# Displacement of a virtual manikin in a cluttered environment : simulation of movement by integrating equilibrium constraints

Zhaopeng Qiu

## To cite this version:

Zhaopeng Qiu. Displacement of a virtual manikin in a cluttered environment : simulation of movement by integrating equilibrium constraints. Automatic. Université Pierre et Marie Curie - Paris VI, 2012. English. NNT : 2012PAO66537 . tel-00836087

HAL Id: tel-00836087
https://theses.hal.science/tel-00836087v1

**Thèse de DOCTORAT**

de

l'Université PARIS VI

Spécialité : mécanique - robotique (Ed SMAER)

# DÉPLACEMENT D'UN MANNEQUIN VIRTUEL DANS UN ENVIRONNEMENT ENCOMBRÉ : SIMULATION DE MOUVEMENT EN INTÉGRANT LES CONTRAINTES D'ÉQUILIBRE

Proposé par : Zhaopeng QIU

Pour obtenir le titre de DOCTEUR de l'Université PARIS VI

Soutenue le 05 décembre 2012 devant le jury :

| | | |
|---|---|---|
| BOULIC Ronan | Maître d'enseignement et de recherche à l'EPFL | Rapporteur |
| CHABLAT Damien | Professeur à l'Ecole Centrale de Nantes | Rapporteur |
| BIDAUD Philippe | Professeur à l'ISIR, Université Paris VI | Examinateur |
| GELIN Rodolphe | Chef de Project de l'Aldebaran Robotics | Examinateur |
| MICAELLI Alain | Directeur de Recherche au LSI, CEA/LIST | Directeur de Thèse |
| ROBERT Thomas | Chargé de Recherche au LBMC, IFSTTAR | Co-directeur |
| ESCANDE Adrien | Chercheur au CNRS-AIST JRL | Co-encadrant |

Directeur de Thèse (CEA) : Alain MICAELLI
Co-directeur (IFSTTAR) : Thomas ROBERT
Co-encadrant : Adrien ESCANDE

# Contents

# Abstract

Nowadays, digital human models are broadly used in many research domains and industries. A digital human can represent human body characteristics and simulate human behaviors and capacities, thus we can evaluate or predict human performances in a scenario via simulations with digital human models. Currently, ergonomic evaluation of human motions depends largely on kinematic human models and Motion Capture techniques. Dynamic human models and associated control techniques have advantages over kinematic ones in terms of physical feasibility and they may lead to a reduced number of experiments with real human subjects. However, the dynamics-based techniques are very time-consuming when they are applied in complex scenarii, especially when the environment is cluttered. In this context, this thesis aims to explore an efficient dynamics-based method for digital human motion simulation in cluttered environments. The method is aimed to be applied in ergonomic study of daily human activities, such as car-ingress or car-egress motions.

At the beginning of this study, we carry out MoCap experiments in which we obtain a set of recorded human motions in some cluttered environments. We learn human motion characteristics and principles in cluttered environments by observing and analyzing the recorded motions. With the help of the obtained heuristic knowledge, we make reasonable hypotheses and choose the main problematics of this research.

Then we work on the topic of balance. A simplified digital human model for balance formulation is proposed in which the human is taken as a mass point located at its center of mass (CoM) interacting with the environment via a few supports (contact or grasps). Support feasibility (stability) brings constrains on exterior wrenches applied at supports. The constraints then result in an admissible space for the pseudo-wrench (a 6D vector calculated from position and acceleration of CoM) of the model. The admissible pseudo-wrench space defines a criterion for verifying balance of a digital human model during its motions. The "stability margin" concept associated with this criterion is studied which evaluates robustness (or quality) of a balance state.

Thereafter, a hierarchical framework is developed which can realize motion simulations via three levels. Firstly, a global CoM trajectory is generated at a global level. The CoM trajectory is generated by optimization under motion state imposition, balance, geometric and timing constraints. Time durations for transition phases are generated as well. Then at a local level, the trajectories of end-effectors (e.g. foot or hand) and whole-body postures are piecewisely generated under the constraints of inverse kinematics and collision-freeness. The results obtained at the previous two levels are finally used as control references in dynamic simulation. A dynamic controller actuates the dynamic digital human model to realize the generated motion.

Approaches at each level of the framework are applied and tested in several scenarii. Eventually, the framework is applied in a car-ingress scenario relying on heuristics of car-ingress motions.

# Acknowledgment

I would like to firstly thank my adviser Alain MICAELLI. During the three years of Ph.D study, I have not only received excellent tutoring from him but also learnt from him the serious, assiduous and persistent attitude towards the scientific life. Throughout the three years, he has always been responsible and serious on both the global research direction and the technique details. Very kind and patient, he makes frequently conversions with me to follow the advancement and discuss the problems in the study.

I would like to thank my co-adviser Thomas ROBERT who is a young researcher in LBMC. During the thesis study, I have passed about one year in the laboratory LBMC for carrying out the experimental work and testing our methods in real scenarii. Specializing in Ergonomics and Biomechanics, he has taught me a lot on MoCap experiments, DHM-based motion reconstructions and ergonomic analysis of human motions. Via frequent face-to-face or telephone conversations, he has inspected seriously my study and has proposed many important research topics.

Many thanks to my co-adviser Adrien ESCANDE who is a young researcher in LSI with vast knowledge and rich experiences in robotics. During the thesis, I have learned a lot from him including mathematical knowledge, programming skills and practical techniques in robotics. As an expert in motion planning, he has helped a lot for developing the motion planning framework in this thesis.

Many thanks to Paul EVRARD, another young researcher in LSI and an expert in robot motion control and programming. He has developed the interface DSIMI for the software XDE which facilitates the simulation work in this study. He has kindly helped me in the programming for carrying out simulations in XDE.

Many thanks to all the colleagues in both LSI and LBMC for their supports to this research.

I am very grateful to my friend ZHENG Xiaohan for always encouraging me and believing in me during the course of this thesis. With her joyful spirit and her optimistic attitude, she has given me a lot of comforts and reduced my stress during the thesis. I am grateful to my friends GAO Zhenlan and WANG Wei for their supports during my thesis.

Finally, I take this opportunity to express the profound gratitude to my beloved parents and my sister for their love and continuous support throughout the three years.

Chapter 1

# Introduction

## 1.1 Context

### 1.1.1 DHM and applications

As computer technologies have been advancing rapidly in recent decades, many new technologies and softwares have been developed which have caused revolutionary changes in many research domains and industries. Based on computer modeling and computation, lots of physical systems now can be simulated on computers via newly developed techniques. Using computer simulations, we can estimate, evaluate and predict the performance of interacting physical systems. For example, CAD (Computer-aided design) models of vehicles are integrated in fluid dynamics simulations for estimating and evaluating their performances in real usage (see Fig. 1.1).

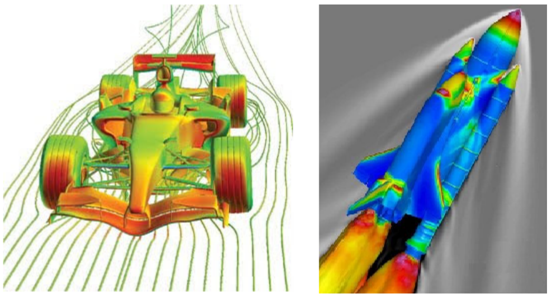


**Figure 1.1:** Examples of applications of computer simulation under software FLUENT® for testing performances of vehicles.

Human body modeling and simulations are more and more demanded in many human-centered or human-related domains, such as ergonomics, biomechanics, computer graphics, robotics, entertainment, etc. Accordingly, many **digital human models (DHM)** have been designed and applied in different domains (see Fig. 1.2). A DHM (also named as "virtual human", "avatar" or "humanoid"), which is a digital human-like system modeled using computer technologies, is able to represent human body's morphological characteristics such as dimensions, shapes or geometries and simulate human behaviors and competences such as postures, manipulations, motions, physical capabilities or even cognitive capabilities. Most

DHMs also integrate functions such as evaluation of the difficulty or discomfort of a simulated task.

**Figure 1.2:** Broadly used digital human models in many research and application domains. Photo sources (from left to right): RPx®, XDE®, Arboris®-Matlab, video game *FIFA 2011*®, film *Avatar*.

### 1.1.2 DHM motion simulation

Simulation of DHM's human-like motions is an essential problem for DHM research and it has a broad application range. On one hand, progresses of DHM researches in biomechanics, ergonomics, anthropometry and robotics can help us to simulate more and more realistic human motions in many domains, particularly computer graphics, movies, video games, etc; on the other hand, simulations of DHMs motions and manipulations can facilitate the studies on structures and behaviors of human beings in terms of biomechanics and ergonomics. In the research domains and industries where traditionally real human subjects are used, it is expected that DHMs can replace real human subjects for evaluating and predicting human performances. With respect to real human subjects, DHMs show advantages mainly in the following aspects:

- A DHM can interact with other virtual models in simulations; humans, on the contrary, can only interact with real objects. It is thus much easier and more economic to build environments for a DHM than for a real human being;
- There is no ethical problem in applications of DHMs in situations or environments that are dangerous for real human subjects, such as traffic accidents, intervention in radioactive areas, natural disasters, outer space, etc;
- The variabilities that exist in human body characteristics such as shape, dimension, gender or age require a large number of human subjects for carrying out a comprehensive analysis; the DHMs, however, can be easily modified to represent different human characteristics;
- In order to collect real data for further analysis with real human subjects, numerous sensors should be placed to measure and record the real motion data; the collected data should probably be further processed for analysis; using DHMs, the required data can be easily obtained, exported and stored.

Because of the above-mentioned merits, DHMs can increase significantly the efficiency and lower the cost.

Fig. 1.3 shows several examples of applications of DHMs in car design process. Instead of using human subjects for testing the models that must be manufactured beforehand, the DHMs can be integrated in virtual environments and interact directly with the CAD models of car body for testing ergonomics and safety of the car. Based on the test results, modifications and improvements could be carried out on the CAD model.

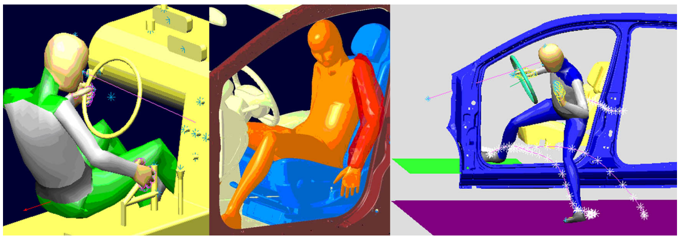


**Figure 1.3:** Several examples of DHM application in car design process (from website of European project **DHErgo**).

However, unlike real human subjects, a DHM itself does not possess the capabilities of thinking, learning, reflex and reaction, which brings difficulties to DHM motion simulation mainly in the following aspects:

1. Computing DHM motions that adapt to the situation or environment;
2. Realization of the generated motion.

In next section, we present recent techniques that deal with the above-mentioned problems.

### 1.1.3 Recent researches on DHM motion simulation

Current techniques for simulating DHM motions can generally be classified into kinematics-based methods and dynamics-based methods.

Based on a kinematic human model, a kinematics-based method realizes the DHM motion by computing a sequence of postures for the DHM under various constraints on its kinematics. Most of these methods rely on human motion data that is recorded with MoCap (Motion Capture) technique; accordingly a kinematic DHM can then realize very realistic motions by replaying the reference ones (see left figure in Fig. 1.4). These data-driven motions realized with help of the kinematics-based method show intrinsic realism even in very complex scenarii. However they suffer important limitations: they depend largely on MoCap data of real human motions, thus it is difficult to replay a motion if the environment or the dimensions of the human model differ significantly from those of the reference motion. Moreover, in kinematics-based methods, little physical factors (such as forces, masses or frictions) are taken into account in the interaction between the kinematic DHM and its environment. Consequently, the physical feasibility of the resulted motions is not ensured.

Dynamics-based methods for DHM motion simulation, originally developed and used in robotics, are paid more and more attention thanks to their better interaction with the environment. These methods rely on dynamic DHMs that take into account both kinematics and dynamics. Motions of the dynamic DHM are simulated based on knowledges on control laws and human ergonomics instead of recorded motion data. For a given scenario, firstly a global planner generates a global motion and specifies local tasks for dynamic controller to execute; then dynamic controllers, relying on control laws, execute these control tasks and generate muscle forces or joint torques to actuate the DHM for realizing its motion. Regarding the data-driven motions realized with kinematics-based method, the knowledge-driven motions with dynamic DHMs show less realism, in particular for complex scenarii and environments. However, the physical feasibility is ensured in these motions since DHM can interact with the environment (see right figure in Fig. 1.4). Another main drawback of dynamics-based methods lies in the high cost of computing time for motion planning because of the high dimension of the configuration space.

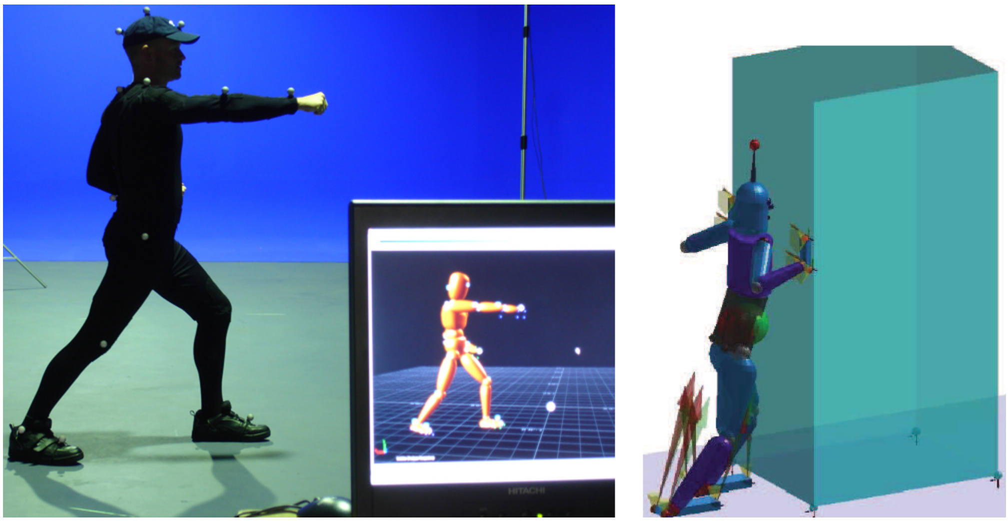


**Figure 1.4:** Example of DHM motion simulations. Left [1]: human motion simulation using motion capture technique; Right: dynamic simulation of a virtual human using dynamic controller (from Thesis of Cyrille COLLETTE [Col09]).

## 1.2 This study

### 1.2.1 Objective

This Ph.D thesis has been jointly supervised and funded by two laboratories: LSI (Laboratory of Interactive simulations) of the CEA (Atomic Energy and Alternative Energies Commission) and LBMC (Laboratory of Biomechanics and Impact Me-

---

[1] From website: http://byronbay.sae.edu/

chanics) of IFSTTAR (French Institute of Science and Technology for Transport, Development and Networks).

Specializing in biomechanics and ergonomics, LBMC focuses on ergonomic analysis and evaluation of human motions in different scenarii using primarily MoCap and kinematic motion adaptation techniques. LSI specializes in interactive simulations using virtual reality techniques. Dynamics-based methods are used in LSI for simulating virtual manikin's motions.

Responding to practical objectives of both laboratories, we intend to simulate DHM motions in complex scenarii by replacing as much as possible the experimental motion data with generic expertise on human movements. Moreover we want to take advantage of the capacity of the dynamics-based approaches to interact with a complex environment. In this background, this thesis aims to explore a dynamics-based approach for realizing DHM motion simulations in cluttered environments.

### 1.2.2 Scientific problematics

According to the context and the objective of this interdisciplinary study, we have summarized problematics in the following aspects: ergonomics of human motions in cluttered environments, dynamic balance and dynamics-based motion simulation.

#### Human motions in cluttered environments

Explicit knowledge on human motions in a cluttered environment is required for DHM motion planning. In this study, above all we need to obtain understanding on a series of human behaviors in cluttered environments such as posture evolutions, feet placements, hand manipulations, collision avoidance, etc. As a human moves in a cluttered environment, local environment surrounding the DHM varies all the time. Thus, knowledge on the influence of local environment on human behaviors is also required. Moreover, balance maintenance is an important aspect which should also been analyzed based on real human motions.

#### Dynamic balance

A DHM moving in cluttered environment should maintain balance all the time which brings dynamic constraints on its motion. Based on the existing literature, we need to formulate a balance criterion for both verifying and quantifying the balance of DHM moving in cluttered environments via various kinds of interactions.

#### Motion simulation

We intend to simulate knowledge-driven dynamic motions of DHMs moving in cluttered environments. DHM motions in cluttered environments are usually dynamic motions with multiple transitions between stances. Constraints on the motion vary since the local environment and support configuration of the DHM changes during its motion. Thus, we aim at carrying out locally motion planning in order to increase the efficiency: motion planner generates piecewisely for each transition the whole-body motion relying on a priori knowledge on human motion ergonomics and balance maintenance. Then, motion should be realized by using dynamic controllers.

## 1.3 Thesis plan

The thesis is organized as follows:

Chapter 1 presents the context of the study, including the DHMs and their applications, recent achievements in researches for DHM motion simulations, as well the objectives and problematics of the study.

Chapter 2 presents the basic knowledge and mathematical notations that are involved in the DHM kinematics. Some mathematical tools used in this thesis such as automatic differentiation, polytopes and B-splines are presented in this chapter.

In Chapter 3, we present the observation and analysis work carried out for real human motions in cluttered environments. We have conducted MoCap experiments in which a series of human motions in cluttered environments have been recorded. By reconstructing the recorded motions using a kinematical DHM, we obtain the motion data including the whole-body postures and some estimated trajectories of end-effectors and center of mass. We then observe and analyze the motions in order to get knowledge on motion principles in cluttered environments. The generic knowledge obtained in this chapter helps us target the key problems in our work.

In Chapter 4, we study the balance problem of DHM during motions. This chapter begins with a state of the art on the topic of balance. Based on a simplified mass point model, a balance criterion is formulated by combining several newly proposed balance criteria in robotics. This criterion is then validated in several scenarii via comparison with some traditional criteria. Based on this criterion, a stability margin evaluates the quality of balance or the capability of resisting disturbances. By extending the simplified model to a whole-body human model, an improved criterion has been proposed in which the whole-body postural angular momentum is taken into account. To verify the correctness of the formulated criteria, balance analysis is carried out on several recorded motions. Relying on these analysis results, we made hypotheses and chose the balance criteria for the following.

A mini chapter gives the overview of a hierarchical framework explored in this study for realizing DHM motions in cluttered environments. This framework consists of three levels which are presented in Chapter 5 and Chapter 6.

In Chapter 5 we present the methodologies at global level of the framework. A state of the art on generation of reference trajectory for multi-step motion of legged robots is firstly presented. Two methods are explored for computing a global CoM trajectory for the DHM during its motion. In both methods, a pre-defined sequence of support configurations provides the balance constraints on the CoM trajectory which is parameterized as a 3D B-spline. The global trajectory then is computed using optimization techniques. The computed CoM trajectory is then used at the following two levels for ensuring the balance during the motion.

In Chapter 6 we present the local planning level and the motion execution level of the framework. This chapter begins with a state of the art on motion planning, collision detection and motion control. The global CoM trajectory obtained at the first level supplies the time durations of transition phases between support configurations. Whole-body motion planning then is carried out locally for each transition phase and accordingly we obtain whole-body postures and end-effector trajectories in each phase. Whole-body collision-freeness, inverse kinematics and balance constraints are all taken into account at this level. At the motion execution level, the results of the previous levels are taken as control references and the generated motion is executed on a dynamic DHM by virtue of dynamic controllers.

The implementation of the framework in a complex car-ingress scenario is presented in Chapter 7. Firstly we analyze a recorded car-ingress motion in order to obtain the information on the environment, the support configurations, the initial and final motion conditions, etc. Then the framework is applied for generating and simulating the car-ingress motion of a DHM. We present in detail the applications of our methodologies at the three levels and show their results. We then evaluate the final motion generated and simulated by our framework by comparing it with the recorded one.

Chapter 8 concludes the entire thesis and draws some perspectives.

## CHAPTER 2

# Basic representations

This chapter aims at presenting some preliminary knowledge, mathematic notations and useful tools used in this thesis. We firstly present the virtual human model used in this study and some mathematic notations and equations on its kinematics. Then we present the vector function derivatives which are very useful in robotics, for example, for computing Jacobian matrix and function gradients. Finally, we present some geometric tools such as polytopes and B-splines. All the representations in this chapter will be frequently used in the following chapters.

## 2.1 Digital human model

### 2.1.1 Representations

**Figure 2.1:** The DHM used in this thesis (shown in interface of software XDE®).

| Body Joints | 45 DoFs |
|---|---|
| Pelvis (free joint) | 6 |
| Lumbar | 3 |
| Torso | 3 |
| Neck | 3 |
| Left/Right Shoulder | 3 (×2) |
| Left/Right Elbow | 2 (×2) |
| Left/Right Hand | 2 (×2) |
| Left/Right Thigh | 3 (×2) |
| Left/Right Knee | 2 (×2) |
| Left/Right Ankle | 2 (×2) |
| Left/Right Toe | 1 (×2) |

**Table 2.1:** A summary of the joints and DoFs of the DHM.

The DHM used in this thesis (see Fig. 2.1) consists of 19 body segments (see Appendix B) and 45 DoFs (degrees of freedom). Its pelvis is defined as the root segment that is supposed to be linked by a free-joint (6 DoFs) with the environment. Five segment chains for each end-effector (feet, hands and head) rooted at the pelvis are modeled by linking segments with only revolute joints (i.e. hinge joints). A summary of the joints of the DHM and their associated DoFs are presented in Table 2.1.

### 2.1.2 Kinematics

#### Position and orientation

The environment where the DHM realizes its motion is associated with a global frame ("space frame" or "world frame" or "absolute frame"). The global frame is supposed to be static: one point is fixed if it does not change its position with respect to this frame. To describe the position and orientation of a body segment of the DHM, a local frame ("body frame") is attached to each body segment. The position of one body segment of the DHM is represented by the global Cartesian coordinates of its local frame's origin:

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.1}$$

and its orientation is represented by its Euler angles:

$$\boldsymbol{\Theta} = \begin{bmatrix} \psi \\ \theta \\ \varphi \end{bmatrix} \tag{2.2}$$

or a quaternion:

$$\bar{\mathbf{q}} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{2.3}$$

or a rotation matrix:

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \in SO(3) \tag{2.4}$$

In fact, the three representations in Equations (2.2)-(2.4) are almost equivalent: they describe the rotation between the three axis of the local frame regarding the global frame. The rotation matrix in Equation (2.4) is also broadly used to describe an operation of rotation. After the rotation $\mathbf{R}$, the point at position $\mathbf{p}$ is moved to the position $\mathbf{p}'$:

$$\mathbf{p}' = \mathbf{R}\mathbf{p} \tag{2.5}$$

The position and orientation of a body segment can be represented by a homogeneous matrix:

$$\mathbf{H} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0}_{3,1} & 1 \end{bmatrix} \in SE(3) \tag{2.6}$$

This matrix is used to represent the position and orientation of a body segment of the DHM; it also describes the transformation between the body frame and the global frame. Using this homogeneous matrix, we can transform the description between different frames. For example, a point $P$ has its position at ${}^{b}\mathbf{p}_0$ in the body frame (${}^{b}*$ and ${}^{s}*$ mean respectively expressions in body frame and in the global frame). We can compute its position in the global frame as:

$$\begin{bmatrix} {}^{s}\mathbf{p}_0 \\ 0 \end{bmatrix} = \mathbf{H} \begin{bmatrix} {}^{b}\mathbf{p}_0 \\ 0 \end{bmatrix} \tag{2.7}$$

### Representations of a mobile frame

A mobile body segment has its velocity twist:

$$\begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \\ v_x \\ v_y \\ v_z \end{bmatrix} \tag{2.8}$$

An adjoint matrix is a transformation matrix for a twist associated with a transformation $\mathbf{T}$ between two frames:

$$\mathbf{Ad}_T = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \hat{\mathbf{p}}\mathbf{R} & \mathbf{R} \end{bmatrix} \tag{2.9}$$

With this matrix we have:

$$\begin{bmatrix} {}^{s}\boldsymbol{\omega} \\ {}^{s}\mathbf{v} \end{bmatrix} = \mathbf{Ad}_T \begin{bmatrix} {}^{b}\omega \\ {}^{b}\mathbf{v} \end{bmatrix} \tag{2.10}$$

For a vector $\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$, its skew-symmetric matrix is defined as:

$$\hat{\mathbf{z}} = \begin{bmatrix} 0 & -z_3 & z_2 \\ z_3 & 0 & -z_1 \\ -z_2 & z_1 & 0 \end{bmatrix} \tag{2.11}$$

then the cross product of z and any $\mathbf{x} \in \mathbb{R}^3$ can be calculated by:

$$\mathbf{z} \wedge \mathbf{x} = \hat{\mathbf{z}}\ \mathbf{x} \tag{2.12}$$

The relation between the angular velocity and a rotation (expressed in global frame) is:

$$\hat{\boldsymbol{\omega}} = \dot{\mathbf{R}}\mathbf{R}^T \tag{2.13}$$

The solution of Equation (2.13) is:

$$\mathbf{R} = e^{\hat{\boldsymbol{\omega}}t} = I + \hat{\mathbf{n}}sin(\|\boldsymbol{\omega}\|\, t) + \hat{\mathbf{n}}^2(1 - cos(\|\boldsymbol{\omega}\|\, t)) \tag{2.14}$$

where

- $\boldsymbol{\omega}$ is a constant vector indicating the velocity of rotation;
- $\mathbf{n}$ is a unit vector indicating the axis of rotation.

Equation (2.14) is called the *Rodrigures' rotation formula*; it gives the rotate matrix for a rotation about the axis $\mathbf{n}$ that has a constant angular velocity. This equation can be used to interpolate between two orientations:

$$\mathbf{R}(t + dt) = \mathbf{R}(t)e^{\hat{\boldsymbol{\omega}}dt} \tag{2.15}$$

### DHM configuration description

Generally, the configuration of a DHM is represented as:

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{n_{dof}} \end{bmatrix} \tag{2.16}$$

The first six elements in $\mathbf{q}$ represent the root body's position and orientation with respect to the global frame:

$$\mathbf{q}(1:6) = \begin{bmatrix} \mathbf{p}_{root} \\ \boldsymbol{\Theta}_{root} \end{bmatrix} = \begin{bmatrix} x_r \\ y_r \\ z_r \\ \psi_r \\ \theta_r \\ \varphi_r \end{bmatrix} \tag{2.17}$$

## 2.2 Derivatives of vector functions

Vector functions are frequently involved in this study. Gradient of vector functions are required in multiple cases. Thus we present here some basic knowledge on derivatives of vector functions.

### 2.2.1 Vector function and its derivatives

Let $\mathbf{x}$ and $\mathbf{y}$ be column vectors of orders $n$ and $m$ respectively:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \tag{2.18}$$

We suppose that each component of the vector $\mathbf{y}$ is a function of $\mathbf{x}$, then the derivative of vector function $\mathbf{y} = \mathbf{y}(\mathbf{x})$ with respect to $\mathbf{x}$ is defined as:

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} =: \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_2}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_1} \\ \frac{\partial y_1}{\partial x_2} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_m}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_1}{\partial x_n} & \frac{\partial y_2}{\partial x_n} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix} \tag{2.19}$$

which is a $n \times m$ matrix.

Let $\mathbf{f}(\mathbf{y})$ be a column vector of order $l$ which is a function of $\mathbf{y}$, then its derivative with respect to $\mathbf{x}$ is a $n \times l$ matrix that can be expressed as:

$$\frac{\partial \mathbf{f}(\mathbf{y})}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \frac{\partial \mathbf{f}(\mathbf{y})}{\partial \mathbf{y}} \tag{2.20}$$

where $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$ is obtained in Equation (2.19).

Equation (2.20) reveals the chain rule for computing derivatives of vector functions.

### 2.2.2 Automatic differentiation

Based on the application of the chain rule in vector calculus, techniques of Automatic Differentiation (AD, also referred to as *Algorithmic Differentiation* or *Computational Differentiation*) have been proposed for computing derivatives of functions in form of computer programs. By applying repeatedly the chain rules on the elementary arithmetic operations and elementary functions in a computer program, derivatives of the programmed function can be obtained in a numerical way. The AD techniques show higher computational efficiency regarding Symbolic Differentiation and Numerical Differentiation.

Some of the existing AD tools include Adiff (Matlab), ADMAT / ADMIT (Matlab), ADMB (C++), etc.

## 2.3 Polytopes

In elementary geometry, a polytope ([GS67], [Grü03]) is a geometric object with facets, edges and vertices, which exists in any general number of dimensions. In 1D space, a polytope is a line segment; In a 2D space, a polytope is a convex polygon; In 3D space, a polytope is a convex polyhedron. In mathematics, a polytope can be represented by a point set in a space of any dimensions ([ABCO$^+$01]).

### 2.3.1 Representations of a polytope

There are two main representations for a polytope ([Fuk03]). It can be expressed in either the *vertex representation* (V-representation):

$$\begin{aligned}\mathbf{X} &= conv(\mathbf{V} = \{\mathbf{v}_1 \in \mathbb{R}^n, \;\cdots\;, \;\mathbf{v}_{N_v} \in \mathbb{R}^n\}) \\ &= \left\{ \mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} = \sum_{i=1}^{N_v} \alpha_i \mathbf{v}_i, \; \alpha_i \geq 0, \sum_{i=1}^{N_v} \alpha_i = 1 \right\}\end{aligned} \tag{2.21}$$

or *half-space representation* (H-representation):

$$\mathbf{X} = \{\, \mathbf{x} \in \mathbb{R}^n \mid \; \mathbf{A}\mathbf{x} \leq \mathbf{b} \; with \; \mathbf{A} \in \mathbb{R}^{m \times n}, \; \mathbf{b} \in \mathbb{R}^m \} \tag{2.22}$$

### 2.3.2 Projection and intersection of polytopes

Given a set $\mathbf{Q} \in \mathbb{R}^{n_Q}$ and a set $\mathbf{P} \in \mathbb{R}^{n_P}$with $n_Q \leq n_P < \infty$, the projection of P onto Q is defined as:

$$proj_Q(\mathbf{P}) := \{\, \mathbf{q} \in \mathbf{Q} \mid \; \exists \mathbf{p} \in \mathbf{P} \; with \; \mathbf{q} = \mathbf{M}_p \mathbf{p} + \mathbf{m}_0 \} \tag{2.23}$$

for some given $\mathbf{M}_p \in \mathbb{R}^{n_Q \times n_P}$ and $\mathbf{m}_0 \in \mathbb{R}^{n_Q}$.

**Theorem 1.** *If $\mathbf{P} \in \mathbb{R}^d \times \mathbb{R}^k$ is a polytope, then the projection of $\mathbf{P}$ onto $\mathbb{R}^d$ is a polytope ([JKMoCED04]).*

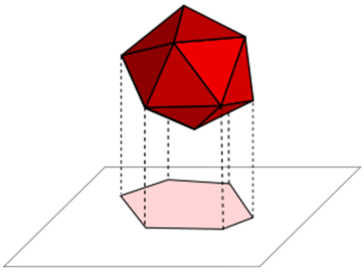


**Figure 2.2:** Illustrating the projection of a 3D polytope onto a 2D surface.

If $\mathbf{P_1} \in \mathbb{R}^k$ and $\mathbf{P_2} \in \mathbb{R}^k$ are two polytopes that intersect with each other, then their intersection (common subspace) $\mathbf{P}_{1\cap 2} = \mathbf{P_1} \cap \mathbf{P_2}$ is also a polytope whose dimension is equal to or smaller than $k$ (see Fig. 2.3).

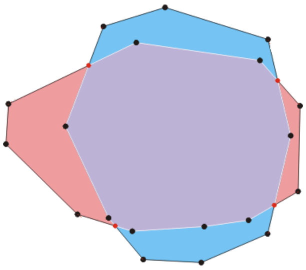


**Figure 2.3:** The intersection of two 2D polytopes (respectively in red and in blue) is also a 2D polytope (in purple).

### 2.3.3 Some hyper-spheres related with a polytope

#### Chebyshev Ball

A "*Chebyshev Ball*" is defined as the minimal radius ball enclosing the entire bounded point set $\mathbf{X}$. The centre of this sphere is the *Chebyshev Centre*. The Chebyshev Ball can be computed by the optimization:

$$\underset{\mathbf{c},r}{min}\{r : \|\mathbf{c} - \mathbf{x}\| \leq r, \forall \mathbf{x} \in \mathbf{X}\} \tag{2.24}$$

whose solution $\mathbf{c}$ and $r$ represent respectively the center and the radius of the Chebyshev ball.

#### Largest Insphere

An "*Insphere*" is defined as a sphere inscribed in a given polytope. Accordingly, a "*Ball Center*" is defined as a point inside the polytope that is the center of the largest Insphere of that polytope. The largest insphere of a given polytope is computed by:

$$\underset{\mathbf{c},r}{max}\,\{r:\ \mathbf{Ax}\leq\mathbf{b},\forall\mathbf{x}\ that\ satisfies\ \|\mathbf{c}-\mathbf{x}\|\leq r\} \tag{2.25}$$

whose solutions **c** and $r$ represent respectively the Ball Center and the radius of this insphere.

The above-mentioned Chebyshev Ball and largest Insphere are important concepts for evaluating approximately the volume of a polytope.

### Residual Ball

For a given point $\mathbf{c}_0$ inside the polytope, the largest hyper-sphere centered at $\mathbf{c}_0$ and included inside the polytope is computed by:

$$\underset{r}{max}\,\{r:\ \mathbf{Ax}\leq\mathbf{b},\forall\mathbf{x}\ that\ satisfies\ \|\mathbf{c_0}-\mathbf{x}\|\leq r\} \tag{2.26}$$

whose solution $r$ is called the *Residual Ball Radius.* The algorithm for computing the Residual Ball Radius is shown in Algorithm 1. The Residual Ball Radius plays an important role for evaluating distance margins for a point inside a given polytope.

**Algorithm 1 r = ResidualRedius**($\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}_0$)

**Input:** H-representation of a polytope **A** and **b**, and a point $\mathbf{c_0}$ inside the polytope

**Output:** The residual ball redius $r$

```
if $\mathbf{A}^* \mathbf{c}_0 > \mathbf{b}$ then
  return "Error"
end if
$m = \mathbf{A}.rows$
$\mathbf{A}_n = \mathbf{A}$, $\mathbf{b}_n = \mathbf{b}$        // Initialize the normalized matrix and vector
for $i := 1$ to $m$ do
  $\mathbf{A}_n$[i, :]= $\mathbf{A}$[i, :]/$norm$($\mathbf{A}$[i, :])
  $\mathbf{b}_n$[i]= $\mathbf{b}$[i]/$norm$($\mathbf{A}$[i, :])
end for
$r = \min(\mathbf{b}_n - \mathbf{A}_n * \mathbf{c}_0)$
return $r$
```

## 2.4 B-splines

B-spline is a parametric curve frequently used in computer graphics, CAD and other related fields. It is defined by its control points and knots partition. The advantage of a B-spline is the possibility to manipulate the local curve properties as well as its derivatives by configuring its control points ([Boe80], [EM96]). Thus, in the field of robotics, many researchers use B-splines to represent motion paths or trajectories ([Bob88], [OL96], [QEMR11]). In this section we presented the mathematic knowledge about B-splines.

### 2.4.1 Representation

A B-spline is associated with a knot vector: $\mathbf{T} = \{t_0,\ t_1,\ \cdots\ t_i,\ \cdots,\ t_m\}$ containing $m+1$ knots that have the relation:

$$t_0 \leq t_1 \cdots \leq t_i \leq \cdots \leq t_m \tag{2.27}$$

A B-spline of degree $k$ is a parametric curve:

$$p\left(t\right) = \sum_{i=0}^{n} N_{i,\,k}(t) a_i,\ t \in [0,\ 1] \tag{2.28}$$

where

- $N_{i,\,k}(t)$: is the $i$-th basis function of degree $k$ ;
- $a_i$: is the $i$-th control point (or *de Boor* points).

The B-spline in Equation (2.28) has $n+1$ control points. The 3 integers $m$, $n$ and $k$ have the following relation:

$$m = k + n + 1 \tag{2.29}$$

We can calculate the basis functions using the **Cox-de Boor** recursion formula:

$$\begin{cases} N_{j,\,0}(t) := \begin{cases} 1 & if\ t_j \leq t \leq t_{j+1} \\ 0 & otherwise \end{cases} & ,\ j = 0, \cdots, m-1 \\ \\ N_{j,\,k}(t) := \frac{t - t_j}{t_{j+k} - t_j} N_{j,\,k-1}(t) + \frac{t_{j+k+1} - t}{t_{j+k+1} - t_{j+1}} N_{j+1,\,k-1}(t),\ j = 0, \cdots, m-k-1 \end{cases} \tag{2.30}$$

If the end conditions (positions or derivations) are imposed, the B-Splines are chosen to be **clamped**, i.e. they go through their initial and final control points. This requires a special configuration of the knot vector whose multiplicity of the first (also the last) knot is $k+1$:

$$\begin{cases} t_i = 0,\ i \in [0, k] \\ t_{i-1} \leq t_i \leq t_{i+1},\ i \in [k+1, n] \\ t_i = 1,\ i \in [n+1, n+k+1] \end{cases} \tag{2.31}$$

In Algorithm 2, we show the algorithm for computing the basis functions for a given parameter configuration of B-spline.

**Algorithm 2 N = calcNs**($u$, $m$, $n$, $k$, **knots**)

**Input:** $n$, $k$, $m$, $u$ and **knots** $\{t_0, \ldots, t_m\}$

**Output:** Basis $N_{0,k}(u), N_{1,k}(u), \ldots, N_{n,k}(u)$ respectively expressed by $N[0], N[1], \ldots, N[n]$

```
if $n + k + 1 \neq m$ then
        show error and return
end if
Initialize $N[0, \ldots, n]$ to 0
if $u = t_0$ then
        $N[0] = 1$
        return
else
        if $u = t_m$ then
                $N[n] = 1$
                return
        end if
end if
Let $u$ be in kont span $[t_j, t_{j+1})$
$N[j] = 1$
for $d := 1$ to $k$ do
        $N[j-d] = \frac{t_{j+1}-u}{t_{j+1}-t_{(j-d)+1}} * N[j-d+1]$
        for $i := j-d+1$ to $k$ do
                $N[i] := \frac{u-t_i}{t_{i+d}-t_i} * N[i] + \frac{t_{i+d+1}-u}{t_{i+d+1}-t_{i+1}} * N[i+1]$
                $N[j] = \frac{u-t_j}{t_{j+d}-t_j} * N[j]$
        end for
end for
```

### 2.4.2 Derivatives of B-splines

The $d$-th derivative of a B-Spline of degree $k$ is another B-spline of degree $k$-$d$ whose knot vector shares the same internal knots with original one. It has $n$+1-$d$ control points that can be computed from the original ones as:

$$a_i^d = \frac{k-d+1}{t_{i+k+1}-t_{i+d}}\left(a_{i+1}^{d-1}-a_i^{d-1}\right) \text{ for } i \in [0,\ \cdots,\ n-d] \tag{2.32}$$

If the denominator in Equation (2.32) is 0, $a_i^d$ is set to be 0.

If the B-spline is clamped, then by dropping the first and the last knot from the original knot vector, we get the new knot vector and the multiplicity of the first (also the last) knot becomes $k$. Then the 1st order derivative of the B-spline can be expressed as:

$$p'(t) = \sum_{i=0}^{n-1} N_{i,\ k-1}(t)a_i^1 \tag{2.33}$$

In this case, the derivative curve has $n$ control points, $m$-1 knots and is in ($k$-1)-th degree. We can verify their relation with Equation (2.29):

$$(m-1) = (k-1)+n+1$$

Moreover the $d$-th derivative curve has $n$+1-$d$ control points, $m$-2$d$ knots and the degree $k$-$d$.

$$p^d(t) = \sum_{i=0}^{n-d} N_{i,\ k-d}(t)a_i^d \tag{2.34}$$

Let's denote :

$$\mathbf{a} = \left[\begin{array}{ccc} a_0, & \cdots, & a_n \end{array}\right]^t \tag{2.35}$$

the vector of the control points of a B-spline and

$$\mathbf{a}^1 = \left[\begin{array}{ccc} a_0^1, & \cdots, & a_{n-1}^1 \end{array}\right]^t \tag{2.36}$$

the vector of control points for the 1st order derivative of the B-spline.

Then using Equation (2.32) we can get:

$$\mathbf{a}^1 = \begin{bmatrix} \frac{k}{t_{k+1}-t_1} & 0 & \cdots & 0 \\ 0 & & \ddots & \vdots \\ \vdots & & & 0 \\ 0 & \cdots & 0 & \frac{k}{t_{k+n}-t_n} \end{bmatrix} \begin{pmatrix} a_1-a_0 \\ \vdots \\ a_n-a_{n-1} \end{pmatrix} \tag{2.37}$$

Let $\mathbf{L}_{n1} = [\mathbf{0}_{n\times 1},\ \mathbf{I}_{n\times n}]$ and $\mathbf{L}_{n2} = [\mathbf{I}_{n\times n},\ \mathbf{0}_{n\times 1}]$, then we have:

$$\left(\mathbf{L}_{n1}-\mathbf{L}_{n2}\right)\mathbf{a} = \begin{bmatrix} a_1-a_0 \\ \vdots \\ a_n-a_{n-1} \end{bmatrix} \tag{2.38}$$

So Equation (2.37) can be rewritten as:

$$\mathbf{a}^1 = \begin{bmatrix} \frac{k}{t_{k+1}-t_1} & 0 & \cdots & 0 \\ 0 & & \ddots & \vdots \\ \vdots & & & 0 \\ 0 & \cdots & 0 & \frac{k}{t_{k+n}-t_n} \end{bmatrix} (\mathbf{L}_{n1} - \mathbf{L}_{n2})\,\mathbf{a} \tag{2.39}$$

Since the knots vector is given and the matrix and are constant, the control points vector of the 1st order derivative of a B-spline can be computed by:

$$\mathbf{a}^1 = \mathbf{T}_{10}\,\mathbf{a} \tag{2.40}$$

where :

$$\mathbf{T}_{10} = \begin{bmatrix} \frac{k}{t_{k+1}-t_1} & 0 & \cdots & 0 \\ 0 & & \ddots & \vdots \\ \vdots & & & 0 \\ 0 & \cdots & 0 & \frac{k}{t_{k+n}-t_n} \end{bmatrix} (\mathbf{L}_{n1} - \mathbf{L}_{n2}) \tag{2.41}$$

In the same way, we can compute the control points vector of the 2nd order derivative by:

$$\mathbf{a}^2 = \mathbf{T}_{21}\,\mathbf{a}^1 \tag{2.42}$$

where $\mathbf{T}_{21}$is a constant matrix.

With the Equation (2.40), Equation (2.42) can be continued as:

$$\mathbf{a}^2 = \mathbf{T}_{21}\,\mathbf{T}_{10}\;\mathbf{a} = \mathbf{T}_{20}\,\mathbf{a} \tag{2.43}$$

Thus we can make a conclusion for control points of derivative splines: the control points vector of the $d$-th derivative of a B-spline can be computed from the origin control points vector by a constant matrix $\mathbf{T}_{d0}$:

$$\mathbf{a}^d = \mathbf{T}_{d0}\,\mathbf{a} \tag{2.44}$$

### 2.4.3 End conditions

Suppose $p_0$ is the initial position of the clamped B-spline. It passes through its first control point.If this B-spline has its degree $k > 2$, its 1st-order and 2nd-order derivative curves are also clamped B-splines.They pass through their first control points. Suppose $p'_0$ and $p''_0$ are respectively the initial values of the 1st-order and 2nd-order derivative curves, then Equation (2.44) can help us to compute the first three control points for the B-spline in function of the initial conditions:

$$a_0 = p_0 \tag{2.45}$$

$$a_1 = \frac{t_{k+1}}{k} {p'}_0 + a_0 \tag{2.46}$$

$$a_2 = \frac{t_{k+1} t_{k+2}}{(k-1)k} {p''}_0 + \left(1 + \frac{t_{k+2}}{t_{k+1}}\right) a_1 - \frac{t_{k+2}}{t_{k+1}} a_0 \tag{2.47}$$

In the same way, the final three control points can be expressed in function of the end conditions:

$$a_n = p_f \tag{2.48}$$

$$a_{n-1} = a_n - \frac{t_{n+k-1} - t_{n-1}}{k} {p'}_f \tag{2.49}$$

$$a_{n-2} = \frac{(1-t_n)(1-t_{n-1})}{k(k-1)} {p''}_f + \left(1 + \frac{1-t_{n-1}}{1-t_n}\right) a_{n-1} - \frac{1-t_{n-1}}{1-t_n} a_n \tag{2.50}$$

where $p_f$, ${p'}_f$ and ${p''}_f$ are respectively final values of the B-spline, its 1st-ordre and 2nd-ordre derivative curves.

### 2.4.4 Discretization

If we discretize the time span of the B-Spline into $l+1$ sampling instants $u_j (0 \leq j \leq l)$, we can use the Cox-de Boor recursion formula to calculate the basis functions for each instant. So the position at the instant is:

$$p\left(u_j\right) = \sum_{i=0}^{n} N_{i,\ k}(u_j) a_i \tag{2.51}$$

Then we can define a matrix $\mathbf{A}_k$ that contains basis function values for all the instants.

$$\mathbf{A}_k = \begin{bmatrix} N_{0,\ k}(u_0) & \cdots & N_{i,\ k}(u_0) & \cdots & N_{n,\ k}(u_0) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ N_{0,\ k}(u_j) & \cdots & N_{i,\ k}(u_j) & \cdots & N_{n,\ k}(u_j) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ N_{0,\ k}(u_l) & \ldots & N_{i,\ k}(u_l) & \cdots & N_{n,\ k}(u_l) \end{bmatrix} \tag{2.52}$$

Then for a certain instant uj, the values of the spline is obtained by:

$$p\left(u_j\right) = \sum_{i=0}^{n} \mathbf{A}_k(j+1,\ i+1) a_i \tag{2.53}$$

The derivative of the B-Spline:

$$p^d\left(u_j\right) = \sum_{i=0}^{n-d} \mathbf{A}_{k-d}(j+1,\ i+1) a_i^d \tag{2.54}$$

with the basis functions matrix:

$$\mathbf{A}_{k-d} = \begin{bmatrix} N_{0,\ k-d}(u_0) & \cdots & N_{i,\ k-d}(u_0) & \cdots & N_{n-d,\ k-d}(u_0) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ N_{0,\ k-d}(u_j) & \cdots & N_{i,\ k-d}(u_j) & \cdots & N_{n-d,\ k-d}(u_j) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ N_{0,\ k-d}(u_l) & \ldots & N_{i,\ k-d}(u_l) & \cdots & N_{n-d,\ k-d}(u_l) \end{bmatrix} \tag{2.55}$$

## CHAPTER 3

# Motions in cluttered environments

In order to get a generic understanding of human motions in cluttered environments, we start this study with some experimental work. In the experiments that we have carried out, motions of several volunteer subjects in a series of designed environments have been recorded. From these experimental data, we aim to obtain generic characteristics of human motions in cluttered environments and the implicit principles involved in these motions. The heuristics of this experimental work, together with some more comprehensive ergonomic knowledge yielded in existing researches in LBMC, can help us determine important problematics and choose reasonable hypotheses necessary for the following work of this study.

In this chapter, Section 3.1 presents briefly the Motion Capture (MoCap) experiments including the experimental materials, the experiment procedures, data processing and motion reconstruction; we present in Section 3.2 the analysis work that has been conducted using the database of the MoCap experiments; Section 3.3 concludes this chapter with some remarks and perspectives.

## 3.1 Experiment introduction

We have conducted a series of MoCap experiments for human motions in a "passage under a beam" scenario. The experiments have been carried out in the MoCap room of the laboratory LBMC. 3 young male volunteer subjects have been selected and 30 motions have been recorded for each subject.

### 3.1.1 Equipments

The MoCap room in LBMC is equipped with 10 Vicon® infra-red cameras (sampling frequency: 100Hz), retro-reflective markers, a 64-channel A/D convertor, a computer controlling all the devices (see Fig. 3.1) and the Vicon® Nexus software.

### 3.1.2 Materials and motion strategies

For each subject, we have conducted 30 experiments with 5 different environment configurations and 3 different movement strategies (see Table 3.1).

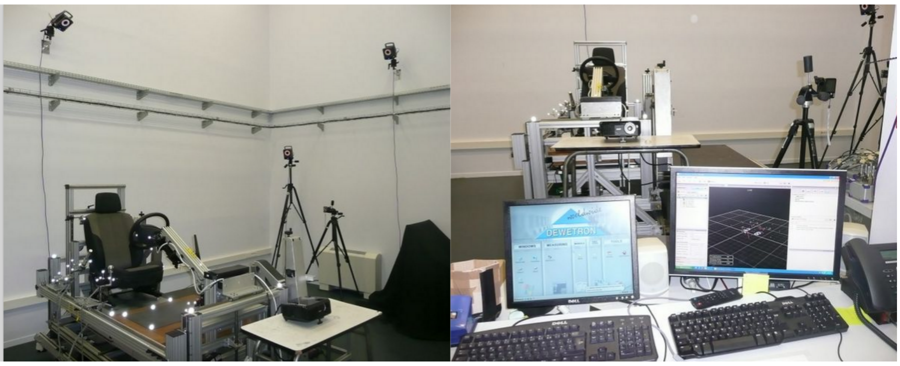


**Figure 3.1:** Mocap rooms with experimental devices - Vicon® cameras and PC (Source: website of LBMC).

The material configurations in our experiments (see Fig. 3.2) include:

- $C1$ : a beam with three different heights ($H1$=1.45$m$, $H2$=1.30$m$ and $H3$ =1.10$m$);
- $C2$ : a 1.30$m$ high beam with a lower obstacle (a second parallel beam at 0.50$m$ high);
- $C3$ : two parallel beams (identical to $C2$) with a seat.

Altogether 5 configurations are defined namely $H1C1$, $H2C1$, $H3C1$, $H2C2$ and $H2C3$.

Before starting his motion, the subject stands naturally with two feet at specified positions. He is required to pass under the beam or between the beams and stops the motion with his feet in a predefined zone (or sitting on the seat for the configuration $C3$), while respecting the movement strategy. Three movement strategies were defined in our experiments:

- $L$ : without any requirements on footsteps or hand grasps, a subject can move freely to accomplish his motion;
- $S1$ : a subject passes with a pre-defined footstep order and without hand grasp;
- $S2$ : a subject passes with a pre-defined footstep order using a left hand grasp.

Two repetitions are performed in each situation, leading to 30 motions per subjects.

### 3.1.3 Experiment procedure

We have carried out the experiments using the optical motion capture system Vicon®MX T40, recording the position of reflective markers via 10 monochrome cameras with a sampling frequency of 100Hz. Altogether 50 markers are placed

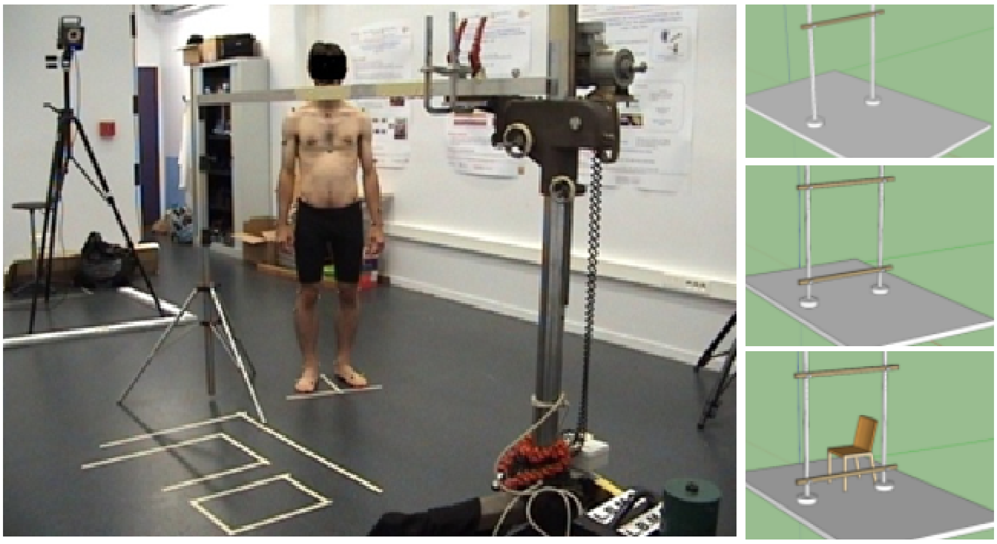


**Figure 3.2:** Left: Subject and experimental materials; Right: scheme of the three material configurations for experiments.

| Subject ID | Configurations | | Strategies | Repetition |
|---|---|---|---|---|
| 01_TR<br>02_JN<br>03_JC | C1 | H1 | L<br>S1<br>S2 | 2 |
| | | H2 | | |
| | | H3 | | |
| | C2 | H2 | | |
| | C3 | H2 | | |
| Motion name example: *01_TR_H1C1S1_2* | | | | |
| Total: 90 motions | | | | |

**Table 3.1:** Summary of MoCap experiments.

on the body of a subject (see Appendix C). As a preliminary step, the experiment operator collects the main anthropometrical dimensions for each subject at the beginning of the experiments. These dimensions will be used to facilitate the adjustment of the virtual human later used in the motion reconstruction procedure.

Once the markers are attached on a subject's body, a *calibration* procedure is then carried out. During this procedure, the subject stands statically with a specified posture, and 3 photographs (face, left profile and back) and simultaneous motion capture are carried out (see Fig. 3.3).

During the motion data recording procedure, the experiment operator gives instructions to the subject about the motion strategy. Before each motion recording, the subject practices for several times his motion to get familiar with the environmental configuration. Once a motion is recorded, the experiment operator verifies the result, save the data and prepare for the next motion.

The motion capture experiment takes altogether about 3 hours for each subject.

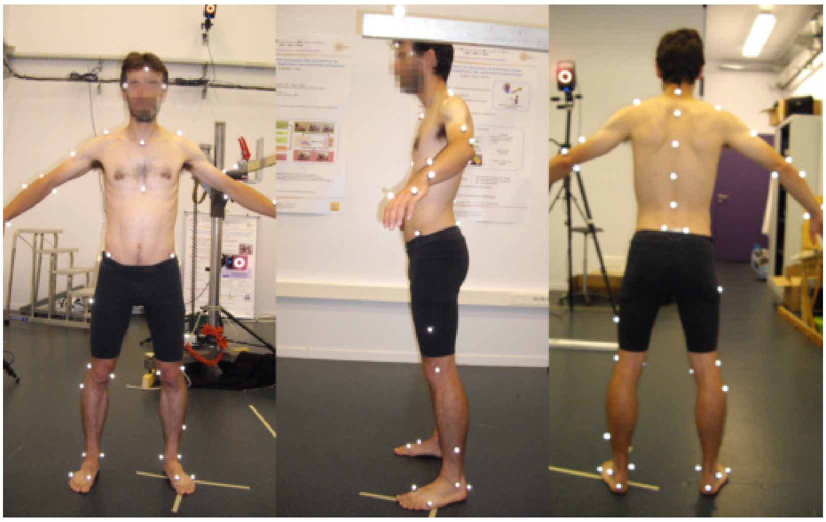
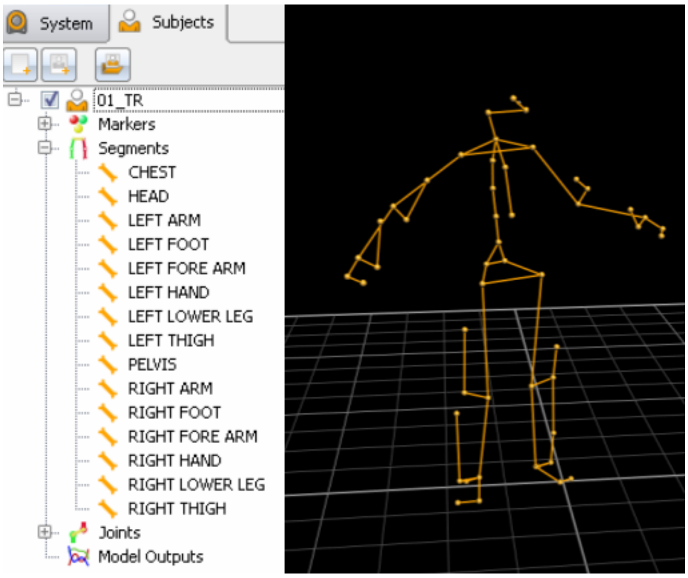


**Figure 3.3:** Calibration procedure for positioning markers regarding the subject's body. The photographs are taken in respectively face, left profile and back directions.

### 3.1.4 Experimental data processing

When all the MoCap experiments are finished, we then have the raw experimental data of the motions, namely the sampled 3D positions of markers attached on the subject's body during the motions.

The first step for processing the recorded data is to labelize the markers in the software Vicon® Nexus (see Fig. 3.4). Using the data recorded in the calibration procedure, we can easily distinguish the markers attached to the subject's body from each other. By naming and linking the markers, we can define a kinematical model of the subject as well as its segments and joints (see Fig. 3.4). With help of this model, Vicon® Nexus automatically distinguishes for each recorded motion the markers (auto-labelization) in all frames. Then in a verification procedure, we examine and correct the labelization errors. Finally the marker trajectories are exported.



- 01_TR
  - Markers
  - Segments
    - CHEST
    - HEAD
    - LEFT ARM
    - LEFT FOOT
    - LEFT FORE ARM
    - LEFT HAND
    - LEFT LOWER LEG
    - LEFT THIGH
    - PELVIS
    - RIGHT ARM
    - RIGHT FOOT
    - RIGHT FORE ARM
    - RIGHT HAND
    - RIGHT LOWER LEG
    - RIGHT THIGH
  - Joints
  - Model Outputs

**Figure 3.4:** Model segments definition in Vicon® Nexus for subject 01_*TR* using captured markers.

### 3.1.5 Motion reconstruction

We have reconstructed all the motions recorded in the experiments using the software RPx developed by LBMC [MWT08] (Two examples are shown respectively in Fig. 3.6 and Fig. 3.7).

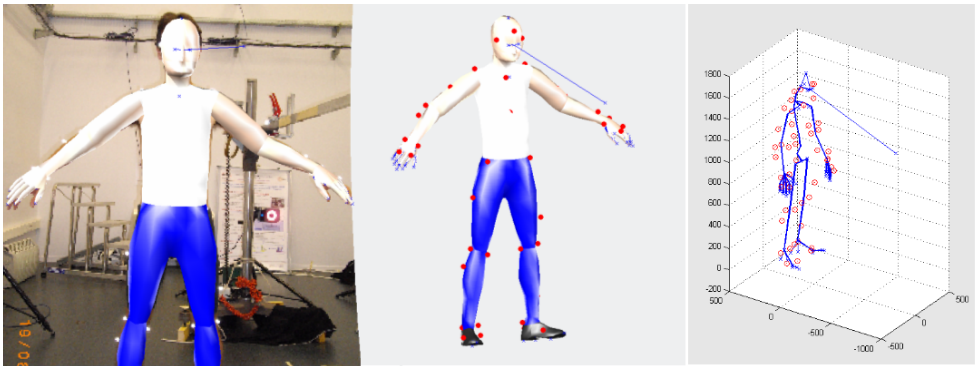


**Figure 3.5:** Left: tuning DHM posture for superposing 3D DHM model with subject's body in the calibrated photo; middle: markers are attached to 3D DHM model in the virtual environment; right: computing articular angles from recorded marker positions based on inverse kinematics and optimization techniques.

In RPx® software, a kinematic digital human model (26 body segments with 62 DoFs), based on the commercial model Ramsis® (see Appendix D), is tailored to subject's body using the measured subject body dimensions. Markers attached to the virtual human in software RPx are defined by superposing the 3D model with the photos calibrated beforehand using the 3D position of the markers, their 2D positions in the photographs and a DLT (Direct linear transformation) approach (see left figure in Fig. 3.5). With the exported trajectories of all the markers, the articular angles are computed in the reconstruction process by IK (Inverse Kinematics) and an optimization technique [WCM+05] (see right figure in Fig. 3.5).

## 3.2 Motion data analysis

We aim to obtain motion principles of human beings in cluttered environments by observing and analyzing the recorded motions in the above-mentioned experiments. We present in this section the analysis of the order and the placements of the supports configurations, the transition time spans and CoM trajectories with different environmental configurations.

**Definition 1.** ***Support****: a point or surface at which the environment and the DHM apply forces or moments on each other.*

In this thesis, we suppose that a DHM can realize two types of supports with the environment: contacts and grasps. However, contact or grasp of DHM on itself is not considered as support in this study.

**Definition 2.** ***Support configuration****: a set of supports that a DHM realizes with the environment.*

One should notice that a support configuration involves both the body segment and the environmental object for each valid support. A DHM motion then can be considered to be composed of a sequence of "transition phases" between instants of support configuration changes.

### 3.2.1 Support configurations

Human beings realize their motions via a sequence of supports with the environment. Thus it is important to observe and understand how a human makes and changes the supports during his motions. Based on the reconstructed motions of each subject, we have estimated the trajectories as well as the velocities and accelerations of the CoM, the end-effectors (hands, feet) and the pelvis from individual segment kinematics and numerical derivation associated to low pass filtering (Butterwoth recursive 2nd order low pass filter with cut-off frequency set at 5Hz). Then by detecting the variation of velocities of each end-effector, an algorithm automatically extracts the support configurations of the subject throughout the motion.

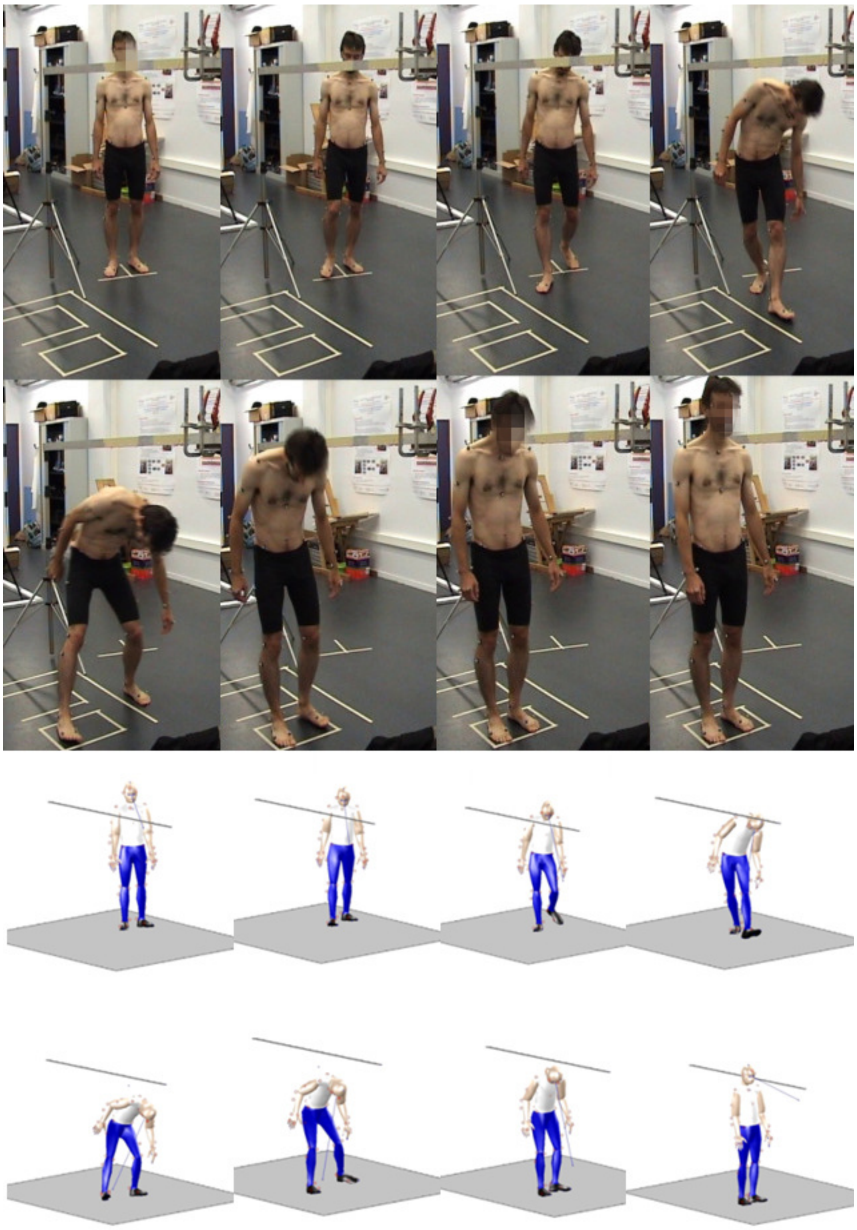


**Figure 3.6:** Snapshots of a reconstructed motion: passing under a beam.

### Supports sequence

The sequences of support configurations in recorded motions for material configurations *C1* and *C2* for all the three subjects with free strategy ($L$) are shown in Table 3.2. The motions for material configurations *C3* have not been included in this analysis since the final motion condition in this scenario is different from the other two. Among the 24 motions that we have observed, there are 23 motions in which the subjects adopted the same sequence of supports ("left foot first" footstep sequence) and transition phases in these motions are illustrated in Fig. 3.8.

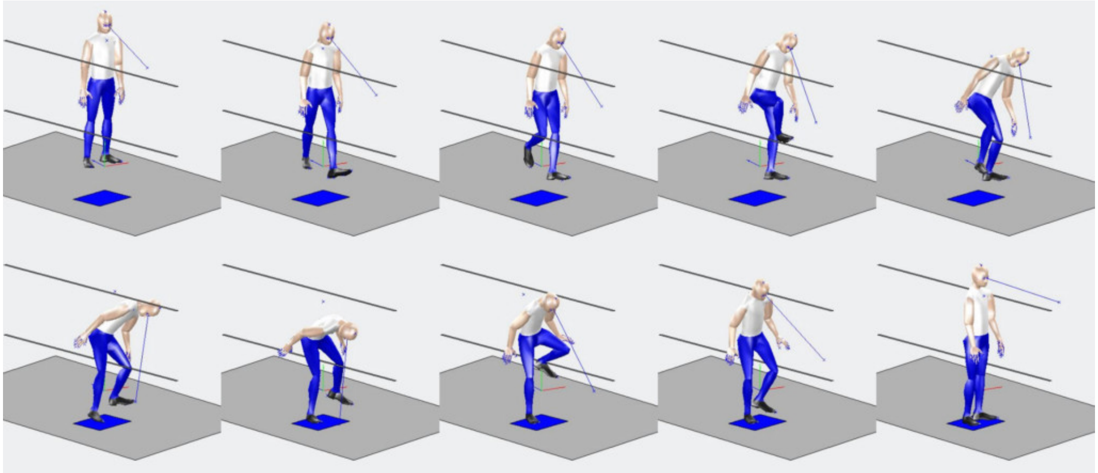


**Figure 3.7:** Snapshots of a reconstructed motion: passing under a beam with environmental configuration $C2$.

Previous analysis on human motion in clutered environment led to similar results. For example Chateauroux et al. ([Cha09], [CW10]) analyzed egress motions of 25 subjects (young and elderly) for 4 types of cars through the interactions between the participant and the environment. Despite the variability of subjects' anthropometry and physical capacities and of the car geometries, they found that only two main strategies were used by the subjects.

Relying on the above-mentioned observation, we conclude here the motion principle for the sequence of supports configurations:

**Conclusion 1.** *A human realizes his motion via multiple support configurations; the human motions show high consistency in terms of their supports sequence for the same scenario.*

In some researches, the supports are classified into different categories for better understanding human motions in terms of interactions with the environment. In ([Cha09]), hand-environment interactions are classified into four different categories according to their function in a human motion: actuation, balance, position perception and comfort. In our study, the supports can be also classified into *active* and *passive* ones according to their importances for the motion. For example, the feet contacts in our recorded motions are active supports since they actuate the human's motion. However, hand grasps in our experiments are passive supports since human subjectives have realized their motions without hand grasp. Precisely speaking, **Conclusion 1** addresses the high consistency for active supports sequence, whereas the passive supports may show randomness in their sequences (see for example analysis of hand contacts during car-ingress-egress movements in [Cau11]).

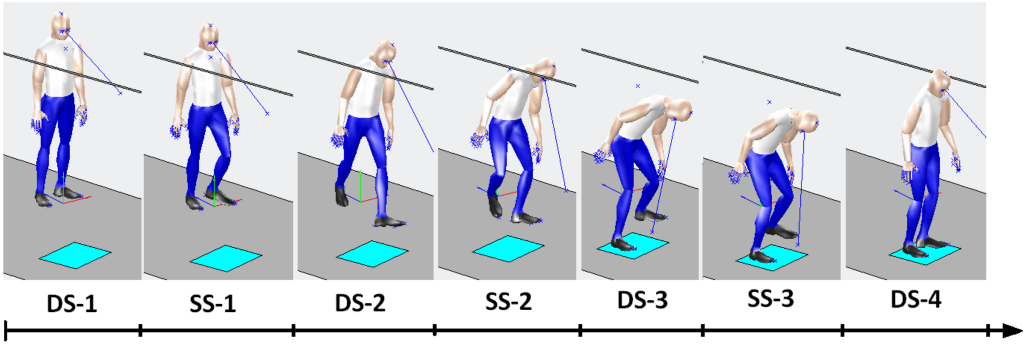


**Figure 3.8:** A motion is decomposed as a sequence of transition phase according to the foot contact changes.

| Motion phases and support configuration sequences | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| NO. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Percent |
| Phases | DS-1 | SS-1 | DS-2 | SS-2 | DS-3 | SS-3 | DS-4 | SS-4 | DS-5 | |
| Support | LR | L | LR | R | LR | L | LR | R | LR | 1/24 |
| | LR | R | LR | L | LR | R | LR | - | - | 23/24 |

**Table 3.2:** Order of support configurations in motions with free strategy for configuration $C1$ and $C2$. 'L': left foot; 'R': right foot. 'DS': double-support phase; 'SS': single-support phase.

### Support placement

Afterward, we have analyzed the placements of the supports. The foot placements for motions of the subject 01_$TR$ are shown in Fig. 3.9. In this example, we have analyzed the motions in which the subject steps firstly his left foot ("left foot first" footstep sequence).

As we have seen from the results, for the same motion conditions, a human realizes his supports at the same zone with small dispersion regarding the motion distance. The same analysis has been carried for motions of the other subjects, and we have observed the same phenomena in their support placements.

**Conclusion 2.** *For the same support sequence, a real human's supports show high consistency in their positions.*

## 3.2.2 Motion timing

### Key frames

The MoCap sampling frequency is $100Hz$, thus human motions in the experiments are recorded and then reconstructed at a rate of 100 *frames*/$s$. In this study, the

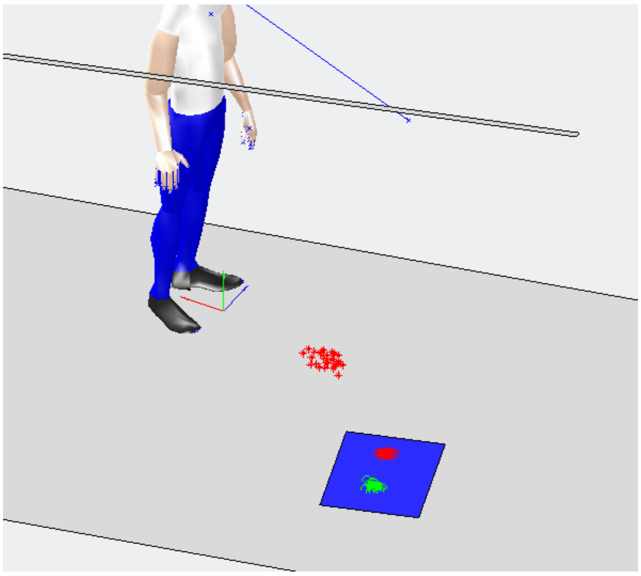


**Figure 3.9:** The foot contact positions of the subject 01_$TR$ in the motions with "left foot first" sequence. Red points are for left foot and green ones are for the right foot.

concept "key frame" related with the support configurations is defined as below.

**Definition 3.** ***Key frame****: a sampling instant at which the support configuration of the human changes.*

The key frames were extracted automatically by our algorithm, by detecting the variations of the hand's or the feet's positions and velocities. Particularly, the first and the last frame in each motion are also considered as key frames.

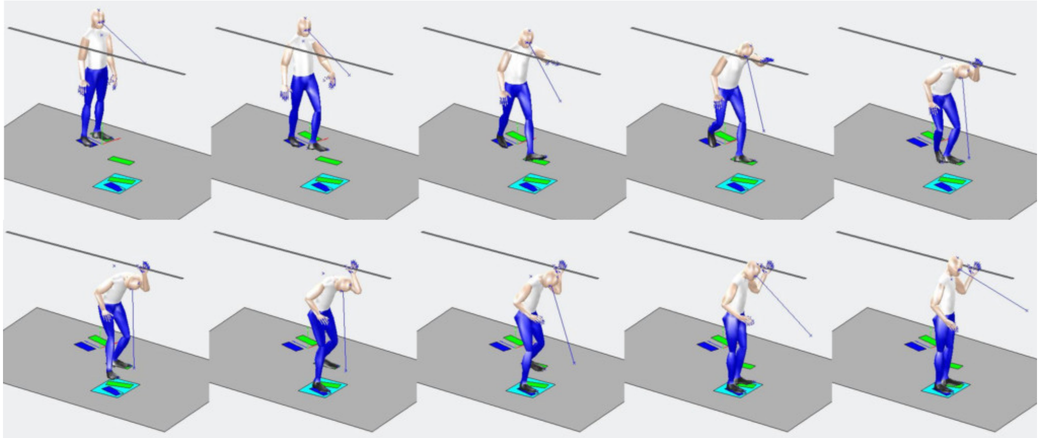


**Figure 3.10:** Snapshots of a reconstructed motion (01_$TR$_$H1C1S2$_1) in which the subject passes under a beam with left hand grasp. The subject has a stature of 1.75m; the beam is 1.45m high.

An example of a recorded motion ($01\_TR\_H1C1S2\_1$) is given in Fig. 3.10. In this motion, the subject walks under a beam with help of a left hand grasp. There are 9 key frames identified namely:

- Beginning of motion (frame 1)
- Left foot lift-off (frame 158)
- Left foot landing (frame 209)
- Right Foot lift-off (frame 222)
- Hand grasp (frame 240)
- Right Foot landing (frame 285)
- Left foot lift-off (frame 296)
- Left foot landing (frame 346)
- End of motion (frame 372)

### Motion timing analysis

The transition time spans in motions of subject $01\_TR$ are listed in Table 3.4. In this section, we want to analyze the motion timing characteristics in order to get knowledge about the influence of the obstacles on the duration of a transition phase. The key frame at which the subject makes his left hand grasp in the 10 motions with motion strategy “S2” are listed in the last column of the table.

We have analyzed the motions of the subject “$01\_TR$” in $C1$ and $C2$ configurations which adopt the “left foot first” footstep sequence shown in Fig. 3.11.

As we have observed from the data in Table 3.4, the environmental configurations have no influence on the durations of the first single-support phase (SS-1: left foot flying) and the second double-support phase (DS-2) since there is no obstacle involved in the two phases. The subject takes about 0.5s to complete his left foot step and spends about 0.1s in the double-phase when he moves on an even ground in an open local environment.

There is important influence of the environmental configurations on the second single-support. The transition phase SS-2 is longer in C2 than in the C1 configuration. It corresponds to the local influence of the environment, as in C2 the flying end-effector (right foot) has to step over the lower obstacle.

In the third double-support phase (DS-3), the subject moves his trunk from the left side of the beam to the right side. As we can see in the table, both the beam height and the lower obstacle have influence on the duration of this phase. For motions in configuration $C1$, difference in beam heights $H1(1.45m)$ and $H2(1.30m)$ does not show obvious influence on the duration. However, the subject spends more time (about 0.58s) in this phase for the height $H3(1.10m)$ than for other heights (about 0.23s). The similar effect of roof heights on human’s discomfort has also been observed, as addressed in [Cau11]. Configuration C2 with beam height $H2$, also makes the subject spend more time (about 0.55s) in this phase than in configuration $H2C1$.

In SS-3 phase, the subject makes a left foot step. To step over the lower obstacle, the subject spends more time in configuration $C2$ (about 0.98s) than in $C1$ (about 0.52s).

| Time distributions (unit: 10ms) | | grasp time |
|---|---|---|
| $H1C1L$_1 | [151 32 18 50 12 55 26 58] | |
| $H1C1L$_2 | [110 48 12 61 27 54] | |
| $H1C1S1$_1 | [175 54 8 62 28 41] | |
| $H1C1S1$_2 | [173 57 8 60 34 49] | |
| $H1C1S2$_1 | [157 51 13 63 11 50] | 240 |
| $H1C1S2$_2 | [152 50 9 70 9 55] | 235 |
| $H2C1L$_1 | [179 47 16 54 20 50] | |
| $H2C1L$_2 | [179 46 14 60 23 54] | |
| $H2C1S1$_1 | [139 51 11 60 41 56] | |
| $H2C1S1$_2 | [197 60 6 55 33 49] | |
| $H2C1S2$_1 | [172 50 11 64 20 46] | 258 |
| $H2C1S2$_2 | [157 57 9 77 16 56] | 250 |
| $H3C1L$_1 | [122 44 11 63 49 65] | |
| $H3C1L$_2 | [130 47 11 59 61 52] | |
| $H3C1S1$_1 | [194 57 5 62 73 44] | |
| $H3C1S1$_2 | [219 52 10 59 63 46] | |
| $H3C1S2$_1 | [144 53 8 68 46 55] | 245 |
| $H3C1S2$_2 | [132 52 9 68 55 55] | 233 |
| $H2C2L$_1 | [126 42 9 81 55 99] | |
| $H2C2L$_2 | [170 45 10 89 58 94] | |
| $H2C2S1$_1 | [155 56 5 79 63 102] | |
| $H2C2S1$_2 | [166 51 9 97 61 112] | |
| $H2C2S2$_1 | [157 46 10 86 39 86] | 260 |
| $H2C2S2$_2 | [236 60 7 80 52 94] | 358 |
| $H2C3L$_1 | [123 27 18 45 11 94 38 73] | |
| $H2C3L$_2 | [152 46 11 106 43 73] | |
| $H2C3S1$_1 | [206 48 6 84 69 80] | |
| $H2C3S1$_2 | [166 47 9 75 74 70] | |
| $H2C3S2$_1 | [118 51 8 88 42 69] | 210 |
| $H2C3S2$_2 | [171 54 5 90 50 85] | 273 |

**Table 3.4:** Time durations of transition phases between foot contact configurations in the experiments for the subject 01_$TR$.

**Conclusion 3.** *An obstacle influences locally the motion timing, namely it influences only the duration of the transition phase in which the human avoids the obstacle. In the same local environment, a transition phase in repeated human motions shows consistency in its duration.*

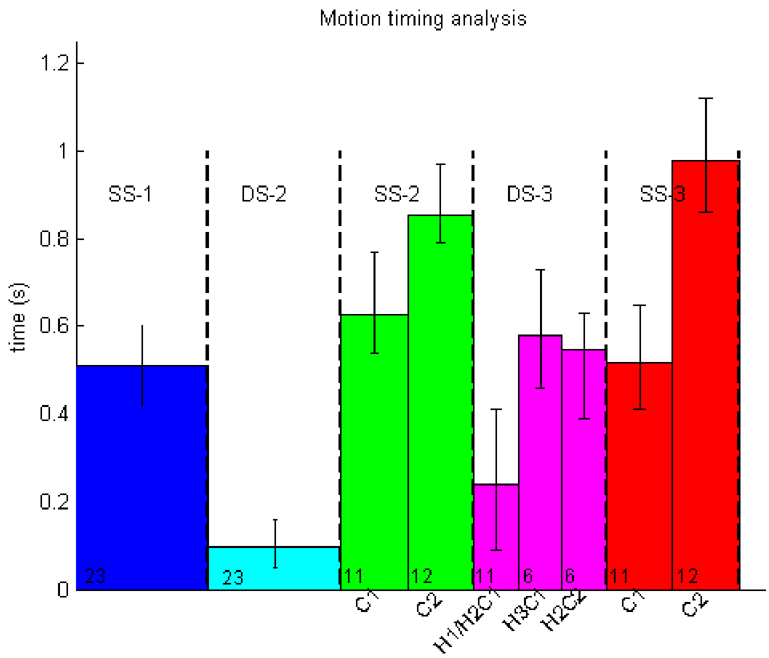


**Figure 3.11:** Motion timing analysis for the subject "01_$TR$" in $C1$ and $C2$ configurations. Altogether 5 transition phases are analyzed. Colored bars show the mean duration of each phase in corresponding material configurations. The maximum and minimum values are shown as well. The number of analyzed motions corresponding to each bar is indicated at its bottom.

### 3.2.3 CoM position analysis

The center of mass (CoM) plays a key role in DHM balance during its motion. Thereafter, we analyze the subjects' CoM positions during the recorded motions. The CoM trajectories of subject 01_$TR$ are shown in Fig. 3.12. The CoM height for the initial standing posture (at the beginning of each motion) is about $0.97m$.

#### Distance to foot contact

Firstly, we analyze the distance from the CoM to contact points, aiming to get knowledge on geometric limits brought by supports. We take the projection of a contact foot's CoM on the ground as the contact point for computing this distance. This distance is computed for any foot contact in every frame of a motion. The largest distances for each foot in all the motions are shown in Table 3.5. We can see that the geometric limit on distances from CoM to contact points are rather consistent in all the motions. This distance limit is estimated to be about $1.0m$ in most motions.

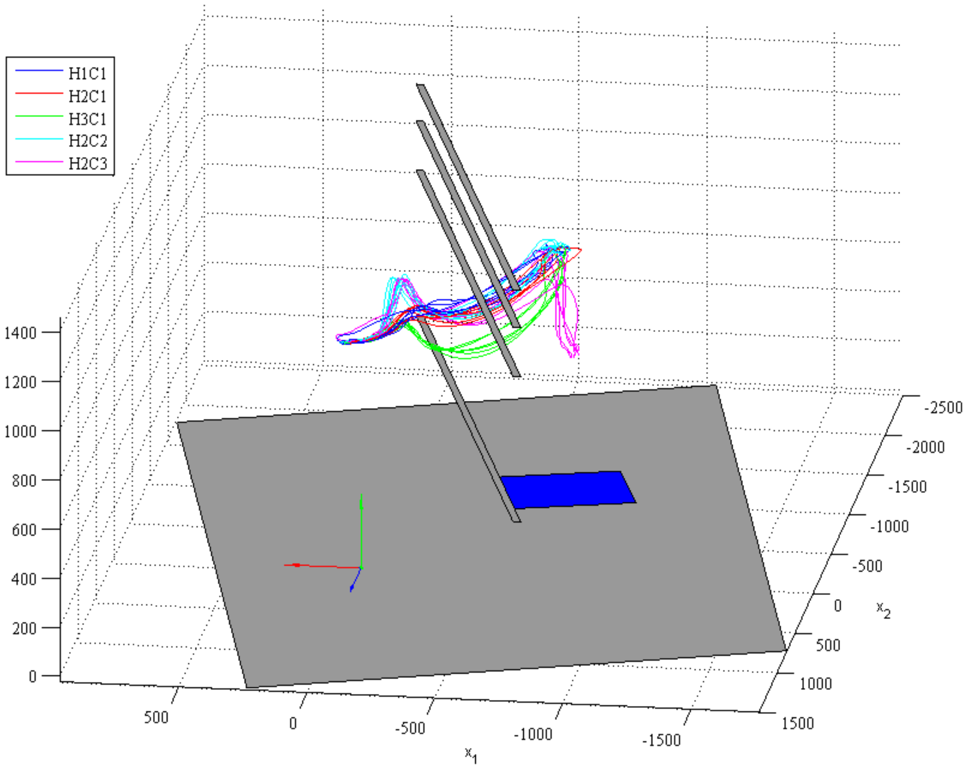


**Figure 3.12:** CoM trajectories of motions for the subject "01_$TR$" in the virtual environment.

| Upper limits for the distance from CoM to feet ($m$) | | |
|---|---|---|
| Feet | left foot | right foot |
| Min | 0.9950 | 0.9962 |
| Max | 1.0331 | 1.1012 |
| Standard deviation | 0.0117 | 0.0261 |
| Average | 1.0101 | 1.0205 |

**Table 3.5:** Maximal distance from the CoM to a foot contact in all the motions realized by subject "01_$TR$".

### Beam height influence

Then we aim to analyze the influence of the beam on the trajectory of CoM. The lowest height of CoM during the phase of trunk transition under the beam in each motion has been given in Fig. 3.13. We can see that the subject lowers his CoM by about 0.12$m$ when he passes under the beam at height $H1$. When the beam height is reduced to $H2$, the CoM height has an insignificant decrease (about 0.8$m$ regarding 0.85$m$ for $H1$). When the beam height is reduced to $H3$, the subject's CoM height decrease significantly to about 0.65$m$. This effect of different beam heights observed here is in agreement with the effect of roof heights on human's discomfort addressed in [Cau11].

CoM trajectories in motions for configurations $H2C2$ and $H2C3$ display a peak before passing under the beam, as shown in Fig. 3.12. The reason for this effect is that the human lifts his leg for stepping over an obstacle which brings the CoM to a high position. However, one can notice that this influence is very local, and that the remaining of the trajectory is very similar between configuration C1H2, C2 and C3. In the same way, one can observe that the seating motion at the end of the configuration C3 alter the CoM trajectory only at the end of the motion. The trajectories during the approach and the passage between the beams are similar between configuration C2 and C3.

**Conclusion 4.** *Obstacles in the cluttered environment influence locally the trajectory of the human's CoM.*

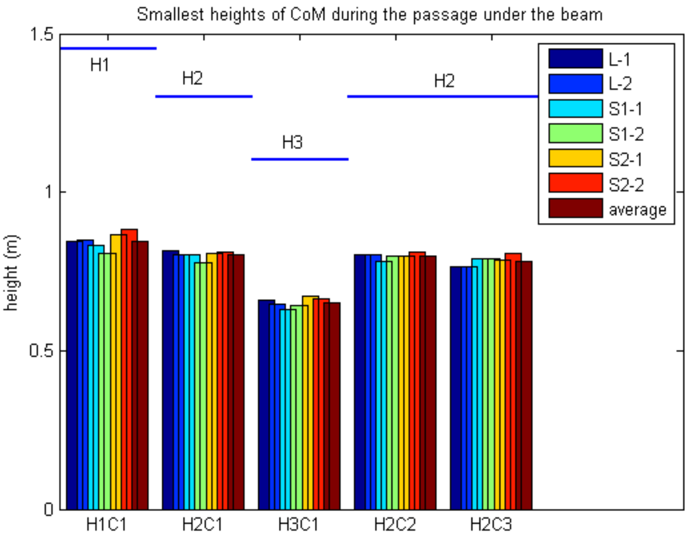


**Figure 3.13:** Illustrating the influence of the configuration on the smallest height of CoM in each motion.

The height difference between the lowest CoM and the beam has also been analyzed in order to understand the height margin for DHM to avoid the overhead obstacle. The estimated height differences in all the motions are shown in Fig. 3.14. In fact, the lower the beam is, the more difficult it will be for the subject to pass under it. It is interesting to observe both results from Fig. 3.13 and Fig. 3.14. CoM height for beam heights H1 and H2 are similar, leading to different height margins between the head and the beam. However, when the beam is lower (height H3), the subject tends to save a minimum margin (about 45 cm) and thus lower more its CoM. The beam seems then to have a non-linear influence on the motion. This phenomenon was already observed by Causse et al. ([Cau11], [CWD11]) while studying the effect of the car roof height on the kinematics of car egress motions.

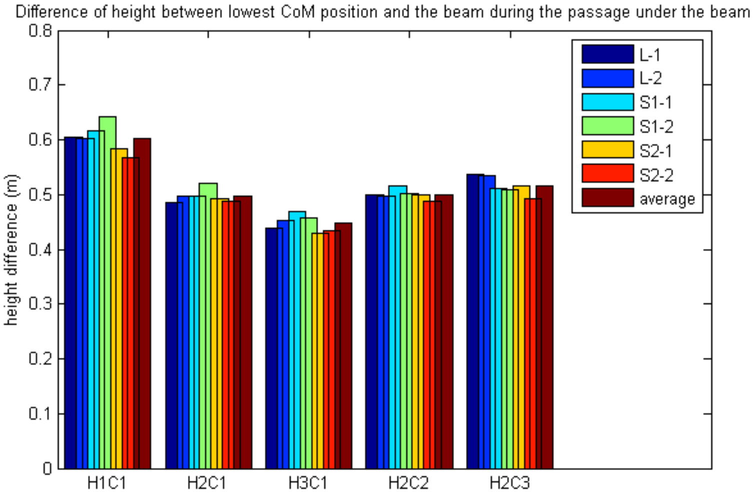


**Figure 3.14:** Illustrating the height difference between the lowest CoM and the beam during the passage under the beam.

## 3.3 Conclusion and perspectives

### 3.3.1 Experimentation

In this section, we have presented the experimental work that has been carried out during the thesis. We have presented the preparation, the materials, the equipments and the procedure of the experiments in which a series of motions in “passage under a beam” scenarii have been recorded. Then the recorded motions have been processed and reconstructed for analysis. The analysis work is concerned with support configuration (order and placement), key frame extraction and timing analysis, and influence of supports and obstacles on CoM position.

### 3.3.2 Motion principles summary

Based on observations and analysis, we have obtained general principle and characteristics of human motions in cluttered environments summarized as follows:

1. For the same scenario with the same initial and final motion conditions, a real human subject’s motions show high consistency and repeatablity in terms of sequence and placement of their support configurations;
2. Support placements constrain the position of human’s CoM;
3. Obstacles constrain locally the trajectory of human’s CoM;
4. Obstacles influence locally the motion timing.

### 3.3.3 Perspective for following works

The generic knowledge on human motions in cluttered environments obtained from observations and analysis of the recorded motions is expected to be used for simulating knowledge-driven motions in the following work. Moreover, based on the knowledge on motion principles and human motion characteristics, we specify for the following work on DHM's knowledge-driven motion simulation the following hypotheses:

- The DHM motion consists of multiple phases with different support configurations;
- DHM realizes its motions via generic type of supports for end-effectors including contacts (planar, non-planar and non-coplanar) and grasps;
- Support configuration sequence and placements can be pre-defined by an expert user for a DHM moving in a given cluttered environment;
- CoM trajectories (including motion timing) are variables that need to be solved relying on multiple constraints (geometry, obstacle avoidance, balance, etc).

## CHAPTER 4

# Balance and stability margins

*Balance* (or *equilibrium*) is one of the most important issues in the study of DHMs and legged robots. We define here the concept of *balance* for a DHM in two cases:

- *Static balance*: the DHM stays in a motionless state with zero velocity and acceleration;

- *Dynamic balance*: the DHM is moving and is able to modify its motion as desired via its supports with the environment without violating physic limits.

During its motion in physical environments, a DHM is always expected to be in *stable balance*. The conception *stable* (also *stability*) here is defined from the aspect of control for evaluating the ability of maintaining balance and resisting disturbances. Hereafter in this thesis without particular explanation, *balance* implies *stable balance*.

In this study, we aim to ensure the balance of a DHM during its movements in cluttered environments. Generally speaking, movements of a DHM, same as human motions in our daily life, are usually dynamic and acyclic. Moreover, during its motions, a DHM can interact with the environment with any part of its body. Thus we impose the following requirements for balance criterion adopted in our study:

- it should be valid in both dynamic and static cases;
- it can treat general support types including contacts (planar, non-planar or non-coplanar) and grasps;
- it should be a quantitative criterion thus it can also be used to evaluate the quality or safety of the balance;
- it should be able to be integrated in motion planning or motion control.

In this chapter, Section 4.1 presents a state of the art about the existing balance criteria; then in Section 4.2 we formulate a general balance criterion based on a mass point model and we compare it with the classical criteria in some particular situations; in Section 4.3 we extend this criterion for a whole-body human model; in Section 4.4, the formulated criterion is applied on some recorded motions to validate its relevance and to analyze influence of the difference between two models; finally we conclude this chapter in Section 4.5.

## 4.1 State of the art

### 4.1.1 Criteria of balance

#### Projection of CoM

Center of mass (CoM) or center of gravity (CoG) is the point at which the net gravity force acts on an object. In static cases, the projection of CoM on the ground is the most important concept for verifying balance of an object ([BL08], [Wie02]). Generally speaking, the object is in static balance (statically stable) if the projection of its CoM on the ground lies inside the convex hull of support (support polygon). If the projection lies outside the polygon, it will be driven by the uncompensated moment to rotate about one point on the boundary of its support polygon, thus the object will lose balance.

This criterion is facing difficulties when there exist non-planar contacts, since the supporting polygon is a planar concept. Some researchers propose some techniques to improve this criterion by computing a polygon to limit the horizontal position of the CoM [BL08].

#### Zero moment point and Centre of pressure

The concept of *Zero Moment Point* (ZMP) was for the first time proposed and applied in late 1960s [VJ69]. Since then, it has become the most popular tool for verifying, analyzing and controlling dynamic balance and it has been broadly used in many studies in different domains. As defined in [SB04], ZMP is the point on the ground where the *tipping moment* acting on the legged robot equals zero. The *tipping moment* is defined as the component of the moment (due to gravity and inertia forces) that is tangential to the supporting surface. The *Centre of Pressure* (CoP) is a concept that is very similar to ZMP. For a body-plane contact, the pressure forces (normal to the contact surface) are equivalent to a resultant force applied at the CoP where the resultant moment is zero. As disserted in [SB04], the CoP and the ZMP are the same point as long as all the ground-sole contacts are co-planar. *Supporting polygon* is a concept associated with the ZMP criterion which is defined as the hull enveloping all the contact surfaces. The system is in dynamical balance if the ZMP is inside the supporting polygon. When the system loses balance, the ZMP is on the boundary of the supporting polygon.

Since balance is ensured by confining a reference point inside a convex polygon, the ZMP thus can be used as a quantitative criterion for assessing the quality of balance. When a system is in balance, the distances from the ZMP to boundaries of the supporting polygon thus can be considered as the stability margins in different directions. The limitations of this stability margin concept lie in two aspect:

1. it is applicable only when the system is in balance; when the system loses balance, it cannot give a distance for the system to regain balance;
2. it measures a distance on ground which does not express directly the stability margin as a dynamic concept. Thus we cannot directly evaluate the disturbance (for example, a force or a wrench) or the variations of the CoM state that the system can resist.

Based on this criterion, many researchers have proposed various criteria, for example, adios-ZMP [HHH$^+$06], FZMP [YAOH05], GZMP [HKKH03], etc.

### Foot rotation indicator (FRI)

FRI is a criterion to verify balance of a legged robot [Gos99]. It is defined as a point on the foot-ground contact surface, during the single-support phase, at which the net ground-reaction force would have to act to keep the foot stationary. The robot is in balance if the FRI point is inside the convex hull of the foot support surface, in which case it is the same point as the CoP or ZMP. However, the FRI point may lie outside the convex hull, meaning that the robot loses balance and its foot rotates. This indicator is an extension of the CoP concept, since it can be used to evaluate the stability as well as the instability of the robot.

### CMP and ZRAM point

The concept of CMP (Centroidal Moment Pivot) is proposed based on the biomechanical conclusion that spin angular momentum or the body's angular momentum about the CoM remains small for human walking on level ground [PHH04a, PHH04b, PGH05]. The CMP location is defined as a point where a line parallel to the ground reaction force, passing through the CoM, intersects with the ground surface. Based on the same principle, the ZRAM (Zero rate of change of angular momentum ) point [GK04] is a point on the ground where the total ground reaction forces would act so that the rate of change of centroidal angular momentum of the robot equals zero.

### Admissible subspaces

Since various physic limits are associated in the definition of balance, to get the admissible balance conditions, it is possible to define for some parameters their *admissible* or *feasible* intervals , for example, unilateral frictional cone for a contact force, force limit for a grasp, etc. Based on the criterion of adios-ZMP [HHH$^+$06], a strong stability criterion has been proposed in [HHK$^+$07] as the *contact wrench cone (CWC)* with the assumption that there is sufficient friction at contacts. The unilaterality of contact forces are used for confining the admissible CWC. The *contact wrench sum (CWS)* should be an internal element of this set to ensure the balance. A similar but more general criterion is defined as the *admissible contact wrench space* in [BB08] which is a set of the admissible generalized wrench defined by constraints on the contact forces (Coulomb model). Stability margin is also proposed in [BB08] for defining the robustness of the postural balance.

## 4.1.2 Comments on the existing criteria

The projection of CoM can deal with planar or non-planar contacts, however, it is only valid in static cases. The ZMP and its variants are limited when the system has non-coplanar or non-planar contacts. Besides, the ZMP criterion does not take into account constraints of contact forces, for example, the friction limits. The FRI point, is applicable only during the single support phase of a legged robot. The ZRAM and CMP take whole-body posture into account for balance definition and analysis. The admissible contact wrench space and admissible perturbation space are general criteria that take into account frictional constraints of contact forces and they can treat cases in which non-coplanar contacts exist.

Based on the above comparison, we formulate the balance criterion by extending the admissible contact wrench space criterion in terms of some more precise limits over the external wrenches applied on general supports. The detailed formulations and analysis are presented in next section.

## 4.2 Formulation of a balance criterion

In this section, we formulate the balance criterion that will be used in this study for defining balance constraints. This section starts with the presentation of a simplified model for the DHM. In this model, a DHM is represented as a mass point at its CoM which interacts with the environment at some supports (contacts or grasps). The constraints over the wrenches applied at the supports then will be projected onto the space of the DHM's dynamics in order to get a criterion that can be easily integrated in motion planning and motion analysis works. This criterion is compared with traditional ones in several simulations in different cases in order to verify its validity. Finally several stability margins are proposed based on this criterion.

### 4.2.1 Simplified mass point model

We explore our criterion using a simplified model (see Fig. 4.1). It has a mass point whose position is denoted as $\mathbf{x} \in \mathbb{R}^3$. This mass point is in dynamic balance while interacting with the environment via a series of supports. A certain (the $i$-th) support is associated with an exterior wrench consists of a force $\mathbf{f}_i = \begin{bmatrix} f_{xi} \\ f_{yi} \\ f_{zi} \end{bmatrix}$ and a moment $\boldsymbol{\tau}_i = \begin{bmatrix} \tau_{xi} \\ \tau_{yi} \\ \tau_{zi} \end{bmatrix}$. Force $\mathbf{f}_i$ is applied at point $\mathbf{r}_i \in \mathbb{R}^3$. All the variables are expressed in the global coordinate system (GCS) with the origin at point $O$. We attach a local coordinate system (LCS) at the contact point $O_i$ whose three axis are respectively (expressed in the GCS): the normal direction $\mathbf{v}_i$, the sagittal direction $\mathbf{s}_i$ and the tangent direction $\mathbf{t}_i$. Suppose that this LCS has its orientation $\mathbf{R}_i$ regarding the GCS which defines the following relations: $\mathbf{n}_i = \mathbf{R}_i \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$, $\mathbf{s}_i = \mathbf{R}_i \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ and $\mathbf{t}_i = \mathbf{R}_i \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$.

The orientation of LCS can be obtained from the local axis as well: $\mathbf{R}_i^t = \begin{bmatrix} \mathbf{s}_i^t \\ \mathbf{t}_i^t \\ \mathbf{n}_i^t \end{bmatrix}$

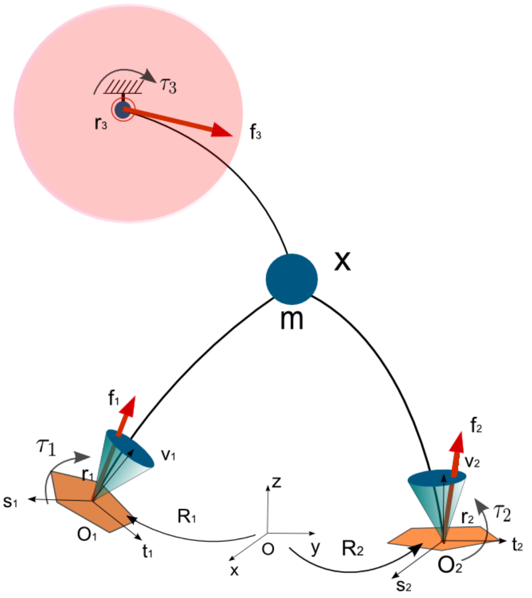


**Figure 4.1:** Simplified dynamic model for balance formulation: a mass point with non-planar contacts and grasps. A force $\mathbf{f}_i = [f_{ix}, f_{iy}, f_{iz}]^t$ applied by the environment at a support point $\mathbf{r}_i = [x_i, y_i, z_i]^t$ should be limited inside the frictional cone or the grasp force sphere. And a moment $\boldsymbol{\tau}_i = [\tau_{xi}, \tau_{yi}, \tau_{zi}]^t$ is applied over the $i$-th support. All the variables are expressed in a global coordinate system (GCS) with its origin at point $\mathbf{O}$.

### 4.2.2 Dynamics formulation

Suppose that the simplified model interacts with the environment at $n$ contact points and $k$ grasps. The dynamic equations of the system are:

$$m\ddot{\mathbf{x}} - m\mathbf{g} = \sum_{i=1}^{n+k} \mathbf{f}_i \tag{4.1}$$

$$\hat{\mathbf{x}}\,(m\ddot{\mathbf{x}} - \, m\mathbf{g}) = \sum_{i=1}^{n+k} (\hat{\mathbf{r}}_i\,\mathbf{f}_i + \boldsymbol{\tau}_i) \tag{4.2}$$

Denote a vector $\mathbf{f} = \begin{bmatrix} \mathbf{f}_c \\ \mathbf{f}_g \end{bmatrix} \in \mathbb{R}^{3(n+k)}$ integrating the $n$ contact forces and the $k$ grasp forces and another vector $\boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{\tau}_c \\ \boldsymbol{\tau}_g \end{bmatrix} \in \mathbb{R}^{3(n+k)}$ for the exterior moments.

The inertial force together with the gravity force is considered as the *pseudo-force* acting on the system:

$$\mathbf{f}_p = m\ddot{\mathbf{x}} - m\mathbf{g} \tag{4.3}$$

Then, Equation (4.1) can be expressed as:

$$\mathbf{f}_p = \mathbf{A}\mathbf{f} \tag{4.4}$$

Equation (4.2) defines the *pseudo-moment* of the model as:

$$\boldsymbol{\tau}_p = \hat{\mathbf{x}}\,\mathbf{f}_p = \mathbf{C}\mathbf{f} + \mathbf{A}\boldsymbol{\tau} \tag{4.5}$$

where

- $\mathbf{A} = \begin{bmatrix} \mathbf{I}, & \cdots, & \mathbf{I} \end{bmatrix}$;
- $\mathbf{C} = [\ \hat{\mathbf{r}}_1, \ \ \cdots, \ \ \hat{\mathbf{r}}_{n+k}\ ]$.

By combining Equation (4.4) and Equation (4.5), we can get the *pseudo-wrench* $\mathbf{w_p}$ composed by the pseudo-force and the pseudo-moment:

$$\mathbf{w}_p = \begin{bmatrix} \mathbf{I} \\ \hat{\mathbf{x}} \end{bmatrix} \mathbf{f}_p = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \mathbf{A} \end{bmatrix} \begin{pmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{pmatrix} \tag{4.6}$$

### 4.2.3 Admissible contact wrenches

#### No slipping at contact

For a contact point ($i$-th support at $\mathbf{r}_i$), the condition of no-slipping is ensured by:

$$\left\| \begin{bmatrix} \mathbf{s}_i^t \mathbf{f}_i \\ \mathbf{t}_i^t \mathbf{f}_i \end{bmatrix} \right\| \leq \mu_i \mathbf{n}_i^t \mathbf{f}_i \tag{4.7}$$

Equation (4.7) defines the frictional cone (Coulomb Model) that limits the contact force for avoiding slipping at a contact.

### No tipping over at contact

Now suppose that a contact is associated with a contact surface which is a convex polygon (2D-polytope) (see Fig. 4.2). Suppose the polygon $S_i$ is defined as (expressed in the LCS with its origin at $O_i$):

$$S_i := \{\mathbf{p} \in \mathbb{R}^2 \mid \mathbf{A}_i\mathbf{p} \leq \mathbf{b}_i\}. \tag{4.8}$$

We recall that the center of pressure (CoP) is defined as a point about which the net exterior moment has only the vertical component. Suppose the contact at point $O_i$ has its CoP at point $O'_i$ with its position in the LCS : $\mathbf{p}_i = \begin{bmatrix} x'_i \\ y'_i \\ 0 \end{bmatrix}$. In the GCS, the CoP $O'_i$ is represented by $\mathbf{r}'_i$. The moment about point $O'_i$ is expressed in the GCS as:

$$\boldsymbol{\tau}'_i = (\mathbf{r}_i - \mathbf{r}'_i) \wedge \mathbf{f}_i + \boldsymbol{\tau}_i \tag{4.9}$$

There is the following relation:

$$\mathbf{R}_i\mathbf{p}_i = \mathbf{R}_i \begin{bmatrix} x'_i \\ y'_i \\ 0 \end{bmatrix} = \mathbf{r}'_\mathbf{i} - \mathbf{r}_i \tag{4.10}$$

So, the moment about point $O'_i$ expressed in the LCS is:

$$\mathbf{R}^t_i\boldsymbol{\tau}'_i = \mathbf{R}^t_i[(\mathbf{r}_\mathbf{i} - \mathbf{r}'_i) \wedge \mathbf{f}_i] + \mathbf{R}^t_i\boldsymbol{\tau}_i = \begin{bmatrix} 0 \\ 0 \\ \tau'_{vi} \end{bmatrix} \tag{4.11}$$

where $\tau'_{vi}$ is a bounded real number.

We input Equation (4.10) and the rotation matrix $\mathbf{R}^t_i = \begin{bmatrix} \mathbf{s}^t_i \\ \mathbf{t}^t_i \\ \mathbf{n}^t_i \end{bmatrix}$ into the first two lines of Equation (4.11):

$$\begin{cases} x'_i = -\mathbf{t}^t_i\tau_i/\mathbf{n}^t_i\mathbf{f}_i \\ y'_i = \mathbf{s}^t_i\tau_i/\mathbf{n}^t_i\mathbf{f}_i \\ \tau'_{vi} = (y'_i\mathbf{s}^t_i - x_i\mathbf{t}^t_i)\mathbf{f}_i + \mathbf{n}^t_i\boldsymbol{\tau}_i \end{cases} \tag{4.12}$$

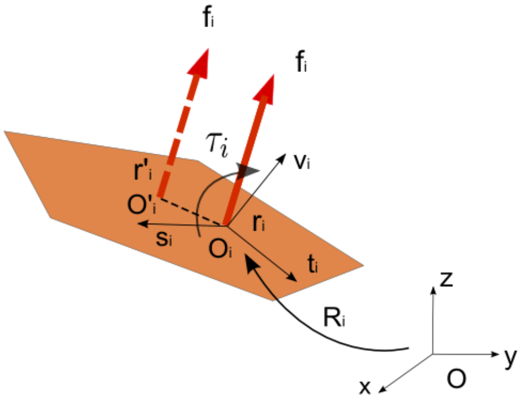


**Figure 4.2:** Center of pressure: limit over the exterior moment applied on a contact surface.

The CoP must be located inside the contact surface:

$$\mathbf{A}_i \begin{bmatrix} x'_i \\ y'_i \end{bmatrix} \leq \mathbf{b}_i \Rightarrow \mathbf{A}_i \begin{bmatrix} -\mathbf{t}_i^t \\ \mathbf{s}_i^t \end{bmatrix} \tau_i \leq \mathbf{b}_i \mathbf{n}_i^t \mathbf{f}_i \tag{4.13}$$

Equation (4.13) defines a limit on the exterior moment applied over the contact at $O_i$ in function of the contact force $\mathbf{f}_i$.

Then we have the constraint imposed by the contact surface:

$$\mathbf{A}_{si} \boldsymbol{\tau}_i - \mathbf{B}_{si} \mathbf{f}_i \leq 0 \tag{4.14}$$

where

- $\mathbf{A}_{si} = \mathbf{A}_i \begin{bmatrix} -\mathbf{t}_i^t \\ \mathbf{s}_i^t \end{bmatrix}$;
- $\mathbf{B}_{si} = \mathbf{b}_i \mathbf{n}_i^t$.

The limit for the vertical component of the moment:

$$\begin{aligned}
&\left| \mathbf{n}_i^t \boldsymbol{\tau}_i + \frac{\mathbf{s}_i^t \boldsymbol{\tau}_i \mathbf{s}_i^t \mathbf{f}_i + \mathbf{t}_i^t \boldsymbol{\tau}_i \mathbf{t}_i^t \mathbf{f}_i}{\mathbf{n}_i^t \mathbf{f}_i} \right| \leq \tau_{lim} \\
\Rightarrow &\left| \mathbf{n}_i^t \boldsymbol{\tau}_i + \frac{\begin{bmatrix} \mathbf{s}_i^t \boldsymbol{\tau}_i & \mathbf{t}_i^t \boldsymbol{\tau}_i \end{bmatrix} \begin{bmatrix} \mathbf{s}_i^t \mathbf{f}_i & \mathbf{t}_i^t \mathbf{f}_i \end{bmatrix}^t}{\mathbf{n}_i^t \mathbf{f}_i} \right| \leq \tau_{lim} \\
\Rightarrow &\left| \mathbf{n}_i^t \boldsymbol{\tau}_i \right| + \frac{\left\| \begin{matrix} \mathbf{s}_i^t \boldsymbol{\tau}_i & \mathbf{t}_i^t \boldsymbol{\tau}_i \end{matrix} \right\|}{\mathbf{n}_i^t \mathbf{f}_i} \left\| \begin{matrix} \mathbf{s}_i^t \mathbf{f}_i & \mathbf{t}_i^t \mathbf{f}_i \end{matrix} \right\| \leq \tau_{lim} \\
\Rightarrow &\left| \mathbf{n}_i^t \boldsymbol{\tau}_i \right| + \mu \left\| \begin{matrix} \mathbf{s}_i^t \boldsymbol{\tau}_i & \mathbf{t}_i^t \boldsymbol{\tau}_i \end{matrix} \right\| \leq \tau_{lim} \qquad (4.15)
\end{aligned}$$

### 4.2.4 Admissible grasp wrenches

We constrain a grasp force $\mathbf{f}_i$ by imposing an upper limit on its magnitude:

$$\|\mathbf{f}_i\| \leq f_g^{\max} \tag{4.16}$$

which defines a sphere in grasp force space. The grasp force vector should be located inside this sphere to avoid grasp breaking.

In the same way, a grasp moment $\boldsymbol{\tau}_i$ must be constrained by:

$$\|\boldsymbol{\tau}_i\| \leq \tau_g^{\max} \tag{4.17}$$

### 4.2.5 Constraints simplification

#### Contact constraints

To simplify the no-slipping constraint described in Equation (4.7), we only check it in sagittal direction and tangent direction of the contact:

$$\left\|\mathbf{s_i}^t \mathbf{f}_i\right\| \leq \mu_i \mathbf{n}_i^t \mathbf{f}_i \tag{4.18}$$

and

$$\left\|\mathbf{t}_i^t \mathbf{f}_i\right\| \leq \mu_i \mathbf{n}_i^t \mathbf{f}_i \tag{4.19}$$

Then no-slipping condition for a contact force thus can be expressed as:

$$\mathbf{A}_{fi} \mathbf{f}_i \leq 0 \tag{4.20}$$

where

- $\mathbf{A}_{fi} = -\begin{bmatrix} \mu_i \mathbf{n}_i^t + \mathbf{s}_i^t \\ \mu_i \mathbf{n}_i^t - \mathbf{s}_i^t \\ \mu_i \mathbf{n}_i^t + \mathbf{t}_i^t \\ \mu_i \mathbf{n}_i^t - \mathbf{t}_i^t \end{bmatrix}.$

We verify constraints defined by Equation (4.15) only in sagittal direction and tangent direction, which gives:

$$\mathbf{A}_{\tau i} \boldsymbol{\tau}_i \leq \mathbf{b}_{\tau i} \tag{4.21}$$

where

- $\mathbf{A}_{\tau i} = \begin{bmatrix} \mathbf{n}_i^t + \mu \mathbf{s}_i^t \\ -\mathbf{n}_i^t - \mu \mathbf{s}_i^t \\ -\mathbf{n}_i^t + \mu \mathbf{s}_i^t \\ \mathbf{n}_i^t - \mu \mathbf{s}_i^t \\ \mathbf{n}_i^t + \mu \mathbf{t}_i^t \\ -\mathbf{n}_i^t - \mu \mathbf{t}_i^t \\ -\mathbf{n}_i^t + \mu \mathbf{t}_i^t \\ \mathbf{n}_i^t - \mu \mathbf{t}_i^t \end{bmatrix}$ *and* $\mathbf{b}_{\tau i} = \begin{bmatrix} \tau_{lim} \\ \tau_{lim} \\ \tau_{lim} \\ \tau_{lim} \\ \tau_{lim} \\ \tau_{lim} \\ \tau_{lim} \\ \tau_{lim} \end{bmatrix}$.

**Grasp constraints**

To simplify the grasp constraints defined by Equation (4.16) and Equation (4.17), we only check this condition along three axis of the GCS:

$$\begin{cases} \mathbf{A}_{gfi}\mathbf{f}_i \leq \mathbf{b}_{gfi} \\ \mathbf{A}_{g\tau i}\boldsymbol{\tau}_i \leq \mathbf{b}_{g\tau i} \end{cases} \tag{4.22}$$

where

- $\mathbf{A}_{gfi} = \mathbf{A}_{g\tau i} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix}$;
- $\mathbf{b}_{gfi} = \begin{bmatrix} f_{gx}^{\max} \\ f_{gx}^{\max} \\ f_{gy}^{\max} \\ f_{gy}^{\max} \\ f_{gz}^{\max} \\ f_{gz}^{\max} \end{bmatrix}$ *and* $\mathbf{b}_{g\tau i} = \begin{bmatrix} \tau_{gx}^{\max} \\ \tau_{gx}^{\max} \\ \tau_{gy}^{\max} \\ \tau_{gy}^{\max} \\ \tau_{gz}^{\max} \\ \tau_{gz}^{\max} \end{bmatrix}$.

### 4.2.6 Synthesis of admissible support conditions

When there are $n(n \geq 1)$ contact wrenches and $k(k \geq 1)$ grasp wrenches, we can define in this case the balance conditions by synthesizing the balance conditions formulated in previous sections.

The $n(n \geq 1)$ contact wrenches are represented by a contact force vector:

$$\mathbf{f}_c = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_n \end{bmatrix} \tag{4.23}$$

and a contact moment vector:

$$\boldsymbol{\tau}_c = \begin{bmatrix} \boldsymbol{\tau}_1 \\ \vdots \\ \boldsymbol{\tau}_n \end{bmatrix} \tag{4.24}$$

For each contact force, Equation (4.20) should be satisfied to avoid slipping, so we have

$$\mathbf{A}_f \mathbf{f}_c \leq 0 \tag{4.25}$$

where

- $\mathbf{A}_f = diag(\mathbf{A}_{f1}, \ \cdots, \mathbf{A}_{fn})$.

In order to get the vertices, we need to enclose the polytope by defining another constraint for the vertical forces:

$$\mathbf{A}_v \mathbf{f}_c = \sum_{i=1}^{n} f_{zi} \leq \mathbf{f}_{limit} \tag{4.26}$$

By combining Equation (4.25) and Equation (4.26), we get:

$$\boxed{\mathbf{A}_c \mathbf{f}_c \leq \mathbf{b}_c} \tag{4.27}$$

where

- $\mathbf{A}_c = \begin{bmatrix} \mathbf{A}_f \\ \mathbf{A}_v \end{bmatrix}$ and $\mathbf{b}_c = \begin{bmatrix} 0 \\ f_{limit} \end{bmatrix}$.

The CoP condition by Equation (4.14) gives:

$$\boxed{\mathbf{A}_s \boldsymbol{\tau}_c - \mathbf{B}_s \mathbf{f}_c \leq 0} \tag{4.28}$$

where

- $\mathbf{A}_s = diag(\mathbf{A}_{si})$ and $\mathbf{B}_s = diag(\mathbf{B}_{si})$.

The limits for the vertical net moments at contact points by Equation (4.21):

$$\boxed{\mathbf{A}_{\tau c} \tau_c \leq \mathbf{b}_{\tau c}} \tag{4.29}$$

where

- $\mathbf{A}_{\tau c} = diag(\mathbf{A}_{\tau i})$ and $\mathbf{b}_{\tau c} = \begin{bmatrix} \mathbf{b}_{\tau 1} \\ \vdots \\ \mathbf{b}_{\tau n} \end{bmatrix}$.

The $k(k \geq 1)$ grasp wrenches are represented as a grasp force vector:

$$\mathbf{f}_g = \begin{bmatrix} \mathbf{f}_{g1} \\ \vdots \\ \mathbf{f}_{gk} \end{bmatrix}$$

and a grasp moment vector:

$$\boldsymbol{\tau}_g = \begin{bmatrix} \boldsymbol{\tau}_{g1} \\ \vdots \\ \boldsymbol{\tau}_{gk} \end{bmatrix} \tag{4.30}$$

In the same manner as in balance condition synthesis for contact wrenches, Equation (4.22) defines the following constraints for all the grasp wrenches:

$$\boxed{\mathbf{A}_g \mathbf{f}_g \leq \mathbf{b}_g} \tag{4.31}$$

$$\boxed{\mathbf{A}_{g\tau} \boldsymbol{\tau}_g \leq \mathbf{b}_{g\tau}} \tag{4.32}$$

where

- $\mathbf{A}_g = diag(\mathbf{A}_{gfi})$ and $\mathbf{A}_{g\tau} = diag(\mathbf{A}_{g\tau i})$;
- $\mathbf{b}_g = \begin{bmatrix} \mathbf{b}_{gf1} \\ \vdots \\ \mathbf{b}_{gfk} \end{bmatrix}$ and $\mathbf{b}_{g\tau} = \begin{bmatrix} \mathbf{b}_{g\tau 1} \\ \vdots \\ \mathbf{b}_{g\tau k} \end{bmatrix}$.

So if we denote all the external wrenches as a vector:

$$\boldsymbol{\psi} = \begin{bmatrix} \mathbf{f}_c \\ \mathbf{f}_g \\ \boldsymbol{\tau}_c \\ \boldsymbol{\tau}_g \end{bmatrix} \tag{4.33}$$

we can conclude the balance conditions by combing Equations (4.27) - (4.32) in the following equation:

$$\boxed{\mathbf{A}_\psi \boldsymbol{\psi} \leq \mathbf{b}_\psi} \tag{4.34}$$

where

- $\mathbf{A}_\psi = \begin{bmatrix} \mathbf{A}_c & 0 & 0 & 0 \\ 0 & \mathbf{A}_g & 0 & 0 \\ -\mathbf{B}_s & 0 & \mathbf{A}_s & 0 \\ 0 & 0 & \mathbf{A}_{\tau c} & 0 \\ 0 & 0 & 0 & \mathbf{A}_{g\tau} \end{bmatrix}$ and $\mathbf{b}_\psi = \begin{bmatrix} \mathbf{b}_c \\ \mathbf{b}_g \\ 0 \\ \mathbf{b}_{\tau c} \\ \mathbf{b}_{g\tau} \end{bmatrix}$.

### 4.2.7 Balance criterion representation

Equation (4.34) defines a polytope in the exterior wrench space namely:

$$\Psi = \left\{ \boldsymbol{\psi} \in \mathbb{R}^{6(n+k)} \, | \mathbf{A}_\psi \boldsymbol{\psi} \leq \mathbf{b}_\psi \right\} \tag{4.35}$$

Equation (4.6) defines the projection of a point in the set $\Psi$ onto the 6D pseudo-wrench space. For all the points in $\Psi$, their projections result in the admissible pseudo-wrench space, a 6D polytope $\mathcal{W}_p$ defined as:

$$\mathcal{W}_p = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \mathbf{A} \end{bmatrix} \Psi := \left\{ \mathbf{w}_p \in \mathbb{R}^6 \, | \;\; \mathbf{H}\mathbf{w}_p \leq \mathbf{h} \; \right\} \tag{4.36}$$

The detailed computation in Equation (4.36) is given in Appendix A.

Thus, the balance criterion thus can be expressed as:

$$\mathbf{H}\mathbf{w}_p \leq \mathbf{h} \tag{4.37}$$

We conclude that the following arguments are equivalent:

- *The system is in balance*
- *The pseudo-wrench of the system is inside* $\mathcal{W}_p$
- *The system satisfies the relation:* $\mathbf{H}\mathbf{w}_p \leq \mathbf{h}$

We can furthermore project the admissible pseudo-wrench space to the space of the pseudo-force which results in the admissible pseudo-force space (see Fig. 4.3):

$$\mathcal{F}_p = \left\{ \mathbf{f}_p \in \mathbb{R}^3 : (\mathbf{H}_1 + \mathbf{H}_2 \hat{\mathbf{x}}) \mathbf{f}_p \leq \mathbf{h} \right\} \tag{4.38}$$

where

- $\mathbf{H}_1 = \mathbf{H}(:, 1:3)$ and $\mathbf{H}_2 = \mathbf{H}(:, 4:6)$.

The admissible pseudo-wrench space can also be projected to get the admissible pseudo-moment space:

$$\mathcal{T}_p = \hat{\mathbf{x}} \mathcal{F}_p \tag{4.39}$$

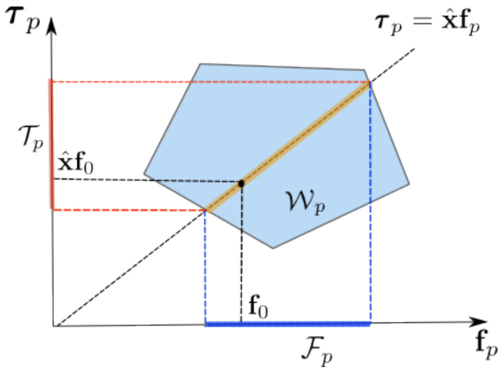


**Figure 4.3:** 2D-illustration of the admissible pseudo-wrench space $\mathcal{W}_p$ (the blue polygon). For a given position $\mathbf{x}$ of the mass point, there is the relation $\boldsymbol{\tau}_p = \hat{\mathbf{x}}\,\mathbf{f}_p$. Thus the actual admissible pseudo-wrench space (brown) is the intersection of $\mathcal{W}_p$ and the hyperplane defined by $\boldsymbol{\tau}_p = \hat{\mathbf{x}}\,\mathbf{f}_p$. Accordingly, the admissible pseudo-force space $\mathcal{F}_p$ (blue) and admissible pseudo-moment space $\mathcal{T}_p$ (red) are projections of this intersection.

### 4.2.8 Comparison with other balance criteria

In this section, we compare this criterion with several traditional ones. In the static case, we compare this criterion with the projection of CoM; in the dynamic case, we compare it with the ZMP criterion.

#### Projection of CoM

In the static case, the model has no acceleration. So the pseudo-force becomes:

$$\mathbf{f}_p = -m\mathbf{g} \tag{4.40}$$

When the model is in static balance, our criterion is expressed as:

$$\mathbf{H}(\mathbf{x})\mathbf{w}_p = -(\mathbf{H}_1 + \mathbf{H}_2\hat{\mathbf{x}})m\mathbf{g} \leq \mathbf{h}$$

where

- $\mathbf{x}$ is the position of the CoM in the 3D space;
- matrix $\mathbf{H}(\mathbf{x})$ and vector $\mathbf{h}$ represent the admissible pseudo-wrench space.

The equation above can be rewritten as:

$$-\mathbf{H}_2\hat{\mathbf{x}}m\mathbf{g} \leq \mathbf{h} + \mathbf{H}_1 m\mathbf{g}$$

In this equation (let $a_g = 9.81$):

$$\hat{\mathbf{x}}m\mathbf{g} = m\begin{bmatrix} 0 & -z_{com} & y_{com} \\ z_{com} & 0 & -x_{com} \\ -y_{com} & x_{com} & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ -a_g \end{bmatrix}$$

$$= m \begin{bmatrix} -a_g y_{com} \\ a_g x_{com} \\ 0 \end{bmatrix} = m \begin{bmatrix} 0 & -a_g \\ a_g & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_{com} \\ y_{com} \end{bmatrix}$$

So the criterion becomes a constraint over the projection of the centre of mass on the ground:

$$-m\mathbf{H}_2 \begin{bmatrix} 0 & -a_g \\ a_g & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_{com} \\ y_{com} \end{bmatrix} \leq \mathbf{h} + \mathbf{H}_1 m\mathbf{g} \tag{4.41}$$

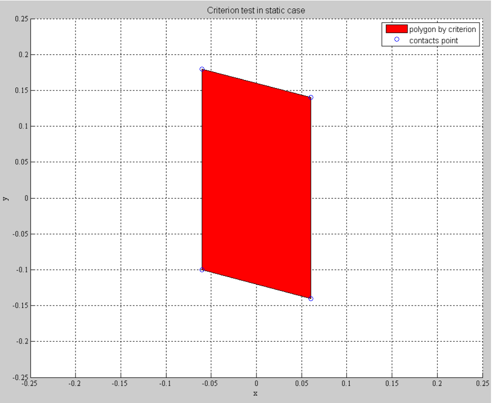


**Figure 4.4:** Simulation result: admissible polygon for the projection of CoM using the admissible pseudo-wrench criterion.

Equation (4.41) defines a polygon on the ground in which the projection of CoM should be located. We have tested the criterion in this case by simulations in Matlab. Here we present one of the simulation trials. In this simulation, a point mass of $60kg$ stays on the ground via 4 contact points $\mathbf{r}_1 = [0.06, 0.14, 0]^t$ , $\mathbf{r}_2 = [0.06, -0.14, 0]^t$, $\mathbf{r}_3 = [-0.06, -0.1, 0]^t$ and $\mathbf{r}_4 = [-0.06, 0.18, 0]^t$. The coefficient of friction is set to be 0.7. We suppose that there is no the external moments applied at contacts. The polygon defined in Equation (4.41) is shown in Fig. 4.4. We can see that this polygon coincides with the supporting polygon defined by the contact points.

### ZMP

In this section, we want to compare the admissible pseudo-wrench criterion with the ZMP criterion in a dynamic case.

We recall here the ZMP criterion: the system is in dynamical balance if the ZMP is inside the supporting polygon; when the system loses balance, the ZMP is on the boundary of the supporting polygon. The ZMP criterion is valid with some assumptions:

1. all the contacts are planar;
2. the friction is large enough so that there is no risk of slipping.

Let's denote the position of CoM

$$\mathbf{x} = [x_{com}, y_{com}, z_{com}]^t$$

and its acceleration

$$\ddot{\mathbf{x}} = [a_x, a_y, a_z]^t.$$

Suppose that the ZMP is located at (in GCS):

$$\mathbf{r}_{zmp} = [x_{zmp}, y_{zmp}, 0]^t.$$

The net moment of all the ground reaction forces about the ZMP is:

$$\mathbf{M}_{zmp} = \sum (\hat{\mathbf{r}}_i - \mathbf{r}_{zmp})\mathbf{f}_i = \begin{bmatrix} 0 \\ 0 \\ \tau_v \end{bmatrix} \tag{4.42}$$

where $\tau_v$ is the net moment around the vertical direction.

The position of ZMP can be computed ($a_g = 9.81$):

$$x_{zmp} = x_{com} - z_{com}\frac{a_x}{(a_z + a_g)} \tag{4.43}$$

$$y_{zmp} = y_{com} - z_{com}\frac{a_y}{(a_z + a_g)} \tag{4.44}$$

Our balance criterion is expressed as:

$$(\mathbf{H}_1 + \mathbf{H}_2\hat{\mathbf{x}})\mathbf{f}_p = (\mathbf{H}_1 + \mathbf{H}_2\hat{\mathbf{x}})m(\ddot{\mathbf{x}} - \mathbf{g}) \leq \mathbf{h} \tag{4.45}$$

In Equation (4.45), $\mathbf{H}_1$ and $\mathbf{H}_2$ depend only on the positions of contact points: they are constant if the model does not change its contacts. For a given position of CoM, Equation (4.45) defines the admissible space of its acceleration. This space can be further transformed to the admissible polygon for ZMP using Equation (4.43) and Equation (4.44).

Here we present a Matlab simulation. In this simulation, the mass point has a mass of $60kg$ and has 4 planar contacts with the ground at points $\mathbf{r}_1 = [0.06, 0.18, 0]^t$, $\mathbf{r}_2 = [0.06, -0.1, 0]^t$, $\mathbf{r}_3 = [-0.06, -0.06, 0]^t$ and $\mathbf{r}_4 = [-0.06, 0.22, 0]^t$. There is no external moment acting at contacts. In order to compare with the ZMP formulation based on the LIPM ([KKK+03]), we suppose that there is no acceleration in the vertical direction ($a_z = 0$). Different coefficients of friction are tested in the simulations. We set $x_{com} = 0$ and $y_{com} = 0$. And we test two different heights of CoM: $z_{com} = 0.8$ and $z_{com} = 0.3$. In Fig. 4.5, we show the results of the admissible polygon for the ZMP in different parameter settings.

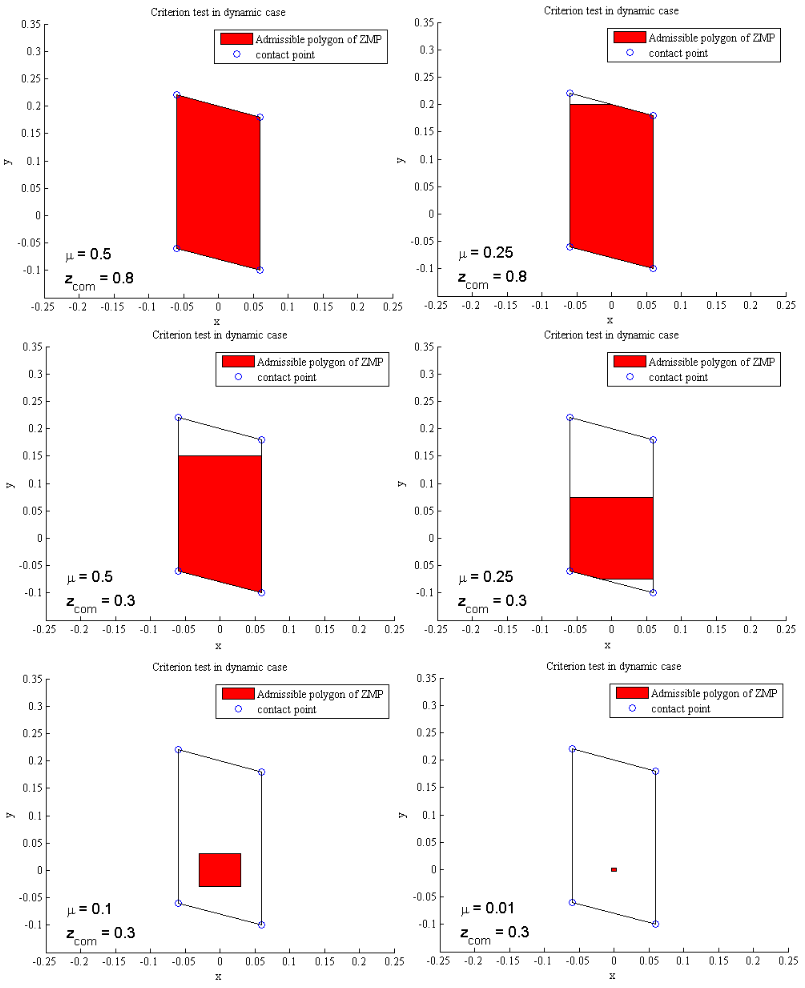


**Figure 4.5:** Admissible ZMP polygons computed from the criterion of admissible pseudo-force space.

### Remarks on the comparisons

From the simulation results in this section, we make the following remarks on the balance criterion of admissible pseudo-wrench space:

1. When only planar contacts exist, it gives the admissible polygon for horizontal components of CoM which is the same as the support polygon defined using the projection of CoM; it has advantage over projection of CoM since it can also treat non-planar contacts as well as grasps;
2. The admissible pseudo-wrench space criterion is equivalent to the ZMP criterion when the friction is large enough: the admissible ZMP polygon coincides with the supporting polygon defined by the contact points;
3. When the coefficient of friction is taken into consideration, the admissible pseudo-wrench space criterion defines more precise balance constraints than ZMP criterion.

## 4.2.9 Stability margin

The concept of "stability margin" indicates the maximum magnitude for all possible perturbations that the system can resist while keeping dynamic balance.

### Residual ball radius

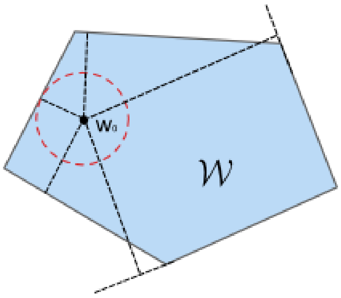


**Figure 4.6:** Illustrating the distances for pseudo-wrench point $\mathbf{w}_0$ from facets of its admissible space $\mathcal{W}$. The smallest distance is then the stability margin.

In [BB08], the stability margin is represented by the "residual ball radius" which is computed by solving a linear optimization problem, as presented in Algorithm 1. In this thesis, we study the stability margin by evaluating firstly the margins of a pseudo-wrench point from all facets of its admissible space (Fig. 4.6). Algorithm 3 presents the function "DISTANCES" for calculating the distances from a given point $\mathbf{w}_0$ to all the facets of a convex point set $\mathcal{W}$ that is represented by a matrix $\mathbf{H} = \{H_{ij}\}$ and a vector $\mathbf{h} = \{h_i\}$. If point $\mathbf{w}_0$ is inside $\mathcal{W}$, this function return a distance vector $\mathbf{d}$ and the stability margin is the smallest element of $\mathbf{d}$. If $\mathbf{w}_0$ is not inside $\mathcal{W}$, function "DISTANCES" will return 0 thus there is no margin for this point.

**Algorithm 3** $\mathbf{d} = DISTANCES(\mathbf{w}_0, \mathbf{H}, \mathbf{h})$

**Input:** a point $\mathbf{w}_0$ and $[\mathbf{H}, \mathbf{h}]$ for a convex point set $\mathcal{W}$

**Output:** $\mathbf{d} = \{d_i\}$ where $d_i$ is the distance from $\mathbf{w}_0$ to the $i$-th facet of $\mathcal{W}$

```
if $\mathbf{H}^* \mathbf{w_0} > \mathbf{h}$ then
    return 0
end if
$m = \mathbf{H}$.rows
$n = \mathbf{H}$.cols
for $i := 1$ to $m$ do
    for $j := 1$ to $n$ do
        $\delta_i = 0$
        $\delta_i \mathrel{+}= \mathbf{H}[i, j]^2$
    end for
    $\mathbf{d}[i] = \frac{1}{sqrt(\delta_i)}(\mathbf{h}[i] - \mathbf{H}[i, :]\mathbf{w}_0)$
end for
return $\mathbf{d}$
```

### Stability margins associated with the balance criterion

Since the balance criterion concerns the pseudo-wrench point and its admissible space. Thus the stability margin can be computed by finding the “residual ball radius” for the pseudo-wrench point.

Let’s suppose that there is a force of perturbation $\mathbf{f}_{pert}$ applied on the CoM and a moment of perturbation $\boldsymbol{\tau}_{pert}$ (Fig. 4.7). The dynamic equations becomes:

$$\mathbf{w}_p = \begin{bmatrix} \mathbf{f}_p - \mathbf{f}_{pert} \\ \hat{\mathbf{x}}(\mathbf{f}_p - \mathbf{f}_{pert}) - \boldsymbol{\tau}_{pert} \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{C} \end{bmatrix} \mathbf{f} \tag{4.46}$$

Now, in three cases, we analyze the stability margin (see Fig. 4.8) for a given pseudo-wrench $\mathbf{w}_0 = \begin{bmatrix} \mathbf{f}_0 \\ \hat{\mathbf{p}}\mathbf{f}_0 \end{bmatrix}$ when facing different exterior perturbations.

**Case 1**:

If there is no perturbation moment and there exits a force of perturbation acting on the mass point. In this case, the stability margin is the value of the minimum perturbation force that can move the pseudo-wrench point outside the admissible pseudo-wrench space. This margin then is the residual ball radius of $\mathbf{f}_0$ in the

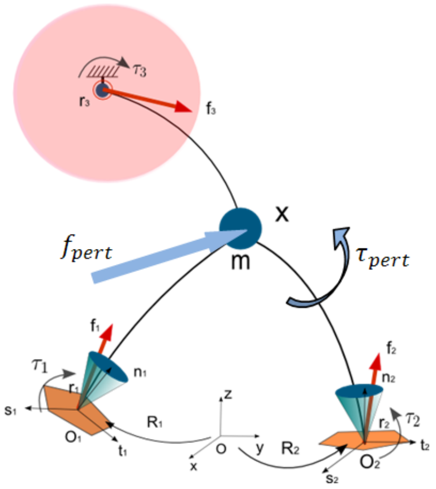


**Figure 4.7:** Illustrating force of perturbation acting at the CoM and the moment of perturbation acting on the model.

admissible pseudo-force space:

$$M_{s1} = min(DISTANCES(\mathbf{f}_0, \mathcal{F}_p))$$

**Case 2**:

If there exists only the perturbation moment and there is no perturbation force acting on the mass point. The stability margin then is the value of the minimum perturbation moment that moves the pseudo-wrench point outside the admissible pseudo-wrench space. In the same manner as case 1, the margin in the case 2 is:

$$M_{s2} = min(DISTANCES(\hat{\mathbf{p}}\mathbf{f}_0, \mathcal{T}_p))$$

**Case 3**:

When both the force and the moment of perturbation exist, the pseudo-wrench point is displaced by the perturbation wrench $\mathbf{w}_{pert} = \begin{bmatrix} \mathbf{f}_{pert} \\ \hat{\mathbf{x}}\mathbf{f}_{pert} + \boldsymbol{\tau}_{pert} \end{bmatrix}$. In this case, the smallest $\mathbf{w}_{pert}$ that drives the pseudo-wrench point outside the admissible space is the stability margin:

$$M_{s3} = min(DISTANCES(w_0, \mathcal{W}))$$

### Some remarks

1. The units of the margins in the case 1 and the case 2 are respectively $N$ and $N \cdot m$. In the case 3, we introduce an unit distance $\alpha = 1m$ to change

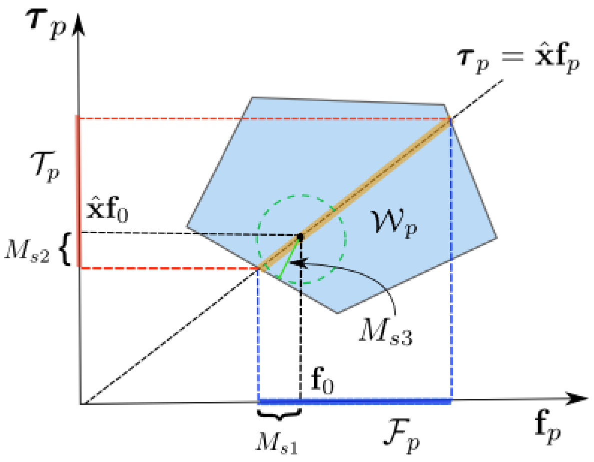


**Figure 4.8:** A 2D illustration of three stability margins. $M_{s1}$: for a pure force of perturbation; $M_{s2}$: for pure moment of perturbation; $M_{s3}$: force and moment of perturbation coexist.

the wrench of perturbation as $\mathbf{w}_{pert} = \begin{bmatrix} \alpha\mathbf{f}_{pert} \\ \hat{\mathbf{x}}\mathbf{f}_{pert} + \boldsymbol{\tau}_{pert} \end{bmatrix}$. Thus the stability margin in case 3 has also the unit $N \cdot m$.

2. The stability margin in case 3 depends on both the reference point for Equation (4.2) and the mass of the model. When we need to compare motions in different scenarii, it is recommended to standardize the stability margin by choosing the CoM as reference point in Equation (4.2).

## 4.3 Extension of the criterion for a whole-body model

The criterion formulated in Section 4.2 can be further extended to take into account the postural factors in the modeling, namely to deal with a model consisting of more than one body. Suppose that the DHM model is formed by $n_b$ bodies that are linked with revolute joints, then the angular momentum of the whole-body model is expressed as:

$$\mathbf{L} = \sum_{j=1}^{n_b} \mathbf{L}_j \tag{4.47}$$

where the $j$-th body has its angular momentum:

$$\mathbf{L}_j = \hat{\mathbf{c}}_j m_j \mathbf{v}_j + \mathbf{R}_j \mathbf{I}_j \mathbf{R}_j^t \boldsymbol{\omega}_j \tag{4.48}$$

where

- $\mathbf{c}_j$, $\mathbf{v}_j$, $\mathbf{R}_j$ and $\boldsymbol{\omega}_j$ are respectively the CoM, velocity, orientation and angular velocity of the $j$-th body with respect to the global frame;

- $m_j$ and $\mathbf{I}_j$ are respectively the mass and the inertial matrix of the $j$-th body.

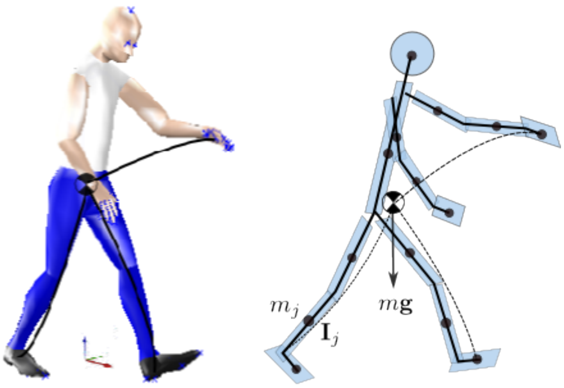


**Figure 4.9:** Illustrating the whole-body model for the extended balance criterion. In this model, each body has its mass and inertia. As shown in the figure, the $j$-th body has its mass $m_j$ and inertia $\mathbf{I}_j$.

The dynamic equation – Equation (4.2) becomes:

$$\dot{\mathbf{L}} = \sum_{i=1}^{n+k} (\hat{\mathbf{r}}_i \mathbf{f}_i + \boldsymbol{\tau}_i) + \hat{\mathbf{x}} m\mathbf{g} \tag{4.49}$$

The admissible pseudo-wrench space is still:

$$\mathcal{W}_p = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \mathbf{A} \end{bmatrix} \boldsymbol{\psi} \tag{4.50}$$

so the balance criterion becomes in this case:

$$\mathbf{H} \begin{bmatrix} m\ddot{\mathbf{x}} - m\mathbf{g} \\ \dot{\mathbf{L}} - \hat{\mathbf{x}} m\mathbf{g} \end{bmatrix} \leq \mathbf{h} \tag{4.51}$$

Equation (4.51) is in fact a more general and precise balance criterion regarding the criterion formulated based on a simplified model, since both the constraints on exterior wrenches applied at supports and the postural balance are taken into account.

## 4.4 Validation and discussion

In this section, we will use the balance criteria formulated in previous sections to evaluate dynamic balance in some recorded motions. Based on both models (simplified DHM model and the whole-body model), we compute their angular momenta and the according stability margins. Based on the comparison between the two models, we study the influence of posture variations on dynamic balance during human motions. We make finally in this section some discussions and hypotheses for our following work.

### 4.4.1 Dynamic balance evaluation

The changes of human-environment interactions occur at key frames, while the admissible pseudo-wrench space depends uniquely on the support configuration. Thus, each transition phase is associated with a constant admissible pseudo-wrench space. In this section, we evaluate dynamic balance of human subject in a recorded motion in the following steps:

1. estimating supports at each key frame;
2. computing a sequence of polytopes (admissible pseudo-wrench spaces) relying on the sequence of support configurations;
3. computing for each frame the pseudo-wrench and tracing the stability margins.

#### Admissible spaces

We make the following hypotheses for calculating the balance criteria for evaluating dynamic balance of human subjects in recorded motions:

- a foot contact is modeled as four contact points (rectangle foot sole) at which the contact forces are applied by the contact surface;
- a hand grasp is built at a fixed point with a constant limit on force magnitude.

With the above-mentioned hypotheses, we can define a sequence of support configurations and compute the corresponding admissible pseudo-wrench spaces using the experimental data (extracted feet and hands trajectories) at key frames. Admissible pseudo-force spaces can be computed using Equation (4.38) since CoM trajectory has been estimated beforehand.

Fig. 4.10 displays an example of the admissible pseudo-force spaces calculated for the first two transition phases in the motion $01\_TR\_H1C1S2\_1$ (some motion clips and the foot contacts can be seen in Fig. 3.10). The support configuration change (left foot lift) occurs at key frame 158. We can notice that the volume of polytope decreases greatly (from the red to the blue one) because of the removal of left foot contact. This decrease occurs in x-direction, since the two feet are located initially along this direction. Due to this decrease, the pseudo-force $\mathbf{f}_p$ becomes very near to the boundary of the polytope, meaning that the human takes a much larger risk of losing balance.

#### Stability margins

Since the subject is in balance during all his real motions, the pseudo-wrench (or pseudo-force) point should be inside the corresponding admissible spaces at any frame if there is no modeling and estimation errors. In other words, the stability margins should be always strictly positive.

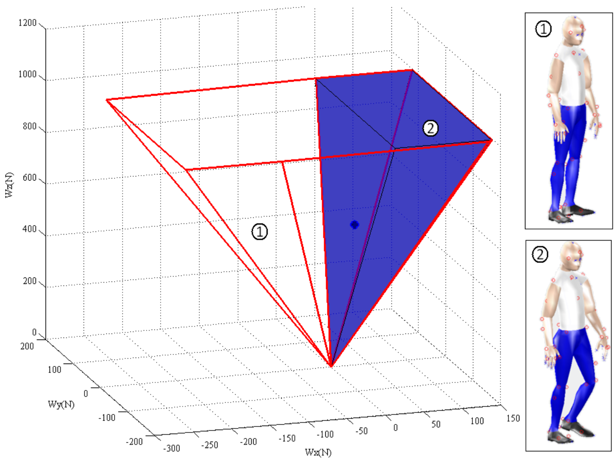


**Figure 4.10:** Two 3D polytopes representing the admissible pseudo-force spaces: before (red) and after (blue) the lift of left foot (at 1.58s). The blue polytope is inside the red one. The solid point is the pseudo-force $\mathbf{f}_p$ around $1.58s$.

We show here the stability margin evaluation result for the example motion $01\_TR\_H1C1S2\_1$ shown in Fig. 3.10. The evolution of the stability margin in admissible pseudo-force space is shown in Fig. 4.11. We can see that the stability margins are always greater than zero, meaning that the balance criterion is respected at all the frames. There are discontinuities at key frames since the changes of support configurations occur at these frames. The lift of the left foot at 1.58s brings a great risk of losing balance, causing the stability margin fall nearly to zero. The hand grasp at $2.4s$ enhances significantly the balance safety.

We have observed similar results for other recorded motions in our experimental database. We have also evaluated dynamic balance of human subject in sit-to-stand motions in the same way and the results are presented in [RQC$^+$11].

### 4.4.2 Stability margins based on two models

In previous section, we have extended the balance criterion to a whole-body dynamic model by taking into account the postural factor in computation of the whole-body angular momentum. In this section, we want to study the influence of the postural effect on the dynamic balance.

#### Angular momentum comparison

For a simple mass point model, its angular momentum (about origin point, same for the following) is:

$$\mathbf{L}_s = \hat{\mathbf{x}} m \dot{\mathbf{x}} \tag{4.52}$$

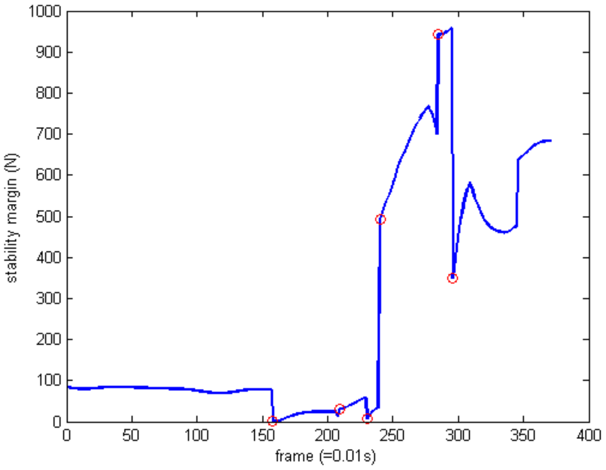


**Figure 4.11:** Evolution of stability margin. Red circles indicate the key frames (the first and the last frames are also key frames which are not indicated in the figure).

For the whole-body model, its angular momentum can be computed from Equation (4.47) as:

$$\mathbf{L}_w = \hat{\mathbf{x}} m \dot{\mathbf{x}} + \mathbf{L}_{pos} = \mathbf{L}_s + \mathbf{L}_{pos} \tag{4.53}$$

where

- $\mathbf{L}_{pos}$ is the component of the angular moment that results from the posture variations.

We here study the influence of the posture variations on the angular momentum. In the analysis of the recorded motions, we have computed the angular momentums of each motion by using respectively Equation (4.52) and Equation (4.53). We then compared the results to show the difference between the two models.

The analysis has been carried out for all the reconstructed motions in our experiment as well as motions in other experiments. Fig. 4.12 shows as an example the analysis result of the recorded motion "$01\_TR\_H1C1L\_1$". As one can see, the angular momenta computed from two models shows small difference in most frames.

### Stability margins comparison

Following the estimations of angular momentum presented in the previous section, the rate of change of the angular momentum can then been computed, which will further be used for estimate the stability margins of the recorded motion.

For the simplified mass point model:

$$\dot{\mathbf{L}}_s = \hat{\mathbf{x}}\, m \ddot{\mathbf{x}} \tag{4.54}$$

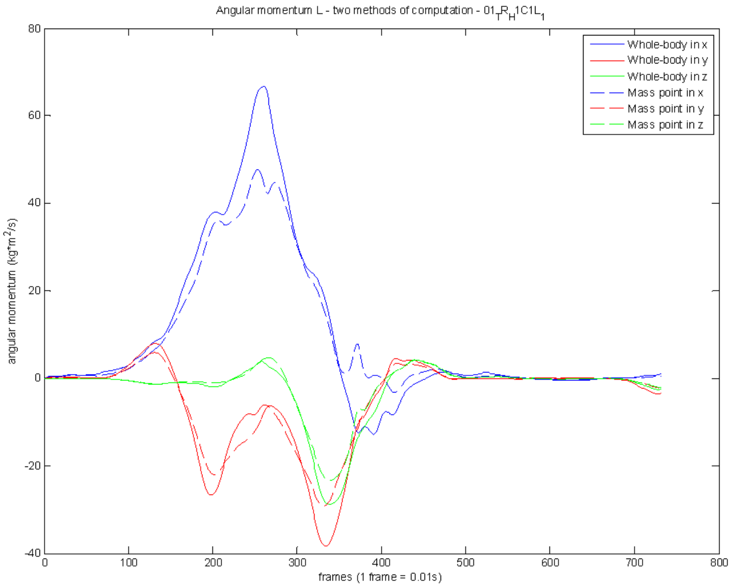


**Figure 4.12:** Comparison of the angular momentum for the movement 01_$TR$_$H1C1L$_1.

And we compute the derivative of angular momentum for the whole-body model in a discrete way:

$$\dot{\mathbf{L}}_w(k+1) = \frac{\triangle \mathbf{L}_w(k+1)}{T} = \frac{\mathbf{L}_w(k+1) - \mathbf{L}_w(k)}{T} \tag{4.55}$$

We recall that the pseudo-wrench can be expressed as:

$$\mathbf{w}_p = \begin{bmatrix} m\ddot{\mathbf{x}} - m\mathbf{g} \\ \dot{\mathbf{L}} - \hat{\mathbf{x}}\, m\mathbf{g} \end{bmatrix} \tag{4.56}$$

where

- $\dot{\mathbf{L}} = \dot{\mathbf{L}}_s$ for the simplifed model and $\dot{\mathbf{L}} = \dot{\mathbf{L}}_w$ for the whole-body model.

Inputting respectively Equation (4.54) and Equation (4.55) into Equation (4.56), we can compute pseudo-wrenches respectively based on the two different models. Accordingly, we can compute and compare the stability margins in each frame of a motion using respectively the two models.

We compare the stability margins based on two models in one of our recorded motions. The stability margins traced for the motion 01_$TR$_$H1C1S2$_1 using the two different methods are shown in Fig. 4.13. We can see in this figure that the difference between stability margins computed with two models are insignificant in most frames. There are only 14 frames in which their difference is larger than $25N.m$.

The same analysis has been carried out on other recorded motions such as the car-ingress motions, car-egress motions, sit-to-stand motions [RQC$^+$11]. And all

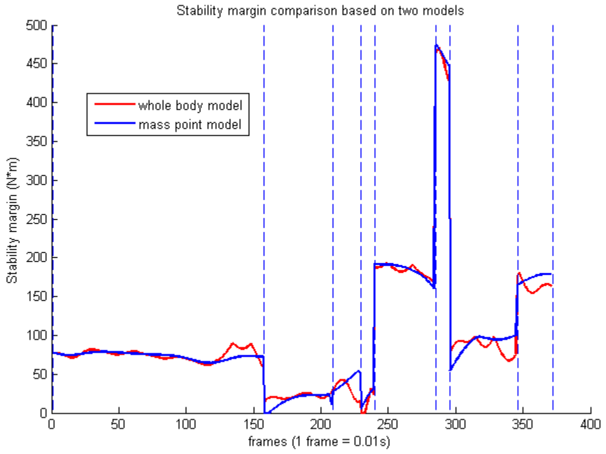


**Figure 4.13:** Comparison of stability margins for the movement $01\_TR\_H1C1S2\_1$.

the results in these analysis validate the balance criterion and show insignificant influence of posture variation on whole-body dynamics.

### Discussion

Based on the results of analysis in this section, we conclude that the postural effect is actually not important for motions in our experimentation. In following work, we can then use the simplified model to generate the dynamically stable motion for a virtual human, with a reasonable stability margin in each frame in order to tolerate errors brought by model difference.

## 4.5 Conclusion

Balance maintenance is an essential problem for DHM study. However, it is not easy to define exactly the *balance* of a virtual human, particularly in complex scenarii and cluttered environments. In this study, we firstly propose our definitions of *static balance* and *dynamic balance*. According to the objective of this study, we then propose the requirements of a general balance criterion for our following work.

Based on a simplified model of DHM, we have formulated a balance criterion by extending some existing criteria. With several pre-defined hypotheses on DHM-environment interactions, physical laws that limit the exterior wrenches applied to DHM supports, result in an *admissible pseudo-wrench space*. In order to maintain balance, the DHM must have its pseudo-wrench (it depends only on position and acceleration of CoM) located inside this space for avoiding violations of some physical conditions at supports. This criterion has been validated in simulation by comparisons with some common criteria. As a quantitative criterion, it is associated with stability margins for assessing the safety of a balance state facing disturbances.

In Section 4.3, we extend this criterion to a whole-body human model. We argue that this extended criterion is more precise than the simplified one (that is based on the simplified model) since the posture variations are also taken into account for balance verification. The simplified one, however, is much easier for motion planning and control.

In Section 4.4, we evaluate the balance of a human subject in several recorded motions using our balance criteria. Accordingly to this evaluation, the simplified DHM model is chosen for our following work.

# Mini-Chapter: framework overview

From this chapter on, we present our methodology for simulating DHM motions in cluttered environments. A hierarchical framework [QEMR12] developed in this thesis for realizing entirely the motion simulation task is presented in detail in Chapter 5, Chapter 6 and Chapter 7. We give in this mini-chapter an overview of this framework and the problematics involved at each level.

The global scheme of this hierarchical framework is shown in Fig. 4.14. After that some preliminary data on the structured environment and the sequence of supports configurations have been specified, this framework then can achieve step by step the DHM motion simulation which has been decomposed into three levels:

1. At the global level, with the pre-defining sequence of support configurations, a global CoM trajectory with timing information is firstly generated by an optimization based method. The DHM maintains its dynamic balance during its motion by tracking strictly this trajectory;
2. At the local level, whole-body collision-free motion is generated piecewisely for all the transition phases between stances. A local sampling-based method is associated with a flying end-effector (a hand or a foot) for locally planning its trajectory;
3. At the execution level, the generated trajectories (CoM, end-effector, joints) serve as control references so that the dynamic DHM can realize the motion by virtue of dynamic controllers.

We have identified the following problematics at each level of the framework:

- The generation of the global CoM trajectory at the first level requires the trajectory parameterization, balance, geometric feasibility and some given conditions for the trajectory. Moreover, since the motion is expected to be dynamic, the motion timing problem is also studied;
- Whole-body motion planning will concern a series of problems including collision detection and avoidance, inverse kinematics, efficient motion planning;
- The execution of the planned motions concerns the application of dynamic controllers. The multi-objective control problem is dealt with in this study.

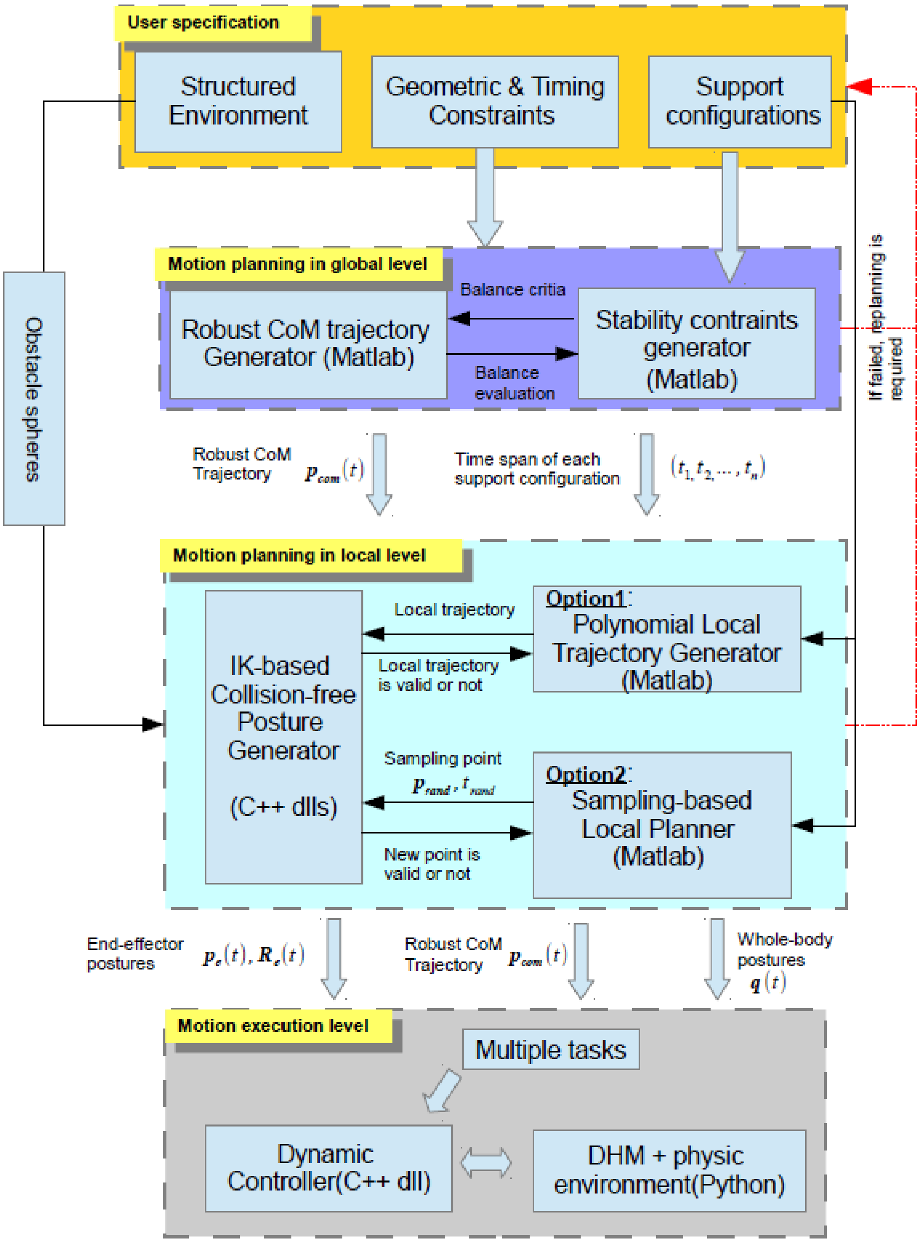


**Figure 4.14:** The global scheme for DHM motion planning and execution.

## Chapter 5

# Global CoM trajectory computation

In this chapter we explore methods at the first level of our framework for computing a global CoM trajectory of the DHM that is aimed to ensure DHM balance during its final motion in a cluttered environment.

We have explored two approaches for generating the global CoM trajectory. In both methods the CoM trajectory is parameterized as a high order 3D B-spline. Velocity, acceleration and jerk of the CoM are accordingly represented by derivatives of this spline which are also B-splines. With the hypothesis that a sequence of support configurations have been defined or generated beforehand, a sequence of polytopes (admissible pseudo-wrench spaces) can be firstly computed for defining balance constraints in our methods.

In the first method, we impose some constraints at "key frames" (an instant at which the support configuration changes, see Chapter 3) to the spline of the CoM trajectory. These imposed constraints are extracted from experimental data. At each key frame, the CoM trajectory is imposed to pass the corresponding CoM position of an real motion recorded in experimentation. Balance constraints are imposed at some sampled instants. The spline parameters are then solved by optimization.

In the second method, experimental data are no longer used: key instants of support change (motion timing) are variables and CoM positions are imposed only at the initial and final frames. An objective of maximizing the stability margin in this method makes the computed CoM trajectory robust for resisting perturbations and tolerating tracking errors at the motion execution level.

The general methods explored in this chapter can generate CoM trajectory for not only DHMs but also other multi-limbed models. Both the two methods are implemented and validated with DHM or legged robot in several scenarii.

## 5.1 State of the art

In this section, we give a state of the art on stable motion generation for DHMs or legged robots.

### 5.1.1 Quasi-static motion generation

To maintain balance during its quasi-static locomotion, a DHM or legged robot must have projection of its CoM inside the support polygon (if only horizontal and coplanar contacts exist) or a feasible region (as described in [BL08] in more general cases). Quasi-static motion are usually generated by computing a sequence of stable postures. The free-climbing motion of a multi-legged robot is studied in [Bre06]. Static equilibrium in this paper is guaranteed by avoiding slipping at each contact point, resulting in a constraint over the CoM position for each stance. The quasi-static motion is realized by finding the sequence of stances and the transitions between them via a two-stage searching approach. In [EKMG09], quasi-static motions of humanoid robot HRP-2 are realized by searching a sequence of contact-sets. A series of stable postures are computed by a Posture Generator. Humanoid motions in complex scenarii such as going under a table and sitting into (getting out of) a chair are successfully realized.

### 5.1.2 Dynamic motion generation

With respect to quasi-static motions, dynamic motions should also satisfy time-related requirements such as dynamic constraints or velocity/acceleration limits. Maintaining dynamic balance makes dynamic motion generation much more complex. Simplified models and motion controllers are usually used for generating dynamic motions.

Much recent work has focused on reference trajectory tracking for realizing dynamic motions. Among these works, biped walking locomotion has been most broadly studied. To maintain dynamic balance during motions, ZMP criterion is adopted in most of the references with some assumptions such as neglecting inertial effect due to posture changes. A walking pattern generating method proposed in [KKK$^+$03], based on the LIPM (Linear Inverted Pendulum Model) and the ZMP criterion, can generate a reference ZMP trajectory for a biped robot. The ZMP trajectory can be easily transformed into the CoM trajectory that can guide the biped's motion with help of preview control. Walking pattern based method has been used in several studies [SN05, HYK$^+$01] for dynamic motion generation and shows good results. Park et al. propose a method in [PR98] for generating a desired ZMP trajectory determined by a fuzzy logic. Eubatur proposes in [EK09] an algorithm for computing ZMP trajectories based on the LIPM which can simulate biped walking motions with improved naturalness. Some studies focus on computing joint angles for dynamic biped motions. Lengagne presents a method for computing the safe motion in a kicking scenario of a humanoid robot [LFR09]. In this method, the motion is generated by solving a Semi-Infinite Problem (SIP): 12 trajectories for joints of the two lower limbs are parametrized as B-splines, and the inequality constraints including the ZMP constraint and physical limits are imposed at discretized time intervals. Optimal joint trajectories are solved and successfully implemented on the humanoid robot HRP-2 to achieve the kicking motion.

Instead of the above-mentioned works on reference trajectory computing, dynamic biped locomotions are realized using physics-based controllers in some researches, particularly in the domains of computer graphics and animation. A physics-based locomotion controller based on a full-body 3D character is proposed in [MdLH10]. A simplified Spring-Load Inverted Pendulum (SLIP) model [K+99] is used to describe the moving character's motion state and to model the stance dynamics. The motion is divided into a sequence of phases. Individual low-dimensional trajectories are pre-computed sequentially. Balance constraints are defined using CoP criterion. The controller carries out optimization for each time step and generates the torques for actuating the motion of the character. Animations of locomotions on challenging terrains have been realized in their work.

### 5.1.3 Discussion

ZMP-based reference trajectory generation methods are only valid for locomotions on horizontal or slightly sloped terrains. The physics-based controllers can realize very interesting locomotions, however, they take into account little about the interactions between the biped model and the environment.

Generic types of DHM-environment interactions are considered in our study, thus we intend to generate CoM reference trajectories based on a general balance criterion, namely the one formulated in Chapter 4. Same like most of the above-mentioned studies, we generate the reference trajectory based on a simplified human model. Only off-line trajectory generation methods are explored in our study since reference generation and motion control are separated in different steps.

## 5.2 Trajectory representation

CoM trajectory we need to compute is expressed in function of time $t$ as:

$$traj := \mathbf{p}(t) = [x(t), y(t), z(t)]^t, t \in [t_{ini}, t_{fin}] \tag{5.1}$$

In our methods, CoM trajectory is represented by a $k$-th order 3D B-spline ($k$ is a positive integer). The 3D B-spline can be regarded as three 1D B-splines in respectively x, y, and z directions that share the same knots and with the same degree. Moreover, the spline is defined at a normalized time interval:

$$\mathbf{p}_s(\bar{t}) = [x_s(\bar{t}), y_s(\bar{t}), z_s(\bar{t})]^t, \bar{t} \in [0, 1] \tag{5.2}$$

### 5.2.1 Mapping trajectory to B-Spline

The trajectory is mapped to the 3D B-spline by a time scaling parameter or time ratio (see Fig. 5.1):

$$\sigma = \frac{1}{t_{fin} - t_{ini}} \tag{5.3}$$

We suppose that $t_{ini} = 0$, thus

$$\sigma = \frac{1}{t_{fin}} \tag{5.4}$$

The time ratio maps the trajectory to the spline as:

$$\mathbf{p}(t) = \mathbf{p}_s(\bar{t}), \bar{t} = \sigma t \tag{5.5}$$

The $d$-th $(1 \leq d \leq k)$ derivative of the CoM trajectory is scaled as:

$$\mathbf{p}^d(t) = \sigma^d \mathbf{p}_s^d(\bar{t}), \bar{t} = \sigma t \tag{5.6}$$

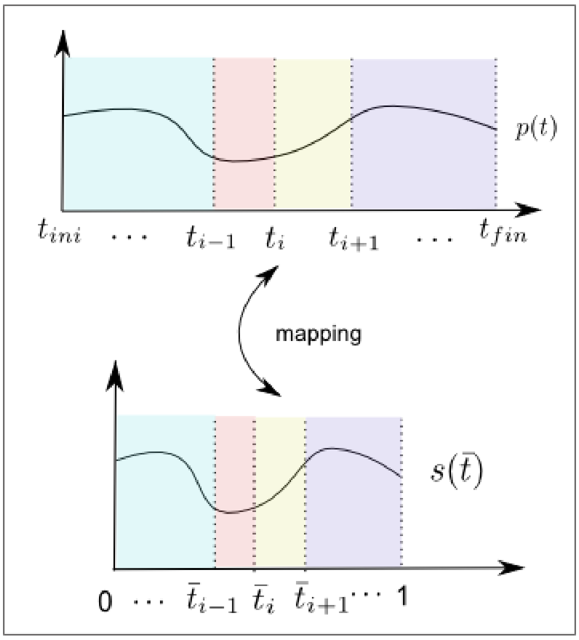


**Figure 5.1:** Mapping between the trajectory and its representing spline

Once the B-spline trajectory and its derivatives are solved, we can compute the CoM trajectory and its derivative curves using Equations (5.5) and (5.6).

### 5.2.2 B-Spline parameterization

The B-spline trajectory is chosen to have $n$+1 control points and to have a knot vector $\mathbf{T} = [t_0, t_1, \cdots, t_m]$ including $m$+1 ($m$=$n$+$k$+1) knots with the relation: $t_0 \leq t_1 \cdots \leq t_m$. To make it possible to impose end conditions of the trajectory, we choose the B-spline to be *clamped* (see Section 2.4.1).

Each 1D spline (in $x$, $y$ or $z$ direction) thus has the following expression:

$$s(\bar{t}) = \sum_{i=0}^{n} N_{i,k}(\bar{t}) a_{si}, \bar{t} \in [0, 1] \tag{5.7}$$

where

- $N_{i,k}$: are the basis functions of the spline;
- $a_{si}$: is the $i$-th control point of the spline.

The 1D B-spline in Equation (5.7) can be rewritten as:

$$\mathbf{s}(\bar{t}) = \mathbf{V}_{N_k}(\bar{t})\mathbf{a}_s, \bar{t} \in [0,1] \tag{5.8}$$

where:

- $\mathbf{V}_{N_k}(\bar{t}) = [N_{0,k}(\bar{t}), \cdots, N_{n,k}(\bar{t})]^t$;
- $\mathbf{a}_s = [a_{s0}, \cdots, a_{sn}]^t$.

The 3D B-spline trajectory is expressed as:

$$\begin{aligned} \mathbf{p}_s(\bar{t}) &= [x_s(\bar{t}), y_s(\bar{t}), z_s(\bar{t})]^t \\ &= diag(\mathbf{V}_{N_k}(\bar{t}))[\mathbf{a}_x, \mathbf{a}_y, \mathbf{a}_z]^t, \bar{t} \in [0,1] \end{aligned} \tag{5.9}$$

### 5.2.3 Derivative splines representation

The $d$-th derivative of the 1D B-spline in Equation (5.7) is also a B-spline which can be computed as:

$$\begin{aligned} \mathbf{s}^d(\bar{t}) &= \sum_{i=0}^{n-d} N_{i,k-d}(\bar{t}) a_{si}^d \\ &= \mathbf{V}_{N_{k-d}}(\bar{t})\mathbf{a}_s^d, \ \bar{t} \in [0,1] \end{aligned} \tag{5.10}$$

where

- $\mathbf{V}_{N_{k-d}}(\bar{t}) = [N_{0,k-d}(\bar{t}), \cdots, N_{n-d,k-d}(\bar{t})]$: are the basis functions of the derivative spline;
- $\mathbf{a}_s^d = [a_{s0}^d, \cdots, a_{s(n-d)}^d]^t$: are control points of the derivative spline which can be computed directly by original control points ($\mathbf{C}_d$ is a constant matrix):

$$\mathbf{a}_s^d = \mathbf{C}_d \mathbf{a}_s \tag{5.11}$$

The $d$-th derivative of the 3D B-spline in Equation (5.9) can be expressed as:

$$\begin{aligned} \mathbf{p}_s^d(\bar{t}) &= [x_s^d(\bar{t}), y_s^d(\bar{t}), z_s^d(\bar{t})]^t \\ &= diag(\mathbf{V}_{N_{k-d}}(\bar{t}))[\mathbf{a}_x^d, \mathbf{a}_y^d, \mathbf{a}_z^d]^t, \bar{t} \in [0,1] \end{aligned} \tag{5.12}$$

## 5.3 Keyframe-based method

In the first step, we have explored a method for generating CoM trajectory based on a small amount of experimental data. Based on a given recorded motion, this method can generate a trajectory that has the same duration as in the recorded motion and passes the real CoM positions at key frames of the recorded motion. Balance constraints are imposed as well at some frames to ensure DHM's balance during the motion. The CoM trajectory in this method is computed by solving an optimization problem.

### 5.3.1 Problem statement

The 3D B-spline representing the CoM trajectory is discretized at a sampling time $T = 0.01s$ which is the same as in MoCap experiments. To represent a motion with $l$ frames, we denote the CoM position at time $jT$ as $\mathbf{p}(j)$ with $0 \le j < l$.

Suppose the recorded motion has $n_k$ key frames:

$$\mathbf{k}_{kf} = [k_1, k_2, \cdots, k_{n_k}]$$

and the CoM position of the subject in this motion at key frame $k_i$ ($k_i \in \mathbf{k}_{kf}$) is $\mathbf{p}^{rec}(k_i)$.

Using the support configuration sequence that is extracted from the recorded motion, the admissible pseudo-wrench spaces $\mathcal{W}_{pi}$ are pre-computed and stored using their H-representation parameters:

$$\{\mathbf{H}_i, \mathbf{h}_i\}, 1 \le i \le n_k$$

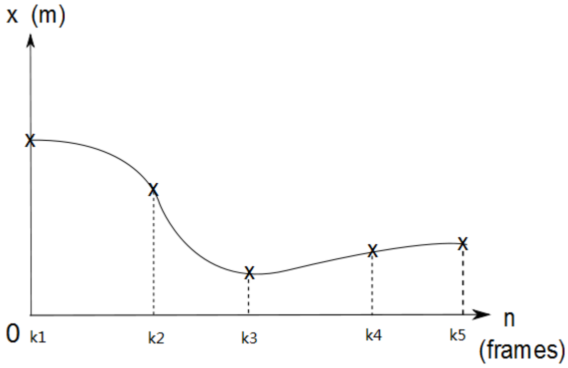


**Figure 5.2:** Example of a spline passing through given positions at key frames. $n$ is the number of frames. k$i$ is the $i$-th key frame.

In this method, we aim at generating a trajectory that has the same duration as in the recorded motion, passes through the real CoM positions at key frames of this motion and respects the balance constraint at each instant. Moreover, in order to obtain a smooth and natural-looking trajectory, we choose objectives of minimizing velocity and jerk. This method then can be stated as the following optimization problem:

$$\begin{aligned}
&for:\ 0 \le j \le l \\
&\qquad\qquad min(\|\dot{\mathbf{p}}(j)\|)\ and\ min(\|\dddot{\mathbf{p}}(j)\|) \\
&\quad subject\ to: \\
&\qquad\qquad \mathbf{p}(k_i) = \mathbf{p}^{rec}(k_i),\ \forall k_i \in \mathbf{k}_{kf} \\
&\qquad\qquad for:\ 1 \le i \le n_k - 1\ and\ for: k_i \le j < k_{i+1} \\
&\qquad\qquad\qquad \mathbf{H}_i \mathbf{w}_p(j) < \mathbf{h}_i
\end{aligned}$$

### 5.3.2 Optimization

#### Vector to optimize

In this method, the vector to optimize is chosen as:

$$\mathbf{u} = [\mathbf{a}_x, \mathbf{a}_y, \mathbf{a}_z]^t \tag{5.13}$$

which includes the control points for splines of the x, y and z direction.

The spline $\mathbf{p}(j)$ as well as its derivatives $\mathbf{p}^d(j)$ are functions of $\mathbf{u}$.

#### Objectives

Minimization of velocity and jerk is implemented for the whole motion:

$$obj = \sum_{j=1}^{l} \dot{\mathbf{p}}(j)^t \mathbf{Q}_v \dot{\mathbf{p}}(j) + \sum_{j=1}^{l} \dddot{\mathbf{p}}(j)^t \mathbf{Q}_j \dddot{\mathbf{p}}(j)$$

where $\mathbf{Q}_v$ and $\mathbf{Q}_j$ are the weighting matrix.

#### Constraints

CoM position constraint:

$$forj = k_i,\ \mathbf{p}(j) = \mathbf{p}^{rec}(k_i),\ \forall k_i \in \mathbf{k}_{kf} \tag{5.14}$$

Zero-velocity constraint for the first and the last frame:

$$\dot{\mathbf{p}}(0) = [0, 0, 0]^t \tag{5.15}$$

$$\dot{\mathbf{p}}(l) = [0, 0, 0]^t \tag{5.16}$$

Dynamic balance constraint at key frames:

$$\mathbf{H}_i \mathbf{w}_p(k_i) < \mathbf{h}_i, \quad \forall k_i \in \mathbf{k}_{kf} \tag{5.17}$$

#### Output

Once the algorithm finds the optimal solution, the trajectory and the derivatives then can be computed by using Equation (5.5) and Equation (5.6).

### 5.3.3 Example

We have firstly tested the keyframe-based method using several experimental recorded motions. In this section, we present one example of CoM trajectory generation. The recorded motion that we used in this example is motion "01_$TR$_$H1C1S2$_1" which has been presented and analyzed in Section 3.2.2. In order to save computation time during the optimization, we evaluate the balance criterion only at key frames, and verify a posteriori that it is satisfied throughout the whole trajectory. If not, we could add key frames and launch a new optimization from previous solution. This work is carried out in Matlab. The trajectory is parameterized as a 5-th order 3D B-spline with 16 control points.

The generated trajectory and its velocity curves are shown in Fig. 5.3 and Fig. 5.4.

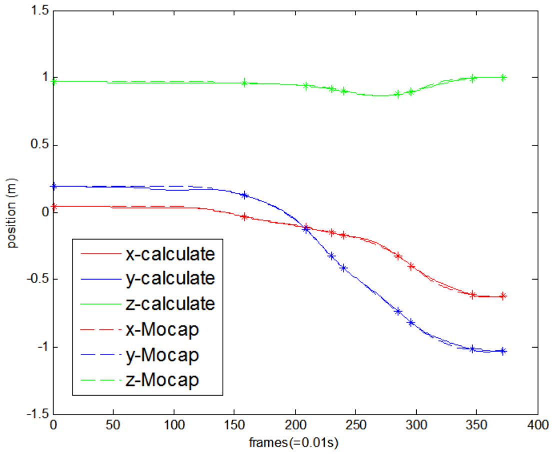


**Figure 5.3:** Comparison of the MoCap trajectory and the generated trajectory: positions expressed in three directions. * indicates the key frames in which the positions are imposed in the splines

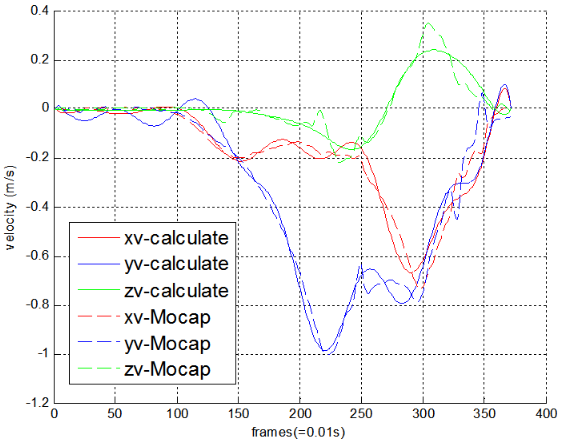


**Figure 5.4:** Comparison of the MoCap trajectory and the generated trajectory: curves of velocity

The generated trajectory fits well with the original one. We have checked a posteriori the dynamic stability constraint for all the frames (see Fig. 5.5). It has been verified that stability constraint is respected throughout the whole motion since the margin stability is positive for all the frames.

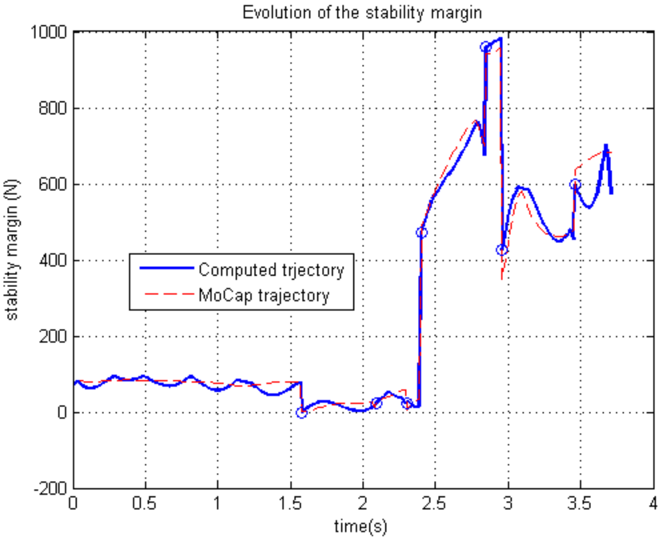


**Figure 5.5:** The stability margin traced for the generated trajectory. Stability margins of the MoCap trajectory is also illustrated for comparison.

The generated CoM trajectory as well as the estimated end-effector (Feet, left hand and head) trajectories of the recorded motion has been used as control references in a dynamic simulation. The simulation is carried out in XDE-Dsimi® (see detailed introduction on motion execution in dynamic simulations in next chapter). The dynamic DHM is associated with a dynamic controller which guides the DHM to track the reference trajectories. The tracking tasks define multiple objectives for the optimization of the controller. Each objective is assigned with a weighting factor which is manually tuned according to relative priority of the task. The DHM has successfully realized the motion in this simulation. Several clips are shown in Fig. 5.6.

The DHM has realized this motion without losing balance in this simulation. Thus the simulation validates the feasibility of the generated CoM trajectory. However, some main drawbacks of the keyframe-based method have been revealed in the simulation which will be discussed in next section.

### 5.3.4 Remarks

The keyframe-based method presented in this section can generate CoM trajectory for a multi-step motion based on key frame informations of a real motion. It shows a feasible result in the simulation that we have carried out, but it is still limited to be used in our study. Firstly, during the simulation work, it has taken a large amount of time to tune the weighting factors for multiple tasks: with non-proper weights, the DHM loses balance during the the motion because of the weak stability margin of the CoM trajectory. Moreover, the motion realized by DHM in the simulation lacks naturalness.

According to the above-mentioned problems and the objectives of this study, we

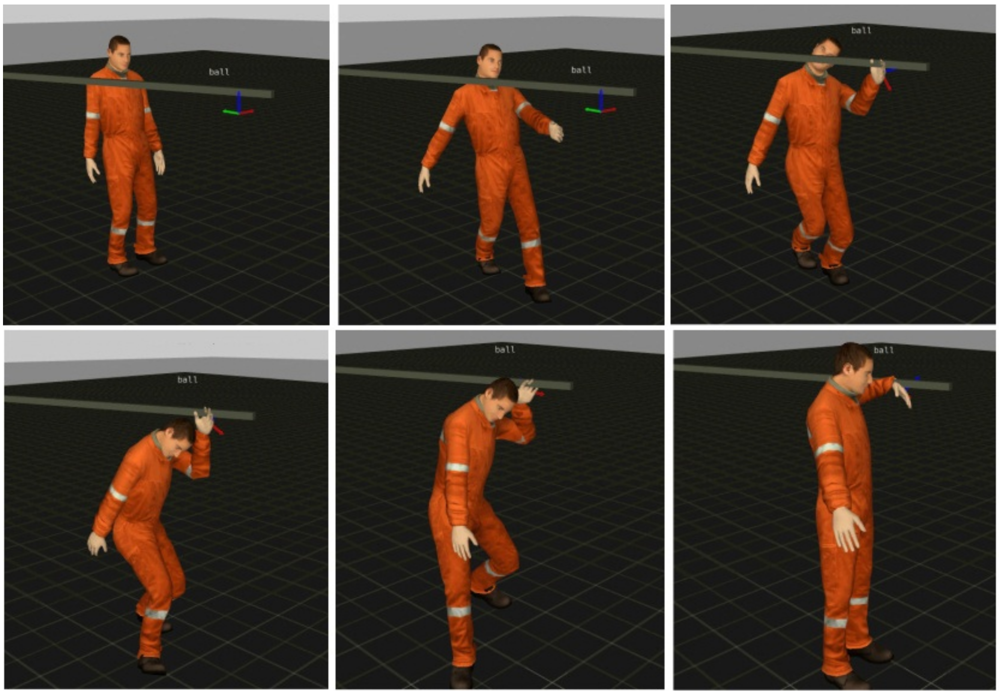


**Figure 5.6:** Clips of the dynamic DHM motion in the simulation in which the CoM trajectory generated by the keyframe-based method is implemented.

thus summarize the main drawbacks of this method as follows:

- Lack of robustness for tolerating modeling errors;
- Too important dependency on experimental data.

In next section, we will explore an improved method that overcomes these shortcomings.

## 5.4 Timing-free method

To overcome the drawbacks of the keyframe-based method presented in the previous section, we have developed a second method which is called “timing-free” method. In this method, the key instants of support change (motion timing) are chosen as variables in optimization. Given a priori a sequence of support configurations without timing information, this method can compute a feasible and stable CoM trajectory with automatically computed durations of transition phases. Moreover, we have added the objective of maximizing the lower bound of stability margins in order to improve the robustness of the trajectory.

### 5.4.1 Problem statement

For a given sequence of support configurations, the admissible pseudo-wrench spaces $\mathcal{W}_i (i \in [0, 1, 2, \cdots, n_t])$ can be pre-computed and expressed in their H-representation with $\mathbf{H}_i$ and $\mathbf{h}_i$. Integer $n_t$ is the number of transitions between pre-defined contact-grasp configurations. We aim to compute a 3D CoM trajectory in function of time $t$:

$$traj := \mathbf{p}(t) = [x(t), y(t), z(t)]^t \tag{5.18}$$

and a time vector corresponding to changes of support configurations:

$$\mathbf{t}_{dist} = [T_0 = 0, T_1, \cdots, T_{n_t}] \tag{5.19}$$

Stability margin of his trajectory at instant $t$ is denotes as $d(t)$.

In order to make the trajectory energy-optimal and natural looking and in order to avoid spline vibrations, we aim to minimize the velocities and the jerks of the trajectory; this trajectory is expected to have large margin of stability and a limited duration of motion while respecting dynamic, geometric, and kinematic constraints:

$$\begin{aligned}
Objective: \; & \mathbf{max}(d_{min}),\ \mathbf{m}in(\|\dot{\mathbf{p}}\|),\ \mathbf{min}(\|\dddot{\mathbf{p}}\|)\ and\ \mathbf{min}(T_{n_t}) \\
Subject\ to: \; & \\
& d_{min} > 0 \\
& \textbf{for}\ \ (T_{i-1} \leq t < T_i, i \in [1, 2, \cdots, n_t]): \\
& \quad (\mathbf{h}_i - \mathbf{H}_i \mathbf{w}_p(t)) \geq d_{min} \\
& \mathbf{A}_{eq}\mathbf{u} = \mathbf{b}_{eq}, \mathbf{A}_{neq}\mathbf{u} \leq \mathbf{b}_{neq}, \\
& f_{nl}(\mathbf{u}) \leq \mathbf{0}
\end{aligned}$$

where

- $d_{min}$: is the lower bound of the stability margin throughout the motion;
- $\mathbf{A}_{eq}, \mathbf{b}_{eq}, \mathbf{A}_{neq}$ and $\mathbf{b}_{neq}$ are linear constraint matrices or vectors for imposing initial or final conditions and lower bounds for time durations;
- function $f_{nl}(\mathbf{u})$ defines some non-linear constraints in this problem such as the geometric constraints (e.g. CoM-support distance limit).

### 5.4.2 Optimization

The vector to optimize is chosen as:

$$\mathbf{u} = [\mathbf{a}_x, \mathbf{a}_y, \mathbf{a}_z, T_1, \cdots, T_{n_t}, d_{min}]^t \tag{5.20}$$

with:

- $\mathbf{a}_x, \mathbf{a}_y, \mathbf{a}_z$: vectors of control points for splines of the x, y and z direction;
- $T_1, \cdots, T_{n_t}$: sequence of time corresponding to changes of support configurations.

### Constraints

The dynamic stability constraint is implemented $n_s$ times during the $i$-th transition:

```
$for \quad j := 1 : n_s$
    $t = T_{i-1} + \frac{j}{n_s}(T_i - T_{i-1}) \ (if \ i = 1, T_{i-1} = 0)$
    $\mathbf{w}_p(t) = [\mathbf{p}(t), \hat{\mathbf{p}}(t)(\ddot{\mathbf{p}}(t) - \mathbf{g})]^t$
    $MARGIN(\mathbf{w}_p(t), \mathcal{W}_i) \geq d_{min}$
$end$
```

The lower bound for stability margin must be positive in order to ensure the dynamic balance:

$$d_{min} > 0 \tag{5.21}$$

The initial and the final positions ($\mathbf{p}_i$ and $\mathbf{p}_f$) of the trajectory are imposed which are the first and last control points of the spline (the spline is *clamped*):

$$[a_{x0}, a_{y0}, a_{z0}]^t = \mathbf{p}_i \tag{5.22}$$

$$[a_{xn}, a_{yn}, a_{zn}]^t = \mathbf{p}_f \tag{5.23}$$

We intend to impose the initial and the final velocities to be zero in some cases:

$$\mathbf{p}_s^1(\bar{t} = 0) = [x_s^1(0), y_s^1(0), z_s^1(0)]^t = [0, 0, 0]^t \tag{5.24}$$

$$\mathbf{p}_s^1(\bar{t} = 1) = [x_s^1(1), y_s^1(1), z_s^1(1)]^t = [0, 0, 0]^t \tag{5.25}$$

During the $i$-th transition, a geometric constraint is also defined to limit the CoM within a reasonable distance from each active support position:

$$\|\mathbf{p}(t) - \mathbf{p}_{ij}\| \leq l_{lim} \tag{5.26}$$

where $l_{lim}$ can be obtained from experimental results (see Chapter 3).

### Objective functions

The lower bound for stability margin is expected to be maximized:

$$obj1 = min(-d_{min}) \tag{5.27}$$

The objective of minimization of velocity and jerk is implemented $n_{vj}$ times during the whole motion:

$$obj2 = \sum_{i=1}^{n_{vj}} \dot{\mathbf{p}}(t)^t \mathbf{Q}_v \dot{\mathbf{p}}(t) + \sum_{i=1}^{n_{vj}} \dddot{\mathbf{p}}(t)^t \mathbf{Q}_j \dddot{\mathbf{p}}(t) \ with \ t = \frac{i}{n_{vj}}(T_{n_t})$$

where $\mathbf{Q}_v$ and $\mathbf{Q}_j$ are the weighting matrices.

The total time of the motion should be limited:

$$obj3 = min(T_{n_t})$$

The synthesis of all the objectives:

$$obj(u) = \varpi_1 obj1 + \varpi_2 obj2 + \varpi_3 obj3 \tag{5.28}$$

where $\varpi_1, \varpi_2$ and $\varpi_3$ are weighting scalars.

#### Results output

When the algorithm successfully finds the optimal solution, we can compute the trajectory as well as the velocity and acceleration curves by using Equations (5.5) and (5.6).

### 5.4.3 Discussions

#### Problem feasibility

In order to avoid wasting time on solving unfeasible optimization problem, we evaluate the feasibility of the CoM trajectory generation problem. The feasibility of the problem is evaluated in viewpoint of the balance constraint. As presented in previous sections, a sequence of polytopes $\mathcal{W}_i (i \in [0, 1, 2, \cdots, n_t])$ serve as balance constraints. Each polytope constrains DHM balance in corresponding transition phase. Moreover, the CoM acceleration is supposed to be continuous in our problem. Thus at a key frame the DHM should satisfy balance constraints for both two consecutive phases before and after this key frame. An example is shown in Fig. 5.7. In this example, it is suppose that the DHM moves via three consecutive support configurations for balance constraint. Point $\mathcal{P}_1$ is the pseudo-wrench of the DHM at the first instant of supports change, and it should be located inside both $\mathcal{W}_1$ and $\mathcal{W}_2$.

**Proposition 1.** *Suppose there are* $n_t + 1$ *polytopes* $\{\mathcal{W}_i, i \in (0, 1, \ldots, n_t)\}$ *defining the balance constraints for a* $n_t$*-transition motion, if* $\exists j \in (1, \ldots, n_t)$ *that* $\mathcal{W}_{j-1} \cap \mathcal{W}_j = \oslash$*, then the motion planning problem is not feasible.*

In fact, if there is no jumping or flying phases during the DHM's locomotion, the trajectory generation problem will always be feasible since any two consecutive support configurations share one or more common supports.

If the CoM trajectory generation problem is feasible, we then take the Chevyshev ball radius of the smallest intersection as criterion for evaluating the difficulty of the problem. For a same DHM with different support configurations, the smaller this value is, the more difficult the CoM trajectory generation problem will be. Moreover, this criterion predicts the upper limit for the lower bound of the stability margin in the timing-free method.

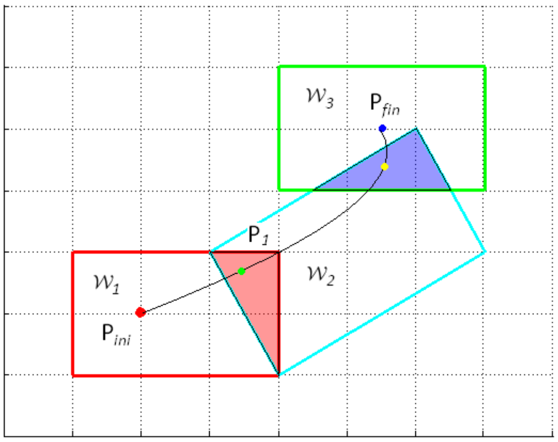


**Figure 5.7:** Illustrating the feasibility of the problem in terms of balance constraints. In the pseuduo-wrench space, three polytopes constrain the balance of the DHM during its motion. The pseudo-wrench of the DHM is continuous curve that is in function of the position and acceleration of CoM. If the intersections of the consecutive polytopes are non-null subspace, the CoM trajectory generation problem is regarded to be feasible in terms of balance insurance.

**Proposition 2.** *Suppose the* $n_t$ *polytopes* $\{\mathcal{W}_{i\cap(i+1)} = \mathcal{W}_i \cap \mathcal{W}_{i+1}, i \in (0,1,\dots,n_t-1)\}$ *are the intersections of the* $n_t+1$ *polytopes for balance constraints. Suppose the Chevyshev ball radius of the intersections are* $\{r_i^{chev}, i \in (0,1,\dots,n_t-1)\}$ *with* $r_{min}^{chev}$ *the smallest radius in this set. Then* $r_{min}^{chev}$ *is regarded as a criterion for evaluating the difficulty of the problem.*

### Choosing internal knots

According to the *de Boor algorithm* (see Equation (2.30) and Algorithm 2), the $j$-th segment of a $k$-th degree B-spline

$$\mathbf{s}_j(\bar{t}) : \bar{t} \in [t_j, t_{j+1}), k \le j \le n$$

that is defined on the knot interval $[t_j, t_{j+1})$ depends only on $k+1$ adjacent control points: $a_{j-k}, \cdots, a_j$. Accordingly, when we include the timing in the optimization process, it is very important to choose the appropriate internal knots vector. The importance of this choice lies mainly in two aspects:

1. When a normalized duration of transition is smaller than the knot interval, there is the risk that too many constraints are imposed so that several control points are over-constrained, making it impossible to find the optimal control points. As shown in the Fig. 5.8, two normalized duration of transition $[\bar{t}_i, \bar{t}_{i+1}]$ and $[\bar{t}_{i+1}, \bar{t}_{i+2}]$ lie inside the knot interval $[t_j, t_{j+1})$ of the 3rd-degree spline shown in the figure. The spline segment on the knot interval $[t_j, t_{j+1})$ depends on 4 control points $a_{j-k}, \cdots, a_j$. During each transition, several dynamic and geometric constraints are imposed at some sampling instants. We can see that, the constraints

in both the two transition periods have influence on the choice of the control point $a_j$ (we say they are active constraints for $a_j$). If we displace the knot $t_j$ to $\bar{t}_{i+1}$, as shown in Fig. 5.9, the constraints during the transition $[\bar{t}_i, \bar{t}_{i+1}]$ becomes inactive for $a_j$.

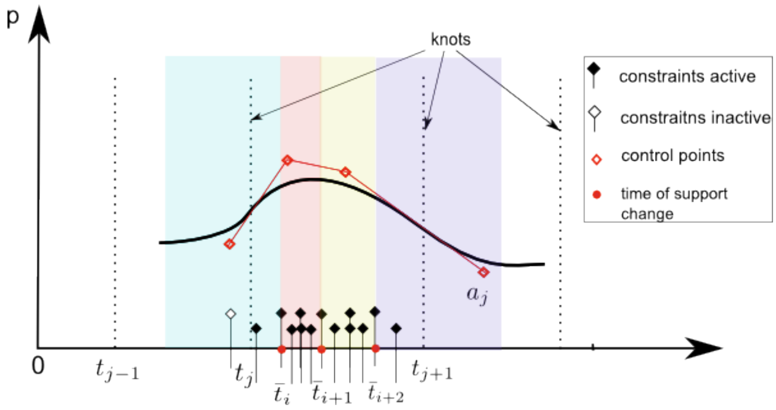


**Figure 5.8:** Large knot interval may cause over-constraints.

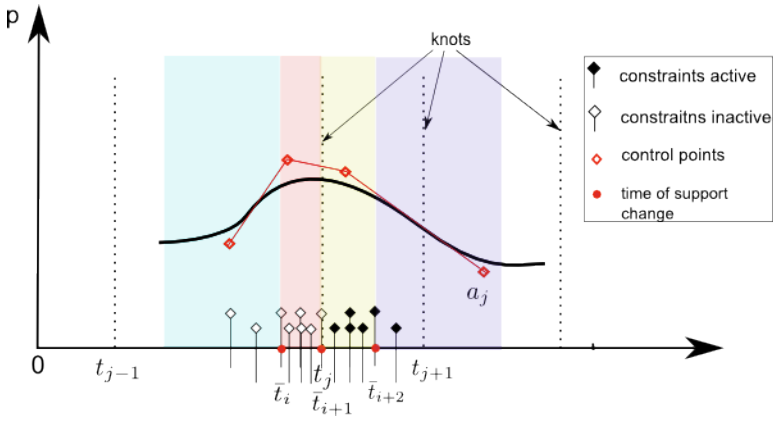


**Figure 5.9:** There are less constraints when the knot interval becomes smaller.

2. When a normalized transition duration covers several knot intervals, more control points are involved for the same constraints which means that the optimization problem has more degrees of freedom in for this duration. This problem won't prevent our algorithm to find the optimal solution, but it may waste the computing time and make the spline less smooth.

Based on the above analyze, we then propose a modification by automatically configuring the internal knots during the optimization: each normalized instant of support configuration change is set to be an internal knot, as shown in Fig. 5.10. Thus, the knots vector turns to:

$$\mathbf{T} = [\,\underbrace{0, \cdots, 0}_{k+1}, \sigma T_1, \cdots, \sigma T_{n_t-1}, \underbrace{1, \cdots, 1}_{k+1}]$$

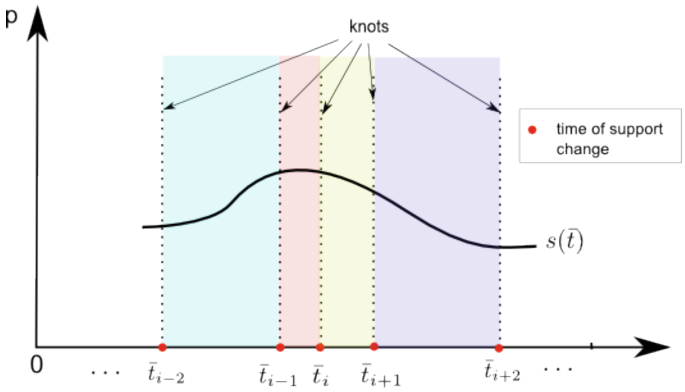


**Figure 5.10:** An automatic configuration of knots.

### 5.4.4 Several scenarii

Here we present the implementation of the timing-free method in several scenarii including biped walking, quadruped climbing walls and DHM moving in complex environment. These simulations are carried out in Matlab. Toolboxes for polytope computation (MPT) and automatic differentiation (Adiff) are used in our methods.

#### Biped walking

This method is firstly tested in a biped walking scenario shown in Fig. 5.11. A $70kg$ biped is supposed to walk on an even terrain. The feet contacts in this scenario are shown in Fig. 5.13. The biped begins its motion with its CoM at $\mathbf{p}_0 = [0, 0, 0.96]$, walks four steps and stops at $\mathbf{p}_f = [0, 1.5, 0.96]$.

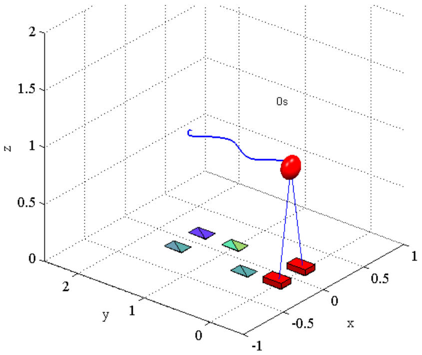


**Figure 5.11:** Scenario illustration: a biped walks multiple steps by tracking the 3D CoM trajectory.

During the whole motion, the position of CoM is designed to have small variations in its $z$-direction by imposing an additional objective in the optimization. The minimum time for single-support phase and double-support phase are set to be respectively $0.4s$ and $0.15s$. We considered in this scenario the stability margin in the 6D pseudo-wrench space. The weighting factors in this trial are

$\varpi_1 = 100, \varpi_2 = 1000, \varpi_3 = 1$ with $\mathbf{Q}_v = \mathbf{I}_{3,3}$ and $\mathbf{Q}_j = 1e\text{-}3*\mathbf{I}_{3,3}$. There are 9 transitions ($n_t = 9$) thus the 4-th order spline has 18 knots and 13 control points for each direction. Our algorithm can successfully compute the solution within 3 minutes (solving with *fmincon* function on a PC with Intel Xeon CPU 2.67GHz). The result in one of the trials is shown in Fig. 5.12. The whole motion has a duration of 3.54s with a stability margin of $25.86 N.m$.

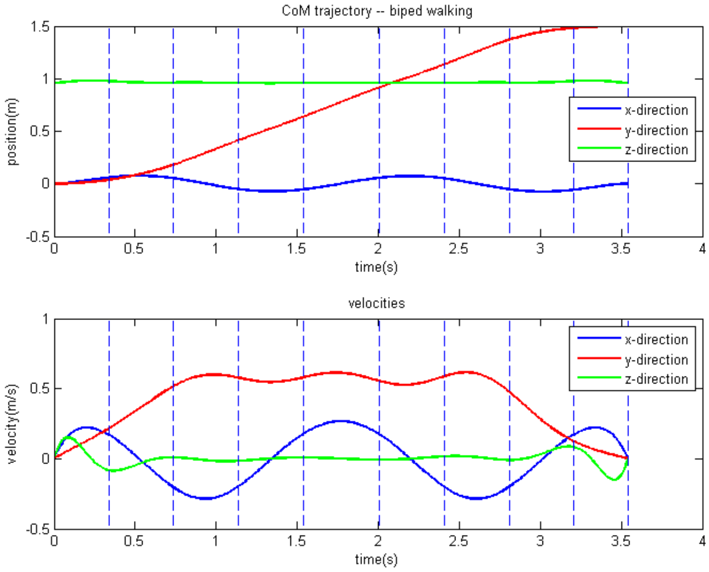


**Figure 5.12:** Trajectory curves generated by the method. Blue dashed lines indicate the time of contact change.

We have traced the ZMP trajectory of the motion as shown in right figure in Fig. 5.13. As we can see, the ZMP point during the whole motion stays always inside its support polygons.

The biped motion for walking on inclined terrains has also been successfully generated via this method. One can see some simulation clips of this motion in Fig. 5.14.

We chose the biped walking scenario mainly for proving the correctness of this method. Although our method does not show advantage in computing efficiency regarding existing ZMP-based methods, the following scenarii can highlight its generality.

### Quadruped climbing parallel walls

This method has been tested in a quadruped scenario shown in Fig. 5.15. A quadruped of mass 120kg moves between two parallel vertical walls. It has a mass point located initially at $\mathbf{p}_i = [0, 0, 1]$ and 4 massless feet with dimension 40cm×10cm. The coefficient of friction between the feet and the two walls is set to be: $\mu = 1.0$. The two walls are separated by 2m ($x = 1$ and $x = \text{-}1$). The maximum normal force for each foot contact is limited to 0.5kN. As one can see,

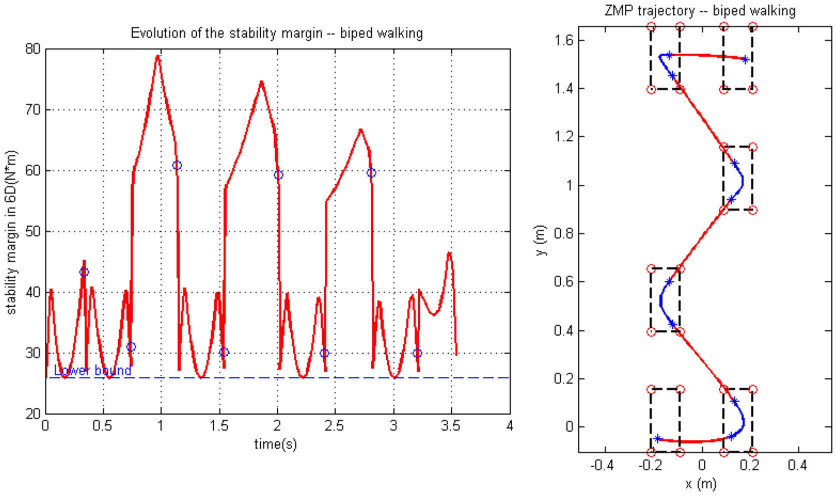


**Figure 5.13:** Left: stability margins of the generated biped walking motion. Right: Footsteps and the trajectory of ZMP computed from the result CoM trajectory (double-support phases in red, single-support phases in blue). Instants of support change are indicated by * in this figure.

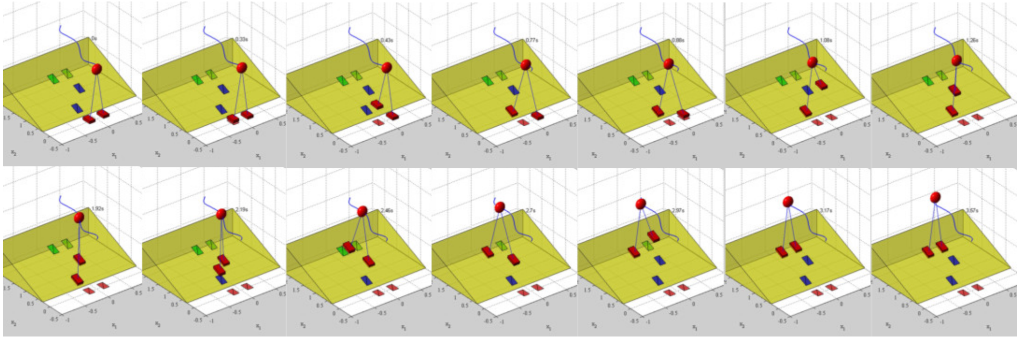


**Figure 5.14:** Motion clips of the biped walking on a 15-degree ramp by tracking the generated CoM trajectory.

the maximum frictional force with only two foot contacts is not large enough for the robot to maintain its static balance.

The strategy of movement is shown in Fig. 5.15. It has been chosen to include phases with only two feet in contact where a statically stable motion is not feasible. The motion consists of $n_t = 9$ transitions via the 9 given contact configurations (the last transition has no contact change). The contacts on the two walls during all the transitions are symmetric. The initial height for the feet contact are respectively $0.85m$ and $1.15m$. During the motion, a foot contact which is removed will be restored at a $0.5m$ higher position. Each foot contact has always the $40cm$ long edge parallel with the ground. The sequence of contacts can thus be automatically computed. The final position of the mass point is $\mathbf{p}_f = [0, 0, 2]$.

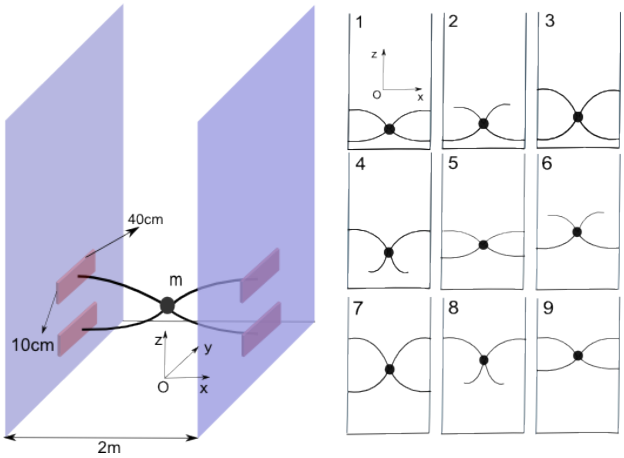


**Figure 5.15:** A quadruped moving between two vertical walls with vertical contacts (left) and the motion strategy with feet contacts sequence (right).

We chose the degree of the spline to be 4 thus there were 18 knots and 13 control points for each direction. The admissible pseudo-force space is used to calculate the stability margin. And a lower limit for each transition time duration is imposed: $\delta t \geq 0.3s$. The coefficients used in the conducted trials are shown in Table 5.1. For all the trials, the weighting matrices are set as: $\mathbf{Q}_v = \mathbf{I}_{3,3}$ and $\mathbf{Q}_j = 1e\text{-}3{*}\mathbf{I}_{3,3}$ with $\varpi_3 = 1$. Running in Matlab on a PC with Intel Xeon CPU (2.67GHz), the algorithm has taken less than 1 minute to successfully find the optimal solution. The calculated trajectory has rather small displacements in x and y direction in each trial.

| Trial No. | Weights($\varpi_1, \varpi_2$ ) | Degree | Duration(s) | $d_{min}$ (N) |
|---|---|---|---|---|
| 1 | (1, 1e3) | 4 | 3.58 | 0.30 |
| 2 | (1e3, 1) | 4 | 3.90 | 196.26 |
| 3 | (1e3, 1e-3) | 4 | 3.92 | 134.01 |
| 4 | (1e4, 1e-3) | 4 | 5.29 | 280.55 |
| 5 | (1e4, 10) | 4 | 5.28 | 280.60 |

**Table 5.1:** Trials with different objective weights

The trajectory generated in the trial #4 as well as its velocity curve are shown in Fig. 5.16 (left). The whole trajectory has a time span of 5.29s and the lower bound of stability margin 280.55$N$. The time durations of transitions between different contact configurations are also shown in Fig. 5.16. One can remark from the figures that: the quadruped decelerates during 2-contact phases so that it can maintain a large stability margin with less supports; it accelerates during 4-contact phases to compensate the velocity loss in 2-contact phases since it has sufficient frictional force with four supports.

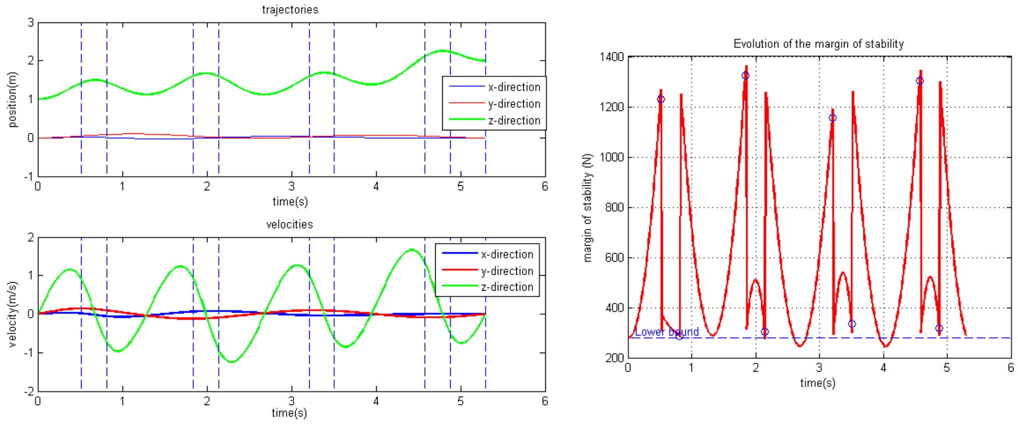


**Figure 5.16:** Left: generated trajectory and the velocity curves for the quadruped scenario. Right: Evolution of stability margin for the trajectory. Blue circles indicate the time of contact change.

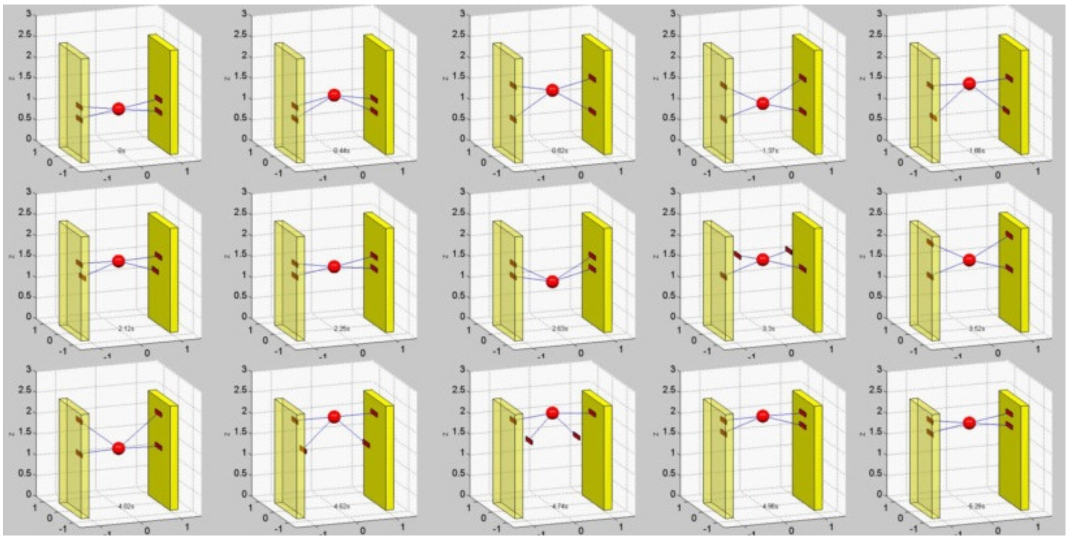


**Figure 5.17:** Clips of simulation: a quadruped moves between two vertical walls by tracking the generated CoM trajectory.

The stability margin is displayed in Fig. 5.16 (right). As we can see, there are several instants where the stability margin is slightly smaller than the lower bound. This is caused by the fact that the stability constraint is imposed at discretized instants in the optimization. Nevertheless, the stability margin remains largely positive. Thanks to the objective of minimizing the velocity and the jerk, the stability margin has no abrupt change during the periods between those discretized instants.

### Implementation in a complex scenario

CoM trajectory for a biped motion in a complex scenario has been generated (see Fig. 5.18). A DHM steps over a vertical board and walks onto a platform under a roof.

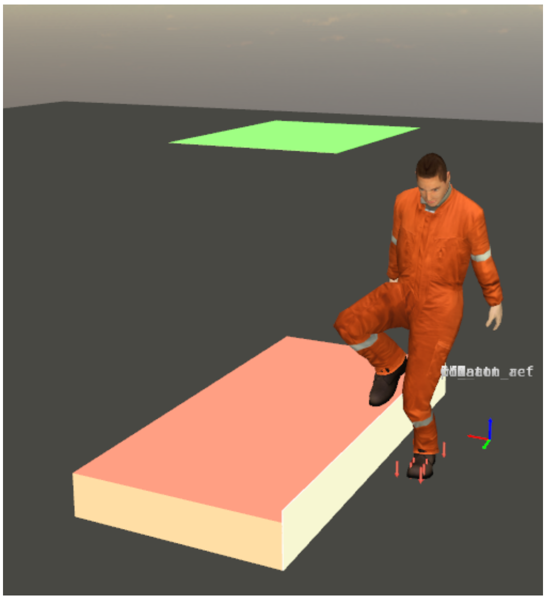


**Figure 5.18:** The environment and the DHM for the scenario where the environment consists of a roof, a platform and a threshold (vertical board) on the floor.

Positions and dimensions of all the objects in this environment are given in Table 5.2. The DHM is 1.75$m$ high and weights 79kg. The initial position and final position of its center of mass are respectively $\mathbf{p}_i = [0, 0, 0.92]^t$ and $\mathbf{p}_f = [0.6, 0.65, 0.86]^t$. Its foot sole is represented as a $28cm \times 12cm$ rectangle. The sequence of foot steps are pre-defined manually in this example (see Fig. 5.19 and Table 5.3).

| Object | Geometry center position (m) | Dimension |
|---|---|---|
| Roof | (0.9, .75, 1.5) | 1.1m × 1.5m × 5cm |
| Threshold | (0.35, 0.75, 0.15) | 5cm × 1.5m × 30cm |
| Platform | (0.9, 0.75, 0.1) | 1.1m × 1.5m × 0.2m |

**Table 5.2:** Scenario specification: the environment

There are 7 support configurations in this motion, accordingly there are 7 transitions between support configurations ($n_t = 7$). The spline is chosen to have a degree 5, thus it has 18 knots and 12 control points. Because of the roof – an overhead board, the biped should lower its CoM during the motion. Thus we define

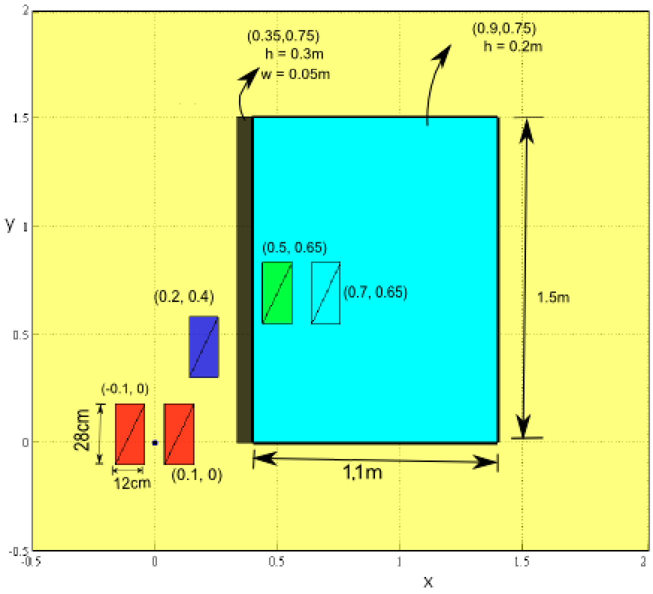


**Figure 5.19:** Foot contacts shown in the virtual environment with position and dimension denotations (the roof is not shown in this figure).

| Order | Position (Feet centers) | Phase |
|---|---|---|
| 1 | RF:(0.1 0.0 0), LF: (-0.1 0.0 0) | double support |
| 2 | RF:(0.1 0.0 0), LF: flying | single support |
| 3 | RF:(0.1 0.0 0), LF: (0.2 0.4 0) | double support |
| 4 | RF: flying, LF: (0.2 0.4 0) | single support |
| 5 | RF:(0.7 0.65 0.2), LF: (0.2 0.4 0) | double support |
| 6 | RF:(0.7 0.65 0.2), LF: flying | single support |
| 7 | RF:(0.7 0.65 0.2), LF: (0.5 0.65 0.2) | double support |

**Table 5.3:** Foot placement sequence. The motion strategy is designed as: DHM walks forward a step with its left foot, then it steps its right foot onto the platform; finally the DHM steps its left foot onto the platform and keeps a static standing posture under the roof.

an additional geometric constraint on the height of CoM, depending on knowledge from our experimental database introduced in Chapter 3:

$$\begin{cases} h_u = 0.91 - 0.08 * arctan(10 * (x - 0.2))/pi \\ h_l = 0.89 - 0.08 * arctan(10 * (x - 0.2))/pi \end{cases} \tag{5.29}$$

Equation (5.29) analytically represents the upper and lower limits of CoM height in function of x-coordinate of CoM using arctangent functions, resulting in two surfaces in the structured environment (see Fig. 5.20). The upper limits at $x = 0$ and $x = 0.6$ are respectively $0.938m$ and $0.876m$ (lower limits are respectively $0.918m$ and $0.856m$). Using these constraints, we force CoM to bend down around the left edge of the roof ($x = 0.35m$) during the motion. CoM height constraints are imposed at some sampled instants.

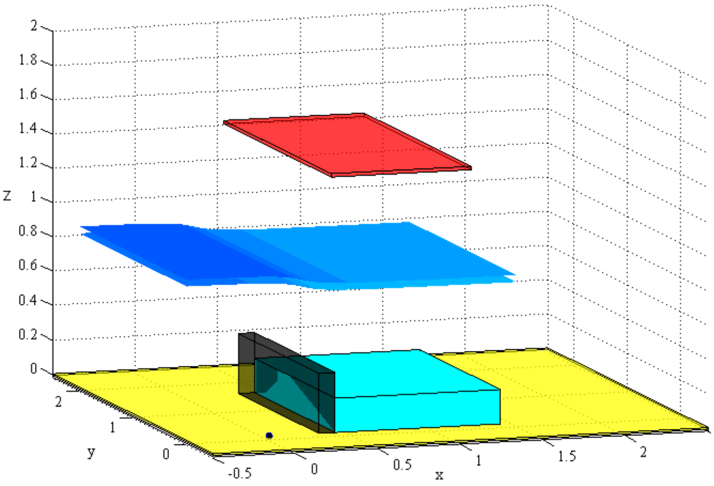


**Figure 5.20:** The geometric constraints defined in this scenario for limiting the CoM height. CoM should be located between the two surfaces defined by Equation (5.29).

The minimum time durations for two phases in which the biped steps over the threshold with each foot is chosen as $0.8s$.

The CoM trajectory generation method successfully found the solutions for several trials with different optimization parameters. One generated global robust trajectory is shown in Fig. 5.21. Parameters of the generated spline are shown in Table 5.4. Detailed information of the solution trajectory can be seen in Fig. 5.22 and Fig. 5.23. The trajectory has its stability margin larger than 20 $N.m$.

Next chapter will carry on the whole-body motion planning and motion execution for this scenario.

| Solution spline parameters in the scenario | | |
|---|---|---|
| Control Point | 1 | (0, 0, 0.92) |
| | 2 | (0, 0, 0.92) |
| | 3 | (0.024, 0.007, 0.972) |
| | 4 | (0.091, 0.027, 0.942) |
| | 5 | (0.118, 0.117, 0.752) |
| | 6 | (0.268, 0.643, 1.076) |
| | 7 | (0.132, 0.160, 0.640) |
| | 8 | (0.842, 0.761, 1.002) |
| | 9 | (0.658, 0.523, 0.660) |
| | 10 | (0.687, 0.656, 0.950) |
| | 11 | (0.6, 0.65, 0.86) |
| | 12 | (0.6, 0.65, 0.86) |
| Knots | 18 | (0, 0, 0, 0, 0, 0, 0.0766, 0.172, 0.222, 0.413, 0.677, 0.867, 1, 1, 1, 1, 1, 1) |
| Key instants | 8 | 0, 0.321, 0.721, 0.930, 1.730, 2.837, 3.637, 4.193 (s) |
| Degree | 5 | |

**Table 5.4:** Parameters of the generated spline mapped to the global CoM trajectory

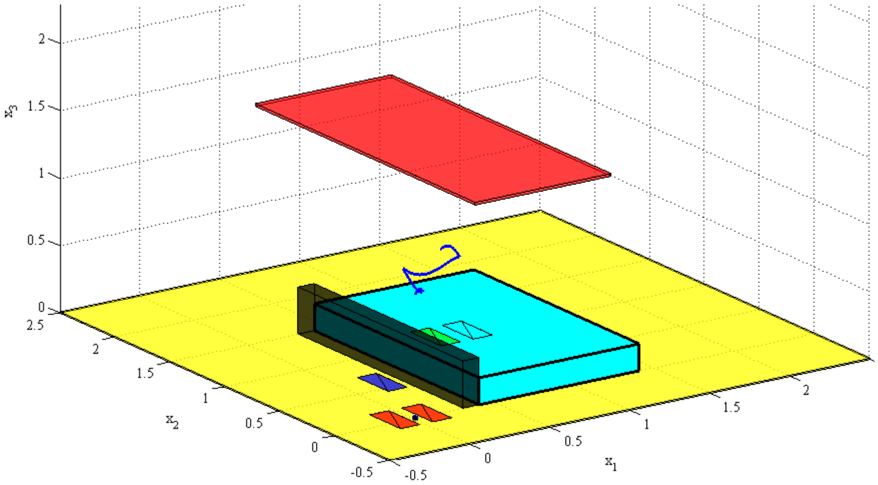


**Figure 5.21:** Illustrating the global CoM trajectory in the structured environment.

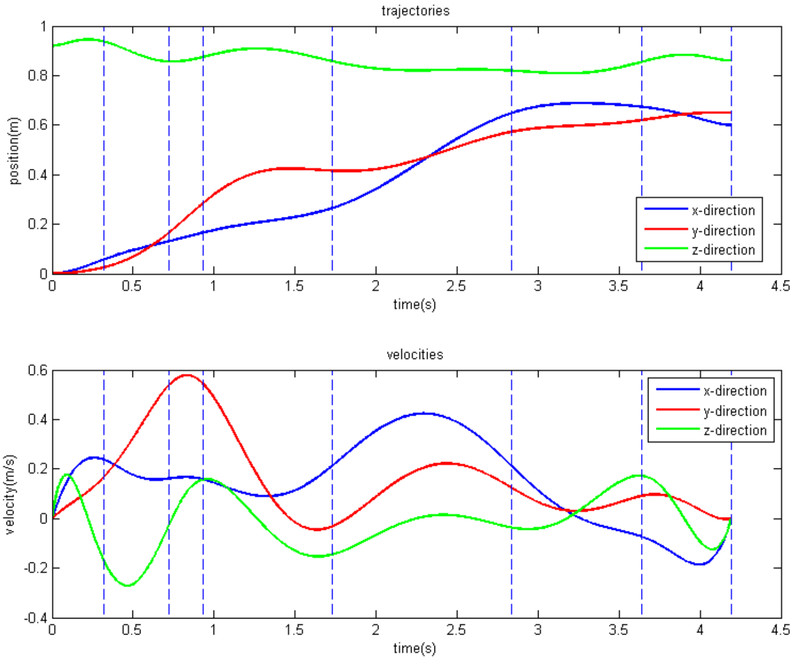


**Figure 5.22:** Global trajectory: position and velocity in x, y and z directions.

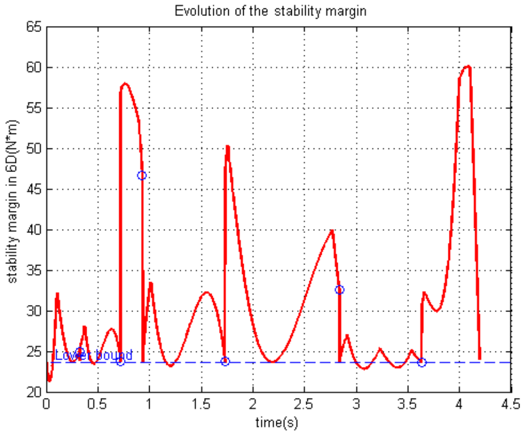


**Figure 5.23:** The stability margins of the global CoM trajectory.

## 5.5 Conclusion

This chapter presents the methodologies of the first level of our framework, namely the generation of the global CoM trajectory. The two methods explored in this chapter can be used to generate a spline CoM trajectory that would ensure stable balance of a DHM or a multi-limbed robot during its dynamic motion. Both the two methods take the balance constraints into account using a series of polytopes computed from a given sequence of support configurations. Besides the dynamic balance constraint, some geometric constraints are imposed to guarantee the kinematic feasibility of CoM positions in transition phases. Relying on some information of the MoCap experimental database, the keyframe-based method can generate a CoM trajectory that passes the real positions of the CoM trajectory of a real motion at its key instants. In order to release its dependency on experimental data and increase the robustness of the spline trajectory, the timing-free method is then explored. In this method, the trajectory is parameterized by a non-uniform B-spline whose control points and knots vector are variables in the optimization. The lower bound for stability margin and time durations for holding each support configuration are also optimized. The high order B-spline guarantees not only the smooth CoM trajectory but also the smooth velocity and acceleration curves. The optimal lower bound of stability margin makes the motion robust to tolerate tracking or modeling errors or to resist perturbations. Both the two methods have been tested for several scenarii and they show quite good performances.

## Chapter 6

# Whole-body collision-free motion planning and simulation

In the previous chapter we have presented some methods at the first level of the framework for computing a global robust CoM trajectory. However, little consideration about whole-body posture and collision avoidance has been taken into account at this level. To successfully realize a motion in cluttered environment, a DHM must be able to avoid all possible collisions with the environment, meanwhile it should maintain balance at any time. We present here planning methods at the second level of the framework for generating whole-body collision-free motions based on the pre-computed global CoM trajectory. At the last level of the framework, the whole-body motion generated at the first two levels will be executed on a dynamic DHM in simulations by virtue of dynamic controllers.

This chapter is organized as follows. In Section 6.1 we introduce a state of the art on several problematics addressed in this chapter. In Section 6.2 we develop methods for generating locally the whole-body collision-free motion. Section 6.3 presents the execution of the generated motion via a dynamic controller in dynamic simulations. In Section 6.4 we carry on with the last scenario in the previous chapter and show results of motion planning and motion execution for this example.

## 6.1 State of the art

In this section, we give a state of the art on several problematics including DHM motion planning, collision detection and motion control.

### 6.1.1 Motion planning for DHMs and humanoid robots

Motion planning has always been an essential problem that is broadly and continuously studied in robotics. One can get an overview of the development of motion planning techniques in some excellent books( [Lat91] and [LaV06]). Usually, motion planning is carried out in a *configuration space* ([LP83]) denoted as $C$. Each configuration of the robot thus becomes a point in this space: $\mathbf{q} \in C$. The *configuration space* is divided by all motion constraints into *C-obstacles* and *free-space* noted respectively as $C_{obs}$ and $C_{free}$. Objective of motion planning then can be regarded as finding a continuous path in $C_{free}$ that connects the starting and the target configurations.

The performance of a motion planning algorithm depends on the computational complexity of the motion planning problem. Both the dimension of the moving system and the type of the constraints have great influence on the efficiency of a planning technique ([Can88]). Traditional planning methods such as grid-based searching approaches and geometric algorithms are proved to be efficient for planing motions of low-dimensional locomotive systems, such as point-like or car-like robots. However, they are likely to be invalid for high-dimensional systems with complex constraints, such as the motion planning for human model in our study. Potential field methods and sampling-based methods show reasonable performance in this circumstance.

### Potential field

A potential field is associated to the $configuration$ *space* which takes the goal configuration as a sink (attractive point) and all the obstacles become outlets of repulsion. A configuration point thus is automatically guided to the goal by the potential field, resulting in a collision-free path namely the solution trajectory. This method has been used in motion planning for the humanoid robot HRP-2 in CNRS-AIST JRL ([EKMG09], [BELK09]). Based on a potential field, their planning algorithm can automatically generate a sequence of optimal contacts which can guide the robot to realize its quasi-static collision-free motion in very complex scenarii.

### Sampling-based planning

Difficulties with potential field methods lie in the modeling of the potential field and the possible trap in local minimums. Thus sampling-based algorithms have been proposed and developed in the last two decades, targeting at solving efficiently the motion planning problems for systems with high-dimensional DoFs. In a sampling-based motion planning method, a new point is randomly sampled in $C_{free}$, then one or more edges connecting feasible configuration points and the new one are tested: if one edge lies entirely in $C_{free}$, it is conserved; if not, it is rejected. Step by step, the method builds a roadmap or a tree (or trees) in $C_{free}$. A single-query or multiple-query step is associated with this process to verify whether the planning objective has been realized. This kind of method does not need explicit description of the subspace $C_{free}$, thus it is broadly used and shows good efficiency. The PRM (Probabilistic Roadmap) [OS94] and RRT (Rapidly-exploring Random Tree) [LaV98, LKJ00] are probably the most successful and popular sampling-based methods in recent motion planning studies.

Hauser presents in [HBL05] a non-gaited locomotion planner for generating multi-step motions of humanoid robots over uneven and sloped terrains. Numerical IK method is used to satisfy closed chain kinematics constraints. A Probabilistic Roadmap planner is used to plan the transition for each step between two contact configurations. In this work, the static equilibrium is considered by keeping the CoM above the *support polygon*.

Zhang presents in [ZPM09] an approach for planning whole-body motion by decomposing the whole-body planning problem into a sequence of low-dimensional problems. A constrained coordination sampling-based planning approach is adopted for solving each sub-problem incrementally under the constraints of collision-freeness and static balance.

In dynamic motion planning cases, the timing consideration brings more constraints and thus higher computational complexity to the motion planning task. In this case, the dynamic balance, kino-dynamic constraints and other related constraints make the free configuration subspace more complex.

Kuffner and his colleagues have obtained excellent achievements on a series of topics from footstep planning [KKN+03] to dynamic motion generation [KKN+02]. In [KNK+01], they present a planning framework in which the RRT method is for the first time implemented in humanoid robot motion planning. In this paper, a collision-free statically-stable motion is firstly generated, and it is furthermore converted into a dynamic motion by being zoomed in time. The dynamic balance is guaranteed by filtering the output path via a dynamic filter "AutoBalancer" [KKT+00] based on ZMP (Zero-moment point) criterion.

Yoshida et al. have presented their planning method [Yos05, YBEL05, YES+06] for HRP-2 robot motions. The randomized planning method can adjust the state dimension, namely the controlled DoFs in the RRT algorithm. The dynamic balance is guaranteed by confining the ZMP inside the support polygon of the robot's feet. The idea of tuning the dimension of controlled state has inspired our work, but a general method for adjusting the controlled state dimension was not presented in their work.

Harada presents in [HHH+07] a walking pattern generator based planning method for humanoid robot. The consideration of collision-freeness is firstly excluded in the motion generation. Then PRM planning method is carried out for the periods in which collisions occur.

### 6.1.2 Collision detection

> "Collision detection is a basic tool whose performance is of capital importance in order to achieve efficiency in many robotics and computer graphics applications, such as motion planning, obstacle avoidance, virtual prototyping, computer animation, physical-based modeling, dynamic simulation, and, in general, all those tasks involving the simulated motion of solids which cannot penetrate one another".
>
> *From [Lat91] Chapter 6*

The obstacle avoidance and collision detection depend largely on the modeling/representation of the environment, namely the obstacles in the workspace. Different methods for obstacle modeling show differences in efficiency and accuracy. The popular methods include bounding polyhedrons, parametric surfaces, potential fields, finite element, etc. The choice of obstacles models relies on the requirement of motion planning task. One can get an overview of the state of the art in collision detection between general geometric models in [HKLR02]. Spheres and cylinders are used in [GCLS95] to represent the manipulator which can efficiently realize real-time collision avoidance in experiments with a 7dof redundant manipulator.

Potential field method for collision avoidance has been presented in Section 6.1.1. This method does not need an explicit algorithm for detecting collision.

The notion of “Bump-surfaces” is proposed in [AA05] which is somehow similar to the potential field idea. The entire robot environment is represented by one simple mathematical entity, namely a higher-dimensional B-Spline surface which can be constructed in linear time. Collision detection thus turns to verifying the height of a path (i.e. value in the extra dimension) in this space. Evaluating of a point on B-spline surfaces is quite simple and fast thus the collision detection is very efficient in this manner. This approach has been successfully used in [XAA07, XAA08, XZA10] and shows ideal performance and effectiveness. However, this method is limited when the mobile system has a complex form that can not easily been described by points, such as the case in our study.

For a mobile system that consists of multiple segments such as DHMs or humanoid robots, self-collision between body segments should also be avoided. Some exact collision detectors using different bounding volume models for humanoid robots have been presented in [OOH$^+$05]. In some studies, distance between any two volumes is expected to have a continuous or smooth gradient. Escande et al. propose in [EMK07] a method for generating strict-convexity bounding volumes (a set of spheres and toruses) that cover a polyhedral convex volume in order to guarantee the continuity of the proximity distance gradient.

### 6.1.3 Motion control

Dynamic DHM or real humanoid robots realize their motions with the help of dynamic controllers, which necessitates the application of various motion control techniques and strategies.

A simple DHM operation, such as a hand manipulation, is usually treated at the control level as a tracking task of reference positions, velocities, forces or even accelerations. Since a DHM is a highly articulated and under-actuated system, its high-dimensional configuration brings difficulties to its motion control. Hogan proposed the impedance control technique [Hog84] that shows excellent performance for controlling robotic manipulation and locomotion. Its principle is to control the impedance, namely the dynamic relation between the position of a controlled frame and the force applied on the system. Anderson and Spong proposed a hybrid impedance control method [AS88] to increase the adaptability to uncertain environments.

Behaviors of DHMs and humanoid robots are complex movements which usually concern multiple operational tasks. To decouple the tasks in the control perspective, the prioritized multi-task control has been proposed. Khatib and Sentis have proposed a decoupling approach based on dynamic projection in operational space [Kha80], [SK05] [SAK07]. Collette has proposed a weighted control scheme using Quadratic Programming techniques in [Col09] treating prioritized tasks as weighted objectives.

## 6.2 Local motion planning

At the second level of our framework, we aim to plan locally the DHM’s whole-body collision-free motion. The trajectories of end-effectors and joints generated at this level are finally used at the third level for realizing the planned DHM motion.

### 6.2.1 Problem definition

We have obtained at the first level a robust global CoM trajectory and the time distribution among the support configurations. Thereafter, we plan the DHM whole-body motion by taking into account balance constraint, kinematic constraints and the collision avoidance constraints. The DHM is supposed to be in balance if it follows the pre-generated robust CoM trajectory all the time, thus the balance requirement turns to an IK problem of imposing the DHM's CoM position. Constraints on whole-body motion vary as the DHM moves, thus we carry out the planning work locally (for a transition phase instead of the entire motion).

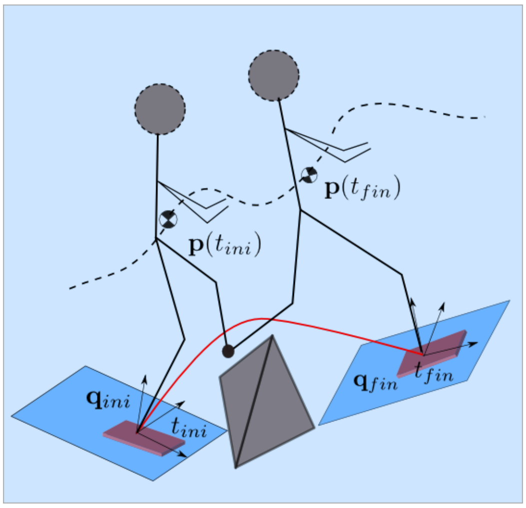


**Figure 6.1:** Illustrating the local planning for a flying foot associated with the imposed CoM trajectory represented by the dashed curve.

We recall here the robust CoM trajectory for the whole motion generated beforehand is represented as:

$$traj := \mathbf{p}(t) = [x(t), y(t), z(t)]^t, t \in [0, t_f] \tag{6.1}$$

The time distribution for the $n_t$ transitions is generated at the same time:

$$\mathbf{T}_{dist} = [0, t_1, t_2, \cdots, t_{n_t} = t_f] \tag{6.2}$$

which is a vector including the entering time for each of the $n_t$ support configurations.

As shown in Fig. 6.1, the problem in this chapter can be formulated as:

$$\begin{aligned} \textbf{For}: &\quad t \in [t_{i-1}, t_i],\ i = 1, 2, \ldots, n_t \\ \textbf{Find}: &\quad [\mathbf{p}_e(t), \boldsymbol{\theta}_e(t)]\ \textit{and}\ \mathbf{q}(t) \\ s.t.: &\quad \textit{support kinematics},\ \textit{CoM position and collision-free constraints} \end{aligned}$$

where

- $\mathbf{q}(t)$ is a whole-body posture without collision;
- $[\mathbf{p}_e(t),\ \boldsymbol{\theta}_e(t)]$ is position-orientation vector of a controlled end-effector.

## 6.2.2 Collision-free posture generator

### Posture generator overview

A *posture generator* [EKM06] is used in this work (see Fig. 6.2) for solving the problem stated in previous section. By imposing configurations of one or more bodies as well as the CoM position, the posture generator can compute a whole-body configuration (if it exists) that is collision-free using optimization technique. A SQP (Sequential Quadratic Programming) optimization solver is used for solving this non-linearly constrained problem.

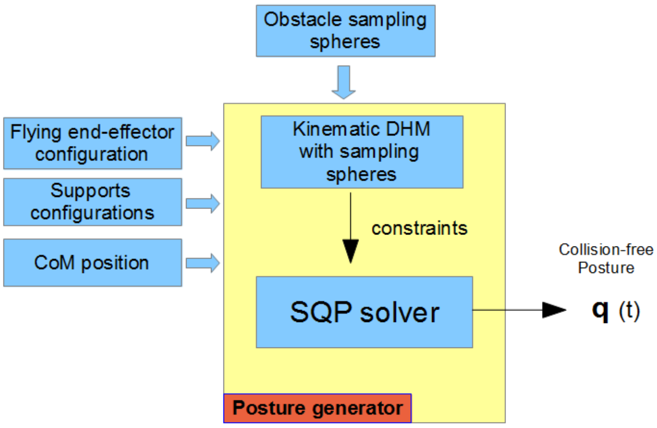


**Figure 6.2:** Posture generator for computing whole-body posture with multiple constraints such as CoM position, body configuration, collision avoidance, etc.

### Collision avoidance

Collision avoidance is integrated in the posture generator as a series of inequality constraints. In order to save computing time, we adopt a sphere-sphere model for collision checking which can ensure a continuous gradient for distance functions (see Appendix F). A series of bounding spheres are sampled for some body segments of the DHM as well as each obstacle (see Fig. 6.3). To impose collision avoidance between a body segment and an obstacle, the distance between each pair of their sampled spheres must be positive. Self-collision avoidance for a pair of body segments can be imposed in the same way.

A bounding sphere is represented as a 4-dimension vector, namely the position

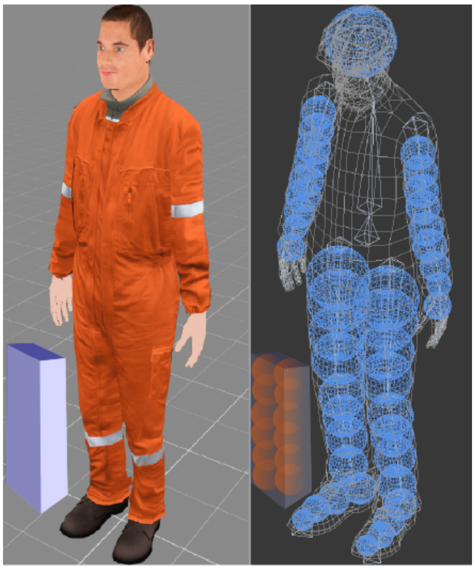


**Figure 6.3:** Collision avoidance: some body segments of the DHM and the obstacles are represented by a series of spheres for collision checking.

of its center as well as its radius:

$$\mathbf{s}_i = \begin{bmatrix} \mathbf{p}_i \\ r_i \end{bmatrix} = \begin{bmatrix} x_{ci} \\ y_{ci} \\ z_{ci} \\ r_i \end{bmatrix} \tag{6.3}$$

Thus a collision occurs when there is at least a pair of spheres intersecting with each other, in other words, when the distance between centers of a pair of spheres is smaller than the sum of their radius.

### 6.2.3 Flying end-effectors

We provide two options for solving the IK problem with open chains of flying end-effectors: interpolating method and sampling-based motion planning method.

We recall that the initial and final configurations of a flying end-effector in each of its flight phases are already given in the pre-defined support configuration sequence (see Fig. 6.1). Meanwhile, timing of each flight phase has also been generated at the first level.

An option for solving the problem is to interpolate the flying end-effector's trajectory using polynomial functions. This trajectory then is inputted into the posture generator for computing the whole-body collision-free motion in this flight phase. This method is rapid but can only be used in rather simple cases, such as a small step in a local environment without nearby obstacles.

The second option is to use sampling-based planning method. Since the initial and final configurations as well as the timing are known, we choose *Bi-RRT* (Bi-directional Rapidly-exploring Random Trees) method for motion planning in this case. Instead of exploring DHM whole-body $\mathcal{C}$-space (configuration space), our Bi-RRT method is implemented in the flying end-effector's configuration-time space ($\mathcal{C}$-$t$ space) which is a low-dimension (7D) space (see Fig. 6.4). The time dimension in this method makes it possible to impose kinematic/kinodynamic constraints. The metric distance in the sampling space is computed by balancing position, orientation and time dimensions. An upper limit is imposed for average velocity of end-effector in our method to avoid brutal change in postures. Of course, both trees must expand in an unique direction along the time axis (time-increasing for the first tree and time-decreasing for the second one). The Bi-RRT algorithm is presented in detail in next section.

### 6.2.4 Bi-RRT

In this section, we present the detailed Bi-RRT planning method. Its basic conception is shown in Fig. 6.4.

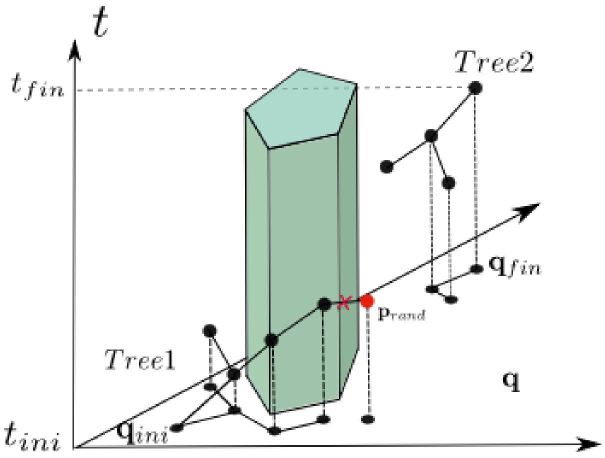


**Figure 6.4:** Bi-RRT in the configuration-time space. The $\mathcal{C}$-obstacle sweeps the time interval and results in a hyper-obstacle in the configuration-time space. The extra dimension makes it possible to impose the kinetic/kinodynamic constraints (e.g. maximum velocity constraint). Of course, trees must expand in an unique time direction (time-increasing for $Tree1$ and time-decreasing for $Tree2$). A newly sampled node (red) is rejected if the average velocity for the new path surpasses the limit.

#### Bi-RRT algorithm

Algorithm 4 addresses the general structure of the Bi-RRT method used in our study. The two trees are initialized by adding their first node with respectively the given configuration as well as its corresponding time. Since the first node in each tree serves as the root, we impose its *parent* to be null. The two trees then will generate random sampling nodes and try to connect with each other within at most $n_{it}$ iterations. Function *RANDOM-POINT()* serves to generate the random sampling point $\boldsymbol{\chi}_{rand}$ within a user-defined sampling zone. Then the function *EXPAND()* will try to generate a new node towards the sampling point $\boldsymbol{\chi}_{rand}$, as introduced in Algorithm 5. Function *NEAREST-NODE()* returns the node $\boldsymbol{\chi}_{near}$ of a tree that is nearest to the new random point. *CHECK-VELOCITY()* checks whether the newly sampled point violates the average velocity limit; if not, *LIMIT-METRIC()* generates a new point (candidate node) $\boldsymbol{\chi}_{new}$ lying on the line connecting $\boldsymbol{\chi}_{near}$ and $\boldsymbol{\chi}_{rand}$ and within a limited metric distance with $\boldsymbol{\chi}_{near}$. Then function *CHECK-PG()* verifies feasibility of the candidate path segment ($\boldsymbol{\chi}_{new}$ to $\boldsymbol{\chi}_{near}$): $n_{pg}$ interpolating points at the path segment are inputted one by one into posture generator for imposing the flying end-effector's configuration. If no feasible posture is found, function *EXPAND()* will stop and the candidate node $\boldsymbol{\chi}_{new}$ then is rejected. If function *CHECK-PG()* returns the confirmative signal, $\boldsymbol{\chi}_{new}$ is then added into the tree and its *parent* is labeled as the index of the node $\boldsymbol{\chi}_{near}$. Function *GET-INDEX()* returns the index of a node in the node array of a tree.

**Algorithm 4** GENERATE-RRT($\mathbf{q}_{ini}, t_{ini}, \mathbf{q}_{fin}, t_{fin}$)

```
$Tree1.node = (\mathbf{q}_{ini}, t_{ini}), Tree2.node = (\mathbf{q}_{fin}, t_{fin})$
$Tree1.parent = 0, Tree2.parent = 0$
for $i = 1$ to $n_{it}$ do
    $\boldsymbol{\chi}_{rand} = [\mathbf{q}_{rand}, t_{rand}] \leftarrow$ RANDOM-POINT()
    EXPAND($Tree1$, $\boldsymbol{\chi}_{rand}$)
    if IS-CONNECTED($Tree1$, $Tree2$) then
        break
        return $CONSTRUCT\_PATH()$
    end if
    EXPAND($Tree2$, $\boldsymbol{\chi}_{rand}$)
    if IS-CONNECTED($Tree1$, $Tree2$) then
        break
        return $CONSTRUCT\_PATH()$
    end if
end for
```

The two trees connect to each other when the function *IS-CONNECTED()* returns a confirmative result. This function examines the nearest distance between the two trees and verifies the condition of connection for the nearest pair of nodes. It

**Algorithm 5** EXPAND($Tree$, $\boldsymbol{\chi}_{rand}$)

```
$\boldsymbol{\chi}_{near} \leftarrow$ NEAREST-NODE($Tree$, $\boldsymbol{\chi}_{rand}$)
if CHECK-VELOCITY($\boldsymbol{\chi}_{near}, \boldsymbol{\chi}_{new}$) $= TRUE$ then
    $\boldsymbol{\chi}_{new} \leftarrow$ LIMIT-METRIC($\boldsymbol{\chi}_{near}, \boldsymbol{\chi}_{rand}, d_{lim}$)
    for $i = 1$ to $n_{pg}$ do
        $e \leftarrow$ CHECK-PG($Tree.\boldsymbol{\chi}_{near}, \boldsymbol{\chi}_{new}$)
        if $e =$ FALSE then
            Break
        end if
    end for
    $Tree.node$.PUSHBACK($\boldsymbol{\chi}_{new}$)
    $Tree.parent$.PUSHBACK(GET-INDEX($\boldsymbol{\chi}_{near}$))
end if
```

is called whenever a new node is added into a tree. If the smallest distance between the two tree is smaller than a threshold value, the functions *CHECK-VELOCITY()* and *CHECK-PG()* are called to verify the kinematic constraint and the feasibility of the connecting path for this pair of nodes. If valid, the algorithm connects the two trees, stops the iterations and returns the solution path.

### Metric distance in Bi-RRT method

Without loss of generality, the configuration of a controlled end-effector consists of the position and the orientation regarding the global coordinate frame:

$$\mathbf{q}_e = [\mathbf{p}_e, \boldsymbol{\theta}_e] = [x, y, z, \alpha, \beta, \gamma] \tag{6.4}$$

The time dimension should be also included for dynamic motion planning, thus the representation for the end-effector's state is expressed as:

$$\boldsymbol{\chi} = [\mathbf{q}_e, t] \tag{6.5}$$

Relying on the $Rodrigues'\ rotation\ formula$ in Equation (2.14), we define the metric distance between two points $\boldsymbol{\chi}_1$ and $\boldsymbol{\chi}_2$ in the $\mathcal{C}$-$t$ space as:

$$d(\boldsymbol{\chi}_1, \boldsymbol{\chi}_2) = \sqrt{\left\|\mathbf{p}_{e1} - \mathbf{p}_{e2}\right\|^2 + \lambda_1 \left\|log(\mathbf{R}(\boldsymbol{\theta}_{e1}, \boldsymbol{\theta}_{e1}))\right\|^2 + \lambda_2 (t_1 - t_2)^2} \tag{6.6}$$

where:

- the scalars $\lambda_1$ and $\lambda_2$ are factors for balancing the relative weights among translation, rotation and time metrics.

In order to limit the translational and rotational velocity, we need to compute the change rate of the configuration (can be regarded as the slope regarding the time-axis):

$$v(\boldsymbol{\chi}_1, \boldsymbol{\chi}_2) = \frac{\|\mathbf{p}_{e1} - \mathbf{p}_{e2}\|}{|t_1 - t_2|} \tag{6.7}$$

$$\omega(\boldsymbol{\chi}_1, \boldsymbol{\chi}_2) = \frac{\|log(\mathbf{R}(\boldsymbol{\theta}_{e1}, \boldsymbol{\theta}_{e2}))\|}{|t_1 - t_2|} \tag{6.8}$$

### 6.2.5 Post-processing

One of the main weaknesses lying in the sampling-based methods, particularly the RRT, is the lack of smoothness for the generated path. In this study, a non-smooth path would result in a final whole-body motion with sudden variations of speed. Thus a RRT smoothing algorithm is implemented in order to get a whole-body motion more natural. The function $SMOOTHING()$ (Algorithm 6) carries out the path filtering in an iterative way: given the raw path, it samples randomly two points that lie on two different segments of the path. The straight line that links the two sampling points is regarded as a candidate segment. Then the functions *CHECK-VELOCITY()* and *CHECK-PG()* are carried out for the new candidate path segment to verify its feasibility at discretized points on it. The number of the discretized points is proportional to the time span between the two points. If it is validated, the path is updated by replacing all portions between the two points on the path with this new segment. This smoothing operation is carried out iteratively until the desired number of iteration is reached.

## 6.3 Dynamic simulation: execution of the generated motion

Once the work presented in previous sections has been successfully carried out, we then have obtained the trajectories of the CoM and the end-effectors. The whole-body collision-free postures are also generated subsequently. All these data are thereafter used as motion references for executing the generated movement in dynamic simulations.

Several tasks are defined for the controller in the dynamic simulation. These tasks result in multiple optimization objectives which are weighted according to their priorities. The tracking tasks are itemized as follows:

- **Center of mass**: the dynamic balance is guaranteed by the global CoM trajectory, thus the CoM of the DHM should track precisely its pre-defined robust trajectory, including the position, velocity and acceleration references;
- **End-effector**: the configuration of a controlled end-effector should track its trajectory generated by local planners, including the position, orientation, translational and angular velocities and accelerations;

**Algorithm 6** SMOOTHING($Path = \{\boldsymbol{\chi}_0, \boldsymbol{\chi}_1, \cdots, \boldsymbol{\chi}_s\}$)

```
Get the number of segments in this path: $s = SIZE(path)$.
for $i = 1$ to $n_{it}$ do
    Two random integers $n_1, n_2 \in [1, s]$ and $n_1 < n_2$.
    Randomly choose two points lying respectively on the $n_1$-th and $n_2$-th segment of the path: $\boldsymbol{\chi}_{r1}$ and $\boldsymbol{\chi}_{r2}$.
    if CHECK-VELOCITY($\boldsymbol{\chi}_{r1}, \boldsymbol{\chi}_{r2}$) $= TRUE$ then
        for $i = 1$ to $n_{pg}$ do
            if $CHECK - PG(\boldsymbol{\chi}_{r1}, \boldsymbol{\chi}_{r2}) = FALSE$ then
                Break
            end if
        end for
    end if
    $Path = \{\boldsymbol{\chi}_0, \boldsymbol{\chi}_1, \cdots, \boldsymbol{\chi}_{n_1-1}, \boldsymbol{\chi}_{r1}, \boldsymbol{\chi}_{r2}, \boldsymbol{\chi}_{n_2} \cdots, \boldsymbol{\chi}_{fin}\}$
end for
```

- **Supports**: a contact or a grasp should be activated or deactivated during the movement according to the motion strategy. The wrenches applied at a contact or a grasp are taken into account in the motion control strategy;
- **Posture**: the joints are also controlled and the reference posture is supplied in our framework; moreover, there is an additional reference that drags the root body towards a vertical orientation.

## 6.4 Example: implementation in a complex scenario

An example of a DHM motion in a complex scenario has been presented in section 5.4.4. For now, we have obtained a global robust CoM trajectory in the structured environment shown in Fig. 5.18 and Fig. 5.19. Here, we carry on planning a whole-body collision-free motion and executing this motion in a dynamic simulation.

### 6.4.1 Local planning

In this motion, the controlled end-effectors are two feet of the DHM. As described in Table 5.3, the DHM makes three steps during the movement, meaning that this motion has three foot-flying phases. As designed for this scenario, there is no obstacles involved in the first step of the left foot. Thus a parametric curve for the left foot if firstly implemented for the position of the left foot, and we suppose that it has no orientation variation during this step. The foot trajectories are generated for the ankle centers of the feet whose position is shown in Fig. 6.6. The posture generator then has validated this curve by computing successfully whole-body postures for all the sampling instants during this step. Bi-RRT methods are then implemented for the second and the third steps. The bounding spheres for collision checking are defined by hand, as shown in Fig. 6.5.

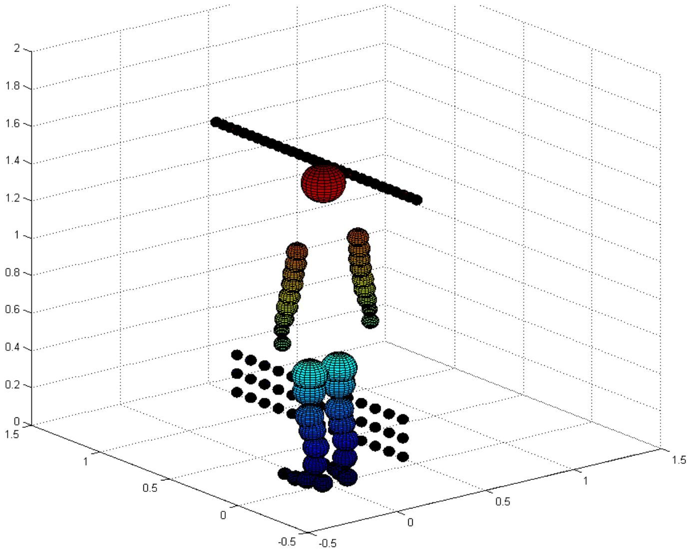


**Figure 6.5:** Demonstration of the bounding spheres for collision detection. For the roof, since the geometric constraints are defined for the CoM position, bounding spheres are only sampled for its left edge.

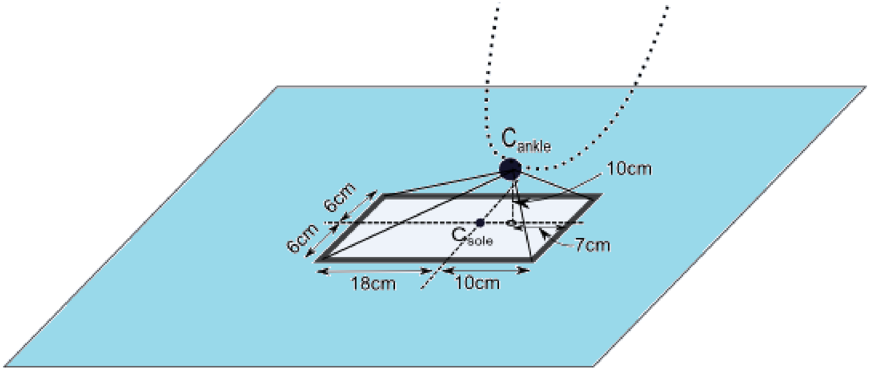


**Figure 6.6:** The foot model of the DHM. It is for the ankle joint center ($C_{angke}$ in the figure) that the local planner generates the reference trajectory.

The final end-effector trajectories in the motion are shown in Fig. 6.7 (only positions are shown).

The trajectories of joints and end-effectors are finally exported with a sampling time $t = 1ms$ and filtered by a low-pass filter with a cutting frequency of $10Hz$.

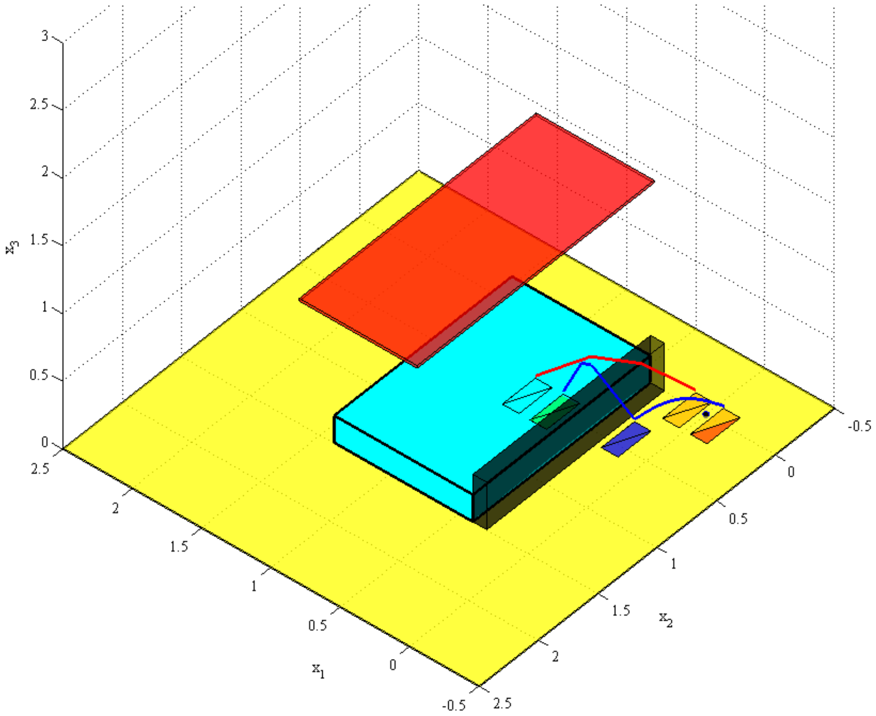


**Figure 6.7:** The end-effector trajectories (for left foot and right foot) are shown in this figure. The reference point are chosen to be the ankles of the feet that are 10$cm$ higher over the sole. The blue trajectory for the left foot consists of two steps: the first one is generated by parametric curve and the second one is generaged using Bi-RRT approach. The red trajectory for right foot consists of only one step and it is generated by Bi-RRT approach.

### 6.4.2 Motion simulation

The dynamic simulation is carried out using the software XDE-Dsimi® developed by laboratory LSI of CEA-LIST. The results generated at the first and second levels are imported into the simulation software and used as control references. Several tracking tasks are then defined for the dynamic controller in the dynamic simulation that are shown in Table 6.1. The gains and weights for each impedance control task are manually tuned. The time step of the simulation is set to be $T = 5ms$.

Some clips of the successful simulation are given in Fig. 6.8. The tracking errors for the CoM and the two feet are traced. One can see the tracking performances in Fig. 6.9 and Fig. 6.10. These results show that the DHM tracks quite well its reference CoM trajectory. Though there are slight errors in feet tracking tasks, the DHM realizes its motion without losing balance. The feasibility and the robustness of our framework are validated by this simulation.

| Multiple tasks in the motion simulation | | |
|---|---|---|
| Task | References | Weight |
| CoM Tracking | Position (3d)<br>Velocity (3d)<br>Acceleration (3d) | 2e4 |
| Feet Tracking | Position-orientation (6d)<br>Velocity (6d)<br>acceleration (3d) | 5e4 |
| Basin | Root body orientation (3d) | 1e4 |
| Posture | Joint angles (39d) | 10 |

**Table 6.1:** Tasks definition in the dynamic simulation for motion execution.

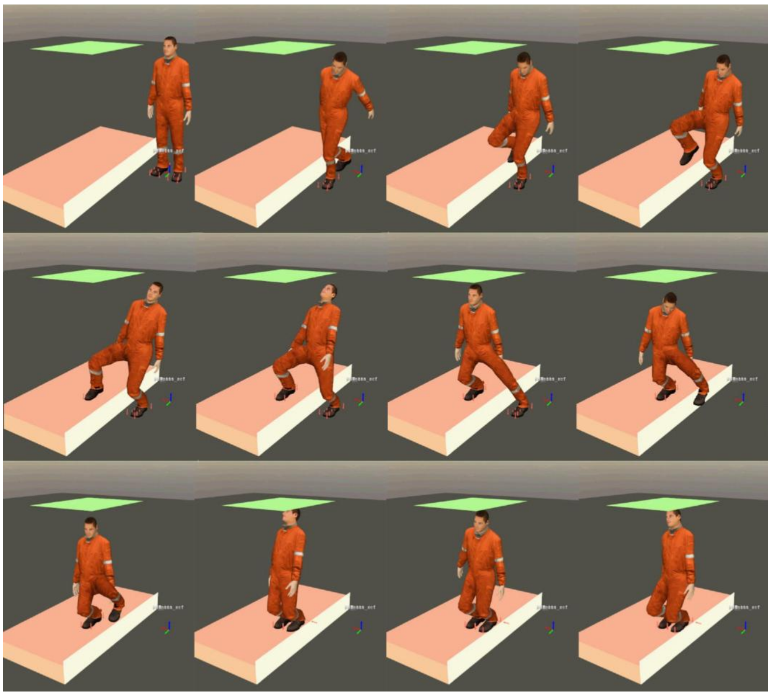


**Figure 6.8:** Clips of the dynamic simulation with the DHM in the complex scnenario under XDE®.

## 6.5 Conclusion

In this chapter, we present the the second and the third levels of the hierarchical framework.

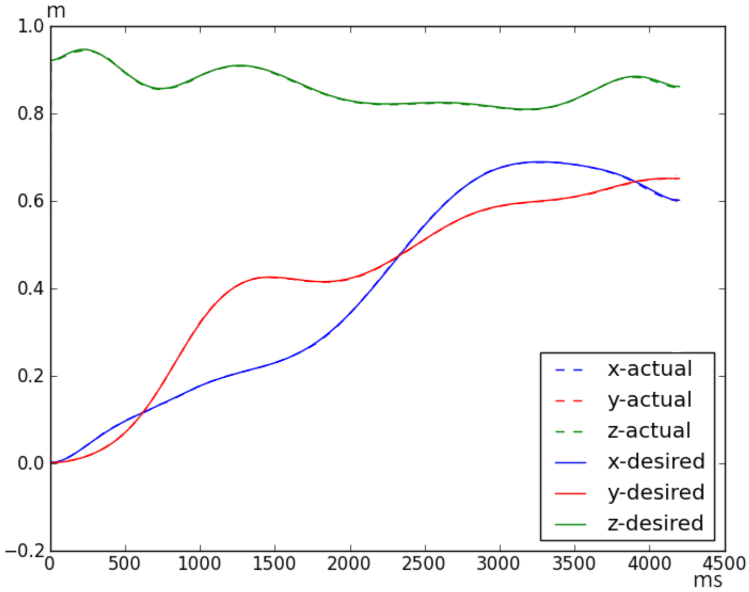


**Figure 6.9:** Comparison of the reference and the actual CoM trajectories in the simulation.

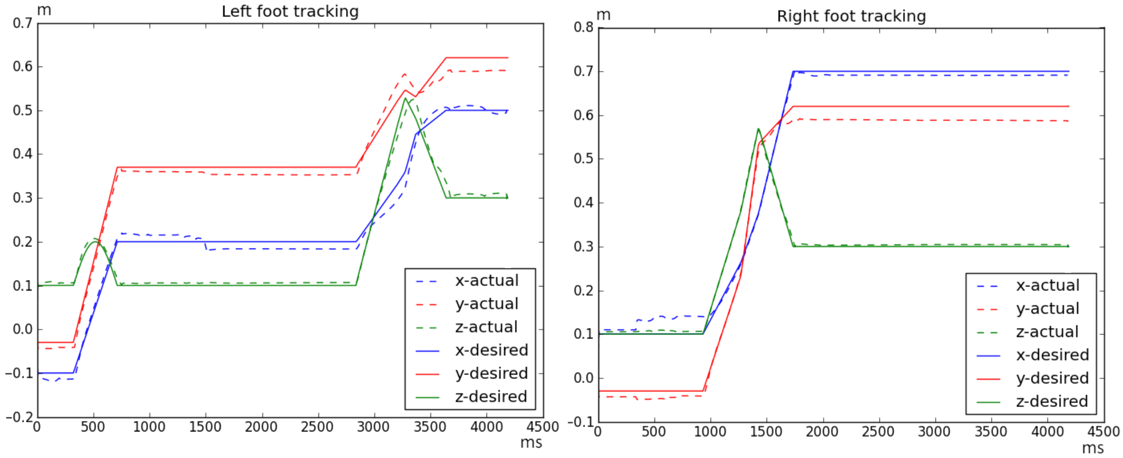


**Figure 6.10:** Comparison of the reference and the actual feet trajectories in the simulation.

At the second level of the framework, we plan locally a whole-body collision-free motion of DHM in a cluttered environment. The CoM trajectory generated at the first level serves as an constraint for whole-body motion planning. An optimization-based posture-generator is used to solve whole-body collision-free posture sequence for the DHM while respecting all constraints. We propose two methods for dealing with open-chain IK problem brought by an flying end-effector. An interpolation method is used in rather simple cases. For general cases, a sampling-based planning method is proposed for generating the flying end-effector's trajectory in a cluttered local environment. This level yields trajectories of end-effectors and whole-body postures during the motion.

At the third level, the results generated at the first two levels are used as control references for a dynamic DHM. The associated dynamic controller defines a series of tracking tasks for CoM, feet, hands (if there are hand grasps), whole-body postures, etc. Dynamic controller solves articular torques to actuate the DHM so that it realizes the planned motion in simulation.

We carry on the example of a DHM moving in a complex scenario presented in Chapter 5. The DHM finally successfully steps onto the platform under a roof without losing balance and collisions.

Chapter 7

# Case study: DHM motion simulation in a car-ingress scenario

In the previous chapters, we have introduced our theoretical and algorithmic work for motion planning and simulation in complex environments. By studying and simulating a car-ingress scenario, we present here the practical application of our framework.

This chapter is organized as follows. Section 7.1 introduces a car-ingress scenario and an example of human motions in this scenario; based on this example, Section 7.2 presents the application of the framework; we evaluate the simulated motion in Section 7.3 by comparing it with the recorded motion and discuss the differences; Section 7.4 concludes this chapter and suggests future improvements.

## 7.1 Scenario introduction

MoCap experiments of car-ingress-egress motions have been conducted during former project and thesis works ([DHE], [Cau11], [Cha09]) in LBMC. Fig. 7.1 shows several clips of human car-ingress motion and the reconstructed motion in one of the experiments. As observed and asserted in [Cha09], there are two global motion strategies for a human entering the vehicle:

1. Right leg first (about 90%);
2. Sitting first (about 10%).

In our case study, we have carried out the motion analysis work based on one of the recorded car-ingress motion of the "right leg first" strategy. A series of clips of its reconstructed motion is shown in Fig. 7.2. We have firstly estimated the placements and the sequence of the support configurations established in this motion.

By detecting the variations of the supports, the following key frames have been extracted:

1. Beginning of motion

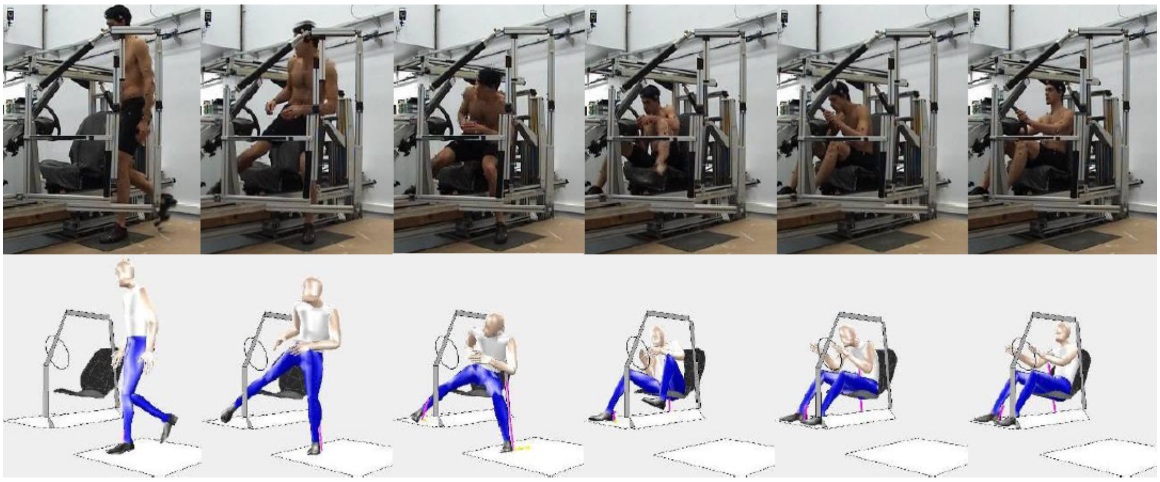


**Figure 7.1:** Clips of a car-ingress motion in MoCap and the reconstructed motion in RPx (from thesis of Julien CAUSSE [Cau11]).

2. Left foot lift-off
3. Left foot landing
4. Right Foot lift-off
5. Right hand grasps the steering wheel
6. Right Foot landing
7. Buttock touches the seat
8. Left foot lift-off
9. Right hand leaves the steering wheel
10. Right hand grasps the steering wheel
11. Left foot landing on car floor
12. Left hand grasps the steering wheel
13. End of motion

In the following sections, we present the application of our framework in a car-ingress motion simulation relying on the extracted knowledge obtained from this motion.

## 7.2 Application of the framework

### 7.2.1 Preliminary work

#### Virtual environment

A DHM is expected to realize its movement in a virtual environment which represents well the real environment of the scenario. To simplify this problem, a structured environment containing simplified objects is constructed and used in our work.

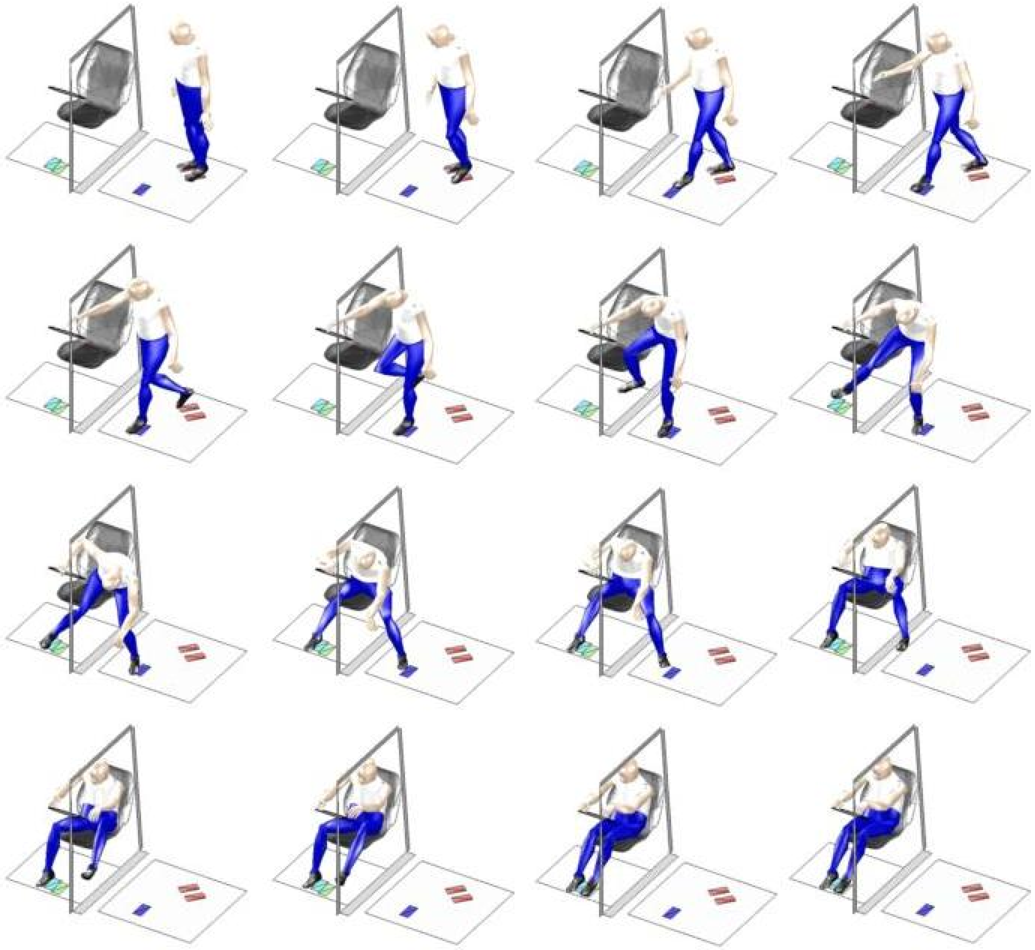


**Figure 7.2:** Clips of the car-ingress motion with "right leg first" strategy. Estimated foot contacts are shown as colored rectangles.

In this car-ingress scenario, 8 objects are characterized as shown in Fig. 7.3. The geometric primitives of the environment are given in Tabel 7.1.

### Obstacles

A series of spheres are sampled for collision detection. In our framework, a file containing sampling spheres characteristics (indices, centers and radius) is generated. In the following work, the obstacle-body pairs for collision detection can be defined according to the motion strategy.

In the car-ingress case, the floor and the ground are not considered as obstacles thus we have defined 6 obstacles (as shown in Fig. 7.4). The number of spheres for each obstacle is given in Table 7.2.

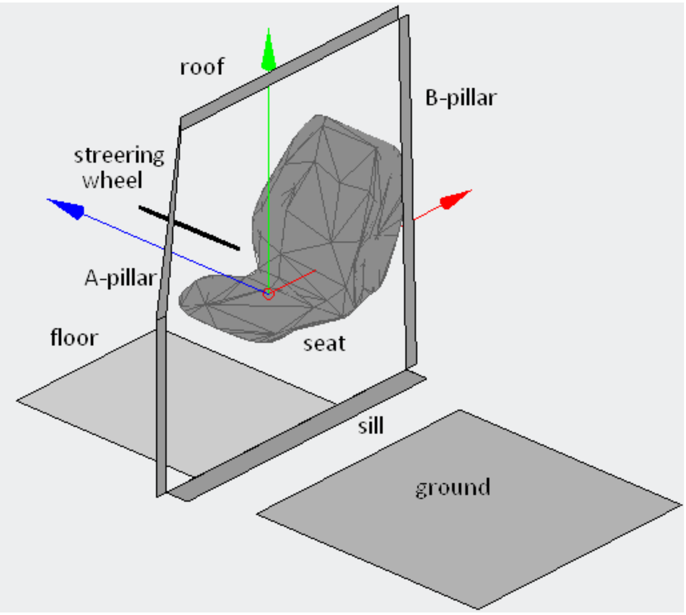


**Figure 7.3:** The structured virtual environment in which the DHM is designed to realize its motion (Interface snapshot of RPx$^{®}$ software). The origin of the world frame is chosen at the H-point of the car-seat.

| Environmental element | Geometric primitives |
|---|---|
| H point | [0, 0, 0] |
| Ground | $z = -485mm$ |
| Floor | $z = -231mm$ |
| Sill | $y = -402mm$, $z = -231mm$ |
| Roof | $y = -193mm$, $z = 934mm$ |
| Steering wheel | $d = 360mm$, $\mathbf{c}$ = [-402, -11, 409], $\theta_{incli} = 55.0°$ |
| B-pillar | $x = 237mm$, $\theta_{incli} = 80.7°$ |
| A-pillar | $x = -942mm$ |
| Seat | A set of vertices |

**Table 7.1:** Environment primitives specification in the car-ingress example.

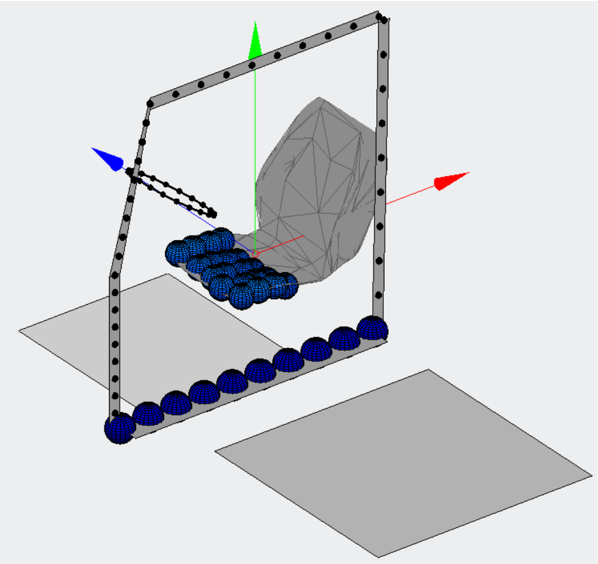


**Figure 7.4:** Automatically sampled spheres for obstacles in the structured environment.

| ID | Obstacle name | No. of sampled spheres |
|---|---|---|
| -1 | Sill | 10 |
| -2 | A-pillar | 20 |
| -3 | B-pillar | 10 |
| -4 | Seat | 20 |
| -5 | Steering wheel | 21 |
| -6 | Roof | 10 |

**Table 7.2:** Summary of obstacles and the numbers of sampled spheres.

### Support configuration

The sequence and placements of support configurations are pre-defined. We can generate manually a sequence of support configurations or extract them from experiments ([AEMPG+08], [CW10], [Cha09]). It is also possible to automatically generate the supports sequence using some supports planning methods proposed in recent researches (e.g [KJNK+01], [CLC+05], [EKM06]).

In this example, we have extracted beforehand the support configuration from the MoCap database (see Fig. 7.5). Placements and sequence of the support configurations are given respectively in Table 7.3 and Table 7.4.

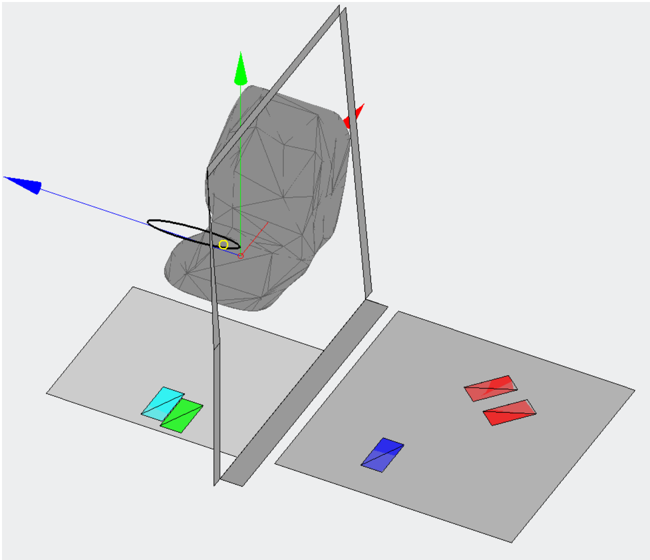


**Figure 7.5:** Supports sequence illustration: footsteps (rectangles) and right hand grasp position (yellow circle).

## 7.2.2 Global CoM trajectory generation

### Motion decomposition

When the buttock touches the seat, the human will be very safe from losing balance. Therefore, in this motion, we consider the instant when the hip touches the seat as end of the motion. Thus at the first level, we generate the CoM trajectory for the period from the beginning of the motion to the instant when the buttock touches the seat. Accordingly, the first 6 support configurations (see gray columns in Table 7.4) are used for computing the balance constraints (polytopes) for the 6 transition phases.

To completely achieve the car-ingress movement, after the buttock touches the seat, the DHM still needs to place its left foot into the car and build (rebuild) the hand grasps. Since these manipulations need little balance considerations, they can be realized using local planning without CoM trajectory imposition.

| Supports placements in the car-ingress scenario | | |
|---|---|---|
| Support | Position ($m$) | Orientation (Euler angles $x$-$y$-$z$) |
| Left foot | [0.191, -1.064, -0.485] | [0, 0, 50°] |
| | [-0.240, -0.737, -0.485] | [0, 0, 90°] |
| | [-0.651, -0.105, -0.232] | [0, 0, 90°] |
| Right foot | [0.283, -0.937, -0.485] | [0, 0, 50°] |
| | [-0.624, -0.012, -0.232] | [0, 0, 90°] |
| Right hand | [ 0.372, -0.832, 0.270] | – |
| | [-0.460, -0.056, 0.569] | – |
| | [-0.423, 0.144, 0.475] | – |
| Left hand | [-0.438, -0.116, 0.513] | – |

**Table 7.3:** Sequence of support configurations for car-ingress scenario.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Left foot | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| Right foot | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Left hand | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Right hand | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| Buttock | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

**Table 7.4:** Sequence of support configurations. ‘1’ indicates that a support is established, ‘0’ indicates non-established. The 6 support configurations in gray are used for generating the global CoM trajectory.

### Admissible pseudo-wrench spaces

As presented in Chapter 3, each support configuration determines an admissible pseudo-wrench subspace which constrains the motion balance. Once the user-supplied sequence of support configurations is given, the module "Polytope computation" (PC, see Appendix E) automatically generates a sequence of polytopes and save their H-representations matrices in a file. This file then is used by the CoM trajectory generator for defining the balance constraints in the optimization solver, as presented in Chapter 5.

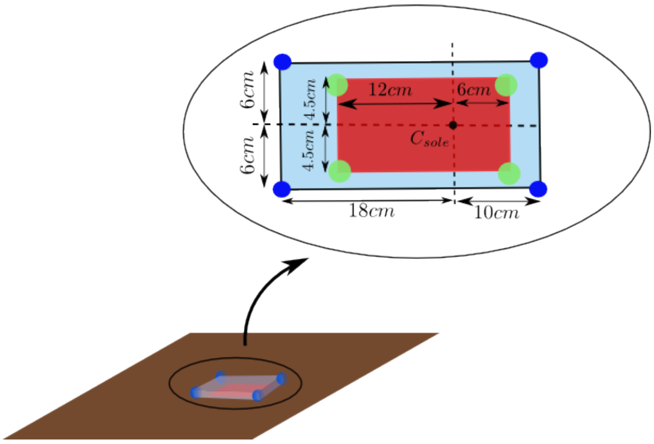


**Figure 7.6:** Different sole sizes for respectively balance criterion calculation and dynamic simulation. The larger rectangle is the sole model of the DHM that has been presented in Fig. 6.6 and the blue circles representing its contact points. The red rectangle is the sole model that is used for computing balance criterion and generating CoM trajectory; the green circles represent its contact points.

In this step, parameters such as sole dimension, force limit for the sum of normal contact forces and force limit for a grasp force should be specified by user. In this application, the forces limits are specified as: $1kN$ for the sum of normal contact forces and $100N$ for the force norm limit of the right hand grasp. In this step, 6 polytopes have been computed.

Though the objective for maximizing the stability margin makes the generated CoM trajectory robust for resisting CoM trajectory tracking errors in motion simulation (at the execution level), the DHM may also lose balance during the motion simulation because of support placement errors: support placement errors may result in different admissible pseudo-wrench spaces (polytopes for motion simulation) from those computed at first level (polytopes for trajectory calculation), Therefore at first level, foot sole dimensions (see Fig. 7.6) as well as the friction coefficients are chosen to be smaller than those for simulation. So in spite of slight foot placement errors, a polytope for trajectory calculation will stay inside the corresponding polytope for simulation and the CoM trajectory will not violate the balance constraints in simulations.

The feasibility of the CoM trajetory generation is assessed by the 6 consecutive

| Variable | Value | Indication |
|---|---|---|
| $\mathbf{x}_i$ | [0.2350, -1.0112, 0.4659]$(m)$ | Initial Position |
| $\mathbf{x}_f$ | [-0.045, -0.1463, 0.210] $(m)$ | Final Position |
| $n_t$ | 6 | No. of Transition Phases |
| $k$ | 5 | Degree of Spline |
| $\mathbf{t}_{lb}$ | [0.15, 0.4, 0.1, 0.5, 0.6, 0.6] $(s)$ | Lower bound for time spans |
| $l_{lim}$ | 1.08$(m)$ | Distance limit for CoM/Footprint |

**Table 7.5:** Motion specification of the car-ingress example.

polytopes, or more precisely, by the smallest volume of their intersections. We have computed the 5 intersections (which are also polytopes) of the 6 polytopes in this example. The smallest Chebychev ball radius of the 5 intersections is $32.63N \cdot m$.

### Global CoM trajectory generation

As presented in Chapter 5, we use a timing-free method for generating a global CoM trajectory which ensures the dynamic balance of the motion. The user has to specify the following parameters:

- Initial and final positions (or even velocities and accelerations);
- Spline parameters;
- Lower bound of the timing vector;
- Geometric constraint parameters;
- Optimization parameters;

The specified parameters in this example are given in Table 7.5. The initial CoM height is chosen to be $6cm$ smaller than the real height in the recorded motion ($0.526m$). Reason of this choice will be argued in the last section. In this example, we do not impose the final velocity to be zero. Moreover, we increased the weighting factor for the jerk objective in z direction in order to get smoother height variations (geometric constraint on CoM height brought by the roof is not defined). A geometric constraint is imposed which limits the distance from CoM to the foot contact center within 1.08m, according to the analysis result in Chapter 3. The CoM trajectory generator takes also the timing of the movement into account as presented in Chapter 5. The minimum time durations for each phase are set to be $[0.15, 0.4, 0.1, 0.5, 0.6, 0.6](s)$ according to experimental motion data.

The CoM trajectory generation is carried out in Matlab (on a workstation with Xeon 3.4GHz and 8GB RAM) and the trajectory has been successfully generated within 3 minutes (see Fig. 7.7). The resulting parameters are given in Table 7.6. The generated 3D CoM trajectory, its position and velocity curves and stability margins are shown in Fig. 7.7, Fig. 7.8 and Fig. 7.9 respectively. This trajectory has a lower bound of $19.4N \cdot m$ for stability margins.

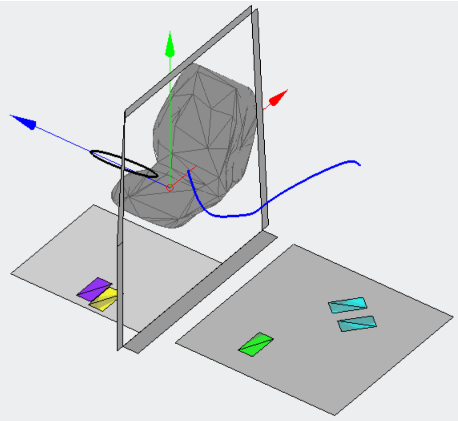


**Figure 7.7:** CoM trajectory illustration in the structured environment.

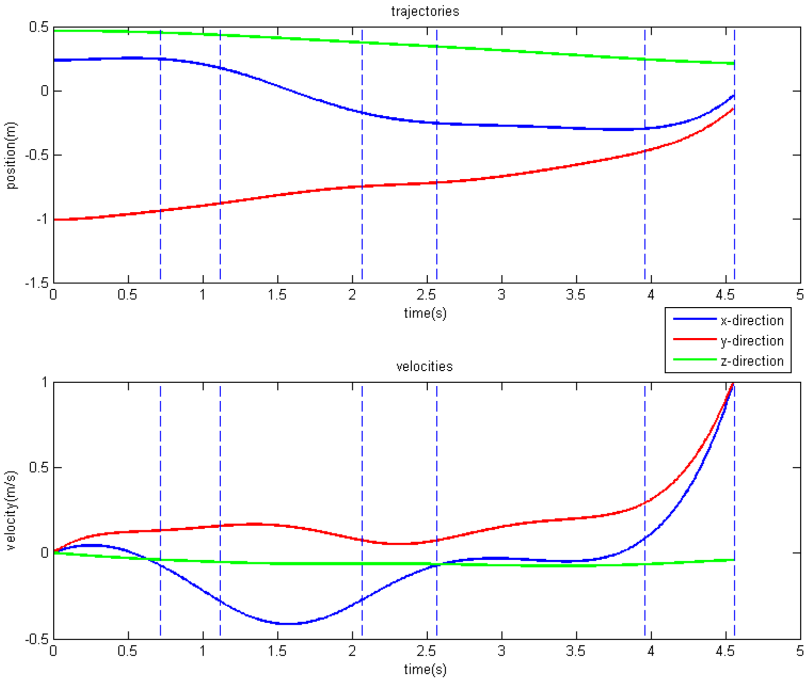


**Figure 7.8:** The generated robust global CoM trajectory for the car-ingress motion (above) and its velocity curves in respectively x, y and z directions.

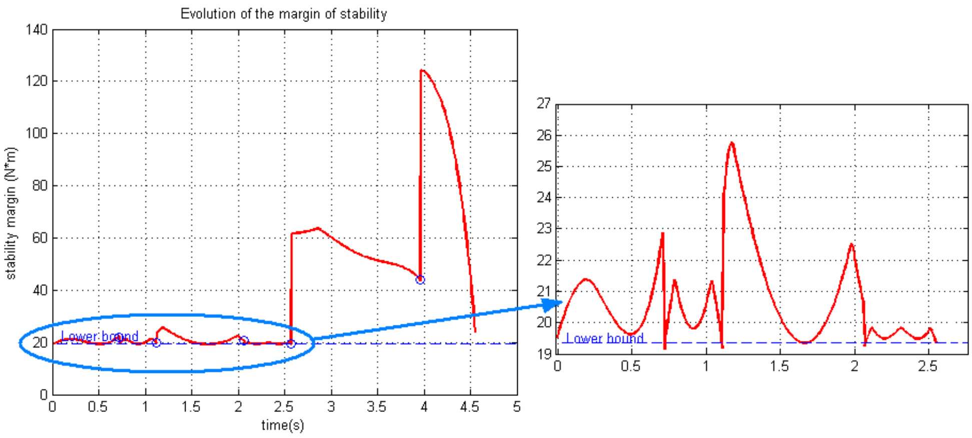


**Figure 7.9:** Trace the stability margin of the generated CoM trajectory. In the right figure, the zoomed stability margin curve for the first 2.56$s$ is demonstrated.

| Variable | Value |
|---|---|
| $\mathbf{a}_x$ | [0.2350, 0.2350, 0.2492, 0.2763, 0.1825, -0.3005, -0.2687, -0.2684, -0.3894, -0.1642, -0.0454] |
| $\mathbf{a}_y$ | [-1.0112, -1.0112, -0.9860, -0.9350, -0.8688, -0.6894, -0.7915, -0.5306, -0.5296, -0.2661, -0.1463] |
| $\mathbf{a}_z$ | [0.4659, 0.4659, 0.4635, 0.4505, 0.4260, 0.3694, 0.3299, 0.2706, 0.2336, 0.2153, 0.2104] |
| $knots$ | [0, 0, 0, 0, 0, 0, 0.1574, 0.2451, 0.4536, 0.5632, 0.8684, 1, 1, 1, 1, 1, 1] |
| $\mathbf{t}_{dist}$ $(s)$ | [0, 0.7176, 1.1176, 2.0679, 2.5679, 3.9592, 4.5592] |
| $d_{min}$ | $19.4N \cdot m$ |

**Table 7.6:** Results of the generated CoM trajectory.

### 7.2.3 Local motion planning

#### Key postures

Once the global CoM trajectory has been generated, we firstly test its eligibility at key frames using the posture generator. Since all the support configurations are given and the CoM trajectory have been generated, we can thus define at the key frames a set of kinematical constraints for the posture generator; the body-obstacle pairs and body pairs for collision avoidance are also defined for the posture generator according to the motion conditions at the key frames. If a whole-body collision-free posture can be found for each key frame, then we can carry on; if not, we need to re-generate the CoM trajectory by modifying some constraints. The generated key postures are saved and will be used as reference postures.

In the car-ingress example, there are 7 key postures (see Fig. 7.10) that has been generated with the CoM trajectory shown in Fig. 7.7 and Fig. 7.8.

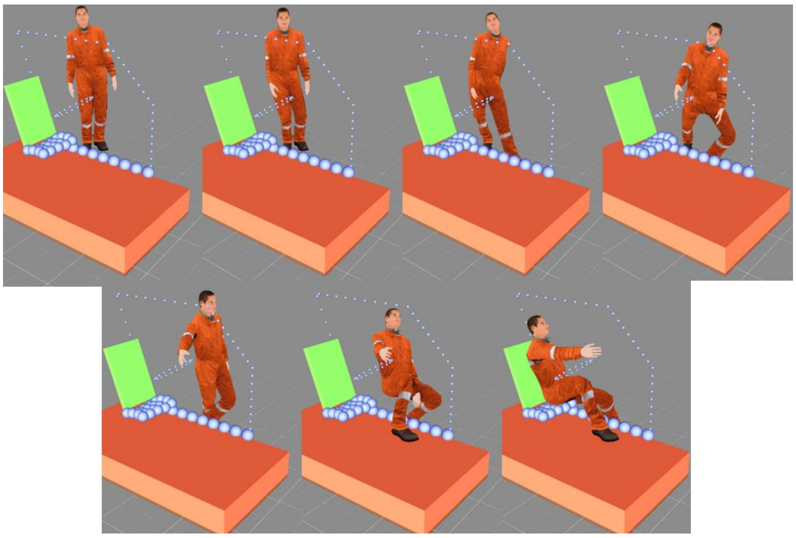


**Figure 7.10:** 7 postures generated at key frames before whole-body motion planning.

#### Whole-body motion planning

The whole-body motion is thereafter generated piecewisely for each transition phase. For a transition phase without displacements of end-effectors, the posture generator, generates a sequence of whole-body postures at user defined sampling instants, taking the pre-computed key posture as reference and taking the CoM trajectory as a constraint.

For a transition phase in which an end-effector makes a displacement, we have two methods for generating the whole-body collision-free motion. We firstly compute an interpolating curve for the flying end-effector using parametric curves or *slerp* (*spherical linear interpolation*) algorithm ([KKS95]). If this method fails to generate the motion (if DHM cannot find feasible postures with the interpolating trajectory at several instants), then we carry on using the Bi-RRT method. In both methods, the posture generator is implemented to verify the inverse kinematics and collision-freeness. The following parameters need to be specified in the Bi-RRT method according to the local environment of the flying end-effector:

- The upper limit for average velocity of the end-effector;
- The metric distance parameters and the radius for sampling new candidate nodes;
- The maximal number of iterations for RRT and smoothing;
- The body-obstacle and body-body pairs for collision avoidance.

In the car-ingress example, we have determined three controlled end-effectors: the two feet and the right hand. The right hand is supposed to move along a straight line between its initial position and the grasp position at the steering wheel. Thus its trajectory is generated by interpolating between the two position which is a 2nd-order polynomial function of time. The first step of the left foot is generated by parametric curve interpolation using trigonometric functions (see green curve in Fig. 7.11). The Bi-RRT method is applied for the two phases in which the DHM steps each foot into the car. During the left foot step-into-car phase, the buttock is always in contact with the seat, thus the CoM constraint is not taken into account in the posture generator. In several trials, the Bi-RRT can always successfully find the feasible paths in about 3-10 minutes for both the two flight phases. We give a discussion on the planning time in section 7.4.1. An example of exported smoothed trajectories for the two feet are shown in Fig. 7.11.

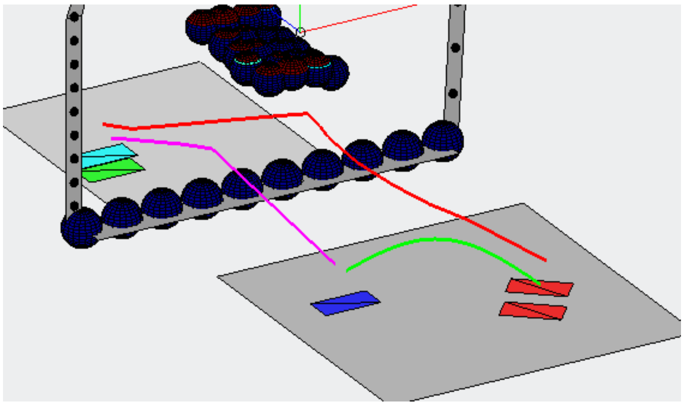


**Figure 7.11:** Feet trajectories generated at the second level for foot flight phases. The body frame of a foot is chosen at its ankle joint. Generated feet orientations are not shown in the figure.

### Trajectories extraction

We need to extract a set of trajectories (end-effector, CoM, joints) from the generated whole-body motion for executing it in the dynamic simulation. In this step, we choose a high sampling frequency so that we will have more choices for the time step of dynamic simulation.

The global CoM trajectory generated at the first level is parameterized as a B-spline whose parameters are solved using optimization techniques. To export the reference CoM trajectory, it is then sampled at a frequency of $1000Hz$ by computing the CoM position at each sampling instant. Its velocities and accelerations (represented also as B-splines) are computed and exported in the same way.

At the second level, once the trajectories of end-effectors are successfully generated, they are sampled at a frequency of $1000Hz$. Thereafter, their first and second order derivatives are obtained using numerical differentiation and smoothing in order to get the reference velocities and accelerations.

The whole-body posture sequence are exported at a $1000Hz$ by interpolation. Files storing all the exported data are then used at next level as references for the dynamic controller.

### 7.2.4 Dynamic simulation

The dynamic simulation is executed using the software XDE-Dsimi®. The simulation is launched with a file written in Python which carries out the following works:

- Construct the virtual environment by importing and configuring 3D models of objects;
- Import the pre-generated references;
- Import the DHM and initialize its position and posture;
- Initialize the dynamic controller and define its multiple tasks;
- Regulate the parameters (weights, gains, etc).

Fig. 7.12 shows a snapshot of the simulation interface demonstrating the virtual structured environment and the virtual digital human model in its initial position.

In this simulation, the virtual environment is constructed by importing into the virtual space a series of 3D models. To better evaluate the motion in terms of collision avoidance, we construct the obstacles by importing models of their sampling spheres (blue spheres in Fig. 7.12).

The time step in the dynamic simulation is set as $5ms$. Files storing the exported data generated at the first two levels are imported in this step to define references for the impedance controller. A series of tasks are defined including: CoM tracking, end-effector tracking, end-effector contacts/grasps, reference posture tracking. An additional task pushing the root body towards a vertical posture is also defined. The gains and weights of the tasks are manually tuned.

The car-ingress motion has been successfully realized. A set of clips of the dynamic motion are shown in Fig. 7.13. Tracking results for the CoM and the feet trajectories are shown in Fig. 7.14 and Fig. 7.15. We can see that, the DHM has followed quite well the reference trajectory of its CoM. There are some tracking errors for the feet as shown in Fig. 7.15. Since stability margins are taken into account during the generation of the CoM trajectory (see Fig. 7.6 and Fig. 7.9), the DHM is able to maintain its dynamic balance during the motion in spite of tracking errors.

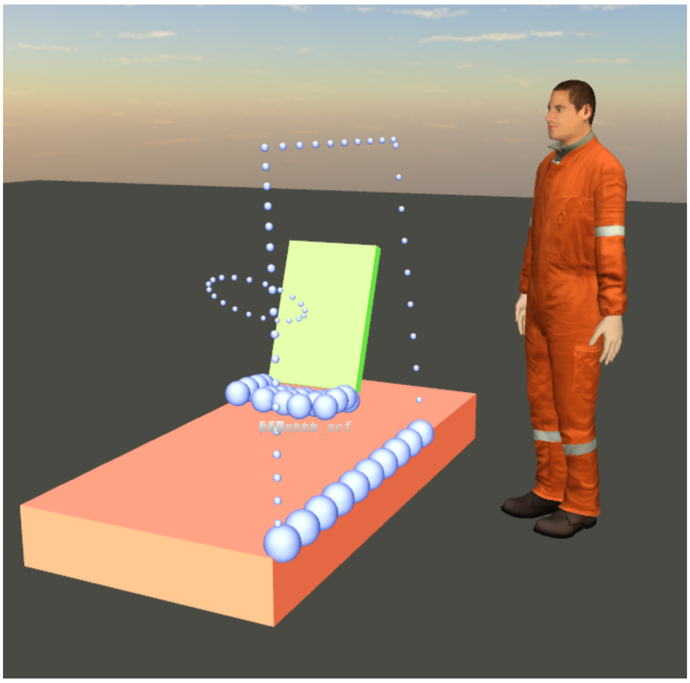


**Figure 7.12:** Structured environment and the dynamic virtual human model.

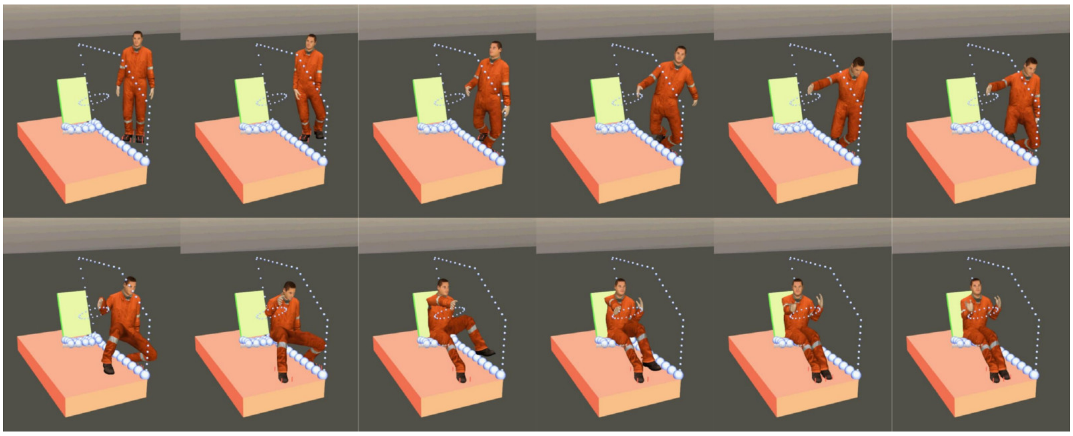


**Figure 7.13:** Clips of the dynamic motion of digital human model in simulation of a car-ingress scenario.

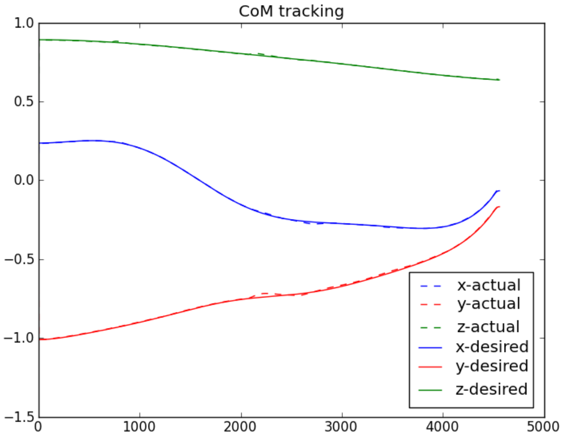


**Figure 7.14:** CoM tracking curves comparison in simulation of a car-ingress scenario.

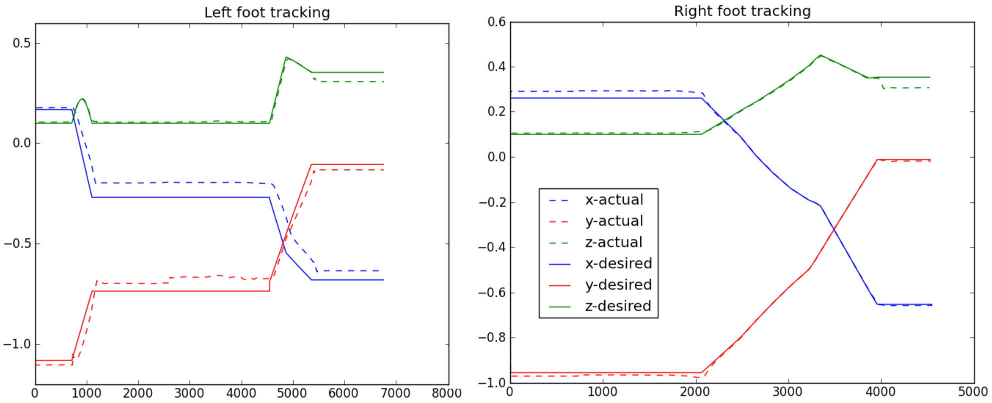


**Figure 7.15:** Feet tracking curves comparison in simulation of a car-ingress scenario.

## 7.3 Result evaluation

### 7.3.1 CoM trajectory

#### Timing

Time durations for the recorded motion and the generated motion are given in Table 7.7. A graphic comparison for motion timing is given in Fig. 7.16. We can see from the comparison results that the total duration of the generated trajectory is much larger than the duration of the recorded motion ($4.55s$ compared with $3.06s$). Durations of the single-support phases in the generated motion are equal to their minimum values that are imposed beforehand. In other phases where the admissible pseudo-wrench spaces are much larger, the generated motion has larger durations than the recorded motion. The observation result in motion timing is in agreement with the objective of maximizing the lower bound of stability margin: small admissible space in a single-support phase results in small stability margins thus the algorithm decreases the duration of single support phase in order to get larger stability margins with larger polytopes (in phases with more supports).

| Generated motion($s$) | 0.0 | 0.71 | 1.11 | 2.06 | 2.56 | 3.95 | 4.55 |
|---|---|---|---|---|---|---|---|
| Recorded motion($s$) | 0.0 | 0.72 | 1.30 | 1.35 | 1.79 | 2.42 | 3.06 |
| Left foot | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| Right foot | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| Left hand | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Right hand | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| Buttock | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**Table 7.7:** Timing and support configurations of the first 6 transition phases in the recorded motion and in the generated motion.

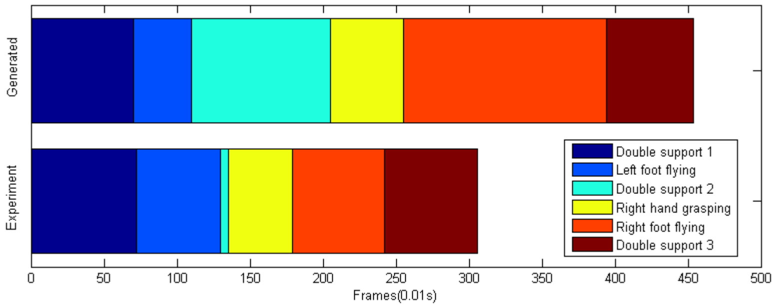


**Figure 7.16:** Timing comparison for the generated and the recorded motions.

### CoM positions

The generated CoM trajectory is compared with the one of the recorded motion in Fig. 7.17. We compare projections of the two trajectories in respectively x-y and y-z planes (see left and right figures in Fig. 7.18). As we can see from the results, the two trajectories are close to each other in x-y directions, but they show important difference in z-direction: the generated CoM trajectory is always lower than the recorded motion and it has no local peak which exists in the recorded one.

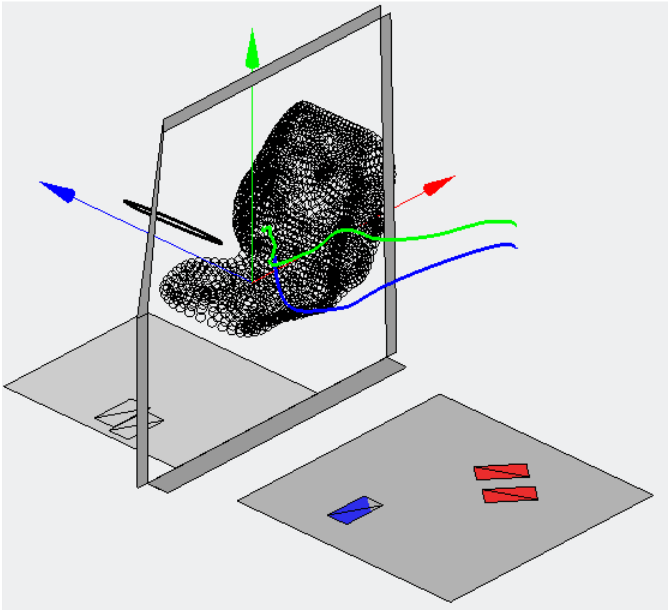


**Figure 7.17:** CoM trajectory comparison for the generated (blue) and the recorded (green) motions.

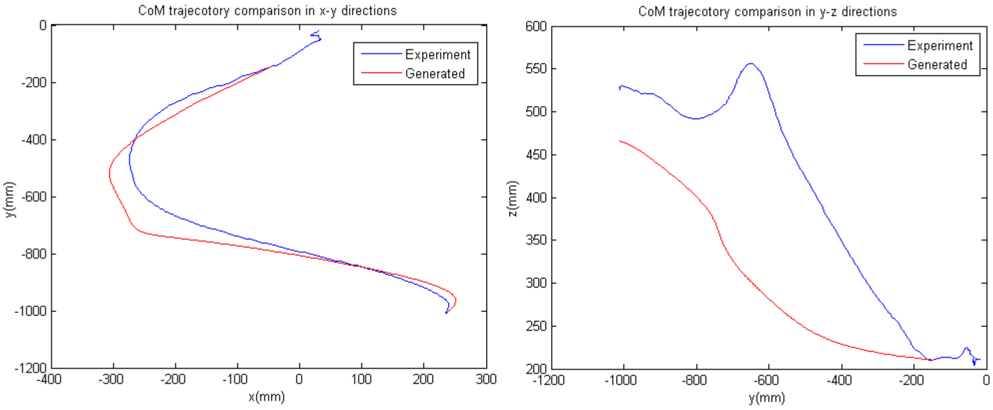


**Figure 7.18:** CoM trajectory comparison for the generated and the recorded motions in respectively x-y (left) and y-z (right) planes.

Reasons for the height difference are:

- we intentionally lower initial CoM height regarding the recorded in CoM trajectory generation step (discussed in Section 7.4.1);
- lack of consideration of local influence of the sill in CoM trajectory generation step.

### 7.3.2 Postures

Because of the different kinematic definition of DHMs in two softwares (XDE® and RPx®), we cannot directly obtain the comparison results for joint angles during the motions. However, the visual comparison of some key postures during the simulated and the recorded car-ingress motions is given in Fig. 7.19.

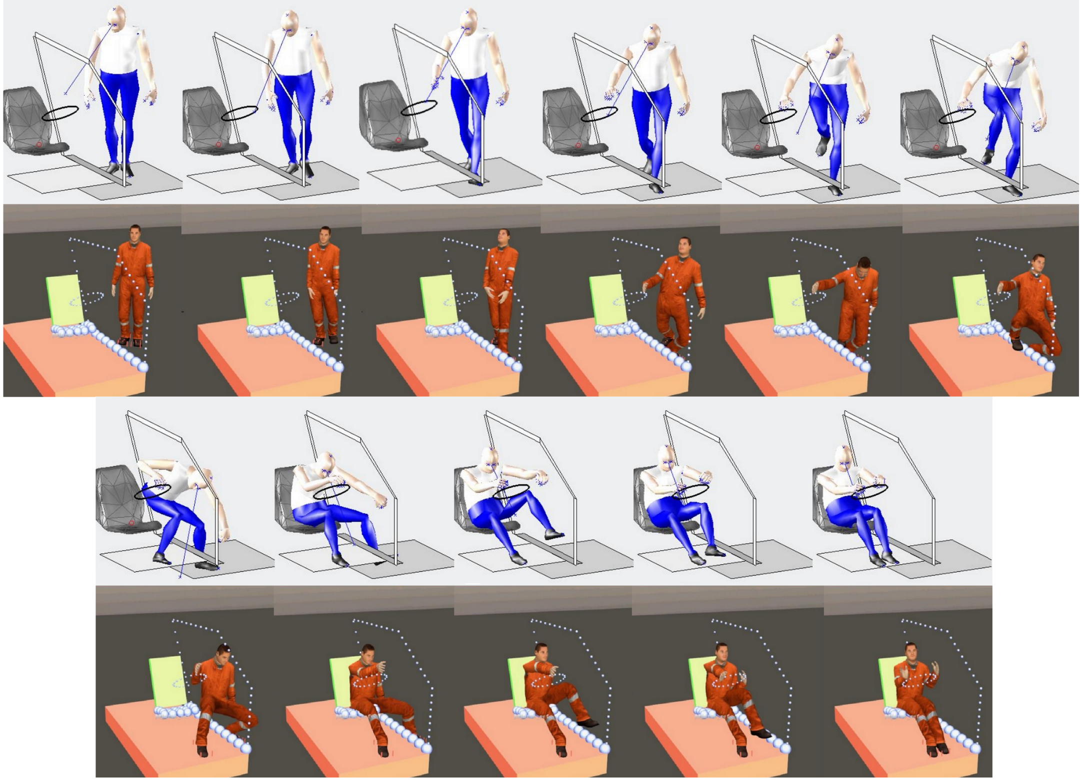


**Figure 7.19:** DHM postures comparison for the simulated and the recorded motions.

As we can see, the DHMs show similar postures in general in both motions. However, the simulated motion shows significant difference in some details from the recorded one including:

- the foot is always horizontal during a contact (there is no toe-lift/heel-strike phases);
- left knee bending is very significant during the right foot step;
- the DHM's trunk keeps vertical while entering the car.

Reasons for the posture differences lie in the following aspects:

- the foot contact is modeled as a rectangle with four contact points which touch (contact established) or leave (contact removed) a supporting surface at the same time;
- the CoM height in the generated trajectory is lower than that in the recorded motion;
- the lack of consideration of naturalness at the motion planning level;
- the adoption of a vertical posture as reference of the DHM'root body in the simulation.

## 7.4 Conclusion and discussions

In this chapter, we present application of the hierarchical framework for planning and simulating DHM motion in a car-ingress scenario.

We firstly present human motion strategies in car-ingress scenario based on some recorded human motions in previous experiments. Then we present the preliminary work that we have carried out including the construction of the structured simplified environment, sampling spheres for obstacle and extraction of the supports configurations. The framework is afterward applied for generating and simulating the car-ingress motion and we show according results at each of the three levels of the framework. Afterwards we have compared the simulated motion with the recorded one in several aspects.

### 7.4.1 Discussions

Though we use experimental motion data for generating the CoM trajectory in practical application in this chapter, however, we have chosen the initial height of CoM to be slightly smaller than its real height of the recorded motion. Reasons for this choice include:

1. There exists a difference between CoM positions of the subject and the DHM because of the difference of their body sizes;
2. In our model, all the four contact points are removed at the same time when a foot contact is removed, and there is no heel-lifting phase as in a real motion. This difference will make CoM position lower than in a real motion;
3. There are errors associated in CoM position computation based on the reconstructed motion;
4. The DHM needs to fully extend its legs to reach a high CoM height, which may bring singularity in dynamic motion control.

In this case study, the planning time seems rather long for only an one-step transition. Two reasons can be argued for this problem:

1. collision avoidance adds hundreds of non-linear inequality constraints in SQP solver in the posture generator, which increases significantly the computing complexity;

2. the feasible zone is narrow regarding the sampling space in Bi-RRT method (it needs more than 100 iterations for generating a path with less than 10 nodes), thus most of the planning time is consumed for verifying non-feasible path segments with posture generator.

### 7.4.2 Perspective

In the future, we should improve this method in the following aspects:

- More precise modeling for multiple foot contact types including full sole contact, heel contact and toe contact;
- Generation of the sequence of support configurations;
- Whole-body considerations in the generation of the CoM trajectory;
- Improve the naturalness of the final whole-body motion.

## Chapter 8

# Conclusion and perspective

## 8.1 Conclusion

This thesis focuses on the simulation of DHM motions in cluttered environments. In human-related research domains such as biomechanics or ergonomics, it is expected that in the long run DHM can replace humans to realize human-like motions so that we will be able to evaluate, estimate or predict human performances without using real human subjects. As a first step towards this objective, we have explored in this study a hierarchical method for simulating DHM motions in cluttered environment.

### Experiments and analysis (Chapter 3)

In the first stage of this study, we have conducted Motion Capture experiments. In these experiments, a series of human motions in the scenario "passage under a beam" have been recorded. Altogether 90 motions have been recorded for three human subjects, five material configurations and three motion strategies. Under software RPx®, the recorded motions are reconstructed using a kinematic DHM model tailored for each subject. Accordingly, we obtained the articular angles for all frames of each motion. The whole-body CoM positions as well as CoM of several body segments then have been estimated from the reconstructed motions.

By observing and analyzing the experimental data, we have obtained heuristic knowledge on human motions in cluttered environments including the sequence and placements of support configurations, transition timing, constraints imposed by supports or overhead obstacles on CoM, etc. We have concluded some human motion principles which have helped us to make hypotheses and determine essential problematics for our following work.

### Dynamic balance during motions (Chapter 4)

Dynamic balance with generic supports has been studied in this chapter. Firstly we propose some definitions of balance and stability of the balance state. Then after a state of the art on balance, we have formulated a balance criterion based on a simplified DHM model (a mass point with multiple generic supports with the environment) by slightly extending several existing balance criteria. Physical conditions for the system to maintain balance, can be transformed to an "admissible pseudo-wrench space" which serves as the balance criterion in this study. The balance criterion has been validated in some scenarii via comparison with some traditional criteria. The concept of "stability margin" associated with the balance criterion helps to estimate and evaluate capability of resisting perturbations and tolerating tracking errors. By extending this criterion for a whole-body DHM model, variation of angular momentum caused by posture change is taken into account. Finally the balance criterion has been used for analyzing some recorded motions and then, for planning DHM motions.

#### Hierarchical framework (Chapter 5 and Chapter 6)

A hierarchical framework has been proposed for simulating dynamic DHM motions in cluttered environments. The framework can realize a dynamic simulation of DHM motion via three steps:

1. at a global level, it generates a robust global CoM trajectory with a pre-defined sequence of support configurations.
2. then at a local level, it generates for each transition phase whole-body collision-free postures and trajectories of the controlled end-effectors while respecting multiple constraints including inverse kinematics, collision-freeness, CoM position impositions.
3. at execution level, the whole-body postures, the end-effector trajectories and the global CoM trajectory are used as control references for realizing the DHM's dynamic motion in the simulation.

Several examples have been introduced at each level of the framework in order to validate the approaches.

#### Car-ingress case study (Chapter 7)

Dynamic DHM is expected to replace real human subjects in ergonomic researches such as this study in LBMC. Thus we have tested our framework in a car-ingress scenario for evaluating its performance. The structured environment, the sequence of support configurations and the initial and final motion conditions have been generated from a reconstructed car-ingress motion. Then the framework has successfully generated and realized the car-ingress motion in a dynamic simulation. The simulated motion has been compared with the one recorded in experiments and the results have been discussed.

## 8.2 Perspective

Several main points in this study should be improved in the future research.

### 8.2.1 Supports planning

Supports planning is not dealt with in our study. In several examples, we predefined the placements of the supports either by hand or using experimental data. In fact, supports planning is an interesting and important research topic in DHM motion research, especially for motions in cluttered environment. In the future, we look forward to exploring a support planning method which can automatically generate the sequence of supports configurations by taking into account human body reachability, environmental configurations as well as the randomness of support placements (as observed in Chapter 3).

### 8.2.2 Global CoM trajectory

During the generation of the global CoM trajectory, we need to define a general geometric constraint on the CoM position. In our study, the geometric constraints come from two aspects: the supports and the overhead obstacle. The DHM cannot realize a support beyond its reachable distance, thus its CoM must be located within a reasonable distance. We have observed this fact in our experiments as introduced in Chapter 3. For avoiding obstacles, we have manually introduced in our study a geometric constraint for CoM in only simple cases. In fact, for a randomly positioned obstacle with a irregular shape, the geometric constraint on CoM position will be quite complex. We expect to deal with this problem in future researches.

### 8.2.3 Local planning

Main drawback at this level lies in the lack of consideration of naturalness. The posture generator that we use in this step for generating whole-body collision-free posture and for validating the sampling end-effector's configuration computes postures which are feasible but not necessarily natural. Moreover, the contacts in our study, particularly in foot stepping phases, is not natural because of the hypothesis that a foot contact is built (or removed) by touching (or removing) all its four points at the same time. Thus in the future, foot motions such like "heel strike" and "toe lift" of DHM should be considered.

### 8.2.4 Variations of subjects

One main advantage of a DHM lies in its ease of modification for representing different human body characteristics. In the future, we need to explore a module to modify the DHM's characteristics (e.g. dimensions, shapes, gender, mass distribution) and capabilities (e.g. joint limits). Accordingly, we need to explore another module for sampling the bounding spheres of DHM body segments.

# List of Figures

## Appendix A

# Polytope computation

---

In this part, we present the method for computing the admissible pseudo-wrench space from the admissible exterior wrench space. Let's recall the expression of admissible exterior wrench space – polytope $\Psi$:

$$\Psi = \left\{ \boldsymbol{\psi} \in \mathbb{R}^{6(n+k)} \, | \mathbf{A}_{\psi} \boldsymbol{\psi} \leq \mathbf{b}_{\psi} \right\}$$

and the expression of admissible pseudo-wrench space – polytope $\mathcal{W}_p$:

$$\mathcal{W}_p = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \mathbf{A} \end{bmatrix} \Psi := \left\{ \mathbf{w_p} \in \mathbb{R}^6 \, | \; \mathbf{H w_p} \leq \mathbf{h} \right\}$$

Polytope $\Psi$ is a known constant for a given support configuration. The polytope $\mathcal{W}_p$ can be computed from $\Psi$ in the following steps:

1. Calculate the vertices $\mathbf{V}$ of the polytope $\Psi$ from its H-representation;
2. Calculate the point set $\mathbf{V}_p$ containing the projections of $\mathbf{V}$ in the 6D pseudo-wrench space;
3. Generate the polytope $\mathcal{W}_p$ who is the convex hull enveloping all the points in $\mathbf{V}_p$;
4. Calculate and return the H-representation of $\mathcal{W}_p$ (i.e. $\mathbf{H}$ and $\mathbf{h}$ ).

The Algorithm 7 shows how the function *generatePolytope*() calculates the matrix $\mathbf{H}1$, $\mathbf{H}2$ and $\mathbf{h}$ for each interaction configuration. This algorithm is coded in Matlab and it uses the MPT toolbox [KGB04] (written by Automation Control Laboratory, ETH, Zurich) to carry out the manipulations of polytopes. In Algorithm 7, "*polytope*", "*extreme*", "*hull*" and "*double*" are functions pre-defined in MPT toolbox.

**Algorithm 7** generatePolytope($\mathbf{A}$, $\mathbf{C}$, $\mathbf{A}_\psi$, $\mathbf{b}_\psi$)

```
$\Psi = polytope(\mathbf{A}_\psi, \mathbf{b}_\psi)$
$\mathbf{V} = extreme(\Psi)$
$\mathbf{V}_p = \mathbf{V} * \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \mathbf{A} \end{bmatrix}^t$
$\mathcal{W}_p = hull(\mathbf{V}_p)$
$[\mathbf{H}, \mathbf{h}] = double(\mathcal{W}_p)$
$\mathbf{H}1 = \mathbf{H}(:, 1:3)$, $\mathbf{H}2 = \mathbf{H}(:, 4:6)$
Return $\mathbf{H}1$, $\mathbf{H}2$ and $\mathbf{h}$
```

## Appendix B

# Body segments of the DHM

| ID | Name |
|---|---|
| 0 | Phantom (*root*) |
| 1 | Lumbar Spine |
| 2 | Thoracic Spine |
| 3 | Neck |
| 4 | Head Center of Gravity |
| 5 | Right Arm |
| 6 | Right ForeArm |
| 7 | Right Center of Prehension |
| 8 | Left Arm |
| 9 | Left ForeArm |
| 10 | Left Center of Prehension |
| 11 | Right Thigh |
| 12 | Right Leg |
| 13 | Right Ankle |
| 14 | Right Toes |
| 15 | Left Thigh |
| 16 | Left Leg |
| 17 | Left Ankle |
| 18 | Left Toes |

**Table B.1:** Body segments of the DHM.

Appendix C

# Markers in MoCap experiments

---

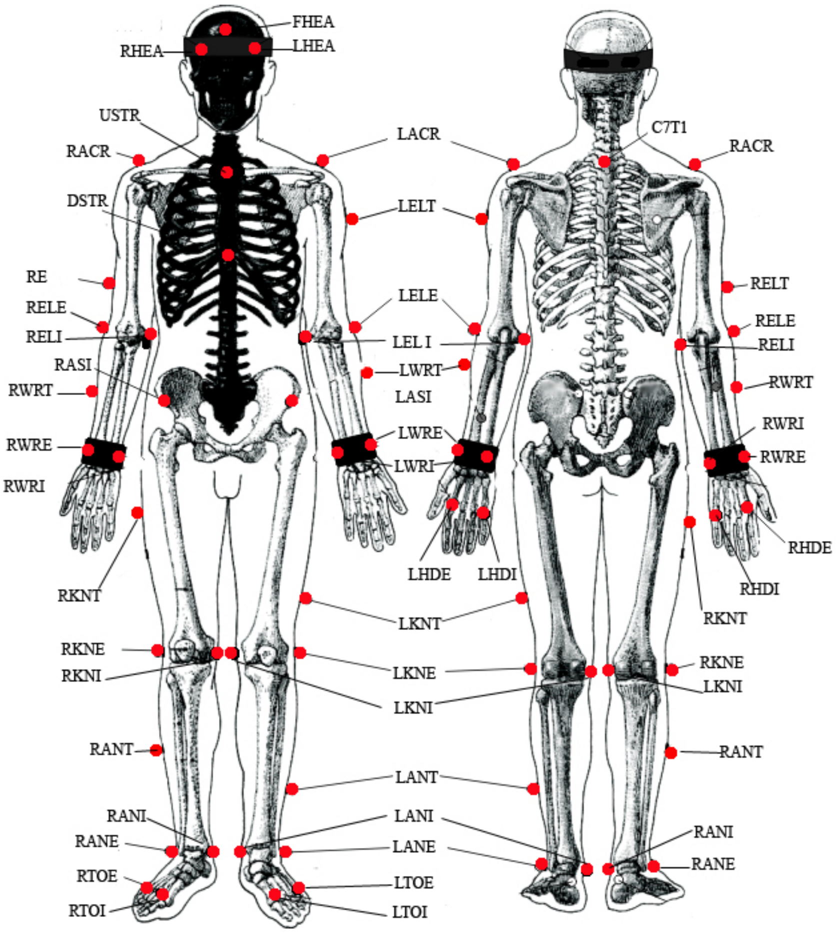


**Figure C.1:** Names and positions of the 50 markers on human body in our MoCap experiments.

## Appendix D

# Ramsis kinematic model

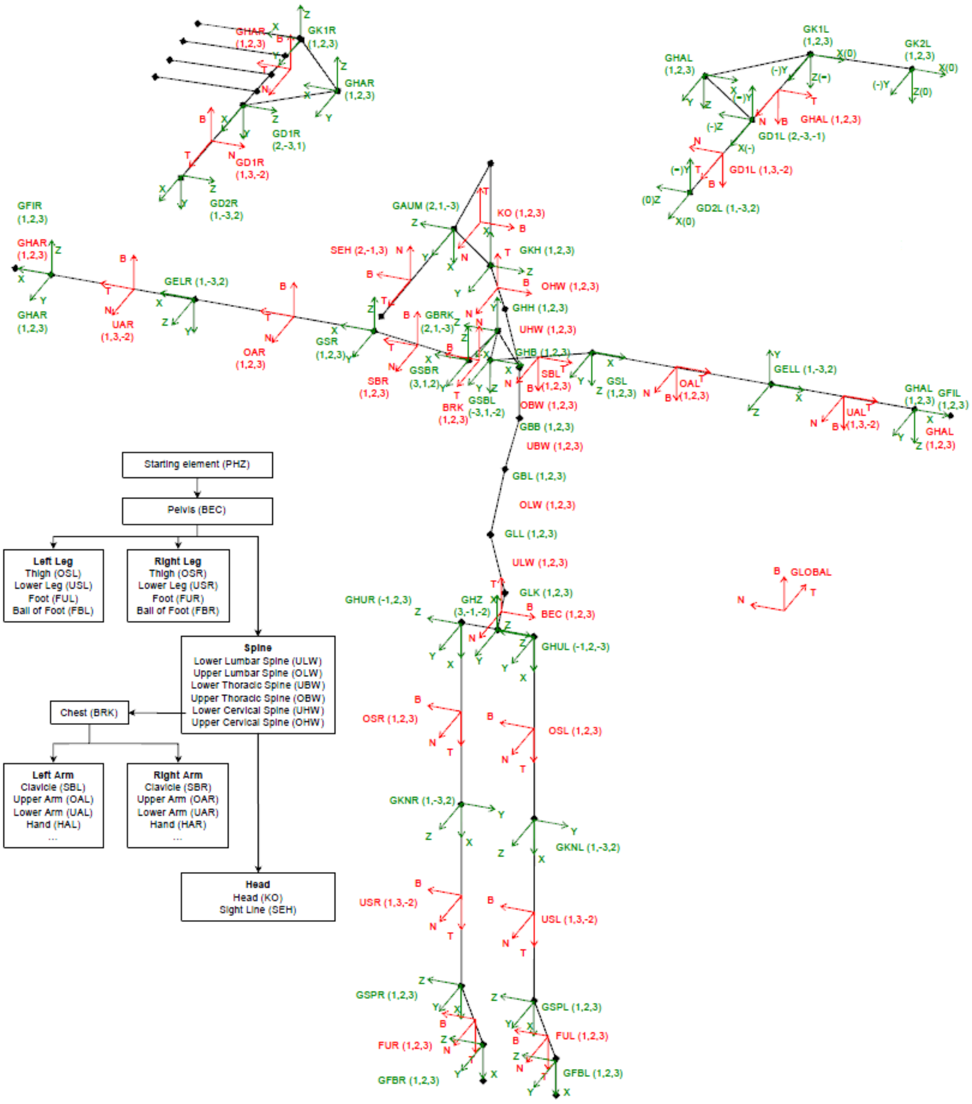


**Figure D.1:** Joint definition of Ramsis® model used in our motion reconstruction using software RPx.

# Appendix E

# Software structure

---

The hierarchical framework runs with a set of software modules including polytope calculation, CoM trajectory generation, auto-differentiation, bi-RRT planning, posture-generator, dynamic motion simulator and other programs for figure plotting and file reading/saving. The structure of the software package is shown in Fig. E.1. Among the above mentioned modules, the dynamic motion simulator XDE-Dsimi® runs with a Python interface associated with C++ libraries; the posture-generator is pre-built C++ libraries (DLLs) which are called and used in Matlab. All the other program modules are coded and executed in Matlab. The user carries out the motion planning and generating work in Matlab, thereafter, the dynamic simulations will be carried out in XDE-Dsimi®.

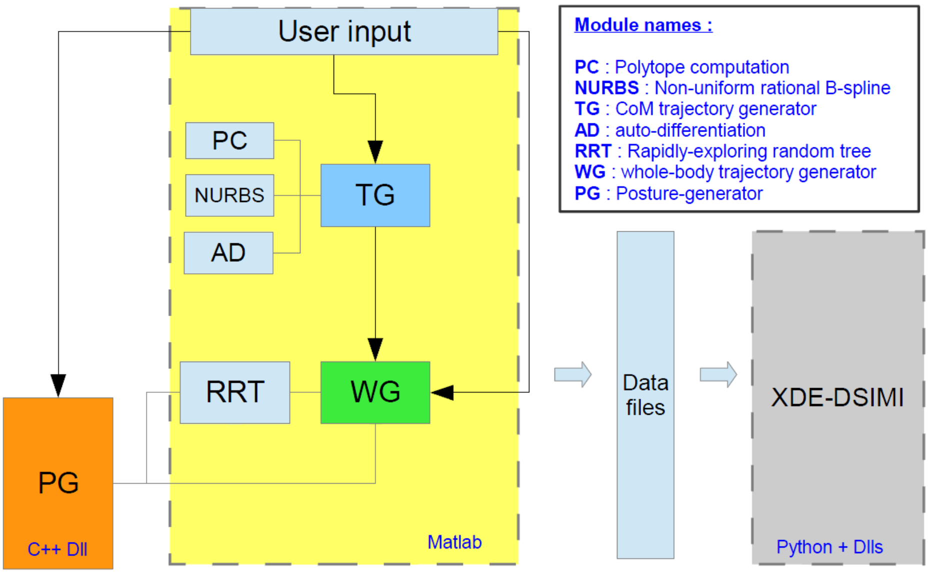


**Figure E.1:** The structure of the motion planning framework with related software modules.

## APPENDIX F

# Gradient of sphere-sphere distance function

---

Suppose that $\mathbf{p}_i$ and $\mathbf{p}_j$ are centers of two different bounding spheres which are functions of the configuration of the DHM, then the distance between the two spheres is:

$$d_{ij}(\mathbf{q}) = \|\mathbf{p}_i(\mathbf{q}) - \mathbf{p}_j(\mathbf{q})\| = \sqrt{([\mathbf{p}_i(\mathbf{q}) - \mathbf{p}_j(\mathbf{q})]^t[\mathbf{p}_i(\mathbf{q}) - \mathbf{p}_j(\mathbf{q})]} \tag{F.1}$$

The gradient of the distance function in Equation (F.1) can be explicitly obtained as shown in the following equations.

$$\frac{\partial d_{ij}(\mathbf{q})}{\partial \mathbf{q}} = \frac{\partial\sqrt{([\mathbf{p}_i(\mathbf{q}) - \mathbf{p}_j(\mathbf{q})]^t[\mathbf{p}_i(\mathbf{q}) - \mathbf{p}_j(\mathbf{q})]}}{\partial \mathbf{q}} \tag{F.2}$$

Let's denote:

$$\mathbf{D}_{ij}(\mathbf{q}) = \mathbf{p}_i(\mathbf{q}) - \mathbf{p}_j(\mathbf{q}) \tag{F.3}$$

Thus:

$$\begin{aligned}
\frac{\partial d_{ij}(\mathbf{q})}{\partial \mathbf{q}} = \frac{\partial\sqrt{\mathbf{D}^t_{ij}(\mathbf{q})\mathbf{D}_{ij}(\mathbf{q})}}{\partial \mathbf{q}} &= \frac{1}{2\sqrt{\mathbf{D}^t_{ij}(\mathbf{q})\mathbf{D}_{ij}(\mathbf{q})}}\frac{\partial(\mathbf{D}^t_{ij}(\mathbf{q})\mathbf{D}_{ij}(\mathbf{q}))}{\partial \mathbf{q}} \\
&= \frac{1}{2\sqrt{\mathbf{D}^t_{ij}(\mathbf{q})\mathbf{D}_{ij}(\mathbf{q})}}\frac{\partial \mathbf{D}_{ij}(\mathbf{q})}{\partial \mathbf{q}}\frac{\partial(\mathbf{D}^t_{ij}(\mathbf{q})\mathbf{D}_{ij}(\mathbf{q}))}{\partial \mathbf{D}_{ij}(\mathbf{q})} \\
&= \frac{1}{2\sqrt{\mathbf{D}^t_{ij}(\mathbf{q})\mathbf{D}_{ij}(\mathbf{q})}}\frac{\partial \mathbf{D}_{ij}(\mathbf{q})}{\partial \mathbf{q}}2\mathbf{D}_{ij}(\mathbf{q}) \\
&= \frac{\partial \mathbf{D}_{ij}(\mathbf{q})}{\partial \mathbf{q}}\frac{\mathbf{D}_{ij}(\mathbf{q})}{\|\mathbf{D}_{ij}(\mathbf{q})\|}
\end{aligned}$$

Finally we have:

$$\frac{\partial d_{ij}(\mathbf{q})}{\partial \mathbf{q}} = (\frac{\partial \mathbf{p}_i(\mathbf{q})}{\partial \mathbf{q}} - \frac{\partial \mathbf{p}_j(\mathbf{q})}{\partial \mathbf{q}})\frac{\mathbf{D}_{ij}(\mathbf{q})}{d_{ij}(\mathbf{q})} \tag{F.4}$$

Let $\mathbf{p}_j$ be center of an obstacle's bounding sphere, then it is a constant vector that does not depend on the configuration of the DHM. We thus get from Equation (F.4) the gradient function for the distance of body-obstacle sphere pair:

$$\frac{\partial d_{ij}}{\partial \mathbf{q}} = \frac{\partial \mathbf{p}_i(\mathbf{q})}{\partial \mathbf{q}} \frac{\mathbf{D}_{ij}(\mathbf{q})}{d_{ij}} \tag{F.5}$$

with:

$$\mathbf{D}_{ij}(\mathbf{q}) = \mathbf{p}_i(\mathbf{q}) - \mathbf{p}_j \tag{F.6}$$

# Bibliography

[AA05] P.N. Azariadis and N.A. Aspragathos. Obstacle representation by bump-surfaces for optimal motion-planning. *Robotics and Autonomous Systems*, 51(2):129–150, 2005.

[ABCO$^+$01] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C.T. Silva. Point set surfaces. In *Proceedings of the Conference on Visualization'01*, pages 21–28. IEEE Computer Society, 2001.

[AEMPG$^+$08] M.O. Ait El Menceur, P. Pudlo, P. Gorce, A. Thévenon, and F.X. Lepoutre. Alternative movement identification in the automobile ingress and egress for young and elderly population with or without prostheses. *International Journal of Industrial Ergonomics*, 38(11):1078–1087, 2008.

[AS88] R.J. Anderson and M.W. Spong. Hybrid impedance control of robotic manipulators. *Robotics and Automation, IEEE Journal of*, 4(5):549–556, 1988.

[BB08] S. Barthélemy and P. Bidaud. Stability measure of postural dynamic equilibrium based on residual radius. *Advances in Robot Kinematics: Analysis and Design*, pages 399–407, 2008.

[BELK09] K. Bouyarmane, A. Escande, F. Lamiraux, and A. Kheddar. Potential field guide for humanoid multicontacts acyclic motion planning. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 1165–1170. IEEE, 2009.

[BL08] T. Bretl and S. Lall. Testing static equilibrium for legged robots. *Robotics, IEEE Transactions on*, 24(4):794–807, 2008.

[Bob88] J.E. Bobrow. Optimal robot plant planning using the minimum-time criterion. *Robotics and Automation, IEEE Journal of*, 4(4):443–450, 1988.

[Boe80] W. Boehm. Inserting new knots into b-spline curves. *Computer-Aided Design*, 12(4):199–201, 1980.

[Bre06] T. Bretl. Motion planning of multi-limbed robots subject to equilibrium constraints: The free-climbing robot problem. *The International Journal of Robotics Research*, 25(4):317–342, 2006.

[Can88] J. Canny. *The complexity of robot motion planning.* The MIT Press, 1988.

[Cau11] Julien Causse. *Analyse cinématique et dynamique du mouvement d'accessibilité à une automobile.* PhD thesis, L'UNIVERSITE CLAUDE BERNARD LYON 1, 2011.

[Cha09] Elodie Chateauroux. *Analyse du mouvement d'accessibilité au poste de conduite d'une automobile en vue de la simulation.* PhD thesis, Ecole doctorale MEGA de Lyon, 2009.

[CLC$^+$05] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins, and T. Kanade. Footstep planning for the honda asimo humanoid. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 629–634. IEEE, 2005.

[Col09] Cyrille Collette. *Commande dynamique d'humains virtuels : équilibre robuste et gestion de tâches.* PhD thesis, l'Université Paris VI, 2009.

[CW10] E. Chateauroux and X. Wang. Car egress analysis of younger and older drivers for motion simulation. *Applied Ergonomics*, 42(1):169–177, 2010.

[CWD11] J. CAUSSE, X. WANG, and L. DENNINGER. Effects of roof height on car ingress/egress movement. 2011.

[DHE] Dhergo project (g.a. 218525), delilivrable 45: Data for case studies 3: - ingress/egress motion.

[EK09] K. Erbatur and O. Kurt. Natural zmp trajectories for biped robot reference generation. *Industrial Electronics, IEEE Transactions on*, 56(3):835–845, 2009.

[EKM06] A. Escande, A. Kheddar, and S. Miossec. Planning support contact-points for humanoid robots and experiments on hrp-2. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2974–2979. IEEE, 2006.

[EKMG09] A. Escande, A. Kheddar, S. Miossec, and S. Garsault. Planning support contact-points for acyclic motions and experiments on hrp-2. In *Experimental Robotics*, pages 293–302. Springer, 2009.

[EM96] P.H.C. Eilers and B.D. Marx. Flexible smoothing with b-splines and penalties. *Statistical Science*, pages 89–102, 1996.

[EMK07] A. Escande, S. Miossec, and A. Kheddar. Continuous gradient proximity distance for humanoids free-collision optimized-postures. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 188 –195, 29 2007-dec. 1 2007.

[Fuk03] K. Fukuda. Cddlib reference manual. *Report version 093a, McGill University, Montréal, Quebec, Canada*, 2003.

[GCLS95] K. Glass, R. Colbaugh, D. Lim, and H. Seraji. Real-time collision avoidance for redundant manipulators. *Robotics and Automation, IEEE Transactions on*, 11(3):448–457, 1995.

[GK04] A. Goswami and V. Kallem. Rate of change of angular momentum and balance maintenance of biped robots. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 4, pages 3785–3790. IEEE, 2004.

[Gos99] A. Goswami. Foot rotation indicator (fri) point: A new gait planning tool to evaluate postural stability of biped robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 1, pages 47–52. IEEE, 1999.

[Grü03] B. Grünbaum. *Convex polytopes*, volume 221. Springer Verlag, 2003.

[GS67] B. Grunbaum and GC Shephard. Convex polytopes. 1967.

[HBL05] K. Hauser, T. Bretl, and J.C. Latombe. Non-gaited humanoid locomotion planning. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 7–12. IEEE, 2005.

[HHH$^+$06] H. Hirukawa, S. Hattori, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, and M. Morisawa. A universal stability criterion of the foot contact of legged robots-adios zmp. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1976–1983. IEEE, 2006.

[HHH$^+$07] K. Harada, S. Hattori, H. Hirukawa, M. Morisawa, S. Kajita, and E. Yoshida. Motion planning for walking pattern generation of humanoid. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 4227–4233. IEEE, 2007.

[HHK$^+$07] H. Hirukawa, S. Hattori, S. Kajita, K. Harada, K. Kaneko, F. Kanehiro, M. Morisawa, and S. Nakaoka. A pattern generator of humanoid robots walking on a rough terrain. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2181–2187. IEEE, 2007.

[HKKH03] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa. Zmp analysis for arm/leg coordination. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 75–81. IEEE, 2003.

[HKLR02] D. Hsu, R. Kindel, J.C. Latombe, and S. Rock. Randomized kinodynamic motion planning with moving obstacles. *The International Journal of Robotics Research*, 21(3):233–255, 2002.

[Hog84] N. Hogan. Impedance control: An approach to manipulation. In *American Control Conference, 1984*, pages 304–313. IEEE, 1984.

[HYK$^+$01] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, and K. Tanie. Planning walking patterns for a biped robot. *Robotics and Automation, IEEE Transactions on*, 17(3):280–289, 2001.

[JKMoCED04] C.N. Jones, E.C. Kerrigan, J.M. Maciejowski, and University of Cambridge. Engineering Dept. *Equality set projection: A new algorithm for the projection of polytopes in halfspace representation.* Citeseer, 2004.

[K+99] D.E. Koditschek et al. Templates and anchors: neuromechanical hypotheses of legged locomotion on land. *Journal of Experimental Biology*, 202(23):3325–3332, 1999.

[KGB04] M. Kvasnica, P. Grieder, and M. Baotić. Multi-Parametric Toolbox (MPT), 2004.

[Kha80] O. Khatib. Commande dynamique dans l'espace opérationnel des robots manipulateurs en présence d'obstacles. *Ph. D. thesis, l'Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, France.*, 1980.

[KJNK+01] J.J. Kuffner Jr, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue. Footstep planning among obstacles for biped robots. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 1, pages 500–505. IEEE, 2001.

[KKK+03] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 1620–1626. Ieee, 2003.

[KKN+02] J.J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue. Dynamically-stable motion planning for humanoid robots. *Autonomous Robots*, 12(1):105–118, 2002.

[KKN+03] J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue. Online footstep planning for humanoid robots. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages 932–937. IEEE, 2003.

[KKS95] M.J. Kim, M.S. Kim, and S.Y. Shin. A general construction scheme for unit quaternion curves with simple high order derivatives. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 369–376. ACM, 1995.

[KKT+00] S. Kagami, F. Kanehiro, Y. Tamiya, M. Inaba, and H. Inoue. Autobalancer: An online dynamic balance compensation scheme for humanoid robots. In *Proc. Int. Workshop Alg. Found. Robot.(WAFR)*, 2000.

[KNK+01] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue. Motion planning for humanoid robots under obstacle and dynamic balance constraints. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 1, pages 692–698. IEEE, 2001.

[Lat91] Jean-Claude Latombe. *Robot motion planning.* Springer, 1991.

[LaV98] S.M. LaValle. Rapidly-exploring random trees: A new tool for path planning. *In*, (98-11), 1998.

[LaV06] S.M. LaValle. *Planning algorithms.* Cambridge Univ Pr, 2006.

[LFR09] S. Lengagne, P. Fraisse, and N. Ramdani. Planning and fast re-planning of safe motions for humanoid robots: Application to a kicking motion. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 441–446. IEEE, 2009.

[LKJ00] S.M. LaValle and J.J. Kuffner Jr. Rapidly-exploring random trees: Progress and prospects. 2000.

[LP83] T. Lozano-Perez. Spatial planning: A configuration space approach. *Computers, IEEE Transactions on*, 100(2):108–120, 1983.

[MdLH10] I. Mordatch, M. de Lasa, and A. Hertzmann. Robust physics-based locomotion using low-dimensional planning. *ACM Transactions on Graphics (TOG)*, 29(4):71, 2010.

[MWT08] G. Monnier, X. Wang, and J. Trasbot. Rpx: A motion simulation tool for car interior design. Editor: Vincent G. Duffy, Publisher: Taylor & Francis Group, 2008.

[OL96] H. Ozaki and C. Lin. Optimal b-spline joint trajectory generation for collision-free movements of a manipulator under dynamic constraints. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 4, pages 3592–3597. IEEE, 1996.

[OOH$^+$05] K. Okada, T. Ogura, A. Haneda, J. Fujimoto, F. Gravot, and M. Inaba. Humanoid motion generation system on hrp2-jsk for daily life environment. In *Mechatronics and Automation, 2005 IEEE International Conference*, volume 4, pages 1772–1777. IEEE, 2005.

[OS94] M.H. Overmars and P. Svestka. A probablisitic learning approach to motion planning. *UU-CS*, (1994-03), 1994.

[PGH05] M.B. Popovic, A. Goswami, and H. Herr. Ground reference points in legged locomotion: Definitions, biological trajectories and control implications. *The International Journal of Robotics Research*, 24(12):1013, 2005.

[PHH04a] M. Popovic, A. Hofmann, and H. Herr. Angular momentum regulation during human walking: biomechanics and control. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 3, pages 2405–2411. Ieee, 2004.

[PHH04b] M. Popovic, A. Hofmann, and H. Herr. Zero spin angular momentum control: definition and applicability. In *Humanoid Robots, 2004 4th IEEE/RAS International Conference on*, volume 1, pages 478–493. IEEE, 2004.

[PR98] J.H. Park and Y.K. Rhee. Zmp trajectory generation for reduced trunk motions of biped robots. In *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, volume 1, pages 90–95. IEEE, 1998.

[QEMR11] Z. Qiu, A. Escande, A. Micaelli, and T. Robert. Human motions analysis and simulation based on a general criterion of stability. In *First International Symposium on Digital Human Modeling.* IEA, 2011.

[QEMR12] Z. Qiu, A. Escande, A. Micaelli, and T. Robert. A hierarchical framework for realizing dynamically-stable motions of humanoid robot in obstaclecluttered environments. In *Humanoid Robots, 2012 12th IEEE-RAS International Conference on.* IEEE, 2012.

[RQC$^+$11] T. Robert, Z. Qiu, J. Causse, A. Escande, and A. Micaelli. A dynamic stability analysis of the sit-to-stand transfer. In *ISB 2011.* ISB, 2011.

[SAK07] L. Sentis and O. Adviser-Khatib. *Synthesis and control of whole-body behaviors in humanoid systems.* Stanford University, 2007.

[SB04] P. Sardain and G. Bessonnet. Forces acting on a biped robot. center of pressure-zero moment point. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 34(5):630–637, 2004.

[SK05] L. Sentis and O. Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, 2(4):505–518, 2005.

[SN05] T. Sugihara and Y. Nakamura. A fast online gait planning with boundary condition relaxation for humanoid robots. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 305–310. IEEE, 2005.

[VJ69] M. Vukobratovic and D. Juricic. Contribution to the synthesis of biped gait. *Biomedical Engineering, IEEE Transactions on*, (1):1–6, 1969.

[WCM$^+$05] X. Wang, N. Chevalot, G. Monnier, S. Ausejo, A. Suescun, and J. CELIGÜETA. Validation of a model-based motion reconstruction method developed in the realman project. *SAE transactions*, 114(7):873–879, 2005.

[Wie02] P.B. Wieber. On the stability of walking systems. 2002.

[XAA07] EK Xidias, PN Azariadis, and NA Aspragathos. Two-dimensional motion-planning for nonholonomic robots using the bump-surfaces concept. *Computing*, 79(2):109–118, 2007.

[XAA08] E.K. Xidias, P.N. Azariadis, and N.A. Aspragathos. Path planning of holonomic and non-holonomic robots using bump-surfaces. *Journal of Computer Aided Design and Applications. v5 i1-4*, pages 497–507, 2008.

[XZA10] EK Xidias, P.T. Zacharia, and NA Aspragathos. Time-optimal task scheduling for articulated manipulators in environments cluttered with obstacles. *Robotica*, 28(3):427–440, 2010.

[YAOH05] C. Yin, A. Albers, J. Ottnad, and P. Haussler. Stability maintenance of a humanoid robot under disturbance with fictitious zero-moment point (fzmp). In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3149–3156. IEEE, 2005.

[YBEL05] E. Yoshida, I. Belousov, C. Esteves, and J.P. Laumond. Humanoid motion planning for dynamic tasks. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 1–6. IEEE, 2005.

[YES$^+$06] E. Yoshida, C. Esteves, T. Sakaguchi, J.P. Laumond, and K. Yokoi. Smooth collision avoidance: Practical issues in dynamic humanoid motion. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 827–832. IEEE, 2006.

[Yos05] E. Yoshida. Humanoid motion planning using multi-level dof exploitation based on randomized method. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3378–3383. IEEE, 2005.

[ZPM09] L. Zhang, J. Pan, and D. Manocha. Motion planning of human-like robots using constrained coordination. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 188–195. IEEE, 2009.