SANTA CATARINA STATE UNIVERSITY – UDESC
COLLEGE OF TECHNOLOGICAL SCIENCE – CCT
GRADUATE PROGRAM IN ELECTRICAL ENGINEERING – PPGEEL

LUCAS SCHULZE

STOCHASTIC MODEL PREDICTIVE CONTROL FOR DYNAMIC LOCOMOTION OF LEGGED ROBOTS

JOINVILLE
2022

LUCAS SCHULZE

STOCHASTIC MODEL PREDICTIVE CONTROL FOR DYNAMIC LOCOMOTION OF LEGGED ROBOTS

Master's thesis presented to the Graduate Program in Electrical Engineering of the Santa Catarina State University, as partial requirement for obtention of the title of Master in Electrical Engineering.

Advisor: Douglas Wildgrube Bertol
Co-advisor: Guilherme Vianna Raffo

JOINVILLE
2022

Schulze, Lucas
Stochastic Model Predictive Control for Dynamic Locomotion of Legged Robots / Lucas Schulze. - Joinville, 2022.
67 p. : il. ; 30 cm.

Orientador: Douglas Wildgrube Bertol.
Coorientador: Guilherme Vianna Raffo.
Dissertação (mestrado) —— Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, Programa de Pós-Graduação em Engenharia Elétrica, Joinville, 2022.

1. Robôs com Pernas. 2. Controle Preditivo Estocástico Baseado em Modelo. 3. Chance-Constraints. 4. Locomoção com Pernas. 5. HyQ. I. Bertol, Douglas Wildgrube . II. Raffo, Guilherme Vianna . III. Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, Programa de Pós-Graduação em Engenharia Elétrica. IV. Título.

LUCAS SCHULZE

STOCHASTIC MODEL PREDICTIVE CONTROL FOR DYNAMIC LOCOMOTION OF LEGGED ROBOTS

Master's thesis presented to the Graduate Program in Electrical Engineering of the Santa Catarina State University, as partial requirement for obtention of the title of Master in Electrical Engineering.

Advisor: Douglas Wildgrube Bertol
Co-advisor: Guilherme Vianna Raffo

EXAMINATION BOARD:

Douglas Wildgrube Bertol, Doctor
Santa Catarina State University (UDESC) (Board's President / Advisor)

Members:

Victor Barasuol, Doctor
Istituto Italiano di Tecnologia (IIT)

Tito Luís Maia Santos, Doctor
Federal University of Bahia (UFBA)

Joinville, January 24, 2022

À minha família

ACKNOWLEDGEMENTS

First of all, I would like to thank my family, especially my parents, Eliane Schwitzki Schulze and Marcio Schulze, for all the love, support and the encouragement to follow my dreams.

I would like to thank my advisors, prof. Douglas Wildgrube Bertol and prof. Guilherme Vianna Raffo, for their guidance, time, and attention. For all the insightful discussions we had and for passing on what they have learned.

I would like to thank the Istituto Italiano di Tecnologia (IIT) for the opportunity to work with the HyQ robot and for the access to IIT's software framework for legged robots. A special thanks to Victor Barausol, who made possible the partnership between UDESC and IIT. I also thank Octavio Villareal for his time and patience to introduce me to the framework.

I thank to UDESC, GASR, LAMAN, and LAPAS for the infrastructural and the offered opportunities. I also thank to the UNIEDU/FUMDES post-graduation program from the Santa Catarina State Government for the financial support.

I would like to express my especial thanks to Aureo Guilherme Dobrikopf and Luan Vinícius Fiorio for all the discussions, brainstorm, insights, and for their friendship.

A big thanks to all my friends for making my life easier with all the fun and laughs during all those years.

"O que não te desafia, não te transforma."
(Desconhecido)

ABSTRACT

Legged robots are considered hybrid dynamics systems due to the presence of continuous and discrete dynamics resultant of the switching of the footholds. Especially on rough terrains, a desired motion plan may not be feasible, while a controller aims to execute it maintaining the robot in equilibrium. In this context, Model Predictive Control (MPC) is gaining popularity due to the possibility to consider constraints and future changes on the footholds and in the terrain. However, to formulate computational tractable optimization problems, simplified mathematical model are employed, leading to system uncertainties that may affect the execution of the motion plan and the robot's stability. Stochastic Model Predictive Control (SMPC) emerged to address the presence of uncertainties in the system description into the controller design, balancing constraint satisfaction and conservatism. Thus, this work proposes the comparison of MPC and SMPC for legged locomotion on rough terrains, and the evaluation of the effect of the angular momentum rate minimization. MPC and SMPC are compared in a simulation scenario with obstacles for the HyQ robot. As result, SMPC and the minimization of the angular momentum rate demonstrated to improve the legged robot locomotion with relation to constraints satisfaction and trajectory smoothness.
Keywords: Legged Robots. Stochastic Model Predictive Control. Chance-Constraints. Legged Locomotion. HyQ.

RESUMO

Robôs com pernas são considerados sistemas com dinâmica híbrida devido à presença de dinâmicas contínuas e discretas resultantes da troca dos pés em contato com o solo. Especialmente em terrenos acidentados, um movimento planejado pode não ser factível, enquanto um controlador tenta executá-lo mantendo o robô em equilíbrio. Nesse contexto, Controle Preditivo Baseado em Modelo (MPC) tem ganhado popularidade devido à possibilidade de considerar restrições e futuras mudanças no terreno e em quais pés estarão em contato com o solo. No entanto, para formular um problema de otimização computacionalmente tratável, modelos matemáticos simplificados são utilizados, gerando incertezas na descrição do sistema que podem afetar a execução de um plano de movimento e o equilíbrio do robô. MPC Estocásticos (SMPC) surgiram de forma a considerar no projeto do controlador a presença de incertezas associadas à descrição do sistema, ponderando entre satisfação das restrições e convervadorismo. Dessa forma, esse trabalho propõe a comparação entre MPC e SMPC para a locomoção com pernas em terrenos acidentados, e a avaliação do efeito da minimização da variação do momento angular. MPC e SMPC são avaliados em simulação em um cenário com obstáculos para o robô HyQ. Como resultado, SMPC e a minimização da variação do momento angular demonstraram melhorar a locomoção com pernas com relação à satisfação das restrições e com uma trajetória mais suave.

Palavras-chave: Robôs com Pernas. Controle Preditivo Estocástico Baseado em Modelo. *Chance-Constraints*. Locomoção com Pernas. HyQ.

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

CNN	Convolutional Neural Network
CoM	Center of Mass
DLS	Dynamic Legged Systems lab
DOF	Degrees of Freedom
GRF	Ground Reaction Force
HAA	Hip Abduction/Adduction
HFE	Hip Flexion/Extension
HyQ	Hydraulic Quadruped
IIT	Istituto Italiano di Tecnologia
IMU	Inertial Measurement Unit
KFE	Knee Flexion/Extension
LF	Left-Front
LH	Left-Hind
LQR	Linear Quadratic Regulator
MPC	Model Predictive Control
MPC-C	MPC with Angular Velocity Constraint
MPC-CL	MPC with Angular Velocity Constraint and $\dot{\mathbf{L}}$ weighting
MPC-L	MPC with $\dot{\mathbf{L}}$ weighting
PD	Proportional-Derivative
RCF	Reactive Controller Framework
RF	Right-Front
RH	Right-Hind
RMPC	Robust Model Predictive Control
RMS	Root Mean Square
ROS	Robot Operating System
SMPC	Stochastic Model Predictive Control

SMPC-L	SMPC with $\dot{\mathbf{L}}$ weighting
VFA	Vision-based Foothold Adaptation
VO	Visual Odometry
ZOH	Zero-Order Hold

LIST OF SYMBOLS

Notations

a	Lower case letters denotes scalars
a	Boldface lower case letters denote vectors
A	Boldface upper case letters denote matrices
a	Boldface italic lower case letters denote time-variant vectors
A	Boldface italic upper case letters denote time-variant matrices

Symbols

t_{sw}	Swing phase time
t_{st}	Stance phase time
f_{step}	Step frequency
t_{step}	Step period
D_f	Duty factor between the leg in stance and the step period
h_s	Step height
$\boldsymbol{\ell}_s$	Step length
n_y	Prediction horizon
n_u	Control horizon
$\boldsymbol{x}$	State-space state vector
$\boldsymbol{u}$	State-space control input vector
$\boldsymbol{y}$	State-space output vector
A	Discrete state-space state matrix
B	Discrete state-space input matrix
C	Discrete state-space output matrix
n	Number of system's states
m	Number of system's control inputs
p	Number of system's outputs
$\bar{\boldsymbol{x}}$	State vector along the prediction horizon
$\bar{\boldsymbol{u}}$	Control input vector along the control horizon
$\bar{\boldsymbol{y}}$	Output vector along the control horizon
$\mathbf{S}^x$	State matrix along the horizons
$\mathbf{S}^u$	Control matrix along the horizons

$\bar{\mathbf{C}}$	Output matrix along the horizons
$\mathbf{1}$	Identity matrix with proper dimension
$\mathbf{0}$	Zero matrix with proper dimension
$\mathbf{J}_{\text{MPC}}$	Cost functional from MPC
$\mathbf{Q}$	Weighting matrix for the states
$\mathbf{P}$	Weighting matrix for the terminal states
$\mathbf{R}$	Weighting matrix for the control inputs
$\mathbf{G}_{\text{c}}$, $\mathbf{E}_{\text{c}}$, $\mathbf{h}_{\text{c}}$	Matrices of the linear combination of the state and control input to define a constraint
$\boldsymbol{w}$	Process noise vector
$\boldsymbol{v}$	Measurement noise vector
$\mathbf{D}$	Discrete state-space noise mapping matrix
$\text{n}_{\mathbf{w}}$	Number of system's noise
$\text{n}_{\mathbf{v}}$	Number of measurement's noise
$\mathscr{N}$	Normal distribution
$\Sigma^{\boldsymbol{w}}$	Noise variance vector
Ω	Convex hull of the polytopic uncertainty
$\boldsymbol{w}_{\text{max}}$	Noise bounded support
$\mathbb{E}(x)$	Expected value of a variable x
$\boldsymbol{z}$	Deterministic component of the system's state
$\boldsymbol{e}$	Random component of the system's state
$\mathbf{K}$	Feedback gain associated to $\boldsymbol{x}_k$ in the affine redefinition of $\boldsymbol{u}_k$
$\boldsymbol{c}$	Affine component of the affine redefinition of $\boldsymbol{u}_k$
$\boldsymbol{\Phi}$	Closed-loop state matrix associated to the affine redefinition of $\boldsymbol{u}_k$
$\bar{\boldsymbol{z}}$	Deterministic component of the system's state along the horizons
$\mathbf{S}^{\Phi}$	Closed-loop state matrix along the horizons
$\mathbf{S}^{c}$	Closed-loop control matrix along the horizons
$\bar{\boldsymbol{c}}$	Affine component of the control input vector along the control horizon
$\bar{\boldsymbol{e}}$	Random component of the system's state along the horizon
$\mathbf{S}^{w}$	Noise matrix along the horizons
$\bar{\boldsymbol{w}}$	Noise vector along the prediction horizon

$\mathbf{J}_{\text{SMPC}}$	Cost functional from SMPC based on the expected value
n_s	Number of noise samples
$\mathbf{J}_{\text{AVG}}$	Cost functional from SMPC based on the average cost among multiple noise samples realization
$\mathbf{J}_{\max}$	Cost functional from SMPC based on the maximum cost among noise multiple samples realization
λ	Constraint's tightened term to compensate for the influence of uncertainty in the system's state
$\bar{\lambda}$	Constraint's tightened term along the prediction horizon
$\mathbf{G}_{\boldsymbol{x}}$, $\mathbf{h}_{\boldsymbol{x}}$	Matrices of a state constraint
$\mathscr{W}$	Uncertainty Set
$\bar{\mathbf{G}}_{\boldsymbol{x}}$	Diagonal composition of the matrix $\mathbf{G}_{\boldsymbol{x}}$
$\mathscr{P}\{\mathscr{X}\}$	Probability of the event $\mathscr{X}$ occurs
p	Maximum acceptable probability level of the constraints to be violated
$\Sigma^{\bar{\boldsymbol{w}}}$	Diagonal composition of the matrix $\Sigma^{\boldsymbol{w}}$
f_{cdf}	Cumulative distribution function of a Gaussian variable with zero mean and unitary variance
f_{p}	Cumulative distribution function for a generic distribution
$\boldsymbol{u}_{ff}$	Feedforward component of $\boldsymbol{u}$
$\mathbf{L}$	Feedback gain of the affine disturbance feedback law
$\boldsymbol{n}$	Affine component of the affine disturbance feedback law
n_{sc}	Number of scenarios evaluated in the Scenario-Based approach
$\mathbf{V}_{\text{f}}$	User-defined forward velocity
$V_{\text{f,x}}$, $V_{\text{f,y}}$	Components x and y of $\mathbf{V}_{\text{f}}$
$\mathbf{r}$	Linear position of the robot with respect to the world frame
Θ	Orientation with respect to the world frame
ω	Body angular velocity with respect to the world frame
$\mathbf{p}$	Position of the contact point
$\hat{\mathbf{p}}$	Nominal foothold position
$\bar{\mathbf{p}}$	Reference position on the ground determined by the elliptical trajectory of the feet
$\mathbf{p}^*$	Optimal foothold position

$\mathbf{H}$	Terrain Heightmap
$\Theta = [\phi, \theta, \psi]^{\mathrm{T}}$	Roll, pitch, and yaw angles in the world frame
$\mathbf{r} = [\mathrm{x}, \mathrm{y}, \mathrm{z}]^{\mathrm{T}}$	Linear positions in the world frame
Θ_{d}	Desired body orientation with respect to the world frame
$\Theta_{\mathrm{d}} = [\phi_{\mathrm{d}}, \theta_{\mathrm{d}}, \psi_{\mathrm{d}}]^{\mathrm{T}}$	Desired Roll, pitch, and yaw angles in the world frame
$\mathbf{r}_{\mathrm{d}}$	Desired body linear position $\mathbf{r}$
$\mathbf{R}_{\mathrm{z}}$	Orthogonal rotation matrix around the z-axis
n	Index of the stance along the reference
p_{cz}	Height of the surface from the world origin
$\mathbf{q}$	Joints position vector
$\mathbf{q}_{\mathrm{d}}$	Desired joints position vector
$\mathrm{n_j}$	Number of system's joints
τ	Joints torque vector
τ_{fb}	Joints torque vector of the legs in swing
τ_{tc}	Computed torque from the trunk controller
$\mathbf{M}$	Joint-space inertia matrix
$\mathbf{V}$	Vector of the Coriolis, centrifugal and gravitational forces
$\mathbf{J}_{\mathrm{c}}$	Contact Jacobian
$\boldsymbol{F}$	Ground reaction forces vector
$\mathbf{M}_{\mathrm{u}}$, $\mathbf{V}_{\mathrm{u}}$, $\mathbf{J}^{\mathrm{T}}_{\mathrm{cu}}$	Unactuated components of $\mathbf{M}$, $\mathbf{V}$, and $\mathbf{J}^{\mathrm{T}}_{\mathrm{c}}$
$\mathbf{M}_{\mathrm{a}}$, $\mathbf{V}_{\mathrm{a}}$, $\mathbf{J}^{\mathrm{T}}_{\mathrm{ca}}$	Actuated components of $\mathbf{M}$, $\mathbf{V}$, and $\mathbf{J}^{\mathrm{T}}_{\mathrm{c}}$
$\mathbf{M}_{\mathrm{ua}}$	Cross-terms of the unactuated and actuated from $\mathbf{M}$
ν	Vector composition of the body angular and linear velocities
m	total mass of the robot
g	Gravitational acceleration value
$\mathbf{g}$	Gravitational acceleration vector
$\mathrm{n_c}$	Number of contact points
$\mathbf{I_B}$	Inertia tensor of the robot in the body frame
$\mathbf{I}$	Inertia tensor of the robot in the world frame
$\mathbf{I}_{\mathrm{leg}}$	Leg's inertia projected in the robot's CoM
$\mathbf{S}()$	Skew-symmetric matrix operator

$\mathbf{R}$	Rotation matrix from the body to the world
W_η	Rotational Jacobian from ω to $\dot{\Theta}$
$\mathbf{I}_{\text{leg}}$	Inertial matrix of the legs in the world frame
$\mathbf{A}_{\text{c}}$	Continuous state-space state matrix
$\mathbf{B}_{\text{c}}$	Continuous state-space input matrix
T_{s}	Sample time
$\boldsymbol{x}_{\text{d}}$	Desired state vector
μ	Friction coefficient
$\boldsymbol{F}_{xi}$, $\boldsymbol{F}_{y_i}$, $\boldsymbol{F}_{zi}$	Linear components of the GRF from ith leg
F_{max}	Maximum ground reaction forces vector
F_{min}	Minimum ground reaction forces vector
$\mathbf{k}_{\mathbf{F}i}$, $\mathbf{f}_{\text{c}i}$	Matrices associated to the cone of friction constraint for the ith leg
$\mathbf{k}_{\mathbf{F}}$, $\mathbf{f}_{\text{c}}$	Diagonal composition of four matrices $\mathbf{k}_{\mathbf{F}i}$ and $\mathbf{f}_{\text{c}i}$
$\mathscr{G}$	Mapping matrix for the swing legs
$\mathcal{W}_{\text{MPC}}$	Resultant wrench associated to the computed GRFs from MPC
$\mathcal{W}_{\text{leg}}$	Wrench associated to the leg's inertia
$\mathcal{W}_{\text{d}}$	Desired wrench for the trunk controller
$\mathcal{W}_{\text{com}}$	Wrench of the CoM
$\mathbf{J}_{\text{ca,st}}$	Contact Jacobian between the stance feet and the CoM
$\mathbf{J}^{\text{T}}_{\text{st,com}}$	Jacobian between the stance feet and the CoM
$\mathbf{q}_{\text{st}}$	Joints position vector of the legs in stance
$\mathbf{Q}_{\text{tc}}$, $\mathbf{R}_{\text{tc}}$	Weighting matrices from the
l_{la}	Effective lever arm length between hydraulic actuator and joint
u_v	Valve position
$\mathbf{d}_{\text{g}}$	State-space known disturbance vector
$\mathbf{E}_{\text{cg}}$	Continuous state-space known disturbance matrix
$\mathbf{E}_{\text{g}}$	Discrete state-space known disturbance matrix
$\bar{\mathbf{d}}_{\text{g}}$	Known vector along the prediction horizon
$\mathbf{S}^{\text{g}}$	Known disturbance matrix along the horizons
ω_{x}, ω_{y}, ω_{z}	Components of the angular velocity ω
$\omega_{\text{x}\text{max}}$, $\omega_{\text{y}_{\text{max}}}$	Maximum values for the ω_{x} and ω_{y} components

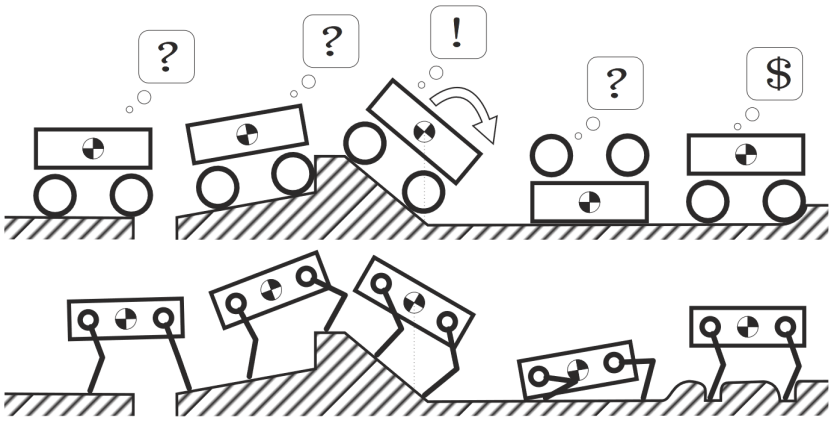
$\omega_{\mathrm{x}\min}$, $\omega_{\mathrm{y}_{\min}}$	Minimum value for the ω_{x} and ω_{y} components
$\mathbf{C}_{\omega}$	Output matrix for the angular velocity ω
$\mathbf{h}_{\omega}$	Vector composition of the angular velocity constraints
$\bar{\mathbf{C}}_{\omega}$	Diagonal composition of the matrix $\mathbf{C}_{\omega}$
$\mathbf{L}$	Angular momentum rate
τ_{GRF}	Projected torques at the CoM exerted by the GRFs
$\mathbf{S}_{\mathrm{skew}}$	Matrix composition of skew-symmetric matrices related to the contact points
$\mathbf{R}_{\mathbf{L}}$	Weighting matrix for the angular momentum rate
$\Sigma^{\boldsymbol{w}}{}_{\mathrm{trial}}$	Analyzed noise variance vector
n_j	Number of constraints violation for the jth constraint, where $j = \omega_{\mathrm{x}\max}$, $\omega_{\mathrm{x}\min}$, $\omega_{\mathrm{y}_{\max}}$, $\omega_{\mathrm{y}_{\min}}$
$\overline{\mathrm{n}}_j$, σ_j	Average and standard deviation of n_j along the simulation trials
$\overline{\mathrm{RMS}}_j$, σ_j	Average and standard deviation of the variable j along the simulation trials
$\mathbf{J}_{\mathrm{c}}$	Cost functional
$\overline{\mathbf{J}_{\mathrm{c}}}$, $\sigma_{\mathbf{J}_{\mathrm{c}}}$	Average and standard deviation of the cost functional $\mathbf{J}_{\mathrm{c}}$ along the simulation trials
t_{c}	Computation time of the predictive controllers
$\overline{t_{\mathrm{c}}}$, $\sigma_{t_{\mathrm{c}}}$	Average and standard deviation t_{c}

CONTENTS

1 INTRODUCTION

Robots are becoming popular day after day. Wheeled robots have been widely used due to their simple structure, energy efficiency, locomotion speed, and fabrication cost (CAMPION; CHUNG, 2008). However, wheeled robots' mobility is in general limited to hard flat terrains without gasps, steps, or obstacles. Legged locomotion is characterized by a series of contact points between the robot's foot and the ground (SIEGWART, 2004), resulting in an improved locomotion on rough terrains. This fact is observable in nature: humans and animals can reach places out of reach of any wheel-based system (REMY, 2011), which is illustrated in Figure 1.

Figure 1 – Wheeled and legged robots' locomotion on different terrain conditions.



Source: (REMY, 2011).

The second advantage of legs is that they provide an active suspension that decouples the body trajectory from the foot trajectory (RAIBERT, 1986). That way, the trunk can travel despite the variations in the terrain.

However, legged robots have more complex power and mechanical system when compared to wheeled robots. Accordingly, several parts of the system need to be coordinated: foothold selection, planning and control of the joints, legs, and trunk.

Controlling legged robots on rough terrains is especially difficult. Given a desired motion task, which may not be feasible, the controller tries to execute it while keeping the robot in equilibrium. A control strategy that has been gaining popularity in this context for legged robots is the Model Predictive Control (MPC) (SLEIMAN et al., 2021; CARLO et al., 2018; VILLARREAL, 2020). MPC has the possibility to impose constraints, consider future support contact locations (footholds), periods of underactuation, and future changes in the terrain.

To formulate an optimization problem computationally feasible, these strategies usually consider a simplified mathematical model of the system based on some assumptions, e.g., the effect of the legs inertia on the Center of Mass (CoM) is negligible. That way, if the assumptions fail in some situations, along with measurement noise and other unmodeled dynamics, they will arise as a disturbance in the system model. Consequently, the computed control input may not be optimal, which may affect the robot's stability.

In the context of predictive control, Robust Model Predictive Control (RMPC) addresses the presence of uncertainties in the system description, considering that the uncertainty has a bounded support. Thus, RMPC approaches require constraints to be satisfied for all possible realizations of uncertainty, treating all the uncertainties and its realizations with equal importance (KOUVARITAKIS; CANNON, 2016). Accordingly, the control actions are usually computed based on the worst-case evaluation of the functional cost and constraints (MESBAH, 2016). However, uncertainty can be characterized as a random variable with a probabilistic distribution based on previous knowledge or statistical analysis of the real system. Hence, Stochastic Model Predictive Control (SMPC) has emerged to consider the presence in the system description of uncertainties characterized as random variables with a certain distribution (MESBAH, 2016).

For robotics, approaches based on SMPC are mainly focused on collision avoidance and improving the trajectory tracking, e.g., wheeled vehicles (HEWING; KABZAN; ZEILINGER, 2019; GRAY et al., 2013), aerial robots (MAMMARELLA et al., 2018; ARUL; MANOCHA, 2021), and biped robots (GAZAR et al., 2021).

Therefore, this work is a comparative study of MPC and SMPC approaches for legged locomotion on rough terrains. In addition, this work aims the following objectives

- Literature review of stochastic predictive control approaches;
- Study of control strategies for legged robots;
- Formulate a stochastic controller that considers the existence of uncertainties in the mathematical model for legged robots;
- Evaluate the effect of the angular momentum rate minimization with relation to trajectory tracking and smoothness;
- Evaluate the formulated controller in a simulation scenario with obstacles and compare with a nominal predictive controller.

1.1 OUTLINE

This work is organized as follows: Chapter 2 reviews the basic concepts related to legged locomotion, MPC, and SMPC; Chapter 3 presents the studied quadruped robot: Hydraulic Quadruped (HyQ) robot, and describes its hardware, software, and control architecture; Chapter 4 proposes the modifications in the MPC and formulates an SMPC; Chapter 5 presents the simulation results in the Gazebo Simulator; Chapter 6 concludes this work and suggests future research directions.

2 LITERATURE REVIEW

This chapter addresses some of the basic concepts related to Legged Locomotion, Model Predictive Control, and Stochastic Model Predictive Control.

2.1 LEGGED LOCOMOTION

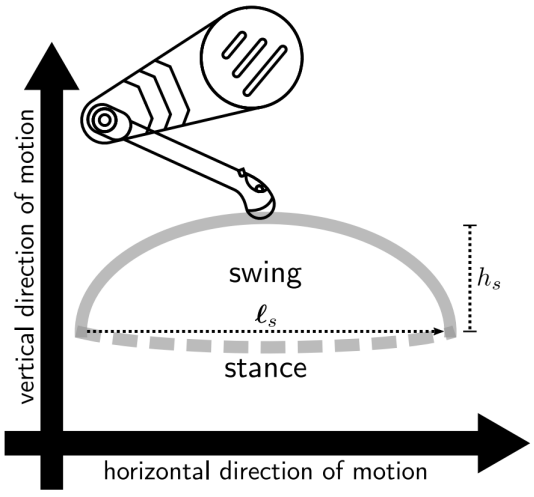
Legged locomotion is achieved by moving the leg in motion patterns, i.e., *gaits*. The early works related to legged locomotion remarks the late 19th century, when Muybridge developed a mathematical description of horses locomotion in different situations (MUYBRIDGE, 1957).

Since the work of Muybridge, some conventions have been naturally defined due to the periodicity and common pattern for the legs motion in multiple gaits. A leg during a gait cycle presents two phases:

- *Stance or Support*: foot is in contact with the ground, while the leg supports the robot weight;
- *Swing*: foot and leg are moving in the air until the next contact point.

An example of a foot trajectory is depicted in Figure 2.

Figure 2 – Example of a foot trajectory: h_s is the step height, and $\boldsymbol{\ell}_s$ is the direction and distance traveled by the foot with respect to the robot's base during stance phase.



Source: (VILLARREAL, 2020).

Gaits can be classified as *static* or *dynamic*. Static gaits require the robot to be in a stable condition during all the locomotion. However, as a trade-off, dexterity and agility are limited for this kind of gait. Dynamics gaits refer to the ones in which the body is not strictly in equilibrium during the entire locomotion, enforcing the robot to continuously take steps to prevent falling (VILLARREAL, 2020).

Among the gait patterns observed by Muybridge, three of them are important to be highlighted for this work:

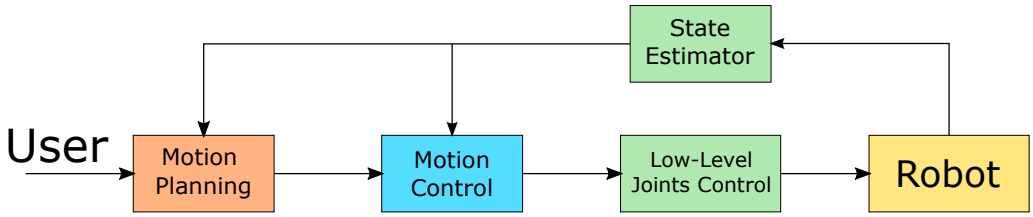
- *Walk/Crawl*: at least three feet are in Stance, and the robot present static equilibrium in the entire locomotion;
- *Trot*: diagonal pairs of legs move at the same time;
- *Gallop*: the robot presents a *ballistic* phase, when there are no legs on Stance.

In this work, all the studies and simulations are developed for *trot*, which is a dynamic gait where the diagonal pairs of legs (left-front/right-hind and right-front/left-hind) move at the same time. Note that, the techniques in this work can be extended for other gait types.

2.2 CONTROL STRATEGIES FOR LEGGED LOCOMOTION

Especially on rough terrains, legged locomotion is a complex task. Several parts of the system and of the locomotion need to be coordinated, e.g., foothold selection, swing leg trajectory, trunk attitude. A general control architecture for legged robots is presented in Figure 3 and explained accordingly in the text below.

Figure 3 – Example of control architecture for legged robots.



Source: the author, 2022.

Based on a motion objective specified by a user or a supervision system, the Motion Planning is responsible to plan the CoM or the trunk trajectory, and the legs trajectory, which includes the selection of the future contact positions. Motion Planning can be performed considering the coupling between the trunk and legs (MASTALLI et al., 2020; DAI; VALENZUELA; TEDRAKE, 2014; FANKHAUSER et al., 2018), or not (VILLARREAL et al., 2020).

The motion planer may consider only general aspects, e.g., kinematics limits, and it may run in a long-time loop. Therefore, a second level is included in the architecture to ensure that the robot will maintain the equilibrium and to include a feedback feature. That way, Motion Control computes adequate torque values in order to accomplish the desired task or a motion plan. There are different approaches for Motion Control, it can consider the full dynamics of the robot (SLEIMAN et al., 2021), or in an uncoupled approach between the trunk and leg joints (BARASUOL et al., 2013; CARLO et al., 2018).

The low-level joint control interacts directly with the actuators, which can be hydraulic (SEMINI et al., 2011), electric (BLEDT et al., 2018; KATZ; CARLO; KIM, 2019; HUTTER et al., 2016), or a hybrid solution (SEMINI et al., 2019; BARASUOL et al., 2018). The controller receives as input joint variables (position, velocity, acceleration) or torques, and the output is the

actuator signal (valve position, motor voltage, flow signal), which can be implemented using a joint position control or a force control using compliance control or inverse dynamics.

2.3 MODEL PREDICTIVE CONTROL

MPC is a class of controllers that uses a process model to predict its behavior and compute a control input according to a desired objective functional at each control step, following the receding horizon strategy (CAMACHO; ALBA, 2007).

Based on the process model, a future sequence of n_y time steps of the system is predicted by optimizing a sequence of n_u control inputs, denominated prediction and control horizon, respectively. Then, only the first sample of the control sequence is applied, and in the next control step, the entire procedure is repeated. This strategy is denominated *receding horizon* (WANG, 2009), and provides a feedback feature to MPC.

To design a linear MPC, consider a discrete-time linear invariant system

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \mathbf{A}\boldsymbol{x}_k + \mathbf{B}\boldsymbol{u}_k, \\ \boldsymbol{y}_k &= \mathbf{C}\boldsymbol{x}_k, \end{aligned} \tag{1}$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ is the system state, $\boldsymbol{u}_k \in \mathbb{R}^m$ is the control inputs, $\boldsymbol{y}_k \in \mathbb{R}^p$ is the system outputs, $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the state matrix, $\mathbf{B} \in \mathbb{R}^{n\times m}$ is the input matrix, and $\mathbf{C} \in \mathbb{R}^{p\times n}$ is the output matrix.

The model given by Equation (1) can be rewritten in a matrix form to represent the states along the horizons

$$\underbrace{\begin{bmatrix} \boldsymbol{x}_{k+1|k} \\ \boldsymbol{x}_{k+2|k} \\ \vdots \\ \vdots \\ \boldsymbol{x}_{k+n_y|k} \end{bmatrix}}_{\bar{\boldsymbol{x}}_k} = \underbrace{\begin{bmatrix} \mathbf{A} \\ \mathbf{A}^2 \\ \vdots \\ \vdots \\ \mathbf{A}^{n_y} \end{bmatrix}}_{\mathbf{S}^x} \boldsymbol{x}[k] + \underbrace{\begin{bmatrix} \mathbf{B} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{AB} & \mathbf{B} & \ddots & \dots & \mathbf{0} \\ \mathbf{A}^2\mathbf{B} & \mathbf{AB} & \ddots & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{A}^{n_y-1}\mathbf{B} & \mathbf{A}^{n_y-2}\mathbf{B} & \dots & \dots & \mathbf{A}^{n_y-n_u}\mathbf{B} \end{bmatrix}}_{\mathbf{S}^u} \underbrace{\begin{bmatrix} \boldsymbol{u}_{k|k} \\ \boldsymbol{u}_{k+1|k} \\ \vdots \\ \vdots \\ \boldsymbol{u}_{k+n_u-1|k} \end{bmatrix}}_{\bar{\boldsymbol{u}}_k}, \tag{2}$$

$$\underbrace{\begin{bmatrix} \boldsymbol{y}_{k+1|k} \\ \boldsymbol{y}_{k+2|k} \\ \vdots \\ \boldsymbol{y}_{k+n_y|k} \end{bmatrix}}_{\bar{\boldsymbol{y}}_k} = \underbrace{\begin{bmatrix} \mathbf{C} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{C} & \ddots & \mathbf{0} \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{C} \end{bmatrix}}_{\bar{\mathbf{C}}} \bar{\boldsymbol{x}}_k, \tag{3}$$

where $\boldsymbol{x}_{k+i|k}$, $\boldsymbol{u}_{k+i|k}$, and $\boldsymbol{y}_{k+i|k}$ are the state, control input, and output at moment $k+i$ obtained at k, respectively.

Consider a cost functional to regulate the system to the origin

$$\mathbf{J}_{\text{MPC}} = \sum_{i=1}^{n_y-1} \|\boldsymbol{x}_{k+i|k}\|^2_{\mathbf{Q}} + \sum_{i=0}^{n_u-1} \|\boldsymbol{u}_{k+i|k}\|^2_{\mathbf{R}} + \|\boldsymbol{x}_{k+n_y|k}\|^2_{\mathbf{P}}, \tag{4}$$

where $\mathbf{Q} \in \mathbb{R}^{n \times n} \geq 0$, $\mathbf{P} \in \mathbb{R}^{n \times n} \geq 0$, and $\mathbf{R} \in \mathbb{R}^{m \times m} > 0$ are weight matrices for the state, terminal state, and input control, respectively; and $\|\mathbf{v}_s\|^2_{\mathbf{Q}_v}$ is the 2-norm of the variable $\mathbf{v}_s$ weighted by $\mathbf{Q}_v$. For ease of notation, the $k+i|k$ index will be only denoted as $k+i$.

Any real system has some limitations or constraints, e.g., actuator saturation, physical, and safety limitations. A constraint in the states, control inputs, and outputs can be written by a linear combination of the state and control input, which is given by

$$\mathbf{G}_c \boldsymbol{u} + \mathbf{E}_c \boldsymbol{x}_k \leq \mathbf{h}_c. \tag{5}$$

Therefore, considering the model given by Equation (1), an optimization problem is formulated to compute a control input sequence to minimize the selected objective functional respecting the constraints, as follows

$$\begin{aligned} \min_{\bar{\boldsymbol{u}}_k} \quad & \mathbf{J}_{\text{MPC}}, \\ s.t. \quad & \boldsymbol{x}_{k+i+1} = \mathbf{A}\boldsymbol{x}_{k+i} + \mathbf{B}\boldsymbol{u}_{k+i}, \\ & \mathbf{G}_c \boldsymbol{u}_{k+i} + \mathbf{E}_c \boldsymbol{x}_{k+i} \leq \mathbf{h}_c. \end{aligned} \tag{6}$$

At each control step, it is necessary to update $\boldsymbol{x}_k$ and solve the optimization problem to determine a new optimal control sequence $\bar{\boldsymbol{u}}_k$. Then, the first control input $\boldsymbol{u}_k$ is applied to the system, and in the next control step all the optimization is done again, following the receding horizon strategy.

2.4 STOCHASTIC MODEL PREDICTIVE CONTROL

The system representation in Equation (1) is considered deterministic and *nominal*, allowing to formulate a standard MPC.

A common way to consider uncertainties in the system description given by Equation (1) is through the following general form

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \mathbf{A}_k \boldsymbol{x}_k + \mathbf{B}_k \boldsymbol{u}_k + \mathbf{D}\boldsymbol{w}_k, \\ \boldsymbol{y}_k &= \mathbf{C}\boldsymbol{x}_k + \boldsymbol{v}_k, \end{aligned} \tag{7}$$

where $\boldsymbol{w}_k \in \mathbb{R}^{n_w}$ and $\boldsymbol{v}_k \in \mathbb{R}^{n_v}$ denote the process and measurement noise, respectively, resulting in additive uncertainties. The matrix $\mathbf{D}$ maps the system states that are influenced by the uncertainties, $\boldsymbol{w}_k$ captures the effects of unknown and unmodeled dynamics, and disturbances on the evolution of the system. Both $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are assumed independent and identically distributed (i.i.d.) variables with known probability distributions. Further, the matrices $\mathbf{A}_k$ and $\mathbf{B}_k$ may assume different values in a compact polytopic set Ω, given by the convex hull:

$$\Omega = Co\{[\mathbf{A}_1, \mathbf{B}_1], [\mathbf{A}_2, \mathbf{B}_2], \ldots, [\mathbf{A}_L, \mathbf{B}_L]\}. \tag{8}$$

implying in multiplicative uncertainties.

As presented in (KOUVARITAKIS; CANNON, 2016), in the controller design, different approaches aim to consider the parametric uncertainties in the system matrices or disturbance signal to improve the controller robustness.

This work focuses only on additive uncertainties, i.e., there is no uncertainty in the system matrices; and there is only uncertainty in the system state evolution. That way, the following uncertain linear system is obtained

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \mathbf{A}\boldsymbol{x}_k + \mathbf{B}\boldsymbol{u}_k + \mathbf{D}\boldsymbol{w}_k, \\ \boldsymbol{y}_k &= \mathbf{C}\boldsymbol{x}_k. \end{aligned} \tag{9}$$

RMPC considers that the uncertainty in Equation (9) has bounded support, i.e., $|\boldsymbol{w}| \leq \boldsymbol{w}_{\max}$. Therefore, the controllers are designed to consider the worst-case situation.

Different from RMPC, SMPC considers the uncertainty as a stochastic variable, with known distribution and in general as an unbounded signal.

As Mesbah (2016), SMPC are predictive controllers that systematically incorporates probabilistic descriptions of the system in the formulation of the optimization problems. That way, constraints in the state, control, and output can be reformulated to be satisfied at a minimum probability level, a concept defined as *chance-constraint*.

A common approach for both RMPC and SMPC is the decomposition of the state $\boldsymbol{x}_k$ in a deterministic component $\boldsymbol{z}_k$, that describes the nominal system behavior (equivalent to the model from nominal MPC), and a random component $\boldsymbol{e}_k$, i.e., $\mathbb{E}[\boldsymbol{e}_k] = 0$, related to the stochastic components of the system. To attenuate the propagation of uncertainties and increase the controller robustness, the control input $\boldsymbol{u}_k$ is redefined as an affine function, with a feedback gain $\mathbf{K}$ associated with $\boldsymbol{x}_k$ and an affine component $\boldsymbol{c}_k$. That way, the following expressions are obtained:

$$\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{z}_k + \boldsymbol{e}_k, \\ \boldsymbol{u}_k &= \mathbf{K}\boldsymbol{x}_k + \boldsymbol{c}_k, \\ \boldsymbol{z}_{k+1} &= \boldsymbol{\Phi}\boldsymbol{z}_k + \boldsymbol{c}_k, \\ \boldsymbol{e}_{k+1} &= \boldsymbol{\Phi}\boldsymbol{e}_k + \mathbf{D}\boldsymbol{w}_k, \end{aligned} \tag{10}$$

where $\mathbf{K}$ can be obtained previously, e.g., a Linear Quadratic Regulator (LQR) gain, or it can be considered as a decision variable in the optimization problem, and $\boldsymbol{\Phi} = \mathbf{A} + \mathbf{B}\mathbf{K}$ defines the closed-loop dynamics. Note that due to the $\boldsymbol{u}_k$ affine redefinition, $\boldsymbol{c}_k$ is a decision variable instead of $\boldsymbol{u}_k$.

The state decomposition from Equation (10) can be rewritten along the horizons

$$\underbrace{\begin{bmatrix} z_{k+1} \\ z_{k+2} \\ \vdots \\ \vdots \\ z_{k+\mathrm{n_y}} \end{bmatrix}}_{\bar{z}_k} = \underbrace{\begin{bmatrix} \boldsymbol{\Phi} \\ \boldsymbol{\Phi}^2 \\ \vdots \\ \vdots \\ \boldsymbol{\Phi}^{\mathrm{n_y}} \end{bmatrix}}_{\mathbf{S}^{\Phi}} z_k + \underbrace{\begin{bmatrix} \mathbf{B} & \mathbf{0} & \dots & \mathbf{0} \\ \boldsymbol{\Phi}\mathbf{B} & \mathbf{B} & \ddots & \mathbf{0} \\ \boldsymbol{\Phi}^2\mathbf{B} & \boldsymbol{\Phi}\mathbf{B} & \ddots & \mathbf{0} \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \boldsymbol{\Phi}^{\mathrm{n_y}-1}\mathbf{B} & \boldsymbol{\Phi}^{\mathrm{n_y}-2}\mathbf{B} & \dots & \boldsymbol{\Phi}^{\mathrm{n_y}-\mathrm{n_u}}\mathbf{B} \end{bmatrix}}_{\mathbf{S}^{c}} \underbrace{\begin{bmatrix} c_k \\ c_{k+1} \\ \vdots \\ \vdots \\ c_{k+\mathrm{n_u}-1} \end{bmatrix}}_{\bar{c}_k}, \tag{11}$$

and

$$\underbrace{\begin{bmatrix} e_{k+1} \\ e_{k+2} \\ \vdots \\ \vdots \\ e_{k+\mathrm{n_y}} \end{bmatrix}}_{\bar{e}_k} = \underbrace{\begin{bmatrix} \boldsymbol{\Phi} \\ \boldsymbol{\Phi}^2 \\ \vdots \\ \vdots \\ \boldsymbol{\Phi}^{\mathrm{n_y}} \end{bmatrix}}_{\mathbf{S}^{\Phi}} e_k + \underbrace{\begin{bmatrix} \mathbf{D} & \mathbf{0} & \dots & \mathbf{0} \\ \boldsymbol{\Phi}\mathbf{D} & \mathbf{D} & \ddots & \mathbf{0} \\ \boldsymbol{\Phi}^2\mathbf{D} & \boldsymbol{\Phi}\mathbf{D} & \ddots & \mathbf{0} \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \boldsymbol{\Phi}^{\mathrm{n_y}-1}\mathbf{D} & \boldsymbol{\Phi}^{\mathrm{n_y}-2}\mathbf{D} & \dots & \mathbf{D} \end{bmatrix}}_{\mathbf{S}^{w}} \underbrace{\begin{bmatrix} w_k \\ w_{k+1} \\ \vdots \\ \vdots \\ w_{k+\mathrm{n_y}-1} \end{bmatrix}}_{\bar{w}_k}, \tag{12}$$

Among the different SMPC strategies, there are some common approaches related to the cost functional, constraints treatment, and stability analysis. Those features will be presented in the sequence.

2.4.1 Cost Functional

Since the state and, possibly, the control input are considered random variables in the SMPC, the nominal cost $\mathbf{J}_{\mathrm{MPC}}$ is influenced by the uncertainties. Accordingly, to properly minimize the desired objective function, some considerations are necessary. In this section, some of the most common approaches are presented following the work (FARINA; GIULIONI; SCATTOLINI, 2016).

The most common approach is to consider the expected value of the original cost functional $\mathbf{J}_{\mathrm{MPC}}$, given by Equation (4), as follows

$$\mathbf{J}_{\mathrm{SMPC}} = \mathbb{E}[\mathbf{J}_{\mathrm{MPC}}] = \mathbb{E}\left[\sum_{i=1}^{\mathrm{n_y}-1} \|x_{k+i}\|^2_{\mathbf{Q}} + \sum_{i=0}^{\mathrm{n_u}-1} \|u_{k+i}\|^2_{\mathbf{R}} + \|x_{k+\mathrm{n_y}}\|^2_{\mathbf{P}}\right], \tag{13}$$

which can be rewritten using the system state decomposition given by Equation (10).

Sample-Based approaches characterize stochastic system dynamics based on a finite set of random realization of the uncertainties (MESBAH, 2016), e.g., estimation of the future values of the system noise w_k. Consequently, there is more freedom to select the cost functional. The straightforward way is to consider w_k along the prediction horizon in the system state and use the nominal cost functional $\mathbf{J}_{\mathrm{MPC}}$. Also, it is possible to minimize the average cost of $\mathrm{n_s}$ noise samples realizations (SCHILDBACH et al., 2014)

$$\mathbf{J}_{\mathrm{AVG}} = \frac{1}{\mathrm{n_s}} \sum_{j=1}^{\mathrm{n_s}} \mathbf{J}^{[j]}_{\mathrm{MPC}}, \tag{14}$$

or to minimize the maximum cost among multiple samples (CALAFIORE; FAGIANO, 2013)

$$\mathbf{J}_{\max} = \max_{j=1,\ldots,\mathrm{n_s}} \mathbf{J}_{\mathrm{MPC}}^{[j]}. \tag{15}$$

2.4.2 Constraints

Due to practical aspects, constraints in the control inputs are usually considered *hard*, i.e., holding the same formulation as Equation (5) (FARINA; GIULIONI; SCATTOLINI, 2016). However, due to the presence of uncertainty in the evolution of $\boldsymbol{x}_k$, it is necessary to include a tightened term λ to compensate for the influence of $\boldsymbol{w}_k$ in the constraints related to the states.

Consider a state constraint in the form

$$\mathbf{G}_{\boldsymbol{x}}\boldsymbol{x}_{k+i} \leq \mathbf{h}_{\boldsymbol{x}}. \tag{16}$$

Based on the state decomposition given by Equation (10), it is reformulated as

$$\mathbf{G}_{\boldsymbol{x}}\boldsymbol{x}_{k+i} = \mathbf{G}_{\boldsymbol{x}}(\boldsymbol{z}_{k+i} + \boldsymbol{e}_{k+i}) \leq \mathbf{h}_{\boldsymbol{x}} - \lambda. \tag{17}$$

In RMPC design, it is common to define λ based on the worst-case of $\boldsymbol{w}_k$

$$\lambda_i = \max_{\boldsymbol{w}_k \in \mathscr{W}} (\bar{\mathbf{G}}_{\boldsymbol{x}}\mathbf{S}^w)_i \boldsymbol{w}_k, \tag{18}$$

where $\bar{\mathbf{G}}_{\boldsymbol{x}}$ is the diagonal composition of $\mathrm{n_y}$ matrices $\mathbf{G}_{\boldsymbol{x}}$, i.e., $\bar{\mathbf{G}}_{\boldsymbol{x}} = \mathrm{diag}(\mathbf{G}_{\boldsymbol{x}}, \ldots, \mathbf{G}_{\boldsymbol{x}})$; and the subscript i denotes the ith row of a matrix. This strategy is verified naturally conservative, since in practical situations, when operating close to the constraints tends to lead to a lower value of the objective function, and the worst-case may never happen. Also, it demands $\boldsymbol{w}_k$ to be bounded by a polytope $\mathscr{W}$, i.e., $\mathscr{W} = \{\boldsymbol{w} : |\boldsymbol{w}| \leq \boldsymbol{w}_{\max}\}$.

In stochastic controllers, state constraints are reformulated to satisfy a minimum probability level

$$\mathscr{P}\{\mathbf{G}_{\boldsymbol{x}}(\boldsymbol{z}_{k+i} + \boldsymbol{e}_{k+i}) \leq \mathbf{h}_{\boldsymbol{x}}\} \geq 1 - \mathrm{p}, \tag{19}$$

where $\mathscr{P}\{\mathscr{X}\}$ denotes the probability of the event $\mathscr{X}$ occurs, and p is the maximum acceptable probability level of the constraints to be violated.

Consider $\boldsymbol{w}_k$ an unbounded signal with Gaussian Distribution, $\mathbb{E}[\boldsymbol{w}_k] = 0$, and variance matrix $\Sigma^{\boldsymbol{w}}$. Then, the tightened term λ_i can be exactly computed as

$$\lambda_i(1-\mathrm{p}) = \sqrt{(\bar{\mathbf{G}}_{\boldsymbol{x}}\mathbf{S}^w)_i \bar{\Sigma^{\boldsymbol{w}}} (\bar{\mathbf{G}}_{\boldsymbol{x}}\mathbf{S}^w)_i^{\mathrm{T}}}\, f_{\mathrm{cdf}}^{-1}(1-\mathrm{p}), \tag{20}$$

where $\bar{\Sigma^{\boldsymbol{w}}}$ is the diagonal composition of $\mathrm{n_y}$ matrices $\Sigma^{\boldsymbol{w}}$, i.e, $\bar{\Sigma^{\boldsymbol{w}}} = \mathrm{diag}(\Sigma^{\boldsymbol{w}}, \ldots, \Sigma^{\boldsymbol{w}})$, and f_{cdf} is the cumulative distribution function of a Gaussian variable with zero mean and unitary variance (BLACKMORE; ONO; WILLIAMS, 2011; BOYD; VANDENBERGHE, 2004).

If the distribution is unknown, it is possible to approximate by the Chebyshev Inequality

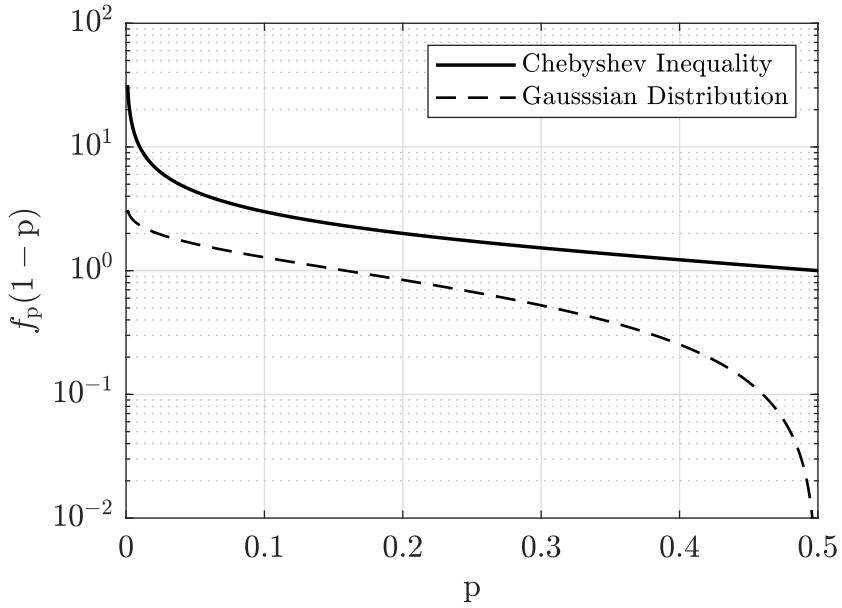
$$\lambda_i(1-\mathrm{p}) = \sqrt{(\bar{\mathbf{G}}_{\boldsymbol{x}}\mathbf{S}^w)_i \bar{\Sigma^{\boldsymbol{w}}} (\bar{\mathbf{G}}_{\boldsymbol{x}}\mathbf{S}^w)_i^{\mathrm{T}}} \sqrt{\frac{(1-\mathrm{p})}{\mathrm{p}}}. \tag{21}$$

Note that both Equations (20) and (21) present the general form

$$\lambda_i(1-\mathrm{p}) = \sqrt{(\bar{\mathbf{G}}_{\boldsymbol{x}}\mathbf{S}^w)_i \Sigma^{\bar{\boldsymbol{w}}} (\bar{\mathbf{G}}_{\boldsymbol{x}}\mathbf{S}^w)_i^{\mathrm{T}}}\, f_{\mathrm{p}}(1-\mathrm{p}). \tag{22}$$

where $f_{\mathrm{p}}(1-\mathrm{p})$ depends on the distribution of $\boldsymbol{w}_k$. A comparison between both cases for f_{p} is presented in Figure 4, where it is possible to verify that the Chebyshev Inequality is more conservative.

Figure 4 – Comparison of function f_{p} for a Chebyshev Inequality and Gaussian Distribution.



Source: Adapted from (FARINA; GIULIONI; SCATTOLINI, 2016).

In less conservative situations, instead of considering an acceptable probability of constraint violation, it can be defined an average number of acceptable constraints violations (FARINA; GIULIONI; SCATTOLINI, 2016).

In Sample-Based approaches, due to the realization of the noise along the prediction horizon, it is possible to directly evaluate the constraints and explicitly compute the tightened term as follows

$$\lambda_i(1-\mathrm{p}) = \arg\min_{\lambda_i^*} \lambda_i^* \;\; s.t. \;\; \mathscr{P}\{(\bar{\mathbf{G}}_{\boldsymbol{x}}\mathbf{S}^w)_i \boldsymbol{w}_k \leq \lambda_i^*\} = 1-\mathrm{p}. \tag{23}$$

Moreover, if it is set $\lambda(1)$, i.e., $\mathrm{p}=0$, then it is desired that the constraints are never violated, which is mathematically unfeasible due to the unbounded consideration about $\boldsymbol{w}_k$. In this case, it is possible to truncate $\boldsymbol{w}_k$, and then is it obtained again the constraint tightened given by Equation (18) for RMPC.

If the system's description or the noise's data are not updated during the controller execution, the tightened terms for the chance-constraints along the horizon, i.e., $\bar{\lambda}$, can be computed offline, previously to runtime. If any of the variables related to λ computation are updated during runtime, $\bar{\lambda}$ needs to be recalculated, increasing the computational cost.

For output constraints, the same procedures for state constraints are applicable.

2.4.3 Stability and Feasibility

As presented by (FARINA; GIULIONI; SCATTOLINI, 2016), general guarantees of recursive feasibility are still open. This occurs especially if $\boldsymbol{w}_k$ is considered unbounded. Even with a small probability, the noise can assume a value large enough that computing a control input sequence that respects the constraints and stabilizes the system is unfeasible.

Considering the uncertainty as a random variable with finitely supported distribution, (CANNON; KOUVARITAKIS; NG, 2009) handle the constraints based on a sequence of tubes associated to a sequence of confidence levels on the predicted future plant state around the nominal state trajectories. Hence, to ensure satisfaction of hard and soft constraints, a set of one-step-ahead linear constraints are formulated by imposing bounds on the probability of constraint violation at each prediction step along an infinite prediction horizon.

Additionally, to guarantee recursive feasibility it is possible to modify the constraints to consider that $\mathbb{E}[\boldsymbol{x}_{k|k}] = \mathbb{E}[\boldsymbol{x}_{k|k-1}]$ along the horizon (FARINA et al., 2015); or directly impose it in the *scenario* approach that will be present in sequence.

2.4.4 Stochastic Predictive Control Approaches

Based on the classification presented in (MESBAH, 2016), some approaches for SMPC are presented.

2.4.4.1 Stochastic Tube MPC

The concept of tubes consists to formulate constraints in the optimization problem to ensure the uncertain future state and control input trajectories to lie in a sequence of sets, known as tubes, which can be used to guarantee recursive stability (KOUVARITAKIS; CANNON, 2016). Tubes are already employed in RMPC strategies (LANGSON et al., 2004; MAYNE; SERON; RAKOVIć, 2005).

In general, the expected value of the cost functional, Equation (13), is employed using the system state decomposition given by Equation (10), where the gain $\mathbf{K}_k$ is constant or at least known previously. Accordingly, the reformulated cost functional depends only on $\boldsymbol{z}_k$ and $\boldsymbol{c}_k$.

Depending on the available information about the disturbance distribution, constraints tightened are computed by Equation (20) or Equation (21).

The tubes and tightened constraints can be computed offline. Therefore, the optimization problem to be solved at each control step usually has the same complexity as the nominal MPC.

2.4.4.2 Affine Parameterization of the Control Policy

In Stochastic Tube MPC, the control input is defined by a feedback gain, $\mathbf{K}_k$, known previously to the optimization, while the affine term sequence $\{\boldsymbol{c}_k, \ldots, \boldsymbol{c}_{\mathrm{n_u}-1}\}$ is the decision variables. In order to include more degrees of freedom, $\mathbf{K}_k$ can also be considered as a decision

variable on each horizon step. Some approaches, e.g., (FARINA; GIULIONI; SCATTOLINI, 2016), reformulate the control law based on the *feedforward* concept, as follows:

$$\boldsymbol{u}_k = \boldsymbol{u}_{ff_k} + \mathbf{K}_k(\boldsymbol{z}_k - \boldsymbol{x}_k). \tag{24}$$

However, this control law leads to a nonconvex optimization problem, which demands its reformulation and new assumptions, resulting in a more conservative problem.

Although the values of $\boldsymbol{w}_{k+i}$ are not available at instant k, they will be known to the controller in the next ith control step. Similar to strategies using Youla parameterization (HESSEM; BOSGRA, 2002), it is defined an affine disturbance feedforward (OLDEWURTEL; JONES; MORARI, 2008; ZHANG et al., 2013)

$$\boldsymbol{u}_k = \mathbf{L}_k \boldsymbol{w}_k + \boldsymbol{n}_k, \tag{25}$$

where $\mathbf{L}_k$ is a feedback gain related to $\boldsymbol{w}_k$, and $\boldsymbol{n}_k$ is an affine term. In some special cases, e.g., number of constraints and features of the state-space, the influence of $\boldsymbol{w}_k$ can be completely attenuated, which makes possible the use of a constant matrix $\mathbf{L}$ that depends only on the nominal description of the system. However, in general, this is not true, which demands the matrix $\mathbf{L}_k$ to be also a decision variable.

In both cases of affine parameterization, the cost functional and constraints treatment are the same as the tube approach, demanding only the modifications due to the different control law.

2.4.4.3 Scenario-Based SMPC

Instead of the general Sample-Based approaches that obtain a sequence of noise realization with a sampling algorithm, e.g., polynomial chaos (MESBAH et al., 2014) and Markov chain (BLACKMORE et al., 2010), the Scenario-Based strategy combines random sampling with robust convex optimization to evaluate a finite number of scenarios n_{sc} given the available information about uncertainty and probability (CAMPI; GARATTI; PRANDINI, 2009).

The possibility to obtain samples of the uncertainty, and to use its records or known realizations, results in a fewer number of assumptions about the distribution and noise types (FARINA; GIULIONI; SCATTOLINI, 2016). The scenarios tend to be more realistic, and consequently, less conservative due to the use of more information about the system.

As the main drawback, the computational cost may prevents its use if the available computational power is not sufficient to solve the calculation procedures in the available time frame. Moreover, another challenge is to select the proper n_{sc} to be evaluated in order to guarantee stability, an acceptable number of constraints violations, and a computational cost adequate to the real application (SCHILDBACH et al., 2014).

2.5 CHAPTER SUMMARY

In this chapter, the fundamental concepts related to MPC and SMPC were presented. From the presented SMPC approaches, the Stochastic Tube MPC is selected to be used due to its

low complexity and low impact on the computational cost.

A brief description of Legged Locomotion and the Control Strategies for this task were presented. The presented control architecture is further extended in the next chapter for the HyQ robot.

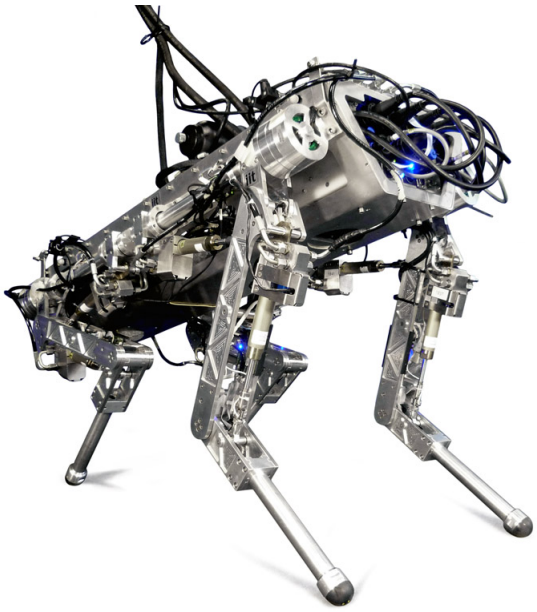
3 SYSTEM OVERVIEW

This chapter describes the hardware, software, and control architecture of the quadruped robot used in this work.

3.1 HYQ

HyQ is a hydraulically actuated quadruped robot developed by the Dynamic Legged Systems lab (DLS) at the Istituto Italiano di Tecnologia (IIT), Figure 5. HyQ has been designed to perform highly dynamic motions, e.g., trotting and jumping, and to navigate on rough terrains. The robot has about 100 kg whose dimensions with fully-stretched legs are (1.0m × 0.5m × 0.98m) (length × width × height) (SEMINI et al., 2011).

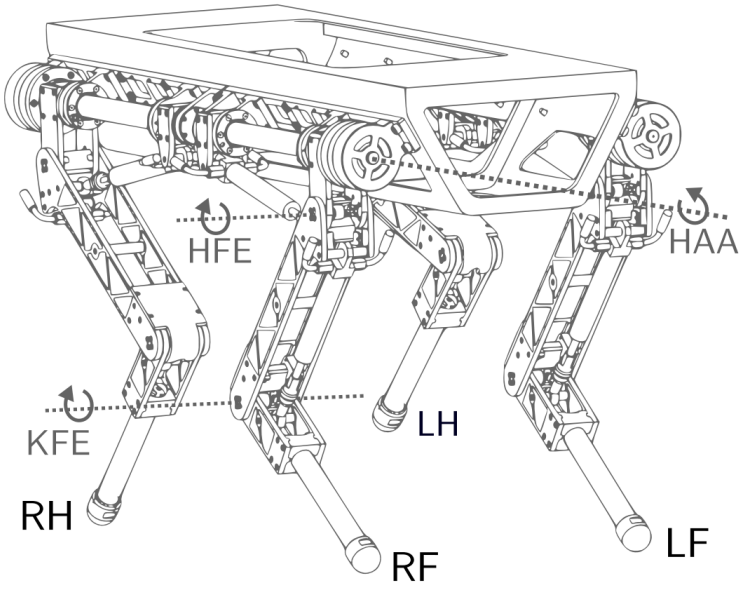
Figure 5 – HyQ robot from Istituto Italiano di Tecnologia.



Source: (IIT, 2021).

Each of the four legs weighs 9 kg and has three Degrees of Freedom (DOF): Hip Abduction/Adduction (HAA), Hip Flexion/Extension (HFE), and Knee Flexion/Extension (KFE). In Figure 6, it is presented a description of the robot's parts.

Figure 6 – Description of the legs and joints of the HyQ robot. The legs are identified as Left-Front (LF), Right-Front (RF), Left-Hind (LH), and Right-Hind (RH).



Source: (VILLARREAL, 2020).

3.2 ACTUATORS AND SENSORS

The four HAA joints are actuated by hydraulic rotary motors with torque sensors. HFE and KFE joints are actuated by hydraulic linear cylinders. Joint torque is obtained from the force applied by the hydraulic cylinder using load cells. All 12 joints are equipped with absolute and relative encoders.

To estimate the pose and velocity of the trunk with respect to the world, an Inertial Measurement Unit (IMU) is used, KVH 1775 model. Exteroceptive sensors are also used: a depth sensor for mapping, ASUS Xtion RGB-3D; and a Lidar for Visual Odometry (VO). The sensors are synchronized by an EtherCAT network, with a maximum latency of 1 ms. A detailed description of the sensors, state estimation scheme, and mapping is available in (NOBILI et al., 2017; VILLARREAL et al., 2019).

3.3 SOFTWARE ARCHITECTURE

HyQ has two onboard computers, a *Control-PC* and a *Vision-PC*. The first one has a Pentium i5 with Real-Time (RT) Linux (Xenomai) patch. It runs most of the tasks, e.g., low-level joints control, state estimator, motion control, motion planning, and supervision tasks. The second computer has a Pentium i5 with Linux regular kernel. The Vision-PC collects the exteroceptive sensors data, computes VO and the elevation map of the surrounding of the robot. Further, Robot Operating System (ROS) is used for interprocess communication.

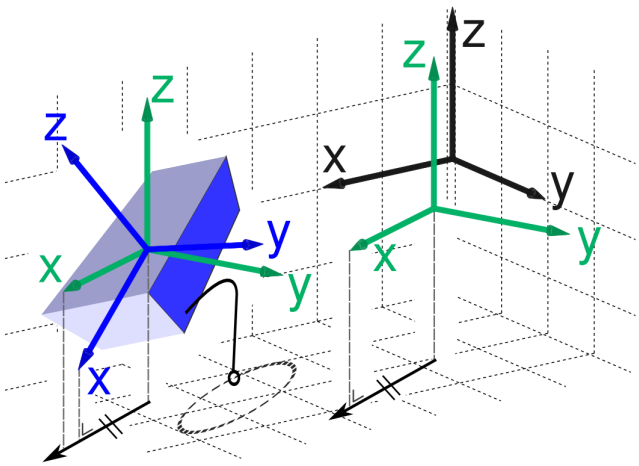
3.4 CONTROL ARCHITECTURE

In this section, the control architecture of the HyQ robot is presented. The same control architecture used in (VILLARREAL, 2020) is employed, which is detailed following the general architecture exemplified in Figure 3.

3.4.1 Reactive Controller Framework

Reactive Controller Framework (RCF) is a framework for robust quadrupedal locomotion on rough terrain (BARASUOL et al., 2013). In RCF, both Motion Control and Motion Planning are performed assuming the legs and trunk decoupled. This decoupling between legs and trunk is performed due to the definition of the reference frame known as the *Horizontal Frame*. The Horizontal Frame coincides with the robot *xy* plane, and it is orthogonal to the gravity vector, with the same yaw orientation of the robot, as depicted in Figure 7. Selecting the horizontal frame as the coordinate frame for the foot planning attenuates the influence of the trunk attitude. Therefore, the foot trajectories are planned assuming they are independent of the trunk attitude.

Figure 7 – Horizontal reference frame (green), robot frame (blue), robot trunk (blue parallelepiped), and world reference frame (black).



Source: (BARASUOL et al., 2013).

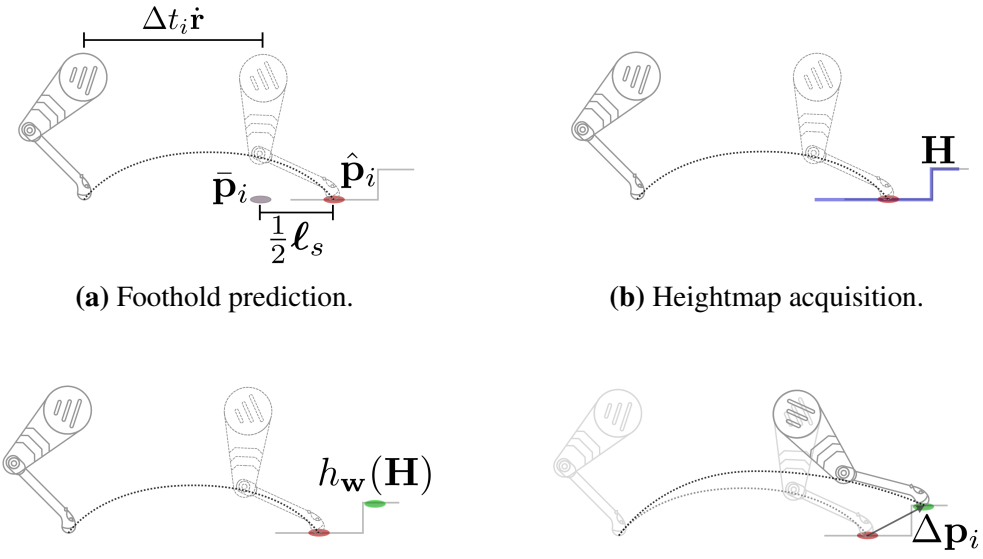
3.4.2 Swing Leg Trajectory Generation

The foot performs ellipsoidal trajectories based on the gait parameters, e.g., D_f, f_{step}, $\boldsymbol{\ell}_s$, and the user-defined forward velocity $\mathbf{V}_f$. To generate the trajectory, a cubic spline is computed for each half of the ellipse. The curve is parametrized based on the gait parameters, and it is obtained with respect to the horizontal frame.

3.4.3 Foothold Selection

To improve the dynamic locomotion on difficult terrains, Villarreal et al. (2019) propose the Vision-based Foothold Adaptation (VFA) to continuously adjust the landing position of the foot based on visual feedback, Figure 8.

Figure 8 – Phases of the VFA.



(a) Foothold prediction.

(b) Heightmap acquisition.

(c) Heightmap evaluation.

(d) Trajectory adjustment.

Source: (VILLARREAL, 2020).

As presented in Figure 8a, the *nominal foothold* is predicted by approximating the foot in swing by ellipsoidal trajectories as

$$\hat{\mathbf{p}}_i = \bar{\mathbf{p}}_i + \frac{1}{2}\boldsymbol{\ell}_s + \Delta t_i \dot{\mathbf{r}}, \tag{26}$$

where $\hat{\mathbf{p}}_i \in \mathbb{R}^3$ is the nominal foothold position, $\bar{\mathbf{p}}_i \in \mathbb{R}^3$ is the center of the ellipsoidal trajectory, $\boldsymbol{\ell}_s \in \mathbb{R}^3$ is the step length vector, $\dot{\mathbf{r}}$ is the velocity of the trunk in the world frame, Δt_i is the remaining swing time given by

$$\Delta t_i = \frac{1 - \mathrm{D}_f}{f_{step}} - t_{swi}, \tag{27}$$

with D_f as the duty factor defined by the ratio between the total stance time and the total swing time, f_{step} is the step frequency, and t_{swi} is the elapsed swing time since the latest lift-off of leg i. All vector variables are given with respect to the world frame.

A heightmap $\mathbf{H}$ around the predicted nominal foothold is extracted from the elevation map constructed by the Vision-PC, as depicted in Figure 8b. In sequence, the heightmap is evaluated using a Convolutional Neural Network (CNN) trained offline, where the evaluation is represented by the function $h_{\mathbf{w}}(\mathbf{H})$, which is illustrated in Figure 8c. The CNN computes the

optimal foothold $\mathbf{p}^*$ taking into account the following criteria: uncertainty in the foot trajectory, terrain roughness, kinematic limits, foot frontal collision, leg collision, and distance to the nominal foothold.

The difference $\Delta\mathbf{p}_i$ between the optimal foothold and the nominal foothold is sent to the Leg Trajectory Controller to adapt the original foot swing trajectory, which is depicted in Figure 8d.

3.4.4 Contact Sequence

To cover the entire time horizon considered by the MPC to control the robot's CoM trajectory, which is detailed later in this chapter, it is considered the time horizon of two gait cycles for the contact sequence and the robot's CoM trajectory generator, resulting in 16 stance changes.

Considering that the gait parameters will remain the same during the next two gait cycles, Equation (26) is modified to obtain the future contact sequence at each stance change as

$$\hat{\mathbf{p}}_i[n] = \bar{\mathbf{p}}_i[n] + \frac{1}{2}\boldsymbol{\ell}_s + \Delta t[n]\dot{\mathbf{r}}, \tag{28}$$

where $\hat{\mathbf{p}}_i[n] \in \mathbb{R}^3$ is the predicted foothold of the leg i at stance change n ($n = 1, \ldots, 16$), $\bar{\mathbf{p}}_i[n] \in \mathbb{R}^3$ is the reference position on the ground determined by the elliptical trajectory of the foot, and $\Delta t[n]$ is the time to a stance to happen since instant k. All the variables are given in the world frame. Since the gait is periodic, f_{step} and D_f are assumed constant, thus $\dot{\mathbf{r}}$ is assumed constant. Then, the time vector $\Delta t[n]$ of the stance changes is directly obtained from the current remaining swing time and f_{step}.

Further, the nominal contact sequence $\hat{\mathbf{p}}_i[n]$ is evaluated by the VFA to adapt the landing position using the same procedure described in Subsection 3.4.3.

3.4.5 Robot's Center of Mass Trajectory Generator

Given the information of each stance change, the CoM position is computed at each stance change based on the step frequency f_{step} and the duty factor D_f, resulting in 16 CoM references.

As input, e.g., selected by the user or a supervision system, the forward linear velocity $\mathbf{V}_\mathrm{f} \in \mathbb{R}^2$ and a desired yaw rate $\dot{\psi}_\mathrm{d}[n]$ at each change n are available.

At current time instant k, the reference for the yaw angle is computed

$$\psi_\mathrm{d}[n] = \psi[k] + \Delta t[n]\dot{\psi}_\mathrm{d}[n], \tag{29}$$

where $\psi_\mathrm{d}[n] \in \mathbb{R}$ is the yaw angle reference at next nth stance, $\psi[k]$ is the current yaw angle, $\Delta t[n]$ is the time to the occurrence of the nth stance from the current instant k. From the forward velocity, the desired linear velocity $\dot{\mathbf{r}}_\mathrm{d}$ for each stance change is defined as follows

$$\dot{\mathbf{r}}_\mathrm{d}[n] = \begin{bmatrix} V_{\mathrm{f,x}} & V_{\mathrm{f,y}} & 0 \end{bmatrix}^\mathrm{T}. \tag{30}$$

The reference position $\mathbf{r}_\mathrm{d}[n] \in \mathbb{R}^3$ for the robot's CoM in the world frame at stance change n is defined by

$$\mathbf{r}_\mathrm{d}[n] = \mathbf{r}[k] + \mathbf{R}_\mathrm{z}(\Delta\psi[n])\Delta t[n]\dot{\mathbf{r}}_\mathrm{d}[n], \tag{31}$$

where $\mathbf{R}_\mathrm{z} \in SO(3)$ is an orthogonal rotation matrix around the z-axis, and $\Delta\psi[n] = \psi_\mathrm{d}[n] - \psi[k]$.

Using the reference position $\mathbf{r}_\mathrm{d}[n]$, the references for the roll $\phi_\mathrm{d} \in \mathbb{R}$ and the pitch $\theta_\mathrm{d} \in \mathbb{R}$ angles are computed based on an estimation of the terrain. The terrain is approximated as a plane surface, and the angles are selected to keep the robot's CoM parallel to the surface. Also, the height p_cz of the surface is estimated, and it is included in $\mathbf{r}_\mathrm{d}$ to keep the robot at a constant height from the ground. Then, the angular references of the body with respect to the world expressed in Euler angles are obtained

$$\Theta_\mathrm{d}[n] = \begin{bmatrix} \phi_\mathrm{d}[n] & \theta_\mathrm{d}[n] & \psi_\mathrm{d}[n] \end{bmatrix}^\mathrm{T}. \tag{32}$$

For the angular rates $\dot{\phi}_\mathrm{d}[n]$ and $\dot{\theta}_\mathrm{d}[n]$, the reference is selected zero. Although it is not properly defined based on the angular position reference, the reference selected as zero induces the MPC to achieve lower angular rates, which tends to deviate less the robot from the angular reference trajectory. Since keeping the robot dynamically stable has a higher priority than following a desired trajectory or maintaining the robot parallel to the terrain, this assumption is acceptable.

For the MPC, the trajectory needs to be resampled at the same MPC discretization time. Therefore, a Zero-Order Hold (ZOH) is employed, resulting in the following state vector reference:

$$\boldsymbol{x}_\mathrm{d}[k] = \begin{bmatrix} \Theta_\mathrm{d}^\mathrm{T}[k] & \mathbf{r}_\mathrm{d}^\mathrm{T}[k] & \mathbf{0} & \dot{\mathbf{r}}_\mathrm{d}^\mathrm{T}[k] \end{bmatrix}^\mathrm{T}. \tag{33}$$

3.4.6 Leg Trajectory Control

This controller computes the torque τ_fb of the swing leg's joints to reach the landing position in the desired switch time.

Given the desired foot trajectory, the desired joint states $(\mathbf{q}_\mathrm{d}, \dot{\mathbf{q}}_\mathrm{d}, \ddot{\mathbf{q}}_\mathrm{d})$ for those legs are obtained using inverse kinematics. Thus, the desired states are tracked using inverse dynamic control to compute a feedforward torque in combination with a Proportional-Derivative (PD) controller. The PD gains are computed to balance the induced resultant impedance for the legs and the tracking error.

The leg trajectory controller runs at 250 Hz, and the resulting torque is summed with the torque associated to the desired trunk motion, i.e., τ_tc. The combined value is sent to the low-level joint torque controller. More details about this controller can be obtained in Barasuol (2013).

3.4.7 Model Predictive Control

Based on the work from Carlo et al. (2018), the MPC designed to control the robot's CoM trajectory in Villarreal (2020) is described.

3.4.7.1 Prediction Model

Mechanical systems can be described by the equations of motion of rigid-body systems (FEATHERSTONE, 2008). The general equations of motion for a system of n_j-DOF with n_c frictionless point contacts are given by

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}(\mathbf{q},\dot{\mathbf{q}}) = \tau + \mathbf{J}_c^T(\mathbf{q})\boldsymbol{F}, \tag{34}$$

where $\mathbf{q} \in \mathbb{R}^{n_j}$ is the vector of generalized coordinates, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n_j \times n_j}$ is the inertia matrix, $\tau \in \mathbb{R}^{n_j}$ is the applied torques vector, $\boldsymbol{F} \in \mathbb{R}^{n_c}$ is the contact forces, $\mathbf{J}_c^T \in \mathbb{R}^{n_j \times n_c}$ is the contact Jacobian, and $\mathbf{V}(\mathbf{q},\dot{\mathbf{q}}) \in \mathbb{R}^{n_j}$ comprises the terms not related to τ and $\boldsymbol{F}$, e.g., the Coriolis effect, and gravity.

Legged robots can be modeled as a *floating-base* kinematic tree, i.e., the base is considered free to move, rather than being fixed in space (FEATHERSTONE, 2008). A virtual joint with 6-DOFs is defined between the robot's CoM and the inertial frame. The floating-base system's dynamics can be analyzed with the same techniques as fixed-base systems. The equations of motion given by Equation (34) are rewritten considering the virtual joint

$$\begin{bmatrix} \mathbf{M}_u & \mathbf{M}_{ua} \\ \mathbf{M}_{au} & \mathbf{M}_a \end{bmatrix} \begin{bmatrix} \dot{\nu} \\ \ddot{\mathbf{q}} \end{bmatrix} + \begin{bmatrix} \mathbf{V}_u \\ \mathbf{V}_a \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \tau \end{bmatrix} + \begin{bmatrix} \mathbf{J}_{cu}^T \\ \mathbf{J}_{ca}^T \end{bmatrix} \boldsymbol{F}, \tag{35}$$

where $\nu = [\omega^T \ \ \dot{\mathbf{r}}^T]^T$; $\mathbf{M}_u \in \mathbb{R}^{6\times 6}$ and $\mathbf{M}_a \in \mathbb{R}^{n_j \times n_j}$ are the direct unactuated and actuated components of the inertia matrix; $\mathbf{M}_{ua} \in \mathbb{R}^{6\times n_j}$ and $\mathbf{M}_{au} \in \mathbb{R}^{n_j \times 6}$ are the cross-terms of the inertia matrix; $\mathbf{V}_u \in \mathbb{R}^6$ and $\mathbf{V}_a \in \mathbb{R}^{n_j}$ are the unactuated and actuated components of $\mathbf{V}$; $\mathbf{J}_{cu}^T \in \mathbb{R}^{6\times n_c}$ and $\mathbf{J}_{ca}^T \in \mathbb{R}^{n_j \times n_c}$ are the unactuated and actuated parts of the contact Jacobian.

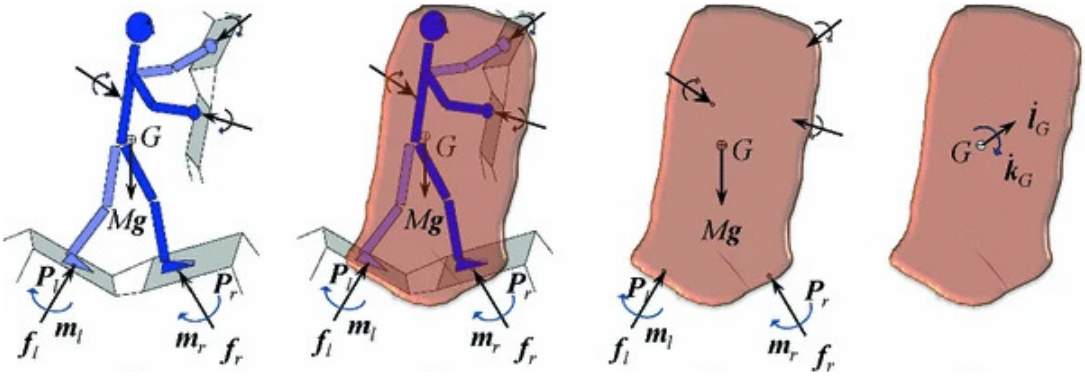
The model given by Equation (35) is referred as the *whole-body dynamics* or the *full-body dynamics* (VILLARREAL, 2020). Moreover, the presence of the cross-terms between the actuated and unactuated joints, and the dependency on the joints configuration in the elements of $\mathbf{M}$, $\mathbf{V}$, and $\mathbf{J}_c^T$, results in a significant complex model with a high degree of nonlinearity.

Considering the problem of controlling the position and orientation of the robot at any given time, instead of the whole-body dynamics, it can be used the *centroidal dynamics*. As presented in (ORIN; GOSWAMI; LEE, 2013), the centroidal dynamics consist on considering all the dynamics and forces applied in the system projected on the CoM of the robot. This concept is depicted in Figure 9.

The centroidal dynamics is obtained by considering the first row of Equation (35)

$$\mathbf{M}_u\dot{\nu} + \mathbf{M}_{ua}\ddot{\mathbf{q}} + \mathbf{V}_u = \mathbf{J}_{cu}^T\boldsymbol{F}, \tag{36}$$

Figure 9 – Representation of a centroidal dynamics.



Source: (ORIN; GOSWAMI; LEE, 2013).

which can be rewritten into the linear and angular dynamics of the robot's CoM in the world frame as follows

$$\begin{aligned} \mathrm{m}(\ddot{\mathbf{r}} - \mathbf{g}) &= \sum_{i=1}^{\mathrm{n_c}} \boldsymbol{F}_i, \\ \mathbf{I}\dot{\omega} + \omega \times \mathbf{I}\omega + \mathbf{I}_{\mathrm{leg}}\ddot{\mathbf{q}} &= \sum_{i=1}^{\mathrm{n_c}} \mathbf{p}_i \times \boldsymbol{F}_i, \end{aligned} \tag{37}$$

where m is the total mass of the robot, $\mathbf{g}$ is the gravity vector, $\mathbf{I}$ is the inertia tensor of the robot given in the world frame, $\mathbf{I}_{\mathrm{leg}}$ is associated to the legs' inertia projected in the robot's CoM, $\mathbf{p}_i$ is the ith contact point given in the base frame, $\boldsymbol{F}_i$ is the Ground Reaction Force (GRF) of the ith foot, where $i =$ LF, RF, LH, RH.

The evolution of the angular position, Θ, cannot be described directly from Equation (37). Therefore, consider the rotational kinematics described by

$$\mathbf{S}(\omega) = \dot{\mathbf{R}}\mathbf{R}^{\mathrm{T}}, \tag{38}$$

where $\mathbf{R} \in SO(3)$ is the rotation matrix from the body to the world coordinates, and the operator $\mathbf{S}(\,)$ is the skew-symmetric matrix, i.e., $\mathbf{S}(\mathbf{a})\mathbf{b} = \mathbf{a} \times \mathbf{b}$. The angular velocity expressed in the world frame can be obtained by

$$\omega = \begin{bmatrix} \cos\theta\cos\psi & -\sin\psi & 0 \\ \cos\theta\sin\psi & \cos\psi & 0 \\ -\sin\theta & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = W_\eta^{-1}(\Theta)\dot{\Theta}. \tag{39}$$

Accordingly, the rate of the Euler angles, $\dot{\Theta}$, can be obtained by

$$\dot{\Theta} = \begin{bmatrix} \cos\psi/\cos\theta & \sin\psi/\cos\theta & 0 \\ -\sin\psi & \cos\psi & 0 \\ \tan\theta\cos\psi & \tan\theta\sin\psi & 1 \end{bmatrix} \omega = W_\eta(\Theta)\omega. \tag{40}$$

Moreover, the inertia tensor in the world frame is dependent of the angular orientation Θ

$$\mathbf{I} = \mathbf{R}\mathbf{I}_{\mathbf{B}}\mathbf{R}^{\mathrm{T}}, \tag{41}$$

where $\mathbf{I_B}$ is the inertia tensor in the body frame.

Combining Equations (37) and (40), Villarreal (2020) defines the following continuous state-space system:

$$\begin{bmatrix} \dot{\Theta} \\ \dot{\mathbf{r}} \\ \dot{\omega} \\ \ddot{\mathbf{r}} \\ \dot{\mathrm{g}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & W_\eta(\Theta) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}_\mathrm{z} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Theta \\ \mathbf{r} \\ \omega \\ \dot{\mathbf{r}} \\ \mathrm{g} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{I}^{-1}\mathbf{S}(\mathbf{p}_\mathrm{LF}) & \dots & \mathbf{I}^{-1}\mathbf{S}(\mathbf{p}_\mathrm{RH}) \\ \mathbf{1}_3/\mathrm{m} & \dots & \mathbf{1}_3/\mathrm{m} \\ \mathbf{0} & \dots & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{F}_\mathrm{LF} \\ \boldsymbol{F}_\mathrm{RF} \\ \boldsymbol{F}_\mathrm{LH} \\ \boldsymbol{F}_\mathrm{RH} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ -\mathbf{I}^{-1}(\omega \times \mathbf{I}\omega + \mathbf{I}_\mathrm{leg}\ddot{\mathbf{q}}) \\ \mathbf{0} \\ 0 \end{bmatrix} \tag{42}$$

where $\mathbf{1}_\mathrm{z} = [0\ 0\ 1]^\mathrm{T}$, g the gravity value, $\mathbf{0}$ is a zero matrix of properly dimensions, and $\mathbf{1}_3$ is an identity matrix of order 3.

Focchi et al. (2016) and Villarreal (2020) make two assumptions about the model. Since the mass of the legs is smaller than the trunk, and the natural frequency of the legs dynamics is higher than the trunk dynamics, the cross-term product $\mathbf{I}_\mathrm{leg}\ddot{\mathbf{q}}$ is neglected. The second assumption is that the Coriolis term $\omega \times \mathbf{I}\omega$ is negligible for low angular rates. Hence, the following compact form is obtained:

$$\dot{\boldsymbol{x}}(t) = \mathbf{A}_\mathrm{c}(\Theta)\boldsymbol{x}(t) + \mathbf{B}_\mathrm{c}(\Theta, \mathbf{p}_\mathrm{LF}, \dots, \mathbf{p}_\mathrm{RH})\boldsymbol{u}(t), \tag{43}$$

where

$$\boldsymbol{x}(t) = \begin{bmatrix} \Theta^\mathrm{T} & \mathbf{r}^\mathrm{T} & \omega^\mathrm{T} & \dot{\mathbf{r}}^\mathrm{T} & \mathrm{g} \end{bmatrix}^\mathrm{T}, \tag{44}$$

and the control input is the vector of GRFs

$$\boldsymbol{u}(t) = \begin{bmatrix} \boldsymbol{F}_\mathrm{LF}^\mathrm{T} & \boldsymbol{F}_\mathrm{RF}^\mathrm{T} & \boldsymbol{F}_\mathrm{LH}^\mathrm{T} & \boldsymbol{F}_\mathrm{RH}^\mathrm{T} \end{bmatrix}^\mathrm{T}. \tag{45}$$

From Equation (42), note that $\mathbf{p}_i$ is zero if the leg i is not in stance. Hence, the system is a hybrid system, i.e., there are interactions between continuous and discrete dynamics (LIBERZON, 2003). However, since the future contact sequence is known, it can be directly replaced in the model, i.e., matrix $\mathbf{B}_\mathrm{c}$. Therefore, the matrix $\mathbf{B}_\mathrm{c}$ is not constant along the horizon, but the MPC does not need to treat the switching events due to the available information about the future contact sequence.

Moreover, the system described by Equation (43) is nonlinear due to the dependency of the angular position Θ. Hence, assuming that the controller will follow the angular trajectory sufficiently close, the matrices $\mathbf{A}_\mathrm{c}$ and $\mathbf{B}_\mathrm{c}$ are linearized around the angular reference Θ_d. Then, the discrete state space equation is obtained using ZOH with a sampling time of $\mathrm{T_s}$, which yields

$$\boldsymbol{x}_{k+1} = \mathbf{A}_k\boldsymbol{x}_k + \mathbf{B}_k\boldsymbol{u}_k, \tag{46}$$

where $\mathbf{A}_k = \mathbf{1}_{13} + \mathrm{T_s}\mathbf{A}_\mathrm{c}(\Theta_\mathrm{d}[k])$, $\mathbf{B}_k = \mathrm{T_s}\mathbf{B}_\mathrm{c}(\Theta_\mathrm{d}[k], \mathbf{p}_\mathrm{LF}[k], \mathbf{p}_\mathrm{RF}[k], \mathbf{p}_\mathrm{LH}[k], \mathbf{p}_\mathrm{RH}[k])$.

3.4.7.2 Cost Functional

To follow a desired state trajectory and to minimize the GRFs, the following cost functional is selected:

$$\mathbf{J}_\mathrm{MPC} = \sum_{i=1}^{\mathrm{n_y}} \|\boldsymbol{x}_{\mathrm{d}k+i} - \boldsymbol{x}_{k+i}\|^2_\mathbf{Q} + \sum_{i=0}^{\mathrm{n_u}-1} \|\boldsymbol{u}_{k+i}\|^2_\mathbf{R}, \tag{47}$$

where $\mathbf{Q} \in \mathbb{R}^{13\times13} \geq 0$ and $\mathbf{R} \in \mathbb{R}^{12\times12} > 0$ are weight matrices for the state deviation from a reference and the GRFs values.

3.4.7.3 Constraints

To the robot does not lose the stability during locomotion, it is important that the stance foot does not slip (KHORRAM; MOOSAVIAN, 2016). Thus, the tangential force should not exceed the friction force related to the normal force, which can be expressed as a friction cone. This condition can be approximated by a pyramidal description, with each component x and y of the contact force $\boldsymbol{F}_i$ does not exceed the friction force

$$\begin{aligned} \|\boldsymbol{F}_{x_i}\| &\leq \mu \boldsymbol{F}_{z_i}, \\ \|\boldsymbol{F}_{y_i}\| &\leq \mu \boldsymbol{F}_{z_i}, \\ 0 &\leq \boldsymbol{F}_{z_i}. \end{aligned} \tag{48}$$

To ensure a minimum distribution of the weight in all the stance legs, Villarreal (2020) also considers a minimum positive force $\mathrm{F_{min}}$ condition. Since there are actuator limitations, there is a maximum possible force that the legs can perform, thus

$$\mathrm{F_{min}} \leq \boldsymbol{F}_{z_i} \leq \mathrm{F_{max}}. \tag{49}$$

Combining Equations (48) and (49), a matrix composition can be formulated

$$\underbrace{\begin{bmatrix} 1 & 0 & -\mu \\ -1 & 0 & -\mu \\ 0 & 1 & -\mu \\ 0 & -1 & -\mu \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix}}_{\mathbf{k}_{\mathbf{F}i}} \boldsymbol{F}_i \leq \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mathrm{F_{max}} \\ -\mathrm{F_{min}} \end{bmatrix}}_{\mathbf{f}_{\mathrm{c}i}}, \tag{50}$$

which can be further constructed for the entire control input vector

$$\mathbf{k}_\mathbf{F} \boldsymbol{u}_k \leq \mathbf{f}_\mathrm{c}, \tag{51}$$

where $\mathbf{k_F} \in \mathbb{R}^{24\times12}$ is the diagonal composition of four matrices $\mathbf{k_{F}}_i$, and $\mathbf{f_c} \in \mathbb{R}^{24}$ is the matrix $\mathbf{f}_{ci}$ stacked four times.

Also, since the legs in swing are not removed from the model, it is necessary to enforce that the computed forces for those legs are zero

$$\mathscr{G}(\boldsymbol{x}_k)\boldsymbol{u}_k = \mathbf{0}, \tag{52}$$

where $\mathscr{G}$ is a matrix that maps the swing legs.

3.4.7.4 Optimization Problem

Given the model described by Equation (46), an optimization problem is formulated to compute the GRFs that minimizes the objective function in Equation (47) without violating the constraints described by Equations (51)-(52)

$$\begin{aligned}
\min_{\bar{\boldsymbol{u}}_k} \quad & \sum_{i=1}^{\mathrm{n_y}} \|\boldsymbol{x}_{\mathrm{d}k+i} - \boldsymbol{x}_{k+i}\|^2_{\mathbf{Q}} + \sum_{i=0}^{\mathrm{n_u}-1} \|\boldsymbol{u}_{k+i}\|^2_{\mathbf{R}} \\
s.t. \quad & \boldsymbol{x}_{k+i+1} = \mathbf{A}_{k+i}\boldsymbol{x}_{k+i} + \mathbf{B}_{k+i}\boldsymbol{u}_{k+i}, \\
& \mathbf{k_F}\boldsymbol{u}_{k+i} \leq \mathbf{f_c}, \\
& \mathscr{G}(\boldsymbol{x}_{k+i})\boldsymbol{u}_{k+i} = \mathbf{0}.
\end{aligned} \tag{53}$$

This problem is solved at 25 Hz and the resultant force at the first horizon is converted to wrench as follows

$$\mathcal{W}_{\mathrm{MPC}} = \sum_{i=1}^{\mathrm{n_c}} \begin{bmatrix} \mathbf{p}_i \times \boldsymbol{F}_i \\ \boldsymbol{F}_i \end{bmatrix} \tag{54}$$

which is sent to the trunk controller, which is detailed later in this chapter. The wrench can be interpreted as the resulting torque and force exerted in the CoM.

3.4.8 Leg Inertia Compensation

The prediction model from MPC neglects the leg inertia. This assumption is acceptable for robots that the legs weight is small in comparison with the total robot weight. However, for HyQ, the legs correspond to around 36% of the total weight.

From Equation (37), the wrench acting in the CoM is

$$\mathcal{W}_{\mathrm{leg}} = \mathbf{M}_{\mathrm{ua}}\ddot{\mathbf{q}}. \tag{55}$$

To prevent high-frequency signals, instead of using the measured joint acceleration $\ddot{\mathbf{q}}$ in $\mathcal{W}_{\mathrm{leg}}$, it is used the desired joint acceleration $\ddot{\mathbf{q}}_{\mathrm{d}}$, which is updated at the same frequency of the leg trajectory control, i.e., 250 Hz. Therefore, the legs inertia is added to the wrench associated to the MPC:

$$\mathcal{W}_{\mathrm{d}} = \mathcal{W}_{\mathrm{MPC}} + \mathbf{M}_{\mathrm{ua}}\ddot{\mathbf{q}}_{\mathrm{d}}. \tag{56}$$

3.4.9 Trunk Controller

While the MPC presented in Subsection 3.4.7 considers a sequence of time horizons, it does not consider the joint torques of the legs in swing or the joints states of the legs in stance.

Therefore, the trunk controller is designed to compute the instantaneous joint torques τ_{tc} to achieve the desired trunk motion, i.e., $\mathcal{W}_{\text{d}}$, while considering all the robot constraints. The whole-body dynamics given by Equation (35) is employed subjected to joint torque limits and maximum joint acceleration. To maintain the legs in stance on the ground, the following condition of zero acceleration for the feet is included:

$$\mathbf{J}_{\text{ca,st}}\ddot{\mathbf{q}}_{\text{st}} + \dot{\mathbf{J}}_{\text{ca,st}}\dot{\mathbf{q}}_{\text{st}} = \mathbf{0}. \tag{57}$$

The wrench of the CoM is defined as:

$$\mathcal{W}_{\text{com}} = \mathbf{J}^{\text{T}}_{\text{st,com}}\boldsymbol{F}, \tag{58}$$

where $\mathbf{J}^{\text{T}}_{\text{st,com}}$ is the Jacobian between the stance feet and the CoM. The following optimization problem is formulated with the decision variable $\mathbf{U}_{\text{tc}} = [\dot{\nu}^{\text{T}} \;\; \boldsymbol{F}^{\text{T}}]^{\text{T}} \in \mathbb{R}^{30}$ to track the desired trunk motion $\mathcal{W}_{\text{d}}$ given by Equation (56) subject to the constraints given by Equations (51) and (57)

$$\begin{aligned}
\min_{\boldsymbol{F},\dot{\nu}} \quad & \|\mathcal{W}_{\text{com}} - \mathcal{W}_{\text{d}}\|^2_{\mathbf{Q}_{\text{tc}}} + \|\mathbf{U}_{\text{tc}}\|^2_{\mathbf{R}_{\text{tc}}} \\
s.t. \quad & \tau_{\text{tc}} = \mathbf{M}\dot{\nu} + \mathbf{V} - \mathbf{J}^{\text{T}}_{\text{c}}\boldsymbol{F}, \\
& \tau_{\text{min}} \leq \tau_{\text{tc}} \leq \tau_{\text{max}}, \\
& \ddot{\mathbf{q}}_{\text{min}} \leq \ddot{\mathbf{q}} \leq \ddot{\mathbf{q}}_{\text{max}}, \\
& \mathbf{J}_{\text{ca,st}}\ddot{\mathbf{q}}_{\text{st}} + \dot{\mathbf{J}}_{\text{ca,st}}\dot{\mathbf{q}}_{\text{st}} = \mathbf{0}, \\
& \mathbf{k}_{\mathbf{F}}\boldsymbol{F} \leq \mathbf{f}_{\text{c}},
\end{aligned} \tag{59}$$

where $\mathbf{Q}_{\text{tc}} \in \mathbb{R}^{6\times 6} \geq 0$ and $\mathbf{R}_{\text{tc}} \in \mathbb{R}^{30\times 30} > 0$ are weight matrices for the wrench tracking and the decision variables, respectively.

3.4.10 Low-Level Joint Torque Control

To properly handle the physical contacts with the ground and its reaction, a *Compliance Controller* is designed to control the joints' torque. This type of controller has the feature to induce a desirable stiffness and damping for the robot leg.

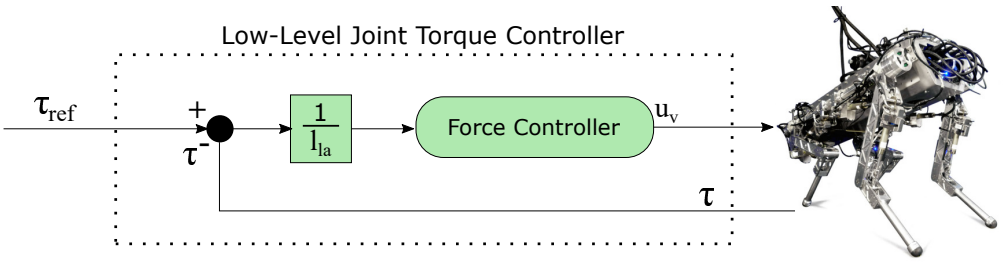
In HyQ, the rod from the hydraulic cylinders is rigidly connected to the joints. Consequently, the joint torque τ is a result of the hydraulic cylinder force and the lever arm length l_{la}, which is defined by the distance between the cylinder position and the attachment point at the joint. Therefore, to control the joint's torques, it is necessary to control the hydraulic forces, which are controlled by the valve position u_v.

Due to the nonlinear dynamics of the hydraulic actuators, a *Force Feedback Linearization* is used (CUNHA, 2013). For this controller, the current state of the hydraulic system is used

to linearize the relation between the control input and the controlled variable within the whole range of operation of the system.

The joint controller runs at 1 *k*Hz in an RT thread, Figure 10 illustrates the control scheme. More details about the torque joint control, and mainly the force controller, can be obtained in (CUNHA, 2013).

Figure 10 – Description of the low-level joint torque controller.

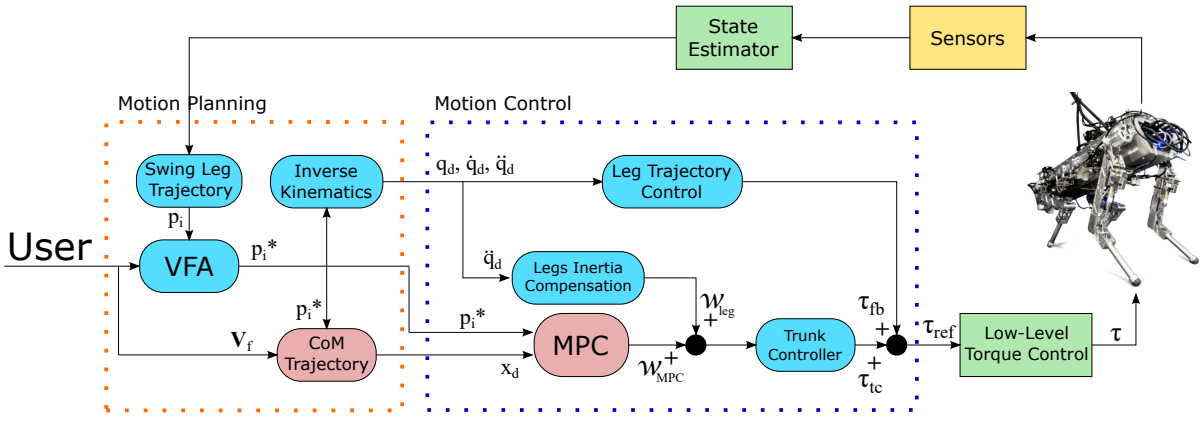


Source: the author, 2022.

3.5 CHAPTER SUMMARY

In this chapter, the hardware and software characteristics of the HyQ robot were presented. The formulated controllers and the respective mathematical models of the control architecture components were detailed. The control architecture and its components are summarized in Figure 11.

Figure 11 – HyQ control architecture, green blocks runs in 1 *k*Hz, blue blocks in 250 Hz, pink-red blocks in 25 Hz.



Source: the author, 2022.

In the next chapter, modifications are proposed for the MPC presented in Subsection 3.4.7, and an SMPC is formulated to improve the system's robustness.

4 STOCHASTIC MODEL PREDICTIVE CONTROL FOR DYNAMIC LOCOMOTION

In this chapter, it is presented the controllers that are proposed in this work. Initially, the modifications in the original MPC designed in (VILLARREAL, 2020) are described. Then, the proposed SMPC is formulated.

4.1 MODEL PREDICTIVE CONTROL

In the model represented by Equation (42), the gravity is included directly in the state vector for convenience. However, it inserts an uncontrollable state that introduces extra complexity in further analysis, e.g., robustness and stability. Still, some control techniques consider that all states are controllable, for example, the design of the gain $\mathbf{K}_k$ by an LQR in Equation (10). Also, it increases the size of the state vector, which increases the memory usage and the computational cost.

Therefore, the model is reformulated to consider gravity as a known disturbance in the model, which can be explicitly considered in the optimization problem. Accordingly, matrices $\mathbf{A}$ and $\mathbf{B}$ are reformulated as follows

$$\begin{bmatrix} \dot{\Theta} \\ \dot{\mathbf{r}} \\ \dot{\omega} \\ \ddot{\mathbf{r}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & W_\eta(\Theta) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}_3 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Theta \\ \mathbf{r} \\ \omega \\ \dot{\mathbf{r}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{I}^{-1}\mathbf{S}(\mathbf{p}_{\mathrm{LF}}) & \dots & \mathbf{I}^{-1}\mathbf{S}(\mathbf{p}_{\mathrm{RH}}) \\ \mathbf{1}_3/\mathrm{m} & \dots & \mathbf{1}_3/\mathrm{m} \end{bmatrix} \begin{bmatrix} \boldsymbol{F}_{\mathrm{LF}} \\ \boldsymbol{F}_{\mathrm{RF}} \\ \boldsymbol{F}_{\mathrm{LH}} \\ \boldsymbol{F}_{\mathrm{RH}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1}_\mathrm{z} \end{bmatrix} \mathrm{g}, \tag{60}$$

which can be rewritten in a compact form as

$$\dot{\boldsymbol{x}}(t) = \mathbf{A}_\mathrm{c}(\Theta)\boldsymbol{x}(t) + \mathbf{B}_\mathrm{c}(\Theta, \mathbf{p}_{\mathrm{LF}}, \dots, \mathbf{p}_{\mathrm{RH}})\boldsymbol{u}(t) + \mathbf{E}_{\mathrm{cg}}\mathbf{d}_\mathrm{g}(t). \tag{61}$$

The discrete model is obtained by the same procedure presented in Subsection 3.4.7.1, yielding to

$$\boldsymbol{x}_{k+1} = \mathbf{A}_k\boldsymbol{x}_k + \mathbf{B}_k\boldsymbol{u}_k + \mathbf{E}_\mathrm{g}\mathbf{d}_{\mathrm{g}_k}, \tag{62}$$

where $\mathbf{E}_\mathrm{g} = \mathrm{T_s}\mathbf{E}_{\mathrm{cg}}$. The modified model is rewritten along the horizons as follows

$$\bar{\boldsymbol{x}}_k = \mathbf{S}^x\boldsymbol{x}_k + \mathbf{S}^u\bar{\boldsymbol{u}}_k + \mathbf{S}^\mathrm{g}\bar{\mathbf{d}}_\mathrm{g}, \tag{63}$$

where $\bar{\mathbf{d}}_\mathrm{g} \in \mathbb{R}^{\mathrm{n_y}}$ is the g value stacked $\mathrm{n_y}$ times, and

$$\mathbf{S}^\mathrm{g} = \begin{bmatrix} \mathbf{E}_\mathrm{g} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{A}_k\mathbf{E}_\mathrm{g} & \mathbf{E}_\mathrm{g} & \ddots & \mathbf{0} \\ \mathbf{A}_k^2\mathbf{E}_\mathrm{g} & \mathbf{A}\mathbf{E}_\mathrm{g} & \ddots & \mathbf{0} \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{A}_k^{\mathrm{n_y}-1}\mathbf{E}_\mathrm{g} & \mathbf{A}_k^{\mathrm{n_y}-2}\mathbf{E}_\mathrm{g} & \dots & \mathbf{E}_\mathrm{g} \end{bmatrix}. \tag{64}$$

4.1.1 Angular Velocity Constraints

As any real system, HyQ robot has physical limits. Since the reference generator in Subsection 3.4.5 does not consider dynamically feasible criteria, maximum angular velocity constraints are considered. Also, another aspect to introduce angular velocities constraints is to keep the robot close to the linearized trajectory.

Accordingly, considering that the angle ψ is not relevant in terms of robot's stability, constraints are considered only in the terms ω_{x} and ω_{y} of the angular velocity

$$\omega_{\mathrm{x}\min} \leq \omega_{\mathrm{x}k+1} \leq \omega_{\mathrm{x}\max}, \tag{65}$$

$$\omega_{\mathrm{y}\min} \leq \omega_{\mathrm{y}k+1} \leq \omega_{\mathrm{y}\max}, \tag{66}$$

which can be rewritten in the matrix form

$$\mathbf{C}_{\omega}\boldsymbol{x}_{k+1} \leq \mathbf{h}_{\omega}, \tag{67}$$

with

$$\mathbf{C}_{\omega} = \begin{bmatrix} \mathbf{0}_{2\times 6} & \mathbf{1}_2 & \mathbf{0}_{2\times 4} \\ \mathbf{0}_{2\times 6} & -\mathbf{1}_2 & \mathbf{0}_{2\times 4} \end{bmatrix}, \tag{68}$$

$$\mathbf{h}_{\omega} = \begin{bmatrix} \omega_{\mathrm{x}\max} & \omega_{\mathrm{y}\max} & -\omega_{\mathrm{x}\min} & -\omega_{\mathrm{y}\min} \end{bmatrix}^{\mathrm{T}}, \tag{69}$$

where $\mathbf{1}_2$ is an identity matrix of order 2.

4.1.2 Angular Momentum Rate

Given the angular momentum definition

$$\mathbf{L} = \mathbf{I}\omega, \tag{70}$$

its time derivative, i.e., the angular momentum rate, is defined as

$$\dot{\mathbf{L}} = \dot{\mathbf{I}}\omega + \mathbf{I}\dot{\omega}. \tag{71}$$

Since the time derivative of the inertia matrix $\dot{\mathbf{I}}$ is usually very low or even null for low angular rates, $\dot{\mathbf{L}}$ can be directly related to the angular acceleration $\dot{\omega}$, and thus, to the smoothness of the angular trajectory. Therefore, the minimization of $\dot{\mathbf{L}}$ is included in the controller design to provide a natural behavior to the robot, similar to a real animal.

According to Newton's law, the rate of change of the angular momentum is equivalent to the resultant effect of the external torques (ORIN; GOSWAMI; LEE, 2013). In the case of the simplified model for HyQ, the external torques are the projected torques at the CoM exerted by the GRFs, which are obtained by

$$\dot{\mathbf{L}}[k] = \tau_{\mathrm{GRF}}[k] = \sum_{i=1}^{\mathrm{n_c}} \mathbf{p}_i[k] \times \boldsymbol{F}_i[k]. \tag{72}$$

Based on the prediction model given by Equation (60), $\dot{\mathbf{L}}$ can be obtained from the control input

$$\dot{\mathbf{L}}[k] = \mathbf{S}_{\text{skew}}[k]\boldsymbol{u}[k], \tag{73}$$

where $\mathbf{S}_{\text{skew}}$ is a matrix of the skew-symmetric matrices related to the contact points

$$\mathbf{S}_{\text{skew}}[k] = \begin{bmatrix} \mathbf{S}(\mathbf{p}_{\text{LF}}[k]) & \mathbf{S}(\mathbf{p}_{\text{RF}}[k]) & \mathbf{S}(\mathbf{p}_{\text{LH}}[k]) & \mathbf{S}(\mathbf{p}_{\text{RH}}[k]) \end{bmatrix}. \tag{74}$$

Thus, the cost functional given by Equation (47) is modified as follow

$$\mathbf{J}_{\text{MPC}} = \sum_{i=1}^{\text{n}_\text{y}} \|\boldsymbol{x}_{\text{d}k+i} - \boldsymbol{x}_{k+i}\|^2_{\mathbf{Q}} + \sum_{i=0}^{\text{n}_\text{u}-1} \|\boldsymbol{u}_{k+i}\|^2_{\mathbf{R}} + \sum_{i=0}^{\text{n}_\text{u}-1} \|\dot{\mathbf{L}}_{k+i}\|^2_{\mathbf{R}_\mathbf{L}}, \tag{75}$$

where $\mathbf{R}_\mathbf{L} \in \mathbb{R}^3 > 0$.

Accordingly, consider the constraint in Equation (67) and the cost functional (75), the following optimization problem is formulated:

$$\begin{aligned}
\min_{\bar{\boldsymbol{u}}_k} \quad & \sum_{i=1}^{\text{n}_\text{y}} \|\boldsymbol{x}_{\text{d}k+i} - \boldsymbol{x}_{k+i}\|^2_{\mathbf{Q}} + \sum_{i=0}^{\text{n}_\text{u}-1} \|\boldsymbol{u}_{k+i}\|^2_{\mathbf{R}} + \sum_{i=0}^{\text{n}_\text{u}-1} \|\dot{\mathbf{L}}_{k+i}\|^2_{\mathbf{R}_\mathbf{L}} \\
\textit{s.t.} \quad & \boldsymbol{x}_{k+i+1} = \mathbf{A}_{k+i}\boldsymbol{x}_{k+i} + \mathbf{B}_{k+i}\boldsymbol{u}_{k+i}, \\
& \dot{\mathbf{L}}_{k+i} = \mathbf{S}_{\text{skew}k+i}\boldsymbol{u}_{k+i}, \\
& \mathbf{k}_\mathbf{F}\boldsymbol{u}_{k+i} \leq \mathbf{f}_\text{c}, \\
& \mathscr{G}(\boldsymbol{x}_{k+i})\boldsymbol{u}_{k+i} = \mathbf{0}, \\
& \mathbf{C}_\omega \boldsymbol{x}_{k+i} \leq \mathbf{h}_\omega.
\end{aligned} \tag{76}$$

4.2 STOCHASTIC MODEL PREDICTIVE CONTROL

In this section, an SMPC is designed based on the Stochastic Tube MPC approach presented in Subsection 2.4.4.1.

Revisiting the continuous state-space system described in Equation (43), the assumptions related to the legs inertia and Coriolis effect influence the evolution of the angular velocity $\boldsymbol{\omega}$.

Also, any measurement of the state vector variables in the real robot has associated some level of uncertainty due to noise or even some kind of delay. However, due to the assumptions in the model formulation, uncertainty is assumed only in the angular velocity variables.

Therefore, based on the description of the discrete state-space with additive uncertainty given by Equation (9), it is included an uncertainty term in the system given by Equation (62) as follows

$$\boldsymbol{x}_{k+1} = \mathbf{A}_k\boldsymbol{x}_k + \mathbf{B}_k\boldsymbol{u}_k + \mathbf{E}_\text{g}\mathbf{d}_{\text{g}_k} + \mathbf{D}\boldsymbol{w}_k, \tag{77}$$

where $\boldsymbol{w}_k \in \mathbb{R}^3$ is assumed as a white noise with Gaussian distribution, i.e., $\boldsymbol{w} \sim \mathscr{N}(0, \Sigma^{\boldsymbol{w}})$, $\Sigma^{\boldsymbol{w}}$ is characterized later in this chapter, and

$$\mathbf{D} = \begin{bmatrix} \mathbf{0}_{3\times 6} & \mathbf{1}_3 & \mathbf{0}_{3\times 3} \end{bmatrix}^\text{T}. \tag{78}$$

Usually, the Stochastic Tube MPC applies the state decomposition given by Equation (10), redefining the control as an affine control law. In the case of the discrete model formulated for HyQ, the challenge is to obtain a proper feedback gain that deals with the hybrid dynamics of the system.

For switching and hybrid systems, an immediately control approach is to find a common Lyapunov function for all the individual subsystems (LIBERZON, 2003). However, this approach is usually over-conservative. In the case of legged systems, during some switching modes, some columns of matrix $\mathbf{B}$ will be null, which results in a not fully controllable system. Hence, a common gain $\mathbf{K}$ is impossible to obtain. Also, in general, these techniques require the solution of an optimization problem. Therefore, it would be necessary to solve an optimization process for each step of the time horizon at each control step, which probably would take more time to compute than the MPC, resulting in a computationally intractable problem.

Therefore, the control input $\boldsymbol{u}$ was not redefined as an affine control law, maintaining $\boldsymbol{u}$ as the decision variable.

Given the existence of uncertainty in the angular velocities, the constraint described by Equation (67) is formulated to satisfy a minimum probability level p as follows

$$\mathscr{P}\{\mathbf{C}_\omega \boldsymbol{x}_{k+1} \leq \mathbf{h}_\omega\} \geq 1 - \mathrm{p}. \tag{79}$$

Therefore, based on the chance-constraints procedure described in Subsection 2.4.2, a tightened term λ_{k+1} is included in the original constraint

$$\mathbf{C}_\omega \boldsymbol{x}_{k+1} \leq \mathbf{h}_\omega - \lambda_{k+1}, \tag{80}$$

where each row i is computed as

$$\lambda_{ki}(1-\mathrm{p}) = \sqrt{(\bar{\mathbf{C}}_\omega \mathbf{S}^w)_i \bar{\Sigma^w} (\bar{\mathbf{C}}_\omega \mathbf{S}^w)_i^{\mathrm{T}}} f_{\mathrm{cdf}}^{-1}(1-\mathrm{p}), \tag{81}$$

where $\bar{\mathbf{C}}_\omega$ is the diagonal composition of $\mathrm{n_y}$ matrix $\mathbf{C}_\omega$, i.e., $\bar{\mathbf{C}}_\omega = \mathrm{diag}(\mathbf{C}_\omega, \ldots, \mathbf{C}_\omega)$. Since the system description given by Equation (77) is updated at each control step, the constraint tightened λ needs to be recomputed at each control step.

Concerning the cost functional, the expected value of the cost functional (75) is employed. Therefore, the following optimization problem is formulated

$$\begin{aligned}
\min_{\bar{\boldsymbol{u}}_k} \quad & \mathbb{E}\left[\sum_{i=1}^{\mathrm{n_y}} \|\boldsymbol{x}_{\mathrm{d}k+i} - \boldsymbol{x}_{k+i}\|_{\mathbf{Q}}^2 + \sum_{i=0}^{\mathrm{n_u}-1} \|\boldsymbol{u}_{k+i}\|_{\mathbf{R}}^2 + \sum_{i=0}^{\mathrm{n_u}-1} \|\dot{\mathbf{L}}_{k+i}\|_{\mathbf{R_L}}^2 \right] \\
\textit{s.t.} \quad & \boldsymbol{x}_{k+i+1} = \mathbf{A}_{k+i}\boldsymbol{x}_{k+i} + \mathbf{B}_{k+i}\boldsymbol{u}_{k+i} + \mathbf{E}_{\mathrm{g}}\mathbf{d}_{\mathrm{g}k+i} + \mathbf{D}\boldsymbol{w}_{k+i}, \\
& \dot{\mathbf{L}}_{k+i} = \mathbf{S}_{\mathrm{skew}k+i}\boldsymbol{u}_{k+i}, \\
& \mathbf{k_F}\boldsymbol{u}_{k+i} \leq \mathbf{f}_{\mathrm{c}}, \\
& \mathscr{G}(\boldsymbol{x}_{k+i})\boldsymbol{u}_{k+i} = \mathbf{0}, \\
& \mathbf{C}_\omega \boldsymbol{x}_{k+i} \leq \mathbf{h}_\omega - \lambda_{k+i}.
\end{aligned} \tag{82}$$

4.3 CONTROLLER DESIGN

In this section, it is presented the model and controller parameters. Also, the procedure for the tightened terms related to chance-constraints is described.

To compute the angular velocity constraints, it is specified the minimum time to the robot overpass the inclination of a terrain or between two obstacles. Thus, considering that the robot takes at least 1 second to overpass an obstacle of 30° in the ϕ coordinate, and 25° in the θ coordinate, the values of the constraints are given by

$$\omega_{\mathrm{x}\max} = -\omega_{\mathrm{x}\min} = 0.5 \text{ rad/s}, \tag{83}$$

$$\omega_{\mathrm{y}\max} = -\omega_{\mathrm{y}\min} = 0.4 \text{ rad/s}. \tag{84}$$

In Table 1, the parameters for the model given by Equation (60), the gait parameters, and the parameters related to the cost functional and the constraints are summarized.

Table 1 – Model and controller parameters.

Parameter	Value	Unit	Parameter	Value	Unit
m	87.404	kg	$\omega_{\mathrm{x}\min}$	-0.5	rad/s
g	-9.81	$\mathrm{m/s^2}$	$\omega_{\mathrm{x}\max}$	0.5	rad/s
D_f	0.65	-	$\omega_{\mathrm{y}\min}$	-0.4	rad/s
f_{step}	1.4	Hz	$\omega_{\mathrm{y}\max}$	0.4	rad/s
$V_{\mathrm{f,x}}$	0.15	m/s	$\mathrm{F}_{\max}$	1000	N
$V_{\mathrm{f,y}}$	0.0	m/s	$\mathrm{F}_{\min}$	10	N
ψ_{d}	0.0	rad	μ	0.7	N
$\mathrm{n_y}$	12	-	$\mathrm{T_s}$	40.0	*m*s
$\mathrm{n_u}$	12	-			

Source: the author, 2021.

The inertia tensor in the body frame is

$$\mathbf{I_B} = \begin{bmatrix} 1.5726 & 0.0284 & -0.2031 \\ 0.02838 & 8.5016 & -0.0045 \\ -0.20314 & -0.0045 & 9.1955 \end{bmatrix}. \tag{85}$$

And the weight terms are selected as

$$\begin{aligned} \mathbf{Q} &= \mathrm{diag}(200, 200, 50, 1, 1, 50, 1, 1, 1, 1, 1, 50), \\ \mathbf{R} &= 1 \times 10^{-10}\mathbf{1}_{12}, \\ \mathbf{R_L} &= 1 \times 10^{-4}\mathbf{1}_{3}. \end{aligned} \tag{86}$$

4.3.1 Computation of Chance-Constraint Tightened

For a Gaussian distribution, to compute the chance-constraint tightened (81), two pieces of information are necessary: the variance $\Sigma^{\boldsymbol{w}}$ of the noise signal, and the maximum acceptable

probability level p of the constraints to be violated. The last one is not directly related to the system, and it can be selected by the controller designer. The first one relies on the information, or at least an estimation, of the level of uncertainty in the system evolution.

To estimate the noise variance, it is analyzed one trial of the MPC formulated in Subsection 4.1 without considering the angular velocity constraints and the minimization of $\dot{\mathbf{L}}$. The simulated scenario and the simulation setup are detailed in the next chapter. It is evaluated the *prediction error*, i.e., the error between the angular velocity at instant k and its predicted value at instant $k-1$, which yields

$$error_{\omega} = \omega_k - \omega_{k|k-1}. \tag{87}$$

Accordingly, all the unmodeled dynamics and errors in the prediction model are assumed to be Gaussian processes with zero mean.

The prediction error of the angular velocities of the analyzed trial is presented in Figure 12. The variance is obtained using the function *var* from MATLAB R2021a, resulting in the values:

$$\Sigma^{\boldsymbol{w}}{}_{\text{trial}} = \begin{bmatrix} 59.25 \times 10^{-3} & 5.00 \times 10^{-3} & 0.74 \times 10^{-3} \end{bmatrix}^{\text{T}}. \tag{88}$$

During the simulation experiments, the obtained value associated to ω_{x} led to an infeasible optimization problem at some control steps, demonstrating to be over-conservative. Since the prediction error is assumed to be a Gaussian process, which is a rough approximation, the variance values cannot be considered as an absolute truth, but only an estimation. Therefore, a reduced value is for first one, resulting in the following vector

$$\Sigma^{\boldsymbol{w}} = \begin{bmatrix} 10.00 \times 10^{-3} & 5.00 \times 10^{-3} & 0.74 \times 10^{-3} \end{bmatrix}^{\text{T}}, \tag{89}$$

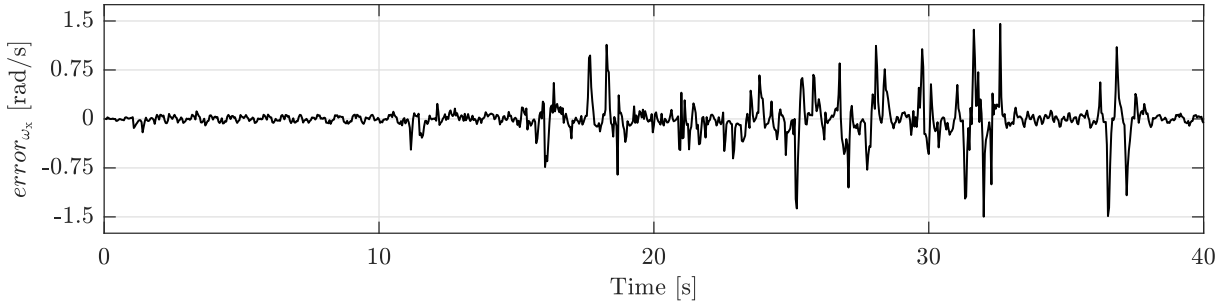
Consider the variance vector $\Sigma^{\boldsymbol{w}}$ given by Equation (89) and a probability level p of 1%, the tightened terms for the chance-constraints along the horizon are computed using Equation (81), which yields

$$\begin{aligned} \bar{\lambda} = \Big[& 0.233 \quad 0.165 \quad 0.329 \quad 0.233 \quad 0.403 \quad 0.285 \quad 0.465 \quad 0.329 \quad 0.520 \quad 0.368 \quad 0.570 \quad 0.402 \ldots \\ & 0.616 \quad 0.435 \quad 0.658 \quad 0.465 \quad 0.698 \quad 0.494 \quad 0.736 \quad 0.520 \quad 0.772 \quad 0.546 \quad 0.806 \quad 0.570 \Big]. \end{aligned} \tag{90}$$

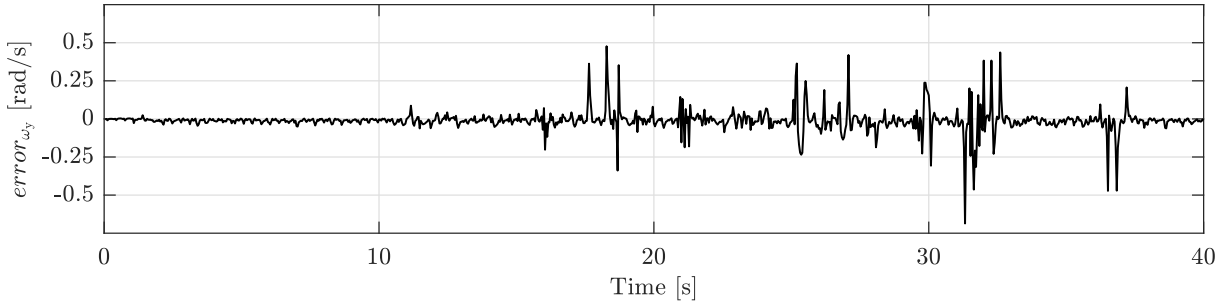
Note that for the constraint of 0.5 rad/s for ω_{xmax}, from the fifth prediction step, i.e., the ninth term of the vector, the adjustment is so large that the problem is infeasible. Therefore, based on a similar argument presented in Kouvaritakis et al. (2010), $\bar{\lambda}$ is truncated in the fourth prediction step

$$\begin{aligned} \bar{\lambda} = \Big[& 0.233 \quad 0.165 \quad 0.329 \quad 0.233 \quad 0.403 \quad 0.285 \quad 0.465 \quad 0.329 \quad 0.465 \quad 0.329 \quad 0.465 \quad 0.329 \ldots \\ & 0.465 \quad 0.329 \quad 0.465 \quad 0.329 \quad 0.465 \quad 0.329 \quad 0.465 \quad 0.329 \quad 0.465 \quad 0.329 \quad 0.465 \quad 0.329 \Big]. \end{aligned} \tag{91}$$

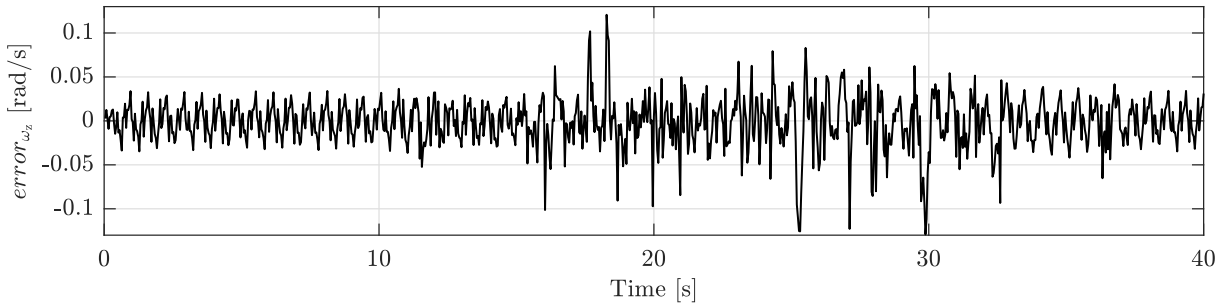
Figure 12 – Prediction error of the angular velocity ω for MPC.



(a) Prediction error for ω_x.



(b) Prediction error for ω_y.



(c) Prediction error for ω_z.

Source: the author, 2022.

4.4 CHAPTER SUMMARY

In this chapter, the modifications in the MPC model were presented. A constraint in the angular velocities was included to maintain the robot close to the linearized trajectory. And the minimization of the angular momentum rate was incorporated in the cost functional.

An SMPC was formulated with chance-constraints for the angular velocity in order to consider the presence of uncertainty in the prediction model and to improve the controller's robustness.

The system's uncertainty was assumed as a Gaussian distribution, which was estimated based on the prediction error.

In the next chapter, simulation results and a comparison between the controllers are presented. For the SMPC, it is expected a decrease in the number of violations of the angular velocity constraints. A smoother trajectory is expected when minimizing $\dot{\mathbf{L}}$.

5 SIMULATION RESULTS

In this chapter, the simulation results are presented for comparison purposes of the proposed controllers.

5.1 SIMULATION SETUP

DLS developed a software framework for legged robots. This framework has a collection of tools and features for navigation, control, terrain estimation, and simulation.

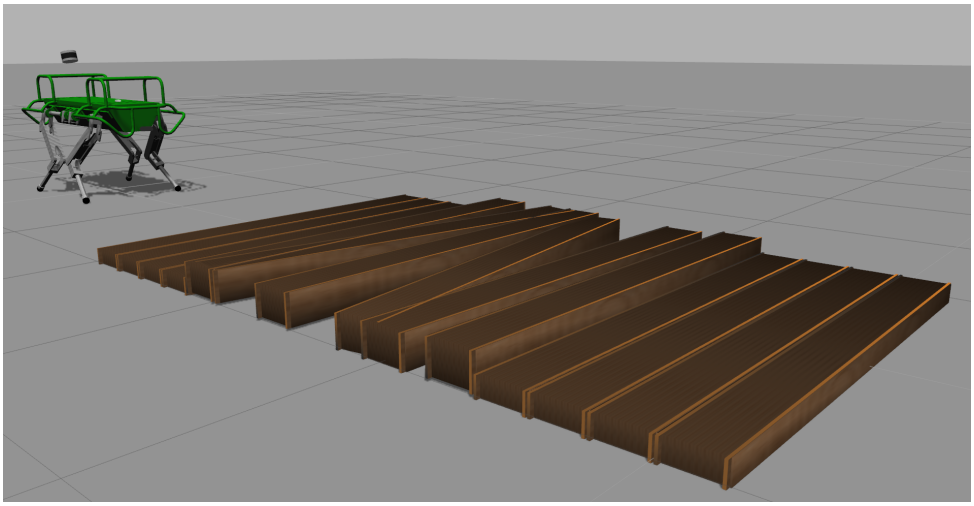
The control architecture in Figure 11 is implemented on the IIT's framework, allowing the simulation of the robot in the Gazebo Simulator (GAZEBO, 2019). Since the dynamics of the hydraulic actuators and also the torque control loops are not considered in the simulation, the desired torques computed by the Trunk Controller and Leg Trajectory Control can be directly applied to the joints. Therefore, the Low-Level Torque Control is not employed.

The MPC and CoM Trajectory Generator run in a single ROS node. This node exchanges data using a ROS topic with another node that runs the entire framework and interacts with the simulation.

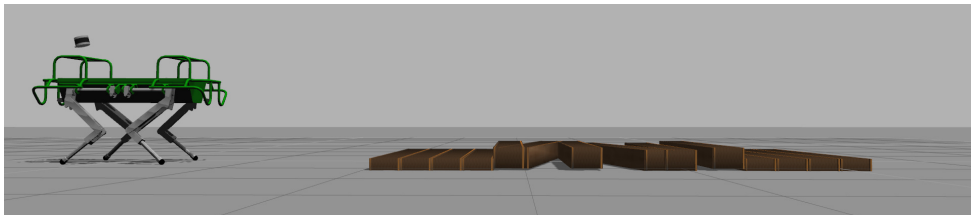
The scenario with obstacles used to compare the controllers is presented in Figure 13. The sequence of obstacles has similar dimensions to a sequence of three industrial pallets at different heights and orientations.

To solve the optimization problems associated with the controllers, the solver QuadProg++ is used (GASPERO; MOYER, 2021), which employs the algorithm presented in (GOLDFARB; IDNANI, 1983). All the simulations are performed in a computer with an Intel Core i7-9700, 16 GB RAM, and Ubuntu 18.04 LTS operating system.

Figure 13 – Robot on the simulated scenario with obstacles.



(a) Robot on simulation with obstacles.



(b) Horizontal view of the simulated scenario.

Source: the author, 2022.

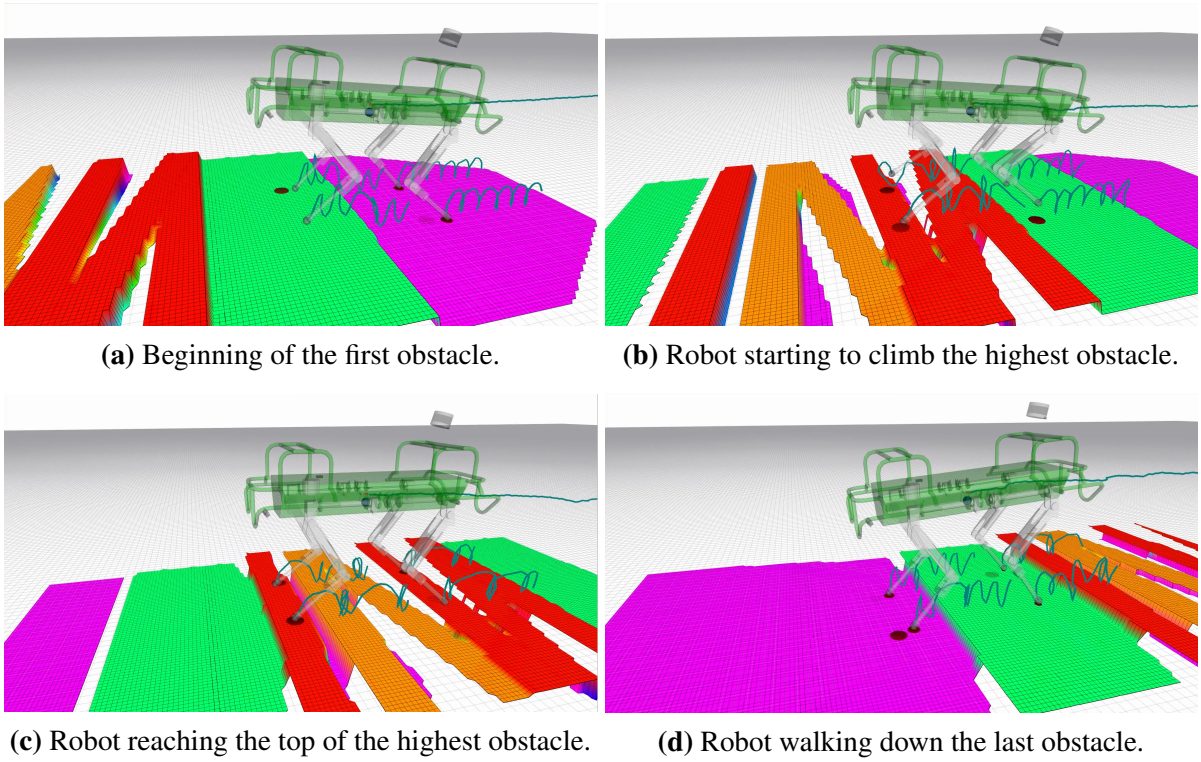
5.2 RESULTS

To evaluate the modifications proposed in Chapter 4, six controllers are designed:

- MPC: MPC with the modified prediction model presented in Section 4.1, used as a baseline for comparison;
- MPC-C: MPC with the angular velocity constraint formulated in Section 4.1.1;
- SMPC: SMPC using the chance-constraints formulation presented in Section 4.2 without considering the angular momentum rate;
- MPC-L: MPC with angular momentum rate minimization;
- MPC-CL: MPC-C with angular momentum rate minimization;
- SMPC-L: SMPC with angular momentum rate minimization.

During the simulation experiments, slightly differences were verified in the results. A possible reason is due to small variations when the start command is sent with relation to the current stage of the swing phase. Then, 30 simulations trials are performed for each designed controller. An example of the HyQ robot locomotion on rough terrain with the designed MPC is presented in Figure 14.

Figure 14 – Example of trajectory from HyQ robot traversing a rough terrain using MPC.



(a) Beginning of the first obstacle.

(b) Robot starting to climb the highest obstacle.

(c) Robot reaching the top of the highest obstacle.

(d) Robot walking down the last obstacle.

Source: the author, 2022.

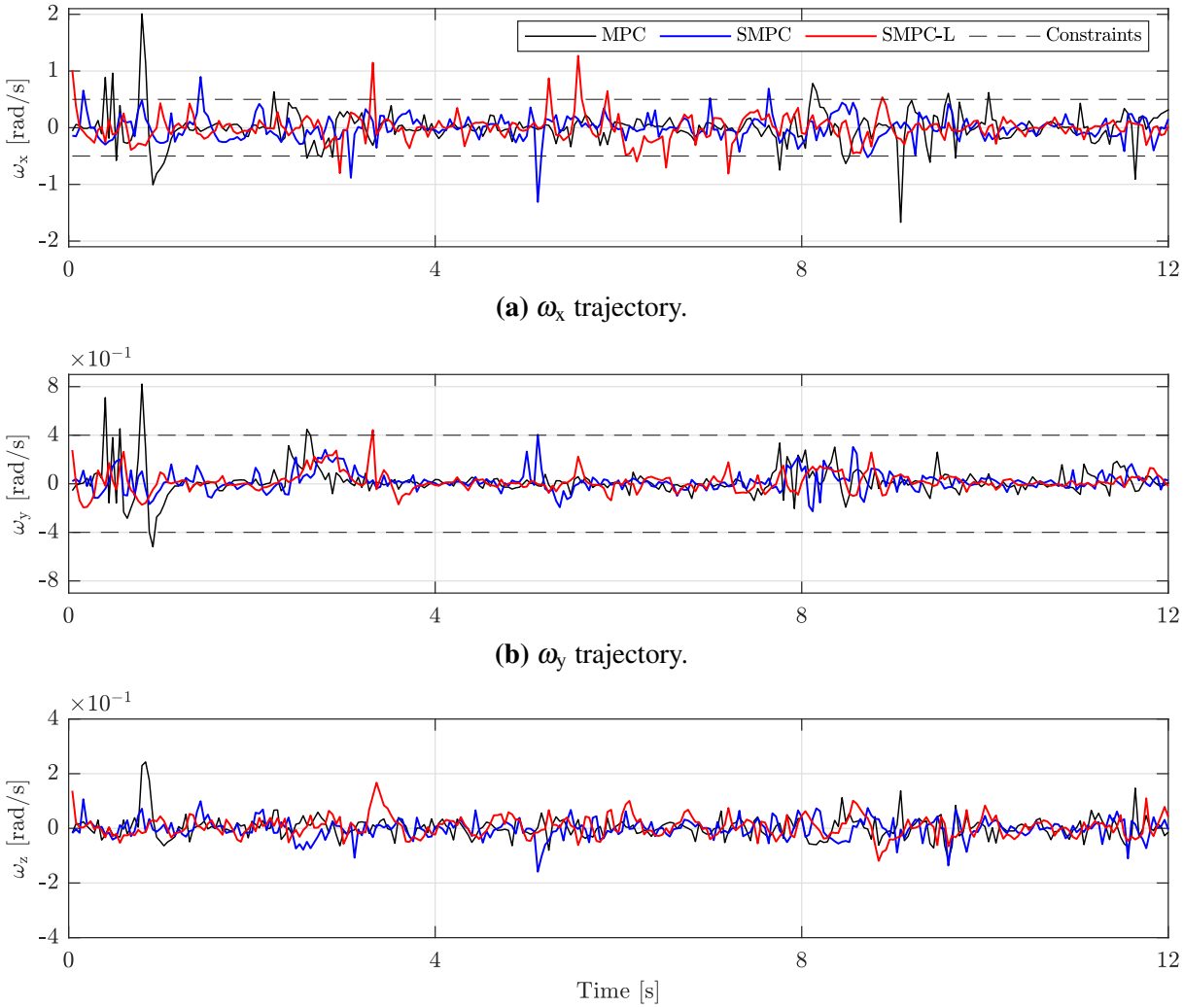
For the linear position and velocity, the trajectories did not change for the designed controllers, which is verified in the analysis of the Root Mean Square (RMS) of the trajectory error presented later in this chapter. This result is expected since the proposed modifications are related to the angular variables. Hence, the initial focus of the simulation results analysis is on the angular variables.

For an initial comparison, a segment of the angular trajectory for the MPC, SMPC, and SMPC-L of one trial is presented in Figure 15. This segment corresponds to the time frame of 12 seconds from the instant that the robot start to climb the highest obstacle, depicted in Figure 14b, until it reaches the top of it, as presented in Figure 14c.

Based on graphical analysis, the results for SMPC and SMPC-L demonstrate that the inclusion of the angular velocity constraints using chance-constraint decreases the number of constraints violations in comparison to MPC. Further, when the constraints are violated, they are violated by a lower value. The minimization of $\dot{\mathbf{L}}$ reduces the oscillation of the angular velocity. However, a significant change of the constraints violations is not verified by the graphical analysis.

As the inclusion of any constraint may lead to a more conservative controller. And minimizing $\dot{\mathbf{L}}$ implies in a trade-off in the optimization, i.e., the controller will minimize the trajectory error while minimizing $\dot{\mathbf{L}}$. In Figure 16, part of the segment analyzed for the angular velocity is presented for the angular positions. Among each trial, small variations occur in the

Figure 15 – Angular velocity comparison on rough terrain.



(a) ω_x trajectory.

(b) ω_y trajectory.

(c) ω_z trajectory.

Source: the author, 2022.

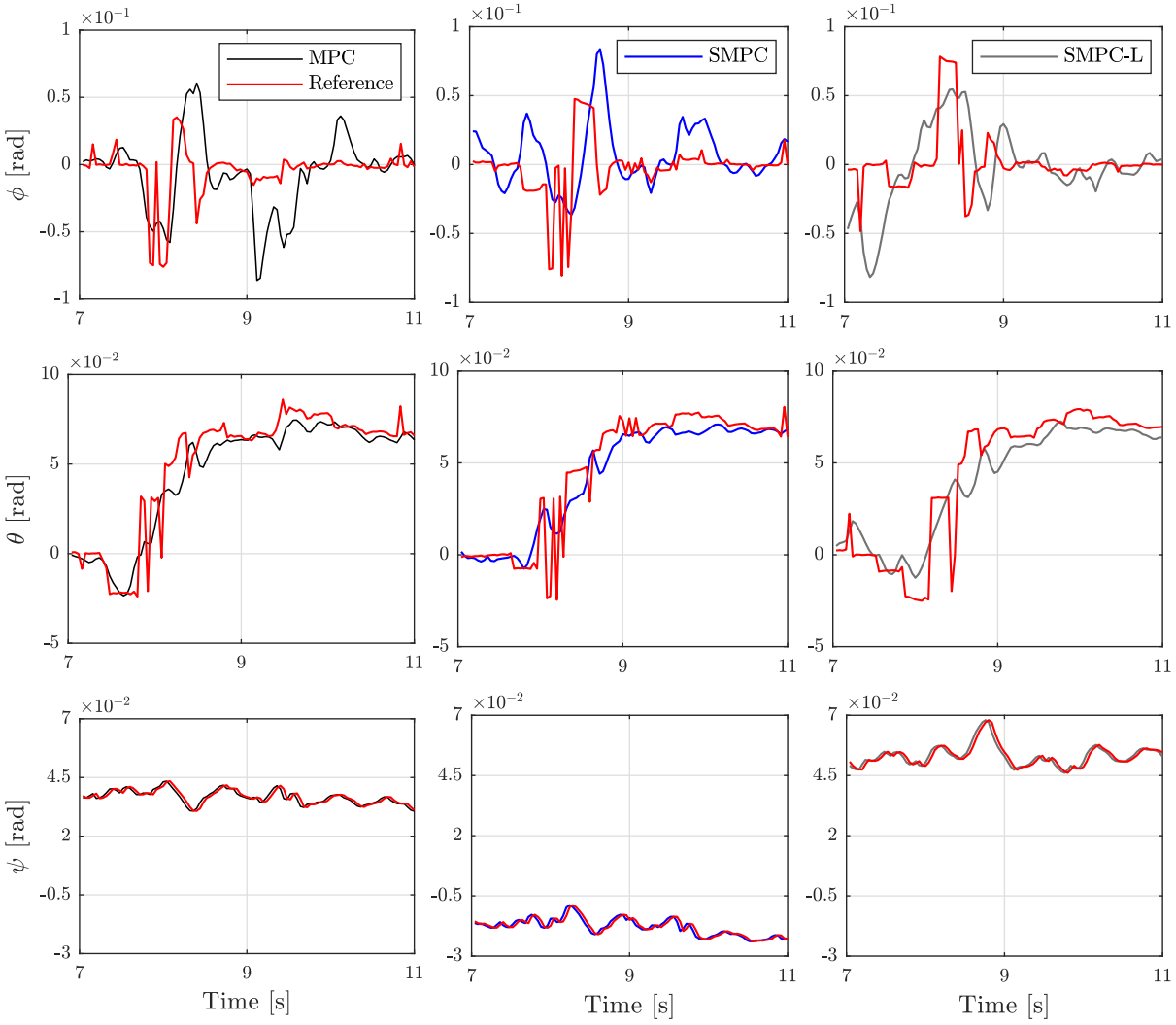
angular reference due to its update based on the terrain estimation and on the current system's state.

For angle ϕ, SMPC and SMPC-L present similar results with a small increase in the tracking error, while the results for θ and ψ are very similar for all the controllers

The results for the angular velocity constraints are evaluated. In Table 2, it is presented the average $\bar{n}_j$ and standard deviation σ_j of the number of constraints violations among all the 30 trials for each jth constraint, where $j = \omega_{x\max}, \omega_{x\min}, \omega_{y_{\max}}, \omega_{y_{\min}}$. In all the tables presented in this chapter, the best results among all the controllers are highlighted in bold.

For ω_x constraints, considering MPC as a baseline, all the cases with angular velocity constraints decreased $\bar{n}_{\omega_{x\max}}$ and $\bar{n}_{\omega_{x\min}}$. SMPC presented the smallest $\bar{n}_{\omega_{x\max}}$, resulting in a reduction of almost 9% with relation to MPC-C and 37% to MPC. In the case of stochastic controllers, minimizing $\dot{\mathbf{L}}$ did not affect significantly the number of constraints violation, but it resulted in the smallest $\sigma_{\omega_{x\max}}$ with a reduction of almost 80% in comparison with MPC. The

Figure 16 – Angular trajectory comparison on rough terrain.



Source: the author, 2022.

Table 2 – Results for the number of constraints violations.

Parameter	MPC	MPC-C	SMPC	MPC-L	MPC-CL	SMPC-L
$\bar{n}_{\omega_{x\max}}$	22.50	15.60	**14.30**	20.83	18.53	14.60
$\sigma_{\omega_{x\max}}$	4.75	4.15	4.77	5.47	4.31	**1.03**
$\bar{n}_{\omega_{x\min}}$	20.83	16.83	14.63	**13.77**	18.93	**13.77**
$\sigma_{\omega_{x\min}}$	**3.86**	4.59	4.72	5.14	4.55	4.61
$\bar{n}_{\omega_{y\max}}$	4.33	1.33	**0.80**	1.57	1.33	0.90
$\sigma_{\omega_{y\max}}$	1.88	3.79	1.19	1.52	1.75	**1.06**
$\bar{n}_{\omega_{y\min}}$	**0.80**	0.83	**0.80**	1.70	2.0	1.70
$\sigma_{\omega_{y\min}}$	1.48	**0.99**	1.16	1.35	1.37	1.34

Source: the author, 2021.

results for $\bar{n}_{\omega_{x\min}}$ are similar to $\bar{n}_{\omega_{x\max}}$, where MPC-L and SMPC-L presented the smallest $\bar{n}_{\omega_{x\min}}$ with a reduction of 33%. For MPC-C and MPC-CL, note that weighting $\dot{\mathbf{L}}$ resulted in an increase of $\bar{n}_{\omega_{x\max}}$ and $\bar{n}_{\omega_{x\min}}$. A possible explanation is that minimizing $\dot{\mathbf{L}}$ contributes to the predicted angular velocity to be smaller based on the prediction model. That way, if unmodeled dynamics

are affecting the system at a given instant, the prediction model may lead to a smaller angular velocity than the real case.

For ω_y, SMPC presented the smallest results for both $\overline{n}_{\omega_{y_{max}}}$ and $\overline{n}_{\omega_{y_{min}}}$. For $\omega_{y_{max}}$, there was a systematic decrease of $\overline{n}_{\omega_{y_{max}}}$ with the inclusion of constraints and the chance-constraints. For $\overline{n}_{\omega_{y_{min}}}$, the inclusion of the angular velocity constraints did not affect the number of violations. In the case of minimizing $\dot{\mathbf{L}}$, it doubled $\overline{n}_{\omega_{y_{min}}}$, which may also occur due to a prediction error in a similar way to MPC-C and MPC-CL for ω_x. However, since the variables are integers, this increase can be considered small, and the values can be considered the same.

In general, the standard deviation did not change significantly for the evaluated controllers, which presented a value of the same order as the average. In the case of ω_y, some controllers presented a value for σ higher than the average. And, since the number of constraints cannot be negative, it demonstrates that in many trials the constraint is not violated.

That way, based on the results described in Table 2, chance-constraints demonstrated to consistently decrease the average number of constraints violations. For the unconstrained cases, minimizing $\dot{\mathbf{L}}$ demonstrated to decrease the average number of constraints violations. While for the stochastic controllers, weighting $\dot{\mathbf{L}}$ did not affect significantly the results.

To evaluate the angular momentum rate, it is analyzed the RMS values of $\dot{\mathbf{L}}$ among all trials. In Table 3, it is presented the average $\overline{\text{RMS}}_{\dot{\mathbf{L}}_j}$ and the standard deviation $\sigma_{\dot{\mathbf{L}}_j}$ of the RMS among all the trials for each associated angular coordinate j, where $j = \phi, \theta, \psi$.

Table 3 – RMS analysis for the angular momentum rate.

Parameter	MPC	MPC-C	SMPC	MPC-L	MPC-CL	SMPC-L
$\overline{\text{RMS}}_{\dot{\mathbf{L}}_\phi}$	7.58	6.69	6.39	7.15	6.67	**6.25**
$\sigma_{\dot{\mathbf{L}}_\phi}$	0.66	0.57	**0.48**	0.69	0.57	0.58
$\overline{\text{RMS}}_{\dot{\mathbf{L}}_\theta}$	17.04	15.02	12.91	10.92	11.08	**10.65**
$\sigma_{\dot{\mathbf{L}}_\theta}$	2.02	1.39	**1.17**	1.45	1.25	1.28
$\overline{\text{RMS}}_{\dot{\mathbf{L}}_\psi}$	6.48	6.39	6.33	5.61	**5.50**	5.54
$\sigma_{\dot{\mathbf{L}}_\psi}$	0.27	0.29	0.20	0.23	**0.18**	0.19

Source: the author, 2021.

For $\dot{\mathbf{L}}_\phi$, while MPC-L presented a value 5% smaller than MPC, minimizing $\dot{\mathbf{L}}$ resulted in almost the same value for the constrained controllers. Still, the stochastic controllers presented a decrease of 15% in $\dot{\mathbf{L}}_\phi$ in comparison with MPC, and 5% for the MPC-C and MPC-L.

For $\dot{\mathbf{L}}_\theta$, the inclusion of the angular velocity constraint decreased $\overline{\text{RMS}}_{\dot{\mathbf{L}}_\theta}$ by 12%, and by 25% for SMPC. Minimizing $\dot{\mathbf{L}}$ demonstrated to decrease by around 35% for all the cases.

Since there are no constraints for ω_z, only weighting $\dot{\mathbf{L}}$ decreased $\dot{\mathbf{L}}_\psi$, which was reduced by at least 13% for all controllers with $\dot{\mathbf{L}}$ minimization.

With the exception of $\sigma_{\dot{\mathbf{L}}_\theta}$ for MPC, the standard deviation presented values in a close range.

To extend the graphical analysis of the trajectory tracking, the RMS of the angular and

linear trajectories errors are analyzed. In Table 4, it is presented the average and the standard deviation of the 30 trials of the RMS error of the angular velocity and position for each controller.

Table 4 – RMS error for angular trajectory tracking.

Parameter	MPC	MPC-C	SMPC	MPC-L	MPC-CL	SMPC-L
$\overline{\text{RMS}}_{\omega_x}(\times 10^{-3})$	200.7	193.4	181.0	202.3	198.7	**179.6**
$\sigma_{\omega_x}(\times 10^{-3})$	**14.90**	17.90	17.80	19.30	16.30	19.30
$\overline{\text{RMS}}_{\omega_y}(\times 10^{-3})$	77.70	75.10	**67.80**	72.40	74.10	68.10
$\sigma_{\omega_y}(\times 10^{-3})$	6.60	**5.10**	6.40	8.50	8.10	7.60
$\overline{\text{RMS}}_{\omega_z}(\times 10^{-3})$	**24.80**	25.20	25.30	31.20	30.60	30.50
$\sigma_{\omega_z}(\times 10^{-3})$	1.60	1.70	**1.40**	2.70	2.90	2.20
$\overline{\text{RMS}}_{\phi}(\times 10^{-3})$	**20.80**	25.80	32.60	21.70	26.90	30.90
$\sigma_{\phi}(\times 10^{-3})$	**1.40**	3.10	6.00	1.90	2.60	3.60
$\overline{\text{RMS}}_{\theta}(\times 10^{-3})$	**9.10**	10.10	11.90	12.00	12.60	13.10
$\sigma_{\theta}(\times 10^{-3})$	**0.90**	1.10	1.90	1.40	1.50	1.70
$\overline{\text{RMS}}_{\psi}(\times 10^{-3})$	**0.90**	**0.90**	**0.90**	1.20	1.20	1.20
$\sigma_{\psi}(\times 10^{-3})$	0.10	0.10	0.10	0.10	0.10	0.10

Source: the author, 2021.

The inclusion of angular velocity constraints decreases the angular velocities at least to the constraints' values, or more due to the chance-constraints. Since the reference for the angular velocities are zero, SMPC and SMPC-L presented the smallest $\overline{\text{RMS}}_{\omega_x}$ and $\overline{\text{RMS}}_{\omega_y}$ with a reduction of about 10%.

Since the inclusion of constraints leads the controller to be more conservative in general, for the angular positions, $\overline{\text{RMS}}_{\phi}$ and $\overline{\text{RMS}}_{\theta}$ presented a systematic increase with the increment of the controllers' conservatism.

With the exception of $\overline{\text{RMS}}_{\omega_y}$, the trade-off by minimizing $\dot{\mathbf{L}}$ in the cost functional is verified by a systematic increase of $\overline{\text{RMS}}$, and accordingly σ, for all the variables.

The variations in σ relay in a close range, around 3% of the $\overline{\text{RMS}}$ or 7% in the worst-case identified when comparing SMPC and SMPC-L.

For the linear trajectory, $\overline{\text{RMS}}$ and σ did not change significantly with the inclusion of the angular velocity constraints and the modification in the cost functional, as verified in Table 5. Since the included constraints and $\dot{\mathbf{L}}$ are directly related to the angular position and velocity, this result was expected.

The average and standard deviation of the cost functional value $\mathbf{J}_c$ and the computational time t_c, $\overline{\mathbf{J}_c}$, $\sigma_{\mathbf{J}_c}$, $\overline{t_c}$, and σ_{t_c}, respectively, are presented in Table 6.

Due to the different cost functional used, the controllers that do not consider $\dot{\mathbf{L}}$ are not comparable with the ones that minimize it. For both cases, the cost functional is larger for the stochastic controllers, which is expected since they are naturally more conservative. All the MPCs presented the same cost with a similar standard deviation.

Table 5 – RMS error for linear trajectory tracking.

Parameter	MPC	MPC-C	SMPC	MPC-L	MPC-CL	SMPC-L
$\overline{\text{RMS}}_{\dot{x}}(\times 10^{-3})$	21.70	21.70	22.90	**21.50**	22.90	22.40
$\sigma_{\dot{x}}(\times 10^{-3})$	**1.50**	2.80	2.60	2.50	2.70	2.10
$\overline{\text{RMS}}_{\dot{y}}(\times 10^{-3})$	35.90	35.90	37.50	**35.30**	37.50	36.80
$\sigma_{\dot{y}}(\times 10^{-3})$	**1.90**	3.10	3.10	3.40	3.20	2.80
$\overline{\text{RMS}}_{\dot{z}}(\times 10^{-3})$	**29.90**	30.50	30.90	30.80	30.80	30.70
$\sigma_{\dot{z}}(\times 10^{-3})$	**0.80**	1.10	1.10	1.10	1.30	1.50
$\overline{\text{RMS}}_{x}(\times 10^{-3})$	**17.70**	17.80	17.80	17.80	**17.70**	17.80
$\sigma_{x}(\times 10^{-3})$	0.10	0.10	0.10	0.10	0.10	0.10
$\overline{\text{RMS}}_{y}(\times 10^{-3})$	**1.60**	**1.60**	1.70	**1.60**	1.70	1.70
$\sigma_{y}(\times 10^{-3})$	**0.10**	0.20	**0.10**	0.20	0.20	0.20
$\overline{\text{RMS}}_{z}(\times 10^{-3})$	**19.40**	19.90	20.70	20.20	21.10	20.80
$\sigma_{z}(\times 10^{-3})$	**1.60**	2.20	2.30	2.30	2.20	2.90

Source: the author, 2021.

Table 6 – Cost functional and computational time analysis.

Parameter	MPC	MPC-C	SMPC	MPC-L	MPC-CL	SMPC-L
$\overline{J_c}$	**-2582.3**	**-2582.3**	-2581.8	-2582.5	**-2582.8**	-2581.6
σ_{J_c}	**0.774**	1.029	1.102	1.205	1.168	1.153
$\overline{t_c}$ $[ms]$	**6.50**	6.60	6.90	6.60	6.80	7.00
σ_{t_c} $[ms]$	0.215	0.270	0.260	0.234	0.190	**0.159**

Source: the author, 2021.

The computation time is defined by the time between the framework publishes into a ROS topic to request the MPC to compute a new wrench and the time to receive the new wrench. This time is not related only to the controller, but it is also subject to delays in the ROS communication. However, this time is analyzed since it defines the entire time between the start of a new control step and the control input is applied. Also, as the computer used for simulation does not have a real-time feature, the computation time is subject to the system dealing with other processes that are not related to the controller or the framework.

The stochastic controllers need to update the tightened terms due to the update of the prediction model at each control step. That way, it was verified an increase of about 4% in the computation time of the stochastic controllers in comparison with the controllers based on the nominal model.

5.3 CHAPTER SUMMARY

In the simulation results, the inclusion of angular velocity constraints demonstrated to not only affect the angular velocity but also decrease $\dot{\mathbf{L}}$, resulting in a smoother trajectory.

The use of chance-constraints demonstrated to systematically decrease the number of constraints violations with a minimal increase in the controllers' conservatism and without

affecting significantly the trajectory tracking.

Due to the update of the matrices from the prediction model, the tightened terms related to the chance-constraints cannot be computed offline. Hence, they are updated at each control step, increasing the computational cost. However, this extra computational cost was verified to be small and did not affect the overall results.

The inclusion of the $\dot{\mathbf{L}}$ in the cost functional demonstrated to improve the smoothness of the robot trajectory. For example, $\overline{\text{RMS}}_{\dot{\mathbf{L}}_\phi}$ and $\overline{\text{RMS}}_{\dot{\mathbf{L}}_\theta}$ were decreased by at least 5% and 35%, respectively, in comparison with the MPC without angular velocity constraints and $\dot{\mathbf{L}}$ weighting.

The combination of the chance-constraint and the minimization of $\dot{\mathbf{L}}$, i.e, SMPC-L, demonstrated to be mutually beneficial: weighting $\dot{\mathbf{L}}$ decreases the number of constraints violations, while the chance-constraints decreases the $\overline{\text{RMS}}_{\dot{\mathbf{L}}}$.

6 CONCLUSION

This work compares MPC and SMPC for legged robot locomotion on rough terrains. It also evaluated the minimization of the angular momentum rate, due to its association with the smoothness of the robot trajectory. As main contribution, the consideration of the presence of uncertainties in the system's description demonstrated to improve the safety and to provide a more natural behavior to the robot locomotion.

To evaluate the proposed modifications, six controllers were designed and compared in simulation experiments. The results were analyzed based on the number of constraints violations and the root mean squared of the trajectory error and the angular momentum rate $\dot{\mathbf{L}}$.

The inclusion of angular velocity constraints demonstrated to improve the locomotion safety even when the motion plan is not properly defined based on the robot characteristics. Furthermore, generate reference trajectories for the CoM not based only in the terrain morphology and kinematic characteristics, but also considering dynamics criteria is indicated as future research. The formulation of the angular velocity constraints as chance-constraints was verified to efficiently consider the presence of uncertainty in the system description. The chance-constraints improved the system's robustness in the simulation experiments, without impacting significantly the trajectory tracking, despite the additional conservatism. The extra computational cost related to the update of the chance-constraints tightened was verified to not be significant and to not impact the overall performance.

The inclusion of $\dot{\mathbf{L}}$ in the cost functional improved the smoothness of the angular trajectory without compromising the trajectory tracking, while it contributed to decrease the number of constraints violations. The combination of the SMPC and the $\dot{\mathbf{L}}$ minimization demonstrated to merge the benefits of handling the uncertainty in the system and to smooth the trajectory, without compromising the trajectory tracking.

One of the key-points of stochastic approaches is the assumption about the uncertainty. In this work, the system's uncertainty was considered as additive in the model with a Gaussian distribution, which was estimated from the simulation results of the MPC. However, the considered distribution type and estimation method did not completely explain the model's uncertainty. Accordingly, possible directions for future researches are the use of system identification and learning based techniques to estimate the system's model uncertainty and compensate it in the controller design.

Due to additional uncertainties in the real robot, e.g., measurement noises and parametric uncertainty, a natural future work is the evaluation of the proposed controllers in the HyQ robot.

Additional contributions of this work are the detailed description of the control architecture from the HyQ robot, which can be employed for other legged robots; the review of the main aspects and approaches of stochastic predictive controllers; the proposed modifications in the state-space representation for dynamic locomotion of legged robots employed by predictive controllers; and the formulation of an SMPC for legged robot locomotion.

BIBLIOGRAPHY

ARUL, S. H.; MANOCHA, D. Swarmcco: Probabilistic reactive collision avoidance for quadrotor swarms under uncertainty. **IEEE Robotics and Automation Letters**, v. 6, n. 2, p. 2437–2444, 2021. Cited in page 22.

BARASUOL, V. **Um sistema de controle reativo para locomoção de robôs quadrúpedes**. PhD Thesis — Universidade Federal de Santa Catarina (UFSC), Florianópolis, Santa Catarina, 2013. Available at: <https://repositorio.ufsc.br/bitstream/handle/123456789/123097/324099.pdf?sequence=1>. Accessed in: 20 mar. 2019. Cited in page 39.

BARASUOL, V. et al. A reactive controller framework for quadrupedal locomotion on challenging terrain. p. 2554–2561, 2013. Cited 2 times in pages 24 and 36.

BARASUOL, V. et al. Highly-integrated hydraulic smart actuators and smart manifolds for high-bandwidth force control. **Frontiers in Robotics and AI**, v. 5, p. 51, 2018. ISSN 2296-9144. Cited in page 24.

BLACKMORE, L. et al. A probabilistic particle-control approximation of chance-constrained stochastic predictive control. **IEEE Transactions on Robotics**, v. 26, n. 3, p. 502–517, 2010. Cited in page 32.

BLACKMORE, L.; ONO, M.; WILLIAMS, B. C. Chance-constrained optimal path planning with obstacles. **IEEE Transactions on Robotics**, v. 27, n. 6, p. 1080–1094, 2011. Cited in page 29.

BLEDT, G. et al. Mit cheetah 3: Design and control of a robust, dynamic quadruped robot. In: **2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2018. p. 2245–2252. Cited in page 24.

BOYD, S.; VANDENBERGHE, L. **Convex optimization**. Cambridge, UK New York: Cambridge University Press, 2004. ISBN 9780521833783. Cited in page 29.

CALAFIORE, G. C.; FAGIANO, L. Robust model predictive control via scenario optimization. **IEEE Transactions on Automatic Control**, v. 58, n. 1, p. 219–224, 2013. Cited in page 29.

CAMACHO, E. F.; ALBA, C. B. **Model predictive control**. [S.l.]: Springer-Verlag London, 2007. Cited in page 25.

CAMPI, M. C.; GARATTI, S.; PRANDINI, M. The scenario approach for systems and control design. **Annual Reviews in Control**, v. 33, n. 2, p. 149–157, 2009. ISSN 1367-5788. Cited in page 32.

CAMPION, G.; CHUNG, W. Wheeled robots. In: **Springer handbook of robotics**. [S.l.]: Springer Berlin Heidelberg, 2008. p. 391–410. Cited in page 21.

CANNON, M.; KOUVARITAKIS, B.; NG, D. Probabilistic tubes in linear stochastic model predictive control. **Systems & Control Letters**, v. 58, n. 10, p. 747–753, 2009. ISSN 0167-6911. Cited in page 31.

CARLO, J. D. et al. Dynamic locomotion in the MIT Cheetah 3 through convex model-predictive control. In: **2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2018. p. 1–9. Cited 3 times in pages 21, 24, and 40.

CUNHA, T. B. **Hydraulic compliance control of the quadruped robot HyQ**. PhD Thesis — Istituto Italiano di Tecnologia (IIT) and University of Genova, Genoa, Italy, 2013. Available at: <https://iit-dlslab.github.io/papers/cunha13phdthesis.pdf>. Accessed in: 30 dez. 2021. Cited 2 times in pages 45 and 46.

DAI, H.; VALENZUELA, A.; TEDRAKE, R. Whole-body motion planning with centroidal dynamics and full kinematics. In: **2014 IEEE-RAS International Conference on Humanoid Robots**. [S.l.: s.n.], 2014. p. 295–302. Cited in page 24.

FANKHAUSER, P. et al. Robust rough-terrain locomotion with a quadrupedal robot. In: **2018 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2018. p. 5761–5768. Cited in page 24.

FARINA, M. et al. An approach to output-feedback mpc of stochastic linear discrete-time systems. **Automatica**, v. 55, p. 140–149, 2015. ISSN 0005-1098. Cited in page 31.

FARINA, M.; GIULIONI, L.; SCATTOLINI, R. Stochastic linear model predictive control with chance constraints – a review. **Journal of Process Control**, v. 44, p. 53 – 67, 2016. Cited 5 times in pages 28, 29, 30, 31, and 32.

FEATHERSTONE, R. **Rigid body dynamics algorithms**. [S.l.]: Springer US, 2008. Cited in page 40.

FOCCHI, M. et al. High-slope terrain locomotion for torque-controlled quadruped robots. **Autonomous Robots**, Springer Science and Business Media LLC, v. 41, n. 1, p. 259–272, 2016. Cited in page 42.

GASPERO, L. D.; MOYER, E. **Quadprog++**. 2021. Available at: <https://github.com/liuq/QuadProgpp>. Cited in page 54.

GAZAR, A. et al. Stochastic and robust mpc for bipedal locomotion: A comparative study on robustness and performance. **2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)**, p. 61–68, 2021. Cited in page 22.

GAZEBO. **Gazebo - Robot simulation made easy**. 2019. Available at: <http://gazebosim.org/>. Accessed in: 20 mar. 2019. Cited in page 54.

GOLDFARB, D.; IDNANI, A. A numerically stable dual method for solving strictly convex quadratic programs. **Mathematical Programming**, Springer Science and Business Media LLC, v. 27, n. 1, p. 1–33, set. 1983. Cited in page 54.

GRAY, A. et al. Stochastic predictive control for semi-autonomous vehicles with an uncertain driver model. In: **16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)**. [S.l.: s.n.], 2013. p. 2329–2334. Cited in page 22.

HESSEM, D. van; BOSGRA, O. A conic reformulation of model predictive control including bounded and stochastic disturbances under state and input constraints. In: **Proceedings of the 41st IEEE Conference on Decision and Control, 2002.** [S.l.: s.n.], 2002. v. 4, p. 4643–4648 vol.4. Cited in page 32.

HEWING, L.; KABZAN, J.; ZEILINGER, M. N. Cautious model predictive control using gaussian process regression. **IEEE Transactions on Control Systems Technology**, IEEE, v. 28, n. 6, p. 2736–2743, 2019. Cited in page 22.

HUTTER, M. et al. Anymal - a highly mobile and dynamic quadrupedal robot. In: **2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2016. p. 38–44. Cited in page 24.

IIT, I. I. of T. **HyQ**. 2021. Available at: <https://robots.ieee.org/robots/hyq/>. Cited in page 34.

KATZ, B.; CARLO, J. D.; KIM, S. Mini cheetah: A platform for pushing the limits of dynamic quadruped control. In: **2019 International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2019. p. 6295–6301. Cited in page 24.

KHORRAM, M.; MOOSAVIAN, S. A. A. Push recovery of a quadruped robot on challenging terrains. **Robotica**, Cambridge University Press (CUP), v. 35, n. 8, p. 1670–1689, jun. 2016. Cited in page 43.

KOUVARITAKIS, B.; CANNON, M. **Model predictive control**. [S.l.]: Springer International Publishing, 2016. Cited 3 times in pages 22, 27, and 31.

KOUVARITAKIS, B. et al. Explicit use of probabilistic distributions in linear predictive control. **Automatica**, v. 46, n. 10, p. 1719–1724, 2010. ISSN 0005-1098. Cited in page 52.

LANGSON, W. et al. Robust model predictive control using tubes. **Automatica**, v. 40, n. 1, p. 125–133, 2004. ISSN 0005-1098. Cited in page 31.

LIBERZON, D. **Switching in systems and control**. [S.l.]: Birkhäuser Boston, 2003. Cited 2 times in pages 42 and 50.

MAMMARELLA, M. et al. Sample-based smpc for tracking control of fixed-wing uav. **IEEE Control Systems Letters**, v. 2, n. 4, p. 611–616, 2018. Cited in page 22.

MASTALLI, C. et al. Motion planning for quadrupedal locomotion: coupled planning, terrain mapping, and whole-body control. **IEEE Transactions on Robotics**, v. 36, n. 6, p. 1635–1648, 2020. Cited in page 24.

MAYNE, D.; SERON, M.; RAKOVIć, S. Robust model predictive control of constrained linear systems with bounded disturbances. **Automatica**, v. 41, n. 2, p. 219–224, 2005. ISSN 0005-1098. Cited in page 31.

MESBAH, A. Stochastic model predictive control: An overview and perspectives for future research. **IEEE Control Systems Magazine**, v. 36, n. 6, p. 30–44, Dec 2016. ISSN 1941-000X. Cited 4 times in pages 22, 27, 28, and 31.

MESBAH, A. et al. Stochastic nonlinear model predictive control with probabilistic constraints. In: **2014 American Control Conference**. [S.l.: s.n.], 2014. p. 2413–2419. Cited in page 32.

MUYBRIDGE, E. **Animals in motion**. New York: Dover Publications, 1957. Cited in page 23.

NOBILI, S. et al. Heterogeneous sensor fusion for accurate state estimation of dynamic legged robots. In: **Robotics: Science and Systems**. [S.l.: s.n.], 2017. Cited in page 35.

OLDEWURTEL, F.; JONES, C. N.; MORARI, M. A tractable approximation of chance constrained stochastic MPC based on affine disturbance feedback. In: **2008 47th IEEE Conference on Decision and Control**. [S.l.: s.n.], 2008. p. 4731–4736. Cited in page 32.

ORIN, D. E.; GOSWAMI, A.; LEE, S.-H. Centroidal dynamics of a humanoid robot. **Autonomous Robots**, Springer Science and Business Media LLC, v. 35, n. 2-3, p. 161–176, jun. 2013. Cited 3 times in pages 40, 41, and 48.

RAIBERT, M. **Legged robots that balance**. Cambridge, Mass: MIT Press, 1986. ISBN 9780262181174. Cited in page 21.

REMY, C. D. **Optimal exploitation of natural dynamics in legged locomotion**. PhD Thesis — ETH Zurich, Zurich, Switzerland, 2011. Available at: <http://hdl.handle.net/20.500.11850/152887>. Accessed in: 12 dez. 2021. Cited in page 21.

SCHILDBACH, G. et al. The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations. **Automatica**, v. 50, n. 12, p. 3009–3018, 2014. Cited 2 times in pages 28 and 32.

SEMINI, C. et al. Brief introduction to the quadruped robot HyQReal. **Istituto di Robotica e Macchine Intelligenti (I-RIM)**, 2019. Cited in page 24.

SEMINI, C. et al. 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**, v. 225, n. 6, p. 831–849, 2011. Cited 2 times in pages 24 and 34.

SIEGWART, R. **Introduction to autonomous mobile robots**. Cambridge, Mass: MIT Press, 2004. ISBN 9780262195027. Cited in page 21.

SLEIMAN, J.-P. et al. A unified MPC framework for whole-body dynamic locomotion and manipulation. **IEEE Robotics and Automation Letters**, v. 6, n. 3, p. 4688–4695, 2021. Cited 2 times in pages 21 and 24.

VILLARREAL, O. **Bridging vision and dynamic legged locomotion**. PhD Thesis — Istituto Italiano di Tecnologia (IIT) and University of Genova, Genoa, Italy, 2020. Available at: <https://iit-dlslab.github.io/papers/villarreal20phd.pdf>. Accessed in: 20 dez. 2021. Cited 9 times in pages 21, 23, 35, 36, 37, 40, 42, 43, and 47.

VILLARREAL, O. et al. Fast and continuous foothold adaptation for dynamic locomotion through cnns. **IEEE Robotics and Automation Letters**, v. 4, n. 2, p. 2140–2147, 2019. Cited 2 times in pages 35 and 37.

VILLARREAL, O. et al. MPC-based controller with terrain insight for dynamic legged locomotion. In: **2020 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2020. p. 2436–2442. Cited in page 24.

WANG, L. **Model predictive control system design and implementation using MATLAB®**. [S.l.]: Springer-Verlag London, 2009. Cited in page 25.

ZHANG, X. et al. Stochastic model predictive control using a combination of randomized and robust optimization. In: **52nd IEEE Conference on Decision and Control**. [S.l.: s.n.], 2013. p. 7740–7745. Cited in page 32.