



HAL
open science

Robust Real-time Whole-Body Motion Retargeting from Human to Humanoid

Luigi Penco, Brice Clément, Valerio Modugno, Enrico Mingo Hoffman, Gabriele Nava, Daniele Pucci, Nikos Tsagarakis, Jean-Baptiste Mouret, Serena Ivaldi

► **To cite this version:**

Luigi Penco, Brice Clément, Valerio Modugno, Enrico Mingo Hoffman, Gabriele Nava, et al.. Robust Real-time Whole-Body Motion Retargeting from Human to Humanoid. HUMANOIDS 2018 - IEEE-RAS 18th International Conference on Humanoid Robots, Nov 2018, Beijing, China. hal-01895145

HAL Id: hal-01895145

<https://hal.science/hal-01895145v1>

Submitted on 14 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Robust Real-time Whole-Body Motion Retargeting from Human to Humanoid

L. Penco¹, B. Clement¹, V. Modugno²,
E. Mingo Hoffman³, G. Nava³, D. Pucci³, Nikos G. Tsagarakis³, J.-B. Mouret¹, S. Ivaldi¹

Abstract— Transferring the motion from a human operator to a humanoid robot is a crucial step to enable robots to learn from and replicate human movements. The ability to retarget in real-time whole-body motions that are challenging for the humanoid balance is critical to enable human to humanoid teleoperation. In this work, we design a retargeting framework that allows the robot to replicate the motion of the human operator, acquired by a wearable motion capture suit, while maintaining the whole-body balance. We introduce some dynamic filter in the retargeting to forbid dangerous motions that can make the robot fall. We validate our approach through several experiments on the iCub robot, which has a significantly different body structure and size from the one of the human operator.

I. INTRODUCTION

Current approaches for motion generation and control in humanoid robots essentially rely on optimizers and planners that search for the optimal sequence of joint commands (typically joint torques or velocities) according to an objective function that fulfills multiple tasks under several constraints [1], [2]. For instance, it is common to have tracking tasks in the objective function to specify that the end-effectors should follow a desired trajectory [3]. Unfortunately, designing and tuning the desired trajectories to realize complex tasks is time-consuming and often requires the expert knowledge of the controller/planner and of the humanoid kinematics/dynamics, which prevents an easy deployment of new tasks [4].

An alternative is to follow an imitation approach: a human performs a movement and the robot attempts to reproduce it [5]. Kinesthetic teaching is now a mature approach for robotic arms and industrial manipulators equipped with torque sensing: it allows the human operator to show the robot the desired trajectories by physically manipulating the robot links. While this approach is relatively straightforward for robotic arms, it can be hardly done for humanoids to demonstrate whole-body movements (because it is not possible to physically guide all the robot links at once while ensuring the robot balance). Motion retargeting in this sense



Fig. 1: Teleoperation of the iCub robot by an operator equipped with the XSens motion capture suit.

represents the extension of the kinesthetic teaching concept at a whole-body level.

However, real-time retargeting (or teleoperation) of entire whole-body movements is challenging for humanoids. First, direct mapping is not possible because of important differences in kinematics (e.g., joint limits, limb lengths) and dynamics (e.g., mass distribution, inertia); second, the robot needs to maintain its balance while imitating the human, so it has to trade-off between imitation and feasibility/safety. Third, since we do not know a priori which motion is going to be retargeted on the robot by the operator, it is not possible to tune offline the controller and/or the retargeting parameters for specific motions: rather, we need to provide a generic solution that is able to handle a variety of motions.

Our requirement is to have a retargeting framework that can handle a variety of whole-body motions shown by the human operator, and generates appropriate robust motions (not only feasible) for the humanoid robot. This work is motivated by our application in the project AnDy [6], where we would like to show directly to the humanoid human-like movements that represents demonstrations of collaborative policies. In this paper we present our current developments to teleoperate the iCub robot (see Figure 1).

The paper is organized as follows. In Section II we discuss the previous works on retargeting and teleoperation

*This work received funding European Commission through the project H2020 AnDy (GA no. 731540) and CogIMon (GA no 644727), and from the European Research Council (ERC) (GA no. 637972, project “ResiBots”).

¹ Inria, CNRS, Université de Lorraine, LORIA, Nancy, France, {luigi.penco, brice.clement, jean-baptiste.mouret, serena.ivaldi}@inria.fr

² Università La Sapienza, Roma, {modugno@diag.uniroma1.it

³ Istituto Italiano di Tecnologia, Genova, {enrico.mingo, gabriele.nava, danielle.pucci, nikos.tsagarakis}@iit.it

of humanoids. In Sections III we present an overview of the framework and in Section IV the corresponding retargeting method. We discuss our teleoperation experiments in Section V and outline the future works in Section VI.

II. RELATED WORK

Human-to-humanoids motion retargeting is an active research field. The first approaches to the retargeting problem consisted in attaching virtual markers to the robot and applying the inverse kinematics to adjust the human reference motion to fit the robot constraints [7],[8],[9].

Several balance controllers have been then proposed for maintaining dynamic stability, since the mass distribution often differs between the human and the robot [10], [9]. Dariush *et al* [10] considered only upper body movements and used task space control with a separate ZMP-controller to ensure balance. Yamane presented a control-based approach to imitate human motions with a force-controlled robot, generating online the joint trajectories under the assumptions that the feet maintain ground contact [11].

Recently, multi-task whole-body QP controllers were applied to retargeting. Otani and Bouyarmane [12] retargeted onto the HRP-4 robot in full-body dynamics simulation a bag retrieval, a door opening and a box lifting task. Their method relied on a-priori knowledge of the contact events and could not be formulated on-line without an off-line pre-processing of the recorded human motion sequence. Kanajar *et al* [13] retargeted a challenging multi-contact motion (climbing over an obstacle) with the Coman robot: they formulated the tasks based on the observed contact states to generate keyframes for the contact points, then optimized according to the robot stability and mechanical constraints. A multi-robot QP controller was also proposed to simulate human-robot physical interaction by retargeting human motions on a simulated human model [14]. In these works, however, QP controllers were used to retarget motions offline and they were specifically tuned to the known demonstrated motions.

The first online retargeting works were for the majority related exclusively to upper body movements [15],[16],[17]. Frequently, the retargeting of the upper-body joints, which is important for manipulation, is done independently from the one of the legs, crucial for balancing and locomotion [10],[18]. At KIST for example, they combined their upper body teleoperation framework [16] with a real-time walking pattern generator [19] to retarget also the walking [18].

In the last few years real-time retargeting of general whole-body motions has been addressed. Koenemann [20] has implemented a teleoperation of a physical robot (NAO) considering also changes in the contact support in real-time. His method is based on a simplified human model and an approximate posture mapping followed by a posture stabilization that allow transitions between double and single stances phases. However, this stabilization is not a robust solution for more complex humanoid robots that are more challenging to maintain stable during the motion, such as the iCub. Furthermore, the center of mass tracking formulation and the simplified model do not take into account the

retargeting of torso movements - we will address this in our approach in Section IV.

Ishiguro *et al* [21] proposed a secured system to teleoperate the robot LIP safety constraints. In their work they conducted some experiments involving dynamic upper-body and leg motions onto the legged robot CHIDORI and humanoid robot JAXON.

Off-line methods have been proposed as well to reconstruct the human motion within the physical constraints imposed by the retargeted subject kinematics and dynamics. Ayusawa and Yoshida [22] proposed a simultaneous morphing parameter identification and motion optimization. In [23] Borno *et al* used instead a LQR-tree formulation to transfer the motion between 3D realistic human models and adapting it to the different body shapes.

Very efficient real-time methods that can reshape a dynamic motion demonstrated by a human and adapt the dynamics of the human to the dynamics of the robot, use an inverse dynamics control scheme with a quadratic programming optimization solver [24], [25]. However, even if they successfully handle dynamic overshoots due to large angular momentum variation, they require a suitable walking pattern generator to handle the linear momentum.

The model predictive control (MPC) instead, is able to deal with great linear momentum variations by considering the full temporal evolution of the system. The optimal trajectory is then updated real-time with sensor measurements. MPC enhances the expressivity capabilities of the motion generator and gives more compliance to the robot that can react to external unexpected contact events [26], [27]. The only drawback of using an MPC approach in a teleoperation framework is the underlying computational cost.

Our approach is based on an inverse kinematics (IK) control scheme with a QP optimization solver. In its simplicity, it is able to guarantee the stability of the teleoperated robot even when the retargeted movements challenge its balance. We use a ZMP retargeting correction that provides robustness and allows generality of the retargeted motion.

We evaluate our framework with several experiments with the iCub humanoid robot being teleoperated by a human operator (see also attached video). Differently from [17] that teleoperated only the upperbody of the iCub, we control its entire body and we are able to retarget whole-body movements that are challenging for the balance (e.g., squat).

We formerly explored an alternative approach based on torque control that allows physical interaction with the teleoperated robot¹. The framework was built upon the CoDyCo torque controller [28] that computes the joint torques that minimize the forces at the contacts [29]. The reason for choosing a desired set of contact forces is that we can use the contact wrench as a fictitious control input of the centroidal momentum dynamics. Instantaneous values for forces are computed so as to follow a desired trajectory of the center of mass and to reduce the system's angular momentum. The

¹Some experiments with the torque controller are shown in this video: <https://youtu.be/-1b0n5shuxg>

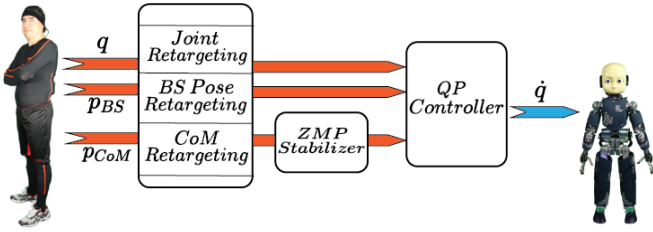


Fig. 2: Teleoperation pipeline.

torque control allows a more flexible interaction with the robot, in particular it enables direct physical interaction between the human and the robot that is teleoperated. However, in the case of iCub, the absence of joint torque sensing² and saturation of force/torque sensors limit the torque control loop, to the point that velocity control seems more appropriate for quasi-static movements. Moreover, the structure of the torque controller ensures stability for complex motions at the price of a reduced mimicry and a notable delay³. In this paper we overcome these limits with a novel framework for online motion retargeting with a velocity-based QP controller that we describe in the next section.

III. FROM MOTION CAPTURE TO ROBOT MOTION

A. Motion capture

The first step for a motion retargeting technique is to track the human pose. Recent developments in human motion capture allow now high-fidelity and high-frequency tracking data. Motion-capture is widely used nowadays in various fields including physiotherapy, surveillance, computer graphics and foremost in the cinema, using external cameras or wearable sensors. For our experiments we used the Xsens MVN system [31]. It is a wearable system consisting of 17 IMUs, providing a real-time estimation of the human posture.

Once the data is acquired from the motion capture system, it can be mapped to feasible corresponding values for the robot that are set as references for the multi-task QP controller (see Fig. 2).

B. QP controller based on stack-of-tasks

We opted for a velocity-based QP controller based on OpenSoT [32], an open-source library implementing QP controllers based on the *stack-of-tasks* [33]. Nowadays, QP controllers have become widespread thanks to their flexibility and various formulations have been proposed in the literature, [34], [12], [35], which take into account joint velocity, acceleration and torque control as well as contact forces. The general form of QP-based controllers is:

$$\min_x \|Ax - b\|_W^2 \quad \text{s.t.} \quad l \leq Cx \leq u \quad (1)$$

where x are the controlled variables (e.g. joint velocities), A is the Jacobian task matrix, b is the desired task to perform,

²Joint torques are estimated online with a model-based approach that exploits the sensor readings from the force/torque sensors distributed on the robot [30].

³These limitations can be observed in the video of the teleoperation with the CoDyCo torque controller: <https://youtu.be/-1b0n5shuxg>

W is a positive definite weight matrix, C is the constraint matrix and finally l (lower) and u (upper) are constraints. QP controllers permit to handle various type of constraints that, depending on the chosen formulation, may include robot dynamics, friction cones, self-collision avoidance, joint limits and many more (see for example [35]).

For the teleoperation this kind of controllers is appropriate since they allow to take into account both Cartesian tasks (body segment positions) and Joint space tasks while satisfying all the robot constraints. Moreover, the tasks can be specified both in a hard and soft priority fashion.

IV. RETARGETING METHOD

The QP controller allows to define Cartesian tasks and a postural task together with their subtasks. Since the iCub body dimension is close to the one of a 5 years old child, there is a great difference with the size of the human operator wearing the xSens suit. For this reason, the direct retargeting of Cartesian tasks that are expressed in the global frame does not make any sense. We tried then to assign Cartesian tasks related to the relative positions of the feet with respect to the hips and of the hands with respect to the shoulders as done in [20]. However, to do so we should accurately measure the ratio between the lengths of the limbs of the robot and of the human, and when starting the streaming of the data from Xsens, the 3D Xsens skeleton initial body orientation with respect the global frame should also match precisely the one of the robot. Furthermore, the MVN Xsens skeleton estimation is affected by some noise that makes the global frame drift on the ground over time. Hence for long teleoperations the retargeting of Cartesian tasks could be compromised because of the change of orientation of the global frame.

For these reasons, we opted for retargeting only the joint angles and simplifying the initialization process for the teleoperation together with the reliability. More specifically, we do not retarget all the joint angles but we separate the postural task in several subtasks: $head_{sub}$ (neck joints), $torso_{sub}$ (torso joints), $lArm_{sub}$ (left arm joints), $rArm_{sub}$ (right arm joints).

We additionally considered a Cartesian task for the ground position of the CoM of the robot (com_{sub}) and a Cartesian task for the height and orientation of the floating base ($base_{sub}$). The global position of the feet is also taken into consideration, in order to keep each foot in contact with the ground whenever it is a support link ($lFoot$ and/or $rFoot$). The posture of the legs is retargeted indirectly through the CoM and floating base tasks. Our resulting selected stack of tasks for the teleoperation is the following:

$$stack = (lFoot + rFoot + head_{sub}) / (com_{sub} + base_{sub} + torso_{sub} + lArm_{sub} + rArm_{sub});$$

where $T_a + T_b$ means that the tasks T_a and T_b are in a *SOFT (Weighted) Priority* relation while T_a / T_b means the tasks T_a and T_b are in a *HARD (Null-Space) Priority* relation.

In single support mode, the task of the foot that is not the support one can be removed and a leg postural task can

added for the lifted leg. However, in this paper we only show experiments in double support.

As constraints, here we mostly care for joint limits and joint velocity limits; the chosen formulation is based on joint velocity.

A. Joint Angles Mapping

The Xsens model of the human has 66 DoFs that correspond to 22 spherical joints. Except for the torso, the other spherical joints can be easily visually assigned to the corresponding ones of the iCub. Less intuitive is the mapping between the individual joints composing the spherical one. Through several tests on simulation we identified the right mapping reported in Fig. 3. For the torso, we used an approximate mapping. We considered the joints that are the most involved in the motion that are jL4L3, j1T12, and jT9T8, corresponding to the vertebrae going from the second lowest lumbar vertebra (L4) to the thoracic vertebra at the level of the breastbone (T8). The joints jT1C7 and jL5S1 (together with the other joints not reported in Fig. 3) have been considered in some tests as well but their contribution is negligible. Each joint value of the torso is obtained from the sum of the corresponding rotations of the joints jL4L3, j1T12, and jT9T8. Except for the yaw of the torso, this mapping is just approximate since the actual angle between the trunk and the hips is generally lower than the one given by the sum of the three joints. However, we noticed that the difference is not so relevant, that is reasonable if we consider that most of the inclination of the torso is generated by the hip movement.

After the identification of the human-iCub joints association, we retarget the variation of the joint angles from the starting posture

$$\Delta q_{i,H} = q_{i,H} - q_{start,H} \quad (2)$$

$$q_{i,R} = q_{start,R} + \Delta q_{i,H} \quad (3)$$

where $q_{start,R}$, $q_{start,H}$ and $q_{i,R}$, $q_{i,H}$ are the human and robot joint position vectors at the start and at each time step respectively and $\Delta q_{i,H}$ is the joint position deviation vector from the starting posture of the human in each time step.

B. Center of Mass Tracking

To track the center of mass we present an improvement of the method proposed in [12],[20], where the robot tracks a normalized offset (denoted as o) between its support feet. Let us consider the 2D ground projection of the human CoM p_{CoM} . The position p_{CoM} with respect to an arbitrary support foot (in our case the left foot) is projected onto the line connecting the two feet. The result is then normalized to get a value in between 0 and 1.

$$o = \frac{(p_{CoM,H} - p_{LFoot,H}) \cdot (p_{RFoot,H} - p_{LFoot,H})}{\|p_{RFoot,H} - p_{LFoot,H}\|^2} \quad (4)$$

So when the human is in a symmetric pose in double support mode the offset has a value around 0.5 and when the human stands on a single foot the offset is either 0 (left foot) or 1 (right foot). The robot 2D CoM ground projection is then

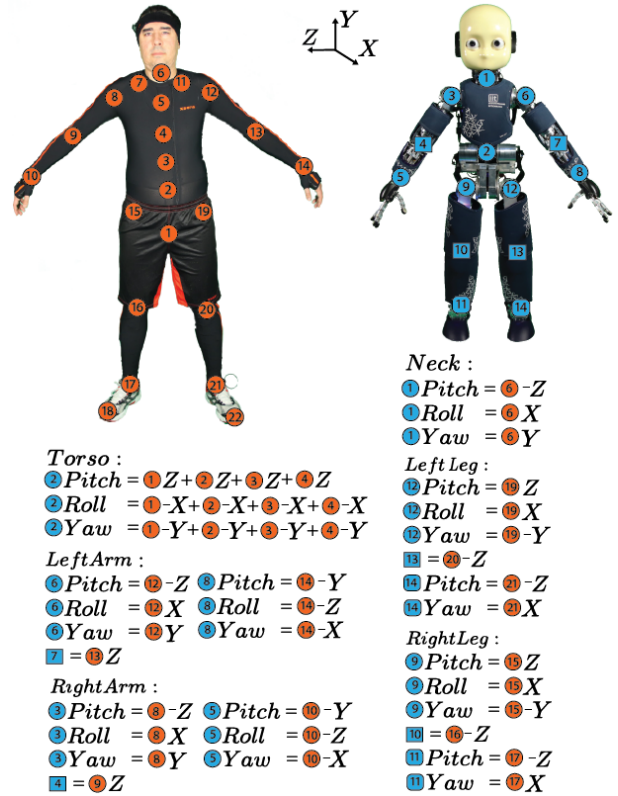


Fig. 3: Mapping between the Xsens joints and the iCub joints.

reconstructed on the line connecting the two feet by means of this offset value.

$$p_{CoM,R} = p_{LFoot,R} + o(p_{RFoot,R} - p_{LFoot,R}) \quad (5)$$

To retarget also changes of the human CoM that are not on the line connecting the two feet, we first measure the maximum backward and forward CoM displacement of the human and of the robot over their support polygon (with the origin lying on the feet line), i.e. $\delta_{CoM_{back,H}}$, $\delta_{CoM_{forw,H}}$ and $\delta_{CoM_{back,R}}$, $\delta_{CoM_{forw,R}}$ respectively. Then, we retarget the human CoM displacement $\Delta_{CoM,H}$ (comprised within the minimum value $-\delta_{CoM_{back,H}}$ and maximum value $\delta_{CoM_{forw,H}}$) onto the robot, by computing the offset

$$o' = \frac{(\Delta_{CoM,H} - (-\delta_{CoM_{back,H}}))}{(\delta_{CoM_{forw,H}} - (-\delta_{CoM_{back,H}}))} \quad (6)$$

from which we get the corresponding robot CoM displacement

$$\Delta_{CoM,R} = o'(\delta_{CoM_{forw,R}} + \delta_{CoM_{back,R}}) - \delta_{CoM_{back,R}} \quad (7)$$

then we apply this displacement in the orthogonal direction of the line connecting the two feet of the robot.

C. Controlling the Floating Base

We can consider the pelvis as the floating base for the human. To control the height of the floating base of the robot

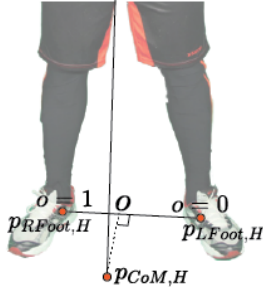


Fig. 4: Determination of the normalized offset from the ground projection of the CoM $p_{CoM,H}$ and the feet positions of the human.

we consider the deviation of the height of the human pelvis $\Delta base_{i,H}$ from the starting value over time

$$\Delta base_{i,H} = base_{i,H} - base_{start,H} \quad (8)$$

In order to follow the human motion, we expect the corresponding deviation of the floating base of the robot $\Delta base_{i,R}$ to be proportional to the one of the human

$$\Delta base_{i,H} = \alpha \Delta base_{i,R} \quad (9)$$

$$\alpha = \frac{h_{base,R}}{h_{base,H}} \quad (10)$$

where α is the ratio between the height of the floating base of the robot and of the pelvis of the human when in N-pose⁴. Then we can calculate the height of the robot base at each time step as:

$$base_{i,R} = base_{start,R} + \Delta base_{i,R}. \quad (11)$$

We also retarget the change of orientation of the floating base in a similar way, by computing the roll, pitch and yaw from the quaternion information given by the motion capture system.

D. ZMP Retargeting Correction

During whole body teleoperation of humanoid robots, disastrous crashes may occur if the desired CoM trajectories recorded from the human do not ensure the balance of the controlled robot when retargeted.

To this scope, we propose a QP-based “preprocessor” that adjusts in real-time the desired commanded CoM to satisfy constraints that represent a condition for dynamic balance. In order to achieve a stable CoM trajectory we employ the linear inverted pendulum model (LIPM) in combination with the Zero Moment Point (ZMP) criterion.

The ZMP is represented with a point on the ground plane where the tipping moments, generated by the gravity and the inertial forces, are equal to zero. A humanoid robot keeps its balance if the ZMP is contained inside the support polygon of the robot.

⁴N pose is a resting pose, where the human stands with lowered arms close to its body.

Through the LIPM model it is possible to establish a simple relation between the ZMP and the CoM dynamics:

$$\ddot{p}_{CoM} = \frac{g}{h} (p_{CoM} - p_{ZMP}) \quad (12)$$

where g represents the gravity acceleration, h is the height of the inverted pendulum and $p_{CoM} = (x_{CoM}, y_{CoM})^T$, $p_{ZMP} = (x_{ZMP}, y_{ZMP})^T$ represent respectively the CoM and ZMP positions of the LIPM on the horizontal plane. By employing the equation (12) is possible to cast a QP problem to obtain an optimal correction of the desired CoM that satisfies the balance condition on the humanoid

$$\begin{aligned} \min_{p_{ZMP}} \quad & (\dot{p}_{CoM}^{des} - \dot{p}_{CoM})^T R (\dot{p}_{CoM}^{des} - \dot{p}_{CoM}) \\ \text{s.t.} \quad & \dot{p}_{CoM} = \dot{p}_{CoM}^{t-1} + \frac{Tg}{h^{t-1}} (p_{CoM} - p_{ZMP}) \\ & lb_{SP} < p_{ZMP} < ub_{SP} \end{aligned} \quad (13)$$

where \dot{p}_{CoM}^{des} is the desired CoM velocity, T is the sampling time, \dot{p}_{CoM}^{t-1} , h^{t-1} are respectively the last CoM velocity and the last CoM height measured from the robot and lb_{SP} and ub_{SP} are the lower and upper bound of the support polygon of the robot. Updating the height of the LIPM at each time step with the actual robot CoM z position provides a better model to estimate the ZMP position and consequently, a more accurate correction.

V. EXPERIMENTS

We set up two experiments to validate our approach. First, through dynamics simulations on the simulated robot in Gazebo, we show how the ZMP retargeting correction is essential to retarget a motion that is dynamically stable onto the robot. Then, we tested our framework by teleoperating the real robot iCub in real-time. Videos of our experiments with the real robot can be seen at https://youtu.be/CjLQu_6ifAE and <https://youtu.be/iZVAacyvYhM>.

Simulated robot – We selected three kind of motions to show the efficacy of our method: squat motion, hip roll exaggerate motion and a grasping motion involving some torso and leg movements (see Figure 7). Figure 5 shows how the ZMP associated to the retargeted motion without the correction lies outside the support polygon, making the robot fall inevitably (that is why we show this in simulation). This is due both to mechanical limitations of the robot, which cannot achieve the same CoM displacement given the retargeted joint values (that might go beyond the robot joint limits), and to the generation of some momentum different from the human one that makes the desired robot CoM trajectory unstable. Hence, the ZMP trajectory is corrected in real-time to stay inside the support polygon and the CoM trajectory is modified accordingly (see Figure 6).

Real robot – We teleoperated the robot trying to move all its links to show the effectiveness of our framework. During the teleoperation the ZMP position of the robot always lies inside the support polygon as expected (see Figure 8). The robot joint trajectories follow the retargeted values guaranteeing the mimicry of the human motion (see Figure 9). Even though the legs joint angles are not taken into

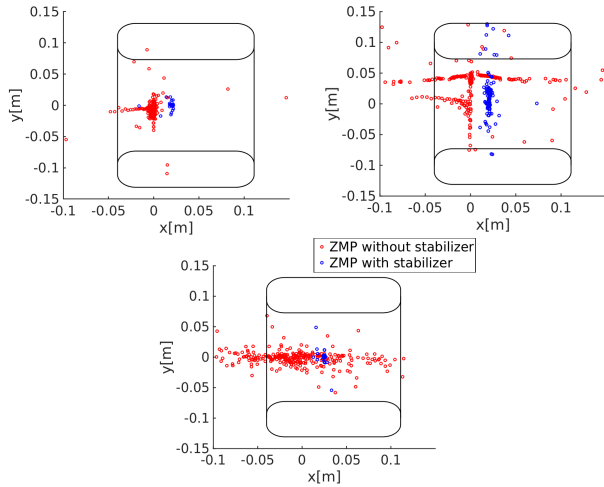


Fig. 5: ZMP position of the simulated robot without (red) and with (blue) the stability correction during the teleoperation while performing a squat motion (up-left), a hip roll exaggerate motion (up-right), a grasping motion involving some torso and leg movements (down).

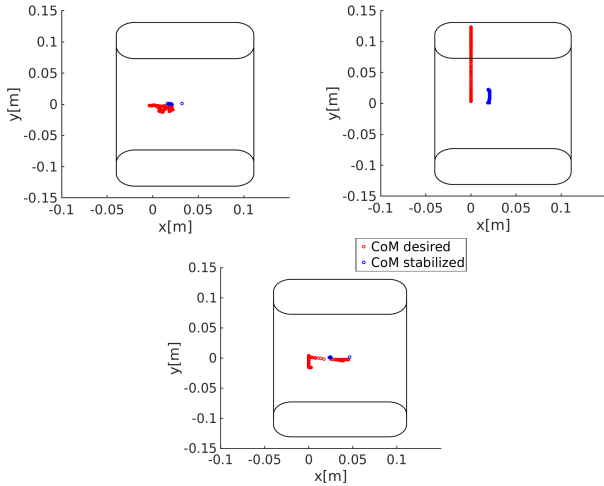


Fig. 6: CoM desired (red) and stabilized (blue) position of the simulated robot during the teleoperation while performing a squat motion (up-left), a hip roll exaggerate motion (up-right), a grasping motion involving some torso and leg movements (down).



Fig. 7: Snapshots of the teleoperated simulated robot and of the Xsens skeleton while performing a squat motion, a hip roll exaggerate motion and a grasping motion during the teleoperation.

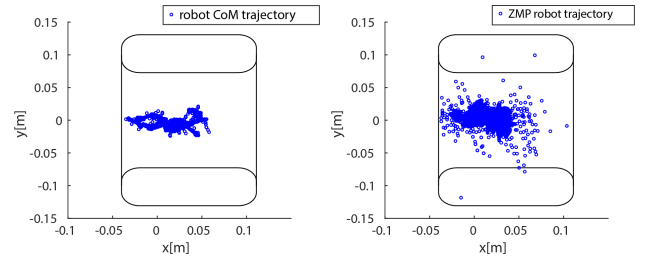


Fig. 8: iCub: CoM (left) and ZMP (right) position trajectories during the teleoperation experiment.

consideration in the stack, the retargeting of the height and of the orientation of the floating base makes the resulting legs joint trajectories close to the human ones. This framework allows to teleoperate the robot for a potentially unlimited amount of time. Even if the human performs some motions that are very challenging for the balance, the ZMP constraints introduced by our framework make the robot maintain its balance. The video attachment at <https://youtu.be/iZVAacyvYhM> shows that even if the operator is falling, the robot does not and keeps its balance.

VI. CONCLUSIONS

We proposed a framework to teleoperate the iCub robot, based on an inverse kinematics (IK) control scheme with a QP optimization solver. It allows a robust real-time re-targeting of generic whole-body motions. The generality is obtained thanks to the mapping of all the main human joints: not only the arms and the legs, but also the torso and the head. The CoM as well is retargeted in a way that does not restrict the range of possible retargeted robot motions. The robustness is given by a ZMP correction approach that guarantees the stability of the retargeted trajectory in double support. In the future we will overcome the double support limitation of this paper, integrate single support and teleoperation with locomotion.

ACKNOWLEDGMENTS

The authors would like to thank Augusto Cesar Francisco for his precious help when wearing the motion capture suit.

REFERENCES

- [1] B. Siciliano, L. Sciavicco, V. Luigi, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, 2011.
- [2] S. Ivaldi, J. Babič, M. Mistry, and R. Murphy, “Special issue on whole-body control of contacts and dynamics for humanoid robots,” *Autonomous Robots*, vol. 40, no. 3, pp. 425–428, 2016.
- [3] N. Mansard, O. Stasse, P. Evrard, and A. Kheddar, “A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks,” in *Advanced Robotics, 2009. ICAR 2009. International Conference on*. IEEE, 2009, pp. 1–6.
- [4] V. Modugno, U. Chervet, G. Oriolo, and S. Ivaldi, “Learning soft task priorities for safe control of humanoid robots with constrained stochastic optimization,” in *HUMANOIDS, 2016*, pp. 101–108.
- [5] B. Akgun, K. Subramanian, and A. L. Thomaz, “Novel interaction strategies for learning from teleoperation,” in *AAAI Fall Symposium: Robots Learning Interactively from Human Teachers*, 2012.
- [6] S. Ivaldi, L. Fritzsche, J. Babič, F. Stulp, M. Damsgaard, B. Graimann, G. Bellusci, and F. Nori, “Anticipatory models of human movements and dynamics: the roadmap of the andy project,” in *DHM, 2017*.

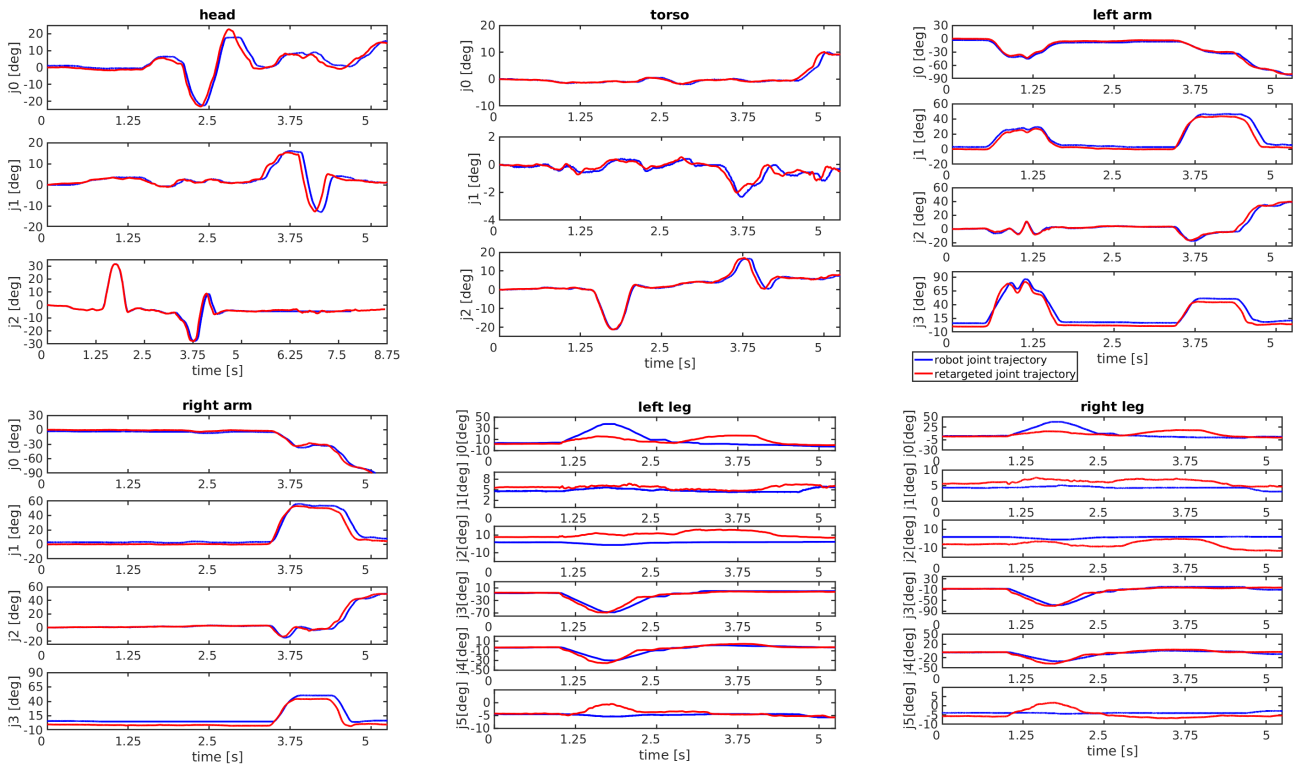


Fig. 9: iCub: joint trajectories (blue) compared to the retargeted values (red) during the teleoperation experiment.



Fig. 10: Snapshots from the teleoperation experiment. Remarkably, the robot does not fall when the human operator falls back (frame on the bottom-right).

- [7] N. S. Pollard, J. K. Hodgins, M. J. Riley, and C. G. Atkeson, "Adapting human motion for the control of a humanoid robot," in *ICRA*, 2002, pp. 1390–1397.
- [8] C. Ott, D. Lee, and Y. Nakamura, "Motion capture based human motion recognition and imitation by direct marker control," in *HUMANOIDS*, 2008, pp. 399–405.
- [9] K. Miura, M. Morisawa, S. Nakaoka, F. Kanehiro, K. Harada, K. Kaneko, and S. Kajita, "Robot motion remix based on motion capture data towards human-like locomotion of humanoid robots," *HUMANOIDS*, pp. 596–603, 2009.
- [10] B. Dariush, M. Gienger, A. Arumbakkam, Y. Zhu, B. Jian, K. Fujimara, and C. Goerick, "Online transfer of human motion to humanoids," *Int. J. Humanoid Rob.*, vol. 06, no. 02, pp. 265–289, 2009.
- [11] K. Yamane, S. O. Anderson, and J. K. Hodgins, "Controlling humanoid robots with human motion data: Experimental validation," in *HUMANOIDS*, 2010, pp. 504–510.
- [12] K. Otani and K. Bouyarmane, "Adaptive whole-body manipulation in human-to-humanoid multi-contact motion retargeting," in *HUMANOIDS*, Nov 2017, pp. 446–453.
- [13] P. Kanajar, D. G. Caldwell, and P. Kormushev, "Climbing over large obstacles with a humanoid robot via multi-contact motion planning," in *IEEE RO-MAN*, Aug 2017, pp. 1202–1209.
- [14] K. Otani, K. Bouyarmane, and S. Ivaldi, "Generating Assistive Humanoid Motions for Co-Manipulation Tasks with a Multi-Robot Quadratic Program Controller," in *ICRA*, 2018.
- [15] A. Brygo, I. Sarakoglou, N. Tsagarakis, and D. G. Caldwell, "Tele-manipulation with a humanoid robot under autonomous joint impedance regulation and vibrotactile balancing feedback," in *HUMANOIDS*, Nov 2014, pp. 862–867.
- [16] S. Y. Shin and C. Kim, "On-line human motion transition and control for humanoid upper body manipulation," in *IROS*, 2010, pp. 477–482.
- [17] L. Fritsche, F. Unverzag, J. Peters, and R. Calandra, "First-person tele-operation of a humanoid robot," in *HUMANOIDS*, Nov 2015, pp. 997–1002.
- [18] D. Kim, B.-J. You, and S.-R. Oh, *Whole Body Motion Control Framework for Arbitrarily and Simultaneously Assigned Upper-Body Tasks and Walking Motion*. Springer, 2013, pp. 87–98.
- [19] S. Hong, Y. Oh, D. Kim, and B. J. You, "Real-time walking pattern generation method for humanoid robots by combining feedback and feedforward controller," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 355–364, Jan 2014.
- [20] J. Koenemann, F. Burget, and M. Bénéwitz, "Real-time imitation of human whole-body motions by humanoids," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 2806–2812.
- [21] Y. Ishiguro, K. Kojima, F. Sugai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Bipedal oriented whole body master-slave system for dynamic secured locomotion with lip safety constraints," in *IROS*, 2017, pp. 376–382.
- [22] K. Ayusawa and E. Yoshida, "Motion retargeting for humanoid robots based on simultaneous morphing parameter identification and motion optimization," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1343–1357, Dec 2017.
- [23] M. A. Borno, L. Righetti, M. J. Black, S. L. Delp, E. Fiume, and J. Romero, "Robust physics-based motion retargeting with realistic body shapes," *Computer Graphics Forum*, vol. 37, pp. 6:1–12, July 2018.
- [24] O. E. Ramos, N. Mansard, O. Stasse, C. Benazeth, S. Hak, and L. Saab, "Dancing humanoid robots: Systematic use of osid to compute dynamically consistent movements following a motion capture pattern," *IEEE Robotics Automation Magazine*, vol. 22, no. 4, pp. 16–26, Dec 2015.
- [25] O. E. Ramos, L. Saab, S. Hak, and N. Mansard, "Dynamic motion capture and edition using a stack of tasks," in *2011 11th IEEE-RAS International Conference on Humanoid Robots*, Oct 2011, pp. 224–230.
- [26] N. Scianca, V. Modugno, L. Lanari, and G. Oriolo, "Gait generation via intrinsically stable mpc for a multi-mass humanoid model," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, Nov 2017, pp. 547–552.
- [27] N. Scianca, M. Cognetti, D. D. Simone, L. Lanari, and G. Oriolo, "Intrinsically stable mpc for humanoid gait generation," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Nov 2016, pp. 601–606.
- [28] F. Romano, G. Nava, M. Azad, J. Čamernik, S. Dafarra, O. Dermy, C. Latella, M. Lazzaroni, R. Lober, M. Lorenzini, *et al.*, "The codyco project achievements and beyond: Toward human aware whole-body controllers for physical human robot interaction," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 516–523, 2018.
- [29] F. Nori, S. Traversaro, J. Eljaik, F. Romano, A. Del Prete, and D. Pucci, "icub whole-body control through force regulation on rigid non-coplanar contacts," *Frontiers in Robotics and AI*, vol. 2, 03 2015.
- [30] M. Fumagalli, S. Ivaldi, M. Randazzo, L. Natale, G. Metta, G. Sandini, and F. Nori, "Force feedback exploiting tactile and proximal force/torque sensing," *Auton. Rob.*, vol. 33, no. 4, pp. 381–398, 2012.
- [31] "Xsens the leading innovator in 3d motion tracking technology." 2017. [Online]. Available: <http://www.xsens.com/products/xsens-mvn/>
- [32] E. M. Hoffman, A. Rocchi, A. Laurenzi, and N. G. Tsagarakis, "Robot control for dummies: Insights and examples using OpenSoT," in *HUMANOIDS*, 2017, pp. 736–741.
- [33] E. M. Hoffman, B. Clement, C. Zhou, N. G. Tsagarakis, J.-B. Mouret, and S. Ivaldi, "Whole-Body Compliant Control of iCub: first results with OpenSoT," in *ICRA 2018 - Workshop*, 2018.
- [34] A. D. Fava, K. Bouyarmane, K. Chappellet, E. Ruffaldi, and A. Kheddar, "Multi-contact motion retargeting from human to humanoid robot," in *HUMANOIDS*, Nov 2016, pp. 1081–1086.
- [35] K. Otani, K. Bouyarmane, and S. Ivaldi, "Generating assistive humanoid motions for co-manipulation tasks with a multi-robot quadratic program controller," in *ICRA*, 2018.