Human-Humanoid Collaborative Carrying

Don Joven Agravante, Andrea Cherubini, Alexander Sherikov, Pierre-Brice Wieber, Abderrahmane Kheddar

Human-Humanoid Collaborative Carrying

International Journal of Robotics Research
XX(X):1–19

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Don Joven Agravante[1,4], Andrea Cherubini[1], Alexander Sherikov[2], Pierre-Brice Wieber[2], and Abderrahmane Kheddar[1,3]

Abstract

In this article we present a complete control framework which aims at making humanoid robots capable of carrying objects together with humans. Firstly, we review collaborative carrying by creating a human-inspired taxonomy. From this taxonomy, we identify the required primitive subtasks. Next, these are formulated into constrained optimization problems for controlling the whole-body motion of a humanoid robot. The subtasks include two walking pattern generators that account for physical collaboration, as well as posture and grasping controllers. Finally, we validate our framework in a variety of collaborative human-robot carrying experiments, using HRP-4.

Keywords

Physical Human-Robot Interaction, Human and humanoid skills/interaction.

1 Introduction

Robotics is moving towards human-centered systems, i.e., research focusing on the different facets of human-robot interaction. An important part of this is physical interaction, particularly collaboration between humans and robots. Several application areas can be envisioned where robots that can collaborate in tasks will be useful. For example, manufacturing and home assistance. In this regard, humanoid robots provide many advantages when working together with humans to perform various tasks. Humans learn to physically collaborate with one another from daily-life gathered experiences. Therefore, a humanoid with a similar range of motion and sensing has the potential to be an intuitive interface.

Carrying objects in collaboration with a human partner, in various postures and situations, is a problem that is rich, unexplored and has a high potential for practical application. Several situations can be envisioned, for instance, in large-scale manufacturing, construction, rescue, and disaster sites. Therefore, in this paper, we focus on this specific task and on its implementation on a humanoid robot.

Early work on enabling human-humanoid carrying was done by Yokoyama et al. (2003) in the Humanoid Robotics Project (HRP), where the HRP-2P humanoid cooperates with a human for a panel transportation task. This system was envisioned to have possible future applications in construction sites. Disregarding the legged aspect, even earlier work in this topic was done by Kosuge et al. (2000). They used *mobile manipulator* robots as in Khatib (1999), with wheels instead of legs for locomotion. The trade-off is having an easier control (from wheeled robotics technology) over more limited mobility (i.e. requiring drivable terrain). Although the locomotion mode is different, this pioneering work revealed one of the main issues early on: coordinating the motion of the mobile/floating base with the motion of the upper body manipulator and the human intention (which is often represented by the interaction force). More recent examples of mobile manipulators doing collaborative carrying are presented in Stückler and Behnke (2011); Lawitzky et al. (2010); Wang and Kosuge (2012). As discussed in Lafaye et al. (2014), the problem of balance of a wheeled manipulator robot can be very similar to

[1]CNRS-University of Montpellier, LIRMM UMR5506, Interactive Digital Human Group, Montpellier, France.
[2]INRIA Rhône-Alpes, BIPOP, 655 Avenue de l'Europe - CS 90051, 38334 Montbonnot Cedex, France.
[3]CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/RL, Tsukuba, Japan.
[4]now with INRIA Rennes - Bretagne Atlantique, Lagadic group, Campus Universitaire de Beaulieu, 35042 Rennes, France.

Corresponding author:
Don Joven Agravante, INRIA Rennes - Bretagne Atlantique, Lagadic group, Campus Universitaire de Beaulieu, 35042 Rennes, France.
Email: don-joven.agravante@inria.fr

the problem of balance of legged robots: even wheeled robots can tilt and fall over Kim et al. (2016). However, this aspect is noticeably missing in these earlier works, where the coupling between locomotion and collaborative manipulation was addressed only through kinematics and compliance coupling. Note that in Wang and Kosuge (2012), a linear inverted pendulum was used. Although this model is often used in standalone balance and walking control, this latter work extended its use by coupling together two inverted pendulums in order to achieve interactive dancing; although the robot used was balanced by design. Neglecting balance was possible because these robots had a very heavy and wide base, resulting in a low Center of Mass, and going slowly enough that they could very hardly loose balance. In contrast with these earlier works, we tackle here the coupling of legged control and balance with manipulation.

Going back to the task of collaborative carrying, it has also been tested on small scale humanoid platforms. For example, in Berger et al. (2013), the NAO humanoid was used in collaborative tasks. However, its main focus was on the use of internal sensors, in place of the hand force/torque sensors commonly used in physical human-robot interaction. NAO is also used in Bellaccini et al. (2014), where the capture point, explained in Koolen et al. (2012); Pratt et al. (2012), is used to guide walking. However, this work is more related to Berger et al. (2013) than the walking algorithm described here in the sense that the IMU data is processed in order to set a reference to NAO's walking algorithm. Another closely related demonstration can be found in McGill and Lee (2011), where Darwin robots are shown carrying a stretcher. However, the human element is removed. In these works where only robots are used, the interest is turned to multi-robot synchronization and communication. A similar work is presented in Wu et al. (2016) with HRP-2 robots in simulations. Both the multi-robot and human aspects have been considered by Hirata and Kosuge (2000). Another demonstration uses the example of lifting a table together with NAO by Sheng et al. (2015). The interest was in using learning algorithms to improve the interaction behavior. The topic of improving and understanding physical interaction of/with humans is a very broad and active research field with demonstrations ranging from industrial robot manipulators to humanoids. For example, studying human-human haptic interactions and applying it to human-robot teams was presented in Ikeura et al. (1994); Reed and Peshkin (2008). The topic of roles and role allocation is also very prevalent Mörtl et al. (2012). The aspect of mutual learning and adaptation has also been studied Ikemoto et al. (2012). Synthesizing controllers which take into account the uncertainty of human behavior prediction was shown in Medina et al. (2015). The recognition of haptic interaction patterns from a labeled dataset using supervised learning was presented in Madan et al. (2015).

Within our research group, human-humanoid collaboration is one of the main topics of interest. Various aspects of this task have been studied: learning, multi-sensory integration, human intention recognition, and design of proactive behaviors. More specifically, homotopy switching between leader and follower controllers was theorized in Evrard and Kheddar (2009a) and exemplified in Evrard and Kheddar (2009b). After this, Bussy et al. (2012b) studied human-human dyads, in order to understand how they cooperate in a table carrying task. The main interest was to extract physical interaction task-primitives and enhance impedance control, with varying levels of proactivity. In Bussy et al. (2012a), the idea of role switching and proactivity while carrying an object and turning is also explored. Contrary to these, this paper does not concentrate on high-level reasoning. Instead, we define all the pipeline that is required by a humanoid to realize collaborative carrying. Specifically, we embed a humanoid robot with a control framework that allows it to achieve a large variety of human-humanoid carrying tasks.

In our previous work Agravante et al. (2014), we explored the area of multi-sensor integration for the carrying task, by leveraging the framework of Bussy et al. (2012a,b), which was designed for collaborative table carrying. Although we successfully integrated vision and haptic information, in a setup where both were necessary, this study highlighted the limitations of the framework at hand. Firstly, it was difficult to extend it to any posture (and therefore objects), since the walking pattern generator (WPG) from Herdt et al. (2010) and the Stack-of-Tasks whole-body controller of Mansard et al. (2009) were abstracted in a layered architecture which approximated the CoM control using the waist frame on HRP-2. This means postures which moved the CoM away from the waist frame will fail (e.g. extending the arms or leaning with the chest). Secondly, the WPG of Herdt et al. (2010) worked well for stand-alone walking, but it was not designed for physical interaction. Although our previous demonstrations were shown to work well, this was because we limited the posture such that the hands are always situated approximately near the CoM. Simply situating the hands higher, while applying the same force, caused the WPG to fail. This was found while working on Agravante et al. (2013) which required this postural change. This postural guideline and its basis is explained in Agravante et al. (2016). With these in mind, we worked on reformulating the entire framework, to have a generic collaborative carrying algorithm for humanoid robots.

Our contributions and enhancements of the collaborative carrying framework are as follows.

- We present a novel *collaborative carrying taxonomy*, inspired by similar works on interaction such as Jarrassé et al. (2012), but with a specific focus on human-robot object transportation. For this, we take a human-centered approach, and observe how human teams collaboratively carry objects. The taxonomy aims at defining several cases, with few descriptors. Firstly, this allows us to focus on this limited set of use-cases, since methods can be generalized to all other cases. Secondly, we can define the primitive subtasks, that are common to all use cases, and that can all be formulated as optimization problems.
- Two pattern generators for *walking under sustained forces* are designed, one for a leader and the other one for a follower robot. Although these WPG have been originally presented in Agravante et al. (2016), further details on the modeling choices and its integration with a whole-body control are given here.
- Our whole-body framework, can simultaneously *account for both the carrying tasks and walking*. We show how collaborative carrying can be formulated as an optimization problem.
- *Generality*, proven by a series of case studies on a *real-size humanoid robot*, with a variety of robot roles (leader/follower), grasp types (hand/body) and carried objects (different shapes and sizes).

The rest of the paper is organized as follows. Section 2 presents the collaborative carrying taxonomy, along with the required primitive subtasks. The walking pattern generators, accounting for physical interaction are presented in Sect. 3. Next, Sect. 4 describes our optimization framework for whole-body control (i.e., to realize the primitive subtasks). After this, Sect. 5 presents the experimental validation of our methods, and Sect. 6 concludes the paper.

2 Decomposing the task of collaborative carrying

Collaborative carrying is a very complex task. To understand and decompose it, we take inspiration from how humans normally do it. This is done systematically by creating a taxonomy. After classifying how humans carry objects together, we generalize this task by designing a Finite State Machine (FSM), which accounts for all collaborative carrying subtasks. The end-goal is to have states which are instances of optimization problems.

2.1 A taxonomy of collaborative carrying

We first focus on various carrying schemes between pairs of human carriers. From this, we want to gain insight on how to program a humanoid, to effectively take the role of one of the human partners in such tasks. Figure 1 shows several real examples of human-human collaborative carrying (left side), with the corresponding simulations of what we envision with a humanoid robot collaborator (right side).

The figure shows the broadness and applicability of this skill, and serves as a useful visualization for breaking it down. Although the examples are very different, any collaborative carrying scenario simply boils down to two important components:

1. the object being carried, and
2. the agents (humans and/or humanoid robots) carrying the object.

From these descriptors, we can already form a taxonomy. However, it would not be useful because the object and agents are generally determined a priori. That is, we consider the problem of *having already a team of agents, whose goal is to move a specified object from one location to another*. We assume that neither the object nor the agent composition can be changed afterwards. For this problem, a components-based taxonomy is useless. Instead, we can consider the relationship between the components:

1. object-agent relation (grasp types),
2. agent-agent relation (relative pose).

These two taxonomies are detailed in the next subsections.

2.1.1 Object-agent relation: grasp types

The *object-agent relation* is simply defined by how the agents grasp the object –what we term as the *grasp type*. We can observe that the same object can be grasped in different ways, for example the cylinder in the simulations of Fig. 1 has three different instances of grasp type. There will also be cases where the same grasp type is used for different objects. In this work, we define two broad classes of grasp types: *hand grasps* and *body grasps*.

Hand grasps are those with contact points located uniquely on the hand/gripper. For example: the ones used to carry the table, stretcher and bucket in Fig. 1. These grasps are the subject of most of the existing robotics literature on grasping. This subclass was elaborated into its own taxonomy in Cutkosky (1989) which indicated the three main aspects to be considered when choosing a grasp:

1. task to do with the object after grasping,
2. object to be grasped,
3. gripper to be used.

The first two points are well-defined in our problem statement. The third one deserves further attention. In fact, we can consider another class of *grippers*, which

Figure 1. Collaborative carrying examples (left), with a human avatar and a humanoid robot mimicking the corresponding postures (right).

characterizes the second grasp type: *body grasps*. Indeed, we define body grasps as those that utilize grasp contact points on body parts not limited to the hand. For example, the arms, shoulder and torso can be used. In Fig. 1, this is the case when carrying the steel pipe or wooden logs. We observe that the object to be grasped requires the use of this kind of grasp. Generally, to grasp large objects, or objects without *natural handles*, the whole body can be used to form a larger *gripper*. The weight and shape of these objects can determine the design of the body grasp postures.

Finally, note that it is common for both agents to use the same grasp type, as shown in Fig. 1. Although there is no restriction for such, we can infer that if both agents have the same capabilities, and if the object is symmetric, using the same grasp type is the most logical outcome.

2.1.2 Agent-agent relation: relative pose

The *agent-agent relation* is not as straightforward as the object-agent relation. Several possible descriptors fit this category, all of which are useful to define a taxonomy:

- team composition (all humans, all robots, ratio in mixed groups, etc.),
- team/agent skill level in task execution (e.g., expert, novice),
- team coordination (leader/s, follower/s, equal collaborators),
- communication modalities (gesture or vision, speech or audition, haptics),
- agents' relative pose.

Firstly, team composition is not relevant in our problem statement (since it is predefined). The four other descriptors are all viable for creating a taxonomy. In this work, we prefer a descriptor that allows a quick and easy classification of unknown cases. For this, the *agents' relative pose* has been chosen among the four. As such, the relative pose between each pair of agents in the carrying team is used. The relative pose itself has two components that can be used independently: translation and orientation. These are quantitative descriptors, that can precisely characterize each collaborative carrying use-case. However, using these quantities directly is too specific, thus inappropriate for classifying the various scenarios. Instead, meaningful range boundaries are needed for classification. For this, we exploit another agent-agent relation descriptor: the communication modality. In particular, we relate haptic communication to relative translation, and vision to relative orientation.

The translation/distance between the agents can simply be classified as 'near' or 'far'. A meaningful range boundary can be defined by considering knowledge from *peripersonal space* in neuroscience, or simply if direct haptic communication (touch) is possible or not, so that we define:

- near: other agent is physically reachable,
- far: other agent is out of reach.

For the orientation, we can define meaningful boundaries by considering vision, more precisely the Field Of View (FOV). To describe the orientation range, let us first define the *nominal* FOV as the FOV when the agent is in a neutral resting position. The *extended* FOV is defined as the FOV of the agent as it looks around, by moving its body to some extent (i.e., without changing stance). With this, we can classify agents as facing:

- front: other agent is in the nominal FOV,
- side: other agent is not in the nominal FOV, but within the extended FOV,
- back: other agent is not in the extended FOV.

These descriptors should be applied to each agent relative to the other. Considering a team with only two agents, all nine permutations are shown in Table 1, where the occurence frequency in *real-world* cases is also given. Note that the table is symmetric with respect to the diagonal (e.g., front-side is equivalent to side-front). Thus, there are six classes in the taxonomy that considers only the agents' relative orientations.

Agent orientation	*Front*	*Side*	*Back*
Front	Common	Rare	Common
Side	Rare	Common	Rare
Back	Common	Rare	Extremely Rare

Table 1. Table of relative orientations for 2 carrying agents.

2.1.3 Applying the taxonomy

All six scenarios of Fig. 1 can be classified according to the proposed taxonomy. Each can be specified according to the three criteria: relative translation, relative orientation, and grasp type. The result is:

(a) table: far, front-front, hand grasp;

(b) pipe-shoulder: far, front-back, body grasp;

(c) stretcher: far, front-back, hand grasp;

(d) bucket: near, side-side, hand grasp;

(e) pipe-side: far, front-back, body grasp;

(f) pipe-front: far, side-side, body grasp.

This shows the ease of classifying given cases with the proposed taxonomy. But more importantly, we are concerned with the practical implications of using the taxonomy to program a humanoid robot. Directly, we note that relative pose and grasp type describe the goal of the humanoid when going towards the object and grasping it. However, the relative pose has a deeper meaning. As already mentioned in subsection 2.1.2, its classes are related to sensing, and to the availability of communication modalities (vision and haptic). Therefore, the required pose of the robot with respect to the human will largely affect what type of sensors are needed, and the best way to use them. For instance, consider a humanoid robot equipped with tactile sensing on the arms and/or shoulders. In a *near relative translation* scenario, these can be used for communication (e.g., tapping the arm as a signal to start/stop walking). On the contrary, *far relative translation* scenarios immediately disallow this possibility. For vision, a *front-facing* robot has the possibility to read the human gestures and body language, making it a better follower. Instead, a *back-facing* robot will be more aware of the surrounding environment, making it a better leader (e.g., this setup can be preferred for visual navigation).

2.2 Collaborative carrying as a Finite State Machine

The taxonomy highlighted the variations of collaborative carrying resulting in specific cases. Differently, this subsection will focus on the core similarities of the task. In order to program a robot, we decompose complex tasks into subtasks that will be easier to program. Formally, we use a Finite State Machine (FSM) to describe the whole task with subtasks as states. A useful decomposition is one where the states can be easily written as optimization problems for Sect. 4. A guideline for doing this is to first consider the state transitions. These must allow a smooth change from one optimization problem to the next. An easy way to do this is to identify brief periods where the motion is minimal. During these periods, the robot is said to be in a *quasi-static* state. More formally, the dynamic effects are small enough to be disregarded. Another important signifier of state transitions are the discrete changes in the robot contact state. Considering these, a first decomposition of the collaborative carrying task into an FSM is shown by Fig. 2.

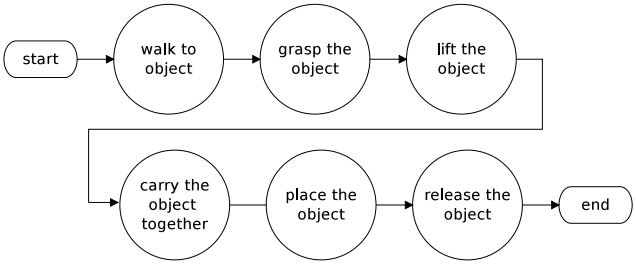

Figure 2. FSM for collaborative carrying.

Notice that there is a natural progression between states, where the robot is quasi-static during transitions. Furthermore, the conditions for transitioning between states are determined by thresholds on relevant sensed variables (for example, the distance between robot and object, or the

force applied to the grasped object). This decomposition also embeds the taxonomy components as targets:

- the agents' *relative pose* determines the target stance for the robot to reach when walking to the object,
- the *grasp type* determines the target posture (for body grasps) or the gripper pose (for hand grasps).

Although this FSM has simple subtasks, it is still not easy to translate some subtasks into optimization problems. Thus, we design a more detailed version of this FSM shown in Fig. 3, with the numbers indicating the transition order in normal carrying.

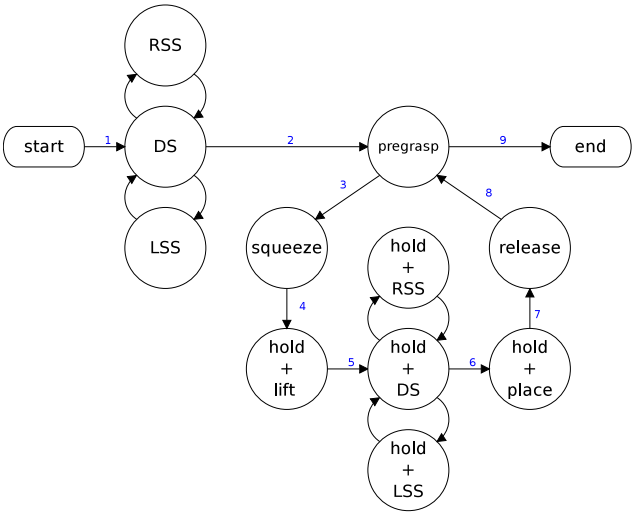

Figure 3. Detailed FSM for collaborative carrying, with each state/subtask corresponding to an optimization problem.

During walking, the contact transitions of the feet occur in a predictable pattern that can be used to define the walking states: left/right single support, and double support (indicated respectively as LSS/RSS and DS in Fig. 3). To decompose grasping, we need a *pregrasp* posture, i.e., a waypoint between grasping and the other states. The next state, *squeeze*, simply moves the robot so that the predefined contacts between the robot and object are made. Figure 4 shows the pregrasp and squeezing postures for two body grasps defined by the frames shown. Force or tactile sensors, when available, can help improve the execution of this state, and trigger the transition to the next state. The *hold* state simply consists in maintaining the contacts between the robot and the object. Note, however, that this state must be active during the carrying walk. Finally, the *release* state, is simply the inverse motion of *squeeze*. Without loss of generality, we consider subtasks *pregrasp*, *squeeze*, and *release* to be realized by the robot as it stands still, in double support.

Each state of the FSM shown in Fig. 3 will be formalized as a separate optimization problem. In the next subsection,

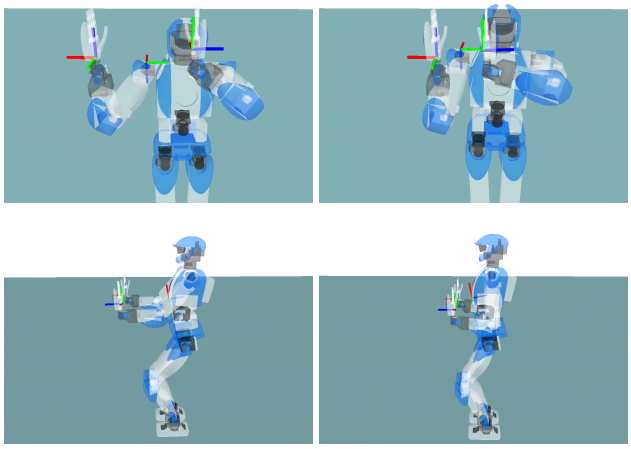

Figure 4. Two examples of "body grasps": pregrasp (left) and squeeze (right) postures.

we outline the general formulation that has been used for all the states. This formulation is instantiated for the walk, in Sect. 3, and for all other states, in Sect. 4. Moreover, note that the FSM also encompasses additional tasks to deal with practical implementation of the main ones, see examples in Vaillant et al. (2016).

2.3 States as optimization problems

All the aforementioned states/subtasks in collaborative carrying require the synthesis of robot low-level motor commands (usually in *joint-space*) from the high-level task goals (generally defined in some *operational space*). Furthermore, multiple goals with varying levels of importance are to be considered. The optimization-based approaches for whole-body control have shown to be effective in Mansard et al. (2009); Bouyarmane and Kheddar (2011); Escande et al. (2014a); Herzog et al. (2015); Vaillant et al. (2016). In contrast with these works, we concentrate on the implementation of collaborative carrying. More precisely, how each state of the previously designed FSM (Fig. 3) can be written as an optimization problem.

Generally, we seek to find the robot control input, represented by the optimization argument $\mathbf{x}$, such that:

$$\begin{aligned} &\underset{\mathbf{x}}{\text{argmin}} && \text{collection of objective functions,} \\ &\text{subject to} && \text{robot constraints,} \\ & && \text{environment constraints,} \\ & && \text{task constraints.} \end{aligned} \tag{1}$$

The formalism chosen here is largely based on Bouyarmane and Kheddar (2011) and on Vaillant et al. (2016). That is, we consider a *weighted quadratic programming*

formulation which allows to use the Euclidean or L^2 norm to define the desired task functions. Specifically, we design the total objective function as a weighted sum of objectives:

$$\sum_i w_i f_i(\mathbf{x}), \tag{2}$$

where each objective function $f_i(\mathbf{x})$ is written in the basic quadratic programming (QP) form:

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Q}_{qp}\mathbf{x} + \mathbf{c}_{qp}^\top \mathbf{x}. \tag{3}$$

Standard QP solvers can be used to obtain the optimal value of $\mathbf{x}$ for this problem. The reference text of Nocedal and Wright (2000) can be used for more information regarding this. Furthermore, it is well known that objectives expressed in linear form:

$$\mathbf{A}_{ls}\mathbf{x} = \mathbf{b}_{ls} \tag{4}$$

can also be incorporated in a QP, by using:

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{2}\left\|\mathbf{A}_{ls}\mathbf{x} - \mathbf{b}_{ls}\right\|^2 \\ &= \frac{1}{2}(\mathbf{A}_{ls}\mathbf{x} - \mathbf{b}_{ls})^\top(\mathbf{A}_{ls}\mathbf{x} - \mathbf{b}_{ls}). \end{aligned} \tag{5}$$

We also consider all constraints to be expressed as linear inequalities (or equalities, without loss of generality):

$$\mathbf{A}_c\mathbf{x} \leq \mathbf{b}_c. \tag{6}$$

Before instantiating this formulation for the various optimization problems, we must highlight a fundamental part of the FSM –walking together with the human, while holding the object. We consider this as a critical part, since other subtasks have regularly been demonstrated separately in robotics literature. That is: walking without additional constraints, contact-free motion of the upper body (servoing to a pregrasp posture), and grasping, are all regular topics in state-of-the-art robotics, each with tractable solutions for many common cases. The next section highlights the need for a redesigned walking pattern generator, that fits the subtask of walking with a human to transport an object. This walking pattern generator will also be expressed through the mentioned QP formulation. The output of this *high-level* QP will be fed to a *lower-level* QP which controls the whole robot body. The high-level QP for walking is detailed next in Sect. 3 and the low-level QP in Sect. 4.

3 Walking designed for physical collaboration

In the robotics literature, walking has historically been treated separately from manipulation. This has resulted in a good understanding of the underlying principles behind walking and locomotion in general. Although this is a good starting point, both manipulation and locomotion need to be consistent with each other, particularly when physical collaboration is necessary. Eventually, both need to be thought of as parts of the whole-body control problem (discussed in the next section). In this section, we start by revisiting the modeling of walking pattern generators (WPG), and eventually redesign these with physical collaboration in mind. The main concepts of this section were published in Agravante et al. (2016). Here, we recall the main points, namely: adding the external wrench into the model, and designing the objectives and constraints based on this wrench. Additionally, we better specify the usage of the new WPGs in the collaborative carrying task.

3.1 Modeling

Before anything else, a choice must be made on how to formulate the reduced dynamic model of the robot to consider physical interaction. Three possibilities for this are proposed in Audren et al. (2014). The differences can be thought of as moving the abstraction layer of the physical interaction. The proposed models are:

1. a model with full knowledge of the object (and/or human),
2. a model that considers the effects of the object (and/or human) on the robot's contact locations and linear forces, requiring additional grasp stability constraints,
3. a model that considers the effects of the object (and/or human) as external wrenches applied on the robot CoM.

In Audren et al. (2014), the first option was chosen and demonstrated in a simulation of a humanoid robot carrying a box in multi-contact. The model showed good results in simulation, and can be implemented on a real robot, by using the hand force sensors to estimate the object's dynamics beforehand. If this model is to be used in the context of collaborative carrying, it must consider the robot, human and object as a single system. The CoM of this system has contributions from the three subsystems. Also, there can be four (2 for the robot, 2 for the human) possible foot contacts, similar to a *quadruped*. This model can be used in simulation to control only the robot, while having perfect knowledge of each subsystem (human, robot, object). However, for real application, it requires important processing and sensing, and may not even be practically realizable. This makes it unusable for our purpose. The second option corresponds to a non-linear model. We have chosen to avoid this complexity, as motivated in Audren et al. (2014). This leaves us with the third option, which is chosen because of its simplicity in terms of implementation on a real robot, considering its

application to human-robot interaction, in general, and to collaborative carrying, in particular.

The development of this reduced model is inspired by Wieber et al. (2015), and detailed to some extent in Agravante et al. (2016). We separate the foot/ground contact forces from other interaction contact forces that are denoted by $\mathbf{h}_{\text{ext}} = [\mathbf{f}_{\text{ext}}^\top \ \mathbf{n}_{\text{ext}}^\top]^\top \in \mathbb{R}^6$. This represents the *external wrench* (from the carried object weight and from the human collaborator), and is expressed in a fixed orientation frame placed on the Center of Mass, $\mathbf{c}$ (CoM). As is common in the literature Kajita et al. (2003); Herdt et al. (2010), we aim at keeping the center of pressure $\mathbf{z}$ (also known as Zero Moment Point, ZMP) within the support polygon (i.e., the convex hull of the feet contact points). We assume that the robot is walking on a flat horizontal ground, at a constant height (c^z), and that the angular momentum is constant. Then, the Newton Euler equations yield the following relationship between CoM and ZMP:

$$\mathbf{z}^{x,y} = \mathbf{c}^{x,y} - \left(\frac{c^z}{g^z - \frac{f^z_{\text{ext}}}{m}}\right)\ddot{\mathbf{c}}^{x,y} + \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}\left(\frac{\mathbf{n}^{x,y}_{\text{ext}}}{mg^z - f^z_{\text{ext}}}\right) + \left(\frac{c^z \mathbf{f}^{x,y}_{\text{ext}}}{mg^z - f^z_{\text{ext}}}\right), \tag{7}$$

with m the robot mass and $\mathbf{g}$ the gravity vector. In the absence of an external wrench, this becomes:

$$\mathbf{z}^{x,y} = \mathbf{c}^{x,y} - \left(\frac{c^z}{g^z}\right)\ddot{\mathbf{c}}^{x,y}, \tag{8}$$

which is the standard expression found in the literature. Guidelines on how to reduce the effects of an external wrench can be inferred from (7): a heavier robot (more massive), lower CoM height, external force aligned with the CoM to reduce the corresponding torque.

3.2 Walking optimization formulation

As detailed in Agravante et al. (2016), we use a triple-integrator system for the CoM, assuming that its trajectory is differentiable three times. The model input is then the CoM jerk. The model output is defined using the ZMP formulation in (7). Next, we apply model predictive control methodology on this system such that we now have future states, controls, and outputs represented. Furthermore, we express the ZMP in a *local foot* coordinate to expose the foot landing positions such that we can define:

$$\mathbf{x} = \begin{bmatrix} \tilde{\mathbf{u}} \\ \breve{\mathbf{r}} \end{bmatrix}, \tag{9}$$

where $\tilde{\mathbf{u}}$ and $\breve{\mathbf{r}}$ are the concatenation, over the preview horizon, respectively of the control inputs $\dddot{\mathbf{c}}$, and of future foot landing positions, expressed in a *local frame* placed at the preceding foot pose. Vector $\mathbf{x}$ will be the argument of the optimization problem (1). The future states and outputs can be written as:

$$\begin{aligned} \tilde{\mathbf{c}} &= \mathbf{S}\mathbf{x} + \mathbf{s}, \\ \breve{\mathbf{z}} &= \mathbf{S}_z\mathbf{x} + \mathbf{s}_z. \end{aligned} \tag{10}$$

Using (10), we can now design various WPGs, that are suited to different types of physical interaction. In particular, we revisit the *leader* and *follower* modalities, designed in Bussy et al. (2012a). We then show how each one drastically changes the way the external wrench is used in the WPG. Before that, we list some objectives and constraints that are common to the formulations:

- The control input (i.e., the CoM jerk) is minimized to smoothen the CoM trajectory. This is done via the objective function:

$$\|\tilde{\mathbf{u}}\|^2 = \left\|\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{x}\right\|^2. \tag{11}$$

- The distance between the ZMP and the foot center must be minimized to increase the stability margin (since unknown disturbances could push the ZMP away from the target). This objective function is:

$$\|\breve{\mathbf{z}}\|^2 = \|\mathbf{S}_z\mathbf{x} + \mathbf{s}_z\|^2. \tag{12}$$

- The ZMP should be maintained within the support polygon (with security margins) using the constraint:

$$\underline{\breve{\mathbf{z}}} \leq \breve{\mathbf{z}} \leq \overline{\breve{\mathbf{z}}}. \tag{13}$$

  Using (10) to expose the argument, it becomes:

$$\underline{\breve{\mathbf{z}}} - \mathbf{s}_z \leq \mathbf{S}_z\mathbf{x} \leq \overline{\breve{\mathbf{z}}} - \mathbf{s}_z. \tag{14}$$

- The feet positions should be constrained:

$$\underline{\breve{\mathbf{r}}} \leq \breve{\mathbf{r}} \leq \overline{\breve{\mathbf{r}}}. \tag{15}$$

  Since $\breve{\mathbf{r}}$ is part of the argument, this can be written as:

$$\underline{\breve{\mathbf{r}}} \leq \begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}\mathbf{x} \leq \overline{\breve{\mathbf{r}}}. \tag{16}$$

Note that objectives (11), (12) are expressed as (5), and constraints (14), (16) as (6). Notations $\underline{x}$ and $\overline{x}$ means respectively lower and upper bounds of whatever x means.

Apart from the mentioned constraints and objectives that are common, the leader and follower WPG are each characterized by a specific formulation of the objective function that controls the CoM. These are given in the remainder of this subsection.

3.2.1 *Walking pattern generator for a follower robot*

A follower robot acts based on the leader's intention. In a collaborative carrying situation, this is represented by the external wrench applied by the carrying partner. Usually, moving the object together implies only a planar motion. This intention is then only $\mathbf{f}_{\text{ext}}^{\text{x,y}}$. Hence, a follower WPG must generate motions that are a function of this. Previous works Agravante et al. (2014); Bussy et al. (2012a) have used a damping control by providing a reference CoM velocity to the WPG which is proportional to the external force. We extend this to perform more complex behaviors, by defining the full impedance model, with diagonal matrices $\mathbf{M}$, $\mathbf{B}$ and $\mathbf{K}$ containing the virtual mass, damping and stiffness parameters:

$$\mathbf{f}_{\text{ext}}^{\text{x,y}} = \mathbf{M}\ddot{\mathbf{c}}^{x,y} + \mathbf{B}\dot{\mathbf{c}}^{x,y} + \mathbf{K}\mathbf{c}^{x,y}. \tag{17}$$

Recalling that state $\tilde{\mathbf{c}}$ contains CoM accelerations, velocities and positions, one can simply define an appropriate impedance parameter matrix $\mathbf{G}_{\text{mbk}}$. An appropriate selection matrix $\mathbf{S}_{\text{f}}$ is also needed to select only the $f_{\text{ext}}^{\text{x}}$, and $f_{\text{ext}}^{\text{y}}$ components, so that the controller will aim at minimizing:

$$\left\|\mathbf{G}_{\text{mbk}}\tilde{\mathbf{c}} - \mathbf{S}_{\text{f}}\tilde{\mathbf{f}}\right\|^2 = \left\|\mathbf{G}_{\text{mbk}}\mathbf{S}\mathbf{x} + \mathbf{G}_{\text{mbk}}\mathbf{s} - \mathbf{S}_{\text{f}}\tilde{\mathbf{f}}\right\|^2. \tag{18}$$

Note that by injecting (10), we could write this objective in the form (5). The optimization problem can be rewritten, including objectives (11), (12), and constraints (13), (15):

$$\begin{aligned} \underset{\mathbf{x}}{\text{argmin}} \quad & w_1 \left\|\mathbf{G}_{\text{mbk}}\tilde{\mathbf{c}} - \mathbf{S}_{\text{f}}\tilde{\mathbf{f}}\right\|^2 + w_2 \left\|\tilde{\mathbf{u}}\right\|^2 + w_3 \left\|\breve{\mathbf{z}}\right\|^2 \\ \text{subject to} \quad & \underline{\breve{\mathbf{z}}} \leq \breve{\mathbf{z}} \leq \overline{\breve{\mathbf{z}}} \\ & \underline{\breve{\mathbf{r}}} \leq \breve{\mathbf{r}} \leq \overline{\breve{\mathbf{r}}} \end{aligned} \tag{19}$$

Notice that the future wrench values are required in $\tilde{\mathbf{f}}$. This is an important aspect in MPC: a model is needed to do a prediction, on which to base the control decision. In this case, the prediction model for $\tilde{\mathbf{f}}$ coincides with the concept of *proactive behaviors*. Having a good model of human intention can be difficult (if at all possible). But in such MPC schemes, the prediction is reevaluated at each sampling time, so it does not need to be very precise in the far future. In this case, we use a simplistic model for predicting the future interaction force, a constant value such that $\mathbf{f}_{\text{N}} = \ldots = \mathbf{f}_1 = \mathbf{f}_0$, with $\mathbf{f}_0$ the current wrench measure (obtained from a force/torque sensor). Of course, any sophistication of the force prediction, e.g. using machine learning from experience or advanced estimation techniques, is also possible. Yet, this prediction model is continuously updated at each sampling period with the current reading of the force sensors. As always in control, the most important element here is that there is feedback of the current state of the system. This whole approach is validated experimentally in Sect. 5. Also note that $\mathbf{s}_{\text{z}}$ requires $\mathbf{f}_{\text{ext}}^{\text{z}}$. For the collaborative carrying task, this is the contribution to carrying the object weight. This comes from (7) where we can infer that a more negative $\mathbf{f}_{\text{ext}}^{\text{z}}$ (i.e., carrying a heavier object) actually helps to reject the external wrench effects. A simulation showing this appears in Sect. 5.

3.2.2 *Walking pattern generator for a leader robot*

For leading, a clear intention is necessary. We propose two ways for formulating the leader intention in the WPG. The first is to have the robot track a reference trajectory, known beforehand. For the collaborative carrying task, this could be generated by knowing where the object is and where it will be transported to. A classical way to track a trajectory in operational space Chung et al. (2008) is:

$$\ddot{\mathbf{c}}^{x,y} = \ddot{\mathbf{c}}_{\text{ref}}^{x,y} + \mathbf{B}(\dot{\mathbf{c}}_{\text{ref}}^{x,y} - \dot{\mathbf{c}}^{x,y}) + \mathbf{K}(\mathbf{c}_{\text{ref}}^{x,y} - \mathbf{c}^{x,y}), \tag{20}$$

where $\mathbf{B}$ and $\mathbf{K}$ are diagonal gain matrices with positive elements. This can be reformulated as an objective function, with an appropriate gain matrix, similar to that of the follower:

$$\left\|\mathbf{G}_{\text{bk}}(\tilde{\mathbf{c}}_{\text{ref}} - \tilde{\mathbf{c}})\right\|^2 = \left\|-\mathbf{G}_{\text{bk}}\mathbf{S}\mathbf{x} - \mathbf{G}_{\text{bk}}\mathbf{s} + \mathbf{G}_{\text{bk}}\tilde{\mathbf{c}}_{\text{ref}}\right\|^2. \tag{21}$$

Again, with (10), we could write this objective in the form (5). A second option consists in including the external wrench in the optimization argument, expanding it as: $\mathbf{x} = [\tilde{\mathbf{u}}^\top \ \breve{\mathbf{r}}^\top \ \tilde{\mathbf{f}}^\top]^\top$. Although the form of (10) is kept, the matrices $\mathbf{S}$ and $\mathbf{S}_{\text{z}}$ and vector $\mathbf{s}_{\text{z}}$ are different as explicitly shown in Agravante et al. (2016). Hence, the derivation of objectives and constraints is analogous. The idea of this formulation is that placing a part of the external wrench in the argument might allow the robot to use the interaction to balance itself. For safety, the applied wrench may need to be constrained and/or minimized.

Using both options, along with objectives (11), (12), and constraints (13), (15), the optimization problem becomes:

$$\begin{aligned} \underset{\mathbf{x}}{\text{argmin}} \quad & w_1 \left\|\mathbf{G}_{\text{bk}}(\tilde{\mathbf{c}}_{\text{ref}} - \tilde{\mathbf{c}})\right\|^2 + w_2 \left\|\tilde{\mathbf{u}}\right\|^2 \\ & + w_3 \left\|\breve{\mathbf{z}}\right\|^2 + w_4 \left\|\tilde{\mathbf{f}}\right\|^2 \\ \text{subject to} \quad & \underline{\breve{\mathbf{z}}} \leq \breve{\mathbf{z}} \leq \overline{\breve{\mathbf{z}}} \\ & \underline{\breve{\mathbf{r}}} \leq \breve{\mathbf{r}} \leq \overline{\breve{\mathbf{r}}} \\ & \underline{\tilde{\mathbf{f}}} \leq \tilde{\mathbf{f}} \leq \overline{\tilde{\mathbf{f}}}. \end{aligned} \tag{22}$$

Since $\tilde{\mathbf{f}}$ is part of the argument, it is trivial to see that all objective functions and all constraints are of the forms (5) and (6), respectively. Note that in the absence of an external wrench and using only velocities in the tracking task results in the *standard WPG* from Agravante et al. (2016).

3.3 Swing foot trajectories

The value of $\check{\mathbf{r}}$ resulting from either of the two WPG gives only the future foot landing positions, i.e., the footprints, in the preceding foot frame. However, the trajectory between footprints must also be generated and provided to the whole-body controller. Since there is no flight phase in walking (both feet never leave the ground together), the foot that is off the ground is referred to as the *swing foot* without ambiguity. In this subsection, we explain how its trajectory is designed, using cubic polynomials. We denote it by $\mathbf{r}_{\text{swdes}}(t)$, expressed in the preceding foot frame. Without loss of generality, the time is reset at each step, and starts when the foot leaves the ground.

The total duration of the swing foot trajectory t_{swf} is predetermined, and the trajectory is designed to start and end in a resting state: $\dot{\mathbf{r}}_{\text{swdes}}(0) = \dot{\mathbf{r}}_{\text{swdes}}(t_{\text{swf}}) = \mathbf{0}$. Furthermore, the initial and final values are known: for x and y they correspond to $r^{\text{x,y}}_{\text{sw0}}$, the current foot location and $r^{\text{x,y}}_{\text{swf}}$ which is generated by the WPG within $\check{\mathbf{r}}$. z is null for both. The z component can be parametrized by step height $r_{\text{sth}} > 0$. This will be attained half way through: $r_{\text{swdes}}\left(t_{\text{swf}}/2\right) = r_{\text{sth}}$.

Given these boundary conditions, the x and y components of the swing foot trajectory are given by:

$$r^{x,y}_{\text{swdes}}(t) = r^{x,y}_{\text{sw0}} + \frac{3}{t^2_{\text{swf}}}\left(r^{x,y}_{\text{swf}} - r^{x,y}_{\text{sw0}}\right)t^2 - \frac{2}{t^3_{\text{swf}}}\left(r^{x,y}_{\text{swf}} - r^{x,y}_{\text{sw0}}\right)t^3, \tag{23}$$

For the z component, we need a concatenation of two cubic polynomials:

$$r^{z}_{\text{swdes}}(t) = \begin{cases} \frac{12 r_{\text{sth}}}{t^2_{\text{swf}}}t^2 - \frac{16 r_{\text{sth}}}{t^3_{\text{swf}}}t^3 & t \le \frac{t_{\text{swf}}}{2}, \\ r_{\text{sth}} - \frac{12 r_{\text{sth}}}{t^2_{\text{swf}}}\left(t - \frac{t_{\text{swf}}}{2}\right)^2 + \frac{16 r_{\text{sth}}}{t^3_{\text{swf}}}\left(t - \frac{t_{\text{swf}}}{2}\right)^3 & t > \frac{t_{\text{swf}}}{2} \end{cases} \tag{24}$$

Figure 5 shows an example of the swing foot moving along the generated trajectory.

Figure 5. Parameterized swing foot trajectory.

4 Whole-body control for collaborative carrying

The last two sections provided important building blocks for the collaborative carrying task. This section aims at wrapping everything together into coordinated whole-body motions. For instance, to generate the described walks, the WPG results: $\tilde{\mathbf{c}}$, $\check{\mathbf{r}}$, and $\mathbf{r}_{\text{sw}}$ (respectively: CoM trajectory, footprints, and swing foot trajectory) must be mapped to robot joint commands, $\mathbf{q}$. To explain how this is done, we start by recalling the base components of the optimization-based whole-body control framework developed in our research group Vaillant et al. (2016). Next, some recurrent objectives and constraints are presented. Finally, we explain how all these components are assembled, to formulate all the collaborative carrying subtasks in the FSM of Fig. 3.

4.1 Whole-body control as an optimization problem

To start detailing our whole-body control framework, we define the optimization argument in (1) as:

$$\mathbf{x} = \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix}. \tag{25}$$

Here, $\mathbf{q}$ defines the robot configuration, i.e. the joint positions along with the floating-base representation Featherstone (2008), and $\boldsymbol{\lambda}$ is the vector of linearized friction cone base weights. This is defined such that all contact forces stacked in a column vector correspond to:

$$\mathbf{f}_{\text{con}} = \mathbf{K}_{\text{fc}}\boldsymbol{\lambda}, \tag{26}$$

with $\mathbf{K}_{\text{fc}} \in \mathbb{R}^{3\text{n}\times\text{nm}}$ a matrix of generators for linearizing the friction cone (n is the number of contact points, and m the number of generators chosen for the linearization). Note that we use a distributed contact model without torque. For example, the foot contacting with a flat floor is represented by four contact points: one at each corner, each with its own contact force.

Next, we can rewrite (1) more specifically as:

$$\begin{aligned} \underset{\mathbf{x}}{\text{argmin}} \quad & \sum_i w_{\text{i}} f_{\text{i}}(\mathbf{x}) + w_{\text{pos}} f_{\text{pos}}(\mathbf{x}, \mathbf{q}_{\text{des}}) + w_{\lambda} f_{\boldsymbol{\lambda}}(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{A}_{\text{base}}\mathbf{x} \le \mathbf{b}_{\text{base}} \\ & \mathbf{A}_{\text{spec}}\mathbf{x} \le \mathbf{b}_{\text{spec}} \end{aligned} \tag{27}$$

We will refer to this as the *base* formulation, since it only specifies the most essential parts of the optimization problem which are always used. The first term in the objective functions represents the collection of objectives f_{i} specific to the subtask, that will be described later on, in Sect. 4.3. The two other objective functions are detailed in Sect. 4.1.1 while the *base* constraints are

detailed in Sect. 4.1.2. The last constraint symbolizes *specific constraints*, which are discussed for each subtask in Sect. 4.3.

4.1.1 Base objective functions

The first base objective function is termed the *posture task*, and represented as f_{pos}. This corresponds to joint positioning at a given posture $\mathbf{q}_{\text{des}}$, with null $\dot{\mathbf{q}}_{\text{des}}$ and $\ddot{\mathbf{q}}_{\text{des}}$, and can be written as:

$$f_{\text{pos}}(\mathbf{x}, \mathbf{q}_{\text{des}}) = \frac{1}{2} \left\| \mathbf{K}(\mathbf{q}_{\text{des}} - \mathbf{q}) - \mathbf{B}\dot{\mathbf{q}} - \ddot{\mathbf{q}} \right\|^2, \tag{28}$$

with $\mathbf{K}$ and $\mathbf{B}$ square diagonal gain matrices with positive values. This objective is of the form (5). The goal of the posture task is to have a *default* configuration of each joint. This implies that its corresponding weight normally has a relatively low value, to avoid conflicting with more important tasks.

The second base objective consists in minimizing $\|\boldsymbol{\lambda}\|^2$:

$$f_{\boldsymbol{\lambda}}(\mathbf{x}) = \|\boldsymbol{\lambda}\|^2 = \left\| \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{x} \right\|^2. \tag{29}$$

This objective function, in conjunction with the posture task, ensures that the complete $\mathbf{Q}_{\text{qp}}$ matrix is positive definite, allowing an easier solution to the QP.

4.1.2 Base constraints

There are four constraints in the base formulation of our optimization problem (27), namely:

$$\begin{aligned} &\boldsymbol{\lambda} \geq \mathbf{0} \\ &\underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau} \leq \overline{\boldsymbol{\tau}} \\ &\underline{\mathbf{q}} \leq \mathbf{q} \leq \overline{\mathbf{q}} \\ &\underline{\dot{\mathbf{q}}} \leq \dot{\mathbf{q}} \leq \overline{\dot{\mathbf{q}}}. \end{aligned} \tag{30}$$

These are active in all states of the collaborative carrying FSM and we refer to them with the inequality:

$$\mathbf{A}_{\text{base}}\mathbf{x} \leq \mathbf{b}_{\text{base}}, \tag{31}$$

where the various constraints, with their lower and upper limits, are reformulated and stacked appropriately.

The first one ensures that the contact forces are inside the friction cone (no slipping). It can be formulated as:

$$\begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{x} \geq \mathbf{0}. \tag{32}$$

The second constraint in (30) places bounds on the actuator torques $\boldsymbol{\tau}$. These can be obtained from the robot dynamic equation:

$$\boldsymbol{\tau} = \mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \boldsymbol{\tau}_{\mathbf{g}} - \mathbf{J}_{\text{con}}^{\top}\mathbf{f}_{\text{con}}, \tag{33}$$

with $\mathbf{H}$ and $\mathbf{C}$ respectively the inertia and Coriolis/centrifugal terms, $\boldsymbol{\tau}_{\mathbf{g}}$ the torques due to gravity, $\mathbf{J}_{\text{con}}$ the stacked contact point Jacobian matrices, and $\mathbf{f}_{\text{con}}$ the stacked vector of contact forces from (26). The constraint can then be rewritten:

$$\underline{\boldsymbol{\tau}} - \mathbf{C}\dot{\mathbf{q}} - \boldsymbol{\tau}_{\mathbf{g}} \leq \begin{bmatrix} \mathbf{H} & -\mathbf{J}_{\text{con}}^{\top}\mathbf{K}_{\text{fc}} \end{bmatrix} \mathbf{x} \leq \overline{\boldsymbol{\tau}} - \mathbf{C}\dot{\mathbf{q}} - \boldsymbol{\tau}_{\mathbf{g}}. \tag{34}$$

The third and fourth constraints in (30) bound joint positions and velocities. The joint accelerations can be exposed by using numerical integration at each time interval k:

$$\begin{aligned} \dot{\mathbf{q}}_{k+1} &= \dot{\mathbf{q}}_k + \ddot{\mathbf{q}}_k \Delta t, \\ \mathbf{q}_{k+1} &= \mathbf{q}_k + \dot{\mathbf{q}}_k \Delta t + \frac{1}{2}\ddot{\mathbf{q}}_k \Delta t^2. \end{aligned} \tag{35}$$

With (35), these constraints become:

$$\begin{aligned} \underline{\dot{\mathbf{q}}} - \dot{\mathbf{q}} &\leq \begin{bmatrix} \mathbf{I}\Delta t & \mathbf{0} \end{bmatrix} \mathbf{x} \leq \overline{\dot{\mathbf{q}}} - \dot{\mathbf{q}}, \\ \underline{\mathbf{q}} - \mathbf{q} - \dot{\mathbf{q}}\Delta t &\leq \frac{1}{2} \begin{bmatrix} \mathbf{I}\Delta t^2 & \mathbf{0} \end{bmatrix} \mathbf{x} \leq \overline{\mathbf{q}} - \mathbf{q} - \dot{\mathbf{q}}\Delta t. \end{aligned} \tag{36}$$

Notice that all the robot limitations must be known. Stacking (32), (34) and (36), yields the explicit expressions of $\mathbf{A}_{\text{base}}$, and $\mathbf{b}_{\text{base}}$, which are omitted here.

4.2 Reusable objectives and constraints

Several objectives and constraints are recurrent in the FSM, and can be written in a re-usable form. For this, let us first define a task vector in the operational space $\mathbf{e}$ (e.g., the pose of any frame on the robot body or on the carried object), and the function mapping it to robot joint space:

$$\mathbf{e} = f_{\mathbf{e}}(\mathbf{q}). \tag{37}$$

Assuming $f_{\mathbf{e}}$ is twice differentiable:

$$\dot{\mathbf{e}} = \mathbf{J}_{\mathbf{e}}\dot{\mathbf{q}}, \tag{38}$$

$$\ddot{\mathbf{e}} = \mathbf{J}_{\mathbf{e}}\ddot{\mathbf{q}} + \dot{\mathbf{J}}_{\mathbf{e}}\dot{\mathbf{q}}, \tag{39}$$

with $\mathbf{J}_{\mathbf{e}}$ the task Jacobian. To servo $\mathbf{e}$ with the *tracking task objective* in Chung et al. (2008):

$$\begin{aligned} &f_{\text{tr}}(\mathbf{x}, \mathbf{e}_{\text{des}}(t)) = \\ &\frac{1}{2} \left\| \mathbf{K}(\mathbf{e}_{\text{des}} - \mathbf{e}) + \mathbf{B}(\dot{\mathbf{e}}_{\text{des}} - \dot{\mathbf{e}}) + (\ddot{\mathbf{e}}_{\text{des}} - \ddot{\mathbf{e}}) \right\|^2, \end{aligned} \tag{40}$$

where $\mathbf{e}_{des}(t)$ denotes the desired task trajectory (i.e., it includes $\mathbf{e}_{\text{des}}, \dot{\mathbf{e}}_{\text{des}}$ and $\ddot{\mathbf{e}}_{\text{des}}$), and $\mathbf{K}$ and $\mathbf{B}$ are square diagonal gain matrices with positive values. These can be tuned by considering the task dynamics equivalent to those of a mass-spring-damper system with unit mass. Typically, to obtain a critically damped system, only $\mathbf{K}$ needs to be tuned, with $\mathbf{B} = 2\sqrt{\mathbf{K}}$. Using (38) and (39), (40) can be written as (5). Formulation (40) is so generic, that several tasks can use it, e.g.: any physical point on the robot, the center of mass, joint positions.

A particular case of the tracking task is the *set-point objective*, where only the reference value is considered, while the reference velocity and acceleration are approximated as null:

$$f_{\text{sp}}(\mathbf{x}, \mathbf{e}_{\text{des}}) = \frac{1}{2} \left\| \mathbf{K}(\mathbf{e}_{\text{des}} - \mathbf{e}) - \mathbf{B}\dot{\mathbf{e}} - \ddot{\mathbf{e}} \right\|^2 . \tag{41}$$

This formulation was used in Bouyarmane and Kheddar (2011). An important example of the set-point task is the aforementioned posture task (28). This is obtained by setting $\mathbf{e} = \mathbf{q}$.

Apart from servoing a body part, another common goal is to keep a certain body part motionless. A common example is to keep the feet in contact with the ground. To this end, we define a *contact constraint* as in Bouyarmane and Kheddar (2011), by nullifying the acceleration of a robot point that is in contact with the environment:

$$\ddot{\mathbf{e}} = \mathbf{0}. \tag{42}$$

Using (39), it can be written as the equality constraint:

$$\begin{bmatrix} \mathbf{J_e} & \mathbf{0} \end{bmatrix} \mathbf{x} = -\dot{\mathbf{J}}_{\mathbf{e}} \dot{\mathbf{q}}. \tag{43}$$

Furthermore, some DOF can be released by adding a selection matrix, as in Vaillant et al. (2016). The interested reader may also refer to the same reference for some improvements on handling the numerical stability of this constraint.

Lastly, it is possible to add collision avoidance as in Vaillant et al. (2016), by computing the distances using the approach explained in Escande et al. (2014b). This is generally used to avoid self-collisions between robot body parts, and is not essential to the discussion of this paper. The interested reader is referred to these two works.

4.3 Objectives and constraints of each subtask

Since the *base* objectives and constraints, that are common to all states of the collaborative carrying FSM, have been explained above, we only detail those specific to each subtask. An important aspect concerns the control of the CoM. In walking states (DS, RSS and LSS), this is servoed using a *tracking task objective* to follow the CoM trajectory output by the WPG (leader or follower). For all the other subtasks, we use a *set-point objective* to pull the CoM towards the middle of the two feet. This objective always has a high weight such that it approximates a quasi-static balancing constraint.

4.3.1 Double support

During the Double Support (DS) states, both feet: $\mathbf{r}_{\text{left}}, \mathbf{r}_{\text{right}}$, must maintain contact with the ground, via *contact constraints*. The CoM is servoed with a trajectory $\mathbf{c}_{\text{des}}(t)$, obtained from the standard WPG of Sect. 3. A set-point torso orientation (characterized by the desired value of the torso frame's orientation, $\boldsymbol{\theta}_{\text{tsdes}}$) is also added to have an upright posture while walking. In summary, the whole-body optimization problem is:

$$\begin{aligned} \underset{\mathbf{x}}{\text{argmin}} \quad & w_{\text{c}} f_{\text{tr}}(\mathbf{x}, \mathbf{c}_{\text{des}}(t)) + w_{\text{ts}} f_{\text{sp}}(\mathbf{x}, \boldsymbol{\theta}_{\text{tsdes}}) + \\ & w_{\text{pos}} f_{\text{pos}}(\mathbf{x}, \mathbf{q}_{\text{des}}) + w_{\boldsymbol{\lambda}} f_{\boldsymbol{\lambda}}(\mathbf{x}), \\ \text{subject to} \quad & \ddot{\mathbf{r}}_{\text{left}} = \mathbf{0}, \\ & \ddot{\mathbf{r}}_{\text{right}} = \mathbf{0}, \\ & \mathbf{A}_{\text{base}} \mathbf{x} \leq \mathbf{b}_{\text{base}}. \end{aligned} \tag{44}$$

4.3.2 Right/left single supports

Single support states (RSS or LSS) occur during walking in between two consecutive double support states. As such, they retain the CoM trajectory tracking task from the standard WPG, as well as the set-point torso orientation task. Differently from DS, only one foot (support) is constrained while the other (swing) is servoed in the air. The generation of the swing foot trajectory: $\mathbf{r}_{\text{swdes}}(t)$ was explained in Sect. 3.3, and is followed using a tracking task objective. Hence, the optimization problem is:

$$\begin{aligned} \underset{\mathbf{x}}{\text{argmin}} \quad & w_{\text{c}} f_{\text{tr}}(\mathbf{x}, \mathbf{c}_{\text{des}}(t)) + w_{\text{sw}} f_{\text{tr}}(\mathbf{x}, \mathbf{r}_{\text{swdes}}(t)) + \\ & w_{\text{ts}} f_{\text{sp}}(\mathbf{x}, \boldsymbol{\theta}_{\text{tsdes}}) + w_{\text{pos}} f_{\text{pos}}(\mathbf{x}, \mathbf{q}_{\text{des}}) + \\ & w_{\boldsymbol{\lambda}} f_{\boldsymbol{\lambda}}(\mathbf{x}), \\ \text{subject to} \quad & \ddot{\mathbf{r}}_{\text{sup}} = \mathbf{0}, \\ & \mathbf{A}_{\text{base}} \mathbf{x} \leq \mathbf{b}_{\text{base}}. \end{aligned} \tag{45}$$

4.3.3 Pregrasping and releasing

The *pregrasp* and *release* states have the same formulation, the only difference being their preceding state. Thus we present only the pregrasp. The *pregrasp* state is a waypoint state that eases the grasping by targeting preplanned grasp point locations. The synthesis of these locations can be formalized either as a stance generation problem Brossette et al. (2015), or by considering caging Rodriguez et al. (2012). In this work, we assume that a set of stable grasp point locations is given, along with the corresponding pregrasp stance, according to the chosen instance of the taxonomy (*grasp type*, see Sect. 2.1.1). For instance, we design the body grasps shown in Fig. 4 for the *pipe-shoulder* and *pipe-front* examples of Fig. 1. In those cases, we parametrize the grasp via the contact frames shown. More generally, we define n operational frames on the robot body. The pose of each one, denoted by: $\mathbf{p}_{\text{gr, i}}$ ($i = 1 \ldots n$), should be servoed to a desired pose: $\mathbf{p}_{\text{grdes, i}}$. This corresponds to n set-point objectives (41). Note, from Fig. 3, that pre-grasp and release are only performed when the robot is standing, in double support. Thus, both foot contact constraints are added as well as the set point task on

the CoM that is needed to maintain balance. In summary, the pregrasp and releasing optimization problems can both be formulated as:

$$\begin{aligned} \underset{\mathbf{x}}{\text{argmin}} \quad & \sum_{i=1}^{n} w_{\text{gr,i}} f_{\text{sp,i}}(\mathbf{x}, \mathbf{p}_{\text{grdes, i}}) + w_{\text{c}} f_{\text{sp}}(\mathbf{x}, \mathbf{c}_{\text{des}}) + \\ & w_{\text{pos}} f_{\text{pos}}(\mathbf{x}, \mathbf{q}_{\text{des}}) + w_{\boldsymbol{\lambda}} f_{\boldsymbol{\lambda}}(\mathbf{x}), \\ \text{subject to} \quad & \ddot{\mathbf{r}}_{\text{left}} = \mathbf{0}, \\ & \ddot{\mathbf{r}}_{\text{right}} = \mathbf{0}, \\ & \mathbf{A}_{\text{base}}\mathbf{x} \leq \mathbf{b}_{\text{base}}. \end{aligned} \tag{46}$$

4.3.4 Squeezing

The *squeeze* state is very similar in formulation to pregrasping and releasing, and it is also performed with the robot standing in double support. However, the $\mathbf{p}_{\text{grdes, i}}$ are defined by the expected contact points on the object, instead of the pregrasp stance. The second difference is in the servoing function. To realize the grasp, set-point tasks in the operational space can be used, similar to pregrasp/release. However, we prefer to include the wrench information related to the contact, $\mathbf{h}_{\text{gr}}$, when available. For example, a simple guarded-move algorithm can be added to the set-point task to signal stopping above some threshold on the wrench. Alternatively, force control can be used for the same purpose. To generalize the different grasping methods, we write f_{gr} as a function of $\mathbf{h}_{\text{gr}}$. In summary, the squeezing optimization problem is:

$$\begin{aligned} \underset{\mathbf{x}}{\text{argmin}} \quad & \sum_{i=1}^{n} w_{\text{gr,i}} f_{\text{gr,i}}(\mathbf{x}, \mathbf{p}_{\text{grdes, i}}, \mathbf{h}_{\text{gr}}) + \\ & w_{\text{c}} f_{\text{sp}}(\mathbf{x}, \mathbf{c}_{\text{des}}) + w_{\text{pos}} f_{\text{pos}}(\mathbf{x}, \mathbf{q}_{\text{des}}) + \\ & w_{\boldsymbol{\lambda}} f_{\boldsymbol{\lambda}}(\mathbf{x}), \\ \text{subject to} \quad & \ddot{\mathbf{r}}_{\text{left}} = \mathbf{0}, \\ & \ddot{\mathbf{r}}_{\text{right}} = \mathbf{0}, \\ & \mathbf{A}_{\text{base}}\mathbf{x} \leq \mathbf{b}_{\text{base}}. \end{aligned} \tag{47}$$

4.3.5 Holding the object while lifting, carrying, and placing it

After successfully squeezing the object, a grasp is maintained by the *hold* state. We chose to formalize this via null motion constraints between the grasping points on the robot body. In principle, it is possible to constrain all permutations of contact pairs. However, this results in numerical problems for the solver, if the closed kinematic chains are not handled properly. Instead, we only use $n-1$ constraints, defined by all pairs of points $(i, i+1)$, with $i = 1, \ldots, n-1$. This approach does not impede changes in object configurations (e.g., motions while holding), but ensures that the grasp form is maintained. This principle comes from caging Rodriguez et al. (2012) where the object being caged moves together with a properly formed cage. Once the object is held, it can be considered as part of the robot. We can then define an operational frame related to the object, $\mathbf{o}$, and servo its pose via a set-point task $f_{\text{sp}}(\mathbf{x}, \mathbf{o}_{\text{des}})$. The motion could be improved with a trajectory generator, possibly including collision avoidance. For collaborative carrying, the hold state is to be realized while *walking* (RSS, DS, LSS), *lifting*, and *placing* (see Fig. 3).

For *holding while lifting* and *holding while placing*, the optimization problem is:

$$\begin{aligned} \underset{\mathbf{x}}{\text{argmin}} \quad & w_{\text{o}} f_{\text{sp}}(\mathbf{x}, \mathbf{o}_{\text{des}}) + w_{\text{c}} f_{\text{sp}}(\mathbf{x}, \mathbf{c}_{\text{des}}) + \\ & w_{\text{pos}} f_{\text{pos}}(\mathbf{x}, \mathbf{q}_{\text{des}}) + w_{\boldsymbol{\lambda}} f_{\boldsymbol{\lambda}}(\mathbf{x}), \\ \text{subject to} \quad & \ddot{\mathbf{p}}_{\text{gr,1}} - \ddot{\mathbf{p}}_{\text{gr,2}} = \mathbf{0}, \\ & \vdots \\ & \ddot{\mathbf{p}}_{\text{gr,n-1}} - \ddot{\mathbf{p}}_{\text{gr,n}} = \mathbf{0}, \\ & \ddot{\mathbf{r}}_{\text{left}} = \mathbf{0}, \\ & \ddot{\mathbf{r}}_{\text{right}} = \mathbf{0}, \\ & \mathbf{A}_{\text{base}}\mathbf{x} \leq \mathbf{b}_{\text{base}}. \end{aligned} \tag{48}$$

For *holding during double support*, the optimization problem is similar, except that $w_{\text{c}} f_{\text{sp}}(\mathbf{x}, \mathbf{c}_{\text{des}})$ is replaced by $w_{\text{c}} f_{\text{tr}}(\mathbf{x}, \mathbf{c}_{\text{des}}(t))$, with $\mathbf{c}_{\text{des}}(t)$ output by either the follower or leader WPG. Furthermore, as for all walking states, the torso orientation is added. Therefore, the optimization problem is:

$$\begin{aligned} \underset{\mathbf{x}}{\text{argmin}} \quad & w_{\text{c}} f_{\text{tr}}(\mathbf{x}, \mathbf{c}_{\text{des}}(t)) + w_{\text{ts}} f_{\text{sp}}(\mathbf{x}, \boldsymbol{\theta}_{\text{tsdes}}) + \\ & w_{\text{pos}} f_{\text{pos}}(\mathbf{x}, \mathbf{q}_{\text{des}}) + w_{\boldsymbol{\lambda}} f_{\boldsymbol{\lambda}}(\mathbf{x}), \\ \text{subject to} \quad & \ddot{\mathbf{p}}_{\text{gr,1}} - \ddot{\mathbf{p}}_{\text{gr,2}} = \mathbf{0}, \\ & \vdots \\ & \ddot{\mathbf{p}}_{\text{gr,n-1}} - \ddot{\mathbf{p}}_{\text{gr,n}} = \mathbf{0}, \\ & \ddot{\mathbf{r}}_{\text{left}} = \mathbf{0}, \\ & \ddot{\mathbf{r}}_{\text{right}} = \mathbf{0}, \\ & \mathbf{A}_{\text{base}}\mathbf{x} \leq \mathbf{b}_{\text{base}}. \end{aligned} \tag{49}$$

Finally, for *holding during single support*, the optimization problem is:

$$\begin{aligned} \underset{\mathbf{x}}{\text{argmin}} \quad & w_{\text{c}} f_{\text{tr}}(\mathbf{x}, \mathbf{c}_{\text{des}}(t)) + w_{\text{sw}} f_{\text{tr}}(\mathbf{x}, \mathbf{r}_{\text{swdes}}(t)) + \\ & w_{\text{ts}} f_{\text{sp}}(\mathbf{x}, \boldsymbol{\theta}_{\text{tsdes}}) + w_{\text{pos}} f_{\text{pos}}(\mathbf{x}, \mathbf{q}_{\text{des}}) + \\ & w_{\boldsymbol{\lambda}} f_{\boldsymbol{\lambda}}(\mathbf{x}), \\ \text{subject to} \quad & \ddot{\mathbf{p}}_{\text{gr,1}} - \ddot{\mathbf{p}}_{\text{gr,2}} = \mathbf{0}, \\ & \vdots \\ & \ddot{\mathbf{p}}_{\text{gr,n-1}} - \ddot{\mathbf{p}}_{\text{gr,n}} = \mathbf{0}, \\ & \ddot{\mathbf{r}}_{\text{sup}} = \mathbf{0}, \\ & \mathbf{A}_{\text{base}}\mathbf{x} \leq \mathbf{b}_{\text{base}}. \end{aligned} \tag{50}$$

Again, $\mathbf{c}_{\text{des}}(t)$ can be output by the follower or leader WPG.

5 Experiments and Results

This section shows how we validated the proposed framework, first in dynamic simulations, and then with experiments on an HRP-4 humanoid from Kawada Industries, with customized ATI Mini40 force/torque sensors in the wrists. Details on using walking and whole-body controllers on the HRP-4 can be found in Kajita et al. (2010). With regards to that work, we are using a similar stabilizing controller to control ground reaction forces, but with a different WPG and whole-body controller, which have been described in the two previous sections. In all experiments, for the walk, we set the swing duration to $t_{\text{swf}} = 0.7\text{s}$, and the stepping height to $r_{\text{sth}} = 0.07\text{m}$. The average forward walking velocity, in the leader CoM trajectory (20), is set to 0.1m.s^{-1}.

5.1 Simulations

The base functionality of the WPG was previously verified and tested, with the results presented in Agravante et al. (2016). Complimentary to those results, we concentrate on the implications of carrying an object together with a human. Specifically, due to the carried object weight, a negative force component in the z direction will be present. Equation (7) shows that an important negative $f^{\text{z}}_{\text{ext}}$ will increase the robustness to external wrenches in x and y directions, by reducing their net effect on the ZMP. Furthermore, if $f^{\text{z}}_{\text{ext}}$ is large enough to be comparable to the robot mass, it reduces the acceleration effects. An intuitive way to interpret these effects, is that the added weight lowers the CoM of the combined (robot and object) system. Fig. 6 shows two simulations showing this effect. In both figures a 150N force pulling the robot forward is applied. To compensate this, the WPG produces a change in the posture. This was previously demonstrated in Agravante et al. (2016). However, on the right figure a weight of $\approx 500\text{N}$ is added. The end result is a less drastic posture change from the WPG output. From this, we see that carrying heavier objects actually helps the stability of the humanoid (obviously, assuming the robot motors can handle the extra load). In this analysis, we consider that the human is also carrying approximately half the object weight, thereby negating torques that would appear with the robot carrying the object alone. However, the presented WPG can also handle torques, also shown in Agravante et al. (2016).

As for whole body control, we present simulations on the designed *pregrasp* and *squeezing* postures for carrying (respectively (46) and (47)). For these, we must define the control frame poses $\mathbf{p}_{\text{gr, i}}$ on the surface of the robot body parts (e.g., shoulder, chest, hands, etc.), and compute the corresponding Jacobians. Some postures have been shown in Fig. 4. However, due to hardware issues (broken wrist joint), we also had to design *one-handed* versions

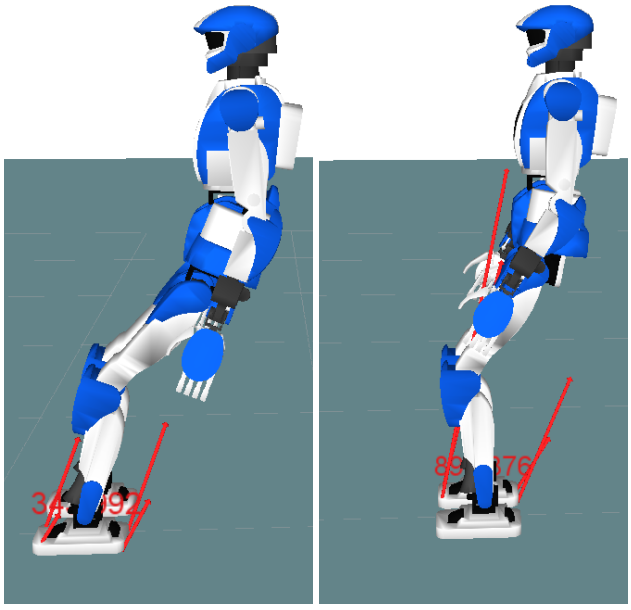

Figure 6. Screenshots of two simulations with 150N force pulling to the left. In the simulation on the right, a vertical force component $f^{\text{z}}_{\text{ext}} \approx -500\text{N}$ is also added.

of these, shown in Fig. 7, along with a grasping motion of the hand (bottom figures). On the HRP-4 hand, the thumb is controlled by one motor, and the four other fingers are actuated together by a second motor. Hence, the four fingers open and close together during squeezing, and this motion is defined by a single *joint position*. Another point of interest is the left arm motion in the front-wrap closing (middle right snapshot in Fig. 7). This is caused by objective function $f_{\text{sp}}(\mathbf{x}, \mathbf{c}_{\text{des}})$ in (47)), which keeps the ground projection of the CoM near the center of the support polygon. Since the squeeze motion moves the chest frame forward, the QP solver uses the left arm to realize this objective.

For integrating the walk and the whole-body controller, recall that at each instant either of the pattern generators (follower (19) or leader (22)) provides a reference CoM position, velocity and acceleration. Since the WPG and whole-body controller run at different loop rates (100ms and 5ms respectively), a simple Euler integration is used to interpolate the reference. In Fig. 8, we compare the CoM and ZMP positions, as requested by the WPG (here: *leader*, with no external forces) and as achieved by the whole body controller. The plot shows that the CoM is tracked well enough and that the robot is actually walking at 0.1m.s^{-1}, as requested. As for the ZMP, note that there is currently no explicit task within the whole-body controller to track it. However, on the real HRP-4 platform, this is done within a low-level *stabilizer* which acts as a final

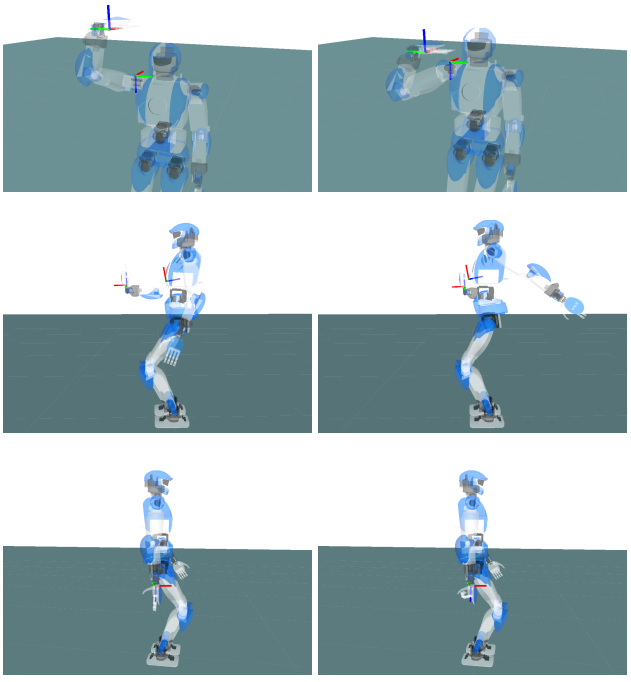

Figure 7. One-handed pregrasp (left) and squeezing postures (right). From top to bottom: shoulder-mounted body grasp, front-wrap body grasp, and right hand grasp.

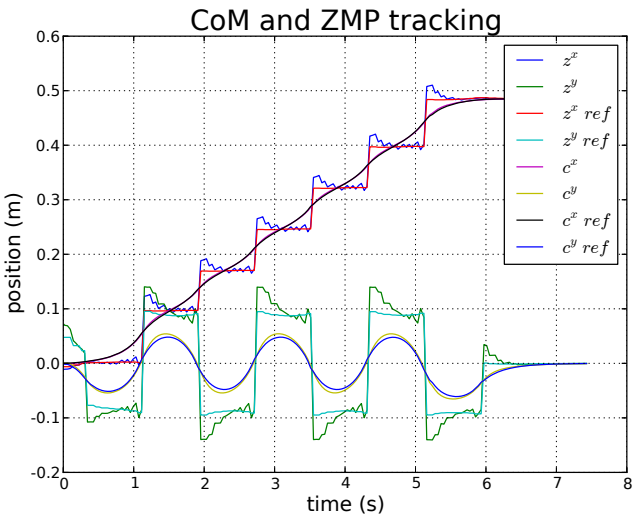

Figure 8. Tracking task of the CoM using the WPG-generated reference CoM along with a comparison of the resulting ZMP.

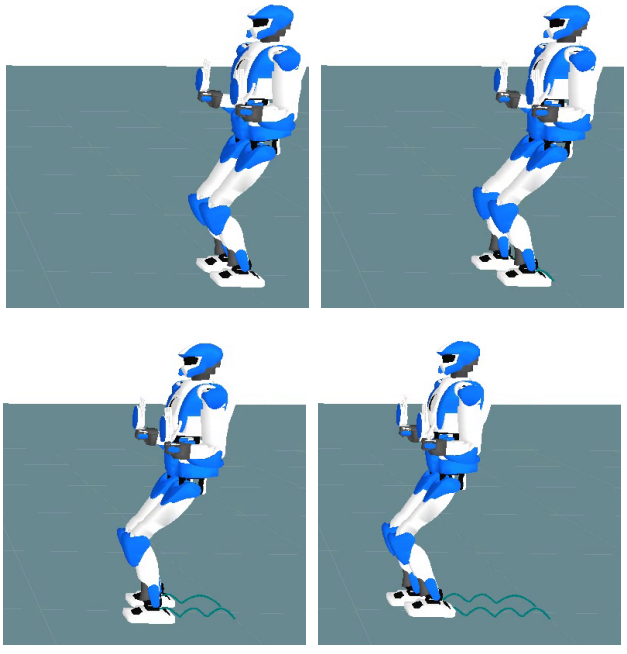

Figure 9. Image sequence of walking while holding, using a front-wrap body grasp.

closed-loop controller to ensure balance Kajita et al. (2010). Therefore, for the ZMP, we see more difference between the curves, although our tests (both dynamic simulations, and real experiments) show that this does not affect the robot balance. Also note that, in terms of integration within the whole-body controller, the two WPGs presented in Sect. 3 function similarly. The only difference is in needing force control in the leader case, and only force sensing in the follower case.

Lastly, we present simulations of walking as *leader*, while holding (Sect.4.3.5). Image sequences of walking while holding using two-handed front-wrap, and shoulder-mounted body grasps, are respectively shown in Figures 9 and 10. Although only some chosen examples are illustrated, either of the two WPG may be used, along with any of the grasps. These examples demonstrate that we are capable of properly servoing the CoM, without the need for an approximate frame, as in our previous works Bussy et al. (2012b); Agravante et al. (2014). This feature allows us to freely modify the posture and to add constraints at ease.

5.2 Real robot experiments

After having verified the framework in simulation, we moved on to experiments on the real HRP-4. Representative tests are shown in the video, attached to this paper, and visible at: https://www.youtube.com/watch?v=VYwZU4_7sMA. Screenshots are also shown in Fig. 11. The figure shows (left to right, top then bottom): shoulder-mounted box carrying as leader, front-wrap box carrying as leader, hand grasped stretcher carrying as follower, and hand grasped bucket carrying as follower. These correspond to four of the six examples introduced in Fig. 1. All four

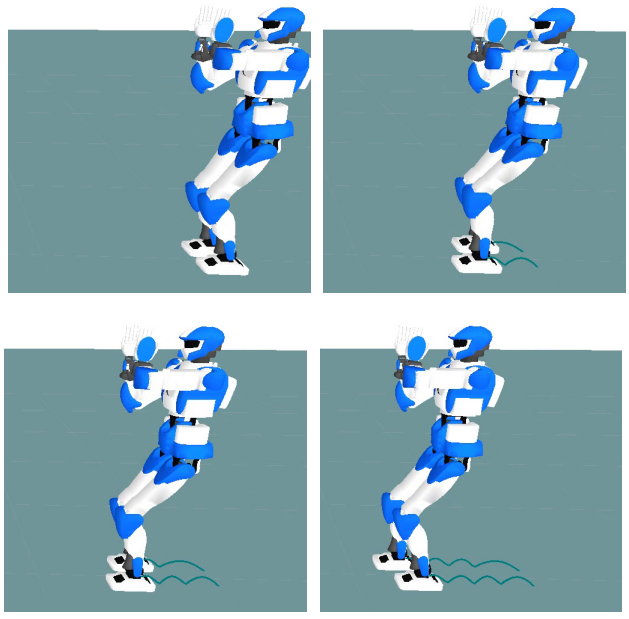

Figure 10. Image sequence of walking while holding, using a shoulder-mounted body grasp.

collaborative carrying scenarios were successful, with the robot acting both as leader and as follower.

Figure 11. Experiments with the HRP-4 carrying various objects in collaboration with a human.

Relevant data from the stretcher carrying task, with the robot walking as follower (bottom left in Fig. 11), are shown in Fig. 12. The top plot shows the CoM and ZMP reference signals, generated by the WPG (19), while the bottom figure shows the forward (pulling) component of

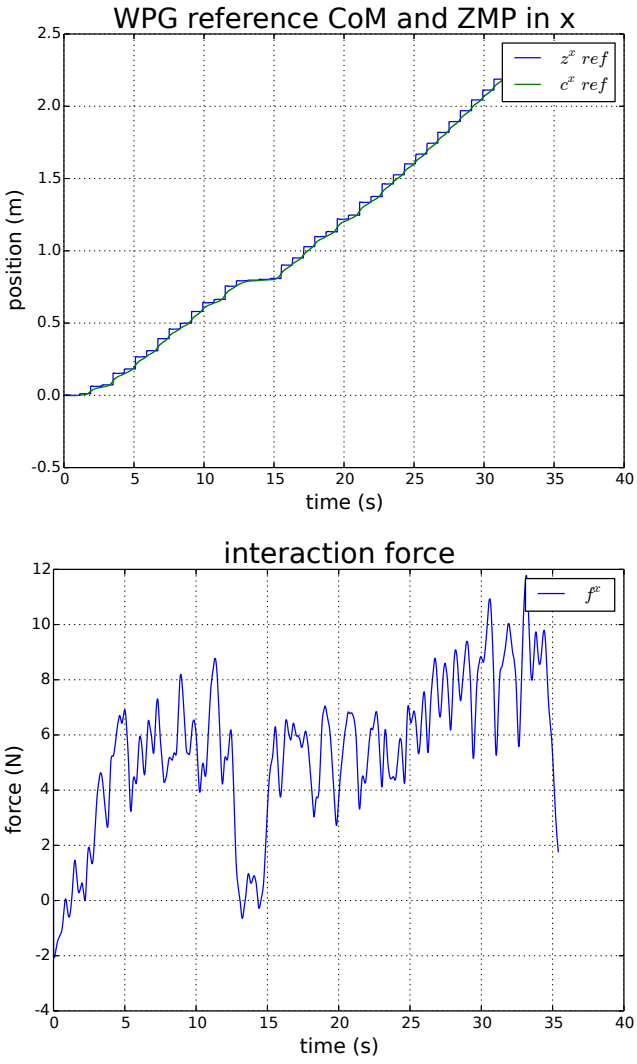

Figure 12. Data from a stretcher carrying scenario with the robot walking as a follower. Top: reference Com and ZMP (as generated by the WPG). Bottom: forward (pulling) component of the interaction force.

the interaction force, measured by the force sensors, and transformed to the CoM frame, $f^{\mathrm{x}}_{\mathrm{ext}}$. Note the pause in the walk (top figure), around the 13 to 15 second mark. This was due to the human stopping (see also the strong decrease of interaction force in the bottom figure), and the robot appropriately reacting. Throughout the experiment, the CoM and ZMP reference values are properly adapted according to the external force, as the robot follows the human intention.

Although these results showed that the overall approach worked well, the real experiments introduce some issues which were not present in simulation. Notice that on the hand grasps (bottom snapshots), the pose had to be redesigned, so that the wrists could support the object

weight. This had to be done because of the low payload capacity of the HRP-4 fingers. Furthermore, the front-wrap carry required the *one-hand* version, because of a broken wrist joint at that moment. Moreover, force sensing is available only in the robot wrists, and no force measurement is available at the other contact points (e.g., on the shoulder and chest). Thus, we had to take this into account when using the sensed data, typically in the *follower* WPG. Lastly, minor issues with the grasp stability could have also been improved, if local force/contact sensing in the other body parts was available.

6 Conclusion

This article explores several aspects of the human-humanoid collaborative carrying task. We started by looking at this task as a whole. To do this, we created a taxonomy based on observing several cases of human teams. We then tried to infer from this, the core principles of collaborative carrying, in order to program it on a humanoid robot. To this end, we created a generic Finite State Machine, encompassing all of the necessary subtasks. Next, we revisited locomotion and balance in relation to physical interaction. For this, we designed two walking pattern generators that not only take into account the physical interaction constraints, but also use it accordingly, to operate as a follower or leader. Finally, we discuss how all of this can be designed as objectives and constraints of an optimization problem for a whole-body controller. We then present simulations and real test cases on the HRP-4 humanoid.

Although the approach presented here proved successful, there are still several areas that can be largely improved with future works. Firstly, one key issue, outlined by the real experiments, is the need for force estimation. Related to this, we outlined the need for distributed force sensing on the entire robot body, instead of only on the wrist. Distributed tactile sensors can improve body grasps as tested with the HRP-2 in Mittendorfer et al. (2015). Another key improvement, that we only briefly mentioned, concerns the wrench prediction model for better proactive behaviors. A current limitation is that the wrench is simply predicted to be constant over the preview horizon. However, since we believe the framework is very well suited for including proactivity, a better perceptual model is necessary. This requires integrating active human perception for intention recognition, a difficult challenge, but also an active research area in physical human-robot interaction. Concerning the walk, although the WPG presented here is in simplified form, its core concepts do not conflict with improvements such as Brasseur et al. (2015) which add robustness, allowing stair climbing. Apart from improving the WPG itself, its integration to whole-body control can also be improved, with works such as Sherikov et al. (2014) which aims at combining the separate QPs. Since this would require reconsidering the QP weight tuning, hierarchies could be added to the constrained optimization problem, as in Escande et al. (2014a). Lastly, a limiting factor for our real experiments was the low-level closed loop stabilizing controller of the HRP-4, which modifies the final joint references sent to the robot actuators Kajita et al. (2010). To solve this, a dedicated stabilizer, consistent with our control framework, should be designed.

7 Acknowledgement

This work is supported in part by the FP7 KoroiBot project (`www.koroibot.eu`) and the PSPC ROMEO 2 (`www.projetromeo.com`) national project.

References

Agravante DJ, Cherubini A, Bussy A, Gergondet P and Kheddar A (2014) Collaborative Human-Humanoid Carrying Using Vision and Haptic Sensing. In: *IEEE International Conference on Robotics and Automation*. pp. 607–612.

Agravante DJ, Cherubini A, Bussy A and Kheddar A (2013) Human-humanoid joint haptic table carrying task with height stabilization using vision. In: *IEEE/RSJ International Conference on Robots and Intelligent Systems*. pp. 4609–4614.

Agravante DJ, Sherikov A, Wieber PB, Cherubini A and Kheddar A (2016) Walking pattern generators designed for physical collaboration. In: *IEEE International Conference on Robotics and Automation*. To appear.

Audren H, Vaillant J, Kheddar A, Escande A, Kaneko K and Yoshida E (2014) Model preview control in multi-contact motion-application to a humanoid robot. In: *IEEE/RSJ International Conference on Robots and Intelligent Systems*. pp. 4030–4035.

Bellaccini M, Lanari L, Paolillo A and Vendittelli M (2014) Manual guidance of humanoid robots without force sensors: Preliminary experiments with NAO. In: *IEEE International Conference on Robotics and Automation*. pp. 1184–1189.

Berger E, Vogt D, Haji-Ghassemi N, Jung B and Amor HB (2013) Inferring guidance information in cooperative human-robot tasks. In: *IEEE-RAS International Conference on Humanoid Robots*. pp. 124–129.

Bouyarmane K and Kheddar A (2011) Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations. In: *IEEE/RSJ International Conference on Robots and Intelligent Systems*. pp. 4414–4419.

Brasseur C, Sherikov A, Collette C, Dimitrov D and Wieber PB (2015) A robust linear MPC approach to online generation of 3D biped walking motion. In: *IEEE-RAS International Conference on Humanoid Robots*. pp. 595–601.

Brossette S, Escande A, Duchemin G, Chrétien B and Kheddar A (2015) Humanoid posture generation on non-Euclidean manifolds. In: *IEEE-RAS International Conference on Humanoid Robots*. Seoul, Korea, pp. 352–358.

Bussy A, Gergondet P, Kheddar A, Keith F and Crosnier A (2012a) Proactive behavior of a humanoid robot in a haptic transportation task with a human partner. In: *IEEE International Symposium on Robot and Human Interactive Communication*. pp. 962–967.

Bussy A, Kheddar A, Crosnier A and Keith F (2012b) Human-humanoid haptic joint object transportation case study. In: *IEEE/RSJ International Conference on Robots and Intelligent Systems*. pp. 3633–3638.

Chung W, Fu LC and Hsu SH (2008) Motion Control. In: Siciliano B and Khatib O (eds.) *Springer handbook of robotics*, chapter 6. Springer, pp. 133–159.

Cutkosky M (1989) On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation* 5(3): 269–279.

Escande A, Mansard N and Wieber PB (2014a) Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research* .

Escande A, Miossec S, Benallegue M and Kheddar A (2014b) A strictly convex hull for computing proximity distances with continuous gradients. *IEEE Transactions on Robotics* 30(3): 666–678.

Evrard P and Kheddar A (2009a) Homotopy switching model for dyad haptic interaction in physical collaborative tasks. In: *EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. IEEE, pp. 45–50.

Evrard P and Kheddar A (2009b) Homotopy switching model for dyad haptic interaction in physical collaborative tasks. In: *IEEE International Symposium on Robot and Human Interactive Communication*. Toyama, Japan, pp. 1–6.

Featherstone R (2008) *Rigid body dynamics algorithms*. Springer.

Herdt A, Diedam H, Wieber PB, Dimitrov D, Mombaur K and Diehl M (2010) Online walking motion generation with automatic footstep placement. *Advanced Robotics* 24(5-6): 719–737.

Herzog A, Rotella N, Mason S, Grimminger F, Schaal S and Righetti L (2015) Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid. *Autonomous Robots* 40(3): 473–491.

Hirata Y and Kosuge K (2000) Distributed robot helpers handling a single object in cooperation with a human. In: *IEEE International Conference on Robotics and Automation*, volume 1. pp. 458–463 vol.1.

Ikemoto S, Amor HB, Minato T, Jung B and Ishiguro H (2012) Physical Human-Robot Interaction: Mutual Learning and Adaptation. *IEEE Robotics Automation Magazine* 19(4): 24–35.

Ikeura R, Monden H and Inooka H (1994) Cooperative motion control of a robot and a human. In: *International Workshop on Robot and Human Communication*. Nogoya, Japan, pp. 112–117.

Jarrassé N, Charalambous T and Burdet E (2012) A framework to describe, analyze and generate interactive motor behaviors. *PloS one* 7(11): e49945.

Kajita S, Kanehiro F, Kaneko K, Fujiwara K, Harada K, Yokoi K and Hirukawa H (2003) Biped walking pattern generation by using preview control of zero-moment point. In: *IEEE International Conference on Robotics and Automation*, volume 2. pp. 1620–1626.

Kajita S, Morisawa M, Miura K, Nakaoka S, Harada K, Kaneko K, Kanehiro F and Yokoi K (2010) Biped walking stabilization based on linear inverted pendulum tracking. In: *IEEE/RSJ International Conference on Robots and Intelligent Systems*. pp. 4489–4496.

Khatib O (1999) Mobile manipulation: The robotic assistant. *Robotics and Autonomous Systems* 26(2–3): 175–183. Field and Service Robotics.

Kim KS, Llado T and Sentis L (2016) Full-body collision detection and reaction with omnidirectional mobile platforms: a step towards safe human–robot interaction. *Autonomous Robots* 40(2): 325–341.

Koolen T, de Boer T, Rebula J, Goswami A and Pratt J (2012) Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models. *International Journal of Robotics Research* 31(9): 1094–1113.

Kosuge K, Sato M and Kazamura N (2000) Mobile robot helper. In: *IEEE International Conference on Robotics and Automation*, volume 1. pp. 583–588 vol.1.

Lafaye J, Gouaillier D and Wieber PB (2014) Linear model predictive control of the locomotion of Pepper, a humanoid robot with omnidirectional wheels. In: *IEEE-RAS International Conference on Humanoid Robots*. pp. 336–341.

Lawitzky M, Mortl A and Hirche S (2010) Load sharing in human-robot cooperative manipulation. In: *IEEE International Symposium on Robot and Human Interactive Communication*. pp. 185–191.

Madan CE, Kucukyilmaz A, Sezgin TM and Basdogan C (2015) Recognition of Haptic Interaction Patterns in Dyadic Joint Object Manipulation. *IEEE Transactions on Haptics* 8(1): 54–66.

Mansard N, Stasse O, Evrard P and Kheddar A (2009) A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks. In: *International Conference on Advanced Robotics*. IEEE, pp. 1–6.

McGill S and Lee D (2011) Cooperative humanoid stretcher manipulation and locomotion. In: *IEEE-RAS International Conference on Humanoid Robots*. pp. 429–433.

Medina JR, Lorenz T and Hirche S (2015) Synthesizing Anticipatory Haptic Assistance Considering Human Behavior Uncertainty. *IEEE Transactions on Robotics* 31(1): 180–190.

Mittendorfer P, Yoshida E and Cheng G (2015) Realizing whole-body tactile interactions with a self-organizing, multi-modal artificial skin on a humanoid robot. *Advanced Robotics* 29(1): 51–67.

Mörtl A, Lawitzky M, Kucukyilmaz A, Sezgin M, Basdogan C and Hirche S (2012) The role of roles: Physical cooperation between humans and robots. *International Journal of Robotics Research* 31(13): 1656–1674.

Nocedal J and Wright S (2000) *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York. ISBN 9780387987934.

Pratt J, Koolen T, de Boer T, Rebula J, Cotton S, Carff J, Johnson M and Neuhaus P (2012) Capturability-based analysis and control of legged locomotion, Part 2: Application to M2V2, a lower-body humanoid. *International Journal of Robotics Research* 31(10): 1117–1133.

Reed K and Peshkin M (2008) Physical Collaboration of Human-Human and Human-Robot Teams. *Haptics, IEEE Transactions on* 1(2): 108–120.

Rodriguez A, Mason MT and Ferry S (2012) From caging to grasping. *International Journal of Robotics Research* .

Sheng W, Thobbi A and Gu Y (2015) An Integrated Framework for Human Robot Collaborative Manipulation. *IEEE Transactions on Cybernetics* 45(10): 2030–2041.

Sherikov A, Dimitrov D and Wieber PB (2014) Whole body motion controller with long-term balance constraints. In: *IEEE-RAS International Conference on Humanoid Robots*. pp. 444–450.

Stückler J and Behnke S (2011) Following human guidance to cooperatively carry a large object. In: *IEEE-RAS International Conference on Humanoid Robots*. pp. 218–223.

Vaillant J, Kheddar A, Audren H, Keith F, Brossette S, Escande A, Bouyarmane K, Kaneko K, Morisawa M, Gergondet P, Yoshida E, Kajita S and Kanehiro F (2016) Multi-contact vertical ladder climbing with an HRP-2 humanoid. *Autonomous Robots* 40(3): 561–580.

Wang H and Kosuge K (2012) Control of a Robot Dancer for Enhancing Haptic Human-Robot Interaction in Waltz. *IEEE Transactions on Haptics* 5(3): 264–273.

Wieber PB, Tedrake R and Kuindersma S (2015) Modeling and control of legged robots. In: Siciliano B and Khatib O (eds.) *Springer handbook of robotics*, second edition, chapter 48. Springer.

Wu MH, Ogawa S and Konno A (2016) Symmetry position/force hybrid control for cooperative object transportation using multiple humanoid robots. *Advanced Robotics* 30(2): 131–149.

Yokoyama K, Handa H, Isozumi T, Fukase Y, Kaneko K, Kanehiro F, Kawai Y, Tomita F and Hirukawa H (2003) Cooperative works by a human and a humanoid robot. In: *IEEE International Conference on Robotics and Automation*, volume 3. pp. 2985–2991.