Actuation-Aware Simplified Dynamic Models for Robotic Legged Locomotion

Romeo Orsolino

Istituto Italiano di Tecnologia, Italy
Università degli Studi di Genova, Italy

Thesis submitted for the degree of:
Doctor of Philosophy (31*st* *cycle*)

February 5, 2019

Romeo Orsolino:
Actuation-Aware Simplified Dynamic Models for Robotic Legged Locomotion
Candidate student for the *PhD Program in Bioengineering and Robotics*
Curriculum: *Advanced and Humanoid Robotics*
Genoa, Italy - December 2018

Tutor:
Dr. Claudio Semini
Dynamic Legged Systems Lab.,
Istituto Italiano di Tecnologia

Co-Tutor:
Dr. Michele Focchi
Dynamic Legged Systems Lab.,
Istituto Italiano di Tecnologia

Reviewers:
Dr. Leonardo Lanari, Associate Professor
Dipartimento di Ingegneria Informatica, Automatica e Gestionale (DIAG),
Università La Sapienza di Roma

Dr. Patrick Wensing, Assistant Professor
Dept. of Aerospace and Mechanical Engineering,
University of Notre Dame

ISTITUTO ITALIANO
DI TECNOLOGIA
DYNAMIC LEGGED SYSTEMS

This dissertation was typeset with LaTeX in Linux.
Template adapted from TU Delft template:
https://www.tudelft.nl/en/tu-delft-corporate-design/downloads/

Declaration

I hereby declare that, except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

February 5, 2019 Romeo Orsolino

Abstract

In recent years, we witnessed an ever increasing number of successful hardware implementations of motion planners for legged robots. If one common property is to be identified among these real-world applications, that is the ability of *online* planning.
Online planning is forgiving, in the sense that it allows to relentlessly compensate for external disturbances of whatever form they might be, ranging from unmodeled dynamics to external pushes or unexpected obstacles and, at the same time, follow user commands. Initially re-planning was restricted only to heuristic-based planners that exploit the low computational effort of simplified dynamic models. Such models deliberately only capture the main dynamics of the system, thus leaving to the controllers the issue of anchoring the desired trajectory to the whole body model of the robot. In recent years, however, we have seen a number of new approaches attempting to increase the accuracy of the dynamic formulation without trading-off the computational efficiency of simplified models.

In this dissertation, as an example of successful hardware implementation of heuristics and simplified model-based locomotion, I describe the framework that I developed for the generation of an omni-directional bounding gait for the HyQ quadruped robot. By analyzing the stable limit cycles for the sagittal dynamics and the *Center of Pressure (CoP)* for the lateral stabilization, the described locomotion framework is able to achieve a stable bounding while adapting to terrains of mild roughness and to sudden changes of the user desired linear and angular velocities.

The next topic reported and second contribution of this dissertation is my effort to formulate more descriptive simplified dynamic models, without trading off their computational efficiency, in order to extend the navigation capabilities of legged robots to complex geometry environments. With this in mind, I investigated the possibility of incorporating feasibility constraints in these template models and, in particular, I focused on the joint torques limits which are usually neglected at the planning stage.

In this direction, the third contribution discussed in this thesis is the formulation of the so called *actuation wrench polytope (AWP)*, defined as the set of feasible wrenches that an articulated robot can perform given its actuation limits. Interesected with the *contact wrench cone (CWC)*, this yields a new $6D$ polytope that we name *feasible wrench polytope (FWP)*, defined as the set of all wrenches that a legged robot can realize given its actuation capabilities and the friction constraints. Results are reported where, thanks to efficient computational geometry algorithms and to appropriate approximations, the FWP is employed for a *one-step receding horizon* optimization of center of mass trajectory and phase durations given a predefined step sequence on rough terrains.

For the sake of reachable workspace augmentation, I then decided to trade off the generality of the FWP formulation for a suboptimal scenario in which a quasi-static motion is assumed. This led to the definition of the, so called, *local/instantaneous actuation region* and of the *global actuation/feasible region*. They both can be seen as different variants of $2D$ linear subspaces orthogonal to gravity where the robot is guaranteed to place its own center of mass while being able to carry its own body weight given its actuation capabilities. These areas can be intersected with the well known frictional *support region*, resulting in a $2D$ linear *feasible region*, thus providing an *intuitive* tool that enables the concurrent online optimization of actuation consistent CoM trajectories and target foothold locations on rough terrains.

February 5, 2019 Romeo Orsolino

Table of Contents

List of Figures

List of Tables

List of Algorithms

Notation

Sets

$\mathbb{R}$	real numbers
$\mathbb{B}$	binary numbers
$\mathbb{N}$	natural numbers

Vector and matrices

x, α	scalars
$\mathbf{x}, \boldsymbol{\alpha}$	vectors
A, B	matrices
A^T	matrix transpose
A^{-1}	matrix inverse
$A^{\#}$	matrix pseudoinverse
$\mathbb{1}_n$	$n \times n$ identity matrix
$[\mathbf{p}]\times$	skew symmetric matrix $= \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix}$

Other symbols

n	Number of actuated joints of the system
$\mathbf{c} \in \mathbb{R}^3$	Center of Mass (CoM) position
$\mathbf{p} \in \mathbb{R}^3$	End-effector position (hand or foot)
$\mathbf{f} \in \mathbb{R}^3$	Force
$\mathbf{w} \in \mathbb{R}^6$	Wrench
$\boldsymbol{\tau} \in \mathbb{R}^n$	Joint-space torques
$\mathbf{q} \in SE(3) \times \mathbb{R}^n$	Point in the robot configurations manifold
$\dot{\mathbf{q}} \in \mathbb{R}^{6+n}$	Generalized joint-space velocity
$\mathbf{q}_i \in \mathbb{R}^{n_l}$	Joint-space positions of the actuated joints of one individual limb
N	Number of degrees of freedom of the system
n_f	Number of branches of the tree-structured robot
n_c	Number of branches of the robot in contact with the environment
n_l	Number of actuated joints of one individual limb
n_i	Degrees of motions allowed by the $i-th$ joint.
m	dimension of the contact wrench: $m = 6$ if a torque component is given, $m = 3$ otherwise
i	limb index
$\mathcal{C}_i$	Friction cone
$\mathcal{A}_i$	Actuation polytope
$\mathcal{B}_i$	Bounded friction cone: $\mathcal{B}_i = \mathcal{C}_i \cap \mathcal{A}_i$
$\mathcal{C}$	Set of friction-consistent contact wrenches $\mathbf{f}_i \in \mathbb{R}^{mn_c}$ and CoM positions $\mathbf{c}_{xy} \in \mathbb{R}^2$
$\mathcal{A}$	Set of actuation-consistent contact wrenches $\mathbf{f}_i \in \mathbb{R}^{mn_c}$ and CoM positions $\mathbf{c}_{xy} \in \mathbb{R}^2$
$\mathcal{Y}_f$	Friction region
$\mathcal{Y}_a$	Local actuation region
$\mathcal{Y}_{fa}$	Local feasible region
$\mathcal{G}_a$	Global actuation region
$\mathcal{G}_{fa}$	Global feasible region

Glossary

List of Acronyms

AWP	Actuation Wrench Polytope
CPR	Cable-driven Parallel Robots
CMM	Centroidal Momentum Matrix
CNN	Convolutional Neural Network
CoM	Center of Mass
CoP	Center of Pressure
CWC	Contact Wrench Cone
CWS	Contact Wrench Sum
DCM	Divergent Component of Motion
DoFs	Degrees of Freedom
ECMP	Enhanced Centroidal Moment Pivot
EoM	Equation of Motion
FZMP	Fictitious Zero Moment Point
FRI	Foot Rotation Indicator
FSW	Feasible Solution of Wrench
FWP	Feasible Wrench Polytope

GIW	Gravito-Inertial Wrench
HyQ	Hydraulically actuated Quadruped
ICP	Instantaneous Capture Point
IP	Iterative Projection
GRFs	Ground Reaction Forces
EoM	Equation of Motions
LPs	Linear Programs
LIP	Linear Inverted Pendulum
LLS	Lateral Leg Spring
MIPs	Mixed-Integer Programs
ML	Machine Learning
MPC	Model Predictive Control
OCP	Optimal Control Problems
ODE	Ordinary Differential Equation
ODI	Ordinary Differential Inclusion
QPs	Quadratic Programs
SEAs	Series Elastic Actuators
SIP	Sequential Iterative Projection
SLIP	Spring Loaded Inverted Pendulum
SOCP	Second Order Cone Program
SQP	Sequential Quadratic Programs
NLPs	Nonlinear Programs
RL	Reinforcement Learning
GWS	Grasp Wrench Set
TO	Trajectory Optimization
MDP	Markov Decision Process

VM	Virtual Model
VRP	Virtual Repellent Point
WFW	Wrench-Feasible Workspace
XCoM	eXtrapolated Center of Mass
ZMP	Zero Moment Point

Acknowledgements

I would like to thank my supervisor, Dr. Claudio Semini, for his assistance during the writing of this thesis and for his support throughout my stay in his lab at IIT. With his constant hard work and focus, he represents a great example for all those who, like me, wish to undertake a research career as an independent professional.

I would like to dedicate a special acknowledgement to my co-supervisor, Dr. Michele Focchi, who has been an outstanding mentor in this adventure which is my PhD. Thanks to his genuine passion for Science and dedication for self-improvement, he has motivated me to always push the limits of my understanding, to improve as a person and as a professional. He always trusted in my potentialities and let me the freedom to explore and develop my own research without, in the same time, ever making me feel alone. I could not wish to have a better mentor.

I would also like to thank all my coworkers of the Dynamic Legged Systems lab (DLS) that have shared with me a more or less long part of their time at IIT during these three years, in particular all the PostDocs who spent a considerable part of their time sharing their valuable knowledge with me: Victor Barasuol, Marco Frigerio and Andreea Radulescu. I also want to mention the PhD students and friends who have gone through the same challenges and difficulties that I had to face during my PhD: Yifu Gao, Carlos Mastalli, Marco Camurri, Josephus (Sep) Driessen, Octavio Villarreal Magaña, Shamel Fahmi, Carlos Isaac Gonzalez Bolivar, Antonios (Tony) Gkikakis and Roodra Pratap Singh Bajwa. A special mention is also required for the current and former members of the DLS lab who always dedicated a great effort and amount of their time in enabling the success of our research: José Colmenares, Felipe Polido, Alex Oleg Posatskiy, Gennaro Raiola, Fabrizio Romanelli, Lidia Furno, Marco Ronchi, Jonathan Michael Brooks, Geoff Fink and Mohamed Emara.

My thoughts are also dedicated to the current and former staff of the Advanced Robotics department of IIT, an amazing group of researchers, engineers, administrators and professionals who represent the heart of IIT and ensure every day an increasing quality of the research produced at IIT: Silvia Ivaldi, Valentina Rosso, Floriana Sardi, Simona Ventriglia, Monica Vasco, Giulia Persano, Laura Repetti, Elisa Repetti, Riccardo Sepe, Lorenzo Baccelliere, Alessio Margan, Stefano Cordasco, Paolo Guria, Francesco Di Dea, Marco Migliorini, Diego Vedelago and Mattia Arena.

Thanks to Prof. Darwin G. Caldwell for supporting the HyQ project and for his valuable suggestions and availability despite all his commitments. Thanks to Prof. Roy Featherstone for allowing me to stand by his side as an assistant during his courses of Spatial Algebra and Computational Robot Dynamics, giving me the chance to live a soft teaching experience that I had never experienced before.

A special thanks is reserved for Dr. Robert Griffin and Dr. Jerry Pratt who gave me the chance to visit the Florida Institute for Human and Machine Cognition (IHMC) during the summer 2018. This was a tremendous opportunity for me to familiarize with the research done in their group, to see how technological issues can be approached differently and how to run and coordinate a successful robotics research lab. All my thoughts go to all the researchers, interns and coworkers of the IHMC robotics lab who made my stay at IHMC great, in particular, I would like to thank George Wiedebach, Stephen McCrory, Duncan Calvert, Doug Stephen, Kenneth Yi-wen Chao, Christopher Shmidt-Wetekam, Boris van Hofslot and Brooke Layton.

Thanks to all my IIT friends who have contributed to build a small family, facing together many small and big adventures of life: Ping, Irene, Francisco, Maggy, Lisa, Amira and Dali.

I want to thank Simone for being an invaluable friend. I want to thank all my long-term friends such as Matilde, Emanuele, Anna, Giorgia, Francesco, Sonia, Julia, Nadia, Meghan, Heather, Ben, Manthan, Bea, Jordi and all the members of the Togo club: Enrico, Salman, Luigi, Matteo, Mattia, Giovanni, Nicoló, Andrea, Giacomo and Toschi. I want to thank my current and former flatmates Laura, Davide and Flavia.

In conclusion, my acknowledgements go to my parents, Elena and Roberto, who have relentlessly supported me throughout my childhood and adult life and have always put my personal fulfillment before their own priorities. To them goes my unlimited gratitude.
I want to thank all my family, especially the youngest generation made of Elena, Greta, Lorenzo, Ilaria, Edoardo, Federico and Emanuele because they represent the most beautiful motivation to be optimistic about our future.

Genova, Istituto Italiano di Tecnologia Romeo Orsolino
February 5, 2019

“Always look at the ratio between the problems you solve and the problems you cause. Then always make sure that this ratio stays strictly greater than 1.”

— *Anonymous*

Chapter 1

Introduction

Legged robots promise to invade our everyday life within a not too distant future. Collaborative robots will enter our houses to provide support to the elderly and for house keeping purposes; autonomous vehicles will deliver heavy materials in construction sites or packages at our doorbell; emergency robots will intervene in dangerous areas after natural calamities to verify the condition of precarious buildings, to flank, and maybe even replace, the first response activities of rescue teams. All these machines will need to move inside human-tailored environments such as houses and industrial plants, or inside unstructured environments like the ruins of an earthquake. In all of these scenarios, wheeled vehicles do not provide a sufficient degree of mobility because of the limited size of the obstacles that they are able to overcome. On the other hand, legged robots potentially offer the desired versatility that allows to cross human-tailored environments such as stairs or sidewalks and unstructured terrains with obstacles of variable size as well as piles of debris and more.

Humanoid robots will be especially suitable for applications that require social interaction with humans, in order to increase the person's trust and comfort. At the same time the robots' bipedal morphology will increase the system's reachable workspace compared to wheeled robots, allowing, for example, to step in and out of cars or to reach objects higher up on the kitchen shelf.

Other tasks will involve the transportation of heavy loads or the crossing of more complex terrains during an exploration task. In such applications the bipedal design will not guarantee the minimum stability and robustness level required for successfully completing the task. In such cases other platforms with a higher number of legs, like quadrupeds and hexapods, will be likely preferred to humanoid robots.

The wide range of other scenarios that I did not mention above will be probably performed in

(a) Typical after earthquake scenario

(b) Firefighter climbing up a steep stair with heavy equipment.

Figure 1-1: Disaster reponse scenarios.

collaboration with other robotic platforms such as aerial (drones, quad-copters and others), wheeled (such as autonomous cars or wheeled robots with human upper body) or caterpillars.

A large amount of industrial and house-keeping tasks will be performed by legged platforms equipped with specialized end-effectors. Such end-effectors might include wheels (for a mixed wheeled-legged locomotion on partially structured terrains), magnetic tools (for grasping tools or for climbing up metallic walls), prehensile grippers/hands and more.

In this thesis I will focus on the locomotion capabilities of legged robots with a special attention for quadrupeds. All the proposed algorithms, however, can be extended to a large variety of legged platforms with little variations. I will report the challenges and difficulties related to the implementation of locomotion algorithms on real robotic hardware, such as the presence of noisy sensor signals, the deterioration of the mechanical parts and the difficulty of properly building an internal map of the surrounding environment.

1-1 Motivation

Having a legged robot move in the real world is a complex task that requires a considerable effort from the technological point of view, especially in making sure that all the building blocks are safe and incapable of causing any damage to humans and to themselves.

From the mechanical point of view this translates in having a robust design, possibly performed in a process made of many consecutive iterations, that ensures energetic efficiency, mechanical robustness and maintainability. All electrical parts should be well isolated and possibly water proof, capable of absorbing shocks and vibrations. The dynamics of the actuators should ensure energetic efficiency, it should be accurately modeled and it should not

(a) Mountain goats walking on a rocks wall.

(b) Human climbing up a bouldering gym.

Figure 1-2: Animals and humans can reach unmatchable climbing skills compared to their robotic counterparts.

be affected, when possible, by unmodeled nonlinear phenomena such as static and viscous friction, stiction, temperature and wear. All sensors should allow repeatable measurements, they should have as little noise as possible and accurate calibration.

From the software perspective, all the components should guarantee the reading of all the sensory data without loss of packages, and all the required new motor commands should be delivered on schedule in order to avoid unexpected and potentially dangerous behaviors. All the software building blocks should then be organized in various layers of increasing complexity, with the lower layers taking care of the most basic tasks that ensure the safety of the system (*e.g.,* input hardware layer for reading sensor data and output for writing the new reference signals to the motors) and the higher layers performing task of increasing abstraction that usually may require a bigger computational burden.

In consistency with this criterion, researcher started dividing the software for locomotion of legged robots into two fundamental blocks which are the motion controller and a motion planner. The latter is usually responsible for choosing the reference motion command (joint trajectory) that all the actuators should perform in order to achieve the desired motion during a given time horizon. The former makes sure, using an appropriate feedback policy, that the current reference command is actually achieved by the actuators. The final goal of the controller and the motion planner are similar and could, theoretically, be performed by one unique block rather than two separate modules. This would correspond to optimizing the full robot description over an arbitrary time horizon and environment but it would also require, however, levels of computational efficiency that are not yet available at the time of writing of this thesis. The separation of the motion generation problem between these two blocks allows to parallelize the high-level goals such as the selection of a proper body trajectory from the platform-specific decisions such as the reference trajectory of the individual joints.

These two distinct targets impose considerably different requirements to the controller and to the motion planner. The controller may use different combinations of feedback and feedforward policies to select the value of the control signal to be sent to each actuator of the system at the next time instant. Besides that, the controller must also make sure that the reference command is delivered at the predefined frequency to the motor. Every single actuator is indeed controlled by a further low level controller (typically a PD controller) which makes sure that the joint tracks the reference position and/or force trajectory with the desired compliance. A delay in the sensors reading or in the delivery of the control signal may considerably increase the tracking error and thus jeopardize the execution of the desired motion. For this reason high-bandwidth real-time safe control is crucial for the success of the overall locomotion task.

The motion planner of a legged robot, on the other hand, must guarantee that the reference command will lead the platform to joint configurations that allow the achievement of higher level locomotion requirements such as a given forward speed provided by the external operator or the tackling of an obstacle. As a consequence, the planner will determine trajectories of the main physical quantities of the system over a larger time horizon than just the controller loop interval and, because of the resulting increase of computational burden, it may work at a lower frequency compared to the motion controller. Motion planners are usually designed using pure feed-forward policies based on the initial state of the robot; in this case the only possible implicit feedback action consists of the continuous feed-forward re-planning based on the actual states of the robot.

1-2 Contributions

This thesis discusses a number of novel contributions that have been recently published in peer-reviewed conference and journal papers of the robotics community:

1. Generation of an omni-directional bounding gait tested on HyQ by combining the fully-dynamic limit-cycle stability criterion for the sagittal plane of the robot and the semi-dynamic Center of Pressure-based criterion for the heading speed [1].
   This article has been awarded the *Best Student Paper Award* at the *20th International Conference on Climbing and Walking Robots (CLAWAR)* held in Porto, Portugal, in September 2017;

2. Projection of the torque limits of an actuated kinematic-tree robot into lower dimensional spaces (more specifically, to the space of centroidal accelerations) [2]. Thanks to this contribution it was then possible to show how to assess the feasibility (static stability and actuation consistency) of a legged robot with multiple non-coplanar contacts with the environment. Besides this I also showed how to employ the above feasibility criterion for *online* generation of trajectories for legged locomotion on complex terrains. This work has been published in *IEEE Robotics and Automation Letters (RA-L)* and is

has been presented at *IEEE International Conference on Intelligent Robots and Systems (IROS)* held in Madrid, Spain, in October 2018;

3. Achievement of a map-based *online* actuation-consistent foothold planner for static locomotion in multi-contact scenarios. In this regard, the computation of the *local actuation region* and of the *global actuation region* are described. These may also represent intuitive tools for the evaluation of the locomotion capabilities of robots in complex terrains subject to heavy payloads.

Point 1. contributes to attesting the importance of combining mathematically (and dynamically) solid motion generation strategies such as limit-cycles analysis with heuristics aimed at reducing the system's complexity.

In point 2., thanks to a mapping from joint torques to centroidal accelerations, a new approach for motion generation is implemented which is able to devise Center of Mass (CoM) trajectories that are guaranteed to be friction- and actuation-consistent.
This approach allows to further extend the above mentioned principle of separation between motion controllers and motion planners: the motion planner, in this new strategy, only optimizes *kinematic* quantities such as the CoM and the end-effectors trajectories that are guaranteed to be consistent with the actuation limits of the system (satisfy the joint torque limits) and with the configuration of the contact points. Thanks to this "exists or not" strategy, the planner does not need anymore to explicitly optimize neither the joint torques nor the contact forces, thus considerably reducing the amount of decision variables in the formulation of the Trajectory Optimization (TO) problem. Being the resulting CoM and feet trajectories verified to be feasible, it will be then the task of the motion controller to find a suitable reference set of contact forces and set of joint torques able to properly track those trajectories.

Point 3. further extends the above method by formulating a motion planner that maps the actuation torque limits to linear 2D sets of CoM feasible positions. The reduced dimensionality of the projected set, obtained by assuming static conditions (only gravity acting on the robot), provides a fast and intuitive tool for assessing the feasibility of a given static configuration.

1-3 Outline

Chapter 2 gives an overview of the most widespread approaches for the generation of motion plans for legged locomotion. The problem is divided into two subproblems which are: (a) the motion planning strategies employed for legged robots and (b) the possible stability and/or feasibility criteria used to define a stable locomotion task.
The two above topics include many different options that may be combined and merged in

different ways; Chapter 2 attempts to explain some of this possibilities and the criteria that must be considered for the definition of new formulations.
Chapter 3 focuses on a specific quadrupedal gait called bounding gait. A software framework is explained for the generation of the bounding gait on the Hydraulically actuated Quadruped (HyQ) robot, based on a mixture of heuristics and simplified models.
In Chapter 4 we discuss a simplified model that projects the feasibility constraints of the motors actuation limits into limits on the wrench space of the robot's CoM. Chapter 5 then discusses a projection of the actuators limits into CoM admissible (x, y) positions that are used in a TO implementation.
Chapter 6 concludes this dissertation by discussing and summarizing the work described and attempts to predict some future developments based on this dissertation.

Chapter 2

Related Works

This chapter describes the main existing literature regarding two key aspects of the generation of legged robots locomotion, these are the *dynamic modeling strategies* and the *stability analysis*. We divide the former in three main categories which are *heuristics*, *model-based optimization* and *data-driven approaches*; the latter topic is instead composed of multiple criteria that can be used in the motion planning formulation in order to evaluate and control the robots balancing and stabilization capabilities. The choice of the stability analysis tool is often dependent on the selected dynamic model.

In Section 2-1-1 we introduce the main challenges related to the use of heuristic strategies for motion planning for legged robots in rough terrains. Section 2-1-2 then describes the broad family of model-based optimization strategies that can be employed for tackling the same problem, these methods often are referred to under term of Trajectory Optimization (TO) and can be either based on full robot's models or on the reduced dynamic models. In Section 2-1-3 we will briefly discuss some of the state of the art Reinforcement Learning (RL)-based strategies for motion generation of legged robots [3].

2-1 Motion Planning Strategies for Legged Locomotion

In this Section we divide the main motion planning strategies for legged robot among three categories that differ in the fundamental perspectives that they assume to tackle the problem, these are *heuristics* (Section 2-1-1), *model-based motion planning* (Section 2-1-2) and *data-driven approaches* (Section 2-1-3).

Heuristics is the family of approaches that aim at having a robot perform tailored motions for a considered application, based on the researcher's experience and experimental observations.

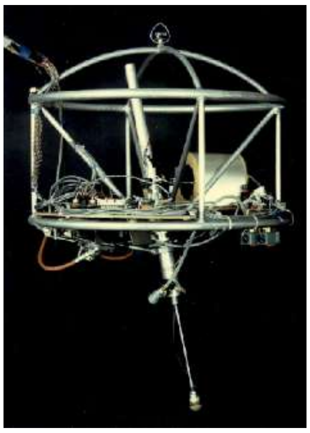
Heuristic strategies have been used for decades in the field of robotic legged locomotion and they still represent now a very useful tool for solving issues that can hardly be formalized mathematically.

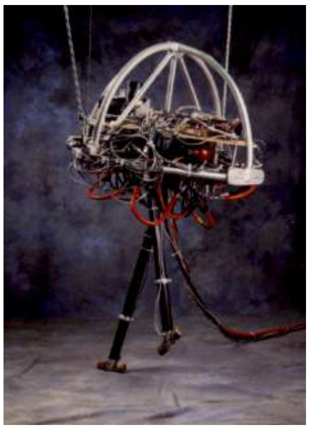
Model-based optimization represents a powerful and elegant method for the generation of motion plans for a wide variety of tasks and challenges ranging from robotic manipulation and grasping to wheeled navigation to legged locomotion in complex environments and more. Optimization is always performed using a model, this can either be a simplified model or a complete description of the physical quantities of the system; in this case, in the domain of legged locomotion, we talk about *whole-body models*. The goal of such models is to describe the motion of the joints given some actuator torque.
Depending on the resulting complexity of the overall formulation, the optimization problem can result in very different solve time and can be affected by the existence of local optimal solutions. The huge amount of possible formulations gives origin to a variety of optimization problems' families which we here roughly classify between convex and non-convex programs.

Data-driven approaches for legged robots represent a large family of strategies that aims at generating motion plans by looking at the collected data that refer to the dynamic evolution of the system in the past time intervals. This is often referred to under the broad term of *Machine Learning (ML)*, in that the algorithm attempts to avoid past mistakes and to take the best decision based on the available set of data. ML algorithms can be broadly classified between *supervised* and *unsupervised learning*. In the former case a training data is available (generated by an external evaluator) that can be used for teaching the algorithm which are the good and the bad samples. In the latter case (usually harder) no training set is provided to the algorithm and, therefore, has to figure out on its own what are the discriminant features that allow it to get as close as possible to the achievement of the task.
A successful ML approach in the motion planning setting is the Reinforcement Learning (RL) which can be seen as an extension of the theory of dynamic programming to the cases where a model of the environment is given but an analytic solution cannot be found [3]. In these cases, RL assumes a Markov Decision Process (MDP) rather than deterministic models.
The above mentioned concepts will be reported and further developed in the following Sections.

2-1-1 Heuristics

Heuristics is defined as a way of solving problems by discovering and learning things experimentally. In other words, heuristics, can be seen as pragmatic approach for solving an issue, or a subset of it, using a strategy that was experimentally noticed to work well in practice. Often heuristic methods are only partially backed by the theory and, for this reason, can be hardly generalized to different systems or to other engineering challenges.


(a) Telescopic 3D Hopper.


(b) Walking biped.

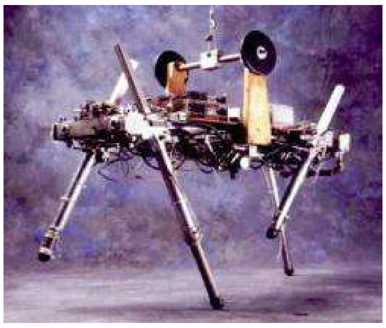

(c) Trotting quadruped.

Figure 2-1: Raibert's robots developed at the Leg Laboratory at CMU and MIT.

Every branch of engineering is filled with heuristic formulas and so is the field of legged robots locomotion. Raibert's seminal works since the early 80's can be considered a tremendous example of heuristic approaches that allowed robots to achieve unprecedented milestones [4].

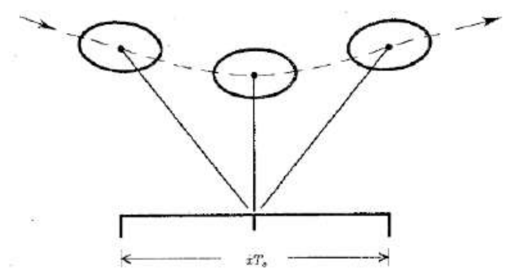
In Fig. 2-1 we can see Raibert's 3D Hopper (left), the biped (center) and quadruped (right), examples of the robots developed at the Leg Laboratory at Carnagie Mellon University (CMU), first, and at the Massachussets Institute of Technology (MIT), later, starting from 1980.
All these robots performed highly dynamics motions thanks to simple principles such as the *neutral point* [5] (see Fig. 2-2a). This can be seen as the point $\mathbf{p}_{des} \in \mathbb{R}^2$ on the flat ground where the robot should place its foot in order to cause no acceleration to its own body:

$$\mathbf{p}_{des} = \mathbf{p}_{ref} + \dot{\mathbf{c}}_{xy}\frac{\Delta t}{2} + k(\dot{\mathbf{c}}_{xy} - \dot{\mathbf{c}}_{xy,des}) \tag{2-1}$$

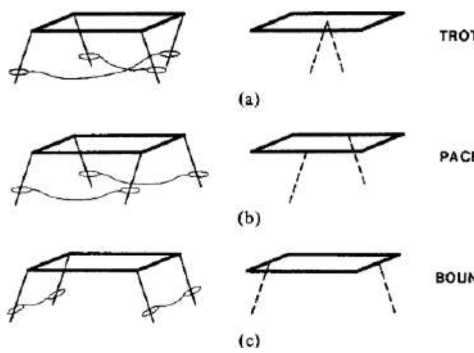
where:

- $\mathbf{p}_{ref} \in \mathbb{R}^2$ is in this case the projection of the hip position on the flat ground;
- $\dot{\mathbf{c}}_{xy} \in \mathbb{R}^2$ is the actual linear horizontal velocity of the robot;
- $\dot{\mathbf{c}}_{xy,des} \in \mathbb{R}^2$ is the desired linear horizontal velocity of the robot;
- Δt is the duration of the support phase;
- k is a gain.

This definition is straightforward for robots with one single leg such as the 3D hopper, however, in its first definition, it could not be easily generalized for multi-legged systems. Another brilliant instance of heuristics, also proposed by Raibert, was to extend the neutral point to



(a) Placing the foot in the neutral point ensures a symmetric stance phase at steady state.



(b) Virtual legs for different quadrupedal gaits for pairs of legs moving synchronously.

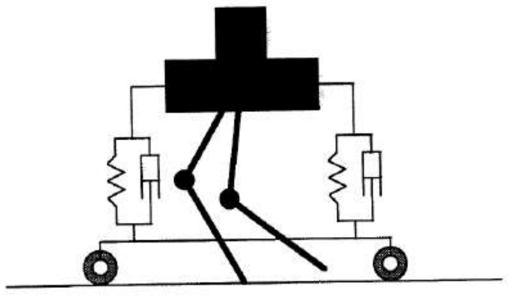
Figure 2-2: Sketch of the neutral point (left) and of the virtual leg for trot, pace and bound gaits (right).

bipedal and quadrupedal gaits where pairs of legs move synchronously and can therefore be seen as one unique leg called *virtual leg* [6]. This strategy works well for all those quadrupedal gaits, such as the trot, pace and bound, where the diagonal, rear, front or ipsilateral legs move in pairs in perfect unison and can therefore be considered as one unique leg placed in the middle (see Fig. 2-2b).

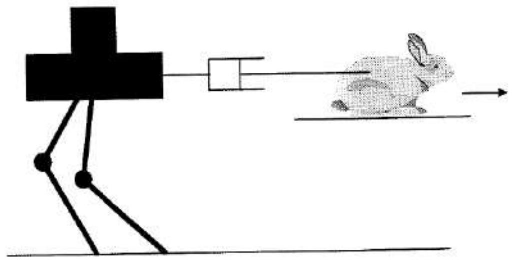
Figure 2-3: The Spring Flamingo robot used to test the virtual model (VM) control strategy.

Another example of heuristic strategy for the global stabilization of legged robots is the well known Virtual Model (VM) control introduced by Pratt *et al.* [7, 8]. This strategy consists in having a robot generate virtual forces that arise from the interaction between the robot and a virtual component. The virtual component can be a trivial spring or damper element in the operational space (task space) [9] or an more complex (nonlinear) arbitrary element. Fig. 2-3 represents the Spring Flamingo, a planar robot equipped with Series Elastic Actuators (SEAs) which was used for the VM proof of concept. As depicted in Fig. 2-4 simple virtual springs and dampers could be used to keep a desired set point height (Fig. 2-4a) or robot velocity (Fig. 2-4b).

The neutral point, the virtual leg and the virtual model are heuristic approaches used to control the global stability of the legged platform. Heuristics, however, is often also used to easily solve smaller subtasks of the locomotion process like the definition of a swing foot trajectory, the distribution of the velocities on a redundant leg (through weighting matrices) or the selection of a suitable reference robot height value. Heuristics is also, for example, the way the duration of the different phases of a movement can be chosen based on observation of biological systems (*e.g.,* stance phase duration of a running cheetah [10]) or the way the orientation of a rough terrain can be estimated [11]. Another example of heuristics can be found in the choice of the controller gains (*e.g.,* the Proportional and Derivative (PD) gains of a VM controller).



(a) A granny walker mechanism can be seen as a simple virtual model made of springs and damper elements.



(b) A virtual damper element that pulls the robot to track a given horizontal speed can be seen as chasing a hare in a dog track.

Figure 2-4: Sketch of the granny walker mechanism (left) and of the dog track bunny mechanism (right) showcasing the concept of virtual model control.

Heuristics, as we will see in the following Sections, is often combined with simplified models, model-based optimization and ML methods to simplify the system and to decouple the subtasks: the computation of the swing leg trajectory, for example, can be considered as a subtask of the walking problem and, under the assumption that the leg's inertia is negligible compared to the one of the trunk, it can be performed after that the decision of the desired footholds has been taken in a decoupled manner (*i.e.,* independently from trajectory of the Center of Mass (CoM)). Another example of this combined approach is the heuristics used for the selection of the values of the weighting matrices employed in the cost function of TO problems where each weight defines the relative priority of all the individual objectives.

As discussed above, heuristics can work extremely well and, for this reason, it is still widely used in the modern strategies for the generation of legged locomotion. In the same time, however, heuristics is strongly related to human intuition and can hardly be formalized and extended to different conditions (to more complex gaits for example). From this point of view, simplified dynamic models play their crucial role, focusing on the most relevant dynamics of the system and attempting to build a bridge between the experimental observations of the high level behaviors (captured by the heuristic policies) and the model-based optimization approaches (more theoretically grounded and, therefore, more easily generalizable).

2-1-2 Model-Based Optimization

Model-based optimization is broad a term that encompasses a large family of underdetermined mathematical formulations that share the redundancy of decision variables with respect to

the number of constraints (so called *hard constraints*) that appear in the model. The resulting existence of multiple solutions usually leads to the addition of a further objective function that discriminates the optimal solution in the set of feasible values defined by the hard constraints. When the objective function is not given the problem then turns into a *feasibility problem* where the goal is to simply find a feasible value that satisfies all the given hard constraints. Depending on the properties of the constraints (*i.e.,* the dynamic model encoded as *state equations*) and the const function to be optimized, it can be divided in various sub-families such convex and non-convex programs. Convex programs are characterized by convex objective functions and convex domains. All the programs that do not meet this requirements fall in the class on non-convex programs. Formulations with convex constraints and concave objective functions can always be transcripted into convex programs. Convex programs can, in turn, be split between as Linear Programs (LPs), Quadratic Programs (QPs) or convex Nonlinear Programs (NLPs), each of which offers different solving solutions.
More categorizations can be performed based on the following elements: continuous or discrete systems, continuous or discrete state spaces (*e.g.,* Mixed-Integer Programs (MIPs)), finite versus infinite dimensional state spaces, continuous versus discrete control sets, time-variant or time-invariant systems, controlled or uncontrolled/autonomous dynamic systems (*e.g.,* the ballistic trajectory of a robot's CoM during a flight phase), stable or unstable dynamics and, for example, deterministic versus stochastic systems [12, 13, 14].

In the large class of convex programs, the easiest type of formulation are arguably the positive definite QPs (quadratic cost) with linear equality constraints: this might be considered to be even simpler than the LPs considering that the one global optimal solution can be found *analytically.* LPs and generically constrained QPs (quadratic cost and linear equality and/or inequality constraints) add one degree of complexity because the solution can not be found in closed form but we still have the guarantee that there exists one unique optimal solution which is therefore also the global solution. LPs (linear cost functions and linear inequality and/or equality constraints) also hold no local optimum (*i.e.,* they only own one global optimum) and, if the polytope made by the constraints of the problem is bounded, then the solution lies on one of the boundaries of that polytope. Based on this property, efficient solvers can be synthesized that ensure LPs to be solved faster than any other formulation presenting a comparable amount of decision variables (*e.g.,* Dantzig's Simplex algorithm) [15, 16].

As anticipated above, both LPs and QPs fall in the larger family of Convex Programs [17] characterized by convex objective functions and convex domains of the decision variables. Thanks to the recent advancements of the hardware performances and in the theory of mathematical programming, convex programming can be considered a mature field that already presents many real world applications with the availability of various off-the-shelf solvers specialized for the different types of convex programs.
Thanks to the maturity of the convex optimization community, it has become possible to

enlarge the size of decision variables and to endow the optimization programs with model predictive capabilities by adding new variables relative to future states of the systems over a discretized time window [18]. The algorithmic and computational speed-up of modern solvers has also allowed Model Predictive Control (MPC) [19], typically applied to slow dynamical systems such as chemical plants, to enter the domain of robotic applications.

Inside the non-convex family fall all those mathematical programs made of either nonlinear objective functions or nonlinear/concave feasible sets. Although there exist nonlinear programs that are also convex, most NLPs present a complex non-convex structure that implies the existence of multiple local minima besides the global minimum. For this reason the performances of most deterministic solvers, such as Interior Point (IP) and Sequential Quadratic Programs (SQP), are affected by the initial starting point and are not guaranteed to find the global optimal solution. Stochastic solvers can represent an alternative that provides increased global convergence and reduced sensitivity to the initial guess compared to deterministic solvers at the cost of a higher solve time [20].

After this brief overview over the different types of optimization problems, we then explain the properties of the main dynamic models that appear in the modern locomotion planning research community and which are used for the formulation of Optimal Control Problems (OCP), MPC and TO programs. We will divide such dynamic models in two main classes: full and simplified models.

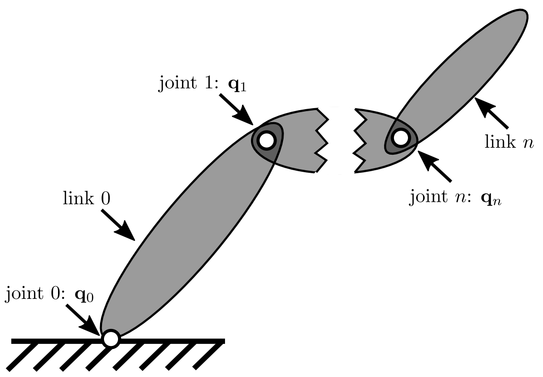
Full Dynamic Models (or *whole-body models*) are complete and accurate descriptions of the dynamic properties of the considered robotic platform. Because of the high number of involved decision variables and of the nonlinearity of the constraints (*e.g.,* nonlinearities in the end-effector Jacobian matrices, in the inertia matrix and in the Coriolis and gravity terms with respect to the joint configuration) this model is currently considered to be too complex to be used in MPC and TO settings (although it can be used in a OCP with no preview for control [21]).

Simplified Dynamic Models represent an intermediate standpoint between the practical effectiveness of heuristics and the accuracy provided by whole-body models. Simplified dynamic models, unlike heuristics, are backed up by measurements and data extracted on biological systems and can therefore, for this reason, be considered a mathematical description of the *real* underlying dynamics of the phenomenon they describe rather than a prior knowledge imposed by the developer. Many types of simplified (or reduced) models have been proposed in the past decades: in the following of this Section I will briefly list few of such models and focus on those which are most relevant for this dissertation, *i.e.,* the Linear Inverted Pendulum (LIP) and the Centroidal Dynamics.

A) **Full Dynamic Models**

Legged robots are floating articulated base kinematic trees with a large number of joints. A joint is a kinematic constraint between two bodies [22, pag. 91]; the most employed types of mechanical joints for modern humanoid and quadruped robots are prismatic and revolute joints, that means that the number of motion freedoms $n_i \in [0, 6]$ allowed by every single $i - th$ joint is one ($n_i = 1$) apart from the virtual base joint ($i = 0$) which does not constrain any motion ($n_0 = 6$). The virtual base joint is indeed a fictitious joint with six degrees of motion that connects the trunk of the legged robot (also called *floating base* link) to the world frame. As a consequence, for most of the legged robots developed in the modern robotics community, the total number of Degrees of Freedom (DoFs) of the system N corresponds to the number of actuated joints n plus the six un-actuated DoFs of the floating base:

$$N = \sum_{i=0}^{n} n_i = n + 6 \tag{2-2}$$



(a) Fixed-base serial mechanical chain.

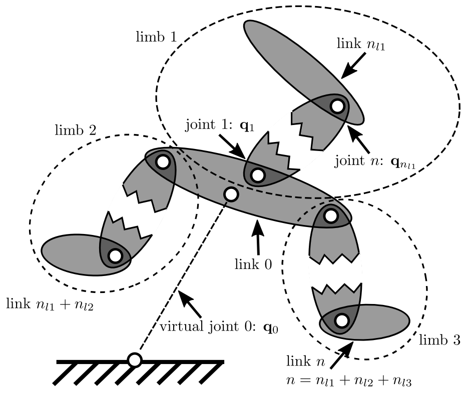

limb 1
link n_{l1}
limb 2
joint 1: $\mathbf{q}_1$
joint n: $\mathbf{q}_{n_{l1}}$
link 0
link $n_{l1} + n_{l2}$
virtual joint 0: $\mathbf{q}_0$
limb 3
link n
$n = n_{l1} + n_{l2} + n_{l3}$
(b) Floating-base tree-structured mechanical chain.

Figure 2-5: Coordinate frames and nomunclature used to describe articulated serial and tree-structured kinematic chains.

The position of every individual joint of the system is fully described by a vector $\mathbf{q}_i \in \mathbb{R}^{n_i}$ which describes its position and orientation relative to the frame of the parent joint. As mentioned above, most modern robots are made of joints that only allow one motion, meaning that their position can be described using one scalar coefficient q_i with the exception of the virtual base joint $\mathbf{q}_0$ (or $\mathbf{q}_b$). This can therefore be described by a linear position $\mathbf{x} \in \mathbb{R}^3$ and an angular orientation $\mathbf{R} \in SO(3)$ with respect to the world frame:

$$\mathbf{q}_0 = \mathbf{q}_b = (\mathbf{x}, \mathbf{R}) \in SE(3) \tag{2-3}$$

where $SE(3) = \mathbb{R}^3 \times SO(3)$ is the special Euclidean group and $SO(3)$ is the special Orthogonal group[1]. Elements of the $SO(3)$ group can be described in multiple ways such as Euler angles

[1]The special Orthogonal group refers to all the 3×3 matrices $\mathbf{R}$ such that: $det(\mathbf{R}) = \pm 1$ and $\mathbf{R}^{-1} = \mathbf{R}^T$

$\boldsymbol{\theta} \in \mathbb{R}^3$ or the Euler parameters, or quaternions, $\boldsymbol{\eta} \in \mathbb{R}^4$: the former represents a minimum set of parameters (*i.e.,* a *minimal* representation) needed to describe an orientation but is prone to singularities; the latter, on the other hand, is a singularity-free redundant set of parameters but it cannot be directly obtained by integration of the angular velocity and, for this reason, adds a further complexity in the computation because of the resulting different number of elements in the position $\mathbf{q}_b$ and in the velocity vector $\dot{\mathbf{q}}_b$ [23, pag. 41]. Alternatively, angular orientations can also be represented by means of rotation matrices and exponential maps [24].

The canonical form of the dynamic equation of motion of the full robot (*i.e., whole-body model*) can be obtained by computing the Lagrangian of the system (sum of the kinetic and potential energy terms of each link) [15, 25]:

$$L(\mathbf{q}, \dot{\mathbf{q}}) = T(\mathbf{q}, \dot{\mathbf{q}}) - U(\mathbf{q}) \tag{2-4}$$

where T represents the total kinetic energy and U the total potential energy of the system. The vector $\mathbf{q} = [\mathbf{q}_b^T, q_1, \dots q_n]^T \in SE(3) \times \mathbb{R}^n$ represents a point in the robot's configuration manifold while $\dot{\mathbf{q}} = [\dot{\mathbf{q}}_b^T, \dot{q}_1, \dots, \dot{q}_n]^T \in \mathbb{R}^{n+6}$ is the generalized joint space velocity vector. The vector $\dot{\mathbf{q}}_b \in \mathbb{R}^6$ represents the joint space velocity of the floating base (refer to the Notation section for a partial list of the main symbols employed in this dissertation and their meaning). Opposed to the joint-space velocity, the spatial velocity of each link can be written as:

$$\boldsymbol{\nu}_i = \begin{bmatrix} \dot{\mathbf{x}}_i \\ \boldsymbol{\omega}_i \end{bmatrix} \in \mathbb{R}^6 \tag{2-5}$$

The spatial notation allows us to describe generic motions which combine arbitrary rotations and translations[2] (*e.g.,* in [23, 26, 22, 27]). The kinetic energy of the system can be written as a function of the links spatial velocities:

$$T = \sum_{i=1}^{n} T_i. \tag{2-6}$$

where:

$$T_i = \frac{1}{2} m_i \dot{\mathbf{x}}_i^T \dot{\mathbf{x}}_i + \frac{1}{2} \boldsymbol{\omega}_i^T I_i \boldsymbol{\omega}_i = \sum_{i=1}^{n} \frac{1}{2} \boldsymbol{\nu}_i^T \mathbf{M}_i \boldsymbol{\nu}_i. \tag{2-7}$$

where $I_i \in \mathbb{R}^{3\times3}$ is the link's angular inertial expressed at the world frame origin:

$$I_i = I_i^{com} + m_i [\mathbf{c}_i] \times [\mathbf{c}_i] \times^T \tag{2-8}$$

The matrix $\mathbf{M}_i \in \mathbb{R}^{6\times6}$ is the symmetric, positive-definite, spatial inertia matrix of the $i-th$ link:

$$\mathbf{M}_i = \begin{bmatrix} m_i \mathbb{1}_3 & m_i[\mathbf{c}_i]\times^T \\ m_i[\mathbf{c}_i]\times & I_i^{com} \end{bmatrix} \tag{2-9}$$

[2]Notice that the decision of concatenating the linear position first and the angular velocity later is purely arbitrary and, with an appropriate rearrangements of all the terms, it leads to the same results as if the order of the two terms was switched. As an abuse of notation, we also define as wrench a quantity that is not the dual of a twist, but a 6D force/moment vector.

and $I_i^{com} \in \mathbb{R}^{3\times3}$ is the link angular inertial in the link's CoM frame[3].
Exploiting the fact that:

$$\boldsymbol{\nu}_i = \mathbf{J}_i \dot{\mathbf{q}}_i \tag{2-10}$$

we can further develop Eq. 2-7 and obtain the total kinetic energy of the system [25]:

$$T(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{i=1}^{N} \frac{1}{2} \dot{\mathbf{q}}_i^T \mathbf{J}_i^T \mathbf{M}_i \mathbf{J}_i \dot{\mathbf{q}}_i = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} \tag{2-11}$$

where $\mathbf{M} \in \mathbb{R}^{(n+6)\times(n+6)}$ is the robot's joint-space inertia matrix of the system and $\mathbf{J}_i \in \mathbb{R}^{6\times n_i}$ is a joint type dependent matrix (*e.g.,* in the case of the floating base joint then $\mathbf{J}_0 = \mathbb{1}_6$) that maps joint-space velocities into task space velocities.

The potential energy of the overall system can instead be expressed as:

$$U(\mathbf{q}) = \sum_{i=1}^{n} m_i g h_i(\mathbf{q}) \tag{2-12}$$

where $h_i(\mathbf{q})$ is a function that provides the links' height parallel to gravity.

By solving the Lagrange Equation:

$$\frac{d}{dt}\frac{\vartheta L(\mathbf{q}, \dot{\mathbf{q}})}{\vartheta \dot{\mathbf{q}}} - \frac{\vartheta L(\mathbf{q}, \dot{\mathbf{q}})}{\vartheta \mathbf{q}} = \boldsymbol{\tau} \tag{2-13}$$

we obtain the dynamic equation of motion in canonical form (check [25] for the full derivation):

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \mathbf{B}\boldsymbol{\tau} \tag{2-14}$$

where we write $\mathbf{M}(\mathbf{q})$ to point out the dependency of the joint space inertia matrix from the robot's configuration. $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n+6}$ accounts for the Coriolis terms and $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^{n+6}$ considers the effect of gravity acting on the links' mass. $\mathbf{B} \in \mathbb{R}^{(n+6)\times n}$ is a selection matrix that selects the torque variables referring to the unactuated floating base.
We can also add one more term to Eq. 2-14 in order to include possible external wrenches $\mathbf{f}$ being applied to the robot:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \mathbf{B}\boldsymbol{\tau} + \mathbf{J}^T(\mathbf{q})\mathbf{f} \tag{2-15}$$

$\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{m\times(n+6)}$ is the Jacobian matrix mapping the joint space velocities into spatial velocities. The same mapping (transposed) is used to map the external wrenches $\mathbf{f} = [\mathbf{w}_1^T, \dots \mathbf{w}_{n_c}^T]^T \in \mathbb{R}^m$ (contact forces between feet and the ground or external pushes/disturbances) into joint-space generalized torques; $m = 6n_c$ where n_c is the number of external wrenches $\mathbf{w} \in \mathbb{R}^6$ acting on the robot (*e.g.,* $n_c = 2$ in the case of a humanoid robot being in double support without any other contact with the environment). In the case of pure

[3]If the link frame i is located in the link's CoM and it is oriented along the link's principal axes, then the matrices I_i^{com} and I_i become 3×3 diagonal matrices and $\mathbf{M}_i$ becomes a 6×6 diagonal matrix.

external forces with no torque component, such as in the case of quadruped robots with point feet, we have instead: $m = 3n_c$. Because of the intermittent nature of legged locomotion, the dimension n_c changes at every switch of contact (*e.g.,* feet touch down or take off), for instance: $n_c = 0$ at flight phase with no external disturbances. This detail makes of Eq. 2-15 a system with switching dynamics which must be dealt with special care in optimal control and motion planning.

It is to be noticed that Eq. 2-15 does not consider the non-holonomy that derives from the non-*integrability* of the angular velocity $\boldsymbol{\omega}$ (see Appendix A-2). For this reason Eq. 2-15 can only be used for honolomic systems such as, for example, fixed-based industrial manipulators. For non-honolomic systems such as the floating-base platforms a new generalized velocity vector $\mathbf{s} \in \mathbb{R}^{n+6}$ must be defined such that $\mathbf{s} = [\boldsymbol{\nu}_b, \dot{q}_1, \dots, \dot{q}_n]^T$ where $\boldsymbol{\nu}_b = \boldsymbol{\nu}_0$ is the spatial velocity of the floating base joint. Eq. 2-15 can thus be re-written in the following form:

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{s}} + \mathbf{C}(\mathbf{q}, s) + \mathbf{g}(\mathbf{q}) = \mathbf{B}\boldsymbol{\tau} + \mathbf{J}^T(\mathbf{q})\mathbf{f} \tag{2-16}$$

which explicitly considers the non-integrability of the angular velocity. Eq. 2-16 can be expanded as in:

$$\underbrace{\begin{bmatrix} \mathbf{M}_b & \mathbf{M}_{bj} \\ \mathbf{M}_{bj}^T & \mathbf{M}_j \end{bmatrix}}_{\mathbf{M}(\mathbf{q})} \underbrace{\begin{bmatrix} \dot{\boldsymbol{\nu}}_\mathbf{b} \\ \ddot{\mathbf{q}}_j \end{bmatrix}}_{\dot{\mathbf{s}}} + \underbrace{\begin{bmatrix} \mathbf{c}_b \\ \mathbf{c}_j \end{bmatrix}}_{\mathbf{C}(\mathbf{q},\mathbf{s})} + \underbrace{\begin{bmatrix} \mathbf{g}_b \\ \mathbf{g}_j \end{bmatrix}}_{\mathbf{g}(\mathbf{q})} = \underbrace{\begin{bmatrix} \mathbf{0}_{6\times n} \\ \mathbf{I}_{n\times n} \end{bmatrix}}_{\mathbf{B}} \boldsymbol{\tau} + \underbrace{\begin{bmatrix} \mathbf{J}_b^T \\ \mathbf{J}_q^T \end{bmatrix}}_{\mathbf{J}(\mathbf{q})^T} \mathbf{f}. \tag{2-17}$$

in order to highlight the dynamic coupling between the underactuated floating-base (first line) and the actuated joints of the kinematic tree (second line).

The formulation given above can be employed equally well for dynamic modeling of humanoids (see, for example, Fig. 2-6), quadrupeds (see, for example, Fig. 2-8) as much as of any other tree-structured robotic platform.
The large number of variables described in this Section and the nonlinearity of the involved terms (*e.g.,* see the bilinear and quadratic terms in the computation of the joint space inertia matrix $\mathbf{M}(\mathbf{q})$ in Eq. 2-9) makes it hard to formulate real-time, or online, optimization-based motion planners based on the described model. Full models have been, however, largely employed in the synthesis of *whole-body controllers* capable of optimizing in real-time a suitable control input (joint torques) for given reference feet and CoM trajectories [21, 28, 29, 30].

B) **Centroidal Dynamics**

The *centroidal dynamics* describes the dynamics of the whole robot, at a given configuration, as a single rigid body. Is is equivalent to the full-body formulation in terms of CoM dynamics and inertia [27]. It exploits and extends the fundamental property of the CoM (of *Center of Gravity* (CoG)) defined as the virtual point where all the mass of the robot can be considered

(a) Atlas, 2012

(b) Atlas, 2018

(c) Valkyrie, 2015

Figure 2-6: Humanoid robots examples:

(a) Atlas developed by Boston Dynamics, 2012 [31];

(b) second version of Atlas developed by Boston Dynamics, 2018 [31];

(c) Valkyrie developed by NASA, 2015 [32].

to be lumped without changing the dynamic behavior of the system as a whole (disregarding the joints dynamics as if they were instantaneously "welded" together). As a consequence, we can assume that the gravity force acting on a robot is acting on the CoM as an applied force (rather than acting on all the bodies as a distributed vector field).

A way to compute the centroidal dynamics is to map the spatial momentum of each single link of the robot into the *centroidal frame*:

$$\mathbf{h}_i = \begin{bmatrix} \mathbf{l}_i \\ \mathbf{k}_i \end{bmatrix} = \mathbf{M}_i \boldsymbol{\nu}_i \in \mathbb{R}^6 \tag{2-18}$$

The *centroidal frame* is a reference frame that is always oriented like the inertial world frame and whose origin is attached at the time-varying CoM of the robot. The quantities $\mathbf{l}_i \in \mathbb{R}^3$ and $\mathbf{k}_i \in \mathbb{R}^3$ in Eq. 2-18 represent, respectively, the linear and angular momentum of the link. This "joints-to-centroid" mapping is enhanced by the Centroidal Momentum Matrix (CMM) $\mathbf{A}_G(\mathbf{q})$ [33]:

$$\mathbf{h}_G = \begin{bmatrix} \mathbf{l}_G \\ \mathbf{k}_G \end{bmatrix} = \sum_{i=1}^{n} {}^i\mathbf{X}_G^T \mathbf{h}_i = \underbrace{\sum_{i=1}^{n} {}^i\mathbf{X}_G^T \mathbf{A}_i}_{\mathbf{A}_G(\mathbf{q})} \dot{\mathbf{q}} = \mathbf{A}_G(\mathbf{q})\dot{\mathbf{q}} \tag{2-19}$$

where $\mathbf{l}_G \in \mathbb{R}^3$ is the centroidal linear momentum $\mathbf{k}_G \in \mathbb{R}^3$ is the centroidal angular momentum. $\mathbf{A}_G(\mathbf{q}) \in \mathbb{R}^{6\times(n+6)}$ directly maps the joint space velocity $\dot{\mathbf{q}} \in \mathbb{R}^{n+6}$ into the robot's

CoM spatial momentum $\mathbf{h}_G \in \mathbb{R}^6$. The matrix $\mathbf{A}_i = \mathbf{M}_i\mathbf{J}_i \in \mathbb{R}^{6\times(n+6)}$ represents the link's momentum matrix expressed in the link's frame and ${}^i\mathbf{X}_G \in \mathbb{R}^{6\times 6}$ is the projection matrix from centroidal to link coordinates. $\mathbf{J}_i$ is the Jacobian matrix that maps joint-space velocities into task space velocities as defined in Eq. 2-10.

The *Trasformation Diagram* (as in [33]) represents a useful tool for visualizing the most meaningful available transformation matrices that can be used to switch among the three most relevant spaces:

- joint configuration space (number of dimensions: $N = 6 + n$);
- system space (number of dimensions: $N = 6n$);
- task/operational space (number of dimensions: $N = 6$).

The centroidal dynamics modeling holds special properties that make it an intermediate tool between the full dynamic description, explained above, and the simplified models. In particular, it allows to map the main dynamics of the system to a lower dimensional space than the joint space (from $n + 6$ to 6 DoFs) without trading off the accuracy of the described physical quantity. It is, in conclusion, especially well suited for applications such as the planning and control of the linear and angular momentum of humanoid robots (whose links' mass can not be neglected, *e.g.,* Fig. 2-6).

C) **Simplified Dynamic Models**

Simplified models are significantly different from the models explained in the previous section and they can often be seen as heuristic attempts to describe the main dynamics of the system. As a matter of fact, the delimiting line which distinguishes between heuristics and simplified dynamic models is not clearly marked [34]: both approaches often lead to simple control strategies that try to capture the main dynamics of the robot during locomotion. Their substantial difference, however, can be found in the amount of quantitative experimental data recorded on real-life biological organisms that demonstrates the descriptive quality of the simplified models. Reduced models, as a matter of fact, are bound to the biological counterparts such as humans or animals by extensive research in biomechanics that has proved how well they describe the behavior of the animal they refer to [35, 36, 37].
In other cases, unlike heuristic strategies, simplified models for legged locomotion are often derived by generic principles of fundamental physics such as the energy and momentum conservation principles [38]. Starting from these generic rules, simplifying hypotheses are then proposed to reduce the math and fit the equations of the specific locomotion task to be tackled.

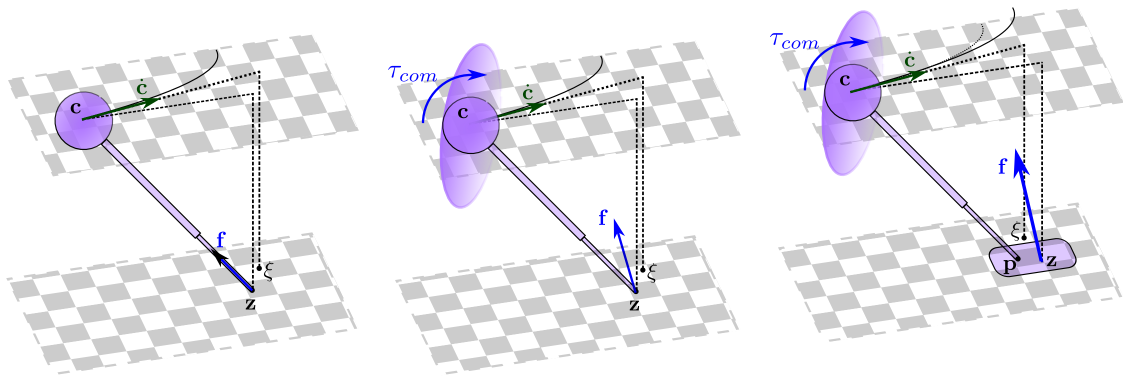
This concept of simplified model is thoroughly developed by Full *et al.* [39] under the name of *template* while the mechanism that connects this to the real model is called the *anchor*. Templates are defined in lower dimensional spaces (*e.g.,* the task/operational space) compared to the original system space. An under-determined task in the system space may thus become fully determined in the space of the template. An example of this strategy is the LIP which captures the pendular nature of human walking. The height restriction implied in the LIP reduces the DoFs that can be exploited to generate a locomotion plan and the linear telescopic leg avoids the nonlinearities involved in the kinematics of the real legs. Motion plans can then be efficiently optimized with an LP and the solution can be mapped back to the full robot model resorting to the full kinematics (*i.e.,* the anchor). Other examples of templates that can be observed in the locomotion pattern of a wide variety of animals ranging from cockroaches to kangaroos are, for example, the Spring Loaded Inverted Pendulum (SLIP) and the Lateral Leg Spring (LLS) model [40].

Keeping the templates idea in mind, strategies for generating locomotion behaviors on legged robots have multiplied in the past decades. A key advantage of the approaches based on heuristics and/or templates compared to the full model description is computational efficiency, that allows to quickly recompute a new control input (analytically, in closed form, solving a LP or a QP) whenever an unexpected disturbance interacts with the system. The price to be paid for this computational speed up is, however, sub-optimality.
The increased safety and robustness that comes with the ability to replan online (or, even better, in real-time) has enabled hardware implementation on possibly any robotic platform. In the rest of this dissertation we will refer to the ability to replan online as *interactivity.*

The interactivity requirement of motion planners plays a key role in the achievement of appropriate robustness levels for hardware implementation; even the most complete and descriptive locomotion formulation will not be useful for robot implementation if it's not fast enough to allow online replanning and thus react to unexpected events or interact with external agents within an acceptable time delay.
The goal is therefore to find the most descriptive locomotion model that complies with the requirement of online execution. This represents a difficult trade-off among:

- the completeness of the employed dynamic model;
- the duration of the time window that defines the predictive capabilities of the planner;
- the available solve time allowed to the planner by the real-time (or online) requirement.

The Linear Inverted Pendulum The LIP represents a powerful bridge between the Zero Moment Point (ZMP) $\mathbf{z} \in \mathbb{R}^2$ and the (nonlinear) inverted pendulum, a commonly used model for describing the humanoid balancing problem. Given the robot's CoM position $\mathbf{c} \in \mathbb{R}^3$ in the



(a) LIP model **(b)** LIP model + flywheel **(c)** LIP model + flywheel + foot

Figure 2-7: Examples of simplified models for legged locomotion: (a) the LIP, (b) the LIP plus flywheel and (c) the LIP plus flywheel and foot. The variables $\mathbf{c} \in \mathbb{R}^3, \mathbf{z} \in \mathbb{R}^2$ and $\boldsymbol{\xi} \in \mathbb{R}^2$ represent, respectively, the CoM, ZMP and the CP of the model.

inertial frame, we can express it's projection $\mathbf{c}_{xy} \in \mathbb{R}^2$ on the (x, y) plane (plane orthogonal to the direction in which gravity acts) as:

$$\mathbf{c}_{xy} = \mathbf{P}\mathbf{c} \tag{2-20}$$

where $\mathbf{P}$ is the projection matrix:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{2-21}$$

The LIP model can then be expressed in the following form:

$$\mathbf{z} = \mathbf{c}_{xy} - \frac{1}{\omega^2}\ddot{\mathbf{c}}_{xy} \tag{2-22}$$

where $\omega = \sqrt{g/c_z}$ is the natural frequency of the LIP. This model was introduced by Kajita *et al.* [41, 42]. Eq. 2-22 holds under the assumption of zero vertical acceleration, no angular dynamics, infinite friction and it represents a simple linear relation between the ZMP and the CoM dynamics. The ZMP is defined as the point on the (flat) ground where the moment produced by the inertial and gravitational forces is parallel to the surface normal (*i.e.,* the robot is not tipping over) [43]. The ZMP will be further discussed in the Section 2-2 where it will be mentioned as one of the most common stability analysis criteria for semi-dynamic locomotion (*i.e.,* it allows CoM accelerations but no flight phases).

Equation 2-22 constrains the contact force to be always aligned along the prismatic joint of the pendulum and, for this reason, the ZMP always corresponds to the contact point of the linear pendulum with the ground. The horizontal contact forces $\mathbf{f}_{xy}$ acting in the LIP can thus be seen as a repulsive spring having a stiffness equal to $m\omega^2$ (where m is the value of the lumped mass in $\mathbf{c}$):

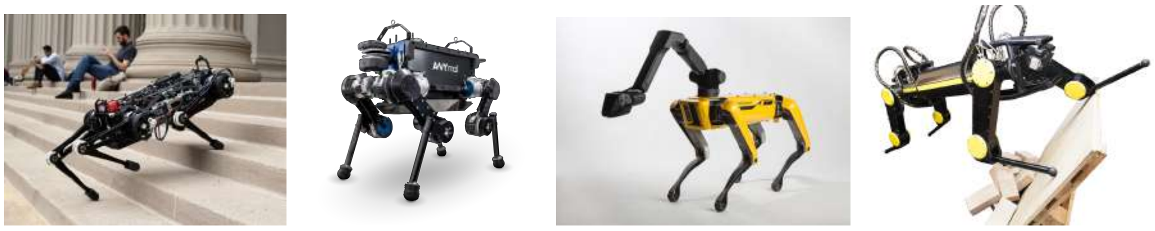
$$\mathbf{f}_{xy} = m\omega^2(\mathbf{c}_{xy} - \mathbf{z}) \tag{2-23}$$

As a side consideration, a friction coefficient constraint in the LIP model would directly result in a limit on the pendulum angle (pitch and roll) as in Fig. 2-7a.
In the more general case in which a non null inertia (a flywheel) is included in the LIP model, then a force redundancy arises (the contact force can be controlled by both the prismatic joint and the flywheel torque) and, therefore, the contact force might not be aligned anymore with the pendulum (see Fig. 2-7b). If the model includes a foot, then the ZMP can be located in any point within the contact surface. The model can be equivalently be modeled as a LIP with varying pivot point coincident to the ZMP position and passive ankle joint or as a LIP with fixed pivot point and actuated ankle joint (adding in this case a further force redundancy to the model) as in Fig. 2-7c. As a last possibility we also mention the possibility of modeling the linear inverted pendulum with a foot and no flywheel: in this case the template is usually known under the name of *table-cart model*.

The Single Rigid Body Dynamics represents a further simplification of the centroidal dynamics explained in Section 2-1-2 consisting in assuming the centroidal inertia to be constant and independent from the robot configuration. This assumption becomes more accurate for legged robots with heavy trunk and lightweight limbs; in this case the fixed inertia matrix can be approximated to be equivalent to the trunk's inertial matrix. This works especially well for quadruped robots, *e.g.,* see Fig. 2-8, whose actuators are often designed to fit in the robots trunk thus obtaining small legs' inertia [44]. In the case of humanoid robots the variations of the centroidal inertia matrix can be larger due to heavier legs, but these can still be approximated to a predefined average value (*e.g.,* value corresponding to the default configuration).
Unlike the centroidal dynamics, the single rigid body dynamics does not capture dynamic phenomena connected to the motion of the swing legs and arms such as the upper limbs' oscillation to reduce trunk's motion and stabilize the walking or the trunk's rocking motion during a jump due to the re-orientation of the legs.

2-1-3 Data-Driven Approaches

Thanks to the wide diffusion of the, so called, Internet of Things (which makes available large amounts of any sort of data, or *Big Data*) and the decreasing cost of parallel computing units (*e.g.,* GPUs) in the past decade, we witnessed a burst of applications where statistical methods are employed to extrapolate decisions and control policies that cannot be taken with any other tool.
Consequently, also in the field of legged locomotion, a constantly growing part of the research community is focusing on data-driven approaches such as Machine Learning (ML) and Reinforcement Learning (RL) for writing algorithms which are capable of teaching a robot how to walk, jump and run. Such algorithms promise to overcome most of the limitations given by heuristics (Section 2-1-1) and model-based approaches (Secton 2-1-2).



(a) MIT Cheetah 3, 2018 **(b)** ANYmal, 2016 **(c)** Spotmini, 2018 **(d)** HyQ2Max, 2016

Figure 2-8: Quadruped robots examples:

(a) Cheetah 3 developed by the Biomimetic Robotics Lab. at MIT, 2018 [45];

(b) ANYmal developed by ANYbotics, 2016 [46];

(c) Spot Mini developed by Boston Dynamics, 2018 [31];

(d) HyQ2Max developed by DLS lab. (IIT) [47].

Impressive results in simulation have been achieved on this track, starting from the recently published papers of DeepMind [48] and Abbeel et al. [49]. Such methods might either be completely unsupervised, meaning that the algorithm has to first explore the policy space by trial and error, or be partially supervised, in which case the algorithm is trained with initial success examples. In the supervised learning setting, the algorithm might be given the available sensor data (joint trajectories, IMU accelerations, joint torque trajectories, *etc.*) regarding past experiments of a given robot performing a specified task. In the unsupervised setting, instead, the algorithm is initially not given any clue about how to reach the final target and this can only be achieved by the, so called, *trial and error*.
Whether supervised or not, the main challenge of these strategies consists in learning policies that are capable of generalizing the successful samples to a diverse range of hardware models, environments and disturbances.

When generality is not obtained then the learning algorithms may lead to brittle policies that fail when executed on scenarios that differ from the one of the training data. This is a major drawback especially when we intend to execute the learned policies on real robotic platforms where several unmodeled dynamics may affect the execution of the task and various unexpected events may occur.

Under such consideration there exist a number of intermediate strategies that try to improve the locomotion behavior (or secondary behaviors) using machine learning approaches starting from the optimal solution of, often more reliable, model-driven trajectory optimization strategies. This finds a more straightforward applicability to the hardware compared to learning the whole locomotion task at once. Examples of that can be found for example in Villarreal *et al.* [50] where, for predefined base trajectories and feet sequences of a quadruped robot, a self-supervised Convolutional Neural Network (CNN) is trained from heuristics to learn the locations on the terrains where the robot should place its feet, considering the terrain morphology, slip conditions and the risk of possible shin-collisions with the terrain.

Similar approaches that merge model-based optimization and machine learning are all those optimization problems whose constraints are obtained as a result of an offline data-driven approach. Such learned constraints are typically low-dimensional *proxy* constraints of high dimensional feasibility constraints such as the joint kinematic limits, the self-collisions and the possible collisions with a complex geometry environment [51, 52, 53].

2-2 Stability and Feasibility Analysis

The dynamic modeling strategies explained in Section 2-1 must be coupled with an appropriate stability criterion for the generation of legged locomotion. This Section presents the main stability criteria employed in the robotics research community, trying to discuss the individual advantages and disadvantages of each one of them[4].

2-2-1 The Support Polygon versus the Support Region

The support polygon criterion has been the first to be applied in robotics and it is still used today thanks to its innate simplicity. It still finds application today (see Fig. 2-9) in safety critical scenarios where stability robustness is more relevant than locomotion speed or whenever high inertial accelerations might cause the robot to reach their feasibility limits (*e.g.,* joint torques and/or speed limits). The stability criterion related to the support polygon states that the projection of the CoM $\mathbf{c}_{xy} \in \mathbb{R}^2$ on a plane perpendicular to gravity has to be within the convex hull of the robot's feet for the system to balance and to be statically stable.

The stability criterion can be stated as:

$$\mathbf{S}(\mathbf{p}_{i,xy})\mathbf{c}_{xy} \leq \mathbf{t}(\mathbf{p}_{i,xy}) \tag{2-24}$$

where $\mathbf{S}(\mathbf{p}_{i,xy}) \in \mathbb{R}^{n_e \times 2}$ and $\mathbf{t}(\mathbf{p}_{i,xy}) \in \mathbb{R}^{n_e}$ represent the set of linear inequalities that delimit the area of the support polygon (where n_e is the number of edges of the convex-hull) given by the projections $\mathbf{p}_{i,xy} \in \mathbb{R}^2$ of the n_c contact points on the (x, y) plane:

$$\mathbf{p}_{i,xy} = \mathbf{P} \cdot \mathbf{p}_i \quad i = 1, \dots n_c \tag{2-25}$$

The condition in Eq. 2-24 can be expressed in terms of the support polygon vertices (*i.e.,* the footholds):

$$\mathbf{c}_{xy} \in ConvHull(\mathbf{p}_{i,xy}) \tag{2-26}$$

[4]The term *stability* is here intended as in the sense defined by Pang *et al.* [54] and it should not be confused with the (a-priori unrelated) notion of Lyapunov stability.

Whenever the contact points do not belong to the same horizontal plane, the criterion in Eq. 2-24 is no longer defined. There exist, however, many works that have approximated this criterion to degenerate cases such as the case with non-horizontal coplanar contacts or the condition in which the surface normals at the contact points are all parallel but their locations are not coplanar, in which case the convex-hull becomes a 3D volume.

In their seminal work, Bretl *et al.* [55] demonstrated that, even if the feet belong to the same horizontal plane, the orientation of the surface normals at the contact locations significantly influences the shape of the areas where the robot can place its CoM to be statically stable. Such areas were efficiently obtained by means of an *Iterative Projection (IP)* algorithm [56], a cutting-plane recursive procedure which will be described in detail in Section 5-2. The first obtained result is the fact that the support area is, in general, not a polygon: it is instead a 2D convex region belonging to the (x, y) plane (orthogonal direction to gravity) and, therefore, this can only be approximated by a linear set of inequalities (a 2D polygon). The second result is the fact that, depending on the normals orientation, the shape of the support *region*, where static stability is ensured, can be considerably different from the support *polygon*, *i.e.,* the convex hull of the contact locations (sometimes they even being two completely disjunct sets). In the general case, the support region is in fact not a polygon at all, although it is usually approximated as a 2D linear set (*i.e.,* a polygon) with a number of edges dependent on the desired approximation accuracy. The support region corresponds to the support polygon (*i.e.,* the convex hull of the contact points) only when all the contacts belong to the same horizontal surface. Their difference becomes more and more apparent with the increase of complexity of the terrain geometry, making the *support region* a much more suitable criterion for *static* rough terrain navigation compared to the *support polygon.*
The static locomotion assumption stems from the consideration of gravity as the only external force acting on the robot at all time during the computation of $\mathbf{S}(\mathbf{p}_{i,xy})$ and $\mathbf{t}(\mathbf{p}_{i,xy})$.

2-2-2 The Zero Moment Point (ZMP) and the Center of Pressure (CoP)

The CoM projection, explained above, has been replaced by a more dynamic stability criterion that has its focus on another two-dimensional *ground reference point*, named Zero Moment Point (ZMP) $\mathbf{z} = (z_x, z_y) \in \mathbb{R}^2$. As anticipated above, the ZMP is defined as the point on the flat ground where the moment produced by the inertial and gravitational forces is parallel to the surface normal (*i.e.,* the robot is not tipping over) [43]. It is a more dynamic criterion than the CoM projection in that it allows non-null inertial accelerations. It cannot be considered, however, a fully dynamic stability criterion since it is undefined in the case of flight phases. The ZMP can be obtained by re-arranging the Newton-Euler equations and it accounts for

(a) Lauron III **(b)** Titan XI. **(c)** Robosimian

Figure 2-9: Example of three robots performing statically stable locomotion:

(a) Lauron series robots by the Forschungszentrum Informatik Karlsruhe (FZI), 2014 [57];

(b) Titan series robots developed at the Tokyo Institute of Technology, 2004 [58];

(c) Robosimian developed by NASA, 2016 [59].

all the gravitational and inertial forces and torques acting on the CoM of the system [60]:

$$\begin{aligned} z_x &= \quad c_x - \frac{\ddot{c}_x}{\ddot{c}_z + g} c_z - \frac{\dot{k}_y}{m(\ddot{c}_z + g)} \\ z_y &= \quad c_y - \frac{\ddot{c}_y}{\ddot{c}_z + g} c_z - \frac{\dot{k}_x}{m(\ddot{c}_z + g)} \end{aligned} \tag{2-27}$$

where $\dot{k}_x$ and $\dot{k}_y$ are the x, y components of the angular momentum rate $\dot{\mathbf{k}}$.

The ZMP stability criterion has been initially proposed by Vukobratović *et al.* [61, 62] in 1968 and has been further developed in multiple following works. It states that dynamic stability is achieved if the ZMP (*e.g.,* intersection of the ZMP line with the plane of support polygon) is inside the support polygon/region.

Since dynamic balance is achieved when the contact forces directly oppose the gravitational and inertial forces, maintaining the ZMP within the contact support polygon ensures that a contact force can exist that is able to achieve this equilibrium, thus controlling the motion. When this point goes out of the polygon it is not longer possible to achieve an equilibrium of the moments and the robot starts to tip over around an edge ot the support polygon.

One key aspect, from this point of view, is the fact that the ZMP can be considered as a *global* stability criterion only when all the contact points are coplanar. In all the other cases the ZMP is instead a *local* stability criterion meaning that it can assess the stability of the specific contact surface between the robot and the terrain (*e.g.,* one local ZMP for each individual foot surface). The local ZMP takes also usually the name of Center of Pressure (CoP) (defined as the point of application of the ground reaction forces). The *global* extension, able to assess the overall stability of the full system having multiple contact points with the environment, is only defined on a flat ground, where all the feet are therefore coplanar.

The ZMP stability condition assumes the support region to correspond to the convex hull S of all the feet in contact with the ground and it states that, if the *global* ZMP $\mathbf{z} \in \mathbb{R}^2$ belongs to it, the global stability of the robot is guaranteed:

$$\mathbf{S}(\mathbf{p}_{i,xy})\mathbf{z} \leq \mathbf{t}(\mathbf{p}_{i,xy}) \tag{2-28}$$

or, equivalently:

$$\mathbf{z} \in ConvHull(\mathbf{p}_{i,xy}) \tag{2-29}$$

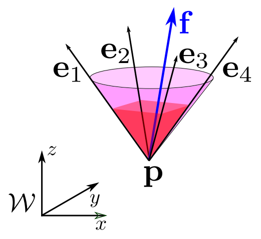
On the other hand, if all the contact surfaces between robot and environment are not coplanar, the convex hull of the contact areas is not defined and the *global* ZMP does not represent a possible stability predictor. In this case, checking the stability of every single *local* ZMP (*i.e.,* making sure that every stance foot and hand has a stable CoP) ensures that each foot does not rotate but does not say anything about the global stability of the whole robot [29, 63]. This approach is most commonly used for humanoid robots considering that quadrupeds are usually approximated to have contact points rather than contact surfaces and, in that case, the *local* ZMPs (*i.e.,* CoPs) simply correspond to the contact points.

Differences between the CoP and the ZMP typically arise in the case of a foot rolling around its edge; in this case the CoP belongs to the edge around which the foot is rotating, the ZMP, however, is not defined because the angular momentum around the edge is non zero [62, 64]. In such case an extended definition of the ZMP, called Fictitious Zero Moment Point (FZMP) (which exists outside of the support region) can be used; alternatively the Foot Rotation Indicator (FRI) can similarly be employed for describing the stability of a legged robot rotating around one edge of its feet. Both the CoP and the ZMP assume that a sufficient friction acts between foot and ground.

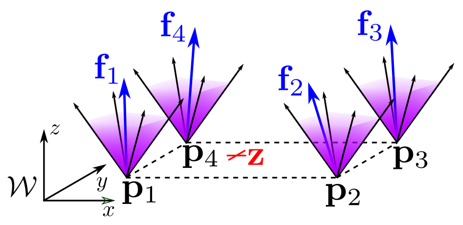
Extensions of the ZMP to non flat environments and to multi-contact scenarios have been proposed [65, 66, 67] and will be the main topic of the following Section.

2-2-3 The Contact Wrench Cone (CWC)

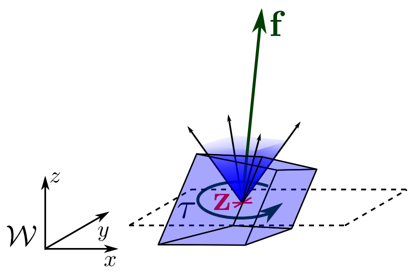
The first analysis of the dynamic stability on non-coplanar contacts that comprises both linear and angular terms was performed by Saida *et al.* [65]. In their work, the stability problem corresponded to the challenge of finding a feasible set of contact wrenches to an arbitrary distribution of contact points. This lead to the proposal of the Feasible Solution of Wrench (FSW), a virtual point that corresponds to the ZMP in the case of pure external forces (no torques) and coplanar contacts. This work has been later further developed in [66] into the currently used formulation that states that the aggregated wrench (also called



(a) Linearized friction cone.



(b) Friction cones of a foot.



(c) Wrench cone of a foot.

Figure 2-10: Different types of contact modeling. A single point contact can be modeled with (left) a friction cone or with a linearized cone (*i.e.*, an inverted pyramid with an arbitrary number of edges). (Middle) a surface contact can be modeled as (middle) a set of individual friction cones or as (right) a unique wrench cone.

Contact Wrench Sum (CWS) or *Gravito-Inertial Wrench (GIW)*) $\mathbf{w}_{GI} \in \mathbb{R}^6$:

$$\mathbf{w}_{GI} = \underbrace{\begin{bmatrix} \dot{\mathbf{l}} \\ \dot{\mathbf{k}}_O \end{bmatrix}}_{\dot{\mathbf{h}}} - \underbrace{\begin{bmatrix} m\mathbf{g} \\ \mathbf{c} \times \mathbf{g} \end{bmatrix}}_{\mathbf{w}_G} \tag{2-30}$$

acting on the robot must remain inside the convex polyhedral cone resulting from the sum of the individual friction cones of each contact point. In Eq. 2-30, the term $\dot{\mathbf{h}}$ represents the rate of linear and angular momentum expressed in the inertial frame $\mathcal{O}$ whilst $\mathbf{w}_G$ represents the momentum change contributed by gravity. The new stability criterion can thus be stated as an Ordinary Differential Inclusion (ODI):

$$\mathbf{w}_{GI} \in CWC \tag{2-31}$$

The CWC is defined as the Minkowski sum of the friction cones of all the contact points:

$$CWC = ConvexCone\left(\begin{bmatrix} \mathbf{e}_i^k \\ \mathbf{p}_i \times \mathbf{e}_i^k \end{bmatrix}\right) \quad \text{with:} \quad k = 1, \dots n_e, \quad i = 1, \dots n_c \tag{2-32}$$

where n_c is the number of contact forces and n_e is the number of edges $\mathbf{e}$ in which the friction cones are discretized (*e.g.*, see Fig. 2-10a). Usually $n_e = 4$, however it is possible to find works in the literature where $n = 3$ or $n_e = 8$. For a short discussion on the equivalence of Minkowski sums and the *convex hull* operator for unbounded polyhedral cones please refer to Appendix B-2.

In the above mentioned works, for the first time, the aggregated wrench $\mathbf{w}_{GI}$ acting on the robot is dealt as *six-dimensional reference point* in opposition to all the previously used two-dimensional ground reference points explained in the above Sections. Therefore, if a centroidal model is used to compute this wrench, this criterion allows us to include both the centroidal linear and angular dynamics in the stability evaluation. This was also the first time that the friction coefficient of each individual contact point can be considered for the evaluation of the global stability of the system.

Dai *et al.* [68, 69] have further exploited the criterion proposed in [66] to optimize robust centroidal trajectories (both linear and angular accelerations) on complex terrains where the venerable ZMP criterion is known to fail. Further works in this direction included the quasi-analytical computation of the CWC constraints only for some specific contacts configurations and surface normal orientations [70].
As previously anticipated, the CWC can be considered to be a generalization of the support region constraint for non-coplanar contact sets with limited friction coefficients. Interestingly enough, in case of coplanar contacts and sufficient friction, the six-dimensional CWC can be *sliced* by the four planes corresponding to $\ddot{c}_x = 0$, $\ddot{c}_y = 0$, $\ddot{c}_z = -g$ and $\ddot{\omega}_z = 0$ thus resulting in the same 2D linear convex set (*i.e.,* the support region) that can be found using the iterative projection algorithm as explained in [55].

Despite the different final goal, these approaches share many similarities with those strategies employed in robotic grasping, a research field closely related to legged locomotion. In this field, in particular, similar definitions are given, such as the Grasp Wrench Set (GWS), in order to identify robust object grasps [71, 72]. In robotic manipulation the goal is usually to achieve *force closure*, namely the condition in which the contacts between the robotic hand and the object have such a configuration that allows to resist any arbitrary external wrench. Under this condition grasp robustness is guaranteed because the robot cannot break the contact from any finger tip without any external non-zero work [73, 74].
Even though the force closure condition is rarely achieved in the problems related to motion planning of legged robots, common tools can be exploited for the benefit of each individual research field.

In this wrench-based family of stability criteria, robustness can be treated in terms of minimal distance between the aggregated wrench $\mathbf{w}_{GI}$ and the facets of the CWC. This generalized distance represents, therefore, the stability margin that must be kept constantly larger than zero in order to ensure robust stability at any time [75]. In this case a six-dimensional ellipsoid can be defined of radius r; the residual radius (*i.e.,* the stability margin) will correspond to the radius r of the largest ellipsoid $\mathcal{B}_r$ centered in $\mathbf{w}_{GI}$ and inscribed in the CWC.
The $\mathbf{w}_{GI}$, just as well as the previously discussed ground reference points, all depend on the instantaneous centroidal state and, as a consequence also the stability margin will continuously change (even if the contacts do not change). A different approach can be found by computing robust (usually convex) regions that are guaranteed to be safe for a predefined level of disturbance. This technique has found application in different works such as [76] where, for a predefined range of centroidal accelerations, a 3D robust region for the position of the CoM is found. Conversely, in [77] a set of admissible CoM accelerations is found for given disturbances in the centroidal position.

2-2-4 The Capture Point

Although many more ground reference points have mentioned over the decades, the most relevant after the CoP is arguably the Instantaneous Capture Point (ICP) $\boldsymbol{\xi} \in \mathbb{R}^2$. The ICP turns upside down the balancing perspective by looking at a foot position on the ground that ensures dynamic stability given the current CoM state, rather than seeking an appropriate CoM target position to stabilize the system for the given support polygon/region (as in the ZMP based approaches).

The ICP is only defined for the LIP model and thus holds similar approximations such as fixed CoM height, flat coplanar contacts and contact force acting as a repulsive spring. The ICP is then defined as the point on the ground where the robot needs to place its foot in order to come to a complete stop given its instantaneous linear velocity [78, 79]:

$$\boldsymbol{\xi} = \mathbf{c}_{xy} + \frac{1}{\omega}\dot{\mathbf{c}}_{xy} \tag{2-33}$$

This can be re-written in the following explicit form:

$$\dot{\mathbf{c}}_{xy} = \omega(\boldsymbol{\xi} - \mathbf{c}_{xy}) \tag{2-34}$$

If we differentiate Eq. 2-34 we obtain:

$$\ddot{\mathbf{c}}_{xy} = \omega(\dot{\boldsymbol{\xi}} - \dot{\mathbf{c}}_{xy}) \tag{2-35}$$

and then plug Eq. 2-22 in Eq. 2-35 we obtain:

$$\mathbf{z} = \mathbf{c}_{xy} - \frac{1}{\omega}(\dot{\boldsymbol{\xi}} - \dot{\mathbf{c}}_{xy}) = \underbrace{\mathbf{c}_{xy} + \frac{1}{\omega}\dot{\mathbf{c}}_{xy}}_{\boldsymbol{\xi}} - \frac{1}{\omega}\dot{\boldsymbol{\xi}} \tag{2-36}$$

In explicit form this corresponds to the following Ordinary Differential Equation (ODE) that describes the ICP dynamics:

$$\dot{\boldsymbol{\xi}} = \omega(\boldsymbol{\xi} - \mathbf{z}) \tag{2-37}$$

The ODEs in Eq. 2-34 and Eq. 2-37 can be seen together as the decomposition of the second order linear dynamics in Eq. 2-22 in two first order linear systems. The former (Eq. 2-34) represents the stable part of the dynamics (*e.g.,* pole having negative real part) showing that the CoM projection converges to the ICP for an infinite horizon. Similarly, the latter (Eq. 2-37) represents the unstable part of the system showing that the ICP will diverge from the ZMP over time. For this reason, ICP has been equivalently called *eXtrapolated Center of Mass (XCoM)* [80] or *Divergent Component of Motion (DCM)* [81]. If the ICP belongs to the support region:

$$\boldsymbol{\xi} \in ConvHull(\mathbf{p}_i) \tag{2-38}$$

or, equivalently:

$$\mathbf{S}(\mathbf{p}_i)\boldsymbol{\xi} \leq \mathbf{t}(\mathbf{p}_i) \tag{2-39}$$

then it is possible to set $\boldsymbol{\xi} = \mathbf{z}$ and $\mathbf{c}_{xy}$ will converge to $\boldsymbol{\xi}$ thus bringing the robot to a complete stop. Based on the ICP, a whole new research line, called Capturability Analysis, is born that looks at the number of steps needed to come to a complete stop, given the limit capabilities of the robot (*e.g.,* finite-sized foot or maximal instantaneous angular momentum rates) [79]. In the capturability framework, a state of the reduced model is said to be *N-steps capturable* if there exists a set of, at least, N steps that will allow to system to come to a stop. The idea is thus that each step will contribute to reduce the kinetic energy of the system until it converges to zero.
Capturability analysis can be seen as a branch of the, more general, *viability analysis* that attempts to identify all those states that do not lead to a fall of the robot [60].

Since they are defined for a linear inverted pendulum model, all the above ground reference points are only defined on a flat ground and completely neglect the vertical dynamics. As a consequence, their applicability to complex geometry environments and to highly dynamic motion is limited. An extension of the ICP to 3D arbitrary terrains can be found in [82] based on the definition of the Enhanced Centroidal Moment Pivot (ECMP) and Virtual Repellent Point (VRP). This strategy, however, focuses on the Newton equation for the centroid:

$$\dot{\mathbf{l}} = m\ddot{\mathbf{c}} = m\mathbf{g} + \sum_{i=1}^{n_c} \mathbf{f}_i \tag{2-40}$$

However, [82] assumes the centroid as a point mass thus discarding the angular momentum change given by the Euler equation:

$$\dot{\mathbf{k}} = I\dot{\boldsymbol{\omega}} = \sum_{i=1}^{n_c} (\mathbf{p}_i - \mathbf{c}) \times \mathbf{f}_i \tag{2-41}$$

The cross product in Eq. 2-41 represents a bilinear constraint that is indeed hard to embed inside real-time, or online, motion planners. Even in the simplified case where the contacts positions $\mathbf{p}_i$ are fixed, the multiplication between CoM position $\mathbf{c}$ and external forces $\mathbf{f}_i$ remains bilinear [83].
The branch of stability analysis approaches that are compatible with locomotion tasks that require large angular momentum variations have been explained in the previous Section 2-2-3. The next Section will instead deal with a more generic method for the analysis of viability that employs tools of nonlinear control theory such as fixed points and limit cycles.

2-2-5 Stable Limit Cycles

We will now go through a different approach for the stability analysis of legged robots that considerably differs from the strategies explained so far mainly based on low dimensional (2D or 6D) centroidal reference points. This approach lies its foundations in the concepts of attractors (*i.e.,* a set of numerical values to which the system tends to evolve for a large set

of initial conditions); in particular, two types of attractors are considered: fixed points and limit cycles. *Fixed points* (or *equilibrium points*) are recurrent states in the execution of a recursive, cyclic, task; in the locomotion domain this may correspond to a safe posture where the robot can safely stand still [60]. *Limit cycles* represent an extension of fixed points and can be defined as recurrent/periodic trajectories that repeat at a constant time interval T [84].
The periodic behavior can be found, for example, in many hybrid systems where discrete events (such as saturation limits) can force the states to stay within a limited portion of the state space [85].

Let us consider an arbitrary nonlinear multi-variable system $\mathbf{a}(\cdot)$ and a state $\mathbf{x}(t) \in \mathbb{R}^d$:

$$\dot{\mathbf{x}}(t) = \mathbf{a}(t, \mathbf{x}(t)) \tag{2-42}$$

In the locomotion setting, the state $\mathbf{x}(t)$ may represent, depending on the application, the set of joint configurations $\mathbf{q}(t) \in \mathbb{R}^{n+6}$, the joint torques $\boldsymbol{\tau}(t) \in \mathbb{R}^n$, the contact forces/wrenches $\mathbf{f}(t) \in \mathbb{R}^m$, the robot's CoM task-space position $\mathbf{c}(t) \in \mathbb{R}^3$ or orientation $\mathbf{R}(t) \in SO(3)$. The function $\mathbf{a}(\cdot)$ may, for example, describe the differential evolution of the state with an impact with the environment.

If the system is cyclic then the function $\mathbf{a}(t, \mathbf{x})$ takes on the same values periodically with a constant time period T. If we assume that T is known we can then write:

$$\mathbf{a}(t+T, \mathbf{x}(t)) = \mathbf{a}(t, \mathbf{x}(t)), \quad \forall t \in \mathbb{R} \tag{2-43}$$

or, equivalently:

$$\mathbf{x}(t+T) = \mathbf{x}(t) \quad \forall t \in \mathbb{R} \tag{2-44}$$

Any solution $\phi = \{\mathbf{x}(t) | t \in [0, \infty)]\}$ of the differential Eq. 2-42 that satisfies this condition is named *T-periodic solutions/orbit* (*i.e.,* a limit cycle). For a locomotion task, a periodic solution ϕ may be written for examples as a trajectory $\{\langle \mathbf{q}(t), \boldsymbol{\tau}(t), \mathbf{f}(t), \mathbf{c}(t), \mathbf{R}(t) \rangle | t \in [0, \infty)]\}$ that satisfies the following conditions:

$$\begin{aligned} \mathbf{q}(t+T) &= \mathbf{q}(t); \\ \boldsymbol{\tau}(t+T) &= \boldsymbol{\tau}(t); \\ \mathbf{f}(t+T) &= \mathbf{f}(t); \qquad \forall t \in \mathbb{R} \\ \mathbf{c}(t+T) &= \mathbf{c}(t); \\ \mathbf{R}(t+T) &= \mathbf{R}(t); \end{aligned} \tag{2-45}$$

Any point $\bar{\mathbf{x}}$ at a specific time instant t^* of the periodic trajectory is a fixed point. If we then consider the phase plot of the same cyclic system, this fixed point $\bar{\mathbf{x}}$ also coincides to the intersection between the periodic trajectory and a lower dimensional plane (of dimension d), called *Poincaré section* or *return section* $\mathcal{S}$, orthogonal to the periodic trajectory in t^*.

It is then possible to define the function $\mathbf{P} : \mathcal{S} \rightarrow \mathcal{S}$, called Poincaré map, that maps a state $\mathbf{x}(t^*)$ on the Poincaré section in another element $\mathbf{x}(t^* + T)$ of the same section:

$$\mathbf{x}(t^* + T) = \mathbf{P}(\mathbf{x}(t^*)) \tag{2-46}$$

Thanks to the property stated in Eq. 2-44, fixed points can be seen as elements of the state space that are mapped into themselves by the return map $\mathbf{P}$:

$$\bar{\mathbf{x}} = \mathbf{P}(\bar{\mathbf{x}}) \tag{2-47}$$

Poincaré maps $\mathbf{P}$ can be viewed as discrete dynamical systems. The stability of the fixed points associated to the map $\mathbf{P}$ is closely related to the stability of the original considered system $\mathbf{a}(t, \mathbf{x})$. For this reason Poincaré maps represent a useful tool to analyze the stability of chaotic and hybrid systems that can be hardly performed in other ways.

Although not being limited to any specific system (the same stability principles hold for both reduced and for full systems, continuous or hybrid systems), fixed points and limit cycles analysis are usually performed numerically and thus suffer from the same numerical limitations of other optimization-based planning strategies (*e.g.,* number of variables exponentially increasing with the number of discretization intervals). For this reason, they are usually performed offline and are typically applied to reduced models such as the dynamics of the centroid (especially if designed for online planning applications).

As mentioned above, the computation of the Poincaré maps is usually done numerically: analytic stability assessment of Poincaré maps can only be performed for a restricted family of systems that show time-reversal symmetries [86]. In all the other cases, a *linearized* Poincaré map can be obtained by computing trajectory *sensitivites* [87, 40, 88].
Consider, for example, a state $\mathbf{x}_0 \in \mathbb{R}^d$ and a discrete map P. The state $\mathbf{x}_1 = P(\mathbf{x}_0)$ is a point on the return section (obtained, for example, by forward simulation from the initial condition $\mathbf{x}_0$). If the fixed point $\bar{\mathbf{x}} \in \mathbb{R}^d$ is asymptotically stable and $\mathbf{x}_0$ belongs to its domain of attraction, then:

$$\lim_{k \to \infty} P^k(\mathbf{x}_0) = \bar{\mathbf{x}} \tag{2-48}$$

and, by definition of fixed point:

$$P^k(\bar{\mathbf{x}}) = \bar{\mathbf{x}} \tag{2-49}$$

First of all, we define an infinitesimally small scalar disturbance ϵ and a perturbation vector $\mathbf{e}_j \in \mathbb{R}^d$ such that:

$$e_j = \begin{cases} \epsilon & \text{if} \quad j = i \\ 0 & \text{otherwise} \end{cases} \qquad \text{for} \quad i = 0, \dots d \tag{2-50}$$

All the elements of the $\mathbf{e}_j$ vector are null apart from the $j - th$ element.
The linearized Poincaré map $\mathbf{DP}^k(\bar{\mathbf{x}}) \in \mathbb{R}^{d \times d}$ can be obtained by performing the three following steps:

1. perturb the considered fixed point $\bar{\mathbf{x}}^k$ with the perturbation ϵ; For this purpose we define the vector $\tilde{\mathbf{x}}^k_j$ such that:
$$\tilde{\mathbf{x}}^k_j = \bar{\mathbf{x}}^k + \mathbf{e}_j \tag{2-51}$$

2. simulate numerically the evolution of the system up to the following intersection with the Poincaré section:
$$\tilde{\mathbf{x}}^{k+1}_j = \mathbf{P}(\tilde{\mathbf{x}}^k_j) \tag{2-52}$$
The location of the Poincaré section is arbitrary, there are, however, practical guidelines to chose where to place it depending on the considered task. In the case of a periodic running gait, for example, the Poincaré section can be conveniently located at the apex of the flight phase; this allows to easily detect a new point on the return map by simply looking for the state on the swing phase that has a null vertical velocity (see Section 3-2-4).

3. compare the deviation from the previous state relative to the intensity of the perturbation ϵ:
$$\mathbf{DP}^k(\bar{\mathbf{x}}) = \frac{1}{\epsilon}\left[\tilde{x}^{k+1}_{i,j} - \tilde{x}^k_{i,j}\right] = \frac{1}{\epsilon}\begin{bmatrix} \tilde{x}^{k+1}_{0,0} - \tilde{x}^k_{0,0} & \cdots & \tilde{x}^{k+1}_{0,d} - \tilde{x}^k_{0,d} \\ \vdots & \ddots & \vdots \\ \tilde{x}^{k+1}_{d,0} - \tilde{x}^k_{d,0} & \cdots & \tilde{x}^{k+1}_{d,d} - \tilde{x}^k_{d,d} \end{bmatrix} \tag{2-53}$$
$\mathbf{DP}^k(\bar{\mathbf{x}})$ gives an estimate of the sensitivity of each individual dimension j from a disturbance in the dimension i.

Once the matrix $\mathbf{DP}^k(\bar{\mathbf{x}})$ is obtained, one can evaluate the limit cycle stability by looking at the largest Eigenvalue λ_{max} of $\mathbf{DP}^k(\bar{\mathbf{x}})$. If $|\lambda_{max}| < 1$ then the map is stable and the periodic solution is also stable (attractor); if $|\lambda_{max}| > 1$ then the periodic solution is unstable (repulsor); $|\lambda_{max}| = 1$ then the periodic solution is marginally stable (saddle) [85].

Limit cycles in legged locomotion have found a wide range of applications [89], especially for the generation of highly dynamic motions where all the stability criteria mentioned so far would fail. They can also be applied to jumping and running gaits (with aerial phases) were the robot switches between ballistic phases and stance phases characterized by high impact forces with the ground. This criterion can be used for humanoid running tasks on a flat ground where we assume the robot's CoM will periodically attain the same values of position, velocity and acceleration at the apex of every other flight phase. The same can be applied to a number of quadrupedal gaits such as, for example, pace, bounding, canter and gallop.
Pioneer in this field was T. McGeer [90] who used limit cycles analysis to prove the existence and the feasibility of passive dynamic walking down small inclines.

The robustness of limit cycles can be measured in terms of their sensitivity to changes of the systems's parameters such as terrain height variations or external disturbances.

Chapter 3

Bounding Gait

3-1 Introduction

Highly dynamic robot mobility has recently gained a lot of interest among robotics researchers. In this Chapter we focus on a specific type of quadrupedal gait, the bounding gait, which becomes useful whenever the robot should cover bigger distances in a shorter time than with a trot. The bounding gait is a dynamical quadrupedal gait characterized by the synchronization of the pair of front and hind legs that lift-off from the ground and touch-down in unison. It is a typical gait of fast runners such as horses, greyhounds and cheetahs which usually employ this kind of run to reach comparable speeds to the ones that can be achieved with the gallop. Most of the robots that were able to perform a bounding gait up today were presenting mass symmetry [91, 92, 93, 10]. However, in nature most of quadrupeds typically show an asymmetrical body mass distribution. In particular, bio-mechanics studies confirmed the asymmetry of quadruped animals from different sides:

- Static measurements on quadruped mammals, mainly dogs and horses, have shown that their Center of Mass (CoM) is always shifted towards the front of the body resulting in an asymmetric structure. A consequence of this is that front limbs bear around the 60% of the animal's weight in *steady state locomotion* [94], [95].
  On the same line, quadruped robots, even if they have a symmetric skeleton, can be equipped with exteroceptive sensors (*e.g.,* different sets of cameras) which are usually positioned in the front to acquire information of the environment, thus shifting the CoM in a similar manner.
- Even in the presence of a perfectly symmetric quadruped, the kinematic limits and the

manipulability properties of front and hind limbs do not allow them to push or pull the trunk with equal ease in any direction [96].

At this respect, the applicability of traditional simplified models used for locomotion (e.g. to generate quadrupedal gaits), such as the *Spring Loaded Inverted Pendulum* (SLIP) model, reach their limits, due to their symmetric structure, compared to the asymmetric nature of the quadrupedal case, even if the motion is restricted to the sagittal plane (2D). This limitation becomes more apparent when the gaits that we want to generate are highly dynamical and include aerial phases such as bounding, galloping or jumping on an obstacle. In this case the dynamics of the orientation of the trunk is highly influenced by the asymmetric distributions of the mass and can not be ignored.

A thorough analysis of an *Asymmetric Spring Loaded Invered Pendulum* (ASLIP) model was introduced by [97] and a further study on the consequences of such model asymmetry was performed by [98]. At the best of our knowledge, other studies, dealing with highly dynamical movements and asymmetric simplified models, can be found in [99, 100, 86].

Many implementations in last decades have already shown impressive dynamic capabilities in performing hops, jumps and dynamic gaits such as Raibert's hoppers and quadrupeds [91]. In more recent years Big Dog by Boston Dynamics has shown the best performances in terms of robustness and agility, however, its implementation details are yet confidential. Specifically about bounding gait the MIT Cheetah has achieved impressive results managing to replan online and jump obstacles while bounding [10, 101, 102].
Despite the excellent existing hardware results in the state of the art, still a lot has to be understood before quadruped robots can imitate quadruped animals and achieve the same agile movements as they do. I believe that extending the range of dynamic motion capabilities of quadruped robots could improve their chances to applied in real world applications such as rescue missions.

In this Chapter we try to perform a step in this direction by understanding how a typically fast and dynamic gait such as bounding can be employed also to perform *omni-directional* movements which might be useful to quickly reach a desired target in a moderately rough and narrow environment (*e.g.,* by passing trough a door without decelerating or by performing agile turning maneuvers with small turning angles). At the same time, staying within the joint torque limits is an important requirement that becomes even more stringent when addressing dynamic motions in moderately rough terrains. It is known indeed that horses during gallop are already very close to peak forces on tendon and bones and having no margin for ground disturbances could result in injury or damage [103]. Similarly, in the case of quadruped robots hitting the torque limits could result in severe loss of stability.

Contribution:
The main contribution of this Section is a framework that implements omni-directional bound-

ing gait on a $80kg$ torque-controlled quadruped robot with an asymmetric mass distribution. We propose the usage of two decoupled criteria to ensure the robot's global stability: in particular we employ the theory of limit cycles and linearized Poincaré maps (as explained in Section 2-2-5) to stabilize the robot's sagittal dynamics while we use the concept of Center of Pressure (CoP) (Section 2-2-2) to prevent the robot from falling during turning maneuvers. In addition, we enhanced the gait's terrain adaptation capabilities, improving the tracking of the desired ground reaction forces (obtained from offline optimization), and addressed the problem of limiting peak torques. Finally we demonstrated the proficiency of the framework showing preliminary experiments on the hardware platform.

Outline:
In Section 3-2 we illustrate a lower dimensional model of the robot which considers the interaction with the ground. This model will be then employed in an offline trajectory optimization problem to devise a baseline stable bounding gait in place (*i.e.,* null linear and heading velocity of the robot's Center of Mass (CoM)). In Section 3-3 we describe the necessary steps that must be undertaken to implement the optimal solution on the full model of the quadruped robot. In Section 3-4 we present simulation results together with preliminary experiments carried out on HyQ, IIT's hydraulic quadruped robot [104] (Fig. 3-1). In Section 3-5 the final discussion of this Chapter is performed where we draft future development directions.

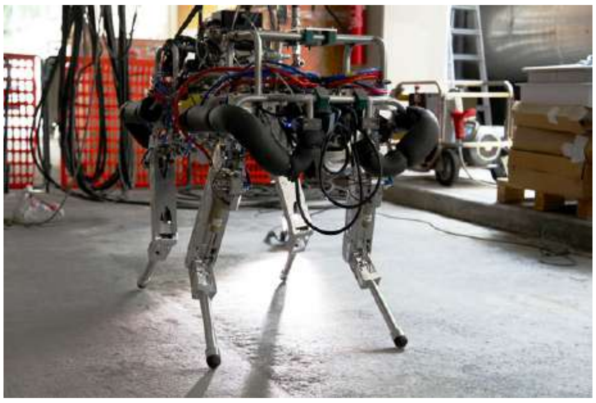


Figure 3-1: HyQ, IIT's quadruped robot [104]

3-2 Optimization

In this Section I will show the required steps to generate a self-stable bounding gait that is flexible enough to react to unexpected external disturbances, terrain height variations or to new commands of an external operator. Therefore the optimization problem that we designed receives the desired horizontal linear speed of the robot and computes the feet trajectories and feedforward torques necessary to realize that motion.
In the subsections 3-2-1 and 3-2-2 we now explain the nomenclature and model that is later used in subsection 3-2-4 to find an optimal solution for the bounding gait.

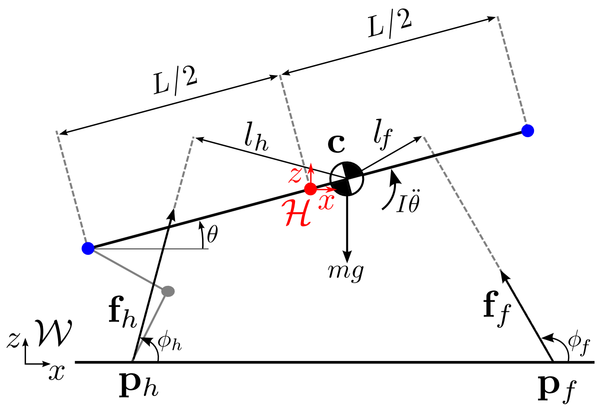


Figure 3-2: Snapshot of the simplified planar model of interaction between the trunk of a quadruped robot in the 2D sagittal plane and the ground. The red dot represents the geometric center of the trunk while the blue dots are the front and hind hips.

3-2-1 An Asymmetric Dynamic Model

Given the above considerations regarding the asymmetrical nature of quadrupeds, we employ the planar simplified dynamic model portrayed in Fig. 3-2. We indicate with $\mathbf{c} = [x, z]^T \in \mathbb{R}^2$ the trunk's CoM position and with $\mathbf{f}_f = [f_{fx}, f_{fz}]^T \in \mathbb{R}^2$ and $\mathbf{f}_h = [f_{hx}, f_{hz}]^T \in \mathbb{R}^2$ the Ground Reaction Forces (GRFs) of front and hind legs respectively. The positions of the front and hind feet are represented with $\mathbf{p}_f = [p_{fx}, p_{fz}]^T \in \mathbb{R}^2$ and $\mathbf{p}_h = [p_{hx}, p_{fh}]^T \in \mathbb{R}^2$. The lever arms of the contact forces with respect to the CoM of the trunk l_f and l_h, are scalars whose amplitude is, in general, different from the quantity $L/2$ (L is the distance between the hind and front hips) and is dependent on the foot contact points $\mathbf{p}_f, \mathbf{p}_h$ and the line of action of the Ground Reaction Forces (GRFs), defined by the angles ϕ_f and ϕ_h. The resulting equations of motion are:

$$\begin{aligned} m\ddot{x} &= f_{fx} + f_{hx} \\ m\ddot{z} &= -mg + f_{fz} + f_{hz} \\ I\ddot{\theta} &= p_{fx}f_{fz} - p_{fz}f_{fx} \end{aligned} \tag{3-1}$$

where the scalar I represents the inertia of the robots trunk computed as mr^2, m is the mass and r is the radius of gyration with respect to the CoM [105].
The generalized coordinates $\mathbf{x} = [x, z, \theta]^T$ fully describe the trunk on the 2D sagittal plane while the forces $\mathbf{f}_f$ and $\mathbf{f}_h$ (where f and h stand for *front* and *hind*) are the interaction forces from the ground acting on the trunk and will be the topic of the next subsection.

3-2-2 Impulse Generation

The accuracy of the model should not influence the performances of the optimization and should be still fast enough to be possibly computed online. For this reason, besides adopting a low dimensional planar simplified dynamic model, we also employ a simplified model of interaction with the environment. Namely we impose the both the normal and tangential components of the GRFs $\mathbf{f}_f$ and $\mathbf{f}_h$ to take on an impulse-like shape as in Fig. 3-3, defined by a 4^{th} order Bézier curves [10]. These curves fit well with the experimental data of the GRFs created by humanoids and quadrupeds [106]. The integral of the overall GRFs over the stance time T_{st} is the impulse expressed by $\mathbf{J}_f \in \mathbb{R}^2$ and $\mathbf{J}_h \in \mathbb{R}^2$:

$$\mathbf{J}_f = \int_0^{T_{st}} \mathbf{f}_f(t)dt \quad \text{and} \quad \mathbf{J}_h = \int_0^{T_{st}} \mathbf{f}_h(t)dt \tag{3-2}$$

The impulsive nature of the GRFs that we assume here implies that this interaction model can not be used to describe the robots behavior in slow or static conditions, *e.g.,* when the robot changes his body pose without lifting the legs from the ground. In such cases the Bézier curves might still be employed for modeling the profile of the contact forces, however, the curve coefficients should be changed in such a way to better fit a quasi-constant value (*e.g.,* step signal).
The interaction model defined in this way can be fully defined by a limited set of parameters which will form the optimization variables of subsection 3-2-4.

- amplitudes on the force profiles: $a_{hx}, a_{hz}, a_{fx}, a_{fz}$. In this way the values of the contact forces $\mathbf{f}_f$ and $\mathbf{f}_h$ can be obtained by multiplication of the impulse amplitudes by the unit impulses $\mathbf{i}_f \in \mathbb{R}^2$ and $\mathbf{i}_h \in \mathbb{R}^2$ of duration T_{st}:

$$\mathbf{f} = diag(a_{fx}, a_{fz}, a_{hx}, a_{hz})\mathbf{i} \tag{3-3}$$

  where: $\mathbf{f} = [\mathbf{f}_f^T, \mathbf{f}_h^T]^T \in \mathbb{R}^4$, $\mathbf{i} = [\mathbf{i}_f^T, \mathbf{i}_h^T]^T \in \mathbb{R}^4$ and:

$$\int_0^{T_{st}} \mathbf{i}_f(t)dt = 1, \qquad \int_0^{T_{st}} \mathbf{i}_h(t)dt = 1. \tag{3-4}$$

- the feet positions: $\mathbf{p}_f, \mathbf{p}_h$

Table 3-1: List of main symbols used in this Chapter

$\mathbf{s} = [\mathbf{x}^T, \dot{\mathbf{x}}^T]^T$	generalized coordinates vector $\in \mathbb{R}^6$
$\mathbf{y}^*$	fixed point
$\mathbf{J}_h$, $\mathbf{J}_f$	impulse given during stance phase of each leg
$\mathbf{f}_h$, $\mathbf{f}_f$	force profile during stance phase of each leg
ϕ_h, ϕ_f	orientation of theGRFs on the plane
θ	pitch angle
φ	yaw angle
$[a_{fx}, a_{fz}]$	force profile of the front limbs
$[a_{hx}, a_{hz}]$	force profile of the hind limbs
$\mathbf{p}_f$, $\mathbf{p}_h$	hind and front feet positions

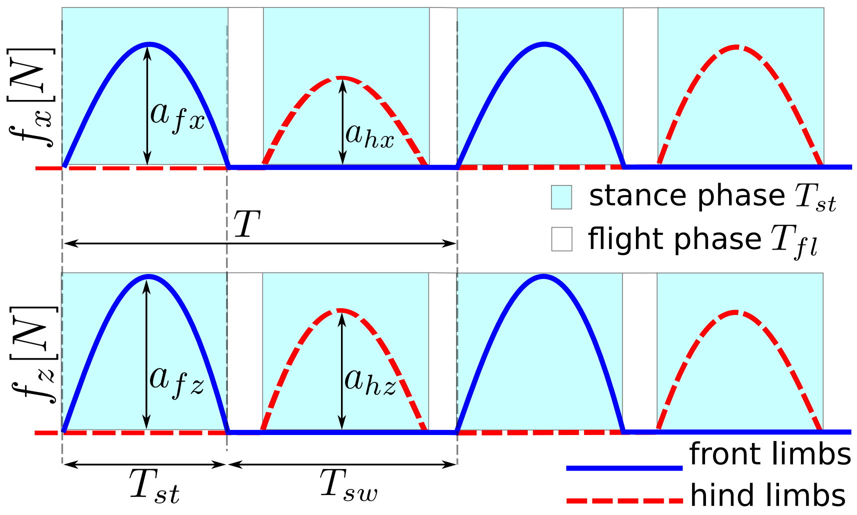


Figure 3-3: Desired tangential (top) and vertical (bottom) GRFs of the front (blue) and hind (red) legs for the generation of a bounding gait in place.

3-2-3 Selection of the Main Gait Parameters

Thanks to the simplified trunk model defined above and the interaction model we can set a complete baseline bounding gait by defining the following independent parameters:

- stance time T_{st};
- z coordinate at the apex of the aerial phase z_{apex};
- desired horizontal speed $\dot{x}$.

We assume here a value of z_{apex} of 7 cm which corresponds to about the 10% of the legs length of HyQ, but this value can be changed to increase the foot clearance from the ground in presence, by instance, of obstacles.
Additional parameters dependent from the above defined variables are (see Fig. 3-3):

- swing time $T_{sw} = 2\sqrt{\frac{2z_{apex}}{g}}$
- cycle time $T = T_{sw} + T_{st}$
- flight time $T_{fl} = \frac{T-2T_{st}}{2}$

3-2-4 Discovery of Periodic Limit Cycles

The decision variables of our optimization problem are the control inputs of the impulse gains in Eq. 3-3, namely the amplitudes of the force profiles at each discretized k^{th} instant:

$\mathbf{u}_k = [a^k_{hx}, a^k_{hz}, a^k_{fx}, a^k_{fz}]$ besides the initial conditions $\mathbf{s}_0 = [\mathbf{x}_0^T, \dot{\mathbf{x}}_0^T]^T$. We decide the initial state of the optimization to be on the apex of the ballistic phase in the middle of the flight phase of duration T_{fl}; this implies a CoM apex height of $z_0 = z_{apex}$ and a null initial speed $\dot{z}_0 = 0$. Since we are interested in obtaining a stable bounding in place we also set $\dot{x}_0 = 0$; in this way the only initial conditions left to be determined by the optimization are θ_0 and $\dot{\theta}_0$. The goal is to minimize the cost function $L(\theta_0, \dot{\theta}_0, \hat{\mathbf{u}})$ at each time step where:

$$\hat{\mathbf{u}} = [a^0_{hx}, a^0_{hz}, a^0_{fx}, a^0_{fz} \dots a^N_{hx}, a^N_{hz}, a^N_{fx}, a^N_{fz}] \tag{3-5}$$

The vector containing all the states at k^{th} instant is $\hat{\mathbf{s}}$:

$$\hat{\mathbf{s}} = [x_0, y_0, \theta_0, \dot{x}_0, \dot{y}_0, \dot{\theta}_0 \dots x_N, y_N, \theta_N, \dot{x}_N, \dot{y}_N, \dot{\theta}_N] \tag{3-6}$$

where N is the number of time samples used to discretize a whole cycle of the bounding gait of duration T. For an integration step $t = 0.001s$ and a cycle time $T = 0.4s$ one obtains $N = T/t = 400$ samples.
The optimization problem is thus defined as:

$$\mathbf{y}^* = \min_{\theta_0, \dot{\theta}_0, u} \quad L(\theta_0, \dot{\theta}_0, \hat{\mathbf{u}}) = \sum_{k=1}^{N} (\theta_k^2 + \dot{\theta}_k^2 + \mathbf{u}_k^T \mathbf{u}_k) \tag{3-7}$$

subject to the hard constraints:

- dynamic model: $\mathbf{s}_{k+1} = \mathbf{f}(\mathbf{s}_k, \mathbf{u}_k)$ with: $k = 1, 2 \dots N$
- periodicity: $\mathbf{s}_0 = \mathbf{s}_N$
- GRFs limits: $u_{min} \leq \mathbf{u}_k \leq u_{max}$

The terms $\theta, \dot{\theta}$ and $\hat{\mathbf{u}}$ in the cost function $L(\theta_0, \dot{\theta}_0, \hat{\mathbf{u}})$ were employed in order to reduce the rocking motion to the minimum needed while concurrently limiting the impulses amplitude. It is interesting to point out that the final values of the resulting impulses returned by the solver respect the principle of conservation of linear momentum for both the vertical and horizontal components, even if this principle is not explicitly enforced:

$$J_{hz} + J_{fz} = mgT \tag{3-8}$$

$$J_{hx} = -J_{fx} \tag{3-9}$$

We found a set of solutions corresponding to different values of the input parameter T_{st} in the range between $50ms$ and $300ms$ and analyzed the stability of the different periodic limit cycles obtained. Such limit cycles can be considered as a fixed point $\mathbf{y}^* = \{\theta_0, \dot{\theta}_0, \hat{\mathbf{u}}\}$ in a Poincaré section. We choose the Poincaré section in coincidence of the apex of the ballistic phase because, as mentioned in Section 2-2-5, this is a convenient choice for the set up of the

optimization problem in that the initial conditions on the linear position and velocity of the CoM **c** are known and the only unconstrained initial conditions are θ_0 and $\dot{\theta}_0$. In this way the solution $\mathbf{y}^*$ represents a fixed point on this map [87].
This analysis of the resulting phase plots (reported in Fig. 3-4) shows that all the eigenvalues λ_i of the discrete linearized Poincaré map have a magnitude bigger than 1 ($\lambda_i > 1$) meaning that the found periodic limit cycle are unstable in open-loop for all the analyzed duty factors D (which is the ratio between the stance time T_{st} and the gait period T such that: $D = T_{st}/T$).
The stabilization of these periodic limit cycles can be performed in two ways:

1. by state feedback;

2. by delaying/anticipating the force impulses.

The former method is inherently present in the joint-space active impedance controller already implemented on acHyQ [107]. The latter option is, therefore, not further investigated in this Chapter although it offers promising perspectives in terms of robustness with respect to the torque limits: the online adaptive adjustment of the swing time duration in presence of, for example, terrain height variations may stabilize the system without need of increased impulse amplitudes during the stance phase.

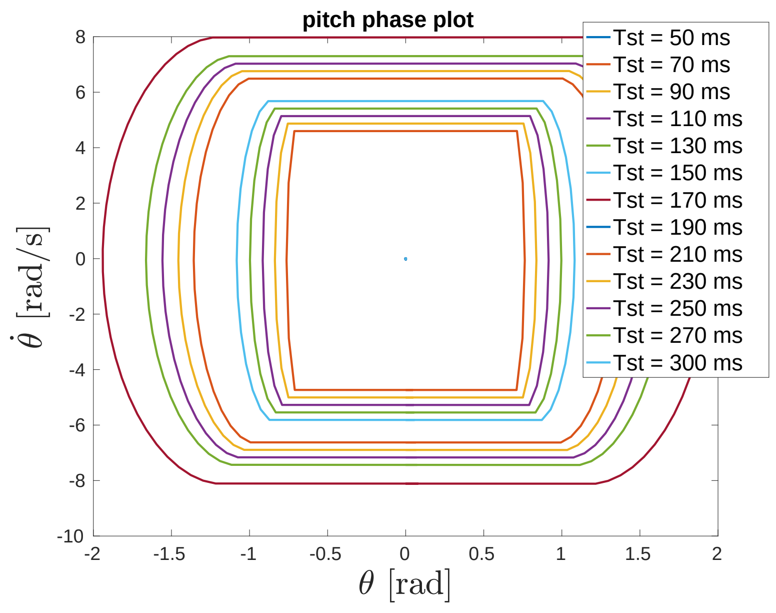


Figure 3-4: Plot of the various periodic limit cycles which the optimizer yields for a range of stance times T_{st} from $50ms$ to $300ms$.

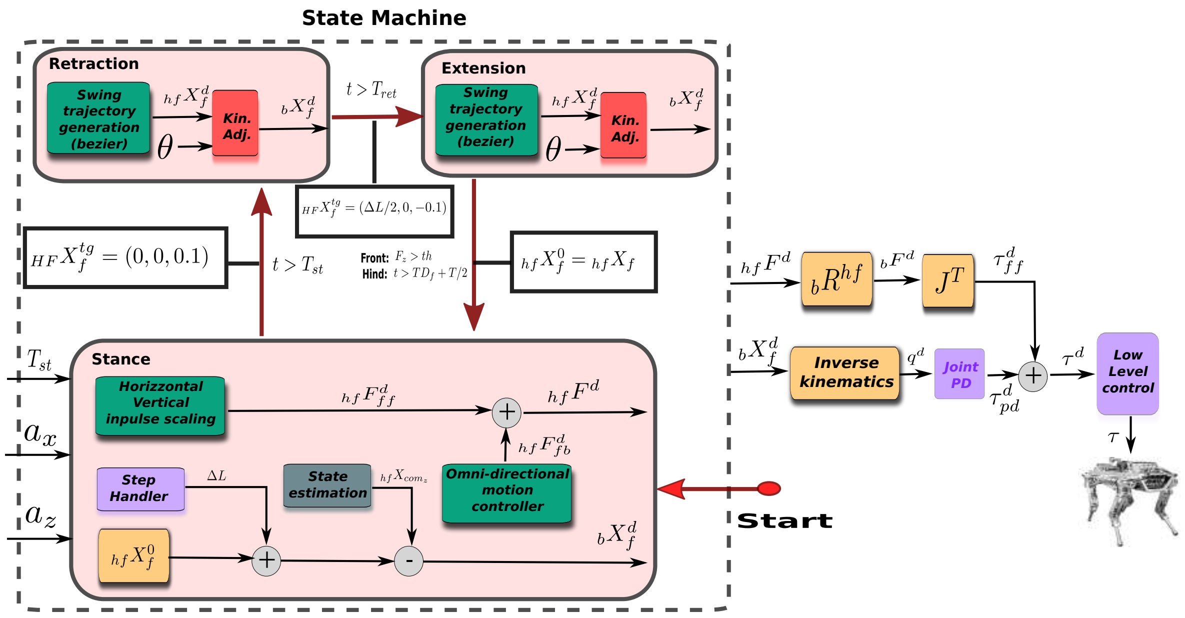


Figure 3-5: Block diagram showing the structure of the proposed controller.

3-3 Implementation

As outlined above in order to achieve a stable bounding in a real system the feedforward force profile obtained in 3-2 is not sufficient and state feed-back is needed in order to make the baseline limit cycles stable. It is then necessary to implement a number of other *correction mechanisms* that generate in a heuristic manner all those references that are not specified by the offline Poincaré analysis explained in the previous Section. These are elements are:

- State machine: it coordinates the switching between the legs' phases;
- Feet swing trajectory generator;
- Haptic touchdown;
- Kinematic adjustment;
- Omni-directional motion controller;
- CoP-based lateral stabilization.

These different blocks that compose the overall control framework are explained in the following subsections. An overview of this structure is given in Fig. 3-5.

3-3-1 State Machine

The state machine (see Fig. 3-5) is the time coordinator of the control framework which ensures that the feedforward impulses computed by the optimizer are deployed in the correct instant. The swing phase is here divided into two main subphases, the *retraction* and the *extension* of duration T_{rt} and T_{ext} such that $T_{sw} = T_{ext} + T_{rt}$. Therefore on each cycle the state machine sequentially goes through three phases:

1. *Stance:* the triggering of this phase is performed with different criteria for the front and hind legs. In particular the stance of the front legs is set when the contact with the ground occurs on both limbs on a haptic basis; the stance of the hind legs is instead triggered when a time corresponding to $T/2$ has elapses. This partition highlights the different roles of the front and hind legs: the front limbs take on their *stabilizing* role against possible model uncertainties or external disturbances, by *sensing* the terrain, allowing the robot to cross moderately rough environments.
   The hind legs, instead, provide the *propelling* power necessary to achieve forward motion. The triggering condition for the stance of the hind legs, after a time $T/2$, ensures on average that the desired timing of the bounding gait is achieved and ensures that trunk reaches the desired pitch and vertical height before the pushing force is propelled.

2. *Leg retraction:* it starts after the time T_{st} has passed. During this phase the feedforward torque is set to zero and the desired swing trajectory is achieved by PD control, in the meanwhile the foot is raised from the ground level to the maximum retraction height z_{rt} (see Fig. 3-6). The quantity z_{rt} is always fixed for every step, which allows to recover from possible accumulated errors in the previous foot cycle.

3. Leg extension: similarly to the previous phase, the extension of the legs is set by the *time-scheduler* when a period of T_{rt} has elapsed from the lift-off instant (see Fig. 3-5) and it finishes after a time of T_{ext} milliseconds in the case of the hind limbs, while for the front limbs it can be extended till when the haptic touch down is triggered.

Each phase distinguishes itself, besides the controller and the triggering criterion, also for the different feet trajectory generated.

3-3-2 Feet Trajectory Generation

Each phase of the state machine generates independently the feet trajectory ${}_{\mathcal{W}}\mathbf{p} = [p_x, p_y, p_z] \in \mathbb{R}^3$ (in the *world frame* $\mathcal{W}$) for the time interval of interest. From now on I will drop the pedex $\mathcal{W}$ and all the feet positions will refer to the world frame unless differently specified and will be indicated by the variable $\mathbf{p}$. The values of the p_x, p_z components are those specified in the offline optimization problem. This implies that, even in conditions of bounding in place with no forward velocity, an oscillatory motion will also be present in the longitudinal direction of robot (along the x axis) besides the oscillations on the vertical axis and on the pitch angle. The p_y component for bounding in gait is instead corresponding to the value of Hydraulically actuated Quadruped (HyQ)'s default configuration. The definition of the feet swing trajectory is done employing a Bézier curve of 4^{th} order. To ensure continuity and smoothness in the transitions among the three phases we impose the initial position and initial velocity of the foot of the following phase to be equal to the final position and final velocity of the previous phase.

During the stance phase the foot performs a horizontal stroke:

$$\Delta L = \dot{x} T_{st} \tag{3-10}$$

where $\dot{x}$ is the robot's desired velocity defined by the external operator. At touch down the horizontal speed of the feet expressed in the base frame is indeed of the same amplitude of the robot's speed but with opposite sign. Continuity and smoothness conditions are respected by choosing suitable values of the Bézier parameters. The resulting foot trajectory in the horizontal frame $\mathcal{H}$ for a speed of $\dot{x} = 0.5m/s$ is shown in Fig 3-6. The horizontal frame $\mathcal{H}$ is the reference frame that shares the same yaw and origin of the base frame but is rotated in pitch and roll like the world frame (hence horizontal).

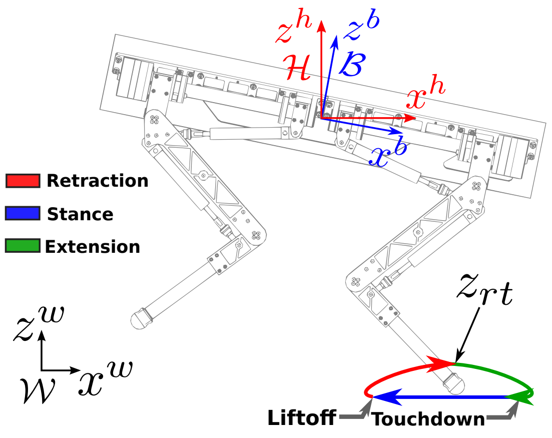


Figure 3-6: Foot trajectory for a bounding speed of $\dot{x} = 0.5m/s$.

3-3-3 Haptic Touchdown

As mentioned in Section 3-3-1 the triggering of the front legs takes place in an event-driven fashion when they actually touch down the ground. This makes our controller more robust against possible external disturbances or changes in the terrain that might accumulate after a few cycles and would mean the force impulse to be applied delayed or in advance (*e.g.*, in the air) causing CoM tracking errors and possible loss of stability.
This choice highlights the sensing function of forelimbs in the discovery of the environment in the direction where the robot is heading to.

3-3-4 Kinematic Adjustments

A desired foot trajectory $\mathbf{p}(t)$ during swing phase is obtained as mentioned above while the desired foot trajectories during stance phase is obtained by double integration of the reference force profiles $\mathbf{u}(t)$ provided by the offline optimization. These trajectories have been planned in the *world frame* $\mathcal{W}$, concurrently with the desired CoM trajectory $\mathbf{c}^d(t)$ and the desired trajectory of the trunk orientation $\boldsymbol{\theta}^d(t)$. The desired position of the base frame $\mathbf{b}^d \in \mathbb{R}^3$ can be easily obtained for every time instant by considering the fixed offset between robot's CoM and the origin of the base frame, as in the case of the single rigid body model (see Section 2-1-2). It is then necessary to map these quantities in a robo-centric reference frame. In particular we need to map them into the base frame $\mathcal{B}$ because in that frame it is defined the inverse kinematics of HyQ that is used to compute the reference joint positions that are then provided to the whole-body controller [28]. The standard transformation from world frame to base frame $\mathcal{B}$ is performed using the desired trajectory of the robot and it can be performed offline. Another possibility, however, is to perform the mapping online and to employ the actual state of the robot (*e.g.,* measured base position and velocity in the world frame $\mathbf{b}^m$ and $\dot{\mathbf{b}}^m$) to partially compensate external disturbances [108]. This mapping is performed in two steps, a transformation from world frame $\mathcal{W}$ to horizontal frame $\mathcal{H}$ first and a transformation from horizontal frame $\mathcal{H}$ to base frame later $\mathcal{B}$:

- *linear kinematic adjustment:* this is the mapping of the desired foot trajectory $\mathbf{p} \in \mathbb{R}^3$ from the world frame $\mathcal{W}$ to the *horizontal frame* $\mathcal{H}$:

$$_{\mathcal{H}}\mathbf{p} = \mathbf{p} - \mathbf{b}^{ref} \tag{3-11}$$

$$_{\mathcal{H}}\dot{\mathbf{p}} = \dot{\mathbf{p}} - \dot{\mathbf{b}}^{ref} \tag{3-12}$$

  where:

$$\mathbf{b}^{ref} = \mathbf{b}^d + \alpha(\mathbf{b}^d - \mathbf{b}^m) \tag{3-13}$$

- *rotational kinematic adjustment:* the feet's current positions in the base frame $_{\mathcal{B}}\mathbf{p}$ can be computed from the corresponding value $_{\mathcal{H}}\mathbf{p}$ through the rotation matrix $^{\mathcal{B}}\mathbf{R}_{\mathcal{H}}(\boldsymbol{\theta}^{ref})$

$$_{\mathcal{B}}\mathbf{p} =^{\mathcal{B}} \mathbf{R}_{\mathcal{H}}(\boldsymbol{\theta}^{ref}) \cdot_{\mathcal{H}} \mathbf{p} \tag{3-14}$$

$$_{\mathcal{B}}\dot{\mathbf{p}} =_{\mathcal{B}} \dot{\mathbf{R}}_{\mathcal{H}}(\boldsymbol{\theta}^{ref}) \cdot_{\mathcal{H}} \mathbf{p} +_{\mathcal{B}} \mathbf{R}_{\mathcal{H}}(\boldsymbol{\theta}^{ref}) \cdot_{\mathcal{H}} \dot{\mathbf{p}} \tag{3-15}$$

  where:

$$\boldsymbol{\theta}^{ref} = \boldsymbol{\theta}^d + \beta(\boldsymbol{\theta}^d - \boldsymbol{\theta}^m) \tag{3-16}$$

  and through the desired $\boldsymbol{\theta}^d$ and the estimated $\boldsymbol{\theta}^m$ orientation of the robot:

$\alpha, \beta \in [0, 1]$ are tuning parameters that can change the amount of error compensation performed by the kinematic adjustment. For $\alpha = \beta = 0$ the linear and angular base reference trajectory will correspond to the desired one; if instead $\alpha = \beta = 1$ the reference base pose will correspond to the estimated one and the homogeneous transformation will make sure that the predefined foot position in the world frame $\mathbf{p}$ is achieved, regardless of possible mismatches between the planned and the estimated base pose. The benefit of the kinematic adjustment, can be therefore summed up in the two following points [108]:

1. keep foot *clearance*: during the feet *swing phase* the strategy explained above ensures that the desired foot clearance from the ground is achieved, regardless of possible deviations between the desired and the estimated pose of the base. This is particularly beneficial in the case of blind locomotion where no exteroceptive information about the surrounding environment is provided to the robot and, therefore, ensuring a predefined minimum clearance from the ground might be crucial to avoid possible unexpected obstacles;

2. base pose and feet *decoupling*: during the *stance phase*, thanks to this strategy, it is possible to ensure that possible disturbances in the feet (*e.g.,* foot slippage or unexpected obstacles) do not affect the desired behavior of the robot's trunk. This is because a *full* kinematic adjustment (*i.e.,* $\alpha = \beta = 1$) will completely remove possible conflicts between the joint-space impedance of the legs and the virtual model [8] of the trunk.

3-3-5 Omni-Directional Motion Control

The computed limit cycles can be stabilized by exploiting the impedance control that we implemented at the joint level. In [10] it is shown that a state feedback is able to stabilize the periodic limit cycle. It is possible to show that a joint impedance controller has the same effect; we do not report the details because not relevant with the goal of this Section [21].
The cyclic motion obtained by the offline optimization is a bounding in place and represents a baseline that does not ensure motion in any specific direction. For this we implemented a forward/lateral motion controller and a yaw controller.
In order to achieve the desired velocity of the robot we implemented a yaw controller, a forward motion controller and a lateral motion controller; all of them are based on the principle of linear momentum conservation:

- *Forward motion controller:* since we are initially interested in assigning a given forward motion speed $\dot{x}_{des}$ to the robot rather than a precise position we designed the forward motion controller as a Derivative controller of the form:

$$a_{ix} = a_{ix}^{*} + K_{d,\dot{x}}(\dot{x}_{des} - \dot{x}) \tag{3-17}$$

  where $i = 1, 2, 3, 4$ is the leg number and a_{ix}^{*} is the optimal amplitude which is obtained by the optimization of Section 3-2-4. Since the values a_{ix}^{*} of the front legs (a_{fx}^{*}) are negative and for the hind legs (a_{hx}^{*}) are positive this will cause an overall net linear momentum in the x direction of amplitude $2K_{d,\dot{x}}(\dot{x}_{des} - \dot{x})$.

- *Lateral motion controller:* the lateral motion controller is designed in the same fashion of the forward motion controller:

$$a_{iy} = K_{d,\dot{y}}(\dot{y}_{des} - \dot{y}) \tag{3-18}$$

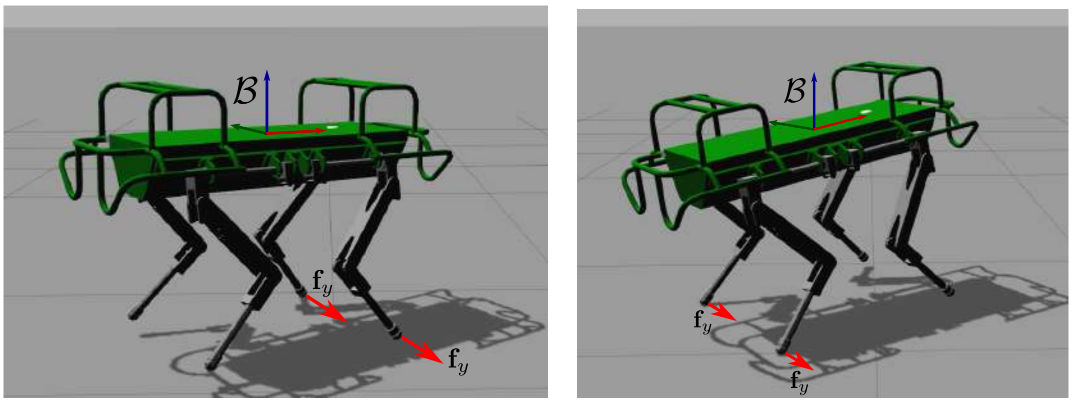


Figure 3-7: *Lateral controller:* a sideways movement is achieved by applying the same force profile to all the four legs during stance.

where $i = 1, 2, 3, 4$ is the leg number. In this case $a^*_{iy} = 0$ by defaults since our dynamic model of section 3-2-1 was restricted to the 2D plane.
In the case of the lateral motion controller all the force profiles will have the same sign and thus they will all propel the robot in the same direction to reach the desired lateral speed $\dot{y}_{des}$ (Fig. 3-7).

- *Yaw controller:* this controller steers the yaw angle φ by creating a net torque around the vertical z of the horizontal frame $\mathcal{H}$ by realizing a lateral force $\mathbf{f}_y$ of the similar profile as the sagittal forces $\mathbf{f}_x$ and $\mathbf{f}_z$ but of different amplitude (Fig. 3-8) and opposite sign for front and hind limbs. This causes a overall torque parallel to the z axis of the horizontal frame $\mathcal{H}$ that allows the robot to adjust its heading velocity.

  We implemented the yaw controller in the form of a Proportional-Derivative controller where for each i^{th} leg the amplitude of the later impulse is computed $a_{i\varphi}$ as:

$$a_{i\varphi} = \frac{K_{p,\varphi}(\varphi_{des} - \varphi) - K_{d,\varphi}(\dot{\varphi}_{des} - \dot{\varphi})}{{}_{\mathcal{H}}p^i_x} \tag{3-19}$$

  The ${}_{\mathcal{H}}p^i_x$ is the x position of the $i-th$ foothold in the horizontal frame $\mathcal{H}$, which represents the lever arm of the forces f^i_y with respect to the CoM of the robot. Notice that, since this component has a different sign for the front and hind legs, this will cause impulses of opposite sign on the lateral forces as desired (while in the case of the lateral motion controller the lateral impulses are all in the same direction defined by $\dot{y}_{des} - \dot{y}$).

In addition to the force impulses performed during the stance phase, it is also important to avoid that the robot stumbles (*e.g.,* during the feet swing phases); for this reason a suitable step height must be enforced as explained in Sec. 3-3-4. The stroke length needs to be

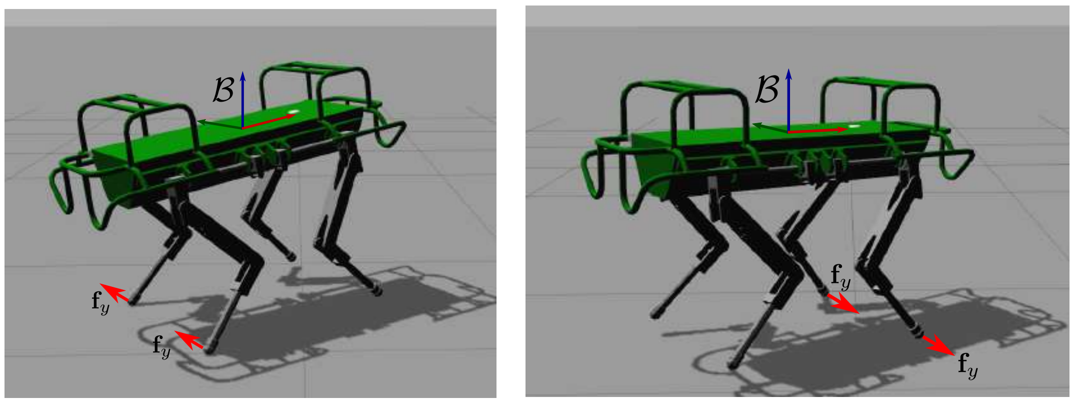
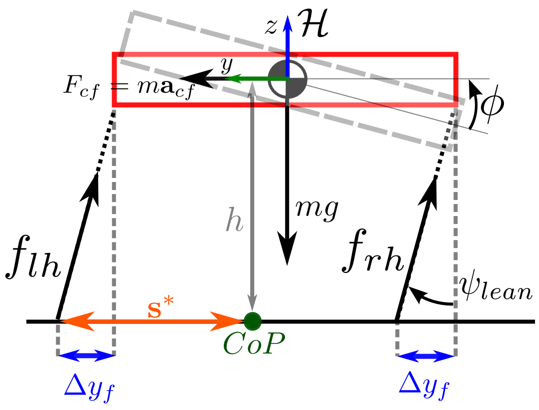
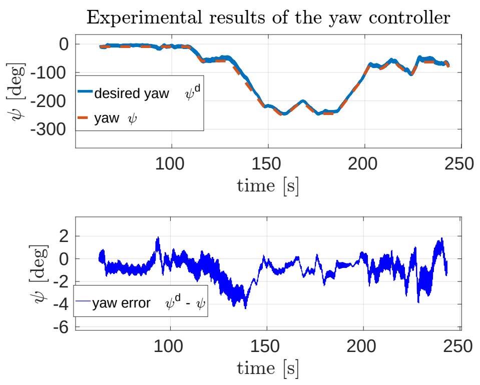


Figure 3-8: *Yaw controller:* a steering of the yaw angle is achieved by applying force profiles of opposite signs to front and hind legs.



(a) Sketch of the frontal section of a generic quadruped (from the back) performing a turn to the right subject to centrifugal forces.



(b) Yaw controller performance (tracking of the heading speed) from experimental data.

Figure 3-9: CoP based lateral stabilization and heading speed controller tracking performances

adjusted according to the speed up to a certain linear speed. In particular the stroke length of each single step is adjusted using Eq. 3-10. However, when the speed becomes too large then the corresponding stroke length exceeds the limits of the leg's workspace and the joint kinematic limits are reached. In this case the stroke length must thus be fixed to the maximal feasible value and the T_{st} should be reduced to match the desired foot speed during the stance. This was not an issue for the speed ranges of the current implementation (up to $0.6 m/s$), however, we leave to future works the implementation of this feature.

3-3-6 Center of Pressure Aware Lateral Stabilization

The above mentioned speed controllers vary the amplitudes of the baseline bounding gait on place, computed offline, in order to adapt it to the user-defined linear and angular velocities. This changes the orientation of the GRFs and, consequently, affects the CoM dynamics and overall dynamic stability of the robot. While we noticed that the sagittal dynamics of the bounding gait (on the (x, z) plane of the horizontal frame $\mathcal{H}$) was successfully stabilized by the joint-space impedance controller, we also realized that the lateral stability was marginally stable, especially during turning maneuvers.

Furthermore, during fast turning maneuvers with high angular and forward linear velocity of the base, centrifugal accelerations a_{cf} will arise that should be appropriately taken into account (see Fig. 3-9a)

$$a_{cf} = \frac{{}_{\mathcal{H}}\dot{x}^2}{r} \tag{3-20}$$

where r is the instantaneous radius of curvature $r =_{\mathcal{H}} \dot{x}^d/\dot{\varphi}^d$, ${}_{\mathcal{H}}\dot{x}^d$ is the desired forward speed in the horizontal frame $\mathcal{H}$ and $\dot{\varphi}^d$ is the desired angular speed.
Considering the effect of the new impulse amplitudes and of the centrifugal acceleration a_{cf}, the CoP (defined on the line between the two stance legs) will then be shifted outwards with respect to the turn, thus reducing the lateral stability margin represented by the distance s between the CoP and the foot along the support line of the double stance phase (*e.g.,* Fig. 3-9a). This also results in unloading the internal leg during the turn. Our strategy consists in moving the feet laterally by an offset Δy_f to restore the desired lateral stability margin s^* and, as a consequence, to re-equilibrate the force distribution between the two stance legs. As a consequence of this offset $\Delta y_f = (a_{cf}h)/g$ the legs will take on a certain leaning angle:

$$\varphi_{lean} = atan(\frac{\Delta y_f}{h}) \tag{3-21}$$

where h is the height of the CoM of the robot. The legs then align with the contact forces (whose lateral component compensates for a_{cf}) and no extra torque is applied on the Hip Abduction/Adduction joints (see Fig. 3-9(b)). Furthermore, it can favorable (in terms of energy efficiency and robustness with respect to the joint kinematic limits) to roll the trunk by the same leaning angle [109, 110] s. t. $\phi = \varphi_{lean}$. The legs, in this way, get aligned with the main axis of the frontal section of their manipulability ellipsoid.

3-4 Experimental Results

The experimental results presented in this Section are composed of data collected both in simulation and on the real hardware. A few samples of the collected data are shown in the

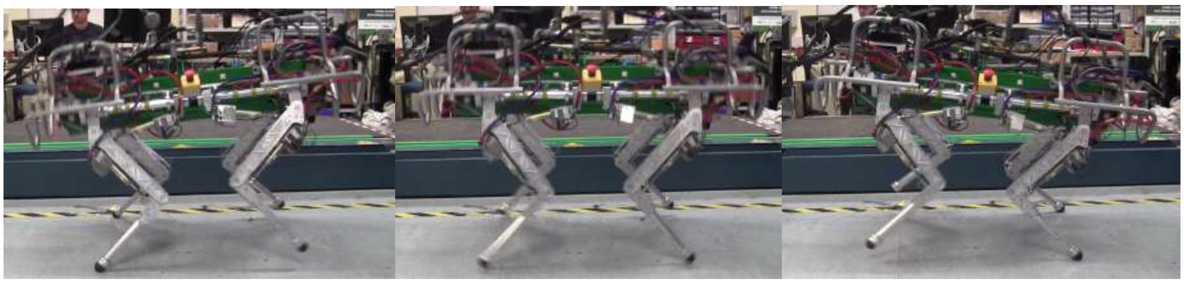
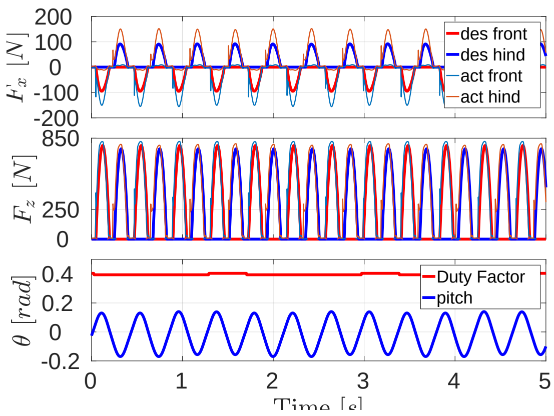
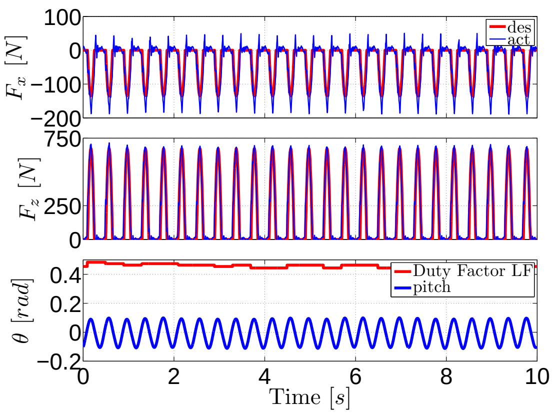


Figure 3-10: Snapshots of an experiment of self-stable bounding gait performed on the HyQ (we can see a flight phase in the middle picture). The time passing between two neighboring frames is of 70 ms.



(a) Simulation results with $D = 0.4$.



(b) Experimental results with $D = 0.45$.

Figure 3-11: Simulation and Experimental results recorded while having HyQ bound on flat terrain. The reported data show the vertical tangential contact force, the vertical contact force, the duty factor and the pitch dynamics (from top to bottom) for the left-front leg.

accompanying video[1].

All simulations and experiments own a transient time where the robot is accelerated from the static configuration to a cyclic steady state were the robot is bounding in place. In order to safely carry out this initial phase we gradually increase the feedforward impulses from zero to the steady state values which have been computed offline by the optimizer (as in Section 3-2-4) in the time span of 10 gait cycles. Similarly also the leg retraction is gradually increased from zero to the desired value in the same time interval.

Once this initial transient has finished the robot enters the steady state regime where it can bound on place in a stable manner. After this, the motion controllers of section 3-3-5 can be activated to move the robot in an arbitrary direction. Plots of this condition are given in Fig. 3-11a: in the upper plot it is possible to see for both front and hind legs the horizontal feedforward forces $\mathbf{f}_x$ and the generated ground reaction forces in the same direction. In

[1] https://www.youtube.com/watch?v=005BMWixqsQ

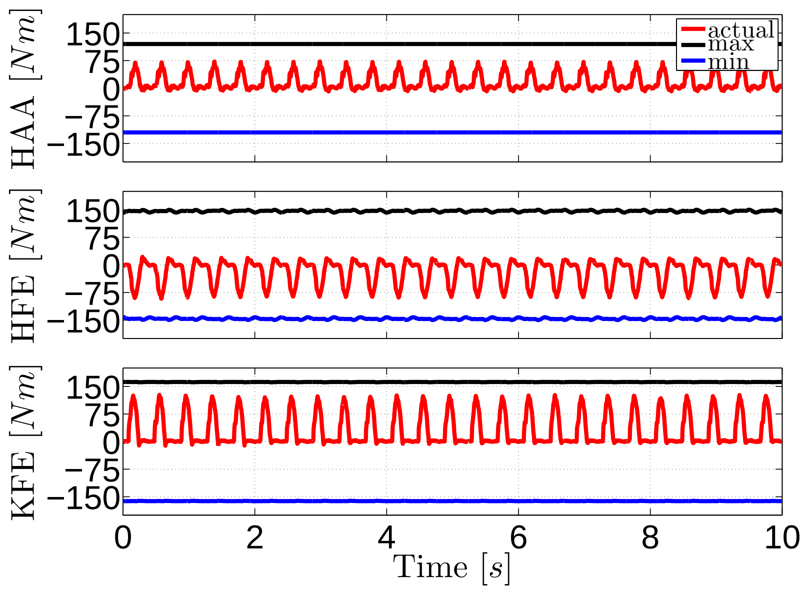


Figure 3-12: Joint torques of the left front limb (LF) in red: Hip Abduction-Adduction (HAA) above, Hip Flexion-Extension (HFE) in the middle and Knee Flexion-Extension (KFE) in the bottom.

particular the shown data refer to the left front and left hind legs.
In the middle plot we can see that the amplitude of the vertical feedforward force profiles of the left front leg a_{fz} is slightly higher than the one of the left hind leg a_{hz}, this is the result of the asymmetric model that keeps into account the fact that the CoM of HyQ's trunk is shifted $2cm$ to the front. This offset of the CoM position highlights the need of a precise model identification [111, 112, 113]
The use of the kinematic adjustment significantly mitigates the spike at touch down which is due to the non zero vertical velocity of the feet at the end of the extension phase. Thanks to the mapping of the generated feet trajectories from world frame to base frame, as in Section 3-3-4, the effects of the vertical and angular velocity of the CoM are cancelled out and the vertical velocity of the feet is strongly reduced, resulting in a softer touch down.
The limit cycle periodicity of the pitch and the almost constant duty factor shown in the lower plot of Fig. 3-11a demonstrate that the system is successfully stabilized in simulation.

We compare the data of the simulation we the data of Fig. 3-11b recorded during the hardware implementation. We find out that the same limit cycle stability of simulation is successfully obtained also on the hardware (see Fig. 3-10 for three snapshots of HyQ bounding). The shown data refer to a bounding gait of total gait cycle of $400ms$ and a duty factor about $D = 0.45$. The retraction of the leg adopted for this experiment was of $7cm$, about one tenth of the legs length.
Fig. 3-12 shows that joint torques are safely far from their limits.

3-5 Summary

In this Chapter we have shown the method and the preliminary results of our approach for the generation of a stable bounding gait. The control framework was tested on HyQ, demonstrating a successful omni-directional bounding in a range of speeds between 0 and $0.6m/s$. Some implementation details such as the kinematic adjustment and the haptic touch down have turned out to play a role of paramount importance in the reduction of impact losses and in the increase of robustness.
The following main concepts and contributions have been presented in the course of this Chapter:

1. an omni-directional bounding gait has been developed in such a way to be able to instantaneously react to possible disturbances or to variations in the desired velocity set by the external operator;
2. the stabilization along the sagittal plane was achieved by means of an offline optimization that determined intensity of the force impulses needed to achieve a stable limit cycle. The resulting stability was evaluated through the eigenvalues of the linearized Poincaré map.
3. the lateral stabilization, in case lateral translations and/or abrupt turns in the yaw direction, was achieved with a stability criterion based on the CoP (defined along the support line of the two stance legs).

The current approach is valid for flat terrain but has also been tested on moderate height variations of the ground, in the order of 5% of the leg length (up to $4cm$); in the future works we intend to test the robustness of our approach with different ground stiffness and damping, with consistent changes in the terrains level and slopes.
Besides this we intend to tackle the problem of hyperdynamic turning motions, *e.g.,* quickle rotate by controlling the roll of the trunk while bounding on curved trajectory.
The optimization implemented in this Chapter is computed offline and provably stabilizes the bounding gait at different speeds allowing the robot to instantaneously choose any arbitrary direction. In the future we intend to embed the optimization in the controller in a Model Predictive Control (MPC) manner in order to continuously adjust the foot position to minimize joint torques.
Another important feature to be added ot the future works is an adaptation to rough terrains by changing the switching time of the back legs.

Chapter 4

Wrench-Based Feasibility Analysis of Legged Locomotion

4-1 Introduction

Legged locomotion in rough terrains requires the careful selection of a contact sequence along with a feasible motion of the Center of Mass (CoM). In case of an unexpected event (e.g. changes in the terrain conditions, human operator commands, external force disturbance, inaccuracies in the state estimation and in the terrain mapping, etc.) replanning is an important feature to avoid accumulation of errors. As a consequence, ideal motion planners for complex terrains face the conflicting requirements to be fast but accurate. Approaches that use simplified dynamic models are fast but less accurate because they only capture the main dynamics of the system [20]. On the other hand, other approaches that use the whole-body model of the robot and provide particularly accurate joint torques and position trajectories are not suitable for online applications in arbitrary terrains because of their high computational complexity. A third option consists in offline learning primitives and behaviors generated with the more accurate whole-body models that can be later quickly realized in real-time [51].

The present Chapter tackles this issue using simplified dynamic models that still contain sufficient details of the system. The use of the centroidal dynamics [27] coupled with the Contact Wrench Cone (CWC)-based planning represents a step in this direction, allowing to remove the limitation of having coplanar contacts (as for Zero Moment Point (ZMP) based approaches) and thus increasing the complexity of motions that can be generated [66, 68]. This has also led to the formulation of algorithms that can efficiently verify robots stability in multi-contact scenarios [76, 114, 115]. Such approaches however still fail to capture some

properties of the robot such as the actuation limits, the joints kinematic limits and the possible self-collisions. These properties become more and more important with the increasing complexity of the environment (motion in confined spaces) and I believe that they should not be neglected in motion planning. To the best of my knowledge, while actuation constraints have been considered at the control level [116, 117], this is the first time that a framework for the formulation of actuation consistent online motion planners for dynamic motion in rough terrains is provided.
As later explained, a key aspect of this strategy consists in devising CoM trajectories that are guaranteed to respect the actuation and friction constraints, without explicitly optimizing neither the joint torques nor the contact forces.

Contribution:
In this Chapter we address the problem of devising actuation-consistent motions for legged robots and, in particular, we propose the four following contributions: a) first, we introduce the concept of Actuation Wrench Polytope (AWP) which complements the CWC, adding the robot-related constraints such as its (configuration dependent) actuation capabilities. The consideration of both the environment-related constraints (the CWC) and the robot-related constraints (the AWP) leads to the definition of a second convex polytope that we call Feasible Wrench Polytope (FWP). This can be seen as a development of the Grasp Wrench Set (GWS) previously proposed for robotic grasping [72]. Disregarding the constraints due to self-collision and kinematic joints limits, the Feasible Wrench Polytope (FWP) can then be used as a sufficient criterion for legged robots stability; b) second, we exploit recent advancements in computational geometry [118, 119] to compute the vertex-description ($\mathcal{V}$-description) of the FWP, drastically reducing the computation time with respect to the double-description ($\mathcal{D}$-description) based methods [120]; c) third, we adapt the concept of vertex-based *feasibility factor* to the FWP, as in [72], to evaluate *online* the feasibility of a motion plan for legged robots. d) Finally, we exploit this feasibility factor for the *online* generation of CoM trajectories that are statically stable and actuation-consistent.

Outline:
The remainder of this Chapter is organized as follows: we first discuss the previous research in the field of wrench based feasibility analysis with a special consideration for the robot stability and actuation-consistency (4-2). We then introduce the computation of the AWP [1] and an efficient strategy to calculate the $\mathcal{V}$-description of the FWP (4-3). Section 4-4 introduces the FWP-based *feasibility metric* and Section 4-5 describes how this can be used for *online* motion planning. Section 4-6 presents the simulations and experimental results we obtained by implementing our strategy on the Hydraulically actuated Quadruped (HyQ) robot [121]. Finally, 4-7 draws the conclusions with a brief discussion on the results and on future developments.

4-2 Related Works

Wrench based feasibility analysis is not a novel idea in robotics. In the field of Cable-driven Parallel Robots (CPR) the set of all the configurations that can be realized respecting the maximum tension in the ropes is indicated under the name of Wrench-Feasible Workspace (WFW) [122], [123]. The WFW is used to analyze the robot's capability to carry loads, but it does not consider constraints that might arise from the interaction with the environment, such as unilaterality and friction. The idea of modeling the wrench admissible region is also present in the field of mechanical fixtures and tolerance analysis [124] where reciprocity of twists and screws is exploited to characterize the mobility conditions of any couple of faces in tolerance chains.

On the other hand, in the robotic grasping community it is common to consider sets of wrenches respecting frictional constraints [71]. When considered, actuation limits take the form of an upper bound on the magnitude of the normal contact force. Composing the contribution of each contact friction cone, a GWS can be defined [72], representing a subset of the task wrench space in which a robust grasp against external disturbances (up to a fixed upper bound) is guaranteed. Such actuation constraints, however, may depend on the joint configuration but they usually disregard the fact that the maximal normal force cannot be coupled with any tangential force component; the most common example of this strategies is to bound the friction cones with a predefined maximum normal force component, regardless the value of the tangential components.

In legged locomotion the seminal work of Takao et al. has studied the problem of finding an analogous polytope named Feasible Solution of Wrench (FSW) in multi-contact configurations [65]. The CWC margin then appeared as a stability criterion for legged locomotion suitable for *non-coplanar* contacts and finite friction coefficients [66]. Dai et al. in [69, 68] have shown how to exploit a CWC margin to obtain a convex optimization formulation that can concurrently plan CoM and joints trajectories of legged robots on complex terrains. This optimization, however, does not show computational performances compatible with online motion planning. On a similar line, Caron et al. [125] have focused on improving the real-time performances of 3D motion planning, either exploiting the double-description of the 6D wrench polyhedra or by considering lower dimensional projections of the CWC defining full-support areas. The latter, coupled with a linear pendulum model, led to the definition of the *pendular support area* [126].

Despite the excellent results shown in this field, the lack of successful experimental implementations on the hardware is mainly due to the fact that often the desired complex movements require torques to exceed the limits of the actuators. Moreover, the actuation capabilities become even more critical when the robot interacts with an environment of complex geometry. Indeed, in these environments, it is very likely that the robot is required to move to kinematically inconvenient configurations. Therefore, an accurate evaluation of the robot actuation capabilities takes on paramount importance.

4-2-1 Static Force polytopes

Actuator force/torque limits and their consequences on the overall performances in the task space have been analyzed for decades in the field on mechanical industrial manipulators [127, 128, 129] and, more recently on cable driven parallel robots (CPR) [130] and robotics hands [72].
Force/wrench ellipsoids (or hyperellipsoids) have been identified as useful tools to assess the control authority at the end-effector of serial mechanical chains. However, they represent a *qualitative* metric and they do not hold any information relative to the absolute magnitude of the wrench that a mechanical chain can exert. Their derivation can be obtained starting from the consideration of the theorem of kinetic energy (or work-energy theorem) that states that the work done by all forces acting on a particle equals the rate of change in the particle's kinetic energy. In case of null kinetic energy change, we have then:

$$\mathcal{P}_i + \mathcal{P}_e = 0 \tag{4-1}$$

where $\mathcal{P}_i$ and $\mathcal{P}_e$ are the internal and external power. The former represents the power generated by the actuators whilst the latter is the power exerted on the environment by the end-effector. In case of an industrial manipulator, they can be written as:

$$\mathcal{P}_i = \boldsymbol{\tau}^T \dot{\mathbf{q}} \quad \text{and} \quad \mathcal{P}_e = -\mathbf{w}^T \boldsymbol{\nu} \tag{4-2}$$

Eq. 4-1 represents a static relationship between the generalized task-space wrenches $\mathbf{w} \in \mathbb{R}^m$ and the generalized joint-space forces $\boldsymbol{\tau} \in \mathbb{R}^n$. Exploiting the mapping from joint to task-space velocities $\boldsymbol{\nu}$ (see, for example, Eq. 2-10) we can then write:

$$\underbrace{\boldsymbol{\tau}^T \dot{\mathbf{q}}}_{\mathcal{P}_i} \underbrace{- \mathbf{w}^T \boldsymbol{\nu}}_{\mathcal{P}_e} = \boldsymbol{\tau}^T \dot{\mathbf{q}} - \mathbf{w}^T \mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} = 0 \tag{4-3}$$

which leads to the following static relationship:

$$\boldsymbol{\tau} = \mathbf{J}(\mathbf{q})^T \mathbf{w} \tag{4-4}$$

where $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{m \times n}$ is the end-effector Jacobian. If we then consider Eq. 4-4 in combination with a unit hypersphere $\mathcal{S}_\tau$ in the joint torque space:

$$\mathcal{S}_\tau = \Big\{ \boldsymbol{\tau} \in \mathbb{R}^n \quad | \quad \boldsymbol{\tau}^T \boldsymbol{\tau} \leq 1 \Big\} \tag{4-5}$$

we can then obtain a new set (the force/wrench ellipsoid $\mathcal{E}_\mathbf{w}$) that describes how $\mathcal{S}_\tau$ is mapped into the task-space:

$$\mathcal{E}_\mathbf{w} = \Big\{ \mathbf{w} \in \mathbb{R}^m \quad | \quad \mathbf{w}^T \mathbf{J} \mathbf{J}^T \mathbf{w} \leq 1 \Big\} \tag{4-6}$$

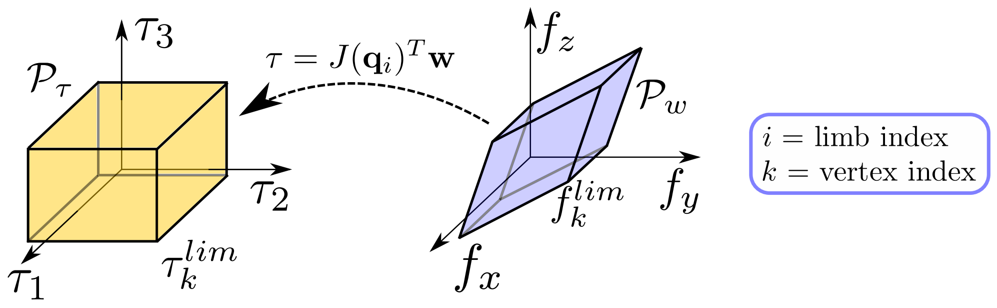


Figure 4-1: The mapping between joint-space torques and the task-space forces at the end-effector. In this example the dimension of the torques space $dim(\mathcal{P}_\tau) = n = 3$ is equal to the dimension of the manifold of the contact forces $dim(\mathcal{P}_\mathbf{w}) = m = 3$.

As of definition, the force ellipsoid $\mathcal{E}_\mathbf{w}$ represents the pre-image of the unit hypersphere $\mathcal{S}_\tau$ in the joint space under the mapping given by $\mathbf{J}(\mathbf{q})^T$. The lengths of the semidiameters of $\mathcal{E}_\mathbf{w}$ are the inverse of the square root of the singular values of the Jacobian $\mathbf{J}$ [15]. The ratio between the greatest and the smallest eigenvalue of $\mathbf{J}$ is, therefore, used as a measure of anisotropy of the ellipsoid and of the force amplification properties of the mechanical chain.

In a similar fashion, further exploiting Eq. 4-4, we can then also analyze the pre-image of the joint torques hypercube $\mathcal{P}_\tau$, *i.e.,* the set of all joint torques $\boldsymbol{\tau}$ comprised within the manipulator's actuator limits:

$$\mathcal{P}_\tau = \left\{ \boldsymbol{\tau} \in \mathbb{R}^n \quad | \quad -\boldsymbol{\tau}^{lim} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}^{lim} \right\} \tag{4-7}$$

The vector $\boldsymbol{\tau}^{lim} \in \mathbb{R}^n$ contains in its elements the hardware dependent lower and upper bounds of the values that limit the generalized joint torque vector $\boldsymbol{\tau}$. The hypercube $\mathcal{P}_\tau$ can therefore be seen also as system of $2n$ linear inequalities that constrain the joint torques [129]. The notation used in Eq. 4-7 assumes symmetric joint torques limits which is usually the case for the most common modern electric actuators; $\mathcal{P}_\tau$ is in this case a zonotope with its center in the origin (see Appendix B-1). However, in the case of actuators with asymmetric joint torque limits (*e.g.,* for velocity dependent torque limits, for hydraulic actuators with different chamber volumes or also for configuration-dependent torque limits as in the case of linear actuators with variable lever-arm) this notation can be updated to include such scenario without loss of the properties that are going to be described in the following sections. The hypercube $\mathcal{P}_\tau$ will still represent then a zonotope but its center will not correspond to the origin of the joint torque space.

The task-space force/wrench polytope $\mathcal{P}_\mathbf{w}$, pre-image of $\mathcal{P}_\tau$, can be written as follows (also see Fig. 4-1):

$$\mathcal{P}_\mathbf{w} = \left\{ \mathbf{w} \in \mathbb{R}^m \quad | \quad -\boldsymbol{\tau}^{lim} \leq \mathbf{J}^T\mathbf{w} \leq \boldsymbol{\tau}^{lim} \right\} \tag{4-8}$$

While the force ellipsoid $\mathcal{E}_\mathbf{w}$ can be used as a qualitative metrics of the robot's force amplification capabilities, the force polytope $\mathcal{P}_\mathbf{w}$ also includes quantitative informations about the maximum and minimum amplitude of the wrench that the robot can perform at the end-effector.

Force ellipsoids and force polytopes have been originally introduced for fixed-base non-redundant serial mechanical chains with $m = n$ where the Jacobian matrix $\mathbf{J}$ is thus invertible and the $\mathcal{V}$-representation of the force polytopes can be easily obtained by the simple inversion of $\mathbf{J}^T$:

$$\mathbf{w}^{lim} = \mathbf{J}^{-T}\boldsymbol{\tau}^{lim} \tag{4-9}$$

where $\mathbf{w}^{lim}$ is a vertex of the force polytope $\mathcal{P}_{\mathbf{w}}$. This is an especially convenient condition in which a one-to-one relation between joint-space torques and task-space wrenches exists. In the case of a three dimensional arm ($n = m = 3$), for example, a violation of one joint torque limit will correspond to a point on a facet of $\mathcal{P}_{\tau}$ and also to another point on a facet of the task-space polytope $\mathcal{P}_{\mathbf{w}}$. Similarly, a violation of two (or three) joint torque limits will correspond to a point on an edge (or a vertex) of the $\mathcal{P}_{\tau}$ and also to another point on an edge (or a vertex) of the task-space polytope $\mathcal{P}_{\mathbf{w}}$. Please see Appendix B-1 for the definitions of facets, edges and vertices for $n = m > 3$.

Successively the force ellipsoid and force polytopes concepts have been extended to redundant manipulators [128] in static conditions. At a later stage, also the concept of global force ellipsoid for parallel manipulators was presented [127], however, such extension has not been yet introduced for force polytopes.

Similar mathematical derivations and conclusions can be obtained for the mapping between joint- and task-space velocities which shares dual properties with respect to joint- and task-space forces. As a consequence, starting from the theorem of the kinetic energy, velocity (or *manipulability*) ellipsoids and polytopes for static equilibrium conditions can be defined.
An extension of the manipulability ellipsoid that considers the effect on gravity on fixed-base redundant manipulators in static conditions was presented in [131]. This extension was derived starting from the dynamic equation of motion for fixed base manipulators where zero joint space velocity $\dot{\mathbf{q}}$ and null external forces $\mathbf{w}$ were assumed:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \underbrace{\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}})}_{=0} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \underbrace{\mathbf{J}_s^T(\mathbf{q})\mathbf{w}}_{=0} \tag{4-10}$$

Under these conditions, the dynamic manipulability ellipsoid $\mathcal{E}_{\mathbf{v}}$ and polytope $\mathcal{P}_{\mathbf{v}}$ were defined as:

$$\mathcal{E}_{\mathbf{v}} = \left\{\mathbf{M}(\mathbf{q})^{-1}(\boldsymbol{\tau} - \mathbf{g}(\mathbf{q})) \quad | \quad \boldsymbol{\tau}^T\boldsymbol{\tau} \leq 1\right\} \tag{4-11}$$

$$\mathcal{P}_{\mathbf{v}} = \left\{\mathbf{M}(\mathbf{q})^{-1}(\boldsymbol{\tau} - \mathbf{g}(\mathbf{q})) \quad | \quad -\boldsymbol{\tau}^{lim} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}^{lim}\right\} \tag{4-12}$$

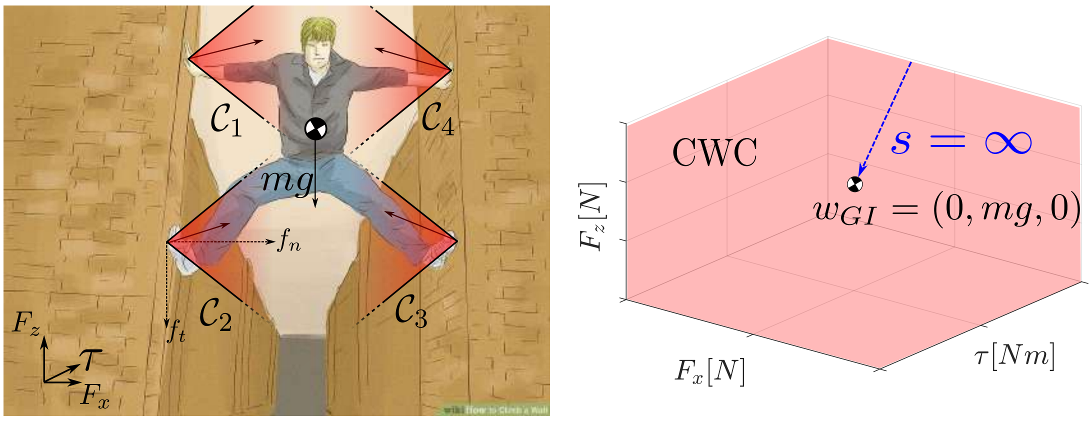
In this case the joint torques τ are employed to determine the feasible accelerations that can be performed at the manipulator's end-effector and they can then be exploited as a manipulability measure.

The further extension of manipulability ellipsoids to dynamic conditions has been studied by considering the complete Equation of Motions (EoM) without imposing any assumption on the joint-space velocity and acceleration [129, 132]
In the following Sections I will describe how such concepts close to the dynamic manipulability ellipsoids have led me to the definition of dynamic force/wrench polytopes for floating-base platforms. I will then explain how these tools can be useful for the analysis of the actuation capabilities of floating-base legged robots. Despite the definition valid for dynamic conditions, for the reasons explained in the continuation of this Chapter, I will use these force polytopes under static assumptions. This will lead me to the definition of the FWP, in Section 4-3-3, and to the formulation of the *local feasible region* in Chapter 5.

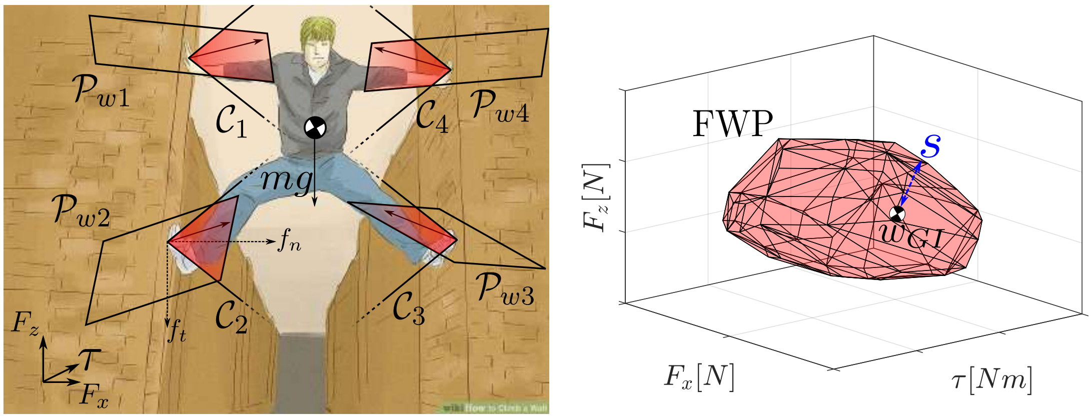
4-3 Wrench-Based Feasibility Analysis

Actuation limits significantly affect the wrench margin of a legged robot, *i.e.,* the amount of wrench that can be exerted on a robot without causing it to lose its contacts stability. As an example let us consider a human(oid) trying to climb a vertical chimney shown in 4-2. Here, the Contact Wrench Cone (CWC) is obtained through the Minkowski sum [133, 19] of the friction cones $\mathcal{C}_i$, represented by the pink areas (Fig. 4-2a *left*) where $i = 1, \dots, n_c$ and $n_c = 4$ is the number of contacts with the ground. In the CWC-based approach the margin is quantified as the minimal generalized distance between the gravito-inertial wrench $\mathbf{w}_{GI}$ and the boundary of the CWC. The CWC margin represents the maximum allowed wrench that can be applied (or rejected in case of a disturbance) in order to keep stable contacts (*i.e.,* no slippage).
In Fig. 4-2a (*right*) the CWC margin s has an infinite value $s = \infty$, *i.e.,* the *force closure* condition is achieved. This happens because the friction cone representation assumes that a contact force with an infinite normal component can be realized at the contact. This misleading result is the consequence of not taking the actuation limits into account.

On the other hand, imposing the actuation limits can be rephrased as adding a further constraint on the magnitude of the admissible contact forces with a *force polytope* $\mathcal{P}_{wi}$, with $i = 1, \dots, n_c$, that depends on the actuation capabilities and on the current configuration, *e.g.,* Fig. 4-2b(*left*). This limits the set of applicable body wrenches (feasible wrenches) that the human(oid) can apply on its own CoM to keep himself stable. In Fig. 4-2b(*right*) the bounded volume represents the result of the Minkowski sum of the four *bounded friction polytopes* (intersection of friction cones $\mathcal{C}_i$ and force polytopes $\mathcal{P}_{wi}$) after considering the torque limits that reflect themselves as maximal contact forces. This convex region is a subset of the CWC and we call it Feasible Wrench Polytope (FWP) in the remainder of this Chapter. This bounded polytope can also be computed as the intersection of the CWC and the Actuation Wrench Polytope (AWP), for more details see 4-3-3.



(a) If only the friction cones $\mathcal{C}_i$ are considered, the set of feasible wrenches (the CWC) corresponds to the whole wrench space. This yields an unlimited feasibility margin, regardless the mass of the human/robot.



(b) If both friction cone $\mathcal{C}_i$ and actuation constraints $\mathcal{P}_{wi}$ are considered, the overall set of feasible wrenches (the FWP) will be a limited set. This yields a limited feasibility margin and, therefore, to a more realistic result than the purely frictional case.

Figure 4-2: Planar example of chimney climbing: unbounded friction cones $\mathcal{C}_i$ give origin to an unbounded feasible wrench sets named CWC (*top*). Bounded friction polytopes $\mathcal{P}_{wi}$, instead, generate the bounded feasible wrench set FWP (*bottom*).

According to our definition, the margin s is limited to a finite value, as in Fig. 4-2b(*right*), showing that the human(oid) might fall if his limbs are not strong enough to support his body's weight and thus achieving a description closer to reality.

A motion planner based on these tools will devise trajectory that are consistent with the actuation limits of the robot, however it is necessary to enforce this consistency also in the underlying motion controller. We consider the development of such controller to be out of the scope of this dissertation and we assume this as given. Examples of suitable controllers, however, can be found in the literature, *e.g.*, [117, 21].

4-3-1 The Dynamic Force/Wrench Polytopes

In this section we illustrate the procedure to compute the *dynamic force/wrench polytopes*, *i.e.,* the set of feasible contact wrenches that a tree-structured robot can perform at its contact points with the environment while moving. For this, let us consider the EoM of a floating-base robot with n_f branches (*e.g.,* legs and/or arms), n_c of them in contact with the environment, each of them having a number n_l^k of actuated Degrees of Freedom (DoFs), $n = \sum_{k=1}^{n_f} n_l^k$ being the total number of actuated joints:

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{s}} + \mathbf{C}(\mathbf{q}, \boldsymbol{s}) + \mathbf{g}(\mathbf{q}) = \mathbf{B}\boldsymbol{\tau} + \mathbf{J}^T(\mathbf{q})\mathbf{f} \tag{4-13}$$

where $\mathbf{q} = \begin{bmatrix}\mathbf{q}_b^T & \mathbf{q}_j^T\end{bmatrix}^T \in SE(3) \times \mathbb{R}^n$ represents the pose of the floating-base system, composed of the pose of the base-frame $\mathbf{q}_b \in SE(3)$ and of the coordinates $\mathbf{q}_j \in \mathbb{R}^n$ describing the positions of the n actuated joints. The vector $\mathbf{s} = \begin{bmatrix}\boldsymbol{\nu}_b^T & \dot{\mathbf{q}}_j^T\end{bmatrix}^T \in \mathbb{R}^{6+n}$ is the generalized velocity, $\boldsymbol{\tau} \in \mathbb{R}^n$ is the vector of actuated joint torques while $\mathbf{C}(\mathbf{q})$ and $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^{6+n}$ are the centrifugal/Coriolis and gravity terms, respectively. $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{(n+6)\times(n+6)}$ is the joint-space inertia matrix, $\mathbf{B} \in \mathbb{R}^{(6+n)\times n}$ is the matrix that selects the actuated joints of the system. $\mathbf{f} \in \mathbb{R}^{m\cdot n_c}$ is the vector of contact forces[1] that are mapped into joint torques through the stack of Jacobians $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{3n_c\times(6+n)}$. If we split 4-13 into its underactuated and actuated parts, we get:

$$\underbrace{\begin{bmatrix}\mathbf{M}_b & \mathbf{M}_{bj}\\ \mathbf{M}_{bj}^T & \mathbf{M}_j\end{bmatrix}}_{\mathbf{M}(\mathbf{q})}\underbrace{\begin{bmatrix}\dot{\boldsymbol{\nu}}_\mathbf{b}\\ \ddot{\mathbf{q}}_j\end{bmatrix}}_{\dot{\mathbf{s}}} + \underbrace{\begin{bmatrix}\mathbf{c}_b\\ \mathbf{c}_j\end{bmatrix}}_{\mathbf{C}(\mathbf{q},\mathbf{s})} + \underbrace{\begin{bmatrix}\mathbf{g}_b\\ \mathbf{g}_j\end{bmatrix}}_{\mathbf{g}(\mathbf{q})} = \underbrace{\begin{bmatrix}\mathbf{0}_{6\times n}\\ \mathbf{I}_{n\times n}\end{bmatrix}}_{\mathbf{B}}\boldsymbol{\tau} + \underbrace{\begin{bmatrix}\mathbf{J}_b^T\\ \mathbf{J}_q^T\end{bmatrix}}_{\mathbf{J}(\mathbf{q})^T}\mathbf{f}. \tag{4-14}$$

By inspecting the actuated part (bottom line corresponding to n equations), that results from the concatenation of the equations of motions of all the branches, we see that $\mathbf{J}_q \in \mathbb{R}^{(m\cdot n_c)\times n}$ is block diagonal and we can use it to map joint torques into contact forces for each leg *individually.* We will see in 4-5-1 that this is convenient because it avoids using the coupling term $\mathbf{J}_b$. Eq 4.14 implicitly assumes that a set of holonomic constraints acts on system. These holonomic constraints are of the form $\phi(\mathbf{q}) = 0$, and may represent the fact that the contact points do not move with respect to an inertial frame under the assumption of rigid contacts. This can be represented as imposing the endeffectors to have null velocity and acceleration:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{J}_q\dot{\mathbf{q}}\\ \ddot{\mathbf{x}} &= \mathbf{J}_q\ddot{\mathbf{q}} + \dot{\mathbf{J}}_q\dot{\mathbf{q}}\end{aligned} \tag{4-15}$$

On a similar line to what defined in [127] as the dynamic manipulability polytope, I can now define a quantity that I call *dynamic force/wrench polytope* $\mathcal{A}_i$ for each individual $i - th$ branch of the tree structured robot:

$$\mathcal{A}_i = \left\{\mathbf{f}_i \in \mathbb{R}^m \quad | \quad \mathbf{M}_{bi}^T\dot{\boldsymbol{\nu}} + \mathbf{M}_i\ddot{\mathbf{q}}_i + \mathbf{c}(\mathbf{q}_i, \dot{\mathbf{q}}_i) + \mathbf{g}(\mathbf{q}_i) = \boldsymbol{\tau}_i + \mathbf{J}_i^T\mathbf{f}_i, \quad -\boldsymbol{\tau}_i^{lim} \le \boldsymbol{\tau}_i \le \boldsymbol{\tau}_i^{lim}\right\} \tag{4-16}$$

[1]Note that the HyQ robot has nearly point feet, henceforth we thus consider for this robot pure forces at the contact point and no contact torque ($m = 3$).

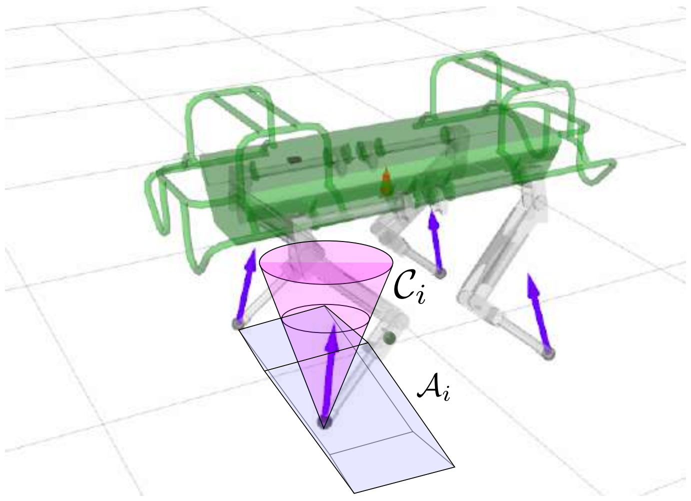


Figure 4-3: Representation of the friction cone $\mathcal{C}_j$ and force polytope $\mathcal{A}_j$ on one foot of the HyQ robot (j is the leg index). The purple arrows represent the contact forces and the green point is the CoP.

where $i = 1, \dots, n_c$ is the contact index and n_c is the number of active contacts between the robot and the environment. The vectors $\mathbf{q}_i \in \mathbb{R}^{n_l}$ and $\boldsymbol{\tau}_i \in \mathbb{R}^{n_l}$ represent the joint-space position and torque of only those joints that belong to the $i - th$ limb while n_l represents the number of actuated DoFs of that limb. If $m = 3$ then the contact wrench $\mathbf{f}_i \in \mathbb{R}^m$ consists of pure forces while if $m = 6$ then a non-zero contact torque is also present. For a partial list of the main symbols employed in this dissertation and their meaning please refer to the Notation section.
In Fig. 4-3 an example of dynamic wrench polytope is drawn for the HyQ robot: each limb of this robot has three actuated DoFs and thus $n_l = 3$. $\mathcal{A}_i$ is then a polytope of $2 \cdot 3 = 6$ facets and $2^3 = 8$ vertices.

Eq. 4-16 purposely omits the first line (six equations) of Eq. 4-14 referred to the unactuated floating base. This corresponds to neglecting the nonholonomic constraint relative to the non-integrability of the angular velocity of the floating base [134].
An alternative approach could consist in preserving this nonholonomic constraint, thus preserving the explicit relationship between actuated joint torques and the wrench acting on the un-actuated base link. This could lead to a definition of a unique unbounded cone defined in the N dimensional space of the underactuated system. The high dimensionality of the resulting set would, however, significantly harm the practical usefulness of this object.
As an opposite, the advantage of computing separate individual force polytopes $\mathcal{A}_i$ for each limb of the robot as proposed in this dissertation is that we can limit the dimensionality of the considered geometrical objects. Furthermore, we can also treat each force polytope $\mathcal{A}_i$ as a wrench capability measure of the corresponding limb. As a final consideration, we can

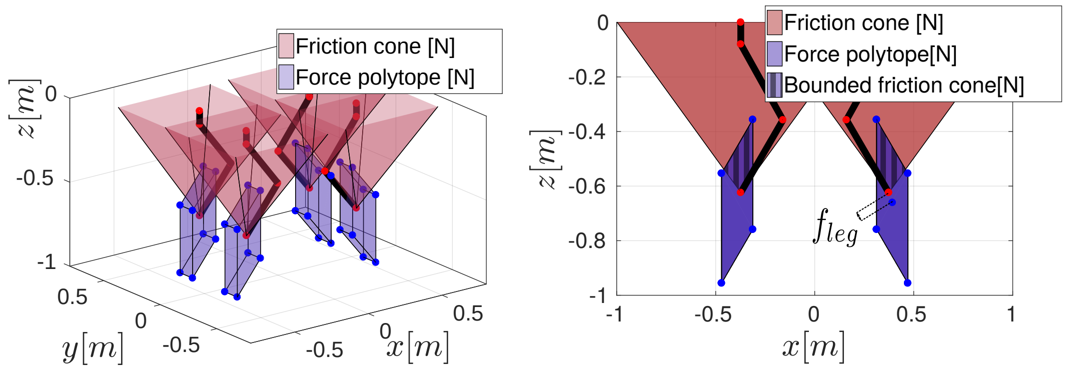


Figure 4-4: Representation of the force polytopes (blue) and of the friction cones (pink) of each single leg in 3D (*left*) and projected on the (F_x, F_z) plane (*right*). The offset f_{leg} is due to the bias term δ in 4-18.

observe that in static conditions ($\dot{\mathbf{q}} = \ddot{\mathbf{q}} = 0$) Eq. 4-16 can be written as:

$$\mathcal{A}_i = \left\{ \mathbf{f}_i \in \mathbb{R}^m \quad | \quad \mathbf{g}(\mathbf{q}_i) = \boldsymbol{\tau}_i + \mathbf{J}_i^T \mathbf{f}_i, \quad -\boldsymbol{\tau}_i^{lim} \leq \boldsymbol{\tau}_i \leq \boldsymbol{\tau}_i^{lim} \right\} \tag{4-17}$$

The term $\mathbf{g}(\mathbf{q}_i)$ represents the effect of gravity acting on the individual limb $i = 0, \dots, n_c$. From a geometrical point of view $\mathbf{g}(\mathbf{q}_i)$ can also be seen as an offset term that translates the polytope $\mathcal{A}_i$ in the same direction of the gravity vector, *i.e.,* towards the negative side of the f_z direction of the wrench space (see, for example, the offset f_{leg} in Fig. 4-4). For a predefined set of torque limits $\boldsymbol{\tau}_i^{lim}$ an increase in the legs mass and, as a consequence, a large offset term $\mathbf{g}(\mathbf{q}_i)$, will cause a decrease in the set of feasible positive contact forces. Under the static conditions as defined in Eq. 4-17, the definition of $\mathcal{A}_i$ corresponds to the static force polytope defined in Eq. 4-12 [131].

4-3-2 The Actuation Wrench Polytope (AWP)

In this Section we illustrate the procedure to compute the AWP that encodes the actuation capabilities of the robot. The next step is to compute the $\mathcal{V}$-representation of the force polytopes $\mathcal{A}_i$; note that this requires the inversion of the limb Jacobian matrix $\mathbf{J}_i^T$:

$$\begin{bmatrix} \mathbf{f}_{i,k}^{\text{lim}} \\ \mathbf{m}_{i,k}^{\text{lim}} \end{bmatrix} = \mathbf{J}_i^{T\#} (\underbrace{\mathbf{M}_{bi}^T \dot{\mathbf{v}} + \mathbf{M}_i \ddot{\mathbf{q}}_i + \mathbf{c}_i + \mathbf{g}_i}_{\delta} - \boldsymbol{\tau}_{i,k}^{\text{lim}}), \quad k = 1, \dots, 2^{n_l} \tag{4-18}$$

where $\boldsymbol{\tau}_{i,k}^{\text{lim}} \in \mathbb{R}^{n_l}$ is the $k-th$ vertex of the joint-space torque set $\mathcal{P}_{\boldsymbol{\tau}}$ (see Fig. 4-5).
$[\mathbf{f}_{i,k}^{\text{lim}\,T}, \mathbf{m}_{i,k}^{\text{lim}\,T}]^T \in \mathbb{R}^m$ is the $k-th$ vertex of the force/wrench polytope $\mathcal{A}_i$. In the HyQ quadruped robot, each leg has three DoFs ($n_l = 3$) and the feet can be approximated as point contacts ($m = 3$), therefore the angular component $\mathbf{m}_{i,k}^{\text{lim}}$ of the vertex is null.
$(.)^\#$ is the Moore-Penrose pseudoinverse, and $\mathbf{J}_i \in \mathbb{R}^{m \times n_l}$ is the Jacobian matrix of the $i-th$

contact point. In the quadruped case $\mathbf{J}_i$ is a square 3×3 matrix and a simple inversion is sufficient. Note that, for redundant and for underactuated limbs, the use of the pseudo-inverse may lead to solutions that actually violate the bounds given by the torque limits. In such cases, solving an inequality-constrained quadratic program is a viable solution [128].
Another option for the computation of the $\mathcal{V}$-representation of $\mathcal{A}_i$ is to perform a vertex-enumeration algorithm on its $\mathcal{H}$-representation; in the case of redundant limbs, differently from Eq. 4-18, this strategy leads to a set of vertices that does not depend on the arbitrary choice of the pseudo-inversion.

In the case of our quadruped robot HyQ, $\mathcal{A}_i$ is a polytope with $2^3 = 8$ vertices and its shape changes nonlinearly with the joint configuration because of the nonlinearities in $\mathbf{J}_i$[2]. As an example, we compute the force polytope for each leg in a quadruped robot. Fig. 4-4(*left*) shows the four force polytopes (together with the friction cones) obtained for a generic quadruped robot. Fig. 4-4(*right*) shows a lateral view that depicts the same force polytopes projected onto the (F_x, F_z) plane.

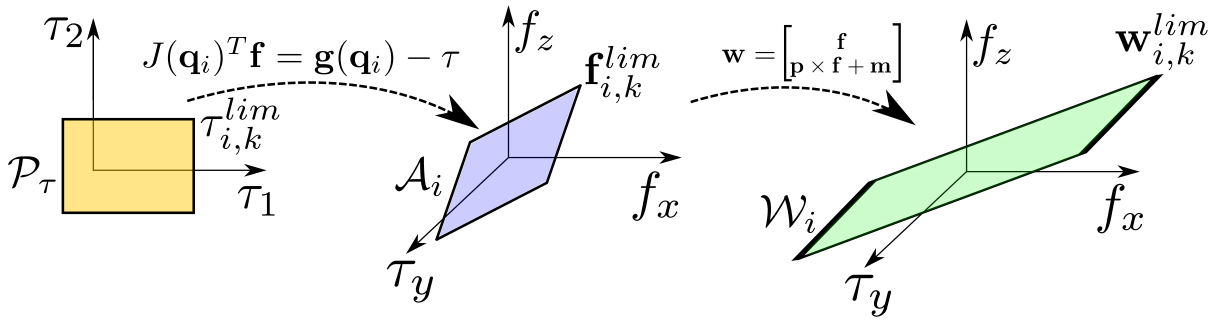


Figure 4-5: Pictorial representation of the subsequent mappings that project the set of joint torques into a set of contact wrenches. In this example the contact wrench is a pure force (f_x, f_z components only) and no angular component $\mathbf{m}^{lim} \in \mathbb{R}^3$ of contact wrench is considered. The force/wrench polytope $\mathcal{A}_i$ and its mapping $\mathcal{A}_i$ are therefore both *flat* polytopes in $\mathbb{R}^3$.

In order to compute the AWP, the next step is to add the torque values that are generated in correspondence to the maximum pure contact forces (as in Fig. 4-5):

$$\mathbf{w}_{i,k} = \begin{bmatrix} \mathbf{f}_{i,k}^{\text{lim}} \\ \mathbf{p}_i \times \mathbf{f}_{i,k}^{\text{lim}} + \mathbf{m}_{i,k}^{lim} \end{bmatrix} \quad \text{with } k = 1, \dots, 2^{n_l}, \quad i = 1, \dots, 2^{n_c} \tag{4-19}$$

where $\mathbf{p}_i \in \mathbb{R}^3$ represents the position of the $i-th$ contact foot and $\mathbf{w}_{i,k} \in \mathbb{R}^6$ represents the wrench that can be realized at that foot, both quantities are expressed in an inertial fixed frame. $\mathbf{f}_{i,k}^{lim}$ and $\mathbf{m}_{i,k}^{lim}$ represent the linear and angular part of the contact wrench obtained for the $i-th$ limb and the $k-th$ vertex of the joint-torques set $\mathcal{P}_w$. Since we focus here on point contacts, however, we have that $m = 3$ and $\mathbf{m}_{i,k}^{lim}$ is null. Therefore, the set of admissible

[2]The torque limits $\tau_{i,k}^{\text{lim}}$ may depend on the joint positions, making the dependency of the polytope from the joints configuration even more complex (*e.g.,* a revolute joint made of a linear actuator with nonlinear lever-arm).

wrenches that can be applied at the CoM by the $i-th$ foot/end-effector is:

$$\mathcal{W}_i = ConvHull(\mathbf{w}_{i,1}, \dots, \mathbf{w}_{i,2^{n_l}}) \tag{4-20}$$

with $i = 1, \dots, n_c$. We now have n_c *wrench polytopes* $\mathcal{W}_i$, one for each limb in contact with the environment. Finally, the AWP corresponds to the Minkowski sum of all the n_c wrench polytopes:

$$AWP = \oplus_{i=1}^{n_c} \mathcal{W}_i \tag{4-21}$$

As defined above, the AWP is a bounded convex polytope in $\mathbb{R}^6$ (see Fig. 4-7a) that contains all the admissible wrenches that can be applied to the robot's CoM that do not violate the actuation limits of the limbs in contact with the environment.
Note that the force polytopes $\mathcal{A}_i$ are zonotopes whose center is not in the origin, even in the case in which the set of feasible joint torques $\mathcal{P}_\tau$ was symmetric. This is because $\mathcal{A}_i$ results from the sum of a symmetric zonotopic term $-\mathbf{J}_i^{T\#}\tau^{\text{lim}}$ and of a nonzero singleton $\mathbf{J}_i^{T\#}\delta$ as in Eq. 4-18. This differs from what is generally done when static force polytopes are considered, *e.g.,* for cable-drive parallel robot and in robotic manipulation for the GWS, where inertial and Coriolis effects are usually ignored and the ropes/fingers are considered massless [135, 72].

4-3-3 The Feasible Wrench Polytope (FWP)

Note that the AWP does not include the constraints imposed by the environment, namely, the terrain normal, the friction cones coefficient and the unilateral contact condition (*e.g.,* the legs can not pull the ground). However, those constraints can be accounted by the CWC [66] (4-7 *center*):

$$CWC = ConvexCone(\hat{\mathbf{e}}_{i,k}), \quad k = 1, \dots n_e, \quad i = 1, \dots n_c \tag{4-22}$$

Here n_e is the number of edges of the linearized friction cone and:

$$\hat{\mathbf{e}}_{i,k} = \begin{bmatrix} \mathbf{e}_{i,k} \\ \mathbf{p}_i \times \mathbf{e}_{i,k} \end{bmatrix} \tag{4-23}$$

where $\mathbf{e}_{i,k} \in \mathbb{R}^3$ is the $k-th$ edge of the contact point i. We subsequently perform the intersection of the CWC with the AWP obtaining a convex polytope that we define as FWP (4-7 *right*):

$$FWP = CWC \cap AWP. \tag{4-24}$$

However, performing the *intersection* of polytopes in 6D is an expensive operation that requires the $\mathcal{D}$-description [119] of both operands. We then propose a more efficient approach for the computation of the FWP that:

1. first computes the intersection between the friction cones $\mathcal{C}_i$ and the force polytopes $\mathcal{A}_i$ obtaining, for each $i - th$ contact, a 3D bounded friction cone $\mathcal{B}_i$ (4-4 *right*) with μ vertices $\mathbf{b}_i^k \in \mathbb{R}^3$ (see Fig. 4-6):

$$\mathcal{B}_i = ConvHull(\mathbf{b}_i^k), \quad \text{with} \quad k = 1, \dots, \mu \tag{4-25}$$

2. then composes the wrench by adding the torque, as in 4-19:

$$\hat{\mathbf{b}}_i^k = \begin{bmatrix} \mathbf{b}_i^k \\ \mathbf{p}_i \times \mathbf{b}_i^k \end{bmatrix} \in \mathbb{R}^6 \quad \text{with} \;\; k = 1, \dots, \mu \tag{4-26}$$

   obtaining in this way the intermediate sets $\hat{\mathcal{B}}_i$:

$$\hat{\mathcal{B}}_i = ConvHull(\hat{\mathbf{b}}_i^k), \quad \text{with} \quad k = 1, \dots, \mu \tag{4-27}$$

3. finally, the FWP is computed through the Minkowski sum of the $\hat{\mathcal{B}}_i$ of the n_c contacts:

$$FWP = \oplus_{i=1}^{n_c} \hat{\mathcal{B}}_i \tag{4-28}$$

The advantage of this proposed method is that the intersection is performed in 3D rather than in 6D, which is computationally faster. This is advantageous also for the final step in 4-28 because it avoids computing vertices that will be removed later (*e.g.,* all the vertices from the AWP with negative contact forces are removed by intersecting with the CWC).

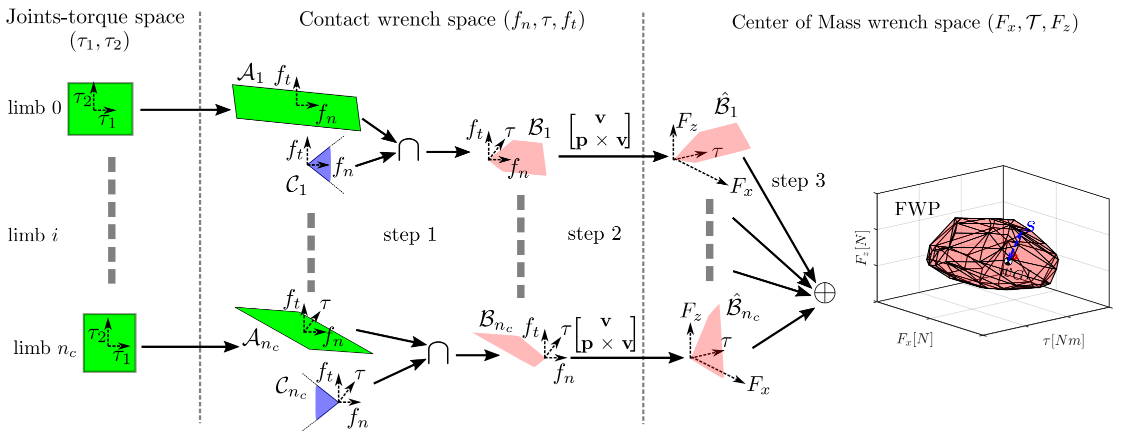

Joints-torque space (τ_1, τ_2)
Contact wrench space (f_n, τ, f_t)
Center of Mass wrench space $(F_x, \mathcal{T}, F_z)$
limb 0
limb i
limb n_c
τ_2
τ_1
$\mathcal{A}_1$ f_t f_n
$\mathcal{C}_1$
$\mathcal{A}_{n_c}$
$\mathcal{C}_{n_c}$
$\cap$
$\mathcal{B}_1$
$\mathcal{B}_{n_c}$
$\begin{bmatrix} \mathbf{v} \\ \mathbf{p} \times \mathbf{v} \end{bmatrix}$
step 1
step 2
step 3
$\hat{\mathcal{B}}_1$
$\hat{\mathcal{B}}_{n_c}$
F_z τ F_x
$\oplus$
FWP
$F_z[N]$
$F_x[N]$
$\tau[Nm]$

Figure 4-6: Planar 2D example of the steps needed to compute the FWP. The steps 1, 2 and 3 refer to the bullet list above of Eq. 4-25, 4-27, 4-26 and 4-28.

Notice that the algorithms used to compute the Minkowski sum of convex cones are significantly different from those meant for bounded polytopes. In particular, as mentioned in Appendix B-2, the Minkowski sum of polyhedral cones corresponds to performing their convex hull (*i.e.,* using the $ConvexCone(\cdot)$ operator, as in Eq. 4-22). The same operation performed on n-dimensional bounded polytopes however, requires specific algorithms based on the $\mathcal{V}$-description [136] and is less computationally efficient.

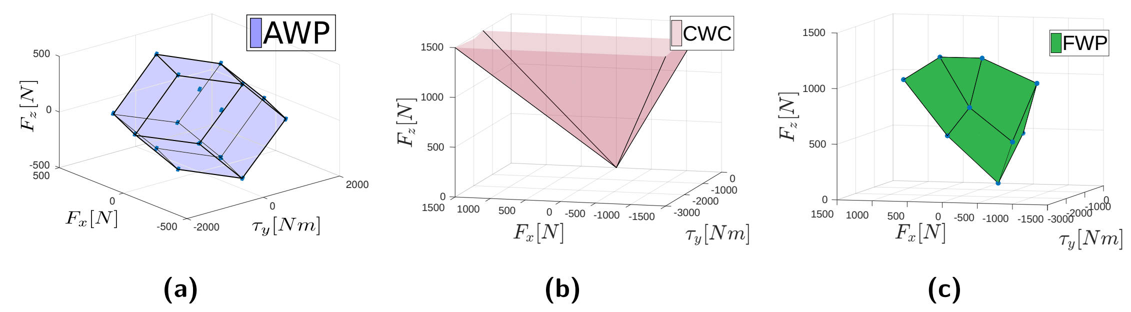


Figure 4-7: Pictorial 3D representation of the six-dimensional feasibility polytopes:

(a) The Actuation Wrench Polytope (AWP);

(b) The Contact Wrench Cone (CWC);

(c) The Feasible Wrench Polytope (FWP).

Besides that, the force polytopes $\mathcal{A}_i$ (and also the intersected polytopes $\mathcal{B}_i$) are 6D *full dimensional* polytopes only when the contact wrench includes a torque component. In the case of a quadruped with point contacts, instead, only pure contact forces are involved and thus the force polytopes $\mathcal{A}_i$ are *flat* 6D polytopes (*i.e.,* they have empty interior in $\mathbb{R}^6$) even if their Minkowski sum is a *full* dimensional polytope in $\mathbb{R}^6$.
Recent advancements in computational geometry enable the Minkowski sum for bounded polytopes in n dimensions to be obtained using only the $\mathcal{V}$-description (without being affected by the flatness of the operands) [119], thus resulting in a considerable speed up that allows Eq. 4-28 to be computed online: for three contact points,*i.e.,* $n_c = 3$, and each polytope $\mathcal{G}_i$ composed of about 100 vertices in 6D, Eq. 4-28 can be solved in about $100ms$ using Politopix [118]).

4-3-4 Polytope Representation for a Planar Model

To achieve a better understanding of the nature of these polytopes, let us consider the simplified case of a planar dynamic model, as in Fig. 4-4 (*right*), where each point of the space is represented through the (x, z) coordinates. In this case the wrench space has three coordinates (F_x, F_z, τ_y) and can be represented in 3D. Fig. 4-7 depicts instead the AWP (*left*), the CWC (*center*) and the FWP (*right*) for the same simplified model. From the drawing it is possible to see that the CWC is a convex cone while the AWP is a bounded polytope; their intersection, the FWP is therefore also a bounded polytope.

4-4 FWP-Based Feasibility Metric

The CWC margin has been proven to be a universal criterion for dynamic legged stability [66]. However, the CWC still lacks knowledge of robot's feasibility constraints such as self-collisions, actuation and kinematic joint limits. Therefore, in order to plan complex motions in unstructured environments we will introduce a more restrictive metric, that we generically call the *feasibility metric*. This metric incorporates all the properties of the CWC criterion, and additionally the robot's actuation limits. We leave to future works the development of another formulation also able to encode kinematic limits and self-collisions.
In order to obtain the *feasibility metric* we first need to compute the robot's gravito-inertial wrench $\mathbf{w}_{GI} \in \mathbb{R}^6$ at the specific robot state that we want to evaluate (also see Eq. 2-30):

$$\mathbf{w}_{GI} = \dot{\mathbf{h}} - \mathbf{w}_G \tag{4-29}$$

with:

$$\dot{\mathbf{h}} = \begin{bmatrix} m\ddot{\mathbf{c}} \\ \dot{\mathbf{k}}_{\mathcal{W}} \end{bmatrix}, \quad \mathbf{w}_G = \begin{bmatrix} m\mathbf{g} \\ \mathbf{c} \times m\mathbf{g} \end{bmatrix} \tag{4-30}$$

where $\mathbf{k}_{\mathcal{W}} \in \mathbb{R}^3$ is the robot's angular momentum and $\mathbf{c}$ is its CoM position (both expressed in the fixed coordinate frame $\mathcal{W}$). The criterion of feasibility can then be defined as:

$$\mathbf{w}_{GI} \in FWP. \tag{4-31}$$

The definition of feasibility metric depends on the type of the representation chosen for the FWP, *i.e.,* half-plane description ($\mathcal{H}$-description) or a vertex description ($\mathcal{V}$-description). Note that these two representations are dual and, according to the Minkowski-Weyl theorem, any polytope can be equivalently described with either representation. In the following we present a formulation of FWP margin for either representation. In the continuation of this Chapter, however, only the vertex-based formulation FWP margin will be employed thanks to the convenient properties of $\mathcal{V}$-description explained in Section 4-3-3.

4-4-1 FWP Half-Plane Description

In the $\mathcal{H}$-representation, the FWP set can be written in terms of half-spaces as:

$$FWP = \{\mathbf{w} \in \mathbb{R}^6 \quad | \quad \hat{\mathbf{a}}_j^T \mathbf{w} \leq b_j, \quad j = 1, \dots n_h\} \tag{4-32}$$

where n_h is the number of half-spaces of the FWP and $\hat{\mathbf{a}}_j \in \mathbb{R}^6$ is the normal vector to the $j-th$ facet pointing outwards with respect to the polytope. The feasibility criterion expressed in 4-31 can thus be written as:

$$\mathbf{H}^T \mathbf{w}_{GI} \leq \mathbf{b} \tag{4-33}$$

where $\mathbf{H} \in \mathbb{R}^{6\times n_h}$ is the matrix whose columns are the normals to all the half-spaces that bound the FWP and $\leq$ is a component-wise operator. The columns of $\mathbf{H}$ can be divided into two blocks $\mathbf{H}_c$ and $\mathbf{H}_a$ in order to distinguish the CWC half-spaces from the AWP half-spaces, respectively: $\mathbf{H} = [\mathbf{H}_c|\mathbf{H}_a]$. If $\mathbf{H}_c^T\mathbf{w}_{GI} > \mathbf{0}$ but $\mathbf{H}_a^T\mathbf{w}_{GI} \leq \mathbf{0}$ then the robot's state is consistent with its actuation capabilities but its contact condition is unstable (*e.g.,* friction limits are violated). Viceversa, if $\mathbf{H}_c^T\mathbf{w}_{GI} \leq \mathbf{0}$ but $\mathbf{H}_a^T\mathbf{w}_{GI} > \mathbf{0}$ then the system has stable contacts but it does not respect the actuation limits. In the latter case, the legged system might still not fall but it will not be able to realize the desired task.

If the $\mathcal{H}$-description of the FWP is given, we can provide a definition of robustness that extends the properties of the CWC margin. In the same line with [68] the feasibility metric can be defined as the margin m, *i.e.,* the generalized distance of the point w_{GI} from the boundaries of the FWP [75]. This corresponds to finding the biggest disturbance wrench $\mathbf{w}_d \in \mathbb{R}^6$ that the system can reject. This is equivalent to computing the largest residual radius m such that the m-ball $\mathcal{B}_m$ (centered in $\mathbf{w}_{GI}$) still lies within the FWP:

$$\mathcal{B}_m \in FWP \tag{4-34}$$

where $\mathcal{B}_m$ is defined as:

$$\mathcal{B}_m = \{\mathbf{w}_{GI} + \mathbf{T}(\mathbf{p}_{w_d})\mathbf{w}_d \quad | \quad \mathbf{w}_d^T\mathbf{Q}\mathbf{w}_d \leq m\} \tag{4-35}$$

where m is the feasibility margin. $\mathbf{p}_{w_d}$ is the disturbance application point and $\mathbf{T}(\mathbf{p}_{w_d}) \in \mathbb{R}^{6\times 6}$ is the adjoint spatial transform that expresses it in the world frame $\mathcal{W}$ [68]. $\mathbf{Q}$ is a positive definite matrix that is used to deal with the non homogeneous angular and linear units of the wrench.

4-4-2 FWP Vertex Description

If only a $\mathcal{V}$-description of the FWP is available, the distance between w_{GI} and the faces of the FWP cannot be computed anymore and a different definition of feasibility metrics is needed. We decide to employ in this case a *feasibility scalar factor* $s \in (-\infty, 1]$ adapted by the scaling factor defined in [72] used to measure the quality of a robotic grasp quality.
Let us consider a matrix $\mathbf{V} \in \mathbb{R}^{6\times n_v}$ whose columns are the vertices $\mathbf{v}_i$ of the FWP, and a vector $\boldsymbol{\lambda} \in \mathbb{R}_+^{n_v}$ of non-negative weights where n_v is the number of vertices of the FWP. Every point inside the FWP can be described with a combination of weights λ_i such that $\sum_{i=0}^{n_v} \lambda_i = 1$. We therefore define the robot to be in a feasible state if, for the corresponding wrench $\mathbf{w}_{GI}$, there exists a $\boldsymbol{\lambda}$ such that $\mathbf{V}\boldsymbol{\lambda} = \mathbf{w}_{GI}$, with $\sum_{i=1}^{n_v} \lambda_i = 1$ and $\lambda_i \geq 0 \quad i = 1, \dots, n_v$.
A preliminary step for the computation of the feasibility factor consists in subtracting the centroid $\mathbf{v}_c$ from all the FWP vertices $\mathbf{v}_i$, obtaining new translated vertices $\hat{\mathbf{v}}_i$ ($\hat{\mathbf{v}}_i = \mathbf{v}_i - \mathbf{v}_c$). This has the effect of shifting the origin of the wrench space in the centroid, that, in a

$\mathcal{V}$-representation, is a good approximation of the most "robust" point (*i.e.,* the Chebishev centre). We then define a new *scaled* or *shrunk* polytope $\mathcal{P}^s$ centered in the origin (which is now also the centroid of the FWP). $\mathcal{P}^s$ can be expressed in terms of its own vertices $\hat{\mathbf{v}}_i^s$ and of a set of multipliers $\boldsymbol{\lambda}_i^s$:

$$\mathcal{P}^s = \left\{ \mathbf{w} \in \mathbb{R}^6 | \mathbf{w} = \sum_{i=1}^{n_v} \lambda_i^s \hat{\mathbf{v}}_i^s, \lambda_i^s \geq 0, \|\boldsymbol{\lambda}^s\|_1 = 1 \right\} \tag{4-36}$$

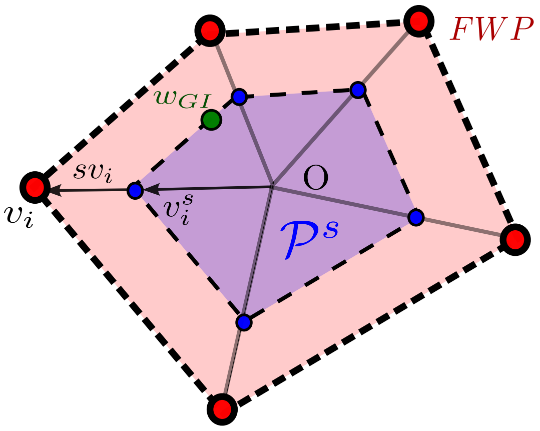


Figure 4-8: A 2D pictorial representation of the shrunk polytope $\mathcal{P}^s$ (*left*): we see that the factor s can be seen as the scaling factor of the shrunk polytope with respect to the FWP.

For a better understanding 4-9 (*left*) illustrates the idea of the shrunk polytope for a 2D representation.

The FWP's vertices $\hat{\mathbf{v}}_i$ are linked to the vertices of $\mathcal{P}^s$ through the feasibility factor s:

$$\hat{\mathbf{v}}_i^s = \hat{\mathbf{v}}_i(1 - s), \quad -\infty < s \leq 1 \tag{4-37}$$

If, for instance, $s = 1$ then $\mathcal{P}^s$ shrinks into the origin. If we impose $\lambda_i = (1 - s)\lambda_i^s$, then we can write the shrunk polytope $\mathcal{P}^s$ in terms of the vertices $\hat{\mathbf{v}}_i$ of the FWP, i.e.:

$$\mathcal{P}^s = \left\{ \mathbf{w} \in \mathbb{R}^6 | \mathbf{w} = \sum_{i=1}^{n_v} \lambda_i \hat{\mathbf{v}}_i, \lambda_i \geq 0, \|\boldsymbol{\lambda}\|_1 = 1 - s \right\} \tag{4-38}$$

We can see the feasibility factor as the scalar s that corresponds to the smallest scaled polytope containing the point $\mathbf{w}_{GI}$. The problem of finding s can be formulated as a (LP) that can be carried out by any general-purpose solver:

$$\begin{aligned} \max_{\boldsymbol{\lambda}, s} \quad & s \\ \text{such that:} \quad & \mathbf{V}\boldsymbol{\lambda} = \mathbf{w}_{GI} \\ & \|\boldsymbol{\lambda}\|_1 = 1 - s \quad s \in (-\infty, 1] \\ & \lambda_i \geq 0 \quad i = 1, \dots, n_v \end{aligned} \tag{4-39}$$

Note that the larger is s, the more robust is the system against disturbances. A negative s means that the point is out of the polytope and the wrench is unfeasible. When s becomes zero, it means the point is on the polytope boundary and that either the friction or actuation limits are violated. Table 4-1 shows the computation time of a Intel(R) Core(TM) i5-4440 CPU @ 3.10GHz with 4 cores for three and four contacts scenarios. The feasibility factor s, unlike the margin m, is not sensitive to the fact of having different units in the wrench space since it does not encode the concept of distance. On the other hand, for the very same reason, the factor s of a given stance configuration cannot be compared to another configuration, because of the different scaling of the two polytopes. Consider for example the FWP computed for a humanoid robot in single support $\mathbf{V}_1$ and in double support $\mathbf{V}_2$. The size of $\mathbf{V}_1$ will be considerably larger than the size of $\mathbf{V}_2$ because of the increased stability enhanced by the second stance leg. In both cases however, if the gravito-inertial wrench is located exactly in the centroid (or in the Chebychev center) of the polytopes $\mathbf{V}_1$ and $\mathbf{V}_2$, the feasibility factors s_1 and s_2 will take on the same values 1.0. Just by looking at the values of s_1 and s_2 we are not able to tell which one corresponds to the most robust condition.

4-5 Online CoM Trajectory Optimization

The FWP factor can be used to plan robust CoM trajectories. Hereafter, we present a brief description of how we used the proposed criterion to plan online motions for our quadruped robot, that do not violate the actuation limits.
We further extended the capabilities of the locomotion framework [28, 38] by replacing the original (*heuristic*) planner with a trajectory optimizer that exploits the proposed feasibility criterion. This framework realizes a statically stable *crawling* gait where a *base motion* phase and a *swing motion* phase are alternated (therefore the base of the robot does not move when a leg is in swing).
To be able to compare with our baseline planner, we opted for a decoupled planning approach where the footholds and the CoM trajectory of the robot are determined sequentially: first the future foothold is computed then a trajectory is computed through optimization to drive the CoM from its actual position to the new support polygon.
In essence, our online Trajectory Optimization (TO) computes during every swing phase the CoM trajectory to be realized in the next base motion phase using a *one-step horizon*. The decision variables of the optimization problem are the X, Y components of the CoM positions, the velocities and the duration of the *base motion* phase Δt_{bm}:

$$\mathbf{\Gamma} = \{\mathbf{c}_x[k], \mathbf{c}_y[k], \dot{\mathbf{c}}_x[k], \dot{\mathbf{c}}_y[k], \Delta t_{bm}\}, \quad \text{with} \quad k = 1, \cdots, N \tag{4-40}$$

The trajectory is discretized in N equally spaced knots (at time intervals $h = \Delta t_{bm}/N$). Note, that we here do not optimize the trunk orientation nor the coordinate z. This is because, for

Table 4-1: Computational features of the feasibility factor s

	3 non-coplanar contacts	4 non-coplanar contacts
FWP vertices	436	1118
variables	437	1119
constraints	7	7
LP time [ms]	90	350

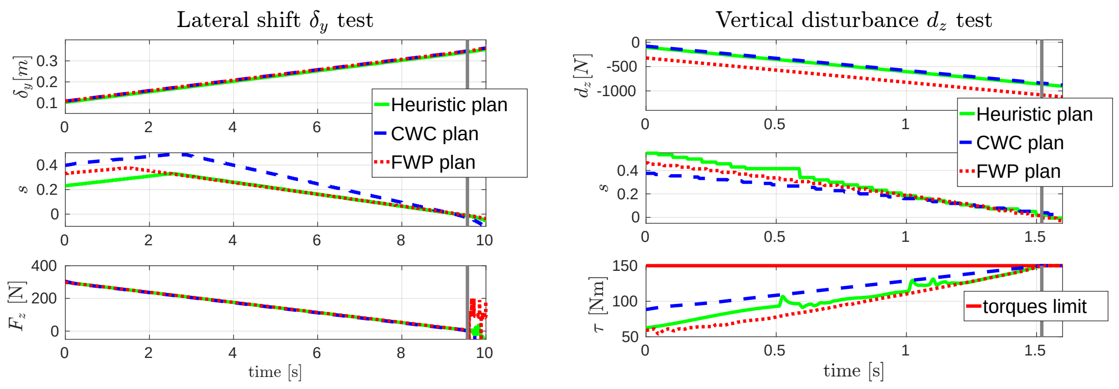


Figure 4-9: Simulation data of the horizontal displacement test (*left*) and vertical disturbance test (*right*). The lateral displacement δ_y gradually unloads the RF leg while the gradual loading d_z leads the torques to hit their limits. Both these cases are captured by the feasibility factor that goes to zero whenever either the friction or the actuation constraints are reached.

quasi static motions, the predominant acceleration term acting on the system is gravity itself, and, therefore, the influence of the z coordinate of the CoM on the stability or on the joint torques is limited compared to the role of the x and x components.
The aim of the optimizer is to maximize the FWP factor s, as in Eq. 4-39, while we enforce back-ward Euler integration constraints along the trajectory and zero velocity at the trajectory extremes.

As a first step to minimize the margin, we evaluate the FWP polytope considering the robot contact configuration during the next triple support phase. Indeed, even if the CoM moves during the 4 stance phase, we need to ensure that the robot will be stable also in the subsequent (triple stance) swing phase. Moreover, triple stance is more demanding than the 4 stance phase in terms of torques because the weight of the robot is distributed on a lower number of legs. Enforcing a triple stance feasibility since the beginning of the trajectory (where most likely the com position can be infeasible) is not an issue for the optimization because the robustness is implemented as a soft constraint in the cost function. On the other hand, this has the effect to naturally drive the CoM toward the future (triple stance) support polygon.

In case of a quasi-static condition ($\ddot{\mathbf{q}} = \dot{\mathbf{q}} = \mathbf{0}$), Eq. 4-18 reduces to:

$$\mathbf{f}_i^{\text{lim}} = \mathbf{J}_i(\mathbf{q}_{0_i})^{-T}\Big(\mathbf{g}(\mathbf{q}_{0_i}) - \mathbf{B}\boldsymbol{\tau}^{\text{lim}}(\mathbf{q}_{0_i})\Big) \tag{4-41}$$

This enables the computation of the FWP just once at the beginning of the optimization. As a matter of fact, thanks to this condition and to the approximation later explained in Section 4-5-1, the FWP becomes only dependent on the contact configuration. Then, to compute the running cost $\sum_{k=1}^{N} \mathcal{L}$, we evaluate the CoM acceleration along the trajectory ($h \cdot \ddot{\mathbf{c}}_{x,y}[k+1] = \dot{\mathbf{c}}_{x,y}[k+1] - \dot{\mathbf{c}}_{x,y}[k], \forall k$) and evaluate the gravito-inertial wrench at each knot

Table 4-2: Trajectory optimization variables of the FWP $\mathcal{V}$-description

	3 non-coplanar contacts	4 non-coplanar contacts
timesteps	10	10
FWP vertices	436	1118
variables $\mathbf{\Gamma}$	41	41
constraints	24	24
time [ms]	75	85

through Eq. 2-30.

To compute the feasibility factor exploiting the $\mathcal{V}$-description we should add the $\boldsymbol{\lambda}$ vector as decision variable and the constraints in 4-39. However, the amount of decision variables would significantly increase due to the high number of vertices in the polytope (*i.e.,* $\boldsymbol{\lambda}$ may have hundreds of elements for each optimization knot) leading to computation times that do not meet the requirements for online planning. Thus we tackled this problem by computing the set of $\boldsymbol{\lambda}$ through a simple Moore-Penrose pseudo-inversion: $\boldsymbol{\lambda}[k] = \mathbf{V}^{\#}\mathbf{w}_{GI}[k]$. In this way the decision variables will be only the states $\mathbf{\Gamma}$ of the system and the number of vertices will influence the size of the TO problem only marginally (see Table 4-2). We noticed that adding a *bias* term in the nullspace of $\mathbf{V}$ to "drive" the solution $\boldsymbol{\lambda}[k]$ toward the FWP centroid $\boldsymbol{\lambda}_0 \in \mathbb{R}^{n_v}$ was giving satisfactory results:

$$\boldsymbol{\lambda}[k] = \mathbf{V}^{\#}\mathbf{w}_{GI}[k] + \mathbf{N}_V\boldsymbol{\lambda}_0, \quad \boldsymbol{\lambda} \in \mathbb{R}^{n_v} \tag{4-42}$$

where $\mathbf{N}_V \in \mathbb{R}^{n_v \times n_v}$ is the null-space projector associated to $\mathbf{V}$. Indeed, if we set $\boldsymbol{\lambda}_0 = [1/n_v, \cdots, 1/n_v]$ as the geometric center of the FWP, the constraints $\|\boldsymbol{\lambda}_0\|_1 = 1, \lambda_{0i} > 0$ are satisfied by construction. Thanks to the one-to-one correspondence between the gravito-inertial wrench and the weights $\boldsymbol{\lambda}$, penalizing the deviation of $\boldsymbol{\lambda} \in \mathbb{R}^{n_v}$ from $\boldsymbol{\lambda}_0$ is equivalent to maximizing the feasibility factor. Therefore, we formulate the running cost computation as:

$$\mathcal{L}(\mathbf{c}[k], \dot{\mathbf{h}}[k], \mathbf{V}) = \|\boldsymbol{\lambda}[k] - \boldsymbol{\lambda}_0\|_2 \tag{4-43}$$

We can therefore finally formulate the trajectory optimization problem as:

$$\begin{aligned} \min_{\mathbf{\Gamma}} \quad & \sum_{k=1}^{N} \mathcal{L}(\mathbf{c}[k], \dot{\mathbf{h}}[k], \mathbf{V}) \\ \text{subject to:} \quad & \dot{\mathbf{c}}_x[k+1] = (\mathbf{c}_x[k+1] - \mathbf{c}_x[k])/h \\ & \dot{\mathbf{c}}_y[k+1] = (\mathbf{c}_y[k+1] - \mathbf{c}_y[k])/h \\ & \dot{\mathbf{c}}_x[0] = \dot{\mathbf{c}}_y[0] = \dot{\mathbf{c}}_x[N] = \dot{\mathbf{c}}_y[N] = \mathbf{0} \end{aligned} \tag{4-44}$$

4-5-1 Computational Issues and Approximations

The TO strategy proposed in this Chapter consists of two main steps:

a) evaluation of the FWP $\mathcal{V}$-description;

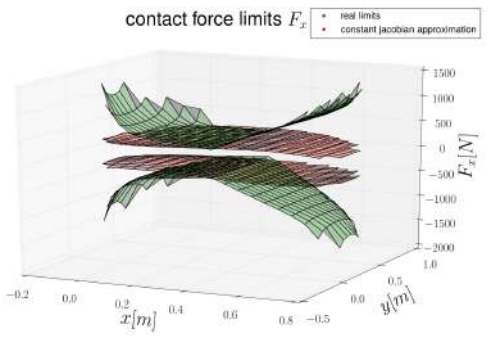
b) solution of the optimization problem in Eq. 4-44

Despite the remarkable computational speed-up obtained by the usage of the $\mathcal{V}$-description (see Table 4-3) and of the Politopix library [118], the evaluation of the FWP, point (a), still represents the most time-consuming step of this pipeline (about $150ms$ needed for a triple-stance configuration). The subsequent step (b), *i.e.,* the solution of the TO problem for a given FWP, requires instead about $75 - 85ms$ for 10 nodes trajectory.

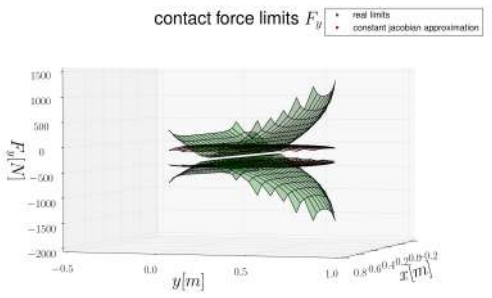
Table 4-3: FWP's $\mathcal{V}$- and $\mathcal{H}$-description computation time with Politopix.

	2 contact points	3 contact points	4 contact points
$\mathcal{V}$-description	$0.03s$	$0.15s$	$0.49s$
$\mathcal{H}$-description	$0.04s$	$1.03s$	$30.21s$

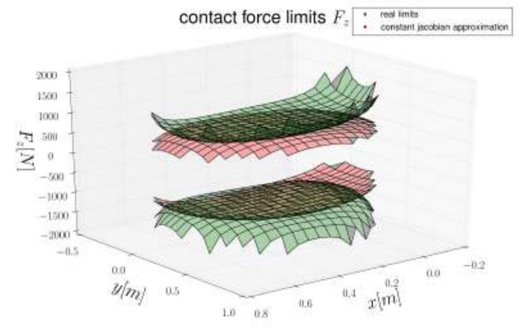
Ideally we should recompute step a and step b iteratively, concurrently optimizing the vertices of the FWP and the trajectory inside it. We decided instead to compute the FWP only once per step, assuming that its vertices do not change during the execution of each phase, as explained in the following paragraph.
We first analyze the influence on the estimated maximum and minimum contact force when the Jacobian matrix is approximated to be constant along a *body motion* of the HyQ robot. In Fig. 4-10 we can see the contact force boundaries when a foot spans its workspace (the foot covers all the x and y positions on a plane located at $z = -0.6m$). The green surfaces represent the real boundaries of the vertical contact force considering the correct leg Jacobian and for each considered foot position. The red surfaces show instead the same force boundaries when a constant Jacobian is evaluated. We can see that in a neighborhood of the default foot configuration($[0.3, 0.2, -0.6]m$ with respect to the base frame of the robot) the approximation is accurate and becomes rough in proximity of the workspace boundaries. We chose to use a constant Jacobian matrix corresponding to an average joint configuration $\mathbf{q}_d$ computed from the heuristic planner. In this way the Jacobian remains constant and we remove the AWP's dependency from the joints position.
As a further simplification we assume a quasi-static motion as in Eq. 4-41. These assumptions allow us to compute the FWP only once at each stance change. Note that all the wrenches are expressed in the fixed frame.



(a) f_x contact force limits



(b) f_y contact force limits



(c) f_z contact force limits

Figure 4-10: Contact force limits on the left-front (LF) leg of HyQ as a function of the foot position computed with real torque limits (*green*) and with Jacobian approximation (*red*).

4-6 Experimental Results

In this section we present simulation results and real experiments with the HyQ robot. The first two simulations validate the feasibility factor formulation based on the $\mathcal{V}$-description of the FWP. After that we highlight the differences between our proposed feasibility metric and a state-of-the-art stability metric. Finally we present a few examples of the behaviors we can obtain with the TO presented in Section 4-5.

4-6-1 Validation of the FWP Feasibility Factor

In a first test, we consider three different motion planners: the baseline heuristic planner, a CWC planner (that incorporates only frictional constraints) and our FWP planner. We have the robot crawling where the CoM trajectory is planned by the three different methods. In all cases, we stop the robot during a triple-stance phase (being more critical for robustness than four-stance phases). The final position of the CoM will be different for each of the planners. We then make the robot displace its CoM laterally with an increasing offset $\delta_y = \epsilon 0.5\ m$ ($\epsilon \in \mathbb{R}$ increases linearly from 0 to 1). The objective is to obtain a gradual unloading of the

lateral legs and therefore violate the unilaterality constraints. Figure 4-9 (*center*) shows the evolution of the displacement δ_y and of the normal component of the contact force F_z at the right-front (RF) leg. As expected the plot shows that for all the cases s drops to zero when the leg RLF becomes unloaded ($F_z = 0$).
In the second test the robot is again stopped during a walk in a three-legs stance configuration. This time a vertical disturbance force was applied at the origin of the base link origin (*i.e.,* the geometric center of the torso). The force is *vertical* and pointing downwards with increasing magnitude $d_z = -\epsilon 1000\ N$ where $\epsilon \in \mathbb{R}$ is linearly increasing from 0 to 1. The joint torques will increase because of the action of this force, eventually making one (or more) of them hit the limits. Since the test is performed in a static configuration and the disturbance force is always vertical, the CoP of the system will not change, being the robot always statically stable. Fig. 4-9 (*right*) shows a plot of the magnitude of the vertical pushing force d_z together with the knee joint torque of the LF leg and the feasibility factors s in the three cases. We can see that, in the case of the static configuration found with the FWP planner, the torque limit is reached for a higher amplitude of the disturbing force (about $-1100N$ compared to $-900N$), showing that this is more robust against external disturbance forces than the configurations selected by the heuristic and by the CWC planners. We can see that in all the cases the feasibility factor s goes to zero when a torque limit is violated.

4-6-2 CWC Margin vs. FWP Feasibility Factor Comparison

The last test highlights the main differences between the feasibility factor s and the traditional stability measures. As state-of-the-art stability metric we consider the CWC-margin, which is obtained by applying Eq. 4-39 on the $\mathcal{V}$-description of the CWC, rather than the FWP as explained in Section 4-4-2. Fig. 4-11 (*above*) shows the results when a crawl gait is evaluated using this method. The red line represents the value of the CWC-margin during the triple-stance phase of a crawling gait. The dashed blue line represents the same walk, evaluated again with the CWC-margin, in the case that an external load of $20kg$ is applied on the CoM of the robot during its walk. The factor referring to the four-legs stance is not directly comparable to the triple-stance phase because the vertices of the FWP have a different scaling. For this reason we only show the values referring to the triple stance. We can see that the same two trials, with and without external load, provide a completely different result if evaluated with the FWP-factor (Fig. 4-11 *bottom*): the blue-dashed line shows that the feasibility is lower in the case with external load. This is consistent with reality as in, when the load increases, even if the stability margin might improve, the risk of hitting the torque limits gets higher.

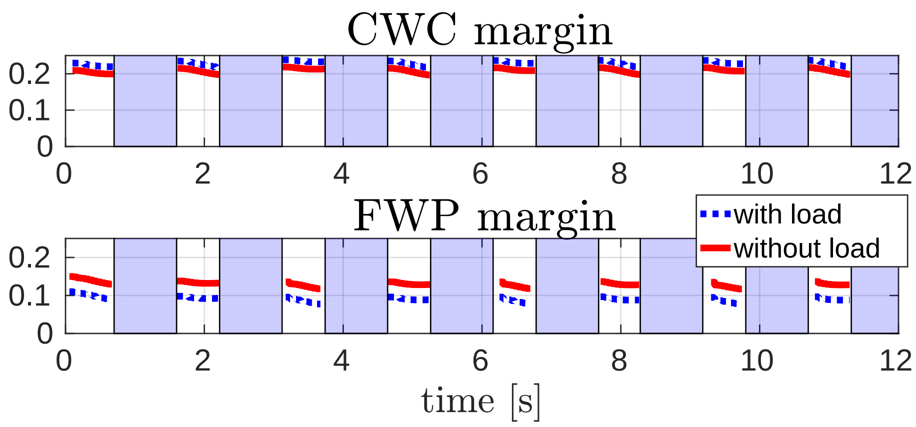


Figure 4-11: Evaluation of the heuristic crawl gait (with and without $20kg$ load) with the CWC margin (*top*) and with the FWP factor (*bottom*).

4-6-3 Craw Simulations

As shown in the accompanying video[3], we report a few simulation and hardware experiments of HyQ performing a crawling gait. At first we see that the heuristic crawl easily hits the torque limits while crawling on a flat ground while carrying an external load of $20kg$ (about 25% of the robot total weight) placed on its CoM. We can then see that the FWP planner, as explained in Section 4-5, finds a new duration Δt_{bm} of the *base motion* phase and a new CoM trajectory that avoids hitting the torque limits at all times, while maintaining the desired linear speed (*e.g.,* slower motion that results in smaller accelerations).
Final simulations show the capability of the planner to optimize feasible trajectories when the robot has a *hindered* joint (i.e. when a specific joint can only realize a significantly smaller torque than the other joints), or when we limit the normal force that a specific leg can realize on the ground.
The video also shows hardware experiments of HyQ crawling on a rough terrain composed of two parallel ramps with different slopes and heights. In this scenario only considering the friction cones for stability would lead the robot to place its CoM in such a way to homogeneously distribute the weight over all the stance legs. This would, however, neglect the different leg retraction of the left legs compared to the right legs and their consequent different torque amplification capabilities. The FWP presented in this Chapter, instead, tends to shift to CoM towards the lower ramp in such a way to add more load on those legs that are in a most favorable configuration.

[3]https://youtu.be/vUx5b5kfRfE

4-7 Summary

The complexity of a motion increases with the complexity of the terrain to be traversed. Moreover, there is a need for online motion replanning to avoid error accumulation. For these reasons, in this Chapter, we presented the concepts of AWP and FWP, two 6-dimensional spaces able to describe the set of feasible wrenches that a robot can exert without violating its actuation capabilities. Besides that, we also proposed a method to efficiently compute the $\mathcal{V}$-description of these two wrench sets. We then adapted a *feasibility factor*, originally proposed for grasping [72], to the $\mathcal{V}$-description of the FWP in order to study the locomotion stability and the actuation-consistency of a given motion plan. Finally we showed how this factor can be used in a CoM trajectory optimization not only for feasibility evaluation but also for motion planning.
Thanks to the efficiency of the vertex-based approach we are able to perform *online* TO where the robot plans during each *swing* the trajectory for the next *base motion* phase in a statically stable crawl [11]. Our approach does not take any assumption on the environment and it is therefore suitable for complex terrain scenarios.

The following list contains the main concepts and definitions that have been proposed in this Chapter:

1. the force polytopes $\mathcal{A}_i$ represent the result of the mapping of the legs' admissible joint torques to the set of admissible contact forces/wrenches at the end-effectors of a floating-base tree-structured robot. These quantities can be used to evaluate the capability of each individual limb to exert a force at its foot/hand in a specific configuration or to carry the robot's body weight (*e.g.,* a stretched leg may correspond to force polytope of reduced value and it can be therefore a good indicator of the necessity of taking a new step [137]);

2. if not intersected with the friction cones, the force polytopes $\mathcal{A}_i$ also consider the possibility of exerting negative normal contact forces and, therefore, they can also be employed for those robotics applications where one or more end-effectors are capable of grasping and/or pulling the ground;

3. unlike the friction cones $\mathcal{C}_i$, the force polytopes $\mathcal{A}_i$ are configuration dependent and, therefore, they need to be recomputed after every motion of the robot (even if no stance change occurs);

4. the Feasible Wrench Polytope (FWP) is a 6-dimensional set that can be efficiently obtained as the Minkowsky sum of all the admissible wrench sets $\hat{\mathcal{B}}_i$ of every contact point $i = 1, \dots n_c$ where n_c is the number of contact points: $FWP = \oplus_{i=1}^{n_c} \hat{\mathcal{B}}_i$. The bounded set $\hat{\mathcal{B}}_i$ is obtained from the intersection of the friction cones $\mathcal{C}_i$ with the force polytopes $\mathcal{A}_i$;

5. the FWP-based feasibility criterion states that if the gravito-inertial wrench $\mathbf{w}_{GI} \in \mathbb{R}^6$ lies within the limits of the FWP, the system is then dynamically stable and there exists then at least one set of joint torques that respects the system's actuation limits. This criterion is not limited to any specific terrain morphology and it can be employed for the analysis and the synthesis of both dynamic and static motions;

6. we formulated an online motion planner based on the approximating assumption that the robot's configuration, and thus the FWP limits, do not change significantly between two following steps of a quadrupedal walk. The motion planner is able to plan both feasible (*i.e.,* statically stable and actuation consistent) CoM trajectories and to optimize the duration of a locomotion phase in such a way to ensure that the robot's state always lies within the limits given by the halfspaces of the FWP.

Possible future works related to the FWP and to the wrench-based feasibility analysis my focus on the removal of the approximations mentioned in Section 4-5-1 and on the integration of planned and reactive locomotion behaviors [138].

Chapter 5

2D Actuation Constraints Projections

5-1 Introduction

The Actuation Wrench Polytope (AWP) and the Feasible Wrench Polytope (FWP) that have been described in the previous Chapter, can be considered as an extension of the Contact Wrench Cone (CWC) to the case where joint-torque limits affect the locomotion capabilities of legged robots on complex environments. The proposed approach is valid for arbitrary contacts (not limited to flat terrains) and to dynamic motions with non-zero linear and angular centroidal accelerations.
The computation time of six-dimensional bounded polytopes (AWP and FWP), however, increases considerably with respect to the case where only six-dimensional convex cones are involved (CWC). In the case of a quadruped robot, for example, the feasibility factor with respect to the FWP constraints can only be computed at about $10Hz$ in a triple stance phase and about $3Hz$ during a quadruple-support phase (see Tab. 4-1). These computation times, as much as the other performances reported in the continuation of this Chapter, have been achieved on a Intel(R) Core(TM) i5-4440 CPU @ 3.10GHz processor with 4 cores.
Online motion planning in Chapter 4 could only be achieved under the assumption that the robot configuration does not significantly differ from the configuration used to compute the FWP limits (see for example the approximations assumed in Section 4-5-1). These computational restrictions imposed by modern processors jeopardize the effectiveness of the FWP constraints.
For the above reasons, the approach presented in this Chapter introduces the concept of *actuation region* and *feasible region* as two-dimensional counterparts of the AWP and FWP respectively. These regions are the projections of the AWP and FWP in the Center of Mass (CoM)

space, obtained by assuming that only gravity acts on the robot. This quasi-static assumption enables the direct mapping from joint-torques to CoM space by means of a modified version of the Iterative Projection (IP) algorithm, originally proposed by Bretl *et al.* [55]. This allows a considerable improvement of the computational performances of our strategy, allowing us to check the feasibility margin with respect to friction cones and joint-torque constraints at a frequency of, at least, $66Hz$ for a triple support phase and $50Hz$ for a four-support phase of a quadruped. As later explained in this Chapter, this also enables the *online* optimization of actuation-aware foothold positions in rough terrains, besides the *online* CoM trajectory planning.

Besides that, the *actuation* and *feasible* regions, unlike the AWP and FWP, can be easily represented in $2D$. This allows us to give a clear and *intuitive* answer to legit questions like: how does the step length change with the increase of the robot's mass? And also: which is the height of the highest step that a robot can step on given its maximal joint torque/force capabilities?

Contributions:
In this Chapter I attempt to give an answer to the above important questions in the following way:

1. I will introduce a modified version of the IP algorithm, adapted to consider the joint-torque limits of legged robots in the form of force/wrench polytope constraints;

2. I will present the concept of *local* and *global* feasible regions and I will discuss their properties. These $2D$ areas provide an intuitive yet powerful understanding of the relation between locomotion capabilities and actuation limits;

3. I will apply the feasible region framework to a motion planning algorithm for legged robots, able to optimize CoM trajectories and foothold location online on arbitrary terrains for predefined step sequences and timings.

The two core building blocks of the work developed in this Chapter are the force/wrench polytopes, already presented in Section 4-2-1, and the IP algorithm, which I mentioned in Section 2-2-1 and which I will briefly describe in the next Section.
Notice that the Figs. 5-5, 5-8 and 5-10 and the Voronoi tassellation-based strategy (is Section 5-3-3) presented in this Chapter have been created by my collaborator Dr. Stéphane Caron[1], who produced them in perspective of a future joint publication, and they have been included in this dissertation with his permission.

[1]LIRMM, CNRS-University of Montpellier, email: stephane.caron@lirmm.fr

5-2 Related Works

The Iterative Projection (IP) algorithm is a method introduced by Bretl *et al.* [55] for the computation of stable support regions for articulated robots having multiple contacts with the environment in arbitrary locations, having arbitrary surface normals and friction coefficients. The IP belongs to a family of cutting-plane computational methods that allow to approximate a target convex set $\mathcal{Y}$ up to a predefined tolerance value. The tuning of this tolerance allows to conveniently adjust the computational performances of the algorithm according to the requirements of the specific application: it enables a rough but fast set reconstruction for high tolerance values and also, on the opposite, a precise set reconstruction with longer solve times when the tolerance value is low. Bretl *et al.* have applied this strategy to the field of legged locomotion with the goal to reconstruct the $2D$ friction-consistent support region $\mathcal{Y}_f$ (see Section 2-2-1 for a definition of support region). In [55], the support region is defined as: *the horizontal cross section the convex cylinder that represents the set of CoM positions at which contact forces exist that compensate for gravity without causing slip (for given foot placements with associated friction models).* In the remainder of this Chapter the support region will be referred to under the name of *friction region*, in order to reduce the confusion with other similar regions that will be defined in the following Sections.

Algorithm 1 reports the procedure presented in [55]; you can see that the algorithm consists of recursively solving a Second Order Cone Program (SOCP) that maximizes the horizontal position of the CoM $\mathbf{c}_{xy} \in \mathbb{R}^2$ along the direction defined by the unit vector $\mathbf{a}_i \in \mathbb{R}^2$ (i being in this case the iteration index) while satisfying the friction constraints.

```
Input: $\mathbf{y}, \mathbf{p}_1, \dots \mathbf{p}_{n_c}, \mathbf{n}_1, \dots \mathbf{n}_{n_c}, \mu_1, \dots, \mu_{n_c}$;
Result: friction region $\mathcal{Y}_f$
initialization: $\mathcal{Y}_{outer}$ and $\mathcal{Y}_{inner}$;
while $area(\mathcal{Y}_{outer}) - area(\mathcal{Y}_{inner}) > \epsilon$ do
    I) compute the edges of $\mathcal{Y}_{inner}$;
    II) pick the edge cutting off the largest fraction of $\mathcal{Y}_{outer}$;
    III) solve the SOCP:
            $\max_{\mathbf{c}_{xy}, \mathbf{f}} \quad \mathbf{a}_i^T \mathbf{c}_{xy}$
         such that:
            (III.a)  $\mathbf{A}_1 \mathbf{f} + \mathbf{A}_2 \mathbf{c}_{xy} = \mathbf{t}$
            (III.b)  $\|\mathbf{B}\mathbf{f}\|_2 \le \mathbf{u}^T \mathbf{f}$
    IV) update the outer approximation $\mathcal{Y}_{outer}$;
    V) update the inner approximation $\mathcal{Y}_{inner}$;
end
```

Algorithm 1: Bretl's IP algorithm

The algorithm considers the robot's CoM position $\mathbf{c}$, mass m, set of n_c point contacts $\mathbf{p}_1, \dots \mathbf{p}_{n_c}$ with corresponding surface normals $\mathbf{n}_1, \dots \mathbf{n}_{n_c}$ and friction coefficients $\mu_1, \dots, \mu_{n_c}$. The constraint (III.a) enforces the static equilibrium of the forces and moments acting on the robot due to gravity $\mathbf{g}$ and contact forces $\mathbf{f} = [\mathbf{f}_1^T, \dots, \mathbf{f}_{n_c}^T]^T \in \mathbb{R}^{3n_c}$. Eq. (III.b) ensures that the friction cones constraint is satisfied. $\mathbf{P} \in \mathbb{R}^{2\times 3}$ is a selection matrix that selects the x, y components of the CoM. The matrix $\mathbf{A}_1$ represents the grasp matrix of the set of point contacts[2], $\mathbf{A}_2$ computes the x, y angular components τ_x and τ_y of the wrench generated by the action of gravity on the CoM $\mathbf{c}$ of the robot expressed with respect to a fixed frame. The matrix $\mathbf{B}$ projects the contact forces $\mathbf{f}_i$ into the local contact frame and $\mathbf{u} \in \mathbb{R}^{3n_c}$ considers the limits of the friction cones in the local contact frames:

$$\mathbf{A}_1 = \begin{bmatrix} \mathbb{1}_3 & \dots & \mathbb{1}_3 \\ [\mathbf{p}_1]\times & \dots & [\mathbf{p}_{n_c}]\times \end{bmatrix} \in \mathbb{R}^{6\times 3n_c}, \quad \mathbf{A}_2 = \begin{bmatrix} \mathbb{0} \\ -m\mathbf{g} \times \mathbf{P}^T \end{bmatrix} \in \mathbb{R}^{6\times 2}, \quad \mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$
$$\mathbf{t} = \begin{bmatrix} -m\mathbf{g} \\ 0 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \mu_1 \mathbf{n}_1 \\ \vdots \\ \mu_{n_c} \mathbf{n}_{n_c} \end{bmatrix} \quad \text{and} \quad \mathbf{B} = diag(\mathbb{1}_3 - \mathbf{n}_1\mathbf{n}_1^T, \dots, \mathbb{1}_3 - \mathbf{n}_{n_c}\mathbf{n}_{n_c}^T) \tag{5-1}$$

Once defined the above quantities, the *friction region* $\mathcal{Y}_f$ was defined by Bretl *et al.* [55] as the set of CoM coordinates $\mathbf{c}_{xy} \in \mathbb{R}^2$ orthogonal to gravity for which there exists a set of contact forces that respects both static equilibrium and friction constraints:

$$\mathcal{Y}_f = \left\{ \mathbf{c}_{xy} \in \mathbb{R}^2 \quad | \quad \exists \mathbf{f} \in \mathbb{R}^{3n_c} \, \text{s.t.} (\mathbf{c}_{xy}, \mathbf{f}) \in \mathcal{C} \right\} \tag{5-2}$$

where:

$$\mathcal{C} = \left\{ \mathbf{f} \in \mathbb{R}^{3n_c}, \quad \mathbf{c}_{xy} \in \mathbb{R}^2 \quad | \quad \mathbf{A}_1\mathbf{f} + \mathbf{A}_2\mathbf{c}_{xy} = \mathbf{t}, \quad \|\mathbf{B}\mathbf{f}\|_2 \le \mathbf{u}^T\mathbf{f} \right\} \tag{5-3}$$

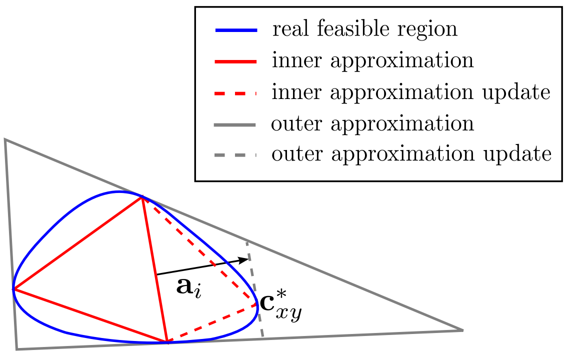


Figure 5-1: Single step of Bretl's iterative projection algorithm

Fig. 5-1 shows the process corresponding to one iteration of the IP algorithm reported in Alg. 1. As it can be seen in step III, the IP does not only maximize the horizontal CoM projection $\mathbf{c}_{xy}$ along the direction $\mathbf{a}_i \in \mathbb{R}^2$, but it also finds a feasible set of contact forces $\mathbf{f}$

[2] $[\cdot]\times$ represents the skew-symmetric operator associated to the cross product.

that respects static equilibrium and friction cone constraints.
Alg. 1 can also be regarded as a projection of the feasible set $\mathcal{C}$ onto a lower dimensional region whose boundary represent the location of the CoM for which the τ_x, τ_y components of the gravity wrench can be equilibrated by the contact forces for a given set of contact points. Exploiting the assumption that the only external force acting on the CoM is gravity we then get a one-to-one mapping between the torque components and the corresponding CoM (x, y) coordinates:

$$c_x = \frac{\tau_y}{mg}, \quad c_y = -\frac{\tau_x}{mg} \tag{5-4}$$

The friction region, as defined in Eq. 5-2, is a 2D convex set but it is not, in general, a linear set (*i.e.,* it is not a polygon). The inner and outer approximations $\mathcal{Y}_{inner}$ and $\mathcal{Y}_{outer}$ used to estimate $\mathcal{Y}_f$ are, however, always 2D polygons by construction. For this reason (and for the additional motivations given in Section 2-2-1) I will therefore refer to $\mathcal{Y}_f$ in the rest of this dissertation with the term *friction region* rather than *friction polygon.*

In the next Section we will see how I modified Alg. 1 in order to obtain a 2D set that does not only respect the static equilibrium and the friction cone constraints, but it is also consistent with the actuation capabilities of the system.
A partial list of the main symbols employed in the remainder of this Chapter and throughout this dissertation is contained in the Notation section.

5-3 The 2D Actuation and Feasible Regions

Algorithm 1 can be extended in a straightforward manner to include also the static force/wrench polytope constraint that I already discussed in Eq. 4-17. The construction and analysis of the resulting 2D polygons will be the topic of the following Section.

5-3-1 The Local Actuation and Feasible Regions

In this Section I propose a slight variations of Alg. 1 to include also the static force polytope $\mathcal{A}_i$ of every individual robot's foot/end-effector in contact with the ground/environment. The resulting procedure can be found in Alg. 2.

```
Input: $\mathbf{y}, \mathbf{p}_1, \dots \mathbf{p}_{n_c}, \mathbf{n}_1, \dots \mathbf{n}_{n_c}, \mu_1, \dots, \mu_{n_c}, \mathbf{g}_1, \dots \mathbf{g}_{n_c}, \mathbf{J}_1, \dots \mathbf{J}_{n_c}, \boldsymbol{\tau}_1^{lim}, \dots \boldsymbol{\tau}_{n_c}^{lim}$;
Result: local feasible region $\mathcal{Y}_{fa}$
initialization: $\mathcal{Y}_{outer}$ and $\mathcal{Y}_{inner}$;
while $area(\mathcal{Y}_{outer}) - area(\mathcal{Y}_{inner}) > \epsilon$ do
    I) compute the edges of $\mathcal{Y}_{inner}$;
    II) pick the edge cutting off the largest fraction of $\mathcal{Y}_{outer}$;
    III) solve the SOCP:
        $\max_{\mathbf{c}_{xy},\mathbf{f}} \quad \mathbf{a}_i^T \mathbf{c}_{xy}$
        such that :
            (III.a) $\mathbf{A}_1\mathbf{f} + \mathbf{A}_2\mathbf{c}_{xy} = \mathbf{t}$
            (III.b) $\|\mathbf{B}\mathbf{f}\|_2 \leq \mathbf{u}^T\mathbf{f}$
            (III.c) $\mathbf{C}\mathbf{f} \leq \mathbf{d}$
    IV) update the outer approximation $\mathcal{Y}_{outer}$;
    V) update the inner approximation $\mathcal{Y}_{inner}$;
end
```

Algorithm 2: Actuation and Friction consistent IP algorithm

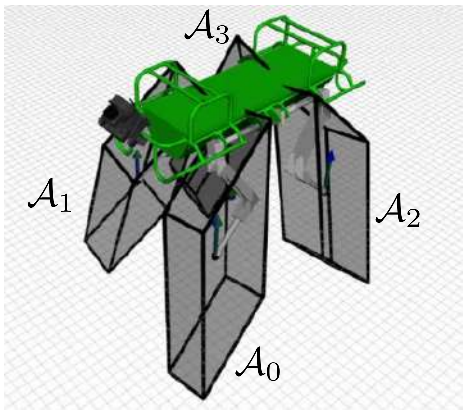


Figure 5-2: Representation of the force polytopes of the HyQ robot in a four-stance configuration ($n_c = 4$).

For compactness I report Eq. 4-17, representing the definition of force/wrench polytopes $\mathcal{A}_i$ of the $i-th$ limb in contact with the environment:

$$\mathcal{A}_i = \Big\{ \mathbf{f}_i \in \mathbb{R}^m \quad | \quad \mathbf{C}_i\mathbf{f}_i = \mathbf{d}_i, \quad -\boldsymbol{\tau}_i^{lim} \leq \boldsymbol{\tau}_i \leq \boldsymbol{\tau}_i^{lim} \Big\} \tag{5-5}$$

where:

$$\mathbf{C}_i = \mathbf{J}_i^T \in \mathbb{R}^{n_l \times m} \quad \text{and} \quad \mathbf{d}_i = \mathbf{g}(\mathbf{q}_i) - \boldsymbol{\tau}_i \in \mathbb{R}^{n_l} \tag{5-6}$$

where n_l is the number of actuated joints of the $i-th$ limb. Differently from the original IP algorithm, I consider in this case the possibility of contact torques being applied at the end-effectors of the robot in contact with the environment; as a consequence I define each individual contact wrench $\mathbf{f}_i \in \mathbb{R}^m$ where $m = 3$ if the considered end-effector is perturbed by a pure force and $m = 6$ if, instead, also a contact torque component is given.

Considering the the joint-space torque variable τ_i Eq. 5-6 only depends on its minimum and maximum values $\boldsymbol{\tau}_i^{lim}$, we can then further re-write Eq. 5-5 to explicitly highlight its dependency from $\boldsymbol{\tau}_i^{lim}$:

$$\mathcal{A}_i = \Big\{ \mathbf{f}_i \in \mathbb{R}^m \quad | \quad \mathbf{C}_i\mathbf{f}_i \leq \mathbf{d}_i^{lim} \Big\} \tag{5-7}$$

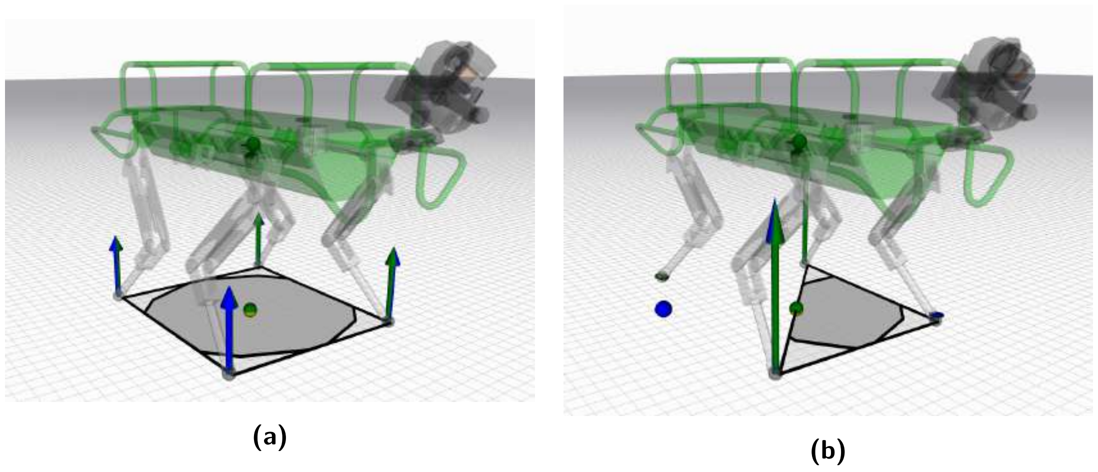


Figure 5-3: Classical friction region (light gray) and feasible region (dark gray) in four-stance (a) and triple-stance (b) conditions.

where:

$$\mathbf{d}_i^{lim} = \mathbf{g}(\mathbf{q}_i) - \boldsymbol{\tau}_i^{lim} \tag{5-8}$$

Eq. 5-7 thus represents a compact notation for the static force polytope initially defined in Eq. 4-17. I will now exploit this notation to introduce the matrix $\mathbf{C} \in \mathbb{R}^{d\times(m\cdot n_c)}$ and the vector $\mathbf{d}$, result of the concatenation of all the matrices $\mathbf{C}_i$ and vectors $\mathbf{d}_i^{lim}$ of all the individual limbs in contact with the environment:

$$\begin{aligned} \mathbf{C} =& diag(\mathbf{C}_1, \dots, \mathbf{C}_{n_c}) \in \mathbb{R}^{d\times(m\cdot n_c)} \\ \mathbf{d} =& \begin{bmatrix} \mathbf{d}_1^{lim} \\ \vdots \\ \mathbf{d}_{n_c}^{lim} \end{bmatrix} \in \mathbb{R}^d \end{aligned} \tag{5-9}$$

where $d = \sum_{i=1}^{n_c} n_{l,i}$ is the sum of the number of joints of all the limbs in contact with the environment (*e.g.*, $d = n$ if all the limbs of the robot are in contact). $\mathbf{C}$ and $\mathbf{d}$ can now be used to redefine the set of actuation-consistent forces/wrenches $\mathcal{A}$ that satisfy all the individual force polytopes $\mathcal{A}_i$ for $k = 1, \dots, n_c$:

$$\mathcal{A} = \Big\{ \mathbf{f} \in \mathbb{R}^{mn_c}, \quad \mathbf{c}_{xy} \in \mathbb{R}^2 \quad | \quad \mathbf{C}\mathbf{f} \leq \mathbf{d} \Big\} \tag{5-10}$$

In analogy with Eq. 5-2, we can define a new set of actuation-consistent CoM positions called *local actuation region*:

$$\mathcal{Y}_a = \Big\{ \mathbf{c}_{xy} \in \mathbb{R}^2 \quad | \quad \exists \mathbf{f} \in \mathbb{R}^{mn_c} \, \text{s. t.} (\mathbf{c}_{xy}, \mathbf{f}) \in \mathcal{A} \Big\} \tag{5-11}$$

As a further observation we notice that we are interested in computing the set of CoM positions $\mathcal{Y}_{fa}$ that simultaneously satisfies both the friction and the actuation constraints (see Fig. 5-3). This can be obtained by considering the intersection of $\mathcal{C}$ and $\mathcal{A}$:

$$\mathcal{C} \cap \mathcal{A} = \Big\{ \mathbf{f} \in \mathbb{R}^{mn_c}, \quad \mathbf{c}_{xy} \in \mathbb{R}^2 \quad | \quad \mathbf{A}_1\mathbf{f} + \mathbf{A}_2\mathbf{c}_{xy} = \mathbf{t}, \quad \|\mathbf{Bf}\|_2 \leq \mathbf{u}^T\mathbf{f}, \quad \mathbf{Cf} \leq \mathbf{d} \Big\} \tag{5-12}$$

Based on Eq. 5-12, the friction- and actuation-consistent region $\mathcal{Y}_{fa}$, called *local feasible region*, can be defined as:

$$\mathcal{Y}_{fa} = \Big\{ \mathbf{c}_{xy} \in \mathbb{R}^2 \quad | \quad \exists \mathbf{f} \in \mathbb{R}^{mn_c} \, \mathrm{s.\,t.} (\mathbf{c}_{xy}, \mathbf{f}) \in \mathcal{C} \cap \mathcal{A} \Big\} \tag{5-13}$$

In analogy with Alg. 1, Alg. 2 explains how $\mathcal{Y}_{fa}$ can be computed efficiently.

Simultaneously imposing the inequality constraints (III.b) and (III.c) in Alg. 2 corresponds to performing an intersection of the polytopes $\mathcal{A}_i$ with the friction cone $\mathcal{C}_i$ of the corresponding contact point. This yields the set of all the contact forces that simultaneously respect both the friction cone constraints and the joint actuation limits of the $j-th$ limb (see for example Fig. 4-3). This corresponds to the *bounded friction cone* $\mathcal{B}_i$ that we defined in Section 4-3-3. Alg. 2, in practice, is equivalent to Alg. 1 with the only difference being the constraint (III.c) relative to the actuation limits.
Another variant of the same IP algorithm can be formulated to compute the *actuation region* $\mathcal{Y}_a$ that only considers actuation constraints and no friction constraints. This can be obtained by simply removing the constraint (III.b) from the SOCP that is solved at the step III of Alg. 2. In this case, being (III.b) the only quadratic constraint, the maximization problem will then turn into a linear program (LP).
Intermediate cases exist where some end-effector present unilateral contacts and other limbs present instead bilateral contacts (*e.g.,* when a robot is climbing a ladder pushing with its feet and pulling with his hands). Such conditions can be captured by the presented IP modification by enforcing only the force polytope constraints on the bilateral contact points and by enforcing both friction pyramids and force polytopes on the unilateral contacts.

The force/wrench polytope $\mathcal{A}_i$, unlike the friction cones $\mathcal{C}_i$, is a configuration-dependent quantity and, as a consequence, its vertices will change whenever the robot changes its configuration. In order to highlight this property, we refer to the resulting friction- and actuation-consistent feasible region $\mathcal{Y}_{fa}$ with the name of *local feasible region.* The term *local* points out the fact that the feasible region $\mathcal{Y}_{fa}$ can be considered to be accurate only in a *neighborhood* of the current robot configuration. The distance between the current CoM projection $\mathbf{c}_{xy}$ and the edges of $\mathcal{Y}_{fa}$ can be considered as a combined measure of the *local* or *instantaneous* robustness of the robot's state with respect to the contacts' stability and joint-space torque limits.
The friction cones (and the friction region $\mathcal{Y}_f$) are only dependent on the contacts' configuration and can thus be recomputed at every stance change. This is a very convenient property to embed the contacts' stability in a motion planning formulation. The force/wrench polytopes (and thus the feasible region $\mathcal{Y}_{fa}$), instead, because of their *local validity* must be recomputed

at every change of configuration and thus make the motion planning formulation harder. However, their local validity is also the key element of the local actuation polytopes that, if properly exploited, can provide an insightful view on the relationship between robot configuration and maximal exerted force at the end-effectors. In the remainder of this Chapter we will drop the adjective *local* for the sake of compactness, however the actuation polytopes $\mathcal{A}_i$ should always be regarded as instantaneous configuration dependent quantities.

Connecting the Feasible Regions and the Feasible Polytopes

In the same way like the friction region $\mathcal{Y}_f$ can be seen as a particular case of the CWC criterion with only gravity acting on the CoM of the robot, in the same way also the local actuation region $\mathcal{Y}_a$ can be seen as a specific case of the AWP and the feasible region $\mathcal{Y}_{fa}$ can be seen as a specific case of the FWP (previously presented in Section 4) as shown in Tab. 5-1.

Table 5-1: Analogies between 2D feasible regions and 6D feasible polytopes

constraint	$2D$ CoM space	$6D$ centroidal wrench	validity
friction	friction region $\mathcal{Y}_f$	CWC	global
joint-torques	actuation region $\mathcal{Y}_a$	AWP	local
friction and torques	feasible region $\mathcal{Y}_{fa}$	FWP	local
joint-torques	actuation region $\mathcal{G}_a$	-	global
friction and torques	feasible region $\mathcal{G}_{fa}$	-	global

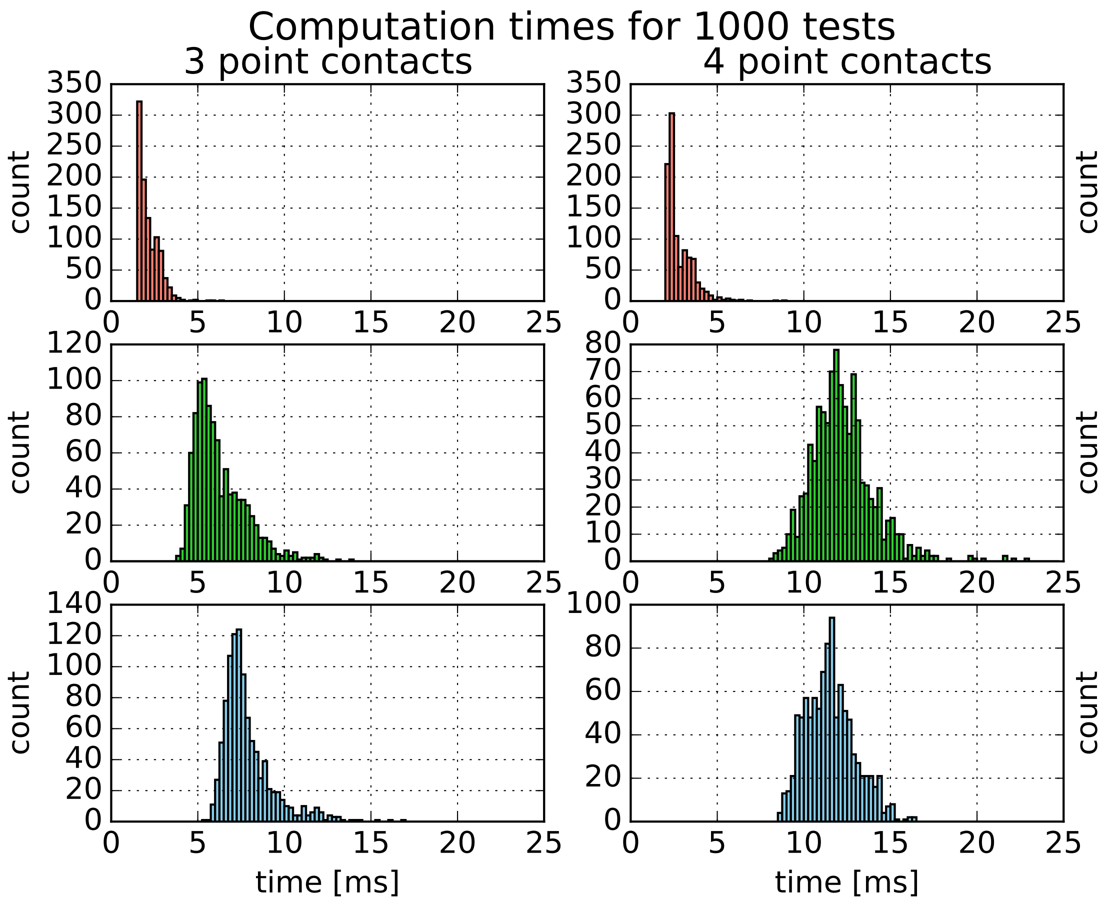


Figure 5-4: Computation time for IP algorithm with only linearized friction cone constraints (red), only force polytope constraints (green) and both friction and actuation constraints (blue). These statics were collected on a 4-cores Intel(R) Core(TM) i5-4440 CPU @ 3.10GHz processor.

This can be demonstrated by slicing, for example, the six-dimensional AWP (FWP) in correspondence of the planes: $f_x = 0, f_y = 0, f_z = mg$ and $\tau_z = 0$. In this way only two Degrees of Freedom (DoFs) are left which correspond to the τ_x and τ_y coordinates of the wrench space. The two-dimensional region that results from this slicing procedure can then be mapped into a set of feasible CoM coordinates $\mathbf{c}_{xy}$ that corresponds indeed to the actuation consistent region $\mathcal{Y}_a$ ($\mathcal{Y}_{fa}$).

Computing the AWP or the FWP, however, can be computationally demanding because of the high dimensionality and large amount of halfspaces and vertices. This is what motivated me to propose a variant of the IP algorithm that allows to directly map joint-torques constraints into $2D$ CoM limits. The computational improvements achieved by this choice are presented in the following Section.

Local Regions Computation Time

The usage of the IP algorithm implies a significant speed up for the computation of the actuation region reaching average computation times in the order of milliseconds (see Fig.

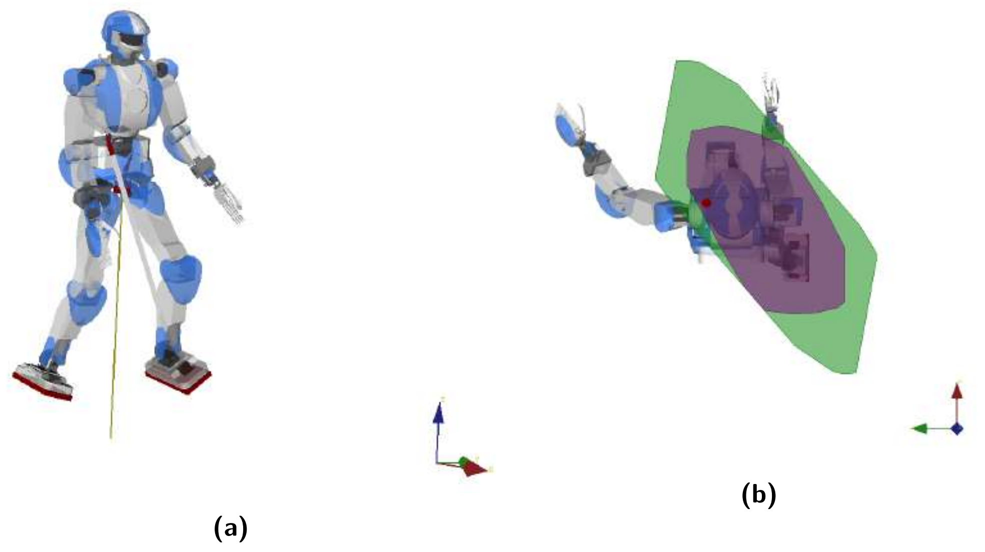


Figure 5-5: Local feasible region (blue) and friction region (green) for the HRP-4 humanoid robot in a configuration with non-coplanar contacts. This Figure has been produced by Dr. Stéphane Caron (LIRMM, CNRS, France) who created it under my request in perspective of a future joint publication and he agreed with the display of this Figure in this dissertation.

5-4).

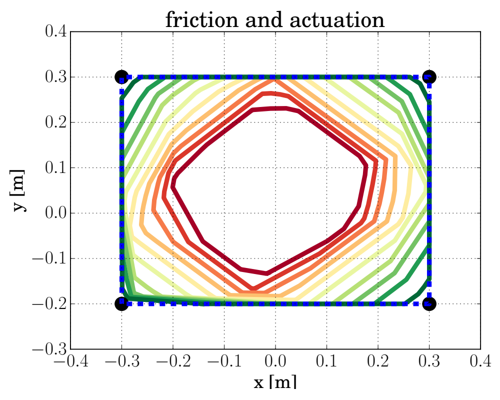
The solve time of the IP algorithm depends on the number of inequality constraints embedded in it (only friction constraints, only actuation constraints, or both). The most favorable scenario is when only friction cones are considered (red in Fig. 5-4): in the case of linearized friction cones with four facets per pyramid, the IP will present $4n_c$ inequalities. The least convenient scenario is instead when both friction pyramids and force/wrench polytope constraints are considered (blue in Fig. 5-4), in this case the IP will include $(4 + 2n_l)n_c$ inequalities (assuming that all the limbs in contact with the ground have same number of DoFs n_l and that the friction cones are linearized with 4 halfspaces). In the case of the Hydraulically actuated Quadruped (HyQ) quadruped this will result in 10 inequalities per contact foot; in the case of a humanoid robot with 6 DoFs per leg, instead, this will result in 14 inequalities per contact foot. Figure 5-5, for example, shows the friction region $\mathcal{Y}_f$ (green) and the feasible region $\mathcal{Y}_{fa}$ in the case of the HRP-4 robot standing still in a configuration with non-coplanar contacts. The last row of Fig. 5-4 shows that, even in such inconvenient condition where all contacts are subject to both friction and actuation constraints, the solve time is below $20ms$ in a four-stance configuration and below $15ms$ in a triple-stance configuration in 99.5% of the computations (blue histogram). This allows the efficient computation of the local feasible region at a frequency of, at least, $50Hz$ in a four stance configuration and $66Hz$ in a triple stance configuration of a quadruped robot. These frequencies could be further increased by reducing the tolerance factor of the IP algorithm.

Modern legged robots are equipped with powerful actuators that are able to sustain the robots own weight even on one single leg in a predefined convenient configuration. Even for such powerful robots, the feasible region can represent a useful tool to intuitively visualize the payload dependency on the specific gait or, finally, the maximal static displacement of the CoM in presence of limited torques and/or damaged limb.

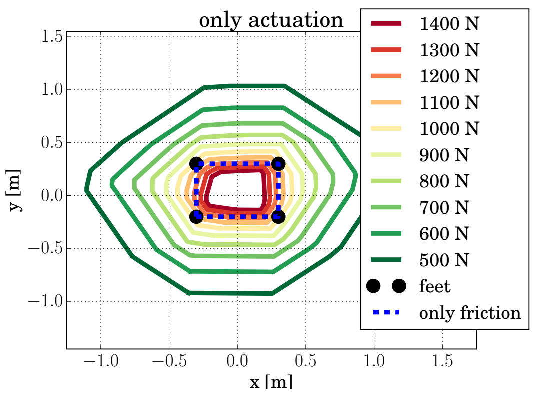
Friction- and Actuation-Consistent Whole-Body Controllers
Having the CoM inside the feasible region $\mathcal{Y}_{fa}$ ensures the existence of a feasible set of contact forces that satisfy the force polytope constraints and the friction cone constraints. We still do not know, however, the exact value of the forces corresponding to this feasible solution. From a control point of view, it is therefore of paramount importance for this approach that a whole-body controller is developed that is capable of determining the sought feasible solution. If the CoM projection lies instead outside of $\mathcal{Y}_{fa}$ then we can conclude that either the friction constraints or the joint-torques limits (or both) will be violated for that specific state of the robot.
The hereby proposed feasible region $\mathcal{Y}_{fa}$ still plays a role in the field of motion planning for the benchmarking of the performances of whole-body controllers (to make sure that they can still find a feasible solution when the CoM projection $\mathbf{c}_{xy}$ lies inside $\mathcal{Y}_{fa}$ or on the edge of $\mathcal{Y}_{fa}$).
Besides this, the $\mathcal{Y}_{fa}$ does not suffer from limitations to specific robot morphologies or specific terrains (*e.g.,* flat terrains). As a consequence the $\mathcal{Y}_{fa}$ can be employed for motion planning of legged robots on rough and complex terrains, where classical simplified models fail because the feasibility constraints, such as joint torque limits, affect more and more the robot's navigation capabilities.
As a showcase for humanoid robots, an example of local feasible region $\mathcal{Y}_{fa}$ for the HRP-4 robot is shown next.

5-3-2 Comparison of Local Feasible Regions

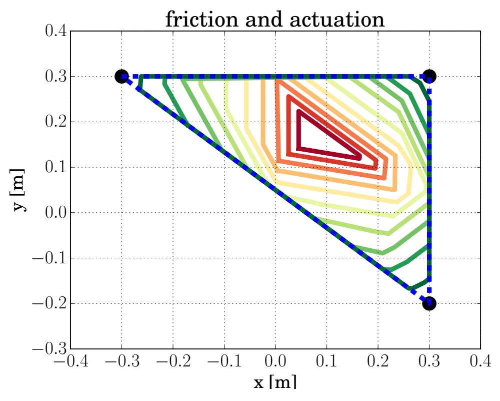
Fig. 5-6 reports various tests of computation of feasible $2D$ areas for different loads applied on the CoM of the robot. This is analogous to computing the feasible regions for different percentages of the torque limits while keeping the load on the robot fixed. The blue dashed lines represent the classical friction region $\mathcal{Y}_f$ as defined by Bretl *et al.* [55].
Figs. 5-6a and 5-6c depict the feasible regions $\mathcal{Y}_{fa}$ for the HyQ robot with four and three coplanar stance feet. Figs. 5-6b and 5-6d, instead, depict the actuation regions $\mathcal{Y}_a$ for the HyQ robot in the same configurations with four and three coplanar stance feet. Such actuation consistent areas $\mathcal{Y}_a$ alone are not directly applicable in the field of legged locomotion where robots typically make and break contacts using their feet and have therefore no possibility to grasp the terrain. Feasible regions $\mathcal{Y}_{fa}$ should be used instead since they include the friction constraints that also encode the unilaterality constraint. Actuation-consistent regions $\mathcal{Y}_a$



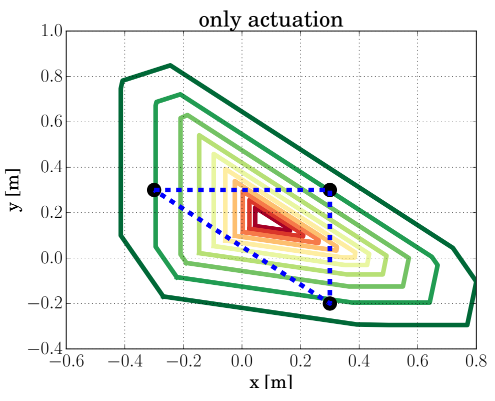
(a)



(b)



(c)



(d)

Figure 5-6: Relation between the load acting on the CoM of the robot (in N) and the shape of the instantaneous feasible region. We can see that the heavier the load, the smaller the area of the feasible regions. The black points represent the stance feet positions of the HyQ quadruped during a four and triple support phases; the dashed blue lines represent the feasible region obtained by consideration of friction constraints only.

however, can be useful in other fields of robotics such as manipulation and or whenever a robot has bilateral contacts with the ground as in the case of climbing robots with magnetic grippers [139] or in the case of heavy-duty walking machines with predefined footstep locations such as the robots of the TITAN series [58] (*e.g.*, see Fig. 2-9b). As visible in Figs. 5-6b and 5-6d the robot's CoM might lean outside of the classical friction region $\mathcal{Y}_f$ (dashed blue line) depending on the magnitude of the load acting on it. This is because the actuation region $\mathcal{Y}_a$ does not consider the friction constraints and, as a consequence, if considered alone, it assumes that also negative normal contact forces are admissible (*i.e.*, pulling from the ground). $\mathcal{Y}_a$ is therefore a valuable tool for all those robotic applications that involve the capability of grasping the ground with one or more of their limbs (*e.g.*, robotics hands, magnetic grippers, *etc.*).

As a final consideration, comparing the figures related to the same number of stance feet

(Fig. 5-6d compared with 5-6c and Fig. 5-6b compared with 5-6a) you can see that the feasible region $\mathcal{Y}_{fa}$ cannot be obtained by simple intersection of the friction region $\mathcal{Y}_f$ and the actuation region $\mathcal{Y}_a$. Although this approximation might be accurate under specific conditions, in general the intersection and projection operators do not commute [56]. Let us consider $\mathcal{C}$ to be the set of contact forces and CoM positions $\mathbf{c}_{xy}$ that respect the static equilibrium and the friction constraints (see Eq. 5-3); let us then also consider $\mathcal{A}$ defined as the set of contact forces and CoM horizontal positions $\mathbf{c}_{xy}$ that respect the static equilibrium, the force polytopes and the friction cones constraints (see Eq. 5-7). We therefore define the *friction region* [55] as:

$$\mathcal{Y}_f = IP(\mathcal{C}), \tag{5-14}$$

the *local actuation region* as:

$$\mathcal{Y}_a = IP(\mathcal{A}) \tag{5-15}$$

and the (actuation- and friction-consistent) *local feasible region* as:

$$\mathcal{Y}_{fa} = IP(\mathcal{C} \cap \mathcal{A}) \tag{5-16}$$

where IP is the Iterative Projection operator. The non-commutativity of projections and intersection can be stated as:

$$\mathcal{Y}_{fa} \neq \mathcal{Y}_f \cap \mathcal{Y}_a \tag{5-17}$$

and, in particular, the following inclusion always holds:

$$\mathcal{Y}_{fa} \subseteq \mathcal{Y}_f \cap \mathcal{Y}_a \tag{5-18}$$

$\mathcal{Y}_{fa}$ is therefore more conservative than the intersection of $\mathcal{Y}_f$ and $\mathcal{Y}_a$. Intuitively, Eq. 5-18 might be explained by considering that there may exist CoM positions that, in same the time:

1. provide feasible force/wrench solutions if the friction cones or force polytopes constraints are considered *individually*;
2. provide unfeasible force/wrench solutions if the friction cones or force polytopes constraints are considered *simultaneously*;

However, the opposite in not possible and consequently $\mathcal{Y}_{fa}$ has to lie inside the intersection of $\mathcal{Y}_f$ and $\mathcal{Y}_a$ as stated in Eq. 5-18.

5-3-3 The Global Actuation and Feasible Regions

Fig. 5-7 reports the results of a few tests where I computed the local actuation region for different CoM positions (along the same segment from $(0, -0.3)$ to $(0, +0.3)$) and for the same set of contact points. As previously anticipated, the local actuation regions $\mathcal{Y}_{fa}$ change as a

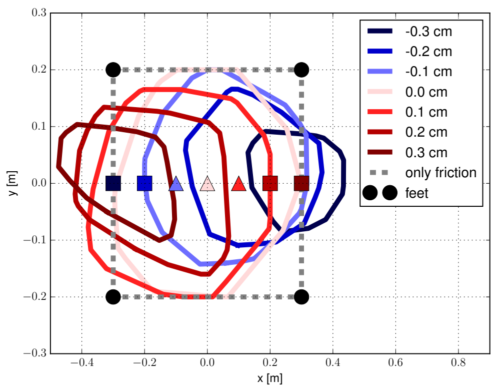


Figure 5-7: Local actuation areas for the same footholds position and for different CoM positions along the same segment. The triangular markers represent those CoM positions that do not belong to their correspondent local feasible region. The squared markers, instead, represent the tested CoM positions that are inside their correspondent local feasible region.

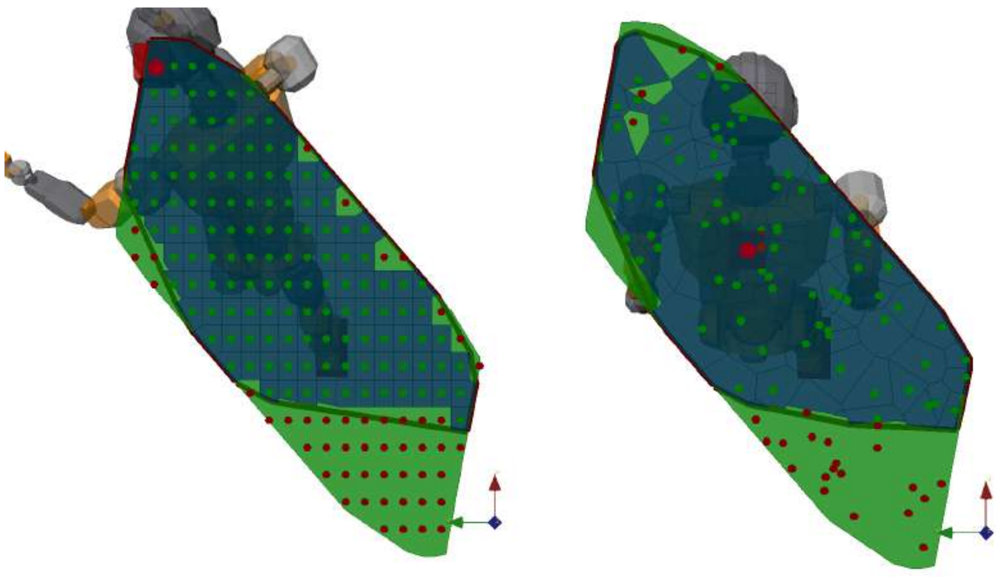
function of the robot configurations while the friction region $\mathcal{Y}_f$ (dashed blue line) is constant as it only depends on the stance locations.

By inspection of the resulting actuation areas $\mathcal{Y}_a$ you can see that some of the CoM positions $\mathbf{c}_{xy}$ used for their computation do not lie within their corresponding local actuation region; such points are marked with squared markers. This is a degenerate condition in which the set of feasible forces/wrenches (constrained by the force/wrench polytopes and thus, ultimately, depending on the limb Jacobians $\mathbf{J}_i$ and on the torque limits $\boldsymbol{\tau}_i^{lim}$) are not able to withstand the weight of the robot lumped into the specified value of $\mathbf{c}_{xy}$. Those cases are therefore to be considered unfeasible, even if the area of the resulting local actuation region is not null. Such degenerate areas might be useful in the very unlikely cases where the CoM position $\mathbf{c}_{xy}$ changes without changing the robot configuration (*e.g.,* when an external asymmetric static load is added onto the trunk of the robot).
Those points that, instead, do belong to their respective local actuation region are instead marked with triangular markers.
Be repeating the test shown in Fig. 5-7 along multiple directions and with multiple $\mathbf{c}_{xy}$ positions (and, consequently different robot configurations) one can notice that the set of feasible CoM positions results in a convex set that we name *global actuation region* $\mathcal{G}_a$:

Definition: we define the static *global actuation region* $\mathcal{G}_a$ as the set of all CoM coordinates $\mathbf{c}_{xy} \in \mathbb{R}^2$, orthogonal to the direction of gravity, where the robot is able to withstand its own weight, considering its own kinematics and torque limits.

The definition above does not include the unilaterality of the contact forces and the friction constraints. We thus define a second global region that also considers such features:

Definition: we define the static *global feasible region* $\mathcal{G}_{fa}$ as the set of all CoM coordinates



(a) Grid sampling and rectangular cells partitioning.

(b) Random uniform sampling with Voronoi tassellation.

Figure 5-8: Friction region (green) and global feasible region (blue) for the JVRC-1 humanoid robot. This Figure has been produced by Dr. Stéphane Caron (LIRMM, CNRS, France) who created it in perspective of a future joint publication and he agreed with the display of this Figure in this dissertation.

$\mathbf{c}_{xy} \in \mathbb{R}^2$, orthogonal to the direction of gravity, where the robot is able to withstand its own weight, considering its own kinematics, its torque limits, the unilaterality of the contact forces and the friction constraints.

$\mathcal{G}_a$ and $\mathcal{G}_{fa}$, unlike their local counterparts $\mathcal{Y}_a$ and $\mathcal{Y}_{fa}$ do not depend on the instantaneous robot configuration but they only depend on the position of the stance feet.

Fig. 5-8 shows the global feasible area $\mathcal{G}_{fa}$ (blue area) and the friction region $\mathcal{Y}_f$ (green area) for the Japan Virtual Robotics Challenge (JVRC-1) humanoid robot simulated in OpenRave [140].

The blue region $\mathcal{G}_{fa}$ in Fig. 5-8a was obtained by consecutively computing the local actuation areas over a two-dimensional grid of points with a predefined resolution. Considering that, by construction, the global feasible area $\mathcal{G}_{fa}$ must be included inside the friction region $\mathcal{Y}_f$ (green), the grid of points was only generated inside $\mathcal{Y}_f$.

The red dots correspond to those CoM locations that do not belong to their own local feasible region. The green grid points correspond to those CoM locations that instead do belong to their own local feasible region. You can see that, if the grid point is green (*i.e.,* it belongs to his own local region) we then intersect the local actuation region with a small rectangle of the same resolution of the grid; in this way we are able to enforce the fact that the local actuation region is only valid in a small neighborhood of the corresponding grid point. The global feasible region (dark green line) $\mathcal{G}_{fa}$ is then obtained as the convex hull of all the small local blue regions. You can see that some grid points which result to be inside $\mathcal{G}_{fa}$ they are actually red and their partition is green because they resulted to be locally unfeasible; this was probably due to numerical artifacts.

The explained strategy yields the desired global actuation region $\mathcal{G}_{fa}$, however, the computation is not efficient (a new IP problem has to be solved at every grid point) and its accuracy depends on the predefined resolution of the grid map. Dr. Stéphane Caron[3], as a part of our ongoing collaboration and with the perfective of a future joint article, proposes a more efficient strategy that is based on Voronoi tassellations and that is shown in Fig. 5-8b. Notice, therefore, that the authorship of this particular strategy belongs to him and it is being included in this dissertation prior to publication but with his permission.
The Voronoi partitions are generated starting from a uniform distribution of n sample points in a convex set (in this case inside $\mathcal{Y}_f$). It was proven that using uniform distributions for the generation of Voronoi diagrams minimized the variance of the areas of the individual Voronoi cells. The usage of uniformly sampled points in $\mathcal{Y}_f$ (green area in Fig. 5-8b) and its partition by means of a Voronoi diagram allows us to compute the same area $\mathcal{G}_{fa}$ as in the case in which a grid map was used. In this case, however, the number of samples was considerably reduced and, as a consequence, also the computation cost significantly decreased.

5-3-4 Sequential Iterative Projection (SIP) Algorithm

In this Section we present an alternative with respect to the grid-based or Voronoi-based evaluation, presented in the previous Section, to compute the global actuation region. This consists of a recursive solution of the Alg. 2 presented in Section 5-3-1. The algorithm is based on the observation that, if the same set of contacts is kept, varying the robot configurations will result in one of the two following events:

1. $\mathcal{Y}_{fa}$ degenerates to a null area (either because the Jacobian matrix $\mathbf{J}_i$, used for the computation of $\mathbf{C}$ in Eq. 5-9, becomes singular or because the torque limits are exceeded);
2. $\mathcal{Y}_{fa}$ has a positive area but the projection of the CoM is not inside it (as mentioned above).

Case 1) occurs most commonly when the robot mass increases beyond the ratio allowed by the torque limits. If, instead, the robot mass is constant (no external disturbances or no external loads) throughout the task then the event 2) usually occurs before event 1).
Based on this observation, starting from a default CoM position we update the configuration along a desired direction, represented by the unit vector $\mathbf{a} \in \mathbb{R}^2$, and sequentially recompute

[3]LIRMM, CNRS-University of Montpellier, email: stephane.caron@lirmm.fr

the instantaneous actuation region (using Alg. 2) until the distance between the edges of $\mathcal{Y}_{fa}$ and the CoM projection becomes smaller than the predefined acceptable tolerance ϵ (see Alg. 3).

```
Input: $\mathbf{y}, \mathbf{p}_1, \dots \mathbf{p}_{n_c}, \mathbf{n}_1, \dots \mathbf{n}_{n_c}, \mu_1, \dots, \mu_{n_c}, \mathbf{g}_1, \dots \mathbf{g}_{n_c}, \mathbf{J}_1, \dots \mathbf{J}_{n_c}, \boldsymbol{\tau}_1^{lim}, \dots \boldsymbol{\tau}_{n_c}^{lim}$;
Result: vertex $\hat{\mathbf{c}}$ of the global feasible region $\mathcal{G}_{fa}$ along the direction $\mathbf{a} \in \mathbb{R}^2$
Initialization: set the initial vertex guess $\hat{\mathbf{c}} \in \mathbb{R}^2$ equal to the default CoM $(x, y)$ coordinates:
  $\hat{\mathbf{c}}_0 = P\mathbf{c}$;
Set $d = \infty$ and $\epsilon \to 0$;
while $d > \epsilon$ do
    I) compute the contact points $\mathbf{p}_1^{\hat{\mathbf{c}}}, \dots, \mathbf{p}_{n_c}^{\hat{\mathbf{c}}}$ in the new CoM frame located in $\hat{\mathbf{c}}_i$;
    II) solve inverse kinematics: $\mathbf{q} = IK(\mathbf{p}_1^{\hat{\mathbf{c}}}, \dots \mathbf{p}_{n_c}^{\hat{\mathbf{c}}})$;
    III) compute $\mathbf{C}$ and $\mathbf{d}$ as in Eq. 5-9 for the joint configuration $\mathbf{q}$;
    IV) $\mathcal{Y}_{fa} = IP(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \mathbf{u}, \mathbf{C}, \mathbf{d})$;
    V) find the intersection $\mathbf{e} \in \mathbb{R}^2$ between the desired direction $\mathbf{a}$ and the edges of $\mathcal{Y}_{fa}$;
    VI) $d = ||\mathbf{e} - \hat{\mathbf{c}}||_2$;
    VII) update the vertex $\hat{\mathbf{c}}$ towards $\mathbf{e}$: $\hat{\mathbf{c}}_{k+1} = \hat{\mathbf{c}}_k + \alpha(\mathbf{e} - \hat{\mathbf{c}}_k)$
end
```

Algorithm 3: Sequential Iterative Projection (SIP) Algorithm

Differently from the tests reported in Fig. 5-7 (where all the CoM were homogeneously distributed along one segment), in this case we increasingly update the new tested CoM position $\hat{\mathbf{c}}$ of a given gain α of the distance d between the current value and the intersection $\mathbf{e}$ between the local region $\mathcal{Y}_{fa}$ and the considered search direction:

$$d = ||\mathbf{e} - \hat{\mathbf{c}}||_2 \tag{5-19}$$

With the usage of a suitable gain α, this allows the distance d to recursively converge to zero; whenever the distance becomes lower than the predefined tolerance ϵ the procedure is stopped and the latest CoM position $\hat{\mathbf{c}}$ is considered to be a vertex of the global feasible region $\mathcal{G}_{fa}$.

This strategy allows to efficiently find a vertex on the edge of the global feasible region by recursive computations of the IP algorithm. The strategy can then be repeated in multiple directions (see for example Fig. 5-9b) in order to reconstruct the entire global feasible region $\mathcal{G}_{fa}$ or just a part of it in the region of most interest.

The SIP algorithm can also be considered as a sequential linearization algorithm that recursively estimates the robustness of the considered CoM along a specific motion direction. The SIP could be used, for instance, to estimate how far ahead in a specific direction the CoM may move without violating neither friction nor actuation constraints.
Alternatively a variant of the SIP can be considered where, rather than testing different CoM positions and/or trunk orientations, different foot positions are tested. This might be interesting for example for foothold planning applications.

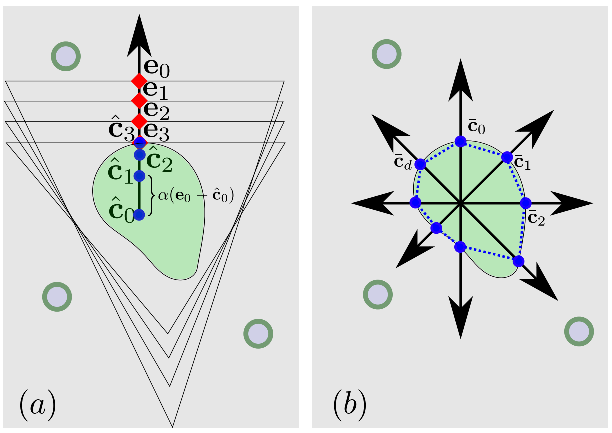


Figure 5-9: (a) depiction of the SIP algorithm. We can see that the distance $d_i = ||\mathbf{e}_i - \hat{\mathbf{c}}_i||_2$ converges to zero if the gain α is properly set; (b) by repeating the SIP algorithm in different directions we can build an estimate of the actuation region.

The underlying idea of the SIP algorithm (*i.e.*, to sequentially change one quantity (CoM position, trunk orientation or feet positions) and to-recompute a new local area till when the convergence criterion is met) remains therefore valid.

As briefly mentioned above, the local actuation areas computation requires the knowledge of the robot limits and configuration. For this reason, unlike the classical friction regions, the actuation-consistent regions may vary depending on the robot height or robot orientation. In the next Section I will attempt to exploit this property by estimating a 3D volume of friction- and actuation-consistent CoM positions.

5-4 3D Feasible Volume

Fig. 5-10 shows a simulation of the JVRC-1 robot climbing a vertical ladder.
In this case, we model the robot with unilateral contact constraints (intersection of friction cones and force polytopes) at the feet and bilateral contacts (only force polytopes and no friction cone constraints) at the hands. We then compute the friction region $\mathcal{Y}_f$ using Alg. 1 setting infinite values of the friction coefficient on the hand contacts. This yields as a result the whole horizontal plane as friction region. This is due to the fact that, because of the opposition of the contacts on the vertical ladder (like in the chimney example of Fig 4-2), the robot is able to exert any contact force, so it could ideally locate its CoM anywhere in the x, y plane (if kinematic limits are not considered).

The blue convex set in Fig. 5-10a is the global feasible region $\mathcal{G}_{fa}$. The blue 3D volume in Fig. 5-10b is the convex hull of multiple $\mathcal{G}_{fa}$ computed at different robot heights. $\mathcal{G}_{fa}$ can therefore be seen as a slice of the blue 3D volume in the direction orthogonal to gravity.

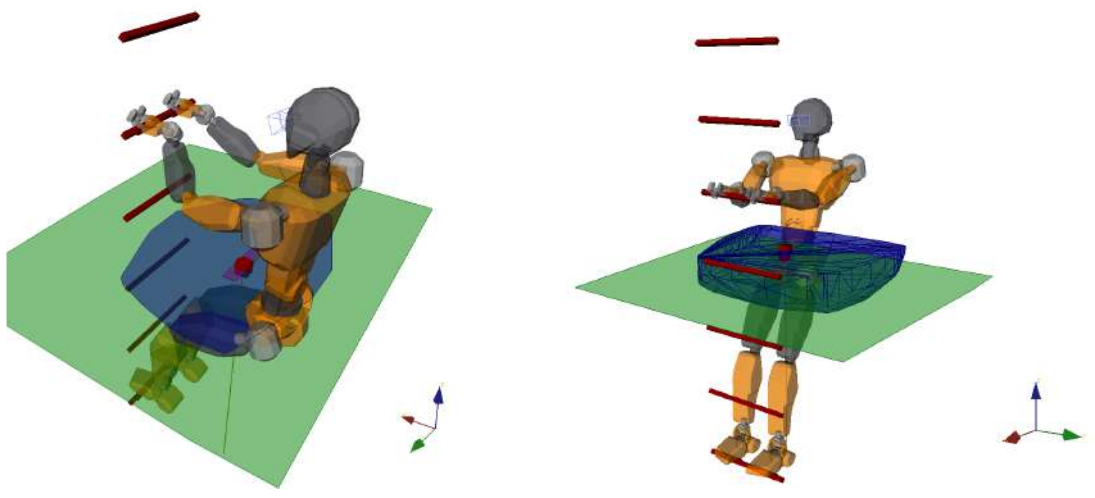

(a) Friction region (green) and global feasible region (blue) for a specific robot height. **(b)** Friction region (green) and global feasible volume (blue)

Figure 5-10: JVRC-1 humanoid robot climbing a ladder. This Figure has been produced by Dr. Stéphane Caron (LIRMM, CNRS, France) who created it in perspective of a future joint publication and he agreed with the display of this Figure in this dissertation.

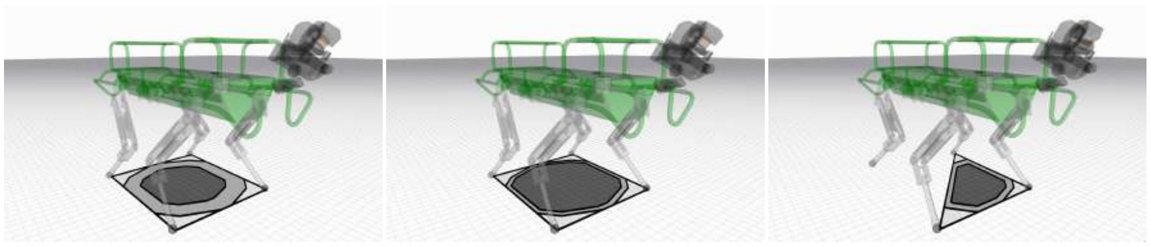
In the ladder climbing scenario, the robot-specific kinematics highly affect the climbing capabilities of the robot. This can be captured by the upper and lower bounds of the 3D feasible volume which, depending on the values of the torque limits may be due to either of two following causes:

1. the area of the global actuation region $\mathcal{G}_{fa}$ converging to zero;
2. the arms and legs Jacobians reaching their kinematic singularities (*i.e.,* point 1 in Section 5-3-4).

The usage of 3D feasible volumes, such as the one shown in this picture, could overcome the typical limitation of static equilibrium approaches for which the CoM height c_z is unobservable (because parallel to gravity). Such 3D volumes could indeed enable the planning of friction- and actuation-consistent robot height trajectories besides the planning of the coordinates orthogonal to gravity.

5-5 Center of Mass Trajectory Optimization

The distance m between the CoM projection $\mathbf{c}_{xy} = \mathbf{P}\mathbf{c}$ and the edges of the instantaneous actuation region can be seen as a static instantaneous measure (or margin) of how far the robot is from hitting one of the torque limits. In this Section we briefly present an online CoM



(a) scaling factor = 0.7 **(b)** scaling factor = 0.9 **(c)** scaling factor = 0.8

Figure 5-11: Local feasible regions:

(a) Classical friction region (transparent);

(b) Local feasible region $\mathcal{Y}_{fa}$ (gray);

(c) Scaled local feasible region $\hat{\mathcal{Y}}_{fa}$ (dark gray) for increased robustness.

trajectory planning strategy that employs the local feasible region $\mathcal{Y}_{fa}$ as its main criterion for balancing ensuring actuation-consistency on rough terrains. The results of this strategy can be seen in the accompanying video[4].
As we only deal with CoM planning in this Section, we will assume the gait sequence, phase timings and step locations to be predefined. As a further assumption, we only consider quasi static gaits where the horizontal acceleration of the base link is negligible at all time. An extension to the dynamic case with non-negligible CoM horizontal accelerations may be included in the future works.
As I consider the quadruped robot HyQ as the main hardware platform for my experiments, I will consider here a static quadrupedal gait called crawl, where only one foot at the time is allowed to be in swing phase while all the other feet have to be in stance. This condition enforces at least three point feet to be always on the ground: thanks to this condition the base link is fully actuated and static stability can always be achieved if an appropriate plan is computed.

The heuristic strategy that we propose is that during the triple stance phases (those which are most critically for stability and feasibility compared to when the robot has four feet on the ground) the CoM projection $\mathbf{c}_{xy}$ is planned to lie on the edge of a conservative local feasibility region $\hat{\mathcal{Y}}_{fa}$. The region $\hat{\mathcal{Y}}_{fa}$ is obtained by scaling the local feasible region of a predefined percentage (typically 0.9 of scaling factor). The scaling procedure can be defined as an affine transformation that preserves straight lines and parallelism relationships among the edges of the feasible region. For the transformation to preserve the angles, it should be performed as a scaling of the vertices of the polygon with respect to the Chebyshev center (*i.e.,* the center of the largest ball inscribed in the feasible region). This can be computed as an LP as long as the $\mathcal{V}$-representation of the local feasible region is given. As an alternative approach

[4]https://www.dropbox.com/s/g1x6gwztghm0qwe/actuation_region_vid.mp4?dl=0

one may scale the local feasible region vertices with respect to their centroid, which can be found analytically as the average of all the vertices. The centroid can be considered a good approximation of the Chebychev center whenever the feasible region presents good symmetry properties. Whenever the feasible region is not symmetric, the centroid might considerably differ from the Chebychev center thus resulting in a value of the robustness margin m lower than desired. Fig. 5-11 shows three scaling examples of the local feasible region $\hat{\mathcal{Y}}_{fa}$ by three different scaling factors $(0.7, 0.9$ and $0.8)$.

As previously mentioned, notice that the actuation margin defined here represents a method to verify whether there exists a set of joint torques that can withstand the robot's weight for the considered configuration. In practice, however, depending on the implementation of the whole-body controller, one of the torque limits might be reached even when the margin m is still positive [2]. This is because, for example, the whole body controller might not explicitly enforce the joint torque constraints[5] [28].
The development of an actuation-consistent whole-body controller is out of the scope of this dissertation, however, suitable implementations of such controllers can be found, by instance, in [141, 117].

5-6 Foothold Planning

The foothold planning strategy that we present in this Section represents a sample-based strategy to improve the navigation capabilities of the HyQ quadruped robot on rough terrains. We attempt to exploit the computational efficiency of the IP algorithm as in Alg. 2 in order to plan foothold locations that ensure the robot's stability and actuation consistency while traversing rough terrains.
As in the previous Section we assume here a static crawling gait where only one foot can break the contact from the ground at the time. We also consider here predefined phase durations and we only attempt to determine the most suitable foothold for the next swing leg. Our strategy consists in sampling a set of candidate footsteps evaluating the height map in a neighborhood of the default step location. The default step location is a simply a function of the user-defined desired linear and angular velocities of the robot and it neither considers the external map of the surrounding environment nor the stability and actuation consistency requirements [11]. We therefore select these p candidate footholds from the map, thus keeping the predefined (x, y) step locations but updating the corresponding z coordinate and normal orientation from the surface perceived by the vision module [142, 143].

[5]The whole-body controller might be equally minimizing all the joint torques rather than explicitly constraining them inside the torque limits or the employed weight matrix might penalize the torque on a specific predefined joint (*e.g.,* the knee torque) that is not the one being closest to the torque limits violation for the current configuration.

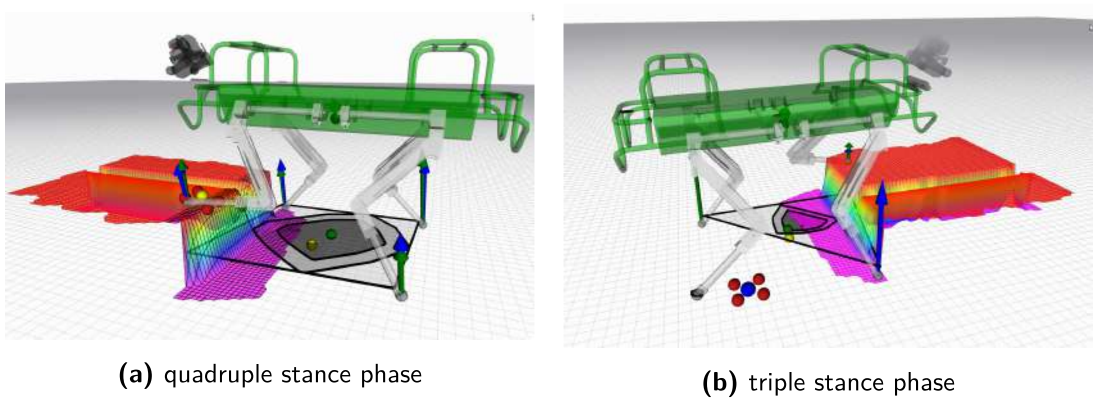

(a) quadruple stance phase **(b)** triple stance phase

Figure 5-12: Vision based foothold planning: the five red balls represent five candidate footholds for the current swing leg. The selected foothold is drawn with a yellow ball and a blue ball. The light gray and the dark gray regions represent, respectively, the feasible region and the scaled feasible region.

The next step consists in computing the local feasibility region $\mathcal{Y}_{fa}^i$ for all the p considered foot locations ($i = 1, \dots, p$) and for the set of feet that will be in stance during the following swing phase. For this heuristics to yield reliable results the local feasible regions must also be computed considering the future position of the CoM in the next triple stance phase. We then exploit the knowledge of the (decoupled) CoM planning policy to plan foothold locations that are coherent with the future reference CoM trajectory enforced by the CoM planner (described in the previous Section).
Fig. 5-12 shows a foothold planning simulation in which five different candidate footholds (red balls on the step) are considered. You can see in Fig. 5-12a that, when a map is available for the (x, y) candidate footholds positions, the corresponding z component is updated according to the value provided by the exteroceptive perception (height map). Fig. 5-12b shows how the local feasible region $\mathcal{Y}_{fa}$ and, consequently, the scaled version $\hat{\mathcal{Y}}_{fa}$ can take on small area when the robot is far from a default configuration (*e.g.,* when one leg is much more retracted than the other legs).

5-7 Summary

In this Chapter I have presented a novel approach for projecting joint-torque constraints from the high dimensional joint space ($n + 6$ DoFs) to the two-dimensional subspace of the task-space orthogonal to gravity (Section 5-3). Despite the static assumption and the local nature of the resulting 2D friction- and actuation-consistent region, this strategy promises to improve the robustness of legged robots against the violation of joint torque limits. Thanks to the computational efficiency of the local feasible region $\mathcal{Y}_{fa}$ estimation, actuation-consistency and *robustness* can be tested online at a minimum of $50Hz$ rate and without any approximation regarding the location and orientation of the contacts. This last point allows are approach to be embedded in map-based foothold optimization strategy that plans feasible footholds on the height map provided by the vision module (see, for example, Fig. 5-12).

In Sections 5-3-3 and 5-4 we also defined the 2D *global feasible region* $\mathcal{G}_{fa}$ and its 3D counterpart. While being less computationally efficient than the local actuation region, these can be seen as useful and intuitive tools for the understanding of the dynamic properties of legged robots and their locomotion capabilities. Example applications for such quantities involve, for example, the selection of a feasible robot height on a multi-contact scenario, the CoM planning for a ladder climbing task or the optimization of the robot pose in order to step on a very high obstacle. The spectrum of possible applications of the 2D actuation-aware regions can be extended also to robotic grasping tasks for the assessment of the grasp feasibility and to industrial manipulators for an intuitive indicator of the dependency between maximal payload and robot configuration.

in the following list I will sum up the main concepts proposed in this Chapter:

1. the 2D *local* actuation region $\mathcal{Y}_a$ and the 2D *local* feasible region $\mathcal{Y}_{fa}$ represent the two-dimensional counterparts of the 6D AWP and FWP under the assumption of static stability (only gravity acting on the robot's CoM);

2. the local actuation region $\mathcal{Y}_a$, depending on the robot's configuration and actuation capabilities, may be larger than the friction region $\mathcal{Y}_f$ (*i.e.,* what is usually known with the name of support region). This is because $\mathcal{Y}_a$ does not consider the friction limits and, therefore, represents the set of all those locations where the robot may place its CoM if it was also able to exert negative normal contact forces (pull the ground instead of just pushing). The local feasible $\mathcal{Y}_{fa}$, instead, does consider the friction limits and, therefore, is inscribed inside the friction region $\mathcal{Y}_f$ by construction;

3. $\mathcal{Y}_a$ and $\mathcal{Y}_{fa}$ can be computed at interactive rates by means of the Iterative Projection (IP) algorithm. This is an alternative approach to the option of computing, respectively, the AWP and FWP first and performing a standard projection on these objects later (see Fig 5-13). This opens the doors to the possibility of devising new motion planners that are capable of evaluating online the actuation consistency besides the locomotion stability on any arbitrary environment (*e.g.,* no simplifying assumption is taken on the robot's morphology or on the structure of the terrain);

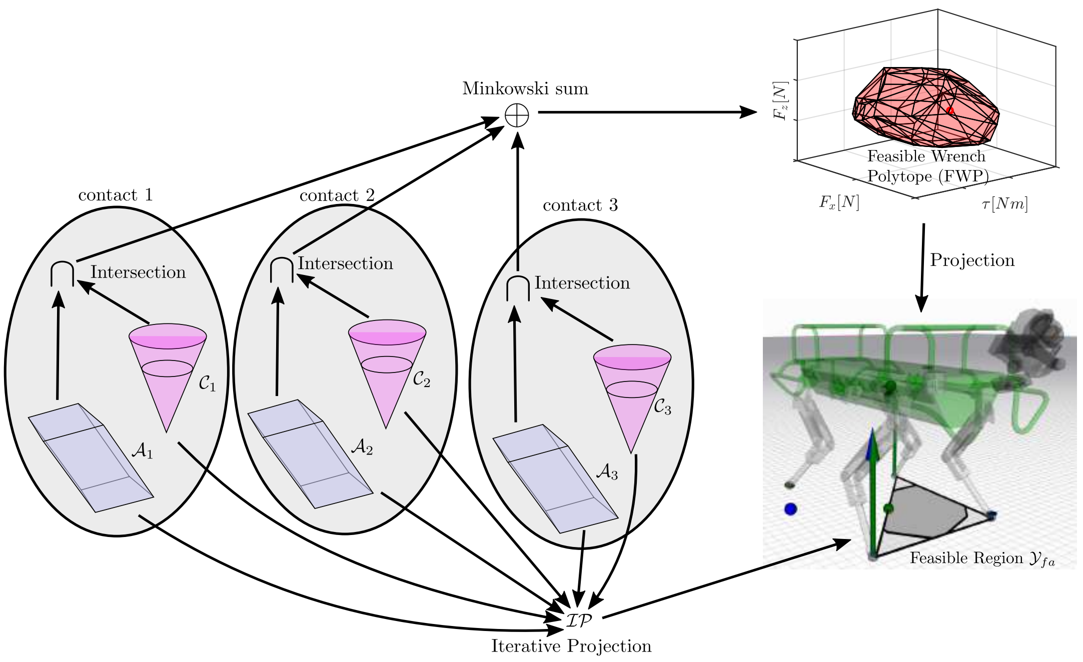


Figure 5-13: Graphic representation of the possible ways to compute the feasible region. One approach consists in performing a standard projection of the FWP while the other method consists in performing the Iterative Projection (as proposed in this Chapter) directly from the friction cones $\mathcal{C}_i$ and the force polytope constraints $\mathcal{A}_i$.

4. as an example of usage of the local feasible region in legged locomotion, we formulate a sample-based online motion planner capable of concurrently optimizing feasible CoM trajectories and foothold locations online. The proposed planner employs the map provided by the onboard perception to select the optimal footholds and it can react to possible external disturbances and/or commands given by the robot's operator;

5. the global actuation region $\mathcal{G}_a$ and the global feasible region $\mathcal{G}_{fa}$, unlike their local counterparts, represent configuration independent quantities and only depend of the position of the contacts. Different ways are proposed in this Chapter to compute these global regions, all of which base on sequential recursion of the IP algorithm. The computation of the $\mathcal{G}_a$ and $\mathcal{G}_{fa}$ can not be achieved online with satisfactory accuracies and, as a consequence, is proposed in this thesis a tool for offline motion planning optimization and for mechanical design optimization.

Chapter 6

Discussion and Conclusions

This Chapter first recalls the main contributions and final conclusions that have already been mentioned in the summary of Chapters 3, 4 and 5. It then continues with a brief discussion of the main take-home messages and drafts some possible future development of the concepts presented in this dissertation.

6-1 Conclusions

1. Chapter 3 introduces the first contribution presented in this dissertation, namely a strategy for the generation of dynamically stable omni-directional bounding on quadruped robots. An offline optimization problem is employed for the synthesis of a baseline stable bounding gait in place where the stability of the simplified $2D$ planar sagittal model of the robot is obtained by *limit cycles analysis*. The result of the optimization is then adapted *online* to the user-defined desired linear and angular velocities or to compensate possible external disturbances by means of heuristic strategies such as the tuning of the contact force impulses or the reactive adjustment of the target footholds using a stability criterion based on the Center of Pressure (CoP).

2. Chapter 4 presented a substantial change of approach with respect to the previous Chapter where the heuristic strategy only valid for bounding gait on moderately rough terrains is exchanged for an optimization-based approach applicable to environments of complex geometry. In this change of perspective key elements of legged locomotion such as friction cones and joint torque limits take on a role of primary importance. The contribution of this Chapter, in particular, consists in a method to project the joint

torque limits into the $6D$ space of the centroidal wrench. The resulting set of $6D$ linear constraints, called Feasible Wrench Polytope (FWP), can be used to devise actuation-consistent motion planners that do not explicitly optimize neither the joint torques, not the contact forces.

3. Chapter 5, in continuity with Chapter 4, has focused on the feasibility constraints that, especially in presence of complex environments, most commonly hamper the successful hardware completion of the locomotion task. In this Chapter, in particular, I attempted to formulate two intuitive indicators of the locomotion capabilities of legged robots, which I defined under the name of *local* and *global feasible regions*. These polygons are indeed defined in the more easily portrayable 2D space compared to 6D wrench space in which the FWP was defined in Chapter 4. This increased intuitiveness and computational efficiency came, however, with an assumption of static stability.

6-2 Discussion

In the realm of online motion planning, simplified dynamic models, or templates [39], are widely used and their inability to consider friction limits, actuation limits and kinematic joint limits has been traded off in exchange of their great descriptive power when it comes to balance and keep stability against gravity. However, when the complexity of the terrain to be negotiated increases, such constraints cannot be omitted anymore and the use of simplified dynamic models has to be put aside in favor of more complex preview models such as the ones based on the full dynamic models (see Section 2-1-2).
In my dissertation (Chapters 4 and 5 in particular) I attempted to find a suitable trade-off between the accuracy of whole-body models and the simplicity of reduced models, a trade-off that could fit well to the problem of legged locomotion on rough terrains and that could result in an increase of robustness when performing hardware experiments on complex geometry environments. My method for achieving this goal consisted in projecting the joint-torque limits into lower dimensional spaces such as the $6D$ centroidal wrench space (Chapter 4) or the $2D$ space of Center of Mass (CoM) positions (Chapter 5) orthogonal to gravity.

The underlying cornerstones of the approach developed throughout this dissertation for the generation of legged locomotion on rough terrains can be summarized in the following list:

1. *interactive* motion planners (*i.e.,* online, or even real-time planners) play a key role in the compensation of the constantly present modeling errors and external disturbances. Suboptimal optimization strategies based on simplified dynamic models can often show superior performances with respect to highly nonlinear formulations based on accurate descriptions of the robot. The strategies presented in this dissertation, in particular the

local feasible regions described in Chapter 5, fit well to this requirement and may result in a considerable increase of the system robustness when negotiating rough terrains;

2. Feasibility constraints in highly dimensional spaces, such as the joint-space torque limits, can be conveniently projected into lower dimensional spaces, *e.g.,* the 6D space of feasible centroidal wrenches or the 2D space (orthogonal to gravity) of feasible CoM positions. Such projections may lose important details about the system dynamics such as the possibility to evaluate the absolute magnitude of the feasibility margin because of the non-homogeneous units (see, for example, the dimensionless feasibility factor in Section 4-4-2), however, they can still be used to assess the feasibility of a given motion plan;

3. In the motion planning setting, the possibility of projecting joint-space torque limits into the centroidal space - or a subset of it - opens the doors to the synthesis of feasible actuation-consistent motion plans without the need of explicitly optimizing neither the joint torques nor the contact forces. Indeed, if the trajectory of the gravito-inertial wrench $\mathbf{w}_{GI}$ is within the limits of the FWP (as in Chapter 4) or if the CoM trajectory is within the limits of the feasible region $\mathcal{Y}_{fa}$ (as in Chapter 5), there is then no further need to explicitly plan the trajectory of the contact forces nor the trajectory of the joint torques. This consideration may substantially reduce the computational burden on the whole-body controller that is no more required to verify the feasibility of the reference trajectories provided by the motion planner and can, therefore, focus on the tracking of the desired trajectories with no need of preview.
   This reduced burden on the motion controller does not, apparently, affect the complexity of the online motion planner. As a matter of fact, the trajectory optimization problem does not require a large number of optimization variables since the contact forces and the joint torques are implicitly embedded in the constraints of the centroidal space variables (*e.g.,* CoM wrenches $\mathbf{w}_{GI}$ in the case of the FWP or CoM horizontal positions $\mathbf{c}_{xy}$ in the case of the feasible region $\mathcal{Y}_{fa}$). These implicit constraints can, however, be highly nonlinear and may, therefore, lead to local minima and to an increased sensitivity on the initial guess of the nonlinear solver;

4. A straightforward approach to the problem of projecting the joint-actuation limits could have been the option to simply clamp the friction cones with a fixed maximal value of the component of the contact forces normal to the ground (rather than intersecting them with the force polytopes) [144]. However, the restrictive consequences of this choice in motion planning scenario would be twofold:

   (a) on one side, it would be hard to select a realistic value of the normal force upper bound without excessively constraining the problem;

(b) on the other side, the information regarding the relationship between the maximal contact forces and the limb configuration would be neglected. In Fig. 4-10 you can see that maximal normal and tangential forces can only be achieved on the boundaries of the legs workspace. Besides that, a maximal normal contact force can only be achieved if coupled with a predefined tangential force (*i.e.,* whose value will correspond to a vertex or an edge of the actuation polytope $\mathcal{A}_i$). A constant normal force value would lose this information and it would mistakenly consider any tangential force component inside the friction cone to be feasible. Considering a constant maximal contact force would, therefore, allow to generate conservative motion plans but it would prevent the robot from optimizing its own body posture and contacts distribution.

6-3 Future Works

Among the future works I would like to mention the possibility of embedding into a simplified model more feasibility constraints such as the kinematic joint limits and the occupancy of the robot which may cause possible self-collisions or collisions with the environment. These constraints are usually roughly over-approximated by box constraints in the task space and a large room for improvement seems to be available. I am indeed convinced that preserving and understanding the relationship between the high dimensional joint-space and the lower dimensional task-space is a key aspect for the generation of agile and dexterous locomotion. Examples of similar approaches attempting to build *proxy* constraints between joint-space feasibility constraints (joint kinematic limits, self-collisions, *etc.*) and task space can already be found in the literature [51, 53]. The same problem can be tackled from a data-driven perspective, attempting to learn from experience the set of viable states that a robot can reach or trying to build an internal map of the

Besides the improvement of motion planners algorithmic efficiency that I dealt with in this dissertation, there is, in my opinion, also a remarkable need in the literature of improved numerical methods specifically tailored for legged locomotion. Considerable advancements have been achieved in recent years, for example, to find efficient approximations to numerical problems such as the bilinearity of the angular momentum and of the phase timings [83, 145]. There still exist, however, a number of numerical challenges, such the complementarity of contact forces and feet accelerations or planning taking into consideration the shape of a complex (non-convex) environment, that still prevent the existing motion planners for legged robots to reach the desired flexibility levels.
Therefore, the future works are in the prospect of reliably implementing on the robotic hardware motion planners for legged robots able to jointly optimize online (*i.e., interactively*) CoM trajectories, feet trajectories, feet contact forces, base angular dynamics and phase durations

on arbitrary terrains.

Appendix A

Differential Geometry and Floating Base Systems

A-1 Singularity of the Orientation Manifold

The $SO(3)$ orientation manifold is a mathematical group that does not admit a *global chart*, meaning that there exists no singularity-free minimal set of coordinates (*e.g.,* 3 coordinates in the case of $SO(3)$ and 2 coordinates in the case of $SO(2)$). This is a numerical issue which is due to the winding of the orientation coordinates: trigonometric functions are not bijective (there exist infinite arguments that can yield the same result) and present discontinuities which make these function not fully invertible. This limitation becomes apparent if we consider, for example, the matrix $B(\Theta)$ that maps the rates of the Euler angles $\Theta = [\alpha, \beta, \gamma]^T$ (a minimal set of coordinates) into the angular velocity $\boldsymbol{\omega}$:

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & \cos(\alpha) & \sin(\beta)\sin(\alpha) \\ 0 & \sin(\alpha) & -\sin(\beta)\cos(\alpha) \\ 1 & 0 & \cos(\beta) \end{bmatrix}}_{B(\Theta)} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} \tag{A-1}$$

We can see that $det(B(\Theta)) = sin(\beta)$ which proves the fact that the mapping $B(\Theta)$ is singular ($det(B(\Theta)) = 0$) for $\beta = 0 + 2k\pi$ with $k \in \mathbb{Z}$.
This issue becomes particularly dangerous whenever we need to compute the Euler angles Θ of a rigid body (*e.g.,* in the development of a simulator) by integration of the Euler rates $\dot{\Theta}$, computation that requires the inversion of $B(\Theta)$ as a preliminary operation in order to obtain

$\dot{\Theta}$ from $\boldsymbol{\omega}$:

$$\dot{\Theta} = B(\Theta)^{-1}\boldsymbol{\omega} \tag{A-2}$$

In order to overcome the numerical singularities of orientation manifold one could consider to switch between two different sets of minimal coordinates that are defined with respect to two different conventions. In this way it is possible to switch from one convention to the other whenever the currently considered one gets close to a singularity.
The second option is to use a non-minimal set of coordinates (*e.g.,* quaternions) that allows to integrate the equations of motion with one unique orientation convention.

Let us consider, for example, the generic dynamic equation of motion of fixed base industrial manipulators which is commonly obtained by solving the Langrange Equation (see Eq. 2-13) of the system:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau} \tag{A-3}$$

In the case of Eq. A-3 which refers to a fixed base manipulator, none of the n actuated joints of the system belongs to $SO(3)$; as a consequence there is no singularity issue in the integration of $\dot{\mathbf{q}} \in \mathbb{R}^n$ and $\ddot{\mathbf{q}} \in \mathbb{R}^n$ and Eq. A-1 is, therefore, well-posed.
If we instead consider a floating base system we can then obtain its equation of motion using, rather than the Lagrange Equation, the Recursive Newton Euler Algorithm (RNEA) [23] that yields an equation of the following form:

$$\mathbf{H}(\mathbf{q})\dot{\mathbf{s}} + \mathbf{C}(\mathbf{q}, \mathbf{s}) = \boldsymbol{\tau} \tag{A-4}$$

where $\mathbf{s}$ is the generalized velocity vector which includes the base's pose $\boldsymbol{\nu} = [\dot{\mathbf{x}}^T, \boldsymbol{\omega}^T]^T$ and the joints velocity $\dot{\mathbf{q}} = [\dot{q}_1, \dots, \dot{q}_N] \in \mathbb{R}^6$:

$$\mathbf{s} = \begin{bmatrix} \boldsymbol{\nu} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{x}} \\ \boldsymbol{\omega} \\ \dot{\mathbf{q}} \end{bmatrix}. \tag{A-5}$$

Eq. A-4 is therefore well suited for floating base systems

A-2 Non-Holonomic Constraints

In the domain of robots' kinematics *holonomic* constraints are all those constraints that can be solely expressed in terms of the joints' position:

$$\mathbf{c}(\mathbf{r}, t) = 0 \tag{A-6}$$

where $\mathbf{r}$ is the generalized position and t is the time variable. A system is said to be holonomic if all its constraints are holonomic.

Non-holonomic constraints are instead those constraints that have to resort to the joints velocities and/or accelerations variables besides the joints position variables in order to be formally described. In other words, the non-holonomic constraints can only be expressed in the following form:

$$\bar{\mathbf{c}}(\mathbf{r}, \dot{\mathbf{r}}, t) = 0 \tag{A-7}$$

The correlation between *non-integrability* and constraints' *anholonomy* arises from the consideration that non-holonomic constraints cannot be integrated in order to be expressed as a function of the joints position only, otherwise they would simply be the derivative of a holonomic constraint.

Typical examples of non-holonomic constraints for wheeled vehicles are the contact constraints that enforce no lateral slippage of the wheels. In the wheeled robots' community, the vehicle's holonomy is often coupled with the omni-directionality of the platform itself: if the system is holonomic that means that a motion can be instantaneously achieved in any arbitrary direction regardless of the joints configuration. If the vehicle is non-holonomic, on the other hand, (*e.g.,* the unicycle model) then the robot can only move in the direction given by the wheels' orientation and, to achieve a lateral motion, it will first need to re-orient its wheels.
One consequence of the above considerations is the fact that the velocity of non-holonomic systems depends on the instantaneous configuration, leading to the conclusion that the number of independent position variables is higher than the number of independent velocity variables (at least one velocity varibles can be expressed as function of the configuration) [23, pag. 41].

In the domain of floating base legged robots, a typical example of non-holonomic constraint regards the equation that enforces the momentum conservation; this can be written by means of the Centroidal Momentum Matrix (CMM) $\mathbf{A}_G(\mathbf{q})$ in the following way:

$$\mathbf{h}_G = \mathbf{A}_G(\mathbf{q})\dot{\mathbf{q}} = 0 \tag{A-8}$$

while the first three rows of Eq. A-8, relative to the linear momentum, encode holonomic constraints, the last three rows regarding the angular momentum are instead non-holonomic [146, 147]. The non-holonomy of the angular momentum of floating base systems (see Wieber *et al.* [134]) is due to the non-*exactness* (and thus non-integrability) of the angular velocity of the floating base link $\boldsymbol{\omega}_0$ (please see [24, pag. 40] for an exhaustive explanation of the non-exactness of the angular velocity).

Appendix B

Computational Geometry

I will recall in this Appendix few of the main concepts and definitions connected to computational geometry that are heavily used in this dissertation. Most definitions are takes by the following sources [133, 136, 119, 19].

B-1 Generic Bounded and Unbounded Polyhedra Definitions

Main definitions and terminology used in sets representation and adopted in this dissertation:

- A convex **polyhedron** $\mathcal{H}$ is a subset of $\mathbb{R}^d$ that solves a finite set of m linear inequalities. The volume of a polyhedron can therefore be either bounded or unbounded. This is a generic definition that may include both (bounded) polytopes and (unbounded) polyhedral cones.

$$\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^d \quad | \quad \mathbf{A}\mathbf{x} \geq \mathbf{b}\} \tag{B-1}$$

  with $\mathbf{A} \in \mathbb{R}^{m \times d}$ and $\mathbf{b} \in \mathbb{R}^m$.

- A convex **polytope** $\mathcal{P}$ is a subset of $\mathbb{R}^d$ that solves a finite set of m linear inequalities and is bounded.

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^d \quad | \quad \mathbf{A}\mathbf{x} \geq \mathbf{b}\} \tag{B-2}$$

  with $\mathbf{A} \in \mathbb{R}^{m \times d}$ and $\mathbf{b} \in \mathbb{R}^m$.

- A convex **polygon** $\mathcal{P}$ is a polytope in dimension $d = 2$:

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^2 \quad | \quad \mathbf{A}\mathbf{x} \geq \mathbf{b}\} \tag{B-3}$$

  with $\mathbf{A} \in \mathbb{R}^{m \times 2}$ and $\mathbf{b} \in \mathbb{R}^m$.

- A convex **zonotope** $\mathcal{Z}$ is a special kind of polytope in $\mathbb{R}^d$ that presents particular similarity with respect to the its center [148, 149]. A zonotope can therefore be fully described by its center $\mathbf{c} \in \mathbb{R}^d$ and its p generators $\mathbf{g} \in \mathbb{R}^d$.

$$\mathcal{Z} = \left\{ \mathbf{c} + \sum_{i=1}^{p} \alpha_i \mathbf{g}_i \quad | \quad \alpha_i \in [-1, 1], \quad \mathbf{g}_i \in \mathbb{R}^d, \quad \mathbf{c} \in \mathbb{R}^d \right\} \tag{B-4}$$

- A convex **polyhedral cone** $\mathcal{C}$ is a subset of $\mathbb{R}^d$ that solves a finite set of m linear inequalities. Geometrically, each linear inequality defines a hyperplane that has to pass through the origin.

$$\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^d \quad | \quad \mathbf{C}\mathbf{x} \geq \mathbf{0}\} \tag{B-5}$$

  with $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{0} \in \mathbb{R}^m$ is a null vector.

Convex polyhedra, polytopes, zonotopes and cones are called d-polyhedra (d-polytopes, d-zonotopes or d-cones) if they have a non-zero interior in $\mathbb{R}^d$; d-polyhedra will have a null interior in all dimensions larger than d (in which case they are nicknamed *flat* polyhedra/polytopes/cones).
In this dissertation we will drop the adjective convex referred to these polyhedra for compactness. All the polyhedra mentioned in this dissertations are therefore to be considered to be convex unless explicitly specified.

In the computational geometry terminology, a hyperplane h of $\mathbb{R}^d$ is a supporting hyperplane of the polyhedron $\mathcal{H}$ if one of the closed halfspaces of h contains $\mathcal{H}$. A *face* $\mathcal{F}$ of $\mathcal{H}$ is a generic term to indicate either an empty set, $\mathcal{H}$ itself or the intersection between $\mathcal{H}$ and a supporting hyperplane. The *faces* of dimension $0, 1, d-1$ and $d-2$ are usually named *vertices, edges, ridges* or *facets* [133].

- A **half-space** is either of the two parts in which a hyperplane divides an affine space.
- A **generator** is a broad term to indicate all the elements of Euclidean space $\mathbb{R}^d$ that can be used to represent the considered set. Depending on the considered type of polyhedron, generators may include *vertices*, *rays* (or *edges*) and *intervals.*

According to the Minkowski-Weil theorem [119], polyhedra can be equivalently described in terms of their half-spaces ($\mathcal{H}$-description) or in terms of their generators ($\mathcal{G}$-, $\mathcal{V}$-, $\mathcal{R}$ or $\mathcal{I}$-description). Polytopes, for example, can be equivalently described in terms of $\mathcal{H}$- and/or $\mathcal{V}$-description. Polyhedral cones $\mathcal{C}$ can be equivalently described in terms of $\mathcal{H}$-description (see Eq. B-5) and/or $\mathcal{R}$-description:

$$\mathcal{C} = \left\{ \sum_{i=1}^{p} \alpha_i \mathbf{r}_i \quad | \quad \forall \alpha_i \geq 0, \quad \sum_{i=1}^{p} \alpha_i = 1, \quad \mathbf{r}_i \in \mathcal{R} \right\} \tag{B-6}$$

where p is the number of rays of the set of rays $\mathcal{R}$:

$$\mathcal{R} = \left\{ \mathbf{r}_1, \dots, \mathbf{r}_p \quad | \quad \mathbf{r}_i \in \mathbb{R}^d \right\} \tag{B-7}$$

A cone, however, can not be represented by $\mathcal{V}$-description as it only owns one vertex which is placed in the origin of the reference frame.

B-2 Minkowsky Sums and Convex Cones

In the following I will discuss the main properties of sum of sets and convex hull algorithm:

- Given two convex sets $\mathcal{A}$ and $\mathcal{B}$, their addition (called **Minkowski sum**), indicated by the operator $\oplus$, is defined as the sum of the all elements of $\mathcal{A}$ with all the elements of $\mathcal{B}$:
$$\mathcal{A} \oplus \mathcal{B} = \{\mathbf{a} + \mathbf{b} \quad | \quad \mathbf{a} \in \mathcal{A}, \quad \mathbf{b} \in \mathcal{B}\} \tag{B-8}$$
which presents a $\mathcal{O}(a \cdot b)$ time (where a is the cardinality of $\mathcal{A}$ and b is the cardinality of $\mathcal{B}$).

- For a given convex set $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \dots \mathbf{s}_n | \mathbf{s} \in \mathbb{R}^d\}$ composed of n finite elements of dimension d, their *convex hull* is defined as the set of all the convex combinations of all its elements:
$$ConvHull(S) = \left\{ \sum_{i=1}^{n} \alpha_i \mathbf{s}_i \quad | \quad \forall \alpha_i \geq 0, \quad \sum_{i=1}^{n} \alpha_i = 1 \right\} \tag{B-9}$$

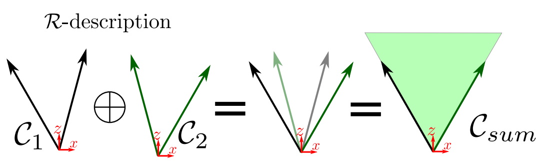
Minkowski sum and convex hull operators can be commuted, meaning that the following property holds:

$$ConvHull(\mathcal{A} \oplus \mathcal{B}) = ConvHull(\mathcal{A}) \oplus ConvHull(\mathcal{B}) \tag{B-10}$$

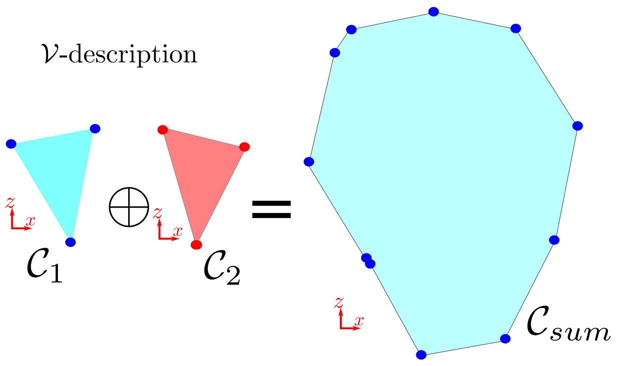
The worst-case output complexity of the problem is $\mathcal{O}(n^{[d/2]})$ although there exist sophisticated algorithms that can compute the convex hull in time $\mathcal{O}(n \log(h))$ where h is the number of points of the resulting convex hull.

For the computation of many locomotion related geometrical objects, such as the Contact Wrench Cone (CWC), it is important to notice that, given the $\mathcal{R}$-representation of two polyhedral cones $\mathcal{C}_1$ and $\mathcal{C}_2$:

$$\begin{aligned} \mathcal{C}_1 &= \left\{ \sum_{i=1}^{p_1} \alpha_i \mathbf{r}_{1,i} \quad | \quad \forall \alpha_i \geq 0, \quad \sum_{i=1}^{p_1} \alpha_i = 1, \quad \mathbf{r}_{1,i} \in \mathcal{R}_1 \right\} \\ \mathcal{C}_2 &= \left\{ \sum_{i=1}^{p_2} \alpha_i \mathbf{r}_{2,i} \quad | \quad \forall \alpha_i \geq 0, \quad \sum_{i=1}^{p_2} \alpha_i = 1, \quad \mathbf{r}_{2,i} \in \mathcal{R}_2 \right\} \end{aligned} \tag{B-11}$$



(a) The Minkowski sum of the $\mathcal{R}$-description of two cones corresponds to their convex hull.



(b) The Minkowski sum of two generic polytopes using their $\mathcal{V}$-description.

Figure B-1: Examples of Minkowski sums using (left) $\mathcal{R}$-description and (right) $\mathcal{V}$-description

the $\mathcal{R}$-representation of their Minkowski sum $\mathcal{C}_{sum}$ (*e.g.*, Fig. B-1a) can be obtained by stacking together (using the *union* operator $\cup$) the set of rays $\mathcal{R}_1$ and $\mathcal{R}_2$ of the two individual cones:

$$\mathcal{C}_{sum} = \mathcal{C}_1 \oplus \mathcal{C}_2 = \left\{ \sum_{i=1}^{p_1+p_2} \alpha_i \mathbf{r}_i \quad | \quad \forall \alpha_i \geq 0, \quad \sum_{i=1}^{p_1+p_2} \alpha_i = 1, \quad \mathbf{r}_i \in \mathcal{R}_1 \cup \mathcal{R}_2 \right\} \tag{B-12}$$

Despite yielding a redundant representation with internal rays, this property allows a considerable speed-up ($\mathcal{O}(p_1 + p_2)$ time) compared to the Minkowski sum of two convex bounded polytopes ($\mathcal{O}(p_1 \cdot p_2)$ time).

Besides the difference of computational performances, also the qualitative result provided by the Minkowski sum can significantly change depending on the type of representation ($\mathcal{G}$-, $\mathcal{H}$-, $\mathcal{V}$-, $\mathcal{R}$ or $\mathcal{I}$-representations) used for the description of the considered geometrical object. Figure B-1b shows, for example, the result of the Minkowski sum of two generic polytopes described by their $\mathcal{V}$-representation; it is possible to see that the qualitative result in this example differs significantly from the case in which the $\mathcal{R}$-representation is used (*e.g.*, Fig. B-1a).

Bibliography

[1] R. Orsolino, M. Focchi, D. G. Caldwell, and C. Semini, "A combined limit cycle - zero moment point based approach for omni-directional quadrupedal bounding," *20th International Conference on Climbing and Walking Robots (CLAWAR), Porto, Portugal*, 2017.

[2] R. Orsolino, M. Focchi, C. Mastalli, D. Hongkai, D. G. Caldwell, and C. Semini, "Application of wrench based feasibility analysis to the online trajectory optimization of legged robots," *IEEE Robotics and Automation Letters (RA-L)*, 2018.

[3] J. Buchli, F. Farshidian, A. Winkler, T. Sandy, and M. Giftthaler, "Optimal and Learning Control for Autonomous Robots," Eidgenössische Technische Hochschule (ETH), Zürich, Tech. Rep., 2017.

[4] M. H. Raibert, *Legged Robots That Balance.* Cambridge, MA, USA: Massachusetts Institute of Technology, 1986.

[5] M. H. Raibert, J. H. B. Brown, and M. Chepponis, "Experiments in balance with a 3d one-legged hopping machine," *The International Journal of Robotics Research*, vol. 3, no. 2, pp. 75–92, 1984. [Online]. Available: https://doi.org/10.1177/027836498400300207

[6] M. H. Raibert, M. Chepponis, and H. B. Brown, "Running on Four Legs As Though They Were One," *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 2, pp. 70 – 82, June 1986.

[7] J. Pratt, P. Dilworth, and G. A. Pratt, "Virtual model control of a bipedal walking robot," *IEEE International Conference on Robotics and Automation (ICRA)*, 1997.

[8] J. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. Pratt, “Virtual model control: An intuitive approach for bipedal locomotion,” *The International Journal of Robotics Research*, vol. 20, no. 2, pp. 129–143, 2001. [Online]. Available: https://doi.org/10.1177/02783640122067309

[9] O. Khatib, “A unified approach for motion and force control of robot manipulators: The operational space formulation,” *IEEE Journal of Robotics and Automation*, 1987.

[10] H.-W. Park, M. Yee, M. Chuah, and S. Kim, “Dynamic Quadruped Bounding Control with Duty Cycle Modulation Using Vertical Impulse Scaling,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[11] M. Focchi, R. Orsolino, M. Camurri, V. Barasuol, C. Mastalli, D. G. Caldwell, and C. Semini, “Heuristic Planning for Rough Terrain Locomotion in Presence of External Disturbances and Variable Perception Quality,” Nov. 2018. [Online]. Available: https://arxiv.org/pdf/1805.10238.pdf

[12] J. T. Betts, “Survey of Numerical Methods for Trajectory Optimization,” *Journal of Guidance Control Dynamics*, vol. 21, pp. 193–207, Mar. 1998.

[13] M. Diehl, “Lecture Notes on Numerical Optimization,” IMTEK - University of Freiburg, Tech. Rep., 2016.

[14] M. Diehl and S. Gros, “Numerical Optimal Control,” IMTEK - University of Freiburg, Tech. Rep., 2017.

[15] B. Siciliano and O. Khatib, *Springer Handbook of Robotics.* Berlin, Heidelberg: Springer-Verlag, 2007.

[16] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

[17] S. Boyd and L. Vandenberghe, *Convex Optimization.* New York, NY, USA: Cambridge University Press, 2004.

[18] Y. Wang and S. Boyd, “Fast Model Predictive Control Using Online Optimization,” *IEEE Transactions on Control Systems Tecnology*, vol. 18, no. 2, pp. 267–278, 2010. [Online]. Available: http://web.stanford.edu/~boyd/papers/pdf/fast_mpc.pdf

[19] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems.* Cambridge University Press, 2017.

[20] C. Mastalli, M. Focchi, I. Havoutis, A. Radulescu, S. Calinon, J. Buchli, D. G. Caldwell, and C. Semini, “Trajectory and foothold optimization using low-dimensional models for

rough terrain locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[21] S. Fahmi, C. Mastalli, M. Focchi, and C. Semini, "Passivity based whole-body control for quadrupedal locomotion on challenging terrain," *ArXiv*, 2018. [Online]. Available: https://arxiv.org/abs/1811.00884?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%253A+arxiv%252FQSXk+%2528ExcitingAds%2521+cs+updates+on+arXiv.org%2529

[22] R. Featherstone, "A beginner's guide to 6-d vectors (part 2) [tutorial]," *IEEE Robotics Automation Magazine*, vol. 17, no. 4, pp. 88–99, Dec 2010.

[23] ——, *Rigid Body Dynamics Algorithms.* Berlin, Heidelberg: Springer-Verlag, 2007.

[24] H. Bruyninckx, *Robot Kinematics and Dynamics.* Leuven, Belgium: Katholieke Universiteit Leuven, Department of Mechanical Engineering, 2010.

[25] N. Mansard, "Numerical Methods for Robotics," LAAS-CNRS Geppetto Team, Tech. Rep., 2015.

[26] R. Featherstone, "A beginner's guide to 6-d vectors (part 1) [tutorial]," *IEEE Robotics Automation Magazine*, vol. 17, no. 4, pp. 88–99, Dec 2010.

[27] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Auton. Robots*, vol. 35, no. 2-3, pp. 161–176, Oct. 2013. [Online]. Available: http://dx.doi.org/10.1007/s10514-013-9341-4

[28] M. Focchi, A. del Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, "High-slope terrain locomotion for torque-controlled quadruped robots," *Autonomous Robots*, 2017. [Online]. Available: http://dx.doi.org/10.1007/s10514-016-9573-1

[29] F. Nori, S. Traversaro, J. Eljaik, F. Romano, A. Del Prete, and D. Pucci, "iCub Whole-Body Control through Force Regulation on Rigid Non-Coplanar Contacts," *Frontiers in Robotics and AI*, pp. 1–18, 2015.

[30] P. M. Wensing and D. E. Orin, "Improved Computation of the Humanoid Centroidal Dynamics and Application for Whole-Body Control," *International Journal of Humanoid Robotics*, vol. 13, no. 1, 2016.

[31] "Boston Dynamics," *https://www.bostondynamics.com/*, 2018.

[32] N. A. Radford, P. Strawser, K. Hambuchen, J. S. Mehling, W. K. Verdeyen, A. S. Donnan, J. Holley, J. Sanchez, V. Nguyen, L. Bridgwater, R. Berka, R. Ambrose, M. Myles Markee, N. J. Fraser-Chanpong, C. McQuin, J. D. Yamokoski, S. Hart, R. Guo, A. Parsons, B. Wightman, P. Dinh, B. Ames, C. Blakely, C. Edmondson,

B. Sommers, R. Rea, C. Tobler, H. Bibby, B. Howard, L. Niu, A. Lee, M. Conover, L. Truong, R. Reed, D. Chesney, R. Platt, G. Johnson, C.-L. Fok, N. Paine, L. Sentis, E. Cousineau, R. Sinnet, J. Lack, M. Powell, B. Morris, A. Ames, and J. Akinyode, "Valkyrie: Nasa's first bipedal humanoid robot," *Journal of Field Robotics*, vol. 32, no. 3, pp. 397–419, May 2015. [Online]. Available: http://dx.doi.org/10.1002/rob.21560

[33] D. E. Orin and A. Goswami, "Centroidal momentum matrix of a humanoid robot: Structure and properties," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 653–659, 2008.

[34] M. H. J. Romanycia and F. J. Pelletier, "What is heuristics?" *Simon Frase University*, 1985.

[35] M. A. Daley, J. R. Usherwood, G. Felix, and A. A. Biewener, "Running over rough terrain: guinea fowl maintain dynamic stability despite a large unexpected change in substrate height," *The Journal of experimental biology*, vol. 209, no. 1, pp. 171–187, 2006. [Online]. Available: http://jeb.biologists.org/content/209/1/171

[36] R. E. Kambic, T. J. Roberts, and S. M. Gatesy, "Guineafowl with a twist: asymmetric limb control in steady bipedal locomotion," *Journal of Experimental Biology*, vol. 218, no. 23, pp. 3836–3844, 2015. [Online]. Available: http://jeb.biologists.org/content/218/23/3836

[37] P. E. Hudson, S. A. Corr, and A. M. Wilson, "High speed galloping in the cheetah (Acinonyx jubatus) and the racing greyhound (Canis familiaris): spatio-temporal and kinetic characteristics," *The Journal of experimental biology*, vol. 215, no. 14, pp. 2425–2434, 2012. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/22723482

[38] M. Focchi, R. Featherstone, R. Orsolino, D. G. Caldwell, and C. Semini, "Viscosity-based Height Reflex for Workspace Augmentation for Quadrupedal Locomotion on Rough Terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[39] R. Full and D. Koditschek, "Templates and anchors: neuromechanical hypotheses of legged locomotion on land," *Journal of Experimental Biology*, vol. 202, no. 23, pp. 3325–3332, 1999. [Online]. Available: http://jeb.biologists.org/content/202/23/3325

[40] I. Poulakakis, E. Papadopoulos, and M. Buehler, "On the stability of the passive dynamics of quadrupedal running with a bounding gait," *The International Journal of Robotics Research*, vol. 25, no. 7, pp. 669–687, 2006. [Online]. Available: https://doi.org/10.1177/0278364906066768

[41] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation," in

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), vol. 1, Oct 2001, pp. 239–246 vol.1.

[42] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1620–1626, 2003.

[43] P. Sardain and G. Bessonnet, "Forces acting on a biped robot. center of pressure-zero moment point," *Trans. Sys. Man Cyber. Part A*, vol. 34, no. 5, pp. 630–637, Sep. 2004. [Online]. Available: http://dx.doi.org/10.1109/TSMCA.2004.832811

[44] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and Trajectory Optimization for Legged Systems through Phase-based End-Effector Parameterization," *IEEE Robotics and Automation Letters (RA-L)*, pp. 1–1, 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8283570/

[45] P. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim, "Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots," *IEEE Transactions on Robotics (TRO)*, vol. PP, pp. 1–14, 01 2017.

[46] M. Hutter, C. Gehring, D. Jud, A. Lauber, D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. A. Höpflinger, "Anymal - a highly mobile and dynamic quadrupedal robot," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 38–44, 2016.

[47] C. Semini, V. Barasuol, J. Goldsmith, M. Frigerio, M. Focchi, Y. Gao, and D. G. Caldwell, "Design of the hydraulically-actuated, torque-controlled quadruped robot hyq2max," *IEEE/ASME Transactions on Mechatronics*, vol. PP, no. 99, pp. 1–1, 2016.

[48] N. Heess, T. Dhruva, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, ZiyuWang, S. M. A. Eslami, M. Riedmiller, and D. Silver, "Emergence of locomotion behaviours in rich environments," *ArXiv ID:1707.02286*, 2017.

[49] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *IEEE Robotics and Automation Letters (RA-L)*, 2018.

[50] O. Villarreal, V. Barasuol, M. Camurri, M. Focchi, L. Franceschi, M. Pontil, D. G. Caldwell, and C. Semini, "Fast and Continuous Foothold Adaptation for Dynamic Locomotion through Convolutional Neural Networks," *ArXiv*, 2018. [Online]. Available: https://arxiv.org/abs/1809.09759

[51] J. Carpentier, R. Budhiraja, and N. Mansard, “Learning Feasibility Constraints for Multi-contact Locomotion of Legged Robots,” in *RSS*, 2017.

[52] J. Carpentier and N. Mansard, “Multi-contact Locomotion of Legged Robots,” *IEEE Transactions on Robotics (TRO)*, Aug. 2018. [Online]. Available: https://hal.laas.fr/hal-01859108

[53] Y.-C. Lin, B. Ponton, L. Righetti, and D. Berenson, “Efficient humanoid contact planning using learned centroidal dynamics prediction,” *ArXiv*, 2018. [Online]. Available: https://arxiv.org/pdf/1810.13082.pdf

[54] J.-S. Pang and J. Trinkle, “Stability characterizations of rigid body contact problems with coulomb friction,” *Journal of Applied Mathematics and Mechanics (ZAMM)*, vol. 80, pp. 643 – 663, 10 2000.

[55] T. Bretl and S. Lall, “Testing static equilibrium for legged robots,” *IEEE Transactions on Robotics (TRO)*, vol. 24, no. 4, pp. 794–807, Aug. 2008. [Online]. Available: http://dx.doi.org/10.1109/TRO.2008.2001360

[56] J. E. Kelley, “The cutting-plane method for solving convex programs,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 4, pp. 703–712, 1960. [Online]. Available: http://www.jstor.org/stable/2099058

[57] A. Roennau, G. Heppner, M. Nowicki, and R. Dillmann, “Lauron v: A versatile six-legged walking robot with advanced maneuverability,” in *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, July 2014, pp. 82–87.

[58] R. Hodoshima, T. Doi, Y. Fukuda, S. Hirose, T. Okamoto, and J. Mori, “Development of titan xi: a quadruped walking robot to work on slopes,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, Sept 2004, pp. 792–797 vol.1.

[59] S. Karumanchi, K. Edelberg, I. Baldwin, J. Nash, J. Reid, C. Bergh, J. Leichty, K. Carpenter, M. Shekels, M. Gildner, D. Newill-Smith, J. Carlton, J. Koehler, T. Dobreva, M. Frost, P. Hebert, J. Borders, J. Ma, B. Douillard, and J. Burdick, “Team robosimian: SemiâĂŘautonomous mobile manipulation at the 2015 darpa robotics challenge finals,” *Journal of Field Robotics*, vol. 34, 10 2016.

[60] P.-B. Wieber, R. Tedrake, and S. Kuindersma, “Modeling and control of legged systems,” in *Springer Handbook of Robotics, 2nd Ed*, B. Siciliano and O. Khatib, Eds. Springer, 2016.

[61] M. Vukobratović and D. Juričić, “Contribution to the synthesis of biped gait,” *Proc. IFAC Symp. Technical and Biological Problem on Control*, Erevan, USSR, 1968.

[62] M. Vukobratović and B. Borovac, "Zero-moment point - thirty five years of its life," *International Journal of Humanoid Robotics*, pp. 157–173, 2004.

[63] P. Wensing, G. Bin Hammam, B. Dariush, and D. E. Orin, "Optimizing foot centers of pressure through force distribution in a humanoid robot," *International Journal of Humanoid Robotics*, vol. 10, 10 2013.

[64] M. B. Popovic, A. Goswami, and H. Herr, "Ground reference points in legged locomotion: Definitions, biological trajectories and control implications," *International Journal of Robotics Research (IJRR)*, pp. 1013–1032, 2005.

[65] T. Saida, Y. Yokokohji, and T. Yoshikawa, "FSW (feasible solution of wrench) for multi-legged robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2003. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1242182

[66] H. Hirukawa, K. Kaneko, S. Hattori, F. Kanehiro, K. Harada, K. Fujiwara, S. Kajita, and M. Morisawa, "A Universal Stability Criterion of the Foot Contact of Legged Robots - Adios ZMP," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

[67] S. Caron, Q.-C. Pham, and Y. Nakamura, "ZMP support areas for multi-contact mobility under frictional constraints," *Transactions on Robotics (TRO)*, vol. 33, pp. 67–80, 2017.

[68] H. Dai and R. Tedrake, "Planning robust walking motion on uneven terrain via convex optimization," in *Humanoids*, 2016.

[69] H. Dai, "Robust multi-contact dynamical motion planning using contact wrench set," Ph.D. dissertation, Massachusetts Institute of Technology, 2016.

[70] Y. Or and E. Rimon, "Characterization of frictional multi-legged equilibrium postures on uneven terrains," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 105–128, 2017. [Online]. Available: https://doi.org/10.1177/0278364916679719

[71] A. Bicchi, "On the Closure Properties of Robotic Grasping," *IJRR*, vol. 14, 1995.

[72] R. Krug, A. J. Lilienthal, D. Kragic, and Y. Bekiroglu, "Analytic Grasp Success Prediction with Tactile Feedback," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[73] V.-D. Nguyen, "Constructing force-closure grasps," *Int. J. Rob. Res.*, vol. 7, no. 3, pp. 3–16, Jun. 1988. [Online]. Available: http://dx.doi.org/10.1177/027836498800700301

[74] H. Dai, A. Majumdar, and R. Tedrake, "Synthesis and optimization of force closure grasps via sequential semidefinite programming," pp. 285–305, 01 2018.

[75] Y. Zheng and K. Yamane, "Generalized distance between compact convex sets: Algorithms and applications," *IEEE Transactions on Robotics (TRO*, vol. 31, no. 4, pp. 988–1003, Aug 2015.

[76] H. Audren and A. Kheddar, "3-d robust stability polyhedron in multicontact," *IEEE Transactions on Robotics (TRO)*, vol. PP, 02 2018.

[77] S. Caron and A. Kheddar, "Multi-contact walking pattern generation based on model preview control of 3d com accelerations," in *Proceedings of the 2016 IEEE-RAS International Conference on Humanoid Robots*, Nov. 2016. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01349880

[78] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," pp. 200 – 207, 01 2007.

[79] T. Koolen, T. de Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models," vol. 31, pp. 1094–1113, 2012.

[80] A. Hof, M. Gazendam, and W. Sinke, "The condition for dynamic stability," *Journal of biomechanics*, vol. 38, pp. 1–8, 02 2005.

[81] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot-1(st) report: Walking gait pattern generation," pp. 1084 – 1091, 11 2009.

[82] J. Englsberger, C. Ott, and A. Albu-Schäeffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Transactions on Robotics (TRO)*, vol. 31, 11 2013.

[83] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, "A convex model of humanoid momentum dynamics for multi-contact motion generation," *IEEE/RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016. [Online]. Available: https://arxiv.org/pdf/1810.13082.pdf

[84] H. K. Khalil, *Nonlinear systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002, the book can be consulted by contacting: PH-AID: Wallet, Lionel. [Online]. Available: https://cds.cern.ch/record/1173048

[85] I. A. Hiskens, "Stability of limit cycles in hybrid systems," in *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, Jan 2001, pp. 6 pp.–.

[86] X. Liu, C. Semini, and I. Poulakakis, "Active compliance hybrid zero dynamics control of bounding on HyQ," *IEEE International Conference on Robotics and Biomimetics, IEEE-ROBIO*, pp. 1047–1052, 2015.

[87] T. S. Parker and L. O. Chua, *Practical Numerical Algorithms for Chaotic Systems.* Berlin, Heidelberg: Springer-Verlag, 1989.

[88] M. Cheng and C. Lin, "Measurement of robustness for biped locomotion using linearized poincare' map," in *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, vol. 2, Oct 1995, pp. 1321–1326 vol.2.

[89] H. Dai and R. Tedrake, "Optimizing robust limit cycles for legged locomotion on unknown terrain," in *IEEE Conference on Decision and Control (CDC)*, Dec 2012, pp. 1207–1213.

[90] T. McGeer, "Passive dynamic walking," *Int. J. Rob. Res.*, vol. 9, no. 2, pp. 62–82, Mar. 1990. [Online]. Available: http://dx.doi.org/10.1177/027836499000900206

[91] M. H. Raibert, "Trotting, pacing and bounding by a quadruped robot," *Journal of Biomechanics*, vol. 23, pp. 79–98, 1990.

[92] H. Kimura, Y. Fukuoka, and K. Takase, "Three-dimensional Adaptive Dynamic Walking," *in Proc. of Int. Conf. on Robotics and Automation*, vol. 3, no. May, pp. 2228–2233, 2002.

[93] D. Papadopoulos and M. Buehler, "Stable Running in a Quadruped Robot with Compliant Legs," in *IEEE Int. Conf. Robotics and Automation*, 2000, pp. 444–449.

[94] L. D. Maes, M. Herbin, R. Hackert, V. L. Bels, and A. Abourachid, "Steady locomotion in dogs: temporal and associated spatial coordination patterns and the effect of speed," *Journal of Experimental Biology*, vol. 211, no. 1, pp. 138–149, 2007.

[95] R. M. Walter and D. R. Carrier, "Rapid acceleration in dogs: ground forces and body posture dynamics," *Journal of Experimental Biology*, vol. 212, no. 12, pp. 1930–1939, 2009.

[96] Y. Fukuoka, Y. Habu, and T. Fukui, "A simple rule for quadrupedal gait generation determined by leg loading feedback: a modeling study," *Scientific Reports*, p. 8169, 2015.

[97] I. Poulakakis and J. W. Grizzle, "The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper," *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1779–1793, 2009.

[98] H. Nie, R. Sun, and C. Xiong, “The Effect of Asymmetrical Body-Mass Distribution on the Stability and Dynamics of Quadruped Bounding,” vol. 22, no. 4, pp. 243–253, 2015.

[99] S. H. Hyon and T. Mita, “Development of a Biologically Inspired Hopping Robot - "Kenken",” *IEEE International Conference on Robotics and Automation (ICRA)*, no. May, pp. 3984–3991, 2002.

[100] S. H. Hyon and T. Emura, “Symmetric walking control: Invariance and global stability,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2005, no. April, pp. 1443–1450, 2005.

[101] H.-W. Park, S. Park, and S. Kim, “Variable-speed Quadrupedal Bounding Using Impulse Planning: Untethered High-speed 3D Running of MIT Cheetah 2,” *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5163–5170, 2015.

[102] H.-W. Park, P. M. Wensing, and K. Sangbae, “Online Planning for Autonomous Running Jumps Over Obstacles in High-Speed Quadrupeds,” *Robotics: Science and Systems (RSS) XI*, 2015.

[103] A. Wilson and P. McGuigan, “Pogo stick horse legs need better track surfaces,” *Journal of Experimental Biology*, vol. 206, p. 1261, 2003.

[104] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, “Design of HyQ - a hydraulically and electrically actuated quadruped robot,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 225, no. 6, pp. 831–849, aug 2011.

[105] C. Truesdell, *Goldstein’s Classical Mechanics (1950).* Springer New York, 1984, pp. 144–147.

[106] J. Englsberger and C. Ott, “Biologically Inspired Deadbeat control for running on 3D stepping stones,” *Humanoids, Seoul, Korea*, vol. 206, pp. 1067–1074, 2015.

[107] C. Semini, V. Barasuol, T. Boaventura, M. Frigerio, M. Focchi, D. G. Caldwell, and J. Buchli, “Towards versatile legged robots through active impedance control,” *The International Journal of Robotics Research (IJRR)*, vol. 34, no. 7, pp. 1003–1020, 2015. [Online]. Available: http://ijr.sagepub.com/content/34/7/1003

[108] V. Barasuol, J. Buchli, C. Semini, M. Frigerio, E. R. De Pieri, and D. G. Caldwell, “A reactive controller framework for quadrupedal locomotion on challenging terrain,” *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2554–2561, 5 2013.

[109] L. R. Palmer and D. E. Orin, “Intelligent control of high-speed turning in a quadruped,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, pp. 47–68, 2010.

[110] K. Tsujita, H. Toui, and K. Tsuchiya, “Dynamic Turning Control of a Quadruped Robot using Oscillator Network,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 19, pp. 1115–1133, 2004.

[111] G. Tournois, M. Focchi, A. Del Prete, R. Orsolino, D. Caldwell, and C. Semini, “Online Payload Identification for Quadruped Robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[112] P. Wensing, G. Niemeyer, and J.-J. E. Slotine, “Observability in inertial parameter identification,” 11 2017. [Online]. Available: https://arxiv.org/pdf/1711.03896.pdf

[113] P. M. Wensing, S. Kim, and J. E. Slotine, “Linear matrix inequalities for physically consistent inertial parameter identification: A statistical perspective on the mass distribution,” *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 1, pp. 60–67, Jan 2018.

[114] A. Del Prete, S. Tonneau, and N. Mansard, “Fast algorithms to test robust static equilibrium for legged robots,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[115] S. Caron, Q.-C. Pham, and Y. Nakamura, “Stability of surface contacts for humanoid robots: Closed-form formulae of the contact wrench for rectangular support areas,” in *Proceedings of the 2015 IEEE-RAS International Conference on Robotics and Automation*, May 2015, pp. 5107–5112.

[116] B. Aceituno-Cabezas, C. Mastalli, D. Hongkai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernando-Lopez, and C. Semini, “Simultaneous Contact, Gait and Motion Planning for Robust Multi-Legged Locomotion via Mixed-Integer Convex Optimization,” in *IEEE Robotics and Automation Letters (RA-L)*, 2018.

[117] V. Samy, S. Caron, K. Bouyarmane, and A. Kheddar, “Post-Impact Adaptive Compliance for Humanoid Falls Using Predictive Control of a Reduced Model,” July 2017, eprint hal:01569819.

[118] V. Delos, “Politopix,” *http://i2m.u-bordeaux.fr/politopix.html*, 2015.

[119] V. Delos and D. Teissandier, “Minkowski sum of polytopes defined by their vertices,” *Journal of Applied Mathematics and Physics*, 2015.

[120] K. Fukuda and A. Prodon, “Double description method revisited,” in *Selected Papers from the 8th Franco-Japanese and 4th Franco-Chinese Conference on Combinatorics and Computer Science*. Berlin, Heidelberg: Springer-Verlag, 1996, pp. 91–111. [Online]. Available: http://dl.acm.org/citation.cfm?id=646124.680737

[121] C. Semini, N. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. Caldwell, "Design of HyQ – a Hydraulically and Electrically Actuated Quadruped Robot," *J. of Systems and Control Eng.*, 2011.

[122] P. Bosscher, A. T. Riechel, and I. Ebert-Uphoff, "Wrench-feasible workspace generation for cable-driven robots," *IEEE Transactions on Robotics (TRO)*, vol. 22, 2006.

[123] S. Bouchard, C. Gosselin, and B. Moore, "On the ability of a cable-driven robot to generate a prescribed set of wrenches," in *ASME Journal of Mechanisms and Robotics*, 2010.

[124] S. Arroyave-Tobon, D. Teissandier, and V. Delos, "Applying screw theory for summing sets of constraints in geometric tolerancing," *Mechanism and Machine Theory*, 2017.

[125] S. Caron, Q.-C. Pham, and Y. Nakamura, "Leveraging Cone Double Description for Multi-contact Stability of Humanoids with Applications to Statics and Dynamics," in *RSS*, 2015.

[126] S. Caron and A. Kheddar, "Dynamic Walking over Rough Terrains by Nonlinear Predictive Control of the Floating-base Inverted Pendulum," in *eprint arXiv:1703.00688*, 2017.

[127] C. Melchiorri, P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Comments on "global task space manipulability ellipsoids for multiple-arm systems" and further considerations" [with reply] p. chiacchio, et al," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 2, pp. 232–236, April 1993.

[128] P. Chiacchio, Y. Bouffard-Vercelli, and F. Pierrot, "Force Polytope and Force Ellipsoid for Redundant Manipulators," *Journal of Robotic Systems*, vol. 14, no. 8, pp. 613–620, 1997. [Online]. Available: http://doi.wiley.com/10.1002/(SICI)1097-4563(199708)14:8{%}3C613::AID-ROB3{%}3E3.0.CO;2-P

[129] P. Chiacchio and M. Concilio, "The dynamic manipulability ellipsoid for redundant manipulators," *IEEE International Conference on Robotics and Automation (ICRA)*, 1998.

[130] A. L. Cruz Ruiz, S. Caro, P. Cardou, and F. Guay, "ARACHNIS: Analysis of Robots Actuated by Cables with Handy and Neat Interface Software," in *Cable-Driven Parallel Robots, Mechanisms and Machine Science*, vol. 32, 2015, pp. 293 – 305.

[131] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Influence of gravity on the manipulability ellipsoid for robot arms," *Journal of Dynamic Systems Measurement and Control*, vol. 114, pp. 723–727, 12 1992.

[132] T. Yoshikawa, "Dynamic manipulability of robot manipulators," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, March 1985, pp. 1033–1038.

[133] K. Fukuda, "Frequently asked questions in polyhedral computation," 1998. [Online]. Available: https://www.cs.mcgill.ca/~fukuda/soft/polyfaq/polyfaq.html

[134] P.-B. Wieber, "Holonomy and nonholonomy in the dynamics of articulated motion," vol. 340, pp. 411–425, 07 2007.

[135] F. Guay, P. Cardou, A. L. Cruz Ruiz, and S. Caro, "Measuring how well a structure supports varying external wrenches," in *New Advances in Mechanisms, Transmissions and Applications*, 2014, pp. 385 – 392.

[136] V. Delos and D. Teissandier, "Minkowski sum of HV-polytopes in Rn," in *eprint arXiv:1412.2562*, 2014.

[137] C. Boussema, M. J. Powell, G. Bledt, A. J. Ijspeert, P. M. Wensing, and S. Kim, "Gait emergence and disturbance recovery for legged robots via the feasible impulse set," *IEEE Robotics and Automation Letters (RA-L)*, 2019.

[138] V. Barasuol, M. Camurri, S. Bazeille, D. Caldwell, and C. Semini, "Reactive trotting with foot placement corrections through visual pattern classification," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[139] M. Eich and T. Vögele, "Design and control of a lightweight magnetic climbing robot for vessel inspection," *19th Mediterranean Conference on Control and Automation*, June 2011.

[140] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Carnegie Mellon University, Robotics Institute, August 2010. [Online]. Available: http://www.programmingvision.com/rosen_diankov_thesis.pdf

[141] A. Herzog, S. Schaal, and L. Righetti, "Structured contact force optimization for kino-dynamic motion generation," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2703–2710, 10 2016.

[142] M. Camurri, M. Fallon, S. Bazeille, A. Radulescu, V. Barasuol, D. G. Caldwell, and C. Semini, "Probabilistic contact estimation and impact detection for state estimation of quadruped robots," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1023–1030, April 2017.

[143] S. Nobili, M. Camurri, V. Barasuol, M. Focchi, D. G. Caldwell, C. Semini, and M. Fallon, "Heterogeneous sensor fusion for accurate state estimation of dynamic legged robots," in *Proceedings of Robotics: Science and Systems*, Boston, USA, July 2017.

[144] B. Aceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, C. J., J. C. Grieco, F.-L. G., and C. Semini, "Simultaneous contact, gait and motion planning for robust multi-legged locomotion via mixed-integer convex optimization," *IEEE Robotics and Automation Letters (RA-L)*, 2017.

[145] B. Ponton, A. Herzog, A. Del Prete, S. Schaal, and L. Righetti, "On time optimization of centroidal momentum dynamics," 2018. [Online]. Available: https://arxiv.org/pdf/1810.13082.pdf

[146] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation.* Boca Raton, FL: CRC Press, 1994. [Online]. Available: http://www.cds.caltech.edu/~murray/books/MLS/pdf/mls94-nonholo_v1_2.pdf

[147] M. T. Mason, "Lecture 5 on Mechanics of Manipulation: Nonholonomic Constraint," Carnegie Mellon University (CMU), Tech. Rep., 2017. [Online]. Available: http://www.cs.cmu.edu/afs/cs/academic/class/16741-s07/www/lectures/lecture5.pdf

[148] M. Althoff, "Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars," Ph.D. dissertation, Technische Universität München (TUM), 2010.

[149] ——, "CORA 2016 Manual," Technische Universität München (TUM), Tech. Rep., 2016.