Real-time Imitation of Human Whole-Body Motions by Humanoids

Jonas Koenemann Felix Burget Maren Bennewitz

***Abstract*— In this paper, we present a system that enables humanoid robots to imitate complex whole-body motions of humans in real time. In our approach, we use a compact human model and consider the positions of the endeffectors as well as the center of mass as the most important aspects to imitate. Our system actively balances the center of mass over the support polygon to avoid falls of the robot, which would occur when using direct imitation. For every point in time, our approach generates a statically stable pose. Hereby, we do not constrain the configurations to be in double support. Instead, we allow for changes of the support mode according to the motions to imitate. To achieve safe imitation, we use retargeting of the robot's feet if necessary and find statically stable configurations by inverse kinematics. We present experiments using human data captured with an Xsens MVN motion capture system. The results show that a Nao humanoid is able to reliably imitate complex whole-body motions in real time, which also include extended periods of time in single support mode, in which the robot has to balance on one foot.**

I. Introduction

Nowadays, a variety of technologies exist that allow for highly accurate capturing of human motions with high frequency. The human data can, for example, be used to generate human-like motions for the high number of degrees of freedom of humanoid robots. By imitating captured human motions, humanoids can be tele-operated and also easily learn new motions.

However, direct imitation of captured movements is typically impossible, e.g., due to differences in the human and humanoid kinematics and the different weight distribution. Depending on the complexity of the motion, it can be challenging to generate feasible motions for the robot and ensure stable execution. Especially, when the human motion leaves the double support mode and contains support mode changes or even extended periods of time in single support, stafe imitation on the humanoid is difficult.

So far, a variety of approaches to imitation of human whole-body or upper body motions have been presented. Many of them rely on an offline step that performs optimization on the human data so as to adapt it to the robot's kinematic structure and constraints [1], [2], [3], [4], [5], [6]. On the other hand, several systems that allow for real-time imitation have been presented. Most of them focus on generating upper-body motions while the legs are neglected or mainly used for balancing [7], [8], [9], others do not consider changes of the support mode [10], [11].

All authors are with the Humanoid Robots Lab and the *BrainLinks-BrainTools* Cluster of Excellence, University of Freiburg, Germany. This work has been supported by the Research Training Group *Embedded Microsystems* (GRK 1103), the SFB/TR-8, and the *BrainLinks-BrainTools* Cluster of Excellence (grant number EXC 1086), all funded by the German Research Foundation (DFG).

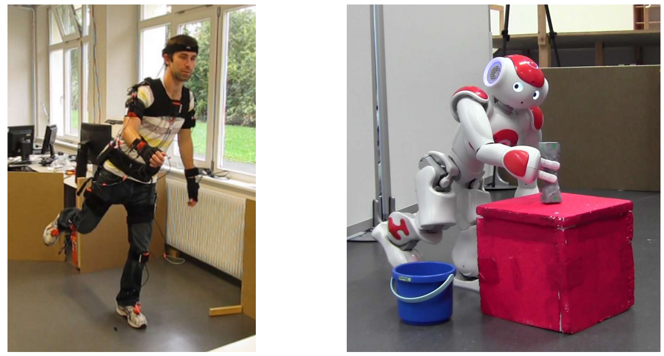

Fig. 1. Imitation of a complex whole-body motion with a humanoid for a tele-operated manipulation task. Note that the robot imitates configurations in which it is required to balance on one foot over a longer period of time.

In this paper, we present an approach that enables humanoids to imitate complex whole-body motions in real time. Instead of relying on a high number of parameters to optimize, we use a compact human model to reduce the computational effort. In particular, we consider the positions of the endeffectors, i.e., the position of the hands and feet, as well as the position of center of mass (CoM) and generate the robot's motion as close as possible to the original motion. Our approach applies inverse kinematics (IK) to generate joint angles for the four kinematic chains given the endeffector positions. Afterwards, we modify the joint angles so as to match the human's CoM position and ensure stability at the same time by using retargeting of the robot's feet and finding statically stable configurations by inverse kinematics.

To the best of our knowledge, our technique is the first one that explicitly imitates also motion sequences with extended periods of time in single support mode, in which balancing on one foot is inevitable. We present experiments with a Nao humanoid reliably imitating complex whole-body motions in real time (see Fig. 1). The human motion is captured with an Xsens MVN motion capture system consisting of inertial sensors attached to the individual body segments. We thoroughly evaluated our approach regarding stability, similarity to the human motion, and computational effort. As the results show, our system generates safe motions for the robot while achieving high similarity to the human and allows for tele-operation in real time. Preliminary results of this work have been published in [12].

II. Related Work

Riley *et al.* [13] described one of the first approach to real-time control of a humanoid by imitation using a simple visual marker system attached to the upper body. The authors apply IK to estimate the human's joint angles and then map it to the robot. Ott *et al.* [7] proposed to use a spring model, in which control points on the robot's skeleton are virtually connected to the markers on the human body. Based on the forces acting on the springs, joint angles are determined that consider the robot dynamics. The authors present experiments in which a humanoid imitates human upper-body motions in real time, the legs are mainly used for balancing. The work by Dariush *et al.* [8] considers imitation as task space control based on low dimensional motion primitives. the authors use a separate ZMP-based balance controller and the lower body is only controlled so as to ensure stability.

Also Yamane *et al.* [11] presented a control-based approach to imitate human motions with a force-controlled robot. Using this technique, joint trajectories for the whole-body of a humanoid can be generated online. The legs are also controlled so as to follow the human motion while maintaining stability. The only assumption is that both feet have ground contact. The authors plan to lift this assumption by integrating techniques to detect stepping motions and adapting the CoM trajectory in the controller according to [14]. In the latter approach, the authors proposed to predict the trajectory of the desired CoP for a number of frames based on the captured motion and then modify the CoM to ensure stability.

Cela *et al.* [15] presented a motion capture system consisting of eight sensors to measure joint angles of the leg and acceleration of the arms. The authors ensure stability during real-time imitation using a feedback control system based on data of an accelerometer placed on the robot's back. With this system, changes from double support to singe support are possible, however, due to the limited set of sensors, no complex motions can be imitated. Recently, Vuga *et al.* [16] introduced an approach to dynamically stable imitation of human motions. The authors use a separate controller for the lower body that ensures stability by allowing imitation in the null space of the balance controller only. In this way, imitation of walking motions is possible.

Stanton *et al.* [10] described a learning approach to determine kinematic mapping between the human and the robot. This technique relies on an initial training phase in which the human is asked to imitate the motions of the robot. Afterwards, the human can tele-operate the robot in real time. Since in this approach no balance controller ist used, the range of motions the robot is able to imitate is constrained.

Suleimann *et al.* [2] focus on the imitation of the upper body. The authors treat the imitation as a constrained optimization problem on a given sequence of captured human motions. Nakaoka *et al.* [17] consider dance movements. In this approach, motion primitives and their parameters are learned offline from observed human motions. Here, the leg motions are not directly imitated but generated from the primitives. In a latter work, Nakaoka *et al.* [4] use a set of models for different leg motions of the robot to ensure that characteristic motions are executed stably during imitation of the dance movement. Also here, the motion models and their parameters are learned in an offline step. For imitation, the particular type of motion primitive is recognized from captured motion data and the leg trajectory are chosen accordingly, so that the robot can safely execute the corresponding sequence. Kim *et al.* [1] also focus on dance movements and use an offline optimization step for determining a kinematic mapping between the human and the robot and ensuring stability of imitated whole-body motions, also during support mode changes. During execution, three online controllers are used for balancing and soft stepping. Chalodhorn *et al.* [3] proposed to apply dimensionality reduction and transform the high-dimensional human motion data in a low-dimensional subspace. The offline optimization of the motions, which takes into account the robot dynamics and stability, is then performed in the reduced subspace. The authors applied the approach to the task of learning to walk by imitation.

In contrast to all the methods above, our system enables a real humanoid to imitate complex whole-body motions that include support mode changes in real time while ensuring static stability. Our approach does not rely on a preprocessing step or on a high number of variables, but uses a compact model of human motions.

III. Human Motion Model

Typically, the captured motion data allows for precisely reproducing the human motions on a virtual model of the human body. However, the execution of the same motion on a robot platform is naturally impossible due to the differences in the number of degrees of freedom and joint range between the two models. In this work, we consider a motion as a sequence of postures f_i

$$f_i = [p_{LHand}^{LShoulder}, p_{LFoot}^{LHip}, p_{RHand}^{RShoulder}, p_{RFoot}^{RHip}], \qquad (1)$$

thus, each posture is defined as the 3D position of the endeffectors, i.e., the hands and feet, relative to the left or right shoulder/hip frame at time i (see Fig. 2). Currently, we just include the endeffectors' positions in the model. However, the model can be easily extended to include the endeffectors' orientations and further features such as the elbow and knee positions. As the proposed method is based on inverse kinematics, additional constraints can be included in an augmented Jacobian or can be solved in the projected Nullspace of the Jacobian. For each posture, we additionally take into account the position of the CoM and the support mode of the demonstrator, which can be estimated from the motion capture data.

By adopting this compact representation of body postures, we account for the limited physical capabilities of humanoid robot platforms with respect to the human body.

IV. Human to Humanoid Posture Mapping

A. Initialization

In order to map motions from the human to the robot model, our system uses a common reference posture, the so-called the T-pose (see Fig. 2). In the following, we will refer to $f_{ref,H}$ and $f_{ref,R}$ as the posture of the human and the robot in the T-pose, defined according to Eq. (1).

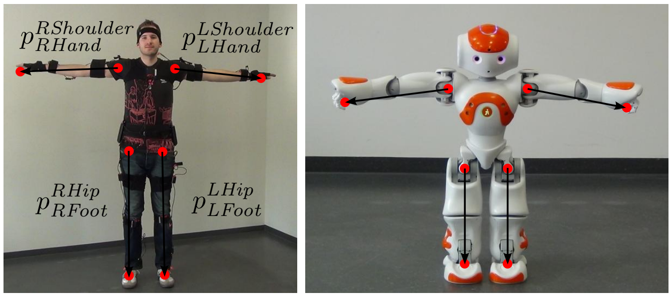

Fig. 2. T-pose of the human and the robot model used as reference for posture mapping.

B. Posture Mapping

To determine the posture change for a new human body posture $f_{i,H}$, our technique first computes the deviation of the endeffector positions relative to their references in $f_{ref,H}$

$$\Delta f_{i,H} = f_{i,H} - f_{ref,H}. \tag{2}$$

In order to imitate the motion of the human, we expect the deviation of the robots endeffector positions Δf_R from their positions in $f_{ref,R}$ to be proportional to the values obtained from Eq. (2), as given by the following equation

$$\Delta f_{i,R} = m \cdot \Delta f_{i,H}, \tag{3}$$

where m represents the proportionality constant, given by the ratio between the limb length of the robot $l_{limb,R}$ and the human $l_{limb,H}$ for the respective kinematic chain, defined as

$$m = \frac{l_{limb,R}}{l_{limb,H}}, \tag{4}$$

where the length of the limbs are obtained from the T-pose. Given $\Delta f_{i,R}$, the robot's posture $f_{i,R}$ at time i is updated as

$$f_{i,R} = f_{ref,R} + \Delta f_{i,R}. \tag{5}$$

For the target positions for each endeffector contained in $f_{i,R}$, we find the corresponding joint angles by an numerical inverse kinematics solver based on the damped least-squares (DLS) method with a weighting matrix to avoid joint limits as proposed in [18]. We chose an iterative solver as opposed to an analytical method since it typically generates continuous motions.

Executing the resulting joint angles will lead to a robot posture that is similar to the captured human posture with respect to the differences between the two models in scale and kinematic structure. Nevertheless, the mapping procedure is insufficient for safe imitation of human motions. First, only the feet *positions* have been considered for the mapping. Thus, the support feet of the robot may not be parallel to the ground. Second, differences in the mass distribution of the robot and the human have to be considered. Finally, the dynamics of the human and the robot are different. For simplification, our approach does not consider the dynamics of the robot but generates a statically stable pose for every point in time as described in the following.

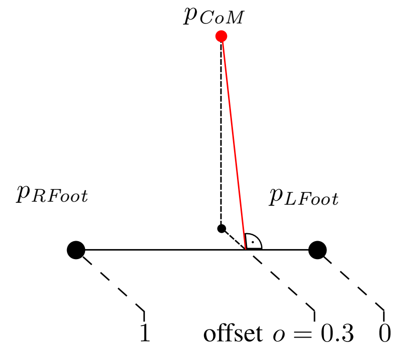

Fig. 3. Determination of the normalized offset given the projection of the center of mass onto the connection line between the feet.

V. Posture Stabilization

To keep the similarity to the human motion, the given unstable robot's posture should be modified as little as possible. Thus, our stabilization method only modifies the configuration of the robot's leg chains. Further, it is important to ensure that the support mode and the trajectory of the CoM of the robot are close to that of the human. Thus, given the unstable robot pose from posture mapping, the current CoM of the human, and its support mode, the goal is to find a statically stable robot posture with similar properties as the human posture.

In summary, our approach works as follows. First, the trajectory of the CoM is adapted to allow for support mode changes and we constrain the changes in the CoM position per time unit to ensure safe execution. Then, the support mode for the robot is determined based on the support mode of the human and the designated position of the CoM. Finally, the endeffector positions of the feet are retargeted to generate a statically stable pose and the corresponding joint configurations are found by an IK solver. These steps are explained in detail in the following.

A. Controlling the Center of Mass

We use a low-dimensional projection of the CoM to describe the position of the CoM p_{CoM} relative to the positions of the feet as a scalar factor. This offset o is determined by the orthogonal projection of the CoM onto the connection line between the feet (see Fig. 3). The offset is normalized between 0 and 1 so that it describes the relative distances of the projected CoM to the feet center positions p_{LFoot} and p_{RFoot}. With this normalization, the offset from the human motion data can be directly translated to the robot. The offset is computed as follows:

$$o = \frac{(p_{CoM} - p_{LFoot}) \cdot (p_{RFoot} - p_{LFoot})}{\|p_{RFoot} - p_{LFoot}\|^2} \tag{6}$$

As the CoM of the human is not necessarily over a single support foot when changing to the single support mode, the trajectory of the offset has to be adapted for the robot to allow for support mode changes. For example, before the robot can safely lift its right foot to balance on a single foot, first the offset has to be 0. Thus, whenever the human stands on a single foot, the offset for the robot is forced to

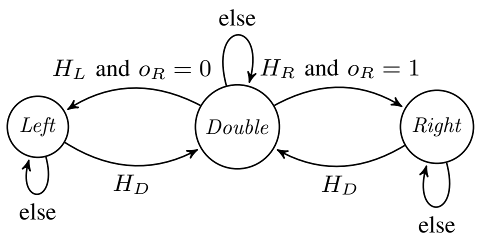

Fig. 4. Robot support mode states and transitions. H_D, H_L, and H_R indicate whether the human is in double, left, or right support mode. o_R refers to the normalized offset (see Fig. 3). Before a change to single mode occurs, the CoM is smoothly shifted to the corresponding leg.

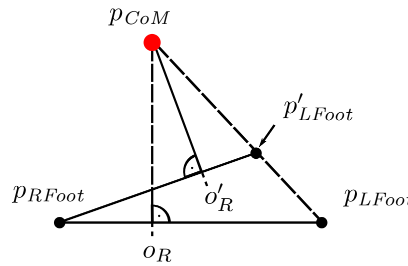

Fig. 5. Double support posture stabilization. Depending on the CoM and the desired offset o'_R, the position of one foot is retargeted and the joint angles of the corresponding leg chain are recomputed so that the resulting posture is statically stable.

be 0 or 1. Obviously, this would result in fast changes in the trajectory of the CoM. Thus, the velocity of the offset is limited to generate smooth and safe trajectories of the CoM. Here, we use a negative quadratic function with the maximum at $o = 0.5$ and close to zero velocity at the borders of the offset range. In practice, this results in safe motion imitation, but on the other hand causes a slight delay in the imitation process when support mode changes occur. Reducing the delay by adapting the function parameters is generally possible, though comes along with a higher risk of loosing balance. A sample trajectory of the human and robot offset is illustrated in Fig. 9 in the experimental section.

B. Controlling the Support Mode

The robot cannot directly imitate the support mode of the human who can almost instantaneously change from double to single support and vice versa. Instead, the robot first has to carefully shift its CoM over the support foot to avoid falling. Our approach uses a finite state machine to model the support mode of the robot based on the support mode of the human (which is either double (H_D), left (H_L), or right (H_R)), the robot's current support mode, and the offset of the robot o_R. The state transitions are illustrated in Fig. 4. When the robot is in double support, it will only change to single support if the human is in single support *and* if its own offset is 0 or 1. If the offset has a value in between, the robot is not allowed to change to single support, even if the the human is already on a single foot. To achieve single support, the offset is smoothly shifted towards 0 or 1 by the offset control described in the previous subsection. Accordingly, the robot needs a few frames to change its support mode.

C. Endeffector Retargeting

It remains to describe how to generate statically stable robot postures after posture mapping and determination of the CoM position and support mode. Our approach finds a new target for either one foot or both feet so that the given offset (see Sec. V-A) is fulfilled.

1) Double Support: In the double support mode, one foot is repositioned so that the CoM, projected on the connection line between the feet, equals to the desired offset factor. The repositioning in double support mode is illustrated in Fig. 5. Here, the left foot position p_{LFoot} is shifted in the direction of the CoM position p_{CoM} to its new target position p'_{LFoot} so that the desired offset o'_R is met. Whether the right or the left foot is moved, depends on the desired offset o'_R and on the current offset o_R which is calculated from the given pose. If $o'_R < o_R$, the left foot is repositioned, otherwise the right foot.

Afterwards our approach calculates new target orientations for the feet so that they have the same orientation and span a plane on the ground. The orientation is given by the direction of the up-pointing vector of the feet, which is the normal $n_{\{L/R\}Foot}$ of the desired plane:

$$n_{\{L/R\}Foot} = p_{CoM} - (p'_{LFoot} + o'_R \cdot (p_{RFoot} - p'_{LFoot})) \quad (7)$$

Accordingly, it points from the projected CoM corresponding to the desired offset o'_R to the new CoM position.

2) Single Support: In single support mode, the foot positions stay the same and it is sufficient to find a new target orientation for the supporting foot so that the posture is statically stable. The direction vector is given by the same formula as above with an offset of either 0 or 1.

With the new target positions and orientations for the feet, we can solve for the joint configurations with IK. We state the IK problem as a 5 DOF problem by the target positions and orientation, given by the new position $p'_{\{L/R\}Foot}$ and the orientation $n_{\{L/R\}Foot}$. In contrast to posture mapping, precision is crucial for stabilization. Therefore, we run the damped least-squares IK solver until the error is below a certain tolerance.

D. Postprocessing

To make our system more robust, we implemented methods for limiting the velocity of the non-supporting endeffectors and avoiding collisions. These methods can be easily integrated into our approach by running the IK solver on the modified target positions whenever a collision or a too fast endeffector velocity is detected.

VI. Experimental Results

We used a Nao v4 developed by Aldebaran Robotics for evaluating our system. Nao is 58 cm tall, weighs 5.2 kg and has 25 degrees of freedom. The human was wearing an MVN Suit by Xsens for capturing the motions. This inertial sensor based tracking device provides an accurate estimate of the human's posture from which the target positions of the endeffectors, used as input for posture mapping, can be extracted. For posture mapping, we run the damped least-squares based IK solver with a fixed number of 30 iterations

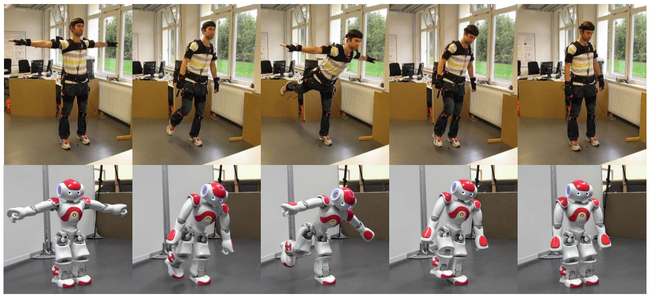

Fig. 6. Nao humanoid imitating a human performing a motion to reach a complex single support posture. Using our approach, the robot can even keep its balance when it is in single support for longer periods of time. The entire motion sequence is contained in the accompanying video. The leftmost image shows the calibration posture for the mapping process.

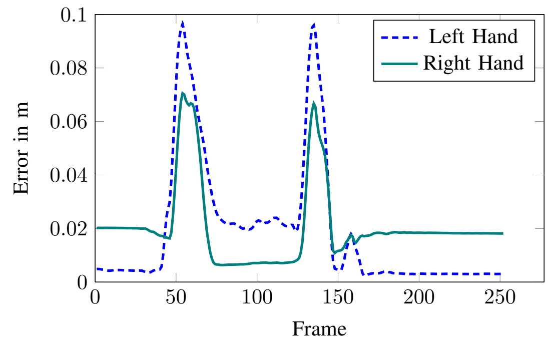

Fig. 7. Deviation of the hand positions from the desired positions for the motion depicted in Fig. 6. The average error is only 1.4 cm.

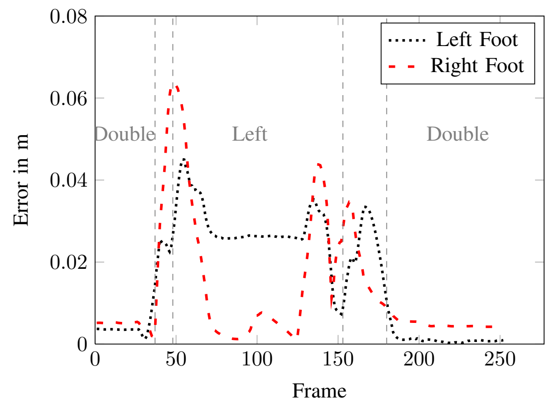

Fig. 8. Deviation of the generated feet positions from the desired ones for the sequence depicted in Fig. 6. The increased error in the left foot during support mode changes results from shifting the CoM within our stabilization method. Furthermore, the human right leg is moved fast backward and forward, respectively, and the robot needs a few time steps to catch up. The average error is only 1.6 cm.

for the arms and 5 iterations for the feet to avoid singular configurations (e.g., leg stretched out). A rough tracking of the feet is sufficient in the posture mapping as the positions are adapted for stability in the stabilization step. Singular configurations are hard to escape during the stabilization process.

The IK solver in the stabilization method was executed until a precision of at least 1 mm or 0.033 rad is reached. We experimentally determined a damping factor of 0.3 for the DLS method, which turned out to be a good compromise between stability and convergence of the IK method. Computations were performed on a single core of a standard desktop CPU (Intel Quadcore i5-2400, 3GHz).

In the following, we evaluate our system in terms of similarity of the robot motion to the demonstrated motion, stability, and computational cost. Finally, we present a teleoperation experiment as an application scenario of our approach.

A. *Similarity to Human Motion*

To evaluate the similarity of the robot's motion computed by our approach and the demonstrated motion, we measured the differences of the corresponding endeffector positions in a complex whole-body motion, which contained fast arm movements and support mode changes. The demonstrator stretched his right foot backwards and lifted the arms while remaining in single support to reach the posture illustrated in Fig. 6. Here, images from this motion sequence are shown together with the humanoid robot imitating the given motion. At the beginning, the demonstrator adopted the T-Pose to calibrate the mapping.

We compare the trajectory of the controlled endeffector position with the trajectory of the desired endeffector positions given by posture mapping from the human model (see Sec. III). The errors are caused by the precision of the IK in posture mapping, by preserving joint limits and by applying endeffector and joint velocity limits in the postprocessing step of the stabilization method. For the feet there is an increased error due to the stabilization method, especially in the case of support mode changes.

Fig. 7 shows the errors of the hand positions of the complex single support sequence in Fig. 6. The hand positions have an increased error when the demonstrator raises and lowers its arms due to the higher velocity. Fig. 8 plots the corresponding errors of the foot positions. The stabilization method uses the legs of the robot to shift its center of mass and realize the changing of the support mode. In the first double support phase, the deviation from the target positions is small. When the demonstrator changes to the left foot, the robot needs some time to shift its CoM to the left foot while the demonstrator continues with the motion. As the demonstrator makes a fast backwards movement of the right leg, the error grows a bit larger for the right foot. This effect is strengthened by the endeffector velocity control of the postprocessing step of the stabilization method. In the left support phase, the left foot keeps balance while the right foot attempts to catch up with its desired position. At the end of the left support phase, the right foot has a high acceleration forwards and is then placed on the ground. Finally, the CoM is shifted towards the neutral position by the stabilization procedure using both feet, resulting in a temporarily increased error in the left and the right foot.

To calculate the average error of the endeffector positions, we repeated the experiment with the complex single support posture of Fig. 6 10 times. Over all repetitions the average

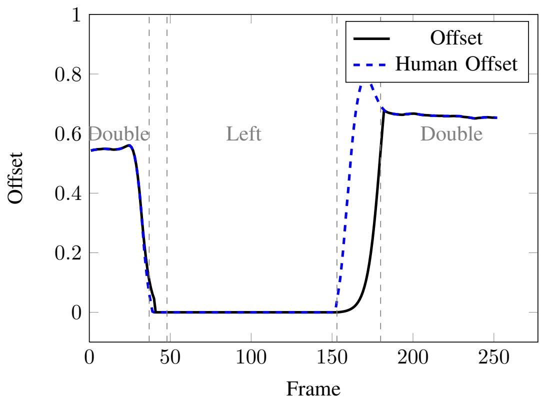

Fig. 9. Evolution of the offset value over time for the sequence depicted in Fig. 6. As can be seen, the robot's offset closely follows the human's value. During support mode changes, the speed of shifting the robot's CoM is limited to ensure stability.

TABLE I

COMPUTATIONAL EFFORT AND NUMBER OF IK ITERATIONS.

	Mean	Max
Posture mapping	1.04 ± 0.35 ms	1.73 ms
Posture stabilization	1.85 ± 3.50 ms	27.08 ms
Total time	3.30 ± 3.57 ms	29.21 ms
# IK iterations	54 ± 178	1784

error of the hands is 1.4 cm and of the feet 1.6 cm. Although the sequences have to be adapted to ensure stability, the motion of the robot is still very similar to the motion of the demonstrator.

B. Ensuring Stability

In the next set of experiments, we evaluate our technique to achieve stability during motion imitation. In particular, we consider the offset value introduced in Sec. V, which is the low-dimensional representation of the projected CoM. In Fig. 9, the temporal evolution of the offset is plotted for the sequence of Fig. 6. As can be seen from the figure, the offset of the robot equals the offset of the demonstrated motion in the double support phase, in which the center of mass is slightly shifted to the left foot. Shortly before the left support phase, the offset is forced to approach a value of 0. The speed of changing the offset value is limited by the offset control so that the CoM is not shifted too fast. The offset value stays 0 during the left support phase until the demonstrator places the right foot back on the ground. Then, the offset approaches its specified value, with limited velocity forced by the offset control.

As the experiments show, our technique controls the robot's CoM so as to imitate the human motion as close as possible while ensuring stability. More specifically, our system generates a trajectory of the robot's CoM that closely follows the human CoM trajectory, achieves the desired support mode changes, and limites the velocity of the CoM changes to ensure safe execution.

C. Computational Costs

To evaluate the computational costs, we measured the calculation times for a long sequence of 1000 frames. It

Fig. 10. Tele-operation with visual feedback. *Left*: Human operator and live view of the robot's camera displayed on a monitor. *Right*: Tele-operated Nao humanoid.

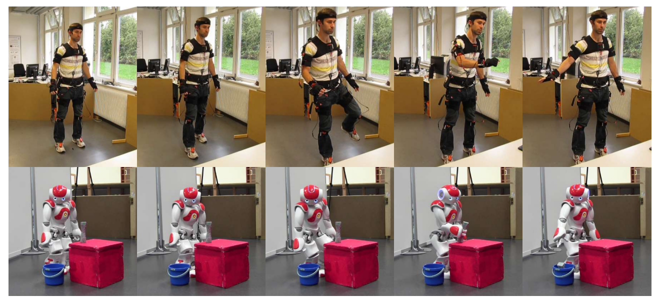

Fig. 11. Tele-operated walking and object manipulation. The complete motion sequence is contained in the video accompanying this paper.

contains different movements such as stepping and reaching, as can be seen in the tele-operation experiment in Sec. VI-D. The calculation times are given in Table I. The time for posture mapping includes finding the target positions for all endeffectors and running the IK solver to find the joint configurations for the robot. In the posture mapping process, the transformations can be calculated in constant computational time and the number of iterations of the IK is fixed to a small number whereas the stabilization time is dominated by the inverse kinematics calculations. Here, the IK solver has to run many iterations to find a solution within the desired precision. The number of iterations of the damped least squares solver are also listed in the table. The total time includes all computations of our system, beginning when the captured human data is obtained, and ending when a statically stable pose has been generated for the robot.

The calculation times are within a few milliseconds on average. Some configurations require a larger amount of iterations to converge to the desired precision, but the computation time does still not exceed 30 ms in the worst case. Even complex whole-body motions can be safely imitated with a rate of 30 frames per second.

D. Tele-Operation

Finally, we use our system to control the robot in a tele-operation setup for object manipulation. In this set of experiments, only the image of the robot's camera is projected onto a screen to the demonstrator wearing the motion capture suit. The camera is always pointing to the right hand of the robot, so that the demonstrator gets a good view of the manipulator and the object to be manipulated. An overview of the experimental setup is shown in Fig. 10.

We performed two sequences including stepping, balancing in single support, and grasping an object. In the first experiment, the demonstrator balances on the left foot while he picks up the object. In this way, the robot is able to reach a distant object as it leans forward and balances the CoM by stretching the right foot backward. A snapshot of this sequence is shown in Fig. 1. After successfully grasping the object, the demonstrator changes to double support and drops the object into the blue bucket. In this experiment, we constrained the right hand of the robot to keep a upright orientation.

In the second experiment, the robot is too far away to reach the object (see Fig. 11). The demonstrator first performs two steps to get closer to the object, before he stretches his right arm to pick-up the object and drops it into the bucket. Both tele-operation experiments were successfully performed on the robot. They are included in the video accompanying this paper.

VII. CONCLUSIONS

We presented a technique for real-time imitation of human whole-body motions. Our approach uses a compact human model and applies inverse kinematics to find robot postures that imitate the human demonstrator. To achieve safe imitation of challenging sequences online, we generate statically stable configurations and constrain the center of mass velocity. The novelty of our system is that it also allows for single support phases where the center of mass has to be actively balanced over the support foot for a longer period of time. Experiments with a Nao humanoid and a MVN Suit by Xsens demonstrate the capability of our approach to reliably generate safe motions of the robot closely following the human reference also for complex motion sequences. The required computational time is only 3 ms on average, thus allowing for real-time tele-operation.

REFERENCES

[1] S. Kim, C. Kim, B. You, and S. Oh, “Stable whole-body motion generation for humanoid robots to imitate human motions,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[2] W. Suleiman, E. Yoshida, F. Kanehiro, J.-P. Laumond, and A. Monin, “On human motion imitation by humanoid robot,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2008.

[3] R. Chalodhorn, D. B. Grimes, K. Grochow, and R. P. N. Rao, “Learning to walk through imitation,” in *Int. Conf. on Artificial Intelligence (IJCAI)*, 2007.

[4] S. Nakaoka, A. Nakazawa, F. Kanehiro, K. Kaneko, M. Morisawa, H. Hirukawa, and K. Ikeuchi, “Learning from observation paradigm: Leg task models for enabling a biped humanoid robot to imitate human dances,” *Int. Journal of Robotics Research (IJRR)*, vol. 26, no. 8, 2007.

[5] A. Ude, C. Atkeson, and M. Riley, “Programming full-body movements for humanoid robots by observation,” in *Robotics and Autonomous Systems*, 2004.

[6] A. Safonova, N. Pollard, and J. K. Hodgins, “Optimizing human motion for the control of a humanoid robot,” in *Int. Symp. on Adaptive Motion of Animals and Machines (AMAM)*, 2003.

[7] C. Ott, D. Lee, and Y. Nakamura, “Motion capture based human motion recognition and imitation by direct marker control,” in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2008.

[8] B. Dariush, M. Gienger, A. Arumbakkam, Y. Zhu, B. Jian, K. FujiMura, and C. Goerick, “Online transfer of human motion to humanoids,” *Int. Journal of Humanoid Robotics (IJHR)*, vol. 6, no. 2, 2009.

[9] M. Do, P. Azad, T. Asfour, and R. Dillmann, “Imitation of human motion on a humanoid robot using non-linear optimization,” in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2008.

[10] C. Stanton, A. Bogdanovych, and E. Ratanasen, “Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning,” in *Proc. of the Australasian Conf. on Robotics and Automation (ACRA)*, 2012.

[11] K. Yamane and J. Hodgins, “Controlling humanoid robots with human motion data: Experimental validation,” in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2010.

[12] J. Koenemann and M. Bennewitz, “Whole-body imitation of human motions with a nao humanoid,” in *Video Abstract Proc. of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2012.

[13] M. Riley, A. Ude, K. Wade, and C. G. Atkeson, “Enabling real-time full body imitation: A natural way of transferring human movements to humanoids,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2003.

[14] K. Yamane and J. Hodgins, “Control-aware mapping of human motion data with stepping for humanoid robots,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[15] A. Cela, J. J. Yebes, R. Arroyo, L. M. Bergasa, R. Barea, and E. Lopez, “Complete low-cost implementation of a teleoperated control system for a humanoid robot,” *Sensors*, vol. 13, no. 2, 2013.

[16] R. Vuga, M. Ogrinc, A. Gams, T. Petric, N. Sugimoto, A. Ude, and J. Morimoto, “Motion capture and reinforcement learning of dynamically stable humanoid movement primitives,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013.

[17] S. Nakaoka, A. Nakazawa, K. Yokoi, H. Hirukawa, and K. Ikeuchi, “Generating whole body motions for a biped humanoid robot from captured human dances,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2003.

[18] T. F. Chang and R. V. Dubey, “A Weighted Least-Norm Solution Based Scheme for Avoiding Joints Limits for Redundant Manipulators,” *IEEE Trans. on Robotics and Automation*, vol. 11, no. 2, Apr. 1995.