AI-Driven Reactive Control Strategy for Industrial Humanoid Robots

Côme Perrot

RÉPUBLIQUE FRANÇAISE
Liberté Égalité Fraternité

Université de Toulouse

Doctorat de l'Université de Toulouse

préparé à l'INSA Toulouse

Stratégie de contrôle réactif basée sur l'IA pour des robots humanoïdes industriels

Thèse présentée et soutenue, le 10 décembre 2024 par

Côme PERROT

École doctorale
SYSTEMES

Spécialité
Robotique

Unité de recherche
LAAS - Laboratoire d'Analyse et d'Architecture des Systèmes

Thèse dirigée par
Olivier STASSE

Composition du jury
M. Stéphane DONCIEUX, Président, Sorbonne Université
M. Nicolas PERRIN-GILBERT, Rapporteur, CNRS Paris-Centre
M. Jean-Baptiste MOURET, Rapporteur, INRIA Nancy
M. Ludovic RIGHETTI, Examinateur, NYU Tadon school of engineering
Mme Anastasia BOLOTNIKOVA, Examinatrice, CNRS Occitanie Ouest
M. Olivier STASSE, Directeur de thèse, CNRS Occitanie Ouest

Membres invités
M. Sébastien BORIA, Airbus

Seul l'inconnu épouvante les hommes.
Mais, pour quiconque l'affronte, il n'est déjà plus l'inconnu.

Only the unknown frightens men.
But for anyone who confronts it, it is no longer the unknown.

— *Antoine de Saint-Exupéry*

ACKNOWLEDGEMENTS

THE work presented in this PhD thesis has been achieved thanks to the support and contributions of many individuals who helped me navigate the challenges I encountered during three intense years.

I would first like to thank Jean-Baptiste Mouret and Nicolas Perrin-Gilbert for reviewing this thesis and for the valuable feedback they have provided. I would also like to extend my sincere gratitude to Anastasia Bolotnikova, Stéphane Doncieux, and Ludovic Righetti for being part of the jury and for the insightful discussions we had during the defense.

This work would not have been possible without the advice and support provided by my supervisor, Olivier Stasse. I would like to thank him for giving me the opportunity to work on this fascinating topic, to carry out experimental work on humanoid robots, and for allowing me the freedom to explore new scientific ideas and develop my own.

I am grateful for having the opportunity to work with Sébastien Boria and Noélie Ramuzat, who have been precious allies in facilitating a fruitful collaboration with Airbus.

Humanoid robot prototypes require a lot of work to get functioning, and none of the experiments would have been possible without the help of many experts, whose contributions are not always acknowledged to the level they deserve. I thank Pierre, Maximilien, and Guilhem for their regular help operating the robot, as well as Filippo, Niels, and Constant, for their punctual yet crucial contributions.

More generally, I am grateful to the many talented and passionate people I had the opportunity to meet at Gepetto, who helped me grow as a scientist and as a person. I would like to make a special mention of Nahuel, Ewen, and Sébastien, who were true mentors and helped me better understand the intricacies of model predictive control. Among all the people I had the chance to meet, and who are too numerous to mention here, a special mention goes to Gabriele, Pierre-Alexandre, Jason, and Wilson.

I would also like to thank Ariane, Ludovic, Sonia-Laure, and Virgile, with whom I shared my office, for their patience and understanding, especially during the final stages of my PhD.

Lastly, a heartfelt thank you goes to my family and friends for their unwavering support throughout this journey.

Merci à tous,

Côme Perrot

CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

$\mathbf{a}$	Set of parameters of the cost function optimized by the RL policy
$a_k(.)$	Activation function k of the cost function
$\mathbf{b}$	Generalized non-linear forces
$\mathbf{c}$	Cartesian position of the Center of Mass of the robot
$f(\mathbf{x}, \mathbf{u})$	Discrete dynamic of the system
$\bar{\mathbf{f}}$	Gap in the dynamic of the FDDP rollout
$\mathbf{J}_i$	Contact Jacobian at contact i
$\mathbf{k}$	Feedforward term of the DDP policy
$\mathbf{K}$	Feedback term of the DDP policy
$l(\mathbf{x}, \mathbf{u})$	Running cost of the Optimal Control Problem
$l_{term}(\mathbf{x})$	Terminal cost of the Optimal Control Problem
$\boldsymbol{\lambda}_i$	Contact force at contact i
$\mathbf{M}$	Joint-space inertia matrix
${}^A\mathcal{M}_B$	Transformation from frame A to frame B as an element of $SE(3)$
N	Number of nodes in the horizon of the MPC
$\mathbf{p}$	Cartesian position of the end-effector of the robot
$\mathbf{q}$	Configuration vector
$\mathbf{r}_k(.)$	Residual model k of the cost function
$\mathbf{s}$	State of the RL environment
$\mathbf{S}$	Actuation matrix
$\boldsymbol{\tau}$	Torque vector
$\mathbf{u}$	Control vector
U	Control sequence
w_k	Weight of the cost k of the cost function
$\mathbf{x}$	State vector

ACRONYMS

AI	Artificial Intelligence
CHOMP	Covariant Hamiltonian Optimisation for Motion Planning
CNRS	Centre national de la recherche scientifique (French National Centre for Scientific Research)
CoM	Center of Mass
Crocoddyl	Contact RObot COntrol by Differential DYnamic Library
DAgger	Dataset Aggregation
DDP	Differential Dynamic Programming
DDPG	Deep Deterministic Policy Gradient
DMP	Dynamic Movement Primitives
DoF	Degrees of Freedom
DQN	Deep Q-Network
FDDP	Feasibility-driven Differential Dynamic Programming
GAIL	Generative Adversarial Imitation Learning
GMM	Gaussian Mixture Model
GPU	Graphics Processing Unit
IL	Imitation Learning
iLQR	Iterative Linear Quadratic Regulator
IRL	Inverse Reinforcement Learning
IOC	Inverse Optimal Control
LAAS	Laboratoire d'analyse et d'architecture des systèmes (Laboratory for the Analysis and Architecture of Systems)
LFC	Linear Feedback Controller
LIPM	Linear Inverted Pendulum
LLM	Large Language Model
MDP	Markov Decision Process
Memmo	Memory of Motion
MoCap	Motion Caputre
MPC	Model Predictive Control
MPPI	Model Predictive Path Integral

NLP	Nonlinear Programming
NMPC	Nonlinear Model Predictive Control
OCP	Optimal Control Problem
PID	Proportional Integral Derivative
PPO	Proximal Policy Optimization
Pro-MP	Probabilistic Movement Primitive
PRM	Probabilistic RoadMap
QP	Quadratic Program
RL	Reinforcement Learning
ROB4FAM	Robots For the Future of Aircraft Manufacturing
ROS	Robot Operating System
RRT	Rapidly exploring Random Trees
SAC	Soft Actor-Critic
SLQ	Sequential Linear Quadratic
SRBD	Single Rigid-Body Dynamics
TO	Trajectory Optimization
TRPO	Trust Region Policy Optimization
URDF	Unified Robot Description Format
WBMPC	Whole-Body Model Predictive Control

Part I

INTRODUCTION

This introduction aims to present the scientific context in which this work is carried out.

This part is separated in two chapters. The first chapter aims to provide a high-level general overview of the potential future uses of robots. The second chapter presents a more detailed discussion of the specific context of this dissertation and outlines the contributions of this work.

1

HUMANOID ROBOTS: TOWARD THE NEXT INDUSTRIAL REVOLUTION?

In Short

The focus of this chapter is to provide a general introduction to robotics for industrial manufacturing, with an emphasis on the potential applications unlocked by humanoid robots. It aims to set the scientific challenges addressed in this dissertation within a broader context.
This chapter poses the question:
Why are humanoid robots an interesting topic?
Rather than attempting to answer this question exhaustively, it seeks to inspire reflection about some of the societal challenges that are inherently linked to robotics.

Contents

1 General context

Automation is a top priority for companies across the industrial world. The number of industrial robots in use worldwide has been steadily increasing, reaching nearly 4 million units in 2023 [182]. These systems are expected to play an even more significant role in the future of manufacturing, with major industrial actors forecasting billions of dollars of investments in automated systems [9].

The integration of robots in industrial processes has been, and will continue to be, a key driver of economic development. By automating repetitive tasks, robots enhance productivity, reduce operational costs, and minimize errors. Moreover, robots present an appealing solution to address the labor shortages that will multiply with the aging of the population. According to the World Health Organization, 1 in 6 people in the world

will be 60 years or older by 2050, with developed countries, such as Japan, already well over this rate [196].

Beyond economic benefits, robots have the potential to significantly improve working conditions by taking over hazardous, strenuous, and monotonous tasks. Industrial environments are known to be the cause of many health issues, whether due to exposure to harmful environments [141] or musculoskeletal disorders caused by repetitive motions [30]. Robots can operate in dangerous environments with exposure to toxic substances, extreme temperatures, or harmful noises. By replacing humans in these settings, robots reduce the risk of work-related injuries and illnesses. Furthermore, they contribute to creating a more comfortable working environment and help prevent musculoskeletal disorders by liberating humans from repetitive motions. Improving working conditions in this manner not only enhances the overall well-being of workers but also reduces the costs associated with workplace injuries and absenteeism.

2 CURRENT TREND IN MANUFACTURING

Always striving to achieve a better productivity, actors in the manufacturing industry have already extensively adopted automation technologies. Nevertheless, significant improvements in this area are still conceivable. According to a McKinsey research, manufacturing is the second industry that holds the most potential when it comes to automation opportunities [53].

However, it is essential to recognize that the manufacturing industry is highly diverse, and not all applications have the same potential for automation. [53] classifies manufacturing sub-sectors into three categories based on the skill level required of workers and the technological complexity of the products:

— Low-skill labor/low product complexity.

— Medium-skill labor/moderate product complexity.

— High-skill labor/high product complexity.

Not all sub-sectors within the manufacturing industry have reached the same level of automation. For low-product complexity industries, the primary barrier to automation is often related to cost. Since these industries usually exploit low labor costs, robots may not present a cost-effective solution.

On the other hand, technical feasibility is the main factor to consider for automation in complex manufacturing processes, such as aerospace manufacturing. The relatively low number of units produced, compared to other fields, means there are fewer repetitive tasks that can be easily automated with classic industrial robots. Moreover, these processes involve more complex work that require high safety and precision levels.

New technologies are necessary to make robots more useful.

While the former problem can be addressed by making existing technologies more cost-efficient, the latter requires the emergence of new technologies, leading to smarter and more flexible robots. New robot's capabilities would increase the number of tasks that can benefit from automation across various sub-sectors.

3 FREEING ROBOTS FROM THEIR FIXED BASE

This dissertation focuses on the new technologies that could enable more effective automation. An interesting avenue to integrate robots into previously unreachable processes is to focus on adaptability, which could allow the integration of robots into existing manufacturing processes with minimal disruption. A type of highly adaptable robots, often referred to as *Collaborative Robots* (Cobots), promises to offer the technology required to work alongside humans in shared workspaces [257].

This broad description encompasses a wide array of technologies. To bring more clarity to the understanding of collaborative robots, [18] proposes a framework defining four types of industrial human-robot collaboration:

— Coexistence: The robot is not in an enclosed secure cell, but does not share its workspace with humans.

— Sequential collaboration: Human and machine work in the same space but not at the same time.

— Cooperation: Robot and Human are both in motion at the same time to work on a common part.

— Responsive collaboration: Real-time adaptation of the Robot to the movement of the worker.

A simplified representation of these categories can be found in Fig. 1.1.

Most current applications fall into the first two categories, and actual responsive collaboration is still in the domain of theory. Nonetheless, the common denominator of all these applications, even the most conservative ones, is the ability of the robot to sense and react, to a certain degree, to its environment. This underscores the fact that reactivity is a major driver for the development of these new methods. More reactivity could bring more interaction capabilities but also adaptability to a broader range of uses.

In order to be actually qualified as cobots, robots must adhere to very strict safety regulations [117]. Since the subject of this work is focused on technologies with a low degree of maturity, it will not tackle cobots but instead aim to add more reactivity to the planning capabilities of robots, a foundational step toward truly collaborative robots.

Reactivity is a first step toward collaboration.

3.1 *Disruptive approach to robotics*

Recently, solutions to bring more flexibility to industrial robots have departed from incremental improvements over manipulator arms. The focus has shifted to humanoid robots because of their disruptive potential [22]. Indeed, their human-inspired design holds promise for a wide array of applications.

Firstly, legged locomotion could enable humanoid robots to traverse a wide variety of terrains that are not accessible by wheeled robots. This enhanced mobility could allow them to operate in complex environments, such as construction sites, disaster zones, or remote areas, where traditional robots might struggle.

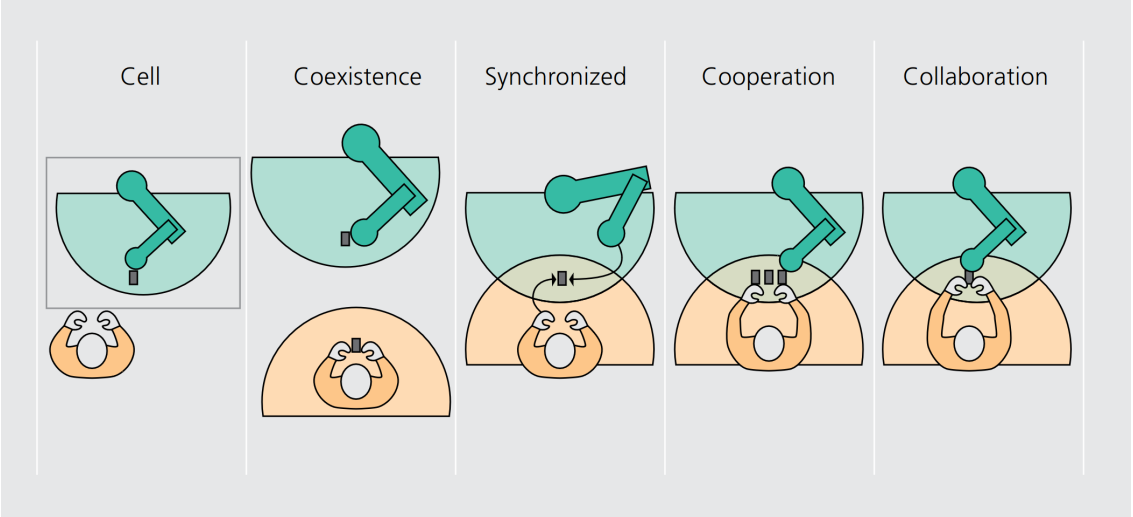


Figure 1.1 – Various levels of cooperation between a human worker and a robot [18].

Secondly, their bipedal form factor with two hands makes humanoid robots potentially ideal for manipulation tasks. They possess the mechanical capabilities to perform intricate operations that require dexterity and precision, similar to those executed by human workers. This capability would lead to new possibilities for collaboration and task sharing between humans and robots.

Moreover, humanoid robots appear to be a sensible solution to adapt to human-optimized environments, as their design is inherently adapted to navigate spaces and use tools created for human use. When integrating robots into the workplace, this adaptability would reduce the need for costly modifications to existing infrastructure.

Lastly, interactions with humans might be facilitated by the similarities in shape and movement between humanoid robots and their human counterparts. This resemblance could enable humanoid robots to exploit non-verbal communication cues, such as gestures and facial expressions, to better understand and respond to human intentions.

Exploiting the full capabilities of humanoid robots is challenging.

However, these promises can only become a reality if control methods fully exploit the capabilities of these platforms. Increased mechanical complexity in robotic solutions results in more challenging controls. For humanoid robots, tasks such as locomotion are far from trivial and have been a research topic for numerous years. Nevertheless, recent advances in Artificial Intelligence (AI) could significantly benefit robot control.

For instance, the Toyota Research Institute proposes a solution analogous to Large Language Models but tailored for robotics, called Large Behavior Models [253]. Such solutions, based on learning and exploiting vast amounts of data, could be the key to bridging the gap between laboratory environments, where most humanoids have been evolving for the past years, and factory floors.

3.2 *Humanoid robots in the industry today*

Numerous proposals to use humanoid robots in industrial settings have recently emerged (Fig. 1.2). Agility Robotics announced a partnership with the logistics provider GXO [6], marking the first formal commercial deployment of humanoid robots. Other humanoid robot manufacturers have announced collaborations with car manufacturers, such as Apptronik with Mercedes-Benz [13] and Figure with BMW [26]. These collaborations demonstrate the growing interest in humanoid robot technology for industrial manufacturing.

Tesla has also announced plans to deploy their Optimus robot for car manufacturing in the coming years [248]. Moreover, Boston Dynamics unveiled a successor to its hydraulic robot Atlas in the form of an electrically actuated robot geared toward industrial applications [31].

Several other companies are advertising humanoid robots for general industrial tasks, such as Sanctuary in North America [222], 1X in Europe [1], and Fourier [25] and Unitree [255] in Asia.

4 GENERAL SUBJECT OF THE DISSERTATION

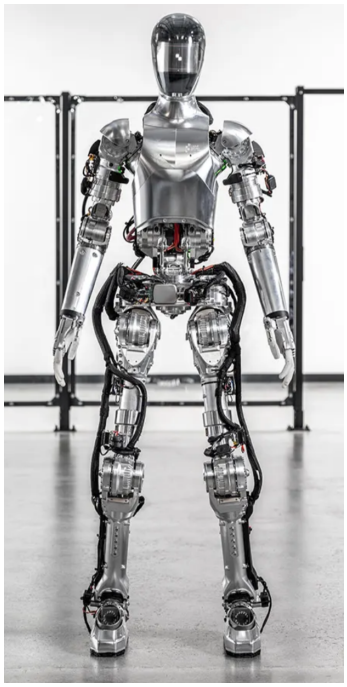
Humanoid robots, although a long-standing research subject, have only recently become a credible solution to address the challenges related to the automation of manufacturing. Their architecture allows them to move through unstructured environments and to exploit the full possibilities of environments designed for humans. This versatility makes them a sensible solution to automate tasks that have been, so far, out of reach for traditional robots. The implementation of robots inside factories also opens up a new realm of possibilities for more tightly integrated collaboration between humans and robots.

However, these promises bring their own set of challenges. Useful applications will only emerge if we manage to exploit these advanced robotic systems to the best of their abilities. Therefore, this dissertation aims to determine how to endow humanoid robots with reactive capabilities. In particular, it investigates whether novel learning-based methods can be leveraged for this purpose.

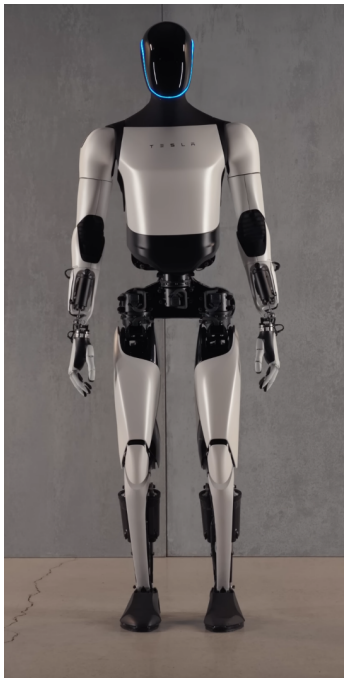
(a) Digit [5]

(b) Apollo [14]

(c) h1 [255]



(d) Figure [85]



(e) Optimus [251]

(f) Atlas [31]

Figure 1.2 – Industrial Humanoids

2

APPLICATION IN AN INDUSTRIAL SETTING

In Short

In this chapter, topic of this dissertation is narrowed down to a specific deburring task carried out by the humanoid robot Talos.
We also explain the nature of the collaboration between Airbus and the Gepetto team, from which this application arose.

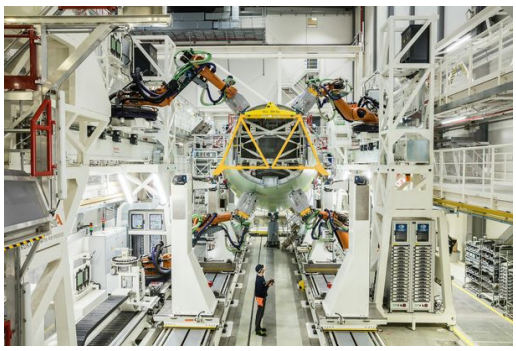
Contents

1 CONTEXT OF THE DISSERTATION

Among the many potential applications of humanoid robots in the industry, this dissertation focuses specifically on their use in aircraft manufacturing. Automation of airplane manufacturing has been a pressing concern for quite some time. Indeed, airplanes are primarily made of metal or composite plates riveted together, which requires drilling tens of millions of holes for every airplane [24]. However, to this day, this tedious task is still largely carried out by human operators (75% of the holes were drilled by hand, according to 2022 data).

Attempts have already been made to automate this task. In 2014, Boeing experimented with using KUKA manipulators on moving platforms (Fig. 2.1) to automate the assembly of the fuselage of their 777 series [91]. This solution involved pairs of robots working in coordination, one inside the fuselage and the other outside, to place fasteners. However, precisely coordinating the robots proved to be challenging, and the system never reached its intended efficiency, requiring human intervention to correct faulty parts. As a result, Boeing decided to revert to a solution called Flextrack, which

Figure 2.1 – Kuka robots in Boeing's assembly line [91].



(a) A320 assembly line in Hamburg [7]

(b) The Flextrack robot [8]

Figure 2.2 – Robotics solutions employed on Airbus's assembly lines

is also utilized by their competitor, Airbus. Flextrack moves on rails attached to the fuselage to drill the holes. Unlike the fully autonomous solution experimented by Boeing, Flextrack requires an operator to move the robot to different sections of the fuselage and to add the fasteners by hand.

Airbus is also heavily investing in modernizing its manufacturing processes. In 2019, a new assembly line for the A320 in Hamburg was inaugurated. It combines the Flextrack robot (Fig. 2.2b) with KUKA robotic arms (Fig. 2.2a), focusing on improving manufacturing efficiency without affecting product quality [7]. Additionally, Airbus makes significant efforts to develop, integrate, and maintain custom robots, aiming at a tight integration of new technologies into existing processes [8]. Because of this strategic orientation, Airbus is a fundamental partner to explore commercial applications of technologies developed inside research labs.

Despite extensive efforts directed at developing the use of robots in aircraft manufacturing, this topic is still far from solved. Aircraft manufacturing is particularly challenging because, to reach maximum efficiency, robots need to be able to move around the plane structure and work inside the fuselage. This involves climbing stairs and navigating cluttered environments. In this context, humanoid robots represent a promising research direction because their architecture allows them to reach the same places a human operator would [256]. However, for this integration to be successful, humanoid robots should be able to work in proximity with humans. That is why, as already detailed in Section 3 of Chapter 1, we believe reactivity to be a central point for the future development of robots.

Humanoids can potentially crawl through narrow spaces [75].

2 SCIENTIFIC COLLABORATION

This work was carried out in the Gepetto team at LAAS-CNRS. Moreover, it is the fruit of a collaboration with Airbus Operations SAS, which provided us with an application case and opportunities to carry out experiments in one of their factories. This collaboration spanned over two projects:

— Robots For the Future of Aircraft Manufacturing

— The Memory of Motion

2.1 *Robots for the Future of Aircraft Manufacturing*

Robots For the Future of Aircraft Manufacturing (ROB4FAM) is a joint laboratory between Airbus Operations and the LAAS-CNRS's Gepetto team, inaugurated in 2019. Its primary objective is to investigate innovative automation strategies for drilling and deburring tasks, which are critical in the aeronautical industry. The collaboration aims to integrate reactive robotic solutions into industrial aeronautic manufacturing processes, enhancing efficiency and safety.

This project was carried out along four axes [215]:

1. The first axis focused on augmenting the robot with reactive motion planning capabilities. This involved using visual servoing to account for uncertainties in the environment or the robot's actuators [192]. By demonstrating manipulation abilities on the robots TIAGo and TALOS [148], [177], the team showcased the robots' ability to plan and execute motions in real-time to perform specific tasks, such as aligning a tool with holes on an airplane part or flipping a wooden piece.
2. The second axis aimed to exploit torque and force measurements to provide a safer and more efficient control scheme. Research in this area included studies on actuator control [212], passivity-based control [211], and benchmarking control strategies [213]. This axis was crucial for enhancing the robots' task efficiency and ensuring safe interactions with human workers.
3. The third axis concerned the perception capabilities of the robot, enabling it to localize itself within a large factory environment [149], [150]. This aspect is a

necessary prerequisite for the robot to navigate and interact with its surroundings effectively.

4. The last axis specifically addressed the equilibrium of a humanoid robot in a complex environment [261].

The work presented in this document is attached to the first axis and involves performing deburring tasks with the humanoid robot TALOS.

2.2 *Memory Of Motion*

Memory of Motion (Memmo) is a European project carried out within the Horizon 2020 Program under Grant Agreement No. 780684. Initiated in January 2018, it brings together experts from the fields of motion planning, optimization, machine learning, and robotics, involving multiple European laboratories, universities, and companies. It is conducted with the support of key partners, among which PAL-Robotics (Spain) and Airbus (France). Similar to the ROB4FAM project, Airbus provides the targeted application, while PAL-Robotics contributes with their TALOS humanoid robot (bottom left in Fig. 2.3).

Figure 2.3 – Illustration of the robotic platforms involved in the Memmo project (Atalante, ANYmal, TALOS, and Solo) [67].

The primary aim of the Memmo project is to develop new control and planning methods to generate complex movements independently of the robot architecture. The scientific approach presented in this project is to exploit a library of pre-computed trajectories to improve the online optimization capabilities of the robots and to use exteroceptive sensors to enhance adaptability.

The Memmo project is a laureate of the Stars of Europe 2022, recognizing the successful collaboration of 11 European partners and the demonstrated results.

3 HARDWARE: THE HUMANOID ROBOT TALOS

The experimental work was conducted on the humanoid robot TALOS (Fig. 2.4), a 32 Degrees of Freedom (DoF) platform manufactured by the Spanish company PAL-Robotics. TALOS was developed in collaboration with the Gepetto team, with the goal of creating a robot capable of complex locomotion and bi-handed manipulation of significant payloads [242]. Standing at 1.75 m tall and weighing around 100 kg, TALOS is equipped with electric actuators and strain wave gearing, enabling it to lift more than 6 kg with a straight arm.

Its 32 DoFs are distributed as follows: two legs with 6 DoFs each, a waist with 2 DoFs, two arms with 7 DoFs each and a 1 DoF gripper, and a head with 2 DoFs. Most joints are fitted with a torque sensor, except for the head, wrists, and grippers. TALOS operates on two Ubuntu computers, one for control and one for vision processes, connected to all actuators and sensors via an EtherCAT bus.

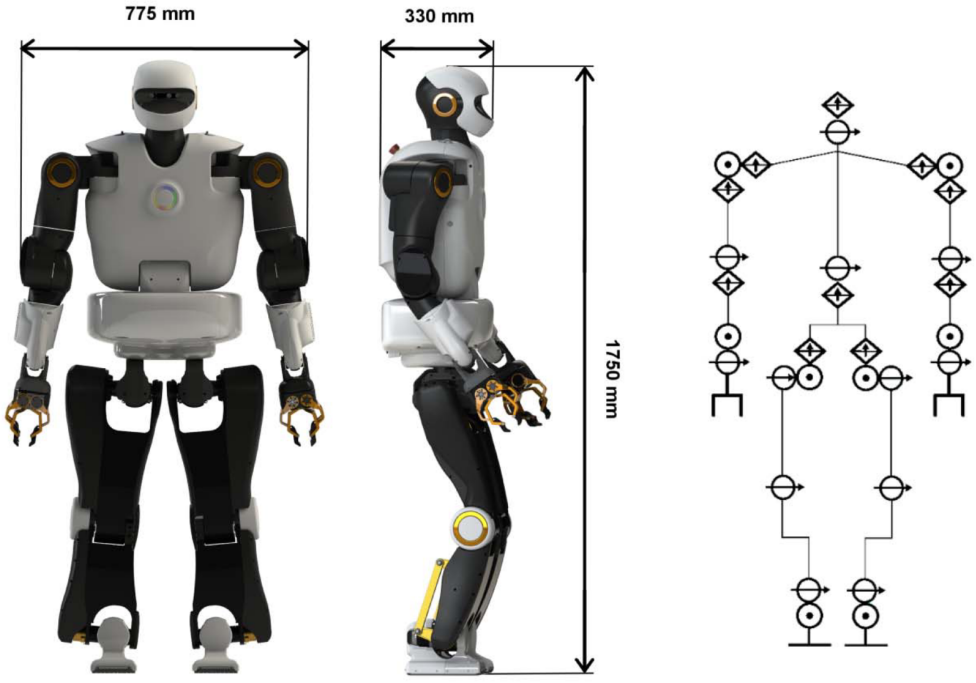


Figure 2.4 – The humanoid robot Talos [242].

The model used at the lab in Toulouse, named Pyrène, is the first TALOS ever produced. While subsequent iterations of TALOS have seen improvements such as updated torque sensors, Pyrène retains some differences due to changes in the production process that could not be replicated.

Because of the need for additional sensors to carry out some experiments, the head of the robot was modified. The modified head integrates a LiDAR with a wide field of view, a stereo camera pointing straight ahead of the robot, and an infrared-based RGB-D camera pointing at the ground in front of the robot (Fig. 2.5).

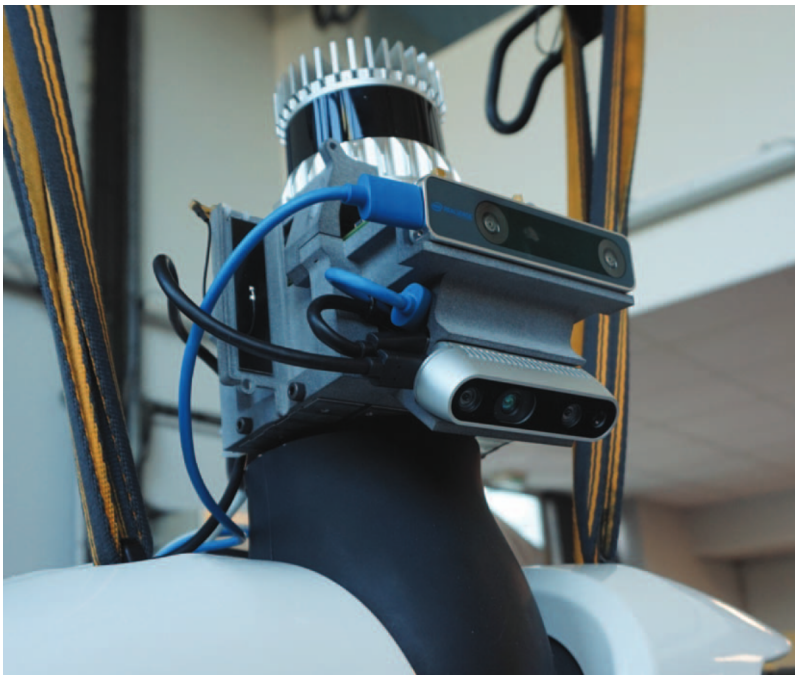


Figure 2.5 – The modified head of Pyrène with a LIDAR (top) and two cameras (middle and bottom) [150].

4 TASK STUDIED IN THE DISSERTATION

Among the various tasks in aircraft manufacturing, this dissertation focuses on deburring, specifically the deburring of holes on an aircraft engine pylon (Fig. 2.6a). The pylon, often made of titanium, is a critical mechanical component that attaches the engine to the wing (Fig. 2.6b). It supports the engine's weight, transfers thrust to the airframe, minimizes engine-wing airflow interaction, and protects the aircraft structure in case of engine failure.

Deburring is a post-drilling operation that removes material residues, which, if left unaddressed, could cause mechanical weakness, particularly in load-bearing parts like the pylon. Due to the size of the pylon and for confidentiality reasons, most of the work in this thesis was conducted on a 3D-printed mockup part provided by Airbus

(a) A320neo engine pylon [169].

(b) A320neo engine [269].

Figure 2.6 – Picture of an engine pylon (top) and the full assembly on a real airplane (bottom). The images are for illustrative purposes, and the pylon shown at the top may not exactly match the setting depicted at the bottom.

(Fig. 2.7a). This mockup represents a small section of the original piece and may have modified dimensions.

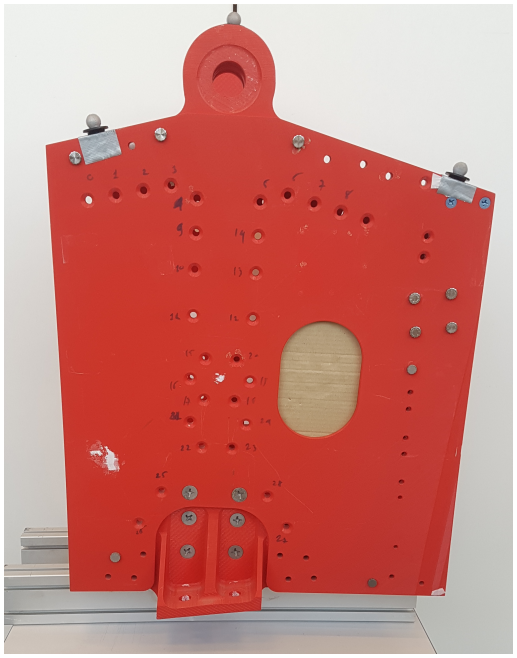
The deburring tool is represented as a 3D-printed part (Fig. 2.7b) rigidly attached to the robot's fingertip (Fig. 2.7c). In this study, the deburring task is simplified to a fine insertion task, as the actual deburring is not performed by the robot.

A fine insertion task is studied as a first step toward autonomous deburring.

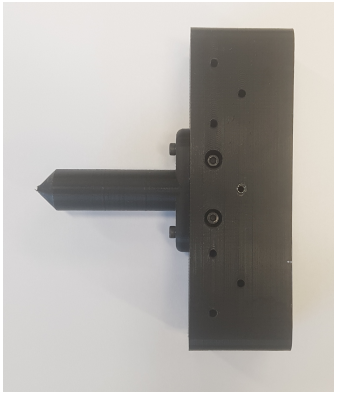
The work presented in this manuscript attempts to undertake this task with the humanoid robot TALOS, which was presented in Section 3. This is particularly challenging because, unlike dedicated manipulator arms, this type of hardware is not specifically designed for high-precision tasks. The diameter of the hole is less than 1 cm, and the robot's inherent flexibility and imprecision necessitate the use of advanced control techniques to achieve the correct placement of the tool.

Moreover, the goal is to control the robot using torque control, which adds another layer of complexity to the task. We believe torque control to be important because it enables finer control of the energy expended by the system compared to position control. This approach promotes safer robotics by limiting the energy input to what is required for a specific movement, preventing dangerous energy increases during unexpected events.

Torque control could unlock safer human-robot collaboration.



(a) 3D printed mockup of an aircraft pylon.



(b) 3D printed tool.

(c) Tool attached to the robot's gripper.

Figure 2.7 – Elements of the experimental setup.

5 GENERAL OUTLINE

In summary, the aim of this work is to realize the fine insertion of a tool inside the holes of an aircraft structure using a torque-controlled humanoid robot. The work presented here is organized according to the following outline:

— Part ii presents the state-of-the-art regarding motion planning and control, which are the two main fields of interest of this thesis. It tackles both model-based approaches that have been the long-standing baseline solutions and learning-based approaches that have emerged more recently.

— Part iii presents the technical and theoretical foundations of the WBMPC used to carry out the experiments on the robot. The main aim of this part is to present the technical choices as well as the challenges associated with deploying a controller on a full-size humanoid robot. The work presented in this part led to a demonstration for the Memmo project in June 2022.

— Part iv focuses on improving the performance of the controller. A first chapter in this part highlights the cost shaping issue that was encountered during the first experimental sessions. Then, a Reinforcement Learning based approach is proposed to alleviate this issue by leveraging experience of the robot.

— Finally, part v presents the conclusions of this work and the future perspectives.

6 PUBLICATIONS

This thesis was conducted at the Laboratoire d'analyse et d'architecture des systèmes (LAAS), an entity linked with the Centre national de la recherche scientifique (CNRS), under the supervision of Olivier Stasse. This work was supported by the cooperation agreement ROB4FAM and the MERLHBOT Région Occitanie project. The use of the experimental platform TALOS was supported by ROBOTEX 2.0 (Grants ROBOTEX ANR-10-EQPX-44-01 and TIRREX-ANR-21- ESRE-0015).

The research undertaken during this thesis led to the writing of several articles:

— C. Perrot and O. Stasse, « Step toward deploying the torque-controlled robot talos on industrial operations », in *International Conference on Intelligent Robots and Systems*, 2023, pp. 10 405–10 411. DOI: 10.1109/IROS55552.2023.10342428

— C. Perrot and O. Stasse, « Industrial operations with torque-controlled robot talos: an rl-mpc hybrid approach », Submitted to Transactions on Automation Science and Engineering, 2024

— C. Roux, C. Perrot, and O. Stasse, « Whole-body mpc and sensitivity analysis of a real time foot step sequencer for a biped robot bolt », in *International Conference on Humanoid Robots*, 2024, pp. 467–474. DOI: 10.1109/Humanoids58906.2024.10769884

Part II

STATE OF THE ART

This part of the manuscript provides a general introduction to the field of industrial robots. It presents the methods, both past and present, that can be used to execute useful motions on these machines.

3

MOTION GENERATION AND CONTROL IN ROBOTICS

In Short

In this chapter, the state of the art regarding the motion planning and control of robot is presented. Particular focus is given to :

— Classical model based approaches

— Emerging learning based solutions

Contents

1 INTRODUCTION

To enable true breakthroughs of robots in the industry, algorithms and techniques for motion generation and control are critical. In this chapter, an analysis of the relevant state of the art regarding robotic control strategies is presented, addressing both model-based and learning-based approaches.

The scope of this thesis is limited to motion planning and control, and we assume that the task to be carried out is already well-defined. Therefore, we will not dwell upon task planning but instead focus on methods that allow us to find the best motion for a pre-defined task and carry out that motion effectively on the robot. A structured inventory of these methods is porvided, including both classical and modern approaches.

2 MODEL BASED APPROACHES

The classical approach to robot motion generation involves separating the problem into two sub-problems: motion planning and control [34]. Motion planning involves finding a collision-free path for the robot between two given configurations, while control involves executing those movements accurately and efficiently while taking into account sensor feedback. This separation stems from the difficulty to control complex robots. Splitting the problem into smaller, more manageable sub-problems thus appears as a sensible strategy.

While some methods blurring the boundary between planning and control have emerged in recent years, this separation remains a useful starting point to get a good understanding of solutions that exist in robotics.

2.1 *Planning*

The problem of motion planning was originally known as the *Piano mover problem*, which involves finding a collision-free path for moving a large, rigid object through a cluttered environment. This purely theoretical problem has been extensively studied [230]–[233], [235]. It has also been applied to concrete industrial problems such as disassembly [83] and optimization of vehicle trajectories [147].

Rather than solving the movement of a complex object through a 3-dimensional space, roboticists study it in the configuration space of the robot [164]. This is a mathematical space that represents all possible configurations of the robot, where each point in the space corresponds to a unique configuration. The problem thus becomes that of finding a path between two points in a high-dimensional space.

We present a quick summary of the different approaches that can be adopted to tackle this problem:

— Deterministic Planning

— Sampling-based planning

— Optimization-based planning

For further references, a comprehensive introduction to motion planning can be found in [132]. A more recent state of the art, applied to hardware similar to the one studied in this thesis, can be found in [193].

2.1.1 *Deterministic Planning*

Deterministic planning is mainly of theoretical interest because it allows for a systematic, repeatable way to generate a trajectory. It can also identify cases where no trajectory exists. The idea is to describe the problem as a deterministic roadmap or a graph. Voronoi diagrams [249] and Canny's algorithm [38] are two example of such methods. Once this is done, a graph path search algorithm like Dijkstra algorithm [68] or A* [104] can be used to find a trajectory. However, these methods are generally

extremely computationally intensive, so they are not a viable solution for most problems that commonly arise in robotics, especially when tackling complicated humanoid systems.

To alleviate this problem, local field-based methods can be used [138]. These methods use a gradient-based approach and are less computationally intensive than deterministic planning methods. However, they are prone to getting stuck in local minima in complex settings.

2.1.2 *Sampling-based planning*

Sampling-based planning algorithms are capable of handling larger problems more efficiently than deterministic methods. They rely on the existence of efficient collision detection algorithms to swiftly determine whether a given configuration is in collision with its environment or not. This capability makes random exploration a viable strategy for quickly discovering new, potentially useful configurations.

These algorithms can generally be categorized into two groups:

1. Single query algorithms, designed for one-time use per problem, prioritize rapid exploration of the configuration space over the quality of its representation. A notable example is the Rapidly exploring Random Trees (RRT) algorithm [151].
2. Multiple-query algorithms, typically employed to provide multiple paths in a quasi-static environment, value an accurate representation of the configuration space. Most of these methods are derived from the Probabilistic RoadMap (PRM) concept, introduced in [131].

Despite their practical efficiency in addressing larger problems than deterministic algorithms, sampling-based methods have certain limitations:

— They offer weaker guarantees. While a solution will eventually be found if one exists, the planner will continue running indefinitely if no solution is available.

— They struggle with problems in which random sampling is unlikely to provide good estimates, such as scenarios involving narrow passages.

— Lastly, The random nature of the exploration may result in paths that are not optimal for the robot to follow.

A more comprehensive description of this topic can be found in [152].

2.1.3 *Optimization-based planning*

In the planning problem formulations discussed so far, optimality has been given little to no importance. The primary focus of planning has been to find a feasible path in a cluttered environment, with any given solution often deemed satisfactory. Even if the quality of the solution was of interest, it is not trivial to define an optimality criterion for a planning task.

Some methods nonetheless carry out motion planning through optimization. Optimal variants of RRT and PRM have been proposed in [130]. These algorithms guarantee convergence towards the globally optimal solution of the motion planning problem.

Methods like Covariant Hamiltonian Optimisation for Motion Planning (CHOMP) [278] and its evolution TrajOpt [226] require an initial trajectory as input. They iteratively improve the initial trajectory to construct an optimal solution.

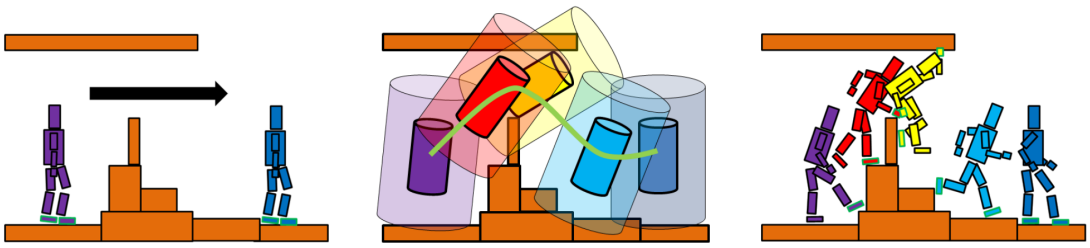
The connection between optimization and motion planning will be further explored in Section 2.3.

2.1.4 *Planning of humanoid movements*

The planning of humanoid robot movement is a complex task due to the unique characteristics of the system. The under-actuation of humanoid robots means that they must rely on external forces to control their Center of Mass (CoM). This results in frequent shifts in contact during locomotion or manipulation, which significantly impacts the system's dynamics and makes the planning process challenging.

The Gepetto team proposed an approach to address these challenges by splitting the complex problem of humanoid planning into several more manageable sub-problems [42]. The problem is divided into three main steps:

— *Contact sequence planner*: This first step aims to compute a sequence of contacts according to the desired behavior of the robot [252]. First, the planner considers only the root of the robot, ensuring that the robot is close enough to obstacles to be within reach of the limbs but at a distance that guarantees contact avoidance. Then, a discrete sequence of statically stable configurations is generated along the found path. [82] extends this approach with new steering method to account for dynamic transitions. The workflow of the approach is summarized in Fig. 3.1.



(a) Initial request (b) Reachability planning (c) Contact sequence

Figure 3.1 – Overview of the contact sequence planner [44]. An initial movement request is first transformed into a trajectory of the root of the robot, which is then used to obtain a discrete contact sequence.

— *Centroidal pattern generator*: This step computes control for the centroidal dynamics of the system to obtain a smooth movement by taking as input the contact sequence generated at the previous stage of the framework [44].

— *Whole-body motion generator*: The last step uses an RRT-based approach to compute whole-body motions and plan the position of the end-effectors while following the desired trajectory of the CoM [44].

The information obtained through this process can then be fed to a controller to obtain the torques that must be sent to the robot. However, splitting the problem in this manner does not guarantee that a solution will lead to a feasible problem for subsequent steps. This means that in the case of infeasible movements, the whole pipeline must be evaluated again, and there are no solutions to address this issue locally. This is one of the reasons that explain the emergence of integrated planning methods, which will be explained in Section 2.3.

2.2 *Control*

Assuming that the motion planning problem has been solved, it is necessary to design a control solution to ensure that the desired trajectory is accurately tracked by the robot. In robot control, this control phase is typically divided into two levels:

— *Low-Level Control:* This controller operates at high frequencies (up to several kHz) directly at the actuator level. Its function is to precisely regulate the actuator dynamics, leveraging the specific characteristics of the actuators, which can either be identified on existing hardware [66] or integrated into new designs [96].

— *High-Level Control:* This controller addresses the architecture-specific requirements and formulates a control law that satisfies system constraints, such as torque limits. It is responsible for bridging the gap between the high-level motion plan and the commands executable by the low-level controller. The high-level controller compensates for model limitations encountered during the planning phase and ensures the plan's feasibility in real-world scenarios.

In the context of robotics, the low-level controller is the distinctive element that differentiates position control from torque control.

Position control translates joint positions into motor commands. It has been successfully applied to execute walking motions on humanoid robots [39]. However, this control type is less effective in handling unforeseen disturbances or achieving compliant behavior.

To overcome these limitations, torque control directly regulates the torque output of the actuators, offering better adaptability and compliance. Nonetheless, the additional modeling required for successful deployment on real hardware poses significant challenges [73].

Given that the design and implementation of low-level controllers, including modeling and compensation of actuator effects, are outside the scope of this PhD, these aspects will not be further elaborated. Instead, we will focus on high-level control by detailing the following strategies:

— *Joint Space Control*

— *Operational Space Control*

For a more comprehensive understanding of motion control, including detailed mathematical developments, the reader can refer to [54], [234].

2.2.1 *Joint Space control*

Joint Space Control involves controlling the robot directly in its joint space. The most straightforward and widely used control scheme in this context is the Proportional Integral Derivative (PID) controller. This control scheme has been studied for many years [276]. One of its advantages lies in its simplicity and clear physical meaning [52]. Additionally, its theoretical foundations have been extensively researched [10], [15]. These factors combine to make PID control a solid choice for industrial applications and explain why it remains prevalent to this day.

However, this approach presents significant limitations when controlling highly non-linear systems. The simple feedback loop may not be sufficient to achieve the desired performance. Moreover, most industrial applications require very precise placement of the robot's end effector, necessitating extremely stiff feedback gains. This means that while tracking performance is improved, compliance is reduced, which may not be optimal in unstructured environments or when dealing with unforeseen disturbances.

The simplest form of PID control falls into the category of independent joint control, meaning that the control input for each joint is not influenced by the state of other joints. To address situations where this assumption is limiting, such as with manipulator arms that have long kinematic chains, a feed-forward term can be incorporated. The most common example in robotics is adding a gravity compensation torque [134].

Another limitation of PID is that a fixed set of gains, tuned for a specific system model might not deliver satisfactory performances on an uncertain system. Two main families of solutions exist to address this issue: adaptive control and robust control. Adaptive controllers [112] augment traditional fixed-gain controllers with a strategy that adjusts the gains based on signals from the closed-loop system to make them time-varying. On the other hand, robust control [3] aims to optimize a single set of gains to achieve the best average performance across various situations, even in the presence of uncertainties such as mathematical model inaccuracies and unmodeled dynamics. Adaptive control is applicable to a wider range of uncertainties, whereas robust control is generally easier to implement.

2.2.2 *Operational Space control*

Despite being a straightforward control technique, operating in the joint space is sometimes not the best way to specify the movement of a robot. The operational space framework was introduced to focus on task execution and make control more intuitive [137].

In this approach, tasks are defined as the error between a current and desired value for a specific robot feature. Originally conceived to control the end-effector position of a manipulator arm, operational space control can also apply to a broader range of features, for example, the position of the feet or the position of the CoM in humanoid robots [76], [221].

The main goal is to regulate this specific quantity to zero. Desired damping can be achieved by specifying values on the derivatives of the error [256]. Another approach

involves designing a control Lyapunov function to ensure the error converges to zero [11].

Other solutions handle inequality tasks [65], [200]. However, this approach is by nature instantaneous, which means that guaranteeing constraints on the system's state over an extended period of time is inherently precarious.

Operational space methods generally rely on solving a Quadratic Program (QP) and can benefit from mature, off-the-shelf solvers [84], [243]. The specificity of the resolution lies in the method chosen to accommodate the hierarchy between tasks. The simplest approach is to weight each goal and rely on the relative weight differences to prioritize tasks [33], [55]. A second approach enforces a strict hierarchy by solving the lower-priority tasks in the null space of the previously solved tasks [129], [187].

For a more comprehensive overview of this approach, refer to [263].

2.3 *Toward unified motion generation and control*

Control methods presented so far often fail to provide assurance for the complete motion of the robot. To tackle this challenge, MPC utilizes a model of the system to forecast its evolution over a specified time. This strategy enables online prediction of constraint violations and timely recomputation of a suitable control strategy. Originally developed for controlling process plants [194], [207], where the slow dynamics permitted online solution re-computation, MPC has now become a viable option for real-time robot control, thanks to advancements in computational methods and power.

The core concept of MPC [175] involves solving an open-loop Optimal Control Problem (OCP) online, using the measured system state as the initial condition. Only a fraction of the resulting control is applied to the system before the process is repeated. A core component of the MPC, distinguishing it from traditional control strategies, is a model of the controlled system that provides foresight into future system states.

Most MPC approaches rely on a two-step pipeline, where a planner generates a reference trajectory, which is then tracked by the controller [86]. However, MPC's online replanning capabilities can blur the lines between control and planning, aiming to unify motion planning and control into a single solution.

Regardless of how Model Predictive Control is integrated into the control pipeline, selecting an appropriate model is crucial for achieving optimal performance. This decision involves a trade-off between model representativity and the allocated computational power for solving the OCP.

For a comprehensive review of MPC applications in legged robots, refer to [241].

2.3.1 *Linear MPC*

The simplest model used for bipedal robots is the Linear Inverted Pendulum (LIPM). Introduced in [127] as a 2D dynamical model. It has since been extensively generalized, as seen in [40]. The LIPM is derived from centroidal dynamics of the robot, neglecting tilting movements around the CoM.

The linearization of the dynamics allows the problem to be approached as a QP. Many dedicated solvers have been developed for this purpose, with an overview provided in [146]. Most strategies rely on an instantaneous whole-body controller to apply the MPC solution to the robot. An overview of linear MPC for bipedal robots is proposed in [265].

[126] uses a reference trajectory computed during footstep planning to improve computational efficiency, though this approach cannot account for inequality constraints. [266] addresses this issue by exploiting improved computational performance and efficient QP solvers to formulate a linear MPC that explicitly considers inequality constraints. [106] further extends this approach to allow simultaneous planning of CoM trajectory and foot placement. Other extensions focus on accounting for external forces on the robot's hand, enabling loco-manipulation applications [171], [181], [183], [244].

2.3.2 *Simplified non-linear MPC*

A more advanced model is the centroidal dynamics, described in [41], which details the dynamics of the Center of Mass and the total angular momentum. A common simplification of this model, Single Rigid-Body Dynamics (SRBD), considers only the inertia of the base link. This approximation makes SRBD primarily suitable for MPC in robots with light limbs, such as lightweight quadruped robots [27].

Since the model is nonlinear, a Nonlinear Model Predictive Control (NMPC) must be employed. [188] proposes a solution that optimizes foot placement and handles local obstacles for the humanoid robot HRP-2. [214] implements a real-time NMPC with terrain adaptation on the HyQ quadruped.

The limitation of the centroidal MPC approach is that it does not guarantee constraint satisfaction for the generated movements. To address this, some methods enforce kinematic limits along the horizon in addition to the centroidal dynamics, ensuring the feasibility of whole-body motions. For instance, [79] employs a parallelized Sequential Linear Quadratic (SLQ) algorithm to compute real-time movements on the HyQ quadruped, considering both the CoM dynamics and the full robot kinematics. Similarly, [93] adopts a Differential Dynamic Programming (DDP)-based solver for an ANYmal robot.

2.3.3 *Whole-body dynamics MPC*

MPC using the full dynamics of the system is a promising solution that could unlock the full potential of robot hardware. Its complexity has long prevented its use on real robots, but thanks to advancements in software and hardware, real-time whole-body MPC applications have become a reality.

Initial applications relied on smooth contact dynamics [118], [145], [189]. This approach does not require a predefined footstep sequence, as it can adapt step timings during optimization, effectively replacing motion planning and control with a single component.

In contrast, [60] demonstrates locomotion on a humanoid robot, Talos, using a precomputed contact sequence. It employs the algorithm developed in [172], which exploits the derivation strategy introduced in [43] to achieve sufficient computational speed.

[140] uses a similar strategy but with implicit contact dynamics in the backward pass of the DDP, eliminating the need for precomputed step sequences.

WBMPC offers the advantage of being seamlessly applicable to various tasks and hardware. For example, [143] uses the same solution as [60] to perform pointing tasks on a manipulator. [101] extends this approach to obstacle avoidance. [59], [144] demonstrate sanding and pointing tasks on a full-size humanoid robot, paving the way for loco-manipulation.

[122], [173] address a shortcoming of the method presented in [172] by accepting hard constraints.

Finally, some solutions [158] combine two types of models within the same horizon, leveraging the accuracy of the full dynamic model while allowing for longer prediction horizons with a simplified model.

2.3.4 *Sampling-based optimization methods*

Methods presented so far can be classified as gradient-based methods, as they rely on first or second order linearization of the dynamics to compute a local improvement direction for the predicted control. However, computing the derivative of the dynamic can be challenging, especially for legged-locomotion applications where interaction with the environment through contact leads to non-smooth physical phenomena. To address this issue, gradient-based methods often rely on specifically tailored models or heuristics, such as using a predefined contact sequence [60].

This weakness limits the generality of these methods and requires additional tuning to apply them to new situations. [153], [210] propose extensions of DDP to tackle non-smoothness in a black-box manner by leveraging sampling techniques.

These methods draw a link between gradient-based methods and sampling-based methods, also referred to as zeroth-order methods. Zeroth-order methods such as Bayesian optimization [87] or Evolutionary Algorithms [268] have been largely used for solving non-convex and non-smooth optimization problems. They have been used, for example, for hyperparameter tuning [57] or to optimize the mechanical design of quadruped robots [78]. They also have more recently been applied to online motion planning and control with the introduction of Model Predictive Path Integral (MPPI) [270].

However, even if they represent a promising lead to increase the generality of control methods for humanoid robots, they suffer from the curse of dimensionality and often rely on parallelized sampling to be efficient. This parallelization is typically achieved using Graphics Processing Units (GPUs), which are not commonly embedded directly on robots due to power and space constraints. This limitation hinders the real-world application of sampling-based methods for the whol-body control of humanoid robots.

3 LEARNING BASED APPROACHES

Differing from traditional model-based approaches, solutions leveraging large quantities of data have demonstrated significant potential to enhance adaptability and perfor-

mance in dynamic and uncertain environments. Learning-based methods, by harnessing the power of data, enable robots to learn from experience, thereby improving their ability to handle complex tasks and varying conditions.

Within learning-based approaches, two main methodologies stand out:

— *Imitation Learning*: This approach focuses on exploiting datasets of expert demonstrations to extract features and patterns that can be used for robot control. Imitation Learning is particularly effective when high-quality expert data is available, allowing robots to learn complex tasks by mimicking observed behaviors.

— *Reinforcement Learning*: It involves the robot interacting with its environment to develop a control policy. Through trial and error, and guided by a reward signal, the robot learns to optimize its actions to achieve desired outcomes.

For further reference, an extensive survey on learning methods applied to legged robots can be found in [98].

3.1 *Imitation Learning*

Imitation Learning (IL) involves training a robot to perform tasks by observing and mimicking expert demonstrations. This paradigm is particularly useful in scenarios where high-quality expert data is available, and exploration through interaction may be impractical or unsafe. The fundamental challenge in IL lies in effectively leveraging the provided data to develop robust control policies. One significant difficulty is the compounding nature of errors during control, where small deviations from the expert trajectory can accumulate over time, leading to a substantial mismatch between the robot's state and the training dataset, particularly in long-horizon tasks.

To address these challenges, several methods have been developed to enhance the robustness and generalization of learned policies. Generative Adversarial Imitation Learning (GAIL) [109] and Dataset Aggregation (DAgger) [217] are two prominent approaches. GAIL leverages adversarial training to encourage the robot's behavior to be indistinguishable from that of the expert, while DAgger iteratively refines the policy by incorporating the robot's own experiences into the training data, thus mitigating error accumulation.

The quality and diversity of demonstration data are crucial for the success of IL. This data can be collected through various means, such as teleoperation, where a human operator directly controls the robot [2], [205], or by retargeting from biological systems, which involves transferring skills from biological entities to robots [260]. Additionally, other control strategies, which may not be suitable for real-time robot control but are valuable for generating training data, can be utilized [197].

An important aspect of IL is the choice of method to represent the policy effectively. In the following sections, we will review three relevant approaches to policy representation:

— Motion Models: Utilizing mathematical models to synthetically represent complex motions.

— Deep Learning: Leveraging neural networks to encode motion policy.
— Inverse Reinforcement Learning: Inferring the underlying cost function that an expert optimizes.

3.1.1 *Motion Models*

We define motion models as mathematical representations that efficiently encode complex movements using a limited set of parameters. This approach is inspired by the study of biological systems, which exhibit highly dynamic movements. In such cases, storing individual control strategies for every possible scenario becomes impractical, prompting the hypothesis of internal synthetic representations [185]. The concept involves extracting these internal models and learning a concise set of parameters through demonstrations.

One widely used tool for this purpose is the Gaussian Mixture Model (GMM) [136], which characterizes movement trajectories as combinations of Gaussian distributions.

Another significant mathematical framework for motion primitives is the Dynamic Movement Primitives (DMP) [116], [224]. DMPs provide a structured approach to capturing and reproducing complex motions, facilitating adaptation to varying environmental conditions and task requirements.

For instance, in the context of teleoperation with communication delays, Probabilistic Movement Primitives (Pro-MPs) [199] have been employed to predict robot movements effectively [201]. [223] offers a comprehensive survey of the evolution and applications of DMPs in robotics research.

3.1.2 *Deep Learning*

Deep learning [154] approaches leverage the powerful synthesis capabilities of neural networks to minimize the need for explicit feature engineering in learning processes. The core principle involves using neural networks, trained on vast amounts of data, to encode complex policies [206], [273].

Recent advancements in deep learning have focused on creating more flexible policies to accommodate multiple tasks, moving towards a generic policy for humanoid robot control. One solution [70] exploits the reasoning capabilities of a Large Language Model (LLM) to develop a generic policy. Diffusion Policies [51] demonstrate graceful handling of multimodal action distributions commonly found in robotic manipulation tasks.

3.1.3 *Inverse Reinforcement Learning*

Inverse Reinforcement Learning (IRL) [190] also referred to as Inverse Optimal Control (IOC) aims to infer a cost function that is minimized by an expert by observing its behavior. One advantage of IRL, compared to other IL methods, is that it offers a more easily explainable result. By analyzing the cost function, insights into the expert's functioning can be gained, which is not possible with deep learning methods.

For example [170], [180] present IOC as a promising strategy to control a humanoid robot. More recently [19] leverages IOC to study the human gait.

3.2 *Reinforcement Learning*

The idea behind Reinforcement Learning (RL) is to define a task through a scalar reward function and learn a policy that maximizes this reward through trial and error interactions with the environment [247].

The work proposed in [179] introduced Deep Q-Network (DQN), the first successful combination of Deep Learning and RL, laying the foundation for several major improvements in this field. Trust Region Policy Optimization (TRPO) [227] addressed instability issues of DQN by using a trust region constraint to regulate the divergence between two updates of the policy. This work was later refined into Proximal Policy Optimization (PPO) [228] which is still one of the most widely used algorithms.

Deep Deterministic Policy Gradient (DDPG) [160], extended DQN to continuous action spaces, which are prevalent in robotics applications. It was followed by numerous variations [90], [99], [100], [178].

The proliferation of available algorithms has also led to a surge in proposed implementations [32], [77], [113], [209]. However, this poses the issue of reproducibility. Indeed, small implementation details can have major effects on performances [4], [105].

The variability of results obtained with RL methods can be further attributed to the design choices made in the control pipeline. We will study three crucial elements in the design of an efficient RL pipeline:

— Hyperparameters: The parameters that need to be chosen before deployment.

— Sample efficiency: The strategies to allow for sufficient trial-and-error to achieve complex behaviors.

— Sim-to-real: The challenge of deploying the obtained policy in the real world.

3.2.1 *Hyperparameters*

An important element that can significantly affect an algorithm's learning ability is the choice of hyperparameters. Although selecting an appropriate algorithm can alleviate this issue, it remains a major difficulty for RL practitioners. This challenge is exacerbated by the fact that RL trainings tend to be extremely computationally intensive, making systematic exploration of the hyperparameter space impractical. While some methods have been proposed to overcome this obstacle [119], hyperparameter tuning is still largely performed by human experts through costly trial and error.

Another crucial design element in RL is the reward function. Algorithms tend to require dense information to converge to a satisfactory policy. However, robotic tasks are often inherently sparse, for example when success can only be evaluated after a sequence of actions. The most common approach is to add hand-tuned reward elements to guide the agent towards favorable regions, but this can be cumbersome and lead to suboptimal results.

Some strategies aim to exploit sparse rewards more intelligently [12], [47], [198]. Other methods rely on demonstrations to overcome the exploration challenge associated with sparse rewards [186], [238], [258], [274]. A complementary approach is to

facilitate shaping by reducing the number elements that needs to be accounted for in the reward. For example, some methods consider constraints independently of the reward [46], [142], [155], effectively reducing the number of penalizations that need to be integrated in the reward.

3.2.2 *Sample efficiency*

Because of its principle, Reinforcement Learning requires a vast amount of data to achieve successful results. For instance, a major advancement presented in [195] required several weeks of runtime on computer clusters. The training amounted to a tremendous quantity of energy expanded to learn how to solve a Rubik's cube. While not diminishing the pioneering role of this publication, the magnitude of resources involved makes this approach impractical for most cases.

Improving the efficiency of the algorithms as therefore been a central focus in the Reinforcement Learning community. One approach is to make RL more sample efficient. This can be achieved by increasing the stability of the algorithm to improve the update-to-data ratio and accelerate convergence [50], [108]. Another solution, often referred to as model-based Reinforcement Learning, involves applying planning methods to learned models of the environment to minimize the number of interactions needed for convergence [102]. Strategies mentioned in Section 3.2.1 can also be applied to limit the initial random exploration phase of the algorithm.

Another approach has been to increase the throughput of simulators to generate more samples in a shorter amount of time. This is primarily driven by advances in GPU-based computation [167]. These technical improvements have unlocked possibilities that once seemed out of reach for RL methods [110], [208], [220].

3.2.3 *Sim-to-real*

Another important consideration is the ability to transfer the policy to real robots. A popular approach is to use domain randomization [49]. Some methods collect data to build more representative models [97], [115].

Another sim-to-real strategy involves separating the training into two phases. In the first phase, training is carried out with privileged information that will not be available to the robot in the real world. In the second phase, the robot learns to imitate the first policy but has access only to an accurate representation of real-world sensor data. This learning from cheating [48] enables the handling of challenging tasks and their transfer to the real world [45], [163].

The last approach is to learn directly on the robot to eliminate the need for models [239], [271]. This approach requires highly sample-efficient learning algorithms and hardware robust enough to withstand the control policy of the RL [36].

4 HYBRID APPROACHES

Both model-based and learning-based methods, as presented in the previous sections, have their strengths and weaknesses. To achieve the best of both worlds, a number of works have aimed to exploit those two methods in an integrated manner. However, as these techniques tend to be at the intersection of two research fields, their taxonomy is not well-defined.

To provide a clearer understanding, we propose to inventory them into two groups:

— Methods that aim for more efficient exploration of the environment

— Methods that seek to achieve learning with safety guarantees

This separation is arbitrary and aims to present solutions particularly relevant to the topic of this dissertation. However, it does not aim to be exhaustive, and some approaches already mentioned, such as model-based RL in Section 3.2.2, could be classified into one of these two categories.

4.1 *Models for exploration*

A major weakness of learning approaches is that they require large amounts of data to be efficient, as explained in Section 3.2.2. In the case of RL, this means that the agent must encounter a lot of successful examples before converging to a satisfactory policy. However, for difficult tasks like locomotion for humanoid robots, this requires a long training time and might altogether prevent the system from converging [174].

A promising strategy to solve this issue is to leverage the structure provided by well-known Trajectory Optimization (TO) approaches, which can provide useful examples and only use RL to explore around those solutions.

[89] build upon work presented in [202] to obtain dynamic quadruped behaviors. The idea is to use TO computed from a simplified model to generate reference trajectories for an RL controller, which takes into account the full dynamics of the system. Similarly, [123], [128] use a model-based planner to provide a reference motion during training to achieve robust legged locomotion. [16] learns by reinforcement the difference between a simplified model and a full model for a footstep planner in the case of humanoid locomotion. These approaches demonstrate superior robustness when compared to model-based methods while benefiting from the knowledge of model-based solutions. However, adding an imitation reward might bias the system toward suboptimal solutions. Care must be taken to only use demonstrations when they are useful, as suggested by [186]. The other main drawback of this approach is that it relies on demonstrations, which might not be available for new applications or very challenging tasks.

Other strategies leverage differentiable simulators [229], [240], where the gradient information from the model of the system is exploited by the learning agent to increase the convergence rate. However, it is not clear if adding gradient information to the simulators is always beneficial [246]. Indeed, it might provide noisy information when

dealing with stiffness or discontinuities in the dynamics, which are common in robotic applications.

Lastly, [166] expands on the method developed by [20] by combining a learned locomotion policy with an MPC for manipulation. The idea is to carry out the precise manipulation task with MPC and use RL only for locomotion, treating the dynamic effects of the arm as a disturbance for the RL controller. However, this approach is only relevant if a good enough model-based controller exists for manipulation and cannot be used to address the shortcomings of MPC.

4.2 *Toward safe learning*

One of the main drawbacks of end-to-end learned architectures is the lack of guarantees they offer, making them unsuitable for safety-critical applications. To address this issue, a solution present in the literature is to leverage model-based methods as a way to ensure safety within a learned pipeline [35], [107].

For example, [272] uses MPC as a function approximator within an RL framework to benefit from the guarantees offered by MPC while learning the model discrepancies between MPC and reality. [216] integrates a differentiable MPC as the final layer of an actor within an actor-critic framework.

[161], [162] modify the action space of the RL agent to act on the tangent space of the constraints, providing guarantees even during the training phase.

5 CONCLUSION

The primary focus of this thesis is on movement planning and control for humanoid robots for manipulation tasks. We thus assume that the task planning step has already been addressed, and that the sequence of movements that needs to be carried out is known. We have seen in this chapter that planning and control for humanoid robots is a challenging task and that, while effective in certain contexts, traditional approaches often lack generality. On the other hand, learning approaches have recently shown promising results but lack the maturity of model-based approaches, which slows down their adoption for industrial applications.

We have chosen to use a Whole-Body Model Predictive Control (WBMPC) as the foundation for our work. This choice is motivated by the promising results MPC has demonstrated in handling both planning and control in a unified framework. The ability of MPC to manage these aspects simultaneously is particularly appealing, as it aligns well with the flexibility expected from a humanoid robot.

However, recognizing the limitations of MPC, particularly its reliance on accurate models and well-defined cost functions, this work aims to explore hybrid approaches that combine MPC and RL. By integrating MPC with RL, we seek to enhance the overall effectiveness and adaptability of our framework without sacrificing the recent progress made thanks to WBMPC.

Part III

A HUMANOID CONTROLLER

This part presents the necessary elements to understand the control structure that was used to carry out experiments on TALOS.

Chapter 4 presents the theoretical foundation of the MPC deployed on the robot.

Chapter 5 focuses on the practical aspects of the experiments.

4

A DEBURRING CONTROLLER

In Short

This chapter presents the theoretical foundation of the controller used in the experiments conducted during this dissertation.

Contents

1 INTRODUCTION

TALOS is controlled in torque using Whole-Body Model Predictive Control (WBMPC).

We chose a torque control approach because it promises better flexibility, adaptability, and safety. For example, torque control allows the robot to be more compliant. If the robot encounters an obstacle or a human, a torque-controlled robot can react more softly and reduce the risk of damage or injury compared to a position-controlled robot that might try to force its way through to reach a specific position. Although this is just a step towards safer robots, and does not render further research on enforcing compliance and permitting human interaction pointless, we believe torque control to be a promising robot control strategy.

Torque control is a step toward safer robots.

In addition, we exploit Model Predictive Control because, contrary to classic control, it has predictability capabilities which can be exploited to offer guarantees on a complex system. The whole-body model is necessary for undertaking general tasks that utilize the full capabilities of the robot. It also offers more flexibility and can be adapted to a wider range of architectures than a simplified model that exploits the specificity of the

hardware. Additionally, the effects of the limbs on the dynamics are more pronounced on TALOS than on other humanoid robots because it has heavier limbs in comparison.

These reasons explain why a WBMPC is chosen to carry out the deburring task. While this work provides few elements to assess the validity of these claims, it is not the core topic of the dissertation. Therefore, we consider these statements as working hypotheses, and their veracity in concrete contexts remains to be demonstrated.

The remainder of the chapter presents the theoretical foundation necessary to implement the controller on the robot.

2 ROBOT MODELLING

To fully describe a humanoid robot with n_j actuated joints and n_P rigid contacts with the environment, we define a configuration vector $\mathbf{q} \in SE(3) \times \mathbb{R}^{n_j}$. This vector is the concatenation of the free-flyer joint's placement and the n_j angular joint positions. Consequently, we can define $\dot{\mathbf{q}}$, the velocity vector of size n_v laying in the tangent space of $SE(3) \times \mathbb{R}^{n_j}$, and $\ddot{\mathbf{q}}$, the acceleration vector. We also define $\boldsymbol{\tau} \in \mathbb{R}^{n_j}$ as the joint torques.

The dynamics of the multi-body system can be described as follows [80]:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^\top \boldsymbol{\tau} + \sum_{p=i}^{n_P - 1} \mathbf{J}_i(\mathbf{q})^\top \boldsymbol{\lambda}_i \tag{4.1}$$

In Eq. (4.1), $\mathbf{M} \in \mathbb{R}^{n_v \times n_v}$ is the joint-space inertia matrix and $\mathbf{b} \in \mathbb{R}^{n_v}$ represents the generalized non-linear forces, accounting for the centrifugal, Coriolis, and gravitational terms. $\mathbf{S} \in \mathbb{R}^{n_v \times n_j}$ is the motion freedom matrix that maps the torques to the actuated part of the dynamics. It translates the fact that the command cannot act directly on the free-flyer joint. For all $i \in [\![1; n_P]\!]$, $\boldsymbol{\lambda}_i$ represents the contact force and $\mathbf{J}_i$ the contact Jacobian at contact i.

In this context, forces $\boldsymbol{\lambda}_p$ abstractly represents either 3D forces for punctual contacts or spatial 6D forces for planar contacts, expressed in their respective contact frame. They must respect the contact model described by the cone K_p:

$$\forall i \in [\![1; n_P]\!], \boldsymbol{\lambda}_i \in K_i$$

Assuming non-slippage conditions, the existence of a contact i with the environment implies the end effector position is fixed. Consequently, the end effector's velocity should be zero, and we can also constrain its acceleration to be zero:

$$\frac{\partial \mathbf{J}_i \dot{\mathbf{q}}}{\partial t} = \dot{\mathbf{J}}_i \dot{\mathbf{q}} + \mathbf{J}_i \ddot{\mathbf{q}} = 0 \tag{4.2}$$

By combining the equality established at Eq. (4.2) with Eq. (4.1), we can obtain the Karush-Kuhn-Tucker conditions of the rigid contact dynamics:

$$\begin{bmatrix} \mathbf{M} & \mathbf{J}_c^\top \\ \mathbf{J}_c & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{S}^\top \boldsymbol{\tau} - \mathbf{b} \\ -\dot{\mathbf{J}}_c \dot{\mathbf{q}} \end{bmatrix} \tag{4.3}$$

Here, $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1 ... \boldsymbol{\lambda}_P]^\top$ and $\mathbf{J}_c = [\mathbf{J}_1 ... \mathbf{J}_P]^\top$ are the concatenation of vectors of contact forces and concatenation of contact Jacobian matrices.

This formulation can be understood as an optimality condition for the problem that minimizes the deviation in acceleration between the constrained and unconstrained motion. It allows expressing contact forces in terms of robot state and actuation [37]. If, we consider the state of the robot to be $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}})$ and the control to be $\mathbf{u} = \boldsymbol{\tau}$, Eq. (4.3) leads to the following force-free partial derivative equation:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = F(\mathbf{x}, \mathbf{u}) \tag{4.4}$$

More information about the derivation of this equation, in a setting similar to the one studied in this document, can be found in [58].

3 OPTIMAL CONTROL

The control approach chosen for the robot is to iteratively solve an OCP leveraging the dynamics presented in Eq. (4.4). The idea is, at each control step, to solve the OCP from the currently measured state and execute only a fraction of the obtained control sequence before repeating this process. This strategy, known as Model Predictive Control (MPC) [81] allows the leverage a model of the system to predict its evolution on a control horizon while exploiting sensors to react to disturbances.

3.1 *General problem formulation*

The objective of optimal control is to find the control sequence $U : t \mapsto \mathbf{u}(t)$ that will take the robot from a starting state $\mathbf{x}_S$ to a goal specified by a set of constraints X_G while minimizing running and terminal costs $L(\mathbf{x}, \mathbf{u})$ and $L_{term}(\mathbf{x})$:

$$\begin{aligned} U^* = \quad & arg\min_U \int_0^T L(\mathbf{x}(t), \mathbf{u}(t))dt + L_{term}(\mathbf{x}(T)) \\ & s.t. \qquad \dot{\mathbf{x}}(t) = F(\mathbf{x}(t), \mathbf{u}(t)) \\ & \qquad \forall t \; g(\mathbf{x}(t), \mathbf{u}(t)) = 0 \\ & \qquad \forall t \; h(\mathbf{x}(t), \mathbf{u}(t)) \leq 0 \\ & \qquad \mathbf{x}(0) = x_S, \; \mathbf{x}(T) \in X_G \end{aligned} \tag{4.5}$$

$g(x(t), u(t))$ and $h(x(t), u(t))$ represent additional equality and inequality constraints. The equality constraints generally encode the task that must be carried out by the robot.

Inequality constraints are often used to represent the feasibility bounds of the system: torque limits, joint limits, self-collisions.

3.2 *Resolution approaches*

The solutions to this problem are formally described by the Hamilton-Jacobi-Bellman equations. However, directly integrating these equations is rarely feasible in practice, especially for high-dimensional states and non-smooth costs. Nonetheless, practical solutions exist to solve this problem:

— *Indirect Methods*: These methods exploit the Maximum Principle of Pontryagin to derive necessary conditions for optimality [29]. The application of these conditions usually results in a set of differential equations subject to boundary conditions. However, even if these approaches can provide highly accurate solutions, they require the first-order necessary conditions to be derived for every new problem instance, which can be cumbersome. Additionally, they do not handle state constraints well, and cases where such constraints are present require an a priori estimation of the constrained arcs of the solution.

— *Direct Methods*: These methods can deal with large systems and are more flexible and robust, though less accurate compared to indirect methods [28]. Direct methods cast the optimization into a Nonlinear Programming (NLP) problem by using a direct transcription into a finite-dimensional parameterization of variables. Once the problem is transcribed, it can be solved using fast and efficient NLP solvers such as Acados [259] and IPOPT [262].

4 ROBOT CONTROL USING CROCODDYL

To solve the continuous problem presented in Eq. 4.5 we chose a direct transcription approach. The continuous OCP will be discretized before being solved. More precisely, the discrete NLP will be solved using Contact RObot COntrol by Differential DYnamic Library (Crocoddyl) [172], an optimal control library that exploits a multiple-shooting variant of Differential Dynamic Programming (DDP).

4.1 *Discrete formulation*

In order to discretize the problem, the control interval $[0, T]$ is split into N sub-intervals. In our case, all the sub-intervals are of equal duration $dt = \frac{T}{N}$, but this is mainly for convenience and does not affect the generality of this approach. Assuming the control is constant over each sub-interval, the aim is to optimize a discrete control sequence $U = [\mathbf{u}_0 ... \mathbf{u}_{N-1}]$.

$$U^* = \underset{U}{arg\,min} \sum_{t=0}^{N-1} l(\mathbf{x}_t, \mathbf{u}_t) + l_{term}(\mathbf{x}_N) \quad s.t. \quad \mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t) \tag{4.6}$$

In this version of the problem, $f(\mathbf{x}_k, \mathbf{u}_k)$ is the discrete dynamic of the system, derived from the numerical integration of Eq. (4.4). l and l_{term} are the running and terminal cost functions.

Constraints are encoded as penalties inside the cost function.

It is worth noting that the strategy employed to solve this problem does not accept hard constraints. Consequently, there are no constraints other than the dynamics in this formulation. This also means that the discrete costs $l_k(\mathbf{x}, \mathbf{u})$ and $l_{term}(\mathbf{x})$ are not direct transcriptions of the costs $L(\mathbf{x}, \mathbf{u})$ and $L_{term}(\mathbf{x})$ defined in Eq. (4.5). Indeed, for the problem to be equivalent, these terms must also translate the equality and inequality constraints $g(\mathbf{x}, \mathbf{u})$ and $h(\mathbf{x}, \mathbf{u})$. In practice, this is often done by encoding the constraints as relaxed penalties.

4.2 *Differential Dynamic Programming*

Even after being discretized, the problem is often too challenging to be solved directly. The complexity of the dynamics coupled with that of the costs typically makes it non-linear and non-convex. This is especially true when tackling complex tasks with full-size humanoid robots. That is why we use DDP, an algorithm introduced in [176].

To solve the NLP, DDP exploits Bellman's optimality principle. This principle states that, given an optimal trajectory from an initial state to a final state, any sub-trajectory within it is also optimal for the sub-problem defined by the starting and ending states of that sub-trajectory. In practice, it offers a solution to solve the optimization of a sequence by recursively solving a sequence of optimizations.

DDP computes optimal improvements around a given control sequence.

However, exploiting Bellman's optimality principle is not sufficient, as the problem solved at each step is still non-linear and non-convex. That is why DDP leverages a quadratic approximation of the cost function and the dynamics. Instead of looking for a global solution at each step, the effect on the cost function of small variations of the control sequence is determined, enabling an improved control sequence to be chosen.

4.2.1 *Bellman equation*

In order to apply Bellman's principle, we define the cost-to-go associated with a partial control sequence $U_i = [u_i ... u_{N-1}]$, starting from a state $\mathbf{x}$ at time i ($i \in [\![0; N-1]\!]$):

$$J(\mathbf{x}, U_i) = \sum_{k=i}^{N-1} l(\mathbf{x}_k, \mathbf{u}_k) + l_{term}(\mathbf{x}_N) \tag{4.7}$$

Here, we assume that the trajectory follows the dynamic i. e. $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \forall k \in [\![i; N-1]\!]$, and $\mathbf{x}_i = \mathbf{x}$. The running and terminal costs l and l_{term} are identical to those defined by Eq. (4.6).

We refer to the optimal cost-to-go as the *value function*, which is given, at time i, by:

$$V(\mathbf{x}, i) = \min_{U_i} J(\mathbf{x}, U_i) \tag{4.8}$$

Using this notation, the Bellman equation for the discrete problem can be written as:

$$V(\mathbf{x},\ i) = \min_{\mathbf{u}}[l(\mathbf{x}, \mathbf{u}) + V(f(\mathbf{x}, \mathbf{u}),\ i+1)] \tag{4.9}$$

This formulation of the value function effectively transforms a minimization over a sequence of controls into a sequence of minimizations over a single control. The problem of finding the optimal control step $\mathbf{u}_i^*$ at time i, can thus be written as:

$$\mathbf{u}_i^* = arg\min_{\mathbf{u}}[l(\mathbf{x}, \mathbf{u}) + V(f(\mathbf{x}, \mathbf{u}),\ i+1)] \tag{4.10}$$

The value function, and the optimal control sequence, can be recursively computed starting from the terminal state by enforcing the terminal condition $V(\mathbf{x}, N) = l_{term}(\mathbf{x})$. This process is often referred to as the backward pass.

4.2.2 *Computation of the backward pass*

As mentioned previously, even after exploiting Bellman's principle, the problem remains non-linear and non-convex. That is why we study the variation of the cost-to-go caused by a small perturbation around an initial trajectory. This approach has the advantage of allowing iterative improvement of the trajectory but also means that a warm-start must be provided to the solver.

Given, at time i, a state-command pair $(\mathbf{x}, \mathbf{u})$ from a warm-start trajectory and perturbations $(\delta\mathbf{x}, \delta\mathbf{u})$, we define the variation of the cost-to-go:

$$\begin{aligned} Q(\delta\mathbf{x}, \delta\mathbf{u}) = & l(\mathbf{x}+\delta\mathbf{x}, \mathbf{u}+\delta\mathbf{u}) \quad + V(f(\mathbf{x}+\delta\mathbf{x}, \mathbf{u}+\delta\mathbf{u}), i+1) \\ & - l(\mathbf{x}, \mathbf{u}) \qquad\qquad\qquad\quad - V(f(\mathbf{x}, \mathbf{u}), i+1) \end{aligned} \tag{4.11}$$

Working on minimizing the Q-value with respect to $\delta\mathbf{u}$ allows us, assuming that the perturbations are small enough, to approximate the variation of the cost-to-go using a second-order Taylor expansion of the costs and the dynamics.

The Q-value can then be written as:

$$Q(\delta\mathbf{x}, \delta\mathbf{u}) \approx \frac{1}{2} \begin{bmatrix} 1 \\ \delta\mathbf{x} \\ \delta\mathbf{u} \end{bmatrix}^\top \begin{bmatrix} 0 & Q_\mathbf{x}^\top & Q_\mathbf{u}^\top \\ Q_\mathbf{x} & Q_{\mathbf{xx}} & Q_{\mathbf{xu}} \\ Q_\mathbf{u} & Q_{\mathbf{ux}} & Q_{\mathbf{uu}} \end{bmatrix} \begin{bmatrix} 1 \\ \delta\mathbf{x} \\ \delta\mathbf{u} \end{bmatrix} \tag{4.12}$$

Here, the subscripts denote differentiation. For readability, we drop the time index i and define the next time step using primes, i. e. $V' = V(i+1)$.

$$\begin{aligned} Q_{\mathbf{x}} &= l_{\mathbf{x}} + f_{\mathbf{x}}^{\top} V'_{\mathbf{x}} \\ Q_{\mathbf{u}} &= l_{\mathbf{u}} + f_{\mathbf{u}}^{\top} V'_{\mathbf{x}} \\ Q_{\mathbf{xx}} &= l_{\mathbf{xx}} + f_{\mathbf{x}}^{\top} V'_{\mathbf{xx}} f_{\mathbf{x}} + V'_{\mathbf{x}} \cdot f_{\mathbf{xx}} \\ Q_{\mathbf{uu}} &= l_{\mathbf{uu}} + f_{\mathbf{u}}^{\top} V'_{\mathbf{xx}} f_{\mathbf{u}} + V'_{\mathbf{x}} \cdot f_{\mathbf{uu}} \\ Q_{\mathbf{xu}} = Q_{\mathbf{ux}} &= l_{\mathbf{ux}} + f_{\mathbf{u}}^{\top} V'_{\mathbf{xx}} f_{\mathbf{x}} + V'_{\mathbf{x}} \cdot f_{\mathbf{ux}} \\ V_{\mathbf{x}} &= Q_{\mathbf{x}} - Q_{\mathbf{xu}} Q_{\mathbf{uu}}^{-1} Q_{\mathbf{u}} \\ V_{\mathbf{xx}} &= Q_{\mathbf{xx}} - Q_{\mathbf{xu}} Q_{\mathbf{uu}}^{-1} Q_{\mathbf{u}} \end{aligned} \tag{4.13}$$

$(l_{\mathbf{x}}, l_{\mathbf{u}}, l_{\mathbf{xx}}, l_{\mathbf{uu}}, l_{\mathbf{ux}})$ represent the derivatives of the cost function with respect to state and control. Similarly, $(f_{\mathbf{x}}, f_{\mathbf{u}}, f_{\mathbf{xx}}, f_{\mathbf{uu}}, f_{\mathbf{ux}})$ represent the derivatives of the dynamics with respect to the state and control variables.

By minimizing the quadratic approximation found in Eq. (4.12) with respect to $\delta\mathbf{u}$, we can deduce the optimal control variation:

$$\delta\mathbf{u}^* = arg\min_{\delta\mathbf{u}} Q(\delta\mathbf{x}, \delta\mathbf{u}) = -Q_{\mathbf{uu}}^{-1}(Q_{\mathbf{u}} + Q_{\mathbf{ux}}\delta\mathbf{x}) \tag{4.14}$$

This result is often rewritten as a linear feedback policy with a feedforward term $\mathbf{k}$ and a feedback term $\mathbf{K}$ (sometimes referred to as Riccati gains):

$$\begin{aligned} \delta\mathbf{u}^*(\delta\mathbf{x}) &= \mathbf{k} + \mathbf{K}\delta\mathbf{x} \\ with\ \mathbf{k} &= -Q_{\mathbf{uu}}^{-1} Q_{\mathbf{u}} \\ \mathbf{K} &= -Q_{\mathbf{uu}}^{-1} Q_{\mathbf{ux}} \end{aligned} \tag{4.15}$$

This policy allows us to determine a quadratic model of the value function at time i:

$$\begin{aligned} \Delta V &= -\frac{1}{2} Q_{\mathbf{u}}^{T} Q_{\mathbf{uu}}^{-1} Q_{\mathbf{u}} \\ V_{\mathbf{x}} &= Q_{\mathbf{x}} - Q_{\mathbf{xu}} Q_{\mathbf{uu}}^{-1} Q_{\mathbf{u}} \\ V_{\mathbf{xx}} &= Q_{\mathbf{xx}} - Q_{\mathbf{xu}} Q_{\mathbf{uu}}^{-1} Q_{\mathbf{u}} \end{aligned} \tag{4.16}$$

The model of the value function and the optimal control variations can be used to carry out the recursion starting from $i = N - 1$ with a condition on the final value function $V(\mathbf{x}, N) = l_{term}(\mathbf{x})$.

4.2.3 *Computation of the forward pass*

Once the backward pass is completed, the policy computed at each time step can be used to improve the state and control sequences. The improved terms are designated with a superscript star notation:

$$\begin{aligned} \mathbf{x}_0^* &= \mathbf{x}_0 \\ \mathbf{u}_i^* &= \mathbf{u}_i + \mathbf{k}_i + \mathbf{K}_i(\mathbf{x}_i^* - \mathbf{x}_i) \\ \mathbf{x}_{i+1}^* &= f(\mathbf{x}_i^*, \mathbf{u}_i^*) \end{aligned} \tag{4.17}$$

It is clear that, apart from time $i = 0$, the state from which the dynamics are rolled out will be different from the state used to compute the optimal policy during the backward pass. That is why DDP computes a policy depending on the state instead of a single optimal control variation. However, the policy is computed from a quadratic approximation and often takes large steps. Therefore, to apply this method to concrete problems, additional heuristics are often necessary. For example, Crocoddyl uses a line-search scheme to effectively scale $\mathbf{k}$ to achieve the longest step along the descent direction given by DDP.

Once the forward pass has been completed, an improved control sequence is obtained. However, since this method works locally, several successive iterations of the backward and forward passes are often necessary. In practice, these iterations are often carried out until convergence. However, in settings with limited computational resources, such as online MPC, the algorithm can be run only a limited number of times. The efficiency of this method then largely relies on the quality of the warm-start provided to the solver.

4.3 *Feasibility-driven Differential Dynamic Programming*

A point that has been omitted so far, is that in the class of Direct methods, there exists several transcription strategies [133]:

— *Single shooting*: Optimizes only the control inputs and relies on a model of the dynamics to compute the state trajectory.

— *Multiple shooting*: Splits the trajectory into smaller intervals and solves the problem using single shooting on each interval.

— *Collocation*: Simultaneously discretizes the state and control trajectories and enforces the dynamics as constraints at discrete points in the trajectory.

DDP is generally considered a single-shooting method, but the Crocoddyl library relies on an improved multiple-shooting version of this algorithm called Feasibility-driven Differential Dynamic Programming (FDDP) [172].

This method adds intermediate state points as decision variables, effectively turning the algorithm into a multiple-shooting optimization. This formulation introduces gaps in the dynamics $\bar{\mathbf{f}}_i$ (also sometimes referred to as defects), representing the difference between the rollout state and the shooting state at each time step i:

$$\bar{\mathbf{f}}_{i+1} = f(\mathbf{x}_k, \mathbf{u}_k) - \mathbf{x}_{i+1} \tag{4.18}$$

In addition, this algorithm neglects the second-order terms of the dynamics to reduce computational load. According to some practitioners, this means that this method should be classified as an Iterative Linear Quadratic Regulator (iLQR) [159], but it uses a very similar approach to DDP, which was designed earlier, hence its name.

This multiple shooting formulation mainly brings two advantages:

— First, it allows the warm-start to be infeasible. This is especially relevant when dealing with complicated systems where generating a state trajectory is possible but finding the corresponding command can be challenging.

— Then, having gaps in the dynamics during early rollouts allows leveraging the better globalization capacity of multiple shooting schemes.

5 RICCATI INTERPOLATION

The drawback of controlling a robot in torque is that it typically needs to be done at a higher frequency than position control. The controller used on the TALOS runs at 2 kHz. However, despite extensive work carried out to speed up the computation of the derivatives [43] and technological advancements in CPU power, the controller cannot run at more than 100 Hz while taking into account the full dynamics of the robot.

That is why we rely on a linear approximation of the policy to interpolate the control at the desired frequency. [62] shows that the feedback term of the policy $\mathbf{K}$ can be interpreted as the sensitivity of the optimal policy with respect to the initial state:

$$\mathbf{K}_0 = \left.\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\right|_{\mathbf{x}_0} \tag{4.19}$$

In practice, the policy used on the robot is the following:

$$\mathbf{u} = \mathbf{u}_0^* + \mathbf{K}_0(\mathbf{x}_{meas} - \mathbf{x}_0^*) \tag{4.20}$$

where $\mathbf{u}_0^* = \mathbf{k}_0$ and $\mathbf{K}$ are results of DDP computation, and $\mathbf{x}_0^*$ is the initial state of the trajectory, corresponding to the state measured at the moment the solver is called. These values are updated every time the solver provides a new result, i. e. at roughly 100 Hz. $\mathbf{x}_{meas}$ corresponds to the current measured state and is updated much more frequently, at 2 kHz in our case.

Riccati interpolation is necessary to reach a sufficient control frequency.

This method has been successfully employed to send torques command at 2 kHz to carry out dynamic movements on a humanoid robot [62].

6 INTERFACING WITH THE ROBOT

In addition to the Riccati interpolation method, there are additional control layers provided by the manufacturer to perform torque tracking. These layers take into account the dynamics of the actuators, ensuring that the commanded torques are accurately exe-

cuted. Unfortunately, these control layers are not open source, and further details about their implementation cannot be disclosed.

5 DEBURRING EXPERIMENTS

In Short

This chapter details the practical design choices made to carry out deburring experiments on the TALOS humanoid. Those experiments were presented in the scope of the Memmo project in June 2022.

Contents

1 INTRODUCTION

This chapter dwells upon the practical design choices and methodologies employed to conduct deburring experiments on the TALOS humanoid robot. The chapter is structured into three main sections, each addressing a critical aspect of the experimental process.

First, Section 2 explores the technical intricacies involved in setting up and executing the deburring experiments. This includes an explanation of how transitions between holes are managed. Additionally, this section discusses the integration of exteroceptive feedback, which is crucial for the robot's ability to adapt to its environment and perform tasks with precision.

Next, Section 3 presents the initial findings obtained from the deburring experiments. These results provide valuable insights into the performance and capabilities of the TALOS humanoid in real-world scenarios. The analysis of these results helps understand-

ing the strengths and limitations of the current setup and paves the way for further improvements.

Finally, Section 4 summarizes the challenges encountered while working with the TALOS platform. By addressing these challenges, the section aims to provide a comprehensive understanding of the complexities involved in conducting such experiments.

2 IMPLEMENTATION

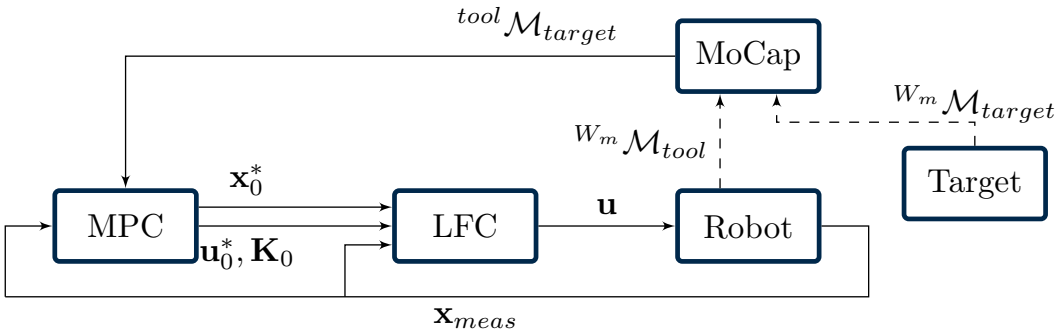


Figure 5.1 – Control structure for the deburring experiment. The Robot node encompasses the hardware and the low-level controllers mentioned in Section 6 of the previous chapter.

2.1 *Control pipeline structure*

The different elements of the control pipeline are integrated inside a Robot Operating System (ROS) architecture. The key components of this architecture, represented in Fig. 5.1, are the following:

— *Linear Feedback Controller (LFC)* [1]: Controller running directly on the robot and handling the interpolation presented in Section 5 of Chapter 4. It outputs torques commands that are transmitted to the custom controller of the robot.

— *MPC*: Node running on a separate computer which solves the OCP using the Crocoddyl library. To allow more interoperability, this node communicates with the robot using predefined ROS messages [2]. That way the implementation of the LFC node is independent of that of the MPC.

— *Motion Caputre (MoCap)*: Node used to detect the position of the target and that of the tool in order to provide real-time measurements.

1. https://github.com/loco-3d/linear-feedback-controller
2. https://github.com/loco-3d/linear-feedback-controller-msgs

2.2 *Model Predictive Control*

The MPC node is responsible for iteratively solving the OCP which provides torque commands $\mathbf{u}_0^*$, as well as the feedback gain $\mathbf{K}_0$ and the state reference $\mathbf{x}_0^*$ for the Linear Feedback Controller.

A major challenge of MPC is designing the right cost function.

The main design choice for the MPC is the structure of the cost function that is optimized by the solver at every step. Another important aspect is the strategy chosen to make the cost function evolve in time thus adapting the behavior of the MPC to the situation.

2.2.1 *Cost function*

The cost function is a scalar function designed to encode the behavior of the robot, which amounts to simultaneously satisfying a variety of goals. In our case, these goals are:

— Bringing the end effector of the robot to a desired position and orientation.

— Preserving the equilibrium of the robot.

— Preventing the robot from exceeding its limits.

The running cost is typically constructed as a sum of $n_c \in \mathbb{N}$ sub-costs:

$$l(\mathbf{x}, \mathbf{u}) = \sum_{k=1}^{n_c} w_k a_k(\mathbf{r}_k(\mathbf{x}, \mathbf{u})) \tag{5.1}$$

$\forall k \in [\![1; n_c]\!]$, $\mathbf{r}_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_j} \to \mathbb{R}^{n_k}$ is a residual model, i. e. a vector function encoding a specific objective (such as the placement of a frame of the robot or the position of the CoM). The size n_k depends on the characteristics of the cost. $a_k : \mathbb{R}^{n_k} \to \mathbb{R}$ is an activation function which translates a residual into a scalar value. w_k is a weight chosen to adjust the relative importance of each cost.

In our approach, the derivative of the cost function with respect to state and command must be computed analytically. Splitting the cost as presented in Eq. (5.1) makes computation easier and allows for the reuse of the same costs for different tasks, thus saving engineering time.

The terminal cost has a similar formulation but depends only on $\mathbf{x}$ and may have a different number of sub-costs $n_{c\_term} \in \mathbb{N}$:

$$l_{term}(\mathbf{x}) = \sum_{k=1}^{n_{c\_term}} w_k a_k(\mathbf{r}_k(\mathbf{x})) \tag{5.2}$$

The difference between the running and terminal cost is that the running cost should encode the task, while the terminal cost should only guarantee that the robot ends up in a controllable state at the end of the horizon. By controllable, we mean a state in which there exists a constraint-satisfying control that can stabilize the robot. In theory, this ensemble is very broad, but it is difficult to characterize in practice. That is why we

choose the same structure for both the terminal and running cost; we simply remove any control-dependent term from the terminal cost.

We choose a general structure for the cost function to achieve all the desired goals. We reuse the architecture presented in [62], which implements a similar task on the same hardware. This architecture includes the following components:

— A constraint cost on the position of the joints
— A cost on the position of the CoM to maintain equilibrium
— A regularization cost on the state of the robot
— A regularization cost on the control
— A goal cost related to the position of the end-effector
— A goal cost related to the orientation of the end-effector
— A goal cost on the speed of the end-effector

CONSTRAINT COST

The most important component of the cost function, and thus the one with the highest weight, is related to constraints. Since the solution we use does not accept hard constraints, they are encoded as penalizations. Constraints aim to prevent the solver from providing a trajectory command that could damage the robot, although in practice this risk is limited by the fact that there are safeties in the lower level of the control. Mechanical limits are modeled as position, velocity, and effort limits at every joint. The values of these limits are typically provided in the Unified Robot Description Format (URDF) of the robot, which in our case was supplied by the manufacturer.

However, during the first set of experiments, only the position of the joints was limited at the level of the OCP (the low-level controller embedded on the robot always has all the limits activated). This was not limiting since the first movements tested on the robot were slow and with no payload.

Another important aspect is that the low-level control had additional limits, the formulation of which is not open source. It means that they could not be implemented in the OCP which lead to manually tuning the position limits to achieve successful movement on the robot. The position limits that had to be tweaked the most were the one of the torso.

This cost is formed from a residual which is the posture of the robot $\mathbf{q}$ and a quadratic barrier activation. The quadratic barrier, represented in Fig. 5.2, is null if the residual is within the fixed bounds and follows a quadratic evolution out of the bounds. The full expression of this cost is:

$$l_{cons}(\mathbf{q}) = \|max(\mathbf{q} - \mathbf{q}_u, 0) + min(\mathbf{q} - \mathbf{q}_l, 0)\|^2 \tag{5.3}$$

With $\mathbf{q}_u$ and $\mathbf{q}_l$ being respectively the upper and lower bounds of the admissible joint positions.

EQUILIBRIUM COST

When working on the whole-body control of a humanoid robot, balance is always a major concern. It is crucial to prevent the robot from falling over both to preserve its integrity and to be able to efficiently carry out tasks. In our case, the movements are slow, and the robot does not move its feet, so a command on the position of the CoM is sufficient. However, more advanced costs would need to be employed for locomotion or loco-manipulation tasks [60].

The equilibrium cost is formulated as follows:

$$l_{cons}(\mathbf{x}) = \|\mathbf{c}(\mathbf{x}) - \mathbf{c}_d\|^2 \tag{5.4}$$

With $\mathbf{c}(\mathbf{x})$ and $\mathbf{c}_d$ the current and desired Center of Mass of the robot.

REGULARIZATION COSTS

Regularization is important to prevent drifting of the unconstrained parts of the robot when several solutions might allow solving a given task (for example, prevent movement of the right arm when manipulating with the left arm). Additionally, it is useful to facilitate the convergence and numerical stability of DDP. We regularize the state, and the control.

The regularization cost is formulated as follows:

$$l_{reg}(\mathbf{x}, \mathbf{u}) = (\mathbf{x} - \mathbf{x}_d)^T \mathbf{R}_\mathbf{x} (\mathbf{x} - \mathbf{x}_d) + (\mathbf{u} - \mathbf{u}_d)^T \mathbf{R}_\mathbf{u} (\mathbf{u} - \mathbf{u}_d) \tag{5.5}$$

This cost prioritizes behaviors that are close to the desired state $\mathbf{x}_d$, built from the initial robot posture with zero velocities. It also penalizes controls that are far from the torques $\mathbf{u}_d$ required to counteract the force of gravity in the desired position. For both of these costs, a weighted quadratic activation is chosen (see Fig. 5.2), $\mathbf{R}_\mathbf{x}$ and $\mathbf{R}_\mathbf{u}$ are positive definite matrices used to tune the relative impact of each joint on the cost.

GOAL RELATED COSTS

The last component of the cost function is related to the task being carried out by the robot. For the deburring operation, there are three goals:

— *Good position*: The position of the end-effector $\mathbf{p}$ must be close to the position of the hole $\mathbf{p}_d$. This is the most important objective, and the tolerance on the error is low given that the diameter of the hole in which the insertion is done is $d_{hole} = 1$ cm.

— *Good orientation*: This goal is necessary to ensure that the orientation of the end-effector R matches the desired orientation R_d (R and R_d are defined as elements of $SO(3)$). This goal is necessary to force the robot to have the tool perpendicular to the aircraft part's surface during insertion.

— *Good speed*: The Cartesian speed of the tool $\mathbf{v}$ should not be too high for the insertion to be safe. This goal has a relatively lower weight than the other two goals in order not to impede performance.

The cost is thus designed as follows:

$$l_{goal} = log(1 + \frac{\|\mathbf{p} - \mathbf{p}_d\|}{\alpha}) + \|R - R_d\|^2 + \|\mathbf{v}\|^2 \tag{5.6}$$

The activation of the position cost is referred to as *Quad Flat Log*, and its shape can be seen in Fig. 5.2. The idea is to have a bigger slope near the objective to encourage the robot to move closer to the target while not risking destabilization when the tool is really far from the target. The sensitivity of the behavior can be tuned with the parameter α, in our case $\alpha = 0.02$.

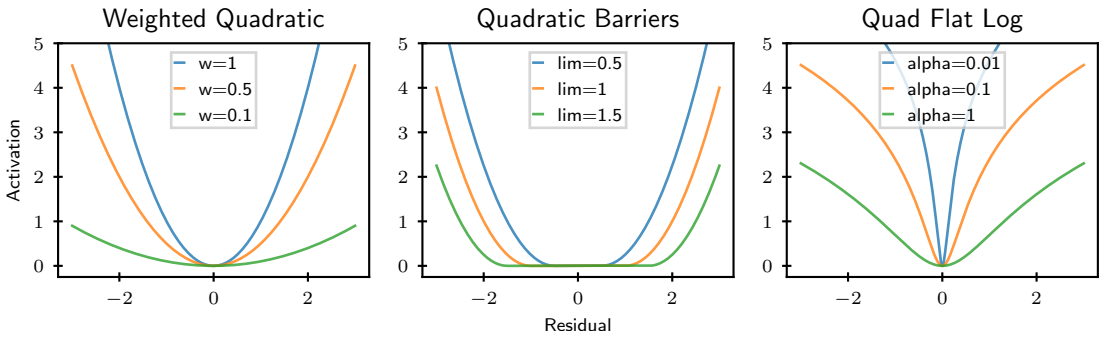


Figure 5.2 – Comparison of the different activation functions that are used to build the cost.

2.2.2 *Transition between tasks*

The structure of the cost function is not the only crucial design point for achieving a successful deburring sequence. A strategy to handle a succession of holes also needs to be devised.

Using a receding horizon strategy, only one new node needs to be provided at every control step.

We are using a receding horizon strategy, which means that the horizon of the MPC is sliding along a trajectory representing the full movement, as can be seen in Fig. 5.3. At every control step, the first node of the horizon is discarded, all other nodes are shifted by one index, and a new one is created at the end of the horizon. This has the advantage of maintaining coherence between successive resolutions of the OCP because the problem is mostly identical from one step to another. That is why we can reuse the previously computed solution as a warm-start for each iteration, which greatly limits the number of DDP passes that need to be carried out to reach a satisfying solution.

In practice, to change the desired target, we just need to update the desired position of the end-effector $\mathbf{p}_d$ when creating a new node. The approach is similar to the one proposed in [143].

To achieve the full deburring sequence, the robot will have to sequentially reach a list of positions before executing a one-axis translation. That is why we use intermediate points situated in the alignment of the hole but at a safe distance from the structure before each insertion.

In practice, the full sequence is implemented as a finite state machine with additional safety checks. For example, if the tool is not aligned with the hole, the insertion is not carried out and the robot goes back to its initial position.

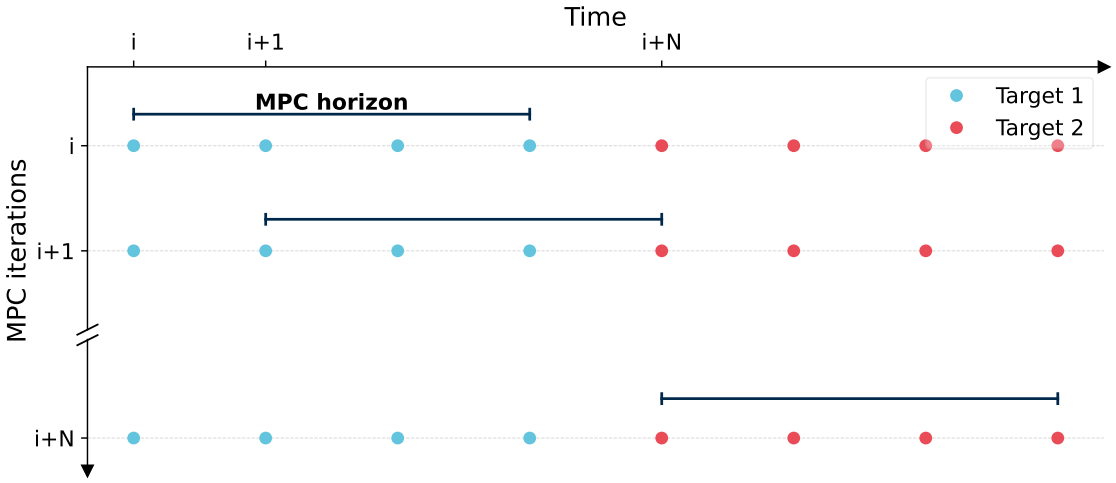


Figure 5.3 – Simplified representation of the receding horizon strategy when transitioning from target 1 to target 2. At every time step, the MPC horizon in shifting along the full trajectory by one node. The transition is complete after N steps, where N is the number of node in the horizon ($N = 4$ in this illustration).

2.3 *Integration of sensory feedback*

To add more flexibility to the experimental setup, the integration of an exteroceptive sensor was necessary. Indeed, conducting the experiments without sensory feedback would require a very precise initial placement of the robot with respect to the aircraft piece, which is not practical.

However, perception is not the main focus of this work, so the problem of detecting and locating the piece in a real setup is not treated in this section. Instead, we focus on the practical solution that we used.

For localization, we relied on a Qualisys Motion Caputre (MoCap) system.

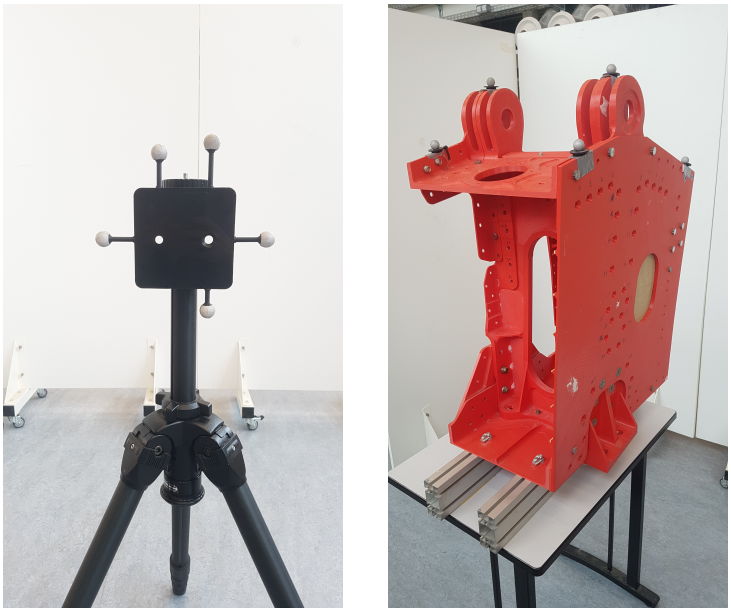
2.3.1 *Experimental setup*

MoCap systems function by detecting reflective markers. Placing several markers in a given configuration allows the system to detect the position and orientation of objects.

The objective of the setup is to be able to detect the target and place it back in the reference frame of the robot. It means that markers should be placed on the target, but also on the robot. Indeed, the reference frame of the MoCap doesn't match that of the robot, so we need two measures to assess the relative position of each element.

TARGET

Two targets were successively used for the experiments (Fig. 5.4).



(a) First target (b) Second target

Figure 5.4 – Pictures of the target used to test deburring movement in lab.

The first one was a 3D printed plate with holes matching the size of those found on the aircraft pylon. The plate was attached to a pole, which allowed us to easily test various deburring configurations. The MoCap markers were rigidly attached to the side of the plate. Since its dimensions were precisely known, locating the holes with respect to the markers was straightforward.

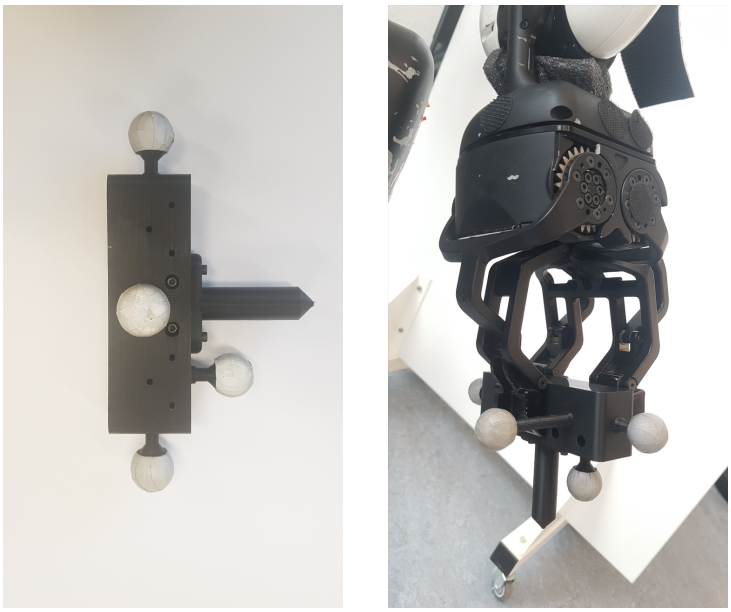
The second target was the 3D printed mockup of the aircraft pylon presented in Chapter 2. MoCap markers were placed on top of the structure. The position of the holes with respect to those markers was measured using the MoCap system itself by placing additional markers in the holes, and removing them for the experiments.

ROBOT

Two solutions were envisioned to link the MoCap to the robot:

— The first solution was to place a set of markers on the waist of the robot and carry out a calibration [191] to identify the placement of the markers in the reference frame of the robot. Once this value was identified, expressing MoCap measures in the reference frame of the robot was straightforward. Another set of markers would be placed on the tool.

— Another solution was to rigidly attach the tool to the end-effector of the robot and place markers on the tool, as seen in Fig. 5.5. The transformation from the frame of the tool to the frame of the fingertip can be computed because the shapes are known, and the tool is fixed to the hand.



(a) Tool fitted with the MoCap markers.

(b) Tool screwed to the fingertips of Talos.

Figure 5.5 – Pictures of the tool used for the deburring tests.

Contrary to the second method, the first one has the advantage of not relying on the internal state estimation of the robot, which can make the system robust to errors in the perceived placement of the tool. However, this solution relies on the precise placement of the waist markers and the quality of the calibration, adding complexity to the setup.

That is why we chose the second method, even if errors can arise from any estimation uncertainties in the forward kinematics of the robot. In practice, it appeared that this was not limiting for our use case. This can be explained by the fact that the flexibilities, which are a major source of uncertainty, are more pronounced if the arm is straight, which increases the lever effect. It is not problematic when the robot is in its initial posture.

2.3.2 *Integration in the MPC*

The MoCap can compute the transformation from the tool to the target, and the position of the target can then be expressed in the reference frame of the robot using the internal estimation of the robot.

$$\begin{aligned} {}^{O}\mathcal{M}_{target} &= {}^{O}\mathcal{M}_{tool}(\mathbf{x})\ {}^{tool}\mathcal{M}_{target} \\ &= {}^{O}\mathcal{M}_{tool}(\mathbf{x})\ ({}^{W_m}\mathcal{M}_{tool})^{-1}\ {}^{W_m}\mathcal{M}_{target} \end{aligned} \tag{5.7}$$

With ${}^{O}\mathcal{M}_{target} \in SE(3)$ being the placement of the target in the reference frame of the robot, which is the value that is fed to the MPC. ${}^{O}\mathcal{M}_{tool}(\mathbf{x}) \in SE(3)$ is the placement of the tool in the reference frame of the robot, which, given a robot configuration, can be computed using the forward kinematics of the robot. ${}^{tool}\mathcal{M}_{target} \in SE(3)$ is the

transformation from the tool to the target, which can be computed by the MoCap using ${}^{W_m}\mathcal{M}_{tool}$ and ${}^{W_m}\mathcal{M}_{target}$, respectively the placement of the tool and the placement of the target in the MoCap world.

Precision of the proprioceptive sensors was not the limiting performance factor.

For the experiment, we used the MoCap only during initialization to set the target appropriately with respect to the robot. We carried out tests where the position of the target was updated online using the new measured value, but this did not improve the precision while adding complexity to the control loop

3 PRELIMINARY RESULTS

3.1 *Protocol*

There are several preliminary steps before deploying a new movement on the robot.

First, the cost function is tuned in the PyBullet simulator [56]. This allows for quick iteration when testing new cost function terms. Indeed, designing the appropriate cost function for a given task requires extensive effort, and the ability to quickly iterate over new designs is crucial for achieving movements on the robot in a reasonable time. In this case, the controller exploits python bindings of a C++ code, but the communication between the simulator and the controller is not done using ROS.

A second step is to test the same movement inside Pal's private simulator. This simulator is more representative than PyBullet and uses the same ROS interface as the robot. It allows for the use of strictly the same code as what will be run on the robot, which is a significant advantage for debugging purposes. This simulator is not open-source, so it is not possible to know the exact modeling elements it contains. However, we know that it simulates actuator dynamics and measurement noise. It also includes the additional safety limits implemented on the robot, which are also closed-source. Due to the differences existing with the first simulator, a new tuning of the cost function is often necessary.

The final step is to run the movement on the robot. Despite running tests on a custom-made simulator, additional tuning is also necessary directly on the real robot, especially to obtain a movement that does not trigger the manufacturer's safety mechanisms.

3.2 *Results*

During our experiments, we successfully obtained a stable reaching movement on the humanoid robot Talos. Thanks to the torque control, the robot could be slightly perturbed without outputting dangerous amounts of energy or becoming unstable. This stability is a crucial aspect for ensuring safe operation in dynamic environments.

However, the precision achieved was not satisfactory. The placement error was above 1 cm, as can be seen in Fig. 5.6. It is significantly higher than our objective to be under 5 mm.

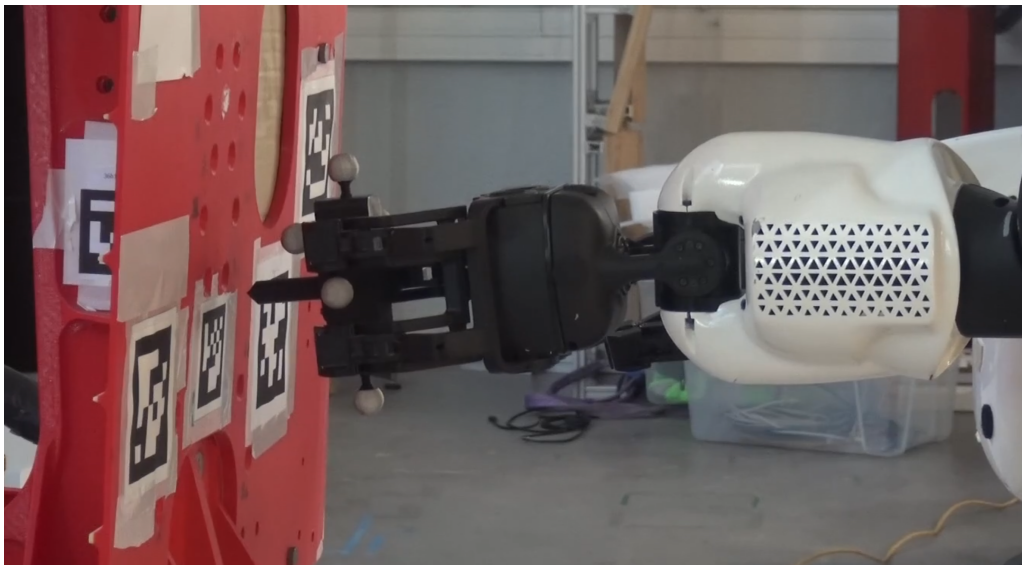


Figure 5.6 – Picture of the robot at the end of the baseline movement. It is clear that the tool is not inserted in the hole.

3.3 *Gain Scheduling*

It became apparent that, despite spending a reasonable amount of time tuning the cost function, it was not possible to find a fixed set of parameters that would lead to a sufficiently precise movement.

Increasing the weight associated to the end-effector's placement task could drive the system closer to the target but did not allow for a stable movement when the target was far away. To address this issue, we resorted to adopting a time-based variable weight for this task.

The idea was to have a first part of the movement with the initially tested weight. Then, once the end-effector stabilized near the target, to steadily increase the weight until the desired precision was reached. For safety reasons, a threshold on the maximum weight was set.

This solution allowed us to improve the precision of the movement and successfully carry out the insertion.

We can see in Fig. 5.7 an experiment on 4 holes of the structure. We can see that for the first target, the error is above the threshold, but since the weight is already at its maximum, it is not further increased. We can also see that for the last two targets, a smaller increase in weight leads to a good precision.

However, this solution is not fully satisfactory. Indeed, this approach led to a stable and precise but slow movement. The achieved insertion time, more than 10 s was significantly larger than that of a human operator and far from exploiting the full capabilities offered by the hardware.

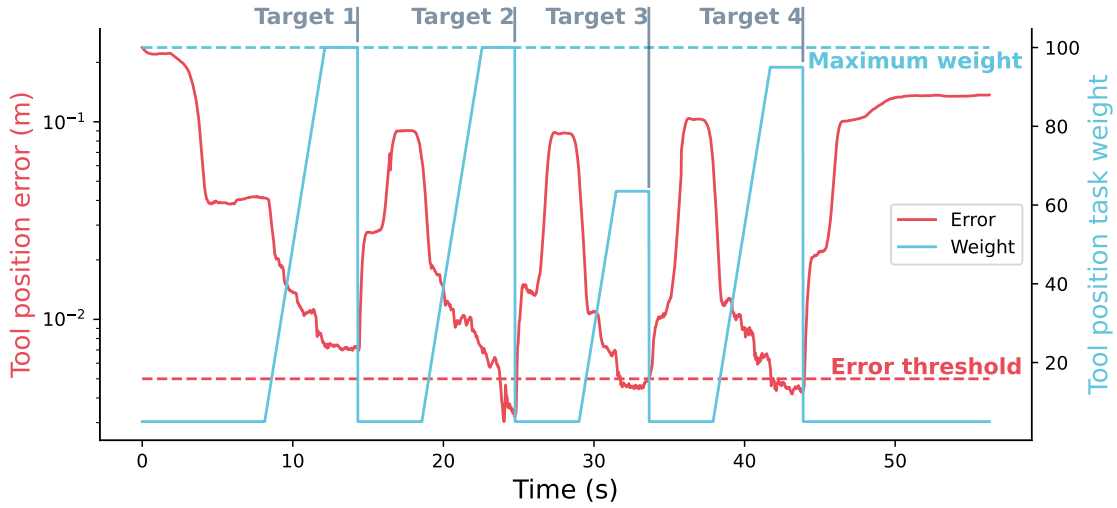


Figure 5.7 – Evolution of the measured error and the weight of the position task with respect to time. The insertion is successful (the error is below the 5 mm threshold) for 3 holes out of 4.

This underwhelming operating speed can be explained by the necessary caution that this solution entailed. Indeed, changing the relative weights of the terms in the cost function can have dangerous effects on the stability of the system. Increasing the weight too much could lead the controller to compromise on other objectives, such as equilibrium, in favor of the insertion task. This is especially true in our case, where the weight was increased by more than a factor of 10 to achieve the desired precision.

In addition to that, a time-based gain scheduling degrades the flexibility of the system, and the lack of reactivity can cause issues if unforeseen situations arise.

These reasons explained why we wanted to further study this problem in the hope of providing a reactive cost-shaping strategy to control the robot more efficiently.

4 EXPERIMENTAL HURDLES

4.1 *Hardware limitations*

Beyond the experimental results, the tests carried out on the robot highlighted some difficulties that arise when working on a complex robotic platform such as a humanoid. These challenges need to be identified and explained, as they underscore why the design of a new demonstration should not rely too heavily on extensive testing directly on the platform. The test procedure should be structured and thought out to extract as much information as possible from any test performed on the robot.

A first significant hurdle is that, because of its dimensions, any test requires at least two persons to attend the robot. This requirement induces a significant labor cost, as well as organizational difficulties.

Additionally, issues with the low-level controller of the robot were encountered. Vibrations could be heard when controlling the wrists in torque, which were linked to the tuning of the low-level controller. However, since the controller is not open source, only people working for the constructor of the robot could alter these elements. This increases the dependency of the researcher to the intervention of a company, as well as making debugging process more complex.

Finally, as mentioned in the introduction (Chapter 2, Section 3), Pyrène is the first prototype of the Talos robot. After several years of extensive use, signs of wear and tear have begun to appear, especially in the electronics. Both wrists are not functioning due to issues with the electronic board. In addition to this, the robot randomly raises errors related to communication between components. Although technically not hard to fix, these issues are time-consuming and prevent intensive use of the hardware.

Problems with the wrists prevented further experiments from being carried out.

However, solutions to these issues, such as making the hardware more reliable or increasing the human resources on the robot, are hardly applicable in practice and beyond the scope of this thesis.

4.2 *Focus on simulation*

Despite hardware limitations preventing us from undertaking all the desired experiments, I still managed to discover fundamental limitations in our control structure. Indeed, I demonstrated that an ill-designed cost function could not yield satisfactory results. Adding the MoCap inside the control loop also highlighted that the error caused by the use of proprioceptive sensors during the movement was not the limiting factor.

These design limitations can be studied in simulation, and since the robotic platform could not be used to carry out further experimentation, I focus in the remainder of this manuscript on improving the simulated behavior of the robot. However, to benefit as much as possible from the gained experience and make going back to the robot easier, I keep the control architecture as close as possible to what ran during the experiments. That is why I do not adopt alternative control strategies to replace MPC or completely change the structure of the cost. I even decide to keep the low-level Riccati interpolation (Chapter 4, Section 5), even if recent work [245] has demonstrated its weaknesses.

Therefore, the following parts mainly tackle simulation results, but with an approach that will make producing movements on the robot easier because the core of the control structure has already been thoroughly tested.

5 CONCLUSION

These results demonstrated a technical solution to carry out the desired task but are not entirely satisfactory from a theoretical viewpoint.

They highlighted the fact that it is very challenging to find an appropriate cost function to carry out a task using WBMPC. This difficulty underscores the need for further exploration into more adaptive and reactive approaches.

Therefore, in the next part of the manuscript, we delve into reactive cost shaping as a potential solution to address these challenges.

Part IV

TOWARD REACTIVE PLANNING

This part highlights the main contribution of this dissertation. I propose a reactive cost function planner to be used in conjunction with the previously presented control structure.

Chapter 6 presents in more detail how a reactive cost function could be leveraged to improve performance. This work was published at the 2023 *International Conference on Intelligent Robots and Systems*.

As a follow-up, Chapter 7 presents a Reinforcement Learning based reactive posture controller. This part was submitted to *Transactions on Automation Science and Engineering*.

6

VARIABLE COST MPC

In Short

This chapter seeks to assess how a variable cost function could benefit a WBMPC. In particular, it studies how information about the robot's posture, collected during an experiment, can be leveraged to improve performance for subsequent runs.
This chapter has been published at *IROS* in 2023 [203]. Sections 1.1, 2 and 3.1 summarize elements presented in the previous chapters. They have been included to maintain the coherence of the chapter.

Contents

1 INTRODUCTION

1.1 *Context presentation*

ROBOTS are nowadays a standard tool in large-scale manufacturing [103]. They excel at performing repetitive tasks in very well-known environments. However, they are yet to reach a huge part of the wide variety of industrial work that exists in our society.

One of the major drawbacks most industrial robots suffer from is lack of mobility. Their design does not allow them to be a relevant solution for many low volume, high added value productions such as the one found in aeronautic manufacturing. According to [139], humanoids are a promising direction to overcome this weakness. However, the resort to humanoid robots induces a higher control complexity which is further heightened when dealing with variability in the environment.

In recent years, Reinforcement Learning (RL) has been successfully used to generate highly dynamical motions on quadruped robots, such as ANYMAL [219], as well as bipedal torque controlled walking robots such as CASSIE [71]. Still, in [94], a comparison with Model Predictive Control (MPC) shows that the latter has a higher rate of success in constrained environments. Despite both approaches being different, the definition of the cost function remains a central point for both RL and MPC. The increasing complexity of the system makes it difficult to properly design such a cost-function. The aim of this chapter is, first, to experimentally find an initial feasible solution for a real situation. Then, design a simple strategy to modify the cost function in order to improve performances.

A widely used motion generation framework for humanoid robot is built upon a Model Predictive Controller for the centroidal dynamics in conjunction with an instantaneous QP-controller for the whole body [39]. A planner provides the reference trajectories to follow for tasks such as gaze direction, end-effectors placement and the overall direction of the robot.

If position control has been quite successful in generating a wide variety of robot behaviors, its capacity to react to external forces is limited to the end-effector, where a force sensor is typically incorporated for that purpose. In [74], [139] torque controlled robots appear as a potential solution for managing interactions with the environment as well as ensuring safe and compliant behavior in unplanned situations.

These situations can arise because of unforeseen events such as changes in the context, or human interaction. It opens up the way for more flexible use of robots than what was achieved with existing position control methods. Recent robots such as Digit [114] are using torque control and demonstrate impressive locomotion performances and robustness. It comes however at the cost of a more complex control architecture on robots with wave generators, and a lack of precision for positioning tasks.

Precision is nonetheless of great importance when executing industrial tasks such as deburring. A simple way to handle this issue is, assuming you can measure it, to apply a strong feedback on the error between the desired and perceived position of the tool. But, on a torque controlled robot, this might lead to a diverging command [213]. [72] has developed a passivity framework which is taking into account the energy of the system to maintain its stability. It was successfully tested on the TORO humanoid robot [139]. This approach assumes that either the desired position or trajectory of the end-effector is given. The passivity approach avoids injecting unsafe amounts of energy in the system if the environment differs too much from what was planned. The whole body instantaneous controller is in charge of absorbing model discrepancies and planner assumptions.

The success and the efficiency of the classical approach lie in the capabilities of the motion planner to generate a desired trajectory that is compatible with the whole body instantaneous controller. It can be done for instance by using a hybrid control approach and planning over a graph of motion primitives for quadrupeds [254]. It can also be done using A* through a discrete set of actions predefined according to the targeted tasks, see [95] for a locomotion example. In order to cope with the complexity of the problems most planners are using heuristics [95], or reason on low dimensional necessary conditions [250].

Still, no matter how advanced the planning heuristic is, it cannot entirely address the fundamental limitation of instantaneous whole body control. This limitation resides in the inability of this technique to account for whole-body related constraints within the MPC horizon. It means that potential conflicts with the constraints can only be detected by the whole-body controller when it is too late. For this reason, [59] proposed a whole-body model predictive control with state feedback at 100 Hz. It can perform trajectory optimization and provides reference torques to a low-level torque loop running at 2 kHz. An extension of this technique was introduced in [62], where the feedback gains of the DDP are directly sent to the low-level controller, improving meaningfully the quality of the generated motion. This approach has several advantages. A significant one is to include all the dynamical effects of the limbs on the balance criteria. This is particularly important with a robot having arms and legs that way more than 10 kg and 15 kg respectively, such as TALOS.

1.2 *Contributions*

WBMPC has successfully been applied to the humanoid robot TALOS to carry out an industrial deburring task similar to the one presented in [177]. As seen in Fig. 6.1, the objective of the task is to insert a 3d printed tool inside the holes of a mockup aircraft part. It simulates deburring, an operation that needs to be undertaken after drilling holes to clean up any material residues.

Experiments were conducted in a lab as well as on an Airbus site. The results that were obtained validate the relevance of using WBMPC for humanoid robots in an industrial context.

This chapter also tackles the issue of cost shaping. It is a challenging aspect for optimal control based approaches that needs to be resolved in order to unlock the full performances of the system. [57] uses a multi-objective optimization in conjunction with Bayesian optimization to find a suitable set of parameters. Because we expect optimal approaches to be too computationally intensive in our case, we choose to use a fixed cost function structure to experimentally explore the environment.

The contributions presented in this chapter are twofold:

— Demonstration of the experimental use of a humanoid robot in an industrial setting;

— Use of advanced cost shaping solutions to enable better performances in this context.

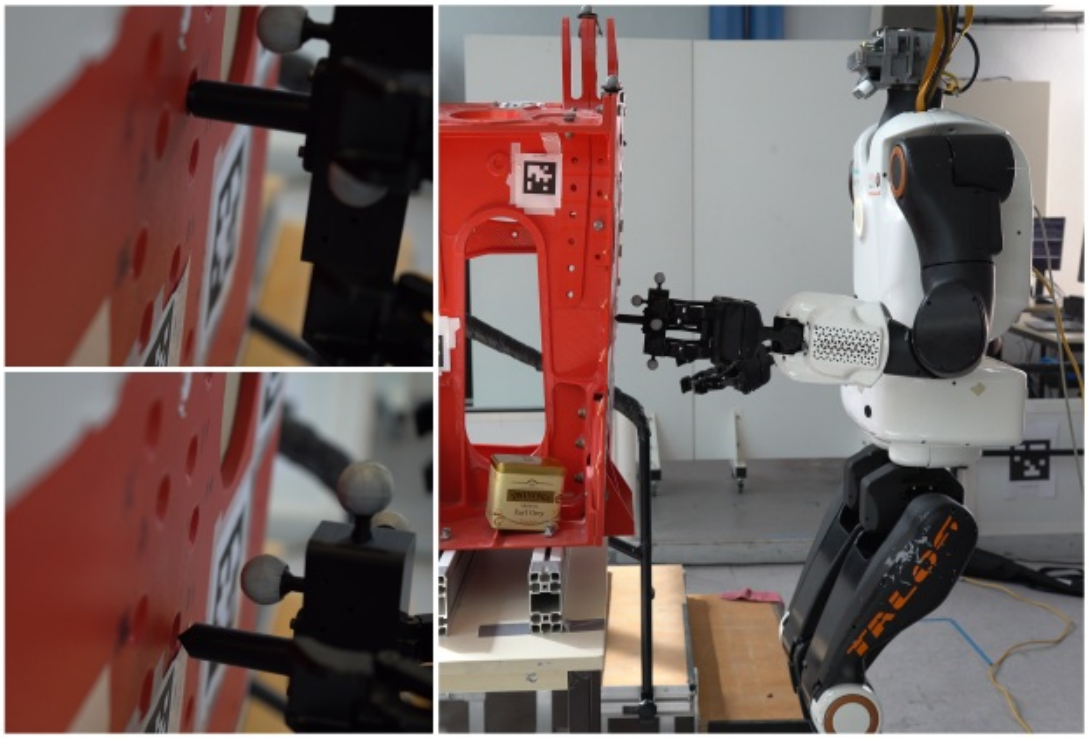


Figure 6.1 – Deburring task, high precision for a fine insertion into a hole using whole-body MPC on a torque controlled robot.

The adopted control architecture is first presented in section 2 before going into more details about the structure of the cost function that was considered in section 3. To finish, the most significant results are exposed in section 4.

2 WHOLE-BODY MODEL PREDICTIVE CONTROL

This section recalls key elements from Chapter 4 to ensure that the current chapter can be understood independently. Readers who have already familiarized themselves with the previous chapter may proceed directly to Section 3.

2.1 *Robot Modelling*

The robot configuration $\mathbf{q} \in SE(3) \times \mathbb{R}^{n_j}$ defines the global position, orientation and posture that a mobile robot has at one given moment. Such configuration evolves under the action of internal and external forces as described by the rigid-body dynamics [267]:

$$\begin{bmatrix} \mathbf{M} & \mathbf{J}_c^\top \\ \mathbf{J}_c & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{S}^\top \boldsymbol{\tau} - \mathbf{b} \\ -\dot{\mathbf{J}}_c \dot{\mathbf{q}} \end{bmatrix}, \tag{6.1}$$

where $\mathbf{M}$ is the inertia matrix, $\mathbf{b}$ stands for Coriolis, centrifugal and gravity forces, joint-motor torques $\boldsymbol{\tau} \in \mathbb{R}^{n_j}$ affect only the n_j actuated joint as indicated by the selection matrix $\mathbf{S} \in \mathbb{R}^{(n_j+6)\times n_j}$, and all contact wrenches $\boldsymbol{\lambda}_i \in \mathbb{R}^6$ are contained in $\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_1 & \cdots & \boldsymbol{\lambda}_i & \cdots \end{bmatrix}$ with the application points described respectively by the Jacobians $\mathbf{J}_i \in \mathbb{R}^{6\times n+6}$ contained in $\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & \cdots & \mathbf{J}_i & \cdots \end{bmatrix}$. The second line of Eq. (6.1) constraints the robot parts under contact to stay motion less during the contact.

Based on this dynamics, the robot configuration q and its time derivative $\dot{\mathbf{q}}$, which are the state $\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}})$, are controlled by inputting desired torques τ on joint motors during some discretization period dt to obtain the next state

$$\mathbf{x}^+ = f(\mathbf{x}, \boldsymbol{\tau}), \tag{6.2}$$

which is predicted from numerical integration of Eq. (6.1).

2.2 *Optimal Control*

For a given initial state $\mathbf{x}_0$, an optimal control sequence $U^* \triangleq \{\boldsymbol{\tau}_0, \boldsymbol{\tau}_1, \cdots, \boldsymbol{\tau}_{N\text{-}1}\}$ is generated according to Eq. (6.2), along a horizon of N time-steps in the future by minimizing the cost function

$$V(\mathbf{x}_0) = \sum_{i=0}^{N-1} l(\mathbf{x}_i, \boldsymbol{\tau}_i) + l_{term}(\mathbf{x}_N), \tag{6.3}$$

that is designed to encode the desired robot behavior with a running cost $l(\cdot,\cdot)$ for each time-step, and a terminal cost $l_{term}(\cdot)$ guiding the robot to end into some safe set of states. This desired behavior is discussed more precisely in Section 3.

The resulting optimal pair of control sequence and robot motion (U^*, X^*) is said *feasible* if it satisfies the dynamics described in Eq. (6.1) [135]. Here, feasibility of the optimal controller is ensured by implicitly imposing the discrete form Eq. (6.2) in Eq. (6.3).

Following an MPC scheme, *i.e.:* at time j, the control sequence U_j^* is generated considering the initial state $\mathbf{x}_0^j$, then only the first control $\boldsymbol{\tau}_0^j$ of the sequence is executed during the discretization time dt arriving to a new state $\mathbf{x}_1^j$, which is used as initial state $\mathbf{x}_0^{j+1} = \mathbf{x}_1^j$ to generate an entire new sequence U_{j+1}^* and this is repeated cyclically [81]. This procedure guarantees that the generated robot motion is part of a feasible path of at least N steps in the future. Feasibility beyond the horizon can also be ensured by making the robot reach some state where the robot can stay safely during indefinite time at the end of the horizon [175]. This property is enforced with the terminal cost $l_{term}(\cdot)$.

In particular, the DDP algorithm is used to minimize the cost function Eq. (6.3) at each iteration of the MPC. The computational efficiency of DDP allows controlling 31 degrees of freedom of the robot TALOS along a horizon of $N = 100$ time-steps with $dt = 10$ ms online (computed during the movement of the robot). DDP has the drawback of not

accepting explicit constraints, though recent results suggest a forthcoming solution to this issue [121]. Here, however, the traditional solution is to consider Eq. (6.2) as an implicit constraint.

DDP produces Riccati gains

$$\mathbf{K}_0 \triangleq \left.\frac{\partial \boldsymbol{\tau}}{\partial \mathbf{x}}\right|_{\mathbf{x}_0} \tag{6.4}$$

evaluated at the initial state, as a partial result of the optimization. Control values are interpolated using these gains, as proposed in [62], to reach an updating period of 0.5 ms on the resulting control law:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_0 + \mathbf{K}_0(\mathbf{x}_{meas} - \mathbf{x}_0), \tag{6.5}$$

with a feedback term based on the measured state $\mathbf{x}_{meas}$ which is updated at every millisecond and a feedforward term $\boldsymbol{\tau}_0 = \mathbf{k}_0$ computed optimally from the measured initial state x_0 at each MPC iteration (every 10 ms). In order to further boost the DDP performance, the pair (U^*, X^*), obtained in the previous MPC iteration, is the warm start at each computation of the control sequence. For the first control sequence, since there is no previous solution to reuse, DDP is iterated starting from a constant trajectory until convergence.

3 DEBURRING CONTROLLER

Contrary to most solutions found in the literature the WBMPC implemented on the robot does not rely on a reference trajectory. Instead, all the information about the task is encoded through the cost function and the robot's trajectory is implicitly generated. This reduces the overall complexity of the control structure because it does not require a higher level planner to be used. It however makes the design of the cost function for a single task much more challenging.

Shaping the cost function is made even more complex by the need to reconcile occasionally conflicting objectives in a single scalar function. Furthermore, the solver does not accept explicit constraints. So the cost function must incorporate relaxed safety constraints and address multiple objectives simultaneously. To simplify the process, a fixed structure is chosen where the cost is composed of sub-costs that incentivize or discourage specific robot behaviors.

3.1 *Cost function structure*

We reuse the cost architecture presented in Chapter 5 because it has already shown interesting results in [59].

The cost function is split into four different sub-costs: constraints, equilibrium, regularization, and goal

$$l(\mathbf{x}, \boldsymbol{\tau}) = w_{cons} l_{cons} + w_{eq} l_{eq} + w_{reg} l_{reg} + w_{goal} l_{goal}. \tag{6.6}$$

Each of the sub-costs has an associated weight, which can be adjusted to define the relative priority of each task.

3.1.1 *Constraints cost*

The first and most highly weighted cost, aims at preserving the integrity of the robot. It is a barrier cost that greatly penalizes any configuration that does not respect the kinematic constraints of the robot: $l_{cons}(\mathbf{x}) = \|max(\mathbf{x} - \mathbf{x}_u, 0) + min(\mathbf{x} - \mathbf{x}_l, 0)\|^2$. With $\mathbf{x}_u$ and $\mathbf{x}_l$ respectively being the upper and lower bounds of the admissible states.

3.1.2 *Equilibrium cost*

Balance is also a major concern when working with humanoid robots. The robot must stay on its feet, throughout the whole operation. It is achieved with an equilibrium cost: $l_{cons}(\mathbf{x}) = \|\mathbf{c}(\mathbf{x}) - \mathbf{c}_d\|^2$ with $\mathbf{c}(\mathbf{x})$ and $\mathbf{c}_d$ the current and desired Center of Mass of the robot. For the deburring task, maintaining the CoM of the full robot over its supporting feet is enough to penalize movements leading to losses of equilibrium. As this cost also preserves the robot integrity, it is set with the second-highest relative weight.

3.1.3 *Regulation cost*

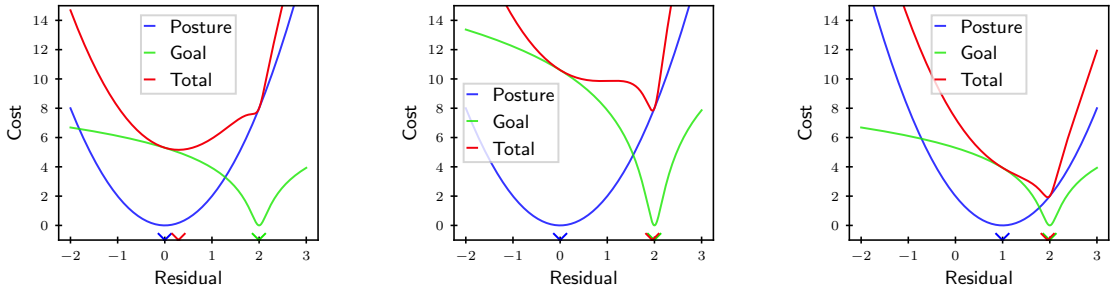
To guarantee the numerical stability of DDP, a regularization cost that ensures uniqueness of the optimal control is added: $l_{reg}(\mathbf{x}, \boldsymbol{\tau}) = (\mathbf{x} - \mathbf{x}_d)^T \mathbf{R}_\mathbf{x} (\mathbf{x} - \mathbf{x}_d) + (\boldsymbol{\tau} - \boldsymbol{\tau}_d)^T \mathbf{R}_{\boldsymbol{\tau}} (\boldsymbol{\tau} - \boldsymbol{\tau}_d)$. It prioritizes behaviors that are close to the desired state x_d built from the initial robot posture, with zero velocities. It also penalizes controls that are far from the torques $\boldsymbol{\tau}_d$ required to counteract the force of gravity in the desired position. $\mathbf{R}_\mathbf{x}$ and $\mathbf{R}_{\boldsymbol{\tau}}$ are positive definite matrices used to tune the relative impact of each joints on the regulation cost.

3.1.4 *Goal related cost*

While constraints, equilibrium and regulation are general enough to be widely used in humanoid robot applications, task-specific components are also required for the cost to be applied in a concrete experiment. For the deburring operation, the goal cost encourages the robot to position correctly its left end-effector and maintain zero velocity. It is designed as follows: $l_{goal} = log(1 + \frac{\|\mathbf{p} - \mathbf{p}_d\|}{\alpha}) + \|R - R_d\|^2 + \|\mathbf{v}\|^2$ with $\mathbf{p}$ and $\mathbf{p}_d$ the actual and desired Cartesian position of the end-effector, R and R_d the actual and desired rotation (defined as elements of $SO(3)$), $\mathbf{v}$ its Cartesian velocity and $\alpha = 0.02$.

3.2 *Cost function shaping*

From the structure presented in Section 3.1 naturally arises a set of parameters that needs to be tuned in order to achieve a specific task. A common approach is to proceed via trial and error either in a simulator or directly on the real robot. For simplicity purposes, a single tuning is often chosen for the whole movement. However, even if



(a) Baseline (b) Variable Weight (c) Variable Posture

Figure 6.2 – Simplified illustration of the cost conflict. Values are just for scale and do not represent the actual value of the cost function for our application. Colored tick on the x-axis indicate the abscissa of the minimum of each function. The distance between the red and green ticks represents the error associated to the cost function.

it is theoretically possible to shape a cost function that exploits the full abilities of the robot in every situation, it is in practice very challenging.

Despite trying to set a clear hierarchy between tasks by choosing weights with different orders of magnitude, the problem of conflict between tasks still arises during the experiments. Indeed, the posture task reference is always the same for all the cases while the desired goal can vary in all the robot workspace. It means that, most of the time, those two costs tend to attract the robot toward different equilibrium. This results in the optimal solution being a trade-off between both costs which leads to poor performances.

That is why we resort to have a cost function that varies in time and along the horizon of the MPC:

$$V(\mathbf{x}_0, t) = \sum_{i=0}^{N-1} l_i^t(\mathbf{x}_i, \boldsymbol{\tau}_i) + l_N^t(\mathbf{x}_N), \tag{6.7}$$

In order to guarantee the coherence of the problem between each iteration we update cyclically each node of the cost function so that only the last one contains new information:

$$\forall i \in [\![0 : N-2]\!], l_i^{t+1} = l_{i+1}^t, \tag{6.8}$$

All the experiments used the same receding horizon approach where the first node of the horizon is discarded and a new custom one is added at the end. This approach permits to have a richer representation of the task while keeping a simple structure for the cost function even if it requires either to hard-code the time sequence or to resort to an external planner.

We tried several approaches to generate this new node in the trajectory as shown in Fig. 6.2.

3.2.1 *Baseline*

The baseline performances is computed using a mostly fixed cost function. The only parameter that changes over time is the desired Cartesian position of the end-effector. It allows the robot to reach several targets during one experiments.

Even if a more advanced tuning could lead to better results, any further improvement is made very challenging because of the sensitivity of the performances to the cost function.

3.2.2 *Variable goal-cost weight*

A straightforward way to solve the cost conflict is to increase the relative weight of the goal cost with respect to the posture cost. A linear scheduling of the weight is chosen so that the cost only increases at the end of the movement when high precision needs to be achieved:

$$w_{goal}(t) = w_{slope}t + w_0 \tag{6.9}$$

This strategy was successfully used to conduct the first set of validating experiments on the real robot.

3.2.3 *Variable posture reference*

Another solution to solve the conflict is to update the reference posture at the same time as the goal:

$$l_{reg} = \|\mathbf{x}(t) - \mathbf{x}_{reg}(t)\| \tag{6.10}$$

where $\mathbf{x}(t)$ is the measured state and $\mathbf{x}_{reg}(t)$ a variable reference state.

This allows to improve performances without tempering with the relative weight of each cost hence preserving the safety of the robot.

To do so we do not use an external logic, but reuse solutions of previous experiments found using our control structure. In practice, we explore the environment of the task using a simple control structure and re-inject the reached posture as a reference for subsequent realizations. This approach is relevant when no expert data is available to guide the resolution.

4 APPLICATION OF THE CONTROL STRUCTURE

To validate the method presented in this chapter, we study a task which consists in reaching a series of points in sequence while achieving a good accuracy (less than 5 mm of error in our case). The accuracy threshold is chosen to match the radius of the hole in which the tool needs to be inserted.

We will explain the software architecture used during both the experiment and the simulation in Section 4.1 before detailing the two phases of test that we carried out:

— First, an exploratory phase conducted on the robot. It aimed at validating that the presented method could reach a precision of 5 mm.

— Then, a performance improvement phase focused on exploiting the full capabilities of the physical system.

4.1 *Control setup*

The control architecture is split into two levels. The computationally expensive optimal control resolution is done at 100 Hz. In the case of experiments on the real robot this part is carried out by an external computer (fitted with an AMD Ryzen 5950X, 16 cores with 64 GB of RAM). A faster control, based on the gains computed by the MPC can then be run directly on the robot at 2 kHz as shown in Eq. (6.5). The MPC implementation was based on Contact RObot COntrol by Differential DYnamic Library (Crocoddyl). The software architecture is summarized in Fig 6.3.

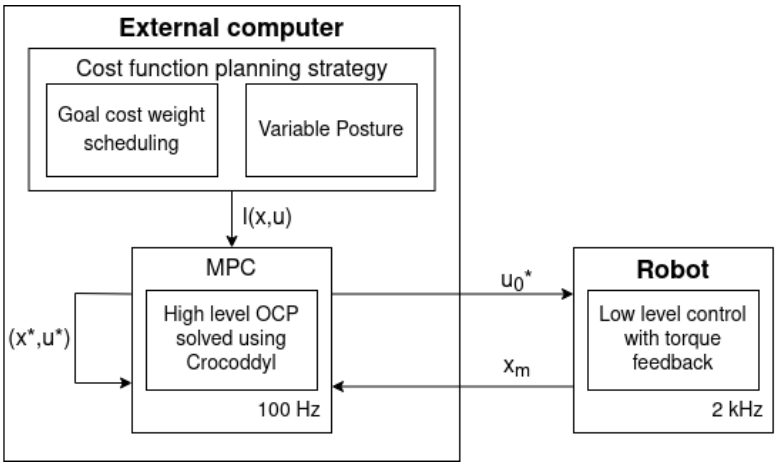


Figure 6.3 – Diagram of structure used to control the robot. $l(\mathbf{x}, \mathbf{u})$ is the cost function optimized by the OCP, $(\mathbf{x}^*, \mathbf{u}^*)$ are the current optimal state and control trajectory produced by the MPC, $\mathbf{u}_0^*$ is the control sent to the robot and $\mathbf{x}_m$ the state measured by the proprioceptive sensors of the robot.

4.2 *Concept validation in the real world*

As mentioned in Section 1.2 experiments were conducted both in our lab and directly on site at an Airbus plant.

Speed was not the focus of this stage, that is why we resorted to the gain scheduling technique to carry out the task. Indeed, it was a straightforward way to achieve the desired result in a setting where stability was not a major concern because of the low movement speed involved.

The robot successfully managed to reliably insert the tool that was fitted on its end-effector in a sequence of 4 holes [1]. In a separate experiment we checked that the robot remained compliant while it inserted the tool by having a human push its arm.

A motion capture system was used to calibrate the position of the aircraft piece with respect to the robot at the beginning of the experiment. Other than that, no visual feedback was required during the experiment and the proprioceptive based movement was precise enough to carry out the task.

4.3 *Performance improvements*

After validating the relevance of the chosen approach, work was done to improve the performances and the achieved movement speed using the PyBullet simulator [56]. This simulator has been used in the past as a validation step before deploying new movements on the robot.

4.3.1 *Benchmark*

First, a benchmark of the three approaches presented in Section 3.2 is showed. The performances of the controllers are evaluated according to two metrics :

— The distance between the center of the hole and the tip of the end effector. The task is considered to be successful if this distance is below 5 mm;

— The time to successfully carry out the task. Which is the time between the beginning of the movement and the moment where the tip of the tool is less than 5 mm away from the hole and stays in this zone.

We set up the robot to reach a precise point in space starting from its default position using all three approaches. The results are compiled in table 6.1.

Method	Baseline	Variable weight	Variable posture
Accuracy (mm)	8.27	0.85	0.24
Reach time (s)	_	1.06	1.24

Table 6.1 – Comparison of the accuracy and of the *Baseline*, *Variable weight* and *Variable posture* methods.

As can be seen in Fig. 6.4, the baseline is not precise enough to reach the desired threshold. This illustrates the limitation that we mentioned in Section 3.2. The other two solutions can solve this issue if tuned properly.

However, it is worth noting that this results comes from a simulation and cannot be directly translated to the real world because of unforecasted disturbances and discrepancies between the model and the real robot.

In particular, the variable weight approach suffers from a major weakness. Changing the relative weight of costs may reduce the significance of the safety related cost. This

1. https://peertube.laas.fr/w/s6UeEXiheSCD47EZrwhksS

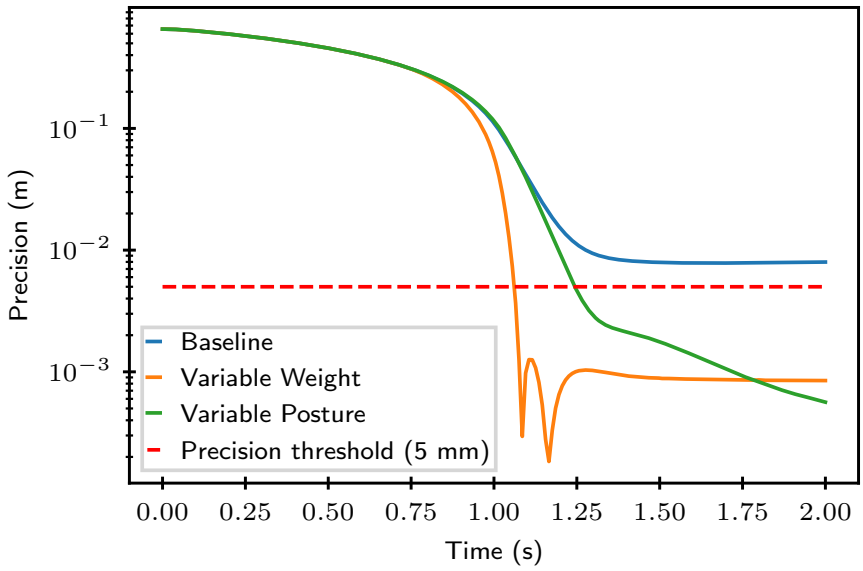


Figure 6.4 – Evolution of the cartesian position of the end effector with respect to time.

can lead to more dangerous movements if done recklessly. In addition to that, higher gains can hinder the stability of the control. We can see oscillations in the movement which indicates a less stable control.

On the other hand, updating the reference posture can solve the cost conflict without altering the relative weight of the tasks. Since the weights of the placement and posture task are relatively low with respect to the limits and stability cost in this setting, this approach is less dangerous for the robot.

4.3.2 *Performances of the variable posture approach*

Because it is less dangerous for the robot while still being efficient, the variable posture approach is tested on a sequence of two holes. Fig 6.5 indicates the robot can precisely reach both holes with a transition time of 0.5 seconds. While we do not have precise data regarding the performances of a human operator for this specific experiment, it has been reported to us by Airbus employees that a worker would take around one second to transition between two holes. It means that the attained performances are in the same order of magnitude of what a human could achieve, which was not the case with the baseline solution.

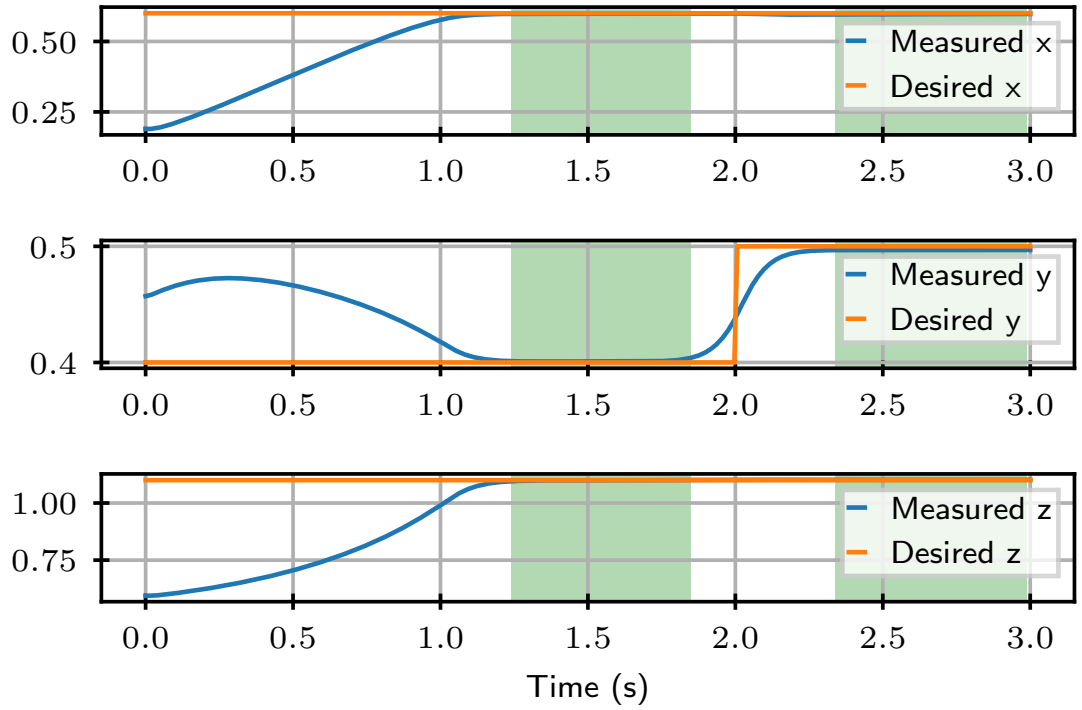


Figure 6.5 – Simulated evolution of the Cartesian position (in meters) of the end effector with respect to time. The distances are given with respect to the center of mass of the robot. The x-axis is oriented toward the front of the robot, the y-axis to the left and the z axis is going up. Regions highlighted in green are when the end-effector is less than 5 mm away from the target position.

5 DISCUSSION

5.1 *Difficulties to deploy an efficient motion*

Even if the control scheme was successfully deployed on the robot, increasing the performances still represents a major challenge. Indeed, in [213] a similar performance improvement as the one described in paragraph 4.3 was applied to the TALOS robot without any particular precaution. It caused the controller to inject a high quantity of energy in the system which, despite the safeties that are implemented, damaged the robot. This is not desirable and may imply to expand the solver and include a passivity constraint as proposed in [72], or a similar approach to prevent this type of behavior. Such an extension is beyond the scope of this work.

To reduce the occurrence of accidents, the manufacturer of the robot, PAL-Robotics, provides a high fidelity simulator which includes a model of the actuators. It also warns the user of possible collision using an energy based criteria for each actuator separately. This is unfortunately not sufficient to guarantee the safety of the robot.

This means that the only tractable way to proceed is to gradually increase performances on the real robot. However, in the case of a complex system like a humanoid robot, this requires extensive manpower (at least two people are needed to operate the robot safely). It also subjects the hardware to high wear and tear.

5.2 *Need for planning to achieve human-like performances*

The proposition made in this chapter to improve performances revolves around injecting relevant information inside the system through the cost function. It differs from traditional motion planning approaches because it does not rely on an external heuristic to provide the necessary information. Instead, it leverages data from previous experiments to achieve the desired performances.

This drives the intuition that work should be done to build a form of memory for the system. This memory would be queried in every situation to select the appropriate parameters of the cost function. It could be populated by exploring the environment using our control approach.

A hybrid MPC/RL approach could be used to achieve this goal. The RL Agent would be trained to maximize a higher level reward function that depends upon the performances (accuracy and speed). It would control the robot through the choice of the parameters of the cost function. This means that the MPC, with the structure presented in this chapter, would still be used on the robot. The reference posture would however be reactively picked by the RL algorithm. [219] successfully deploys Proximal Policy Optimization to control a quadruped robot. However, the approach we present would be much more computationally intensive because of the more advanced control structure that would need to be simulated. That is why off-policy algorithms, such as Soft Actor-Critic, that are known to be more sample efficient would be more appropriate.

6 CONCLUSION

This chapter demonstrates the use of high frequency MPC to carry out a position task with an accuracy of few millimeters with the humanoid robot TALOS, controlled in torque. Strategies regarding the shaping of the cost function are the main focus of this chapter. Simulations show that changing the reference posture during the movement can improve the speed of completion of the task to human like levels.

In the short term, we plan to demonstrate the shown results on the robot. We also plan to continue this work by leveraging machine learning as a planning tool to reactively choose the appropriate reference configuration for a wide range of situations.

7

RL-BASED REACTIVE CONTROLLER

In Short

This chapter builds upon the conclusions drawn in the previous chapter to propose a reactive planner that exploits RL. This solution enhances the WBMPC to improve the robot's performance. In addition to better overall performance, it addresses some limitations of the MPC and offers the ability to incorporate additional constraints into the control structure. This chapter was submitted to *Transactions on Automation Science and Engineering*.
Section 2 recap elements from earlier chapters to allow this chapter to be read independently. Readers who have already familiarized themselves with the previous chapters may proceed directly to Section 3.

Contents

1 INTRODUCTION

1.1 *Context presentation*

Assembly and insertion tasks constitute a significant segment within the production cycle of industrial items [124]. Despite extensive research efforts aimed at automating these tasks, they continue to be largely carried out by human workers. Current solutions predominantly rely on manipulator arms controlled by positional inputs. To facilitate closer collaboration between humans, and robots and to adapt to less structured work environments, the focus has shifted toward the adoption of lightweight robotic arms [23], accompanied by the incorporation of compliance mechanisms.

Another fundamental characteristic of small batch production is variability in the working environment which does not accommodate well classic manipulator arms. The prospect of mobile base manipulators emerges as a promising solution for augmenting the flexibility of industrial robots. A majority of solutions discussed in the literature involve a wheeled platform equipped with a robotic arm [69]. In contrast, some researchers [139] advocate for the use of legged robots, particularly in scenarios characterized by cluttered and hard-to-reach environments, such as those encountered in aircraft manufacturing sites.

Our viewpoint posits that, regardless of the robot's architectural design, a comprehensive understanding of the entire system's dynamics is imperative to fully harness the platform's performance capabilities. In this context, demonstrating such feat on a humanoid robot appears as a significant milestone that can be achieved on our way to bring robots to the factory alongside humans. As exemplified in [21], a proposed approach involves a unified force-control scheme paired with a reinforcement-learning policy. This combination facilitates the acquisition of skills in contact-intensive manipulations, even for rigid position-controlled robots.

Following up on Chapter 6, the idea is to use a hybrid approach combining MPC and RL to improve the performances previously reached. We leverage the memory and exploration capabilities of learning based methods to extend an MPC while safe guarding the guarantees offered by a model based approach.

The application has been chosen in the scope of the ROB4FAM project. It aims at carrying deburring tasks on aircraft parts. Because we are working on solutions with a low Technology Readiness Level, the task is simplified into the insertion of a 3d printed tool into the holes of the structure.

The main topic of this chapter is to execute an insertion task on a torque controlled humanoid robot.

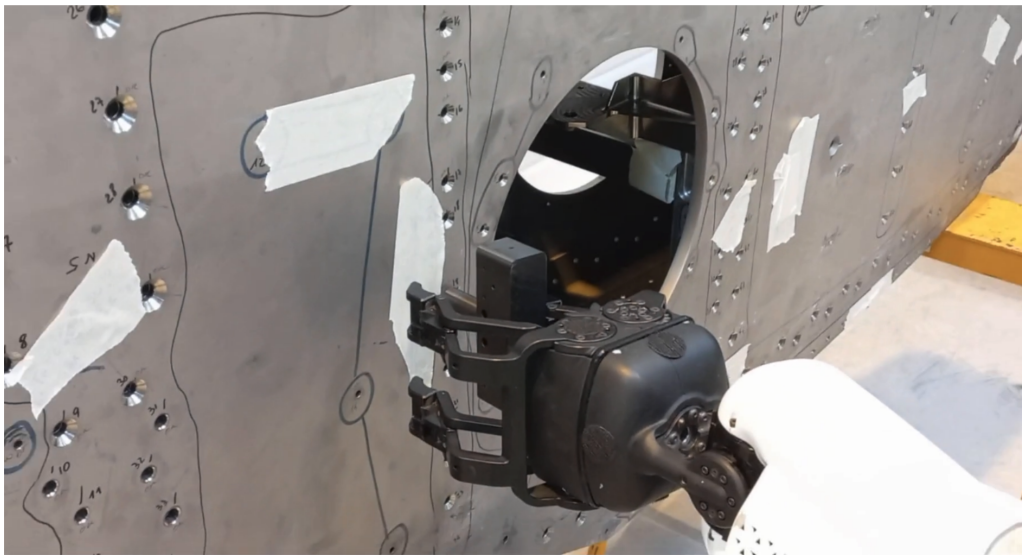


Figure 7.1 – Deburring task, high precision for a fine insertion into a hole using WBMPC on a torque controlled robot.

1.2 *Related work*

Most of the literature regarding industrial insertion tasks carried out by robot focuses on manipulator arms. In that case, the main challenge of the task is to control the applied forces, even in the presence of uncertainties. This property known as compliance is a pivotal factor in enhancing human-robot interaction and adaptability. It can be introduced through two distinct avenues: passive approaches and active methodologies.

Passive approaches [264] rely on specialized compliant devices tailored for each task. However, they have the drawback of not being controllable or universal.

On the other hand, active methodologies depend on their ability to measure interaction forces with the environment. This can be achieved using a dedicated sensor [64], yet this solution specializes the robot and may not be suitable for settings where versatility is crucial. A promising alternative is to reconstruct interaction forces through joint torques [165], which enables the embedding of compliance inside the control. Active force control strategies can be declined into two categories: hybrid force/position control and impedance control [111]. However, these traditional approaches often lack adaptability and robustness when faced with new situations.

To overcome the shortcomings of classic solutions, learning-based methods have been gradually applied to industrial tasks. There are two main ways to integrate data-based approaches into industrial robotic tasks: Imitation Learning [275] and Reinforcement Learning. Imitation Learning involves extracting information from example movements, while Reinforcement Learning relies on interaction with the environment. For instance, [225] proposes solving industrial insertion tasks for electronic components using RL.

[277] suggests combining traditional active force control strategies with learning-based optimization to carry out peg-in-hole assembly tasks.

As opposed to the literature regarding fine insertion tasks using manipulator arms, the examples of such tasks carried out by legged manipulator are few. A lot of inspiration can however be drawn from the literature regarding whole-body control and loco-manipulation.

A straightforward way to handle manipulation for mobile based robot is to treat the movement of the base and the manipulation as independent tasks. This might however be suboptimal when trying to achieve very dynamic movements. In addition to that, even if this approach might be viable in some cases with quadrupeds, it is unlikely that it will work for humanoids (the subject of this manuscript) because of the inherent instability of those architectures. [20] proposes a unified loco-manipulation approach for quadrupeds based on hierarchical planning. [184] presents a versatile planning framework for loco-manipulation on humanoid robots.

MPC has been widely used in for the control of humanoid robot thanks to its ability to handle constraint and provide guarantees. [59] uses an MPC augmented with a memory of motion to carry out whole-body manipulation tasks, although not reaching the precision we are looking for in our case. [144] proposes a way to handle multi-contact and interaction with the environment.

MPC approach excel when the dynamic of the system are well known. But it often relies on local online solvers that only handle differentiable constraints. Reinforcement Learning as been gaining traction over the last few years because it can use rewards that more directly encode the task of interest. RL on complicated hardware such as legged robot as been made accessible thanks to the scaling ability offered by novel hardware [220]. [208] exploits these technical advancements to obtain real-world locomotion. [237] highlights the need for good modeling of the robot to carry out sim to real learning. Another approach [239] exploits more sample efficient algorithm to train directly on the robot, bypassing the need for a model of the system. [88] proposes a data based approach for simultaneous manipulation and locomotion. [63] proposes a Reinforcement Learning based approach for box loco-manipulation on humanoid robots.

[123] and [92] uses examples from trajectory optimization to drive the exploration of the RL agent.

Even if some solutions tackle the constraint with RL methods [46], [142], [155] they do not guarantee respect of the constraints during training.

Combining the advantages of learning approaches with classical model based trajectory optimization has been the subject of several publications. [272] uses MPC as a function approximator inside an RL framework to benefit from the guarantees offered by the MPC while learning the model discrepancies between the MPC and the reality. [166] expand on the method developed by [20] by incorporating a learned locomotion policy with an MPC for manipulation, demonstrating an effective way to incorporate specialized controllers within an RL solution. However, this approach does not extend the guarantees offered by the MPC to the entire system.

To address this limitation, solutions that provide safety guarantees have been proposed in [216] and [162]. [216] integrate a differentiable MPC as the final layer of an actor within an actor-critic framework, although this approach is constrained by the requirement for a differentiable MPC, which restricts the variety of applicable model-based methods. On the other hand, [162] modify the action space of the RL agent to act on the tangent space of the constraints, thereby offering guarantees even during the training phase.

1.3 *Contributions*

The objective of the task is to insert a 3d printed tool inside the holes of a mockup aircraft part. This task is similar to the one presented in [177]. It simulates deburring, an operation that needs to be undertaken after drilling holes to clean up material residues.

We propose to extend the previously used MPC with a learned policy that reactively shapes the cost function. This addition to the control structure allows us to extract more performance from the robotic platform in use.

Our method also notably provides a systematic approach to explore the cost function space of the whole body controller.

This work represents an efficient use of the possibilities of RL while maintaining a degree of safety (brought by the MPC) that is necessary to envision large scale deployment of robots in the industry.

The contributions presented in this chapter are :

— Reactive cost shaping using Reinforcement Learning

— High speed movements to carry out precise deburring task with a humanoid robot

— Approach to provide systematic cost function space exploration using Reinforcement Learning

— Foundational work toward deploying reinforcement learning in safety requiring setups

Theoretical foundations of the MPC are given in section 2. Then further explanation regarding RL a provided in section 3. Details regarding the full implementation are explained in section 4. Then the results are detailed in section 5.

2 WHOLE-BODY MODEL PREDICTIVE CONTROL

2.1 *Robot Modelling*

Similarly to what has been presented in Chapter 4, we describe the attitude of the robot at a given time, we use the generalized configuration $\mathbf{q} \in SE(3) \times \mathbb{R}^{n_j}$, with n_j the number of robot's controlled joints. It defines the global position and orientation of the base as well as the posture of the robot's joints. We model the evolution of this

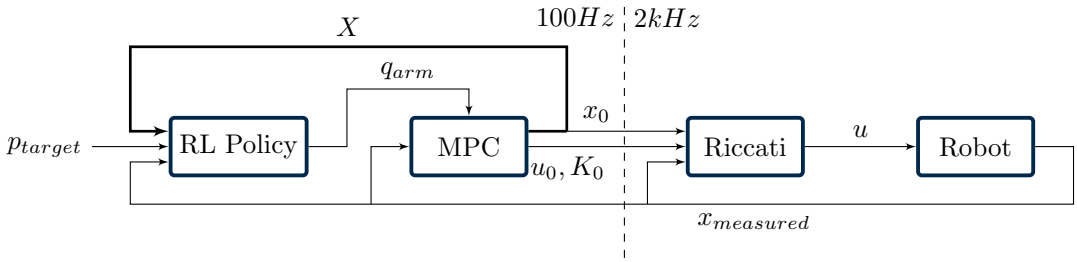


Figure 7.2 – Control structure implementing RL tuned MPC cost function.

configuration as resulting from internal and external forces, as described by the rigid-body dynamics [267]:

$$\begin{bmatrix} \mathbf{M} & \mathbf{J}_c^\top \\ \mathbf{J}_c & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{S}^\top \boldsymbol{\tau} - \mathbf{b} \\ -\dot{\mathbf{J}}_c \dot{\mathbf{q}} \end{bmatrix}, \tag{7.1}$$

where $\mathbf{M}$ is the inertia matrix, $\mathbf{b}$ stands for Coriolis, centrifugal and gravity forces, joint-motor torques $\boldsymbol{\tau} \in \mathbb{R}^{n_j}$ affect only the actuated joint as indicated by the selection matrix $\mathbf{S} \in \mathbb{R}^{(n_j+6)\times n_j}$, and all contact wrenches $\boldsymbol{\lambda}_i \in \mathbb{R}^6$ are contained in $\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_1 & \cdots & \boldsymbol{\lambda}_i & \cdots \end{bmatrix}$ with the application points described respectively by the Jacobians $\mathbf{J}_i \in \mathbb{R}^{6\times n+6}$ contained in $\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & \cdots & \mathbf{J}_i & \cdots \end{bmatrix}$.

The second line of Eq. (7.1) imposes constraints on the parts of the robot that are in contact with the environment, requiring them to remain stationary. The explicit formulation of contacts in the dynamics assumes that the contact sequence is known a priori. In our case, this assumption is not limiting, as we focus on maintaining the robot's equilibrium without moving the feet. However, similar control methods have been successfully applied to locomotion tasks [60]. Furthermore, [140] proposes a generalization that eliminates the need for pre-computed contact sequence. In this work, we do not address loco-manipulation as it is beyond the scope of our study.

Using the dynamics described in Eq. (7.1), the state of the robot $\mathbf{x}$ is controlled by inputting the desired command torques $\mathbf{u} = \boldsymbol{\tau}$ on joint motors during some discretization period dt. In our case, the state is formed by the robot configuration $\mathbf{q}$ and its time derivative $\dot{\mathbf{q}}$, i. e. :

$$\mathbf{x} = (\mathbf{q}, \dot{\mathbf{q}}) \tag{7.2}$$

2.2 *Optimal Control Problem*

Given the model of the robot dynamics, we can formulate an optimal control problem to find the torque sequence that minimizes the cost along a horizon of size $N \in \mathbb{N}$. Specifically, we are looking for the control sequence $U^* = \{\mathbf{u}_0^*, ..., \mathbf{u}_{N-1}^*\}$ that will take the robot from a starting state $\mathbf{x}_0$, which will be in our case the current state of the

system, to a desirable final state while minimizing the running and terminal costs l_t and l_{term}. This problem is a continuous control problem, but we discretize it to make it computationally tractable. The formulation of the discrete-time optimal control problem is as follows:

$$\begin{Bmatrix} \mathbf{x}_0^*, ..., \mathbf{x}_N^* \\ \mathbf{u}_0^*, ..., \mathbf{u}_{N-1}^* \end{Bmatrix} = arg \min_{X,U} \sum_{t=0}^{N-1} l_t(\mathbf{x}_t, \mathbf{u}_t, \mathbf{a}_t) + l_{term}(\mathbf{x}_N, \mathbf{a}_N) \\ s.t. \quad \mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t) \tag{7.3}$$

Where $f(\mathbf{x}_k, \mathbf{u}_k)$ describes the discrete-time dynamics of the system and corresponds to the numerical integration of Eq. (7.1). $l_t(t \in [\![0; N-1]\!])$ and l_{term} are respectively the running and terminal costs function. $\mathbf{a}_t(t \in [\![0; N]\!])$ encompasses all the hyperparameters of the cost function that need to be properly chosen. More details regarding the formulation of these parameters will be provided in Section 4.

To generate the appropriate control for the robot, we use Crocoddyl, which relies on FDDP, a modified version of DDP (Differential Dynamic Programming) [176]. The principle of DDP is to find an optimal solution to the problem by locally searching for a solution around an initial guess (X^0, U^0). As explained in Chapter 4, FDDP provides a more robust strategy by casting the single shooting problem into a multiple shooting formulation. This approach has the advantage of accepting unfeasible warm-starts, i. e. initial guess that does not satisfy the dynamics described in Eq. (7.1).

We chose Crocoddyl for its ability to generate control at high enough frequency for a complex system like the humanoid robot TALOS. We used it to solve Eq. (7.3) at each iteration of the MPC. The computational efficiency of the FDDP implementation allows controlling $n = 22$ joints of the robot TALOS along a horizon of $N = 100$ time-steps with $dt = 10$ ms online (computed during the movement of the robot).

DDP has a limitation in that it does not accept explicit constraints, except for the dynamics, which is treated as an implicit constraint. Although there are some techniques to address this issue, such as those proposed in [125] and [120], they have not yet been applied to practical cases like the one we study. Instead, our method builds upon a controller that has already demonstrated its ability to control humanoid robots for advanced tasks, as shown in [203], [144] and [61].

2.3 *Optimal Control Policy*

The optimal policy is extracted from the solution provided by Crocoddyl following an MPC scheme. At time j, an optimal control sequence U_j^* is generated, considering $\mathbf{x}_0^j$ as the initial state. Only the first control $\mathbf{u}_0^j$ of the sequence is executed during the discretization time dt, resulting in a new state $\mathbf{x}_1^j$. This state is used as the initial state $\mathbf{x}_0^{j+1} = \mathbf{x}_1^j$ to generate an entire new sequence U_{j+1}^*. This control strategy is repeated cyclically, as described in [81]. The procedure ensures that the generated robot motion is part of a feasible path of at least N steps (corresponding to 1 second in our case) in

the future. Feasibility beyond the horizon can also be achieved by making the robot reach a state where it can stay safely indefinitely at the end of the horizon, as described in [175]. This property is enforced with the terminal cost $l_{term}(\cdot)$.

To improve the performance of DDP, the pair (U^*, X^*), obtained in the previous MPC iteration, is used as the warm start at each computation of the control sequence. For the first control sequence, DDP is iterated starting from a constant trajectory until convergence.

DDP not only produces an optimal torque but also Riccati gains:

$$\mathbf{K}_0 \triangleq \left.\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\right|_{\mathbf{x}_0}, \tag{7.4}$$

which can be interpreted as sensitivity to the initial state. According to [62], these gains can be used to generate an approximation of the optimal policy:

$$\mathbf{u} = \mathbf{u}_0 + \mathbf{K}_0(\mathbf{x} - \mathbf{x}_0), \tag{7.5}$$

with a feedback term based on the measured state $\mathbf{x}$ (updated at 2 kHz) and a feed-forward term $\mathbf{u}_0$ computed optimally from the measured initial state $\mathbf{x}_0$ at each MPC iteration (every 10 ms). This allows for a significant reduction in the updating period of the control law, resulting in better performance on the system. However, we also noticed that it caused higher frequencies in the control which may cause wear and tear on the system in the long run.

2.4 *Parameter optimization*

The application of MPC to real-world problems presents a significant challenge in the form of parameter tuning. The previously introduced formulation highlights two sets of parameters that contribute to the practical complexity of implementing Crocoddyl in a real-world context:

— The choice of the warm-start (X^0, U^0).

— The formulation of the cost function $l_t(\mathbf{x}, \mathbf{u}, \mathbf{a})$ and $l_{term}(\mathbf{x}, \mathbf{a})$ which is equivalent, in our case, to choosing the right $\mathbf{a}$.

A prevalent suggestion for resolving the initial guess problem is the use of a memory of motion to warm start the non-linear solver, as proposed in [168] and [156]. However, this approach raises several concerns, including the construction of the memory, trajectory encoding, and the selection of the appropriate warm-start when required. In our case, we presume that the method outlined in section 2.3 is sufficient, provided the cost function is correctly defined.

The issue of cost tuning is a critical aspect of optimal control-based approaches, as it significantly impacts the performance of the system. Traditionally, this problem has been addressed through laborious trial and error methods, which are time-consuming and not particularly efficient. The search for generalizable cost function tuning method is still an ongoing research endeavor.

The general idea presented in this chapter is to find the parameters that optimize the average performance of the controller on an ensemble of tasks, in our case an ensemble of targets to reach. Existing solutions either leverage expert demonstrations [236] or exploration of the parameter space with an evolutionary algorithm [57].

However, they often limit to finding the best set of parameters, i. e. $\mathbf{a}^*$ constant over time. We believe that reactivity is necessary to have the emergence of complex behaviors. Therefore, we introduce RL to find an optimal policy that links observations and task to appropriate cost function parameters.

3 REINFORCEMENT LEARNING AGENT

The solution we propose incorporates an RL-based reactive cost planner, which dynamically tunes the MPC's cost function to influence its behavior, as shown in Fig. 7.2.

As explained in Section 2.3 of Chapter 3, MPC typically relies on a planner to generate a reference trajectory to be followed by the controller. Our approach differs from this traditional architecture, because the proposed planner explicitly accounts for the controller's specifications and limitations, which are often overlooked in standard designs. To the best of our knowledge, this is the first time such an approach has been applied to a humanoid robot control scheme of this nature.

3.1 *Markov Decision Process*

The cost tuning problem is formulated as a policy search within a Markov Decision Process (MDP), defined as a tuple (S, Θ, T, r):

- S: The state space, comprising the robot's state and task-related information.
- Θ: The action space, consisting of a continuous, reduced set of the MPC's cost function parameters.
- $T : S \times S \times \Theta \to [0, +\infty[$: The probability density of reaching state $\mathbf{s}' \in S$ when executing action $\mathbf{a}_t \in \Theta$ from state $\mathbf{s} \in S$.
- $r : S \times \Theta \to R$: The reward associated with each transition.

Our objective is to find a policy that maps the environment's state $\mathbf{s} \in S$ to an action $\mathbf{a} \in \Theta$, enabling the MPC to converge to a feasible solution and avoid local minima. This optimal policy is defined as:

$$\pi^*(\mathbf{s}) = arg\max_{\mathbf{a}} Q(\mathbf{s}, \mathbf{a}) \tag{7.6}$$

With Q defined as:

$$Q(\mathbf{s}, \mathbf{a}) = r(\mathbf{s}, \mathbf{a}) + \gamma \max_{\mathbf{a}} \mathop{\mathbb{E}}_{\mathbf{s}' \sim T}[Q(\mathbf{s}', \mathbf{a})] \tag{7.7}$$

Here, $r(\mathbf{s}, \mathbf{a})$ represents the reward obtained by executing action $\mathbf{a}$ from state $\mathbf{s}$, and $\mathbf{s}'$ is the resulting state. $\gamma \in [0, 1]$ is the discount factor, which determines the importance of future rewards relative to immediate rewards. This recursive formulation of the

Q-function is based on Bellman's Principle of Optimality and allows us to leverage experimental (or simulation) data to refine the Q-function.

The specificity of our approach lies in the definition of T, which links together subsequent states $\mathbf{s}$, and $\mathbf{s}'$. This probability density must reflect that the MPC, presented in part 2.3, is used to control the robot. In practice, this is achieved by incorporating the MPC into the simulated environment during the training phase.

3.1.1 *Action*

Following upon the conclusion exposed in the Chapter 6. We choose to modulate the reference robot state $\mathbf{x}_r$ of the state regularization cost. Further detail on exactly how this action is integrated in the general structure of our MPC cost will be provided in part 4.

This approach allows us to address the issue of fixed reference posture leading to cost conflicts, which negatively impacts the performances of the controller.

3.1.2 *State*

The complexity of our approach with respect to more common RL usage resides in the presence of the MPC inside the environment. It means that identifying the sole state of the robot is not sufficient to accurately predict future outputs. That is why the state should also reflect the internal state of the controller. The environment's state should encompass:

— The robot's state,

— The controller's state,

— Information about the current task, in our case, the position of the target to reach.

3.1.3 *Reward*

The reward should be designed to be maximal when desired behavior arises on the robot, *i.e.:* the robot precisely reaches the designated target. Essentially it should align with the MPC's cost function defined in Eq. (7.3). The flexibility of RL should however make the cost definition task much simpler for the higher level problem and will not require us to add complicated extra regularization terms.

The high level performances we seek to optimize are:

— The time to reach the target,

— The distance to the target at the end of the movement.

3.1.4 *Reinforcement Learning Algorithm*

In order to carry out the proposed approach, we employ the Soft Actor-Critic (SAC) [100] algorithm. The choice of SAC is primarily driven by its superior sample efficiency compared to state-of-the-art on-policy algorithms, such as Proximal Policy Optimization (PPO) [228].

In our case, the need for a sample-efficient algorithm is particularly critical, as the MPC must be integrated within the RL environment. This configuration necessitates substantial computational resources, making sample efficiency a key factor in the algorithm's selection.

SAC is an off-policy RL algorithm that builds upon the foundations of Deep Deterministic Policy Gradient (DDPG) [160]. It introduces a few essential modifications, such as the use of a stochastic policy, the incorporation of an entropy term in the objective function, and the employment of twin Q-functions to mitigate the overestimation bias. These improvements contribute to SAC's robustness, stability, and sample efficiency, making it an ideal choice for our proposed approach.

4 REACTIVE COST SHAPING

4.1 *Cost function structure*

The structure the cost function beyond the general case, which is a scalar function that takes the state and control as arguments: $l : \mathbb{R}^{n_s} \times \mathbb{R}^{n_c} \rightarrow \mathbb{R}$ (where n_s and n_c represent the size of the state and control spaces, respectively). By imposing additional structure, we can render the exploration tractable and reduce the size of the space to be explored. However, this approach may constrain the ability of the RL planner to influence the robot's behavior.

We nonetheless demonstrate, in our case, that even when acting on a limited set of parameters, the Reactive Cost Shaping agent can significantly impact the system's overall performance. To achieve this, we build upon the previously presented architecture as a foundation for our experiments. This architecture consists of eight distinct costs:

1. State limit cost:
$$l_{\mathbf{x}\_lim} = \|max(\mathbf{x} - \mathbf{x}_u, 0) + min(\mathbf{x} - \mathbf{x}_l, 0)\|^2 \tag{7.8}$$
It ensures that the state $\mathbf{x}$ remains within the admissible bounds, defined by $\mathbf{x}_u$ (upper bound) and $\mathbf{x}_l$ (lower bound).
2. Control limit cost:
$$l_{\mathbf{u}\_lim} = \|max(\mathbf{u} - \mathbf{u}_u, 0) + min(\mathbf{u} - \mathbf{u}_l, 0)\|^2 \tag{7.9}$$
Similar to the state limit cost, it constrains the control $\mathbf{u}$ within the admissible bounds, defined by $\mathbf{u}_u$ and $\mathbf{u}_l$.
3. Center of mass tracking cost:
$$l_{com} = \|\mathbf{c}(\mathbf{x}) - \mathbf{c}_d\|^2 \tag{7.10}$$
Encourages the robot's current center of mass $\mathbf{c}(\mathbf{x})$ to be close to the desired center of mass $\mathbf{c}_d$.

4. End-effector position cost:

$$l_{EF\_pos} = log(1 + \frac{\|\mathbf{p} - \mathbf{p}_d\|}{\alpha}) \tag{7.11}$$

Drives the actual end-effector position $\mathbf{p}$ toward the desired position $\mathbf{p}_d$, $\alpha = 0.02$.

5. End-effector orientation cost:

$$l_{EF\_rot} = \|R - R_d\|^2 \tag{7.12}$$

Promotes the alignment of the end-effector orientation R (defined as an element of $SO(3)$) with the desired orientation R_d.

6. Regularization of the end-effector velocity:

$$l_{EF\_vel} = \|\mathbf{v}\|^2 \tag{7.13}$$

Penalizes the end-effector's Cartesian velocity $\mathbf{v}$, preventing unstable end-effector movements.

7. Control regularization cost:

$$l_{\mathbf{u}\_reg} = (\mathbf{u} - \mathbf{u}_d)^\top \mathbf{R}_\mathbf{u} (\mathbf{u} - \mathbf{u}_d) \tag{7.14}$$

This cost encourages the control $\mathbf{u}$ to be close to the desired control $\mathbf{u}_d$, which is the torque required to counteract the force of gravity in the desired position. The positive definite matrix $\mathbf{R}_\mathbf{u}$ is used to tune the relative impact of each joint on the regulation cost.

8. State regularization cost:

$$l_{\mathbf{x}\_reg} = (\mathbf{x} - \mathbf{x}_r)^\top \mathbf{R}_\mathbf{x} (\mathbf{x} - \mathbf{x}_r) \tag{7.15}$$

This cost promotes the alignment of the state $\mathbf{x}$ with the reference state $\mathbf{x}_r$. The positive definite matrix $\mathbf{R}_\mathbf{x}$ is used to tune the relative impact of each joint on the regulation cost.

4.2 *Cost function shaping*

Incorporating a high-level cost planner into the pipeline necessitates the use of a time-varying and horizon-dependent cost function. Given our MPC control scheme (outlined in part 2.3) and the receding horizon approach, maintaining coherence of the cost function between resolutions is crucial. Indeed, it allows us to leverage previous MPC solutions as a warm-start for new resolutions. Consequently, the cost function nodes are updated cyclically:

$$\forall i \in [\![0 : N-2]\!], l_i^{t+1} = l_{i+1}^t, \tag{7.16}$$

In practical terms, the sole new information introduced to the problem is associated with the final running node. Accordingly, we require only one parameter set, $\mathbf{a}$, for each time step.

The parameter selection for optimization is guided by the conclusions drawn during previous experiments, which demonstrated that appropriate posture selection significantly impacts control performance. Consequently, the RL will not optimize the entire hyperparameter set of the cost function. Our primary focus is the adjustment of the state regularization cost reference, $\mathbf{x}_r$. We introduce multiple strategies for modifying this parameter, aiming to establish a comprehensive benchmark for our approach.

4.2.1 *Baseline MPC*

For comparison, we include results obtained when the reference posture remains constant. This posture corresponds to the robot's initial state $\mathbf{x}_0$ throughout the movement:

$$\mathbf{a}_t = \mathbf{x}_0 \tag{7.17}$$

We consider this approach to represent a reasonable performance level for an MPC fine-tuned by a trained operator with extensive system knowledge. It illustrates the best achievable outcome, given a reasonable time allocation for tuning. We refer to this strategy as *baseline* MPC throughout the rest of the chapter.

4.2.2 *Posture feedback*

This method, referred to as *posture feedback*, provides a simple solution to address the regularization issue. The currently measured state $\hat{\mathbf{x}}_t$ serves as a regularization reference to prevent potential cost conflicts:

$$\mathbf{a}_t = \hat{\mathbf{x}}_t \tag{7.18}$$

However, this approach creates a feedback loop between the robot's movement and the OCP's cost function. While no stability proof is offered, the underlying assumption is that priority costs (state and control limits) will ensure the system's safety. It means that, regardless of its quality, no regularization reference should result in harmful control.

4.2.3 *Cost shaping policy*

The third strategy involves employing a policy trained by reinforcement learning to determine the optimal posture reference:

$$\mathbf{a}_t = \pi(s) \tag{7.19}$$

with π the policy introduced in Eq. (7.6). This strategy is the primary contribution of this part and will be simply referred to as *RL* in the next parts.

4.3 *Reinforcement Learning*

Additional design choices were made in order to deploy the RL policy. The full set of hyperparameters used for the training is provided in appendix 8.1. The full code, including the MPC and the architecture for carrying out trainings, is available online [1].

4.3.1 *Environment*

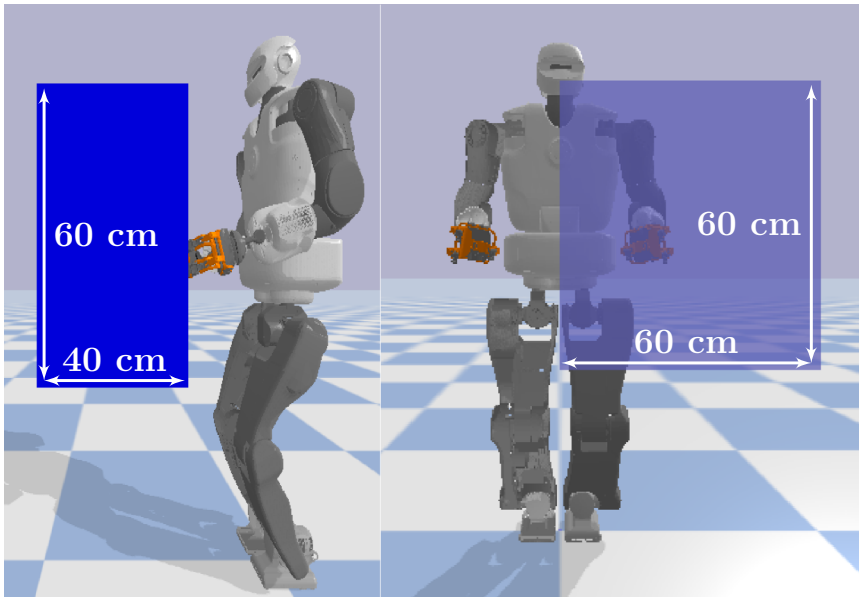


Figure 7.3 – Illustration of the workspace (in blue) from which targets are sampled to carry out benchmarks.

The environment is designed to evaluate the control structure's ability to precisely reach a set of targets. At the beginning of each episode, the robot starts in its initial *half-sitting* posture. A target is randomly chosen inside a predefined workspace. The episode is successful if the robot brings the tip of the tool placed in its end effector, less than 5 mm away from the target. This precision threshold corresponds to the radius of the hole in which the insertion should take place in reality. A representation of this setup can be seen on Fig. 7.3.

The workspace from which targets are sampled is a $400 \times 600 \times 600$ mm box in front of the robot. Since the task is carried out using the left arm, we chose the target space to be on the left side of the robot. The results can be extended by symmetry to the other side by changing the hand used by the robot.

The tests were conducted using the PyBullet simulator [56]. To increase the robustness of training, additional effects were added to the simulator:

1. https://github.com/ComePerrot/talos-deburring

— The computation delay of the OCP ($\approx$ 10 ms).

— Additional limits beyond the ones modeled in the MPC. They represent additional safeguard strategies implemented by the constructor, PAL Robotics, to protect the robot.

4.3.2 *Reward*

The reward given during the policy training aims to optimize two criteria: reach time and precision. Additional rewards penalize failures and minimize torque outputted by the MPC.

Our proposal is to have a fixed duration for each episode, along with a positive reward when the robot gets close to its objective. This approach encourages both speed and precision with a single reward. The episode is interrupted if the robot exceeds a limit, and a significant negative reward is associated with the failure.

The detailed formulation of the reward is the following:

$$r = w_s r_s + w_d r_d + w_\emptyset r_\emptyset + w_{trunc} r_{trunc} \tag{7.20}$$

With

— $r_s = 1$ in case of success, 0 otherwise.

— $r_d = 1 - \|\mathbf{p} - \mathbf{p}_d\|$ with $\mathbf{p}$ and $\mathbf{p}_d$ defined as in Eq. (7.11)

— $r_\tau = -\bar{\tau}$ with $\bar{\tau}$ the average joint torque during one time step.

— $r_{trunc} = -1$ if the episode has been truncated, 0 otherwise.

Additionally, w_s, w_d, w_τ and w_{trunc} are positive scalar weights. Their value can be found in appendix 8.2.

4.3.3 *State*

As mentioned in part 3.1.2 the state of the RL environment is not limited to that of the robot. It must also reflect the internal state of the MPC as well as information regarding the task.

That is why, the state is composed of:

— The state of the robot $\mathbf{x}$ as defined in Eq. (7.2).

— Three nodes sampled from the current optimal trajectory given by the MPC: $\{\mathbf{x}_{\lfloor \frac{N}{3} \rfloor}, \mathbf{x}_{\lfloor \frac{2N}{3} \rfloor}, \mathbf{x}_N\}$.

— The Cartesian position of the target $\mathbf{p}_d$.

4.3.4 *Action*

As detailed in part 4.2, the RL policy only adjusts the reference posture of the MPC. This reduction in the action space of the agent speeds up training by making exploration easier. Since we are studying a specific insertion movement, we can further reduce the action space by choosing to modify the posture of the joints that move the most in this

kind of movement. As a consequence, we reduce the action to the three joints of the left arm (joint numbers 1,2 and 4 when counting from the robot's shoulder):

$$\mathbf{a} = \mathbf{q}_{arm} \in \mathbb{R}^3.$$

The output of the neural network is scaled to match the kinematic limits of the three chosen joints.

5 RESULTS

5.1 *Test of the baseline on site*

The baseline MPC approach was extensively tested in the lab and also at an Airbus site, demonstrating its deployability in various conditions. It demonstrated that the MPC was a fitting strategy to control the robot in torque, allowing physical interaction with a human during movement, as shown in Fig. 7.4. However, it became clear from the series of tests that the baseline was not a fitting strategy to reliably achieve the desired precision.

Figure 7.4 – Fine insertion task by a torque-controlled robot in an Airbus factory. The torque control allows a human to interact safely with the robot during movement.

Unfortunately, due to technical difficulties with the robot, we were unable to carry out the benchmark of the new proposed strategies on real hardware. Nonetheless, we conducted an evaluation of these control strategies in simulation.

5.2 *Evaluation Methodology*

To evaluate the performances of each method presented in part 4.2, 125 targets are uniformly sampled from the working space. The number of target reached, i. e. the tip of the tool is within 5 *mm* of the target, as well as the time to reach the target, and the final placement error, serve as the comparison criteria.

For the *RL* strategy, the algorithm we employ (SAC) generates a stochastic policy. However, during the benchmark, we use the average of the distribution as the action chosen by the policy, effectively converting it into a deterministic policy. This choice is made because repeatability is crucial in industrial applications, and exploration is not necessary during the benchmark. Additionally, it facilitates comparison with other approaches, which are also deterministic.

To provide more depth to this benchmark, we attempted to implement an end-to-end RL policy to carry out the deburring task. This policy leveraged a training architecture similar to the one designed for the approach combining RL and MPC. In this setup, the MPC was replaced by an impedance controller that followed a reference position provided by the policy. However, we were unable to obtain a policy that was sufficiently precise to reach the targets. As a consequence, we decided not to include this approach in the presented results.

It is worth noting that this test does not aim to evaluate the absolute reachability capabilities offered by the robot. To provide a more straightforward comparison of the various control strategies, the study was restricted to the case where the feet of the robot are not moving. Considering this case as representative of the absolute performance of a legged robot would be neglecting the main strength of the platform. The mobile base allows the robot to move freely to reach holes that are not reachable from its initial position.

Nonetheless, it is essential to test targets that are mechanically not reachable in our case to assess the robustness of our approach. An unreachable target should not trigger any movement that jeopardizes the system.

5.3 *Performance Improvement*

The results according to the three previously introduced criteria are summarized in table 7.1.

The first conclusion that can be drawn from these results aligns with that of [203]. Using a single fixed cost function over time is not suitable when trying to achieve precise movements with our architecture.

The proposed variable posture appears to be an easy-to-implement solution to address this issue and significantly increases the robot's reaching performance. However, it falls short of the RL-based approach and suffers from major limitations, further detailed in part 5.4, from which the latter method is exempt.

Regarding reach time and placement error, the performances are computed only on reached targets, meaning that the set of targets is different for each method, preventing

	Baseline	Posture Feedback	RL
Successes	5% (6)	50% (63)	74% (93)
Avg reach time (s)	1.2	1.9	2.1
Avg position error (mm)	3.1	2.7	2.4

Table 7.1 – Comparison of the performances of the proposed control strategies along three criteria (success, average reach time, average position error). The benchmark consists of 125 targets. Average reach time and average position error are considered only on successful attempts.

any meaningful interpretation. Indeed, the *RL* method reaches more targets, which are further away from the robot, significantly degrading average performances.

To assess the actual reach time performance, we compare the results for targets reached by both the *Posture Feedback* and *RL* methods (which represents 59 holes). The baseline method was excluded from this comparison due to its overall low success rate. Indeed, the ensemble of target reached with this approach is too small to make any meaningful statistical analysis.

	Baseline	Posture Feedback	RL
Avg reach time (s)	*N.A.*	1.92	1.86
Avg position error (mm)	*N.A.*	2.7	2.4

Table 7.2 – Comparison of the reach time and placement error of the *Variable Posture* and *RL* methods. The table summarizes results on 59 targets successfully reached by both methods.

We can conclude that the *RL* method significantly improves the number of targets reached while not affecting the overall performance of the system with respect to the other criteria. We even notice a marginal improvement over the variable posture approach in terms of both reach time and position error.

5.4 *Safety assessment*

In the context of industrial applications, it is imperative to evaluate the constraint satisfaction of each method. We categorize and examine the benchmark's failure cases in table 7.3. A **catastrophic failure** is defined as a movement during which any of the limits (position, velocity, torque, or additional simulator limits) are not satisfied. Instances where the target is not reached, but are safe to execute on the real robot, are classified as **failures**. **Successes** are defined as per part 5.2.

The RL-based approach appears to be the only method that prevents the system from triggering limits in every situation.

The other methods, which are based on MPC, do not exhibit the same results. This discrepancy can be attributed to some well documented limitations of MPC [157]. More

	Baseline	Posture Feedback	RL
Catastrophic failures	2% (2)	28% (35)	0% (0)
Failures	94% (117)	22% (27)	26% (32)
Successes	5% (6)	50% (63)	74% (93)

Table 7.3 – Constraint satisfaction of the proposed control strategies for a benchmark of 125 targets. **Catastrophic failure** means a limit has been infringed. **Failure** means the target has not been reached, but the movement is safe.

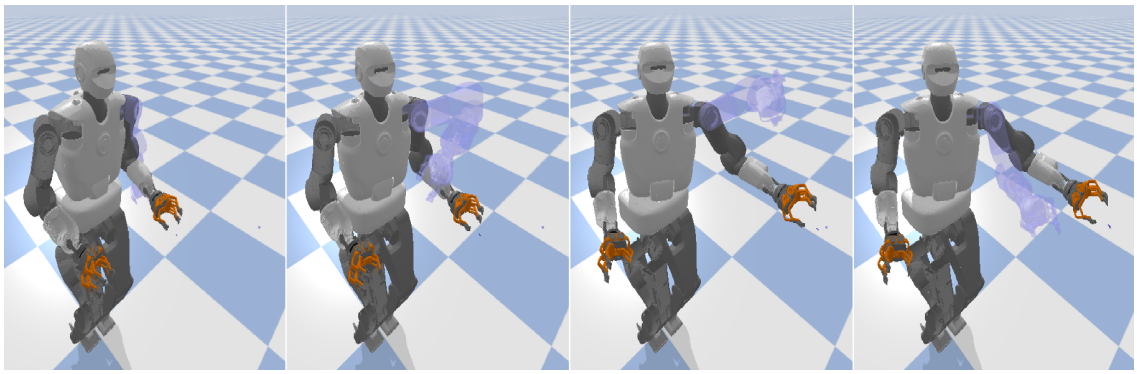


Figure 7.5 – Snapshot of the pointing movement done by the robot. The reference posture sent by the RL policy can be seen as the transparent left arm.

specifically, our approach employs penalization instead of hard constraints and utilizes an interpolation of the optimal policy with no safety guarantees to increase control frequency. While these shortcomings are not typically limiting, they become problematic in challenging tasks. Solutions have been proposed to address these issues. [125], [120] suggest methods to implement hard constraints instead of penalization. Additionally, [158] proposes an interpolation method that leverages Riccati gains while ensuring constraint satisfaction. However, the effectiveness of these methods to carry out complex tasks on humanoid robots remains to be demonstrated.

5.5 *Model Mismatch*

Not all catastrophic failures can be explained by the arguments presented in the previous section. To ensure representativeness, we incorporated additional safety constraints that mirror those implemented on the robot. However, these constraints cannot be directly integrated into the MPC and must be considered in a black-box manner.

We document the cause of failures for each method in table 7.4. Catastrophic failures caused by limits not implemented in the MPC are categorized as **unmodeled catastrophic failures**, while failures that should be prevented by the MPC are designated as **modeled catastrophic failures**.

	Baseline	Posture Feedback	RL
Unmodeled Catastrophic Failures	2% (2)	5% (6)	0% (0)
Modeled Catastrophic Failures	0% (0)	23% (29)	0% (0)

Table 7.4 – Comparison of the sources of catastrophic failures for each control structure on a benchmark of 125 targets. **Unmodeled failures** refer to those caused by limits not incorporated in the MPC, while **modeled failures** correspond to limits that are explicitly considered within the MPC.

The RL-based approach can mitigate the shortcomings of the chosen MPC implementation. Moreover, it can adapt to model mismatch to account for limits that are not explicitly formulated in the MPC.

5.6 *Movement analysis*

When examining the snapshot of the movement in Fig. 7.5, it appears that the posture transmitted by the RL policy fluctuates significantly from one time step to another. This fluctuation persists throughout the movement, even when the tool has reached its target. At first glance, this may seem like a pathological behavior, as one would anticipate the system to maintain a fixed posture and cease movement once the target is reached. However, this behavior results in only minor movements of the robot, which do not substantially increase the distance to the target and therefore are not penalized during training.

This behavior could potentially be reduced by incorporating a specific RL reward that encourages more stable solutions. But the aim of this part was to limit as much as possible the reward engineering for the RL algorithm and test what behavior would emerge with high level rewards.

Furthermore, this situation illustrates the benefits of incorporating a model-based controller within the control pipeline, rather than solely relying on RL. Even if the training has not fully converged to the optimal solution, additional performance can be gained by exploring the parameter state. This is especially relevant in the context of industrial deployment where data may be limited, and additional performance must be obtained without compromising the safety of the system.

6 DISCUSSION

6.1 *Proof of concept for hybrid RL/MPC approach*

The proposed RL/MPC approach was implemented on the robot and resulted in stable movements. However, due to issues related to the control of the wrists, the position of

the tip of the tool could not be precisely controlled. This prevented us from carrying out the benchmark in the real world. Although the approach's stability is demonstrated, the extent of performance gains observed in simulation remains uncertain for real-world applications. Nonetheless, this work serves as a compelling proof of concept for deploying reinforcement learning on delicate hardware. Exploiting the advantages of currently existing controllers appears to us as the most promising way to leverage the unmatched exploration and generalization abilities of RL in real world settings.

6.2 *Improve training performances*

Despite promising results, the training stability of the RL policy was inconsistent. For each hyperparameter set, five training sessions were conducted, and only the best was evaluated. Furthermore, our method's integration of MPC within the environment during training makes it computationally intensive. This factor is especially limiting because our controller does not run on GPU preventing us from leveraging recent advancement in GPU computation [167].

Additionally, we observed that the policy output tends to fluctuate rapidly. This behavior is primarily due to the construction of our training reward, which does not explicitly penalize high-frequency control. Such high-frequency control is not penalized by the other terms of the reward since the posture cost has a relatively low weight in the MPC. The MPC tends to filter out noise on the reference posture meaning that it will not negatively affect the performances. However, this also indicates that the policy has not fully converged.

These issues underscore the need for improved sample efficiency, which we believe could be achieved by leveraging transfer learning to bootstrap the policy. The challenging aspect is to provide a guide for the training without limiting the exploration capabilities of the RL training. That is why we believe that inputting privileged information directly into the experience buffer of the policy, as proposed in [238], could significantly speed up training.

7 CONCLUSION

This chapter extends previously demonstrated results related to the use of high frequency MPC to carry out a fine insertion task with an accuracy of few millimeters with the humanoid robot TALOS, controlled in torque. A novel RL/MPC approach is presented to improve the overall performance of the system through cost shaping. Simulation benchmark showcases the improvement in performance brought by the proposed approach, whether it is regarding target reachability or safety of the system.

In the short term, we plan to demonstrate the shown results on the robot. We also plan to continue this work by leveraging transfer learning in order to speed up the training and make it more stable.

8 APPENDIX

8.1 *Hyperparameters*

We used the SAC implementation proposed by [209]. Table 7.5 lists the chosen value for the parameters of the training:

Table 7.5 – SAC Hyperparameters

Parameter	Value
learning rate	0.0003
discount factor	0.99
replay buffer size	1000000
number of samples per minibatch	256
number of hidden layers (all networks)	2
number of hidden units per layer	256
nonlinearity	*ReLU*
target smoothing coefficient τ	0.005
target update interval	1
gradient steps	1

8.2 *Reward*

Table 7.7a details the relative weights for the MPC function that is presented in part 3.1. It is worth noting that, since the terms are not normalized, the weights also compensate discrepancies in the order of magnitude of costs.

Table 7.7b lists the weights chosen for the RL cost function Eq. (7.20).

Table 7.6 – Tables regrouping reward parameters for MPC and RL

Weight		Value
State limit	w_{x_lim}	1000
Control limit	w_{u_lim}	500
Center of mass tracking	w_{com}	500
End-effector position	w_{EF_pos}	5
End-effector orientation	w_{EF_rot}	1
End-effector velocity	w_{EF_vel}	2
Control regularization	w_{u_reg}	0.001
State regularization	w_{x_reg}	0.02

(a) MPC reward parameters

Weight		Value
Success	w_s	10
Distance to target	w_d	5
Torque regularization	w_ϕ	0.001
Failure penalization	w_{trunc}	500

(b) RL reward parameters

Part V

CONCLUSION

In this closing part, the thesis work is summarized, and the main contributions are highlighted. Immediate limitations of the work are identified, and continuation work is mentioned. Lastly, a more general opening, challenging some hypothesis on which this work is based is proposed.

8

HUMANOID ROBOTS: TOWARD THE NEXT INDUSTRIAL REVOLUTION?

In Short

This concluding chapter serves as a wrap-up of the work carried out during the thesis. It summarizes the strengths and weaknesses of the contributions and places them back in the general industrial context introduced at the beginning of this document.

Contents

1 SUMMARY

This dissertation has explored the integration of humanoid robots within industrial manufacturing operations, with a specific focus on employing a TALOS robot to perform deburring tasks on aircraft parts.

Initially, a precise insertion task was successfully executed using state-of-the-art Whole-Body Model Predictive Control. In this part, attention was drawn to the theoretical complexity of the approach and the substantial technical effort required to adapt it to new settings. Despite encountering technical challenges, experiments were successfully conducted both in the laboratory and at an Airbus site, demonstrating the feasibility of the approach on real hardware.

Following this, further investigation was carried out regarding cost shaping, a major issue with the proposed approach. First, a study confirmed that cost shaping was indeed a limiting factor for performance and indicated that adjusting the regularization cost of the MPC could help mitigate this problem.

Then, a novel approach leveraging Reinforcement Learning was introduced to reactively tune the regularization cost. This method not only significantly improved simulated performance but also showcased the ability to harness the exploration capabilities of Reinforcement Learning while retaining the guarantees provided by MPC.

Conducted within the framework of the ROB4FAM and Memmo projects, this work contributes to the ongoing efforts of the Gepetto team in designing reactive control solutions that fully exploit the capabilities of humanoid robots. The findings lay the groundwork for further advancements in autonomous reactive deburring and drilling, ultimately enhancing the role of humanoid robots in industrial settings.

2 PERSPECTIVES

The perspectives that arise from this work are of two kinds.

First, a legitimate question is to identify the next steps necessary to achieve the desired goal, which is using humanoid robots for deburring and drilling tasks in aircraft manufacturing. Knowing some major limitations of the proposed work, can we identify concrete short-term developments that can be undertaken to drive the proposed solution closer to real-world applications?

The second perspective departs from the applicative aspect and is of a broader scientific nature. It relates to the hypotheses formulated at the beginning of this work about using a torque-controlled humanoid robot animated via MPC. Reflecting upon their viability in a broader context is important for the soundness of the work that has been proposed. This is especially true since the field of humanoid robots has started attracting more attention, and new learning-based methods have demonstrated their potential.

2.1 *Follow-up work*

Follow-up work naturally emerges from some limitations of that has been presented so far. These limitations include:

— The results of Part iv were only demonstrated in simulation.

— The proposed solution is only a fraction of the full control pipeline required for autonomous work in a factory.

— The RL-MPC approach is complicated and requires extensive tuning as well as significant computational resources.

VALIDATING RESULTS ON REAL HARDWARE

Unfortunately, the performance of the approach I proposed could not be evaluated on the real robot due to technical difficulties with the control of the wrist. However, simulation is not sufficient to perfectly assess the extent to which the theoretical benefits of the proposed solution can be realized in practical applications.

Tests on real hardware are necessary to judge if the improvements observed in simulation translate to the real world. Although care has been taken to ensure that the simulator is representative of real-world conditions and the architecture rests upon a MPC that has been extensively tested before, running the new architecture on the actual robot is not a trivial task.

INTEGRATION INTO FULL CONTROL PIPELINES

Continuing in the direction elaborated in this work to get closer to real-world applications would also require extensive technical and integration work. Essentially, it would mean putting together the different solutions that have been developed in the scope of the collaboration with Airbus:

— Enhancing the robot with advanced perception capabilities. This includes using computer vision and sensor fusion techniques to provide the robot with a comprehensive understanding of its environment.

— Implementing task planning algorithms to autonomously choose and execute the appropriate high-level tasks.

— Incorporating force feedback into the control system to allow the robot to perform tasks with greater precision and safety. This is particularly important for delicate operations such as deburring and drilling.

— Unlocking loco-manipulation capabilities by integrating the proposed controller with a robust walking controller. This could enable the robot to navigate complex environments while performing manipulation tasks, enhancing its overall versatility.

The integration part is decisive in order to obtain actual solutions to real-world problems but is beyond the scope of what can be reasonably done in a research lab. These tasks need to be undertaken by industrial actors who have extensive real-world experience and can thus define appropriate characteristics for the systems to fulfill.

From a scientific standpoint, the loco-manipulation issue is especially relevant to the results shown in this document. Indeed, the controller chosen for the deburring task can also be used for bipedal locomotion [60]. Putting the two together is thus theoretically straightforward. However, it is hard to predict how the complexity of the task will influence the quality of the solution. In our case, the quality of the results might be primarily driven by the warm-start provided to the solver. Some solutions propose to leverage demonstration encoded as a memory of motion to drive the system [59], but this is still an open research question.

LEVERAGING TRANSFER LEARNING

The hybrid approach presented in this thesis proposes to conjointly exploit the knowledge contained in the models of the WBMPC and the exploration capabilities of RL. The idea is to allow the system to interactively refine the quality of the movement by interacting with the environment while guaranteeing that it is mostly stable during exploration.

However, this solution has the major drawback of being very complex. Both in terms of tuning, where RL hyperparameter tuning is added to the already cumbersome cost function shaping, and in terms of computational expenses. Indeed, this approach necessitates extensive computations during training to generate enough examples for the RL to learn, and at runtime to compute the solution of the MPC online.

The specificity of our approach is that we make the assumption that the existing controller, in our case the WBMPC, has good properties that we want to safeguard. In

an industrial setting, it makes sense to try and incrementally build upon an existing solution rather than discarding it altogether. In this context, the lever to reduce the overall complexity of our approach is to reduce the complexity of the RL layer, i.e., to make it more sample-efficient and easier to tune.

A potential solution to achieve this endeavor would be to leverage a form of transfer learning to speed up training. For example, using demonstrations that have either been generated with other control strategies [238] or, in the case of humanoid robots, that are retargeted from human movement. This could help guide the system during the initial training phase as well as requiring simpler reward functions to achieve similar objectives, thus significantly simplifying the integration of RL into existing control structures.

However, it is important to note that, in our case, it is not straightforward to find suitable demonstrations since the MPC is in the control loop.

2.2 *General prospects*

So far, the highlighted perspectives arise as a natural continuation of the work proposed in this document. However, as already mentioned, this thesis lays its foundation on two major hypotheses: regarding the architecture of the hardware studied and the control strategy used to generate motions. These hypotheses are the result of the extensive robotics experience accumulated in the Gepetto team over the years, but they are yet to be proven.

Even if all the intricacies regarding the debate that could surround these hypotheses go beyond the scope of this thesis, we can succinctly look at potential arguments that could challenge the approach adopted.

THE FUTURE OF WBMPC

Model Predictive Control has been a long-standing solution for controlling robots, with WBMPC recently emerging as a promising generic approach to fully exploit the possibilities offered by the hardware. The general trend has been that improving the models contained in these control structures is the main enabler for better performance.

However, this comes at the cost of additional complexity. These methods require extensive tuning, are difficult to master, demand significant computational power at runtime, and often limit the generality of the model that can be used.

Recently, learning-based methods have also shown interesting results and appear to alleviate some issues of MPC. While not solving the problem of tuning, they tend to be easier to implement and can be used for a wide variety of applications regardless of the underlying model. They essentially tackle model complexity by leveraging computation during training and conveniently do not require a lot of energy at runtime.

However, their widespread adoption is hindered by safety concerns associated with the lack of guarantees provided by training. Additionally, the lack of industrial examples in safety-sensitive settings such as aerospace engineering contributes to the inertia slowing down the diffusion of learning-based solutions.

This fact should, nonetheless, not prevent us from investigating the relevance of end-to-end learning methods and prompts us to better define the safety specifications to establish meaningful comparisons between these competing approaches.

THE FUTURE OF HUMANOID ROBOTS

Humanoid robots hold a lot of potential when it comes to designing more adaptable automation technologies. Their architectural proximity to humans could allow them to gradually replace human workers with minimal disturbance to existing manufacturing processes.

While the versatility of such robots is beneficial in exploratory research carried out in labs, their industrial usefulness remains to be demonstrated. Indeed, despite the ongoing trend, at the time of writing, the only notable commercial application of humanoid robots involves Agility Robotics' Digit [6].

Given that humanoid robots are yet to demonstrate actual superiority compared to other types of robots, identifying the specific roles we want them to fulfill in our society is becoming increasingly important. Defining these roles would make it more straightforward to develop performance metrics, such as those proposed by [17], to meaningfully assess their effectiveness. This would provide a rational approach to evaluate the potential of human-shaped robots.

APPENDIX

A

SYNTHÈSE EN FRANÇAIS

Sommaire

1 INTRODUCTION

Bien que faisant l'objet de recherches scientifiques depuis plusieurs décennies, les robots humanoïdes ne sont apparus que récemment comme une solution crédible pour automatiser des tâches industrielles [22]. Leur architecture pourrait leur permettre de se déplacer dans des environnements non structurés et d'exploiter facilement les spécificités d'un cadre originellement conçu pour les humains. Cette versatilité promet d'offrir des possibilités pour automatiser des tâches dangereuses ou difficiles qui étaient jusqu'à présent hors de portée des robots traditionnels.

Cependant, bien que la récente annonce du premier déploiement commercial de robots humanoïdes [6] nous rapproche d'une utilisation réelle de cette technologie dans l'industrie, toutes les promesses des robots humanoïdes ne pourront être tenues que si nous parvenons à exploiter le plein potentiel de ces plateformes. C'est pourquoi cette thèse vise à déterminer comment doter les robots humanoïdes de capacités réactives.

Elle a été réalisée au sein de l'équipe Gepetto au LAAS-CNRS et en collaboration avec *Airbus Operations SAS*. Cette collaboration s'est déroulée dans le cadre de deux projets :

— Le laboratoire joint Robot For the Future of Aircraft Manufacturing (ROB4FAM) ;

— Le projet européen H2020 Memory of Motion (Memmo).

Le cadre choisi pour cette thèse est d'étudier la pertinence d'utiliser des robots humanoïdes pour automatiser des tâches d'ébavurage et de perçage. En pratique, il s'agit d'utiliser un robot humanoïde TALOS pour effectuer des tâches d'insertion avec une grande précision (figure A.1), un premier pas vers l'automatisation complète de l'ébavurage.

Plus précisément, seront abordés la planification de mouvement et le contrôle des robots pour effectuer cette tâche. Cette thèse adopte une approche à l'intersection des méthodes de contrôle basées modèles classiques et des méthodes plus novatrices basées sur l'apprentissage. Dans un premier temps, le contrôleur corps complet choisi pour effectuer la tâche d'insertion est présenté (Partie iii). Ensuite, une méthode exploitant

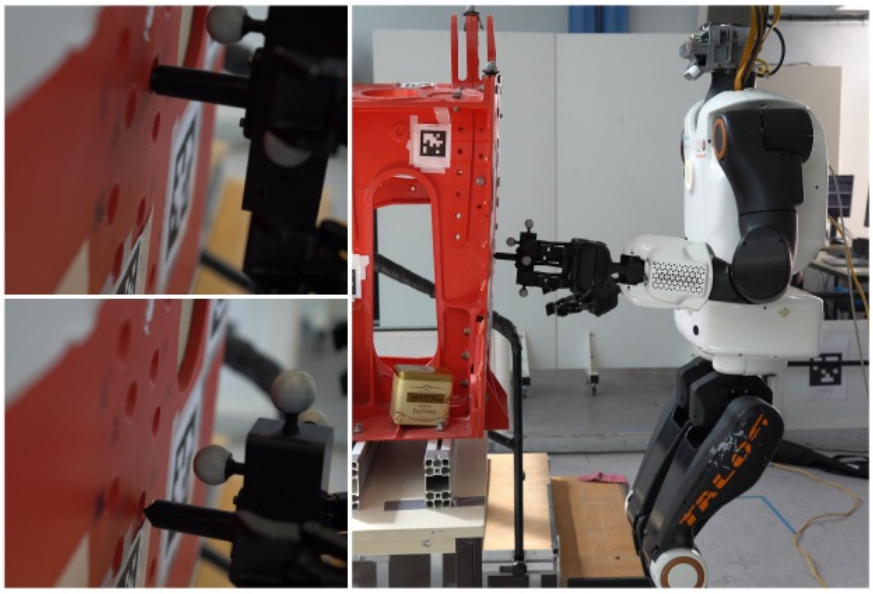


FIGURE A.1 – Insertion d'un outil dans un trou par un robot commandé à l'aide d'un contrôleur prédictif.

l'apprentissage par renforcement est adoptée pour le réglage réactif de la fonction de coût (Partie iv).

2 UN CONTRÔLEUR CORPS COMPLET GÉNÉRIQUE

La base technique de cette thèse repose sur l'utilisation d'un contrôleur prédictif corps complet. Cette stratégie de contrôle nous permet de traiter simultanément la planification de mouvement et le contrôle.

De plus les contrôleurs prédictifs disposent, comme leur nom l'indique, de capacités de prédiction qui, contrairement aux méthodes de contrôle classique, permettent d'anticiper l'évolution du système afin de garantir qu'il respecte les contraintes physiques et réalise la tâche souhaitée au mieux.

Dans notre cas, le modèle dynamique complet du corps du robot permet de rendre le contrôleur plus générique. En effet, cette stratégie peut-être adaptée à une large gamme d'architecture pour réaliser des tâches variées. De plus, les effets dynamiques des membres sont plus prononcés sur TALOS que sur d'autres robots humanoïdes, car il possède des membres plus lourds en comparaison.

Des expériences ont été menées en laboratoire, ainsi que dans un atelier d'Airbus, afin de valider l'utilisation de cette stratégie de contrôle sur le robot TALOS. Les résultats obtenus ont permis de démontrer la faisabilité technique de cette approche dans un cadre industriel réel. Toutefois, ces expériences ont également mis en lumière plusieurs limitations, en particulier liées à la quantité de réglages nécessaire pour trouver une fonction de coût approprié pour effectuer la tâche d'insertion. Les performances glo-

bales du robot ont été satisfaisante, mais perfectible, notamment en termes de rapidité d'exécution.

3 FONCTION DE COÛT RÉACTIVE

L'un des aspects critique de l'approche proposée réside dans la définition de la fonction de coût. Dans notre cas, la fonction de coût joue un rôle central, car elle permet de gérer la priorité entre les différentes tâches.

Les résultats expérimentaux ont montré que, bien que la méthode soit techniquement viable, elle restait limitée par las ajustements manuels de la fonction de coût nécessaire pour obtenir le mouvement désiré. C'est pourquoi la suite de la thèse porte sur la possibilité de mettre en place une stratégie permettant de modifier de manière réactive la fonction de coût du contrôleur.

Dans un premier temps l'impact des variations de la fonction de coût pendant la réalisation du mouvement est analysée en simulation [203]. Ensuite une méthode de planification réactive de la fonction de coût exploitant l'apprentissage par renforcement est proposée, comme présentée dans la Figure A.2.

Le recours à l'apprentissage par renforcement permet au robot d'exploiter ses expériences passées et d'élaborer une forme de mémoire de mouvement permettant d'obtenir de meilleures performances. En simulation, cette méthode utilisant à la fois l'apprentissage par renforcement et le contrôleur prédictif atteint de meilleures performances en termes de précision et de vitesse d'atteinte de la cible que la méthode classique. De plus, cette méthode permet de surmonter les défauts de modélisation de la dynamique du robot, limitation souvent décisive pour obtenir de bonnes performances dans le monde réel.

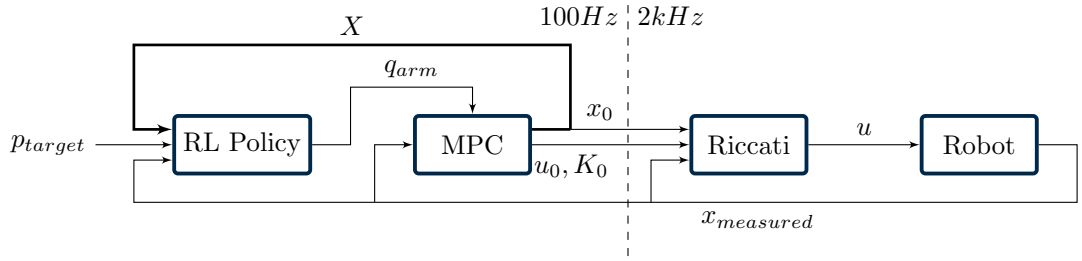


FIGURE A.2 – Structure de contrôle combinant contrôle prédictif et apprentissage par renforcement.

4 CONCLUSION

Cette thèse a exploré les défis et les opportunités liés à l'intégration des robots humanoïdes dans les environnements industriels, en se concentrant sur l'utilisation du

robot TALOS pour accomplir des tâches de précision telles que l'ébavurage. Les premiers résultats expérimentaux, obtenus grâce à l'utilisation d'un contrôleur prédictif, ont démontré la faisabilité technique de cette approche, bien que certaines limitations, notamment en termes de définition de la fonction de coût, aient été identifiées.

Face à ces limitations, l'intégration de l'apprentissage par renforcement a permis d'élaborer une méthode de planification réactive de la fonction de coût du contrôleur prédictif.

BIBLIOGRAPHY

[1] 1X, *Neo*. [Online]. Available: https://www.1x.tech/androids/neo (visited on 07/24/2024) (cit. on p. 7).

[2] P. Abbeel, A. Coates, and A. Y. Ng, « Autonomous helicopter aerobatics through apprenticeship learning », *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1608–1639, 2010. DOI: 10.1177/0278364910371999 (cit. on p. 30).

[3] C. Abdallah, D. Dawson, P. Dorato, *et al.*, « Survey of robust control for rigid robots », *IEEE Control Systems Magazine*, vol. 11, no. 2, pp. 24–30, 1991. DOI: 10.1109/37.67672 (cit. on p. 26).

[4] R. Agarwal, M. Schwarzer, P. S. Castro, *et al.*, « Deep reinforcement learning at the edge of the statistical precipice », in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, *et al.*, Eds., vol. 34, Curran Associates, Inc., 2021, pp. 29 304–29 320 (cit. on p. 32).

[5] Agility Robotics, *Agility robotics launches next generation of digit: world's first human-centric, multi-purpose robot made for logistics work*, 2023. [Online]. Available: https://agilityrobotics.com/content/agility-robotics-launches-next-generation-digit (visited on 08/15/2024) (cit. on p. 8).

[6] Agility Robotics, *Gxo signs industry-first multi-year agreement with agility robotics*, Jun. 2024. [Online]. Available: https://agilityrobotics.com/content/gxo-signs-industry-first-multi-year-agreement-with-agility-robotics (visited on 07/24/2024) (cit. on pp. 7, 107, 111).

[7] Airbus, *Airbus inaugurates new a320 structure assembly line in hamburg*, Oct. 2019. [Online]. Available: https://www.airbus.com/en/newsroom/press-releases/2019-10-airbus-inaugurates-new-a320-structure-assembly-line-in-hamburg (visited on 07/25/2024) (cit. on p. 10).

[8] Airbus, *Pioneering a robust robotics strategy*, Oct. 2023. [Online]. Available: https://www.airbus.com/en/newsroom/stories/2023-10-pioneering-a-robust-robotics-strategy (visited on 07/25/2024) (cit. on p. 10).

[9] F. Ajewole, A. Kelkar, D. Moore, *et al.*, *Unlocking the industrial potential of robotics and automation*, https://www.mckinsey.com/industries/industrials-and-electronics/our-insights/unlocking-the-industrial-potential-of-robotics-and-automation, Jan. 2023. (visited on 07/23/2024) (cit. on p. 3).

[10] J. Alvarez-Ramirez, I. Cervantes, and R. Kelly, « Pid regulation of robot manipulators: stability and performance », *Systems & Control Letters*, vol. 41, no. 2, pp. 73–83, 2000, ISSN: 0167-6911. DOI: 10.1016/S0167-6911(00)00038-4 (cit. on p. 26).

[11] A. D. Ames and M. Powell, « Towards the unification of locomotion and manipulation through control lyapunov functions and quadratic programs », in *Control of Cyber-Physical Systems: Workshop held at Johns Hopkins University, March 2013*, D. C. Tarraf, Ed. Heidelberg: Springer International Publishing, 2013, pp. 219–240, ISBN: 978-3-319-01159-2. DOI: 10.1007/978-3-319-01159-2_12 (cit. on p. 27).

[12] M. Andrychowicz, F. Wolski, A. Ray, *et al.*, « Hindsight experience replay », in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/453fadbd8a1a3af50a9df4df899537b5-Paper.pdf (cit. on p. 32).

[13] Apptronik, *Apptronik and mercedes-benz enter commercial agreement*, 2023. [Online]. Available: https://apptronik.com/news-collection/apptronik-and-mercedes-benz-enter-commercial-agreement (visited on 07/24/2024) (cit. on p. 7).

[14] Apptronik, *Apollo*. [Online]. Available: https://apptronik.com/apollo (visited on 08/15/2024) (cit. on p. 8).

[15] S. Arimoto, « Stability and robustness of pid feedback control for robot manipulators of sensory capability », in *Robotics research: First international symposium*, MIT press, 1984, pp. 783–799 (cit. on p. 26).

[16] S. H. Bang, C. A. Jové, and L. Sentis, *Rl-augmented mpc framework for agile and robust bipedal footstep locomotion planning and control*, 2024. arXiv: 2407.17683 [cs.RO] (cit. on p. 34).

[17] V. Batto, T. Flayols, N. Mansard, *et al.*, « Comparative metrics of advanced serial/parallel biped design and characterization of the main contemporary architectures », in *IEEE International Conference on Humanoid Robots*, 2023, pp. 1–7. DOI: 10.1109/Humanoids57100.2023.10375224 (cit. on p. 107).

[18] W. Bauer, M. Bender, M. Braun, *et al.*, « Lightweight robots in manual assembly–best to start simply », *Frauenhofer-Institut für Arbeitswirtschaft und Organisation IAO, Stuttgart*, vol. 1, 2016 (cit. on pp. 5, 6).

[19] F. Bečanović, V. Bonnet, R. Dumas, *et al.*, « Force sharing problem during gait using inverse optimal control », *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 872–879, 2023. DOI: 10.1109/LRA.2022.3217398 (cit. on p. 31).

[20] C. D. Bellicoso, K. Krämer, M. Stäuble, *et al.*, « Alma - articulated locomotion and manipulation for a torque-controllable robot », in *IEEE International Conference on Robotics and Automation*, 2019, pp. 8477–8483. DOI: 10.1109/ICRA.2019.8794273 (cit. on pp. 35, 82).

[21] C. C. Beltran-Hernandez, D. Petit, I. G. Ramirez-Alpizar, *et al.*, « Variable compliance control for robotic peg-in-hole assembly: a deep-reinforcement-learning approach », *Applied Sciences*, vol. 10, no. 19, 2020, ISSN: 2076-3417. DOI: 10.3390/app10196923 (cit. on p. 80).

[22] F. Berruti, D. Lewandowski, and J. Tilley, « The robot renaissance: how human-like machines are reshaping business », *Mckinsey Direct*, Mar. 2024. [Online]. Available: https://www.mckinsey.com/capabilities/operations/our-insights/the-robot-renaissance-how-human-like-machines-are-reshaping-business (visited on 07/23/2024) (cit. on pp. 5, 111).

[23] R. Bischoff, J. Kurth, G. Schreiber, *et al.*, « The kuka-dlr lightweight robot arm - a new reference platform for robotics research and manufacturing », in *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, 2010, pp. 1–8 (cit. on p. 80).

[24] J. Blackman, *"yes, the driller has to drill, but it also has to compute" - and other airbus rules for industry 4.0*, Jan. 2019. [Online]. Available: https://www.rcrwireless.com/20190124/fundamentals/the-driller-has-to-drill-but (visited on 07/25/2024) (cit. on p. 9).

[25] L. Blain, *Gr-1 general-purpose humanoid robot will carry nearly its own weight*, Jul. 2023. [Online]. Available: https://newatlas.com/robotics/fourier-gr1-humanoid-robot/ (visited on 07/24/2024) (cit. on p. 7).

[26] L. Blain, *Figure's humanoid robots are about to enter the workforce at bmw*, Jan. 2024. [Online]. Available: https://newatlas.com/robotics/figure-bmw-humanoid/ (visited on 07/24/2024) (cit. on p. 7).

[27] G. Bledt, P. M. Wensing, and S. Kim, « Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the mit cheetah », in *IEEE International Conference on Intelligent Robots and Systems*, 2017, pp. 4102–4109. DOI: 10.1109/IROS.2017.8206268 (cit. on p. 28).

[28] T. J. Böhme and B. Frank, « Direct methods for optimal control », in *Hybrid Systems, Optimal Control and Hybrid Vehicles: Theory, Methods and Applications*. Cham: Springer International Publishing, 2017, pp. 233–273, ISBN: 978-3-319-51317-1. DOI: 10.1007/978-3-319-51317-1_8 (cit. on p. 42).

[29] T. J. Böhme and B. Frank, « Indirect methods for optimal control », in *Hybrid Systems, Optimal Control and Hybrid Vehicles: Theory, Methods and Applications*. Cham: Springer International Publishing, 2017, pp. 215–231, ISBN: 978-3-319-51317-1. DOI: 10.1007/978-3-319-51317-1_7 (cit. on p. 42).

[30] R. Bonfiglioli, Y. Caraballo-Arias, and A. Salmen-Navarro, « Epidemiology of work-related musculoskeletal disorders », *Current Opinion in Epidemiology and Public Health*, vol. 1, no. 1, pp. 18–24, 2022. DOI: 10.1097/PXH.0000000000000003 (cit. on p. 4).

[31] Boston Dynamics, *An electric new era for atlas*. [Online]. Available: https://bostondynamics.com/blog/electric-new-era-for-atlas/ (visited on 07/24/2024) (cit. on pp. 7, 8).

[32] A. Bou, M. Bettini, S. Dittert, *et al.*, *Torchrl: a data-driven decision-making library for pytorch*, 2023. arXiv: 2306.00577 [cs.LG] (cit. on p. 32).

[33] K. Bouyarmane and A. Kheddar, « Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations », in *IEEE International Conference on Intelligent Robots and Systems*, 2011, pp. 4414–4419. DOI: 10.1109/IROS.2011.6094483 (cit. on p. 27).

[34] M. Brady, J. Hollerbach, T. L. Johnson, *et al.*, « Robot motion: planning and control », in Cambridge, Massachusetts: MIT press, 1982, ch. Basics of Robot motion planning and control, ISBN: 9780262021821 (cit. on p. 22).

[35] L. Brunke, M. Greeff, A. W. Hall, *et al.*, « Safe learning in robotics: from learning-based control to safe reinforcement learning », *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, no. Volume 5, 2022, pp. 411–444, 2022, ISSN: 2573-5144. DOI: 10.1146/annurev-control-042920-020211 (cit. on p. 35).

[36] D. Büchler, R. Calandra, and J. Peters, « Learning to control highly accelerated ballistic movements on muscular robots », *Robotics and Autonomous Systems*, vol. 159, p. 104 230, 2023, ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2022.104230 (cit. on p. 33).

[37] R. Budhiraja, J. Carpentier, C. Mastalli, *et al.*, « Differential dynamic programming for multi-phase rigid contact dynamics », in *IEEE International Conference on Humanoid Robots*, 2018, pp. 1–9. DOI: 10.1109/HUMANOIDS.2018.8624925 (cit. on p. 41).

[38] J. F. Canny, *The complexity of robot motion planning*. Cambridge, Massachusetts: MIT press, 1988, ISBN: 9780262031363 (cit. on p. 22).

[39] S. Caron, A. Kheddar, and O. Tempier, « Stair climbing stabilization of the hrp-4 humanoid robot using whole-body admittance control », in *IEEE International Conference on Robotics and Automation*, 2019, pp. 277–283. DOI: 10.1109/ICRA.2019.8794348 (cit. on pp. 25, 66).

[40] S. Caron, Q.-C. Pham, and Y. Nakamura, « Zmp support areas for multicontact mobility under frictional constraints », *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 67–80, 2017. DOI: 10.1109/TRO.2016.2623338 (cit. on p. 27).

[41] J. Carpentier and N. Mansard, « Multicontact locomotion of legged robots », *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1441–1460, 2018. DOI: 10.1109/TRO.2018.2862902 (cit. on p. 28).

[42] J. Carpentier, A. del Prete, S. Tonneau, *et al.*, « Multi-contact locomotion of legged robots in complex environments – the loco3d project », in *RSS Workshop on Challenges in Dynamic Legged Locomotion*, Boston, United States, Jul. 2017, 3p. [Online]. Available: https://laas.hal.science/hal-01543060 (cit. on p. 24).

[43] J. Carpentier, G. Saurel, G. Buondonno, *et al.*, « The pinocchio c++ library : a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives », in *IEEE/SICE International Symposium on System Integration (SII)*, 2019, pp. 614–619. DOI: 10.1109/SII.2019.8700380 (cit. on pp. 28, 47).

[44] J. Carpentier, S. Tonneau, M. Naveau, *et al.*, « A versatile and efficient pattern generator for generalized legged locomotion », in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 3555–3561. DOI: 10.1109/ICRA.2016.7487538 (cit. on p. 24).

[45] E. Chane-Sane, J. Amigo, T. Flayols, *et al.*, « Soloparkour: constrained reinforcement learning for visual locomotion from privileged experience », in *Conference on Robot Learning (CoRL)*, 2024 (cit. on p. 33).

[46] E. Chane-Sane, P.-A. Leziart, T. Flayols, *et al.*, *Cat: constraints as terminations for legged locomotion reinforcement learning*, 2024. arXiv: 2403.18765 [cs.RO] (cit. on pp. 33, 82).

[47] E. Chane-Sane, C. Schmid, and I. Laptev, « Goal-conditioned reinforcement learning with imagined subgoals », in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, Jul. 2021, pp. 1430–1440 (cit. on p. 32).

[48] D. Chen, B. Zhou, V. Koltun, *et al.*, « Learning by cheating », in *Proceedings of the Conference on Robot Learning*, L. P. Kaelbling, D. Kragic, and K. Sugiura, Eds., ser. Proceedings of Machine Learning Research, vol. 100, PMLR, 2020, pp. 66–75. [Online]. Available: https://proceedings.mlr.press/v100/chen20a.html (cit. on p. 33).

[49] X. Chen, J. Hu, C. Jin, *et al.*, *Understanding domain randomization for sim-to-real transfer*, 2022. arXiv: 2110.03239 [cs.LG] (cit. on p. 33).

[50] X. Chen, C. Wang, Z. Zhou, *et al.*, *Randomized ensembled double q-learning: learning fast without a model*, 2021. arXiv: 2101.05982 [cs.LG] (cit. on p. 33).

[51] C. Chi, Z. Xu, S. Feng, *et al.*, *Diffusion policy: visuomotor policy learning via action diffusion*, 2024. arXiv: 2303.04137 [cs.RO] (cit. on p. 31).

[52] Y. Choi and W. K. Chung, *PID trajectory tracking control for mechanical systems*. Springer Science & Business Media, 2004, vol. 298, ISBN: 9783540400417. DOI: 10.1007/978-3-540-40041-7 (cit. on p. 26).

[53] M. Chui, K. George, J. Manyika, *et al.*, « Human + machine: a new era of automation in manufacturing », *McKinsey & Company*, vol. 13, 2017. [Online]. Available: https://www.mckinsey.com/capabilities/operations/our-insights/human-plus-machine-a-new-era-of-automation-in-manufacturing (visited on 07/23/2024) (cit. on p. 4).

[54] W. K. Chung, L.-C. Fu, and T. Kröger, « Motion control », in *Springer handbook of robotics*, B. Siciliano and O. Khatib, Eds., Cham: Springer International Publishing, 2016, pp. 163–194, ISBN: 978-3319325507. DOI: 10.1007/978-3-319-32552-1 (cit. on p. 25).

[55] C. Collette, A. Micaelli, C. Andriot, *et al.*, « Dynamic balance control of humanoids for multiple grasps and non coplanar frictional contacts », in *IEEE International Conference on Humanoid Robots*, 2007, pp. 81–88. DOI: 10.1109/ICHR.2007.4813852 (cit. on p. 27).

[56] E. Coumans and Y. Bai, *Pybullet, a python module for physics simulation for games, robotics and machine learning*, http://pybullet.org (cit. on pp. 58, 75, 92).

[57] E. D'Elia, J.-B. Mouret, J. Kober, *et al.*, « Automatic tuning and selection of whole-body controllers », in *IEEE International Conference on Intelligent Robots and Systems*, 2022, pp. 12 935–12 941. DOI: 10.1109/IROS47612.2022.9981058 (cit. on pp. 29, 67, 87).

[58] E. Dantec, « A whole-body predictive control approach to biped locomotion », Theses, INSA de Toulouse, May 2023. [Online]. Available: https://theses.hal.science/tel-04141817 (cit. on p. 41).

[59] E. Dantec, R. Budhiraja, A. Roig, *et al.*, « Whole body model predictive control with a memory of motion: experiments on a torque-controlled talos », in *IEEE International Conference on Robotics and Automation*, 2021, pp. 8202–8208. DOI: 10.1109/ICRA48506.2021.9560742 (cit. on pp. 29, 67, 70, 82, 105).

[60] E. Dantec, M. Naveau, P. Fernbach, *et al.*, « Whole-body model predictive control for biped locomotion on a torque-controlled humanoid robot », in *IEEE International Conference on Humanoid Robots*, 2022, pp. 638–644. DOI: 10.1109/Humanoids53995.2022.10000129 (cit. on pp. 28, 29, 53, 84, 105).

[61] E. Dantec, M. Naveau, P. Fernbach, *et al.*, « Whole-body model predictive control for biped locomotion on a torque-controlled humanoid robot », in *IEEE International Conference on Humanoid Robots*, 2022, pp. 638–644. DOI: 10.1109/Humanoids53995.2022.10000129 (cit. on p. 85).

[62] E. Dantec, M. Taïx, and N. Mansard, « First order approximation of model predictive control solutions for high frequency feedback », *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4448–4455, 2022. DOI: 10.1109/LRA.2022.3149573 (cit. on pp. 47, 52, 67, 70, 86).

[63] J. Dao, H. Duan, and A. Fern, « Sim-to-real learning for humanoid box loco-manipulation », in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 16 930–16 936. DOI: 10.1109/ICRA57147.2024.10610977 (cit. on p. 82).

[64] E. Dean-Leon, J. R. Guadarrama-Olvera, F. Bergner, *et al.*, « Whole-body active compliance control for humanoid robots with robot skin », in *IEEE International Conference on Robotics and Automation*, 2019, pp. 5404–5410. DOI: 10.1109/ICRA.2019.8793258 (cit. on p. 81).

[65] W. Decre, R. Smits, H. Bruyninckx, *et al.*, « Extending itasc to support inequality constraints and non-instantaneous task specification », in *IEEE International Conference on Robotics and Automation*, 2009, pp. 964–971. DOI: 10.1109/ROBOT.2009.5152477 (cit. on p. 27).

[66] A. Del Prete, N. Mansard, O. E. Ramos, *et al.*, « Implementing torque control with high-ratio gear boxes and without joint-torque sensors », *International Journal of Humanoid Robotics*, vol. 13, no. 01, p. 1 550 044, 2016. DOI: 10.1142/S0219843615500449 (cit. on p. 25).

[67] *Développer une nouvelle méthode de contrôle des robots complexes*, 2022. [Online]. Available: https://www.horizon-europe.gouv.fr/developper-une-nouvelle-methode-de-controle-des-robots-complexes-31750 (visited on 07/29/2024) (cit. on p. 12).

[68] E. W. Dijkstra, « A note on two problems in connexion with graphs », *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959 (cit. on p. 22).

[69] A. Dömel, S. Kriegel, M. Kaßecker, *et al.*, « Toward fully autonomous mobile manipulation for industrial environments », *International Journal of Advanced Robotic Systems*, vol. 14, no. 4, p. 1 729 881 417 718 588, 2017. DOI: 10.1177/1729881417718588 (cit. on p. 80).

[70] D. Driess, F. Xia, M. S. M. Sajjadi, *et al.*, *Palm-e: an embodied multimodal language model*, 2023. arXiv: 2303.03378 [cs.LG] (cit. on p. 31).

[71] H. Duan, A. Malik, J. Dao, *et al.*, « Sim-to-real learning of footstep-constrained bipedal dynamic walking », in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 10 428–10 434. DOI: 10.1109/ICRA46639.2022.9812015 (cit. on p. 66).

[72] J. Englsberger, A. Dietrich, G.-A. Mesesan, *et al.*, « Mptc - modular passive tracking controller for stack of tasks based control frameworks », in *16th Robotics: Science and Systems, RSS 2020*, 2020. [Online]. Available: https://elib.dlr.de/136932/ (cit. on pp. 66, 77).

[73] J. Englsberger, G. Mesesan, A. Werner, *et al.*, « Torque-based dynamic walking - a long way from simulation to experiment », in *IEEE International Conference on Robotics and Automation*, 2018, pp. 440–447. DOI: 10.1109/ICRA.2018.8462862 (cit. on p. 25).

[74] J. Englsberger, A. Werner, C. Ott, *et al.*, « Overview of the torque-controlled humanoid robot toro », in *IEEE International Conference on Humanoid Robots*, 2014, pp. 916–923. DOI: 10.1109/HUMANOIDS.2014.7041473 (cit. on p. 66).

[75] A. Escande, A. Kheddar, and S. Miossec, « Planning contact points for humanoid robots », *Robotics and Autonomous Systems*, vol. 61, no. 5, pp. 428–442, 2013, ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2013.01.008 (cit. on p. 11).

[76] A. Escande, N. Mansard, and P.-B. Wieber, « Hierarchical quadratic programming: fast online humanoid-robot motion generation », *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014. DOI: 10.1177/0278364914521306 (cit. on p. 26).

[77] J. Eschmann, D. Albani, and G. Loianno, *Rltools: a fast, portable deep reinforcement learning library for continuous control*, 2024. arXiv: 2306.03530 [cs.LG] (cit. on p. 32).

[78] G. Fadini, S. Kumar, R. Kumar, *et al.*, « Co-designing versatile quadruped robots for dynamic and energy-efficient motions », *Robotica*, vol. 42, no. 6, pp. 2004–2025, 2024. DOI: 10.1017/S0263574724000730 (cit. on p. 29).

[79] F. Farshidian, E. Jelavic, A. Satapathy, *et al.*, « Real-time motion planning of legged robots: a model predictive control approach », in *IEEE International Conference on Humanoid Robots*, 2017, pp. 577–584. DOI: 10.1109/HUMANOIDS.2017.8246930 (cit. on p. 28).

[80] R. Featherstone, *Rigid Body Dynamics Algorithms*, 1st ed. Springer New York, NY, Oct. 2008, pp. IX, 272, ISBN: 978-0-387-74314-1. DOI: 10.1007/978-1-4899-7560-7 (cit. on p. 40).

[81] E. Fernandez-Camacho and C. Bordons-Alba, *Model Predictive Control in the Process Industry* (Advances in Industrial Control), 1st ed. Springer London, Dec. 1995, pp. XVIII, 239, ISBN: 978-1-4471-3010-9. DOI: 10.1007/978-1-4471-3008-6 (cit. on pp. 41, 69, 85).

[82] P. Fernbach, S. Tonneau, A. Del Prete, *et al.*, « A kinodynamic steering-method for legged multi-contact locomotion », in *IEEE International Conference on Intelligent Robots and Systems*, 2017, pp. 3701–3707. DOI: 10.1109/IROS.2017.8206217 (cit. on p. 24).

[83] E. Ferre and J.-P. Laumond, « An iterative diffusion algorithm for part disassembly », in *IEEE International Conference on Robotics and Automation*, vol. 3, 2004, pp. 3149–3154. DOI: 10.1109/ROBOT.2004.1307547 (cit. on p. 22).

[84] H. J. Ferreau, C. Kirches, A. Potschka, *et al.*, « Qpoases: a parametric active-set algorithm for quadratic programming », *Mathematical Programming Computation*, vol. 6, pp. 327–363, 2014. DOI: 10.1007/s12532-014-0071-1 (cit. on p. 27).

[85] Figure AI. [Online]. Available: https://www.figure.ai/about-us (visited on 07/24/2024) (cit. on p. 8).

[86] P. Foehn, A. Romero, and D. Scaramuzza, « Time-optimal planning for quadrotor waypoint flight », *Science Robotics*, vol. 6, no. 56, eabh1221, 2021. DOI: 10.1126/scirobotics.abh1221 (cit. on p. 27).

[87] P. I. Frazier, *A tutorial on bayesian optimization*, 2018. arXiv: 1807.02811 [stat.ML] (cit. on p. 29).

[88] Z. Fu, X. Cheng, and D. Pathak, « Deep whole-body control: learning a unified policy for manipulation and locomotion », in *Proceedings of The 6th Conference on Robot Learning*, K. Liu, D. Kulic, and J. Ichnowski, Eds., ser. Proceedings of Machine Learning Research, vol. 205, PMLR, Dec. 2023, pp. 138–149. [Online]. Available: https://proceedings.mlr.press/v205/fu23a.html (cit. on p. 82).

[89] Y. Fuchioka, Z. Xie, and M. Van de Panne, « Opt-mimic: imitation of optimized trajectories for dynamic quadruped behaviors », in *IEEE International Conference on Robotics and Automation*, 2023, pp. 5092–5098. DOI: 10.1109/ICRA48891.2023.10160562 (cit. on p. 34).

[90] S. Fujimoto, H. van Hoof, and D. Meger, « Addressing function approximation error in actor-critic methods », in *Proceedings of the 35th International Conference on Machine Learning*, J. Dy and A. Krause, Eds., ser. Proceedings of Machine Learning Research, vol. 80, PMLR, Jul. 2018, pp. 1587–1596 (cit. on p. 32).

[91] D. Gates, « Boeing abandons its failed fuselage robots on the 777x, handing the job back to machinists », *The Seattle Times*, 2019. [Online]. Available: https://www.seattletimes.com/business/boeing-aerospace/boeing-abandons-its-failed-fuselage-robots-on-the-777x-handing-the-job-back-to-machinists/ (visited on 07/25/2024) (cit. on pp. 9, 10).

[92] G. Grandesso, E. Alboni, G. P. R. Papini, *et al.*, « Cacto: continuous actor-critic with trajectory optimization—towards global optimality », *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3318–3325, 2023. DOI: 10.1109/LRA.2023.3266985 (cit. on p. 82).

[93] R. Grandia, F. Farshidian, R. Ranftl, *et al.*, « Feedback mpc for torque-controlled legged robots », in *IEEE International Conference on Intelligent Robots and Systems*, 2019, pp. 4730–4737. DOI: 10.1109/IROS40897.2019.8968251 (cit. on p. 28).

[94] R. Grandia, F. Jenelten, S. Yang, *et al.*, « Perceptive locomotion through nonlinear model-predictive control », *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3402–3421, 2023. DOI: 10.1109/TRO.2023.3275384 (cit. on p. 66).

[95] R. J. Griffin, G. Wiedebach, S. McCrory, *et al.*, « Footstep planning for autonomous walking over rough terrain », in *IEEE International Conference on Humanoid Robots*, 2019, pp. 9–16. DOI: 10.1109/Humanoids43949.2019.9035046 (cit. on p. 67).

[96] F. Grimminger, A. Meduri, M. Khadiv, *et al.*, « An open torque-controlled modular robot architecture for legged locomotion research », *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3650–3657, 2020. DOI: 10.1109/LRA.2020.2976639 (cit. on p. 25).

[97] S. Gronauer, M. Kissel, L. Sacchetto, *et al.*, « Using simulation optimization to improve zero-shot policy transfer of quadrotors », in *IEEE International Conference on Intelligent Robots and Systems*, 2022, pp. 10 170–10 176. DOI: 10.1109/IROS47612.2022.9981229 (cit. on p. 33).

[98] S. Ha, J. Lee, M. van de Panne, *et al.*, *Learning-based legged locomotion; state of the art and future perspectives*, 2024. arXiv: 2406.01152 [cs.RO] (cit. on p. 30).

[99] T. Haarnoja, A. Zhou, P. Abbeel, *et al.*, *Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor*, 2018. arXiv: 1801.01290 [cs.LG] (cit. on p. 32).

[100] T. Haarnoja, A. Zhou, K. Hartikainen, *et al.*, *Soft actor-critic algorithms and applications*, 2019. arXiv: 1812.05905 [cs.LG] (cit. on pp. 32, 88).

[101] A. Haffemayer, A. Jordana, M. Fourmy, *et al.*, « Model predictive control under hard collision avoidance constraints for a robotic arm », in *Ubiquitous Robots 2024*, Korea Robotics Society, New York (USA), United States, Jun. 2024. [Online]. Available: https://laas.hal.science/hal-04425002 (cit. on p. 29).

[102] D. Hafner, T. Lillicrap, J. Ba, *et al.*, *Dream to control: learning behaviors by latent imagination*, 2020. arXiv: 1912.01603 [cs.LG] (cit. on p. 33).

[103] M. Hägele, K. Nilsson, J. N. Pires, *et al.*, « Industrial robotics », in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Cham: Springer International Publishing, 2016, pp. 1385–1422, ISBN: 978-3-319-32552-1. DOI: 10.1007/978-3-319-32552-1_54 (cit. on p. 65).

[104] P. E. Hart, N. J. Nilsson, and B. Raphael, « A formal basis for the heuristic determination of minimum cost paths », *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968. DOI: 10.1109/TSSC.1968.300136 (cit. on p. 22).

[105] P. Henderson, R. Islam, P. Bachman, *et al.*, « Deep reinforcement learning that matters », *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018. DOI: 10.1609/aaai.v32i1.11694 (cit. on p. 32).

[106] A. Herdt, N. Perrin, and P.-B. Wieber, « Walking without thinking about it », in *IEEE International Conference on Intelligent Robots and Systems*, 2010, pp. 190–195. DOI: 10.1109/IROS.2010.5654429 (cit. on p. 28).

[107] L. Hewing, K. P. Wabersich, M. Menner, *et al.*, « Learning-based model predictive control: toward safe learning in control », *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. Volume 3, 2020, pp. 269–296, 2020, ISSN: 2573-5144. DOI: 10.1146/annurev-control-090419-075625 (cit. on p. 35).

[108] T. Hiraoka, T. Imagawa, T. Hashimoto, *et al.*, *Dropout q-functions for doubly efficient reinforcement learning*, 2022. arXiv: 2110.02034 [cs.LG] (cit. on p. 33).

[109] J. Ho and S. Ermon, « Generative adversarial imitation learning », in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, *et al.*, Eds., vol. 29, Curran Associates, Inc., 2016 (cit. on p. 30).

[110] D. Hoeller, N. Rudin, D. Sako, *et al.*, « Anymal parkour: learning agile navigation for quadrupedal robots », *Science Robotics*, vol. 9, no. 88, eadi7566, 2024. DOI: 10.1126/scirobotics.adi7566 (cit. on p. 33).

[111] N. Hogan, « Impedance control: an approach to manipulation », in *1984 American Control Conference*, 1984, pp. 304–313. DOI: 10.23919/ACC.1984.4788393 (cit. on p. 81).

[112] T. Hsia, « Adaptive control of robot manipulators - a review », in *IEEE International Conference on Robotics and Automation*, vol. 3, 1986, pp. 183–189. DOI: 10.1109/ROBOT.1986.1087696 (cit. on p. 26).

[113] S. Huang, R. F. J. Dossa, C. Ye, *et al.*, « Cleanrl: high-quality single-file implementations of deep reinforcement learning algorithms », *Journal of Machine Learning Research*, vol. 23, no. 274, pp. 1–18, 2022. [Online]. Available: http://jmlr.org/papers/v23/21-1342.html (cit. on p. 32).

[114] J. Hurst, *Enable humans to be more human*, https://youtu.be/qc8wwrdvHlc, 2022 (cit. on p. 66).

[115] J. Hwangbo, J. Lee, A. Dosovitskiy, *et al.*, « Learning agile and dynamic motor skills for legged robots », *Science Robotics*, vol. 4, no. 26, eaau5872, 2019. DOI: 10.1126/scirobotics.aau5872 (cit. on p. 33).

[116] A. Ijspeert, J. Nakanishi, and S. Schaal, « Trajectory formation for imitation with nonlinear dynamical systems », in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2, 2001, 752–757 vol.2. DOI: 10.1109/IROS.2001.976259 (cit. on p. 31).

[117] International Organization for Standardization, *ISO 15066:2016 Robots and robotic devices — Collaborative robots*, 2016. [Online]. Available: https://www.iso.org/standard/62996.html (visited on 07/24/2024) (cit. on p. 5).

[118] K. Ishihara, T. D. Itoh, and J. Morimoto, « Full-body optimal control toward versatile and agile behaviors in a humanoid robot », *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 119–126, 2020. DOI: 10.1109/LRA.2019.2947001 (cit. on p. 28).

[119] M. Jaderberg, V. Dalibard, S. Osindero, *et al.*, *Population based training of neural networks*, 2017. arXiv: 1711.09846 [cs.LG] (cit. on p. 32).

[120] W. Jallet, A. Bambade, E. Arlaud, *et al.*, « Proxddp: proximal constrained trajectory optimization », working paper or preprint, Dec. 2023. [Online]. Available: https://inria.hal.science/hal-04332348 (cit. on pp. 85, 97).

[121] W. Jallet, A. Bambade, N. Mansard, *et al.*, « Constrained differential dynamic programming: a primal-dual augmented lagrangian approach », in *IEEE International Conference on Intelligent Robots and Systems*, 2022, pp. 13 371–13 378. DOI: 10.1109/IROS47612.2022.9981586 (cit. on p. 70).

[122] W. Jallet, E. Dantec, E. Arlaud, *et al.*, « Parallel and Proximal Constrained Linear-Quadratic Methods for Real-Time Nonlinear MPC », in *Robotics: Science and Systems*, Delft, Netherlands, Jul. 2024. [Online]. Available: https://inria.hal.science/hal-04575334 (cit. on p. 29).

[123] F. Jenelten, J. He, F. Farshidian, *et al.*, « Dtc: deep tracking control », *Science Robotics*, vol. 9, no. 86, eadh5401, 2024. DOI: 10.1126/scirobotics.adh5401 (cit. on pp. 34, 82).

[124] J. Jiang, Z. Huang, Z. Bi, *et al.*, « State-of-the-art control strategies for robotic pih assembly », *Robotics and Computer-Integrated Manufacturing*, vol. 65, p. 101 894, 2020, ISSN: 0736-5845. DOI: https://doi.org/10.1016/j.rcim.2019.101894 (cit. on p. 80).

[125] A. Jordana, S. Kleff, A. Meduri, *et al.*, « Stagewise implementations of sequential quadratic programming for model-predictive control », working paper or preprint, Dec. 2023. [Online]. Available: https://laas.hal.science/hal-04330251 (cit. on pp. 85, 97).

[126] S. Kajita, F. Kanehiro, K. Kaneko, *et al.*, « Biped walking pattern generation by using preview control of zero-moment point », in *IEEE International Conference on Robotics and Automation*, vol. 2, 2003, 1620–1626 vol.2. DOI: 10.1109/ROBOT.2003.1241826 (cit. on p. 28).

[127] S. Kajita and K. Tani, « Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode », in *IEEE International Conference on Robotics and Automation*, 1991, 1405–1411 vol.2. DOI: 10.1109/ROBOT.1991.131811 (cit. on p. 27).

[128] D. Kang, J. Cheng, M. Zamora, *et al.*, « Rl + model-based control: using on-demand optimal control to learn versatile legged locomotion », *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6619–6626, 2023. DOI: 10.1109/LRA.2023.3307008 (cit. on p. 34).

[129] O. Kanoun, F. Lamiraux, P.-B. Wieber, *et al.*, « Prioritizing linear equality and inequality systems: application to local motion planning for redundant robots », in *IEEE International Conference on Robotics and Automation*, 2009, pp. 2939–2944. DOI: 10.1109/ROBOT.2009.5152293 (cit. on p. 27).

[130] S. Karaman and E. Frazzoli, « Sampling-based algorithms for optimal motion planning », *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011. DOI: 10.1177/0278364911406761 (cit. on p. 23).

[131] L. Kavraki, P. Svestka, J.-C. Latombe, *et al.*, « Probabilistic roadmaps for path planning in high-dimensional configuration spaces », *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996. DOI: 10.1109/70.508439 (cit. on p. 23).

[132] L. E. Kavraki and S. M. LaValle, « Motion planning », in *Springer handbook of robotics*, B. Siciliano and O. Khatib, Eds., Cham: Springer International Publishing, 2016, pp. 139–161, ISBN: 978-3319325507. DOI: 10.1007/978-3-319-32552-1 (cit. on p. 22).

[133] M. P. Kelly, *Transcription methods for trajectory optimization: a beginners tutorial*, 2017. arXiv: 1707.00284 [math.OC] (cit. on p. 46).

[134] R. Kelly, « Pd control with desired gravity compensation of robotic manipulators: a review », *The International Journal of Robotics Research*, vol. 16, no. 5, pp. 660–672, 1997. DOI: 10.1177/027836499701600505 (cit. on p. 26).

[135] E. C. Kerrigan, « Robust constraint satisfaction : invariant sets and predictive control », AAI28126035, Ph.D. dissertation, 2001 (cit. on p. 69).

[136] S. M. Khansari-Zadeh and A. Billard, « Learning stable nonlinear dynamical systems with gaussian mixture models », *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011. DOI: 10.1109/TRO.2011.2159412 (cit. on p. 31).

[137] O. Khatib, « A unified approach for motion and force control of robot manipulators: the operational space formulation », *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987. DOI: 10.1109/JRA.1987.1087068 (cit. on p. 26).

[138] O. Khatib, « Real-time obstacle avoidance for manipulators and mobile robots », *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986. DOI: 10.1177/027836498600500106 (cit. on p. 23).

[139] A. Kheddar, S. Caron, P. Gergondet, *et al.*, « Humanoid robots in aircraft manufacturing: the airbus use cases », *IEEE Robotics and Automation Magazine*, vol. 26, no. 4, pp. 30–45, 2019. DOI: 10.1109/MRA.2019.2943395 (cit. on pp. 66, 80).

[140] G. Kim, D. Kang, J.-H. Kim, *et al.*, *Contact-implicit mpc: controlling diverse quadruped motions without pre-planned contact modes or trajectories*, 2023. arXiv: 2312.08961 [cs.RO] (cit. on pp. 29, 84).

[141] J. Kim, H. Park, E. Ha, *et al.*, « Combined effects of noise and mixed solvents exposure on the hearing function among workers in the aviation industry », *Industrial health*, vol. 43, no. 3, pp. 567–573, 2005. DOI: 10.2486/indhealth.43.567 (cit. on p. 4).

[142] Y. Kim, H. Oh, J. Lee, *et al.*, « Not only rewards but also constraints: applications on legged robot locomotion », *IEEE Transactions on Robotics*, vol. 40, pp. 2984–3003, 2024. DOI: 10.1109/TRO.2024.3400935 (cit. on pp. 33, 82).

[143] S. Kleff, A. Meduri, R. Budhiraja, *et al.*, « High-frequency nonlinear model predictive control of a manipulator », in *IEEE International Conference on Robotics and Automation*, 2021, pp. 7330–7336. DOI: 10.1109/ICRA48506.2021.9560990 (cit. on pp. 29, 54).

[144] S. Kleff, J. Carpentier, N. Mansard, *et al.*, « On the derivation of the contact dynamics in arbitrary frames: application to polishing with talos », in *IEEE International Conference on Humanoid Robots*, 2022, pp. 512–517. DOI: 10.1109/Humanoids53995.2022.10000208 (cit. on pp. 29, 82, 85).

[145] J. Koenemann, A. Del Prete, Y. Tassa, *et al.*, « Whole-body model-predictive control applied to the hrp-2 humanoid », in *IEEE International Conference on Intelligent Robots and Systems*, 2015, pp. 3346–3351. DOI: 10.1109/IROS.2015.7353843 (cit. on p. 28).

[146] D. Kouzoupis, G. Frison, A. Zanelli, *et al.*, « Recent advances in quadratic programming algorithms for nonlinear model predictive control », *Vietnam Journal of Mathematics*, vol. 46, no. 4, pp. 863–882, 2018. DOI: 10.1007/s10013-018-0311-1 (cit. on p. 28).

[147] F. Lamiraux, J.-P. Laumond, C. Van Geem, *et al.*, « Trailer truck trajectory optimization: the transportation of components for the airbus a380 », *IEEE Robotics and Automation Magazine*, vol. 12, no. 1, pp. 14–21, 2005. DOI: 10.1109/MRA.2005.1411414 (cit. on p. 22).

[148] F. Lamiraux and J. Mirabel, « Prehensile manipulation planning: modeling, algorithms and implementation », *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2370–2388, 2022. DOI: 10.1109/TRO.2021.3130433 (cit. on p. 11).

[149] T. Lasguignes, I. Maroger, M. Fallon, *et al.*, « Icp localization and walking experiments on a talos humanoid robot », in *2021 20th International Conference on Advanced Robotics (ICAR)*, 2021, pp. 800–805. DOI: 10.1109/ICAR53236.2021.9659474 (cit. on p. 11).

[150] T. Lasguignes, G. Gobin, and O. Stasse, « Lidar-based localization system for kidnapped robots », in *2023 Seventh IEEE International Conference on Robotic Computing (IRC)*, 2023, pp. 35–42. DOI: 10.1109/IRC59093.2023.00013 (cit. on pp. 11, 14).

[151] S. LaValle, « Rapidly-exploring random trees: a new tool for path planning », *Research Report 9811*, 1998 (cit. on p. 23).

[152] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006, ISBN: 978-0521862059 (cit. on p. 23).

[153] Q. Le Lidec, F. Schramm, L. Montaut, *et al.*, « Leveraging randomized smoothing for optimal control of nonsmooth dynamical systems », *Nonlinear Analysis: Hybrid Systems*, vol. 52, p. 101 468, 2024, ISSN: 1751-570X. DOI: https://doi.org/10.1016/j.nahs.2024.101468 (cit. on p. 29).

[154] Y. LeCun, Y. Bengio, and G. Hinton, « Deep learning », *nature*, vol. 521, no. 7553, pp. 436–444, 2015. DOI: 10.1038/nature14539 (cit. on p. 31).

[155] J. Lee, L. Schroth, V. Klemm, *et al.*, *Evaluation of constrained reinforcement learning algorithms for legged locomotion*, 2023. arXiv: 2309.15430 [cs.RO] (cit. on pp. 33, 82).

[156] T. S. Lembono, C. Mastalli, P. Fernbach, *et al.*, « Learning how to walk: warm-starting optimal control solver with memory of motion », in *IEEE International Conference on Robotics and Automation*, 2020, pp. 1357–1363. DOI: 10.1109/ICRA40945.2020.9196727 (cit. on p. 86).

[157] S. Lengagne, N. Ramdani, and P. Fraisse, « Planning and fast replanning safe motions for humanoid robots », *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1095–1106, 2011. DOI: 10.1109/TRO.2011.2162998 (cit. on p. 96).

[158] H. Li and P. M. Wensing, *Cafe-mpc: a cascaded-fidelity model predictive control framework with tuning-free whole-body control*, 2024. arXiv: 2403.03995 [cs.RO] (cit. on pp. 29, 97).

[159] W. Li and E. Todorov, « Iterative linear quaqratic regulator design for nonlinear biological movement systems », in *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO,*, INSTICC, SciTePress, 2004, pp. 222–229, ISBN: 972-8865-12-0. DOI: 10.5220/0001143902220229 (cit. on p. 47).

[160] T. P. Lillicrap, J. J. Hunt, A. Pritzel, *et al.*, *Continuous control with deep reinforcement learning*, 2019. arXiv: 1509.02971 [cs.LG] (cit. on pp. 32, 89).

[161] P. Liu, D. Tateo, H. B. Ammar, *et al.*, « Robot reinforcement learning on the constraint manifold », in *Proceedings of the 5th Conference on Robot Learning*, A. Faust, D. Hsu, and G. Neumann, Eds., ser. Proceedings of Machine Learning Research, vol. 164, PMLR, Nov. 2022, pp. 1357–1366. [Online]. Available: https://proceedings.mlr.press/v164/liu22c.html (cit. on p. 35).

[162] P. Liu, K. Zhang, D. Tateo, *et al.*, « Safe reinforcement learning of dynamic high-dimensional robotic tasks: navigation, manipulation, interaction », in *IEEE International Conference on Robotics and Automation*, 2023, pp. 9449–9456. DOI: 10.1109/ICRA48891.2023.10161548 (cit. on pp. 35, 83).

[163] A. Loquercio, E. Kaufmann, R. Ranftl, *et al.*, « Learning high-speed flight in the wild », *Science Robotics*, vol. 6, no. 59, eabg5810, 2021. DOI: 10.1126/scirobotics.abg5810 (cit. on p. 33).

[164] Lozano-Perez, « Spatial planning: a configuration space approach », *IEEE Transactions on Computers*, vol. C-32, no. 2, pp. 108–120, 1983. DOI: 10.1109/TC.1983.1676196 (cit. on p. 22).

[165] A. de Luca and R. Mattone, « Sensorless robot collision detection and hybrid force/motion control », in *IEEE International Conference on Robotics and Automation*, 2005, pp. 999–1004. DOI: 10.1109/ROBOT.2005.1570247 (cit. on p. 81).

[166] Y. Ma, F. Farshidian, T. Miki, *et al.*, « Combining learning-based locomotion policy with model-based manipulation for legged mobile manipulators », *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2377–2384, 2022. DOI: 10.1109/LRA.2022.3143567 (cit. on pp. 35, 82).

[167] V. Makoviychuk, L. Wawrzyniak, Y. Guo, *et al.*, *Isaac gym: high performance gpu-based physics simulation for robot learning*, 2021. arXiv: 2108.10470 [cs.RO] (cit. on pp. 33, 99).

[168] N. Mansard, A. DelPrete, M. Geisert, *et al.*, « Using a memory of motion to efficiently warm-start a nonlinear predictive controller », in *IEEE International Conference on Robotics and Automation*, 2018, pp. 2986–2993. DOI: 10.1109/ICRA.2018.8463154 (cit. on p. 86).

[169] A. Manufacturing and Design, *First a320neo engine pylon assembled in toulouse*, 2013. [Online]. Available: https://www.aerospacemanufacturinganddesign.com/news/first-a320neo-engine-pylon-assembled-in-toulouse-111113/ (visited on 07/30/2024) (cit. on p. 15).

[170] I. Maroger, N. Ramuzat, O. Stasse, *et al.*, « Human trajectory prediction model and its coupling with a walking pattern generator of a humanoid robot », *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6361–6369, 2021. DOI: 10.1109/LRA.2021.3092750 (cit. on p. 31).

[171] S. Mason, N. Rotella, S. Schaal, *et al.*, « An mpc walking framework with external contact forces », in *IEEE International Conference on Robotics and Automation*, 2018, pp. 1785–1790. DOI: 10.1109/ICRA.2018.8461236 (cit. on p. 28).

[172] C. Mastalli, R. Budhiraja, W. Merkt, *et al.*, « Crocoddyl: an efficient and versatile framework for multi-contact optimal control », in *IEEE International Conference on Robotics and Automation*, 2020, pp. 2536–2542. DOI: 10.1109/ICRA40945.2020.9196673 (cit. on pp. 28, 29, 42, 46).

[173] C. Mastalli, S. P. Chhatoi, T. Corbéres, *et al.*, « Inverse-dynamics mpc via nullspace resolution », *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 3222–3241, 2023. DOI: 10.1109/TRO.2023.3262186 (cit. on p. 29).

[174] G. Matheron, N. Perrin, and O. Sigaud, « Understanding failures of deterministic actor-critic with continuous action spaces and sparse rewards », in *Artificial Neural Networks and Machine Learning - ICANN 2020*. Springer International Publishing, 2020, pp. 308–320, ISBN: 9783030616168. DOI: 10.1007/978-3-030-61616-8_25 (cit. on p. 34).

[175] D. Mayne, J. Rawlings, C. Rao, *et al.*, « Constrained model predictive control: stability and optimality », *Automatica*, vol. 36, no. 6, pp. 789–814, 2000, ISSN: 0005-1098. DOI: 10.1016/S0005-1098(99)00214-9 (cit. on pp. 27, 69, 86).

[176] D. Mayne, « A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems », *International Journal of Control*, vol. 3, no. 1, pp. 85–95, 1966. DOI: 10.1080/00207176608921369 (cit. on pp. 43, 85).

[177] J. Mirabel, F. Lamiraux, T. L. Ha, *et al.*, « Performing manufacturing tasks with a mobile manipulator: from motion planning to sensor based motion control », in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, 2021, pp. 159–164. DOI: 10.1109/CASE49439.2021.9551576 (cit. on pp. 11, 67, 83).

[178] V. Mnih, A. P. Badia, M. Mirza, *et al.*, « Asynchronous methods for deep reinforcement learning », in *Proceedings of The 33rd International Conference on Machine Learning*, M. F. Balcan and K. Q. Weinberger, Eds., ser. Proceedings of Machine Learning Research, vol. 48, New York, New York, USA: PMLR, Jun. 2016, pp. 1928–1937 (cit. on p. 32).

[179] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, « Human-level control through deep reinforcement learning », *nature*, vol. 518, no. 7540, pp. 529–533, 2015. DOI: 10.1038/nature14236 (cit. on p. 32).

[180] K. Mombaur, A. Truong, and J.-P. Laumond, « From human to humanoid locomotion—an inverse optimal control approach », *Autonomous robots*, vol. 28, pp. 369–383, 2010 (cit. on p. 31).

[181] M. morisawa, R. Cisneros, M. Benallegue, *et al.*, « Sequential trajectory generation for dynamic multi-contact locomotion synchronizing contact », *International Journal of Humanoid Robotics*, vol. 17, no. 01, p. 2 050 003, 2020. DOI: 10.1142/S0219843620500036 (cit. on p. 28).

[182] C. Müller, « World robotics 2023 - industrial robots », IFR Statistical Department, VDMA Services GmbH, Frankfurt am Main, Germany, Tech. Rep., 2023. [Online]. Available: https://ifr.org/img/worldrobotics/Executive_Summary_WR_Industrial_Robots_2023.pdf (visited on 07/23/2024) (cit. on p. 3).

[183] M. Murooka, K. Chappellet, A. Tanguy, *et al.*, « Humanoid loco-manipulations pattern generation and stabilization control », *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5597–5604, 2021. DOI: 10.1109/LRA.2021.3077858 (cit. on p. 28).

[184] M. Murooka, I. Kumagai, M. Morisawa, *et al.*, « Humanoid loco-manipulation planning based on graph search and reachability maps », *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1840–1847, 2021. DOI: 10.1109/LRA.2021.3060728 (cit. on p. 82).

[185] F. A. Mussa-Ivaldi, « Modular features of motor control and learning », *Current Opinion in Neurobiology*, vol. 9, no. 6, pp. 713–717, 1999, ISSN: 0959-4388. DOI: 10.1016/S0959-4388(99)00029-X (cit. on p. 31).

[186] A. Nair, B. McGrew, M. Andrychowicz, *et al.*, « Overcoming exploration in reinforcement learning with demonstrations », in *IEEE International Conference on Robotics and Automation*, 2018, pp. 6292–6299. DOI: 10.1109/ICRA.2018.8463162 (cit. on pp. 32, 34).

[187] Y. Nakamura and H. Hanafusa, « Optimal redundancy control of robot manipulators », *The International Journal of Robotics Research*, vol. 6, no. 1, pp. 32–42, 1987. DOI: 10.1177/027836498700600103 (cit. on p. 27).

[188] M. Naveau, M. Kudruss, O. Stasse, *et al.*, « A reactive walking pattern generator based on nonlinear model predictive control », *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 10–17, 2017. DOI: 10.1109/LRA.2016.2518739 (cit. on p. 28).

[189] M. Neunert, M. Stäuble, M. Giftthaler, *et al.*, « Whole-body nonlinear model predictive control through contacts for quadrupeds », *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, 2018. DOI: 10.1109/LRA.2018.2800124 (cit. on p. 28).

[190] A. Y. Ng and S. J. Russell, « Algorithms for inverse reinforcement learning », in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML '00, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 663–670, ISBN: 1558607072 (cit. on p. 31).

[191] T. D. V. Nguyen, V. Bonnet, M. Sabbah, *et al.*, « Figaroh: a python toolbox for dynamic identification and geometric calibration of robots and humans », in *IEEE International Conference on Humanoid Robots*, 2023, pp. 1–8. DOI: 10.1109/Humanoids57100.2023.10375232 (cit. on p. 56).

[192] A. Nicolin, J. Mirabel, S. Boria, *et al.*, « Agimus: a new framework for mapping manipulation motion plans to sequences of hierarchical task-based controllers », in *2020 IEEE/SICE International Symposium on System Integration (SII)*, 2020, pp. 1022–1027. DOI: 10.1109/SII46433.2020.9026288 (cit. on p. 11).

[193] A. Nicolin, « Planning of visual servoing tasks for robotics », Theses, INSA de Toulouse, Feb. 2022. [Online]. Available: https://laas.hal.science/tel-03659668 (cit. on p. 22).

[194] M. Nikolaou, « Model predictive controllers: a critical synthesis of theory and industrial needs », in ser. Advances in Chemical Engineering, vol. 26, Academic Press, 2001, pp. 131–204. DOI: 10.1016/S0065-2377(01)26003-7 (cit. on p. 27).

[195] OpenAI, I. Akkaya, M. Andrychowicz, *et al.*, *Solving rubik's cube with a robot hand*, 2019. arXiv: 1910.07113 [cs.LG] (cit. on p. 33).

[196] W. H. Organization. « MS Windows NT kernel description ». (2022), [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ageing-and-health (visited on 07/23/2024) (cit. on p. 4).

[197] A. Paolillo, T. S. Lembono, and S. Calinon, « A memory of motion for visual predictive control tasks », in *IEEE International Conference on Robotics and Automation*, 2020, pp. 9014–9020. DOI: 10.1109/ICRA40945.2020.9197216 (cit. on p. 30).

[198] G. Paolo, M. Coninx, A. Laflaquière, *et al.*, « Discovering and Exploiting Sparse Rewards in a Learned Behavior Space », *Evolutionary Computation*, pp. 1–31, Feb. 2024, ISSN: 1063-6560. DOI: 10.1162/evco_a_00343 (cit. on p. 32).

[199] A. Paraschos, C. Daniel, J. Peters, *et al.*, « Using probabilistic movement primitives in robotics », *Autonomous Robots*, vol. 42, pp. 529–551, 2018. DOI: 10.1007/s10514-017-9648-7 (cit. on p. 31).

[200] K. C. Park, P. H. Chang, and S. H. Kim, « The enhanced compact qp method for redundant manipulators using practical inequality constraints », in *IEEE International Conference on Robotics and Automation*, vol. 1, 1998, 107–114 vol.1. DOI: 10.1109/ROBOT.1998.676327 (cit. on p. 27).

[201] L. Penco, J.-B. Mouret, and S. Ivaldi, « Prescient teleoperation of humanoid robots », in *IEEE International Conference on Humanoid Robots*, 2023, pp. 1–8. DOI: 10.1109/Humanoids57100.2023.10375166 (cit. on p. 31).

[202] X. B. Peng, P. Abbeel, S. Levine, *et al.*, « Deepmimic: example-guided deep reinforcement learning of physics-based character skills », *ACM Trans. Graph.*, vol. 37, no. 4, Jul. 2018, ISSN: 0730-0301. DOI: 10.1145/3197517.3201311 (cit. on p. 34).

[203] C. Perrot and O. Stasse, « Step toward deploying the torque-controlled robot talos on industrial operations », in *International Conference on Intelligent Robots and Systems*, 2023, pp. 10 405–10 411. DOI: 10.1109/IROS55552.2023.10342428 (cit. on pp. 17, 65, 85, 95, 113).

[205] R. Peters, C. Campbell, W. Bluethmann, *et al.*, « Robonaut task learning through teleoperation », in *IEEE International Conference on Robotics and Automation*, vol. 2, 2003, 2806–2811 vol.2. DOI: 10.1109/ROBOT.2003.1242017 (cit. on p. 30).

[206] D. A. Pomerleau, « Alvinn: an autonomous land vehicle in a neural network », in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed., vol. 1, Morgan-Kaufmann, 1988. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1988/file/812b4ba287f5ee0bc9d43bbf5bbe87fb-Paper.pdf (cit. on p. 31).

[207] S. J. Qin and T. A. Badgwell, « An overview of industrial model predictive control technology », in *AIche symposium series*, New York, NY: American Institute of Chemical Engineers, 1971-c2002., vol. 93, 1997, pp. 232–256 (cit. on p. 27).

[208] I. Radosavovic, T. Xiao, B. Zhang, *et al.*, « Real-world humanoid locomotion with reinforcement learning », *Science Robotics*, vol. 9, no. 89, eadi9579, 2024. DOI: 10.1126/scirobotics.adi9579 (cit. on pp. 33, 82).

[209] A. Raffin, A. Hill, A. Gleave, *et al.*, « Stable-baselines3: reliable reinforcement learning implementations », *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html (cit. on pp. 32, 100).

[210] J. Rajamäki, K. Naderi, V. Kyrki, *et al.*, « Sampled differential dynamic programming », in *IEEE International Conference on Intelligent Robots and Systems*, 2016, pp. 1402–1409. DOI: 10.1109/IROS.2016.7759229 (cit. on p. 29).

[211] N. Ramuzat, S. Boria, and O. Stasse, « Passive inverse dynamics control using a global energy tank for torque-controlled humanoid robots in multi-contact », *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2787–2794, 2022. DOI: 10.1109/LRA.2022.3144767 (cit. on p. 11).

[212] N. Ramuzat, F. Forget, V. Bonnet, *et al.*, « Actuator model, identification and differential dynamic programming for a talos humanoid robot », in *2020 European Control Conference (ECC)*, 2020, pp. 724–730. DOI: 10.23919/ECC51009.2020.9143817 (cit. on p. 11).

[213] N. Ramuzat, O. Stasse, and S. Boria, « Benchmarking whole-body controllers on the talos humanoid robot », *Frontiers in Robotics and AI*, vol. 9, 2022, ISSN: 2296-9144. DOI: 10.3389/frobt.2022.826491 (cit. on pp. 11, 66, 77).

[214] N. Rathod, A. Bratta, M. Focchi, *et al.*, « Model predictive control with environment adaptation for legged locomotion », *IEEE Access*, vol. 9, pp. 145 710–145 727, 2021. DOI: 10.1109/ACCESS.2021.3118957 (cit. on p. 28).

[215] *Rob4fam (robots for the future of aircraft manufacturing)*. [Online]. Available: https://homepages.laas.fr/ostasse/hugo/project/rob4fam/ (visited on 07/25/2024) (cit. on p. 11).

[216] A. Romero, Y. Song, and D. Scaramuzza, *Actor-critic model predictive control*, 2024. arXiv: 2306.09852 [cs.RO] (cit. on pp. 35, 83).

[217] S. Ross, G. Gordon, and D. Bagnell, « A reduction of imitation learning and structured prediction to no-regret online learning », in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, G. Gordon, D. Dunson, and M. Dudík, Eds., ser. Proceedings of Machine Learning Research, vol. 15, Fort Lauderdale, FL, USA: PMLR, Apr. 2011, pp. 627–635 (cit. on p. 30).

[218] C. Roux, C. Perrot, and O. Stasse, « Whole-body mpc and sensitivity analysis of a real time foot step sequencer for a biped robot bolt », in *International Conference on Humanoid Robots*, 2024, pp. 467–474. DOI: 10.1109/Humanoids58906.2024.10769884 (cit. on p. 17).

[219] N. Rudin, D. Hoeller, M. Bjelonic, *et al.*, « Advanced skills by learning locomotion and local navigation end-to-end », in *IEEE International Conference on Intelligent Robots and Systems*, 2022, pp. 2497–2503. DOI: 10.1109/IROS47612.2022.9981198 (cit. on pp. 66, 78).

[220] N. Rudin, D. Hoeller, P. Reist, *et al.*, « Learning to walk in minutes using massively parallel deep reinforcement learning », in *Proceedings of the 5th Conference on Robot Learning*, A. Faust, D. Hsu, and G. Neumann, Eds., ser. Proceedings of Machine Learning Research, vol. 164, PMLR, Nov. 2022, pp. 91–100. [Online]. Available: https://proceedings.mlr.press/v164/rudin22a.html (cit. on pp. 33, 82).

[221] C. Samson, B. Espiau, and M. L. Borgne, *Robot Control: The Task Function Approach*. USA: Oxford University Press, Inc., 1991, ISBN: 0198538057 (cit. on p. 26).

[222] Sanctuary AI, *Sanctuary ai unveils phoenix - a humanoid general-purpose robot designed for work*, 2023. [Online]. Available: https://sanctuary.ai/resources/news/sanctuary-ai-unveils-phoenix-a-humanoid-general-purpose-robot-designed-for-work/ (visited on 07/24/2024) (cit. on p. 7).

[223] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, *et al.*, « Dynamic movement primitives in robotics: a tutorial survey », *The International Journal of Robotics Research*, vol. 42, no. 13, pp. 1133–1184, 2023. DOI: 10.1177/02783649231201196 (cit. on p. 31).

[224] S. Schaal, « Dynamic movement primitives -a framework for motor control in humans and humanoid robotics », in *Adaptive Motion of Animals and Machines*, H. Kimura, K. Tsuchiya, A. Ishiguro, *et al.*, Eds. Tokyo: Springer Tokyo, 2006, pp. 261–280, ISBN: 978-4-431-31381-6. DOI: 10.1007/4-431-31381-8_23. [Online]. Available: 10.1007/4-431-31381-8_23 (cit. on p. 31).

[225] G. Schoettler, A. Nair, J. Luo, *et al.*, « Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards », in *IEEE International Conference on Intelligent Robots and Systems*, 2020, pp. 5548–5555. DOI: 10.1109/IROS45743.2020.9341714 (cit. on p. 81).

[226] J. Schulman, Y. Duan, J. Ho, *et al.*, « Motion planning with sequential convex optimization and convex collision checking », *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014. DOI: 10.1177/0278364914528132 (cit. on p. 24).

[227] J. Schulman, S. Levine, P. Abbeel, *et al.*, « Trust region policy optimization », in *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, Eds., ser. Proceedings of Machine Learning Research, vol. 37, Lille, France: PMLR, Jul. 2015, pp. 1889–1897 (cit. on p. 32).

[228] J. Schulman, F. Wolski, P. Dhariwal, *et al.*, *Proximal policy optimization algorithms*, 2017. arXiv: 1707.06347 [cs.LG] (cit. on pp. 32, 88).

[229] C. Schwarke, V. Klemm, J. Tordesillas, *et al.*, *Learning quadrupedal locomotion via differentiable simulation*, 2024. arXiv: 2404.02887 [cs.RO] (cit. on p. 34).

[230] J. T. Schwartz and M. Sharir, « On the "piano movers'" problem i. the case of a two-dimensional rigid polygonal body moving amidst polygonal barriers », *Communications on pure and applied mathematics*, vol. 36, no. 3, pp. 345–398, 1983. DOI: 10.1002/cpa.3160360305 (cit. on p. 22).

[231] J. T. Schwartz and M. Sharir, « On the "piano movers" problem. ii. general techniques for computing topological properties of real algebraic manifolds », *Advances in applied Mathematics*, vol. 4, no. 3, pp. 298–351, 1983. DOI: 10.1016/0196-8858(83)90014-3 (cit. on p. 22).

[232] J. T. Schwartz and M. Sharir, « On the piano movers' problem: iii. coordinating the motion of several independent bodies: the special case of circular bodies moving amidst polygonal barriers », *The International Journal of Robotics Research*, vol. 2, no. 3, pp. 46–75, 1983. DOI: 10.1177/027836498300200304 (cit. on p. 22).

[233] J. T. Schwartz and M. Sharir, « On the piano movers' problem: v. the case of a rod moving in three-dimensional space amidst polyhedral obstacles », *Communications on Pure and Applied Mathematics*, vol. 37, no. 6, pp. 815–848, 1984. DOI: 10.1002/cpa.3160370605 (cit. on p. 22).

[234] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*. London: Springer, 2012, ISBN: 9781447104490. DOI: 10.1007/978-1-4471-0449-0 (cit. on p. 25).

[235] M. Sharir and E. Ariel-Sheffi, « On the piano movers' problem: iv. various decomposable two-dimensional motion-planning problems », *Communications on Pure and Applied Mathematics*, vol. 37, no. 4, pp. 479–493, 1984. DOI: 10.1002/cpa.3160370406 (cit. on p. 22).

[236] J. Silvério, S. Calinon, L. Rozo, *et al.*, « Learning task priorities from demonstrations », *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 78–94, 2019. DOI: 10.1109/TRO.2018.2878355 (cit. on p. 87).

[237] R. P. singh, Z. Xie, P. Gergondet, *et al.*, « Learning bipedal walking for humanoids with current feedback », *IEEE Access*, vol. 11, pp. 82 013–82 023, 2023. DOI: 10.1109/ACCESS.2023.3301175 (cit. on p. 82).

[238] L. Smith, J. C. Kew, T. Li, *et al.*, *Learning and adapting agile locomotion skills by transferring experience*, 2023. arXiv: 2304.09834 [cs.RO] (cit. on pp. 32, 99, 106).

[239] L. Smith, I. Kostrikov, and S. Levine, « Demonstrating a walk in the park: learning to walk in 20 minutes with model-free reinforcement learning », *Robotics: Science and Systems (RSS) Demo*, vol. 2, no. 3, p. 4, 2023 (cit. on pp. 33, 82).

[240] Y. Song, S. Kim, and D. Scaramuzza, *Learning quadruped locomotion using differentiable simulation*, 2024. arXiv: 2403.14864 [cs.RO] (cit. on p. 34).

[241] M. M. Sotaro Katayama and Y. Tazaki, « Model predictive control of legged and humanoid robots: models and algorithms », *Advanced Robotics*, vol. 37, no. 5, pp. 298–315, 2023. DOI: 10.1080/01691864.2023.2168134 (cit. on p. 27).

[242] O. Stasse, T. Flayols, R. Budhiraja, *et al.*, « Talos: a new humanoid research platform targeted for industrial applications », in *IEEE International Conference on Humanoid Robots*, 2017, pp. 689–695. DOI: 10.1109/HUMANOIDS.2017.8246947 (cit. on p. 13).

[243] B. Stellato, G. Banjac, P. Goulart, *et al.*, « Osqp: an operator splitting solver for quadratic programs », *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020. DOI: 10.1007/s12532-020-00179-2 (cit. on p. 27).

[244] B. J. Stephens and C. G. Atkeson, « Push recovery by stepping for humanoid robots with force controlled joints », in *IEEE International Conference on Humanoid Robots*, 2010, pp. 52–59. DOI: 10.1109/ICHR.2010.5686288 (cit. on p. 28).

[245] R. Subburaman and O. Stasse, « Delay robust model predictive control for whole-body torque control of humanoids », in *IEEE International Conference on Humanoid Robots*, 2024, pp. 600–606. DOI: 10.1109/Humanoids58906.2024.10769888 (cit. on p. 61).

[246] H. J. Suh, M. Simchowitz, K. Zhang, *et al.*, « Do differentiable simulators give better policy gradients? », in *Proceedings of the 39th International Conference on Machine Learning*, K. Chaudhuri, S. Jegelka, L. Song, *et al.*, Eds., ser. Proceedings of Machine Learning Research, vol. 162, PMLR, Jul. 2022, pp. 20 668–20 696. [Online]. Available: https://proceedings.mlr.press/v162/suh22b.html (cit. on p. 34).

[247] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018, ISBN: 0262039249 (cit. on p. 32).

[248] M. Sweney, *Elon musk claims tesla will start using humanoid robots next year*, Jul. 2024. [Online]. Available: https://www.theguardian.com/technology/article/2024/jul/23/elon-musk-tesla-humanoid-robots-optimus (visited on 07/24/2024) (cit. on p. 7).

[249] O. Takahashi and R. Schilling, « Motion planning in a plane using generalized voronoi diagrams », *IEEE Transactions on Robotics and Automation*, vol. 5, no. 2, pp. 143–150, 1989. DOI: 10.1109/70.88035 (cit. on p. 22).

[250] R. Tedrake, « Lqr-trees: feedback motion planning on sparse randomized trees », *Robotics: Science and Systems V*, 2009 (cit. on p. 67).

[251] Tesla, *Optimus - gen 2*, 2023. [Online]. Available: https://youtu.be/cpraXaw7dyc (visited on 08/15/2024) (cit. on p. 8).

[252] S. Tonneau, A. Del Prete, J. Pettré, *et al.*, « An efficient acyclic contact planner for multiped robots », *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 586–601, 2018. DOI: 10.1109/TRO.2018.2819658 (cit. on p. 24).

[253] Toyota Research Institute, *Toyota Research Institute Unveils Breakthrough in Teaching Robots New Behaviors*, 2023. [Online]. Available: https://pressroom.toyota.com/toyota-research-institute-unveils-breakthrough-in-teaching-robots-new-behaviors/ (visited on 07/24/2024) (cit. on p. 6).

[254] W. Ubellacker and A. D. Ames, « Robust locomotion on legged robots through planning on motion primitive graphs », in *IEEE International Conference on Robotics and Automation*, 2023, pp. 12 142–12 148. DOI: 10.1109/ICRA48891.2023.10160672 (cit. on p. 67).

[255] Unitree, *Unitree h1*. [Online]. Available: https://www.unitree.com/h1/ (visited on 07/24/2024) (cit. on pp. 7, 8).

[256] J. Vaillant, A. Kheddar, H. Audren, *et al.*, « Multi-contact vertical ladder climbing with an hrp-2 humanoid », *Autonomous Robots*, vol. 40, no. 3, pp. 561–580, 2016. DOI: 10.1007/s10514-016-9546-4 (cit. on pp. 11, 26).

[257] J. Van, « Two northwestern university engineers are developing cobots – machines that, unlike robots, cooperate with workers without displacing them », *Chicago Tribune*, 1996. [Online]. Available: https://peshkin.mech.northwestern.edu/cobot/chitrib/jonvan.html (visited on 07/24/2024) (cit. on p. 5).

[258] M. Vecerik, T. Hester, J. Scholz, *et al.*, *Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards*, 2018. arXiv: 1707.08817 [cs.AI] (cit. on p. 32).

[259] R. Verschueren, G. Frison, D. Kouzoupis, *et al.*, « Acados—a modular open-source framework for fast embedded optimal control », *Mathematical Programming Computation*, vol. 14, no. 1, pp. 147–183, Mar. 2022. DOI: 10.1007/s12532-021-00208-8 (cit. on p. 42).

[260] P. M. Viceconte, R. Camoriano, G. Romualdi, *et al.*, « Adherent: learning human-like trajectory generators for whole-body control of humanoid robots », *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2779–2786, 2022. DOI: 10.1109/LRA.2022.3141658 (cit. on p. 30).

[261] N. A. Villa, P. Fernbach, M. Naveau, *et al.*, « Torque controlled locomotion of a biped robot with link flexibility », in *IEEE International Conference on Humanoid Robots*, 2022, pp. 9–16. DOI: 10.1109/Humanoids53995.2022.10000135 (cit. on p. 12).

[262] A. Wächter and L. T. Biegler, « On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming », *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Mar. 2006. DOI: 10.1007/s10107-004-0559-y (cit. on p. 42).

[263] P. M. Wensing, M. Posa, Y. Hu, *et al.*, « Optimization-based control for dynamic legged robots », *IEEE Transactions on Robotics*, vol. 40, pp. 43–63, 2024. DOI: 10.1109/TRO.2023.3324580 (cit. on p. 27).

[264] D. E. Whitney *et al.*, « Quasi-static assembly of compliantly supported rigid parts », *Journal of Dynamic Systems, Measurement, and Control*, vol. 104, no. 1, pp. 65–77, 1982 (cit. on p. 81).

[265] P.-B. Wieber, « Model predictive control for biped walking », *Humanoid Robotics: A Reference*, pp. 1077–1097, 2018 (cit. on p. 28).

[266] P.-b. Wieber, « Trajectory free linear model predictive control for stable walking in the presence of strong perturbations », in *IEEE International Conference on Humanoid Robots*, 2006, pp. 137–142. DOI: 10.1109/ICHR.2006.321375 (cit. on p. 28).

[267] P.-B. Wieber, R. Tedrake, and S. Kuindersma, « Modelling and control of legged robots », in *Springer handbook of robotics*, B. Siciliano and O. Khatib, Eds., Cham: Springer International Publishing, 2016, pp. 1203–1234, ISBN: 978-3319325507. DOI: 10.1007/978-3-319-32552-1 (cit. on pp. 68, 84).

[268] D. Wierstra, T. Schaul, T. Glasmachers, *et al.*, « Natural evolution strategies », *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 949–980, 2014 (cit. on p. 29).

[269] Wikipedia, *Cfm international leap*. [Online]. Available: https://en.wikipedia.org/wiki/CFM_International_LEAP (visited on 07/30/2024) (cit. on p. 15).

[270] G. Williams, P. Drews, B. Goldfain, *et al.*, « Aggressive driving with model predictive path integral control », in *IEEE International Conference on Robotics and Automation*, 2016, pp. 1433–1440. DOI: 10.1109/ICRA.2016.7487277 (cit. on p. 29).

[271] P. Wu, A. Escontrela, D. Hafner, *et al.*, « Daydreamer: world models for physical robot learning », in *Proceedings of The 6th Conference on Robot Learning*, K. Liu, D. Kulic, and J. Ichnowski, Eds., ser. Proceedings of Machine Learning Research, vol. 205, PMLR, Dec. 2023, pp. 2226–2240 (cit. on p. 33).

[272] M. Zanon and S. Gros, « Safe reinforcement learning using robust mpc », *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3638–3652, 2021. DOI: 10.1109/TAC.2020.3024161 (cit. on pp. 35, 82).

[273] T. Zhang, Z. McCarthy, O. Jow, *et al.*, « Deep imitation learning for complex manipulation tasks from virtual reality teleoperation », in *IEEE International Conference on Robotics and Automation*, 2018, pp. 5628–5635. DOI: 10.1109/ICRA.2018.8461249 (cit. on p. 31).

[274] Z. Zhu, K. Lin, B. Dai, *et al.*, *Learning sparse rewarded tasks from sub-optimal demonstrations*, 2020. arXiv: 2004.00530 [cs.LG] (cit. on p. 32).

[275] Z. Zhu and H. Hu, « Robot learning from demonstration in robotic assembly: a survey », *Robotics*, vol. 7, no. 2, 2018, ISSN: 2218-6581. DOI: 10.3390/robotics7020017 (cit. on p. 81).

[276] J. G. Ziegler and N. B. Nichols, « Optimum settings for automatic controllers », *Transactions of the American society of mechanical engineers*, vol. 64, no. 8, pp. 759–765, 1942. DOI: 10.1115/1.4019264 (cit. on p. 26).

[277] P. Zou, Q. Zhu, J. Wu, *et al.*, « Learning-based optimization algorithms combining force control strategies for peg-in-hole assembly », in *IEEE International Conference on Intelligent Robots and Systems*, 2020, pp. 7403–7410. DOI: 10.1109/IROS45743.2020.9341678 (cit. on p. 82).

[278] M. Zucker, N. Ratliff, A. D. Dragan, *et al.*, « Chomp: covariant hamiltonian optimization for motion planning », *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013. DOI: 10.1177/0278364913488805 (cit. on p. 24).

COLOPHON

This Ph.D. thesis uses the `classicthesis` typeset by André Miede, *vielen Dank*.

Final version as of January 20, 2025

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

Université de Toulouse

Titre : Stratégie de contrôle réactif basée sur l'IA pour des robots humanoïdes industriels

Mots clés : Robots Industriels, Robots Humanoïdes, Intelligence Artificielle, Apprentissage par Renforcement, Contrôle Prédictif

Résumé : Cette thèse porte sur l'intégration de robots humanoïdes au sein des chaînes de production d'avions de ligne. Elle a été réalisée dans le cadre commun de deux projets : Robot For the Future of Aircraft Manufacturing (ROB4FAM) et le projet européen H2020 Memory of Motion (Memmo). ROB4FAM est un laboratoire joint entre Airbus Operations et l'équipe Gepetto du LAAS-CNRS. L'objectif de cette collaboration est l'étude de stratégies innovantes d'automatisation afin de réaliser des tâches, omniprésentes dans l'industrie aéronautique, de perçage et d'ébavurage. Memmo se concentre sur le
développement de solution de contrôle pour les robots. L'aboutissement de ce projet vise à obtenir des générateurs de mouvement réactifs exploitant des méthodes d'apprentissage et pouvant facilement être appliqués à une grande variété d'architecture robotiques.
Les robots humanoïdes ont récemment attiré une attention significative en raison de leur potentiel pour réaliser des tâches jusqu'alors inaccessibles aux robots. Cependant, concevoir des solutions de contrôle qui exploitent pleinement les capacités de ces systèmes représente un défi scientifique sur lequel l'équipe Gepetto se concentre depuis sa création. Le travail présenté contribue à cette problématique en étudiant le contrôle d'un robot humanoïde TALOS pour effectuer des tâches d'insertion avec une grande précision, un premier pas vers l'automatisation complète des tâches d'ébavurage et de perçage.
Dans un premier temps, un contrôleur prédictif corps-complet est déployé pour effectuer une tâche d'insertion sur un robot humanoïde contrôlé en couple. Ensuite, le problème du réglage de fonction de coût, limitant les performances rencontrées lors de la première campagne expérimentale, est étudié en avec plus d'attention en simulation. Enfin, une approche utilisant l'apprentissage par renforcement est introduite pour résoudre ce problème. La méthode proposée améliore significative les performances simulées. Elle permet d'exploiter les capacités d'exploration de l'apprentissage par renforcement tout en maintenant les garanties associées au contrôle prédictif.

Title: AI-Driven Reactive Control Strategy for Industrial Humanoid Robots

Key words: Industrial Robots, Humanoid Robots, Artificial Intelligence, Reinforcement Learning, Model Predictive Control

Abstract: This dissertation focuses on the integration of humanoid robots inside of industrial aircraft manufacturing operations. It was conducted in the context of two projects : Robot For the Future of Aircraft Manufacturing (ROB4FAM) and the European project H2020 Memory of Motion (Memmo). ROB4FAM is a joint laboratory between Airbus Operations and the LAAS-CNRS's Gepetto. Its goal is to investigate innovative automation strategies for drilling and deburring tasks, which are ubiquitous in the aeronautical industry. Memmo focuses on developing learning-based reactive motion control strategies that can easily be applied to a wide variety of robot architectures.
Humanoid robots recently have gained significant attention due to their potential to perform tasks that have been previously unattainable for robots. However, designing control solutions that fully utilize the capabilities of these systems presents a scientific challenge that has been the focus of the Gepetto team since its creation. The presented work contributes to this topic by studying the control of a humanoid robot TALOS to perform a fine insertion task, a step towards achieving autonomous reactive deburring
and drilling.
First, a state of the art Whole-Body Model Predictive Controller is used to carry out a precise insertion task on a torque-control humanoid robot. Next, the performance-restricting cost shaping issue encountered during the initial set of experiments is studied in simulation. Finally, an approach leveraging Reinforcement Learning is introduced to address this issue. The proposed method demonstrates a significant improvement in simulated performances. It allows to exploit the exploration abilities of Reinforcement Learning while maintaining the guarantees associated with Model Predictive Control.