Emergence of complex behaviors from coordinated predictive control in humanoid robotics

Aurélien Ibanez

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

of the

Pierre & Marie Curie University
Sorbonne Universités, Université Pierre & Marie Curie Paris VI

Doctoral School of Mechanical, Acoustic, Electronic and Robotic Sciences of Paris
École Doctorale de Sciences Mécanique, Acoustique, Électronique et Robotique de Paris

in

Robotics

presented by

Aurélien IBANEZ

Emergence of complex behaviors from coordinated predictive control in humanoid robotics

---

Defended on the 25th of September, 2015

Committee

M.	Yoshihiko	Nakamura	Professor at University of Tokyo	Referee
M.	Bruno	Siciliano	Professor at Universitá degli Studi di Napoli Federico II	Referee
M.	Faiz	Ben Amar	Professor at Université Pierre & Marie Curie	Examiner
M.	Christian	Ott	Department Head at German Aerospace Center (DLR)	Examiner
M.	Ludovic	Righetti	Group Leader at Max Planck Institute for Intelligent Systems	Examiner
M.	Philippe	Bidaud	Professor at Université Pierre & Marie Curie	Adviser
M.	Vincent	Padois	Associate professor at Université Pierre & Marie Curie	Adviser

Abstract

Rising to the challenge of motor control for systems involved in multi-objective and highly-constrained activities is a requirement to enable the emergence of efficient and robust behaviors. Conflicts between the objectives and constraints defining the activity indeed require the elaboration of complex coordination strategies to ensure performance, feasibility and safety. Such performance and safety concerns are particularly relevant to *humanoid systems*, being explicitly designed to evolve in human environments.

Anticipating the behavior of the system and formulating an objective-driven control problem, *predictive control* provides means to meet such requirements. Although multi-objective predictive approaches enable the definition of complex and constrained strategies *coordinating* the motor activity of the system, their computational cost is a critical drawback from practical applications.

In this perspective, the work presented in this dissertation aims at considering multi-objective predictive control for feasible and practical applications.

A control architecture is proposed to this purpose as a *multi-objective*, *two-layered* controller exploiting the respective advantages of predictive and instantaneous formulations.
Multiple objectives are considered at the predictive level with respect to *reduced models*, allowing the formulation of computationally-efficient optimization problems, to elaborate *complex coordination strategies* conforming objectives to constraints in an anticipated manner, at a cheapened cost.
Feasibility of these strategies is then enforced at the instantaneous level with a multi-objective whole-body controller employing a more accurate model of the controlled system and its constraints, to finally output an optimal control policy of its motor activity.

The contribution of this work takes the form of the validation of the benefits from such an approach in its development for *practical* challenges and applications of humanoid robotics. Complex coordination strategies are effectively *emerging* in simulation and real-time implementation with various humanoids, such as the iCub and TORO robots and virtual human models, providing increases in performance and robustness to the execution of complex activities.

The computational demand of these developments is contained with the introduction of reduced multi-objective models, enabling computationally-favorable formulations of the control problem and its *distribution*. Indeed, model reductions are performed in this work to lower the cardinality, enforce advantageous expressions and allow sequential and parallel distributions of critical multi-objective control problems, therefore offering opportunities to reduce the computational cost of the predictive level.

Despite the resulting approximations on the dynamics of the system at the predictive level, complex behaviors are emerging, exploiting elaborate coordination strategies between conflicting objectives and constraints to increase performance and robustness against disturbances.

Keywords: *Humanoid Robotics, Whole-Body Dynamic Control, Distributed Model Predictive Control, Mixed-Integer Programming, Biped Balance and Walking, Task-posture Coordination.*

Acknowledgements

I would like to first thank my advisers Philippe Bidaud and Vincent Padois for their appreciated support, tutoring, expertise and insights. Over the course of this Ph.D, I undoubtedly learnt a lot from our discussions and now consider them as valuable friends. Thank you for your continuous support.

Special thanks go to Joseph Salini which was always available to help me on virtually any topic, and more specifically for introducing optimization-based control formulations to me.

Thanks to Pauline Maurice for her rigorous mind to which I would recurrently confront my sometimes loose ideas, and to Nicolas Perrin for reviewing my work and for introducing me to new ways of formulating ideas.

More generally, I am thankful to the Institute for Intelligent Systems and Robotics for its welcome and resources, and to the French Ministry of Higher Education and Research and the Ecole Normale Supérieure de Cachan for supporting my Ph.D.

Thanks to the German Aerospace Center, DLR, for providing me with the opportunity to implement my work on their TORO robot. This was made possible in particular thanks to Christian Ott and his team. Thanks to Johannes Englsberger, Bernd Henze and Alexander Werner for devoting so much of their time to my stay.

I would like to particularly thank my family who provided me with everything I needed to pursue my goals, especially their love and support. Thanks to my friends who made the effort to support and care about my work — or at least pretend to.

This thesis is dedicated to Sarra, to whom I am deeply grateful for bringing me so much love and happiness. Her constant support is and has been of immense value to me. Merci infiniment.

Table of contents

List of Figures

Table of notations and acronyms

Unless explicitely stated otherwise, scalar variables are in lower case, vectors are bold lower case symbols and matrices are upper case bold symbols.
The operator $<$ compares two scalar values and $\prec$ compares two vectors term-to-term.
An euclidean norm is denoted $\|\bullet\|$.
Operations denoted $\mathbf{a} \cdot \mathbf{b}$ and $\mathbf{a} \times \mathbf{b}$ are respectively the dot and cross products between $\mathbf{a}$ and $\mathbf{b}$.
$\mathbb{N}$ is the set of non-negative integers and $\mathbb{R}$ the set of real numbers.

APA	Anticipated Postural Adjustments
BB	Branch and Bound
BoS	Base of Support
CoM	Center of Mass
CoP	Center of Pressure
CP	Capture Point
CPU	Central Processing Unit
DMC	Dynamic Matrix Control
DMPC	Distributed Model Predictive Control
DS	Double Support
FRI	Foot Rotation Indicator
FTS	Force-Torque Sensor
GPC	Generalized Predictive Control
GPU	Graphics Processing Unit
HPC	High-Performance Computing
HRI	Human-Robot Interaction
IDCOM	IDentification and COMmand
IMU	Inertial Measurement Unit
LIP	Linear Inverted Pendulum
LP	Linear Program
MIMO	Multiple Inputs - Multiple Outputs
MIP	Mixed-Integer Programming
MIQP	Mixed-Integer Program
MPC	Model Predictive Control
MPHC	Model Predictive Heuristic Control
MSD	MuskuloSkeletal Disorder
ODE	Ordinary Differential Equation
PID	Proportional Integral Derivative
QP	Quadratic Program
RHC	Receding Horizon Control
SS	Single Support
ZMP	Zero-Moment Point

CHAPTER 1

Predictive control and its motivation in humanoid robotics

The timeline of human history is paved with the design of tools and machines of increasing complexity; their growing influence eventually leading to go beyond the difference between living bodies and their lifeless counterparts, with the universal concept of *mechanism* underlain by the cause and effect principle. Similarly to the mechanisms of nature, man-made machines serve an effect governed by the internal and external influences *of* and *on* its parts; these mechanisms are apprehended as *dynamical systems*, with states determined by an history of actions. One of the most interdisciplinary current fields of research, *control theory*, henceforth emerged with the aim to exploit this causality by *controlling* the effect of a dynamical system through the manipulation of internal and external interactions.

Control theory in robotics and automation mainly focuses on the modification of the behavior of these dynamical systems by *feedback*. Since the knowledge of mechanisms is generally incomplete, uncertainties — parametric or epistemic — in their deterministic models, as dynamical systems, are unavoidable: their behavior, captured as a set of *outputs*, is therefore compared to a desired one and this difference is employed by a *controller* to compute a new set of actions, or *inputs*, *fed back* to the system with the expectation to correct the outputs in an appropriate direction.

Applications of these systems are motivated by *objectives* to be reached: their controller must accordingly drive the system towards states which achieve these goals. In robotics, these objectives are commonly related to the environment, with sensor-based control and task functions for example which establish a virtual link between the robot and the environment; objectives are therefore subject to changes, relatively to the evolution of the environment. Objectives can moreover be specified at different levels, depending on the application. Indeed, a self-driving car for example globally aims at reaching a specific location; this objective is however commonly reduced to smaller space and time scales: an itinerary is planned *a priori* and the activity objective is decomposed into a set of intermediate target locations to reach over time, ultimately leading to the initial global objective.

Additionally, the system and its application are commonly subject to *constraints*, issued from safety, specification or intrinsic — physical, for example — limitations. The trajectory of the dynamical system towards its objectives must comply with these constraints, and the control actions from the controller therefore should not induce their violation. In the example of the self-driving car, a safety constraint would consist in maintaining the car within the boundaries of the road, specification of the activity could impose a maximum allowed velocity and a maximum steering angle may physically constrain the system.

For performance and safety reasons, the controller must thus concurrently consider the objectives and constraints of the activity. *Optimal control* provides a mathematical framework particularly suitable to the formulation of controllers under these requirements. Indeed, control policies are derived from an optimization problem which allows through its form to explicitly write the control problem as objectives to be reached under a certain set of constraints. Optimal control is in such a way an ideal candidate to formulate the control problem consistently with the specification of the desired system behavior.

The presence of constraints in the control problem however raises the issue of *suboptimality* and *viability*. They indeed define an admissible domain of control solutions which is not necessarily convex in the general case, depending on the system and application characteristics. The controlled system being a dynamical system, its evolution is a continuous trajectory over time: the set of constraints and objectives can therefore lead to inextricable states or regions, as illustrated in the following section, since not all transitions between admissible states are admissible. These regions can contain a suboptimal equilibrium, therefore preventing the system from reaching the desired global optimum. Additionally, these steady regions can over time be excluded from the admissible domain due to potential evolutions of the constraints. The system is henceforth in a *non-viable* state where constraints violation is unavoidable.
Considering this *future evolution* of the dynamical system with respect to the set of constraints and its changes is therefore a requirement to guarantee the optimality and admissibility of the control solutions over time.

The set of objectives can furthermore be temporarily conflicting. In the example of the self-driving car, the system might be designed to concurrently minimize both the travel distance and duration. However, in cases of traffic jam the travel distance objective must be temporarily sidelined in favor of the duration goal to achieve the best overall performance.
While writing the control problem as a multi-objective optimization problem allows to leave to the controller the decision of the appropriate compromise between objectives instead of specifying an *a priori* itinerary to follow, the preview of the *future outcomes* of this control policy provides additional information and degrees of freedom to elaborate the appropriate control strategy to achieve the best performance.

Predictive control suitably proposes to write the control problem under these considerations. Predictive control denotes a subclass of optimal control techniques which takes advantage of a receding, future preview horizon to predict the system output and compute the appropriate instantaneous control input.

An analogy between this control approach and anticipative mechanism humans employ to rapidly decide their reaction can be made: figure 1.1 illustrates for example how, employing a learned behavior model of his car, an experienced driver can anticipate an escape trajectory on the occurrence of an obstacle, while keeping control of its vehicle and staying at best within the bounds of the road. Conversely, a purely reactive behavior might lead to a sudden divergence of the vehicle without consideration of potential future car instability, lane departure

or obstacle motion.

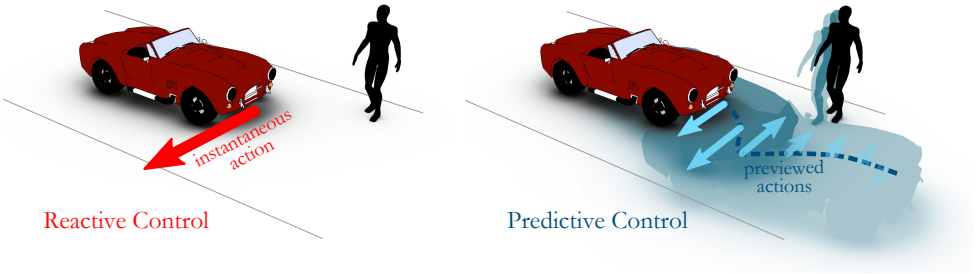

Figure 1.1.: Illustration of predictive approaches. Anticipation of the behavior of the vehicle and obstacles allows to preview an optimal escape trajectory and avoid sudden and potentially infeasible or unstable reactions.

The control problem behind these techniques is formulated as an optimization problem over a future time window, describing an optimal control strategy as a future horizon of control inputs with respect to a performance criterion. Only the first step of this policy is applied to the system, and the calculations are repeated at each control step.

The exploitation of a future time window, along with the optimization form of the control problem, allows to explicitly account for constraints and produce control strategies over time within their bounds; these strategies optimally coordinate the control degrees of freedom with respect to the objectives of the activity, in order to achieve the best feasible performance regarding these goals. Desired behaviors are therefore naturally specified through the objectives and constraints they are composed of, and the preview horizon enables the elaboration of more complex strategies.

Following these remarks, this chapter first highlights the major contributions of predictive control to the resolution of multi-objective and constrained control problems. A discussion on how these contributions are particularly relevant to the control of humanoid systems is then conducted. The control problem of complex activities for humanoids is henceforth stated to identify the challenge this work takes on.

1.1. Predictive control as a means to safely conform goals to constraints

Originally introduced for the control of industrial processes [Richalet1978], predictive control was rapidly developed and accepted as one of the superior adaptive control techniques for a plant which has large dynamic variations [Clarke1987]. When employing a model of the controlled process, predictive control is one of the single methodologies to handle constraints in a systematic way during the design and implementation of the controller [Garcia1989]. Its versatility in models used and the explicit formulation and use of a performance criterion made predictive control specifically successful in the chemical process industries at first, and was rapidly adapted to diverse control applications [Lee2011] with

highly dynamic, multi-degrees of freedom, free-floating and constrained systems in aerospace [Kale2005, Van2006], robotics [Shim2003, Klanvcar2007, Tassa2012] and computer graphics [Mordatch2012].

In humanoid robotics, predictive control has been mainly applied to the biped balance control problem. Kajita *et al.* first introduced the Zero-Moment Point (ZMP) preview control in [Kajita2003] to efficiently handle the unstable dynamics inherent to biped walking by previewing reduced dynamics of the system over a preview horizon. Wieber engaged a significant step forward in [Wieber2006], taking advantage of the predictive control ability to account for constraints. By extending the ZMP preview control with constraints on the Center of Pressure (CoP), the system was constricted to viable states.
The predictive walking control problem consists in these works in finding an optimal trajectory of the Center of Mass (CoM) of the robot that will avoid tip-over configurations. The model versatility in predictive control allowed Diedam *et al.* and Herdt *et al.* to append feet positions to this control problem in [Diedam2008, Herdt2010] along with a second criterion of desired CoM velocity, thus allowing for the computation of optimal steps to take, given a predefined gait time-pattern.

1.1.1. Defining activities at a higher level

Predictive control inherits one of the main assets of optimal control: rather than defining control policies as behaviors to track as in classical feedback controllers, the desired motor activity is specified through explicit objectives to be reached. Such an approach allows for a greater variety of resulting behaviors as it prevents from inputting excessive *a priori* information in the control system.

The desired activity is indeed formulated as an optimization problem to be solved whereas feedback controllers generally constrain the control solutions to a specific, given form; tasks to perform are thus defined at a higher-level, therefore alleviating the liability of pre-specifying unrequired restraints on control policies. The expected solutions can nevertheless be explicitly constrained to an admissible domain when necessary: the optimization form of the control problem naturally allows to consider constraints on the desired control policy.
This high-level specification therefore defines the activity explicitly with its objectives and constraints. Without enforcing a definite form of the motor activity, the resulting increase in the dimension of the admissible solutions space and its explicit confinement are beneficial to the performance and safety of the tasks execution in the case of unforeseen or unexpected scenarii.

Coupled to the employment of a future time window, the high-level specification of activities releases an even greater degree of freedom in finding solutions to a complex control problem.

This greater degree of freedom however induces a significant increase in the computational cost of MPC approaches, which can be critical for some applications. Nevertheless, multiple developments can be found in the literature dealing with the computational issue of MPC problems. Distributed MPC approaches for example provides tools to distribute the computational load [Christofides2013] and explicit MPC proposes a reformulation of constrained MPC problems to derive explicit solutions, as introduced by Bemporad *et al.* first in the case of

linear MPC [Bemporad2002a]. It is worth noticing that the complexity of the MPC problem is also related to the form and dimension of the model employed. MPC approaches in robotics usually employ linearized models as reduced models, and more generically methods such as model-order reduction allow to reduce the complexity of the model, as illustrated in [Hovland2006] on large-scale systems.

MPC approaches raises further challenges which receive an increasing attention over the years. Despite their intrinsically stable and safe design, the closed-loop stability and the feasibility concerns of MPC problems are subject to deeper analysis. Details on the stability of MPC problems can be found in [Lazar2007] and the feasibility issue of MPC problems is addressed in [Kerrigan2001] for example.

1.1.2. Anticipating non-viable and suboptimal steady states

The ability of predictive control to preview constraints on the system and react accordingly is the one of the most relevant reasons to adopt predictive control. Generally being highly redundant systems, humanoids for example are often required to execute multiple tasks simultaneously, such as performing manipulation while maintaining balance. These tasks, however, might be conflicting as simply illustrated with biped walking: in order to move forward, balance must be temporarily conceded, and conversely. Moreover, these tasks must be executed while respecting a complex set of constraints issued from safety reasons, induced from the evolution of the robot in populated environments, and from the physical limitations of the robot (*e.g.* joint and actuator limits, slippage and balance dynamics constraints, collision-free paths). The conjoint effect of conflicts between objectives and constrained evolution can potentially lead to suboptimal steady states, and even non-viable states, as illustrated in figure 1.2. In this example,

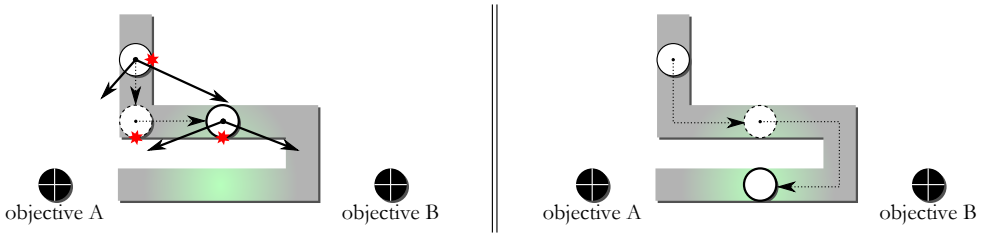

Figure 1.2.: Reaching conflicting objectives under constraints. *From left to right* — reactive and predictive approaches. Instantaneously minimizing the greatest error, a reactive approach tends to draw the system towards objective B until a constraint is hit and no further motion is possible. In the predictive method, the system temporarily renounces to objective A to eventually reach a consensus between A and B.

the system must reach two conflicting objectives A and B of equal priority, while constrained to given path boundaries. A reactive approach without additional *ad hoc* or heuristic rules, simply consisting into driving the motion towards the minimization of the greatest error, will tend to draw the system from its initial state towards the furthest objective, *i.e.* B, in order to reach a consensus between A and B. Such a policy will bring and maintain the system in an inextricable state in collision with the constraints, being largely suboptimal. A

predictive strategy with a long enough horizon would however preview the evolution of the constraints, and infer an optimal path involving a temporary relaxation of the objective A in favor of B to finally recover and reach the global optimum with respect to A and B. Furthermore, evolution of constraints can drive the system from this suboptimal equilibrium to a situation where constraint violation is unavoidable; the equilibrium state is thus non-viable. For example, if the admissible domain is progressively contracted around the optimum over time, all suboptimal solutions are non-viable while the problem is still feasible; previewing the evolution of constraints would however permit to maintain admissibility of the solution. Changes in the control strategy can nevertheless be envisioned when reaching non-viable states in a reactive framework to ensure safe alternative behaviors [Rubrecht2012].

In this sense predictive control allows to conform objectives to constraints while preventing suboptimal equilibrium situations and non-viable states, and thus guarantee — under the validity condition of the problem hypotheses — the safety of the adopted control policy. Note that the model information in the predictive control problem can also be apprehended as a set of constraints, and predictive approaches thus account for the system dynamics and limitations when inferring an optimal control strategy. This consideration strongly augments the control policy safety in the case of complex and unstable systems, where predictive control has the capacity to recover from unexpected disturbances with the preview of the system future behavior and the computation of an appropriate strategy with respect to the expected system behavior and imposed constraints.

The definite suitability of predictive control to applications where multiple conflicting objectives are to be reached safely, under a complex set of evolving constraints, distinctly motivates its application to humanoid robotics. Humanoid systems are indeed commonly employed in complex activities, composed of several elementary operations and subject to severe constraints, in changing and disturbance-prone environments.

Humanoids are implicitly designed, through their human-like structure, to interact with human environments. This physical involvement in complex and populated environments brings a critical requirement for *safety*, which implies performance and robustness prerequisites with respect to the specified activities. These activities generally consist in multi-task operations, restrained by an evolving and demanding set of constraints and subject to external disturbances.
The underlying control requirements of such activities are distinctly met by predictive control techniques, which makes them relevant candidates to support the control of humanoids motor activity.

Their computational cost however requires a specific attention to the formulation of the control problem and the implementation of its resolution to envision real-life applications.

1.2. From the motivation behind collaborative robots to the need for predictive control

Behind *humanoid robotics* lies a vast range of fields and applications with diffuse, and almost inexistent frontiers: the coalescence of knowledge issued from cognitive, biomechanical, electronic, computer, automation and mechanical sciences is employed in addressing factual and anticipated challenges, centered around the objective of *assisting the human*.
While electronic, computer, automation and mechanical developments contribute to the set of tools we possess to craft robotic solutions, cognitive and biomechanical sciences provide means to not only understand human activity but also identify mechanisms which can inspire the robotic approaches, and lead to efficient robot design and control.

The *assistive* function of robots nowadays spans a wide range of nature. Although developments of applications where robots replace the humans in harmful environments or tasks are still spreading, recent advances in robotics along with current societal and economic issues drew the emergence of a *collaborative* robotic assistance. As illustrated in figure 1.3, the growing interest and fast development of collaborative robotics already allow the introduction of industrial-grade solutions on production processes.

Figure 1.3.: Industrial-grade collaborative robots. *From left to right* — KUKA IIWA, Universal Robots UR5, ABB YuMi, Audi PART4you.

The developments in haptic technologies and soft robotics, along with the miniaturization and growing efficiency of actuators and power sources provided tools supporting the progress of two robotics forks: *collaborative* robotics and *assistive* robotics to address unemployment, competitiveness and population aging challenges. *Collaborative* robotics focus on the integration of safe robots able to cooperate and interact with human partners while *assistive* robotics mainly comprise wearable rehabilitation devices such as orthoses and exoskeletons.

Both applications inevitably imply the necessity of a durable, flexible and predominantly *safe* physical interaction between the robot and its human partner; this statement is *a fortiori* legitimate for humanoid robotics and its applications, being explicitly designed to interact with human environments.

Achieving safe interaction between human and robots is mainly twofold: both mechanical design and control have a strong influence on the degree of harm a robot can cause. Indeed, as further discussed in [DeSantis2008], minimizing injury risk may not only come from increasingly lightweight and compliant mechanisms but also from consistent, adaptive and anticipative control. Control and

planning nevertheless strongly rely on the perception and understanding of both the environment and the human activity.

Predictive control techniques offer tools which can provide control with this anticipative characteristic, and additionally help deeper perception and further comprehension of the human activity as it allows for the robust control of virtual humans. In this sense, predictive control can not only strongly contribute to the development of robust and efficient control of humanoid robots, but also ultimately allow for *safe* collaborative interaction between robots and humans.

1.2.1. Humanoid robotics: a wide span of applications and opportunities

Despite its fast expansion, the recent origin of humanoid robotics and its ambitious goals makes it a challenging field. By definition, humanoid robots are mechanisms with a structure which resembles the human body. By extension however, the field of humanoid robotics spreads to systems *partially* echoing the human structure — such as upper or lower bodies solely — which heavily broadens the scope of possible applications. Indeed, while fully humanoid mechanisms are various and innovative developments are still appearing (see figure 1.4 for examples), *partial* humanoids are even more diverse in structure and function as illustrated in figure 1.5.
The range applications of humanoid robotics is thus vast and growing.

Figure 1.4.: Humanoid robots and their various shape and applications. *From left to right* — Asimo (home assistant, Honda, 2004), iCub (cognitive studies, RobotCub & IIT, 2004), TORO (interaction with the environment, DLR, 2013), Atlas (disaster response, Boston Dynamics, *ca.* 2013), SCHAFT (disaster response, SCHAFT, *ca.* 2013).

The structure of humanoid systems fundamentally allows them to interact with human environments. This ability is widely used in various contexts for diverse applications: ASIMO for example is presented with the aim to "genuinely help people" and "help with important tasks like assisting the elderly or a person confined to a bed or a wheelchair"[1]. Its objective is clearly stated as being an assistant to persons in their everyday life. Their human-like structure is also employed in more severe applications as demonstrated in the DARPA's Robotics

[1] `http://asimo.honda.com/asimo-history`

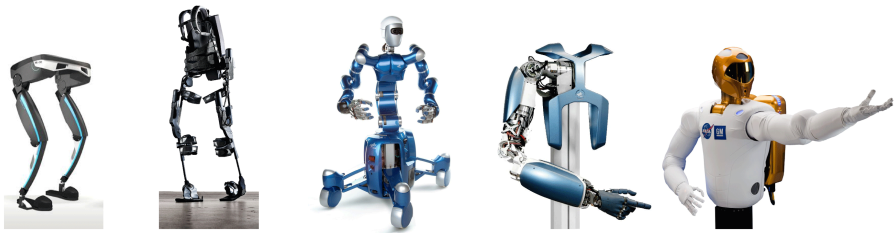

Figure 1.5.: Diverse partially humanoid systems and their applications. *From left to right* — Hercule (assistance, RB3D), Ekso (rehabilitation, Ekso Bionics), Rollin' Justin (manipulation in human environments, DLR), Hand Arm System (lightweight human-like manipulation, DLR), Robonaut (dexterous manipulation, NASA).

Challenge[2] which confronts several humanoid robots to tasks involving interaction with human environment and devices in a disaster response scenario.

The familiar, human-like design of humanoids moreover allows humans to better understand and anticipate the robot behavior, thus offering solutions to both acceptability and safety issues. This benefit can furthermore be enhanced by the implementation of natural, human-like control techniques. The terms *human-like* and *natural* control generally refers in this chapter to techniques either inspired by principles observed in humans or producing motion and behaviors similar in appearance to the human.

Humanoid structures additionally bring research platforms to experiment and study biomechanical and cognitive hypotheses and observations. The iCub robot was for example designed to promote further developments in enactive cognition [Sandini2007], with the hypothesis that the internal model of the world in human cognition is strongly influenced by the body structure. In the same perspective virtual humanoids can be employed to mimic humans in action and extract various indicators, for perception or evaluation purposes, in the assessment of musculoskeletal disorders (MSD) risk [Maurice2014] for example. Their contribution to the analysis of the human motion is demonstrated in various applications, as in [Demircan2012] with the characterization of golf swing motions from professional players, and to address health issues through the extraction of non-observable metrics with the estimation of the center of mass motion of the elderly [Cotton2011].

The concurrent use of these three assets — understanding of human motion and human-like structure and control — offers opportunities to tighten the link between the human and the robot. Wearable devices such as exoskeletons, orthoses and even artificial limbs indeed depend on a deep understanding of the principles governing human motor activity and on compatible mechanics and control.

1.2.2. Safety-driven performance challenges

A vast majority of these applications is designed to safely interact with populated human environments. With respect to control, the *safety* of this interaction can be defined from two point of views: safe *for the robot*, and safe *for the human*. While

[2] http://www.theroboticschallenge.org/

the robot safety relies predominantly on its ability to perform multiple tasks simultaneously without losing balance, the human safety is essentially dependent on the mutual understanding of each other's motion, and the capacity of the robot to appropriately react to the human activity.

The structure of the robot itself and the complexity of human activity brings an intricate set of constraints in which the control system must find a solution to reach its desired objectives while maintaining biped balance; the set of tasks to achieve moreover being possibly conflicting itself. Reacting to human activity might furthermore be insufficient: rapid changes in the situation might lead to dangerous and inextricable human-robot configurations. Providing control with anticipative characteristics would thus help ensure the safety of the human-robot interaction (HRI).

In this perspective, one of the main challenges which ensues from humanoid applications is the ability to accommodate at the control level, in an anticipated manner, potential conflicts between objectives and constraints the robot is subject to. Predictive control proposes appropriate techniques to address this challenge, enabling the computation of anticipated and constrained coordination strategies.

1.3. Further challenges

In a more focused perspective, safe HRI requires safe robot reactions to disturbances. While models of human activity allow to anticipate (and avoid) desired (or troublesome) interactions with humans, unexpected disturbances might come from flaws in perception, control and/or mechatronics. In the case of inherently dangerous systems or tasks, the recovery strategy adopted by the robot will define the safety degree of such unexpected interaction. Requirements are diverse depending on the system and application: maintaining balance will be predominant for massive robots and task performance will be imperative when executing dangerous tasks; constraint conformity will be prevalent in highly-constrained environments, systems or tasks.

Allowing to conform multiple objectives to constraints, predictive control is suitable to meet these requirements: balance can be maintained while still achieving optimal performance of concurrent and prioritized tasks under constraints. However, the increment in computational cost resulting from predictive approaches is one of the major deterrents, despite the developments in state-of-the-art problem formulations and resolution techniques. Indeed, efficient and safe control of humanoid robots requires whole-body motion to be regulated at fast rates. The dimension of predictive problems rapidly forbids their straightforward implementation on physical systems with real-time requirements. While multi-rate control architectures can be set up, they expect a specific attention drawn towards stability issues raised by inherent delays and asynchrony. Alternative approaches consist in distributing the main problem into sub-problems of lower cardinality to exploit the multi-core architecture of recent computing boards. High-performance computing (HPC) techniques indeed propose to exploit the parallel architecture of GPUs and CPUs (graphics/central processing units) to distribute heavy com-

putations. Nevertheless their application is still confined to a given set of problem classes, and new solver and problem formulations are yet to be explored.

From a more distant point of view, safe and efficient humanoid motor control supports the implementation of robust virtual human manikins. The contribution of predictive approaches to the performance and diversity of humanoid motion control, with the constrained, high-level specification of activities in an anticipative manner can significantly broaden their range of application. However, the high degree of freedom of virtual humans and the complexity of human activities raise several control challenges to take on.

1.4. Predictive control of complex humanoid activities

The control problem of humanoid robots can be interpreted as one of the three great challenges of humanoid robotics. The involvement of humanoids in real-life applications suggests the ability to handle complex activities while interacting with a changing and constrained environment. As illustrated in figure 1.6, these challenge can be organized as three interconnected major problems: perception, decision and motor control.

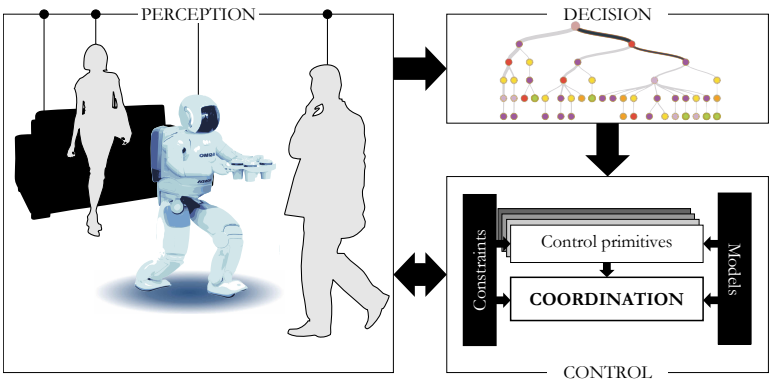

Figure 1.6.: Humanoid robotics as three major challenges: perception, decision and motor control.

From the control point of view, the humanoid is required to perform simultaneous elementary operations, such as walking and manipulating objects. The motion of the system is regulated in order to reach elementary objectives, which are expressed with respect to models with the lowest level of abstraction possible to reach the utmost feasible accuracy. The coordination of these elementary tasks is crucial to safety and performance, in order to ensure the respect of constraints the robot is subject to while exploiting the capacities of the whole-body. Complex activities furthermore generally require the system to be able to safely perform transitions between sets of tasks.

These sets of tasks and their transitions can be seen as selected and triggered by a decision layer, which generally exploits highly abstracted models of the system and the activity, allowing to generate a sequence of elementary operations which will lead over time to the global goal of the activity. This sequencing is achieved with respect to not only rules and objectives describing the activity to

perform, but also percepts from the current state of the system in its environment. These rules intrinsically carry anticipatory characteristics: they aim at driving the system over time towards the objective of the activity.

A perception layer is therefore required to extract information from the scene with percepts of various nature, from low-level interaction wrenches, robot state and obstacles location to higher-level cues in the case of human-crowded environments for example.

The works presented in this document mainly focus on the control component of the humanoid robotics challenge. Predictive control is considered as bridging the levels of abstraction of the decision and control layers, by outputting at the control level optimal and anticipated coordination strategies between objectives with respect to reduced models.

1.4.1. The control problem of complex activities

The control of multi-objective, constrained activities is commonly supported by the exploitation of a repertoire of motor activity primitives, handling the execution of elementary operations, which are coordinated at the instantaneous level to compute the whole-body motor inputs fed to the robot. The coordination process is one of the major requirements of the control problem of complex activities: it allows to enforce constraints and handles conflicts between motor primitives. The resulting control architecture can thus be interpreted as depicted on figure 1.7.

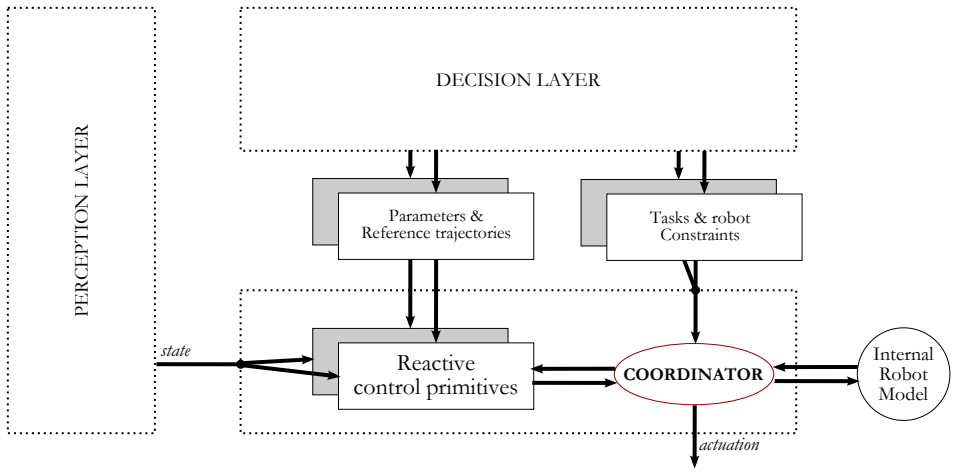

Figure 1.7.: Reactive multi-objective control architecture. Control primitives with their own parameters and reference trajectories are coordinated at the instantaneous level with respect to constraints to produce admissible whole-body outputs.

This two-layered control architecture can be seen as the combination of predictive and reactive machines, similarly to the human sensorimotor and ideomotor integration. As suggested in [Paillard1994] and [Hommel2009], a predictive machine employs motor models of the executed activities to represent actions in terms of their perceived effects, to supervise and select action plans processed by a reac-

tive machine.

In humanoid robotics, coordination is generally achieved through hierarchy within the set of primitives, in strict [Siciliano1991, Sentis2007, Mansard2009] or weighted [Shen2007, Salini2010] formulations or both, and reference parameters and trajectories are the outputs of a planning process, or more generally of a decision layer.

The emergence of MPC expressions of task primitives, and their contribution to task performance and safety, led to their consideration within such multi-objective architectures. However, employing reduced models for computational reasons, their application is generally reduced to the previewed adjustment of the reference trajectories and parameters of instantaneous — *reactive* — primitives. Indeed, the enforcement of constraints and the whole-body coordination of objectives is still required at the instantaneous level with a refined model of the controlled system. This approach results in architectures such as illustrated in figure 1.8: reactive primitives can be preceded by single-objective, task-dependent MPC problems which outputs optimal reference trajectories and parameters with respect to task-dependent reduced models.

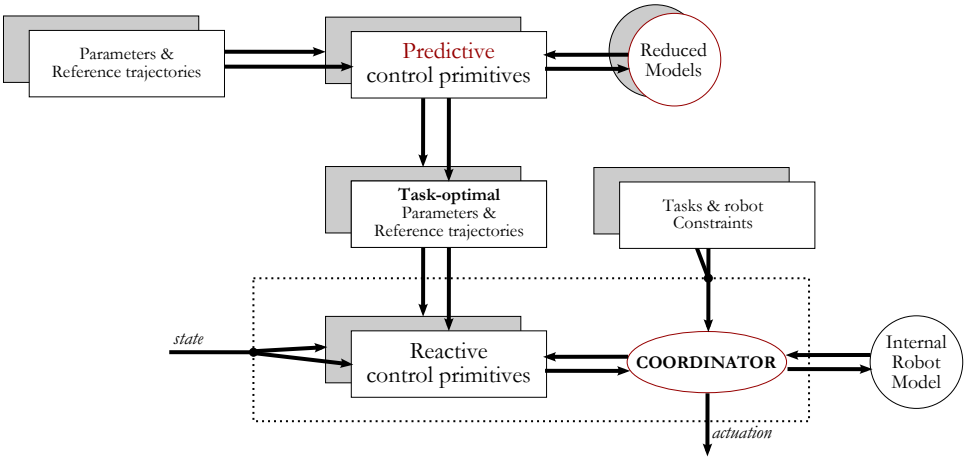

Figure 1.8.: Decentralized multi-objective predictive control architecture. Task-dependent MPC problems are employed to anticipate and adjust the reference trajectories and parameters of reactive primitives.

Such architectures can be seen as *decentralized* at the predictive level: MPC problems are treated independently, and coordination solely occurs at the instantaneous level. The multi-objective characteristic of the control architecture is therefore limited, and the advantages of predictive formulations are partially exploited.

A straightforward solution would consist in writing the entire control problem as a single, *centralized* and whole-body MPC problem. However, the dimension and form of the resulting optimization problem severely forbids real-time applications: whole-body models and constraints are indeed rarely of a computationally-efficient form, and admissible domains are commonly non-convex; state-of-the-art solvers are currently unable to efficiently handle such problems for systems with

high degrees of freedom. This approach can however demonstrate the ability to generate complex behaviors, for example at the cost of real-time requirements and with relaxed dynamical constraints in [Tassa2012] or by limiting the MPC problem to kinematics [Geoffroy2014].

The multi-objective predictive control problem can nevertheless be addressed with a different approach. As illustrated in figure 1.9, multiple objectives of the centralized MPC problem can be distributed into smaller sub-problems with their own models and objectives.

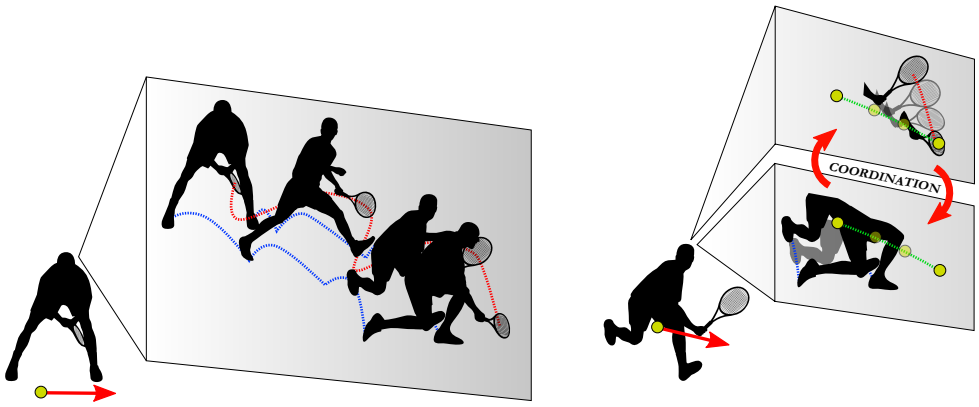

Figure 1.9.: Illustration of a distributed approach to multi-objective predictive control. The global predictive problem can be decomposed into sub-problems, with a coordination requirement to consider inter-dependencies and global objectives and constraints.

These sub-problems remain however parts of a global activity, and share a single system: dependencies and couplings thus exist. Coordination of the sub-problems is therefore a critical requirement to acknowledge these connections, consider constraints globally and drive the overall solution towards a global optimum as the desired activity demands.

1.4.2. Objective of this work

The objective of this work is henceforth to address the multi-objective predictive control problem of complex activities, for feasible applications to humanoid systems.
To enable the multi-objective consideration at the predictive level while maintaining computational feasibility, the various predictive control primitives are coordinated with their objectives, reduced models and constraints to conjointly optimize a set of parameters and reference trajectories instead of individually, with a distributed, two-layered predictive control architecture as illustrated in figure 1.10. Coordination strategies are therefore anticipated with respect to reduced models and constraints, and are refined at the instantaneous level using reactive primitives and a more complete model of the system.

To this purpose, multi-objective activities composed of two of the most occurring tasks a humanoid is expected to perform, manipulation and walking,

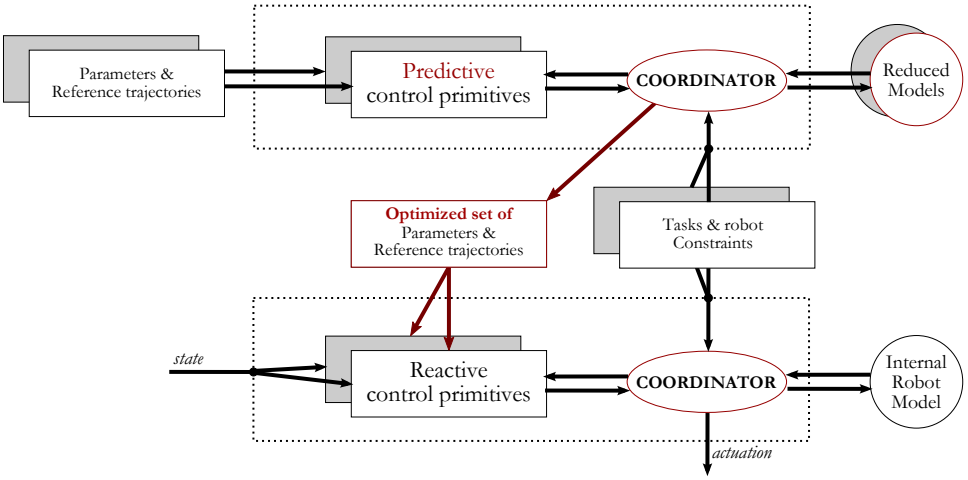

Figure 1.10.: Two-layered, multi-objective predictive control architecture. Predictive primitives are coordinated with respect to reduced models and constraints, their outputs being coordinated at the instantaneous level with a refined model of the system.

are considered in disturbed scenarii. Various coordination methods are introduced at the predictive level in this work to propose distinct approaches to multi-objective predictive control architecture. First, the structure of the task-posture multi-objective MPC is exploited to write an approximation of the coordination problem as a sequential distributed problem. An approach to decomposing a multi-objective problem into smaller, local sub-problems is then proposed to write predictive coordination as the distributed cooperation of conflicting coupled sub-processes. Last, the non-convex MPC problem addressing the coordination of discrete and continuous effects composing the hybrid nature of biped walking is written in a computationally-efficient form using mixed-integer models.
These formulations are applied to real-life scenarii considering external disturbances and applications, in both simulation and reality, to physical systems such as the iCub and TORO robots and virtual human models.

The major contribution of this work is therefore the consideration of a **two-layered, multi-objective predictive control architecture**, to enable the emergence of complex motor coordination strategies while maintaining computational feasibility.

1.4.3. Outline

This contribution is presented and validated through the introduction of approaches to the elaboration of such coordinated predictive control architectures, applied to various multi-objective activities, humanoid systems, and scenarii.

Chapter 2
Control techniques for humanoid systems

In order to introduce the technical framework behind the regulation of humanoid behaviors, the second chapter of this document first highlights the characteristics

of humanoid dynamics to raise challenges motor control must take on. The high degree of freedom of humanoid systems presents both advantages and drawbacks, and controlling their motion fundamentally requires the control of the interaction forces between the robot and its environment. Motor control consists in exploiting these multiple inputs to the system in order to execute a desired activity. This activity generally comprises of several concurrent objectives to reach, which the control architecture must coordinate in order to ensure the best execution performance.
Various control architectures are thus suggested to handle the issue of simultaneous task execution.

A selection of classical feedback formulations of regulators defining behaviors which allow to reach these elementary objectives is introduced, and is apposed to predictive control approaches which exploit a future preview horizon to optimize a given cost function capturing the goals to achieve. The distributed predictive control framework is then proposed to exploit the structure of the problem and subdivide it into sub-problems of smaller computational cost, while preserving the characteristic of previewed coordination between multiple objectives and constraints.

Chapter 3
Structure-driven decomposition: task-posture coordination

This framework is applied in a third chapter to the task-posture coordination problem of a humanoid robot. This multi-objective problem is formulated in a predictive manner to exploit a future horizon to compute coordination strategies between conflicting balance and manipulation tasks under known disturbances.

Single-objective predictive control formulations are first introduced for both tasks, and are coupled into a multi-objective distributed problem which mirrors the structure of both the robot and its activity. Simulation results on the iCub robot illustrate that anticipated coordination strategies allow to effectively compromise between objectives and thus solve conflicts, to finally provide an overall performance gain to the activity.

These results furthermore provide insight on the influence of priorities in multi-objective problems, enabling a refined tuning of the performance of the controller, while preserving the ability to temporarily relax objectives of higher priority in favor of the overall performance of the activity.

Chapter 4
An implicitly distributed problem: multi-objective balance control

In the case of more complex multi-objective control problems, no apparent structure seems exploitable. In a fourth chapter, a dual decomposition technique is applied to a multi-objective predictive formulation of postural balance control, aiming at minimizing both tip-over and slippage risks.

This approach allows to strictly decompose the multi-objective MPC problem into coupled and conflicting sub-problems. The coordination problem is translated into a non-cooperative game problem, providing the control architecture

with a gain in modularity which allows to naturally enforce approximations on the couplings between sub-systems.

A parallel algorithm is proposed to solve the resulting distributed control problem as a set of sub-problems of various time scales and approximation levels, able to successfully solve conflicts between objectives to achieve robust postural balance against unknown external disturbances. Conflicts are solved through the anticipated coordination of the robot linear and angular dynamics, which is evaluated in simulation on the iCub robot.

Chapter 5
Emergence of complex behaviors from hybrid models and MPC

The coordination problem is drawn to a higher level in a fifth chapter, where processes of different natures are coordinated in an anticipative manner to handle the hybrid characteristic of biped walking and its conflicting objectives.

Coordination strategies between discrete contact shifts and continuous postural adjustments are captured in a single predictive optimization problem aiming at reaching a desired walking objective while ensuring postural stability. The use of mixed-integer variables allows to describe discrete actions in the model and to handle non-convex admissible domains.

Simulation results on the iCub robot demonstrate that the specification of walking activities as the concurrency between displacement and balance, along with the anticipated adjustment of the gait patterns and system dynamics allows a significant gain in robustness against external disturbances. Complex walking behaviors are emerging from concurrent effect of the high-level activity specification, external disturbances and the system state and are demonstrated in simulation. Not only the humanoid is able to track a variable target by automatically adjusting its gait and recover from unknown external disturbances, it also anticipates the effects of the hybrid control policy on both objectives and accordingly compromises between them: the walking objective can be relaxed with respect to balance constraints in critical phases to eventually recover a nominal behavior.

The mixed-integer MPC formulation thus allows to draw discrete actions into the motor control layer, which are commonly supported by the decision layer, and thus enables a more consistent coordination of discrete events with continuous adjustments.

Chapter 6
Applications of the multi-objective MIQP controller

The diversity and robustness of behaviors resulting from this controller are exploited in the last chapter in applications and envisioned use-cases.

Preliminary experimental results are proposed on the TORO robot which highlight the benefits from such formulations and expose challenges raised by the implementation of computationally-demanding predictive control techniques.

Simulation-based applications are then considered with the control of virtual human models. A three-layered control architecture is considered, with a decision

layer organizing multi-objective predictive and reactive control levels. Robotics-based techniques allow to filter and enforce models on captured motion data in a simulation framework to extract kinematically and dynamically-consistent metrics from the human motion. These metrics are coupled to classification techniques to allow the recognition of the captured human activity.

Applications of multi-objective control techniques for humanoid systems are therefore feasible under real-time requirements, and simulation-based approaches with the use of virtual humans provide tools to contribute to the perception layer.

1.4.4. Publications

A selection of novel developments introduced in this work has been peer-reviewed and validated with the publication of the contributions listed hereafter:

Salini, Joseph, Vincent Padois, Aurelien Ibanez, Philippe Bidaud, and Axel Buendia. **A Goal driven perspective to generate humanoid motion synthesis**. In Field Robotics–Proceedings of the 14th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, pp. 889-897. World Scientific, 2011.

Ibanez, Aurelien, Philippe Bidaud, and Vincent Padois. **Unified preview control for humanoid postural stability and upper-limb interaction adaptation**. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, pp. 1801-1808. IEEE, 2012.

Ibanez, Aurelien, Philippe Bidaud, and Vincent Padois. **Previewed impedance adaptation to coordinate upper-limb trajectory tracking and postural balance in disturbed conditions**. In Nature-Inspired Mobile Robotics–Proceedings of the 16th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, pp. 519-528. World Scientific, 2013.

Ibanez, Aurelien, Philippe Bidaud, and Vincent Padois. **A distributed model predictive control approach for robust postural stability of a humanoid robot**. In Robotics and Automation (ICRA), 2014 IEEE International Conference on, pp. 202-209. IEEE, 2014.

Ibanez, Aurelien, Philippe Bidaud, and Vincent Padois. **Automatic optimal biped walking as a mixed-integer quadratic program**. In Advances in Robot Kinematics, pp. 505-516. Springer International Publishing, 2014.

Ibanez, Aurelien, Philippe Bidaud, and Vincent Padois. **Emergence of humanoid walking behaviors from mixed-integer model predictive control**. In Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on, pp. 4014-4021. IEEE, 2014.

Granata, Consuelo, Aurelien Ibanez, and Philippe Bidaud. **Human activity understanding: a multilayer approach combining body movements and contextual descriptors analysis**. In International Journal of Advanced Robotic Systems, 2015.

CHAPTER 2

Control techniques for humanoid systems

Contents

This chapter introduces the formalism supporting the works developped in this document. Humanoid systems — *virtual* or *physical* — are regarded as polyarticulated mechanical systems with actuation. The challenges that motion control of highly-articulated systems in interaction with the environment must take on are inferred from the equations of motion describing the dynamics of such systems.

The interaction of a humanoid system with its surroundings is inevitable: not only can the motion of such systems only be achieved through the reaction of the environment as contact actions, they are by design expected to interact with the environment. Indeed, in general and functional terms, actuated systems are expected to *act* on their environment to alter it in order to achieve a desired *objective*, this alteration ranging from the simple movement initiation (*e.g.* manipulation tasks) to deeper transformations (*e.g.* manufacture processes). These interactions must thus be *controlled* with the aim to ensure not only the regulation of the system motion, but also the accomplishment of the expected objectives with the best possible performance.
In the case of humanoid robots however, the scope of interactions is much wider: beyond the sheer *action-reaction* couple with the environment, robots are expected to *interact* with humans. At the physical level, although a variety of established models have been developed to describe potential reactions of diverse types of environments (*e.g.* contact models), human reaction is a far more complex reality due to its individual characteristics. This complexity extends to the *cognitive* aspect of HRI, where humans might not only react to the system behavior before physical contact but will also anticipate its future actions.

As motion control is still an open problem in humanoid robotics due to the high number of variables and constraints describing their dynamics, their complex spatial structure and the several time and space scales they evolve in (with diverse objectives, motion, constraints and disturbances), interaction control is a far greater challenge. Interaction control indeed raises the issue of the definition of objectives, constraints and performance it is governed by and furthermore brings the problem of the interaction model. Achieving *robust* behavior is regardless a step towards a safer and more efficient humanoid evolution in interaction with its environment.

Activities performed by humanoid systems may require the simultaneous execution of conflicting tasks in the case of heavy object manipulation for example, where the operator is expected to guarantee a given manipulation precision while maintaining its postural balance. A task is here defined as a task function similarly to [Espiau1990, Samson1991], where reaching a null value is equivalent to reaching the associated desired objective. Multi-task control consists in the generation of motion which minimizes a set of task functions aiming at executing concurrent elementary operations which lead to the global activity objective. Hierarchical strategies are mainly found in the robotics literature, defining a relative *priority* between these multiple tasks to solve potential *conflicts* between elementary operations. Such hierarchy can be of different nature: cascaded projections in the kernel of tasks of higher priorities [Siciliano1991] lead to strict prioritization, or a consensus can be defined as a barycenter by relative weighting of task functions [Shen2007] in a multi-objective minimization problem. Each of these

multi-task control approaches have specific advantages and drawbacks, and hybrid formulations [Kanoun2009, Escande2014, Liu2014] can be found to remedy part of their shortfalls, in terms of completion results for higher priority tasks or constraints consideration for example. Performance of all of these methods still are largely determined by minimization definition and strategies for each of the considered elementary tasks nonetheless.

Common task realization methods are driven from automation techniques and comprises of *reactive* approaches with feedback controllers [Franklin1994]. The task function is optimized in these approaches by the local generation of trajectories which tend to draw the system towards dynamics in equilibrium around the targeted objective. Reactive strategies provide as prevalent advantage the ability to reject disturbances (from a flawed system identification or the environment) and thus ensure in these terms a certain tracking performance. The reactive approach nevertheless only allows the execution of *local* strategies — in both time and space — which, under large disturbances or in the case of complex and constrained trajectories, might generate discontinuities or large motion amplitudes (and lead to constraint violations, for example) up to dangerous or instable states. Moreover a reactive characteristic of control imposes reaction delays and confines to the *a posteriori* generation of control strategies.

The composition of reactive approaches with predictive formulations allows to ensure tracking performance while preparing for the upcoming constraints and objectives of the activity, and thus engage it with an optimal efficiency, as illustrated by biped balance preview control methods [Wieber2006]. A significant increase in the system postural stability is observed and the explicit consideration of constraints over a future horizon makes predictive control specifically relevant to highly-constrained tasks such as biped walking.

A majority of elementary operations required to reach a given activity goal involves specific parts of the system in a prevalent manner (*e.g.* upper-body for manipulation tasks and lower-body for walking). This remark allows to consider the system and its activity as a tree of interconnected specialized subsystems. Reaching the activity objective consists in this perspective in finding appropriate hierarchy and coordination strategies within this tree in order to achieve optimal performance over the entire execution timeline. Defining predictive strategies instead of reactive, local policies is potentially beneficial to the resolution of couplings and conflicts between these subsystems while previewing the evolution of constraints on their execution. In the predictive framework, whole-body control of a humanoid system can be assimilated in these terms to the coordinated optimization of coupled, predictive problems with local and global objectives. This class of problem finds answers in the distributed predictive control framework [Venkat2006].

This chapter is naturally built around these lines. A description of the dynamics of humanoid systems in interaction with the environment is first introduced to suggest challenges and requirements a control architecture must take up and meet to involve such systems in complex activities. Common reactive approaches for elementary task execution are then exposed to which reply predictive con-

trol techniques. Last, the distributed predictive control formalism is described regarding optimal control of multi-variable, multi-objective systems.

2.1. Dynamics of humanoid systems

Humanoid robots or virtual manikins are generally modeled as systems composed of rigid bodies, arranged in tree structure with a base body as their root, called *floating-base*. The displacement of the robot in space is captured with respect to the position and orientation of a reference frame $\mathcal{R}_b$ attached to this body, with respect to a given reference inertial frame $\mathcal{R}_0$, called *world frame*. Being free-floating systems, the base is henceforth treated as linked with a 6-DoF virtual unactuated joint to the world, defining the pose $\boldsymbol{q}_b \in SE(3)$ of $\mathcal{R}_b$ with respect to $\mathcal{R}_0$, with $SE(3)$ the special Euclidean group, as illustrated in figure 2.1; the associated twist $\boldsymbol{\nu}_b$ is in $\mathbb{R}^6$.

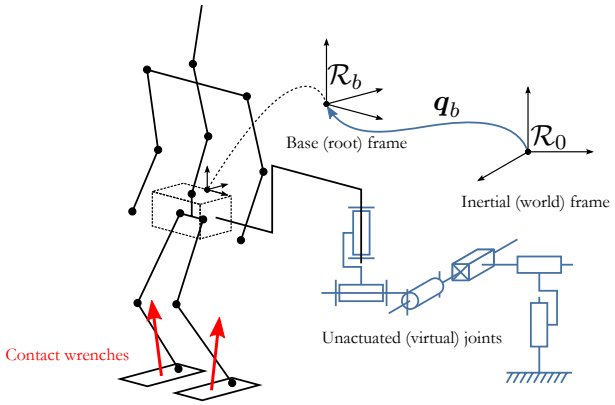

Figure 2.1.: Kinematic representation of humanoid systems. The root body of the tree structure of the mechanism is free-floating in a reference inertial frame $\mathcal{R}_0$.

The equations of motion for such systems can be derived [Murray1994] from the Lagrange formalism and take the form

$$\underbrace{\begin{bmatrix} \mathbf{M}_b & \mathbf{M}_{bj} \\ \mathbf{M}_{bj}^T & \mathbf{M}_j \end{bmatrix}}_{\mathbf{M}(\boldsymbol{q})} \underbrace{\begin{bmatrix} \dot{\boldsymbol{\nu}}_b \\ \ddot{\boldsymbol{q}}_j \end{bmatrix}}_{\dot{\boldsymbol{\nu}}} + \underbrace{\begin{bmatrix} \boldsymbol{n}_b \\ \boldsymbol{n}_j \end{bmatrix}}_{\boldsymbol{n}(\boldsymbol{q},\boldsymbol{\nu})} + \underbrace{\begin{bmatrix} \boldsymbol{g}_b \\ \boldsymbol{g}_j \end{bmatrix}}_{\boldsymbol{g}(\boldsymbol{q})} = \begin{bmatrix} \mathbf{0}_6 \\ \mathbf{S}\boldsymbol{\tau} \end{bmatrix} + \boldsymbol{\gamma}_c, \tag{2.1}$$

where $\boldsymbol{q}$ parameterizes the configuration of the free-floating system, called generalized coordinates. For the sake of simplicity, joints of the tree structure are assumed[1] to evolve in linear configuration spaces in this dissertation: that is, $\boldsymbol{q}_j \in \mathbb{R}^n$ parameterizes the joint configurations in the joint space $\mathbb{R}^n$, with n the degree of freedom of the tree structure, and $\boldsymbol{q} \in SE(3) \times \mathbb{R}^n$. The system velocity is subsequently represented with $\boldsymbol{\nu} \in \mathbb{R}^{n+6}$, concatenating the floating-base twist $\boldsymbol{\nu}_b$ and the generalized joint velocities $\dot{\boldsymbol{q}}_j \in \mathbb{R}^n$. $\mathbf{M}$ is the kinetic energy — or mass — matrix of the system and $\boldsymbol{n}$ and $\boldsymbol{g}$ are respectively the vectors of non-linear effects (Coriolis and centrifugal) and of gravity, in the generalized coordinates space. Indices $\bullet_b$, $\bullet_j$ and $\bullet_{bj}$ denote their definition with

[1] humanoids being generally articulated around revolute joints, this assumption holds

respect to the base, the joints and both, respectively. $\mathbf{S} \in \mathbb{R}^{n \times n_a}$ is a matrix representing the actuation characteristics, with $n_a \leq n$ the number of actuated degrees of freedom, and $\boldsymbol{\tau} \in \mathbb{R}^{n_a}$ is the actuation vector in the generalized coordinates relative to the joints. Vector $\boldsymbol{\gamma}_c$ captures wrenches from contacts with the environment in the generalized coordinates space.

Equations (2.1) clearly exhibits that, the floating-base being unactuated, humanoid systems are *underactuated*. It indeed appears that contact wrenches $\boldsymbol{\gamma}_c$ play a prevalent role in the dynamics of the floating-base, that is the 6 first lines of equations (2.1).

It can be noted that differential equations (2.1) are generally constrained. Indeed, regarding humanoid systems inputs $\boldsymbol{\tau}$, joint configurations $\boldsymbol{q}_j$ and their temporal derivatives are generally bounded by technological limits. Non-sliding requirements of contacts might additionally impose constraints on the contact wrenches $\boldsymbol{\gamma}_c$ and on the system configuration and velocity and acceleration $(\boldsymbol{q}, \boldsymbol{\nu}, \dot{\boldsymbol{\nu}})$.

Humanoid robots are generally brought into motion through a large number n_a of actuators.The ASIMO robot [Sakagami2002] is for example actuated through 57 joints in its version of 2011 [Asimo2011]. This great actuation degree induces that distinct system configurations allow it to reach a given pose with, for example, one of its hands. This characteristic, for a given task, is called *redundancy* and brings a set of advantages and drawbacks, among which in particular the ability to simultaneously execute several tasks with the algebraic issues caused by the infinity of input joint solutions for a given configuration of the system.

*

The dynamics of humanoid systems thus present specific properties[2] that control must handle and take advantage of. Their underactuation imposes the consideration and control of contact wrenches while their redundancy offers the opportunity to execute complex activities through the simultaneous execution of elementary operations. The control problem for such systems generally consists in that sense to solve, under constraints issued from technological limits, physics and the activity itself, the inputs $\boldsymbol{\tau}$ to be applied to the actuators and the interactions $\boldsymbol{\gamma}_c$ with the environment, with the objective to reach a desired goal.

2.2. Control of multi-objective activities

Humanoid systems present multiple inputs, their actuators, and their redundancy characteristic brings the ability to execute simultaneously several actions. Humanoids are in these terms considered as systems with multiple inputs and outputs (MIMO) that control must handle and generate, respectively. While a diversity of control architectures for MIMO systems can be found in the literature, robotic approaches mainly rely on output hierarchization methods allowing

[2] the discussion on humanoid system dynamics introduced in this section is an outset solely and will be further examined throughout the rest of this document.

to define *a priori* the relative prevalence of operations to execute in order to reach an expected behavior.

2.2.1. Multiple inputs/outputs control architectures

Control of MIMO systems commonly requires the setup of a control architecture adequate to the problem addressed, essentially for performance (optimality), setup costs (control cycle duration) and maintenance (modularity) reasons.

Centralized architectures are the most intuitive and present the best optimality results among the different control architectures for large systems. Such architectures consists in employing a model of the complete process, including interconnections between subsystems it is composed of. The monolithic structure inherent to this approach holds various drawbacks (computational cost, model identification, versatility, extraction of the system global state) which brought the paradigm of *decentralized* control to be rapidly adopted and subject to various developments [Siljak1996].

Decentralized control emphasizes on the decomposition of the system in autonomous yet interconnected agents[3]. Figure 2.2 illustrates a simplified decentralized architecture: agents $S1$ and $S2$ are autonomously regulated by their respective controllers $C1$ and $C2$ and interaction between these two agents is embodied by mutual effects between their states x_1 and x_2. In the case of strong interactions — they can be considered as disturbances to reject otherwise — ensuring global performance, or even stability, of a decentralized architecture might require the setup of *ad hoc* stabilization methods [Wang1973].

Distributed approaches propose to alleviate the drawbacks from decentralized architecture by establishing protocols of information transmission between the several controllers (*cf.* figure 2.2). Such information can be of different nature, from a state estimation of other subsystems to a direct forking of control inputs. Computational costs of this type of architectures can be reduced by establishing a certain topology of the information network: as an example an entirely connected architecture (information is shared among the whole set of controllers) can be reduced to a partially connected network in which neighbor subsystems (subjects to the largest interaction) solely exchange information. Distributed architectures are further developed in the case of predictive control at the end of this chapter.

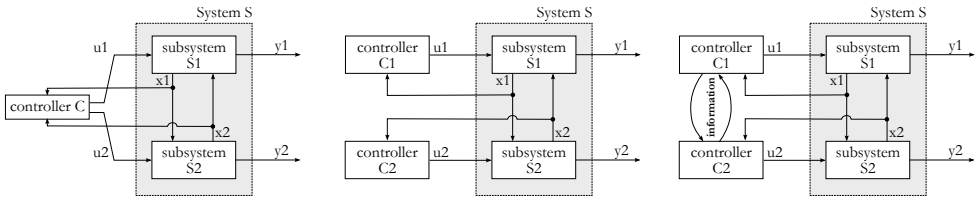

Figure 2.2.: Different control architecture for MIMO systems. *From left to right* — centralized, decentralized and distributed architectures.

[3] an agent is defined in this document as a system with an objective function to minimize and a set of control input variables, therefore defining an optimization problem

An alternative to distributed control consists in an architecture with two levels as illustrated in figure 2.3, where a top level coordinates the action of local controllers of the lower level. The coordination algorithm aims at finding a *consensus* between the different elementary actions, with respect to a global cost functions and different constraints restraining the system and its sub-agents. In the case of conflicting operations a hierarchy is generally setup [Scattolini2009] between the concurrent actions in order to achieve the *utmost* performance in reaching the global activity goal.

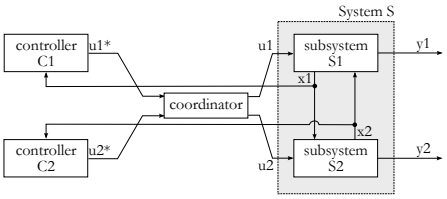

Figure 2.3.: Example of hierarchical control architecture for a two inputs/outputs system.

The framework of whole-body control of humanoid systems is relevant to the scope of hierarchical architectures. The engaged activity indeed involves subsystems with local objectives (the upper and lower parts of the system for example) performing concurrent elementary operations (manipulation and postural balance). A hierarchical approach therefore allows to define at a higher level a relative priority between the various local objective to be reached, priority which *a priori* authorizes the execution *at best* of the expected global activity.

2.2.2. Hierarchical control of humanoids

The robotics literature presents two main classes of hierarchical architecture for the control of humanoid robots, distinct by the high level coordination algorithm they employ. The first approach is a strict hierarchization which relies on a cascaded projection in the kernel of higher priorities operations [Siciliano1991]. The second [Shen2007] adopts a coordination strategy by relative weighting, hence defining through optimization costs a consensus between conflicting objectives among a global cost.

These methods find applications in humanoid control for the simultaneous execution of several dynamical tasks, for example respectively in the works of Sentis [Sentis2007] and Salini *et al.* [Salini2010].
Regarding implementation on a physical system, the generated control input must meet continuity requirements which is challenging during transitions between distinct priority sets. In the case of strict projections such transitions can be obtained [Mansard2007] by writing the control policy of each elementary operation in the form of an ordinary differential equation (ODE) defining the expected decrease of the error to the objective. The framework of weighted hierarchy allows to perform continuous transitions in a more intuitive manner, imposing for example a smooth transition of the weights during a priority shift [Salini2011].
A second critical requirement in humanoid control is the respect of constraints.

These constraints can, as an illustration, limit the actuators capacity and restrain contact wrenches to a given domain (adherent state, for example). While equality constraints can be considered algebraically in projection methods, to guarantee for example the static state of contact points with the ground as in [Sentis2006], inequality constraints can in this framework only be approximated by projection *a posteriori* of solutions or by definition of tasks aiming at repelling solutions from the boundaries of the admissible domain, with methods such as potentials [Khatib1987]. The optimization problem formed by weighting methods allows, through numerical methods, to generate solutions with a constrained domain. Inequality constraints can indeed be explicitly considered in a constrained optimization problem, as demonstrated in the case of frictional contacts [Abe2007] or extended more generally to every linear constraint [Salini2010] for actuation limits for example. More generally, optimization-based hierarchies similarly benefit from this ability [Saab2013, Escande2014].

It should be noted that not only constraints are generally of various nature and time-dependent, they are also potentially subject to uncertainties. As an example, constraints enforcing specific contact conditions with the ground are defined with respect to parameters such as friction coefficients or contact surfaces which are challenging to identify. As a consequence, strict enforcement of constraints do not necessarily guarantee the admissibility of solutions. Without a further uncertainty analysis of the control problem, margins can nevertheless be defined to cover uncertainties or to consider extreme scenarii, and constraints relaxed as objectives can be appended to repel solutions from the admissibility bounds.

2.2.2.1. Strict hierarchy through cascaded projection

This method exploits the redundancy property of humanoid robots in an algebraic manner. A synthesis of the associated formalism is introduced in this paragraph, as described by Sentis in [Sentis2006].

The redundancy property can be expressed through the properties of the Jacobian matrices of the tree-structure system. The projection $\dot{\boldsymbol{x}}$ in the world frame $\mathcal{R}_0$ (*operational* space) of a generalized velocity $\boldsymbol{\nu}$ can indeed be achieved through the Jacobian matrices $\mathbf{J}_b$ and $\mathbf{J}$ of the considered point.

$$\dot{\boldsymbol{x}} = \mathbf{J}_b\left(\boldsymbol{q}\right)\boldsymbol{\nu}_b + \mathbf{J}\left(\boldsymbol{q}\right)\dot{\boldsymbol{q}}_j,$$

where

$$\mathbf{J}_b \triangleq \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\nu}_b} \quad \text{and} \quad \mathbf{J} \triangleq \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{q}_j}.$$

The notation of the dependency of $\mathbf{J}$ to the generalized coordinates $\boldsymbol{q}$ is dropped in the rest of this document for the sake of simplicity, *i.e.* $\mathbf{J} \triangleq \mathbf{J}\left(\boldsymbol{q}\right)$. The twist $\dot{\boldsymbol{x}}$ can be described by 6 independent parameters, that is $\dot{\boldsymbol{x}} \in \mathbb{R}^6$, and $\dot{\boldsymbol{q}}_j$ by n generalized velocities[4]. The matrix $\mathbf{J}$ thus stands for a linear map from a n-dimensional vector space to a vector space of dimension 6. The rank theorem hence proves that its kernel is of non-null dimension if $n > 6$, which is generally the case in humanoid robotics. Consequently, if $\mathbf{P}$ denotes the projector in the

[4] *cf.* Sec. 2.1

kernel of $\mathbf{J}$, $\mathbf{P}$ and $\mathbf{J}$ hold

$$\dot{\boldsymbol{x}} = \mathbf{J}_b\boldsymbol{\nu}_b + \mathbf{J}\dot{\boldsymbol{q}} = \mathbf{J}_b\boldsymbol{\nu}_b + \mathbf{J}\dot{\boldsymbol{q}}_j + \mathbf{0} = \mathbf{J}_b\boldsymbol{\nu}_b + \mathbf{J}\left(\dot{\boldsymbol{q}}_j + \mathbf{P}\dot{\boldsymbol{q}}_j^*\right) \quad \forall \dot{\boldsymbol{q}}_j^* \in \mathbb{R}^n,$$

where $\dot{\boldsymbol{q}}_j^*$ is any vector of the generalized velocities space. It hence exists an infinity of articular velocities $\dot{\boldsymbol{q}}_j + \mathbf{P}\dot{\boldsymbol{q}}_j^*$ producing the twist $\dot{\boldsymbol{x}}$ of operational velocity for a given point of the robot and a given base twist $\boldsymbol{\nu}_b$. Any vector $\dot{\boldsymbol{q}}_j^*$ projected in the kernel of $\mathbf{J}$ therefore do not interfere at the operational level with the singular solution $\dot{\boldsymbol{q}}_j$. This superabundance of generalized velocities corresponding to a given twist $\dot{\boldsymbol{x}}$ constitutes the redundancy property of humanoid systems, and allows to consider their involvement in multi-task activities employing the entirety of their degrees of freedom.

Strict hierarchy methods take advantage of this algebraic result in humanoid robotics, which guarantee the execution of tasks of higher priority by cascaded projection of the solutions of lower priority in their kernel. Constraints on the system are considered as tasks of highest priority to ensure the respect of their operational objective. Tasks related to the activity are placed at an intermediate level and residual redundancy is employed to bring the final solution towards a given reference, generally a desired posture of the humanoid. In terms of joint inputs $\boldsymbol{\tau}$ this hierarchy writes

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{constraints} + \mathbf{P}^T_{constraints}\left(\boldsymbol{\tau}_{tasks} + \mathbf{P}^T_{tasks}\boldsymbol{\tau}_{posture}\right),$$

where vectors $\boldsymbol{\tau}_{...}$ represent the actuation vectors relative to the different operation categories[5] and matrices $\mathbf{P}_{...}$ are the associated kernel projectors. Operational space control methods employ virtual efforts $\boldsymbol{F}$ drawing the system towards the various objectives (kinematic or dynamic) of the activity. In the joint space they take the form $\boldsymbol{\tau} = \mathbf{J}^T\boldsymbol{F}$. The control input hence write, defining virtual efforts $\boldsymbol{F}_{...}$ and the associated Jacobian matrices $\mathbf{J}_{...}$ for each operation category,

$$\begin{aligned}\boldsymbol{\tau} \;=\; & \mathbf{J}^T_{constraints}\boldsymbol{F}_{constraints} \\ & + \left(\mathbf{J}_{tasks}\mathbf{P}_{constraints}\right)^T\boldsymbol{F}_{tasks} \\ & + \left(\mathbf{J}_{posture}\mathbf{P}_{tasks}\mathbf{P}_{constraints}\right)^T\boldsymbol{F}_{posture},\end{aligned}$$

where virtual efforts $\boldsymbol{F}_{...}$ can be defined as feedback controllers regulating the error to the associated objective.

This method therefore proposes an algebraic form of the input solution $\boldsymbol{\tau}$. Inequality constraints such as bounds on the input torques can in this framework be verified and enforced *a posteriori* solely, thus potentially leading to sub-optimal solutions. Formulations as constrained optimization problems allow to overcome this drawback as exposed in the following paragraph.

2.2.2.2. Relative weighting hierarchy

This method considers a set of minimization objectives aggregated into a single cost function under constraints. The generated solution therefore represents a consensus between elementary objectives, parameterized by their relative priorities defined as weights.

[5] between constraints, tasks, and posture in this case

The formalism described in [Salini2010] writes objectives to reach in the form of desired twist derivatives $\dot{\boldsymbol{t}}_j^*$ or wrenches $\boldsymbol{w}_j^*$, expressed at a point of the robot (operational objective) or directly in the joint space. The associated task function T_j is defined as

$$T_j\left(\boldsymbol{q},\boldsymbol{\nu},\boldsymbol{\tau},\boldsymbol{F}^c\right) \triangleq \left\| \mathbf{E}_j\left(\boldsymbol{q}\right) \begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{F}^c \end{bmatrix} - \mathbf{f}_j\left(\boldsymbol{q},\boldsymbol{\nu}\right) \right\|,$$

where the matrix $\mathbf{E}_j$ and the vector $\mathbf{f}_j$ are defined to write T_j as the error between the *effective* twist derivative $\dot{\boldsymbol{t}}_j$ (respectively wrench $\boldsymbol{w}_j$) produced by $(\boldsymbol{\tau},\boldsymbol{F}^c)$ from state $(\boldsymbol{q},\boldsymbol{\nu})$ and the *desired* twist derivative $\dot{\boldsymbol{t}}_j^*$ (respectively wrench $\boldsymbol{w}_j^*$). $\boldsymbol{F}^c$ represents the operational wrench of contact actions which Jacobian $\mathbf{J}_c$ is such that, with the notation introduced in equation (2.1), $\boldsymbol{\gamma}_c = \mathbf{J}_c^T \boldsymbol{F}^c$.

The multi-objective control problem thereafter consists in finding, under constraints, the pair of actions $(\boldsymbol{\tau},\boldsymbol{F}^c)$ which minimizes a set of n_t tasks T_j. This set is centralized in a single cost function defined as a weighted sum of the n_t elementary objectives T_j. The control problem thus writes, for a state $(\boldsymbol{q},\boldsymbol{\nu})$ of the system

$$\min_{\boldsymbol{\tau},\boldsymbol{F}^c} \sum_{j=1}^{n_t} \omega_j T_j^2\left(\boldsymbol{q},\boldsymbol{\nu},\boldsymbol{\tau},\boldsymbol{F}^c\right)$$

$$\text{s.t.} \quad \begin{cases} \text{equation (2.1)} \\ \mathbf{G} \begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{F}^c \end{bmatrix} \preceq \mathbf{h} \text{ et } \mathbf{C} \begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{F}^c \end{bmatrix} = \mathbf{b} \end{cases}$$

where ω_j balance the relative prevalence of objectives T_j and the matrix-vector pairs $(\mathbf{G},\mathbf{h})$ and $(\mathbf{C},\mathbf{b})$ respectively write the inequality and equality constraints defining the admissible domain of variables $(\boldsymbol{\tau},\boldsymbol{F}^c)$.

This formalism allows to notably account for technological limitations on the system (bounded joint torques, positions, velocities and accelerations) and additionally the null-velocity condition of the solids in contact. A linear approximation of the admissible cones can also guarantee their non-sliding under a dry friction model of Coulomb. Nevertheless, in the case of conflicting objectives T_j, the resolution in the form of a consensus do not guarantee the proper execution of prevalent objectives, as opposed to projection methods. Hybrid methods as described in [Kanoun2009] define similar optimization problems at each priority level, solved in a recursive manner, to take advantage of the characteristics of both methods.

*

Hierarchical control structures provide tools to handle constraints and conflicts between simultaneous local desired objectives, which makes them specifically relevant to the whole-body control problem of humanoid robots.
The following sections describe reactive and predictive approaches allowing to determine, through the definition of a minimization strategy of an error to a given objective, instantaneous desired goals such as desired twists $\boldsymbol{t}_j^d$ or wrenches $\boldsymbol{w}_j^d$ for a given system state.

2.3. Reactive feedback controllers

Reactive approaches aim at enforcing an expected instantaneous behavior on the controlled system, and feedback methods define such a control policy relatively to an error between the current state of the system and an objective they aim. Some of the most widely spread techniques of the literature are exposed in this section in their canonical form. A deeper analysis and more elaborate formulations such as H_∞ control can be found in [Franklin1994] and [Skogestad2007].

2.3.1. Proportional-integral-derivative controller

Proportional-integral-derivative (PID) control is one of the most widespread feedback control policies in automation, notably for its robustness properties in the control of unidentified processes.

The control input $\boldsymbol{u}$ is defined as a weighted sum of terms capturing the current error to the objective $\mathbf{e}$, its accumulation over time and its rate of change $\dot{\mathbf{e}}$. The control policy thus takes the form

$$\boldsymbol{u}\left(t\right) \triangleq \mathbf{K}_p \mathbf{e}\left(t\right) + \mathbf{K}_i \int_0^t \mathbf{e}\left(x\right) dx + \mathbf{K}_d \dot{\mathbf{e}}\left(t\right),$$

where matrices $(\mathbf{K}_p, \mathbf{K}_i, \mathbf{K}_d)$ are the controller parameters defining the behavior of the input relatively to the error, its accumulation and rate of change.

An application to robotics can be illustrated by the computation of a desired acceleration $\dot{\boldsymbol{t}}^d$ of a point of the robot in order to reach a target position $\boldsymbol{x}^c$ with the velocity $\boldsymbol{t}^c$ and acceleration $\dot{\boldsymbol{t}}^c$ with a proportional-derivative method

$$\dot{\boldsymbol{t}}^d \quad \text{s.t.} \quad \dot{\boldsymbol{t}}^d - \dot{\boldsymbol{t}}^c = \mathbf{K}_p\left(\boldsymbol{x}^c - \boldsymbol{x}\right) + \mathbf{K}_d\left(\boldsymbol{t}^c - \boldsymbol{t}\right),$$

where $\boldsymbol{x}$ and $\boldsymbol{t}$ are respectively the current linear position and velocity of the considered frame.[6] This control policy can also be written in the form of a desired virtual action $\boldsymbol{F}_{task}$ to reach an objective $\boldsymbol{x}_c$ with null velocity

$$\boldsymbol{F}_{task} \triangleq \mathbf{K}_p\left(\boldsymbol{x}^c - \boldsymbol{x}\right) - \mathbf{K}_d \boldsymbol{t}.$$

This virtual action can be interpreted as the effect of a virtual string-damper system with unstretched free position $\boldsymbol{x}^c$.
In a multi-objective control framework, $\boldsymbol{F}_{task}$ and $\dot{\boldsymbol{t}}^d$ can be employed to define elementary objectives in a strict or weighted strategy respectively.

Such control policies however do not *explicitly* define the reaction of the system to an external action. For the execution of tasks in contact with the environment, an explicit specification of the *apparent* dynamics of the system might prove necessary to ensure an expected behavior. Impedance control methods [Hogan84] proposes for example to unify the definition of the desired acceleration and effort with respect to a known external action.

[6] A pose error $\mathbf{e}$ including orientations would require to resort to a non-minimal representation of the orientation such as quaternions.

2.3.2. Impedance control

Impedance control allows to define an expected mechanical behavior of the system in interaction with the environment. Under an external wrench $\boldsymbol{F}^E$ on the effector of the system, the impedant expected behavior can be defined as

$$\mathbf{A}\left(\boldsymbol{x}-\boldsymbol{x}^c\right)+\mathbf{B}\left(\boldsymbol{t}-\boldsymbol{t}^c\right)+\mathbf{C}\left(\dot{\boldsymbol{t}}-\dot{\boldsymbol{t}}^c\right)=\boldsymbol{F}^E$$

where matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ respectively define the desired apparent stiffness, damping and inertia of the system. The desired acceleration $\dot{\boldsymbol{t}}^d$ of the effector results in

$$\dot{\boldsymbol{t}}^d \quad \text{s.t.} \quad \dot{\boldsymbol{t}}^d-\dot{\boldsymbol{t}}^c=\mathbf{C}^{-1}\left[\mathbf{A}\left(\boldsymbol{x}^c-\boldsymbol{x}\right)+\mathbf{B}\left(\boldsymbol{t}^c-\boldsymbol{t}\right)+\boldsymbol{F}^E\right].$$

This expression is similar to the proportional-derivative policy from the previous paragraph, with as distinction the consideration of the interaction wrench $\boldsymbol{F}^E$. Impedance control thus allows to define a compromise between position tracking and amplitude of contact actions. The explicit definition of the apparent stiffness $\mathbf{A}$ of the system in the operational space furthermore allows the modulation of this compromise in the various directions considered.

PID and impedance methods nevertheless require the definition *a priori* of the feedback parameters. Optimal control literature proposes computation techniques for these parameters, considering an optimization problem as illustrated in the case of linear quadratic regulators in the following paragraph.

2.3.3. Linear quadratic regulator

Linear quadratic regulators (LQR) allows to define an optimal linear feedback gain $\mathbf{K}$ which relates the control input $\boldsymbol{u}$ to a given state $\boldsymbol{x}$ of the system

$$\boldsymbol{u}\triangleq\mathbf{K}\boldsymbol{x}.$$

The optimality of gain $\mathbf{K}$ is generally defined with respect to a cost function over an infinite future horizon. The canonical form of this cost function g is quadratic and aims at minimizing a norm of the state $\boldsymbol{x}$ conjointly to the control input $\boldsymbol{u}$ amplitude. This cost function can be written in the form

$$g\left(\boldsymbol{u},\boldsymbol{x}\right)=\int_0^\infty\left(\boldsymbol{x}^T\left(t\right)\mathbf{Q}\boldsymbol{x}\left(t\right)+\boldsymbol{u}^T\left(t\right)\mathbf{R}\boldsymbol{u}\left(t\right)\right)dt,$$

where $\mathbf{Q}$ and $\mathbf{R}$ balance the relative prevalence of state over input in the optimization problem.

The computation of a future state $\boldsymbol{x}$ in the previous definition requires a model of the controlled process. A linear time-invariant system can for example be considered, described by the first order ODE $\dot{\boldsymbol{x}}=\mathbf{A}\boldsymbol{x}+\mathbf{B}\boldsymbol{u}$. The reactive control policy minimizing the cost function g therefore writes $\boldsymbol{u}\triangleq\mathbf{K}\boldsymbol{x}$ where $\mathbf{K}$ is deduced from the resolution of a Ricatti algebraic equation.

*

Such formulations transport, by employing a model of the process over a preview horizon, the definition problem of feedback parameters to the definition of a cost function which characterizes the optimality of these parameters. The preview of the process relatively to an horizon of inputs is exploited in the predictive control framework with a view to generate, notably, an optimal control horizon restrained to a set of constraints governing the state, inputs or outputs of the system.

2.4. Predictive control approaches

This section describes predictive control methods under the formalism of Model Predictive Control (MPC, which is also referred to in the literature as Dynamical Matrix Control (DMC), Generalized Predictive Control (GPC) and Receding Horizon Control (RHC). The *predictive* term translates the preview over a future horizon of the consequences of solutions on the system state.

Predictive approaches compute at each control cycle, regarding a model of the controlled process and its current actual state, an optimal horizon of control parameters with respect to a cost function, generally evaluating a tracking error to a reference trajectory. The first element of this horizon solely is applied to the system, and the computation is actualized at the next control cycle.

With the aim to produce at each control cycle solutions minimizing an error over a receding future horizon, these methods typically employ numerical optimization algorithms under constraints to solve problems of types LP (Linear Program) or QP (Quadratic Program).

An early apparition of an MPC problem can indeed be found in an LP form in the works of Propoi [Propoi1963] on optimal control topics, as mentioned in [Quin1997]. About ten years later the Model Predictive Control term emerges (MPHC, Model Predictive Heuristic Control) with Richalet *et al.* in their works on industrial processes control [Richalet1978], optimizing a quadratic cost function with a constrained iterative heuristic algorithm on an identified model (IDCOM, Identification and Command). The works of Cutler and Ramaker later propose in [Cutler1980] to write the control problem as a least-squares minimization program (DMC), to reach in 1983 the formulation with Morshedi [Cutler1983] of a QP problem explicitly considering constraints in the optimization problem (QDMC).

Despite the differences in terminology and formulations, all these methods rely on shared fundamental principles: the control problem is solved with the optimization of the future evolution of control inputs, previewed with respect to a model of the controlled process [Bemporad1999]. It is worth noticing that the preview horizons for inputs and outputs are not necessarily identical: objectives on inputs and outputs can indeed be defined distinctly. The illustration in fig-

ure 2.4 presents in these terms the concept of predictive control. A canonical formulation of an MPC problem is described in the following paragraph.

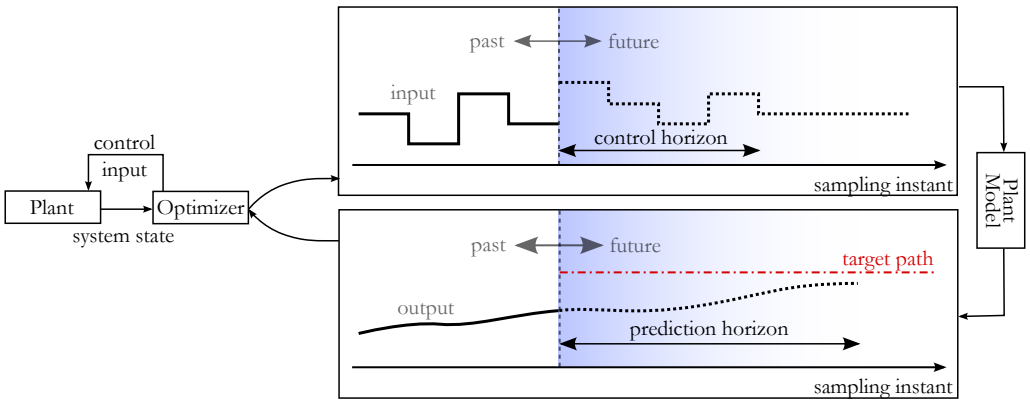

Figure 2.4.: Concept of Model Predictive Control. At each control cycle, an optimal horizon of control inputs is computed regarding the expected evolution of the system outputs with respect to a given model.

2.4.1. A Model Predictive Control problem

For the sake of example, let us consider the time-invariant system described in discrete time by the following process

$$\forall k \in \mathbb{N}^*, \forall \boldsymbol{u}_k \in \mathcal{K}_u^k, \begin{cases} \boldsymbol{x}_{k+1} &= \mathbf{f}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right), \quad \boldsymbol{x}_k \in \mathcal{K}_x^k \\ \boldsymbol{y}_k &= \boldsymbol{g}\left(\boldsymbol{x}_k\right) \end{cases}, \tag{2.2}$$

where $\boldsymbol{u}$, $\boldsymbol{x}$ and $\boldsymbol{y}$ are respectively the control inputs, state variables and outputs of the system as represented in figure 2.5. Equation (2.2) can be written in the

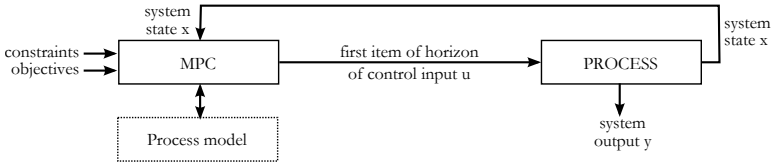

Figure 2.5.: Example of a Model Predictive Control architecture.

case of a linear process, in state-space representation,

$$\forall k \in \mathbb{N}^*, \forall \boldsymbol{u}_k \in \mathcal{K}_u^k, \begin{cases} \boldsymbol{x}_{k+1} &= \mathbf{A}\boldsymbol{x}_k + \mathbf{B}\boldsymbol{u}_k, \quad \boldsymbol{x}_k \in \mathcal{K}_x^k, \\ \boldsymbol{y}_k &= \mathbf{C}\boldsymbol{x}_k \end{cases} \tag{2.3}$$

where matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ implicitly describe the linearity of the system with respect to input variables $\boldsymbol{u}$. The domains $\mathcal{K}_u^k \subseteq \mathbb{R}^m$ and $\mathcal{K}_x^k \subseteq \mathbb{R}^n$ are bounded by constraints on the system inputs and state, related to technological limitations for example or the environment.

Without loss of generality, the control and preview horizons (*cf.* figure 2.4) are here considered as identical. If $\boldsymbol{\mathcal{U}}_{k+N|k} \triangleq \left\{\boldsymbol{u}_{k|k}, \ldots, \boldsymbol{u}_{k+N-1|k}\right\}$ denotes an horizon of input variables $\boldsymbol{u}$, the cost function g_k at time k, over an horizon of $N \in \mathbb{N}^*$

time steps and for the measured state $\boldsymbol{x}_k$, is defined as follows

$$g_k\left(\boldsymbol{x}_k, \boldsymbol{\mathcal{U}}_{k+N|k}\right) = \sum_{i=0}^{N-1} g_{i|k}\left(\boldsymbol{x}_{k+i|k}, \boldsymbol{u}_{k+i|k}\right), \tag{2.4}$$

where $\boldsymbol{x}_{k+i|k}$ and $\boldsymbol{u}_{k+i|k}$ are respectively the state and input variables of the system, previewed at time $k+i$ from the instant k. Indicators $g_{i|k}$ are from $\mathbb{R}^n \times \mathbb{R}^m$ to $\mathbb{R}$ and define a performance index at time $k+i$ of the variables previewed from time k.

The predictive control problem thus amounts to, for a given state $\boldsymbol{x}_k$ of the system, an optimal horizon of admissible inputs ${\boldsymbol{\mathcal{U}}^*}_{k+N|k}$ with respect to g_k and respecting the constraints. The problem hence writes

$$\begin{aligned} &\text{Find } {\boldsymbol{\mathcal{U}}^*}_{k+N|k} = \arg\left[\min_{\boldsymbol{\mathcal{U}}_{k+N|k}}\left[g_k\left(\boldsymbol{x}_k, \boldsymbol{\mathcal{U}}_{k+N|k}\right)\right]\right] \\ &\text{s.t.} \quad \forall i \in \{0, \ldots, N-1\} \begin{cases} \boldsymbol{x}_{k+i+1|k} = \mathbf{f}\left(\boldsymbol{x}_{k+i|k}, \boldsymbol{u}_{k+i|k}\right), \\ \boldsymbol{x}_{k+i+1|k} \in \mathcal{K}_x^{k+i+1|k}, \\ \boldsymbol{u}_{k+i|k} \in \mathcal{K}_u^{k+i|k}, \\ \boldsymbol{x}_{k|k} = \boldsymbol{x}_k. \end{cases} \end{aligned} \tag{2.5}$$

The solution ${\boldsymbol{\mathcal{U}}^*}_{k+N|k} \triangleq \left\{{\boldsymbol{u}^*}_{k|k}, \ldots, {\boldsymbol{u}^*}_{k+N-1|k}\right\}$ denotes the optimal horizon of inputs with respect to g_k, and ${\boldsymbol{u}^*}_{k|k}$ solely is applied to the system; the problem is updated and computed again at next control cycle $k+1$ with g_{k+1} as objective function.

In the absence[7] of any constraints on the inputs and state variables, *i.e.* $\forall k \in \mathbb{N}^*, \mathcal{K}_u^k = \mathbb{R}^m$ and $\mathcal{K}_x^k = \mathbb{R}^n$, a *simple* form of g_k can allow a closed-form solution of the MPC problem (2.5). However, one of the main advantages of predictive techniques is their capacity to explicitly account for a set of constraints on the system; in numerous applications the MPC problem is thus solved through numerical methods. These aspects of the MPC framework are discussed in the next paragraph.

2.4.2. A reference MPC formulation: Quadratic Programming

A widespread form of cost function in the case of linear processes is quadratic[8] with respect to each of the input $\boldsymbol{u}$ and output $\boldsymbol{y}$ variables of the system, optimized under constraints on the inputs $\boldsymbol{u}$ solely.

It can be generally written, with $\boldsymbol{\mathcal{Y}}_{k+N|k} \triangleq \left\{\boldsymbol{y}_{k+1|k}, \ldots, \boldsymbol{y}_{k+N|k}\right\}$ denoting an horizon of outputs of the linear process (*cf.* equation (2.3)), a quadratic cost function g_k

$$g_k\left(\boldsymbol{x}_k, \boldsymbol{\mathcal{U}}_{k+N|k}\right) = \boldsymbol{\mathcal{Y}}_{k+N|k}^T \mathbf{P} \boldsymbol{\mathcal{Y}}_{k+N|k} + \boldsymbol{\mathcal{U}}_{k+N|k}^T \mathbf{R} \boldsymbol{\mathcal{U}}_{k+N|k}, \tag{2.6}$$

[7] or in the case of equality constraints of favorable form.

[8] the formulation of a task function as an euclidean norm of the difference between a linear variable and a reference leads to a quadratic problem in that case.

where $\mathbf{P}$ and $\mathbf{R}$ are symmetric definite positive matrices defining for example weighted euclidean norms on the inputs and outputs of the system. A common form aims at minimizing the amplitude of outputs $\boldsymbol{y}$ and the changes in inputs $\boldsymbol{u}$, which brings

$$\mathbf{P} \triangleq \omega_{\boldsymbol{y}} \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}, \quad \mathbf{R} \triangleq \omega_{\boldsymbol{u}} \mathbf{D}^T \mathbf{D} \text{ with } \mathbf{D} \triangleq \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix},$$

where $\omega_{\boldsymbol{y}}$ and $\omega_{\boldsymbol{u}}$ are scalar homogenization coefficients weighting the relative prevalence of each of the errors on the inputs and outputs in the optimization problem.
From the discrete description of the linear process in equation (2.3), the recurrence can be concatenated over the horizon N as

$$\boldsymbol{\mathcal{Y}}_{k+N|k} = \mathbf{G}\boldsymbol{x}_k + \mathbf{H}\boldsymbol{\mathcal{U}}_{k+N|k}, \tag{2.7}$$

with

$$\mathbf{G} \triangleq \begin{bmatrix} \mathbf{CA} \\ \vdots \\ \mathbf{CA}^N \end{bmatrix} \quad \text{and} \quad \mathbf{H} \triangleq \begin{bmatrix} \mathbf{CB} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{CAB} & \mathbf{CB} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{N-1}\mathbf{B} & \mathbf{CA}^{N-2}\mathbf{B} & \dots & \mathbf{CB} \end{bmatrix}.$$

The minimization objective described in equation (2.6) explicitly rewrites, following equation (2.7), as a function of the optimization variable $\boldsymbol{\mathcal{U}}_{k+N|k}$ and of the current state $\boldsymbol{x}_k$ of the system

$$\begin{aligned} g_k\left(\boldsymbol{x}_k, \boldsymbol{\mathcal{U}}_{k+N|k}\right) &= \boldsymbol{\mathcal{U}}_{k+N|k}^T \left(\mathbf{H}^T\mathbf{P}\mathbf{H} + \mathbf{R}\right) \boldsymbol{\mathcal{U}}_{k+N|k} \\ &\quad + 2\left(\mathbf{H}^T\mathbf{P}\mathbf{G}\boldsymbol{x}_k\right)^T \boldsymbol{\mathcal{U}}_{k+N|k} + \boldsymbol{x}_k^T\mathbf{G}^T\mathbf{P}\mathbf{G}\boldsymbol{x}_k. \end{aligned} \tag{2.8}$$

2.4.2.1. Solution under linear constraints

Constraints on solutions $\boldsymbol{u}$ are frequently bounds to respect to account for limitations of the system; the constraint $\{\boldsymbol{u} \in \mathcal{K}_u\}$ thus usually take the form of a set of linear inequalities, *i.e.* $\mathcal{K}_u$ is a polyhedron. For stability reasons the polyhedron $\mathcal{K}_u$ must contain the origin. In this paragraph we assume these conditions to hold.
In matrix form, with $\mathbf{F}$ denoting a selection matrix and $\mathbf{f}$ an upper bound vector, we write

$$\left\{\forall i \in \{0, \dots, N-1\}, \boldsymbol{u}_{k+i|k} \in \mathcal{K}_u^{k+i|k}\right\} \Leftrightarrow \mathbf{F}\boldsymbol{\mathcal{U}}_{k+N|k} \preceq \mathbf{f}.$$

The predictive control problem (2.5) hence writes as a Quadratic Program in its canonical form. Indeed $\boldsymbol{x}_k^T\mathbf{G}^T\mathbf{P}\mathbf{G}\boldsymbol{x}_k$ is strictly positive and constraints of the problem are defined with respect to $\boldsymbol{u}$ solely. Therefore, writing $\mathbf{Q} \triangleq \mathbf{H}^T\mathbf{P}\mathbf{H} + \mathbf{R}$ and $\mathbf{r} \triangleq \mathbf{H}^T\mathbf{P}\mathbf{G}\boldsymbol{x}_k$, the problem (2.5) translates

$$\boldsymbol{\mathcal{U}}^*{}_{k+N|k} = \arg\left[\min_{\boldsymbol{\mathcal{U}}_{k+N|k}} \left(\tfrac{1}{2}\boldsymbol{\mathcal{U}}^T_{k+N|k}\mathbf{Q}\boldsymbol{\mathcal{U}}_{k+N|k} + \mathbf{r}^T\boldsymbol{\mathcal{U}}_{k+N|k}\right)\right] \quad \text{s.t.} \quad \mathbf{F}\boldsymbol{\mathcal{U}}_{k+N|k} \preceq \mathbf{f} \tag{2.9}$$

which can be numerically solved with standard iterative algorithms of quadratic optimization [Boyd2004].

2.4.2.2. Solution without constraints

It can be noted that the objective function g_k described in equation (2.8) is convex with respect to $\boldsymbol{\mathcal{U}}_{k+N|k}$, $\mathbf{H}^T\mathbf{P}\mathbf{H} + \mathbf{R}$ being definite positive if $\mathbf{C}\mathbf{B}$ is invertible.

Proof. $\mathbf{P}$ is assumed to be symmetric definite positive. The definition domain of the linear map associated to $\mathbf{P}$, without the null element, is denoted $\mathcal{K}_P \subseteq \mathbb{R}^{\dim \mathbf{P}}$. Thus $\forall \mathbf{X} \in \mathcal{K}_P, \mathbf{X}^T\mathbf{P}\mathbf{X} > 0$. For all $\mathbf{Y} \in \mathcal{K}_H$, we write $\mathbf{X} = \mathbf{H}\mathbf{Y} \in \mathcal{K}_P$. Indeed, $\mathbf{C}\mathbf{B}$ invertible $\Rightarrow$ $\mathbf{H}$ invertible $\Rightarrow$ $\{\mathbf{Y} \neq \mathbf{0} \Leftrightarrow \mathbf{X} \neq \mathbf{0}\}$. Hence, $\forall \mathbf{Y} \in \mathcal{K}_H, \mathbf{X} = \mathbf{H}\mathbf{Y} \in \mathcal{K}_P \Rightarrow (\mathbf{H}\mathbf{Y})^T \mathbf{P} (\mathbf{H}\mathbf{Y}) > 0$. In other terms, $\forall \mathbf{Y} \in \mathcal{K}_H, \mathbf{Y}^T \left(\mathbf{H}^T\mathbf{P}\mathbf{H}\right) \mathbf{Y} > 0$, *i.e.* $\mathbf{H}^T\mathbf{P}\mathbf{H}$ is symmetric definite positive. □

Therefore, in the case where constraints on the inputs $\boldsymbol{\mathcal{U}}_{k+N|k}$ are not active, the problem (2.9) is reduced to the cancellation of the gradient $\boldsymbol{\nabla} g_k$ of g_k with respect to $\boldsymbol{\mathcal{U}}_{k+N|k}$, and writes

$$\text{Find } \boldsymbol{\mathcal{U}}^*{}_{k+N|k} \quad \text{s.t.} \quad \boldsymbol{\nabla} g_k \left(\boldsymbol{x}_k, \boldsymbol{\mathcal{U}}^*{}_{k+N|k}\right) = \mathbf{0} \tag{2.10}$$

which, computing the gradient

$$\boldsymbol{\nabla} g_k \left(\boldsymbol{x}_k, \boldsymbol{\mathcal{U}}_{k+N|k}\right) = \mathbf{Q}\boldsymbol{\mathcal{U}}_{k+N|k} + \mathbf{r} = \left(\mathbf{H}^T\mathbf{P}\mathbf{H} + \mathbf{R}\right)\boldsymbol{\mathcal{U}}_{k+N|k} + \mathbf{H}^T\mathbf{P}\mathbf{G}\boldsymbol{x}_k,$$

brings as optimal horizon of control solutions

$$\boldsymbol{\mathcal{U}}^*{}_{k+N|k} = -\left(\mathbf{H}^T\mathbf{P}\mathbf{H} + \mathbf{R}\right)^{-1}\mathbf{H}^T\mathbf{P}\mathbf{G}\boldsymbol{x}_k. \tag{2.11}$$

It should be noted that this solution is equivalent to a linear feedback control of the form $\boldsymbol{u} = -\mathbf{K}\boldsymbol{x}$ defined in an optimal manner, similar to an LQR control over a *finite* horizon N.

This MPC formulation in a quadratic form can be met in the robotics literature, with and without constraints, in works addressing the control problem of biped walking as displayed in the next paragraph.

2.4.3. An example in humanoid robotics: ZMP Preview Control

The Zero-Moment Point Preview Control [Kajita2003] is a widespread predictive control approach to biped locomotion in humanoid robotics. Introduced by Kajita *et al.*, it relies on a linear model of the dynamics of the robot, reduced to its center of mass, and has been largely integrated into control algorithms for biped robots since. This linear model enables the expression of an MPC problem as a QP, thus allowing fast control rates.

2.4.3.1. Notions of biped balance

The term *balance* is here defined in its common sense for humans, as opposed to fall. A system state is considered in that sense as *out of balance* if it inexorably leads to a fall.
Although established predictive approaches [Kajita2003, Wieber2006, Herdt2010] propose rich and robust methods, the control problem of biped balance is still a challenge from the variety of disturbances (in terms of nature, amplitude and frequency) humanoid systems experience while being restrained to critical constraints on their dynamics. Postural balance is, although essential, not the main objective of a humanoid robot indeed. The ongoing activity thus leads to large and diverse disturbances (manipulation, collisions) that the balance system must handle, and the system motion is almost exclusively supported by contact actions with the environment, actions which not only cause additional disturbances (*e.g.* in the case of uneven terrain) but are also severely restrained by the contacts physics.

Contacts

As exhibited in the equations of motion (2.1), free-floating motion of the humanoid is directly characterized by gravity and inertial (including centrifugal) effects, and by the reaction $\boldsymbol{\gamma}_c$ from the environment to the system. This reaction is generally the result from contacts, in the form of a contact wrench which represents the equivalent effect of the potentially multiple contacts with the environment. This wrench is induced by infinitesimal contact forces distributed over the contact surfaces. Motion of the system, and therefore its stability, are thus characterized by this distribution of contact forces.

The center of pressure allows to capture part of this distribution, being the point where the moment resulting from the normal contact forces is null. The amplitude of tangential forces being generally significantly lower than of normal forces, normal effects are commonly considered as playing a prevalent role in the stability of humanoids. Nevertheless, tangential forces notably have a critical impact on the stability of contacts as they define the contact regime (friction, slippage) with respect to normal forces.
It is worth noticing that in the general case, the definition of the CoP defines an entire axis rather than a single point. However, in the case of coplanar contacts, the CoP can be taken as the intersection between the contact plane and this axis, as proposed in [Sardain2004]. This case will be assumed to hold in the entirety of this dissertation for the sake of simplicity.

In the case of coplanar contacts, the unilateral characteristic of contact forces imposes by definition that the CoP lies within the convex hull of the contact surfaces. Furthermore, if the CoP reaches a bound of this convex hull, no torque can be generated from normal contact forces in the direction of the tangent to this bound towards the interior of the convex hull: actuation is lost in this direction. As a result, not all inertial and gravity effects can be compensated at the level of the contact surface, and henceforth the bodies in contact, and the whole system itself, are potentially unbalanced torque-wise. Angular motions of the bodies in contact are a threat to postural balance, as it does not preserve contact stability

and potentially leads to tip-over motions: consequently, a widespread criterion for postural balance is to maintain the CoP as far as possible from the bounds of the contact convex hull.

The foot rotation indicator (FRI) was introduced as a criterion allowing to extend the CoP-based definition of postural stability which do not allow to distinguish between the marginal state of static equilibrium and its complete loss [Goswami1999]. The FRI indeed can leave the convex hull of contacts and thus provides an indicator of the instability induced by the angular motion of the feet. Nevertheless, if the balance control problem is treated as avoiding — and not recovering from — instabilities, CoP-based criteria are sufficient.

Zero-Moment Point

The Zero-Moment Point [Vukobratovic1972] proposes to capture the position of the CoP without the explicit knowledge of contact wrenches thus, according to the equations of motion, as a function of the motion of the system. Its definition as the point on the ground where the tipping moment[9] due to inertia and gravity effect is null makes it coincident with the CoP, being both defined with respect to opposite wrenches. Nevertheless, their definition relies on the existence of a contact surface[10] and this coincidence is therefore unfounded when contacts are distributed over distinct surfaces.

A widespread model in humanoid robotics is a linear inverted pendulum (LIP): the system being reduced to a point mass at its CoM $\boldsymbol{x}$, neglecting rotational effects and assuming a constant distance z from the CoM to the contacts plane, the position $\boldsymbol{p}$ of the CoP/ZMP in the contacts plane follows analogue dynamics

$$\boldsymbol{p} = \boldsymbol{x} - \frac{z}{g}\ddot{\boldsymbol{x}}, \tag{2.12}$$

where g is the gravity amplitude. The derivation of this expression of the ZMP position from the equations of motion can be found[11] in sections 3.1 and 4.1.

The assumption of a constant altitude of the CoM is exploited to derive a linear model from these simplified dynamics by neglecting the vertical acceleration $\ddot{z}$ and the variations of z. Additionally, it reduces the dimension of the problem by considering solely the evolution of the CoM in the horizontal plane. This hypothesis is valid for common walking and balance motions, where the vertical motion of the CoM is slower than the horizontal one. Neglecting the rotational inertial effects also lowers the dimension of the problem, and is similarly valid for common walking motions although bracing behaviors with notably the upper limbs are critical in extreme scenarii.
Both of these hypotheses are dropped in Chapter 4 to exploit the effects they neglect.

2.4.3.2. Linear model

In order to derive a QP expression of the balance MPC problem, a linear model is deduced from the simplified linear dynamics (2.12). The CoM and position $\boldsymbol{x}$

[9] *i.e.* the moment around the axis orthogonal to the ground is not considered
[10] not necessarily planar, *cf.* [Sardain2004]
[11] *cf.* p. 56 and p. 84

and velocity $\dot{\boldsymbol{x}}$ being related, the CoM jerk is defined as sole input variable to the system, *i.e.* $\boldsymbol{u} \triangleq \frac{\partial \ddot{\boldsymbol{x}}}{\partial t}$. In discrete time of sampling period δt, the evolution of the CoM dynamics $\hat{\boldsymbol{x}} = [\boldsymbol{x}\ \dot{\boldsymbol{x}}\ \ddot{\boldsymbol{x}}]^T$ can be previewed with the setup of an explicit integration scheme. With the notation $\bullet_k \triangleq \bullet\,(k\delta t)$, the linear process in discrete time of state $\hat{\boldsymbol{x}}$ and input $\boldsymbol{u}$, whose output $\boldsymbol{p}$ is deduced from equation (2.12) for a given CoM altitude z, can be written as described in [Kajita2003] as follows

$$\left\{ \begin{array}{rcl} \hat{\boldsymbol{x}}_{k+1} & = & \mathbf{A}\hat{\boldsymbol{x}}_k + \mathbf{B}\boldsymbol{u}_k \\ \boldsymbol{p}_k & = & \mathbf{C}_k\boldsymbol{x}_k \end{array} \right. \tag{2.13}$$

where

$$\mathbf{A} \triangleq \begin{bmatrix} 1 & \delta t & \delta t^2/2 \\ 0 & 1 & \delta t \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B} \triangleq \begin{bmatrix} \delta t^3/6 \\ \delta t^2/2 \\ \delta t \end{bmatrix}, \quad \mathbf{C}_k \triangleq \begin{bmatrix} 1 & 0 & -\frac{z}{g} \end{bmatrix}.$$

This linear model (2.13) allows to write, with $\boldsymbol{\mathcal{U}}_{k+N|k} \triangleq \left\{\boldsymbol{u}_{k|k}, \ldots, \boldsymbol{u}_{k+N-1|k}\right\}$ and $\boldsymbol{\mathcal{P}}_{k+N|k} \triangleq \left\{\boldsymbol{p}_{k+1|k}, \ldots, \boldsymbol{p}_{k+N|k}\right\}$, a preview horizon of the outputs $\boldsymbol{p}$ for N time steps in the linear form

$$\boldsymbol{\mathcal{P}}_{k+N|k} = \mathbf{P}_x\hat{\boldsymbol{x}}_k + \mathbf{P}_u\boldsymbol{\mathcal{U}}_{k+N|k},$$

as described[12] in [Wieber2006]. With the objective of reducing tip-over risk, control aims at minimizing the gap between the output $\boldsymbol{\mathcal{P}}_{k+N|k}$ and an horizon of *stable* references $\boldsymbol{\mathcal{P}}^r_{k+N|k}$, commonly located at the center of the sustentation polygon of the robot.

2.4.3.3. Predictive control

The predictive control problem therefore writes, $\mathcal{K}_{k+i}$ denoting the sustentation hull at step $k+i$, as the constrained minimization of an error of the outputs to the references

$$\begin{array}{c} \min\limits_{\boldsymbol{\mathcal{U}}_{k+N|k}} \ \frac{1}{2}\left(\omega_p \left\|\boldsymbol{\mathcal{P}}_{k+N|k} - \boldsymbol{\mathcal{P}}^r_{k+N|k}\right\|^2 + \omega_u \left\|\boldsymbol{\mathcal{U}}_{k+N|k}\right\|^2\right) \\ \text{s.t.} \qquad \text{s.t.} \quad \forall i \in \{1, \ldots, N\}, \quad \boldsymbol{p}_{k+i|k} \in \mathcal{K}_{k+i} \end{array}$$

where $\left\|\boldsymbol{\mathcal{U}}_{k+N|k}\right\|^2$ is a regularizing term aiming at minimizing the amplitude of input $\boldsymbol{u}$ and coefficients (ω_p, ω_u) define the prevalence of the central tracking objective relatively to the regularization term. This problem can be translated in a canonical form similar to equation (2.14) as demonstrated by Wieber in [Wieber2006], who solves the control problem with a QP algorithm. Besides, these same works show a closed-form solution in the absence of constraints, with a similar development to the one presented in Sec. 2.4.2.2.

[12] a detail of matrices $(\mathbf{P}_x, \mathbf{P}_u)$ can be found in Sec. 3.1.

*

Predictive control approaches provide tools with characteristics particularly relevant to humanoid systems. Such systems and their activities are indeed subject to numerous constraints with critical dynamics, and the ability to consider them explicitly over a preview horizon allows to define optimal and anticipative strategies rather than local policies.
However, their high degree of freedom and the absence of closed-form solution for complex problems under constraints raise the issue of computational cost. Distribution techniques are thus introduced in the next section to propose approaches and solutions to the decomposition of large problems into coupled subproblems, and therefore alleviate some of the cost brought by the implementation of predictive methods.

2.5. Distributed predictive control

Distributed control is introduced as an alternative architecture to centralized and decentralized control. On the one hand, MPC applications to industrial process of large dimensions rapidly met the limitations of centralized approaches. On the other hand, the lack of stability guarantee of decentralized methods might in some cases prejudice the advantages brought by the cost reduction they imply. Distributed approaches propose to decompose the system into autonomous subagents organized into a network of information exchange, in order to alleviate the drawbacks of centralized architecture by taking advantage of the sparsity of most large systems: interaction between sub-systems can generally be reduced to closest neighbors.
Setting up distribution techniques nevertheless requires to address the coordination problem of the several local controllers. In this perspective Model Predictive Control is particularly relevant as it provides a temporal window — the preview horizon — to exploit in order to establish a coordination strategy. Distributed predictive control (DMPC) is thus the application of MPC techniques to distributed architectures.

Two major classes of distributed architectures can be distinguished from the scope of the respective cost functions of the local controllers: a DMPC architecture is said to be *cooperative* if it employs global cost functions, and *non-cooperative* otherwise. Among each of these classes a distinction is made according to the flow of information exchange (*cf.* figure 2.6). In the case of unilateral communication between controllers, their evaluation can be performed sequentially; this approach is thus *sequential*, as opposed to *parallel* methods where controllers bilaterally exchange information.

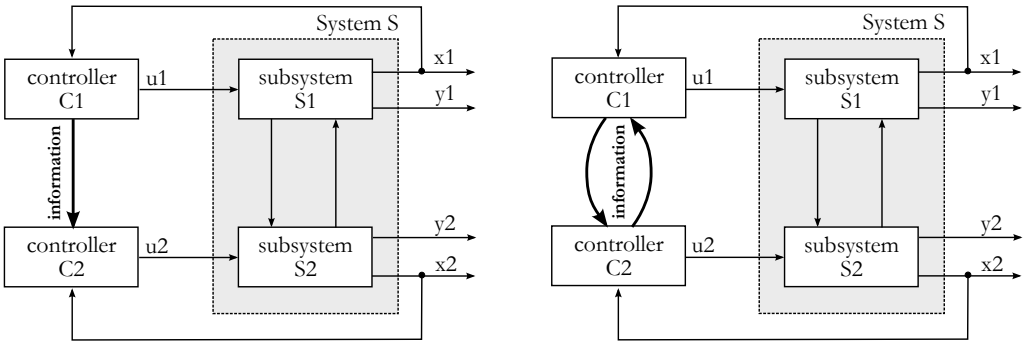

Figure 2.6.: Sequential and parallel DMPC networks. A unilateral information flow allows to sequentially execute local controllers, while bilateral communication requires a parallel resolution.

*

Performance of distributed methods relies predominantly on the coordination of these coupled sub-agents. A specific attention is thus brought to the coupling model between sub-systems in the next paragraph, and different communication strategies between controllers are later discussed to meet various control requirements.

2.5.1. Coupling models

Let us consider, for the sake of illustration, the following global system in discrete time

$$\forall k \in \mathbb{N}^*, \forall \boldsymbol{u}_k \in \mathcal{K}_u^k, \left\{ \begin{array}{rcl} \boldsymbol{x}_{k+1} & = & \mathbf{f}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right), \quad \boldsymbol{x}_k \in \mathcal{K}_x^k, \\ \boldsymbol{y}_k & = & \boldsymbol{g}\left(\boldsymbol{x}_k\right) \end{array} \right. \tag{2.14}$$

where $\boldsymbol{u}$, $\boldsymbol{x}$ and $\boldsymbol{y}$ are respectively the control, state and output variables of the system. The system is decomposed into s sub-systems whose inputs, states and outputs are concatenated into $\boldsymbol{u}$, $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively:

$$\boldsymbol{u} \triangleq \left[\boldsymbol{u}^{(1)T} \ \ldots \ \boldsymbol{u}^{(s)T}\right]^T, \quad \boldsymbol{x} \triangleq \left[\boldsymbol{x}^{(1)T} \ \ldots \ \boldsymbol{x}^{(s)T}\right]^T, \quad \boldsymbol{y} \triangleq \left[\boldsymbol{y}^{(1)T} \ \ldots \ \boldsymbol{y}^{(s)T}\right]^T.$$

In the general case the coupling between a sub-system $i \in \{1 \ldots s\}$ and the others is described by a process whose state $\boldsymbol{x}^{(i)}$ and output $\boldsymbol{y}^{(i)}$ can be dependent on the whole set of states $\boldsymbol{x}$, inputs $\boldsymbol{u}$ and outputs $\boldsymbol{y}$ of the global system:

$$\forall k \in \mathbb{N}^*, \forall \boldsymbol{u}_k \in \mathcal{K}_u^k, \left\{ \begin{array}{rcl} {\boldsymbol{x}_{k+1}}^{(i)} & = & \mathbf{f}^{(i)}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right), \quad \boldsymbol{x}_k \in \mathcal{K}_x^k, \\ {\boldsymbol{y}_k}^{(i)} & = & \boldsymbol{g}^{(i)}\left(\boldsymbol{x}_k\right) \end{array} \right. \tag{2.15}$$

to which the fully decoupled model can be opposed

$$\forall k \in \mathbb{N}^*, \forall {\boldsymbol{u}_k}^{(i)} \in \mathcal{K}_u^{k(i)}, \left\{ \begin{array}{rcl} {\boldsymbol{x}_{k+1}}^{(i)} & = & \mathbf{f}^{(i)}\left({\boldsymbol{x}_k}^{(i)}, {\boldsymbol{u}_k}^{(i)}\right), \quad {\boldsymbol{x}_k}^{(i)} \in \mathcal{K}_x^{k(i)}, \\ {\boldsymbol{y}_k}^{(i)} & = & \boldsymbol{g}^{(i)}\left({\boldsymbol{x}_k}^{(i)}\right) \end{array} \right. \tag{2.16}$$

which amounts to decomposing the process described in equation (2.14) into s independent sub-processes.

Several sub-classes of coupling can be deduced from these two cases, the most common in the literature being input couplings:

input couplings

$$\forall k \in \mathbb{N}^*, \forall \boldsymbol{u}_k \in \mathcal{K}_u^k, \left\{ \begin{array}{rcl} {\boldsymbol{x}_{k+1}}^{(i)} & = & \mathbf{f}^{(i)}\left({\boldsymbol{x}_k}^{(i)}, \boldsymbol{u}_k\right), \quad {\boldsymbol{x}_k}^{(i)} \in \mathcal{K}_x^{k(i)} \\ {\boldsymbol{y}_k}^{(i)} & = & \boldsymbol{g}^{(i)}\left({\boldsymbol{x}_k}^{(i)}\right), \end{array} \right. \tag{2.17}$$

state couplings

$$\forall k \in \mathbb{N}^*, \forall {\boldsymbol{u}_k}^{(i)} \in \mathcal{K}_u^{k(i)}, \left\{ \begin{array}{rcl} {\boldsymbol{x}_{k+1}}^{(i)} & = & \mathbf{f}^{(i)}\left(\boldsymbol{x}_k, {\boldsymbol{u}_k}^{(i)}\right), \quad \boldsymbol{x}_k \in \mathcal{K}_x^k, \\ {\boldsymbol{y}_k}^{(i)} & = & \boldsymbol{g}^{(i)}\left({\boldsymbol{x}_k}^{(i)}\right), \end{array} \right. \tag{2.18}$$

input and state couplings

$$\forall k \in \mathbb{N}^*, \forall \boldsymbol{u}_k \in \mathcal{K}_u^k, \left\{ \begin{array}{rcl} {\boldsymbol{x}_{k+1}}^{(i)} & = & \mathbf{f}^{(i)}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right), \quad \boldsymbol{x}_k \in \mathcal{K}_x^k, \\ {\boldsymbol{y}_k}^{(i)} & = & \boldsymbol{g}^{(i)}\left({\boldsymbol{x}_k}^{(i)}\right). \end{array} \right. \tag{2.19}$$

When distributing a centralized model, *i.e.* decomposing a system into coupled sub-systems, designing the couplings amounts to defining the degree of reduction of the original problem. Furthermore, couplings implicitly define the topology of the network of sub-agents, and thus largely influence the choice of coordination techniques between the respective controllers of these sub-systems.

2.5.2. Decomposition and coordination

Distributed architecture prevalently rely on a decomposition of the system into sub-systems of lower dimension, and single out from decentralized approaches by the coordination of sub-agents. Table 2.1 presents a nomenclature of optimal control architectures in the case of two coupled sub-systems.

This coordination aims at bringing the distributed optimization toward a *collective* optimum (in the sense of Pareto) in the case of *cooperative* algorithms, and generally requires the setup of *iterative* methods. Conversely, *non-cooperative* algorithms imply the respective optimization from each sub-agent of a *local* cost function, whose evolution is also subject to — known — actions from the other sub-systems. As a result actions are taken *individually*, each controller aiming at accepting a change in a variable under the sole condition of a local benefit, and thus tends to produce solutions drawing the system towards a Nash equilibrium [Nash1951]. This equilibrium is a stable state around which one — or all — system(s) would undergo a handicap with respect to its respective local objective.

Input variables	$\boldsymbol{u}^{(1)} \in {\mathcal{K}_u}^{(1)}$	$\boldsymbol{u}^{(2)} \in {\mathcal{K}_u}^{(2)}$
Cost functions	$\tilde{g}^{(1)}\left(\boldsymbol{u}^{(1)}\right), g^{(1)}\left(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}\right), \tilde{g}^{(2)}\left(\boldsymbol{u}^{(2)}\right), g^{(2)}\left(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}\right)$ $g\left(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}\right) \triangleq \omega_1 g^{(1)}\left(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}\right) + \omega_2 g^{(2)}\left(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}\right)$	
Decentralized	$\min_{\boldsymbol{u}^{(1)} \in {\mathcal{K}_u}^{(1)}} \tilde{g}^{(1)}\left(\boldsymbol{u}^{(1)}\right)$	$\min_{\boldsymbol{u}^{(2)} \in {\mathcal{K}_u}^{(2)}} \tilde{g}^{(2)}\left(\boldsymbol{u}^{(2)}\right)$
Non-cooperative distributed (Nash)	$\min_{\boldsymbol{u}^{(1)} \in {\mathcal{K}_u}^{(1)}} g^{(1)}\left(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}\right)$	$\min_{\boldsymbol{u}^{(2)} \in {\mathcal{K}_u}^{(2)}} g^{(2)}\left(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}\right)$
Cooperative distributed (Pareto)	$\min_{\boldsymbol{u}^{(1)} \in {\mathcal{K}_u}^{(1)}} g\left(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}\right)$	$\min_{\boldsymbol{u}^{(2)} \in {\mathcal{K}_u}^{(2)}} g\left(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}\right)$
Centralized	$\min_{\left(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}\right) \in {\mathcal{K}_u}^{(1)} \times {\mathcal{K}_u}^{(2)}} g\left(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}\right)$	

Table 2.1.: Nomenclature of control architectures for two coupled sub-systems.

2.5.2.1. Non-cooperative and cooperative distributions

An illustration of the influence of the coordination type of sub-agents is made in this paragraph through the study of two explicitly coupled sub-systems, described by the following scalar processes

$$\begin{aligned} y^{(1)} &= u^{(1)} + u^{(2)} \\ y^{(2)} &= -2u^{(1)} + u^{(2)} \end{aligned}$$

to which are respectively associated the local cost functions

$$\begin{aligned} g^{(1)}\left(u^{(1)}, u^{(2)}\right) &= \left(u^{(1)} - 5\right)^2 + \left(y^{(1)}\right)^2 \\ g^{(2)}\left(u^{(1)}, u^{(2)}\right) &= \left(u^{(2)}\right)^2 + \left(y^{(2)} - 5\right)^2 \end{aligned}$$

and the global objective function of the system, combining these two local objectives with a weight $a \in [0, 1]$

$$g_a\left(u^{(1)}, u^{(2)}\right) = a g^{(1)}\left(u^{(1)}, u^{(2)}\right) + (1 - a) g^{(2)}\left(u^{(1)}, u^{(2)}\right).$$

Two agents are considered, with respective outputs $y^{(1)}$ and $y^{(2)}$, each of them having control of the inputs $u^{(1)}$ and $u^{(2)}$ respectively.

Non-cooperative control

In non-cooperative control problems, each agent i aims at minimizing its local cost function $g^{(i)}$. From the point of view of agent 1, optimality is achieved through the resolution of

$$u^{(1)*}\left(u^{(2)}\right) \triangleq \arg \min_{u^{(1)} \in \mathcal{K}_u{}^{(1)}} g^{(1)}\left(u^{(1)}, u^{(2)}\right),$$

in order to find the best action $u^{(1)*}$ to take with respect to the action $u^{(2)}$ transmitted by agent 2, where $g^{(1)}$ explicitly write, in a convex form

$$g^{(1)}\left(u^{(1)}, u^{(2)}\right) = \left(u^{(1)} - 5\right)^2 + \left(u^{(1)} + u^{(2)}\right)^2$$

whose partial derivative with respect to $u^{(1)}$ cancels out where

$$u^{(1)*}\left(u^{(2)}\right) = \frac{5 - u^{(2)}}{2}.$$

In a similar manner agent 2 follows the optimal trajectory for any action $u^{(1)}$ from agent 1

$$u^{(2)*}\left(u^{(1)}\right) = \frac{5 + 2u^{(1)}}{2}.$$

The solution of this non-cooperative distributed control problem is the intersection $\mathbf{n}$ of these two optimal local trajectories, which is the Nash equilibrium depicted in figure 2.7

$$\mathbf{n} = \left(\frac{5}{6}, \frac{10}{3}\right).$$

Assuming that this Nash equilibrium exists, the resolution of non-cooperative problems solely requires the transmission of information. Indeed, each sub-problem being defined with respect to a local cost function, collaboration of the sub-agents is not required.

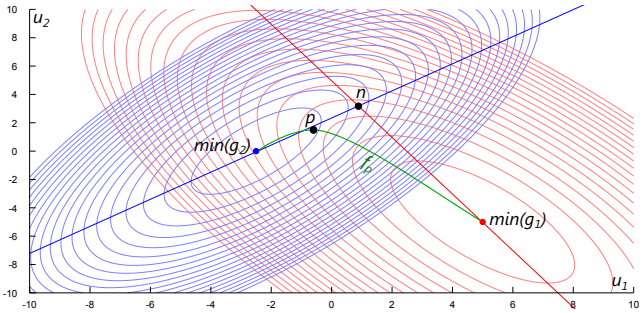

Figure 2.7.: Pareto and Nash optima for a pair of two coupled sub-systems. In the case of coupled sub-systems, the Nash and Pareto optima, respectively n and p, are not equivalent. Furthermore, the Pareto optimum is defined over an entire frontier f_p.

Cooperative control

Cooperative control problems are team problems where sub-agents cooperate to minimize a global cost function g_a.
Agent 1 therefore aims at finding the optimal trajectory

$$u^{(1)*}\left(u^{(2)}\right) \triangleq \arg \min_{\boldsymbol{u}^{(1)} \in \mathcal{K}_u{}^{(1)}} g_a\left(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}\right)$$

in order to collaborate to the minimization of the system objective function, with respect to the action $u^{(2)}$ taken by agent 2. Similarly, agent 2 solves

$$u^{(2)*}\left(u^{(1)}\right) \triangleq \arg \min_{\boldsymbol{u}^{(2)} \in \mathcal{K}_u{}^{(2)}} g_a\left(\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}\right).$$

These coupled sub-problems therefore aim conjointly at minimizing the global objective g_a over $u^{(1)}$ and $u^{(2)}$ simultaneously. The optimum of g_a with respect to both $u^{(1)}$ and $u^{(2)}$ is subsequently the solution of the coupled sub-problems. This cost function explicitly writes

$$\begin{aligned} g\left(u^{(1)}, u^{(2)}\right) = \; & a\left(u^{(1)} - 5\right)^2 + a\left(u^{(1)} + u^{(2)}\right)^2 \\ & + (1-a)\left(u^{(2)}\right)^2 + (1-a)\left(-2u^{(1)} + u^{(2)} - 5\right)^2 \end{aligned}$$

which is convex, and meets a Pareto optimum (by cancelling out the two partial derivatives) over the Pareto frontier $\mathbf{f}_p$ for $a \in [0,1]$ which writes

$$\forall a \in [0,1], \mathbf{f}_p(a) : \begin{cases} {u^{(1)}}_p(a) = -\dfrac{5(3a-2)}{7a^2 - 4a - 4} \\ {u^{(2)}}_p(a) = \dfrac{5a(7a-6)}{7a^2 - 4a - 4}. \end{cases}$$

The Pareto frontier $\mathbf{f}_p(a)$ along with the global optimum $\mathbf{p}$ of g_a for the weight $a = 1/2$ can be found on figure 2.7, with

$$\mathbf{p} = \left(-\frac{10}{17}, \frac{25}{17}\right).$$

The resolution of cooperative problems is therefore more demanding than for non-cooperative architectures: an optimization problem of a cost function over the input variables of all sub-agents is required.

This illustration demonstrates how, in the case of largely coupled systems, the Nash equilibrium is different from the Pareto global optimum, while they are coincident for independent sub-systems as illustrated in figure 2.8. The literature furthermore proves closed-loop instability cases for Nash equilibria [Venkat2006].

Non-cooperative coordination techniques are therefore not reliable in a large number of coupling cases, despite the computational gain they provide by downgrading the sub-system optimizations to a local level. The literature however

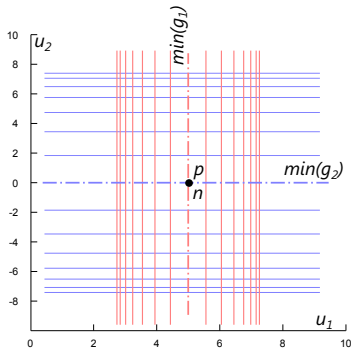

Figure 2.8.: Pareto and Nash optima for a pair of two independent sub-systems. In the case of independent sub-systems, these optima are equivalent.

proposes non-cooperative formulations with a guarantee of closed-loop stability. An example is illustrated by Camponogara in [Camponogara2002] with the Cheng and Krogh algorithm, ensuring online control stability by imposing a stability constraint on the predicted state vector [Cheng1997].
Despite these results, the gap between the Nash equilibrium and the global optimum (in a centralized perspective) in the case of heavily coupled sub-systems requires the local consideration of the influence of each action on the entire system, and thus brings a cooperative coordination approach [Venkat2006].

The works of Zhang and Li, in this perspective, propose in [Zhang2007] the minimization by each sub-agent of its own local cost and of the objective of its directly connected neighbors, in an unconstrained problem. Venkat *et al.* [Venkat2008] present a less versatile approach which exploits at the local level the knowledge of a global model of the entire system, thus allowing a local optimization of the cost function of the whole system. The induced computational cost is compensated by the properties of the algorithm regarding convergence towards a centralized optimum, stability and admissibility in the case of interruption of the optimization process. A termination criterion can thus be defined while guaranteeing stability and constraints validity of the sub-optimal solution. Liu *et al.* introduced an iterative cooperative algorithm under a Lyapunov stability constraint in their works [Liu2009]. This algorithm is a clear illustration of DMPC approaches and serves the following process

- at each control cycle, a feasible — respecting a stability constraint — initial prediction is inferred for each of the sub-agents;
- optimization iterations
  - Each sub-agent solves its respective MPC problem under the stability constraint, assuming that all other sub-systems follow the trajectory of the previous iteration, and thus compute an optimal trajectory for its input.
  - All sub-agents transmit their respective trajectories and hold the value of the global cost function with respect to this set of trajectories.
  - If a given termination criterion is met, such as a convergence threshold, each sub-agent applies the first element of its input trajectory

associated with the smallest value of the cost function. Otherwise, the algorithm performs an additional iteration.

This approach presents essential stability and convergence properties, and its main advantage is the quasi-centralized characteristic of the method: each subagents has access to the global cost function of the system. The literature presents alternatives of distributed formulations converging towards a global optimum while exploiting local cost functions. These alternatives rely on various decomposition techniques, among which is the dual decomposition method presented in the following paragraph.

2.5.2.2. Dual decomposition

Dual decomposition takes advantage of the existence of an unconstrained maximization problem, the dual problem, associated to any constrained minimization problem. This method has been applied to optimization problems with large cardinality for several decades [Dantzig1961] and the Uzawa algorithm, widely adopted constrained optimization tool, can be seen as a first formulation as a projected gradient algorithm applied to the dual problem [Arrow1958]. Applications to the control of compound systems can be found in the literature with the works of Wakasa *et al.* [Wakasa2008] and Rantzer poposes in [Rantzer2009] to consider dual decomposition techniques for distributed control. Giselsson *et al.* extended these works with an asymptotic stability condition for a distributed architecture [Giselsson2010]. To overcome the slow convergence of the proposed algorithm the authors introduce a termination criterion which guarantees a given distance to optimality.

Dual problem

The following constrained minimization problem over the variable x is considered

$$p^* \triangleq \min_x g(x) \quad \text{s.t.} \quad f_i(x) \leq 0 \text{ et } h_j(x) = 0, \forall (i,j) \in [1..m] \times [1..n], \quad (2.21)$$

where p^* is the minimal value of the cost function g to find, and m and n are respectively the numbers of equality and inequality constraints the minimization problem is subject to. The Lagrangian $\mathcal{L}$ of the problem writes

$$\mathcal{L}(x, \lambda, \nu) \triangleq g(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{j=1}^{n} \nu_i h_i(x)$$

with (λ, ν) the Lagrange multipliers of the problem. The primal problem (2.21) is equivalent to the canceling out of the Lagrangian. In the rest of this paragraph the cost function g is assumed to be convex. Let g_d denote the dual function to g defined over the set of Lagrange multipliers as

$$g_d(\lambda, \nu) \triangleq \min_x \mathcal{L}(x, \lambda, \nu)$$

whose minimization cancels out the gradient of the Lagrangian in the direction of x, convex with respect to x. In order to furthermore cancel out the components in the direction of λ and ν (the Lagrangian is concave in these directions), the following dual problem associated to problem (2.21) can be written

$$d^* \triangleq \max_{\lambda \geq 0, \nu} g_d(\lambda, \nu). \tag{2.22}$$

It can be proved that in the case of a quadratic convex cost function g and linear constraint functions f_i and h_j the dual problem is strictly equivalent to the primal problem. In other terms

$$p^* = d^*.$$

These conditions are generally met by MPC problems which typically write as QP problems (*cf.* Sec. 2.4.2).

Decomposition

The dual decomposition technique consists in replacing coupling variables between sub-systems by as many pairs of local variables under equality constraints. This constraints are then relaxed by Lagrange multipliers acting as market prices, thus drawing solutions towards a consensus between sub-systems.

An illustration of the method is provided in this paragraph in the case of two coupled sub-systems. Let us consider the following optimization problem

$$\min_{u^{(1)}, u^{(2)}} g\left(u^{(1)}, u^{(2)}\right) \text{ avec } g\left(u^{(1)}, u^{(2)}\right) = g^{(1)}\left(u^{(1)}, u^{(2)}\right) + g^{(2)}\left(u^{(2)}\right) \tag{2.23}$$

where $g^{(1)}$ et $g^{(2)}$ are the local cost functions of sub-agents 1 and 2, respectively, and are convex. The two sub-systems are in this problem coupled by the variable $u^{(2)}$, and thus must contribute to a global optimum of the objective function g. The problem (2.23) can easily be rewritten as the constrained optimization problem

$$\begin{aligned} &\min_{u^{(1)}, u^{(2)}, w} g_w\left(u^{(1)}, u^{(2)}, w\right) \\ &\text{s.t.} \quad \left\{ \begin{array}{rcl} g_w\left(u^{(1)}, u^{(2)}, w\right) & = & g^{(1)}\left(u^{(1)}, w\right) + g^{(2)}\left(u^{(2)}\right) \\ w - u^{(2)} & = & 0 \end{array} \right. \end{aligned} \tag{2.24}$$

whose Lagrangian writes, with the multiplier ν

$$\begin{aligned} \mathcal{L}\left(u^{(1)}, u^{(2)}, w, \nu\right) &\triangleq g_w\left(u^{(1)}, u^{(2)}, w\right) + \nu\left(w - u^{(2)}\right) \\ &= g^{(1)}\left(u^{(1)}, w\right) + g^{(2)}\left(u^{(2)}\right) + \nu\left(w - u^{(2)}\right). \end{aligned}$$

The function g_d dual to g_w has the form

$$\begin{aligned} g_d(\nu) &\triangleq \min_{u^{(1)}, u^{(2)}, w} \mathcal{L}\left(u^{(1)}, u^{(2)}, w, \nu\right) \\ &= \min_{u^{(1)}, u^{(2)}, w} g^{(1)}\left(u^{(1)}, w\right) + g^{(2)}\left(u^{(2)}\right) + \nu\left(w - u^{(2)}\right) \end{aligned}$$

and the problem (2.24) is thus equivalent to

$$\max_{\nu} \left\{ \min_{u^{(1)},w} \left[g^{(1)} \left(u^{(1)}, w \right) + \nu w \right] + \min_{u^{(2)}} \left[g^{(2)} \left(u^{(2)} \right) - \nu u^{(2)} \right] \right\} \tag{2.25}$$

which can be distributed into three coupled sub-problems

$$\begin{cases} \min\limits_{u^{(1)},w} \tilde{g}^{(1)} \left(u^{(1)}, w, \nu \right) \triangleq \min\limits_{u^{(1)},w} \left[g^{(1)} \left(u^{(1)}, w \right) + \nu w \right], & \text{(2.26a)} \\ \min\limits_{u^{(2)}} \tilde{g}^{(2)} \left(u^{(2)}, \nu \right) \triangleq \min\limits_{u^{(2)}} \left[g^{(2)} \left(u^{(2)} \right) - \nu u^{(2)} \right], & \text{(2.26b)} \\ \max\limits_{\nu} \tilde{g}^{(3)} \left(w, u^{(2)}, \nu \right) \triangleq \max\limits_{\nu} \nu \left(w - u^{(2)} \right). & \text{(2.26c)} \end{cases}$$

These three sub-problems (2.26) constitute a *non-cooperative* distributed architecture as each of the three cost functions, although employing shared variables, only have a local scope. The problem (2.24), and by extension (2.23), which is a cooperative optimization problem aiming at finding a global optimum for the cost function $g \triangleq g^{(1)} + g^{(2)}$ of the whole system shared by two sub-agents, is therefore reduced by dual decomposition to the search of a Nash equilibrium between the local cost functions $\tilde{g}^{(i)}$ respective to the three sub-agents (2.26). Non-cooperative methods can thus be employed to solve the resulting problem although the initial problem required the setup of cooperative algorithms. Rantzer propose for example in [Rantzer2009] to solve the dual problem with an iterative gradient descent.

The dual decomposition technique therefore allows to strictly match a Nash equilibrium to a global optimum. The centralized characteristic of cooperative distributed architectures is thus circumvented with the introduction of relaxed constraints in the form of local maximization problems, coupled to a set of local minimizations.

*

The computational cost of predictive control problems rapidly grows with the size of the controlled process and the number of constraints it is subject to. Taking advantage of the sparsity — inherent or approximated — of the problem leads to distributed architectures of lower complexity. The resulting topology of the distributed problem however severely impacts the properties of the solution: sub-optimality is conceded with sequential architectures in favor of simplicity while parallel problems are generally more representative of the actual system, non-cooperative problems assume the existence of an equilibrium between the several sub-agents and cooperative architectures may suffer from slow convergence. The diversity in coupling models and distribution methods — along with their consequences — imposes a specific attention when distributing a control problem. Distribution techniques for centralized problems are therefore essentially problem-dependent.

2.6. Conclusion

The dynamics of humanoid systems presents characteristics that control must handle and exploit. **Contact forces** characterize the motion of the system and its **stability**, and the **redundancy** of humanoid architectures allows to consider the execution of complex activities involving the simultaneous realization of elementary operations. This concurrency of actions, despite being a compelling asset, raises two major control challenges: the handling of **conflicts** between elementary objectives and the individual **performance** of each executed action with respect to the global objective of the activity.

Conflicts are generally apprehended at the level of the whole system by hierarchic control methods which are required to account for the various constraints to which the robot and its activity are subject to. A compromise is generally obtained between the different tasks to perform in the form of a consensus or a strict set of prevalence.

The execution of theses tasks is generally handled by reactive controllers which guarantee a certain control robustness with respect to unknown disturbances. However in the case of severely constrained activities or instable involved dynamics, predictive methods can augment the control performance by exploiting a preview horizon to define optimal solution trajectories.

In the case of simultaneous and conflicting operations, the predictive approach allow to generate a wider variety of behaviors by holding a temporal window to settle a **coordination strategy** between tasks and constraints instead of an **instantaneous compromise**. It is thus a favorable framework to the generation of complex behaviors for humanoid robots. The size of the problem however requires the introduction of a decomposition and coordination effort in order to exploit the topology of the system while guaranteeing a certain optimal characteristic of the produced solutions.

* * *

The challenge of multi-objective whole-body control is handled by several state-of-the-art techniques which are generally formulated at the instantaneous level. The advantages of predictive control motivates the consideration of multi-objective control over a preview horizon to produce coordination strategies between the entire set of degrees of freedom, elementary tasks and constraints. The resulting computational cost due to the size of the problem and its form however prevents practical application without considering a reduction of the control problem.

This reduction may come from the restriction of predictive approaches to a selection of tasks and from the approximation and distribution of the control problem. The diversity and span of model reduction and distribution however restrain their application on a case-by-case basis. The following chapter therefore introduces the distribution of a multi-objective predictive control problem driven by the spatial structure of the system and its objective and demonstrates how, through the exploitation of a preview window, predictive techniques allow to generate coordination strategies over time to solve conflicts between elementary tasks.

CHAPTER 3

Structure-driven decomposition of a multi-objective predictive problem: task-posture coordination

Contents

The coordination problem of postural balance with a concurrent manipulation task is handled in this chapter in a predictive framework. This multi-objective activity is particularly representative of coordination challenges humanoid robots often have to taken on, and is used in this chapter as the support for a distributed, whole-body predictive control architecture.

When performing an activity, a humanoid robot must achieve one or several given goals while maintaining its postural balance. In robotics, meeting the desired objective is commonly achieved through the setup of feedback controllers defining the *impedance* of the system with respect to an error to the target. Humans for example regulate their motor activity with the cocontraction of their muscles, ensuring a certain stiffness of their posture around the objective to reach. This impedance is however variable and is continuously adapted in order to accommodate the tonicity of the motion to the effects and constraints of the environment; this motor activity principle is gradually applied to robotic systems with a mechanical design allowing to adjust their impedance [Enoch2012] to increase the robustness of the system against external disturbances.

When executing manipulation activities, humanoid systems might be subject to external wrenches mainly related to the manipulation task, external actions which can be of great amplitude as illustrated in the various scenarii of the Darpa Robotics Challenge[1] pictured in figure 3.1. These wrenches act as disturbances not only on this operation, but also on the postural stability of the system. The instability and prevalent characteristics of postural balance tend to make it a constraint with which the main task must comply: the control problem is hence to guarantee the best performance of both of these concurrent operations. Conflicts are however to be expected: manipulation requires a high stiffness to ensure precision while postural stability would benefit from a moderate transmission of disturbances from the environment. The control problem can thus be formulated as the computation of compatible trajectories for each task; to do so, a future window can be exploited to preview the influence of the control parameters on the performance of both manipulation and posture objectives.

Predictive primitives are hence to be formulated for both tasks, accounting for the external action from the environment. While these tasks appear to have objectives independent from each other, they are both performed by a single system and couplings thus occur: coordination of the controllers is therefore required to account for these interactions. The structure of humanoid robots however suggests, as illustrated by human motion, that the upper limbs mainly support the manipulation task while the lower limbs ensure postural balance. The system can hence be seen as the aggregation of two coupled sub-systems, its upper and lower limbs, whose motion is regulated with respect to the manipulation and balance objectives, respectively.

In this chapter, the manipulation-posture coordination problem is addressed as the coordination problem between two predictive agents with independent objectives and coupled systems. The predictive control problem is first formulated

[1] `http://www.theroboticschallenge.org/`

for each sub-agent, thus building a multi-objective problem which is distributed according to the inherent structure of the whole system.

Figure 3.1.: Concurrent manipulation and balance under large disturbances. *Courtesy of DARPA*

3.1. Single objective MPC for postural balance under disturbances

Large disturbances scenarii initiated the development of various control formulations for biped balance under external actions. Kanzaki *et al.* introduced bracing behaviors to anticipate and recover from a disturbance in a predictive control framework [Kanzaki2005]. The action on the CoM is previewed with respect to heuristics (impact conditions) and information issued from stereo-vision. The interaction is modeled in this work as a disturbance on the velocity of the CoM resulting from an impact. This model is however circumscribed to a certain class of interactions, and its extension requires the identification *a priori* of the expected impact. Nevertheless, this approach demonstrates a notable gain in postural balance robustness. Prahlad *et al.* explicitly consider disturbance forces in their works [Prahlad2008]. Postural balance is compensated by the injection of torque at the ankle level, opposed to the measured external forces. This approach uniquely relies on the ankle actuators capabilities and do not distribute the compensation of the disturbance over the whole body. Morisawa *et al.* propose to consider the measured external forces as a disturbance on the ZMP [Morisawa2010]. Disturbance rejection is therefore supported by the entire body through the control of the CoM with respect to a reference ZMP position and an estimation of the disturbance.

With a view to anticipate and distribute the rejection of disturbances on the postural balance of a robot, this section addresses the consideration of known external forces on the CoM as disturbances on the ZMP, in a predictive control framework.

3.1.1. Disturbed locomotion model

A humanoid system is considered as subject to an external force $\boldsymbol{F}^{E,G}$ on its CoM G of position $\boldsymbol{c}$ in the horizontal plane, under the action of coplanar, horizontal and unilateral contacts with the ground. The position $\boldsymbol{p}$ of the ZMP is deduced from the system dynamics under certain assumptions. The Newton-Euler equations for the system[2] write at any given point A

$$M \begin{bmatrix} \ddot{\boldsymbol{c}}^T & \ddot{z} \end{bmatrix}^T = M\boldsymbol{g} + \boldsymbol{F}^{E,G} + \boldsymbol{F}^c \tag{3.1}$$

$$\dot{\boldsymbol{H}}_G + \mathbf{AG} \times M \begin{bmatrix} \ddot{\boldsymbol{c}}^T & \ddot{z} \end{bmatrix}^T = \mathbf{AG} \times \left(M\boldsymbol{g} + \boldsymbol{F}^{E,G} \right) + \boldsymbol{\mathcal{M}}_A^c \tag{3.2}$$

where M is the total mass of the system centered around G, $\boldsymbol{g}$ the gravity vector, $\boldsymbol{F}^c$ the contact wrenches at the feet and z the CoM altitude to the ground. $\boldsymbol{H}_G$ is the angular momentum of the robot around its CoM, whose rate of change is neglected in the following, and $\boldsymbol{\mathcal{M}}_A^c$ denotes the resulting moment of the contact wrenches around A, *i.e.* $\boldsymbol{\mathcal{M}}_A^c \triangleq \mathbf{AG} \times \boldsymbol{F}^c$. Let us consider the case where the external force do not compensate the effects of gravity, *i.e.* $\left\| \left(\boldsymbol{F}^{E,G} \cdot \mathbf{z} \right) \right\| < \| M\boldsymbol{g} \|$, with $\mathbf{z}$ such that $M\boldsymbol{g} = -\| M\boldsymbol{g} \| \mathbf{z}$. The CoP P of coordinates $\boldsymbol{p}$ in the contact plane is by definition the point where the moment of contact wrenches is null, *i.e.*

$$\boldsymbol{p} \quad \text{s.t.} \quad \boldsymbol{\mathcal{M}}_P^c = \mathbf{0}.$$

Equation (3.2) therefore brings, under the hypothesis of null CoM vertical velocity and acceleration

$$\boldsymbol{p} = \boldsymbol{c} - \frac{Mz}{Mg - \left(\boldsymbol{F}^{E,G} \cdot \mathbf{z} \right)} \ddot{\boldsymbol{c}} + \frac{z}{Mg - \left(\boldsymbol{F}^{E,G} \cdot \mathbf{z} \right)} \boldsymbol{F}^{E,G}, \tag{3.3}$$

where g is the gravity amplitude and $\left(\boldsymbol{F}^{E,G} \cdot \mathbf{z} \right)$ the vertical component of the force $\boldsymbol{F}^{E,G}$ on the CoM. This equation can be assimilated to the dynamics of a linear inverted pendulum, of constant altitude and subject to an external action $\boldsymbol{F}^{E,G}$ applied to its CoM G of position $\boldsymbol{c}$, illustrated in figure 3.2.
Additional details on the hypotheses enabling this simplified model (3.3) of the system dynamics can be found in Sec. 2.4.3, p. 36.
Relating the CoP position to the CoM dynamics and the external force applied to the system, it can be interpreted as an extension of the classical ZMP model [Kajita2003, Wieber2006] : an additional input, the external force $\boldsymbol{F}^{E,G}$, is required to output the CoP position.

3.1.2. Model predictive control

This model is exploited to formulate a predictive control primitive for the locomotion objective. Since inputs $\boldsymbol{c}$ and $\ddot{\boldsymbol{c}}$ are related by definition, their use is redundant. Let the input variable $\boldsymbol{u}$

$$\boldsymbol{u} \triangleq \frac{\partial \ddot{\boldsymbol{c}}}{\partial t}$$

[2] the 6 first lines of the equation (2.1), p. 23, related to the floating-base of the system

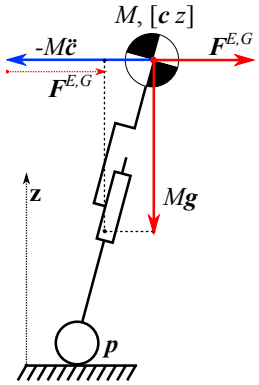

Figure 3.2.: Linear Inverted Pendulum model under an external disturbance $\boldsymbol{F}^{E,G}$. The system of mass $M$is reduced to its center of mass, its dynamics taking the form of an inverted pendulum articulated around the center of pressure $\boldsymbol{p}$.

denote the CoM jerk of the system. In discrete time of sampling period δt, an explicit integration scheme can be written as follows

$$\hat{\boldsymbol{c}}_k \triangleq \begin{bmatrix} \boldsymbol{c}_k \\ \dot{\boldsymbol{c}}_k \\ \ddot{\boldsymbol{c}}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{c}_{k-1} + \delta t \dot{\boldsymbol{c}}_{k-1} + \frac{\delta t^2}{2}\ddot{\boldsymbol{c}}_{k-1} + \frac{\delta t^3}{6}\boldsymbol{u}_{k-1} \\ \dot{\boldsymbol{c}}_{k-1} + \delta t \ddot{\boldsymbol{c}}_{k-1} + \frac{\delta t^2}{2}\boldsymbol{u}_{k-1} \\ \ddot{\boldsymbol{c}}_{k-1} + \delta t \boldsymbol{u}_{k-1}. \end{bmatrix} \tag{3.4}$$

where $\bullet_k \triangleq \bullet(k\delta t)$ and $\hat{\boldsymbol{c}}_k$ thus concatenates the CoM state at time $k\delta t$. Relation (3.4) linearly translates the relation between $\boldsymbol{c}$ and $\ddot{\boldsymbol{c}}$, in discrete time. From equations (3.3) and (3.4), the following linear process in discrete time can be deduced, for a given altitude z of its CoM

$$\left\{ \begin{array}{rcl} \hat{\boldsymbol{c}}_{k+1} & = & \mathbf{A}\hat{\boldsymbol{c}}_k + \mathbf{B}\boldsymbol{u}_k \\ \boldsymbol{p}_k & = & \mathbf{C}_k\hat{\boldsymbol{c}}_k + \frac{z}{\alpha_k}\boldsymbol{F}_k^{E,G} \end{array} \right. \tag{3.5}$$

where

$$\alpha \triangleq Mg - \left(\boldsymbol{F}^{E,G} \cdot \mathbf{z}\right)$$

denotes the descending vertical component of the forces on the CoM and

$$\mathbf{A} \triangleq \begin{bmatrix} 1 & \delta t & \delta t^2/2 \\ 0 & 1 & \delta t \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B} \triangleq \begin{bmatrix} \delta t^3/6 \\ \delta t^2/2 \\ \delta t \end{bmatrix}, \quad \mathbf{C}_k \triangleq \begin{bmatrix} 1 & 0 & -\frac{Mz}{\alpha_k} \end{bmatrix},$$

as previously introduced in Sec. 2.4.3.
The set of inputs $\left\{\boldsymbol{c}, \ddot{\boldsymbol{c}}, \boldsymbol{F}^{E,G}\right\}$ to the disturbed locomotion model is thus reduced to $\left\{\boldsymbol{u}, \boldsymbol{F}^{E,G}\right\}$ through the integration scheme (3.4). The altitude z of the CoM is assumed constant in this problem, consistently with the null vertical velocity hypothesis.

With a formulation of the locomotion objective identical to the ZMP Preview Control [Kajita2003], the predictive control problem of the disturbed balance task aims at computing from this model an optimal trajectory of CoM, minimizing over N time steps the difference between the ZMP $\boldsymbol{p}$ and a reference $\boldsymbol{p}^r$ given

as ensuring safe postural balance. At control cycle k, the following objective function to minimize is thus defined, with $\boldsymbol{\mathcal{U}}_{k+N|k} \triangleq \left\{\boldsymbol{u}_{k|k}, \dots, \boldsymbol{u}_{k+N-1|k}\right\}$,

$$g_k\left(\hat{\boldsymbol{c}}_k, \boldsymbol{\mathcal{U}}_{k+N|k}\right) = \sum_{i=0}^{N-1} g_{i|k}\left(\hat{\boldsymbol{c}}_{k+i|k}, \boldsymbol{u}_{k+i|k}\right) \tag{3.6}$$

where, at previewed time $k+i$, the error $g_{i|k}$ to the objective writes

$$g_{i|k}\left(\hat{\boldsymbol{c}}_{k+i|k}, \boldsymbol{u}_{k+i|k}\right) \triangleq \omega_p \left\|\boldsymbol{p}_{k+i|k} - \boldsymbol{p}^r_{k+i|k}\right\|^2 + \left\|\boldsymbol{u}_{k+i|k}\right\|^2$$

which aggregates a tracking term of reference $\boldsymbol{p}^r$ and a regularizing term on the inputs $\boldsymbol{u}$, weighted by the positive scalar ω_p. The locomotion objective is therefore written in these terms as the tracking of a predefined path of safe CoP positions. This path corresponds to the evolution of the center of the contact convex hull, thus enabling the greatest margin possible homogeneously in all horizontal directions. A QP-form of the MPC problem is deduced

$$\begin{aligned} \text{Find } {\boldsymbol{\mathcal{U}}^*}_{k+N|k} = \arg \left[\min_{\boldsymbol{\mathcal{U}}_{k+N|k}} \left[\sum_{i=1}^{N} g_{i|k}\left(\hat{\boldsymbol{c}}_{k+i|k}, \boldsymbol{u}_{k+i|k}\right) \right] \right] \\ \text{s.t.} \quad \begin{cases} (3.5), \\ \hat{\boldsymbol{c}}_{k|k} = \hat{\boldsymbol{c}}_k. \end{cases} \end{aligned} \tag{3.7}$$

An additional set of linear inequality constraints $\boldsymbol{p}_{k+i|k} \in \mathcal{K}_{k+i|k}$, where $\mathcal{K}$ is the convex hull of the contacts with the ground, $\boldsymbol{p} \in \mathcal{K}$ defining can be appended to write the stability condition of the postural balance[3] without prejudicing the QP-form. However, as exhibited in the results section of this chapter, this constraint is not required in reasonably disturbed conditions and can be naturally appended if required in other scenarii.

Similarly to the problem presented in section 2.4.3, the quadratic control problem can be solved with QP algorithms if inequality-constrained or in closed-form otherwise. Denoting $\boldsymbol{\mathcal{P}}_{k+N|k} \triangleq \left\{\boldsymbol{p}_{k+1|k}, \dots, \boldsymbol{p}_{k+N|k}\right\}$, the process described in equation (3.5) can be developed in matrix form over the preview horizon

$$\boldsymbol{\mathcal{P}}_{k+N|k} = \mathbf{P}^\alpha_x \hat{\boldsymbol{c}}_k + \mathbf{P}^\alpha_u \boldsymbol{\mathcal{U}}_{k+N|k} + \boldsymbol{\mathcal{P}}^\alpha_{k+N|k}, \tag{3.8}$$

where

$$\mathbf{P}^\alpha_x \triangleq \begin{bmatrix} 1 & \delta t & \frac{\delta t^2}{2} - \frac{Mz}{\alpha_{k+1}} \\ \vdots & \vdots & \vdots \\ 1 & N\delta t & N^2\frac{\delta t^2}{2} - \frac{Mz}{\alpha_{k+N}} \end{bmatrix}, \quad \mathbf{P}^\alpha_u \triangleq \begin{bmatrix} \frac{\delta t^3}{6} - \delta t \frac{Mz}{\alpha_{k+1}} & \cdots & 0 \\ \vdots & \ddots & 0 \\ (1+3N^2)\frac{\delta t^3}{6} - \delta t \frac{Mz}{\alpha_{k+N}} & \cdots & * \end{bmatrix}$$

and

$$\boldsymbol{\mathcal{P}}^\alpha_{k+N|k} \triangleq \left\{ \begin{array}{ccc} \frac{z}{\alpha_{k+1}} \boldsymbol{F}^{E,G}_{k+1} & \cdots & \frac{z}{\alpha_{k+N}} \boldsymbol{F}^{E,G}_{k+N} \end{array} \right\},$$

which represents the horizon of disturbances on the ZMP associated to a given horizon of forces $\boldsymbol{F}^{E,G}$ on the CoM.

Problem (3.7) henceforth holds the closed-form solution ${\boldsymbol{\mathcal{U}}^*}_{k+N|k}$

$${\boldsymbol{\mathcal{U}}^*}_{k+N|k} = -\left(\mathbf{P}^{\alpha T}_u \mathbf{P}^\alpha_u + \frac{1}{\omega_p}\mathbf{I}\right)^{-1} \mathbf{P}^{\alpha T}_u \left[\mathbf{P}^\alpha_x \hat{\boldsymbol{c}}_k - \left(\boldsymbol{\mathcal{P}}^r_{k+N|k} - \boldsymbol{\mathcal{P}}^\alpha_{k+N|k}\right)\right], \tag{3.9}$$

[3] *cf.* Sec. 2.4.3, p. 36

where $\boldsymbol{\mathcal{P}}^r_{k+N|k} \triangleq \left\{\boldsymbol{p}^r_{k+1|k}, \ldots, \boldsymbol{p}^r_{k+N|k}\right\}$ is the horizon of reference ZMP positions, defining the locomotion objective. The optimal horizon of inputs $\boldsymbol{\mathcal{U}}_{k+N|k}$ defines a desired trajectory of the CoM through equation (3.4). The instantaneous rejection of unknown disturbances, such as model errors, can be handled for example at a faster rate by a feedback controller which tracks an interpolation of this optimal desired trajectory. The set of reference CoP paths $\boldsymbol{\mathcal{P}}^r_{k+N|k}$ is, in such an architecture, translated through the previewed optimization (3.7) into an optimal trajectory of the CoM. This type of architecture allows to benefit from the predictive approach while alleviating the drawbacks issued from its computational cost: model reduction and slow control rate, as illustrated with the two-layered predictive control architecture in Sec. 1.4.2, figure 1.10.

3.1.3. Predictive and reactive control primitives

Assuming the weighted reactive framework presented in Sec. 2.2.2.2 as coordinating reactive primitives at the instantaneous level, this two-layered control architecture can be envisioned as depicted in figure 3.3.

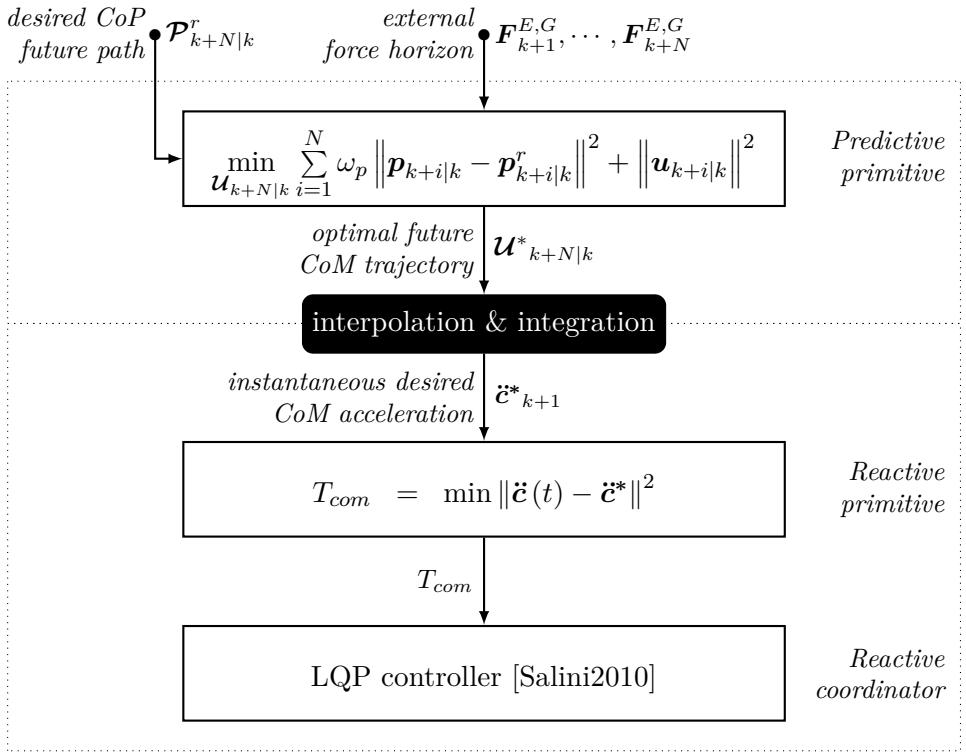

Figure 3.3.: Illustration of a two-layered control architecture for the locomotion problem combining predictive and reactive control primitives.

The optimal horizon $\boldsymbol{\mathcal{U}}^*{}_{k+N|k}$ of CoM jerks resulting from the predictive primitive (3.7) implicitly describes an optimal CoM trajectory. An integration process of these optimal jerks is thus performed, and only the first element of this trajec-

tory is used at this step until the next computation.[4] This first CoM trajectory element can be written as an instantaneous desired CoM acceleration $\ddot{\boldsymbol{c}}^*$ to be reached at the next control step $k+1$.
In a weighted reactive framework[5], reaching this desired CoM trajectory is achieved through the definition of the task function T_{com} minimizing the error between the *effective* CoM acceleration $\ddot{\boldsymbol{c}}(t)$ and the *desired* one $\ddot{\boldsymbol{c}}^*$. This task function,which represents a reactive, instantaneous control primitive, writes for example

$$T_{com}\left(\boldsymbol{q}, \boldsymbol{\nu}, \boldsymbol{\tau}, \boldsymbol{F}^c\right) \triangleq \left\|\ddot{\boldsymbol{c}} - \ddot{\boldsymbol{c}}^*\right\|^2 .$$

This reactive primitive can thereupon be instantaneously coordinated at the whole-body level with other primitives, using hierarchic architectures with strict, weighted or cascaded priorities, to output admissible whole-body motor inputs ; such an approach thus results in a decentralized predictive architecture as illustrated in Sec. 1.4.1, figure 1.8, as the reactive level solely is coordinated.

3.2. Single objective MPC for manipulation under disturbances

The robotics literature presents control methods for manipulation tasks by defining an expected reactive behavior of the manipulator in interaction, with the example of impedance control [Hogan84] whose parameters can be defined in an optimal manner [Love95]. Feedback controllers generally define dynamics around a desired objective, such as reference position, velocity or acceleration or desired interaction characteristics. Predictive controllers can be appended to the reactive control level to optimally specify this *a priori* objective[6]. In the case of manipulation tasks, the objective is commonly a set of desired positions and velocities of the effector, objective defined by the task to accomplish. While predictive approaches can be employed to adjust these desired trajectories, it is applied in this section to the adaptation of the controller parameters such as stiffness or damping as these parameters embody more explicitly the effects of the adaptation.

The formulation thus considered is notably distinct from the approach with biped balance: a model of the system behavior *in closed-loop* is previewed over a future horizon to compute an optimal horizon of the feedback controller parameters. In this section, a future horizon of disturbances on the effector is assumed to be known. In the case of manipulation tasks this assumption is legitimate as the interaction is voluntarily engaged (the timing of the interaction can thus be deduced) and effectors are generally equipped with sensors providing sufficient information to estimate a force profile at the contact. The predictive controller aims at regulating in an anticipative manner the apparent stiffness and damping of the system in order to maximize the performance of a manipulation task with respect to a known external force.

[4] Note that an interpolation process might be required if the time-discretization of the preview horizon in the MPC problem is not consistent with the sampling control period of the reactive control level.

[5] *cf.* Sec. 2.2.2.2, p. 28

[6] *cf.* Sec. 1.4.2, figure 1.10, p. 15

Nevertheless, the principle supporting this approach is similar to the developments of Sec. 3.1: predefined inputs to the system are adjusted by a predictive primitive, feeding its outputs to a reactive controller for instantaneous motor control.

With $\hat{\boldsymbol{x}} \triangleq [\boldsymbol{x}\ \dot{\boldsymbol{x}}\ \ddot{\boldsymbol{x}}]^T$ denoting the dynamics state of the effector, the manipulation objective is defined in this problem as the minimization of

$$\min_{\boldsymbol{\tau}} \|\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}^r\|^2$$

where $\hat{\boldsymbol{x}}^r$ represents a desired state of the effector. In a reactive framework, a feedback controller can be defined with the task function

$$T_{hand}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\tau}, \boldsymbol{F}^c\right) \triangleq \|\ddot{\boldsymbol{x}} - \ddot{\boldsymbol{x}}^*\|^2$$

which tends to minimize the manipulation objective at each control cycle, with a proportional-derivative controller for example

$$\ddot{\boldsymbol{x}}^* \triangleq K_p\left(\boldsymbol{x}^r - \boldsymbol{x}\right) + K_d\left(\dot{\boldsymbol{x}}^r - \dot{\boldsymbol{x}}\right), \tag{3.10}$$

where K_p and K_d are the proportional and derivative gains. According to this setup, a predictive approach is formulated in this section to define optimal gains K_p and K_d with respect to the manipulation objective and the external force on the effector.

3.2.1. Disturbed manipulation model

A partial model of the system is considered, corresponding to the upper body which mainly supports the manipulation activity. To derive a local model of the upper-body, it is assumed to be fixed to the environment and is thus considered as a fixed manipulator as illustrated in figure 3.4. Similarly to the reduced balance model presented in Sec. 3.1, this approximation aims at formulating a reduced model capturing the relevant characteristics of the manipulation dynamics to enable the setup of an MPC problem, by removing the free-floating equations from the system dynamics[7] (2.1). This assumption is discussed at the end of this section. It is therefore denoted

$$\boldsymbol{q} \triangleq \boldsymbol{q}_j, \quad \boldsymbol{n} = \boldsymbol{n}_j, \quad \boldsymbol{g} = \boldsymbol{g}_j,$$

and full actuation is assumed, that is $n_a = n$.

Whole-body control consists in finding, with respect to a dynamics model of the manipulator, the joint torques $\boldsymbol{\tau}$ such that, for a state $(\boldsymbol{q}, \dot{\boldsymbol{q}})$ of the system

$$\mathbf{M}\left(\boldsymbol{q}\right)\ddot{\boldsymbol{q}}^* + \boldsymbol{n}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right) + \boldsymbol{g}\left(\boldsymbol{q}\right) = \boldsymbol{\tau} \tag{3.11}$$

where $\ddot{\boldsymbol{q}}^*$ verifies

$$\mathbf{J_e}\left(\boldsymbol{q}\right)\ddot{\boldsymbol{q}}^* = \ddot{\boldsymbol{x}}^* - \dot{\mathbf{J}}_\mathbf{e}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right)\dot{\boldsymbol{q}}$$

with $\mathbf{J_e}$ the Jacobian matrix of the effector. Model matrices $\mathbf{M}$, $\boldsymbol{g}$ and $\boldsymbol{n}$ are respectively the mass, gravity and non-linear effects matrices of the upper limbs,

[7] *cf.* Sec. 2.1, p. 23

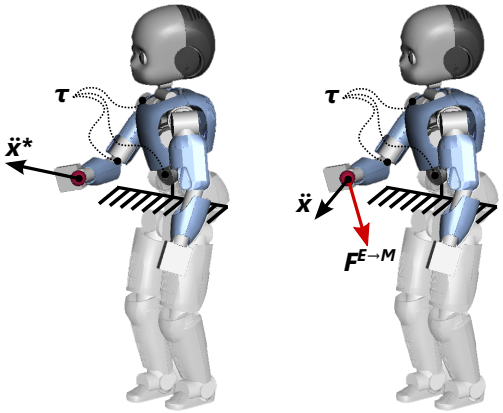

Figure 3.4.: Partial manipulation model. The upper-body is assumed to be fixed. *From left to right* — free motion, in interaction with the environment.

as described in the equations of motion[8] (2.1).
Under the effect of an action $\boldsymbol{F}^{E,M}$ from the environment at the effector, and under the joint actuation $\boldsymbol{\tau}$ solving (3.11), the system is however in the state $(\boldsymbol{q}_{\boldsymbol{\tau}}, \dot{\boldsymbol{q}}_{\boldsymbol{\tau}})$ and its dynamics write

$$\mathbf{M}\left(\boldsymbol{q}_{\boldsymbol{\tau}}\right)\ddot{\boldsymbol{q}} + \boldsymbol{n}\left(\boldsymbol{q}_{\boldsymbol{\tau}}, \dot{\boldsymbol{q}}_{\boldsymbol{\tau}}\right) + \boldsymbol{g}\left(\boldsymbol{q}_{\boldsymbol{\tau}}\right) = \boldsymbol{\tau} + \mathbf{J_e}^{T}\left(\boldsymbol{q}_{\boldsymbol{\tau}}\right)\boldsymbol{F}^{E,M} \tag{3.12}$$

where $\ddot{\boldsymbol{q}}$ and $\dot{\boldsymbol{q}}_{\boldsymbol{\tau}}$ verify

$$\mathbf{J_e}\left(\boldsymbol{q}_{\boldsymbol{\tau}}\right)\ddot{\boldsymbol{q}} + \dot{\mathbf{J}}_{\mathbf{e}}\left(\boldsymbol{q}_{\boldsymbol{\tau}}, \dot{\boldsymbol{q}}_{\boldsymbol{\tau}}\right)\dot{\boldsymbol{q}}_{\boldsymbol{\tau}} = \ddot{\boldsymbol{x}}$$

which relates the cartesian acceleration of the effector under $\boldsymbol{F}^{E,M}$. The equations (3.11) and (3.12) are then written in the operational space [Khatib1987], which brings when subtracting equation (3.12) to (3.11) in the cartesian space of at the effector

$$\boldsymbol{\Lambda}\left(\boldsymbol{q}_{\boldsymbol{\tau}}\right)\ddot{\boldsymbol{x}} - \boldsymbol{\Lambda}\left(\boldsymbol{q}\right)\ddot{\boldsymbol{x}}^{*} = \boldsymbol{\Delta}_{n,g} + \boldsymbol{\Lambda}\left(\boldsymbol{q}\right)\dot{\mathbf{J}}_{\mathbf{e}}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right)\dot{\boldsymbol{q}} - \boldsymbol{\Lambda}\left(\boldsymbol{q}_{\boldsymbol{\tau}}\right)\dot{\mathbf{J}}_{\mathbf{e}}\left(\boldsymbol{q}_{\boldsymbol{\tau}}, \dot{\boldsymbol{q}}_{\boldsymbol{\tau}}\right)\dot{\boldsymbol{q}}_{\boldsymbol{\tau}} + \boldsymbol{F}^{E,M},$$

where

$$\boldsymbol{\Lambda} \triangleq \left(\mathbf{J_e}\mathbf{M}^{-1}\mathbf{J_e}^{T}\right)^{-1}$$

represents the operational mass of the system at the effector, and

$$\boldsymbol{\Delta}_{n,g} \triangleq \boldsymbol{n}\left(\boldsymbol{q}_{\boldsymbol{\tau}}, \dot{\boldsymbol{q}}_{\boldsymbol{\tau}}\right) + \boldsymbol{g}\left(\boldsymbol{q}_{\boldsymbol{\tau}}\right) - \boldsymbol{n}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right) - \boldsymbol{g}\left(\boldsymbol{q}\right).$$

In the rest of this section, the variations of the dynamics model matrices are neglected:

$$\begin{array}{lll} \mathbf{M}\left(\boldsymbol{q}_{\boldsymbol{\tau}}\right) \approx \mathbf{M}\left(\boldsymbol{q}\right) \approx \mathbf{M}, & & \boldsymbol{\Delta}_{n,g} \approx \mathbf{0}, \\ \mathbf{J_e}\left(\boldsymbol{q}_{\boldsymbol{\tau}}\right) \approx \mathbf{J_e}\left(\boldsymbol{q}\right) \approx \mathbf{J_e} & \text{and} & \dot{\mathbf{J}}_{\mathbf{e}}\left(\boldsymbol{q}_{\boldsymbol{\tau}}, \dot{\boldsymbol{q}}_{\boldsymbol{\tau}}\right) \approx \dot{\mathbf{J}}_{\mathbf{e}}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right) \approx \dot{\mathbf{J}}_{\mathbf{e}}. \end{array}$$

This hypothesis assumes that the difference between $(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $(\boldsymbol{q}_{\boldsymbol{\tau}}, \dot{\boldsymbol{q}}_{\boldsymbol{\tau}})$ induces position and velocity-related changes which can be neglected with respect to

[8] *cf.* Sec. 2.1, p. 23

acceleration-related variations. This assumption is motivated by the postulate that changes in derivative are prevalent at the instantaneous level.
With $J^\dagger$ denoting a generalized inverse of $\mathbf{J_e}$, the acceleration $\ddot{\boldsymbol{x}}$ of the effector under $\boldsymbol{F}^{E,M}$ can henceforth be written, with the expression of the desired acceleration $\ddot{\boldsymbol{x}}^*$ defined by the feedback controller (3.10)

$$\boldsymbol{\Lambda}\ddot{\boldsymbol{x}} = \boldsymbol{\Lambda}K_p\left(\boldsymbol{x}^r - \boldsymbol{x}\right) + \boldsymbol{\Lambda}K_d\left(\dot{\boldsymbol{x}}^r - \dot{\boldsymbol{x}}\right) - \boldsymbol{\Lambda}\dot{\mathbf{J}}_\mathbf{e}\mathbf{J}_\mathbf{e}{}^\dagger\left(\dot{\boldsymbol{x}} - \boldsymbol{\nu}\right) + \boldsymbol{F}^{E,M} \tag{3.13}$$

whre $\boldsymbol{\nu}$ is the velocity defined by

$$\boldsymbol{\nu} \triangleq \mathbf{J}_\mathbf{e}\dot{\boldsymbol{q}}.$$

The matrix $\mathbf{J_e}^\dagger$ is chosen as the generalized inverse of $\mathbf{J_e}$ which minimizes the kinetic energy of the system.

The simplified model (3.13) therefore describes a local cartesian model of the effector behavior under an external action $\boldsymbol{F}^{E,M}$ (*cf.* figure 3.4) with respect to the parameters $\{K_p, K_d\}$ of the feedback controller and the objectives $\{\boldsymbol{x}^r, \dot{\boldsymbol{x}}^r\}$ of the manipulation task.

Discussion

The consideration of an external force $\boldsymbol{F}^{E,M}$ could be directly considered for the computation of the whole-body actuation in (3.11). However, since the knowledge of the external action can be incomplete and subject to uncertainties, its effect on the system is solely considered at the predictive level.

Furthermore, the fixed-base model can be interpreted as the assumption that the upper limbs CoM acceleration fully compensates the external action $\boldsymbol{F}^{E,M}$ at the effector. Indeed, the consideration of the free-floating base of the system would require to consider an external action from the environment (the lower limbs, in this particular case) on the root of the upper limbs, denoted $\boldsymbol{F}$ in the case of equation (3.11) and $\boldsymbol{F}_{\boldsymbol{\tau}}$ for equation (3.12). With $\mathbf{J_F}$ the Jacobian of the point of application of this wrench, the equations of motion rewrite for the two cases

$$\begin{aligned}
\mathbf{M}\left(\boldsymbol{q}\right)\ddot{\boldsymbol{q}}^* + \boldsymbol{n}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right) + \boldsymbol{g}\left(\boldsymbol{q}\right) &= \begin{bmatrix} \mathbf{0}_6 \\ \boldsymbol{\tau} \end{bmatrix} + \mathbf{J_F}{}^T\boldsymbol{F}, \\
\mathbf{M}\left(\boldsymbol{q}_{\boldsymbol{\tau}}\right)\ddot{\boldsymbol{q}} + \boldsymbol{n}\left(\boldsymbol{q}_{\boldsymbol{\tau}}, \dot{\boldsymbol{q}}_{\boldsymbol{\tau}}\right) + \boldsymbol{g}\left(\boldsymbol{q}_{\boldsymbol{\tau}}\right) &= \begin{bmatrix} \mathbf{0}_6 \\ \boldsymbol{\tau} \end{bmatrix} + \mathbf{J_e}{}^T\left(\boldsymbol{q}_{\boldsymbol{\tau}}\right)\boldsymbol{F}^{E,M} + \mathbf{J_F}{}^T\boldsymbol{F}_{\boldsymbol{\tau}},
\end{aligned}$$

with the notation $\dot{\boldsymbol{q}} \triangleq \boldsymbol{\nu}$. Subtracting the linear lines of these equations related to the unactuated joints of the free-floating base, would write

$$M_u\left(\ddot{\boldsymbol{c}}_u - \ddot{\boldsymbol{c}}_{u\boldsymbol{\tau}}\right) = \boldsymbol{F} - \boldsymbol{F}_{\boldsymbol{\tau}} - \boldsymbol{F}^{E,M},$$

where M_u is the total mass of the upper limbs, and $\boldsymbol{c}_u$ is the upper limbs CoM cartesian position for joint positions $\boldsymbol{q}$ and $\boldsymbol{c}_{u\boldsymbol{\tau}}$ its counterpart for $\boldsymbol{q}_{\boldsymbol{\tau}}$. Assuming that the upper limbs CoM acceleration fully compensates the external action $\boldsymbol{F}^{E,M}$ at the effector, brings

$$M_u\ddot{\boldsymbol{c}}_{u\boldsymbol{\tau}} \triangleq M_u\ddot{\boldsymbol{c}}_u + \boldsymbol{F}^{E,M} \quad \Rightarrow \quad \mathbf{0} = \boldsymbol{F} - \boldsymbol{F}_{\boldsymbol{\tau}}.$$

Henceforth, an identical development to the one presented in this section can be followed and the model (3.13) therefore holds under this assumption.

3.2.2. Previewed adaptation

In order to preview the evolution of the system over a finite future horizon, the following integration scheme is defined, according to equation (3.13)

$$\left\{\begin{array}{lcl} \boldsymbol{x}_k &=& \boldsymbol{x}_{k-1} + \delta t \dot{\boldsymbol{x}}_{k-1} + \frac{\delta t^2}{2} \ddot{\boldsymbol{x}}_{k-1} \\ \dot{\boldsymbol{x}}_k &=& \dot{\boldsymbol{x}}_{k-1} + \delta t \ddot{\boldsymbol{x}}_{k-1} \\ \ddot{\boldsymbol{x}}_k &=& \boldsymbol{\Lambda}^{-1} \boldsymbol{F}_k^{E,M} + \ddot{\boldsymbol{x}}_k^* - \dot{\mathbf{J}}_\mathbf{e} \mathbf{J}_\mathbf{e}^\dagger \left(\dot{\boldsymbol{x}}_k - \boldsymbol{\nu}_k\right) \end{array}\right. \tag{3.14}$$

where $\boldsymbol{\nu}_k$ is deduced from equation (3.13) at time $k-1$

$$\boldsymbol{\nu}_k = \dot{\boldsymbol{x}}_{k-1} + \left(\boldsymbol{\Lambda} \dot{\mathbf{J}}_\mathbf{e} \mathbf{J}_\mathbf{e}^\dagger\right)^{-1} \left[\boldsymbol{\Lambda}\left(\ddot{\boldsymbol{x}}_{k-1} - \ddot{\boldsymbol{x}}_{k-1}^*\right) - \boldsymbol{F}_{k-1}^{E,M}\right],$$

and $\ddot{\boldsymbol{x}}_k^*$ denotes the desired effector acceleration deduced from the PD-feedback controller (3.10)

$$\ddot{\boldsymbol{x}}_k^* \triangleq K_{p_k}\left(\boldsymbol{x}_k^r - \boldsymbol{x}_k\right) + K_{d_k}\left(\dot{\boldsymbol{x}}_k^r - \dot{\boldsymbol{x}}_k\right).$$

The initialization $\boldsymbol{\nu}_0$ is obtained by identification on the measured state of the system. While $\boldsymbol{\Lambda}$, $\mathbf{J}_\mathbf{e}$ and $\dot{\mathbf{J}}_\mathbf{e}$ are functions of the joint coordinates and their first derivative (and therefore functions of $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$), they are assumed to be constant over the preview horizon. This approximation allows to maintain a linear form of the integration process (3.14), and can be interpreted as the assumption that the manipulator cartesian motion over the preview horizon has no influence on the generalized-to-cartesian coordinates mapping of the system.

The control problem consists in finding, at each control cycle, an horizon of task parameters K_p and K_d which minimize the manipulation objective, *i.e.* $\|\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}^r\|$. According to a measured system state $(\boldsymbol{x}_k, \dot{\boldsymbol{x}}_k)$ at time k, the predictive control problem thus writes with respect to a future profile of external actions $\boldsymbol{F}^{E,M}$

$$\begin{array}{c} \min\limits_{\boldsymbol{K}_{k+N|k}} \sum\limits_{i=1}^{N} \omega_m \left\| \hat{\boldsymbol{x}}_{k+i|k} - \hat{\boldsymbol{x}}_{k+i|k}^r \right\|^2 + r_{k+i|k}^m \\ \text{s.t.} \quad \left\{\begin{array}{l} (3.14), \\ \hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_k, \end{array}\right. \end{array} \tag{3.15}$$

where $\boldsymbol{K}_{k+N|k}$ denotes an horizon $\left\{K_{p_{k+1|k}}, \cdots, K_{p_{k+N|k}}, K_{d_{k+1|k}}, \cdots, K_{d_{k+N|k}}\right\}$ of task parameters. The terms r^m are regularization costs defined to minimize the difference between optimization variables $\{K_p, K_d\}$ and their predefined reference values $\{{K_p}^r, {K_d}^r\}$, giving preference to this particular solution.
The optimal horizon of task parameters which solves the MPC problem (3.15) is denoted ${\boldsymbol{K}^*}_{k+N|k}$. This horizon represents previewed adjustments from the reference values $\{{K_p}^r, {K_d}^r\}$ resulting in dynamics of the effector mainly minimizing a tracking error to the objectives of the manipulation task.

It can be noted that this problem is quartic due to the form of the model (3.13), the linear integration process (3.14) and the quadratic objective function of (3.15). Therefore, no convexity properties can be inferred in the general case.

3.2.3. Predictive and reactive control primitives

As previously introduced, the adjustments $\boldsymbol{K}^*{}_{k+N|k}$ of the tracking gains are employed at the instantaneous level in the proportional-derivative controller (3.10). Similarly to the locomotion problem formulation in Sec. 3.1, this results in a two-layered predictive control architecture as depicted in figure 3.5: a predictive primitive is exploited to anticipate adjustments in the predefined task inputs, which are supported at the reactive level by a tracking control primitive.

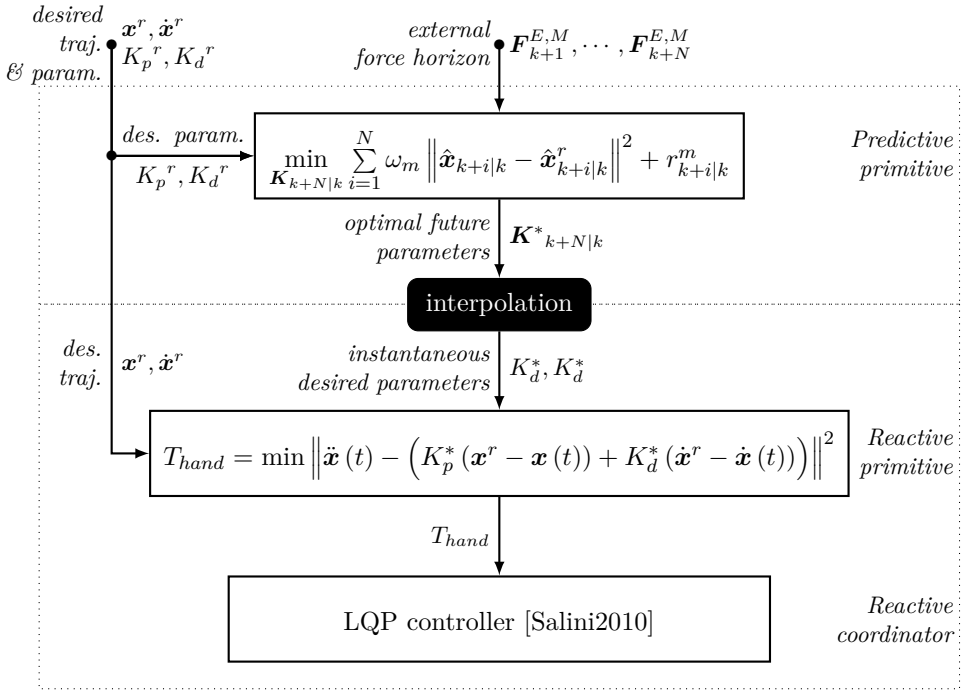

Figure 3.5.: Illustration of a two-layered control architecture for the manipulation problem combining predictive and reactive control primitives.

3.3. Multi-objective distributed MPC for task-posture coordination

Motion control of a humanoid system evolving in a dynamic environment in which it performs interaction tasks requires the guarantee of a robust control of its postural balance while adjusting its manipulation operation to conform it to constraints related to balance.

The coordination of motor activity for humans can be interpreted as a dialog between two distinct centers of the central nervous system involved in the motor control [Paillard1994]. A predictive agent decides of actions at a high level with respect to the objective of the activity, which assumes an *a priori* knowledge of expected effects of actions taken, and a reactive agent execute local motor adjustments required by the expected reactivity to the environment. The nervous system produces anticipated postural adjustments (APA) aiming at minimizing the effects of a central activity — said *focal* — on the body balance. Recipro-

cally, the motor activity related to the focal task simultaneously adjusts to the constraints issued from postural balance. The whole-body motor activity is thus organized in order to coordinate the APAs and the focal activity [Wing97] in order to maximize the performance of the actions with respect to the global activity objective. In the case of robotic manipulation, the reactive control approaches define a desired behavior of the manipulator against effects from the environment. The resulting behavior induces disturbance [Schaffer03] on the lower limbs of the robot, disturbances which must be considered at the postural balance control level, but also by the manipulation controller to adjust its behavior with respect to the state, objective and constraints of the supporting limbs it disturbs.

A multi-objective predictive control method is therefore proposed to solve the adaptation problem of the manipulation task parameters, simultaneously with the control of postural balance of a humanoid system. A distributed control architecture coordinating an horizon of inputs for both reduced representations of manipulation and balance tasks introduced in paragraphs 3.1 and 3.2 is presented as a relevant framework to the task-posture coordination problem. The dynamics of the upper limbs are previewed under the effects of an external action on the effector, which allows to evaluate an horizon of disturbances on the dynamics of the lower-limbs. The previewed dynamics of these two sub-systems are then optimized over a future horizon with respect to a criterion of both postural stability and manipulation performance.

3.3.1. Force-coupled models

The humanoid system is considered as a set of two sub-systems, the upper and lower limbs, coupled by the transmission of wrenches. The transmission of torques is neglected in this formulation for simplicity, thus reducing the problem to translational dynamics.
A reduced representation of each of these subsystems is employed under the hypotheses formulated in paragraphs 3.1 and 3.2, in order to formulate a distributed problem accounting, at a higher level of abstraction and in an anticipated manner, for the coupling between the two concurrent tasks.

The upper limbs are assumed to be subject to a known external force $\boldsymbol{F}^{E,M}$. The cartesian dynamics of the upper limbs at the effector level, developed to equation (3.13), are interpreted as writing the compensation of the inertial effects $\boldsymbol{\Lambda}\ddot{\boldsymbol{x}}$ by the external action $\boldsymbol{F}^{E,M}$ and a reaction wrench $\boldsymbol{F}^{G,M}$ from the lower limbs[9] to the upper ones. This hypothesis thus assumes

$$(3.13) \Leftrightarrow \boldsymbol{\Lambda}\ddot{\boldsymbol{x}} = \boldsymbol{F}^{E,M} + \boldsymbol{F}^{G,M},$$

and $\boldsymbol{F}^{G,M}$ is therefore deduced by identification over the model (3.13):

$$\boldsymbol{F}^{G,M} = \boldsymbol{\Lambda} K_p \left(\boldsymbol{x}^r - \boldsymbol{x}\right) + \boldsymbol{\Lambda} K_d \left(\dot{\boldsymbol{x}}^r - \dot{\boldsymbol{x}}\right) - \boldsymbol{\Lambda}\dot{\mathbf{J}}\mathbf{J}^{\dagger}\left(\dot{\boldsymbol{x}} - \boldsymbol{\nu}\right). \tag{3.16}$$

In the static equilibrium case, this assumption is verified: since each limb is in static equilibrium, the external action is serially compensated until the linking

[9] this action is the joint wrench at the fixed joint in the fixed manipulator model of figure 3.4

spherical[10] joint between the upper-body and the lower-limbs. In the general case however, motion of the lower and upper segments influence the transmitted action. The validity of this model therefore decreases with high accelerations of the limbs.

The lower-limbs are henceforth subject to three external actions: the effects of gravity, contacts with the ground and the transmitted action $-\boldsymbol{F}^{G,M}$. The translational Newton-Euler equation of this sub-system therefore writes

$$M_l \ddot{\boldsymbol{c}}_l = M_l \boldsymbol{g} + \boldsymbol{F}^c - \boldsymbol{F}^{G,M},$$

where M_l and $\ddot{\boldsymbol{c}}_l$ are respectively the total mass and CoM acceleration of the lower limbs. Consistently with the hypothesis of the upper limbs, the dynamics of the CoM $\boldsymbol{c}_u$ of the upper limbs are neglected, *i.e.* $M_u \ddot{\boldsymbol{c}}_u \ll M_l \ddot{\boldsymbol{c}}_l$, where M_u is the total mass of the upper limbs. Since the CoM $\boldsymbol{c}$ of the whole system writes

$$M\boldsymbol{c} \triangleq (M_l + M_u)\boldsymbol{c} \triangleq (M_l + M_u)\frac{M_l \boldsymbol{c}_l + M_u \boldsymbol{c}_u}{M_l + M_u},$$

it brings, with $M_u \ddot{\boldsymbol{c}}_u \ll M_l \ddot{\boldsymbol{c}}_l$,

$$M_l \ddot{\boldsymbol{c}}_l \approx M \ddot{\boldsymbol{c}}.$$

In the disturbed postural balance of Sec. 3.1, described in figure 3.2, it is thus denoted, by identification over equation (3.1):

$$\boldsymbol{F}^{E,G} = -\boldsymbol{F}^{G,M} - M_u \boldsymbol{g},$$

which is defined with respect to the effector acceleration, the external action $\boldsymbol{F}^{E,M}$ and the manipulation task parameters $\{K_p, K_d\}$ as described in the model (3.13) and deduced in equation (3.16).
Figure 3.6 presents an illustration of the coupling thus formulated between the two models.
The upper and lower limbs models are henceforth coupled by the transmitted force $\boldsymbol{F}^{G,M}$:

$$\begin{cases} \boldsymbol{\Lambda}\ddot{\boldsymbol{x}} = \boldsymbol{F}^{E,M} + \boldsymbol{F}^{G,M}(K_p, K_d) & \textit{cf.}\ (3.13) \quad (3.17a) \\ \boldsymbol{p} = \boldsymbol{p}\left(\boldsymbol{F}^{G,M}\right) & \textit{cf.}\ (3.3) \quad (3.17b) \end{cases}$$

3.3.2. Sequential distribution

The overall performance of the activity is evaluated as the sum of the manipulation and balance tasks tracking errors, respectively denoted by h^m and h^p as written in MPC sub-problems (3.15) and (3.7) respectively. The centralized, non-convex[11] MPC problem hence writes, in a weighted manner

$$\min_{\boldsymbol{u}_{k+N|k}, \boldsymbol{K}_{k+N|k}} \sum_{i=1}^{N} \omega_m h^m_{k+i|k} + r^m_{k+i|k} + \omega_p h^p_{k+i|k} + r^p_{k+i|k}$$

[10] *cf.* previous model hypothesis, torques are assumed to be not transmitted

[11] this problem can be formulated as a quadratic problem by considering the manipulation feedback controller targets as optimization variables instead of proportional and derivative gains, in the case of an horizontal external disturbance; its resolution can thus be obtained by inversion in the unconstrained case. However, the dimension and properties of the matrix therefore considered suggest the setup of iterative inversion methods for computational efficiency.

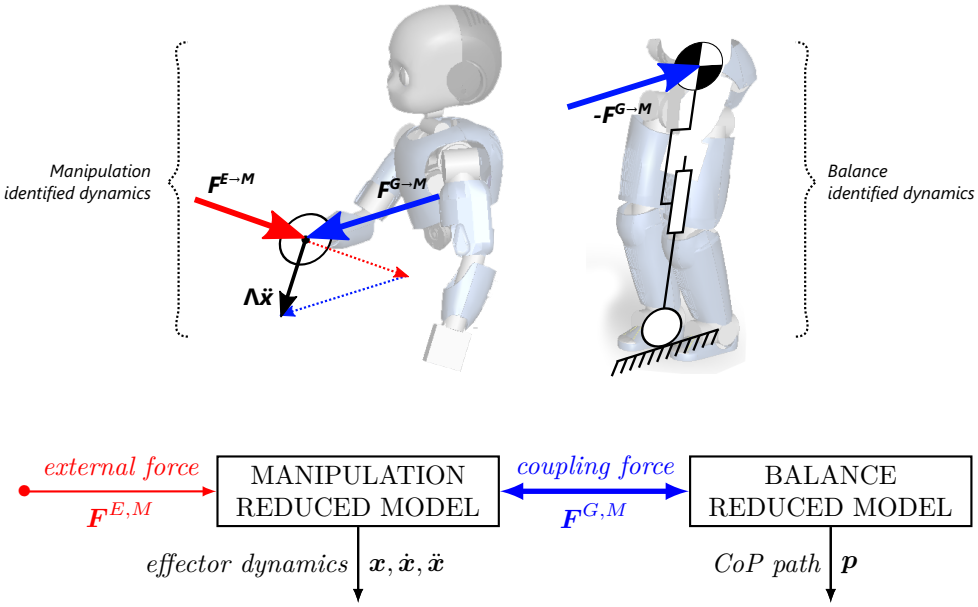

Figure 3.6.: Illustration of the force-coupled task-posture model. The models of disturbed manipulation and balance are coupled through the transmission of a force $-\boldsymbol{F}^{G,M}$ to the center of mass, resulting from the external action $\boldsymbol{F}^{E,M}$ on the hand of the robot and the manipulation dynamics.

where r^m and r^p are respectively the regularization terms on the inputs for the manipulation and balance tasks. For convenience, this problem is rewritten

$$\min_{\boldsymbol{\mathcal{U}}_{k+N|k},\boldsymbol{K}_{k+N|k}} \omega_m g_k^m\left(\boldsymbol{K}_{k+N|k}\right) + \omega_p g_k^p\left(\boldsymbol{\mathcal{U}}_{k+N|k},\boldsymbol{K}_{k+N|k}\right). \tag{3.18}$$

The two sub-objectives g^m and g^p of this cost function are strongly conflicting. Indeed, in the static equilibrium case with the targets reached, *i.e.* $\ddot{\boldsymbol{x}} = \boldsymbol{0}$ and $h^m = h^p = 0$, the manipulation sub-objective h^m with sub-model (3.17a) requires $\boldsymbol{F}^{G,M} = -\boldsymbol{F}^{E,M}$; on the other hand, the balance sub-objective h^p with sub-model (3.17b) demands $\boldsymbol{F}^{G,M} = \boldsymbol{0}$.

As written in equation (3.9), the quadratic locomotion sub-problem can be solved in closed-form, and this result can be exploited to reduce the computational cost of the centralized problem (3.18). The coupled models of (3.17) and equation (3.9) indeed write that the optimal solution ${\boldsymbol{\mathcal{U}}^*}_{k+N|k}$ to the locomotion sub-problem (3.15) is a function of the horizon of transmitted action $\boldsymbol{F}^{G,M}$, and is therefore a function of the horizon $\boldsymbol{K}_{k+N|k}$ of manipulation parameters. The following decomposition of the centralized problem (3.18) is thus performed

$$\begin{cases} \min\limits_{\boldsymbol{K}_{k+N|k}} \omega_m g_k^m\left(\boldsymbol{K}_{k+N|k}\right) + \omega_p g_k^p\left(\boldsymbol{\mathcal{U}}^*{}_{k+N|k},\boldsymbol{K}_{k+N|k}\right) & \text{(3.19a)} \\ \text{s.t.} \quad \begin{cases} \boldsymbol{\mathcal{U}}^*{}_{k+N|k} = \arg\min\limits_{\boldsymbol{\mathcal{U}}_{k+N|k}} \omega_p g_k^p\left(\boldsymbol{\mathcal{U}}_{k+N|k},\boldsymbol{K}_{k+N|k}\right), \\ \boldsymbol{\mathcal{U}}_{k+N|k} \in \mathcal{K}_U^k, \\ \boldsymbol{K}_{k+N|k} \in \mathcal{K}_K^k, \end{cases} & \text{(3.19b)} \end{cases}$$

where $\mathcal{K}_U^k$ and $\mathcal{K}_K^k$ are the admissibility domains of problems (3.15) and (3.7). This sequential distribution allows to reduce the cardinality of the considered problems and exploit the convexity of the MPC sub-problem of disturbed postural balance. Indeed, the adopted sequential resolution (3.19) consists in, at each optimization iteration on the inputs $\boldsymbol{K}_{k+N|k}$ of the manipulation task, deducing an optimal horizon ${\boldsymbol{\mathcal{U}}^*}_{k+N|k}$ of input jerks $\boldsymbol{\mathcal{U}}_{k+N|k}$ of the balance task with respect to g^p.

Discussion

This decomposition (3.19) of the problem (3.18) is not equivalent to (3.18). Let $\boldsymbol{K}^+_{k+N|k}$ denote the solution of sub-problem (3.19a). Dropping the time-dependent notation for the sake of simplicity in the rest of this section, the following relation holds:

$$(3.19) \Rightarrow \left| \begin{array}{l} \forall \boldsymbol{\mathcal{U}} \in \mathcal{K}_U, \quad \forall \boldsymbol{K} \in \mathcal{K}_K, \\ \quad \omega_m g^m\left(\boldsymbol{K}^+, \boldsymbol{\mathcal{U}}^*\right) + \omega_p g^p\left(\boldsymbol{K}^+, \boldsymbol{\mathcal{U}}^*\right) \leq \omega_m g^m\left(\boldsymbol{K}, \boldsymbol{\mathcal{U}}^*\right) + \omega_p g^p\left(\boldsymbol{K}, \boldsymbol{\mathcal{U}}\right). \end{array} \right.$$

This relation is not sufficient to solve the centralized problem (3.18). The sequential distribution therefore assumes that

$$\forall \boldsymbol{\mathcal{U}} \in \mathcal{K}_U, \quad \forall \boldsymbol{K} \in \mathcal{K}_K, \qquad g^m\left(\boldsymbol{K}, \boldsymbol{\mathcal{U}}^*\right) \leq g^m\left(\boldsymbol{K}, \boldsymbol{\mathcal{U}}\right).$$

For this hypothesis to be true in the general case, a necessary condition would be that g^m is not a function of $\boldsymbol{\mathcal{U}}$, implying that the motion of the lower-limbs have no influence on the manipulation error.

The sequential distribution (3.19) therefore assumes a unilateral coupling between the manipulation and balance models, which induces sub-optimality with respect to the manipulation objective g^m.

Nevertheless, it allows to coordinate the manipulation and balance predictive primitives (3.15) and (3.7), captured in the sub-objectives g^m and g^p, within a multi-objective MPC problem of reduced computational complexity, as validated in the following section.

3.4. Results

The control approach presented in this chapter is validated by the simulation of an iCub humanoid robot performing two concurrent tracking tasks: postural balance is ensured through the tracking of a stable ZMP reference path and the manipulation tasks consists in the tracking of a predefined reference trajectory. An illustration of this scenario is proposed in figure 3.7.
A known external force is applied at the hand of the humanoid in the longitudinal and lateral directions as shown in figure 3.8.

3.4.1. Compared controllers

In order to distinctively gain insight on the contribution of the consideration of the effort on the CoM, of the coupled model and of the weights ω_m and ω_p of the manipulation and balance predictive primitives, four controllers are compared:

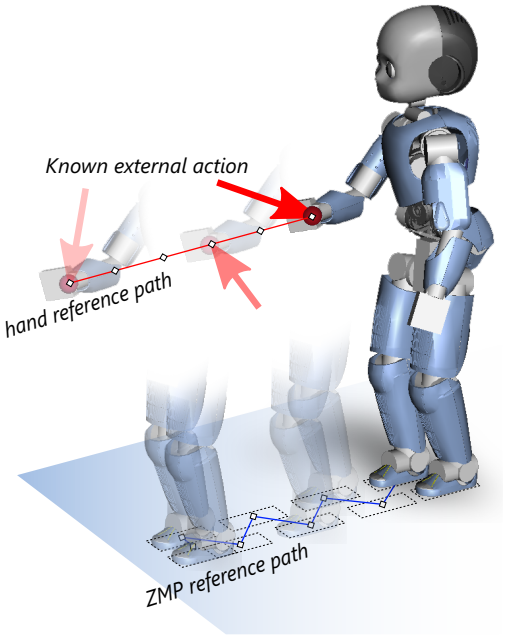

Figure 3.7.: Simulation scenario for a disturbed task-posture activity. The walking and manipulation tasks are defined as the tracking of a hand reference path and a set of center of pressure positions to achieve. A known external force is applied to the hand of the robot, where the manipulation task is defined.

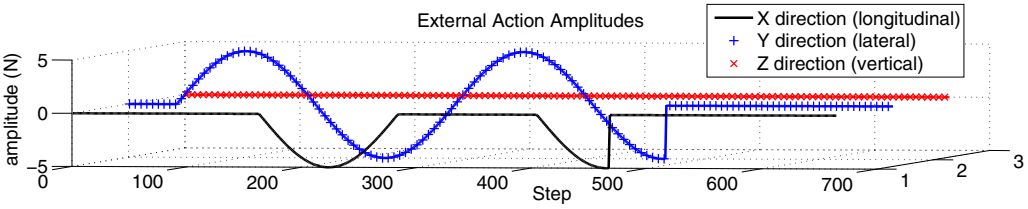

Figure 3.8.: External action known profile, applied to the hand of the robot. Semi-sinusoidal and sinusoidal force profiles are applied in the longitudinal and lateral directions, respectively.

- $\boldsymbol{\mathcal{C}_{r1}}$: ZMP Preview Control without constraints [Wieber2006],
- $\boldsymbol{\mathcal{C}_{r2}}$: ZMP Preview Control with disturbance consideration (3.7), as described in Sec. 3.1, assuming the force on the hand to be applied directly on the CoM, *i.e.* $\boldsymbol{F}^{E,G} \triangleq \boldsymbol{F}^{E,M}$,
- $\boldsymbol{\mathcal{C}_{dmp1}}$: Distributed MPC (3.19) with $\omega_m/\omega_p = 10$,
- $\boldsymbol{\mathcal{C}_{dmp2}}$: Distributed MPC (3.19) with $\omega_m/\omega_p = 0.1$.

Controller $\boldsymbol{\mathcal{C}_{dmp1}}$ favors the hand tracking task against the ZMP task in the DMPC problem (3.19), and controller $\boldsymbol{\mathcal{C}_{dmp2}}$ is parameterized in a symmetrical manner. Since predictive primitives g^m and g^p are both defined[12] as an error on

[12] sidelining low-weight regularization objectives

the translational acceleration of a frame of interest, this relative prevalence of primitives naturally translates as a relative predominance between the manipulation and locomotion objectives in the predictive coordination.

Similarly to the single-objective MPC problems for manipulation and locomotion, the outputs $\boldsymbol{K}^{+}_{k+N|k}$ and $\boldsymbol{\mathcal{U}}^{*}{}_{k+N|k}$ of the multi-objective DMPC problem (3.19) are exploited for controllers $\boldsymbol{\mathcal{C}_{dmp1}}$ and $\boldsymbol{\mathcal{C}_{dmp2}}$ at the instantaneous level by reactive control primitives. In a weighted hierarchy framework as described in [Salini2010], the following primitives are defined:

- *Manipulation primitive*: a proportional derivative feedback controller outputs a desired cartesian acceleration $\ddot{\boldsymbol{x}}^{*}$ of the hand with respect to desired position and velocity $(\boldsymbol{x}^{r}, \dot{\boldsymbol{x}}^{r})$. The associated task function T_{hand} to be minimized writes the square error between the effective hand acceleration and its desired value, $T_{hand} \triangleq \left\| \ddot{\boldsymbol{x}}\left(t\right) - \ddot{\boldsymbol{x}}^{*} \right\|^{2}$, and therefore:

$$T_{hand} \triangleq \left\| \ddot{\boldsymbol{x}}\left(t\right) - \left(K_{p}^{*}\left(\boldsymbol{x}^{r} - \boldsymbol{x}\left(t\right)\right) + K_{d}^{*}\left(\dot{\boldsymbol{x}}^{r} - \dot{\boldsymbol{x}}\left(t\right)\right) \right) \right\|^{2},$$

  where K_{p}^{*} and K_{d}^{*} are extracted at each control step as the first elements of the optimal horizon $\boldsymbol{K}^{+}_{k+N|k}$.

- *Locomotion primitive*: the optimal horizon $\boldsymbol{\mathcal{U}}^{*}{}_{k+N|k}$ defines an optimal trajectory of the CoM to track. This trajectory is integrated to extract the desired CoM acceleration $\ddot{\boldsymbol{c}}^{*}$ to be reached at the current control step. The task function T_{com} therefore associated, to be minimized, writes

$$T_{com} \triangleq \left\| \ddot{\boldsymbol{c}}\left(t\right) - \ddot{\boldsymbol{c}}^{*} \right\|^{2}.$$

These task functions are aggregated as a weighted sum to define the quadratic cost funtion of an LQP-based controller as described in [Salini2010]. Additional task functions of lower weights are appended for regularization purposes, mainly defining a reference joint configuration and driving the solution towards the minimization of torques and contact forces with the ground. Various constraints are additionally considered in a linear form, accounting for non-sliding contact conditions and joint position, velocity and torque bounds.
The resulting two-layered, predictive control architecture is depicted in figure 3.9 for the controllers $\boldsymbol{\mathcal{C}_{dmp1}}$ and $\boldsymbol{\mathcal{C}_{dmp2}}$.

The reference predictive controllers $\boldsymbol{\mathcal{C}_{r1}}$ and $\boldsymbol{\mathcal{C}_{r2}}$ are similarly supported at the instantaneous level by the same manipulation and locomotion reactive control primitives. These predictive controllers being single-objective, no task-posture coordination is performed at the predictive level, as illustrated in figure 3.10. They are introduced to provide reference performance values, and $\boldsymbol{\mathcal{C}_{r2}}$ differs from $\boldsymbol{\mathcal{C}_{r1}}$ with the consideration of the external action in the locomotion MPC problem, thus allowing to highlight of the contribution of the predictive coordination of the locomotion MPC with the manipulation predictive problem.
The weights of the manipulation and balance reactive primitives T_{hand} and T_{com} in the LQP-based reactive coordinator [Salini2010] are tuned using $\boldsymbol{\mathcal{C}_{r1}}$ resulting in satisfactory performance. The deduced weighting values are maintained for

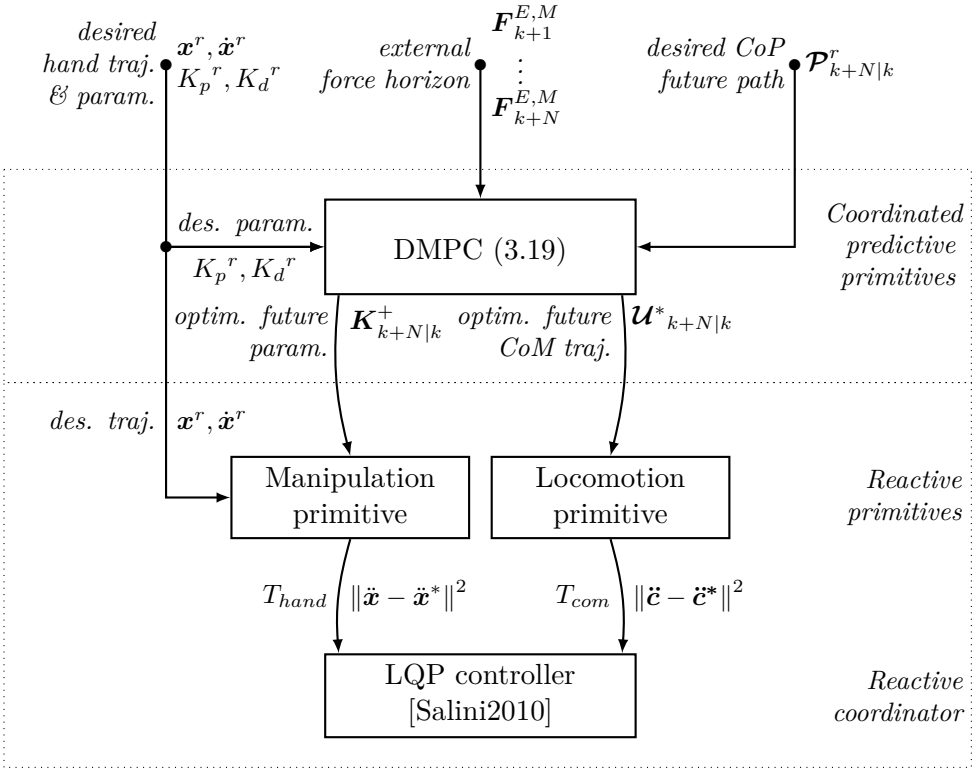

Figure 3.9.: Illustration of the multi-objective distributed predictive control architecture for controllers $\boldsymbol{\mathcal{C}_{dmp1}}$ and $\boldsymbol{\mathcal{C}_{dmp2}}$. Predictive primitives are coordinated through the resolution of the DMPC problem (3.19) and an LQP-based controller coordinates reactive primitives at the instantaneous level.

the four controllers $\boldsymbol{\mathcal{C}_{r1}}$, $\boldsymbol{\mathcal{C}_{r2}}$, $\boldsymbol{\mathcal{C}_{dmp1}}$ and $\boldsymbol{\mathcal{C}_{dmp2}}$.

Simulation was performed with the open-source dynamics simulator Arboris-Python [BarthelemyArboris] developed at ISIR. A dynamics model of the iCub robot [Sandini2007] is employed with 38 degrees of freedom and four contact points for each foot for a total mass of $\approx 27kg$. A preview horizon of $1.2s$ is considered in all predictive primitives, with a sampling period of $10^{-2}s$ equal to the control period.

3.4.2. DMPC resolution

Due to its sequential distribution, an iterative resolution of the DMPC problem (3.19) is considered in this section; iterations on the manipulation task parameters $\boldsymbol{K}_{k+N|k}$ being required solely.

To reduce the size of the DMPC problem, only proportional gains K_p are taken as optimization variables; derivative gains K_d are deduced from a critically-damped tuning of the gains, *i.e.* $K_d = 2\sqrt{K_p}$. The derivative gain K_d is indeed redundant: at any control or preview step k, any desired acceleration $\ddot{\boldsymbol{x}}^*$ can be obtained with a proportional feedback term solely. However, in case of unavailability of an optimal solution from the predictive level due to computation failure

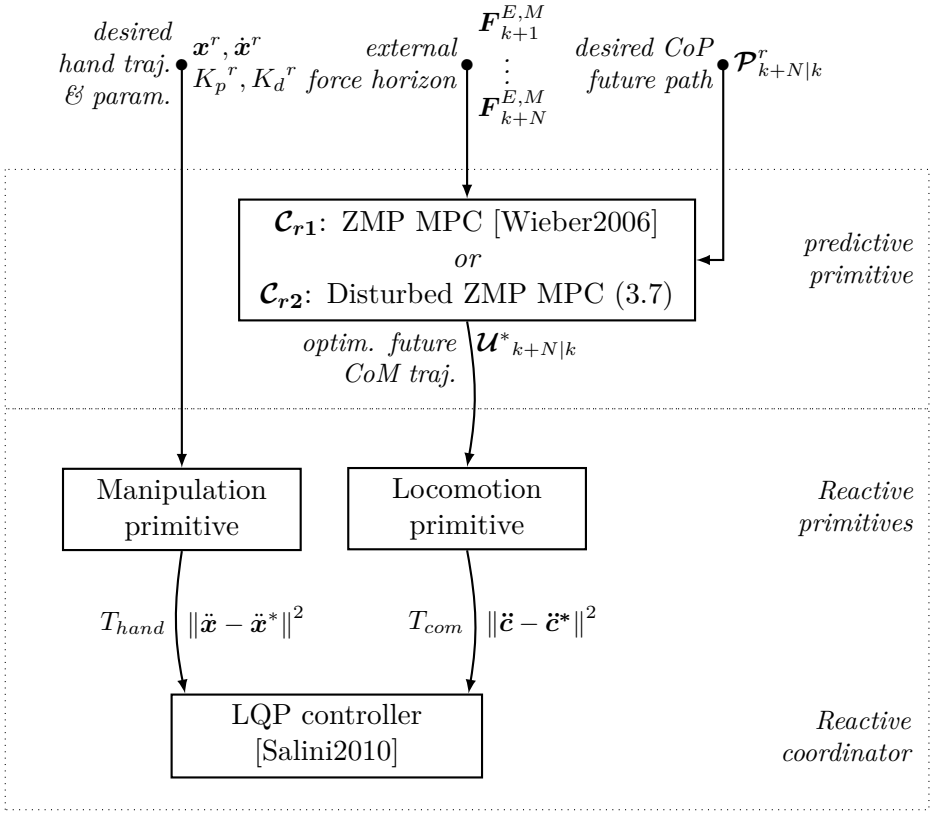

Figure 3.10.: Illustration of the single-objective predictive control architecture for reference controllers $\boldsymbol{\mathcal{C}_{r1}}$ and $\boldsymbol{\mathcal{C}_{r2}}$. A single predictive primitive is employed and an LQP-based controller coordinates reactive primitives at the instantaneous level.

for example, the employment of a stable feedback controller at the reactive level allows to safely hold the previous task parameter values. The derivative term of the feedback controller is therefore maintained, with a critically damped value as stated previously.

The sequential distribution requires, for each iteration of $\boldsymbol{K}_{k+N|k}$, to solve the locomotion sub-problem (3.7) in order to evaluate the locomotion cost g^p of the global objective of (3.19a). It therefore induces that, in the general case, the gradient value of $\omega_m g^m + \omega_p g^p$ is not available in closed-form. In the current formulation, the locomotion problem being subject to linear equalities solely[13], it is nevertheless the case as seen in (3.9). Following this remark and preserving the future ability to employ numerical methods to solve the locomotion sub-problem (3.7), a Nelder-Mead simplex algorithm [Nelder1965] is solving[14] the sub-problem (3.19a) as a not derivative-based method.
It can be noted that the dimension of the simplex problem can be reduced by imposing a piecewise-linear[15] form of the previewed adaptation strategy of ma-

[13] bounds on the CoP are indeed not considered
[14] in the case of the current problem formulation, employing a gradient-based algorithm would however enable better convergence properties
[15] of a greater sampling period

nipulation task parameters, through the linear interpolation of a reduced number of degrees of freedom $\boldsymbol{K}_{k+N|k}$.

3.4.3. Overall influence of the relative prevalence of primitives

The cumulative tracking errors for each task and each of the four controllers are presented in figure 3.11.

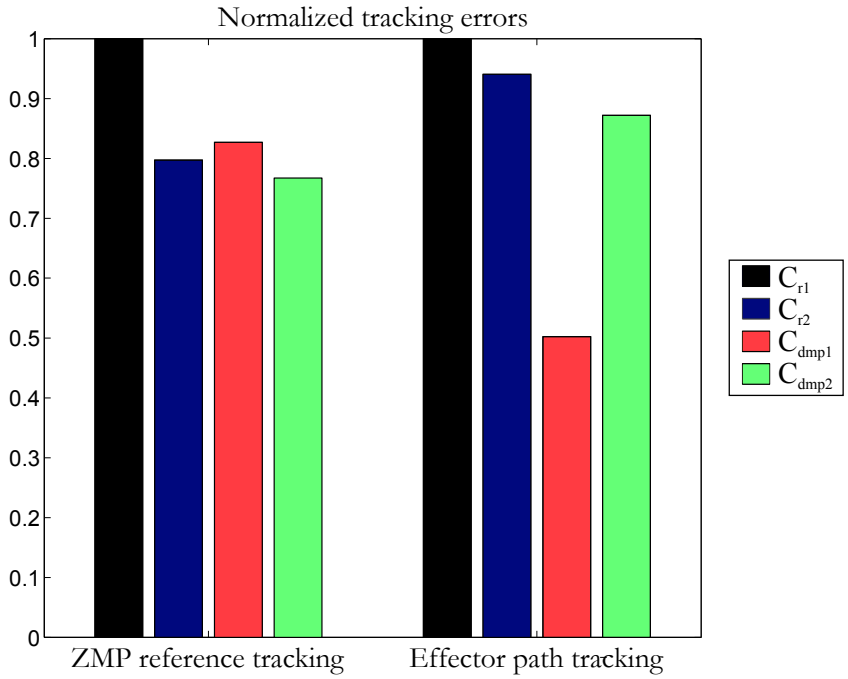

Figure 3.11.: Comparison of the cumulative errors of the manipulation and balance tasks for the four controllers. Controller $\boldsymbol{\mathcal{C}_{r1}}$ is set as a reference. The consideration of the known external action on the balance problem with controller $\boldsymbol{\mathcal{C}_{r2}}$ allows a significant gain in balance performance, and coordinated controller $\boldsymbol{\mathcal{C}_{dmp1}}$ adds a precision gain to the manipulation task by coordinating the manipulation and balance task parameters. A different weighting of manipulation and balance tasks provides an overall gain for both tasks with controller $\boldsymbol{\mathcal{C}_{dmp2}}$.

These results illustrate a notable gain — as expected — in balance performance provided by the consideration of the known force applied to the hand for the controller $\boldsymbol{\mathcal{C}_{r2}}$. In this setup, the force $\boldsymbol{F}^{E,M}$ is assumed to be directly applied to the CoM of the robot. This approximation, despite its largely arguable validity, allows the MPC problem (3.7) to anticipate disturbances generally overestimated in amplitude but in a comparable direction. Indeed, in the general case, the external action is partially compensated by the motion of the upper limbs.

The prevalence of the manipulation objective in the controller $\boldsymbol{\mathcal{C}_{dmp1}}$ and the adaptation of the manipulation task parameters in the DMPC problem (3.19) lead to a significant increase of the manipulation tracking performance, at the expense of the balance task. Indeed, figure 3.11 exhibits an error decrease greater than -40% for the locomotion objective, and a increase of around $+10\%$ for the manipulation task error, with respect to $\boldsymbol{\mathcal{C}_{r2}}$. The task-posture coordination

strategy deduced from the DMPC controller $\mathcal{C}_{dmp1}$ therefore significantly favors the manipulation over locomotion.
It can be noted that despite this increase in the ZMP cumulative tracking error, performance of the locomotion task is still higher than with controller $\mathcal{C}_{r1}$. This increase in performance can however be assumed to be mainly the result of the consideration of the known external action in $\mathcal{C}_{dmp1}$.

The symmetrical parameterization of the controller $\mathcal{C}_{dmp2}$ balances more efficiently the contribution of the manipulation and locomotion objectives in the computation of the task-posture coordination strategy. Figure 3.11 indeed shows that the use of the DMPC controller $\mathcal{C}_{dmp2}$ provides an overall gain in performance for both conflicting tasks with respect to controllers $\mathcal{C}_{r1}$ and $\mathcal{C}_{r2}$. The gain in manipulation performance of controller $\mathcal{C}_{dmp1}$ is compromised in favor of the locomotion task, resulting in decreases of around -10% and -5% in manipulation and locomotion tracking error with respect to $\mathcal{C}_{r2}$.

Discussion

The variations in overall performance between controllers $\mathcal{C}_{dmp1}$ and $\mathcal{C}_{dmp2}$ raises the issue of the definition of hierarchies in multi-objective problems.

In weighted strategies such as adopted at both the predictive and reactive levels of controllers $\mathcal{C}_{dmp1}$ and $\mathcal{C}_{dmp2}$, the relative objective weights specify the position of the optimum on the Pareto frontier; strict approaches on the other hand do not rely on such specification.
However, in the case of strict hierarchies the specification on the optimum is even more constraining, and generally on purpose, for safety reasons for example. This class of hierarchy can be interpreted as a sequential distribution of the problem, under constraints: the various objectives are sequentially reached using the solutions of the higher priority objectives.
Weighted approaches nevertheless enables a finer definition of the activity optimum: priorities can evolve continuously to translate to optimum over the Pareto frontier.

It should be noted that, when not conflicting, all sub-objectives are reached in weighted approaches as well as in strict hierarchies.
The challenge is therefore to handle conflicts, and the use of predictive formulations, providing a greater degree of freedom, allows to refine their definition. Conflicts are indeed generally temporary: defining the sub-objectives as a performance over a time window therefore allows, if the overall performance would benefit from it, to temporarily sideline one of the sub-objective, even if of a higher priority. As explicitly formulated, the global objective is indeed an overall gain in performance over time.

In this remark lies one of the main distinctions between multi-objective predictive and instantaneous control methods: while the latter defines instantaneous control policies, the former specifies strategies over time. The influence of weights in the optimum is therefore partially alleviated, as no specification is performed in predictive approaches on the instantaneous compromise between objectives.
Being able to sideline an objective nevertheless requires such infringements to be admissible: weighted approaches are therefore more appropriate in this perspec-

tive. Learning approaches can furthermore be for example envisioned to deduce appropriate weights with respect to the desired activity and external or internal percepts.

3.4.4. Coordination strategy

The adaptation strategy of the feedback controller stiffness K_p adopted by the controllers $\boldsymbol{\mathcal{C}_{dmp1}}$ and $\boldsymbol{\mathcal{C}_{dmp2}}$ is presented in figure 3.12.

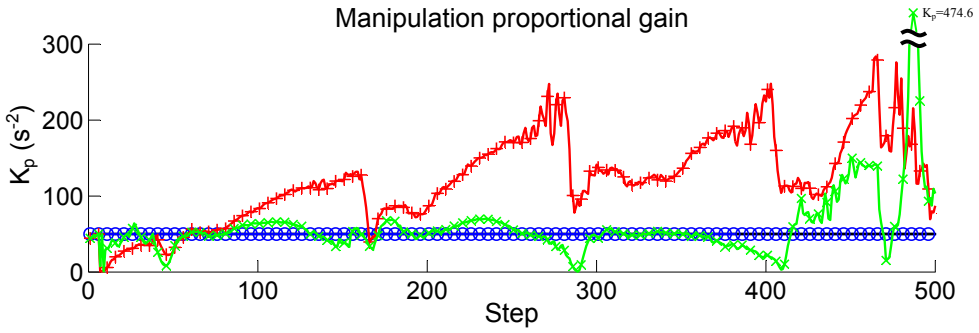

Figure 3.12.: Adaptation strategy of the feedback controller stiffness for the manipulation task. Controller $\boldsymbol{\mathcal{C}_{dmp1}}$ tends to favor high stiffness values to maintain manipulation performance, with temporary relaxations to recover balance. Controller $\boldsymbol{\mathcal{C}_{dmp2}}$ adopts a smoother strategy with a lower stiffness to compromise between manipulation and balance. Black solid: $\boldsymbol{\mathcal{C}_{r1}}$, blue circles: $\boldsymbol{\mathcal{C}_{r2}}$, red plus: $\boldsymbol{\mathcal{C}_{dmp1}}$, green crosses: $\boldsymbol{\mathcal{C}_{dmp2}}$

From an overall point of view, controller $\boldsymbol{\mathcal{C}_{dmp1}}$ tends to opt for a stiffness greater than the reference regularization value ($K_p{}^r = 50s^{-2}$) in order to prevalently minimize the manipulation tracking error. *A contrario* the strategy adopted by the controller $\boldsymbol{\mathcal{C}_{dmp2}}$ is in average close to the reference value $K_p{}^r = 50s^{-2}$ to favor postural balance.

Temporary decreases in the manipulation stiffness can be observed for both DMPC controllers around time steps 170, 290, 410 and 475 at which the external force is close to its maximum amplitude. These drops can be interpreted as a means to recover balance through the reduction of the transmitted effort $-\boldsymbol{F}^{G,M}$ to the CoM as deduced in equation (3.16).

This strategy is a clear illustration of the momentary relaxation of the manipulation sub-objective in favor of postural balance. Despite opposite priority setups between controllers $\boldsymbol{\mathcal{C}_{dmp1}}$ and $\boldsymbol{\mathcal{C}_{dmp2}}$, they both adopt a similar strategy at these critical instants. The DMPC approach thus enables coordination between the conflicting manipulation and locomotion objectives.

Moreover, the weighted priority specification does not prevent task-posture bilateral coordination: the future time window provides a sufficient degree of freedom to temporarily sideline the objective with highest priority in favor of the lowest priority objective, while still influencing over time the relative overall performance of the two manipulation and locomotion tasks.

3.4.5. Detailed performance insights

A detailed view of the evolution of the lateral position of the CoP is presented in figure 3.13.

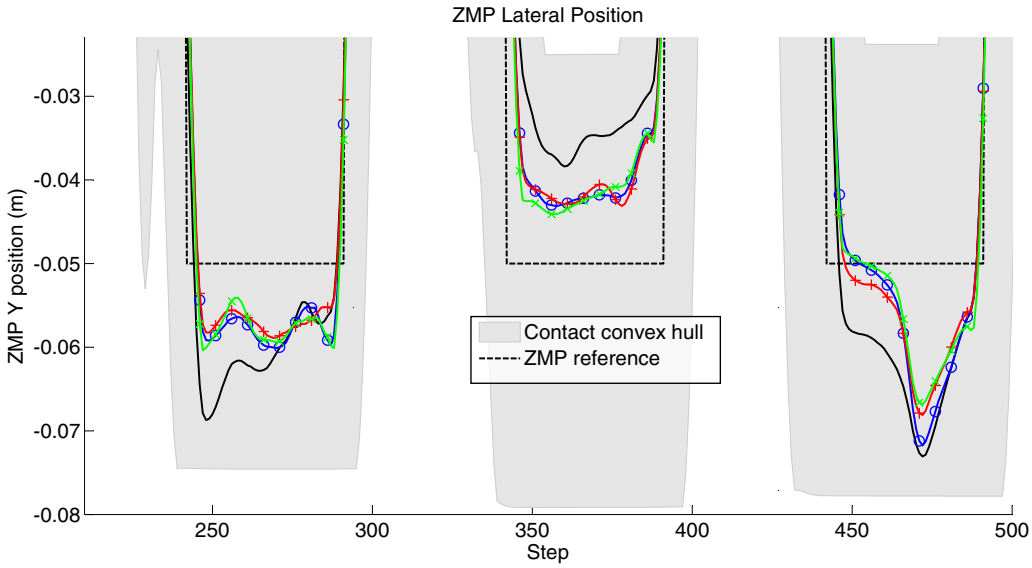

Figure 3.13.: Detail of the evolution of the CoP in the lateral direction. The overall tracking performance of controllers $\mathcal{C}_{r2}$, $\mathcal{C}_{dmp1}$ and $\mathcal{C}_{dmp2}$ is relatively equivalent. Short terms effects are however better apprehended by coordinated controllers $\mathcal{C}_{dmp1}$ and $\mathcal{C}_{dmp2}$ as depicted at time step 475. Black solid: $\mathcal{C}_{r1}$, blue circles: $\mathcal{C}_{r2}$, red plus: $\mathcal{C}_{dmp1}$, green crosses: $\mathcal{C}_{dmp2}$

The performance gain provided by the consideration of the external action is notable for controllers $\mathcal{C}_{r2}$, $\mathcal{C}_{dmp1}$ and $\mathcal{C}_{dmp2}$, for example around time steps 250 and 450 where the ZMP preview controller $\mathcal{C}_{r1}$ displays an increase in the tracking error above +100% with respect to the three other controllers.

However, short term effects such as depicted at time step 475 require a better estimation and adaptation of the transmission to the CoM of the external force, requirement which distributed controllers $\mathcal{C}_{dmp1}$ and $\mathcal{C}_{dmp2}$ meet. Indeed, a temporary relaxation of the manipulation task stiffness can be observed for both controllers on figure 3.12 around time step 475. This decrease allows a significant gain in the ZMP tracking performance, that the sole consideration of the external effort cannot provide as shown with controller $\mathcal{C}_{r2}$ in figure 3.13, thus furthermore validating the contribution of the coordinated predictive primitives in the DMPC formulation introduced in this chapter.

This distributed predictive control approach has also been applied to a scenario of task-posture coordination for a manipulation task disturbed by a succession of smooth impacts of varied amplitude. The results detailed in [Ibanez2012] demonstrate a gain in robustness for both conflicting tasks and a decreased dependence of the manipulation error to the amplitude of the disturbances.

3.4.6. Discussion

Despite the multiple approximations performed to introduce the DMPC problem (3.19), the use of a reactive layer allows to confront the previewed coordination strategy, computed with respect to reduced models, to a refined model of the system dynamics and constraints. The predictive layer is therefore employed to perform adjustments on the predefined specification of the manipulation and locomotion tasks at a higher-level, based on the preview of models capturing the major couplings between these two conflicting objectives.
The validity of the exploited reduced models is consequently of a lessened importance, since stricter constraints and models are enforced at the instantaneous level.

Nevertheless, one of the most arguable hypotheses of the development and validation of this DMPC problem is the assumption of a known future horizon of the external action. In the case of manipulation activities, this external force is resulting from a voluntary action of the robot: the time of contact and its amplitude can therefore be apprehended *a priori*, based on the desired approach trajectory and estimation of the target object characteristics. During contact, measurements of the interaction wrench can furthermore be employed to adjust this estimated profile of the external force, since most manipulator systems are equipped with FTS sensors at the effector level.

3.5. Conclusion

The future time window provided by the predictive control framework is exploited in this chapter under two different perspectives in the case of elementary tasks subject to an external action from the environment.

Although the ability of predictive control to account for constraints on the system is generally favored, the adopted approach benefits here from the preview horizon to account for a known future profile of external actions on the CoM in the case of disturbed postural balance control. An optimal future trajectory minimizing the tip-over risk is then computed with respect to this consideration and the tracking objective of the balance task.

The application of predictive control to the disturbed manipulation problem also takes advantage from the preview of a known action but with the aim to optimally define the parameters of a feedback controller, minimizing over a future horizon the error to a desired target and regularized around predefined reference parameter values. The objective of this control problem is thus to anticipate and adapt the behavior of the effector around an *a priori* expected behavior in order to maximize the task performance.

When these two elementary tasks are performed simultaneously, their objectives can be conflicting: while manipulation performance requires a high stiffness, it leads to a greater disturbance on the CoM of the system which threatens postural balance.

Considering an approximation of the structure of the system, a coupled model is formulated which accounts for the effects of one task on the other and therefore allows to anticipate coordination strategies to maximize the overall performance of both tasks. The expected behavior of the effector is thus adapted in a predictive manner conjointly with the trajectory of the CoM to reach a compromise between manipulation performance and postural balance.

To **reduce computational costs**, the multi-objective MPC problem is distributed sequentially, taking advantage from its structure. The resulting **distributed MPC** problem is validated with the simulation of disturbed task-posture activities where the controller generates coordination strategies allowing to **solve conflicts** increase the **global performance** of the activity.

These results furthermore demonstrate that, while the weighted priority employed in the formulation of the multi-objective MPC allows to define an **overall compromise** between the two conflicting sub-objectives, the previewed coordination still enables the **temporary relaxation** of the tasks of higher priority as opposed to **instantaneous compromises** defined at the reactive level. The contribution of predictive coordination strategies to the generation of complex behaviors is therefore exhibited: the consideration of conflicts over time prevents an excessive specification of priorities between objectives, which could prejudice the overall performance of the multi-objective activity. Constraints can nevertheless be naturally enforced in the predictive optimization problem, and strict hierarchies can still be employed to restrain solutions to a safe admissible domain at the instantaneous level moreover.

* * *

The approach presented in this chapter relies on arbitrary models of the subsystems supporting the elementary objectives to reach and of the coupling between them, in order to propose a suitable distributed architecture of the multi-objective predictive control problem. Indeed, the sub-systems and their objectives are identified with respect to the structure of the system and the activity to execute, and the distribution of the problem is performed accordingly.

However and as introduced in the next chapter, systematic methods can be employed to distribute a centralized problem into coupled optimization sub-problems, and explicitly expose their couplings without *a priori* assumptions on the structure of the problem. Such methods are applied to the coordination of the reduced dynamics of the system to find consensual strategies between conflicting objectives in a biped walking activity under unknown disturbances.

CHAPTER 4

An implicitly distributed problem: multi-objective balance control

Contents

This chapter introduces a multi-objective predictive approach to robust whole-body control to ensure postural stability of multi-legged locomotion systems, more particularly bipeds, against external perturbations. These perturbations, which may be of different nature (unexpected actions from the environment, uncertainty on contact conditions), affect the stability of contacts and consequently the control of the locomotion system as a whole, possibly causing the biped to tip-over or the contacts to slip. Several types of compensatory motions can be involved to improve the control of biped postural stability: horizontal [Kajita2003, Wieber2006] and vertical [Park2000] dynamics of the system, angular momentum [Pratt2006, Nenchev2008], actions induced by the upper-limbs[1] or recovery steps [Stephens2007, Herdt2010].

The framework of Distributed Model Predictive Control offers to combine in an optimal way these different motor activities. The predictive aspect of such an architecture brings robustness to postural balance through the preview of future consequences of inputs on the system, and compute coordination strategies between the various objectives and inputs accordingly. The distributed aspect allows to deal with conflicting objectives at a reduced computational cost, with the additional ability to adapt preview horizons to the dynamics of the controlled sub-processes.

A two-layered, multi-objective predictive control architecture for biped postural stability is developed in this chapter according to this approach. Several models of the system dynamics are coordinated in order to simultaneously optimize multiple balance and contacts stability objectives, previewed over different future horizons.

The control problem of balance and stable biped locomotion for humanoid robots is often tackled through the control of some characteristic point such as the Zero-Moment Point [Vukobratovic1972] or Capture Point [Pratt2006] (CP). The most robust applications of such criteria generally implement a predictive problem in order to provide anticipative features to the reaction of the system. A significant breakthrough in balance control involved a ZMP criterion in a predictive approach [Kajita2003] and later in a Model Predictive Control problem [Wieber2006], introducing explicitly a set of constraints acting on the ZMP into the optimization problem and leading to significant improvements in robustness against strong perturbations. Robust control algorithms for walking activities were developed based on CP dynamics [Englsberger2011] and are subject to further developments in a MPC framework [Krause2012].

Predictive approaches are not applied to all aspects of robot control; indeed the formulation of an optimization problem over a preview horizon tends to regulate control inputs, its main advantage being the rejection of short-term disturbances. In that sense long preview horizons are not suitable to the control of systems of high-frequency dynamics. The MPC framework yet provides an indisputable gain in robustness for balance control. However, the computational cost it adds makes it more favorable to models of the system dynamics of a higher level of abstraction. A constrained Linear Inverted Pendulum (LIP) model of constant altitude is commonly used to derive the dynamics of the ZMP or CP. In addition, a significant amount of valuable developments based

[1] *cf.* Chapter 3, p. 53

on the ZMP Preview Control framework [Kajita2003] rely solely on the control of the horizontal dynamics of the center of mass of the robot to maintain balance, and rotational effects are commonly neglected in the ZMP equations. In order to increase balance robustness against strong perturbations, such as external pushes or ground irregularities, the introduction of additional controlled degrees of freedom is necessary; focus has been recently put in this direction towards the computation of recovery steps [Stasse2009, Herdt2010, Stephens2010] to avoid the fall of the robot. These approaches hold tip-over as the main fall scenario. Contact stability is largely considered as a low-level (high priority) constraint exclusively [Sentis2007, Mansard2009, Salini2011] and no optimization is made towards the minimization of slippage risk. Nevertheless slippage is also a major concern as it leads to uncontrollable traction from the ground, and contact conditions strongly influence motor activity for example in sit-to-stand motions [Barthelemy2008].

Contribution of this chapter is directed to three aspects of predictive balance control, in an attempt to answer to the foregoing remarks. First, postural stability is maintained through both balance (tip-over) and contacts stability (slippage) maximization. Second, balance control takes advantage of both horizontal and vertical CoM dynamics, along with rotational effects through angular momentum of the robot around its CoM. Last, several models of different levels of abstraction and objectives are combined within a single distributed predictive problem, and span multiple preview horizons.

4.1. Multi-objective balance model as coupled reduced models

In highly disturbed contexts such as biped walking, the modulation of the controlled behavior of the system provides a significant gain in rejecting disturbances such as terrain irregularities [Park2001] for example. Disturbances of different nature can also be alleviated through the adjustment of the expected dynamics of the system: Lim *et al.* indeed demonstrate in [Lim2001] how the impedance adaptation of the legs allow to handle impacts resulting from the contact shifts when walking.

Results [Kim2011, Mizrahi2015] suggest that humans adapt their apparent stiffness to optimize their dynamics in order to reduce the metabolic cost of walking. The effective stiffness of the legs increases with the gait pace to support the increment of propelling energy required at greater rates, and stiffness control of the lower limbs allows to attenuate the impact forces. Adaptation strategies of the vertical dynamics of the center of mass can also be observed with downwards compensatory motions to recover from unexpected external disturbances. However, these motions are still generally unexploited in robotic controllers governing the balance problem through the control of the center of mass of the robot. As examples, the vertical velocity of the CoM of the robot is commonly assumed to be null [Kajita2003, Stephens2007, Herdt2010] or is plainly not considered as a balancing variable [Pratt2006, Englsberger2011]. Angular momentum con-

trol however demonstrated efficient contribution to the postural stability problem [Pratt2006, Stephens2007].

Furthermore, tipping-over is considered as the main falling scenario in state-of-the-art biped balance controllers. However, while widespread sensors such as inertial measurement units (IMU), force-torque sensors (FTS) and vision provide information about disturbances on and state of balance, few allow to capture data on the contact parameters which are subject to a great uncertainty. Indeed the ground supporting the feet is usually irregular in both geometry and nature, and identification of the exact contact surfaces and friction parameters is particularly challenging. A consequence of this remark is that slippage risk should be of a comparable concern as tip-over when addressing the postural balance problem.

This section thus introduces a balance control formulation minimizing both tip-over and slippage risks: contacts stability is explicitly accounted for in the preview problem along with biped balance. To achieve this objective CoM control must involve more degrees of freedom than just its horizontal dynamics: optimal future trajectories of the CoM linear acceleration and rate of angular momentum are computed online to maximize postural stability.

4.1.1. Postural stability dynamics

Postural stability of a humanoid robot is solely supported by contact reactions from the environment. A straightforward analysis of biped balance raises two main failure cases: tipping over and slippage. Such situations are commonly characterized by the relative amplitudes of the contact actions (friction) and their distribution on the contact areas (tip-over): their main influence in the system dynamics can hence be found respectively in the force and momentum expressions of the system equations of motion[2]. Two postural stability criteria are derived from the whole-body dynamics expressed at the CoM G of the system. These dynamics correspond in the Lagrange formalism to the equations of motion projected in the variety spanned by the free-floating unactuated degrees of freedom and are illustrated in figure 4.1.

Let $\boldsymbol{c}$, z respectively denote the horizontal and vertical positions of the CoM of the robot. With F_n and F_t the respective normal and tangential norms of the resulting reaction force from the ground, the force equation yields

$$F_n^2 = M^2 \left(\ddot{z} + g\right)^2, \quad F_t^2 = M^2 \|\ddot{\boldsymbol{c}}\|^2, \tag{4.1}$$

where M is the total mass of the system and g the gravity acceleration amplitude. Note that a contact situation requires $\mathbf{F}_n \cdot \mathbf{z} > 0$, and thus the constraint

$$\forall t \in \mathbb{R}, \quad \ddot{z}\left(t\right) > -g \tag{4.2}$$

needs to be enforced.
The momentum equation at any point A writes, as already introduced[3] in equation (3.2)

$$\boldsymbol{\mathcal{M}}_A^c + \mathbf{AG} \times M\mathbf{g} = \mathbf{AG} \times M \begin{bmatrix} \ddot{\boldsymbol{c}}^T & \ddot{z} \end{bmatrix}^T + \dot{\mathbf{H}}_G, \tag{4.3}$$

[2] the 6 first lines of the equation (2.1), p. 23, related to the floating-base of the system
[3] *cf.* Sec. 3.1, p. 56

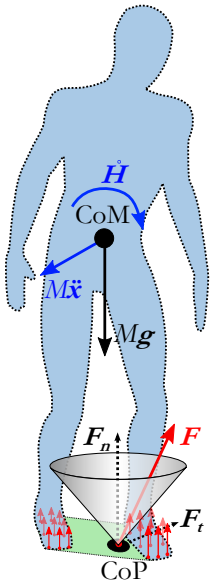

Figure 4.1.: Overview of postural stability protagonists. The dynamics of the system expressed at its center of mass suggest that postural stability is supported by the CoM 3D dynamics, angular momentum and external forces.

where $\dot{\mathbf{H}}_G$ is the rate of change of angular momentum of the system around its CoM and $\boldsymbol{\mathcal{M}}^c_A$ the moment of the contact forces around A.

Angular momentum of the robot around the CoM is assumed in this chapter to be supported by the upper body solely, with the torso as its root body of angular position $\boldsymbol{a}$ along the longitudinal and lateral axes. The validity of this hypothesis decreases with high accelerations of the system bodies. However, the moment of inertia of the torso is prevalent in humanoid structures, and its root position in the kinematics chain provides it with a central role in the upper body motion. The rate of change of the robot angular momentum is henceforth written $\dot{H}_{G_x} \triangleq \ddot{a}_x I_x$, where I_x is the upper body inertia around the robot CoM and along the x axis, with a similar expression along the y axis.

Assuming that all contacts are coplanar in plane $(\mathbf{x}, \mathbf{y})$, the center of pressure is the point on the ground of position $\mathbf{p}$ where $\boldsymbol{\mathcal{M}}^c_P \cdot \mathbf{x} = \boldsymbol{\mathcal{M}}^c_P \cdot \mathbf{y} = 0$, and hence holds

$$\begin{array}{rcl} (g + \ddot{z})\,(p_x - c_x) + z\ddot{c}_x - \ddot{a}_y I_y / M & = & 0 \\ (g + \ddot{z})\,(p_y - c_y) + z\ddot{c}_y - \ddot{a}_x I_x / M & = & 0. \end{array} \tag{4.4}$$

This last equation (4.4) along with equation (4.1) can be interpreted as the coupled outputs of three processes of independent respective states $(\boldsymbol{c}, \dot{\boldsymbol{c}})$, $(z, \dot{z})$ and $\left(H_{G_x}, \dot{H}_{G_x}\right)$ as illustrated in figure 4.2.

These output are protagonists in the postural stability problem: indeed, ZMP position $\mathbf{p}$ captures the contact forces distribution on the ground which is directly related to the tip-over risk, and the relative amplitude of contact forces determines the contact state and thus allow to apprehend slippage risk. Postural objectives are therefore defined in the following paragraphs with respect to these indicators.

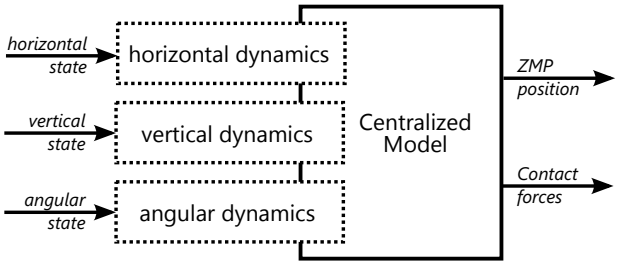

Figure 4.2.: Interpretation of the system dynamics as output-coupled processes. Independent horizontal, vertical and angular dynamics of the system contribute to the definition of stability outputs.

4.1.2. Tip-over risk

A common way to maximize biped balance is to maximize the distance between the CoP and the edges of the sustentation convex hull. This is usually performed by setting a predefined reference position $\mathbf{p}^r$ in the center of the contacts hull as a tracking objective for the ZMP $\mathbf{p}$, and hence maximizing postural balance is equivalent to minimizing a distance h^t between $\mathbf{p}$ and $\mathbf{p}^r$. The minimal tip-over risk situation $\mathbf{p} = \mathbf{p}^r$ can be written, assuming $(g + \ddot{z}) \neq 0$, as the balance objective

$$h^t \triangleq \left[\begin{array}{c} (g+\ddot{z})\,(p_x^r - c_x) + z\ddot{c}_x - \ddot{a}_y I_y/M \\ (g+\ddot{z})\left(p_y^r - c_y\right) + z\ddot{c}_y - \ddot{a}_x I_x/M \end{array} \right]^2 = 0. \tag{4.5}$$

A reduced model can be also deduced from equation (4.3) neglecting rotational effects $\ddot{\boldsymbol{a}}$, which brings the biped balance simplified objective

$$h^b \triangleq \left[\begin{array}{c} (g+\ddot{z})\,(p_x^r - c_x) + z\ddot{c}_x \\ (g+\ddot{z})\left(p_y^r - c_y\right) + z\ddot{c}_y \end{array} \right]^2 = 0. \tag{4.6}$$

A similar expression of the ZMP can be found in [Park2000] and is strictly equivalent, under the assumption $\forall t \in \mathbb{R},\ \dot{z}\,(t) = 0$, to the widespread expression of the ZMP objective [Kajita2003, Wieber2006, Herdt2010].

4.1.3. Slippage risk

Slippage risk is evaluated in this chapter through a non-Euclidean distance h^s to the Coulomb friction cone. A simple dry friction model considers two contact regimes:

- static friction: $F_t < \mu F_n$;
- kinetic friction (slippage): $F_t = \mu F_n$.

Note that in this section, the contact condition (4.2) is assumed to be respected and thus $F_n \neq 0$ holds. The contact margin m can therefore be defined, in the case of static friction, as a distance to the slippage regime $F_t = \mu F_n$

$$m \triangleq \frac{\mu^2 {F_n}^2}{\mu^2 {F_n}^2 - {F_t}^2}, \quad F_n \neq 0, \quad \mu^2 {F_n}^2 \neq {F_t}^2.$$

To maximize contacts stability, slippage regimes should be avoided: tangential efforts are not controllable in such cases. The distance m must hence be minimized. Minimizing m however would draw the solutions towards $F_t \to 0$ which would drastically hinder the motion of the system. Instead, the objective to be reached should rather draw the system *away* from the limit $F_t \to \mu F_n$; the definition of a logarithmic barrier over m is considered as a more relevant formulation in this problem. The objective is hence to minimize $h^s = \log(m)$, capturing a distance to the friction limit $F_t = \mu F_n$, written if $F_t < \mu F_n$ as

$$h^s \triangleq -\log\left(\mu^2 {F_n}^2 - {F_t}^2\right) + \log\left(\mu^2 {F_n}^2\right), \quad F_n \neq 0, \quad {F_n}^2 \neq {F_t}^2. \tag{4.7}$$

The distance h^s is a logarithmic frontier to the slippage regime: in a minimal slippage risk situation $F_t = 0 \Rightarrow h^s = 0$, and $h^s \to \infty$ when $F_t \to \mu F_n^-$.

4.2. Multi-objective centralized predictive problem

Postural balance is considered in the previous section as predominantly subject to tip-over and slippage risks. These risks are evaluated through cost functions to be minimized, being defined relatively to the CoM horizontal, vertical and angular dynamics. This section therefore introduces the formulation of a MPC problem exploiting these dynamics over a preview horizon to maximize postural balance.

4.2.1. Control parameters

In order to minimize tip-over and slippage risks over a preview horizon, the involved dynamics of the system must be previewed from the current state of the robot according to a future horizon of control parameters. These control parameters must thus allow to compute the involved dynamics and evaluate the cost functions; according to the formulations (4.5), (4.6) and (4.7), CoM linear acceleration and rate of angular momentum are suitable candidates as control parameters.
Similarly to [Kajita2003], these trajectories are captured by the definition of the jerks $\boldsymbol{u} \triangleq \dddot{\boldsymbol{c}}$, $v \triangleq \dddot{z}$ and $\boldsymbol{w} \triangleq \dddot{\boldsymbol{a}}$. In discrete time of period dt, dynamics $\hat{\boldsymbol{c}} \triangleq \begin{bmatrix} \boldsymbol{c} & \dot{\boldsymbol{c}} & \ddot{\boldsymbol{c}} \end{bmatrix}^T$, $\hat{\boldsymbol{z}}$ and $\hat{\boldsymbol{a}}$ can be deduced from a simple integration scheme

$$\forall i \in \mathbb{N}, \quad \begin{cases} \hat{\boldsymbol{c}}_{k+i+1|k} &= \mathbf{A}\hat{\boldsymbol{c}}_{k+i|k} &+ \mathbf{B}\boldsymbol{u}_{k+i|k}, \\ \hat{\boldsymbol{z}}_{k+i+1|k} &= \mathbf{A}_{\mathbf{z}}\hat{\boldsymbol{z}}_{k+i|k} &+ \mathbf{B}_{\mathbf{z}} v_{k+i|k}, \\ \hat{\boldsymbol{a}}_{k+i+1|k} &= \mathbf{A}\hat{\boldsymbol{a}}_{k+i|k} &+ \mathbf{B}\boldsymbol{w}_{k+i|k}. \end{cases}$$

Thus any function of $\hat{\boldsymbol{c}}$, $\hat{\boldsymbol{z}}$ or $\hat{\boldsymbol{a}}$ is written as a function of $\boldsymbol{u}$, v or $\boldsymbol{w}$, respectively. Matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{A}_{\mathbf{z}}$ and $\mathbf{B}_{\mathbf{z}}$ can be inferred from the formulations developed in Sec. 2.4.3 and Sec. 3.1.

4.2.2. Centralized optimization

The control problem solved in this chapter aims at maximizing postural stability over a preview horizon; that is, maximize biped balance and contacts stability.

Cost functions h^t, h^b and h^s involve different dynamics (vertical, horizontal and rotational) and capture these objectives at different levels: h^t measures a tip-over risk with a view to vertical, horizontal and rotational dynamics of the system, h^b relies on a simpler model of the system to quantify a balance error to a reference CoP position and h^s defines a contacts stability criterion as a slippage risk with respect to vertical and horizontal dynamics. This difference in insight provided by each cost function leads to the exploitation of different preview horizons n_t, n_b and n_s respectively. In discrete time of period dt, the postural stability maximization problem hence writes

$$\begin{cases} \min \underbrace{\omega_t \sum_{i=1}^{n_t} h^t_{k+i|k}}_{H^t_{n_t}} + \underbrace{\omega_b \sum_{i=1}^{n_b} h^b_{k+i|k}}_{H^b_{n_b}} + \underbrace{\omega_s \sum_{i=1}^{n_s} h^s_{k+i|k}}_{H^s_{n_s}}, \\ \text{s.t. equation (4.2)} \end{cases} \tag{4.8}$$

where ω_t, ω_b and ω_s are scalar coefficients defining the relative predominance of each individual objective and accounting for differences in dimensions. Horizons $\boldsymbol{\mathcal{U}}$, $\boldsymbol{\mathcal{V}}$ and $\boldsymbol{\mathcal{W}}$ of control inputs $\boldsymbol{u}$, v and $\boldsymbol{w}$ described in the previous paragraph are denoted at control step k

$$\begin{aligned} \boldsymbol{\mathcal{U}}^m_k &\triangleq \begin{bmatrix} \boldsymbol{u}_{k|k} & \dots & \boldsymbol{u}_{k+m-1|k} \end{bmatrix}^T, \\ \boldsymbol{\mathcal{V}}^m_k &\triangleq \begin{bmatrix} v_{k|k} & \dots & v_{k+m-1|k} \end{bmatrix}^T, \\ \boldsymbol{\mathcal{W}}^m_k &\triangleq \begin{bmatrix} \boldsymbol{w}_{k|k} & \dots & \boldsymbol{w}_{k+m-1|k} \end{bmatrix}^T. \end{aligned}$$

Let $N \triangleq \max(n_t, n_b, n_s)$; the optimization problem derived from the postural stability cost equation (4.8) writes

$$\begin{cases} \min\limits_{\boldsymbol{\mathcal{U}}^N_k, \boldsymbol{\mathcal{V}}^N_k, \boldsymbol{\mathcal{W}}^{n_t}_k} H^t_{n_t}\left(\boldsymbol{\mathcal{U}}^{n_t}_k, \boldsymbol{\mathcal{V}}^{n_t}_k, \boldsymbol{\mathcal{W}}^{n_t}_k\right) + H^b_{n_b}\left(\boldsymbol{\mathcal{U}}^{n_b}_k, \boldsymbol{\mathcal{V}}^{n_b}_k\right) + H^s_{n_s}\left(\boldsymbol{\mathcal{U}}^{n_s}_k, \boldsymbol{\mathcal{V}}^{n_s}_k\right) \\ \text{s.t. equation (4.2).} \end{cases} \tag{4.9}$$

Note that no straightforward convexity properties can be found in H^t, H^b and H^s. However an admissible initial guess can be derived from an unconstrained implementation of a ZMP Preview controller [Kajita2003, Wieber2006] for example and a local optimum within its neighborhood can be computed. The optimization problem is regularized with the introduction of additional strictly convex objective functions for each control parameter. A generalized Tikhonov regularization method is introduced with the following additional convex objective functions

$$\begin{aligned} H^u_n &\triangleq \omega_u \sum_{i=0}^{n-1} \|\boldsymbol{u}_{k+i|k}\|^2, & H^z_n &\triangleq \sum_{i=1}^{n} \|\boldsymbol{\Omega}_z(\hat{\boldsymbol{z}}_{k+i|k} - \hat{\boldsymbol{z}}^r)\|^2, \\ H^v_n &\triangleq \omega_v \sum_{i=0}^{n-1} v^2_{k+i|k} & \text{and} \quad H^w_n &\triangleq \sum_{i=1}^{n} \|\boldsymbol{\Omega}_w(\hat{\boldsymbol{a}}_{k+i|k} - \hat{\boldsymbol{a}}^r)\|^2, \end{aligned}$$

where ω_u and ω_v are scalar coefficients, $\boldsymbol{\Omega}_z$ and $\boldsymbol{\Omega}_w$ diagonal weighting matrices. Cost functions H^u and H^v can be interpreted as objective terms giving preference to smaller norm solutions. Additional costs H^z and H^w describe supplementary

tasks maintaining the CoM vertical position and torso orientation around reference states; note that desired reference states $\hat{\boldsymbol{z}}^r$ and $\hat{\boldsymbol{a}}^r$ are not necessarily constant and can be used to add a desired trajectory as an additional objective.

The postural stability control problem is written as the following regularized optimization problem

$$\begin{cases} \min\limits_{\boldsymbol{\mathcal{U}}_k^N, \boldsymbol{\mathcal{V}}_k^N, \boldsymbol{\mathcal{W}}_k^{n_t}} G_{n_t}^t\left(\boldsymbol{\mathcal{U}}_k^{n_t}, \boldsymbol{\mathcal{V}}_k^{n_t}, \boldsymbol{\mathcal{W}}_k^{n_t}\right) + G_{n_b}^b\left(\boldsymbol{\mathcal{U}}_k^{n_b}, \boldsymbol{\mathcal{V}}_k^{n_b}\right) + G_{n_s}^s\left(\boldsymbol{\mathcal{U}}_k^{n_s}, \boldsymbol{\mathcal{V}}_k^{n_s}\right) \\ \quad \text{s.t. equation (4.2)}, \end{cases} \tag{4.10}$$

where

$$\begin{aligned} G_{n_t}^t\left(\boldsymbol{\mathcal{U}}_k^{n_t}, \boldsymbol{\mathcal{V}}_k^{n_t}, \boldsymbol{\mathcal{W}}_k^{n_t}\right) &\triangleq H_{n_t}^t\left(\boldsymbol{\mathcal{U}}_k^{n_t}, \boldsymbol{\mathcal{V}}_k^{n_t}, \boldsymbol{\mathcal{W}}_k^{n_t}\right) + H_{n_t}^w\left(\boldsymbol{\mathcal{W}}_k^{n_t}\right), \\ G_{n_b}^b\left(\boldsymbol{\mathcal{U}}_k^{n_b}, \boldsymbol{\mathcal{V}}_k^{n_b}\right) &\triangleq H_{n_b}^b\left(\boldsymbol{\mathcal{U}}_k^{n_b}, \boldsymbol{\mathcal{V}}_k^{n_b}\right) + H_{n_b}^u\left(\boldsymbol{\mathcal{U}}_k^{n_b}\right) + H_{n_b}^z\left(\boldsymbol{\mathcal{V}}_k^{n_b}\right), \\ \text{and } G_{n_s}^s\left(\boldsymbol{\mathcal{U}}_k^{n_s}, \boldsymbol{\mathcal{V}}_k^{n_s}\right) &\triangleq H_{n_s}^s\left(\boldsymbol{\mathcal{U}}_k^{n_s}, \boldsymbol{\mathcal{V}}_k^{n_s}\right) + H_{n_s}^v\left(\boldsymbol{\mathcal{V}}_k^{n_s}\right) \end{aligned}$$

are the three regularized cost functions of problem equation (4.9).
The following section presents the distribution method proposed to solve such an optimization problem.

4.3. Resolution as a non-cooperative distributed problem

Concurrency between balance models is first distributed and conflicts between stability objectives are next exposed and relaxed through dual decomposition. The resulting DMPC problem is solved with an implementation of a parallelized projected gradient algorithm.

4.3.1. Sequential distribution

A sequential distribution is performed to uncouple the rotational variable $\boldsymbol{\mathcal{W}}_k$ from the linear ones. Problem equation (4.10) can be divided into two optimization problems, with $N \triangleq \max\left(n_b, n_s\right)$

$$\begin{cases} \begin{cases} \min\limits_{\boldsymbol{\mathcal{U}}_k^N, \boldsymbol{\mathcal{V}}_k^N} G_{n_b}^b\left(\boldsymbol{\mathcal{U}}_k^{n_b}, \boldsymbol{\mathcal{V}}_k^{n_b}\right) + G_{n_s}^s\left(\boldsymbol{\mathcal{U}}_k^{n_s}, \boldsymbol{\mathcal{V}}_k^{n_s}\right) \\ \quad \text{s.t. equation (4.2)}, \end{cases} & \text{(4.11a)} \\ \begin{cases} \min\limits_{\boldsymbol{\mathcal{U}}_k^{n_t}, \boldsymbol{\mathcal{V}}_k^{n_t}, \boldsymbol{\mathcal{W}}_k^{n_t}} G_{n_t}^t\left(\boldsymbol{\mathcal{U}}_k^{n_t}, \boldsymbol{\mathcal{V}}_k^{n_t}, \boldsymbol{\mathcal{W}}_k^{n_t}\right) \\ \quad \text{s.t. equation (4.2)} \end{cases} & \text{(4.11b)} \end{cases}$$

which are coupled by the variables $\boldsymbol{\mathcal{U}}_k$ and $\boldsymbol{\mathcal{V}}_k$. The distribution adopted consists in solving the problem (4.11b) after (4.11a): $\boldsymbol{\mathcal{U}}_k$ and $\boldsymbol{\mathcal{V}}_k$ are considered as inputs to the problem (4.11b). The centralized problem (4.10) is hence distributed as

follows

$$\begin{cases} \begin{cases} \text{Find } \boldsymbol{\mathcal{U}}_k^{N*}, \boldsymbol{\mathcal{V}}_k^{N*} \text{ that solve} \\ \min\limits_{\boldsymbol{\mathcal{U}}_k^N, \boldsymbol{\mathcal{V}}_k^N} G_{n_b}^b\left(\boldsymbol{\mathcal{U}}_k^{n_b}, \boldsymbol{\mathcal{V}}_k^{n_b}\right) + G_{n_s}^s\left(\boldsymbol{\mathcal{U}}_k^{n_s}, \boldsymbol{\mathcal{V}}_k^{n_s}\right) \\ \quad \text{s.t. equation (4.2)}, \end{cases} & \text{(4.12a)} \\ \begin{cases} \text{Find } \boldsymbol{\mathcal{W}}_k^{n_t *} \text{ that solves} \\ \min\limits_{\boldsymbol{\mathcal{W}}_k^{n_t}} G_{n_t}^t\left(\boldsymbol{\mathcal{U}}_k^{n_t *}, \boldsymbol{\mathcal{V}}_k^{n_t *}, \boldsymbol{\mathcal{W}}_k^{n_t}\right) \\ \quad \text{s.t. equation (4.2)} \end{cases} & \text{(4.12b)} \end{cases}$$

which is not strictly equivalent. Nevertheless such a distribution can be interpreted as the consideration of angular momentum as a non-conflicting means to increase the balance performance: optimal CoM linear dynamics are first computed unaware of rotational effects through equation (4.12a), and the torso angular dynamics are subsequently adjusted according to equation (4.12b) in order to maximize the postural stability of the system. This sequential distribution (4.12) of the centralized problem (4.10) is adopted in this chapter to reduce the influence of H^t over $\boldsymbol{\mathcal{U}}_k$ and $\boldsymbol{\mathcal{V}}_k$ to a unilateral pairing, due to the error-prone approximation[4] made on the angular momentum.

The sequential distribution of the centralized problem (4.10) is depicted in figure 4.3. A centralized sub-problem (4.12a) is still present in the resulting architecture, accounting for the coupling between predictive primitives based on the objectives G^b and G^s which capture tip-over and slippage risks respectively.

4.3.2. Dual decomposition

The coupling between balance and contact stability objectives G^b and G^s of problem equation (4.12a) cannot be handled safely through a straightforward sequential distribution as introduced in the previous paragraph. Their concurrency indeed induces that any change in $\boldsymbol{\mathcal{U}}_k$ or $\boldsymbol{\mathcal{V}}_k$ that would benefit G^b might deteriorate the distance G^s which is optimized on the same variables: such problems can be referred to as cooperative — or team — problems. While a centralized approach is the most natural approach to solving this class of problems, a modularity requirement for the control architecture would favor a distributed structure of the problem which in turn may reduce the computational cost of the predictive problem in some cases.

In this perspective, dual decomposition allows to set up a market mechanism where G^b and G^s are locally optimized while both problems are driven to a consensus. The resulting strictly equivalent problem is of a simpler structure — non-cooperative — and thus more favorable to common solving algorithms.

[4] angular momentum of the system is assumed to be supported by the upper bodies solely through the angular motion of the torso

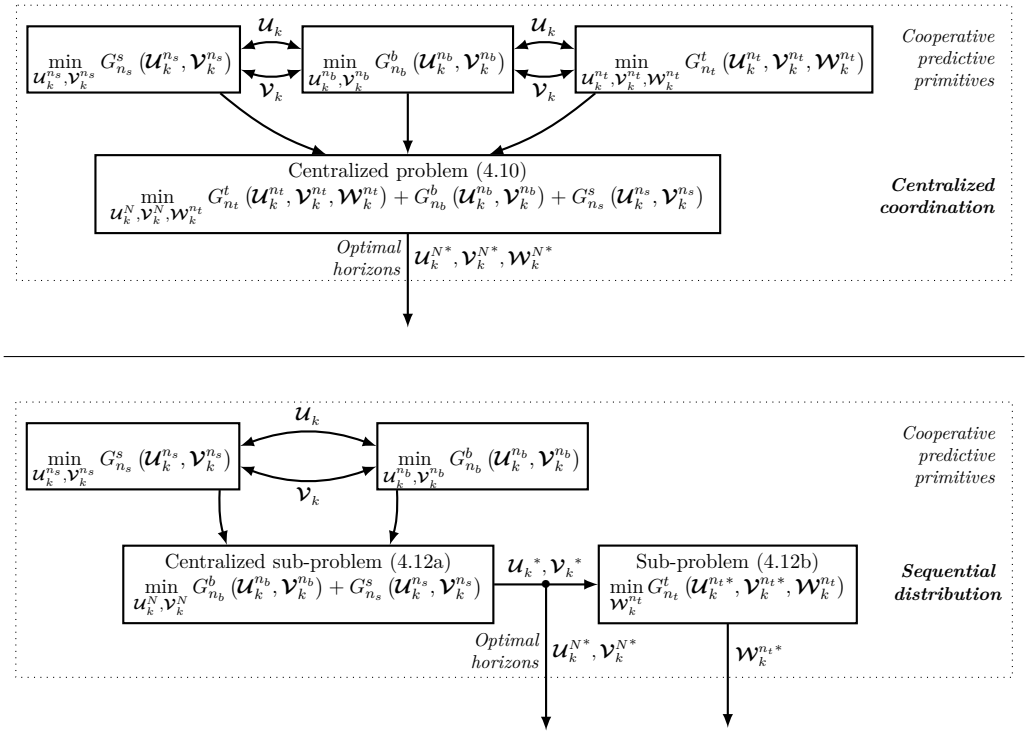

Figure 4.3.: Illustration of the sequential distribution performed on the centralized predictive problem (4.10). One of the two couplings between predictive primitives is assumed to be unilateral, leading to the sequential resolution of sub-problems (4.12a) and (4.12b).

Problem equation (4.12a) can be written as the following constrained problem by the introduction of local versions $(\boldsymbol{\Phi}, \boldsymbol{\Theta})$ of the coupling variables $(\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{V}})$

$$\left\{ \begin{array}{l} \min\limits_{\substack{\boldsymbol{\mathcal{U}}_k^{n_b}, \boldsymbol{\mathcal{V}}_k^{n_s}, \\ \boldsymbol{\Phi}_k^{n_s}, \boldsymbol{\Theta}_k^{n_b}}} \; G_{n_b}^b \left(\boldsymbol{\mathcal{U}}_k^{n_b}, \boldsymbol{\Theta}_k^{n_b}\right) + G_{n_s}^s \left(\boldsymbol{\Phi}_k^{n_s}, \boldsymbol{\mathcal{V}}_k^{n_s}\right) \\ \qquad \text{s.t.} \left\{ \begin{array}{l} (4.2), \\ \boldsymbol{\Phi}_k^n - \boldsymbol{\mathcal{U}}_k^n = \mathbf{0}, \\ \boldsymbol{\Theta}_k^n - \boldsymbol{\mathcal{V}}_k^n = \mathbf{0} \end{array} \right. \end{array} \right. \tag{4.13}$$

where $n \triangleq \min(n_b, n_s)$. The Lagrangian $\mathcal{L}$ of this problem writes, with the Lagrange vector multipliers $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$

$$\mathcal{L} \triangleq G_{n_b}^b \left(\boldsymbol{\mathcal{U}}_k^{n_b}, \boldsymbol{\Theta}_k^{n_b}\right) + G_{n_s}^s \left(\boldsymbol{\Phi}_k^{n_s}, \boldsymbol{\mathcal{V}}_k^{n_s}\right) + \boldsymbol{\phi}_k^{nT} \left[\boldsymbol{\Phi}_k^n - \boldsymbol{\mathcal{U}}_k^n\right] + \boldsymbol{\theta}_k^{nT} \left[\boldsymbol{\Theta}_k^n - \boldsymbol{\mathcal{V}}_k^n\right]$$

and problem (4.13) — problem (4.12a) consequently — is strictly equivalent to

$$\left\{ \begin{array}{l} \max\limits_{\boldsymbol{\phi}_k^n, \boldsymbol{\theta}_k^n} \; \min\limits_{\substack{\boldsymbol{\mathcal{U}}_k^{n_b}, \boldsymbol{\mathcal{V}}_k^{n_s}, \\ \boldsymbol{\Phi}_k^{n_s}, \boldsymbol{\Theta}_k^{n_b}}} \; \mathcal{L}\left(\boldsymbol{\mathcal{U}}_k^{n_b}, \boldsymbol{\mathcal{V}}_k^{n_s}, \boldsymbol{\Phi}_k^{n_s}, \boldsymbol{\Theta}_k^{n_b}, \boldsymbol{\phi}_k^n, \boldsymbol{\theta}_k^n\right) \\ \qquad \text{s.t. } (4.2). \end{array} \right.$$

This problem is separable, and can be dissociated into four subproblems

$$\begin{cases} \min\limits_{\boldsymbol{\mathcal{U}}_k^{n_b},\boldsymbol{\Theta}_k^{n_b}} G_{n_b}^b\left(\boldsymbol{\mathcal{U}}_k^{n_b},\boldsymbol{\Theta}_k^{n_b}\right) + \boldsymbol{\theta}_k^{nT}\boldsymbol{\Theta}_k^n - \boldsymbol{\phi}_k^{nT}\boldsymbol{\mathcal{U}}_k^n & (4.14a) \\ \min\limits_{\boldsymbol{\Phi}_k^{n_s},\boldsymbol{\mathcal{V}}_k^{n_s}} G_{n_s}^s\left(\boldsymbol{\Phi}_k^{n_s},\boldsymbol{\mathcal{V}}_k^{n_s}\right) + \boldsymbol{\phi}_k^{nT}\boldsymbol{\Phi}_k^n - \boldsymbol{\theta}_k^{nT}\boldsymbol{\mathcal{V}}_k^n & (4.14b) \\ \max\limits_{\boldsymbol{\phi}_k^n} \boldsymbol{\phi}_k^{nT}\left[\boldsymbol{\Phi}_k^n - \boldsymbol{\mathcal{U}}_k^n\right] & (4.14c) \\ \max\limits_{\boldsymbol{\theta}_k^n} \boldsymbol{\theta}_k^{nT}\left[\boldsymbol{\Theta}_k^n - \boldsymbol{\mathcal{V}}_k^n\right] & (4.14d) \\ \text{s.t.} \quad (4.2), \end{cases}$$

which is rewritten for the sake of simplicity

$$\begin{cases} \min\limits_{\boldsymbol{\mathcal{U}}_k^{n_s},\boldsymbol{\Theta}_k^{n_b}} J^b\left(\boldsymbol{\mathcal{U}}_k^{n_b},\boldsymbol{\Theta}_k^{n_b},\boldsymbol{\theta}_k^n,\boldsymbol{\phi}_k^n\right), \quad \text{s.t.} \quad (4.2) & (4.15a) \\ \min\limits_{\boldsymbol{\Phi}_k^{n_s},\boldsymbol{\mathcal{V}}_k^{n_s}} J^s\left(\boldsymbol{\Phi}_k^{n_s},\boldsymbol{\mathcal{V}}_k^{n_s},\boldsymbol{\phi}_k^n,\boldsymbol{\theta}_k^n\right), \quad \text{s.t.} \quad (4.2) & (4.15b) \\ \min\limits_{\boldsymbol{\phi}_k^n} J^\phi\left(\boldsymbol{\phi}_k^n,\boldsymbol{\Phi}_k^n,\boldsymbol{\mathcal{U}}_k^n\right) & (4.15c) \\ \min\limits_{\boldsymbol{\theta}_k^n} J^\theta\left(\boldsymbol{\theta}_k^n,\boldsymbol{\Theta}_k^n,\boldsymbol{\mathcal{V}}_k^n\right), & (4.15d) \end{cases}$$

where

$$\begin{aligned} J^\phi\left(\boldsymbol{\phi}_k^n,\boldsymbol{\Phi}_k^n,\boldsymbol{\mathcal{U}}_k^n\right) &= \boldsymbol{\phi}_k^{nT}\left[\boldsymbol{\mathcal{U}}_k^n - \boldsymbol{\Phi}_k^n\right], \\ J^\theta\left(\boldsymbol{\theta}_k^n,\boldsymbol{\Theta}_k^n,\boldsymbol{\mathcal{V}}_k^n\right) &= \boldsymbol{\theta}_k^{nT}\left[\boldsymbol{\mathcal{V}}_k^n - \boldsymbol{\Theta}_k^n\right]. \end{aligned}$$

The set of problems (4.15) can be interpreted as a market mechanism where $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ regulate the cost of violating the coupling between the objectives G^b and G^s. Such a distributed architecture represents a non-cooperative parallel problem: each subproblem optimizes a local cost function with its own variables. Communication between them is sufficient to reach a Nash equilibrium of problem (4.15) which is strictly equivalent to the desired Pareto optimum of the centralized problem (4.12a).

4.3.3. Discussion

The distributed form (4.15) is depicted in figure 4.4, exhibiting the non-cooperative, communicating flow of information between the four local sub-problems.

Despite its apparent complexity, the resulting distributed architecture presents better modularity characteristics than the centralized formulation. Indeed, the explicit formulation of couplings between the primitives G^s and G^b through the definition of cost functions J^ϕ and J^θ allows to naturally manipulate the structure of the problem. More precisely, solely changing one of the predictive primitives and the communication flow allows to result in a partially sequential distribution of the problem.
For example, if J^s is assumed to have no influence on the CoM horizontal dynamics $\boldsymbol{\mathcal{U}}_k$ to let J^b specify its optimal horizon, the sole primitive (4.14b) related to J^s is to be modified accordingly. As illustrated in figure 4.5, a partially sequential distribution can be naturally obtained through the modification of the J^s primitive (4.14b) following the assumption, and of minor communication bypasses to

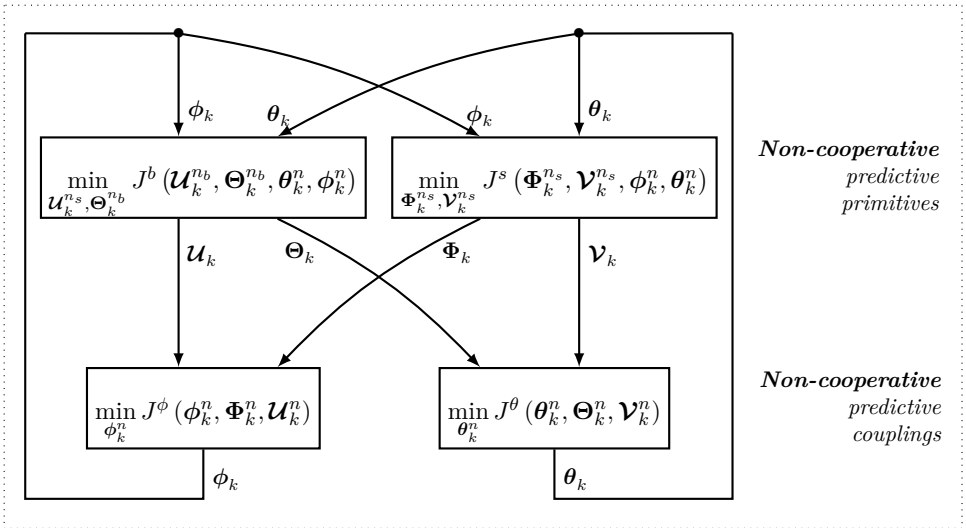

Figure 4.4.: Illustration of the equivalent non-cooperative distribution (4.15) performed on the centralized predictive sub-problem (4.12a). Predictive couplings are introduced to formulate the cooperative sub-problem into a communicating distributed problem.

account for the resulting disregard to the coupling (4.14c) between J^b and J^s with respect to $\boldsymbol{\mathcal{U}}_k$.

This explicit exposure of the couplings thus allows to enforce hierarchies between predictive primitives without changing the overall decomposition of the centralized problem. The resulting modularity is particularly interesting in the case of complex multi-objective activities, where priorities can evolve with respect to the task conditions.

It can be observed that, despite the decomposition, the maximum dimension of the distributed sub-problems (4.15) is the same as the equivalent centralized problem (4.12a). The computational advantages of the distributed approach is therefore unclear in this case.

However, a different distribution of the problem can be envisaged: instead of exposing the coupling between G^s and G^b with respect to $\boldsymbol{\mathcal{U}}_k$ and $\boldsymbol{\mathcal{V}}_k$ as performed in (4.13), the preview horizon of primitives G^s and G^b can be split into distinct parts with continuity constraints. These constraints are then formulated as maximization sub-problems using Lagrange multipliers, resulting in a new dual decomposition of problem (4.12a). In the case of a preview horizon split into two equal parts, the maximum dimension of the distributed sub-problems is thus reduced by half from the original centralized dimension.
A distribution relative to the predictive primitives is nonetheless preferred in this chapter to exhibit the contribution of the distributed approach to the coordination of primitives.
The distribution separating objective primitives can moreover be employed to exploit their particular forms, in cases where G^s and G^b present better individual convergence properties than the centralized function $G^s + G^b$ for example.

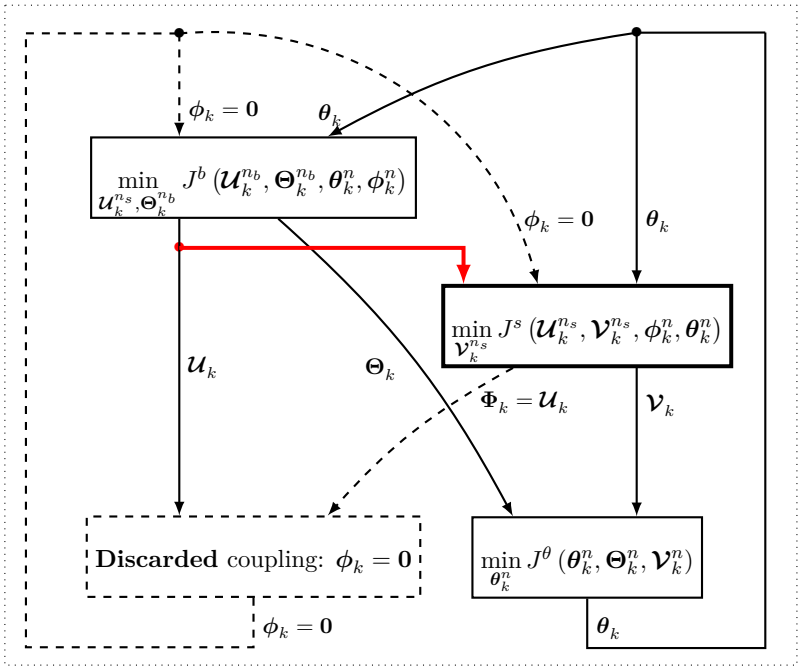

Figure 4.5.: Example of a partially sequential distribution resulting from the modification of the communication flow (*dashed and thick lines*) and of a single predictive primitive (*thick block*). The modularity of the non-cooperative DMPC problem (4.15) allows to naturally manipulate the couplings between predictive primitives.

Furthermore, a combination of these two distribution approaches, *i.e.* over the preview horizon or over the objective primitives, can be envisioned.

4.3.4. Algorithm

Problems (4.14a) and (4.14b) present no peculiar convexity properties; however an admissible local optimum around an initial guess is considered as satisfactory in this development. The linear constraint (4.2) and the previous remarks license the setup of a projected gradient algorithm to solve problem (4.15).
The parallelized algorithm described in Algorithm 1 is henceforth proposed, with the projector $\boldsymbol{\Pi}_g^m : \mathbb{R}^m \to \mathbb{R}^m$ on the variety delimited by equation (4.2) over m preview steps. Descent steps α are determined using a backtracking line search method.
This algorithm is naturally decomposed into four processes. As previously stated, the distributed problem (4.15) is non-cooperative: the four processes are solely communicating the updated outputs of their respective predictive sub-problems at each iteration of the algorithm. This structure enables the natural distribution of the gradient computations and backtracking methods to distinct, communicating computation threads. The overall computational cost of the multi-objective MPC problem is therefore potentially reduced significantly through its distribution, as discussed in Sec. 4.3.3; however parallel computing raises several challenges.

Algorithm 1 Parallel projected gradient algorithm for the DMPC problem (4.15)

Require: $\epsilon, \boldsymbol{\mathcal{U}}^{N}_{k\ 0}, \boldsymbol{\mathcal{V}}^{N}_{k\ 0}, \boldsymbol{\Phi}^{n_s}_{k\ 0}, \boldsymbol{\Theta}^{n_b}_{k\ 0}, \boldsymbol{\phi}^{n}_{k0}, \boldsymbol{\theta}^{n}_{k0}$;

$i := 0$;

while $\|\boldsymbol{\nabla}_{\boldsymbol{\mathcal{U}},\boldsymbol{\Theta}} J^b\| + \|\boldsymbol{\nabla}_{\boldsymbol{\Phi},\boldsymbol{\mathcal{V}}} J^s\| + \|\boldsymbol{\nabla}_{\boldsymbol{\phi}} J^{\phi}\| + \|\boldsymbol{\nabla}_{\boldsymbol{\theta}} J^{\theta}\| > \epsilon$ **do**

*/** α is computed with a backtracking line search **/*

*/** Process 1: J^b **/*

$\mathbf{d}_{\boldsymbol{\mathcal{U}}} := \boldsymbol{\nabla}_{\boldsymbol{\mathcal{U}}} J^b_i$; $\quad \boldsymbol{\mathcal{U}}^{n_b}_{k\ i+1} := \boldsymbol{\Pi}^{n_b}_{g} \left(\boldsymbol{\mathcal{U}}^{n_b}_{k\ i} - \alpha_{\boldsymbol{\mathcal{U}}} \mathbf{d}_{\boldsymbol{\mathcal{U}}}\right)$;

$\mathbf{d}_{\boldsymbol{\Theta}} := \boldsymbol{\nabla}_{\boldsymbol{\Theta}} J^b_i$; $\quad \boldsymbol{\Theta}^{n_b}_{k\ i+1} := \boldsymbol{\Pi}^{n_b}_{g} \left(\boldsymbol{\Theta}^{n_b}_{k\ i} - \alpha_{\boldsymbol{\Theta}} \mathbf{d}_{\boldsymbol{\Theta}}\right)$;

*/** Process 2: J^s **/*

$\mathbf{d}_{\boldsymbol{\Phi}} := \boldsymbol{\nabla}_{\boldsymbol{\Phi}} J^s_i$; $\quad \boldsymbol{\Phi}^{n_s}_{k\ i+1} := \boldsymbol{\Pi}^{n_s}_{g} \left(\boldsymbol{\Phi}^{n_s}_{k\ i} - \alpha_{\boldsymbol{\Phi}} \mathbf{d}_{\boldsymbol{\Phi}}\right)$;

$\mathbf{d}_{\boldsymbol{\mathcal{V}}} := \boldsymbol{\nabla}_{\boldsymbol{\mathcal{V}}} J^s_i$; $\quad \boldsymbol{\mathcal{V}}^{n_s}_{k\ i+1} := \boldsymbol{\Pi}^{n_s}_{g} \left(\boldsymbol{\mathcal{V}}^{n_s}_{k\ i} - \alpha_{\boldsymbol{\mathcal{V}}} \mathbf{d}_{\boldsymbol{\mathcal{V}}}\right)$;

*/** Process 3: J^{ϕ} **/*

$\boldsymbol{\phi}^{n}_{k\,i+1} := \boldsymbol{\phi}^{n}_{k\,i} - \alpha_{\boldsymbol{\phi}} \left[\boldsymbol{\mathcal{U}}^{n}_{k\,i} - \boldsymbol{\Phi}^{n}_{k\,i}\right]$;

*/** Process 4: J^{θ} **/*

$\boldsymbol{\theta}^{n}_{k\,i+1} := \boldsymbol{\theta}^{n}_{k\,i} - \alpha_{\boldsymbol{\theta}} \left[\boldsymbol{\mathcal{V}}^{n}_{k\,i} - \boldsymbol{\Theta}^{n}_{k\,i}\right]$;

$i := i + 1$;

end while

return $\boldsymbol{\mathcal{U}}^{n*}_{k} = \boldsymbol{\Pi}^{n}_{g} \left(\frac{\boldsymbol{\mathcal{U}}^{n}_{k\,i} + \boldsymbol{\Phi}^{n}_{k\,i}}{2}\right), \quad \boldsymbol{\mathcal{V}}^{n*}_{k} = \boldsymbol{\Pi}^{n}_{g} \left(\frac{\boldsymbol{\mathcal{V}}^{n}_{k\,i} + \boldsymbol{\Theta}^{n}_{k\,i}}{2}\right)$;

Typical convergence results for objective functions J^b and J^s with this algorithm are shown in figure 4.6: concurrency is clearly visible as both objectives successively and alternately compromise until they reach a mutual agreement.

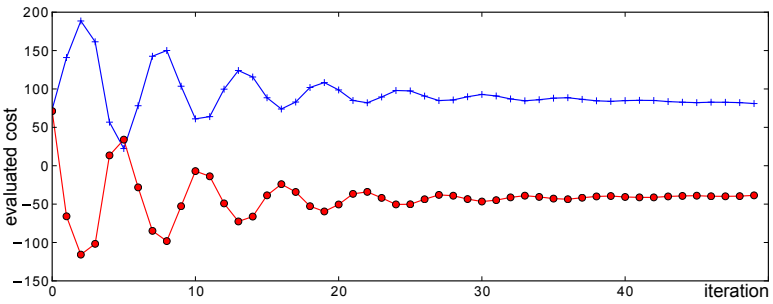

Figure 4.6.: Typical convergence results for conflicting objective functions J^b and J^s across the parallel resolution. These cost functions capture the slippage and tip-over risks respectively, and a compromise is successively alternated until convergence. Conflicts between the two objective are depicted with the simultaneous increase and decrease of the cost functions. *Blue cross: J^b - red circle: J^s.*

The gap between decoupled variables $(\boldsymbol{\mathcal{U}}, \boldsymbol{\Phi})$ and $(\boldsymbol{\mathcal{V}}, \boldsymbol{\Theta})$ is simultaneously reduced and converges towards a consensus for the values of $(\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{V}})$, as shown in figure 4.7. This figure shows that convergence of the pair $(\boldsymbol{\mathcal{V}}, \boldsymbol{\Theta})$ is slower than $(\boldsymbol{\mathcal{U}}, \boldsymbol{\Phi})$; this can be interpreted as vertical dynamics being the preponderant source of conflict between balance and friction objectives. Therefore, and accordingly to the remarks of Sec. 4.3.3, a sequential distribution of one of the two primitives J^s and J^b can be envisioned with respect to the horizontal dynamics $\boldsymbol{\mathcal{U}}$,

as it appears to be a secondary source of conflicts between these primitives.

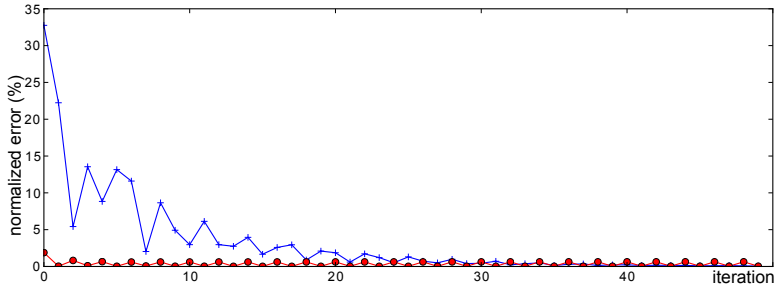

Figure 4.7.: Typical convergence of constraint relaxations $\boldsymbol{\mathcal{U}} = \boldsymbol{\Phi}$ and $\boldsymbol{\mathcal{V}} = \boldsymbol{\Theta}$ towards a consensus. These indicators capture the conflicts between slippage and tip-over objectives: their temporary optimal variables are progressively drawn closer until a consensus is reached. *Blue cross:* $\frac{\|\boldsymbol{\Theta}_k^n - \boldsymbol{\mathcal{V}}_k^n\|}{\|\boldsymbol{\mathcal{V}}_k^n\|}$ - *red circle:* $\frac{\|\boldsymbol{\Phi}_k^n - \boldsymbol{\mathcal{U}}_k^n\|}{\|\boldsymbol{\mathcal{U}}_k^n\|}$.

It is worth noticing that a suboptimal solution can be obtained by stopping the algorithm before convergence is reached: an admissible solution with respects to the coupling constraints is obtained by projections $\boldsymbol{\mathcal{U}}_k^{nadm} = \frac{\boldsymbol{\mathcal{U}}_k^n + \boldsymbol{\Phi}_k^n}{2}$ and $\boldsymbol{\mathcal{V}}_k^{nadm} = \frac{\boldsymbol{\mathcal{V}}_k^n + \boldsymbol{\Theta}_k^n}{2}$ onto the feasible sets $\boldsymbol{\mathcal{U}} = \boldsymbol{\Phi}$ and $\boldsymbol{\mathcal{V}} = \boldsymbol{\Theta}$.
Convergence results presented in figures 4.6 and 4.7 are obtained from the simulations presented in Sec. 4.4: less than 30 descent iterations were required to reach convergence.

Figure 4.8 depicts this parallelized algorithm within the sequential distribution (4.12). The distributed structure of the problem is distinctly highlighted, and coordination of the conflicting predictive primitives is obtained through communicating iterations.

4.4. Results

Simulations have been carried out using Arboris-Python [BarthelemyArboris], an open-source dynamics simulator developed at ISIR with the Python programming language. An accurate model of an iCub robot [Sandini2007] is simulated with $32 + 6$ degrees of freedom.

4.4.1. Compared frameworks

Instantaneous whole-body control of the model is ensured by an LQP-based controller [Salini2010] accounting for joint and contact constraints. Contacts between the robot and the ground are solved according to a dry friction model of static coefficient $\mu = 0.5$. The whole-body controller takes as inputs desired trajectories of the CoM position and torso orientation, and the performance of the DMPC algorithm introduced in this work is compared to the one of a state of the art predictive framework.

A low-priority task is defined for both frameworks to try and maintain all joints position around a predefined reference describing a standing pose of the

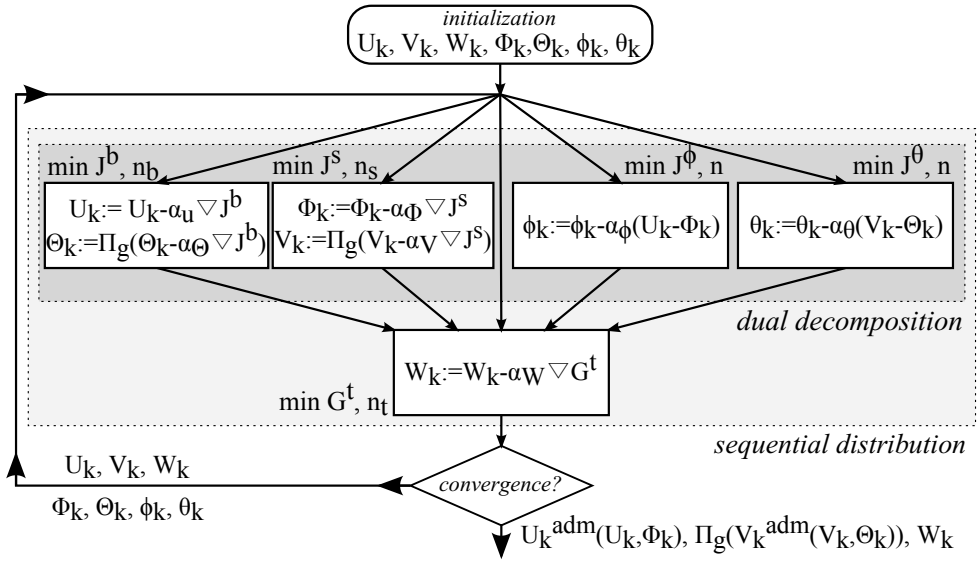

Figure 4.8.: Flow chart of the iterative gradient descent algorithm solving the non-cooperative and sequential distributed problems (4.15) and (4.12). This algorithm is inherently parallelized and allows simultaneous resolution operations for four sub-problems. Coordination is achieved through communicating iterations.

robot. Trajectory tracking tasks for both the CoM position and the torso orientation are included in the whole-body control problem as top-level priority tasks (weight a hundred times greater). The predictive framework is centered around an unconstrained implementation of the ZMP Preview Control as described in [Wieber2006] which provides optimal horizontal trajectories for the CoM of the robot. Orientation of the torso is indirectly maintained by the low priority regularizing task, specifying a reference joint configuration of the robot, while CoM altitude is regulated with a stiff proportional-derivative controller in order to enforce the constant-altitude assumption made in the ZMP Preview Control formulation [Kajita2003].
In the second framework all CoM and torso desired trajectories are output from the DMPC algorithm introduced in this chapter, as shown in figure 4.9.

4.4.2. Simulation results

The scenario supporting the simulation is a walking activity under an unknown external action from the environment on the waist of the robot, as illustrated with a selection of simulation snapshots in figure 4.10.
The profile of the external action is shown in figure 4.11 which exhibits two impulses in the horizontal plane shortly preceding the changes of supporting foot.

To allow for comparison, regularization of H^b is set up to the same order of magnitude as in the ZMP Preview Control problem, that is $\omega_u/\omega_b = 1.e^{-6}$. As the cost function G^b is the closest in insight — among G^b, G^t and G^s — to the objective function minimized in the ZMP Preview Control framework, its horizon n_b is set similarly to the ZMP Preview control problem and covers $1.0s$. The

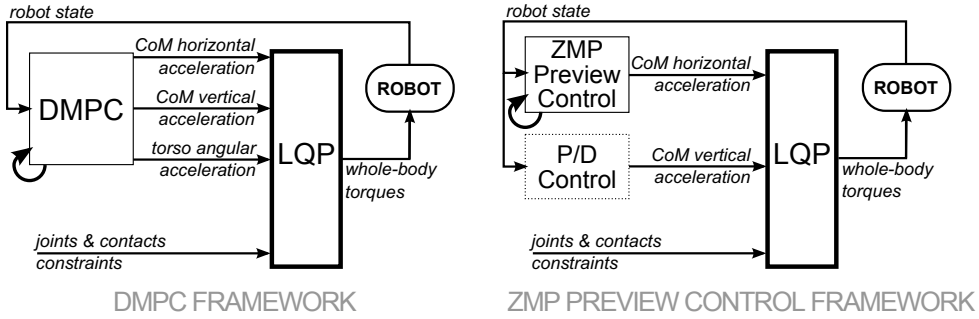

Figure 4.9.: Predictive control frameworks compared in the disturbed task-posture scenario. The DMPC controller of this chapter is compared to a ZMP Preview Control method coordinated at the whole-body level instantaneously with an LQP-based controller.

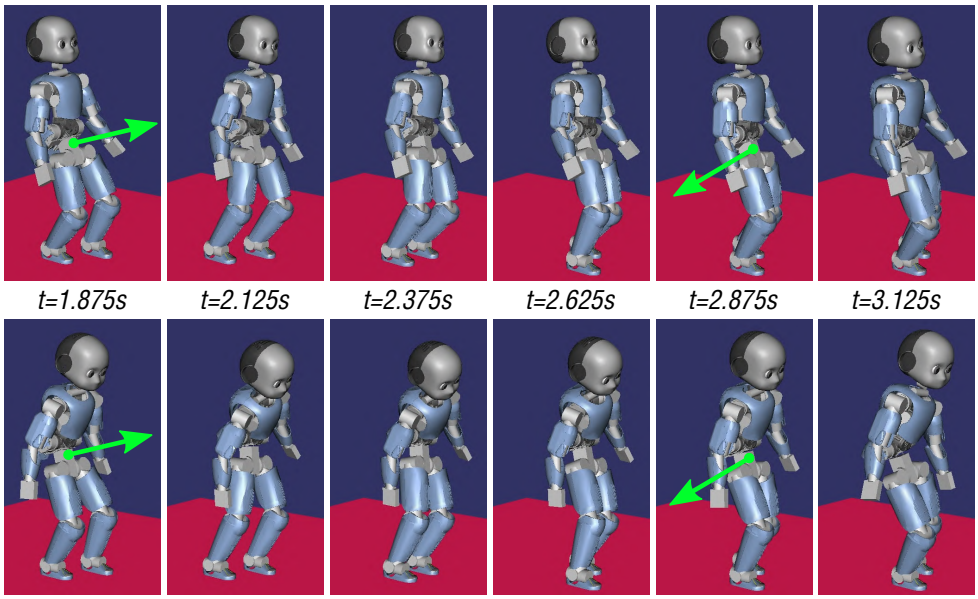

Figure 4.10.: Simulation snapshots for both control frameworks. The vertical and rotational adjustments of the DMPC controller allows to concurrently minimize tip-over and slippage risks. *Top: ZMP Preview control – bottom: DMPC.*

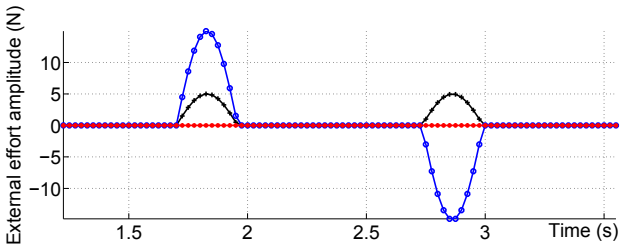

Figure 4.11.: Amplitude profile of the unknown external action applied on the waist of the robot. Smooth impulses are applied to the waist of the robot in the lateral and longitudinal directions. *Blue circle: lateral – black cross: longitudinal – red star: vertical.*

selected values for preview horizons n_t and n_s both span $0.25s$.

Different values of n_t and n_s from $n_b/8$ to n_b resulted in a lower performance of the controller during critical phases. This remark validates the need for different preview horizons depending on the level of insight provided by each model and control objective, and the predominant dynamics they involve. This result furthermore raises the issues of model reduction and horizon length with MPC approaches. While the level of insight of the model is generally restrained by computational constraints[5], the length of the preview horizon employed in the MPC problem influences the form of the solution, depending on the dynamics the reduced model captures. Moreover, as the time discretization of the horizon is also constrained by stability requirements, the preview length directly affects the dimension of the optimization problem.
The setup of MPC approaches therefore requires a tuning phase of the preview horizon to compromise between computational complexity, stability and performance.

Nevertheless, and as demonstrated in the formulation developed in this chapter with the balance objectives G^b and G^t, the predictive coordination of models of various accuracy with distinct preview horizons can be employed to take advantage of multiple formulations of a single predictive task. Approximations on the coupling between models of different complexity can be performed to alleviate the computational cost of finer models, as it is achieved in this chapter with the sequential distribution of the more detailed balance primitive G^t with respect to the coarser balance model employed in G^b.

Figures 4.12 and 4.13 show the evolution of the CoP position on the ground for both frameworks, with regard to the tracked reference position.
These figures show that, as expected, the concomitant use of both horizontal and vertical degrees of freedom of the CoM along with the control of angular momentum allows for a significant gain in balance performance. This improvement in reference tracking can reach up to more than 50% during critical intervals in the lateral direction, as shown in figure 4.13 at instants $t \approx 2.0s$ and $t \approx 3.0s$ when the ZMP Preview Control framework almost fails at maintaining the robot postural balance.

An additional benefit from the introduction of the vertical control of the CoM is displayed in figure 4.14 which maps the amplitude ratio ν between tangential and normal components of the resulting contact force with the ground.
This ratio can be interpreted as a required friction coefficient: if the actual friction coefficient μ, characterizing the contact conditions with the ground, is lower than or equal to ν, then slippage occurs and might lead to postural instability. In that sense, the lowest required friction coefficient minimizes the risk of slippage in case of ill-estimation of the current contact conditions. Actual friction coefficient μ might be overestimated in a variety of usual cases in unstructured environments: strong drops in admissible traction can be observed for example if the robot steps over a wet area or an unexpected region of different material. Fig-

[5] linear models are for example preferred to formulate quadratic convex objective functions

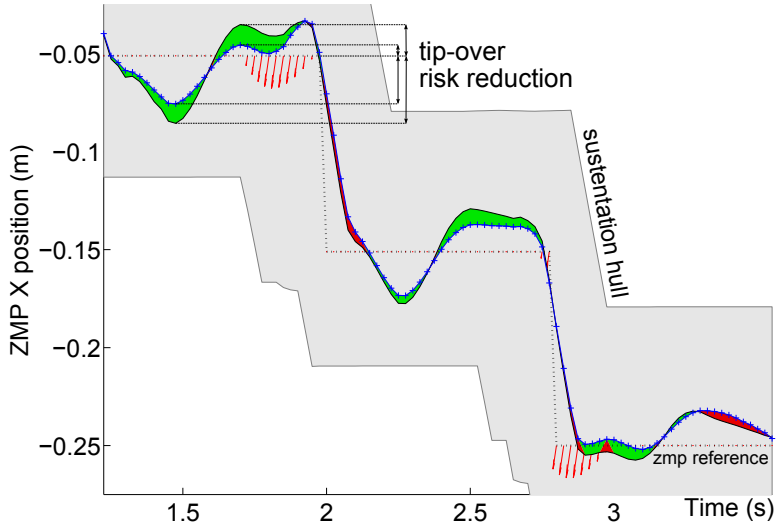

Figure 4.12.: Evolution of the ZMP in longitudinal direction. A decrease in tip-over risk from the DMPC controller is observable through a better tracking of the ZMP. *Blue cross: DMPC – black solid: ZMP Preview Control – red arrows: normalized external effort.*

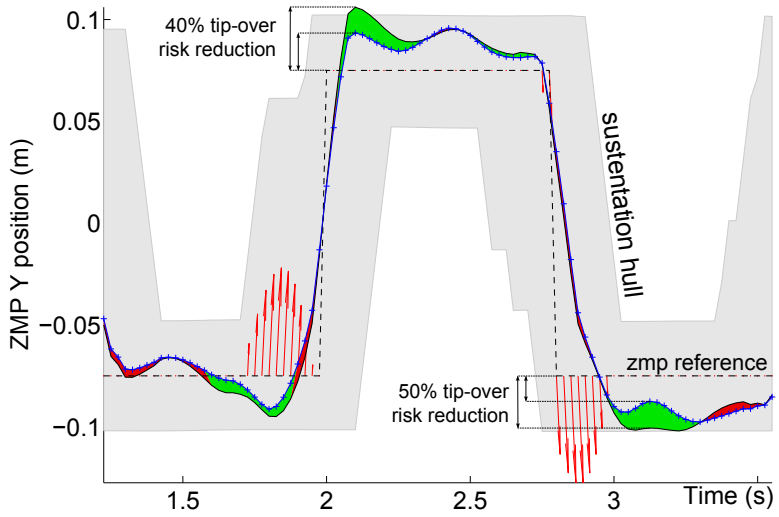

Figure 4.13.: Evolution of the ZMP in lateral direction. The contribution to tip-over risk minimization brought by the DMPC controller is depicted and prevents critical balance states. *Blue cross: DMPC – black solid: ZMP Preview Control – red arrows: normalized external effort.*

ure 4.14 shows that the robot can control the reaction forces from the ground until a limit of $\mu_{min} = 0.25$ with the DMPC framework and down to $\mu_{min} = 0.34$ with the ZMP Preview Control framework. Relatively to the expected friction coefficient $\mu = 0.5$ these figures bring an uncertainty tolerance — robustness — of 50% for the DMPC framework and less than 35% for the ZMP Preview Control framework. The consequences of this higher robustness are illustrated in the multimedia attachment[6] of [Ibanez2014a], presenting a disturbed scenario where postural

[6] `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6906610`

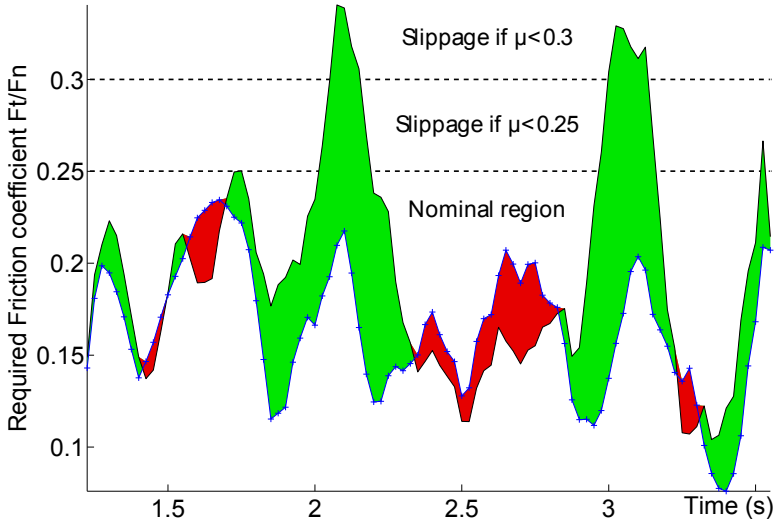

Figure 4.14.: Evolution of the required friction coefficient. The relative amplitudes of tangential and normal contact forces is significantly decreased by the anticipated coordination of horizontal and vertical center of mass dynamics of the DMPC controller, thus reducing slippage risk in case of uncertainties on the contact conditions. *Blue cross: DMPC – black solid: ZMP Preview Control.*

stability is maintained with the DMPC controller while the ZMP Preview Control framework fails at ensuring contacts stability.

The DMPC framework is thus prominently more robust to unexpected disturbances than the ZMP Preview Control framework. Note that, as desired[7] and due to the exponential growth of the slippage cost function H^s, no noticeable performance gain can be found in the nominal region and only slippage-prone peaks are cut down.

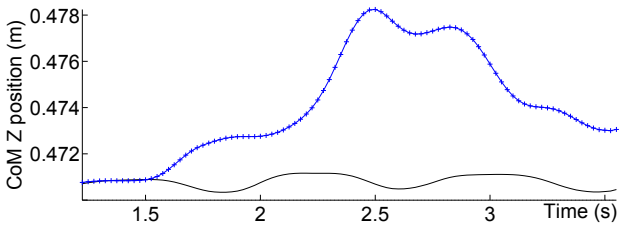

Figure 4.15.: Evolution of the center of mass altitude. The CoM vertical dynamics are exploited by the DMPC controller, concurrently to its horizontal dynamics, to compromise between slippage and tip-over risks. *Blue cross: DMPC – black solid: ZMP Preview Control.*

Finally, the DMPC framework allows to automatically regulate the altitude of the CoM of the robot and its variations, as shown in figure 4.15. The ZMP Preview Control framework bounds the CoM to stay at a constant altitude in order to preserve the validity of the cart-table model, while the DMPC framework takes

[7] *cf.* Sec. 4.1.3, p. 87

benefit from this degree of freedom in order to compromise between balance and stability objectives.
Multi-objective MPC formulations therefore allow to coordinate at the predictive level distinct degrees of freedom of the system to maximize the activity performance.

4.5. Conclusion

The control problem of postural stability of a humanoid robot is approached in this chapter in a multi-objective distributed model predictive control framework. This novel formulation takes advantage of different models of the balance dynamics, coordinated along multiple future horizons and of distinct accuracy, and efficiently increases the robustness of the system against strong and unexpected external perturbations.

Contributions of this predictive control architecture are highlighted with the simulation of a walking activity under external pushes: significant gains in balance and contacts stability are observed compared to a ZMP Preview Control framework. Tip-over and slippage risks are noticeably minimized with the predictive control of the center of mass horizontal and vertical dynamics, and of the robot angular momentum.

The multi-objective predictive approach provides means to anticipate complex coordination strategies between conflicting tip-over and slippage objectives: the rotational and linear dynamics of the system are conjointly exploited over a future horizon to achieve an overall gain in both tasks performance. The problem is **distributed** through a dual decomposition technique providing insight on the conflicts between objectives by explicitly exposing them, and enables a **modularity** of the control structure which can be exploited to take advantage of approximations on the coupling between models. This technique can furthermore be employed to decompose the problem into sub-problems of smaller sizes, therefore significantly **reducing the computational cost** of the predictive control level by taking advantage on multi-core systems of the resulting non-cooperative, parallel structure of the problem thus formulated.

The predictive control framework is moreover exploited in this chapter at **different time scales and levels of abstraction**. Finer models are used conjointly to reduced ones, over various future preview horizons which allows for a refined tuning of the MPC controller performance. The computational cost of refined models is reduced through a sequential distribution of the problem exploiting an approximation on the coupling between models, while still benefiting the overall activity performance.

* * *

Biped systems nevertheless comprise of other degrees of freedom than the sole postural adjustments to be exploited to solve the balance problem. Shifts in the contact configuration indeed provide decisive means to recover balance through the change in the admissible set of postural adjustments. Contact shifts are furthermore compulsory for walking activities, yet their discrete nature makes their integration as control parameters challenging. Biped walking is indeed hybrid in nature: continuous regulations of the posture are performed simultaneously to discrete footsteps. Efficient gaits must consistently coordinate postural adjustments with contact shifts to ensure optimal performance, although discrete actions are generally supported at the decision level rather than the control level. This challenge is met in the following chapter with the introduction of a mixed-integer MPC problem generating optimal coordination strategies between discrete and continuous parameters, with respect to both balance and walking objectives.

CHAPTER 5

Emergence of complex behaviors from hybrid models and MPC

Contents

Humanoid robots employ locomotion systems which are essentially of hybrid nature, combining discrete supports and multi-body dynamics. Legged locomotion couples these two sub-systems in different ways, and distinct behaviors emerge depending on the desired performance and constraints [Miguel2013].
The choice of balance strategy is all the more important in situations where the robot undergoes large physical disturbances. Appropriate strategies in such cases generally lie between whole-body balancing and changes in the base of support (BoS). The robotics literature presents various solutions within this range, involving torque compensation [Prahlad2008], bracing behaviors [Kanzaki2005] at the center of mass, angular momentum rejection [Pratt2006] and shifts in foot placement [Pratt2006, Herdt2010].

Changes in the BoS, however, have the potential to provide a greater degree of stabilization than whole-body adjustments in a fixed-support configuration [Maki2003], and may, on one hand, even be necessary depending on the activity and the constraints the system is subject to. On the other hand control of these changes may be more challenging. It requires indeed the determination of the adequate time, duration, distance and direction of the shift, while regarding both constraints on the system and the desired motion of the robot. With the Capture Region approach [Pratt2006], Pratt *et al.* propose to solve the problem of when and where to take a step, along with suitable CoM and angular momentum behaviors. Although it has largely demonstrated its efficiency in push-recovery cases, this method lacks consideration of the constraints the robot is subject to and of the engaged activity, which could potentially affect the feasibility of the expected step. Furthermore, little insight on the suitable duration of the step is given.

Predictive approaches are appropriate to preview the influence of the duration and placement of the step taken, and can be applied to the Capture Region method as Krause *et al.* [Krause2012] propose in a MPC framework. MPC indeed provides a future time window to estimate the evolution of the system's state, and the formulation of an optimization problem is favorable to the consideration of constraints and objectives the robot must comply with. However, the additional complexity generally requires the use of reduced models, constraints approximations and predefined entities or heuristics in order to obtain computationally-efficient formulations. Moreover the hybrid nature of biped walking, involving continuous evolution of the system's motion and discrete changes in constraints and forces acting on it as illustrated in figure 5.1, tends to prevent straightforward formulations. In order to confine the resolution to the continuous, smooth part of the problem, Herdt *et al.* [Herdt2010] propose to exploit a previously given horizon of contacts activation, pre-specifying the time and duration of the steps, to simultaneously optimize the CoM trajectory with contacts positions. Another method consists in relaxing the problem: Mordatch *et al.* [Mordatch2012] make for example use of hard-constraints relaxation for rigid contacts, in order to ensure regularity of the model.

This chapter proposes an original approach to the capture, in a predictive framework, of the influence of both CoM dynamics and changes in the BoS with respect to constraints on the system and the ongoing walking activity. No relax-

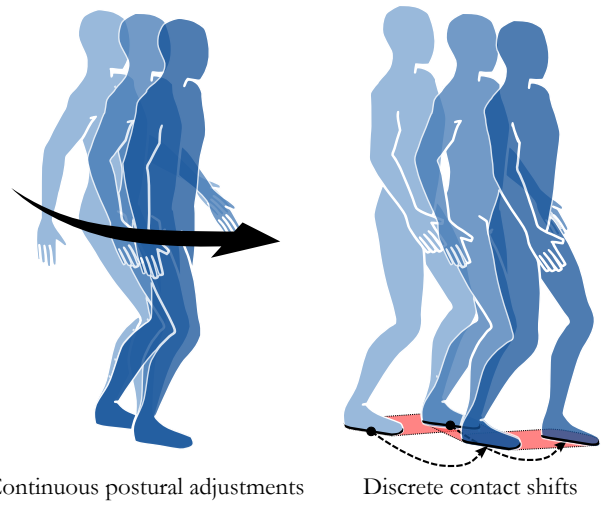

Figure 5.1.: Illustration of the hybrid nature of biped walking. Strategies combine continuous postural adjustments with discrete changes in the contact configuration.

ation of the hybrid nature of biped walking is carried out, and discrete events are described using a computationally favorable, redundant set of highly-constrained integer variables. Behavior of the system is specified at a high level, as a sole ratio between balance and performance of the walking activity. The resulting MPC problem takes the form of a quadratic, linearly constrained mixed-integer program (MIQP) which allows to determine, over a preview horizon, an optimal strategy between changes in the BoS and CoM behavior, subject to multiple constraints, maximizing balance and performance of a walking activity.

A linearly-constrained, mixed-integer set of variables is first described to characterize admissible changes in the BoS. The Zero-Moment Point approach is employed to capture, in terms of balance, the state of the system with respect to the BoS. The resulting model allows the linear expression of several constraints related to the walking activity.

This model of the system is then employed in an MPC framework, and the control objective is defined as a compromise between balance and performance of the walking task, which is formulated as the tracking of a desired CoM trajectory. An optimal horizon of changes in the BoS and CoM trajectory is computed from a MIQP, without the use of pre-defined gait patterns or heuristics.

Simulations results last validates this approach in various scenarii. Performance of the tracking task is demonstrated in a sinusoidal trajectory tracking case, and the introduction of large physical disturbances on the robot exhibits the conjoint adaptation of gait pattern and CoM trajectory. Last, the influence of the balance and walking weights is exhibited in a push recovery case where the system behavior, as a response to an unknown external action, significantly varies for different values of the weights ratio.

5.1. Hybrid nature of biped walking: a Mixed-Integer model

In order to adopt the appropriate balance strategy, control algorithms should exploit a model of the system describing the effect of various quantities of interest on the postural stability of the robot. Indeed, employing this model to evaluate the performance of different evolutions of each of these quantities may lead to the identification of the best combination to select.

Under several hypotheses, the Zero-Moment Point [Vukobratovic1972] approach captures the balance state of the system by relating its CoM dynamics to the base of support. However, the hybrid nature of biped walking clearly differentiates CoM dynamics changes from BoS shifts: the former is continuous in essence, while the latter is restrained to discrete events. MPC methods, writing the control problem as an optimization problem, generally consider the continuous members of the ZMP model solely as degrees of freedom. Kajita *et al.* [Kajita2003] and Wieber [Wieber2006] for example adjusted the CoM dynamics to maintain balance; Herdt *et al.* [Herdt2010] isolated the discrete aspect of changes in the BoS in a priorly-defined activation matrix, allowing to add the determination of the footsteps location to the MPC problem.

However, the time of activation of discrete changes in the BoS might noticeably affect the performance of both balance and the walking task. This section therefore describes a model considering simultaneously the CoM dynamics, amplitude and instants of changes in the BoS. Aiming at a computationally-efficient formulation, the use of a redundant set of integer and real variables to describe the contact state of the robot allows to keep a linear form of the various constraints the system is subject to.

Mixed-Integer Programming (MIP) [Osiadacz1990, Bemporad1999] indeed proposes an ideal framework to account for logic events and non-convex admissible domains. MIP approaches generally employ integer variables to describe logic and combinatorial systems, these variables either being used as discrete state/input values or as triggers to activate and deactivate constraints in the optimization problem. The latter use offers opportunities to regard non-convex admissible domains as an arrangement of convex regions, an integer variable specifying in which convex region the — now convex — problem is currently considered.

While integer programs are NP-hard, efficient algorithms and solvers are available for specific classes of programs, exploiting the form of the optimization problem or employing heuristics. For example, quadratic MIPs (MIQPs) can be solved using branch and bound algorithms which solve a graph of QP problems resulting from the consideration of integer variables as real variables[1]. Real-valued QP problems being convex, the resolution of these relaxed sub-problems is computationally efficient.

[1] *cf.* App. A, p. 171

5.1.1. Mixed-Integer biped walking description

To capture the discrete nature of changes in the BoS, with a view to future optimization, a choice of linearly-constrained, redundant descriptors of the contact state is proposed. The choice of these descriptors is driven by the aim of formulating the balance problem as a MIQP. Therefore, the constraints they are subject to must be linear to preserve the convexity of the problem, leading to the introduction of intermediate — and possibly redundant — variables.

5.1.1.1. Core descriptors of the base of support

The amplitude and position of the BoS are described by the bounding box of the feet centers, as illustrated in figure 5.2: real-valued variables $(\boldsymbol{a}, \boldsymbol{b}) \in \mathbb{R}^2 \times \mathbb{R}^2$ are defined as the upper and lower bounds, respectively, of the position in the two horizontal directions of the feet in contact.[2] Bounding positions are chosen instead of left and right feet positions for biped systems in order to obtain a more natural expression of the BoS convex hull.

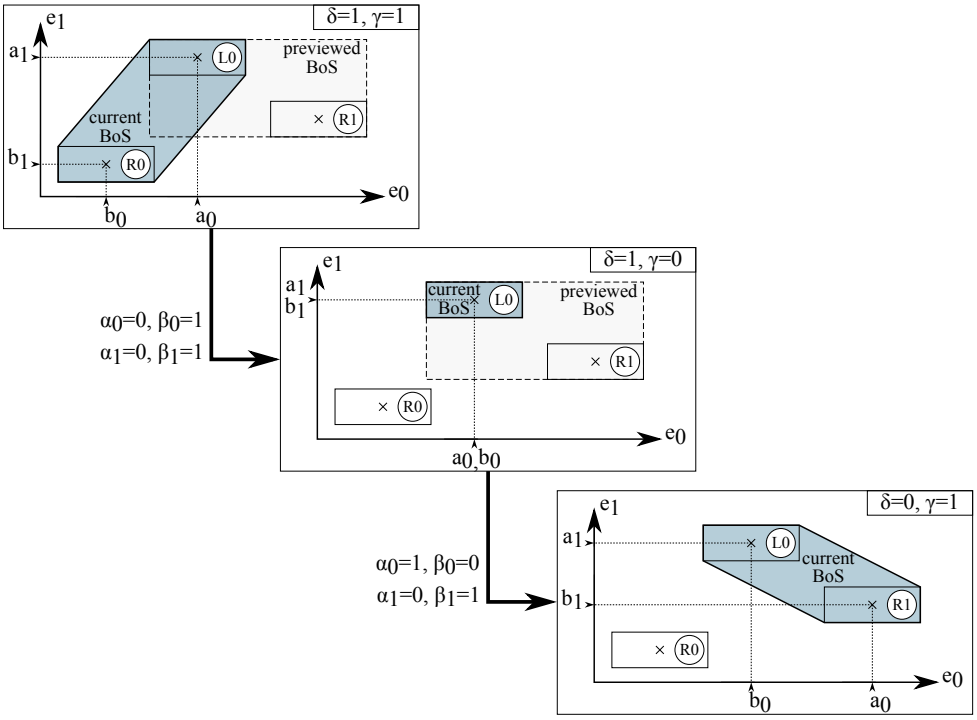

Figure 5.2.: Illustration of the mixed-integer description of the base of support. Real valued $\boldsymbol{a}$ and $\boldsymbol{b}$ describe bounds of the position of the feet in contact, with their respective binary rising/falling edges $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Binary variable γ differentiates single and double support phases, and δ restrains the evolution of bounds $\boldsymbol{a}, \boldsymbol{b}$ during transitions from double to single support. Current BoS is linearly expressed (*dark gray.*) and previewed BoSs are approximated by their bounding box (*light gray.*).

[2] This description implies that the feet contact and lift-off the ground parallel to it, and are rigid bodies.

The discrete essence of changes in the BoS leads to the definition of several constraints, called shape constraints, restraining variables $(\boldsymbol{a}, \boldsymbol{b})$. Essentially, $(\boldsymbol{a}, \boldsymbol{b})$ must be piecewise-constant. This constraint can be linearly expressed with the introduction of binary variables $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \{0,1\}^2 \times \{0,1\}^2$ as rising and falling edges of $\boldsymbol{a}$ and $\boldsymbol{b}$, respectively. Indeed, by definition $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ write

$$\forall i \in \{0,1\}, \forall t \in \mathbb{R}, \quad \begin{cases} \alpha_i(t) = 1 \quad \Rightarrow \quad \alpha_i(t^-) = \alpha_i\left(t^+\right) = 0 & (5.1) \\ \alpha_i(t) = 0 \quad \Rightarrow \quad a_i(t^-) = a_i\left(t^+\right) = a_i(t) & (5.2) \end{cases}$$

and similarly for the pair $(\boldsymbol{\beta}, \boldsymbol{b})$. Variables $\boldsymbol{\alpha}, \boldsymbol{\beta}$ can have their value set to 1 at distinct instants solely (5.1) to translate the discrete characteristic of contact shifts, and $\boldsymbol{\alpha} = \mathbf{0}$ (*resp.* $\boldsymbol{\beta} = \mathbf{0}$) imposes constancy of $\boldsymbol{a}$ (*resp.* $\boldsymbol{b}$) as written in (5.2).
Note that changes in $\boldsymbol{a}$ and $\boldsymbol{b}$ are subject to the following additional constraints

$$\forall i \in \{0,1\}, \forall t \in \mathbb{R}, \quad b_i(t) \leq a_i(t), \tag{5.3}$$
$$\forall i \in \{0,1\}, \forall t \in \mathbb{R}, \quad \alpha_i(t) + \beta_i(t) \leq 1, \tag{5.4}$$
$$\forall t \in \mathbb{R}, \quad \alpha_0(t) + \beta_0(t) = \alpha_1(t) + \beta_1(t). \tag{5.5}$$

Indeed, the definition of $\boldsymbol{a}$ and $\boldsymbol{b}$ as respectively upper and lower bounds has to be enforced (5.3); also, as only one foot can move at a time, it imposes that, in a given direction, only one of the bounds $\boldsymbol{a}$ or $\boldsymbol{b}$ can change (5.4). Last, a change in a first direction for either bound $\boldsymbol{a}$ or $\boldsymbol{b}$ must allow a change in the other direction (5.5).

Constraints (5.1)–(5.4) define the bounding box of the support surface as restricted between two bi-dimensional, piecewise constant variables $\boldsymbol{a}$ and $\boldsymbol{b}$. However, the nature of the changes occurring in $\boldsymbol{a}$ and $\boldsymbol{b}$ imposes the consideration of additional variables and constraints to define and guarantee the admissibility of such changes.

5.1.1.2. Additional descriptors and constraints

Bounds $(\boldsymbol{a}, \boldsymbol{b})$ are furthermore implicitly related to feet positions: thus there exists couplings between the two directions of $\boldsymbol{a}$ and $\boldsymbol{b}$, called admissibility constraints.

First, single support phases (SS) impose that $\boldsymbol{a}$ equals $\boldsymbol{b}$, as the BoS is reduced to one foot (*cf.* figure 5.2). SS phases must hence be differentiated from double support ones (DS): the binary variable $\gamma \in \{0,1\}$ is introduced to this purpose, and allows to define this constraint linearly.

Second, each change in the pair $(\boldsymbol{a}, \boldsymbol{b})$ leads to the alternation of γ, as any shift in the BoS corresponds to either a transition SS→DS or DS→SS. Note that, as a result, γ is fully defined by $(\boldsymbol{a}, \boldsymbol{b})$ and $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ assuming that an initial value of γ is known.

Last, potential changes in $\boldsymbol{a}$ and $\boldsymbol{b}$ from DS to SS depend on the configuration of the feet, relatively to bounds $\boldsymbol{a}$ and $\boldsymbol{b}$, in the previous DS phase. Indeed, as illustrated in figure 5.2, only the pairs (a_1, a_0) or (b_1, b_0) can change when leaving the first DS configuration. On the contrary, leaving the last DS configuration can only lead to changes in the pairs (b_1, a_0) or (a_1, b_0). This coupling can be

linearly expressed with the introduction of the binary variable $\delta \in \{0, 1\}$: the first configuration in figure 5.2 corresponds to $\delta = 1$ and the last to $\delta = 0$. Note that this relation is bilateral: when leaving SS to DS, changes in $(\boldsymbol{a}, \boldsymbol{b})$ *set* the value of δ, and when switching back to SS, the value of δ *restrains* potential changes in $(\boldsymbol{a}, \boldsymbol{b})$.
Figure 5.3 proposes typical evolutions of this set of BoS variables, and admissibility constraints are further detailed in App. B.

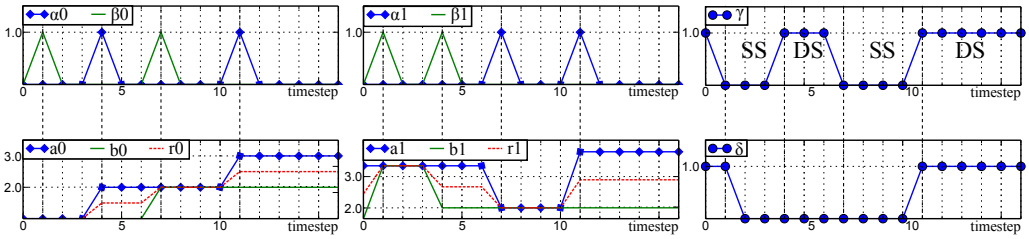

Figure 5.3.: Typical evolution of the mixed-integer, linearly-constrained variables $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \delta, \gamma)$, related to the base of support. Variable $\boldsymbol{r} = (\boldsymbol{a} + \boldsymbol{b})/2$ is displayed here to describe the center of the supporting surface.

5.1.2. Balance constraints

Contact state of the robot can therefore be described by the linearly-constrained, mixed-integer set of variables $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \delta, \gamma)$. The ZMP model, neglecting rotational effects, relates CoM dynamics to the BoS; state $\boldsymbol{\xi}$ of the system in a balance perspective can thus be reduced to

$$\boldsymbol{\xi} \triangleq [\quad \boldsymbol{a} \quad \boldsymbol{b} \quad \boldsymbol{\alpha} \quad \boldsymbol{\beta} \quad \delta \quad \gamma \quad \boldsymbol{c} \quad \dot{\boldsymbol{c}} \quad \ddot{\boldsymbol{c}} \quad]^T, \tag{5.6}$$

where $\boldsymbol{c} \in \mathbb{R}^3$ is the position of the CoM in the world frame. In the rest of this chapter, time is sampled at discrete control instants t_i, and notation v_j for function v of time t denotes the value $v(t_j)$, and $v_{j|k}$ the value $v(t_j)$ estimated from control time t_k.

As stated earlier, the major contribution of such a description of the contact state is that, in discrete time, shape and admissibility constraints can be put in the linear form

$$\forall k \in \mathbb{N}, \quad \mathbf{A}_{cl}\boldsymbol{\xi}_k + \mathbf{A}_{cr}\boldsymbol{\xi}_{k+1} \preceq \boldsymbol{f}_c, \tag{5.7}$$

where $\mathbf{A}_{cl}$ and $\mathbf{A}_{cr}$ are $n_c \times 19$ matrices, and $\boldsymbol{f}_c$ a vector in $\mathbb{R}^{n_c}$.

Biped postural stability, in non-sliding cases, can be obtained by avoiding tip-over of the humanoid: with coplanar contacts, the center of pressure must stay away from the edges of the BoS. The ZMP approach [Vukobratovic1972] allows the computation of the position $\boldsymbol{p}$ on the ground of the CoP from the dynamics of the system. Ignoring rotational effects compared to translational ones, and neglecting vertical acceleration of the CoM compared to its horizontal acceleration, $\boldsymbol{p}$ writes

$$\boldsymbol{p} = \boldsymbol{h} - \frac{\boldsymbol{c} \cdot \boldsymbol{e}_2}{g}\ddot{\boldsymbol{h}} \quad \text{where} \quad \boldsymbol{h} \triangleq \boldsymbol{c} - (\boldsymbol{c} \cdot \boldsymbol{e}_2)\boldsymbol{e}_2, \tag{5.8}$$

with $\boldsymbol{e}_2$ the ascendant vertical direction, g the gravity amplitude and $\boldsymbol{h}$ the horizontal position of the CoM.

In order to have a linear expression of $\boldsymbol{p}$ with respect to $\boldsymbol{\xi}$ and considering the approximation on the vertical acceleration of the CoM, the altitude of the CoM is considered as constant in the rest of this chapter. The relevant CoM dynamics are thus the horizontal ones $\boldsymbol{h}$, therefore system state $\boldsymbol{\xi}$, matrices $\mathbf{A}_{cl}$ and $\mathbf{A}_{cr}$ and vector $\boldsymbol{f}_c$ are modified accordingly.

Additionally, the balance constraint is linear with respect to the horizontal dynamics of the CoM, but changes in the supporting surface being reflected by changes in state variables $(\boldsymbol{a}, \boldsymbol{b})$, the inclusion constraint is quadratic with respect to the state $\boldsymbol{\xi}$ of the system. Nevertheless, as shown in [Herdt2010], overestimating the convex hull of this surface by its bounding box in forward and lateral directions brings the definition of a set of linear inequality constraints with respect to $\boldsymbol{\xi}$. Note that this approximation is solely made for future potential DS phases: constraints for an established BoS are fully implemented without overestimation (*cf.* figure 5.2). Under this overestimation[3], the CoP constraints write

$$\forall k \in \mathbb{N}, \quad \mathbf{A}_p \boldsymbol{\xi}_k \preceq \boldsymbol{f}_p, \tag{5.9}$$

where $\mathbf{A}_p$ is a $n_p \times 16$ matrix, and $\boldsymbol{f}_p$ a vector in $\mathbb{R}^{n_p}$.

The previous description of the contact state, with the ZMP model, brings the resulting mixed-integer biped model at time t_k

$$\boldsymbol{\xi}_k \triangleq \left[\begin{array}{cccccc} \boldsymbol{a}_k & \boldsymbol{b}_k & \boldsymbol{\alpha}_k & \boldsymbol{\beta}_k & \delta_k & \gamma_k \end{array} \begin{array}{ccc} \boldsymbol{h}_k & \dot{\boldsymbol{h}}_k & \ddot{\boldsymbol{h}}_k \end{array} \right]^T$$

$$\text{s.t.} \quad \left\{ \begin{array}{rcl} \mathbf{A}_{c,r} \boldsymbol{\xi}_k & \preceq & \boldsymbol{f}_c - \mathbf{A}_{c,l} \boldsymbol{\xi}_{k-1}, \\ \mathbf{A}_p \boldsymbol{\xi}_k & \preceq & \boldsymbol{f}_p, \\ (\boldsymbol{\alpha}_k,\ \boldsymbol{\beta}_k) & \in & \{0,1\}^2 \times \{0,1\}^2, \\ (\delta_k,\ \gamma_k) & \in & \{0,1\} \times \{0,1\}, \\ (\boldsymbol{a}_k,\ \boldsymbol{b}_k) & \in & \mathbb{R}^2 \times \mathbb{R}^2, \\ \left(\boldsymbol{h}_k,\ \dot{\boldsymbol{h}}_k,\ \ddot{\boldsymbol{h}}_k\right) & \in & \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2, \end{array} \right. \tag{5.10}$$

which describes and restrains the balance state of the system.

5.1.3. Walking motion constraints

The model described in (5.10) allows the consideration of a large variety of linear constraints, inherent to the control problem of walking humanoids. Relevant constraints regarded in this paper are maximum leg span, maximum swinging foot average velocity and minimal SS/DS phases durations. Such constraints will ban strategies involving too large or fast steps, and hinder solutions such as fast series of small steps. Note that additional constraints such as maximal SS/DS durations, bounds on the CoM velocity and acceleration, or bounds on positions of the feet can also be put in a linear form with respect to the system state $\boldsymbol{\xi}$.
Let $\bar{v}$ denote the maximal swinging foot velocity, $\bar{s}$ the maximal step length and $\bar{t}$

[3] remarks on this approximation are provided in Sec. 5.3.4

the largest of minimum durations of SS and DS phases. Constraints on leg span, foot velocity and SS/DS durations write

$$\sum_{t_{k+j}-t_k \leq \max(\bar{s}/\bar{v},\bar{t})} \mathbf{A}_{w,j}\boldsymbol{\xi}_{k+j} \preceq \boldsymbol{f}_w, \quad j \in \mathbb{N}, \tag{5.11}$$

where $\mathbf{A}_{w,j}$ is a $n_w \times 19$ matrix, and $\boldsymbol{f}_w$ a vector in $\mathbb{R}^{n_w}$. It can be shown[4] that $\boldsymbol{f}_w$ depends on a history of states $\boldsymbol{\xi}$ prior to t_k, and can be written in the form

$$\boldsymbol{f}_w = \boldsymbol{f}_{w_0} + \sum_{t_k - t_{k-j} < \max(\bar{s}/\bar{v},\bar{t})} \mathbf{B}_{w,j}\boldsymbol{\xi}_{k-j}, \quad j \in \mathbb{N}^*. \tag{5.12}$$

The condition $t_{k+j} - t_k \leq \max\left(\bar{s}/\bar{v}, \bar{t}\right)$ (*resp.* $t_k - t_{k-j}$) states that the forward (*resp.* backward) history influence is irrelevant beyond the time step where both minimum duration and maximum average velocity constraints are necessarily respected, and thus avoids the stacking up of unnecessary constraints. A constraint to avoid the overlapping of feet is furthermore added, in the form of a linearized collision avoidance constraint.

This model is employed in the following section to evaluate and optimize the performance, in terms of walking activity and balance, of admissible evolutions of the system state.

5.2. Predictive walking as a Mixed-Integer Quadratic Program

The walking control problem, in a predictive framework, consists in finding an optimal and admissible horizon of future system states $\boldsymbol{\xi}_{k+j|k}$ which ensures the performance of an ongoing walking activity while maximizing balance of the system. The descriptors of the system state presented in Sec. 5.1 being taken as variables of an optimization problem, these objectives can be written in the form of quadratic functions, leading to the formulation of a MPC problem as a mixed-integer quadratic program.

5.2.1. Model preview

Considering approximations exploited in the ZMP model, a minimal set of parameters can be obtained to describe an horizon of system states $\boldsymbol{\xi}_{k+j|k}$. Indeed, a discrete integration scheme brings, with piecewise constant CoM jerks $\boldsymbol{u} = \partial^3\boldsymbol{h}/\partial t^3$

$$\forall j \in \mathbb{N}^*, \quad \hat{\boldsymbol{h}}_{k+j+1|k} = \mathbf{A}_h\hat{\boldsymbol{h}}_{k+j|k} + \mathbf{B}_h\boldsymbol{u}_{k+j+1|k}, \tag{5.13}$$

where $\hat{\boldsymbol{h}} = [\boldsymbol{h}\ \dot{\boldsymbol{h}}\ \ddot{\boldsymbol{h}}]^T$ is the CoM horizontal dynamics, and $\mathbf{A}_h$, $\mathbf{B}_h$ integration matrices[5]. The minimal set of variables required to preview $\boldsymbol{\xi}_{k+j+1|k}$ from $\boldsymbol{\xi}_{k+j|k}$ is hence denoted $\boldsymbol{\chi}_{k+j+1|k}$, defined as

$$\boldsymbol{\chi} \triangleq [\quad \boldsymbol{a} \quad \boldsymbol{b} \quad \boldsymbol{\alpha} \quad \boldsymbol{\beta} \quad \delta \quad \gamma \quad \boldsymbol{u} \quad]^T.$$

[4] *cf.* App. B, p. 175

[5] *cf.* Sec. 2.4.3, p. 36 and Sec. 3.1, p. 56.

The model (5.10) can thus be written as the Linear Time-Invariant process

$$\forall j \in \mathbb{N}^*, \quad \boldsymbol{\xi}_{k+j+1|k} = \mathbf{Q}\boldsymbol{\xi}_{k+j|k} + \mathbf{T}\boldsymbol{\chi}_{k+j+1|k}, \tag{5.14}$$

where $\mathbf{Q}$ and $\mathbf{T}$ are state description matrices derived from (5.13). Relation (5.14) allows to preview an horizon $\boldsymbol{C}_{k,N}$ of N future states $\boldsymbol{\xi}_{k+j|k}$ from an horizon $\boldsymbol{X}_{k,N}$ of N future inputs $\boldsymbol{\chi}_{k+j|k}$ and the actual state $\boldsymbol{\xi}_{k|k} = \boldsymbol{\xi}_k$. The preview writes, with $\mathbf{P}$ and $\mathbf{R}$ combinations of $\mathbf{Q}$ and $\mathbf{T}$

$$\boldsymbol{C}_{k,N} \triangleq \begin{bmatrix} \boldsymbol{\xi}_{k+1|k} \\ \vdots \\ \boldsymbol{\xi}_{k+N|k} \end{bmatrix} = \mathbf{P}\boldsymbol{\xi}_k + \mathbf{R} \begin{bmatrix} \boldsymbol{\chi}_{k+1|k} \\ \vdots \\ \boldsymbol{\chi}_{k+N|k} \end{bmatrix},$$

denoted as

$$\boldsymbol{C}_{k,N} = \mathbf{P}\boldsymbol{\xi}_k + \mathbf{R}\boldsymbol{X}_{k,N}. \tag{5.15}$$

This linear equation (5.15) allows to aggregate the linear equalities (5.7), (5.9) and (5.11) into

$$\mathbf{A}\boldsymbol{X}_{k,N} \preceq \boldsymbol{f} \tag{5.16}$$

where $\boldsymbol{f}$ depends on a history of actual and previous states $\boldsymbol{\xi}$, as shown in (5.7), (5.11) and (5.12).

5.2.2. Walking performance criterion

A walking activity can be interpreted as reaching a target position with a desired horizontal velocity, which can be expressed at the level of the CoM to involve the model state $\boldsymbol{\xi}$. It is in such terms a tracking task, whose performance can be evaluated as a tracking error. Let $\hat{\boldsymbol{h}}^r$ denote the desired trajectory of the CoM. Objective J_w of the walking activity can be written, over a preview horizon, as the minimization of

$$J_{wk} \triangleq \sum_{j=1}^{N} \left\| \mathbf{S} \left(\hat{\boldsymbol{h}}_{k+j|k} - \hat{\boldsymbol{h}}^r_{k+j|k} \right) \right\|^2, \tag{5.17}$$

where $\mathbf{S}$ is a 6×6 weighting selection matrix, diagonal, defining whether position, velocity and/or acceleration are tracked in each of the two horizontal directions. For example, a standstill activity aiming at $\dot{\boldsymbol{h}} \to \mathbf{0}$ can be expressed with $\hat{\boldsymbol{h}}^r = \mathbf{0}$ and the only non-null terms of $\mathbf{S}$ corresponding to velocity in both horizontal directions; that is, $J_{wk} = \sum_{j=1}^{N} \|\dot{\boldsymbol{h}}_{k+j|k}\|^2$.

This weighting of the relative influence of acceleration, velocity and position tracking in both directions allows the specification of various walking activities. Indeed, the previous example of a standstill activity specification can also be defined as a desired CoM position instead of a null velocity, or a combination of the two.

Additionally, the specification in the two horizontal directions can be finely tuned, a walking activity generally requiring a motion in a single direction without specific requirements in the orthogonal one.

5.2.3. Biped balance objective

While performance of the walking activity is essential, robustness of the posture of the robot is also a major objective in walking motions. Balance is guaranteed, under the assumptions of the model, by the CoP constraint (5.9) in (5.16). Nevertheless, as a tip-over situation occurs when the CoP reaches the edges of the BoS, robustness of the balance state of the robot can be captured as a distance to the edges. The balance maximization objective can thus be written as the minimization of J_b

$$J_{bk} \triangleq \sum_{j=1}^{N} \left\| \boldsymbol{p}_{k+j|k} - \boldsymbol{r}_{k+j|k} \right\|^2, \tag{5.18}$$

where $\boldsymbol{r} = (\boldsymbol{a}+\boldsymbol{b})/2$ is the center of the BoS, *i.e.* the point at the greatest distance from the edges of the supporting surface.

Secondary objectives are added to the control problem in these works for regularization purposes. They aim at minimizing CoM jerks, avoid excessive changes in solutions from one control step to another, keep track of the previous BoS size and prefer DS phases over SS ones. These regularization objectives are written in the form of a quadratic cost q, which can be used to reward desired behaviors in order to favor short and frequent stepping over slower gaits, for example.

5.2.4. MIQP form of the walking MPC problem

Objectives (5.17) and (5.18) are quadratic with respect to $\boldsymbol{X}_{k,N}$, and are considered in the global cost function J

$$J_k \triangleq \omega_b J_{bk} + \omega_w J_{wk} + q = \boldsymbol{X}_{k,N}^T \mathbf{H} \boldsymbol{X}_{k,N} + \boldsymbol{d}^T \boldsymbol{X}_{k,N}, \tag{5.19}$$

where $\mathbf{H}$ is a positive definite matrix, $\boldsymbol{d}$ a vector and (ω_b, ω_w) scalar weights defining a compromise between balance robustness and tracking performance. The MPC problem finally writes

$$\begin{aligned} &\min_{\boldsymbol{X}_{k,N}} \ \boldsymbol{X}_{k,N}^T \mathbf{H} \boldsymbol{X}_{k,N} + \boldsymbol{d}^T \boldsymbol{X}_{k,N} \\ &\text{s.t.} \quad \begin{cases} \mathbf{A}\boldsymbol{X}_{k,N} \preceq \boldsymbol{f} \\ \boldsymbol{\xi}_{k|k} = \boldsymbol{\xi}_k \\ \boldsymbol{\xi}_{k+j+1|k} = \mathbf{Q}\boldsymbol{\xi}_{k+j|k} + \mathbf{T}\boldsymbol{\chi}_{k+j+1|k}, \\ \left(\boldsymbol{a}_{k+j|k},\ \boldsymbol{b}_{k+j|k}\right) \in \mathbb{R}^2 \times \mathbb{R}^2, \\ \left(\boldsymbol{\alpha}_{k+j|k},\ \boldsymbol{\beta}_{k+j|k}\right) \in \{0,1\}^2 \times \{0,1\}^2, \\ \left(\delta_{k+j|k},\ \gamma_{k+j|k}\right) \in \{0,1\} \times \{0,1\}, \\ \boldsymbol{u}_{k+j|k} \in \mathbb{R}^2, \end{cases} \end{aligned} \tag{5.20}$$

which is a Mixed-Integer Quadratic Program in canonical form. The computational complexity of problem (5.20) is strongly related to the number of non-real variables due to their combinatorial nature. Nevertheless its QP form allows[6] the use of fast algorithms and binary variables can be sampled at a lower frequency, as discussed in Sec. 5.3.4. Furthermore, for closed-loop stability considerations,

[6] *cf.* App. A, p. 171

Bemporad *et al.* demonstrate that quadratic MPC problems on linear hybrid systems can be treated with the stability and safety analysis tools developped for hybrid systems [Bemporad2002b].

5.3. Results: emergence of complex behaviors

Contribution of the MPC formulation (5.20) is exposed with several simulation results, demonstrating the variety of behaviors generated in diverse scenarii.

Opening results illustrate that, without the use of any heuristics other than motion constraints and controller weights, the MPC (5.20) automatically generates an intuitive gait pattern to follow a varying reference velocity. A second set of results exhibits optimal modifications of the gait pattern to accommodate an ongoing walking activity to large physical disturbances on the robot. Last, results in a push-recovery scenario demonstrate how weights in (5.19) allows to implicitly define, at a high level, different behaviors of the humanoid.

Simulations presented in this section are performed using the Arboris-Python simulator [BarthelemyArboris], and whole-body instantaneous motion from optimal outputs of the MPC (5.20) is ensured by an LQP-based controller [Salini2010] for an iCub [Sandini2007] robot model. This reactive level of the resulting two-layered setup is similar to the one presented in Chapters 3 and 4, with the reactive primitives related to the feet motion control tracking in this case the targets resulting from the outputs of the MIQP problem (5.20).
The system state is previewed over an horizon of $1.0s$, and simulations are run with a time step of $dt = 1.e^{-2}s$. The humanoid weighs $\approx 27kg$ with a height of $\approx 1m$.

5.3.1. Gait generation: sinusoidal velocity tracking

A first simulation scenario is performed to illustrate the behavior of the MIQP problem in nominal conditions. A walking activity is specified as the tracking of a sinusoidal CoM velocity in the forward direction, and a null velocity in the lateral direction. Major weights (ω_b, ω_w) in the cost function (5.19) are equal, and forward and lateral directions have the same importance in the tracking error function (5.17), *i.e.* $S_{ij} \in \{0, 1\}$.

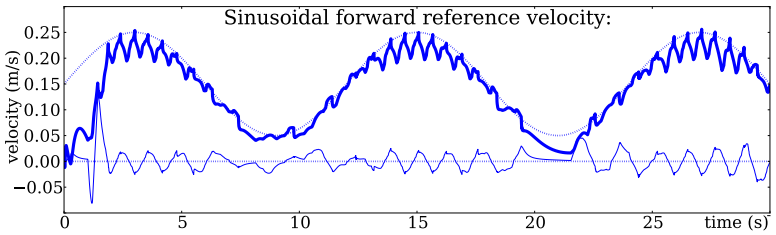

Figure 5.4.: CoM velocity tracking performance of a sinusoidal forward reference velocity. The controller continuously adapts the gait pattern and the CoM trajectory to track the desired CoM velocity. *Thick line:* forward — *thin line:* lateral — *dotted lines:* reference velocities

Performance of the tracking task is depicted in figure 5.4: both forward and lateral CoM velocities tend towards the reference $\dot{\boldsymbol{h}}^r$. However, oscillations are still visible as the result of the alternation between SS and DS phases. Indeed, velocity of the CoM is restrained by the CoP constraints (5.9) its acceleration is subject to, constraints that are all the more restrictive in single support phases. Consequently, velocity tracking is hindered during SS phases, and DS phases *release* DoFs of the CoM, allowing the recovery of the tracking task. Nevertheless, in periods of fast reference velocity $\dot{\boldsymbol{h}}^r$, a new SS phase is rapidly required and the robot do not have time, due to regularizing terms q, to fully reach the required velocity.

This last effect is illustrated in figure 5.5 which displays SS/DS alternations with the average foot velocity during each SS phase.

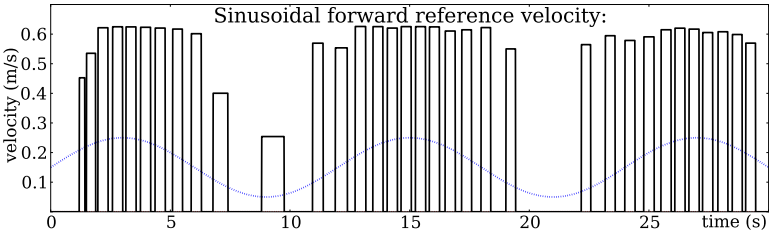

Figure 5.5.: Generated gait pattern from the tracking of a variable CoM velocity. Steps frequency and average velocity is adjusted to the desired CoM velocity. *Solid:* average foot velocity during single support— *dotted:* CoM reference velocities

Indeed, time periods of high reference velocity such as $t = 15s$ require steps of high velocity and frequency, while periods of medium reference velocity such as $t \in [10s, 13s]$ lead to slower, sparser steps and a better CoM velocity tracking.

From a more distant point of view, figure 5.5 exposes an intuitive strategy computed from (5.20): steps taken automatically tend to be longer, faster and more frequent as the desired walking velocity grows higher.

It can be noted that equal weights $\omega_b = \omega_w$ in (5.19) tend to give a higher priority to the walking task, particularly in cases of high velocity tracking. This is mainly due to objectives J_w and J_b not being normalized, hence letting differences in magnitude have an influence.

This effect can be observed in figure 5.6 which presents an extract of CoM, CoP and feet positions. Although balance is still ensured, the tracking task slightly prevents, especially in SS phases, the CoP from staying in the middle of the BoS.

Figure 5.6 shows that the feet tend to be aligned in the forward direction as a result of the null velocity tracking in the lateral direction, in order to minimize the lateral velocity of the CoM during footsteps. However, cases of low forward reference velocities require small steps: the linearized non-overlapping constraint on the feet forces a lateral gap between them, hence inducing a loss of performance of the tracking task in the lateral direction, as shown in figure 5.4 around $t = 22s$

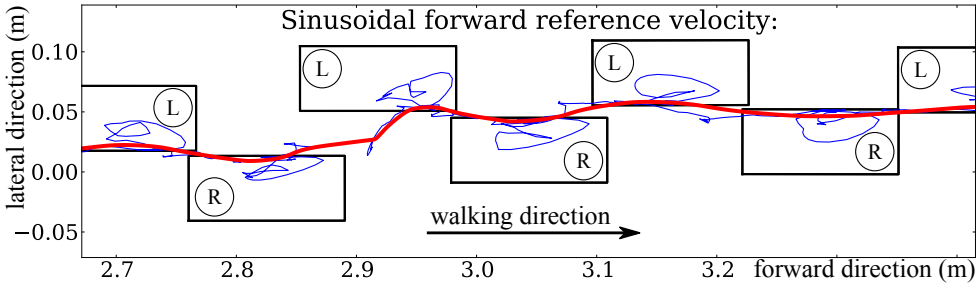

Figure 5.6.: Evolution of the actual CoP and CoM with generated footsteps. Footsteps tend to be aligned to minimize lateral CoM velocity as desired, and are closer to each other in the longitudinal direction for low values of the CoM reference velocity. *Thick line:* CoM position — *thin line:* actual CoP position

for example.

The MIQP controller nonetheless successfully coordinates changes in the BoS with postural adjustments to reach both the walking and balance objectives: an adequate gait pattern is continuously and automatically coordinated with CoM adjustments to concurrently track a CoM velocity while maintaining postural balance.

5.3.2. Activity adjustments: walking under large disturbances

This simulation aims at exhibiting, for an identical set of weights in the cost function (5.19), differences in the optimal behavior against various unknown disturbances.

The humanoid has to perform a walking task with a target forward velocity of $0.20 m.s^{-1}$, and a null desired lateral velocity. Three cases are compared: an unknown lateral impact of $+60N$ (more thant a fifth of the total weight of the robot) is applied to the head of the humanoid during a period of $0.1s$, then no effort is applied to the robot and, last, an opposite impact of $-60N$ is considered. It can be noted that applying the external action to the head brings additional disturbances as large rotational effects are induced, effects which are not taken into account in the ZMP model (5.8). Snapshots of the different cases are provided in figure 5.7.

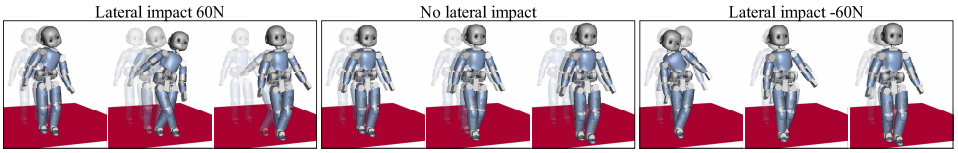

Figure 5.7.: Snapshots of a walking activity under different unknown external impacts, applied to the head of the humanoid. In nominal conditions, a gait pattern is computed to track the desired CoM velocity. The humanoid successfully adapts this nominal gait to recover from unexpected disturbances.

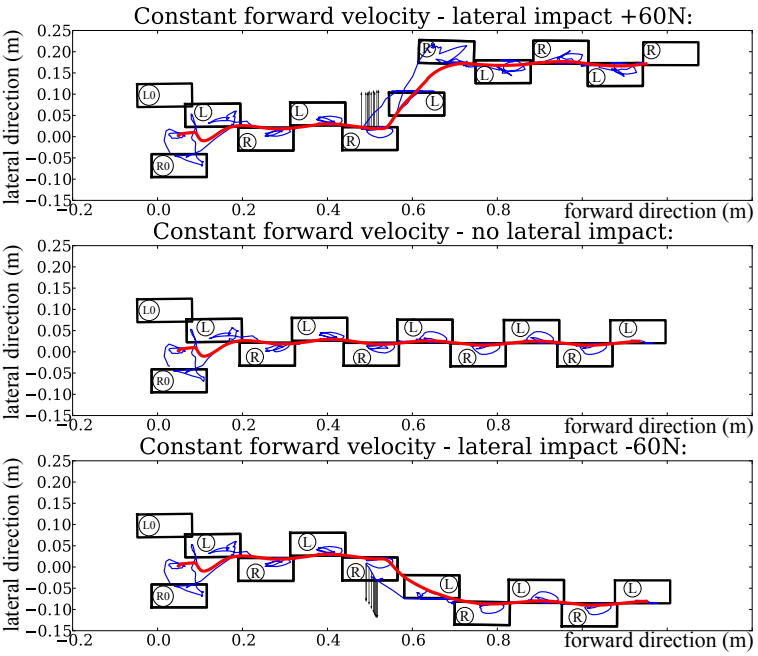

Figure 5.8.: Evolution of the actual CoP and CoM with the adaptation of generated footsteps to an unknown impact. Footsteps are naturally adjusted in the direction of the unexpected disturbance (*black arrows.*) and a nominal gait pattern is ultimately recovered. *Thick line:* CoM position — *thin line:* actual CoP position

Figure 5.8 shows that changes in the system state from the impact lead to an intuitive modification of the steps taken: balance is ensured with a lateral drift of the robot in the direction of the impact.
For the impact of $+60N$, the right foot crosses the left one: such a solution can be excluded by a set of linear constraints[7] in (5.20), or self-collisions may be handled at the whole-body reactive control level.

The impact occurring during an ongoing step, figure 5.9 shows that the speed and duration of this step is slightly altered as a response to the disturbance.

Moreover, a major aspect of the recovery strategy is illustrated: the DS phase following the impact is noticeably shorter than in the undisturbed case. A lateral step is indeed taken almost immediately in order to improve balance.

These modifications in the gait pattern as a response to the impact seem *natural* from the balance point of view. Nevertheless, as illustrated in figure 5.10, although a null lateral velocity obviously cannot be reached during the impact, the desired CoM velocities are rapidly recovered.
Conversely, it can be noted that the tracking task in the forward direction is almost unaffected by the impact. The recovery steps taken are hence compatible with the ongoing walking activity, as objective function J_w is still part of the optimization problem.

[7] *cf.* App. B, p. 175

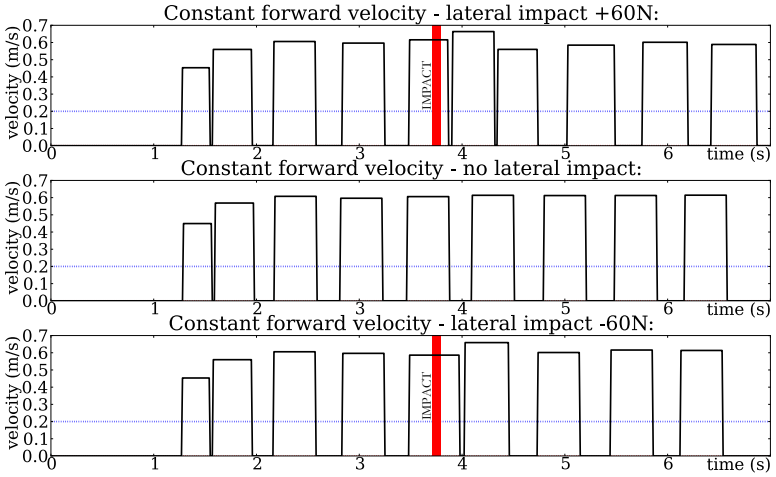

Figure 5.9.: Generated gait pattern from CoM velocity tracking as a response to different impacts. At the unexpected time of impact, a recovery step is inserted within the nominal gait pattern, with high velocity and frequency. The nominal gait pattern is automatically recovered. *Solid:* average foot velocity during single support— *dotted:* CoM reference velocities

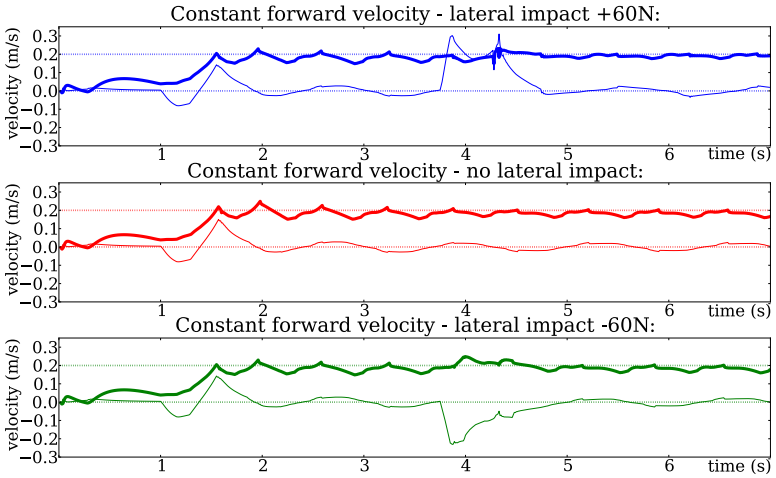

Figure 5.10.: CoM velocity tracking performance against the different impacts while walking. The controller adapts the gait pattern and CoM trajectory to maintain balance while minimizing the tracking error to the reference CoM velocity. Performance in the forward direction is unaltered by the adjustments, and the lateral tracking objective is temporarily dropped to recover balance. *Thick line:* forward — *thin line:* lateral — *dotted lines:* reference velocities

These results therefore further validate the contribution of the multi-objective predictive level: objectives can be temporarily relaxed to guarantee the respect of constraints. In this scenario, the walking objective is indeed momentarily sidelined to allow the system to recover from the unexpected disturbance. A nominal gait pattern is rapidly and automatically restored once the conditions allow it.

5.3.3. Behavior specification: push-recovery

A last simulation scenario demonstrates how the balance behavior of the humanoid can be specified with the sole ratio ω_w/ω_b. Differences in behavior are illustrated in a push recovery case. The humanoid must stand still, *i.e.* $\dot{\boldsymbol{h}}^r = \mathbf{0}$, and experiences an unknown impact of $50N$ in the forward direction and $-25N$ in the lateral one, during a period of $0.1s$.

The standing objective is incrementally relaxed in three cases. A first set of weights targets the best activity performance with $\omega_w/\omega_b = 5.0$, a second is placed as reference with $\omega_w/\omega_b = 1.0$ and a last set gives a higher priority to the balance objective J_b with $\omega_w/\omega_b = 0.5$. The difference in the resulting behavior of the humanoid is illustrated in figure 5.11 for the three cases.

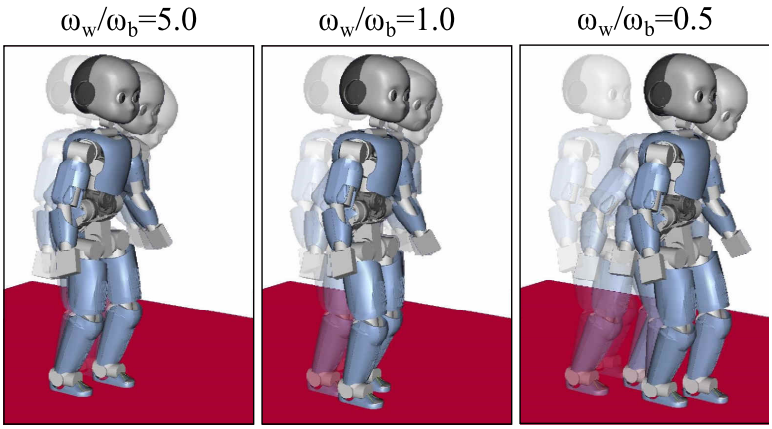

Figure 5.11.: Influence of the relative weights of the walking and balance objectives ω_w and ω_b, respectively, in a push-recovery scenario. The strategy of recovery steps is strongly influenced by the ratio ω_w/ω_b.

As expected, results provided in figure 5.12 show that priority of the tracking objective J_w, relatively to J_b, has a direct influence on the evolution of the CoM velocity. Although the humanoid eventually reaches the null reference velocity in all cases, control with $\omega_w/\omega_b = 5.0$ attains the standstill objective around three times faster than with $\omega_w/\omega_b = 0.5$.

This gain in performance is achieved with a noticeably different gait pattern for the three cases. Indeed, as shown in figure 5.13 the first controller stops the CoM with one step, the second with two and the third recovers a standstill state after three steps.

It can be observed that the duration of the first recovery step grows with the decrease of the ratio ω_w/ω_b, which can be interpreted as the concurrence of two effects. During SS phases, the dynamics of the CoM are conditioned by the CoP constraint (5.9), hence potentially affecting the tracking objective J_w as shown in figure 5.12; on the other hand, a longer step duration allows to place the BoS as desired, in favor of the balance objective J_b. Essentially, DS phases provides the CoM with a greater degree of freedom, while SS phases are required to adapt the BoS in regard to the balance objective.

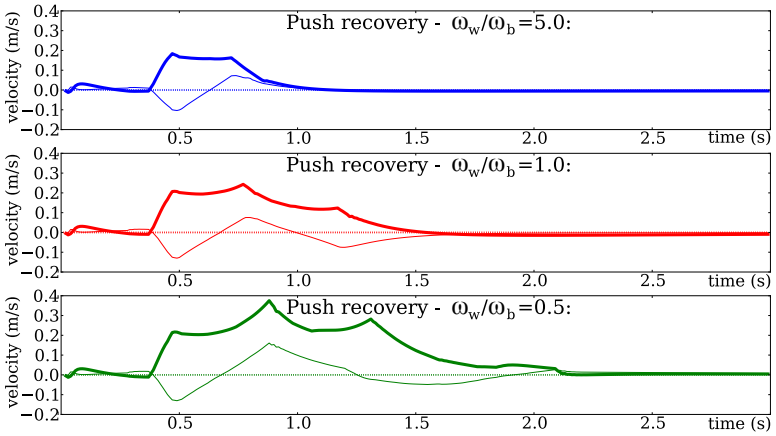

Figure 5.12.: Influence of the objective weights on the CoM velocity tracking performance. Reducing the relative influence of the walking objective leads to a greater error in the walking task, as expected.
Thick line: forward — *thin line:* lateral — *dotted lines:* reference velocities

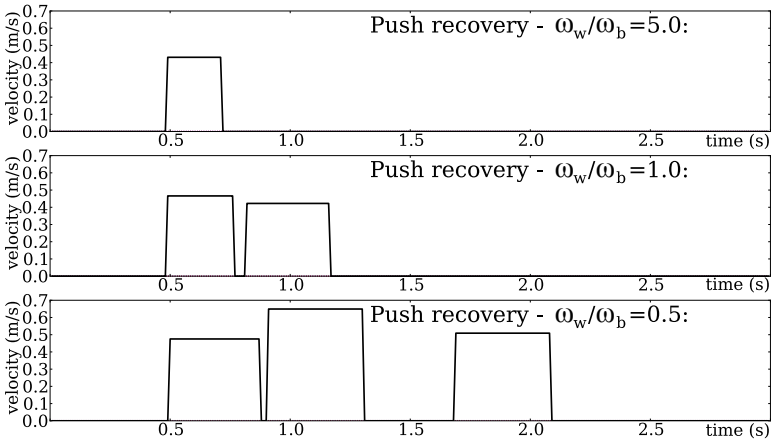

Figure 5.13.: Generated gait pattern from CoM velocity tracking for different objective weights in a push recovery scenario. The sole relative influence of the walking objective significantly affects the recovery gait pattern.
Solid: average foot velocity during single support

Figure 5.14 indeed confirms these remarks. The first recovery step tends to be further away from the initial feet configuration as the ratio ω_w/ω_b decreases, which leads, in average, to a better placement of the CoP with respect to the BoS.

Further insight is provided through this scenario on the influence of relative weights of the sub-objectives in their centralized, multi-objective coordination problem.
These weights distinctly specify the resulting behavior of the system: providing a lower priority to the standstill objective indeed tends to allow a greater number of recovery steps to be taken. This result is one of the contribution of the optimization-form of the control problem. Behaviors are defined as an organized set of objectives to reach, and modifications within this set directly influence the

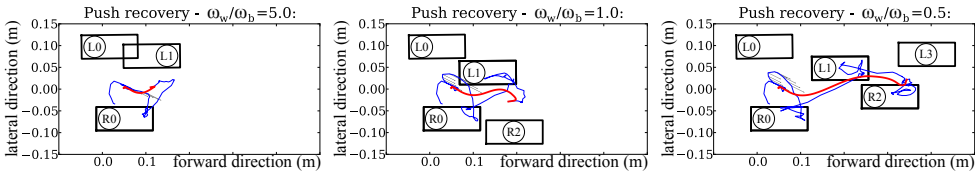

Figure 5.14.: Evolution of the actual CoP and CoM with generated footsteps for an identical impact and different objective weights (ω_b, ω_w). The variation of the relative influence of the walking task over balance induces diverse push-recovery strategies. *Thick line:* CoM position — *thin line:* actual CoP position

resulting motion. Activities are thus defined at a relatively high level, therefore preventing an excessive specification of the expected behavior.

The weighted specification of priorities might however raise a concern for safety, as strict hierarchies are not enforced. Nevertheless, strict constraints are enforced in the optimization problem. The influence of weights is moreover reduced by the predictive formulation of the control problem. As demonstrated in these results, the standstill objective is sidelined in the three cases to guarantee postural stability despite increases in its priority.

5.3.4. Remarks and discussions

As observed in figures 5.6, 5.8 and 5.14, the CoP happens to reach the edges of the BoS, thus putting the system at tip-over risk although constraints on the CoP (5.9) are specified with a safety margin of $\approx 30\%$, and are guaranteed in problem (5.20). The ZMP model (5.8) used as an approximation of the CoP ignores, among others, rotational effects. However such effects may yet be of large magnitude in the cases studied in this section: fast reference velocities and sudden disturbances are considered, leading to rapid steps and, as a consequence, fast movements of the swinging leg. Moreover, the application of the external force to the head of the robot induces rotational effects from the entire upper body of the humanoid (*cf.* figure 5.7), thus increasing the gap between the actual CoP and estimated ZMP (5.8).

It can be noted that this gap might be partially reduced by setting a lower maximal foot velocity. Additionally, a reactive primitive can be introduced at the instantaneous level to minimize the rate of change of angular momentum of the system, therefore reducing the gap between the actual CoP and estimated ZMP.

Another approximation regarding the CoP is the overestimation of the convex hull of the contact points by its bounding box, in order to write linear constraints with respect to state $\boldsymbol{\xi}$. Nevertheless, outputs from the MIQP can be validated at each control step to ensure that the previewed CoP remains inside the convex hull of the BoS.

If not, a fast QP program as described in [Wieber2006] can be set up to recompute a valid CoM trajectory (possibly with new feet positions as in [Herdt2010]), taking as input the changes in BoS provided by the MPC problem (5.20). Such

a safety setup was not necessary in the presented simulations. Indeed, this overestimation solely occurs on a previewed BoS configuration for a DS phase ; the current BoS convex hull is fully considered as explained in Sec. 5.1.2, and both current and future BoS for SS phases are accurately represented. Therefore, the output desired CoM state is uniquely potentially inadmissible at the sole instant when switching from a SS phase to a DS phase.

Results introduced in this section were obtained with the GUROBI OPTIMIZER [Gurobi]. A non uniform time sampling scale was used, in order to keep a reduced size of the optimization vector while avoiding the accumulation of integration errors in the preview. Furthermore, variables associated to changes in the BoS were sampled at a lower frequency than variables describing the CoM dynamics. Indeed, the complexity of the MIQP (5.20) is strongly related to the number of integer variables, and it can arbitrarily be considered that changes in the BoS occur at a lower frequency than CoM adjustments.

Solving the MIQP took an average of $40ms$[8], and note that since MIQP solvers generally employ branch-and-bound based algorithms, computation time can largely be influenced by the order of variables and the use of heuristics, helping at reaching the best nodes faster.

Lastly, it can be noted that the model of the BoS presented in Sec. 5.1 omits DoFs from the rotations of the feet. Nevertheless, while orienting the BoS in the motion or disturbance direction is beneficial to balance as it exploits the generally anisotropic shape of the feet, their consideration as optimization variables forbids the formulation of a linear model, due to CoP constraints (5.9). However, predefined foot rotations can be accounted for as explained[9] in Sec. 6.2.2.2.

These simulation results further demonstrate the contribution of predictive coordination to the generation of safe and complex behaviors. Discrete events are considered and adjusted at the control level instead of being independently specified from a decision layer, and their effect on the system state with respect to its activity are anticipated to compute appropriate postural adjustments.
The resulting strategies are bilateral: continuous adjustments are coordinated with respect to the constraints induced by discrete changes in the BoS, and conversely.

More elaborate stepping strategies can additionally be observed in scenarii under complex disturbances, as exhibited in the multimedia attachment[10] of [Ibanez2014b], where an iCub robot successfully recovers balance, in simulation, from disturbances induced by continuous and unexpected horizontal translations of the ground.

[8] on a dual-core Intel® Core i7®-2620M for 74 DoFs and 275 constraints in average.
[9] *cf.* p. 156
[10] `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6943127`

5.4. Conclusion

A multi-objective predictive control level is employed in this chapter to compute optimal coordination strategies between discrete changes in the base of support and continuous postural adjustments of a humanoid system, with respect to walking and balance objectives.

Simulation results demonstrate a vast range of resulting behaviors of the robot, for a single specification of the activity and under various and disturbed conditions. These motions imply elaborate coordination strategies not only between discrete and continuous degrees of freedom, but furthermore between objectives and constraints the system is subject to. The resulting complex behaviors are **emerging** from the multi-objective predictive formulation of the coordination problem, and the **evolution** of the conditions of the system within its environment: **no explicit specification** on the expected motion is performed.

This formulation of a multi-objective predictive level moreover demonstrates how optimal **logical actions**, generally handled at the decision level, can be computed at the **control level**.

This multi-objective predictive level exploits the Mixed-Integer Programming framework to capture discrete events in the optimization problem and consider non-convex admissible domains of the solutions.

A mixed-integer, linearly constrained description of the dynamics model of the system and its constraints is first performed, capturing balance dynamics through the CoM motion constrained by the configuration of the contacts with the ground.

A balance objective is introduced as a tracking task on the CoP, conjointly with a walking objective aiming at reaching a desired CoM motion. These concurrent objectives are therefore conflicting, both being defined with respect to the CoM dynamics.

Simulation results nevertheless validate the ability of the henceforth formulated multi-objective predictive problem to find coordination strategies allowing to reach both of these objectives. The specification of these strategies being performed at a high-level with the sole definition of objectives to reach under constraints, a **wide span of system behaviors** is observed: while the balance objective is temporarily sidelined, although balance constraints are still enforced, to allow the tracking of a desired CoM velocity in nominal walking activities, the system recovers from unexpected disturbances with **momentary relaxations** of the walking objective.

* * *

Despite the relative computational efficiency obtained through the use of linear constraint and the formulation of a quadratic optimization problem, the order of magnitude of computation times is challenging the implementation of this Mixed-Integer predictive controller. The following chapter however takes on this challenge with the setup of a multi-rate, two-layered control architecture for a real-time implementation of this controller on a real humanoid robot.

Real-life applications can also benefit from simulation-time implementations: applications to the control of virtual humans are indeed furthermore envisioned to take advantage from the robustness of the coordinated walking behaviors and their wide range.

CHAPTER 6

Applications of the multi-objective MIQP controller

This chapter introduces application cases of the two-layered, multi-objective predictive control control architecture developed in Chapter 5.

The computational cost of complex predictive control approaches is one of the major obstacles to their dissemination to control architectures of real systems. Employing favorable problem formulations nevertheless allows to envision multi-rate control architectures, predictive models of increased complexity being henceforth solved at a lower rate than reduced MPC problems.

Such architectures are therefore three-layered. At the higher level, complex models are employed to preview computationally-demanding coordination strategies. A second predictive layer is executed at a faster rate to perform anticipated adjustments on these strategies. A reactive control layer last employs a more accurate model of the system and its constraints to coordinate at the instantaneous level the whole-body motion in an admissible manner.
Multi-rate setups however raise multiple challenges: delays are inserted in the control problem, and control stability is threatened.

Nevertheless, real-life applications do not necessarily rely on real-time implementations, simulation tools being widely employed, for example in computer graphics, computer-aided design and motor analysis applications. Virtual human models are increasingly used within this range, for character animation, ergonomic assessment and captured motion reproduction for example. In these applications, physical models are generally employed more often than cognitive models, the definition of *human-like motions* being more accessible than the reach of *human-like decisions*.

While their growing interest even leads to the establishment of specific standards such as [Afnor2008], the use of physical human models is nevertheless still challenging. Two major challenges can indeed be identified. At a higher level, the generation of automatic behaviors is generally required to enable the manikin to evolve as autonomously as possible. Second, these behaviors should not only be *realistic*, they must be *feasible*: dynamically-consistent motions are therefore expected.

This chapter henceforth proposes applications of the multi-objective predictive controller introduced in Chapter 5 to take on part of these challenges.

A real-time implementation of the computationally-demanding MIQP controller is first performed on the humanoid robot TORO, through the setup of a three-layered, multi-rate predictive control architecture. Despite challenges issues from inconsistencies between the various models employed at the different rates, the resulting experimental results demonstrate the feasibility of such implementations, and the MIQP controller effectively generate walking behaviors adapting to variations in objectives, constraints and disturbances.

Second, this controller is envisaged in simulation with the control problem of virtual humans, allowing to compute dynamically-consistent motor activity with and without captured human motion as inputs. The robustness and variety of motions obtained with this control architecture motivates its application to be

considered in the reproduction of partially captured human motion within an activity recognition perspective.

6.1. Real-time implementation on TORO

The computation time of the MIQP controller observed[1] for preview horizons of sufficient length and accuracy makes impossible a straightforward implementation on a real humanoid system, which generally requires a much faster motor control rate.

The compatibility of the computationally-demanding MIQP problem (5.20) with real-time requirements is nevertheless assessed in this section, through its implementation on the TORO humanoid robot. The TOrque controlled humanoid RObot (TORO) is a full humanoid robot developed at the German Aerospace Center (Deutsches Zentrum für Luft und Raumfahrt, DLR) and is presented in figure 6.1, with as main characteristics its 25 torque-controlled joints, $174cm$ height and a mass of $76.4kg$. A 6-DoF FTS sensor located at each ankle allows to measure the ground reaction wrenches at each foot. The main control loop is operating at $1kHz$ supported by an Intel® Core i7® processor located in the backpack of the robot, running a Real-Time Linux distribution based on Kernel 3.0. For additional specifications, the reader is referred to [Englsberger2014].

This section first addresses the challenge of interfacing a slower control task with the whole-body controller of TORO. A multi-rate control architecture is therefore setup with the addition of an intermediate, computationally-efficient MPC controller. Experimental results are then studied to gain qualitative insight on the issues raised by such architectures and solutions are envisioned to take on these challenges.

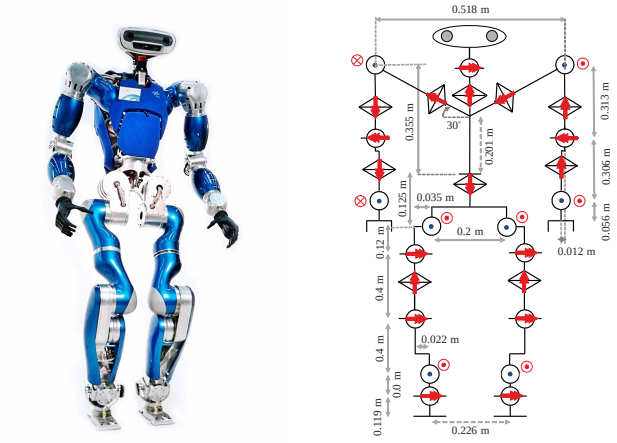

Figure 6.1.: Overview of the humanoid robot TORO (2014), from the German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt, DLR). *Courtesy of Institut für Robotik und Mechatronik, DLR and [Englsberger2014].*

[1] an average computation time of $40ms$ is necessary to obtain the results of Sec. 5.3, p. 117

6.1.1. Multi-rate control setup

The whole-body control loop of the TORO robot running at $1kHz$, a direct implementation of the MIQP controller is impossible with the current problem formulation and solver. However, while postural balance requires fast motor adjustments, the gait adaptation rate is assumed to be less critical. Shifts in the contact state are indeed not only slower in essence as they involve motion of the feet, they also induce greater changes in the balance problem. Conversely to adjustments which imply fast responses, step adaptations can thus be seen as measures of a higher-level and greater timescale.
The encapsulation of the MIQP controller in a higher-level control loop is therefore envisioned to output at a slower rate optimal gait parameters and their corresponding CoM/ZMP trajectories to the whole-body controller.

6.1.1.1. Real-time multi-rate architecture

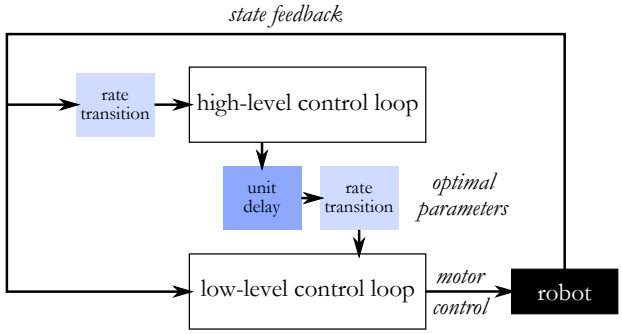

Figure 6.2.: Overview of the multi-rate control architecture. The MIQP controller is encapsulated in a high-level control loop with a sampling period of $100ms$. Optimal gait parameters are outputted to a $1kHz$-control loop producing joint control outputs.

The control architecture illustrated in figure 6.2 is considered in this implementation. Accounting for the order of magnitude of the MIQP solving duration, a high-level control loop with a sampling period $T^* = 100ms$ is interfaced with a faster control loop running at $1kHz$ (sampling period $T = 1ms$).
The presence of a unit delay before the slower-to-faster rate transition in figure 6.2 ensures real-time deterministic results by forcing the faster task to wait for a delayed update from the slower task. This deterministic behavior is a requirement for a safe real-time implementation, and consists in the process described in figure 6.3. The unit delay runs at rate R, and uses the output of the slow task (of equal rate R) to update its internal state. The rate transition from the slower rate R to the faster rate r is a zero-order hold, and thus holds the internal state of the delay as its output at rate r, which is used by the fast task until a new update is performed. The resulting behavior is therefore independent from the duration of the slow task, assuming that it stays below its sampling period. However, this delay greatly influences the behavior of the process, and its effects are discussed later in this section.

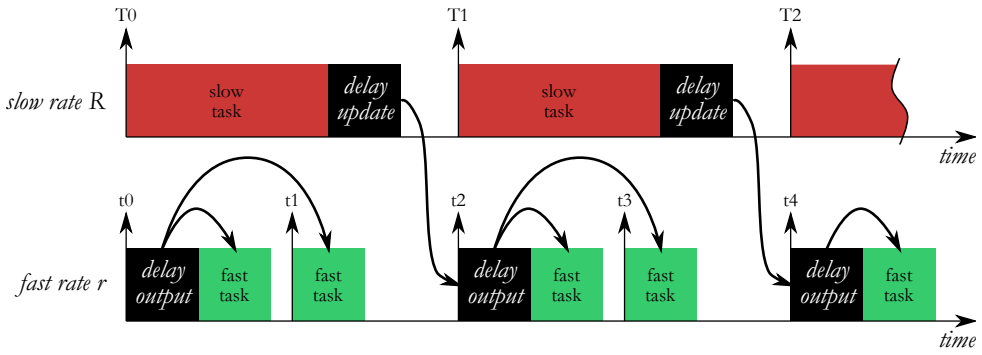

Figure 6.3.: Illustration of the slower-to-faster rate transition. To ensure deterministic results, independent from the duration of the slow task, a unit delay is inserted which holds its output, fed when required to the fast task.

6.1.1.2. High-level to low-level interface

To ensure the feasibility of solving the MIQP problem (5.20) within its sampling period $T^* = 100ms$, the dimensions of the preview vectors are required to be kept to the lowest admissible. To do so, both the duration and sampling period of the preview horizon are adjusted. With orders of magnitude around 10 steps over an horizon of $1.5s$, the preview sampling period is significantly greater than the low-level control period $T = 1ms$. The approximation of the CoM dynamics by the linear preview equation (5.15) is thus eminently distant from the low-level control model.
While this approximate time-discretization of gait parameters is acceptable for non-dynamics related outputs such as footsteps duration and position, CoM-related — and, consequently, ZMP-related — outputs of the MIQP problem can therefore not be directly considered as instantaneous desired values. The optimal, rough CoM and ZMP trajectories resulting from the MIQP problem can nevertheless be considered as reference inputs to a finer MPC problem., which in turn computes — at a faster rate — optimal ZMP positions over a refined preview horizon.

Since the original resolution of gait parameters is assumed to be sufficient, the optimization problem of this envisioned faster MPC controller should consider dynamics-related variables solely. Classical ZMP preview controller formulations such as [Kajita2003] and [Wieber2006] are potential candidates as their implementation at fast rates has been widely demonstrated. However, to maintain consistency of the solution with the formulation of the walking activity objective in the MIQP problem (5.20) and therefore with the optimal gait parameters, a sole ZMP tracking error objective is insufficient. Indeed, the CoM and ZMP trajectories outputted from the MIQP problem are optimal with respect to both ZMP and CoM tracking objectives.

The ZMP Preview Control formulation described in Sec. 2.4.3 is thus extended to account for a CoM tracking objective, and more precisely to minimize the error over a preview horizon to reference CoM velocities. This choice echoes the formulation of the walking task considered in the simulation results presented in Sec. 5.3, where the walking objective is written as a desired CoM velocity

to track. The extended ZMP preview controller hence solves, with $\boldsymbol{\mathcal{U}}_{k+N|k} \triangleq \left\{\boldsymbol{u}_{k|k}, \ldots, \boldsymbol{u}_{k+N-1|k}\right\}$ denoting an horizon of input CoM jerks $\boldsymbol{u}$

$$\min_{\boldsymbol{\mathcal{U}}_{k+N|k}} \sum_{j=1}^{N} \eta_b \left\|\boldsymbol{p}_{k+j|k} - \tilde{\boldsymbol{r}}^r_{k+j|k}\right\|^2 + \eta_w \left\|\dot{\boldsymbol{h}}_{k+j|k} - \dot{\tilde{\boldsymbol{h}}}^{\,r}_{\,k+j|k}\right\|^2 + \eta_r \left\|\boldsymbol{u}_{k+j|k}\right\|^2$$
$$\text{s.t.} \quad \begin{cases} (5.8), \\ (5.13), \\ \boldsymbol{h}_{k+j|k} = \boldsymbol{h}_k, \end{cases} \tag{6.1}$$

where η_b, η_w and η_r are scalar weights defining the relative prevalence of the balance, walking and regularization objectives respectively. Values $\tilde{\boldsymbol{r}}^r$ and $\dot{\tilde{\boldsymbol{h}}}^{\,r}$ are respectively the reference ZMP position and CoM velocity, interpolated from the outputs of the MIQP controller. This problem is an unconstrained QP, and a closed-form solution allows its execution at the $1kHz$ low-level control rate.

The objective $\|\boldsymbol{u}\|^2$ being introduced for regularization purposes solely, the MPC problem (6.1) is therefore compromising between balance and walking objectives, compromise defined by the relative values of η_b and η_w. Equations (5.8) and (5.13) allow to deduce an instantaneous desired ZMP position from the optimal horizon $\boldsymbol{\mathcal{U}}^*_{k+N|k}$ solving problem (6.1). This desired ZMP position is outputted to a position-based ZMP controller, computing desired CoM position and velocity to minimize the instantaneous ZMP tracking error. Details on the formulation of this position-based ZMP controller can be found in [Krause2012].
Simultaneously, gait-related outputs $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \delta, \gamma)$ of the MIQP problem are translated into step target position and duration if an SS phase is ongoing or required. These parameters are fed to a feet controller which computes the adequate desired feet position and velocity, and an inverse kinematics solver finally outputs the corresponding desired joint positions to the robot.
The overall control architecture is illustrated in figure 6.4.

6.1.1.3. Synchronization and interpolation

As the MIQP controller outputs discrete events and due to the presence of the unit delay, the first element of its previewed solution is required to be synchronized with its control period T^*, as illustrated in figure 6.5. Indeed, an optimal horizon of solutions previewed from the state of the system at time t is not available until $t + T^*$ (*cf.* figure 6.3). This setup implies that if a decision to initiate or end a SS phase occurs, it can only be applied after T^* is elapsed; similarly, all outputs computed from time t for previewed instants lower than $t + T^*$ are unusable. Therefore, the first instant of the preview horizon at time t must be greater or equal to $t + T^*$, and the earliest availability of the control output decision then imposes that this first instant exactly concurs with $t + T^*$.
Moreover, the MPC paradigm suggests that only the first previewed output — the closest in time to the input state — should be applied to the system. Since a new solution will only be available at $t + 2T^*$, a single output solely should be considered between $t + T^*$ and $t + 2T^*$.[2] To account for this behavior during the

[2] Note that this remark is only valid for discrete outputs. Indeed, continuous outputs are refined at the lower-level by the extended ZMP preview controller.

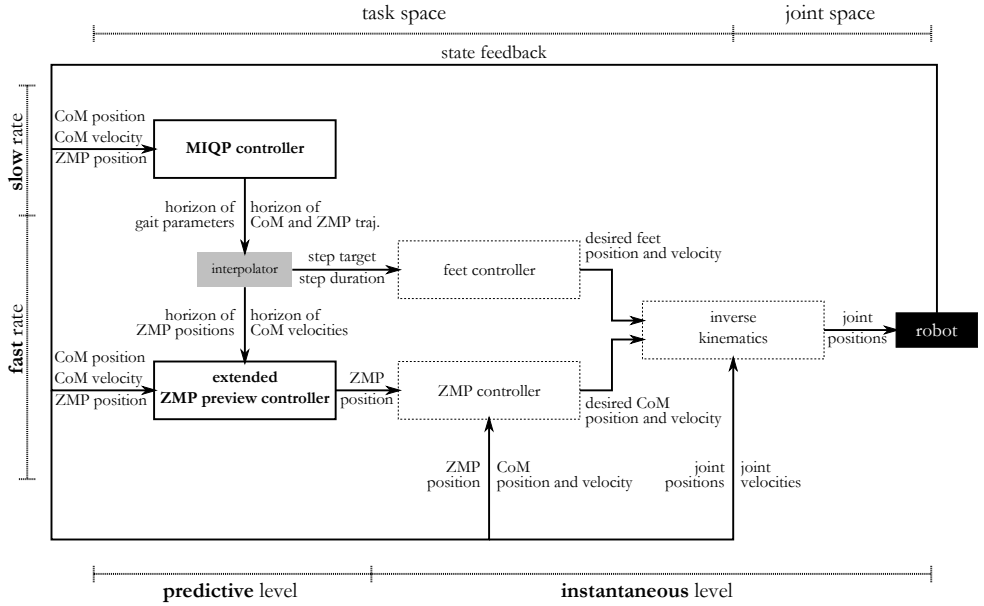

Figure 6.4.: Detailed multi-rate control architecture. The slow-rate output of the MIQP controller is interpolated and feeds an extended ZMP preview controller. Step target position and duration, along with instantaneous desired ZMP position are outputted to the existing whole-body position-based controller which computes the corresponding joint positions.

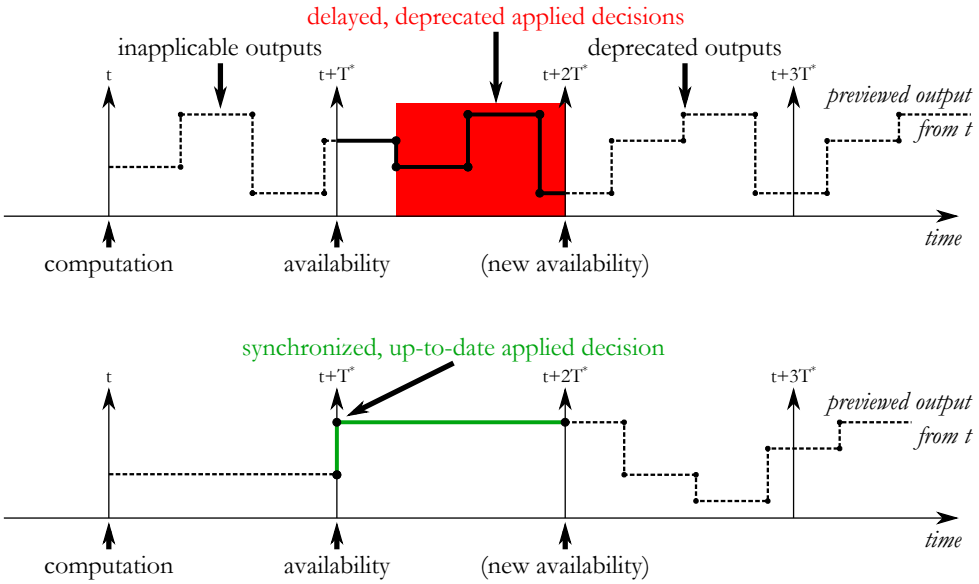

Figure 6.5.: Illustration of the synchronization requirements of the preview horizon with the sampling time for discrete events. *From top to bottom* — incorrect preview horizon discretization, synchronized preview horizon.

preview, the second instant of the preview horizon at time t must consequently be greater or equal to $t + 2T^*$. An illustration of these remarks is provided in figure 6.5.

As previously introduced, the extended ZMP preview controller relies on interpolated values $\tilde{\boldsymbol{r}}^r$ and $\dot{\tilde{\boldsymbol{h}}}^r$ of MIQP outputs. Horizon $\dot{\tilde{\boldsymbol{h}}}^r$ is directly obtained from the linear interpolation of the integrated MIQP outputs $\boldsymbol{u}$ (*cf.* equation (5.13)). The horizon of reference ZMP positions $\tilde{\boldsymbol{r}}^r$ is defined as being the center of the feet in contact with the ground, accordingly to the horizon of MIQP outputs. By nature, $\tilde{\boldsymbol{r}}^r$ is piecewise constant as changes in the BoS are discrete. Values $\tilde{\boldsymbol{r}}^r$ should thus be interpolated from outputs $(\boldsymbol{a}, \boldsymbol{b})$ while maintaining the piecewise-constant characteristic. With $interp_{pw}$ such interpolation process, and $interp_{pw}(\bullet, t)$ the interpolated value of $\bullet$ at time t, it should write

$$\tilde{\boldsymbol{r}}^r_{kT^*+iT}(t) \triangleq interp_{pw}\left(\frac{\boldsymbol{a}_{kT^*} + \boldsymbol{b}_{kT^*}}{2}, t\right), (k,i) \in \mathbb{N}^2, iT \leq T^*,$$

which reads that the interpolated value $\tilde{\boldsymbol{r}}^r_{kT^*+iT}(t)$ at instant $kT^* + iT$ (between kT^* and $(k+1)T^*$) for time t is deduced from the piecewise constant interpolation of outputs $(\boldsymbol{a}, \boldsymbol{b})$ available[3] at instant kT^*. However, due to the rough time-discretization of the MIQP preview horizon imposed by the previous synchronization requirements and the large rate difference between MIQP and low-level updates, jumps in $\tilde{\boldsymbol{r}}^r$ are expected to occur at each update of the MIQP output. To smooth these changes in reference $\tilde{\boldsymbol{r}}^r$, a linear interpolation between kT^* and $(k-1)T^*$ is performed while ensuring a piecewise-constant form of the reference. This linear interpolation can be outlined as the following formulation

$$\begin{aligned}\tilde{\boldsymbol{r}}^r_{kT^*+iT}(t) \triangleq \quad & (1 - iT/T^*)\, interp_{pw}\left(\tfrac{\boldsymbol{a}_{(k-1)T^*} + \boldsymbol{b}_{(k-1)T^*}}{2}, t\right) \\ & + (iT/T^*)\, interp_{pw}\left(\tfrac{\boldsymbol{a}_{kT^*} + \boldsymbol{b}_{kT^*}}{2}, t\right), (k,i) \in \mathbb{N}^2, iT \leq T^*.\end{aligned}$$

This linear interpolation combines the previous solution to the current one and preserves the piecewise constancy of both solutions.
A similar smoothing of the changes in step target duration and position is performed to the inputs of the feet controller, using a cosine interpolation to avoid discontinuities.

6.1.2. Parameterization and experimental setup

The validity of the multi-rate implementation of the MIQP controller is assessed in three scenarii. An identical parameterization of the controller is held through the three cases to demonstrate its generic characteristic and its ability to generate various behaviors depending on the objectives and disturbances.

The high-level controller is setup with a preview horizon of $1.5s$, discretized in 6 samples for integer variables and 8 for real-valued ones. Balance and walking objectives have a relative weight ratio of $\omega_w/\omega_b = 0.5$, and maximum step span is constrained within $[0.0cm, 0.1cm]$ and $[0.22m, 0.27m]$ in the longitudinal and

[3] these values, however, were computed from instant $(k-1)T^*$

lateral directions, respectively. The lower bounds of these constraints are introduced in order to avoid collisions of the knees and feet, while the upper bounds implicitly forbid singular configurations[4] of the legs. SS and DS phases durations are constrained to be greater than $0.7s$ and $0.3s$, respectively. Note that this set of constraints limits the reachable global velocity of the robot to a maximum of $0.1m/s$. A safety margin of $\approx 35\%$ on the CoP constraints (5.9) is additionally introduced to consider a smaller admissible BoS in the MIQP problem (5.20).

The high-level MIQP controller is initially replaced by predefined gait parameters to assess the performance of the low-level, extended ZMP preview controller. The high-level controller is then employed in three scenarii.

First, the walking objective is aiming at a varying target CoM velocity $\dot{\boldsymbol{h}}^r$. This velocity is set to null in the lateral direction, and successively takes the values $0.1m/s$, $0m/s$ and $0.1m/s$ in the longitudinal direction.

Second, the robot tracks a $0.1m/s$ forward velocity and the walking motion is disturbed by manually maintaining the robot in position.

A last scenario defines a standstill activity as $\dot{\boldsymbol{h}}^r = \mathbf{0}$, and unexpected external pushes and pulls are applied to the back of the robot.

For the three cases, the velocity objective is heterogeneously weighted with a 10^{-2} weight ratio between the lateral and longitudinal directions in the matrix $\mathbf{S}$ of the walking cost function[5] (5.17), the walking activity mainly consisting in the definition of a forward velocity to reach, and the lateral motion of the system being compulsory from the collision avoidance requirements of the legs and feet.

6.1.3. Low-level, extended ZMP preview controller performance

The extended ZMP preview controller is setup with a preview horizon of $2.0s$, discretized in 200 samples. The objective function is weighted with $\eta_b = 1.0$, $\eta_w = 10^{-3}$ and $\eta_r = 10^{-7}$ for the balance, walking and regularization objectives respectively.

It can be noted that the preview horizon duration and the relative weighting η_w/η_b is inconsistent with the parameterization of the MIQP controller. The number of samples over the preview horizon is tuned to take as few computational resources as possible, this controller being executed at $1kHz$. The resulting sampling period of the preview horizon, along with the relative values of η_b, η_w and η_r, have however a significant influence on the convexity of the QP problem (6.1). The horizon duration was therefore concurrently tuned with the objective weights to achieve the best performance of this low-level controller, and parameters consistency with the MIQP setup was thus sidelined.
Moreover, strict consistency of the balance and objective weights ratio is irrelevant: the optimization variables of the QP and MIQP controllers are significantly distinct in number and nature.

[4] this consideration is induced by the use of a position-based control framework relying on inverse kinematics

[5] *cf.* Sec. 5.2.2, p. 115

For the ZMP tracking performance evaluation, the velocity reference $\dot{\tilde{\boldsymbol{h}}}^r$ of the low-level MPC problem (6.1) is set to null. A predefined gait pattern with steps of $0.1m$ length and total duration of $1.2s$ with SS phases of $0.7s$ is inputted to the extended ZMP preview controller and feet controller.

Tracking performance is depicted on figure 6.6 with the evolution of the measured CoP position, computed from the FTS sensors at the feet, alongside with the reference ZMP position $\tilde{\boldsymbol{r}}^r$ located at the center of the feet and the BoS boundaries.

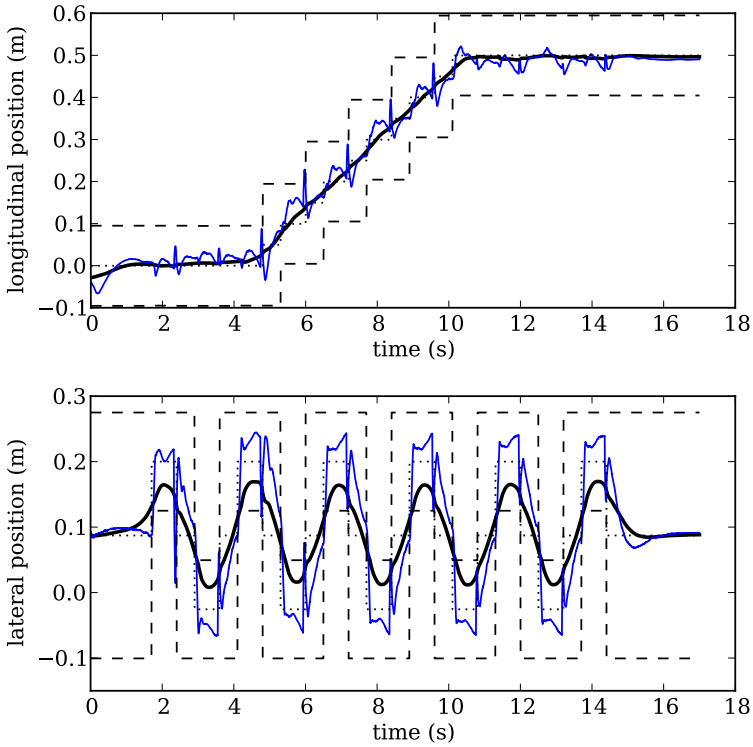

Figure 6.6.: Tracking performance of the extended ZMP preview controller. *From top to bottom* — longitudinal direction, lateral direction. *Thin solid blue: measured CoP position — thick solid black: CoM position — dotted blue: ZMP reference — dashed black: actual BoS boundaries.*

From an overall point of view, the extended ZMP preview controller allows to maintain the CoP within a region spanning around 50% of the entire area of support thus effectively avoiding tip-over risk. Overshoots are significantly visible in SS phases, and are nevertheless of an acceptable amplitude with respect to the BoS span. Short-term peaks in the tracking error can additionally be observed during SS→DS transitions, resulting from the impact of the foot with the ground. Related variations are indeed observed on the evolution of the total ground reaction force amplitude as shown in figure 6.7.
These transitional effects can be interpreted as the result from the contact configuration shifts on the ground reaction force instantaneous distribution (and

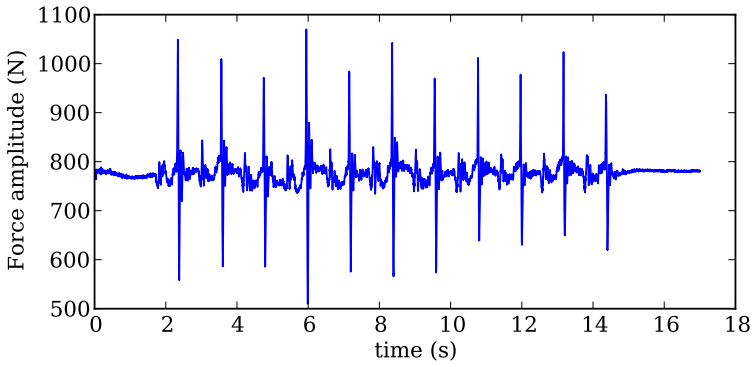

Figure 6.7.: Total ground reaction force amplitude. Variations of significant amplitude are observed during SS↔DS transitions.

therefore on the CoP position), and conversely from the reaction of the extended ZMP preview controller to these sudden changes.

It can be noted that these effects are in a direction compatible with the instantaneous direction of the CoM; therefore, removing DS phases would potentially allow to exploit these variations by synchronizing desired shifts in the ZMP reference with the instantaneous changes in the CoP position. However, DS phases are required by the formulation of the MIQP walking problem formulation, which is able to employ DS phases to delay or ignore stepping.

In the rest of this section, this extended ZMP preview controller takes as inputs the interpolated outputs of the high-level, slow-rate MIQP controller.

6.1.4. Adapting constraints to objectives: varying target velocity

The validity of the MIQP formulation (5.20) in the automatic generation of gait parameters and CoM trajectories to satisfy balance and walking objectives is assessed in a simple walking scenario, aiming at a varying target velocity of the CoM. This target is manually switched between $0.0m/s$ and $0.1m/s$ values. Such changes being unexpected, the horizon of references $\dot{\boldsymbol{h}}^r$ in the objective function of MIQP problem (5.20) is assumed to be constant over the preview horizon, and thus no anticipation on the variations of the reference velocity can be performed.

Snapshots of the resulting behavior of the robot are provided in figure 6.8. When the target $\dot{\boldsymbol{h}}^r$ is increased from $0.0m/s$ to $0.1m/s$, footsteps are automatically and continuously generated to reach this velocity. After reverting to a null velocity, the robot naturally stops to achieve the standstill objective, and footsteps are generated again when the target velocity is once more set to $0.1m/s$. Constraints on the CoP, defined by the contact configuration, are thus adapted with respect to the walking objective.

However, as illustrated in figure 6.9, this objective is not fully reached, despite being feasible[6]. The evolution of the CoM velocity shows that the overall varia-

[6] constraints on the problem are indeed set up to allow a maximum forward CoM velocity of $0.1m/s$

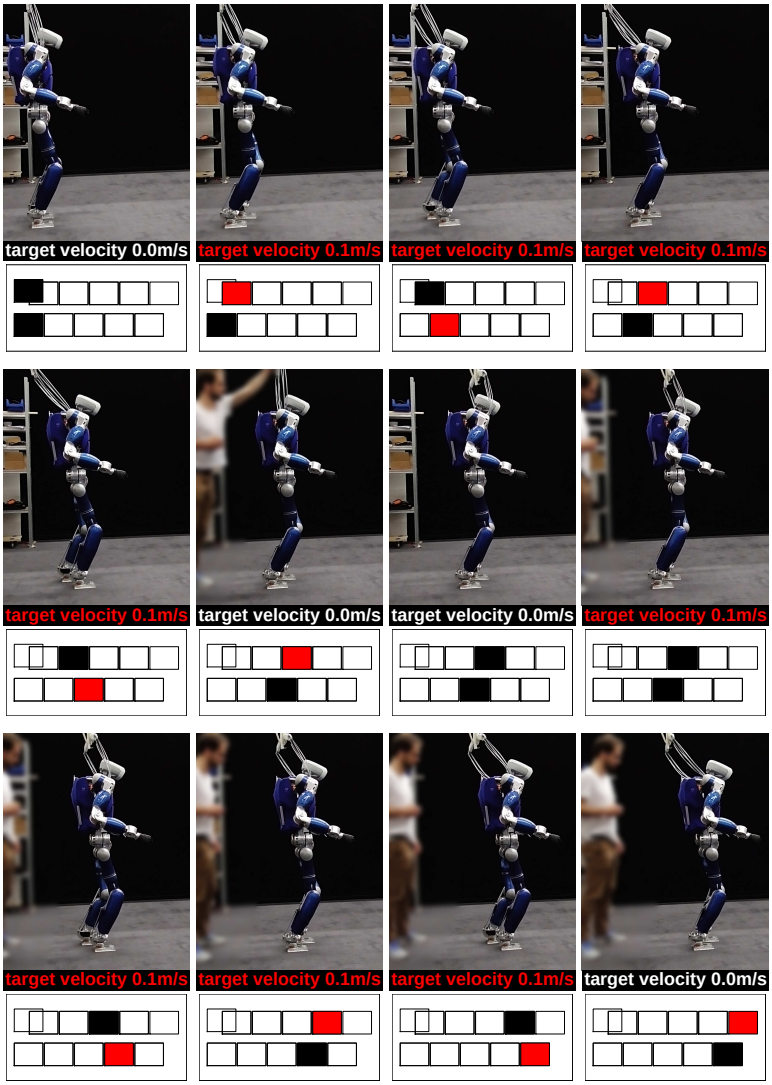

Figure 6.8.: Snapshots of the walking activity with varying target CoM velocity. *From left to right, top to bottom* — footsteps are triggered and continuously generated from the switch from the $0.0m/s$ to $0.1m/s$ target; the robot then automatically stops with the $0.0m/s$ target, and walks again when reverted to $0.1m/s$.

tion of the target velocity is effectively tracked, with noticeable delays due to the time of bringing the robot into motion and stopping it. A significant tracking error nevertheless persists.

The SS and DS phases durations in table 6.1 show that although SS phases are generally as short as the constraints allow[7], DS phases last noticeably longer than what would be expected from the objective.
Indeed, reaching a $0.1m/s$ velocity with a maximum step length of $0.1m$ and a minimum SS duration of $0.7s$ requires a DS duration of $0.3s$; however results

[7] the lower bound constraint is set to $0.7s$ in this setup

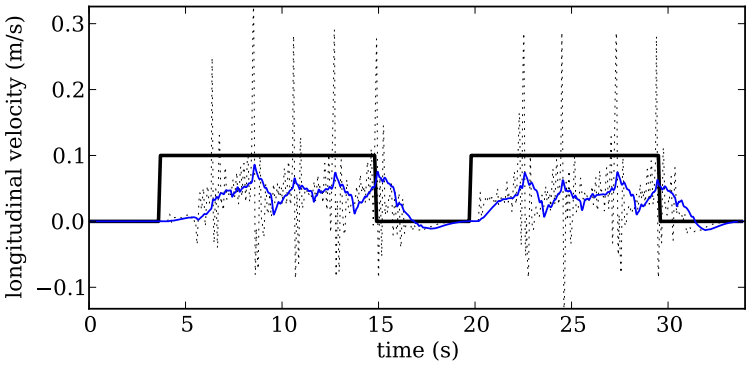

Figure 6.9.: Evolution of the CoM velocity with respect to its varying high-level target. The motion of the robot is consistent with the variations of target velocity, despite a significant tracking error. *Thin solid blue: 1s-window average of the CoM velocity — thick solid black: target CoM velocity — dotted black: instantaneous CoM velocity.*

target vel. $(m.s^{-1})$	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
DS duration (s)	-	2.2	-	2.0	-	2.0	-	2.1	-	(7.5)	-	1.9	-	2.7	-	2.0	-
SS duration (s)	0.9	-	0.7	-	0.7	-	0.7	-	0.7	-	0.7	-	0.7	-	0.7	-	0.7

Table 6.1.: Durations of the DS and SS phases of the walking motion. Although SS phases are hitting the minimal allowed duration to reach the velocity objective, DS phases last noticeably longer than required by the velocity target.

in table 6.1 indicate DS phases longer than $1.9s$. This outcome can be interpreted as a difficulty to trigger a new step, and thus results from the influence of the balance objective over the walking target velocity. Consisting in a reduction of the admissible domain of the CoP positions, DS→SS transitions are indeed more demanding with respect to the CoP constraints and objective than SS→DS shifts.

The seemingly unnecessarily long DS durations can therefore be considered as resulting from an incorrect ZMP tracking: the actual behavior resulting from the low-level is inconsistent with the expected, previewed behavior from the MIQP controller. Indeed, details of the CoP evolution depict oscillations during DS phases, as shown in figure 6.10.

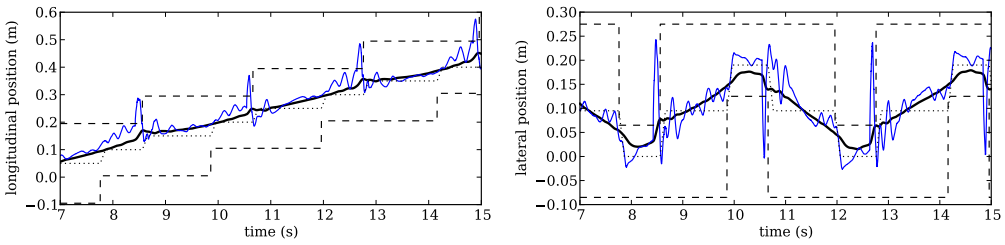

Figure 6.10.: Detailed ZMP tracking performance. *From left to right* — longitudinal direction, lateral direction. *Thin solid blue: measured CoP position — thick solid black: CoM position — dotted blue: ZMP reference — dashed black: actual BoS boundaries.*

These oscillations present a frequency similar to the $10Hz$ control rate of the high-level controller, suggesting that they result from high-level to low-level inconsistencies. The system cannot recover promptly from the disturbances occurring during SS→DS shifts at the landing of the swinging foot, nor rapidly reach an admissible state allowing to trigger the next step.
Inconsistencies of the CoP and CoM behaviors between the higher and lower levels, coupled to the implementation delay of the MIQP controller can indeed induce such outcomes: when tracking a previously computed horizon of CoP and CoM trajectories, the system might reach states which would lead the high-level controller to choose a significantly distinct — or even simply delayed — strategy. These effects are discussed[8] in more details in App. C.

The overall ZMP tracking performance is nevertheless sufficient to allow the robot to successfully walk and maintain balance, as illustrated in figure 6.11.

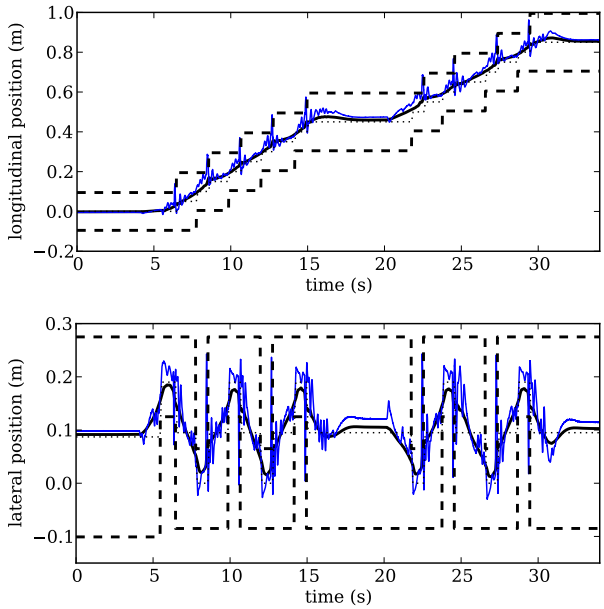

Figure 6.11.: Overall ZMP tracking performance. *From top to bottom* — longitudinal direction, lateral direction. *Thin solid blue: measured CoP position — thick solid black: CoM position — dotted blue: ZMP reference — dashed black: actual BoS boundaries.*

A static error can however be observed on this figure during long DS phases: the velocity of the CoM being null during these periods, the CoP should concur with the CoM. This error suggests a measurement error of the CoP and/or CoM.

Potential causes are biases in the FTS sensors at the ankles, and incorrect CoM position computation. The latter indeed relies on a dynamical model of the system which requires a thorough identification process.[9]

[8] *cf.* p. 183

[9] Note that CoM computation assumes in this implementation an horizontal orientation of the feet, which is debatable on soft grounds as it is the case in these experiments (thin carpeting).

Issues related to the implementation delay of the MIQP controller can moreover be attenuated by increasing the high-level control rate. Indeed, as depicted on figure 6.12, the average computation time to solve the MIQP problem (5.20) ($23.1ms$) is significantly lower than the control period ($100ms$), and more than 98% of the computations are below $55ms$.
This gap suggests that a refined tuning of the problem and solver parameters could therefore allow to safely reach a control period under $60ms$, hence providing more than a 65% increase in control rate.

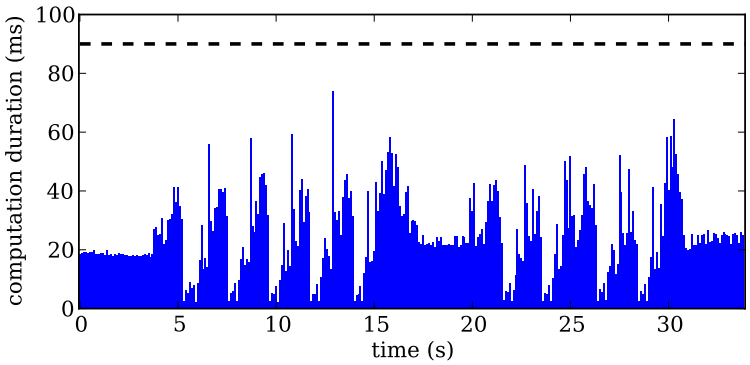

Figure 6.12.: Overview of the MIQP solving computation time. A maximum duration of $90ms$ is imposed to guarantee the $10Hz$ high-level control rate. The average computation duration of $23.1ms$ is significantly under the allocated time span and suggests opportunities to increase the high-level control rate. *Blue bars: MIQP computation duration — dashed black: computation duration limit.*

Disturbances on the CoP during SS phases are also noticeable on figures 6.10 and 6.11. Further causes than the high-level to low-level inconsistencies and delay can be envisioned.

Indeed, as illustrated on figure 6.13, the MIQP controller continuously adjusts the gait parameters, such as target step duration and position. These adjustments can cause large variations on the desired swinging foot velocity and position, therefore leading to significant changes in the foot acceleration. Variations of the swinging foot acceleration imply variations of the rate of change of the system angular momentum, which has a direct influence on the CoP position[10].
The continuous adaptation of the gait parameters therefore induces disturbances on the CoP.

These disturbances can be accounted for at the whole-body level for example, with the use of additional degrees of freedom to maintain a minimal rate of change of angular momentum, or directly as a feed-forward disturbance estimation to the low-level ZMP controller. Effects of the changes of gait parameters are nevertheless smoothened with the low-level cosine-interpolation previously introduced.

Figure 6.13 depicts for example a smooth variation of the step target duration. It can be noted that this step target duration is constant in the last instants of

[10] *cf.* Sec. 4.1.1, p. 85

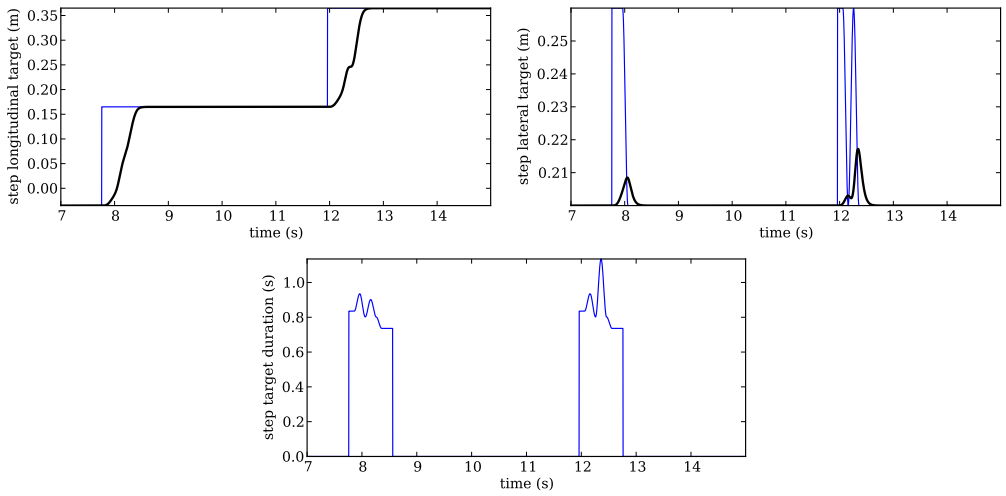

Figure 6.13.: Detail of the gait parameters evolution for the left foot. *From left to right, top to bottom* — step target and current positions in the longitudinal direction and in the lateral direction, step target duration. *Thin blue: target foot position — thick black: current swinging foot position.*

the step. This results from a constraint in the MIQP problem (5.20) forbidding changes in step targets as soon as a SS→DS transition closer than $\approx 250ms$ in the future has been previewed.[11]

An additional remark on the results shown in figure 6.13 is that the lateral step target position is more subject to adjustments than the forward one. This greater variation is induced by the lower weight of the walking objective in the lateral direction in the weighting matrix $\mathbf{S}$ from the walking objective definition[12] (5.17).

Although the ability of the MIQP controller to compute coordination strategies to mainly adapt the BoS constraints to the walking objective has been demonstrated in this scenario, influence of disturbances induced by the two-levelled control architecture on the velocity tracking is observed.
A second scenario is therefore considered to explicitly assess the influence of disturbances on the generated gait pattern.

6.1.5. Conforming objectives to constraints: walking in interaction

In order to validate the behavior of the MIQP controller against unexpected disturbances, a scenario involving a physical interaction with the robot while walking is considered. The robot is requested to track a target velocity of $0.1m/s$. During the walking motion, an external force — estimated between $10N$ and $20N$ — is applied to the back of the robot to restrain and prevent it from moving forward, and is later released.

[11] A similar constraint could also be considered for DS→SS transitions. Such a locking mechanism would thus help reducing the excessive DS phase durations observed in this implementation.

[12] *cf.* Sec. 5.2.2, p. 115

While the use of predefined gait parameters would lead to the fall[13] of the robot in such scenarii, the robot successfully stops stepping until the external restraint is released, and resumes its walking motion from then on. This behavior is captured on the snapshots of figure 6.14.

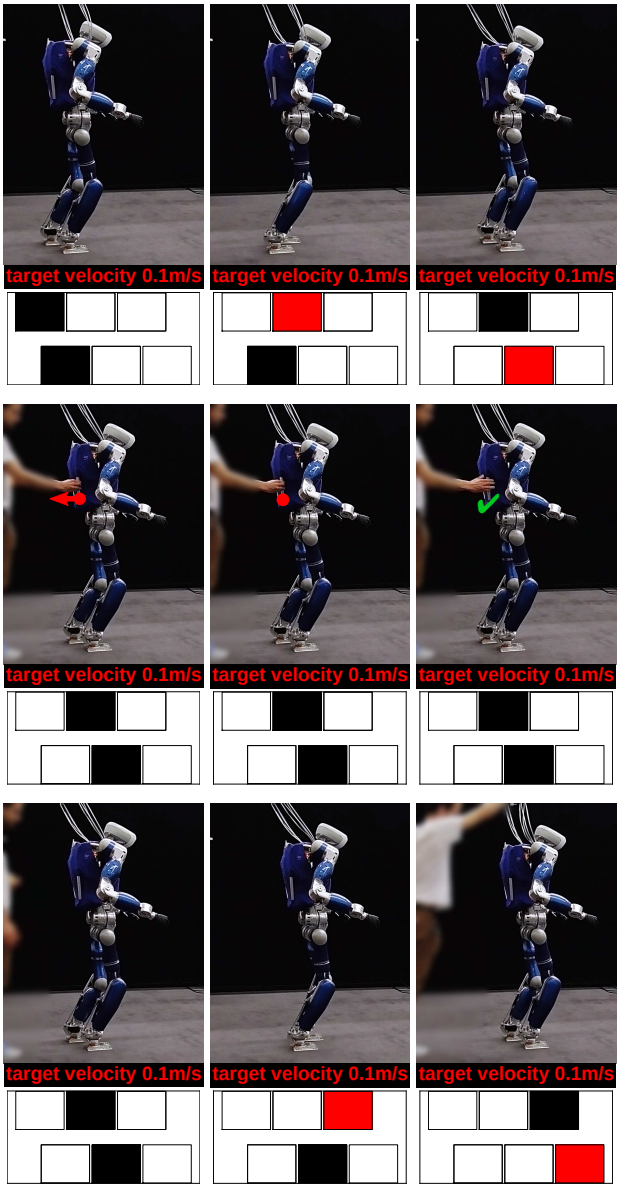

Figure 6.14.: Snapshots of the walking activity in interaction. *From left to right, top to bottom* — footsteps are triggered and continuously generated from the $0.0m/s$ to $0.1m/s$ target velocity; the robot then automatically stops when held in position by an external action, and walks again when released.

[13] or, in the case of online-adapted step positions, to unnecessary in-place stepping

As illustrated in figure 6.15, the robot cannot track the target velocity of $0.1m/s$ due to the external action. Indeed, the system acts in a compliant manner with respect to the disturbance: no stepping motion is outputted from the MIQP controller under these conditions, thus preventing the system to reach the velocity target.

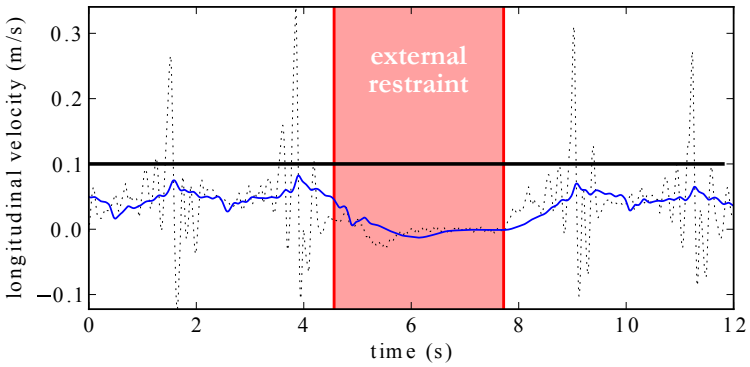

Figure 6.15.: Evolution of the CoM velocity with respect to the velocity objective and external restraint. *Thin solid blue:* $1s$*-window average of the CoM velocity — thick solid black: target CoM velocity — dotted black: instantaneous CoM velocity.*

Figure 6.16 indicates that the CoP cannot reach an admissible position which would allow stepping while the external action is applied to the system. In these particular conditions, the coordination of the evolution of BoS constraints with the balance and walking objectives in the MIQP controller demonstrates its suitability to the walking problem: despite the velocity target, the controller outputs no stepping motion due to the concurrent consideration of balance objective and constraints.

The walking objective is therefore conformed to the disturbances and balance objective and constraints.

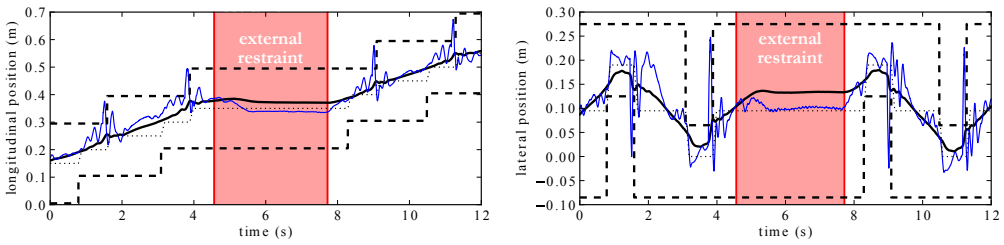

Figure 6.16.: Detail of the evolution of the CoP against the external restraint. *From left to right* — longitudinal direction, lateral direction. *Thin solid blue: measured CoP position — thick solid black: CoM position — dotted blue: ZMP reference — dashed black: actual BoS boundaries.*

6.1.6. Discarding objectives against disturbances

A last scenario exhibits a converse behavior of the robot. The same controller is requested to track a null target velocity, while longitudinal pulls and pushes[14] are applied to the robot; as depicted on figure 6.17, the system adapts the contact configuration to the disturbances, by taking backward and forward steps despite the standstill objective $\dot{\boldsymbol{h}}^r = \mathbf{0}$.

In this case, not only the velocity target is not reached, but the controller takes actions — by triggering footsteps — *against* the walking objective as a response to the external disturbances, in favor of the balance objective and constraints.

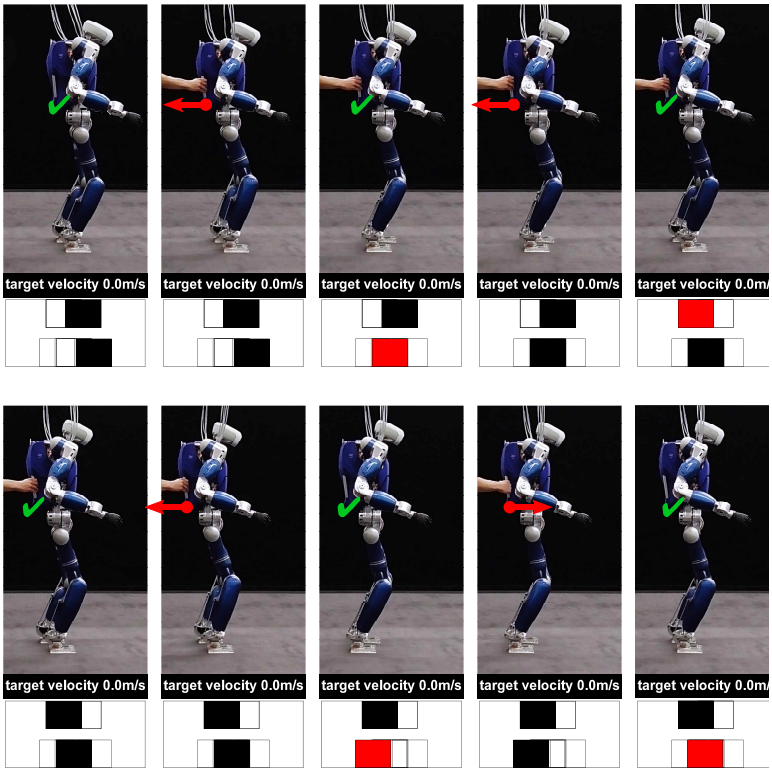

Figure 6.17.: Snapshots of the standstill scenario under external disturbances. *From left to right, top to bottom* — Despite the $0.0m/s$ target velocity, the robot maintains balance by stepping in the direction of the external action.

The resulting CoM velocity tracking performance is illustrated in figure 6.18. As expected, the disturbances and resulting recovery stepping motion significantly prejudice the walking objective. The evolution of the CoP in figure 6.19 shows that the external disturbances drive the CoP towards the virtual BoS boundaries[15] considered in the MIQP problem, leading the high-level controller to trigger a recovery step despite the expected — and effectively *previewed* — variations of the CoM velocity it induces.

[14] these external actions are estimated from the FTS sensors as culminating between $20N$ and $30N$, applied during $\approx 1.5s$

[15] resulting from the introduction of a $\approx 35\%$ safety margin on the CoP constraints (5.9)

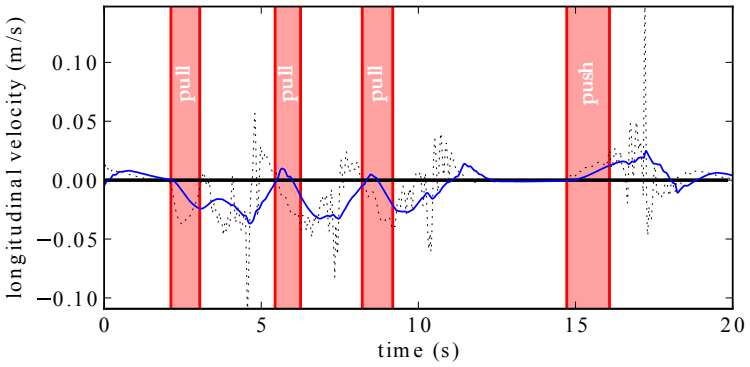

Figure 6.18.: Evolution of the CoM velocity with respect to the null velocity objective and external disturbances. *Thin solid blue: 1s-window average of the CoM velocity — thick solid black: target CoM velocity — dotted black: instantaneous CoM velocity.*

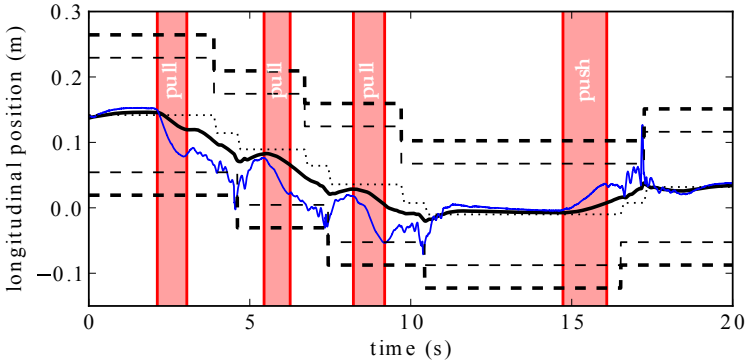

Figure 6.19.: Evolution of the CoP in the longitudinal direction under external longitudinal actions. *Thin solid blue: measured CoP position — thick solid black: CoM position — dotted blue: ZMP reference — thick dashed black: actual BoS boundaries — thin dashed black: virtual BoS boundaries.*

The walking objective is thus purposely and temporarily relaxed to engage a step, therefore allowing to maintain balance against the external disturbances.

6.1.7. Discussion

The three previous scenario for an identical parameterization of the high-level controller demonstrate the generic characteristic of the MIQP MPC formulation (5.20). Indeed, it allows to automatically generate gait parameters allowing to track a desired CoM velocity, while continuously adapting the behavior of the system with respect to disturbances and constraints. The walking motion being defined as an objective rather than a requirement, it can therefore be temporarily sidelined to avoid unsafe motions.

However, the computational cost it implies leads to the setup of a multi-rate architecture, which raises several issues as observed throughout these prelimi-

nary experimental results. The control delay with the consideration of discrete variables indeed bring stability and interpolation issues, and inconsistencies between the high-level and low-level MPC formulations and parameters additionally challenge the multi-rate implementation.

Effective computational times nevertheless suggest that the rate gap between the two control levels can be further reduced to alleviate these issues. The conjoint behavior of the high-level and low-level MPC formulations can moreover be subject to a more detailed analysis to improve performance in tracking solutions previewed at the higher level.
Furthermore, as these two MPC controllers are based on a ZMP model which involves accelerations of the CoM, the use of a position-based whole-body controller brings additional control inconsistencies: an implementation employing a torque-based whole-body formulation may improve low-level CoP control.

Additionally, the use of peripheral degrees of freedom such as the upper limbs could allow to control or reduce disturbances induced by the motion of the legs and feet, which is subject to rapid changes due to the slow-rate discrete changes in feet targets.

Distribution methods can last be envisioned to distribute the computational load of the MIQP problem and achieve greater control rates of the high-level controller. It can be noted that the currently employed solving algorithm is a branch and bound process confined to a single computing thread. This algorithm being fundamentally parallel[16], multiple computing cores can be exploited to support the resolution of the MIQP problem and henceforth decrease computation times, thus allowing a faster rate of the high level predictive controller.

6.2. Control of virtual humans and applications

Virtual humans are digital representations of humans, primarily designed to evolve in simulation environments. Their main characteristic is to reproduce the human behavior, with varying levels of accuracy depending on the application. This behavior to be reproduced can be cognitive or physical, and the latter solely is considered in this section.

The most exploited contribution of virtual humans is the ability to extract information from their simulation in a wide variety of activities, environments and conditions, and at a reduced cost. In design processes for example, evaluations and trials are indeed faster and cheaper to setup in simulation than in real-life conditions.

Virtual humans, being kinematic and dynamic models of the human physical characteristics, furthermore allow to extract non-perceivable information from simulation. An example can be found in the works of Cotton *et al.*: measurements of the limbs cartesian positions are coupled to force recordings at the contacts with the ground to compute the CoM position of the human, with respect to a physical model specifying the kinematics and mass distribution of the human [Cotton2011].

[16] *cf.* App. A, p. 171

The degree of complexity and accuracy of the models employed in the representation of the human characteristics is greatly dependent on the application. For problems such as CoM or external forces estimation, or kinematic analysis, can benefit from the reduced complexity of rigid body structures with revolute joints, therefore closely similar to humanoid robotic systems. However, a deeper analysis of the human dynamics, with internal forces for example, would require the use of refined models enabling the representation of internal tensions and deformations such as muscular models [Nakamura2003] or complex contact models between deformable bodies.

Following the modeling challenge, the generation of the motor activity of human models is to be handled. While a major requirement of this motion is feasibility to effectively capture the phenomena of interest, *realism* is also required by many applications. Despite the growing knowledge of principles regulating humans motor activity and promising state-of-the-art developments in computer graphics [Mordatch2012], motion realism is still an obstacle. This difficulty can be alleviated by the use of input reference motions captured on humans. The use of captured motion along with automatically generated motor activity to extrapolate realist behaviors is nevertheless an open problem.

In this section, a human model as a tree-structure of rigid bodies linked by revolute joints is considered, therefore enabling the use of classical robotics control approaches to regulate its motor activity. Control of this model is supported by the control architecture described in [Salini2011]: a whole-body LQP controller coordinates at the instantaneous level a set of reactive primitives, which are organized and sequenced by a decision layer based on the SPIROPS software relying on employing fuzzy logic [BuendiaSpirops].
The multi-objective predictive layer introduced in Chapter 5 is apposed to the instantaneous layer to support balance and walking tasks while generating appropriate gait patterns.

The autonomous generation of gaits and robustness enabled by the MIQP controller are exploited in a human activity recognition application. Partially captured human motion is used as an input to the control problem, and motion of the lower limbs is automatically generated as a result from the outputs of the MIQP controller. Features extracted from the resulting motion are classified into elementary actions, which are in-turn interpreted in their sequence as an activity being performed, based on a training reference dataset.

6.2.1. Control architecture

The control architecture considered to support these applications is three-layered, as depicted in figure 6.20:

- *Reactive layer*: various reactive primitives are coordinated at the instantaneous level by a multi-objective LQP controller [Salini2010].
- *Predictive layer*: predictive coordination is confined to the balance and walking coordination problem. The multi-objective MIQP controller introduced in Chapter 5 is computing optimal gait parameters and CoM trajectory.

- *Decision layer*: the SPIROPS AI software is employed to parameterize, organize and sequence the various predictive and reactive primitives [Salini2011].

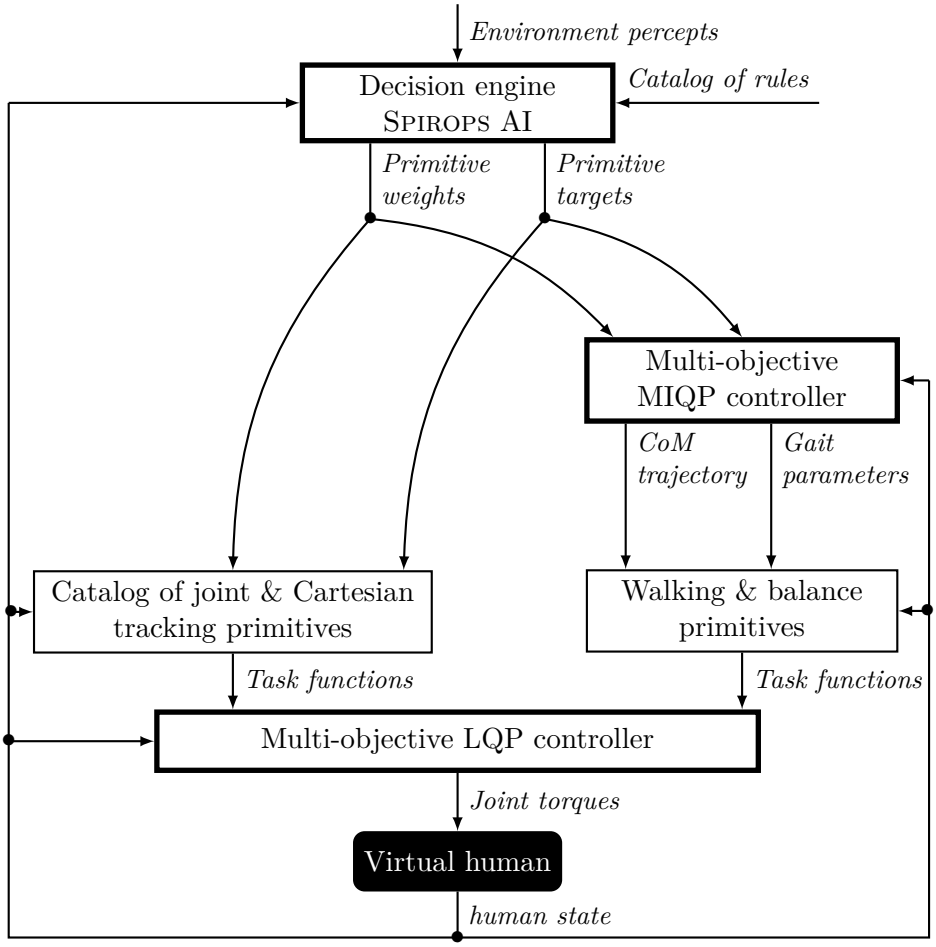

Figure 6.20.: Three-layered control architecture considered for the control of virtual humans. A decision engine organizes and sequences reactive primitives from a catalog, and walking and balance are supported by a multi-objective predictive layer. The resulting prioritized set of reactive primitives is coordinated at the instantaneous level by an LQP-based controller.

6.2.1.1. Elementary tasks

The form of the multi-objective LQP controller introduced in [Salini2010] and described[17] in Sec. 2.2.2.2 allows to consider reactive primitives — task functions — formulated as square errors to be minimized, between reference and effective values of twist derivatives or wrenches in the operational or joint space. This therefore enables the definition of the following classes of tracking primitives T_i:

[17] *cf.* p. 28

$$
\begin{aligned}
&\text{operational space acceleration: } T_i \triangleq \left\| \mathbf{S_{t_i}} \left(\dot{\boldsymbol{t}}_i - \dot{\boldsymbol{t}}_i^* \right) \right\|^2, \\
&\text{joint space acceleration} \quad : T_i \triangleq \left\| \mathbf{S_{q_i}} \left(\ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}^* \right) \right\|^2, \\
&\text{operational space wrench} \quad : T_i \triangleq \left\| \mathbf{S_{w_i}} \left(\boldsymbol{w}_i - \boldsymbol{w}_i^* \right) \right\|^2, \\
&\text{joint torque} \quad : T_i \triangleq \left\| \mathbf{S}_{\tau_\mathbf{i}} \left(\boldsymbol{\tau} - \boldsymbol{\tau}^* \right) \right\|^2,
\end{aligned}
$$

where $\dot{\boldsymbol{t}}$ is a Cartesian twist derivative of a point of interest, $\ddot{\boldsymbol{q}}$ is the generalized acceleration vector of the system, $\boldsymbol{w}$ a Cartesian wrench and $\boldsymbol{\tau}$ the actuation vector. Values $\bullet^*$ are the instantaneously desired values of the corresponding variable, and matrices $S_{\bullet_i}$ are diagonal selection matrices[18] for task T_i.

Desired values $\bullet^*$ are either directly deduced from the primitive objective, regulated by reactive feedback controllers or outputs of predictive primitives.

The resulting set of reactive primitives is coordinated at the instantaneous level by an LQP-based controller to output whole-body actuation. This controller minimizes a cost function defined as a weighted sum of task functions T_i, with respect to linear constraints. The constraints accounted for in this implementation are mainly joint limits (positions, velocities and torques) and non-sliding contact conditions.

6.2.1.2. Automatic walking

The reactive layer is preceded by a multi-objective predictive layer. This layer solely consists in the MIQP controller introduced in Chapter 5 in these applications to mainly benefit from the contribution of predictive coordination at the balance and walking level.

The automatic generation of stable gait patterns first enables to infer motion of the lower limbs in the case of incomplete or erroneous captured motion data. The capture of contact information and lower-limbs motion is indeed challenging in unstructured environments: visual obstructions are frequent for this part of the body (furniture, clothes, legs proximity) and contact information requires invasive or complex instrumentation such as force plates or embedded FTS sensors in the shoes.

This automatic generation is also an asset as it solely requires the definition of a high-level target, such as a goal position or velocity of the CoM. It therefore not only allows to simplify the definition of targets by the decision layer, but also reduces the parameterization required to generate a vast range of motions.

6.2.1.3. Task sequencing and transitions

When performing complex activities, the set of elementary operations to be concurrently executed is changing over time to reach new intermediate goals. The weights of primitives and their targets are therefore bound to evolve over time.

The regulation of this evolution is handled by a decision layer supported by the SPIROPS AI software [BuendiaSpirops]. This decision engine employs fuzzy logic to compute the *interest* of performing an elementary operation, according to percepts of the environment and the virtual human, and to a set of rules specified by the user. The use of fuzzy logic induces that these interests take real values

[18] these matrices can also be employed for changes of basis and homogenization purposes

between 0 and 1, and are therefore particularly suitable to modulate the weights of the primitives attached to the elementary tasks, between a null weight — the task is not performed — and its maximal predefined value [Salini2011].

The decision layer is thus able to sequence elementary tasks and handles smooth transitions between primitive weights, based on a description of the activity to perform through a set of predefined rules. This framework is similarly employed to adjust targets during the execution of the activity.

6.2.2. Robust dynamic motion replay

The principle of motion replay is to map measured motions to a virtual model. The main interest of this technique is to map accessible measurements to non-perceptible — or challenging — variables. For example, Cartesian poses of the limbs can be measured using marker-based motion capture systems, and joint angles can be inferred as extracted from the replay of these poses. Direct measurement of the joint angles would require the setup of invasive and technically-challenging goniometers, thus favoring the setup of motion replay techniques.

Two classes of motion replay can be distinguished. A kinematic motion replay provides access to kinematics-related variable solely. It generally consists in exploiting constraints of the kinematic structure of the system to compute a set of indicators distinct in nature from the measured data. The set of measurements is therefore mapped to a complete, *kinematically-consistent* description of the motion. Assuming a valid kinematics model of the system, inverse kinematics are commonly employed to this purpose.
However, the existence of kinematic singularities or a set of measurements of insufficient dimension can challenge the elaboration of this mapping, being non-invertible in such cases. These problems are generally ill-posed due to the redundancy of humanoid systems: to an insufficient set of Cartesian poses, an infinity of configurations can be mapped. However, patterns can be identified in human motions [Lacquaniti1982], and thus contribute to the elaboration of human-inspired models to solve this redundancy [Vetter2002].

Dynamic replay is a more challenging approach to motion replay. In addition to kinematic constraints, a dynamics model of the system is required which implies the identification of dynamic parameters. Such an identification is more challenging than of a kinematics model: mass-related parameters for example are indeed undeniably less accessible than limb lengths or joint limits. While inverse dynamics methods provides tools to address the motion replay problem, they face the same obstacles as inverse kinematics approaches: inverse dynamics problems are commonly ill-posed. Dynamic replay nonetheless present a growing interest in various fields, providing access to non-perceptible values from joint torques or internal wrenches to muscle activation and tension [Nakamura2003].

As it appears that redundancy is one of the major challenges to motion replay, the use of control techniques for redundant robots seems promising. However, being mainly designed for systems of rigid bodies limits their range of application. The control approach to motion replay consists in considering the measured motion as a target to be reached by a controller, as implemented in [Yamane2003] for example. The replayed motion is therefore the resulting motion of the vir-

tual human, with a feasibility bounded by the accuracy of the model and of the validity of the simulation environment in reproducing real conditions.

The approach adopted in this section separates the motion replay problem into two of the mappings to be performed: kinematic and dynamic mappings. The captured data is first mapped to a kinematically-consistent motion enforcing kinematic constraints of the model on the measurements, from which are extracted various trajectories to be tracked by a dynamics controller in order to infer the corresponding dynamically-consistent motion of the virtual human.

6.2.2.1. Kinematic replay

The kinematic replay relies on the accuracy of the kinematic model of the virtual human. This model is first scaled with respect to the height and mass of the subject from whom the motion is captured, using anthropomorphic tables. A more precise identification of the dimensions of the model can be performed on a per-segment basis, and similarly with joint limits.

The motion capture system is assumed to provide Cartesian positions of a limited set of markers positioned on the limbs of the subject. Spring-damper systems are attached to the virtual human for each marker, between the measured marker absolute position and their attachment point on the limbs of the virtual human, as illustrated in figure 6.21.

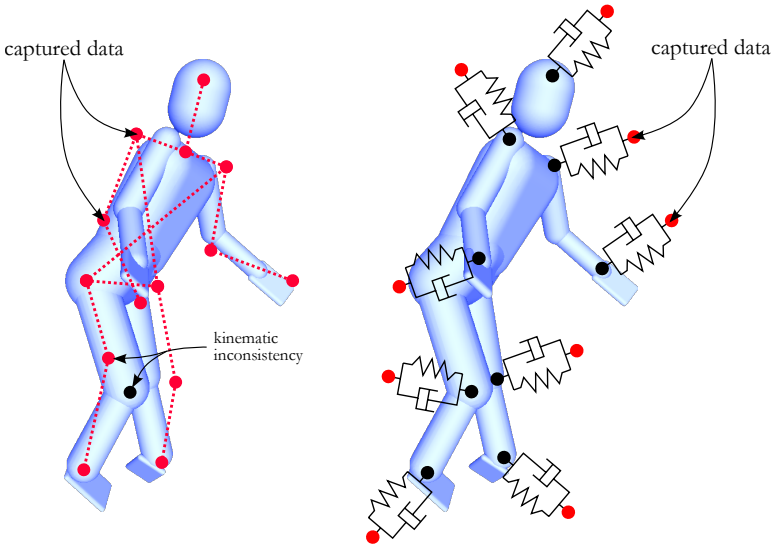

Figure 6.21.: Illustration of the kinematic replay approach. Spring-damper systems produce forces on the virtual human to define an equilibrium configuration with respect to the recorded marker positions. *From left to right* — the recorded motion presents kinematic inconsistencies with the model, and conflicts are solved as an equilibrium configuration between the inconsistent targets.

These systems produce forces on the virtual human, and its equilibrium configuration is a kinematically-consistent representation of the recorded motion. Kinematic inconsistencies are represented with this approach as conflicts between attractive forces, solved as an equilibrium configuration. Redundancy can fur-

thermore be solved by setting up similar spring-damper systems at the joint level or by the introduction of a gravity effect, driving the virtual human towards a predefined reference configuration, and being a particular solution in the null-space of the therefore ill-posed problem.
To further tune the resolution of inconsistencies in the measurements, the parameters of these spring-damper systems can be tuned individually to introduce priorities among the markers. As setup by Demircan *et al.* in a dynamic replay framework [Demircan2008], the highest priority can be given to the markers of highest interest, for example to the ones attached to the distal limbs, head and waist of the subject.

The motion of the virtual human has obviously no physical validity: real interaction forces with the environment are not represented. However, the resulting motion respects kinematic constraints of the model; it allows to extract a greater number of virtual measurements such as joint configurations and velocities, and segment positions and orientations which are used as kinematically-consistent targets to a dynamics whole-body controller.

6.2.2.2. Dynamically-consistent motion

Since marker-based motion capture requires additional instrumentation to accurately capture contact information with the ground, the captured motion of the upper limbs solely is assumed to be available. This assumption is verified for cluttered environments in real-life conditions: markers of the lower limbs generally suffer from visual obstruction, thus greatly challenging the identification of the contact conditions of the subject with the ground.

Motion of the lower limbs is henceforth generated automatically, with respect to balance and walking criteria. The balance objective ensures physical feasibility of the produced motion, while the walking criterion aims at generating a displacement consistent with the observed motion of the subject. The multi-objective MIQP predictive controller introduced in Chapter 5 is a suitable candidate to find motor strategies which coordinates these conflicting objectives. In off-line conditions, the preceding kinematic replay allows to extract an horizon of target CoM displacements and velocities. This horizon can otherwise be inferred from instantaneous measurements, assuming for example a constant CoM velocity over the preview horizon.

The MIQP biped model is extended to account for predefined rotations of the feet, assumed constant over the preview horizon. These rotations are considered in the model with a rotation of the reference frame in which the base of support is described. Rotation of the feet is inferred from the orientation of the waist in this implementation. However, actual feet rotation measurements can be similarly employed if available.
It can be noted that if foot positions are available from the capture, an additional objective can be naturally appended to the problem formulation[19] (5.20) to account for these desired contact configurations. This objective would write as a

[19] *cf.* Sec. 5.2, p. 114

quadratic error between the parameters of the previewed and measured bases of support.

This approach to the motion of the lower-limbs nevertheless gives a higher priority to feasibility than to replay accuracy of the motion of the lower limbs. For applications not specifically relying on the analysis of the human balance principles, motion of the upper limbs is generally of greater interest indeed, thus motivating this choice of implementation. Furthermore, the generation of stable walking motions is a requirement to valid motion replay: tracking captured motions necessarily implies the ability of the virtual human to successfully balance.

Dynamically-consistent motion of the upper limbs is obtained with the tracking of the kinematically-consistent data extracted from the preceding kinematic replay. A Cartesian acceleration tracking primitive is associated to each segment with a proportional-derivative controller, taking as reference the segment pose and velocity resulting from the kinematic replay. As performed in [Demircan2008] in a strict hierarchy framework, greater weights for these primitives are given to the hands, head and waist of the virtual human in the LQP controller. In case of conflicts with the balance task, these priorities tend to prevent tracking errors in the points of interest of the virtual human, releasing instead degrees of freedom related to the most interior limbs.

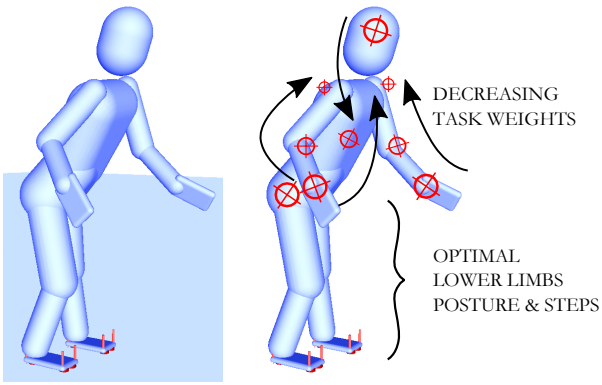

Figure 6.22.: Illustration of the dynamic replay approach. The dynamically-consistent motion of the virtual human are obtained with the tracking of the kinematically consistent motion of the upper limbs, while the lower limbs support balance and displacement objectives and constraints.

This approach to the whole-body problem of dynamic motion replay is depicted in figure 6.22. Accuracy of the extreme limbs is favored for the upper part of the body, while motion of the lower limbs is generated with respect to higher level objectives, maintaining feasibility of the posture while tracking the displacements of the subject. Despite this apparent segmentation of the virtual human, the motor activity is coordinated at the whole-body level in both predictive and reactive layers.

The physical accuracy of the motion is however strongly related to the accuracy of the simulation framework outputting the motion of the virtual human from

the control inputs. This implementation is supported by the XDE[20] software, developed by CEA-LIST [Merlhiot2012], designed with a specific attention to the simulation of interacting multi-physics systems.

6.2.2.3. Application to the recognition of human activity

An implementation of this approach to the motion replay problem is employed in the works of Granata *et al.*, to support a multilayer framework [Granata2015] enabling the recognition of elementary tasks and the interpretation of their sequence as an activity, from RGB-Depth sensor measurements and a reference dataset.

With a view to consolidate human-robot interactions, this framework aims at enabling the encoding of human activity from context descriptors and measured human behavioral signals. The proposed approach employs a multi support vector machine as a classification layer trained with features such as indicators of the human motor activity, temporal segments and object affordances to recognize elementary operations composing an activity. The sequence of these operations is henceforth interpreted at a higher level as activities with discrete hidden Markov models.

The motion replay technique presented in this section is exploited in this perspective to extract dynamically-consistent indicators of the considered human motor activity from inaccurate skeleton data, as depicted in figure 6.23.

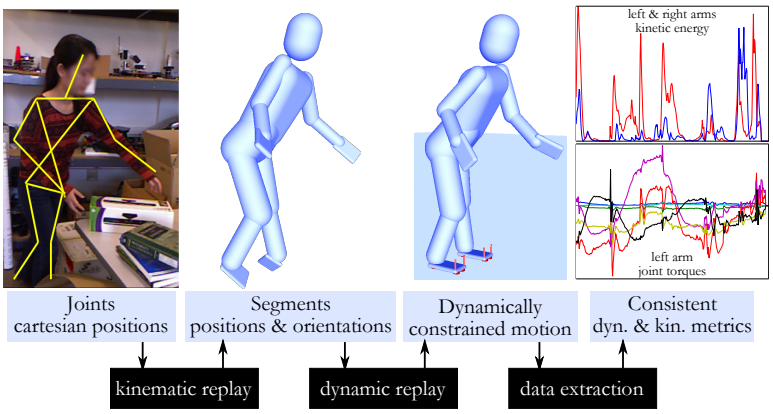

Figure 6.23.: The dynamic motion replay approach is employed to extract dynamically-consistent indicators of the human motor activity from incomplete and inaccurate captured data. A kinematic replay first enforces kinematic consistency and provides kinematic data to be tracked by a controller-based dynamic replay. This dynamic filter enforces dynamic constraints on the motion, from which are extracted dynamic and kinematic metrics.

To consider non-invasive and general public applications, an RGB-Depth sensor (Microsoft® Kinect®[21]) is employed as a marker-less and low-cost motion capture system. Critical inaccuracies are observed in the captured data, especially for

[20] http://www.kalisteo.fr/lsi/en/aucune/a-propos-de-xde
[21] https://msdn.microsoft.com/en-us/library/jj131033.aspx

the lower limbs: erroneous leg identification often occurs due to occlusions or ambiguous poses. Additionally, the accuracy of the capture of the feet positions is not sufficient to infer the contact state of the human. The captured data is therefore considered as incomplete. This observation motivates the motion replay approach introduced in this section which allows to automatically generate consistent motions of the lower limbs.

Although the proximity of the resulting motion of the lower-limbs to the actual motion is arguable, the resulting motor activity of the upper limbs is sufficiently accurate to extract relevant motor features from the perceived human activity. These metrics are extracted in this way from the captured data of the dataset, and are used in combination with temporal segmentation of the activity and object affordances provided by the same dataset to train a classifier of elementary operations. An interpretation layer is henceforth trained on the resulting identified sequences of operations, labeled as activities.

Training and validation are performed on the Cornell Activity Dataset[22] (CAD-120), and exhibit performances comparable to state-of-the-art approaches such as [Koppula2013].

The MIQP controller demonstrates its efficiency in the replay of dynamic activities, implying displacements of the virtual human of great amplitude and velocity, outputting stable motions for 95% of the 120 captured activities of the dataset. The coordination at the predictive level therefore allows to robustly compromise between desired displacements and balance constraints for *everyday life* human activities.[23]

6.3. Conclusion

Applications of the multi-objective predictive control problem to the walking and balance control of humanoid systems, formulated as an MIQP optimization problem, are presented in this chapter.

The computational cost of this formulation is first handled under real-time requirements with the setup of a multi-rate control architecture on the humanoid robot TORO. Despite technical challenges, the multi-objective predictive layer effectively anticipates coordination strategies between conflicting objectives, thus producing complex adaptation behaviors to objectives and disturbances.

Additionally, simulation-based applications exploit the main characteristics of this formulation in the control of virtual humans, enabling tools contributing to the perception and understanding of the human activity surrounding the robot.

First, preliminary experimental results on the TORO humanoid robot assess how the computationally-demanding MIQP MPC problem can be employed in a multi-rate control architecture, and allow the system to automatically and continuously anticipate gait patterns as a response to walking and balance objectives and constraints, and external disturbances.

[22] `http://pr.cs.cornell.edu/humanactivities`

[23] The dataset indeed considers activities such as stacking/unstacking/picking objects, microwaving food, cleaning objects and taking food.

Predictive primitives are employed at two different control rates with distinct problem complexities. Coordination strategies are elaborated at the slower rate, and are tracked by a decentralized predictive level and then coordinated instantaneously using whole-body motion. Although inconsistencies between the various employed models raise several challenges, the resulting control architecture allows the system to exhibit behaviors solving conflicts between objectives, constraints and unexpected disturbances. The humanoid robot efficiently maintains balance while tracking a walking objective, despite external actions such as pushes and pulls.

These preliminary results therefore demonstrate the **feasibility** of the application of computationally-demanding multi-objective predictive control architectures to physical systems.

As opposed to physical applications, virtual humans are generally not subject to real-time requirements. Despite the virtual characteristic of these systems, real-life applications can nonetheless be considered.

The robustness of the generated motion from the MIQP controller is first exploited in a partial motion replay application. Incomplete captured motion data is tracked by the upper limbs of a virtual human, while balance and walking are mainly supported by the multi-objective MPC problem. Although accuracy of the lower limbs motion is sidelined, this approach enables the generation of feasible motions from incomplete data. This motion replay is employed in a human activity recognition perspective: kinematically and dynamically-consistent motions are mapped from a partial set of captured data, allowing to extract metrics describing the motion of the subject. These metrics are considered as features used in combination with additional percepts to classify sub-activities, their sequence being translated into labeled activities.

Such results demonstrate how multi-objective predictive control, and robust control of humanoid robots in general, can **contribute to the perception** aspect of robotics. The setup of virtual humans indeed allows to infer additional information from a limited set of percepts, information which can be henceforth used at the decision or control levels to compute appropriate behaviors of the robot.

While the feasibility of the implementation of computationally-demanding predictive control architectures is demonstrated, the application to virtual humans of the robust controllers thus formulated enables the contribution of control formulations to span beyond the sole regulation of the motor activity of physical systems. Virtual humans indeed provide opportunities to significantly extend the range of perception of robotics system, with the development of **human-centered** tools for the capture, generation and analysis of human motor activity.

CHAPTER 7

Conclusion

Involving robotic systems in complex activities requires the conjoint consideration of two critical aspects of such problems: objectives are to be reached as the activity demands, while respecting a set of intrinsic and extrinsic constraints. Not only can objectives and constraints be of various nature, be defined at distinct levels, and evolve over time, they implicitly define optimal *and* infeasible behaviors of the system with respect to various and fluctuating factors. While robust perception should capture the evolution of these factors, efficient and safe control must elaborate motor strategies to reach such optimal behaviors while avoiding the infeasible ones.

Instantaneous control policies can be defined to effectively guide the system towards an optimum while avoiding unsafe situations. However, changes in the environment, disturbances and evolutions of the objectives bring the need for continuous adjustments of such policies, to adapt to the fluctuations of the situation. Indeed, instantaneous targets outputted from the decision layer to the instantaneous level are generally the result of static planning phases, unaware of evolutions of the control problem. To this purpose, a decision level is generally introduced to translate these changes into policy adjustments. The level of abstraction of such layers is generally high nonetheless, and raises optimality and versatility issues. Indeed, decision layers commonly imply the definition of rules, either manually specified or issued from a learning process, which rely on highly abstracted models to scale down the decision problem. This reduction inevitably introduce biases, from the specified heuristics or adopted abstractions, thus favoring particular solutions over others. Furthermore, generalization of the therefore specified adjustment strategies is debatable: their validity is limited to the range of envisioned situations.

Predictive control is considered in this dissertation as an intermediate level, **bridging the distant levels of abstraction of the instantaneous and decision levels**. Formulating the activity as an optimization problem with its objectives and constraints over a preview horizon, this framework is particularly relevant to the control problem of complex activities. Not only optimality and feasibility are explicitly considered, the evolution of the system, the environment and the activity can also be apprehended and anticipated over a preview horizon. Nevertheless, formulating such optimization problems can be challenging, and their resolution even be intractable for highly complex activities and systems.

Reduced models and activities are therefore considered in this work at the predictive level to cope with such issues: computationally-favorable models of the system are employed to execute a selection of sub-activities. These sub-activities consist in objectives specified at an **intermediate level**, in-between the local objectives tracked at the instantaneous level and the global goals of the activity.

This approach henceforth alleviates the challenge of formulating a complex activity as a set of instantaneous, local objectives to be reached: a bridging level of abstraction is introduced to **elaborate and refine adjustments** in the final control policy. Handling the elaboration of control strategies at the predictive level indeed allows, through the optimal definition of these strategies, to **reduce the influence of the decision layer** and the biases it may introduce, as demonstrated in this dissertation with the automatic generation of walking patterns for example. Additionally, decomposing a complex activity into sub-activities from the decision layer to the instantaneous level commonly induces the definition of conflicting sub-objectives to be reached simultaneously. Refining the strategies specified at the decision level allows to perform adjustments in the control policy **solving** these potential **conflicts**, as exhibited in this work for manipulation and walking activities.

7.1. Contributions

The general contribution of this work is the introduction of a multi-objective, predictive control level in the control architecture for complex activities.
The practical feasibility of such an approach and the gains it provides are evaluated in various applications to central control challenges of humanoid robotics, highlighting a selection of the contributions and concerns yielded by predictive control methods.

A first formulation addresses the task-posture problem for humanoid robots. Manipulation activities can commonly be considered at a lower level as the coordination problem between balance and manipulation objectives. Such a decomposition raises the concern of conflicts between these objectives, especially in disturbed conditions. A reduction of the problem is performed with an approximation of the coupling between these conflicting objectives and an identification of the main dynamics they involve. The resulting coupled, reduced models are employed in a predictive control problem, aiming at conjointly adjusting the manipulation and balance control policies to simultaneously optimize the manipulation and balance sub-objectives, with respect to a known external disturbance applied at the hand. Approximations on the coupling between the reduced models furthermore allows to perform a sequential distribution of the optimization problem to significantly **reduce its complexity**.

Simulation results on the iCub humanoid robot demonstrate that coordination strategies can be elaborated at the predictive level to **solve conflicts** and obtain an **overall performance increase** for both tasks. Additionally, priorities are defined at the predictive level in a weighted manner, which influences the resulting performance as expected while still allowing temporary relaxations of the objectives of higher priority. This insight further highlights one of the major contributions of predictive control to the problem of complex activities: the specification of soft hierarchies **does not inflexibly induce a constant prevalence** of the objectives of higher priority; **relaxations can emerge** if beneficial to the global objective of the activity, enabled by the exploitation of a future time window. Additionally, constraints can be naturally introduced in predictive

formulations, expressed as optimization problems: prevalence can therefore be enforced nonetheless if required.

A second formulation considers a walking activity as the concurrent minimization of tip-over and slippage risks. A cooperative optimization problem is thus formulated at the predictive level, aiming at coordinating the free-floating dynamics of the system, expressed at its center of mass, in order to maximize postural stability expressed as a multi-objective problem. A dual decomposition method is employed to distribute this cooperative problem into a parallel problem: a set of local sub-problems and their couplings are arranged into a parallel architecture.

The non-cooperative architecture obtained from this systematic distribution method presents various advantages. The modularity enabled by the explicit consideration of local sub-objectives and their couplings allows to naturally approximate the topology of the problem. Indeed, couplings between sub-problems can for example be omitted by solely altering the flow of information within the parallel architecture. This provides a significant gain in **versatility**, allowing to naturally consider additional sub-problems and **organize the activity control problem** with the specification of bilateral or unilateral couplings to compromise between accuracy and complexity.
Furthermore, this type of architecture is suitable for parallel resolution algorithms, therefore offering opportunities to significantly **decrease the computational complexity** of the predictive level.

Simulation results depict the conflicting nature of the tip-over and slippage minimization objectives. These conflicts are effectively solved through the elaboration of coordinated control strategies, to provide a significant increase in the postural stability of an iCub humanoid robot. Furthermore, these results provide insight on the influence of the level of abstraction of the models employed, and the duration of the future horizon they are previewed over. Indeed, different expressions of the tip-over risk are introduced in this formulation, with distinct horizons being identified as providing the best performance.

Despite the increase in performance observed with the previous formulation, the exhibited adjustments in control policies have a relatively local influence. The predictive level is drawn **closer to the decision layer** in a third novel approach, considering discrete actions in the optimization problem. The inputs to this problem are significantly reduced, consisting in a sole position or velocity to be tracked, and a prevalence ratio between balance and walking. Walking activities are therefore approached at a **higher level** indeed, as the coordination problem of discrete changes in the base of support with continuous postural adjustments, aiming at maximizing balance and following a specified displacement.

The predictive problem is centralized in this formulation, and no distribution is performed to reduce the resulting computational cost. However, the hybrid reduced model considered in this problem allows the definition of a convex mixed-integer optimization problem which can benefit from fast and efficient solvers.

The walking motion being defined optimally, simulation results demonstrate that the **emerging behaviors** are **adjusted** with respect to the fluctuations of objectives and the presence of unexpected disturbances. Additionally, a **range**

of behaviors can be explored with the modification of the priority ratio between balance and walking, defining the coordination strategies to be adopted.

The main contribution of this approach is the absence of pre-specified behaviors to be considered, as it could be required at the decision level; no distinction is enforced between walking motions, standstill behaviors and recovery stepping. Therefore, the resulting motion of the system is an **optimal** and complex **combination** of these strategies, implicitly defined with respect to the objectives to reach, the constraints to respect and the perceived and anticipated variations of the system state.

The computational cost of this formulation is nevertheless challenging its implementation on physical systems with **real-time requirements**. Two predictive levels are considered for a real-time implementation on the humanoid robot TORO, running at different rates. The strategies of discrete actions emerging from the walking predictive controller are considered at a higher level, and a slower rate. A second predictive layer is introduced to support continuous postural adjustments at a faster rate. Predictive control is thus envisioned in this architecture at two intermediate levels between the decision and instantaneous layers, providing adjustments in the control policy **at diverse levels with distinct models**.

Despite raising multiple challenges, this architecture demonstrates in preliminary experimental results its ability to adjust control strategies to objectives, constraints and disturbances in real-time.

Practical applications are not necessarily subject to real-time requirements nonetheless. **Virtual humans** are indeed interesting simulation tools supporting various applications. The motion replay problem is for example considered in this dissertation. A robust control architecture is employed to enable the tracking of captured human motion data, while respecting kinematic and dynamic constraints. A mapping from captured kinematic data of the upper limbs to whole-body feasible dynamic metrics is performed with the generation of feasible and compatible motions of the lower limbs of a virtual human model. The resulting metrics are for example exploited in a **human activity** recognition application.

7.2. Perspectives

In a short-term perspective, the further application of a predictive level to the control of virtual humans in both autonomous and motion replay applications is envisioned. Virtual humans are indeed subject to a growing interest with the increasing attention drawn to human-centered applications. Their availability in evaluation and design industrial tools is nevertheless generally limited to kinematic or static implementations, therefore leaving room for significant improvements.

Such improvements are expected to contribute to the design of collaborative and assistive robots morphologies and control policies. The control architecture presented in this dissertation is envisaged in the autonomous control of virtual humans to allow the evaluation of ergonomic indicators in human-robot collabo-

rative activities [Maurice2014]. These indicators can be employed in assessment-only applications with the replay of captured motions or the simulation of workstation and task designs for example, and be further applied to the automatic generation of feasible motions for the human-centered optimization of the morphology of collaborative robots [Maurice2015].

Furthermore, virtual humans can be employed to capture and infer data from the human activity surrounding the robot, therefore providing deeper percepts for the decision and control layers to consider.

Improvements are also envisaged in the implementation of the two-layered walking controller on the TORO robot. The high-level problem formulation and parameterization can be further developed to reach faster control rates, and a deeper analysis of the conjoint behavior of the two control levels could provide significant increases in the overall control performance.

Distribution techniques can additionally be employed to further reduce the complexity of the mixed-integer problem to reach a sequential or parallel architecture of smaller mixed-integer sub-problems.

Moreover, the generalization of the mixed-integer approach to multi-contact motions is promising, offering a computationally-favorable framework to support at the control level a wider range of activities. However, the expression of kinematic constraints at a reduced level is challenging, and the consideration of non-convex contact surfaces would require the introduction of additional integer variables. Constraint relaxations can nevertheless be envisioned to reduce the complexity of the optimization problem.

A long-term objective from the developments of this work is to reach a complete and generic predictive framework for humanoid systems in complex activities. This framework would organize a catalog of common predictive primitives in a parallel, distributed manner with the consideration of their couplings. Systematic distributions methods such as dual decomposition can be employed to parallelize the multi-objective problem as demonstrated in this work, and critical predictive primitives such as balance and walking could be specifically designed with methods such as the ones introduced in this dissertation. Systematic model reduction techniques, such as projection-based model-order reductions, can additionally be envisaged to contribute to the generic characteristic of such a framework.

Last, the control problem of complex activities is incontestably not confined to the field of humanoid robotics. Multi-layered control architectures combining predictive and instantaneous multi-objective controllers are expected to contribute to the performance of redundant systems such as complex robotic manipulators or multi-agents systems.

Appendices

List of appendices

APPENDIX A

Branch and bound algorithm for Mixed-Integer Quadratic Programming

Branch and bound (BB) is an algorithm design paradigm widely applied to combinatorial optimization problems and, more specifically, to integer programming [Lawler1966]. The principle of this class of algorithms is to explore the solution set through the consecutive search of solutions within increasingly smaller subsets (*branching*). Although the number of subsets is generally increasing throughout the iterations of the algorithm as a result from branching, a *bounding* method allows to discard (*pruning*) subsets being identified as not containing a potentiel solution.

The initial problem is therefore solved through the resolution of a tree graph of sub-problems defined over an increasingly smaller domain, the initial problem as its root. The graph is constructed and explored with recursive iterations comprising of branching and bounding phases, as follows:

- *Branching*: the set of solutions of each child node i of the tree is decomposed into n_i smaller sub-sets. To guarantee an existing solution to be found, the union of the smaller sub-sets must cover the parent solution set. The resulting n_i sub-problems constitute n_i child nodes of the node i.
- *Bounding*: for each node j of the tree, the upper and lower bounds of the value of the cost function to minimize are computed.
- *Pruning*: If the lower bound of some node j of the entire graph is greater than the upper bound of an other node k, the node j can be safely discarded from the tree. Indeed, the solution sub-set associated to this node necessarily does not contain the minimum.

Recursion stops when a termination criterion is met, with a threshold on the minimal upper bound for example. Performance of such algorithms is vastly dependent on the branching procedure, determining the directions of search.

A.1. Application to Mixed-Integer Quadratic Programs

The application of BB algorithms to MIQPs exploits the quadratic form of the optimization problem. The key principle of such applications is to relax the integer-value constraints of the mixed-integer problem: the MIQP is therefore written as a real-valued QP.

Iterations rely in that way on the accessibility of an optimum for real-valued QPs: fast solvers are indeed vastly available. The resulting process can be described as follows, starting with the bounding phase for convenience:

- *Bounding*: the minimum of the cost function of the convex, real-valued QP associated to each node is computed.
- *Pruning*: for each node where the solution respects integer-value constraints (it is *integer-feasible*), if the previously computed minimum is greater than the one of an other integer-feasible node, it is discarded.
- *Branching*: integer-feasible nodes are kept in memory and do not have children. Branching occurs for each non-integer-feasible node. For each variable not respecting the integer-value constraint it is subject to in the original MIQP problem, the solution set is splitted into two sub-sets in the direction of this variable around its real, infeasible optimal value.

The following splitting process can be applied: considering the integer variable $x \in \mathbb{N}$, and its real-valued counterpart $\tilde{x} \in \mathbb{R}$, an optimum to the relaxed problem is found, for example $\tilde{x}^* = 0.6$. Since $\tilde{x}^* \notin \mathbb{N}$, this solution is not feasible. The relaxed solution set

$$\{\tilde{x} \ / \ \tilde{x} \in \mathbb{R}\},$$

considered in the current node to obtain $\tilde{x}^* = 0.6$, can be split into two sub-sets excluding the non-feasible solution $\tilde{x}^* = 0.6$. These subsets thus write

$$\{\tilde{x} \ / \ \tilde{x} \in \mathbb{R}, \tilde{x} < \tilde{x}^*\} \quad \text{and} \quad \{\tilde{x} \ / \ \tilde{x} \in \mathbb{R}, \tilde{x} > \tilde{x}^*\}.$$

However, since the pursued solution x^* is expected to be an integer, the sub-sets $]0, \tilde{x}^*[$ and $]\tilde{x}^*, 1[$ are not admissible with respect to x for $\tilde{x}^* = 0.6$. This allows to obtain even smaller sub-sets without ignoring feasible solutions, that is

$$\{\tilde{x} \ / \ \tilde{x} \in \mathbb{R}, \tilde{x} \leq 0\} \quad \text{and} \quad \{\tilde{x} \ / \ \tilde{x} \in \mathbb{R}, \tilde{x} \geq 1\}.$$

This process is illustrated in figure A.1. When all real-valued counterparts of integer-valued variables are integers[1], the branching is stopped for this node. Branching can similarly be stopped before the resolution in some cases, if the sub-set is reduced to a singleton for example. In the case depicted in figure A.1, the two child sub-problems lead to feasible integer optima, on the bound of their respective sub-set. The solution corresponding to the minimal value of the cost function, that is the solution of the sub-problem 2, will be kept as solution to the original MIQP problem.

A.2. Computationally-efficient implementation

An illustration of the overall search process is proposed in figure A.2. Although the optimum may be encountered during the exploration of the search graph, it cannot be identified as the solution until branching stops for all sub-problems. The number of child nodes can therefore be high; however, the QPs at a same level being independent, their resolution can be naturally parallelized. Highly-efficient solvers moreover employ heuristics to guide the exploration of the search graph, and the integer characteristics of the original problem can be further exploited with the setup of a cutting-planes method for example to reduce the solution set.

[1] since numerical solvers are employed, an integer tolerance is used to decide if $\tilde{x}^* \in \mathbb{N}$

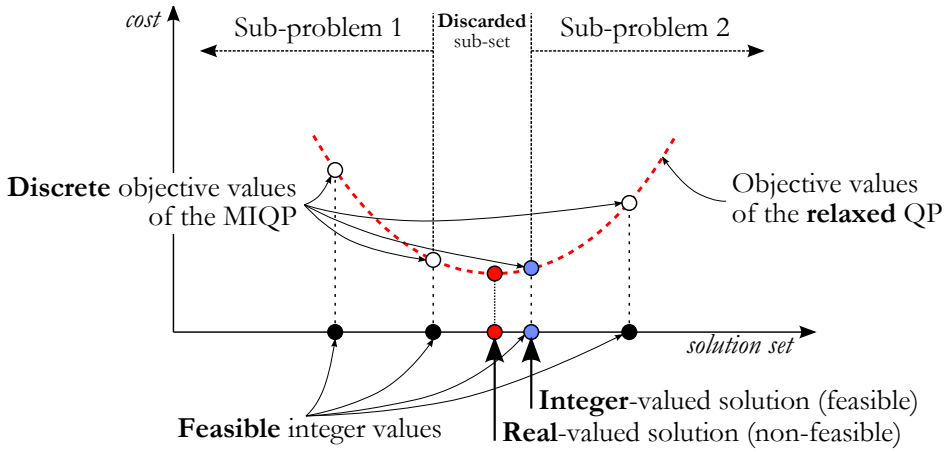

Figure A.1.: Illustration of the branching process for MIQP problems. The relaxed QP problem enables the computation of a real-valued optimum. The solution set is splitted into two sub-sets around this infeasible optimum, leading to the creation of two child nodes in the search graph.

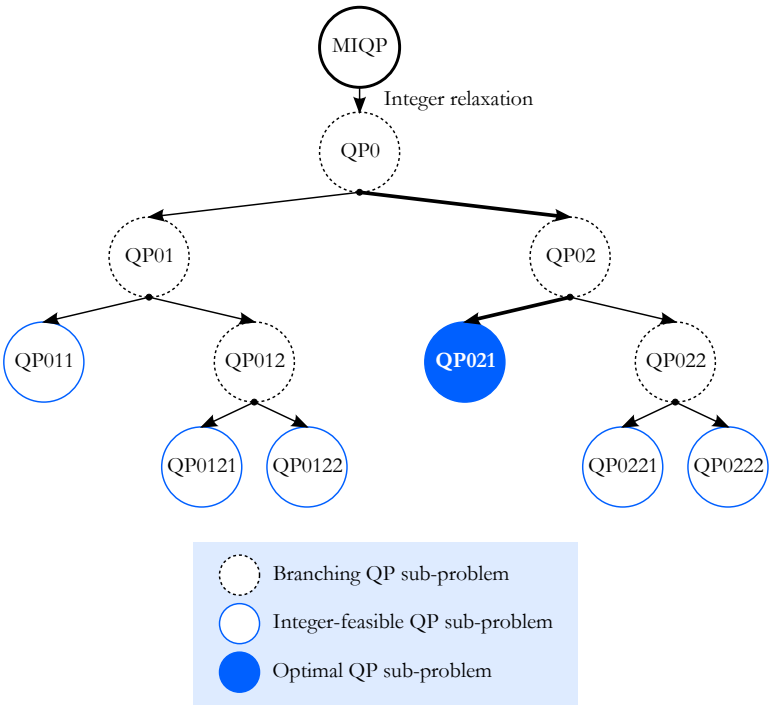

Figure A.2.: Illustration of the branch and bound graph search for MIQP problems. The MIQP is first solved with real-valued variables. While a node of the graph is not integer-feasible, branching continues. When all sub-problems are compatible with the original formulation of the MIQP problem, the node with the minimal value of the cost function is selected as solution.

APPENDIX B

Linear constraints for the Mixed-Integer walking model

This appendix describes some of the major constraints the mixed-integer model introduced in Sec. 5 is subject to. In order to result in a computationally-efficient formulation of the balance and walking MPC problem, the inputs of this model are chosen to write these constraints in a linear form as demonstrated in this appendix.

Notation

In this appendix, $\bullet_{x_i}$ denotes the scalar discretized value of the vector variable $\bullet$ in the longitudinal direction x at the discrete instant i. Direction y is the lateral one, orthogonal to x and in the plane of the ground.
Variables are named and denoted as in Sec. 5.

Definitions

The descriptors of the BoS introduced[1] in Sec. 5.1 are recalled as follows:

- $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^2 \times \mathbb{R}^2$ are the upper and lower bounds of the feet centers;
- $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \{0,1\}^2 \times \{0,1\}^2$ are the respective rising/falling edges of $\boldsymbol{a}$ and $\boldsymbol{b}$;
- $\gamma \in \{0,1\}$ differentiates SS phases from DS ones ($\gamma = 1 \Leftrightarrow$ DS);
- $\delta \in \{0,1\}$ distinguishes configurations of the feet with respect to the bounds.

B.1. Description constraints

These constraints are part of the definition of the descriptors of the model.

B.1.1. Shape constraints

Shape constraints describe the piecewise-constant evolution of the variables $(\boldsymbol{a}, \boldsymbol{b})$ describing the shape of the base of support.

Bounding

The definition of $(\boldsymbol{a}, \boldsymbol{b})$ as upper and lower bounds, respectively, of the BoS writes

$$\forall i, \quad \begin{cases} b_{x_i} \leq a_{x_i}, \\ b_{y_i} \leq a_{y_i}. \end{cases}$$

[1] *cf.* p. 109

Constancy

By definition of the variables $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, a discontinuity in $\boldsymbol{a}$ (*resp.* $\boldsymbol{b}$) is only possible if $\boldsymbol{\alpha} = \mathbf{1}$ (*resp.* $\boldsymbol{\beta} = \mathbf{1}$).) In other terms, $\boldsymbol{\alpha} = \mathbf{0}$ (*resp.* $\boldsymbol{\beta} = \mathbf{0}$) leads to constancy of $\boldsymbol{a}$ (*resp.* $\boldsymbol{b}$). Considering an upper bound $\boldsymbol{s} \in \mathbb{R}^2$ of the discontinuities in $\boldsymbol{a}$ and $\boldsymbol{b}$, it writes

$$\forall i, \quad \begin{cases} |a_{x_{i+1}} - a_{x_i}| & \leq & \alpha_{x_{i+1}} s_x, \\ |b_{x_{i+1}} - b_{x_i}| & \leq & \beta_{x_{i+1}} s_x, \\ |a_{y_{i+1}} - a_{y_i}| & \leq & \alpha_{y_{i+1}} s_y, \\ |b_{y_{i+1}} - b_{y_i}| & \leq & \beta_{y_{i+1}} s_y, \end{cases}$$

and thus in linear form

$$\forall i, \quad \begin{cases} a_{x_{i+1}} - a_{x_i} & \leq & \alpha_{x_{i+1}} s_x, \\ a_{x_i} - a_{x_{i+1}} & \leq & \alpha_{x_{i+1}} s_x, \\ b_{x_{i+1}} - b_{x_i} & \leq & \beta_{x_{i+1}} s_x, \\ b_{x_i} - b_{x_{i+1}} & \leq & \beta_{x_{i+1}} s_x, \end{cases} \tag{B.1}$$

and similarly in the y direction.

Simultaneity

Discontinuities of $\boldsymbol{a}$ and $\boldsymbol{b}$ resulting from a shift in the contact configuration, it is required to guarantee that allowing a discontinuity of one of the bounds in one direction simultaneously allows a discontinuity in the orthogonal direction. This requirement writes

$$\forall i, \quad \alpha_{x_i} + \beta_{x_i} = \alpha_{y_i} + \beta_{y_i}. \tag{B.2}$$

Sequentiality

A change in the contact configuration implies that in a given direction, only one of the boudns $\boldsymbol{a}$ or $\boldsymbol{b}$ can change at a given time Therefore,

$$\forall i, \quad \begin{cases} \alpha_{x_i} + \beta_{x_i} & \leq & 1, \\ \alpha_{y_i} + \beta_{y_i} & \leq & 1, \end{cases} \tag{B.3}$$

B.1.2. Admissibility constraints

Admissibility constraints define the evolution of redundant variables γ and δ with respect to the other descriptors of the BoS and its changes.

Single and double support alternation

Biped walking consists in changing the contact configuration with the alternation between simple and double support phases. Indeed, sliding motions are not considered in this model. It is therefore required that a discontinuity in either $\boldsymbol{a}$

or $\boldsymbol{b}$ in any direction induces a non-null discontinuity in γ. According to equation (B.2), the definition of this constraint on γ can be written with respect to the values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in any of the directions x or y. That is,

$$\left\{\begin{array}{rcll} \gamma_{i+1} & = & 0 & \text{if } \alpha_{x_{i+1}} + \beta_{x_{i+1}} = 1 \text{ and } \gamma_i = 1, \\ \gamma_{i+1} & = & 1 & \text{if } \alpha_{x_{i+1}} + \beta_{x_{i+1}} = 1 \text{ and } \gamma_i = 0, \\ \gamma_{i+1} & = & \gamma_i & \text{if } \alpha_{x_{i+1}} + \beta_{x_{i+1}} = 0, \end{array}\right.$$

which can be rewritten, as $\alpha_x + \beta_x \in \{0, 1\}$ according to equation (B.3) and recalling that $\gamma \in \{0, 1\}$

$$\forall i, \quad \left\{\begin{array}{rcl} \gamma_{i+1} + \gamma_i + \alpha_{x_{i+1}} + \beta_{x_{i+1}} & \leq & 2, \\ \alpha_{x_{i+1}} + \beta_{x_{i+1}} - \gamma_i & \leq & \gamma_{i+1}, \\ |\gamma_{i+1} - \gamma_i| & \leq & \alpha_{x_{i+1}} + \beta_{x_{i+1}}, \end{array}\right.$$

which finally takes the linear form

$$\forall i, \quad \left\{\begin{array}{rcl} \gamma_{i+1} + \gamma_i + \alpha_{x_{i+1}} + \beta_{x_{i+1}} & \leq & 2, \\ \alpha_{x_{i+1}} + \beta_{x_{i+1}} - \gamma_i & \leq & \gamma_{i+1}, \\ \gamma_{i+1} - \gamma_i & \leq & \alpha_{x_{i+1}} + \beta_{x_{i+1}}, \\ \gamma_i - \gamma_{i+1} & \leq & \alpha_{x_{i+1}} + \beta_{x_{i+1}}. \end{array}\right.$$

Single support

By definition of the binary variable γ, and with the definition of $\boldsymbol{a}$ and $\boldsymbol{b}$, a single support phase — that is, $\gamma = 0$ — imposes that $\boldsymbol{a}$ and $\boldsymbol{b}$ are equal. With $\boldsymbol{s} \in \mathbb{R}^2$ an upper bound of the difference $\boldsymbol{a} - \boldsymbol{b}$, it writes

$$\forall i, \quad \left\{\begin{array}{rcl} |a_{x_i} - b_{x_i}| & \leq & \gamma_i s_x, \\ |a_{y_i} - b_{y_i}| & \leq & \gamma_i s_y, \end{array}\right.$$

which takes the linear form

$$\forall i, \quad \left\{\begin{array}{rcl} a_{x_i} - b_{x_i} & \leq & \gamma_i s_x, \\ b_{x_i} - a_{x_i} & \leq & \gamma_i s_x, \\ a_{y_i} - b_{y_i} & \leq & \gamma_i s_y, \\ b_{y_i} - a_{y_i} & \leq & \gamma_i s_y. \end{array}\right.$$

Contact configuration history

In order to keep the information from an SS→DS transition, constraining the next DS→SS shift, the contact configuration descriptor δ must remain constant over each DS phase. This requirement writes

$$\forall i, \quad |\delta_{i+1} - \delta_i| \leq 1 - \gamma_i,$$

which is written in linear form

$$\forall i, \quad \left\{\begin{array}{rcl} \delta_{i+1} - \delta_i + \gamma_i & \leq & 1, \\ \delta_i - \delta_{i+1} + \gamma_i & \leq & 1. \end{array}\right. \tag{B.4}$$

Contact configuration enforcement

At time step i, bringing an SS→DS transition ($1 \geq \gamma_{i+1} > \gamma_i \geq 0$), the constraint (B.4) is not active ($\gamma_i = 0$). A variation of δ is therefore possible and must be restrained in order to capture the contact configuration adopted in the incoming DS phase. Conversely, during the DS→SS transition ($0 \leq \gamma_{i+1} < \gamma_i \leq 1$), the constraint (B.4) is active ($\gamma_i = 1$) and must therefore restrain the variations of the bounds $\boldsymbol{a}$ and $\boldsymbol{b}$, that is $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. A configuration of type $\delta = 1$ imposes that the bounds vary simultaneously in the two directions , therefore

$$\forall i, \quad \begin{cases} |\alpha_{x_i} - \alpha_{y_i}| & \leq & 1 - \delta_i, \\ |\beta_{x_i} - \beta_{y_i}| & \leq & 1 - \delta_i, \end{cases}$$

and it can be noticed that this constraint is not active for the other type of configuration $\delta = 0$. In linear form, it writes

$$\forall i, \quad \begin{cases} \alpha_{x_i} - \alpha_{y_i} + \delta_i & \leq & 1, \\ \alpha_{y_i} - \alpha_{x_i} + \delta_i & \leq & 1, \\ \beta_{x_i} - \beta_{y_i} + \delta_i & \leq & 1, \\ \beta_{y_i} - \beta_{x_i} + \delta_i & \leq & 1. \end{cases} \tag{B.5}$$

Similarly, a configuration of type $\delta = 0$ imposes that a pair of discontinuities (α_x, β_y) or (β_x, α_y) are subject to simultaneous changes, which writes

$$\forall i, \quad \begin{cases} |\alpha_{x_i} - \beta_{y_i}| & \leq & \delta_i, \\ |\beta_{x_i} - \alpha_{y_i}| & \leq & \delta_i, \end{cases}$$

and it can be noticed that this constraint is not active for $\delta = 1$. It takes the linear form

$$\forall i, \quad \begin{cases} \alpha_{x_i} - \beta_{y_i} - \delta_i & \leq & 0, \\ \beta_{y_i} - \alpha_{x_i} - \delta_i & \leq & 0, \\ \beta_{x_i} - \alpha_{y_i} - \delta_i & \leq & 0, \\ \alpha_{y_i} - \beta_{x_i} - \delta_i & \leq & 0. \end{cases} \tag{B.6}$$

B.2. Walking constraints

The ability to write constraints related to the walking motion enables a refined definition of feasibility of changes in the BoS from its mixed-integer abstraction.

Maximum foot average velocity

The mixed-integer model not capturing the instantaneous motion of the swinging foot, but rather its target position and swinging duration, instantaneous foot velocity cannot be constrained. Nevertheless, the average foot velocity for this swinging motion, defined as the ratio between the traveled distance and the duration of the step, can be bounded. Let $\boldsymbol{v} \in \mathbb{R}^2$ denote the upper constraint of this velocity. According to the sequentiality constraint (B.3) of contact shifts, variations of $\boldsymbol{r} \triangleq \frac{\boldsymbol{a}+\boldsymbol{b}}{2}$ allow to capture the distance traveled during the swinging motion. The bound $\boldsymbol{v}$ restrains the duration of a single support phase ($\gamma = 0$)

with respect to the variations of $\boldsymbol{r}$ during this phase. In other terms, $\boldsymbol{v}$ bounds, according to the duration of the SS phase, the difference in position $\boldsymbol{r}$ between the DS phases surrounding it.
Let d_k^+ denote the last index of the k^{th} DS phase, and d_k^- its first. The velocity constraint therefore writes, in the x direction for example

$$\frac{\left| r_{x_{d_{k+1}^-}} - r_{x_{d_k^+}} \right|}{\left(d_{k+1}^- - d_k^+ \right) dt} \leq \frac{v_x}{2}, \tag{B.7}$$

where $|r_{x_{d_{k+1}^-}} - r_{x_{d_k^+}}|$ is half of the traveled distance during the step, and $\left(d_{k+1}^- - d_k^+ \right) dt$ its duration. Recalling that the variable $\boldsymbol{r}$ does not vary during DS phases and denoting n_d the total number of DS phases over the preview horizon, it can be shown

$$\begin{array}{c} \forall i \quad \text{s.t.} \quad \gamma_i = 1, \ \forall j > i \quad \text{s.t.} \quad \gamma_j = 1, \\ \exists\, (k,l) \in [1, n_d]^2 \quad \text{s.t.} \quad \left\{ l \geq k, i \in \left[d_k^-, d_k^+ \right], j \in \left[d_l^-, d_l^+ \right] \right\} \end{array}$$

and moreover

$$\begin{aligned} r_{x_j} - r_{x_i} &= (r_{x_j} - r_{x_{d_k^+}}) + (r_{x_{d_k^+}} - r_{x_i}) = (r_{x_{d_l^-}} - r_{x_j}) + (r_{x_j} - r_{x_{d_k^+}}) \\ &= r_{x_{d_l^-}} - r_{x_{d_k^+}} = r_{x_{d_l^-}} - r_{x_{d_k^+}} + \sum_{h=k+1}^{l-1} r_{x_{d_h^-}} - r_{x_{d_h^+}} \\ &= \sum_{h=k}^{l-1} r_{x_{d_{h+1}^-}} - r_{x_{d_h^+}}. \end{aligned}$$

This brings, taking the absolute value and assuming (B.7) to be verified

$$\left| r_{x_j} - r_{x_i} \right| \leq \sum_{h=k}^{l-1} \left| r_{x_{d_{h+1}^-}} - r_{x_{d_h^+}} \right| \Rightarrow \left| r_{x_j} - r_{x_i} \right| \leq \frac{v_x}{2} dt \sum_{h=k}^{l-1} \left(d_{h+1}^- - d_h^+ \right).$$

However it is recalled that

$$i \leq d_k^+, \quad j \geq d_l^- \quad \text{and} \quad \sum_{h=k}^{l-1} \left(d_{h+1}^- - d_h^+ \right) \leq j - i.$$

Finally, if $\gamma_i = \gamma_j = 1$

$$\begin{array}{c} \forall i \quad \text{s.t.} \quad \gamma_i = 1, \ \forall j > i \quad \text{s.t.} \quad \gamma_j = 1, \\ \dfrac{\left| r_{x_j} - r_{x_i} \right|}{(j-i)\, dt} \leq \dfrac{v_x}{2} \end{array} \tag{B.8}$$

Furthermore, it can be noted that (B.1) brings

$$\forall j > i, \quad \left| r_{x_j} - r_{x_i} \right| \leq (j-i)\, s_x$$

and therefore

$$\forall i, \ \forall j \neq i, \ \frac{\left| r_{x_j} - r_{x_i} \right|}{(j-i)\, dt} \leq \frac{s_x}{dt}.$$

The inequality (B.8) can henceforth be written in the general case

$$\forall i,\ \forall j > i, \quad \frac{\left| r_{x_j} - r_{x_i} \right|}{(j-i)\,dt} \leq \frac{v_x}{2} + (2 - \gamma_i - \gamma_j)\,\frac{s_x}{dt}, \tag{B.9}$$

which, written this way, solely restrains the evolution of $\boldsymbol{r}$ in the case where $\gamma_i = \gamma_j = 1$ (and is thus equivalent to (B.8)). The constraint (B.9) takes the linear form

$$\forall i,\ \forall j > i, \begin{cases} \dfrac{a_{x_j} - a_{x_i} + b_{x_j} - b_{x_i}}{(j-i)\,dt} \leq v_x + 2(2 - \gamma_i - \gamma_j)\dfrac{s_x}{dt} \\ \dfrac{a_{x_i} - a_{x_j} + b_{x_i} - b_{x_j}}{(j-i)\,dt} \leq v_x + 2(2 - \gamma_i - \gamma_j)\dfrac{s_x}{dt} \end{cases} \tag{B.10}$$

and similarly in the y direction.

It can be noted that in the case of an ongoing SS phase, enforcement of condition (B.7) requires to keep track of the contact configuration of the previous DS phase. The constraint (B.9) must therefore be written not only over the preview horizon, but also with i being in the past. More precisely, the index i considered must start from the last known DS configuration. For example, if the control problem is solved at step k, which corresponds to a SS phase (*i.e.* γ_k=0), and with an horizon of N time steps, the index i should take the values

$$i \in [i_k, k+N-1] \quad \text{s.t.} \quad \begin{cases} i_k & < & k, \\ \gamma_{i_k} & = & 1, \\ \nexists j_k \in\,]i_k, k] & / & \gamma_{j_k} = 1. \end{cases}$$

Minimum SS and DS phases durations

Let d_{SS} and d_{DS} denote the maximum number of time steps over which SS and DS phases can spread, respectively. For SS phases, this implies that as soon as a SS phase begins ($\gamma_i = 1$ and $\gamma_{i+1} = 0$), the variable γ must be null during at least d_{SS} time steps. This constraint can be translated as

$$\text{if } \gamma_{i+1} = 0 \text{ and } \gamma_i = 1, \quad \sum_{j=i+1}^{j \leq i+d_{SS}} \gamma_j \leq 0.$$

It can be generalized to its equivalent

$$\text{if } \gamma_i = 1, \quad \sum_{j=i+1}^{j \leq i+d_{SS}} \gamma_j \leq d_{SS}\gamma_{i+1},$$

which is indeed not active if $\gamma_{i+1} \neq 0$ (*cf.* $\gamma \in \{0, 1\}$). Similarly, it can be further generalized into

$$\forall i, \quad \sum_{j=i+1}^{j \leq i+d_{SS}} \gamma_j \leq d_{SS}\,(1 - \gamma_i + \gamma_{i+1})\,,$$

which is effectively active if and only if $\gamma_{i+1} = 0$ and $\gamma_i = 1$. A shift in indices concludes

$$\forall i, \quad \sum_{j=i}^{j\leq i+d_{SS}-1} \gamma_j \leq d_{SS}\left(1+\gamma_i-\gamma_{i-1}\right). \tag{B.11}$$

An identical development brings the minimum duration constraint of DS phases

$$\forall i, \quad \sum_{j=i}^{j\leq i+d_{DS}-1} \gamma_j \geq d_{DS}\left(\gamma_i-\gamma_{i-1}\right). \tag{B.12}$$

Similarly to the velocity constraint, constraints (B.11) and (B.12) require to keep track of a previous history. Indeed, constraining the duration of SS phases require to write the constraint (B.11) with the starting index i corresponding to the first index of the current SS phase, and conversely for DS phases.

Maximum leg span

Constraining the maximum leg span is straightforward with the descriptors chosen in this formulation. Indeed, the difference between the upper and lower bounds $\boldsymbol{a}$ and $\boldsymbol{b}$ directly translates the current leg span in both directions. Defining an upper bound of the difference $(\boldsymbol{a}-\boldsymbol{b})$ therefore defines a directional maximum leg span. With $\boldsymbol{s}\in\mathbb{R}^2$ this upper bound, it writes

$$\forall i, \quad \boldsymbol{a}_i-\boldsymbol{b}_i \preceq \boldsymbol{s}.$$

However, defining a leg span as a maximum distance between the feet is not directly possible in a linear form (this is indeed equivalent to defining an upper bound on the square norm of $(\boldsymbol{a}-\boldsymbol{b})$) . A linearization of this constraint can nevertheless be considered if required.

Non-overlapping feet

The definition of a non-overlapping constraint can be approached in a similar manner to the maximum leg span: defining a lower bound of the difference $(\boldsymbol{a}-\boldsymbol{b})$ indeed allows to define a rectangular inadmissible domain around the center of the feet. Since $\boldsymbol{a}=\boldsymbol{b}$ during SS phases, this constraint should be active during DS phases solely. With $\boldsymbol{L}\in\mathbb{R}^2$ the dimensions of the inadmissible area around the center of the feet, this constraint writes

$$\forall i, \quad \boldsymbol{a}_i-\boldsymbol{b}_i \preceq \gamma_i\boldsymbol{L}+(1-\gamma_i)\boldsymbol{s},$$

with $\boldsymbol{s}$ the upper bound of the difference $(\boldsymbol{a}-\boldsymbol{b})$, this constraint with respect to the lower bound $\boldsymbol{L}$ is therefore active if and only if $\gamma_i=1$ (DS phase).
Similarly to the maximum leg span constraint, if a disc domain is to be considered, a linearization can be considered.

Non-crossing legs

Since the mixed-integer model does not account for the motion of the leg, some changes in the BoS might bring a crossing of the legs. While the resulting collisions between the lower limbs can be avoided with an additional controller (at the instantaneous level, for example), strictly forbidding contact configurations

implying a crossing of the legs allows a stricter definition of feasibility. Due to the structure of humanoids, legs crossing brings collision issues in the lateral direction mainly. Avoiding such configurations can be achieved by imposing that if a change in the upper or the lower bound was performed to trigger an SS phase, the same bound solely must change when recovering a DS phase. This translates, for the upper bound

$$\forall i, \forall j < i, \text{ if } \gamma_i = 1 \text{ and } \gamma_{j-1} \text{ and } \forall k \in [j, i-1] \ \gamma_k = 0, \quad \alpha_{y_i} = \alpha_{y_j}.$$

This can be generalized into

$$\forall i, \forall j < i, \quad \left|\alpha_{y_i} - \alpha_{y_j}.\right| \leq \left(\sum_{j \leq k < i} \gamma_k\right) + (1 - \gamma_i) + (1 - \gamma_{j-1}),$$

and similarly for changes β_y in the lower bound.

APPENDIX C

On multi-rate model inconsistency

Results from the multi-rate implementation[1] on TORO, although demonstrating the feasibility of obtaining coordination strategies between walking and balance objectives and constraints in a real-time framework, exhibit tracking performance issues.

Potential causes of such issues are identified as inconsistencies between the three successive balance controllers[2] employed :

- at a slow-rate at the predictive level (MIQP controller),
- at a faster rate at the predictive level (Extended ZMP Preview controller),
- at a faster rate at the instantaneous level (ZMP controller).

C.1. Inconsistent tracking

While inconsistencies between the second and the third controller at the faster rate — induced by their respective use of dynamic and kinematic models — have a considerable influence on the overall tracking performance, inconsistencies between the slow and the fast rates have a more noticeable impact on the resulting behavior of the robot. Indeed, discrete variables are exploited at the slower rate. Inconsistencies between the MIQP controller[3] and the combination of the controllers of the faster rate are therefore addressed in this appendix.

Inconsistencies between these rates are induced by the model reduction performed at the faster rate, the Extended ZMP Preview controller indeed not considering changes in the BoS as variables, and by the differences in time discretization of the preview horizon to allow for fast resolution of the MIQP controller.

Both MPC controllers output optimal control strategies with respect to a previewed, expected evolution of the system outputs under this control policy. If the resulting, actual evolution of the system is different from the previously previewed one, the newly computed control strategy is potentially inconsistent with the previous.

One of the differences in evolution of system outputs is a slow tracking at the instantaneous level of the CoP path previewed by the MIQP controller. As illustrated in figure C.1, this can result in a delay of the previous control strategy from the MIQP. Indeed, the change in constraints induced by the previewed control strategy are not compatible with the actual evolution of the system output: however, delaying the strategy allows to maintain feasibility.

A second difference can be expected from the distinct time discretization between the MIQP controller and the Extended ZMP Preview controller. The former indeed exploits a coarser time discretization of its preview horizon, to

[1] *cf.* Sec. 6.1, p. 132
[2] *cf.* figure 6.5, p. 136
[3] *cf.* Sec. 5.2, p. 114

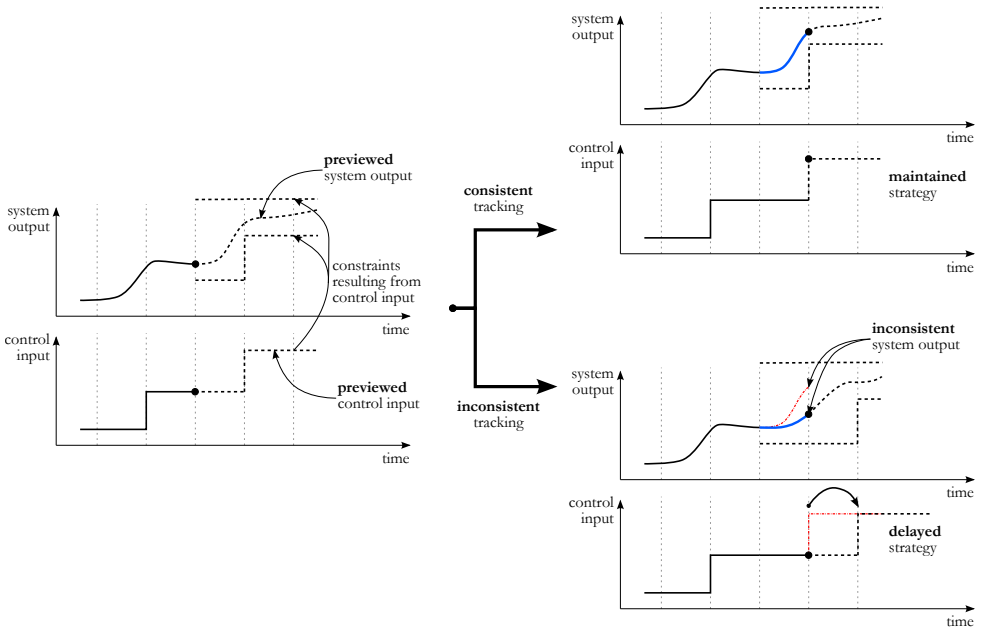

Figure C.1.: Illustration of the effect of inconsistent tracking between the fast and slow rates. If the actual evolution of the system output is not consistent with the evolution previously previewed at the higher level (slower rate), delaying the strategy can be a solution to maintain feasibility.

reduce computational cost. The Extended ZMP Preview controller commonly outputs strategies illustrated in figure C.2. Short-term actions such as the anticipatory impetus depicted in figure C.2, exploited to trigger motion of the CoM in the desired direction, are not fully captured with a coarser time discretization of the preview horizon. As a result, state inputs fed back to the MIQP controller can be considered as inconsistent with the expected evolution, and thus induce a significant change in the discrete strategy to be adopted.

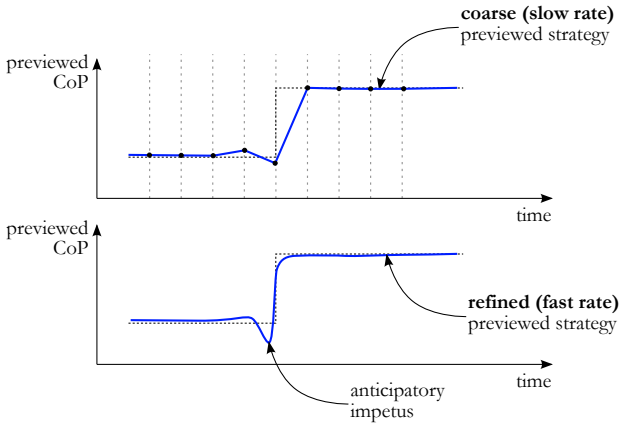

Figure C.2.: Illustration of the possible source of inconsistencies between the fast and slow rates. The coarser time discretization of the slower rate do not allow to capture short-term effects. These effects can be interpreted at the slow rate as disturbances and induce changes in the control strategy.

For example, the anticipatory impetus driven by the low level ZMP Preview controller can be interpreted at the higher level as a disturbance on the CoM, which can lead to the decision of a stepping strategy in the direction of this disturbance to recover balance, which is opposed to the previously computed direction.
These significant changes in discrete strategies can henceforth lead to oscillations as observed in Sec. 6.1. Such oscillations induce a temporary instability of the controller, which slows down the convergence of the system outputs towards the desired, expected targets, therefore inducing delays in the adopted strategies as previously discussed.

C.2. Effects of delayed preview strategies

The effect of delayed discrete strategies is depicted in figure C.3. In the ideal case, without disturbances, previewed strategies are expected to be consistent between each other: new strategies should cover the previous one, and solely append new solutions at the end of the previous horizon. The resulting actual strategy, consisting in the application of the first element of each previewed strategy over time, is consistent with all previewed horizons.

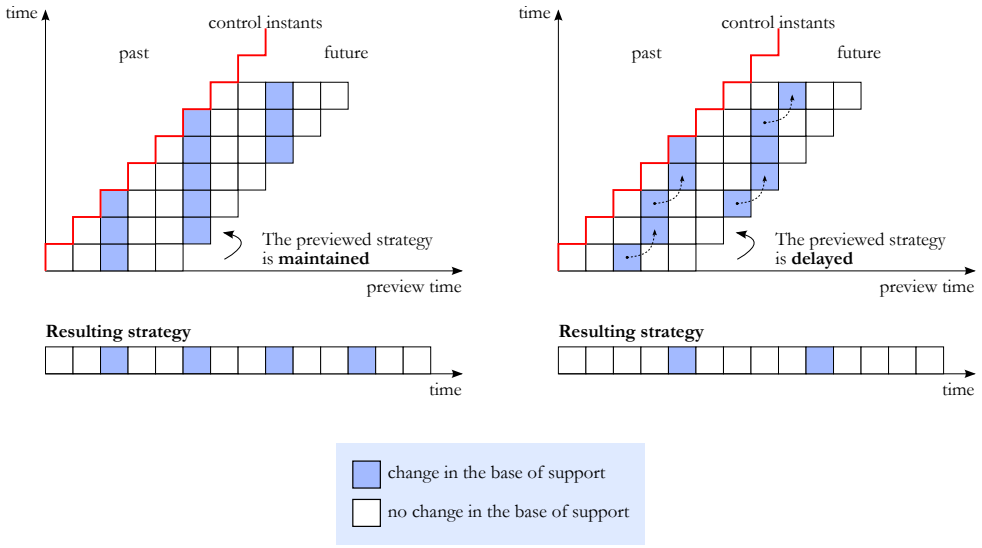

Figure C.3.: Illustration of the effect of delayed previewed strategies on the resulting applied strategy.

However, if these previewed strategies are sometimes delayed, because of tracking inconsistencies for example as previously discussed, the successive horizons are not stackable. The resulting actual strategy is inconsistent with the previewed ones. As an example, all previewed strategies in figure C.3 expect two idle time steps between each change in the base of support. Despite this consistency in the computation of solutions, in the presence of delays, the resulting applied strategy is significantly different: more than two idle steps indeed separates discrete changes.

This illustration can be related to the results observed[4] in Sec. 6.1.4: in the nominal case, and despite a correct behavior being exhibited[5] with the single-rate simulation, double support phases are indeed much longer with the multi-rate implementation than expected. An extract of the evolution of the previewed strategies for the experimental results of Sec. 6.1.4 is represented in figure C.4. It can be noticed that incoming DS phases are delayed, while SS→DS changes are not: as observed in Sec. 6.1.4, the resulting DS phases are longer than expected although the duration of SS phases stay consistent with the objective.

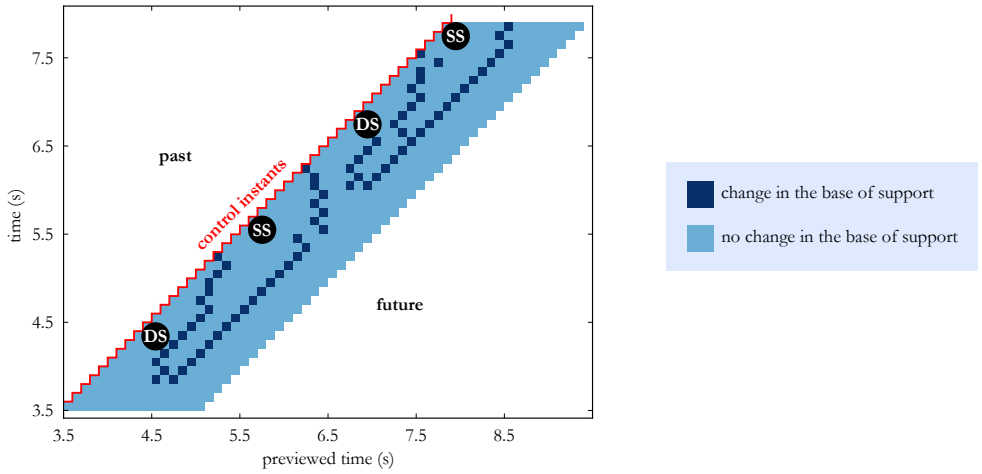

Figure C.4.: Extract of experimental results on inconsistent previewed strategies. The previewed strategies are not stackable. This inconsistency is mainly induced by the control strategies being delayed, especially for DS→SS transitions.

DS→SS transitions indeed induce a reduction of the admissible domain of the CoP: an inconsistent — too slow, for example — tracking therefore prevent such transitions to effectively occur at the expected time; such strategies are delayed consequently. Conversely, SS→DS changes benefit from an increase in the admissible domain, and are thus less subject to delays.

[4] *cf.* p. 140
[5] *cf.* Sec. 5.3, p. 117

Bibliography

[Abe2007] Yeuhi Abe, Marco da Silva, and Jovan Popović. Multiobjective control with frictional contacts. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 249–258. Eurographics Association, 2007.

[Afnor2008] AFNOR. Nf en iso 15536 ergonomics - computer manikins and body templates. Association française de normalisation (AFNOR), 2008.

[Arrow1958] Kenneth Joseph Arrow, Leonid Hurwicz, and Hirofumi Uzawa. *Studies in linear and non-linear programming*. Stanford University Press, 1958.

[Asimo2011] Asimo 2011 key specifications. `http://world.honda.com/ASIMO/technology/2011/specification`.

[Barthelemy2008] Sébastien Barthélemy and Philippe Bidaud. Stability measure of postural dynamic equilibrium based on residual radius. In *Advances in Robot Kinematics: Analysis and Design*, pages 399–407. Springer, 2008.

[BarthelemyArboris] S. Barthelemy, J. Salini, and A. Micaelli. Arboris-python.

[Bemporad1999] Alberto Bemporad and Manfred Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.

[Bemporad2002a] Alberto Bemporad, Francesco Borrelli, Manfred Morari, et al. Model predictive control based on linear programming˜ the explicit solution. *IEEE Transactions on Automatic Control*, 47(12):1974–1985, 2002.

[Bemporad2002b] Alberto Bemporad, WP Maurice H Heemels, and Bart De Schutter. On hybrid systems and closed-loop mpc systems. *IEEE Transactions on Automatic Control*, 47(5):863–869, 2002.

[Boyd2004] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[BuendiaSpirops] Axel Buendia. Spirops ai, a behavior production tool. `http://www.spirops.com`.

[Camponogara2002] Eduardo Camponogara, Dong Jia, Bruce H. Krogh, and Sarosh Talukdar. Distributed model predictive control. In *Control Systems*, volume 1, pages 44–52. IEEE 22, 2002.

[Cheng1997] X. Cheng and B.H. Krogh. Stability-constrained model predictive control with state estimation. In *Proceedings of the American Control Conference*, pages 2493–2498, 1997.

[Christofides2013] Panagiotis D Christofides, Riccardo Scattolini, David Munoz de la Pena, and Jinfeng Liu. Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51:21–41, 2013.

[Clarke1987] David W Clarke, C Mohtadi, and PS Tuffs. Generalized predictive control –part i. the basic algorithm. *Automatica*, 23(2):137–148, 1987.

[Cotton2011] Sébastien Cotton, Michele Vanoncini, Philippe Fraisse, Nacim Ramdani, Emel Demircan, AP Murray, and T Keller. Estimation of the centre of mass from motion capture and force plate recordings: a study on the elderly. *Applied Bionics and Biomechanics*, 8(1):67–84, 2011.

[Cutler1980] C. Cutler and B. L. Ramaker. Dynamic matrix control-a computer control algorithm. In *Proceedings of the joint automatic control conference*, volume 1, pages Wp5–B, 1980.

[Cutler1983] C. Cutler, A. Morshedi, and J. Haydel. An industrial perspective on advanced control. In *American Institute of Chemical Engineers Annual meeting*, 1983.

[Dantzig1961] George B. Dantzig and Philip Wolfe. The decomposition algorithm for linear programs. *Econometrica: Journal of the Econometric Society*, 4:767–778, 1961.

[DeSantis2008] Agostino De Santis, Bruno Siciliano, Alessandro De Luca, and Antonio Bicchi. An atlas of physical human–robot interaction. *Mechanism and Machine Theory*, 43(3):253–270, 2008.

[Demircan2008] Emel Demircan, Luis Sentis, Vincent De Sapio, and Oussama Khatib. Human motion reconstruction by direct control of marker trajectories. In *Advances in Robot Kinematics: Analysis and Design*, pages 263–272. Springer, 2008.

[Demircan2012] Emel Demircan, Thor F Besier, and Oussama Khatib. Muscle force transmission to operational space accelerations during elite golf swings. In *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 1464–1469. IEEE, 2012.

[Diedam2008] Holger Diedam, Dimitar Dimitrov, P-B Wieber, Katja Mombaur, and Moritz Diehl. Online walking gait generation with adaptive foot positioning through linear model predictive control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems*, pages 1121–1126. IEEE, 2008.

[Englsberger2011] Johannes Englsberger, Christian Ott, Maximo A Roa, A Albu-Schaffer, and Gerhard Hirzinger. Bipedal walking control based on capture point dynamics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems*, pages 4420–4427. IEEE, 2011.

[Englsberger2014] Johannes Englsberger, Alexander Werner, Christian Ott, Bernd Henze, Maximo A Roa, Gianluca Garofalo, Robert Burger, Alexander Beyer, Oliver Eiberger, Korbinian Schmid, et al. Overview of the torque-controlled humanoid robot toro. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robotics*, pages 916–923, 2014.

[Enoch2012] Alexander Enoch, Andrius Sutas, Shinichiro Nakaoka, and Sethu Vijayakumar. Blue: A bipedal robot with variable stiffness and damping. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robotics*, pages 487–494. IEEE, 2012.

[Escande2014] Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, page 0278364914521306, 2014.

[Espiau1990] Bernard Espiau, François Chaumette, and Patrick Rives. Une nouvelle approche de la relation vision-commande en robotique. Rapport de recherche RR-1172, INRIA, 1990.

[Franklin1994] Gene F Franklin, J David Powell, and Abbas Emami-Naeini. Feedback control of dynamics systems. 1994.

[Garcia1989] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: theory and practice –a survey. *Automatica*, 25(3):335–348, 1989.

[Geoffroy2014] Perle Geoffroy, Nicolas Mansard, Maxime Raison, Sofiane Achiche, and Emo Todorov. From inverse kinematics to optimal control. In *Advances in Robot Kinematics*, pages 409–418. Springer, 2014.

[Giselsson2010] Pontus Giselsson and Anders Rantzer. Distributed model predictive control with suboptimality and stability guarantees. In *Proceedings of the IEEE Conference on Decision and Control*, pages 7272–7277. IEEE, 2010.

[Goswami1999] Ambarish Goswami. Postural stability of biped robots and the foot-rotation indicator (fri) point. *The International Journal of Robotics Research*, 18(6):523–533, 1999.

[Granata2015] C. Granata, A. Ibanez, and P. Bidaud. Human activity understanding: a multilayer approach combining body movements and contextual descriptors analysis. *International Journal of Advanced Robotic Systems*, 12, July 2015.

[Gurobi] Gurobi Optimization Inc. http://www.gurobi.com.

[Herdt2010] Andrei Herdt, Holger Diedam, Pierre-Brice Wieber, Dimitar Dimitrov, Katja Mombaur, and Moritz Diehl. Online walking motion generation with automatic footstep placement. In *Advanced Robotics*, volume 24, pages 719–737, 2010.

[Hogan84] Neville Hogan. Impedance control: An approach to manipulation. In *American Control Conference*, 1984.

[Hommel2009] Bernhard Hommel. Action control according to tec (theory of event coding). *Psychological Research PRPF*, 73(4):512–526, 2009.

[Hovland2006] S Hovland, K Willcox, and JT Gravdahl. Mpc for large-scale systems via model reduction and multiparametric quadratic programming. In *Proceedings of the IEEE Conference on Decision and Control*, pages 3418–3423. IEEE, 2006.

[Ibanez2012] Aurelien Ibanez, Philippe Bidaud, and Vincent Padois. Unified preview control for humanoid postural stability and upper-limb interaction adaptation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems*, pages 1801–1808. IEEE, 2012.

[Ibanez2014a] Aurelien Ibanez, Philippe Bidaud, and Vincent Padois. A distributed model predictive control approach for robust postural stability of a humanoid robot. In *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 202–209. IEEE, 2014.

[Ibanez2014b] Aurelien Ibanez, Philippe Bidaud, and Vincent Padois. Emergence of humanoid walking behaviors from mixed-integer model predictive control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems*, pages 4014–4021. IEEE, 2014.

[Kajita2003] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Kajiwara, Kensuke Harada, Kazuhito Yokoi, and Hirohisa Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *Proceedings of the IEEE International Conference on Robotics & Automation*, 2003.

[Kale2005] MM Kale and AJ Chipperfield. Stabilized mpc formulations for robust reconfigurable flight control. *Control Engineering Practice*, 13(6):771–788, 2005.

[Kanoun2009] Oussama Kanoun, Florent Lamiraux, Pierre-Brice Wieber, Fumio Kanehiro, Eiichi Yoshida, and Jean-Paul Laumond. Prioritizing linear equality and inequality systems: Application to local motion planning for redundant robots. In *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 2939–2944, may 2009.

[Kanzaki2005] Shigeru Kanzaki, Kei Okada, and Masayuki Inaba. Bracing behavior in humanoid through preview control of impact disturbance. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robotics*, pages 301–305. IEEE, 2005.

[Kerrigan2001] Eric Colin Kerrigan. *Robust constraint satisfaction: Invariant sets and predictive control*. PhD thesis, University of Cambridge, 2001.

[Khatib1987] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation*, 3(1):43–53, 1987.

[Kim2011] Seyoung Kim and Sukyung Park. Leg stiffness increases with speed to modulate gait frequency and propulsion energy. *Journal of Biomechanics*, 44(7):1253–1258, 2011.

[Klanvcar2007] Gregor Klančar and Igor Škrjanc. Tracking-error model-based predictive control for mobile robots in real time. *Robotics and Autonomous Systems*, 55(6):460–469, 2007.

[Koppula2013] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013.

[Krause2012] Manuel Krause, Johannes Englsberger, Pierre-Brice Wieber, and Christian Ott. Stabilization of the capture point dynamics for bipedal walking based on model predictive control. In *Robot Control*, volume 10, pages 165–171, 2012.

[Lacquaniti1982] Francesco Lacquaniti and John F Soechting. Coordination of arm and wrist motion during a reaching task. *The Journal of Neuroscience*, 2(4):399–408, 1982.

[Lawler1966] Eugene L Lawler and David E Wood. Branch-and-bound methods: A survey. *Operations research*, 14(4):699–719, 1966.

[Lazar2007] Mircea Lazar, WPMH Heemels, Alberto Bemporad, and Siep Weiland. Discrete-time non-smooth nonlinear mpc: Stability and robustness. In *Assessment and Future Directions of Nonlinear Model Predictive Control*, pages 93–103. Springer, 2007.

[Lee2011] Jay H Lee. Model predictive control: review of the three decades of development. *International Journal of Control, Automation and Systems*, 9(3):415–424, 2011.

[Lim2001] Hun-ok Lim, Samuel A Setiawan, and Atsuo Takanishi. Balance and impedance control for biped humanoid robot locomotion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems*, volume 1, pages 494–499. IEEE, 2001.

[Liu2009] Jinfeng Liu, David Mu noz de la Peña, and Panagiotis D. Christofides. Distributed model predictive control of nonlinear process systems. *American Institute of Chemical Engineers Journal*, 55(5):1171–1184, 2009.

[Liu2014] Mingxing Liu, Yang Tan, and Vincent Padois. Generalized hierarchical control. *Autonomous Robots*, pages 1–15, 2014.

[Love95] L.J. Love and W.J. Book. Environment estimation for enhanced impedance control. In *Proceedings of the IEEE International Conference on Robotics & Automation*, volume 2, pages 1854 – 1859, 1995.

[Maki2003] Brian E Maki, William E Mcilroy, and Geoff R Fernie. Change-in-support reactions for balance recovery. *Engineering in Medicine and Biology Magazine, IEEE*, 22(2):20–26, 2003.

[Mansard2007] Nicolas Mansard and Francois Chaumette. Task sequencing for high-level sensor-based control. *IEEE Transactions on Robotics*, 23(1):60–72, 2007.

[Mansard2009] Nicolas Mansard, Oussama Khatib, and Abderrahmane Kheddar. A unified approach to integrate unilateral constraints in the stack of tasks. *IEEE Transactions on Robotics*, 25(3):670–685, 2009.

[Maurice2014] Pauline Maurice, Philipp Schlehuber, Vincent Padois, Philippe Bidaud, and Yvan Measson. Automatic selection of ergonomie indicators for the design of collaborative robots: A virtual-human in the loop approach. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robotics*, pages 801–808. IEEE, 2014.

[Maurice2015] Pauline Maurice. *Virtual ergonomics for the design of collaborative robots*. PhD thesis, Sorbonne Universités, UPMC Univ Paris 06, 2015.

[Merlhiot2012] Xavier Merlhiot, Jérémie Le Garrec, Guillaume Saupin, and Claude Andriot. The xde mechanical kernel: Efficient and robust simulation of multibody dynamics with intermittent nonsmooth contacts. In *The 2nd Joint International Conference on Multibody System Dynamics*, 2012.

[Miguel2013] Antonio F Miguel. The emergence of design in pedestrian dynamics: Locomotion, self-organization, walking paths and constructal law. *Physics of life reviews*, 2013.

[Mizrahi2015] Joseph Mizrahi. Mechanical impedance and its relations to motor control, limb dynamics, and motion biomechanics. *Journal of medical and biological engineering*, 35(1):1–20, 2015.

[Mordatch2012] Igor Mordatch, Emanuel Todorov, and Zoran Popović. Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics (TOG)*, 31(4):43, 2012.

[Morisawa2010] Mitsuharu Morisawa, Fumio Kanehiro, Kenji Kaneko, Nicolas Mansard, Joan Sola, Eiichi Yoshida, Kazuhiro Yokoi, and J Laumond. Combining suppression of the disturbance and reactive step-

ping for recovering balance. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems*, pages 3150–3156. IEEE, 2010.

[Murray1994] Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC PressI Llc, 1994.

[Nakamura2003] Yoshihiko Nakamura, Katsu Yamane, Ichiro Suzuki, and Yusuke Fujita. Dynamic computation of musculo-skeletal human model based on efficient algorithm for closed kinematic chains. In *Proceedings of the 2nd International Symposium on Adaptive Motion of Animals and Machines*, 2003.

[Nash1951] John Nash. Non-cooperative games. *The Annals of Mathematics*, 54(2):286–295, 1951.

[Nelder1965] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[Nenchev2008] Dragomir N Nenchev and Akinori Nishio. Ankle and hip strategies for balance recovery of a biped subjected to an impact. *Robotica*, 26(05):643–653, 2008.

[Osiadacz1990] Andrzej J Osiadacz. Integer and combinatorial optimization, george l. nemhauser and laurence a. wolsey, wiley-interscience series in discrete mathematics and optimization, new york, 1988, isbn 0-471-82819-x, 763pp,£ 71.90. *International Journal of Adaptive Control and Signal Processing*, 4(4):333–334, 1990.

[Paillard1994] J. Paillard. L'intégration sensori-motrice et idéomotrice. In M. Richelle, J. Requin, and M. Robert, editors, *Traité de Psychologie Expérimentale*, chapter III.6, pages 925–961. Presses Universitaires de France, 1994.

[Park2000] Jong Hyeon Park and Hyun Chul Cho. An online trajectory modifier for the base link of biped robots to enhance locomotion stability. In *Proceedings of the IEEE International Conference on Robotics & Automation*, volume 4, pages 3353–3358. IEEE, 2000.

[Park2001] Jong Hyeon Park. Impedance control for biped robot locomotion. *IEEE Transactions on Robotics and Automation*, 17(6):870–882, 2001.

[Prahlad2008] Vadakkepat Prahlad, Goswami Dip, and Chia Meng-Hwee. Disturbance rejection by online zmp compensation. *Robotica*, 26(1):9, 2008.

[Pratt2006] Jerry Pratt, John Carff, Sergey Drakunov, and Ambarish Goswami. Capture point: A step toward humanoid push recovery. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robotics*, pages 200–207. IEEE, 2006.

[Propoi1963] A. I. Propoi. Use of linear programming methods for synthesizing sampled-data automatic systems. In *Automation and Remote Control*, volume 24, pages 837–844, 1963.

[Quin1997] S. Joe Qin and Thomas A. Badgwell. An overview of industrial model predictive control technology. In *American Institute of Chemical Engineers Symposium Series*, volume 93, pages 232–256, 1997.

[Rantzer2009] Anders Rantzer. Dynamic dual decomposition for distributed control. In *American Control Conference*, pages 884–888. IEEE, 2009.

[Richalet1978] J. Richalet, A. Rault, J. L. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. In *Automatica*, volume 14, pages 413–428, 1978.

[Rubrecht2012] Sébastien Rubrecht, Vincent Padois, Philippe Bidaud, Michel De Broissia, and Max Da Silva Simoes. Motion safety and constraints compatibility for multibody robots. *Autonomous Robots*, 32(3):333–349, 2012.

[Saab2013] Layale Saab, Oscar E Ramos, François Keith, Nicolas Mansard, Philippe Soueres, and J Fourquet. Dynamic whole-body motion generation under rigid contacts and other unilateral constraints. *IEEE Transactions on Robotics*, 29(2):346–362, 2013.

[Sakagami2002] Yoshiaki Sakagami, Ryujin Watanabe, Chiaki Aoyama, Shinichi Matsunaga, Nobuo Higaki, and Kikuo Fujimura. The intelligent asimo: System overview and integration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems*, volume 3, pages 2478–2483. IEEE, 2002.

[Salini2010] Joseph Salini, Sébastien Barthélemy, and Philippe Bidaud. Lqp-based controller design for humanoid whole-body motion. In *Advances in Robot Kinematics: Motion in Man and Machine*, pages 177–184. Springer Netherlands, 2010.

[Salini2011] Joseph Salini, Vincent Padois, and Philippe Bidaud. Synthesis of complex humanoid whole-body behavior: a focus on sequencing and tasks transitions. In *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 1283–1290. IEEE, 2011.

[Samson1991] Claude Samson, Bernard Espiau, and Michel Le Borgne. *Robot Control: The Task Function Approach*. Oxford University Press, 1991.

[Sandini2007] G. Sandini, G. Metta, and D. Vernon. The icub cognitive humanoid robot: An open-system research platform for enactive cognition. In *50 Years of Artificial Intelligence*, Lecture Notes in Computer Science, chapter 32, pages 358–369. Springer, 2007.

[Sardain2004] Philippe Sardain and Guy Bessonnet. Forces acting on a biped robot. center of pressure-zero moment point. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 34(5):630–637, 2004.

[Scattolini2009] Riccardo Scattolini. Architectures for distributed and hierarchical model predictive control - a review. *Journal of Process Control*, 19(5):723–731, 2009.

[Schaffer03] Alin Albu-Schaffer, Christian Ott, Udo Frese, and Gerd Hirzinger. Cartesian impedance control of redundant robots: Recent results with the dlr-light-weight-arms. In *Proceedings of the IEEE International Conference on Robotics & Automation*, volume 3, pages 3704–3709, 2003.

[Sentis2006] Luis Sentis and Oussama Khatib. A whole-body control framework for humanoids operating in human environments. In *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 2641–2648. IEEE, 2006.

[Sentis2007] Luis Sentis. *Synthesis and control of whole-body behaviors in humanoid systems*. PhD thesis, Stanford university, 2007.

[Shen2007] W. Shen and J. Gu. Multi-criteria kinematics control for the pa10-7c robot arm with robust singularities. In *IEEE International Conference on Robotics and Biomimetics*, pages 1242–1248, 2007.

[Shim2003] David H Shim, H Jin Kim, and Shankar Sastry. Decentralized nonlinear model predictive control of multiple flying robots. In *Proceedings of the IEEE Conference on Decision and Control*, volume 4, pages 3621–3626. IEEE, 2003.

[Siciliano1991] Bruno Siciliano and J.J.E. Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. In *5th International Conference on Advanced Robotics*, pages 1211–1216, 1991.

[Siljak1996] D. D. Šiljak. Decentralized control and computations: status and prospects. In *Annual Reviews in Control*, volume 20, pages 131–141, 1996.

[Skogestad2007] Sigurd Skogestad and Ian Postlethwaite. *Multivariable feedback control: analysis and design*, volume 2. Wiley New York, 2007.

[Stasse2009] Olivier Stasse, Paul Evrard, Nicolas Perrin, Nicolas Mansard, and Abderrahmane Kheddar. Fast foot prints re-planning and motion generation during walking in physical human-humanoid interaction. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robotics*, pages 284–289, 2009.

[Stephens2007] Benjamin Stephens. Humanoid push recovery. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robotics*, pages 589–595. IEEE, 2007.

[Stephens2010] Benjamin J Stephens and Christopher G Atkeson. Push recovery by stepping for humanoid robots with force controlled joints. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robotics*, pages 52–59. IEEE, 2010.

[Tassa2012] Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems*, pages 4906–4913. IEEE, 2012.

[Van2006] WR Van Soest, QP Chu, and JA Mulder. Combined feedback linearization and constrained model predictive control for entry flight. *Journal of guidance, control, and dynamics*, 29(2):427–434, 2006.

[Venkat2006] Aswin Venkat. *Distributed model predictive control: theory and applications*. PhD thesis, University of Wisconsin, 2006.

[Venkat2008] Aswin N. Venkat, Ian A. Hiskens, James B. Rawlings, and Stephen J. Wright. Distributed mpc strategies with application to power system automatic generation control. In *IEEE Transactions on Control Systems Technology*, volume 16, pages 1192–1206, 2008.

[Vetter2002] Philipp Vetter, Tamar Flash, and Daniel M Wolpert. Planning movements in a simple redundant task. *Current Biology*, 12(6):488–491, 2002.

[Vukobratovic1972] Miomir Vukobratović and J. Stepanenko. On the stability of anthropomorphic systems. In *Mathematical Biosciences*, volume 15, pages 1–37, 1972.

[Wakasa2008] Yuji Wakasa, Mizue Arakawa, Kanya Tanaka, and Takuya Akashi. Decentralized model predictive control via dual decomposition. In *Proceedings of the IEEE Conference on Decision and Control*, pages 381–386. IEEE, 2008.

[Wang1973] S.H. Wang and E. Davison. On the stabilization of decentralized control systems. In *IEEE Transactions on Automatic Control*, volume 18, pages 473–478, 1973.

[Wieber2006] Pierre-Brice Wieber. Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robotics*, pages 137–142, 2006.

[Wing97] A.M. Wing, J. Randall Flanagan, and J. Richardson. Anticipatory postural adjustments in stance and grip. *Experimental Brain Research*, 116:122–130, 1997.

[Yamane2003] Katsu Yamane and Yoshihiko Nakamura. Dynamics filter-concept and implementation of online motion generator for human figures. *IEEE Transactions on Robotics and Automation*, 19(3):421–432, 2003.

[Zhang2007] Yan Zhang and Shaoyuan Li. Networked model predictive control based on neighbourhood optimization for serially connected large-scale processes. *Journal of process control*, 17(1):37–50, 2007.