Human-Humanoid Collaborative Carrying
Don Joven Agravante, Andrea Cherubini, Alexander Sherikov, Pierre-Brice Wieber and Abderrahmane Kheddar, Senior Member, IEEE

Abstract—This paper contributes to the field of physical human-robot collaboration. We present a complete control framework which aims at making humanoid robots capable of carrying objects together with humans. Firstly, we design a template identifying the primitive subtasks necessary for collaborative carrying. Then, these subtasks are formulated as 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 carrying experiments, using the HRP-4 humanoid robot.

Index Terms—Physical human-robot interaction, humanoid robotics, task-space optimization control.

I. INTRODUCTION

PHYSICAL human-robot collaboration implies that the interaction forces between the human and robot must be used to achieve a common goal. 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 experiences. Therefore, a humanoid with a similar range of motion and sensing has the potential to be an intuitive interface. Within physical collaboration, carrying objects with a human in various poses and situations is a problem that is rich, unexplored and has high potential for practical application.

Early work on enabling human-humanoid carrying was done in [1] via the Humanoid Robotics Project (HRP), where the HRP-2P humanoid cooperates with a human for a panel transportation task. Disregarding the legged aspect, even earlier work in this topic was done in [2]. The authors used mobile manipulator robots as in [3], with wheels instead of legs. That work revealed an important issue: coordinating the motion of the mobile base with that of the upper robot body and with the human intention (generally represented by the interaction force). More recent examples of collaborative carrying mobile manipulators include [4], [5], [6]. Although these works discuss the handling of interaction forces and coordination, balance and integrity constraints handling is missing. Indeed, as discussed in [7], [8], even wheeled robots can fall over in challenging scenarios. One of the main contributions of this paper is to tackle the coupling of balanced legged locomotion and collaborative manipulation.

The task of collaborative carrying has also been tested on small scale humanoid platforms, e.g., NAO in [9]. However [9] focused on the use of internal sensors, instead of the wrist force/torque sensors commonly used in physical human-robot interaction. NAO is also used in [10], where the capture point [11], [12] guides walking; a different more robust variant of this idea is detailed in [13] with iCub. A similar work is [14], where Darwin robots carry a stretcher (no human is participating). When only robots are used, e.g., [15], the interest is turned to multi-robot synchronization and communication. Both multi-robot and human aspects are considered in [16], while [17] addresses table lifting with NAO, using machine learning to improve interaction.

Understanding and improving physical human-robot interaction is a very broad and active research field. For example, [18], [19], [20], [21], [22], [23] study human-human haptic interaction and apply it to human-robot teams. Role allocation and role switching (e.g., between leader and follower) have been studied in [24], [20], [25]. The common ground of these methods is to infer human intention and steer the robot to a proactive or versatile physical interaction behavior. The authors in [26] address mutual learning and adaptation, whereas [27] focuses on the uncertainty of human behavior prediction. Haptic interaction recognition using supervised learning is presented in [28]. A bounded memory model is investigated in [29] for human-robot adaptation. Recently, differential game theory is used to design an observer of human control strategy [30]. All these works, and others, can constitute a plugin brick in follower/leader role allocation in the control scheme we propose, see Sec. VI-C.

Our previous framework, specific to table carrying [31], could not be extended to any posture (and therefore objects), since the used walking pattern generator (WPG) [32] and whole-body controller considered the center of mass (CoM) to be coincident with the robot waist. Hence, any posture moving the CoM away from the waist (e.g., extending the arms or leaning with the chest) would fail. More generally, the
WPG of [32] worked well for stand-alone walking, but was not designed for physical interaction, as explained in [33].

Given these limitations, we reformulate the entire pipeline to have a generic framework for humanoid collaborative carrying with viable perspective/view on how to generate humanoid motion under postural constraints (dictated by the object to be carried jointly and the taxonomy of the task), and under sustained external forces (that can no longer be considered as external 'perturbations'), and with either leader or follower behaviors (that can be sequenced smoothly or combined at will). In particular, our contributions are:

- While various pHRI controllers have been proposed for robotic arms or task-space leader/follower pHRI, our work is the first to go beyond in expanding pHRI control to walking pattern gaits. This enables walking under sustained forces in either leader or follower modes that can be sequenced or combined at will.
- Subsequently, our work is also the first to explicitly integrate the force wrench inherent to pHRI in the model predictive formulation of this new walking pattern generator. We detail the modeling choices and, more importantly, the integration with whole-body control.
- Our whole-body framework can simultaneously and uniformly account for both walking and carrying under postural constraints. We explain how collaborative carrying can be formulated and implemented as an optimization problem.
- The framework is validated in a series of experiments on the HRP-4 humanoid robot, with a variety of robot roles (leader/follower), grasp types (hand/body) and carried objects (different shapes and sizes).
- We provide the source code of all these components.

We structured the paper as follows. Section II presents the collaborative carrying taxonomy, along with the required primitive subtasks. Section III provides a review of task-space control using quadratic optimization which is used throughout our work. The walking pattern generators accounting for physical interaction are presented in Section IV. Section V describes our optimization framework for whole-body control. Finally, Section VI presents the experimental validation.

II. THE TASK OF COLLABORATIVE CARRYING

To understand collaborative carrying, we take inspiration from how humans do it. This is done by creating a taxonomy, i.e., an abstraction layer that provides a scaffold for our control optimization framework. Then, we design a Finite State Machine (FSM) accounting for all collaborative carrying subtasks in order to map each state to an optimization problem.

A. A taxonomy of collaborative carrying

We consider the problem of having a pair of agents, whose goal is to move a specified object from one location to another. Figure 1 shows several real examples of human-human collaborative carrying (left), with the corresponding simulations of what we envision with a humanoid robot collaborator (right).
We assume that neither object nor agent composition can be changed afterwards, and consider the following relationships:

- **Agent-object relation (grasp type).** We consider two broad classes of grasp types: **hand grasps** and **body grasps**. **Hand grasps** are those with contact points located uniquely on the hand/gripper [34]. **Body grasps** are those that utilize grasp contacts on body parts not limited to the hand (e.g., arms, torso, see Fig. 1).
- **Agent-agent relation (relative pose).** In our taxonomy, we relate this to the inter-agent communication modes, touch and vision. For translations, we check whether direct touch between the agents is possible (near relation) or not (far). For the orientation, we consider the agents’ Field Of View (FOV), specifically: the nominal (when the perceiving agent is in a resting position), and the extended (as the agent looks around, by moving its body) FOV. Then, we can classify agents as facing front (other agent in the nominal FOV), side (other agent not in the nominal but in the extended FOV), back (other agent not in the extended FOV).

All six scenarios of Fig. 1 can be easily classified according to the proposed taxonomy. But more importantly, we are concerned with the practical implications of using the taxonomy to program a humanoid robot.

### B. Collaborative carrying as a Finite State Machine

To make a robot capable of collaborative carrying, we must decompose this complex task into subtasks that are easier to program. Formally, we use an FSM to describe the whole task, with subtasks as states. The FSM should be general enough for all cases encompassed by the taxonomy. The FSM that we create is specific to our experiments and hence minimalistic as our aim is to conduct the experiment assuming no gross variations in the setting conditions. Only light uncertainties in contact, sensing... are handled in the detailed implementation. However, the FSM is open to be programmed to handle cases where things could turn wrong. The current limitation however is that every unlucky situation should be programmed a priori (and this is not realistic either).

A useful decomposition is one where the states can be easily mapped to optimization problems. We first consider the state transitions. These should include brief periods where the motion is minimal. We assume that the robot is in quasi-static state (i.e., dynamic effects can be disregarded), with discrete changes in its contact state. The state transitions can be triggered either by relevant sensed variables (when available), or by human input (in case of shared autonomy). Considering this, a collaborative carrying FSM is shown in Fig. 2.

While walking, the feet contacts occur in a predictable pattern that can be used to define the walking states: left/right single support, and double support. To decompose grasping, we need a *pregrasp* posture, i.e., a waypoint between grasping and the other states. The next state, *squeeze*, moves the robot to generate predefined contacts between its body and the object. Figure 3 shows the pregrasp and squeezing postures for two body grasps. The *hold* state maintains the contacts between robot and object. Note that it must be active throughout the carrying walk. Finally, the *release* state, is simply the inverse of *squeeze*. Force or tactile sensors, when available, can trigger transitions between these states.

Figure 4 shows how the various parts of the framework fit together, and which section of the paper details which part. The active state in the FSM of Fig. 2 defines the subtask to be realized by the robot. Each subtask corresponds to two quadratic optimization problems (see Sec. III), one to be solved by the walking pattern generator (WPG), and a second one to be solved by the whole body controller (WBC). Specifically, the WPG (described in Sec. IV) computes the desired center of mass and footstep positions, depending...
on: the subtask, the robot role (leader or follower), and the external wrench. Then, the desired center of mass and footstep positions, along with the subtask to be realized, are input to the WBC (described in Sec. V) that yields the joint commands to be sent to the robot.

![Block diagram of the overall collaborative carrying framework (CoM: Center of Mass).](image)

Fig. 4. Block diagram of the overall collaborative carrying framework (CoM: Center of Mass).

## III. INTRODUCTION TO QUADRATIC OPTIMIZATION

Recently, optimization-based approaches have shown to be very effective for controlling humanoid robots. This is true for both WPG [35] and WBC, e.g. [36]. In general, we seek the robot control input, represented by the optimization argument $x$ that minimizes a collection of objective functions, subject to various constraints (from the robot hardware, environment, and task).

The formalism chosen here is largely based on [37]. We consider a weighted quadratic programming formulation which allows to use the $L^2$ norm to define a number of objective functions $f_i(x)$:

$$f_i(x) = \|A_i x - b_i\|^2,$$

so that all optimization problems are of the form:

$$\text{argmin} \quad \sum_i w_i \|A_i x - b_i\|^2$$

subject to $A_i x \leq b_c.$ (2a)

The control input $x$ is defined via the objectives (2a), which are regarded in accordance to their scalar weights $w_i > 0$. Equality and inequality constraints must also be satisfied; for the sake of space and readability, these are both expressed in a compact form (2b). When this is not possible, the optimization problem is infeasible and the constraints are said to be in conflict. Infeasibility can be accounted for, by relaxing the conflicting constraints as is common in hierarchical optimization [38], or as explained later in the paper for individual optimization problems.

Formulation (2) allows the use of off-the-shelf Quadratic Programming (QP) solvers implementing efficient algorithms and suitable for real-time applications [39].

The limitation of weighted-priority task space formulations is in the difficulty in automating or having a systematic procedure for tuning the weights $w_i$ among the tasks in the cost function. This has to be done ad-hoc in all our experiments and some gains may even changed depending on the state in which the robot is operating.

## IV. WALKING DESIGNED FOR PHYSICAL COLLABORATION

In robotics, walking has historically been treated separately from manipulation. However, manipulation and locomotion must be consistent, in particular during collaborative carrying. Eventually, both need to be thought of as parts of the whole-body control problem (discussed in the next section). In this section, we revisit the modeling of walking pattern generators (WPG), and redesign them with physical collaboration in mind. This section was partially published in [33]. Here, we recall the main points, namely the addition of the external wrench into the model, and the design of objectives and constraints, based on this wrench. Additionally, we better specify the usage of our new WPG in the collaborative carrying task.

### A. Modeling

First, we use a centroid-based dynamic model accounting for physical interaction. Three possible versions of such models were proposed in [40]:

1) a model with full knowledge of object and/or human that considers robot, human and object as a single system,

2) a model that considers the effects of the object and/or human as external wrenches applied at their contact locations,

3) a model that considers the effects of the object and/or human as external wrenches applied on the robot CoM.

The first model can be used (as was done in [40]) in simulation to control the robot, while having perfect knowledge of each subsystem (human, robot, object). However, for real applications with a human in the loop, it requires excessive processing and sensing, and is practically unfeasible. The second option yields a non-linear model, that is much more complicated than an inverted pendulum. Since contact due to physical interaction between the object-human and the robot does not contribute to locomotion (i.e. the contact does not support locomotion, but the robot still has to deal with it\(^1\)), we opt for the third model. That is, we use an inverted pendulum subject to an external force for its simplicity and generality in terms of implementation on a real humanoid robot.

The development of this reduced model is inspired by [35], and described in [33]. We separate the foot/ground contact forces from other interaction contact forces that are denoted by $h = [F^T \ n^T]^T \in \mathbb{R}^6$. 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 (CoM), e. As is common in the literature [32], [41], we aim at keeping the center of pressure $z$ (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 and that the angular momentum is constant. We also set the CoM at constant height $c^e$. Although this is a common assumption, recent Walking Pattern Generators – that relax such constraint – could also be used in our framework.

\(^1\)As a result, the humanoid built-in stabilizer for standalone walking is perturbed. In fact, a novel stabilization scheme could account for such a contact to be less conservative; this constitutes interesting future work.
Newton and Euler equations yield the following relationship between CoM and ZMP:

\[
\begin{align*}
\mathbf{z}^{x,y} &= \mathbf{c}^{x,y} - \left( \frac{\mathbf{c}^T \mathbf{c}}{g^2} \right) \mathbf{c}^{x,y} \\
&\quad + \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \left( \begin{bmatrix} \mathbf{b}^{x,y} \end{bmatrix} \mathbf{m} \frac{\mathbf{g}^T \mathbf{f} - f^T}{m} \right),
\end{align*}
\]  

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

\[
\mathbf{z}^{x,y} = \mathbf{c}^{x,y} - \left( \frac{\mathbf{c}^T \mathbf{c}}{g^2} \right) \mathbf{c}^{x,y},
\]  

From (3), we can infer that a heavier robot, lower CoM height or an external force aligned with the CoM, will all reduce the effects of the external wrench.

To generate smooth motions of the CoM, we assume its trajectory to be differentiable three times. Having defined:

\[
\begin{align*}
\mathbf{\hat{c}}_k &= \begin{bmatrix} \mathbf{c}_k^T \mathbf{\hat{c}}_k^T \mathbf{\hat{c}}_k^T \mathbf{\hat{c}}_k^T \mathbf{\hat{c}}_k^T \mathbf{\hat{c}}_k^T \mathbf{\hat{c}}_k^T \mathbf{\hat{c}}_k^T \mathbf{\hat{c}}_k^T \mathbf{\hat{c}}_k^T \end{bmatrix}^T, \\
\mathbf{u}_k &= \begin{bmatrix} \mathbf{u}_k^T \mathbf{u}_k^T \mathbf{u}_k^T \mathbf{u}_k^T \mathbf{u}_k^T \mathbf{u}_k^T \mathbf{u}_k^T \mathbf{u}_k^T \end{bmatrix}^T, \\
\mathbf{f}_k &= \begin{bmatrix} \mathbf{f}_k^T \mathbf{f}_k^T \mathbf{f}_k^T \mathbf{f}_k^T \mathbf{f}_k^T \mathbf{f}_k^T \mathbf{f}_k^T \mathbf{f}_k^T \mathbf{f}_k^T \end{bmatrix}^T,
\end{align*}
\]

we can obtain the following time-varying discrete time linear model:

\[
\begin{align*}
\mathbf{\hat{c}}_{k+1} &= \mathbf{\Gamma}_k \mathbf{\hat{c}}_k + \mathbf{\Psi}_k \mathbf{u}_k \\
\mathbf{z}_{k+1} &= \mathbf{\Phi}_{k+1} \mathbf{\hat{c}}_{k+1} + \mathbf{\Psi}_{k+1} \mathbf{f}_{k+1} \\
&= \mathbf{\Phi}_{k+1} \mathbf{\Gamma}_k \mathbf{\hat{c}}_k + \mathbf{\Phi}_{k+1} \mathbf{\Psi}_k \mathbf{u}_k + \mathbf{\Psi}_{k+1} \mathbf{f}_{k+1},
\end{align*}
\]

with matrices \( \mathbf{\Gamma}_k \), \( \mathbf{\Psi}_k \), \( \mathbf{\Phi}_{k+1} \) and \( \mathbf{\Psi}_{k+1} \) derived from (3) and detailed in [33]. Based on this model, we can now formulate the walking pattern generators.

### B. Model predictive control for walking

Model Predictive Control (MPC) consists in controlling a system so that future states are taken into account. A common MPC methodology consists in iteratively applying the model over \( N \) discrete steps (noted \( k = 1 \ldots N \)), resulting in a new problem formulation where the predicted states are a function of the current state and of the current and future control inputs.

Since the CoM trajectory is differentiable three times, we control it through the CoM jerk. Then, to apply MPC, we define:

- the control input \( \mathbf{x} = \begin{bmatrix} \mathbf{\tilde{u}}^T & \mathbf{\tilde{r}}^T \end{bmatrix}^T \), with \( \mathbf{\tilde{u}} \) and \( \mathbf{\tilde{r}} \) the concatenation, over the preview horizon, respectively of the CoM jerk \( \mathbf{u}_k \) and of future foot landing positions (footsteps), expressed in a local frame placed at the preceding foot;
- the system state \( \mathbf{c} \) as the concatenation of \( \mathbf{\hat{c}}_k \);
- the system output \( \mathbf{c}_G \) as the concatenation of \( \mathbf{z}_k \);
- the predicted external wrench \( \mathbf{\tilde{f}} \) as the concatenation of \( \mathbf{f}_{k+1} \), considered a perturbation or part of the control input, depending on the WPG design (see below).

Then, propagating (5) over the preview horizon yields:

\[
\begin{align*}
\mathbf{c} &= \mathbf{U}_x \mathbf{c}_0 + \mathbf{U}_u \mathbf{\tilde{u}} \\
\mathbf{c}_G &= \mathbf{O}_x \mathbf{c}_0 + \mathbf{O}_u \mathbf{\tilde{u}} + \mathbf{O}_f \mathbf{\tilde{f}},
\end{align*}
\]

where matrices \( \mathbf{U}_x, \mathbf{U}_u, \mathbf{O}_x, \mathbf{O}_u, \mathbf{O}_f \) can be obtained as in [32]. We then express the second equation of (6) in terms of the foot landing positions, as in [32]. This is done by expressing the global ZMP positions \( \mathbf{\tilde{z}}_G \) through the ZMP positions \( \mathbf{\tilde{z}} \) in local coordinates of the supporting foot such that:

\[
\mathbf{\tilde{z}}_G = \mathbf{V}_r \mathbf{\tilde{r}} + \mathbf{V} \mathbf{\tilde{r}} + \mathbf{R} \mathbf{\tilde{z}},
\]

where \( \mathbf{\tilde{r}} \) is a vector of future foot landing positions expressed in a local frame assigned to the preceding foot positions. The expression \( \mathbf{V}_r \mathbf{\tilde{r}} + \mathbf{V} \mathbf{\tilde{r}} \) produces positions of the feet in the global reference frame. Lastly, \( \mathbf{R} \) is a block diagonal matrix of the foot rotation matrices. This is also detailed in [32].

We can rewrite (6) and (7) as:

\[
\begin{align*}
\mathbf{c} &= \begin{bmatrix} \mathbf{U}_u & 0 \end{bmatrix} \begin{bmatrix} \mathbf{\tilde{u}} \\ \mathbf{\tilde{r}} \end{bmatrix} + \mathbf{U}_x \mathbf{c}_0 \\
\mathbf{\tilde{z}} &= \mathbf{R}^T \begin{bmatrix} \mathbf{O}_x & -\mathbf{V} \end{bmatrix} \begin{bmatrix} \mathbf{\tilde{u}} \\ \mathbf{\tilde{r}} \end{bmatrix} + \mathbf{R} \begin{bmatrix} \mathbf{O}_x \mathbf{c}_0 + \mathbf{O}_f \mathbf{\tilde{r}} - \mathbf{V}_r \mathbf{\tilde{r}} \end{bmatrix},
\end{align*}
\]

or more concisely as:

\[
\begin{align*}
\mathbf{c} &= \mathbf{S} \mathbf{x} + \mathbf{s} \\
\mathbf{\tilde{z}} &= \mathbf{S}_z \mathbf{x} + \mathbf{s}_z.
\end{align*}
\]

As argument of the optimization problem (2) we use \( \mathbf{x} \). Then, the objectives and constraints common to both WPG formulations are listed below.

- The CoM jerk is minimized to smoothen the trajectory. This is done via objective function:

\[
\mathbf{f}_{\text{com}}(\mathbf{x}) = \| \mathbf{\tilde{u}} \|_2^2 = \| \begin{bmatrix} \mathbf{I} & 0 \end{bmatrix} \mathbf{x} \|_2^2.
\]

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

\[
\mathbf{f}_{\text{zmp}}(\mathbf{x}) = \| \mathbf{\tilde{z}} \|_2^2 = \| \mathbf{S}_z \mathbf{x} + \mathbf{s}_z \|_2^2.
\]

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

\[
\begin{align*}
\mathbf{\tilde{z}} - \mathbf{S}_z \mathbf{x} &\leq \mathbf{z} \\
\mathbf{z} - \mathbf{S}_z \mathbf{x} &\leq \mathbf{\tilde{z}} - \mathbf{S}_z \mathbf{x},
\end{align*}
\]

- The feet positions should be constrained:

\[
\begin{align*}
\mathbf{\tilde{r}} - \mathbf{S}_z \mathbf{x} &\leq \begin{bmatrix} \mathbf{I} & 0 \end{bmatrix} \mathbf{x} - \mathbf{\tilde{r}}.
\end{align*}
\]

Note that objectives (10), (11) are expressed as (1), and constraints (13), (15) as (2b).

Using (9), we can now design various WPG, suited to different types of physical interaction. In particular, we revisit the leader and follower modalities [20]. Apart from the common constraints and objectives cited above, the leader and follower optimization problems are each characterized by a specific formulation of the objective function controlling the CoM. Yet, both walking pattern generators output the CoM jerk and footsteps over the preview horizon: \( \mathbf{\tilde{u}} \) and \( \mathbf{\tilde{r}} \). These are then

\footnote{Throughout this paper, we denote with \( \mathbf{z} \) and \( \mathbf{\pi} \) lower and upper bounds (respectively) of variable \( x \).}
processed, to obtain the desired CoM trajectory \(c_{\text{des}}(t)\) and the desired swing foot trajectory \(r_{\text{swdes}}(t)\) for moving between consecutive footsteps.

### 1) Walking pattern generator for a follower robot

A follower robot acts based on the leader’s intention. In our work, this is represented by the external wrench applied by the carrying partner. Usually, moving the object together implies only a planar motion such that the intention can be defined by \(f^{x,y}\). Hence, a follower WPG depends on these. Previous works \cite{20, 31} 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 \cite{42} of the follower, with diagonal matrices \(M, B, K\) containing the virtual mass, damping and stiffness parameters:

\[
\dot{e}^{x,y} = M\ddot{e}^{x,y} + B\dot{e}^{x,y} + Ke^{x,y}.
\]

Matrices \(M, B, K\) affect the robot response as that of a second order system. We do not add them to the optimization argument, as they can be tuned in simulation and from experiments, see also the interesting work in \cite{15}. Further- more, since these matrices are multiplied by the states in the preview horizon, that would lead to a nonlinear optimization problem, much more challenging for real time control. Using an impedance parameter matrix \(G_{mbk}\) and a selection matrix \(S_{f}\) (for choosing either \(f^{x}\) or \(f^{y}\)), this expression can be propagated, so that the MPC will aim at minimizing:

\[
f_{\text{fol}}(x) = \left\| G_{mbk}c - S_{f}\hat{\mathbf{f}} \right\|^2;
\]

Note that by injecting (9), we have expressed this objective as (1). The optimization problem, including objectives (10), (11), (17), and constraints (12), (14) is:

\[
\begin{align*}
\text{argmin}_{x} & \quad w_{\text{com}}f_{\text{com}}(x) + w_{\text{zmp}}f_{\text{zmp}}(x) + w_{\text{fol}}f_{\text{fol}}(x) \\
\text{subject to} & \quad \hat{\mathbf{z}} \leq \mathbf{z} \leq \bar{\mathbf{z}} \\
& \quad \hat{\mathbf{f}} \leq \mathbf{f} \leq \bar{\mathbf{f}}.
\end{align*}
\]

Notice that the future wrench values are required in \(\hat{\mathbf{f}}\), to make the robot proactive. Having a good model of human intention can be difficult, but if the force can be measured (e.g. by a force/torque sensor) at each iteration, we can use \(f_{\text{N}} = \cdots = f_{1} = f_{0}\), with \(f_{0}\) the current measure. This model is simplistic but it works well in our experiments (see Sec. VI); possibly because (i) collaborators intuitively avoid jerky motions (so interaction forces vary smoothly), and (ii) the current force is measured and fed back to the MPC at sufficiently high frequency (\(> 10\text{Hz}\)) for adapting properly and timely. Developing more refined interaction models would constitute a whole research theme per se, and will unlikely lead to any practical improvement.

The core part of the following WPG is impedance (17). Impedance control has been discussed in the literature several times since \cite{42}. Recently, \cite{44} discusses it in the context of collaborative carrying by mobile manipulators. The interested reader can refer to these papers for a more in depth discussion on impedance control. Here, it suffices to say that (17) imposes a relation between (interaction) force and (CoM) motion, which is a mechanical impedance. The novelty here is in using impedance control in an MPC framework together with balance constraints for walking (18). Although its fidelity (i.e., maintaining the imposed force-motion relation) has been shown in \cite{33}, it has inherent limits due to the balance constraint (which takes priority). The main limiting factor in (18) is the allowable instantaneous force change given the ZMP constraints. With some assumptions, this can be derived from the ZMP equation (3) as shown in \cite{45}. Then, if the change in applied force is below such allowable limit, the robot can adjust its posture to handle the sustained force as shown in the simulations of \cite{33} where up to 150N are applied to the robot.

### 2) Walking pattern generator for a leader robot

For leading, a clear intention is necessary. The robot should track a reference trajectory, known beforehand. For collaborative carrying, this can be generated by knowing where the object is and where it will be transported to. A classic way for tracking a trajectory in operational space \cite{46} is:

\[
\ddot{e}^{x,y} = \ddot{e}^{x,y}_{\text{ref}} + B(\ddot{e}^{x,y}_{\text{ref}} - \ddot{e}^{x,y}) + K(e^{x,y}_{\text{ref}} - e^{x,y}),
\]

where \(B\) and \(K\) 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:

\[
f_{\text{lea}}(x) = \left\| G_{bk}(\hat{\bar{e}}_{\text{ref}} - \hat{\bar{e}}) \right\|^2 = \left\| G_{bk}\hat{\bar{e}}_{\text{ref}} - G_{bk}\hat{\bar{e}} - G_{bk}s \right\|^2.
\]

Furthermore, with the robot acting as leader, the external wrench should be included in the optimization argument, expanding it as \(x = [\hat{\mathbf{u}}^T, \hat{\mathbf{f}}^T]^T\). The idea is that placing a part of the external wrench in the argument allows the robot to use the interaction to balance itself. However, for safety, \(\bar{\mathbf{f}}\) should be bounded and minimized:

\[
f_{\text{for}}(x) = \left\| \mathbf{f} \right\|^2 = \left\| [0, 0, 1]x \right\|^2.
\]

The optimization problem, including again objectives (10), (11), and constraints (12), (14), becomes:

\[
\begin{align*}
\text{argmin}_{x} & \quad w_{\text{com}}f_{\text{com}}(x) + w_{\text{zmp}}f_{\text{zmp}}(x) \\
& \quad + w_{\text{for}}f_{\text{for}}(x) + w_{\text{lea}}f_{\text{lea}}(x) \\
\text{subject to} & \quad \hat{\mathbf{z}} \leq \mathbf{z} \leq \bar{\mathbf{z}} \\
& \quad \hat{\mathbf{f}} \leq \mathbf{f} \leq \bar{\mathbf{f}}.
\end{align*}
\]

Since \(\bar{\mathbf{f}}\) is now part of the argument, objectives and constraints are still of the forms (1) and (2b), respectively.

### C. Feasibility and stability of the MPC

To conclude this section, we provide some insight into the feasibility of the QP problems (18) and (22). It has already been reported in \cite{35} that the nominal MPC for walking with fixed footstep positions and without external forces is always
feasible. It can be easily demonstrated that this property retains for the MPC formulations presented here. Perpetual feasibility, however, does not guarantee that the generated CoM motion does not diverge, leading to robot fall.

The standard approach to avoid divergence in MPC is to approximate an infinite preview horizon [47], for instance:

- It is possible to impose a terminal, so-called capturability, constraint to ensure that within a particular preview horizon the system can be stopped, e.g. [48]. Such constraint effectively prevents divergence, but may lead to infeasibility of the considered optimization problems. A detailed discussion of this topic can be found in [49].
- The second option is to use a “long enough” preview horizon, as justified in [50]. A bulk of previous works [11], [20], [32], [41], [51], [31] validated this approach in practice and reached a consensus on the length of the preview horizon, which should span 2 footsteps (the time horizon is less important than the number of footsteps in practical implementations).

We use the second approach for the sake of simplicity; it was proven to be sufficient. In future works, it may be interesting to study the possibility of using a terminal constraint to allow the robot to resist excessive force applied by the human. For example, while following the human using (18), the robot may be led to a fall. In this case, switching to (22) would allow the robot to regain balance by resisting the human.

V. Whole-body Control for Collaborative Carrying

The previous 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 $c_{des}(t)$ and $r_{wdes}(t)$ (respectively CoM and swing foot desired trajectory) must be mapped to robot joint commands, $q$. To explain how this is done, we start by recalling the optimization-based whole-body control framework developed in our research group [37]. Next, recurrent objectives and constraints are presented. Finally, we explain how all the components are assembled to realize the collaborative carrying.

A. Whole-body control as an optimization problem

To start detailing our whole-body control framework, we define the optimization argument in (2) as:

$$
\mathbf{x} = \begin{bmatrix} \dot{\mathbf{q}} \\ \lambda \end{bmatrix}.
$$

(23)

Here, $\mathbf{q}$ defines the robot configuration, i.e. the joint positions along with the floating-base representation, and $\lambda$ is the vector of linearized friction cone base weights. This is defined so that all contact forces $f_{con}$ stacked in a column vector yield:

$$
f_{con} = K_{fc} \lambda,
$$

(24)

with $K_{fc} \in \mathbb{R}^{3n \times nm}$ a matrix of generators for linearizing the friction cone ($n$ is the number of contact points, $m$ the number of generators for linearization).

For each state (i.e., subtask) of the FSM of Fig. 2, we solve the following optimization problem:

$$
\text{argmin}_{\mathbf{x}} \quad \sum_{j} w_{j} f_{\text{base},j}(\mathbf{x}) + \sum_{k} w_{k} f_{\text{spec},k}(\mathbf{x})
$$

subject to

$$
\begin{align*}
A_{\text{base}} \mathbf{x} & \leq b_{\text{base}} \\
A_{\text{spec}} \mathbf{x} & \leq b_{\text{spec}}.
\end{align*}
$$

(25)

In Section V-B, we present the objectives and constraints that are recurrent in the collaborative carrying FSM, specifically the tracking and set-point objectives, and the contact constraint. Then, the base objective functions ($f_{\text{base}}$), e.g. posture, and base constraints ($A_{\text{base}} \mathbf{x} \leq b_{\text{base}}$), e.g. joint limits, which are applied all along the experiments (and hence are common to all FSM states), are detailed in Sec. V-C. Instead, the collection of objectives ($f_{\text{spec}}$) specific to each FSM state, will be described, along with the specific constraints ($A_{\text{spec}} \mathbf{x} \leq b_{\text{spec}}$), in Sec. V-D.

In what follows, $w_{\xi}$ denotes the weight of task $\xi$ (i.e. the task function that it multiplies), subscripts $\text{pos}$ denote posture, $\text{con}$ is contact, $\text{des}$ is desired, $\text{spec}$ is specific, $\text{base}$ is base, $\text{sp}$ is set point, $\xi$ is CoM, $\text{sw}$ is swing foot, $\text{tr}$ is trajectory, $\text{gr}$ is grasp, $\text{o}$ is object, $\text{sup}$ is support. Others subscripts and variables with less occurrence are defined at their use.

B. Reusable objectives and constraints

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

$$
e = f_e(q).
$$

(26)

Assuming $f_e$ is twice differentiable, and naming $J_e$ the task Jacobian:

$$
\dot{e} = J_e \dot{q},
$$

(27)

$$
\ddot{e} = J_e \ddot{q} + \dot{J}_e \dot{q}.
$$

(28)

we define the tracking task objective as:

$$
f_{tr}(\mathbf{x}, e_{\text{des}}(t)) = \frac{1}{2} \|K(e_{\text{des}} - e) + B(\dot{e}_{\text{des}} - \dot{e}) + (\ddot{e}_{\text{des}} - \ddot{e})\|^2,
$$

(29)

where $e_{\text{des}}(t)$ denotes the desired task trajectory (i.e., it includes $e_{\text{des}}, \dot{e}_{\text{des}}$, and $\ddot{e}_{\text{des}}$), and $K$ and $B$ 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 $K$ needs to be tuned, with $B = 2\sqrt{K}$. Using (27) and (28), (29) can be written as (2a).

A particular case of the tracking task is the set-point objective, where only the reference position is considered, while the reference velocity and acceleration are set to zero:

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

(30)

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, by nullifying the acceleration of a robot point that is in contact with the environment:

$$\ddot{c} = 0.$$  \hfill (31)

Using (28), this can be written as the equality constraint:

$$[J_e \ 0] x = -\dot{J}_e \dot{q}. \hfill (32)$$

C. Base objectives and constraints

1) Base objective functions:
The first base objective function is termed the posture task, and represented as $f_{pos}$. This corresponds to positioning joints at a given posture $q_{des}$, with null $\dot{q}_{des}$ and $\ddot{q}_{des}$:

$$f_{pos}(x, q_{des}) = \frac{1}{2} \| K (q_{des} - q) - B \ddot{q} - \dot{q} \|^2, \hfill (33)$$

with $K$ and $B$ square diagonal gain matrices with positive values. Note that this is a typical example of set-point task (30), obtained with $e = q$. Exposing the joint accelerations via numerical integration at each time interval $k$ of duration $\Delta t$:

$$\ddot{q}_{k+1} = \ddot{q}_k + \dddot{q}_k \Delta t,$$

$$\dot{q}_{k+1} = \dot{q}_k + \dddot{q}_k \Delta t + \frac{1}{2} \ddddot{q}_k \Delta t^2, \hfill (34)$$

it is straightforward to show that objective (33) is of the form (2a). The goal of the posture task is to have a default configuration of each joint. Hence, its weight $w_{pos}$ normally has a low value, to give priority to more important tasks with higher weights, that will instead induce motion of selected joints. The second base objective consists in minimizing $\| \lambda \|^2$:

$$f_{x}(x) = \| \lambda \|^2 = \| [0 \ 1] x \|^2. \hfill (35)$$

As shown in (37), this objective function, joined with (33), allows an easier numeric solution to the QP problem.

2) Base constraints:

There are four constraints in the base formulation of our optimization problem (25), namely:

$$\lambda \geq 0$$

$$\tau \leq \tau \leq \tau \hfill (36a)$$

$$\dot{q} \leq q \leq \dot{q} \hfill (36b)$$

$$\ddot{q} \leq \dddot{q} \leq \dddot{q} \hfill (36c)$$

$\tau$ being the applied joint torques.

Firstly, (36a) ensures that the contact forces are inside the friction cone (no slipping). This can be formulated as:

$$[0 \ 1] x \geq 0. \hfill (37)$$

Second, (36b) places bounds on the torques $\tau$. These can be obtained from the robot dynamic equation:

$$\tau = H \ddot{q} + C \dot{q} + \tau_g - J^T_{con} f_{con}, \hfill (38)$$

with $H$ and $C$ respectively the inertia and Coriolis/centrifugal terms taking into account the floating-base [52], $\tau_g$ the torques due to gravity, $J_{con}$ the stacked contact point Jacobian matrices, and $f_{con}$ the stacked vector of contact forces from (24). The constraint can then be rewritten:

$$\tau - C \ddot{q} - \tau_g \leq [H - J^T_{con} K_{ec}] x \leq \tau - C \ddot{q} - \tau_g. \hfill (39)$$

The third and fourth constraints, (36c) and (36d), bound joint positions and velocities. With (34), these become:

$$\dot{q} - \dddot{q} \leq \frac{1}{2} [I \Delta t^2 - 0] x \leq \dddot{q} - \dddot{q}. \hfill (40a)$$

$$\ddot{q} - \dddot{q} \Delta t \leq \frac{1}{2} [I \Delta t^2 - 0] x \leq \dddot{q} - \dddot{q} \Delta t. \hfill (40b)$$

Stacking (37), (39), (40a) and (40b), yields the explicit expressions of $A_{base}$, and $b_{base}$ in (25).

D. Specific objectives and constraints of each FSM state

Here we detail the objectives and constraints specific to each FSM state. An important aspect concerns the control of the CoM. In walking FSM states (double support, right and single support), this is servoed using a tracking task objective (29) to follow the CoM trajectory output by the WPG (leader or follower).

That is, $c_{des}(t)$ and $r_{swdes}(t)$ are both generated by the WPG detailed in Sec. IV. For all other FSM states, we use a set-point objective (30) to attract the CoM towards the middle of the two feet by setting $c_{des}$ accordingly.

1) Double support:

During the double support state, both feet: $r_{left}, r_{right}$, must maintain contact with the ground, via contact constraints. The CoM is servoed with a trajectory $c_{des}(t)$, obtained from the standard WPG of Sec. IV.

In summary, the whole-body optimization problem is:

$$\arg\min_x \quad w_c f_{tr}(x, c_{des}(t)) + w_{pos} f_{pos}(x, q_{des}) + w_{\lambda} f_{\lambda}(x),$$

subject to $r_{left} = 0$, $r_{right} = 0$,

$$A_{base} x \leq b_{base}. \hfill (41)$$

2) Right/left single support:

While walking, single support states occur between two consecutive double support states. As such, they retain the CoM trajectory tracking task from the standard WPG. Differently from double support, only one foot supports the weight and is constrained to the ground, while the other (swinging) is servoed in the air to track $r_{swdes}(t)$ (any swing foot desired trajectory). Hence, the optimization problem is:

$$\arg\min_x \quad w_c f_{tr}(x, c_{des}(t)) + w_{sw} f_{tr}(x, r_{swdes}(t)) + w_{pos} f_{pos}(x, q_{des}) + w_{\lambda} f_{\lambda}(x),$$

subject to $r_{sup} = 0$,

$$A_{base} x \leq b_{base}. \hfill (42)$$

3) Pregrasping, squeeze and releasing:

The pregrasp, squeeze and release states have the same formulation, the only difference being their preceding state. Thus, without loss of generality, we only present the pregrasp. The pregrasp state is a waypoint state that eases the grasping by targeting a set of $n$ preplanned pregrasp point locations, $\{p_{grdes, 1} \cdots p_{grdes, n}\}$. The synthesis of these locations can be
formalized either as a stance generation problem [53], or by considering caging [54]. Here, 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 Sec. II). For instance, we design the body grasps shown in Fig. 3 for the pipe-shoulder and pipe-front examples of Fig. 1. In those cases, we parametrize the grasp via the contact frames shown in Fig. 3. More generally, we define \( n \) operational frames on the robot body. The pose of each one, denoted by: \( p_{\text{gr},i} \) (\( i = 1 \ldots n \)), should be servoed to a desired pose: \( p_{\text{grdes},i} \). This corresponds to \( n \) set point objectives (30). Note, from Fig. 2, 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, squeeze, and release optimization problems can all be formulated as:

\[
\arg\min_{x} \sum_{i=1}^{n} w_{\text{gr},i} f_{\text{sp}}(x, p_{\text{grdes},i}) + w_{\text{pos},i} f_{\text{pos}}(x, p_{\text{des}}) + w_{\lambda} f_{\lambda}(x), \\
\text{subject to } \ddot{\mathbf{p}}_{\text{left}} = 0, \\
\ddot{\mathbf{p}}_{\text{right}} = 0, \\
\mathbf{A}_{\text{base}} x \leq \mathbf{b}_{\text{base}}. 
\]  

(43)

4) Holding the object while lifting, carrying, placing it:

After having successfully squeezed 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, if the closed kinematic chains are not handled properly, this results in numerical issues for the solver. 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 [54] where the object being caged moves along 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 \( f_{\text{sp}}(x, p_{\text{des}}) \). We assume here that \( p_{\text{des}} \) is provided beforehand, for example by a high-level plan (as for the grasp points in Sec. V-D3). For collaborative carrying, the hold state is to be realized while lifting, walking, and placing (see Fig. 2). In the following paragraphs, we present the corresponding optimization problems. In each case, we highlight the components (either in the cost function or constraints) that vary from one problem to the other.

The optimization problem is:

\[
\arg\min_{x} \sum_{i=1}^{n} w_{\text{gr},i} f_{\text{sp}}(x, p_{\text{grdes},i}) + \sum_{i=1}^{n} w_{\text{pos},i} f_{\text{pos}}(x, p_{\text{des}}) + w_{\lambda} f_{\lambda}(x), \\
\text{subject to } \ddot{\mathbf{p}}_{\text{left}} = 0, \\
\ddot{\mathbf{p}}_{\text{right}} = 0, \\
\mathbf{A}_{\text{base}} x \leq \mathbf{b}_{\text{base}}. 
\]

(44)

For holding during double support, the optimization problem is similar, except that \( w_{\text{c}} f_{\text{sp}}(x, p_{\text{des}}) \) is replaced by \( w_{\text{c}} f_{\text{fr}}(x, p_{\text{des}}(t)) \), with \( p_{\text{des}}(t) \) output by either the follower or leader WPG. Therefore, the optimization problem is:

\[
\arg\min_{x} \sum_{i=1}^{n} w_{\text{gr},i} f_{\text{sp}}(x, p_{\text{grdes},i}) + \sum_{i=1}^{n} w_{\text{pos},i} f_{\text{pos}}(x, p_{\text{des}}) + w_{\lambda} f_{\lambda}(x), \\
\text{subject to } \ddot{\mathbf{p}}_{\text{left}} = 0, \\
\ddot{\mathbf{p}}_{\text{right}} = 0, \\
\mathbf{A}_{\text{base}} x \leq \mathbf{b}_{\text{base}}. 
\]

(45)

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

\[
\arg\min_{x} \sum_{i=1}^{n} w_{\text{gr},i} f_{\text{sp}}(x, p_{\text{grdes},i}) + \sum_{i=1}^{n} w_{\text{pos},i} f_{\text{pos}}(x, p_{\text{des}}) + w_{\lambda} f_{\lambda}(x), \\
\text{subject to } \ddot{\mathbf{p}}_{\text{left}} = 0, \\
\ddot{\mathbf{p}}_{\text{right}} = 0, \\
\mathbf{A}_{\text{base}} x \leq \mathbf{b}_{\text{base}}. 
\]

(46)

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

E. Note on feasibility

As indicated in Sec. III, conflicts may arise between the QP constraints. Although constraint relaxation is a viable strategy to recover from infeasibility, it may result in control inputs which are either physically inconsistent, or not executable by the hardware. Other strategies are detailed below.

Object handling motions –pregrasping, squeezing, and releasing – are defined with the help of a contact stance planner as in [53]. Since this planner guarantees feasibility only at specific postures, we have to verify, through simulations, that the interpolated motions are also feasible. For example,
the second scenario from below in Fig. 1 turned out to be unfeasible on our humanoid, because of the body grasp configuration.

Walking may not be feasible due to discrepancies between the reduced model employed in the WPG and the whole body model, namely because of these WPG assumptions:
(a) absence of kinematic and joint limits;
(b) zero rate of angular momentum;
(c) constant vertical component of the external force.

Issues of such kind are traditionally addressed with proxy constraints, which reflect limitations of the reduced model [55].

All the constraints in the WPG can be seen as proxies:
- bounds on the ZMP positions are chosen depending on the size of the feet, while safety margins in these constraints implicitly account for (b);
- feasible regions for the landing foot positions address kinematic limits in (a) and can be estimated using simulations as in [32];
- bounds on the external wrench reflect dynamic limits in (a) and can also be chosen empirically in simulations.

Although it is also possible to avoid assumption (c) as in [56], or to address (b) using a reduced model from [40], the three proxy constraints presented above were sufficient in this work.

VI. 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. The robot is position controlled, with joint set points updated every 5ms by the whole body controller described in Sec. V.

In all experiments, for the walk, we set the swing duration to \( t_{swh} = 0.7s \), and the stepping height to \( t_{stb} = 0.07m \). The average forward walking velocity, in the leader CoM trajectory (19), is set to 0.1m.s\(^{-1}\). All simulations are run with the same general parameters and timing constraints as on the real robot, using a 2.7 GHz i7 processor. The source code that we used in all these experiments is public and available\(^4\). The main components of this code are dynamic and continuously evolving. Yet, the latest versions of the QP controller and of the walking pattern generator are also available online\(^5,6\).

A. Simulations

The base functionality of the WPG accounting for external wrench was previously verified and tested, with the results presented in [33]. Complementary 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 (3) shows that an important negative \( f^z \) will increase the robustness to external wrenches in \( x \) and \( y \) directions, by reducing their net effect on the ZMP. Furthermore, if \( f^z \) is comparable to the robot weight, it reduces the acceleration effects. An intuitive way to interpret this is that the added weight lowers the CoM of the combined (robot and object) system. Hence, carrying heavier objects actually helps the humanoid stability (assuming the robot motors can handle the extra load).

Concerning whole body control, we present simulations on the designed pregrasp and squeezing postures output by (43). 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. 3. However, due to hardware issues (broken wrist joint), we also had to design one-handed versions of these, shown in Fig. 5, 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 squeezing (middle right in Fig. 5). This is caused by objective function \( f_{sp}(\mathbf{x}, c_{des}) \) in (43), 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 the WPG (be it follower (18) or leader (22)) provides a reference CoM position, velocity and acceleration. In Fig. 6, 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

---

\(^4\)https://gite.lirmm.fr/multi-contact/hri_scenario
\(^5\)https://github.com/jrl-umi3218/Tasks
\(^6\)https://github.com/bip-team/humoto-module-wpg03-collaboration
the robot is actually walking at 0.1 m s⁻¹, as requested. As for the ZMP, the approximation of null angular momentum rate leads to the slight tracking error visible in the figure. Nevertheless, our tests (both dynamic simulations and real experiments) show that this ZMP tracking error does not affect robot balance.

![CoM and ZMP tracking](image)

Fig. 6. Tracking task of the CoM using the WPG-generated reference CoM along with a comparison of the resulting ZMP.

Lastly, we present simulations of walking as leader, while holding (Sec. V-D4). Image sequences of walking while holding using two-handed front-wrap, and shoulder-mounted body grasps, are respectively shown in Figures 7 and 8. 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, while maintaining the desired robot posture.

![Fig. 7. Walking while holding, using a front-wrap body grasp.](image)

![Fig. 8. Walking while holding, using a shoulder-mounted body grasp.](image)

B. 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 available at: https://youtu.be/lHG4AbAvt4.

Screenshots of collaborative carrying experiments are shown in Fig. 9. This 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 collaborative carrying scenarios were successful, with the robot acting as both leader and follower. For the two missing examples: first, table carrying (first scenario in Fig. 1) was validated in our previous work [20], [31]; second, the example requiring a side body grasp (fifth scenario in Fig. 1) is kinematically infeasible for HRP-4, as stated in Sec. V-E.

![Fig. 9. Experiments with HRP-4 carrying various objects with a human. Top left: 15 N carton box. Top right: 15 N carton box (for this experiment please refer to time 0:23 of the attached video; the robot starts in a half-sitting posture with its right arm preparing to hold the box; while the robot grasps the box, the left arm motion, which was not programmed ad hoc, results from the weight of the box and from the CoM/ZMP equilibrium tasks). Bottom left: 13 N stretcher. Bottom right: 8 N bucket.](image)

Relevant data from the stretcher carrying task, with the robot walking as follower (bottom left in Fig. 9), are shown in Fig. 10. The top plot shows the CoM and ZMP reference signals, generated by the WPG (18), together with the measured values. A significant difference, due to impact, is only observed on contact transitions (footsteps). Meanwhile, the bottom figure shows the forward (pulling) component of the interaction force, measured by the two wrist force sensors, and then low-pass filtered and transformed to the CoM frame, $f_x$. Although the whole wrench is considered by the WPG, we only show this component, as it is the most relevant for this experiment. Note the pause in the walk (top figure), around the 13 to 15 second mark, corresponding to a strong decrease of interaction force (since the human stopped). Throughout the experiment, the CoM and ZMP reference values are properly adapted to the external force, as the robot follows the human.

Furthermore, Fig. 11 shows the low-pass filtered force/torque sensor data in the corresponding force sensor frames. In such frames, the $x$ component roughly corresponds to the object weight, showing the robot is carrying around...
10 N per hand throughout the experiment. According to our prior calibration data, each hand weighs 7 N. Therefore, the robot is supporting a total of 6 N of the stretcher weight which is about 13 N, hence it is approximately sharing the load with the human. The y component roughly corresponds to the previously discussed interaction force in Fig. 10. Finally, the z component coincides with the grasping forces applied on the stretcher in between both hands. This remains around 5 N throughout the test.

These results show that the overall approach works well, although force sensing is available only at the robot wrists, and not at the other contact points (e.g., on the shoulder and chest). The grasp stability could be improved, if force/contact sensing was available on other body parts.

C. Remarks on leader-follower role switching

Our framework supports a leader and a follower control setting, but this does not imply that the role of each partner is predefined once and for all by a binary rule [24]. We have already experimented table transportation with role switching and a proactive behavior based on force and visual sensing [20], [31]. Leader/follower switching was devised from human-human learned patterns. Role distribution can span on each subspace of the task by means of our QP controller. Concerning the strategy on how and when the roles are distributed, any approach can be plugged on top of our controller.

The simplest role distribution can be done through smooth sequencing during walking. The switch in the role can be triggered by any condition: those given by the mission of the robot or those that are inherent to robot’s integrity violation (e.g., losing balance, workspace limitations, etc.). In a simple experiment, we show that we can switch between the two WPG formulations (18) and (22) instantly at will. The robot starts out as a follower. After around 50 s, we switch the roles, and the robot becomes leader. It then tracks a predefined trajectory that consists of going backwards 0.2 m in 10 s.

The Figure 12 shows generated WPG references and interaction forces (top) along with photos of the experiment (bottom). Notice that, even in the presence of sensor noise, the robot can follow the intent of the human leader. During the leading phase, note that the generated reference force is quite low (gain tuning here was more coarse than in the simulations). This allows the robot to balance itself, without relying on force control.

Another option is to enable equal partnerships with adaptive or versatile priority, eventually a continuous blending as in the seminal work [24]. Although we didn’t try experimentally this option, our framework extends to this possibility by combining (18) and (22) into:

\[
\begin{align*}
\arg\min_x & \quad w_{com}f_{com}(x) + w_{zmp}f_{zmp}(x) + w_{for}f_{for}(x) \\
& \quad + \mathcal{W}[f_{rea}(x), f_{for}(x)] \\
\text{subject to} & \quad \underline{z} \leq \tilde{z} \leq \bar{z} \\
& \quad \underline{f} \leq \tilde{f} \leq \bar{f} \\
& \quad \underline{f} \leq \tilde{f} \leq \bar{f}.
\end{align*}
\]

In (47), \(\mathcal{W}\) is a function that outputs the result of any implemented strategy (among those cited in the introduction) to assign role distribution or sharing. However, if these strategies apply well in joint manipulations in the object’s task-space...
sharing, it is unlikely that they bring any added-value in the walking process that is a hybrid process which is not a task that is ‘shared’. In (47), the constraints on forces can be dropped; this is also the case in (22) as their main purpose is to secure the human partner.

VII. CONCLUSION

This article explores several aspects of human-humanoid collaborative carrying. We started by looking at this task as a whole, to infer the core principles, in order to program them on a humanoid robot. To this end, we created a 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 them accordingly, to operate as a follower or leader. Then, we discussed how all of this can be designed as objectives and constraints of an optimization problem for a whole-body controller. We finally presented simulations and real test cases on the HRP-4 humanoid.

Although our approach 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 [57]. Another key improvement 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 human perception for intention recognition, a difficult challenge, but also an active research area in physical human-robot interaction. Methods combining machine learning and dynamical systems, e.g. [58] could be investigated as they could also encompass the planning part.

Concerning the walk, although the WPG presented here is simplified, its core concepts do not conflict with improvements such as those in [59] which add robustness, allowing stair climbing. Another possible direction for future investigation in WPG design is addition of the terminal capturability constraint as indicated in Sec. IV-C. Apart from improving the WPG itself, its integration in whole-body control can also be improved, with works such as [48] which aims at combining the separate QPs.

Lastly, a limiting factor in 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 [60]. To solve this, a dedicated stabilizer, consistent with our framework, should be designed. Another challenging topic of on-going work consists in applying the multi-robot version of our QP controller [61] to include the human as an additional ‘robot’. However, going beyond simulation [62] requires estimation of the human’s full or partial pose and inertia parameters, to effectively close the overall control loop in a pHRI setting. Finally, once the system is improved in terms of performance, user-related studies, with several different users (possibly also instrumented) is also worthy of investigation.

REFERENCES


Andrea Cherubini received the M.Sc. in Mechanical Engineering in 2001 from the University of Rome La Sapienza and a second M.Sc. in Control Systems in 2003 from the University of Sheffield, U.K. In 2008, he received the Ph.D. in Control Systems from the University of Rome La Sapienza. From 2008 to 2011, he was postdoc at Inria Rennes, France. Andrea has co-authored over 60 papers in International peer-reviewed Conferences and Journals and is currently Associate Professor in Robotics at University of Montpellier, France.

Alexander Sherikov received the Specialist degree in computer science from Petrozavodsk State University, Petrozavodsk, Russian Federation, in 2008, the M.S. degree in robotics from the Örebro University, Örebro, Sweden, in 2012, and the Ph.D. degree in automatic control from Grenoble University, Grenoble, France, in 2016. From 2016 to 2017, he was a Research Engineer at Inria Grenoble. Since 2017 he has been a Software Engineer with PAL Robotics, Barcelona, Spain.

Pierre-Brice Wieber graduated from École Polytechnique in 1996 and received his PhD degree in Robotics from École des Mines de Paris in 2000. He has been at INRIA Grenoble Rhône-Alpes since 2001, except when visiting the CNRS-AIST Joint Robotic Laboratory, UMI3218/RL as an EU FP7 Marie Curie Fellow (2008–2010). He served as Associate Editor for IEEE Transactions on Robotics (2014-2018). His research interests include the modeling and control of humanoid and manipulator robots.

Abderrahmane Kheddar (SM’12) received a BSCS degree from the Institut National d’Informatique (ESI), Algiers, the MSc and PhD degrees in robotics, both from the University of Pierre and Marie Curie, Paris 6. He is presently Director of Research at the CNRS working at the CNRS-AIST Joint Robotic Laboratory, Tsukuba, Japan; and the Interactive Digital Humans team at CNRS-UM LIRMM at Montpellier, France. His research interests include humanoid robotics at large with a focus on contact and haptics. He is a founding member of the IEEE Transactions on Haptics and served on its editorial board (2007-2010); served as associate editor and then editor in the IEEE Transactions on Robotics (2010-2018) and is in the editorial board of other robotic journals. He is a member of the National Academy of Technologies of France, knight in the National Order of Merit of France, and a Senior Member of the IEEE Society.