

Université Fédérale



Toulouse Midi-Pyrénées

# THÈSE

En vue de l'obtention du

**DOCTORAT DE L'UNIVERSITÉ FÉDÉRALE  
TOULOUSE MIDI-PYRÉNÉES**

Délivré par :

*l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

---

Présentée et soutenue le 09/2016 par :

**MAXIMILIEN NAVEAU**

---

**ADVANCED HUMAN INSPIRED WALKING STRATEGIES FOR  
HUMANOID ROBOTS**

---

## JURY

MME CHRISTINE CHEVALLEREAU	Directeur de recherche	Rapporteur
M. CHRISTIAN OTT	Chef de département, DLR	Rapporteur
M. PIERRE-BRICE WIEBER	Chargé de Recherche	Membre du Jury
M. FLORENT LAMIRAUX	Directeur de recherche	Membre du Jury
M. OLIVIER STASSE	Directeur de recherche	Directeur de thèse

---

**École doctorale et spécialité :**

*EDSYS : Robotique 4200046*

**Unité de Recherche :**

*Laboratoire d'analyse et d'architecture des systèmes*

**Directeur de Thèse :**

*M. Olivier STASSE*

**Rapporteurs :**

*Mme Christine CHEVALLEREAU et M. Christian OTT*



## Acknowledgment

First of all, I would like to thank C. Chevallereau and C. Ott for reviewing this thesis. As well as the jury members P. Souères and P.B. Wieber. I would also like to thank O. Stasse who guided me during the past three years. He was of great support scientifically and during the rough times.

I gratefully acknowledge the European Commission for founding the FP7 Project KoroBot 611909 and my thesis. It allowed me to have very interesting collaborations with the different partners of the project. I would like to personally thank M. Kudruss and the team from Heidelberg who hosted me in several occasions for interesting workshops and conferences.

I would like to thank all the researchers with whom I worked and specifically I. Ramirez, M. Karklinsky and A. Mukovskiy.

I also gratefully acknowledge S. Boria and B. Duprieux from Airbus/Future of the Aircraft Factory for their help and support.

I want to acknowledge the Gepetto team for their hospitality during these three years. I personally thank C. Benazeth, the LAAS-CNRS engineer maintaining the HRP-2 robot in good shape. I thank him for his good work and patience.

Finally my thanks go to all my family and friends for their reviews and support during the writing of my thesis.



# Contents

<b>Introduction</b>	<b>1</b>
<b>1 NMPC walking pattern generator</b>	<b>21</b>
1.1 Introduction . . . . .	23
1.1.1 Motivation . . . . .	23
1.1.2 Related work . . . . .	24
1.1.3 Contribution of the chapter . . . . .	25
1.2 Derivation of the dynamics . . . . .	26
1.2.1 Discretization of CoM dynamics . . . . .	26
1.2.2 Linear inverted pendulum dynamics . . . . .	27
1.2.3 Automatic foot step placement . . . . .	27
1.3 Nonlinear Model Predictive Control . . . . .	28
1.3.1 The controller . . . . .	28
1.3.2 The cost function . . . . .	29
1.3.3 The constraints . . . . .	30
1.3.4 Additional constraint : local obstacle avoidance . . . . .	33
1.3.5 The solver . . . . .	33
1.3.6 The line search . . . . .	34
1.4 Dynamic Filter . . . . .	35
1.5 Experiments with HRP-2 . . . . .	36
1.5.1 Experimental setup . . . . .	36
1.5.2 Robustness to perturbation . . . . .	37
1.5.3 Computation time . . . . .	38
1.5.4 Cost function gains . . . . .	38
1.5.5 qpOASES solver . . . . .	38
1.6 Conclusion . . . . .	39
<b>2 Multicontact Locomotion</b>	<b>41</b>
2.1 Previous work . . . . .	43
2.2 Generation of the center-of-mass trajectory . . . . .	45
2.2.1 Dynamic model and constraints . . . . .	45
2.2.2 Objective function . . . . .	48
2.2.3 Optimal control formulation . . . . .	49
2.3 Motion Generation . . . . .	50
2.3.1 Definition of the contact sequence . . . . .	50
2.3.2 End-effector trajectories . . . . .	50
2.3.3 Whole-body generation . . . . .	50
2.4 Results and simulation . . . . .	51
2.5 Contribution to a parallel work on this problem . . . . .	54
2.6 Conclusion . . . . .	58

<b>3</b>	<b>Pulling a hose</b>	<b>59</b>
3.1	Motivation . . . . .	61
3.2	Picking motion . . . . .	62
3.3	The walking pattern generator . . . . .	64
3.4	Pulling strategy . . . . .	64
3.4.1	Hybrid controller . . . . .	64
3.4.2	Simulation analysis . . . . .	65
3.5	Walking task . . . . .	66
3.6	Experimental results . . . . .	67
3.7	Conclusion and future work . . . . .	70
<b>4</b>	<b>Robust human-inspired power law trajectories for humanoid HRP-2 robot</b>	<b>71</b>
4.1	Introduction . . . . .	73
4.1.1	Power laws governing human motion . . . . .	73
4.1.2	Guiding trajectories for humanoid robots . . . . .	74
4.1.3	Robust trajectories with contracting dynamics . . . . .	75
4.2	Reactive walking pattern generator . . . . .	75
4.3	Regularization of contracting oscillators . . . . .	75
4.3.1	Morphed Andronov-Hopf oscillators . . . . .	76
4.3.2	Temporal regularization of a dynamical system . . . . .	76
4.4	Results . . . . .	78
4.4.1	Integration inside the Stack-of-Tasks framework . . . . .	79
4.4.2	Dynamic simulation results . . . . .	80
4.4.3	Experimental results . . . . .	82
4.5	Conclusions . . . . .	83
<b>5</b>	<b>Learning Movement Primitives for the Humanoid Robot HRP2</b>	<b>85</b>
5.1	Motivation . . . . .	87
5.2	Related work . . . . .	87
5.2.1	Modeling of whole-body movements in computer graphics . . . . .	88
5.2.2	Biological motor control of multi-step sequences . . . . .	88
5.2.3	Related approaches in humanoid robotics . . . . .	89
5.3	System architecture . . . . .	90
5.3.1	Human data . . . . .	90
5.3.2	Robotics Implementation . . . . .	93
5.3.3	Overall architecture . . . . .	93
5.4	Results . . . . .	94
5.4.1	Experimental setup . . . . .	94
5.4.2	Kinematic re-targeting . . . . .	95
5.4.3	Experimental results . . . . .	96
5.5	Conclusions . . . . .	99

---

<b>6</b>	<b>HRP-2 as Universal Worker Proof of Concept</b>	<b>101</b>
6.1	Fast re-planning for moving obstacle avoidance . . . . .	103
6.2	Reactive walking pattern generator . . . . .	104
6.3	Whole body motion for screwing . . . . .	105
6.4	3D walking . . . . .	106
6.5	Conclusion . . . . .	108
	<b>Conclusion and Perspectives</b>	<b>109</b>
<b>A</b>	<b>Annexe</b>	<b>115</b>
A.1	The dynamic filter . . . . .	117
A.1.1	Dynamic filtering for walking . . . . .	117
A.1.2	Technical details . . . . .	118
A.1.3	Conclusion . . . . .	119
	<b>Bibliography</b>	<b>121</b>





# Introduction



## Context and KoroiBot project

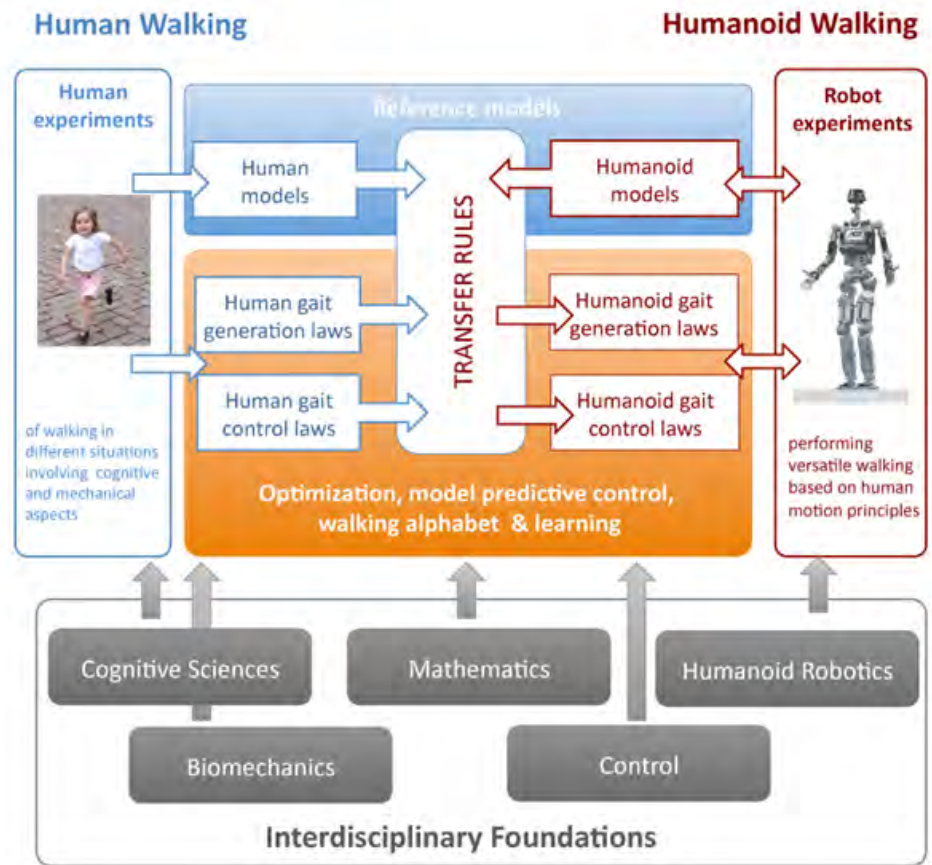


Figure 1: Graphical representation of the scientific approach of the KoroiBot project.

## Context

This thesis has been written in the context of the European project KoroiBot (<http://www.koroirobot.eu/>). The goal of the KoroiBot project is to enhance the ability of humanoid robots to walk in a dynamic and versatile way, and to bring them closer to human capabilities. As depicted in Fig. 1, the KoroiBot project partners have to study human motions and use this knowledge to control humanoid robots via optimal control methods. Human motions are recorded with motion capture systems and stored in an open source data bank which can be found at <https://koroirobot-motion-database.humanoids.kit.edu/>. With these data several possibilities are exploited. Assuming that humans minimize some criteria we can use inverse optimal control methods to find those criteria. Walking alphabet and learning method [Mandery 2016] are also used to transfer human behaviors to robots. Finally, optimal control is used to build controllers that can eventually use these learned human behaviors while ensuring the robot safety. Research and innovation works in KoroiBot mainly target novel motion control methods for existing hardware. It also derives optimized design principles for next robot generations.

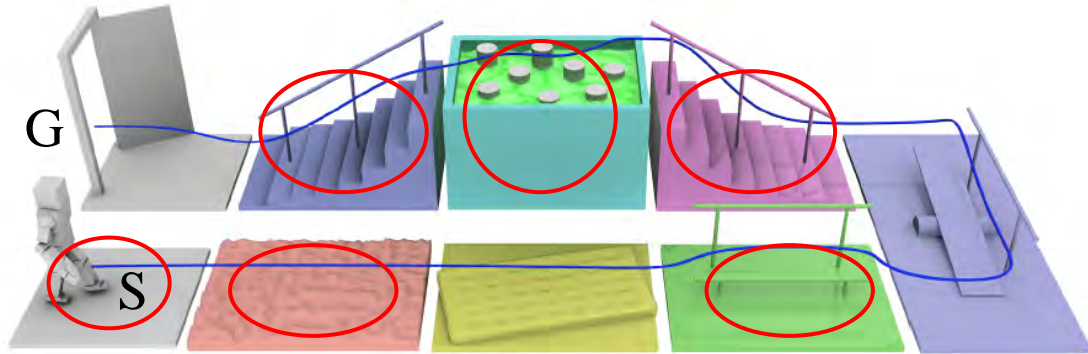


Figure 2: Challenges of the KoroiBot project. In red the challenges chosen by the LAAS-CNRS.

### KoroiBot and robot challenges

In addition to this ambitious scientific aspect of the project, there is an important technical component. Following the blue line in Fig. 2, the different challenges of the project are to make humanoid robots:

- walking on a flat ground,
- walking on an uneven ground,
- walking on a mattress,
- walking on a beam with/without handrail,
- walking on a seesaw with/without handrail,
- climbing a stair case with/without handrail,
- walking on stepping stones,
- going down a stair case with/without handrail,
- and walking through a door.

All the team owning a robot has to perform some of these challenges. In Fig. 3 we can see all the robot hosted by different partners. Ideally, the algorithms developed in the frame of the KoroiBot project has to be integrated on all the robots. In practice the partners chose parts of the challenge to be realized on their different robotic platforms

### KoroiBot Key Performance Indicator (KPI)

In the European project KoroiBot we need to evaluate the improvement in terms of robot control and human likeness. In this context and in collaboration with the H2R project, a detailed set of key performance indicators (KPI) have been proposed [Torricelli 2015]. These KPI try to capture all the bipedal locomotion patterns.

A set of specific sub-function of the global motor behavior is analyzed (see Fig. 4). The different sub-function are divided in two. First, the sub-function associated to body posture task with no locomotion. And second the same sub-function but including the robot body transport. The initial condition may vary depending on the experiment to perform. This is the idea of the



*Figure 3: List of the humanoid robot used in the KoroïBot project.*

























intertrial variability. For example standing still on a horizontal flat surface is easy to reproduce at will. Hence there is no intertrial variability. On the contrary, while doing push recovery, it is difficult to reproduce the exact same push several times. The sub-function are also classified taking into account the changes in the environment or not.

Each of these functions can be evaluated for different robots using the criteria depicted in Fig. 5. The performance are classified in two sub categories, quantitative performance and human likeness. In addition there are indications on the last two columns if the criteria is applicable on a standing task or on a locomotion task.

All the partners owning a humanoid robot need to perform an evaluation of these KPI. However some movement are not possible yet on some robots of the consortium. Hence, the different team picked meaningful criteria considering the current state of their robot and controllers.

### **The work done in the KoroïBot context**

In LAAS-CNRS, the Gepetto team own the HRP-2 and the Romeo robots depicted respectively as the second and fourth robot in Fig. 3 from the left to the right. The Romeo robot is the

		Function		
		Body Posture	Body Transport	
Environment	Stationary	No	Static Horizontal Surface 	Horizontal Ground at Constant Speed 
			Static Inclined Surface 	Sloped Ground 
		Yes	Different Static Surfaces  	Stairs 
				Variable Slopes 
				Irregular Terrain 
				Slippery Surface 
	In Motion	No	Continuous Surface Tilts 	Treadmill at Constant Speed 
			Continuous Surface Translations 	Soft Terrain with Constant Compliance 
			Bearing Constant Weight 	Bearing Constant Weight 
		Yes	Pushes 	Pushes 
			Sudden or Pseudorandom Surface Tilts 	Treadmill at Variable Speed (Including Start-Stop) 
			Sudden Surface Translations 	Seesaw 
			Body Sway Referenced Platform (BSRP) 	Soft Ground with Variable Compliance 

Note: All the disturbances may be applied in sagittal and frontal directions.

Figure 4: The motor skills considered in the benchmarking scheme. This scheme is limited to bipedal locomotion skills. The concept of intertrial variability is analogous to the concept of unexpected disturbances.



		Abilities		Benchmarks		
		Name	Description	Benchmark	Applicability	
					Posture	Transport
Performance	Stability	Intratrial Stability	Ability to Maintain Equilibrium Within a Single Trial	Time Until Falling	X	
				Cycles Until Falling	X	X
		Intertrial Stability	Ability to Maintain the Equilibrium Across Different Trials	Success Rate Across $N$ Different Trials	X	X
				Gross Body Equilibrium	Ability to Maintain Equilibrium Over the Base of Support	Energy Stability Margin (ESM)
		Maximum Accepted Disturbance Amplitude	X			X
		Maximum Accepted Disturbance Frequency	X			X
	Efficiency	Global Energy Consumption	Ability to Transport Body with Low Energetic Costs	Specific Energetic cost of Transport $C_{et}$		X
				Specific Mechanical Cost of Transport $C_{mt}$		X
		Passivity	Ability to Minimize Joint Torques During Walking	Passive Gait Measure (PGM)		X
		Reaction Time	Ability to Promptly React to Disturbance or External Command	Time from Input and Initiation of Motor Action	X	X
Human Likeness	Kinematics	Gross Body Motion	Motion of the Whole Body Expressed by Global Variables	CoM Trajectory (Correlation, Dynamic Time Warping)	X	X
				Gait Harmony		X
				Body Sway (Frequency Response Function)	X	
				Natural Looking Motion	X	X
		Individual Joint Motion	Motion of the Single Joints or Limbs Taken Separately	Joint Trajectory (Correlation, Dynamic Time Warping)	X	X
				Knee, Ankle Forefoot Rocker		X
		Interlimb Coordination	Ability to Coordinate Between Different Body Parts	Symmetry (Ratio Index)	X	X
				Trunk/Arm Motion	X	X
	Intralimb Coordination	Ability to Move Multiple Joints of the Same Limb Coordinately	Kinematic Synergies	X	X	
	Dynamics	Gross Body Kinetics	Forces Exerted Between the Whole Body and the Environment	Ground Reaction Forces (Correlation, Dynamic Time Warping)	X	X
		Single-Joint Kinetics	Force Exerted Among Limbs	Joint Torques (Correlation, Dynamic Time Warping)	X	X
		Dynamic Similarity	Ability of Having Leg Pattern Dynamically Similar to Most Legged Animals	Froude Number (Dimensionless Gait Velocity)		X
		Dynamicity	Ability to Use Falling State for Body Progression	Dynamic Gait Measure (DGM)		X
		External Compliance	Ability to Respond Resiliently to External Disturbances	Impulse Response Function (IRF)	X	X
		Internal Compliance	Ability to Store and Release Energy	Active/Net Joint Torque	X	X

Figure 5: The motor abilities and related benchmarks are classified in two categories: performance and human likeness. The performance category includes all those abilities related to stability (ability of maintaining equilibrium) and efficiency. The human likeness category includes all those abilities related to typical human behavior, under the perspective of kinematics and dynamics. For each ability, a specific benchmark has been identified. The last column specifies in what classes of motor skills (i.e., the function category of Fig. 4) the corresponding benchmark is applicable.

first human size prototype of Softbank Robotics and has very limited locomotion capabilities. Therefore I was in charge of integrating our algorithms on the HRP-2 robot. Among the challenges from Fig. 2 we picked the one with a red circle, i.e:

- walking on a flat ground,
- walking on an uneven ground,
- walking on a beam with/without handrail,
- climbing a stair case with/without handrail,
- walking on stepping stones,
- and going down a stair case with/without handrail.

In addition to those challenges we added the perturbation rejection. At the beginning of the project we evaluated the KPI on the HRP-2 robot. In our case, we picked the KPI sub-functions meaningful considering the above selected challenges:

- horizontal ground at constant speed,
- stairs,
- soft terrain with constant compliance,
- bearing constant weight (the robot's own weight).

Coupled with the following benchmarks:

- success rate across N different trials,
- specific energy cost of transport,
- specific mechanical cost of transport,

All these choices are shown in the tables from Fig. 4 and Fig. 5 by red ellipses. The mathematical details and results are presented below in section "KoroiBot Key Performance Indicator (KPI)".

**Reactive walking** The results pointed out some interesting points. First the ground walking velocity and versatility of HRP-2 can be improved. In this context we did a collaboration, presented in Chap. 1, with our mathematician colleagues from the University of Heidelberg. This collaboration leads to a new real time walking pattern generator with increased functionalities like obstacle avoidance. In addition to this, the implementation of the dynamic filter, presented by Nishiwaki and al. in [Nishiwaki 2009], increase the range of the reachable CoM velocity.

**Multicontact motion generation** The second problem that arose from the KPI was the energy consumption during the stair climbing. One possible way for solving this problem is to distribute the robot weight over multiple contacts. Hence, we designed an innovative multi-contact controller to allow generic locomotion and the use of a handrail. Chap. 2 explains this contribution in details.

**Fire hose manipulation** The second application concerns perturbation rejection and has been done in a collaboration with the Japanese national institute of Advanced Industrial Science and Technology (AIST). This application is inspired from the DARPA robotic challenge. The idea of this work is to see if a humanoid robot with average power and size like HRP-2 is able, using a state of the art controller, to pull a stiff fire hose. The details of the experiment are written in Chap. 3



**One third power law** In the frame of the KoroiBot project, researchers studied the human motion to extract quantitative data. In fact they noticed that humans adapt their velocities in function of the curvature of their trajectories. The scientific question that we tried to answer is the following: "Can this law extracted from human motion help humanoid robots to walk ?" A fruitful collaboration with these experts allowed us to answer this question. This resulted in the design of an innovating controller for tracking cyclic trajectories of the center of mass. Chap. 4 present this work in details.

**Human motion primitives and model predictive control** Another collaboration with human motion experts was done to study if motion primitives extracted from human behavior could be applied to humanoid robot. To answer this question we propose a whole body controller using upper body movement primitives extracted from human behavior and lower body movement computed by a walking pattern generator. Chap. 5 show these fused bottom up and top down approaches.

**Reactive controllers** In this thesis we studied reactive behavior for humanoid robot. We designed applications that make the robot facing real case scenarios. The first application was done in the frame of a collaboration with Airbus/Future of Aircraft Factory. The idea was to build controllers for test case scenarios and evaluate if humanoid robot could go in a factory. We used online planner for obstacle avoidance, visual servoing coupled with a walking pattern generator to place the robot in a desired position, a whole body controller for screwing tasks, and center of pressure based walking pattern generator to climb stairs. Chap. 6 explains this contribution in details.

## Problem statement and state of the art

### Problem Statement

Robot behavior realization can be formalized using mathematical optimization. Considering a robot with  $N$  degrees of freedom,  $\mathbf{q}$  a vector of information on its internal parameters and the environment  $\mathbf{v} \in \mathcal{R}^m$ . For a given behavior let us assume that it exists a function  $f(\mathbf{q}, \mathbf{v}, t) : \mathcal{R}^{n \times m + 1} \rightarrow [0, 1]$  such as it is equal to 0 when the behavior is realized. The problem amount to find a trajectory  $\mathbf{q}(t)$  such that

$$\begin{aligned} & \text{minimize} && f(\mathbf{q}, \mathbf{v}, t) \\ & \text{subject to} && g(\mathbf{q}, \mathbf{v}, t) < 0 \\ & && h(\mathbf{q}, \mathbf{v}, t) = 0 \end{aligned} \tag{1}$$

with  $g$  being unilateral constraint and  $h$  bilateral constraints.

A common approach in robotics is to build an approximation  $\hat{f}$  of  $f$ . It depends on the estimated current state of the environment  $\hat{\mathbf{v}}(t)$ , and an estimation of the current state of the robot in this environment  $\hat{\mathbf{q}}(t)$ . The formulations of  $\hat{\mathbf{v}}(t)$  and  $\hat{\mathbf{q}}(t)$  are injected in (1) to solve the optimization problem. In this document we will not address the problem of building  $\hat{\mathbf{v}}$  but assume that an appropriate software module is providing the necessary information. Therefore

we will assume that a geometric description of the environment is available, and that a system is able to give a sufficiently accurate position of the robot in this environment. Practically this is done through a motion capture system. In the following we will present the formulation of (1) for the locomotion problem.

### The locomotion generic optimal control problem

The state of the problem  $\mathbf{x}$  is usually composed of the robot whole body configuration coupled with the whole body velocity. Let us now denote by  $\mathbf{q}$  the whole body configuration and by  $\dot{\mathbf{q}}$  the whole body velocity. The future contact point can be precomputed by a planner or included in the state of the problem. The control of this system  $\mathbf{u}$ , can be the next derivative of the state, i.e  $\ddot{\mathbf{q}}$ , or the contact wrench  $\phi = [\mathbf{f}_k \ \tau_k]^T$  with  $k \in \{0, \dots, K\}$ ,  $K$  being the number of contact. Or the motor torques  $\mathbf{T}$ . We denote by  $\underline{\mathbf{x}}$  and  $\underline{\mathbf{u}}$  the state and control trajectories.

The main idea here is to be able to compute joint trajectories satisfying the general equation of the dynamics, the initial and terminal constraints, and keeping balance. The following optimal control problem (OCP) represents a generic form of the locomotion problem:

$$\min_{\underline{\mathbf{x}}, \underline{\mathbf{u}}} \sum_{s=1}^S \int_{t_s}^{t_s + \Delta t_s} \ell_s(\mathbf{x}, \mathbf{u}) dt \quad (2a)$$

$$s.t. \quad \forall t \quad \dot{\mathbf{x}} = \text{dyn}(\mathbf{x}, \mathbf{u}) \quad (2b)$$

$$\forall t \quad \phi \in \mathcal{K} \quad (2c)$$

$$\forall t \quad \mathbf{x} \in \mathcal{B}_x \quad (2d)$$

$$\forall t \quad \mathbf{u} \in \mathcal{B}_u \quad (2e)$$

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (2f)$$

$$\mathbf{x}(T) \in \mathcal{X}_* \quad (2g)$$

where  $t_{s+1} = t_s + \Delta t_s$  is the start time of the phase  $s$  (with  $t_0 = 0$  and  $t_S = T$ ). Constraint (2b) makes sure that the motion is dynamically consistent. Constraint (2c) enforces balance with respect to the contact model. Constraints (2d) and (2e) imposes some bounds on the state and the control. Constraint (2f) imposes the trajectory to start from a given state (typically estimated by the sensor of the real robot). Constraint (2g) typically imposes the terminal state to be viable [Wieber 2008]. The cost (2a) is typically decoupled  $\ell_s(\mathbf{x}, \mathbf{u}) = \ell_x(\mathbf{x}) + \ell_u(\mathbf{u})$  whose parameters may vary depending on the phase.  $\ell_x$  is generally used to regularize and to smooth the state trajectory while  $\ell_u$  tends to minimize the forces, then producing a more dynamic movement. The resulting control is stable as soon as  $\ell_x$  comprehends the L-2 norm of one time derivative of the robot center of mass (CoM) denoted by  $\mathbf{c}$ , [Wieber 2015]. Problem (2) is a difficult problem to solve in its generic form. And specifically (2b) is a challenging constraint. Most of the time the shape of the problem varies from one solver to another only on the formulation of this constraint. Hence, in the next section we will describe this equation in more details.

## Robot dynamics

In this section we will present the instantaneous dynamics of a poly-articulated rigid system based on [Orin 2013] and [Kajita 2003b]. We will then present some humanoid robotics specific formulation.

### General formulation of a robot dynamic

In general the Lagrangian dynamics of a robot is expressed :

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{S}^T \mathbf{T} + \sum_{k=1}^K \mathbf{J}_k^T(\mathbf{q}) \begin{bmatrix} \mathbf{f}_k \\ \boldsymbol{\tau}_k \end{bmatrix} \quad (3)$$

with  $\mathbf{q} = [\mathbf{x} \ \boldsymbol{\theta} \ \hat{\mathbf{q}}]^T$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$  being the generalized state of the robot of size  $\mathcal{R}^{dim(SE(3))+N}$  and its derivatives, and  $N$  being the number of actuated joints of the system. Lets assume that  $dim(SE(3)) = 6$  to consider 3 translation and 3 rotations. It is composed by the free flyer position  $\mathbf{x}$  and orientation  $\boldsymbol{\theta}$  which is the position of an arbitrary joint position and orientation in the ground frame, typically the center of the robot pelvis.  $\hat{\mathbf{q}}$  is the configuration vector of the joints. The matrix  $\mathbf{M} \in \mathcal{R}^{(6+N) \times (6+N)}$  is the generalized inertia matrix described in [Wieber 2005] and [Sherikov 2016].  $\mathbf{C}$  models the centrifugal and Coriolis effects,  $\mathbf{G}$  is the action of the gravity field,  $\mathbf{S} = [\mathbf{0}_{N \times 6} \ \mathbf{I}_{N \times N}]$  is a matrix selecting the joint torques  $\mathbf{T}$ .  $\mathbf{J}_k^T(\mathbf{q})$  is the contact Jacobian,  $\mathbf{f}_k$  and  $\boldsymbol{\tau}_k$  are the forces and torques applied at the contact  $k$ . The first 6 equations of Eq. 3 correspond to the Newton-Euler equations.

### Underactuated dynamics and centroidal momentum

In [Orin 2013], the above equations are reformulated to express the centroidal momentum dynamics :

$$\mathbf{h}_G = \mathbf{A}_G(\mathbf{q})\dot{\mathbf{q}} \quad (4)$$

with  $\mathbf{h}_G = [\mathcal{K} \ \mathcal{L}]^T \in \mathcal{R}^6$  being the spatial momentum composed by the linear momentum  $\mathcal{K}$  and the angular one  $\mathcal{L}$ .  $\mathbf{A}_G$  is the first six lines of the inertia matrix  $\mathbf{M}$ . If we express the time derivative of the robot total momentum expressed at the center of mass (4) we get the first six lines of (3):

$$\dot{\mathcal{K}} = m\ddot{\mathbf{c}} = \sum_{k=1}^K {}^0R_k \mathbf{f}_k + m\mathbf{g} \quad (5)$$

$$\dot{\mathcal{L}} = \sum_{k=1}^K (\mathbf{p}_k - \mathbf{c}) \times {}^0R_k \mathbf{f}_k + \boldsymbol{\tau}_k \quad (6)$$

with again  $\mathbf{f}_k$  being the forces expressed at the contact point in the contact frame,  ${}^0R_k$  the rotation matrix between the contact frame and the inertia frame,  $\mathbf{p}_k$  the contact point position,  $\mathbf{c}$  the robot center of mass position, and  $m$  the robot mass. After a simple rewriting we obtain:

$$m(\ddot{\mathbf{c}} - \mathbf{g}) = \sum_{k=1}^K {}^0R_k \mathbf{f}_k \quad (7)$$

$$m\mathbf{c} \times (\ddot{\mathbf{c}} - \mathbf{g}) + \dot{\mathcal{L}} = \sum_{k=1}^K \mathbf{p}_k \times {}^0R_k \mathbf{f}_k + \boldsymbol{\tau}_k \quad (8)$$

This formulation shows that the dynamics of a poly-articulated robot can be reduced into its center of mass. The influence of each body on the center of mass is express through the term  $\dot{\mathcal{L}}$ . This term express the influence of the translation and rotation of each body on the center of mass dynamics.

## The control horizon

The equations expressed in Eq. 3 are expressed instantaneously. However controlling a robot dynamical movement needs anticipation due to the inertia. A control can be instantaneously satisfactory at time  $t$ , but not at time  $t + \Delta t$  because of the presence of an obstacle for example. Imagine a car moving at constant speed ( $70 \text{ km h}^{-1}$ ) with an instantaneous controller. This controller will not take into account that in few second the car needs to do a 90 deg turn. So when the turn comes the car has exactly  $\Delta t$  seconds to react. Obviously the car will crash. On the contrary a human driver will anticipate the turn and slow down the vehicle before turning. That is the reason why anticipation and future prediction is needed. The inconvenient is that the complexity of the problem is equal to number of freedom multiply by the horizon size. Typically, HRP-2 has 36 degrees of freedom and we use a preview duration of 1.6 s (360) which gives 12960 free variables. This kind of problem cannot be solved yet in 5 ms with limited computational resources (see Fig. 6). The vertical axis corresponds to the instantaneous problem and the horizontal one show the size of the predicted future. In the following section we will present the optimal control problem for the locomotion of humanoid robot that researchers need to solve.

In the following we list some of the main algorithms solving (2) and show how they correspond to some specific choices of the generic template.

## State of the Art

### Hybrid controls

Other approaches exists for making robots walk on two legs. Historically passive walkers aims at achieving long walking behavior with a low energy footprint. This is realized through mechanical storage systems such as springs (DURUS [Reher 2016]), and efficient transmissions. In general the control approach tries to consider mainly the centroidal dynamics and uses stability criteria such as the Lyapunov criteria or the Poincaré map. They are less computationally expensive than optimal control methods. It has been shown in [Razavi 2015] that a stability criteria can be found using the method of Poincaré on the centroidal momentum. The authors were then able to implement an efficient controller law from this stability criteria. The work proposed by [Kaddar 2015] shows how to generate whole body dynamical motion using the same approach but using the whole body dynamic.

Those controllers are less expensive than optimal control because they required 1 control per phase. Typically for walking on flat floor, 2 phases are taken into account: single support and double support phase. On the contrary, optimal control methods need to discretize the dynamics and then to compute one control variable for each discretization step. Classically for walking 8 control variables needs to be computed for one step.

For standard walking the hybrid control is a good method [Westervelt 2007]. In fact, in free space with infinite flat ground walking motions can be considered as cyclic. Periodic phase and dynamics can be extracted from the Newton-Euler equations and in particular from the centroidal momentum.

One major difficulty is that aperiodic gait are complicated to manage as it would need a large number of controllers to drive the robot. Walking on uneven ground and maneuvering around an obstacles are typical case where the gait is aperiodic [Grizzle 2010]. In this thesis we will have to handle such cases like going through stepping stones or maneuver around obstacles, etc.

One major advantage of this method is that discontinuous phenomena, like impacts when the lands, are easily modeled using hybrid control. The impacts can then be manage by the controller after they occur. In the case of optimal control method it is rather expensive to include such phenomena. However in the context of the KoroiBotproject the motion we generated are not dynamical enough to consider huge impacts. To handle the foot landing we design sufficiently smooth trajectories to not provoke this kind of discontinuities in the robot states. Typically the foot trajectory ends with zero acceleration and velocity. Moreover, on HRP-2 specifically, there is a closed loop stabilizer which avoid perturbations and fit at best the smooth trajectories. Hence in this thesis we will focus on using optimal control methods.

## Whole body formulations

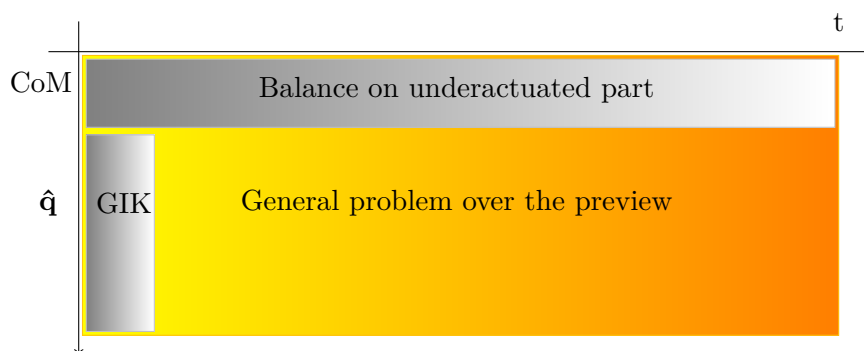


Figure 6: Classical approach (in gray) organization vs. a more global handling of the problem [Todorov 2014]

Fig. 6 depicts the classical approaches used so far. Indeed the full problem is nonlinear and has around 10 thousand variables and is represented by the whole rectangle. To be able to solve it researchers used heavy framework for nonlinear optimization. In fact in [Koch 2014] the authors implemented the above OCP and generated trajectories for the HRP-2 robot. The software used in this context computes multiple shooting method to solve nonlinear problems. The solution obtained was a smooth trajectory that allow a HRP-2 robot to step over a 20 cm

high obstacle. The experiment has been done in LAAS-CNRS. This motion is, for now in LAAS-CNRS, the record in terms of obstacle height. The bottom heck of this approach is the computation time. These trajectories took several ours to be computed. Another formulation implement the complete problem but use simplifications in the contact model. In [Tassa 2014] the authors uses Differential Dynamic Programming (DDP) to solve the problem. This is a very efficient way to compute a solution. However the current contact model is a spring and damper model. It produce good enough contact forces to perform multicontact reaching. However no walking could be performed as the the model provides none dynamically consistent force. An implementation of this software has been used with HRP-2 in [Koenemann 2015]. The solver was installed in a remote computer as it still needs multiple powerful CPUs to be used online. In [Koenemann 2015] the authors used a off board non-real-time Windows on a 12-core  $4GHz$  CPU that computes the DDP under  $100ms$ . The PID controller of the robot runs on an embedded computer using a real time linux with a  $2.93GHz$  CPU. The connection between the two computer is done via wifi.

To summarize this, there are not yet powerful techniques that can be used with limited time and computational resources. As a consequence researchers try to simplify the problem. The next paragraph presents these approaches.

## Mixed formulation

One simplification consist in taking into account the future of the under-actuated part of the dynamic plus the whole body instantaneous dynamics. In Fig. 6 this would correspond to the two gray areas but fused together.

During the DARPA robotics challenge the authors of [Kuindersma 2014] used a custom active set solver for quadratic programming to solve a mixed formulation. For the under-actuated formulation they used the seminal work of Kajita [Kajita 2003a]. In addition they used linearized friction cones and optimized the robot joint accelerations.

A recent thesis [Sherikov 2016] pursue this work of integrating under-actuated preview control inside a instantaneous whole body controller. The main difference between Sherikov and Kuindersma’s work is the under-actuated model predictive control used. In fact the walking task of the Sherikov’s framework is able to automatically find the foot steps following the formalism of [Herd 2010a]. In Kuindersma’s framework the foot steps are precomputed by a planner.

## Walking patterns in 3D

From this subsection on, the hypothesis of separating the under-actuated and whole body motion is made. The hypothesis is most used in the literature and correspond to the two separated gray rectangles in Fig. 6. This means that trajectories for the CoM, the free flyer and the end effectors are computed first. And only afterward the whole body control is deduced from these preliminary calculations. The work presented in this section are locomotion algorithm allowing multiple contacts.

An iterative scheme is proposed in [Hirukawa 2007] that can be written as an implicit optimization scheme whose cost function is the distance to a given CoM trajectory and given forces distributions. The resulting forces satisfies (2c) by construction of the solution. There is no condition on the angular momentum (6) neither on the viability of the final state (2g),

however the reference trajectory enforced by the cost function is very likely to play the same role.

In [Qiu 2011], (2c) is explicitly handled (using the classic linear approximation of the quadratic cones). As in [Perrin 2015], (2g) is indirectly handled by minimizing the jerk. No condition on the angular momentum (6) is considered. Additionally, the proposed cost function maximizes the robustness of the computed forces  $\phi$  and minimizes the execution time. Finally, constraints are added to represent the limitation of the robot kinematics.

In [Perrin 2015],  $\dot{\mathcal{L}}$  is null by construction of the solution. Moreover, (2c) is supposed to always hold by hypothesis and is not checked, while (2g) is not considered but tends to be enforced by minimizing the norm of the jerk of the CoM, like in [Kajita 2003a]. These assumptions result in an (bilinear)-constrained quadratic program that is solved by a dedicated numerical method.

In [Rotella 2015], (2c) is handled under a simple closed form solution, while (2g) is not considered. To stabilize the resolution, the cost function tends to stay close to an initial trajectory of both the CoM and the angular momentum, computed beforehand from a kinematic path. Consequently, (6) is not considered either (as it will simply stay close to the initial guess).

## Walking patterns in 2D

In addition to the previous remarks, another difficulty is the bilinear form of the dynamics (8). When the contacts are all taken on a same plane, a clever reformulation of the dynamics makes it linear [Kajita 2003a], by neglecting the dynamics of both the CoM altitude and the angular momentum. In that case, (8) boils down to the constraint of the center of pressure point (CoP) to be in the support polygon.

Kajita et al. [Kajita 2003a] did not explicitly check the constraint (2c). In exchange,  $\ell_u$  is used to keep the control trajectory close to a reference trajectory provided a priori. Similarly, (2g) is not checked either. In exchange,  $\ell_x$  tends to stabilize the robot at the end of the trajectory by minimizing the jerk of the CoM. These three simplifications turns (2) into a simple unconstrained problem of linear-quadratic regulation that is implicitly solved by integrating the corresponding Riccati equation. Interesting motions can be performed using this walking pattern generator. [Evrard 2009] shows that tele-cooperation was possible using humanoid robot and the walking pattern generator from [Kajita 2003a].

The LQR was reformulated into an explicit OCP [Herdt 2010b], directly solved as quadratic program. The OCP formulation makes possible the formulation of inequality constraints. (2c) is then explicitly checked under its CoP form.

A modification of this OCP is proposed in [Sherikov 2014] where (2g) is nicely approximated by the capturability constraint, which is a linear constraint on the CoM and its first time derivative in case of 2D contacts.

## Computing the contact placements

When considering an explicit OCP formulation, additional static variables can be added to the problem. Typically, the placement of the contact, that are given as invariant in (2), might be computed at the same time. This was first proposed in [Herdt 2010b] for a 2D WPG, and similarly used in [Sherikov 2014] and other works by the same authors. Similarly, it was proposed in [Rotella 2015] to include it in the proposed 3D WPG, but this feature was not implemented

nor demonstrated. In both cases, the placements of the contacts are unlimited (or similarly limited to a convex compact set). The problem becomes much harder when the contacts might be taken among a discrete set of placements. In [Deits 2014], the problem was formulated as a mixed-integer program (i.e. having both continuous and discrete variables) in case of flat contact, and solved using an interior-point solver to handle the discrete constraints. In [Mordatch 2012], the same problem is handled using a dedicated solver relying on a continuation heuristic, and demonstrated for animating the motion of virtual avatars.

This is not an exhaustive bibliography. Many papers need to be cited but also replaced in their scientific context. So for sake of clarity, a more detailed bibliography is written at the beginning of each chapter. This allows a better understanding on the state-of-art which is specific to each chapter of this thesis.

## KoroiBot Key Performance Indicator (KPI)

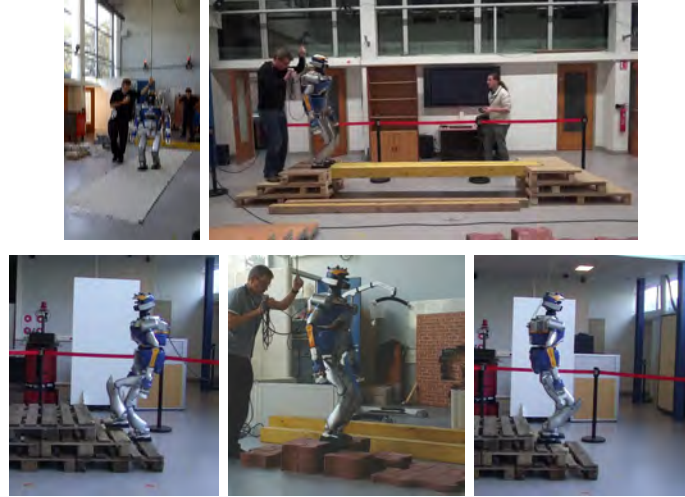


Figure 7: Sample of the experimental setup of the KoroiBot project in LAAS-CNRS

In the KoroiBot project we used key performance indicators (KPI) to analyze the behavior of the robot at the beginning and at the end of the project. These results lead us toward the improvements to be made. In 2013 the algorithm mostly used and implemented on HRP-2 in LAAS-CNRS where the walking pattern generators described in [Morisawa 2007] and in [Herd 2010a]. The performance indicators chosen were:

- the execution time  $T_M$ ,
- the time to compute the motion  $T_{think}$ ,
- the total energy consumed during a walking distance  $D$ ,

$$E_{walk} = \int_{t_{begin}}^{t_{end}} \tau \omega dt + \int_{t_{begin}}^{t_{end}} R k_c^2 \tau^2 dt, \quad \tilde{E}_{walk} = E_{walk} / D$$

with  $E_{walk}$  being the total energy consumed counting the integral over time of the mechanical power and the electrical power dissipated by the motors.  $\tau$  and  $\omega$  being respectively the



joint torques and velocities, and  $R$  with  $k_c$  being respectively the motor resistances and torque constants.

- the mechanical energy consumed during a walking distance  $D$ ,

$$E_{meca} = \int_{t_{begin}}^{t_{end}} \tau \omega dt, \quad \tilde{E}_{meca} = E_{meca}/D$$

with  $E_{meca}$  being the integral over time of the mechanical power,

- the deviation from the planned trajectory  $P_{T_r}$ ,
- and the maximum speed reached  $V_{max}$ .

The trajectories were generated offline and repeatedly played on the robot to analyze their robustness. Samples of the experimental setups can be seen in Fig. 7.

### Straight walking

$T_M$	$T_{think}$	$\tilde{E}_{walk}$	$\tilde{E}_{meca}$	$P_{T_r}^\theta$	$P_{T_r}^x$	$P_{T_r}^y$	$V_{max}$
7 s	2 s	47250 J/m	47070 J/m	-0.0013 rad	0.026 m	0.052 m	0.125 m/s

From these results we can observe that the HRP-2 is very repeatable. However the room for improvement lies in the maximum velocity speed which is quite slow.

### 3D walking

Here 3D walking means walking on flat ground but possibly a different height, like stair case or stepping stones. The walking pattern generators [Morisawa 2007, Herdt 2010a] normally does not allow the possibility to perform movement such as climbing stairs. For this reason we designed smooth feet 3D trajectories using B-Splines and a prior knowledge of the environment. Empirically we found trajectories that fits kinematics and dynamics constraints. The results of these implementation is shown in [Naveau 2014]

### Walking in stairs or on stepping stones

$T_M$	$T_{think}$	$\tilde{E}_{walk}$	$\tilde{E}_{meca}$	$P_{T_r}$	$V_{max}$
7 s	2 s	256150, J/m	255270 J/m	-	-

This results shows the KPI value when the robot goes up the stair case. The important consumption of energy during the climbing up stairs is to be noted. Diminishing this power consumption is an important factor as it will decrease the loss of electrical energy and the mechanical stress on the robot. The repeatability of this experiment is around 40 %.

The results of the KPI, when the robot goes downstairs, are very similar to the going up experiment. The repeatability of the going down stairs motion is around 95 %. The only point is that the global energy is twice lower than when the robot is going up. The reason is that it is less fighting the gravity while going down.

For the stepping stone experiment the KPI were also very similar to the above two experiments. Though, as expected, the energy consumption is between going up and going down stairs. The repeatability of this experiment is around 50 %.

## Walking on a beam

The major problem of this experiment is its success rate which is around 20 %. In order to improve this rate we need to take care of the robot balance and the robot feet placement. Indeed the two main reasons of the robot fall were the default of balance on the beam and the drift which is important as the beam length is 3 *m* long.

## Step over obstacles

This manipulation has not been performed in the frame of this thesis. It is a work published in [Koch 2014] already presented in the above sections. The improvement to be made here is on the computation time. As explained before in subsection "Whole body formulations", these trajectory took several hours to be computed.

## Contributions

The scientific contributions of my thesis are :

- the design a novel real time walking pattern generator using nonlinear model predictive control (Chap. 1),
- the implementation of two multicontact pattern generators using the centroidal dynamics. They both take the contact placements as input and produce dynamically stable center of mass (CoM) motion (Chap. 2).

In the context of the KoroBot project several technical contributions are to be noted:

- the integration of a hose manipulation by HRP-2. The robot pick the hose from the floor and pull it to a desired position without falling,
- the inclusion of the one third power law as control law for humanoid walking,
- the elaboration of transfer rules to use upper body human motions integrated with a walking pattern generator (Chap. 5),
- the study of potential application for humanoid robots in an industrial environment (Chap. 6) including,
  - the use of fast planning in order to make HRP-2 walking towards a desired position while avoiding obstacles,
  - the integration of visual servoing to track a target position while walking,
  - the use of the whole body motion generator (Stack-of-Tasks) to evaluate the kinematic feasibility of screwing motions,
  - the design of feet 3D trajectory to allow the robot to climb stairs or go across stepping stones and beams,
- the implementation of the dynamic filter on the LAAS-CNRS HRP-2 (An. A),

## Publications

### Journal

#### Accepted

- M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur and P. Soueres. *A Reactive Walking Pattern Generator Based on Nonlinear Model Predictive Control*. Robotics and Automation Letters, 2017

#### Under revision

- A. Mukovskiy, C. Vassallo, M. Naveau, O. Stasse, P. Souères and M. A. Giese. *Learning Movement Primitives for the Humanoid Robot HRP2*, 2016. (Submitted)
- A. Orthey, V. Ivan, M. Naveau, Y. Yang, O. Stasse and S. Vijayakumar. *Homotopic particle motion planning for humanoid robotics*. 2015. (Submitted)

### Conference

#### Published

- M. Karklinsky, M. Naveau, A. Mukovskiy, O. Stasse, T. Flash and P. Soueres. *Robust human-inspired power law trajectories for humanoid HRP-2 robot*. In Int. Conf. on Biomedical Robotics and Biomechatronics, 2016
- J. Carpentier, S. Tonneau, M. Naveau, O. Stasse and N. Mansard. *A Versatile and Efficient Pattern Generator for Generalized Legged Locomotion*. In Int. Conf. on Robotics and Automation, 2015
- M. Kudruss, M. Naveau, O. Stasse, N. Mansard, C. Kirches, P. Soueres and K. Mombaur. *Optimal Control for Multi-Contact, Whole-Body Motion Generation using Center-of-Mass Dynamics for Multi-Contact Situations*. In Int. Conf. on Humanoid Robotics, 2015
- M. Naveau, J. Carpentier, S. Barthelemy, O. Stasse and P. Soueres. *METAPOD - Template META-PrOgramming applied to Dynamics: CoP-CoM trajectories filtering*. In Int. Conf. on Humanoid Robotics, 2014
- O. Stasse, A. Orthey, F. Morsillo, M. Geisert, N. Mansard, M. Naveau and C. Vassallo. *Airbus/Future of Aircraft Factory, HRP-2 as Universal Worker Proof of Concept*. In IEEE/RAS International Conference on Humanoid Robot (ICHR), 2014

#### Submitted

- I. G. Ramirez-Alpizar, M. Naveau, C. Benazeth, O. Stasse, J.-P. Laumond, K. Harada and E. Yoshida. *Motion Generation for Pulling a Fire Hose by a Humanoid Robot*, 2016. (Submitted)



# NMPC walking pattern generator

---



In the present chapter we present a real-time nonlinear model predictive control (NMPC) executable on the humanoid robot HRP-2. This chapter has been developed in the frame of the collaborative project KoroBot and published in [Naveau 2017]. Following the idea of “walking without thinking”, we propose a walking pattern generator that takes into account simultaneously the position and orientation of the feet. A requirement for an application in real-world scenarios is the avoidance of obstacles. Therefore, an extension of the pattern generator that directly considers the avoidance of obstacles is derived. The algorithm uses the whole-body dynamics to correct the center of mass trajectory of the underlying simplified model. The pattern generator runs in real-time on the embedded hardware of the humanoid robot HRP-2 and experiments demonstrate the increase in performance with the correction.

In Sec. 1.1.1 we present the motivation and the related works. Sec. 1.2 is a reminder of the LIPM equation as well as the principles used in [Herdt 2010a]. Sec. 1.3 depicts the formulation of the problem as a sequence of locally linearized quadratic problems and the real-time feasible solution by applying the idea of the so called “real-time” iteration. A particular treatment of the dynamical filter is given in Sec. 1.4. Finally, our practical contribution, showing that the algorithm can be implemented in real-time on the humanoid robot HRP-2, is detailed in Sec. 1.5.

## 1.1 Introduction

### 1.1.1 Motivation

The recent DARPA robotics challenge have shown the need for humanoid robots with an increased level of functionality enabled by proper control. Such complex robots must provide a simple interface for humans and handle as much as possible the motion generation autonomously. A general scheme is to use a motion planner to find an optimal path over a discrete set of foot-step transitions between two quasi-static poses [Chestnutt 2010, Hornung 2012]. The foot-steps transition are given by a statistical exploration of a whole-body controller together with a walking pattern generator. The planner then finds a feasible sequence of quasi-static poses and foot-step transitions which minimizes a cost function and avoids obstacles. This solution is then improved online while ensuring feasibility, see for instance [Perrin 2012]. In general it is not possible to realize real-time motion planning by directly using the controller itself because it is not possible to run more than one or two instances of the same controller before collision. Therefore, when the planner fails it is necessary to solve a continuous local problem which will provide a feasible solution different from the precomputed one [Chestnutt 2010]. The statistical exploration can be advantageously used to cast an optimization problem to find an initial guess [Chestnutt 2010]. Recently, *Deits* proposed to define the area of convergence for a local convex problem with linear constraints [Deits 2014] for a template model. With template models the inertia related to the whole-body motion is ignored, regulated to zero or corrected. In this chapter it is corrected by means of a dynamic filter. It is shown in the experimental section that it is drastically improving the performances over [Herdt 2010a] on the same robot. The use of template model is a practical solution on platforms with limited computation capabilities. Even if advanced whole-body motion controllers are now closer to real-time feasibility, e.g. the one proposed by *Todorov* which was recently applied to HRP-2 [Koenemann 2015], they still need powerful multi-core CPUs which limit their integration on humanoid robots due to heat and power consumption.

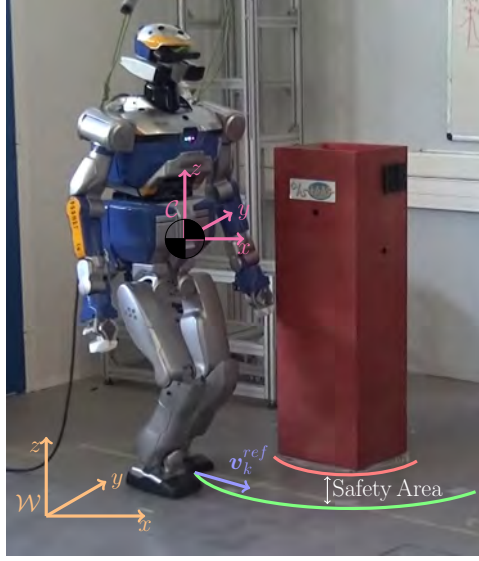


Figure 1.1: HRP-2 avoiding reactively on an obstacle, even if the reference velocity  $v_k^{ref}$  drives it into it. The upper body geometry is taken into account by setting a constraint (in green) such that the robot is sufficiently away from the obstacle (in red).

Another improvement of this chapter over the method developed in [Herdt 2010a] is the nonlinear formulation which here allows to deal with obstacle. More precisely [Herdt 2010a] integrates the information provided by statistical exploration of the controller feasibility between two foot-step transitions. It makes possible to correct foot-steps while having a guarantee over their feasibility. It is realized by reformulating the optimization problem to generate balanced Center-of-Pressure (CoP) and Center-of-Mass (CoM) trajectories where the free variables are the jerk of the CoM as well as foot-step positions and orientations. The feasible foot-steps, i.e. free of self-collision and singularities, are specified through linear constraints. This works well for level ground walking, unfortunately integrating obstacles with linear constraints implies a pre-processing of the environment or to use a different solver.

The present chapter shows that obstacles can be dealt with in real-time using a nonlinear scheme. Although not demonstrated in this chapter, it can be coupled with a real-time planner. The proposed method would provide a local feasible solution while the planner is looking for a global feasible solution [Perrin 2012].

### 1.1.2 Related work

Previous works have proposed to apply Model Predictive Control (MPC) to humanoid robots walking by considering either the whole body or a template model. When a model is available for a robot, MPC has several advantages. It can be very fast when using analytical solutions [Morisawa 2007, Tedrake 2015, S. Faraji 2014]. However such formulation makes generally some specific assumptions to find the derivation. This makes difficult the extension to other walking functionalities. On the other hand MPC schemes formulated as an optimization problem with a finer discretization grid can be more easily modified to include various walking modes inside a single formulation [Sherikov 2014]. In addition MPC as an optimization problem is becoming increasingly popular [Feng 2013], because for a given class of problems it allows using efficient



off-the-shelf solvers. Moreover several methods exist to increase the efficiency of solvers for NMPC problems. For instance, it is possible to use warm-starts or use a sub-optimal solution while maintaining feasibility [Boyd 2004]. The goal in humanoid robotics is to find a problem formulation which realizes all the needed functionalities and copes with the robot capabilities. The locomotion problems described in [Mordatch 2012], that include multi-contacts and consider the whole robot model over a time horizon, are not yet solvable in real-time and strongly depend on the models used to represent the physics.

Despite numerous efforts to address this large scale nonlinear problem with roughly ten thousand variables [Koenemann 2015, Dai 2014b], no solution yet exists to generate physically consistent controls in real-time using humanoid robot embedded computers. On the other hand template models projecting the overall robot dynamics to its CoM are used in research works [Wensing 2014, Orin 2012, Kajita 2003a, Engelsberger 2015], and already showing promising performance. Motion generation with template models can sometimes be solved analytically, and in such cases provide fast solutions that are particularly well suited for platforms with limited computational power. However, when increased CPU power is available, MPC-based solutions with the whole model are much more complete and reliable. Furthermore, as they can be easily modified, they provide more adaptive functionalities. In this chapter, with a bottom-up approach, we are trying to increase the functional level of a control architecture that already works on an existing humanoid robot, HRP-2 [Herdt 2010a]. The point of this chapter is to present extensions of the linear MPC scheme presented in [Herdt 2010a], that allows automatic foot placement in real-time. For instance, the problem depicted in Fig. 1.1 shows the humanoid robot HRP-2 driven by a desired velocity provided by the user. The former scheme [Herdt 2010a] was specifically formulated as a cascade of two quadratic programs (QPs). Foot-step orientations are solution of the first problem, while the second solution of the second QP provides the CoM trajectory and foot-step positions. This separation is efficient because the constraints are linear. If an obstacle has to be taken into account then the constraints have the shape depicted in Fig. 1.2, which is not convex anymore. To maintain the convexity, the solution would be to pre-process the obstacle and the feasibility area of the foot-steps. However a linearization of the obstacle boundary is equivalent to adding a linear constraint as depicted in Fig. 1.2. The algorithm proposed in this chapter is doing a similar operation and therefore no pre-processing is necessary. This is one of the major contribution of this chapter in comparison to [Herdt 2010a]. The proposed nonlinear extension takes into account the exact expression of constraints such as, for instance, locally avoiding a convex obstacle. Other formulations for walking motion generation have already been proposed. [Deits 2014] is using mixed-integer convex optimization for planning foot-steps with Atlas. [Ibanez 2014] is using mixed-integer convex optimization for MPC control and foot steps timing. In this work we introduce three nonlinear inequalities to handle balance, foot step orientation and obstacle avoidance. This new real time walking pattern generator has been successfully tested on the humanoid robot HRP-2 as depicted in Fig. 1.1. A key ingredient for achieving real-time performance was the following observation: *one real-time iteration of the nonlinear scheme is enough to find a reasonable solution.*

### 1.1.3 Contribution of the chapter

- It proposes a nonlinear reformulation of classical walking pattern generator able to find simultaneously foot-step positions and orientations.

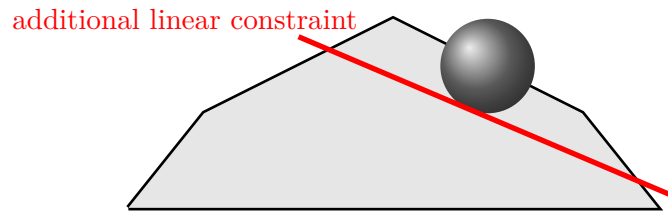


Figure 1.2: Walkable zone distorted by a convex obstacle

- It introduces nonlinear constraints able to cope with obstacles in the environment.
- It shows experimentally that one iteration of the nonlinear iterative scheme provides a suboptimal but sufficient solution for practical cases.
- Thanks to the use of a dynamical filter that corrects the CoM trajectory to compensate the limitation of the template model as in [Nishiwaki 2009] the whole body dynamics can be taken into account. This technical implementation has a strong impact on the robot performances.
- The whole algorithm runs in real time on the embedded hardware of the human-size humanoid robot HRP-2.

To be completely fair we are not doing NMPC using sensor feedback on the walking pattern generator. The feedback loop of the algorithm is done by the dynamic filter (see Fig. 1.3). In fact the sensor feedback is already done by the commercial stabilizer from Kawada Industry. Additional ongoing work is done to close the loop but the stabilizer is a closed-source software. Hence it is rather difficult to be exactly sure of its behavior.

## 1.2 Derivation of the dynamics

In this work the well known Linear Inverted Pendulum Model (LIPM) from [Kajita 2003a] is used as the template model of the robot's dynamics and the following assumptions are made: 1) the angular momentum produced by the rotations of all the robot parts is supposed to be zero, 2) the robot CoM evolves on a horizontal plane, 3) the normals of the contact forces have to be collinear. As a consequence, each quantity can be expressed as a function of three degrees of freedom (DoFs), which are the projection of the robot CoM  $(x, y)$ -position on the ground plane and its free-flyer orientation  $\theta$  around the vertical axis  $z$ . The reader is kindly referred to [Herdt 2010a] for a detailed description of the several terms omitted for a sake of clarity in the following section.

### 1.2.1 Discretization of CoM dynamics

In order to obtain a smooth trajectory, one controls the robot CoM through its jerk  $\ddot{c}^\nu$  on a preview horizon, where  $c$  denotes the position of the CoM in the world frame and  $\nu \in \{x, y\}$  is used to simplify the notation. This is done by applying a constant sampling period  $T$  and by assuming a piecewise constant jerk on each interval, i.e.  $\ddot{c}_k^\nu(t) \equiv \text{constant}$ ,  $t \in [kT, (k+1)T]$ ,  $k \in \{0, 1, \dots, N\}$ , where  $N$  is the length of the preview horizon.

The following time-stepping scheme maps the current state of the frame  $c_k^\nu$  to the future states by

$$\hat{c}_{k+j}^\nu = A^j \hat{c}_k^\nu + \sum_{i=0}^{j-1} A^i B \ddot{c}_{k+i}^\nu, \quad j \in [0, N], \quad (1.1)$$

$$\hat{c}_k^\nu = \begin{bmatrix} c_k^\nu \\ \dot{c}_k^\nu \\ \ddot{c}_k^\nu \end{bmatrix}, \quad A = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix}. \quad (1.2)$$

To express the CoM over the preview horizon the vector  $C_{k+1}^\nu$  of size  $\mathcal{R}^N$  and its derivatives are defined as

$$\begin{aligned} C_{k+1}^\nu &= \begin{bmatrix} c_{k+1}^\nu & \cdots & c_{k+N}^\nu \end{bmatrix}^T, & \dot{C}_{k+1}^\nu &= \begin{bmatrix} \dot{c}_{k+1}^\nu & \cdots & \dot{c}_{k+N}^\nu \end{bmatrix}^T, \\ \ddot{C}_{k+1}^\nu &= \begin{bmatrix} \ddot{c}_{k+1}^\nu & \cdots & \ddot{c}_{k+N}^\nu \end{bmatrix}^T. \end{aligned}$$

Using eq. (1.1), the above vectors can be expressed as a function of the initial state  $\hat{c}_k^\nu$  and the CoM jerk  $\ddot{C}_{k+1}^\nu$ . The latter belongs to the free-variable vector of the optimization problem described in section 1.3.

### 1.2.2 Linear inverted pendulum dynamics

In this chapter the balance criteria used is the one that have the center of pressure (CoP) in the convex hull of the robot's support polygon, which is defined by the contacts with the ground [Kajita 2003a] (see Sec. 1.3.3). Hence, the CoP has to be expressed in terms of the system's free variables, i.e. the CoM jerk. Using the assumptions made in the introduction of Sec. 1.2, the robot CoP can be expressed as a linear function of the CoM, i.e.

$$z_{k+n}^\nu = \begin{bmatrix} 1 & 0 & -h/g \end{bmatrix} \hat{c}_{k+n}^\nu, \quad \nu \in \{x, y\}, n \in [0, N-1],$$

with  $h = c^z - z^z$  being the height of the CoM with respect to the ground and  $g$  the norm of the gravity vector. Using eq. (1.1), a recursive expression for the future evolution of the CoP for a fixed horizon of  $N$  sampling steps is given by

$$z_{k+n}^\nu = \begin{bmatrix} 1 & 0 & -h/g \end{bmatrix} \left[ A^n \hat{c}_k^\nu + \sum_{i=0}^{n-1} A^i B \ddot{c}_{k+i}^\nu \right]. \quad (1.3)$$

As in Sec. 1.2.1, the vector  $Z_{k+1}^\nu = \begin{bmatrix} z_{k+1}^\nu & \cdots & z_{k+N}^\nu \end{bmatrix}^T$ , of size  $\mathcal{R}^N$ , is used to describe the CoP on the preview horizon. This vector can then be expressed in terms of  $\hat{c}_k^\nu$  and  $\ddot{C}_{k+1}^\nu$ .

### 1.2.3 Automatic foot step placement

The adaptive placement of the feet, with the aim to ensure balance of the robot even under external perturbations, is a key-feature of the algorithm. To this end, consider a frame  $\mathcal{F}$  attached to the support foot, with its current position and orientation on the ground given by  $f_k^\eta$ , with  $\eta \in \{x, y, \theta\}$ . The future steps, also free variable of the optimization problem, are

denoted by

$$\begin{aligned} F_{k+1}^\eta &= \begin{bmatrix} f_{k+1}^\eta & f_{k+2}^\eta & \cdots & f_{k+N}^\eta \end{bmatrix}^T \\ F_{k+1}^\eta &= v_{k+1} f_k^\eta + V_{k+1} \tilde{F}_{k+1}^\eta \end{aligned} \quad (1.4)$$

with  $F_{k+1}^\eta$  of size  $\mathcal{R}^N$  representing the foot support position at each time step and  $\tilde{F}_{k+1}^\eta$  of size  $\mathcal{R}^{nf}$  the actual free variables of the problem. The vector  $v_{k+1} \in \mathbb{R}^N$  and matrix  $V_{k+1} \in \mathbb{R}^{N \times nf}$  indicate which step falls in which sampling interval (see [Herdt 2010a] for more details). Sampling times correspond to rows, steps to columns, and  $nf$  is the maximum number of double support phases in the preview.

In theory, the usage of a single point mass as model prevents the definition of an orientation. In [Herdt 2010a] a frame attached to the center mass is defined and the orientation of this frame and the feet directions are optimized. In this work only the foot step orientations are optimized, and the orientation of the robot free-flyer is computed from this solution. Let  $\text{ff}^\theta(t)$ ,  $f^{\theta,L}(t)$  and  $f^{\theta,R}(t)$  be respectively the orientation of the free-flyer, the left foot and the right foot at any time  $t$ . Hence  $\text{ff}^\theta(t)$  is by convention :

$$\begin{bmatrix} \text{ff}^\theta(t) \\ \dot{\text{ff}}^\theta(t) \\ \ddot{\text{ff}}^\theta(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(f^{\theta,L}(t) + f^{\theta,R}(t)) \\ \frac{1}{2}(\dot{f}^{\theta,L}(t) + \dot{f}^{\theta,R}(t)) \\ \frac{1}{2}(\ddot{f}^{\theta,L}(t) + \ddot{f}^{\theta,R}(t)) \end{bmatrix}.$$

### 1.3 Nonlinear Model Predictive Control

Solving the orientation problem separately from the position problem is a workaround to linearize the CoP (eq. (1.13)) and foot position (eq. (1.15)) constraints derived below. However, computing separately the orientation and then injecting the solution into the position QP amounts to solve a different problem than the nonlinear combination of both. In the following the nonlinear problem will be derived and analyzed, and an appropriate approach allowing the real-time execution of the algorithm on the robot will be proposed.

#### 1.3.1 The controller

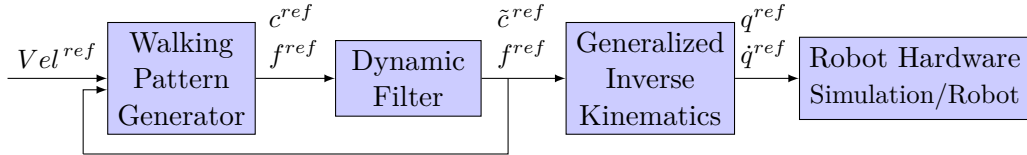


Figure 1.3: The control scheme:  $Vel^{ref}$  is the input velocity.  $c^{ref}$  and  $f^{ref}$  are respectively the CoM and the feet 3D trajectories  $\tilde{c}^{ref}$  is the CoM trajectory filtered.  $q^{ref}$ ,  $\dot{q}^{ref}$  denote respectively the generalized coordinate vector and its derivation.

A scheme of the controller is shown in Fig. 1.3. This open-loop controller is used for tracking respectively a referenced linear and angular velocity. In the first step, the walking pattern generator (WPG) computes the foot steps and CoM jerk from the given velocity

$Vel_{k+1}^{ref} = \begin{bmatrix} Vel_{k+1}^{x,ref} & Vel_{k+1}^{y,ref} & Vel_{k+1}^{\theta,ref} \end{bmatrix}$ . Then it uses an Euler integration scheme to compute the CoM trajectory from its jerk and polynomials of fifth order to retrieve 3D trajectories for the feet from the foot step planning. The CoM computed by the WPG is then filtered (see Sec. 1.4) and sent altogether with the feet trajectory to a generalized inverse kinematics algorithm. The output is a whole-body walking trajectory that can be applied directly on the robot. The WPG is then reinitialized with the current reference velocity input and with the corrected initial states of the dynamic filter.

### 1.3.2 The cost function

The cost function used in the NMPC is given by

$$\min_{U_k} \frac{\alpha}{2} J_1(U_k) + \frac{\alpha}{2} J_2(U_k) + \frac{\beta}{2} J_3(U_k) + \frac{\gamma}{2} J_4(U_k) \quad (1.5)$$

with  $\alpha$ ,  $\beta$  and  $\gamma$  being the weights of the cost function and  $U_k$  the free variables of the problem defined as

$$U_k^{x,y} = \begin{bmatrix} \ddot{C}_k^x & \tilde{F}_k^x & \ddot{C}_k^y & \tilde{F}_k^y \end{bmatrix}^T, \quad U_k^\theta = \tilde{F}_k^\theta, \quad U_k = \begin{bmatrix} U_k^{x,y} & U_k^\theta \end{bmatrix}^T. \quad (1.6)$$

$J_1(U_k)$  is the cost function related to the linear velocity tracking

$$J_1(U_k) = \|\dot{C}_{k+1}^x - Vel_{k+1}^{x,ref}\|_2^2 + \|\dot{C}_{k+1}^y - Vel_{k+1}^{y,ref}\|_2^2.$$

$J_2(U_k)$  is the cost function related to the angular velocity tracking

$$J_2(U_k) = \|F_{k+1}^\theta - \int Vel_{k+1}^{\theta,ref} dt\|_2^2.$$

Experiments have shown that a different weight between linear and angular velocities was not necessary at this stage.  $J_3(U_k)$  is the cost function minimizing the distance between the CoP and the projection of the ankle on the sole

$$J_3(U_k) = \|F_{k+1}^x - CoP_{k+1}^x\|_2^2 + \|F_{k+1}^y - CoP_{k+1}^y\|_2^2. \quad (1.7)$$

$J_4(U_k)$  is the cost function minimizing the norm of the control

$$J_4(U_k) = \|\ddot{C}_{k+1}^x\|_2^2 + \|\ddot{C}_{k+1}^y\|_2^2.$$

The above minimization function can then be express in a canonical form

$$\begin{aligned} \min_{U_k} \quad & \frac{1}{2} U_k^T Q_k U_k + p_k^T U_k, \\ \text{with } Q_k = & \begin{bmatrix} Q_k^{x,y} & 0 \\ 0 & Q_k^\theta \end{bmatrix}, \quad p_k = \begin{bmatrix} p_k^{x,y} \\ p_k^\theta \end{bmatrix}, \quad Q_k^\theta = \alpha \mathbb{I}_{nf}, \\ p_k^\theta = & \alpha \left( \begin{bmatrix} 1 & \dots & nf \end{bmatrix} T_{step} Vel_{k+1}^{\theta,ref} + \begin{bmatrix} 1 \\ \vdots \\ f_k^\theta \\ 1 \end{bmatrix} \right). \end{aligned} \quad (1.8)$$

The reader is kindly referred to [Herdt 2010a] for the defintion of  $Q_k^{x,y}$  and  $p_k^{x,y}$ . The matrix  $Q_k^\theta$  and  $p_k^\theta$  are derived because we use a slightly different method than [Herdt 2010a] to deal with the orientation.

### 1.3.3 The constraints

First of all the balance of the robot has to be ensured, then the feasibility of the foot step needs to be verified. Finally, the nonlinear constraint which implements the obstacle avoidance is described. It is one of the contribution introduced by this chapter. The following exposition is based on [Herdt 2010a].

#### 1.3.3.1 Balance constraint

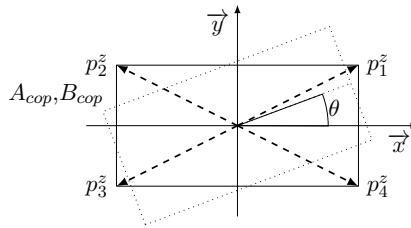


Figure 1.4: Shape of the foot with the position vector  $p_i^z$  describing the support polygon and  $\theta$  representing its orientation. The  $4 \times 2$  matrix  $A_{cop}$  and the  $4 \times 1$  vector  $B_{cop}$  are the linear algebra representation of the edges.

The CoP has to remain inside the support polygon [Wieber 2002]. This polygon is depicted in Fig. 1.4. The set of linear inequalities representing the convex polygon is denoted as  $A_{cop}$  and  $B_{cop}$ . Only one foot is modeled as a support polygon for two reasons: 1) HRP-2 feet are symmetrical, 2) the sampling period of the problem is designed in a way that no iteration of the optimization problem falls into a double support phase. The CoP at instant  $k$ , ( $z_k = [z_k^x \ z_k^y]^T$ ),

see Sec. 1.2.2) lies inside the support polygon if and only if

$$A_{cop} R(f_k^\theta) (z_k - f_k) \leq B_{cop} \quad (1.9)$$

$$\begin{bmatrix} A_{cop,k}^{x,\theta} & A_{cop,k}^{y,\theta} \end{bmatrix} (z_k - f_k) \leq B_{cop} \quad (1.10)$$

$$R(f_k^\theta) = \begin{bmatrix} \cos(f_k^\theta) & \sin(f_k^\theta) \\ -\sin(f_k^\theta) & \cos(f_k^\theta) \end{bmatrix}, \quad (1.11)$$

where  $f_k = [f_k^x \ f_k^y]^T$ ,  $A_{cop,k}^{x,\theta}$  is the left column of  $A_{cop} R(f_k^\theta)$  and  $A_{cop,k}^{y,\theta}$  is the right one. Using eq. (1.4) the constraint for each time step of the preview horizon is defined by

$$D_{k+1}(U_k^\theta) \begin{bmatrix} Z_{k+1}^x - v_{k+1} f_k^x - V_{k+1} \tilde{F}_{k+1}^x \\ Z_{k+1}^y - v_{k+1} f_k^y - V_{k+1} \tilde{F}_{k+1}^y \end{bmatrix} \leq b_{cop \ k+1} \quad (1.12)$$

With  $b_{cop \ k+1} = [B_{cop} \dots B_{cop}]^T$  and  $D_{k+1}(U_k^\theta) =$

$$\begin{bmatrix} A_{cop,k+1}^{x,\theta} & & 0 & A_{cop,k+1}^{y,\theta} & & 0 \\ & \ddots & & & \ddots & \\ 0 & & A_{cop,k+N}^{x,\theta} & & 0 & A_{cop,k+N}^{y,\theta} \end{bmatrix}.$$

From eq. (1.12), the canonical form of the constraint is

$$A_{cop,k}(U_k^\theta) U_k^{x,y} \leq \overline{U_{cop,k}}, \quad (1.13)$$

where  $A_{cop,k}(U_k^\theta)$  is a matrix depending on  $U_k^\theta$  which makes this constraint nonlinear. And  $\overline{U_{cop,k}}$  is the upper bound vector. The last steps of the derivation are detailed in [Herd 2010a].

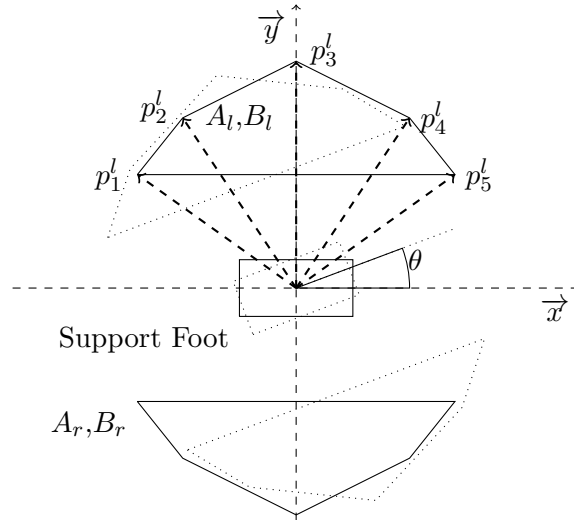


Figure 1.5: Shape of the selected convex polygon boundary of the foot placement. The  $5 \times 2$  matrix  $A_{r,l}$  and the  $5 \times 1$  vector  $B_{r,l}$ , define the convex hull as a set of linear inequalities.

### 1.3.3.2 Foot step feasibility constraint

This constraint uses the same convex hull as in [Herd 2010a] to ensure the feasibility of the steps [Perrin 2010]. For HRP-2 this convex hull is shown in Fig. 1.5. The set of linear inequalities representing this convex polygon is defined by  $A_{foot}$  and  $B_{foot}$ . Instead of  $r$  or  $l$  the lower index  $foot$  is used because the problem is symmetrical. The constraint, representing the fact that the swing foot has to land inside the convex hull, is given as

$$A_{foot}R(\theta)(f_{k+1} - f_k) \leq B_{r,l}. \quad (1.14)$$

In the exact manner as in eq. (1.13), the vector and matrices depicted in Sec. 1.2 are used to express this constraint for each previewed foot step. More details are presented in [Herd 2010a]. The canonical form of the constraint is

$$A_{foot,k}(U_k^\theta) U_k^{x,y} \leq \overline{U_{foot,k}}, \quad (1.15)$$

where  $A_{foot,k}(U_k^\theta)$  depends on  $U_k^\theta$  like  $A_{cop,k}(U_k^\theta)$ , which makes this constraint nonlinear. And  $\overline{U_{foot,k}}$  is the upper bound vector.

### 1.3.3.3 Foot orientation constraint

One additional feasibility constraint considers the maximum and minimum angle between both feet

$$-\theta_{thresh} \leq F_{k+1}^\theta - F_k^\theta \leq \theta_{thresh}, \quad (1.16)$$

with the canonical form

$$\begin{aligned} \underline{U_{\theta,k}} &\leq A_\theta U_k^\theta \leq \overline{U_{\theta,k}} \\ \text{with : } A_\theta &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & 0 \\ \ddots & 0 & -1 & 1 \end{bmatrix}, \\ \overline{U_{\theta,k}} &= \begin{bmatrix} \theta_{thresh} + f_k^\theta & \theta_{thresh} & \dots & \theta_{thresh} \end{bmatrix}^T, \\ \underline{U_{\theta,k}} &= \begin{bmatrix} -\theta_{thresh} + f_k^\theta & -\theta_{thresh} & \dots & -\theta_{thresh} \end{bmatrix}^T. \end{aligned} \quad (1.17)$$

In practice the bound  $\theta_{thresh} = 0.05rad$  takes into account the hardware limits. At this stage, the optimization problem allows the robot to place its feet anywhere inside the convex hull at any moment. In [Herd 2010a], the velocity of the foot is limited by bounding the feasible foot step area that corresponds to a maximum velocity. We chose to use the same idea extended to all the foot steps degrees of freedom. This significantly decreases the variation of accelerations before foot landing.



### 1.3.4 Additional constraint : local obstacle avoidance

Here, only the convex obstacles are considered. For simplification the obstacle is defined as a circle  $C = \{(p^x, p^y) \in \mathcal{R}^2, (p^x - x_0)^2 + (p^y - y_0)^2 = R^2\}$  Where  $x_0$  and  $y_0$  are its center coordinates in the world frame and  $R$  its radius. The previewed foot steps are feasible if they are outside the circle. This constraint does not depend on the orientation of the foot steps. For the  $j^{th}$  previewed step, at iteration  $k + j$  the constraint is expressed by

$$\left(f_{k+j}^x - x_0\right)^2 + \left(f_{k+j}^y - y_0\right)^2 \geq R^2 + m^2 \quad (1.18)$$

$$\iff U_k^T H_{obs,j} U_k + A_{obs,j} U_k \geq \underline{U_{obs,j}}, \quad (1.19)$$

with  $H_{obs,j}$  a selection matrix,  $A_{obs,j}$  a vector depending on  $x_0$  and  $y_0$ , and  $m$  a security margin taking into account the swept volume of the robot.

### 1.3.5 The solver

This paragraph presents the method used to solve the problem detailed in the previous sections. The non-linearity of the constraint and the still quadratic objective classifies the former LQR scheme as a nonlinear least squares optimization problem, which has the general form

$$\min_{U_k} \frac{1}{2} \|l(U_k)\|_2^2 \quad (1.20a)$$

$$\text{s.t. } \underline{h} \leq h(U_k) \leq \bar{h}. \quad (1.20b)$$

In general, derivative-based methods in the form of sequential quadratic programming SQP can be used for nonlinear optimization problems. These methods are called SQP because at each iteration a second order approximation of the nonlinear problem is calculated. Here, the least squares structure can be exploited to solve eq. (1.20) more efficiently using a generalized Gauß-Newton method. Starting with an initial guess  $U_{k-1}$  the method iterates  $U_k = U_{k-1} + \Delta U_k$ , where the increment  $\Delta U_k$  is obtained from the solution of the following QP approximation under the following form

$$\min_{\Delta U_k} \frac{1}{2} \|l_{k-1} + (\nabla_{U_k} l_k|_{U_{k-1}})^T \Delta U_k\|_2^2 \quad (1.21a)$$

$$\text{s.t. } \underline{h} - h_{k-1} \leq (\nabla_{U_k} h_k|_{U_{k-1}})^T \Delta U_k \leq \bar{h} - h_{k-1} \quad (1.21b)$$

with

$$l_k := l(U_k), \quad h_k := h(U_k).$$

Reformulating eq. (1.21) as a QP in canonical form, we get

$$\min_{\Delta U_k} \frac{1}{2} \Delta U_k^T \tilde{Q}_k \Delta U_k + \tilde{p}_k^T \Delta U_k \quad (1.22a)$$

$$\text{s.t. } \underline{\tilde{U}}_k \leq \tilde{A}_k \Delta U_k \leq \overline{\tilde{U}}_k \quad (1.22b)$$

with

$$\begin{aligned}
\tilde{Q}_k &= Q_k, \quad \tilde{p}_k = \begin{bmatrix} \frac{1}{2}(U_{k-1}^{x,y})^T Q_k^{x,y} + p_k^{x,y} \\ \frac{1}{2}(U_{k-1}^\theta)^T Q_k^\theta + p_k^\theta \end{bmatrix} \\
\tilde{A}_k &= \begin{bmatrix} A_{cop,k}(U_{k-1}^\theta) & \nabla_{U_k^\theta}^T A_{cop,k}|_{U_{k-1}^\theta} U_{k-1}^{x,y} \\ A_{foot,k}(U_{k-1}^\theta) & \nabla_{U_k^\theta}^T A_{foot,k}|_{U_{k-1}^\theta} U_{k-1}^{x,y} \\ 0 & A_\theta \\ H_{obs,j} U_{k-1} + A_{obs,j} & 0 \end{bmatrix}, \\
\tilde{U}_k &= \begin{bmatrix} -\infty \\ -\infty \\ \overline{U_{\theta,k}} \\ \underline{U_{obs,j}} \end{bmatrix} - h_{k-1}, \quad \tilde{\bar{U}}_k = \begin{bmatrix} \overline{U_{cop,k}} \\ \overline{U_{foot,k}} \\ \overline{U_{\theta,k}} \\ +\infty \end{bmatrix} - h_{k-1}, \\
h_{k-1} &= \begin{bmatrix} A_{cop,k}(U_{k-1}^\theta) U_{k-1}^{x,y} \\ A_{foot,k}(U_{k-1}^\theta) U_{k-1}^{x,y} \\ A_\theta U_{k-1}^\theta \\ U_{k-1}^T H_{obs,j} U_{k-1} + A_{obs,j} U_{k-1} \end{bmatrix}, \\
&\forall j \in 1, \dots, nf.
\end{aligned}$$

In this work the NMPC scheme is based on the idea of the so called "real-time iteration" [Bock 2007, Diehl 2002]. At each time instant of the control loop the nonlinear problem resolution requires the use of a SQP method. However by carefully initializing the applied SQP method and by preserving the state from the last iteration, the computational effort can be reduced to solving a single QP (one iteration of the respective SQP method) at each time. Furthermore, the computational process can be separated into three phases, two of which can be completed in advance without knowledge of the actual process state. In this way, the feedback delay can be drastically reduced. Therefore, instead of solving eq. (1.20) we recalculate its linearization once at each iteration of the control loop and solve a single QP eq. (1.22) in each iteration. This allows a real-time execution on the robot even for the proposed nonlinear formulation.

### 1.3.6 The line search

From experiments, we never found optimal solution of the linearized problem that did not satisfy the nonlinear problem constraints. However to decrease the time consumption of the algorithm we limited the maximum time and active set recalculation of the solver. In that case, the solver provides a sub-optimal solution which does not necessarily satisfy the constraint of the original problem. The classical approach, in that case, is to use a line search. It basically consists in finding a scalar  $s$  which scales the found solution:

$$U_k = U_{k-1} + s\Delta U_k \quad (1.23)$$

The idea is to choose  $s := 1$  for a start and verify the constraints using  $U_k$ . If the constraints are not verified we decrease  $s := 0.6s$  and we update  $U_k$  using 1.23. The algorithm check the

constraints using the updated  $U_k$  and decrease again  $s$  if needed. The algorithm end if  $s$  is too small or if the constraints are verified. This way we verify that the constraint are verified and that the solver does not take more than the allocated time to find a solution.

## 1.4 Dynamic Filter

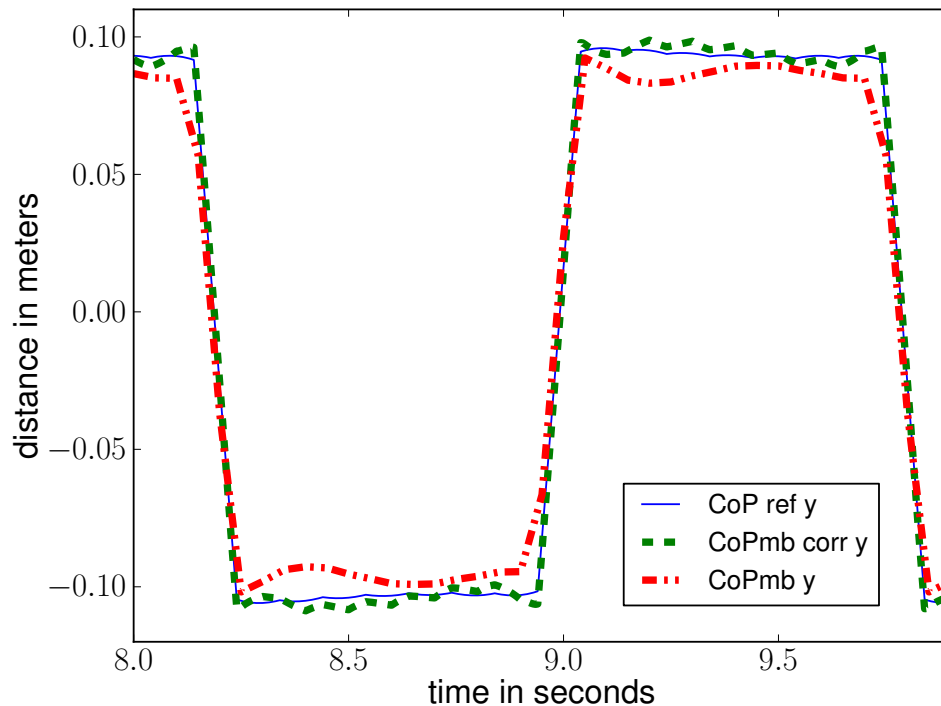


Figure 1.6: Result of the dynamic filtering on the CoP. In solid blue, the reference CoP computed by the solver. In dash-dot-dot red, the CoP multi-body. In dashed green, the CoP multi-body recomputed after correction.

Recall that the algorithm presented in this chapter and the one presented in [Herdt 2010a] assume that the inertial effect of the legs and the arms are neglected. An interesting fact is that the algorithm in [Herdt 2010a] that was successfully implemented on the HRP-2 in the Japan Robotic Laboratory (JRL), turned out to be unstable for its first test on another HRP-2 robot located at LAAS-CNRS. In order to cope with this difficulty, we used the dynamic filter introduced by Kajita [Kajita 2003a]. This filter aims to correct the difference between the referenced CoP computed by the pattern generator and the CoP reconstructed from the joint trajectories finally realized on the robot. In order to do so, a second model predictive control is used. This technique is often seen as applying a Newton Raphson method on the following equation  $z^{ref}(t) = RNEA(q(t))$ , where  $RNEA$  is the Recursive Newton-Euler Algorithm applied on the multi-body robot model. It computes the multi-body CoP from  $q$ , the generalized spatial state vector at time  $t$ . In general this method does not guarantee the convergence, and might suffer from numerical instability. However, it has proven its efficiency for this specific problem [Nishiwaki 2007]. Indeed in practice one iteration of the dynamic filter is sufficient to reduce

considerably the error on the CoP (see Fig. 1.6). More technical details about this algorithm are shown in Annexe A.

## 1.5 Experiments with HRP-2

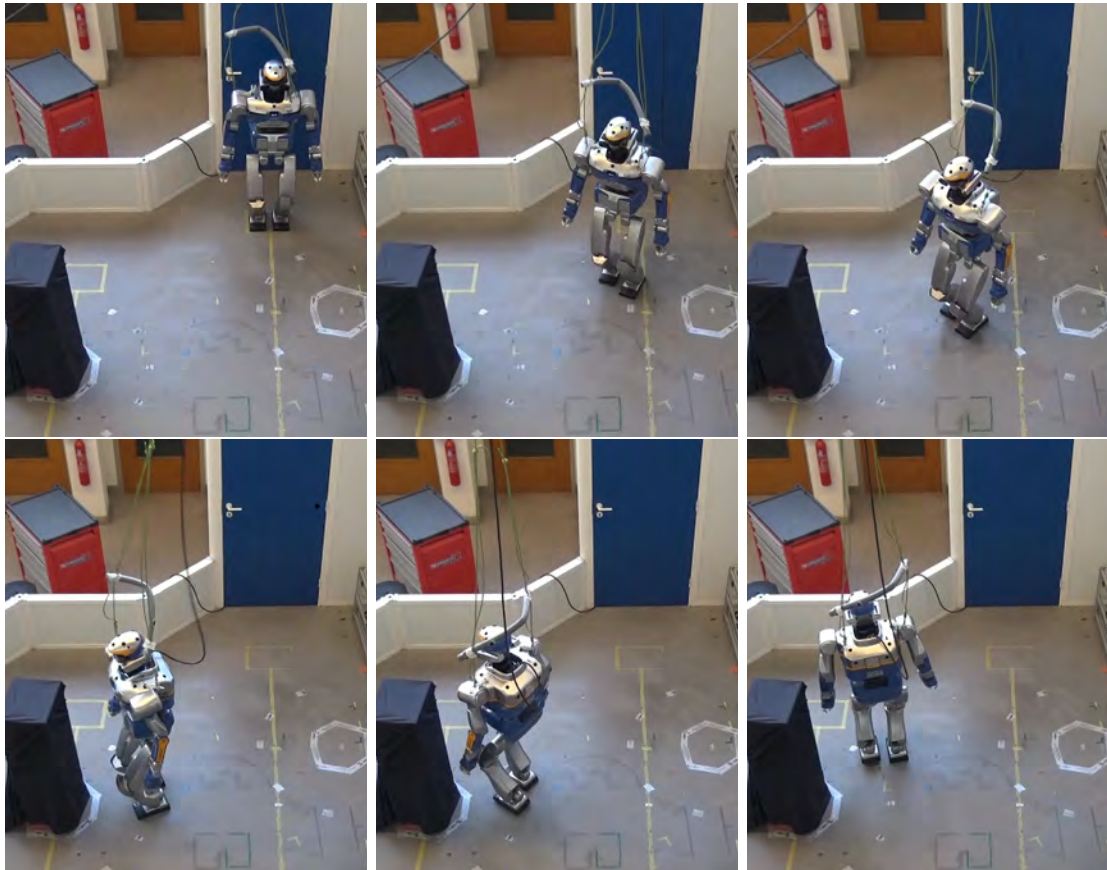


Figure 1.7: Experiment on the HRP-2 robot using the setup B.

In this section two experiments on the HRP-2 humanoid robot are presented. As described in the introduction they correspond to local situations where a foot-step planner using a discrete set of foot-step transitions may fail. We consider the case where only a reference velocity is given to drive the robot. It corresponds to a sensor-based behavior such as the one presented in [Garcia 2015]. The integration with a reactive planner such as the one presented [Perrin 2012] is left for future work. In the first experiment, the reference velocity drives the robot towards an obstacle which can be avoided thanks to the WPG. The second experiment shows the robot performing a circular trajectory and avoiding an obstacle.

### 1.5.1 Experimental setup

The duration of one full step is  $0.8\text{ s}$ , including single support ( $0.7\text{ s}$ ) and double support ( $T = 0.1\text{ s}$ ). During the experiment the preview horizon of the NMPC is two full steps, while

the preview horizon of the dynamic filter is equal to one full step in order to insure real time feasibility on HRP-2.

Fig. 1.8 depicts the two experimental setups. The upper figure and Fig. 1.1 show the output of the algorithm in the situation *A*. The forward velocity is set to  $Vel_{k+1}^{ref} = [0.2, 0, 0]$  and the obstacle to avoid is the red box. In Fig. 1.8 the box is represented by the inner red circle while the security margin is represented by the outer green circle. The robot is allowed to step on the green circle but not inside it. This margin prevents the upper body from colliding with the obstacle.

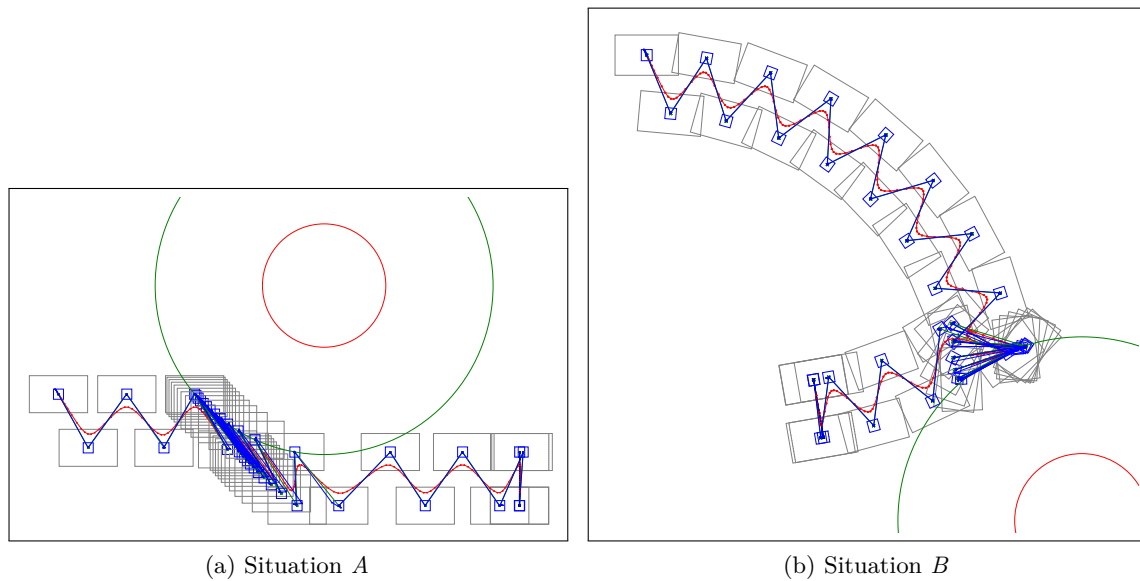


Figure 1.8: Center-of-mass and center-of-pressure trajectories for obstacle avoidance and foot-step orientation using NMPC. Situation *A* : Constant forward velocity. Situation *B* : Constant forward and angular velocity.

The setup *B*, depicted in Fig. 1.7 and 1.8 is quite similar. A constant velocity  $Vel_{k+1}^{ref} = [0.2, 0, 0.2]$  including rotation around the vertical axis is sent to the walking pattern generator. The robot starts to describe a circle and get stuck in front of the obstacle. As the constraint is locally linearized, and because the reference velocity in translation is going towards the constraint, the robot is blocked in translation. Thus, it stops moving forward and continues to turn on spot, as the angular velocity is not conflicting with the constraint. Once the robot has passed the obstacle, it can freely move forward and describe a circle again.

### 1.5.2 Robustness to perturbation

A disturbance test case has been performed in simulation. The disturbance is introduced as a force added to the CoM acceleration in the walking pattern generator. This force is applied during  $100\text{ ms}$ . Two kind of disturbances were considered: on the sagittal plane (both directions) and on the coronal plane (both directions). In both cases, we considered two walking situations: forward and on spot.

On the coronal plane, the maximum lateral force that can be handled is  $90\text{ N}$ , equivalent to  $-0.63\text{ J}$ , and  $-45\text{ N}$ , equivalent to  $0.675\text{ J}$ . The asymmetry comes from the fact that the robot might be in a different walking situation during the push. The push may occur when the robot can perform a step without collision, or when it cannot. In the latter case, the magnitude of the force that can be rejected is smaller. We found roughly the same values for the two walking situations.

When walking on spot, the maximum forward and backward perturbation is  $\pm 115\text{ N}$ , equivalent to  $\pm 0.86\text{ J}$ , as the problem is symmetrical. When walking forward, the maximum disturbance is smaller in the forward direction. The interval found is  $[-160; 70]\text{ N}$ , equivalent to  $[-1.12; 1.54]\text{ J}$ .

### 1.5.3 Computation time

This algorithm runs online on the HRP-2 CPU board (*Intel(R) Core2(TM) Duo E7500*, one core used,  $2.8\text{ GHz}$ ,  $3\text{ Mb}$  of cache size, on Ubuntu 10.04 LTS). So only counts the iteration when the NMPC is computed. Thus the statistics apply only when the walking pattern generator is computed. The time measurement has been performed on the complete control architecture (see Fig. 1.3).

<i>Time consumption</i>	experiment A	experiment B
Average ( <i>ms</i> )	3.95	4.00
Standard deviation ( <i>ms</i> )	0.14	0.18
Minimum ( <i>ms</i> )	3.34	3.085
Maximum ( <i>ms</i> )	4.34	5.19

The robot is controlled at a period of  $5\text{ ms}$ . Over all the experiences, there was only one iteration over  $5\text{ ms}$ . It is due to the stabilizer which consumes more CPU time when the robot is in a configuration leading to a kinematic singularity. The algorithm is still computed every  $100\text{ ms}$  to simplify the double support phase handling.

### 1.5.4 Cost function gains

The cost function gains are :  $\alpha = 2.5$ ,  $\beta = 10^3$  and  $\gamma = 10^{-5}$ . As specified in Sec. 1.3.2,  $\alpha$  is the reference tracking gain,  $\beta$  is a gain maintaining the CoP close to the center of the foot, and  $\gamma$  is the regularization gain. They were chosen according to their experimental performance. The chosen cost function gives different foot steps compared to [Deits 2014]. Where the minimization of a cost-to-go criteria as in [Deits 2014] was used, here the robot follows a velocity prescribed by the user and can differ from it locally to avoid an obstacle. This local method runs in real time at a lower level of control which copes with potential evolution of the environment after a first planning.

### 1.5.5 qpOASES solver

The nonlinear problem is linearized analytically (see Sec. 1.3.5) to form a quadratic problem with linear constraints. The off-the-shelf solver qpOases [Ferreau 2014] is used to solve the respective QP. This solver is a primal solver implementing an online active set strategy.

## 1.6 Conclusion

In this chapter we presented a real-time embedded nonlinear walking pattern generator. Nonlinear inequalities make possible to choose the foot step automatically while considering orientation and local avoidance of convex obstacles. Its performance was demonstrated in two different experiments using the humanoid robot HRP-2. The computational cost of the walking pattern generator is  $2\text{ ms}$  on the robot. An extension to our method would be to use a planner in addition to the walking pattern generator.





# Multicontact Locomotion

---



In this chapter we address the multi-contact locomotion problem with various number of contact and non planar one. This problem is important in humanoid robotics because it generalizes bipedal locomotion and thus expands the functional range of humanoid robots. The work done in this frame is the product of a collaboration under the Koroibot project and published in [Kudruss 2015]. In this chapter, we propose a complete solution to compute a fully dynamic multi-contact motion of a humanoid robot. We decompose the motion generation by first computing a dynamically consistent trajectory using only the centroidal dynamics. The second stage consist in finding the whole-body movement following this trajectory. A simplified dynamic model of the humanoid is used to find optimal contact forces as well as a kinematic feasible center-of-mass trajectory from a predefined series of contacts. We demonstrate the capabilities of the approach by making the real humanoid robot platform HRP-2 climb stairs with the use of a handrail. The experimental study also shows that the use of the handrail lowers the power consumption of the robot by **25%** compared to a motion where only the feet are used.

In Section 2.6, we present the previous work done on this problem. In Section 2.2, we first describe the formulation of the Optimal Control Problem (OCP) used to compute the CoM trajectory, and briefly discuss the particular numerical scheme used to solve this problem. In Section 2.3, we explain how to process the CoM trajectory and contact forces to create a feasible motion for the humanoid. Finally, we show the results we obtained from experiments on the real robot HRP-2 performing a stair climbing motion with hand rail support in section 2.4. Section 2.5 quickly presents a work done in collaboration with Justin Carpentier pursuing multi-contact locomotion related research using OCP and multiple shooting strategy.

## 2.1 Previous work

The generation of whole-body motions using multiple contacts between the robot and its environment extends the form of bipedal locomotion with a potential high impact on humanoid robot functionalities. It enables a robot to climb ladders, crawl, evolve in cluttered environment and less impressively, but yet very useful, climb stairs [Dai 2014a, Audren 2014]. Staircase climbing is an important basic behavior for humanoid robots aiming at evolving in an industrial environment (see Fig. 2.1). The DARPA Robotics challenge has illustrated the difficulty of this realization. Two main classes of approaches can be distinguished in the resolution of the multi-contact locomotion problem. On the one hand, the problem is approached all at once, by trying to find a complete trajectory of the system typically using a numerical resolution scheme. On the other hand, the problem is decomposed into several sub problems, typically by following the seminal approach used in ground-level bipedal locomotion.

Ideally, multi-contact motion generation includes a dynamic model of the humanoid, a model of its actuators and takes into account all its constraints over a finite preview-window. The number of degrees of freedom (DoFs) and the size of the needed preview-window make this approach useful for motions [Koch 2012] exploiting the whole robot, but its computational complexity prevents an execution in real-time.

The first class of approaches mostly tries to make the problem tractable by proposing various ways of approximating the complete problem or improving the mathematical properties arising from a specific formulation. [Dai 2014a] proposed to work on the under-actuated dynamics and consider only the constraints related to inverse kinematics. [Todorov 2014] proposed to

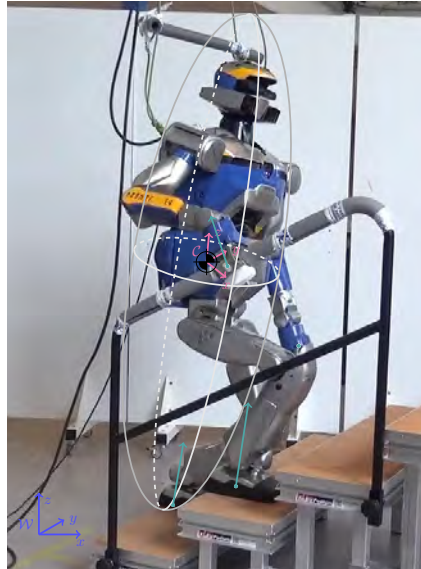


Figure 2.1: HRP-2 climbing stairs with the support of the handrail. The overlay shows the reduced model (center of mass (CoM, black) at waist, CoM frame (magenta), gray inertia ellipsoid, contacts (teal dots) and contact forces (teal arrows) as well as world coordinate system  $\mathcal{W}$  (blue).

reformulate the contact model, and use differential dynamic programming to solve the related optimization problem efficiently. In practice, the contact model is sufficient for transition with the robot hands for object reaching for example. But for walking, the contact model gives inconsistent data which results in unwanted or unbalanced behavior, like walking on the feet edges.

The authors of [Herzog 2015] implemented a planner for the linear and angular momentum taking into account multiple contacts. They used friction cones and solved the problem using SQP methods linearizing the dynamics. The tracking of the spatial momentum is done via an LQR providing the ankle wrench to a low level torque controller. This planner computes dynamically consistent multicontact trajectories but in around 5 min. Hence it is not suitable for real time motion generation.

Another approach is presented in [Hirukawa 2007], where the authors consider a set of contact points for which a dynamically consistent CoM trajectory is found. The forces are subject to the linearized friction cone constraints. This iterative algorithm assumes a predefined partitioning of the external forces applied to the system. In [Ott 2011] a stabilization process based upon the work of Cheng [Hyon 2007] is proposed. They propose to handle contact forces via the Coulomb friction cone typically used in grasping. They design a force based stabilizer that take into account the CoM dynamics and the joint position. It assumes a quasi-static joint motion, i.e. joint accelerations and velocity are assumed to be zero to map the external wrench to the joint torques. This condition imposes restrictions on the possible motions.

3D walking, i.e. stair walking or stepping stone, has already been investigated by the community. [Kajita 2003a] shown that simple CoP control can be used in simulation to climb stairs. [Naveau 2014] shown that the dynamic filter, i.e. the second stage of Kajita, allows to take into account the whole body dynamics while climbing the stairs. In [Englsberger 2013] the author proposed a 3D extension of the capture point. They formalized the non stable part

of the linear inverted pendulum dynamics as a 3D point. They show uneven ground walking and stair climbing motion controlling this point dynamics. All these algorithms plan the CoM trajectories for either predefined or online computed foot contacts. Extensive simplifications of the humanoid dynamic model result in a linear inverted pendulum model, which allows the fast generation of walking motions. The extensions proposed in [Nishiwaki 2012] can handle steep slopes of  $10^\circ$  and show the humanoid robot platform HRP-2 [Kaneko 2004] hitting obstacles and appropriately adapting its feet trajectory.

While these simplified models have a small computational footprint and show consistency with the elementary parts of the human gait, they all assume zero variation of the angular momentum about the CoM, which is once more a strong limitation for achieving complex dynamic movements. In practice having zero variation of the angular momentum about the CoM means that a parallel controller use the free DoFs to cancel this variation of the angular momentum impose by the controlled DoFs. Typically while walking the legs impose a momentum on the CoM and a swing motion of the robot arms can be used to cancel this momentum. These angular effects can be integrated directly inside the dynamic model based on the centroidal momentum described in [Orin 2013]. It consist in modeling the robot as rigid body with a constant inertia. Hence, the variation of angular momentum created by contacts between the robot and its environment on the CoM can be approximated. Thereby, the robot is model using the centroidal momentum.

### Contribution of the chapter

- It proposes a mathematical formulation of the reduced multi-contact CoM dynamics of a humanoid as an optimal control problem (OCP).
- It experimentally shows that the humanoid robot platform HRP-2 is able to climb stairs with a handrail support using this approach.
- It demonstrates from experimental study on the robot that handrail support reduces the motor power consumption by 25%.

The OCP is able to find feasible CoM trajectories and contact forces for predefined contact sets. The constraints are the contact model and the kinematic limits of the whole-body system. The template model includes the major effects on the under-actuated part and is applicable for any combination of contact (ground level, biped walk on non-flat floor, multi-contact like using the handrail during stair climbing, etc). In collaboration with Steve Tonneau, we computed the contact sequences using his multicontact planner presented in [Tonneau 2015a]. The following scheme Fig. 2.2 present the architecture of the controller.

## 2.2 Generation of the center-of-mass trajectory

### 2.2.1 Dynamic model and constraints

#### Inertia ellipsoid model

The template model is composed by the number of considered contact points  $M \in \mathbb{N}$  located at positions  $\mathbf{p}_i \in \mathbb{R}^3$ , for  $i = 1 \dots M$ . The force applied at  $\mathbf{p}_i$ , denoted by  $\mathbf{f}_i = [f_{i,x} \ f_{i,y} \ f_{i,z}]^\top$ , is

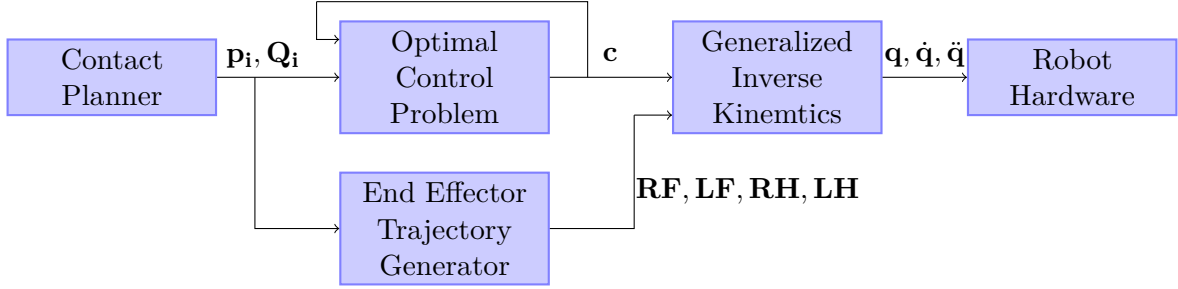


Figure 2.2: Overview of the joint trajectories ( $\mathbf{q}$ ) generation process, with  $\mathbf{p}_i, \mathbf{Q}_i$  the contact positions and orientations,  $\mathbf{c}$  the CoM trajectory, and  $\mathbf{RF}, \mathbf{LF}, \mathbf{RH}, \mathbf{LH}$  the end-effector trajectories.

given in a local coordinate system, with the  $z$ -axis normal to the contact surface at  $\mathbf{p}_i$ . The component  $f_{i,z}$  is the normal force applied at contact point  $\mathbf{p}_i$  and  $(f_{i,x}, f_{i,y})$  is the tangential force applied at  $\mathbf{p}_i$ . In the context of the stair climbing motion,  $M = 3$ , and the involved robot limbs are the right hand  $p_{rh}$ , the left foot  $p_{lf}$  and right foot  $p_{rf}$ .

Following the ideas from [Orin 2013, Nishiwaki 2012], we only consider the effect of the whole-body dynamics of a humanoid robot on the CoM. The centroidal dynamics is described as

$$m \begin{bmatrix} \ddot{\mathbf{c}} - \mathbf{g} \\ \mathbf{c} \times (\ddot{\mathbf{c}} - \mathbf{g}) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \dot{\mathcal{L}} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^M \mathbf{Q}_i \mathbf{f}_i \\ \sum_{i=1}^M \mathbf{p}_i \times \mathbf{Q}_i \mathbf{f}_i \end{bmatrix} \quad (2.1)$$

where  $\mathbf{c} = [c_x \ c_y \ c_z]^\top \in \mathbb{R}^3$  is the CoM and  $m \in \mathbb{R}$  is the mass of the humanoid multi-body system,  $\ddot{\mathbf{c}} = [\ddot{c}_x \ \ddot{c}_y \ \ddot{c}_z]^\top \in \mathbb{R}^3$  is the CoM acceleration,  $\dot{\mathcal{L}}$  is the derivative of the robot angular momentum due to the angular speed of the robot body parts, and  $\mathbf{Q}_i \in SO(3)$  is the  $3 \times 3$  rotation matrix that transforms forces  $\mathbf{f}_i$  in the local contact coordinate system at  $\mathbf{p}_i$  into the global world coordinate system.

In the following, the contribution to the variation of the angular momentum  $\mathcal{L}$  is separated between the part caused by the general angular acceleration of the robot and the internal robot body movements  $\dot{\mathcal{L}} = \mathbf{I}_c \dot{\omega} - \sigma$ , where  $\mathbf{I}_c$  is the inertia matrix of the whole body considered as a single rigid body computed for the configuration at time  $t_0$ ,  $\dot{\omega}$  is the angular acceleration (3D) of the root frame attached to the waist body of the robot, and  $\sigma$  is a function of the whole-body configuration, velocity and acceleration [Orin 2013] that does not depend on  $\omega$ . Eq. 2.1 is then rewritten in form of

$$\mathbf{M}(\mathbf{c}) \begin{bmatrix} \ddot{\mathbf{c}} \\ \dot{\omega} \end{bmatrix} = \sum_i \mathbf{J}_i^T \mathbf{f}_i + \mathbf{S}^T \sigma + \mathbf{b}(\mathbf{c}), \quad (2.2)$$

with

$$\mathbf{M}(\mathbf{c}) = \begin{bmatrix} m\mathbf{1}_3 & \mathbf{0}_{3 \times 3} \\ m\mathbf{c} \times & \mathbf{I}_c \end{bmatrix}, \quad \mathbf{J}_i^T = \begin{bmatrix} \mathbf{Q}_i \\ \mathbf{p}_i \times \mathbf{Q}_i \end{bmatrix}, \quad (2.3)$$

$$\mathbf{S}^T = \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{1}_3 \end{bmatrix}, \quad \mathbf{b}(\mathbf{c}) = \begin{bmatrix} m \mathbf{g} \\ m\mathbf{c} \times \mathbf{g} \end{bmatrix}, \quad (2.4)$$

where  $\mathbf{c} \times \in \mathbb{R}^{3 \times 3}$  (resp.  $\mathbf{p}_i \times$ ) is the skew matrix associated with vector  $\mathbf{c}$  (resp.  $\mathbf{p}_i$ ).

The equilibrium constraints (2.2) are rewritten according to [Boyd 2007] as non-linear first-order differential equation in form of

$$\frac{d}{dt} \begin{bmatrix} \mathbf{c} \\ \theta \\ \dot{\mathbf{c}} \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{c}} \\ \omega \\ (\mathbf{M}(\mathbf{c}))^{-1} \left( \sum_i \mathbf{J}_i^T \mathbf{f}_i + \mathbf{S}^T \sigma + \mathbf{b}(\mathbf{c}) \right) \end{bmatrix}. \quad (2.5)$$

### Linear constraints

The applied contact model treats the feet contacts as unilateral and the hand contact on the rail as bilateral. This is achieved by defining proper bounds onto the applied contact forces given by

$$\begin{bmatrix} -600 \text{ N} \\ -600 \text{ N} \\ 0 \text{ N} \end{bmatrix} \leq \mathbf{f}_{\text{lf/rf}} \leq \begin{bmatrix} 600 \text{ N} \\ 600 \text{ N} \\ 600 \text{ N} \end{bmatrix}, \quad \begin{bmatrix} -150 \text{ N} \\ -150 \text{ N} \\ -150 \text{ N} \end{bmatrix} \leq \mathbf{f}_{\text{rh}} \leq \begin{bmatrix} 150 \text{ N} \\ 150 \text{ N} \\ 150 \text{ N} \end{bmatrix}, \quad \begin{bmatrix} -600 \text{ Nm} \end{bmatrix} \leq \sigma \leq \begin{bmatrix} 600 \text{ Nm} \end{bmatrix}$$

The bounds are defined from empirical study. Contact forces can only be applied when a contact is established. When the contact is released (or moving), we define the lower and upper bounds for the contact force to be zero. The description of a moving contact is denoted by  $\|\dot{\mathbf{p}}_i\|_2 > 0$ . We refer to the contact complementarity to indicate if the end effector is in contact or not given by

$$\|\dot{\mathbf{p}}_i\|_2 \cdot \|\mathbf{f}_i\|_2 = 0 \quad (2.6)$$

However, note that the complementarity is not explicitly treated yet, but is predefined in the contact configuration.

### Friction cone constraints

The applied friction model requires the contact forces to satisfy the so called *friction cone constraints*, which are given for  $M$  contact points by

$$\|(f_{i,x}, f_{i,y})\|_2 = \sqrt{(f_{i,x})^2 + (f_{i,y})^2} \leq \mu_i f_{i,z}, \quad i = 1 \dots M, \quad (2.7)$$

where  $\mu_i > 0$  is the friction coefficient of the contact point  $\mathbf{p}_i$ . The friction cones  $K_1, \dots, K_M \subseteq \mathbb{R}^3$  can be defined as

$$K_i = \{\mathbf{x} \in \mathbb{R}^3 | x_1^2 + x_2^2 \leq (\mu_i x_3)^2, x_3 \geq 0\}, i = 1 \dots M,$$

and by following this notation, the friction cone constraints of (2.7) can be formulated as  $\mathbf{f}_i \in K_i, i = 1 \dots M$ .

### Kinematic constraints

For the kinematic feasibility, simplified constraints on the limb lengths relative to the CoM position in global coordinates, denoted by  $\mathbf{c} \in \mathbb{R}^3$ , are defined by

$$\underline{\mathbf{L}}_i \leq \|\mathbf{c} - \mathbf{p}_i\|_2 \leq \overline{\mathbf{L}}_i, i = 1, \dots, M. \quad (2.8)$$

We define the leg lengths for  $\mathbf{p}_{lf}, \mathbf{p}_{rf}$  using  $\underline{L}_{lf/rf} = 0.64m$  and  $\overline{L}_{lf/rf} = 0.8m$  for the stair climbing motion.

### 2.2.2 Objective function

Before giving the complete formulation of the optimal-control problem, we first define the cost terms used for the trajectory optimization over a given time interval. The first term  $\ell_0$  keeps the CoM close to the support foot contacts,

$$\ell_0 = \|f_{lf}\|_2^2 \cdot \|c^{x,y} - p_{lf}^{x,y}\|_2^2 + \|f_{rf}\|_2^2 \cdot \|c^{x,y} - p_{rf}^{x,y}\|_2^2.$$

The second term  $\ell_1$  uses the complementarity (2.6) to track a reference height depending on the current foot contact height,

$$\ell_1 = \|(\mathbf{f}_{lf,z} + \mathbf{f}_{rf,z})(\mathbf{c}_z - \mathbf{c}_z^{ref}) - \mathbf{f}_{lf,z}\mathbf{p}_{lf,z} - \mathbf{f}_{rf,z}\mathbf{p}_{rf,z}\|_2^2.$$

The four next terms  $\ell_2, \ell_3, \ell_4$  are used to penalize a swaying motion of  $\mathbf{c}$  in  $z$  direction and stabilize the rotational DoFs.

$$\ell_2 = \|\dot{\mathbf{c}}_z\|_2^2, \ell_3 = \|\omega_x\|_2^2, \ell_4 = \|\omega_y\|_2^2, \ell_5 = \|\omega_z\|_2^2.$$

The last term  $\ell_6$  acts as a regularization term,

$$\ell_6 = \|\ddot{\mathbf{c}}\|_2^2 + \|\dot{\omega}\|_2^2.$$

Table 2.1: Objective weights

$\omega_0$	$\omega_1$	$\omega_2$	$\omega_3$	$\omega_4$	$\omega_5$	$\omega_6$
0.05	0.0005	1.0	1.0	1.0	0.1	1.0



### 2.2.3 Optimal control formulation

We formulate an optimal control problem to search for the best CoM trajectory respecting the dynamics (2.5) and subject to the constraints from Section 2.2.1 in terms of a combination of different optimization criteria from Section 2.2.2.

#### Formulation

The variables of interest are the states and the controls over the time horizon. The state is composed of the CoM position and velocity and the angular velocity of the robot:  $\mathbf{x} := (\mathbf{c}, \dot{\mathbf{c}}, \theta, \omega)$ . The control is composed of the contact forces for the active contact at time  $t$  and the internal angular momentum variation:  $\mathbf{u} := (\mathbf{f}_1, \dots, \mathbf{f}_M, \sigma)$ .

The OCP minimizes an objective function of Lagrange type on a finite time horizon  $t \in [0, T]$  given by

$$\min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} \int_0^T l(\mathbf{x}(t), \mathbf{u}(t)) dt \quad (2.9a)$$

$$s.t. \quad \dot{\mathbf{x}}(t) = g(\mathbf{x}(t), \mathbf{u}(t)), \quad t \in [0, T], \quad (2.9b)$$

$$\mathbf{x}(0) = \mathbf{x}_0, \quad (2.9c)$$

$$0 \leq h(\mathbf{x}(t), \mathbf{u}(t)), \quad t \in [0, T]. \quad (2.9d)$$

where  $l(\mathbf{x}, \mathbf{u}) = \sum_j w_j \ell_j(\mathbf{x}, \mathbf{u})$  is the running cost, with a positive weight  $w_j \in \mathbb{R}$  from Table 2.1 adjusting the relative importance and scaling for each term  $\ell_j$ ;  $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$  is representing the dynamics of the system defined in (2.5);  $\mathbf{x}_0$  is the initial (measured) state of the system; and  $h(\mathbf{x}, \mathbf{u}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_c}$  are the mixed state-control path constraints defined by concatenating the friction cone (2.7), the kinematic constraints (2.8) as well as the complementarity constraints (2.6), where the latter is defined via the contact sequence and are not explicitly treated in the OCP. Note that  $\dot{\mathbf{p}}_i$  are not part of the decision variables of the OCP.

#### Discretization

Following a direct approach to optimal control, the control  $u(\cdot)$  is discretized on a time grid  $0 = t_0 < t_1 < \dots < t_K = T$  by means of base functions parametrized by the parameters  $\alpha$ . We use a piecewise linear discretization which yields smoother CoM trajectories, i.e. for  $k = 0, \dots, K-1$ :

$$u(t) \Big|_{[\tau_k, \tau_{k+1}]} := (\alpha_{k,1}(\tau_{k+1} - t) + \alpha_{k,2}(t - \tau_k)) / (\tau_{k+1} - \tau_k)$$

#### Multiple shooting resolution

Applying the direct multiple shooting approach for optimal control, we further parametrize the state trajectory  $\mathbf{x}(\cdot)$  by solving initial value problems for the differential equation (ODE) (2.9b) separately on the same grid used for the control discretization. The model dynamics (2.9b) are adaptively discretized by making use of state-of-the-art ODE solvers. Continuity of the trajectory in the solution of the OCP is enforced by constraints.

From this discretization, a large but structured nonlinear programming problem is obtained. This problem can be solved efficiently with a tailored structure-exploiting sequential quadratic

programming (SQP) method [Leineweber 2003]. First- and second-order derivatives, required by the SQP method to solve the discretized OCP, involve the computation of sensitivities of the ODE solution according to the principle of internal numerical differentiation (see [Bock 1981] for details).

## 2.3 Motion Generation

The computation of the trajectories, depicted in Fig. 2.2 and needed for the realization of the motion, is a five step process. (i) A set of contact points has to be found. (ii) The trajectories of the center of mass is computed by the *OCP*. (iii) The full trajectory of the end effectors are computed using B-splines. (iv) The generalized inverse kinematics algorithm is used to compute the actuated joint trajectories of the robot. (v) The robot perform the computed motion.

In the final process, a set of joint trajectories ( $\mathbf{q}$ ) is provided for the position-controlled humanoid robot HRP-2. Note that these trajectories are dynamically consistent and that the contacts are realized according to the predefined schedule of the OCP formulation. The change of angular momentum  $\sigma$  was of small effect for the stair climbing motion and was neglected. In cases where  $\sigma$  affects the motion, it can be rejected by methods of resolved momentum control using the free limbs during the execution of the motion.

### 2.3.1 Definition of the contact sequence

For each contact the time interval and the contact state, active or not, are defined. We also specify which body is in contact, the position of the contact, the friction coefficient, and the contact normal vector. The contact sequence is a collection of such contact specifications. Such a contact sequence is typically realized through a contact planner [Audren 2014, Tonneau 2015b]. An extension to this work would be to additionally optimize the position of the contacts on a given set of planar contact surfaces.

### 2.3.2 End-effector trajectories

In this chapter, the hand, the feet and the CoM trajectories are represented using B-splines. In the frame of the Koroibot project, human hand and foot trajectories are studied using inverse optimal control. In addition, meaningful optimization criteria minimized by the under-actuated part of the human dynamics are investigated and will be included in the cost function of problem (2.9a). At this stage of the project, this is still on-going work and the B-spline parameters are found using heuristics. Although not in the scope of this chapter, other approaches are possible.

### 2.3.3 Whole-body generation

The final whole-body motion is generated by applying the stack-of-tasks (SoT) scheme implementing a generalized inverse kinematics (GIK) as shown in [Mansard 2009]. Given the CoM, the root orientation and end-effector trajectories the SoT framework computes the complete trajectory of all the DoFs of the system. This is done by specifying tasks for the SoT. Those tasks are defined as a simple proportional derivative (PD) controller tracking the corresponding reference trajectory. The tasks are the following: a task tracking the CoM trajectory along the  $x, y, z$  axes ( $T_{Com}$ ), a task for each end-effector position and orientation specification ( $T_{RH,LH,RF,LF}$ ),

a task to control the orientation of the waist ( $T_W$ ), and a task regulating the posture of the robot around a nominal posture ( $T_q$ ). The hierarchy of the tasks defined as the lexicographic order  $T_{Com} \prec T_{RH,LH,RF,LF} \prec T_W \prec T_q$ . The dynamical consistency of the solution with respect to the robot model and the contact forces is implicitly given by the properties of the CoM trajectory computed by the OCP.

The corresponding contact forces and joint torques are then reconstructed with a dynamic simulator. For each time step, the contact forces are computed as the minimal forces corresponding to the joint trajectory  $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$  and respecting the contact model by

$$\min_{\mathbf{f}_1 \dots \mathbf{f}_M} ||RNEA(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) - \sum_i \mathbf{J}_i^T \mathbf{f}_i||^2$$

such that  $\mathbf{f}_i \in K_i$ ,  $i = 1..M$ , where *RNEA* stands for the *Recursive Newton Euler Algorithm* [Luh 1980]. The motion can be checked to be dynamically consistent if the residual is null for all time instants of the movement.

## 2.4 Results and simulation

### Experimental setup

We have considered the experimental setup depicted in Fig.2.1 as proof-of-concept. The goal is to make the humanoid robot HRP-2 climbing a stair case with a handrail as additional support. The height, the depth, and the width of the stairs are respectively 15cm, 30cm, and 1m. The hand rail is a cylinder with a diameter of 3cm. In this case the handrail is needed to avoid excessive power consumption in the leg motors. Starting and ending in a half-sitting position

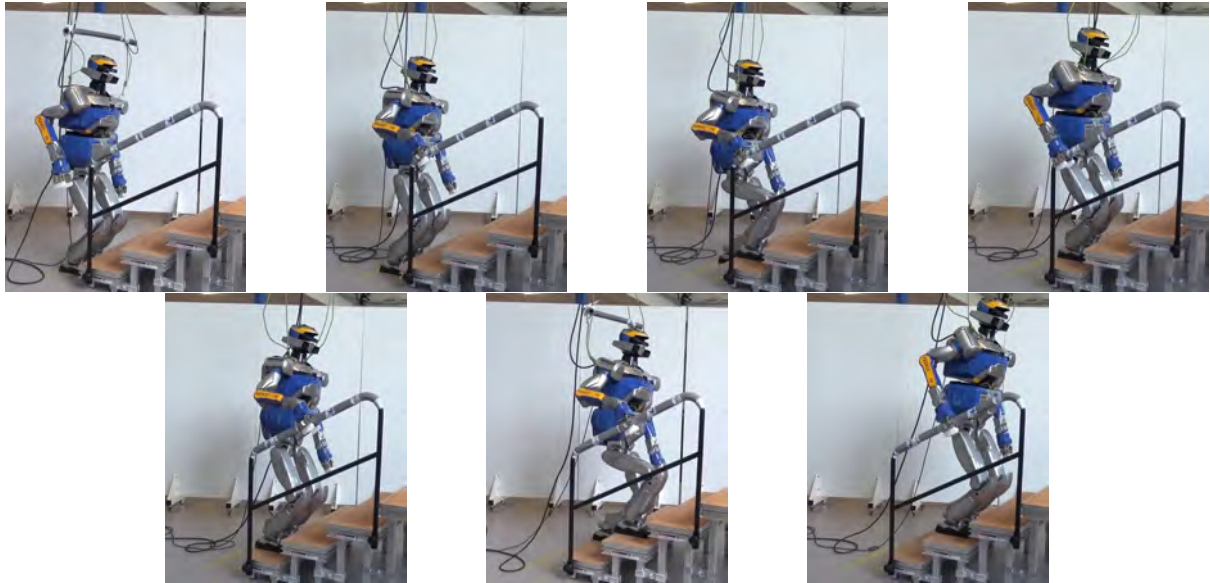


Figure 2.3: Set of contact stances realized by the humanoid robot HRP-2 using the method proposed in this chapter. The experiment has been realized 5 times in a row without failure. The accompanying video shows the realization of the experiments.

of the robot, the motion is divided into 3 phases that are executed twice: (i) the right hand establishes contact with the handrail, (ii) the right foot is set on top of the stair in front, (iii) the left foot is lifted and placed next to the right one on the stair. The robot is in double support phase during the movement of an end-effector and triple support phase during the CoM transition phase. See Fig. 2.3 for snapshots of the real humanoid robot HRP-2 performing the motion.

The robot moves an end-effector in 1.4s and moves its center of mass position in 0.1s during a transition phase. The timing of the phase durations is crucial for the robot because they implicitly define the velocity of each limb. From experience in ground-level walking, the period of single support and double support are usually around 0.7s and 0.1s. However, in this example the robot has to go through a larger distance at each phase than for ground-level walking. Keeping the same schedule as in ground-level walking makes the robot reaching its actuators limits in speed and current more likely.

All the computations (contact planning and WPG) were performed offline. A possible contact planner is open-source and available at <https://github.com/humanoid-path-planner>. The OCP is solved using the proprietary software MUSCOD provided by the Interdisciplinary Center for Scientific Computing (IWR) of Heidelberg University. This software offers a OCP toolkit (e.g. integration and numerical-differentiation routines) along with an efficient sparse sequential-quadratic-program solver. In this experiment the solver condensed the problem before solving it.

The approach presented in this chapter has been developed to prove the concept of using a template model for a multi-contact controller. In the control chain, only the OCP part is not yet run in real time. In fact the computation time for the motion in the video attachment is  $\sim 30$  minutes. The large computational foot print is due to (i) calculating the motion all at once on the whole preview-window of 18.4s, (ii) an over parametrization of the problem (3003 DoFs), (iii) not exploiting the intrinsic sparsity resulting from the template model. Future work will include a tailored implementation of the algorithm considering these bottlenecks and allow a real-time execution on the robot. Despite this, the inverse dynamics run in 1ms on the HRP-2 CPU (*Intel(R) Core2(TM) Duo E7500, frequency 2.8GHz, 1 core, 3Mb of cache*) under Ubuntu 10.04 LTS.

### Forces during contact transition

Fig. 2.4 shows the forces measured during one experiment. It appears that almost all the forces are acting along the  $z$ -axis. The right hand has an important role as it can realize forces up to 200N and more remarkably also exert negative forces at 9s and 14s, i.e., the robot also pulls itself using the grip on the handrail. Almost no torque is applied at the level of the feet, the propulsion of the robot is more visible in the tangential forces along the  $x$ - and  $y$ -axis.

Let us now focus on the  $z$ -axis. At the beginning of the motion, the robot is stable on its feet and no other contact with its environment is established (phase not shown in the graph). At 6s the hand comes into contact with the handrail causing perturbations in the feet force distribution. The next transition appears just before 8s. The robot shifts its CoM to the left foot and puts the right foot on the first stair. Then the robot has three contacts with its environment and starts to use the hand contact. It pushes with the right leg and pulls with the hand to climb the stair. This particular motion excites the flexibility located under the ankle of HRP-2. Between

10s and 12s the flexibility perturbs the system but the forces on the hand compensate for it. The hand contact stabilizes the robot and allows it to move the hand towards the next grasping position at 12s, where the robot is back to a stable state again. During the hand movement, the robot's CoM is affected by the flexibility exertion but not enough to fall down due the stabilizing influence of the grasping contact before and after the double support phase. The motion is repeated once. The only difference is that the hand does not release contact with the handrail at the end of the motion. This helps the robot to stabilize and go back to an equilibrium state.

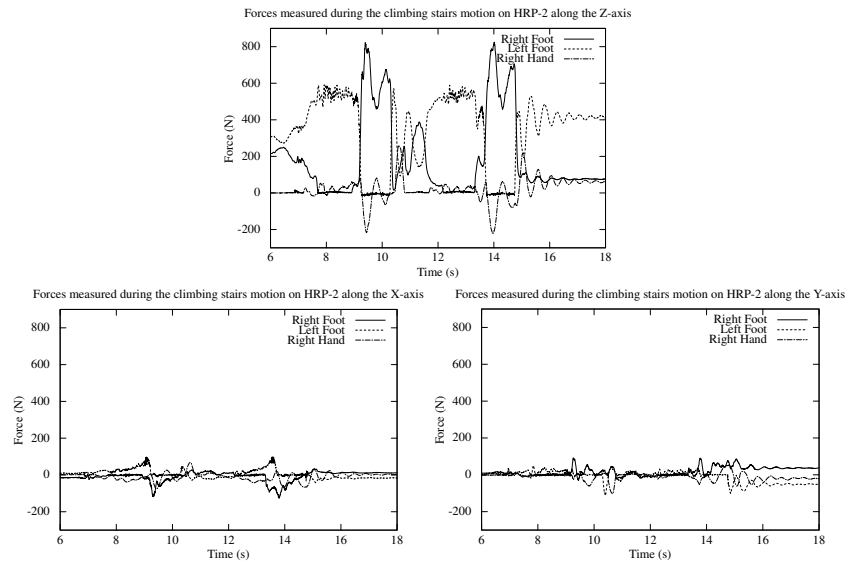


Figure 2.4: Measured forces on the HRP-2 humanoid robot during the motion depicted in Fig. 2.3.

## Comparison between OCP and reality

In Fig. 2.5 the measured  $z$  forces are compared with the guessed ones from the OCP. We can clearly see here that the two match at the beginning of the motion. However the compliance placed under the HRP-2 ankle get excited when the robot lifts its weight from the ground to the first step. This causes unplanned behavior as the flexibility is not modeled in the OCP. As future work, we want to include a model of the flexibility inside the template model of the dynamics.

## Current consumption

A severe limitation in climbing stairs with foot contacts only for human-sized humanoid robot is the current consumption. After performing a large number of experiments on a 15cm staircase using a different algorithm [Morisawa 2007], it appears that the rate of success was highly dependent on the battery charge level. Based on this observation, and using a model of the robot actuator, the maximum current amplitude was detected to be 40A on the right knee as depicted in Fig. 2.6. It is mostly due to the fact that the weight shifting is performed by one support leg. Using several contact points during weight shifting allows to distribute the load across several actuators. Therefore, the current asked for the right knee for the same motion using multiple contacts does not exceed 30A. This allows to perform the motion depicted in Fig. 2.3 5 times in a row even with low battery charge level.

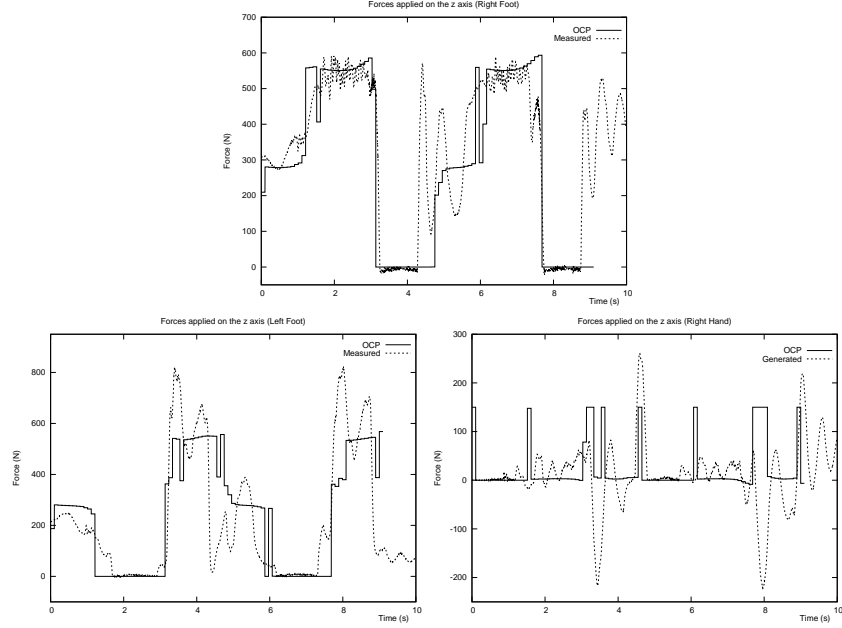


Figure 2.5: Measured forces on the HRP-2 humanoid robot during the motion depicted in Fig. 2.3 compared with the OCP solution. The graphs present respectively the right foot, left foot, and right hand  $z$  forces against time.

## 2.5 Contribution to a parallel work on this problem

My personal contribution to this chapter [Carpentier 2015] is on the experimentation on the real robot. I will therefore briefly depict the general idea corresponding to the OCP and present in a bit more detail the performance and the experiments done with this formulation.

### OCP formulation

As explained in the paragraph named *Formulation* in [Carpentier 2015], the OCP is as follow :

$$\min_{\substack{\underline{x}=(\mathbf{c},\mathbf{h},\mathcal{L}), \\ \underline{u}=\phi}} \int_0^T \ell_h(\mathbf{x}) + \ell_\kappa(\mathbf{x}) + \ell_{\mathcal{L}}(\dot{\mathbf{x}}) + \ell_\phi(\mathbf{u}) dt \quad (2.10a)$$

$$s.t. \quad \forall t \quad \dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \quad (2.10b)$$

$$\forall t \quad \phi \in \mathcal{K} \quad (2.10c)$$

$$\mathbf{x}(0) = (\mathbf{c}_0, \mathbf{0}, \mathbf{0}) \quad (2.10d)$$

$$\mathbf{x}(T) = (\mathbf{c}^*, \mathbf{0}, \mathbf{0}) \quad (2.10e)$$

$$\dot{\mathbf{h}}(0) = \dot{\mathcal{L}}(0) = \dot{\mathbf{h}}(T) = \dot{\mathcal{L}}(T) = \mathbf{0} \quad (2.10f)$$

where:

\*  $\ell_h(\mathbf{x}) = \lambda_h \|\mathbf{h}\|^2$  is the norm of the linear momentum,

\*  $\ell_\kappa(\mathbf{x}) = \sum_{k=1}^K \kappa(\mathbf{c}, \mathbf{p}_k)$  represents the kinematic limits of the robot's whole body by setting

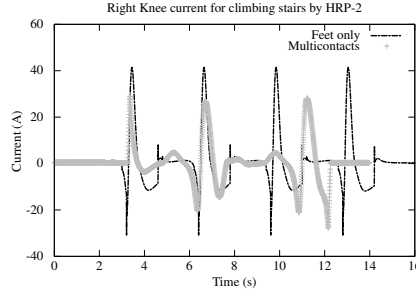


Figure 2.6: Current consumption comparison between 2-contact locomotion and 3-contacts locomotion for climbing a 15 cm staircase.

an exponential barrier on the distance between the COM and the contact points:

$$\kappa(\mathbf{c}, \mathbf{p}_k) = \exp(\|\mathbf{c} - \mathbf{p}_k\| - u_b) + \exp(-\|\mathbf{c} - \mathbf{p}_k\| + l_b) \quad (2.11)$$

- \*  $\ell_{\mathcal{L}}(\dot{\mathbf{x}}) = \lambda_{\mathcal{L}} \|\dot{\mathcal{L}}\|^2$  is the norm of the angular momentum time derivative,
- \*  $\ell_{\phi}(\mathbf{u}) = \|\phi\|^2$  is the norm of the forces applied at the contact point,
- \*  $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$  is the equation of the template model dynamics,
- \*  $u_b, l_b$  are the arbitrary upper and lower bounds,
- \* the weight  $\lambda_h$  is adapted depending on the phase: for support phase involving large displacement (like a large movement of the swing foot), the weight is divided by 10 with respect to its nominal value,
- \* and  $\mathcal{K}$  are the friction cones.

The main differences between both formulations are:

- the shape of the cost function,
- how the problem is regulated via a slack variable Eq. (2.9) or not Eq. (2.10),
- the parametrization of the system,
- the use of sparsity,
- and how linear constraint are represented in the problem.

It appeared that the second formulation is much more appropriate for the solver.

## Experimental results

Two main experiments carried out on the HRP-2 robot are presented. The first experiment concerns the generation of a classic walking motion: it is a unitary test but it is important to properly understand the behavior of our solver compared to classical walking pattern generators. The second experiment is the same climbing stairs scenario depicted in Fig. 2.7 and in Fig. 2.3, where the robot has to make use of the handrail to help its ascension of the stairs. Additional movements of running and standing up were simulated but not executed by the robot so the user is kindly asked to look at [Carpentier 2015] for more details.



Figure 2.7: Experiment 1: Walking in straight line with enjambment of 100cm.

## Experimental setup

All the computations (contact planning and WPG) were performed offline on a Intel Xeon(R) CPU E3-1240 v3 @ 3.40GHz. In this case the contact planner used is the open-source Humanoid Path Planner available at <https://github.com/humanoid-path-planner>. The OCP is also solved using the proprietary software MUSCOD provided by the Interdisciplinary Center for Scientific Computing (IWR) of Heidelberg University. In this experiment, however, we used the sparsity solver to solve the problem. For the walking experiments, we used a closed-loop control provided with the robot (called the stabilizer) to stabilize the movements of the rubber bush inside each foot [Mikami 2014]. The stabilizer was not used for climbing the stairs, because it is not able to handle hand contacts.

### Experiment 1: large enjambment on a flat ground

In this first experiment, a sequence of cyclic contacts is manually generated with enjambment of 100 cm. The timings are fixed (single support: 1.0 s; double support: 0.1 s). The total duration of the trajectory is 8.2 s. We then compute a feasible COM trajectory using the proposed OCP. The foot trajectories are a collection of splines connecting the desired configuration given by the contact sequence and ensuring a zero velocity and acceleration during take-off and landing of the foot. The experiment is summarized by Fig. 2.7 to 2.9. The enjambment of 100 cm is higher than the one performed in [Garcia 2014] or [Naveau 2017] that was up to 80 cm using a classical CoP walking pattern generator.

Fig. 2.8 reports the numerical behavior of the OCP solver. A near optimal solution (i.e. KKT tolerance below  $10^{-6}$ ) is obtained in 4 s after 50 iterations of the multiple shooting algorithm. The objective function decreases rapidly in the beginning, and slows down its progression as the algorithm tries to fulfill the path constraints. After a feasible solution is found, every new iteration (i.e. what is computed during one iteration of a MPC) lasts 40 ms. The overall



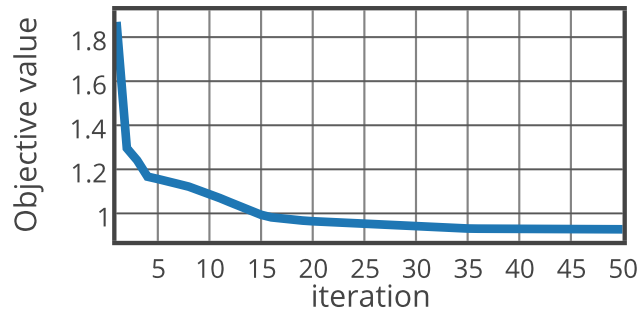


Figure 2.8: Experiment 1: Evolution of the cost function along the iterations.

movement is depicted in Fig. 2.7 (see also the accompanying video). The steps are very large for the humanoid robot (which is 1.60 m tall).

Fig. 2.9 shows the ZMP trajectory on the Y axis resulting from the OCP, compared to the estimation coming from force sensors measurement. The ZMP is very similar to what could be obtained by a classical WPG with assumption of flat contact. The proper tracking on the real robot shows the dynamic consistency of the output of the OCP.

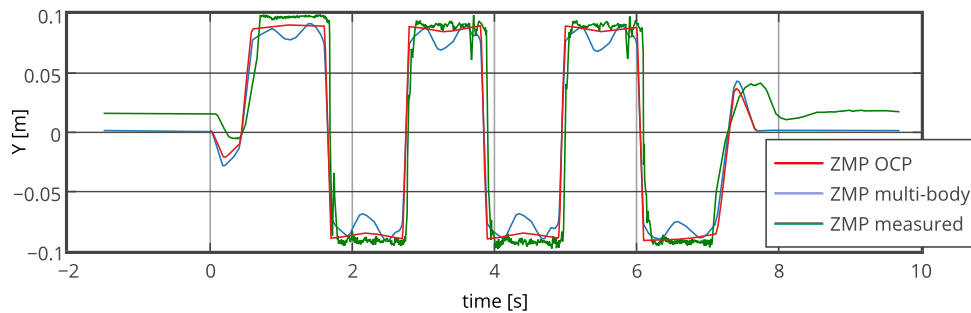


Figure 2.9: Experiment 1: ZMP trajectories obtained from the OCP, the multi-body dynamics and the measurements.

## Experiment 2: Climbing stairs equipped with a handrail

In the climbing scenario, the contact sequence given by the planner is not cyclic anymore and takes around 1s to be computed. The computation of a feasible trajectory to climb one stair is done in less than 5.5s after 85 iterations.

Fig. 2.11 illustrates both the forces computed by the solver and the forces exerted on the real robot. The simulated and measured forces do not match exactly but they have similar variations. In both cases, we observe that the robot makes use of its right hand either for pulling or pushing. The oscillations in the forces response are mainly due to the presence of a flexibility part in the robot's feet and to the compliance of the handrail. These two physical disturbances are not considered in our framework.

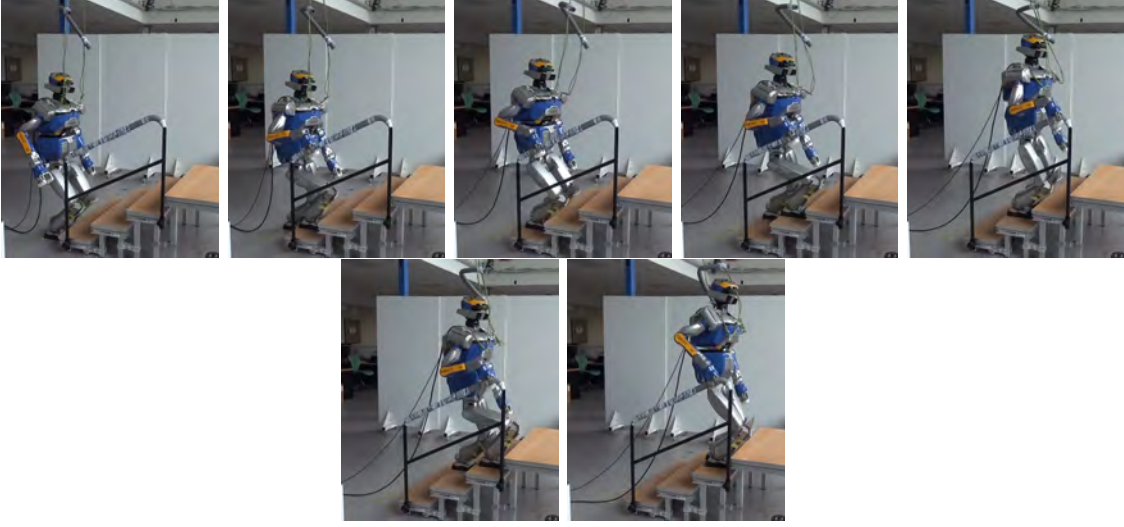


Figure 2.10: Experiment 2: Climbing the stairs of 15cm height while using the handrail.

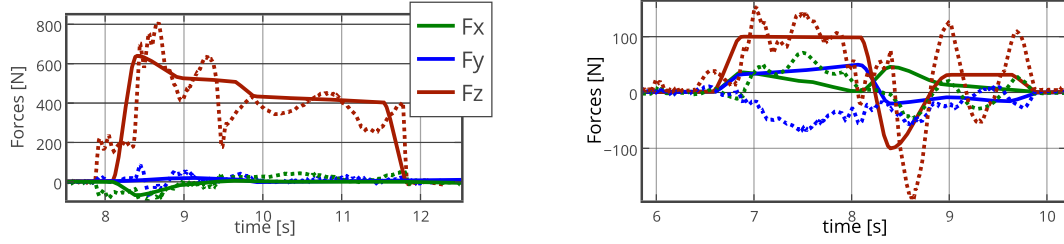


Figure 2.11: Experiment 2: Reference (solid line) and measured (dotted line) forces at the right foot (on the left) and hand (on the right) during one contact phase. The reference forces are properly tracked (even if some flexibility can be observed).

## 2.6 Conclusion

In this chapter we presented a method to generate a whole-body motion using multiple contact for a humanoid robot. It is based on an optimal control problem formulation focused on the under-actuated part of the robot's dynamics. Combined with state-of-the-art motion-generation algorithms the approach generates feasible trajectories that enable a humanoid robot HRP-2 to climb stairs. The generated motion was performed 5 times in a row on the robot. The current method was not optimized for speed and the next step is to have a real-time feasible implementation on the robot. The approach can also be applied to other kind of motions due to the general formulation, which will be investigated in future work.

CHAPTER 3

# Pulling a hose

---



From this chapter on we will present technical applications implemented on the real HRP-2. This chapter discusses a strategy for a humanoid robot to pull a fire hose while walking towards a desired position and orientation. This work has been submitted to *Humanoids* 2016. In this chapter the motion planning for picking the fire hose from the floor is firstly presented. Then a hybrid controller on the robot's wrist holding the fire hose is implemented for pulling it. The proposed controller can automatically determine the pulling force according to the robot's walking velocity. Through simulation analysis it is shown that when the robot walks while pulling the fire hose a drift in the walking direction is generated. To cope with this drift and to direct the robot to a desired position and orientation, a walking task is introduced. Using a motion capture system, the robot's chest position and orientation is monitored and feed to the robot's walking pattern generator to correct the orientation drift and to determine where to walk and when to stop walking. Through experimental results the validity of the proposed strategy was confirmed. It is shown that the proposed hybrid controller contributes to the improvement of the robot's balance when walking. The video of the application can be found at <https://youtu.be/K33d2VTHTTA>

### 3.1 Motivation

After Japan's 2011 earthquake and Fukushima's nuclear power plant disaster, the importance of developing robots capable of helping/replacing humans in dangerous situations or with increased capabilities have raised quickly. This natural disaster showed to the robotics community the lack of case studies that had been conducted towards applications in real case scenarios. Having this as a motivation, in this work we focus on the task of pulling a fire hose by a humanoid robot.

Previous works on manipulation tasks by humanoid robots include pushing objects, pivoting, lifting, etc. Hwang et al. discussed whole body motions of a humanoid robot pushing a wall [Hwang 2003]. Harada et al. proposed a controller for pushing manipulation by a humanoid robot where the desired trajectory of the ZMP is modified to push an object [Harada 2003]. They also discussed planning the robot's gait in real time to push a heavy object [Harada 2004]. Takubo et al. discussed pushing a heavy object by a humanoid robot. The center of mass (CoM) trajectory is modified based on the forces acting on the robot's hands [Takubo 2005a]. They also discussed a pushing method using multiple contact points between the robot and the object [Takubo 2005b]. Nozawa et al. proposed a full-body motion controller for a humanoid robot to push a heavy object by considering the friction forces acting on the robot's arms [Nozawa 2012]. Murooka et al. proposed a whole-body pushing motion by a humanoid robot considering force and balance and using selected contact points with the object to be pushed [Murooka 2015]. Murooka et al. also discussed the manipulation strategy to use for various objects by a humanoid robot. The forces applied on the robot are known but not the object dynamical parameters. Lifting, sliding and tilting the object are the three strategy that are taken into account [Murooka 2014]. Harada et al. discussed the task of lifting an object and then walk while holding it [Harada 2005].

In this work, we discuss the task of picking and then pulling a real fire hose by a humanoid robot while it walks. The fire hose is empty and rolled up into a reel that is fixed to the floor as shown in Fig. 3.1(a) the nozzle of the hose lays on the floor. A humanoid motion planner is used to generate the picking motion of the humanoid robot HRP-2 to avoid any self-collision

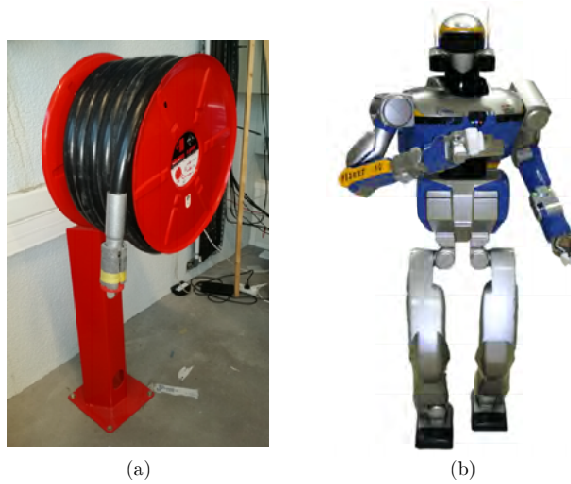


Figure 3.1: The fire hose is empty and rolled up into a reel used in this work is shown in (a) and the humanoid robot HRP-2 is shown in (b).

when the robot leans forward to pick the hose. Then a user friendly walking pattern generator, which input is a velocity relative to the ground plane, is used to generate the walking motion. For pulling the hose a hybrid controller on the robot's wrist holding the hose was implemented. Through simulation analysis it is shown that when the robot is walking while pulling the hose, a drift on the robot's walking direction is generated. A walking task is introduced to correct this drift. Using a motion capture system, the robot's chest position and orientation is tracked in real time. The walking task will compute the desired reference velocity for the robot to arrive to a desired position/orientation. Experimental results confirm the validity of the proposed motion for picking and pulling a fire hose. Moreover we show that the hybrid controller contributes to the improvement of the robot's balance when walking. It must be pointed out that whereas most of the previous work on pushing/pulling is based on force control, in this work we use a combination of position and force control to tackle the drift generated by the fire hose when the robot is walking.

This chapter is organized as follows: in section 3.2 we present the motion planning for the robot to pick the fire hose from the floor. In section 3.3 we briefly introduce the walking pattern generator used in this work. In section 3.4 we show the strategy for a humanoid robot to pull a fire hose and analyze simulation results. In section 3.5 we introduce a walking task to correct the drift in the robot's walking direction generated by the hose. In section 3.6 we show the experimental results of the proposed strategy using the HRP-2 humanoid robot. Finally, in section 3.7 we give the conclusion of this work and briefly discuss future work.

## 3.2 Picking motion

The complex motion of leaning forward and picking the hose from the floor pose several problems. First auto collisions are possible between the femurs and the hip joints due to the structure design of the HRP-2 humanoid robot. Second a reactive approach does not find a good solution because of local minima. For these reasons we employed the Humanoid Path Planner (HPP) to compute the motion for picking the hose from the floor. This exact same versatile library

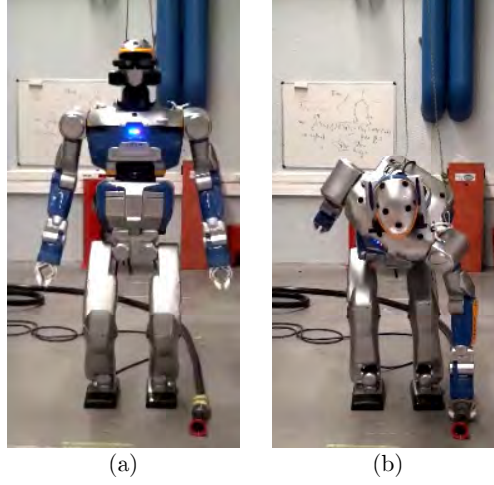


Figure 3.2: Snapshots of the experiment on the HRP-2 robot picking a fire hose from the floor.

we used to plan the contact point on the multi-contact climbing stair motion in Chap. 2. As a reminder HPP, [hpp 2016] is an open source software able to compute a collision-free path from a given starting robot configuration to a goal one. This path is subject to a set of constraints such as balance conditions, a particular posture for specific DOF(s), etc. It is also able to randomly search for a robot configuration satisfying a set of desired constraints, including environment and self-collision checking. For picking the fire hose from the floor, first we search for a robot configuration with the following constraints:

1. The robot's center of mass must be at the center of the support polygon.
2. Both of the robot's feet should keep in contact with the floor.
3. The left wrist height must be such that it can reach the hose without colliding with the floor.
4. The robot's waist orientation is fixed and facing forward.

Furthermore, we use the robot's half sitting configuration to limit the results from the random search to configurations that satisfy:

$$\mathbf{q}_{\text{res}} = \mathbf{q}'_{\text{rand}} + (\mathbf{I}_n - \mathbf{J}^+ \mathbf{J}(\mathbf{q}'_{\text{rand}}))(\mathbf{q}_{\text{hs}} - \mathbf{q}'_{\text{rand}})$$

where  $\mathbf{q}'_{\text{rand}}$  is the projection of a random configuration  $\mathbf{q}_{\text{rand}}$  into the constrained space,  $\mathbf{q}_{\text{hs}}$  is the robot's half sitting configuration,  $\mathbf{J}$  is the Jacobian of the robot tasks and  $\mathbf{I}_n$  is the identity matrix of  $n \times n$ , and  $\mathbf{q}_{\text{res}}$  is the projection of  $\mathbf{q}_{\text{hs}}$  on the constrained tangent space at  $\mathbf{q}'_{\text{rand}}$ . The desired robot configuration is the projection of  $\mathbf{q}_{\text{res}}$  into the constrained space  $\mathbf{q}'_{\text{res}}$ . After obtaining the picking configuration, we compute two free collision paths, the first starting at the half sitting configuration and finishing on  $\mathbf{q}'_{\text{res}}$ , and the second starting at  $\mathbf{q}'_{\text{res}}$  and going back to the half sitting configuration, both of them subject to constraints 1, 2 and 4 during the whole path. From the obtained path we extract the information of the robot's center of mass position, both feet's position and orientation, and the robot's configuration angles. These information will be use by an inverse kinematic framework for redundant robots that prioritizes robot tasks by projecting low priority tasks into the kernel of high priority tasks [Mansard 2009].

### 3.3 The walking pattern generator

In order to control the displacement of the humanoid robot HRP-2 we used the walking pattern generator presented in Chap. 1 and published in [Naveau 2017]. As a reminder, this walking pattern generator allows the user to control a humanoid robot almost like a mobile platform. The input of the walking pattern generator is a velocity  $[\dot{x}, \dot{y}, \dot{\theta}]$  relative to the ground plane, and is tracked by the center of mass. The walking pattern generator computes automatically the foot transitions and the center of mass trajectory, so that the robot is balanced when walking.

In this walking pattern generator, the input velocity is tracked by the center of mass if all the constraints are respected. As a consequence the walking pattern generator will create a swinging motion of the center of mass from one foot to the other even if the input velocity is zero on the robot coronal plane. The robot will move its center of mass to maintain balance while walking in any circumstances. This swinging motion has to be taken into account when designing the hand tasks to avoid auto-collision and the walking task to make the robot converge toward a specific goal.

### 3.4 Pulling strategy

In this section we explain the strategy for pulling the fire hose while walking. First, we described the hybrid controller used to control the robot's wrist holding the hose. Then, simulation results are shown for analyzing and demonstrating the robot's behavior when pulling the fire hose while walking.

#### 3.4.1 Hybrid controller

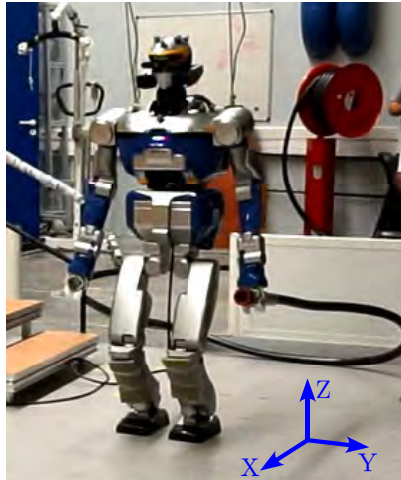


Figure 3.3: The robot coordinate frame.

In order to pull the fire hose while walking a hybrid controller is implemented on the robot's wrist holding the hose. Position control is employed to keep a fix distance between the waist of the robot and it's wrist, and impedance control is employed to pull the hose. According to the frame attached to the robot (as shown in Fig. 3.3), position control is used in  $Y$  direction and impedance control in  $X$  and  $Z$  directions. The impedance controller on the left wrist is defined



similarly to the one proposed by Harada et al. [Harada 2004] as:

$$m\ddot{\mathbf{x}}_{lw} + c\dot{\mathbf{x}}_{lw} = \mathbf{f}_{lw} - \mathbf{f}_d - \mathbf{f}_{\text{pull}}, \quad (3.1)$$

where  $\mathbf{x}_{lw} = [x_{lw} \ y_{lw} \ z_{lw}]^T$  represents the robot's left wrist position, and  $m$  and  $c$  are the desired mass and damping coefficients, respectively. The force applied to the left wrist of the robot is represented by  $\mathbf{f}_{lw}$ ,  $\mathbf{f}_d$  is the desired force in the left wrist, and  $\mathbf{f}_{\text{pull}}$  represents the desired pulling force, defined as:

$$\mathbf{f}_{\text{pull}} = \begin{cases} \mathbf{W}\mathbf{v}^{\text{ref}} & \text{if } z_{la} = z_{ra} = h_a, \\ 0 & \text{otherwise} \end{cases}, \quad (3.2)$$

where  $z_{la}$  and  $z_{ra}$  are the  $Z$  direction components of the left and right ankles,  $h_a$  is the height of the ankles when making full contact with the floor,  $\mathbf{v}^{\text{ref}}$  is the reference velocity vector of the robot when walking and  $\mathbf{W}$  is a diagonal matrix. Like this, the robot will pull the hose only at the double support phase when walking and will not pull the hose when standing still. Furthermore, the diagonal matrix  $\mathbf{W}$  allows to select in which direction to pull the hose.

The orientation of the wrist will be kept constant with respect to the orientation of the robot's waist, i.e. it will rotate in the same direction, same amount as the robot's waist.

### 3.4.2 Simulation analysis

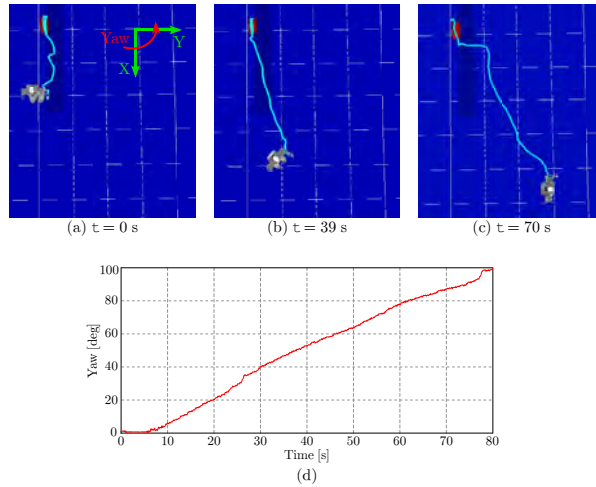


Figure 3.4: Snapshots of the top view of the simulation of HRP-2 robot holding a hose in (a) to (c), and the change in the robot's yaw angle in (d).

The humanoid robot HRP-2 is simulated using the OpenHRP software version 3.1.2. This software simulates the dynamics of the hose and HRP-2 during the walk. To generate the walking motion of the robot the walking pattern generator described in section 3.3 is used. The fire hose is approximated by a simplified model consisting of rigid cylinders connected by rotational joints with the rotation axis parallel to the cross section of the cylinders.

Fig. 3.4 shows the simulation results of the yaw orientation angle of the robot's waist when using a fire hose with a length of 7 m partially rolled up at the starting time (Fig. 3.4(a)). The hose has a total mass of 13.2 kg, which includes the mass of the nozzle and the coupling of the hose. A uniform mass distribution of 1.8 kg/m, which correspond to the real fire hose, is assumed

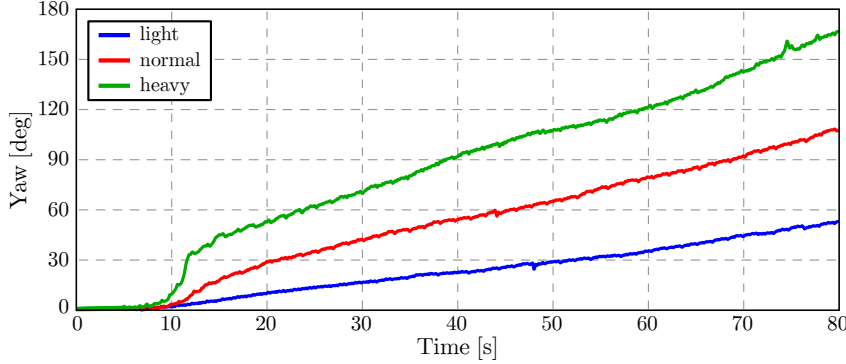


Figure 3.5: Change in the robot's yaw angle for three different weights per length unit. Light's hose mass is 4.52 kg, normal's hose mass is 9.04 kg and heavy's mass is 13.56 kg

for the cylinders. The reference velocity given to the pattern generator is  $\mathbf{v}_{ref} = [0.1 \ 0.0 \ 0.0]^T$  and it is constant through all the simulation. As it can be seen in Fig. 3.4, after walking few steps the robot begins to drift to its left with an almost constant velocity. It can be inferred that the force generated by the hose on the robot's wrist generates a slip on the robot's feet, thus generating a drift in the robot's yaw angle.

Furthermore, it was discovered that the amount of drift depends on the mass of the fire hose. The change in the yaw angle in simulation using hoses with different weights is shown in Fig. 3.5. The light, normal and heavy hoses have a total mass of 4.52, 9.04 and 13.56 kg, respectively and a length of 1.8 m for each of them. From this results it can be observed that the drift on the orientation of the robot depends on the disturbance given by pulling the hose on its left wrist.

### 3.5 Walking task

In this section we introduced a walking task to deal with the drift generated on the yaw angle of the robot when walking while holding the fire hose, as explained in the previous section. Then, simulation results are shown to confirm the validity of the proposed walking task.

To cope with the drift in the walking direction explained in the previous section, we introduced a walking task  $\mathbf{e}_w$  which is defined as:

$$\mathbf{e}_w = \mathbf{x} - \mathbf{x}_d, \quad (3.3)$$

where vector  $\mathbf{x} = [x_c \ y_c \ \phi_c]^T$  includes the robot's chest  $X$  and  $Y$  direction position and its yaw orientation angle  $\phi$ . Similarly, the desired position and yaw angle is represented by  $\mathbf{x}_d$ . Therefore, the desired reference velocity for walking  $\mathbf{v}^{ref}$  is obtained as:

$$\mathbf{v}^{ref} = -\lambda \mathbf{e}_w - \mathbf{\Lambda} \int \mathbf{e}_w(t) \, dt, \quad (3.4)$$

where  $\lambda$  is an adaptive gain and the matrix  $\mathbf{\Lambda}$  is a diagonal matrix of adaptive gains. The obtained reference velocity is used as an input to the walking pattern generator that will calculate the footsteps of the robot in real time, and also as an input to the impedance controller that will compute the desired pulling force according to the robot's reference velocity. Fig. 3.6 shows the simulation results for a walking task with  $\mathbf{x}_d = [4.0 \ 0.0 \ 0.0]^T$  when the robot starting position

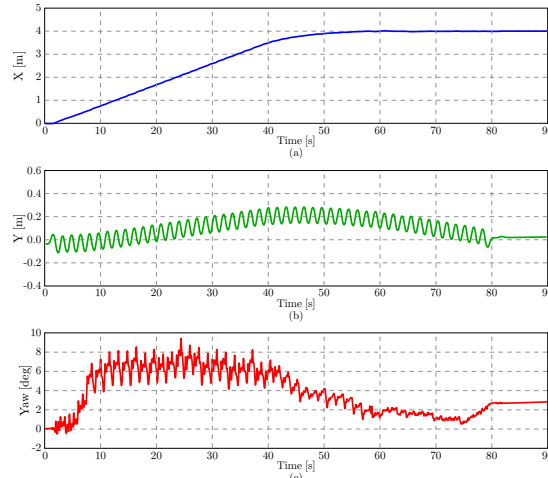


Figure 3.6: Robot's waist trajectory in simulation. In (a) position in  $X$  direction, in (b) position in  $Y$  direction and in (c) yaw angle.

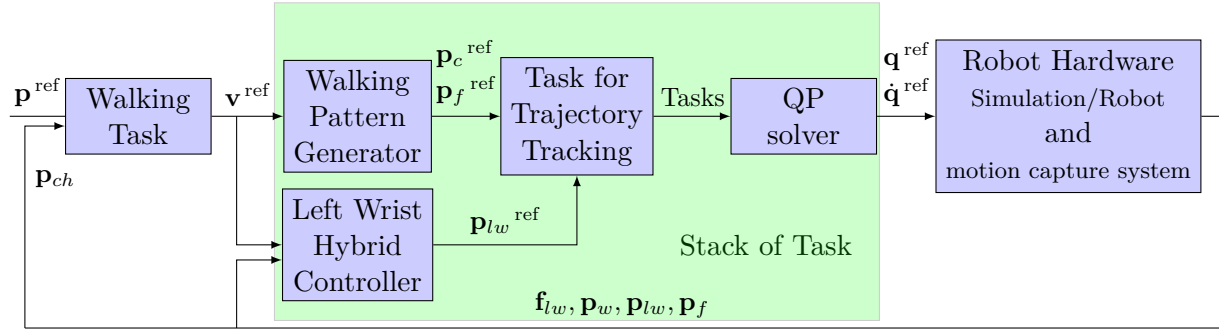


Figure 3.7: This scheme describe the feedback loop used to control the humanoid robot HRP-2. With  $\mathbf{p}^{\text{ref}}$  as the user defined pose,  $\mathbf{v}^{\text{ref}}$  as the velocity computed from the walking task,  $\mathbf{p}_c^{\text{ref}}$  as the center of mass reference trajectory,  $\mathbf{p}_f^{\text{ref}}$  as the feet reference trajectories,  $\mathbf{q}^{\text{ref}}, \dot{\mathbf{q}}^{\text{ref}}$  being respectively the generalized position and velocity vector,  $\mathbf{p}_{lw}^{\text{ref}}$  as the reference left wrist position,  $\mathbf{f}_{lw}, \mathbf{p}_w, \mathbf{p}_{lw}, \mathbf{p}_f, \mathbf{p}_{ch}$  being respectively the measures of the left wrist force sensor, the waist position, the left wrist position, the feet position and the chest position.

is  $\mathbf{x}_i = [0.0 \ 0.0 \ 0.0]^T$ . A maximal velocity of 0.10 and 0.15 m/step for the  $X$  and  $Y$  directions, respectively, and 5 deg/step are set to the walking task. It can be seen that the robot reaches the desired position within 80 seconds and also that the walking task is able to correct the drift generated by the hose.

### 3.6 Experimental results

In Fig. 3.7, the feedback control loop used in the experiments is shown. The walking task uses the robot's chest position given by the motion capture system to compute the reference velocity  $\mathbf{v}_{\text{ref}}$  for the walking pattern generator.  $\mathbf{v}_{\text{ref}}$  is also used by the impedance controller to compute the pulling force to be applied. The reference positions for the feet and the center of mass are computed by the walking pattern generator, and the reference position of the left wrist is computed by the hybrid controller. These reference positions are sent to a generalized inverse kinematics framework [Mansard 2009] that will compute the robot's joints

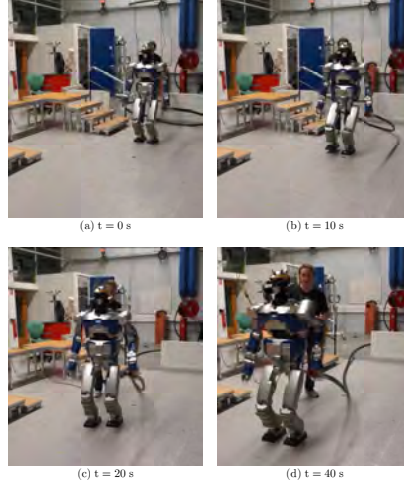


Figure 3.8: Snapshots of the experiment on the HRP-2 robot pulling a fire hose.

angles reference trajectories to be executed by the robot. The control framework uses the robot's forward kinematics to compute the current positions of the robot bodies, and the force sensors information to feedback the system.

The fire hose used in this work has a weight of 1.8 kg/m, and a total length of 30 meters. There is no water inside. It is empty and rolled up into a reel which is fixed to the floor, as shown in Fig. 3.1(a). For the picking experiment, the nozzle of the fire hose lays on the floor as shown in Fig. 3.2(a) where the robot is at its half sitting configuration, i.e. its starting posture. Fig. 3.2(b) shows the robot posture at the picking time. It can be seen that the robot successfully reaches the hose laid on the floor, without losing balance and without any self-collisions.

For the walking experiment the robot will pull the hose only in its forward walking direction, i.e. the diagonal elements of matrix  $\mathbf{W}$  in equation (3.2) are  $w_{11} = \beta$ ,  $w_{22} = 0.0$ , and  $w_{33} = 0.0$ , where  $\beta$  is a constant. Using a motion capture system composed of 10 infrared cameras (Motion Analysis Corp. [moc 2016]), the position of the robot's chest is tracked on real time, with a sampling frequency of 200 Hz. The robot's starting position is  $x_c = -1.1$  m,  $y_c = 1.56$  m and the desired position given to the walking task is  $\mathbf{x}_d = [1.0 \ 1.5 \ 0.0]^T$  with a tolerance of 5 cm in  $X$  and  $Y$  direction and 5 deg for the yaw angle. This means that the robot will stop walking when all of the errors are within the tolerance value. The walking task is set with a maximal velocity of 0.1 and 0.15 m/step for the  $X$  and  $Y$  directions, respectively, and 5 deg/step. The step duration is fixed and lasts 0.8 s. Fig. 3.8 shows snapshots of the experiment carried out on the HRP-2 humanoid robot, and Fig. 3.9 shows the corresponding robot's chest trajectory tracked by the motion capture system.<sup>1</sup> It can be seen that the robot reaches the desired position/orientation, having walked more than 2 m while pulling the fire hose. The change in the robot's orientation from Fig. 3.8(c) to (d) can be observed if we carefully look at the orientation of the feet. Also, the pulling movement of the robot's left arm can be appreciated by looking at the position of the left arm's elbow in Fig. 3.8(d).

Furthermore, Fig. 3.10 shows the final steps of the ZMP trajectories of the robot during the experiment, the bold red line represents the reference trajectory computed by the walking pattern generator and the thin blue line represents the real ZMP trajectory computed based on

<sup>1</sup>The video of the experiment can be found in the multimedia attachment.

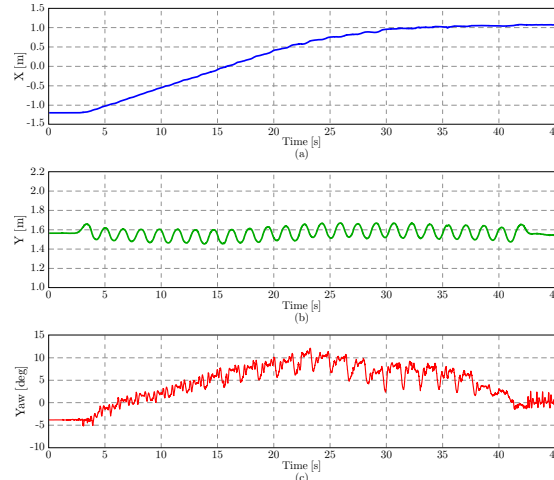


Figure 3.9: Robot's chest trajectory in experiment. In (a) position in  $X$  direction, in (b) position in  $Y$  direction and in (c) yaw angle.

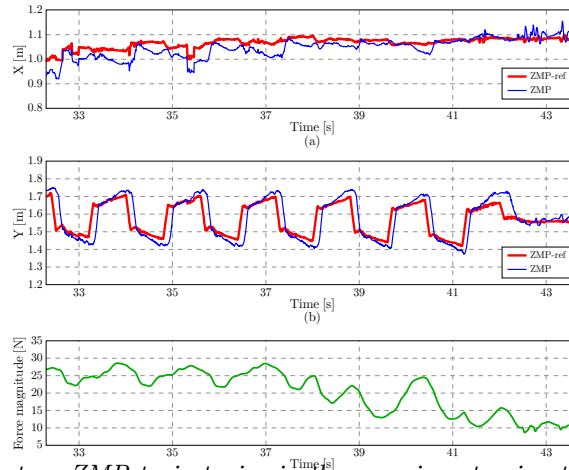


Figure 3.10: Robot's final steps ZMP trajectories in the experiment using the proposed hybrid controller on the left wrist of the robot. In (a) position in  $X$  direction, in (b) position in  $Y$  direction and in (c) the magnitude of the force applied by the hose on the left wrist.

the force sensors of the robot's ankles and the position of the feet. It can be seen that the ZMP, particularly in  $Y$  direction, moves away from the reference when the robot is pulling the hose, i.e. during the double support phase. This indicates that the robot is losing balance at those points. Nevertheless we were able to successfully reproduce the experiment 4 times out of 8 trials.

Fig. 3.11 shows the final steps of the ZMP trajectories of the robot for the experiment where no impedance control is being applied to wrist holding the hose i.e. the robot's joint angles of the left arm are fixed. Here, it can be observed that in  $Y$  direction the ZMP has rebounds immediately after the left foot has landed on the floor. On the contrary, during with the hybrid controller on the wrist there are no rebounds since the impedance controller is allowing the left arm to absorb part of the disturbance generated by the hose. This means, that the impedance control on the wrist of the robot contributes to the improvement of the robot's balance while walking.

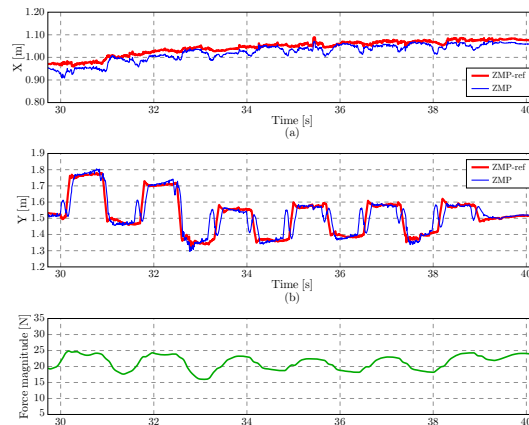


Figure 3.11: Robot's final steps ZMP trajectories in the experiment with the joint angles of the left arm fixed. In (a) position in  $X$  direction, in (b) position in  $Y$  direction and in (c) the magnitude of the force applied by the hose on the left wrist.

### 3.7 Conclusion and future work

This chapter discussed a strategy for a humanoid robot to pull a fire hose while walking towards a desired position and orientation. The main results in this chapter are summarized as follows:

1. We proposed a hybrid controller for the robot's wrist holding the fire hose. Position control is used to guarantee no self-collision, while impedance control is employed to pull the hose according to the walking velocity of the robot.
2. Through simulation analysis it was discovered that the hose generates a disturbance on the robot's walking dynamics which in turn produced a drift on the robot's yaw angle.
3. We introduced a walking task to direct the robot to a desired position/orientation and at the same time correct the drift generated when holding the fire hose and walking.
4. We showed experimental results that verified the validity of the proposed controller for pulling the hose and the walking task introduced to correct the walking direction of the robot.
5. We showed that the proposed hybrid controller applied to the wrist holding the hose contributes to the improvement of the robot's balance when walking.

In this chapter we have demonstrated the capability of a humanoid robot to pull a fire hose. However, as shown in Fig. 3.10 the center of pressure of the robot is further and further from its reference as the robot pulls the hose. One possible way to solve this problem is to take the external forces, in this case the hose on the wrist, into the walking pattern generator algorithm. In the mathematical formulation of the walking pattern generator external forces can be introduced as an additional acceleration on the center of mass. Preliminary results have been shown in simulation Chap. 1 of this thesis. As future work we will implement this force feedback on the real HRP-2.

# Robust human-inspired power law trajectories for humanoid HRP-2 robot

---





This chapter presents the collaboration, done in the frame of the KoroiBot project, with our colleague experts in human motion from the Weizmann Institute of Science. This work has been published in [Karklinsky 2016]. We studied the interest of using the one-third power law for humanoid robot walking control. The one-third power law dictates how human speed of motion depends on its curvature. We observe that humanoid robots, following a reference trajectory, benefit from this law by reducing closed-loop drift and generating more human-like motion. To robustly execute reference trajectories, we use contracting morphed Andronov-Hopf oscillators, regularized to follow a power law while converging to a planned cyclic trajectory. We used the walking pattern generator described in Chap. 1 to make HRP-2 follow these guiding dynamics to walk along elliptic trajectories. In dynamic simulation, we observe minimal geometric drift with the one-third power law, outperforming constant speed and other power laws. Close-loop experiments on HRP-2 resulted in a small drift of all power law motions from the reference trajectory, showing the efficiency of the control architecture. We observed that the one-third power law controller demanded less compensatory action, and therefore lowered the burden on the hardware. Slowing in curved movement regions also allowed for faster overall movement. In this chapter we will briefly introduce how the walking pattern generator were used in this context. Then we will present the regularization of contracting oscillators. And finally experimental results on HRP-2 will be analyze.

## 4.1 Introduction

### 4.1.1 Power laws governing human motion

The speed of human motion is characterized by the one-third power law behavior; movement speed  $v$  decreases when curvature  $\kappa$  increases, following the quantitative relation  $v = \gamma\kappa^{-\beta}$ , with  $\gamma$  the piecewise constant velocity gain factor and  $\beta = 1/3$  the exponent. This law, often termed the two-thirds power law due to equivalent formulation determining angular speed  $A = \gamma\kappa^{2/3}$ , was first found for drawing hand motion [Lacquaniti 1983].

Power law behaviors appear to emerge from jerk minimization [Huh 2015]. Alternatively, the specific one-third power law, which is equivalent to moving with a constant equi-affine speed [Flash 1996], may result from equi-affine metrics used by the human brain [Flash 2007], possibly in a mixture with Euclidean and full affine metrics [Bennequin 2009].

Tuned to perception of biological motion, the human visual system perceives one-third power law motion as uniform, rather than movement with constant speed [Viviani 1992]. Coupling between the motor and visual systems in the human brain is supported by stronger and more widespread brain activation patterns [Dayan 2007], and greater event-related desynchronization [Meirovitch 2015], occurring when subjects watch one-third power law motion, compared with other power law motions. Even imagined movements slow down in curved regions according to the one-third power law [Papaxanthis 2012, Karklinsky 2015]. These evidences suggest that humans will perceive a humanoid robot moving according to the one-third power law as more human-like, finding his motions more predictable and easier to follow.

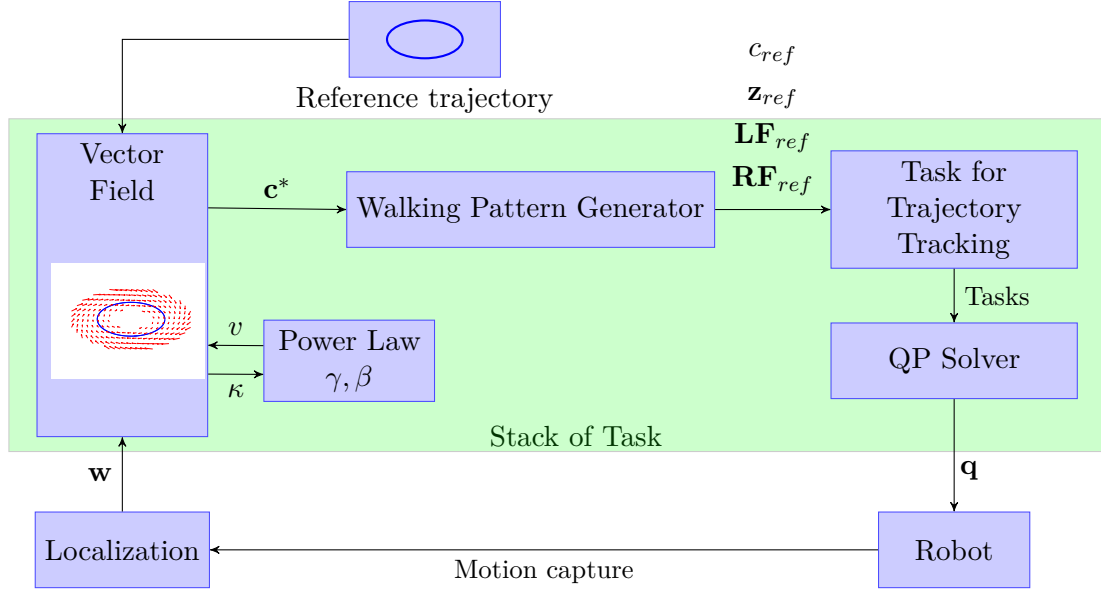


Figure 4.1: Power law based closed-loop control. An ellipse reference trajectory is specified and used, together with the power law defined by the velocity gain factor  $\gamma$  and  $\beta$  exponent constants, to generate the reference vector field.  $(x, y, \theta)$  define the position and orientation of a frame attached to the robot's center of mass. The reference vector field defines center of mass velocity  $\mathbf{c}^* = [\dot{x}, \dot{y}, \dot{\theta}]$  for the walking pattern generator. This particular walking pattern generator provides reference trajectories for the balanced center of mass  $\mathbf{c}_{ref}$ , the associated center of pressure  $\mathbf{z}_{ref}$ , and the feet  $\mathbf{LF}_{ref}, \mathbf{RF}_{ref}$ . A whole body motion generator uses the reference trajectories to generate a command  $\mathbf{q}$  realized by the robot (for HRP-2, this is the configuration vector). The localization component uses position measurements to provide a position  $\mathbf{w} \in SE(3)$  of the robot, that again is used as input to the vector field to provide the correcting velocity vector  $\mathbf{c}^*$ . This chapter shows that the closed-loop approach reduces motion drifts, but not entirely, and suggests that using human-like power laws may help robots to perform faster, more accurate, and more stable motion.

#### 4.1.2 Guiding trajectories for humanoid robots

Planning humanoid robot locomotion is classically accomplished by constructing an optimal sequence of footstep transitions, taken from a discrete set of possibilities, that is kept small to allow reasonable computation time [Chestnutt 2010, Hornung 2012]. The limitations of this approach in restricted scenarios yield an alternative approach; planning a reference trajectory which allows generation of the needed contacts [Perrin 2011]. This reference trajectory is usually generated considering balance constraints and geometrical constraints such as feasibility and manipulability. This approach significantly reduces the burden on the motion planner, but raises the demand for controllers capable of finding footsteps or contacts in real time. Planning general contacts is still a very hard problem [Escande 2013], and finding a center of mass trajectory for a given set of contacts was only recently accomplished [Carpentier 2015]. For the restricted case of walking on flat ground, several new approaches [Deits 2014, Naveau 2017, Herdt 2010a] allow automatic finding of footstep positions. Therefore, it is now possible to move a humanoid robot by providing only a reference trajectory [Naveau 2017, Herdt 2010a].

In the current study, we attempt to robustly regulate the walking motion on flat floor given a planned reference trajectory. A stable regulation process is important for correcting drifts,

that appear due to the interaction of the humanoid robot's soles with the ground [Stasse 2006]. The closed-loop control does reduce the drift. However, in this study we observe that the speed profile of the reference trajectory is crucial. The naive approach of close-loop regulation to obtain constant reference speed is unsatisfactory for HRP-2; a steady-state error is still apparent. Speed modulation according to the one-third power law nullifies this drift entirely.

### 4.1.3 Robust trajectories with contracting dynamics

Contracting dynamics provide a robust control policy for generating motion patterns; in presence of bounded noise exponential time convergence to a limit cycle or point is guaranteed [Lohmiller 1998]. By mapping an attractor of a contracting dynamical system to a movement primitive, Giese et al. [Giese 2009b] suggested a general control method for high dimensional systems. In robotics, contraction was already used for robust synchronization of phases of control pattern generators [Seo 2010] as well as for learning a set of dynamic motion primitives [Perk 2006]. In this work, we show how contracting dynamics are useful to robustly generate kinematic power law behavior.

## 4.2 Reactive walking pattern generator

In order to test power law motions on a humanoid robot we used again the walking pattern generator already presented in Chap. 1. In this walking pattern generator, tracking the input velocity by the center of mass has a lower priority than footsteps placement and balancing, therefore the robot creates a swing motion of the center of mass from one foot to another as already explained in Chap. 3. The swinging motion causes perturbations with respect to the reference trajectory [Stasse 2006]. As seen in our dynamic simulations, the swinging motion induces delays and sometimes shifts away from the planned reference trajectory. To mitigate the effect of the center of mass swing motion, we considered the barycenter of the robot's feet as the robot's location. This point reflects the robot's position, without the coronal oscillations. In addition to the walking pattern generator, we used a whole body motion generator [Mansard 2009] to compute a motor command realizing the reference trajectory. Finally, two local feedback loops are used to bring back the robot to the reference trajectory; a PID controller over the orientation of the robot, and more importantly a vector field providing the correcting linear velocity for the walking pattern generator. We describe the construction and regularization of this vector field in the next section.

## 4.3 Regularization of contracting oscillators

In this section we will use the work done by M. Karklinsky and A. Mukovskiy and presented in [Karklinsky 2016]. For a given reference trajectory M. Karklinsky and A. Mukovskiy designed a contracting system using Andronov-Hopf oscillator. The reference trajectory is a specific solution of the global dynamics. As a reminder, a dynamical system is contracting if all solutions in the contraction region converge exponentially to each other. If a dynamical system is contracting, there exists an attractor in this contraction region, and all trajectories in the region converge exponentially to it [Lohmiller 1998]. Contraction guarantees the exponential decays of perturbations. Partial contraction is contraction towards any of the particular solutions

residing inside the flow-invariant manifold of the dynamics [Pham 2007]. For the current work, it suffices that the reference trajectory is an invariant one dimensional submanifold of partially contracting dynamics, with local contraction towards but not along the trajectory. The partial contraction is easy to construct for an arbitrary trajectory, as demonstrated by M. Karklinsky and A. Mukovskiy in [Karklinsky 2016] for cyclic trajectories. Trajectories that are the natural test ground of power laws in human motion. Each partially contracting system are regularized to obey a power law on all orbits, including the reference trajectory.

### 4.3.1 Morphed Andronov-Hopf oscillators

This section recalls the construction of a stable dynamical system with desired limit cycle trajectory from an angular morphing of the basic Andronov-Hopf oscillator.

#### Angular morphed Andronov-Hopf oscillator

The Andronov-Hopf oscillator [Park 2009a]:

$$\begin{aligned}\dot{\theta} &= \omega \\ \dot{\rho} &= \alpha(1 - \rho)\end{aligned}\tag{4.1}$$

With the winding angle  $\theta$  and radius  $r_0(\theta)$ , defined as the orientation of the path's normal and its radius of curvature. They are the direction and radius of the osculating circle as well.  $\rho = \frac{1}{r^2} = \frac{1}{x^2+y^2}$ ,  $x, y$  being the Cartesian coordinates. And with  $\omega, \alpha > 0$  being two constants. For a path with positive (negative) curvature everywhere, like elliptic curves, the direction and radius of the osculating circle define  $(\theta, r_0(\theta))$  at each point. These coordinates on the path naturally extend to coordinates that cover a band around it. For a constant  $B = \min_{\theta}(r_0(\theta)) - \varepsilon > 0$ , in the compact band of width  $B$  around the path, each point in the  $(x, y)$  plane will map locally to one closest point with coordinates  $(\theta, r_0)$ . Its coordinates will therefore be  $(\theta, r(\theta))$ ; the winding angle and the distance from the center of the osculating circle of the closest point. To define a dynamical system on the angular coordinates, we define the coordinate  $\rho(\theta) = \frac{1}{r^2(\theta)}$ , and use the dynamics from Eq. 4.1. It is partially contracting, since the Jacobian of the  $\rho$  subsystem is uniformly negative definite  $J_{\rho} < -\alpha < 0$ . This oscillator can be morphed [Ajallooeian 2013]:

$$\begin{aligned}\dot{\theta} &= \omega \\ \dot{\rho} &= \alpha(F(\theta) - \rho) + \omega \frac{dF}{d\theta}\end{aligned}\tag{4.2}$$

It is still partially contracting, with  $F(\theta) = \frac{1}{r_0^2(\theta)} = \kappa^2(\theta)$  the squared path curvature. Global partial contraction in the  $\theta, \rho$  coordinates to the limit cycle results in partial contraction to the reference path in the band of width  $B$ .

### 4.3.2 Temporal regularization of a dynamical system

In this section, we define the power law regularization of dynamical systems and examine the conditions for its applicability. Under reasonable conditions, the regularized dynamical system has exponential convergence to the limit cycle of the original system. For some special cases the regularized system is contracting.

### 4.3.2.1 Power law regularization

We focus on regularization with a power law  $v = h(\kappa) = \gamma\kappa^{-\beta}$ , with  $\gamma$  a global constant and  $\beta$  the exponent value. This is the basic most useful example of the general Euclidean invariant dependency of speed upon path geometry,  $v = h(\kappa, \frac{d\kappa}{ds}, \dots, \frac{d^n \kappa}{ds^n})$  (see [Bennequin 2009]).

**Definition 1** *For a dynamical system  $\dot{\mathbf{x}} = F(\mathbf{x})$ , and a power law  $v = h(\kappa)$ , we define the  $h$ -regularization of  $F$ , denoted by  $\dot{\mathbf{x}} = F_h(\mathbf{x})$ , as  $F_h(\mathbf{x}) = \frac{F(\mathbf{x})}{|F(\mathbf{x})|} h(\kappa(\mathbf{x}))$ , with  $\kappa$  the curvature of the orbit at point  $\mathbf{x}$ .*

At each point, the orbit of  $F_h$  geometrically coincides with the orbit of  $F$ . Additionally, the speed along each orbit of  $F_h$  satisfies the law  $h$ . Each velocity vector has the same direction in both system  $F_h$  and  $F$ , but not necessarily the same norm. Our power laws are positively monotone so the directions of flows of  $F$  and  $F_h$  match.

**Assumptions 1** *We consider a region  $U$  which is a compact trapping region in the plane for  $F$  without fixed points. We require that for  $F$ 's orbits  $\kappa$  is defined and bounded, and therefore  $h$  is defined and bounded; for each  $\mathbf{x} \in U$ ,  $h(\mathbf{x}) \geq C_1 > 0$  for some global constant  $C_1$ . We require that  $F$  has bounded speed  $|F(\mathbf{x})| < C_2$  for some global constant  $C_2$ .*

These assumptions guarantee that, if  $F$  is contracting in  $U$  to some limit cycle, then each solution converges globally exponentially to this limit cycle in  $U$  (see Lohmiller and Slotine's [Lohmiller 1998], Theorem 1). While we do not claim that  $F_h$  is generally contracting, our assumptions yield global exponential convergence of  $F_h$  to the limit cycle of  $F$ . This is true because  $|F_h(\mathbf{x})| = \frac{h(\mathbf{x})}{|F(\mathbf{x})|} |F(\mathbf{x})| \geq \frac{C_1}{C_2} |F(\mathbf{x})|$ . The limit cycle of  $F_h$  is geometrically identical to that of  $F$ , and movement on it obeys the  $h$  power law.

### 4.3.2.2 Curvature of the orbits of the circular Andronov-Hopf oscillator

Power law regularization has singularities in inflection points and along straight trajectories; if  $\kappa = 0$  and  $\beta > 0$ , the power law speed is infinite. This can be overcome in several ways. For most practical needs the power law speed  $v$  can be bounded; Viviani and Stucchi [Viviani 1992] defined  $v = \gamma(\kappa + \alpha)^{-\beta}$ , with some constant  $\alpha > 0$  preventing the singularity at  $\kappa = 0$ . Alternatively, we can restrict the discussion to regions of positive curvature. For the morphed Andronov-Hopf oscillators, around a planned cyclic trajectory with positive curvature, there exists a band of bounded nonzero curvature, guaranteeing that the regularization process will result in finite speeds.

The dynamics of  $\rho(\theta)$  define  $\kappa(\rho)$  along each orbit. For circular Andronov-Hopf oscillators, the dynamics are invariant with respect to rotations around the origin  $(x, y) = 0$ . Therefore local curvature of the vector flow  $\kappa(\rho)$  is a function of  $\rho$  only, independent of  $\theta$ . For any integral trajectory its local curvature  $\kappa(\rho)$  equals zero exactly where it crosses a circle  $\rho_{\kappa=0}$  concentric to the unit limit cycle and bigger. Therefore, in the circular band  $C \geq \rho \geq \rho_{\kappa=0} + \varepsilon$ , for arbitrarily small  $\varepsilon$  and arbitrarily large  $C$ , the curvature is strictly positive  $\kappa \geq C_1 > 0$ , speed is bounded, and our assumptions hold, allowing power law regularization that results in exponential convergence to the limit cycle.

### 4.3.2.3 A contracting one-third power law regularized elliptic oscillator

For the one-third power law, any elliptic system generated as a linear transformation of the regularized unit circle oscillator is contracting. The argument of the unit circle regularized oscillator, based on the circular symmetry of  $\kappa$ , holds for uniform scaling. A global equi-affine transformation of the plane conserves the one-third power law and therefore the transformation of the unit circle dynamics to elliptic dynamics using the combination of a scaling and an equi-affine transformation conserves contraction of the regularized system. As a consequence a velocity vector field can be computed from any elliptic reference trajectory. This velocity vector field has the property to make a point mass converge toward the reference trajectory if this point mass track the velocity of this vector field. In Fig. 4.2 we can see the initial unit cycle oscillator in subFig. 4.2.a. This unit cycle can be morphed without loss of the contracting property of the reference trajectory (see subFig. 4.2.b). The subFig. 4.2.c-4.2.f correspond to the velocity vector field regularized by the power law with  $\beta = -1/3, 0, 1/3, 2/3$ . We can easily see here the influence of  $\beta$  on the vector field. If  $\beta$  is small the velocity will be high when the curvature is high.

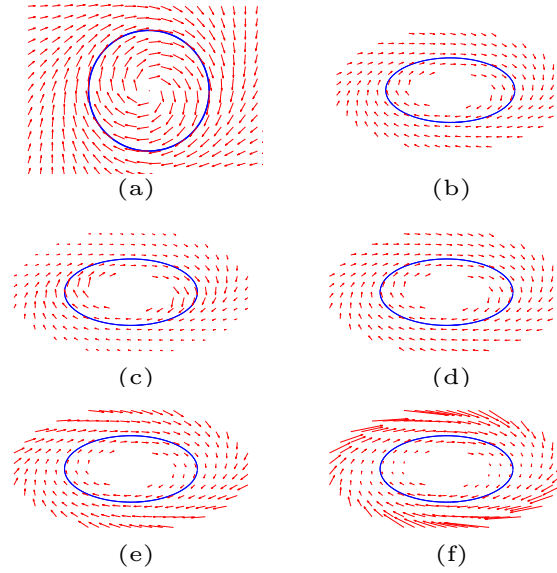


Figure 4.2: Morphing and regularization of an Andronov-Hopf oscillator. a) The unit cycle oscillator. b) an elliptic morphing. c)-f) Four power law regularizations according to  $\beta = -1/3, 0, 1/3, 2/3$  power laws respectively. Each regularization keeps directions but changes speeds of the elliptic vector field.

## 4.4 Results

In order to evaluate the effect of applying the one-third power law to humanoid robot walking, we integrated the two third power law inside the Stack-of-Task framework. We present dynamic simulations and real robot experiments testing motions generated by the walking pattern generator (Fig. 4.1). We compared four different power laws (Fig. 4.2), with exponents  $\beta = -1/3, 0, 1/3, 2/3$ . We explain the methodology and describe the results.

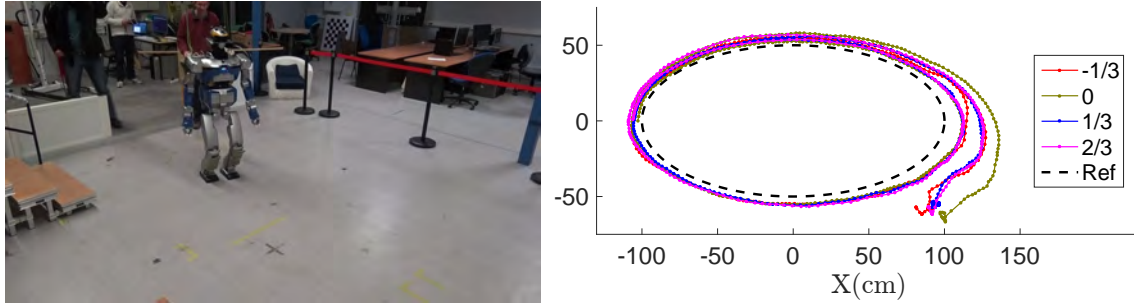


Figure 4.3: Experimental setup, the robot walks according to a reference ellipse, obeying one of four different power law speeds. Trajectories for each power law converge around a limit cycle ellipse.

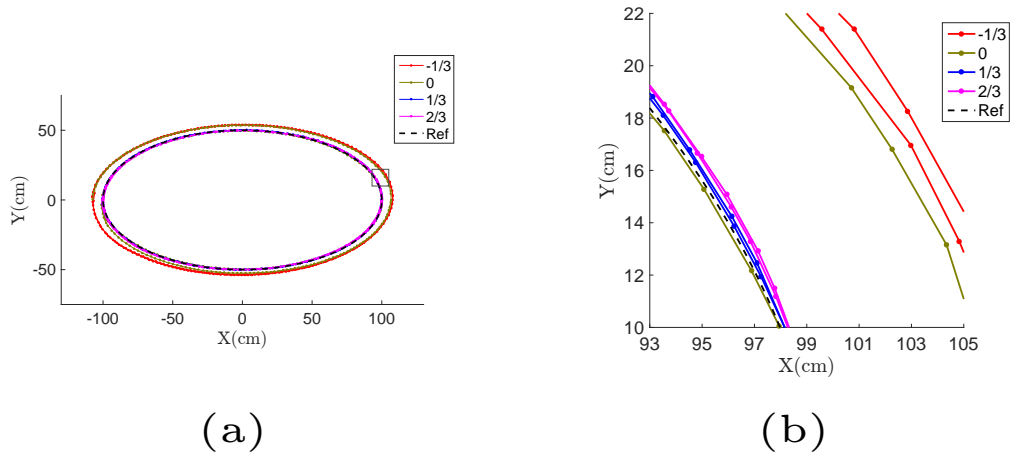


Figure 4.4: a) Dynamic simulation paths, with different reference speeds. b) Zoom in on the box in plot a. Simulations with positive  $\beta = 1/3, 2/3$  have no drift, less than 1 cm from the reference path. Simulations with constant speed and negative power law  $\beta = 0, -1/3$  drift outside of the reference elliptic path.

#### 4.4.1 Integration inside the Stack-of-Tasks framework

Fig. 4.1 present the architecture of the system. It shows that the "Vector Field" output is  $\mathbf{c}^*$ , but in fact the vector field only provides linear velocity assuming that the robot is heading forward. Hence, inside the vector field box there is a Proportional Integrative Derivative (PID) controller which track the orientation of the velocity vector. It tries to minimize the error  $e = \text{atan}(\frac{\dot{y}}{\dot{x}}) - \theta$ . With  $[\dot{x} \ \dot{y}]$  the linear velocity extracted from the vector field and  $\theta$  the orientation of the robot base. The singularities of the  $\text{atan}$  function is dealt with inside the PID controller.

The control period of the HRP-2 robot is 5 ms. 2 ms are consumed by the walking pattern generator, 1 ms by the QP solver and 1 ms by the robot low level controller which includes the stabilizer. The vector field has 1 ms left to be computed. Hence, in terms of computation time, we had to include efficient C++ code inside the Stack-of-Tasks.

SIM					
Ref. $\beta$	Sim. $\beta$	$R^2$	MSD (cm <sup>2</sup> )	A (m <sup>2</sup> )	T (s)
−0.33	−0.21	0.46	27.22	1.820	140.0
0	0.05	0.27	8.96	1.661	134.0
<b>0.33</b>	<b>0.33</b>	<b>0.99</b>	<b>0.05</b>	<b>1.573</b>	<b>130.0</b>
0.67	0.60	0.94	0.13	1.569	127.2
EXP					
Ref. $\beta$	Exp. $\beta$	$R^2$	MSD (cm <sup>2</sup> )	A (m <sup>2</sup> )	T (s)
−0.33	−0.23	0.70	53.13	1.888	150.4
0	0.02	0.03	30.51	1.812	142.4
<b>0.33</b>	<b>0.34</b>	<b>0.79</b>	<b>39.00</b>	<b>1.855</b>	<b>134.4</b>
0.67	0.57	0.90	46.11	1.884	133.6
REF			0	1.571	120.0

Table 4.1: Results from dynamic simulation (SIM) and actual robot experiment (EXP); for different reference power laws (Ref.  $\beta$ ), simulated power law exponent (Sim.  $\beta$ ) and actual motion power law exponent (Exp.  $\beta$ ) calculated using nonlinear regression [Maoz 2005] with  $R$  squared error ( $R^2$ ), mean squared distance to the reference path (MSD), area (A) and duration (T) of the generated ellipse trajectory are given, with those of the reference frame (REF). Geometrically, in simulation the one-third power law,  $\beta = 1/3$ , is most exact, and in experiment the constant speed was more exact. Temporally, higher  $\beta$  exponents yield faster motions for both simulation and experiment, but always slower than the reference.

## 4.4.2 Dynamic simulation results

### 4.4.2.1 Simulation and trajectory analysis

To simulate motion, we used the OpenHRP simulator, that computes the contact forces and HRP-2's rigid body mechanics, and includes a model of the compliance of the robot ankles. We implemented the control architecture depicted in Fig. 4.1 in OpenHRP. We analyzed the trajectories of the center of mass using MATLAB. To overcome the coronal swing motion we applied a procedure for finding the middle points of each sway. Each middle point was the average of two consecutive local signed curvature extrema with opposite signs, with time defined as the average of the times of these two points. We used the middle points trajectory for all analysis purposes. The values of the power law exponents  $\beta$  were calculated using nonlinear regression estimation [Maoz 2005]. Repeating the procedure with log – log linear regression yielded similar  $\beta$  values. Speed was extracted from the middle points trajectory using a noise-insensitive filter [Holoborodko 2008]. Curvature was extracted from the reference ellipse.

### 4.4.2.2 Power law patterns are reproduced

The theoretical speed profiles are compared with the one measured in simulation (Fig. 4.5, left) and the one measured from the motion capture system (Fig. 4.5, right). The simulation yielded positions of maximal speed slightly shifted with respect to maximal speed positions predicted by the power law based on the curvature of the actual path; for  $\beta = -1/3, 1/3, 2/3$  shifts along



the trajectory, of 22, 4,  $-15$  cm of the simulated with respect to reference speed peaks occurred. Unpredictably, constant speed  $\beta = 0$  power law yielded an oscillatory curvature-dependent speed profile.

#### 4.4.2.3 Drift correction by one-third power law

As predicted, the simulated one-third power law resulted in reduced drift compared to constant speed and other power laws. For  $\beta = 1/3$  the path was most similar to the reference path, with next best being the  $\beta = 2/3$  power law. The constant speed  $\beta = 0$  and  $\beta = -1/3$  yielded drifts; the simulated elliptic paths were larger than the reference elliptic path, as reflected by area and mean squared distance (see table 4.1 and Fig. 4.4). Interestingly, the constant speed  $\beta = 0$  path deviated from the reference frame gradually and not immediately upon movement initiation, as seen in Fig. 4.4.b.

#### 4.4.2.4 Increase in $\beta$ exponent decreases duration

Simulated motions took more time than the reference time, that was always two minutes per lap. The power law exponent affected movement duration; the higher the  $\beta$  the faster the motion, so its duration was closer to the reference behavior (table 4.1).

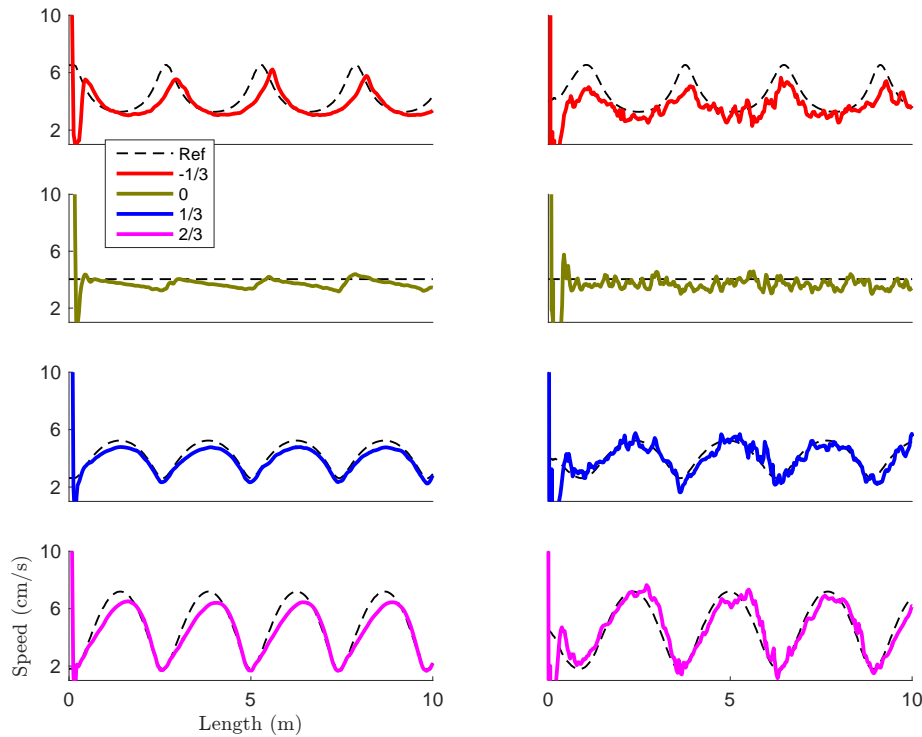


Figure 4.5: Speed profiles for dynamic simulations (Left) and robot experiment (Right), with four different reference speeds (Ref). Both simulated and actual robot motions reproduce the spatio-temporal power law patterns.

### 4.4.3 Experimental results

#### 4.4.3.1 Experimental setup and trajectory analysis

We tested the four power laws with the actual HRP-2 robot; the robot started standing approximately 60 cm behind the tip of the ellipse, and walked until completing two laps. We used center of mass trajectories measured with the motion capture system, low pass filtered at 0.1 Hz. Analysis was identical to that of simulated motion.

#### 4.4.3.2 Power law patterns are reproduced

The reference speed power laws were noisily reproduced by the robot (see table 4.1 for  $\beta$  values and their  $R^2$  errors, and figure 4.5 for speed profiles).

#### 4.4.3.3 Geometrical drift does not fully vanish

Overall, experimental results showed larger drifts than simulation. The constant reference speed resulted in the lowest drift (table 4.1), outperforming the positive  $\beta$  power laws. Different convergence trajectories (see figure 4.3), may arise from different initial positioning.

#### 4.4.3.4 Increased $\beta$ decreases duration

The actual movement of the robot took longer time than simulation. The trend we reported for the simulations; the higher the  $\beta$  the faster the motion, was fully reproduced in the experiment.

#### 4.4.3.5 Evaluating controller corrections

Ref. $\beta$	-0.33	0	0.33	0.67
Norm (m)	1.416	0.950	0.642	1.124
Orientation (deg)	76.83	89.60	60.45	77.28
Force ( $kN \times s$ )	21.93	23.54	19.80	21.01

Table 4.2: Analysis of the odometry frame and forces. For each of the four power laws the distance from center (Norm), body orientation (Orientation) and integrated norm tangential force measured on ankle (Force) are given. The one-third power law is outperforming all other power laws.

To estimate the feedback correction done by the vector field and the PID controllers, we analyzed the internal odometry frame of the walking pattern generator. For each reference power law we examine the last crossing point of center of mass trajectory with the  $x$  axis. We measured deviation from the baseline, the planned position and orientation, to gain error values reflecting the accumulated error along the two laps, caused by sliding. Additionally we examined tangential forces on the ankle, indicating the amount of compensated sliding. The one-third power law showed smaller accumulated errors, compared to the other power laws (table 4.2). Less error needing compensation implicates lower burden on the hardware.

## 4.5 Conclusions

In simulation and experiment, we tested how stable generation of power laws may help humanoid robot walking. The simulations and experiments reproduced the reference power law’s temporal patterns. The one-third power law, used by humans, appears beneficial for drift reduction. The one-third power law reduced drift in simulations and also resulted in less need for sliding compensation in actual robot motion. Finally, our simulations and experiments showed that using higher  $\beta$  exponent power laws allows faster movements.

Having shown the importance of using a path-adjusted speed profile, it is still not fully determined how to select an optimal speed profile for stabilized and accurate robotic locomotion, given a planned reference trajectory. The different perspectives describing how the human motor system selects movement speed, based on either optimization [Flash 1985, Todorov 2002] or geometric invariance [Flash 2007, Bennequin 2009], may prove beneficial for this aim.

Except for systematic examination in locomotion of the benefits of different speed regularities, we suggest two additional applications of the speed modulation according to the geometry of the reference trajectory. One is to use power laws as a constraint for kino-dynamic motion planning [Pham 2009], the other is to find bounds on curvature when planning the motion of the free-flyer [Orthey 2015a].



# Learning Movement Primitives for the Humanoid Robot HRP2

---



Skilled human behavior is highly predictive and relies on predictive planning. It adapts its behaviors according to task constraints with relevance for motor behaviors in the future. Recent work in predictive control has shown great promises for generating complex behaviors [Koenemann 2015]. They are however important combinatorial challenges. In this chapter we propose a control framework combining biologically-inspired online planning for the robot upper body and model predictive control for the robot lower body. The two controllers are linked together using a model predictive control method. The computational foot print can be compared to the model predictive control presented in [Herdt 2010a]. The human-like upper body motion is generated via a network of coupled dynamical movement primitives. This network is embedded within a nonlinear dynamical system that generates coordinated behavior. The lower body is controlled by the walking pattern generator presented in Chap. 1. This combination ensures the flexibility of the system as well as the safety of the robot.

This chapter is structured as follows: we will present the motivation behind this work in Sec. 5.1. A quick overview of related works in areas of computer graphics and humanoid robotics is given in Sec. 5.2. Then, in Sec. 5.3 we will explain successively the top down approach, the bottom up approach and the overall architecture of the controller. Finally, in Sec. 5.4 some preliminary results are presented, which demonstrate the feasibility of the approach in simulation and on the real HRP-2 robot. This work is done in the frame of a collaboration with our colleague, experts in human motion analysis, in the KoroBot project. This chapter is based on the submitted paper [Mukovskiy 2016]. The video of the application can be found at [https://youtu.be/3LT\\_QEiZzSo](https://youtu.be/3LT_QEiZzSo).

## 5.1 Motivation

The modeling and the synthesis of online-reactive multi-action sequences are important topics both in computer graphics and in humanoid robotics. The most challenging part of the online control of complex whole-body behaviors is the coordination of the different tasks.

In the current work we present a novel approach that combines the walking pattern generator (WPG) presented in Chap. 1, with an online kinematic planning architecture for full-body movements. The upper body motion pattern generator is based on learned dynamical movement primitives. And the lower body motion generator is based on model predictive control. The approach is suitable for the online planning and control of reactive behaviors that integrates locomotion and reaching. It also includes highly flexible re-planning even in short time horizons. The locomotion planned by the WPG is combined with the planned motion of the upper body using the dynamic filter presented in An. A. This ensures that the overall behavior results in a dynamically stable gait with the predefined CoM velocities and predefined upper body motion. The proposed method is characterized by a much smaller computational complexity than a direct motion planning using complex dynamical model. The global computational complexity can be compared with the one of standard WPG algorithms [Herdt 2010b].

## 5.2 Related work

The general problem of motion generation in a dynamic environment is challenging. The continuous change of the environment and error in state estimations implies a relatively high

control frequency, typically at least 10 HZ. It also implies a reasonable preview horizon duration, typically at least 2 steps or around 1.6 s for a human size robot. Solving a model predictive control problem for a humanoid robot with 30 degrees of freedom in a horizon of 1.6 s is still an open issue. This implies that a priori knowledge or approximation of the problem lowering the complexity is needed.

Current solutions range from near real-time whole body Model Predictive Control with regularized modeling of contacts in order to decrease the associated computational cost [Tassa 2012, Koenemann 2015]. To a precise modeling of contact phases, which requires hours of offline computation time [Koschorreck 2012].

Another challenging issue for the generation of human-like behaviors is the sequential planning of multi-step sequences, where individual steps can be associated with different sub-goals or constraints (like contact with goal objects or step-length constraints). This problem being multidisciplinary, we quickly review associated work in computer graphics, biological motor control, and humanoid robotics.

### 5.2.1 Modeling of whole-body movements in computer graphics

The problems of kinematic synthesis of such complex whole body movements has been addressed extensively in computer graphics, e.g. [Levine 2012], and many learning-based approaches have been proposed that provide low-dimensional parametrization of classes of whole body motion [Wang 2008]. Recently, more attention has been given to methods for the blending of learned motion primitives, whose concatenations over time have to satisfy additional task constraints. For example, in [Feng 2012] captured motion samples were blended exploiting a prioritized 'stack of controllers'. In [Shoulson 2014] the instantaneous blending weights of controllers were prioritized by their serial order. In [Huang 2014] the coordination between locomotion and arm pointing in the last step was modeled by blending and selecting arm pointing primitives dependent on the gait phase.

### 5.2.2 Biological motor control of multi-step sequences

Human motor behavior including action sequences has been shown to be highly predictive. This has been investigated, for example, in a recent study on the coordination of walking and reaching [Land 2013]. Human subjects had to walk towards a drawer and to grasp an object, which was located at different positions in the drawer. Participants optimized their behavior already multiple steps before the object contact, consistent with the hypothesis of *maximum end-state comfort* during the reaching action [Weigelt 2010, Rosenbaum 2008]. This means that the steps prior to the reaching were modulated in a way that optimizes the distance for the reaching action [Land 2013]. In [Mukovskiy 2015] our partners have proposed a learning-based framework that is based on movement primitives that are learned from motion capture data, and which reproduces these human planning strategies for an application in computer animation. The underlying architecture is simple and approximates complex full-body movements by dynamic movement primitives that are modeled by nonlinear dynamical systems. These primitives are constructed from kinematic primitives that are learned from trajectory sets by anechoic demixing. Similar to related approaches in robotics [Gams 2008, Buchli 2006], the method generates complex movements by the combination of a small number of learned



dynamical movement primitives. Our partners have previously demonstrated the advantages of this approach for the adaptive online generation of multi-step sequences with coordinated arm movements [Mukovski 2015, Giese 2009a].

### 5.2.3 Related approaches in humanoid robotics

Several approaches have been proposed in robotics for the synthesis of walking combined with grasping movements. Indeed, the DARPA robotic challenge valve manipulation task forced the researchers to find efficient and robust methods to perform reaching and manipulation tasks. [Ajoudani 2014] proposed a hybrid controller, where the robot is using a goal-driven fast foot step planner in combination with visual servoing for the reaching and grasping of the valve. [Kuindersma 2015] proposes a complete control architecture for the humanoid robot Atlas that is able to localize the robot and automatically finds foot step around or over obstacles in order to reach a user defined goal. The architecture contains also a whole-body controller which allow the robot to get up from a lying down position, or to do complex task like turn the valve. Another team (IHMC) presented an architecture in [Johnson 2015] with a more sophisticated control of the locomotion in [Englsberger 2014]. All three mentioned control architectures can make a humanoid robot reach, and then grasp or manipulate objects as required for the robotic challenge. To our knowledge an online simultaneous coordination of both tasks has not been demonstrated so far. Other solutions for the combination of walking and vision-controlled reaching of a static and mobile target during walking were proposed in [Stasse 2008] and [Brandao 2013].

Randomized motion planning methods allow the generation of complex whole body motion in constrained environment but at a high computational cost [Dalibard 2013, Kanoun 2011]. [Gienger 2010] proposed an algorithm for the computation of optimal stance locations with respect to the position of a reaching target, where a dynamical systems approach was used to generate the reaching behavior. [Yoshida 2007] used a task priority approach, based on a generalized inverse kinematics, in order to organize several sub-tasks, including stepping, hand motion, and gaze control. Other work has exploited global path planning in combination with walking pattern generators (WPGs) [Kajita 2003a] in order to generate collision-free dynamically stable gait paths.

A first attempt to transfer human reaching movements to humanoid robots by using motion-primitives was proposed in [Taïx 2013]. In this work the primitives were extracted by using PCA and the behavior was successfully implemented on the HRP-2 robot, only involving the trunk and arm joints. The use of motion primitives in robotics was also proposed in [Gams 2013], also including the integration of force-feedback. [Ijspeert 2013] and [Ajallooeian 2013] proposed systems based on dynamical movement primitives that can be modulated in real-time for the generation of complex movements. However their approach focus on learning the in an efficient way the input data. In [Ijspeert 2013] the transfer to robot is tackled by using a low level torque controller managing the feasibility of the system. This could result in a different motion than the one learned. So the transfer from balanced human locomotion to balanced robot locomotion is still an open question.

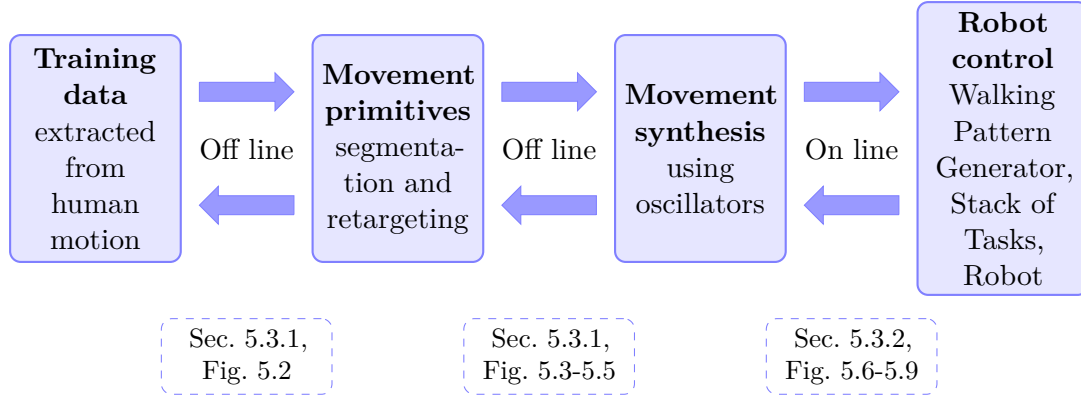


Figure 5.1: This graph represents a general overview of the system architecture

### 5.3 System architecture

The global architecture of the system is depicted Fig.5.1. It is composed by two processes done offline. The first is the data acquisition of human motion and the second is the segmentation and retargeting of this motion to be played on humanoid robots. Then two blocks work in real time. The first is the generation of the learned retargeted motion and the second is the controller of the humanoid robot. More details are provided in the following section.

#### 5.3.1 Human data

##### Drawer opening task

The modeling of the coordination of walking and reaching was based on a motion capture data set from humans who opened a drawer. The participants walked towards a drawer, opened it with their left hand and reached for an object inside with their right hand. The initial distance from the drawer and the position of the object inside were varied [Mukovskiy 2015], see Fig. 5.2. For more detail the reader is kindly asked to read [Mukovskiy 2016]. A video of the motion is available here <https://tinyurl.com/he3dnh2>. The motion is cut into three different gait :

1. normal walking,
2. adaptive steps, i.e. walking motion plus reaching the drawer handle with the left hand,
3. grasping, i.e. stop walking plus reaching the target with the right hand.

##### Learning of the kinematic primitives

In order to learn low-dimensional representation of every individual segmented motion we apply the anechoic demixing algorithm [Omlor 2011, Chiovetto 2013]. In this approach the joint angle trajectories are learned unsupervisedly as an *anechoic mixture model*:

$$\underbrace{\xi_i(t)}_{\text{angles}} = m_i + \sum_j w_{ij} \underbrace{\sigma_j(t - \tau_{ij})}_{\text{sources}}$$

Here, for each angle trajectory  $\xi_i(t)$  is represented as the linear mixture of  $j$  source signals  $\sigma_j(t)$  with the linear weights  $w_{ij}$  plus the angle mean value  $m_i$ . These source signals can be

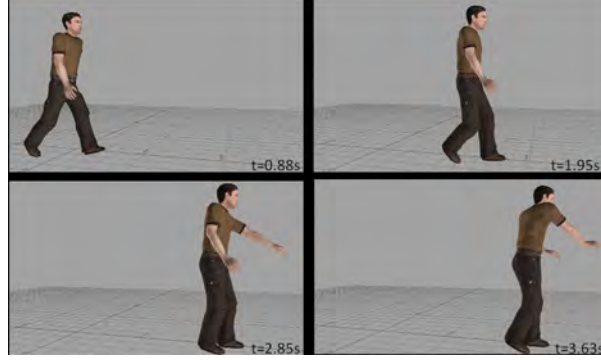


Figure 5.2: Illustration of important intermediate postures of the human behavior: step with initiation of reaching, standing while opening of drawer, and reaching for the object.

delayed individually with time delays  $\tau_{ij}$ , which are different for different angles and source components. Very good approximations can be computed with less than 4-5 learned source functions [Giese 2009a]. In our standard implementation we first extract the mean angle values from our data. Then we estimate the weights of additional non-periodic source component, which shape is usually given, but not inferred from the data. The shape function serving as nonperiodic source is taken as follows:  $s_0(t) = \cos(\pi t/T)$ , where the time length of trajectory samples (and the period of periodic sources) is  $T$ . For the first gait three periodic sources and a non periodic one were sufficient. For the other two gait, the four sources did not provide a good enough approximation quality. Therefore additional two sources were learned from the difference between the original trajectories and the learned one using three periodic sources and one non periodic. Such constrained step-wise approach simplifies the blending between different motion styles, since then the delays of the sources are identical over styles. The resulting shapes of learned source functions are presented in Fig. 5.3.

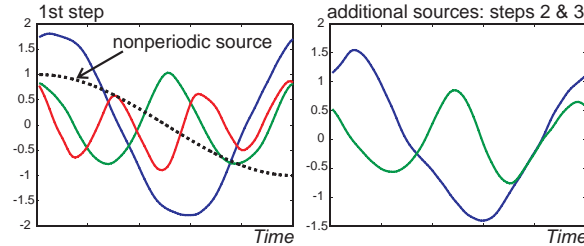


Figure 5.3: Extracted source signals.

### Online kinematic motion synthesis of multi-action sequences

In order to make the rendering of whole-body trajectories online we propose an architecture built upon the autonomous dynamical systems regarded as central pattern generators (CPGs), [Giese 2009a]. By this approach, our kinematic primitives are generated online by dynamical systems, or dynamic primitives, DMPs [Buchli 2006, Ijspeert 2013]. For this we map the solutions of the dynamic primitives onto source signals by Support Vector Regression (using a Radial Basis Function kernel and the LIBSVM Matlab<sup>®</sup> library [Chang 2001]). The resulting architecture is

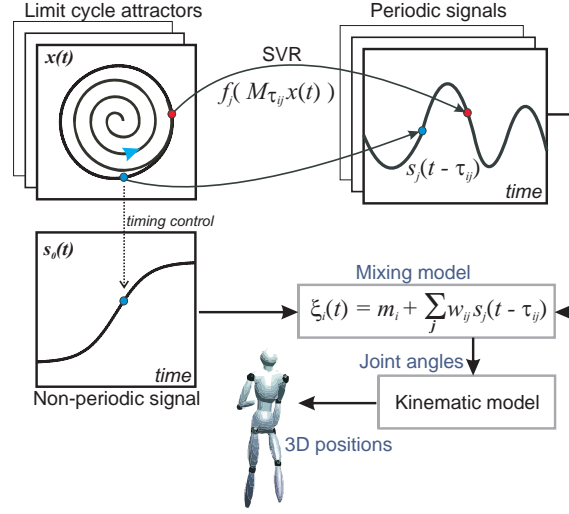


Figure 5.4: Architecture for the online synthesis of body movements using dynamic primitives, [Giese 2009a]. This architecture is fused into one block named **Kinematic Pattern Synthesis** in Fig. 5.6

summarized at Fig. 5.4.

For the periodic DMPs we chose a limit cycle oscillator (Andronov-Hopf oscillator) as canonical dynamics. It can be characterized by the differential equation system (with  $\omega$  defining the eigenfrequency), for the pair of state variables  $[x(t), y(t)]$ :

$$\begin{aligned}\dot{x}(t) &= [1 - (x^2(t) + y^2(t))]x(t) - \omega y(t) \\ \dot{y}(t) &= [1 - (x^2(t) + y^2(t))]y(t) + \omega x(t)\end{aligned}$$

The online phase-shifting is modeled as an additional rotation of the oscillator phase plane, such that, due to the circular shape of the attractor limit cycle of Andronov-Hopf oscillator, the trajectories on the attractor exhibit simple time-delays, c.f. [Giese 2009a]. The instantaneous phase of the leading DMP also controls the start-triggering event for the non-periodic source. All the periodic DMPs are phase-coupled in order to assure the coordinated globally stable dynamics. The methods of coupling was designed based on Contraction Theory, [Park 2009b].

To model the actions with variable styles we do learning of nonlinear mappings between task parameters and action parameters (the sources weights). This mapping is learned by Locally Weighted Linear Regression method (LWLR), [Mukovskiy 2015], and the relevant task parameters are steps lengths and timings. For the synthesis of multi-step step sequences the step lengths are computed from the current estimated target distance. The simplified strategy taken for the online synthesis is the following: the step lengths in the first action can be modified in range of training data (additional steps can be introduced by higher level supervision algorithm, if target distance is too large); the step length in the second action is computed to realize maximum comfort distance for reaching. The smooth interpolation between the morphable weights of the kinematic primitives at the moments of concatenation of action is described with all technical details in [Mukovskiy 2015].

## Processing

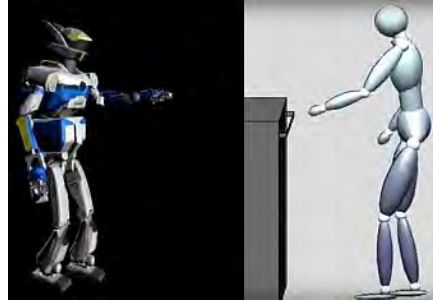


Figure 5.5: The result of the retargeting of the drawer opening task onto the unconstrained skeleton of the HRP-2 robot. The movie is available here [<http://tinyurl.com/j8qnbtp>]

Once the recorded motion is learned, it is animated in MotionBuilder (Autodesk), using an 'actor' puppet whose geometric parameters were adapted to the recorded subject. The trajectory was first normalized in duration and mapped to the robot using the Denavit-Hartenberg (DH) convention. The joint limits are not taken into account at this stage. Fig. 5.5 is a frame of the movie [<http://tinyurl.com/j8qnbtp>]. It shows the animated HRP-2 in MotionBuilder (Autodesk) before kinematic re-targeting. At this point the trajectory are still not feasible (see Sec. 5.4). After that the trajectories are kinematically re-targeted using inverse kinematics and scaled down to fit HRP-2 kinematics and dynamics constraints. The details of this process is in Sec. 5.4. The articular trajectory obtained are then learned using the architecture depicted above and in Fig. 5.4.

### 5.3.2 Robotics Implementation

#### Walking pattern generator

In this application we used the pattern generator described in Chap. 1 as well. The interest of this choice is that the lower part of the robot will be driven by the learned human CoM velocity and pelvis rotation. Moreover the walking pattern generator being a bottom up approach and a well tested algorithm in the humanoid robotic platform HRP-2 at LAAS-CNRS we can warranty the safety of the robot in terms of auto-collision and balance. The other main advantage is that the implementation include the dynamic filter, often seen as a kind of Newton-Raphson iteration [Stasse 2013]. In other word we can see the upper body motion as perturbation on the robot dynamic and cancel it via the use of the dynamic filter. For more detail about the dynamic filter itself the reader is kindly ask to refer to An. A.

### 5.3.3 Overall architecture

In the following we give the brief overview of the proposed robotics platform implementation (see Fig. 5.6). The module labelled 'Kinematic Pattern Synthesis' is the system described in the previous Subsec. 5.3.1. This module computes the upper body trajectory and the pelvis linear and angular velocity. The walking pattern generator than computes the CoP and feet trajectory and send them to the dynamic filter. In turn, the dynamic filter computes the correction to apply to the CoM dynamics form the CoM, the feet and the upper body trajectories. The

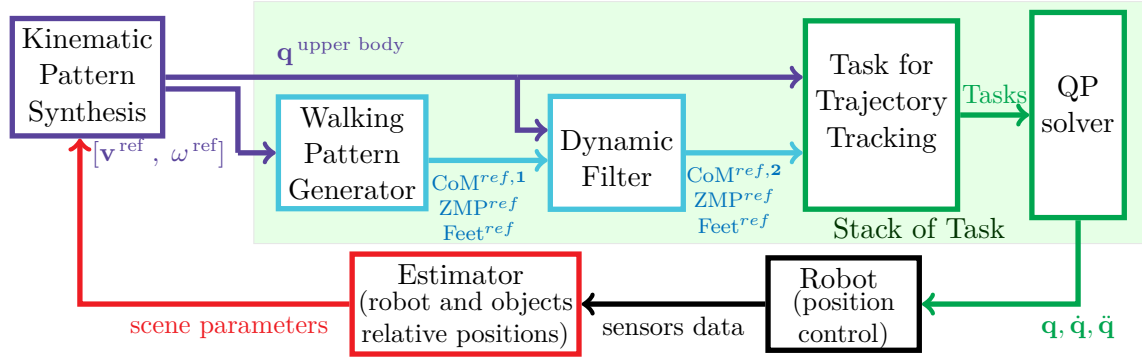


Figure 5.6: This scheme describe the feedback loop used to control the humanoid robot HRP-2. With  $[v^{ref}, \omega^{ref}]$  as respectively the linear and angular velocity and  $q^{upper body}$  the upper body joint trajectories computed from the kinematic pattern synthesis.  $q, \dot{q}, \ddot{q}$  are respectively the generalized position and velocity vectors computed using the Stack of Tasks (SoT).

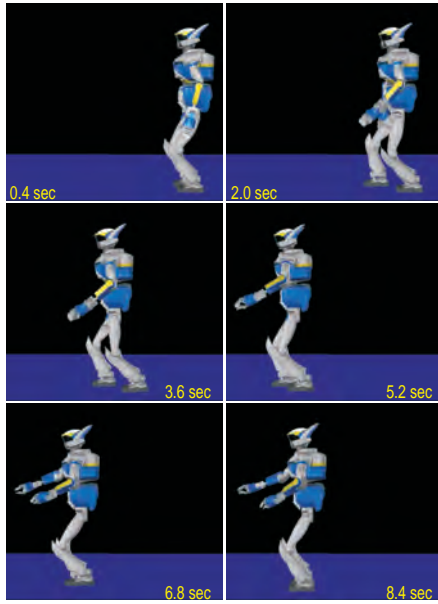
whole body trajectory is then computed by a generalized inverse kinematics using the corrected CoM, the feet, and the upper body articular trajectories. The framework used is called the 'Stack-of-Task' a.k. SoT. Once the whole body trajectory computed the robot track them and an estimator evaluates the relevant state of the robot and the scene parameters. The 'Kinematic Pattern Synthesis' use these data to recompute another set of upper body trajectory and pelvis velocities. This architecture has not been tested online yet. However the offline movement ran on HRP-2 indicates that this approach is feasible. In the next section we will discuss in more details the obtained results during the simulations and experiments.

## 5.4 Results

In this section we will discuss the results obtained from experiments. We will firstly introduce the setup of the experiment, then we will present the first results using the upper body trajectories re-targeted using kinematics. The discussion will be about the bottom heck of this approach and which possible solution exists. We will therefore explain in the next paragraph which solution we chose.

### 5.4.1 Experimental setup

The synthesis architecture described in Sec. 5.3.2 was first tested open loop control in simulation using the OpenHRP simulator and the HRP-2 robot model. In these simulations, the robot starts from the parking position and makes a transition to a normal step. At the end of this step the linear and angular pelvis velocities were determined and used as initial conditions. At the end of the last action a spline interpolation of pelvis angular and positional coordinates was used to change the robots state back to the parking position (introducing two additional steps on the spot). A snapshot of the executed behavior is presented in Fig. 5.7a. The drawer was not used here because the robot needs more teaching trajectories to really execute the task. In this chapter we show a proof of concept concerning the architecture of control.



(a) Off-line synthesised trajectories generated with the OpenHRP simulator.



(b) Real HRP-2 robot performing walking-reaching sequences.

Figure 5.7: Experiment using the HRP-2 in LAAS-CNRS

#### 5.4.2 Kinematic re-targeting

As explain above the re-targeting process is divided in two steps, first the human body trajectory are scaled and mapped on the robot joint, and second the inverse kinematics is used to respect kinematic constraints. This kinematic mapping is not sufficient to make HRP-2 walking. Indeed we can see in Fig. 5.5 that the feet are not flat on the ground while satisfying the joint limits. An additional problems are the numerical singularities and discontinuities in the articular trajectories. The former forbid the use of stabilizers using fast inverse kinematics like the commercial one implemented on HRP-2 by Kawada. The latter is incompatible with the dynamic filter as the articular trajectory has to be at least  $\mathcal{C}^2$  (see An. A).

A final point concern the dynamics. In fact the kinematic re-targeting concern only the kinematics, there are no notion of balance. This could result in a CoP out of the support polygon.

One possible workaround is to develop an optimization problem taking into account the whole body dynamics, like in [Ramos 2015]. The idea is to find a feasible solution for the upper body knowing the lower body dynamics which is a simple linear inverted pendulum dynamics. Those trajectories would then be consistent regarding the robot dynamics as well as the closest possible to the referenced human motion.

Another possible workaround is when a good knowledge of the system is available. One can find a morphing criteria on the trajectory so that the dynamic of the linear pendulum is not too much perturbed. Usually the heavy body velocities and acceleration heavily impact the robot dynamic. So scaling the trajectory of these bodies reduce the impact on the dynamic. This allow the dynamic filter to create dynamically consistent trajectory in only one iteration.

### 5.4.3 Experimental results

For the experiment the trajectories were re-sampled, resulting in a normalized duration of 1.6 sec for each action. The data was split into two subsets, separating the stored pelvis trajectories and the upper body articular trajectories. The pelvis position trajectories were rescaled, ensuring the maximally admissible velocity for the HRP-2 robot (0.5 m/sec). The pelvis linear and angular velocity was used as input to the walking pattern generator.

For our application we did not implement an optimization problem because of a lack of time. Indeed designing an optimization problem fast enough to keep the real-time aspect of the architecture is quite a challenge. Moreover we just wanted to make a proof of concept regarding the feasibility of the global approach. From experience we know that HRP-2 main weights are in its waist and chest. The two bodies are joint with two degrees of freedom, pitch and yaw. The yaw motion is very problematic as it creates angular momentum around the vertical axis. In fact, the dynamic filter does not compensate for such momentum. Hence we decided to scale the yaw angle between the chest and the waist of the robot. For compensation, a fraction of the yaw-angle trajectory was added back to the trunk yaw-angle. After this compensation, customized inverse kinematics methods were applied to correct the upper body arm reaching motion in order to satisfy joint limit constraints.

For the final application on the real robot HRP-2 we filtered the upper body trajectories using a Savitzky-Golay Filter. The result is at least a piece-wise  $\mathcal{C}^2$  polynomial trajectories with no time delay.

After training, for the learned parameters, the system generates very natural-looking coordinated three-step sequences for total goal distances between 2.34 and 2.94 m. This is illustrated by <http://tinyurl.com/jtkc6g7>. If the specified goal distance exceeded this interval, the system automatically introduced additional gait steps, adapting the behavior for goal distances above 3 meters. <http://tinyurl.com/zu55rox> presents two examples of generated sequences for larger goal distances.

The high degree of real-time online adaptivity is demonstrated in the <http://tinyurl.com/hnxmluuk>. When avatar approaches the target, and during the second gait cycle the target jumps away towards a more distant position, where it can not longer be reached with the originally planned number of steps, the online planning algorithm automatically introduces an additional steps and adjusts the others, so that the behavior can successfully be accomplished.

A movie of the full 3-action sequence is presented in <http://tinyurl.com/jfda5ql>, and a movie showing a 4-action sequence can be found in <http://tinyurl.com/j7dobcn>. As final step, the architecture was also tested using the real HRP-2 robot, see Fig. 5.7b.

The captured movie of the 3-action sequence realized on the real HRP-2 robot is presented in <http://tinyurl.com/hfyhmv6>, and a demo of a 4-action sequence is shown in <http://tinyurl.com/j52c8dz>.

#### 5.4.3.1 Feasible motion

The experiment has been successfully performed 5 times in a row. The forces measured on the vertical axis ( $z$ ) are depicted in Fig. 5.8. The maximum force is less than 700 N which is safe for the robot. For comparison the robots weight is around 56 kg, hence the forces applied to the feet in static posture is :  $56 * \text{gravity} = 56 * 9.81 = 549.36 \text{ N}$ . The impact forces are less than 20 %



higher than the weight of the robot. If this ratio is higher than 45 % the robot force sensors may break. To summarize, this motion is safe to be performed on our humanoid robot HRP-2.

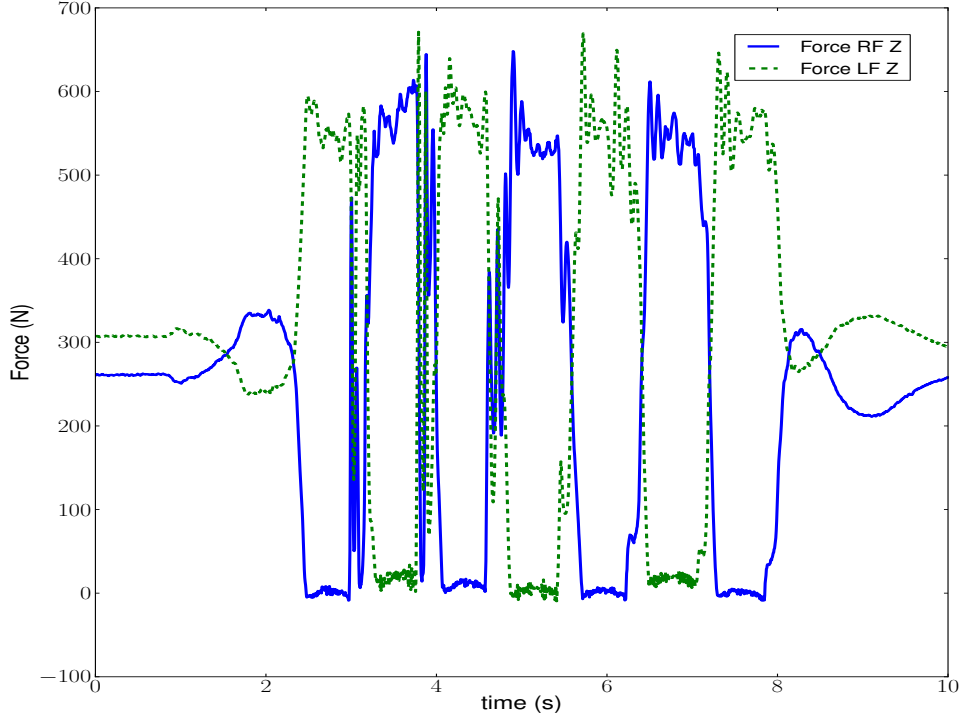


Figure 5.8: Forces on the vertical axis ( $z$ ) measured during the experiment.

#### 5.4.3.2 The role of the dynamic filter

In Fig. 5.9 we can see the effect of the dynamic filter on the  $CoP_{mb}$ . The graphs show the comparison between :

- the referenced CoP computed from the linearized inverse pendulum model ( $CoP$ , plain blue),
- the CoP computed from the multibody dynamics ( $CoP_{mb}$ , dotted green),
- the CoP computed from the multibody dynamics after the dynamic filter correction of the whole body ( $CoP_{mb\,fil}$ , dotted red),
- and the CoP computed from the multibody dynamics after the dynamic filter correction of the lower body only ( $CoP_{mb\,fil\,lb}$ , dotted magenta).

The first graph represent the trajectories on the sagittal plane (axis  $x$ ). And the second graph on the coronal plane (axis  $y$ ).

The average and maximum distance between the reference and the non corrected multibody CoP are :

$$mean||CoP_{mb} - CoP|| = 0.028\,m, \quad max||CoP_{mb} - CoP|| = 0.053\,m$$

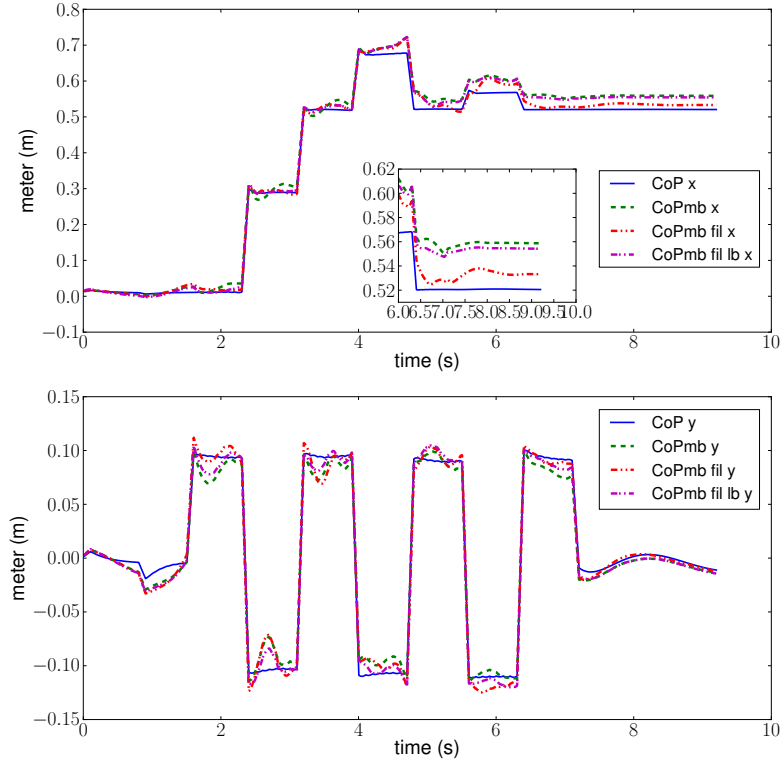


Figure 5.9: These graphs show the theoretical behavior of the CoP. In blue there is the referenced CoP computed from the linearized inverted pendulum model. In green you can find the equivalent CoP but computed using the whole body model. We call it the multibody CoP ( $CoP_{mb}$ ). In red there is the  $CoP_{mb\,fil}$  computed after the dynamic filter correction. And the magenta line is the CoP with only the lower body filtered ( $CoP_{mb\,fil\,lb}$ ). In the upper graph a zoom on the end of trajectory shows the efficiency of the dynamic filter.

The average and maximum distance between the reference and the corrected multibody CoP are :

$$mean||CoP_{mb\,fil} - CoP|| = 0.015\,m, \quad max||CoP_{mb\,fil} - CoP|| = 0.052\,m$$

Statistically the filtered  $CoP_{mb\,fil}$  is closer, in  $L_2$  norm, to the reference than the  $CoP_{mb}$ . This correction makes the difference between balanced and unbalanced motion. Indeed we tested the motion using the dynamical filter only for leg motion (static upper body) and without the dynamic filter in simulation. In this contexts the robot succeeded to walk until the end of the motion but fall down when converging toward a static posture. In fact the robot arm are lifted to reach the target. Therefore the center of mass is leaning forward and the CoP comes closer to the foot edges (see the zoom graph in the first graph of the Fig. 5.9). The flexibility under the robot soles amplify this phenomena and the CoP goes beyond the the edge of the feet. Hence the robot falls. The maximum errors are not so different. It is due to the last step forward motion where the robot decelerate and point its arms forward. The motion itself make the  $CoP_{mb}$  moving away from the reference. The dynamic filter is not able to fully compensate for that much perturbation in a short time. It will rather decrease the perturbation step by step. We can see in Fig. 5.9 that at the end the next step (5 s) the  $CoP_{mb\,fil}$  (in red) is back to its

reference. As a conclusion, the whole body motion has to be taken into account by the dynamic filter to generate dynamically feasible motions.

## 5.5 Conclusions

In this chapter, we have presented an architecture that combines a flexible online generation of coordinated full-body movements using dynamic movement primitives with a control architecture that is based on a walking pattern generator, which exploits nonlinear model predictive Control. The proposed architecture is suitable for the planning of complex coordinated full-body movements in real-time, and generates dynamically feasible behavior of the robot with appropriate balance control during walking. The high computation speed distinguishes the proposed framework from other approaches, which exploit optimum control for the synthesis of dynamically feasible complex full-body movements. The functionality and flexibility of the proposed architecture was demonstrated by simulation using the OpenHRP physics simulator and also on the real HRP-2 robot.

The shown results represent first feasibility tests for this type of architecture. Future work will have to extend the training sets by inclusion of training sets that maximize the parameter variation of each individual action, and which include only dynamically feasible behaviors, generated with the robot simulator. This will make the planned trajectories more similar to dynamically feasible behaviors and in this way might further increase the flexibility and computational efficiency of the proposed architecture. However it is, for the moment, limited to plain ground walking. As further work we may extend our architecture to multicontact locomotion for humanoid robots. This would require the use of another filter for correcting the 3D CoM trajectory. To our knowledge, no such model predictive control exists yet.



# HRP-2 as Universal Worker Proof of Concept

---



This chapter presents a technical integration in a test scenario from Airbus/Future of Aircraft Factory. It is a collaboration between the LAAS-CNRS laboratory and the Airbus french industry (see [Stasse 2014]). In this chapter, we present a preliminary proof of concept (PoC) aiming at introducing humanoid robots in an aircraft factory. The PoC was aiming at demonstrating the capacity of HRP-2 to deal with three aspects needed in a factory: reactivity to change in the environment, visual feedback and on-line motion generation. The limits reached in this PoC are here highlighted to draw some research direction focused on the needs of Aircraft manufacturers. The video of the application can be found at <https://youtu.be/iFV-13XlJvI>.

## 6.1 Fast re-planning for moving obstacle avoidance

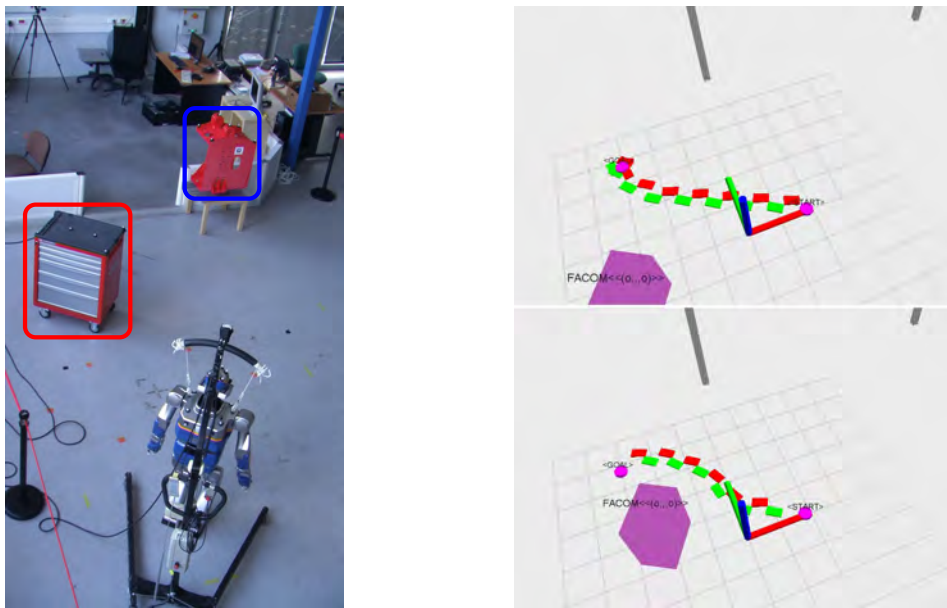


Figure 6.1: Situation of the experiment: The robot has to go in the vicinity of the pylon engine depicted in blue in the left picture while avoiding the red moving toolbox. The top right picture show the first planning and the bottom right the re-planned trajectory after the tool box got into the way.

The setup of this scenario is depicted in Fig. 6.1. The robot has to go toward a pylon engine (in the blue rectangle in Fig. 6.1) and do some screw motion. The pylon engine is the mechanical part connecting the aircraft engine and the wing. Using information provided by a motion capture system the robot is able to plan autonomously footsteps from its current location to the pylon engine. A human may move the toolbox (in the red rectangle in Fig. 6.1) such that it crosses the path of the robot. The robot is then able to change its footsteps and avoid the toolbox.

The robot is searching over a set of pre-defined action which are known to be feasible. The small foot-print of the action set allows for real-time planning search over a cost-function. The cost function includes a metric from the starting point to the robot current location and a metric from the robot current location to its final goal. This solution is currently based on the family of A\* algorithms as proposed in the following papers [Chestnutt 2010, Kuffner 2002, Hornung 2012]



*Figure 6.2: Robot tracking the engine pylon using a motion capture system*

with demonstrations on various humanoid robots such as ASIMO, HRP-2 or NAO. The method here is based on [Perrin 2012]. More precisely, from a set of quasi static half-steps, the robot trajectory is speed up using an analytical pattern generator coupled with the collision detection algorithm called PQP to test if the trajectories are without collision.

The robot is able to plan, in less than  $2.4\text{ s}$  (3 steps), a path from its current location to the final one as depicted in the right side of Fig. 6.1. If the toolbox is put on the robot path and if the robot is three steps away it can avoid it. The three steps are a limitation related to the pattern generator which needs this information on the future. This experiment has been performed in 2003 and it was quite difficult to find in real-time a full whole-body trajectory which avoid obstacles and maintain the robot balance. For this reasons we develop the algorithm presented in Chap. 1. The goal of this experiment is to show the achieved reactive capabilities of the system in this specific context.

A\* approaches are using a limited set of actions to simplify the problem solving. However in situation a bit more complex the robot tends to make unnecessary long sequence of steps because it is exploring only this limited sequence of actions. This was the main point of using a more aggressive approach proposed in [Perrin 2012]. In order to adapt the plan more efficiently, the system would need a rather different control system for balancing. This is the subject of the second experiment.

## 6.2 Reactive walking pattern generator

The balance control law of the robot takes as an input a velocity reference, and the system try to find footsteps such that the robot Center-Of-Mass is following as much as it can the velocity reference. In this PoC, two ways were tried to compute a reference velocity: visual servoing, and Euclidian distance between the object pose and the robot pose using a Motion Capture system. A more detailed description of the balance control law developed in the context of the French Research Project R-Blink is available in [Herdt 2010b]. A first experiment with this setup was realized in [Dune 2011].

In Fig. 6.2 the robot is able to follow the position of the engine pylon given by the motion capture system in the frontal plan. This library was successfully used for the experiments described in [Dune 2011]. It was interesting to test it on a different HRP-2 to check the portability of the software.





*Figure 6.3: HRP-2 doing a whole body screwing motion holding a 3D printed industrial screwdriver. On the left side one can see the visual servoing done on a mockup of the engine pylon. On the right side the robot is controlled via motion capture data as the visual servoing did not work on the 3D printed engine pylon.*

Unfortunately if the ViSP library has been working quite well with the wooden mockup made by LAAS of the engine pylon, (as shown on the third section) it did failed with the 3D-print provided by Airbus. The main reason is the lack of sharp edge in the back of the pylon. We tried various strategies such as including a Kalman Filter, introducing knowledge in the tracker, but it turns out to be easier to use the Motion Capture System. If this control law shows great promises, at this time it needed further improvement in the robustness part to be used in a repeatable setup. Our line of research was to improve the balance control by including the dynamic filter. The dynamic filter has already been implemented and tested [Naveau 2014] and in Chap. 1 and the resolved momentum control is being implemented.

### 6.3 Whole body motion for screwing

The goal of this third behavior was to check if the humanoid robot HRP-2 is able to make the basic motion necessary to perform screwing action on the engine pylon. Note that an extender of 10 cm is suppose to be at the extremity of the electric screwdriver.

The behavior realized was based on the stack of tasks [Mansard 2009] a framework which is combining different control laws together and takes advantage of humanoid robots redundancy. Its software implementation has been used since 2006 to implement various demonstrators. The goal of the mathematical formulation is to enforce properties which make the control safer by checking strictly some limits. The efficiency is preserved and online changes of the control are till possible [Escande 2014].

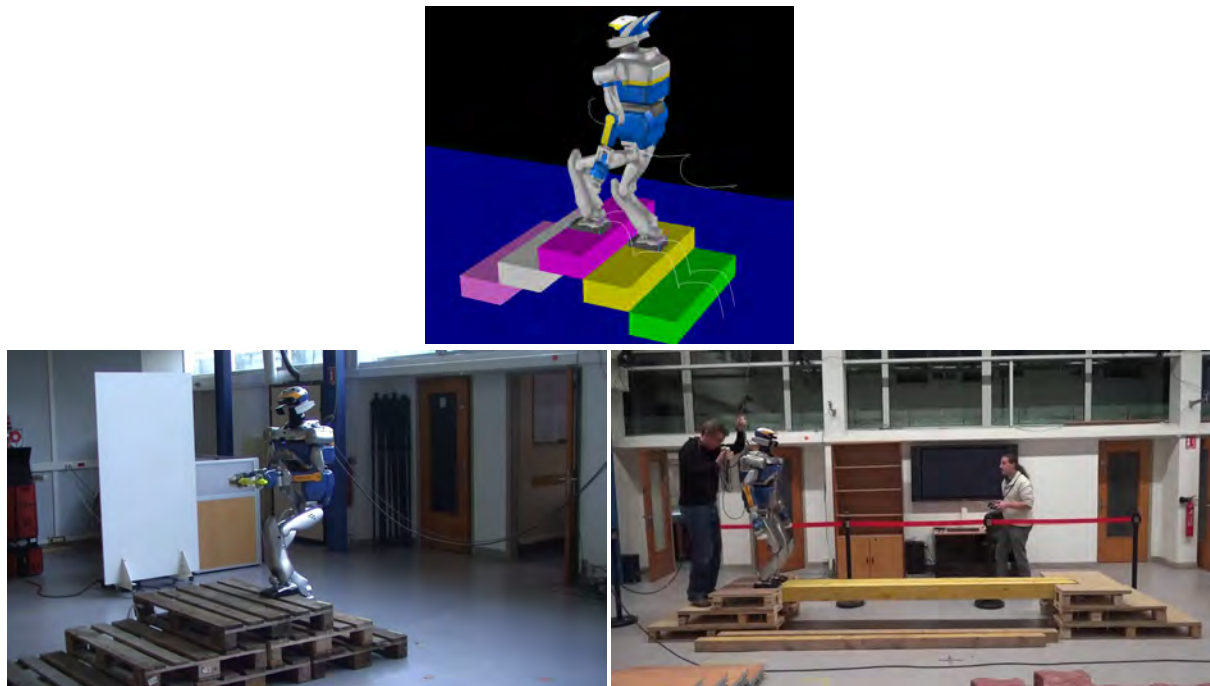
In the frame of the PoC, the main point was to test the work space of the HRP-2 humanoid with a 3D print of an AIRBUS screwdriver. The robot was able to reach most of the positions in the frontal plane of the engine pylon. On the wood mockup we have been able to realize a behavior where the robot is visually tracking the point by a whole body motion without moving the feet. However the vision process did not work properly on the 3D-print, so we decided to switch to the Motion Capture. This is demonstrated in the Fig. 6.3. It has to be noticed that the transition between the points is not formally proved or checked. It worked because the robot is highly redundant and we did not push the robot to the limit. In order to reach the screws,

we would need the plan the whole body trajectories offline. During the process we have tested acceleration joint control to improve the behavior of the system.

The behavior was very much improved but we found one problem when using posture tasks. For this reason we came back to the usual kinematic control. In addition we have tested very high gain showing that the robot is able to go up to 1.6s between the transitions. The momentum involved by this fast motion cannot be recovered by the current robot stabilizer and still has to be taken into account at a higher level. For this reason, we kept a rather low gain for the robot and we are currently in the process of improving it. Finally we noted that when the robot is lowering down it is close to self-collision. This can be fixed by using self-collision avoidance, however it calls for a deep interaction between planning and control. This is especially true when using vision. A slight drift in rotation may prevent the convergence of the controller to a screwing point.

## 6.4 3D walking

In the frame of the KoroBot project we had to make the robot going through stair cases and stepping stones. Moreover partners from Airbus asked if it was feasible to walk and climb stairs with an industrial tool in the robot hand. Hence the goal of this fourth behavior was to design a walking pattern generator able to make HRP-2 climbing and going down stairs with a tool in the gripper.



*Figure 6.4: The first two images depicts HRP-2 climbing stair in the 3D environment of OpenHRP and on the real stairs. The last one show HRP-2 ready to walk on a beam.*

At the beginning of my thesis there were no controller in LAAS-CNRS able to make HRP-2 going through stair cases. Two developments had to be made. First the development of need

3D trajectories for the feet. And second the development of new 3D CoM trajectories. We decided to use the existing walking pattern generator from [Stasse 2009] and extend it to 3D walking. The results are presented in [Naveau 2014]. The video of the motion can be found here: <https://www.youtube.com/watch?v=kBeLa5Rsy4w>.

We started with the design of feet trajectories. Order 5 splines are usually used but they are quite unstable. So we decided to implement 5 order B-splines. A B-spline of order  $n$  is a piecewise polynomial function of degree  $n$  in a variable  $x$ . It is defined by a knot vector  $T = [t_0, \dots, t_{Nt}]$  and control points  $P = [P_0, \dots, P_{Np}]$ . With  $Nt$  and  $Np$  being respectively the number of knots and the number of control points. The B-spline is then defined recursively from the knot vector and the control points. The trajectory generated is inside the convex hull of the control points. We computed B-spline of order 5 with zero velocity and acceleration during take off and landing to avoid impacts. The trajectories depends on the stair height. We enable HRP-2 to walk on 10 cm and 15 cm height steps. For the CoM height trajectory we designed a finite state machine coupled with heuristics to take into account the kinematics of the robot.

From the first picture of Fig. 6.4 we can see the simulation results obtained. The white lines depict the motion of the CoM and the feet. First experiments were performed on wooden pallet. The wood add compliance to the system and help absorbing the uncertainty of the model. In [Naveau 2014] we show that using the dynamic filter from An. A we are able to take into account the upper body posture in the COM trajectory generation. We were able to make HRP-2 carry a 3D-printed industrial screwdriver.

For the robot KPI we designed leg cross over trajectories to enable the robot to walk on the beam. See Fig. 6.4. This specific motion is trigger if self collision Self collision are detected using the line between the lift off position and the landing position of singing foot. We compute the distance between the support foot and that line to verify if the swing foot will collide with the robot support leg.



Figure 6.5: A more realistic setup.

Further tests were performed using more realistic stairs (see Fig. 6.5). On these stiffer stairs there were a higher rate of failure. Indeed the impacts, when the foot landed, were more important due the difference of stiffness. These impacts are due to model uncertainties and were absorbed by the wood flexibility. To cope with this problem we decided to implement multicontact pattern generator (see Chap. 2, and as further work to implement admittance control on the feet.

## 6.5 Conclusion

In this PoC we proved that the robot HRP-2 is able to evolve in an environment with moving objects such as a toolbox pushed by a human. We have shown that using a localization system whole body motion and goal driven walking can be performed. We proved that the robot was able to go through stairs cases with industrial tools in its grippers.

The current limitation of the system is the complexity of the environment and the large space of possible motion for the robot. With the current setup on which we agree for the PoC the current techniques seem applicable. It is however possible to improve the speed of the robot and try the next level where the robot is really trying to perform the screwing action. We are looking forward to continue this line of action by trying more challenging environment and motion.

# Conclusion and Perspectives



In the frame of the KoroiBot project, researchers integrated and developed new controllers to improve humanoid robot locomotion. Collaborations with the KoroiBot partner universities and the Airbus industry were fruitful. In this context, the contributions of this thesis can be classified in two parts. On one hand, there are theoretical contributions concerning the locomotion of humanoid robot. On the other hand, technical contributions were done in industrial and academic context. In the next section we will summarize the work that has been explained in this thesis. And in a second part we will present the further improvements that can be made to increase the performance of the existing algorithms.

## Contributions

In this section we will recall the different approaches we used to improve the locomotion of humanoid robot throughout this thesis. The improvements will be made in comparisons with the existing state of the art and in function of the Key Performance Indicators (KPI) chosen in the KoroiBot project.

### Theoretical contributions

In collaboration with our mathematician colleagues from the University of Heidelberg we designed two novel controllers improving the locomotion of humanoid robots. First we will summarize the properties of the walking pattern generator from Chap. 1. And more specifically the improvement made in comparison with similar algorithm ran on the same HRP-2 robot. Secondly we will remind the contribution made in Chap. 2 and the improvements in terms of KPI.

### Flat ground locomotion

Chap. 1 presents a new real time walking pattern generator. Indeed the KPI evaluated at the beginning pointed out that the velocity and versatility of the existing walking pattern generators [Herdt 2010a] should be improved. To answer the first problem we evaluated the maximum velocity that the robot could reach without falling in the LAAS-CNRS experimental room using the state-of-the-art controller [Herdt 2010a]:  $V_{max} = 0.125 \text{ m/s}$ . This limitation was overcome using the dynamic filter used by Nishiwaki in [Nishiwaki 2009]. This filter takes into account the whole body trajectory on a control preview horizon. It modifies the trajectory of the CoM so that the CoP computed via the whole body model fit at best the CoP computed via the linear inverted pendulum model. The maximum achievable velocity using this filter was:  $V_{max} = 0.4 \text{ m/s}$ . The question of versatility was addressed by formulating [Herdt 2010a] as a nonlinear model predictive control. This allows two interesting developments, first the simultaneous optimization of the foot step orientation and position. And second the implementation of nonlinear constraints like obstacle avoidance constraints.

### Multi-contact locomotion

The second problem that arose from the KPI was the energy consumption during the stairs climbing. In fact the total mechanical energy consumed without using the handrail where:  $\tilde{E}_{meca} = \left( \int_{t_{begin}}^{t_{end}} \tau \omega dt \right) / D = 255270 \text{ J/m}$ . With  $\tilde{E}_{meca}$ ,  $\tau$ ,  $\omega$ , and  $D$  being respectively the mechanical energy normalized over the walked distance, the joint torques, the joint velocity,

and the walked distance. For comparison the mechanical energy for normal walking is around  $47070 J/m$  on HRP-2. We assumed that distributing the weight over more than two limbs would decrease the energy consumption. This hypothesis were confirmed from the results. The energy consumed by HRP-2 while climbing stairs with the handrail support was  $191450 J/m$ . We have 25% less energy consumed. So the contribution of this chapter is the development of a new optimal control problem for the generation of multicontact locomotion using the centroidal momentum. The solver use a multiple shooting approach and linearize the constraints at each iteration. Therefore the only hypothesis made is that the global angular momentum is approximated using the global orientation of the robot. The implementation of this first approach was a bit naive and the computation time was about  $30 min$  but the real robot HRP-2 was able to perform the motion. Another collaboration with a colleague from LAAS-CNRS resulted in a reformulation of the problem using the same solver. This new formulation is solvable in  $100 ms$  which is close to be real time. As future work researchers in LAAS-CNRS will integrate this work in the HRP-2 embedded computer to perform online multicontact locomotion generation.

### Technical contributions

In the previous chapters we were focusing on controlling the humanoid robot and its balance for locomotion. In the context of KoroBot we explored three scientific problems in a novel way using the high level walking controller presented in Chap. 1. We conclude by potential industrial applications.

### Reactive control for pulling a stiff fire hose with HRP-2

In the context of perturbation rejection and potential application for humanoid robots in dangerous environments we made HRP-2 pulling a stiff fire hose without water. This work has been done in collaboration with the Japanese national institute of Advanced Industrial Science and Technology (AIST) and is described in Chap. 3. This application is inspired from the DARPA robotic challenge. The idea of this work is to see if a humanoid robot with average power and size like HRP-2 is able, using a state of the art controller, to pick up and pull a stiff fire hose. We used whole body kinematic planning to make HRP-2 pick up the fire hose. During the walk we identify a major drift in orientation due to forces applied by the hose on the robot. To correct this drift we used a PI controller where the robot position is measured via a motion capture system. We also identify impacts when the robot feet touch the ground while walking so we implemented an impedance controller to make the hand loose in single support and pull the hose in double support. Results show that a average size and power humanoid robot like HRP-2 using state of the art controller is able to pick-up and pull a fire hose toward a desired position and orientation.

### Does the two third power law help humanoid robots to walk?

In Chap. 4 we saw that in the frame of the KoroBot project, researchers studied the human motion to extract quantitative data. Those experts noticed that humans adapt their velocities in function of the curvature of their trajectories. The question is, does this law extracted from human motion help the robot to walk? A fruitful collaboration with these experts allowed us to



answer this question. This resulted in the design of an innovating controller for tracking cyclic trajectories of the center of mass. This controller is based on Andronov-Hopf oscillator. Limit cycle contracting systems can be implemented that create a velocity vector field exponentially converging toward a cyclic trajectory. The amplitude of the vector field can be mapped to fit the two third power law or any function that keeps the contracting property of the vector field. The law used is:  $V = \gamma k^\beta$ . With  $V$  the velocity on the ground plane ( $x, y, \text{yaw}$ ),  $\gamma$  a scaling factor,  $k$  the curvature, and  $\beta$  an integer. The controller we designed use the velocity extracted from the vector field and send it to the walking pattern generator of Chap. 1. The idea was to compare the quality of the trajectory tracking using several value of  $\beta$ , knowing that  $\beta = 1/3$  is the value extracted from human motion. By comparing different power laws for  $\beta \in \{-1/3, 0, 1/3, 2/3\}$ , we show that the quality of the trajectory tracking is better using the  $\beta = 1/3$ , the value extracted from human motion. The accumulated distance error to the reference trajectory in  $L_2$  norm and the accumulated orientation error in degrees are minimal using the power law with  $\beta = 1/3$ . To conclude, the two third power law help humanoid robots to walk in a more efficient manner.

### **Fusion of human motion primitives and model based control**

Chap. 5 depicts another collaboration with human motion experts. In this chapter we propose a whole body controller using upper body movement primitives extracted from human behavior and lower body movement computed by a walking pattern generator. The motivation of this hybrid approach is that generating leg trajectories for walking maintaining balance is different for humanoid robots and humans. Indeed, kinematic and dynamic constraints are different from humanoid robots to humans. The difference in number of degrees of freedom and the weight of each limbs illustrate this. Therefore we used the algorithm depicted in Chap. 1 that produce secure walking motion generation for the legs and human motion primitives for the upper body trajectory. The example of motion used in this context was a reaching and grasping motion. Human motion primitives were learned using an architecture composed of Andronov-Hopf oscillators which could be seen as central pattern generators plugged in parallel. After the learning, the upper body trajectories are simply linear combinations of these oscillators. This allow real time motion generation of those trajectories. The proposed application is therefore a real time controller allowing whole body motion.

### **HRP-2 as the universal worker**

A collaboration with the Airbus Industry leaded us toward designing real time reactive controllers for possible applications of humanoid robots in an industry. Hence, we designed applications that make the robot facing real life perturbations. The first example is a robot walking in a hangar. In this context the robot must be able to move without colliding with potential obstacles. In collaboration with former colleagues in LAAS-CNRS we used online planner for obstacle avoidance to direct the robot through the 5 m length experimental room. The second task was to place the robot in a position relative to a specific part. Visual servoing coupled with a walking pattern generator was used to answer this problematic. Finally the robot was asked to drill or screw a part. A study of kinematic feasibility was performed using the Stack of Tasks, which is the whole body controller containing a generalized inverse kinematics used and implemented in LAAS-CNRS.

## Perspectives

The problem of locomotion for humanoid robot is not fixed yet. Several good controllers already propose quite good performance (see the DARPA robotics challenge) [Ajoudani 2014, Kuindersma 2015, Johnson 2015]. However the bottle neck of these approaches are the execution time and the robustness. As future work, we should focus on implementing real time, reactive, and robust controllers for humanoid robots in various and increasingly complex environments. In this thesis we implemented several applications that reflect this idea. And we faced unsolved scientific problems. One of them concern the online generation of CoM and end effector trajectories in a cluttered environment. For example when HRP-2 is going up or down the stairs, the kinematic constraints are not well formulated. They do not reflect the properties of the environment. Hence to be able to perform such motion, we had to find heuristics to provide 3D continuous trajectories that fit all classical constraints. That is to say, we have to take into account the kinematic constraints including singularities, self collisions, collisions with the environment, as well as the dynamic constraints like maximum joint torque constraints etc. One potential solution is to extend the work of [Herdt 2010a, Brasseur 2015] to generate 3D convex bounds that capture all the above constraints. Those bounds would be computed from stochastic exploration of the configuration space taking into account the robot dynamics and kinematics. Another one would be to use fast re-planning strategy to obtain feasible way points for the feet and CoM and interpolate between them using B-splines for example.

In the same idea concerning walking on stepping stones is a problem. In fact the problem can be described as a set of convex walkable regions. And this problem formulation is not convex. Numbering the walkable region and using mixed integer algorithm to solve the problem is a possibility [Deits 2014]. Considering the possibility of designing a real time walking pattern generator from this idea is appealing. We could derive a nonlinear formulation of it. A possible application would be to drive the robot with a user input velocity and the robot would be able to avoid obstacles and climb stairs automatically.

The goal, to be achieved in control nowadays, is to integrate all the previous algorithms depicted or cited in this thesis. The problem of the locomotion of humanoid robots would be solved if we were able to find a general locomotion controller. It would use the whole body dynamic and automatically find the end-effector positions and orientations given a model of the environment.

This thesis shows that humanoid robot flat ground walking is a topic nearly solved. Application using high level controller can already be demonstrated. The extension of this work, the multicontact locomotion, is a very important topic nowadays. It captures the idea of a generic algorithm that can be used for flat ground locomotion or parkour motion execution. There are plenty of room for improvement in this thesis work and interesting perspectives concerning the robot locomotion performance and versatility. Other European project like the KoroïBot project should arise to see these improvements becoming a reality.

APPENDIX A

# Annexe

---



## A.1 The dynamic filter

In this annexe we will present in details the dynamic filter algorithm. This work has been published in [Naveau 2014]. First of all we will briefly present the algorithm in the context of humanoid walking. Then we will explicit technical details on the implementation. Finally we will show that the walk of humanoid robot on flat ground is improved.

### A.1.1 Dynamic filtering for walking

#### filtering on a sub-sampled walking pattern generator

This work is based upon the real time walking control system described by Nishiwaki in [Nishiwaki 2009]. One key ingredient is to generate real-time dynamically stable CoM and Center-of-Pressure (CoP) trajectories using the cart-table model [Kajita 2003a]. Then to correct the inertial effects induced by the legs movement Nishiwaki uses a sub-sampled dynamical filter. Indeed, the results tested on HRP-2 have shown that the CoP trajectory followed by the robot are not the desired one, the difference can be as big as 15 mm. To avoid this problem, two kind of solution exist :

1. regulating the linear and angular momentum [Kajita 2010]
2. or/and correcting the CoM trajectory by taking into account the dynamics of the robot using a dynamical filter.

The two solution are not incompatible. In practice, the first solution is mostly used to reject instantaneous perturbations. The problem is that it uses additional degrees of freedom that could have been used for manipulation tasks for example. For this reason, in our specific case, we have only implemented the dynamical filter as the commercially available stabilizer on the HRP-2 at LAAS-CNRS integrates already a perturbation rejection strategy.

### Mathematical derivation

In the following list the mathematical process is described.

1. The walking pattern generator (WPG) defines a trajectory for the CoM, the feet and the CoP. The desired CoM and CoP trajectories are respectively noted  $\mathbf{CoM}^*$  and  $\mathbf{CoP}^*$ .
2. The dynamic filter starts by computing the joint trajectories using an analytical inverse kinematics (*AIK*). This inverse kinematics is very fast to compute and its main assumption is that the CoM and the free flyer are rigidly connected.

$$(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = AIK(ponctual\_model, \mathbf{c}, \dot{\mathbf{c}}, \ddot{\mathbf{c}}, \mathbf{X}^f)$$

with  $\mathbf{c}, \dot{\mathbf{c}}, \ddot{\mathbf{c}}$  et  $\mathbf{X}^f$  being respectively the position, the velocity, the acceleration of the CoM and the feet position.

3. With the joint trajectory, it is possible to compute the inverse dynamics (*ID*) and to find the CoP matching up to the real robot motion. We call it the multi-body CoP noted  $\mathbf{CoP}^{MB}$ .

$$\begin{aligned}
(\mathbf{f}, \boldsymbol{\tau}) &= ID(model\_complet, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \\
CoP_x^{MB} &= -\frac{\tau_y}{mg} \\
CoP_y^{MB} &= \frac{\tau_x}{mg} \\
CoP_z^{MB} &= 0 \\
\Delta CoP &= CoP^* - CoP^{MB}
\end{aligned}$$

4. This provides an error between  $CoP^*$  and  $CoP^{MB}$ . This error is computed over a time window and is injected in a preview control (PC) in the shape of a LQR described by Kajita and al. [Kajita 2003a]. The result of this step is an error for the CoM in position, velocity and acceleration.

$$\Delta CoM = PC(\Delta CoP)$$

5. We can then sum this error on the reference CoM ( $CoM^*$ ) to correct the trajectory.

$$\overline{CoM}^* = CoM^* + \Delta CoM$$

With all  $CoM$  relative vector being composed by  $\begin{bmatrix} c_x & \dot{c}_x & \ddot{c}_x & c_y & \dot{c}_y & \ddot{c}_y \end{bmatrix}^T$

6. Finally, the new CoM trajectory is used to compute the joint trajectory using a Generalize Inverse Kinematics.

The scheme depicted in Fig. A.1 represents the above steps. This dynamic filter is based on the Newton-Raphson algorithm using the CoM trajectory as free variables. This method does not guarantee the convergence. However during tests on HRP-2 with other WPG, we observed that the CoP trajectory is corrected in one iteration. The idea is to implement this method on different WPG like the WPG of Andrei Herdt [Herdt 2010b], Morisawa's [Morisawa 2007], and on the one describe in Chap. 1. In the next section we will present technical details that are important for the implementation of such method.

### A.1.2 Technical details

Fig.A.1 present a detailed scheme of the dynamical filter using different acronyms. For a start lets define them all :

1. FIFO stands for First In First Out buffer. In the Fig.A.1, the duration of the FIFO is  $T$  and the sampling period is  $dt$ . We can define the number of sampling by  $n = T/dt$
2. W.P.G. stands for Walking Pattern Generator,
3. A.I.K. stands for Analytical Inverse Kinematics,
4. F.D. stands for Finite Differentiation,
5. I.D. stands for Inverse Dynamics,
6. L.I. stands for Linear Interpolation,
7. ctrl stands for Control,

This algorithm depends on three key algorithms. The first one is the analytical inverse kinematics. The purpose of this one is to provide an approximation of the whole body movement induced by the first computation of  $CoM^*$  and  $Feet^*$ . As we use it, the A.I.K provides us the lower body configuration. This simple fact imposes that the velocity and acceleration of the robot

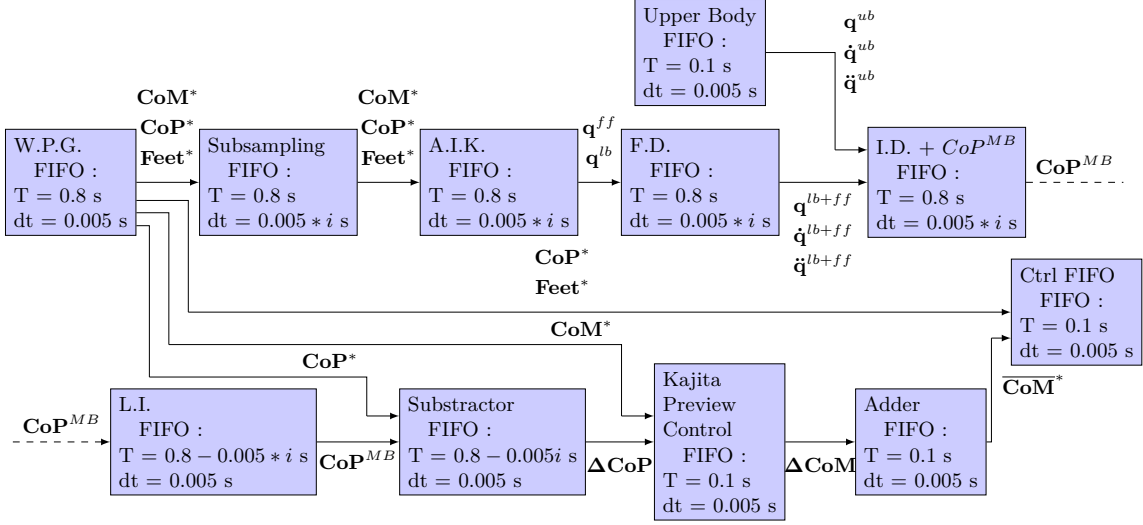


Figure A.1: This scheme describe the action of the dynamic filter. It contains all needed information in order to filter a desired CoM trajectory ( $\text{CoM}^*$ ) knowing the associated feet and CoP trajectories ( $\text{Feet}^*$ ,  $\text{CoP}^*$ ). See Subsec. A.1.2 for all the details.

has to be computed from the position. Hence we use finite differentiations subject to noise and discontinuities to compute the velocity and the acceleration. Therefore, one constraint of this algorithm is that the original  $\text{CoM}^*$  and  $\text{Feet}^*$  provided by the W.P.G. must be at least  $\mathcal{C}^2$ . The next algorithm is the inverse dynamic also know as RNEA. This algorithm is computed several times per control iteration. Hence to be able to use the dynamic filter online we need to compute it as fast as possible. Several solutions exist but one specifically provides extremely good results, i.e. the template programming see [Naveau 2014]. One most recent version of this algorithm is even more efficient than the one in [Naveau 2014]. It is open source and can be found here [pin 2016]. The last algorithm that needs introduction is the seminal Kajita preview control which computes CoM trajectories from input CoP reference.

In Chap. 1 the impact of this algorithm on the walk of the humanoid robot HRP-2 is demonstrated.

### A.1.3 Conclusion

In this annexe we presented the dynamic filter, key algorithm for stable ground walking. Firstly, we derived the theoretical aspect of the algorithm and secondly, we shown several technical aspect regarding the implementation. This annexe has for main purpose to express the tricks and details of the implementation of the dynamical filter. This should provide a good guide for someone willing to implement this algorithm.





# Bibliography

- [Ajallooeian 2013] M. Ajallooeian, J. van den Kieboom, A. Mukovskiy, M. Giese and A. Ijspeert. *A general family of morphed nonlinear phase oscillators with arbitrary limit cycle shape*. Physica D: Nonlinear Phenomena, 2013. (Cited in pages 76 and 89.)
- [Ajoudani 2014] A. Ajoudani, J. Lee, A. Rocchi, M. Ferrati, E. M. Hoffman, A. Settini, D. G. Caldwell, A. Bicchi and N. G. Tsagarakis. *A manipulation framework for compliant humanoid COMAN: Application to a valve turning task*. In Int. Conf. on Humanoid Robotics, 2014. (Cited in pages 89 and 114.)
- [Audren 2014] H. Audren, J. Vaillant, A. Kheddar, A. Escande, K. Kaneko and E. Yoshida. *Model preview control in multi-contact motion-application to a humanoid robot*. In Int. Conf. on Intelligent Robots and Systems, 2014. (Cited in pages 43 and 50.)
- [Bennequin 2009] D. Bennequin, R. Fuchs, A. Berthoz and T. Flash. *Movement timing and invariance arise from several geometries*. PLoS computational biology, 2009. (Cited in pages 73, 77 and 83.)
- [Bock 1981] H. G. Bock. Numerical treatment of inverse problems in chemical reaction kinetics. Springer, 1981. (Cited in page 50.)
- [Bock 2007] H. G. Bock, M. Diehl, P. Kühn, E. Kostina, J. Schlöder and L. Wirsching. *Numerical Methods for Efficient and Fast Nonlinear Model Predictive Control*. In Assessment and Future Directions of Nonlinear Model Predictive Control. 2007. (Cited in page 34.)
- [Boyd 2004] S. Boyd and L. Vandenberghe. Convex optimization. Cambridge University Press, 2004. (Cited in page 25.)
- [Boyd 2007] S.-P. Boyd and B. Wegbreit. *Fast Computation of Optimal Contact Forces*. Transactions on Robotics, 2007. (Cited in page 47.)
- [Brandao 2013] M. Brandao, L. Jamone, P. Kryczka, N. Endo, K. Hashimoto and A. Takanishi. *Reaching for the unreachable: integration of locomotion and whole-body movements for extended visually guided reaching*. In Int. Conf. on Humanoid Robotics, 2013. (Cited in page 89.)
- [Brasseur 2015] C. Brasseur, A. Sherikov, C. Collette, D. Dimitrov and P. B. Wieber. *A robust linear MPC approach to online generation of 3D biped walking motion*. In Int. Conf. on Humanoid Robotics, 2015. (Cited in page 114.)
- [Buchli 2006] J. Buchli, L. Righetti and A. J. Ijspeert. *Engineering Entrainment and Adaptation in Limit Cycle Systems - from biological inspiration to applications in robotics*. Biological Cybernetics, 2006. (Cited in pages 88 and 91.)
- [Carpentier 2015] J. Carpentier, S. Tonneau, M. Naveau, O. Stasse and N. Mansard. *A Versatile and Efficient Pattern Generator for Generalized Legged Locomotion*. In Int. Conf. on Robotics and Automation, 2015. (Cited in pages 54, 55 and 74.)

- [Chang 2001] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. (Cited in page 91.)
- [Chestnutt 2010] J. Chestnutt. Motion planning for humanoid robots, chapitre Navigation and Gait Planning. 2010. (Cited in pages 23, 74 and 103.)
- [Chiovetto 2013] E. Chiovetto, A. d’Avella, D. Endres and M. A. Giese. *A unifying algorithm for the identification of kinematic and electromyographic motor primitives*. In Bernstein Conference, 2013. not reviewed. (Cited in page 90.)
- [Dai 2014a] H. Dai, A. Valenzuela and R. Tedrake. *Whole-body motion planning with simple dynamics and full kinematics*. In Int. Conf. on Humanoid Robotics, 2014. (Cited in page 43.)
- [Dai 2014b] H. Dai, A. Valenzuela and R. Tedrake. *Whole-body motion planning with centroidal dynamics and full kinematics*. In Int. Conf. on Humanoid Robotics, 2014. (Cited in page 25.)
- [Dalibard 2013] S. Dalibard, A. El Khoury, F. Lamiriaux, A. Nakhaei, M. Taïx and J. Laumond. *Dynamic Walking and Whole-Body Motion Planning for Humanoid Robots: an Integrated Approach*. the International Journal of Robotics Research, 2013. (Cited in page 89.)
- [Dayan 2007] E. Dayan, A. Casile, N. Levit-Binnun, M. Giese, T. Hendler and T. Flash. *Neural representations of kinematic laws of motion: evidence for action-perception coupling*. Proceedings of the National Academy of Sciences, 2007. (Cited in page 73.)
- [Deits 2014] R. Deits and R. Tedrake. *Footstep Planning on Uneven Terrain with Mixed-Integer Convex Optimization*. In Int. Conf. on Humanoid Robotics, 2014. (Cited in pages 16, 23, 25, 38, 74 and 114.)
- [Diehl 2002] M. Diehl. Real-Time Optimization for Large Scale Nonlinear Processes, volume 920 of *Fortschritt-Berichte VDI Reihe 8, Meß-, Steuerungs- und Regelungstechnik*. 2002. (Cited in page 34.)
- [Dune 2011] C. Dune, A. Herdt, E. Marchand, O. Stasse, P.-B. Wieber and E. Yoshida. *Vision based control for Humanoid Robots*. In Int. Conf. on Intelligent Robots and Systems, 2011. (Cited in page 104.)
- [Englsberger 2013] J. Engelsberger, C. Ott and A. Albu-Schäffer. *Three-dimensional bipedal walking control using Divergent Component of Motion*. In Int. Conf. on Intelligent Robots and Systems, 2013. (Cited in page 44.)
- [Englsberger 2014] J. Engelsberger, T. Koolen, S. Bertrand, J. Pratt, C. Ott and A. Albu-Schäffer. *Trajectory generation for continuous leg forces during double support and heel-to-toe shift based on divergent component of motion*. In Int. Conf. on Intelligent Robots and Systems, 2014. (Cited in page 89.)

- [Englsberger 2015] J. Engelsberger, P. Kozłowski and C. Ott. *Biologically inspired deadbeat control for running on 3D stepping stones*. In Int. Conf. on Humanoid Robotics, 2015. (Cited in page 25.)
- [Escande 2013] A. Escande, A. Kheddar and S. Miossec. *Planning contact points for humanoid robots*. Journal of Robotics and Autonomous Systems, 2013. (Cited in page 74.)
- [Escande 2014] A. Escande, N. Mansard and P.-B. Wieber. *Hierarchical quadratic programming: Fast online humanoid-robot motion generation*. the International Journal of Robotics Research, 2014. (Cited in page 105.)
- [Evrard 2009] P. Evrard, N. Mansard, O. Stasse, A. Kheddar, T. Schauß, C. Weber, A. Peer and M. Buss. *Intercontinental, multimodal, wide-range tele-cooperation using a humanoid robot*. In Int. Conf. on Intelligent Robots and Systems, 2009. (Cited in page 15.)
- [Feng 2012] A. W. Feng, Y. Xu and A. Shapiro. *An example-based motion synthesis technique for locomotion and object manipulation*. In the International Symposium on Computer Animation, 2012. (Cited in page 88.)
- [Feng 2013] S. Feng, X. Xinjilefu, W. Huang and C. G. Atkeson. *3D walking based on online optimization*. In Int. Conf. on Humanoid Robotics, 2013. (Cited in page 24.)
- [Ferreau 2014] H. Ferreau, C. Kirches, A. Potschka, H. Bock and M. Diehl. *qpOASES: A parametric active-set algorithm for quadratic programming*. Mathematical Programming Computation, 2014. (Cited in page 38.)
- [Flash 1985] T. Flash and N. Hogan. *The coordination of arm movements: an experimentally confirmed mathematical model*. The journal of Neuroscience, 1985. (Cited in page 83.)
- [Flash 1996] T. Flash and A. A. Handzel. *Affine differential geometry of human arm trajectories*. Abstracts of the Society for Neuroscience, 1996. (Cited in page 73.)
- [Flash 2007] T. Flash and A. Handzel. *Affine differential geometry analysis of human arm movements*. Biological Cybernetics, 2007. (Cited in pages 73 and 83.)
- [Gams 2008] A. Gams, L. Righetti, A. J. Ijspeert and J. Lenarčič. *A dynamical system for online learning of periodic movements of unknown waveform and frequency*. In Int. Conf. on Biomedical Robotics and Biomechatronics, 2008. (Cited in page 88.)
- [Gams 2013] A. Gams, B. Nemec, L. Zlajpah, M. Wächter, A. J. Ijspeert, T. Asfour and A. Ude. *Modulation of Motor Primitives using Force Feedback: Interaction with the Environment and Bimanual Tasks*. In Int. Conf. on Intelligent Robots and Systems, 2013. (Cited in page 89.)
- [Garcia 2014] M. Garcia, O. Stasse and J.-B. Hayet. *Vision-driven walking pattern generation for humanoid reactive walking*. In Int. Conf. on Robotics and Automation, 2014. (Cited in page 56.)
- [Garcia 2015] M. Garcia, O. Stasse, J.-B. Hayet, C. Dune, C. Esteves and J.-P. Laumond. *Vision-guided motion primitives for humanoid reactive walking: decoupled vs. coupled approaches*. the International Journal of Robotics Research, 2015. (Cited in page 36.)

- [Gienger 2010] M. Gienger, M. Toussaint and C. Goerick. *Whole-body motion planning—building blocks for intelligent systems*. In Motion Planning for Humanoid Robots. 2010. (Cited in page 89.)
- [Giese 2009a] M. A. Giese, A. Mukovskiy, A. Park, L. Omlor and J. J. E. Slotine. *Real-Time Synthesis of Body Movements Based on Learned Primitives*. In Stat. and Geom. Appr. to Vis. Mot. Anal., LNCS5604. Springer, 2009. (Cited in pages 89, 91 and 92.)
- [Giese 2009b] M. Giese, A. Mukovskiy, A. Park, L. Omlor and J. Slotine. *Real-Time Synthesis of Body Movements Based on Learned Primitives*. In Cremers D, Rosenhahn B, Yuille A L (eds): Statistical and Geometrical Approaches to Visual Motion Analysis, Lecture Notes in Computer Science, 2009. (Cited in page 75.)
- [Grizzle 2010] J. W. Grizzle, C. Chevallereau, A. D. Ames and R. W. Sinnet. *3D bipedal robotic walking: models, feedback control, and open problems*. IFAC Proceedings Volumes, 2010. (Cited in page 13.)
- [Harada 2003] K. Harada, S. Kajita, K. Kaneko and H. Hirukawa. *Pushing manipulation by humanoid considering two-kinds of ZMPs*. In Int. Conf. on Robotics and Automation, 2003. (Cited in page 61.)
- [Harada 2004] K. Harada, S. Kajita, K. Kaneko and H. Hirukawa. *Real-time planning of humanoid robot's gait for force controlled manipulation*. In Int. Conf. on Robotics and Automation, 2004. (Cited in pages 61 and 65.)
- [Harada 2005] K. Harada, S. Kajita, H. Saito, M. Morisawa, F. Kanehiro, K. Fujiwara, K. Kaneko and H. Hirukawa. *A humanoid robot carrying a heavy object*. In Int. Conf. on Robotics and Automation, 2005. (Cited in page 61.)
- [Herd 2010a] A. Herdt, N. Perrin and P.-B. Wieber. *Walking without thinking about it*. In Int. Conf. on Intelligent Robots and Systems, 2010. (Cited in pages 14, 16, 17, 23, 24, 25, 26, 28, 30, 31, 32, 35, 74, 87, 111 and 114.)
- [Herd 2010b] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur and M. Diehl. *Online walking motion generation with automatic footstep placement*. International Journal of Advanced Robotic Systems, 2010. (Cited in pages 15, 87, 104 and 118.)
- [Herzog 2015] A. Herzog, N. Rotella, S. Schaal and L. Righetti. *Trajectory generation for multi-contact momentum control*. In Int. Conf. on Humanoid Robotics, 2015. (Cited in page 44.)
- [Hirukawa 2007] 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 Int. Conf. on Robotics and Automation, 2007. (Cited in pages 14 and 44.)
- [Holoborodko 2008] P. Holoborodko. *Smooth Noise Robust Differentiators*, 2008. (Cited in page 80.)

- [Hornung 2012] A. Hornung, A. Dornbush, M. Likhachev and M. Bennewitz. *Anytime search-based footstep planning with suboptimality bounds*. In Int. Conf. on Humanoid Robotics, 2012. (Cited in pages 23, 74 and 103.)
- [hpp 2016] *Humanoid Path Planner (HPP)*, 2016. (Cited in page 63.)
- [Huang 2014] Y. Huang and M. Kallmann. *Planning Motions for Virtual Demonstrators*. In Intelligent Virtual Agents. Springer, 2014. (Cited in page 88.)
- [Huh 2015] D. Huh and T. Sejnowski. *Spectrum of power laws for curved hand movements*. Proceedings of the National Academy of Sciences, 2015. (Cited in page 73.)
- [Hwang 2003] Y. Hwang, A. Konno and M. Uchiyama. *Whole body cooperative tasks and static stability evaluations for a humanoid robot*. In Int. Conf. on Intelligent Robots and Systems, 2003. (Cited in page 61.)
- [Hyon 2007] S. Hyon, J. G. Hale and G. Cheng. *Full-Body Compliant Human-Humanoid Interaction: Balancing in the Presence of Unknown External Forces*. Transactions on Robotics, 2007. (Cited in page 44.)
- [Ibanez 2014] A. Ibanez, P. Bidaud and V. Padois. Advance in robot kinematics, chapitre Automatic Optimal Biped Walking as a Mixed-Integer Quadratic Program. 2014. (Cited in page 25.)
- [Ijspeert 2013] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor and S. Schaal. *Dynamical movement primitives: Learning attractor models for motor behaviors*. Neural Computation, 2013. (Cited in pages 89 and 91.)
- [Johnson 2015] M. Johnson, B. Shrewsbury, S. Bertrand, T. Wu, D. Duran, M. Floyd, P. Abeles, D. Stephen, N. Mertins, A. Lesmanet *al.* *Team IHMC's lessons learned from the DARPA robotics challenge trials*. Journal of Field Robotics, 2015. (Cited in pages 89 and 114.)
- [Kaddar 2015] B. Kaddar, Y. Aoustin and C. Chevallereau. *Arm swing effects on walking bipedal gaits composed of impact, single and double support phases*. 2015. (Cited in page 12.)
- [Kajita 2003a] 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 Int. Conf. on Robotics and Automation, 2003. (Cited in pages 14, 15, 25, 26, 27, 35, 44, 89, 117 and 118.)
- [Kajita 2003b] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa. *Resolved Momentum Control: Humanoid Motion Planning based on the Linear and Angular Momentum*. In Int. Conf. on Intelligent Robots and Systems, 2003. (Cited in page 11.)
- [Kajita 2010] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro and K. Yokoi. *Biped walking stabilization based on linear inverted pendulum tracking*. In Int. Conf. on Intelligent Robots and Systems, 2010. (Cited in page 117.)

- [Kaneko 2004] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi and T. Isozumi. *Humanoid Robot HRP-2*. In Int. Conf. on Robotics and Automation, 2004. (Cited in page 45.)
- [Kanoun 2011] O. Kanoun, J. Laumond and E. Yoshida. *Planning foot placements for a humanoid robots: a problem of inverse kinematics*. the International Journal of Robotics Research, 2011. (Cited in page 89.)
- [Karklinsky 2015] M. Karklinsky and T. Flash. *Timing of continuous motor imagery: the two-thirds power law originates in trajectory planning*. Journal of neurophysiology, 2015. (Cited in page 73.)
- [Karklinsky 2016] M. Karklinsky, M. Naveau, A. Mukovskiy, O. Stasse, T. Flash and P. Soueres. *Robust human-inspired power law trajectories for humanoid HRP-2 robot*. In Int. Conf. on Biomedical Robotics and Biomechatronics, 2016. (Cited in pages 73, 75 and 76.)
- [Koch 2012] K. H. Koch, K. Mombaur and P. Soueres. *Optimization-based walking generation for humanoid robot*. In SYROCO, 2012. (Cited in page 43.)
- [Koch 2014] K. H. Koch, K. Mombaur, O. Stasse and P. Soueres. *Optimization based exploitation of the ankle elasticity of HRP-2 for overstepping large obstacles*. In Int. Conf. on Humanoid Robotics, 2014. (Cited in pages 13 and 18.)
- [Koenemann 2015] J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz and N. Mansard. *Whole-body Model-Predictive Control Applied to the HRP-2 Humanoid*. In Int. Conf. on Intelligent Robots and Systems, 2015. (Cited in pages 14, 23, 25, 87 and 88.)
- [Koschorreck 2012] J. Koschorreck and K. Mombaur. *Modeling and Optimal Control of Human Platform Diving With Somersaults and Twists*. Optimization and Engineering, 2012. (Cited in page 88.)
- [Kudruss 2015] M. Kudruss, M. Naveau, O. Stasse, N. Mansard, C. Kirches, P. Soueres and K. Mombaur. *Optimal Control for Multi-Contact, Whole-Body Motion Generation using Center-of-Mass Dynamics for Multi-Contact Situations*. In Int. Conf. on Humanoid Robotics, 2015. (Cited in page 43.)
- [Kuffner 2002] J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba and H. Inoue. *Dynamically-stable motion planning for humanoid robots*. Autonomous Robots, 2002. (Cited in page 103.)
- [Kuindersma 2014] S. Kuindersma, F. Permenter and R. Tedrake. *An efficiently solvable quadratic program for stabilizing dynamic locomotion*. In Int. Conf. on Robotics and Automation, 2014. (Cited in page 14.)
- [Kuindersma 2015] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion and R. Tedrake. *Optimization-based locomotion planning, estimation, and control design for the Atlas humanoid robot*. Autonomous Robots, 2015. (Cited in pages 89 and 114.)

- [Lacquaniti 1983] F. Lacquaniti, C. Terzuolo and P. Viviani. *The law relating the kinematic and figural aspects of drawing movements*. Acta psychologica, 1983. (Cited in page 73.)
- [Land 2013] W. M. Land, D. A. Rosenbaum, S. Seegelke and T. Schack. *Whole-body posture planning in anticipation of a manual prehension task: Prospective and retrospective effects*. Acta Psychologica, 2013. (Cited in page 88.)
- [Leineweber 2003] D. B. Leineweber, I. Bauer, H. G. Bock and J. P. Schlöder. *An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part 1: Theoretical aspects*. Computers & Chemical Engineering, 2003. (Cited in page 50.)
- [Levine 2012] S. Levine, J. M. Wang, A. Haraux, Z. Popović and V. Koltun. *Continuous Character Control with Low-Dimensional Embeddings*. the International Symposium on Computer Animation, 2012. (Cited in page 88.)
- [Lohmiller 1998] W. Lohmiller and J. Slotine. *On contraction analysis for non-linear systems*. Automatica, 1998. (Cited in pages 75 and 77.)
- [Luh 1980] J. Y. Luh, M. W. Walker and R. P. Paul. *On-line computational scheme for mechanical manipulators*. Journal of Dynamic Systems, Measurement, and Control, 1980. (Cited in page 51.)
- [Mandery 2016] C. Mandery, O. Terlemez, M. Do, N. Vahrenkamp and T. Asfour. *Unifying Representations and Large-Scale Whole-Body Motion Databases for Studying Human Motion (to appear)*. Transactions on Robotics, 2016. (Cited in page 3.)
- [Mansard 2009] N. Mansard, O. Stasse, P. Evrard and A. Kheddar. *A Versatile Generalized Inverted Kinematics Implementation for Collaborative Working Humanoid Robots: The Stack of Tasks*. In International Conference on Advanced Robotics (ICAR), 2009. (Cited in pages 50, 63, 67, 75 and 105.)
- [Maoz 2005] U. Maoz, E. Portugaly, T. Flash and Y. Weiss. *Noise and the two-thirds power law*. In Advances in Neural Information Processing Systems, 2005. (Cited in page 80.)
- [Meirovitch 2015] Y. Meirovitch, H. Harris, E. Dayan, A. Arieli and T. Flash. *Alpha and Beta Band Event-Related Desynchronization Reflects Kinematic Regularities*. Journal of Neuroscience, 2015. (Cited in page 73.)
- [Mikami 2014] Y. Mikami, T. Moulard, E. Yoshida and G. Venture. *Identification of HRP-2 foot’s dynamics*. In Int. Conf. on Intelligent Robots and Systems, 2014. (Cited in page 56.)
- [moc 2016] *Motion Analysis*, 2016. (Cited in page 68.)
- [Mordatch 2012] I. Mordatch, E. Todorov and Z. Popović. *Discovery of complex behaviors through contact-invariant optimization*. ACM Transactions on Graphics (TOG), 2012. (Cited in pages 16 and 25.)
- [Morisawa 2007] M. Morisawa, K. Harada, S. Kajita, S. Nakaoka, K. Fujiwara, F. Kanehiro, K. Kaneko and H. Hirukawa. *Experimentation of Humanoid Walking Allowing Immediate Modification of Foot Place Based on Analytical Solution*. In Int. Conf. on Robotics and Automation, 2007. (Cited in pages 16, 17, 24, 53 and 118.)

- [Mukovskiy 2015] A. Mukovskiy, W. Land, T. Schack and M. Giese. *Modeling of predictive human movement coordination patterns for applications in computer graphics*. Journal of WSCG, 2015. (Cited in pages 88, 89, 90 and 92.)
- [Mukovskiy 2016] A. Mukovskiy, C. Vassallo, M. Naveau, O. Stasse, P. Souères and M. A. Giese. *Learning Movement Primitives for the Humanoid Robot HRP2*, 2016. (Submitted). (Cited in pages 87 and 90.)
- [Murooka 2014] M. Murooka, S. Noda, S. Nozawa, Y. Kakiuchi, K. Okada and M. Inaba. *Manipulation strategy decision and execution based on strategy proving operation for carrying large and heavy objects*. In Int. Conf. on Robotics and Automation, 2014. (Cited in page 61.)
- [Murooka 2015] M. Murooka, S. Nozawa, Y. Kakiuchi, K. Okada and M. Inaba. *Whole-body pushing manipulation with contact posture planning of large and heavy object for humanoid robot*. In Int. Conf. on Robotics and Automation, 2015. (Cited in page 61.)
- [Naveau 2014] M. Naveau, J. Carpentier, S. Barthelemy, O. Stasse and P. Soueres. *METAPOD - Template META-PrOgramming applied to Dynamics: CoP-CoM trajectories filtering*. In Int. Conf. on Humanoid Robotics, 2014. (Cited in pages 17, 44, 105, 107, 117 and 119.)
- [Naveau 2017] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur and P. Soueres. *A Reactive Walking Pattern Generator Based on Nonlinear Model Predictive Control*. Robotics and Automation Letters, 2017. (Cited in pages 23, 56, 64 and 74.)
- [Nishiwaki 2007] K. Nishiwaki and S. Kagami. *Walking control on uneven terrain with short cycle pattern generation*. In Int. Conf. on Humanoid Robotics, 2007. (Cited in page 35.)
- [Nishiwaki 2009] K. Nishiwaki and S. Kagami. *Online Walking Control System for Humanoids with Short Cycle Pattern Generation*. the International Journal of Robotics Research, 2009. (Cited in pages 8, 26, 111 and 117.)
- [Nishiwaki 2012] K. Nishiwaki, J. E. Chestnutt and S. Kagami. *Autonomous navigation of a humanoid robot over unknown rough terrain using a laser range sensor*. the International Journal of Robotics Research, 2012. (Cited in pages 45 and 46.)
- [Nozawa 2012] S. Nozawa, Y. Kakiuchi, K. Okada and M. Inaba. *Controlling the planar motion of a heavy object by pushing with a humanoid robot using dual-arm force control*. In Int. Conf. on Robotics and Automation, 2012. (Cited in page 61.)
- [Omlor 2011] L. Omlor and M. A. Giese. *Anechoic Blind Source Separation using Wigner Marginals*. J. of Machine Learning Res., 2011. (Cited in page 90.)
- [Orin 2012] D. E. Orin, A. Goswami and S.-H. Lee. *Centroidal dynamics of a humanoid robot*. Autonomous Robot, 2012. (Cited in page 25.)
- [Orin 2013] D. E. Orin, A. Goswami and S.-H. Lee. *Centroidal dynamics of a humanoid robot*. Autonomous Robots, 2013. (Cited in pages 11, 45 and 46.)



- [Orthey 2015a] A. Orthey. *Exploiting Structure in Humanoid Motion Planning*. PhD thesis, Univ. Toulouse III - Paul Sabatier, France, 2015. (Cited in page 83.)
- [Orthey 2015b] A. Orthey, V. Ivan, M. Naveau, Y. Yang, O. Stasse and S. Vijayakumar. *Homotopic particle motion planning for humanoid robotics*. 2015. (Submitted). (Not cited.)
- [Ott 2011] C. Ott, M. A. Roa and G. Herzinger. *Posture and Balance Control for Biped Robots based on Contact Force Optimization*. In Int. Conf. on Humanoid Robotics, 2011. (Cited in page 44.)
- [Papaxanthis 2012] C. Papaxanthis, C. Paizis, O. White, T. Pozzo and N. Stucchi. *The relation between geometry and time in mental actions*. PloS one, 2012. (Cited in page 73.)
- [Park 2009a] A. Park, A. Mukovskiy, J. E. Slotine and M. Giese. *Design of Dynamical Stability Properties in Character Animation*. In VRIPHYS, 2009. (Cited in page 76.)
- [Park 2009b] A.-n. Park, A. Mukovskiy, J.-J. E. Slotine and M. A. Giese. *Design of Dynamical Stability Properties in Character Animation*. In VRIPHYS, 2009. (Cited in page 92.)
- [Perk 2006] B. E. Perk and J. E. Slotine. *Motion Primitives for Robotic Flight Control*. CoRR, 2006. (Cited in page 75.)
- [Perrin 2010] N. Perrin, O. Stasse, F. Lamiriaux and E. Yoshida. *Approximation of feasibility tests for reactive walk on hrp-2*. In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pages 4243–4248. IEEE, 2010. (Cited in page 32.)
- [Perrin 2011] N. Perrin, O. Stasse, F. Lamiriaux and E. Yoshida. *Weakly collision-free paths for continuous humanoid footstep planning*. In Int. Conf. on Intelligent Robots and Systems, 2011. (Cited in page 74.)
- [Perrin 2012] N. Perrin, O. Stasse, L. Baudouin, F. Lamiriaux and E. Yoshida. *Fast humanoid robot collision-free footstep planning using swept volume approximations*. Transactions on Robotics, 2012. (Cited in pages 23, 24, 36 and 104.)
- [Perrin 2015] N. Perrin, D. Lau and V. Padois. *Effective Generation of Dynamically Balanced Locomotion with Multiple Non-coplanar Contacts*. In the International Journal of Robotics Research, 2015. (Cited in page 15.)
- [Pham 2007] Q. Pham and J. E. Slotine. *Stable concurrent synchronization in dynamic system networks*. Neural Networks, 2007. (Cited in page 76.)
- [Pham 2009] Q. Pham, S. Caron and Y. Nakamura. *Kinodynamic Planning in the Configuration Space via Admissible Velocity Propagation*. In Robotics System and Science, 2009. (Cited in page 83.)
- [pin 2016] *Pinocchio official website. This C++ template library propose fast algorithm to compute the dynamics of rigid body kinematic chains*. <http://stack-of-tasks.github.io/pinocchio>, 2016. Accessed: 2015-06-28. (Cited in page 119.)

- [Qiu 2011] Z. Qiu, A. Escande, A. Micaelli and T. Robert. *Human motions analysis and simulation based on a general criterion of stability*. In International Symposium on Digital Human Modeling, 2011. (Cited in page 15.)
- [Ramirez-Alpizar 2016] I. G. Ramirez-Alpizar, M. Naveau, C. Benazeth, O. Stasse, J.-P. Laumond, K. Harada and E. Yoshida. *Motion Generation for Pulling a Fire Hose by a Humanoid Robot*, 2016. (Submitted). (Not cited.)
- [Ramos 2015] O. E. Ramos, N. Mansard, O. Stasse, C. Benazeth, S. Hak and L. Saab. *Dynamic Whole Body Motion Generation for the Dance of a Humanoid Robot*. Robotics and Automation Magazine, 2015. (Cited in page 95.)
- [Razavi 2015] H. Razavi, A. M. Bloch, C. Chevallereau and J. W. Grizzle. *Restricted discrete invariance and self-synchronization for stable walking of bipedal robots*. In American Control Conference, ACC 2015, Chicago, IL, USA, July 1-3, 2015, 2015. (Cited in page 12.)
- [Reher 2016] J. Reher, E. A. Cousineau, A. Hereid, C. M. Hubicki and A. D. Ames. *Realizing Dynamic and Efficient Bipedal Locomotion on the Humanoid Robot DURUS*. 2016. (Cited in page 12.)
- [Rosenbaum 2008] D. A. Rosenbaum. *Reaching while walking: reaching distance costs more than walking distance*. Psych. Bull. Rev., 2008. (Cited in page 88.)
- [Rotella 2015] N. Rotella, A. Herzog, S. Schaal and L. Righetti. *Humanoid Momentum Estimation Using Sensed Contact Wrenches*. In Int. Conf. on Humanoid Robotics, 2015. (Cited in page 15.)
- [S. Faraji 2014] C. G. A. S. Faraji S. Pouya and A. J. Ijspeert. *Versatile and robust 3D walking with a simulated humanoid robot (Atlas): A model predictive control approach*. In Int. Conf. on Robotics and Automation, 2014. (Cited in page 24.)
- [Seo 2010] K. Seo, S. Chung and J. E. Slotine. *CPG-based control of a turtle-like underwater vehicle*. Autonomous Robots, 2010. (Cited in page 75.)
- [Sherikov 2014] A. Sherikov, D. Dimitrov and P.-B. Wieber. *Whole body motion controller with long-term balance constraints*. In Int. Conf. on Humanoid Robotics, 2014. (Cited in pages 15 and 24.)
- [Sherikov 2016] A. Sherikov. *Balance preservation and task prioritization in whole body motion control of humanoid robots*. PhD thesis, INRIA, 2016. (Cited in pages 11 and 14.)
- [Shoulson 2014] A. Shoulson, N. Marshak, M. Kapadia and N. Badler. *ADAPT: The Agent Development and Prototyping Testbed*. IEEE Transaction on Visualization and Computer Graphics, 2014. (Cited in page 88.)
- [Stasse 2006] O. Stasse, A. J. Davison, R. Sellaouti and K. Yokoi. *Real-time 3d slam for humanoid robot considering pattern generator information*. In Int. Conf. on Intelligent Robots and Systems, 2006. (Cited in page 75.)

- [Stasse 2008] O. Stasse, B. Verelst, A. Davison, N. Mansard, F. Saidi, B. Vanderborght, C. Esteves and Y. K. *Integrating Walking and Vision to increase Humanoid Autonomy*. the International Journal of Humanoid Robotics, 2008. (Cited in page 89.)
- [Stasse 2009] O. Stasse, P. Evrard, N. Perrin, N. Mansard and A. Kheddar. *Fast foot prints replanning and motion generation during walking in physical human-humanoid interaction*. In Int. Conf. on Humanoid Robotics, 2009. (Cited in page 107.)
- [Stasse 2013] O. Stasse. Habilitation thesis. Paul Sabatier University, CNRS, Toulouse, 2013. (Cited in page 93.)
- [Stasse 2014] O. Stasse, A. Orthey, F. Morsillo, M. Geisert, N. Mansard, M. Naveau and C. Vassallo. *Airbus/Future of Aircraft Factory, HRP-2 as Universal Worker Proof of Concept*. In IEEE/RAS International Conference on Humanoid Robot (ICHR), 2014. (Cited in page 103.)
- [Taïx 2013] M. Taïx, M. Tran and E. Souères P. Guigon. *Generating human-like reaching movements with a humanoid robot: A computational approach*. Journal of Computational Science, 2013. (Cited in page 89.)
- [Takubo 2005a] T. Takubo, K. Inoue and T. Arai. *Pushing an object considering the hand reflect forces by a humanoid robot in dynamic walking*. In Int. Conf. on Robotics and Automation, 2005. (Cited in page 61.)
- [Takubo 2005b] T. Takubo, K. Inoue and T. Arai. *Pushing operation for humanoid robot using multipoint contact states*. In Int. Conf. on Intelligent Robots and Systems, 2005. (Cited in page 61.)
- [Tassa 2012] Y. Tassa, T. Erez and E. Todorov. *Synthesis and stabilization of complex behaviors through online trajectory optimization*. In Int. Conf. on Intelligent Robots and Systems, 2012. (Cited in page 88.)
- [Tassa 2014] Y. Tassa, N. Mansard and E. Todorov. *Control-limited differential dynamic programming*. In Int. Conf. on Robotics and Automation, 2014. (Cited in page 14.)
- [Tedrake 2015] R. Tedrake, S. Kuindersma, R. Deits and K. Miura. *A closed-form solution for real-time ZMP gait generation and feedback stabilization*. In Int. Conf. on Humanoid Robotics, 2015. (Cited in page 24.)
- [Todorov 2002] E. Todorov and M. Jordan. *Optimal feedback control as a theory of motor coordination*. Nature neuroscience, 2002. (Cited in page 83.)
- [Todorov 2014] E. Todorov. *Analytically-invertible dynamics with contacts and constraints: Theory and implementation in MuJoCo*. In Int. Conf. on Robotics and Automation, 2014. (Cited in pages 13 and 43.)
- [Tonneau 2015a] S. Tonneau, N. Mansard, C. Park, D. Manocha, F. Multon and J. Pettre. *A Reachability-based planner for sequences of acyclic contacts in cluttered environments*. In the International Symposium on Robotics Research, 2015. (Cited in page 45.)

- [Tonneau 2015b] S. Tonneau, N. Mansard, C. Park, D. Manocha, F. Multon and J. Pettr . *A Reachability-based planner for sequences of acyclic contacts in cluttered environments*. In the International Symposium on Robotics Research, 2015. (Cited in page 50.)
- [Torricelli 2015] D. Torricelli, J. F. V. Vargas, K. Mombaur, N. Tsagarakis, A. J. del Ama, A. Gil-Agudo, J. C. Moreno and J. L. Pons. *Benchmarking Bipedal Locomotion: A Unified Scheme for Humanoids, Wearable Robots, and Humans*. Robotics and Automation Magazine, 2015. (Cited in page 4.)
- [Viviani 1992] P. Viviani and N. Stucchi. *Biological movements look uniform: evidence of motor-perceptual interactions*. Journal of experimental psychology. Human perception and performance, 1992. (Cited in pages 73 and 77.)
- [Wang 2008] J. M. Wang, D. J. Fleet and A. Hertzmann. *Gaussian Process Dynamical Models for Human Motion*. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2008. (Cited in page 88.)
- [Weigelt 2010] M. Weigelt and T. Schack. *The Development of End-State Comfort Planning in Preschool Children*. Exper. Psych., 2010. (Cited in page 88.)
- [Wensing 2014] P. M. Wensing and D. E. Orin. *3D-SLIP Steering for High-Speed Humanoid Turns*. In Int. Conf. on Intelligent Robots and Systems, 2014. (Cited in page 25.)
- [Westervelt 2007] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi and B. Morris. *Feedback control of dynamic bipedal robot locomotion*, volume 28. CRC press, 2007. (Cited in page 13.)
- [Wieber 2002] P.-B. Wieber. *On the stability of walking systems*. In Proc. of the Inter. Workshop on Humanoid and Human Friendly Robotics, 2002. (Cited in page 30.)
- [Wieber 2005] P.-B. Wieber. *Somme comments on the structure of the dynamics of articulated motion*. In Fast Motions in Biomechanics and Robotics, 2005. (Cited in page 11.)
- [Wieber 2008] P.-B. Wieber. *Viability and predictive control for safe locomotion*. In Int. Conf. on Humanoid Robotics, 2008. (Cited in page 10.)
- [Wieber 2015] P.-B. Wieber, R. Tedrake and S. Kuindersma. *Handbook of robotics, chapitre Modeling and Control of Legged Robots*. 2015. (Cited in page 10.)
- [Yoshida 2007] E. Yoshida, A. Mallet, F. Lamiriaux, O. Kanoun, O. Stasse, M. Poirier, P.-F. Dominey, J.-P. Laumond and K. Yokoi. *‘Give me the Purple Ball’ – he said to HRP-2 N.14*. In Int. Conf. on Humanoid Robotics, 2007. (Cited in page 89.)

---

## Résumé en français:

Cette thèse traite du problème de la locomotion des robots humanoïdes dans le contexte du projet européen KoroïBot. En s'inspirant de l'être humain, l'objectif de ce projet est l'amélioration des capacités des robots humanoïdes à se mouvoir de façon dynamique et polyvalente. Le cœur de l'approche scientifique repose sur l'utilisation du contrôle optimal, à la fois pour l'identification des coûts optimisés par l'être humain et pour leur mise en œuvre sur les robots des partenaires roboticiens. Cette thèse s'illustre donc par une collaboration à la fois avec des mathématiciens du contrôle et des spécialistes de la modélisation des primitives motrices.

Les contributions majeures de cette thèse reposent donc sur la conception de nouveaux algorithmes temps-réel de contrôle pour la locomotion des robots humanoïdes avec nos collègues de l'université d'Heidelberg et leur intégration sur le robot HRP-2. Deux contrôleurs seront présentés, le premier permettant la locomotion multi-contacts avec une connaissance a priori des futures positions des contacts. Le deuxième étant une extension d'un travail réalisé sur de la marche sur sol plat améliorant les performances et ajoutant des fonctionnalités au précédent algorithme. En collaborant avec des spécialistes du mouvement humain nous avons implémenté un contrôleur innovant permettant de suivre des trajectoires cycliques du centre de masse. Nous présenterons aussi un contrôleur corps-complet utilisant, pour le haut du corps, des primitives de mouvements extraites du mouvement humain et pour le bas du corps, un générateur de marche. Les résultats de cette thèse ont été intégrés dans la suite logicielle "Stack-of-Tasks" du LAAS-CNRS.

**Mots clés:** Locomotion bipède et humanoïdes, Robots humanoïdes, Commande prédictive non linéaire

---

## Abstract:

This thesis covers the topic of humanoid robot locomotion in the frame of the European project KoroïBot. The goal of this project is to enhance the ability of humanoid robots to walk in a dynamic and versatile fashion as humans do. Research and innovation studies in KoroïBot rely on optimal control methods both for the identification of cost functions used by human being and for their implementations on robots owned by roboticist partners. Hence, this thesis includes fruitful collaborations with both control mathematicians and experts in motion primitive modeling.

The main contributions of this PhD thesis lies in the design of new real time controllers for humanoid robot locomotion with our partners from the University of Heidelberg and their integration on the HRP-2 robot. Two controllers will be shown, one allowing multi-contact locomotion with a prior knowledge of the future contacts. And the second is an extension of a previous work improving performance and providing additional functionalities. In a collaboration with experts in human motion we designed an innovating controller for tracking cyclic trajectories of the center of mass. We also show a whole body controller using upper body movement primitives extracted from human behavior and lower body movement computed by a walking pattern generator. The results of this thesis have been integrated into the LAAS-CNRS "Stack-of-Tasks" software suit.

**Key words:** Humanoid and Bipedal Locomotion, Humanoid Robots, Nonlinear Model Predictive Control