Temporal models of motions and forces for Human-Robot Interactive manipulation

Kevin Desormeaux

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *25/10/2019* **par :**

KEVIN DESORMEAUX

Temporal models of motions and forces for Human-Robot Interactive manipulation.

JURY

RACHID ALAMI	Directeur de recherche, LAAS-CNRS	Président du Jury
PHILIPPE FRAISSE	Professeur à l'Université de Montpellier, LIRMM	Rapporteur
YOUCEF MEZOUAR	Professeur à SIGMA Clermont, Institut Pascal	Rapporteur
AURÉLIE CLODIC	Ingénieur de Recherche, LAAS-CNRS	Examinatrice
CHRISTOPHE GUETTIER	Docteur, Safran Electronics & Defense	Examinateur
DANIEL SIDOBRE	Maître de conférence à l'Université Paul Sabatier, LAAS-CNRS	Directeur de Thèse

---

École doctorale et spécialité :
EDSYS : Robotique 4200046
Double mention :
EDSYS : Informatique 4200018
Unité de Recherche :
Laboratoire d'analyse et d'architecture des systèmes
Directeur de Thèse :
Daniel SIDOBRE
Rapporteurs :
Philippe FRAISSE et *Youcef MEZOUAR*

Acknowledgments

Je tiens tout d'abord à remercier l'ensemble des membres du jury, à commencer par Philippe Fraisse et Youcef Mezouar pour avoir accepté d'être mes rapporteurs ainsi que pour l'attention qu'ils ont portée à ce manuscrit et la qualité de leurs critiques. Je remercie également Christophe Guettier de Safran Electronics & Defense, Aurélie Clodic et Rachid Alami pour leur participation au jury en tant qu'examinateurs. Merci encore à Rachid, président du jury, mais aussi directeur de l'équipe Robotique et InteractionS pour m'avoir accepté dans son équipe.

Les travaux réalisés dans le cadre de cette thèse ont été effectués au sein du Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS, à Toulouse. J'exprime toute ma reconnaissance à Daniel Sidobre, mon directeur de thèse, pour m'avoir offert cette opportunité. Je mesure la chance que j'ai eu d'avoir eu un encadrant capable d'autant de patience et disposant d'une grande qualité d'écoute. Les nombreuses heures qu'il a pu me consacrer, et la qualité de ses conseils, ont été le facteur déterminant dans la réussite de cette thèse.

Je tiens également à remercier tous les membres de l'équipe RIS qui ont grandement contribué à construire un environnement de travail agréable et je mesure de manière plus générale la chance que j'ai eu de pouvoir évoluer dans ce laboratoire. On sait tous qu'être doctorant c'est être confronté à beaucoup de stress et de pression, et travailler ici a de manière certaine contribué à alléger ce fardeau. N'étant pas du genre à étaler publiquement mes sentiments, je préfère remercier ici encore une fois toutes les personnes avec qui j'ai eu la chance d'évoluer sans les nommer.

Il convient toutefois d'apporter toute ma reconnaissance à nos ingénieurs Matthieu Herrb et Anthony Mallet, dont on ne louera jamais assez leur importance. Sans eux je serai toujours coincé avec des problèmes de code et de software sans aucun espoir de m'en sortir. Merci également à Xavier Dollat de l'atelier mécanique qui a pu confectionné de nombreuses pièces pour la réalisation de mes expérience, ainsi que pour les conseils liés à l'impression 3D, qui m'ont d'ailleurs aidé à franchir le pas, étant désormais l'heureux détenteur d'une de ces imprimantes. Remerciements également à Liviu Nicu, directeur du LAAS au cours de ces dernières années, pour tout son travail à la tête du laboratoire.

Pour terminer merci à mes amis et à ma famille. Je dédie cette thèse à ma mère que je n'oublie pas.

Contents

List of Figures

List of Tables

List of Multimedia

[**video-1**] *Online Trajectory Generation: Reactive Control With Return Inside an Admissible Kinematic Domain*. https://youtu.be/7L-y168fg9g. (cited on pages 95 and 101).
[**video-2**] *Reactive Trajectory Control: Online Trajectory Switching*. https://youtu.be/0fuZNY1N5Q8. (cited on pages 98 and 102).
[**video-3**] *Fully Autonomous Door Opening at LAAS-CNRS*. https://youtu.be/_mOtfrssyuY. (cited on page 119).

Introduction

Contents

0.1 Context

The first robot was invented in the 1950s by George Devol who founded together with Joseph Engelberger the world's first robot manufacturing company, Unimation. This robot, called Unimate, was sold to General Motors and replaced workers in dangerous tasks (Fig. 1).

It was in the 70s when the interest for robotics really emerged. It was barely half a century ago, and since then robots have been replacing humans in the industry. They are mainly use in tasks that are repetitive or considered too dangerous for humans. Robotics significantly increased the production capacities of industries while significantly reduced the costs related to human workers. The widespread use of industrial robots in the manufacturing world marked the beginning of an era of industrial automation. In 2015, more than 1.64 million industrial robots were in operation worldwide according to International Federation of Robotics (IFR).

However, this era of industrial automation is recently undergoing changes. In the industry of the past decades robots were confined in cages with prohibited access to humans. These robots were not capable of dealing with the safety risks related to the human presence. With the recent advances made in the field of Artificial Intelligence (AI), sensors, computer sciences and more, the collaboration between humans and robots is becoming possible. The close cooperation of humans and machines is motivated by the increased need for flexibility, adaptability and reusability of assembly systems. The main motivation being to get the best of both worlds, which is allowing humans and robots to combine their respective strengths.

Industry is not the only domain that benefits from the growth of robotics. Another kind of robots called social robots is making its appearance.

In both cases these robots will have to adapt to human presence. They will have to ensure the physical safety of humans but also their psychological safety by providing interactions with sufficient ergonomics.

In the context of this thesis, whose main objective is Online Trajectory Generation (OTG) for Human-Robot Interactions (HRI), we will have to find models of motion that satisfy the needs for efficient and flexible robots while ensuring the physical and psychological safety of humans.

Figure 1: Unimate, the grandfather of industrial robots.

0.2 Contributions

The work presented in this manuscript is focused on trajectory generation, mainly for Human-Robot Interactions.

The main contribution of this thesis is the extension of the softMotion library, a library for trajectory generation. The algorithm behind this library builds smooth cubic polynomial trajectories adapted for the context of Human-Robot Interactions. To our knowledge, it is the first complete algorithm that satisfies simultaneously all the following criteria to generate safe, efficient, adaptable and human-friendly motions:

- Real-time capable.
- General initial and end conditions for both velocity and acceleration.
- General bounded jerk, acceleration and velocity.
- Asymmetric bounds.
- Time optimal.

A second contribution is the extension of the previous algorithm to cope with non-admissible initial configurations, opening the way to trajectory generation under non-constant motion constraints. This feature is essential in the context of physical Human-Robot Interactions, as the robot must adapt its behavior in real time to preserve both the physical and psychological safety of humans. We have

also developed an architecture for robot control, based on reactive trajectory control, which has been designed and demonstrated for real-world applications. Finally we investigated the role of kinematics in the definition of ergonomics properties of motions. This work is intended to determine the input parameters of the previous controllers in order to take into account the ergonomic constraints associated to the human's presence.

This thesis was an opportunity for us to validate our software within a research contract in collaboration with the Safran group. The door opening demonstration presented in chapter 4 reproduces elements of this project.

0.3 Publications

Some of the contributions of this thesis have been published in a journal and in an international conference:

- **Kevin Desormeaux**, Daniel Sidobre. *Online Trajectory Generation: Reactive Control With Return Inside an Admissible Kinematic Domain.* IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 4-8 nov. 2019. *(Accepted)*

- Daniel Sidobre, **Kevin Desormeaux**. *Smooth Cubic Polynomial Trajectories for Human-Robot Interactions.* Journal of Intelligent & Robotic Systems, vol. 95, no. 3, pages 851–869, September 2019. https://doi.org/10.1007/s10846-018-0936-z.

Other publications:

- **Kevin Desormeaux**, Daniel Sidobre. *Smooth Cubic Polynomial Trajectories for Human-Robot Interactions.* Journée des Jeunes Chercheurs en Robotique (JJCR), 07 november 2017.

- **Kevin Desormeaux**, Daniel Sidobre. *Modèles temporels du mouvement et des forces d'interaction pour la manipulation interactive avec un robot.* Congrès des doctorants EDSYS 2017.

0.4 Manuscript organization

The manuscript outline is detailed hereafter:

Chapter 1 presents an analysis on the context of tomorrow's robotics. We reuse this state of the art to replace our work on trajectory generation inside this context.

Chapter 2 introduces a complete mono-dimensional trajectory generation algorithm that build trajectories from arbitrary initial and final conditions, subject to general asymmetric bounds on jerk, acceleration and velocity. We identify and address the problem the time-optimal curve regarding the length of the motion, which can be non-linear and exhibit discontinuities. The extension of this algorithm to the multi-dimensional case is presented at the end of the chapter.

Chapter 3 completes the algorithm presented in the previous chapter so that it can cope with inadmissible initial configurations. This extension opens the way towards trajectory generation with non-constant motion constraints, an essential feature for physical Human-Robot Interactions. In a second part we present how to build an efficient control architecture as well as a strategy for Reactive Trajectory Control. The chapter is concluded by experimental results.

Chapter 4 introduces briefly some experimental work, notably a user study which purpose is to investigate the impact of kinematics on the ergonomic properties of motion. This work aims towards better Human-Robot Interactions and is in adequacy with the research presented in the previous chapters.

Chapter 5 concludes the work accomplished during this thesis, and opens some perspectives.

Chapter 1

State of the art in motion generation

Contents

1.1 Context

1.1.1 What's a robot?

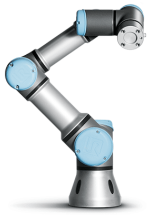
So what's a robot? Even if the answer might appears obvious, it is not. Robots are very diverse in regards to size, capacities, and design. There is no definition on which there is consensus, and there are still debates among the robotic community. So first lets take a general definition that is widely accepted: "A robot is an autonomous machine capable of sensing its environment, carrying out computations to make decisions, and performing actions in the real world[1]."

Now using this definition it is easy to find out why it is still subject to debate. When a person is asked to think of a robot, chances are she remembers the last viral YouTube video of the latest Boston Dynamic robot, such as Atlas (Fig. 1.1b).

[1]source: https://robots.ieee.org/learn/

(a) The famous Nao robot of SoftBank Robotics.

(b) Atlas of Boston Dynamics.



(c) A robotic arm of Universal.

Figure 1.1: Examples of popular robots. Nao is one of the most sold robot worldwide. Atlas is famous for being one of the most advanced humanoid robot. Universal is a recognized brand producing robotic arms.

Or maybe an industrial assembly line with many robotic arms. It is also true that our representation of a robot is largely influenced by science fiction. Now if you ask this person if a smoke detector can be considered as a robot, she will probably laugh, and this question might appear absurd to the majority. However a smoke detector is a robot if we use the previous definition. It can sense the presence of smoke and produces an alarm alerting humans in the vicinity. If it is acceptable to label a smoke detector, a rice-cooker, or a thermostat as simple robots this idea is disturbing. When asked to define a robot, robotics pioneer Joseph Engelberger once said, “I don’t know how to define one, but I know one when I see one!”.

What a Humanoid legged robot, a robotic arm or an autonomous vacuum have in common that differentiates them from a dishwasher, or a thermostat? It can be the level of sophistication. However this could be difficult to evaluate. A more visible marker is the ability to affect the world by moving in it or by manipulating things. Accepting this, we propose an updated definition: "A robot is an autonomous machine capable of sensing its environment, carrying out computations to make decisions, and performing actions altering the real world by producing motions." This definition is in line with the one proposed by the International Standard of Organization, which defines a robot as a reprogrammable, multifunctional manipulator designed to move material, parts, tools or specialized devices through variable programmed motions for performance of a variety of tasks.

This definition introduces the subject of this thesis: motion generation for robots. More precisely we are interested in motion generation for robots capable of interacting with humans. In the continuation of this state of the art, we explore the state of robotics, the expectations towards the future generation of robots and

the issues related to HRI. Finally we will see how the field of motion generation can contribute to the emergence of more sophisticated generations of robots, suitable for Human-Robot Interactions.

1.1.2 State of Robotic

Robotics is an amalgamation of several disciplines, from mathematics and computer science to electronics, computer vision, sensors and many more. As a result, its evolution depends on the progresses made in each of these disciplines. The constant developments made in those fields lead to the emergence of new generations of robots that are always more sophisticated. Robots are becoming cheaper, more efficient, robust, flexible and easy to use. From an era of heavy intimidating automatic machines, we are shifting towards safer and more friendly mechatronic systems. Robotics is no longer part of a distant future, and robots such as the educational Cozmo (Fig. 1.2e), or dust cleaners (Fig. 1.2b) are already popular (Fig. 1.2). Industrial automation is nothing new. The farming community already benefits from the addition of robots. Advanced robots are used to complete tasks which otherwise would put Humans lives at risk such as exploration in hostile environments. Nevertheless we are not yet able to see advanced robots in our everyday life, and the possibilities to interact with such robots are limited. The reality is that there remain huge challenges ahead for robotics, and practical home robots are still many years away.

So the question is : what are the difficulties restraining the development of robotics? It is nothing new actually. Robotics components are still costly. Robotics is still in its early days, and its arrival raises questions. Some are legitimate, such as the fear of job destruction. There are also the ethical questions behind the use made of these advances in robotics. Will we see one day killer robots? Fact is that society is not yet fully prepared to accept the major changes coming with the robotic revolution, but acceptance is on the way.

However these are not the only reasons slowing down the growth of robotics. **A more practical reason that kept robots limited to factories and research labs remains today : it is the addition of the Human presence**. This addition can simply be the result of humans and robots evolving in common places. Sometimes it will be the need for humans and robots to collaborate.

1.1.3 Towards the future of Industry

Two decades ago, the idea was suggested that the sequence of technological revolutions was not over, leading to a new universal technological revolution. The Second Machine Age is a term adopted by [Brynjolfsson 2014]. However the more popular term of Industry 4.0 was adopted after Germany began promoting its industrial development plan. The phrase fourth industrial revolution was first introduced by [Schwab 2016] at the 2015 World Economic Forum. In 2019, at the World Economic Forum meeting in Davos, Japan promoted another round of advancements called

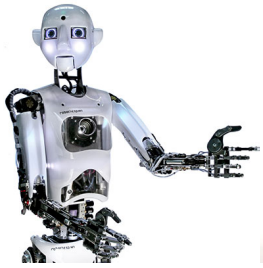

(a) Aerospace


(b) Consumer


(c) Disaster Response


(d) Drones


(e) Education


(f) Entertainment

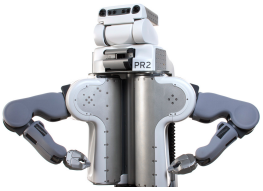

(g) Exoskeletons


(h) Humanoids


(i) Industrial


(j) Medical


(k) Military


(l) Research


(m) Autonomous cars


(n) Telepresence

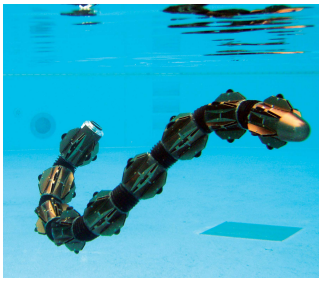
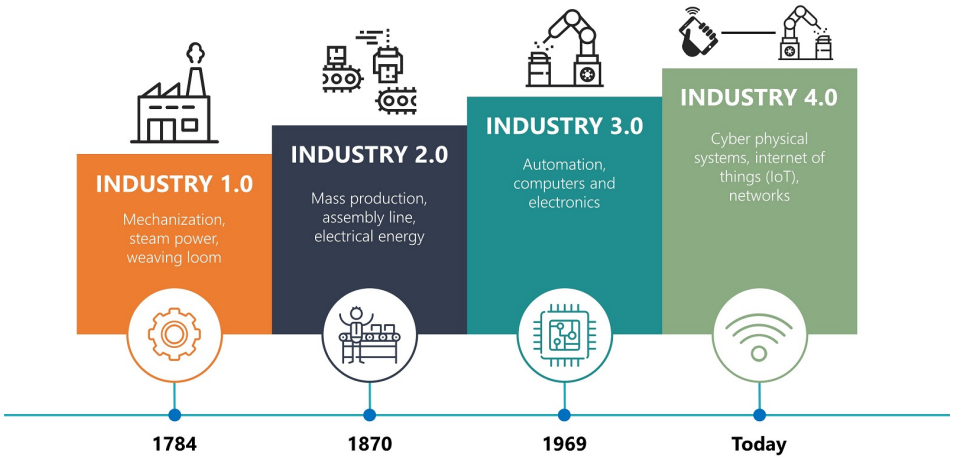
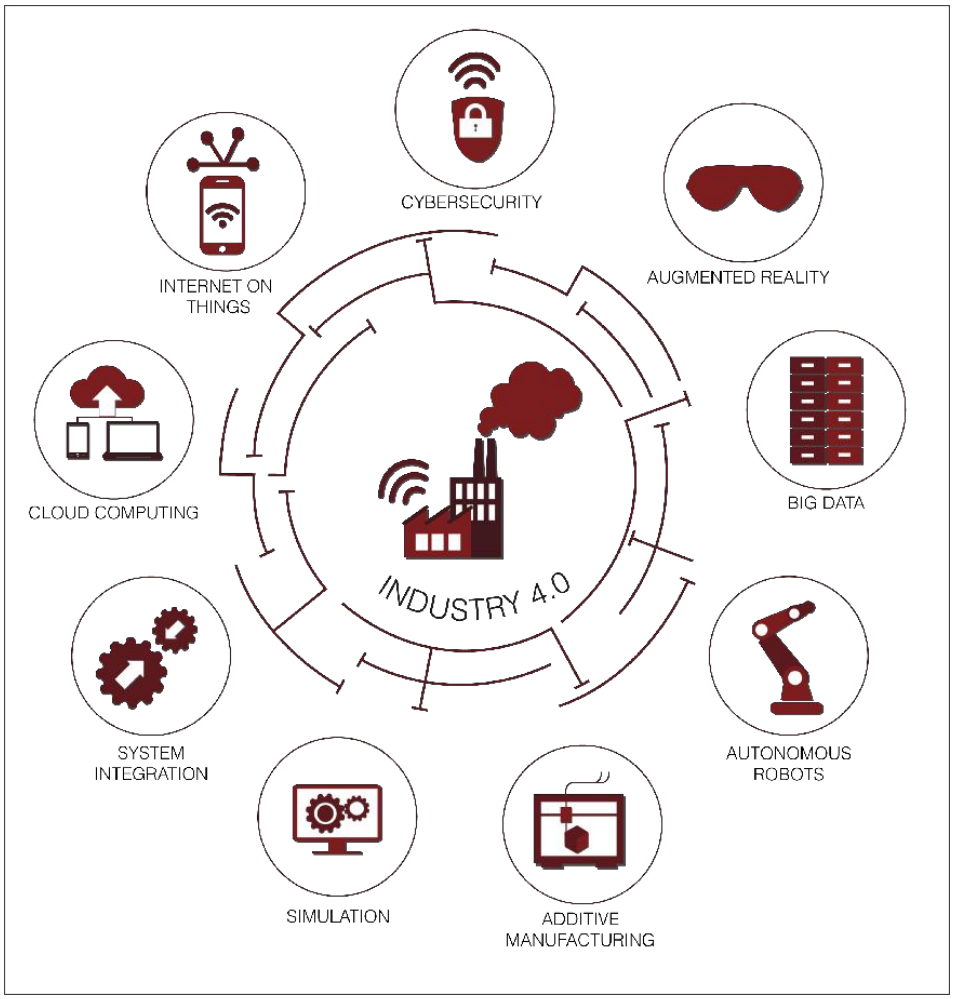

(o) Underwater

Figure 1.2: A classification of robots proposed by IEEE



(a) Evolution of the industry.



(b) Components of the Industry 4.0.

Figure 1.3: The Industry 4.0 also referred as the fourth industrial revolution.

Society 5.0.

So it is admitted that we are on the verge of a new industrial era, shifting from the third one to a fourth one (see Fig. 1.3a). In this new era, industries are often referred as smart factories. Machines are augmented with web connectivity and monitored by a system that can visualize the entire production chain and make decisions on its own. The trend is towards automation and data exchange in manufacturing technologies.

Some emerging fields are robotics, artificial intelligence (AI), nanotechnology, quantum computing, biotechnology, the (Industrial) Internet of Things, fifth-generation wireless technologies (5G), additive manufacturing, fully autonomous vehicles and more (Fig. 1.3b). Thanks to its multidisciplinary aspect, robotics will benefit from emerging technologies it is sure that robotics will be a major component of that industrial revolution. Now these are the questions we have to answer: **What is expected from those future generations of robots? What will be their role? What are the scientific challenges to be addressed?**

1.2 Human-Robot Interaction

1.2.1 Motivations

1.2.1.1 Robotics in Smart Factories

In the industry of the last decades or Industry 3.0 (Fig. 1.3a), and still to this day, industrial robots are usually confined in cages with prohibited access to Humans (Fig. 1.4). They accomplish repetitive tasks in a perfectly controlled environment. They cannot adapt to changes that would require an external intervention from a Human to reprogram them. Most of the time they are only composed of a robotic arm, and have a limited workspace. They are imposing, hard to manoeuvre, and not versatile. Reprogramming them requires expertise and is extremely time-consuming (Fig. 1.7). They are often used in assembly lines, and the car industry offer good examples of such environment (Fig. 1.4). These robots are vastly used in the industry because we can control their environment and remove or avoid most difficulties. This model for robots becomes obsolete as soon as the environment cannot be controlled. In the IEEE classification of robots, figuring fifteen categories (Fig. 1.2), the latter industrial model is obsolete in most of them. It is notably the case for the robotic of service and social robots due to the presence of Humans.

A trend of the industry of the future will be interconnected systems and automation. Thanks to the progress made in AI, Industrial Internet of Things and many other, machines will be provided with more autonomy. At the same time, a second trend is making its appearance, that is Human-Robot Collaboration (HRC).

The close cooperation of humans and machines in hybrid assembly is motivated by the increased need for flexibility, adaptability and reusability of assembly systems. Future production systems will be characterized by individualized products under the conditions of a highly flexible mass production. Thus, new solutions for

Figure 1.4: A typical assembly line composed of robotic arms confined in cages.

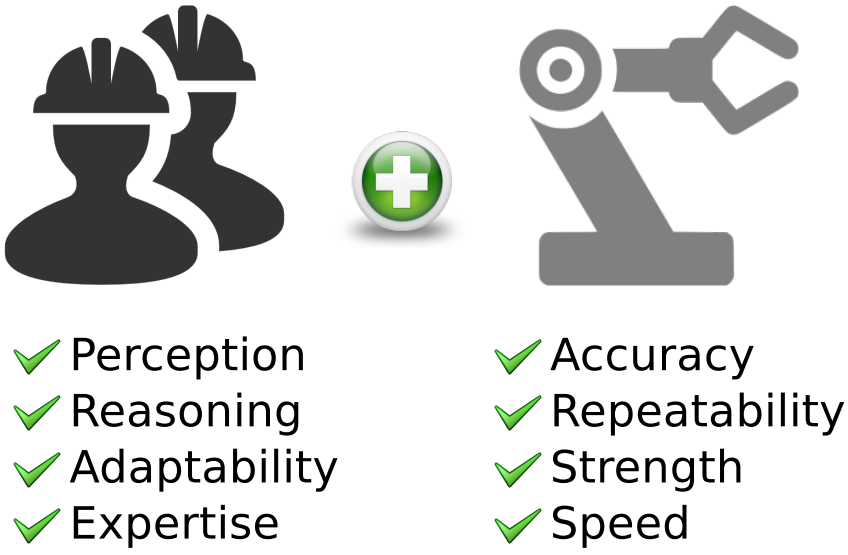


Figure 1.5: Human-Robot Cooperation combines the respective strength of the Human and the robot.

increased flexibility and inter-operability, such as flexible robotic equipment and intelligent decision making software platforms, must be investigated. To this end,

"Classical" industrial robot	**Future production assistant**
fixed installation	flexibly relocatable (manually or on mobile robots)
periodic, repeatable tasks; seldom changes	frequent task changes; tasks seldom repeated
programmed online / offline by a robot specialist	instructed online by a process expert supported by offline methods
infrequent interaction with the worker only during programming	frequent interaction with the worker, even force / precision assistance
worker and robot separated by fences	workspace sharing with the worker
profitable only with medium to large lot sizes	profitable even with small lot sizes

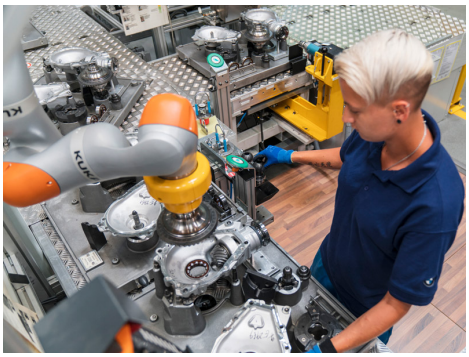
Figure 1.6: Comparison between classical industrial robots and future production assistants from [Bischoff].

robots should be quickly and intuitively operated by humans, while guaranteeing a safe close interaction [Villani 2018].

In that regard, previous generations of robots will become obsolete in many situations. The industry of the future, often mentioned as industry 4.0, will see robots and humans cooperate in order to combine their respective strengths. The main reason behind the emergence of cobots is to get the best of both worlds (see Fig. 1.5). HRC allows to combine human's cognitive abilities and adaptability with the accuracy, repeatability and strength of robots. In these new forms of cooperation robots are designated as cobots for collaborative or cooperative robots. Cobots are smaller, more flexible, and often cheaper to deploy. Focus is made on a easy to use aspect. With these qualities small and medium-sized enterprises should benefit from this new form of automation. The cobot market is expected to know a rapid growth. In a paper dedicated to the Kuka Light Weight Robot (LWR), [Bischoff] offer a similar comparison between the classical industrial robots and the cobots (see Fig. 1.6).

In a study on safety for pHRI, [Bicchi 2008] reviews the advantages presented by such cooperation: "Humans have poor open-loop accuracy, tire easily, and are subject to repetitive stress injuries. In contrast robots have minimal sensing, not for lack of sensors but for their inability to interpret sensory input. However robots have high accuracy and speed and work indefinitely. Even with increasing sensorization, robots are not about to match a human's literally millions of sensory receptors". Indeed tasks that are easy for us Humans, can be very challenging or

(a) A universal cobot used in car assembly production line.



(b) LBR iiwa of KUKA cooperating with BMW employee in Dingolfing's factory.

Figure 1.7: Illustrations of cobots. The industry of the future?

yet impossible for automated systems. For example in [Russakovsky 2015], Humans inputs are used to extract information from pictures. [Michalos 2014] promotes a hybrid solution that combines the respective strengths of Humans and automated systems. The simulation experiments are promising, as they indicate significant savings in terms of productivity and operator's working conditions. A survey on human-machine cooperation in assembly is proposed in [Krüger 2009]. The authors study the forms of cooperation between a human and a robot that can be used in assembly processes as well as the organizational and economical aspects. They describe the advantages of combining the respective strengths of a human and a robot, and advocate in favour of these new kinds of cooperation, even compared to fully automated systems. These robots will have to be adaptable, flexible, and reusable. [Villani 2018] proposes an in depth review of industrial HRC. The authors offer a comparison between fully automated systems and hybrid human-robot solution. Challenges inherent to human-robot collaboration are addressed, such as safety and intuitive ways to program and interact with robots. In place of considering safety as a requirement that limits performance, they propose to inspect **performance oriented solutions**, meaning that **performances should be optimized subject to the constraint of safety**. The paper reviews the different applications where collaborative robots have been used and sort out the advantages presented.

1.2.2 Cobots

The term of cobot was first introduced in [Colgate 1996]. A cobot is a robotic device, which manipulates objects in collaboration with a human operator. In its original definition a cobot's task was to offer guidance to the human's motion. This guidance was provided by using virtual surfaces to constrain the motion. To ensure safety the cobot did not provide motive power, which was given by the human. Since then its meaning has evolved to become a robot that physically interacts with Humans in a shared workspace and in a safe manner. They can also provide motive

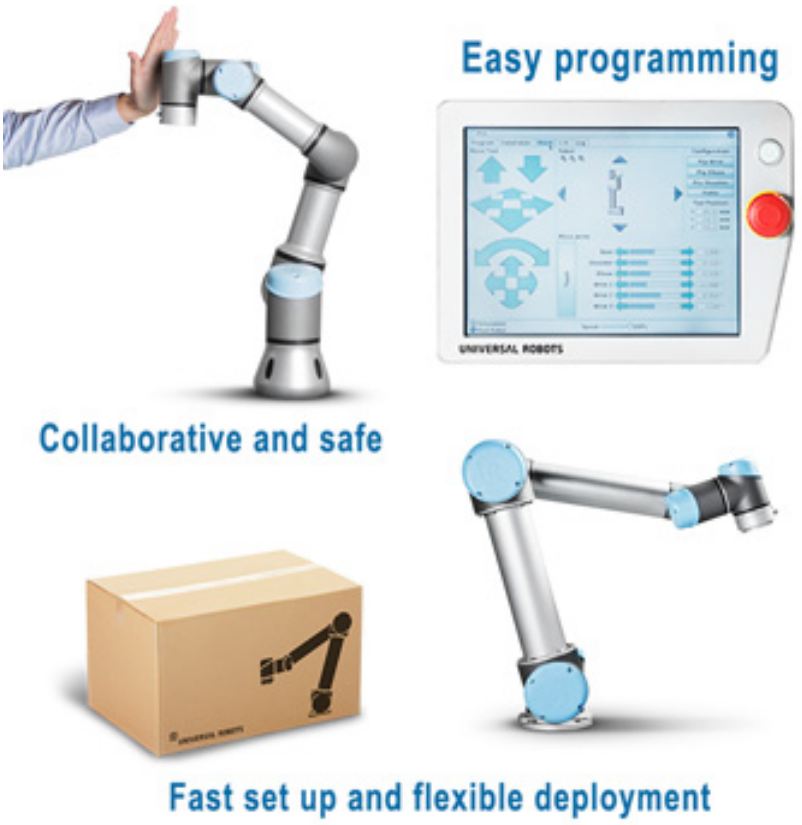


Figure 1.8: Advantages of cobots illustrated by universal.

power. The latter definition includes a large panel of robots. From medical robots and exoskeletons, to social robots, through collaborative industrial robots that are able to work in cooperation with humans. However we will mainly focus on two types of cobots: industrial and social cobots.

1.2.2.1 Industrial Cobots

Industrial cobots have varying degrees of autonomy, diverse designs and can fulfil a wide variety of tasks.

We list here the main expectations towards a cobot:

- Act safely,
- Pleasant to work with,
- Reduce effort and stress on Human operator,
- Flexible,
- Intuitive.

As they have to share workspace with humans safety is the main requirement. Naturally a lot of works have been conducted in that regard (Sec.1.2.3). Their design is an example as they are smaller, lighter, and with rounded shape (Fig. 1.7a).

They mainly fulfil two objectives. The first one is to give more flexibility to industries, and to compensate the weakness of fully automated systems by integrating humans in the loop (Sec.1.2.1.1). The second one is to reduce arduousness at work.

To accomplish these objectives it is also necessary to make these cobots easy to use. They have to be easy to reprogram and that should be done by non-experts. They can be provided with more sophisticated interfaces (Fig. 1.8). This is also the subject of entire fields of robotics such as programming by demonstration ([Restrepo 2018]).

Another concern raised by the close cooperation between humans and robots is ergonomics (Sec.1.2.4). Often disregarded in favour of safety, or only considered for social robots (Sec.1.2.2.2), this aspect of the cooperation deserve more attention.

1.2.2.2 Social Robots

Often confused, service and social robots refer to two different types of robots. Again no official definition exists. The International Organization for Standardization (ISO) defines a service robot as a robot performing useful tasks for humans excluding industrial applications. These tasks are typically ungrateful and repetitive for humans. Vacuum robots like the Roomba (Fig. 1.2b) are a perfect example to illustrate this category.

Social robots are autonomous robots designed to interact and communicate with humans. Unlike service robots, social robots must behave according to social rules and behavior. This is a fairly new branch of robotics requiring a multidisciplinary approach by integrating the advances made in AI, psychology, neuroscience, human factors and many more. Unlike most robots, social robots are designed to be human-like. They have heads, eyes, screens that can take the form of touch pad to communicate with humans. They interact with people in a natural human-like manner in diverse applications such as education, entertainment, communication. Notable works on social works are [Breazeal 2003, Breazeal 2016, Fong 2003].

1.2.3 Safety during Human-Robot Interaction

As mentioned earlier, Human presence has been a restraining factor to the growth of robotics. With the apparition of the first cobot in 1996 finding solutions enabling Humans and robots to share common workspaces has been a major concern. It appears that two main challenges have to be addressed: **safety and ergonomics**. Safety is the most obvious requirement and during the last years many advances have been made. Many studies on the matter have been published and the progresses made in HRI safety partly come from technological progresses made in control, vision, materials and many more.

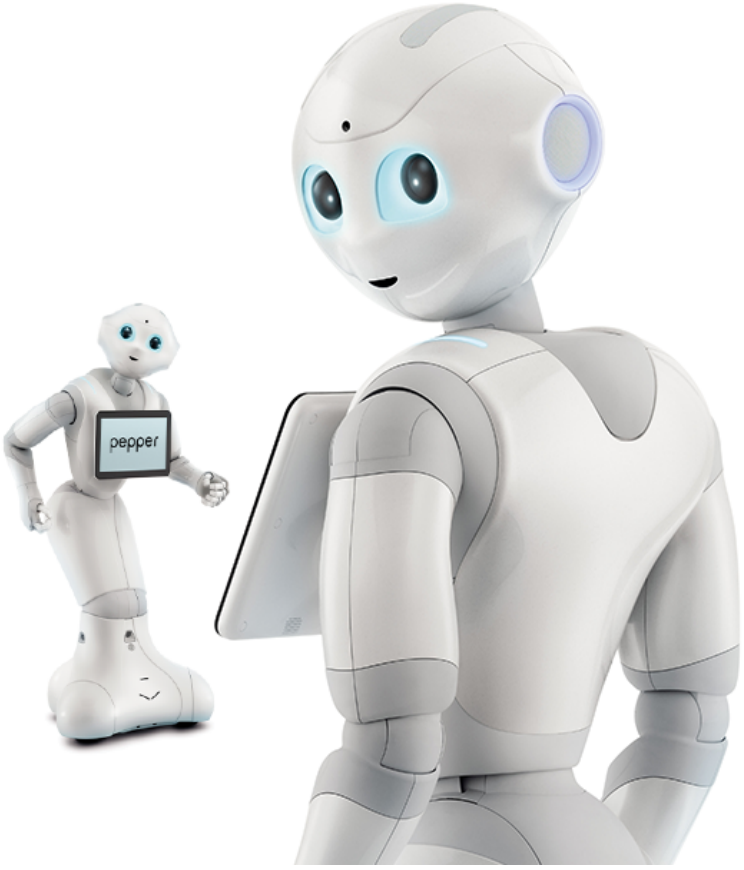


Figure 1.9: Pepper, a well known social robot.

In the following we present the main safety standards as well as some literature on safety for human-robot interaction.

1.2.3.1 ISO standards

We give here a brief explanation of the main safety standards. A more in depth analysis can be found in [Villani 2018] (Fig. 1.10). The International Organization for Standardization (ISO) proposes three standards which specify the security requirements for industrial robots : ISO-10218, ISO-15066, ISO-13482.

[ISO-10218-1 2011, ISO-10218-2 2011, ISO-15066 2016] applies to industrial robot systems, although the safety principles presented can be useful to other areas of robotics. [ISO-13482 2014] applies to personal care robots.

ISO-10218 is considered the central safety standard for industrial robots. It is composed of two parts: [ISO-10218-1 2011] describes the safety requirements for robot manufacturers. [ISO-10218-2 2011] contains safety requirements for the robotic integrator.

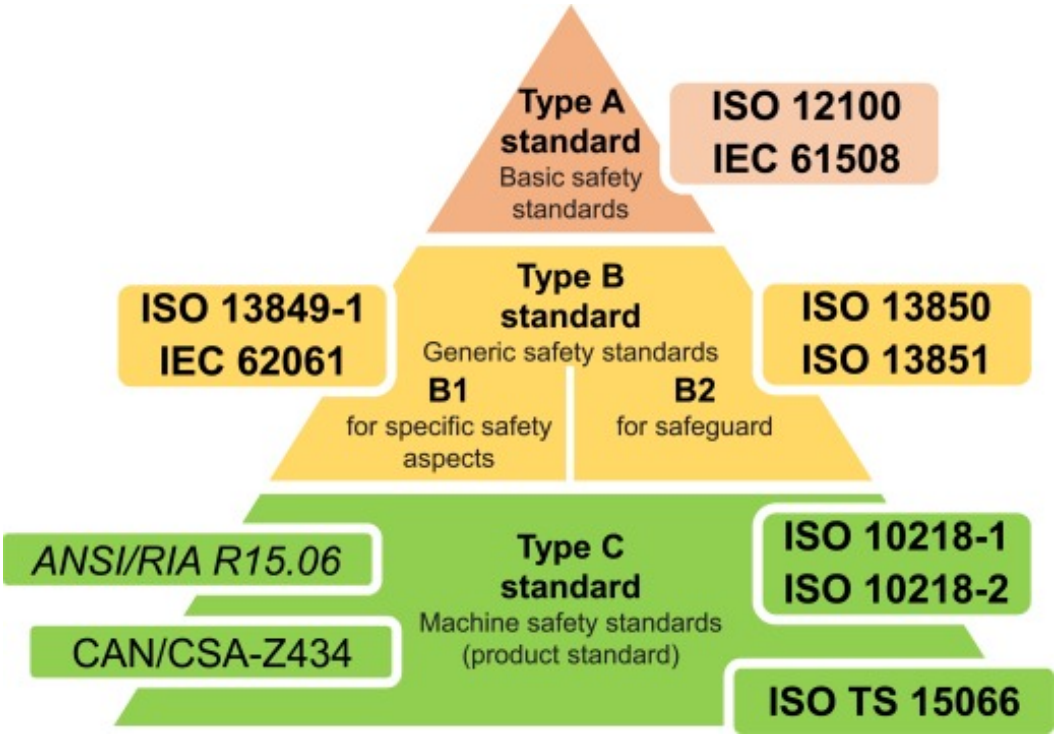


Figure 1.10: A classification of safety standards as provided in [Villani 2018]. The specifications of Type C category have priority.

ISO-15066 supplements the requirements and guidance on collaborative industrial robot operation given in ISO-10218. [ISO-15066 2016] idea is to allow contact between Humans and Robots, thus enabling them to share workspaces. If there were to be any incidental contact between human and machine, it shall not result in pain or injury. Thus ISO/TS 15066 provides guidelines for the design and implementation of a collaborative workspace that reduces risks to people. It specifies:

- Definitions,
- Important characteristics of safety control systems,
- Factors to be considered in the design of collaborative robot systems,
- Built-in safety-related systems and their effective use,
- Guidance on implementing the following collaborative techniques: safety-rated monitored stop; hand guiding; speed and separation monitoring; power and force limiting.

This standard wants to be considered as a first step towards the development of pHRI and plans to evolute as applications are deployed and technology develops.

ISO-13482 specifies requirements and guidelines for the inherently safe design, protective measures, and information for use of personal care robots, in particular the following three types of personal care robots:

- mobile servant robot,
- physical assistant robot,
- person carrier robot.

[ISO-13482 2014] describes hazards associated with the use of these robots, and provides requirements to eliminate, or reduce, the risks associated with these hazards to an acceptable level. ISO-13482 covers human-robot physical contact applications. The scope of ISO ISO-13482 is limited primarily to human care related hazards but, where appropriate, it includes domestic animals or property. Nevertheless this standard is limited. It doesn't apply to earthbound robots, robots travelling faster than 20 km/h, robot toys, robots as medical devices and much more. Attention is drawn to the fact that for hazards related to impact (e.g. due to a collision) no exhaustive and internationally recognized data (e.g. pain or injury limits) exist at the time of publication (2014).

We have to keep in mind that these standards are yet very limited, and do not cover all scenarios. They evolute in reaction of robotics applications, thus are always behind and might not cover the latest applications or technologies.

1.2.3.2 Literature on HRI safety

> A robot may not injure a human being or, through inaction, allow a human being to come to harm.
>
> — Asimov, First law of robotics

Under no circumstances should a robot cause harm to people. Since safety is the main requirement to allow humans to evolve near robotic systems, there is already quantity of work on the matter. Numerous papers cover the theme of safety via the analysis of injury mechanisms. [Haddadin 2009] present an evaluation and classification of possible injuries during physical Human–Robot Interaction (pHRI). The impact's velocity is demonstrated to be the dominant factor in the injury severity, since above a certain mass, potential injury would only depend on the impact's velocity. A critic of the standard ISO10218 is made. For the authors the standard defines truly conservative safety requirements. Indeed the intention of the standard is to keep velocity very low, without compromise, and thus at the expense of efficiency. [Haddadin 2012] formulate the relation between robot mass, velocity, impact geometry and resulting injuries qualified in medical terms. The results are then used to generate motions with safe velocity bounds that explicitly consider the dynamic properties of the manipulator and human's injuries.

The design of robots plays an important part in the making of safe interactions, and the Kuka LWR is a perfect example [Bischoff]. This design implicates rather small and light-weight robots with rounded edges. These robots are equipped of torque sensors to detect collisions, and are capable of mechanical compliance. The design of intrinsically safe robots is studied in [Bicchi 2008]. It is noted that nature comes as a source of inspiration, for example the human arm. Humans can vary the compliance of their arm for different tasks, and even during different phases of a task. This variable compliance can be very useful in the making of safe and fast motions. The human arm is typically controlled to move slowly when it is stiff, and to be compliant while moving fast.

This example can directly be applied on some robotics systems. [Tonietti 2005] present a Variable Stiffness Actuator (VAC) along with a control approach. It allows controlling joint position and stiffness independently at the same time. The control approach consider stiffness similarly as a kinematic bound: by maximizing stiffness under a safe maximum stiffness bound. Hence accuracy is maximized under safety constraints. This illustrates how mechanical means and control strategies can be applied together to obtain safety-performance trade-off.

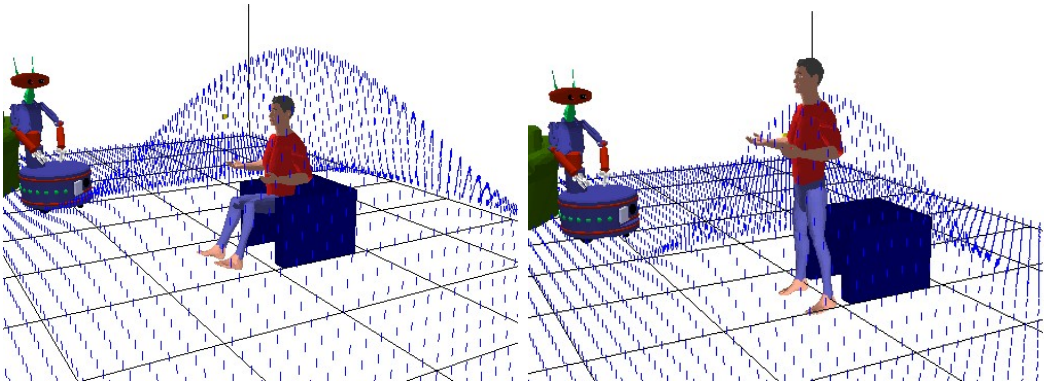
Not the least, safety comes mainly from planning, decision-making and reactive control. The works on collision detection/reaction and obstacle avoidance are numerous [Haddadin 2008, Kulić 2006, Kulić 2005, Kulić 2007].

Finally an in-depth study of methods for safe HRI can be found in [Lasota 2017]. This study also considers an important aspect of safety that is often neglected: psychological safety. It is in fact essential that the human perceives interaction with the robot as safe, and that interaction does not lead to any psychological discomfort or stress as a result of the robot's motion, appearance, conduct or any other attribute. The interaction should not only assure physical safety, but also psychological safety through ergonomics properties.

1.2.4 Interaction Ergonomics

Stress at work is a major preoccupation for our societies and this phenomenon is not bound to a particular field. [Danna 1999, Ganster 2013] proposes multidisciplinary reviews on work stress and well-being at work. A review based on cost of work-related stress is presented in [Hassard 2018]. The cost of this social phenomenon is hard to estimate according to the authors. For [Rosch 2001], the health costs related to job stress is about $300 billion a year for American companies alone.

The origins of job related stress are multiple. In this work we are interested by the stress caused by robots. While safety remains the main criterion to enable humans and robots to share the same workspace, others emerge such as the level of stress and discomfort the human can feel in the vicinity of the robot. In this context, a robot should not cause excessive stress and discomfort to the human for extended periods of time. Some works have been conducted to evaluate the impact of cooperation with cobots on the mental safety of human's beings. [Butler 2001] explores the interactions between humans and mobile personal robots. This paper



(a) (b)

Figure 1.11: Figure illustrating the notion of safety grid presented in [Sisbot 2007]. The robot takes into account that a human feels more threatened when he is sitting rather than standing.

describes the level of comfort the robot causes human subjects according to a variety of behaviors. These behaviors are defined by the robot's speed, distance, design and more. In [Arai 2010], experiments are conducted to study the factors causing stress on a human operator working with a robot in a production assembly system. These factors can be the distance from a swinging robot to an operator, speed at robot's movement towards an operator and so on. Similarly [Fujita 2010] studies the mental strains exerted by a robot hand over motion on a human operator.

In order to guarantee a certain level of comfort and safety for humans working near the robot, we can act mainly on two aspects of the interaction. The first aspect is based on the robot's behavior and social considerations. The second one is based on the robot's motion characteristics. A similar classification is made in [Lasota 2017]. In the following we present some of these works treating these two aspects of the interaction.

1.2.4.1 Behavioral ergonomics

The understanding of human social behavior constitutes an important field of study for roboticians. They attempt to extract the implicit rules and codes that define humans interactions in order to anthropomorphize a class of robots referred as social robots (Sec.1.2.2.2). Such an approach can be used for the robot to anticipate Humans actions as a Human will be more efficient and more satisfied of the interaction if the robot can anticipate his actions [Hoffman 2007]. It can also be used for a robot to communicate its intents. A user will feel more comfortable knowing the goal of the robot early in its movements [Dragan 2013b, Dragan 2013a].

The study conducted in [Lasota 2015] shows that human-aware robots increase the level of safety and comfort of the participants, while increasing their performances. The authors demonstrate that maintaining physical safety by simply preventing collisions, as they are about to occur can lead to low levels of perceived

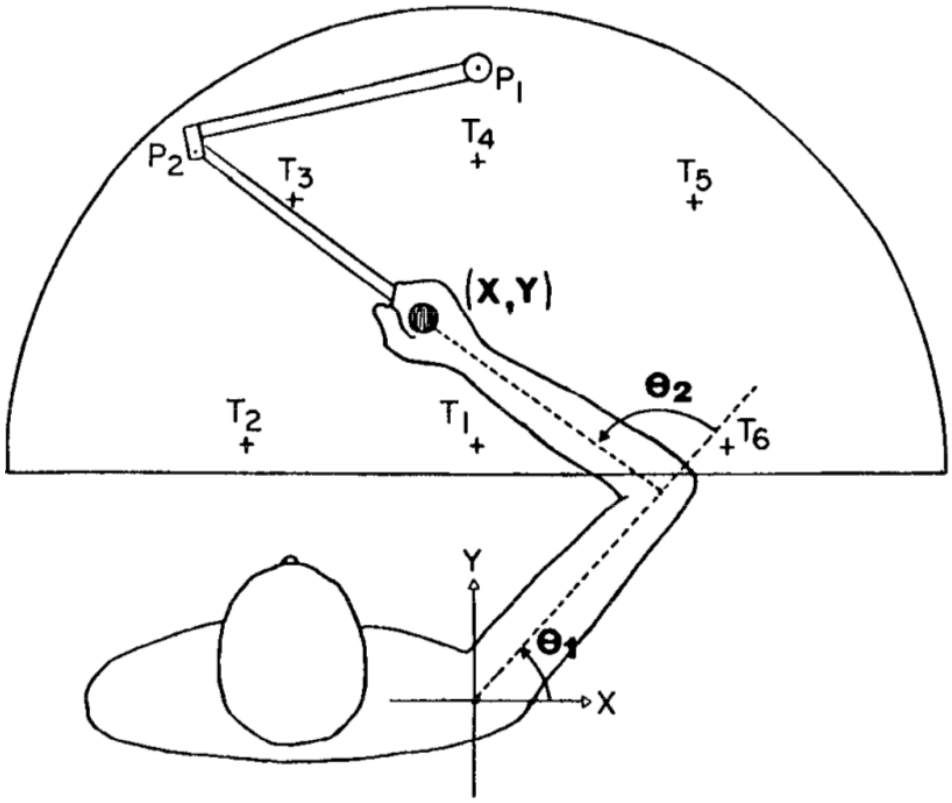


Figure 1.12: A depiction of several similar studies ([Abend 1982, Morasso 1981, Bizzi 1984, Flash 1985]). The formation of humans or monkeys arm motions unconstrained in the horizontal plane is investigated.

safety and comfort among humans.

[Sisbot 2007] presents a human aware motion planner to generate not only safe and effective paths, but also socially acceptable paths. The planner takes into account humans by reasoning about their posture and field of vision. For example a human will feel more comfortable if the robot operate in its field of vision rather than in its back. Similarly a human will feels less threatened when standing in relation to when he is sitting (See Fig 1.11). An extension of this work is presented in [Sisbot 2010] that takes into account the task constraints as well as human kinematics.

These work are usually integrated at a decision level. It is also possible to design more acceptable motions by looking at their properties.

1.2.4.2 Motion ergonomics

Human societal behavior is being studied and mimicked to provide robots with social skills. The same can be applied to human's gesture and movement. In the first instance the resulting works are integrated at the highest layers of planning. The robot movement can be adapted to respect implicit social rules (Fig. 1.11).

But it makes sense only if the motion has been designed beforehand at a lower level to satisfy ergonomics properties.

Since we are used to velocity profiles and trajectories similar to our own by interacting with other individuals throughout our lives, it is reasonable to think we are going to feel more comfortable when interacting with a robot reproducing human-like movements. Thus, as for the behavioral ergonomics, the research of motions with good ergonomics properties will use the human model as a source of inspiration.

Among the works covering this field, the model described by [Hogan 1984] is very popular. Referred as minimum-jerk model, its intent is to be an organizing principle for a class of voluntary movements. It implements the result obtained in a previous study on primates where the objective was to find a general principle for motion control ([Hogan 1982]). At the same time similar studies (Fig. 1.12) were conducted on primates and humans [Abend 1982, Morasso 1981, Bizzi 1984]. It was found that humans generate roughly straight paths with single peaked bell shaped velocity profile for point-to-point motions. This generalization is mainly true if the movement is expressed in Cartesian space, or task space. It is less generalizable in joint space. According to the authors of [Hogan 1984], the previous observations suggest that the underlying goal behind these voluntary movements was to make the motion as smooth and graceful as possible. To verify this theory they used dynamic optimization with an objective function that minimizes the square of the jerk over the duration of the movement, since maximizing the smoothness implies minimizing the jerk. The model's outputs were relatively close to the experimental results and it was assumed that the minimum-jerk model provides a good general description of voluntary arm movements. For [Hogan 1984], the main quality of this model is to be an organizing principle, hence it can be generalized to most voluntary motion and provides superior predictive capabilities. In [Flash 1985] this model is extended to cope with multi-axis. It must also be mentioned that despite the quality of these studies, there were conducted on little panels of subjects.

The minimum-jerk model was built around the hypothesis that the human behavior could be derived from a single organizing principle, which provides a simple generalizable model. However with this assumption comes a trade-off that is a lack of flexibility and adaptability. Yet flexibility and adaptability are some of the most researched feature for future generations of robots (Sec. 1.2.2.1). The model was also simplified to assume symmetrical velocity and acceleration profiles. However the observed results showed that the acceleration phase of a point-to-point movement is often shorter than the deceleration phase [Abend 1982, Morasso 1981, Bizzi 1984]. Different works confirmed that velocity and acceleration curves are asymmetric for a large variety of motions [Beggs 1972, Ostry 1987, Nagasaki 1989], especially for skilled motions. [Flanagan 1990] demonstrates that human movements cannot be generalized by bell-shaped and symmetrical velocity profiles. Similar results are presented in [Shin 2015] showing that human-like movements cannot be reduced to bell-shaped velocity profiles, but depend on the characteristics and the purposes of given tasks. It is also known that the minimum-jerk model fail to achieve its ob-

jective of reproducing human-like motions for curved paths. [Huber 2009] proposes the "decoupled minimum-jerk" model in order to enhance the ergonomic properties of the original model.

To resume human-like motion tends to be smooth, thus with constrained jerk. It cannot be generalized and its characteristics depend on its purpose. Furthermore it is assumed that robots with human-like movements will be optimal for human-robot interactions. The majority of research works hence focus on mimicking humans. Yet the pertinence of robots with their own motion properties can still be investigated.

In order to generate suitable motions for HRI a part of my work during this thesis was dedicated to the research of motions possessing satisfying ergonomic properties.

1.3 Motion generation for Human-Robot Interaction

1.3.1 Architecture for autonomous robots

An autonomous robot architecture can be a very complex system. Its complexity and number of components is directly linked to the nature of the robot, and the complexity of the task itself. The figure 1.13 illustrates an architecture based on the works of [Alami 1998]. Three different layers are depicted: the decisional layer, the execution control layer, and the functional layer. The components placed inside these layers are related to our work that is mainly part of the functional layer. This architecture can be seen as a generic architecture for autonomous robots.

- The decision layer produces the task plan and supervises its execution.
- The execution control layer that generally takes the role of the supervisor, hence it controls the execution of the tasks emitted by the decisional layer, and the validity of those requests.
- The functional layer executes the tasks given by the above layers. This layer communicates with the robots to either send commands to the motors or to retrieve informations such as the axis angular positions. It can also retrieves informations from other sensors that are needed for the task, treat those informations and then transmit them to the execution control layer in a understandable way. In the architecture that is specific to our work in this thesis, the functional layer is subdivided in two components: the trajectory planner and the trajectory controller. The role of each of these components is explained in more details in Sec.3.3.2.

1.3.2 Motion of a body

Kinematics is a branch of classical mechanics that describes the motion of bodies without considering the forces that cause them to move. It differs then from the field of kinetics or dynamics that is concerned with the relationship between motion

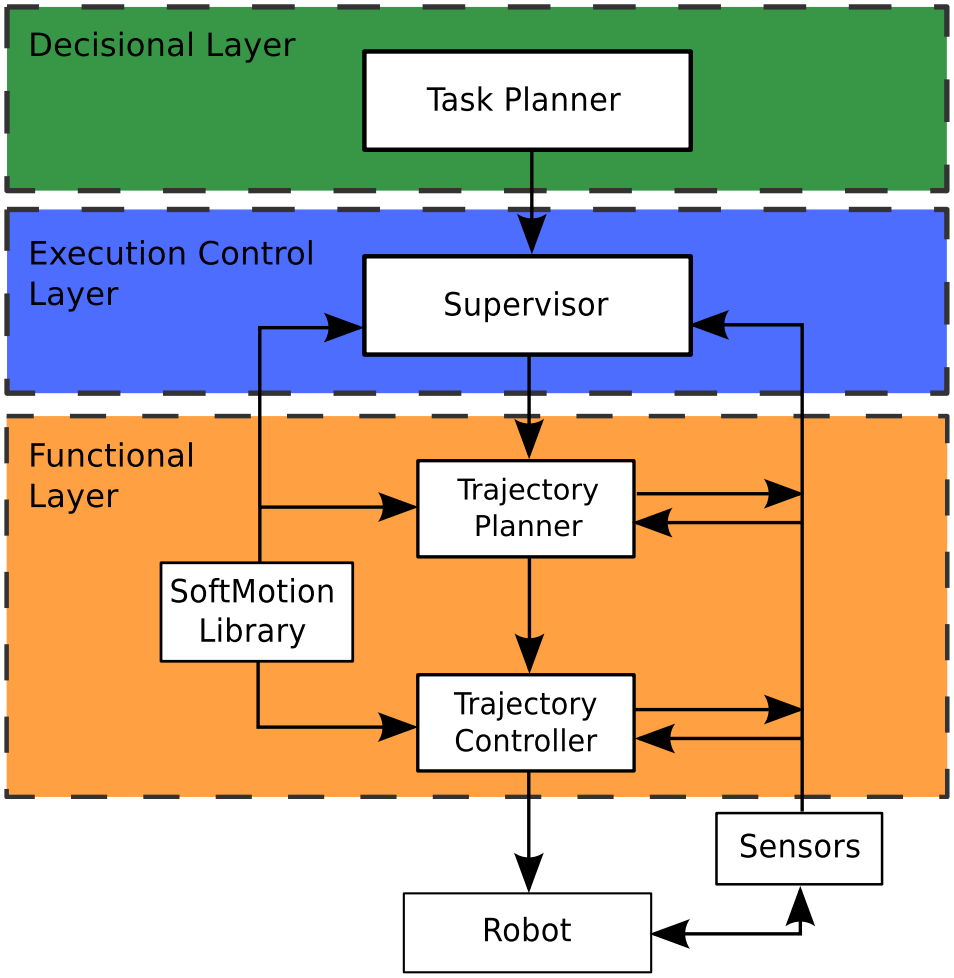


Figure 1.13: Our work in this thesis is mainly part of the functional layer.

and its causes, specifically, forces and torques. Simply put, kinematics describe the evolution of a body according to time by studying its position and its derivatives such as velocity and acceleration. During this thesis we focused our work on the study of kinematics.

1.3.2.1 Kinematics representation of a body

We first consider the localisation of a body $\mathcal{B}$ in a 3D space (See Fig. 1.14). The reference frame $\mathcal{F}_W$ is defined by an origin, O_W, and an orthogonal basis (x_W, y_W, z_W). To $\mathcal{B}$ is associated the frame $\mathcal{F}_B$

To define the translation between two frames, three kinds of coordinates systems are commonly used:

- The Cartesian coordinates system
- Spherical coordinates system

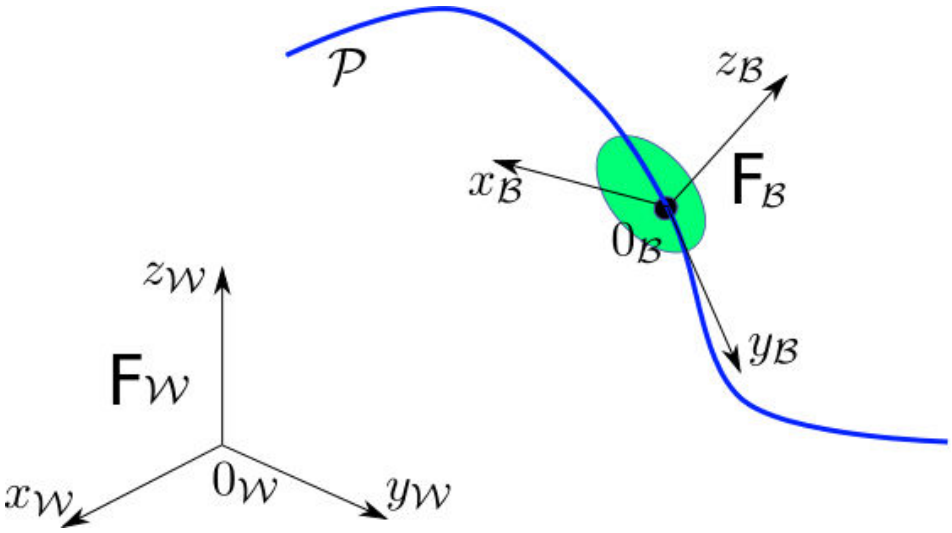


Figure 1.14: Rigid body localization in $\mathbb{R}^3$.

- Cylindrical coordinate system

For the orientation several choices are available:

- 3 angles (Euler angles or Bryant angles)
- Rotation matrix
- Quaternion
- Axis-angle

The homogeneous transformation matrix is often used to represent the transformation between two frames. The homogeneous transformation matrix between the frame $\mathcal{F}_B$ and the frame $\mathcal{F}_W$ is written like so:

$$Th_{[W,B]} = \begin{pmatrix} & R & & P \\ & & & \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1.1}$$

where R is a rotation matrix 3×3 and P a 3×1 matrix for the position of O_B expressed in $\mathcal{F}_W$. Given a point b which is localized in the frame $\mathcal{F}_B$ by:

$$b_B = [x_b, y_b, z_b, 1]^{\mathcal{T}}$$

then coordinates of b in frame $\mathcal{F}_W$ are given as:

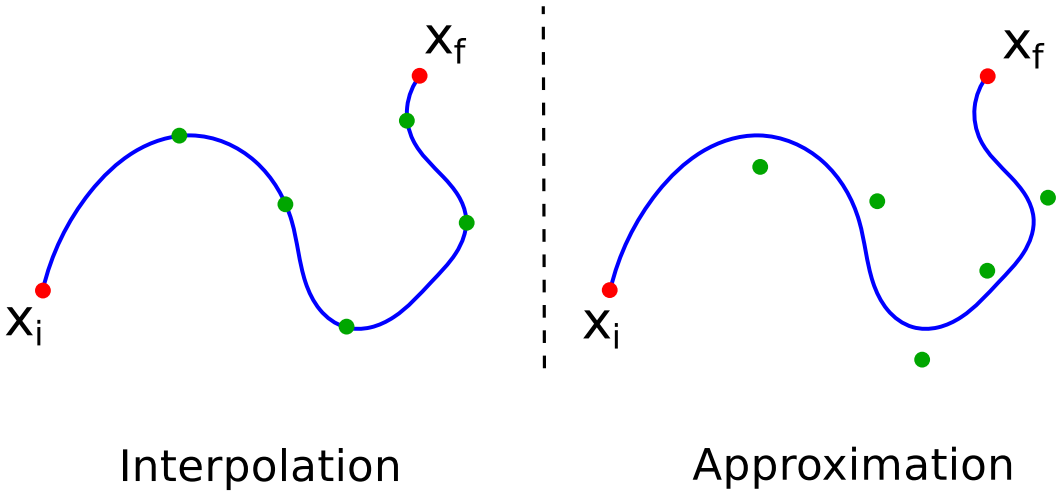


Figure 1.15: Differences between interpolation and approximation.

$$b_W = Th_{[W,B]} * b_B$$

Now given a third frame $\mathcal{F}_D$ associated to a body D, and given the transformation $Th_{[B,D]}$ between $\mathcal{F}_B$ and $\mathcal{F}_D$ we obtain the transformation between $\mathcal{F}_D$ and $\mathcal{F}_W$:

$$Th_{[W,D]} = Th_{[W,B]} * Th_{[B,D]}$$

In this thesis various representations where used such as axis-angle, quaternion and transformation matrices. In this manuscript we will only use the last one. Readers can refer to [Siciliano 2016] for more details on the different representations and their relations.

1.3.2.2 Paths

A path between an initial configuration x_i and a final configuration x_f is a geometric representation of a body movement that does not consider time. It can be defined in the configuration space (joint space), or in the task space (Cartesian space). In this thesis the task space is $\mathbb{R}^3$. Paths defined by a series of points define two types of problems: interpolation and approximation. Interpolation consists in generating a path that goes through all points. With approximation the path must be defined close to the points (See Fig. 1.15).

Notable works covering path planning are [Latombe 1990, LaValle 2006, Kavraki 2008].

1.3.2.3 Trajectories

Trajectories are continuous and derivable functions of time, defining the evolution of the position of the robot. They can be defined in task space or joint space. A trajectory can also be seen as the combination of a path with a law of temporal evolution.

Trajectories can be planned in joint space or Cartesian space. Trajectories planned in joint space present several advantages:

- They can be used directly to control actuators without need to compute inverse kinematics.
- No need to deal with the redundant joints or singularities of multi-DOF manipulators.
- The kinematic constraints can be considered while generating the trajectories. On the contrary, these constraints should be tested after inverse kinematics for Cartesian space trajectories

Trajectories can be classified in different categories. Figure 1.16 presents an example of classification. We can also differentiate online and offline trajectory generation. Online Trajectory Generation (OTG) regroups the solutions to generate trajectories in real-time, typically under $1ms$. In this manuscript we will not consider the multi-points type of trajectories, also called via-points trajectories. The softMotion library already dispose of the faculty to generate via-points trajectories, either interpolated or approximated, after previous works presented in [Broquere 2010, Zhao 2014].

A summary of trajectory generation in an industrial context can be found in [Biagiotti 2008], while [Khalil 2004, Dombre 2007] offer the same for robotic manipulators.

1.3.3 Trajectory Generation for Human-Robot Interaction

1.3.3.1 Objectives and requirements

The aim of this thesis is to improve Human-Robot Interactions by working on robots motions. More specifically this thesis is focused on trajectory generation and control.

Why trajectories? From the previous bibliography we can extract the key features the future generations of robots must present:

- Efficiency (Sec.1.2.1).
- Adaptability and flexibility (Sec.1.2.1).
- Safety (Sec.1.2.3).

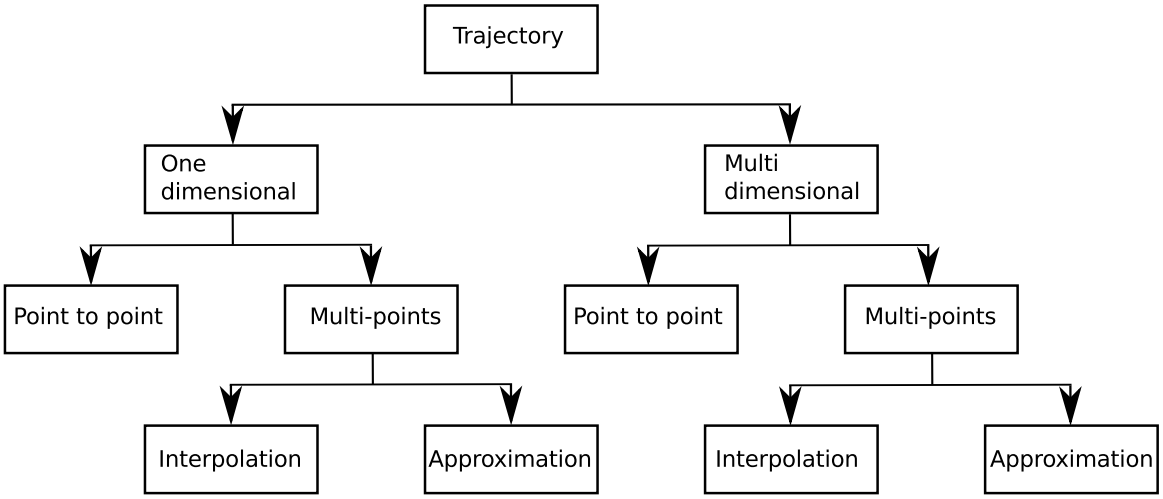


Figure 1.16: A common classification among trajectories [Biagiotti 2008].

- Ergonomics (Sec.1.2.4).

Paths, which are pure geometric representation, have been the most widely used approach to describe movements. Paths are usually generated by a slow task planner, high in the architecture presented in Sec.1.3.1. Time consuming algorithms are used so they can be optimized and avoid obstacles. Hence they are planned offline and are used as inputs for a trajectory planner. Since the work we present is part of the functional layer (Fig. 1.13) we need efficient models with real-time capabilities. We also need to produce efficient motions, generally optimal in time, while considering the human safety and ergonomics properties.

Since paths do not possess a description of the time evolution they cannot be used in this context. Using trajectories instead of paths provides significant advantages [Sidobre 2012]:

- Not only the position can be planned and controlled, but also the velocity, the acceleration, the jerk and eventually higher derivative, allowing to define the smoothness properties.
- The travel time can be optimized taking into account kinematic constraints.
- The movement is more precisely described allowing a better control.

From the state of the art we presented, and more precisely from Sec.1.2.3 and Sec.1.2.4, these properties are necessary to guarantee safe motions from a physical and psychological point of view, as well as efficient motions.

Nonetheless, the question of the choice of a trajectory model among the multiple models available is important to satisfy the requirements of future robots.

In order to produce efficient and safe motions, the model must be time-optimal under kinematic constraints that define safety, both physical and psychological.

This model have to be simple enough for real-time constraints that are linked to safety. It should also be capable to generate human-friendly movements to satisfy ergonomics constraints for a large variety of applications. Smooth trajectories appear as excellent candidates since they possesses the advantages necessary to ensure safety, ergonomics, efficiency and adaptability required in the making of collaborative motions.

1.3.3.2 Smooth trajectories

Smooth trajectories were first introduced by [Hogan 1984], which was introduced in Sec.1.2.4.2. The first objective was to reduce wear on systems and improve path tracking [Craig 1986, Kyriakopoulos 1988]. These qualities make smooth trajectories particularly interesting for machining. For instance the study [Rivera-Guillen 2010] shows that tool-life can be improved up to 60%. One can resume the advantages of working with smooth trajectories as follow:

- Improve accuracy, thus moves can be executed more rapidly and accurately.
- Extend the life span of the manipulators as vibrations are reduced thereby preventing actuators to be damaged and reducing wear of the robot joints.
- Reduce stress and discomfort of human co-worker.

Because of their qualities, smooth trajectories can be used in many contexts and can be employed in the making of more efficient and flexible robots. The smoothness of a trajectory can be defined by the number of derivative of the position and the extreme values of these derivatives. It is generally accepted that a smooth trajectory has at least continuous speed and acceleration, hence a bounded jerk. Considering a constant jerk during a period of time and its triple integral with respect to time, the obtained trajectory is defined by a cubic polynomial function of time. As these cubic functions are simple and their properties well known, they are easy to manipulate, and thus are widely used.

However higher order polynomials, especially quintics, are often used in the literature to obtain smooth trajectories. The main reason being the need to compute trajectories with the possibility to specify position, velocity and acceleration at both ends, so that the robot is able to react quickly to unforeseen events, an imperative in HRI context to ensure safety [Kroger 2010]. One quintic is enough to compute a trajectory that meet these criteria, hence justifying their use [Craig 1986]. Unfortunately, quintics generate more computational burden than cubics and their behavior between the way-points is more unpredictable and less faithful to the expected trajectory. This is explained by the tendency to oscillate of the quintics and generally higher order trajectories [Macfarlane 2003]. A solution is to use more than one quintic, but doing so they loose all interest, as simpler solutions exist. More recently, sequences of cubic polynomial functions have been used to define a smooth trajectory joining arbitrary positions, velocities and accelerations [Broquere 2010, Zhao 2014].

A well-known model of smooth trajectory is the minimum-jerk model. Minimum-jerk trajectories and smooth trajectories were actually introduced together in the works of Hogan. This model was aimed to reproduce human motions to provide ergonomics properties. We discuss this model more in the following.

1.3.3.3 Minimum-jerk model

In order to generate suitable movements for a Human-Robot interaction, human motion appears naturally as a source of inspiration. Among the numerous works that have covered this field, the model described by [Hogan 1982] is very popular. This mathematical model describes the organization of a class of voluntary arm movements. We have introduced this model in details in Sec.1.2.4.2. This model present some flaws as it aims to provide an organizing principle for voluntary arm motions, while human movements cannot be generalized so easily.

Still minimum-jerk model has been popular and used in order to obtain coordinated and natural human-like motion. [Kyriakopoulos 1988] used a minimax approach to minimize the maximum jerk. This work was later extended by [Piazzi 1997, Piazzi 2000] where interval analysis is used to globally minimizes the absolute maximum value of the jerk along a trajectory. This global minimization avoids a flaw present in minimum-jerk works, generally subjects to get stuck in local minima. In [Amirabdollahian 2002], fifth order polynomials for minimum-jerk control are used with symmetric or asymmetric jerk bounds.

In most of the different approaches listed above kinematic constraints are not considered and the time has to be set a priori. Some studies have considered minimizing a mixed criterion such as [Gasparetto 2008, Zanotto 2011]. [Gasparetto 2008] adopts a trade-off between a short execution time and smoothness of the motion by using a mixed criterion minimizing both the jerk and time. Kinematic constraints are taken as inputs, avoiding setting the time a priori.

This approach tries to overcome a major drawback of the model that is a lack of flexibility and efficiency. Another approach is the one proposed in [Villani 2018]. In place of considering safety as a requirement that limits performance, performance oriented solutions should be considered, meaning that performances should be optimized subject to the constraint of safety. In term of trajectory generation this means that we should look for the time-optimal trajectory under some kinematic constraints that have been chose accordingly to the context. This approach satisfies all the requirements of efficiency, flexibility, and safety (physical and psychological). Our work in trajectory generation follows this approach that can be named constrained-jerk model.

1.3.3.4 Constrained-jerk model

In this approach, a suitable application dependent maximum jerk J_{max} is established through experiments or information provided by the manufacturer. This threshold is determined according to certain criteria related to the nature of the

task. Once the maximum jerk is defined the problem left is that of a classic time's optimization. This is done by maximizing the jerk under the constraint $J < J_{max}$. It provides time-optimal trajectories under task dependent constraints defined by the user. **This approach can also be extended to acceleration, velocity and other derivatives. The main limitation of this model is that kinematic constraints must be specified by the user.** However, unlike the previous approach, it provides qualities that are essential to build the future generations of collaborative robots that we expect to be efficient, adaptable and reusable (Sec.1.2.1).

[Liu 2002] proposes a real-time algorithm to generate smooth trajectories from current velocity under constraints on jerk, acceleration and velocity. Optimal in most cases, this paper points the difficulty of managing non-null initials and finals conditions. In the calculation steps for the maximum reachable speed, if the motion is too short to reach the maximum speed or acceleration, it becomes very difficult to compute an analytical solution online. A suboptimal strategy is then adopted by keeping the initial speed for a certain period. A similar approach is presented in [Haschke 2008] to generate third order time-optimal trajectories. The main contribution concerns the ability to handle arbitrary initial conditions, while end conditions must stay at rest. Nonetheless this work encounters numerical problems and produces infinite jerk for short displacements. In that case, second order trajectories are employed as a fallback solution. Similar results can be found in [Kroger 2006], except it is for quadratic trajectories and hence not smooth, since the jerk is not bounded. [Kroger 2010] introduced a general Online Trajectory Generation algorithm and an instance of it using third order polynomials. It brings a manipulator from an arbitrary initial state to an arbitrary final state except for the final acceleration, which is always null. This work as been extended in [Kroger 2012] such that time-variant kinematic motion constraints are considered.

Third order polynomial trajectories offer a simple solution to generate jerk bounded trajectories as they are easy to manipulate, keep the jerk bounded and can be generated online. They also avoid some major drawbacks of higher degree polynomials such as the tendency to oscillate, and are better for approximation [Macfarlane 2003]. Moreover it has been demonstrated that a concatenation of at most seven cubic is enough to represent any time-optimal trajectory [Broquere 2008]. [Herrera-Aguilar 2006] proposes seven cubic equations to obtain Soft Motions for robot service applications. It was extended later in [Broquere 2008] to compute online time-optimal trajectories given any initial and final conditions, under bounded jerk, acceleration and velocity, for any number of independently acting axis. A solution for time-imposed trajectories is presented in [Broquere 2010] which leads to a simpler axis-synchronization for multi-axis systems. Yet, this solution had drawbacks, since it was not possible to both impose the time and keep the jerk bounded. An improvement was proposed in [Zhao 2014], which allows to have an imposed time and bounded jerk for sufficient large motions. For a point to point movement in an N-dimensional space, the time optimum straight-line motion is obtained by projecting the constraints on the line [Sidobre 2012].

Polynomial and trigonometric models have also been combined in order to design

smooth trajectories [Macfarlane 2003, Nguyen 2008]. [Macfarlane 2003] introduces an online method to compute smooth trajectories for industrial robots. Concatenations of fifth order polynomials are employed to join the waypoints approximating the desired trajectories. Oscillations due to the use of quintics are here corrected by sine-wave template for accelerations. The solution presented is not optimal, and the jerk continuity is not guaranteed either, despite the use of fifth order polynomials. Trigonometric and polynomial models are also combined to design s-curve motions of any order from rest to rest in [Nguyen 2008].

In more recent works [Ezair 2014] displays a new algorithm to generate smooth trajectories of any order under kinematic constraints and for multi-axis systems. A key point is its ability to deal with any general initial and final state. The algorithm builds trajectories from an input speed, which is updated iteratively by binary search until a near-optimal cruise speed is found (or peak when the motion is too short). A recursive approach is used to build higher order trajectories from lower ones. An interesting addition of this paper is the introduction of asymmetric constraints. Unfortunately this work presents some limits other than its non-optimality regarding a time criterion. Binary search in non-linear systems can lead to difficulties, such as being trapped in local minima. Moreover, it does not guarantee a solution can be found.

1.4 Conclusion

This chapter presented a state of the art on robotics, starting from an overview on the current state of robotics and ending with the problematic of motion generation for robots. A new definition of a robot has been proposed that emphasizes the crucial role of motion generation in robotics as it is a common concern to all robots.

Moreover motion generation has been a restricting factor to the growth of robotics in the past. For the emergence of human-robot cooperation, robots have to generate motions ensuring the safety of humans, both physical and physchological. Because of this requirement, robots were confined in cages in the past decades, but thanks to the constant technological progress this will no more be the case. This new type of cooperation, that combines the respective strengths of humans and robots, is strongly motivated by the needs for flexibility, adaptability and reusability of assembly systems of the industry 4.0. In this context, motion generation, and more specifically trajectory generation, has an important role to play in the making of efficient, flexible and safe robots.

Smooth trajectories provide the properties necessary to guarantee safe motions from a physical and psychological point of view.

The constrained jerk model is an interesting solution to build smooth trajectories as it is performance based : the performance is optimized under safety constraints which must be specified. This constraints can be specified by the user, dertermined by the environment or the task. Hence this model offer great flexibility.

Our work in trajectory generation that we present in the next chapters is based

on this model.

Chapter 2

Trajectory generation

Contents

2.1 Introduction

The previous chapter presented a state of the art where in the first part we tried to understand what the future generation of robots should be capable of, and what properties should they have. In the second part we focused on how the field of motion generation, and more precisely trajectory generation, could incorporate those needs in order to produce better motion for the Human-Robot Interactions. We finally presented the constrained-jerk model, which we believe, is the most suitable model of trajectories in order to provide robots motions with efficiency, flexibility, safety, and ergonomics.

During our work we choose to examine the sequences of third order polynomial functions, one of the simpler models to build smooth trajectories, as smooth trajectories must have bounded jerk. To our knowledge (See Table. 2.1), there is no work proposing a complete answer to trajectory generation satisfying simultaneously all the following criteria to build safe, efficient, adaptable and human-friendly robots:

Table 2.1: Classification of main trajectory planning algorithms in HRI context. The references are: [1]: [Liu 2002], [2]: [Macfarlane 2003], [3]: [Lambrechts 2005], [4]: [Kroger 2006], [5]: [Nguyen 2008], [6]: [Broquere 2008], [7]: [Haschke 2008], [8]: [Kroger 2010], [9]: [Ezair 2014], Ours: [Sidobre 2019]

Reference	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	Ours
$J_{max} \in \mathbb{R}$	+	+	+		+	+	−	+	+	+
$V_I \in \mathbb{R}$	+	+		+		+	+	+	+	+
$A_I \in \mathbb{R}$				+		+	+	+	+	+
$V_F \in \mathbb{R}$		+		+		+		+	+	+
$A_F \in \mathbb{R}$				+		+			+	+
Asymmetric bounds									+	+
Online	+	+		+		+	+	+	+	+
Optimal	−			+		+	+	+		+

"—": the criterion is not satisfied for every scenario.

- Real-time capable.
- General initial and end conditions for both velocity and acceleration.
- General bounded jerk, acceleration and velocity.
- Asymmetric bounds.
- Time optimal.

The algorithm must be real-time capable and must accept general conditions for the situation where the robot must react to unforeseen events (this aspect will be more developed in chapter 3). Like we explained in Sec.1.3.3.3 we are looking for performance based solutions, hence time-optimal trajectories with bounded jerk, acceleration and velocity in order to guarantee safety, both physical and psychological. The general asymmetric bounds offer more flexibility to the model, which can be used for example to improve the reproduction of human gesture (Sec.1.2.4.2). The algorithm we present here is the first, to our knowledge, to fulfill simultaneously all those criteria. Despite the many works in the field, trajectory generation still encounters serious difficulties especially for short and asymmetric moves. These harsh points have been encountered in several previous papers, we attempt to address them in this work. The time-optimal curve regarding the length of the motion might be non-linear and present discontinuities, particularly for short motions or velocity shift.

2.1.1 Contributions

2.1.1.1 The softMotion library

The work presented in this chapter, and its extension in the next chapter, are a contribution to the development of the softMotion library[1]. The softMotion library is a trajectory generation library that has been developed at LAAS since 2006. Numerous works contributed to its development [Herrera-Aguilar 2006, Broquere 2008, Broquere 2010, Sidobre 2012, Zhao 2014]. The work presented in this chapter was published in [Sidobre 2019]. It is the result of a rewriting of the softMotion library. The library that was originally partially written in C has been rewritten in $C++$. The new version of the library possesses now more features as we can see in Table. 2.1 by comparing [Broquere 2008] and [Sidobre 2019]. The rework of softMotion began in 2015 before the beginning of this thesis. Some features like the generation of via-points trajectory that have been published in [Broquere 2010, Zhao 2014] have been reused in the newest version and we will not discuss these parts in this chapter.

2.1.1.2 Scientific contribution

We propose the first algorithm to generate smooth trajectories defined by a chain of cubic polynomial functions that joins two arbitrary conditions defined by position, velocity and acceleration in minimum time under asymmetric bounds on velocity, acceleration and jerk. We also attempt to understand the harsh points associated to non-linear systems that only few papers address [Costantinescu 2000, Gerelli 2009, Bianco 2011].

2.1.2 Organization of the chapter

This chapter is organized as follow: The first section presents the mono-dimensional algorithm. The time-optimal cubic polynomials trajectories are introduced in Sec.2.2.1. The duration of the trajectories according to the length is presented in Sec.2.2.2. The time-optimal trajectories are developed in Sec.2.2.3. Sec.2.2.4 gives some details in the solving of the related equations. Discussions are presented in Sec.2.2.6. The second section extends the algorithm to the multi-dimensional case Sec.2.3. Finally, the last section concludes this chapter Sec.2.4.

2.2 Smooth Cubic Polynomial Trajectories

2.2.1 Time-optimal cubic polynomial trajectories

One key lesson to be drawn from this bibliography is that solutions are well known for large movements, but shorter movements exhibit more complex behavior. This problem has been approached in some works [Liu 2002, Macfarlane 2003,

[1] https://git.openrobots.org/projects/softmotion/wiki

Haschke 2008], but avoided by the majority. To better understand these short moves, we first develop a graphical approach based on the phase diagram and then plot the time-optimal solution as a function of the move length x_f. From these results we then introduce the first complete algorithm to compute the time-optimal motion.

2.2.1.1 Problem definition

The initial condition $C_i = (x_i, v_i, a_i)$ is defined by the position x_i, the velocity v_i and the acceleration a_i. The final condition is similarly defined by $C_f = (x_f, v_f, a_f)$. Without loss of generality, we choose $x_i = 0$, i.e. we define x_i as the origin. The trajectory must comply with the asymmetric bounds $\mathcal{B} = (J_{min}, J_{max}, A_{min}, A_{max}, V_{min}, V_{max})$ for jerk, acceleration and velocity. These kinematic bounds define an admissible kinematic domain noted $\mathcal{D}$.

From these conditions, our objective is to find the time-optimal trajectory $\mathcal{T}(t)$, which is known to be a series of at most seven trajectory segments that each saturates one of the bounds [Broquere 2008, Liu 2002]. Each of these trajectory segments S_n is a polynomial cubic function of time t and it is characterized by a constant jerk $J_n \in \{J_{min}, J_{max}, 0\}$, an initial instant τ_n, a duration T_n and its initial condition $C_n = (x_n, v_n, a_n)$. For all t such that $\tau_n \leqslant t \leqslant (\tau_n + T_n)$:

$$\begin{aligned} S_n(t) &= \frac{1}{6}J_n(t-\tau_n)^3 + \frac{1}{2}a_n(t-\tau_n)^2 + v_n(t-\tau_n) + x_n \\ \dot{S}_n(t) &= \frac{1}{2}J_n(t-\tau_n)^2 + a_n(t-\tau_n) + v_n \\ \ddot{S}_n(t) &= J_n(t-\tau_n) + a_n \end{aligned} \tag{2.1}$$

2.2.1.2 The phase diagram

These trajectories are well described with a phase diagram (Fig. 2.1) where abscissa is velocity and ordinate is acceleration. In this diagram, constant jerk trajectories associated to third degree polynomial functions define horizontal parabolas, constant acceleration motions associated to second degree polynomial functions are horizontal lines and constant velocities motions are associated to points of the null acceleration axis. The equations of the parabolas are obtained by eliminating the time in the two last equation of (Eq.2.1):

$$\dot{S}_n = \frac{\ddot{S}_n{}^2}{2J_n} + v_n - \frac{{a_n}^2}{2J_n} \tag{2.2}$$

The area where acceleration and speed constraints are verified is plotted in green in the figures. The acceleration bounds A_{min} and A_{max} define two horizontal boundary lines while the left and right boundaries are parabolas respectively defined by (V_{min}, J_{max}) and (V_{max}, J_{min}).

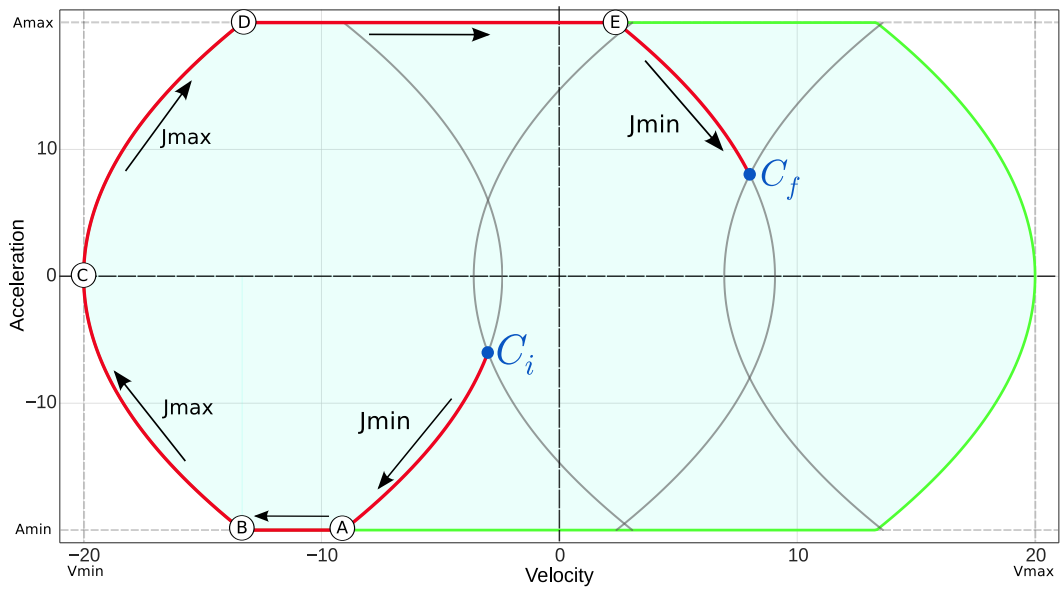


Figure 2.1: Phase diagram of a classical seven segments trajectory for a large negative motion. The green area of admissible conditions $\mathcal{D}$ is limited by the acceleration bounds (A_{min}, A_{max}) and the parabolas associated to velocity bounds (V_{min}, V_{max}). The jerks J_{min} and J_{max} define four condition parabolas passing throw the initial conditions C_i and the final condition C_f. The red curve joins C_f from C_i with the following 7 segments: $J_{min} \rightarrow A_{min} \rightarrow J_{max} \rightarrow V_{min} \rightarrow J_{max} \rightarrow A_{max} \rightarrow J_{min}$. The segment with saturated speed V_{min} holds on point C.

By continuously varying the length x_f from a large negative value to a large positive one, the trajectory in the phase diagram reaches successively different limit shapes. These shapes are defined by a sequence of trajectory segments. For example, the large negative values are associated to the classical seven segments trajectory plotted in red in Fig. 2.1 and the limit case of this trajectory sequence is reached when the minimum velocity V_{min} segment lasts zero seconds. Just after this limit, five segments only define the trajectories. We associate the type (JAJVJAJ) to these sequence of trajectories and the type (JAJAJ) to the one without the V_{min}/V_{max} segment where J stands for jerk segment and A for acceleration segment.

Introducing the notion of type allows decomposing the problem in three sub-problems, the last one being trivial:

1. Find the possible types of the solution.
2. For each possible type, compute the associated trajectory.
3. Select the faster trajectory.

In the following, we firstly develop a general method to compute the border trajectories between two types.

2.2.1.3 Local parabolas

From the initial condition C_i defined by v_i and a_i, the two jerk bounds allows only two possible optimal motions, which define two condition parabolas in the phase diagram (See Fig. 2.1). Similarly, only two motions i.e. two condition parabolas are possible to reach the final condition C_f defined by v_f and a_f. As these four condition parabolas are symmetric with respect to the horizontal axis of zero acceleration, they only define two relative configurations depending on whether only two parabolas intersect (Fig. 2.1) or the two interior parabolas also intersect (Fig. 2.3). A trajectory joining C_i to C_f and composed by two jerk bounds segments is called a direct trajectory. In the phase diagram, these direct trajectories are associated to arcs of condition parabolas. When C_i and C_f are on the same parabola, the direct trajectory is reduced to one arc. If a part of these two arcs of parabolas is outside of $\mathcal{D}$, it is replaced by an horizontal segment associated to the acceleration A_{min} or A_{max}.

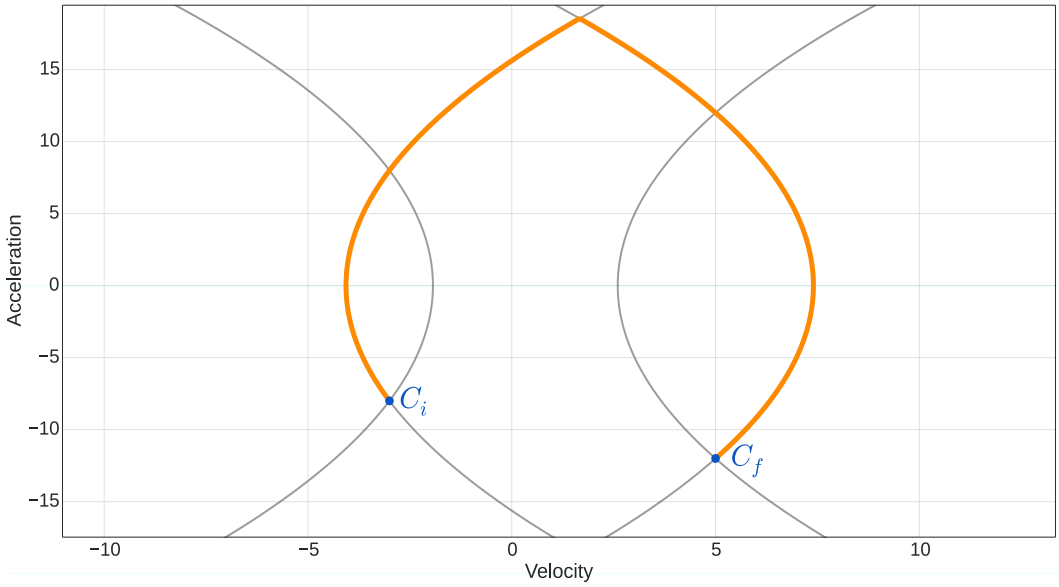
Each pair of initial and final conditions defines at least one direct trajectory, for example the orange trajectory of Fig. 2.2a. When the internal parabolas also intersect defining shortcut trajectories, up to three direct trajectories can exists (See Fig. 2.3).

2.2.1.4 Varying the trajectory length

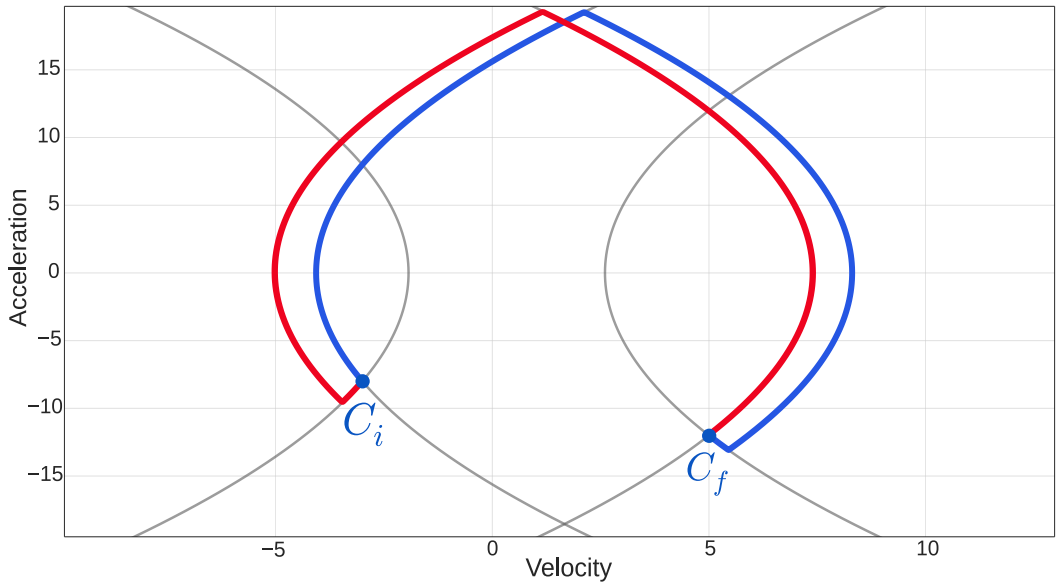
Now, we propose to explore the possibility of varying the trajectory length x_f in the neighborhood of a direct trajectory of length x_d by adding an optimum motion, i.e. a J_{min} or J_{max} parabolic arc A^p. For example, the red trajectory of the Fig. 2.2b is the result of adding a small arc, associated to a negative jerk segment, at the beginning of the motion that translate the negative parabola toward the left. The length of the red trajectory is $x_f < x_d$. Similarly, adding a small positive motion at the end of the trajectory increases the length of the motion, see for example the blue trajectory in Fig. 2.2b.

The modification of the length is less intuitive in the case of Fig. 2.4 where direct trajectory doesn't reach the zero acceleration line. In this case, adding a short J_{min} jerk trajectory at the beginning of the motion, translates the parabola with J_{max} jerk to the right. The extension of this added segment brings the end of the first segment to progressively reach the zero acceleration axis until a cusp appears (Fig. 2.5). Continuing to extend this added segment create a loop (Fig. 2.6) that can be extended to the V_{min} limit. From the cusp, the second segment translates back to the left. Generally, during this extension the length x_f is not monotonically decreasing as we will see bellow.

This complex behavior appears each time one parabola of the direct trajectory doesn't cross the zero acceleration line to reach the condition C_i or C_f. In the particular case of the Fig. 2.5, it appears on both sides.



(a) The direct trajectory $x_f = x_d$.

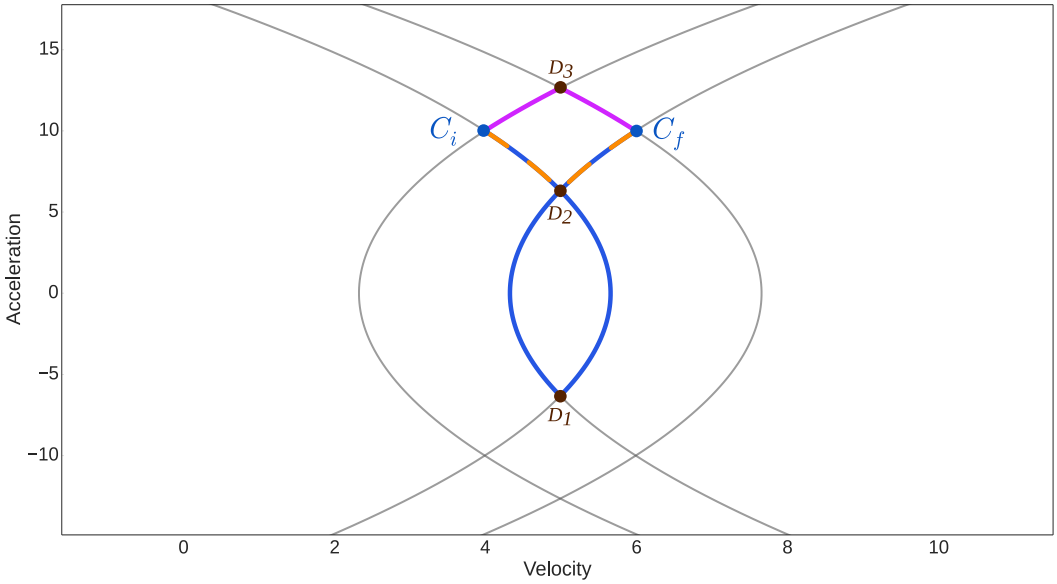


(b) Trajectories with $x_f < x_d$ in red and $x_f > x_d$ in blue.

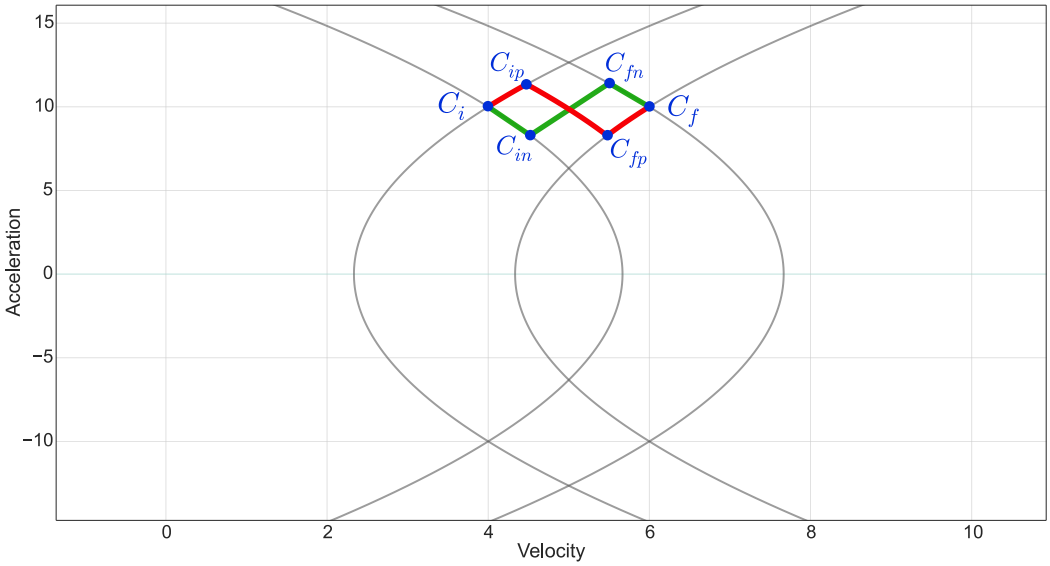
Figure 2.2: Evolution of the phase of the trajectory with the length x_f around the direct trajectory.

2.2.1.5 Internal parabolas also intersect

In the case of the Fig. 2.3 where the internal parabolas also intersect, three direct trajectories are possible: (C_i, D_3, C_f), (C_i, D_2, C_f) and (C_i, D_1, C_f). The last trajectory provides an evolution similar to the case of the Fig. 2.2 and can be extended to both V_{min} and V_{max}. The two first trajectories introduce alternative trajectories



(a) Three direct trajectories for the same configuration.



(b) Illustration of shortcut trajectories.

Figure 2.3: Limits trajectories in the case where the two parabolas defined by C_i intersect the two parabolas defined by C_f. Three direct trajectories are present, the main one is in blue (Fig. 2.3a). The red and green trajectories are two shortcuts trajectories (Fig. 2.3b). ($J_{min}=-30, J_{max}=30, A_{min}=-20, A_{max}=20, V_{min}=-20, V_{max}=20, a_i=10, v_i=4, a_f=10, v_f=6$).

for a range of values of x_f. Two samples, $(C_i, C_{ip}, C_{fp}, C_f)$ and $(C_i, C_{in}, C_{fn}, C_f)$, illustrate these shortcut trajectories in the Fig. 2.3b.

The trajectory (C_i, D_1, C_f) is endlessly expandable, but $(C_i, C_{ip}, C_{fp}, C_f)$ and $(C_i, C_{in}, C_{fn}, C_f)$ have shorter curves in the phase diagram where they generally

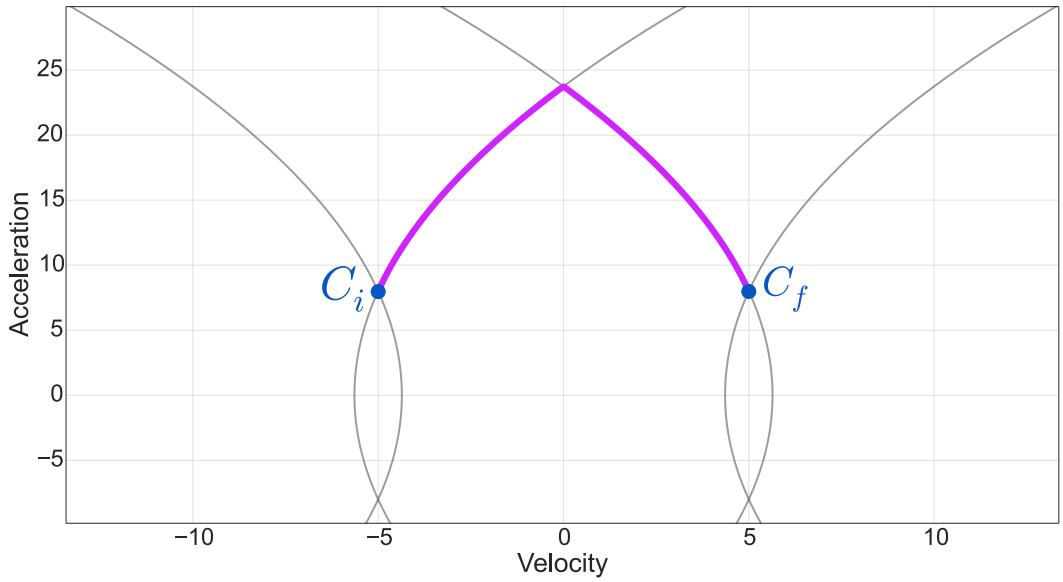


Figure 2.4: A direct trajectory. ($J_{min}=-50$, $J_{max}=50$, $A_{min}=-30$, $A_{max}=30$, $V_{min}=-30$, $V_{max}=30$, $a_i=8$, $v_i=-5$, $a_f=8$, $v_f=5$).

define faster trajectories for a small interval of values.

To summarize, in all cases the direct trajectory (Fig. 2.4) can be respectively extended to long negative motions and to long positive motions. The behavior of these trajectories is more complex for short motions, i.e. around the direct trajectories, where shortcut trajectories can exist (Fig. 2.3). This point is developed further and constitutes a major contribution of our work.

2.2.2 Duration of the trajectories according to the length

As expected, trajectories show a more complex behavior around the direct trajectory. We are going to plot the optimal time of these trajectories versus their length x_f to obtain the curve $\mathcal{C}_{opt}$. In a first stage, we propose to plot the duration of the trajectory according to its length as a parameterized curve $\mathcal{C}(x_f, t_f)$. We choose the time length of the added arc A^p (Sec.2.2.1.4) as parameter, thereby defining a parameter t_n for left negative side and a parameter t_p for right positive.

2.2.2.1 A simple case

In order to simplify the presentation, we firstly plot the curves $\mathcal{C}(t_n)$ and $\mathcal{C}(t_f)$ (see Fig. 2.7) for the particular case of the Fig. 2.6 where internal parabolas doesn't intersect and without considering the bounds. Using the conditions of the Fig. 2.6, we compute the times of the trajectory segments and then apply the equations (Eq.2.1). The times t_n and t_p are null for the direct trajectory (Fig. 2.2a). A time $t_n > 0$ defines a parabolic arc A^p between the initial acceleration a_0 and

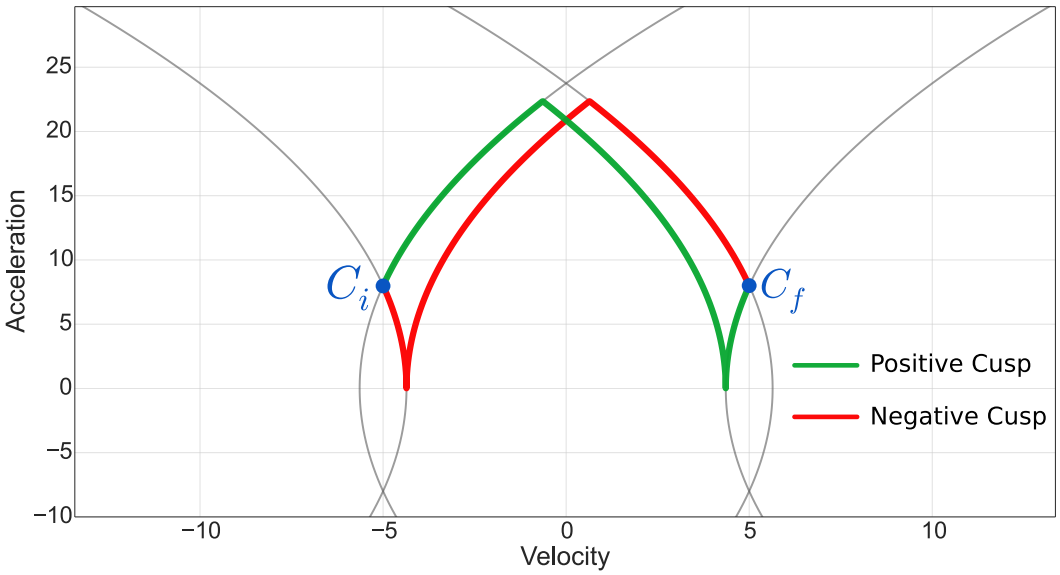


Figure 2.5: The direct trajectory displayed in Fig. 2.4 can be extended by adding a positive or a negative arc of parabola. Two limit cases are displayed, a negative and a positive cusps. By extending them more, we obtain trajectories with a loop (Fig. 2.6).

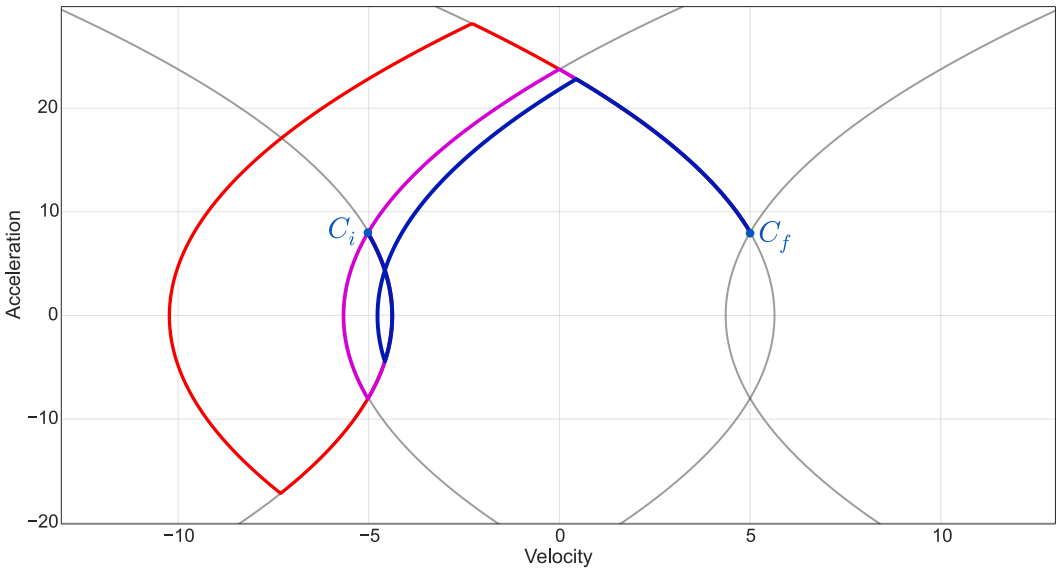


Figure 2.6: A small negative loop in blue reaching the symmetric condition of C_i and a larger one in red. They are obtained by extending the negative cusp of Fig. 2.5.

the acceleration $a_n = a_0 + J_{min} t_n$. Respectively, $a_p = a_0 + J_{max} t_p$ for a positive

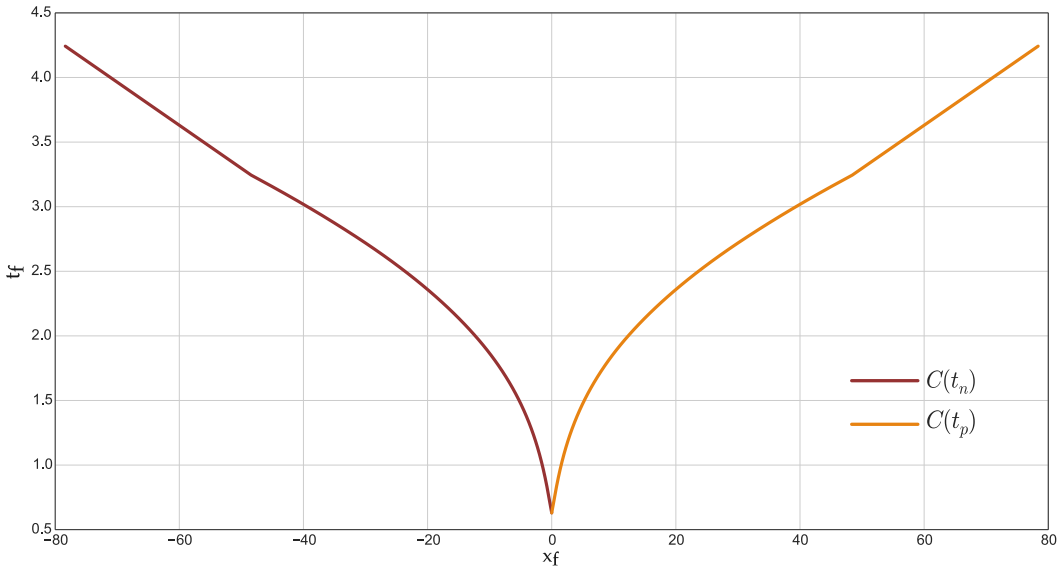


Figure 2.7: The optimal time solutions (t_f) versus the trajectory length (x_f). A trajectory length of zero means the start and end positions are the same. For example, the trajectory with $x_f = 0$ length is here associated to the direct trajectory of Fig. 2.4, and the null length is explained by the presence of symmetries. (J_{min}=−50, J_{max}=50, A_{min}=−30, A_{max}=30, V_{min}=−30, V_{max}=30, a_i=8, v_i=−5, a_f=8, v_f=5).

parabolic arc. For example, in the case of the Fig. 2.6, the first purple negative arc starts from C_i with acceleration a_i and ends at $-a_i$. Its duration is $t_n = -2a_i/J_{min}$.

Using (2.1), we obtain the condition at the end of the first segment associated to the arc A^p. This condition defines the second parabolic arc, enabling the computation of the intersection of the two last parabolic arcs. From this intersection we determine the durations t_2 and t_3 of the two last segments and then the length x_f using (2.1). More details are given in appendix A.

It is worth noting that these geometric constructions are simple, but depend on the relative position of C_i and C_f, in particular, the arc A^p of parabola can be the first or the last segment.

This procedure defines the duration $t_f(t_n) = t_n + t_2 + t_3$ and the length $x_f(t_n)$ of the trajectory, namely a point of the parametric curve $\mathcal{C}(t_n)$. The repetition of this procedure allows to plot the curve $\mathcal{C}(t_n)$ defining the duration of the trajectory versus its length (Fig. 2.7). A similar procedure is used to plot $\mathcal{C}(t_p)$.

2.2.2.2 Effects of the bounds

The acceleration bounds A_{min} and A_{max} limit the increase of the time parameters t_n and t_p and the arc A^p to some time t_j defining the intersecting point of the jerk parabolic trajectory with the boundary line. From this point, we follow the same

process, but by defining the intermediate parabola from the end of the acceleration segment of time t_s. To keep the same parameter t_n (resp. t_p) we define t_s by $t_s + t_j = t_n$ (resp. $t_s + t_j = t_p$).

Similarly, when the intermediate parabola reaches the velocity bound V_{min} or V_{max}, we define the duration of the constant velocity segment t_v by $t_v + t_a + t_j = t_n$ (resp. t_p) where t_a is the duration of the constant acceleration segment reaching the limit velocity parabola. In some cases, the parabola defined by t_j can reach the minimal or maximal velocity parabolas before the acceleration bounds: the geometric constructions are similar, but t_a is null.

These bounds introduce changes in the nature of the portions of the curves $\mathcal{C}(t_n)$ and $\mathcal{C}(t_p)$ (See Fig. 2.7), in particular the large values of t_n (resp. t_p) correspond to straight line segments associated to constant (minimum or maximum) velocity motions. The second parabola can also reach the second acceleration bound, resulting in a change in the trajectory sequence without affecting the parameterization of the curve.

2.2.3 The time-optimal trajectories

In the previous case (Fig. 2.7), the parametric curves $\mathcal{C}(t_n)$ and $\mathcal{C}(t_p)$ give directly the time-optimal function $\mathcal{C}_{opt}(x_f)$ that associates the optimal time to the length x_f. In general, the non-linearities in the definition of these functions generate far more complex curves. We will now consider two types of non-linearities: the influence of the velocity that distorts the curve and the presence of shortcuts that split the curve in two.

2.2.3.1 Influence of the velocity

Having now defined a tool to plot the time-optimal function versus the length of the motion, we can study the influence of the different parameters. Considering the case of the Fig. 2.7, we shift the initial and final velocities by 15 (from (-5, 5) to (10, 20)). This shift just translate the phase diagram to the right, but the parametric curves $\mathcal{C}(t_n)$ and $\mathcal{C}(t_p)$ plotted in the Fig. 2.8 have now a more complex shapes and, for a range of values of x_f, there are multiple associated trajectories with different time t_f. Therefore the time-optimal function $\mathcal{C}_{opt}(x_f)$ is no more directly defined by the union of the two curves, but by the minimum time for each value of x_f. The resulting curve may exhibit discontinuities, which can impact the computation of the time-optimal trajectory.

For example such discontinuity is present in the case of Fig. 2.8 at $x_f = \lambda$. For $x_f < \lambda$ the optimal solution begin with a jerk negative segment, but for $x_f >= \lambda$ the optimum trajectory begins with a positive jerk. The corresponding trajectories and their derivatives are plotted in the Fig. 2.9.

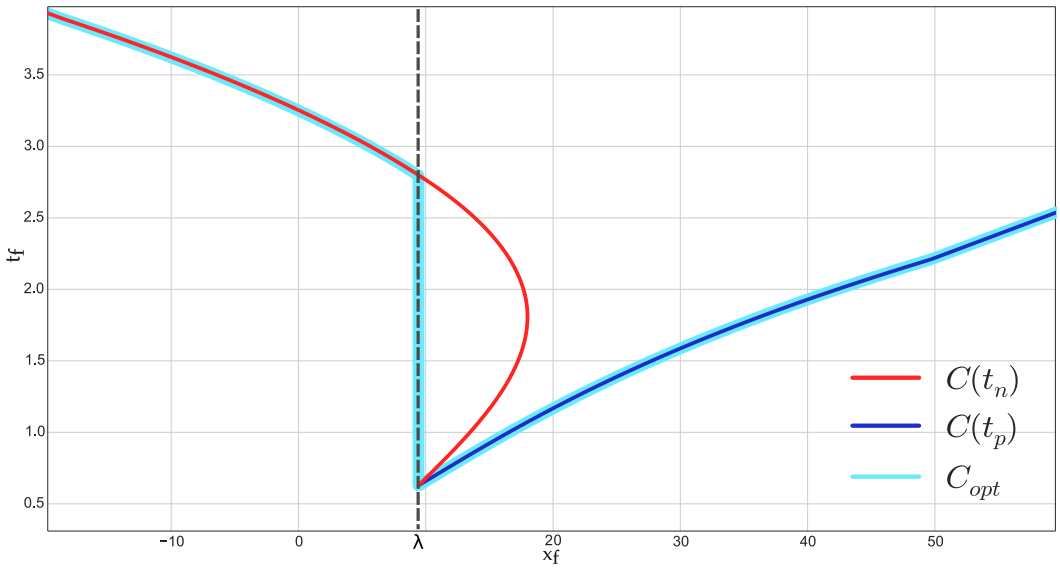


Figure 2.8: A shift of the initial and final velocities distort the curves $\mathcal{C}(t_n)$ and $\mathcal{C}(t_p)$. It can be noted that the curve $\mathcal{C}_{opt}(x_f)$ has a discontinuity for $x_f = \lambda$. ($J_{min}=-50$, $J_{max}=50$, $A_{min}=-30$, $A_{max}=30$, $V_{min}=-30$, $V_{max}=30$, $a_i=8$, $v_i=10$, $a_f=8$, $v_f=20$).

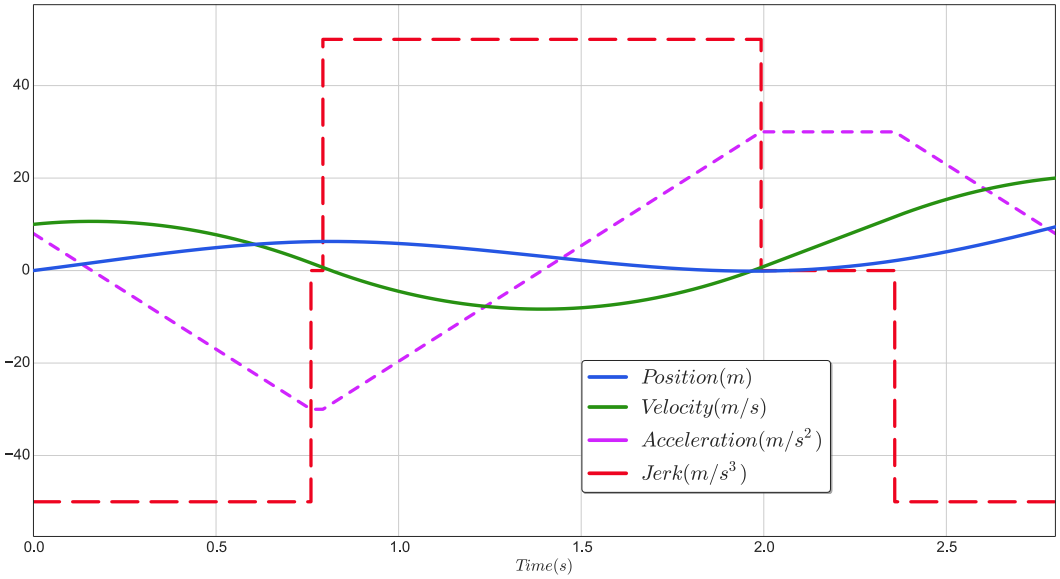
2.2.3.2 Duration in the presence of shortcut

The shortcut solutions associated with intersecting parabolas (Sec.2.2.1.5) also introduce discontinuities in the optimal function $\mathcal{C}_{opt}(x_f)$. The curve of the Fig. 2.10, plotted from a similar case of the Fig. 2.3, shows a small lens shaped curve just below the cusp point of the parametric curves. Beside the main curve ($\mathcal{C}(t_n)$ and $\mathcal{C}(t_p)$) obtained by the previous procedure, the two parts ($\mathcal{C}(tl_n)$ and $\mathcal{C}(tl_p)$) of the lens shaped curve are similarly plotted from the two shortcut trajectories $(C_i, C_{in}, C_{fn}, C_f)$ and $(C_i, C_{ip}, C_{fp}, C_f)$, where the parameters tl_n and tl_p define respectively the duration of the first segments (C_i, C_{in}) and (C_i, C_{ip}). We depict on Fig. 2.11 and Fig. 2.12 the effect of the discontinuities of the situation presented in Fig. 2.13 on the time-optimal trajectory.

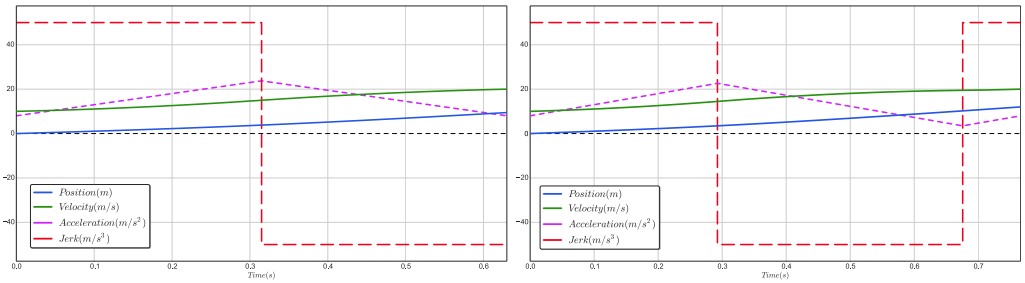
In the neighborhood of the singular case where the interior parabolas become tangential, the lens shaped curve joins the main curve and disappears to extend the main curve on both sides (Fig. 2.13). It can be noted that for some values of x_f, the curve exhibit up to five solutions (Fig. 2.13).

We have seen the influence of the initial and final velocity and acceleration on the optimal curve $\mathcal{C}_{opt}(x_f)$. The jerks J_{min} and J_{max} deform also the curve, but do not introduce new particular case. The values of the bounds influence the shape of $\mathcal{D}$, which change the nature of the function defining the curve $\mathcal{C}_{opt}(x_f)$, but do not introduces new types of discontinuities.

Considering only the jerk bounds, it is possible to compute analytically the



(a) Solution in 5 segments for $x_f = 9.44920 < \lambda$



(b) Direct trajectory for $x_f = \lambda = 9.4492105$

(c) Solution in 3 segments $x_f \gg \lambda$

Figure 2.9: Major effects of discontinuities on the time representation of the optimal trajectory and its derivative.

parametric curve $\mathcal{C}(x_f(t_n), t_f(t_n))$ (See Appendix A). The expression obtained is large and complicated and generates an even more complicated derivative. Unfortunately, we could not manage to analytically compute the zero of the derivative of $\mathcal{C}_{opt}(x_f)$ with respect to x_f associated to its points of discontinuity.

In the algorithm presented below, we chose to compute all the solutions (up to five) and then select the optimal one. An alternative approach would be to numerically compute the zero of the derivative of $\mathcal{C}(t_n)$ and $\mathcal{C}(t_p)$ relatively to x_f and then compute directly the optimum.

2.2.3.3 The time-optimal algorithm

From the previous elements, several strategies are possible to compute the time optimum trajectory $T_{opt}(x_f)$. However, as our main motivation is real-time control, we propose now a fast algorithm:

1. Compute the local parabolas.

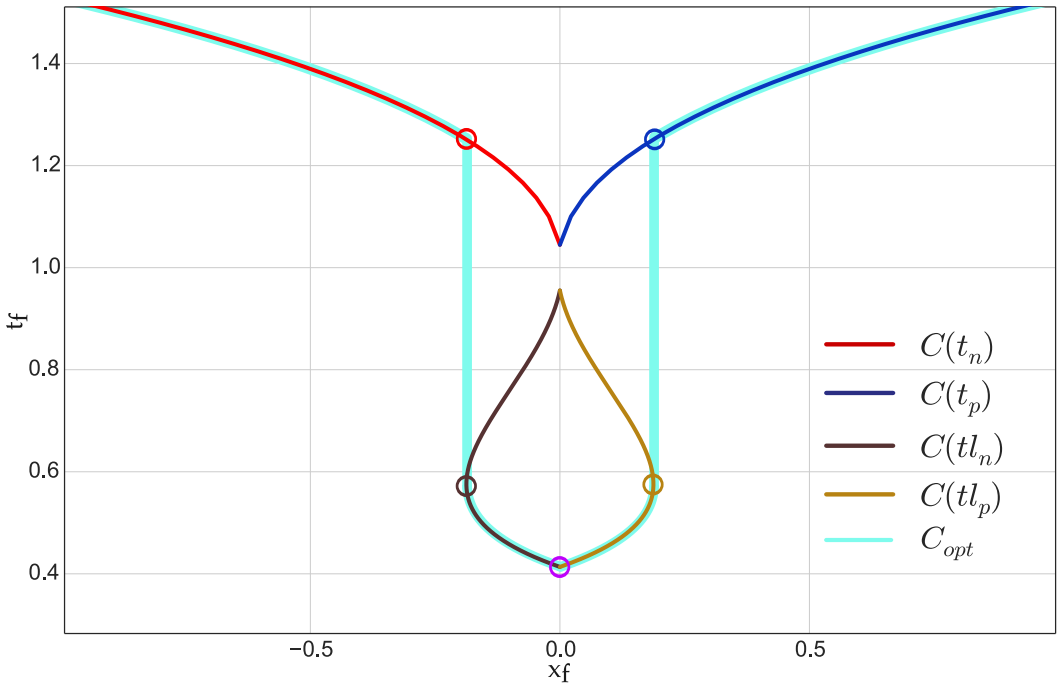


Figure 2.10: Shortcut trajectories introduce discontinuities in the time-optimal curve $\mathcal{C}_{opt}$. The color circles are placed on the sides of the discontinuities that are depicted in the phase diagram in Fig. 2.11, 2.12. (J_{min}=−40, J_{max}=40, A_{min}=−30, A_{max}=30, V_{min}=−30, V_{max}=30, a_i=20, v_i=−4.99, a_f=20, v_f=4.99).

2. Compute the singular limit trajectories.

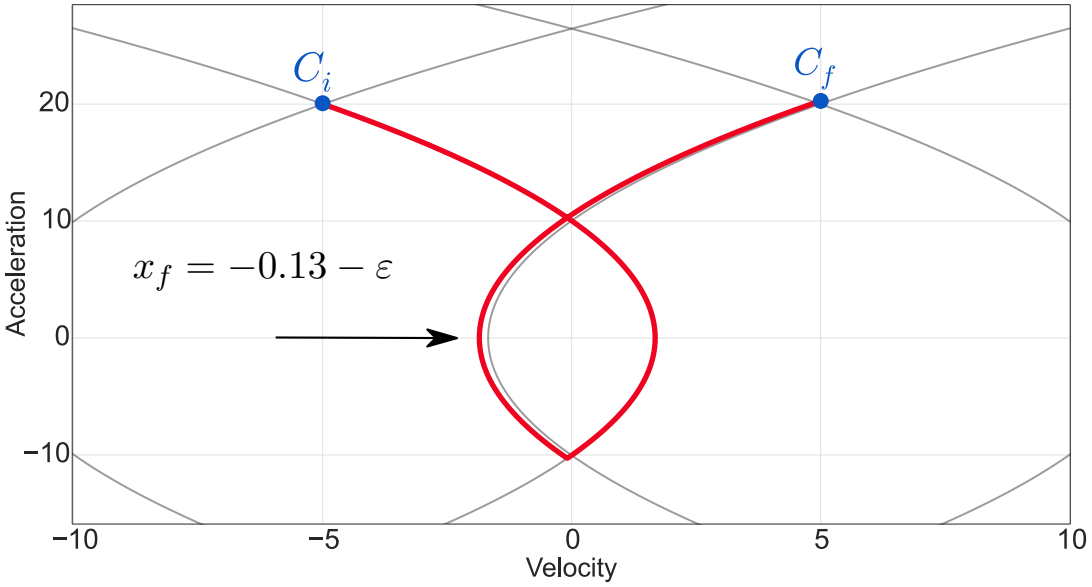
3. From x_f, determine what are the possible sequences of trajectories.

4. For each possible sequence compute the time optimum trajectory.

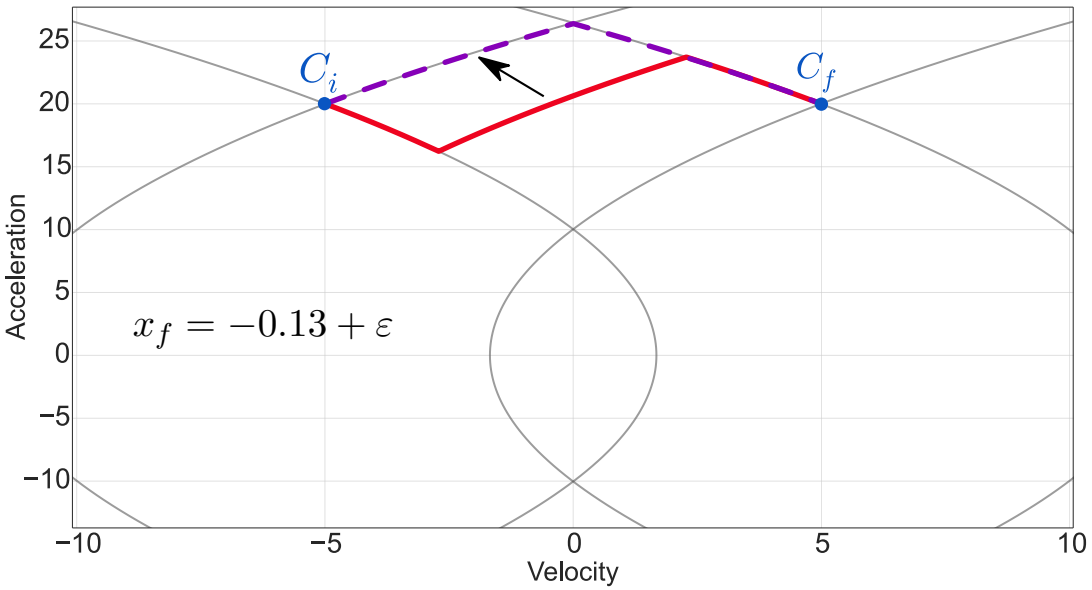
5. Select the faster trajectory.

The first three stages of this algorithm were previously detailed and the last one is trivial, therefore the following sections will detail how to compute a sequence of trajectory segments from the length x_f. We begin by showing how to transform the associated systems of equations in a quartic polynomial equation and then we will detail how to solve these quartic equations.

The seven segment trajectory labeled as (JAJVJAJ) doesn't generally include all the seven segments. When the V segment exists, it is the only one for which the duration is varying accordingly with x_f. If the V segment is not reached, the sequence comprises at most five elements (JAJAJ). This problem can always be reduced further to a sequence of three segments: Each time an acceleration segment is reached, the first or the last jerk segment duration is fixed and defined by $t_j = (A_b - a_i)/J$ where $A_b \in \{A_{min}, A_{max}\}$ and $J \in \{J_{min}, J_{max}\}$. Therefore



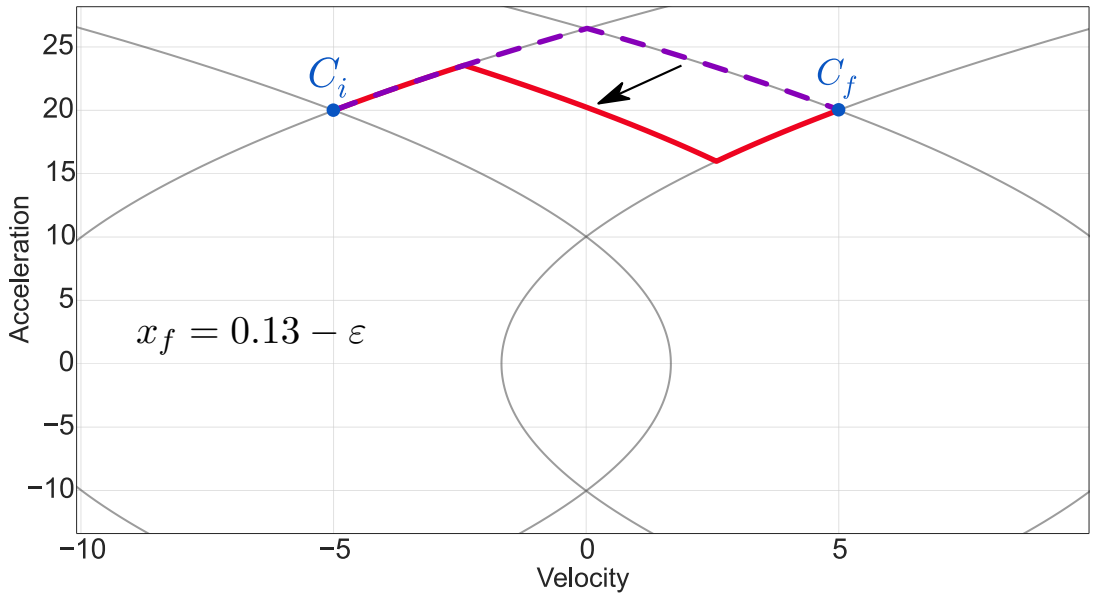
(a) Negative loop trajectory corresponding to the situation depicted by the red circle on the time-optimal curve $\mathcal{C}_{opt}$ of Fig.2.10. If x_f is increased, the shape of the time-optimal trajectory change drastically (see (b)) as there is a discontinuity on $\mathcal{C}_{opt}$.



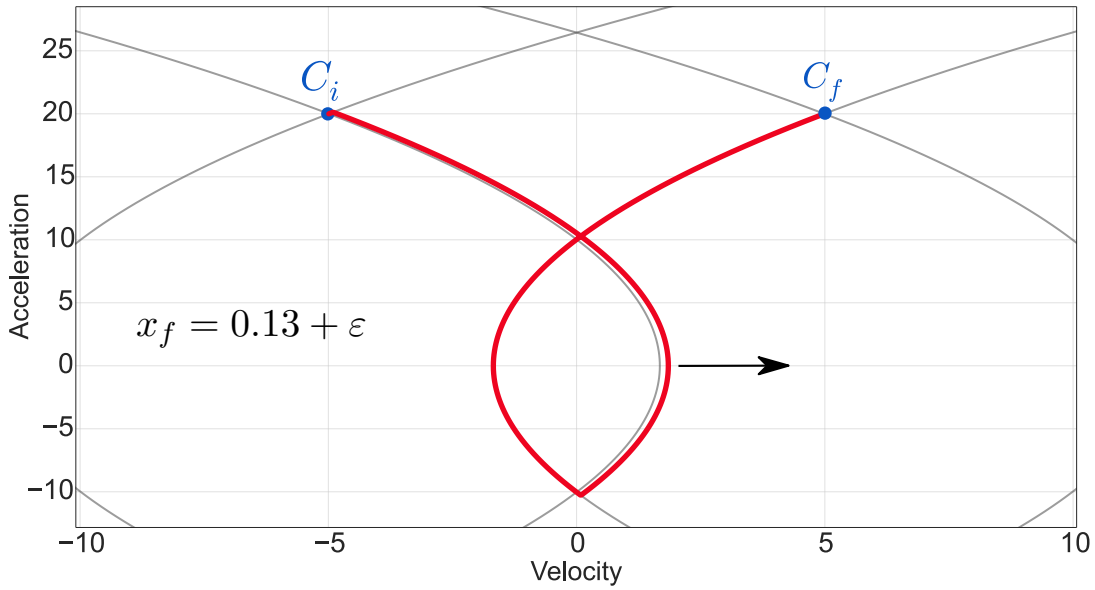
(b) Negative shortcut trajectory corresponding to the situation depicted by the brown circle on the time-optimal curve $\mathcal{C}_{opt}$ of Fig. 2.10. By increasing x_f we follow the lens shaped curve of $\mathcal{C}_{opt}$ until the direct trajectory depicted in purple here and by the purple circle in (a).

Figure 2.11: The discontinuity of the time-optimal curve $\mathcal{C}_{opt}$ for a negative x_f is reflected on the shape of the trajectory in the phase diagram.

after simplification, the four problems left to solve are: (JJJ) when no acceleration bounds are reached, (AJJ) or (JJA) when only one acceleration bound is reached and (AJA) when both the acceleration bounds are reached.



(a) Positive shortcut trajectory corresponding to the situation depicted by the golden circle on the time-optimal curve $\mathcal{C}_{opt}$ of Fig. 2.10. From the direct trajectory we increased x_f until we reach a discontinuity on $\mathcal{C}_{opt}$. By increasing x_f more we leave the lens shaped curve of 2.10 and the trajectory will now be a positive loop that can be extended indefinitely. See (b).



(b) Positive loop trajectory corresponding to the situation depicted by the blue circle on the time-optimal curve $\mathcal{C}_{opt}$ of Fig. 2.10.

Figure 2.12: The discontinuity of the time-optimal curve $\mathcal{C}_{opt}$ for a positive x_f is reflected on the shape of the trajectory in the phase diagram.

The next paragraph details the reduced problem, that is, when it is reduced to a sequence of three segments (Sec.2.2.4). The (v) case is trivial because only one segment is varying.

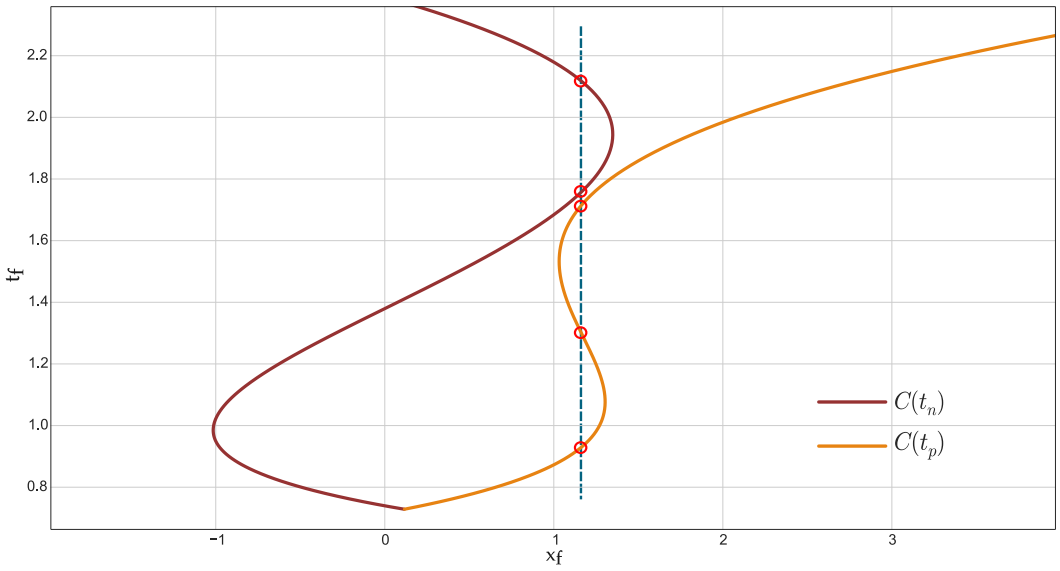


Figure 2.13: Curve exhibiting up to 5 solutions when initial and final velocities are shifted. (J_{min}=−40, J_{max}=50, A_{min}=−55, A_{max}=50, V_{min}=−40, V_{max}=70, a_i=−39.0, v_i=17.205, a_f=−39.0, v_f=−17.105).

2.2.4 Solving the reduced problem

2.2.4.1 Solving the JJJ problem

After reducing the problem, the three jerk trajectories problem (JJJ) is defined by seven parameters: the initial condition (a_i, v_i), the final condition (a_f, v_f, x_f) and two jerks J_a and J_b. The unknowns are the durations of the three segments (t_1, t_2, t_3). Two conditions (a_1, v_1, x_1) and (a_2, v_2, x_2) are associated to the transitions between the segments and defined by:

$$\begin{aligned} a_1 &= J_a * t_1 & +a_i & & \\ v_1 &= J_a * {t_1}^2/2 & +a_i * t_1 & +v_i & \\ x_1 &= J_a * {t_1}^3/6 & +a_i * {t_1}^2/2 & +v_i * t_1 & \end{aligned} \tag{2.3}$$

$$\begin{aligned} a_2 &= J_b * t_2 & +a_1 & & \\ v_2 &= J_b * {t_2}^2/2 & +a_1 * t_2 & +v_1 & \\ x_2 &= J_b * {t_2}^3/6 & +a_1 * {t_2}^2/2 & +v_1 * t_2 & +x_1 \end{aligned} \tag{2.4}$$

Then the system of polynomial equations to solve can be written as:

$$\begin{aligned} a_f &= J_a * t_3 & +a_2 & & \\ v_f &= J_a * {t_3}^2/2 & +a_2 * t_3 & +v_2 & \\ x_f &= J_a * {t_3}^3/6 & +a_2 * {t_3}^2/2 & +v_2 * t_3 & +x_2 \end{aligned} \tag{2.5}$$

Using algebra systems like Maple[2] and Maxima[3], the solution of this system of equation can be expressed from the roots of a quartic equation. Firstly we define a set of intermediate variables (k_1 to k_4) and the coefficients of the polynomial equation:

$$
\begin{aligned}
k_1 &= 2 * J_b - J_a \\
k_2 &= {J_b}^2 + J_a * (J_a - 2 * J_b) \\
k_3 &= J_a - J_b \\
k_4 &= 2 * J_a * v_f \\
c_4 &= -J_b * ({J_b}^3 + J_a * (J_a * (5 * J_b - 2 * J_a) - 4 * {J_b}^2)) \\
c_3 &= 0 \\
c_2 &= -6 * (2 * J_a * k_2 * v_i - {J_b}^2 * {a_i}^2 \\
&\quad + J_a * (k_1 * {a_i}^2 + 2 * k_2 * v_f) + {a_f}^2 * (J_a * k_1 - {J_b}^2)) \\
c_1 &= -8 * (a_i * (3 * J_a * J_b * v_i - (3 * {J_a}^2 * v_i + J_b * {a_i}^2)) \\
&\quad + J_a * {a_i}^3 - 3 * J_a * k_3 * x_f) \\
&\quad + a_f * (3 * J_a * k_3 * v_f - {a_f}^2 * k_3)) \\
c_0 &= 3 * (4 * J_a * v_i * (J_a * v_i - ({a_i}^2 + k_4 - {a_f}^2)) + {a_i}^4 \\
&\quad + 2 * ((k_4 - {a_f}^2) * {a_i}^2 + 2 * {J_a}^2 * v_f * v_f) \\
&\quad + {a_f}^2 * ({a_f}^2 - 4 * J_a * v_f))
\end{aligned}
$$

The quartic polynomial equation is then defined by: $c_4 * x^4 + c_2 * x^2 + c_1 * x + c_0 = 0$. If r_i is one of its roots, the solution can be written as:

$$
\begin{aligned}
t_1 &= -(k_6 * (k_5 - {a_i}^2 + 2 * (J_a * (r_i * a_i - v_f) \\
&\quad - Jb * r_i * a_i) + k_3 * J_b * {r_i}^2 + {a_f}^2)) * 0.5 \\
t_2 &= r_i \\
t_3 &= (k_6 * (k_5 - ({a_i}^2 + k_4) - J_b * k_3 * {r_i}^2 \\
&\quad + a_f * (2 * k_3 * r_i + a_f))) * 0.5
\end{aligned}
$$

with:

$$
\begin{aligned}
k_5 &= 2 * J_a * v_i \\
k_6 &= 1/(J_a * k_3 * r_i)
\end{aligned}
$$

As the times t_1, t_2 and t_3 must be positive, only the positive solutions define a valid trajectory. This system can have up to four solutions, but we never find a particular case with more than three admissible triplets. It must be noted that the trajectories can begin with one of the two jerk bounds, defining two different

[2] https://fr.maplesoft.com/

[3] http://maxima.sourceforge.net/

problems, which defines up to three solutions each. So, in some cases like the one of the Fig. 2.13, five different trajectories composed of a sequence of potentially optimum segments can be computed.

The last step is to compute the time-optimal solution, which is the one that minimize $t_1 + t_2 + t_3$.

2.2.4.2 Solving the AJA problem

In this case the jerk J_a of the first and last segments is null in the equations (Eq.2.3) and (Eq.2.5) that respectively become (Eq.2.6) and (Eq.2.7):

$$\begin{aligned} a_1 &= a_i \\ v_1 &= a_i \times t_1 + v_i \\ x_1 &= a_i \times {t_1}^2/2 + v_i \times t_1 \end{aligned} \tag{2.6}$$

$$\begin{aligned} a_f &= a_2 \\ v_f &= a_2 \times t_3 + v_2 \\ x_f &= a_2 \times {t_3}^2/2 + v_2 \times t_3 + x_2 \end{aligned} \tag{2.7}$$

Using an algebra system, we obtain the results:

$$\begin{aligned} k_1 &= a_i - a_f \\ k_2 &= {k_1}^{-1} \\ k_3 &= a_i \times ((-J_a \times v_i) + {a_f}^2/2 + a_i \times (-a_f + a_i/2)) \\ k_4 &= a_f \times J_a \times v_i \\ k_5 &= 12 \times {J_a}^2 \times {v_i}^2 \\ k_6 &= a_f - a_i \\ k_7 &= (3 \times (12 \times a_i \times {J_a}^2 \times (2 \times a_f \times k_6 \times x_f + k_1 \times {v_f}^2) \\ &\quad + a_f \times (a_i \times (a_i \times (4 \times {a_f}^3 + a_i \times (a_i \times (4 \times a_f - a_i) \\ &\quad - 6 \times {a_f}^2)) - (k_5 + a_f^4)) + 12 \times a_f \times {J_a}^2 \times {v_i}^2)))^{0.5} \\ k_8 &= k_6 \times {J_a}^{-1} \\ k_9 &= k_1 \times a_f \\ k_{10} &= 2 \times J_a \times v_f \\ k_{11} &= 1/\sqrt{3} \\ k_{12} &= (k_6 \times (12 \times a_i \times {J_a}^2 \times (2 \times a_f \times x_f - {v_f}^2) \\ &\quad + a_f \times (k_5 + a_i \times (a_i \times (3 \times {a_f}^2 + a_i \times (a_i - 3 \times a_f)) \\ &\quad - {a_f}^3))))^{0.5} \end{aligned}$$

And finally the system has two solutions (t_1, t_2, t_3) and (t'_1, t_2, t'_3) with:

$$
\begin{aligned}
t_1 &= {a_i}^{-1} \times k_2 \times {J_a}^{-1} \times ((-k_7/6) + k_4 + k_3) \\
t_2 &= k_8 \\
t_3 &= ({a_f}^{-1} \times {J_a}^{-1} \times (k_{11} \times k_2 \times k_{12} + k_{10} + k_9))/2 \\
t'_1 &= {a_i}^{-1} \times k_2 \times {J_a}^{-1} \times (k_7/6 + k_4 + k_3) \\
t'_3 &= {a_f}^{-1} \times {J_a}^{-1} \times ((k_{10} + k_9)/2 - (k_{11} \times k_2 \times k_{12})/2)
\end{aligned}
$$

2.2.4.3 Solving the JJA problem

The system of polynomial equations to solve is defined by (Eq.2.6), (Eq.2.4) and (Eq.2.5). Using an algebra system, we obtain the following result where r_i is one solution of the quartic equation defined by the coefficients c_i with $0 \leq i \leq 4$:

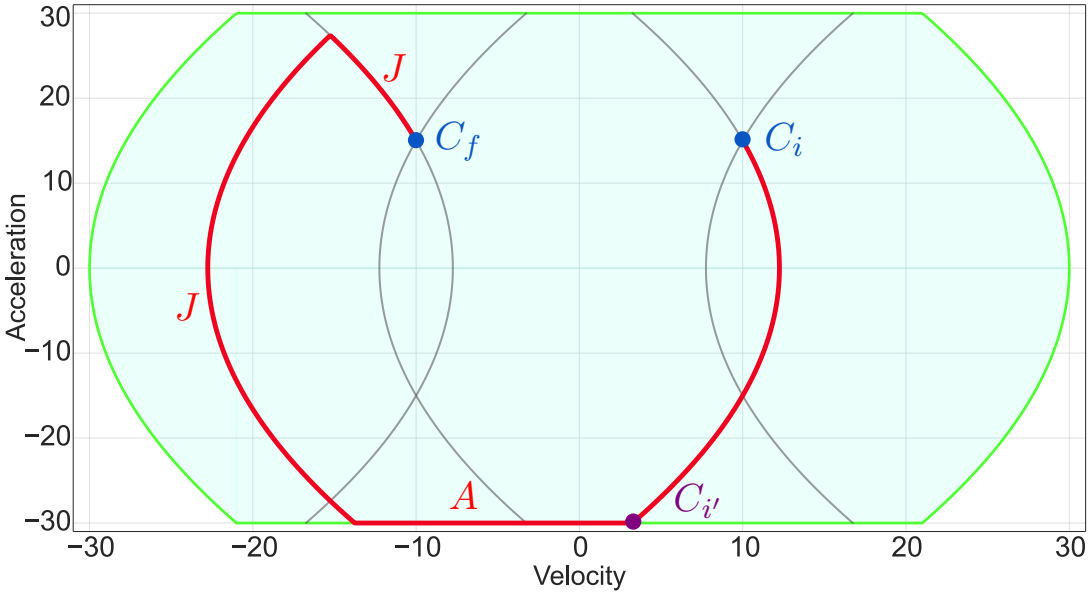
$$
\begin{aligned}
k_1 &= 1 - 2 \times a_f \\
k_2 &= 2 \times a_f \\
k_3 &= {a_f}^2 \times J_b + k_1 \times J_a \\
c_4 &= 3 \times {a_f}^4 \times {J_b}^4 \\
&\quad + J_a \times (2 \times k_1 \times {a_f}^2 \times J_b + (4 \times (a_f - 1) \times a_f + 1) \times J_a)) \\
c_3 &= 4 \times a_f \times J_b \times ({a_f}^3 \times {J_b}^2 \\
&\quad + J_a \times ((3 \times a_f - 1) \times J_a - 3 \times {a_f}^2 \times J_b)) \\
c_2 &= 6 \times J_b \times (2 \times J_a \times k_3 \times v_i + ((k_2 - 1) \times J_a - {a_f}^2 \times J_b) \times {a_i}^2) \\
c_1 &= 12 \times a_f \times (2 \times J_a \times (a_f \times J_b - J_a) \times v_i + (J_a - a_f \times J_b) \times {a_i}^2) \\
c_0 &= 3 \times (4 \times {J_a}^2 \times {v_i}^2 + a_i \times (4 \times J_a \times (k_2 - a_i) \times v_i + {a_i}^3)) \\
&\quad + 4 \times (3 \times {J_a}^2 \times (2 \times a_f \times x_f - {v_f}^2) - 2 \times a_f \times {a_i}^3) \\
t1 &= -{J_a}^{-1} \times (a_i + a_f \times J_b \times r_i) \\
t2 &= r_i \\
t3 &= -\frac{Ja^{-1} \times (2 \times Ja \times v_i - ({a_i}^2 + 2 \times Ja \times v_f) + J_b \times k_3 \times {r_i}^2)}{2 \times a_f}
\end{aligned}
$$

2.2.4.4 Solving the AJJ problem

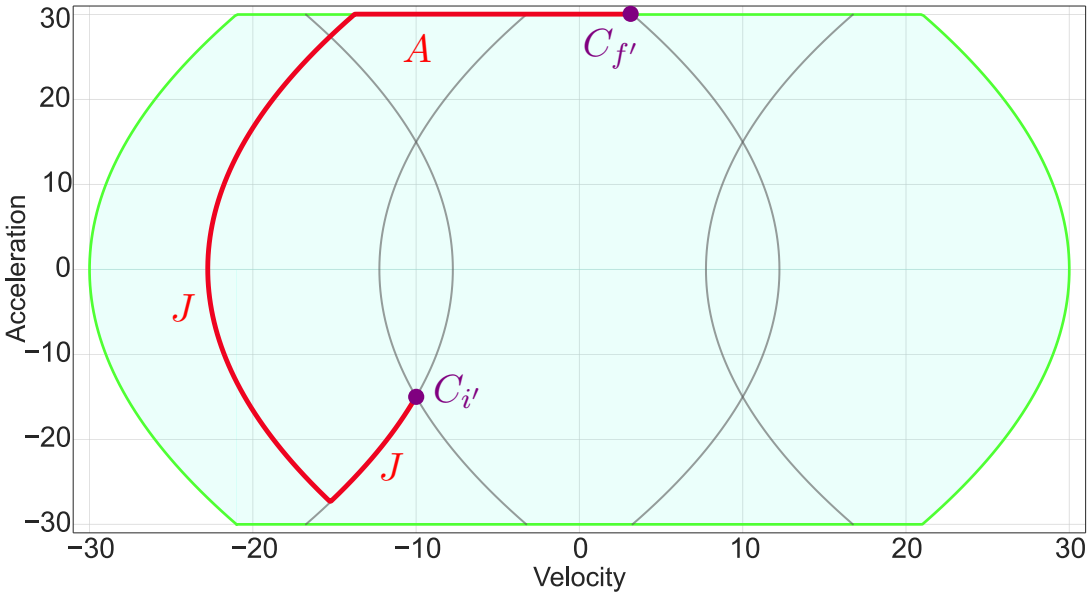
An (AJJ) system can be solved similarly as a (JJA) one or using a symmetry with respect to the acceleration axis to build an equivalent (JJA) system (see Fig. 2.14).

2.2.5 Solving the quartic polynomial equation

It is well known that solving a quartic polynomial equation is difficult. The analytical solutions have been known since the 16th century, but this approach is time consuming and can fail for some particular equations. Numerical algorithms like Newton-Raphson based algorithms are efficient, but require initial information about the root, precisely the information we do not have in our case. Recent works



(a) The problem here can be reduced to a (AJJ) problem as the first jerk segment is fixed.



(b) The (AJJ) sequence presented in (a) can be transformed by symmetry according to the acceleration axis. The symmetric of $C_{f'}$ becomes the $C_{i'}$ of the (JJA) sequence, and the same can be applied to $C_{i'}$ that becomes $C_{f'}$ after transformation.

Figure 2.14: A (AJJ) sequence can be transformed into a (JJA) by use of symmetry. The solution presented in 2.2.4.3 can then be used to solve the (AJJ) problem.

have proposed to associate the two approaches: the analytical results are used as inputs for a numerical solver [Strobach 2010]. This has generated a new class of faster and more accurate algorithms [Flocke 2015, Strobach 2015].

We used a solver derived from the one of Schwarze [Schwarze 1990] to compute

Table 2.2: Comparison of the mean computation times for different trajectory lengths. Results obtained for 5×10^5 runs.

Case	General	Near the direct trajectory	With cruising velocity phase
Times (μs)	1.04	2.45	0.854

a first approximation of the solution. To improve the accuracy, we directly applied a three dimensional Newton method to the durations of the three segments. The analytical expression of the derivative of the polynomial functions (Eq.2.1) is given in appendix B.

2.2.6 Discussion

The characteristics of our method are summarized in the classification table 2.1, which compare the possibilities of the online trajectory generators. The implementation[4] of this algorithm on a system equipped with a Intel Core i7 processor running at 2.2 GHz gives a mean time of 1.02 μs with a standard deviation of 0.81 μs observed for 10^8 random tests, allowing to use it in real time and for planning (See table 2.2). These performances are comparable or better than the previous algorithms that do not always give the optimal solution.

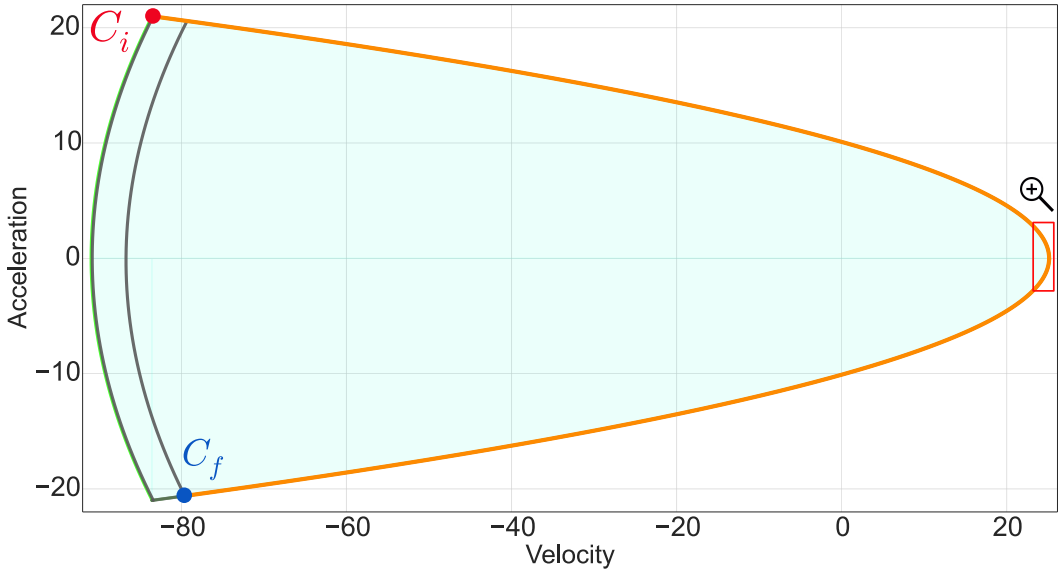
For the multi-dimensional case, computation times are given in Sec.2.3.2 and a comparison with similar works is made.

The longest times are relative to particular case where the Newton-Raphson method have difficulty to improve the accuracy of the solution. The figure 2.15 gives an example of such a configuration where the analytical solution is not precise enough and the numerical one struggle to converge. Fortunately these cases are hardly relevant. We can see it in Fig. 2.15 which present a heavy asymmetric configuration. Anyway, the system always returns a solution for such configurations, eventually a sub-optimal solution.

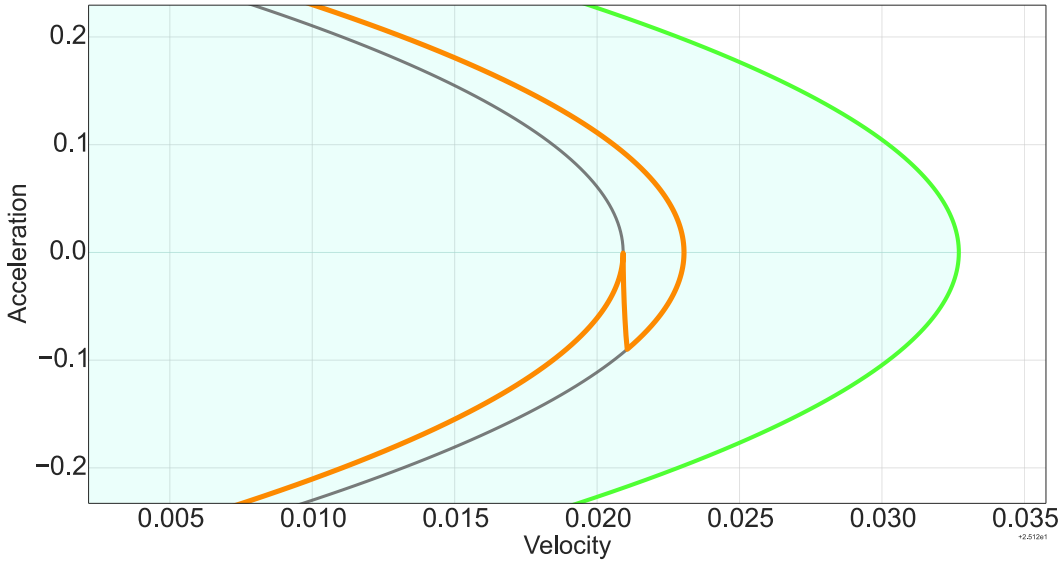
Solving the optimal trajectory problem in the vicinity of the direct trajectory opens the way to an intensive use of these simple trajectories for control and planning. Concerning the trajectory control, where the objective is to compute in real time a trajectory to bring back smoothly the mobile to the target trajectory from the current state, we can notice that the connection trajectories are short and consequently close to the direct trajectories. In this case, the proposed trajectory generator provides a good solution. Similarly for multi-axis trajectory generation, one classical solution is to compute the time optimal trajectory for each axes, select the slowest one and synchronize the other axes with the selected one. The proposed trajectory generator can improve the calculation of the time optimal trajectories.

Sampling-based motion planners are really efficient to find a polygonal path,

[4]The documentation, the softMotion library and examples are available at `https://git.openrobots.org/projects/softmotion/wiki`.



(a) In this case parabolas are almost identical.



(b) The zoomed area where the positive cusp produces a very short jerk segment.

Figure 2.15: Illustration of a difficult case, where the Newton algorithm has difficulty to converge. ($J_{min}=-2.02754$, $J_{max}=29.7968$, $a_i=20.9815$, $v_i=-83.4179$, $a_f=-20.6076$, $v_f=-79.5853$).

even in the case of cluttered and high-dimensional space, but planning efficient and smooth trajectories is more difficult. Here also the proposed trajectory generator could improve the smoothing of an initial trajectory built from the polygonal path.

The generator can be used in joint or operational spaces. In the first case kinematic bounds can be directly deduced from joint characteristics, whereas Cartesian space is suitable to incorporate the constraints related to safety and ergonomics.

The non-symmetrical bounds can be employed to enhance the ergonomics properties of our model as they allow designing more natural human-like motions. They can also be very useful in the making of vertical motions under gravity, or motions in the presence of a human, for which an approaching move is more scaring than a withdrawal move.

As this work explains the discontinuities of the time-optimal curve $\mathcal{C}_{opt}$ and solves the optimum time problem, it will contribute to the development of trajectory based robotic architectures. In these architectures, trajectories will be used as the main support of communication and facilitate the link between planning and control, leading towards an improvement of robots motions.

By explaining the complex behavior of the jerk bounded trajectories, this work defines also a step in the solving of the snap bounded optimal trajectory problems, where the snap is the derivative of the jerk.

2.3 Multi-dimensional Trajectories

In section 2.2 we presented the general algorithm to generate third degree polynomial point to point trajectories in the mono-dimensional case. This work was extended for systems with multiple degrees of freedom. This extension requires the synchronization of the mono-dimensional trajectories generated for each axis. This is a well known problem that has been addressed in multiple works [Biagiotti 2008, Frisoli 2013, Blaha 2014, Kroger 2011, Kroger 2010, Broquère 2011].

We generally distinguish three level of synchronization :

- Phase synchronization.
- Time synchronisation.
- Independent axes.

Phase synchronization or full synchronization is a particular type of time synchronization used to generate homothetic trajectories that are 1D straight lines in a multidimensional space. This is the most interesting kind of multi-axis synchronization, as in a variety of tasks, robotics systems will need to generate straight line motions. However for the vast majority of situations, only rest to rest motions allows to obtain this type of synchronization. Then for every other situation that requires non-null initial or final conditions, time synchronization is employed. Even if time synchronization doesn't generally generate straight-line motions, it is generally close to it for long motions. These two types of synchronization present similar computation costs, but complying with the constraints may be more difficult for time-synchronization, so we choose the phase synchronization when it is possible. When the time-synchronization is not possible, each axis evolves independently, and an independent mono-dimensional trajectory is generated for each axis.

2.3.1 Notations

First lets use $\mathcal{K}$ to denote the number of axis to synchronize. The synchronized trajectories are defined on a time-interval $[t_I, t_F]$.

$\mathcal{X}_t^k$, $\mathcal{V}_t^k$, $\mathcal{A}_t^k$, $\mathcal{J}_t^k$ such that $t \in [t_I, t_F]$ and $k \in [1, \mathcal{K}]$ are the position, velocity, acceleration and jerk at any time instant t along the synchronized axis k. $\mathcal{J}_I$, $\mathcal{A}_I$, $\mathcal{V}_I$, $\mathcal{X}_I$ are used for the vectors of initial jerks, accelerations, velocities and positions. Similarly for the final motion state we use: $\mathcal{J}_F$, $\mathcal{A}_F$, $\mathcal{V}_F$, $\mathcal{X}_F$.

$\mathcal{B}_k = (J_{min}^k, J_{max}^k, A_{min}^k, A_{max}^k, V_{min}^k, V_{max}^k)$ with $k \in [1, \mathcal{K}]$ defines the vectors of motion bounds.

2.3.2 Phase synchronized trajectories

As mentioned full synchronization is the most desired kind of multi-axis synchronization in most robotic applications as it generates straight 1-D lines in multi-dimensional spaces. We obtain an homothetic trajectory by synchronizing mono-dimensional trajectories such that at any time instant $t \in [t_I, t_F]$ the following property holds :

$$\frac{\mathcal{X}_t^k - \mathcal{X}_I^k}{\mathcal{X}_t^j - \mathcal{X}_I^j} = \lambda \qquad \forall j, k \in [1, \mathcal{K}], \qquad t \in [t_I, t_F] \tag{2.8}$$

Put differently, this means that at any instant of time, each trajectory has completed the same percentage of their respective length. An example of phase synchronized trajectory is plotted in Fig. 2.16. The phase synchronization is only available if :

$$\mathcal{A}_I, \mathcal{V}_I, \mathcal{A}_F, \mathcal{V}_F \quad \text{and} \quad (\mathcal{X}_F - \mathcal{X}_I) \quad \text{are collinear.} \tag{2.9}$$

This is why phase synchronized trajectories are often considered for rest to rest motion only, as this condition is hardly encountered with general initial and final conditions. If the condition of collinearity is verified, the time optimal trajectory is computed along the line after projecting on it all the constraints and have selected the most restrictive ones. For that we have to compute the scaling factors λ_k. These scaling factors scale the initial motion bounds vectors $\mathcal{B}_k$ into a new one ${}^{new}\mathcal{B}_k$ such that ${}^{new}\mathcal{B}_k$ will also be collinear with (Eq.2.9).

Then we have :

$$\lambda_k = \frac{\mathcal{X}_F^k - \mathcal{X}_I^k}{\mathcal{X}_F^s - \mathcal{X}_I^s} \tag{2.10}$$

$\mathcal{B}_k$ is then adjusted into ${}^{new}\mathcal{B}_k$ like so :

$$
\begin{aligned}
{}^{new}V^k_{max} &= \lambda_k V^k_{max} \\
{}^{new}V^k_{min} &= \lambda_k V^k_{min} \\
{}^{new}A^k_{max} &= \lambda_k A^k_{max} \\
{}^{new}A^k_{min} &= \lambda_k A^k_{min} \\
{}^{new}J^k_{max} &= \lambda_k J^k_{max} \\
{}^{new}J^k_{min} &= \lambda_k J^k_{min}
\end{aligned} \tag{2.11}
$$

The phase-synchronized trajectory we obtain is optimal in time.

The generation of phase synchronized trajectories for $\mathcal{K} = 7$ on a system equipped with a Intel Core i7 processor running at 2.2 GHz gives a mean time of 23.77 μs with a standard deviation of 3.22 μs observed for 10^5 random tests. In [Frisoli 2013] the mean time for the synchronization of four joints is 150 μs. In [Kroger 2011] the average execution time for a 6-DOF system is 135 μs.

2.3.3 Time synchronized trajectories

Time synchronization can be used even if (Eq.2.9) is not respected. However this synchronization is easier to obtain for rest to rest motions [Blaha 2014]. The common method described in [Blaha 2014, Biagiotti 2008, Broquère 2011] is similar to the phase-synchronization one. The first step is to find the slowest axis that will impose the synchronization time $t_F = max(t^k_f)$. Then we have to compute the set of time-scaling factors ϕ_k:

$$
\phi_k = \frac{t^k_f}{t_F} \tag{2.12}
$$

Time-scaling is done by scaling $\mathcal{B}_k$ into ${}^{new}\mathcal{B}_k$:

$$
\begin{aligned}
{}^{new}V^k_{max} &= \phi_k V^k_{max} \\
{}^{new}V^k_{min} &= \phi_k V^k_{min} \\
{}^{new}A^k_{max} &= \phi^2_k A^k_{max} \\
{}^{new}A^k_{min} &= \phi^2_k A^k_{min} \\
{}^{new}J^k_{max} &= \phi^3_k J^k_{max} \\
{}^{new}J^k_{min} &= \phi^3_k J^k_{min}
\end{aligned} \tag{2.13}
$$

This time-scaling ensure that all axes have the same duration, however the multi-dimensional trajectory is not homothetic. For rest to rest motions, the phase synchronized method depicted in paragraph 2.3.2 should be preferred as it produces straight lines.

It is important to note that time-synchronized trajectories do not provide the ability to control the path because each variation of a parameter modifies the path.

Increasing the travel time opens the way to an infinite number of solutions where the system must calculate a suitable one. Unfortunately, the choice of a solution is generally difficult and depends strongly on the particularities of each specific case. Some solutions are presented in [Kroger 2010, Blaha 2014, Broquère 2011]. An easy one is to time-scale the $\mathcal{K} - 1$ fastest axes to the slowest one with three jerk segments trajectories. Three jerk trajectories are presented in section 3.3.5. This method presents some drawbacks: the jerk on the second segment might not be bounded, or a solution might not exist. However in most situations this method will present a valid solution. The implementation of this method is easy with fast computation times.

2.4 Conclusion

With the emergence of HRI, the problem of the generation of safe, efficient and human-friendly movements must be addressed. The review of the state of the art shows that no complete solution for the making of collaborative motions exists yet. To the best of our knowledge, the algorithm presented herein that uses trajectory of class $\mathcal{C}^2$ defined by a chain of cubic polynomial functions is the first one that:

- joins two arbitrary conditions defined by position, velocity and acceleration,
- in minimum time under general and asymmetric bounds on velocity, acceleration and jerk.

By explaining graphically the behavior of the optimal trajectories, this work allows to explain and solve the difficulties highlighted by the previous works. The proposed trajectory generator completes the existing tools for planning and controlling multi-axis and multi-points cubic polynomial trajectories, which open the way for more flexible and friendly robots. Close cooperation between humans and industrial robots needs more flexibility, adaptability and reusability, which can be improved by the models and tools developed in this work. From an HRI point of view, the constrained jerk model approach makes easier the consideration of the different types of constraints related to safety and ergonomics. All of this allows the robot to adapt its behaviour accordingly to the situation and to propose better interactions.

However we still don't know how the kinematic constraints should be adjusted to respond to any kind of situation. From the state of the art we know that smooth motions are suitable for Human-Robot interactions, and smoothness is acquired by limiting the jerk and may be acceleration. We do not know in details how each kinematic parameter contributes to define qualities and properties of motions. This question will be discussed in Chapter 4 with the presentation of some experiments.

The presented algorithm only computes a trajectory between two admissible configurations. If the system has to reduce the kinematic constraints in reaction to an unforeseen events and plan a new trajectory, the initial configuration will be

outside of $\mathcal{D}$. An extension of this algorithm is presented in Chapter 3 and proposes a solution to this problem.

Even if the model of the chain of cubic polynomial trajectories is efficient, the questions related to higher degrees polynomial still remains. These models are necessary to solve specific problems. The underactuated vehicles, for example, need one more derivative to control the motion obtained by integration. It is the case, for example, to obtain a jerk bounded horizontal move with a quadrotor. The same problem appears also for double and, more generally, multiple pendulum. Given the difficulties encountered to solve the cubic trajectories, the higher degree appears as really challenging.

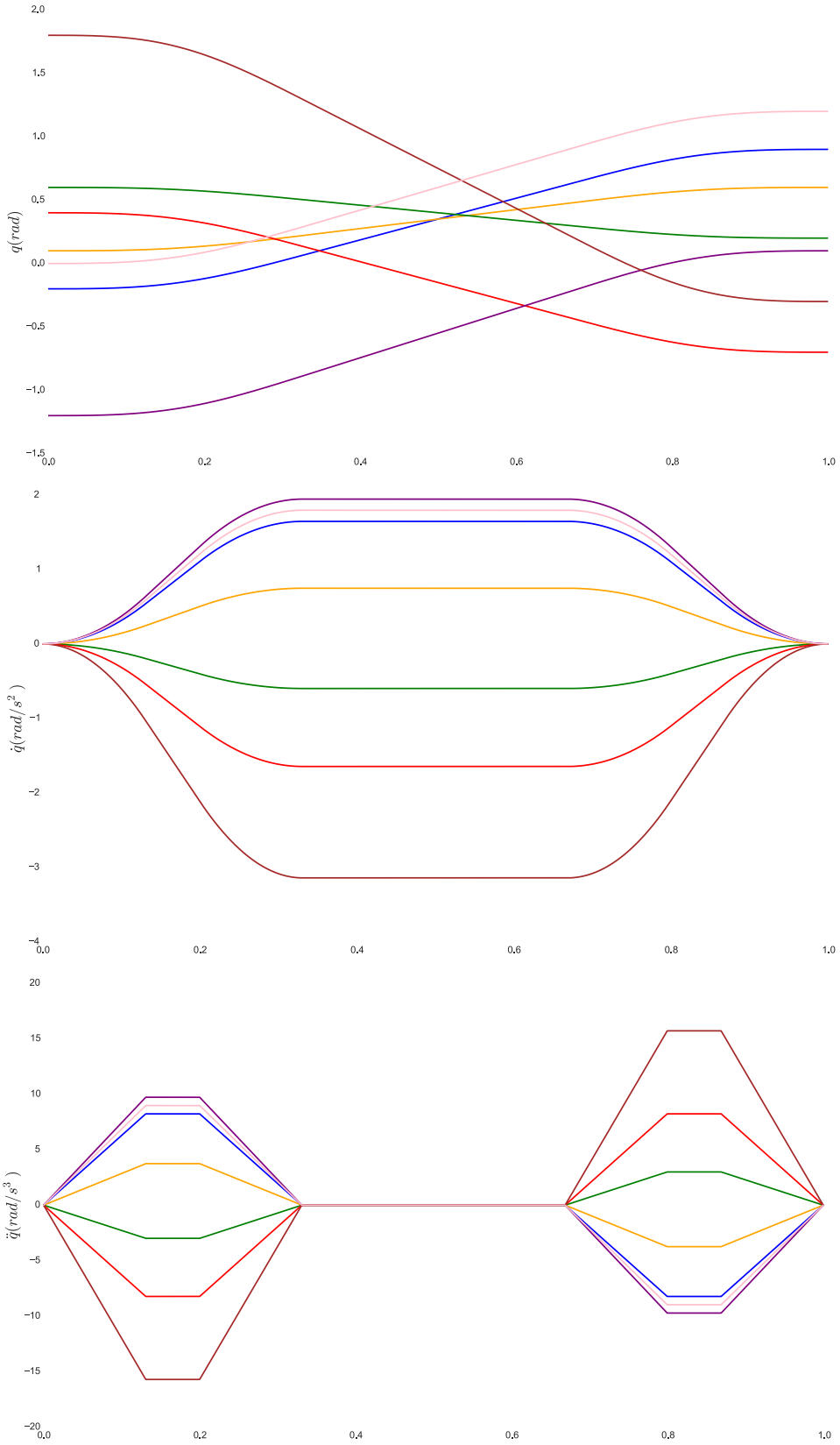


Figure 2.16: An example of phase synchronized trajectory for $\mathcal{K} = 7$.

Chapter 3

Reactive Trajectory Control

Contents

3.1 Introduction

In the previous chapter we presented an algorithm that generates time-optimum trajectories from arbitrary initial and final conditions, subject to asymmetric bounds on jerk, acceleration and velocity. We raised the problem of non-linearity related to short motions and asymmetric bounds that has been encountered in previous OTG

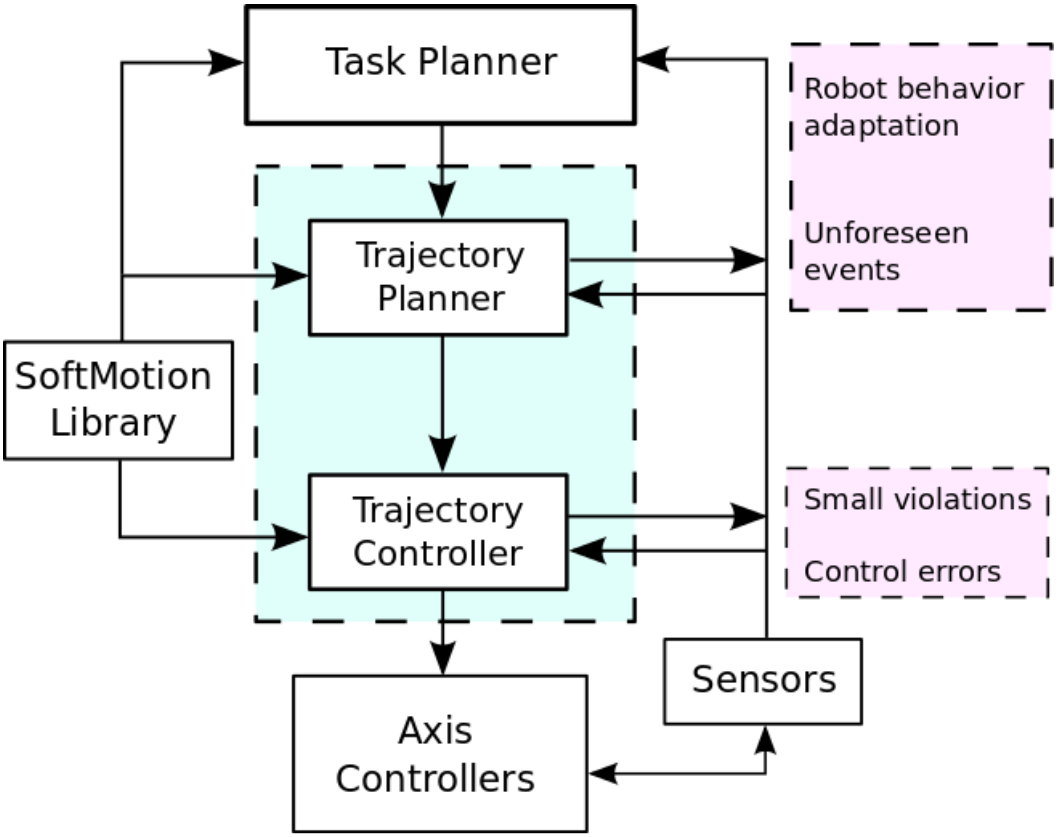


Figure 3.1: An architecture for Reactive Trajectory Control.

work but not treated. We explained how these non-linearities introduce discontinuities in the time-optimal curve and we proposed a solution. The algorithm is extended to be multi-dimensional. The proposed solution is based on sequences of segment of third degree polynomial functions. That classifies our model under the constrained jerk category. By limiting the jerk we can produce smooth trajectories and provide the generated motions with ergonomics properties. The constrained jerk model offer adaptability and flexibility since the parameterization of the kinematic bounds allows to define the properties of the motion. The system can adapt the robot behavior by choosing kinematic constraints that fit with the situation. Safety or ergonomics reasons can motivate these changes in constraints allowing the robot to adapt its behavior according to the presence of humans. With these qualities, the softMotion library can be used for the making of safe and desirable motions by a variety of robots, from industrials to collaborative.

The modifications of the kinematic bounds can occur while the robot is in motion. The bounds can be either extended or reduced, modifying the shape of $\mathcal{D}$. If $\mathcal{D}$ is extended, the current configuration of the system remains in $\mathcal{D}$ and the previous algorithm can cope with this situation by planning a new trajectory with

$\mathcal{B}$ updated. But when the elements of $\mathcal{B}$ are reduced, $\mathcal{D}$ is reduced as well, and the current configuration of the system can be outside of $\mathcal{D}$. In this situation the current state is inadmissible and the previous algorithm cannot plan a trajectory from this state. Non-constant motion constraints are not the only reason responsible for this situation. Vibrations, impacts or errors can be the cause. We then distinguish two kinds of situations causing the current state to be inadmissible: non constant motion constraints and control related issues.

Multiple strategies are available to restore the system in an admissible state from which the previous algorithm can compute a new trajectory. These strategies can be to stop the robot or to time-scale the trajectory. Another solution is to generate a new trajectory for the system to reach $\mathcal{D}$ and enable the previous algorithm to recover its function from there. This is the solution we will present and discuss in the first part of this chapter. We propose an algorithm that takes into account the cause of the deregulation, non constant motion constraints or control issues, to define a strategy that brings back the state of the system into $\mathcal{D}$.

Rest to rest motions are usually easy to compute and to experimentally apply. When the robot is moving, computing and executing a new trajectory raises numerous control problems. There is a gap between theory and reality that could explain why numerous works only produce simulation results. In the second part of this chapter we identify the difficulties of Reactive Trajectory Control and we propose solutions. The last part of this chapter is made of experimental results to prove the viability of our work.

3.1.1 Contributions

In this chapter we present an extension of the previous algorithm in order to cope with invalid initial configurations. We study the cause of the deregulation to propose an algorithm that builds a trajectory restoring the system under the kinematic limits. When the state of the system becomes admissible again, the previous algorithm can compute a trajectory from there. This work has been proposed and accepted for IROS 2019- IEEE/RSJ International Conference on Intelligent Robots and Systems. This extension also contributes to the softMotion library.

We also identify and address the difficulties of Reactive Trajectory Control, more specifically, the issues to address when the robot is not in a rest configuration and the system has to switch to a new trajectory.

All the work presented in this chapter is a contribution to the field of Reactive Trajectory Control. Experimental results are presented to validate our contributions.

3.1.2 Organization of the chapter

The chapter is organized as follow: The first section presents the extension to cope with non-admissible initial configurations. The first half of this section is related to the problem of non constant motion constraints (Sec.3.2.3). The second present a

different version of the algorithm for control related issues (Sec.3.2.4). The second section identify the difficulties of Reactive Trajectory Control (Sec.3.3). The third section are experimental results validating the work presented (Sec.3.4).

3.2 Trajectory Generation From Inadmissible State

3.2.1 Notations

We will use the following notations for the rest of the paper: $\mathcal{T}_r$ is the trajectory reaching $\mathcal{D}$ from C_i. $\mathcal{T}_r$ is composed of at most two trajectory segments with the characteristics described in Sec. 2.2.1.1. $C_{i'}$ denotes the end condition of $\mathcal{T}_r$, which is located on the limits of the extended phase diagram (see Fig. 3.2, Fig. 3.5). $\mathcal{T}_{opt}$ is the classical time-optimal trajectory introduced in the previous chapter. In this situation $\mathcal{T}_{opt}$ is computed from $C_{i'}$ to C_f. Finally $\mathcal{T}_{ext}$ names the overall trajectory, that is the concatenation of $\mathcal{T}_r$ and $\mathcal{T}_{opt}$.

3.2.2 Problem description

In the situations where C_i is outside of $\mathcal{D}$, our previous algorithm cannot compute the time-optimal trajectory $\mathcal{T}_{opt}$ joining C_i to C_f. Moreover being time-optimal is not sufficient and the previous heuristics must be adapted.

Similarly to the general problem of finding $\mathcal{T}_{opt}$ between two conditions, there is an infinite number of trajectories that can restore the integrity of the system. Furthermore it is important to note that for the computation of $\mathcal{T}_r$ the end-point is not specified, expanding the range of solutions. To simplify the construction of $\mathcal{T}_r$ the objectives must be clearly specified. We enumerate the following conditions the solution must verify:

C.1 Valid constraints cannot be intentionally transgressed.

C.2 Invalid constraints cannot be further violated.

C.3 Time-optimality of $\mathcal{T}_r$ in accord with C.1 and C.2.

In C.1 definition, the word "intentionally" is used to distinguish from situations where initial acceleration and velocity are under their respective limits but only temporarily. In Fig. 3.3 T_7 has both velocity and acceleration under their maxima, however velocity must be transgressed due to the acceleration value. For this reason $C_i \notin \mathcal{D}$.

We add a third condition that will serve as an heuristic for the construction of $\mathcal{T}_r$. C.3 purpose is to restore the system into the admissible domain $\mathcal{D}$ as fast as possible. The objective is to find the trajectory $\mathcal{T}_r$ that restore the fastest the robot's velocity and acceleration inside their limits. Jerk can be switched instantly. $\mathcal{T}_r$ end-point is then a valid initial condition from which the general algorithm can compute the time-optimal trajectory $\mathcal{T}_{opt}$ joining C_f. However to lower the total time the system stays outside $\mathcal{D}$, C.3 alone justify the transgression of initially

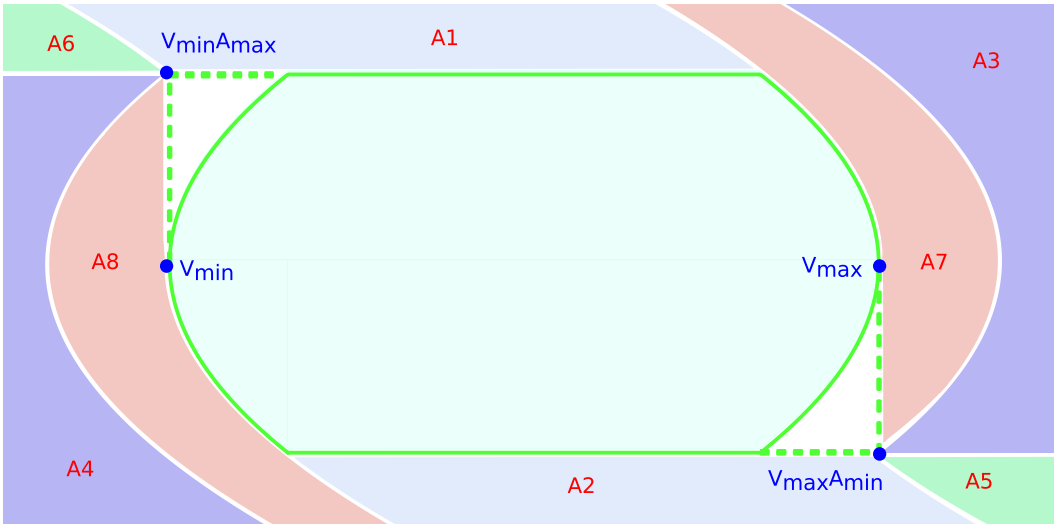


Figure 3.2: The extended phase diagram is built from the initial C_i configurations that define areas associated with a trajectory type. Some areas are similar due to symmetries, and have been sorted by colors.

valid constraints or to further transgress the invalids. This is prevented by adding the conditions C.1 and C.2. This choice is motivated by the study of the cause of the deregulation. Little transgressions come usually at control level due to tracking error or excessive vibrations. In this case using a higher jerk will only accentuate the problem. When bounds are largely exceeded it is because they have been redefined at higher level possibly in response to some events. The new bounds correspond to a policy defined by an entity that has the authority and that can be trusted since higher in the architecture hierarchy (Fig. 3.1).

This distinction is important for another reason (Fig. 3.1). Significant violations of the kinematic limits are not due to control but are decided at a higher level mainly motivated by safety and ergonomics reasons. Small violations are mainly due to control errors, and safety is not engaged. In the first context we are dealing with non-constant motion constraints. In the second it is control's transgressions.

Depending on the context, the condition C.3 can be relaxed. With non-constant motion constraints safety is the priority so the objective is to restore the integrity of the system as fast as possible, thus C.3 is maintained. In the second context where violations are small, C.3 can be relaxed in specific cases to prioritize the time-optimization of $\mathcal{T}_{ext}$ over $\mathcal{T}_r$.

3.2.3 Non-constant motion constraints

3.2.3.1 The extended phase diagram

A first step is to extend the phase diagram previously defined in Sec.2.2.1.2. The extension of $\mathcal{D}$ is represented by the dashed green line in (Fig. 3.2). The other

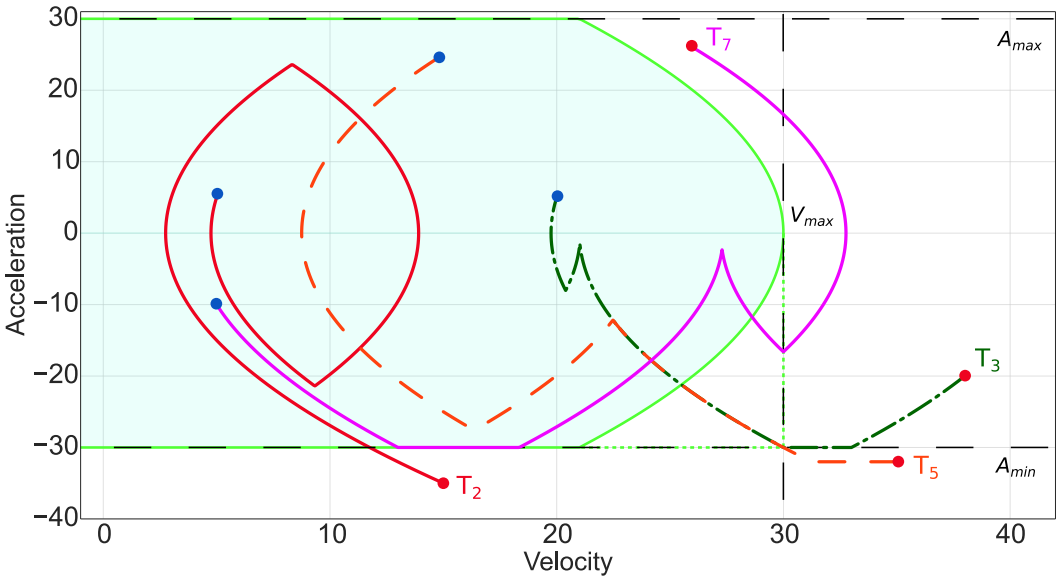


Figure 3.3: Example of trajectories illustrating the algorithm (Alg. 2) for different C_i. They are numbered accordingly to areas described in Fig. 3.2

two symmetrical corners are not available since the robot cannot stay in this area without breaking the speed limit.

The construction of $\mathcal{T}_r$ depends only on the velocity and acceleration value of the starting point. The distance x_f only matters in the control context that will be discussed in (Sec.3.2.4). The choice of $\mathcal{T}_r$ can then be made from the location of C_i in the phase diagram.

The phase diagram is a useful tool that provides a global view of all the possible trajectories available to construct $\mathcal{T}_r$ according to C_i location. We can now distinct a total of eight areas and the belonging of C_i to one of those will be responsible for the shape of $\mathcal{T}_r$ (Fig. 3.2). The belonging of C_i to one of those area is easy to verify and suitable for real-time applications. For a specific area there is a unique trajectory $\mathcal{T}_r$ that restore the system with respect to the imposed conditions (Sec.3.2.2).

3.2.3.2 Construction of the extended phase diagram

We define a set A of areas defining a partition of the phase diagram (Fig. 3.2). Let's denote $\mathcal{P}^{-}_{\mathcal{V}_{min}}$ and $\mathcal{P}^{+}_{\mathcal{V}_{min}}$ the parabolas defined by $\mathcal{V}_{min}$ and respectively $\mathcal{J}_{min}$ and $\mathcal{J}_{max}$. Similarly, $\mathcal{P}^{-/+}_{\mathcal{V}_{max}}$, $\mathcal{P}^{-/+}_{\mathcal{V}_{max}\mathcal{A}_{min}}$ and $\mathcal{P}^{-/+}_{\mathcal{V}_{min}\mathcal{A}_{max}}$ are the parabolas that pass through the points $\mathcal{V}_{max}$, $\mathcal{V}_{max}\mathcal{A}_{min}$ and $\mathcal{V}_{min}\mathcal{A}_{max}$. In the same way, the maximum acceleration linear segments are $\mathcal{A}_{min}$, $\mathcal{A}_{max}$ and the vertical maximum velocity segments are $\mathcal{V}_{min}$ and $\mathcal{V}_{max}$.

The area $A6$ is, for example, defined by $\mathcal{P}^{-}_{\mathcal{V}_{min}\mathcal{A}_{max}}$ and $\mathcal{A}_{max}$. Its symmetric counterpart $A5$ is defined by $\mathcal{P}^{+}_{\mathcal{V}_{max}\mathcal{A}_{min}}$ and $\mathcal{A}_{min}$.

These parabolas are centred on zero acceleration axis and are defined by a jerk

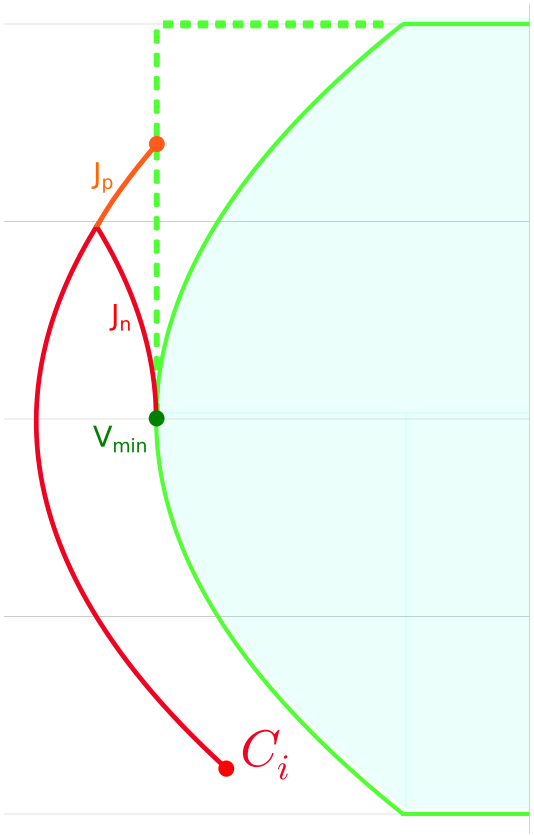


Figure 3.4: The orange trajectory represents $\mathcal{T}_r$ computed with (Alg.2). It is the fastest trajectory the reach $\mathcal{D}$. The red trajectory is the fastest to reach $\mathcal{V}_{min}$. It is not the fastest to reach $\mathcal{D}$, but can be the fastest by a good margin to reach C_f if $\mathcal{T}_{opt}$ is composed of a $\mathcal{V}_{min}$ segment.

J_0, and by a velocity $\mathcal{V}_0$ along the null acceleration, or by a point defined by its velocity and acceleration (V_i, A_i). The jerks J_{min} and J_{max} define four condition parabolas passing through the initial conditions C_i and the final condition C_f. We denote them as ${C_i}^-$, ${C_i}^+$, ${C_f}^-$, ${C_f}^+$.

By using these properties it is easy to test the belonging of C_i to one of those areas.

3.2.3.3 Construction of $\mathcal{T}_r$

The algorithm for the construction of $\mathcal{T}_r$ is presented in (Alg. 2). It relies on the phase diagram presented in Fig. 3.2.

3.2.4 Transgressed motion state

Small violations are more susceptible to be linked to control rather to a decrease of the bounds. It can be due to vibrations caused by a high jerk, to the incapacity

Algorithm 1 getArea

if $((C_i.a > \mathcal{A}_{max})$**and**$(C_i{}^-.\mathcal{V}_0 > \mathcal{P}^-_{\mathcal{V}_{min}\mathcal{A}_{max}}.\mathcal{V}_0)$**and**$(C_i{}^-.\mathcal{V}_0 < \mathcal{P}^-_{\mathcal{V}_{max}\mathcal{A}_{max}}.\mathcal{V}_0))$ **then**
  return $A1$
end if
if $((C_i.a < \mathcal{A}_{min})$**and**$(C_i{}^+.\mathcal{V}_0 > \mathcal{P}^+_{\mathcal{V}_{min}\mathcal{A}_{min}}.\mathcal{V}_0)$**and**$(C_i{}^+.\mathcal{V}_0 < \mathcal{P}^+_{\mathcal{V}_{max}\mathcal{A}_{min}}.\mathcal{V}_0))$ **then**
  return $A2$
end if
if $((C_i.a > \mathcal{A}_{min})$**and**$(C_i{}^-.\mathcal{V}_0 > \mathcal{P}^-_{\mathcal{V}_{max}\mathcal{A}_{min}}.\mathcal{V}_0)$ **then**
  return $A3$
end if
if $((C_i.a < \mathcal{A}_{max})$**and**$(C_i{}^+.\mathcal{V}_0 < \mathcal{P}^+_{\mathcal{V}_{min}\mathcal{A}_{max}}.\mathcal{V}_0)$ **then**
  return $A4$
end if
if $((C_i.a < \mathcal{A}_{min})$**and**$(C_i{}^+.\mathcal{V}_0 > \mathcal{P}^+_{\mathcal{V}_{max}\mathcal{A}_{min}}.\mathcal{V}_0)$ **then**
  return $A5$
end if
if $((C_i.a > \mathcal{A}_{max})$**and**$(C_i{}^-.\mathcal{V}_0 < \mathcal{P}^-_{\mathcal{V}_{min}\mathcal{A}_{max}}.\mathcal{V}_0)$ **then**
  return $A6$
end if
if $(C_i.a > 0.0)$ **then**
  if $((C_i{}^-.\mathcal{V}_0 > \mathcal{A}_{max})$**and**$(C_i{}^-.\mathcal{V}_0 =< \mathcal{P}^-_{\mathcal{V}_{max}\mathcal{A}_{min}}.\mathcal{V}_0)$ **then**
    return $A7$
  end if
else
  if $((C_i.v > \mathcal{V}_{max})$**and**$(C_i{}^-.\mathcal{V}_0 =< \mathcal{P}^-_{\mathcal{V}_{max}\mathcal{A}_{min}}.\mathcal{V}_0)$ **then**
    return $A7$
  end if
end if
if $(C_i.a > 0.0)$ **then**
  if $((C_i.v < \mathcal{V}_{min})$**and**$(C_i{}^+.\mathcal{V}_0 >= \mathcal{P}^+_{\mathcal{V}_{min}\mathcal{A}_{max}}.\mathcal{V}_0)$ **then**
    return $A8$
  end if
else
  if $((C_i{}^+.\mathcal{V}_0 < \mathcal{V}_{min})$**and**$(C_i{}^+.\mathcal{V}_0 >= \mathcal{P}^+_{\mathcal{V}_{min}\mathcal{A}_{max}}.\mathcal{V}_0)$ **then**
    return $A8$
  end if
end if

Algorithm 2 *compute_*$\mathcal{T}_r$

$\mathcal{T}_r \leftarrow [\,]$
$area \leftarrow getArea()$
switch $(area)$
case $A1$**:**
  $\mathcal{T}_r \leftarrow \mathcal{J}_{min}$ jerk segment reaching $\mathcal{A}_{max}$
case $A2$**:**
  $\mathcal{T}_r \leftarrow \mathcal{J}_{max}$ jerk segment reaching $\mathcal{A}_{min}$
case $A3$**:**
  $\mathcal{T}_r \leftarrow \mathcal{J}_{min}$ jerk segment reaching $\mathcal{A}_{min}$
  $\mathcal{T}_r \leftarrow \mathcal{A}_{min}$ constant linear acceleration segment reaching $\mathcal{V}_{max}\mathcal{A}_{min}$
case $A4$**:**
  $\mathcal{T}_r \leftarrow \mathcal{J}_{max}$ jerk segment reaching $\mathcal{A}_{max}$
  $\mathcal{T}_r \leftarrow \mathcal{A}_{max}$ constant linear acceleration segment reaching $\mathcal{V}_{min}\mathcal{A}_{max}$
case $A5$**:**
  $\mathcal{T}_r \leftarrow \mathcal{A}_{min}$ constant linear acceleration segment reaching $\mathcal{P}^{+}_{\mathcal{V}_{max}\mathcal{A}_{min}}$
  $\mathcal{T}_r \leftarrow \mathcal{J}_{max}$ jerk segment reaching $\mathcal{V}_{max}\mathcal{A}_{min}$
case $A6$**:**
  $\mathcal{T}_r \leftarrow \mathcal{A}_{max}$ constant linear acceleration segment reaching $\mathcal{P}^{-}_{\mathcal{V}_{min}\mathcal{A}_{max}}$
  $\mathcal{T}_r \leftarrow \mathcal{J}_{min}$ jerk segment reaching $\mathcal{V}_{min}\mathcal{A}_{max}$
case $A7$**:**
  $\mathcal{T}_r \leftarrow \mathcal{J}_{min}$ jerk segment reaching $\mathcal{V}_{max}$
case $A8$**:**
  $\mathcal{T}_r \leftarrow \mathcal{J}_{max}$ jerk segment reaching $\mathcal{V}_{min}$
end switch
return $\mathcal{T}_r$

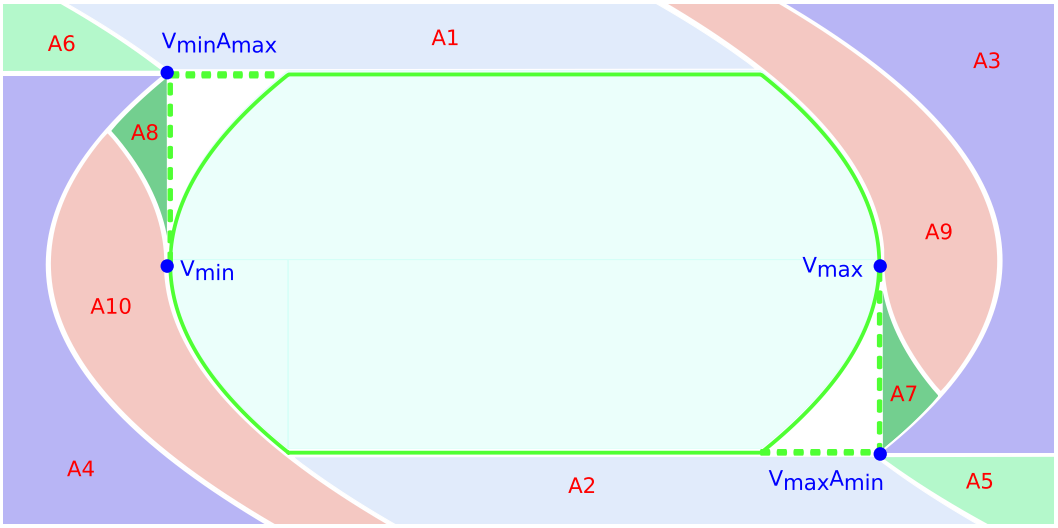


Figure 3.5: A phase diagram (based on Fig. 3.2) optimized for the reasons mentioned in Sec.3.2.4.1.

of the system to follow the commands with a sufficient accuracy, to external forces etc. It is important to make the distinction with non-constant motion constraints as safety is not the concern in this situation. Therefore it is less interesting to prioritize $\mathcal{T}_r$ time-optimization that is linked to safety over the standard time-optimization of the whole trajectory that is $\mathcal{T}_{ext}$. To do that the condition C.3 have to be loosen. In most case the time-minimization of $\mathcal{T}_{ext}$ does not go against that of $\mathcal{T}_r$, or it is not significant enough to be worth it. However we will study two particular instances where this optimization can be interesting.

3.2.4.1 Construction of the extended phase diagram

The first thing to do is to study the situations that present a significant benefit from the minimization of the duration of $\mathcal{T}_{ext}$ over $\mathcal{T}_r$. Since in this instance violations are small, C_i should not belong to areas $A3$ and $A4$. For areas $A5$ and $A6$ there is only one entry point to $\mathcal{D}$, and the solution presented in (Alg. 2) is already the fastest. In fact, by studying the phase diagram (see Fig. 3.2) there is two areas where optimizing the duration of $\mathcal{T}_r$ can really be at the expense of the overall duration. These areas are $A7$ and $A8$ that are symmetrical and present the same scenario. In this scenario the trajectory $\mathcal{T}_{opt}$ has a $\mathcal{V}_{min}$ or respectively $\mathcal{V}_{max}$ segment.

An example is described in Fig. 3.4 in which $C_i \in A8$. The orange positive jerk segment reaching $\mathcal{V}_{min}$ defines the fastest trajectory to enter $\mathcal{D}$ and correspond to the solution of (Alg. 2). The trajectory plotted in red is the fastest to reach the constant velocity segment $\mathcal{V}_{min}$. The time gained on $\mathcal{T}_r$ with the first solution can be negligible, since this scenario happens in areas $A7$ and $A8$ that are close to $\mathcal{D}$. However $\mathcal{T}_{opt}$ must produce a relatively long motion after that to rejoin $\mathcal{V}_{min}$.

With the red trajectory, $\mathcal{T}_{opt}$ is already on $\mathcal{V}_{min}$. Hence the time gained on $\mathcal{T}_{ext}$ by choosing the red trajectory can be significant for large negative motions that reach $\mathcal{V}_{min}$. Moreover we can argue that the resulting trajectory will appear more smoother and more natural by eliminating parasitic movements.

To propose a solution we first need to bring modifications to the phase diagram presented in Fig. 3.2. We introduce two more areas by splitting $A7$ and $A8$ (see Fig. 3.5). The construction of this phase diagram optimized for the context of control is presented in (Alg. 3).

3.2.4.2 Construction of TR

The algorithm for the construction of $\mathcal{T}_r$ is presented in (Alg. 4). It relies on the phase diagram optimized for the context of control (see Fig. 3.5). The construction of $\mathcal{T}_r$ for areas $A1$ to $A6$ is identical to (Alg. 2).

*compute_*$\mathcal{T}_{opt}$() denotes the algorithm presented in the previous chapter.

3.2.4.3 Optimization example

Fig. 3.6 and Fig. 3.7 illustrates the optimization proposed in Alg.4. For both pictures it is a large negative motion so that the trajectories must stay on V_{min}. It is at this condition that the optimization can be done. In Fig. 3.6 the bounds are symmetric. As we can see in Table. 3.1 the results are as expected. The duration of $\mathcal{T}_r$ is shorter with Alg. 2, hence the system is brought back into $\mathcal{D}$ faster. However the duration of $\mathcal{T}_{ext}$ is shorter with Alg. 4. In this example the gain on $\mathcal{T}_{ext}$ is roughly $0.1147s$. But this gain can be drastically increased in some configurations, as it depends on multiple parameters. One of these parameters is J_{min}. In Fig. 3.7 the same configuration as in Fig. 3.6 is used, with the exception of J_{min} that is decreased. Now the gain on $\mathcal{T}_{ext}$ is $10.865s$, hence $\mathcal{T}_{ext}$ takes 42.28% less time with Alg. 4.

Obviously the example chosen is quite extreme since it is a very asymmetrical jerk, but it shows that there is an interest at investigating further these behaviours. The proposed optimization is not exhaustive and can be vastly improved.

	Fig. 3.6a	Fig. 3.6b	Fig. 3.7a	Fig. 3.7b
$\mathcal{T}_r$	0.92	1.11054	0.92	1.98565
$\mathcal{T}_{opt}$	13.2897	12.9845	24.7803	12.8497
$\mathcal{T}_{ext}$	14.2097	14.095	25.7003	14.8353

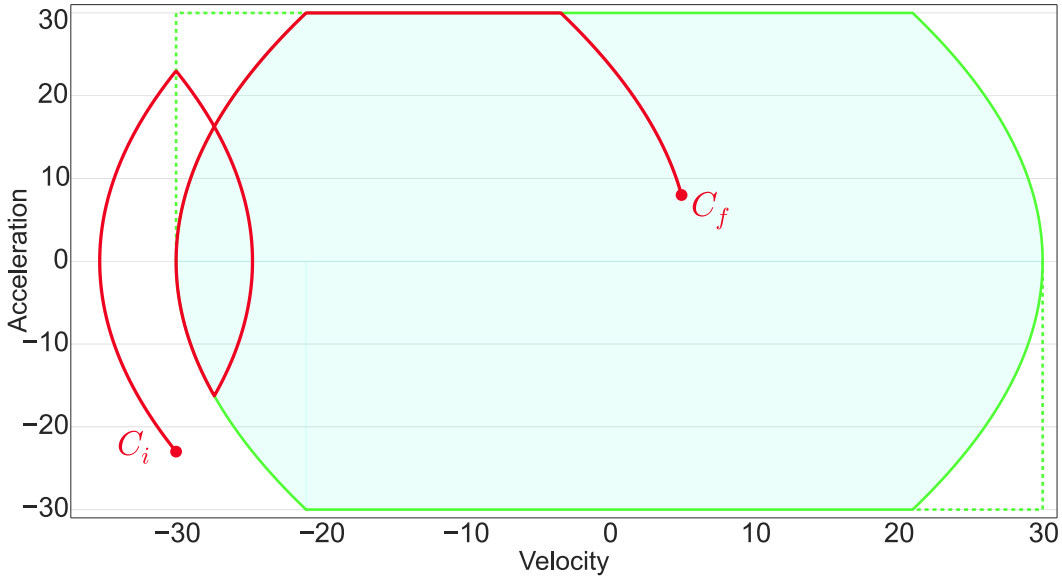
Table 3.1: This table synthesizes the duration expressed in seconds of each trajectories for each figure.

Algorithm 3 getArea

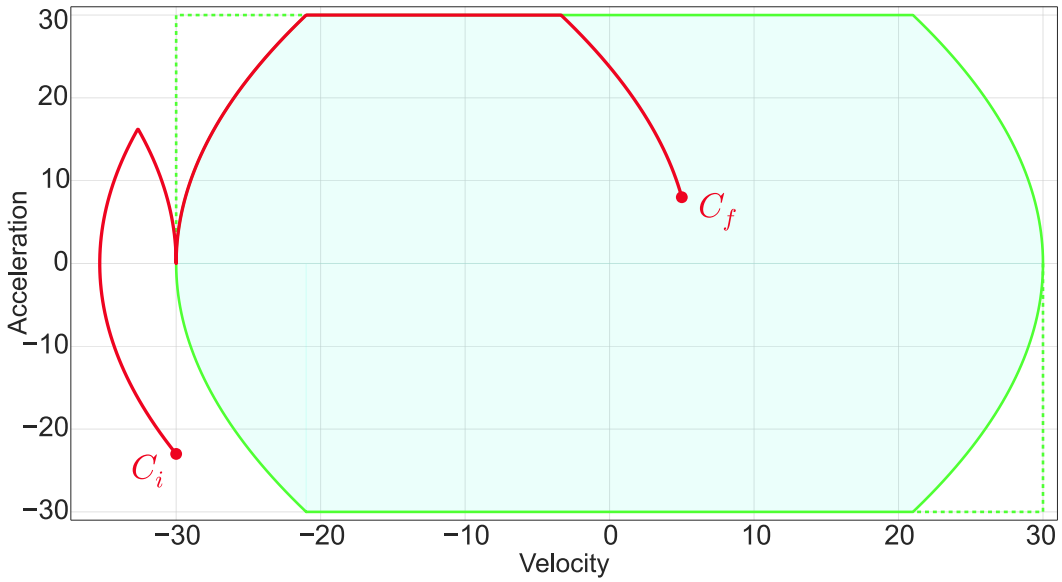
if $((C_i.a > \mathcal{A}_{max})$**and**$(C_i{}^-.\mathcal{V}_0 > \mathcal{P}^-_{\mathcal{V}_{min}\mathcal{A}_{max}}.\mathcal{V}_0)$**and**$(C_i{}^-.\mathcal{V}_0 < \mathcal{P}^-_{\mathcal{V}_{max}\mathcal{A}_{max}}.\mathcal{V}_0))$ **then**
  return $A1$
end if
if $((C_i.a < \mathcal{A}_{min})$**and**$(C_i{}^+.\mathcal{V}_0 > \mathcal{P}^+_{\mathcal{V}_{min}\mathcal{A}_{min}}.\mathcal{V}_0)$**and**$(C_i{}^+.\mathcal{V}_0 < \mathcal{P}^+_{\mathcal{V}_{max}\mathcal{A}_{min}}.\mathcal{V}_0))$ **then**
  return $A2$
end if
if $((C_i.a > \mathcal{A}_{min})$**and**$(C_i{}^-.\mathcal{V}_0 > \mathcal{P}^-_{\mathcal{V}_{max}\mathcal{A}_{min}}.\mathcal{V}_0)$ **then**
  return $A3$
end if
if $((C_i.a < \mathcal{A}_{max})$**and**$(C_i{}^+.\mathcal{V}_0 < \mathcal{P}^+_{\mathcal{V}_{min}\mathcal{A}_{max}}.\mathcal{V}_0)$ **then**
  return $A4$
end if
if $((C_i.a < \mathcal{A}_{min})$**and**$(C_i{}^+.\mathcal{V}_0 > \mathcal{P}^+_{\mathcal{V}_{max}\mathcal{A}_{min}}.\mathcal{V}_0)$ **then**
  return $A5$
end if
if $((C_i.a > \mathcal{A}_{max})$**and**$(C_i{}^-.\mathcal{V}_0 < \mathcal{P}^-_{\mathcal{V}_{min}\mathcal{A}_{max}}.\mathcal{V}_0)$ **then**
  return $A6$
end if
if $((C_i.v > \mathcal{V}_{max})$**and**$(C_i{}^+.\mathcal{V}_0 < \mathcal{V}_{max})$**and**$(C_i{}^-.\mathcal{V}_0 < \mathcal{P}^-_{\mathcal{V}_{max}\mathcal{A}_{min}}.\mathcal{V}_0))$ **then**
  return $A7$
end if
if $((C_i.v < \mathcal{V}_{min})$**and**$(C_i{}^-.\mathcal{V}_0 > \mathcal{V}_{min})$**and**$(C_i{}^+.\mathcal{V}_0 > \mathcal{P}^+_{\mathcal{V}_{min}\mathcal{A}_{max}}.\mathcal{V}_0))$ **then**
  return $A8$
end if
if $C_i{}^-.\mathcal{V}_0 =< \mathcal{P}^-_{\mathcal{V}_{max}\mathcal{A}_{min}}.\mathcal{V}_0)$ **then**
  if $((C_i.a <= 0.0)$**and**$(C_i{}^+.\mathcal{V}_0 >= \mathcal{V}_{max})$ **then**
    return $A9$
  else if $((C_i.a > 0.0)$**and**$(C_i{}^-.\mathcal{V}_0 >= \mathcal{P}^-_{\mathcal{V}_{max}\mathcal{A}_{max}}.\mathcal{V}_0))$ **then**
    return $A9$
  end if
end if
if $C_i{}^+.\mathcal{V}_0 >= \mathcal{P}^+_{\mathcal{V}_{min}\mathcal{A}_{max}}.\mathcal{V}_0)$ **then**
  if $((C_i.a <= 0.0)$**and**$(C_i{}^+.\mathcal{V}_0 < \mathcal{P}^+_{\mathcal{V}_{min}\mathcal{A}_{min}}.\mathcal{V}_0)$ **then**
    return $A10$
  else if $((C_i.a > 0.0)$**and**$(C_i{}^-.\mathcal{V}_0 <= \mathcal{V}_{min}))$ **then**
    return $A10$
  end if
end if

Algorithm 4 *compute_*$\mathcal{T}_r$

$\mathcal{T}_r \leftarrow [\,]$
$area \leftarrow getArea()$
switch $(area)$
case $A1$**:**
  $\mathcal{T}_r \leftarrow \mathcal{J}_{min}$ jerk segment reaching $\mathcal{A}_{max}$
case $A2$**:**
  $\mathcal{T}_r \leftarrow \mathcal{J}_{max}$ jerk segment reaching $\mathcal{A}_{min}$
case $A3$**:**
  $\mathcal{T}_r \leftarrow \mathcal{J}_{min}$ jerk segment reaching $\mathcal{A}_{min}$
  $\mathcal{T}_r \leftarrow \mathcal{A}_{min}$ constant linear acceleration segment reaching $\mathcal{V}_{max}\mathcal{A}_{min}$
case $A4$**:**
  $\mathcal{T}_r \leftarrow \mathcal{J}_{max}$ jerk segment reaching $\mathcal{A}_{max}$
  $\mathcal{T}_r \leftarrow \mathcal{A}_{max}$ constant linear acceleration segment reaching $\mathcal{V}_{min}\mathcal{A}_{max}$
case $A5$**:**
  $\mathcal{T}_r \leftarrow \mathcal{A}_{min}$ constant linear acceleration segment reaching $\mathcal{P}^{+}_{\mathcal{V}_{max}\mathcal{A}_{min}}$
  $\mathcal{T}_r \leftarrow \mathcal{J}_{max}$ jerk segment reaching $\mathcal{V}_{max}\mathcal{A}_{min}$
case $A6$**:**
  $\mathcal{T}_r \leftarrow \mathcal{A}_{max}$ constant linear acceleration segment reaching $\mathcal{P}^{-}_{\mathcal{V}_{min}\mathcal{A}_{max}}$
  $\mathcal{T}_r \leftarrow \mathcal{J}_{min}$ jerk segment reaching $\mathcal{V}_{min}\mathcal{A}_{max}$
case $A7$**:**
  $\mathcal{T}_r \leftarrow \mathcal{J}_{min}$ jerk segment reaching $\mathcal{V}_{max}$
case $A8$**:**
  $\mathcal{T}_r \leftarrow \mathcal{J}_{max}$ jerk segment reaching $\mathcal{V}_{min}$
case $A9$**:**
  $\mathcal{T}_r \leftarrow \mathcal{J}_{min}$ jerk segment reaching $\mathcal{V}_{max}$
  $\mathcal{T}_{opt} \leftarrow compute\_\mathcal{T}_{opt}()$
  if ($\mathcal{T}_{opt}$ has V_{max} segment) **then**
    $\mathcal{T}_r \leftarrow [\,]$
    $\mathcal{T}_r \leftarrow \mathcal{J}_{min}$ jerk segment reaching $({C_i}^{-}, \mathcal{V}^{+}_{max})$ intersection
    $\mathcal{T}_r \leftarrow \mathcal{J}_{max}$ jerk segment reaching $\mathcal{V}_{max}$
  end if
case $A10$**:**
  $\mathcal{T}_r \leftarrow \mathcal{J}_{max}$ jerk segment reaching $\mathcal{V}_{min}$
  $\mathcal{T}_{opt} \leftarrow compute\_\mathcal{T}_{opt}()$
  if ($\mathcal{T}_{opt}$ has V_{min} segment) **then**
    $\mathcal{T}_r \leftarrow [\,]$
    $\mathcal{T}_r \leftarrow \mathcal{J}_{max}$ jerk segment reaching $({C_i}^{+}, \mathcal{V}^{-}_{min})$ intersection
    $\mathcal{T}_r \leftarrow \mathcal{J}_{min}$ jerk segment reaching $\mathcal{V}_{min}$
  end if
end switch
return $\mathcal{T}_r$



(a) $\mathcal{T}_{ext}$ using Alg. 2.



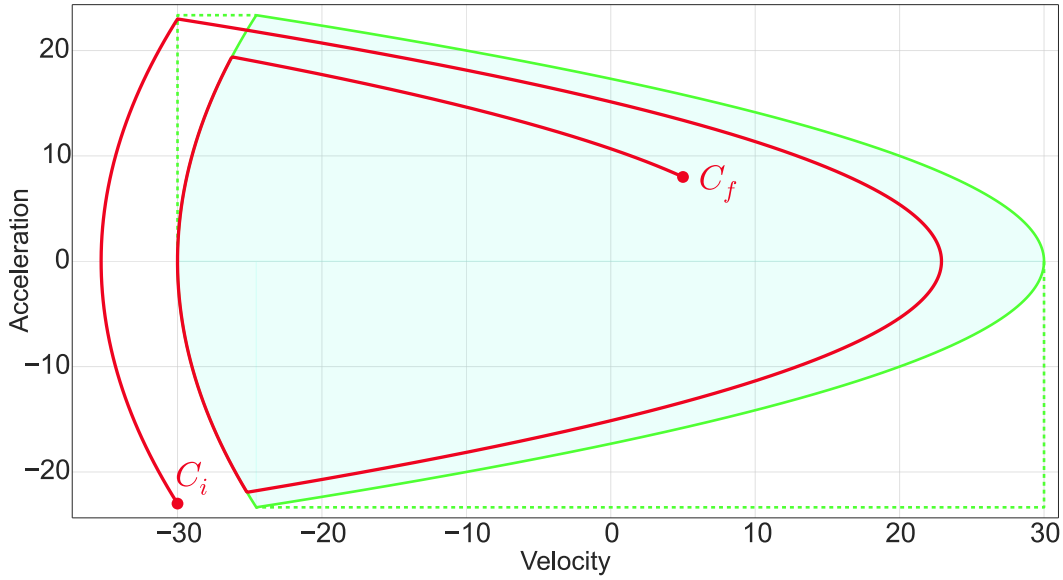
(b) $\mathcal{T}_{ext}$ using Alg. 4.

Figure 3.6: Configuration used : (J_{min}=−50, J_{max}=50, A_{min}=−30, A_{max}=30, V_{min}=−30, V_{max}=30, a_i= − 23, v_i=−30, a_f=8, v_f=5, x_f= − 400).

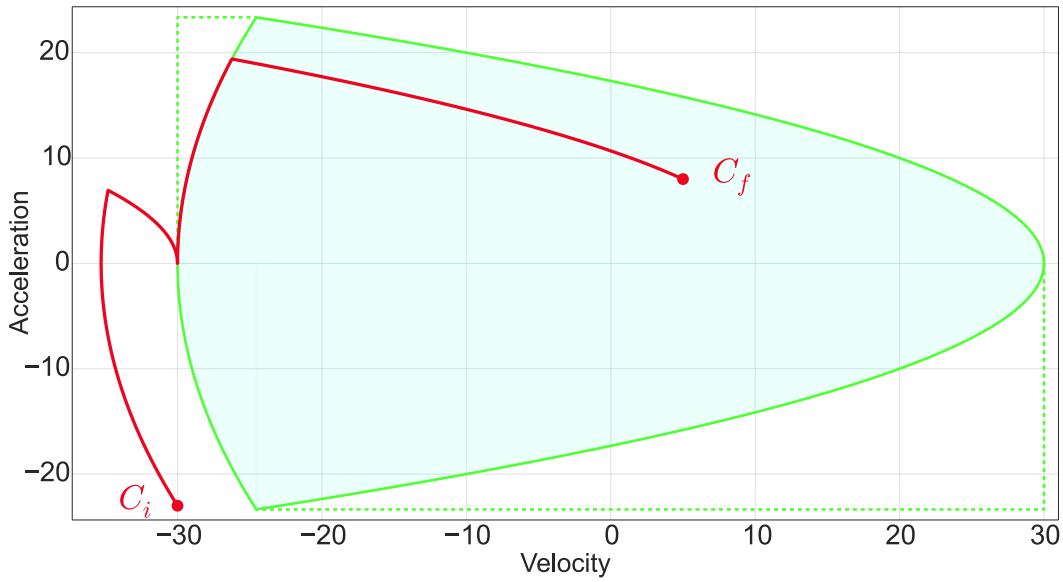
3.2.5 Discussion

3.2.5.1 Comparison with related works

To our knowledge there is no work proposing a thorough analysis on time variant kinematic constraints. In [Kroger 2010, Zhao 2017] the adopted solution is to first lower acceleration then velocity. It is motivated in [Zhao 2017] to ensure the time-



(a) $\mathcal{T}_{ext}$ using Alg. 2.



(b) $\mathcal{T}_{ext}$ using Alg. 4.

Figure 3.7: Same configuration as Fig. 3.6 except for J_{min} that is decreased to -5.

optimality of $\mathcal{T}_r$. But it is true only when the initial acceleration and velocity have the same sign.

We illustrate our point by comparing the different solutions (see Fig. 3.8). C_i must belong to either $A5$ or $A6$. Using equations (Eq. 2.1) we can compute the duration of the two trajectories. The duration of $\mathcal{T}_r$ with our method is $0.277s$ while it takes $0.295s$ using the algorithms described in [Kroger 2010, Zhao 2017].

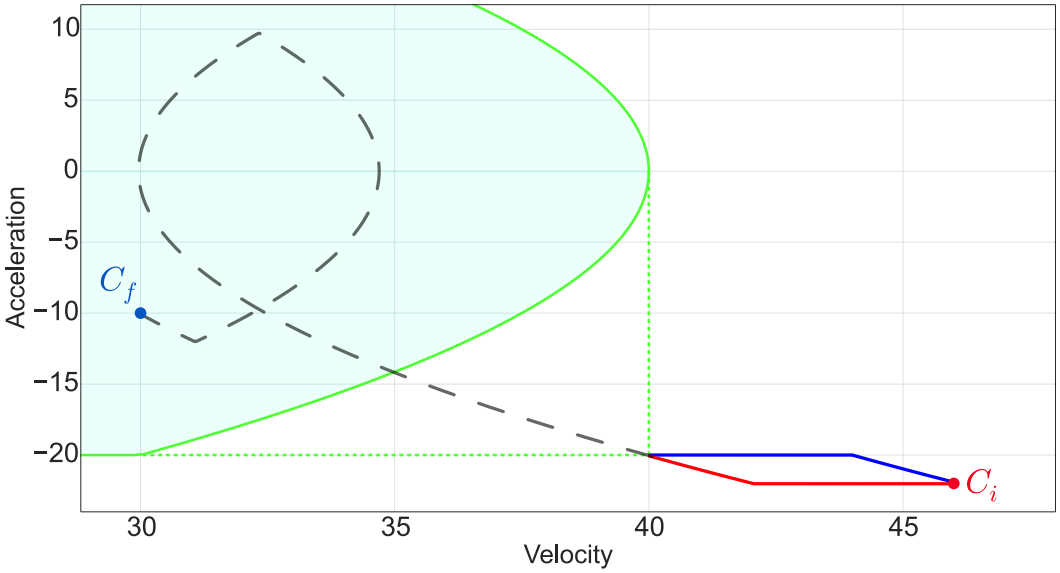


Figure 3.8: Our solution to reach $\mathcal{D}$ here in red, and the one presented in [Kroger 2010, Zhao 2017] in blue (Sec. 3.2.5.1). We choose the following configuration: $C_i = (0, -20, -22)$ and $V_{max} = 40, A_{min} = -20$ so C_i belongs to $A5$.

Table 3.2: Mean computation times for the different trajectories.

Case	$\mathcal{T}_{opt}$	$\mathcal{T}_r$	$\mathcal{T}_{ext}$
Times (μs)	1.04	0.12	1.16

3.2.5.2 Discussion : Time-scale techniques comparison

Time scaling is a simple concept often use in trajectory generation and control. By replacing time with an increasing monotonous function of time, it allows to preserve the multi-axis synchronization and therefore the path. It can be used for multi-axis synchronization, torque correction, obstacle avoidance and more ([Szadeczky-Kardoss 2006, Kiss 2007, Zhao 2015]). This technique is often used to cope with transgressed kinematic bounds, as it is simple to apply and keep the geometry of the reference trajectory. However, the time-scaling factor cannot be set abruptly and it is best to generate a smooth function to represent its evolution. Despite this caution we cannot ensure that the scaled portion of the trajectory will be smooth enough as we have no control and visibility on the kinematic bounds. In practice this method is not reliable enough, and lacks flexibility, as we do not tune the kinematic parameters.

3.2.6 Performances

The implementation of this extended algorithm for one DOF (computation of $\mathcal{T}_{ext}$) on a system equipped with an Intel Core i7 processor running at 2.2 GHz gives a mean time of 1.16 μs with a standard deviation of 0.39 μs observed for 10^8 random tests, it certainly allows online usage. In [Kroger 2012] no computation times are specified. Since the paper is an extension of [Kroger 2010] that claims an average execution time of 135 μs for a 6DOF system it can be assumed that it is higher. In [Zhao 2017] the average execution time for a 6DOF system is 1.1 ms. It appears that our solution provides better execution times than the previous works. The table 3.2 gives a view of the computation times for the different trajectories.

3.3 Reactive Trajectory Control

We presented an extension of the algorithm developed in section 2.2.1 in order to cope with invalid initial configurations. This addition allows the system to react in real-time to unforeseen events by generating a new trajectory that will bring back the system into its admissible kinematic domain. This feature is essential to ensure a satisfactory level of safety when humans operate in the robot's vicinity (Sec.1.2.3). It is also used in control for small violations of the limits.

We will now study how to switch the robot trajectory in real time for a real applicative context. Switching from a trajectory to a new one suppose to know precisely the kinematic state of the robot at the switching time. Unfortunately, robots do not always provide the required velocity and acceleration measurements, but only the position. Moreover, motions are usually not computed at the lowest control layer, and once the trajectory is ready to be executed by the controller it might be outdated. In the next paragraph we will study how to estimate the current kinematic state and build an efficient controller.

3.3.1 Notations

For the rest of the chapter, we will use the following notations :

FRI : Fast Research Interface. It is a simple user interface provided by KUKA for its Light-Weight Robot IV. In our case we are using the interface proposed by the university of Stanford (See `https://cs.stanford.edu/people/tkr/fri/html/`).

ar_track_alvar : A ros wrapper for Alvar, an open source AR tag tracking library (See `http://wiki.ros.org/ar_track_alvar`).

GenoM : Generator of Modules (See `https://www.openrobots.org/wiki/genom`).

t_c is the period of the trajectory controller, usually $5ms$. t_p is the period of the trajectory planner, usually $50ms$ (See Fig. 3.9).

q, $\dot{q}$, $\ddot{q}$ refer respectively to the measured or estimated angular position, velocity and acceleration.

x, $\dot{x}$, $\ddot{x}$ to the measured (or estimated) Cartesian position, velocity and acceleration.

$\overline{q}$, $\overline{x}$ are the angular and Cartesian commands in position.

$\overline{\mathcal{K}}$, $\overline{\tau}$ are respectively the commanded stiffness and torques.

For the following we need to distinguish different types of trajectories. $\mathcal{T}_o$ denotes the current motion being executed. $\mathcal{T}_o$ joins C_i to C_f and is computed by the trajectory planner before being sent to the trajectory controller. It can be a rest to rest motion.

$\mathcal{T}_n$ refers to a new trajectory that is computed at a time t_{Switch} by the trajectory planner while the trajectory controller is busy executing $\mathcal{T}_o$. The final condition C_f^n can be C_f, however it is **not a rest to rest motion**. It starts at C_i^n that is the estimation of the robot current state at t_{Switch} that the trajectory planner possesses. This estimate has been made by the trajectory controller and was communicated upwards in the architecture (see Fig. 3.1).

$\mathcal{T}_j$ refers to the junction trajectory. Unlike $\mathcal{T}_o$ or $\mathcal{T}_n$, $\mathcal{T}_j$ is generated by the trajectory controller. Its function is to smooth the junction between $\mathcal{T}_o$ and $\mathcal{T}_n$. $\mathcal{T}_j$ is computed at t_{Exec} that is the time at which the trajectory controller receives $\mathcal{T}_n$.

C_i^j and C_f^j denotes respectively $\mathcal{T}_j$ initial and end conditions.

C_r denotes the real state of the robot, at any instant t. It is important to note that C_r is unknown and need to be estimated. In Fig. 3.11 C_r and its estimate C_i^j are illustrated for t_{Exec}.

ε_{err} is the distance between C_i^n and C_r at t_{Exec}.

$$\varepsilon_{err} = \varepsilon_{imp} + \varepsilon_{off}$$

$Th_{[x,y]}$ is the homogeneous transformation matrix between the frames $\mathcal{F}_x$ and $\mathcal{F}_y$.

More explanations are given in the section 3.3.4 related to the junction trajectories.

3.3.2 Reactive Control Architecture

First we present in more details the architecture introduced in Fig. 3.1. The architecture presents a higher granularity as it simplifies the input for each node that can run at different frequencies according to the nature of the task. Trajectories are used as the main support of information between a slow task planner and the fast robot controllers, thus simplifying the overall communication. For the sake of simplicity, we will only focus on the components presented in Fig. 3.9. The classic task planner does not participate here.

The trajectory planner that is placed higher in the hierarchy of modules can take the role of an interface for the task planner by proposing elementary actions. These actions can be defined as point to point motions in angular or joint space, via-points trajectories or more sophisticated tasks. It can also be asked to change the control parameters, such as the kinematic limits, the tolerable errors and more. It can be connected to sensors to react much faster to events than a task planner.

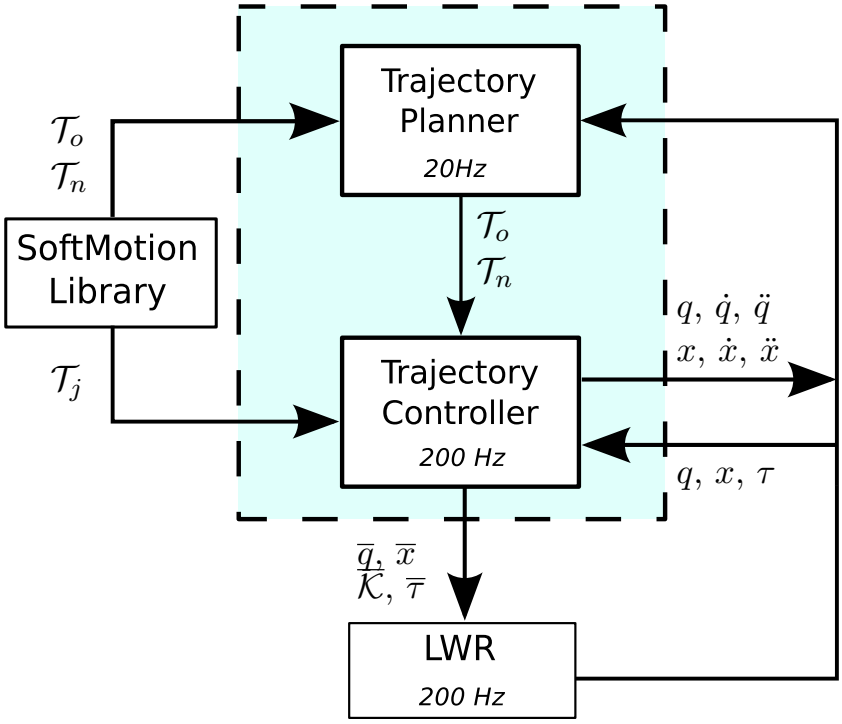


Figure 3.9: An architecture for Reactive Trajectory Control. See Sec.3.3.1 for notations.

The trajectory controller executes the trajectory and communicates with the robot. Its role is mainly to interact with the *FRI* by sending new commands to the robot according to a control strategy and update the informations sent by the robot. It only interacts with the trajectory planner from which it receives new trajectories to execute and the control parameters. In return it reports the control data such as the state of the current motion or the updated position of the robot (q, $\dot{q}$, $\ddot{q}$, x, $\dot{x}$, $\ddot{x}$). Usually it must run at the same frequency as the robot controller, and that frequency must be maximized to have the control data as close as possible to the reality.

Each cycle the robot controller reports the robot's state, and waits for new commands ($\overline{q}$, $\overline{x}$). It possesses internal kinematic limits as well. When receiving a new command, its feasibility according to these internal limits is checked. If it cannot reach the new state without exceeding one of these limits it will stop the motion abruptly. This verification is mainly done for velocity and acceleration, jerk is usually not checked. For robots possessing torque sensors, a similar verification is made for forces. This emphasizes the importance of using trajectories as these verifications can be made easily beforehand.

To summarize, the trajectory planner is an interface with the heart of the planning and control system, while the trajectory controller is more specialized in the interfacing with the robot controller. The first does basic planning and still run fast enough to react instantaneously to unforeseen events. The role of the second

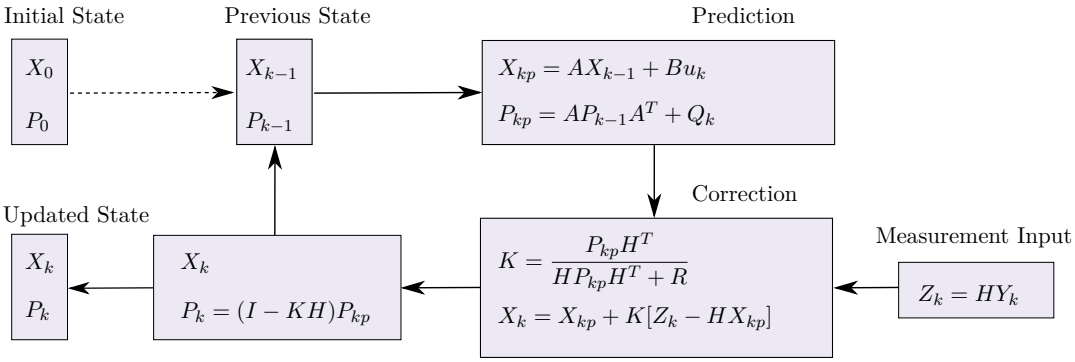
must be minimized in order to maximize its frequency and thus minimize t_c. By playing on the granularity of these two modules, the efficiency of the system can be maximized. Trajectories are used as the main support of communication and facilitate the link between planning and control.

This multi-level functional architecture however presents a risk. By having nodes running at different frequencies, we must keep the synchronization between them. It is also true for the synchronization between the robot controller and the trajectory controller. By maximizing their frequencies, it is harder to keep the synchronization. If a package is lost, the control data are not refreshed. If this happens when a new motion $\mathcal{T}_n$ has to be planned, C_i^n will not be accurate. Then the motion might be stopped abruptly for two reasons. It can be one or more of the kinematic limits that is exceeded because the first command sent was too far off from the real state C_r of the robot. Similarly an imprecise C_i^n can generate a not smooth enough switching trajectory. This can causes the arm to shake during the switching, producing vibrations and excessive torques. These were the main difficulties encountered when we tried to apply our work on a real robot, the Kuka LWR4 (Fig. 3.12).

These difficulties can be overcome at different levels. The first one is to ensure the synchronization by adapting the frequency of each node so that it can maintain it. The addition of timestamps to the communications allows to detect a synchronization problem and to react accordingly. Well-known techniques such as Kalman Filters can be employed to improve the estimation of the current state (Sec.3.3.3). Finally the low-level controller plays an important role by smoothing the junction between two trajectories when a switch occurs, as we will see in the following (Sec.3.3.4).

3.3.3 Current state estimation

During this thesis, our work was mainly applied on a Kuka LWR4 (Sec. 3.4.1). The Kuka *FRI* library only returns position measurements, and we do not possess accelerometers. We then had to find a way to estimate precisely the current velocity and acceleration of the robot from the position measurements. Similarly the current position must be estimated taking into account the delay from the last measurements. A well-known solution is Kalman filtering ([Puglisi , Shaowei 2012, Welch 1997]). For the sake of simplicity, we are using a single dimensional Kalman Filter to estimate the angular coordinates (See Fig. 3.10, Eq.3.1).



X [3×1] : State Matrix
P [3×3] : Process Covariance Matrix
K [3×1] : Kalman Gain Matrix
Q [3×3] : Process Noise Covariance Matrix
B [3×1] : Control Matrix
A [3×3] : System Dynamics matrix
R [1×1] : Sensor Noise Covariance Matrix
Y [3×1] : Measurements Matrix
H [1×3] : Measurement Adaptation matrix
u [1×1] : Controle Variable Matrix (jerk)

Figure 3.10: Depiction of a Single Dimensional Kalman Filter used to estimate the current angular state of the Kuka arm (Fig. 3.12).

$$A = \begin{pmatrix} 1 & t_c & \frac{t_c^2}{2} \\ 0 & 1 & t_c \\ 0 & 0 & 1 \end{pmatrix} \qquad B = \begin{pmatrix} \frac{t_c^3}{6} \\ \frac{t_c^2}{2} \\ t_c \end{pmatrix} \qquad Q = \begin{pmatrix} \frac{t_c^5}{20} & \frac{t_c^4}{8} & \frac{t_c^3}{6} \\ \frac{t_c^4}{8} & \frac{t_c^3}{6} & \frac{t_c^2}{2} \\ \frac{t_c^3}{6} & \frac{t_c^2}{2} & t_c \end{pmatrix}$$
$$P = \begin{pmatrix} \frac{t_c^3}{2} & 0 & 0 \\ 0 & \frac{t_c^2}{2} & 0 \\ 0 & 0 & \frac{t_c}{2} \end{pmatrix} \qquad H = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \qquad R = \begin{pmatrix} 0.05 \end{pmatrix} \tag{3.1}$$

3.3.4 Junction Trajectory

Having a good estimate of the current state of the system is not enough. When the trajectory planner has to generate a new trajectory $\mathcal{T}_n$ at a time t_{Switch}, C_i^n already diverge from C_r. The distance, or error, between C_i^n and C_r at t_{Switch} is composed of two parts : the imprecision of the estimate ε_{imp}, and the offset due to the time delay between the moment C_i^n was updated by the trajectory controller, and the moment the trajectory planner uses it. This delay is mainly function of t_c and t_p.

However $\mathcal{T}_n$ can be executed only at t_{Exec}, the moment the controller receives it. Hence another offset is added to the previous error. ε_{off} denotes the overall

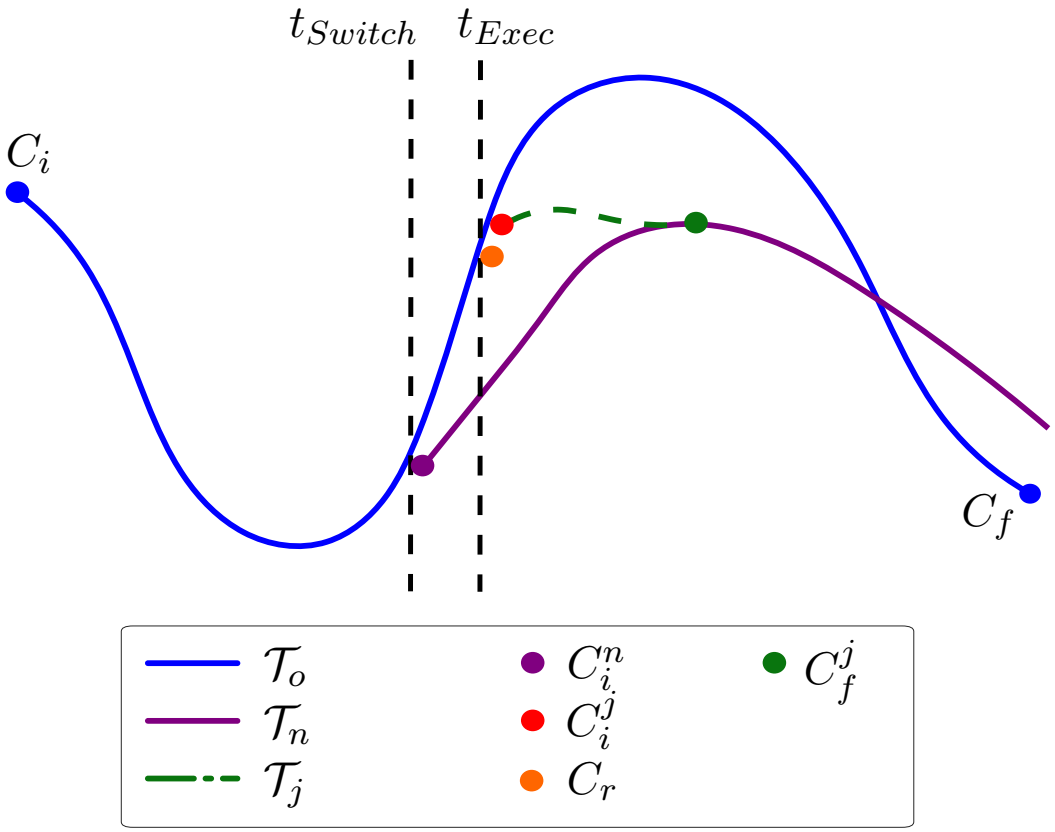


Figure 3.11: Illustration of our control strategy to smooth the junction between two trajectories. The notations are introduced in Sec. 3.3.1.

offset. ε_{err} is the overall error, that is the distance between C_i^n and C_r at t_{Exec}.

$$\varepsilon_{err} = \varepsilon_{imp} + \varepsilon_{off}$$

If the trajectory controller tried to send the commands from $\mathcal{T}_n$ directly to the robot controller, the *FRI* will return an error and stop the motion abruptly for the two reasons mentioned earlier (Sec.3.3.2). A too large ε_{err} can force the *FRI* to return one of two errors : one of the kinematic limits is exceeded because the first command sent was too far from the real state C_r of the robot. Or it can be that the trajectory switch is not smooth enough, making the arm shake in reaction, producing vibrations and excessive torques.

So it is essential to keep ε_{err} as low as possible. It has to be noted that $\varepsilon_{imp} << \varepsilon_{off}$ in the vast majority of the cases. Moreover we have more leverage on ε_{off} than we have on ε_{imp} as we already tuned a Kalman Filter (Sec.3.3.3). A first thought is to make the trajectory controller to compute $\mathcal{T}_n$ in order to vastly reduce ε_{off}. But it is not its role, and if its workload is increased, t_c must be increased as well, and so ε_{off}. Another possibility is to reduce t_p but again there is a limit to that. We need

the trajectory planner to run slower than the trajectory controller. ε_{off} is also vastly due to delay from the two-way communication between the trajectory controller and the trajectory planner. This delay being function of t_c and t_p, reducing t_p only lower it to an extend, but that's not enough to make ε_{err} tolerable.

The solution to the problem is actually a compromise between the two solutions mentioned above. The trajectory controller can produce a junction trajectory $\mathcal{T}_j$ that will link the estimated state of the robot at t_{Exec} to a condition C_f^j belonging to $\mathcal{T}_n$. The role of $\mathcal{T}_j$ is to smooth the junction between $\mathcal{T}_o$ and $\mathcal{T}_n$, and not to replace $\mathcal{T}_n$. We want to conserve the shape of $\mathcal{T}_n$ as much as possible, as well as an eventual time-synchronisation among the axes. We also need a fast computation time for $\mathcal{T}_j$ so that t_c can be kept as low as possible.

The use of the three jerk segments method presented in the next paragraph allows to overcome most of these difficulties (Sec.3.3.5).

3.3.5 Three jerk segment trajectory

Three segments trajectories were first introduced in [Broquere 2010] for trajectory approximation. An imposed time motion between two points involves seven constraints: three initial conditions, three final conditions and the imposed time t_{imp}. A single polynomial segment is only defined by 5 parameters (Sec. 2.2.1.1). Then one segment is not enough. Adding another segment is not suitable also ([Broquere 2010]). Using three segments we have 15 parameters, and we add constraints of continuity between the segments, for a total of 13 constraints.

We have now two choices. We can fix the duration of the segments, either by optimizing the three times or by simply choosing $t_1 = t_2 = t_3 = \frac{t_{imp}}{3}$ with t_{imp} being a multiple of $3t_c$ ([Broquere 2010]). We obtain a system with 13 parameters where only the three jerks are unknown. We have now a solution for an imposed time motion between two motion conditions. If t_{imp} is big enough, a solution will exist, however we have no guarantee that the computed jerks will be bounded.

[Zhao 2014] introduces a variant that fixes the jerk on the first and third segments with $J_1 = J_3$. The unknown parameters in the system are then J_2 and the segments durations.

3.3.6 Construction of $\mathcal{T}_j$

With one of the two variants introduced in Sec.3.3.5 the trajectory controller can compute an imposed time motion between C_i^j and C_f^j. Since the solution is made of only 3 segments it is extremely fast to compute.

There is now two parameters to set before computing $\mathcal{T}_j$: the duration of $\mathcal{T}_j$, denoted t_{imp}, and C_f^j. The choice of C_f^j should be optimized first. This choice is usually a compromise between the smoothness of the junction, and the preservation of $\mathcal{T}_n$ hence we do not want a $\mathcal{T}_j$ too long. Once C_f^j is chosen, t_{imp} can be optimized to be the lower multiple of $3t_c$ that guarantees bounded jerk on $\mathcal{T}_j$.

3.3.7 Closed Loop Reactive Trajectory Controller

The junction trajectory we presented can have multiple functions. Indeed the presented solution is very flexible and can be employed to achieve a variety of objectives. Notably to build closed loop trajectory controllers. Indeed we can correct error and delay by continuously planning a new trajectory in a receding-horizon (model predictive) control fashion. $\mathcal{T}_j$ will then be planned between C_i^j and a motion condition of $\mathcal{T}_o$ further ahead. In our implementation the closed loop controller will periodically measure the error between C_i^j and the theoretical position on $\mathcal{T}_o$. If it exceeds the tolerable error, a short junction trajectory will be planned to correct it. This solution offers flexibility as our controller is then parameterized by the periodicity at which it looks at the error, by the tolerable error, and by the inputs of $\mathcal{T}_j$ that are t_{imp} and C_f^j.

3.4 Experimental Evaluation

3.4.1 Experimental setup

The experimental results exposed in the rest of the chapter are obtained on a real Kuka LWR4 (Fig. 3.12). This robotic arm is commanded in position. It can either be angular or Cartesian commands that are sent every cycle. In the following we plotted the angular trajectories. The arm possesses 7 degrees of freedom making him redundant. As stated earlier the *FRI* only returns position measures, therefore acceleration and velocity are estimated through a Kalman Filter (Sec.3.3.3). The arm possesses torques sensors and can be force-commanded. We can also control its stiffness inside impedance controllers. But we do not need these functionalities in the following. The whole system architecture was already presented in section 3.3.2. For our needs in localization systems we have the choice between the solutions presented in Fig. 3.13.

3.4.2 Non constant motion constraints

3.4.2.1 Illustration of kinematic bounds abrupt decrease for different areas

The trajectory controller runs at $10ms$ while the trajectory planner has a sampling time of $50ms$ (Fig. 3.9). We illustrate our algorithm for different areas in the context of non constant motion constraints presented in section 3.2.3. The results are illustrated in Fig. 3.14 and Fig. 3.15 for one area of each symmetric pair. We present in more details the results for $A2$ and $A4$ by adding the description of $\mathcal{T}_{ext}$ in the phase diagram (see Fig. 3.16 and Fig. 3.17).

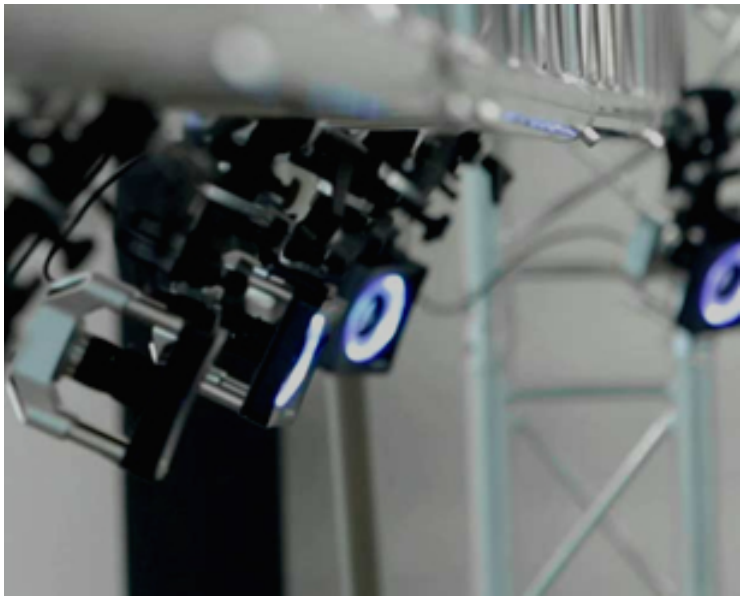
The trajectory planner sends a first trajectory to the trajectory controller. The purple vertical dashed line indicates the moment when the trajectory's limits $\mathcal{L}_{\mathcal{T}}$ are decreased and the current configuration of the arm is outside of $\mathcal{D}$. The trajectory planner must compute a new trajectory to restore the system into $\mathcal{D}$. Following

Figure 3.12: A KuKa LWR4 robotic arm used for our experiments.

Alg. 2 the trajectory $\mathcal{T}_{ext}$ joining C_f according to the updated $\mathcal{L}_{\mathcal{T}}$ is computed and sent to the trajectory controller. The brown vertical dashed line indicates the moment the system is brought back into $\mathcal{D}$, that is the end of $\mathcal{T}_r$ and the beginning of $\mathcal{T}_{opt}$.

3.4.2.2 Example of application

In this experiment (see [video-1]) we are adapting a robot's behaviour according to the presence of a Human (Fig. 3.18). We keep the configuration presented above, and we continue to apply the work presented in section 3.2.3. The robot possesses 3 different behaviours, which are industrial, ergonomic and safe. The human is wearing an helmet that can be tracked by a motion capture system (Fig. 3.13a). Every cycle the trajectory planner measures the distance d separating the human and the robotic arm. Similarly to the work presented in [Sisbot 2010], the trajectory planner will choose to apply one of the behaviour according to d (See Tab. 3.3).



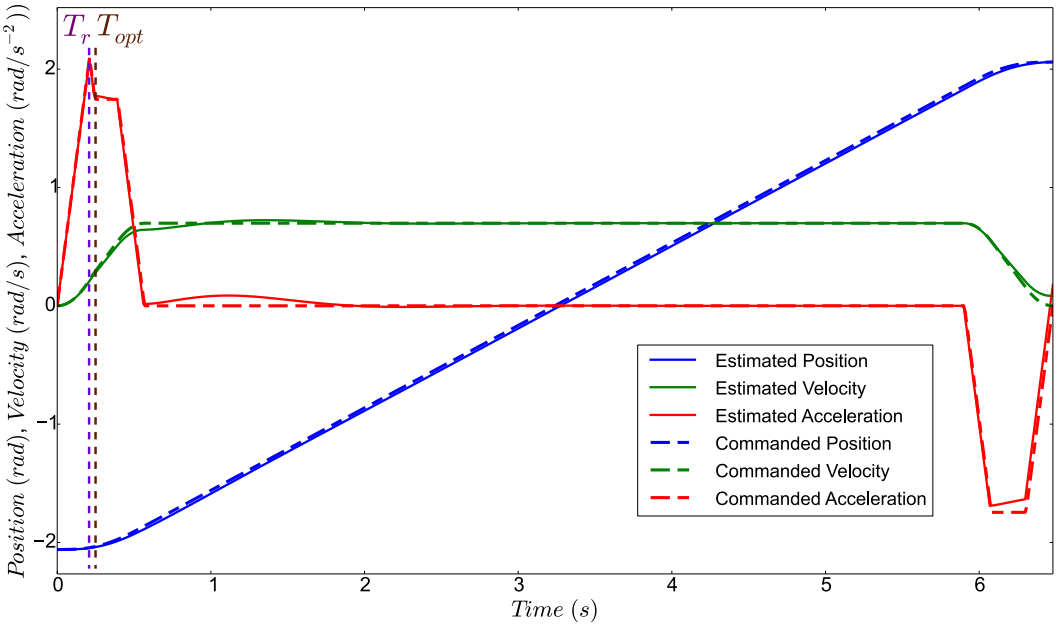
(a) Optitrack, a motion capture system used at LAAS.

(b) Kinect XBOX360 and Asus Xtion. These cameras are mainly used for QR code tracking.

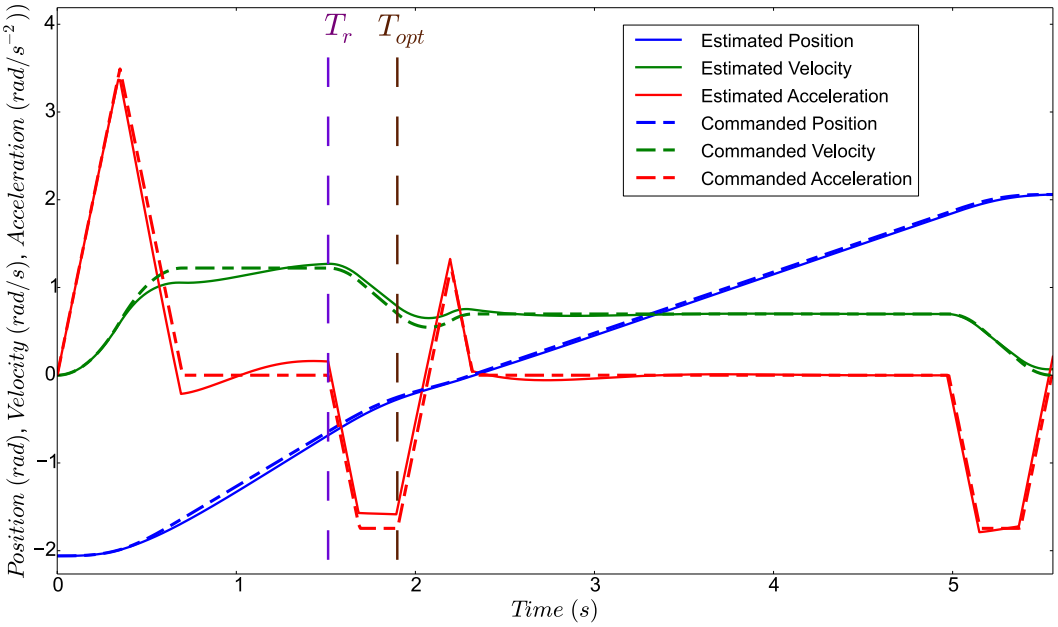
Figure 3.13: The different vision systems used during our experiments.

The industrial behaviour is activated when no human are detected in the robot's vicinity. In this context efficiency is favoured and the robot operates with high velocity, acceleration and jerk. When a Human enters the collision zone, that is in the immediate vicinity of the robot (<2.5 meters), the robot is adopting the safe behaviour and uses low $\mathcal{V}, \mathcal{A}, \mathcal{J}$. As the kinematic bounds are linearly scaled for each behaviour, the robot can be stopped as the velocity reach 0 for a certain d (See Tab. 3.3). Between these two zones the robot adopts an ergonomic behaviour by decreasing its speed and acceleration .

The value depicted in Table 3.3 are quite arbitrary, as we do not know exactly how the kinematic bounds should be set to correspond as well as possible to each type of scenario. In fact they depend on multiple factors and we believe it is essential



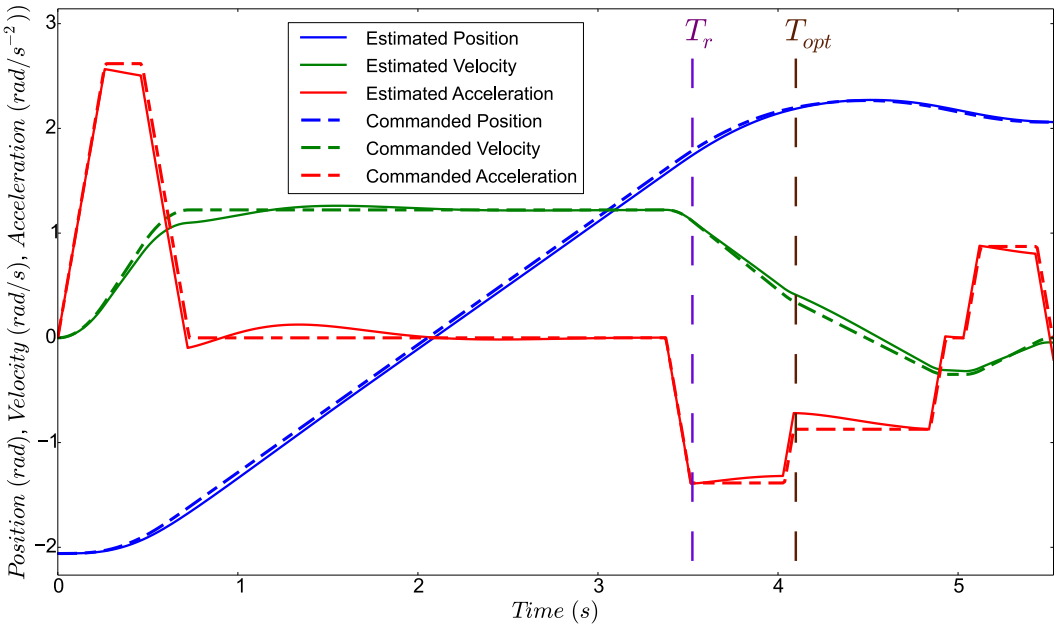
(a) $C_{i'} \in A1$.



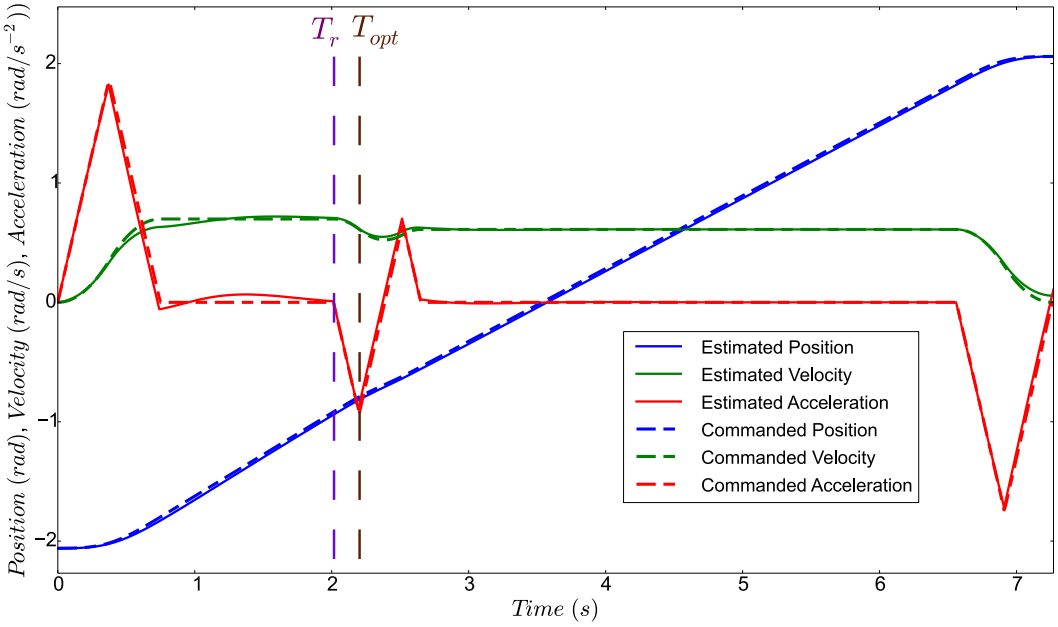
(b) $C_{i'} \in A3$.

Figure 3.14: Illustration of motion constraints abrupt switch on a KUKA-LWR4 joint axis.

to understand their ergonomic properties (this is the theme of a user study presented in section 4.1).



(a) $C_{i'} \in A5$.

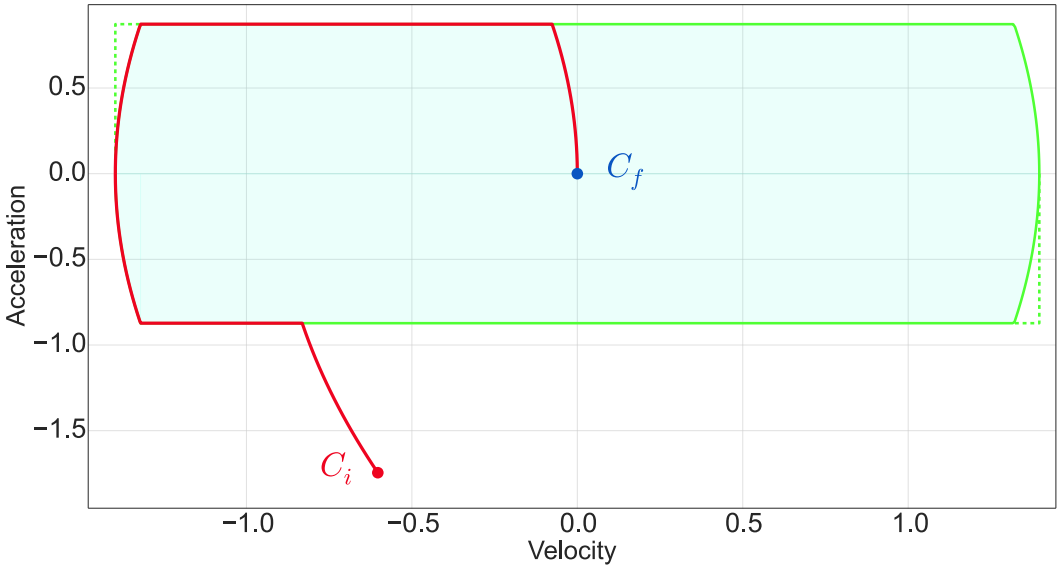


(b) $C_{i'} \in A7$.

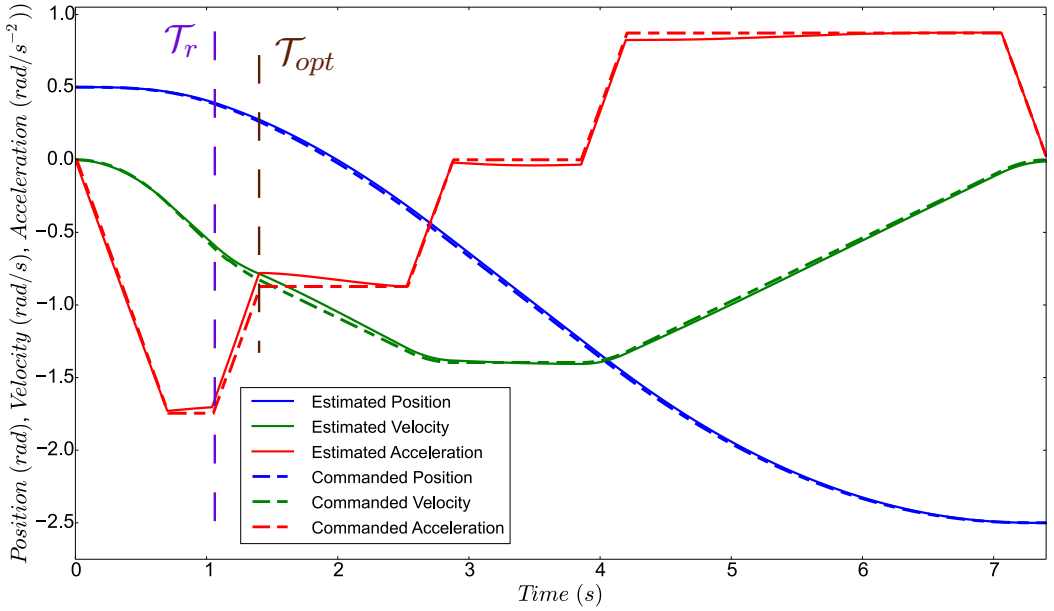
Figure 3.15: Illustration of motion constraints abrupt switch on a KUKA-LWR4 joint axis.

3.4.3 Visual servoing evaluation

In Sec.3.3.4 we proposed a three jerk segments trajectory to smooth the junction between $\mathcal{T}_o$ and $\mathcal{T}_n$. This solution is put to the test within a visual servoing experiment (see [video-2]). We keep constant motion constraints as the aim is to



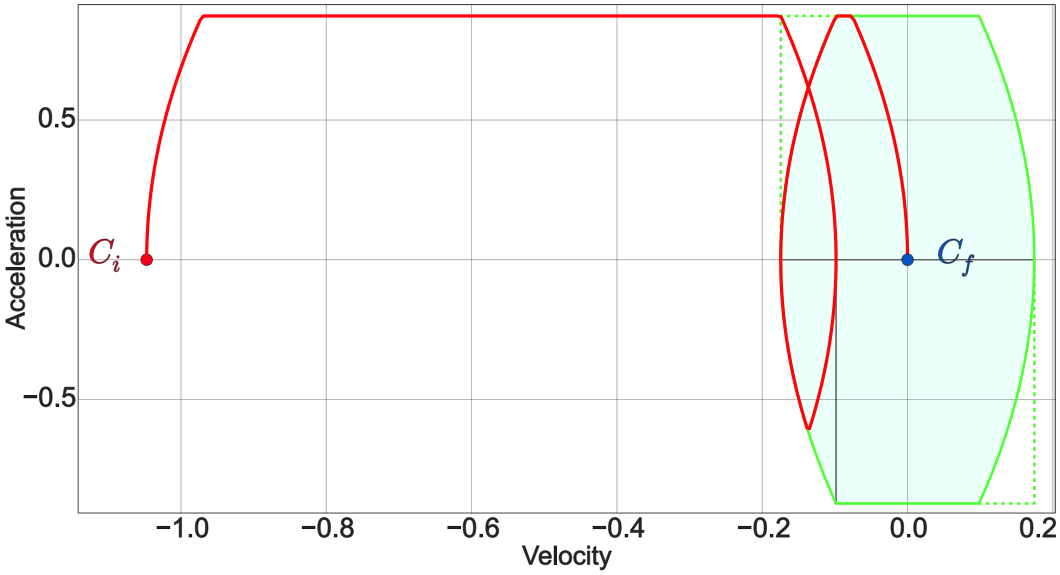
(a)



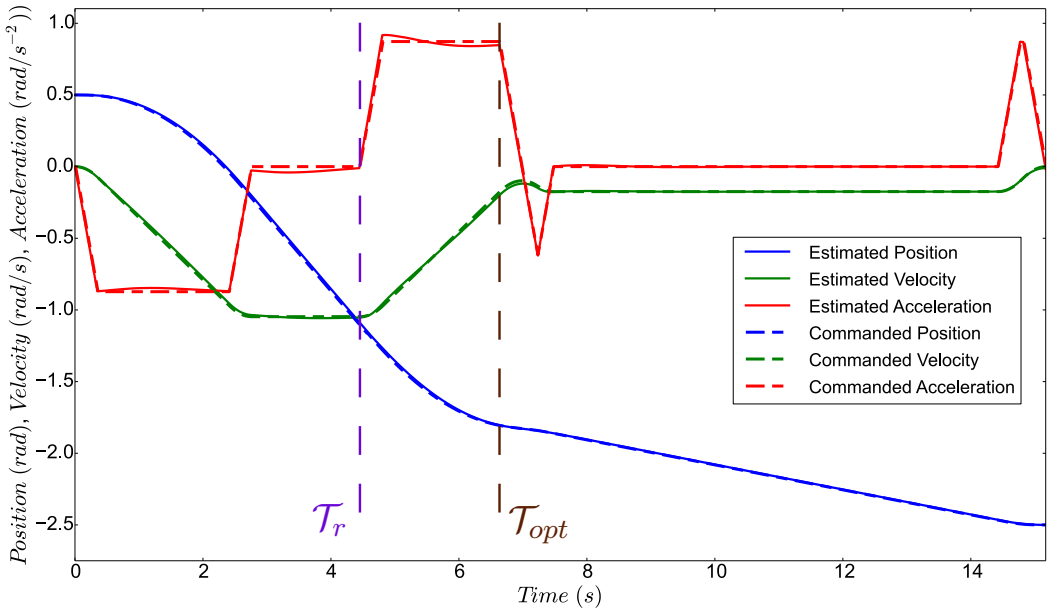
(b)

Figure 3.16: Acceleration bounds are decreased and the current configuration of the arm belongs to $A2$. $\mathcal{T}_r$ is made of a positive jerk segment reaching A_{min} at $t = 0.7s$.

evaluate the work presented in Sec.3.3.4. During this experiment the end effector of the robotic arm has to follow a wooden board held by a person (Fig. 3.19). The system architecture is the one presented in Sec.3.3.2. The trajectory controller runs at $200Hz$ and the planner at $10Hz$. To follow the board, the trajectory planner recomputes a trajectory every cycle according to the updated pose of the board. The aim is to keep the end effector in front of the board.



(a)



(b)

Figure 3.17: Speed bounds are drastically decreased, and the arm is in an invalid configuration belonging to $A4$. A new trajectory is computed to bring the system into $\mathcal{D}$.

The board position can be tracked by placing three AR tags on it. The ros package ar_track_alvar can identify and track the poses of multiple AR tags that are each considered individually. The Cartesian coordinates of these tags are then expressed in the kinect frame $\mathcal{F}_{cam}$. By calibrating $\mathcal{F}_{cam}$ relatively to the arm's end effector frame $\mathcal{F}_{ee}$ we can retrieve the Cartesian pose of each AR tags in $\mathcal{F}_{ee}$ and

Figure 3.18: Experiment illustration (see [video-1]). The robot is accomplishing a task while a human is walking towards it. The trajectory planner is informed by the motion capture system (helmet) of the human presence and adapts its behaviour accordingly.

build the frame associated to the wooden board $\mathcal{F}_{board}$, expressed in $\mathcal{F}_{ee}$.

From the desired pose of the end effector $\mathcal{P}_{board}$ in $\mathcal{F}_{board}$, we retrieve the Cartesian goal position $\mathcal{P}_{arm}$ for the generated trajectories (Eq.3.2).

$$\mathcal{P}_{arm} = Th_{[arm,ee]} * Th_{[ee,board]} * \mathcal{P}_{board} \tag{3.2}$$

Every cycle the trajectory planner plans a new point-to-point motion at the condition that the board is visible and the motion feasible. This motion links $C_{i'}$ to C_f that is built from P_{arm} after transformation to angular coordinates with the inverse kinematic model.

The experimentation can be kept running for a long time while the trajectory planner runs at $10Hz$. We can then exert a lot of stress on our system, and doing so, be able to evaluate the faculty of $\mathcal{T}_j$ to effectively smooth the junction between $\mathcal{T}_o$ and $\mathcal{T}_n$.

Robot's behaviour	Safe	HRI	Industrial
$d(meters)$	< 2.5	< 6	> 6
$\mathcal{V}$	$0 < ... < 70\%$	$= 70\%$	70%
$\mathcal{A}$	$0 < ... < 10\%$	$10\% < ... < 40\%$	$= 40\%$
$\mathcal{J}$	$0 < ... < 10\%$	$10\% < ... < 40\%$	$= 40\%$

Table 3.3: Mapping of the distance separating the human and the robot to the robot's behaviour. For a defined behaviour, $\mathcal{V}, \mathcal{A}, \mathcal{J}$ are linearly scaled between two bounds according to d. These bounds are a percentage of the maximum kinematic values that are defined internally. In our experiment the HRI behaviour of the robot is defined by a constant velocity fixed at 70% of the absolute permissible velocity. Acceleration and jerk are linearly scaled between 40% and 10% of their maximum values, 40% when $d = 6$ and 10% when $d = 2.5$. In this experiment the kinematic bounds are symmetric.

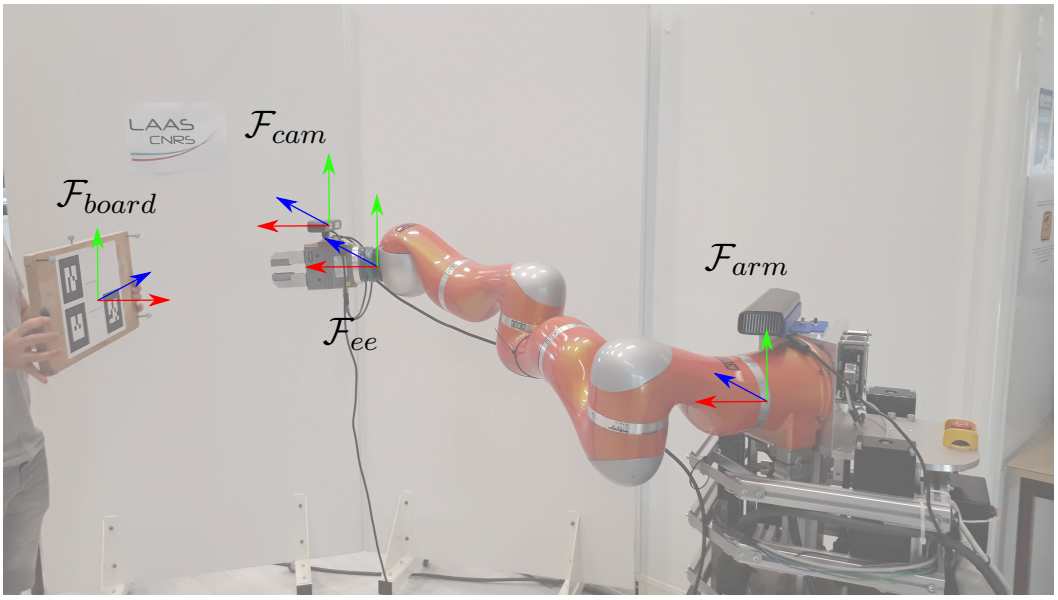


Figure 3.19: Visual servoing experiment (see [video-2]). The arm's end effector is following the wooden board. The board is tracked by a Kinect camera placed on its gripper.

Before the use of a junction trajectory we tried to lower t_p and make the trajectory planner adjust $C_{i'}$ by estimating the offset between the time at which $C_{i'}$ is updated by the trajectory controller and the time at which $\mathcal{T}_j$ is received and executed. However this solution gave poor results as most of the time the arm was stopped for the reasons mentioned in section 3.3.4. When the arm was not stopped we could hear and notice the arm shake.

With the junction trajectory this is no more the case as the transition between $\mathcal{T}_o$ and $\mathcal{T}_n$ is not noticeable. The arm is not shaking or emitting undesirable sounds.

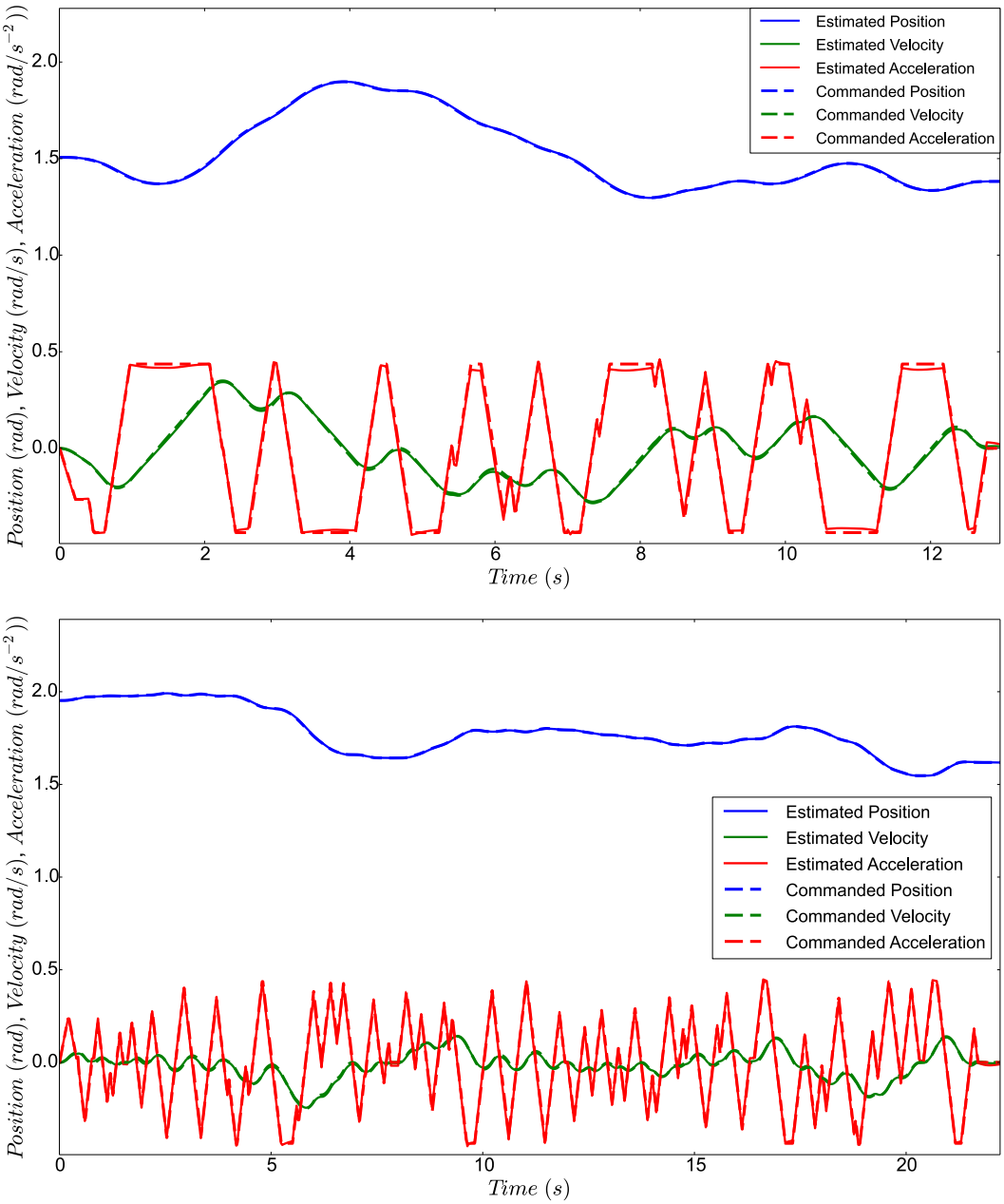


Figure 3.20: Evolution of angular trajectories during some visual servoing experiments. As the trajectory planner runs at $10Hz$, trajectories are replanned multiple times while the robot is moving. Junction trajectories are used to smooth the trajectories switch.

We plotted the evolution of some joint axes during the experiments (see Fig. 3.20, Fig. 3.21, Fig. 3.22). As we can see the arm is able to follow its reference trajectory. It has to be noted that during the experiment the system could exit $\mathcal{D}$, and in this

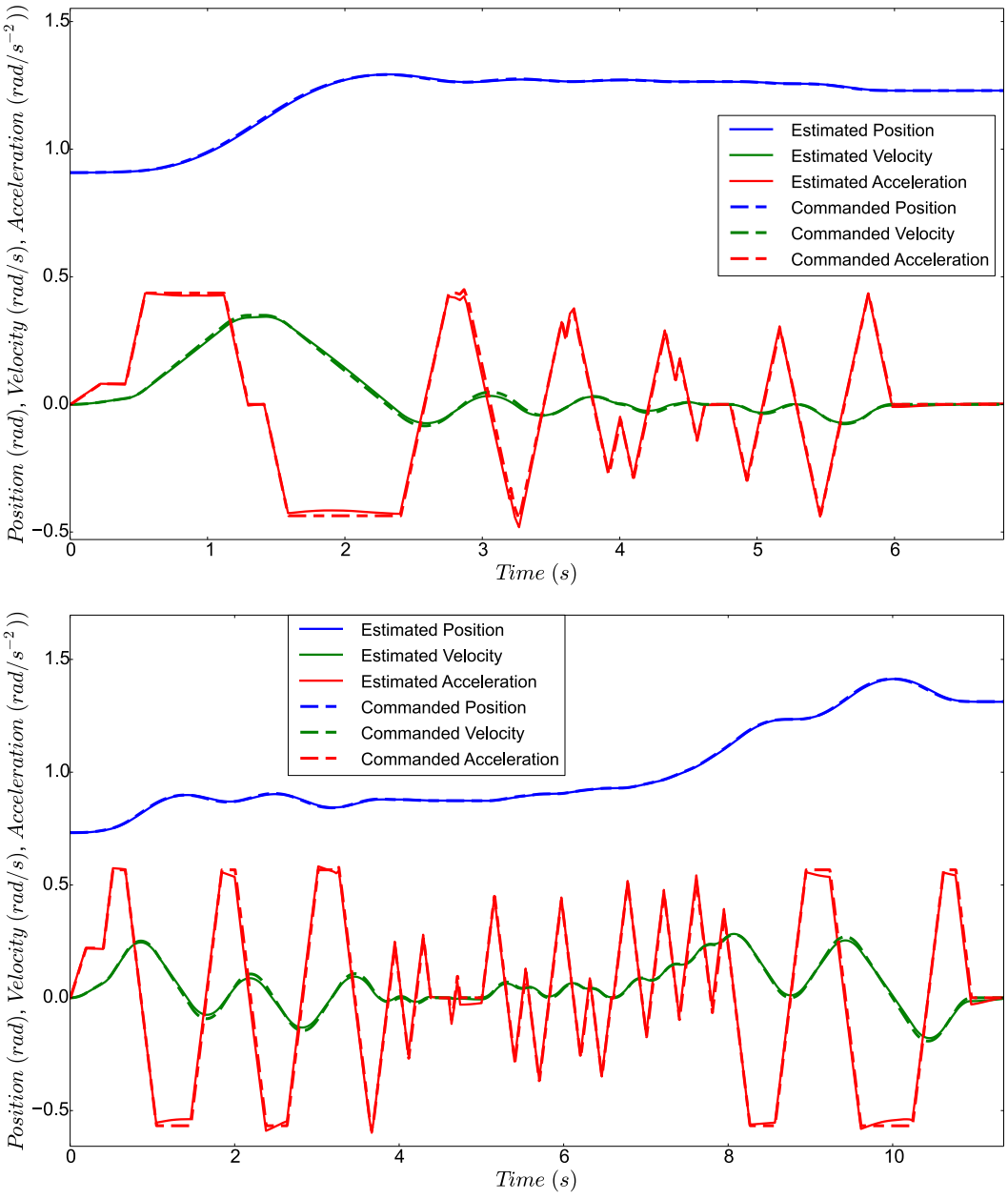


Figure 3.21: Evolution of angular trajectories during some visual servoing experiments. As the trajectory planner runs at $10Hz$, trajectories are replanned multiple times while the robot is moving. Junction trajectories are used to smooth the trajectories switch.

case $\mathcal{T}_r$ was computed following Alg.2.

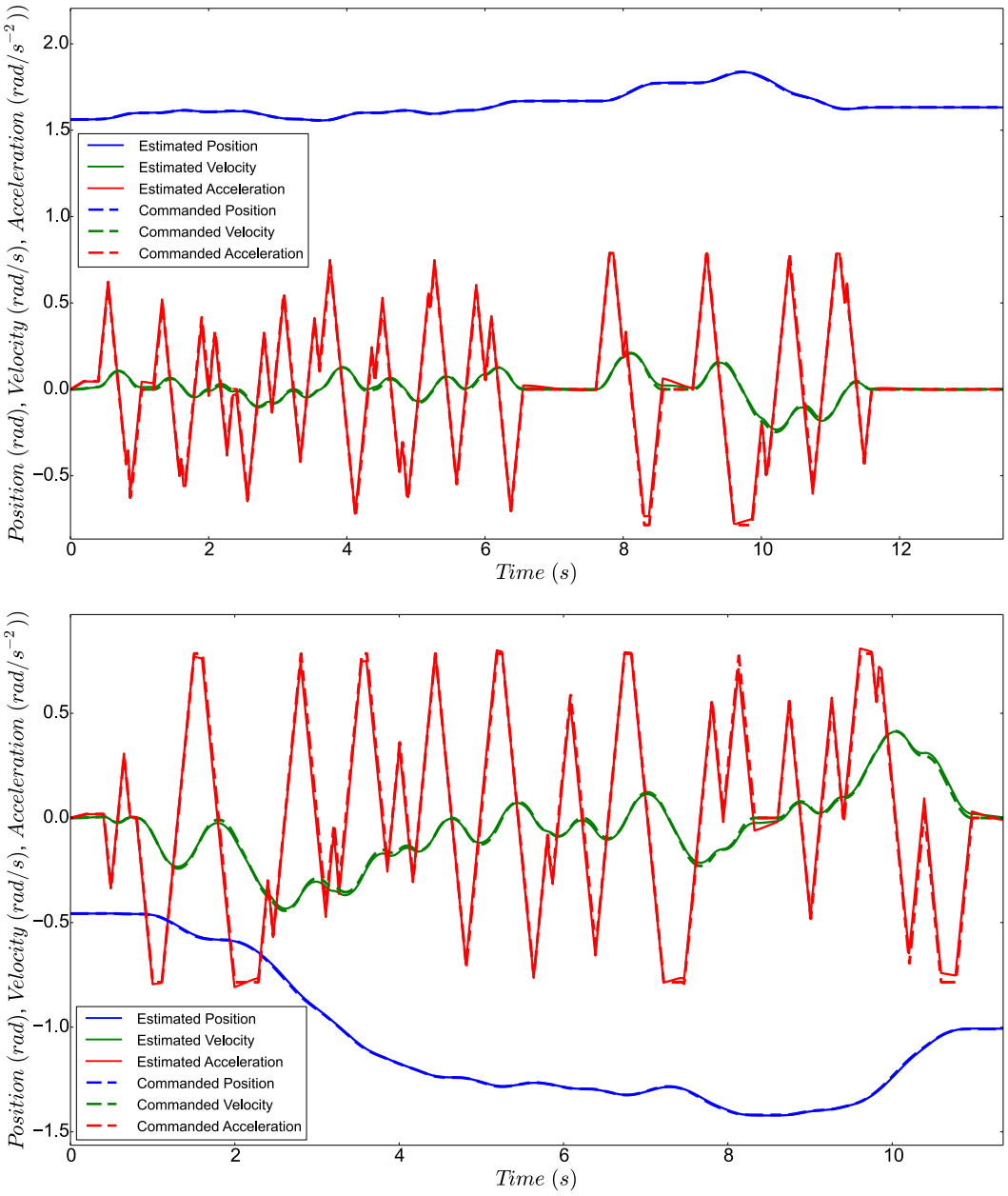


Figure 3.22: Evolution of angular trajectories during some visual servoing experiments. As the trajectory planner runs at $10Hz$, trajectories are replanned multiple times while the robot is moving. Junction trajectories are used to smooth the trajectories switch.

3.5 Conclusion

This chapter presented an extension of the previous algorithm to cope with non admissible initial configurations. The adopted solution is to generate a new trajec-

tory to bring back the system under the kinematic limits from where the previous algorithm can recover its functionality.

The system can be put outside of the kinematic bounds for two reasons: reduction of the motion constraints in reaction to an unforeseen event, or control related issues. We propose two different version of the algorithm that take into account these factors to propose an adapted solution.

This extension is essential for the HRI field, as a robot that interacts with humans needs to constantly adapt its behaviour. This can be done by reducing its speed to preserve the physical safety, or by limiting the jerk and acceleration to produce smooth motions that are more able to provide good ergonomics and preserve the psychological safety of the human.

This extension raised a more general problem in reactive trajectory control. Switching from a trajectory to another, hence when the robot is not in a rest configuration, can be difficult. When the robot receives a new trajectory, he is no more in the configuration that was used to compute the trajectory since he is moving. The offset between the theoretical position and the real one can force the robot controller to stop the motion abruptly. We then propose a control strategy to smooth the junction between two trajectories at the trajectory controller level. This solution takes benefits from the proposed control architecture. Having a functional layer subdivided in a trajectory planner and a trajectory controller improve the efficiency of the system. The trajectory controller can run faster and dispose of better estimations of the current state of the system. The use of trajectories as a support of communication improves the link between the different components of the architecture. The simplicity and flexibility of trajectories has been demonstrated. From the general problem of computing a motion between two configurations, they have been used to restore a system to a valid state as well as contributing to an efficient reactive control by facilitating the communications between a trajectory planner and a trajectory controller, and finally they can be used for general closed loop control.

These works have been validated by experimental results and contributes to the field of reactive trajectory control. We have demonstrated the importance of these contributions for Human-Robot Interactions as we need non-constant motion constraints. Nonetheless we do not know how these constraints should be adapted to answer the best to each situation. We do know that velocity plays an important role in safe interactions. Smooth trajectories seems to offer good ergonomics properties, hence we need a limited jerk to preserve the psychological safety. But we still have the questions of the values and the relationship between the jerk, the acceleration, the velocity, and how they each contribute to define qualities for the motion. It would be interesting to know how to set these values according to desired quality for motions, for example gracious motions. It could also be used to adapt the robot to the personal preferences of individuals, or categories of individuals. For example a robot could be slower and smoother around old persons. Knowing how to define a motion according to a quality will ease the link between planning and control and the communications could be simplified further. This question will be discussed in

the next chapter in the presentation of an HRI experiment.

Chapter 4

Experiments

Contents

4.1 User study : Ergonomic Properties of Motions

4.1.1 Context

In the previous chapter we extended our OTG so that invalid initial configurations can be considered. This work enables the use of non constant motion constraints. Non constant motion constraints is an indispensable feature for Human-Robot Interactions. To preserve physical safety of humans during interactions the system must be able to adapt in real-time its velocity according to the human proximity. The same can be say for the psychological safety, the robot must adapt its behaviour to the human presence and preferences. The contribution presented in the previous chapter constitutes a major step towards the improvement of Human-Robot physical Interactions. Trajectories offer better descriptions of motions and

allow to add properties and qualities to enhance these descriptions. As a consequence they improve the communication between the decisional, executional and functional layers.

When the aim is to guarantee physical safety, we do know that velocity is the main factor. Many studies covered the theme of safety via the analysis of injury mechanisms (Sec. 1.2.3.2). Some velocity thresholds have been proposed to preserve humans from injuries when collisions occur.

However, when the aim is to preserve the psychological safety, we do not know how these constraints should be adapted to answer the best to each kind of situation. Limiting the jerk by generating smooth trajectories has been used as a generic approach to provide comfort during interactions. Some works also proposed to minimize the acceleration. We do not know if these restrictions make sense when the velocity is low. We do not know how each kinematic variable interacts with the others. For example there could be a ratio to preserve between the constraints. There are many questions for which we have no answers. Works on the subject are very limited (Sec.1.2.4.2).

In response, we decided to investigate the motion's ergonomics by conducting a user study which was conceived in collaboration with an ergonomist intern, Yuliya Zdanchuk. Yuliya's role was to propose tools to evaluate the user experience during the experiment. The objective was to study the role of kinematics in the ergonomic properties of motions, through user experience.

A preliminary work addressing the ergonomics properties in relation with the kinematics of a robotic arm in the presence of human is detailed in the following.

4.1.2 Methodology

This study made participants collaborate with a robotic arm in order to realise a simple task within an industrial settings. The task is described in Sec.4.1.4. This task was accomplished multiple times by the participants. For each iteration of the task, the only varying parameters are the kinematic constraints of the motions: jerk, acceleration and velocity. The subject had access to a computer with an interface allowing him to tune the parameters. The parameters were anonymous, and the subject had no information given on the nature of these parameters. This interface is showed in Fig. 4.1. In this interface the parameters are:

- A: Acceleration
- B: Jerk
- C: Speed

For each parameter 3 values are possible. They correspond to a percentage of the maximum value the kinematic bound can take, in that case the maximum allowed values by the Kuka arm controller. The kinematic bounds are symmetric for this experiment. The maximum velocities are given by the manufacturer. For jerk and

Figure 4.1: Screen shot of the interface used by the participants to set the kinematic bounds. A was for acceleration, B for jerk, C for speed. Participants didn't had this information. We can also see a launch button used to begin a new test, and an eval button that opens a new Godspeed questionnaire to fill.

acceleration we had to estimate those values through tests. For acceleration and jerk the values are 10%, 50% and 100% of the maximum allowed. For velocity it is 40%, 70%, 100%. These values were determined thanks to pre-tests. For example, with a too low velocity the task was judged extremely boring.

Once the participant has finished choosing the parameters he could launch the task via the interface. The study was divided in 3 parts:

- The first part begin with an habituation test with all parameters at the maximum. After this follow three imposed tests. In each of these three tests, one parameter is at the minimum and the others at the maximum. Evaluations E.1 and E.2 were conducted (Sec.4.1.2) after each test. These imposed tests allow to evaluate each variable independently. Moreover they give an idea of the purpose of each variable to the user, allowing him to be faster on the rest of the experiment. The subject was then asked to give its favourite motion among those, and to justify its choice.

- In the second part the subject is free to choose the parameters and can do as many tests as he desires. He must stop once he finds the combination of parameters that are the best for him. After each test the subject must fill the Godspeed questionnaire (E.2). We found inspiration from a previous user study we conducted for this part. This previous study was made in collaboration with Nathan Compan, another ergonomist intern. The subject

Figure 4.2: The robotic arm used to accomplish the task.

had to observe a sequence of tasks where the parameters where imposed. The evaluation took place at the end of the study and the results were difficult to use. The protocol presented a major drawback: since the evaluation was made at the end, the subject didn't remembered the majority of the previous tests. The subject was also not active, and they often felt bored and less involved the further the study progressed. By allowing the participant to take an active role we hoped to improve its implication and concentration throughout the experiment. Moreover since he had complete freedom on the choice of parameters, with no constraints on the number of attempts, we can hope that the choice made matches the preferences.

- Once the participant found its favourite parameters, a semi directive interview took place (E.3).

The subject only had to tune the parameters, launch a test, fill a questionnaire and repeat the process until he found its motion of choice. The system knew when the tin container was placed inside the box by the participant thanks to torque measurements and vision. The hand used by the participant to deposit the object was also tracked by a motion capture system, mainly for security reasons.

4.1.3 Evaluation

We have decided to adopt three techniques to evaluate interactions:

E.1 Sentence completion

E.2 Godspeed questionnaire

E.3 Semi directive interview

We have adopted metrics related to socio-cognitive skills to measure the user's emotional state, through the use of questionnaires.

Godspeed questionnaires are often used to measure the user's attitude towards a system. The first problem encountered was related to the type of questionnaire to be adopted. Developing a valid questionnaire can take a considerable amount of time and the absence of standardization makes it difficult to compare results with other studies. It is why we decided to adopt standardized measurement tools for HRI, in addition to some measures that we found interesting for our research. As part of our survey we adopted the Godspeed questionnaire ([Bartneck 2009]) which uses semantic differential scales to evaluate the attitude towards the robot. Such a questionnaire contains questions (variables) on five concepts (latent variables): anthropomorphism, animation, sympathy, intelligence perceived and perceived safety. Perceived safety is a measure of the user's comfort level during the interaction with the robot as well as the perception of the level of danger. The variables were: anxious - relaxed, agitated - calm, serene - surprised (reverse item). The questionnaire have sufficient internal consistency and reliability. To confirm this, we have calculated the Cronbach's alpha for the latent variable perception security (High Cronbach Alpha coefficient: perceived safety; $\alpha = 0.845$).

Sentences completion have also been introduced on a particular dimension of the interaction which is the perception of the robot's movement. The objective was to evaluate the user experience in a qualitative way. The user's experience being subjective and complex, this method has the advantage of allowing the participants to express themselves freely, which is not the case with questionnaires.

A Semi-directive interview was conducted at the end of the experiment. Open-ended questions on the choice of the preferred movement were also introduced to gather informations on the reasons for this choice.

4.1.4 Task Description

We used a Kuka arm mounted on a table. A receptacle box is fastened to the end of the arm (See Fig. 4.2). The participants sat at a table in front of the robotic arm. The task for the participant consisted in deposing a tin container into the receptacle box.The complete task is made of 4 steps:

S.1 The arm positions the receptacle box in front of the participant so that he can easily place the tin inside the receptacle box.

S.2 The subject deposits the tin inside the receptacle box.

S.3 The arm empties the receptacle box in a bin at the opposite side of the table.

S.4 The arm comes back at its home position of S.1.

The setup was designed to simulate an industrial assembly chain and recreate a work environment.

4.1.5 Objectives

We had a few hypothesis to confirm, and a lot of interrogations that we mentioned in Sec.4.1.1. The collaboration between roboticists and ergonomists is recent, and there is no questionnaire adapted to the criteria we wanted to evaluate. Creating a new questionnaire would take a considerable amount of time and it was not in the scope of this thesis. Thus, this user study aimed to be a preliminary work and we were expecting to encounter difficulties. However it was seen as a way to improve our tools to evaluate Human-Robot Interactions. Above these constraints, our objective was to propose a tool to improve the conditions of interaction between a manipulator robot and a human operator.

The hypothesis and interrogations listed bellow are related to the choice of the participants favourite motion.

Hypothesis:

H.1 The subjects will choose to minimize the jerk.

H.2 The subjects will maximize the velocity.

Interrogations:

I.1 Will the subjects minimize both acceleration and jerk?

I.2 Do acceleration and jerk minimizations offer similar ergonomic properties? Are they interchangeable?

I.3 Is there one motion, ie one set of parameters, that will be dominant?

I.4 Will the subjects maintain a ratio between all three parameters?

4.1.6 Results

The user study took place at LAAS. The only condition to participate was to be fluent in french. We got a total of 21 subjects not including pre-tests. 8 were women (38.1%). 12 had no experience with robots prior this study, 6 close to none. Only one had experimented with a robotic arm before.

Unfortunately, a part of the results were unusable. E.1 had a flaw in its construction that could lead to different interpretations for some questions. We only analyse the choice of the favourite imposed motion for E.1. Using a Godspeed questionnaire after every test was also too much, the participants started to loose implication and interest quickly. Hence the Godspeed questionnaires were often filled mechanically. Thus we retain no results from E.2.

We will only evaluate the choice of the favourite motion from the imposed tests and from the subjects free choices.

4.1.6.1 Imposed tests

Three imposed tests were proposed to the subject at the beginning of the experiment. In each of these three tests, one parameter is at the minimum and the others at the maximum. Once he had performed these three tests, the subject had to choose its favourite among them and explain its choice. Participants preferred the motion with minimum jerk as it is shown in Fig. 4.3. When they had to justify their choice, for both the minimum jerk and the minimum acceleration the main reason mentioned was that the motion was smoother. For the choice of minimum velocity there is not an obvious reason. They found the motion less worrisome and less noisy. These justifications are summarized in a word cloud (See Fig. C.2)), however we chose to not translate those from french to not compromise the meaning.

4.1.6.2 Motion of choice

Having three parameters, and three values for each, makes a total of 27 different motions. Since we had only 21 subjects, the interrogation I.3 can be seen as too optimistic. Indeed we can see in the cross tabulation of Fig. 4.4 that no motion was chosen more than twice.

If we analyse each parameter independently with Fig. C.1 we can notice that the velocity was minimized by only 2 subjects. The minimum velocity was often judged boring and inefficient by the participants, even thought we took this parameter into account after the pre-tests.

The jerk was preferred at a medium value by the majority. Only 4 times the acceleration was minimized. Never jerk and acceleration were both minimized, and only once they got both maximized. This is interesting as acceleration was maximized 9 times and jerk 6 times. When acceleration is maximized the jerk is preferred medium for the majority of subjects. Same when the jerk is maximized, the acceleration is also preferred medium. 9 times out of 21 either jerk or acceleration was maximized and the second kept at a medium value.

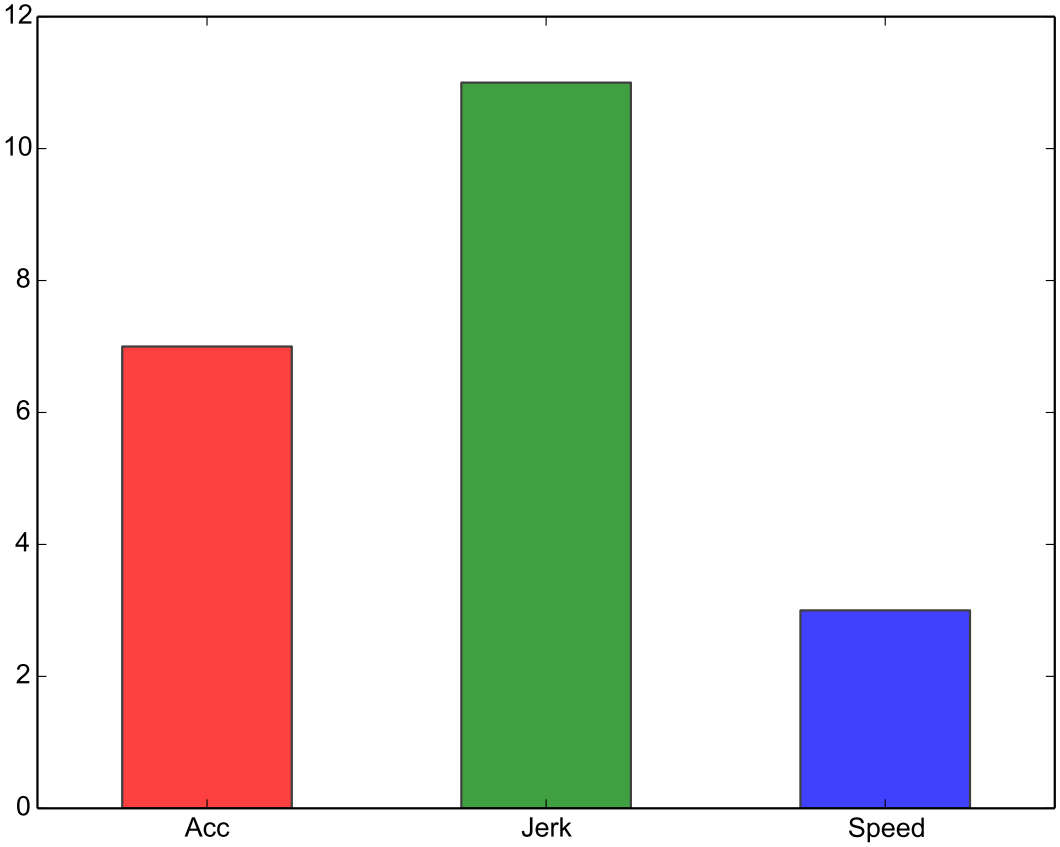


Figure 4.3: Favourite imposed test. Three imposed tests in total with one parameter minimized each time while the others are maximized (See Fig. 4.1). The test with minimum jerk is clearly preferred. The reasons are depicted in a word cloud in Fig. C.2.

This could mean that participants have difficulties to differentiate the perception of the acceleration and the jerk, hence they could have similar properties. This is confirmed by the word cloud of Fig. C.2. Subjects found that minimizing either jerk or acceleration increased smoothness of the motion. However it appears that smoothness is never maximized, but either minimized above a constraint. The constraint being either jerk or acceleration at a medium value. This with the fact that velocity is never maximized could induce that the participants preferred a performance based motion under the constraint of smoothness. This theory is supported by the answers given by the participants when we asked them to motivate their choices. Their main motivation was to make a compromise between speed and comfort. The second motivation was the smoothness of the motion. Their motivations are summarized in a word cloud in Fig. C.3.

Finally to the question "What parameter had the most impact for you?" asked during the semi-directive interview, the answer was acceleration for the majority

Jerk	Max			Medium			Min		All
Acceleration	Max	Medium	Min	Max	Medium	Min	Max	Medium	
Velocity									
Max	0	2	0	2	1	2	2	1	10
Medium	1	2	1	2	1	1	1	0	9
Min	0	0	0	1	1	0	0	0	2
All	1	4	1	5	3	3	3	1	21

Figure 4.4: Cross tabulation (Velocity×Acceleration×Jerk) showing the different motions chose by the participants as their favourite. The reasons are depicted in a word cloud in Fig. C.3.

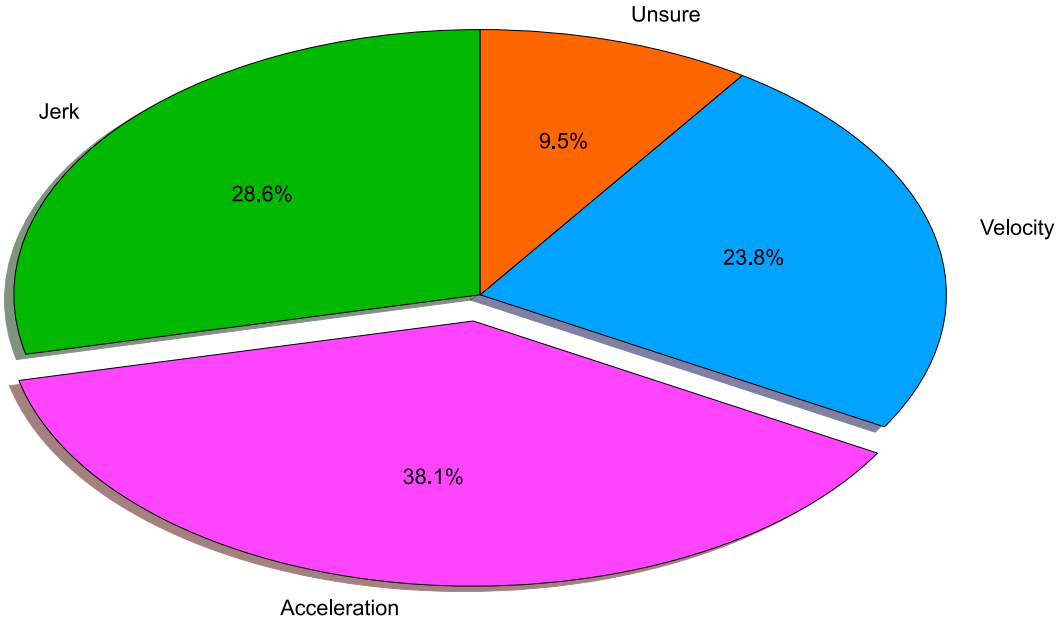


Figure 4.5: Answers to the question: "What parameter had the most impact for you?" during E.3.

(See Fig. 4.5).

To summarize:

- The first hypothesis H.1 is not confirmed by this study. The majority of participants chose a medium jerk.

- H.2 is validated to an extent. Maximum velocity was the number one choice, but medium values came close. One thing seems clear, it is that a high velocity is desired from an ergonomic point of view during an interaction.
- We got an answer for I.1 as never acceleration and jerk were both minimized by the participants. This could induce that maximizing smoothness is not desired from a robot.
- It appears from the results that the interrogation I.2 was legitimate. However this study is not sufficient to draw conclusions.
- More participants would be needed to answer I.3 and I.4.

4.1.7 Conclusion

We presented a user study to investigate the role of kinematics in the ergonomics properties of motions. This work is in adequacy with what we presented in the previous chapter, that is non constant motion constraints. With a trajectory generation algorithm taking into account non constant motion constraints, a robot can adapt its behaviour in real time according to the situation. The situation might imply to preserve the physical safety or the psychological safety of a human co-worker. Relations between physical safety and velocity on the robot's motion are established. However we know little on the relation between kinematics and ergonomics. Thus we proposed this study to investigate this relation trough user's experience. This study aimed to be a preliminary work in collaboration with ergonomists.

The most interesting result is that participants preferred performance based motions under the constraint of smoothness. Acceleration and jerk seems to have similar roles, i.e. their minimisation is a way to increase the smoothness of the motion. This property however is more apparent on the jerk.

We cannot draw any definitive conclusion from this study and more in-depth analysis are required. Giving a more active role to the participants constituted an improvement as we felt more involvement. We also learnt that too much questionnaires is counterproductive since at one point participants responded mechanically. There is a need for new tools too evaluate Human-Robot interactions, especially adapted questionnaires.

Finding organizing principles to define ergonomics properties of motions would lead to a better description of motions and improve the link between decisional and functional layers. Despite the fact that we don't have these organizing principles, personal preferences of individuals can be used directly as inputs to the trajectory generator we presented.

4.2 Autonomous door-opening

4.2.1 Introduction

The completely autonomous opening of a door was one of the experimental works carried out during this thesis. The objective was to open a distant door localized by a vision system in an autonomous manner. The autonomous opening of a distant door is a complex task requiring several components. From the hardware perspective we need a vision system to localize the door and its handle, a mobile base to reach the door, a robotic arm to reach the handle and open the door, as well as a tool to grasp the handle. The software architecture is complex as well: we need a task planner, a supervisor, the localization, the base controllers, the arm controllers and more. This project has been accomplished during the last year of this thesis, and contributes to the validation of the work presented in chapter 2 and chapter 3. We briefly present it in the following.

For a video of the experiment see [video-3].

4.2.2 The robotic platform Jido

This task was realized on Jido, a mobile manipulation platform (Fig. 4.6). Jido is made of several components:

- A mobile platform from Neobotix.
- A robotic arm Kuka LWR IV already introduced in Sec.3.4.1.
- A camera kinect xbox360.
- A gripper from SCHUNK.

We present the architecture and the software components in the following.

4.2.3 Jido's software architecture

The Jido's architecture illustrated in figure 4.7 is made in its integrity of ROS nodes, some being generated with GenoM. The arm's trajectory controller and planner have been presented in previous chapters. The base controller is regrouping the roles of these two nodes into one for the base. The gripper's controller is a ROS node interacting with an arduino to send OPEN/CLOSE commands to the gripper. The supervisor is a module written in GenoM which embodies both the decision and the execution level. Its role consists to supervise the execution of the whole task and the synchronization of the diverse components. We detail more the vision components in the following.

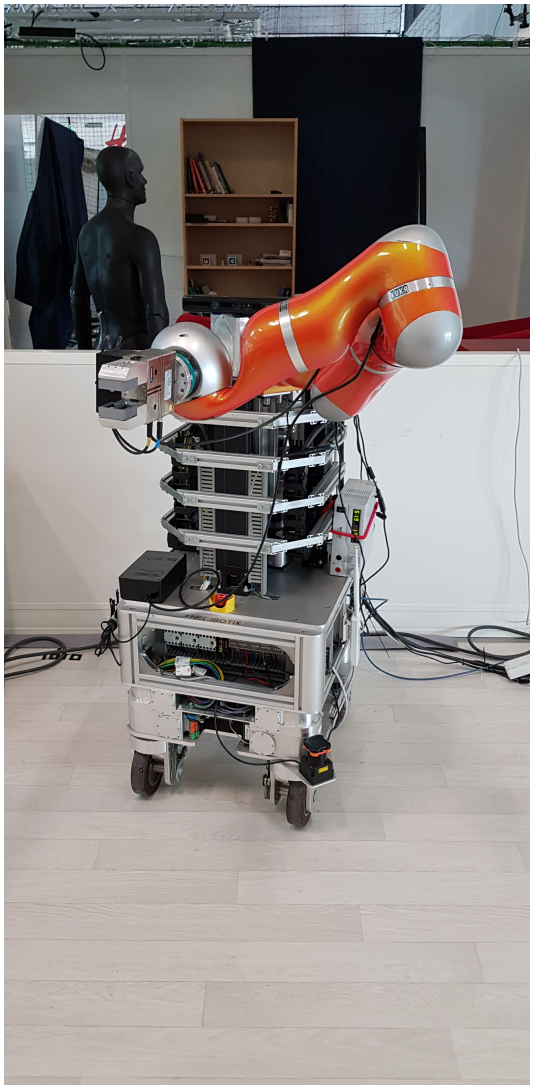


Figure 4.6: Jido, a mobile robotic platform of LAAS-CNRS.

4.2.4 Localization

Only one localization sensor is used for this task, which is a kinect camera fixed on top of the base with a 3D printed support. To get the coordinates of the door and the handle we use a similar method than the one presented in Sec.3.4.3.

For the calibration of the kinect relatively to the arm we used a linear regression algorithm [Cashbaugh 2018]. The vision process is summarized in Appendix.D (Fig. D.1 and Fig. D.2).

4.2.5 Task decomposition

The objective is to open the door fully, with an opening angle of roughly 90 degrees. Depending on the type of door and handle, we have to adopt different strategies.

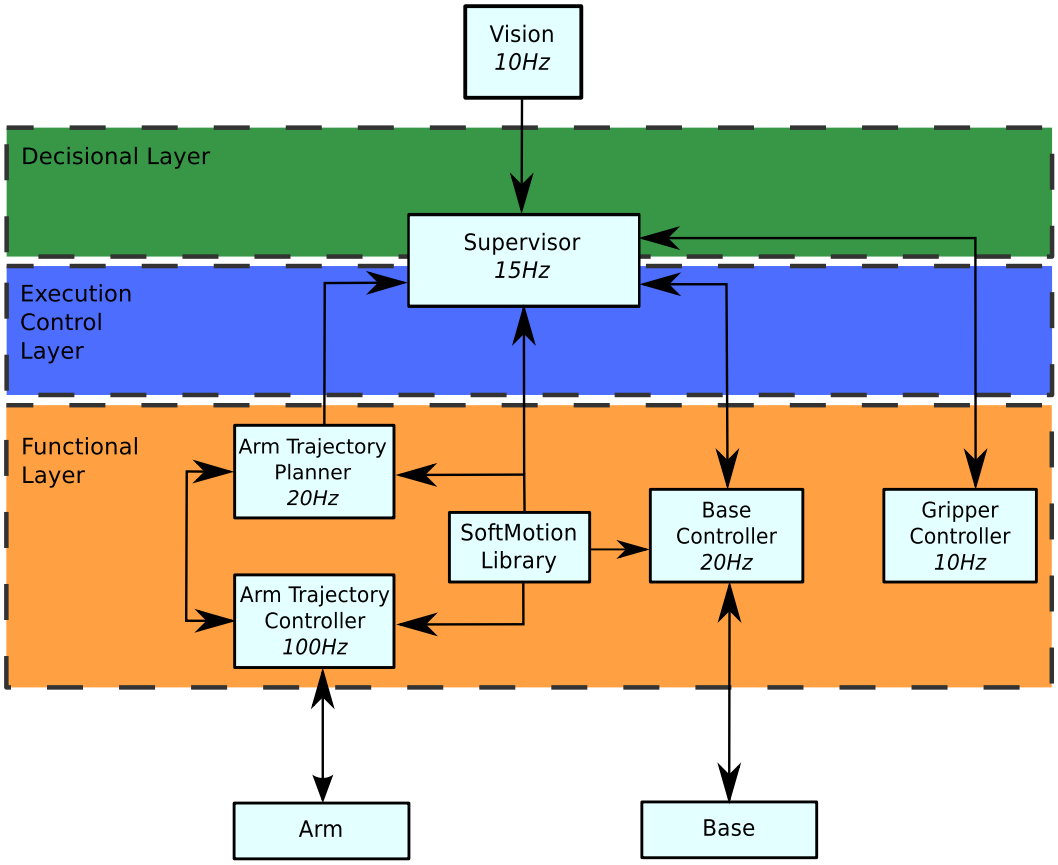


Figure 4.7: A simplified version of the Jido's software architecture.

The door can be pushed or pulled, and with a spring mechanism. In our case we are using the available door we have at LAAS (Fig. D.1a). This door needs to be pushed and does not have any spring mechanism. To accomplish this task, the arm alone cannot open the door fully as it doesn't possess the reach necessary to do so. Two main solutions are then available:

- The arm opens slightly the door and pushes it.
- The arm and the base realize a synchronized motion to open the door with the base moving and the arm still grasping the handle.

In our case we don't want to release the handle during the opening phase, thus we preferred the second solution. This task can be broken down as follow:

- Localization of the door.
- Position the base relatively to the door.
- The arm grasps the handle.

- The handle is pushed down.
- The door is slightly pushed.
- The handle is raised up.
- Full opening of the door with a synchronized motion of the base and arm.
- Move the base to free the way.

Videos of the realization of the whole task can be found at `https://github.com/kdesorme/Thesis_videos`. In the following we discuss the arm-base synchronization for the full opening of the door, which constitutes the most challenging aspect of this task.

4.2.6 Arm-Base synchronisation

The door's opening trajectory is the description of the evolution of $\mathcal{F}_{EE}$ in $\mathcal{F}_{Arm}$. The process to obtain this trajectory is described in Fig. 4.8. However the arm alone cannot open the door fully and we need an arm-base synchronized trajectory. Since the base crosses the door by moving forward on a straight line, we can split the trajectory by giving the x component to the base, while the arm ensures the rest. If the arm has $\mathcal{K}$ DOFS, and the base $\mathcal{N}$, the synchronization is accomplished simply by generating a phased-synchronized via-points trajectory of dimension $\mathcal{K}+\mathcal{N}$. This synchronization is achieved by the supervisor which then sends the respective trajectories to the arm and base. The supervisor controls their execution by ensuring that one module is not behind on its trajectory and by monitoring the forces. For example simple time-scale techniques can be used to keep the synchronization between the arm and base for the full duration of the opening. In our case these techniques were not needed, and the use of kinematics only were sufficient to accomplish this task. The arm was operated in joint impedance control mode, but we finally didn't use a lowered stiffness as it wasn't required. The execution was precise enough so that no excessive forces were exerted.

Nonetheless we found that using a stiffness trajectory control was essential. When we switch the control strategy of the arm, from a position control to an impedance control, the arm can deviate from its position. This might be explained by the time required to apply the default stiffness associated to the new control strategy. The arm is not as stiff for a moment and with the heavy gripper fixed at its extremity the arm can move or even fall in the worst case. The position's deviation can cause the *FRI* to return an interpolation error. The solution found for this problem was to set a stiffness following a smooth trajectory directly after the control's strategy switch. By setting the stiffness smoothly the *FRI* stopped to return interpolation errors.

We also have to ensure that the arm doesn't collide with the left door. The via-points were computed in the cartesian space, but we convert them to angular coordinates with the inverse kinematic model. Since the arm is redundant the

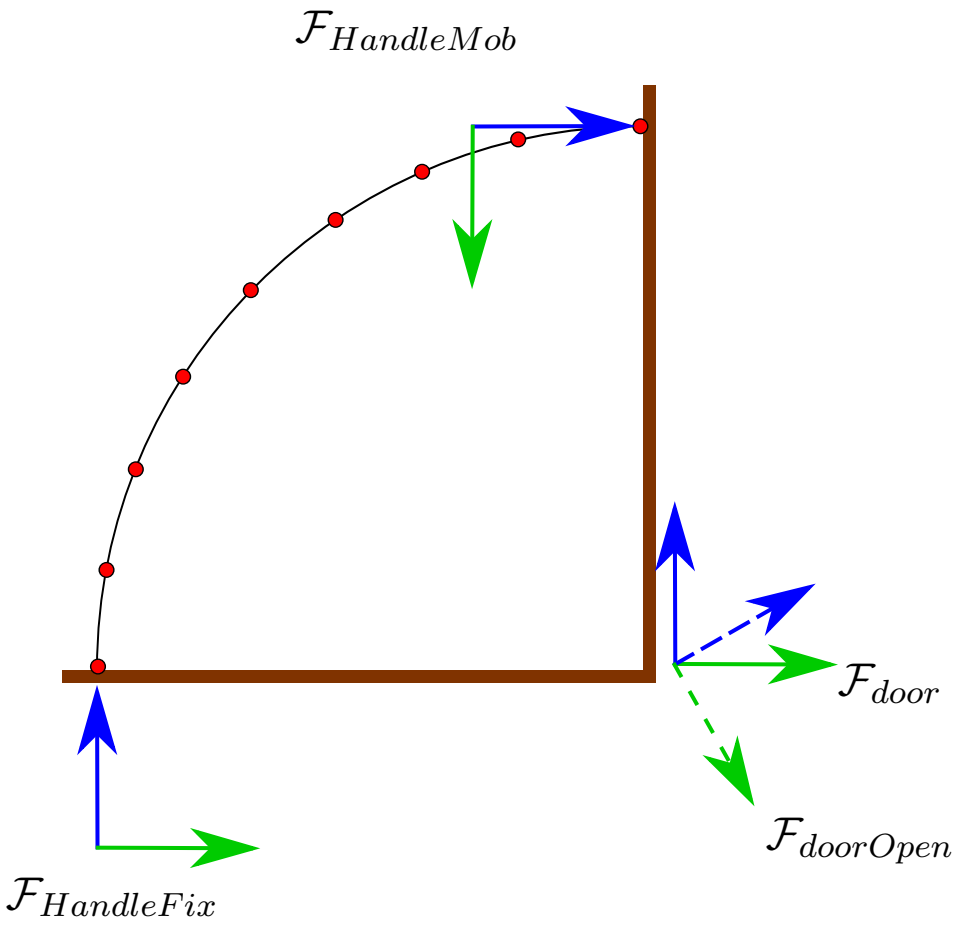


Figure 4.8: With the relations between the frames depicted in Fig. D.2, we express the coordinates of each frame relatively to others according to the opening angle of the door. By repeating the process with different opening angles we can obtain a series of $\mathcal{F}_{EE}$ expressed in $\mathcal{F}_{Arm}$. From there we can compute a via-points trajectory describing the door's opening.

inverse kinematic model allows to choose a desired pose. We have a total of eight poses that are determined by the configurations of the shoulder, elbow and wrist. Thus we choose the configuration that maximizes the distance between the left door with the wrist and elbow of the robotic arm. This solution is enough to ensure that the elbow doesn't collide, however it is not sufficient for the wrist. Since the Kuka LWR IV has a large wrist, if we open the door by grasping the handle with a 90 degrees angle, the wrist will scratch the left door during the opening. Hence the gripper must grasp the handle with an angle, in our case 20 degrees at least. Those difficulties could also have been overcome at the execution layer, however the supervisor depicted in Fig. 4.7 has been created for the sole purpose of this task and is quite rudimentary.

4.2.7 Conclusion

The autonomous opening of a door is a complex task for any kind of robot. It is even more complex when a synchronization between the different modules is desired. In this experiment we demonstrated that our trajectory models can be used effectively to accomplish such task. The realisation of the task with the sole use of kinematics validates the quality of our trajectory models.

The supervisor used is rudimentary, but a future direction of our work will be the integration of our trajectory models at the execution level, notably for the purpose of collision avoidance.

It was also seen in this experiment that our trajectory models can be used to command stiffness effectively. In the future we are considering to extend the use of our library to dynamics. An example of that is to implement a mixed controller, commanding the position and the stiffness according to velocity. We have seen the interest of such a controller in Sec. 1.2.3.2.

Chapter 5

Conclusion

Contributions

The work presented in this manuscript is focused on trajectory generation, mainly for Human-Robot Interactions.

In chapter 1 we presented a state of the art to understand the context of today's robotics and its future evolution. The collaboration between humans and robots will be a key component of tomorrow's robotics. After decades of replacing humans by robots in the industries, it is the opposite that is likely to happen in a near future. The motivation behind this change is to get the best of both world by combining the respective strengths of humans and robots. Robots are still expected to be efficient, however they will have to adapt to the human presence. This addition requires safe and flexible robots. Robots will have to be easily commanded and programmed by humans for a variety of tasks. But most importantly they must preserve both the physical and psychological safety of the humans.
We have chosen the use of smooth trajectories as a model of motions, because they fulfil all those criteria.

In chapter 2 we presented an OTG for the generation of smooth trajectories adapted to the context of Human-Robot Interactions. The algorithm based on sequences of segment of third degree polynomial functions is the first to our knowledge to satisfy simultaneously all the following criteria that are essentials for the generation of safe, efficient, adaptable and human-friendly motions:

- Real-time capable.
- General initial and end conditions for both velocity and acceleration.
- General asymmetric bounds on jerk, acceleration and velocity.
- Time optimal.

This algorithm also takes into account the non-linearity of the time-optimal curve regarding the length of the motion. The presence of discontinuities in the time-optimal curve impacts the construction of the time-optimal algorithm, and this problem was not addressed until now. The algorithm is also extended to the multi-dimensional case.

In chapter 3 we complete this algorithm so that it can cope with inadmissible initial configurations. This extension opens the way towards trajectory generation with non-constant motion constraints. This feature is essential in the field of physical Human-Robot Interactions to allow the robot to adapt its behaviour in real-time either to preserve the physical safety of humans or to provide the interaction with good ergonomics properties. However switching trajectory online in real-world applications is difficult. Since the robot is moving, switching to another trajectory requires that the trajectory is planned from the configuration of the system at the moment the new trajectory is ready to be executed. Or in most cases it is not feasible for two reasons: either because we don't know the current configuration of the system as we don't have acceleration and velocity measurements, or because the trajectory is outdated since not planned at the low-level controller. To answer these difficulties we built an architecture, controllers, as well as a strategy adapted to Reactive Trajectory Control on real-world applications. We validated this solution in adequacy to the previous works with experimental evaluations.

Thanks to the previous contributions it is possible to adapt a robot's behavior during interactions by adapting the kinematics constraints. If we know how to adapt the robot's velocity to preserve the physical integrity of humans based on the numerous works on the subject, we know little on the relation between kinematics and ergonomics.

In chapter 4 we investigated the role of kinematics in the definition of ergonomics properties of motions. This was done by conducting a user study that aimed to be a preliminary work. We got some interesting results from which futures studies can be based on.

Perspectives

Although it seems that the trajectory generation algorithm we presented is complete there is room for future improvements. For the multi-dimensional case we didn't considered the full complexity presented by the time-synchronization problem. We have developed tools that can be used to take into account the dynamic of the task, but we didn't explored this possibility yet. However we have proved the efficiency of our model for motions, as well as its simplicity. These tools we developed, can now be integrated at any level of an architecture for autonomous robots. Our trajectory model can be used as a support of communications to represent motions to consider both kinematics and the dynamic of the task.

Time-synchronisation for general conditions

In section 2.3.3 we discussed the difficulties in the time-synchronization of non rest to rest motions. These difficulties are present for motions with general conditions, but other factors such as the length of the motion have to be considered. To

implement a complete algorithm each of these cases should be considered in order to develop appropriate solutions.

The solution we proposed was to use 3 segments trajectories (Sec.3.3.5). This solution presents some drawbacks. Depending on the variant employed, the jerk is not bounded on some segments. Moreover we don't consider the case where no synchronization is possible on the time of the slowest axis. Even if in most situations our algorithm returns a valid solution, there is room for improvements. A possibility would be to use more than one 3 segments trajectory. We could also consider to use a 3 segments trajectory to join the rest configuration and a general trajectory to join the final condition from there. Then we smooth the junction between these two trajectories using our via-points algorithm. With this solution the trajectory could be time-synchronized at the beginning, and phase synchronized at one point.

Fourth degree polynomial trajectories

In specific situations, fourth degree polynomial trajectories can be needed to take into account the snap. We don't know if it is needed for HRI. Yet the extension of our trajectory generation algorithm to consider the snap would be challenging. Moreover, the two dimensional phase diagram which was an essential tool to the comprehension of trajectories behaviour would be obsolete. A three dimensional phase diagram would probably be impossible to comprehend.

Controlling the dynamic of the task

During this thesis mostly kinematics were considered. Nonetheless the tools we developed can be used to consider dynamics as well. Modifying the stiffness of an impedance or admittance controller is delicate because it can generate large variation of position or forces. In paragraph 4.2.6 we have seen how a trajectory controller could be use to apply smoothly a stiffness to a robotic arm.

Notably we consider the development of a mixed controller able to vary the compliance of the axes as they move. This is a feature that humans naturally possess. Humans can vary the compliance of their arm for different tasks, and even during different phases of a task. This variable compliance can be very useful in the making of safe and fast motions. The human arm is typically controlled to move slowly when it is stiff, and to be compliant while moving fast. In this control approach stiffness can be considered similarly as a kinematic bound, by maximizing it under a safe maximum stiffness bound. This bound can be determined by the distance to a human. This approach is similar to the performance-based approach we presented for our general trajectory generation algorithm.

Ergonomics properties of motion

The user study presented in section 4.1 was a preliminary work to study the relation between kinematics and ergonomics of movements. This work can be used to

conduct another study based on the conclusions we presented. Moreover this experiment contributed to improve our tools to evaluate Human-Robot Interactions from which future studies can benefit. Results from these studies would also ease the integration of our work in the decision and execution layer by offering better descriptions of motions.

A three layer architecture

In a near future we will be working on the integration of our work at the decision and execution control layer. As we have seen trajectories constitute an excellent mean to describe motions. Their use as a support of communication directly from a task planner could be beneficial for the whole system. For example it is better to use trajectories as an input of a collision avoidance module rather than paths. For example it can be used to manage the position error at via-points, or to take into account moving objects. The integration of the softMotion library into a motion planner developed at LAAS is a future work perspective.

Appendix A

Analytic expression of the parametric curve $\mathcal{C}(x_f(t_n), t_f(t_n))$

We consider here the particular case where the trajectory is defined by:

- Three non null jerk segments associated to parabolas in the phase diagram.
- The first and the last segments are associated to $J_{min} < 0$..
- The third segment begin with a positive acceleration a_2.

The first trajectory segment is entirely defined by the initial conditions a_i and v_i and the duration t_1 using the equations (2.3).

The parabola associated to the second segment cross the abscissa axis at v_i' and the one associated to the third segment at v_f' with:

$$v_i' = v_i - \frac{{a_i}^2}{2J_{max}} \qquad\qquad v_f' = v_f - \frac{{a_f}^2}{2J_{min}}$$

And the acceleration at the intersection point of the two parabolas is a_2 defined by:

$$a_2 = \sqrt{\frac{2(v_i' - v_f') \times J_{max} \times J_{min}}{J_{max} - J_{min}}}$$

We choose the positive solution as defined in the hypothesis and compute the durations t_2 and t_3 of the two last segments.

$$t_2 = \frac{(a_2 - a_1)}{J_{max}} \qquad\qquad t_3 = \frac{(a_f - a_2)}{J_{min}}$$

From the equations (2.4) and (2.5) and using an algebraic calculator, the values of x_f and $t_f = t_1 + t_2 + t_3$ can be easily computed. Then the zeros of the derivative of the x_f function with respect to t_1 would define the points of discontinuity of the optimal curve C_{opt}. Unfortunately, we cannot manage to obtain this points in an usable form.

Appendix B

Derivative of the cubic polynomial functions

The derivatives of the functions x_f, v_f and a_f relatively to the three times t_1, t_2 and t_3 can be grouped in a matrix. The newton method uses the inverse of this matrix to compute the times increments that, at the first oder, compensate the errors of the functions.

$$\begin{pmatrix} \frac{\partial x_f}{\partial t_1} & \frac{\partial x_f}{\partial t_2} & \frac{\partial x_f}{\partial t_3} \\ \frac{\partial v_f}{\partial t_1} & \frac{\partial v_f}{\partial t_2} & \frac{\partial v_f}{\partial t_3} \\ \frac{\partial a_f}{\partial t_1} & \frac{\partial a_f}{\partial t_2} & \frac{\partial a_f}{\partial t_3} \end{pmatrix} = \begin{pmatrix} lx_1 & lv_1 & la_1 \\ lx_2 & lv_2 & la_2 \\ lx_3 & lv_3 & la_3 \end{pmatrix}^{-1}$$

In the (JJJ) case defined by the equations (2.3), (2.4) (2.5), using an algebraic calculator we obtain:

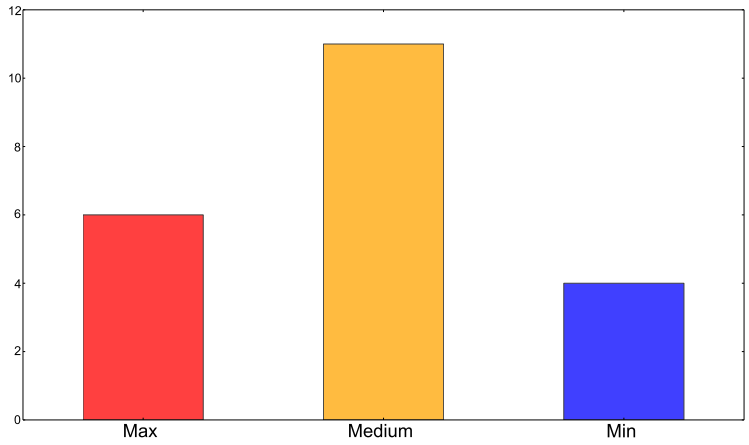
$$
\begin{aligned}
k_1 &= J_a \times t_1 \\
k_2 &= J_b \times t_2 \\
k_3 &= J_a \times t_3 \\
k_4 &= a_i + k_3 + k_2 + k_1 \\
k_5 &= a_i + J_b \times (t_3 + t_2) + k_1 \\
k_6 &= J_a \times k_5 - J_b \times k_4 \\
k_7 &= {t_1}^2 \\
k_8 &= {t_2}^2 \\
k_9 &= {t_3}^2 \\
k_{10} &= (J_a \times (k_9 + k_8 + k_7))/2 \\
k_{11} &= t_1 \times a_i \\
k_{12} &= t_2 \times (a_i + k_1) \\
k_{13} &= t_3 \times (a_i + J_a \times (t_2 + t_1)) \\
k_{14} &= v_i + k_{13} + k_{12} + k_{11} + k_{10} \\
k_{15} &= a_i + J_a \times (t_3 + t_2 + t_1) \\
k_{16} &= J_b \times k_{15} - J_a \times k_5 \\
k_{17} &= J_a \times k_7 \\
k_{18} &= J_a \times k_9 + J_b \times k_8 + k_{17} \\
k_{19} &= t_3 \times (a_i + k_2 + k_1) \\
k_{20} &= v_i + k_{19} + k_{12} + k_{11} + k_{18}/2 \\
k_{21} &= (-k_{15}) + a_i + k_3 + k_2 + k_1 \\
k_{22} &= v_i + k_{19} + k_{12} + k_{11} + (J_b \times (k_9 + k_8) + k_{17})/2 \\
k_{23} &= 1/(J_a \times k_{21} \times k_{22} + k_{16} \times k_{20} + k_6 \times k_{14}) \\
lx_1 &= k_6 \times k_{23} \\
lv_1 &= (J_b \times k_{20} - J_a \times k_{22}) \times k_{23} \\
la_1 &= (k_4 \times k_{22} - k_5 \times k_{20}) \times k_{23}] \\
lx_2 &= J_a \times k_{21} \times k_{23} \\
lv_2 &= J_a \times (-(k_{19} + k_{12} + k_{11} - (-k_{18}/2)) \\
&\quad + k_{13} + k_{12} + k_{11} + k_{10}) \times k_{23} \\
la_2 &= (k_{15} \times k_{20} - k_4 \times k_{14}) \times k_{23}] \\
lx_3 &= k_{16} \times k_{23} \\
lv_3 &= (J_a \times k_{22} - J_b \times k_{14}) \times k_{23} \\
la_3 &= (k_5 \times k_{14} - k_{15} \times k_{22}) \times k_{23}]))
\end{aligned}
$$

The computation is done similarly in the case of (AJA), (AJJ) and (JJA) se-

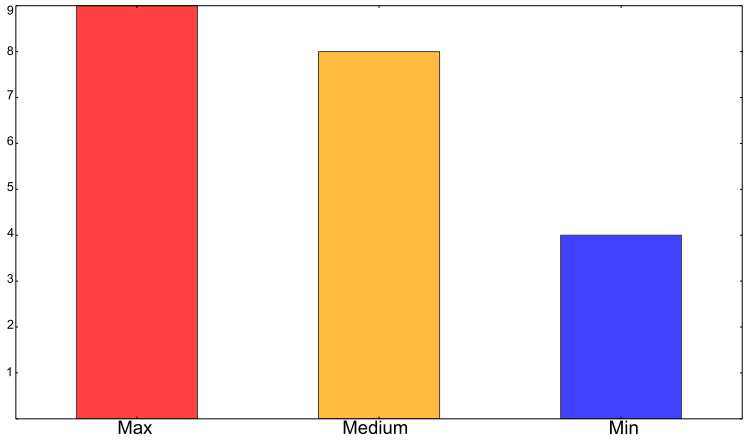
quences.

Appendix C

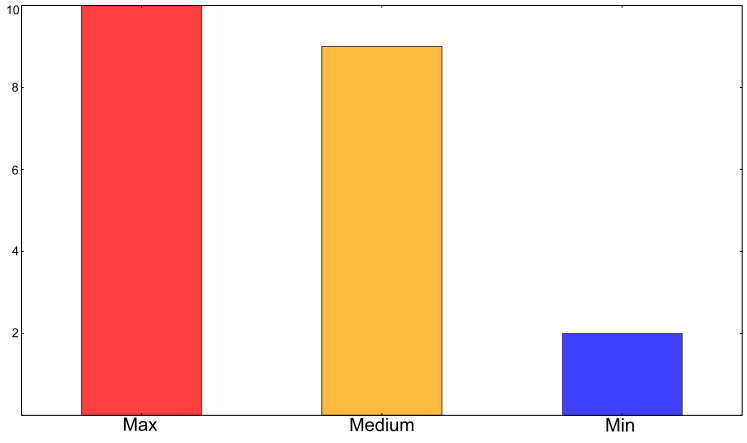
User study : Ergonomic Properties of Motions



(a) Jerk

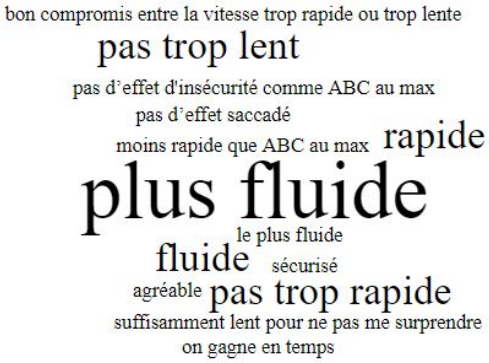


(b) Acceleration



(c) Velocity

Figure C.1: Histograms depicting the values chosen by the subjects for their favourite motion.



(a) Choice of minimum jerk.

(b) Choice of minimum acceleration.

(c) Choice of minimum velocity.

Figure C.2: Motivations for the choice of one of the imposed tests.

Figure C.3: Reasons mentioned for the choice of the favourite motion during E.3.

Appendix D

Autonomous door-opening

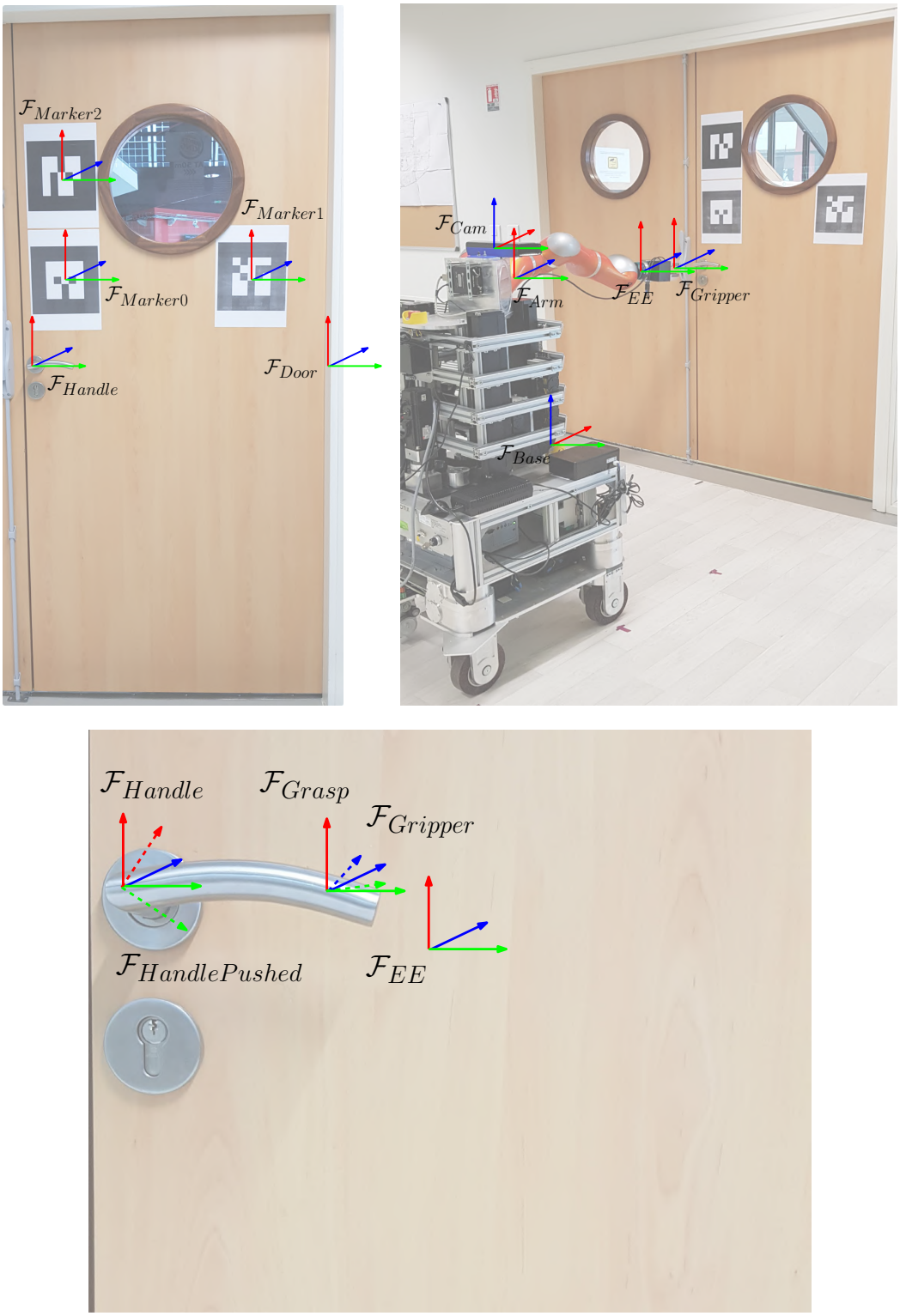


Figure D.1: Illustration of the different frames for the localization process.

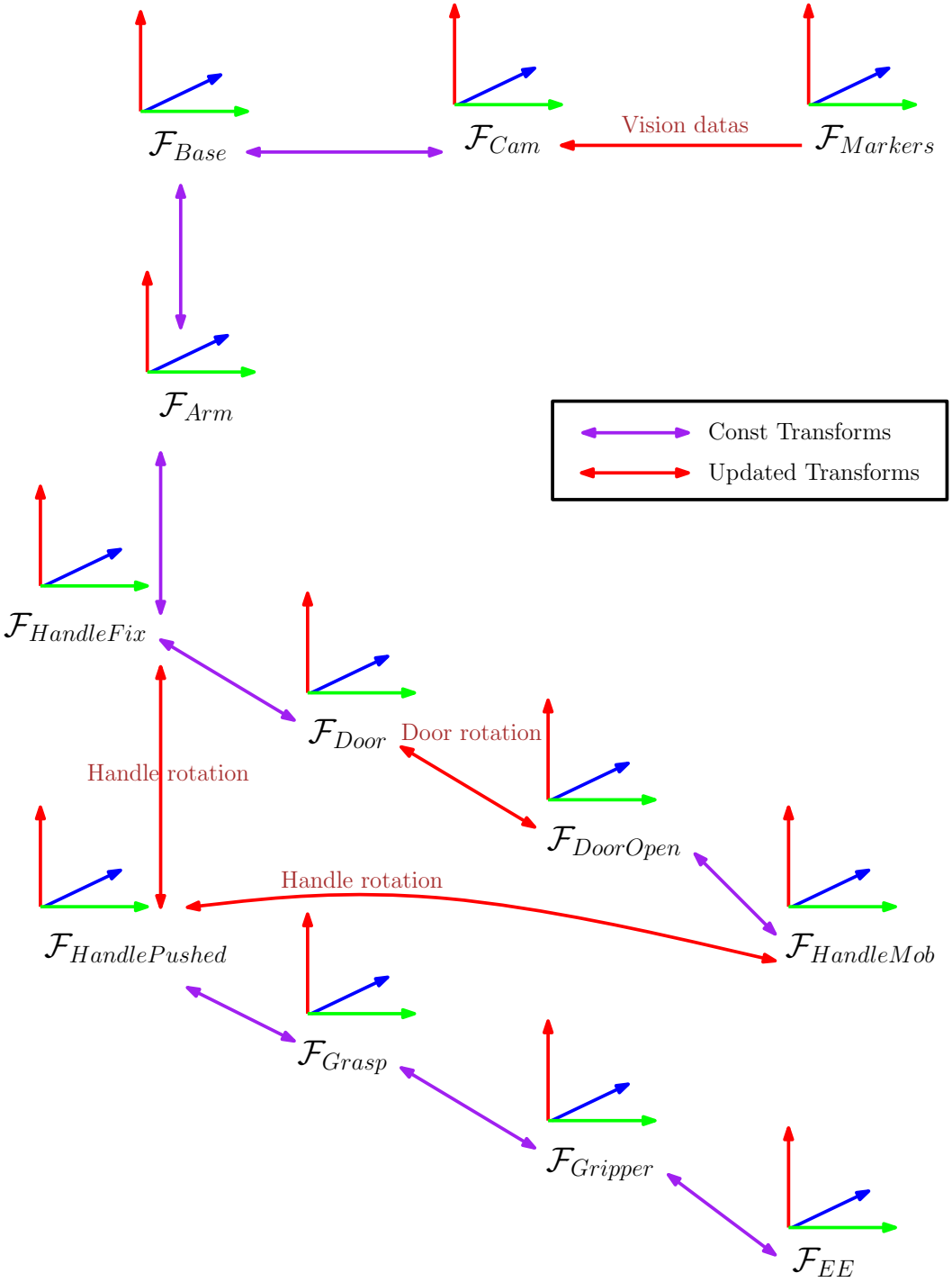


Figure D.2: This figure summarizes the localization process.

Bibliography

[Abend 1982] W. Abend, E. Bizzi and P. Morasso. *HUMAN ARM TRAJECTORY FORMATION.* Brain, vol. 105, no. 2, pages 331–348, 1982. (Cited in pages 27 and 28.)

[Alami 1998] R. Alami, R. Chatila, S. Fleury, M. Ghallab and F. Ingrand. *An Architecture for Autonomy.* The International Journal of Robotics Research, vol. 17, no. 4, pages 315–337, April 1998. (Cited in page 29.)

[Amirabdollahian 2002] F. Amirabdollahian, R. Loureiro and W. Harwin. *Minimum jerk trajectory control for rehabilitation and haptic applications.* volume 4, pages 3380–3385, 2002. (Cited in page 36.)

[Arai 2010] T. Arai, R. Kato and M. Fujita. *Assessment of operator stress induced by robot collaboration in assembly.* CIRP Annals - Manufacturing Technology, vol. 59, no. 1, pages 5–8, 2010. (Cited in page 26.)

[Bartneck 2009] Christoph Bartneck, Dana Kulić, Elizabeth Croft and Susana Zoghbi. *Measurement Instruments for the Anthropomorphism, Animacy, Likeability, Perceived Intelligence, and Perceived Safety of Robots.* International Journal of Social Robotics, vol. 1, no. 1, pages 71–81, January 2009. (Cited in page 113.)

[Beggs 1972] W. D. A. Beggs and C. I. Howarth. *The movement of the hand towards a target.* The Quarterly Journal of Experimental Psychology, vol. 24, no. 4, pages 448–453, November 1972. (Cited in page 28.)

[Biagiotti 2008] Luigi Biagiotti. Trajectory Planning for Automatic Machines and Robots. Springer, 2008. (Cited in pages 33, 34, 65, and 67.)

[Bianco 2011] C. Guarino Lo Bianco and O. Gerelli. *Online Trajectory Scaling for Manipulators Subject to High-Order Kinematic and Dynamic Constraints.* IEEE Transactions on Robotics, vol. 27, no. 6, pages 1144–1152, December 2011. (Cited in page 43.)

[Bicchi 2008] Antonio Bicchi, Michael A. Peshkin and J. Edward Colgate. *Safety for Physical Human–Robot Interaction.* In Bruno Siciliano and Oussama Khatib, editors, Springer Handbook of Robotics, pages 1335–1348. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. (Cited in pages 18 and 25.)

[Bischoff] Rainer Bischoff, Johannes Kurth, Günter Schreiber, Ralf Koeppe and Alin Albu-Schäffer. *The KUKA-DLR Lightweight Robot Arm - a New Reference Platform for Robotics Research and Manufacturing.* page 8. (Cited in pages 18 and 25.)

[Bizzi 1984] E. Bizzi, N. Accornero, W. Chapple and N. Hogan. *Posture control and trajectory formation during arm movement.* The Journal of Neuroscience:

The Official Journal of the Society for Neuroscience, vol. 4, no. 11, pages 2738–2744, November 1984. (Cited in pages 27 and 28.)

[Blaha 2014] Lukas Blaha. *Multi-axis Time Synchronization for Uncoordinated Motion Planning with Hard Constraints.* IFAC Proceedings Volumes, vol. 47, no. 3, pages 3845–3850, 2014. (Cited in pages 65, 67, and 68.)

[Breazeal 2003] Cynthia Breazeal. *Toward sociable robots.* Robotics and Autonomous Systems, vol. 42, no. 3-4, pages 167–175, March 2003. (Cited in page 21.)

[Breazeal 2016] Cynthia Breazeal, Kerstin Dautenhahn and Takayuki Kanda. *Social Robotics.* In Bruno Siciliano and Oussama Khatib, editors, Springer Handbook of Robotics, pages 1935–1972. Springer International Publishing, Cham, 2016. (Cited in page 21.)

[Broquere 2008] Xavier Broquere, Daniel Sidobre and Ignacio Herrera-Aguilar. *Soft motion trajectory planner for service manipulator robot.* In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2808–2813. IEEE, 2008. (Cited in pages 37, 42, 43, and 44.)

[Broquere 2010] Xavier Broquere, Daniel Sidobre and Khoi Nguyen. *From motion planning to trajectory control with bounded jerk for service manipulator robots.* In Robotics and Automation (ICRA), pages 4505–4510. IEEE, 2010. (Cited in pages 33, 35, 37, 43, and 93.)

[Broquère 2011] Xavier Broquère. *Planification de trajectoire pour la manipulation d'objets et l'interaction homme-robot.* PhD thesis, 2011. (Cited in pages 65, 67, and 68.)

[Brynjolfsson 2014] Erik Brynjolfsson and Andrew McAfee. The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies. W. W. Norton & Company, 1st edition, 2014. (Cited in page 13.)

[Butler 2001] John Travis Butler and Arvin Agah. *Psychological effects of behavior patterns of a mobile personal robot.* Autonomous Robots, vol. 10, no. 2, pages 185–202, 2001. (Cited in page 25.)

[Cashbaugh 2018] Jasmine Cashbaugh and Christopher Kitts. *Automatic Calculation of a Transformation Matrix Between Two Frames.* IEEE Access, vol. 6, pages 9614–9622, 2018. (Cited in page 120.)

[Colgate 1996] J. Edward Colgate, J. Edward, Michael A. Peshkin and Witaya Wannasuphoprasit. Cobots: Robots For Collaboration With Human Operators. 1996. (Cited in page 19.)

[Costantinescu 2000] D. Costantinescu and E. A. Croft. *Smooth and time-optimal trajectory planning for industrial manipulators along specified paths.* Journal of robotic systems, vol. 17, no. 5, pages 233–249, 2000. (Cited in page 43.)

[Craig 1986] John J. Craig. Introduction to Robotics: Mechanics and Control. Pearson/Prentice Hall, 1986. (Cited in page 35.)

[Danna 1999] Karen Danna and Ricky W Griffin. *Health and Well-Being in the Workplace: A Review and Synthesis of the Literature.* Journal of Management, vol. 25, no. 3, page 28, 1999. (Cited in page 25.)

[Dombre 2007] Etienne Dombre and Wisama Khalil. Robot Manipulators: Modeling, Performance Analysis and Control. 2007. (Cited in page 33.)

[Dragan 2013a] Anca Dragan and Siddhartha Srinivasa. *Generating legible motion.* Berlin, Germany, 2013. (Cited in page 26.)

[Dragan 2013b] Anca D. Dragan, Kenton CT Lee and Siddhartha S. Srinivasa. *Legibility and predictability of robot motion.* In 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 301–308. IEEE, 2013. (Cited in page 26.)

[Ezair 2014] B. Ezair, T. Tassa and Z. Shiller. *Planning high order trajectories with general initial and final conditions and asymmetric bounds.* The International Journal of Robotics Research, vol. 33, no. 6, pages 898–916, May 2014. (Cited in pages 38 and 42.)

[Flanagan 1990] J. Randall Flanagan and David J. Ostry. *Trajectories of human multi-joint arm movements: Evidence of joint level planning.* In Experimental Robotics I, pages 594–613. Springer, Berlin, Heidelberg, 1990. (Cited in page 28.)

[Flash 1985] T. Flash and N. Hogan. *The coordination of arm movements: an experimentally confirmed mathematical model.* The Journal of Neuroscience: The Official Journal of the Society for Neuroscience, vol. 5, no. 7, pages 1688–1703, July 1985. (Cited in pages 27 and 28.)

[Flocke 2015] N. Flocke. *Algorithm 954: An Accurate and Efficient Cubic and Quartic Equation Solver for Physical Applications.* ACM Transactions on Mathematical Software, vol. 41, no. 4, pages 1–24, October 2015. (Cited in page 62.)

[Fong 2003] Terrence Fong, Illah Nourbakhsh and Kerstin Dautenhahn. *A survey of socially interactive robots.* Robotics and Autonomous Systems, vol. 42, no. 3–4, pages 143–166, March 2003. (Cited in page 21.)

[Frisoli 2013] A. Frisoli, C. Loconsole, R. Bartalucci and M. Bergamasco. *A new bounded jerk on-line trajectory planning for mimicking human movements in robot-aided neurorehabilitation.* Robotics and Autonomous Systems, vol. 61, no. 4, pages 404–415, April 2013. (Cited in pages 65 and 67.)

[Fujita 2010] Marina Fujita, Ryu Kato and Arai Tamio. *Assessment of operators' mental strain induced by hand-over motion of industrial robot manipulator.* pages 361–366. IEEE, 2010. (Cited in page 26.)

[Ganster 2013] Daniel C Ganster and Christopher C Rosen. *Work Stress and Employee Health: A Multidisciplinary Review.* JOURNAL OF MANAGEMENT, page 39, 2013. (Cited in page 25.)

[Gasparetto 2008] Alessandro Gasparetto and Vanni Zanotto. *A technique for time-jerk optimal planning of robot trajectories.* Robotics and Computer-Integrated Manufacturing, vol. 24, no. 3, pages 415–426, June 2008. (Cited in page 36.)

[Gerelli 2009] O. Gerelli and C. G. Lo Bianco. *Nonlinear Variable Structure Filter for the Online Trajectory Scaling.* IEEE Transactions on Industrial Electronics, vol. 56, no. 10, pages 3921–3930, October 2009. (Cited in page 43.)

[Haddadin 2008] Sami Haddadin, Alin Albu-Schäffer, Alessandro De Luca and Gerd Hirzinger. *Collision detection and reaction: A contribution to safe physical Human-Robot Interaction.* 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3356–3363, 2008. (Cited in page 25.)

[Haddadin 2009] Sami Haddadin, Alin Albu-Schäffer and Gerd Hirzinger. *Requirements for Safe Robots: Measurements, Analysis and New Insights.* The International Journal of Robotics Research, vol. 28, no. 11-12, pages 1507–1527, November 2009. (Cited in page 24.)

[Haddadin 2012] Sami Haddadin, Simon Haddadin, Augusto Khoury, Tim Rokahr, Sven Parusel, Rainer Burgkart, Antonio Bicchi and Alin Albu-Schäffer. *On making robots understand safety: Embedding injury knowledge into control.* The International Journal of Robotics Research, vol. 31, no. 13, pages 1578–1602, November 2012. (Cited in page 24.)

[Haschke 2008] Robert Haschke, Erik Weitnauer and Helge Ritter. *On-line planning of time-optimal, jerk-limited trajectories.* In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3248–3253. IEEE, 2008. (Cited in pages 37, 42, and 44.)

[Hassard 2018] Juliet Hassard, Kevin R. H. Teoh, Gintare Visockaite, Philip Dewe and Tom Cox. *The cost of work-related stress to society: A systematic review.* Journal of Occupational Health Psychology, vol. 23, no. 1, pages 1–17, January 2018. (Cited in page 25.)

[Herrera-Aguilar 2006] Ignacio Herrera-Aguilar and Daniel Sidobre. *Soft motion trajectory planning and control for service manipulator robot.* In Workshop on Physical Human-Robot Interaction in Anthropic Domains at IROS, pages 13–22, 2006. (Cited in pages 37 and 43.)

[Hoffman 2007] Guy Hoffman and Cynthia Breazeal. *Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team.* In Proceedings of the ACM/IEEE international conference on Human-robot interaction, pages 1–8. ACM, 2007. (Cited in page 26.)

[Hogan 1982] N. Hogan. *Control and Coordination of Voluntary Arm Movements.* pages 522–528, June 1982. (Cited in pages 28 and 36.)

[Hogan 1984] N. Hogan. *An organizing principle for a class of voluntary movements.* The Journal of Neuroscience: The Official Journal of the Society for Neuroscience, vol. 4, no. 11, pages 2745–2754, 1984. (Cited in pages 28 and 35.)

[Huber 2009] Markus Huber, Helmuth Radrich, Cornelia Wendt, Markus Rickert, Alois Knoll, Thomas Brandt and Stefan Glasauer. *Evaluation of a novel biologically inspired trajectory generator in human-robot interaction.* In RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication, pages 639–644, Toyama, Japan, September 2009. IEEE. (Cited in page 29.)

[ISO-10218-1 2011] ISO-10218-1. *ISO-10218-1*, 2011. (Cited in page 22.)

[ISO-10218-2 2011] ISO-10218-2. *ISO-10218-2*, 2011. (Cited in page 22.)

[ISO-13482 2014] ISO-13482. *ISO-13482*, 2014. (Cited in pages 22 and 24.)

[ISO-15066 2016] ISO-15066. *ISO-15066*, 2016. (Cited in pages 22 and 23.)

[Kavraki 2008] Lydia E. Kavraki and Steven M. LaValle. *Motion Planning.* In Bruno Siciliano Prof and Oussama Khatib Prof, editors, Springer Handbook of Robotics, pages 109–131. Springer Berlin Heidelberg, 2008. (Cited in page 32.)

[Khalil 2004] W. Khalil and E. Dombre. Modeling, Identification and Control of Robots. Butterworth-Heinemann, July 2004. (Cited in page 33.)

[Kiss 2007] B. Kiss and E. Szadeczky-Kardoss. *On-line time-scaling control of a kinematic car with one input.* In 2007 Mediterranean Conference on Control & Automation, pages 1–6, Athens, Greece, June 2007. IEEE. (Cited in page 86.)

[Kroger 2006] Torsten Kroger, Adam Tomiczek and Friedrich M. Wahl. *Towards on-line trajectory computation.* In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 736–741. IEEE, 2006. (Cited in pages 37 and 42.)

[Kroger 2010] T. Kroger and F.M. Wahl. *Online Trajectory Generation: Basic Concepts for Instantaneous Reactions to Unforeseen Events.* IEEE Transactions on Robotics, vol. 26, no. 1, pages 94–111, February 2010. (Cited in pages 35, 37, 42, 65, 68, 84, 85, 86, and 87.)

[Kroger 2011] Torsten Kroger. *Online Trajectory Generation: Straight-Line Trajectories*. IEEE Transactions on Robotics, vol. 27, no. 5, pages 1010–1016, October 2011. (Cited in pages 65 and 67.)

[Kroger 2012] Torsten Kroger. *On-line trajectory generation: Nonconstant motion constraints*. In Robotics and Automation (ICRA), pages 2048–2054. IEEE, 2012. (Cited in pages 37 and 87.)

[Krüger 2009] J. Krüger, T.K. Lien and A. Verl. *Cooperation of human and machines in assembly lines*. CIRP Annals, vol. 58, no. 2, pages 628–646, 2009. (Cited in page 19.)

[Kulić 2005] Dana Kulić and Elizabeth A. Croft. *Safe planning for human-robot interaction*. Journal of Robotic Systems, vol. 22, no. 7, pages 383–396, July 2005. (Cited in page 25.)

[Kulić 2006] Dana Kulić and Elizabeth A. Croft. *Real-time safety for human–robot interaction*. Robotics and Autonomous Systems, vol. 54, no. 1, pages 1–12, January 2006. (Cited in page 25.)

[Kulić 2007] Dana Kulić and Elizabeth Croft. *Pre-collision safety strategies for human-robot interaction*. Autonomous Robots, vol. 22, no. 2, pages 149–164, January 2007. (Cited in page 25.)

[Kyriakopoulos 1988] K. J. Kyriakopoulos and G. N. Saridis. *Minimum jerk path generation*. pages 364–369 vol.1, April 1988. (Cited in pages 35 and 36.)

[Lambrechts 2005] Paul Lambrechts, Matthijs Boerlage and Maarten Steinbuch. *Trajectory planning and feedforward design for electromechanical motion systems*. Control Engineering Practice, vol. 13, no. 2, pages 145–157, February 2005. (Cited in page 42.)

[Lasota 2015] Przemyslaw A. Lasota and Julie A. Shah. *Analyzing the Effects of Human-Aware Motion Planning on Close-Proximity Human–Robot Collaboration*. Human Factors: The Journal of the Human Factors and Ergonomics Society, vol. 57, no. 1, pages 21–33, February 2015. (Cited in page 26.)

[Lasota 2017] Przemyslaw A. Lasota, Terrence Fong and Julie A. Shah. *A Survey of Methods for Safe Human-Robot Interaction*. Foundations and Trends in Robotics, vol. 5, no. 3, pages 261–349, 2017. (Cited in pages 25 and 26.)

[Latombe 1990] Jean-Claude Latombe. Robot Motion Planning. Springer, 1990. (Cited in page 32.)

[LaValle 2006] Steven M. LaValle. Planning Algorithms. Cambridge University Press, New York, NY, USA, 2006. (Cited in page 32.)

[Liu 2002] Steven Liu. *An on-line reference-trajectory generator for smooth motion of impulse-controlled industrial manipulators.* In 7th International Workshop on Advanced Motion Control., pages 365–370. IEEE, 2002. (Cited in pages 37, 42, and 44.)

[Macfarlane 2003] S. Macfarlane and E.A. Croft. *Jerk-bounded manipulator trajectory planning: design for real-time applications.* IEEE Transactions on Robotics and Automation, vol. 19, no. 1, pages 42–52, February 2003. (Cited in pages 35, 37, 38, 42, and 44.)

[Michalos 2014] George Michalos, Sotiris Makris, Jason Spiliotopoulos, Ioannis Misios, Panagiota Tsarouchi and George Chryssolouris. *ROBO-PARTNER: Seamless Human-Robot Cooperation for Intelligent, Flexible and Safe Operations in the Assembly Factories of the Future.* Procedia CIRP, vol. 23, pages 71–76, 2014. (Cited in page 19.)

[Morasso 1981] P. Morasso. *Spatial control of arm movements.* Experimental Brain Research, vol. 42, no. 2, April 1981. (Cited in pages 27 and 28.)

[Nagasaki 1989] H. Nagasaki. *Asymmetric velocity and acceleration profiles of human arm movements.* Experimental brain research, vol. 74, no. 2, pages 319–326, 1989. (Cited in page 28.)

[Nguyen 2008] Kim Doang Nguyen, Teck-Chew Ng and I.-Ming Chen. *On algorithms for planning S-curve motion profiles.* International Journal of Advanced Robotic Systems, vol. 5, no. 1, pages 99–106, 2008. (Cited in pages 38 and 42.)

[Ostry 1987] David J. Ostry, James D. Cooke and Kevin G. Munhall. *Velocity curves of human arm and speech movements.* Experimental Brain Research, vol. 68, no. 1, pages 37–46, 1987. (Cited in page 28.)

[Piazzi 1997] A. Piazzi and A. Visioli. *An interval algorithm for minimum-jerk trajectory planning of robot manipulators.* volume 2, pages 1924–1927, December 1997. (Cited in page 36.)

[Piazzi 2000] Aurelio Piazzi and Antonio Visioli. *Global minimum-jerk trajectory planning of robot manipulators.* IEEE transactions on industrial electronics, vol. 47, no. 1, pages 140–149, 2000. (Cited in page 36.)

[Puglisi] L J Puglisi, R J Saltaren and C E Garcia Cena. *On the Velocity and Acceleration Estimation from Discrete Time-Position Sensors.* Control Engineering and Applied Informatics, page 11. (Cited in page 90.)

[Restrepo 2018] Susana Sanchez Restrepo. *INTUITIVE, ITERATIVE AND ASSISTED VIRTUAL GUIDES PROGRAMMING FOR HUMAN-ROBOT COMANIPULATION.* PhD thesis, February 2018. (Cited in page 21.)

[Rivera-Guillen 2010] J. R. Rivera-Guillen, R. J. Romero-Troncoso, A. Osornio-Rios, A. Garcia-Perez and I. Torres-Pacheco. *Extending tool-life through jerk-limited motion dynamics in machining processes: An experimental study.* JSIR Vol.69(12), pages 919–925, December 2010. (Cited in page 35.)

[Rosch 2001] Paul J. Rosch. *ISMA-USA newsletter.* The International Stress Management Association, page 14, 2001. (Cited in page 25.)

[Russakovsky 2015] Olga Russakovsky, Li-Jia Li and Li Fei-Fei. *Best of both worlds: Human-machine collaboration for object annotation.* In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2121–2131, Boston, MA, USA, June 2015. IEEE. (Cited in page 19.)

[Schwab 2016] Klaus Schwab. *The Fourth Industrial Revolution.* January 2016. (Cited in page 13.)

[Schwarze 1990] Jochen Schwarze. *Graphics Gems.* pages 404–407. Academic Press Professional, Inc., San Diego, CA, USA, 1990. (Cited in page 62.)

[Shaowei 2012] Wang Shaowei and Wan Shanming. *Velocity and acceleration computations by single-dimensional Kalman filter with adaptive noise variance.* Przeglad Elektrotechniczny, pages 283–287, 2012. (Cited in page 90.)

[Shin 2015] Sung Yul Shin and ChangHwan Kim. *Human-Like Motion Generation and Control for Humanoid's Dual Arm Object Manipulation.* IEEE Transactions on Industrial Electronics, vol. 62, no. 4, pages 2265–2276, April 2015. (Cited in page 28.)

[Siciliano 2016] Bruno Siciliano and Oussama Khatib. Springer Handbook of Robotics. Springer Handbooks. Springer International Publishing, 2 edition, 2016. (Cited in page 32.)

[Sidobre 2012] Daniel Sidobre and Wuwei He. *Online task space trajectory generation.* In IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2012. (Cited in pages 34, 37, and 43.)

[Sidobre 2019] Daniel Sidobre and Kevin Desormeaux. *Smooth Cubic Polynomial Trajectories for Human-Robot Interactions.* Journal of Intelligent & Robotic Systems, vol. 95, no. 3, pages 851–869, September 2019. (Cited in pages 42 and 43.)

[Sisbot 2007] Emrah Akin Sisbot, L.F. Marin-Urias, R. Alami and T. Simeon. *A Human Aware Mobile Robot Motion Planner.* IEEE Transactions on Robotics, vol. 23, no. 5, pages 874–883, October 2007. (Cited in pages 26 and 27.)

[Sisbot 2010] Emrah Akin Sisbot, Luis F. Marin-Urias, Xavier Broquère, Daniel Sidobre and Rachid Alami. *Synthesizing Robot Motions Adapted to Human*

Presence. International Journal of Social Robotics, vol. 2, no. 3, pages 329–343, September 2010. (Cited in pages 27 and 95.)

[Strobach 2010] Peter Strobach. *The fast quartic solver*. Journal of Computational and Applied Mathematics, vol. 234, no. 10, pages 3007–3024, September 2010. (Cited in page 62.)

[Strobach 2015] PETER Strobach. *The Low-Rank LDLT Quartic Solver*. AST-Consulting Technical Report, vol. 10, no. 2.1, pages 3955–7440, 2015. (Cited in page 62.)

[Szadeczky-Kardoss 2006] E. Szadeczky-Kardoss and B. Kiss. *Tracking Error Based On-line Trajectory Time Scaling*. In 2006 International Conference on Intelligent Engineering Systems, pages 80–85, London, UK, 2006. IEEE. (Cited in page 86.)

[Tonietti 2005] G. Tonietti, R. Schiavi and A. Bicchi. *Design and Control of a Variable Stiffness Actuator for Safe and Fast Physical Human/Robot Interaction*. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pages 526–531, Barcelona, Spain, 2005. IEEE. (Cited in page 25.)

[Villani 2018] Valeria Villani, Fabio Pini, Francesco Leali and Cristian Secchi. *Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications*. Mechatronics, vol. 55, pages 248–266, November 2018. (Cited in pages 18, 19, 22, 23, and 36.)

[Welch 1997] Greg Welch. *An Introduction to the Kalman Filter*. page 16, 1997. (Cited in page 90.)

[Zanotto 2011] Vanni Zanotto, Alessandro Gasparetto, Albano Lanzutti, Paolo Boscariol and Renato Vidoni. *Experimental Validation of Minimum Time-jerk Algorithms for Industrial Robots*. Journal of Intelligent & Robotic Systems, vol. 64, no. 2, pages 197–219, November 2011. 00033. (Cited in page 36.)

[Zhao 2014] Ran Zhao, Daniel Sidobre and Wuwei He. *Online via-points trajectory generation for reactive manipulations*. In IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pages 1243–1248. IEEE, 2014. (Cited in pages 33, 35, 37, 43, and 93.)

[Zhao 2015] Ran Zhao and Daniel Sidobre. *Dynamic Obstacle Avoidance using Online Trajectory Time-scaling and Local Replanning:*. In Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics, pages 414–421, Colmar, Alsace, France, 2015. SCITEPRESS - Science and and Technology Publications. (Cited in page 86.)

[Zhao 2017] Ran Zhao and Svetan Ratchev. *On-line trajectory planning with time-variant motion constraints for industrial robot manipulators.* pages 3748–3753, Singapore, May 2017. IEEE. (Cited in pages 84, 85, 86, and 87.)

Abstract: It was in the 70s when the interest for robotics really emerged. It was barely half a century ago, and since then robots have been replacing humans in the industry. This robot-oriented solution doesn't come without drawbacks as full automation requires time-consuming programming as well as rigid environments.

With the increased need for adaptability and reusability of assembly systems, robotics is undergoing major changes and see the emergence of a new type of collaboration between humans and robots. Human-Robot collaboration get the best of both world by combining the respective strengths of humans and robots.

But, to include the human as an active agent in these new collaborative workspaces, safe and flexible robots are required. It is in this context that we can apprehend the crucial role of motion generation in tomorrow's robotics. For the emergence of human-robot cooperation, robots have to generate motions ensuring the safety of humans, both physical and physchological. For this reason motion generation has been a restricting factor to the growth of robotics in the past.

Trajectories are excellent candidates in the making of desirable motions designed for collaborative robots, because they allow to simply and precisely describe the motions. Smooth trajectories are well known to provide safe motions with good ergonomic properties.

In this thesis we propose an Online Trajectory Generation algorithm based on sequences of segment of third degree polynomial functions to build smooth trajectories. These trajectories are built from arbitrary initial and final conditions, a requirement for robots to be able to react instantaneously to unforeseen events. Our approach built on a constrained-jerk model offers performance-oriented solutions : the trajectories are time-optimal under safety constraints. These safety constraints are kinematic constraints that are task and context dependent and must be specified. To guide the choice of these constraints we investigated the role of kinematics in the definition of ergonomics properties of motions.

We also extended our algorithm to cope with non-admissible initial configurations, opening the way to trajectory generation under non-constant motion constraints. This feature is essential in the context of physical Human-Robot Interactions, as the robot must adapt its behavior in real time to preserve both the physical and psychological safety of humans.

However, only considering the trajectory generation problem is not enough and the control of these trajectories must be adressed. Switching from a trajectory to another is a difficult problem for most robotic systems in real applicative contexts. For this purpose we propose a strategy for the Reactive Control of these Trajectories as well as an architecture built around the use of trajectories.

Keywords: Online Trajectory Generation, Human-Robot Interaction, Reactive Trajectory Control, Smooth Trajectories, Motion Ergonomics