DLR-IB-RM-OP-2017-229

Balancing and Static Walking Control for a Compliantly Actuated Quadruped Exploiting System Inherent Elasticities

Masterthesis

Yuri Federigi

Deutsches Zentrum
DLR für Luft- und Raumfahrt

Institut für Robotik und Mechatronik	BJ.: 2017
	IB.Nr.: DLR-IB-RM-OP-2017-229

MASTERTHESIS

BALANCING AND STATIC WALKING CONTROL FOR A COMPLIANTLY ACTUATED QUADRUPED EXPLOITING SYSTEM INHERENT ELASTICITIES

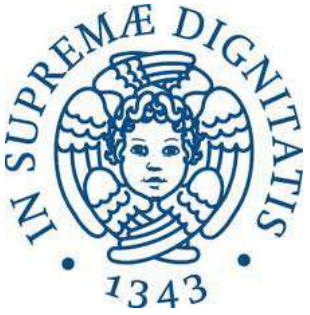
Freigabe:	Der Bearbeiter:	Unterschriften
	Yuri Federigi	
	Betreuer:	
	Dominic Lakatos	
	Der Institutsdirektor	
	Prof. Alin Albu-Schäffer	

Dieser Bericht enthält 68 Seiten, 62 Abbildungen und 7 Tabellen

Ort: Oberpfaffenhofen	Datum: 25.11.2017	Bearbeiter: Yuri Federigi	Zeichen:

Università di Pisa

Facoltà di Ingegneria



Laurea Magistrale in Ingegneria
Robotica e dell'Automazione

Balancing and Static Walking Control for a Compliantly Actuated Quadruped Exploiting System Inherent Elasticities

Relatori:

Prof. Antonio Bicchi

Dominic Lakatos

Controrelatore:

Prof. Marco Gabiccini

Candidato:

Yuri Federigi

Anno Accademico 2016/2017

Abstract

Soft actuators have been designed as a promising novel technology to achieve robust and natural-like robotic behaviors.

The present thesis addresses the control of a quadrupedal robot, equipped with series elastic actuators (SEA), to achieve balancing and quasi-static locomotion on rough terrains. Thousands of years of biological evolution proofed static gaits, in combination with the elasticity provided by muscles and tendons, to be the most effective ones on challenging surfaces.

The balancing of walking machines has to deal with different discontinuous and highly non-linear contact situations; additionally, SEAs provide a decoupling between actuation and end-effectors. Overcoming the latter with a torque feedback at joints level, which reshapes the springs dynamics, showed robustness issues in real world applications, especially for highly compliant systems, which are designed to efficiently execute highly dynamic locomotion tasks. Therefore, in this context, we present a methodology which exploits directly the inherent spring dynamics by controlling the system equilibrium point.

The locomotion task is then fulfilled designing a step planner which, taking advantage of an intuitive set of task coordinates, uses only contact detection and the robot attitude to sense the terrain surface and properly control the load distribution during the walking phases succession.

The proposed method is evaluated in simulated environments with and without obstacles, to compare the achieved performances and, finally, tested during experiments on the DLR's quadrupedal robot, Bert, showing a consistent robustness on a variety of ground surfaces.

Sommario

Gli attuatori dotati di cedevolezza passiva costituiscono una tecnologia innovativa, progettata per ottenere robot robusti e con movimenti naturali.

Questa tesi presenta il sistema di controllo di un robot quadrupede, dotato di *series elastic actuators* (SEA), capace di affrontare la locomozione quasi-statica su terreni sconnessi. Millenni di evoluzione biologica hanno dimostrato che le andature statiche, in combinazione con l'elasticità garantita da muscoli e tendini, sono le più performanti su superfici difficili.

Al fine di mantenere l'equilibrio, il sistema di controllo di un robot in grado di camminare deve districarsi agevolmente tra situazioni di contatto differenti, discontinue ed altamente non lineari; inoltre, l'utilizzo di SEA introduce un disaccoppiamento tra l'attuazione ed i punti di contatto con il terreno. Il superamento di quest'ultimo attraverso la chiusura di un *feedback* di coppia a livello dei giunti si è dimostrato essere non particolarmente robusto in applicazioni reali, specialmente per sistemi ad elevata cedevolezza, progettati per affrontare efficentemente compiti altamente dinamici. Dunque, in questo contesto, presentiamo una metodologia che sfrutta direttamente le dinamiche naturali delle molle controllando il punto di equilibrio del sistema.

Successivamente, la locomozione è implementata attraverso uno *step planner* che, avvalendosi di un intuitivo set di coordinate di *task*, usa solamente la percezione del contatto per modellare la superficie del terreno e controllare adeguatamente la distribuzione del carico durante l'alternanza delle fasi di locomozione.

Il metodo proposto viene valutato in ambienti simulati con e senza ostacoli, per poterne confrontare le prestazioni e, in fine, testato sul robot quadrupede di proprietà del DLR, Bert, dimostrando una robustezza considerevole su diversi tipi di terreno.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The idea of building a robotic system capable of step out a factory building and emulate similar performances of living beings has always fascinated mankind.

A particularly promising approach to achieve such a results is to apply the bio inspired concept of soft actuation, where elastic elements play the role of muscles and tendons in mammals bodies.

However, this choice leads to the need of different control systems, not only to overcome the increased complexity introduced by the elastic decoupling and by the increased size of the dynamics but also to fully exploit the advantages of this new paradigm.

1.1 Motivation and State of the Art

During the past decades, the expertise developed in the industrial environment let the robotics community develop legged machines which substantially reduced the gap with their biological counterparts. Quadrupedal robots, as Titan ([4], Fig. 1.1b) and Little Dog ([2], Fig. 1.1a), make use of common actuation principles, where the structure is rigidly connected with the contact surfaces driving all the harmful peak loads produced during locomotion directly to the gearboxes. A different choice is the use of hydraulic actuation principles, such as in Big Dog ([5], Fig. 1.2a) and HyQ ([13], Fig. 1.2b), which, however, is well known to be robust and powerful but energetically highly inefficient.

(a) Little Dog **(b)** Titan

Figure 1.1: Examples of rigid quadrupedal robots

Source: Boston Dynamics, Tokyo Institute of Technology

(a) Big Dog **(b)** Hyq

Figure 1.2: Examples quadrupedal robots with hydraulic actuation

Source: Boston Dynamics, Italian Institute of Technology

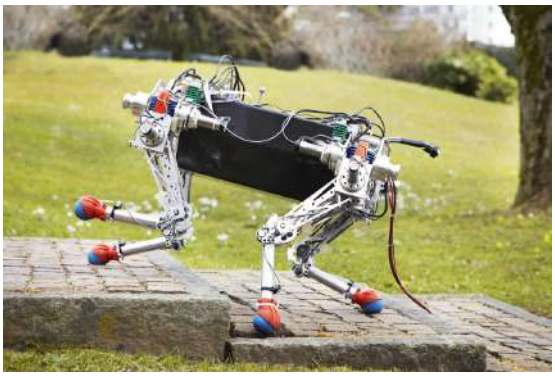


Figure 1.3: StarlETH quadruped robot with SEAs

Source: ETH Zurich

To cope with unstructured environments, this robots have to be equipped with a perfect planning, a considerable number of sensors and high gains controllers to completely cancel any undesirable dynamic effects. However, highly dynamical maneuvers highlighted a lack of robustness due to the bandwidth limited response of the control system ([7]) in combination with the impossibility to passively adapt to unexpected disturbances. Moreover, the not decoupled joints and links bring the inertia of the rigid body directly to the end effector in an unidirectional energy exchange with remarkable safety drawbacks ([6]).

It has been shown (StarlETH, Fig 1.3, [3]) that the inclusion of passive elastic elements, i.e. springs, between links and joints (*series elastic actuator*, [9]) allows the system to deal with the presented issues. On the other hand, besides doubling the size of the dynamics, the decoupling introduced with this new structural conception increased the challenge of developing sustainable controllers from the computational cost point of view.

The increasing need to deal with more and more uncertainties and unstructured terrains required this machines to switch from the *versatility through complexity* to the *versatility through simplicity* paradigm. Considering what has been reported, our aim is to provide a new minimalist and robust way of controlling robots which make use of series elastic actuators to achieve safe and robust locomotion without relying on any a priori information about the terrain.

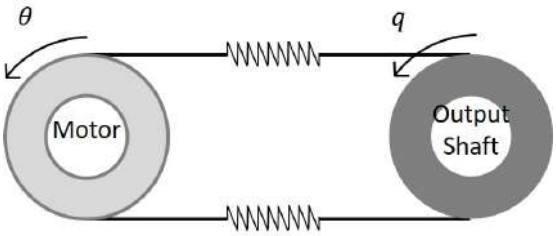


Figure 1.4: SEA sketch

1.2 Main idea

From a first level of approximation, introducing elastic components between links and joints corresponds to create a spring-mass system where one has direct control over one side of the spring θ but not on the position of the other side q (i.e. the spring elongation), w.r.t. Fig. 1.4.

In this context, considering the task of controlling the final mass position, without interfering with the spring inherent dynamics, it is possible to compute the elastic potential energy which, counterbalancing the gravity potential energy, realizes a desired steady state mass position q, Fig 1.5. Simplistically speaking, this method is equivalent to have indirect control on the equilibrium point around which the system is let free to oscillate.

Referring to Fig 1.5b, the dynamics of the system is

$$m\ddot{q} = mg - k(q - \theta) - c\dot{q}, \tag{1.1}$$

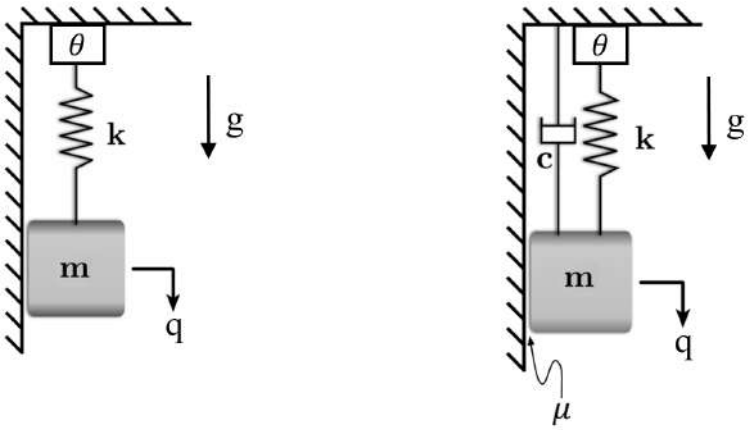
hence, in steady state conditions

$$mg = k(q - \theta). \tag{1.2}$$

From which it is possible to compute the motor position that realize a desired mass position q_{des} by setting the system equilibrium point,

$$\theta = q_{des} - \frac{mg}{k} \tag{1.3}$$

Since every physical system is non conservative, due to the damping effect of friction, the mass converges at the desired position, as shown in Fig. 1.6.



(a) Spring-mass general system

(b) Spring-mass system with friction

Figure 1.5: Sketch of the main idea

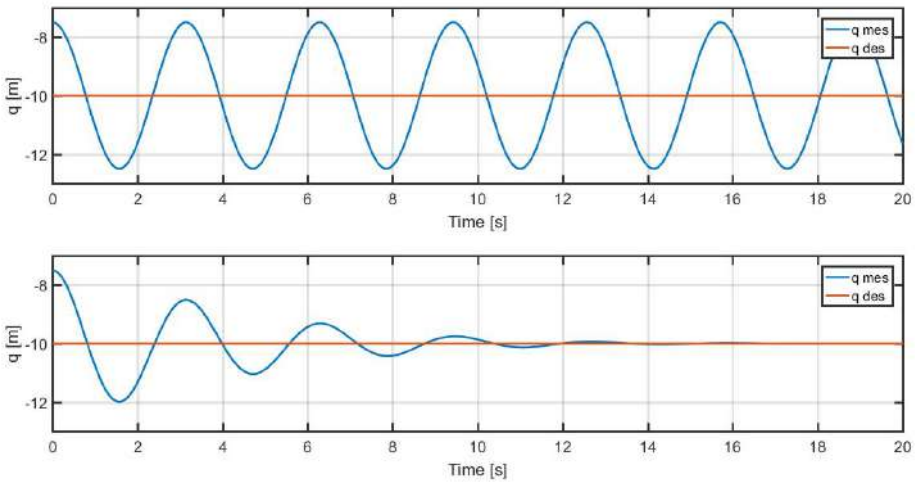


Figure 1.6: Spring-mass system simulation examples without friction (top) and with friction (bottom)

1.3 Bert, the quadruped

The derived framework has been tested on the DLR robotic platform Bert, Fig. 1.7.

Bert is a quadrupedal robot equipped with SEAs. The distances between the shoulders are $l_{hs} = 0.3m$ (hip to shoulder) and $l_{hh} = 0.18m$ (hitp to hip, which is equal to shoulder to shoulder). It is equipped with angular position sensors at motor and joint level and with an IMU which has been used to sense the Tait–Bryan angles, i.e. the attitude of the trunk $roll = \varphi$ and $pitch = \vartheta$ w.r.t. the gravity field.

Each leg is a planar structure with two DOF actuated by two *Savöx SV 1270 TG* servo motors which are connected to the correspondent output shaft by two torsional springs with constant $k = 1.75\frac{Nm}{rad}$. Motors and springs are allocated inside the trunk and the motion is transmitted by a belt-pulley mechanism, as depicted in Fig. 1.8. Shank and thigh are $l_1 = l_2 = 0.08m$, hence, the maximal extension of the leg is $0.16m$, which, however, must not be reached to evade singular configurations. An additional pulley is placed in the shoulder to couple the shoulder joint angles q_1 and the knee joint angles $q_2 - q_1$, where q_2 is the upper output shaft angle.

The trunk weight is $2.2Kg$, the thigh one is $0.05Kg$ and the shank weights $0.03Kg$; hence the dynamical effects of the legs swing are neglectable w.r.t. the trunk mass.

The long term purpose of the presented robot is lean and autonomous planetary exploration, meanwhile it can be efficiently used for search and rescue surveys in unsafe environments, such as buildings affected by serious natural disasters.

Figure 1.7: Bert, the DLR quadruped

Figure 1.8: Bert, a glimpse of the internal elements

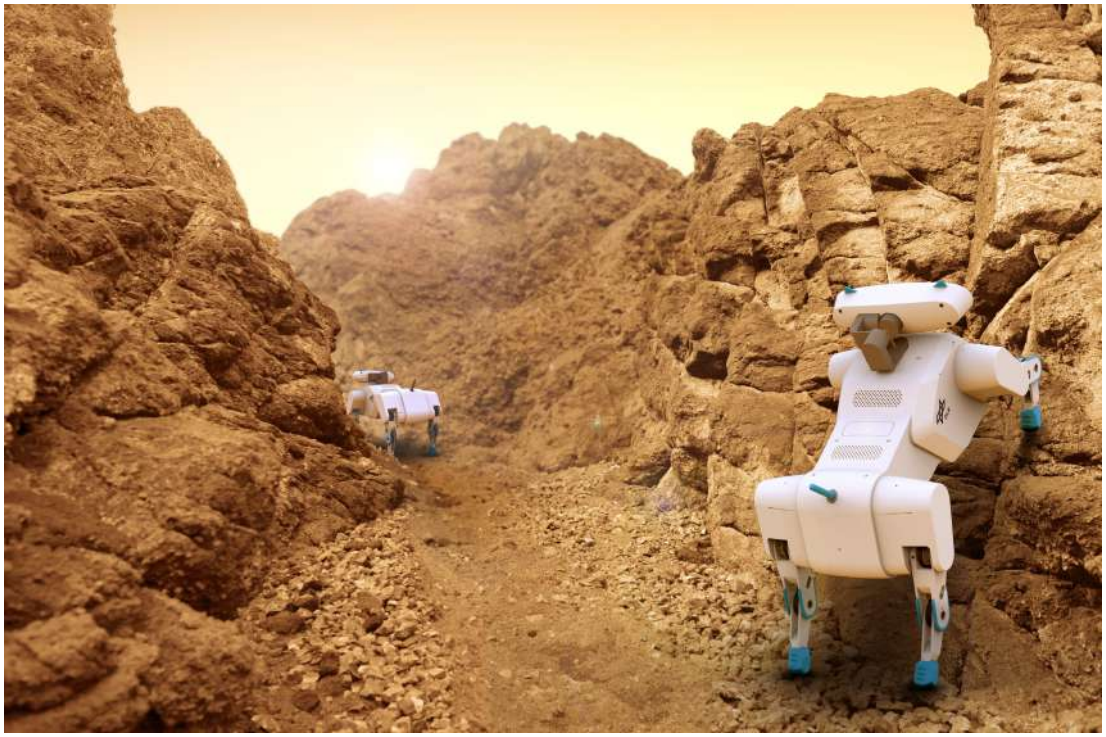


Figure 1.9: Bert, long term application as a lean and autonomous planetary explorer

Chapter 2

Modeling

In this chapter we are going to provide some basic analysis tools for the presented robotic system which have been used extensively to derive the balance and locomotion control further presented in Chap. 3 and 4.

2.1 Forward Kinematics

In this section we are going to use the common rototranslation matrices $H \in SE(3)$ to express the kinematics analysis of the system structure.
The *forward kinematics* of the robot can be analyzed taking into account that the legs structure is the same for each leg.
It is important to notice that the system has no DOF in y direction and around the z axis (yaw).

The reference plane $z = 0$ is the plane passing through the feet positions at the initial time and it is always perpendicular to the gravity field. The *World frame* lies on the reference plane, in the intersection of the two diagonal of the convex hull of the feet projection on the plane. The z axis is aligned with the gravity and the x axis is parallel to the projection of the longitudinal axis of the robot; the y axis is the consequence of the application of the right hand rule. All the further heights are computed along the gravity vector.

Tab. 2.1 and 2.2 summarize the $SE(3)$ matrices which compose the *forward*

H_B^W	$T_x(x_b)T_z(z_b)R_x(\varphi)R_y(\vartheta)$
$H_{S_{F_r}}^B$	$T_x(\frac{l_{hs}}{2})T_y(-\frac{l_{hh}}{2})R_y(q_{1_{F_r}})$
$H_{S_{F_l}}^B$	$T_x(\frac{l_{hs}}{2})T_y(\frac{l_{hh}}{2})R_y(q_{1_{F_l}})$
$H_{S_{R_l}}^B$	$T_x(-\frac{l_{hs}}{2})T_y(\frac{l_{hh}}{2})R_y(q_{1_{R_l}})$
$H_{S_{R_r}}^B$	$T_x(-\frac{l_{hs}}{2})T_y(-\frac{l_{hh}}{2})R_y(q_{1_{R_r}})$

Table 2.1: *Forward kinematics* of the shoulders/hips

H_S^K	$T_z(-l_1)R_y(q_2 - q_1)$
H_K^F	$T_z(-l_2)$

Table 2.2: *Forward kinematics* of the single leg

kinematics, where $T_i(k)$ models a translation of k along the axis i and $R_i(k)$ is a rotation around the axis i by the angle k.

The coordinates vectors used in the following treatise are
$x = [x_b \quad z_b \quad \varphi \quad \vartheta]^T$,
$q = [q_{1_{F_r}} \quad q_{2_{F_r}} \quad q_{1_{F_l}} \quad q_{2_{F_l}} \quad q_{1_{R_l}} \quad q_{2_{R_l}} \quad q_{1_{R_r}} \quad q_{2_{R_r}}]^T$ and
$\theta = [\theta_{1_{F_r}} \quad \theta_{2_{F_r}} \quad \theta_{1_{F_l}} \quad \theta_{2_{F_l}} \quad \theta_{1_{R_l}} \quad \theta_{2_{R_l}} \quad \theta_{1_{R_r}} \quad \theta_{2_{R_r}}]^T$.

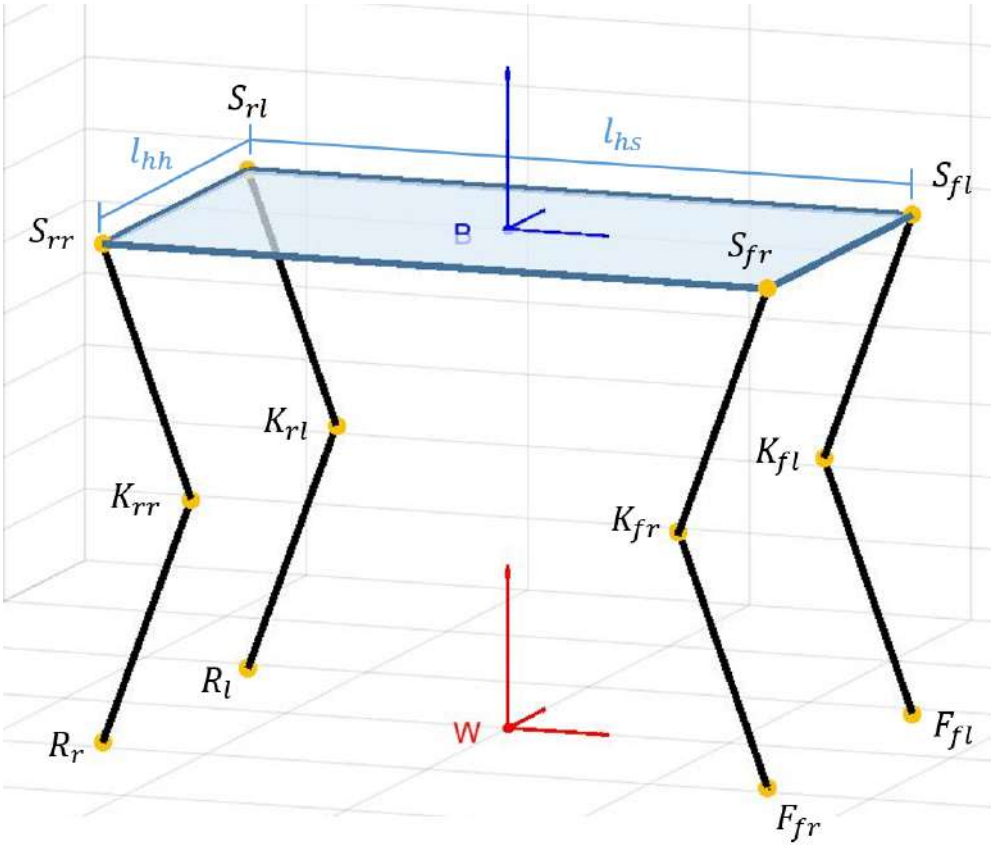


Figure 2.1: Quadruped robot sketch

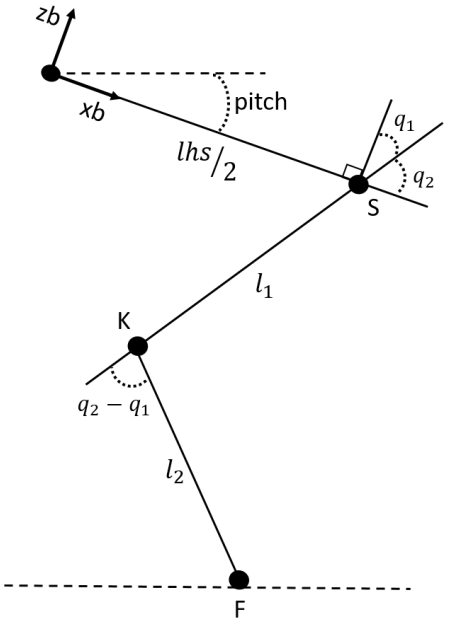


Figure 2.2: Single leg sketch

2.2 Constraints

The contact constraints have been modeled as holonomic constraints on the feet positions, i.e. each foot projection on the reference plane is always constrained to the correspondent vertex of the support polygon.

The formulation of these kind of position equalities corresponds to the closure of the kinematic chains at feet level (neglecting the frame orientation in the closure points), hence, it requires the forward kinematic model from the *World frame* to each feet, seen in Sec. 2.1, as well as the support polygon parametrization, to model the ground kinematic chains.

Observing Fig.2.3 it is clear that, on rough terrains, the feet projections along the gravity vector on the reference plane (which is perpendicular to the gravity field, as defined in Sec. 2.1) form a general quadrilateral even if the legs are planar structures. Therefore, Fig. 2.4 depicts a sketch of the support polygon in the most general case.

Given the position of the feet it is straightforward to compute the intersection of the two diagonals, W which is the position of the *World frame*, and the distances between the feet d_1, d_2, d_3 and d_4.

Applying the *Low of Sine* it is possible to write the following relationships

$$\begin{aligned}
\frac{d_1}{\sin\alpha_1} &= \frac{p_1}{\sin\beta_2} = \frac{p_2}{\sin\beta_1} \\
\frac{d_2}{\sin\alpha_2} &= \frac{p_2}{\sin\beta_4} = \frac{p_3}{\sin\beta_3} \\
\frac{d_3}{\sin\alpha_3} &= \frac{p_3}{\sin\beta_6} = \frac{p_4}{\sin\beta_5} \\
\frac{d_4}{\sin\alpha_4} &= \frac{p_4}{\sin\beta_8} = \frac{p_1}{\sin\beta_7},
\end{aligned} \tag{2.1}$$

where, the angles between two vectors has been computed as

$$\gamma = \cos^{-1}\left(\frac{u \cdot v}{\|u\|_2\|v\|_2}\right), \tag{2.2}$$

w.r.t. Fig. 2.5.

From (2.1) we can derive the distances between the diagonal intersection and

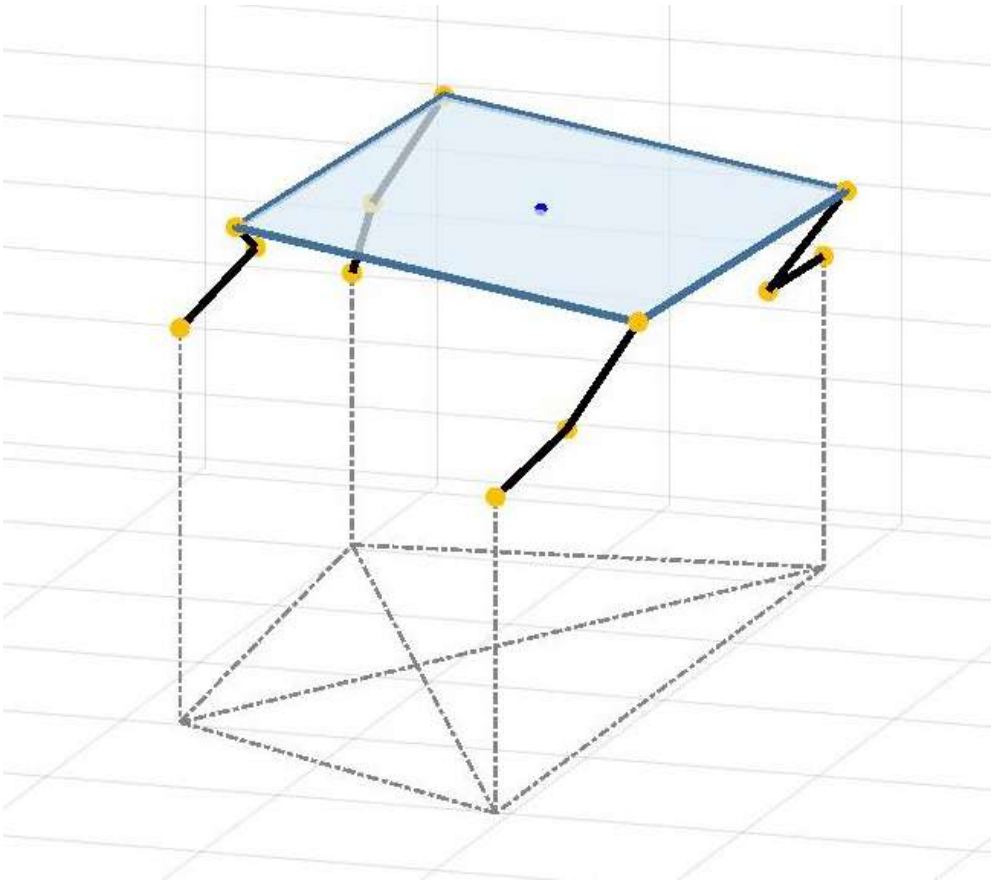


Figure 2.3: Feet projections on the reference plane forming a general quadrilateral

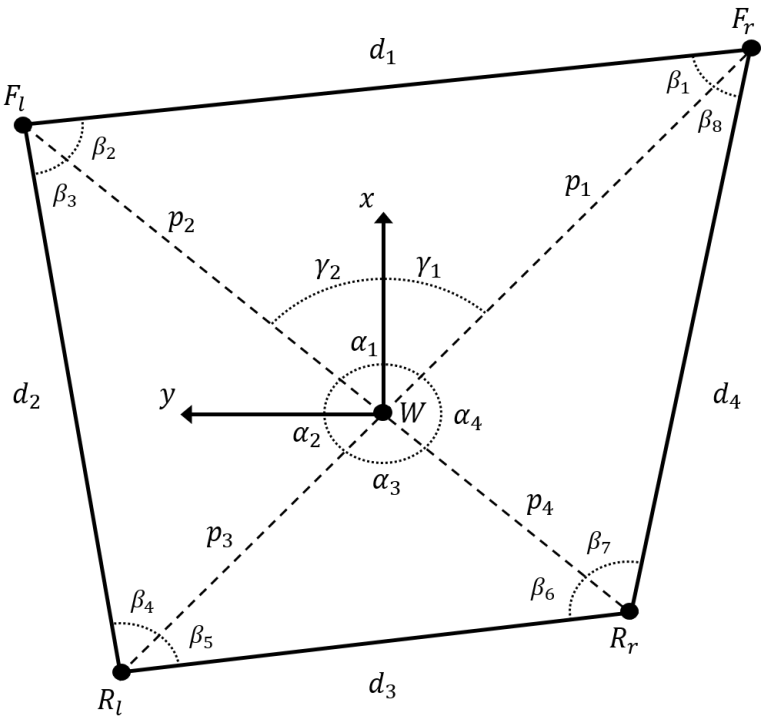


Figure 2.4: Constraints derivation sketch

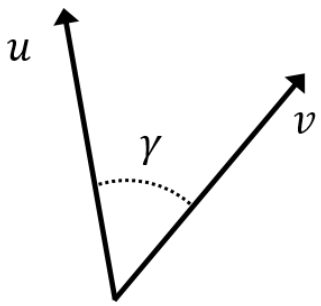


Figure 2.5: Computation of the angle between two vectors

the vertices

$$\begin{aligned} p_1 &= d_1 \frac{\sin \beta_2}{\sin \alpha_1} \\ p_2 &= d_2 \frac{\sin \beta_4}{\sin \alpha_2} \\ p_3 &= d_3 \frac{\sin \beta_6}{\sin \alpha_3} \\ p_4 &= d_4 \frac{\sin \beta_8}{\sin \alpha_4}. \end{aligned} \tag{2.3}$$

With (2.3), and the angles γ_1 and γ_2, we can write four $H \in SE(3)$ matrices from which it is trivial to extract the closed form of the vectors from W to each foot

$$\begin{aligned} H_{F_r} &= R_z(-\gamma_1)T_x(p_1) \\ H_{F_l} &= R_z(-\gamma_2)T_x(p_2) \\ H_{R_l} &= R_z(\pi - \gamma_1)T_x(p_3) \\ H_{R_r} &= R_z(\pi + \gamma_2)T_x(p_4), \end{aligned} \tag{2.4}$$

where $R_z \in SE(3)$ and $T_x \in SE(3)$ have the classical interpretation of rotation around z and translation along x.

Finally, using the *forward kinematic* results to extract $\vec{r}^{W}_{F_r}$, $\vec{r}^{W}_{F_l}$, $\vec{r}^{W}_{R_l}$, $\vec{r}^{W}_{R_r}$ and the vectors $\vec{p_1}$, $\vec{p_2}$, $\vec{p_3}$ and $\vec{p_4}$, extracted from (2.4), the constraints formulation is

$$\phi(x, d, q) = \begin{bmatrix} C1_x \\ C2_x \\ C3_x \\ C4_x \\ C1_z \\ C2_z \\ C3_z \\ C4_z \end{bmatrix}, \tag{2.5}$$

where

$$\begin{aligned} C1 &= \vec{r}^{W}_{F_r}(x, q) - \vec{p_1}(d) \\ C2 &= \vec{r}^{W}_{F_l}(x, q) - \vec{p_2}(d) \\ C3 &= \vec{r}^{W}_{R_l}(x, q) - \vec{p_3}(d) \\ C4 &= \vec{r}^{W}_{R_r}(x, q) - \vec{p_4}(d) \end{aligned} \tag{2.6}$$

and the d vector is $[d_1 \quad d_2 \quad d_3 \quad d_4]^T$.

2.3 Inverse Kinematics

The computation of the joint angles given the position of the feet and the coordinate of the trunk x means not only finding the solution of the *inverse kinematic* problem but also creating a one-to-one map between the *task coordinates*, which will be introduced in Sec. 3.1, and the joint angles. This map will be used extensively for solving the *virtual equilibrium* as function of the task coordinates in Chap. 3.

Considering the presented robotic system, the inverse kinematics can be solved "leg-wise" and, taking into account that the legs are planar, it is possible to find a closed form solution.

Given the position of one shoulder expressed in *World frame* v_s, from Sec. 2.1, and the position of the relative foot in *World frame* v_f, from Sec. 2.2, the vector from the foot to the shoulder is $v_{fs} = v_s - v_f = [x_{fs} \quad 0 \quad z_{fs}]$.

The knee angle $q_2 - q_1$ has to respect the *Carnot theorem* (Fig. 2.6),

$$v_{fs}^2 = l_1^2 + l_2^2 - 2l_1 l_2 \cos(\pi - [q_2 - q_1]). \tag{2.7}$$

With a simple trigonometric equivalence, (2.7) can be rewritten as

$$v_{fs}^2 = l_1^2 + l_2^2 + 2l_1 l_2 \cos(q_2 - q_1), \tag{2.8}$$

hence,

$$q_2 - q_1 = \cos^{-1}\left(\frac{v_{fs}^2 - l_1^2 - l_2^2}{2l_1 l_2}\right). \tag{2.9}$$

The ankle angle is

$$q_f = \alpha + \beta - \frac{\pi}{2}, \tag{2.10}$$

where α can be computed with the *Carnot theorem* too as

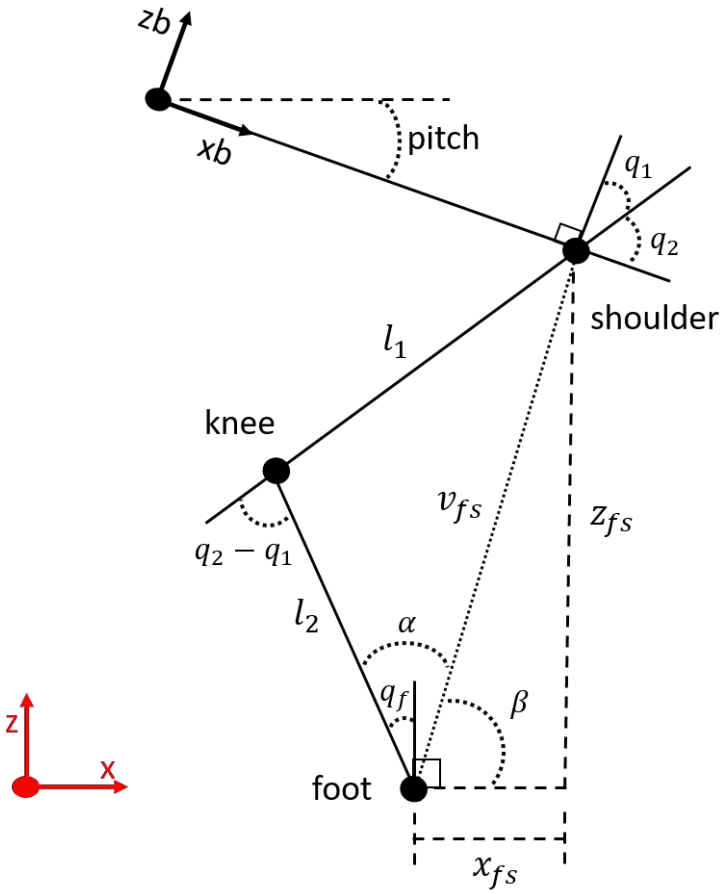


Figure 2.6: Inverse kinematic sketch

	Front leg	**Rear leg**
Knee inside	$\alpha > 0$ $(q_2 - q_1) > 0$	$\alpha < 0$ $(q_2 - q_1) < 0$
Knee outside	$\alpha < 0$ $(q_2 - q_1) < 0$	$\alpha > 0$ $(q_2 - q_1) > 0$

Table 2.3: Inverse kinematics multiple cases depending on knee inside/outside

$$\alpha = \cos^{-1}\left(\frac{l_1^2 - l_2^2 - v_{fs}^2}{-2l_2 v_{fs}}\right) = \cos^{-1}\left(\frac{v_{fs}^2 + l_2^2 - l_1^2}{2l_2 v_{fs}}\right) \tag{2.11}$$

and β is the angle between x_{fs} and z_{fs}

$$\beta = atan2\left(\frac{z_{fs}}{x_{fs}}\right) \tag{2.12}$$

Therefore, from Fig. 2.6, one can notice that

$$\vartheta = -q_2 - q_f \curvearrowright q_2 = -\vartheta - q_f. \tag{2.13}$$

Finally, substituting (2.10), (2.11) and (2.12) into (2.13) and (2.9), we can obtain q_1 and q_2

$$q_1 = q_2 \pm \cos^{-1}\left(\frac{v_{fs}^2 - l_1^2 - l_2^2}{2l_1 l_2}\right) \tag{2.14}$$

$$q_2 = -\vartheta - \left[atan2\left(\frac{z_{fs}}{x_{fs}}\right) - \frac{\pi}{2} \pm \cos^{-1}\left(\frac{v_{fs}^2 + l_2^2 - l_1^2}{2l_2 v_{fs}}\right)\right] \tag{2.15}$$

The multiple cases can be solved fixing the configuration of the leg as a *knee inside* or *knee outside* structure, where the adjectives *inside* and *outside* are referred to the space between the legs. Tab. 2.3 shows the matching.

Chapter 3

Balance control via virtual equilibrium

The set of task coordinates we choose in Sec. 3.1 creates a straightforward bridge connecting the planning phase with the robot control one. Later in this chapter, we are going to derive two frameworks which implement the main idea of this thesis and close the gap with the low level joints position control. The first one is supposed to be used with three feet on the ground and should be able to keep the robot balance while performing the step; the latter considers four feet in contact with the ground and should give us the possibility to unload and load one leg in order to consequently switch to the first one.

The conjunct use of these two frameworks covers all the possible contact situations of a quadruped robotic system during the static walking task, which have been extensively presented in Chap. 4.

Both this frameworks should compute the motors angles θ which realize the equilibrium point of the system relative to a desired set of task coordinates z.

3.1 Task coordinates

The system presented in Chap. 2 can be in two different contact situations: three legs in contact with the ground, during the leg swing phase, and four legs in contact.

Feet on the ground	d **vector**
Fl-Rl-Rr	$d = [d_2 \quad d_3]^T$
Fr-Rl-Rr	$d = [d_3 \quad d_4]^T$
Fr-Fl-Rr	$d = [d_1 \quad d_4]^T$
Fr-Fl-Rl	$d = [d_1 \quad d_2]^T$

Table 3.1: Additional task coordinates for the three legs in contact situation

For each of these we want to define a set of *task coordinates*, to be used in Sec. 3.2 and 3.3, in which the description of the static locomotion task is particularly intuitive.
During both the contact situations it is clearly meaningful to have control on the DOF of the trunk $x = [x_b \quad z_b \quad \varphi \quad \vartheta]^T$.

When the system has three legs in contact with the ground there are six holonomic constraints acting on the three legs so it is necessary to select others two task coordinates to create a one-to-one map; the most natural choice is to use two of the three distances between the feet. Additionally, this choices allow the control system to reduce the internal tension between the legs. Considering the notation introduced in Sec. 2.2, Tab.3.1 summarize this selection for each triplet of legs in contact.

Consequently, the set of selected task coordinates for the three legs in contact situation is $z_h = [x \quad d]^T$.

During the four contact points phases all the feet are constrained at the height that they have at the end of the previous step. In continuity with the choice of z_h, and considering that the parametrization of the size of a generic quadrilateral requires at least three sides, the second set of task coordinates can be chosen as $z_a = [x_b \quad z_b \quad \varphi \quad \vartheta \quad d_2 \quad d_3 \quad d_4]^T$. As shown in Sec. 3.3 an additional task coordinate a will be added with the aim of controlling the force distribution of one leg w.r.t. the total load, in order to achieve the unload and load of the leg before the step.

3.2 Three-Contact-Points framework

Consider the system modelled in Chap. 2,the Euler-Lagrange equations of motion are

$$\frac{d}{dt}\left(\frac{\partial L(y,\dot{y})}{\partial \dot{y}}\right)^T - \left(\frac{\partial L(y,\dot{y})}{\partial y}\right)^T = \tau^{ext}, \tag{3.1}$$

where $y = (z_h, q, \theta)$. Assuming statical conditions, the Lagrangian $L(y,\dot{y}) = T(y,\dot{y}) - U(y)$ becomes $L(y,\dot{y}) = -U(y)$ and, taking into account the constraints presented in Sect. 2.2, the system in the equilibrium can be described as

$$\left(\frac{\partial U(x,q,\theta)}{\partial q}\right)^T - \left(\frac{\partial \phi(z_h,q)}{\partial q}\right)^T \lambda = 0, \tag{3.2}$$

$$\left(\frac{\partial U(x,q,\theta)}{\partial z_h}\right)^T - \left(\frac{\partial \phi(z_h,q)}{\partial z_h}\right)^T \lambda = 0, \tag{3.3}$$

where $\lambda \in \mathbb{R}^{dim(\phi)}$ are the Lagrange multipliers. Solving (3.2) for λ and substituting in (3.3)

$$\left(\frac{\partial U(x,q,\theta)}{\partial z_h}\right)^T - \left(\frac{\partial \phi(z_h,q)}{\partial z_h}\right)^T \left(\frac{\partial \phi(z_h,q)}{\partial q}\right)^{-T} \left(\frac{\partial U(x,q,\theta)}{\partial q}\right)^T = 0 \tag{3.4}$$

which, for the sake of compactness, can be rewritten as

$$\left(\frac{\partial U(x,q,\theta)}{\partial z_h}\right)^T + \Phi_{qz}^T \left(\frac{\partial U(x,q,\theta)}{\partial q}\right)^T = 0 \tag{3.5}$$

where Φ_{qz} has been defined as the Jacobian matrix mapping the task velocities $\dot{z_h}$ to the joints velocities $\dot{q}$

$$\Phi_{qz} = -\left(\frac{\partial \phi(z_h,q)}{\partial q}\right)^{-1} \left(\frac{\partial \phi(z_h,q)}{\partial z_h}\right). \tag{3.6}$$

Remark 1 *Note that the matrix $\frac{\partial \phi(z_h,q)}{\partial q}$ has dimensions $dim(\phi) \times dim(q)$, hence for computing λ from (3.2) and substituting it in (3.3), and also for computing Φ_{qz}, it is necessary that the constraints are the same number that the joint angles. Taking into account that this framework works with three legs constrained to the ground, $dim(\phi) = 6$.*

Explicitating equation (3.5) in matrix form and the joints torques produced by the potential energy of the system in gravitational and elastic parts $\frac{\partial U(x,q,\theta)}{\partial q} = \frac{\partial U_g(z_h,q)}{\partial q} + \frac{\partial U_e(q,\theta)}{\partial q}$

$$\left(\frac{\partial U_e(q,\theta)}{\partial q}\right)^T = -\Phi_{qz}^{-T}\begin{bmatrix}\left(\frac{\partial U(x,q,\theta)}{\partial x}\right)^T\\ 0\end{bmatrix} + \left(\frac{\partial U_g(x,q)}{\partial q}\right)^T, \tag{3.7}$$

Assumption 1 *The elastic potential is such that*

$$U_e(q,\theta) = (\theta - q)^T K(\theta - q), \tag{3.8}$$

where K is the stiffness matrix. This is verified in case of series elastic actuator (SEA):

In the general case (3.7) could not be solved, but, considering Assumption 1, (3.7) can be solved for θ as

$$\theta = K^{-1}b(z_h,q) + q, \tag{3.9}$$

where the notation has been compacted defining the right hand side of (3.7) as

$$b(z_h,q) = -\Phi_{qz}^{-T}\begin{bmatrix}\left(\frac{\partial U(x,q,\theta)}{\partial x}\right)^T\\ 0\end{bmatrix} + \left(\frac{\partial U_g(x,q)}{\partial q}\right)^T. \tag{3.10}$$

Remark 2 *The jacobian Φ_{qz} has dimensions $dim(q)\times dim(z)$, which means that, considering also Remark 1, when the system has three legs in contact with the ground, the number of chosen task coordinates has to be six. This imply also that (3.9) produces the six motor angles relative to the three legs in contact; however, considering that there are not contact forces on the lifted leg, it is always possible to consider $\theta_{lifted} \approx q_{lifted}$. This approximation neglects the dynamical effects (low if low swing speed) and the gravitational force induced by the weight of the leg itself, which is also very low compared with the trunk weight as seen in Sec.1.3. Furthermore, the exact placement of the feet is not required for the locomotion control presented in Chap.4.*

Additionally, the right had side of (3.9) can be expressed as a function of only z_h using the map between task coordinates and joints positions (derived in Sect. 2.3) obtaining an expression of the motors positions which realize the equilibrium position of the system w.r.t. to the desired set of task coordinates and respect the contact constraints.

Remark 3 *Under Assumption 4.1 the quantity $b(z_h)$ is the joints torques vector as function of the task coordinates.*

Proof: The proof of Remark 3 comes straightforwardly from (3.9) noticing that this latter can be written as

$$K(\theta - q) = b(z_h). \tag{3.11}$$

3.3 Four-Contact-Points framework

Considering the task of realizing a desired positioning relatives to the support quadrilateral with four feet in contact, and the subtask of controlling the force distribution of the next foot to be lifted w.r.t. the other feet, the above calculation can be modified as follow.

From (3.2) the expression of the contact forces, through the Lagrange multipliers, is

$$\lambda = \left(\frac{\partial \phi(z,q)}{\partial q}\right)^{-T} \left(\frac{\partial U(x,q,\theta)}{\partial q}\right)^{T}, \tag{3.12}$$

which, taking into consideration, the constraints formulation of Sect. 2.2, yields the contact forces vector

$$\lambda = [\lambda_{fr}^{tang} \quad \lambda_{fl}^{tang} \quad \lambda_{rl}^{tang} \quad \lambda_{rr}^{tang} \quad \lambda_{fr}^{norm} \quad \lambda_{fl}^{norm} \lambda_{rl}^{norm} \quad \lambda_{rr}^{norm}]^T, \tag{3.13}$$

where the expressions *tangential* and *normal* are referred to the plane that entirely contains the support polygon.

Is it now possible to define a dimensionless coordinates which encodes the

normal force distribution of the i-th leg w.r.t. the others as

$$a_i = \frac{3\lambda_i^{norm} - \sum_{\substack{j=1 \\ j\neq i}}^{j=4} \lambda_j^{norm}}{\sum_{j=1}^{j=4} \lambda_j^{norm}}. \tag{3.14}$$

Definition (3.14) ensures that if $a_i = 0$ the normal contact force on the i-th leg is equal to one third of the sum of normal forces acting on the other three legs. While, if $a_i = -1$ it ensures that the normal contact force on the i-th leg is zero and the total force is beared by the other legs.

Combining the *Virtual Equilibrium* (VE) formulation of (3.5) with the definition (3.14), we obtain

$$\begin{cases} \left(\frac{\partial U(x,q,\theta)}{\partial z_a}\right)^T + \Phi_{qz}^T \left(\frac{\partial U(x,q,\theta)}{\partial q}\right)^T = 0 \\ a_i \sum_{j=1}^{j=4} \lambda_j^{norm} - \left(3\lambda_i^{norm} - \sum_{\substack{j=1 \\ j\neq i}}^{j=4} \lambda_j^{norm}\right) \end{cases} . \tag{3.15}$$

This a set of $dim(z_a)+1 = 8$ equations linear in θ, hence they can be rewritten as $A_i\theta = b_i$ and, if all the constraints are fulfilled, solved for θ.

3.4 Moving the Equilibrium point

Given two different sets of task coordinates z_0 and z_1, in order to move the VE of the system from the one relative to z_0 to the one relative to z_1, keeping the oscillatory behaviour as small as possible, it is required a method to provide smooth and continuous transitions from two different sets of task coordinates. Defining the transition function $P(t)$, t_0 and t_1 respectively the initial and final transition time, this can be expressed through the following conditions

$$\begin{aligned} P(t_0) &= z_0 \\ \dot{P}(t_0) &= 0 \\ \ddot{P}(t_0) &= 0 \\ P(t_1) &= z_1 \\ \dot{P}(t_1) &= 0 \\ \ddot{P}(t_1) &= 0. \end{aligned} \tag{3.16}$$

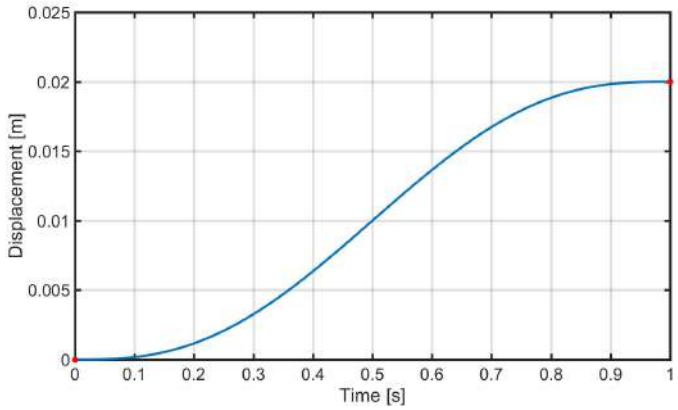


Figure 3.1: x_b trajectory example with *2 cm* displacement at *2 cm/s*

Selecting $P(t)$ from the polynomials space of degree d, rewriting the conditions (3.1) in matrix form

$$\begin{bmatrix} t_0^d & t_0^{d-1} & \dots & t_0^2 & t_0 & 1 \\ dt_0^{d-1} & (d-1)t_0^{d-2} & \dots & t_0 & 1 & 0 \\ d(d-1)t_0^{d-2} & \dots & \dots & 1 & 0 & 0 \\ t_1^d & t_1^{d-1} & \dots & t_1^2 & t_1 & 1 \\ dt_1^{d-1} & (d-1)t_1^{d-2} & \dots & t_1 & 1 & 0 \\ d(d-1)t_1^{d-2} & \dots & \dots & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_d \\ p_{d-1} \\ \vdots \\ p_2 \\ p_1 \\ p_0 \end{bmatrix} = \begin{bmatrix} z_0 \\ 0 \\ 0 \\ z_1 \\ 0 \\ 0 \end{bmatrix}, \tag{3.17}$$

and, more compactly,

$$H(t_0, t_1)p = c(z_0, z_1), \tag{3.18}$$

where $H \in \mathbb{R}^{(d+1)\times(d+1)}$.

This corresponds to the problem of finding a polynomial function which matches n known values and its m derivatives (note that m can but has not to be equal for all the values and one should consider $m = \max_r P^{(r)}(t)$). This problem is known in literature as *Hermite Interpolation* and the solution

$$p = H^{-1}(t_0, t_1)c(z_0, z_1) \tag{3.19}$$

ensure the existence of the polynomial function $P(t)$ with degree $d \leq n(m+1)-1$, where each coefficient is a function of t_0, z_0, t_1 and z_1.

With the defined conditions (3.1) $n = 2$ and $m = 2$, therefore $d = 5$.

Chapter 4

Locomotion control

Locomotion control for *underactuated* robots is a challenging problem, it involves the planning of a sequence of joint angles which moves the robot to a desired position while keeping the balance and rejecting external disturbances.

Tackling such an elaborate task usually involves an hierarchical approach, e.g. [3] and [2], where the highest level is reserved to the planning phase, in a similar fashion in this chapter we are going to derive a step planner for the static locomotion task and afterwards, we present an implementation which provides smooth task coordinates trajectories to the balance control depicted in Chap. 3.

4.1 Static gait

Biosciences have studied for a long time locomotion of legged mammals highlighting how a low speed static gait is incredibly effective on rough terrains.

Definition 4.1 (McGhee [1])*:* The *support polygon* (SP) of a legged robot is the minimum convex area which contains all the legs contact points or their projections on a given plane perpendicular to the gravity field.

Definition 4.2 (McGhee [1])*:* A legged robot is *statically stable at time* t if and only if the vertical projection (w.r.t. the gravity field) of its center of gravity (CoG) lies within the support polygon at the given time t.

Definition 4.3 (McGhee [1])*:* The magnitude of the *stability margin at time*

Figure 4.1: Goats on almost vertical wall, amazing balancing skills

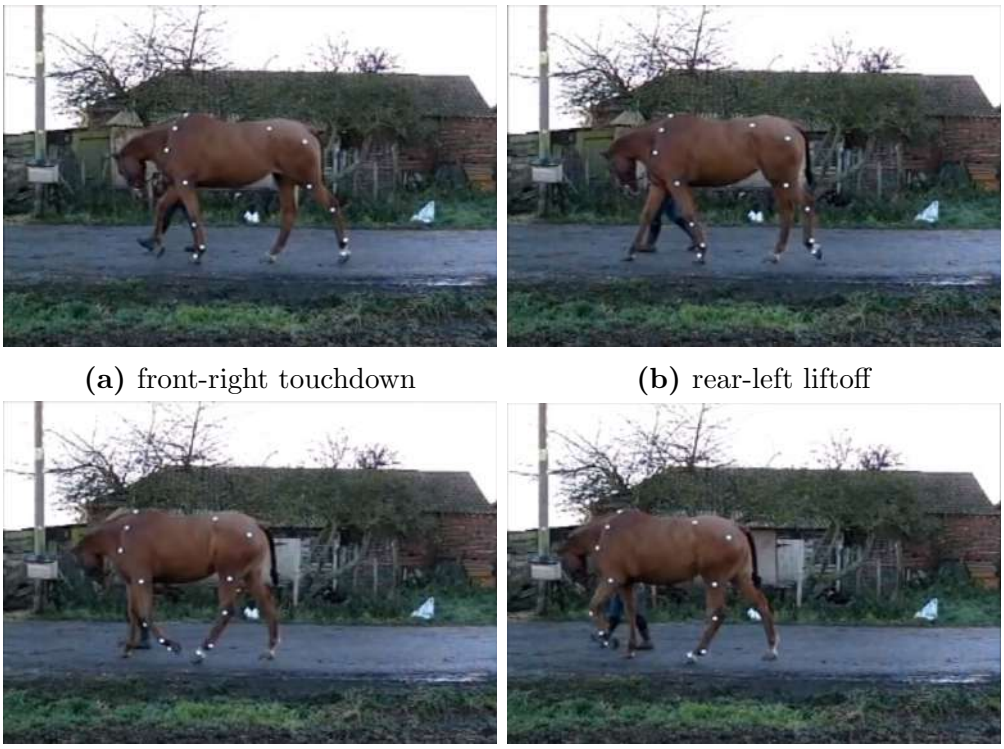
Source: *Medavia.co.uk*

t (SM) is equal to the shortest distance from the vertical projection of the CoG to any point of the boundary of the SP. If the system is statically stable at time t the SM is positive, otherwise it is negative.

4.2 Step Planner

The stability properties of the gait are tightly linked to the relative position between the CoG projection on the reference plane and the feet positions (i.e. the vertices of the SP) which, in turn, are the results of the selected step sequence. Considering a quadruped robot ($n = 4$), there are exactly 6 possible feet sequence choices. Despite the number of all the possible permutations of the four feet is $n! = 24$, one should consider that the periodicity of the task imply the use of circular permutations instead of classic ones, therefore all the possible feet sequences are $(n-1)! = 6$.
In [1] the optimal static gait has been derived, where, in this context, the optimality is evaluated w.r.t. the biggest SM, but the presented proof requires each rear foot to be placed in the exact footprint of the foot ahead of it at approximately the same moment of the front one liftoff.

The depicted situation is showed in Fig. 4.2 with the necessary condition, which is evident in Fig. 4.2c.



(a) front-right touchdown

(b) rear-left liftoff

(c) rear-left touchdown and front-left liftoff

(d) front-left touchdown

Figure 4.2: Optimal static gait horse example

Source: https://www.youtube.com/watch?v=ziBPaMMUzgM

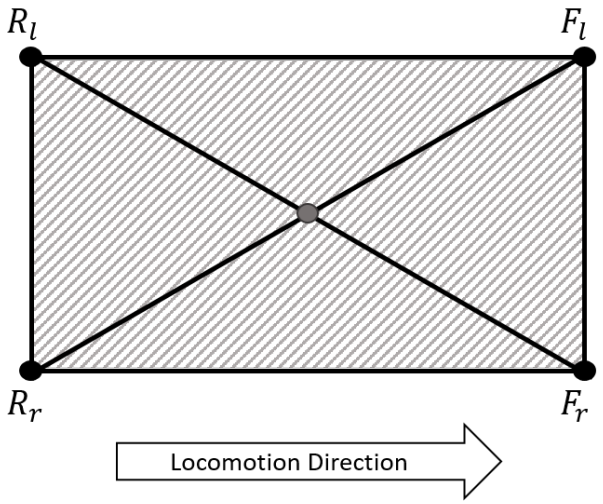
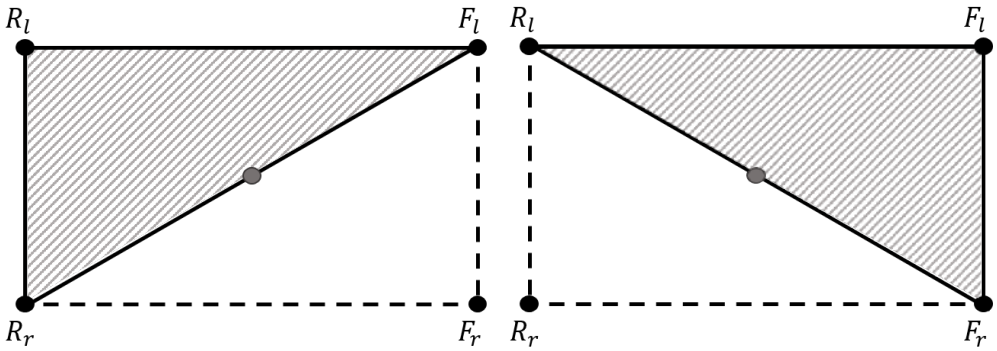


Figure 4.3: Initial SP and considered locomotion direction

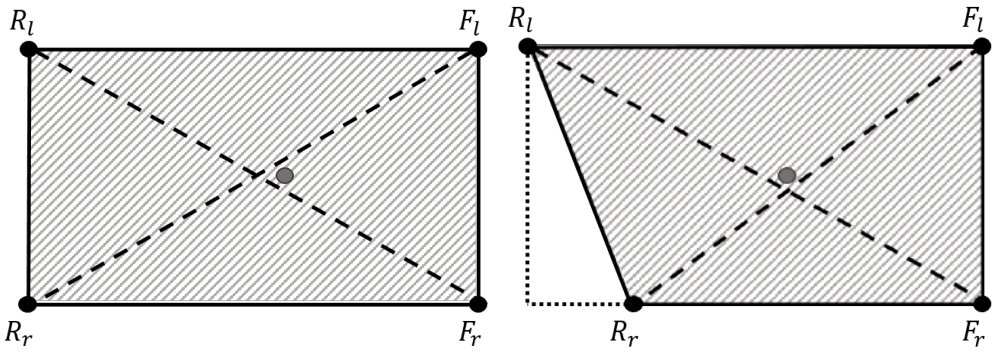


(a) Front foot first step **(b)** Rear foot first step

Figure 4.4: First step possible choices

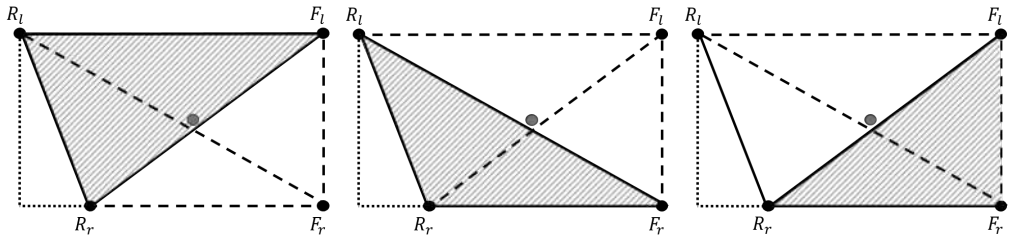
This requirement limits the use of the provided results to those robotic systems with a leg to trunk length ratio large enough to reach the projection of the trunk middlepoint on the ground with all the feet; the value of this ratio for the system presented in Sec. 1.3 is close to 1/2 ($\frac{l_1+l_2}{l_{hs}} = \frac{0.16}{0.3} = 0.5\overline{3}$), which precludes the possibility to employ the optimal sequence from [1] without being dangerously close to leg's singularity.

Consider the aim of finding a step sequence for the quadruped presented in Sec. 1.3 which keeps the system in a statically stable condition $\forall t$ starting from a neutral initial position as in Fig. 4.3 and consider also, without any loss of generality, a forward desired motion of the CoG. The first choice to be taken is the selection of the first foot to lift. Fig. 4.4 shows that this first choice is straightforward because, with a forward motion of the CoG, the liftoff of one of the front legs (Fig. 4.4a) leads to an unstable configuration.



(a) Before **(b)** After

Figure 4.5: First step



(a) Fr liftoff **(b)** Fl liftoff **(c)** Rl liftoff

Figure 4.6: Ipotetic scenrios for second step

According with this *maximal stability policy*, the first step (Fig. 4.5) is performed by the *rear-right* foot and it is preceded by a CoG displacement to achieve a desired SM.

Considering the feet configuration depicted by Fig.4.5b, the foot for the second step can be selected among the set Fr, Fl, Rl. Fig. 4.6 shows the stability scenario for each of these choices highlighting how the most stable one is Fr. It is now fundamental to note that the SM of each configuration is a function of all the previous *Step Lengths* (SL), but the comparison between different step sequences is SL-independent because it is always possible to consider the same SL for every sequence.

Hence, performing the second step with the *front-right* foot and considering, as a first level of approximation, an equal SL for all the steps, the obtained configuration is showed in Fig. 4.7.

The set of feet not already lifted among which is possible to pick the foot for the next step is now only composed by Fl and Rl. As depicted in Fig. 4.8, both this choices have $SM = 0$ and this leads to the inevitable need of moving

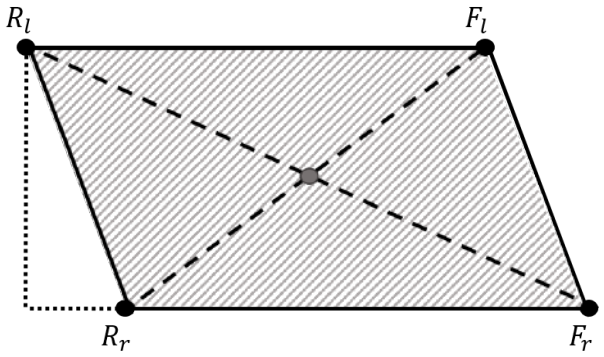
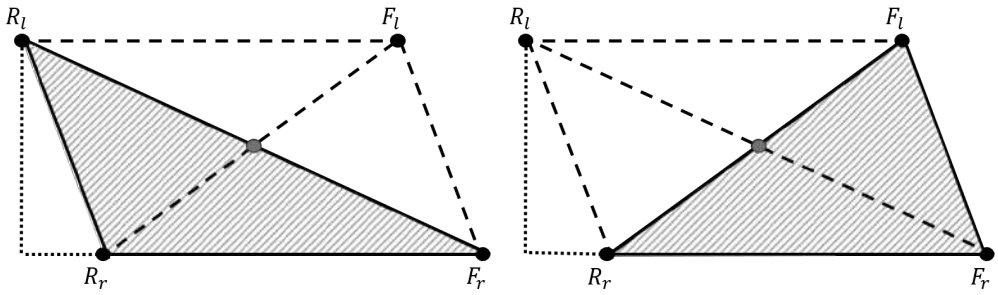


Figure 4.7: Configuration after second step



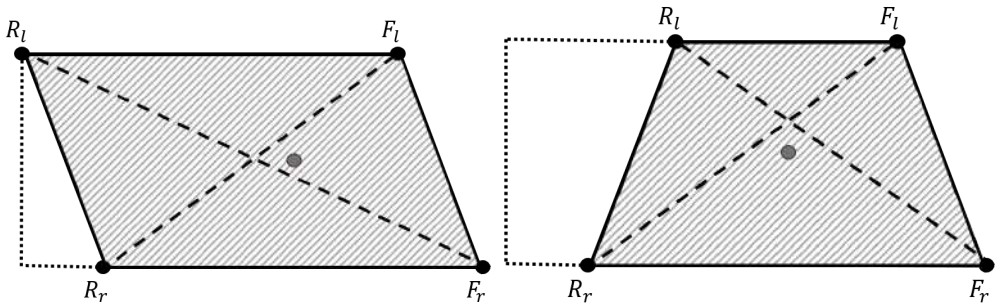
(a) Selection of Fl **(b)** Selection of Rl

Figure 4.8: Third step possible scenarios

the CoG; comparing Fig. 4.8a and 4.8b it is clear how performing the third step with the *rear-left* foot allows to move the CoG in the direction of motion, while selecting the *front-left* foot impose to move the CoG in the opposite direction w.r.t. the locomotion direction.

The feet configurations, respectively before and after the third step, are illustrated by Fig. 4.9.

The only possible option for the last step is the *front-left* foot, Fig.4.10 de-



(a) CoG movement before the step **(b)** Feet configuration after the step

Figure 4.9: Third step

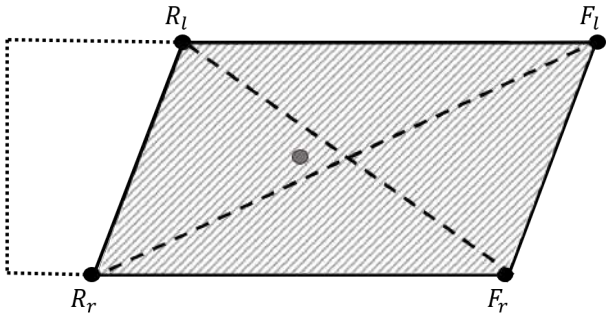


Figure 4.10: Final configuration after fourth step

pict the final configuration after the fourth step. It is now evident that before repeating the sequence it is required to move the CoM forward to achieve again the static stability for the Rr step.

Summarizing, the selected step sequence is: $R_r - F_r - R_l - F_l$.

4.3 Workspace

The presented locomotion approach consist of a succession of steps and CoG movements, each of these action is constrained by two kinds of feasibility conditions which they have to respected.

Feasibility condition 1: the three points *Shoulder-Knee-Foot* of each leg should always form a triangle where the vertices are not collinear.

Considering only one leg and observing Fig. 4.11 it is clear that this condition derives directly from the *Triangle Inequality*

$$v_{fs} = \sqrt{{x_{fs}}^2 + {z_{fs}}^2} \leq l_1 + l_2 \rightarrow x_{fs} \leq \sqrt{(l_1 + l_2)^2 - {z_{fs}}^2}. \tag{4.1}$$

Taking into account that the *Body frame* is rigidly linked to all the shoulders, the previous inequality leads to a feasible workspace for the x_b task coordinate of

$$x_b \in \left[0, \sqrt{(l_1 + l_2)^2 - \max_{i \in \Lambda} {z_{fs_i}}^2}\right), \tag{4.2}$$

where Λ is the set which enumerates all the legs, $\Lambda = [Fr_{leg}, Fl_{leg}, Rl_{leg}, Rr_{leg}]$.

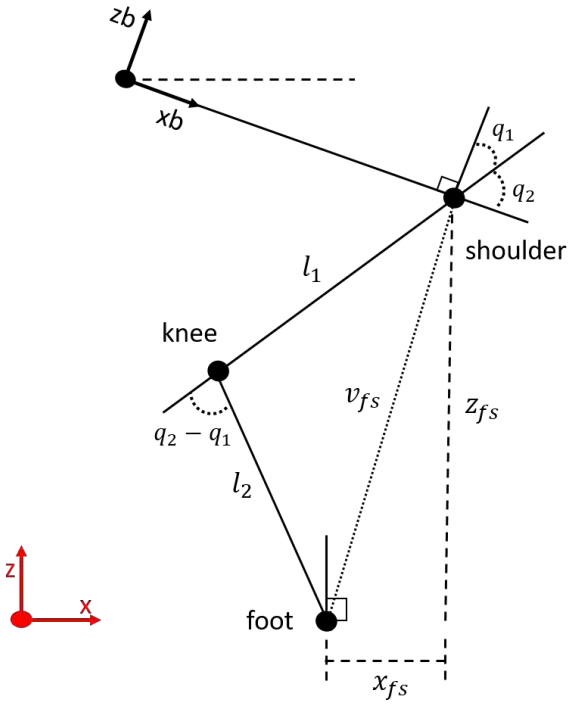


Figure 4.11: IK feasibility constraint

	Hip/Shoulder (q_1)	**Knee** (q_3)
Front leg	$[-120°, 60°]$	$[-120°, 120°]$
Rear leg	$[-60°, 120°]$	$[-120°, 120°]$

Table 4.1: Joints limits [min, max]

Note that this condition also ensures that all the solutions of the *inverse kinematic* algorithm, presented in Sec. 2.3, always belongs to $\mathbb{R}$.

The presented workspace is further constrained by the joints limits reported in Tab. 4.1; the same limits can be rearranged in an *inner-outer* disposition, Tab. 4.2, to highlight when the contact happens with an inner part of the structure or with an outer one.

Inspecting Tab. 4.2, it appears evident that the tightest limitation on the x_b workspace is given by the inner contact limit between the thigh and the shoulder's case. Fig. 4.12 shows an example of limit configuration for the *front-right* leg with feet perpendicularly under the shoulder and knee angle at initial value of 30°.

	Inner	Outer
Front Hip/Shoulder (q_1)	60°	−120°
Front Knee (q_3)	120°	−120°
Rear Hip/Shoulder (q_1)	−60°	120°
Rear Knee (q_3)	−120°	120°

Table 4.2: Joints limits, inner outer disposition

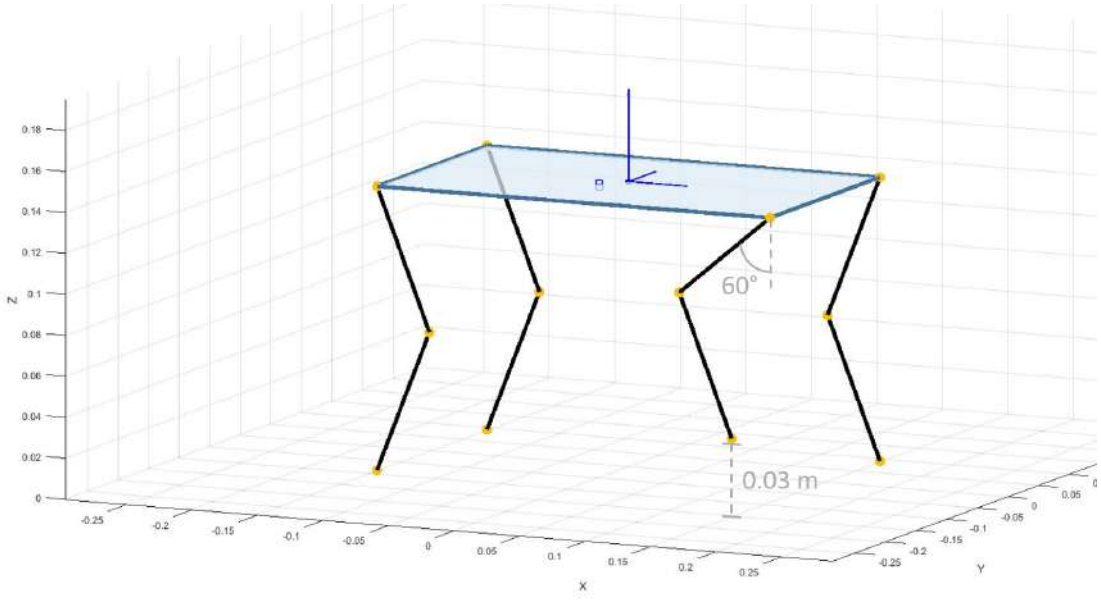


Figure 4.12: Joint limit configuration example for the *front-right* leg

4.4 Static walking algorithm

In accordance with the results previously presented in this chapter a static locomotion algorithm has been implemented in the Simulink enviroment through a *Finite State Machine* (*Chart* Simulink object). Fig. 4.13 shows an overview of the implemented walking algorithm that we are going to describe in details.

4.4.1 Init

The algorithm is initialized considering all the four feet in contact with the ground, a balanced contact forces distribution ($a^{des} = 0$) and the CoG at a known height w.r.t. the support plane. The reference plane where the SP lies, and from which all the heights are considered, is the plane that pass through all the feet at their initial position. All the heights are then considered parallel to the gravity field and perpendicular to the reference plane.

The initial selected leg is the last of the circular gait sequence presented in Sec. 4.2 ($[Rr - Fr - Rl - Fl]$), i.e. Fl, and the selection of one leg is coded by a set of four flags, h_{flags}, where each flag is linked to the correspondent foot of the ordered sequence and only one flag can be set at each time.

4.4.2 Stance

During the stance phase the algorithm waits for a new desired velocity command; if the desired speed of the CoG is not zero the computation goes to the selection of the next leg to lift.

This phase is executed after the initialization and also after the loading of each leg, i.e. after the conclusion of each step. The contact detection, which determines the conclusion of the steps, is, as explained in detail in Sec .4.4.6, done observing the joint torques and not with contact sensors under the feet. This introduces the needs of a robustness enhancement to keep the feet positions always as close as possible to the current ones because, as seen in Sec. 2.2, they plays a fundamental role in the constraints computation which, in turn, are essential for the *virtual equilibrium* computation.

To dismiss the dependency of the constraints computation on the contact

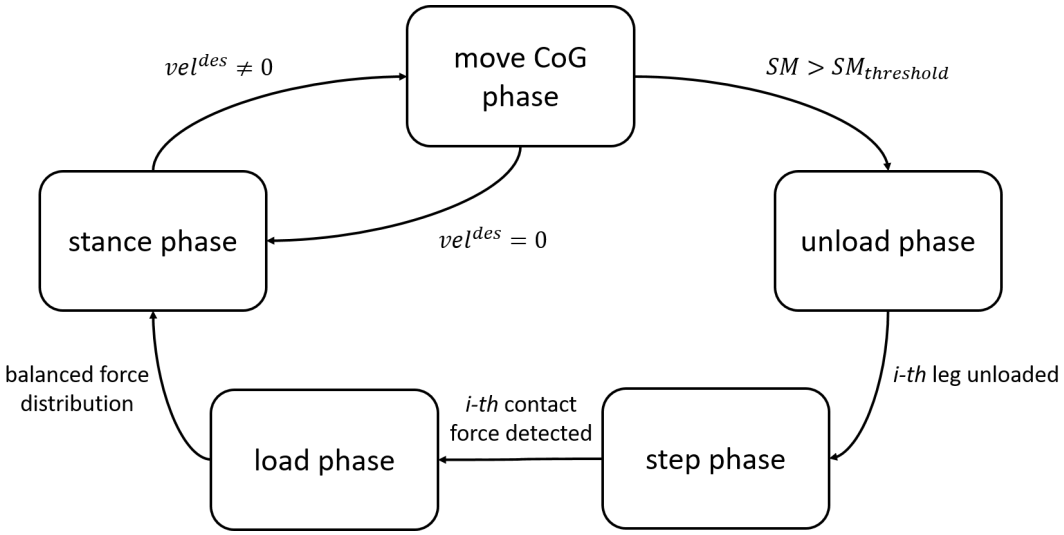


Figure 4.13: Locomotion algorithm overview

detection the direct assessment of the feet positions has been added to this phase. In Fig. 4.14 it is shown how the computation of the feet positions in *world frame* (W) can be done evaluating firstly the position of the *Body frame* (B) , r_W^B, and then the vectors from B to each foot using the kinematics, presented in Sec. 2.1, and the measured values of the euler and joints angles.

4.4.3 Next leg

In addition to the simple selection of the next leg to lift, this algorithm phase take care of providing discontinuous values of a^{des} such that the θ^{des} produced by the balance controller are perfectly smooth. Having smooth motor angles makes much more easy for the system to stay close the desired *virtual equilibrium* and to reduces the oscillations.

The *four contact points framework*, presented in Sec. 3.3, controls the force distribution on the selected leg w.r.t. the sum of the forces distribution on the others three legs through (3.14). Hence, when $a_{fr}^{des} = 0$ the force on the *forward right* leg, f_{Fr}^{gc}, is

$$f_{Fr}^{gc}|_{a_{fr}^{des}=0} = \frac{1}{3}(f_{Fl}^{gc} + f_{Rl}^{gc} + f_{Rr}^{gc}), \tag{4.3}$$

but, in the general case (i.e. when the projection of the CoG is not on the intersection on the SP diagonals), $f_{Fr}^{gc} \neq f_{Fl}^{gc} \neq f_{Rl}^{gc} \neq f_{Rr}^{gc}$.

Therefore, changing the selected leg is equal to switch the indexes of (4.3)

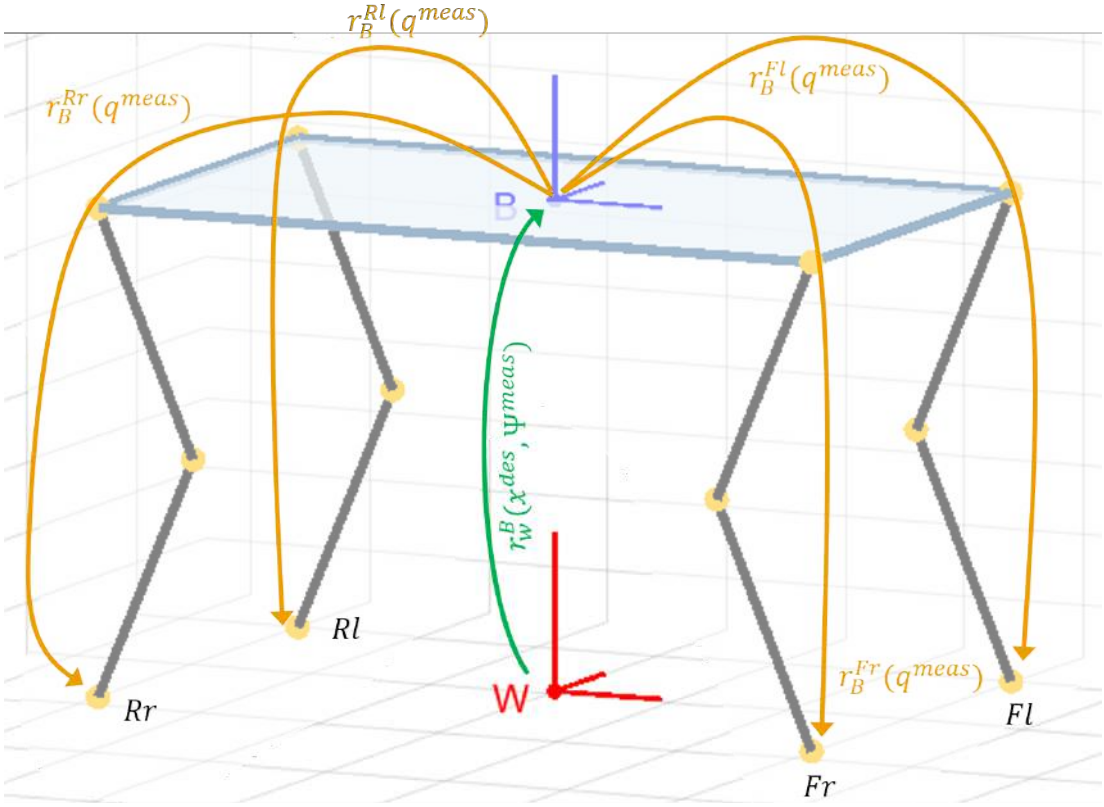


Figure 4.14: Feet positions computation sketch

which providing a discontinuous transition of the *virtual equilibrium* through a discontinuous transition in θ^{des}. Fig. 4.15 shows the natural behaviour of θ^{des} when the flags that select the controlled leg (h_{flags}) changes.

Considering the general transition from leg k, selected by the set of flags h_{flags_k}, to the leg $k+1$, selected by $h_{flags_{k+1}}$ it is possible to achieve perfectly smooth θ^{des} using the following algorithm.

Algorithm 1 Computation of discontinuous a^{des} for achieving continuous θ^{des}

procedure COMPUTE NEXT ACOORD
  $\theta \leftarrow Solve$ (3.15) $for\ \theta\ with\ a_i = a_k^{des}$
  $q \leftarrow inverse\ kinematic(z^{act})$
  $f_k^{gc} \leftarrow Solve$ (3.12) $with\ q\ and\ \theta$
  $new\ a_{k+1}^{des} \leftarrow Solve$ (3.14) $with\ f_k^{gc}$
end procedure

In Fig. 4.16 it can be noticed that Alg. 1 produces a discontinuous a^{des} to achieve continuous θ^{des} during all h_{flags} changes.

The selection of the next leg to lift is not a locomotion phase itself, hence

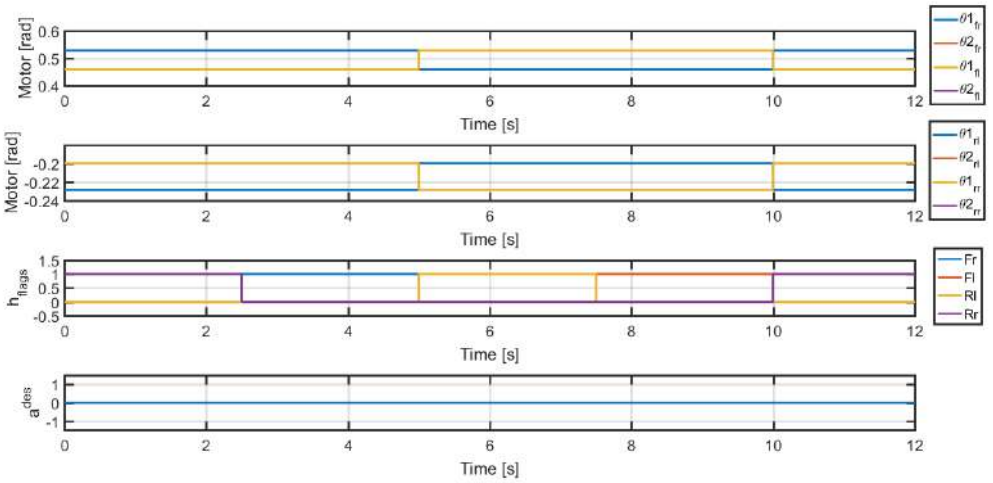


Figure 4.15: Motor desired angles during leg switch

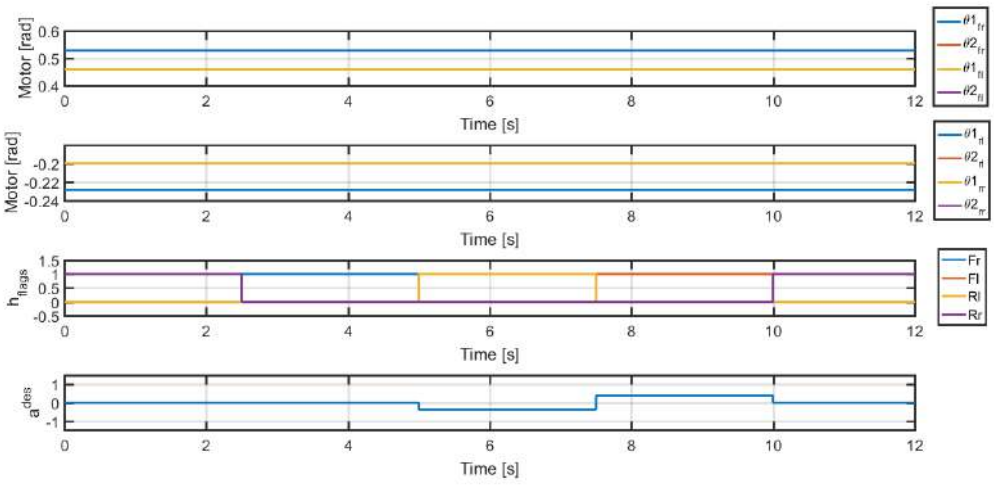
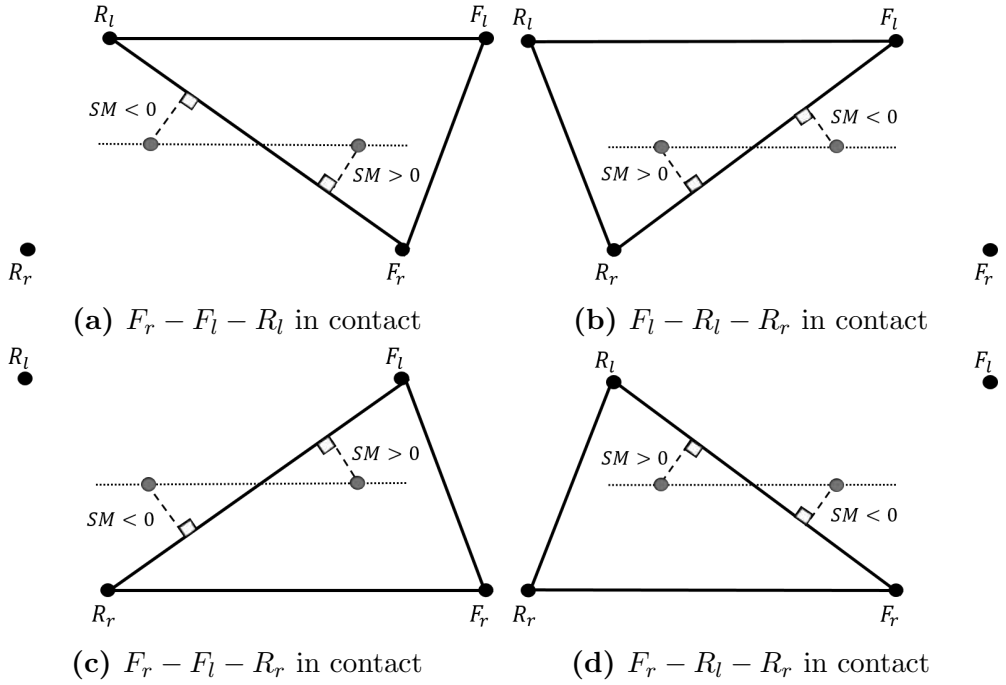


Figure 4.16: Motor desired angles during leg switch applying Alg. 1



(a) $F_r - F_l - R_l$ in contact **(b)** $F_l - R_l - R_r$ in contact

(c) $F_r - F_l - R_r$ in contact **(d)** $F_r - R_l - R_r$ in contact

Figure 4.17: SM computation cases

it might be merged with the CoG movement, at the beginning or end of it, but since its particular functionality of switching among different systems it has been kept separated.

4.4.4 Move CoG

Before the liftoff of each foot the algorithm finds a stable configuration w.r.t. only the three feet that stay in contact, this means achieving a CoG position where $SM > SM_{threshold} > 0$.

Considering the planar legs structure, it has already been highlighted that the CoG can only be displaced along the x_b and z_b axis, therefore, the commanded displacement which allow the system to reach a stable position with three feet in contact can only be in the x_b direction.

Given the projection of the center of gravity (CoG) on the SP plane the edge of the triangle, composed by the projection of the three feet in contact, to be considered for the computation of the SM depends on which leg is to be lifted. Fig. 4.17 depicts the four possible cases, which are also summarized in Tab. 4.3

Given the point P and the line l that passes through points Q_1 and Q_2, the

Foot to be lifted	Line w.r.t. the SM is computed
Rr	*Rl-Fr*
Fr	*Rr-Fl*
Rl	*Rr-Fl*
Fl	*Rl-Fr*

Table 4.3: SM computation cases

SM is the minimum distance from P to l with negative sign if $P \notin \triangle_{Q_1Q_2Q_3}$.

$$SM = s\frac{\|(Q_2 - Q_1) \times (P - Q_1)\|_2}{\|Q_2 - Q_1\|_2} \tag{4.4}$$

where s is computed through Alg. 2.

Algorithm 2 Computation of s

```
procedure isInsideTriangle(P, Q1, Q2, Q3)
    if [SameSide(P, Q1, Q2, Q3) and
        SameSide(P, Q2, Q1, Q3) and
        SameSide(P, Q3, Q1, Q2)]        then
        s ← 1
    else
        s ← −1
    end if
end procedure
function SameSide(c1, c2, a, b)
    ▷ Return true if c1 and c2 lie on the same side of the line through a and b
    cp1 = (b − a) × (p1 − a)
    cp2 = (b − a) × (p2 − a)
    if (cp1 · cp2 ≥ 0) then
        return true
    else
        return false
    end if
end function
```

To achieve locomotion on uneven terrains the height of the *Body frame* has to be bounded between two extreme values, $z_{b_{max}}$ and $z_{b_{min}}$. If the trunk is too high

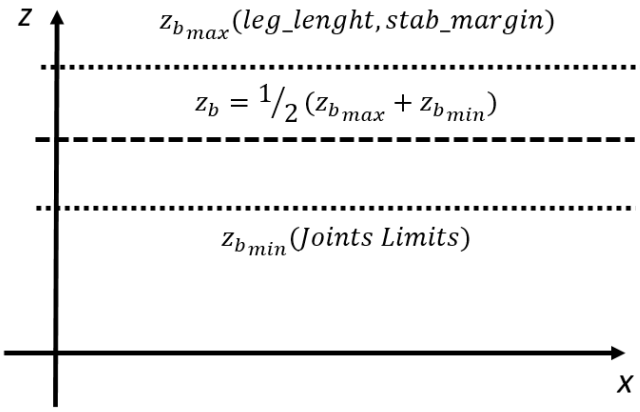


Figure 4.18: z_b limits on uneven terrains

it will not be possible to displace the CoG along x_b enough to reach the desired SM without loosing the contact between, at least, one of the feet and the ground; while, if it is too low, during the longitudinal movement one of the shoulders will run against its joints limits and this will cause the impossibility to keep a zero attitude of the trunk (e.g. if one of the front thigh hit the inner part of the shoulder's case during a forward CoG movement the shoulder will rise generating a negative pitch error). Fig. 4.18 shows the described dependencies.

The computation of $z_{b_{max}}$ and $z_{b_{min}}$ is presented in Alg. 3.

Algorithm 3 Computation of $z_{b_{max}}$ and $z_{b_{min}}$

procedure BODYHEIGHTLIMITS
  $x_{b_{future}} \leftarrow x_{b_{act}} + x_{SMdispl}$ ▷ CoG pos after displacement
  $SF_{max} \leftarrow MaxShoulderFeet\_xDist(x_{b_{future}})$
  $z_{b_{max}} \leftarrow \sqrt{(l_1 + l_2)^2 - SF_{max}}$ ▷ See Fig. 4.19
  $z_{b_{min}} \leftarrow z_{b_{max}}$
  while $\sim HitJointsLimits(x_{b_{future}}, z_{b_{min}})$ **do**
    $z_{b_{min}} \leftarrow (z_{b_{min}} - 0.0005)$
  end while
end procedure
function MAXSHOULDERFEET_XDIST(x_b)
  $\overline{SF}x_{F_r} \leftarrow (Shoulder_x - Foot_x)_{F_r}$ ▷ From the kinematic
  $\overline{SF}x_{F_l} \leftarrow (Shoulder_x - Foot_x)_{F_l}$
  $\overline{SF}x_{R_l} \leftarrow (Shoulder_x - Foot_x)_{R_l}$
  $\overline{SF}x_{R_r} \leftarrow (Shoulder_x - Foot_x)_{R_r}$
  return $max(\overline{SF}x_{F_r}, \overline{SF}x_{F_l}, \overline{SF}x_{R_l}, \overline{SF}x_{R_r})$
end function

After the computation of the limits the CoG is moved to $z_b = \frac{1}{2} z_{b_{max}} + z_{b_{min}}$,

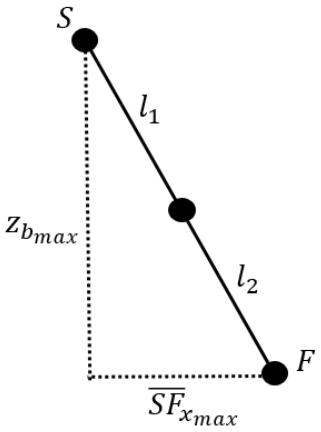


Figure 4.19: computation of $z_{b_{max}}$ from max *Shoulder-Foot* longitudinal distance

as shown in Fig. 4.18, to keep it as far as possible from the extreme points.
If ($z_{b_{max}} = z_{b_{min}}$ it means that the desired $SM_{threshold}$ is too big for this terrain surface and this joints limits therefore the locomotion algorithm will give an error code 1 and stop.

4.4.5 Unload

The unload phase provides a smooth transition of a^{des} from 0 to -1 which commands the *Four-Contact-Points* framework to unload the leg currently selected by the h_{flags}. The transition function is a polynomial with smoothness guaranteed by the *Hermite interpolation* presented in Sec. 3.4.

4.4.6 Step

The step phase is the only locomotion phase where the robot has only three feet in contact with the ground so it is also the only algorithm phase where the *Three-Contact-Points* framework is activated; the switch between the two framework is guaranteed smooth if done when the selected leg is already fully unloaded, as shown in Fig. 4.20.

The movement of the leg is commanded changing the desired feet positions and, in accordance with them, the desired *d task coordinates*; this affects the *inverse kinematics* of the leg which produces the desired motor angles θ, as ex-

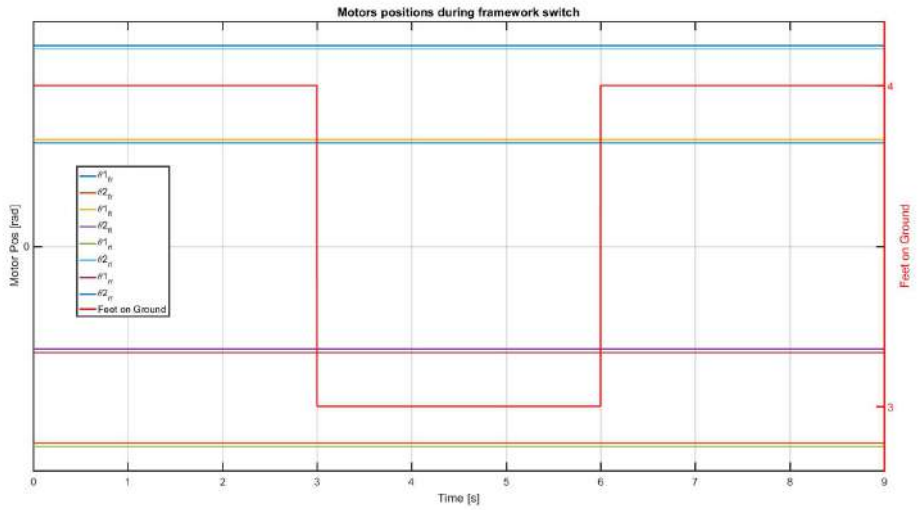


Figure 4.20: Motor desired angles during framework switch

plained in Sec. 3.2. The step trajectory are two smooth polynomials in x and z, obtained as illustrated in Sec. 3.4, which are shown in Fig. 4.21.

Since the presented robotic system do not make use of any visual feedback, the controller has no information on the terrain profile. The contact with the floor, which determines the end of the step, is implemented observing the contact forces, which are calculated as

$$f_{gc} = J^{-T} K(\theta - q), \tag{4.5}$$

where $f_{gc} = [f^x_{Fr} \ f^z_{Fr} \ f^x_{Fl} \ f^z_{Fl} \ f^x_{Rl} \ f^z_{Rl} \ f^x_{Rr} \ f^z_{Rr}]^T$ is the contact forces vector, $J \in \mathbb{R}^{8\times8}$ is the *jacobian matrix* of the system (always invertible if the legs are not in a singular configuration) and K is the stiffness matrix of the springs. Hence, the condition for the contact detection on leg i is $f^z_i > f_{threshold}$ ans the threshold has to be tuned tanking into account the different amount of friction in each joint.

Observing Fig. 4.21 it is possible to notice that the maximum step length is reached on flat ground. While during locomotion on uneven terrains the step is shorter because, if the leg is facing a positive slope (i.e. the final point of the step is higher that the initial one), the contact is detected before reaching the maximum and, if it is facing a negative slope or a little hole, after the maximum distance the leg will keep extending while coming back along x to evade kinematic singularities.

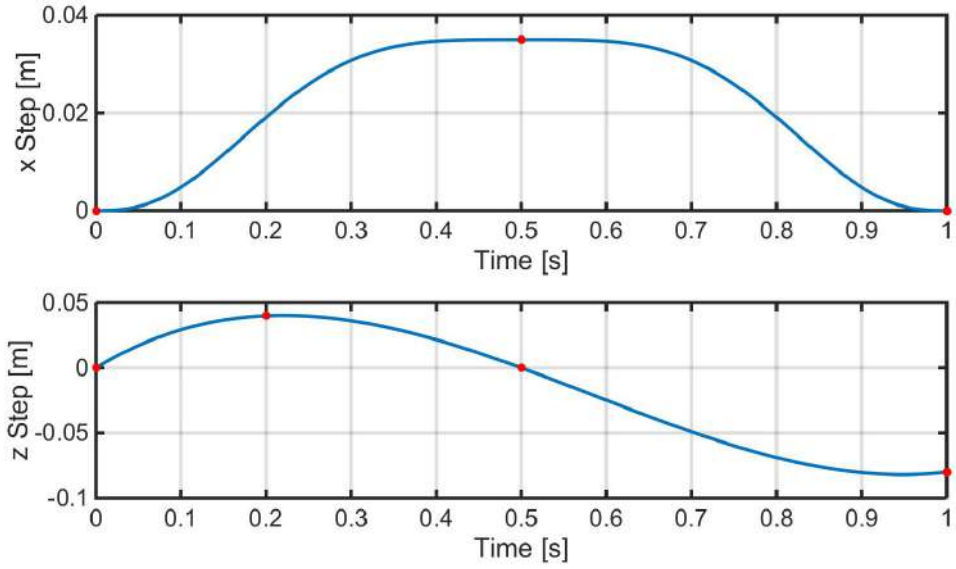


Figure 4.21: Step trajectories

Since the steps are not equal among them the maximal step length (i.e. the step length achieved on flat grounds) is not a fixed length but is computed as the sum of the longitudinal distance between the foot and the shoulder at the beginning of the step and a fixed distance S, as depicted in Fig. 4.22.

4.4.7 Load

At the beginning of the load phase the *Four-Contact-Points* framework is reactivated. This phase of the algorithm is the dual phase of the unload one; therefore, it provides a smooth transition, through *Hermite interpolation*, of a^{des} from -1 to 0 to bring back $\frac{1}{4}$ of the total load on the controlled leg.

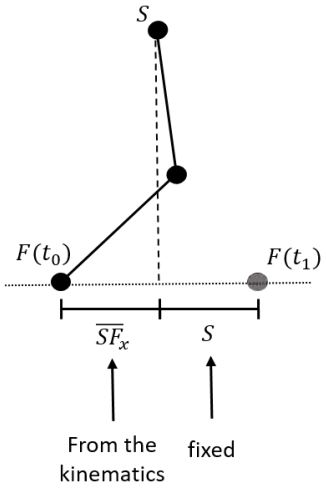


Figure 4.22: Computation maximal step length

Chapter 5

Simulations

The proposed method has been evaluated in a *Simulink* environment which takes the desired motor angles as input and simulates the complete dynamics of the system, including the contacts with the floor. These latter were modeled as spring-damper systems with unidirectional normal forces (i.e. the foot can not be pulled but only pushed); the used spring constant is $10^6 \frac{N}{m}$ and the damping is $2000 \frac{Ns}{m}$. The friction coefficient with the floor has been set to 0.3.

A custom visualizer with the robot structure has been coded to show the stability property in every locomotion phase.

A set of incremental complexity simulations have been performed to evaluate the limits of the system. Considering the strong link between the accuracy of the *virtual equilibrium* following and the load/unload performances, which are crucial for a robust locomotion, we firstly evaluate the precise goal reaching capabilities of the control system. Fig. 5.1 and 5.2 shows that the desired position and orientation of the CoG, during a purely displacing phase (i.e. when no steps are made), is followed with high accuracy.

Having proofed the satisfactory *virtual equilibrium* following, in order to achieve a stable locomotion, we show in Fig. 5.3 the normal ground contact forces on each leg during the unloading (from $t = 2s$ to $t = 4s$) and subsequent loading (from $t = 6s$ to $t = 8s$) of the front right leg. On the right ordinate axis it is

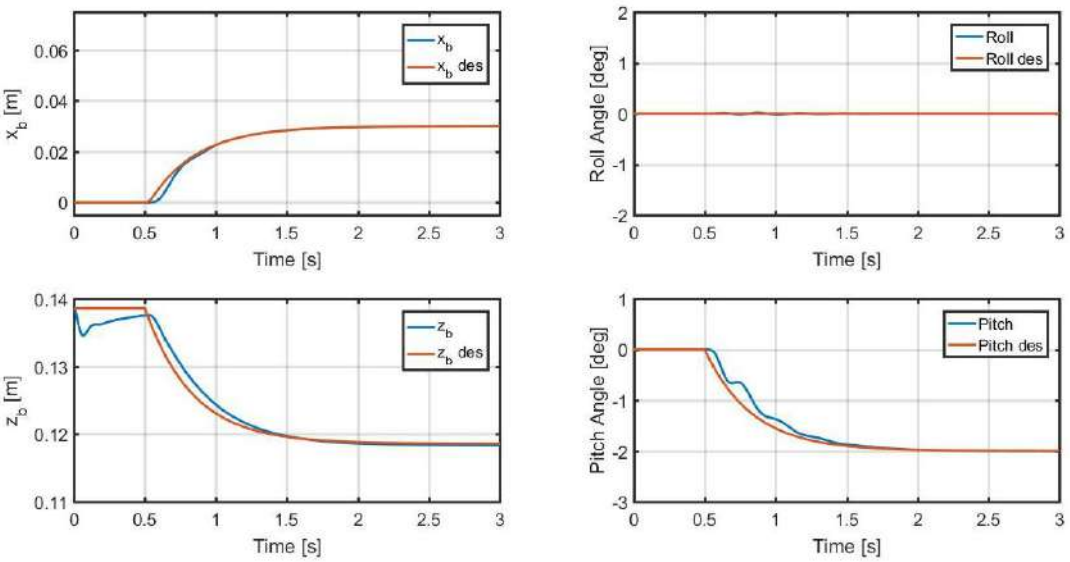
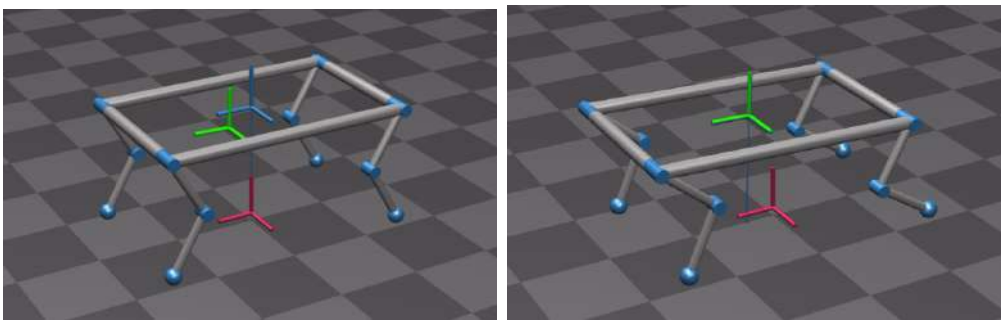


Figure 5.1: Simulation 1: precise goal reaching capabilities, *task coordinates*



(a) Start time **(b)** Final time

Figure 5.2: Simulation 1: visualization of precise goal reaching capabilities

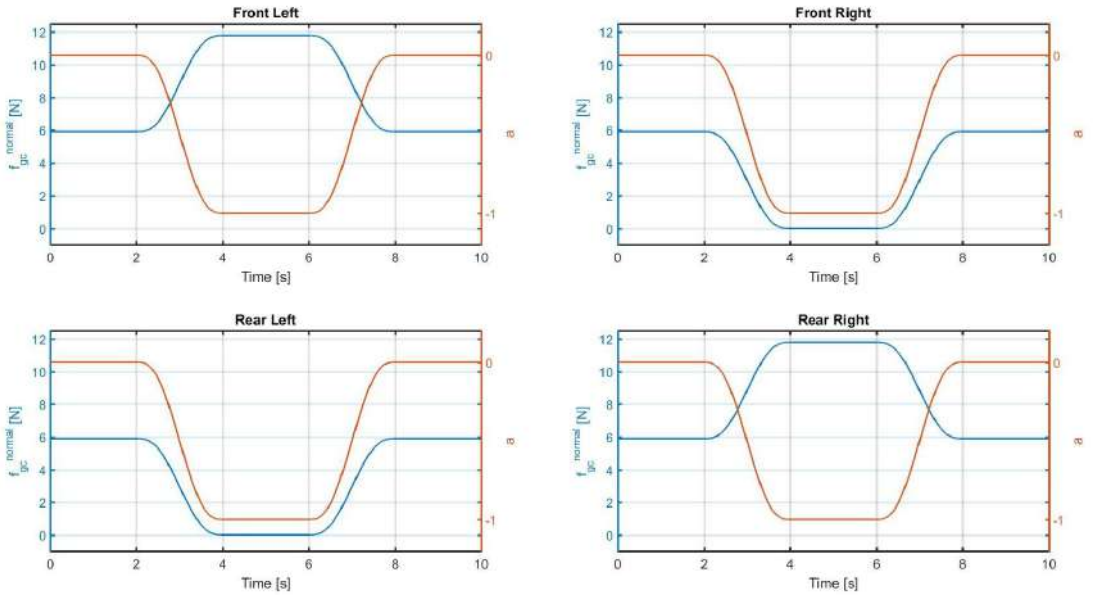
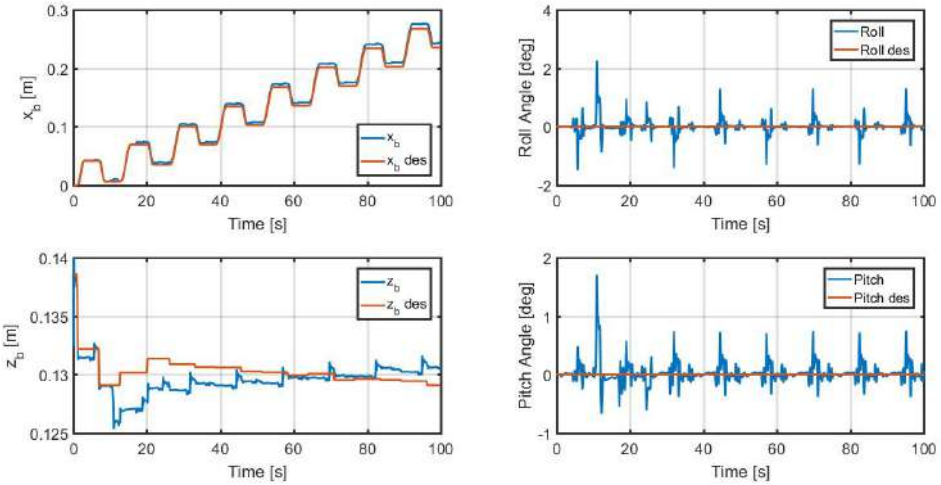


Figure 5.3: Simulation 2: loading unloading of *front right* leg

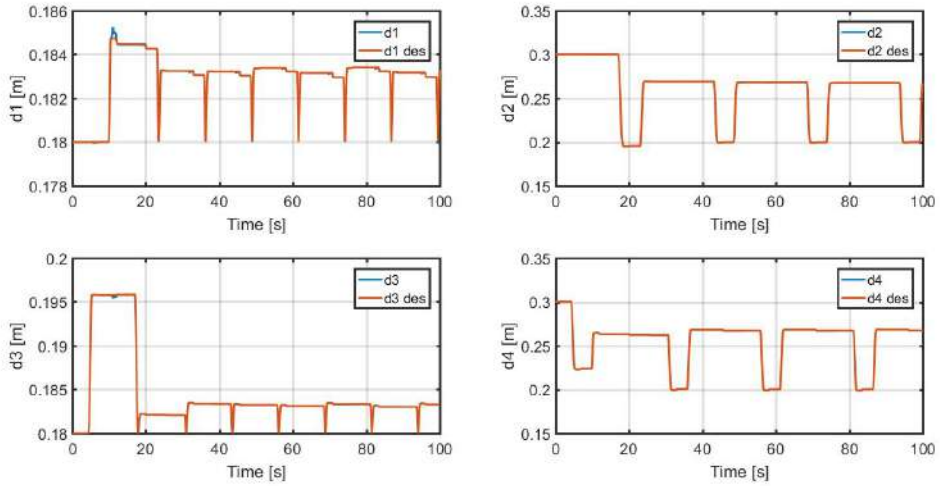
reported the plot of the a^{des} *task coordinate*, which control the force distribution on the selected leg, in turn reported on the left ordinate; we can observe the force profile going correctly to zero (leg unloaded) and coming back to the balanced load distribution when a^{des} goes to -1 and back to 0. It is here interesting to notice the force distribution coupling between the legs: is has been observed in simulation and experiments that the load distribution of the $i-th$ leg is always coupled with the $i+2$, in other words, the coupling acts on the legs on the two sides of each diagonal of the shoulders/hips rectangle.

Once demonstrated the locomotion's requirements separately we run three simulations with incremental level of harshness to proof the walking capabilities and limits.
Considering the first one of the introduced set of simulations, where the robot walks on even terrain, Fig. 5.4 and 5.5 depict respectively the *task coordinates* and the achieved joints angles accuracy. From the joints angles comparison is has also been validated the strategy of using the direct joints angles as motors angles to command the swing phase (Remark 2). Meanwhile, Fig. 5.6 show the stability margin of the robot before each one of the four possible steps.

The same paradigm of evaluation has being used for *Simulation 4* (in Figg.

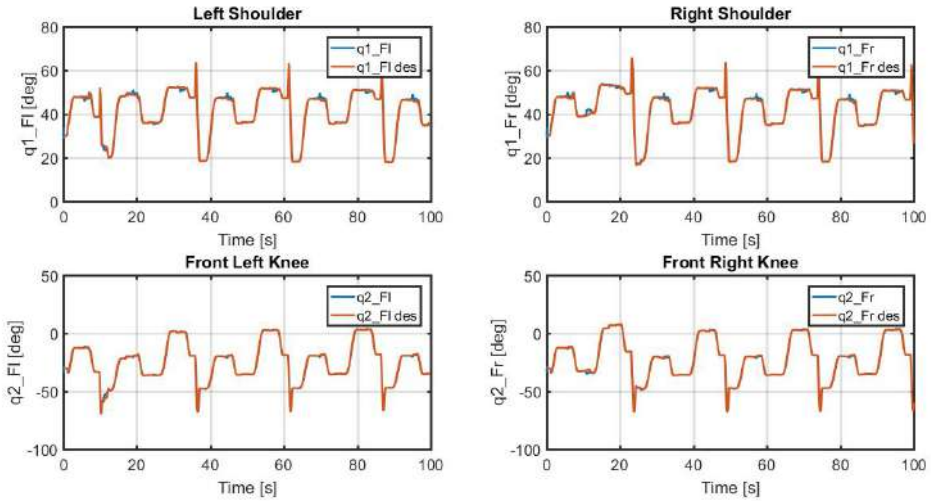


(a) Trunk *task coordinates*

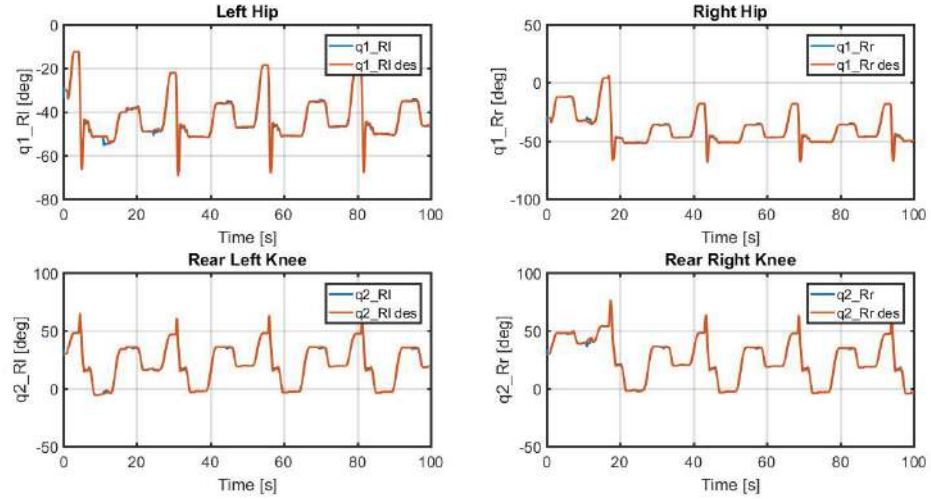


(b) Feet *task coordinates*

Figure 5.4: Simulation 3: locomotion on even terrain, *task coordinates*



(a) Front legs joints angles



(b) Rear legs joints angles

Figure 5.5: Simulation 3: locomotion on even terrain, joint angles

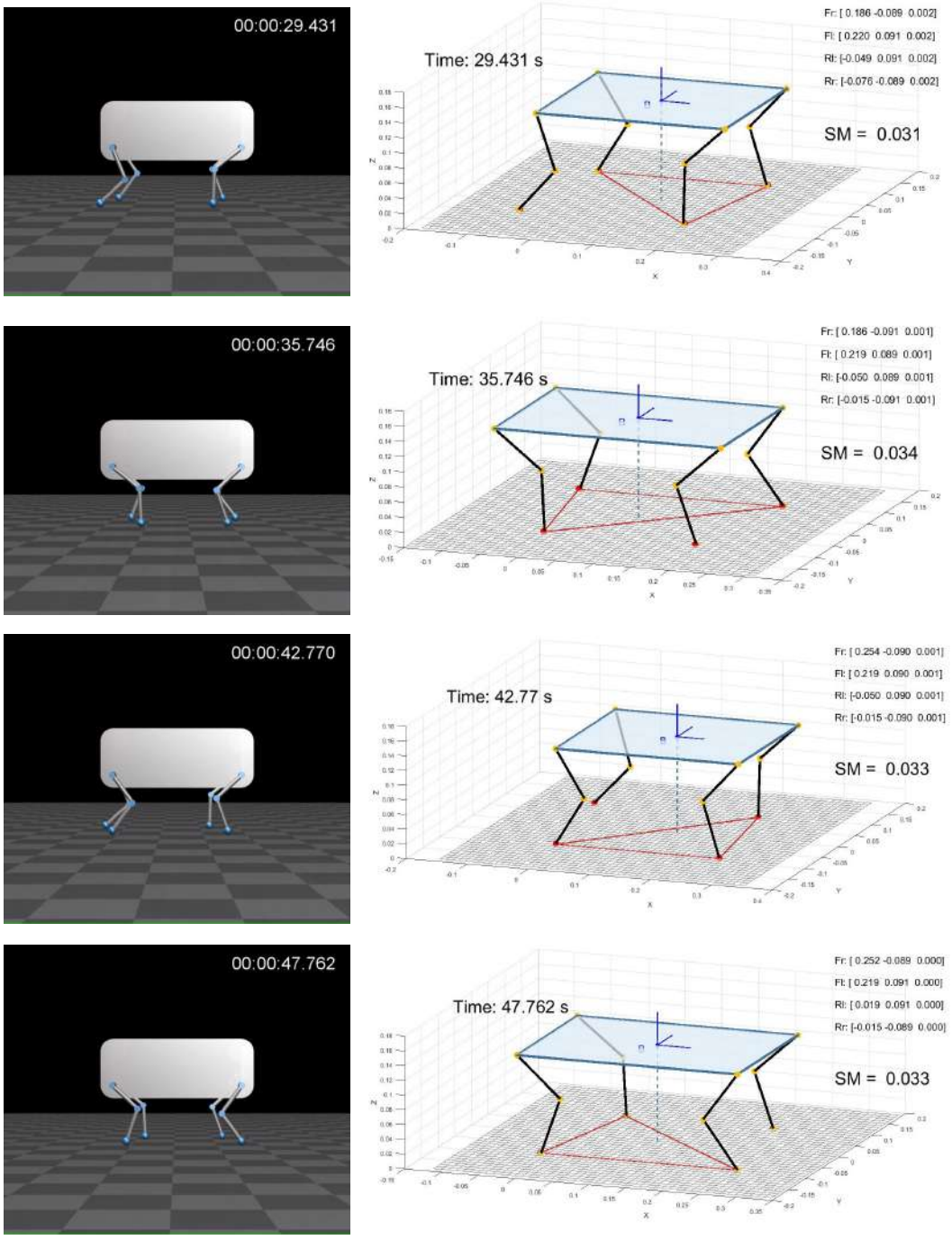
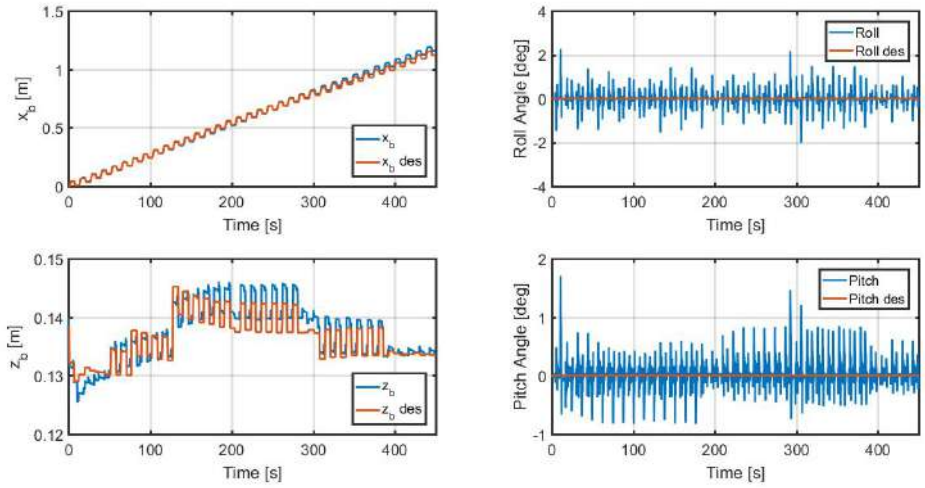
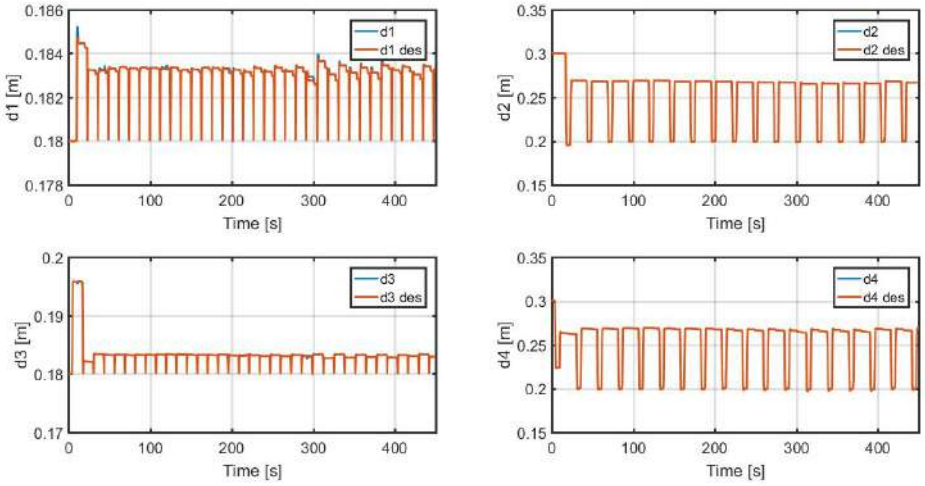


Figure 5.6: Simulation 3: locomotion on even terrain, stability margin



(a) Trunk *task coordinates*

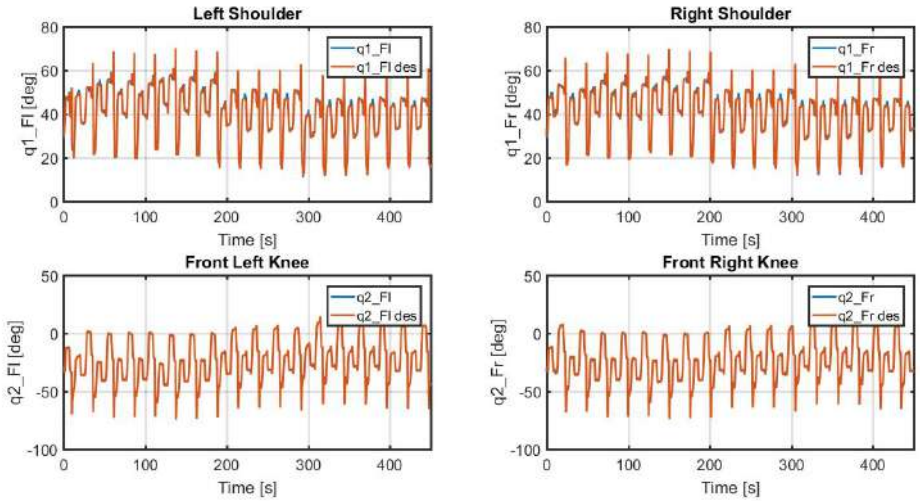


(b) Feet *task coordinates*

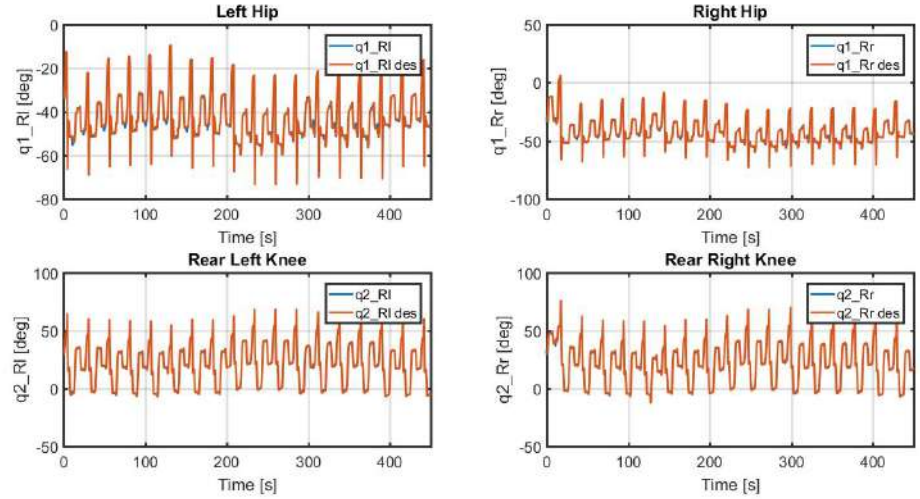
Figure 5.7: Simulation 4: locomotion uneven terrain, *task coordinates*

5.7, 5.8 and 5.9) where the robot climbed over a set of obstacles of which it was completely not aware. In this context, the stability margin (Fig. 5.9) has been reported for the hardest balancing moments. The shape of the obstacle is depicted in Fig. 5.10.

With the aim of exploring the limits of the control system, the last of the three complete simulations used a more rough set of obstacles (Fig. 5.14) which, in terms of versatility, are far more demanding of the structural joints limits of the robot. Hence, in this simulation we disabled the joints limits. Figg. 5.11, 5.12 and 5.13 show the results.



(a) Front legs joints angles



(b) Rear legs joints angles

Figure 5.8: Simulation 4: locomotion on uneven terrain, joint angles

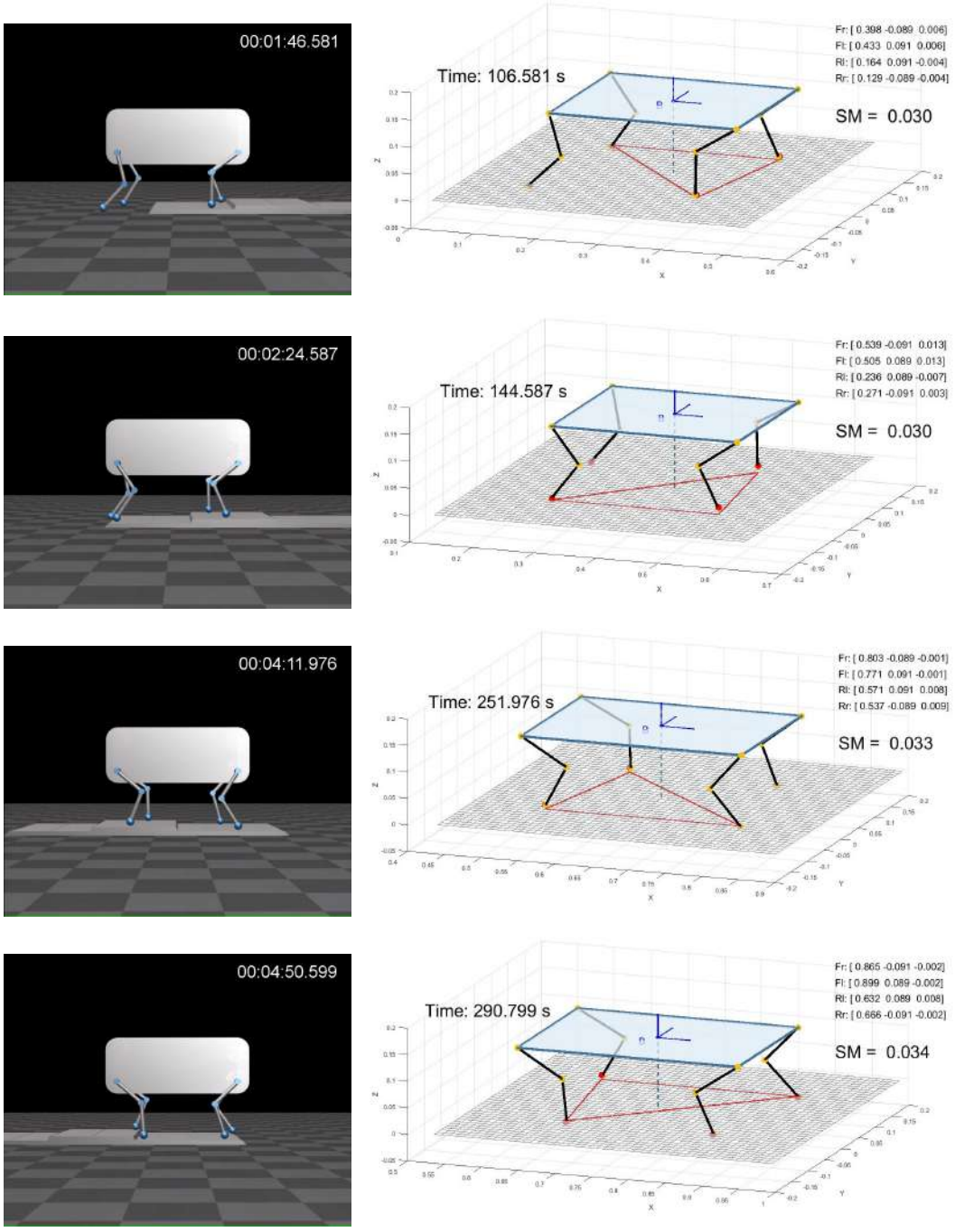


Figure 5.9: Simulation 4: locomotion on uneven terrain, stability margin

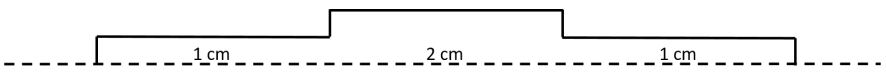
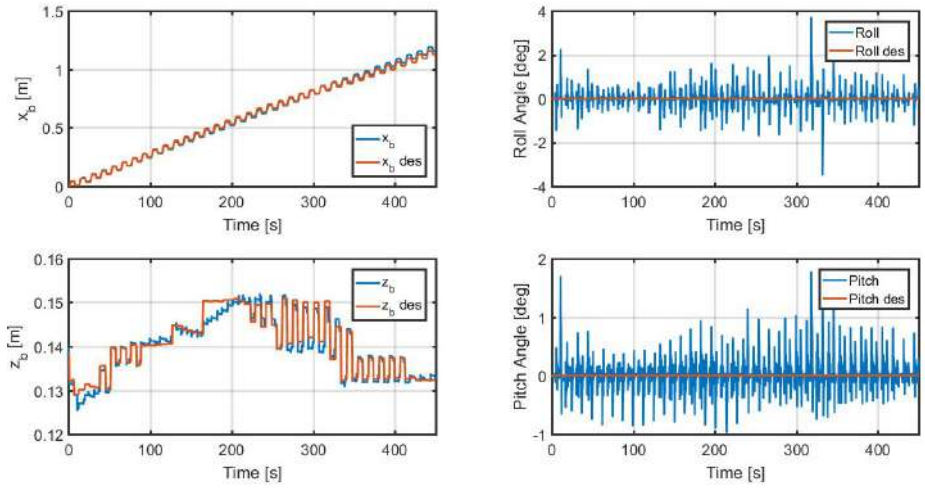
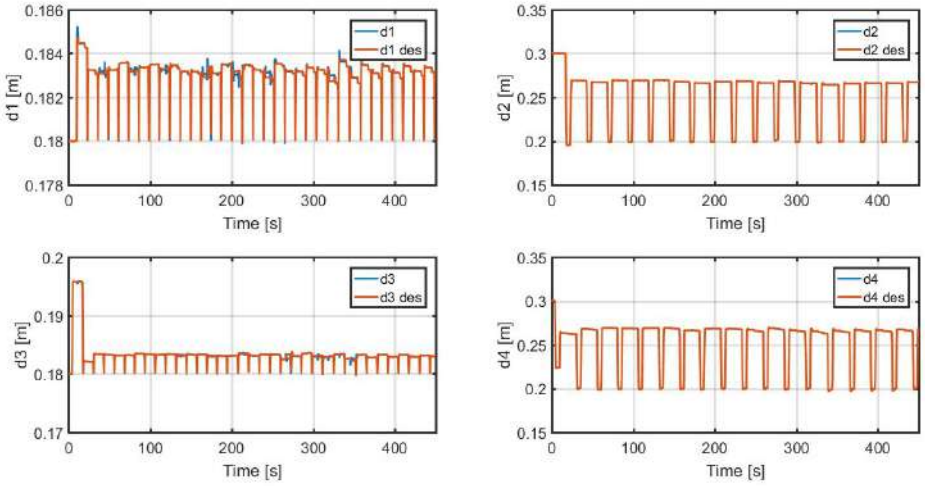


Figure 5.10: Obstacles used in *Simulation 4*



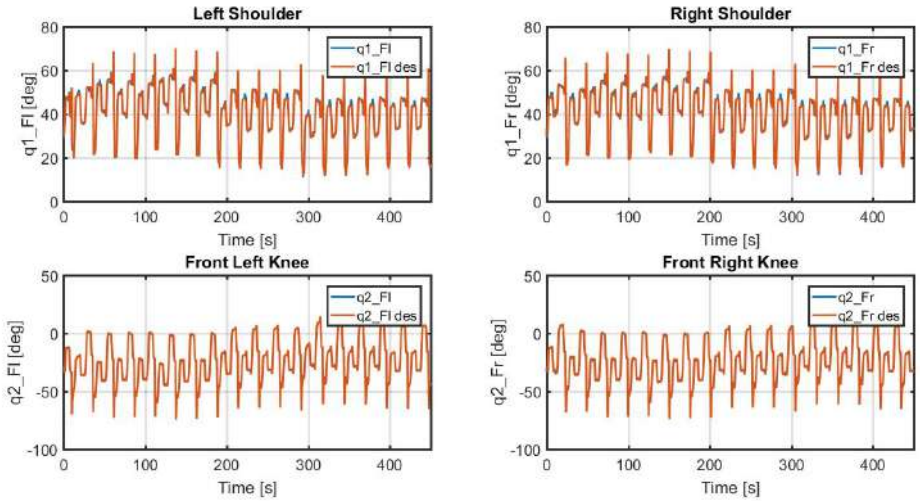
(a) Trunk *task coordinates*



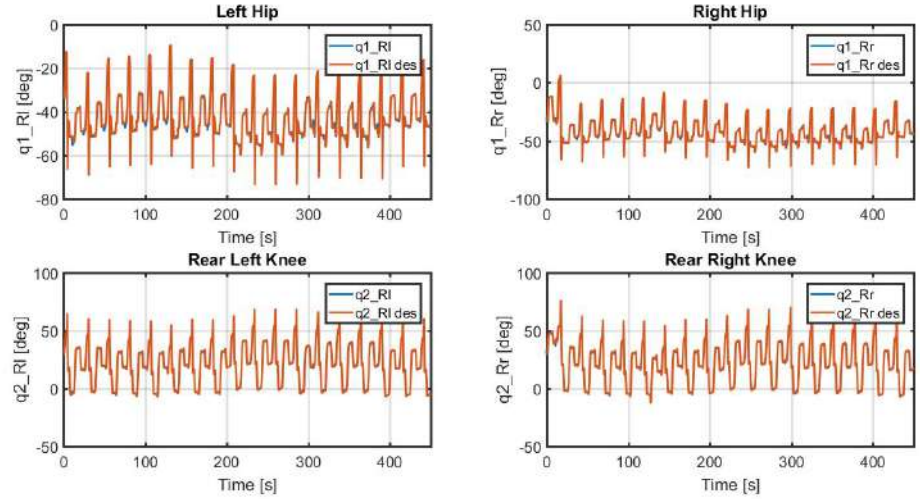
(b) Feet *task coordinates*

Figure 5.11: Simulation 5: locomotion rough terrain, *task coordinates*

From this set of incremental difficulty simulations we can infer that, in ideal environments, the proposed control system achieved a level of robustness and stability which goes much further than the actual possibilities of the robot. The desired trajectories have been followed completely in both joints and task space; the stability was always respected with a $SM > 3cm$, showing no correlation with the harshness of the terrain.



(a) Front legs joints angles



(b) Rear legs joints angles

Figure 5.12: Simulation 5: locomotion on rough terrain, joint angles

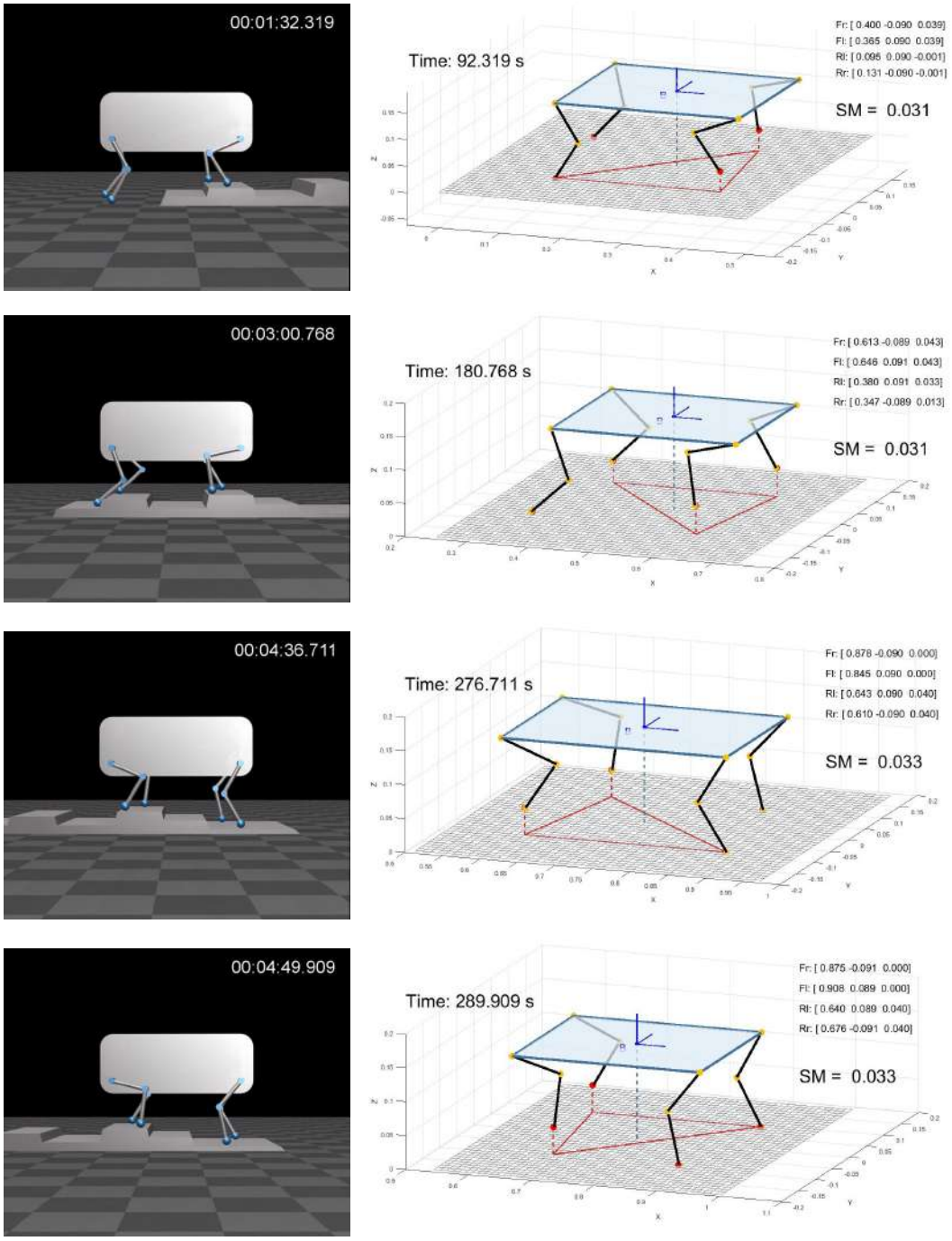


Figure 5.13: Simulation 5: locomotion on rough terrain, stability margin

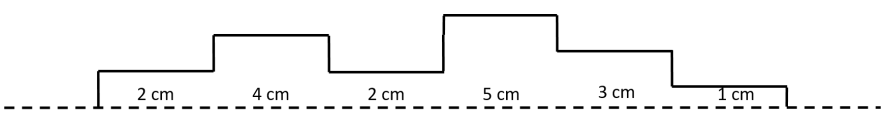


Figure 5.14: Obstacles used in *Simulation 5*

Chapter 6

Experiments and Conclusions

In this chapter we are going to present the results of two experiment on the robotic platform described in 1.3 with the aim of infer a conclusive response on the effectiveness of the proposed methodology.

The first experiment is composed by a preliminary walking on flat floor, followed by an obstacle which is 8% the distance between the trunk and the floor. The experimental data have been evaluated in terms of stability and a graphical results has been produced using the same visualizer used for the simulations and the experiment's log files as inputs; Figg. 6.1 and 6.2 reports the stability margin in graphical and numerical form proofing that the achieved stability margin is not influenced by little obstacles. Figg. 6.3 and 6.5 contain data of the whole experiment, therefore, being not clearly understandable where the obstacle was, we can infer that the desired trajectory has been followed in terms of attitude and joints angles. Fig. 6.4 shows the joints limits, computed by the planner according to Alg. 3 in Sec. 4.4.4; these quantities give us a clarifying picture of the proximity to the joints limits in every time of the execution. Inspecting Alg. 3, the reader will notice that when a desired displacement of the CoG is going to generate a joints limits collision the computed $Z_b^{max} = Z_b^{min}$. Hence, the closer Z_b^{min} is to Z_b^{max} and the closer is the system to its limits. Since the joints limits violations represent an unfeasible situation (because the desired displacement of the CoG is the minimum required amount to respect the desired $SM_{threshold}$, therefore an impossibility to achieve the desired displacement is an impossibility to solve the stability requirement) the planner is coded to stop the execution until a reset

state rises.

In the second experiment we are going to show the stability properties on a more rough and uncertain terrain, we put a number of obstacles in a random disposition along the path of the robot, Fig. 6.6, and we covered them with a flexible rubber carpet to avoid the slippage of the feet down from the sharp edges of the tiles. Figg. 6.7, 6.8 and 6.10 show that the stability margin has been respected also in this test while the trajectory following have not suffered the increased harshness of the terrain. From Fig. 6.9, it is possible to see that the system was close to the stop condition due to the joints limits.

From the performed experiments we can conclude that the system achieved a satisfactory stability level. Highly compliant systems are especially suited to be efficient in dynamic tasks, due to their capabilities to store and release energy into the elastic components, but the decoupling introduced by the springs harm the overall system controllability. Considering the results attained in this work, we can infer that, thanks to the spring passive disturbance rejection, which acts like a natural feedback controller, this controllability is not really required to perform static task without any change to the system hardware.

To increase further the reachable stability during the static walking task we considered to include an identified friction model in the system one. Since, the system is let free to oscillate around the desired virtual equilibrium the force disturbance acting on each shoulder after the relative step, especially in case of aggressive contacts with a rigid soils, creates oscillations which, in presence of friction, make the system slowly drift from the desired virtual equilibrium. This phenomena, if not compensated with an ad-hoc attitude reference or with the inclusion of the friction in the model, affects the loading and unloading capabilities on the long run. From the hardware point of view, the main restrainers to an improved stability are the leg length and the joints limits. The latter can be overcame setting the robot in *knee outside* configuration, but this will also require higher torques which can be produced with more stiff springs. Regarding the joints limits, it comes directly from the optimality proof in [1] that the optimal crawling gait can be reached only from those systems which have legs long enough to put each rear foot in the footprint of the front one on the same side of the trunk. The computation of an exact leg length to achieve the optimal static gait strongly depends on the desired trunk height during the locomotion, however a leg to trunk length ratio of $\frac{3}{4}$ is reasonably enough.

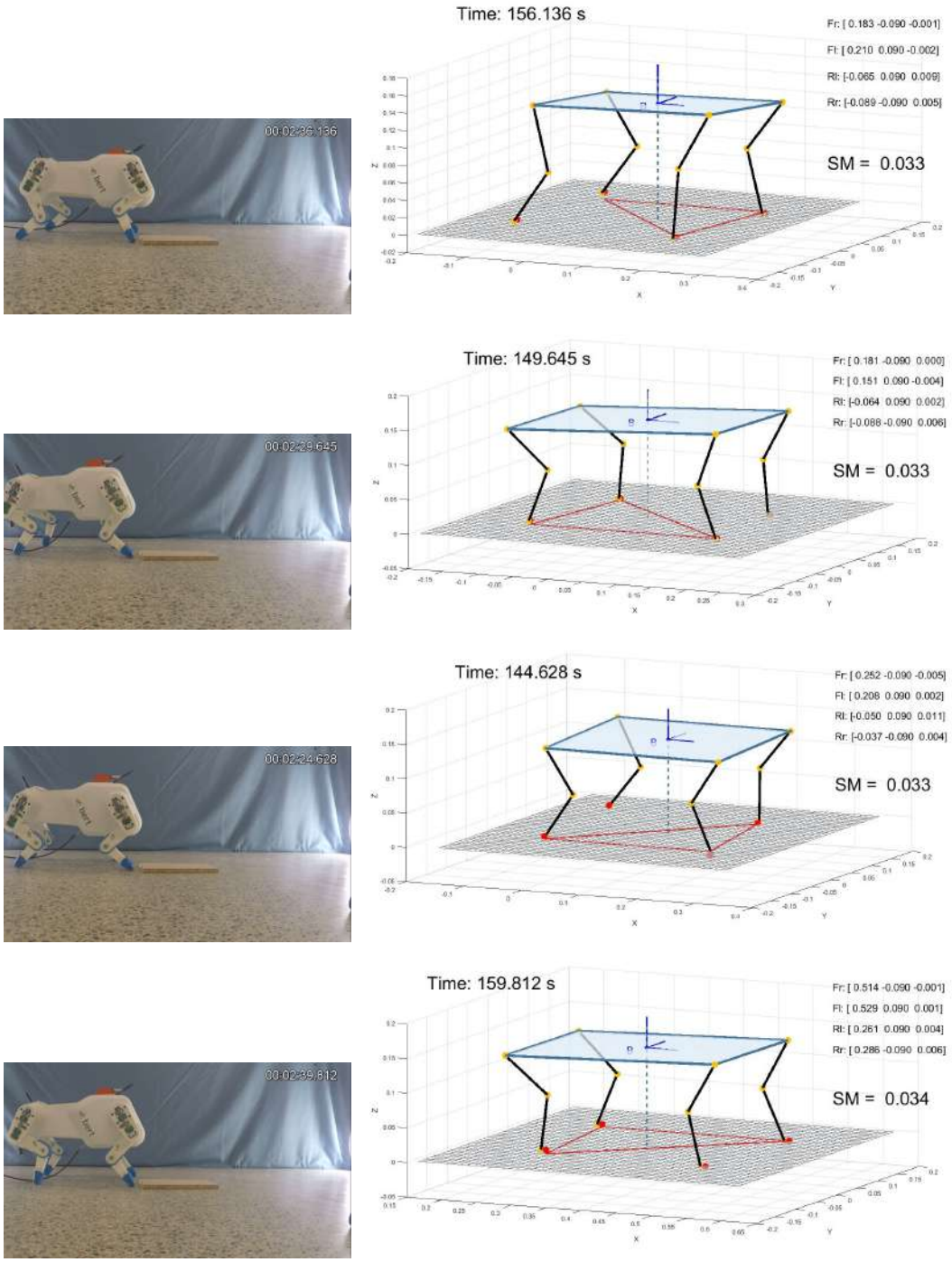


Figure 6.1: Experiment 1: locomotion on flat ground, stability margin

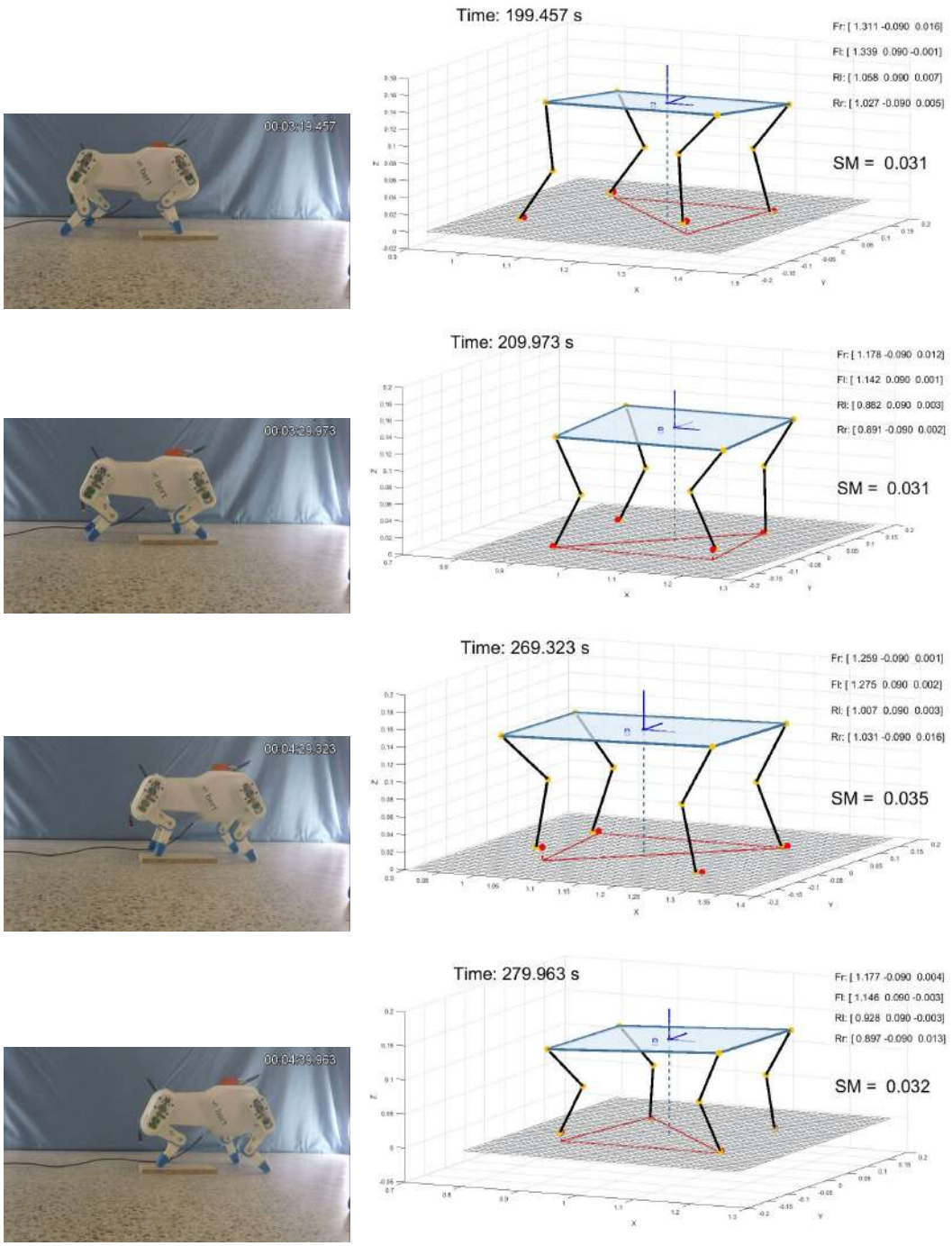


Figure 6.2: Experiment 1: locomotion on uneven terrain, stability margin

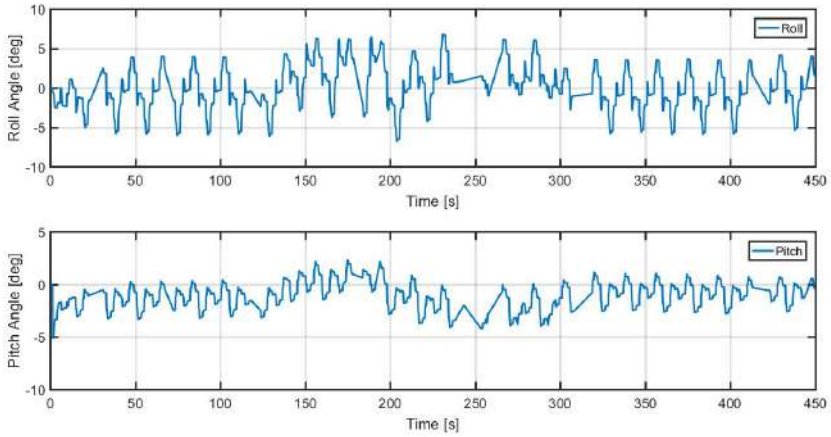


Figure 6.3: Experiment 1: locomotion on uneven terrain, measured euler angles

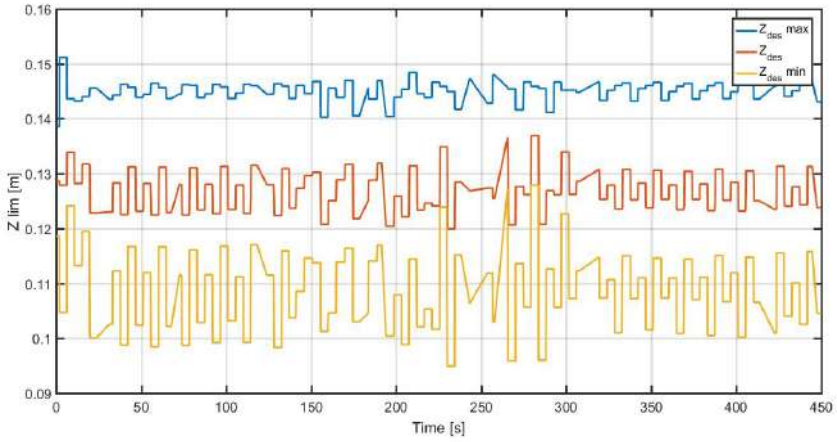
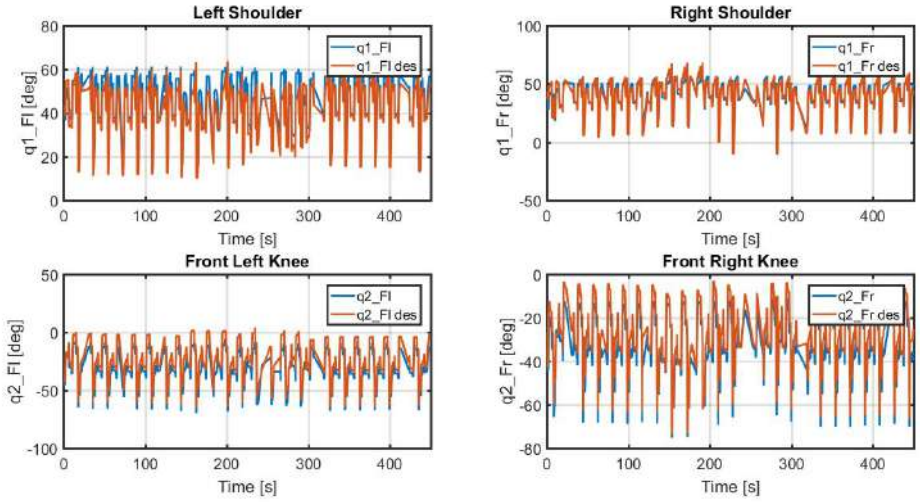
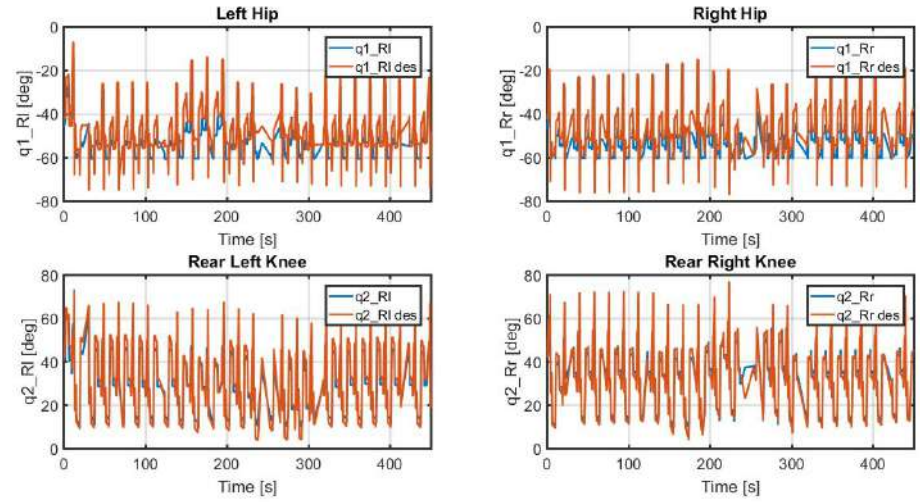


Figure 6.4: Experiment 1: locomotion on uneven terrain, limits on the Z_b^{des} *task coordinate*



(a) Front legs joints angles



(b) Rear legs joints angles

Figure 6.5: Experiment1: locomotion on uneven terrain, joint angles

Figure 6.6: Experiment 2: obstacles

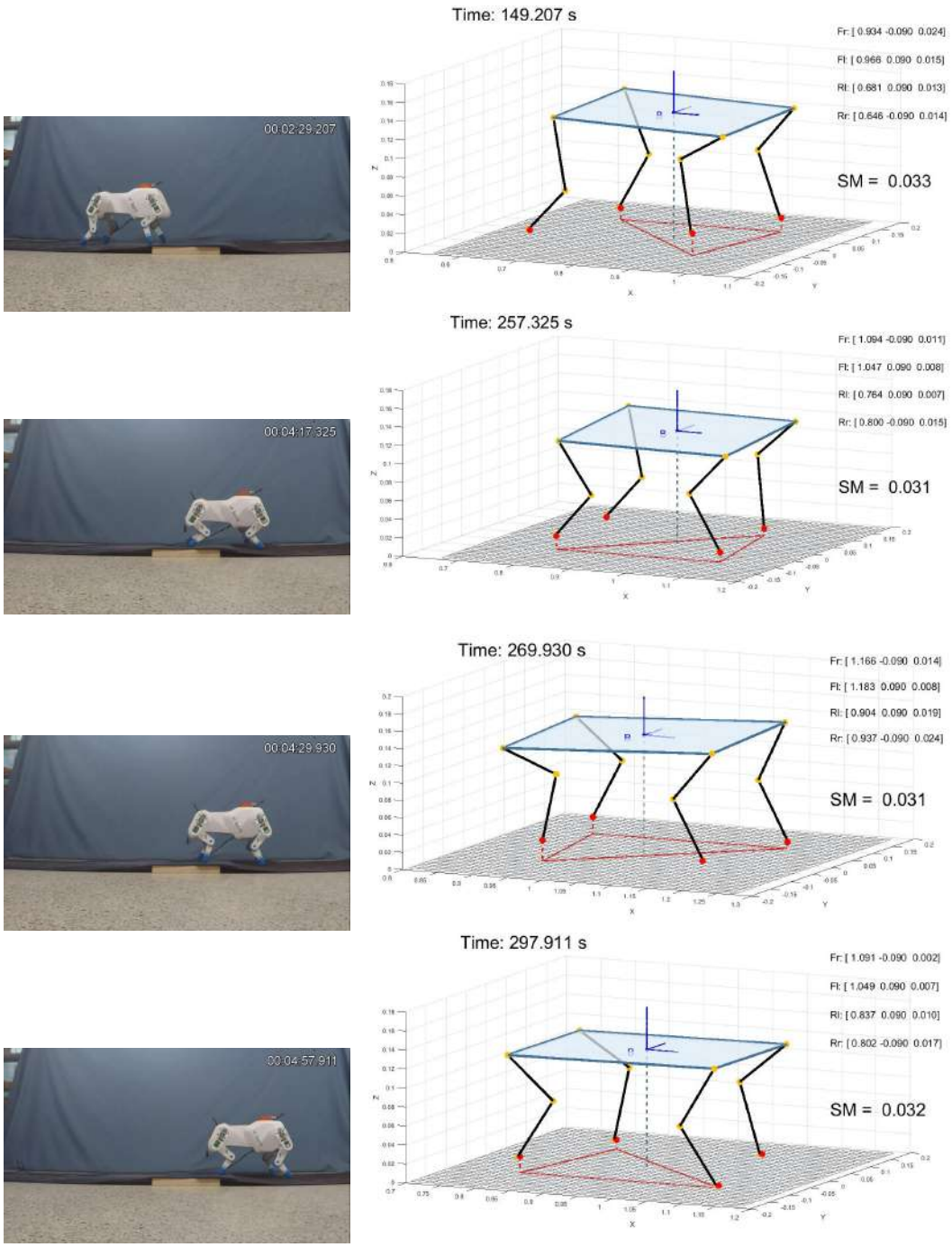


Figure 6.7: Experiment 2: locomotion on rough terrain, stability margin

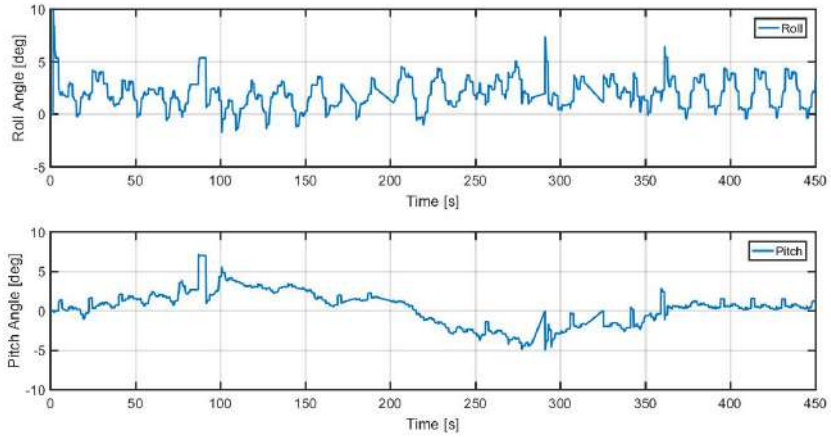


Figure 6.8: Experiment 2: locomotion on rough terrain, measured euler angles

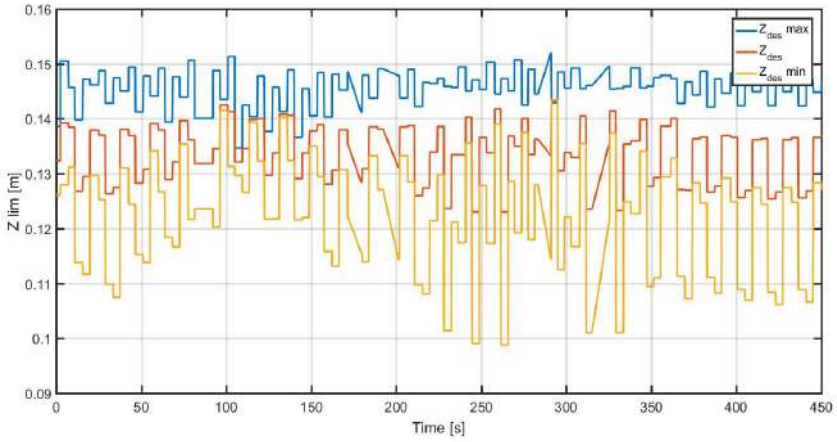
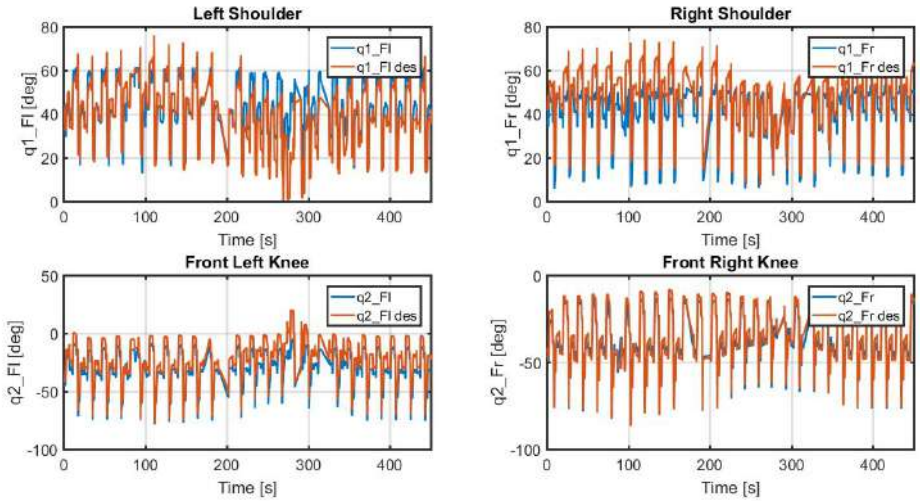
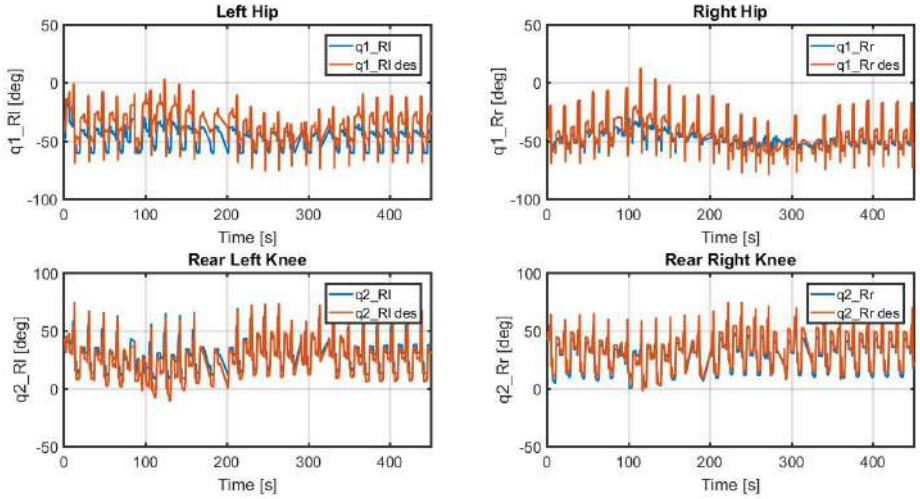


Figure 6.9: Experiment 2: locomotion on rough terrain, limits on the Z_b^{des} *task coordinate*



(a) Front legs joints angles



(b) Rear legs joints angles

Figure 6.10: Experiment2: locomotion on rough terrain, joint angles

Bibliography

[1] R. B. McGhee and A. A. Frank, *On the Stability Properties of Quadruped Creeping Gaits*, Mathematical Biosciences, vol.3, no.2, pp.331-351, 1968.

[2] J. Zico Kolter, Mike P. Rodgers and Andrew Y. Ng, *A Control Architecture for Quadruped Locomotion Over Rough Terrain*, *IEEE International Conference on Robotics and Automation (ICRA)*, May 2008, Pasadena, CA, USA

[3] Marco Hutter, *Design and Control of Legged Robots with Compliant Actuation*, PhD thesis, 2013.

[4] R. Hodoshima, T. Doi, Y. Fukuda, S. Hirose,T. Okamoto, J. Mori, *Development of TITAN XI: a Quadruped Walking Robot to Work on Slopes*, *IEEE International Conference on Intelligent Robots and Systems*, Sep 2004, Sendai, Japan

[5] M. Raiber, K. Blankespoor, G. Nelson, R. Playter, *BigDog, the Rough-Terrain Quadruped Robot*, *17th World Congress The International Federation of Automatic Control*, Jul 2008, Seoul, Korea

[6] A. Bicchi, M.A. Peshkin and J.E. Colgate, *Safety for Physical Human-Robot Interaction*, *Springer Handbook of Robotics*, 2008, Springer

[7] D. Lakatos, A. Albu-Schäffer, C. Rode and F. Loeffl, *Dynamic Bipedal Walking by Controlling only the Equilibrium of Intrinsic Elasticities*, *IEEE International Conference on Humanoid Robots (Humanoids)*, Nov 2016, Cancun, Mexico

[8] D. Lakatos, C. Rode, A. Seyfarth and A. Albu-Schäffer, *Design and Control of Compliantly Actuated Bipedal Running Robots: Concepts to Exploit Natural System Dynamics*, *IEEE International Conference on Humanoid Robots (Humanoids)*, Nov 2014, Madrid, Spain

[9] G.A. Pratt and M. Williamson, *Series Elastic Actuators*, *IEEE Intelligent Robots and Systems*, Aug 1995, Pittsburgh, PA, USA

[10] S. Kajita and B. Espiau, *Legged Robots*, *Springer Handbook of Robotics*, 2008, Springer

[11] M.H. Raibert, *Legged Robots that balance*, 1986, MIT Press

[12] G.A. Pratt, *Legged robots at MIT: what's new after Raibert*, *IEEE Robotics & Automation Magazine*, Vol. 7, Issue 3, Sep 2000

[13] C. Semini, N.G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella and D. G. Calwell, *Design of HyQ - a hydraulically and electrically actuated quadruped robot*, *Journal of Systems and Control Engineering*, Vol. 225, Issue 6, Feb 2011