

# Generating Assistive Humanoid Motions for Co-Manipulation Tasks with a Multi-Robot Quadratic Program Controller

Kazuya Otani

Karim Bouyarmane

Serena Ivaldi

**Abstract**—Human-humanoid collaborative tasks require that the robot take into account the goals of the task, interaction forces with the human, and its own balance. We present a formulation for a real-time humanoid controller which allows the robot to keep itself balanced, while also assisting the human in achieving their shared objectives. We achieve this with a multi-robot quadratic program controller, which solves for human dynamics reconstruction and optimal robot controls in a single optimization problem. Our experiments on a simulated robot platform demonstrate the ability to generate interaction motions and forces that are similar to what a human collaborator would produce.

## I. INTRODUCTION

In recent years, improved sensing capabilities and safer hardware have allowed robots to move into shared spaces with humans, such as factory floors and homes. In many cases, the robots do not replace humans; they complement human capabilities and relieve them of arduous tasks. Co-manipulation systems combine the reasoning abilities of human operators with the precision and power of robots, which results in increased productivity and human comfort.

An ideal co-manipulation robot would be able to:

- 1) Learn general categories of motions to be performed
- 2) Adapt to slight variations in conditions (e.g. human or object position)
- 3) Actively aid the human in achieving shared goals

In this paper, we propose a humanoid robot controller for physical human-robot interaction (pHRI) tasks that addresses the third point: a controller that can produce optimal motions in real-time, considering both the robot's balance and human assistance goals. We achieve this using a multi-robot quadratic program controller, which maintains an internal representation of the human's whole-body dynamics along with those of the robot and any other manipulated objects.

Intuitively, this corresponds to how humans adjust their motions based on the anticipated effects of interactions with other objects on their own dynamics. We bend down lower when picking up heavy objects than for light objects. Similarly, we lean back and pull harder when trying to help an adult stand up, than when helping a small child. By incorporating a whole-body dynamics model of the human in our controller, we take a first step towards proactive human-humanoid interaction, as opposed to reactive interaction

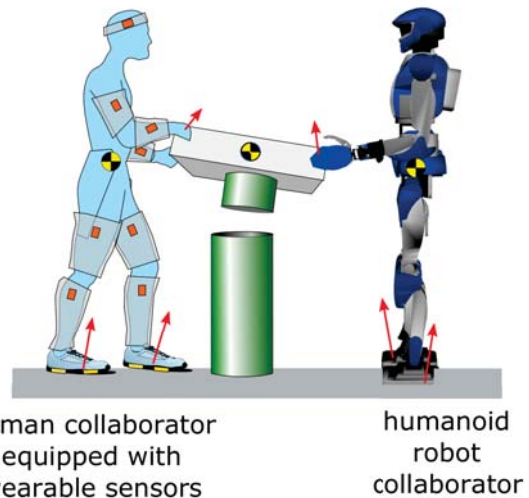


Fig. 1. In collaborative tasks, the humanoid robot must reason about: goals of the task, interaction forces with the human, and its own balance. A human collaborator equipped with wearable sensors can provide further information about the human's movement during the task.

based on recent sensor data. We demonstrate in simulation experiments that the controller can generate interaction motions/forces similar to what a human partner would produce, in co-manipulation and balance assistance tasks.

## A. Related Work

Control algorithms for pHRI have several modalities available for the robot's perception of the human: vision (e.g. gesture/posture detection), sound (e.g. voice commands), bio-signals (e.g. EMG), and force/pressure sensors (e.g. force-torque on end effectors). Of these modalities, force-torque sensors (haptic data) have been used most often, due to the simplicity of the underlying mechanism. The haptic data is usually used to regulate the robot's impedance (resistance to motion caused by external forces) in order to generate compliant behavior [1] [2].

Hybrid methods using multiple sensing modalities have been shown to improve capabilities in certain tasks. Early human-humanoid interaction work by Yokoyama et al showed that visual and haptic feedback, along with voice commands, allow a humanoid robot to collaboratively carry a large object with a human [3]. Agravante et al use a similar approach to collaboratively carry a table while balancing a free-rolling ball on the table [4]. They use a decoupled approach which separates manipulation and balance control, while the human is sensed as an external force.

e-mail: kazuya.otani@inria.fr, karim.bouyarmane@loria.fr, serena.ivaldi@inria.fr

The authors have the following affiliations:

- Inria Nancy, Villers-lès-Nancy, F-54600, France
- CNRS, Loria, UMR 7503, Vandœuvre-lès-Nancy, F-54500, France
- Université de Lorraine, Loria, UMR 7503, Vandœuvre-lès-Nancy, F-54500, France

A limitation in most existing controllers for physical human-robot interaction is that the representation of the human state is vastly simplified as an external force or end effector poses. The whole-body configuration and dynamics of the human are rarely considered, which limits the types of interaction that are possible. Much of this is due to limitations in sensing: getting information about a human's whole-body configuration is much more difficult than sensing interaction forces. However, we feel that recent advances in whole-body motion tracking suits [5] and visual pose estimation (e.g. [6]) allow us to relax this constraint, and investigate the possible applications. For example, having a kinematic model of the human body could allow us to reason about the ergonomics (e.g. posture, exertion) of a collaborative motion.

Among the works that do consider a full human model, there have been applications in: optimizing human comfort in selecting handoff configurations [7], offline trajectory optimization for predicting the motion of an exoskeleton-assisted human [8], finding configurations for a fixed-base manipulator that minimize human exertion in co-manipulation [9]. These methods are used for higher-level planning and/or only consider kinematic data. The work in this paper was developed to test our hypothesis that there are certain tasks, such as co-manipulation and balance assistance, in which having a whole-body dynamics model of the human is useful for controlling interaction motions in real-time.

For humanoid control, quadratic program (QP) controllers have become ubiquitous in recent years due to their robustness, speed, and flexibility. QP controllers allow intuitive encoding 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 [10] and De Lasa et al [11] demonstrated multi-objective control for generating natural movements in computer animated characters. In the recent DARPA Robotics Challenge, many of the teams relied on a QP for their joint-level inverse dynamics controller [12], [13], [14].

Our approach makes use of the multi-robot QP (MRQP) controller introduced by Vaillant et al for animated characters [15], and applied to real robots in [16]. The MRQP extends QP-based humanoid controllers to consider the combined dynamics of multiple robots.

To reconstruct human dynamics, we use a QP controller to map the motion capture data onto a simulated human. The use of a task-based controller and an approximate model for the reconstruction of human motions has been shown to be quite reliable in prior works. In [17], a task-based controller that seeks to minimize position error (against human markers) and muscular effort on a musculoskeletal model are shown to provide a good match with motion capture data. In [18], a QP controller is used to simulate a rigid body tree model of a human, and is empirically shown to produce similar trajectories to real humans for multiple motions.

## B. Contributions

Our contribution is a formulation for reconstructing the dynamics of human motion and solving a humanoid control problem in a single optimization, using a MRQP controller to explicitly model the whole-body dynamics of both the human and the robot. This allows for real-time control of a humanoid robot during co-manipulation tasks, generating motions that simultaneously keep the robot balanced and assist the human. We demonstrate that the controller can assist the human in symmetric co-manipulation tasks and maintaining balance. The use of a full-body model for the human, as well as a combined human-robot dynamical system, allows a richer "vocabulary" for specifying desired behaviors as optimization objective functions.

The rest of this paper is organized as follows: Section II reviews the formulation of QP and MRQP controllers. Section III outlines our approach for modeling human dynamics and physical interaction in a MRQP. Section IV demonstrates the advantages of our controller through experiments in simulation. Section V discusses the results and limitations of this method. Section VI draws conclusions and discusses possibilities for future work.

## II. PRELIMINARIES

### A. Weighted QP control

QP control for humanoid robots consists in solving, at each control time-step, the following optimization problem:

$$\min_{\ddot{q}, \tau, \lambda} \sum_k w_k \|\ddot{y}_k - \ddot{y}_k^d\|^2, \quad (1)$$

- s.t. dynamics constraints  
no-slip contacts (with environment)  
forces within friction cones  
joint position, velocity, torque limits  
collision avoidance

The decision variables of this problem are the generalized acceleration  $\ddot{q}$ , where  $q$  denotes the configuration of the robot including its 6D free-floating base,  $\tau$  are the actuation joint torques, and  $\lambda$  the coefficients of contact forces basis vectors along linearized friction cones.

By using a linear approximation of the friction cones at the contact points, and writing the joint limits and collision-avoidance constraints to be linear in the decision variables, the optimization problem above is a QP [19].

$y_k$  denote quadratic objectives that encode desired motion, which we will call "tasks".  $\ddot{y}_k^d$  is the desired acceleration of the task, which can be derived from a desired trajectory or set-point of the task.  $w_k$  is the weight/soft priority of the task. The tasks can be mapped from the configuration space to any kind of 'operational space' with Jacobians, as shown in [11]. One possible mapping is with PD regulation towards a set-point:

$$\ddot{y}_k^{des} = k_p(y_k^{ref} - y_k) - k_v\dot{y}_k \quad (3)$$

$$\ddot{q}_{des} = J_k^{-1}(\ddot{y}_k^{des} - \dot{J}_k\dot{q}) \quad (4)$$

The QP controller is often at the bottom of a planning hierarchy, whose top layers produce higher-level plans using simplified models of the robot [20] or multi-contact plans with nonlinear optimization [21].

### B. Multi-Robot QP control

In [16], the QP control framework is extended to control systems of multiple “robots” interacting among each other or with a dynamic environment. Let  $n$  be the number of such entities. In a collaborative robot-robot co-manipulation scenario,  $n = 3$  (two robots and the manipulated object).

The MRQP framework is based on the assumption that each entity  $i$  ( $i = 1 \dots n$ ) can be modeled with the general dynamics equation of motion

$$M_i(q_i)\ddot{q}_i + N_i(q_i, \dot{q}_i) = \sum_p J_{i,p}^T(q_i) f_{i,p} + S_i^T \tau_i, \quad (5)$$

which accounts for all possible actuated/non-actuated/under-actuated cases through the control selection matrix  $S_i$ . The general equation (5) also accounts for all possible fixed-base/floating-base cases through the inclusion (or exclusion) of the the base frame configuration in  $q$ . We can thus use this equation to model humanoid robots (possibly multiple), as well as floating objects (e.g. boxes) or articulated passive environments objects (e.g. doors).

The  $n$  entities physically interact with each other by exchanging contact forces, which come in action-reaction pairs according to Newton’s second law. Each contact force  $f_{i,p}$  on entity  $i$  is either applied by the fixed inertial environment, or by one of the other entities  $j$  and appears with an opposite sign in that entity’s equation as  $f_{i,p} = -f_{j,p'}$ .

By rearranging the forces, we can keep exactly one representative of each action-reaction pair ( $f_{i,p}, f_{j,p'}$ ) as decision variables of the MRQP, showing that there exists a permutation matrix  $\Psi$  such that we can combine all of the equations of motion of the  $n$  entities in one single equation:

$$M(q)\ddot{q} + N(q, \dot{q}) = J_0^T F^0 + (J_- - \Psi^T J_+)^T F^- + S\tau, \quad (6)$$

where  $q$ ,  $\tau$ ,  $N$  denote the stacked vectors of generalized configurations, actuation torques, and nonlinear effects respectively,  $M$  and  $S$  are the stacked block-diagonal matrices of  $M_i$  and  $S_i$  respectively,  $F^0$  is the stacked vector of fixed inertial environment contact forces with corresponding stacked block-diagonal Jacobian matrix  $J_0$ ,  $F^-$  the stacked vector of single representatives of forces between entities,  $J_-$  and  $J_+$  are Jacobian matrices corresponding to the contact points between the different entities of the multi-robot system. (Detailed derivations of these components in [16]) We can also derive no-slip contact constraints between the different entities and with the environment as  $(J_- - \Psi J_+)\dot{q} = 0$  and  $J_0\dot{q} = 0$ .

Once the combined dynamics are formed, the formulation is equivalent to a single-robot QP controller, with the addition of the constraints for contacts and collision-avoidance between the entities. Denoting  $\lambda^0$  and  $\lambda^-$  as the coefficients along the linearized friction cone generators of  $F^0$  and  $F^-$  respectively:

$$\min_{\ddot{q}, \tau, \lambda^0, \lambda^-} \sum_k w_k \|\ddot{y}_k - \ddot{y}_k^d\|^2, \quad (7)$$

$$\begin{aligned} \text{s.t. } & M\ddot{q} + N = J_0^T F^0 + (J_- - \Psi^T J_+)^T F^- + S\tau \\ & J_0\ddot{q} + \dot{J}_0\dot{q} = 0 \\ & (J_- - \Psi^T J_+)\ddot{q} + (\dot{J}_- - \Psi^T \dot{J}_+)\dot{q} = 0 \\ & \lambda = (\lambda^0, \lambda^-) \geq 0 \\ & q_{min} \leq q \leq q_{max} \\ & \dot{q}_{min} \leq \dot{q} \leq \dot{q}_{max} \\ & \tau_{min} \leq \tau \leq \tau_{max} \end{aligned} \quad (8)$$

This optimization problem can be solved at real-time rates (200Hz) to control multiple robots. With an interface that simplifies the specification of desired behavior as quadratic objective tasks, the multi-robot QP allows for the generation of complex interaction behaviors from easily-interpretable objectives. The tasks can be written for the combined system (e.g. combined center-of-mass) or imply desired behavior of the whole system through a task on a single robot (e.g. task on the position of the co-manipulated object that drives all the robots in contact with the object). In previous works, this has been used for collaborative carrying between two robots, environment object (box, door) manipulation, and human-to-humanoid motion retargeting of manipulation motions.

In this paper, we consider three entities ( $n = 3$ ): humanoid robot, human (incorporated using the method described in the next section), and a co-manipulated object.

## III. METHODS

### A. Human model

We incorporate a whole-body dynamics model of the human into the multi-robot QP as one entity of the system whose dynamics are modeled by equation (5). The simulated human tracks the real human’s motions, which the robot then uses as a virtual approximation of the real human to reason about the human-robot system’s combined dynamics.

In formulating our multi-robot QP, we assume that the whole-body configuration of the human is known. To capture human motion we use an Xsens MVN inertial motion capture suit [5], a wearable system with 17 wireless IMUs embedded in a lycra suit, which tracks motion at 240Hz. IMU data are mapped onto the motion of a calibrated biomechanical model with 23 segments, connected via 22 three-dimensional joints. The motion capture system provides: link positions/orientations and joint angles, along with their first and second time derivatives.

The Xsens motion capture data is mapped onto our simulated human model, which is a 22-joint rigid body tree model, described in a URDF format. The limb lengths and masses are parametrized as a function of the subject’s height and weight, based on average anthropometric coefficients [22], with further customization possible to match a subject’s specific measurements.

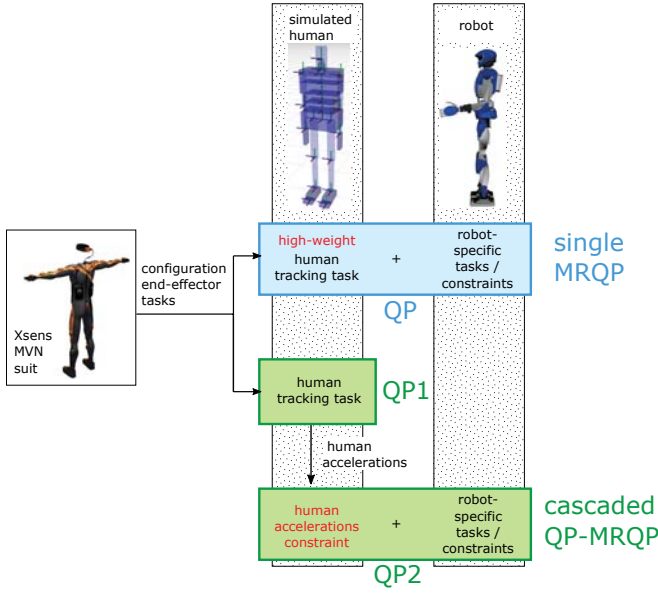


Fig. 2. Diagram showing the difference between our two methods for human motion reconstruction. In both methods, the motion capture data is mapped onto our parametrized 22-joint human model, which is then used to compute the robot’s controls.

### B. Human motion reconstruction

To accurately reconstruct human motion with our MRQP, we build on the ideas from [23] and set high-weight motion tracking objectives on the human “robot”, which take motion capture data as time-varying setpoints.

We use a posture task (objective on desired generalized configuration  $q_{des}$ ) to track the joint angles of the human’s motion. On top of that, we set human motion tracking tasks on end effector pose (position and orientation tasks), since this is the most important feature of human motion for applications involving physical interaction (these are the links most often in contact with the human and external environment). These two tasks combine to produce an approximation of the motion capture data on our simulated human model.

To integrate the human motion reconstruction into a MRQP, we investigated two methods (illustrated in Figure 2).

1) *Single MRQP*: Perform human motion reconstruction directly in the multi-robot QP, ensuring that the simulated human closely tracks the real human’s motion by setting human motion tracking task weights to be significantly higher than robot’s task weights. This is the “soft priority” approach.

2) *Cascaded QP-MRQP*: Have a single-robot QP for human motion tracking with the simulated human (QP1), and a multi-robot QP for calculating the robot’s motion given the simulated human’s dynamics (QP2). The solution for the human motion  $\ddot{q}_{human}$  from QP1 is set as an equality constraint on the human model’s motion in QP2. This is the “hard priority” approach.

Method 1 has the advantage that it is simpler to implement and is less computationally expensive (it only solves one

QP). The disadvantage of this method is that it requires tuning of the weights to trade off between rigidly tracking human motion (so the controller has an accurate estimates of the real human’s motion) and achieving the robots tasks. In practice, we found that this method could reconstruct the human motion with reasonable accuracy (see Section IV and Figure 4).

Method 2 decouples the human model’s motion from the optimization for the robot’s motion, which allows stricter tracking of the motion capture data by the simulated human. However, in our experiments we found that it was less stable. We often saw the QP fail because of violated constraints, especially for fast movements. Based on our investigations, we believe this is because accumulated numerical errors lead to drift between the simulated human’s states  $q_{human}$  in QP1 and QP2, which means that large  $\ddot{q}_{human}$  commands cause discrepancies in end effector motions large enough for other QP2 constraints (friction cone, no-slip contact) to be violated. In this context, Method 1 can be interpreted as a way to allow some “slack” in the human motion tracking to make the QP solvable.

For the reasons above, our implementation used Method 1 for integrating human motion tracking into the MRQP framework.

Collaboration between humans and robots often involves co-manipulation of objects (e.g. collaborative pick & place of large/heavy objects). In those cases, the objects also need to be modeled and added to the QP problem. Following the MRQP framework, objects are modeled as independent entities with their own dynamics equations. We make the assumption that the geometrical model and inertial parameters of the manipulated object are known.

### C. Robot objectives

The quadratic objectives in the MRQP controller define the metrics that the robot seeks to minimize in its motions. The tasks that we use in our approach can be separated into two categories: individual objectives for the robot, and interaction objectives that define the robot’s behavior as a function of the human’s motion.

For the robot’s individual tasks, we set objectives that keep the robot balanced and in a natural posture away from singularities.

Balance is encoded in a center-of-mass (CoM) task, which encourages the robot to keep its CoM above the center of its support polygon, and a CoM bound constraint, which limits  $\ddot{p}_{CoM}^{des}$  to ensure that the ground projection of the robot’s CoM does not go outside its support polygon. The CoM bounds are defined in hyperplane representation  $\{p_{CoM} \mid Ap_{CoM} \geq b\}$ , and implemented as a damping behavior that slows down the CoM as it nears the boundaries of the convex hull:

$$\dot{d} + \ddot{d}\Delta t \geq -\xi \frac{d - d_s}{d_i - d_s} \quad (9)$$

with  $d$  as the distance between  $p_{CoM}$  and the nearest hyperplane,  $d_i$  the interaction distance at which damping turns on,  $d_s$  the security distance (minimum distance by which to stay inside convex hull),  $\xi$  the damping coefficient.



We include a posture (full configuration space) task on the robot, 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 acts as a prior on reasonable motions and helps generate more natural-looking movements.

#### D. Interaction objectives

The tasks described above encourage the robot to be individually balanced. We then add objectives that define the desired interaction between the robot and the human. These objectives define the robot's motion as an implicit function of the human's motion, which is solved for by the MRQP at each timestep.

The first interaction task is a regularization/minimization objective on the robot's contact forces with other entities ( $\lambda_-$  in Equation (7)), to avoid unrealistic behaviors in which the humanoid leans on/pushes other entities excessively.

For the experiments in this paper, we assumed that the human and the robot are performing a symmetric motion in which they face each other and perform mirrored versions of each other's motions. To achieve this, we set the desired pose of the robot's end effector to be a mirrored version of the human's end effector pose.

To derive the mirrored pose, we perform mirroring operations on the transform (where  $T = \{x, y, z, q_w, q_x, q_y, q_z\}$ ) from the human's reference frame to their end effector  $T_{H \text{ ref}}^{H \text{ hand}}$ , which is then projected from the robot's reference frame to  $T_{R \text{ ref}}^{R \text{ hand}}$ . The reference frames in world frame  $\{T_W^{H \text{ ref}}, T_W^{R \text{ ref}}\}$  are chosen to be at the midpoints between each robot's feet at initialization, with  $x, y, z$  axes pointing in the forward, left, and up directions respectively. We defined a "mirroring" operation  $T_{\text{mirrored}} = \text{Mirror}(T_H)$  on position and orientation:

$$\begin{aligned} q_H = \{w, x, y, z\} &\rightarrow q_{\text{mirrored}} = \{z, y, x, w\} \\ x_{\text{mirrored}} &= x_H \\ z_{\text{mirrored}} &= z_H \\ y_{\text{mirrored}} &= \begin{cases} y_H & \text{if no contact} \\ y_0^R + (y_0^H - y_H) & \text{if contact} \end{cases} \end{aligned} \quad (10)$$

where  $y_0$  are the coordinates of the end effectors when the contact is initially established. The mirroring rule for the  $y$  direction amounts to "mirror when there's no contact, move together in longitudinal direction when there is contact". The desired robot end effector position is calculated with:

$$(T_W^{R \text{ hand}})_{\text{des}} = \text{Mirror}(T_{H \text{ ref}}^{H \text{ hand}}) \cdot T_W^{R \text{ ref}} \quad (11)$$

Some other interaction tasks that can be used are:

- Collision avoidance between robots
- Minimize distance/orientation errors between human and robot end-effectors (e.g. for a handoff)
- Simulated human joint torque - induce the robot to carry more load

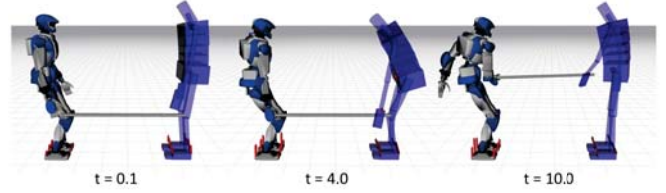


Fig. 3. Steps of Experiment 1. See attached video for a better view.

## IV. EXPERIMENTS

To demonstrate the effectiveness of our pHRI framework, we performed several experiments in dynamics simulation with an HRP-4 humanoid robot. We show that our controller can generate realistic interaction motions/forces for symmetric placement and balance assistance tasks.

These experiments were done using recorded motion of a human during a human-human collaborative task: only one partner was equipped with an XSens MVN motion capture suit [5], as the second partner is replaced by the robot in our simulations. Obviously this is not completely analogous to using this pHRI controller in real life; the limitations are discussed in Section V.

#### A. Symmetric placement task

The first experiment is a collaborative pick-and-place experiment, in which the human and robot work together to move a pole from one side of their bodies to the other. This task shows a simple application of the mirroring heuristic for generating robot follower motion. The pole is modeled as a floating-base unarticulated robot whose dynamics are incorporated into the combined dynamical system when contact is established.

This is a simple motion that could possibly be replicated using other pHRI controllers. Even a simple impedance control framework would allow this motion, provided that the human exerts the extra torque needed to "push" the robot in the right direction. However, our controller reduces the amount of force the human needs to exert by encoding an assumption about the desired motion in the "mirroring" task.

#### B. Balance assistance

In this experiment, the human leans their CoM outside of their own support polygon while holding onto a pole together with the robot. In the initial recording of the human-human motion, this required the partner (who the robot replaces) to pull back on the pole, keeping the human in balance.

The simulation results (depicted in Figures 5 and 6) show that our controller calculates the effect that the human's leaning will have on the combined system, and generates a motion in which the robot leans back and braces itself to pull on the barbell/human. This assistive behavior emerges from the robot's individual CoM task, as well as the human model's motion tracking task.

This experiment shows the advantage gained from modeling the whole-body dynamics of the human. Other approaches that use a less complete representation of the human

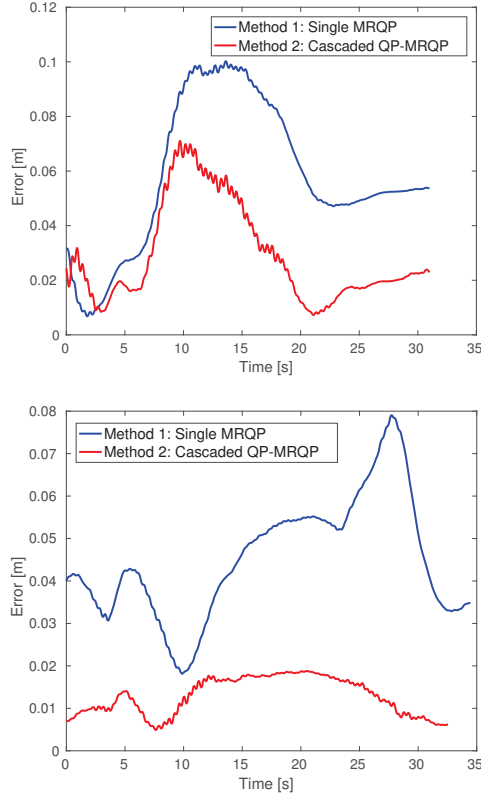


Fig. 4. Tracking errors: difference of end effector (top) and CoM (bottom) position between human motion capture data and our simulated human model in Experiment 1. Results shown for both methods of human motion reconstruction. As expected, the Cascaded QP-MRQP method tracks closer. The Single MRQP method tracks within 10cm error.

would have a hard time cleanly formulating the robot's assistive behavior in this situation.

### C. Implementation details

Below is a complete list of the tasks used in the experiments, along with their relative weights. All tasks were implemented with the PD set-point regulator described in Equation 3.

- Robot individual tasks: center-of-mass (100), posture (1)
- Robot interaction tasks: right end effector (50), contact force minimization (4)
- Human tracking tasks: center-of-mass (200), posture (60), left end effector (14), right end effector (60)

Contact change events were manually annotated in the recorded data, and handled via the method described in our prior work on motion retargeting [24]. In co-manipulation scenarios, we assume that the contact changes are simultaneous for the human and robot.

## V. LIMITATIONS OF TESTING PHRI IN SIMULATION

Our experiments were done in simulation, using pre-recorded motion of a human during a human-human collaborative task. Thus, our experiments show how a robot running our controller would react to a predetermined human

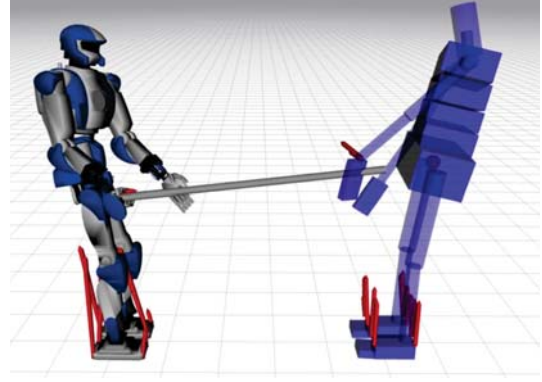


Fig. 5. Robot assisting human with balance in Experiment 2. Forces on robot and human shown in red.

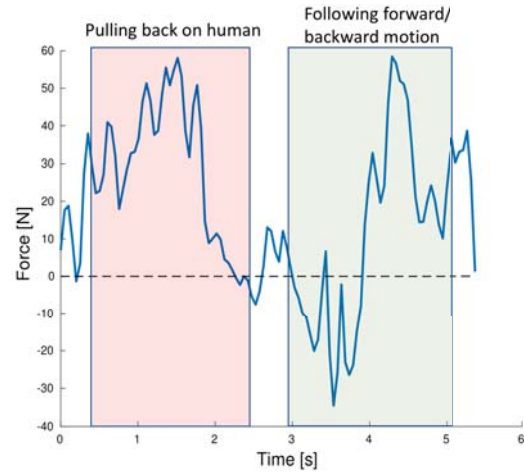


Fig. 6. Interaction forces (forces felt by the robot) in the  $x$ -direction (forward/backward) on the robot's right end effector during Experiment 2. Our controller generates realistic interaction forces that keep the collaborative task balanced.

motion. This means that motions which actively affect the human's motion (pushing/pulling CoM, encouraging human to lower arms, etc.) were excluded from these experiments. The experiments shown here demonstrate our controller's ability to generate motions/forces that are similar to what the human partner executed during the initial recording of the human-human motion. We also assume that the robot knows the goals of the task *a priori*, e.g. symmetric motion with the human.

To apply our framework in real pHRI scenarios, the robot's low-level motor control must allow some compliance so that the human can influence the robot's motion, and to account for small errors in the MRQP's human motion reconstruction (shown in Figure 4). The MRQP controller outputs a set of control setpoints  $\{\ddot{q}_{des}, \tau_{des}, \lambda_{des}\}$ ; in our simulation experiments, the HRP-4 was controlled with joint position control by integrating  $\ddot{q}_{des}$  twice to get  $q_{des}$ . To add compliance to the robot's motions, we can use a motor torque controller that takes  $q_{des}$  and  $\tau_{des}$  as inputs.

## VI. CONCLUSIONS AND FUTURE WORK

This paper presents a formulation of humanoid control for pHRI tasks, using a multi-robot QP to model the whole-body dynamics of the human. We show in simulation experiments that the controller can generate assistive motions for accomplishing collaborative tasks. This method is flexible and is easily adapted to varying robot morphologies, as well as different motion objectives.

Our future work will focus on implementing this controller on a real robot for physical experiments. Some additional avenues for future work are described below:

1) *Integration with methods for predicting human intent:* With our current approach, we make *a priori* assumptions about the type of motion the human will want to make. Integration of intent prediction algorithms into our framework would allow the robot to be more flexible; it won't have to make strong assumptions on the type of motion to be performed. Instead, the robot can be trained to perform a diverse set of collaborative motions, and infer the human's intended motion online. Thereafter, the robot will be able to actively help the human achieve the task with motion-specific interaction objectives. ("Intent" can consist of goal configurations, trajectories, speed, etc.; some these are demonstrated in [25].)

2) *Better modeling of human dynamics, reactions to external forces:* To get a more accurate dynamics model of the human body than a rigid body tree, we can use neuromuscular human simulations which simulate individual muscles and tendons [26]. However, these simulations are too computationally expensive to run in real-time. One way around this is to have a surrogate model for the neuromuscular simulation (e.g. a neural network) which can be used as an inexpensive, real-time approximation to the full model and run in real-time as part of the QP controller.

## 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, and Pauline Maurice for implementing the human URDF model generator.

## REFERENCES

- [1] K. Kosuge and N. Kazamura, "Control of a robot handling an object in cooperation with a human," in *Robot and Human Communication, 1997. RO-MAN'97. Proceedings., 6th IEEE International Workshop on*. IEEE, 1997, pp. 142–147.
- [2] R. Ikeura and H. Inooka, "Variable impedance control of a robot for cooperation with a human," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 3. IEEE, 1995, pp. 3097–3102.
- [3] K. Yokoyama, H. Hando, T. Isozumi, Y. Fukase, K. Kaneko, F. Kanehiro, Y. Kawai, F. Tomita, and H. Hirukawa, "Cooperative works by a human and a humanoid robot," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 3. IEEE, 2003, pp. 2985–2991.
- [4] D. J. Agravante, A. Cherubini, A. Bussy, P. Gergondet, and A. Kheddar, "Collaborative human-humanoid carrying using vision and haptic sensing," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 607–612.
- [5] "Xsens MVN," <http://www.xsens.com/products/xsens-mvn/>.
- [6] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.
- [7] A. Bestick, R. Bajcsy, and A. D. Dragan, "Implicitly assisting humans to choose good grasps in robot to human handovers," in *International Symposium on Experimental Robotics*. Springer, 2016, pp. 341–354.
- [8] M. Millard, M. Sreenivasa, and K. Mombaur, "Predicting the motions and forces of wearable robotic systems using optimal control," *Frontiers in Robotics and AI*, vol. 4, p. 41, 2017.
- [9] W. Kim, J. Lee, L. Peternel, N. Tsagarakis, and A. Ajoudani, "Anticipatory robot assistance for the prevention of human static joint overloading in human-robot collaboration," *IEEE Robotics and Automation Letters*, 2017.
- [10] 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.
- [11] 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.
- [12] K. Bouyarmane, J. Vaillant, F. Keith, and A. Kheddar, "Exploring humanoid robots locomotion capabilities in virtual disaster response scenarios," in *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*. IEEE, 2012, pp. 337–342.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] K. Bouyarmane, J. Vaillant, K. Chappellet, and A. Kheddar, "Multi-robot and task-space force control with quadratic programming," 2017.
- [17] O. Khatib, E. Demircan, V. De Sapio, L. Sentis, T. Besier, and S. Delp, "Robotics-based synthesis of human motion," *Journal of Physiology-Paris*, vol. 103, no. 3, pp. 211–219, 2009.
- [18] P. Maurice, "Virtual ergonomics for the design of collaborative robots," Ph.D. dissertation, Paris 6, 2015.
- [19] 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.
- [20] S. Feng, "Online hierarchical optimization for humanoid control," 2016.
- [21] K. Bouyarmane and A. Kheddar, "Humanoid robot locomotion and manipulation step planning," *Advanced Robotics*, vol. 26, no. 10, pp. 1099–1126, 2012.
- [22] D. B. Chaffin, G. Andersson, B. J. Martin *et al.*, *Occupational biomechanics*. Wiley New York, 1999.
- [23] 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.
- [24] K. Otani and K. Bouyarmane, "Adaptive whole-body manipulation in human-to-humanoid multi-contact motion retargeting," 2017.
- [25] G. Maeda, M. Ewerton, R. Lioutikov, H. B. Amor, J. Peters, and G. Neumann, "Learning interaction for collaborative tasks with probabilistic movement primitives," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*. IEEE, 2014, pp. 527–534.
- [26] S. L. Delp, F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, and D. G. Thelen, "Opensim: open-source software to create and analyze dynamic simulations of movement," *IEEE transactions on biomedical engineering*, vol. 54, no. 11, pp. 1940–1950, 2007.