Adaptive Whole-Body Manipulation in Human-to-Humanoid Multi-Contact Motion Retargeting

Kazuya Otani, Karim Bouyarmane

Adaptive Whole-Body Manipulation in Human-to-Humanoid Multi-Contact Motion Retargeting

Kazuya Otani Karim Bouyarmane

Abstract— **We propose a multi-robot quadratic program (QP) controller for retargeting of a human's multi-contact loco-manipulation motions to a humanoid robot. Using this framework, the robot can track complex motions and automatically adapt to objects in the environment that have different physical properties from those that were used to provide the human's reference motion. The whole-body multi-contact manipulation problem is formulated as a multi-robot QP, which optimizes over the combined dynamics of the robot and any manipulated objects. The multi-robot QP maintains a dynamic partition of the robot's tracking links into fixed support contact, manipulation contact, and contact-free tracking links, which are re-partitioned and re-instantiated as constraints in the multi-robot QP every time a contact event occurs in the human motion. We present various experiments (bag retrieval, door opening, box lifting) using human motion data from an Xsens inertial motion capture system. We show in full-body dynamics simulation that the robot is able to perform difficult single-stance motions as well as multi-contact-stance motions (including hand supports), while adapting to objects of varying inertial properties.**

I. Introduction

Recent developments in human motion tracking technologies allow high-fidelity, high-frequency tracking of human pose. The data from these systems can be used to generate human-like motions on humanoid robots.

Human-to-humanoid motion retargeting can be applied to teleoperation in disaster scenarios, where a robot may be better equipped to enter hazardous environments, as well as learning from demonstration, in which a robot learns a generalizable representation of a task from human demonstration. In particular, we believe that motion retargeting is a powerful tool for bootstrapping robot motion planning in complex multi-contact motions. Other examples of tasks in this category are the vehicle egress task in the DARPA Robotics Challenge, and walking through cluttered manufacturing environments.

However, direct mapping of motion from humans to humanoid robots is not possible, due to differences in kinematics (e.g. joint limits, limb lengths) and dynamics (e.g. mass distribution). A motion retargeting method must generate a robot motion that is feasible and stable, while also following the salient parts of the human demonstration as closely as possible.

This work is concerned with retargeting of multi-contact whole-body locomotion and manipulation motions. This presents additional challenges, such as changes in contact supports and changes in dynamics due to the manipulated objects.

Our approach differs from existing work in human-to-humanoid motion retargeting where manipulated objects, if any, are light relative to the robot's own weight. Typically, the manipulated object's dynamics are represented as a known external force, or simply as a disturbance to the control system. We use a multi-robot quadratic program (QP) framework, which explicitly models the combined dynamics of the robot and manipulated object. This allows the robot to manipulate heavy objects that significantly affect its dynamics. It can also generalize human demonstrations performed on an object of specific properties to situations with objects of different physical properties (e.g. mass, friction, spring constant). Other desired behaviors, such as collision avoidance between bulky or cumbersome manipulated objects and the robot/environment, can be enforced cleanly within this formulation. Using this framework, we extend the range of activities possible in human-to-humanoid motion retargeting.

To demonstrate the capabilities of our method, we show a simulated robot imitating human demonstrations in several whole-body loco-manipulation tasks. We show that the robot can adapt its movement to account for objects of varying properties.

The rest of this paper is organized as follows. Section II covers prior work on motion retargeting and humanoid control. In Section III we summarize multi-contact motion retargeting with a QP controller, as introduced in [1]. In Section IV we introduce additionally the new multi-robot QP formulation for multi-contact motion retargeting. It is experimentally applied in Section V, where various whole-body manipulation tasks are retargeted from a human to humanoid robots in full-body dynamics simulation. Finally, Sections VI and VII conclude the paper by discussing the limitations of our method and indicating directions for future work.

II. Related Work

Human-to-humanoid motion retargeting has been an active topic of research for the past two decades. Pollard et al [2] showed one of the first approaches to motion retargeting of human movements for humanoids. They applied inverse kinematics and edited the human reference movement to keep the robot within its joint angle and velocity limits. They only imitated upper body movements of a humanoid, and hence did not consider balance. Darius et al [3] formulated upper body imitation as task space control, and had a separate ZMP-based controller that ensures balance. Abe et al [4]

e-mail: kazuya.otani@inria.fr, karim.bouyarmane@loria.fr
The authors have the following affiliations:
- Inria, Villers-lès-Nancy, F-54600, France
- CNRS, Loria, UMR 7503, Vandœvre-lès-Nancy, F-54500, France
- Université de Lorraine, Loria, UMR 7503, Vandœuvre-lès-Nancy, F-54500, France

proposed a formulation for tracking manipulation motions where the external forces are seen as perturbations to the controller, and where the friction and balance constraints are not accounted for. Nakaoka et al [5] demonstrated a humanoid robot that was able to imitate human dances including lower-body motions by using a motion primitive-based approach. Yamane et al [6] introduced the concept of a "dynamics filter", which calculates the center of pressure based on a simplified model and modifies the center of mass trajectory if necessary to keep the robot balanced. They make the assumption that the robot is on flat ground and that only the feet will make contact with the environment. More recently, Koenemann et al [7] showed a real-time teleoperation method that was based on a simplified human model. Their method could handle transitions between double and single stances phases. They also showed manipulation of light objects.

QP-based humanoid controllers have become ubiquitous in recent years, due to their robustness, speed, and flexibility. QP controllers allow intuitive encodings of constraints and objectives in an optimization problem that can be solved at real-time control rates. Early QP control work of Abe et al [8] and De Lasa et al [9] demonstrated multi-objective control for generating natural movements in computer animated characters. In the recent DARPA Robotics Challenge, a majority of the teams relied on a QP for their low-level full-body controller [10][11].

This work builds on the work by Di Fava et al [1], who used a QP controller [12][13] to track human motion with Cartesian space "tasks". Because the QP reasons explicitly about contact friction cones, they were able to deal with multiple non-coplanar contacts. Their method ran in real-time for motions without contact changes, but it required an offline processing step for motions that had contact events.

Recently, there have been developments in incorporating external objects and other robots into a single QP formulation, which allows explicit optimization over physically and behaviorally coupled systems. Vaillant et al [14] introduced this for multi-character control in animation, and Bouyarmane et al [15] implemented this on multi-robot motions.

III. Background : Multi-contact motion capture tracking QP

In [1], human-to-humanoid multi-contact motion retargeting with contact changes is addressed in a two-stage approach. In the first stage, the sequence of contact change events

$$L = (\eta_1, \dots, \eta_\mu)\,, \tag{1}$$

is extracted from offline processing of the recorded motion. A contact change event is encoded into the tuple $\eta_j = (l_j, s_j, t_j, \sigma_j)$ with l_j the robot link index, s_j the environment surface index, t_j the timing of the contact change event, and $\sigma_j \in \{0, 1\}$ a binary variable encoding the event type (0 for a contact breaking event, 1 for a contact making event). In the second stage of the approach, the recorded motion is tracked online by a QP controller, by

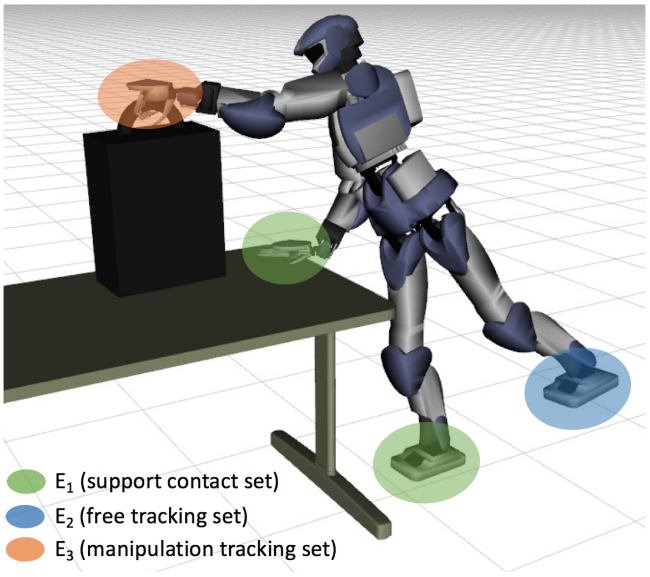

Fig. 1. Illustration of the contacts/tracking sets partition at a given phase of the bag retrieval motion. E_1 is the fixed contact supports set, which includes the right foot and the right hand. E_2 is the free tracking set, with only the left foot in this situation. E_3 is the manipulation tracking set, which includes the left hand holding the handle of the bag. At each contact event (addition or removal), the three sets (E_1, E_2, E_3) are updated.

decomposing the links of the human/robot in two sets. E_1 is the *contacts set*, the set of links of the human/robot that are in contact with the environment. E_2 is the *active tracking set*, the set of links at the extremities the human/humanoid robot's limbs that are freely moving and could potentially make contact with external objects during the motion (we might refer to these as *end-effectors*). E_2^0 is the *total tracking set*, a constant set that contains all the links at the extremities of the limbs that are tracked when they are not in contact, $E_2 \subset E_2^0$ (note that links not belonging to E_2^0 can be in E_1, e.g. when sitting on the buttocks). The sets E_1 and E_2 are separate, i.e. $E_1 \cap E_2 = \emptyset$, and more precisely $E_2 = E_2^0 \setminus E_1$. For a given contacts/tracking sets distribution (E_1, E_2), the QP that is instantiated for the robot to track the human motion is the following (we highlight the dependency on $(\boldsymbol{E_1}, \boldsymbol{E_2})$):

$$\begin{aligned}
\min_{\ddot{q},\tau,f_{\boldsymbol{E_1}}} \quad & \sum_{k\in\boldsymbol{E_2}} w_k ||\ddot{x}_k - \ddot{x}_k^d(x_k^{\text{hum}})||^2 \\
& + w_c \left|\left|\ddot{c} - \ddot{c}^d\big(h(\boldsymbol{E_1})\big)\right|\right|^2 \qquad (2)\\
& + w_0 ||\ddot{q} - \ddot{q}^d(q_0)||^2\,, \\
\text{subject to} \quad & M\ddot{q} + N = J_{\boldsymbol{E_1}}^T f_{\boldsymbol{E_1}} + S^T\tau\,, \qquad (3)\\
& J_{\boldsymbol{E_1}}\ddot{q} + \dot{J}_{\boldsymbol{E_1}}\dot{q} = 0\,, \qquad (4)\\
& f_{\boldsymbol{E_1}} \in \mathcal{C}_{\boldsymbol{E_1}}\,, \qquad (5)\\
& \text{torque limits, joint limits, collision avoidance,} \qquad (6)
\end{aligned}$$

where q is the configuration of the robot, M is the mass matrix of the robot, N the nonlinear effects and gravity term, τ the actuation torques at the joints of the robot, f_{E_1} the set of point contact forces applied at the contact links in E_1, J_{E_1} the contact point Jacobians of the links in E_1, S the all-

DoFs-to-actuated-DoFs selection matrix, $\mathcal{C}_{E_1}$ the Cartesian product of the linearized friction cones at the contact points in E_1.

The constraints for torque limits, joint angle/speed limits, and collision avoidance (6) are described in [1], and are excluded here to emphasize the other equations which are dependent on the tracking/contact sets partition (E_1, E_2). The torque limits are formulated as straightforward inequalities. Joint limits and collision avoidance constraints are formulated as velocity dampers on accelerations of joints $\ddot{q}$ and distance functions $\ddot{d}$.

In the cost function (2), the symbol $g \in \{\text{`}x_k\text{'}, \text{`}c\text{'}, \text{`}q\text{'}\}$ denotes objectives, or "tasks": an end-effector pose (position, orientation) task, a horizontal projection of the center-of-mass (CoM) task, and a full-configuration task, respectively, with corresponding weights w_g and stiffness κ_g. A task is specified by its desired acceleration $\ddot{g}^d(g^{\text{ref}})$ that tracks a reference value g^{ref} as:

$$\ddot{g}^d(g^{\text{ref}}) = \ddot{g}^{\text{ref}} - \kappa_g(\dot{g} - \dot{g}^{\text{ref}}) - 2\sqrt{\kappa_g}(g - g^{\text{ref}})\,. \quad (7)$$

The first component of (2) is a task on end-effector pose x_k. The reference value x_k^{ref} for $k \in E_2$ is set at the corresponding link marker pose from the human motion capture data x_k^{hum}:

$$\forall k \in E_2 \quad x_k^{\text{ref}} = x_k^{\text{hum}}\,. \quad (8)$$

The second component is a task on the CoM position. The reference value for the horizontal components of the CoM is set at $h(E_1)$, which denotes the center of the projected horizontal support polygon for the set of contacts in E_1 (*i.e.* the center of the convex hull of the ground projections of the contact points in E_1):

$$c^{\text{ref}} = h(E_1)\,. \quad (9)$$

The vertical component of c^{ref} is set to track the human motion. Because maintaining balance is crucial to safe operation of the robot, the CoM task is given a high weight relative to the other tasks.

The last component is a full-configuration task, whose reference is set to a rest posture q_0 with a low weight. This task ensures that the optimization is well-conditioned. In practice, it also helps generate more natural-looking movements.

The multi-contact motion retargeting algorithm uses the QP controller to track the human's motion, while reformulating the objectives and constraints of the QP every time a contact change event (i.e. a re-partitioning of (E_1, E_2)) is reached. For contact additions, a *contact approach and stabilization* phase is inserted in a fixed time window in order to ensure stable contacts.

IV. Adaptive manipulation multi-contact motion retargeting

In this work, we incorporate the dynamics of manipulated objects into the motion capture QP framework reviewed in Section III. In doing so, we make the following assumptions. First, we assume that any manipulated object's inertial and friction parameters are known to the robot. Second, it is assumed that the robot can identify relevant contact surfaces on the objects. These assumptions are reasonable if: the robot is operating in a known environment, a human operator can provide this information, or the robot has a perception system that can estimate these parameters. Given these assumptions, we can model the whole-body manipulation task in a multi-robot QP.

A. Motion retargeting with manipulation as multi-robot QP

The main idea is to retarget environment manipulation motions performed by the human, to the humanoid robot manipulating its own environment. The environment of the robot may have different inertial properties from the environment being manipulated in the human motion capture data. The simplest example of such a situation is the human lifting a box of a certain mass in their demonstration, and the robot performing the same motion on boxes of different masses (demonstrated in Section V). To achieve this, we integrate the multi-robot QP into the multi-contact motion retargeting framework.

Suppose that the robot is manipulating an environment object, which may be a free-floating object (e.g. box), a fixed-base articulated mechanism (e.g. door, drawer), with passive joints, spring-loaded joints (knob of a door), or actuated joints (a second robot). As an abuse of language, we call all these objects "robots" (hence the "multi-robot" terminology). The general dynamical equation of motion of the environment object takes the form

$$M_o\ddot{q}_o + N_o = J_o^T f_o + S_o^T \tau_o\,, \quad (10)$$

where τ_o are actuation torques. q_o is the configuration of the environment object (e.g. the 6D position/orientation of a box, or the configuration of an articulated object plus its 6D base position/orientation if the base is not fixed). f_o are all the external forces applied on the object. These include forces applied by the external fixed environment ($f_{e,o}$) and the forces applied by the robot ($f_{r,o}$) : $f_o = (f_{e,o}, f_{r,o})$. Decomposing f_o, Equation (10) becomes

$$M_o\ddot{q}_o + N_o = J_{e,o}^T f_{e,o} + J_{r,o}^T f_{r,o} + S_o^T \tau_o\,. \quad (11)$$

Extending the framework described in Section III, we redefine E_1 as the *support contacts set*, the set of links of the robot that are in contact with the *fixed* environment. E_2 is now defined as the *free tracking set*, the set of limb extremities ("end-effectors") that are freely moving without any contact. We introduce a new set E_3 as the *manipulation tracking set*, the set of links of the robot that are in contact with the manipulated environment object. We denote f_{E_1} as the forces applied by the fixed environment on the robot, and f_{E_3} as the set of forces applied by the manipulated object on the robot. The robot's equation of motion (3) becomes:

$$M\ddot{q} + N = J_{E_1}^T f_{E_1} + J_{E_3}^T f_{E_3} + S^T\tau \quad (12)$$

According to Newton's third law, the force applied by the robot on the object must be equal and opposite to the force applied by the object on the robot. Thus we have

$$f_{E_3} = -f_{r,o}\,. \quad (13)$$

The minimal set of forces of the problem are thus:

$$f_{\text{multi-robot}} = (f_e, f_{E_3}), \tag{14}$$

where $f_e = (f_{E_1}, f_{e,o})$ denotes the external forces applied by the fixed environment on the robot and the object, and f_{E_3} the internal (to the combined system) forces exchanged between the robot and the object. We ensure that surfaces in contact move together with:

$$J_{E_3}\ddot{q} + \dot{J}_{E_3}\dot{q} = J_{r,o}\ddot{q}_o + \dot{J}_{r,o}\dot{q}_o, \tag{15}$$

which complements both (4) and the following equation:

$$J_{e,o}\ddot{q}_o + \dot{J}_{e,o}\dot{q}_o = 0\,. \tag{16}$$

Let us now rewrite the motion capture tracking QP (2) as a multi-robot QP for manipulating the environment object. The multi-robot QP now tracks the links that are in E_2 (free tracking set) and in E_3 (manipulation tracking set). We define the multi-robot quantities by stacking the dynamics equations, as in [14]:

$$\begin{aligned}
q_{\text{multi-robot}} &= (q, q_o)\,, && (17)\\
M_{\text{multi-robot}} &= \text{blockdiag}(M, M_o)\,, && (18)\\
N_{\text{multi-robot}} &= (N, N_o)\,, && (19)\\
S_{\text{multi-robot}} &= \text{blockdiag}(S, S_o)\,, && (20)\\
\tau_{\text{multi-robot}} &= (\tau, \tau_o)\,, && (21)\\
J_e &= \text{blockdiag}(J_{E_1}, J_{e,o})\,, && (22)
\end{aligned}$$

The new motion capture tracking multi-robot QP becomes (with subscript '$multi-robot$' shortened to 'mr', and with dependency on set partition $(\boldsymbol{E_1}, \boldsymbol{E_2}, \boldsymbol{E_3})$):

$$\begin{aligned}
\min_{\ddot{q}_{\text{mr}}, \tau_{\text{mr}}, f_{\text{mr}}} \quad & \sum_{k \in \boldsymbol{E_2} \cup \boldsymbol{E_3}} w_k ||\ddot{x}_k - \ddot{x}_k^d(x_k^{\text{hum}})||^2 \\
& + w_c \left|\left|\ddot{c}_{\text{mr}} - \ddot{c}_{\text{mr}}^d\big(h(\boldsymbol{E_1})\big)\right|\right|^2 \\
& + w_0 ||\ddot{q} - \ddot{q}^d(q_0)||^2\,, \qquad (23)\\
\text{subject to} \quad & M_{\text{mr}}\ddot{q}_{\text{mr}} + N_{\text{mr}} = J_e^T f_e + \\
& \left[J_{\boldsymbol{E_3}} - J_{r,o}\right]^T f_{\boldsymbol{E_3}} + S_{\text{mr}}^T \tau_{\text{mr}}\,, \qquad (24)\\
& J_{\boldsymbol{E_1}}\ddot{q} + \dot{J}_{\boldsymbol{E_1}}\dot{q} = 0\,, \qquad (25)\\
& J_{e,o}\ddot{q}_o + \dot{J}_{e,o}\dot{q}_o = 0\,, \qquad (26)\\
& J_{\boldsymbol{E_3}}\ddot{q} + \dot{J}_{\boldsymbol{E_3}}\dot{q} = J_{r,o}\ddot{q}_o + \dot{J}_{r,o}\dot{q}_o\,, \qquad (27)\\
& (f_{\boldsymbol{E_1}}, f_{\boldsymbol{E_3}}, f_{e,o}) \in \mathcal{C}_{\boldsymbol{E_1}} \times \mathcal{C}_{\boldsymbol{E_3}} \times \mathcal{C}_{e,o}\,, \qquad (28)\\
& \text{torque limits, joint limits, collision avoidance} \qquad (29)
\end{aligned}$$

The CoM task is now written for the multi-robot CoM $c_{\text{multi-robot}}$, that is, the *combined* CoM of the system containing all the floating-base entities (robots, objects) in contact with each other. In the case of the robot manipulating a free-floating box, $c_{\text{multi-robot}}$ is the CoM of the {robot, box} system. See Section IV-C for more details on the CoM task and balancing during manipulation.

For the sake of simplicity, we showed here a particular instantiation of the general multi-robot QP formulation: the simplified case in which the multi-robot system is restricted to one robot and one object. The general formulation in [14] with an arbitrary number N of entities is much more expressive and allows control of multi-robot multi-object co-manipulation systems, where for example two robots collaborate to manipulate an object. We have implemented our approach within this general N-entity framework, with other possible use cases in mind. Among these extensions we can mention:

- human-robot co-manipulation motion retargeting to robot-robot co-manipulation,
- human-human co-manipulation motion retargeting to robot-robot co-manipulation

B. Dealing with contact change events

We adapt the motion capture tracking algorithm described at the end of Section III in the following way. The contact events η_j now also encode the pair of entities between which the contact event happens, in a variable γ_j. γ_j takes a value in the set

$$\gamma_j \in \{(\text{rob}, \text{env}), (\text{rob}, \text{obj}), (\text{obj}, \text{env})\}\,. \tag{30}$$

l_j now encodes the index of the link/surface of the first entity in γ_j, s_j the index of the link/surface of the second entity in γ_j. Therefore, $\eta_j = (\gamma_j, l_j, s_j, t_j, \sigma_j)$. When time t_j of a contact event is reached, all three sets E_1, E_2 and E_3 are updated and the multi-robot QP is reformulated. Algorithm 1 presents the overall logic of our tracking method.

When a contact event is detected, the robot temporarily pauses human motion tracking to ensure that the contact is handled before proceeding.

For contact removals, this amounts to moving the CoM of the system into the new support polygon before removing the support. For example, when the demonstration motion goes from three contacts (e.g. two feet and one hand on the table) to two contacts (e.g. one hand and one foot), the robot must bring its CoM to be between the two planned support surfaces before removing the contact. Similarly, if the robot is about to release a manipulated object, it must first check that its individual (not multi-robot) CoM lies within its support polygon.

For contact additions, we add two temporary end effector tasks to ensure a stable contact, as described in [1]: a surface vector stabilization task to ensure that the robot's end-effector surface is parallel to the object surface, and a surface transform task to drive the distance between the two surfaces to zero.

Once the contact change conditions are fulfilled, the robot resumes tracking the demonstration.

In our experiments described below, we first record the human demonstration and play it back afterwards for the robot to track. However, our algorithm is compatible with real-time tracking, as long as there is a mechanism for contact event detection. One caveat is that (as mentioned in [7]), contact and CoM stabilization add delay to the tracking, so movements with contact changes occurring in rapid succession may be difficult to track. However, we can compensate for the delay by "buffering" the human demonstration trajectory.

Algorithm 1: Updating the tracking sets (E_1, E_2, E_3)

Input: Sequence of multi-robot contact change events $L = (\eta_1, \ldots, \eta_\mu)$

Data: $\boldsymbol{E_1}$: support contact set
$\boldsymbol{E_2}$: free tracking set
$\boldsymbol{E_3}$: manipulation tracking set

initialize $t \leftarrow 0$
initialize $j \leftarrow 1$
initialize E_1, E_2, E_3
repeat
  if $t < t_j$ *or* $j > \mu$ **then**
    Solve QP (23)
    $t \leftarrow t + dt$
  else $t = t_j$: handle contact event
    if $\gamma_j = (rob, env)$ **then**
      if $\sigma_j = 0$ **then**
        $E_1 \leftarrow E_1 \setminus \{l_j\}$
        Ensure CoM in support polygon of E_1
        $E_2 \leftarrow E_2 \cup \{l_j\}$
      else $\sigma_j = 1$
        Stabilize contact
        $E_2 \leftarrow E_2 \setminus \{l_j\}$
        $E_1 \leftarrow E_1 \cup \{l_j\}$
    else if $\gamma_j = (rob, obj)$ **then**
      if $\sigma_j = 0$ **then**
        $E_3 \leftarrow E_3 \setminus \{l_j\}$
        Ensure CoM in support polygon of E_1
        $E_2 \leftarrow E_2 \cup \{l_j\}$
      else $\sigma_j = 1$
        Stabilize contact
        $E_2 \leftarrow E_2 \setminus \{l_j\}$
        $E_3 \leftarrow E_3 \cup \{l_j\}$
    Update objective and constraints of QP (23)
    increment $j \leftarrow j + 1$
until $t = t^{end\ of\ motion}$;

C. Center of Mass tracking

Balance of the robot in quasi-static scenarios is equivalent to the CoM ground projection being inside the support polygon. We encode this desired behavior in the CoM task c. In the previous work [1], the CoM task reference value was set *independently of the human CoM* at $h(E_1)$, the center of the support polygon formed by the ground projections of the contact surfaces in E_1 (we call those surfaces *support surfaces*). In Section IV-A we presented a direct extension of this approach, adapted to manipulation retargeting and multi-robot QP formulation, by replacing the CoM of the robot with the CoM of the combined system {robot, object} when the robot is manipulating a free-floating object, $c_{\text{multi-robot}}$. In this section, we propose an additional approach that allows *tracking of the human CoM motion* during multi-contact (but non-manipulation) phases, resulting in smoother and more robust contact support changes.

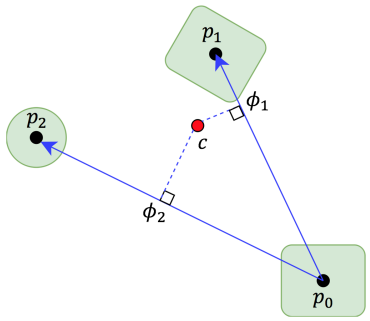

Fig. 2. Finding normalized offset values for CoM tracking. The contact surfaces correspond to a phase in the bag retrieval task in Section V, with foot contacts p_0 and p_1, hand contact p_2.

We formulate a generalization of the method proposed in Koenemann et al [7], where the robot tracks a normalized offset (denoted as ϕ) between its support surfaces. We define a *support point* p_i as the center of a support surface that is in contact with the fixed environment.

Let us denote c^{hum} as the 2D ground projection of the human CoM. First, c^{hum} is projected onto the vector from an arbitrary reference support p_0^{hum} to each of the other support points $p_i^{\text{hum}}, i = 1, \ldots, n_{\text{supports}}$, following Koenemann's formulation, to get the normalized offsets:

$$\phi_i^{\text{hum}} = \frac{\left(c^{\text{hum}} - p_0^{\text{hum}}\right) \cdot \left(p_i^{\text{hum}} - p_0^{\text{hum}}\right)}{\left\| p_i^{\text{hum}} - p_0^{\text{hum}} \right\|^2} . \tag{31}$$

The target coordinates for the robot's CoM projection can then be calculated as a linear combination of these offsets with:

$$\psi(E_1, c^{\text{hum}}) = p_0^{\text{rob}} + \sum_{i=0}^{n_{\text{supports}}} \phi_i^{\text{hum}} (p_i^{\text{rob}} - p_0^{\text{rob}}) . \tag{32}$$

We finally replace the reference CoM in (9) with

$$c^{\text{ref}}_{\text{multi-robot}} = \psi(E_1, c^{\text{hum}}) . \tag{33}$$

The process is illustrated in Fig. 2. In the case of double support, this amounts to the formulation of [7]. Our formulation generalizes to situations in which there are hands in contact with the environment. Compared to tracking CoM coordinates in Cartesian space, tracking the CoM in normalized coordinates is more robust to differences in robot size, as well as slight offsets in support position.

V. Experiments

To demonstrate the advantages of our method, we conducted three whole-body manipulation tasks in simulation on the HRP-2 and HRP-4 humanoid robots. Two of the tasks require balancing while manipulating objects, and the other demonstrates a scenario with a non-free-floating environment object. Footage of both the human demonstrations and robot motions are in the supplementary video.

For each experiment, we first record a human demonstration to guide the robot in the task. We use an Xsens inertial motion capture suit to capture human pose. The

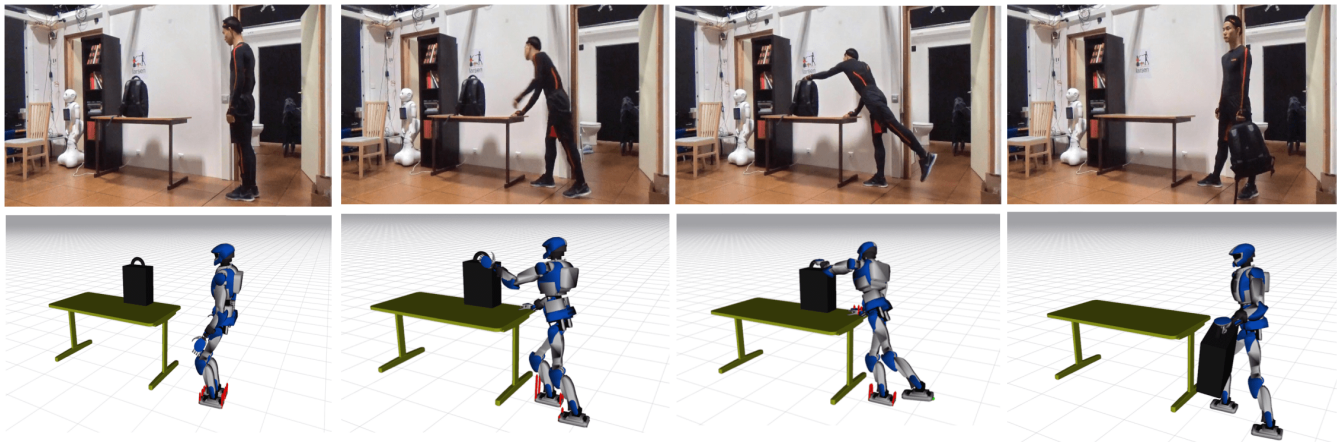

Fig. 3. First demonstration task: robot must use its hand as a support on the table while reaching for a bag.

demonstration data from the motion capture system is then annotated with contact change events (additions or removals of contacts points). In the experiments described below, this post-processing step is done manually, as it is out of the scope of this paper. However, there are many ways to extract contact event information as an offline step (as shown in [1]), or in real-time, using Xsens's contact detection API or force sensors on the human's end-effectors. After recording the human motion, we set up an identical environment for the humanoid robot in simulation, and have it use the algorithm described above to follow the human's demonstration.

In the first experiment, we demonstrate a motion to pick up a bag from a table while using the table as support to lean forward (shown in Fig. 3). This task requires multiple interactions with both static and floating-base environment objects. For this particular task, we slowed down playback of the human demonstration motion to account for contact change events that violated our quasi-static CoM stability assumption. This will be discussed further in Sections VI and VII.

For the second task, we retarget a door-opening motion to demonstrate a simple example of a non-floating-base environment object in the multi-robot QP. The human demonstration is carried out on a door whose hinge joint is not spring-loaded (spring constant $k = 0$). In the simulation, we vary the spring constant of the door joint between $k = 0, 10, 100$. Figure 5 shows the changes in actuation that the multi-robot QP computes in order to execute a visually similar motion on doors of varying resistance.

The third task demonstrates adaptation to a free-floating environment object of varying inertial parameters. It is sequenced as follows: squat down, pick up a box from the floor, shift to balance on one leg, place box on table. This is depicted in Fig. 6. The human naturally tends to push the floating leg backwards to maintain their joint CoM within the support polygon and reach further with their arms.

The original demonstration was conducted with a box of mass 0.35kg. For the experiment, we varied the mass of the box to show the effects on the robot's behavior. The results are shown in Fig. 7. Note that as the box gets

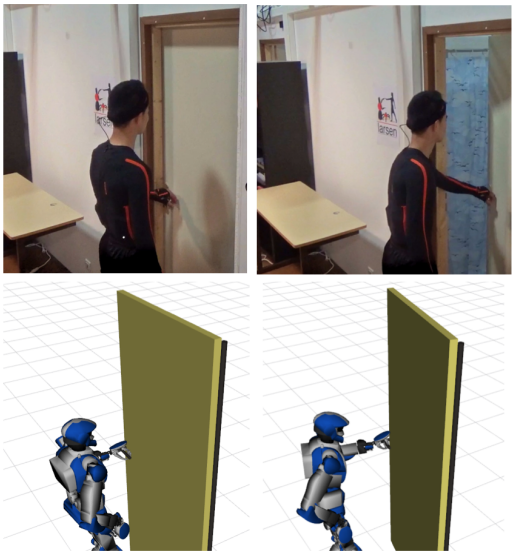

Fig. 4. Second task: opening a door.

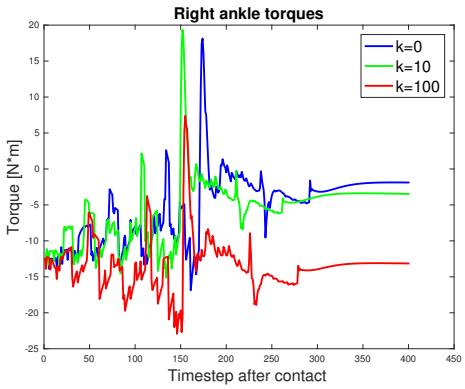

Fig. 5. Torques from right ankle during door opening task, for varying spring constants. Higher ankle torques for higher spring constants because the robot must "push" against the door's resistance.

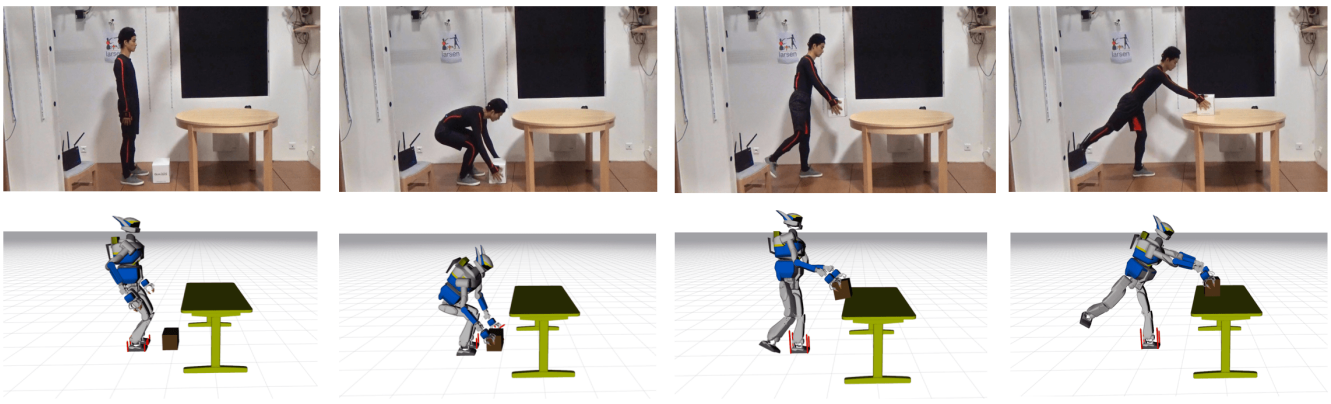

Fig. 6. Third demonstration task: picking up a box, balancing on one foot and placing on table

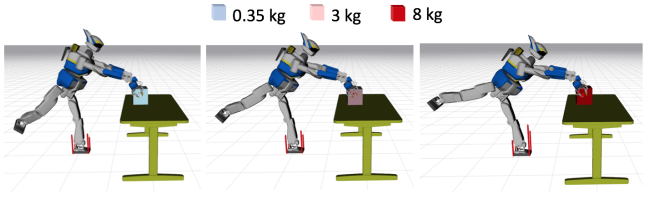

Fig. 7. The robot following a single human demonstration for boxes of varying mass: 0.35kg (left), 3kg (center), 8kg(right). Notice that the robot moves its right foot backward to compensate for the heavier box, maintaining the combined CoM position

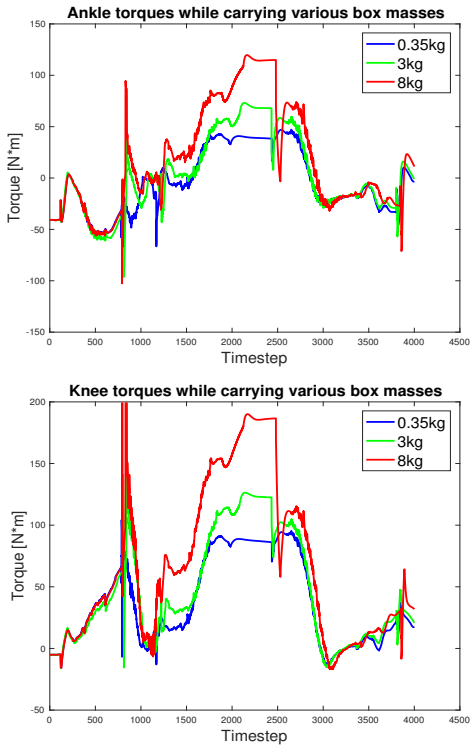

Fig. 8. Motor torques in left ankle and left knee during Task 1. A clear difference can be seen in the motor torque commands for each box weight

heavier, the robot moves its right foot further backwards to compensate and keep the combined CoM inside its single-foot support polygon. In doing so, it "ignores" other lower-weight objectives such as human motion tracking (right foot position) and reference posture tracking. This behavior emerges naturally from the weighted prioritization scheme in the QP. For this task, using a single-robot QP (without the box incorporated into the dynamics model) fails, because the robot does not compensate for the gravitational force acting on the box.

In Fig. 8, we plot the motor torque commands output by the multi-robot QP (23). It is clear that as the box gets heavier, the controller compensates with higher torques in the lower body, even with visually similar motions. Although our robots use position-controlled joints (a setpoint q_{des} set by double-integrating $\ddot{q}$ output from QP), this would be useful as feed-forward torque commands for torque-controlled robots.

VI. Conclusions

In this work, we have presented a framework for adaptive whole-body manipulation tasks in human-to-humanoid motion retargeting. The incorporation of the dynamics of all "robots" into a single optimization allows for automatic adaptation of the human demonstration to environment objects of varying physical properties. It also opens up possibilities for collaborative motions between robot-robot and human-robot teams. We demonstrated through experiments in simulation the capabilities of our framework.

An unsolved issue is the tracking of highly dynamic motions with contact set changes. In our experiments, we assume that the human's demonstrations can be tracked with quasi-statically stable motions - hence our conditions for bringing the CoM above the support polygon before removing contacts. Another limitations is that the system does not yet run in real-time (addressed below in Future Work). We also make assumptions about the robot operating in a known environment with a model provided by the human operator (this was the case, for example, in the DARPA Robotics Challenge). However, there has been progress towards automatic extraction of environment model parameters using a Kalman-filter approach [15], friction

coefficients from vision [18], and specially designed neural networks [19].

VII. Future Work

In the near future, we will implement the algorithm presented here on a real humanoid robot. We plan on using force controllers on the end effectors (as described in [15]) to account for differences between expected and real-world contact forces. We will also integrate touch-sensing gloves and shoes so that contact events can be detected on-the-fly and used for real-time motion tracking.

Many common human motions are not quasistatically stable; in fact, they use momentum to achieve better energy efficiency. In humanoid robots, this insight has led to planning and execution of dynamic locomotion using simplified models such as ZMP [20] and centroidal momentum [21]. However, these methods often assume full authority over the robot's future movements. The difficulty with tracking human motions in real-time is that the human dictates the reference motion for the robot, and the robot does not know where they will go in the future. We believe that the first step towards highly dynamic motion tracking would be to adopt some of the methods based on simplified models used in locomotion planning, such as the capture point [22]. Examining the contact wrench set is also a good way to measure feasibility, and there has been some recent work in constructing a feasible region of CoM positions and accelerations based on this notion [23].

Acknowledgements

This work received funding from the European Commission (GA no. 731540, H2020 project "An.Dy"). The authors would like to thank the IDH team in LIRMM and JRL for their support in using the multi-robot QP framework.

References

[1] A. Di Fava, K. Bouyarmane, K. Chappellet, E. Ruffaldi, and A. Kheddar, "Multi-contact motion retargeting from human to humanoid robot," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 2016, pp. 1081–1086.

[2] N. S. Pollard, J. K. Hodgins, M. J. Riley, and C. G. Atkeson, "Adapting human motion for the control of a humanoid robot," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 2. IEEE, 2002, pp. 1390–1397.

[3] B. Dariush, M. Gienger, A. Arumbakkam, Y. Zhu, B. Jian, K. Fujimura, and C. Goerick, "Online transfer of human motion to humanoids," *International Journal of Humanoid Robotics*, vol. 6, no. 02, pp. 265–289, 2009.

[4] Y. Abe and J. Popović, "Interactive animation of dynamic manipulation," in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2006, pp. 195–204.

[5] S. Nakaoka, A. Nakazawa, F. Kanehiro, K. Kaneko, M. Morisawa, H. Hirukawa, and K. Ikeuchi, "Learning from observation paradigm: Leg task models for enabling a biped humanoid robot to imitate human dances," *The International Journal of Robotics Research*, vol. 26, no. 8, pp. 829–844, 2007.

[6] K. Yamane and J. Hodgins, "Control-aware mapping of human motion data with stepping for humanoid robots," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 726–733.

[7] J. Koenemann, F. Burget, and M. Bennewitz, "Real-time imitation of human whole-body motions by humanoids," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2806–2812.

[8] Y. Abe, M. Da Silva, and J. Popović, "Multiobjective control with frictional contacts," in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2007, pp. 249–258.

[9] M. De Lasa, I. Mordatch, and A. Hertzmann, "Feature-based locomotion controllers," in *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4. ACM, 2010, p. 131.

[10] S. Feng, X. Xinjilefu, C. G. Atkeson, and J. Kim, "Optimization based controller design and implementation for the atlas robot in the darpa robotics challenge finals," in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*. IEEE, 2015, pp. 1028–1035.

[11] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2016.

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

[13] ——, "On the dynamics modeling of free-floating-base articulated mechanisms and applications to humanoid whole-body dynamics and control," in *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*. IEEE, 2012, pp. 36–42.

[14] J. Vaillant, K. Bouyarmane, and A. Kheddar, "Multi-character physical and behavioral interactions controller," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 6, pp. 1650–1662, 2017.

[15] K. Bouyarmane, J. Vaillant, K. Chappellet, and A. Kheddar, "Multi-robot and task-space force control with quadratic programming," Mar. 2017, working paper or preprint. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01495662

[16] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar, "Generation of whole-body optimal dynamic multi-contact motions," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1104–1119, 2013.

[17] T. Moulard, F. Lamiraux, K. Bouyarmane, and E. Yoshida, "Roboptim: an optimization framework for robotics," in *Robomec*, 2013, p. 4p.

[18] M. Brandao, K. Hashimoto, and A. Takanishi, "Friction from vision: A study of algorithmic and human performance with consequences for robot perception and teleoperation," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 2016, pp. 428–435.

[19] G. Martius and C. H. Lampert, "Extrapolation and learning equations," *arXiv preprint arXiv:1610.02995*, 2016.

[20] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2. IEEE, 2003, pp. 1620–1626.

[21] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*. IEEE, 2014, pp. 295–302.

[22] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*. IEEE, 2006, pp. 200–207.

[23] H. Audren and A. Kheddar, "3D robust stability polyhedron in multi-contact," Feb. 2017. [Online]. Available: https://hal-lirmm.ccsd.cnrs.fr/lirmm-01477362

[24] K. Bouyarmane, J. Vaillant, N. Sugimoto, F. Keith, J.-i. Furukawa, and J. Morimoto, "Brain-machine interfacing control of whole-body humanoid motion," *Frontiers in systems neuroscience*, vol. 8, 2014.