control system. In such a plant, sensor, controller and actuator are linked via, e.g., field buses, Local Area Networks, ATM networks, and so on; an important parameter for performance is the network load, and thus it is clear that transmitting data packets at a constant step, when the system behaves properly, increases the load and degrades the performances of the control loop.

Periodic control may be undesirable in some situations due to its conservative nature. Periodic control system, indeed, require to select the sampling period before the model is deployed; one has to ensure that this period is adequate against a wide range of uncertainties. As a consequence, sampling period may be chose shorter than necessary; consider, for example, a fixed-rate controller implemented on a CPU running concurrent tasks. Computing a control action that brings no improvement to the process is clearly a waste of computational resources; this (useless) control action, moreover, will be actuated by an appropriate device, increasing its wear without any reasonable profit.

To get over these disadvantages, in the last decade a novel control paradigm has been proposed, the Event-Based (EB) Control, also called ‘Event-Triggered Control’, ‘Asynchronous Control’, ‘Aperiodic Control’. It can be referred as a way to acquire measurements, take decisions and/or apply actions ‘only when needed’, that is, when a significant event has occurred (such as the arrival of a data packet to a node of the network, a measured variable exceeding a prescribed threshold, and so on). EB Control gained popularity specially in relation to the growth of networked wired (and wireless) control systems, which raise the importance of explicitly addressing energy, computation and communication constraints when designing feedback control loops.

This policy is expected to bring some advantages with respect to the Fixed-Rate case, particularly in presence of battery-operated devices (which typically are the sensors) which communicate via a network of some sort. The main advantages could be summarized as follows:

- *Reduction of communications*: As transmissions of the measured variables occur only when necessary, Event-Based control reduces network load, lowering the chances of packet losses and communication delays;

- *Reduction of sensor's battery consumption*: In the majority of cases, a sensor is not fed by the electrical network, but operates on battery power. Lowering the number of transmissions means reducing the sensor energy consumption and thus extending the device's battery life;

- *Reduction of actuator wear*: As control action takes place only when needed instead at a fixed rate, Event-Based control may prolong the actuator's life.

Besides, EB Control is closer to the human nature as a controller, that is: when a human performs manual control, his behavior is event-triggered rather than time-triggered, in the sense that a control action will take place only when the output has deviated 'enough' from the desired set point.

EB Control went in and out of fashion throughout the years. One of the reasons is that it lacks of a unified theory as powerful as the one for the Fixed Rate case; nonetheless, many works have been produced on the matter. Many authors have presented modifications of well-know controller structures (such as PID) to get event-based realisations with advantages in terms of CPU computational load and/or transmission rates. Stability and robust stability criteria, under a wide variety of hypotheses, have been researched; many authors dealt with these topics by exploiting state-feedback control which produces good results but, as we know, it is hardly applicable in practical cases, as the state of a process may be unknown or not (fully) measurable. This problem has been counteracted mainly in two ways: by introducing state observers, or, as other authors do, choosing instead output-feedback control.

There is another main reason why time-triggered control still dominates. From a control point of view, it is frequently assumed that a possible real-time implementation will be able to guarantee a deterministic sampling interval; this is, in fact, not always true. Determinism is in fact not always assured because of a number of implementation issues impossible to discuss here, but typically related either to the "general-purpose" (i.e. not real-time specific) design of the part of the used architecture, or –even if a real-time specific design approach is taken– to the unavoidable detriment to determinism induced by advanced microprocessor features like for example caching and pipelining. No doubt such features exhibit enough positive effects to justify their use (e.g., in terms of computational efficiency), but the mentioned problems often remain.

From a real-time point of view, instead, it is generally assumed that control loops are always periodic, with fixed period and hard deadlines. It should be obvious that event-based control loops are aperiodic; moreover, in a large number of cases, deadlines are soft rather than hard; this prevents from using all of the theory of rate monotonic scheduling and its extensions.

Recently [1] provided a simple (sufficient) stability condition for a process controlled by an event-based realisation of a PID. The theorem presented therein relied on a Zero Order Holder to actuate the control action; as a result, the state matrix of the overall system resulted to be poorly manageable. We conjecture that replacing the Zero Order Holder with an Impulse Holder will simplify the structure of the matrix, thus leading to a new proof of the theorem, under (hopefully) more simple hypotheses. This, in conjunction with a robustness analysis, is the task of the present work; it is important to point out that the main focus of this work is on autotuning, so both the stability condition and the robustness bounds will be formulated in terms that are easily applicable in an autotuning context.

This work is structured as follows:

- Chapter 1 provides a detailed insight on EB control; here it is examined what EB Control is, the application-related motivations of an EB realisation are provided and finally a review of literature works shows the progresses achieved so far and the major research lines on the matter;

- Chapter 2 revisits and further develops the scenario sketched out so far, establishing a methodologically grounded relationship between control synthesis techniques of industrial interest (such as PID autotuning) and EB controller realisation. The chapter is concluded with a short (and surely not exhaustive) list of possible research directions;

- Chapter 3 reports the main result of this work on stability. First, the mathematical framework is established, providing the notation, the preliminary results and the general hypotheses which constitute the context of application of this work. Then, following the choice of the more suitable discretisation step and the construction of the state matrix of the EB system, a simple (sufficient) stability theorem is proved. The Chapter is concluded with a corollary and some remarks;

- Chapter 4 devises the robustness analysis which, for the sake of clarity, is carried out in a restricted scope. Bounds on uncertainty are given in terms of robustness regions. At the end of the Chapter, having given stability and robustness criteria effective under arbitrary switching, a triggering rule is presented;

- Chapter 5 reports some simulation examples, which strengthen the theoretical results obtained so far;

- Chapter 6 summarizes the major contributes of this work, furthermore envisaging some points worthy of future research.

Chapter 1

Background and related work

This chapter provides some background material on event-based control, evidences the application-related motivations for event-based controller realisations, and synthetically describes – by means of a review of related work in the literature – the major research lines on the matter.

1.1 Event-based control

Consider a control loop with a Continuous-Time (CT) process, and suppose that a continuous-time feedback controller was synthesised, that adheres to some desired specifications. To realise that controller digitally, one needs to (i) acquire a sample of the controlled variable, (ii) compute the control signal, and (iii) send that signal to a convenient holder, that finally governs the actuator. To this end, two routes can be followed:

1. perform the actions (i) to (iii) above *periodically* and *synchronously*, which is called here a Fixed-Rate (FR) realisation of the continuous-time controller,

2. or perform the same actions "only when needed" (for the moment, whatever this means), which results in an Event-Based (EB) realisation of the same controller.

The typical and general scheme for an EB control loop is shown in Figure 1.1, where the main elements that can be recognised are outlined below.

1. The *Plant*, that is assumed to be described by a continuous-time dynamic system—in the context of this work, SISO (Single-Input, Single-Output).

2. The *Sensing event generator*, that generalises the idea of "sampler", and is devoted to acquiring a value $y(t_k)$ of the controlled variable $y(t)$ when this is deemed necessary, for example (but in principle not necessarily) because

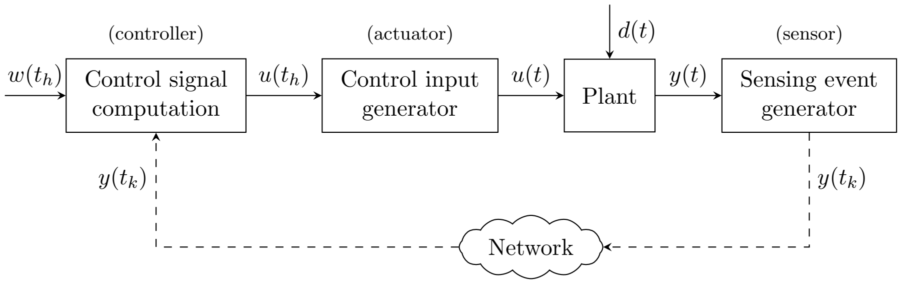

Figure 1.1: Event-based control loop.

the present value of $y(t)$ differs in magnitude from the last acquired one $y(t_{k-1})$ by more than a prescribed threshold.

3. The *Control signal computation*, that generalises the idea of "time - discretised controller", and has the role of computing the control signal to be sent to the control input generator at the instants t_h when this is deemed conveninent, for example because the last received $y(t_k)$ and the presently available reference signal $w(t_h)$ lead to "large enough" a movement of the actuator with respect to its present position. Note that in principle the time indexes counting the sensing and the actuation events may be different and independent, whence the two different subscripts adopted.

4. The *Control input generator*, that generalises the idea of "holder".

5. The *Network*, over which communications take place, and that can therefore introduce time-varying and/or unpredictable delay, and even give rises to communication losses.

As can be seen, acquiring a sample only when deemed necessary results in practice in a time-varying sampling time; thus, no discrete-time model (as commonly intended) is available. Consequently, the FR theory does not hold (although many authors approach to EB control as an extension of the FR case), and thus one needs

1. to (re-)assess stability and robustness criteria, performance guarantees and so forth;

2. and to set up convenient tuning methodologies, or – more practically – to find suitable ways to apply synthesis rules conceived for CT controllers in such a way that the EB realisation does not introduce detrimental effects.

Moreover, the FR theory considers essentially (and as far as the main focus is set on applications, in practice exclusively) the Zero-Order Holder (ZOH) as the input for the actuator; despite almost all the approaches to EB control use a ZOH as input, other kinds of holders, such as the Impulsive Holder (IH), should be investigated.

It is clear, then, that there is a strong need for a system theory for EB control, which should answer three basic questions:

- In which situation should information be transmitted?
- Which information should be transmitted?
- How should the control input be generated?

1.2 Application-related motivations

In a nutshell, as should now be clear, EB control consists of replacing the FR sampler with an *event generator*, that dictates when a new control signal is to be computed based on some *event triggering rule*, in turn requiring some information on the signals of interest.

An EB controller realisation is thus readily viewed a means to reduce the communication among sensors, controller and actuator in a loop, by triggering a communication among said components only after an event has indicated (for example) that the error has trespassed a tolerable bound.

According to the literature, especially concerning applications and technologies, three are the major motivations for using EB control:

1. mitigate the network load,
2. reduce the energy consumed for transmissions,
3. reduce the actuator stress.

Given the increasing capabilities of control networks and field buses, the main technology-related reason seems at present to be the increasing use of wireless sensors, that quite often operate on battery power.

1.3 Brief literature review

EB control is still a matter for research and draws much interest, as shown by works like [2] and papers quoted therein; therefore, the literature offers a wide variety of works on a wide range of topics. It is worth noticing that, as of the event-generating mechanism, the literature distinguishes between *event-triggered* control, where an event is generated at the violation of a certain condition, and *self-triggered* control, when the next event is decided at the previous step.

In [3], the author proposed a modification of the classical PID scheme to obtain an event-based controller which reduced the CPU utilization with only minor degradation on performances; this work has been extended by Durand et al. [4], which aimed at further reducing the number of samples and CPU calculations. In [5], tha authors took both the previous works and found new improvements both in the reduction of sampling/calculations and in performances guarantees.

Regarding stability, [6] proposes a novel event-triggering scheme for a non-linear CT system such that the resulting event-triggered system, controlled via a non-linear state-feedback, is asymptotically stable (provided that the CT system is stabilizable). In [7] instead, given an Input-to-State Stable (ISS) discrete-time system, the event-triggering strategies that stabilizes the system, both in a linear and a non-linear context, are investigated. The results are extend also to the self-triggering context. In [8], given a process controlled via output-feedback and a triggering condition, asymptotic stability by means of a Linear Matrix Inequality (LMI) is ensured. This result is then extended to the self-triggering context and to a state-feedback controller with a state observer. The paper [9] proposed instead an event-based control loop which adopts a state-feedback approach including a model of the CT system, versus which the current plant state is evaluated. The authors prove that the approximation between EB and CT can be made arbitrarily tight by a suitable choice of the threshold parameter of the event generator. In [10] the previous work is extended by investigating the reference tracking properties of the loop; incidentally, the experimental evaluation shows that the loop is robust versus severe model uncertainties. Finally, in [11] the case when the plant state measurement is not available (or, somehow equivalently, a measurement noise is present) is considered; this forces the authors to quit the state-feedback approach in favor of an output-based approach, which, coupled with a state observer, is able to guarantee a stable behavior.

To the author's knowledge, there exist very few works which deal with event-based control robust versus model uncertainties, while there exist works like [12] that study robustness versus network-induced time delays. For an overview of the wide range of application of the EB strategies, the interested reader is referred to [13], which presents EB strategies not only in the same scope illustrated so far, but examines its relevance in other fields, such as the design of communication protocols in multi-agent systems, the development of EB estimation techniques and fault diagnosis. In the end, the paper presents further research directions.

Recently Leva et al. [1] provided a simple (sufficient) stability condition for an EB realisation of a PID. The authors, in their framework, assumed that any event could be triggered only on a time instant that is an integer multiple of a quantum q_s; this happens basically because events are triggered by a sensor which polls the measured variable with step q_s, and it is clear that between two subsequent steps nothing can happen. This framework shows a strong resemblance (at least for the underlying hypotheses) with the one adopted in [14], where this approach was called "Periodic Event-Triggered Control" (or PETC), in opposition to the "Continuous Event-Triggered Control" (CETC), where the measured variable is monitored continuously. With the former strategy, it is possible to reach a satisfying compromise between the FR realisation and the EB one, as the event-transmission has now a periodic nature, without sacrificing the benefits deriving from a reduced rate of transmission. Moreover, this approach is more suited for digital implementation than CETC, which requires dedicated hardware to detect events.

Besides, the paper [1] showed that there exists a close relationship between EB control and switching systems. In a discrete-time context, indeed, if the event-generating mechanism affects the discretisation procedure of the system we get a different model at every event. This suggests that it is possible to study an EB realisation with the theory of the switched systems; a survey on the main problems of switched systems, the progresses achieved and the open problems is available in [15] and in [16], to which the interested reader is referred. Regarding the problem of robust control of switching systems, we can mention [17, 18, 19, 20].

The design of triggering rules (for EB realisations) and switching rules (for switching systems) is also a widely discussed topic. Though the best result would be assessing stability under arbitrary switching/regardless of the triggering rule, there exist in literature many results which also give specific triggering rule to guarantee asymptotic stability and good error tracking properties.

The already mentioned paper [6], for example, presents a theorem which, given a (not quadratic) Lyapunov-based triggering rule, ensures the asymptotic stability of the system and moreover bounds from below the inter-event time. The authors of the paper [21] claim to have extended this result, leading to a theorem which guarantees stability and a minimum dwell time under more general conditions, which allow to consider more general types of Lyapunov functions, such as quadratic ones.

In [22], two theorems are proposed which guarantee asymptotic stability by means of LMIs, given a performance requirement and a switching rule expressed in terms of this requirement. The authors consider both a case in which the state information is available (thus allowing for state-feedback control) and a case in which it is not (leading to output-feedback control); in the end, the robust design problem is addressed.

The paper [14] is mainly focused on quadratic triggering conditions, showing that some others triggering conditions (based, for example, on state error or input error) may be written in a quadratic form; in [20], a theorem is stated whose proof leads to the construction of a switching signal by concatenation. However, as marked out therein, this concatenation does not provide any hint to build the aforementioned signal.

Another topic of great interest, in switching systems, is that of stability under dwell time. In general, given a switched system, asymptotic stability of each subsystem is not sufficient to prove the asymptotic stability of the switched system, as the movement of the state could "inflate" (in some norm) before decreasing; this is typically due, for example, to a couple of complex conjugate eigenvalues. However, provided that each subsystem does not switch before a prescribed amount of time, stability could be ensured. There exist many results on the relationship between the dynamic matrix spectral radius and minimum dwell time; for example, it is known that, if $D_{i,\tau} = e^{A_i \tau}$ has a spectral radius ≥ 1, then τ is its minimum dwell time. Basing on this, [23] proposes an algorithm that finds the minimum dwell time by picking τ in an interval which is progressively narrowed, until the chosen τ makes the spectral radius greater than one.

This review does not claim to be exhaustive nor to cover all of the possible topics on EB control; rather, it covers the issues that will be treated in the remainder of this work.

Chapter 2

Research context and problems

This chapter re-visits the EB control *scenario* as sketched out so far, adopting however the viewpoint that characterises the research path to which this thesis belongs, that is, establishing a methodologically grounded relationship between control synthesis techniques of high industrial interest, such as PID autotuning, and EB controller realisations.

In extreme synthesis, the main idea could be stated as follows. It is well known how an FR realisation impacts the behavior of a digitally realised controller and modifies the stability, performance and robustness properties of the loop with respect to the “ideal” ones as stemming from the same loop described as an entirely CT system. As a consequence, techniques exist to select the sole additional parameter needed for an FR realisation, i.e., the sampling time. We would like to establish an analogous framework for the realisation of the same controllers in EB form.

2.1 Motivations for EB control, revisited

Virtually any controller starts out as the result of some continuous-time (CT) design. The “traditional” realisation path then relies on discretisation to obtain an FR digital controller, the evolution of which occurs periodically and is triggered by a sampler. Assuming that the sampling of the controlled variable and the actuation (or “holding”) of the control signal are synchronous, a strong theory exists – at least in the linear context, that however covers most applications – to analyse and assess the so obtained (hybrid) control loop.

Limiting thus the scope to the first two reasons, an open problem is that the physical structure of an EB control loop is not standardised: for example, the controller might reside in the same hardware device as the sensor or the actuator, or be separated from both. According to the literature, specifically that on “networked control”, a particularly common and interesting case is when the sensor is separated from the other components. This is in fact quite reasonable in general, as both the actuator device and any possible “central” computational unit seldom experience power shortage issues. As such, although different *scenarii* can be envisaged, the most interesting one, and the sole addressed herein, is that of an energy-critical sensor connected via a network – the load of which may be of concern – to a less energy-sensitive control and actuation equipment. As a further simplification, we here disregard the possibility of faulty or missed communications, which seems a reasonable hypotheses in applications like process control (contrary e.g. to mobile ones).

As a further consequence, in the addressed *scenario* it seems quite natural to have that sensor dictate when transmissions need to occur. In this respect, the two motivations above are apparently intertwined, yet each one preserves some peculiarities. For example, network load is related to the number of transmissions, both from the sensor and towards it if this is possible, while (critical) energy consumption also depends on the amount of computational load delegated to the sensor.

Remarks. Although the matter is more technology-oriented than methodological, it would be nice to define the cost of a control realisation in terms of which and how many operations it requires on the part of energy- and/or communication-critical components. This would imply, as an initial and incomplete list, the number of transmissions but also that of measurements made by a sensor in order to detect when to transmit, the number of possible communications toward the sensor if this is to be envisaged, the quantity of information to transmit, and so forth. In the first place this should lead to a reasoned taxonomy of EB configurations, and possibly to some clue to select the most suited one for the problem at hand. More in perspective, if some connection can be established between this matter and the system-theoretical analysis dealt with in the following, the expected cost could become part of an optimised synthesis—for example, helping choose the thresholds quite inherently required by any event generator.

2.2 Assessing an EB loop

As a starting point, suppose that a CT controller has been synthesised successfully (for the purpose of this manuscript, no matter how) and the resulting control loop is characterised by convenient stability and/or performance indices (phase margin, set point step IAE, or whatever is deemed appropriate).

2.2.1 Establishing the counterpart

When FR realisations are considered, the main issue is the selection of the sampling time T_s, and several criteria for that are available. Taking the CT loop as the counterpart for the FR one is thus natural, and the available theory allows to carry out the analysis quite in depth.

The same can be done for EB realisations, as shown e.g. in [9, 10]. This is a correct approach, since FR and EB are compared to a common basis, and covers any issue concerning the quality (*lato sensu*) of the obtained responses. On the other hand, however, such an analysis by itself says nothing on the cost-related questions as stemming from the typical motivations adduced for EB control. For such purposes, that may prove very relevant, the FR realisation should be somehow brought into play as a further counterpart.

Remarks. Analysis and assessment methodologies requiring knowledge of a quite reliable process model and/or of the process state produce neat and interesting results, as shown by the quoted works, but should be complemented with synthesis tools “for the EB-specific part” that only require knowledge of the CT controller and of basic (nominal) properties of the CT loop as forecast in the tuning—in fact, quite often this exhausts the available information. In this respect, establishing relationships with FR seems beneficial.

2.2.2 Comparing with FR control

In most works where EB and FR realisations of the same CT controller are somehow – and more or less explicitly – compared, very few (if any) words are spent on two relevant issues, namely which discretisation method was adopted and why, and how the FR sampling time was chosen. The first issue has also to do with the way the controller state is to be computed, i.e., with the *controller update rule* – a matter more relevant in EB than in FR – and is better dealt with later on, while the second is now briefly considered.

Sampling rate selection for FR

Crudely speaking for brevity, neglecting the T_s issue can impair EB/FR comparisons, especially because many FR implementations – also industrial ones, by the way – tend to oversample quite significantly. As such, the question to answer is how much a typical FR realisation can be “downsampled” with respect to typical sampling time selection criteria, so as to avoid biasing the comparison owing to an excessive penalisation of FR.

Basically, sampling time selection criteria follow three (possibly combined) reasoning paths, that can be summarised as follows.

1. *Proportionality to the cutoff frequency.* Assuming that an estimate $\widehat{\omega}_c$ is available for the closed-loop cutoff frequency ω_c, the sampling frequency $2\pi/T_s$ is constrained to exceed a multiple of $\widehat{\omega}_c$. The proportionality factor k_s is typically suggested to be in the range $(20, 100)$ and for which the author never encountered a such an advised value lower than 10.

2. *Phase margin reduction.* Since the effect of sampling and holding is quite well approximated in the Nyquist band by a delay of $T_s/2$, a phase margin reduction of $0.5\widehat{\omega}_c T_s$ is to be expected. If also the computation delay is accounted for, considering that said delay cannot exceed T_s otherwise the controller is not executing properly, a more pessimistic estimate for the phase margin reduction is $1.5\widehat{\omega}_c T_s$. In fact, the organisation and timing of the read/compute/actuate cycle has an impact on the estimate under question, but treating the matter in detail would stray from the scope of this work. As such, it can be assumed "on average" to estimate the phase margin reduction as $\widehat{\omega}_c T_s$, thereby constraining the sampling time to fulfil the inequality $\widehat{\omega}_c T_s < \Delta\varphi_m$, where $\Delta\varphi_m$ (in radians) is the accepted phase margin reduction. Of course also the choice of $\Delta\varphi_m$ could be discussed extensively, but to give a first-cut figure, a value greater than 0.175 (10°) is seldom encountered.

3. *Open-loop frequency response attenuation.* The magnitude of the open-loop frequency response $L(j\omega)$ at the Nyquist frequency $\omega_N = \pi/Ts$ indicatively quantifies the amount of measurement noise that can deteriorate signal components within the control band due to aliasing. Assuming (conservatively if the CT controller is sensibly synthesised) that $|L(j\omega)|$ rolls off with a $-20db/dec$ slope above the cutoff frequency, a required attenuation of $20dB$ – which is not particularly stringent – requires the Nyquist frequency to be ten times the cutoff, thus the resulting constraint is similar to that of the first item above with $k_s = 20$.

To present some illustrative figures, a very simple example is now examined. Consider the FOPDT (First Order Plus Dead Time) process

$$P(s) = \mu \frac{e^{-sD}}{1 + sT}, \quad D \geq 0 \tag{2.1}$$

in the asymptotically stable case, i.e., $T > 0$, Also, without loss of generality, assume for the purpose of this treatise that $\mu > 0$.

Given the above, take as CT controller the PI one

$$R_{PI}(s) = K\left(1 + \frac{1}{sT_i}\right) \tag{2.2}$$

and suppose the loop to be in nominal conditions, i.e., the process and the model used for the tuning coincide.

To synthesise the PI (2.2), employ the well known IMC (Internal Model Control) tuning rule

$$T_i = T, \quad K = \frac{T}{\mu(D + \lambda)} \tag{2.3}$$

where λ is traditionally interpreted as the desired closed-loop dominant time constant. This yields the (nominal) open-loop transfer function

$$L_n(s) = R_{PI}(s)P(s) = \frac{e^{-sD}}{s(D+\lambda)} \tag{2.4}$$

thus a cutoff frequency (and incidentally a phase margin) estimate given by

$$\widehat{\omega}_c = \frac{1}{D+\lambda}, \quad \widehat{\varphi}_m = \frac{\pi}{2} - \frac{D}{D+\lambda} \tag{2.5}$$

that in this ideal case are exact, thus avoiding any influence of process/-model mismatch and – as a related fact – of the particular procedure used to parametrise (2.1) if it has to be identified.

To further simplify the *scenario*, take a desired (nominal) phase margin φ_m° as design parameter, which leads to

$$\lambda = \frac{\varphi_m^\circ + 1 - \pi/2}{\pi/2 - \varphi_m^\circ} D \tag{2.6}$$

and bringing (2.5) in, to

$$\widehat{\omega}_c = \frac{1}{D}\left(\frac{\pi}{2} - \varphi_m^\circ\right), \quad \widehat{\varphi}_m = \varphi_m^\circ. \tag{2.7}$$

Applying k_s-based criteria for selecting T_s, and taking the equality sign for convenience, gives therefore

$$T_s = \frac{2\pi D}{k_s\left(\frac{\pi}{2} - \varphi_m^\circ\right)} \tag{2.8}$$

while if the $\Delta\varphi_m$-based criterion is taken, again with the equality sign and with angles in radians, the result is

$$T_s = \frac{D\Delta\varphi_m}{\frac{\pi}{2} - \varphi_m^\circ} \tag{2.9}$$

Supposing now $D = 1$ and $\varphi_m^\circ = \pi/4$, the k_s-based criteria with $k_s = 20$ give $T_s = 0.4$, while the $\Delta\varphi_m$-based one with $\Delta\varphi_m = 5^\circ$ yields $T_s = 0.1$. As such, a "reasonable T_s as suggested by the CT tuning" can be the average value of 0.25. Supposing furthermore an average situation between a dominant-rational and a dominant-delay process, i.e. for example $T = 1$, the resulting T_s is 1/4 of the integral time, and the control signal is computed about 25 times within the duration of a set point step response, evaluating the settling time as $5/\omega_c$. For completeness, λ turns out to be about 0.273, and K approximately 0.785.

Although the classical criteria just mentioned were used here with quite loose constraints, it can be easily observed that there would be still some room for downsampling. Figure 2.1 shows the closed-loop step response of the considered nominal system with $T_s = 0.25$ and $T_s = 1$. As can be seen, the major

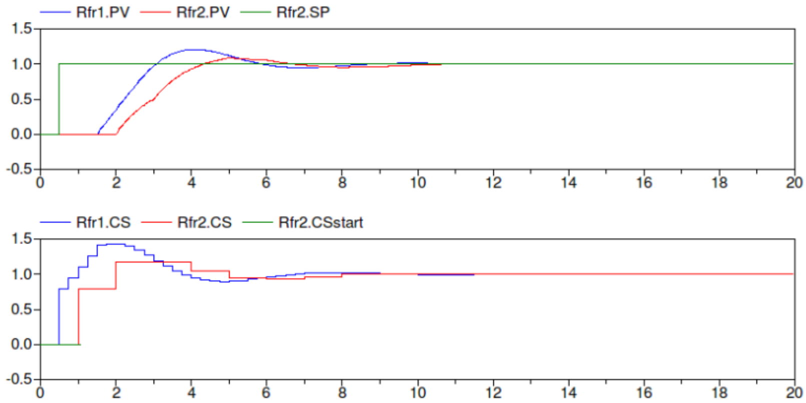

Figure 2.1: IMC-PI example 1 – Set Point (SP) step responses of the Process variable (PV) and the Control Signal (CS) with $T_s = 0.25$ and $T_s = 1$.

disadvantage of the larger T_s is a not immediate reaction to the set point step at $t = 0.5$, while for the rest of the transient the introduced degradation is definitely acceptable also if compared to those typically observed in EB-centric works. In particular, moreover, the settling part of the transient (quite expectedly) is hardly affected. As a final remark, this simple example backs up an advice frequently heard (but not so frequently adhered to) in the industry, i.e., to "execute the controller about four times in the closed-loop step response rise time": according to such a suggestion, 0.25 is quite good a choice.

Incidentally, the example allows to foresee another major advantage of EB, namely the prompt reaction to disturbances as they generate an event "immediately" (or better, at the time scale of the sensor internal sampling) without the need to with for the next *control* sampling.

Remarks. Assuming a CT loop as ultimate reference, both an FR and an EB realisation deteriorate the control quality. For FR, *given the discretisation method*, T_s is the independent variable against which the deterioration (whatever index is chosen) is to be viewed. It would be nice to have something analogous for EB, to give sense to statements such as "which realisation provides the lower cost with the same deterioration" (which is again comparing both FR and EB to CT) but also "how many transmissions can one save with EB compared to an FR realisation providing the maximum acceptable stability degree decrease" (which relates EB and FR more directly), or similar ones. In any case, a stability analysis of the EB scheme should be made and some stability criteria should be devised.

The discretisation method

In the FR case, discretisation can be carried out with several methods (the "exact" one relying on the sampling transformation $z = e^{sT_s}$, where s and z are respectively the Laplace and $\mathbb{Z}$ transform variables, the forward or backward difference one, the Tustin one, the ZOH – Zero Order Holder - one, and so forth). Each of said methods has a specific possible impact on the stability properties of the FR controller and thus of the FR loop, and rigorous methods are available to analyse the matter.

In the EB case, things are apparently totally different. The controller update rule replaces the mere discretisation, in the same way as the event triggering rule replaces periodic sampling. However, whereas periodic sampling only requires to fix T_s and to know the variable to measure, event triggering is a more articulated process, and according to the literature may require additional information, such as the set point, or computational burden, such as that for maintaining a model of the CT loop. Such a model can also be used for control generation, see [10] for both details on these aspects and a literature review on the general matter.

At present, this aspect does not seem to have yet received all the attention it deserves. The most frequently adopted solution is to apply a ZOH policy between two subsequent control events, possibly adapting the controller parameters based on the actual time elapsed since the previous one.

Remarks. When a comparison brings an FR counterpart into play, the discretisation method is to be accounted for. Also, here too it would be advisable to reduce the need for a process model to a minimum, for example – in a view to the relevant issue of (auto)tuning, for example – to be able of exploiting the synthetic information available in that context. Note that the required information is related to transmission cost, whence a further reason of interest. Of course, finally, deeper research on the controller update rule is in order.

2.2.3 Limit cycles

Several EB-related papers deal with the problem of limit cycles. However it seems that the interplay between the control update and the event triggering rules has not yet been fully investigated as a source for that phenomenon, which it apparently is.

Remarks. There seems to be room for further research here, but the matter is quite complicated, as also in a CT or FR context the determination of possible limit cycles and their stability properties is not a simple task unless for almost trivial dynamic structures. Chances are that if the EB paradigm is brought in, the matter is hardly tractable. However, as sketched out later on, maybe a proper selection of the triggering rule can allow for an analysis of the induced switching system.

2.3 Tuning for an EB realisation

As far as the PI(D) structure is considered, the great majority – not to say, virtually the totality – of tuning rules are conceived in the CT domain, as even a short glance at the huge review [24] immediately testifies.

More precisely, said rules invariantly start from some CT process description. This can be parametric (most frequently, a transfer function of structure decided *a priori*) or nonparametric (e.g., some characteristic points of a frequency response). In any case, the starting point to obtain such descriptions is a record of samples of some time-domain response, like a step or a relay one.

Consistently with the idea of a reference CT controller, as in this particular context complementing the analysis with an FR counterpart does not seem relevant, the most straightforward way to approach (auto)tuning of EB controllers seems to rely on tuning rules conceived for CT ones. Things are however different when it comes to turn a tuning *rule* into a tuning *procedure*, as in this respect some points need addressing.

The first one is how to manage the experiment—step, relay, or whatever. Since tuning occurs sparingly, one may think to carry out the operation using FR, but this requires extra controller/sensor communication, and the sensor must be capable of that (neglecting the additional consumption). However, since a process *stimulus* is applied, even if no information is fed to the sensor, transmissions are likely to be generated. The problem can then be posed and tackled of how to obtain the necessary tuning-related information from an EB sampling. Additionally, the samples' distances in time themselves may convey some information, and this too needs studying.

Remarks. A first aspect to study is how to obtain useful tuning information from EB sampling, relying e.g. to signal reconstructors conceived specifically for a given *stimulus*. For example, if relay feedback is used and the filtering hypothesis of the describing function approach is taken, the signal to be reconstructed is basically a sine wave with at most a few harmonics, and similar ideas can be used for other types of experiments. Then, the problem can be addressed of how a tuning procedure can possibly suggest EB-specific quantities like thresholds. Notice that in a similar way one could seek clues for T_s in a view to FR realisation, and although this is far simpler, nonetheless the autotuning literature is quite silent on that. Having both FR- and EB-related clues could finally allow for a forecast comparison of the two realisations.

2.4 Envisaged research directions

According to the previous remarks, we can envisage the following research directions.

- A first step would be the development of a reasoned taxonomy of EB configurations with respect to the energy- and/or the communication-critical components of the control loop. This could help firstly in choosing the best configuration for the problem at hand, and then in integrating the expected cost in optimised synthesis techniques;

- When the moment comes for a comparison between an EB and a CT realisation, there will be need for analysis and assessment methodologies which require only the knowledge of the CT controller and the basic (nominal) properties of the CT loop as forecast in the tuning;

- Up to now, EB realisations lack some (ideally, one) free tuning parameters which enable to tell which realisation provides the lower cost with the same deterioration (which is comparing both FR and EB to CT) and to govern the transmission saving given the maximum acceptable deterioration (which is instead comparing EB to FR);

- Another relevant issue is how to extract informations from EB sampling, which could be difficult as the sampling theorem could be violated. This opens another issue, i.e. how to turn this informations into EB-specific parameters like thresholds; if the same is done in the FR context, this would allow for a forecast comparison of the two realisations;

- Regardless of the presence of the mentioned parameters, stability and robustness criteria should be devised;

- The need for information of the controller update rule should be reduced to the minimum, avoiding wherever possible the explicit necessity for a process model;

- It should be advisable to study what happens when come into play network-induced delays and packet losses, multi-loop systems, and actuation and sampling are no more synchronous.

In this work, we will focus in particular on stability and robustness analysis.

Chapter 3

Stability

This chapter presents the main result regarding stability. The mathematical framework (which comprises the notation and some well-known properties) is established; then a set of hypotheses is presented, which lead to a class of event-based controllers that is very close to the fixed-rate case. Finally, after some preliminary results, a sufficient stability criterion is presented.

3.1 Notation and preliminaries

In the remainder of this work will be done an extensive use of singular values and matrix norms. Here are presented all the definitions and the properties which will be used throughout this thesis.

Given a matrix A and its spectrum $\rho(A) = \{\lambda_i | \lambda_i \text{is an eigenvalue of A}, \forall i\}$, A^H denotes the *conjugate transpose* of a matrix, A^T denotes the transpose of a matrix and $v^*(\cdot)$ denotes a signal sampled at events. The *singular values* of a matrix are then defined as follows.

Definition 1 *For a $m \times n$ complex matrix A, the square roots of the eigenvalues of $A^H A$ are called singular values of A.*

Singular values are collectively indicated with the notation $\sigma(A)$; they are always nonnegative scalars, so they can be ordered in descending order: $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$.

Another useful definition is that of a matrix norm. As there exists a wide variety of definitions, we will restrict our scope to the induced norms and, in particular, to the p-norms.

Definition 2 *Given vector norms on K^n and K^m (K being the real or complex field), the corresponding induced norm or operator norm in the space of the $m \times n$ matrices is defined as*

$$\begin{aligned} ||A|| &= \max\{||Ax|| : x \in K^n with \quad ||x|| = 1\} \\ &= \sup\left\{\frac{||Ax||}{||x||} : x \in K^n with \quad x \neq 0\right\} \end{aligned} \tag{3.1}$$

The operator norm corresponding to the p-norm for vectors is defined as

$$||A||_p = \sup_{||x|| \neq 0} \frac{||Ax||_p}{||x||_p}$$

Of particular interest is the 2-norm, also called *Euclidean norm* or *spectral norm*, which is defined as

$$||A||_2 = \sigma_1(A)$$

All the p-norms have the important property that for every matrix $A \in \mathbb{R}^{m \times n}$ and $x \in \mathbb{R}^n$ we have $||Ax||_p \leq ||A||_p \cdot ||x||_p$.

Now we can define the *condition number* of a matrix.

Definition 3 *The condition number of matrix A is defined as*

$$\kappa_p(A) = ||A||_p ||A^{-1}||_p$$

where $|| \cdot ||_p$ *denotes any of the p-norms.*

For our purposes, it is necessary to introduce the notion of contractivity of a matrix.

Definition 4 *A matrix A is said to be contractive or to be a contraction if there exists a constant* $L \leq 1$ *such that*

$$||Ax|| \leq L||x||$$

In light of the previously stated property of the p-norms, is it immediate to see that the aforementioned constant L can be identified with the p-norm of a matrix A; if p=2, then a simple contractivity criterion is to check whether the greatest singular value of a matrix A is lesser than 1. This fact was already known (and it's mentioned, for example, in [25, 26, 27]) and it may seem trivial to state it, but for the sake of clarity it has been preferred to give a detailed explanation of all the definitions used from here on.

3.2 General hypotheses

Hypothesis 1 *The process under control is described by a linear, time-invariant (LTI) single-input,single-output (SISO) model*

$$\begin{cases} \dot{x}_P(t) = A_P x_P(t) + b_P u(t - \tau) \\ y(t) = c_P x_P(t) \end{cases} \tag{3.2}$$

where t is the continuous time, $u(t) \in \mathbb{R}$ *the control signal,* $y(t) \in \mathbb{R}$ *the controlled variable* $x_P(t) \in \mathbb{R}^{n_P}$ *the process state vector,* $A_P \in \mathbb{R}^{n_P \times n_P}, b_P \in \mathbb{R}^{n_P \times 1}, c_P \in \mathbb{R}^{1 \times n_P}$ *constant matrices, and finally* $\tau \in \mathbb{R}, \tau \geq 0$ *a constant delay.*

Note that model 3.2 is strictly proper, without any loss of generality for our purposes. Furthermore it is supposed to be asymptotically stable, or to have at most just one pole in the origin.

Hypothesis 2 *A continuous-time LTI SISO controller that stabilises the nominal closed-loop system containing model 3.2 is available, and has the form*

$$\begin{cases} \dot{x}_R(t) = A_R x_R(t) + b_R(w(t) - y(t)) \\ u(t) = c_R x_R(t) + d_R(w(t) - y(t)) \end{cases} \tag{3.3}$$

where $x_R(t) \in \mathbb{R}^{n_R}$ *is the controller state,* $w(t) \in \mathbb{R}$ *the reference signal to be followed by* $y(t), A_R \in \mathbb{R}^{n_R \times n_R}, b_R \in \mathbb{R}^{n_R \times 1}, c_R \in \mathbb{R}^{1 \times n_R}$ *constant matrices, and* $d_R \in \mathbb{R}$ *a constant scalar.*

Controller 3.3 can be the result of an (auto)tuning procedure, and encompasses direct input/output feedthrough, like, e.g., a PI(D).

Hypothesis 3 *Controller 3.3 is realised with digital technology, and computes the discrete-time control u*(k) at events, which occur at time instants* t_k *counted by an integer* $k \in N$*, and in general not evenly spaced in time.*

Hypothesis 4 *Events are triggered by a single source (that here we assume to be the sensor).*

Hypothesis 5 *The time between two events is an integer multiple of a quantum* $q_s \in \mathbb{R}, q_s > 0$*.*

$$\forall t_h \leq t_k, \quad t_k - t_h = \varsigma(k,h) q_s$$

According to Hypothesis 4, two quantities can be defined:

- the *a priori step duration* $\overline{T}_s(k)$ that is decided at the k-th event;
- and the *a posteriori step duration* $\underline{T}_s(k)$, i.e., the time actually elapsed from the k-th to the (k + 1)-th event.

In literature, this approach has been called "Periodic Event-Triggered Control" (or PETC; see, for example, [14]), in opposition to the "Continuous Event-Triggered Control" (CETC), where the event-triggering condition is monitored continuously. With the former strategy, we are able to reach a satisfying compromise, as the event-transmission has now a periodic nature without sacrificing the benefits deriving from a reduced rate of transmission. Moreover, this approach is more suited for digital implementation than CETC, which requires dedicated hardware to detect events.

In practice, events can occur at the termination of $\overline{T}_s(k)$ or earlier, no matter why. In the former case $\underline{T}_s(k) = \overline{T}_s(k)$, while in the latter $\underline{T}_s(k) < \overline{T}_s(k)$. Notice that the event generation mechanism is in part reactive and in part proactive, the timeout being in fact the simplest way to decide when the next event has to occur. About this, literature distinguishes between "self-" and "event-triggered" (see for example [28]); the results of this work are applicable in both contexts, however.

Furthermore, Hypothesis 5 is well consistent with the way sensor electronics is typically designed. Most frequently, in fact, the sensor has a low-power part that is always active and polls the measured variable at frequency $1/q_s$. A high-power part takes conversely care of transmitting "when deemed necessary", i.e., based on a *triggering rule*, and is kept off otherwise.

Hypothesis 6 *If process 3.2 contains a delay, this can be approximated in the control-relevant frequency band by a rational transfer function, so that one can take as nominal continuous-time process model one with rational dynamics only.*

Assuming this may seem peculiar, but in fact many (auto)tuning methods rely on such models, typically obtained via Padé approximations. And even if the used tuning method is not of this type, in any non-pathological case it is possible to approximate a delay, within the control band, with simple enough a rational expression. No doubt this could somehow diminish the generality of the proposed approach, but nonetheless the variety of the usable tuning rules is still very large.

Hypothesis 7 *There is an upper bound for the time between two subsequent events, i.e.,*

$$\forall k, \quad \sigma(k) \in \Sigma = 1, \ldots, N, \quad 1 \leq N < \infty,$$

where

$$\sigma(k) := \varsigma(k+1, k).$$

This is realistic, as for safety reasons all real sensors encompass some "keep-alive" timeout, at the end of which an event is triggered unconditionally.

Note that we chose to preserve the notation from [1] for the inter-event time; this has nothing to do with the one for the singular values.

Hypothesis 8 *The control signal is applied only when an event occurs and is kept zero otherwise, i.e. the control action takes the form*

$$u(t) = \sum_{k=-\infty}^{+\infty} \delta(t - t_k)(c_R x_R(t_k) + d_R(w(t_k) - y(t_k)))$$

where $\delta(t - t_k)$ is the Dirac delta.

This is the so called 'impulse control', as presented in [2]. In a real case, however, applying an impulse as control action could be dangerous, so we interpose a low-pass filter between the controller and the process to smooth the impulse. From now on we will consider the augmented process model which encompasses the filter and allows to tune freely the (impulse) controller.

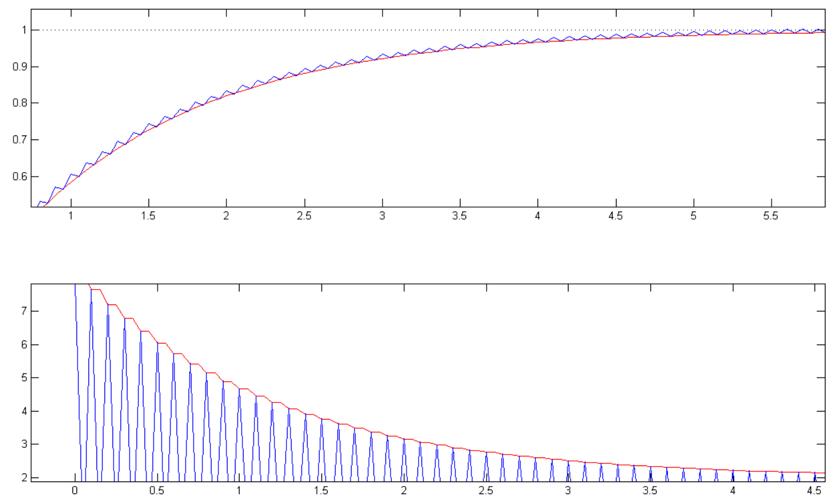

Figure 3.1: The (scaled) impulsive control follows correctly the ZOH output.

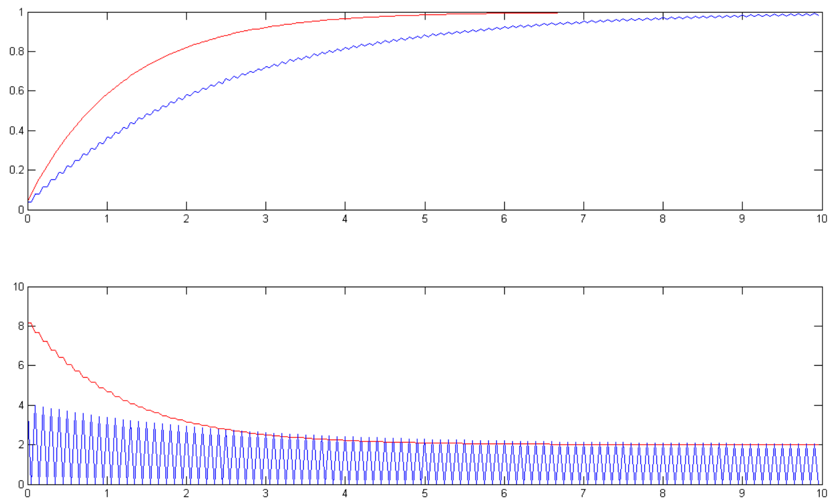

Figure 3.2: The output due to impulsive control diverges if no scaling factor is applied.

However, since we replace the Zero Holder with and impulsive actuation, the latter must be scaled in order to provide the correct output for the controller; the importance of the scaling factor can be summarized in Figures 3.1 and 3.2. We will see that the necessity of a scaling factor can be avoided with a suitable choice of the discretisation step for the controller.

As a final remark, one can notice that if the time constant of the filter is large enough, i.e. greater than q_s, the control action seen by the process can be well approximated to the one produced by a zero order holder.

Hypothesis 9 *When an event is triggered by the sensor, this results in the computation and actuation of a new control value. The delay between the triggered event and the control actuation is either negligible or known and constant, so that it can be taken as a part of the process model.*

This is the most strict hypothesis among those introduced, but is definitely realistic in at least two cases, both of interest for process control. The first one is when sensor, controller and actuator are co-located, and the reason for using an event-based controller is to reduce the actuator wear. In this case, the delay between sensor event and actuation is practically negligible. The second case (more central to this work) is when sensor, controller and actuator communicate via a network, but the underlying communication protocol is designed in such a way to practically eliminate packet collisions, that are the primary source of network-induced (variable) delays. At present not all protocols are capable of doing that, but a great research effort is being spent on the matter, see for example [29, 30, 31, 32], and solutions suited for the addressed context are arising; for example, in [33] a synchronisation scheme is proposed that, thanks to a novel and completely control-theoretical design, permits to make virtually any existing communication protocol slotted, thus making communication delays practically invariant. When such solutions will eventually become part of industrial systems, the hypothesis under question will be safely applicable to even more real-life cases.

3.3 Stability under arbitrary switching

In this section, some preliminary considerations will allow to reveal the system's switching nature; a sufficient stability criterion, under arbitrary switching, is then derived. The section is concluded with a corollary and some remarks.

3.3.1 Preliminary results

Before delving into the details of the theorem, we present the two following lemmas.

Lemma 1 *Given a square, invertible matrix $A \in \mathbb{R}^{n\times n}$, be σ_1 and σ_n its greatest and least singular value, respectively. Then*

$$\sigma_1(A^{-1}) = \sigma_n(A)^{-1}$$

Proof.

$$\begin{aligned} \sigma_1(A^{-1}) &= \sqrt{\lambda_1[(A^{-1})^T A^{-1}]} &= \sqrt{\lambda_1[(A^T)^{-1}A^{-1}]} = \\ &= \sqrt{\lambda_1[(AA^T)^{-1}]} &= \sqrt{\lambda_n^{-1}(AA^T)} = \\ &= \sqrt{\lambda_n(AA^T)}^{-1} &= \sigma_n(A^T)^{-1} \end{aligned}$$

But for a square, invertible matrix A, transposition doesn't change singular values, so $\sigma_n(A^T)^{-1} = \sigma_n(A)^{-1}$, thus concluding the proof.■

In light of this result, it is clear that the condition number of a matrix in the case of a 2-norm is $\kappa_2(A) = \frac{\sigma_1(A)}{\sigma_n(A)}$; it is worth noticing that, by definition, κ_2 is always greater or equal than 1.

Lemma 2 *[34, Th. 9] Let A and B be $n \times n$ matrices. If $1 \le k \le i \le n$ and $1 \le l \le n - i + 1$, then*

$$\sigma_{i+l-1}(A)\sigma_{n-l+1}(B) \le \sigma_i(AB) \le \sigma_{i-k+1}(A)\sigma_k(B)$$

In particular,

$$\sigma_i(A)\sigma_n(B) \le \sigma_i(AB) \le \sigma_i(A)\sigma_1(B)$$

and further

$$\sigma_n(A)\sigma_n(B) \le \sigma_n(AB), \qquad \sigma_1(AB) \le \sigma_1(A)\sigma_1(B)$$

For the proof of the lemma, refer to [34]. We will be interested in particular to the last two results.

3.3.2 Choice of the discretisation step

As pointed out in Hypothesis 8, a wise choice of the discretisation step for the controller can simplify the robustness analysis and the proof of the theorem.

Given that we are interested in confronting the results obtained with a Zero Holder (ZOH) with the results of the Impulse actuation (IH) and that, among all the possible discretisation steps, we focus only on $\sigma(k)q$ and q, there are four possible situations. In this section we summarize pros and cons of every case.

ZOH and step $\sigma(k)q$

This situation was analyzed in [1] and was the starting point for this work. The dynamic matrix of the system is

$$\begin{bmatrix} (A^*_{P,q})^\sigma - \left(\sum_{h=0}^{\sigma-1}(A^*_{P,q})^{\sigma-h-1}\right) b^*_{P,q} d_R c_P & \left(\sum_{h=0}^{\sigma-1}(A^*_{P,q})^{\sigma-h-1}\right) b^*_{P,q} c_R \\ -\left(\sum_{h=0}^{\sigma-1}(A^*_{R,q})^{\sigma-h-1}\right) b^*_{R,q} c_P & (A^*_{R,q})^{\sigma-1} \end{bmatrix}$$

Leva and Papadopoulos presented a theorem which grants asymptotic stability under arbitrary switching under few hypotheses; the great disadvantage is that this matrix is a function of σ, and one of our goals is to eliminate this dependence.

ZOH and step q

This approach was not analyzed as we aim to substitute the Zero Holder with and Impulse Holder; however, for the sake of completeness, the dynamic matrix is reported below:

$$\begin{bmatrix} (A^*_{P,q})^\sigma - \left(\sum_{h=0}^{\sigma-1}(A^*_{P,q})^{\sigma-h-1}\right) b^*_{P,q} d_R c_P & \left(\sum_{h=0}^{\sigma-1}(A^*_{P,q})^{\sigma-h-1}\right) b^*_{P,q} c_R \\ -b^*_{R,q} c_P & A^*_{R,q} \end{bmatrix}$$

IH and step $\sigma(k)q$

This approach replaces the Zero holder with an impulse actuation; since the energy of the control action must be preserved, a scaling factor is introduced. In the ZOH case, the control action was released in a span of time of duration $\sigma(k)q$, while in the IH case the control action is completely released in a step of duration q; thus the scaling factor is $\sigma(k)$.

The dynamic matrix is the following:

$$\begin{bmatrix} (A^*_{P,q})^{\sigma} - (A^*_{P,q})^{\sigma-1} b^*_{P,q} d_R c_P \sigma(k) & (A^*_{P,q})^{\sigma-1} b^*_{P,q} c_R \sigma(k) \\ -\left(\sum_{h=0}^{\sigma-1} (A^*_{R,q})^{\sigma-h-1}\right) b^*_{R,q} c_P & (A^*_{R,q})^{\sigma-1} \end{bmatrix}$$

The IH then suppresses the summations in the first row of the matrix, but leaves unchanged the second row. Although this is the correct way to examine the problem, the next subsection shows that further simplifications are possible, provided to accept a non strictly correct treatise of the problem.

IH and step q

According to Hypothesis 3, controller matrices should be discretised at step $\sigma(k)q$, as illustrated in the previous subsection; despite this fact, we choose to discretise all the matrices (both process and controller) at step q, always activating the controller at events. This approach may seem inconsistent, but in the next section it is shown that it has the advantage to produce a state matrix which does not depend on σ. Moreover, it implies that $\sigma(k) = 1$; thus no scaling factor is needed.

3.3.3 State matrix of the system

Suppose model 3.2 to be an exact description of the process; in presence of a delay in the continuous-time process, the state can be conveniently augmented to accomodate its rational approximation, preserving in this way the matrix notation. Now recall Hypotheses 1, 5 and 8; at the beginning of the k-th event, the nominal process is described by:

$$\begin{cases} x^*_P(k+1) = A^*_P(\underline{T}_s(k)) x^*_P(k) + b^*_P(\underline{T}_s(k)) u^*(k) \\ y^*(k+1) = c_P x^*_P(k+1) \end{cases} \tag{3.4}$$

where

$$A^*_P(\underline{T}_s(k)) := e^{A_P \underline{T}_s(k)}, \quad b^*_P(\underline{T}_s(k)) := \int_0^{\underline{T}_s(k)} e^{(A_P \underline{T}_s(k) - \xi)} b_P \, d\xi \tag{3.5}$$

Coming to the controller, let it be turned at the beginning of step k into a discrete-time one by some method of choice, using as discretisation period the sensor's sampling time q_s. This means computing $u^*(k)$ as the output of the dynamic system

$$\begin{cases} x^*_R(k) = A^*_{R,q} x^*_R(k-1) + b^*_{R,q}(w^*(k-1) - y^*(k-1)) \\ u^*(k) = c^*_{R,q} x^*_R(k) + d^*_{R,q}(w^*(k) - y^*(k)) \end{cases} \tag{3.6}$$

where the mentioned discretisation method provides matrices $A^*_{R,q}, b^*_{R,q}, c^*_{R,q}$ and the scalar $d^*_{R,q}$ - we do not explicitly indicate the dependence of those functions on the continuous-time matrices to lighten the notation. At the same instant, the process state and output are related to their values at the beginning of the previous step by 3.4, with the time indices shifted back by one.

Let us now evaluate both the state of the process and the state of the controller. At the beginning of the k-th event:

- The **state of the process** x_P is computed at a fixed rate, here assumed to be equal to the sensor's sampling time q_s:

$$\begin{aligned} x_P^*(\sigma q) &= (A_{P,q}^*)^\sigma x_P^*(0) + \sum_{h=0}^{\sigma-1} (A_{P,q}^*)^{\sigma-h-1} b_{P,q}^* u(h) = \\ &= (A_{P,q}^*)^\sigma x_P^*(0) + (A_{P,q}^*)^{\sigma-1} b_{P,q}^* u^*(0) \end{aligned} \tag{3.7}$$

- the **state of the controller** x_R is computed once, when the event occurs:

$$x_R^*(\sigma q) = A_{R,q}^* x_R^*(0) + b_{R,q}^*(w^*(0) - c_P x_P^*(0))$$

Putting it all together at step k we have an a posteriori closed-loop discrete-time system with state vector $x^*(k) = [x_P^*(k) x_R^*(k)]^T$ and dynamic matrix

$$\begin{bmatrix} {A_{P,q}^*}^\sigma - {A_{P,q}^*}^{\sigma-1} b_{P,q}^* d_R^* c_P & {A_{P,q}^*}^{\sigma-1} b_{P,q}^* c_R^* \\ -b_R^* c_P & A_{R,q}^* \end{bmatrix}$$

which can be further decomposed into

$$\begin{bmatrix} {A_{P,q}^*}^{\sigma-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{P,q}^* - b_{P,q}^* d_R^* c_P & b_{P,q}^* c_R^* \\ -b_R^* c_P & A_{R,q}^* \end{bmatrix} =$$

$$= \begin{bmatrix} A_{P,q}^* & 0 \\ 0 & I \end{bmatrix}^{\sigma-1} \begin{bmatrix} A_{P,q}^* - b_{P,q}^* d_R^* c_P & b_{P,q}^* c_R^* \\ -b_R^* c_P & A_{R,q}^* \end{bmatrix} =$$

$$= {A_{OL,q}^*}^{\sigma-1} A_{CL,q}^*$$

where I denotes the identity matrix of appropriate order, and $A_{OL,q}^*, A_{CL,q}^*$ denote the Open Loop and the Closed Loop matrices sampled at step q, respectively.

Now let us define:

- $b_{CL,\sigma_i,q}^* = \begin{bmatrix} {A_{P,q}^*}^{\sigma_i-1} b_{P,q}^* d_R^* \\ b_{R,q}^* \end{bmatrix}$
- $k = 1 = \sigma_1 q, k = 2 = (\sigma_1 + \sigma_2) q, \dots k = \sum_{i=1}^k \sigma_i q;$
- $A_{CL,\sigma_i,q}^* = {A_{OL,q}^*}^{\sigma_i-1} A_{CL,q}^*;$
- $x^*(0) = x_0;$

- and just for the sake of clarity, $\prod_{i=1}^{n} A_i = A_n A_{n-1} \cdots A_2 A_1$

Then, the state of the system at k-th event will be

$$\begin{aligned} x^*(k) &= \left[\prod_{i=0}^{k-1} A^*_{CL,\sigma_{k-i},q}\right] x_0 + \left[\sum_{i=1}^{k-1}\left(\prod_{j=0}^{i-1} A^*_{CL,\sigma_{k-j},q}\right) b^*_{CL,\sigma_{k-i},q} w(k-1-i)\right] \\ &+ b^*_{CL,\sigma_k,q} w(k-1) \end{aligned}$$

and the state matrix of the system is

$$A_{EB} = \prod_{i=0}^{k-1} A^*_{CL,\sigma_{k-i},q} \tag{3.8}$$

This reveals the system's switching nature, $\sigma(k)$ playing the role of the switching signal; to guarantee stability of the EB system, given the unpredictability of $\sigma(k)$, it is required to prove the stability of the system with dynamic matrix 3.8 under arbitrary switching in Σ. A sufficient condition for this is expressed by the following theorem.

3.3.4 A simple stability theorem

Theorem 1 *The system described by the dynamic matrix 3.8 is asymptotically stable under arbitrary switching in Σ if for each $\sigma(k)$ both the Open Loop and Closed Loop matrices are diagonalizable and it holds that $|\lambda_{MAX}(A^*_{CL,q})|\kappa_2(T_q) \leq \kappa_2(T_\sigma)^{-1}$ and $|\lambda_{MAX}(A^*_{CL,q})|\kappa_2(T_q) < 1$, where T_q, T_σ are matrices which diagonalize $A_{CL,q}, A_{OL,q}$ respectively.*

Proof. By hypothesis:

- $\exists T_\sigma, det(T_\sigma) \neq 0 \mid T_\sigma^{-1} A_{OL,q} T_\sigma = diag(\lambda_i) = D_\sigma$, where λ_i is an eigenvalue of $A_{OL,q}$;
- $\exists T_q, det(T_q) \neq 0 \mid T_q^{-1} A_{CL,q} T_q = diag(\eta_i) = D_q$, where η_i is an eigenvalue of $A_{CL,q}$.

Let $T_o = T_\sigma^{-1} T_q$, then 3.8 may be expanded and written as follows:

$$\begin{aligned} A_{EB} &= T_\sigma T_\sigma^{-1}(A_{OL,q})^{\sigma_k - 1} T_\sigma T_\sigma^{-1} T_q T_q^{-1} A_{CL,q} T_q T_q^{-1} T_\sigma \cancel{T_\sigma^{-1}} \ldots \\ \ldots & \quad \cancel{T_\sigma} T_\sigma^{-1}(A_{OL,q})^{\sigma_1 - 1} T_\sigma T_\sigma^{-1} T_q T_q^{-1} A_{CL,q} T_q T_q^{-1} T_\sigma T_\sigma^{-1} \\ &= T_\sigma D_\sigma^{\sigma_k - 1} T_o D_q T_o^{-1} D_\sigma^{\sigma_{k-1}-1} T_o D_q T_o^{-1} \ldots D_\sigma^{\sigma_1 - 1} T_o D_q T_o^{-1} T_\sigma^{-1} \end{aligned}$$

Coming to the singular values, we have, by Lemma 2:

$$\sigma_1(A_{EB}) \leq \sigma_1(T_\sigma)[\sigma_1(D_\sigma)]^{\sum_{i=0}^{k-1}\sigma_i - k} \sigma_1(T_o)^k \sigma_1(D_q)^k \sigma_1(T_o^{-1})^k \sigma_1(T_\sigma^{-1})$$

Because $\lambda_{MAX}(D_\sigma) = 1$ due to the matrix structure and by Lemma 1 we may write:

$$\sigma_1(A_{EB}) \leq \frac{\sigma_1(T_\sigma)}{\sigma_n(T_\sigma)} \left[\frac{\sigma_1(T_o)}{\sigma_n(T_o)}\right]^k \sigma_1(D_q)^k$$

By definition of contractivity of a matrix, we want $\sigma_1(A_{EB})$ to be lesser than 1. Let $M = \frac{\sigma_1(T_\sigma)}{\sigma_n(T_\sigma)} = \kappa_2(T_\sigma)$, then it must be:

$$\sigma_1(D_q)^k < \frac{1}{M} \left[\frac{\sigma_n(T_o)}{\sigma_1(T_o)}\right]^k$$

Again, by Lemma 2, we may write:

$$\sigma_1(D_q)^k \leq \frac{1}{M} \left[\frac{\sigma_n(T_\sigma^{-1})\sigma_n(T_q)}{\sigma_1(T_\sigma^{-1})\sigma_1(T_q)}\right]^k = \frac{1}{M} \left[\frac{\sigma_n(T_\sigma)\sigma_n(T_q)}{\sigma_1(T_\sigma)\sigma_1(T_q)}\right]^k = \frac{1}{\kappa_2(T_\sigma)^{k+1}\kappa_2(T_q)^k}$$

If we call $\sigma_1(D_q) = |\lambda_{MAX}(A^*_{CL,q})| = a$, $\kappa_2(T_q) = b$ and $\kappa_2(T_\sigma) = c$, given that all of these quantities are positive, we have to find when $(ab)^k \leq c^{-(k+1)}$. There are four possible scenarios:

1. $ab < 1,\ c < 1$;
2. $ab > 1,\ c < 1$;
3. $ab < 1,\ c > 1$;
4. $ab > 1,\ c > 1$.

The first two scenarios will never arise, because, as previously stated, $c > 1$ by its own nature; by hypotheses, we are working in scenario 3. But, in this case, we have to ensure that the left hand side of the inequality decreases faster than the right hand side, i.e. $ab \leq \frac{1}{c}$. Having granted it by hypotheses, the inequality holds, and thus the matrix A_{EB} is contractive. ■

It is worth noticing that if $ab = |\lambda_{MAX}(A^*_{CL,q})|\kappa_2(T_q) > 1$ the theorem doesn't cease to hold at all. Instead, it will hold up to a certain $k = \overline{k}$; the following corollary can then be stated.

Corollary 1 *If $|\lambda_{MAX}(A^*_{CL,q})|\kappa_2(T_q) > 1$, Theorem 1 still holds, but only up to greatest $k = \overline{k}$ such that the following inequality holds:*

$$\sigma_1(D_q)^k \leq \frac{1}{\kappa_2(T_\sigma)^{k+1}\kappa_2(T_q)^k}$$

3.3.5 Summary and conclusion

In this chapter, by suitably constraining the way the EB controller realisation is obtained, a sufficient condition for the stability of the switching system induced by the EB realisation was obtained, and this result is expressed by Theorem 1 and Corollary 1. The proof of Theorem 1 does not rely on a particular controller structure; thus extensions to other types of controllers with respect to the most typical “industrial” ones, such as state-feedback ones, could therefore be envisaged.

Concerning possible disturbances, one can notice that their influence on the stability of the closed-loop system, given its linear (switching) nature, can only be exerted by inducing a particular switching sequence. Therefore, once stability is guaranteed under arbitrary switching, it cannot be disrupted by construction.

The stability analysis part is thus concluded; we now move to robustness.

Chapter 4

Robustness

Model 3.2 quite obviously will never be an exact description of the process, due to uncertainties affecting the parameters, unmodeled dynamics and so on. In this chapter we will investigate the conditions under which a controller, tuned on the nominal system, assures stability also with the perturbed system.

4.1 Foreword

In Theorem 1, besides the diagonalizability of the matrices, the other hypotheses implicitly define a robustness region for our system. It is clear, indeed, that any controller (tuned on the nominal system) that succeeds in keeping any process affected by uncertainty in this region is robust with respect to asymptotic stability.

This region could be difficult to define analytically; as such, it could be more practical to numerically plot it. In the case of an IMC PID, for example, this region could be plotted in the $q_s - \lambda$ plane, where λ is the only tuning parameter of the controller.

4.2 Causes of uncertainty

Given a nominal process model $P(s, \theta_n)$, where θ_n is the parameters vector (in nominal conditions), we do not have a credible evaluation of variability for this process, because the model is "wrong". There are three main reasons for this:

- The process model $P(s, \theta_n)$ is poorer than the real process, in the sense that it could be under-parameterized, may neglect some dynamics, and so on;
- Even if we could afford a perfect model of the process, if it comes from an assembly line parameter variations between a product and another are inevitable;

- Last, there exists a slow time-variance that does not appear on the controller's time scale, but becomes important on the long run.

Not all of this causes can be properly counteracted; in particular, uncertainty due to time-variance is the worst case, and it is little remediable. Unmodeled dynamics pose a severe problem too, and in this case we can formulate only bland hypotheses. Parametric uncertainty is for sure the luckiest scenario, provided that we can formulate the problem in terms of few representative parameters. If it is not the case, dominating the complexity may become very challenging.

Note that despite identification methods can provide e.g. confidence interval on the obtained parameters, it would not be correct to use them to determine the set for robust control; it is conceptually wrong because telling that, for example, according to data $\mu \in [\underline{\mu}, \overline{\mu}]$ with a 95% confidence is just telling how much the model is incapable of explaining the data. It is, indeed, unrelated with the time variability of the parameters.

A correct approach to the problem would be, instead, finding some robustness index and expressing it as a function of the nominal data. In this way it is possible to express the greatest tolerable error, which implies that the controller tuning will be robust up to a certain bound; moreover, it is a more "extended" indication than other "punctual" indicator like, e.g., the phase margin or the gain margin. It is alway possible, however, that finding analytical bound may be a tough task (even impossible or not practicable); in this case may be more simple to numerically plot the stability / robustness region. Some examples of the construction of stability/stabilizing regions could be found in [35], [36] and in [37] (which uses a generalized Hermite-Biehler Theroem described in [38]).

Once the regions have been found, a population of processes is generated to test the correctness of the work; processes generated only in the vertices of the region do not suffice as often said regions are non-convex.

4.3 Modeling the uncertainty

For the sake of simplicity, we will conduct the analysis in a restricted scope.

- Third order process, which encompasses the process itself, a rational approximation of the delay (we will use a first-order Padé approximation) and the filter;
- PID controller, tuned on the nominal process through an IMC procedure;
- The (parametric) uncertainty does not change the order of the system nor the number of the eigenvalues.

Given a process model $P(s, \theta)$ in the frequency domain, let $(A^*_{P,q}(\theta_P), b^*_{P,q}(\theta_P), c^*_{P,q})$ be its realisation in the state space; the parametric uncertainty affects only the state matrix and the input channel, meaning that $A^*_{P,q}(\theta_P) = A^*_{P,q}(\overline{\theta} + \Delta\theta)$ and $b^*_{P,q}(\theta_P) = b^*_{P,q}(\overline{\theta} + \Delta\theta)$.

For a generic state matrix affected by uncertainties, if parameter variations are not excessive, every element in the matrix can be approximated as

$$a_{ij}(\theta_P) = a_{ij}(\overline{\theta}_P + \Delta\theta) \simeq a_{ij}(\overline{\theta}_P) + \nabla_\theta a_{ij}|_{\overline{\theta}}\, \Delta\theta$$

and so we may say

$$A(\theta_P) = A(\overline{\theta}_P + \Delta\theta) \simeq A(\overline{\theta}_P) + A_\Delta$$

Applying this to $A_{CL,q}$ and $A_{P,q}$ we obtain

$$A_{CL,q}(\theta) \simeq A_{CL,q}(\overline{\theta}) + A_{CL,q,\Delta} \quad (4.1)$$
$$A_{P,q}(\theta) \simeq A_{P,q}(\overline{\theta}) + A_{P,q,\Delta} \quad (4.2)$$

The perturbed matrices must then satisfy two kind of bounds; with the first, we require that their eigenvalues lie in the unitary circle (and are, possibly, distinct, to allow diagonalization) while the second is the “implicit” bound stated within Theorem 1.

As pointed out in the previous chapter, there are not trivial relationships between the eigenvalues of two matrices and those of their sum, unless said matrices possess some particular feature, like, e.g., symmetry, positive definiteness, and so on. Even if we know that the eigenvalues of $A(\overline{\theta})$ lie inside the unitary circle, determining the position of the eigenvalues of A_Δ does not help in localizing the eigenvalues of their sum; we must therefore study the matrix sum as a whole.

Robustness analysis begins from the frequency domain; having the transfer function, it is more convenient to localize the eigenvalues through criteria which study the coefficient of the characteristic polynomial. Given a polynomial $P(z) = a_n z^n + a_{n-1}z^{n-1} + \cdots + a_1 z + a_0$, to check if its roots lie in the unitary circle one may employ one of the following criteria:

- **Ordered coefficients**: *The roots of a polynomial $P(z)$ lie in the unitary circle if its coefficients are positive and in descending order, that is: $a_n \geq a_{n-1} \geq \cdots \geq a_1 \geq a_0 \geq 0$.* This is a sufficient condition of simple use;

- **Jury Criterion**: The Jury Stability Criterion is the discrete-time equivalent of the Routh-Hurwitz Criterion. It requires to build a table from the coefficients of the characteristic polynomial; by checking some inequalities, it proves that the eigenvalues lie in the unitary circle. It is a necessary and sufficient condition;

- **Gershgorin Circle Theorem**: *All of the eigenvalues of a matrix A lie in the union of the closed circles centered on the diagonal elements and whose radii are the sums of the off-diagonal element on the same row.* This theorem is a sufficient condition which could be useful only if the centers of the circles would lie in the unitary circle;

- **Bezoutians Matrices**: As explained in [39], Bezoutians matrices can be built starting from the characteristic polynomial. By checking some properties, like, e.g., rank, positive definiteness, an so on, it is possible to solve the root localization problem;

- **Bauer-Fike Theorem**[40]:*If* μ *is an eigenvalue of* $A + E \in \mathbb{C}^{n \times n}$ *and* $X^{-1}AX = D = diag(\lambda_1, \dots, \lambda_n)$*, then*

$$\min_{\lambda \in \lambda(A)} |\lambda - \mu| \leq \kappa_p(X)||E||_p$$

where $||\cdot||_p$ *denotes any of the p-norms.* This theorem is not very useful, as the bounds it allows to build are not fully contained in the unitary circle; however, it allows to draw an important consideration: if the condition number κ_2 is large, slight changes in A can induce large changes in the eigenvalues;

- **Jury-Bezout equations**: This is not a theorem or a novel method; it consists in computing the Jury inequalities and the bezoutian matrix, as proposed in [41]. The Jury Criterion ensures that the eigenvalues lie in the unitary circle; by requiring the bezoutian matrix to be positive definite (by mean of the Sylvester Criterion), we ensure that the eigenvalues are real and distinct. Even if analytical bounds cannot be computed, it is still possible to numerically plot the robustness region.

We will use the **Observability Canonical Form**. Given a transfer function G(z) in the discrete time, its state space representation is built as follows:

$$G(z) = \frac{b_{n-1}z^{n-1} + b_{n-2}z^{n-2} + \dots + b_0}{z^n + a_{n-1}z^{n-1} + \dots + a_0} \longrightarrow$$

$$A_{ob} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 1 \\ -a_0 & -a_1 & \cdots & \cdots & -a_{n-1} \end{bmatrix}, c_{ob} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$$

$$b_{ob} = \begin{bmatrix} \beta_{n-1} \\ \beta_{n-2} \\ \vdots \\ \beta_0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ a_{n-1} & 1 & \cdots & 0 \\ \vdots & \cdots & \ddots & 0 \\ a_1 & \cdots & a_{n-1} & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} b_{n-1} \\ b_{n-2} \\ \vdots \\ b_0 \end{bmatrix}$$

As $A_{ob} = A_{ob}(\theta)$, we will work with its approximation:

$$A_{ob} = A_{ob}(\theta) \simeq A_{ob}(\overline{\theta}_P) + \Delta A_{ob} =$$

$$= \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \\ -a_0(\overline{\theta}_P) & \cdots & -a_{n-1}(\overline{\theta}_P) \end{bmatrix} + \begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ -\nabla_\theta a_0|_{\overline{\theta}}\,\Delta\theta & \cdots & -\nabla_\theta a_{n-1}|_{\overline{\theta}}\,\Delta\theta \end{bmatrix}$$

In next sections the characteristic polynomials of both the Open Loop and Closed Loop matrices are computed; then the Jury-Bezout criterion is applied to determine the robustness region. Note that this is quite conservative: by the Jury-Bezout equations we require that the eigenvalues of a matrix are real, distinct, and lie in the unitary circle. The diagonalizability hypothesis of the theorem, instead, is looser, and may allow for couples of complex conjugate eigenvalues.

The Jury-Bezout equations are computed via the wxMaxima script reported in Attachment A; it contains a function, `jury_bezout_equations(p,var)` which takes as input the polynomial `p` and the variable `var` in which it must be evaluated and returns a column vector whose components are, top-down, the n Jury inequalities and the n Bezout inequalities (i.e., the principal minors of the bezoutian matrix); both must be posed ≥ 0.

4.4 Process State Matrix

Recall Hypotheses 1, 6 and 8; then the process model in the frequency domain is

$$P(s, \theta_P) = \mu \frac{e^{-sD}}{1 + s\tau} \frac{1}{s\tau_F + 1}$$

Approximating the delay with Padé rational transfer function, we obtain

$$\begin{aligned} P(s, \theta_P) &\simeq \mu \frac{1}{1 + s\tau} \frac{s - D/2}{s + D/2} \frac{1}{s\tau_F + 1} = \\ &= \frac{\mu s - D\mu/2}{s^3\tau\tau_F + s^2(D\tau\tau_F/2 + \tau_F + \tau) + s(1 + D\tau/2) + D\tau_F/2} \end{aligned}$$

Let us now discretise the transfer function; the selected method is Forward Euler (FE):

$$FE: \; s = \frac{z - 1}{T}$$

As all matrices are discretised at step q, T=q; note that only Backward Euler (BE) and the Tustin (TU) methods do preserve stability. However, if stability could be ensured with FE, it will surely be also with BE and TU.

The discrete transfer function is then

$$P^*(z, \theta_P) = \frac{N^*(z, \theta_P)}{D^*(z, \theta_P)}$$

where

$$
\begin{aligned}
N^*(z,\theta_P) &= z\frac{\mu}{q} - \frac{D\mu}{2} - \frac{\mu}{q} \\
D^*(z,\theta_P) &= z^3\frac{\tau_F\tau}{q^3} + z^2\frac{qD\tau\tau_F + 2q\tau - 6\tau\tau_F + 2q\tau_F}{2q^3} + \\
&+ z\frac{q^2D\tau - 2qD\tau\tau_F + q^2D\tau_F - 4q\tau + 6\tau\tau_F + 2q^2 - 4q\tau_F}{2q^3} + \\
&+ \frac{(\tau_F - q)(\tau - q)(qD - 2)}{2q^3} \\
&= a_3z^3 + a_2z^2 + a_1z + a_0 \qquad (4.3)
\end{aligned}
$$

The state-space model of the open-loop process is then

$$
A^*_{P,q}(\theta_P) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\frac{a_0}{a_3} & -\frac{a_2}{a_3} & -\frac{a_1}{a_3} \end{bmatrix}
$$

$$
b^*_{P,q}(\theta_P) = \begin{bmatrix} 0 \\ \frac{q^2\mu}{\tau\tau_F} \\ -\frac{q^2\mu(4\tau\tau_F - q\tau_F - q\tau)}{\tau^2\tau_F^2} \end{bmatrix} \qquad c^*_{P,q} = [1\ 0\ 0]
$$

Let $A^*_{P,q}(\overline{\theta}_P)$ be the state matrix evaluated in nominal conditions; then

$$
\Delta A^*_{P,q} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \Delta_0 & \Delta_1 & \Delta_2 \end{bmatrix}
$$

where

$$
\begin{aligned}
\Delta_0 &= -\frac{q(\tau_n - q)(\tau_F - q)}{2\tau_n\tau_F}(D - D_n) + -\frac{q(D_nq - 2)(\tau_F - q)}{2\tau_n^2\tau_F}(\tau - \tau_n) \\
\Delta_1 &= q\frac{2\tau_n\tau_F - q\tau_F - q\tau_n}{2\tau_n\tau_F}(D - D_n) + q\frac{D_nq\tau_F - 4\tau_F + 2q}{2\tau_n^2\tau_F}(\tau - \tau_n) \\
\Delta_2 &= -\frac{q}{2}(D - D_n) + \frac{q}{\tau_n^2}(\tau - \tau_n)
\end{aligned}
$$

The Jury-Bezout equations for the approximated characteristic polynomial are then computed (via the aforementioned wxMaxima script); the resulting vector will be a function of the parameters D, μ and τ. To plot the robustness region, it is sufficient to evaluate the vector within a reasonable set of values for the parameters and to plot only those values which make positive all of the components.

4.5 Closed-Loop State Matrix

Let us now introduce the controller R(s) and close the loop; we are not interested in finding bounds for the general case, so we will focus our attention on the case when a tuning method is given. For our purposes, we will choose the IMC tuning procedure, which is carried out on the nominal system; as P(s) is a third-order system, at least a PID(s) is required, and is computed as follows.

Let $T_o(s)$ be the desired closed loop transfer function; it has the form

$$T_o(s) = \frac{1}{(1+\frac{s}{\omega c})(1+\frac{s}{10\omega_c})}$$

where ω_c is the cut frequency of the transfer function, which will be used as a tuning parameter for the controller. If $P_1(s) = \frac{N_{1,P}}{D_{1,P}}$ is a minimum phase transfer function, then the controller $R_1(s)$ may be built as

$$R_1(z) = \frac{T_{o,q}}{1-T_{o,q}} \frac{D_{1,P,q}}{N_{1,P,q}}$$

where $T_{o,q}$ and $D_{1,P,q}, N_{1,P,q}$ are transfer functions discretised at step q_s. This is not the case, however; given a non-minimum phase process $P_1(s) = \frac{N_{Pmp}N_{Pnmp}}{D_P}$ and the previous desired closed loop transfer function T_o, the $PID_{IMC}(z)$ is built as follows:

$$PID_{IMC}(z) = \frac{T_{o,q}}{1-T_{o,q}} \frac{D_P}{N_{Pmp}}$$

Notice that, in this case, the closed loop transfer function that we actually see is

$$T_{nmp} = \frac{N_T N_{Pnmp}}{D_T + N_T(N_{Pnmp}-1)}$$

so the non-minimum phase part must be taken in account in the design process.

In an identical manner as shown in the previous section, a state-space model $(A^*_{CL,q}(\theta_P), b^*_{CL,q}(\theta_P), c^*_{CL,q})$ has been built and the Jury-Bezout inequalities have been found. Again, instead of analytical bounds, it has been preferred to compute the robustness region by sweeping the process' parameters subject to perturbations.

4.6 The implicit bound

Once ensured that the eigenvalues of the Open Loop and the Closed Loop matrices are Schur, real and distinct, it remains only to satisfy the bounds

$$|\lambda_{MAX}(A^*_{CL,q})|\kappa_2(T_q) \le \frac{1}{\kappa_2(T_\sigma)}, \qquad |\lambda_{MAX}(A^*_{CL,q})|\kappa_2(T_q) < 1$$

The main difficulty is the presence of the condition numbers. The problem is not the condition numbers themselves, but the fact they are referred to the diagonalization matrices, thus computing analytical bounds is strongly not recommended. Instead, given a controller $R(z, \theta_n)$ tuned on the nominal process, it is more convenient to plot the robustness region by sweeping the values of the parameters μ, τ, D (taken from reasonable sets).

4.7 Choice of the triggering rule

Once stability and robustness have been ensured under arbitrary switching, one is free to design the triggering rule that best fits to his needs; in our scope, this task primarily consists in achieving some requirements on the (*a posteriori*) control step duration. Basically, one wants it

- to increase as rapidly as possible towards its allowed maximum if the sensor triggers no event; to this end, it is employed the "send on delta" policy, that is: an event is generated only if the controlled variable (polled at rate $1/q_s$) differs in magnitude from the last transmitted by more than a prescribed amount Δy. This is what we mean with "only when needed";
- to allow reacting as soon as possible to an event, the minimum reaction time being q_s;
- to avoid event transmissions after the first one triggered by a controlled variable's variations.

It is important to recall that, in our framework, both the process and the controller are discretised at step q_s; it may seem reasonable, then, to choose it as a "small but reasonable" sampling period if it was adopted for a fixed-step realisation. However, to avoid an excessive event crowding, the lower bound for $\sigma(k)q_s$ should be chosen at least a decade greater than q_s; the upper bound, N, is instead a reasonable time-out. To achieve this, a subset $\tilde{\Sigma}$ is defined, with cardinality $\tilde{N} < N$ so that $\tilde{\sigma_1}$ be greater than 1, $\tilde{\sigma_1} q_s$ be a 'small but reasonable' sampling period if adopted for a fixed-rate realisation. For example, a good choice could be the set $\tilde{\Sigma} = \{10, 20, 50, 100, 200, 500\}$.

The *a priori* period $\overline{T}_s$ is first initialised to $\tilde{\sigma_1} q_s$; then, if a step of *a priori* duration $\sigma_i(k)q_s$ elapses, the next period is set as $\sigma_{i+1}(k)q_s$, until σ_N is reached. If conversely a step ends due to a sensor event, the next period is reset to $\tilde{\sigma_1} q_s$ and the system is forced to make it elapse. This surely may result in ignoring some events, but, as previously stated, $\tilde{\sigma_1} q_s$ is the same step that would be chosen for a fixed-rate realisation, so the stability of the system is preserved.

The finite state automaton in Figure 4.1 summarizes the selection of the *a priori* step duration. In figure, branches labeled with $\underline{T}_s = \overline{T}_s$ are traversed "by time-out" (i.e., when the *a priori* step duration elapses); branches labeled with $q_s \leq \underline{T}_s < \overline{T}_s(se)$, where "se" stands for "sensor event", are traversed in the opposite case.

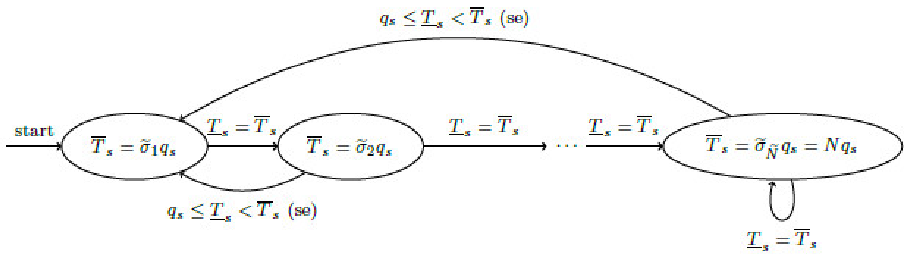

Figure 4.1: Finite state automaton for the *a priori* step duration selection (the time index k is dropped to lighten the notation).

The presented triggering rule can be slightly modified to take into account the possibility of a set point change after the loop has settled to an equilibrium; in this case, the *a priori* step duration is large and a set point variation will not cause an event until said duration elapses, which may happen long after the new set point was varied. Such an undesired behaviour can be counteracted in two ways. The first consists in generating a sensor event corresponding to the introduction of a new set point; this fully preserve the introduced hypotheses as the reason for *sensor*-generated events becomes irrelevant, once stability and robustness have been ensured as done herein. Quite obviously, in this way we must introduce further communications towards the sensor. The second possibility is to force *one* control computation (not a controlled variable measurement) when the set point is modified; this violates Hypothesis 4, as a second source of events is introduced. Furthermore, the value of the control signal will be computed with an outdated controlled variable. However, one can assume that said outdated value is close to the last transmitted one (otherwise some sensor events would have been triggered), and view the fact as an impulsive disturbance of moderate entity.

It is worth noticing that, although we granted stability under arbitrary switching, the automaton structure prevents some system jumps from happening. This implies that the stability condition may be loosen, so that only the non contractive jumps are the ones excluded by triggering rule. It is better to keep in mind, however, that the triggering rule may change due to unpredictable factors, and thus it is convenient to ensure stability for all the possible system jumps.

Moreover, having ensured *asymptotic* stability, the only possible source of limit cycles resides in numerical quantisation effects. This makes it easy to govern said cycles by acting on the rule parameter(s), like, e.g., the threshold in the send on delta one.

Finally, one may set $\Sigma = \{1, N\}$, which is consistent with our approach. If such a choice is adopted, most likely the majority of the control actions will be triggered by sensor events rather than time-outs; employing a send on delta policy means obviously means that control actions will be computed in response to variations of the controller variable greater than Δy. Such a situation is keen to generate larger actuator movements than the ones caused by control events triggered by timeouts; it is reasonable to conjecture that having some intermediate *a priori* step values induces a smoother actuator operation. Of course, if the actuator wear is not a relevant issue, less intermediate values can be chosen, without compromising the analysis.

Chapter 5

Simulations examples

This chapter presents a few simulations to show the validity of our approach, both for stability and robustness. It should be noted that the examples that follow are just explicative, and are not necessarily referred to physical systems.

5.1 Finding a suitable model

The great advantage of the theorem presented in Section 3.3 is that it does not rely on a particular structure for the matrices of the process model and the controller. However, for consistency with the approach illustrated in Chapter 4, we will chose the same canonical realization both for the process and the controller.

5.1.1 The process model

As said in Chapter 4, the process model takes the following form:

$$P(s, \theta_P) = \mu \frac{e^{-sD}}{1 + s\tau} \frac{1}{s\tau_F + 1}$$

Obviously, both the filter time constant and the discretisation step are constant parameters, not susceptible to variations.

The choice of the filter time constant is based on the largest *a priori* step duration; as we want the exponential decay of the impulsive control action to be as similar as possible to a constant control action, τ_F must slow down the decay until a new event occurs, which can happen at most after $\sigma_{MAX} q_s$ expires. At the end of the transient, there will be an error between the exponential decay and the ZOH control action which takes the form $1 - \eta$; we want this error to be little, so we impose:

$$\eta = 0.2,\ \sigma_{MAX} = 500 \longrightarrow e^{-\frac{\sigma_{MAX} q_s}{\tau_F}} \geq 1 - \eta \longrightarrow \tau_F \leq -\frac{\sigma_{MAX} q_s}{\ln(1 - \eta)}$$

$$\tau_F \leq 896.28 \longrightarrow \tau_F = 896$$

Notice that η is related to the send-on-delta threshold δ, as a smaller η will generate less events than a larger η.

5.1.2 The controller model

For our purposes, we are interested in two main type of controllers, the PI and the PID. Both will be tuned according to the IMC procedure, in nominal conditions; the tuning parameters will be λ for the PI and ω_c for the PID, and they represents the closed loop poles.

The PI controller takes the following form:

$$PI_{IMC}(s) = \frac{1 + s\tau_n}{s\mu_n(\lambda + D_n)}$$

where λ is the desired closed loop time constant. For the PID, given that we want a closed loop transfer function

$$T_o(s) = \frac{1}{(1 + \frac{s}{\omega c})(1 + \frac{s}{10\omega_c})}$$

in the case of a minimum phase process P the controller may be computed as:

$$PID_{IMC}(s) = \frac{T_{o,q}}{1 - T_{o,q}} \frac{1}{P_q}$$

where $T_{o,q}$ and P_q denotes the respective transfer functions discretised at step q_s. If, conversely, P is not minimum phase, instead of $\frac{1}{P}$ there will be $\frac{D_P}{N_{Pmp}}$, where N_{Pmp} denotes the minimum phase part of the process model P. Notice that, in this case, the closed loop characteristic polynomial will take in account also the non-minimum phase part. We remind again that the theorem does not rely on a particular type of controller; the choice of the PID structure is therefore totally arbitrary.

Finally, we will use the triggering rule presented in Section 4.7, with the set $\Sigma = \{10, 20, 50, 100, 200\}$

5.1.3 The FR comparison

To prove the advantages in terms of signal tracking and transmission saving, in the following simulation examples the EB plots are overlapped to the ones produced by a FR realisation. For the latter, it has been chosen a sampling rate $T_s = 0.1$ not to unduly favor the EB realisation. The blue line indicates the FR realisation, while the red one is the EB realisation.

5.2 First-order delay-free process and PI controller

The first simulation example we present is a very simple one (almost trivial); P(s) is a first order process without delay and so a PI(s) will suffice to control this system. The open loop stability region is reported in Figure 5.1.

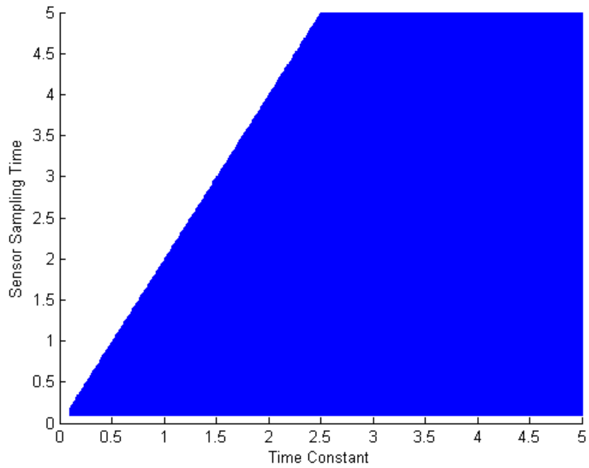

Figure 5.1: The stability region is not the whole plane but the area below the line $\tau = q/2$.

As it can be noticed, this region is not the whole $\tau - q$ plane; this is due to the use of Forward Euler as discretisation method, which does not preserve stability for every sampling time. From the Jury-Bezout equations we know that the process will be asymptotically stable with $2\tau > q$; let's say then $\tau = 2$ and $\mu = 1$.

In the nominal case, we get the same Jury-Bezout equations in closed loop (because of the controller choice), the only difference is τ being replaced by the tuning parameter λ; we can select then $q = 0.4$ and tune the PI(s) with $\lambda = 2$. The canonical realizations in nominal conditions are then:

$$A^*_{P,q} = 1 - \frac{q_s}{\tau_n} = 0.8,\ b^*_{P,q} = \frac{\mu_n q}{\tau} = 0.2,\ c_P = 1$$

$$A^*_{R,q} = 1,\ b^*_{R,q} = \frac{q}{\mu_n \lambda} = 0.2,\ c_R = 1,\ d^*_R = \frac{\tau_n}{\mu_n \lambda} = 1$$

In this situation, we suppose that the filter is integrated within the controller; if its time constant has been well tuned, the process will be fed with a constant control action, and so, for the simulation, we may suppose the presence of a ZOH.

5.2.1 Stability

Running the simulator yields the results reported in figures 5.2–5.4. The upper plot shows the set point, the measured variable (both in the fixed rate and in the event-based case) and two load disturbance steps; moreover, to validate the idea of non-sensor events expressed in Section 4.7, two ramp-like set point variation

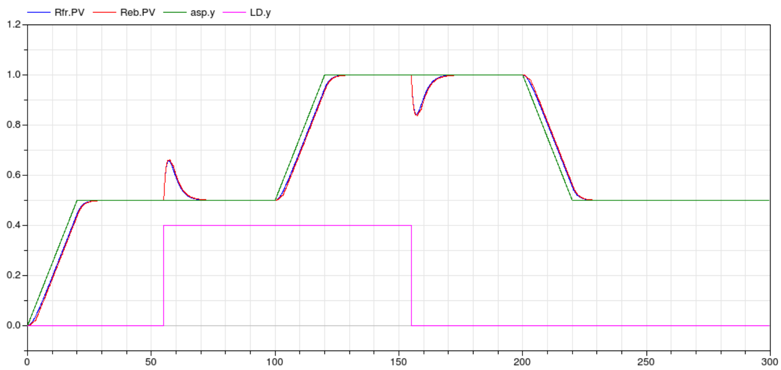

Figure 5.2: Measured variable, reference (green line) and load disturbances (purple line).

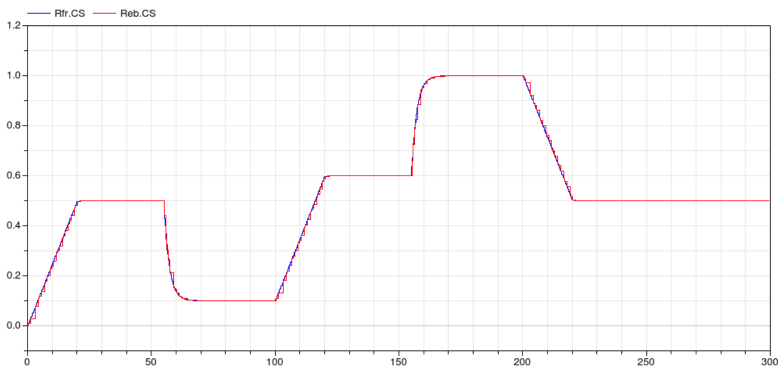

Figure 5.3: Control signal.

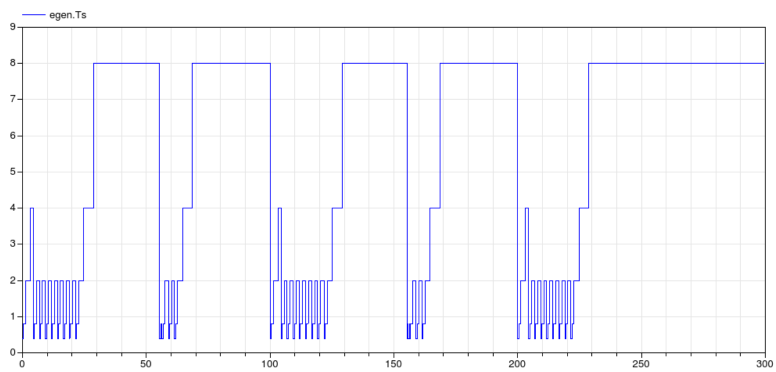

Figure 5.4: Inter-event time. Notice that once reached the steady state, the *a priori* step duration quickly reaches its maximum value.

have been introduced. Each ramp forces the computation of one control action; the plot shows that this is enough to trigger the sensor-originated necessary ones. As can be observed, the controlled variable plots are practically identical.

Regarding the control signal plots, it can be seen that the EB realization, during the transients, needs to update the control action's value less often than its FR counterpart; this implies that also the actuator will be used fewer times, and on the other hand that we will waste less computational resources to compute and update the control action's value.

Finally, the lower plot shows the values of T_s, allowing to appreciate the rate of growth of the step duration, which quickly escalates to its maximum value once it has been reached a steady state, reducing the needed sensor's transmissions. Note that there are more event hauls in response to the set point variation than to the disturbances.

5.2.2 Robustness

To test the robustness of our controller, we first plot the robustness regions. Keeping the same controller as in the previous section (i.e. $\lambda = 2$ and $q = 0.4$), a sweep on the values of τ from 0.1 to 10 (with step 0.01) yields a straight line starting from $\tau = 0.2$, as reported in Figure 5.5. Obviously this line falls completely into the region in Figure 5.1 as, for the process alone, once fixed the sensor sampling time the regions must coincide. In closed loop we have instead the region reported in Figure 5.6.

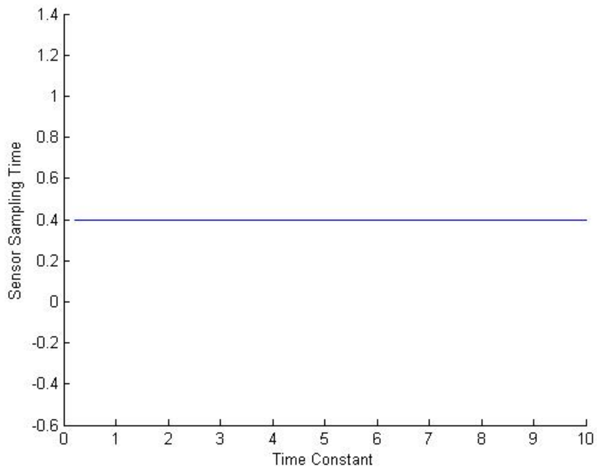

Figure 5.5: The robustness region in open loop.

Let $\mu = 1$ and $\tau = 1.5$; the simulation results are provided in Figures 5.7 - 5.9.

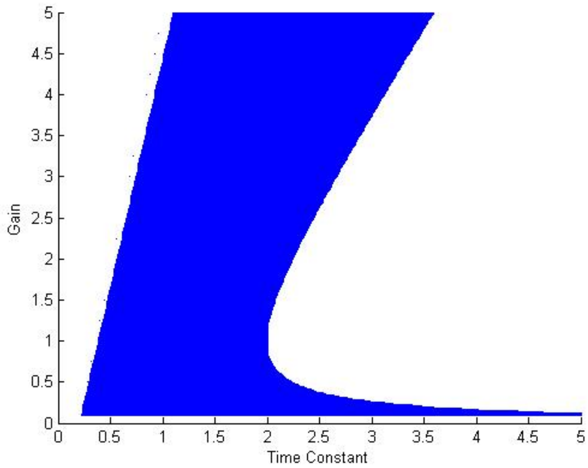

Figure 5.6: The robustness region in closed loop.

In Figure 5.7, the measured variable does not present appreciable deviations from the nominal case; the only remarkable change affects the peaks in response to the load disturbances. At a closer examination, in fact, it is possible to notice that they are more pronounced with respect to the nominal case.

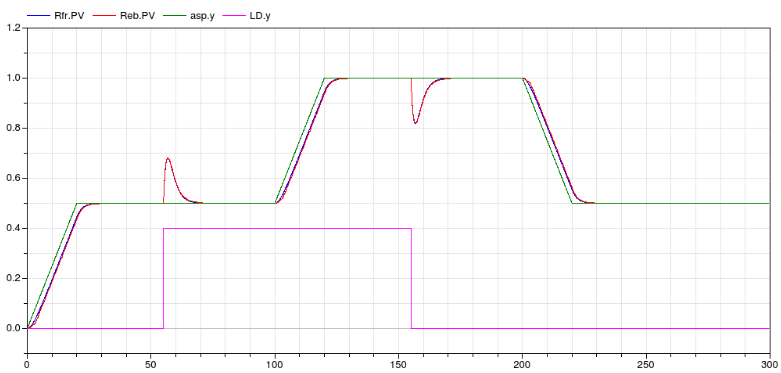

Figure 5.7: Measured variable, reference (green line) and load disturbances (purple line).

In Figure 5.8, the control signal too is very similar to the plot in Figure 5.3, the only remarkable feature being a smoother approach to the steady state following the ramp-like set point changes. The response to the load disturbances remains almost unchanged.

Finally, in Figure 5.9 the step duration rate growth has not deviated too much from the nominal case; it is possible to notice only a slightly greater event crowding following the load disturbances. The number of event hauls following the ramp-like variations has remained almost unchanged.

In the end, we can state that, following an accurate construction of the robustness regions, a PI controller tuned on the nominal process can ensure a stable behavior even with a perturbed process, with only minor changes in performances.

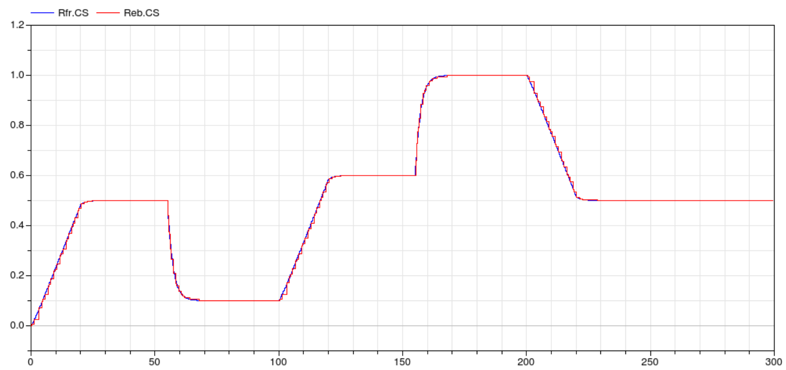

Figure 5.8: Control signal.

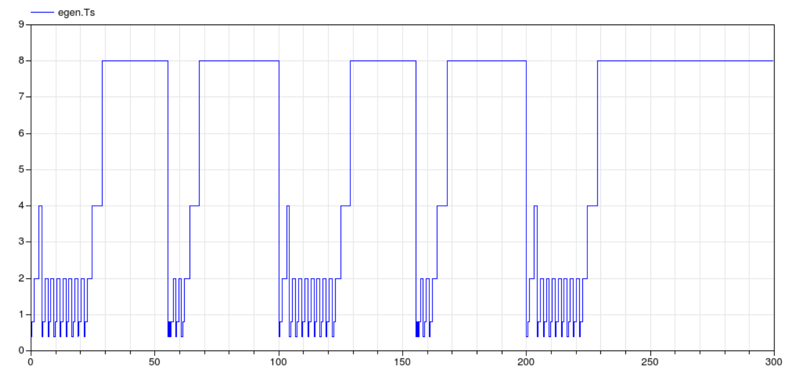

Figure 5.9: Inter-event time.

5.3 First order process with delay and PID controller

We now consider a more challenging example; for the sake of simplicity, the control action is considered again to be constant (following a well-done filter tuning). Following a sweep on the values of D, τ, q we obtain the stability region in open loop reported in Figure 5.10. This region does not overlaps with the positive orthant as we are employing Forward Euler as discretisation method;

if instead Backward Euler or Tustin were used, the sweep over the parameters' values would have returned the positive orthant.

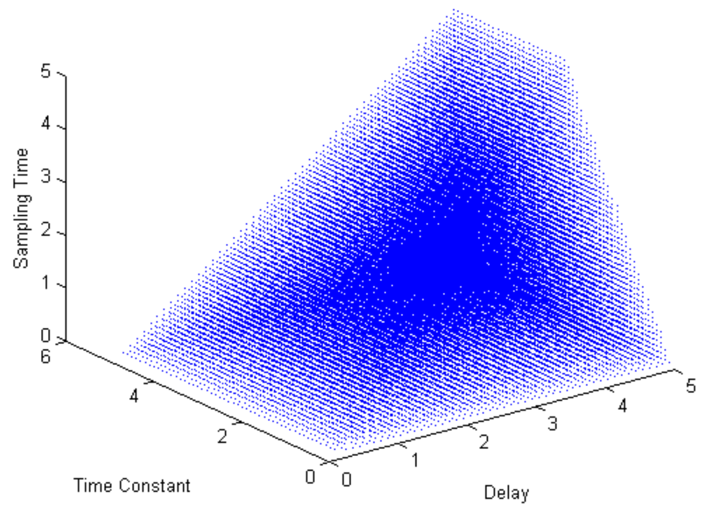

Figure 5.10: The stability region in open loop.

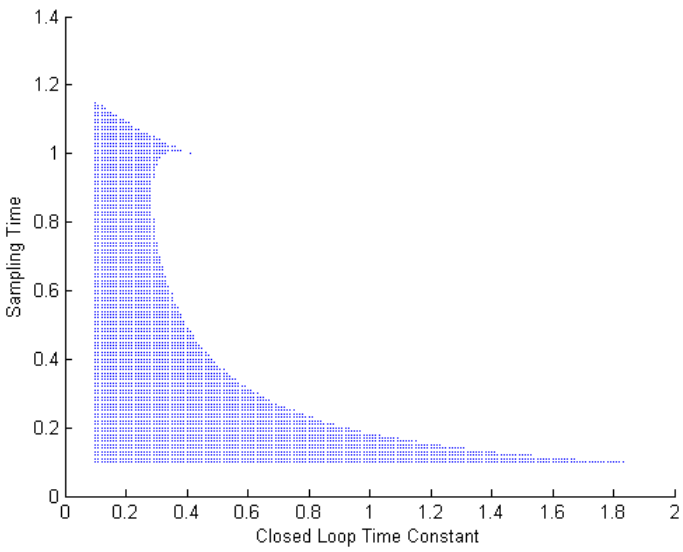

Figure 5.11: The stability region in closed loop.

For the process, we set again $\mu_n = 1$ and $\tau_n = 2$ and choose $D_n = 0.5$; in Figure 5.11 it is reported the stability region in closed loop. Let the controller be tuned with $\omega_c = 0.2$ and let $q = 0.1$; the canonical realization of the process is then:

$$A^*_{P,q} = \begin{bmatrix} 0 & 1 \\ \frac{(\tau-q)(Dq-2)}{2\tau} & -\frac{qD\tau-4\tau+2q}{2\tau} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.72 & 1.7 \end{bmatrix}$$

$$b^*_{P,q} = \begin{bmatrix} \frac{q\mu}{\tau} \\ -q\mu\frac{qD\tau-\tau+q}{\tau^2} \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.12 \end{bmatrix}, \quad c_{P,q} = [1\ 0]$$

and that of the controller is:

$$A^*_{R,q} = \begin{bmatrix} 0 & 1 \\ 11q\omega_c - 1 & 2 - 11q\omega_c \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.78 & 1.78 \end{bmatrix}$$

$$b^*_{R,q} = \begin{bmatrix} 9.2 \\ 7.976 \end{bmatrix} \cdot 10^{-4}, \quad c_{R,q} = [1\ 0], \quad d^*_{R,q} = \frac{10q^2\tau\omega_c^2 D}{\mu} = 0.004$$

5.3.1 Stability

Simulating the system in nominal conditions yields results reported in the Figures 5.12 - 5.14; as can be observed, the plot are not so different from the previous example.

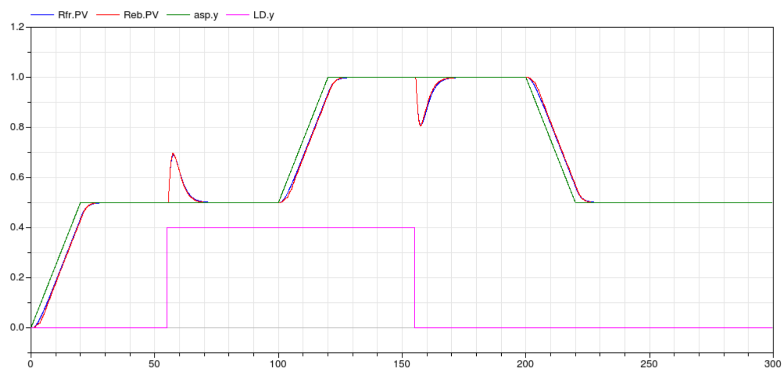

Figure 5.12: Measured variable, reference (green line) and load disturbances (purple line).

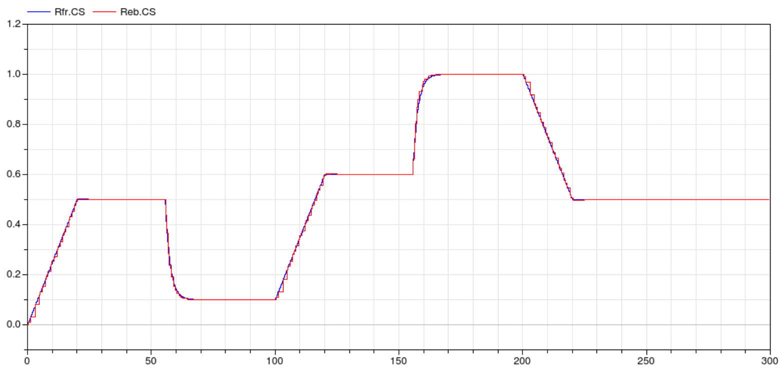

Figure 5.13: Control signal.

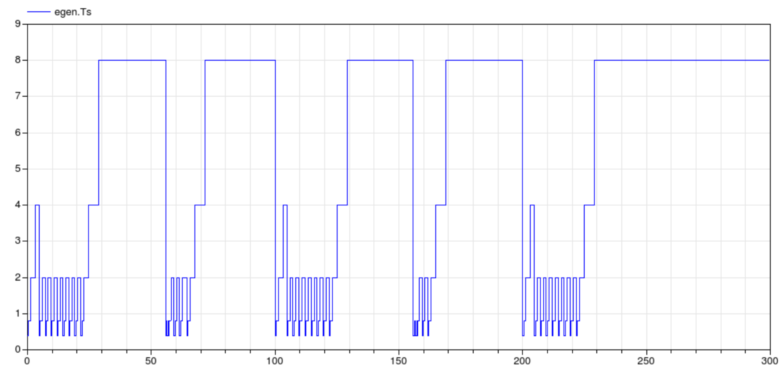

Figure 5.14: Inter-event time.

5.3.2 Robustness

Keeping the same controller as in the previous section (i.e. $\omega_c = 0.2$ and $q = 0.1$), a sweep on the values of D and τ returns the robustness region in open loop reported in Figure 5.15. The closed loop robustness region is instead reported in Figure 5.16.

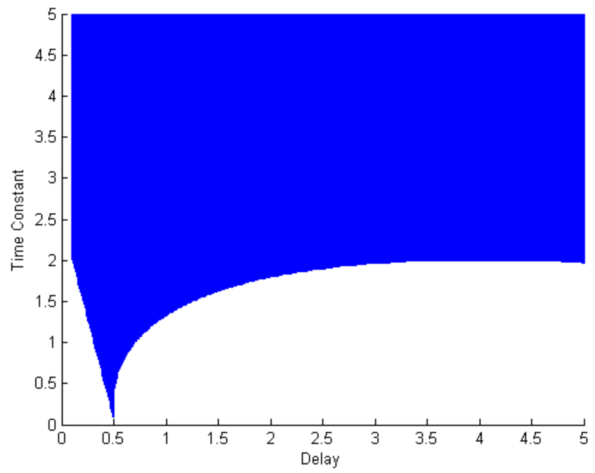

Figure 5.15: The robustness region in open loop.

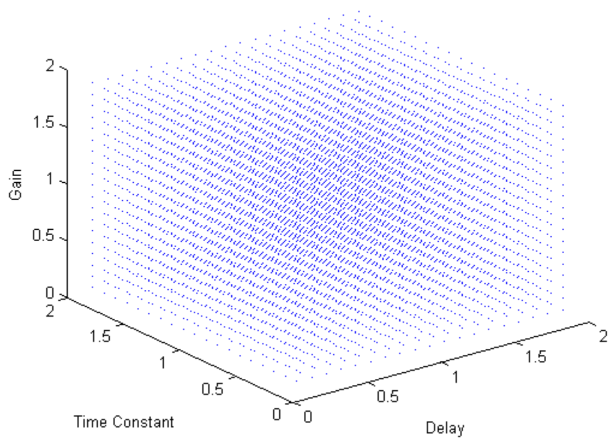

Figure 5.16: The robustness region in closed loop.

Let's set the $D = 1$ and again be $\mu = 1, \tau = 1.5$; the simulator yields the plots reported in Figures 5.17 - 5.19.

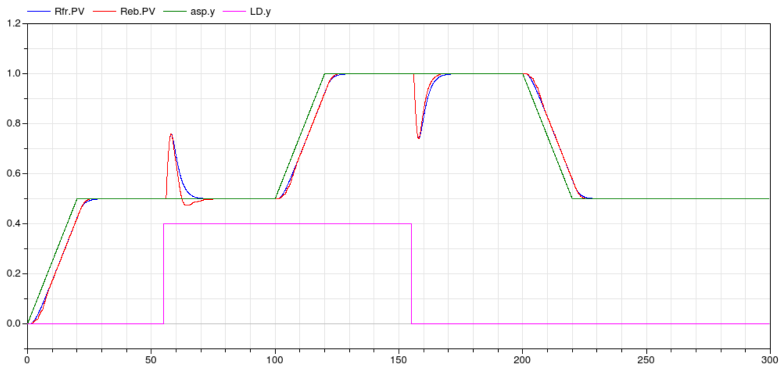

Figure 5.17: Measured variable, reference (green line) and load disturbances (purple line).

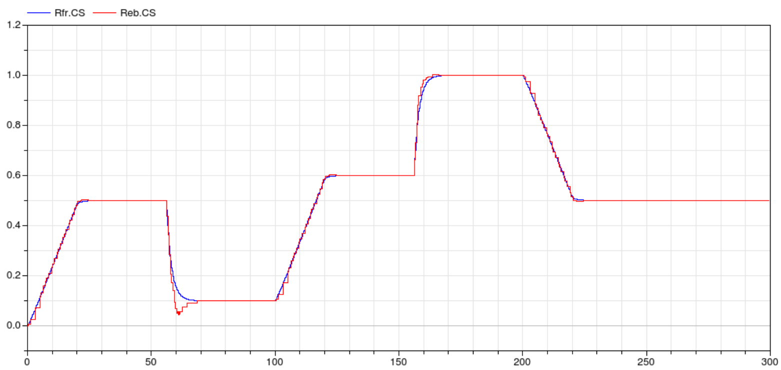

Figure 5.18: Control signal.

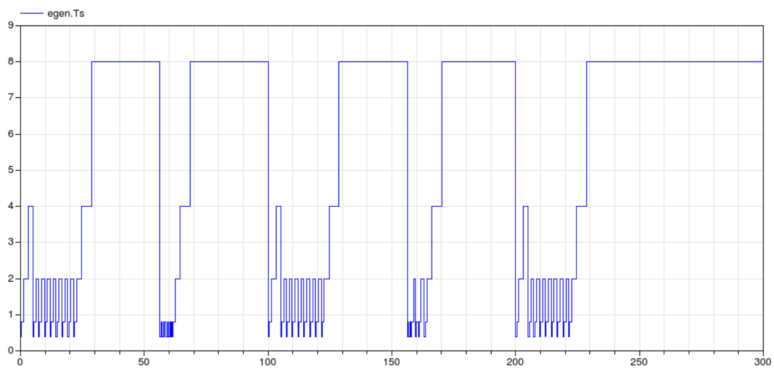

Figure 5.19: Inter-event time.

The plots are somehow more meaningful than their counterpart in the PI case. In the upper plot, besides the increment of the peaks height (more noticeable here than previously) and, again, the almost unchanged response to the ramps, the most remarkable feature is the presence of an undershoot, with the EB realisation, in response to the first load disturbance. Note that the measured variable in the EB realisation deviates from its FR counterpart also in response to the second load disturbance. This is an effect most likely due to the introduction of a derivative action, which amplifies the errors produced both by model uncertainties and load disturbances.

In the second plot, the control signal of the EB realisation consequently presents an undershoot in response to the first load disturbance, and deviates significantly from the FR realisation in response to the second disturbance. The plots present again a smoother approach to the steady state following the ramp-like set point variations.

Finally, the different behavior in response to the load disturbances influences also the transmission rate of the EB realisation. As can be seen, in response to the first disturbance there is an higher rate of event hauls than in the nominal case; a slightly higher rate of hauls (always with respect to the nominal case) is present also in response to the second disturbance.

In the end, we can state that even with a PID, following an accurate construction of the robustness regions, the controller proves to be robust against model uncertainties, the only secondary effect being the presence of an undershoot in response to a load disturbance.

Chapter 6

Conclusions and future developments

The subject of this work was the stability and the robustness analysis of an EB realisation of an industrial controller. Background on the Event-Based (EB) Control paradigm was first given and reviewed; a detailed insight on its motivations, its advantages with respect to the Fixed Rate (FR) case and the open problems was presented. Among all the possible research directions illustrated in Chapter 2, we chose to focus on the search of stability and robustness criteria.

A set of general hypotheses was then given, in accordance with the literature but also formalising facts that are almost ubiquitously true in the applications; through some preliminary results, the (induced) switching nature of the controlled process became clear. A sufficient stability theorem, under arbitrary switching, for the EB realisation of such a controlled process was proven. Although the PID structure i sthe most typically used in the applications, the proof of the theorem does not rely on this particular type of controller; extensions to, e.g., state-feedback or other type of controllers can then be envisaged. Incidentally, the theorem proved also that the EB control loop rejects disturbances effectively.

The robustness problem was then addressed; contrary to the approach adopted for stability, the structures of the process and the controller were here fixed to simplify the analysis. Due to the difficulty to find analytical bounds, numerical plots of the robustness regions were devised. As both stability and robustness were assessed under arbitrary switching, the triggering rule that best fit our needs was then presented. Simulation examples reported prove the correctness of our idea, evidencing a reduced rate of transmission and actuation with respect to the FR case.

The present work is, of course, far from being exhaustive. The first issue that needs to be improved is the stability analysis. We proved a theorem under the hypothesis that the controller is tuned with constant parameters, but the correct approach, as specified in Chapter 3.3, requires a time-variant controller; this will probably lead to a new, more general proof, which preferably encompasses the one previously presented.

Robustness analysis needs strong improvements, too. Here we presented only the case of parametric uncertainty, but it is clear that an exhaustive treatise must comprehend also the case of unstructured uncertainty, because, as stated in Chapter 4, parametric uncertainty is just a possible situation out of many (and just the more simple!). To this end, a possible approach consists in finding an overestimate to the (unknown) uncertainty — for example, see [42].

A third issue is that of the choice of the controller. Although the stability theorem does not rely on the controller structure, robustness analysis does so, and therefore a wise choice of the controller may simplify the analysis. It is important to point out, however, that even robustness analysis should be carried out in a general context, such that its results can be safely applied regardless of the controller type.

Finally, in light of the scenario outlined in Section 2.4, some long-term improvements could be envisaged in order to proceed towards a more complete treatise:

- according to Hypothesis 9, in our framework actuation and sampling occur synchronously and there are no packet losses; as stated, not all communication protocols at present day can offer such guarantees, thus more research in this direction is recommended, and the same can be said for multi-loop systems;

- the EB realisation presented in this thesis, in fact, does not solve the problem of determining a parameter capable of forecasting the transmission saving with respect to a FR realisation; a candidate could be the threshold of the send-on-delta policy. It would be advisable also defining a methodology to turn informations from EB sampling into a value of threshold;

- if possible, this parameter should be used to compare our EB realisation with the other existing, so to build a possible taxonomy of EB-realisation w.r.t energy- and/or communication-critical components.

As a final remark, note that here the CT controller was just taken as fixed. In fact, such controller would most likely come from a tuning procedure that is far from being error- and approximation-free. As such, if the required conditions are not fulfilled, one could well think about “slightly” modifying the controller so as to impose the situation — another interesting subject for future research.

Appendix A

In this attachment is reported the Maxima script used to compute the Jury-Bezout equations.

```
kill(all);

/* ──────────────────────────────────────────────────────────────────
   Return a monic polynomial with the same roots of p(var)
*/
monic_poly(p,var):=block([pe,pdeg,pem],
 pe     : expandwrt(p,var),
 pdeg   : hipow(pe,var),
 pem    : expandwrt(pe/coeff(pe,var,pdeg),var),
 return(pem)
);


/* ──────────────────────────────────────────────────────────────────
   Return the list of coefficients in the polynomial p(var)
   ordered by decreasing var powers
*/
poly_coeffs(p,var):=block([pe,pdeg,coeffs],
 pe     : expandwrt(p,var),
 pdeg   : hipow(pe,var),
 coeffs : makelist(
          coeff(pe,var,pdeg-i),i,0,pdeg
          ),
 return(coeffs)
);

/* ──────────────────────────────────────────────────────────────────
   Compute "next" row of a Jury table: 1st row is a0...an with the
   polynomial written e.g. as a0*z^n+...+an; nrow starts from 0
*/
next_jury_row(prevRow,nrow):=block([rl,a0,an,rr,nr],
 rl : length(prevRow),
 a0 : prevRow[1],
 an : prevRow[rl-nrow],
 rr : makelist(prevRow[rl-nrow-i],i,0,rl-nrow-1),
 for j:1 thru nrow do
     rr:append(rr,[0]),
```

```
 nr : prevRow-an/a0*rr,
 return(nr)
);

/* ------------------------------------------------------------------
   Compute the Jury table of the polynomial p(var)
*/
jury_matrix(p,var):=block([coeffs,cl,rr,Jm],
 coeffs : poly_coeffs(p,var),
 cl     : length(coeffs),
 rr     : coeffs,
 Jm     : matrix(rr),
 for i:0 thru cl-2 do (
     rr : next_jury_row(rr,i),
     Jm : addrow(Jm,matrix(rr))
     ),
 return(ratsimp(Jm))
);

/* ------------------------------------------------------------------
   Compute the Bezoutiant matrix of a generic monic polynomial of
   degree n, defining its coefficients as a[i], i.e., supposing
   (with x as var name) that the polynomial is written as
   p(x) = x^n+a_1x^(n-1)...+a_n
*/
gen_bezoutiant(n):=block([sigma,s,eqs,Bel,B],
 sigma : makelist((-1)^i*a[i],i,1,n+1),
 s     : makelist(s[i],i,1,n+1),
 eqs   : makelist(
           sigma[k]
           -1/k*(sum(
                  (-1)^(i-1)*sigma[k-i]*s[i],i,1,k-1
                 )
           +(-1)^(k-1)*s[k]),k,1,n+1
         ),
 Bel   : subst(a[n+1]=0,
                append([n],
                    makelist(
                      rhs(solve(eqs,s)[1][i]),i,1,n+1
                    )
                )
         ),
 B     : hankel(Bel),
 for i:1 thru length(Bel)-n do (
     B : submatrix(length(Bel)-i+1,B,length(Bel)-i+1)
 ),
 return(B)
);

/* ------------------------------------------------------------------
   Compute the Bezoutiant matrix of the polynomial p(var)
*/
bezoutiant(p,var):=block([mpcs,pdeg,gb,slist,B],
 mpcs   : poly_coeffs(monic_poly(p,var),var),
```

```
 pdeg    : length(mpcs)-1,
 gb      : gen_bezoutiant(pdeg),
 slist   : makelist(a[i]=mpcs[i+1],i,1,pdeg),
 B       : factor(subst(slist,gb)),
 return(B)
);

/* ---------------------------------------------------------------------
   Compute the principal minors of matrix M
*/
principal_minors(M):=block([m,pms],
 m       : M,
 pms     : [ratsimp(determinant(m))],
 for i:1 thru length(M)-1 do (
  m      : submatrix(length(m),m,length(m)),
  pms    : append(pms,[ratsimp(determinant(m))])
 ),
 return(reverse(pms))
);

/* ---------------------------------------------------------------------
   Express the Jury-Bezout equations of the polynomial p(var)
*/
jury_bezout_equations(p,var):=block([jm,B,jbeqs],
 jm       : jury_matrix(p,var),
 B        : bezoutiant(p,var),
 pmB      : principal_minors(B),
 jbeqs    : addrow(col(jm,1),transpose(matrix(pmB))),
 return(jbeqs)
);
```

Bibliography

[1] A. Leva and A.V. Papadopoulos. Tuning of event-based industrial controllers with simple stability guarantees. *Journal of Process Control*, 2013.

[2] K. J. Åström. Event based control. In A. Astolfi and L. Marconi, editors, *Analysis and Design of Nonlinear Control Systems*, pages 127–147. Springer Berlin Heidelberg, 2008.

[3] K.E. Arzen. A simple event-based PID controller. In *Proceedings of the 14th IFAC World Congress*, pages 423–428, Beijing, August 1999.

[4] Sylvain Durand, Nicolas Marchand, et al. Further results on event-based pid controller. *Proceedings of the European Control Conference 2009*, pages 1979–1984, 2009.

[5] X. Nan, L. Yujuan, and Z. Jinhui. Some improvements on event-based PID controllers. In *Proceedings of the 32th Chinese Control Conference*, Xi'an, China, July 26-28 2013.

[6] X. Wang and M. D. Lemmon. Event design in event-triggered feedback control systems. In *Proceedings of the 47th Conference on Decision and Control*, Cancun, Mexico, Dec. 9-11 2008.

[7] A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos. Event-triggered control for discrete-time systems. In *Proceedings of the American Control Conference*, pages 4719–4724, 2010.

[8] C. Xia and H. Fei. Stability of event-triggered output-feedback control systems. In *Proceedings of the 30th Chinese Control Conference*, Yantai, China, July 22-24 2011.

[9] D. Lehmann and J. Lunze. A state-feedback approach to event-based control. *Automatica*, 46(1):211–215, 2010.

[10] D. Lehmann and J. Lunze. Extension and experimental evaluation of an event-based state-feedback approach. *Control Engineering Practice*, 19(2):101 – 112, 2011.

[11] D. Lehmann and J. Lunze. Event-based output-feedback control. In *19th Mediterranean Conference on Control and Automation*, Aquis Corfu Holiday Palace, Corfu, Greece, June 2011.

[12] S. Al-Arequi, D. Gorges, and S. Liu. Robust event-based control and scheduling of networked embedded control systems. *Estimation and Control of Networked Systems*, 4(1):7–14, 2013.

[13] Q. Liua, Z. Wang, X. He, and D.H. Zhou. A survey of event-based strategies on control and estimation. 2:90–97, 2014.

[14] W. Heemels, M. Donkers, and A. Teel. Periodic event-triggered control based on state feedback. In *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Orlando, FL, USA, December 12-15 2011.

[15] Hai Lin and P.J. Antsaklis. Stability and stabilizability of switched linear systems: A survey of recent results. *Automatic Control, IEEE Transactions on*, 54(2):308–322, Feb 2009.

[16] R. Shorten, F. Wirth, O. Mason, K. Wuff, and C. King. Stability criteria for switched and hybrid systems. *SIAM Review - The Flagship Journal of the Society for Industrial and Applied Mathematics*, 49(4):545–592, 2007.

[17] X. Ji. Quadratic robust stabilization of a class of switched linear systems with uncertainties. In *International Conference on Intelligent Computation Technology and Automation*, 2008.

[18] Z. Xiaoli. Average dwell time approach robust stabilization of switched linear systems with time-delay. In *Control and Decision Conference*, Guilin, June 2009.

[19] X. Liu and G. Duan. Robust stabilization of switched systems with input saturation. In *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, Guangzhou, August 2005.

[20] Z. Sun. A general robustness theorem for switched linear systems. In *Proceedings of the 2005 IEEE*, Limassol, Cyprus, June 2005.

[21] R. Postoyan, A. Anta, D. Nesic, and P. Tabuada. A unifying lyapunov-based framework for the event-triggered control of nonlinear systems. In *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Orlando, FL, USA, December 12-15 2011.

[22] A. Trofino, D. Assmann, C. C. Scharlau, and D. F. Coutinho. Switching rule design for switched dynamic systems with affine vector fields. In *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, Shangai, P.R. China, December 16-18 209.

[23] Z. Sun and S. S. Ge. *Stability Theory of Switched Dynamical Systems*. Springer-Verlag, London, 2011.

[24] A. O'Dwyer. *Handbook of PI and PID controller tuning rules*. World Scientific Publishing, Singapore, 2003.

[25] R. Bhatia. *Matrix Analysis*. Spinger-Verlag, New York, 1952.

[26] C.C. Paige, G.P.H. Styan, B.Y. Wang, and F. Zhang. Hua's matrix equality and Schur complements. *International Journal Of Information And Computer Sciences*, 4(1):124–135, 2008.

[27] Y. Zhang. Singular value inequalities of Lewent type. *Taiwanese Journal Of Mathematics*, 2013.

[28] W. Heemels, K. Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. In *2012 51st IEEE Conference on Decision and Control(CDC)*, pages 3270–3285, 2012.

[29] J. Elson and K. Römer. Wirelss sensor networks: a new regime for time synchronisation. *SIGCOMM Computer Communication Review*, 33(1):149–154, 2003.

[30] L. Mottola and G.P. Picco. Programming wirelss sensor networks: Fundamental concepts and state of the art. *ACM Computing Surveys*, 43(3):19:1–19:51, April 2011.

[31] G. Zhou, Y. Wu, T. Yan, T. He, C. Huang, J.A. Stankovic, and T.F. Abdelzaher. A multifrequency mac specially designed for wireless sensor networks application. *ACM Transaction in Embedded Computing Systems*, 9(4):39:1–39:41, March 2010.

[32] C. Raghavendra, K. Sivalingam, and T. Znati. *Wireless Sensor Networks*. Spinger-Verlag, London, UK, 2006.

[33] A. Leva and F. Terraneo. Low power synchronisation in wireless sensor networks via simple feedback controllers: the FLOPSYNC scheme. In *2013 American Control Conference(ACC)*, Washington, DC, USA, June 17-19 2013.

[34] J. K. Merikoski and R. Kumar. Inequalities for spreads of matrix sums and products. *Applied Mathematics E-Notes*, (4):150–159, 2004.

[35] Y. J. Huang and Y. Wang. Robust pid tuning strategy for uncertain plants based on the kharitonov theorem. *ISA transactions*, 39(4):419–431, 2000.

[36] M. T. Söylemez, N. Munro, and H. Baki. Fast calculation of stabilizing pid controllers. *Automatica*, 39(1):121–126, 2003.

[37] G. J. Silva and A. Datta andS. P. Bhattacharyya. New results on the synthesis of pid controllers. *Automatic Control, IEEE Transactions on*, 47(2):241–252, 2002.

[38] M. Ho, A. Datta, and S. P. Bhattacharyya. Generalizations of the hermite–biehler theorem. *Linear Algebra and its Applications*, 302:135–153, 1999.

[39] G. Heinig and K. Rost. Introduction to bezoutians. In *Numerical Methods for Structured Matrices and Applications*, volume 99 of *Operator Theory Advances and Applications*. Birkhauser Springer Basel AG, 2010.

[40] G. Golub and C. F. Van Loan. *Matrix Computations*. The John Hopkins University Press, third edition, 1996.

[41] Moira Paggini. Il Criterio di Sylvester sulle radici reali dei polinomi, 2011-2012.

[42] A. Leva and A. M. Colombo. Estimating model mismatch overbounds for the robust autotuning of industrial regulators. *Automatica*, 36(12):1855–1861, 2000.