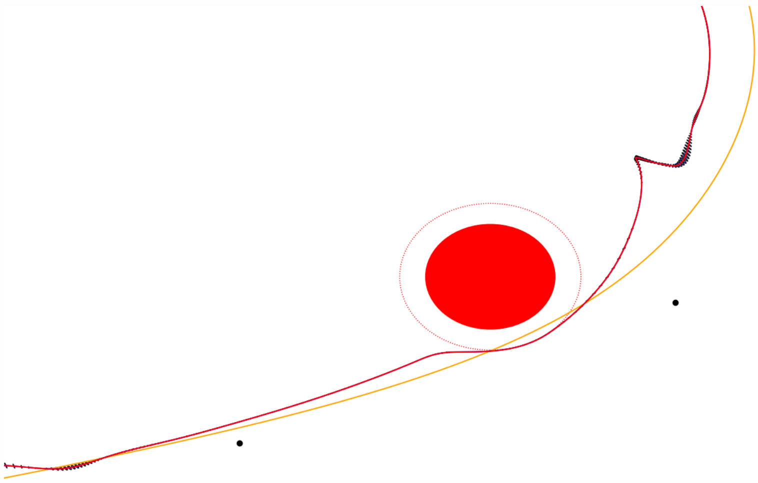
Robust Trajectory Planning



ROB 2023

August 24, 2023

Author :
J. SOUEIDAN
jonas.soueidan@ensta-bretagne.org

Supervisor:
Marco Cognetti
marco.cognetti@laas.fr

Abstract

The use of robots in our daily lives means that they need to be able to cope with the complexity of real environments, which are often uncertain. Planning a robot's movement is usually done in two stages: first an offline plan is calculated, and then the robot's commands are sent in an open loop to follow this trajectory. This approach works in an ideal world, where the robot can perform exactly the planned movement. However, its implementation fails in most practical cases due to the inevitable uncertainties in the information used by the planner, such as the internal parameters of the system. For this reason, the planned trajectory is in practice robustly executed using a motion controller that "closes the loop" between the planned motion and the actual motion, thus providing robustness. The idea of this report is to compute trajectories for the robot that are robust with respect to the uncertainties of the parameters (e.g. mass, inertia, friction coefficients of the robot), while guaranteeing a certain safety with respect to certain constraints (limits of the actuators, zones not to be crossed, obstacles to be avoided).

Résumé

L'utilisation de robots dans notre vie quotidienne implique qu'ils soient capables de faire face à la complexité des environnements réels, souvent incertains. La planification du mouvement d'un robot se fait généralement en deux étapes : un plan hors ligne est d'abord calculé, puis les commandes du robot sont envoyées en boucle ouverte pour suivre cette trajectoire. Cette approche fonctionne dans un monde idéal, où le robot peut effectuer exactement le mouvement planifié. Cependant, sa mise en œuvre échoue dans la plupart des cas pratiques en raison des incertitudes inévitables affectant les informations utilisées par le planificateur, comme les paramètres internes du système. Pour cette raison, la trajectoire planifiée est en pratique exécutée de manière robuste en utilisant un contrôleur de mouvement qui "ferme la boucle" entre le mouvement planifié et le mouvement réel, fournissant ainsi la robustesse. L'idée de ce rapport est de calculer des trajectoires pour le robot qui sont robustes par rapport aux incertitudes des paramètres (par exemple la masse, l'inertie, les coefficients de frottement du robot), tout en garantissant une certaine sécurité vis-à-vis de certaines contraintes (limites des actionneurs, zones à ne pas franchir, obstacles à éviter).

Acknowledgments

I wish to give a special thanks to Marco Cognetti and Tommaso Belvedere for the continuous help and support that they gave me all throughout my internship, and because without their insightful advice, this work would not be as it is. I would also like to thank the LAAS-CNRS for having me, and more specifically the RIS team under Simon Lacroix, that welcomed me with open hands inside their laboratory. And finally, I would like to thank my colleagues in the aerial robots team, Gianluca Corsini, Hai Nguyen, Riccardo Fuser, Alessia Frusco and José Otavio Brochado, that accompanied me during the last six months.

Contents

1. Introduction

Working as a researcher in a laboratory is completely different from working in a company, even as an intern. The final goal when doing research is to provide innovations, new ways and ideas to improve on what we already know, and for people to use those discoveries to improve our ways of doing things. In a way, research is the preamble for companies to design practical innovations. The aim of this report is to give a brief description of the environment and the work assigned to interns in the research industry, or at least my experiences in that particular regard, as well as an overview of the completion of my work.

When working with autonomous robots, and drones in particular, having a safe critical system is mandatory to ensure that unwanted behaviour do not arise when the device is running. Planning actions in advance, like finite-state machines, has a limit when the robot happens to face unaccounted events, such as sensor failures, any kind of discrepancies, noises or even unintended uses for commercial products as an example. The focus of safe critical systems is to provide a framework for a given system that would always behave as intended with respect to given constraints.

To ensure that unwanted reactions does not occur when running, there are different methods that can be used, but this project will mainly focus on Sensitivity, from [1] and Control Barrier Functions (CBF), from [2]. Sensitivity is a framework that tries to predict ahead of time, with given boundaries for uncertainties, the behaviour of the system. Simply put, how would the system (and in our case the expected trajectory) evolve if the internal parameters where to change from some reference values, due to uncertainties. CBF functions where developed as a way to impose absolute constraints on a system, and works as an optimisation problem, often in concordance with Control Value Function (CVF), to design a complete controller. In summary, CBF can be seen as choosing the best inputs to satisfy all the given constraints.

These two frameworks bring together a great synergy, as CBF can impose hard constraints on a low level, bringing an all around safety, whereas sensitivity bring the robustness to the system, such that discrepancies made at the conception can be taken into account, and therefore reducing their effect on the overall behaviour.

The report will give two different frameworks that can be used in different situations, depending on the use case, as well as a few results and interpretation for further in-depth explanations, improvements and future work on the subject.

2. The Research Environment

2.1. The LAAS-CNRS research center

2.1.1. The LAAS-CNRS

The Laboratoire d'Analyse et d'Architecture des Systèmes, abbreviated as LAAS, is a distinguished research laboratory operating under the rigorous auspices of the Centre National de la Recherche Scientifique (CNRS), located in Toulouse, France. As of this January, LAAS employs a cohort of 554 researchers. This figure does not encompass interns, administrative staff, and logistics personnel, which could contribute an estimated 150 additional individuals to the organization's workforce. Originally established in 1968 by Jean Lagasse as the "Laboratoire d'Automatique et de ses Applications Spatiales," its focal point evolved to transcend the spatial industry, adopting a global perspective that encompasses a comprehensive understanding of systems and their applications.

Closely collaborating with the Université de Toulouse, LAAS serves as a crucible for innovation across a spectrum of disciplines. Its influence extends to informatics, robotics, automation, energy, micro and nano systems, optics, and super high-frequency technology. Guided by five principal axes—Health and Environment, Industry of the Future, Energy, Space and Transport, and Mobility—the laboratory aspires to unravel the intricacies of complex systems while conscientiously considering their potential real-world applications. This multifaceted approach positions LAAS at the forefront, adeptly identifying nascent challenges in the realms of science and industry and devising solutions to surmount them.

LAAS boasts a legacy that resonates with eminent researchers, exemplified by Georges Giralt, an iconic figure in the realm of robotics. In 1977, Giralt pioneered the Robotics and Artificial Intelligence research department, unveiling Hilare, a seminal mobile autonomous robot. This heritage continues to flourish, as LAAS remains an unrivaled global institution, propelling advancements across diverse domains. Its contributions reverberate through international conferences, earning it esteemed recognition and accolades, while simultaneously serving as a crucible for prestigious symposiums and lectures.

The yearning for innovation and the relentless pursuit of knowledge are the cornerstones that underpin LAAS's enduring influence. Its symbiotic partnership with Université de Toulouse and its strategic orientation across various disciplines position it to shape the trajectory of both present and future scientific and industrial landscapes. LAAS stands as a testament to the power of collaborative research and unfaltering dedication, a beacon that illuminates the path to transformational discoveries and solutions that transcend borders.

2.1.2. The organisation of the Laboratory

The organizational structure of LAAS mirrors that of a conventional industry company, designed with meticulous precision. The administrative facet features Mr. Mohamed Kaaniche as the director, complemented by three deputies and an executive assistant. This administrative core collaborates with various auxiliary services encompassing communication, finance, human resources, and logistics. Additionally, the lab maintains dedicated technical services responsible for overseeing every facet of the lab's infrastructure, ranging from software to prototyping tools. Their mandate is to cultivate optimal conditions for researchers to immerse themselves in their work, devoid of infrastructural concerns. Beyond infrastructure, the administrative unit fosters coherence among diverse teams, fostering a collective identity within this multifaceted entity. Their support extends to organizing meetings, conferences, and scientific events, facilitating communication and publication of research articles, coordinating missions to other research centers and conferences abroad, and nurturing the lab's social fabric through sport, culture, and events, like the annual party and tournaments, fostering a strong sense of unity among members.

The scientific domain of the laboratory is compartmentalized into six distinct departments: RISC (Informatics, Networks, and Trusted Systems), ROB (Robotics), DO (Decision and Optimisation), HOPES (Super High Frequency and Photonics), MNBT (Micro Nano and Bio Technologies), and GE (Energy-related research, spanning from electricity to rocket propulsion). Each department operates under a representative responsible for managing the various research teams within the respective field. These teams' trajectories are influenced by the lab's administration and the industry's research directions. Furthermore, permanent members from the lab align with specific fields. Upon establishing a research team, a manager is designated, alongside permanent researchers who fortify the team's structure and lay the foundation for future projects. The subsequent selection of permanent team members is guided by their projects and their vision for the field's future. This paves the way for Ph.D. and post-Ph.D. positions, eventually culminating in intern positions, effectively rounding out the team.

This organizational framework bestows autonomy upon each department, fostering a coherent and manageable administrative landscape. Within each department, further divisions into research teams facilitate a concentrated focus on specific domains. For instance, within the Robotics department, three distinct research teams emerge: GEPETTO, specializing in anthropomorphic systems; RAP, dedicated to actions and perceptions; and RIS, engaged in exploring interactions with robots. These research teams, in turn, are subdivided into smaller groups centered around specific projects, typically consisting of three to five individuals.

This hierarchical framework not only imparts distinctiveness to each department but also streamlines administrative operations. By delving into smaller units of research teams and project groups, LAAS optimizes its research endeavors, embodying a dynamic and focused approach to scientific exploration.

A global overview of the organisation can be found in Appendix A.

2.2. Working as a Researcher

2.2.1. Project Management

The research within LAAS-CNRS unfolds through a structured hierarchy of research groups nested within research teams that align with specific departments, yielding a straightforward hierarchy. Each research group designates a supervisor from among its members, ideally representing the various entities engaged in the project. This practice ensures a localized representative at each relevant location. Typically, the project's funding originator assumes the role of project supervisor—often the entity that secured funding, frequently from public entities like universities, governmental bodies, or agencies. Occasionally, corporate funding plays a role, or even a combination of both public and private sources.

Factors dictating project funding are multifaceted. Initial presentations before a panel of judges outline the project's foundation, articulating existing challenges, potential improvements, and proposed solutions. This presentation accounts for adaptability during project evolution.

Subsequently, financial discussions unfold between the investing entity represented by the panel and the hosting laboratory. These discussions span considerations such as personnel recruitment, material costs, publication and conference fees, and outcome ownership.

Project funding predominantly arises from project calls issued by companies or public institutions. In some instances, funding stems from internal lab resources, drawn from permanent researchers' personal research allocations. For instance, my current project derives its financing from our lab's internal resources, utilizing a portion of the permanent researchers' personal research funds.

Each project is overseen by a funding supervisor who bridges the connection between the project and its financial support. This role involves ensuring alignment with investors' interests while propelling project advancement. In tandem, an administrative supervisor, often the head of the research team, assumes responsibility for interfacing with laboratory administration. This encompasses matters like salaries, insurance, equipment, infrastructure, and events. This dual supervision framework ensures all aspects of the project remain within both the investor's parameters and LAAS's purview, curtailing any unauthorized allocation of funds.

This meticulously structured framework streamlines project oversight, guaranteeing that financial interests are upheld and operational activities remain within the predefined boundaries. This equilibrium ensures transparent and responsible utilization of funding while facilitating collaborative and impactful research outcomes.

2.2.2. Research: Autonomy and Collaboration

Conducting research demands a distinct profile due to its inherent autonomy. The research process unfolds through a series of well-defined phases. The initial step entails selecting a topic—more precisely, identifying a pertinent issue within the scientific landscape. Subsequently, the task involves delving into existing literature, collating prior work conducted by fellow scientists. This endeavor shapes a comprehensive database, effectively encapsulating the state of the art within the chosen domain. Achieving this necessitates ongoing engagement with research papers, publications, books, attendance at conferences, and dialogues with other experts.

Subsequent to topic selection comes the crucial task of identifying a specific facet to target. This step ultimately delineates the precise issue demanding resolution, and concurrently sparks the quest for preliminary insights into potential solutions. Subsequently, further exploration of existing literature ensues, intended to ensure that no parallel endeavors are underway. Should such efforts be absent, research expands to seek insights from analogous issues in disparate domains that might have already been resolved. Subsequent stages involve delving into theoretical frameworks—a realm often characterized by innovation, yielding fresh ideas and demonstrations to propel solutions. This is then succeeded by the practical application phase, where results are verified. Frequently, outcomes achieved here are substantial enough to warrant publication in conferences.

Should the process encounter an impasse, the conventional recourse involves revisiting the literature to unearth fresh insights. Alternatively, if conclusive breakthroughs remain elusive, project recalibration might be warranted. The recursive nature of the research approach underscores the distinctive attributes required for successful engagement. This includes an innate rigor essential for managing theoretical intricacies, coupled with the ability to navigate autonomous work environments—where even team endeavors often involve individual tasks. Moreover, an unwavering wellspring of motivation and passion directed toward the field is indispensable, given the significant self-investment entailed. Research encapsulates an iterative journey, transforming concepts into setbacks until a breakthrough is finally attained.

A researcher's actual workflow necessitates an unwavering commitment to organization and rigor. This imperative is twofold: firstly, to effectively disseminate findings among fellow project members; and secondly, to facilitate recalibration when confronted with roadblocks. A weekly meeting cadence is typically adhered to, offering a platform for team members to present their progress to peers and the project supervisor. Meetings are an integral cornerstone in the research paradigm, serving a multifaceted role. They provide a mechanism for maintaining a comprehensive project overview, fostering interconnections among team members, delivering invaluable feedback on weekly accomplishments, and ultimately enabling members to provide insights and solutions to potential hurdles—benefitting from the collective wisdom accumulated.

This meticulous and iterative research process underscores the balance between autonomy and collaboration, as well as the imperative of adaptability and dedication in the face of challenges.

2.2.3. Bridging Research and Application

Working as an intern in the realm of research bears a semblance to an assistant's role within a project. The initiation of an internship entails familiarizing oneself with the entirety of the project—engaging in comprehensive literature review encompassing diverse facets related to the central objective. This preliminary groundwork is pivotal, serving to preemptively obviate any time loss upon embarking on active tasks. Once a solid foundational understanding is acquired, the initial tasks and concepts typically surface during project meetings, often stemming from varied project members. This collective input largely shapes the working plan, effectively charting the course for the internship's duration.

Research, characterized by its protracted timeline to yield results worthy of publication, often entails interns being entrusted with highly specific tasks, precluding substantial innovation. Instances include constructing a simulation environment for a designated system, establishing an experimental platform, scrutinizing system specifications, or pioneering initial implementations of theoretical equations posited in prior papers.

In my particular case, my internship aligned with the second scenario. My mandate involved implementing and applying existing insights—an endeavor that fortuitously hinged on my supervisor's original input rather than previously explored ground. To elucidate, my advisor, in consonance with the team, held nascent notions ripe for pursuit. The groundwork was partially laid, awaiting action to bring these nascent ideas to fruition. For an intern aspiring to thrive in the research milieu, poised to pursue a subsequent Ph.D., the prospect of engaging with a nascent project, an unexplored canvas rife with potential for innovative thought, was indeed a fortuitous match.

Within the laboratory setting and during my internship tenure, my role aligned with the research process's introductory and conclusive phases. These encompass poring over literature—covering past, present, and future initiatives—and propelling the actual research process, whereby novel ideas germinate, evolve, and are actualized. The sole omission was the intervening problem-solving phase. Although seemingly brief, this intermediary stage is of paramount significance, as it molds the final product's value. A well-defined problem domain, a pertinent issue to surmount, and a comprehensive idea set coalesce into a cohesive framework, simplifying the research process and implementation phases. By curbing divergence from the core issue and minimizing iterations, this strategy ensures smoother research endeavors.

Ultimately, my internship transpired at the confluence of academic pursuit and practical application—a privileged vantage point enabling me to contribute to fresh ideas while witnessing their translation into tangible outcomes. This experience illuminated the nuanced intricacies of research, underlining the indispensability of cohesive planning, continuous innovation, and the iterative journey that ushers ideas from conception to realization.

2.3. Description of my project

2.3.1. Innovation in Drone Research

Flying platforms have captivated human imagination since time immemorial, with flight representing an aspirational dream for generations. From airplanes to helicopters, humanity has triumphed over the challenges of aviation in recent decades, ushering convenience into everyday life. Commercial flights and leisurely helicopter tours for tourists are now commonplace, and the advent of drones—equipped with cameras capable of capturing breathtaking aerial vistas—has further integrated flight into our daily experiences. This evolution has propelled drones into the limelight, garnering attention due to their limitless potential applications. Their roles span from military-grade drones for unmanned interventions to First Person View (FPV) drones that offer pure recreational indulgence. In recent years, the research landscape has witnessed a surge in papers exploring autonomous robots and assistive tools, reflecting the collective aspiration to develop ever-capable aides. Consequently, the realm of flying platforms has come into sharper focus, with rotor-based drones emerging as a promising foundation for airborne assistance.

Presently, due to the intricate maneuvers involved, human control remains integral for most quadrotor operations. The most advanced autonomy offerings on the market revolve around algorithms designed to track moving targets using cameras, albeit still under human supervision. Other functionalities include automatic return when the drone ventures out of range. However, within the research arena, drone automation has yet to surge far ahead. Recent research papers have progressed beyond mere trajectory tracking to explore higher echelons of autonomy. These include trajectory and path planning, safety protocols that mandate obstacle avoidance, controllers fortified to withstand real-world conditions based on simulations, and high-order decision algorithms enabling proactive interactions by robots. This last aspect extends beyond flying platforms, embracing the capabilities of six degrees of freedom movements.

Advancements in the field have largely manifested through novel controller models that strive to incorporate human actions for precise drone responses. Safety considerations are paramount, prompting mechanisms to ensure a safe distance between the drone and its operator, thus averting potential accidents in the event of failure. Additionally, high-order control algorithms are being developed to facilitate the execution of complex tasks. However, autonomous control within the drone domain, particularly for quadrotors, remains nascent. The recent innovations, although promising, have materialized only in the past few years and are still maturing to attain substantial enhancements viable for commercial application.

Given the early developmental stage, current efforts often emphasize unidirectional innovation rather than exploring the amalgamation of different techniques for superior performance. Yet, the potential for synergy among these diverse approaches remains largely untapped, representing an avenue for cascading improvements.

In conclusion, my internship experience granted me a unique vantage point at the crossroads of academic exploration and pragmatic application. This opportunity enabled me to contribute fresh insights while witnessing their tangible transformation. This journey underscored the nuanced intricacies of research, underscoring the significance of meticulous planning, sustained innovation, and the iterative process that shepherds ideas from inception to fruition.

2.3.2. Outline of my project

During my internship, I embarked on a mission to fuse two distinct frameworks: Control Barrier Functions (CBF) and Sensitivity analysis. CBF, a method employed to impose stringent constraints on a robot's behavior through optimization, meets Sensitivity—an emerging field investigating trajectory variations in response to internal system parameters, thus enhancing robustness.

CBF operates on the premise of employing Quadratic Programming (QP), a nimble optimization technique suited for real-time control applications. It hinges on enforcing hard constraints using predefined functions on state variables—constraining factors such as maximum speed and stipulated distance boundaries. These constraints are subsequently linearized into input constraints, allowing the solving algorithm to deduce input sets that adhere to the stipulated constraints.

Sensitivity analysis, on the other hand, revolves around tracking state variable variations over time concerning fluctuating internal parameters. This endeavor yields a trajectory "tube", outlining the trajectory's boundary shifts within an initial prescribed set as the internal parameters evolve. Armed with the tube trajectory, an optimization problem can be formulated to adapt the trajectory during the planning phase. The objective is to curtail overall sensitivity by minimizing system vulnerability to internal parameter fluctuations—ultimately fortifying the system's robustness.

Over the past six months, my focus honed in on forging a functional synergy between these two theories. The overarching aim was to establish an integrated framework capable of enhancing the efficacy of critical planning and control processes. Depending on which foundational framework takes the lead, the amalgamation could amount to the refinement of a secure critical planner or a more resilient controller.

3. Drone Trajectory Planning

3.1. Problem Formulation

3.1.1. Introduction to Sensitivity

The *sensitivity* approach takes into account uncertainties as a bounded sets, and can estimate the given output bounds. The works of [1], [3] and [4] depict the sensitivity framework, and an example of trajectory optimisation through tubes trajectories. By mixing this approach with *Control Lyapunov Functions* and *Control Barrier Functions* (CLF-CBF) control method (see 3.1.2), we could therefore guarantee better safety and robustness of our systems to any kind of bounded uncertainties. The brief mathematical explanation of the framework is depicted below.

Let's first consider a dynamic system of the form :

$$\dot{q} = f_1(q,p) + f_2(q,p) \cdot u = f(q,p,u) \tag{3.1}$$

Where $p \in \mathbb{R}^{n_p}$ is a vector defining the parameters of our system, that we assume bounded, i.e. $\forall i \leq n_p, \quad p_i \in [p_{c_i} - \delta_i p_i, p_{c_i} + \delta_i p_i]$, with p_c the *nominal* parameter vector. We also denote Δp the vector of the δ_i. The *state sensitivity* is defined as:

$$\Pi(t) = \frac{\partial q(t)}{\partial p}\Big|_{p=p_c} \tag{3.2}$$

which represents the evolution of the state w.r.t. variations in the parameter vector p, and is valuated on the nominal value $p = pc$. This quantity has been introduced to improve the system behaviour in presence of parameter inaccuracies, such that one can minimize some norm of $\Pi(t)$, as an optimal shape of the trajectory with a minimal state sensitivity would make the closed-loop state evolution $q(t)$ in the perturbed case as close as possible to its evolution for the nominal case. We also define the *control sensitivity* as :

$$\Theta(t) = \frac{\partial u(t)}{\partial p}\Big|_{p=p_c} \tag{3.3}$$

which quantifies the amount of variations that would occur on the inputs w.r.t. deviations in p, also evaluated for $p = pc$. This metric designed for considering the discrepancy between the control parameters pc and the true ones p may result in some undesired inputs variation: in any system, components such as actuators are specifically chosen for the desired application, hence they need to be operated as close as possible to the conditions they were designed for. The dynamic of Π can be computed by :

$$\dot{\Pi} = \frac{\partial f}{\partial q}\Pi + \frac{\partial f}{\partial u}\Theta + \frac{\partial f}{\partial p}, \quad \Pi(t_0) = 0 \tag{3.4}$$

As shown in the works of [1], [3] and [4], it can be used to guarantee safety of a given controller u for bounded p uncertainties on a given trajectory. The proposed approach is then to optimize a given trajectory $r_d(a,t)$, where the vector a is the shaping parameter. We express the optimisation problem as :

$$a^*(p) = \underset{a}{argmin} \quad J(\Pi,\Theta)$$

Various constraints can also be added with dependence to the Θ and Π sensitivities, e.g. to ensure that $\forall p, \forall t : u \in U$. This optimisation task returns the best trajectory for a minimum deviation for any value of p. With those works, we would now like to construct a new method, incorporating the advantages of CBFs, to guarantee the safety and the robustness of our system.

3.1.2. Introduction to Control Barrier Functions

The *Control Barrier Function* (CBF) theory was developed for Safety Critical System, with works from [2] and [5]. CBFs are a guarantee for a system to stay within given bounds, and with the addition of *Control Lyapunov Functions* (CLF), we can also guarantee convergence. The CLF-CBF approach is also being refined with the *Hamilton-Jacobi Reachability* (HJ reachability) methods, to improve robustness, but at the cost of the "curse of dimensionality", as detailed in [6] and [7]. The mathematical basics for the framework are depicted below.

Let's consider a non-linear control affine system of the form :

$$\dot{q} = f(q) + g(q)u \tag{3.5}$$

With the state $q \in \mathbb{R}^{n_q}$, f and g two locally Lipschitz functions and $u \in U \subset \mathbb{R}^{n_u}$, with U the set of admissible inputs. According to [2], we can then define a *Quadratic Program* (QP) :

$$\begin{aligned} u^*(q) = \underset{(u,\delta \in \mathbb{R}^{n_u+1})}{argmin} \ & J(q,u,\delta) \qquad (CLF - CBF - QP) \\ s.t. \quad & L_f V(q) + L_g V(q) u \leq -\gamma(V(q)) + \delta \\ & L_f B(q) + L_g B(q) u \geq -\alpha(B(q)) \end{aligned}$$

Where $J(q,u,\delta)$ is our cost function to be minimised, $V(q)$ the Lyapunov control function, definite and positive, $B(q)$ the control barrier function, continuously differentiable. The δ variable is here to relax the stability of the system to guarantee that the system is solvable, and hence enforce the safety. Note that any number of constraints and CBFs can be added to the Quadratic Program to enforce safety on different aspects and levels. This controller was established as Lipschitz continuous, which guarantees the existence and uniqueness of the solution to an initial value problem, hence giving a proof for stability and safety.

With this method, given that we can find both V and B functions, the controller gives a guarantee for both stability and safety. In this case, the B function require a relative degree of one regarding the inputs, but by using Exponential CBF, or more generally High Order CBF, as depicted in [8].

In our framework, as we already have a given external controller for our systems, we will focus on the use of CBFs as a safety filter on a given input, without using any CLF.

3.1.3. Uniformed Formulation

We will use equation (3.1), with an unknown given controller, that can track the given trajectory $r_d(a,t)$.

$$\begin{cases} \dot{\xi} = g(\xi,q,a,p_c,k_c,t), \quad \xi(t_0) = \xi_0 \\ u = h(\xi,q,a,p_c,k_c,t) \end{cases} \tag{3.6}$$

The details can be found in [1], [3] and [4]. We will also be using the dynamic equations for the sensitivity variables from the above mentioned papers, just as equation (3.4) :

$$\begin{cases} \dot{\Pi}(t) = f_q\Pi + f_u\Theta + f_p \\ \dot{\Pi}_\xi(t) = g_q\Pi + g_\xi\Pi_\xi \\ \Theta(t) = h_q\Pi + h_\xi\Pi_\xi \end{cases} \tag{3.7}$$

Where the lower case variable represent the derivative variable, such as $f_p = \frac{\partial f}{\partial p}$. All other dependencies are omitted for readability. We also define $\Pi_\xi(t) = \frac{\partial \xi(t)}{\partial p}\Big|_{p=p_c}$ as the *controller internal sensitivity.*

From the online implementation of QP solvers, we need a standard formulation for our equations mentioned above. The standard way of implementing the QP is defined as follow :

$$\begin{aligned} \underset{x}{minimize} \quad & \frac{1}{2} \cdot x^T Q x + P^T x \\ subject \quad to \quad & Gx \leq h \\ & Ax = b \\ & lb \leq x \leq ub \end{aligned} \tag{3.8}$$

It is also quite easy to go from an optimisation function such as $\frac{1}{2}\|u - u_{ref}\|^2$ (least square optimization problem) to $\frac{1}{2} \cdot u^T Q u + P^T u$ by setting $P = -Q^T \cdot u_{ref}$. For the proof, see here. This manipulation is done to shift between a regular cost function optimisation, to a filter, staying as close as possible to the initial value.

3.2. Merging the frameworks

The CLF-CBF method is also not suited to take system uncertainties into account, which can lead to unauthorized behaviors of our systems. For instance, such CLF-CBF based controllers rely on a "fixed" modelisation of our system, as an example, the mass, inertia, friction or drag coefficients are computed and obtained through experiments. On the opposite, sensitivity is not designed for control, however it take uncertainties into account, and would provide an additional layer of robustness. The motivation of this work is to give a global framework of how to incorporate the CLF-CBF benefits with respect to the sensitivity of the system to provide an improved method for designing safe critical systems.

3.2.1. Online Control : Sensitivity Reducing QP

The first method to join both framework is to induce sensitivity into the CLF-CBF framework, by designing a *Sensitivity Reducing QP* (SR-QP) that would opperate on the control level of the system. Considering an *online* approach, we can estimate at each time step t_k, the sensitivity of the next chosen input u_{k+1} for the system, for a given number of steps into the future, using the equations from (3.7). To estimate the next values, we need to use a numerical integration method, such as Euler or Runge–Kutta fourth-order (RK4), and from a numerical point of view, we need something fast, accurate and convergent on a small number of iterations. The comparison between the two methods can be found in Appendix C, and for easier implementation, the Euler integration was chosen with a significant time-step. For a faster implementation, we will first restrain ourselves to a single time step, an develop the optimisation problem in the form of a QP.

First, we will rework equations (3.7) :

$$\begin{aligned}
&\dot{\Pi}(t) = f_q \Pi + f_u \Theta + f_p \\
&\Rightarrow \dot{\Pi} = (\frac{\partial f_1}{\partial q} + \frac{\partial f_2}{\partial q} \cdot u) \cdot \Pi + f_2 \cdot \Theta + (\frac{\partial f_1}{\partial p} + \frac{\partial f_2}{\partial p} \cdot u) \\
&\Rightarrow \dot{\Pi} = \frac{\partial f_1}{\partial q} \cdot \Pi + f_2 \cdot \Theta + \frac{\partial f_1}{\partial p} + \frac{\partial f_2}{\partial q} \cdot u \cdot \Pi + \frac{\partial f_2}{\partial p} \cdot u
\end{aligned} \tag{3.9}$$

However, we have to keep in mind that as $f_2 \in \mathbb{R}^{n_q \times n_u}$, the expressions of $\frac{\partial f_2}{\partial q}$ and $\frac{\partial f_2}{\partial p}$ are not easily defined, as we are using differentiation on matrices w.r.t vectors. The extension from the Jacobian computation method seems to be the best way to go, and therefore we will obtain 3-dimension matrices, also known as *tensors*. More explanation can be found in the next part of the report. From an optimisation point of view, we are interested in the relation between $\dot{\Pi}$, the sensitivity *variation* or *speed*, and u the input. The relation between the two would allow us to impact the sensitivity of the overall system by controlling the evolution of the sensitivity Π. To integrate it in the quadratic program, we need a linear relation between the two variables. Let's now start again from (3.9) :

$$\dot{\Pi} = \frac{\partial f_1}{\partial q} \cdot \Pi + f_2 \cdot \Theta + \frac{\partial f_1}{\partial p} + \frac{\partial f_2}{\partial q} \cdot u \cdot \Pi + \frac{\partial f_2}{\partial p} \cdot u$$

$$\Rightarrow \dot{\Pi} = \delta + \frac{\partial f_2}{\partial q} \cdot u \cdot \Pi + \frac{\partial f_2}{\partial p} \cdot u$$

From this set of equations, we can see that the sensitivity variation w.r.t u only really depends on f_2 and Π, the other term, δ, being constant and independent from the choice of u. To reduce the impact of our input u on the sensitivity of the system, we can consider only the second part of the equation :

$$\frac{\partial f_2}{\partial q} \cdot u \cdot \Pi + \frac{\partial f_2}{\partial p} \cdot u \leq \beta \tag{3.10}$$

Where β is a variable to be minimized, with a slight abuse of the inequality that is not define for matrices, and will be discussed later. By doing so, we obtain a constraint for the QP if we manage to make it linear in u. Let's do a quick study of $\frac{\partial f_2}{\partial q} \cdot u \cdot \Pi$. We know that $f_2 \in \mathbb{R}^{n_q \times n_u}$, so $\frac{\partial f_2}{\partial q} \in \mathbb{R}^{n_q \times n_u \times n_q}$ is a tensor of dimension 3, $u \in \mathbb{R}^{n_u}$ is a vector and $\Pi \in \mathbb{R}^{n_q \times n_p}$ is 2 dimensional matrix. An intuitive feeling would tell that the order of multiplication is not relevant as we do not multiply on the same dimensions with the two product. The vector product is summing all the slices along the second index, while the matrix product is multiplying all the slices. The proof of said product commutation can be found in the Appendix D. From this we get :

$$\left(\frac{\partial f_2}{\partial q} \cdot \Pi + \frac{\partial f_2}{\partial p} \right) \cdot u \leq \beta \tag{3.11}$$

Of the form $A \cdot u + B \leq \beta$, that can be used in a QP. We can then rewrite a new Sensitive Aware QP:

$$\begin{aligned} u^*(q) = & \underset{(u,\beta \in \mathbb{R}^{n_u+1})}{argmin} \ J(q,u,u_{ref},\beta) \\ s.t. \quad & L_f B(q) + L_g B(q) u \geq -\alpha(B(q)) \\ & \left(\frac{\partial f_2}{\partial q} \cdot \Pi + \frac{\partial f_2}{\partial p} \right) \cdot u \leq \beta \end{aligned} \tag{3.12}$$

Now, we are left to deal with the dimensional issue of this new sensitive constraint. In fact, expression (3.11) is in *matrix* form, of size $n_q \times n_p$, and we need a way to express the minimality for this constraint. First option would be considering each and every coefficient of the matrix as a constraint, i.e. $\forall i \in [\![0,n_q]\!], \forall j \in [\![0,n_p]\!]$:

$$\left(\left(\frac{\partial f_2}{\partial q} \cdot \Pi + \frac{\partial f_2}{\partial p} \right) \cdot u \right)_{i,j} \leq \beta_{i,j}$$

With this approach, we would add $n_q \times n_p$ constraints to our system, but ensure linearity w.r.t u for the QP. The other method would be to find a *function* φ, such that :

$$\varphi \left(\left(\frac{\partial f_2}{\partial q} \cdot \Pi + \frac{\partial f_2}{\partial p} \right) \cdot u \right) = \varphi \left(\frac{\partial f_2}{\partial q} \cdot \Pi + \frac{\partial f_2}{\partial p} \right) \cdot \varphi(u)$$

This function would allow for a smooth evaluation of the overall constraint and avoid including too many constraints into the system, rendering it way more easier to solve. However, finding such a function is not trivial, as it could be the subject of an entire publication in theoretical mathematics and, thus, we will use the first option.

To fit the equations in (3.8), let's first go back to [9]. From this work, we have the definition of a few basics from tensor arithmetic. The first one needed is the mode-n unfolding, going from a tensor to a matrix. The second one is the definition of the mode-n product from a tensor and vector, as the inner product of the mode-n fibers with said vector. In other words, $\left(\frac{\partial f_2}{\partial q} \cdot \Pi + \frac{\partial f_2}{\partial p} \right) \cdot u$ can be written as inner products with the mode-n unfolding of the tensor. Denoting $\boldsymbol{\mathcal{X}} = \frac{\partial f_2}{\partial q} \cdot \Pi + \frac{\partial f_2}{\partial p}$ the tensor, and using the same tensor notations from [9] $\forall i \in [\![0,n_q]\!], \forall j \in [\![0,n_p]\!]$:

$$(\boldsymbol{\mathcal{X}} \cdot u)_{i,j} = X_{2_{j,i}} \cdot u$$

So, to unfold the product as $n_q \times n_p$ row size matrix, we can write it down as :$X_2^T \cdot u$, which is a $n_q \cdot n_p \times 1$ matrix, of the required form for the QP implementation 3.8, with $G = X_2^T$. As we also want to minimize β, that correspond to the h for the QP, we have to include all β coefficient inside the input u, by expending all matrices with ones and zeros, to match the equations.

So, by expanding $u^* = [u^T \quad \beta^T]^T \in \mathbb{R}^{n_u + n_q \cdot n_p}$,

$$G = \left[\begin{array}{c|c} X_{2_{0,:}} & \\ \cdot & \\ \cdot & -I \\ \cdot & \\ X_{2_{n_q \cdot n_p,:}} & \end{array} \right]$$

Where I is the identity matrix of size $n_q \cdot n_p$. By using this defined matrix G and vector u, we get a set of equations such as $\forall i \in [\![0,n_q \cdot n_p]\!], \quad X_{2_{i,:}} \cdot u - \beta_i \leq 0$. Based on 3.8, this also leads to $h = 0$, and obviously A and b are also equal to 0 as we do not have any equality constraints in our system. The same way of expending matrices and vectors can be applied for higher dimensions systems, as well as including Barrier Functions and relaxation variables inside the

QP. According to test benches, the solving QP algorithms can handle pretty efficiently up until a hundred dimensions, which is more than necessary.

As a side note from (3.10), we can see that if we minimize beta, the sensitivity speed coefficients can grow infinitely negative. It can be be taken into account by adding the respective opposite constraint to keep it close to zero, at the expense of doubling the constraints. During the simulation, such a case never occurred, as sensitivity tends to grow, but it still should be mentioned.

The main setback of using such an implementation of a single step minimisation, is that we only take into consideration the *drift* term of the state sensitivity, and we do not propagate it over time, hence resetting the sensitivity to zero at each step. This induce the loss of the dynamical evolution of the sensitivity over time, making the use of such a method not useful as it is, as shown later in the results.

3.2.2. Offline Planning : Sensitivity Aware QP

The second method of merging the frameworks is to implement a CBF-QP as part of the controller used in the sensitivity framework, and finding an expression of the new state sensitivity. For this we will develop a *Sensitivity Aware QP* (SA-QP), designed for taking deviations into account and capable of propagating the sensitivity dynamics. For offline planning, we need to come up with a way to compute the sensitivity dynamics of the QP problem. Such computations have already been done in previous work in the Artificial Intelligence field, like in [10], where the next part is inspired from.

Based on the standard QP formulation (3.8) mentioned above, one can write the corresponding KKT conditions of such optimisation problem, and it leads to the Lagrangian:

$$L(x,\nu,\lambda) = \frac{1}{2}x^T Qx + P^T x + \nu^T(Ax - b) + \lambda^T(Gx - h) \tag{3.13}$$

where ν and λ are the dual variables on equality and inequality constraints respectively. From 3.13, we can write the KKT conditions for stationarity, primal feasibility and complementary slackness as :

$$\begin{aligned} Qx^* + P + A^T\nu^* + G^T\lambda^* &= 0 \\ Ax^* - b &= 0 \\ diag(\lambda^*)(Gx^* - h) &= 0 \end{aligned} \tag{3.14}$$

where x^*, ν^* and λ^* are the optimal primal and dual variables of the optimisation problem, and $diag(.)$ represents a diagonal matrix from a given vector. From this set of conditions, we can take the differentials, leading to the results from [10] :

$$\begin{aligned} dQx^* + Qdx + dP + dA^T\nu^* + A^T d\nu & \\ + dG^T\lambda^* + G^T d\lambda &= 0 \\ dAx^* + Adx - db &= 0 \\ diag(\lambda^*)(dGx^* + Gdx - dh) & \\ + diag(Dx^* - h)d\lambda &= 0 \end{aligned} \tag{3.15}$$

Written in a more compact matrix form :

$$\begin{bmatrix} Q & G^t & A^T \\ diag(\lambda^*)G & diag(Gx^* - h) & 0 \\ A & 0 & 0 \end{bmatrix} \begin{bmatrix} dx \\ d\lambda \\ d\nu \end{bmatrix} = \\ -\begin{bmatrix} dQx^* + dP + dA^T\nu^* + dG^T\lambda^* \\ diag(\lambda^*)dGx^* - diag(\lambda^*)dh) \\ dAx^* - db \end{bmatrix} \tag{3.16}$$

Renaming both KKT matrices for readability, we end up with :

$$K_x \begin{bmatrix} dx \\ d\lambda \\ d\nu \end{bmatrix} = -K_d \\ \Leftrightarrow \begin{bmatrix} dx \\ d\lambda \\ d\nu \end{bmatrix} = -K_x^{-1}K_d \tag{3.17}$$

We therefore have to keep in mind that K_x has to be invertible, which heavily depends on whether or not the different constraints are active. The derivation process requires some conditional behaviour for implementation.

We established a way to compute the differentials of the optimal primal and dual variables of the QP. The next step is to merge it with the Sensitivity framework given in (3.7). We obtain the following set of equations :

$$\begin{cases} \dot{q} = f_1(q,p) + f_2(q,p) \cdot u^* \quad q(t_0) = q_0 \\ \dot{\xi} = g(\xi,q,a,p_c,k_c,t), \quad \xi(t_0) = \xi_0 \\ u = h(\xi,q,a,p_c,k_c,t) \\ (u^*,\lambda^*,\nu^*) = QP(q,p_c,u) \end{cases} \\ \begin{cases} \dot{\Pi}(t) = f_q\Pi + f_u\Theta^* + f_p, \quad \Pi(t_0) = \Pi_0 \\ \dot{\Pi}_\xi(t) = g_q\Pi + g_\xi\Pi_\xi, \quad \Pi_\xi(t_0) = \Pi_{\xi 0} \\ \Theta(t) = h_q\Pi + h_\xi\Pi_\xi \\ (\Theta^*(t),\Lambda^*(t),N^*(t)) = -K_x^{-1}(K_{d,q}\Pi + K_{d,u}\Theta) \end{cases} \tag{3.18}$$

where $\Theta^*(t)$, $\Lambda^*(t)$ and $N^*(t)$ are defined as the sensitivity of the QP optimal primal and dual variables with respect to the internal parameters p, defined in the same way as in section 2, and $K_{d,x}$ corresponds to the derivative of the K_d matrix w.r.t to x. With this system of equations, we can compute the evolution of the system sensitivity across time, with a QP optimisation problem inside the control loop. Such improvement can be very useful for Trajectory and Motion Planners, as we can have the safety guarantees from multiple Barrier Functions inside a QP safety filter, while having the robustness of sensitivity accounted for.

Finally the last improvement that can be made is providing the system equations with the sensitivity augmented state. That can be accomplished be computing the state uncertainty ellipsoid at a given time t_k, extract the range of variation in the desired direction in the state space, and taking the worst case scenario value. The equations are :

$$K_\Pi = \Pi^T W \Pi \\ \alpha^* = \sqrt{n^T K_\Pi n} \\ q* = q \pm \alpha^* \tag{3.19}$$

Where K_Π is a positive semi-definite matrix defining the ellipsoid, W represents the diagonal matrix Δp, n the direction vector and finally α^* represents the maximum deviation. For a distance constraint, we can choose n as the worse direction for each constraint, and obtain the worse deviation for the state resulting from $\alpha*$.

3.3. Systems Modelisation

3.3.1. Unicycle Environment

We will be using the standard unicycle model, with states $q = [x,y,v,\theta,\omega]^T$, $p = [r_w,b,a,m,I]^T$ the internal parameters vector and $u = [\dot{w}_r,\dot{w}_l]^T$ the input vector. r_w represent the radius of the wheel, b the distance of the wheel from the center of mass, a a slight forward displacement for the position to involve the rotation speed, m the mass and I the inertia. We obtain the following state equations :

$$\dot{q} = \begin{pmatrix} v*\cos\theta - a*\omega*\sin\theta \\ v*\sin\theta + a*\omega*\cos\theta \\ -a*\omega^2 \\ \omega \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m} & 0 \\ 0 & 0 \\ 0 & \frac{1}{I} \end{pmatrix} \cdot \begin{pmatrix} \frac{r_w}{2} & \frac{r_w}{2} \\ \frac{r_w}{2*b} & -\frac{r_w}{2*b} \end{pmatrix} \cdot \begin{bmatrix} \dot{w}_r \\ \dot{w}_l \end{bmatrix} \tag{3.20}$$

We can rewrite the equations in the form $\dot{q} = f_1(q,p) + f_2(q,p) \cdot u$ so that it becomes easier to use as an affine input system. We can denote a desired trajectory $r_d(\alpha) = [x_d,y_d]^T$ where α is a vector of parametric description. The inertia of the wheels have been neglected, and the r_w and b parameters intervention moved from the translation and rotational speeds $vand\omega$ to the acceleration, for a simpler expression. For the controller design, we will use a linear feedback control loop, defined as such :

$$\begin{aligned} \dot{\xi} &= rd - r = g(q,\alpha) \\ u = A(q,p)^{-1} &* (v(\xi,\alpha) + b(q,p)) = h(\xi,q,\alpha,p) \end{aligned} \tag{3.21}$$

With $A = L_{f_2}L_{f_1}r(q) \in \mathbb{R}^{2\times 2}$, $b = L_{f_1}^2 r(q)$ and $v = [\ddot{r}_d + k_v \cdot (\dot{r}_d - \dot{r}) + k_p \cdot (r_d - r) + k_i \cdot \xi_{xy}]$. We also define $\xi = [\xi_x,xi_y]$, and $r = [x,y]^T$ the actual trajectory of the unicycle.

To implement the sensitivity equations from (3.7), we use symbolic computation to compute the different jacobian matrices needed, leaving the details to improve readability.

Let's now define a simple control barrier function that impose an inequality constraints to avoid a given obstacle, i.e. :

$$h_i(q) = (x - x_{obj,i})^2 + (y - y_{obj,i})^2 - r_{obj,i}^2 \tag{3.22}$$

where $(x_{obj,i},y_{obj,i},r_{obj,i})$ define for each obstacle $i \in \mathbb{N}$ described as a sphere, a set of coordinates$(x_{obj,i},y_{obj,i})$ and a radius $r_{obj,i}$. In case the state variables inside the barrier function (in this example, x and y) are not of relative degree 1 with respect to the inputs of the system, we have to move on the exponential control barrier function. The fundamental theory is depicted in [2], and it leads to the given inequality to be respected for the system to remain safe :

$$L_{f_1}^{(r_i)} h_i(q) + L_{f_2} L_{f_1}^{(r_i-1)} h_i(q) \cdot u + \sum_{j=1}^{r_i-1} k_j L_{f_1}^{(r_i-j)} h_i(q) \geq 0 \tag{3.23}$$

where r_i represent the lowest relative degree of the concerned outputs in h_i, and k_j constants that needs to be tuned for each constraints. Details can be found in [2].

This formulation allows to design more easily barrier function for all kind of constraints on the systems states. Using symbolic computation, such expressions (3.23) can be computed very easily from the state definition, the evolution function f and the initial constraint h_i.

From such control barrier functions, we are able to construct the matrices G and the corresponding vector h from (3.8), such that for N_c given constraints :

$$G = \begin{bmatrix} L_{f_2} L_{f_1}^{(r_1-1)} h_1(q) \\ . \\ . \\ . \\ L_{f_2} L_{f_1}^{(r_{N_c}-1)} h_{N_c}(q) \end{bmatrix}$$
$$h = -\begin{bmatrix} L_{f_1}^{(r_1)} h_1(q) + \sum_{j=1}^{r_1-1} k_j L_{f_1}^{(r_1-j)} h_1(q) \\ . \\ . \\ . \\ L_{f_1}^{(r_{N_c})} h_{N_c}(q) + \sum_{j=1}^{r_{N_c}-1} k_j L_{f_1}^{(r_{N_c}-j)} h_{N_c}(q) \end{bmatrix} \tag{3.24}$$

3.3.2. Quadrotor Environment

We will be using a standard Quadrotor model, with a few modifications to fit the SR-QP and the SA-QP. First, we consider a classic modelisation, with 4 aligned rotors at an equal distance from the center of mass. We note $q(t) = [x,y,z,v_x,v_y,v_z,\phi,\theta,\psi,p,q,r]^T$ the state of the quadrotor, with (x,y,z) as the position, (v_x,v_y,v_z) the translation speeds, (ϕ,θ,ψ) the roll, pitch and yaw of the drone, and (p,q,r) the angular velocity vector. We can write this set of equations :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} \tag{3.25}$$

$$\begin{pmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{pmatrix} = \frac{1}{m} \cdot R(\phi,\theta,\psi) \cdot \begin{pmatrix} 0 \\ 0 \\ F_t \end{pmatrix} + \begin{pmatrix} -b_x \cdot v_x \\ -b_y \cdot v_y \\ -b_z \cdot v_z + g \end{pmatrix} \tag{3.26}$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & sin(\phi)tan(\theta) & cos(\phi)tan(\theta) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & \frac{sin(\phi)}{cos(\theta)} & \frac{cos(\phi)}{cos(\theta)} \end{pmatrix} \cdot \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{3.27}$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} =$$
$$\begin{pmatrix} \frac{1}{I_x}[(I_y - I_z)qr + lK_t(w_4^2 - w_2^2)] \\ \frac{1}{I_y}[(I_z - I_x)pr + lK_t(w_1^2 - w_3^2)] \\ \frac{1}{I_z}[(I_x - I_y)pq + K_d(w_1^2 - w_2^2 + w_3^2 - w_4^2)] \end{pmatrix} \tag{3.28}$$

For the control of the drone, we use a cascaded PID approach, and inner control for Attitude/Altitude and an outer control for Translation/Position. The Position gains are set up such that the response time is slower, as it is used to track the desired roll and pitch angle that would lead to the ideal position from the attitude PID, and because small angular variations will cause big disturbances in positioning. Attitude and Altitude control makes sure the drone stays at the desired height and the desired angles to ensure the drones moves in the right direction. This PID controller can be improved in various ways, or even be switched with other designs like a Geometric Controller in [11] , but it is very easy to implement.

For the sensitivity framework, the entries w_1,w_2,w_3 and w_4 needs to be used in a affine way regarding the previous equations. The input linearization in the form of 3.1 is not detailed, as well as the g and h functions for the sake of readability as the expressions are based on the same as in 3.3.1, and very detailed in the literature. In the same way, the jacobian matrices are omitted, as the expressions are not relevant in this report.

For the CBF used in the experiments, we will use (3.22), but as one dimensional constraint on the altitude, for the sake of easier implementation as a proof of concept. However, more complex functions can be used, such as in [12] and [13].

4. The outcome of my work

4.1. Theory Work Difficulties

4.1.1. Theoretical Foundation

When embarking on the intricate journey of research, one of the initial hurdles that materializes is the essential need for a solid theoretical foundation. This phase precedes the tangible experimentation, whether it takes place within a simulated digital realm or involves hands-on manipulation in a real-world experimental setup. Within the expansive domain of control theory, comprehending fundamental concepts like minimizing errors or identifying optimal input configurations for desired outputs may initially appear relatively straightforward. However, as the research voyage delves deeper into the intricacies of scientific exploration, it unfailingly steers towards more complex and nuanced territories.

These uncharted territories include the realm of optimization problems, the function of solvers, the nuances of convergence proofs, and the intricacies of establishing safety assurances. The complexity of these subjects transcends the realm of practical application, necessitating a plunge into the theoretical depths of mathematics. While established theoretical principles often prove versatile when applied to common problems, extrapolating their significance within the context of specific research endeavors demands a nuanced approach. In such instances, the pursuit of highly specialized mathematical behaviors that have largely been explored within theoretical confines invariably requires a more intensive and time-consuming investigative process.

Moreover, navigating through the theoretical underpinnings of research brings its own set of mathematical challenges to the fore. Drawing from personal experience, during my research endeavors, I encountered instances where unraveling derivatives for optimization problem solutions was imperative. Additionally, traversing the intricate landscape of three-dimensional matrix (tensor) multiplication involving matrices and vectors was an indispensable task. While the theoretical foundation exists to address these challenges, the practical execution often demands a comprehensive understanding of the context and a fine-tuned ability to decipher complex notational variations.

It is within this intricate tapestry of mathematical exploration that the quest for precise references, especially when grappling with different notational conventions, emerges as a task of paramount importance. This undertaking necessitates a meticulous and thorough exploration across a diverse array of academic sources, each contributing unique insights to the underlying mechanisms that underpin these mathematical concepts. Only through such an extensive and exhaustive journey can one adequately grasp the complexities and nuances that pave the way for practical application.

In essence, navigating the labyrinthine landscape of theoretical groundwork within research is a multifaceted endeavor. It demands an inherent curiosity, a relentless pursuit of knowledge, and an unwavering commitment to unraveling the intricacies that shape our understanding of the theoretical foundations underpinning scientific exploration. Through this voyage, researchers uncover the synergy between theory and practice, bridging the gap between abstract concepts and tangible outcomes.

4.1.2. Navigating References in Research

When engrossed in the realm of research, a pivotal skill emerges: the ability to unearth and harness references. This proficiency holds sway not just in the context of research endeavors, but also for personal enrichment. In the arena of research, the act of discovering apt references isn't always straightforward. While certain fields or topics may boast shared foundations, fostering a common framework for terminology and models, venturing into more specialized realms can be akin to navigating a labyrinth. These intricate corridors are replete with esoteric theories and recent publications that often play second fiddle to foundational literature, leading to a trove of overlooked insights. Furthermore, when seeking to forge new paradigms or establish nomenclature, choices become subject to fervent debates, potentially obscuring these pioneering works from initial searches. Often, uncovering pertinent papers mandates an exhaustive trawl through references to ensure no gems remain undiscovered.

Beyond the realm of research, maintaining a pulse on discoveries and publications holds immense personal value. It serves as a conduit for broadening intellectual horizons, offering fresh perspectives on specific concepts and potential methodologies for troubleshooting problems—especially those tethered to mathematical intricacies. In some serendipitous instances, one might stumble upon an epiphany, a spark ignited by the work of fellow researchers. It proves stimulating to venture beyond one's immediate field of study, traversing the domains of biology, physics, and even psychology—particularly pertinent when engaging with human subjects. Such excursions serve two primary purposes. Firstly, they ignite creativity and innovative thinking, introducing novel concepts and unexplored avenues that can break the deadlock of stagnation or point toward future enhancements. Secondly, these cross-disciplinary sojourns often unveil strikingly similar solutions, unveiling mathematical quandaries that transcend boundaries. In the course of my endeavors, I've encountered optimization problems—a versatile conundrum that traverses diverse scientific fields.

The odyssey through references, both as a cornerstone of rigorous research and a source of personal growth, underscores the interconnectedness of knowledge across domains, beckoning curious minds to bridge the gap between what's known and what's waiting to be discovered.

4.2. Simulation Issues

4.2.1. Designing Practical Models

My work found its practical application by leveraging two distinct models: the unicycle and the quadrotor. This approach enabled the exploration of the newly developed techniques across different environments and varying levels of complexity. Striving for optimal outcomes necessitated a judicious selection of the model description and formulation, delving into diverse configurations to achieve the finest results. The delicate balance involved selecting a model that mirrors the real drone sufficiently for realistic data gathering, while avoiding undue computational intricacies.

The duet of models—unicycle and quadrotor—stood as the foundation for my investigation, with both equipped with stable controllers that facilitated the execution of basic trajectories. B-splines and Lissajous curves served as the bedrock for these trajectories, enabling straightforward expressions and derivative computations. The crux of my approach lay in incorporating an optimization problem with one or more Control Barrier Functions (CBF), thereby subjecting the integrated system to comprehensive testing. Fulfilling the initial hypothesis for the new controller also posed an essential criterion. For my research, this entailed explicit formulations of system dynamics and controllers, a prerequisite for calculating the Jacobians that govern sensitivity dynamics. Furthermore, specific requirements governed the model dynamics for the positions of the unicycle (x, y) and the quadrotor (x, y, z, ψ).

For the unicycle, a feedback linearization controller coupled with a PID guideline for 2D trajectory tracking necessitated that both x and y share the same relative degree concerning the two inputs (right and left wheel accelerations). This condition was paramount not only for the controller but also for the CBF employed for obstacle avoidance.

In contrast, the quadrotor was steered by an attitude controller and a position controller—a design comprising six PID controllers, each tailored to a distinct degree of freedom. Translating this design to the four rotor speeds, the inputs for the quadrotor, was the chosen approach. To address the relative degree discrepancy between thrust and moments, a basic altitude boundaries CBF was preferred. This simplified implementation from a programming standpoint while preserving accuracy.

Additionally, adherence to control affine dynamics was pivotal for both the sensitivity and CBF frameworks. Aligning with these prerequisites demanded adapting state-of-the-art modelizations for both the unicycle and the quadrotor, drawing from sources such as [1], [14], and [15]. For the unicycle, the tracked position was shifted from the center of mass, augmenting the standard kinematic model into a dynamic model to accommodate potential variations in mass and inertia calculations. Similarly, the quadrotor required meticulous modelization, with choices ranging from quaternions to rotation matrices or Euler angles. Opting for the Euler angles representation, as seen in section 3.3.2, emerged as the optimal decision. This representation facilitated the computation of drone dynamics, proving accurate in hovering positions, and when utilizing motor speeds as inputs. This choice, in contrast to thrust/moments controlled drones, enabled the inclusion of thrust and torque coefficients for sensitivity, essential for accuracy. Moreover, the incorporation of the speed drag coefficient further refined the representation, intensifying sensitivity disturbances during simulation.

4.2.2. Simulation Challenges

The simulation phase of this project proved to be the most time-consuming endeavor, fraught with unforeseen challenges that tested my problem-solving prowess. The journey began with model selection, a seemingly straightforward task that morphed into a significant setback as the project progressed. The initial models, while adequate for the evolving project, fell short of meeting the requisite relative degree for the online controller (3.2.1). The process of reconfiguring these models to align with project demands, ensuring their seamless integration with the broader framework, consumed considerable debugging time.

The journey through simulation was rife with further obstacles. Ensuring system stability emerged as a paramount concern, given my reliance on PID controllers and guideline principles. The tuning process for these controllers is undeniably intricate. For each modelisation of both systems, it demanded meticulous attention. Factors influencing simulation stability spanned the integration time step, controller frequency, PID coefficients, and the initial set of nominal internal parameters. Remarkably, the path to stability was divergent for the unicycle and the quadrotor models.

In the case of the unicycle, the stabilization process followed a relatively straightforward trajectory due to the inherent simplicity of the system. Contrarily, the quadrotor—a highly unstable system—demanded an intricate approach involving the implementation of six distinct PID controllers.

Initiating the stability tuning process commences with the selection of an appropriate integration time step, typically skewed toward lower values for enhanced accuracy and minimal drift. The subsequent step involves configuring the controller frequency, a pivotal determinant of system responsiveness. However, the crux of stability rests within the fine-tuning of the PID coefficients. In this realm, a simulation loop comes into play, generating the Mean Squared Error (MSE) of the system tracking the desired trajectory. This loop then becomes the cost function for an optimization algorithm, which, in my case, was scipy.optimize.minimize. Nonetheless, the intricacy deepened when dealing with the quadrotor model, necessitating a segmented optimization process due to the staggering 18 coefficients that demanded attention.

This division allowed for a stepwise approach, beginning with the altitude (z) coefficients, linked exclusively to the thrust of the four rotors. Subsequently, the focus shifted to the attitude controller, a component requiring further division as each angle objective was selected independently. The final stage homed in on the position coefficients, which in turn dictated the desired angles for the attitude controller. As a nuanced layer, the Higher Order Control Barrier Function (HOCBF) made its presence felt, accentuating the challenges. The relative degree of position exceeding one mandated the incorporation of two tuning coefficients for the Quadratic Program (QP) optimization problem, ensuring both stability and adherence to barrier function constraints. As the project neared its conclusion, attention shifted to sensitivity dynamics—a crucial domain requiring the computation of Jacobians for system dynamics and the controller with respect to system parameters. The unicycle model's simplicity allowed for manual calculations. However, the Quadrotor model posed a challenge, leading to the adoption of symbolic computation for practicality.

Symbolic computation, while advantageous, came with its own costs. Learning the library and integrating it into the project demanded time and effort. Debugging ensured accurate model representation and interaction with the simulation environment. This investment, though significant, empowered the project with essential capabilities for tackling sensitivity dynamics in the Quadrotor model.

Symbolic computation's inclusion exemplified the project's commitment to accuracy, despite the learning curve and implementation challenges. This commitment ultimately enhanced the project's depth and precision.

4.3. Final Results

4.3.1. Results from Simulation

Online SR-QP

In the context of the SR-QP method employing an online controller, noteworthy results emerge. The chosen environments were a unicycle following a B-spline trajectory without any obstacle, with a Δp of one-tenth of the original nominal parameters p_c. For the quadrotor, the trajectory was a static 3D point, with the same value for Δp as one-tenth of p_c. For performance comparison, a multitude of robots were generated with sets of randomized parameters, given that each set of parameters generated is applied to a standard design and to the SR-QP design. This allows for a more accurate evaluation of the SR-QP.

The efficacy of the SR-QP is nuanced, presenting distinct outcomes. In cases such as the unicycle or drone take-off, the benefits might not be immediately apparent. The unicycle scenario displays minimal deviation from our approach, possibly due to system robustness overpowering internal parameter disturbances. For the quadrotor, an initial phase of PID stabilization could contribute to the QP's divergence upon activation. The QP's performance is intrinsically linked to simulation frequency, the displacement of internal parameters Δp, and the matrix Q, which shapes new input behaviors through the cost function. When simulating multiple drone trajectories with varying internal parameters, toggling the Sensitivity QP on or off, and considering different time steps and matrix Q configurations, divergence patterns fluctuate. Some specific integration frequencies exhibit slightly greater robustness in QP-equipped drones, possibly due to optimized coefficients for this frequency. Nevertheless, higher simulation rates reveal instability in approximately 1 or 2 out of a hundred cases. This suggests the QP might steer drones toward states where internal parameters have less influence, potentially leading to minimal control input—akin to the drone attempting to fall to counteract induced disturbances from movement. In select instances, the QP exhibits successful convergence, even demonstrating faster convergence when parameter uncertainties are narrow. A comprehensive presentation of simulation outcomes can be found in Appendix E.

The execution time for solving the QP across a single simulation averaged around $0.003s$ during testing, with occasional spikes reaching $0.01s$ for both simulation environment. This rapid response time positions the QP for effective online utilization across diverse scenarios.

Offline SA-QP

The SA-QP method also yields a blend of outcomes. For the simulation environment, the unicycle was tasked to follow a B-spline trajectory, with two additional obstacles resting on the trajectory, to activate the CBF constraints. For the quadrotor, the tasked trajectory was a 3D Lissajous trajectory to follow, with a height limitation to activate the CBF constraint. In the same way as for the SR-QP, the internal parameters deviation boundaries were set to one-tenth for both systems.

In contrast to the SR-QP approach, the SA-QP in the quadrotor scenario does not introduce instability during the initial phase without active constraints. Only upon reaching maximum altitude does the QP initiate altitude regulation, imposing a new thrust input set for the drone.

For the unicycle, a comparison between the SA-QP and the conventional QP implementation does not manifest groundbreaking distinctions. Trajectories and sensitivities remain closely aligned to one another, such that the benefits of the SA-QP are not worth noting for trajectory tracking or sensitivity minimisation.

A notable advantage of the new method arises from the augmented state's added boundary encompassing maximum deviation. While the standard implementation in simulation adheres to constraints, real-world disturbances might lead to constraint violations. The augmented state introduces an additional "reactive zone" for system evolution, mitigating the risk of crossing boundaries. In theory, this offers heightened robustness against uncertainties in internal parameters.

Moreover, in the drone scenario, the standard QP elevates system sensitivity when constraints are active, unlike the newly developed approach that takes sensitivity into account within the QP formulation. This underscores how the foundational framework, depending on the system and imposed constraints, could lead to greater state deviations. Incorporating sensitivity as a variable enhances robustness and fortifies safety assurance.

The most compelling simulation results can be explored in Appendix F.

4.3.2. Room for Improvement

Despite concerted efforts, the outcomes of the preceding phase left room for greater satisfaction. An analysis of the various simulations unveiled that the implementation of CBF QP had an unsettling impact on system stability. In particular, when a constraint was active, the CBF QP amplified sensitivity over time, resulting in potential exponential growth that could breach barriers. Conversely, the augmented sensitivity controller exhibited the intriguing behavior of expanding safety distances from obstacles as the simulation converged. This observation suggests the potential superiority of the sensitivity augmented controller in real-world scenarios, where disturbances and inaccuracies can play a disruptive role absent in simulations.

To extract more significant insights, refining the model to encompass additional complexities such as Gaussian noises and frictions could yield richer findings by challenging the controller's tracking abilities. Altering tuning coefficients for the controller and CBF constraints introduces an avenue for dramatic system behavior transformation. However, it is crucial to maintain system stability amid such adjustments, as theoretical superiority might lie outside the stable parameter space.

Exploration extends to internal system parameters—dimensions, mass, inertia, and drags—whose manipulation could significantly influence outcomes. These parameters, though, must align with real-world platform attributes for simulation accuracy. The selection of uncertainty boundaries for these parameters also impacts sensitivity dynamics. Consequently, choices must strike a balance between meaningful results, realistic representation, and divergence avoidance. Similar to previous considerations, altering these parameters necessitates recalibrating tuned coefficients.

Fine-tuning potential extends to simulation integration time step and controller frequency. Higher rates promise heightened stability by enabling quicker error correction, while excessively slow rates can exacerbate divergence and hinder imposing Sensitivity-based constraints on mounting sensitivities. Lastly, task design alterations could enhance differences between old and new methods. Sensitivity's efficacy shines in challenging scenarios where traditional methods falter due to internal parameter inaccuracies.

In conclusion, while the project journey has unearthed intriguing insights and advancements, the door remains open for refinement and exploration across various dimensions, each promising to unravel deeper layers of understanding and improvement potential.

4.3.3. Future work

Moving forward, the natural progression of the project entails a concerted effort to enhance the simulation framework. This endeavor is pivotal not only for optimizing outcomes but also for establishing the credibility of the newly developed method. By refining the simulation process, we can unlock a more comprehensive understanding of the method's potential. This foundational step holds the key to validating the merits of a Sensitivity-based framework, thereby setting the stage for subsequent phases and bolstering the case for its adoption across a spectrum of applications.

Parallel to simulation refinement, a promising avenue for advancement involves translating these meticulously crafted models into real-world terrain experiments. As underscored by the earlier exploration of sensitivity dynamics, the most pronounced benefits arise when accounting for sensitivity factors in practical scenarios. Bridging the gap between simulation and reality, these real terrain experiments provide an unparalleled opportunity to showcase the tangible advantages of integrating Sensitivity-based approaches. This step not only substantiates the theoretical prowess but also extends the scope of potential applications, highlighting the method's relevance across diverse contexts.

By uniting these two distinct frameworks, a gateway emerges towards the realm of advanced algorithms, paving the way for heightened capabilities and unprecedented outcomes. In the realm of controllers, our proposal introduces a Sensitivity-aware Quadratic Program (QP) that holds the promise of not only enforcing rigorous hard-constraints as a safety filter but also reduce the drift of sensitivity over time, thus fortifying online robustness. Nevertheless, the potential of the SR-QP method remains untapped,and could be ascended to greater potency by encompassing multiple evolutionary time steps within the QP sensitivity, the same way Non-linear Model Predictive Control (NMPC) works. While the current options appear somewhat limited, delving into broader optimization problems might be a pathway to explore. Alternatively, innovative perspectives on the project could yet unveil opportunities to preserve the QP formulation. Another avenue could involve applying the SA-QP—designed for offline prediction within each time step, optimizing it iteratively, although such an approach might lack elegance and efficiency in real-time applications.

Turning to the SA-QP method, it proffers a mechanism for computing trajectory sensitivity utilizing a QP controller, alongside the incorporation of a judiciously conservative constraint rooted in the previous time-step's sensitivity. This innovation's impact reverberates profoundly in trajectory planning, mirroring earlier endeavors by enabling sensitivity optimization algorithms on the resultant trajectory without necessitating manual assurance of system constraints. An illustrative application could encompass Rapidly-exploring Random Trees (RRT) exploration algorithms, where this novel method crafts trajectories between points, meticulously integrating safety boundaries and furnishing sensitivity metrics for robustness optimization. Admittedly, this process might entail a certain time investment to yield the ultimate optimized trajectory. Nonetheless, as this unfolds offline, temporal considerations are less of a concern.

In essence, this amalgamation of frameworks unravels vistas of potential, whether amplifying controller capabilities or streamlining trajectory planning, this convergence epitomizes the perpetual pursuit of refining methodologies to unlock new dimensions of performance and innovation.

5. Conclusion

In the realm of research, a dynamic journey unfolds, weaving through theory, simulation, and application. This expedition begins with a deep dive into theoretical foundations, unearthing historical and contemporary insights. Balancing autonomy and collaboration, research thrives on team interactions and individual innovation.

Navigating theoretical intricacies, researchers grapple with mathematical challenges. Simulation, a linchpin, unveils compatibility, stability, and tuning complexities. Merging frameworks like Control Barrier Functions (CBF) and Sensitivity, simulations reveal the interplay between robustness and constraints.

The fruition of this work manifests in the development of two pioneering methods: the Sensitivity-Regulated Quadratic Program (SR-QP) for real-time applications and the Sensitivity-Augmented Quadratic Program (SA-QP) for offline planning. Both methodologies have been successfully realized during this internship, offering fresh perspectives that cast a transformative light on the future of safety-critical systems.

Results might not immediately satisfy, yet they illuminate pathways for improvement. The quest for innovation thrives on iterative processes, parameter adjustments, and refining methodologies. The intersection of theories not only broadens perspectives but also sparks avenues for novel solutions.

Looking ahead, the journey holds promise. Fine-tuning simulations for credibility and embracing real-world experiments stand as prospects. The synergy between frameworks births advanced algorithms. This ongoing narrative of research marries theory with practice, autonomy with collaboration, and aspiration with realization—navigating towards the horizon of innovation. Through each phase, from theory's inception to application's impact, we chart a course that advances knowledge and propels progress for safe and robust trajectory planning.

Bibliography

[1] P. Robuffo Giordano, Q. Delamare, and A. Franchi, "Trajectory generation for minimum closed-loop state sensitivity," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 286–293, 2018.

[2] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," 2019.

[3] P. Brault, Q. Delamare, and P. Robuffo Giordano, "Robust trajectory planning with parametric uncertainties," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11095–11101, 2021.

[4] P. Brault and P. R. Giordano, "Tube-based trajectory optimization for robots with parametric uncertainty."

[5] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[6] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert, "Robust control barrier-value functions for safety-critical control," 2021.

[7] S. Tonkens and S. Herbert, "Refining control barrier functions through hamilton-jacobi reachability," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 13355–13362, 2022.

[8] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference (ACC)*, pp. 322–328, 2016.

[9] B. W. Bader and T. G. Kolda, "Algorithm 862: Matlab tensor classes for fast algorithm prototyping," *ACM Trans. Math. Softw.*, vol. 32, p. 635–653, dec 2006.

[10] B. Amos and J. Z. Kolter, "Optnet: Differentiable optimization as a layer in neural networks," 2021.

[11] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se(3)," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 5420–5425, 2010.

[12] M. Khan, M. Zafar, and A. Chatterjee, "Barrier functions in cascaded controller: Safe quadrotor control," in *2020 American Control Conference (ACC)*, pp. 1737–1742, 2020.

[13] H. Zhang, X. Zhang, T. Li, S. Zhang, and X. Zhang, "Barrier function enhanced geometric controller for safe control of a quadrotor uav," in *2022 International Conference on Advanced Robotics and Mechatronics (ICARM)*, pp. 187–192, 2022.

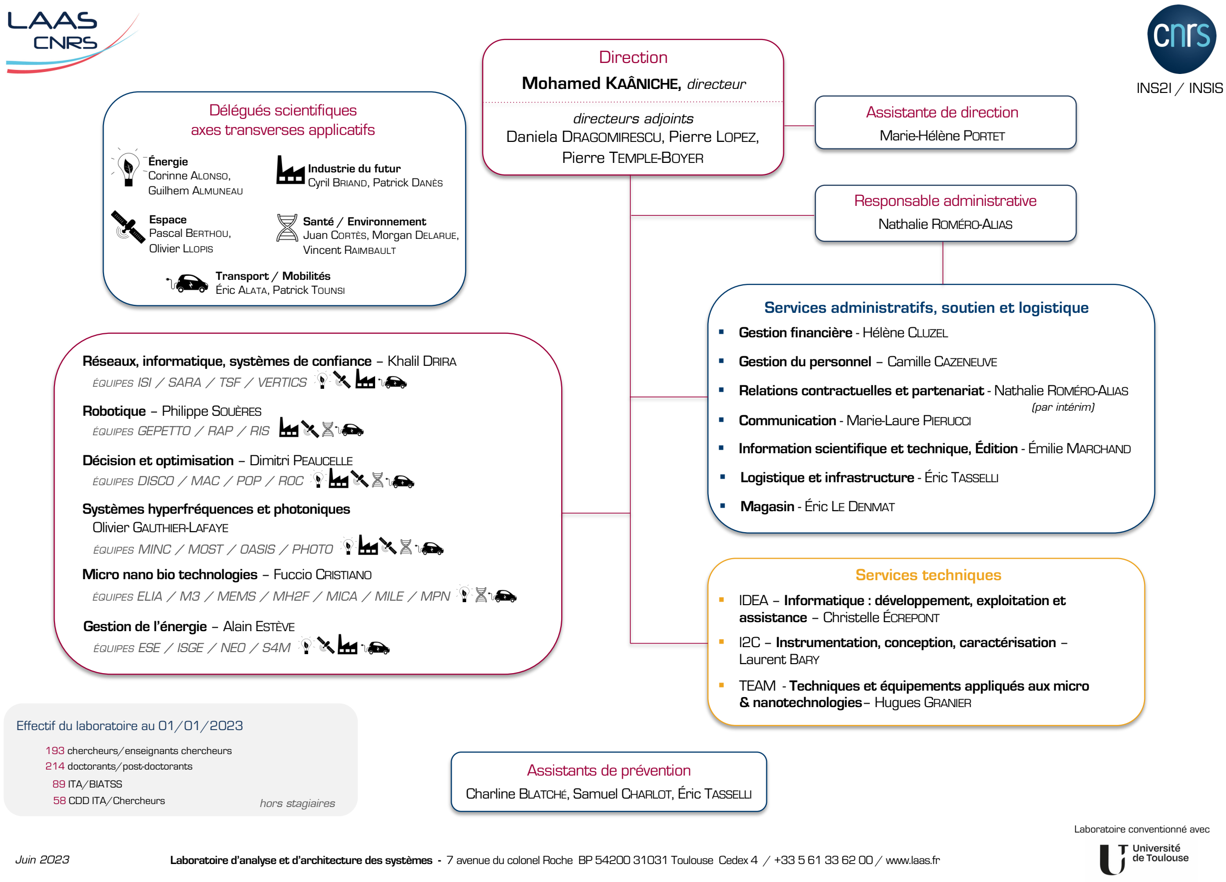
[14] V. Mistler, A. Benallegue, and N. M'Sirdi, "Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback," pp. 586 – 593, 02 2001.

[15] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics and Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[16] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

Appendices

A. Organisation chart of the LAAS-CNRS



B. Gantt Diagram

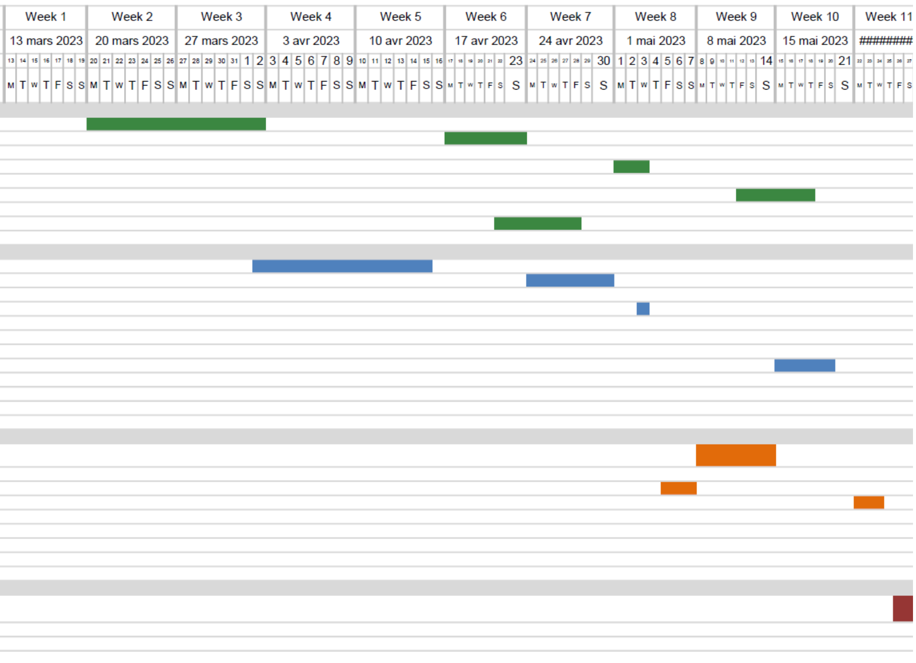
PFE Trajectory Planning

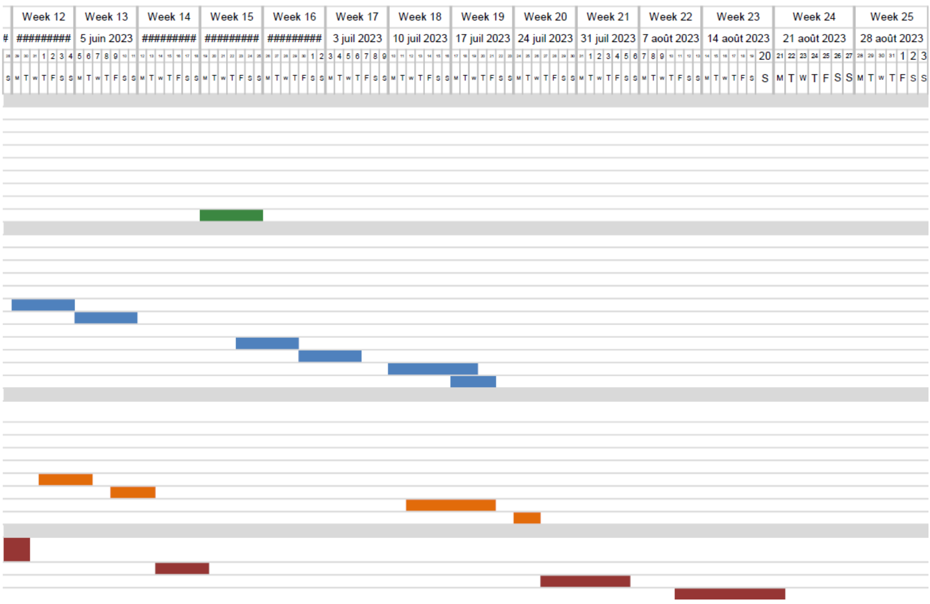
LAAS-CNRS

Project Start Date 3/15/2023 (mercredi)

Project Lead Jonas Soueidan

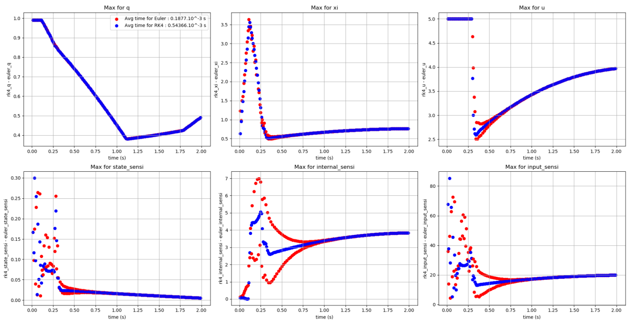
WBS	TASK	START	END
1	**Litterature and references**	-	-
1.1	Project references	lun 3/20/23	dim 4/02/23
1.2	Sensitivity and CBF	lun 4/17/23	dim 4/23/23
1.3	Modelisation	ven 5/01/20	jeu 5/07/20
1.3.1	Unicycle	lun 5/01/23	mer 5/03/23
1.3.2	Quadrotor	dim 2/04/18	sam 2/10/18
1.4	Tensor Multiplication and Commutation	ven 5/12/23	jeu 5/18/23
1.5	Optimisation Problems	sam 2/03/18	ven 2/09/18
1.5.1	Quadratic Program	sam 4/22/23	ven 4/28/23
1.5.2	KKT conditions	lun 6/19/23	dim 6/25/23
2	**Theory**	-	-
2.1	Exploring ideas	dim 4/02/23	sam 4/15/23
2.2	First Methods	lun 4/24/23	dim 4/30/23
2.3	Modelisation	mer 2/14/18	ven 2/16/18
2.3.1	Unicycle	mer 5/03/23	mer 5/03/23
2.3.2	Quadrotor	ven 5/29/20	jeu 6/04/20
2.3.2.1	Model	lun 5/29/23	dim 6/04/23
2.3.2.2	Control	lun 6/05/23	dim 6/11/23
2.4	Control Barrier Function	lun 5/15/23	sam 5/20/23
2.5	Exploring ideas	ven 6/23/23	jeu 6/29/23
2.6	Second method	ven 6/30/23	jeu 7/06/23
2.7	Updating Models	lun 7/10/23	mer 7/19/23
2.8	Finalising Equations	lun 7/17/23	ven 7/21/23
3	**Simulation**	-	
3.1	Simulation Environment	lun 5/08/23	dim 5/14/23
3.2	Unicycle Environment	mar 5/05/20	jeu 5/07/20
3.2.1	Evolution Dynamics	ven 5/05/23	dim 5/07/23
3.2.2	Symbolic Equations	lun 5/22/23	mer 5/24/23
3.3	Quadrotor	mer 2/14/18	ven 2/16/18
3.3.1	Evolution Dynamics	jeu 6/01/23	mar 6/06/23
3.3.2	Symbolic Equations	ven 6/09/23	mar 6/13/23
3.4	Updating Models	mer 7/12/23	ven 7/21/23
3.5	Symbolic Quadratic Program	lun 7/24/23	mer 7/26/23
4	**Others**	-	-
4.1	Experiments 1	ven 5/26/23	mar 5/30/23
4.2	Experiments 2	mer 6/14/23	lun 6/19/23
4.3	Experiments 3	jeu 7/27/23	sam 8/05/23
4.4	Report Writting	ven 8/11/23	lun 8/21/23



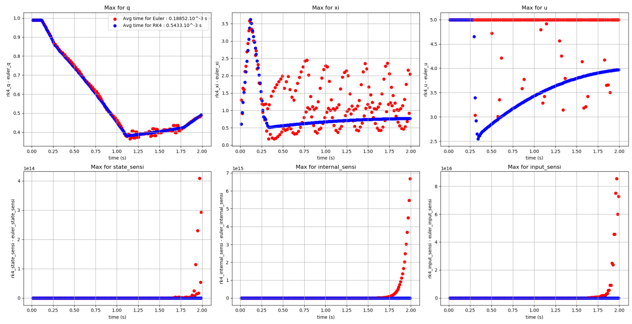


C. Figures for Euler/RK4

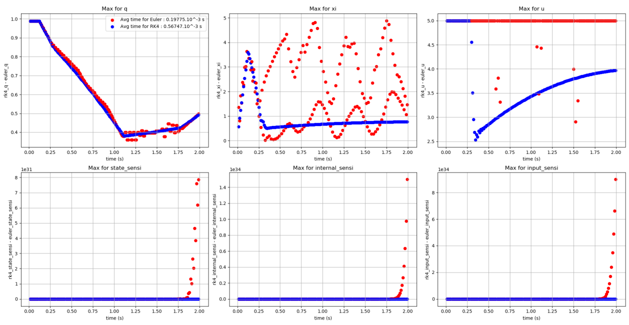
As shown in the Figures C.1 below, Euler and RK4 gives us two significant results : RK4 is more accurate, converge faster, and more stable with an increasing time step size, but is around 3 times slower, the average times are around $0.2*10^{-3}$ for Euler and $0.6*10^{-3}$ for RK4. For single iterations, the computer execution times are more than reasonable, but for optimisation problems, that could be an issue. The model used was a unicycle, the details can be found in the section 3.3.1. Using a quadrotor model, that is more complex, will be much more time expensive, and it will be even worse for an hexarotor model.



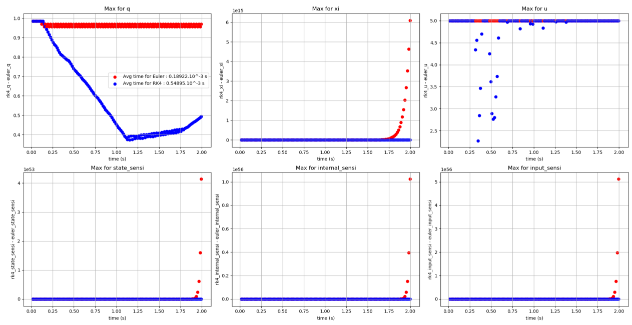
(a) $dt = 0.010$



(b) $dt = 0.011$



(c) $dt = 0.012$



(d) $dt = 0.015$

Figure C.1.: Comparison Euler RK4 for various Integration time steps. Blue is RK4, red is Euler.

D. Tensor Product Commutation

Tensor Product Commutation Theorem

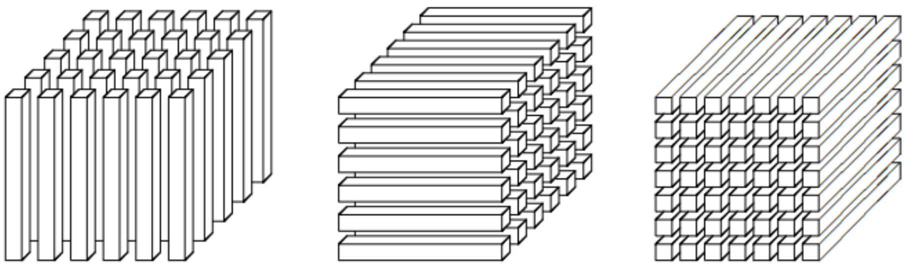
Using the *n-mode* product for tensor multiplication with Matrices and Vectors, if the modes are distinct, the operation are commutative (see [9] and [16] for reference on tensor multiplication)

We have a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_q \times n_u \times n_q}$, a vector $v \in \mathbb{R}^{n_u}$ and a matrix $U \in \mathbb{R}^{n_q \times n_p}$. Proceeding element-wise *n-mode* operations, we get :

$$
\begin{aligned}
&\forall i \in [\![0,n_q]\!], \forall j \in [\![0,n_q]\!], \forall k \in [\![0,n_p]\!] : \\
&(\boldsymbol{\mathcal{X}} \bullet_{n_u} v)_{(i,j)} = \textstyle\sum_{l=1}^{n_u} x_{(i,l,j)} \cdot v_{(l)} \\
\Rightarrow \quad & [(\boldsymbol{\mathcal{X}} \bullet_{n_u} v) \times_{n_q} U]_{(i,k)} = \textstyle\sum_{m=1}^{n_q} (\sum_{l=1}^{n_u} x_{(i,l,m)} \cdot v_{(l)}) \cdot u_{(m,k)} \\
&= \textstyle\sum_{m=1}^{n_q} \sum_{l=1}^{n_u} x_{(i,l,m)} \cdot v_{(l)} \cdot u_{(m,k)} \\
&= \textstyle\sum_{l=1}^{n_u} \sum_{m=1}^{n_q} x_{(i,l,m)} \cdot u_{(m,k)} \cdot v_{(l)} \\
&= \textstyle\sum_{l=1}^{n_u} (\sum_{m=1}^{n_q} x_{(i,l,m)} \cdot u_{(m,k)}) \cdot v_{(l)} \\
&[(\boldsymbol{\mathcal{X}} \bullet_{n_u} v) \times_{n_q} U]_{(i,k)} = [(\boldsymbol{\mathcal{X}} \times_{n_q} U) \bullet_{n_u} v]_{(i,k)}
\end{aligned}
$$

Which conclude our proof.

As an easier way to comprehend, we are performing operation on *mode-n fibers*, which can be seen as the higher order rows and columns, obtained by fixing every index but one, see Figure D.1. Matrix multiplication with a tensor can be seen as multiplying each given fiber by the matrix, and vector multiplication is the outer product between the fiber and the given vector. It becomes quite clear that when the modes, i.e. the fibers, are different, order of operation does not matter ([9]).



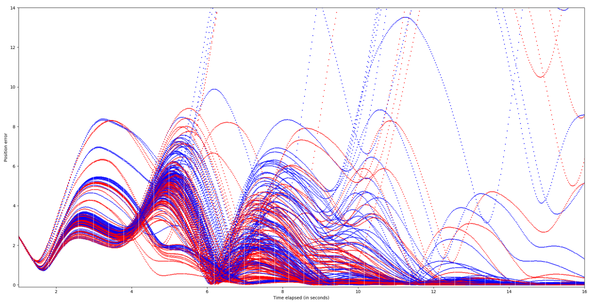
(a) Mode-1 (column) fibers: $\mathbf{x}_{:jk}$

(b) Mode-2 (row) fibers: $\mathbf{x}_{i:k}$

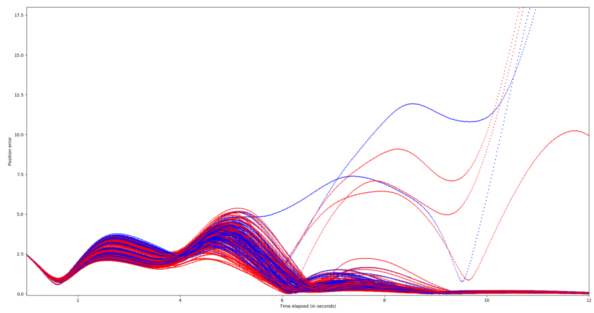
(c) Mode-3 (tube) fibers: $\mathbf{x}_{ij:}$

Figure D.1.: Fibers of a 3rd-order tensor

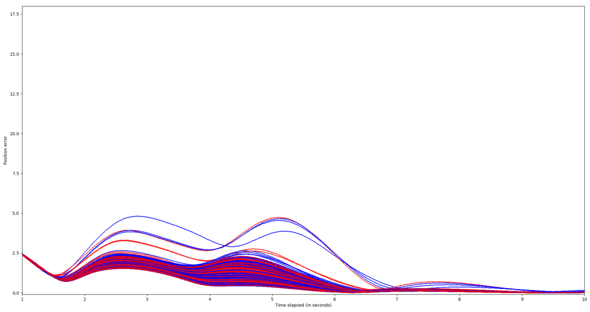
E. SR-QP Method Results



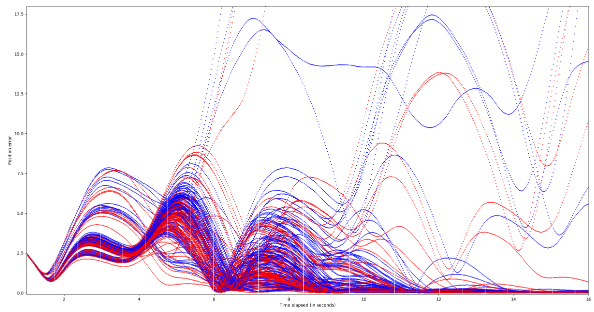
(a) $40Hz$



(b) $60Hz$



(c) $80Hz$



(d) $100Hz$

Figure E.1.: Comparison on a Quadrotor for different integration frequencies. In blue with the standard controller, and in red with the SR-QP. Each figures contains 100 simulations, for 50 different sets of internal parameters.

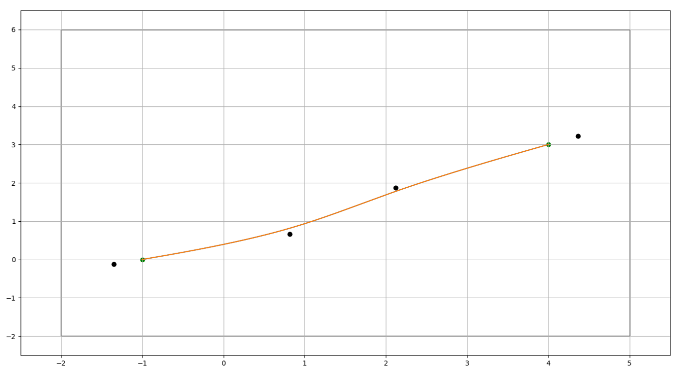
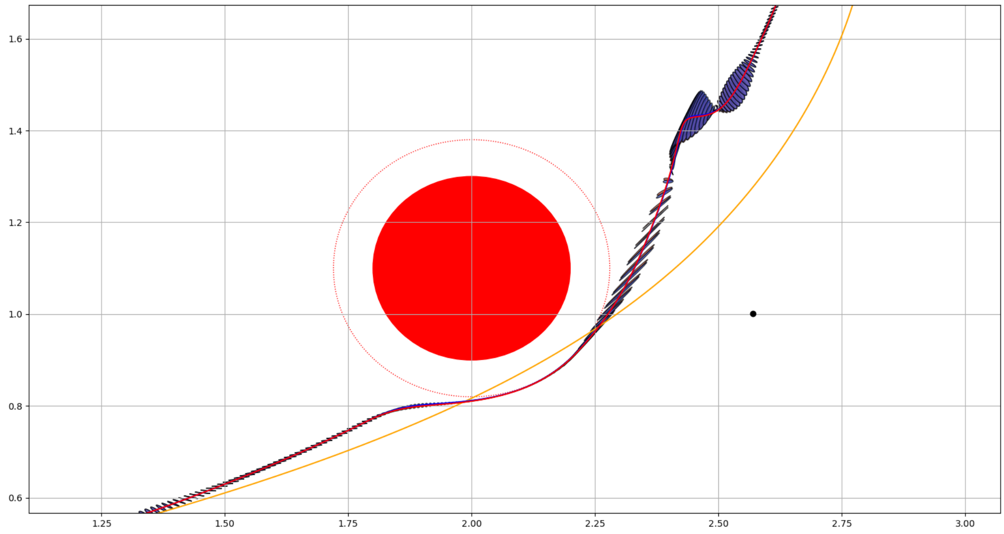


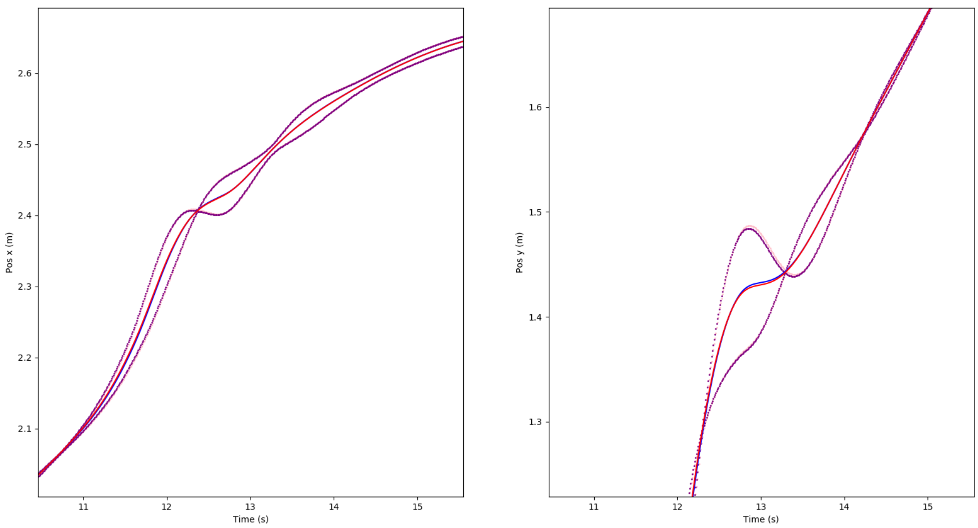
Figure E.2.: Unicycle trajectory (x,y) tracking with and without SR-QP

As we can see on figure E.2, the tracking is done identically both for the standard model and the SR-QP method.

F. SA-QP Method Results

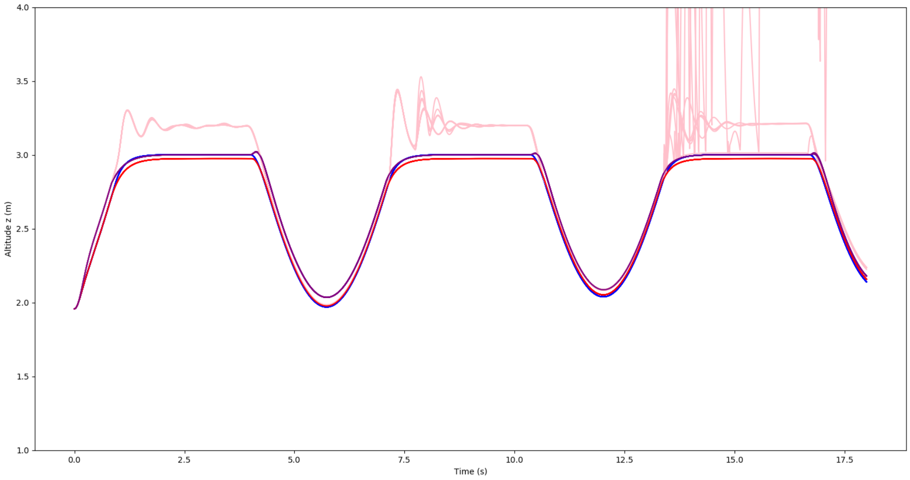


Trajectory, obstacle and state sensitivity ellipsoids

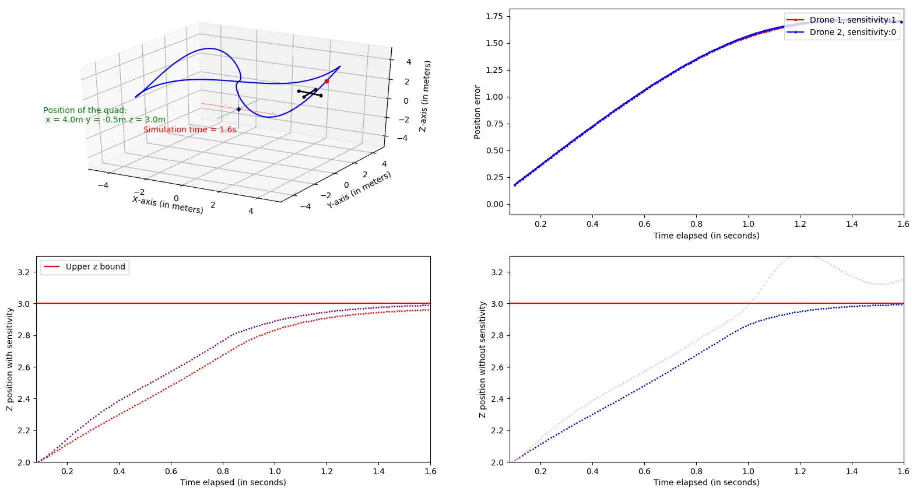


Positions $x$and y with respective deviations

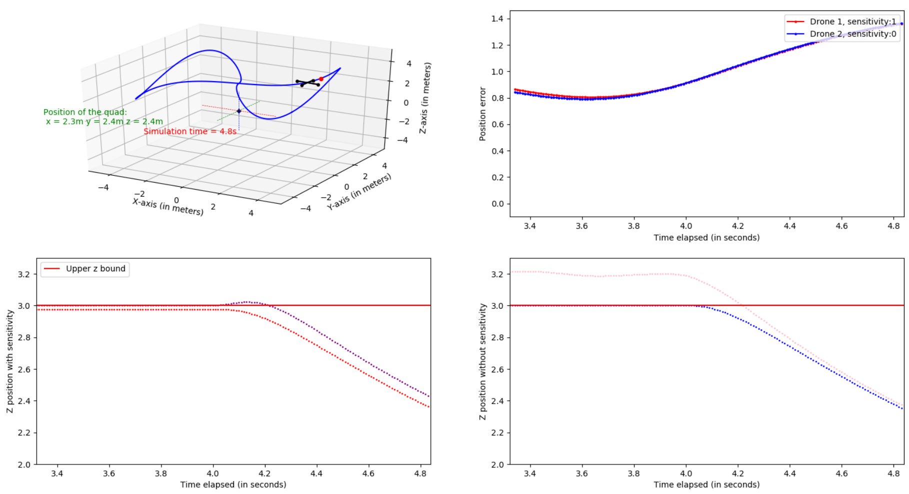
(a) Unicycle simulation



Quadrotors altitudes and sensitivity deviations



Trajectories at $t = 1.6s$



Trajectories at $t = 4.8s$

(b) Quadrotor simulation

Figure F.1.: Positions are blue for the standard controller, red for the sensitivity aware controller, sensitivity deviations are pink for the standard controller and purple for the sensitivity aware controller.