



HAL
open science

Commande événementielle : applications aux systèmes robotiques

Bruno Boisseau

► **To cite this version:**

Bruno Boisseau. Commande événementielle : applications aux systèmes robotiques. Automatique / Robotique. Université Grenoble Alpes, 2017. Français. NNT : 2017GREAT037 . tel-01690033

HAL Id: tel-01690033

<https://theses.hal.science/tel-01690033>

Submitted on 22 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Automatique - Productique**

Arrêté ministériel : 25 mai 2016

Présentée par

Bruno BOISSEAU

Thèse dirigée par **Nicolas MARCHAND**,
codirigée par **John-Jairo MARTINEZ-MOLINA**,
et coencadrée par **Thibaut RAHARIJAONA**

préparée au sein du
laboratoire GIPSA-lab, département automatique
dans l'école doctorale **EEATS**

Commande événementielle : application aux systèmes robotiques

Thèse soutenue publiquement le **30 juin 2017**,
devant le jury composé de:

Luc DUGARD

Directeur de recherche, GIPSA-lab – CNRS, Président du jury

Isabelle FANTONI

Directeur de recherche, Heudiasyc – CNRS, Rapporteur

Nacim RAMDANI

Professeur, PRISME – Université d'Orléans, Rapporteur

Alexandre SEURET

Chargé de recherche, LAAS – CNRS, Examineur

Tobias GURDAN

Ingénieur, Intel Allemagne, Examineur

José Fermi GUERRERO-CASTELLANOS

Professeur, Benemérita Universidad Autónoma de Puebla, Examineur

Sylvain DURAND

MCF, ICUBE – INSA Strasbourg, Invité

Nicolas MARCHAND

Directeur de recherche, GIPSA-lab – CNRS, Directeur de thèse

John-Jairo MARTINEZ-MOLINA

MCF, GIPSA-lab – Grenoble-INP, Codirecteur de thèse

Thibaut RAHARIJAONA

MCF, ISM – Aix-Marseille Université, Coencadrant de thèse



UNIVERSITÉ GRENOBLE ALPES
ÉCOLE DOCTORALE EEATS
Electronique, Electrotechnique, Automatique, Traitement du signal

T H È S E

pour obtenir le titre de

docteur en sciences

de l'Université Grenoble Alpes

Mention : Automatique et Productique

Présentée et soutenue par
Bruno BOISSEAU

Event-based control: application to robotic systems

Thèse dirigée par **Nicolas MARCHAND**,
codirigée par **John-Jairo MARTINEZ-MOLINA**,
et coencadrée par **Thibaut RAHARIJAONA**

préparée au laboratoire GIPSA-lab, département automatique
soutenue le **30 juin 2017**

Jury :

<i>Président :</i>	Luc DUGARD	- Directeur de recherche, GIPSA-lab – CNRS
<i>Rapporteurs :</i>	Isabelle FANTONI	- Directeur de recherche, Heudiasyc – CNRS
	Nacim RAMDANI	- Professeur, PRISME – Université d'Orléans
<i>Examineurs :</i>	Alexandre SEURET	- Chargé de recherche, LAAS – CNRS
	Tobias GURDAN	- Ingénieur, Intel Allemagne
	José Fermi GUERRERO	- Professeur, Benemérita Universidad Autónoma de Puebla
<i>Invité :</i>	Sylvain DURAND	- MCF, ICUBE – INSA Strasbourg
<i>Directeur :</i>	Nicolas MARCHAND	- Directeur de recherche, GIPSA-lab – CNRS
<i>Codirecteur :</i>	John-Jairo MARTINEZ	- MCF, GIPSA-lab – Grenoble-INP
<i>Coencadrant :</i>	Thibaut RAHARIJAONA	- MCF, ISM – Aix-Marseille Université

Remerciements

Je tiens tout d'abord à remercier mes directeurs de thèse pour m'avoir fait confiance et permis de travailler en autonomie tout en ayant m'épaulant lorsque le besoin s'en faisait ressentir. L'automatique étant un domaine dans lequel mes connaissances étaient quasi-inexistantes avant le début de ma thèse, la confiance et la patience dont ils ont fait preuve à mon égard a été très appréciable.

Je tiens également à remercier les différents membres du jury pour avoir accepté de participer à l'évaluation de mon travail.

Mes remerciements vont ensuite à tous les doctorants du Gipsa, l'ambiance détendue qui règne dans ce laboratoire a fait que je me sentais un peu comme à la maison. Je tiens en particulier à remercier Swann pour toutes les longues pauses qu'il m'a fait prendre et les discussions (quelquefois) intéressantes que nous avons partagées. Nassim pour son humour sans limites, Raouia pour sa joie de vivre incessante, et Insaf pour sa générosité. Nicolas pour sa gentillesse et les discussions que l'on a partagées. Josué, Juan et Jonatan pour leur soutien, leur bonne humeur, la culture qu'ils m'ont fait découvrir et tous les cours de Mexicain qu'ils m'ont prodigués. Etienne avec qui j'ai eu la joie de travailler tout en échangeant certaines recettes de bonnes purées. Vincent avec qui j'ai pris beaucoup de plaisir à collaborer et à qui je souhaite toute la réussite possible pour son beau projet. Matthieu pour toutes les discussions fructueuses et avisées mais surtout pour son enthousiasme. François pour sa bonne humeur continuelle et tout le stock de gâteaux qu'il a partagé. Maëlle pour son soutien, spécialement en période de rédaction. Diego et Nadia pour leur gentillesse sans faille. Bojan pour m'avoir convaincu de l'existence de l'aléatoire véritable à travers ses conversations.

Merci également à mes tout premiers collègues de bureau pour leur bienveillance à mon égard : Tahar pour tous les débats sans fin, Waleed pour le fait qu'il soit le meilleur, Mihaly pour son humour et également (parfois) son sérieux. Je tiens particulièrement à remercier mes derniers collègues de bureau pour m'avoir supporté pendant ma période de rédaction : Miguel (dit *Mich-Mich*) pour sa gentillesse et ses connaissances scientifiques qui n'ont d'égal que son manque de volonté à apprendre notre belle langue, et Yashank pour sa bonne humeur et pour m'avoir souvent fait rire.

Merci également à toutes les personnes que j'ai eu la chance de côtoyer lors de mes séjours à l'atelier, Jonathan pour les nombreuses discussions techniques, Julien pour les conseils de sioux en électronique, Pascal pour le développement informatique, et toutes les autres personnes que j'ai pu croiser et qui m'ont beaucoup appris.

Un grand merci à Patricia pour sa présence bienveillante, en particulier lors du déjeuner. Merci à Sonia et Fany chez qui l'ambiance était toujours détendue et rafraichissante.

Je tiens également à remercier tous les membres du département Biorobotique à Marseille, en particulier les doctorants que j'ai pu y croiser : Augustin qui m'a beaucoup appris en ayant la patience de répondre à mon flot ininterrompu de question et avec qui j'ai pris beaucoup de plaisir à travailler quelquefois tard dans la nuit, malgré la charge de travail, je garde un excellent souvenir de cette période. Fabien (le gentil) pour les discussions enrichissantes que nous avons partagées (et la porte cassée). Stefano pour sa gentillesse et sa passion qui n'ont d'égal que son amour des discussions endiablées. Roman (dit *l'ingénieur*) pour son amour des mouches et ses connaissances approfondies des fruits et légumes. Erik que je n'ai malheureusement pas croisé assez à mon goût mais avec qui j'ai pris beaucoup de plaisir à échanger.

Un grand merci également à Marc pour toute l'aide apportée et son calme légendaire. Je tiens également à exprimer ma gratitude à Julien dont il est difficile de décrire sa bonne humeur communicative avec de simples mots.

Ma reconnaissance va ensuite à ma famille, mes parents qui ont toujours été présents tout au long de mes études et sans qui je n'aurais jamais pu en arriver là, ma soeur et son mari qui m'ont toujours soutenu, le simple fait de les voir me redonnait toujours le moral. Bien sûr je tiens à remercier Célia, son rire n'a jamais failli à me redonner le sourire. Merci également à ma belle-famille, en particulier Sylvie et Valter qui ont toujours été présents, les soirées sur leur terrasse m'ont toujours fait oublier le stress et les doutes qu'apportent inexorablement l'accomplissement d'un doctorat.

Et enfin je tiens à remercier Noëlie pour son soutien inconditionnel, ses encouragements, son enthousiasme, sa patience, sa douceur et sa folie. En bref, merci d'être toi.

Acronyms

CETC	Continuous Event-Triggered Controller
DARE	Discrete Algebraic Riccati Equation
ECP	Educational Control Products
ESC	Electronic Speed Controller
FCB	Flight Controller Board
IAE	Integral Absolute Error
IAU	Integral Absolute control signal
IMU	Inertial Measurement Unit
ISS	Input to State Stable
LQ	Linear Quadratic
LTI	Linear Time Invariant
MIMO	Multi Input Multi Output
MOCA	Motion Capture Area
MPI	Maximum Positively Invariant
mRPI	minimal Robustly Positively Invariant
MSI	Minimum Sampling Interval
MSP	Multiwii Serial Protocol
PETC	Periodic Event-Triggered Controller
PID	Proportional Integral Derivative
PPM	Pulse Position Modulation
PRBS	Pseudo Random Binary Sequence
PSD	Power Spectral Density
PWL	PieceWise Linear

PWM	Pulse Width Modulation
RCB	Rotors Control Board
RPI	Robustly Positively Invariant
SISO	Single Input Single Output
SOD	Send-On-Delta
TACO	Thrust Active Control
UAV	Unmanned Aerial Vehicle
UB	Uniformly Ultimately Bounded
VTOL	Vertical Take Off and Landing
ZOH	Zero Order Hold

Contents

Acronyms	iii
Introduction	1
1 Background and definitions	7
2 Related work	13
2.1 Event-based PID controllers	13
2.2 Lyapunov derived techniques	14
2.2.1 Lyapunov sampling	14
2.2.2 Lyapunov derivative approach	16
2.3 Input-to-State stability approach	16
2.4 Ultimate bound approach	17
2.5 Self-triggering	17
3 Discrete time event-based controller	19
3.1 Event function design	21
3.2 A simple example	22
3.2.1 Performance indexes	23
3.2.2 Simulation results	23
3.3 Implementation on a real-time system	25
3.3.1 Description of the system	26
3.3.2 Angular positions and torques	26
3.3.3 Dynamic model	27
3.3.4 Time-triggered control design	29

3.3.5	Introducing a reference signal	29
3.3.6	Observer design	30
3.3.7	Disturbance cancellation	31
3.3.8	Event function	31
3.4	Real-time experimental results	32
3.4.1	Scenario	32
3.4.2	Obtained results	32
3.5	Discussion	35
4	Event-switched approach	37
4.1	Polyhedral RPI sets for characterizing nominal performances	38
4.2	Event-switched system	39
4.3	Event-set computation	40
4.3.1	Reducing the complexity of the event-set	42
4.4	Event-switched control design with guaranteed performances	42
4.4.1	Main results	42
4.4.2	Analytic discussion of performances	44
4.4.3	Observer based approach	46
4.4.4	Proposed algorithm	47
4.5	Simulation of the event-switched approach	50
4.5.1	Performance indexes	52
4.5.2	Results	53
4.5.3	Reducing computation complexity	55
5	Experimental setup	57
5.1	Objectives	57

5.2	Quadcopter	58
5.2.1	Euler formalism	58
5.2.2	Quadcopter modeling	61
5.3	Flying arena and experimental setup	63
5.3.1	Position sensor	64
5.3.2	Control toolchain	65
5.3.3	Onboard control	66
5.4	Quadcopter setup	66
5.4.1	The X4-MaG platform	66
5.4.2	The Flexbot platform	68
5.4.3	The AVE platform	69
5.4.4	The Inductrix platform	69
5.4.5	The TACO platform	71
5.4.6	Motor control	71
5.5	Platform choice	74
6	Application of the event-switched approach	77
6.1	Objectives	77
6.2	Preliminar steps	79
6.2.1	Position controller design	79
6.2.2	Position controller implementation	82
6.2.3	Identification	84
6.3	User assistance	86
6.4	Event-switching for constrained states	88
6.4.1	Problem statement	88
6.4.2	Definition of the constraint set	88

6.4.3	Event-set design	90
6.5	Experimental results	91
6.6	Improving identification	93
6.7	Summary	97
7	Conclusions and perspectives	101
7.1	Conclusions	101
7.2	Perspectives	102
	Bibliography	105

List of Figures

1	Comparison of sampling mechanisms.	2
2	Different schemes of control.	3
1.1	Non-convex and Convex 2-dimensional sets.	8
1.2	Stability illustration.	10
2.1	Lyapunov sampling principle.	15
3.1	Comparison of update policy (step input).	24
3.2	Comparison of update policy (transient response).	24
3.3	Performance indexes for different update mechanisms.	25
3.4	ECP gyroscope used as a real-time platform.	26
3.5	Experimental results: time-triggered strategy	33
3.6	Experimental results: event-triggered strategy with disturbances rejection	34
3.7	Experimental results: disturbances estimation	35
4.1	Scheme of event-switched control strategy.	48
4.2	Geometrical interpretation of the event-switched control strategy.	49
4.3	Simulation of a time-triggered system and an even-switched system.	51
4.4	Worst disturbance scenario for mRPI approximation (\mathcal{S}_1) verification.	51
4.5	Comparison between time-triggered and event-switched systems subjected to the same uniformly distributed disturbances.	54
4.6	Power spectral density comparison between time-triggered and event-switched systems.	54
4.7	The set \mathcal{E}^* is the maximum volume ellipse included in \mathcal{S}_2^*	56
5.1	Schematic view of a quadcopter.	59

5.2	Euler angles illustration.	59
5.3	Schematic view of the control process of a human piloted UAV and an automatic flight system using Vicon system.	64
5.4	MOCA flying arena at GIPSA-lab.	65
5.5	3D printed frame of the flexbot quadrotor and the flight controller board.	68
5.6	Experimental AVE platform.	70
5.7	Inductrix quadcopter.	70
5.8	Emax frame of the TACO Platform.	71
5.9	Setup of the test bench for the couple ESC/motor.	73
5.10	Profile input for the tuning of the motor control loop.	73
5.11	Measured speed for different tuning parameters.	74
5.12	Tuning of the ESC's gain.	75
6.1	Schematic view of the collision avoidance approach used here.	79
6.2	Diagram of the position control process	83
6.3	Experimental step response along x axis in closed-loop.	84
6.4	Experimental response to special procedures0 designed for take off and landing.	84
6.5	Identification of the closed-loop system.	85
6.6	Diagram of the speed control process.	87
6.7	Experimental step response of the speed along x axis in closed loop.	87
6.8	Constraint set \mathcal{X} is represented in red, and the MPI set \mathcal{P} in yellow.	89
6.9	The MPI set \mathcal{P} is represented in yellow, the event-set \mathcal{S} in green, and its ellipsoidal approximation \mathcal{E} in red stripes.	91
6.10	Altitude of the UAV.	92
6.11	Trajectory of the UAV in the (x, y) plane.	93
6.12	Recorded data of a flight using event-switched approach to keep the quadcopter in a $1.5m$ square.	94

6.13 An event is triggered once any of the state X or Y leave the event-set \mathcal{E} (in green).	95
6.14 Trajectory of the UAV in the (x, y) plane.	95
6.15 Trajectory of the state along x axis.	96
6.16 PRBS inputs for identification.	97
6.17 Identification data and simulation of the identified model.	98

List of Tables

3.1	Mechanical constants of the gyroscope actuator.	27
3.2	Performance indexes obtained with different control strategies	32
4.1	Theoretical bounds on performance indexes.	52
4.2	Performance indexes obtained with uniformly distributed disturbances. . .	52
4.3	Performance indexes obtained with truncated normally distributed distur- bances.	53
5.1	Comparison of the different platforms.	75
6.1	Tuning of the different discrete controllers used for the position control. Sampling time $T_s = 10\text{ms}$	82

Introduction

General context

Control system theory has first been built for analog controllers. In this context, it was relevant to use a continuous framework to design a control feedback function. Nowadays, digital technologies are supplanting analog solutions due to several advantages (cost, noise resistance, integration...). ZOH¹ maintains a control signal constant between periodic triggers given by a digital clock. Periodic controllers are also referred as time-triggered because the update only depends on time. A continuous time system is usually discretized in time domain and the synthesis of a digital controller is done through discrete time framework. Many methods have been developed to design a controller for discrete systems (see for instance [Ogata, 1987]). Nevertheless, a discretized model is not always available for nonlinear continuous systems. A typical procedure consists in designing the controller in the continuous time domain, and use a discrete time version with a sufficiently small sampling period. However, digital components can be incompatible with too high sampling rate, then desirable closed-loop performances can not be achieved. Another disadvantage of discrete-time control is the amount of communication required to perform closed-loop control in case of networked control systems (NCS), for example when the controller and/or the actuator are connected through a network. In that case, periodic control framework requires a constant communication bandwidth to update the informations periodically. One solution to decrease the bandwidth consumption is to increase the sampling period, however performances can decrease and a bottleneck is facing when the sampling period is too high for the dynamics of the system.

Instead of sampling a signal equidistantly in time, it is possible to sample depending on the signal's level. Such a concept is close to the concept of Riemann and Lebesgues integral and was discussed in [Åström and Bernhardsson, 2002; Marchand, 2008]. A comparison of a continuous signal sampled with these two different approaches is depicted in Figure 1. The latter one can result in a moderation of update of the control signal to achieve the same performance as the time-triggered system [Åström and Bernhardsson, 2002]. Event-based (or event-triggered) control aims to improve the periodic sampling scheme by proposing a method in which updates are triggered by an event function. With classical time-triggered approaches, the control law is computed and the control signal is updated at a fixed sampling period whether this is really necessary or not. Conversely, event-based procedures do not require these periodic computations and updates, but call for resources only when they are strictly necessary. In other words, time-triggered controllers decide *how* to actuate *periodically*, while event-triggered controllers decide not only *how* but also *when* to actuate. As a consequence, event-based control refers to a set of two functions:

¹Zero Order Hold

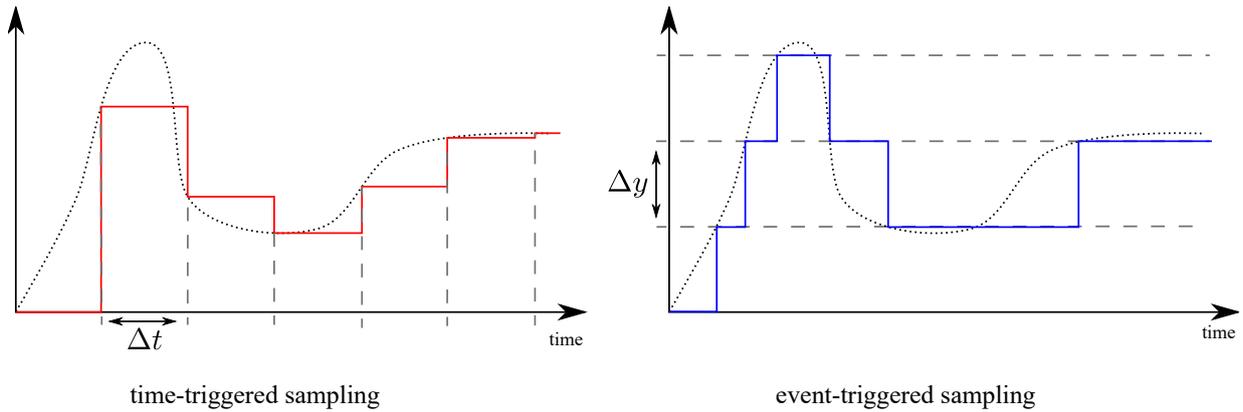


Figure 1: Comparison of sampling mechanisms. Dotted line shows the continuous signal, the time-triggered and the event-triggered sampled signals are represented in solid red and blue lines respectively.

1. A feedback function
2. An event function which will indicate if the control signal needs to be updated.

However, implementing a continuous event function and feedback is difficult as most of electronic nowadays relies on digital component. One solution is to check the event condition periodically, this has been called PETC² in [Heemels et al., 2008]. In this case, the sensors are sampled at every discrete instants $t_i = i \times T_s$, with $i \in \mathbb{N}$ and $T_s \in \mathbb{R}^+$ the sampling period, and a criterion is evaluated to check if the controller will be also updated. Approaches which consider a continuous event function will be referred as CETC³. Figure 2 depicts several usual controller forms.

Thesis contributions

The main theoretical results of the thesis are the following:

- The first contribution concerns the extension of an event-based controller proposed in [Marchand et al., 2013] to discrete time linear systems. This contribution has been published in [Boisseau et al., 2015].
- The second contribution is to provide a theoretical framework to design a novel and innovative event-switched controller for discrete linear systems with disturbances. This contribution has been published in [Boisseau et al., 2016].

²Periodic Event-Triggered Controller

³Continuous Event-Triggered Controller

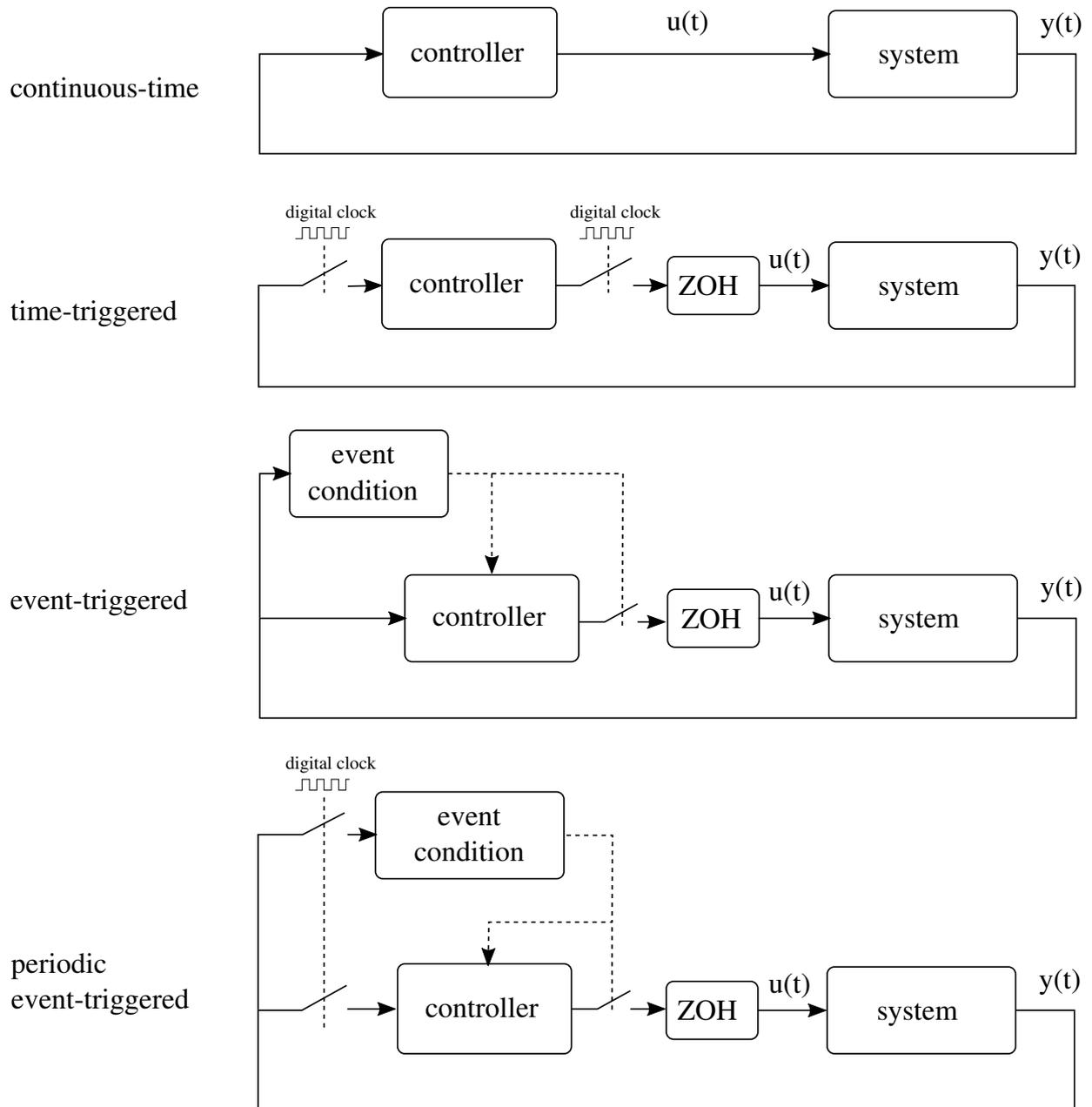


Figure 2: Different schemes of control. Dotted lines show a trigger signal.

The main practical results of the thesis are the following:

- Discrete event-based control of a gyroscope actuator;
- Event-switched approach for human control of quadcopter in a constrained environment.

Publications

The following contributions have been published during this thesis:

- International conferences
 - “Event-based speed control on a sensor-less miniature thruster” T. Raharijaona, L. Dola, B. Boisseau, JJ. Martinez-Molina, N. Marchand, F. Ruffier. IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), July 2014, Besançon, France.
 - “Event-based LQR with integral action” S. Durand, B. Boisseau, JJ. Martinez-Molina, N. Marchand, T. Raharijaona. IEEE conference on Emerging Technology and Factory Automation (ETFA), September 2014, Barcelona, Spain.
 - “Attitude control of a gyroscope actuator using event-based discrete-time approach” B. Boisseau, S. Durand, JJ. Martinez-Molina, T. Raharijaona, N. Marchand. International Conference on Event-based Control, Communication, and Signal Processing (EBCCS), June 2015, Krakow, Poland. **Best paper award**
- Journal paper
 - “Event-switched control design with guaranteed performances” B. Boisseau, JJ. Martinez-Molina, T. Raharijaona, S. Durand, N. Marchand. International Journal of Robust and Nonlinear Control. *Accepted in September 2016 - To appear*

Thesis outline

This thesis is organized in four main parts. The first chapter is dedicated to introduce the concepts used in the manuscript. The second chapter establishes the context of the present work in event-based control through an analysis of the literature. The third chapter details the design of a discrete event-based controller and its implementation on a real-time system. The fourth chapter introduces event-switched approach, which contains the main theoretical result of this thesis. Chapter five presents several quadcopters used as

experimental platforms which have been developed during my PhD. Chapter six shows the application of the event-switched approach for collision avoidance between an aerial vehicle and a static environment. Finally chapter seven draws the conclusions and presents further ideas for future work.

Background and definitions

Nomenclature

\mathbb{R} denotes the set of real numbers while \mathbb{R}_+ denotes the set of positive real numbers. \mathbb{N} denotes the set of positive integers, and \mathbb{Z} is used for the set of integers. $A^T \in \mathbb{R}^{m \times n}$ denotes the transpose of the matrix $A \in \mathbb{R}^{n \times m}$. The absolute value of x is denoted $|x|$. \dot{x} denotes the time derivative of x . The union of sets is denoted \cup , and \cap denotes the intersection of sets. The empty set is denoted \emptyset .

Definitions

Definition 1.1

A set \mathcal{S} is said to be convex if by taking any two points in \mathcal{S} , the obtained line belongs to \mathcal{S} . Figure 1.1 depicts a non-convex and a convex 2-dimensional sets. This condition can be expressed by:

$$\alpha x_1 + (1 - \alpha)x_2 \in \mathcal{S}, \quad \forall \alpha \in [0, 1] \text{ and } \forall x_1, x_2 \in \mathcal{S}$$

Definition 1.2

The convex hull of a set \mathcal{S} , denoted by $\text{conv}(\mathcal{S})$, is the smallest convex set that contains \mathcal{S} .

Definition 1.3

The Minkowski sum of $\mathcal{A} \subset \mathbb{R}^n$ and $\mathcal{B} \subset \mathbb{R}^n$, which will be denoted $\mathcal{A} \oplus \mathcal{B}$, is defined as follows: $\mathcal{A} \oplus \mathcal{B} = \{a + b \mid a \in \mathcal{A}, b \in \mathcal{B}\}$.

Definition 1.4

The relative complement of $\mathcal{A} \subset \mathbb{R}^n$ in $\mathcal{B} \subset \mathbb{R}^n$, which will be denoted $\mathcal{B} \setminus \mathcal{A}$, is defined as follows: $\mathcal{B} \setminus \mathcal{A} = \{x \in \mathcal{B} \mid x \notin \mathcal{A}\}$. Obviously $(\mathcal{A} \cup \mathcal{B}) \setminus \mathcal{A} = \mathcal{B}$ and $(\mathcal{A} \cap \mathcal{B}) \setminus \mathcal{A} = (\mathcal{A} \cap \mathcal{B}) \setminus \mathcal{B} = \emptyset$.

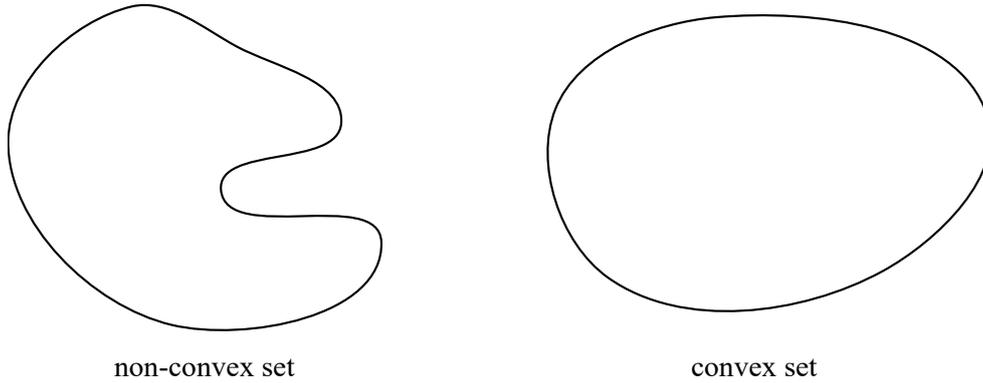


Figure 1.1: Non-convex and Convex 2-dimensional sets.

Definition 1.5

The p -norm of a vector $x \in \mathbb{R}^n$ is defined as:

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}$$

The 2-norm is also called the euclidean norm and will be denoted $\|x\| = \sqrt{x^T x}$.

Stability definitions

The aim of this section is to briefly introduce the concept of stability which is widely spread in control systems theory and will be used in the next chapters. Details and proofs are not provided here but can be found in standard books (e.g. [Khalil, 2002], [Astrom and Murray, 2008]).

Consider the following system:

$$\dot{x}(t) = f(x(t), u(t)) \quad (1.1)$$

where $x(t) \in \mathbb{R}^n$ denotes the state of the system, $u(t) \in \mathbb{R}^m$ represents the control input, and $f : \mathcal{D} \mapsto \mathbb{R}^n$ represents a locally Lipschitz function, with $\mathcal{D} \subseteq \mathbb{R}^n$.

An autonomous system will denote any system without input:

$$\dot{x}(t) = f(x(t)) \quad (1.2)$$

where $x(t) \in \mathbb{R}^n$ denotes the state of the system, and $f : \mathcal{D} \mapsto \mathbb{R}^n$ represents a locally Lipschitz function, with $\mathcal{D} \subseteq \mathbb{R}^n$.

Equilibrium

Definition 1.6

x_e is an equilibrium point for continuous time autonomous system (1.2) if $f(x_e) = 0$.

The concept of stability of an equilibrium point x_e of system (1.2) determines if solutions nearby x_e remain close, get closer or move further away.

If an equilibrium point x_e exists, it is always possible to perform a coordinate change $\tilde{x}(t) = x(t) - x_e$ so that the origin is an equilibrium for the new system. Therefore, stability of the origin will be discussed in the sequel without loss of generality.

Lyapunov stability

Definition 1.7

The equilibrium point $x_e = 0$ is a stable equilibrium point for system (1.2) if for each $\epsilon > 0$, there exist $\delta > 0$ such that:

$$\|x(0)\| < \delta \implies \|x(t)\| < \epsilon, \quad \forall t \geq 0$$

Theorem 1.1 (From [Khalil, 2002])

If $x = 0$ is an equilibrium point, Lyapunov theory states that the origin is stable for system (1.2) if a continuously differentiable function $V : \mathcal{D} \mapsto \mathbb{R}$, with $0 \in \mathcal{D} \subseteq \mathbb{R}^n$ exists such that all the following conditions are satisfied:

$$\begin{aligned} V(0) &= 0 \\ V(x) &> 0, \quad \forall x \in (\mathcal{D} \setminus \{0\}) \\ \dot{V}(x) &\leq 0, \quad \forall x \in \mathcal{D} \end{aligned}$$

V can be seen as an energy function of the system, if the energy is always decreasing, then the origin can be considered as stable.

In the sequel, a system will be called stable if the origin is a stable equilibrium point for this system.

Asymptotic stability

The equilibrium point $x = 0$ of system (1.2) is said to be asymptotically stable if for any initial condition near the origin, the solution of the system will converge exactly to the origin.

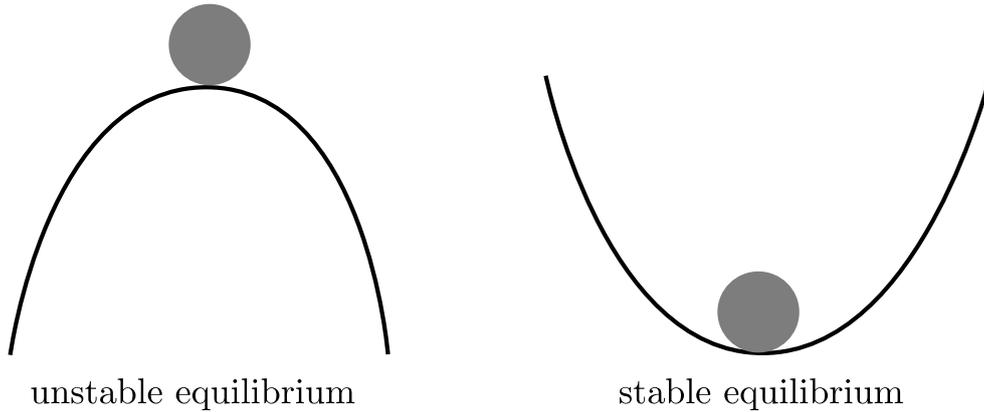


Figure 1.2: Stability illustration.

Theorem 1.2 (From [Khalil, 2002])

The equilibrium point $x = 0$ is asymptotically stable for (1.2) if a continuously differentiable function $V : \mathcal{D} \mapsto \mathbb{R}$, with $0 \in \mathcal{D} \subseteq \mathbb{R}^n$ exists such that all the conditions of Theorem 1.1 are satisfied, and moreover:

$$\dot{V}(x) < 0, \quad \forall x \in (\mathcal{D} \setminus \{0\})$$

Lyapunov's theory has been first designed to handle the problem of stability regarding motion (see [Lyapunov, 1992] for an english translation). The concept of stability is more intuitive for mechanical systems, as depicted in Figure 1.2 for a marble on a surface. If the friction is not considered, then the stable equilibrium is not asymptotically stable, however if the friction is taken into account, the equilibrium point is asymptotically stable.

Global asymptotic stability

When an equilibrium point x_e is asymptotically stable for system (1.2), the domain (or basin) of attraction defines all the initial conditions which will make solutions of system (1.2) converge to x_e . An equilibrium point is said globally asymptotically stable for system (1.2) if for any initial condition in \mathbb{R}^n , the solution will converge to x_e . In other words, when the basin of attraction is \mathbb{R}^n .

Theorem 1.3 (From [Khalil, 2002])

The equilibrium point $x = 0$ is said globally asymptotically stable for system (1.2) if a continuously differentiable function $V : \mathbb{R}^n \mapsto \mathbb{R}$ exists such that the following conditions

are satisfied:

$$\begin{aligned} V(0) &= 0 \\ V(x) &> 0, \quad \forall x \neq 0 \\ \|x\| \rightarrow \infty &\implies V(x) \rightarrow \infty \\ \dot{V}(x) &\leq 0, \quad \forall x \neq 0 \end{aligned}$$

The stable equilibrium point in Figure 1.2 is globally asymptotically stable if there is friction.

Linear systems

Continuous LTI¹: Given an initial condition $x(0)$, a continuous LTI system is fully described by the following state space representation:

$$\begin{aligned} \dot{x} &= Ax(t) + Bu(t) \\ y &= Cx(t) + Du(t) \end{aligned}$$

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{q \times n}$, $D \in \mathbb{R}^{q \times m}$, and where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the input vector, and $y \in \mathbb{R}^q$ is the output vector. This system will be referred as an autonomous system if $B = 0$ and $D = 0$.

The origin of an autonomous continuous LTI system $\dot{x} = Ax(t)$ is globally asymptotically stable if and only if the matrix A is Hurwitz, meaning that all the eigenvalues of A have a strictly negative real part. Moreover, according to Lyapunov's theory, an autonomous continuous LTI system is stable if and only if there exists a strictly positive definite matrix P such that: $A^T P + PA < 0$. A Lyapunov function of the system is described by: $V = x^T P x$.

Discrete LTI: Given an initial condition x_0 , a discrete LTI system is fully described by the following state space representation:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned}$$

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{q \times n}$, $D \in \mathbb{R}^{q \times m}$, and where $x_k \in \mathbb{R}^n$ is the state vector, $u_k \in \mathbb{R}^m$ is the input vector, and $y_k \in \mathbb{R}^q$ is the output vector. This system will be referred as an autonomous system if $B = 0$ and $D = 0$.

The origin of an autonomous discrete LTI system is globally asymptotically stable if and only if all the eigenvalues of A are strictly contained in the unit circle. Moreover, an

¹Linear Time Invariant

autonomous discrete LTI system is stable if there exists a strictly positive definite matrix P such that: $A^T P A - P < 0$.

Discrete disturbed LTI: We consider now a discrete LTI system subject to an exogenous unknown disturbance $w_k \in \mathbb{R}^v$:

$$\begin{aligned}x_{k+1} &= Ax_k + Ew_k \\y_k &= Cx_k\end{aligned}$$

with $E \in \mathbb{R}^{n \times v}$. Obviously this system does not converge to the origin if w_k is not equal to 0. Rather than studying the convergence to a point, convergence to a set will be of interest.

Definition 1.8 (From [Blanchini, 1999])

A set $\mathcal{S} \subseteq \mathbb{R}^n$ is said to be a RPI² set for a discrete time autonomous system $x_{k+1} = f(x_k, w_k)$, if for any initial condition x_0 in \mathcal{S} and any disturbances w_k in \mathcal{W} , x_k will belong to \mathcal{S} for all instants $k \in \mathbb{N}^+$.

Definition 1.9 (From [Blanchini, 1994])

A discrete time autonomous system $x_{k+1} = f(x_k, w_k)$ is said to be Uniformly Ultimately Bounded (UB) to the set $\mathcal{S} \subseteq \mathbb{R}^n$ if for all initial conditions $x_0 \in \mathbb{R}^n$ and any disturbances w_k in \mathcal{W} , there exists a positive instant $T(x_0)$ such that x_k remains in \mathcal{S} for all $k \geq T(x_0)$.

Definition 1.10 (From [Rakovic et al., 2005])

A set $\mathcal{S} \subseteq \mathbb{R}^n$ is said to be the mRPI³ set for a discrete time autonomous system $x_{k+1} = f(x_k, w_k)$ if \mathcal{S} is a RPI set for this system, and if \mathcal{S} is included in all possible closed RPI sets for this system.

²Robustly Positively Invariant

³minimal Robustly Positively Invariant

Related work

Contents

2.1	Event-based PID controllers	13
2.2	Lyapunov derived techniques	14
2.2.1	Lyapunov sampling	14
2.2.2	Lyapunov derivative approach	16
2.3	Input-to-State stability approach	16
2.4	Ultimate bound approach	17
2.5	Self-triggering	17

This chapter aims to determine the current challenges in event-based control and the proposed solutions by analyzing the literature. Designing an event-based control loop is often done in two consecutive steps:

- Find a feedback function which makes the system stable.
- Compute an event function which guarantees the performances of the event-triggered system.

Methods to find a state feedback gain for the system are well-known and will not be detailed here.

2.1 Event-based PID controllers

For SISO¹ systems, PID² controllers are widespread in industry. The implementation is well-known, the tuning is intuitive and some systematic methods can be performed (see

¹Single Input Single Output

²Proportional Integral Derivative

[Ziegler and Nichols, 1942] for a well-known tuning rule or [Skogestad, 2001] for a more recent one). In [Årzén, 1999], an event-based PID controller and simulation results are presented. Each time the difference between the current error (between the output and a given reference) and the error at the last event reaches a certain threshold, the control signal is computed and updated. The comparison between error and threshold is checked in a PETC fashion, which makes it more realistic for implementation purpose. However the integral part can lead to important overshoot. To overcome this issue, an exponential forgetting factor is proposed in [Durand and Marchand, 2009]. Simulations show that it is possible to stabilize a system with less update compared to a time-triggered PID. Even if it is not possible to achieve asymptotic stability, [Sande et al., 2005] shows the practical stability of the event-based PID. In [Sánchez et al., 2009] some other event conditions have been investigated and compared on a real-time system, the comparison has been conducted by using some performances indicators which will be introduced and used in the sequel.

However, for high order or MIMO³ systems, this approach is hardly extensible. Several techniques are presented in the sequel, which can be extended to a more general class of systems.

2.2 Lyapunov derived techniques

2.2.1 Lyapunov sampling

To guarantee the stability, the event function should trigger an event when the performance of the system is not satisfactory. One way to quantify the performances is to consider a Lyapunov function of the system. A Lyapunov function can be seen as a function of the energy of the system. Hence its value should decrease over time for a stable system. The method called *Lyapunov sampling*, ensures that a Lyapunov function of the event-triggered system is decreasing, and so the stability of the system. In [Velasco et al., 2009] it is proposed to trigger an event (and so to enforce an update of the control signal) based on some level crossing of the state x with some contour curves build upon a Lyapunov function $V(x)$. Figure 2.1 depicts this mechanism for a 2 dimensional system. Each time the trajectory of the state x crosses a sampling curve from the outside to the inside, an event is triggered and the control signal is updated. Usually the Lyapunov function is described by a quadratic function of the state, then the contour curves are ellipsoids of constant energy in the phase plan. The contour curves are built upon the relation:

$$V(x_{i+1}) = \eta V(x_i) \quad (2.1)$$

with $i \in \mathbb{N}$, $0 < \eta < 1$, the subscript denotes the discrete event update instant.

³Multi Input Multi Output

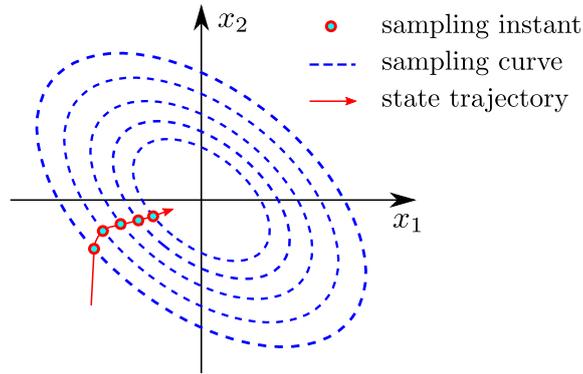


Figure 2.1: Lyapunov sampling principle for a second order system. Everytime the state's trajectory is crossing a sampling curve, a control update is triggered.

The value of η has to be calculated in order to guarantee to never stop triggering events. The event-based system is ensured to be stable only if an infinite number of events occurs, which means that the Lyapunov function should always be decreasing. However, calculating a lower bound on the η value requires heavy offline computation, because a non-convex optimization has to be solved even for linear systems. An extension of this method has been proposed in [Durand et al., 2011]. It provides an event condition which does not require offline computation, the event condition is constructed and checked online. The main difference with the work presented in [Velasco et al., 2009] is that the parameter η in (2.1) is not fixed anymore but is adapted over time. Practical experimentation on a DC servo-motor leads to a reduction of about 90% of samples compared to the time-triggered implementation while achieving almost the same performances.

However, the main issue is that the time between two events keeps reducing as the state converges to the origin. A challenge when designing a CETC is to ensure that the resulting system is stable but also implementable. The minimum time between two events has to be bounded. This value is also called MSI⁴. If this value is not bounded, Zeno phenomenon is not guaranteed to be avoided. Zeno phenomenon consists in an accumulation of an infinite number of events in a finite time, therefore the implementation on a digital controller can not be done because the sampling rate is limited. If a strictly positive lower bound on the MSI exists for the event-triggered system, then it is possible to check the event condition periodically. Stability is necessarily ensured if the sampling period is lower than the MSI time bound.

⁴Minimum Sampling Interval

2.2.2 Lyapunov derivative approach

In [Marchand et al., 2013], a CETC mechanism is proposed for nonlinear systems affine on the control of the form:

$$\dot{x} = f(x) + g(x)u$$

A stabilizing controller $u = k(x)$ is found by extension of a seminal method proposed in [Sontag, 1989]. Given a Lyapunov function V , the event function is built to compare the derivative of the Lyapunov function if the control is held constant versus if the control is updated. An event is triggered as soon as $\dot{\bar{V}} \leq \sigma \dot{V}$, where \bar{V} denotes the value of the Lyapunov function if the control is kept constant, and $0 < \sigma < 1$ is a degree of freedom to tune the performances of the resulting event-based system. The event function forces an update whereas holding the control value does not guarantee a sufficient rate of decay of the Lyapunov function. The event-triggered system is proved to have a lower bounded MSI for a set of initial conditions. Moreover the control is ensured to be continuous at the origin. Time delay for this class of systems has been considered in [Durand et al., 2014b]. An extension for linear systems is proposed in [Durand et al., 2014a] where the strategy is tested on a real-time platform, and where the continuous feedback is designed using optimal control theory.

Chapter 3 proposes an extension of this method for discrete time linear systems.

2.3 Input-to-State stability approach

In [Tabuada, 2007], a CETC scheme is proposed for nonlinear systems of the form:

$$\dot{x} = f(x, u) \tag{2.2}$$

where $x \in \mathbb{R}^n$ represents the state of the system, $u \in \mathbb{R}^m$ represents the control input, and a function $f : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$.

To construct a stable event mechanism for the nonlinear system (2.2), a feedback controller $u = k(x)$ which renders the closed-loop system ISS⁵ is needed. Moreover, the control feedback function k is required to be Lipschitz at the origin. Unlike the above method in Section 2.2.2, the event function does not depend only on the current value of the state, but also on its past value. More specifically, an update of the controller is triggered based on the value of the euclidean norm of the error e between the actual state and its value at the last event. In other words, an event is triggered when the state has evolved “too much” since the last actuation. The event condition is built such that the resulting event-triggered system is ISS. With this approach, it is also possible to consider a (small constant) delay in the actuation chain. The resulting event-based system is ensured to have a lower bound

⁵Input to State Stable

on the inter event time for any bounded initial condition, which makes this system immune to Zeno phenomenon.

2.4 Ultimate bound approach

In [Heemels et al., 2008], the PETC mechanism for perturbed linear systems is introduced as well as the notion of *uniform* and *non-uniform* mechanisms. In the *uniform* mechanism, the event can only be triggered periodically, whereas in the *non-uniform* case, the event is triggered as soon as the event-condition is satisfied and a minimum time has passed since the last event. The latter is like CETC with a minimum time interval between events. Note that Zeno phenomenon can not happen in both cases because a MSI between events is considered in the design. The event condition is described in a geometric form, an event is triggered if the state does not belong to a given set. The *non-uniform* mechanism is supposed to detect the exact crossing of the state with the boundary of a set. Stability analysis is done using a PWL⁶ and an impulsive model of the continuous plant along with the event-based controller. The resulting event-based system is shown to be ultimately bounded in a closed set. Another method based on ultimate bound is also proposed in [Grüne et al., 2010], where an event condition based on the set membership of the state ensures that the resulting event-based system's state will stay in a given set.

Chapter 4 presents a novel method based on RPI set and ultimate bound to derive an event condition for discrete time perturbed LTI systems which ensure the same nominal performances as a state-feedback controller (in term of mRPI set). The main difference with the existing event-based control method, is that the control signal is not kept constant between events but is rather switched to the equilibrium value (which is 0 for a stabilization around the origin). It can be seen as a car going in a straight line, if the vehicle goes straight enough (e.g. there is no disturbances), there is no need to steer the wheels, but if the car moves away from the trajectory, the wheels will be controlled to bring back the vehicle to the trajectory.

2.5 Self-triggering

All the aforementioned methods require to check an event function periodically or continuously. With *self-triggered* control, the next update is scheduled only depending on the actual state value and the model knowledge. In [Velasco et al., 2009], by extension of Lyapunov sampling presented in Section 2.2.1, a trigger mechanism is built by approximating the time needed for the state to reach the next sampling curve. Another method

⁶PieceWise Linear

of self-triggered control for polynomial and state-dependent homogeneous systems based on ISS property is detailed in [Anta and Tabuada, 2010].

Discrete time event-based controller

Contents

3.1	Event function design	21
3.2	A simple example	22
3.2.1	Performance indexes	23
3.2.2	Simulation results	23
3.3	Implementation on a real-time system	25
3.3.1	Description of the system	26
3.3.2	Angular positions and torques	26
3.3.3	Dynamic model	27
3.3.4	Time-triggered control design	29
3.3.5	Introducing a reference signal	29
3.3.6	Observer design	30
3.3.7	Disturbance cancellation	31
3.3.8	Event function	31
3.4	Real-time experimental results	32
3.4.1	Scenario	32
3.4.2	Obtained results	32
3.5	Discussion	35

In this chapter it is proposed to design a periodic event-triggered control for a linear discrete time system. This work was published in [Boisseau et al., 2015]. The aim is to ensure the stability of the discrete LTI system while updating the control less often than with time-triggered control. Event mechanism designed here is an extension of the work in [Durand et al., 2014a] for the discrete-time domain. In [Durand et al., 2014a] an event-mechanism is designed to be checked continuously, while the implementation is periodic. The hypothesis behind is that the sampling period of the system is shorter than the MSI of the event-triggered system. Conversely, it is proposed here to consider the periodic nature

of the controller by designing a discrete event-function. Simulation results will be shown and an implementation on a real-time system will be performed and discussed as well.

Let consider a discrete LTI system:

$$x_{k+1} = Ax_k + Bu_k \quad (3.1)$$

where $k \in \mathbb{N}$, $x \in \mathbb{R}^n$ denotes the state, $u \in \mathbb{R}^m$ the control input, $A \in \mathbb{R}^{n \times n}$, and $B \in \mathbb{R}^{n \times m}$. The pair (A, B) is assumed to be controllable, and the state to be fully measured, the second assumption will be relaxed later.

We aim to design an event-based feedback which stabilizes the system while it does not necessarily need to update the control signal every sampling instant. The work presented in [Marchand et al., 2013; Durand et al., 2014a] will be extended to design a feedback law and an event-function for the discrete system (3.1). By designing an event-based control for a discrete time system, we do not have to prove that a lower bound exist for the MSI (introduced in the previous chapter), as the event-function is not checked continuously.

First we will design a discrete time feedback law which stabilizes the system using an optimal controller in the sense that it minimizes a LQ¹ cost function. LQ optimal control focuses on stabilizing a system and minimizing a cost function J expressed by:

$$J = \sum_{k=0}^N (x_k^T Q x_k + u_k^T R u_k) \quad (3.2)$$

The first term in (3.2) corresponds to the energy of the states x_k , while the second term corresponds to the energy of the control u_k . The term $N \in \mathbb{N}$ refers to the horizon. $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$ are tunable positive definite matrices to weight the importance of the minimization of the states or the control energy cost. The control signal stabilizing the system and minimizing J for an infinite horizon is defined as follows:

$$u_k = -Kx_k \quad (3.3)$$

$$\text{with } K = (R + B^T P B)^{-1} B^T P A \quad (3.4)$$

where P is a solution of the DARE²:

$$P = Q + A^T \left(P - P B (R + B^T P B)^{-1} B^T P \right) A \quad (3.5)$$

System (3.1) with control input (3.3) becomes:

$$x_{k+1} = (A - BK) x_k \quad (3.6)$$

¹Linear Quadratic

²Discrete Algebraic Riccati Equation

3.1 Event function design

The event-based control signal is defined as follows:

$$u_k = \begin{cases} -Kx_k & \text{if } \xi(x_k^*) \geq 0 \\ u_{k-1} & \text{otherwise} \end{cases} \quad (3.7)$$

The event-function $\xi : \mathbb{R}^n \mapsto \mathbb{R}$ will be constructed to select the most appropriate update behavior. The event-triggered system consists in solution of (3.1) given controller (3.7), the trajectory of the state is described by:

$$x_{k+1}^* = \begin{cases} (A - BK)x_k^* & \text{if } \xi(x_k^*) \geq 0 \\ Ax_k^* - BKx_i^* & \text{otherwise} \end{cases} \quad (3.8)$$

where $x^* \in \mathbb{R}^n$ denotes the state of the event-triggered system. The positive integer $i \in \{0, k-1\}$ denotes the value of k at the last event.

To assess if a control signal update is necessary, a quantitative comparison between both update policies will be performed. The convergence speed of each will be evaluated. P in (3.5) defines a Lyapunov function $V(x_k) = x_k^T P x_k$ for system (3.6). The event-function ξ will be designed such that V is a Lyapunov function for event-based system (3.8).

At each time step $V(x_k^*)$ is evaluated. We forecast the decrease of the Lyapunov function (i.e. the difference between two successive instants) if the control signal is held ($\Delta H(x_k^*)$) or if an event occurs and the control signal is updated ($\Delta E(x_k^*)$) by using model (3.8):

$$\Delta H(x_k^*) = (Ax_k^* - BKx_i^*)^T P (Ax_k^* - BKx_i^*) - V(x_k^*) \quad (3.9)$$

$$\Delta E(x_k^*) = ((A - BK)x_k^*)^T P ((A - BK)x_k^*) - V(x_k^*) \quad (3.10)$$

Then, an event is triggered and the control signal updated as soon as:

$$\xi(x_k^*) \geq 0 \quad (3.11)$$

$$\text{with } \xi(x_k^*) = \Delta H(x_k^*) - \sigma \cdot \Delta E(x_k^*) \quad (3.12)$$

where $0 < \sigma \leq 1$ is a degree of freedom to tune the performances of the event-based system.

We want to prove that V is a Lyapunov function for system (3.8). Let consider the difference of V between two successive instants, expressed by:

$$\Delta V(x_k^*) = V(Ax_k^* + Bu_k) - V(x_k^*) = \begin{cases} \Delta E(x_k^*) & \text{if } \xi(x_k^*) \geq 0 \\ \Delta H(x_k^*) & \text{otherwise} \end{cases} \quad (3.13)$$

Two cases arise from (3.13):

- If the control is updated, then $\Delta V(x_k^*) = \Delta E(x_k^*)$ is strictly negative, as V is a Lyapunov function for system (3.6).
- If the control is not updated, then $\Delta V(x_k^*) = \Delta H(x_k^*)$. And because $\xi(x_k^*) < 0$, $\Delta H(x_k^*) < \sigma \Delta E(x_k^*) < 0$.

It results that $\Delta V(x_k^*)$ is upper bounded by $\sigma \Delta E(x_k^*)$. As $\Delta E(x_k^*)$ is always negative and σ is strictly positive, the value of $V(x_k^*)$ is then monotonically decreasing along the trajectory of x^* , meaning that V is a Lyapunov function for system (3.8). Moreover if $\sigma = 1$, the upper bound for $\Delta V(x_k^*)$ will be $\Delta E(x_k^*)$, meaning that the control is kept constant only if it results in a better convergence of the Lyapunov function V .

3.2 A simple example

Let consider a double integrator with a sampling time $T_s = 10^{-2}$ s. The discrete time model is given by discrete time system (3.1) with:

$$A = \begin{bmatrix} 1 & 0.01 \\ 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0.01 \end{bmatrix}$$

Using the weighting matrices:

$$Q = \text{diag} \left(\begin{bmatrix} 100 & 1 \end{bmatrix} \right) \quad \text{and} \quad R = 0.01 \quad (3.14)$$

and solving the DARE (3.5), leads to the following state-feedback gain:

$$K = \begin{bmatrix} 30.29 & 8.35 \end{bmatrix} \quad (3.15)$$

This state-feedback can be seen as a Proportional Derivative (PD) controller as the second state corresponds to the time-derivative of the first one.

Given the state feedback K , three updates policy will be compared:

1. A time-triggered policy which consists in updating the controller periodically at the sampling time T_s .
2. A policy which trigger an update as soon as the error is changing too fast.
3. The policy described in Section 3.1.

The second policy will be called SOD³ and corresponds to the mechanism described in [Årzén, 1999]. Given the current error e_k , the error at the last update e_i , and a threshold μ , an update is triggered as soon as $|e_k - e_i| > \mu$.

The third policy will be tested for several values of σ in order to compare the different results.

Initial state is set to $x_0 = [1 \ 0]^T$ for all the simulations.

3.2.1 Performance indexes

Several performance indicators introduced in [Sánchez et al., 2009] will be used:

- The number (Nb) of updates required to perform the test bench.
- The IAE⁴ index, which gives information about the reference tracking performances, and is defined as follows:

$$\text{IAE} = \sum_{k=0}^n |e_k| \cdot T_s \quad (3.16)$$

- By analogy, the IAU⁵ index gives information about the control effort, and is defined as follows:

$$\text{IAU} = \sum_{k=0}^n |u_k| \cdot T_s \quad (3.17)$$

3.2.2 Simulation results

Figure 3.1 shows the simulation results for the different updating mechanisms. During this simulation σ and μ have been set to 0.2 and 0.01 respectively. Figure 3.2 shows a zoom between 1.5 and 3.5s.

This simulation shows that while the presented approach needs less updates, it also performs better in term of convergence speed to the reference. The SOD policy does not permit to converge exactly but leads to a periodic oscillation around the reference. Table 3.3 shows performance indicators obtained after the simulation. The proposed approach obtains a smaller IAE value at the price of an increased IAU. The number of updates is seriously limited while achieving better performances in term of energy contained in the error signal. The SOD approach has the greatest IAE index, certainly due to the constant oscillations around the equilibrium. The time-triggered mechanism obtains the minimal IAU

³Send-On-Delta

⁴Integral Absolute Error

⁵Integral Absolute control signal

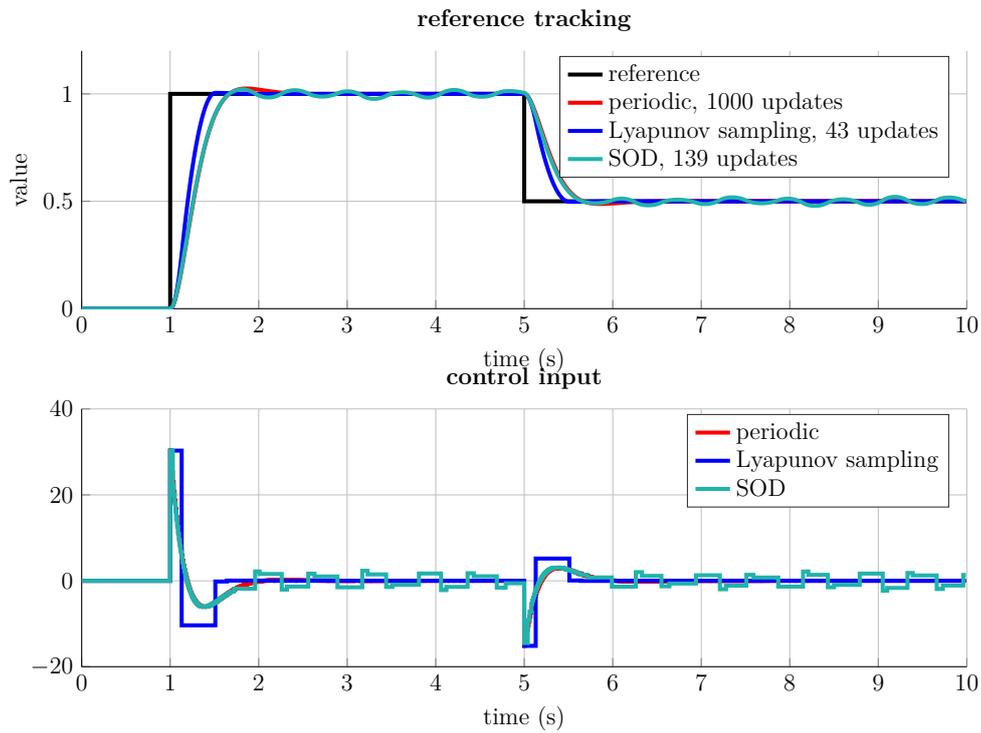


Figure 3.1: Comparison of update policy (step input).

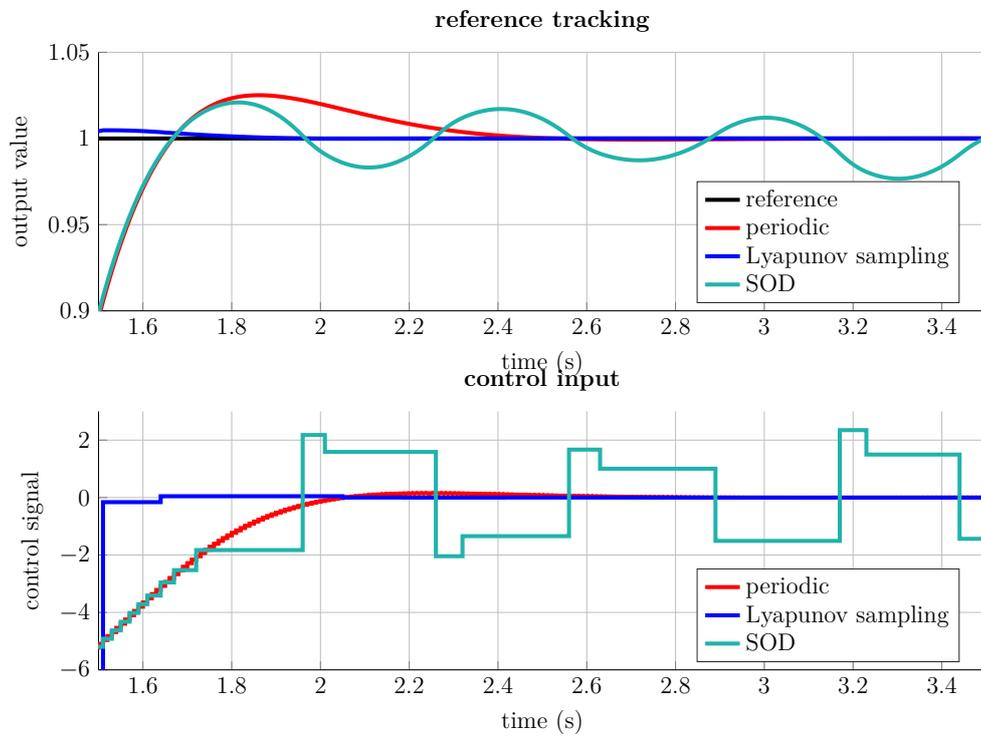


Figure 3.2: Comparison of update policy (transient response).

	Nb	IAE	IAU
Time-triggered	1000	0.445	7.56
Event-based			
$\sigma = 0.2$	43	0.327	11.87
$\sigma = 0.9$	66	0.364	9.998
$\sigma = 1$	377	0.381	9.102
Send on Delta			
$\mu = 1 \cdot 10^{-2}$	139	0.499	17.06
$\mu = 1 \cdot 10^{-3}$	218	0.448	8.32

Figure 3.3: Performance indexes for different update mechanisms.

value which means that the periodically update policy consumes less energy to stabilize the system.

Even if simulation results are very encouraging, some limitations were not considered:

- The value of the control signal is changing very fast with the proposed policy, however an actuator would have a limited bandwidth.
- Bounds on the control signal have not been considered, however due to physical limitations, bounds on the control signal can not be neglected.
- Disturbances and/or model uncertainties would certainly decrease the performances of the proposed policy.

The proposed method will be implemented and tested on a MIMO real-time system in the sequel.

3.3 Implementation on a real-time system

The experimental platform in the present work is depicted in Figure 3.4. This is a gyroscope M750p from ECP⁶ systems, where classical LQ optimal control has been previously investigated in [Martinez and Khennouf, 2011].

⁶Educational Control Products

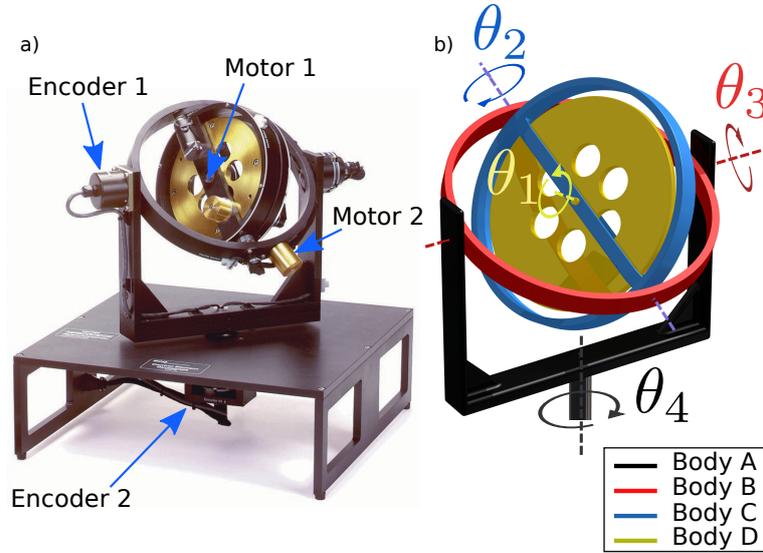


Figure 3.4: a) ECP gyroscope (model M750p), its actuators and sensors
 b) CAD model of the gyroscope and illustration of the relative angles

3.3.1 Description of the system

The gyroscope consists of 4 (rigid) rotating masses. Each of the 4 rigid bodies has an angular position θ_p relative to its rotating gimbal axis p , with $p = \{1, 2, 3, 4\}$. One can refer to Figure 3.4 for a representation of these angles. More precisely, a high inertia brass rotor (body D) is suspended in an assembly with four angular degrees of freedom. The rotor spin torque is provided by a DC motor (motor 1) whose angular position is defined as θ_1 . The first transverse gimbal assembly (body C) is driven by another motor (motor 2) to effect motion about axis 2. The relative position between bodies C and B is noted as θ_2 . The subsequent gimbal assembly, body B, rotates with respect to body A around axis 3. The relative angle between body B and A (θ_3) is measured by encoder 1. Similarly, body A rotates around a vertical axes. The relative angle (θ_4) is measured by encoder 2. Two manual brakes may be used to lock the relative position between either bodies A and B or body A and the base frame, in order to reduce the system degrees of freedom.

3.3.2 Angular positions and torques

The gyroscope is assumed to be symmetric and the center of all rigid bodies (A, B, C and D) lies at the center of body D (the rotor). As a result, only the rotational dynamics need to be taken into account. The following norms are applied hereafter:

- The angular position θ_1 of the rotor (body D) is not of importance: only the angular velocity $\omega_1 = \dot{\theta}_1$ is considered.

K_A	0.067 kg.m^2
I_B	0.012 kg.m^2
J_B	0.018 kg.m^2
K_B	0.030 kg.m^2
I_C	0.0092 kg.m^2
J_C	0.023 kg.m^2
K_C	0.022 kg.m^2
I_D	0.015 kg.m^2
J_D	0.027 kg.m^2

Table 3.1: Mechanical constants of the gyroscope actuator.

- The angular position θ_2 of the rotor drum (body C) is set to 0 if the rotor drum (body C) is perpendicular to the inner gimbal (body B).
- The angular position θ_3 of the inner gimbal (body B) is 0 if the inner gimbal (body B) is perpendicular to the outer gimbal (body A).
- Since the outer gimbal (body A) is able to rotate freely and the gyroscope is assumed to be symmetric, θ_4 can be reset to $\theta_4 = 0$ at any angular position of the outer gimbal (body A).

The angular position of the 4 rigid bodies in the gyroscope can be controlled with the 2 internal torques T_1 and T_2 , provided by DC motors. T_1 rotates the D body around its axis (flying wheel driver) while T_2 rotates the C body around the second axis (longitudinal).

3.3.3 Dynamic model

The gyroscope is a complex nonlinear system. However, for a constant angular velocity ω_1 , it can be modelled as a multivariable linear system. Thus, considering small variations around the operating point defined by the angular speed $\omega_1 = \Omega$ and the angles $\theta_p = 0$ for $p = 2$ to 4, gives:

$$\begin{aligned}
 \dot{\omega}_2 &= \frac{J_D \Omega}{I_C + I_D} \omega_4 + \frac{T_2}{I_C + I_D} \\
 \dot{\omega}_3 &= -\frac{T_1}{J_B + J_C} \\
 \dot{\omega}_4 &= -\frac{J_D \Omega}{I_D + K_A + K_B + K_C} \omega_2
 \end{aligned} \tag{3.18}$$

The numerical values of the inertia of the four bodies are presented in Table 3.1. The (fixed) angular velocity for ω_1 is $\Omega = 42 \text{ rad/s}$.

The actuators are limited to:

$$\begin{aligned} |T_1| &\leq 0.2 Nm \\ |T_2| &\leq 3.0 Nm \end{aligned} \quad (3.19)$$

and the angles are limited to:

$$|\theta_p| \leq 20^\circ \quad \text{for } p = 2 \text{ to } 4 \quad (3.20)$$

A state-space representation of system (3.18) in the continuous-time domain is obtained in the form:

$$\begin{aligned} \dot{x} &= A_c x(t) + B_c u(t) \\ y(t) &= C x(t) \end{aligned} \quad (3.21)$$

where state, input, disturbance and output vectors are respectively defined by:

$$\begin{aligned} x &:= (\theta_3 \quad \theta_4 \quad \omega_2 \quad \omega_3 \quad \omega_4)^T \\ u &:= (T_1 \quad T_2)^T \\ y &:= (\theta_3 \quad \theta_4)^T \end{aligned} \quad (3.22)$$

and where:

$$\begin{aligned} A_c &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \frac{J_D \Omega}{I_C + I_D} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{J_D \Omega}{I_D + K_A + K_B + K_C} & 0 & 0 \end{bmatrix}, \\ B_c &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{I_C + I_D} \\ -\frac{1}{J_B + J_C} & 0 \\ 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (3.23)$$

The discretization of system (3.23) with sampling time T_s is computed as follows:

$$\begin{aligned} A &= e^{A_c T_s}, \\ B &= \left(\int_0^{T_s} e^{A_c \tau} d\tau \right) B_c \end{aligned} \quad (3.24)$$

The resulting discrete-time state space representation is given by:

$$x_{k+1} = A x_k + B(u_k + d_k), \quad (3.25)$$

$$y_k = C x_k \quad \text{with } k \in \mathbb{N} \quad (3.26)$$

where the subscript k in x_k denotes the k^{th} sample of the state: $x_k = x(k \cdot T_s)$.

3.3.4 Time-triggered control design

Let us define the cost function:

$$J = \sum_{k=0}^{\infty} (x_k^T Q x_k + u_k^T R u_k) \quad (3.27)$$

$$Q = \text{diag} \left([10 \ 70 \ 1 \ 0 \ 1] \right) \quad R = \begin{bmatrix} 3 & 0 \\ 0 & 0.02 \end{bmatrix} \quad (3.28)$$

By solving the DARE (3.5), a stabilizing state-feedback gain K is computed such that the cost function J is minimized:

$$K = \begin{bmatrix} -1.66 & 0 & 0 & -0.37 & 0 \\ 0 & -8.77 & 1.28 & 0 & -0.85 \end{bmatrix} \quad (3.29)$$

The system (3.25) with the control $u_k = -Kx_k$ is asymptotically stable for all $k \in \mathbb{N}$.

3.3.5 Introducing a reference signal

We want to stabilize the output around a slow varying reference $r_k = [\theta_3 \ \theta_4]^T$. At the steady state, the state will be $x_{ss} = N_x r_k$ and the control will be $u_{ss} = N_u r_k$.

$$x_{ss} = Ax_{ss} + Bu_{ss} \quad (3.30)$$

$$N_x r_k = AN_x r_k + BN_u r_k \quad (3.31)$$

$$y_{ss} = r_k = CN_x r_k \quad (3.32)$$

N_x and N_u can be computed by solving:

$$\begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix} \cdot \begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix} \quad (3.33)$$

In this example $N_x = [1 \ 1 \ 0 \ 0 \ 0]^T$. The DC gain is infinite for both of the control input, in consequence $N_u = 0$.

Let define the tracking error between the state and the reference $e_k = x_k - x_{ss}$. By using the control:

$$u_k = -K e_k \quad (3.34)$$

the dynamics of e can be expressed as:

$$e_{k+1} = x_{k+1} - N_x r_{k+1} = Ax_k + Bu_k - N_x r_{k+1} \quad (3.35)$$

because of the slow varying hypothesis of the reference:

$$e_{k+1} = Ax_k + Bu_k - N_x r_k \quad (3.36)$$

$$e_{k+1} = A(e_k + x_{ss}) - BK e_k - x_{ss} \quad (3.37)$$

$$e_{k+1} = (A - BK)e_k + (A - I) \cdot x_{ss} \quad (3.38)$$

From (3.30) and since $N_u = 0$: $x_{ss} = Ax_{ss}$, therefore:

$$e_{k+1} = (A - BK)e_k \quad (3.39)$$

If the eigenvalues of $A - BK$ are located inside the unit circle, the tracking error will stabilize around 0, so the state x will converge to the reference r . Moreover V is a Lyapunov function for system (3.39).

3.3.6 Observer design

In the previous section, a state-feedback was designed under the assumption that the state was available at any time. However, this is not the case in practice as only θ_3 and θ_4 can be measured. To overcome this issue, an observer will be designed in order to estimate the state vector.

A Luenberger observer consists in a simulation of the system given the control input plus a correction term depending on the estimation error. The dynamics of the estimated state can be expressed as follows:

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L \cdot (y_k - \hat{y}_k) \quad (3.40)$$

$$\hat{y}_k = C\hat{x}_k \quad (3.41)$$

where $\hat{x} \in \mathbb{R}^n$ denotes the estimated state, $L \in \mathbb{R}^{n \times p}$ denotes the observer matrix gain.

Let us define the observer error as $\hat{o}_k = x_k - \hat{x}_k$, its dynamics is described by:

$$\hat{o}_{k+1} = (A - LC)\hat{o}_k \quad (3.42)$$

L is computed such that the eigenvalues of $A - LC$ are located inside the unit circle. In consequence, \hat{x}_k will converge to x_k .

Using the control $u_k = -K\hat{x}_k$ in system (3.25) leads to:

$$x_{k+1} = (A - BK)x_k - BK\hat{o}_k \quad (3.43)$$

As \hat{o} will converge to zero, and because $A - BK$ is Hurwitz, x will also converge to zero.

Remark 3.1

The dynamics of the observation error (3.42) does not depend on the control u_k .

Remark 3.2

Introducing a reference signal as explained in Section 3.3.5 can be extended to the case where $u_k = -K\hat{e}_k$ with $\hat{e}_k = \hat{x}_k - x_{ss}$.

3.3.7 Disturbance cancellation

We designed a state feedback and an observer without considering any disturbances in the model, however this is likely not to be the case. Disturbances d_1 and d_2 will be included in the model (3.18):

$$\begin{aligned}\dot{\omega}_2 &= \frac{J_D \Omega}{I_C + I_D} \omega_4 + \frac{1}{I_C + I_D} (T_2 + d_2) \\ \dot{\omega}_3 &= -\frac{1}{J_B + J_C} (T_1 + d_1) \\ \dot{\omega}_4 &= -\frac{J_D \Omega}{I_D + K_A + K_B + K_C} \omega_2\end{aligned}\quad (3.44)$$

Let define the augmented state vector:

$$\bar{x} = \begin{bmatrix} x \\ d \end{bmatrix}\quad (3.45)$$

where $d = [d_1 \ d_2]^T$. By considering constant disturbances ($d_{k+1} = d_k$), the dynamics of \bar{x} can be described by:

$$\bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{B}u_k\quad (3.46)$$

$$y_k = \bar{C}\bar{x}_k\quad (3.47)$$

where:

$$\bar{A} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad \bar{C} = [C \ 0]\quad (3.48)$$

where I and 0 denote identity and zero matrices of appropriate dimension. An observer is designed as described in Section 3.3.6 to obtain $\hat{\hat{x}} = [\hat{x} \ \hat{d}]^T$, an estimation of \bar{x} . Therefore the following control:

$$u_k = -K\hat{x}_k - \hat{d}_k\quad (3.49)$$

will stabilize system (3.44) and cancel the disturbances d .

3.3.8 Event function

A periodic event-based controller will be used as described in Section 3.1:

$$u_k = \begin{cases} -K\hat{e}_k - \hat{d}_k & \text{if } \xi(\hat{e}_k) \geq 0 \\ u_{k-1} & \text{otherwise} \end{cases}\quad (3.50)$$

where $\hat{e}_k = \hat{x}_k - x_{ss}$ and with the event-function:

$$\xi(\hat{e}_k) = \Delta H(\hat{e}_k) - \sigma \cdot \Delta E(\hat{e}_k)\quad (3.51)$$

We recall that $\Delta H(\hat{e}_k)$ and $\Delta E(\hat{e}_k)$ denote respectively the decrease of V if the control is kept constant or updated.

		Nb	IAE	IAU
Time-based LQR strategy without dry friction cancellation		1000	1.29	2.01
Event-based LQR strategy without dry friction cancellation	$\sigma = 0.9$	176	1.48	2.03
	$\sigma = 0.4$	129	1.91	2.36
Time-based LQR strategy with dry friction cancellation		1000	0.46	2.08
Event-based LQR strategy with dry friction cancellation	$\sigma = 0.9$	421	0.46	2.04
	$\sigma = 0.4$	246	0.53	2.17

Table 3.2: Performance indexes obtained with different control strategies

3.4 Real-time experimental results

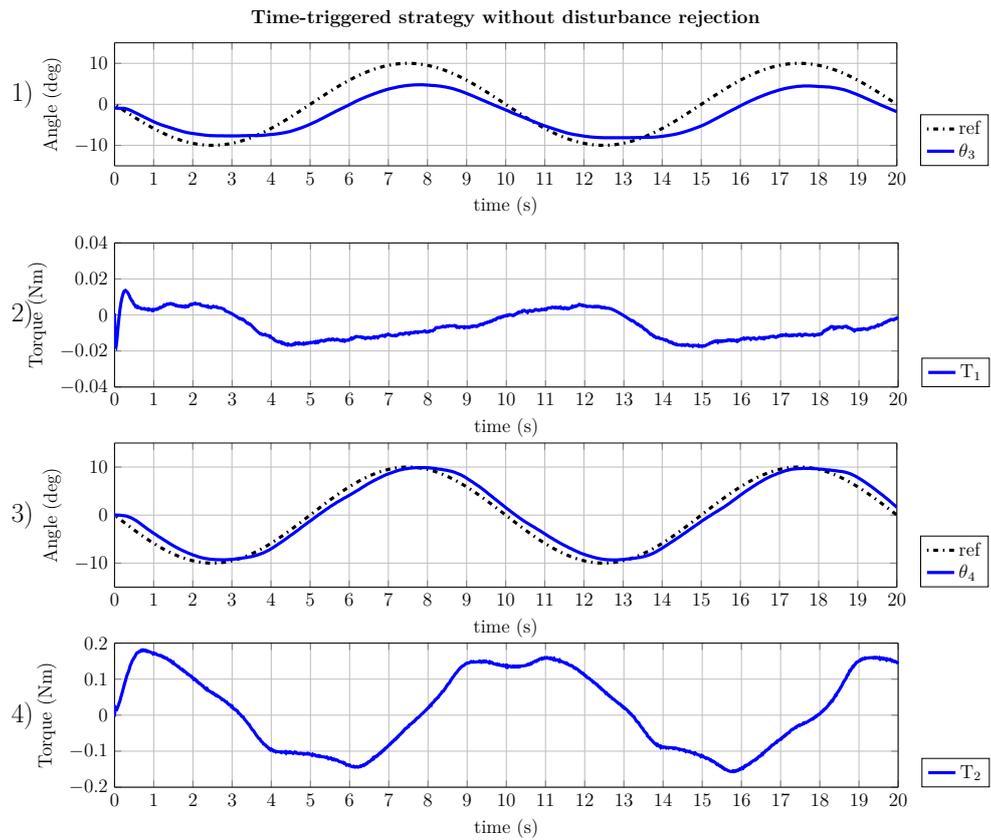
3.4.1 Scenario

We propose to control the system such that θ_3 and θ_4 track a given sinus reference. The frequency of the references will be chosen to be “slow enough” because of the slow varying reference hypothesis (see Section 3.3.5). In this example the sinus will have a frequency $f = 0.1$ Hz. Prior to this, ω_1 is stabilized around a constant value to satisfy the assumption of linear form (3.23). The computation and the real-time control are done with a computer running xPC Target and Matlab/Simulink.

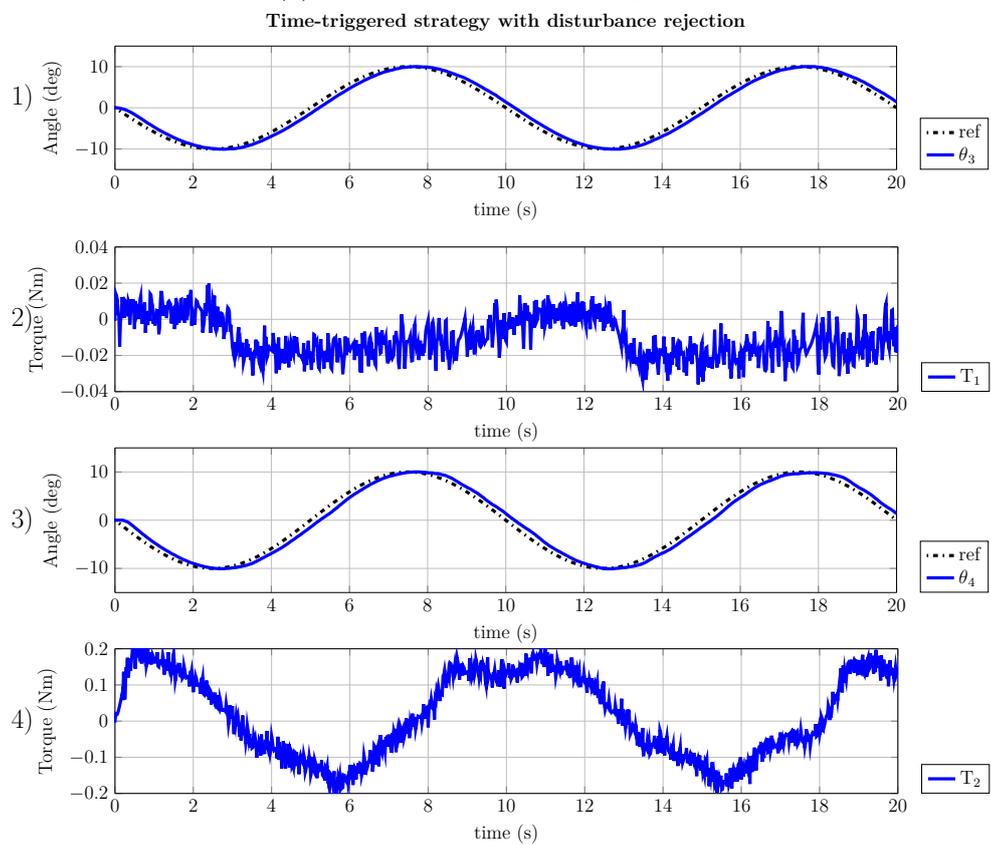
3.4.2 Obtained results

Figure 3.5 shows the experimental results for the time-triggered control without disturbances rejection. Plot 1) illustrates the setpoint and the measured angle θ_3 , while plot 2) shows the applied control torque T_1 . Plots 3) and 4) show θ_4 , the setpoint and the control torque T_2 . Figure 3.5a and Figure 3.5b compare the results for a time-triggered controller without and with disturbance rejection respectively. Figure 3.6 illustrates the event-triggered system output with disturbances rejection for $\sigma = 0.4$ (in Figure 3.6a) and $\sigma = 0.9$ (in Figure 3.6b). The extra plot 5) indicates when the control input signal is updated, '0' means the control is held and '1' means the control is updated. Figure 3.7 shows the disturbances estimation in the time-triggered case, the top plot shows \hat{d}_1 while the bottom one depicts \hat{d}_2 .

The performance indexes obtained for the experimental results are reported in Table 3.2 and discussed in the sequel.



(a) Without disturbances rejection



(b) With disturbances rejection

Figure 3.5: Experimental results: time-triggered strategy

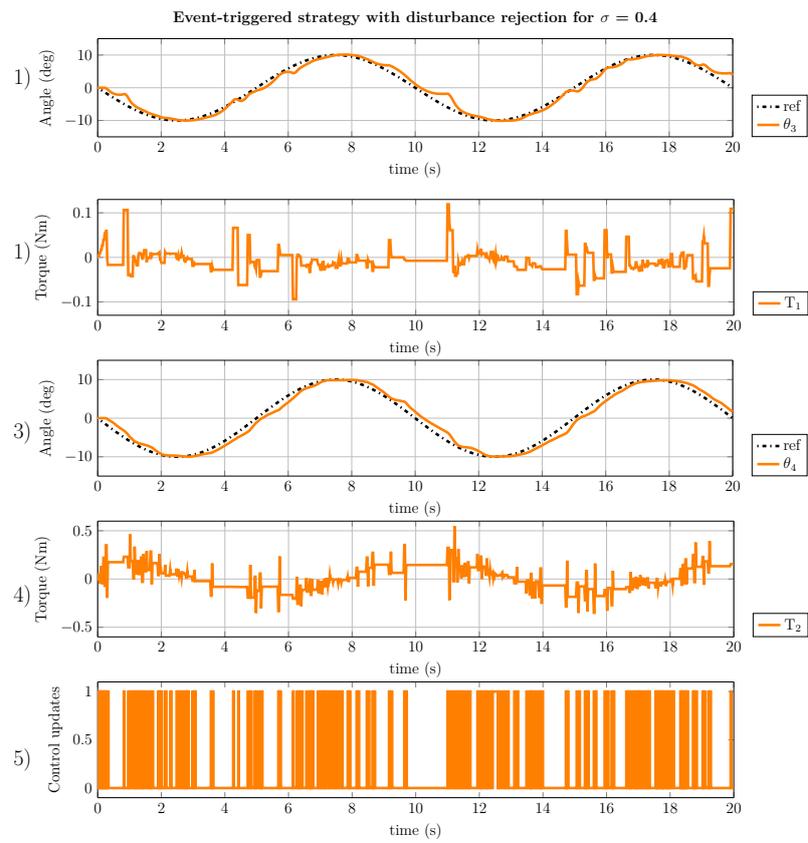
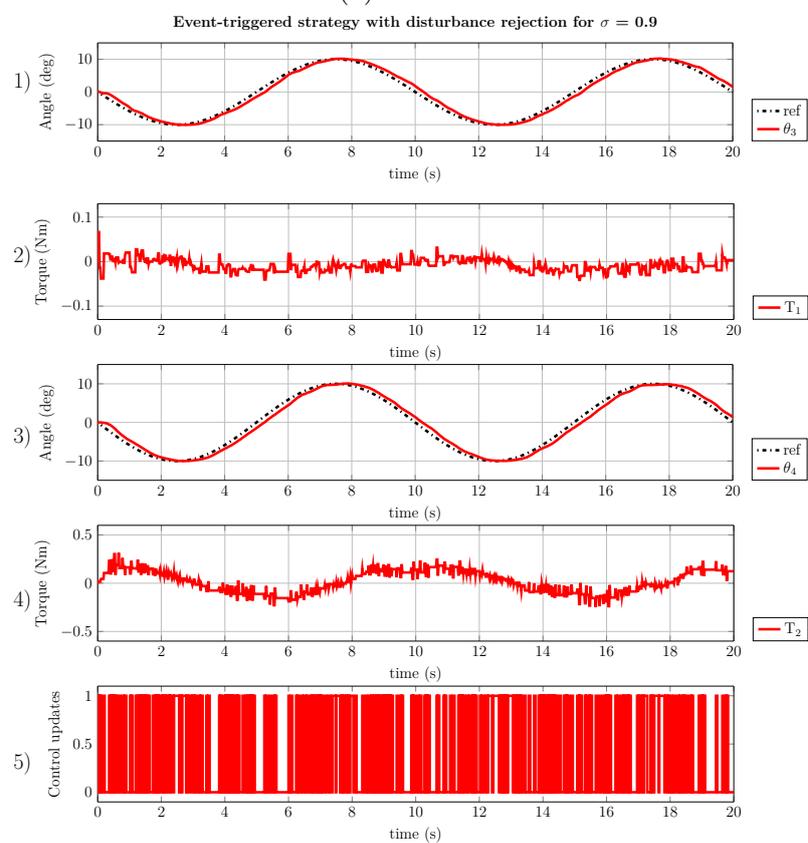
(a) $\sigma = 0.4$ (b) $\sigma = 0.9$

Figure 3.6: Experimental results: event-triggered strategy with disturbances rejection

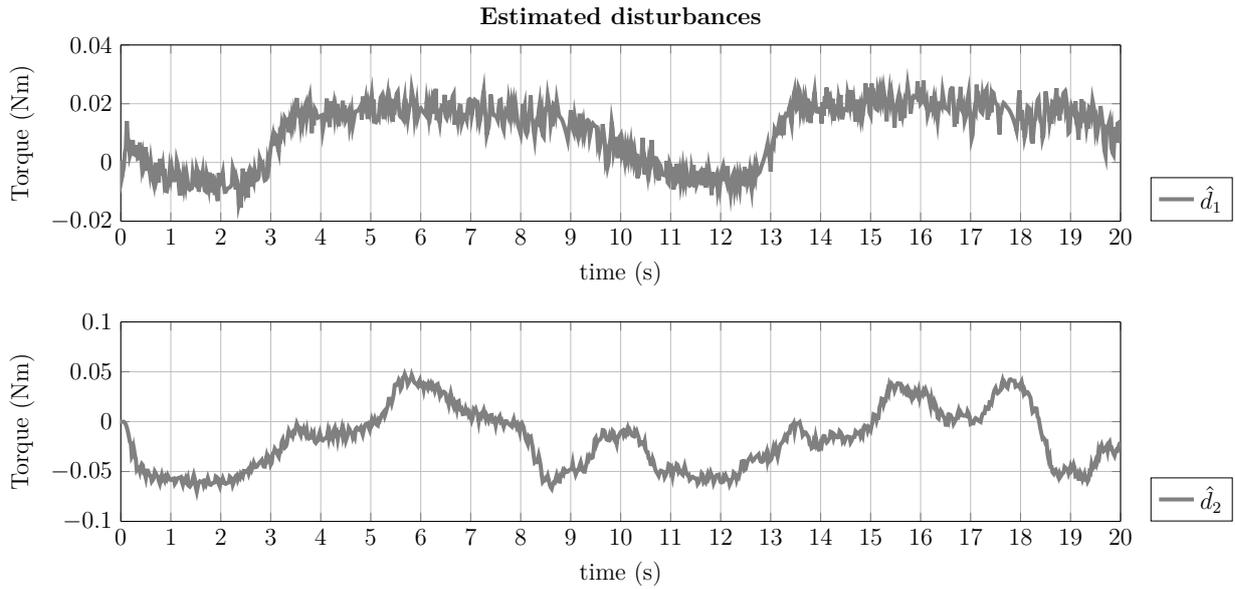


Figure 3.7: Experimental results: disturbances estimation

3.5 Discussion

The performances for the event-triggered strategy with $\sigma = 0.9$ are almost equal to the time-triggered case with about 58% less control updates. θ_3 and θ_4 look similar in the time-triggered case (Figure 3.5b) and the event-triggered case (Figure 3.6b). Nevertheless, the control input changes more abruptly in the event-triggered case. The smaller σ is, the lower the frequency of updates is, with a higher IAE and IAU performances indexes. Tuning σ is a trade off between the energy in the error signal and the number of updates.

The importance of the disturbance observer is highlighted by comparing Fig. 3.5a and 3.5b, it can be noticed that the tracking of θ_3 is really better when using the disturbances cancellation. Moreover this difference is also visible in Table 3.2, IAE index is three times higher when the dry friction cancellation is not used.

Event-switched approach

Contents

4.1	Polyhedral RPI sets for characterizing nominal performances	38
4.2	Event-switched system	39
4.3	Event-set computation	40
4.3.1	Reducing the complexity of the event-set	42
4.4	Event-switched control design with guaranteed performances . .	42
4.4.1	Main results	42
4.4.2	Analytic discussion of performances	44
4.4.3	Observer based approach	46
4.4.4	Proposed algorithm	47
4.5	Simulation of the event-switched approach	50
4.5.1	Performance indexes	52
4.5.2	Results	53
4.5.3	Reducing computation complexity	55

The main goal of this chapter is to provide a systematic method dedicated to discrete disturbed LTI systems which limit the number of control updates while keeping the same performances as the time-triggered system. The mRPI set of a disturbed system gives a way to quantify its nominal performances, as it describes the minimal region in which the states of the system will stay once inside.

Let us consider the following discrete-time linear time-invariant system:

$$x_{k+1} = Ax_k + Bu_k + Ew_k \quad (4.1)$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $E \in \mathbb{R}^{n \times f}$, and $k \in \mathbb{N}$. $x_k \in \mathbb{R}^n$ denotes the current state of the system, x_{k+1} is the successor state, $u_k \in \mathbb{R}^m$ denotes the control input, and $w_k \in \mathcal{W}$ denotes the unknown bounded disturbances. \mathcal{W} is assumed to be a compact convex

polytope in \mathbb{R}^f defined by p vertices \mathcal{W}_i as follows: $\mathcal{W} \triangleq \text{conv} \{\mathcal{W}_i\}$ for $i = \{1, \dots, p\}$. The set \mathcal{W} is taken to include the origin, and $\text{rank}(E)$ is assumed to be greater than or equal to n : the set $E\mathcal{W}$ therefore also includes the origin. It is also assumed that the matrix pair (A, B) is *stabilizable* using the linear discrete-time state feedback:

$$u_k = -Kx_k \quad (4.2)$$

where $K \in \mathbb{R}^{n \times m}$ is the state-feedback matrix. The dynamics of the closed-loop system can therefore be described by the following equation:

$$x_{k+1} = \underbrace{(A - BK)}_{A_{CL}} x_k + Ew_k \quad (4.3)$$

where all the eigenvalues of the closed-loop matrix A_{CL} are assumed to be located inside the unit circle. In what follows, the system dynamics (4.3) will be referred to as the **time-triggered system**.

The first problem here is how to find an approximation for the mRPI set describing all possible trajectories of the system's state, starting at the origin (in the case of null stabilization) in the presence of bounded disturbances. \mathcal{S}_1 will denote the nominal performance set of a system. Since no means of computing an exact mRPI have yet been found when the system's dynamics are not nilpotent, a polyhedral approximation for the mRPI set will be used to characterize what has been called the *nominal performance* of the time-triggered system.

4.1 Polyhedral RPI sets for characterizing nominal performances

Polyhedral RPI sets for linear systems can be computed using methods such as those developed in [Kofman et al., 2007], [Seron et al., 2008] and [Martinez et al., 2009]. These methods are very useful for computing RPI sets in polyhedral form, especially in the case of stable systems where the eigenvalues of the matrix A_{CL} are real values. A systematic method to obtain polyhedral approximations for the mRPI set is presented in [Rakovic et al., 2005]. Polyhedral invariant set computations have many advantages over traditional ultimate-bound methods of computation based on the use of quadratic Lyapunov functions, as discussed in [Haimovich et al., 2008] and [Martinez, 2015]. In particular, these methods may be less conservative (i.e., they give smaller sets) than ellipsoids describing quadratic Lyapunov functions.

A refined polyhedral approximation for the mRPI set can be obtained by implementing a sequence of *outer* approximations of the mRPI set, as described in [Olaru et al., 2010] and [Martinez, 2015]. This method consists in building a sequence of RPI sets recursively, based

on the Minkowski sum between the image of a RPI set given by the linear transformation A_{CL} and the polyhedral set EW . This gives:

$$\Phi(s+1) = A_{CL}\Phi(s) \oplus EW, \quad \Phi(0) = \Psi \quad (4.4)$$

where A_{CL} is a stable matrix (all its eigenvalues are located inside the unit circle) and EW is a polyhedral set which includes all the disturbance trajectories. $\Phi(s)$ is the linear transformation. $\Psi \in \mathbb{R}^n$ is a polyhedral describing an initial RPI set estimation of the state trajectories, which can be computed using the method presented in [Martinez, 2015]. Therefore, [Olaru et al., 2010] states that for any $\delta > 0$, there exists $s^* \in \mathbb{N}$ such that the following relation is true:

$$\Omega_\infty \subseteq \Phi(s^*+1) \subseteq \Phi(s^*) \subseteq \Psi \quad (4.5)$$

where Ω_∞ denotes the exact mRPI set. Then, $\Phi(s^*) \rightarrow \Omega_\infty$ as $s^* \rightarrow \infty$, that is:

$$\Omega_\infty \subseteq \Phi(s^*) \subseteq \Omega_\infty \oplus \mathbb{B}_p^n(\varepsilon) \quad (4.6)$$

where $\mathbb{B}_p^n(\varepsilon)$ denotes a n -dimensional ball with the radius ε with respect to the p -norm.

Remark 4.1

The initial set Ψ in the set recursion (4.4) can in fact be any RPI set for system (4.3).

4.2 Event-switched system

Let \mathcal{S}_2 be some set in \mathbb{R}^n . Let now define the following event-switched control law:

$$u_k = \begin{cases} 0_m & \text{if } x_k \in \mathcal{S}_2 \\ -Kx_k & \text{otherwise} \end{cases} \quad (4.7)$$

where 0_m denotes the m -dimensional null vector. In the sequel, an **event-switched system** will refer to a system of the form (4.1) with a control input as in (4.7). It is given by the following closed-loop equation:

$$x_{k+1} = \begin{cases} Ax_k + Ew_k & \text{if } x_k \in \mathcal{S}_2 \\ (A - BK)x_k + Ew_k & \text{otherwise} \end{cases} \quad (4.8)$$

Remark 4.2

For the sake of simplicity, the case of stabilization around the origin is studied here. In the general case, the control law will also depend on the a priori control input u_{eq} and the state vector x_{eq} both required at equilibrium, for instance:

$$\bar{u}_k = u_{eq} - K(x_k - x_{eq})$$

and the event-switched control would be:

$$\bar{u}_k = \begin{cases} u_{eq} & \text{if } x_k \in \mathcal{S}_2 \\ u_{eq} - K(x_k - x_{eq}) & \text{otherwise} \end{cases}$$

The signals u_{eq} and x_{eq} can be computed at the plant level at all times. These signals can be desired path-planning or reference model signals, for example.

Problem statement: Given the plant model (4.1), a state feedback (4.2) and a RPI set \mathcal{S}_1 for the time-triggered state-feedback loop (4.3), find an event condition (i.e. a condition based on a set \mathcal{S}_2) and an event-switched mechanism (4.7) such that the set \mathcal{S}_1 is a RPI set for both the time-triggered system (4.3) and the event-switched system (4.8).

In this chapter, the event-switched mechanism is assumed to be given *a priori*. The aim is to set the event condition giving the event-triggered system (4.8) the same performances as the time-triggered system (4.3) (in terms of RPI set). The event condition makes the mechanism to switch from the open loop system to the system controlled by a state feedback law.

In what follows, the performances are given by the following four indicators:

- The bound δ_x on the 2-norm of the state x ;
- The bound δ_u on the 2-norm of the control input u ;
- The bound δ_d on the 2-norm of the changing rate of the control input: $\Delta u = u_{k+1} - u_k$;
- The number of communications required between the controller and the actuator.

The first index reflects the ability of the controller to stabilize the system, the second one indicates the cost in terms of the control power, the third one reflects the changes in the control input, which are usually to the aging of the actuator, and the fourth one denotes the cost in terms of the communication between the controller block and the actuator.

4.3 Event-set computation

Given a RPI set approximation \mathcal{S}_1 for system (4.3), the aim is now to compute a set \mathcal{S} such that:

$$x_k \in \mathcal{S} \implies x_{k+1} \in \mathcal{S}_1, \text{ with } x_{k+1} = Ax_k + Ew_k \quad (4.9)$$

Definition 4.1

A set \mathcal{S} is said to be an event-set for system (4.1) if and only if it satisfies condition (4.9) for a given set \mathcal{S}_1 .

The following method is developed here for computing the set of maximum volumes meeting (4.9), denoted \mathcal{S}_2^* . That is, \mathcal{S}_2^* contains all the possible sets which also satisfy the condition (4.9).

The set \mathcal{S}_1 is taken to be a polyhedral set with a given half-space representation, that is:

$$\mathcal{S}_1 \triangleq \{x \in \mathbb{R}^n : Hx \leq P\} \quad (4.10)$$

where $H \in \mathbb{R}^{l \times n}$ and $P \in \mathbb{R}^l$.

Condition (4.9) can be translated into a family of inequalities which can be used to find all the possible values of x_k , via the matrices \bar{H} and \bar{P}_i , satisfying the following condition:

$$x_{k+1} \in \mathcal{S}_1 \Leftrightarrow Hx_{k+1} \leq P \quad (4.11)$$

$$H(Ax_k + EW_i) \leq P \quad (4.12)$$

$$\underbrace{HA}_{\bar{H}}x_k \leq \underbrace{P - HEW_i}_{\bar{P}_i} \quad (4.13)$$

where W_i denotes the i^{th} vertex of the polyhedron \mathcal{W} .

Therefore, by defining the set:

$$\mathcal{S}_2^* \triangleq \{x \in \mathbb{R}^n : \bar{H}x \leq \bar{P}_i, \text{ for } i = \{1, \dots, p\}\} \quad (4.14)$$

Then the set \mathcal{S}_2^* will include all the event-sets.

$$x_k \in \mathcal{S}_2^* \implies x_{k+1} \in \mathcal{S}_1, \text{ with } x_{k+1} = Ax_k + Ew_k \quad (4.15)$$

This is possible because \mathcal{S}_1 and \mathcal{W} are convex and compact sets including the origin.

Remark 4.3

Please note that:

- \mathcal{S}_2^* is a convex set because it can be described by the intersection between a finite number of hyperplanes in (4.14).
- \mathcal{S}_2^* is possibly empty if the inequalities in (4.14) have no solution.
- \mathcal{S}_2^* is not necessarily included in \mathcal{S}_1 . In this particular case, $\mathcal{S}_2^* \setminus \mathcal{S}_1$ can be reached one time step before entering \mathcal{S}_1 .
- Any set included in \mathcal{S}_2^* is an event-set for system (4.1) and the set \mathcal{S}_1 .

4.3.1 Reducing the complexity of the event-set

In order to apply the event-switched control (4.7), it is necessary to verify at each sampling time if the state of the system is inside the set \mathcal{S}_2 . If this set is polyhedral, then the computational complexity to check the inclusion condition, $x_k \in \mathcal{S}_2$, depends on the order of the system and the number of hyperplanes describing the set.

An alternative strategy might be to use an ellipsoidal set. This allows to check a simpler inclusion condition, $x_k \in \mathcal{S}_2$, where:

$$\mathcal{S}_2 \triangleq \{x \in \mathbb{R}^n : x^T M x \leq c\} \quad (4.16)$$

where $M \in \mathbb{R}^{n \times n}$, and $c \in \mathbb{R}$.

The advantage of using an ellipsoidal set is that the complexity of the inclusion condition's evaluation only depends on the order n of the system, whereas a polyhedral set can be described by an arbitrary number of hyperplanes.

4.4 Event-switched control design with guaranteed performances

4.4.1 Main results

Since stabilization at a point for the time-triggered system (4.1) without perturbation is considered, we assume the very classical property for the pair (A, B) :

Assumption 4.1

The pair (A, B) of system (4.1) is stabilizable.

Let state three assumptions that enable to guaranty that, if there is a robustly positively invariant set for the time-triggered system, it will still be a robustly positively invariant set in the proposed event-switched framework. In other words, with the proposed event-switched control approach, the system will remain in the same RPI set as the time-triggered system if it starts in it. In addition we also want to guarantee that the property of ultimate boundedness is kept for the event-switched system. This will ensure that for any initial condition, the state of the event-triggered system joins the RPI set in finite number of time steps. For this, the following assumptions are sufficient:

Assumption 4.2

The set \mathcal{S}_2 is an event-set for system (4.1) and a given set \mathcal{S}_1 .

Assumption 4.3

\mathcal{S}_1 is a RPI set for system (4.1) with the stabilizing control input (4.2) and unknown bounded disturbances w_k .

Assumption 4.4

System (4.1) subjected to the stabilizing control (4.2) and bounded disturbances w_k is UB to the set \mathcal{S}_1 .

With the above assumptions, one can state the main result of this chapter:

Theorem 4.1

Under Assumptions 4.1, 4.2 and 4.3, \mathcal{S}_1 is a RPI set for system (4.1) subject to the event-switched control input (4.7).

Proof. It must now be proved that the set \mathcal{S}_1 satisfies Definition 1.8 for system (4.1) with the event-switched control input (4.7). The dynamics of this system were described in equation (4.8), which is recalled here:

$$x_{k+1} = \begin{cases} Ax_k + Ew_k & \text{if } x_k \in \mathcal{S}_2 \\ (A - BK)x_k + Ew_k & \text{otherwise} \end{cases}$$

Let us assume the existence of an arbitrary state x_k in \mathcal{S}_1 . As far as the dynamics of system (4.8) are concerned, there are two possible cases:

- $x_k \in \mathcal{S}_2$, which means that x_{k+1} will also be in \mathcal{S}_1 because the set \mathcal{S}_2 meets the condition (4.9) from Assumption 4.2.
- $x_k \notin \mathcal{S}_2$, which means that x_{k+1} will still be in \mathcal{S}_1 because it is a RPI set for the associated dynamics resulting from Assumption 4.3.

This proves that \mathcal{S}_1 is a RPI set for system (4.8), because for any x_0 in \mathcal{S}_1 , x_k will be in \mathcal{S}_1 for all values of $k \in \mathbb{N}$. □

Theorem 4.2

Under Assumptions 4.1, 4.2, and 4.4, system (4.1) with the event-switched control input (4.7) is UB within \mathcal{S}_1 .

Proof. It now has to be proved that for any initial state x_0 , there exists $K(x_0) \in \mathbb{N}$ such that x_k will be in \mathcal{S}_1 for all $k \geq K(x_0)$. When dealing with system (4.8), there are two possible cases:

- $x_0 \in \mathcal{S}_2$ means that x_k will join \mathcal{S}_1 in $K(x_0) = 1$ time step resulting from (4.9).

- $x_0 \notin \mathcal{S}_2$, based on Assumption 4.4, x_k will join \mathcal{S}_1 in a finite number of time steps, namely $K(x_0)$.

Once x_k is in \mathcal{S}_1 , *i.e.* once k is greater than or equal to $K(x_0)$, there are again two possibilities:

- $x_k \in \mathcal{S}_2$, which means that $x_{k+1} = Ax_k + Ew_k$ will be in \mathcal{S}_1 , for all $k \geq K(x_0)$ because the set \mathcal{S}_2 satisfies condition (4.9) based on Assumption 4.2.
- $x_k \notin \mathcal{S}_2$, which means that $x_{k+1} = (A - BK)x_k + Ew_k$ will be in \mathcal{S}_1 , for all $k \geq K(x_0)$, based on Assumption 4.4.

Therefore, there exists $K(x_0)$ such that, for any $x_0 \in \mathbb{R}^n$, x_k will be in \mathcal{S}_1 for all $k \geq K(x_0)$. This means that system (4.8) is UB within \mathcal{S}_1 , based on Definition 1.9. \square

Remark 4.4

These theorems will still hold if \mathcal{S}_2 is an empty, non-polytopic or non-convex set.

4.4.2 Analytic discussion of performances

In this section, it is assumed that \mathcal{S}_1 is a RPI set for system (4.1), \mathcal{S}_2 is an event-set for system (4.1) and \mathcal{S}_1 . It is also assumed that the initial state x_0 of system (4.1) is in the RPI set \mathcal{S}_1 . In order to assess the performances of the present strategy and compare them with those obtained with a classical approach, an upper bound is calculated for the 2-norm of i) the state x , ii) the control input u , and iii) the changing rate of the control Δu . In what follows, these indicators will be denoted δ_x , δ_u , and δ_d , respectively. In addition, the superscripts ^{es} and ^{tt} will refer to the *event-switched* and *time-triggered* systems, respectively. Variables without any superscript will be taken to refer to both the time-triggered and the event-switched systems.

4.4.2.1 Maintaining the state performances

Preserving the bound δ_x on the maximal 2-norm of the state results directly from Theorem 4.1: the fact that the systems (4.1) under time-triggered control (4.2) and event-switched control (4.7) have the same RPI set means that the maximum 2-norm of the state will be bounded by the radius of the smallest 2-norm ball containing the RPI set.

$$\|x\|_2 \leq \delta_x, \text{ with } \delta_x \triangleq \min\{\gamma \mid \mathcal{S}_1 \subseteq \mathbb{B}_2^n(\gamma)\} \quad (4.17)$$

4.4.2.2 Maintaining the control signal performances

Maintaining the amplitude Maintaining the bound δ_u also results from Theorem 4.1. In the time-triggered case, the controller (4.3) is linear, and the control input u is therefore bounded in a set \mathcal{U} which is the projection of the set \mathcal{S}_1 in the control space by the linear matrix mapping $-K$. In the event-switched case, the non-linear controller (4.7) can switch between two modes. In one mode, when $x \notin \mathcal{S}_2$, the control input $u = -Kx$ is obviously bounded in the set \mathcal{U} . In the other mode, when $x \in \mathcal{S}_2$, the control input signal is 0_m , which is in \mathcal{U} because \mathcal{S}_1 includes the origin, and the mapping between \mathcal{S}_1 and \mathcal{U} is linear.

$$u \in \mathcal{U}, \text{ with } \mathcal{U} \triangleq -K \cdot \mathcal{S}_1 \quad (4.18)$$

It follows directly that:

$$\|u\|_2 \leq \delta_u, \text{ with } \delta_u \triangleq \min\{\gamma \mid \mathcal{U} \subseteq \mathbb{B}_2^n(\gamma)\} \quad (4.19)$$

Maintaining the changing rate The changing rate of the control input is:

$$\Delta u_k = u_{k+1} - u_k \quad (4.20)$$

In the case of the time-triggered system, this gives:

$$\Delta u_k^{tt} = u_{k+1}^{tt} - u_k^{tt} \quad (4.21)$$

$$\Delta u_k^{tt} = -Kx_{k+1}^{tt} - (-Kx_k) \quad (4.22)$$

$$\Delta u_k^{tt} = -K((A_{CL} - I_n)x_k + Ew_k) \quad (4.23)$$

It follows that:

$$\Delta u_k^{tt} \in \mathcal{D}^{tt}, \forall k \in \mathbb{N} \quad (4.24)$$

$$\text{with } \mathcal{D}^{tt} \triangleq -K((A_{CL} - I_n)\mathcal{S}_1 \oplus E\mathcal{W}) \quad (4.25)$$

The bound of the 2-norm of the changing control input rate in the time-triggered system is therefore:

$$\|\Delta u_k^{tt}\|_2 \leq \delta_d^{tt}, \text{ with } \delta_d^{tt} \triangleq \min\{\gamma \mid \mathcal{D}^{tt} \subseteq \mathbb{B}_2^n(\gamma)\} \quad (4.26)$$

In the case of the event-switched system, several possibilities have to be considered:

$$u_k^{es} = \begin{cases} 0_m & \text{if } x_k \in \mathcal{S}_2 \\ -Kx_k & \text{otherwise} \end{cases} \quad (4.27)$$

$$u_{k+1}^{es} = \begin{cases} 0_m & \text{if } x_{k+1}^{es} \in \mathcal{S}_2 \\ -K(Ax_k + Ew_k) & \text{if } x_{k+1}^{es} \notin \mathcal{S}_2 \text{ and } x_k \in \mathcal{S}_2 \\ -K(A_{CL} \cdot x_k + Ew_k) & \text{if } x_{k+1}^{es} \notin \mathcal{S}_2 \text{ and } x_k \notin \mathcal{S}_2 \end{cases} \quad (4.28)$$

Therefore:

$$\Delta u_k^{es} = u_{k+1}^{es} - u_k^{es} = \begin{cases} 0_m & \text{if } x_{k+1}^{es} \in \mathcal{S}_2 \text{ and } x_k \in \mathcal{S}_2 \\ Kx_k & \text{if } x_{k+1}^{es} \in \mathcal{S}_2 \text{ and } x_k \notin \mathcal{S}_2 \\ -K(Ax_k + Ew_k) & \text{if } x_{k+1}^{es} \notin \mathcal{S}_2 \text{ and } x_k \in \mathcal{S}_2 \\ -K((A_{CL} - I_n)x_k + Ew_k) & \text{if } x_{k+1}^{es} \notin \mathcal{S}_2 \text{ and } x_k \notin \mathcal{S}_2 \end{cases} \quad (4.29)$$

It follows that:

$$\Delta u_k^{es} \in \mathcal{D}^{es}, \forall k \in \mathbb{N} \quad (4.30)$$

$$\text{with } \mathcal{D}^{es} \triangleq 0_m \cup K \cdot \mathcal{S}_1 \cup -K(A \cdot \mathcal{S}_2 \oplus E \cdot \mathcal{W}) \cup \mathcal{D}^{tt} \quad (4.31)$$

Note that the expression for \mathcal{D}^{es} can be simplified because the origin is included in \mathcal{S}_1 :

$$\mathcal{D}^{es} = K \cdot \mathcal{S}_1 \cup -K \cdot (A \cdot \mathcal{S}_2 \oplus E \cdot \mathcal{W}) \cup \mathcal{D}^{tt} \quad (4.32)$$

and since $(A \cdot \mathcal{S}_2 \oplus E \cdot \mathcal{W})$ is included in \mathcal{S}_1 , in view of condition (4.9), this means that $\min\{\gamma \mid -K \cdot (A \cdot \mathcal{S}_2 \oplus E \cdot \mathcal{W}) \subseteq \mathbb{B}_2^n(\gamma)\} \leq \min\{\gamma \mid \mathcal{D}^{tt} \subseteq \mathbb{B}_2^n(\gamma)\}$. The bound of the 2-norm of the changing control input rate in the event-switched system is therefore:

$$\|\Delta u_k^{es}\|_2 \leq \delta_d^{es}, \quad \text{with } \delta_d^{es} \triangleq \max\left(\delta_d^{tt}, \min\{\gamma \mid K \cdot \mathcal{S}_1 \subseteq \mathbb{B}_2^n(\gamma)\}\right) \quad (4.33)$$

Therefore, if $\delta_d^{tt} \geq \|K \cdot \mathcal{S}_1\|_2$, then $\delta_d^{es} = \delta_d^{tt}$. In other words, the 2-norm of the changing control input rate has the same least upper bound in both the time-triggered and the event-switched system if $\delta_d^{tt} \geq \min\{\gamma \mid K \cdot \mathcal{S}_1 \subseteq \mathbb{B}_2^n(\gamma)\}$

Remark 4.5

If \mathcal{S}_1 is the mRPI set and the event-set \mathcal{S}_2 is the largest set \mathcal{S}_2^* satisfying condition (4.9), then $(A \cdot \mathcal{S}_2 \oplus E \cdot \mathcal{W}) = \mathcal{S}_1$ and the set \mathcal{D}^{es} becomes:

$$\mathcal{D}^{es} = K \cdot \mathcal{S}_1 \cup -K \cdot \mathcal{S}_1 \cup \mathcal{D}^{es} \quad (4.34)$$

In addition, if the bound of the disturbance is symmetric, then \mathcal{S}_1 will also be symmetric, and therefore:

$$\mathcal{D}^{es} = K \cdot \mathcal{S}_1 \cup \mathcal{D}^{tt} \quad (4.35)$$

4.4.3 Observer based approach

The above approach involves checking the value of the state vector at each time step k in order to check whether or not the state of the system belongs to the event-set \mathcal{S}_2 . Depending on the result, it is then decided whether or not to apply a control signal without detracting from the performances of the scheme. However, it is assumed throughout this strategy that

all the state information is available. When the state of system (4.1) cannot be completely measured and only the output y_k is available, the complete system is modeled as follows:

$$\begin{aligned}x_{k+1} &= Ax_k + Bu_k + Ew_k \\y_k &= Cx_k\end{aligned}\tag{4.36}$$

One possible solution here is to construct a Luenberger observer (assuming that the matrix pair (A, C) is observable):

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L[y_k - C\hat{x}_k]\tag{4.37}$$

where \hat{x} is the observed state, and L the observer gain. The challenge here is then to find how to use the event-switched method presented above in the case of system (4.36)-(4.37) ensuring accurate performances.

The observation error $e_k = x_k - \hat{x}_k$ has the following dynamics:

$$e_{k+1} = (A - LC)e_k + Ew_k\tag{4.38}$$

Let \mathcal{S}_e denotes a RPI set in the observation error space, it is a set in which the observation error will be bounded once the observer has converged. The observed state actually has the following dynamics:

$$\hat{x}_{k+1} = (A - BK)\hat{x}_k + LCe_k\tag{4.39}$$

where the term e_k may be considered as a disturbance bounded in a RPI set \mathcal{S}_e . A RPI set for system (4.39) can be computed with the method presented in Section 4.1, and will be denoted $\hat{\mathcal{S}}_1$. Lastly, an event-set \mathcal{S}_2 can be found as described in Section 4.3 in the observed state space, where the complete system with an observer can be in an open-loop configuration when the observed state is in \mathcal{S}_2 , while still ensuring that $\hat{\mathcal{S}}_1$ is a RPI set for system (4.39).

4.4.4 Proposed algorithm

In the present chapter, a new approach to designing event-switched control is presented. With the present **event-switched control approach**, the system does not have to be controlled between events, and efficient performances can still be obtained even in the presence of bounded disturbances. A switching condition is checked periodically in order to determine whether or not a dynamic control input has to be applied. Figure 4.1 presents the overall control strategy.

The first step in the present method consists in finding the nominal performances of the controlled system subjected to bounded disturbances. The nominal performances correspond to a RPI set for the system. We take \mathcal{S}_1 to denote this set. Note that as \mathcal{S}_1 is a RPI set, once the state has entered this set, it will remain there. In order to be

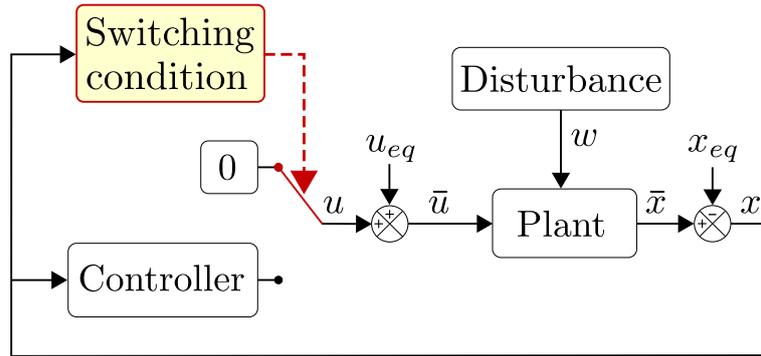


Figure 4.1: Event-switched control strategy for a plant with an exogenous disturbance w . The switching condition is used to determine whether an action has to be performed by the controller or not.

less conservative, \mathcal{S}_1 can be approximated as the mRPI set. The second step consists in finding an event-set \mathcal{S}_2 . When the state of the system is located inside \mathcal{S}_2 , the control input will switch off and the system will run in the (possibly unstable) open-loop mode. Whenever the state no longer belongs to \mathcal{S}_2 , the control input switches on again. Figure 4.2 illustrates the whole process in the case of a 2-dimensional system (in order to simplify the spatial representation). It is worth noting that the event-set \mathcal{S}_2 is not included in \mathcal{S}_1 in this particular example, $\mathcal{S}_2 \setminus \mathcal{S}_1$ can only be reached one time step before joining \mathcal{S}_1 . Three different cases arise here:

- 1: When the state does not belong to both the nominal performance set \mathcal{S}_1 and the event-set \mathcal{S}_2 , the linear control input is applied to the system and the state will join either \mathcal{S}_1 or \mathcal{S}_2 after a finite number of time steps.
- 2: When the state belongs to \mathcal{S}_1 but not to \mathcal{S}_2 , the linear control input is applied. At the next time step, the state will remain in \mathcal{S}_1 and can join $\mathcal{S}_2 \cap \mathcal{S}_1$.
- 3: When the state belongs to \mathcal{S}_2 , the control input is switched off, and only the equilibrium value u_{eq} is applied to the control input (or 0_m in the null stabilization case). Due to the mathematical properties of \mathcal{S}_2 , the state will belong to \mathcal{S}_1 at the next time step (and can also continue to belong to $\mathcal{S}_2 \cap \mathcal{S}_1$).

Lastly, designing an event-switched control strategy with the approach presented here can be done using a systematic method. Given any stabilizable system (4.1):

- Find a linear state-feedback K such that the eigenvalues of $A - BK$ are inside the unit circle.
- Compute an approximation for the mRPI set of the system. See [Rakovic et al., 2005] for an efficient method.

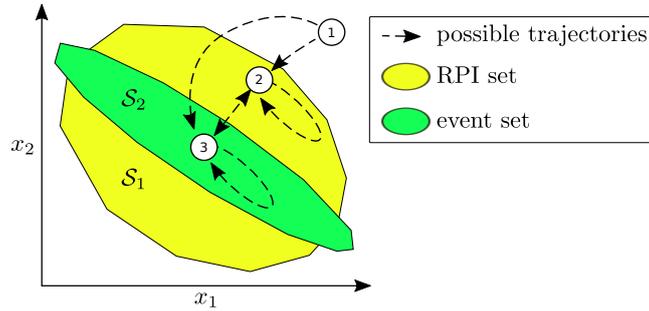


Figure 4.2: Geometrical interpretation of the event-switched control strategy illustrating three different possible state trajectories of the event-switched system. \mathcal{S}_1 is a RPI set and \mathcal{S}_2 is the event-set. It is worth noting that \mathcal{S}_2 is not included in \mathcal{S}_1 .

- Compute the event-set \mathcal{S}_2^* as explained in Section 4.3.
- Take any set $\mathcal{S}_2 \subseteq \mathcal{S}_2^*$ as an event-set.

All these steps are performed to compute an event condition which ensures efficient performances. This event condition can be checked online as described in Algorithm 1.

Algorithm 1 Applying the event-switched control ONLINE

Require: event-set \mathcal{S}_2

At each time step:

if $x \in \mathcal{S}_2$ **then**

 Apply open-loop control $u = 0$

else

 Apply closed-loop control $u = -K \cdot x$

end if

We recall that the event-set \mathcal{S}_2 is computed offline, then the inclusion condition checked at each sampling time is very simple to implement on practical applications. The computational effort required for this step depends on the dimension and the topology of the event-set. If the event-set is a polyhedron, the complexity is related to the number of vertices describing the set. To be more precise, it will consist in verifying as many inequalities as the polyhedron has faces. If the set is an ellipsoid, then the complexity is quadratic in the order of the system. The use of an ellipsoidal event-set was discussed in Section 4.3.1, the results obtained with an implementation of this method will be illustrated in Section 4.5.3.

The cost of the offline computation is highly related to the considered system. The time required for this computation is given in Section 4.5 for a given example.

4.5 Simulation of the event-switched approach

In this section, the event-switched control approach presented above is applied to the following academic example:

$$x_{k+1} = \underbrace{\begin{bmatrix} 1 & 0.65 \\ 0 & 1 \end{bmatrix}}_A x_k + \underbrace{\begin{bmatrix} 0.211 \\ 0.65 \end{bmatrix}}_B (u_k + w_k) \quad (4.40)$$

Note that in this particular case, the disturbance matrix E is taken to be equal to B .

A stabilizing static state-feedback gain was found using the LQ optimal control method with identity weighting matrices of suitable dimension.

$$u_k = - \underbrace{\begin{bmatrix} 0.575 & 1.217 \end{bmatrix}}_K x_k \quad (4.41)$$

The dynamics of the linear discrete time-triggered closed-loop system are therefore:

$$x_{k+1} = A_{CL}x_k + Bw_k \quad (4.42)$$

with $A_{CL} \triangleq A - BK = \begin{bmatrix} 0.879 & 0.393 \\ -0.374 & 0.209 \end{bmatrix}$

It can be noticed that the poles ($0.54 \pm 0.19 \cdot i$) of the closed-loop system are located inside the unit circle. The system is therefore stable, whereas the open-loop system is unstable. During the simulation, w will be bounded in the set $\mathcal{W} = [-1; +1]$.

Both classical time-triggered closed-loop system (4.42) and system (4.40) subjected to switched control input (4.7) are simulated in parallel, in order to compare the performances of these two systems when the same disturbance is applied. The results of the simulation during the first 10 time steps (in order to clearly show the trajectory of the system) are presented in Figure 4.3. The nominal performance set \mathcal{S}_1 is an approximation of the mRPI set obtained via the algorithm given in [Kolmanovsky and Gilbert, 1998]. The set \mathcal{S}_2^* is obtained as described in Section 4.3. In our implementation using MATLAB and the MPT toolbox [Herceg et al., 2013], the time required to compute these two sets was approximately 0.8 seconds. These sets, as well as the system trajectories, are plotted in Figure 4.3a. The control inputs are also presented with respect to the time steps in Figure 4.3b. It can be seen that the control signal is 0 when the system trajectory is in the set \mathcal{S}_2^* at time steps 3, 4, 5, 7 and 8.

In order to test the accuracy of the mRPI set approximation, the discrete time system (4.42) was simulated with the disturbance reaching the bounds. The results obtained, which are presented in Figure 4.4, clearly show that the mRPI set approximation is satisfactory, since the state x still belongs to the set \mathcal{S}_1 with the worst disturbance scenario, and is located very near the vertices.

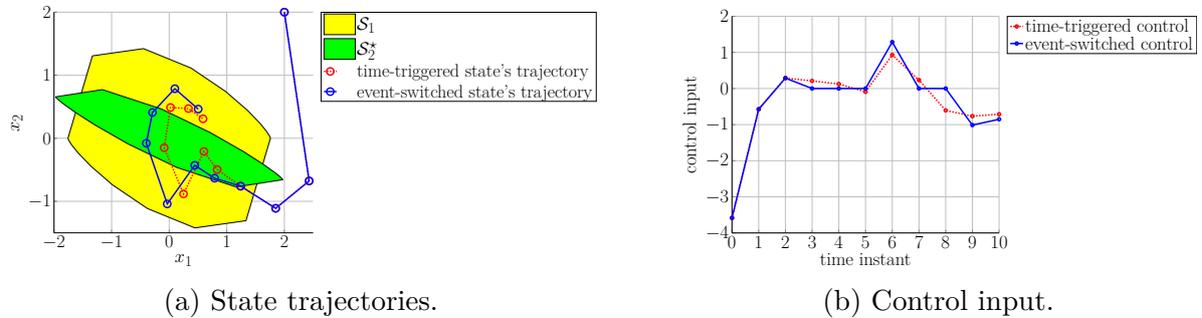


Figure 4.3: Simulation of the time-triggered system (in red) and the event-switched system (in blue) with the same disturbances and initial condition ($x_1 = x_2 = 2$) during 10 time steps. When the state of the event-switched system is in \mathcal{S}_2^* (green set), open-loop control is applied. The set \mathcal{S}_1 (yellow) describes a RPI set for both the time-triggered and the event-switched systems.

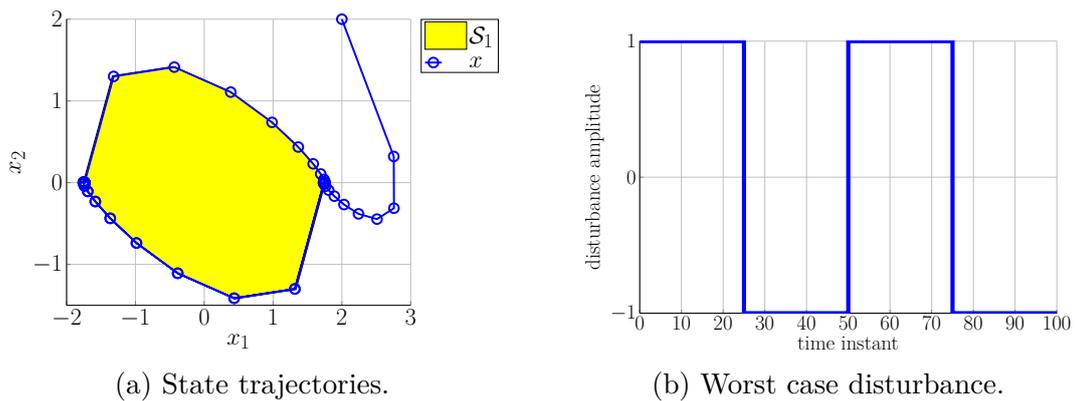


Figure 4.4: Worst disturbance scenario for mRPI approximation (\mathcal{S}_1) verification.

In the rest of this section, the simulation was run for $N = 10^7$ periods of time in order to have a sufficiently good distribution of the disturbances and reliable performance indexes, as described in the next section.

4.5.1 Performance indexes

The following performance indexes are used to assess the efficiency of the process:

- The maximum 2-norm of the state: $\max(\|x\|_2)$, which is bounded by δ_x in (4.17);
- The maximum 2-norm of the control input: $\max(\|u\|_2)$, which is bounded by δ_u in (4.19);
- The maximum 2-norm of the rate of variation of the control input: $\max(\|\Delta u\|_2)$, which is bounded by δ_d in (4.24) and (4.33);
- The percentage time spent in the closed loop: %c (this index also gives the rate of data bus use between the controller and the actuator in the case of networked control systems). The maximum value is 100% (in the case of time-triggered control), and the aim is to reduce this percentage with the event-switched control strategy presented here.

In this particular example, the bound δ_d on $\|\Delta u\|_2$ is the same in the case of both the time-triggered and the event-switched system because $\delta_d^{tt} \geq \min\{\gamma \mid K \cdot \mathcal{S}_1 \subseteq \mathbb{B}_2(\gamma)\}$, refer to Section 4.4.2.2 for further details.

The bound values are given in Table 4.1. The performance indexes obtained in the simulations are given in Tables 4.2-4.3 and discussed below. Note that the case of event-switching strategy with \mathcal{E}^* will be introduced later in Section 4.5.3.

δ_x	δ_u	δ_d
1.87	1.57	1.97

Table 4.1: Theoretical bounds on performance indexes.

	%c	$\max(\ x\ _2) \leq \delta_x$	$\max(\ u\ _2) \leq \delta_u$	$\max(\ \Delta u\ _2) \leq \delta_d$
Time-triggered strategy	100	1.69	1.48	1.92
Event-switching strategy with \mathcal{S}_2^*	48.31	1.71	1.57	1.95
Event-switching strategy with \mathcal{E}^*	49.66	1.70	1.52	1.95

Table 4.2: Performance indexes obtained with uniformly distributed disturbances.

	%c	$\max(\ x\ _2) \leq \delta_x$	$\max(\ u\ _2) \leq \delta_u$	$\max(\ \Delta U\ _2) \leq \delta_d$
Time-triggered strategy	100	1.58	1.42	1.79
Event-switching strategy with \mathcal{S}_2^*	39.62	1.64	1.55	1.90
Event-switching strategy with \mathcal{E}^*	39.44	1.64	1.52	1.90

Table 4.3: Performance indexes obtained with truncated normally distributed disturbances.

4.5.2 Results

Figure 4.5 shows the results of the simulation when a uniformly distributed disturbance is applied. The simulated reachable set denoted \mathcal{S}_x after 10^7 time steps starting at the origin in the case of the time-triggered system can be seen in Figure 4.5a, and in the case of the event-switched system, in Figure 4.5b. \mathcal{S}_x is the convex hull where the state remains during the simulation. It can also be noted from Figure 4.5b that the state continues to stay in \mathcal{S}_1 throughout the simulation, as \mathcal{S}_x is included in \mathcal{S}_1 . This is the expected behavior of the present event-switched control method as \mathcal{S}_1 is a RPI set for both the time-triggered and the event-switched systems.

The performance indexes introduced in Section 4.5.1 are presented in Tables 4.2 and 4.3 in the case of uniform and truncated normal disturbance distributions, respectively. The results show that the 2-norms of the state are bounded as predicted by δ_x , as explained in Section 4.4.2.1. This can also be seen with a geometrical interpretation, as \mathcal{S}_x is included in \mathcal{S}_1 . The control input and the changing rate of the control input are also bounded by δ_u and δ_d , respectively. However, it can be seen that in the event-switched case, the reachable set \mathcal{S}_x is slightly larger than the one obtained using the time-triggered control. This difference may be attributable to the random nature of the disturbances and their impact on the bound of the reachable set \mathcal{S}_x : see for instance probabilistic invariant sets for linear systems [Kofman et al., 2012]. It is recalled, however, that the state trajectories occurring in the worst disturbance scenario, which are shown in Figure 4.4a, belong to the boundaries of the set \mathcal{S}_1 in both cases.

It is also worth considering the frequency domain of the state and the control input in the above simulation. Figure 4.6 shows the power spectral density (PSD) of the control input u , and the states x_1 and x_2 (where $x^T = [x_1 \ x_2]$). It can be noted that the magnitude of these values is generally greater in the case of the event-switched control system than with the classical approach. However, the overall shape is similar in both cases. It can be seen from Figure 4.6a that the magnitude of the control input is greater at high frequencies in the case of the event-switched method. But in x_1 in Figure 4.6b and x_2 in Figure 4.6c, lower values were obtained at high frequencies in the case of the event-switched strategy. This is a particularly noteworthy result of the simulation: applying unstable dynamics to some events can result in less energy in the high frequency domain of the state.

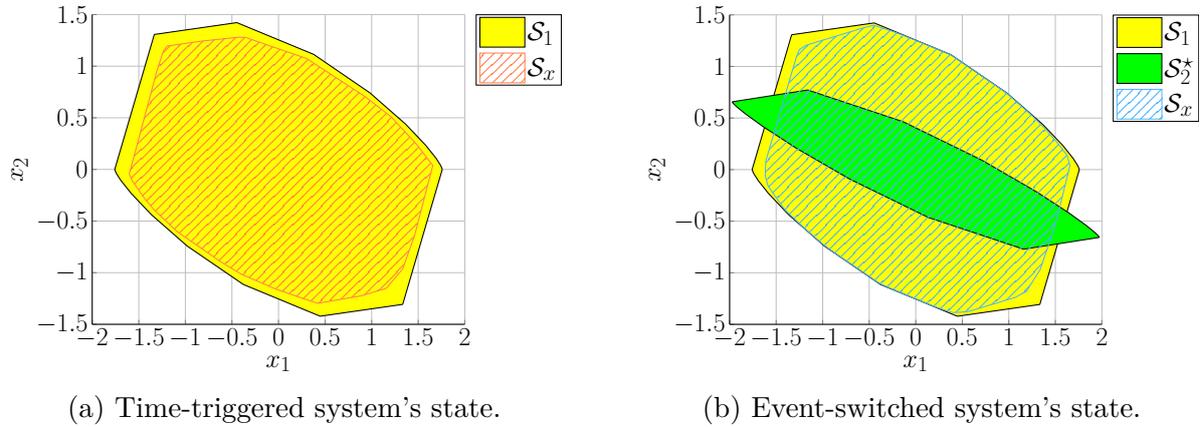


Figure 4.5: Comparison between time-triggered and event-switched systems subjected to the same uniformly distributed disturbances. \mathcal{S}_1 is an approximation for the mRPI set computed offline, as mentioned in Section 4.1, and \mathcal{S}_x is the reachable set from the origin after 10^7 periods of time.

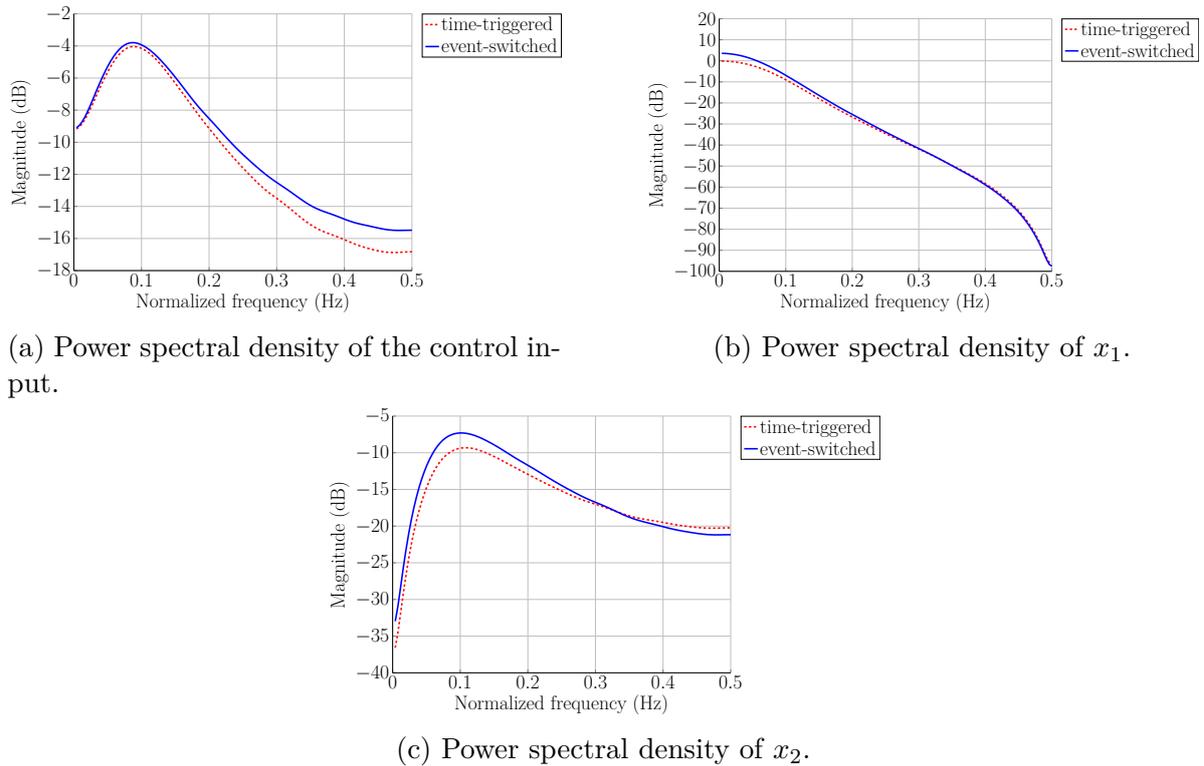


Figure 4.6: Power spectral density comparison between time-triggered and event-switched systems.

4.5.3 Reducing computation complexity

To use the switching mechanism (4.7), one has to check at each time instant whether the state belongs to $\mathcal{S}_2^* = \{x \in \mathbb{R}^n : Rx \leq Q\}$ with $R \in \mathbb{R}^{f \times n}$ and $Q \in \mathbb{R}^f$. For this purpose, one only needs to check whether the state meets the condition imposed in the set definition: $Rx \leq Q$. In the above example, the set \mathcal{S}_2^* is a polyhedron defined by the intersection between 20 half-spaces, which means calculating $2 \times 20 = 40$ multiplications, 20 additions and checking 20 inequalities at each time step.

To simplify the control algorithm, another event-set can be defined inside \mathcal{S}_2^* , such as the largest ellipse inscribed in \mathcal{S}_2^* . In the above example, this would reduce the number of operations to 6 multiplications, 3 additions and 1 inequality per time step. In return, this means that the resulting event-switched system would switch to open-loop dynamics less frequently as the volume of the event-set is smaller.

Computing the largest ellipsoid in a polyhedron is a convex optimization problem, as stated in [Boyd and Vandenberghe, 2004, Section 8.4.1]. Let define the ellipse \mathcal{E}^* as follows:

$$\mathcal{E}^* \triangleq \{Tx + d \mid \|x\| \leq 1\}, \text{ with } T \text{ a symmetric matrix} \quad (4.43)$$

In order to maximize the volume of \mathcal{E}^* , the constraint which has to be met is that every element of \mathcal{E}^* has to satisfy the inequality defining \mathcal{S}_2^* . This could be written as follows:

$$\begin{aligned} & \underset{T,d}{\text{maximize}} && \log(\det T) \\ & \text{subject to} && \|TR\| + Rd \leq Q, \text{ and } T \text{ symmetric} \end{aligned} \quad (4.44)$$

\mathcal{E}^* can also be defined in a more popular form than (4.43), as follows:

$$\mathcal{E}^* \triangleq \{(x - d)^T S (x - d) \leq 1\}, \text{ with } S = T^{-2} \quad (4.45)$$

The event-set \mathcal{E}^* thus obtained is presented in Figure 4.7. A simulation was run with the event-set denoted \mathcal{E}^* , and the results obtained are presented in Tables 4.2 and 4.3. As expected, the total amount of time during which the plant was allowed to be in the open-loop mode was shorter than with the event-set \mathcal{S}_2^* . However, the difference amounted to only about 1%, which means that the advantages of the present method were not greatly decreased. The performance indexes obtained were almost the same as in Section 4.5.2. Choosing the set where the system is allowed to be in the open loop mode therefore amounts to making a trade-off between the complexity of the event conditions and the performances.

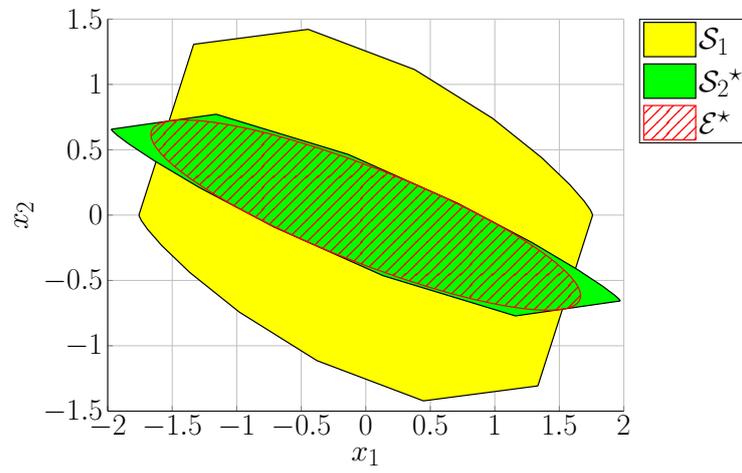


Figure 4.7: The set \mathcal{E}^* is the maximum volume ellipse included in \mathcal{S}_2^* .

Experimental setup

Contents

5.1	Objectives	57
5.2	Quadcopter	58
5.2.1	Euler formalism	58
5.2.2	Quadcopter modeling	61
5.3	Flying arena and experimental setup	63
5.3.1	Position sensor	64
5.3.2	Control toolchain	65
5.3.3	Onboard control	66
5.4	Quadcopter setup	66
5.4.1	The X4-MaG platform	66
5.4.2	The Flexbot platform	68
5.4.3	The AVE platform	69
5.4.4	The Inductrix platform	69
5.4.5	The TACO platform	71
5.4.6	Motor control	71
5.5	Platform choice	74

5.1 Objectives

The aim of this chapter is to present the different parts involved in an experimental application of the event-switched approach that will be presented in the next chapter. This chapter is also dedicated to detail the technical developments done during this PhD.

5.2 Quadcopter

A quadcopter platform has been chosen for our experiments for its agility and for the few components required to build such a platform. This type of platform has been widely studied thus it has the benefit of hindsight in the sense that trusted model and controller can be found in the literature. Furthermore quadcopter belongs to the class of Vertical Take Off and Landing (VTOL) aircraft which means that the necessary space for maneuvers is limited to small spaces.

A quadcopter is typically made of several parts, that is:

- A rigid structure,
- A flight controller board,
- A power source (battery),
- A receiver,
- Four motors with propellers eventually with ESC¹.

Two opposite rotors are spinning clockwise and the other two are spinning anticlockwise, as shown on Figure 5.1, in order to obtain zero torque when all the rotors are at the same speed. The usual way for a person to pilot a quadcopter with a remote control is to command the attitude and the thrust of the UAV². The control is achieved by the differential control of each rotor's speed. Thus torques can be created around each axis and the total thrust can be controlled.

5.2.1 Euler formalism

This section introduces the Euler formalism which will be used in the sequel. Let define $\{\vec{x}, \vec{y}, \vec{z}\}$ as unit vectors along each axis. A direct orthonormal inertial frame will be denoted $\{\mathbb{I}\}$ for which the unit vectors are denoted by $\{\vec{i}_1, \vec{i}_2, \vec{i}_3\}$. A second frame denoted by $\{\mathbb{B}\}$ is attached to the quadrotor such that the origin corresponds to its center of gravity. Unit vectors of $\{\mathbb{B}\}$ will be denoted $\{\vec{b}_1, \vec{b}_2, \vec{b}_3\}$. These vectors are the axis of frame $\{\mathbb{B}\}$ with respect to the frame $\{\mathbb{I}\}$. The axes of the body frame $\{\mathbb{B}\}$ are defined as follows: the x axis is between the motors 1 and 2, the y axis is between the motors 2 and 3, and the z axis is pointing from the bottom up. A rotation matrix $R \in SO(3)$ will describe the orientation of the body frame $\{\mathbb{B}\}$ relatively to the inertial frame $\{\mathbb{I}\}$. Thus, by construction: $\vec{b}_1 = R\vec{i}_1$, $\vec{b}_2 = R\vec{i}_2$, and $\vec{b}_3 = R\vec{i}_3$.

¹Electronic Speed Controller

²Unmanned Aerial Vehicle

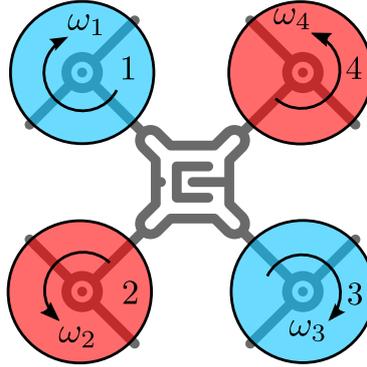


Figure 5.1: Schematic view of a quadcopter, motors 1 and 3 are spinning clockwise at the speed ω_1 and ω_3 respectively, while motors 2 and 4 are spinning anticlockwise at the speed ω_2 and ω_4 respectively.

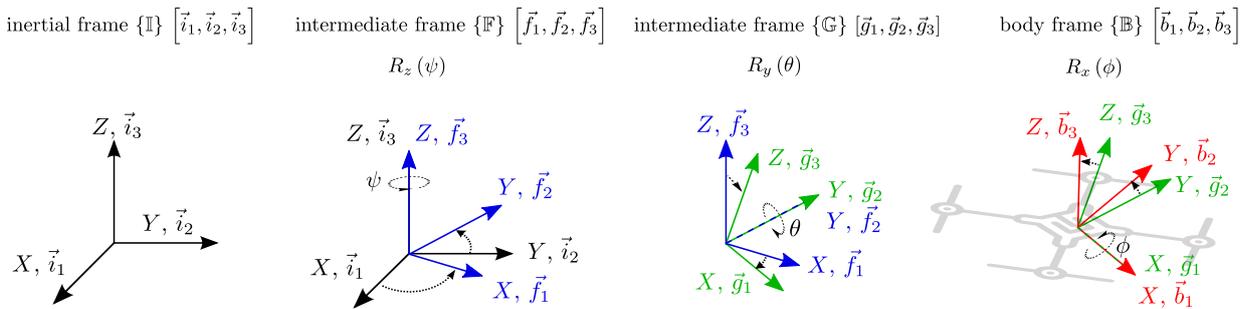


Figure 5.2: The Euler angles ϕ , θ , and ψ are defined as the three angles transforming the inertial frame into the body frame by a combination of three successive rotations.

The orientation of the frame $\{\mathbb{B}\}$ with respect to the frame $\{\mathbb{I}\}$ is fully described by the rotation matrix R . Any orientation of the frame $\{\mathbb{B}\}$ can be reached by rotating the frame $\{\mathbb{I}\}$ three times by the angle ϕ (roll), θ (pitch), and ψ (yaw). However these angles are different, depending on the convention used to construct the rotation matrix R . Here, the Z-Y-X intrinsic Tait–Bryan angles convention will be used. Intrinsic rotation means that rotations are applied sequentially around the axis of the rotated frame, as opposed to the extrinsic convention where each rotation is applied around the axis of the inertial frame. The frame $\{\mathbb{I}\}$ is obtained after three rotations of the frame $\{\mathbb{B}\}$. Firstly $\{\mathbb{B}\}$ is rotated around its axis z by an angle ψ , this results in an intermediate frame $\{\mathbb{F}\}$. Secondly this frame is rotated around the axis y of $\{\mathbb{F}\}$ by an angle θ , the resulting frame is denoted $\{\mathbb{G}\}$. This new frame is then rotated around the axis z of $\{\mathbb{G}\}$ by an angle ϕ . This sequence of rotation applied to the inertial frame $\{\mathbb{I}\}$ gives the body frame $\{\mathbb{B}\}$ as illustrated in Figure 5.2.

Given the previously presented convention, the rotation matrix R is expressed as follows:

$$R = R_z(\psi) \cdot R_y(\theta) \cdot R_x(\phi) \quad (5.1)$$

$$R = \underbrace{\begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{R_z(\psi)} \cdot \underbrace{\begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}}_{R_y(\theta)} \cdot \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{pmatrix}}_{R_x(\phi)} \quad (5.2)$$

$$R = \begin{pmatrix} \cos \psi \cdot \cos \theta & \cos \psi \cdot \sin \theta \cdot \sin \phi - \sin \psi \cdot \cos \phi & \cos \psi \cdot \sin \theta \cdot \cos \phi + \sin \psi \cdot \sin \phi \\ \sin \psi \cdot \cos \theta & \sin \psi \cdot \sin \theta \cdot \sin \phi + \cos \psi \cdot \cos \phi & \sin \psi \cdot \sin \theta \cdot \cos \phi - \cos \psi \cdot \sin \phi \\ -\sin \theta & \cos \theta \cdot \sin \phi & \cos \theta \cdot \cos \phi \end{pmatrix} \quad (5.3)$$

Remark 5.1

Note that every intrinsic rotation has an extrinsic equivalent. For example, the X-Y-Z extrinsic rotation is equivalent to the Z-Y-X intrinsic convention used here.

Rotation speed vector of the body frame $\{\mathbb{B}\}$ denoted $[p \ q \ r]^T$ is related to the angular speed of the Euler angles $[\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_x(\phi) \begin{bmatrix} \dot{\theta} \\ 0 \\ 0 \end{bmatrix} + R_x(\phi) \cdot R_y(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (5.4)$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = J^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (5.5)$$

Inverting the matrix J^{-1} allows to deduce the derivative of the Euler angles from the angular rotation speed of the body frame:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = J \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (5.6)$$

$$J = \begin{pmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{pmatrix} \quad (5.7)$$

Note that J is not defined when the pitch angle θ is equal to 90° . Furthermore, when

the angles ϕ and θ are small enough the following approximation can be made:

$$J \approx \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (5.8)$$

Remark 5.2

The quaternion formulation can also be used (see e.g. [Altmann, 2005, Chapter 12]). Rotation is represented by a complex vector instead of a real matrix. The singularity of which rotation matrix suffers is avoided. However, this formulation will not be studied here.

5.2.2 Quadcopter modeling

In this section a physical model of a quadcopter is presented inspired by different models found in the literature (e.g. [Hamel et al., 2002], [Pounds et al., 2010], [Bouabdallah et al., 2004]).

As detailed in [Mahony et al., 2012], considering the quadcopter as a rigid body we can write:

$$v = \dot{\xi} \quad (5.9)$$

$$m\dot{v} = -mg\vec{i}_3 + F \quad (5.10)$$

$$\dot{R} = R\Omega_{\times} \quad (5.11)$$

$$I\dot{\Omega} = -\Omega \times I\Omega + \tau \quad (5.12)$$

where $\xi = [x \ y \ z]^T$ denotes the position of the quadrotor center of gravity in the inertial frame $\{\mathbb{I}\}$, and $v = [v_x \ v_y \ v_z]^T$ its speed. m denotes the mass of the UAV, g the gravity constant, $I \in \mathbb{R}^3$ the constant inertia matrix, $\Omega = [p \ q \ r]^T$ the angular velocity of the body frame $\{\mathbb{B}\}$ with respect to $\{\mathbb{I}\}$. F expresses all the non conservative forces acting on the UAV and τ all the moments introduced by the rotors. The term $-\Omega \times I\Omega$ denotes the gyroscopic effect due to the rotation of the rigid body.

Ω_{\times} denotes the skew-symmetric matrix, such that $\Omega_{\times}h = \Omega \times h$ for any vector $h \in \mathbb{R}^3$ where \times stands for the vector cross product:

$$\Omega_{\times} = \begin{pmatrix} 0 & p & -q \\ -p & 0 & r \\ q & -r & 0 \end{pmatrix} \quad (5.13)$$

5.2.2.1 Forces acting on the UAV

The major nonconservative force acting on the robot is due to the total thrust T_Σ generated by the rotation of the propellers. T_Σ can be expressed as the sum of the thrust T_i of all the individual rotors i :

$$T_\Sigma = \sum_{i=1}^4 T_i \cdot \vec{b}_3 \quad (5.14)$$

$$T_i = c_T \omega_i^2 \quad (5.15)$$

where ω_i is the rotation speed of the rotor i and $c_T > 0$ is a constant depending upon the air density and the rotor geometry.

However because of the speed of the UAV, an apparent wind alters the shape of the propeller. As a result the thrust generated by the propeller is not orthogonal to the hub of the rotor. This is called the flapping effect, and it has been widely studied for helicopter (e.g. [Prouty, 1986], [Johnson, 1994]) or more recently for quadrotor [Hoffmann et al., 2007]. The force for each rotor can be modeled by applying a rotation matrix R_{flap_i} to the equation 5.15:

$$F_i = R_{flap_i} T_i \cdot \vec{b}_3 \quad (5.16)$$

where R_{flap_i} depends on the direction of the propellers, the angular velocity and the speed of the UAV. Approximation of this matrix can be found in [Prouty, 1986].

In the sequel the flapping effect will not be considered, therefore we will always consider $R_{flap} = I_3$. As a result the forces acting on the UAV can be written as:

$$F = R T_\Sigma \cdot \vec{b}_3 = R \sum_{i=1}^4 (c_T \omega_i^2 \cdot \vec{b}_3) \quad (5.17)$$

5.2.2.2 Moments acting on the UAV

The rotors are spinning and produce a thrust, due to the distance l between the center of the UAV and the propeller, a resulting moment is produced around x and y axis of the body frame $\{\mathbb{B}\}$:

$$\tau_x = \bar{l} \cdot c_T (\omega_2^2 + \omega_3^2 - \omega_1^2 - \omega_4^2) \quad (5.18)$$

$$\tau_y = \bar{l} \cdot c_T (\omega_3^2 + \omega_4^2 - \omega_1^2 - \omega_2^2) \quad (5.19)$$

where:

$$\bar{l} = \frac{l\sqrt{2}}{2} \quad (5.20)$$

Please remark that the flapping effect has been ignored.

A moment around z axis of the body frame $\{\mathbb{B}\}$ is produced due to the drag of the propellers and is modeled as:

$$\tau_z = c_Q (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \quad (5.21)$$

where $c_Q > 0$ is the constant drag coefficient of the rotors.

Therefore, all the moments can be written in a matrix form:

$$\tau = \begin{pmatrix} \tau_x \\ \tau_y \\ \tau_z \end{pmatrix} = \begin{pmatrix} -\bar{l} \cdot c_T & \bar{l} \cdot c_T & \bar{l} \cdot c_T & -\bar{l} \cdot c_T \\ -\bar{l} \cdot c_T & -\bar{l} \cdot c_T & \bar{l} \cdot c_T & \bar{l} \cdot c_T \\ c_Q & -c_Q & c_Q & -c_Q \end{pmatrix} \cdot \begin{pmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{pmatrix} \quad (5.22)$$

Combining the equations (5.17) and (5.22) in a matrix form gives the relation which links the thrust T and the moments τ to the speed of each motor:

$$\begin{pmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{pmatrix} = \underbrace{\begin{pmatrix} c_T & c_T & c_T & c_T \\ -\bar{l} \cdot c_T & \bar{l} \cdot c_T & \bar{l} \cdot c_T & -\bar{l} \cdot c_T \\ -\bar{l} \cdot c_T & -\bar{l} \cdot c_T & \bar{l} \cdot c_T & \bar{l} \cdot c_T \\ c_Q & -c_Q & c_Q & -c_Q \end{pmatrix}}_{\Gamma} \cdot \begin{pmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{pmatrix} \quad (5.23)$$

5.3 Flying arena and experimental setup

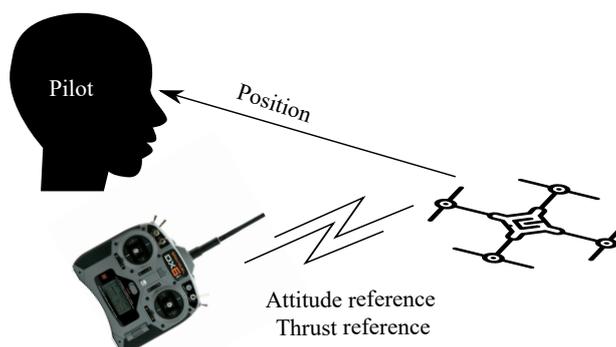
Our goal is now to perform an automatic flight of a quadcopter, in other words our aim is to replace the pilot with a controller. As depicted in Figure 5.3a, a human pilot is using visual feedback to decide how to act on the remote control in real-time. Two kinds of control modes are widely spread:

- **Angle mode** where the desired angles ϕ^* , θ^* , the angular rotation speed of the yaw $\dot{\psi}^*$ and the desired thrust T^* are directly mapped to the position of the stick.
- **Acrobatic mode** where the user controls the desired thrust T^* and the UAV angle rate of change $(\dot{\phi}^* \quad \dot{\theta}^* \quad \dot{\psi}^*)^T$.

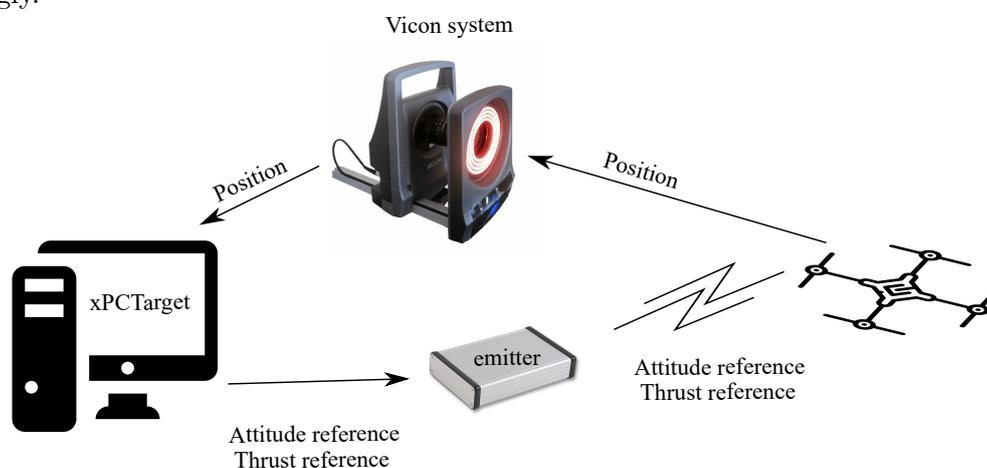
In the sequel, Angle mode is assumed to be used.

A motion capture system from Vicon³ will be used to get the position and the orientation of the UAV in order to compute which setpoints have to be sent to the quadcopter

³www.vicon.com



(a) To perform a flight, the human pilot is getting a visual feedback and act on the remote control accordingly.



(b) The Vicon system is used as a visual feedback to perform an automatic flight, a controller is running in real-time on a ground station to calculate which control input has to be sent to the UAV.

Figure 5.3: Schematic view of the control process of a human piloted UAV and an automatic flight system using Vicon system.

as depicted on Figure 5.3b. The room MOCA (which stands for MOtion Capture Area) at GIPSA-Lab shown in Figure 5.4, is fully equipped to perform real-time control of multicopters. In this room, a volume of 5.6m x 6.6m x 2.9m is available as a total flying area.

5.3.1 Position sensor

To retrieve the real-time position of a multicopter in real-time, the room MOCA is equipped with 12 Vicon cameras which are able to compute in real-time the position and the orientation of any object. The cameras emit modulated infrared light. Markers have to be fixed



Figure 5.4: MOCA flying arena at GIPSA-lab.

on the object (at least 3 markers in a non symmetrical configuration). After a calibration process, the reflection of the markers permits to retrieve the position and the orientation of the object at a maximum sampling frequency of 500 Hz. A dedicated platform performs the computation and sends the results through the network.

5.3.2 Control toolchain

Real-time control of the quadcopter is done using a ground station running xPC Target. This computer receives the information from the Vicon system through UDP packets, computes the corresponding control $\Lambda^* = (\phi^* \ \theta^* \ \psi^*)^T$ and sends it to an emitter via UDP packets. This framework is depicted in Figure 5.3b.

The emitter receives UDP packets from the ground station and sends a radio signal compatible DSM2, which is a proprietary wireless communication protocol popular among the hobbyist. This standard uses a 2.4 GHz band radio link. The emitter has first to be bind with a particular receiver, the bind procedure is launched by sending a specific UDP packet to the emitter. Once the binding is done, a simplex communication channel is effective between the real-time controller and the UAV. Simplex means that the communication can only occur in one direction, in other words the UAV can not send informations through this radio link. The emitter has been developed in GIPSA-Lab by Julien Minet, research engineer.

5.3.3 Onboard control

Once the UAV's receiver gets the setpoint Λ^* sent by the ground station, the FCB⁴ has to control the motors in order to achieve the desired orientation and thrust.

The FCB gets on board an IMU⁵ to reconstruct in real-time the attitude of the UAV. The IMU consists in an accelerometer to detect orientation and acceleration, and a gyroscope to detect rotation in space. Eventually, a magnetometer would be also used to get the direction of the magnetic north. Usually an extended Kalman filter or a complementary filter is used to reconstruct the attitude with the data received from the IMU. Given the estimation of the orientation, a controller is used to compute the desired torques around each axis.

Different control approaches can be found in the literature. The most common one among the hobbyist community is to use two PI control-loops. The inner PI is used to control the rotational speed, while the outer loop gives angular speed reference to the inner loop from the error between the desired angles and the estimated angles. Inverting the matrix Γ in (5.23) relates the rotation speed of the rotors to the thrust and the torques.

The FCB will be considered as a black box system whose inputs are the setpoints transmitted by the UAV's receiver and the output are the drone angles. We will refer as *flight controller* the onboard program running on the FCB.

5.4 Quadcopter setup

Different UAVs have been used and developed during my thesis. This section aims to describe all the different platforms I have used and contributed to.

5.4.1 The X4-MaG platform

The X4-MaG is a platform which has been developed jointly between GIPSA-Lab and the biorobotics research team of ISM (Institut des Sciences du Mouvement) during the PhD work of Augustin Manecy [Manecy, 2015]. This platform offers a high modularity. Different levels of complexity can be handled, depending on the chosen hardware setup. Three dedicated electronic boards can be jointly used:

- A Nanowii, which is an arduino compatible board largely used as a FCB;

⁴Flight Controller Board

⁵Inertial Measurement Unit

- A RCB⁶, a specific electronic board developed in the biorobotics lab by Marc Boyron, research engineer. This component aims at performing a close-loop control of the motor speed;
- A Gumstix Overo AirSTORM board, which is a powerful board running an embedded linux operating system and providing WiFi connectivity.

With these three components, three different configurations can be used:

- Nanowii only: This is the basic configuration where the Nanowii is directly connected to the ESC. The attitude reconstruction and the attitude control are directly computed on this board.
- Nanowii and RCB: The attitude reconstruction and control are still computed on the Nanowii which sends motor speed references to the RCB. The speed of the motors is measured by Hall effect sensors and a controller is implemented on the RCB to achieve the required speed.
- Nanowii, RCB and Gumstix: The Nanowii sends the IMU informations to the Gumstix, which performs the computation of the attitude reconstruction and control. Then, the Gumstix sends motor speed references to the RCB.

A specific open-source toolbox called RT-MaG has been developed to easily program the Gumstix and also to generate real-time code for the ground station. C code is produced from a Simulink diagram, and the compilation is done directly on the Gumstix, which avoids the setup of a cross compilation toolchain. A Quarc toolbox is used to run real-time code on the ground station. The ground station is used to send the position information received from the Vicon system to the UAV, and also to visualize and log the data sent by the Gumstix. Communication between all the boards are done with serial protocol, the built-in receiver of the Nanowii is using DSM2 protocol while the Gumstix use standard WiFi.

The main weaknesses of X4-MaG platform are:

- The complexity of having an onboard computer. On one hand, it offers a great computing power, on the other hand, it is more subject to bug. While it offers a high level of abstraction for programming from a Simulink diagram, it also requires a sophisticated compilation toolchain which can be more complicated to set up.
- The complexity of having three different boards communicating.

⁶Rotors Control Board

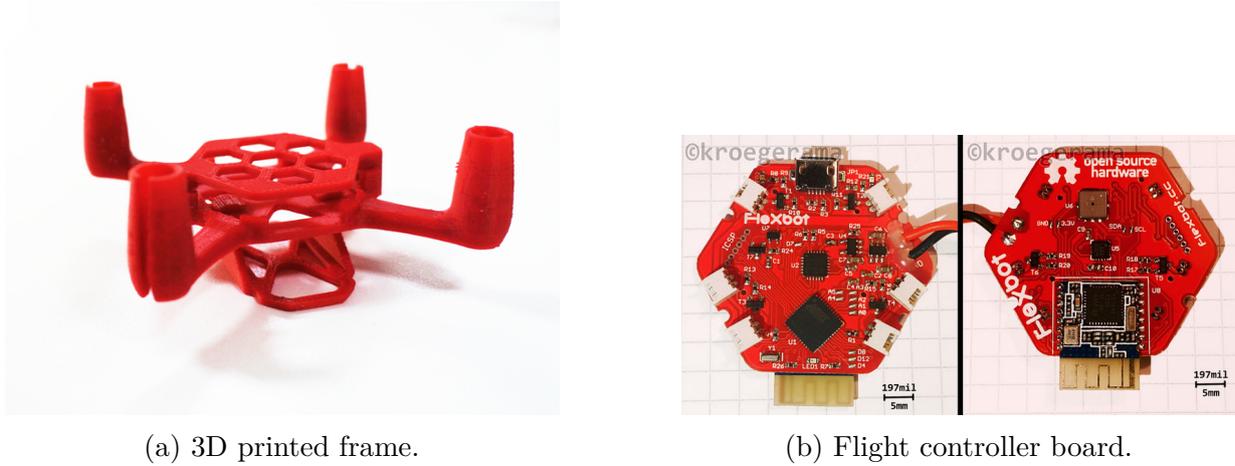


Figure 5.5: 3D printed frame of the flexbot quadrotor and the flight controller board.

I have contributed to debug and develop some parts of the toolbox RT-MaG, the RCB firmware, to robustify the communication link between the different boards, and I have performed several flying test indoor.

5.4.2 The Flexbot platform

Flexbot platform comes from a crowdfunding project which has developed open source hardware multicopters controllable with a smartphone.

The frame is 3D printed and the CAD model are open source. The frame of the quadcopter is shown on Figure 5.5a

The flight controller board features an ATmega32u4 processor, an IMU sensor (MPU6050), a magnetometer (HMCL5883L) and a barometer (BMP085). The motors are directly connected to the card as some transistors are mounted to run the DC motors. The board can be programmed with arduino, and the default firmware is based on MultiWii. A bluetooth low energy module is also mounted and communicate through a serial bus with the processor. Figure 5.5b shows the flight controller board.

An application for smartphone is provided to control the multicopter with bluetooth protocol. The data received by the flight controller board is organized with MSP⁷. As the flexbot is not compatible with DSM2 protocol, a specific emitter had to be developed for the MOCA room in order to communicate between the ground station and the platform. It consists in an arduino MEGA, an Ethernet shield and a bluetooth shield. The arduino board is connected to the same local network as the ground station which packs the command to fulfill the MSP and sends them through UDP to the IP address of the

⁷Multiwii Serial Protocol

arduino board at a specific port. The arduino board runs an UDP server which listens to the communication, pair the bluetooth device and relay the data received by UDP to the flexbot's bluetooth receiver.

This solution has been found to be reliable, but was only possible because the bluetooth module of the Flexbot and the arduino shield were both the same. However, Flexbot changed its bluetooth module after the first batch, the bluetooth shield was then not able to be paired with the flight controller board anymore. To overcome this issue, a Raspberry Pi and a bluetooth dongle have been used. The communication protocol had to be reverse engineered, an embedded program has been developed to run an UDP server, and to send the data through bluetooth protocol.

The main weaknesses of the Flexbot platform are:

- The lack of robustness of the 3D printed frame which makes it really easy to break;
- The lack of closed-loop control of the motors which means that the thrust is decreasing with the battery's voltage.

5.4.3 The AVE platform

The AVE (Aerial Vehicle efficient) quadcopter has been developed in GIPSA-Lab using commercial off-the-shelf components excepting the frame. As shown in Figure 5.6, the robot is equipped with a 3D printed structure, the span of the frame is 25cm. The frame has been designed by Josué Colmenares-Vazquez, PhD student at GIPSA-Lab. The total weight is 306g including the 2S battery. The flight autonomy is about 10 min in normal use.

A Crius AIOP FCB has been used for this robot. This board features a microcontroller (ATMega 2560), a Motion Processing Unit (MPU6050), a 3 axis magnetometer (HMC5883L), and an altimeter (MS5611). The board can be programmed with arduino, and the default firmware is based on MultiWii. A DSM2 radio receiver is connected to the PWM⁸ input of the board. Motors are controlled through Afro 12A Electronic Speed Controllers (ESC) which are connected to the PWM outputs of the FCB. As detailed latter in Section 5.4.6, the ESCs perform a closed-loop control of the motor's speed.

5.4.4 The Inductrix platform

The Inductrix UAV shown in Figure 5.7 is a commercial quadcopter for hobbyists developed by Blade[®]. It features a FCB, and 4 brushed motors. Communication is done via DSM

⁸Pulse Width Modulation

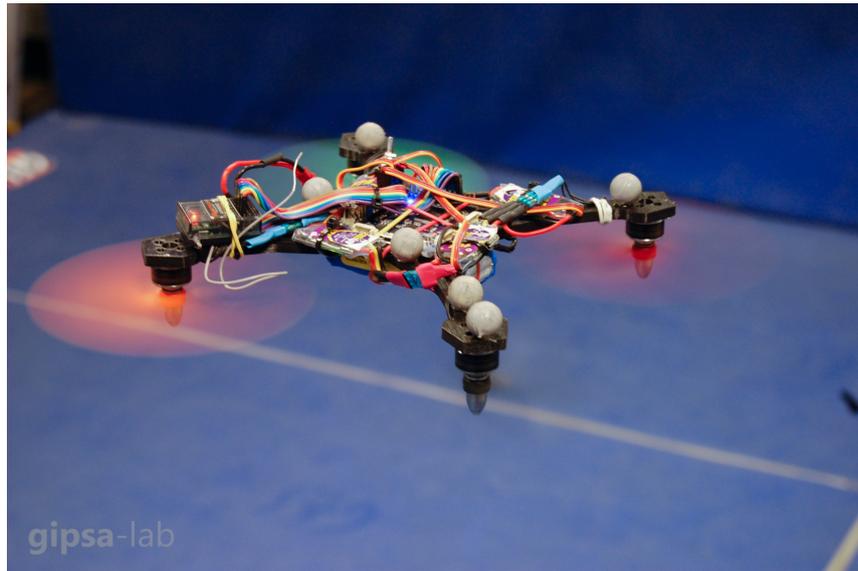


Figure 5.6: Experimental AVE platform.

protocol and the power is delivered by a 1S battery. The main advantage of this drone is its small size and the robustness of its frame. Due to the limited weight (about 27g with the battery), the impact in case of a crash does not damage the UAV heavily, and the propellers are also protected by the frame. The motors are directly connected to the FCB with connectors, therefore the replacement is easy in case of failure. However, the main disadvantages are the autonomy (around 5 minutes) and the lack of closed-loop control of the motors, which means that the speed of the motors drop with the battery state of charge. Moreover the FCB is proprietary which means that there is no possibility to upload custom code and it is not possible to know how the attitude control is performed onboard.



Figure 5.7: Inductrix quadcopter.



Figure 5.8: Emax frame of the TACO Platform.

5.4.5 The TACO platform

The TACO⁹ platform is built on a commercial platform sold by Emax[®] shown on Figure 5.8. This is a carbon and glass fiber frame made of different parts which are assembled with screws and locknuts. The total weight of this frame is about 185g, the length of the arm where the motors are mounted is 25cm. To connect the ESC to the battery, a power distribution board is used to ease the cable management. The FCB is a Naze32 which features a STM32F103CBT6 32-bit ARM Cortex M3 processor, a MPU6500 IMU accelerometer and gyroscope and a MHC5983 magnetometer. The standard version of Betaflight is used as the flight controller. LittleBee 20A ESCs are used in conjunction with XNOVA motors. The UAV is powered by a 3S battery and gives a flight autonomy around 10 minutes. The complete quadrotor mass is around 500g. While X4-MaG uses a dedicated processor to control the speed of the motors, this task is handled by the ESCs on TACO as detailed later in Section 5.4.6.

5.4.6 Motor control

A common problem experienced with quadcopters is the time-variant thrust response due to a drop of the battery voltage. In other words for a same command received by the flight controller, the resulting thrust given by the motors will depend on the battery state of charge. This time-variant response of the quadcopter requires a high integrator gain in the position controller which results in adding too much phase. For the Flexbot and the Inductrix, the DC motors are controlled directly by the FCB through some transistors. But for AVE and TACO drones, as more power is required for the motors, ESCs are used in order to provide the necessary amount of power and to handle the 3-phases of the motors. However, most of the ESCs do not achieve closed-loop control of the motor speed, and are then sensitive to the voltage drop of the battery. BLHeli is an open source project intended to replace the official firmware of different ESCs. The main advantage is that it provides

⁹Thrust Active Control

a sensorless closed-loop control mode of the motor speed. Therefore, the rotation speed should not be impacted by the battery's state of charge. Several adjustable parameters are available, however finding the good values for our setup is quite difficult without any objective measurement. For this reason, a test bench for the couple motor/ESC has been build in order to quantify the effect of the different tuning parameters.

5.4.6.1 Test bench setup

The only data needed is the speed of the motor in order to measure the response time and to check if the closed-loop control is able to reject a voltage drop. A Hall effect sensor is used to measure the speed. For the considered model of the motor, the poles are directly visible by the sensor. If it would not be the case, some magnets can be glued around the motor to trigger the sensor. An arduino Uno has been programmed to compute the frequency given by the hall sensor effect, this frequency is then sent over USB to a PC periodically. The whole setup is depicted on Figure 5.9.

The PC sends several setpoints through USB to another arduino card which generate the PPM¹⁰ signal for the ESC: 1ms pulses translate to zero throttle, 2ms pulses are full throttle. Figure 5.10 shows the input profile send to the ESC: the first part (from 0 to 10 seconds) consists in an ascending and descending ramp, the second part (from 10 to 20 seconds) tests several step responses, the third part (from 20 to 22 seconds) consists in a high frequency reference, and during the last part, the voltage of the power supply is dropped by 1 Volt. A Labview real-time software has been built to send, receive and save the different data.

5.4.6.2 Results

18 configurations of control parameters have been tested. Figure 5.11 shows the measured speed of the motor for 4 different tuning parameters. It can be seen that the response to the ramp input is close for all the closed-loop parameters. To determine the best values for the proportional gain (K_p) and the integral gain (K_i), two important behaviors will be considered: the rejection of a voltage drop and the step response. Figure 5.12 shows a detailed zoom around two regions of interest:

- It can be seen clearly on Figure 5.12a that a voltage drop of 1 Volt (near 24 s) results in a speed drop of about 15 Hz for the open-loop control, whereas all the closed-loop responses are less impacted and tend to recover the speed drop. Note that as the drop was manually applied to the power supply, it is not synchronized between all the experiments.

¹⁰Pulse Position Modulation

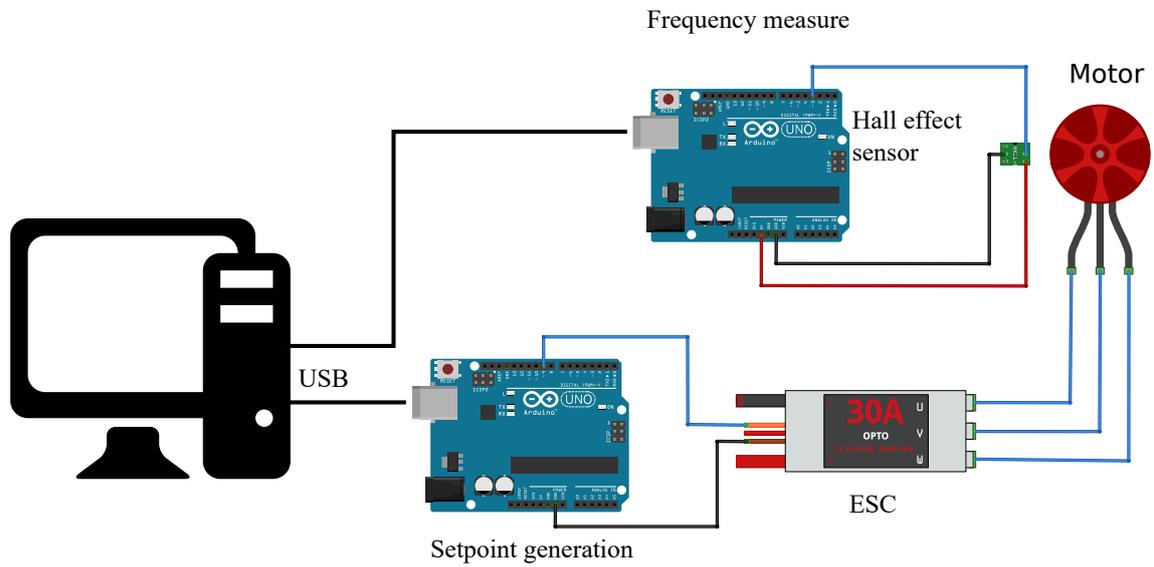


Figure 5.9: Setup of the test bench for the couple ESC/motor.

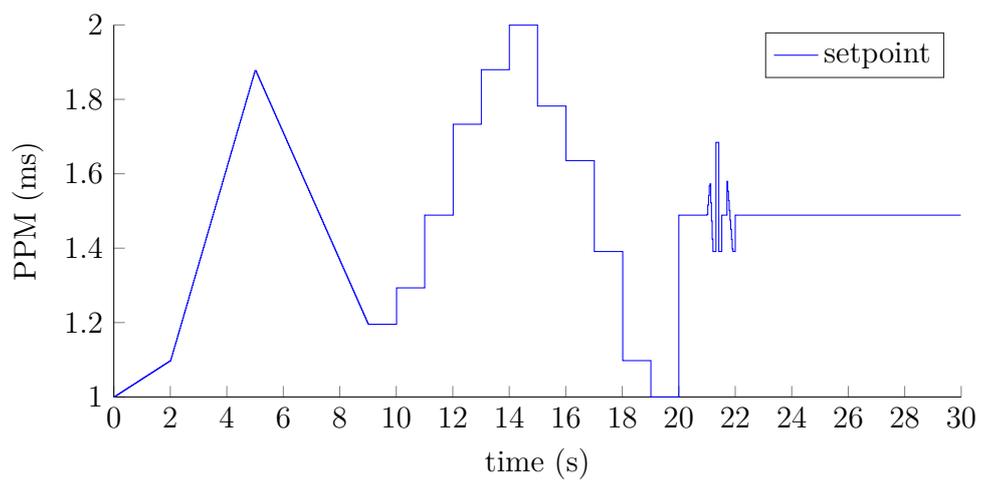


Figure 5.10: Profile input for the tuning of the motor control loop.

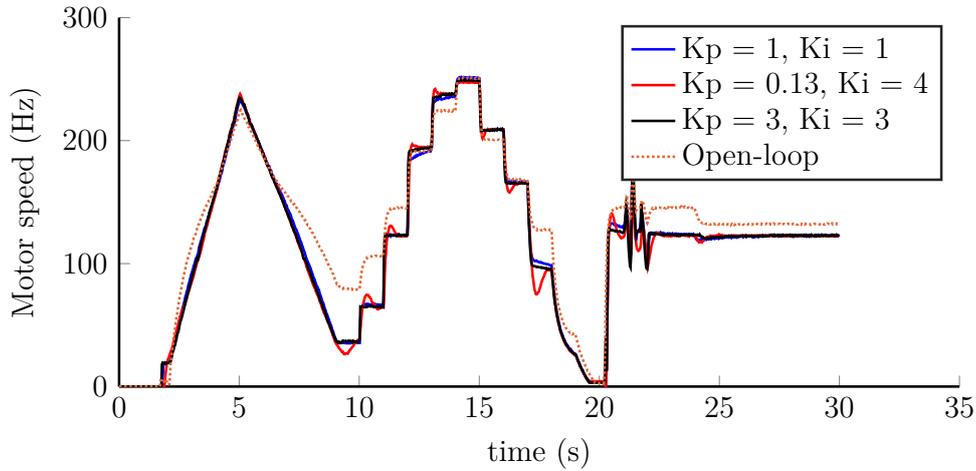


Figure 5.11: Measured speed for different tuning parameters.

- Figure 5.12b shows the response to a step input, at $t = 11$ s the PPM signal input of the ESC goes from 1.29 ms (29 % full speed) to 1.49 ms (49 % full speed). Note that the open-loop response has not been plotted because it does not converge to the same value.

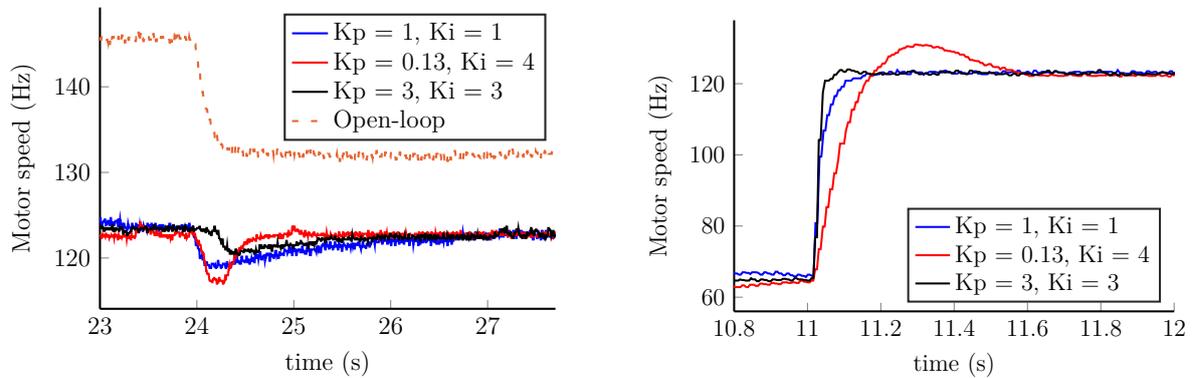
The parameters which have been found to give the best performances in terms of both disturbance rejection and step response are $K_p = 3$, and $K_i = 3$. A high value of K_i tends to provide faster disturbance rejection but it leads to an important overshoot of the step response if the proportional gain K_p is not high enough.

5.5 Platform choice

To conduct the experiments, the following requirements have to be met:

- Safety is an important requirement as the MOCA room is small and the experimental part will be demonstrated beyond people;
- The platform should be “crashproof”, this is an important requirement as crashes are more likely to occur during development.

The characteristics of each platform are reported in Table 5.1. It has been chosen to conduct the experiments with the Inductrix platform as this UAV has a small mass (27g), therefore the impact will carry less energy. Due to the well designed frame, this UAV is also resistant to crash. However as the motors are not speed controlled, the performances



(a) Disturbance rejection. A 1 Volt drop is applied to the power supply. Note that the drop is not applied exactly at the same time instant between the different experiments.

(b) Step response. At $t = 11$ s the PPM signal input of the ESC goes from 1.29 ms (29 % full speed) to 1.49 ms (49 % full speed).

Figure 5.12: Tuning of the ESC's gain.

	Inductrix	AVE	X4-MaG	Flexbot	TACO
Battery	1S 45C 150mAh	2S 2400mAh	2S 2400mAh	1S 25C 350mAh	3S
Autonomy	5 minutes	10 minutes	10 minutes	8 minutes	10 minutes
Motors	Brushed	Brushless T-motors	Brushless T-motors	Brushed	Brushless X-NOVA 2205
Closed-loop motors control	×	✓	✓	×	✓
Flight controller	proprietary	Multiwii	Custom	Multiwii	Betaflight
Size	cm	cm	cm	cm	cm
Mass (w/ battery)	27g	300g	350g	40g	500g
Safety	✓✓✓✓✓	✓✓	✓	✓✓✓✓✓	✓
Crash robustness	✓✓✓✓✓	✓✓	✓✓✓✓✓	✓✓	✓✓✓✓✓

Table 5.1: Comparison of the different platforms.

depend on the state of charge of the battery. We do not intend to change the FCB firmware, therefore the proprietary FCB is not an issue in the present case.

Application of the event-switched approach

Contents

6.1	Objectives	77
6.2	Preliminar steps	79
6.2.1	Position controller design	79
6.2.2	Position controller implementation	82
6.2.3	Identification	84
6.3	User assistance	86
6.4	Event-switching for constrained states	88
6.4.1	Problem statement	88
6.4.2	Definition of the constraint set	88
6.4.3	Event-set design	90
6.5	Experimental results	91
6.6	Improving identification	93
6.7	Summary	97

6.1 Objectives

In this chapter the event-switched method introduced in Chapter 4 will be used with the Inductrix platform presented in Section 5.4.4. While allowing a person to remotely control an UAV, we want several objectives to be met: the first one is to ease the maneuverability of the vehicle by providing an assistance to the pilot. Usually the attitude and the thrust of the quadcopter are controlled via the remote controller. However this method requires an experienced pilot as the first flights can be challenging for a beginner. Rather than the angles, controlling the speed of the robot seems more intuitive for a user. A speed

controller will be introduced later in Section 6.3 to achieve this goal. The second objective is to avoid a collision between the vehicle and the environment. The quadcopter will be modeled as a point in the space. The environment will be defined such that the UAV should remain in an origin centered square of $1.5\text{m} \times 1.5\text{m}$. This square will be called virtual limits or virtual walls. The height will not be constrained, only a 2-dimensional problem (in the horizontal plan) will be addressed here. Collision avoidance encompasses two problematics:

- How to detect an obstacle;
- How and when to take the UAV control over the pilot.

Obstacle detection can be done using onboard sensors. In [Achtelik et al., 2009] collision avoidance detection has been performed with a laser sensor and a stereo camera to achieve autonomous flight. Both measures of the sensors are merged to deliver an estimation of the position using an extended Kalman filter. A simultaneous localization and mapping (SLAM) algorithm is run on a ground station and a LQ controller brings the quadcopter to a safe trajectory if necessary. Generating this trajectory is a path planing problem. Artificial Potential Field (APF) method introduced in [Khatib, 1986] is a widely used method for path planning, it consists in assigning a potential function to the space, and simulating the robot reacting to the potential field. The robot is attracted to the target which has the lowest potential, while obstacles reject the vehicle. The space is discretized and an optimal problem is solved. A gradient based method has been proposed in [Khatib, 1986] to generate a trajectory. However this method can suffer from local minima, that is when the virtual forces sum up to zero while the target is not reached. Heuristic optimization method has been used in [Rathbun et al., 2002] to generate an optimal trajectory while avoiding local minima. A survey of different techniques for path planing can be found in [Goerzen et al., 2009].

Obstacle detection will not be discussed here, sensing will be done with the Vicon system. It is assumed that the positions of any element are known in real-time. Therefore sensing can be considered as perfect here because of the accuracy and precision of the measures. The origin will be considered as the target point when a collision is about to occur. Due to the convexity of the environment, the minimum distance trajectory to reach this target will be considered as safe, therefore a path planing method will not be necessary. Two different tasks have to be performed on the ground station as illustrated on Figure 6.1: 1/ the first one is to design a controller which determines *how* to act on the quadcopter to avoid a collision, 2/ and the second one is to design an event condition to check *when* it is necessary to act on the quadcopter in order to avoid a collision.

Event-based control gives a framework to design a stabilizing function and a trigger function. The event-switched approach presented in Chapter 4 will be used to design the trigger condition. When a collision is about to occur, the UAV will intend to go back to

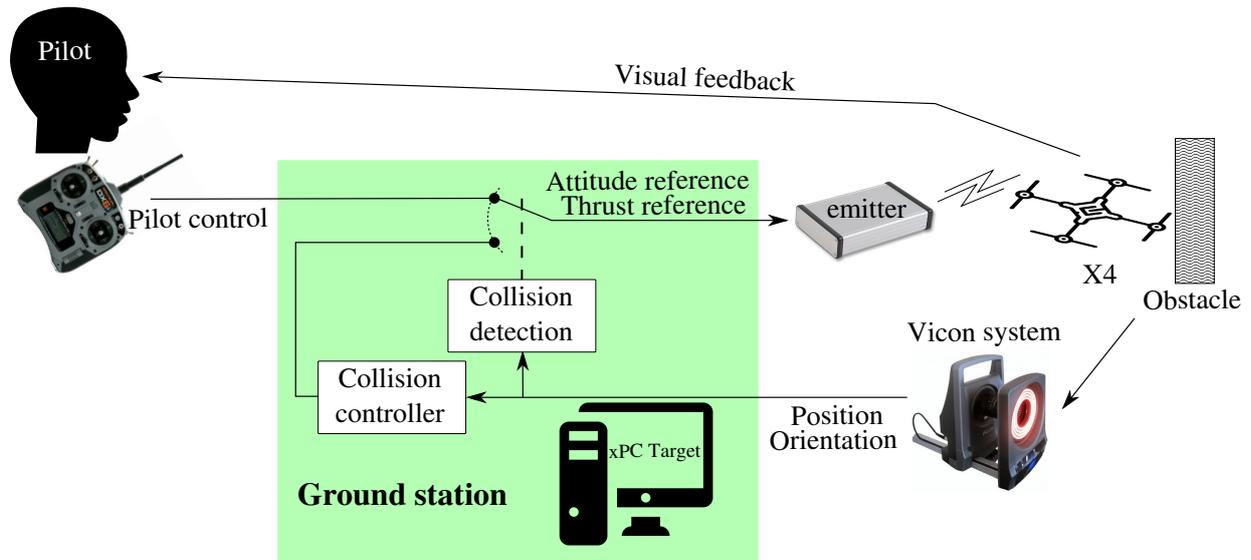


Figure 6.1: Schematic view of the collision avoidance approach used here.

the origin. To achieve this task, a PID controller introduced in the sequel will be used as a stabilizing control input. Event-switched approach aims at closing the control-loop only when a state constraint is about to be violated, while allowing open-loop otherwise. User inputs will be modeled as bounded unknown disturbances. The model of the robot will be considered as the open-loop dynamics. The event-switched approach gives all the initial conditions which guarantee that the constraints will not be violated considering the quadcopter dynamics and the worst possible user inputs.

6.2 Preliminary steps

6.2.1 Position controller design

The low level controller implemented on the FCB is able to stabilize the attitude of the vehicle. However, it can not perform position control because the IMU drift is too large to allow the acceleration double-integration to be successful. The position controller is implemented on the ground station and uses the position measure from the Vicon system.

Once an event is triggered, a control signal which brings the UAV to a safe position will be applied. The position controller is built upon a simplified model of the drone. It is assumed that the dynamics of the attitude controller is negligible beyond the dynamics of the position. In other words, the desired angles are considered to be reached quickly

enough. The obtained model of the system is then:

$$\sum \vec{F} = m\ddot{\xi} \quad (6.1)$$

$$R \cdot T - m \cdot \vec{g} = m\ddot{\xi} \quad (6.2)$$

with:

$$R = \begin{pmatrix} \cos \psi \cdot \cos \theta & \cos \psi \cdot \sin \theta \cdot \sin \phi - \sin \psi \cdot \cos \phi & \cos \psi \cdot \sin \theta \cdot \cos \phi + \sin \psi \cdot \sin \phi \\ \sin \psi \cdot \cos \theta & \sin \psi \cdot \sin \theta \cdot \sin \phi + \cos \psi \cdot \cos \phi & \sin \psi \cdot \sin \theta \cdot \cos \phi - \cos \psi \cdot \sin \phi \\ -\sin \theta & \cos \theta \cdot \sin \phi & \cos \theta \cdot \cos \phi \end{pmatrix} \quad (6.3)$$

Recall that T , ϕ , θ , and ψ are respectively, the total thrust, the roll, the pitch and the yaw angle.

Our aim is now to find the setpoints $(\phi^* \ \theta^* \ T^*)$ which are related to the desired accelerations along each axis: $(\ddot{x}^* \ \ddot{y}^* \ \ddot{z}^*)$.

$$\text{From (6.2)} \implies m \cdot \ddot{x} = T \cdot (\cos \psi \cdot \sin \theta \cdot \cos \phi + \sin \psi \cdot \sin \phi) \quad (6.4)$$

$$(6.2) \implies m \cdot \ddot{y} = T \cdot (\sin \psi \cdot \sin \theta \cdot \cos \phi - \cos \psi \cdot \sin \phi) \quad (6.5)$$

$$(6.2) \implies m \cdot \ddot{z} = T \cdot \cos \theta \cdot \cos \phi - m \cdot g \quad (6.6)$$

$$\text{From (6.6)} \implies T \cdot \cos \theta \cdot \cos \phi = m \cdot \ddot{z} + m \cdot g \quad (6.7)$$

$$(6.4) \cdot \sin \psi - (6.5) \cdot \cos \psi \iff T \cdot \sin \phi = m \cdot (\ddot{x} \cdot \sin \psi - \ddot{y} \cdot \cos \psi) \quad (6.8)$$

$$\frac{(6.8)}{(6.7)} \implies \phi = \tan^{-1} \left(\cos \theta \cdot \frac{\ddot{x} \cdot \sin \psi - \ddot{y} \cdot \cos \psi}{\ddot{z} + g} \right) \quad (6.9)$$

$$(6.4) \cdot \cos \psi + (6.5) \cdot \sin \psi \iff T \cdot \sin \theta = m \cdot \frac{\ddot{x} \cdot \cos \psi + \ddot{y} \cdot \sin \psi}{\cos \phi} \quad (6.10)$$

$$\frac{(6.10)}{(6.7)} \implies \theta = \tan^{-1} \left(\frac{\ddot{x} \cdot \cos \psi + \ddot{y} \cdot \sin \psi}{\ddot{z} + g} \right) \quad (6.11)$$

$$(6.6) \implies T = m \cdot \frac{(g + \ddot{z})}{\cos \theta \cdot \cos \phi} \quad (6.12)$$

Thus the desired accelerations $(\ddot{x}^* \ \ddot{y}^* \ \ddot{z}^*)^T$ can be considered as virtual inputs and are related to the following setpoints:

$$\phi^* = \tan^{-1} \left(\cos \theta \cdot \frac{\ddot{x}^* \cdot \sin \psi - \ddot{y}^* \cdot \cos \psi}{\ddot{z}^* + g} \right) \quad (6.13)$$

$$\theta^* = \tan^{-1} \left(\frac{\ddot{x}^* \cdot \cos \psi + \ddot{y}^* \cdot \sin \psi}{\ddot{z}^* + g} \right) \quad (6.14)$$

$$T^* = m \cdot \frac{(g + \ddot{z}^*)}{\cos \theta \cdot \cos \phi} \quad (6.15)$$

The dynamics of the position along each axis can be considered as a double integrator, where the inputs $(\ddot{x}^* \ \ddot{y}^* \ \ddot{z}^*)^T$ are transformed to angles values from the nonlinear relationships (6.13)-(6.15). By neglecting the attitude dynamics, the rotational speed of the yaw angle $\dot{\psi}$ is considered as a control input signal. Therefore the yaw's dynamics can be considered as an integrator.

For the implementation on the Inductrix platform introduced in Section 5.4.4, the angles and thrust setpoints need to be converted to 10 bits unsigned integers in order to be transmitted through the emitter. The maximum command for the angles ϕ^* and θ^* on the Inductrix UAV corresponds to $\frac{\pi}{6} rad = 30 deg$ and the minimum to $-\frac{\pi}{6} rad$. An angle α bounded between $\frac{\pi}{6}$ and $-\frac{\pi}{6}$ radians will be converted to a 10 bits unsigned integer a as follows:

$$a = 512 + 512\alpha \cdot \frac{6}{\pi} \quad (6.16)$$

To identify the scaling factor for the thrust, a simple procedure is used: the thrust command is increased until hover is reached, the thrust integer command needed to hover will be denoted b_h . Giving b_h and the mass of the Inductrix m , the thrust T^* in Newton can be converted to a 10 bits unsigned integer b as follows:

$$b = T^* \frac{b_h}{mg} \quad (6.17)$$

b_h has been found to be about 600 with a fully charged battery. We recall that since the speed of the motors are not subject to a closed-loop control on the Inductrix platform, this value will decrease with the state of charge of the battery. The maximum thrust the quadcopter can produce will be denoted \bar{T} and can be approximated as follows:

$$\bar{T} = 1023 \frac{mg}{b_h} = 0.452N \quad (6.18)$$

From the bounds of the angles ϕ , θ , and the thrust T , bounds of the control inputs $(\ddot{x}^* \ \ddot{y}^* \ \ddot{z}^*)^T$ can be deduced. The symbol $\bar{\ddot{x}}$ will denote the higher bounds of the acceleration along x axis, while $\underline{\ddot{x}}$ will denote the lower bound. Same kind of notations will be

	Proportional gain	Integral gain	Derivative gain
x and y position	4.35	0.43	1
z position	4.5	0.8	2.2
ψ angle	1.3	0.1	0

Table 6.1: Tuning of the different discrete controllers used for the position control. Sampling time $T_s = 10\text{ms}$.

used for the accelerations along y and z axis.

$$\bar{\ddot{x}} = \frac{\bar{T} \cdot \sin\left(\frac{\pi}{6}\right)}{m} = 8.37m \cdot s^{-2}, \quad \underline{\ddot{x}} = -\bar{\ddot{x}} = -8.37m \cdot s^{-2} \quad (6.19)$$

$$\bar{\ddot{y}} = \frac{\bar{T} \cdot \sin\left(\frac{\pi}{6}\right)}{m} = 8.37m \cdot s^{-2}, \quad \underline{\ddot{y}} = -\bar{\ddot{y}} = -8.37m \cdot s^{-2} \quad (6.20)$$

$$\bar{\ddot{z}} = \frac{\bar{T}}{m} - g = 6.92m \cdot s^{-2}, \quad \underline{\ddot{z}} = -g = -9.81m \cdot s^{-2} \quad (6.21)$$

6.2.2 Position controller implementation

Standard PID discrete controllers will be implemented to stabilize the position along each axis, and a PI controller will stabilize the yaw angle. The controllers will run on the ground station as depicted on Figure 6.2 at the frequency of 100Hz. Faster sampling could have been done up to 500Hz with the Vicon system, but was not necessary due to the limited dynamics of the model. Dynamics along x and y axis will be considered to be the same in the sequel, therefore the same controller structure and gains will be used. Considering a slow varying reference x_k^* , the virtual control input \ddot{x} will be computed as follows:

$$\ddot{x}_k^* = K_p(x_k^* - x_k) + K_d(-\dot{x}_k) + K_i \sum_{p=0}^k (x_p^* - x_p) T_s \quad (6.22)$$

where K_p represents the proportional gain, K_d the derivative gain and K_i the integral gain. The same structure will also be used for the controller along the z axis. Tuning of the gains was done by try and error until satisfactory performances were achieved. The different gains are summarized in Table 6.1.

Step response along x axis is plotted in Figure 6.3. Please note that cancellation of the steady state error was not targeted during the tuning, the controller needs only to bring the UAV in the origin neighborhood. We are more interested in the transient response rather than the steady state, because an event function will trigger this controller to bring back the UAV to the origin, and will give back the control to the pilot before reaching the target. A special procedure for take off and landing has been considered: it consists in generating a ramp reference instead of a step along z axis (this is equivalent to bound the

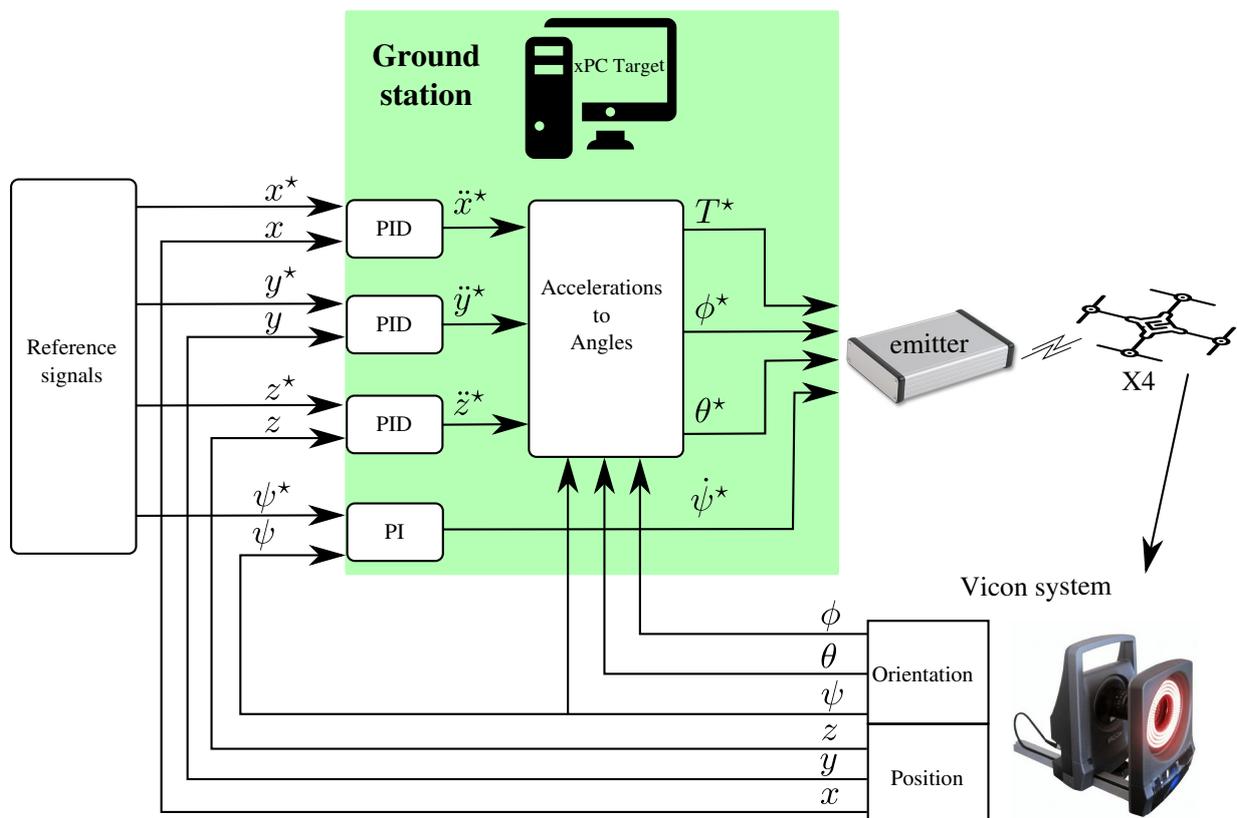


Figure 6.2: Diagram of the position control process

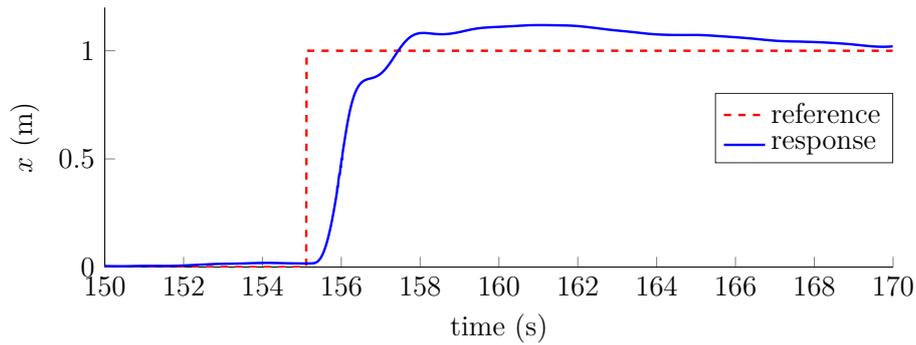


Figure 6.3: Experimental step response along x axis in closed-loop.

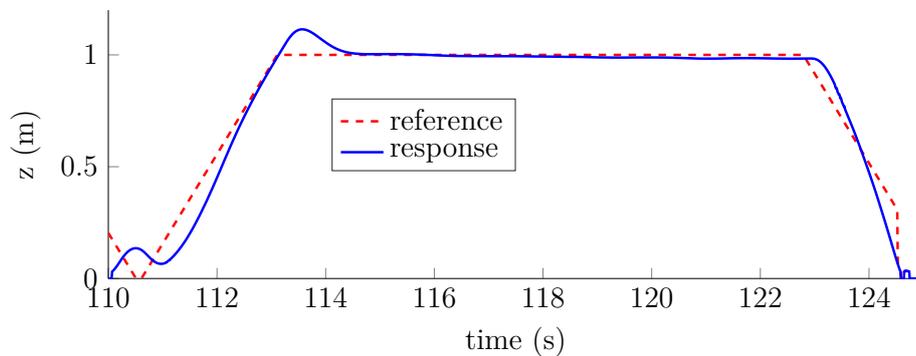


Figure 6.4: Experimental response to special procedures0 designed for take off and landing.

time derivative of the reference input z^*). Measured response is plotted in Figure 6.4. The integral gain has to compensate the decreasing battery state of charge, therefore it leads to an overshoot when applying a step input.

6.2.3 Identification

The controller along z axis is needed to keep the quadcopter at a constant height, while the controllers along x and y are needed to bring the quadcopter near the origin to avoid a collision. In order to apply the event-switched method, a discrete LTI model of the closed-loop system is needed. The obtained model will reflect the dynamics of the physical platform, in addition to the attitude and position controller. Identification will be performed along x axis. The dynamics along y axis will be considered similar to the x axis. The altitude will be regulated around a constant height ($z^* = 0.8m$), the quadcopter will be placed at $x = 1m$ and $y = 0m$ before closing the control loop.

A model can be built from the response of the system when the PID controller (6.22) is applied with a setpoint $x^* = 0$. The experimental response is shown in Figure 6.5. The

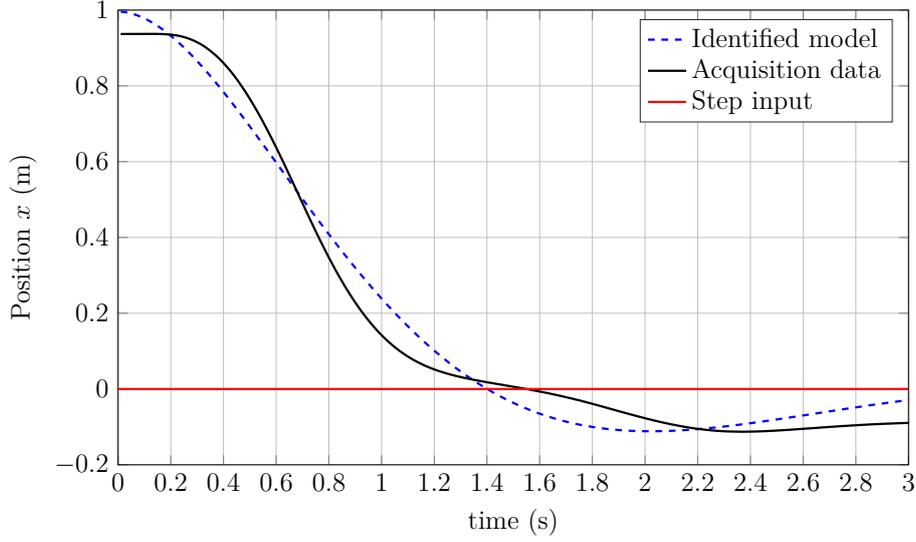


Figure 6.5: Identification of the closed-loop system. Measured data are plotted in black and the corresponding model response in dashed blue.

following continuous autonomous second order system has been obtained:

$$\dot{X}(t) = \bar{A}_c \cdot X(t) \quad (6.23)$$

where $X = (x \ \dot{x})^T$, and with $\bar{A}_c = \begin{pmatrix} 0 & 1 \\ -3.374 & -2.091 \end{pmatrix}$

Considering a sampling time of $T_s = 0.01s$, the discrete time closed-loop dynamics is described by:

$$X_{k+1} = \bar{A} \cdot X_k \quad (6.24)$$

with $\bar{A} = \begin{pmatrix} 0.9998 & 0.009896 \\ -0.03339 & 0.9791 \end{pmatrix}$. This model is stable and represents the dynamics of the UAV when a stabilizing control to the origin is applied.

A double integrator will be considered as an open-loop model. Considering a sampling time of $T_s = 0.01s$, the model is described as follows:

$$X_{k+1} = A \cdot X_k + B \cdot u_k \quad (6.25)$$

where u_k denotes the unknown but bounded user input (considered as a perturbation here), and with $A = \begin{pmatrix} 1 & 0.01 \\ 0 & 1 \end{pmatrix}$ and $B = \begin{pmatrix} 0 \\ 0.01 \end{pmatrix}$

6.3 User assistance

Usually the user directly controls the angles ϕ and θ with the remote control. However, it can be challenging for a beginner to achieve a stable flight. To assist the user to pilot the quadcopter, speed controllers will be used on the horizontal plane (x, y) . Thus the user will control the speed with the remote controller rather than the pitch and the roll angles. The right stick of the remote control will be mapped to speed setpoint v_x^* and v_y^* along the x and the y axis of the body frame. As dynamics for speed can be considered as a simple integrator, a PI controller will be used. Implementing the controllers in the body frame is important because the integral part should compensate a steady-state error due to a mechanical disturbance. To transform the speed from the inertial frame to the body frame, only the horizontal plane (x, y) will be considered, thus only the rotation matrix $R_z(\psi) = \begin{pmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{pmatrix}$ is needed. Speed of the UAV in the inertial frame, denoted by $v_x = \dot{x}$ and $v_y = \dot{y}$, will be estimated by applying a first order filter on the position given by the Vicon system:

$$\hat{v}_x(t) = \frac{s}{1 + 0.06s} x(t) \quad (6.26)$$

where s denotes the Laplace variable and \hat{v}_x the estimation of the speed along x axis of the inertial frame. Direct approximation of the derivative from measurements (using backward Euler approximation) can also be computed, however it is very sensible to noise. Filter (6.26) has been tuned to attenuate the noise of the obtained time-derivative with an admissible delay. Estimated speed in the body frame along x and y axis will be denoted \hat{v}_x^B and \hat{v}_y^B respectively and can be computed as follows:

$$\begin{pmatrix} \hat{v}_x^B(t) \\ \hat{v}_y^B(t) \end{pmatrix} = R_z(\psi)^T \begin{pmatrix} \hat{v}_x(t) \\ \hat{v}_y(t) \end{pmatrix} \quad (6.27)$$

The PI controllers will provide acceleration setpoints in the body frame which need to be transformed in the inertial frame. Schematic view of the control implementation is depicted in Figure 6.6.

The left stick will be mapped to the setpoint ψ^* , thus the user can directly control the yaw of the UAV.

PI controllers have been tuned manually until satisfactory performances were achieved. Important focus has been given on steady state error as it is primordial to stop any motion when the user is putting the stick to the center. Response to a step for the speed around x axis is shown in Figure 6.7. The yaw angle ψ was regulated around 0, therefore speed was the same between inertial and body frames.

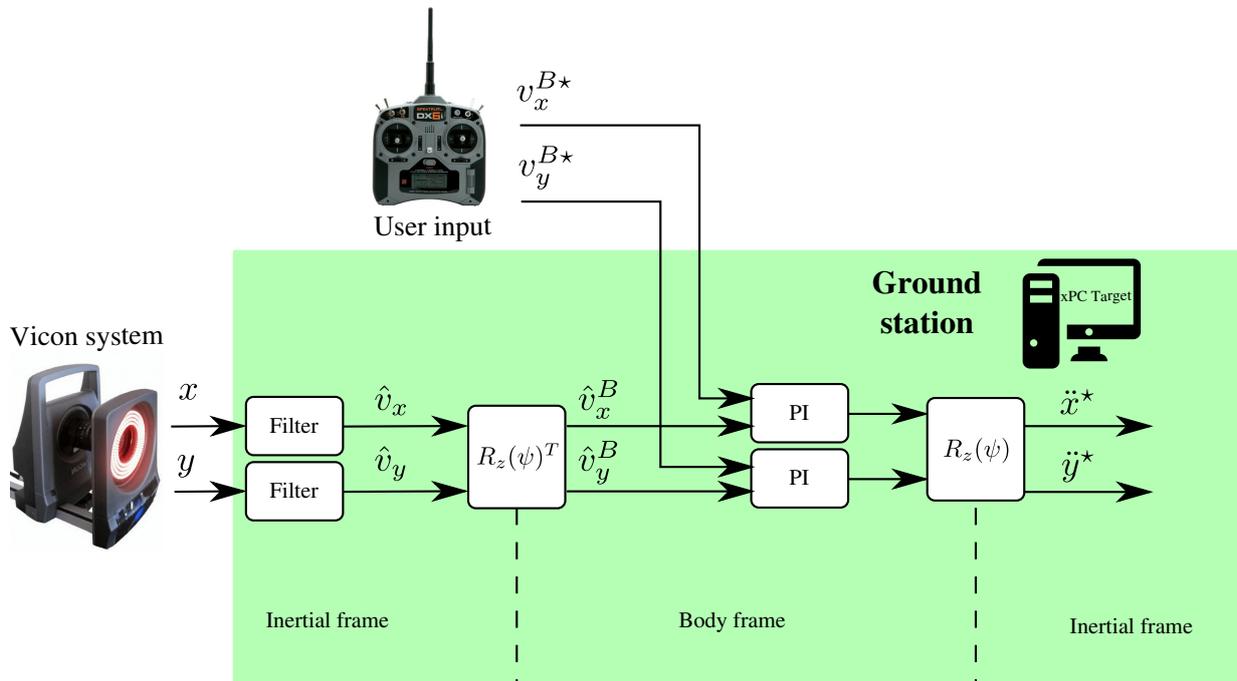


Figure 6.6: Diagram of the speed control process.

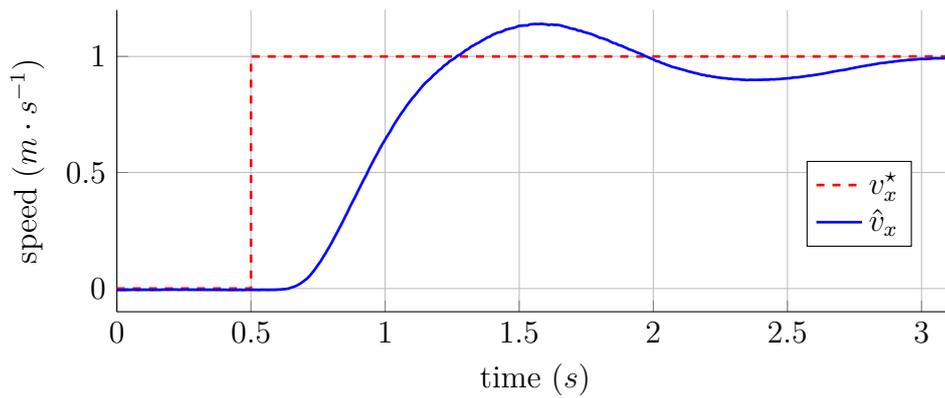


Figure 6.7: Experimental step response of the speed along x axis in closed loop. The reference is plotted in dashed red line, while the measured response is in blue.

6.4 Event-switching for constrained states

6.4.1 Problem statement

The objective is to avoid a collision between the vehicle and the environment while a pilot is controlling the robot. Event-switched method introduced in Chapter 4 will be used to switch between the user inputs and a collision avoidance program, which consists in a PID controller which brings the robot near the origin. Controller introduced in Section 6.2.2 will be used with the setpoint x^* set to 0. The quadcopter will be modeled as a point in the space. Only a 2-dimensional problem will be addressed here. The environment will be defined such that the UAV should remain in an origin centered square of $1.5\text{m} \times 1.5\text{m}$. This square will be called virtual limits or virtual walls. Due to the symmetry of the problem, and because the dynamics along x and y axes will be considered similar, only the constraints and the dynamics along x axis will be considered in the sequel.

Given a set $\mathcal{X} \in \mathbb{R}^2$, the discrete LTI autonomous model (6.24) and the model (6.25) depending on the unknown user input u_k , the aim is to design an event-set $\mathcal{S} \in \mathbb{R}^2$ such that the state trajectory of the switched system:

$$X_{k+1} = \begin{cases} AX_k + Bu_k & \text{if } X_k \in \mathcal{S} \\ \bar{A}X_k & \text{otherwise} \end{cases} \quad (6.28)$$

is constrained in the set \mathcal{X} . This set will define all the specified constraints. In the following, the design of the constraint set \mathcal{X} and the event-set \mathcal{S} will be detailed. Thereafter, they will be implemented in the event-switched control strategy.

6.4.2 Definition of the constraint set

Virtual limits define the set which represents the bounds on the position:

$$\mathcal{V} = \{X \in \mathbb{R}^2 : H_v X \leq P_v\} \quad (6.29)$$

with H_v and P_v describing the virtual limits

$$H_v = \begin{pmatrix} 1 & 0 \\ -1 & 0 \end{pmatrix} \text{ and } P_v = \begin{pmatrix} 0.75 \\ 0.75 \end{pmatrix} \quad (6.30)$$

If the trajectory of the vehicle is constrained in \mathcal{V} , the position of the robot will be bounded between $[-0.75; +0.75]\text{cm}$. This could be enough to describe the constraint set. However, we propose to also consider the validity domain $\bar{\mathcal{X}}$ of model (6.24). This autonomous model represents the dynamics of the quadcopter when the PID control input introduced in Section 6.2.2 is applied to stabilize the vehicle near the origin. Because of the actuator

saturation, this model is valid only inside a specific region $\bar{\mathcal{X}} \in \mathbb{R}^2$ of the state-space. If integral gain is neglected in the PID controller (6.22), the stabilizing input can be expressed in a state-feedback form: $\ddot{x}_k^* = -K \cdot X_k$, with $K = \begin{pmatrix} 4.35 & 1 \end{pmatrix}$. Then the input constraint set $\bar{\mathcal{U}} = \{\ddot{x}_k^* \in \mathbb{R} : \ddot{x} \leq \ddot{x}^* \leq \bar{\ddot{x}}\}$ can be projected in the state space, the resulting set is expressed by: $\bar{\mathcal{X}} = \{X \in \mathbb{R}^2 : \ddot{x} \leq -KX \leq \bar{\ddot{x}}\}$.

The constraint set \mathcal{X} is defined such that both the validity domain and the virtual limits are considered:

$$\mathcal{X} = \bar{\mathcal{X}} \cap \mathcal{V} \quad (6.31)$$

Given \mathcal{P} a positive invariant set for system (6.24), event-switched approach provides a systematic way to design an event-set \mathcal{S} such that the states of switched system (6.28) remains in \mathcal{P} .

The aim is to constrain the states of system (6.28) in \mathcal{X} , however this set is not necessarily positive invariant for this system. While it has been chosen to consider the mRPI set in Chapter 4, we will not consider disturbance for the stable dynamic, and we propose here to consider the MPI¹ set in \mathcal{X} instead. Methods to compute MPI set for discrete time linear systems are well known and the reader can refer to [Kerrigan, 2000] for further details. Figure 6.8 shows the MPI set \mathcal{P} of system (6.24) included in the constraint set \mathcal{X} .

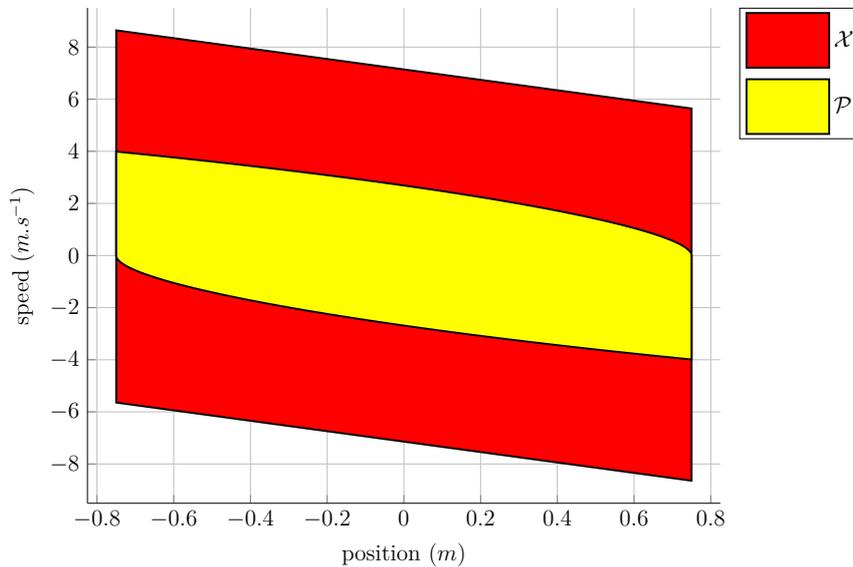


Figure 6.8: Constraint set \mathcal{X} is represented in red, and the MPI set \mathcal{P} in yellow.

¹Maximum Positively Invariant

6.4.3 Event-set design

The second step is to design the event-set \mathcal{S} dependent on the set \mathcal{P} and the open-loop dynamics (6.25). The control input u given by the pilot is considered as a bounded stochastic disturbance as the user input is not known in advance. In Chapter 4 it is proposed to design an event-set \mathcal{S} such that:

$$\begin{aligned} X_k \in \mathcal{S} &\implies X_{k+1} \in \mathcal{P} \\ \text{with } X_{k+1} &= A \cdot X_k + B \cdot u_k \end{aligned} \quad (6.32)$$

For robustness purpose we rather choose to design \mathcal{S} such that:

$$\begin{aligned} X_k \in \mathcal{S} &\implies X_{k+h} \in \mathcal{P}, \forall h \in \mathbb{N} \text{ and } 1 \leq h \leq p \\ \text{with } X_{k+h} &= A^h \cdot X_k + \sum_{k=0}^{h-1} A^{h-1-m} \cdot B \cdot u_k \text{ and } h \geq 1, h \in \mathbb{N} \end{aligned} \quad (6.33)$$

Here, $p = 10$ has been chosen. It means that the state will stay in the set \mathcal{P} even if the control law is switched 10 sampling time later.

After computation, the event-set \mathcal{S} is represented by 136 inequalities which define all the vertex of \mathcal{S} . As discussed in Chapter 4 it can be time consuming to check in real-time if the state belongs to \mathcal{S} . Consequently, to reduce the complexity, the polyhedral set is approximated by the biggest ellipsoidal set $\mathcal{E} : \{x \in \mathbb{R}^2 : x^T Q x \leq 1\}$ included in \mathcal{S} . The value of Q has been found by solving a convex optimization problem (see Section 4.5.3 for details):

$$Q = \begin{pmatrix} 5.8938 & 2.3197 \\ 2.3197 & 1.1998 \end{pmatrix}$$

Control input \ddot{x}^* will be switched between the position controller $\ddot{x}_k^* = -K X_k$ and the user input $\ddot{x}_k^* = u_k$. When the position controller is used, the dynamics can be modeled as an autonomous system. The resulting switched system is expressed as follows:

$$X_{k+1} = \begin{cases} AX_k + Bu_k & \text{if } X_k \in \mathcal{E} \\ \bar{A}X_k & \text{otherwise} \end{cases} \quad (6.34)$$

The control inputs \ddot{x}^* and \ddot{y}^* will both be switched to the position controller every time an event is triggered (when the state leave the set \mathcal{E}) along any axis x or y . The MPI set \mathcal{P} , the polyhedral event-set \mathcal{S} , and the ellipsoidal event-set \mathcal{E} are depicted in Figure 6.9.

Computation time

The inclusion condition $X_k \in \mathcal{S}$ will be checked on the ground station, therefore we should not be limited by the computational power. However, for further development it is inter-

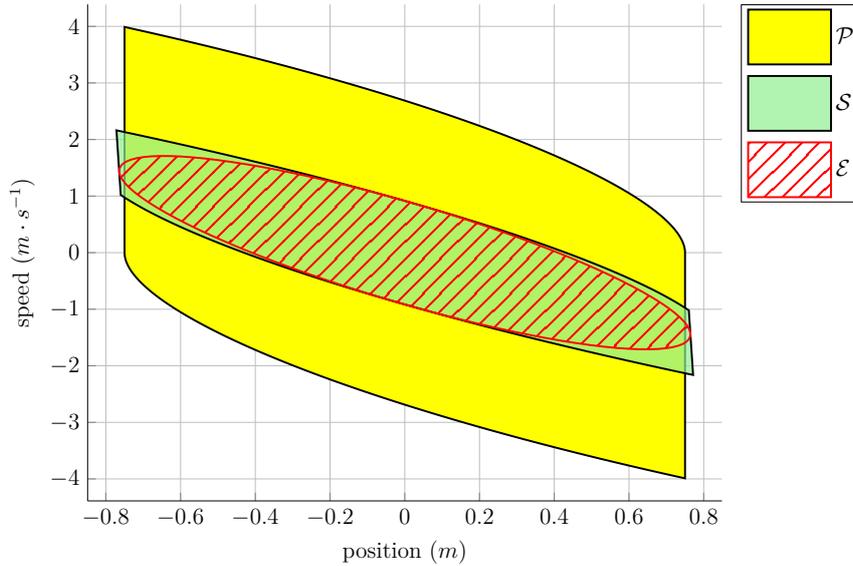


Figure 6.9: The MPI set \mathcal{P} is represented in yellow, the event-set \mathcal{S} in green, and its ellipsoidal approximation \mathcal{E} in red stripes.

esting to check if the inclusion condition could be performed onboard and how long does it takes.

On the Naze32 FCB, the CPU is running at 72MHz. 7 cycles are needed to perform an integer 32bits multiplication (4 for loading operands, 2 to store result, 1 for the multiplication itself), and 47 cycles are needed for a floating point multiplication. Using the condition based on the event set \mathcal{S} leads to 272 multiplications, 136 additions, 136 comparisons. 3808 clock cycles = $52.9\mu s$ are needed to check the state inclusion in \mathcal{S} with integer operations and 14688 clock cycles = $204\mu s$ with floating point unit operations.

Using the condition based on the event set \mathcal{E} leads to 6 multiplications, 3 additions, 1 comparison. 70 clock cycles = $972ns$ are needed to check the state inclusion in \mathcal{E} with integer operations and 310 clock cycles = $4.31\mu s$ with floating point unit operations. Using the event-set \mathcal{E} rather than \mathcal{S} reduces by 98% the execution time of the event condition. It can be noticed that the floating point operations are more time consuming due to the fact that the Naze32 does not include a floating point unit.

6.5 Experimental results

Implementation has been done on the ground station, the same event-set was used for the x and the y axis. The quadcopter was stabilized at a constant height ($z^* = 0.8m$). While a human pilot was moving the sticks of the remote control randomly, event-switched

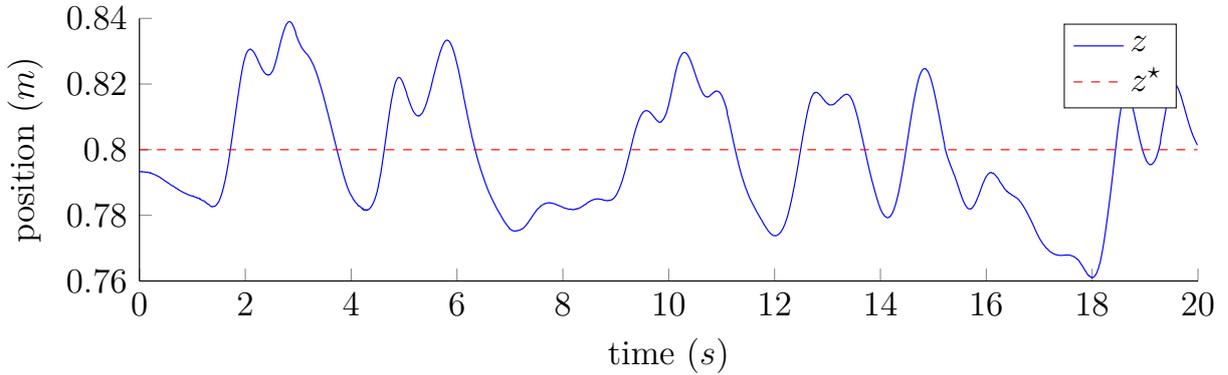


Figure 6.10: Altitude of the UAV.

control should prevent a collision between the UAV and the virtual limits. An experiment of 20 seconds has been conducted. The altitude of the UAV is shown in Figure 6.10, the height was regulated around 0.8m with less than 6% error. Trajectory of the UAV in the horizontal plane is shown in Figure 6.11, one can observe that the quadcopter does not collide with the virtual limits. Position along x and y axis are plotted in Figure 6.12a, the measured speed is shown in Figure 6.12b. The recorded user input expressed in the inertial frame is shown in Figure 6.12c, the estimated speed in 6.12b. Figure 6.12d shows when the control input is switched between the position controller and the user input, when the event is '1' the pilot can not control the UAV. Trajectory of the states X and Y in the state space are depicted in Figure 6.13a and 6.13b respectively, an event is triggered when the state does not belong to \mathcal{E} . A stabilizing control is then applied, which brings back the states in \mathcal{E} . It can be noticed that the state does not leave \mathcal{P} as intended.

While this experiment shows that our approach works as intended, there are some known issues, sometimes the UAV can collide with the virtual limits. A recording of the trajectory in the horizontal plane during such an “accident” is shown in Figure 6.14. It can be seen that the state X leaves the set \mathcal{P} , despite the fact that the stabilizing control $-Kx_k$ has been applied to the system.

This behavior can be explained by several hypotheses:

- Communication fault between the ground station and the FCB has occurred, therefore the correct stabilizing input could not be transmitted to the robot;
- The closed-loop model is not good enough, therefore the set \mathcal{P} is not positive invariant for the real system;
- The open-loop model is not good enough, therefore the event condition does not guarantee the positive invariance of the set \mathcal{P} for the switched system.

The first one can be solved by including more robustness delay in (6.33). For the second

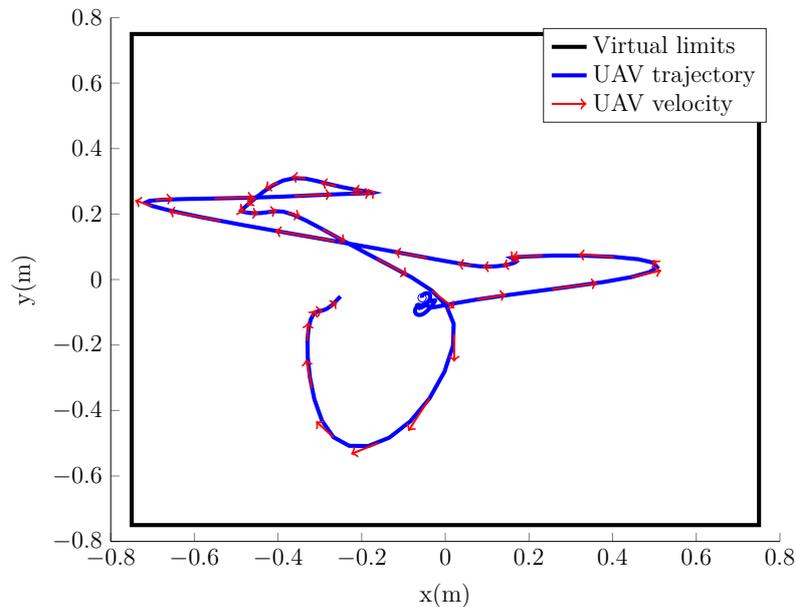


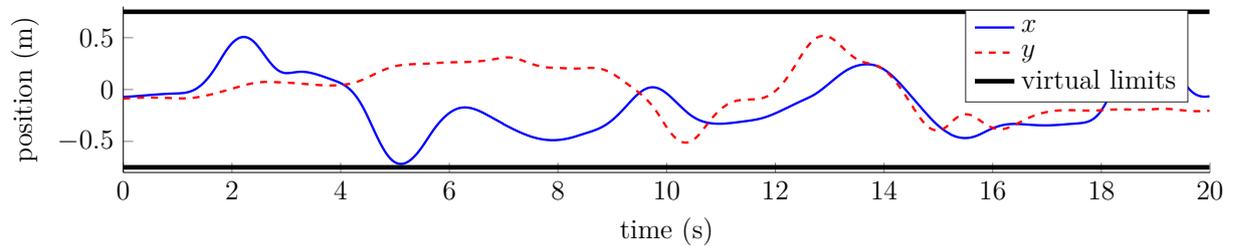
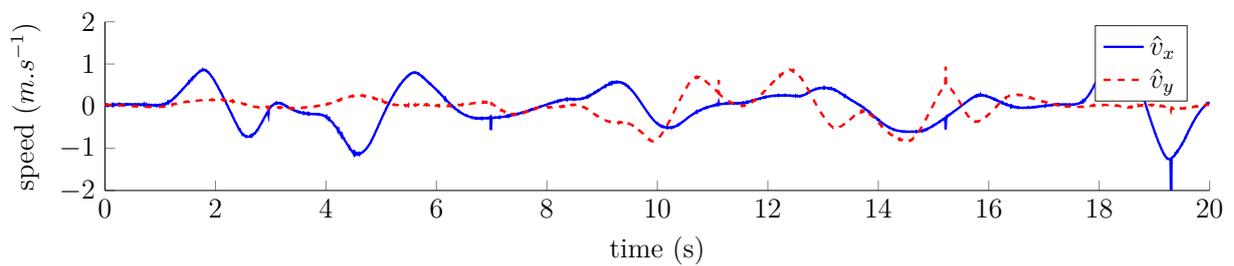
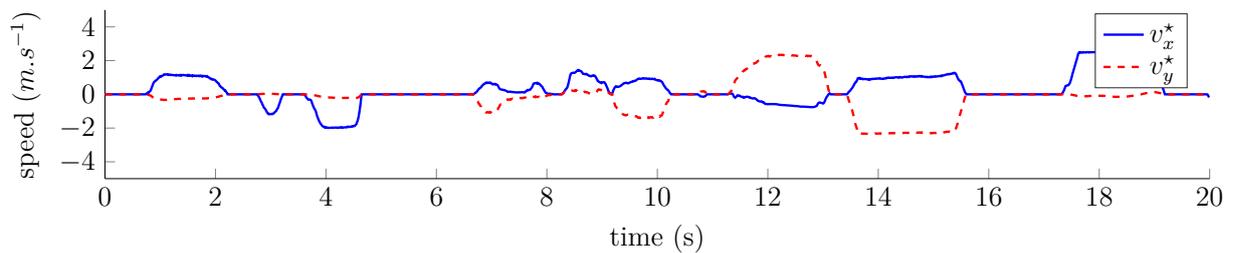
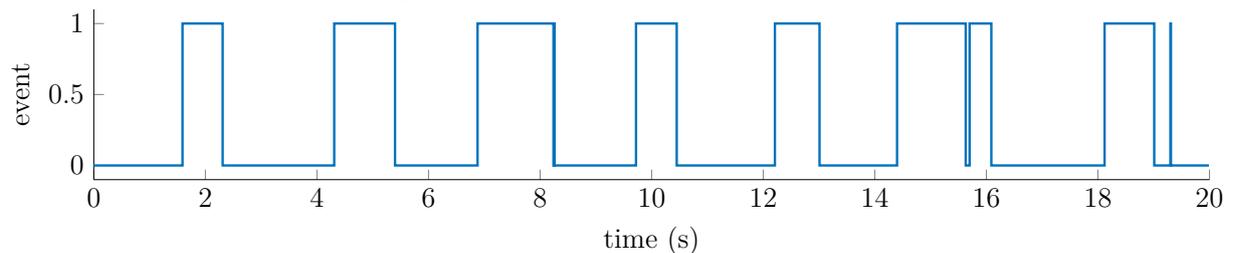
Figure 6.11: Trajectory of the UAV in the (x, y) plane.

one, some robustness can be obtained by the addition of a bounded disturbance in the closed-loop model. For the third one increasing the bounds on the user input to compute the positive invariant set will also increase the robustness. Of course, the best way to solve the second and the third issues is to build a better closed-loop and open-loop model. Position dynamics along x and the y axis of the inertial frame were modeled using the same equations, it would be better to consider dynamics in the body frame using separate models for both axes. Finally, a big issue which has been experienced is the performances dependence of the battery state of charge. This issue renders the real system time variant. Integrator gains had to be kept high to overcome this problem, however it was at the expense of the phase margin.

6.6 Improving identification

We propose to identify an open-loop model of the UAV in the body frame. This model will reflect the dynamics of the mechanic system and the attitude controller implemented on the FCB. While hovering at a constant height, a PRBS² is applied to the inputs \dot{x}^* and \dot{y}^* . The position of the UAV is recorded to construct a nonparametric model. A PRBS provides a richer frequency content than a step input, therefore the identification is better to model the high frequency dynamics. To limit the duration of the experiment, a sampling frequency of 20 Hz has been chosen and the parameter p of the PRBS which describes the

²Pseudo Random Binary Sequence

(a) Position along x in blue and y in dashed red line.(b) Speed along x in blue and y in dashed red line.(c) User input along x in blue and y in dashed red line.(d) Event along x in blue and y in dashed red. The event switches to 1 when the state does not belong to the event-setFigure 6.12: Recorded data of a flight using event-switched approach to keep the quadcopter in a $1.5m$ square.

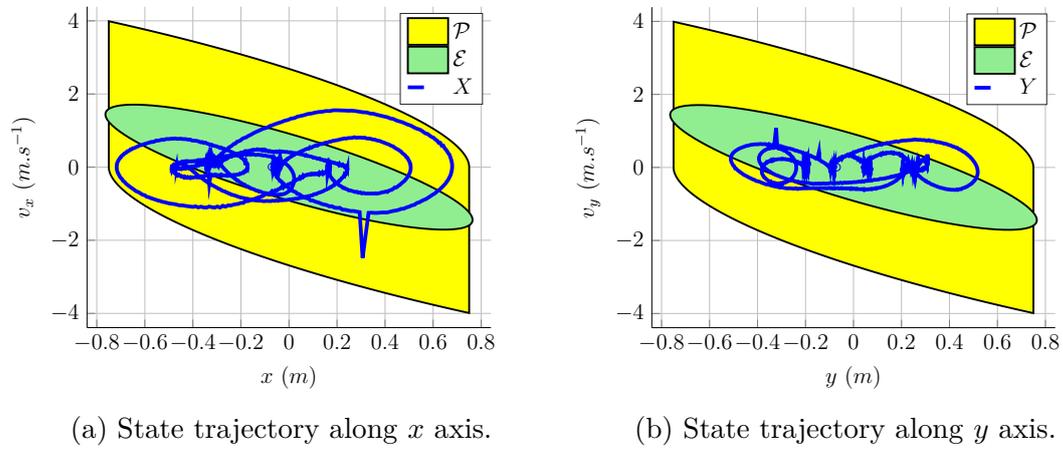


Figure 6.13: An event is triggered once any of the state X or Y leave the event-set \mathcal{E} (in green).

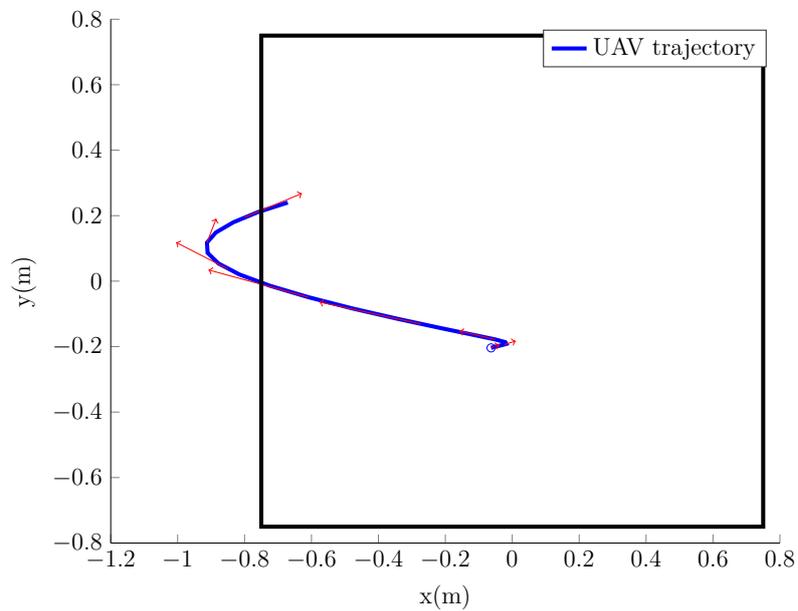
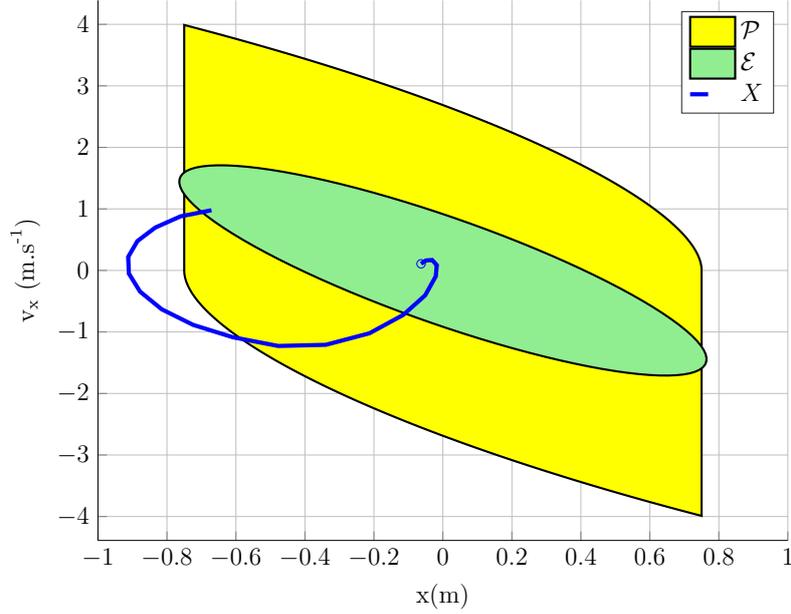


Figure 6.14: Trajectory of the UAV in the (x, y) plane. A collision has occurred between the quadcopter and the virtual limit.

Figure 6.15: Trajectory of the state along x axis.

flatness of the frequency spectrum has been set to 2. It means that the identification signal will have less power in higher frequencies. The number of registers of the PRBS will be set to 8, therefore the experiments should last at least $(2^8 - 1)/40 = 6.38s$. The amplitude of the PRBS has been set to $1.2m \cdot s^{-2}$.

We aim to model a second order system around each axis, the input of the MIMO model will be the accelerations setpoint \ddot{x}_B^* and \ddot{y}_B^* expressed in the body frame, and the output will be the accelerations along x and y axis in the body frame. Vicon system will record the position of the UAV, and an estimation of the acceleration in the body frame will be computed offline. This acceleration will be respectively denoted $\hat{\ddot{x}}_B$ and $\hat{\ddot{y}}_B$ along x and y axis of the body frame. The PRBS inputs for both input signals are shown in Figure 6.16. A 4th order ARMAX model with 2 inputs has been used to fit the recorded data, the simulation of the identified model and the measured output is plotted in Figure 6.17. This model can be converted into the following discrete time state-space model:

$$M_{k+1} = A_m M_k + B_m c_k \quad (6.35)$$

$$\begin{pmatrix} \ddot{x}_k^B \\ \ddot{y}_k^B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} M_k \quad (6.36)$$

$$\text{with } A_m = \begin{pmatrix} 1.546 & 0.001395 & 1 & 0 \\ 0.005295 & 1.612 & 0 & 1 \\ -0.6784 & 3.485 \cdot 10^{-5} & 0 & 0 \\ 0.00175 & -0.7307 & 0 & 0 \end{pmatrix} \text{ and } B_m = \begin{pmatrix} 0.02257 & -0.001995 \\ 0.001802 & 0.02586 \\ 0.04878 & -0.004107 \\ 0.004597 & 0.04363 \end{pmatrix}$$

where M_k represents the state of the system, and $c_k = (\ddot{x}_B^* \quad \ddot{y}_B^*)^T = R_z^T(\psi) (\ddot{x}_k^* \quad \ddot{y}_k^*)^T$

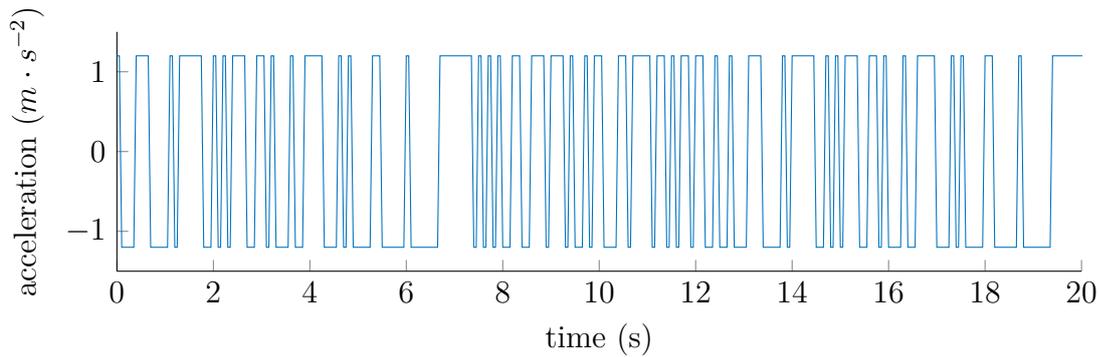
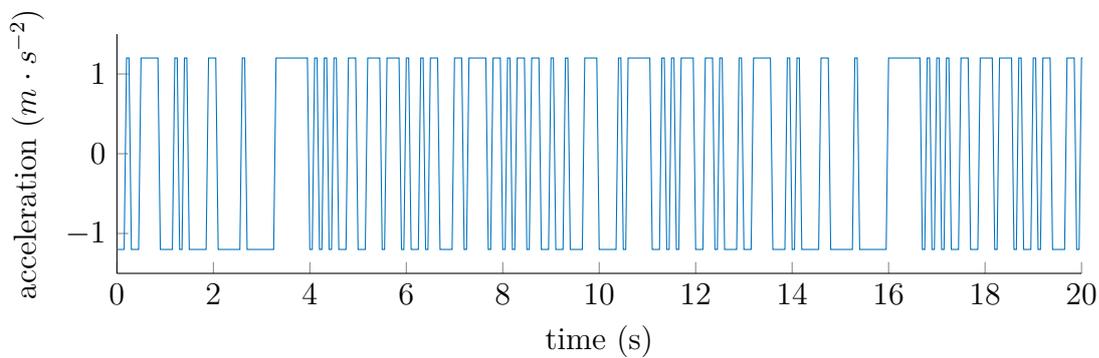
(a) \ddot{x}^* input.(b) \ddot{y}^* input.

Figure 6.16: PRBS inputs for identification.

the control input. This system can be augmented to model the position dynamics, the resulting model would be an 8th order model.

For further development, we propose to use this model to design a better controller and to design the event set.

6.7 Summary

In this chapter, event-switched approach has been used to design an event condition which switches between the user input (considered as unknown bounded disturbances) and a controller to constrain the position of the UAV inside virtual limits. Firstly, a stabilizing controller which brings back the robot near the origin of the inertial frame $\{\mathbb{I}\}$ has been designed, secondly a closed-loop model has been identified, and finally the event condition has been computed. This approach has been tested on a real-time platform with a user moving the sticks of the remote control randomly. While most of the time the user input is switched off and the UAV is brought near the origin of the inertial frame to avoid a collision,

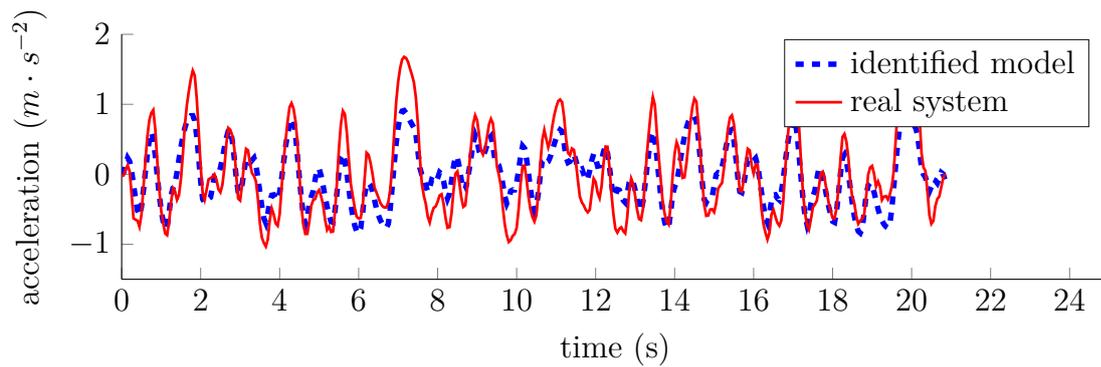
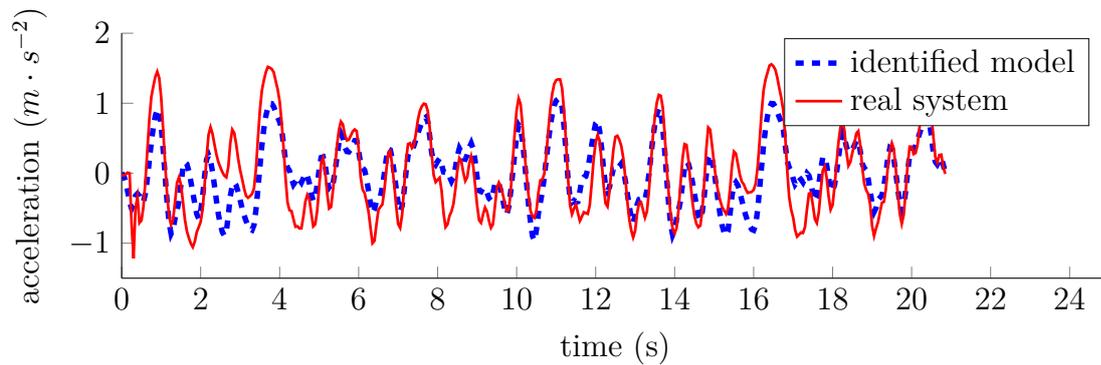
(a) Along x axis of the body frame.(b) Along y axis of the body frame.

Figure 6.17: Identification data and simulation of the identified model. Estimated acceleration of the quadcopter along x and y axis of the body frame are represented in red, and the response of the identified model is plotted in dashed blue line.

there were some cases where the user input was switched off but the states constrained were still violated. This behavior could be explained if the identified model and the open-loop model were not accurate enough. As a consequence, we have proposed to improve the identification part.

Conclusions and perspectives

7.1 Conclusions

This thesis proposes two event-based design methods ensuring different performances. The first one aims to guarantee a minimum rate of convergence of a Lyapunov function while the second one ensures *a priori* specified state constraint. Experiments have been conducted to demonstrate the benefits and limitations of each technique. In the sequel, the thesis contributions are summarized:

- I An event-based controller is proposed for discrete LTI system in Chapter 3. This is an extension of the work presented in [Durand et al., 2014a] in the discrete time domain. The event condition, which is checked periodically, is based on a Lyapunov function V_k . The basic idea is to compare the rate of decay of V_k if the control is updated or not. The stability of the event-triggered system is ensured, and furthermore the rate of convergence of the Lyapunov function can be bounded. Simulations have shown the efficiency of the proposed techniques compared to a time-triggered controller and a Send-On-Delta controller proposed in [Årzén, 1999].
- II The previous approach has been tested on a real-time system. The aim was to stabilize two angles of a gyroscope actuator around a given reference. To implement a state-feedback controller, an observer has been designed to estimate the whole state vector. Furthermore, disturbances have also been introduced into the observer design to be able to cancel their effects. The event-based controller has been implemented as well as a classical time-triggered controller. Results have shown that the number of control updates has been significantly reduced (more than 50%), moreover the disturbance cancellation improves the reference tracking significantly.
- III A new event-based controller has been introduced in Chapter 4. Event-switched technique aims to keep the ultimate bound of a disturbed discrete LTI system while reducing the number of control updates. This approach consists in applying time-triggered control if the state is too far from the origin, while opening the control loop otherwise rather than holding the last control input signal value. A region of the state-space is computed which corresponds to all the initial conditions which guarantee that the event-switched system will be bounded in a specified set.

IV Event-switched method has been applied to avoid collision between a human piloted quadcopter and virtual limits in Chapter 6. A switching condition between user and stabilizing input has been designed which guarantee specified state constraints. Event-switched approach was used to design a high level monitoring collision detection which ensures that the specified constraints are not violated. Real-time implementation has been done on a ground station with a Vicon localization system, and while most of the time the UAV does not collide with the virtual limits, there were some cases when a collision has been experienced.

Event-based control strategies have shown some encouraging results to reduce the number of control updates while keeping some performances. Performance specifications will determine which event-based method will be necessary. If stability is only of interest, methods based on Lyapunov derivative techniques are well adapted. However, if practical stability needs to be achieved while ensuring some state constrains, event-switched method can be used.

7.2 Perspectives

The first method presented in Chapter 3 could be improved to consider delay in the system as in [Durand, 2013]. Also including bounded disturbances would be a great amelioration to extend this technique to a wider class of systems. Another improvement would be to reduce the number of communications required between the sensors and the observer by considering the stability of the observation error instead of the system state.

Results for event-switched method introduced in Chapter 4 could be improved by considering inter-sample behavior in case of the application on continuous LTI systems. Another idea would be to use a Lyapunov based technique when the state is outside a RPI set, and uses the event-switched approach when the state has reached this set. This way, instead of sampling as fast as possible when the state is not in the event-set, control signal updates would be required only when a Lyapunov function of the system is not converging fast enough.

While event-switched approach demonstrates its ability to keep the UAV away of the virtual limits most of the time, there is still room for improvement. Collision has happened sometimes and could be explained by several factors. As the design of the event condition depends strongly on the model, improving it would certainly improves performances.

To simplify the design, it has been decided to consider similar dynamics along x and y axis, and the event condition has been designed in the inertial frame. For improvement we advise to implement the event condition in the body frame, by considering a different model of the quadcopter along the axis x and y of the body frame. It has been proposed

to identify the response of the robot to acceleration inputs rather than the position of the UAV when closed-loop control is applied. Furthermore, a MIMO model has been identified which considers the coupling between the inputs. PRBS control inputs have been used to properly identify the dynamics of the system in a wider frequency range. However, using a higher order model increases the complexity of the event-based control design. A good balance has to be found between complexity and accuracy of the model.

Event-switched technique is not adapted when the constraints set is not convex, improvement can be done to avoid this limitation. Rather than using the absolute position of the quadcopter and the obstacle, the relative distance between each object can be considered. Also, instead of returning to the origin of the inertial frame when a collision is about to occur, including a path planning method would be necessary to generate a safe trajectory consisting in a set of waypoints the UAV has to visit sequentially.

A Vicon motion tracking system has been used to implement the collision avoidance, using onboard sensors to perform odometry and obstacle detection would allow to achieve autonomous flight. Besides the technical implementation, the main difficulty would be to deal with the uncertainty of the measurements, however it can be easily included in the design of the event function. Similarly, implementing the position controller and the event function on the FCB would be a great step forward to achieve autonomous flight, as a ground station would not be required. Also, using another platform which implement closed-loop control of the motors speed would lead to better results as the dynamics would not be dependent on the battery state of charge.

Bibliography

- M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy. Autonomous navigation and exploration of a quadrotor helicopter in gps-denied indoor environments. In *First Symposium on Indoor Flight*, Mayagüez, Puerto Rico, July 2009.
- S. L. Altmann. *Rotations, Quaternions, and Double Groups*. Dover Publications Inc., 2005. ISBN 0486445186.
- A. Anta and P. Tabuada. To sample or not to sample: Self-triggered control for nonlinear systems. *IEEE Transactions on Automatic Control*, 55(9):2030–2042, Sept. 2010. ISSN 0018-9286. doi: 10.1109/TAC.2010.2042980.
- K. E. Årzén. A simple event-based PID controller. In *Proceedings of the 14th IFAC World congress*, volume 32, pages 8687–8692, Beijing, P.R. China, July 1999. doi: 10.1016/S1474-6670(17)57482-0.
- K. J. Åström and B. Bernhardsson. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 2, pages 2011–2016, Las Vegas, NV, USA, Dec. 2002. doi: 10.1109/CDC.2002.1184824.
- K. J. Astrom and R. M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, Princeton, NJ, USA, 2008. ISBN 0691135762, 9780691135762.
- F. Blanchini. Ultimate boundedness control for uncertain discrete-time systems via set-induced Lyapunov functions. *IEEE Transactions on Automatic Control*, 39(2):428–433, Feb. 1994. ISSN 0018-9286. doi: 10.1109/9.272351.
- F. Blanchini. Survey paper: Set invariance in control. *Automatica*, 35(11):1747–1767, 1999. ISSN 0005-1098. doi: 10.1016/S0005-1098(99)00113-2.
- B. Boisseau, S. Durand, J. J. Martinez-Molina, T. Raharijaona, and N. Marchand. Attitude control of a gyroscope actuator using event-based discrete-time approach. In *2015 International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, pages 1–6, Krakow, Poland, June 2015. doi: 10.1109/EBCCSP.2015.7300677.
- B. Boisseau, J. J. Martinez, T. Raharijaona, S. Durand, and N. Marchand. Event-switched control design with guaranteed performances. *International Journal of Robust and Non-linear Control*, 2016. doi: 10.1002/rnc.3692.

- S. Bouabdallah, P. Murrieri, and R. Siegwart. Design and control of an indoor micro quadrotor. In *IEEE International Conference on Robotics and Automation, ICRA'04*, volume 5, pages 4393–4398, New Orleans, LA, USA, Apr. 2004. doi: 10.1109/robot.2004.1302409.
- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.
- S. Durand. Event-based stabilization of linear system with communication delays in the measurements. In *American Control Conference*, pages 152–157, Washington, DC, USA, June 2013. doi: 10.1109/ACC.2013.6579829.
- S. Durand and N. Marchand. Further results on event-based PID controller. In *European Control Conference*, pages 1979–1984, Budapest, Hungary, Aug. 2009.
- S. Durand, N. Marchand, and F. Guerrero Castellanos. Simple Lyapunov sampling for event-driven control. In *Preprints of the 18th IFAC World Congress*, volume 44, pages 8724–8730, Milan, Italy, Aug. 2011. doi: 10.3182/20110828-6-IT-1002.02396.
- S. Durand, B. Boisseau, J. J. Martinez-Molina, N. Marchand, and T. Raharijaona. Event-based LQR with integral action. In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pages 1–7, Barcelona, Spain, Sept. 2014a. doi: 10.1109/etfa.2014.7005067.
- S. Durand, N. Marchand, and F. Guerrero Castellanos. Event-based Stabilization of Non-linear Time-Delay Systems. In *19th IFAC World Congress (IFAC WC 2014)*, volume 47, pages 6953–6958, Le Cap, South Africa, Aug. 2014b. doi: 10.3182/20140824-6-za-1003.01941.
- C. Goerzen, Z. Kong, and B. Mettler. A survey of motion planning algorithms from the perspective of autonomous uav guidance. *Journal of Intelligent and Robotic Systems*, 57(1):65, 2009. ISSN 1573-0409. doi: 10.1007/s10846-009-9383-1.
- L. Grüne, S. Jerg, O. Junge, D. Lehmann, J. Lunze, F. Müller, and M. Post. Two complementary approaches to event-based control. *Automatisierungstechnik*, 58(4):173–182, Apr. 2010. doi: 10.1524/auto.2010.0832.
- H. Haimovich, E. Kofman, and M. Seron. Analysis and improvements of a systematic componentwise ultimate-bound computation method. In *Proceedings of the 17th IFAC World Congress*, volume 41, pages 1319–1324, Seoul, South Korea, July 2008. doi: 10.3182/20080706-5-kr-1001.00226.
- T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski. Dynamic modelling and configuration-stabilization for an X4-flyer. *{IFAC} Proceedings Volumes*, 35(1):217 – 222, 2002. ISSN 1474-6670. doi: 10.3182/20020721-6-ES-1901.00848. 15th IFAC World Congress.

- W. Heemels, J. Sandee, and P. V. D. Bosch. Analysis of event-driven controllers for linear systems. *International Journal of Control*, 81(4):571–590, Apr. 2008. ISSN 0020-7179, 1366-5820. doi: 10.1080/00207170701506919.
- M. Herceg, M. Kvasnica, C. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In *Proceedings of the European Control Conference*, pages 502–510, Zürich - Switzerland, July 2013. <http://control.ee.ethz.ch/~mpt>.
- G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, volume 2, page 4, Hilton Head, SC, USA, Aug. 2007. doi: 10.2514/6.2007-6461.
- W. Johnson. *Helicopter Theory*. Dover Books on Aeronautical Engineering Series. Dover Publications, 1994. ISBN 9780486682303.
- E. C. Kerrigan. *Robust constraint satisfaction: Invariant sets and predictive control*. PhD thesis, University of Cambridge, 2000.
- H. Khalil. *Nonlinear Systems*. Pearson Education. Prentice Hall, 2002. ISBN 9780130673893.
- O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, mar 1986. doi: 10.1177/027836498600500106.
- E. Kofman, H. Haimovich, and M. Seron. A systematic method to obtain ultimate bounds for perturbed systems. *International Journal of Control*, 80(2):167–178, Feb. 2007.
- E. Kofman, J. A. D. Doná, and M. M. Seron. Probabilistic set invariance and ultimate boundedness. *Automatica*, 48(10):2670–2676, 2012. ISSN 0005-1098. doi: 10.1016/j.automatica.2012.06.074.
- I. Kolmanovsky and E. G. Gilbert. Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical Problems in Engineering*, 4(4):317–367, 1998. doi: 10.1155/S1024123X98000866.
- A. M. Lyapunov. The general problem of the stability of motion. *International Journal of Control*, 55(3):531–534, mar 1992. doi: 10.1080/00207179208934253.
- R. Mahony, V. Kumar, and P. Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics Automation Magazine*, 19(3):20–32, Sept 2012. ISSN 1070-9932. doi: 10.1109/MRA.2012.2206474.
- A. Manecy. *Strategies for bio-inspired visual guidance : application to control an UAV and to track a target*. PhD thesis, Université Grenoble Alpes, July 2015.

- N. Marchand. Stabilization of lebesgue sampled systems with bounded controls: the chain of integrators case. *IFAC Proceedings Volumes*, 41(2):10265–10270, 2008.
- N. Marchand, S. Durand, and F. Guerrero Castellanos. A general formula for event-based stabilization of nonlinear systems. *IEEE Transactions on Automatic Control*, 58(5):1332–1337, 2013. doi: 10.1109/TAC.2012.2225493.
- J. Martinez. Minimal RPI sets computation for polytopic systems using the bounded-real lemma and a new shrinking procedure. In *1st IFAC Workshop on Linear Parameter Varying systems (IFAC LPVS)*, volume 48, pages 182–187, Grenoble, France, Oct. 2015. doi: 10.1016/j.ifacol.2015.11.134.
- J. Martinez and H. Khennouf. Approche d'état pour la commande optimale d'un système gyroscopique. In *9e Colloque sur l'Enseignement des Technologies et des Sciences de l'Information et des Systèmes (CETSI 2011)*, page 6, Trois-Rivières, Canada, Oct. 2011.
- J. Martinez, M. Seron, and J. De Doná. Multi-sensor longitudinal control with fault tolerant guarantees. In *European Control Conference*, pages 4235–4240, Budapest - Hungary, Sept. 2009.
- K. Ogata. *Discrete-time Control Systems*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1987. ISBN 0-132-16102-8.
- S. Oлару, J. De Doná, M. Seron, and F. Stoican. Positive invariant sets for fault tolerant multisensor control schemes. *International Journal of Control*, 83(12):2622–2640, 2010. doi: 10.1080/00207179.2010.535215.
- P. Pounds, R. Mahony, and P. Corke. Modelling and control of a large quadrotor robot. *Control Engineering Practice*, 18(7):691–699, 2010. doi: 10.1016/j.conengprac.2010.02.008.
- R. Prouty. *Helicopter performance, stability, and control*. PWS Engineering, 1986. ISBN 9780534063603.
- S. Rakovic, E. Kerrigan, K. Kouramas, and D. Mayne. Invariant approximations of the minimal robust positively invariant set. *IEEE Transactions on Automatic Control*, 50(3):406–410, Mar. 2005. ISSN 0018-9286. doi: 10.1109/TAC.2005.843854.
- D. Rathbun, S. Kragelund, A. Pongpunwattana, and B. Capozzi. An evolution based path planning algorithm for autonomous motion of a uav through uncertain environments. In *Proceedings. The 21st Digital Avionics Systems Conference*, volume 2, pages 8D2–1–8D2–12, Irvine, CA, USA, Oct. 2002. doi: 10.1109/DASC.2002.1052946.
- J. Sánchez, M. A. Guarnes, and S. Dormido. On the application of different event-based sampling strategies to the control of a simple industrial process. *Sensors*, 9(9):6795, 2009. ISSN 1424-8220. doi: 10.3390/s90906795.

- J. H. Sandee, W. P. M. H. Heemels, and P. P. J. van den Bosch. Event-driven control as an opportunity in the multidisciplinary development of embedded controllers. In *American Control Conference*, volume 3, pages 1776–1781, Portland, OR, USA, June 2005. doi: 10.1109/ACC.2005.1470225.
- M. M. Seron, X. W. Zhuo, J. A. De Dona, and J. J. Martinez. Multisensor switching control strategy with fault tolerance guarantees. *Automatica*, 44(1):88–97, 2008. ISSN 0005-1098. doi: 10.1016/j.automatica.2007.05.024.
- S. Skogestad. Probably the best simple pid tuning rules in the world. In *AIChE Annual meeting*, Reno, NV, USA, Nov. 2001.
- E. D. Sontag. A ‘universal’ construction of Artstein’s theorem on nonlinear stabilization. *Systems & Control Letters*, 13(2):117 – 123, 1989. ISSN 0167-6911. doi: 10.1016/0167-6911(89)90028-5. URL <http://www.sciencedirect.com/science/article/pii/0167691189900285>.
- P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, Sept. 2007. ISSN 0018-9286. doi: 10.1109/TAC.2007.904277.
- M. Velasco, P. Martí, and E. Bini. On Lyapunov sampling for event-driven controllers. In *Proceedings of the 48th IEEE Conference on Decision and Control, held jointly with the 28th Chinese Control Conference*, pages 6238–6243, Shanghai, China, Dec. 2009. doi: 10.1109/CDC.2009.5400541.
- J. G. Ziegler and N. B. Nichols. Optimum settings for automatic controllers. *The American Society Of Mechanical Engineer (ASME)*, 64(11):759–768, 1942. doi: 10.1115/1.2899060.

Commande événementielle : application aux systèmes robotiques

Résumé — La théorie du contrôle a d'abord été conçue pour des contrôleurs analogiques. Dans ce contexte, il était pertinent d'utiliser un environnement continu pour concevoir une boucle de rétroaction. Les technologies analogiques ont été remplacées par des technologies numériques en raison de plusieurs avantages (coût, résistance au bruit, intégration, ...). Le signal de commande est maintenu constant entre les déclenchements ordonnés par une horloge périodique. Le principal inconvénient des contrôleurs périodiques est la bande passante constante requise pour effectuer un contrôle en boucle fermée dans le cas des systèmes commandés en réseau. Pour diminuer le nombre d'informations échangées il est possible d'augmenter la période d'échantillonnage, cependant une limite est imposée par la dynamique du système considéré. Dans ce contexte, la commande événementielle vise à compléter la commande périodique en proposant une méthode où les mises à jour du signal de commande sont déclenchées seulement en cas de nécessité. Dans cette thèse, deux contrôleurs événementiels sont présentés ainsi que des essais sur des dispositifs expérimentaux. La première méthode est basée sur une fonction de Lyapunov et garantit une certaine vitesse de convergence des solutions du système commandé par événement. Des expériences ont été menées sur une plateforme expérimentale temps-réel. La politique de mise à jour proposée montre une réduction significative du nombre de mises à jour du signal de commande tout en assurant des performances satisfaisantes. La seconde méthode appelée *event-switched* est dédiée aux systèmes à temps discret sujet à des perturbations bornées. Cette approche est basée sur la théorie des ensembles et en particulier sur la propriété d'invariance. Cette méthode a ensuite été adaptée pour un problème d'évitement de collision entre un drone commandé par un pilote et un environnement statique. Les résultats, après mise en œuvre sur un système temps-réel, sont présentés et discutés.

Mots clés : Commande événementielle, robotique, drone, UAV, ensembles invariants.

Event-based control: application to robotic systems

Abstract — Control system theory has first been built for analog controllers. In this context, it was relevant to use a continuous framework to design a control feedback function. Nowadays, digital technologies are supplanting analog solutions due to several advantages (cost, noise resistance, integration...). Signal is held constant between periodic triggers given by a digital clock. Periodic controllers are also referred as time-triggered because the update is only dependent on time. The main disadvantage of time triggered control is the amount of communication required to perform closed-loop control in case of networked control systems, for example when the controller and/or the actuator are connected through a network. In that case, periodic control framework requires a constant communication bandwidth to update the informations periodically. One solution to decrease the bandwidth consumption is to increase the sampling period, however performances can decrease and a bottleneck is facing when the sampling period is not adapted for the dynamics of the system. In that context, event-based (or event-triggered) control aims to improve the periodic sampling scheme by proposing a method in which updates are triggered by an event function. With classical time-triggered approaches, the control law is computed and the control signal is updated at a fixed sampling period whether this is really necessary or not. Conversely, event-based procedures do not require these periodic computations and updates, but call for resources only when they are strictly necessary. In this thesis, two event-based controllers for discrete time systems are presented and experiments on real-time platforms have been performed. The first method is based on a Lyapunov function and guarantees a certain speed of convergence of the event-based system solution. Experiments have been performed on a real-time gyroscope platform. The proposed update policy shows a significant reduction of the number of control update, while ensuring satisfactory performances. The second method called *event-switched* control is dedicated to discrete time systems subject to bounded disturbances. This approach is based on set theory and guarantees practical stability of the system solution. This method has been adapted to a problem of collision avoidance between a remotely piloted aerial vehicle and a static environment. Real-time implementation results are shown and the obtained performances are discussed.

Keywords: Event-based control, robotics, drone, UAV, invariant sets.
