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

Kazuya Otani, Karim Bouyarmane, Serena Ivaldi

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 stable, while also assisting the human in achieving their shared objectives. This is achieved with a multi-robot quadratic program controller, which solves for human motion reconstruction and optimal robot controls in a single optimization problem. Our experiments on a simulated robot platform demonstrate the ability to generate interactions motions and forces that are similar to what a human collaborator would produce.

I. INTRODUCTION

Robots have traditionally been used in isolation, e.g. within cages in factories. This was because robots were only capable of repeating preprogrammed behavior, and did not have any means of adjusting to the environment around them. In recent years, improved sensing capabilities and 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 motions/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 stability and human assistance goals. This is achieved via 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. We demonstrate in simulation experiments that the controller can generate interaction motions/forces similar to what a human partner would produce.

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, Vandœvre-lès-Nancy, F-54500, France
- Université de Lorraine, Loria, UMR 7503, Vandœvre-lès-Nancy, F-54500, France

human collaborator
equipped with
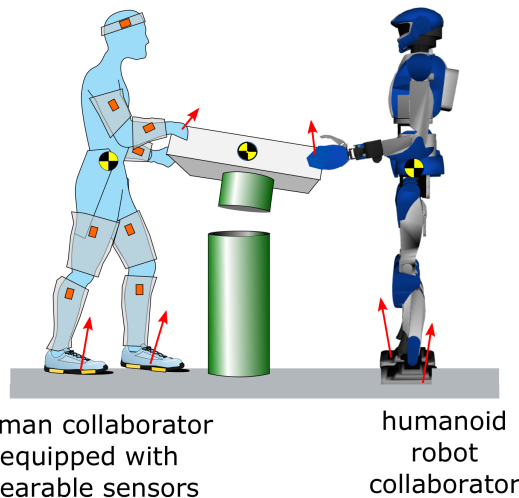wearable sensors

humanoid
robot
collaborator

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 movement during the task.

A. Related Work

Control algorithms for physical human-robot interaction (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, EEG [1]), 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 [2] [3].

Recently, hybrid methods using multiple sensing modalities have been shown to improve capabilities in certain tasks. Agravante et al show that a combination of visual and haptic feedback can be used 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.

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 the dynamics of the human are rarely considered, which limits the types of interaction that are possible. For example, it

is impossible to reason about the human's ergonomics (e.g. posture, exertion) without at least a kinematic model of the human body.

Among the works that do consider a full human model, there have been applications in: optimizing human comfort in selecting handoff configurations [5], offline trajectory optimization for predicting the motion of an exoskeleton-assisted human [6], finding configurations for a fixed-base manipulator that minimize interaction stresses in a co-manipulation task [7].

For humanoid control, quadratic program (QP) 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], [12].

Our approach makes use of the multi-robot QP (MRQP) controller introduced by Vaillant et al for animated characters [13], and applied to real robots in [14]. The MRQP extends QP-based humanoid controllers to consider the combined dynamics of multiple robots.

We use a QP controller to map the motion capture data onto a simulated human; the use of a task-based controller and a rigid-body-tree model for the reconstruction of human motions has been shown to be quite reliable in prior works. In [15], a controller that seeks to minimize position error (against human markers) and muscular effort on a musculoskeletal model are shown provide a good match with motion capture data. In [16], a QP controller is used to simulate human motion 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 multi-robot QP 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 stable and assist the human. We demonstrate that the controller can assist the human in symmetric co-manipulation tasks, as well as in maintaining balance.

The rest of this paper is organized as follows: Section II reviews the use of QP and MRQP controllers. Section III outlines our approach for modeling human dynamics and physical interaction in a multi-robot QP. Section IV demonstrates the advantages of our controller through experiments in simulation. Section V discusses the results and provides additional insights. Section VI draws the 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:

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

$$\text{subject to:} \begin{cases} \text{dynamics constraints} \\ \text{contacts no-slip (with environment)} \\ \text{forces within friction cones} \qquad (2) \\ \text{joint position, velocity, torque limits} \\ \text{collision avoidance} \end{cases}$$

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

Using a linear approximation of the friction cones at the contact points, and by carefully writing the joint limits and collision-avoidance constraints to be linear in the decision variable vector, the optimization problem above is a QP [17].

y_k denote tasks (mapping from the configuration space to any kind of 'operational space'), $\ddot{y}_k^d$ 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 of the task.

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 [18] or nonlinear optimization-based [19] multi-contact planning [20].

B. Multi-Robot QP control

In [14], 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, $n = 3$ (the 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

$$M_i \ddot{q}_i + N_i = \sum_p J_{i,p}^T f_{i,p} + S_i^T \tau_i, \qquad (3)$$

which accounts for all possible actuated/non-actuated/under-actuated cases through the selection matrix S_i. The cases of joints that are passively actuated (spring-loaded) are also accounted for in the general formulation by writing the corresponding terms in τ_i as a function of q_i (e.g. Hooke's law or a deformation law [21]). Finally the general equation (3) also accounts for all possible fixed-base/floating-base cases through the inclusion (or not) of the the base configuration in q. We can thus use this equation to model humanoid robots (possibly multiple ones), as well as floating objects or articulated passive environments (e.g. doors).

The n entities physically interact with each other by exchanging contact forces that come in action-reaction pairs.

Each contact force $f_{i,p}$ in equation (3) is either applied by the fixed inertial environment, or is applied 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 Ψ such that we can combine all the equations of motion of the n entities in one single equation:

$$M(q)\ddot{q} + N(q,\dot{q}) = J_0^T F^0 + (J_1 - \Psi^T J_2)^T F^1 + S\tau \quad (4)$$

where q, τ, N denote respectively the stacked vectors of configurations, actuation torques, and nonlinear effects respectively, M and S are the stacked block-diagonal matrices of M_i and S_i block-diagonal elements respectively, F^0 is the stacked vector of fixed inertial environment contact forces with corresponding stacked block-diagonal Jacobian matrix J_0, F^1 the stacked vector of single representatives of action-reaction pairs, and J_1 and J_2 are Jacobian matrices corresponding to the contact points between the different entities of the multi-robot system (see detailed derivations of these matrices in [14]). It can be also demonstrated that the no-slip contact constraints between the different entities and with the environment can be written as $J_1\dot{q} = \Psi J_2\dot{q}$ and $J_0\dot{q} = 0$.

Once the combined dynamics are formed, the formulation is the same as a single-robot QP controller, with the addition of the constraints for contacts and collision-avoidance between the entities and denoting λ^0 and λ^1 as the coefficients along the linearized friction cone generators of F^0 and F^1 respectively:

$$\min_{\ddot{q}, \tau, \lambda^0, \lambda^1} \sum_k w_k ||\ddot{y}_k - \ddot{y}_k^d||^2, \quad (5)$$

$$\text{subject to:} \begin{cases} \text{combined dynamics equation} \\ \text{contacts no-slip (with environment)} \\ \text{forces within friction cones} \\ \text{joint position, velocity, torque limits} \\ \text{collision avoidance} \\ \text{contacts no-slip (between robots)} \end{cases} \quad (6)$$

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 CoM) or benefit from the reduced representation to imply desired behavior 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 adaptive human-to-humanoid motion retargeting.

In this paper, we consider three entities ($n = 3$): humanoid robot, human (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 is simulated by equation (3). 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 [22], 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 Xsens biomechanical model with 23 segments, connected via 22 three-dimensional joints. The motion capture system provides: positions/orientations of link and joint angles, along with their first and second 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 that is compatible with our multi-robot QP software. The limb lengths and masses are parametrized as a function of the subject's height and weight, based on average anthropometric coefficients [23], with further customization possible to match a human's specific measurements.

B. Human motion reconstruction

To accurately reconstruct human motion with our MRQP, we set high-weight motion tracking objectives on the human "robot", which take motion capture data as time-varying setpoints.

We use a configuration task to track the joint angles of the human's motion. On top of that, we set human motion tracking objectives on end effector pose, 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 a good approximation of the motion capture data on our simulated human.

To integrate the human motion reconstruction into a multi-robot QP, 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 tracks the real human's motion well by setting human motion tracking task weights to be several orders of magnitude higher than robot 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 considering the human's dynamics (QP2). The solution for the human motion $\ddot{q}_{\text{human}}$ from QP1 is set as an equality constraint on the human model's motion in QP2. This is the "hard priority" approach.
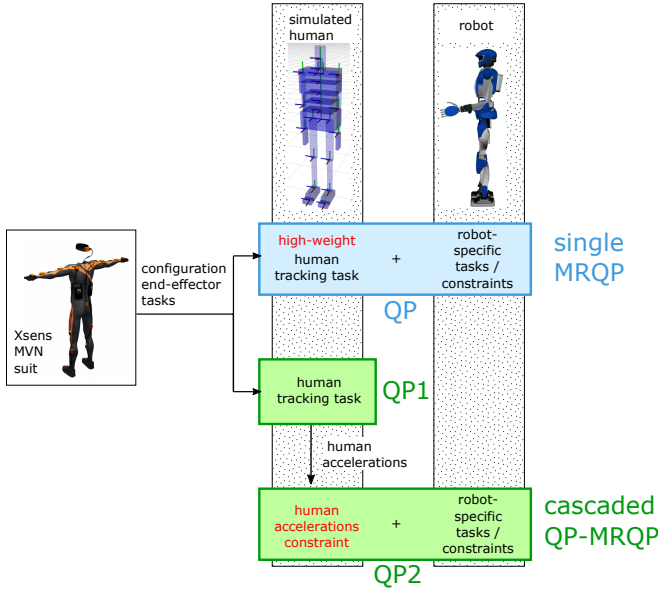
Fig. 2. Diagram showing the difference between our two methods for human motion reconstruction. In both methods, the motion capture data is mapped onto our parametrized 22-joint human model, which is then used to compute the robot's controls.

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 tracking of the real task performed by the human) 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. Accumulated numerical errors caused drift between the simulated human's states in QP1 and QP2, which meant that large $\ddot{q}$ commands would cause differences in end effector motions large enough for other 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 multi-robot QP 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 multi-robot QP 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 to the MRQP controller.

C. Robot objectives

The quadratic objectives in the QP 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 CoM bound objective, which sets $\ddot{p}_{\mathrm{CoM}}^{\mathrm{des}}$ to ensure that the ground projection of the robot's CoM does not go outside the convex hull of its support polygon, defined in hyperplane representation $\{p_{\mathrm{CoM}} \mid Ap_{\mathrm{CoM}} \geq b\}$. This is implemented as a damping behavior that slows down the CoM as it nears the boundaries of the convex hull:

$$\dot{d} + \ddot{d}\Delta_{dt} \geq -\xi \frac{d - d_s}{d_i - d_s} \tag{7}$$

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

We include a full-configuration 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 helps generate more natural-looking movements.

Contact changes are handled via the method described in [24], and we assume that the contact changes are simultaneous for the human and robot.

D. Interaction objectives

The tasks described above ensure that the robot is individually stable. 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, to avoid unrealistic behaviors in which the humanoid leans on/pushes other entities excessively.

We chose this motion because it is frequently performed in manufacturing environments between two human collaborators.

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 from the human's reference frame to their end effector $T_{\mathrm{Human\ ref}}^{\mathrm{Human\ hand}}$, which is then projected

from the robot's reference frame to $T_{\text{Robot ref}}^{\text{Robot hand}}$. The reference frames $\{T_{\text{World}}^{\text{Human ref}}, T_{\text{World}}^{\text{Robot ref}}\}$ are chosen to be at the midpoint between feet at initialization, with x, y, z axes pointing in the forward, left, and up directions respectively. We defined a "mirroring" operation $T^{\text{Mirrored}} = \mathbf{Mirror}(T^{\text{Human}})$ on position and orientation:

$$q_{\text{H}} = \{w, x, y, z\} \rightarrow q_{\text{mirrored}} = \{z, y, x, w\} \tag{8}$$

$$x_{\text{mirrored}} = x_{\text{H}} \tag{9}$$

$$z_{\text{mirrored}} = z_{\text{H}} \tag{10}$$

$$y_{\text{mirrored}} = \begin{cases} y_{\text{H}} & \text{if no contact} \\ y_0^{\text{R}} + (y_0^{\text{H}} - y_{\text{H}}) & \text{if contact} \end{cases} \tag{11}$$

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

$$(T_{\text{W}}^{\text{R hand}})_{\text{des}} = \mathbf{Mirror}(T_{\text{H ref}}^{\text{H hand}}) \cdot T_{\text{W}}^{\text{R ref}} \tag{12}$$

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

IV. EXPERIMENTS

To demonstrate the effectiveness of our pHRI framework, we performed several experiments in full 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 [22], as the second partner is replaced by the robot in our simulations. Obviously this is not completely analogous to testing a pHRI controller in real life, and 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
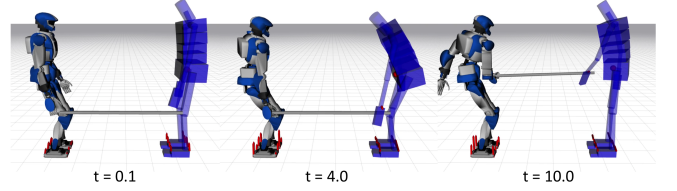

Fig. 3. Steps of Experiment 1. See attached video for a better view.
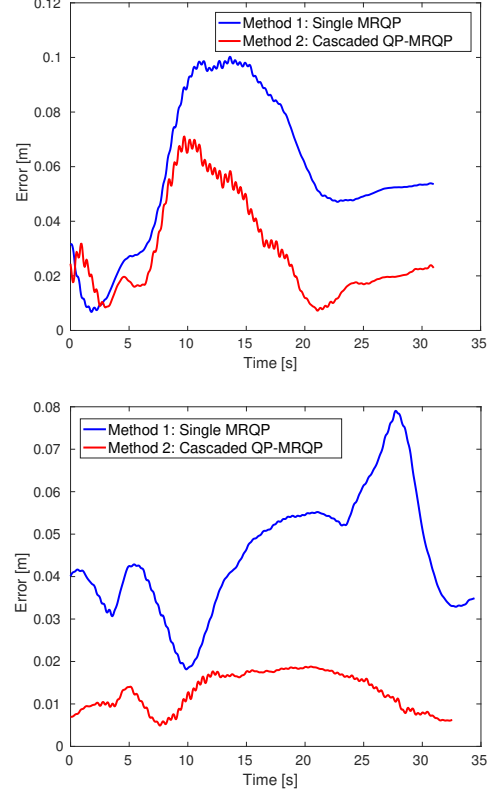

Fig. 4. Tracking errors: difference of end effector (top) and CoM (bottom) position between human motion capture data and our simulated human model in Experiment 1. Results shown for both methods of human motion reconstruction. As expected, the Cascaded QP-MRQP method tracks closer. The Single MRQP method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

amount of force the human needs to exert by encoding an assumption about the desired motion in the "mirroring" task. In the next section, we will demonstrate a behavior that is only possible with pHRI formulations that consider the whole-body human model.

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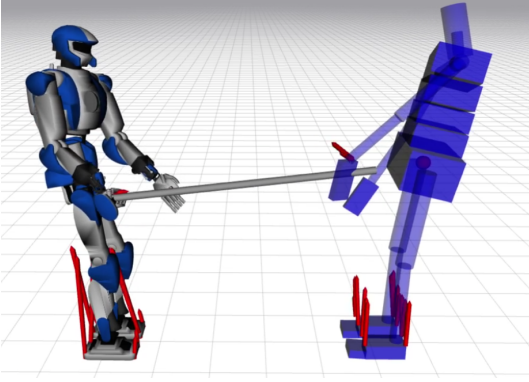
Fig. 5. Robot assisting human with balance in Experiment 2. Forces on robot and human shown in red.
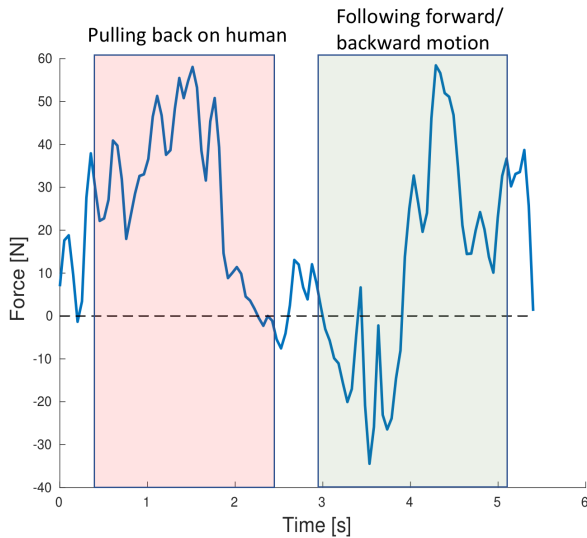

Fig. 6. Interaction forces (forces felt by the robot) in the x-direction (forward/backward) on the robot's right end effector during Experiment 2. Our controller generates realistic interaction forces that keep the collaborative task stable.

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 utilize a less complete representation of the human would have a hard time cleanly formulating the robot's assistive behavior in this situation.

V. LIMITATIONS OF TESTING pHRI IN SIMULATION

Our experiments were done in simulation, using 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 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. This assumes that the robot knows the goals of the task *a priori*, e.g. symmetric motion with the human.

To extend our framework into 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 $\{\ddot{q}_{\text{des}}, \tau_{\text{des}}, \lambda_{\text{des}}\}$; in our simulation experiments, the HRP-4 was controlled with joint position control by integrating $\ddot{q}_{\text{des}}$ 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 τ_{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 highly 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 tasks, and infer the human's intended task online. Thereafter, the robot will be able to actively help the human achieve the task with task-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, we can use neuromuscular human simulations which simulate individual muscles and tendons [26] and thus provide much better predictions. However, these simulations are too computationally expensive to run in real-time. A surrogate model for neuromuscular simulation (e.g. a neural network) can be used as an inexpensive, real-time approximation to the full model and run in real-time as part of the QP controller.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[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] 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.

[6] 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.

[7] 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.

[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] 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.

[11] 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.

[12] 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.

[13] 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.

[14] K. Bouyarmane, J. Vaillant, K. Chappellet, and A. Kheddar, "Multi-robot and task-space force control with quadratic programming," 2017.

[15] 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.

[16] P. Maurice, "Virtual ergonomics for the design of collaborative robots," Ph.D. dissertation, Paris 6, 2015.

[17] 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.

[18] S. Feng, "Online hierarchical optimization for humanoid control," 2016.

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

[20] K. Bouyarmane and A. Kheddar, "Humanoid robot locomotion and manipulation step planning," *Advanced Robotics*, vol. 26, no. 10, pp. 1099–1126, 2012.

[21] K. Bouyarmane and A. Kheddar, "Fem-based static posture planning for a humanoid robot on deformable contact support," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*. IEEE, 2011, pp. 487–492.

[22] "Xsens MVN," http://www.xsens.com/products/xsens-mvn/.

[23] D. B. Chaffin, G. Andersson, B. J. Martin *et al.*, *Occupational biomechanics*. Wiley New York, 1999.

[24] 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.

[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.