Exploiting Task Intervals for Whole Body Robot Control

Michael Gienger, Herbert Janßen and Christian Goerick
Honda Research Institute Europe GmbH
Carl-Legien-Straße 30
63073 Offenbach, Germany
Email: michael.gienger@honda-ri.de

***Abstract*— This paper presents a whole body motion algorithm and shows some steps towards its feasibility in complex scenarios. We employ the framework of *Liégeois* [1] which solves the redundant inverse kinematics problem on velocity level. To make the controller suitable for a variety of different applications, task descriptors for the relative effector positions as well as a one- and two-dimensional attitude representation are proposed. The inverse kinematics are extended by allowing for "displacement intervals" which are formulated in task space. The proposed control scheme guarantees that the effector motion lies within the specified interval. However, the motion inside the interval is determined by optimization criteria, which can effectively be utilized to generate a more flexible and robust motion. We will discuss an example and show simulation and experimental results on the humanoid robot ASIMO.**

I. Introduction

In common classical motion control algorithms the tracking of the desired trajectories is very accurate. A vast amount of literature is addressing the problem of high accuracy motion control. This makes sense in most applications, like e. g. assembly robots, medical robots, etc. In some applications, the tracking accuracy of a reference trajectory is not very critical, or there are at least some phases where the accuracy may be lower than in others. For example, a "reaching" or "pointing" of a humanoid robot to an object does not necessarily have to precisely follow the commanded trajectory. A certain impreciseness is permitted, and sometimes even desired, since machine-like motions look somewhat strange when performed by humanoid or animal robots.

To circumvent this problem, we propose a novel formulation to allow for displacement intervals in task space. These intervals are valid regions around a given task variable, in which the control task has to be achieved. Analogous to the null space motion, the displacement intervals are exploited to satisfy one or several cost functions. By choosing appropriate criteria, the motion can be influenced in almost arbitrary manners, e. g. with respect to joint limit or collision avoidance, energy minimization, etc.

Another key issue addressed in this paper is the representation of the task. Much can be gained by describing the task in a way such that the motion only gets constrained where necessary. We will show how to achieve a robust motion by selecting a meaningful task description, and will present alternative descriptions for effector positions and spatial orientations.

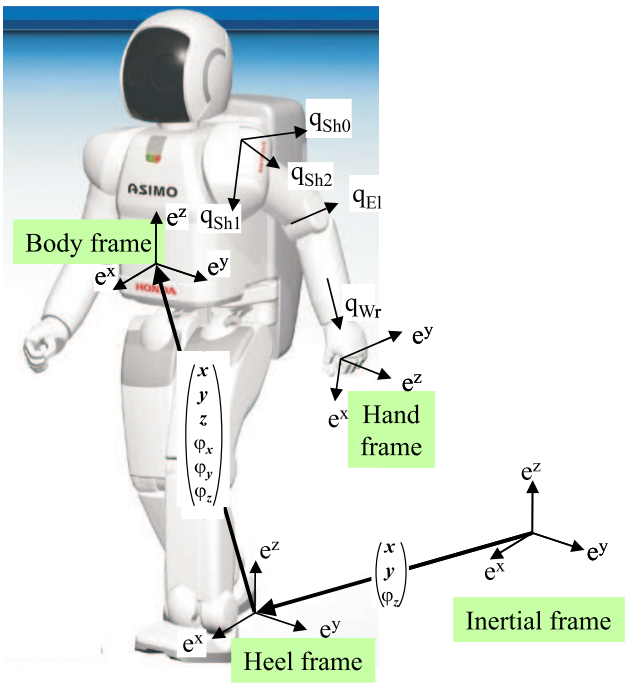

Figure 1: Kinematic model for whole body motion

The literature on redundant and whole body control is large. A comprehensive book on redundant control techniques has been published by *Nakamura* [2]. *Buss* [3] gives an overview on rigid body kinematics with tree-like structure. Particular emphasis is devoted to the inverse kinematics problem, addressing Jacobian transpose, pseudo-inverse and damped least squares methods. How to deal with ill-conditioned or singular mappings is shown by *Maciejewski* [4]. Pseudo-inverse method and extended Jacobian approach are compared and applied to a 30 dof robot by *Tevatia et al.* [5]. The balance problem is addressed in [6]–[8]. *Sian et al.* develop a teleoperation framework for their humanoid robot *HRP-2* that enables the remote operator to command different effectors. A similar method has been developed by *Nishiwaki et al.* [9].

They propose a whole body motion generator that adaptively selects postures from a predefined set (kneeling, standing, etc.) in order to reach the target. Many works focus on exploiting the null space to satisfy further objectives. Joint limit and obstacle avoidance are addressed in [10]–[15]. While above mentioned works are based on resolved motion rate control, *Khatib* employs the *Operational Space Formulation* which describes the relationship between end effector accelerations and forces [16], [17].

The paper is structured as follows: The second section briefly reviews the employed motion control which follows the velocity-based scheme proposed by *Liégeois* [1]. The issue of describing a task with the appropriate variables is key to robust motion generation and is often underestimated. We discuss this topic in section three and give some examples on how to represent positions and spatial orientations of effectors. In section four, a formulation for the displacement intervals is derived, and it is shown how to incorporate this formulation into the inverse kinematics computation. A validation of the proposed methods is given in section five. We discuss an example and show simulation and experimental results on the humanoid robot ASIMO.

II. Redundant control scheme

The kinematic model of the robot is depicted in figure 1. In initial configuration, the x-axis points forward, the z-axis points upward and the y-axis accordingly to the left. Pan, roll and tilt describe a rotation about the z-, x- and y-axis, respectively. The kinematic structure and degrees of freedom of the links correspond to those of the humanoid robot *ASIMO* [18].

A. Robot model

The first link corresponds to the heel coordinate system comprising three degrees of freedom. It is centered between the feet and aligned with the heel edge. Its degrees of freedom are translations in the x- and y-direction as well as a rotation about the z-axis. The consecutive links correspond to the body segments of the robot. The pelvis is undergoing three translations and rotations with respect to the heel frame. The head is connected to the upper body with pan and tilt joints. Further, the two arms comprise 5 dof each. Three joints drive the shoulder, one joint is located in the elbow, and the hand can be rotated with respect to the forearm with another degree of freedom. An additional coordinate system with some offset to the hand origin defines a hand reference point. All together, the model comprises 21 dof. The state vector consists of the degrees of freedom that can directly or indirectly be driven and such comprises

$$\boldsymbol{q} = \begin{pmatrix} ({}_I x_{hl}\ {}_I y_{hl}\ {}_I\varphi_{z,hl})^T \\ ({}_{hl}x_{ub}\ {}_{hl}y_{ub}\ {}_{hl}z_{ub})^T \\ ({}_{hl}\varphi_{x,ub}\ {}_{hl}\varphi_{y,ub}\ {}_{hl}\varphi_{z,ub}\ )^T \\ \boldsymbol{\varphi}_{arm,L}^T \\ \boldsymbol{\varphi}_{arm,R}^T \\ (\varphi_{pan}\ \varphi_{tilt})^T \end{pmatrix} \tag{1}$$

where left indices describe the coordinate frame in which the respective vector is represented, with *I*, *hl* and *ub* denoting the inertial, heel and upper body frames.

B. Inverse Kinematics

To map the task space magnitudes into joint space, the framework "redundancy resolution" first proposed by *Liégeois* [1] for redundant systems is employed. The basis for the following calculations is the task Jacobian $\boldsymbol{J}$, which relates the task velocities to the state velocities: $\dot{\boldsymbol{x}} = \boldsymbol{J}\dot{\boldsymbol{q}}$. A detailed derivation of the Jacobian for the presented kinematic description is given in [19]. The state motion rates can be computed as

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\#}\,\dot{\boldsymbol{x}} - \alpha\,\boldsymbol{N}\boldsymbol{W}^{-1}\left(\frac{\partial H}{\partial \boldsymbol{q}}\right)^T \tag{2}$$

with

$$\begin{aligned} \boldsymbol{J}^{\#} &= \boldsymbol{W}^{-1}\boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^T)^{-1} \\ \boldsymbol{N} &= \boldsymbol{E} - \boldsymbol{J}^{\#}\boldsymbol{J} \end{aligned} \tag{3}$$

where $\boldsymbol{J}^{\#}$ is a weighted generalized pseudo-inverse of $\boldsymbol{J}$ with weighting matrix $\boldsymbol{W}$. Matrix $\boldsymbol{E}$ denotes the identity matrix. We choose a diagonal weighting matrix with elements corresponding to the work range of the individual degrees of freedom. Such, each joint will contribute to the overall motion according to its available motion range. Scalar H is an arbitrary optimization criterion, whose gradient is mapped into the null space with projection matrix $\boldsymbol{N}$ and scalar α defining the step width. In this example, a joint limit avoidance criterion is employed. It penalizes the squared deviation of the joint angles from the joint zero points (see e. g. [14]). The gradient computes as

$$\left(\frac{\partial H}{\partial \boldsymbol{q}}\right)^T = \left(\frac{q_{i,0} - q_{0,0}}{q_{max,0} - q_0} \ldots \frac{q_{i,n} - q_{0,n}}{q_{max,n} - q_n}\right)^T \quad . \tag{4}$$

Indices "max" may be replaced by "min", when the joint is in the respective range. This accounts for an asymmetric work range of individual joints.

Equation (2) does not yet consider the constraints that are required to maintain balance during standing and walking. These are handled within a separate walking and balancing controller (see [18], [20]) correcting the center of gravity position by shifting the robot's upper body in x- and y direction. To account for this, we constrain these degrees of freedom by modifying equation (2) and regard them as external disturbance (see [19]).

III. Task description

The task vector is composed of three subsets: left hand, right hand and head motion. Usually, the hand tip position is described in Cartesian coordinates x, y and z with respect to the world or robot-base fixed coordinate frame. The spatial orientation of the effectors are commonly described with a set of unit rotations (e. g. Kardan- or Euler angles) or Quaternions.

In many tasks, symmetry can be exploited, such that an attitude representation with less degrees of freedom becomes feasible. Such descriptions have the advantage of reducing the dimension of the task vector, making the null space motion more effective regarding the chosen optimization criteria.

A. Relative effector position description

Under certain circumstances, it is advantageous to control the position difference between the end effectors. This is usually the case for coordinated bi-manual movements, like tool usage. One example is the task of pouring the contents of a bottle into a glass, where the bottle is held with one hand, while the glass is held with the other hand.

Instead of specifying the bottle and glass position independently and adding constraints to close the kinematic chain, the Jacobian can directly be derived to describe the difference between left and right hand reference points. In the following, index b denotes the effector reference point that describes the bottle tip, index g describes the reference point coinciding with the center of the glass orifice.

$$\boldsymbol{x}_{diff} = \boldsymbol{x}_b - \boldsymbol{x}_g \tag{5}$$

This position difference can be projected in any reference frame. The speed difference $\dot{\boldsymbol{x}}_b - \dot{\boldsymbol{x}}_g$ can be written in joint velocities as

$$\dot{\boldsymbol{x}}_{diff} = (\boldsymbol{J}_b - \boldsymbol{J}_g)\, \dot{\boldsymbol{q}} \quad . \tag{6}$$

B. One-dimensional attitude description

In certain scenarios, the effector's spatial orientation may be described with one degree of freedom. This is the case if the relevant variable is the angle between an effector-fixed vector with respect to some reference vector. An illustrative example is again the motion of a bottle whose content is being poured into a glass. Here, the effect of having the content streaming out does only depend on the angle between the bottle axis and the vertical direction. The rotation of the bottle about its symmetry axis and the direction of the plane spanned by bottle axis and $\boldsymbol{g}$-vector don't play a role.

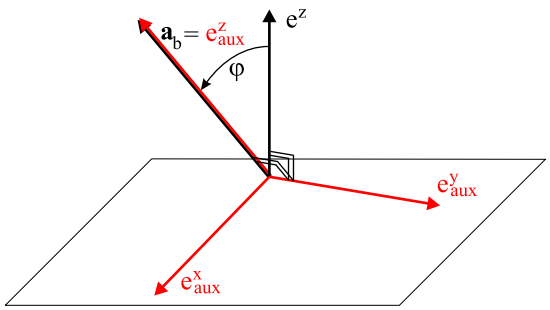

Figure 2: One-dimensional hand attitude representation

Using this example (see figure 2), angle φ denotes the angle between bottle axis $\boldsymbol{a}_b$ and the vertical direction $\boldsymbol{e}^z$. To compute the joint motion for such a representation, the Jacobian relating $\delta\varphi$ to infinitesimal joint motion increments $\delta\boldsymbol{q}$ needs to be derived. For this, a coordinate system with $\boldsymbol{a}_b$ being the z-axis is constructed. The y-axis is the normalized outer product of $\boldsymbol{a}_b$ and $\boldsymbol{e}^z$, the x-axis results accordingly. The rotation matrix $\boldsymbol{A}_{aux-I}$ that rotates a body from the inertial (I) frame into the constructed frame is composed of the row unit vectors of the derived coordinate system [21]. Since the Jacobian ${}_I\boldsymbol{J}_R$ relating the inertial effector angular velocity already has been computed (see e. g. [19]), the desired Jacobian projection can be written as

$$\dot{\varphi} = \boldsymbol{J}_{1D-att}\, \dot{\boldsymbol{q}} = (\boldsymbol{A}_{aux-I}\ {}_I\boldsymbol{J}_R)_{row\ y}\ \dot{\boldsymbol{q}} \quad . \tag{7}$$

Figure 2 shows that angle φ is only influenced by a rotation about the $\boldsymbol{e}^y_{aux}$-axis. This is described with the y-row of $\boldsymbol{J}_{1D-att}$ in eq. (7). It should be noted that for φ being exactly $0°$ or $180°$, this description becomes singular, since the rotation axis is undefined.

C. Two-dimensional attitude description

In certain applications like grasping or manipulating cylindrical objects or for controlling a camera along a gaze axis, symmetry aspects can effectively be exploited to represent the task. In such cases, the direction of the axis should be constrained, but the rotation of the effector about this axis may be unconstrained (see figure 3).

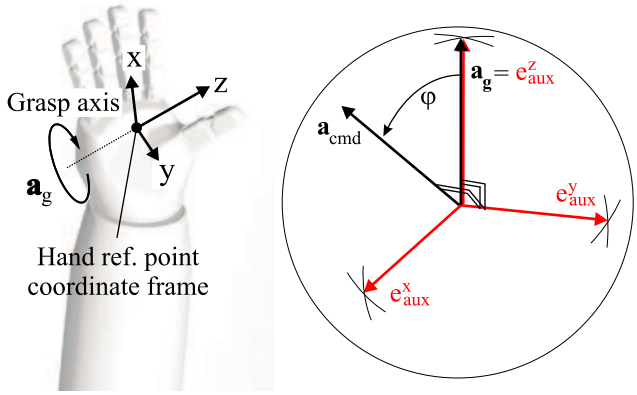

Figure 3: Two-dimensional hand attitude representation

To describe such tasks in a suitable way, we propose to compose the task vector as a 3-dimensional vector of unit length as target axis to be aligned with the effector-fixed z-axis. The kinematics can be derived in a similar way as proposed for the 1-dimensional attitude representation. In this case, angle φ is defined as the angle between actual axis $\boldsymbol{a}_g$ and target axis $\boldsymbol{a}_{cmd}$. The auxiliary coordinate system is set up in the same way as shown before. The non-relevant parameter is the rotation of the effector about the $\boldsymbol{e}^z_{aux}$-axis. The $\boldsymbol{e}^y_{aux}$-axis rotates the effector axis into the target axis, while the rotation about the $\boldsymbol{e}^x_{aux}$-axis should be zero. The Jacobian projection becomes

$$\begin{pmatrix} 0 \\ \dot{\varphi} \end{pmatrix} = \boldsymbol{J}_{2D-att}\, \dot{\boldsymbol{q}} = \left(\boldsymbol{A}_{aux-I}\ {}_I\boldsymbol{J}_R\right)_{rows\ x,y}\ \dot{\boldsymbol{q}} \quad . \tag{8}$$

Only the rows associated with x- and y-rotations need to be considered. This projection becomes singular for axes being parallel to the actual axis. In the case where φ is almost zero, $\dot{\varphi}$ is set to zero. For φ being $180°$, any axis in a plane perpendicular to the target axis may be chosen. For controlling the hand attitude, the rotation axis is set to the lower arm longitudinal axis. For the head description, the pan axis is chosen. This way, the mapping becomes defined for any case.

IV. Displacement Intervals in Task Space

The following section presents a method to assign a displacement boundary to the reference trajectory. The novel aspect is to use a task-specific interval to satisfy additional criteria, similar as redundant control techniques exploit the null space. If the effector reference point is within the permissible interval, a cost function gradient is mapped into the task space. However, since the displacement interval boundaries should be constrained individually for each task space element, the resulting motion is adaptively clipped to account for these limits. It will be shown how to do the clipping for position and the 2-D attitude representations. The proposed scheme allows alterations of the size of the intervals dynamically at run-time. Assuming the cost function to correspond to an end effector force, there is an analogy to compliance control. However, the displacement intervals can be exploited for arbitrary optimization criteria, as e. g. obstacle avoidance, etc. Therefore, the approach can be seen as "virtual compliance" regarding the chosen cost functions.

In the following, the joint limit avoidance cost function H is used as the task interval cost function. Since it is formulated with respect to the state vector, the gradient with respect to the task space is

$$\frac{\partial H}{\partial \boldsymbol{x}} = \frac{\partial H}{\partial \boldsymbol{q}}\ \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}} = \nabla \boldsymbol{H}^T\ \boldsymbol{J}^{\#} \quad . \tag{9}$$

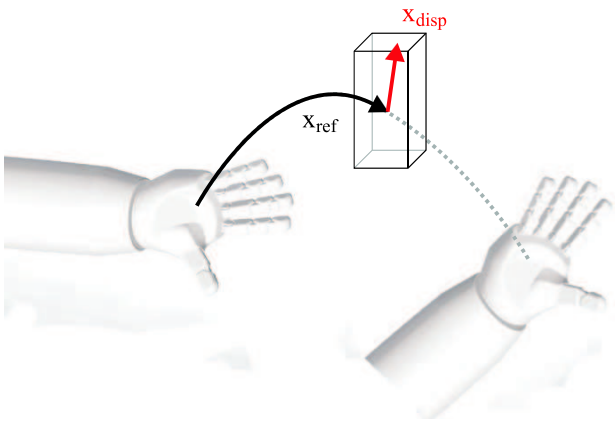

Figure 4: Virtual displacement cuboid

These terms are already known through solving the inverse kinematics. To describe the displacement interval in position coordinates, many solutions are thinkable: ellipsoids, cuboids or other 3-d shapes. We choose a cuboid because the computational complexity is low and the interval can be described in a physically transparent way. The cuboid can be conceived as a virtual box around the reference point, in which the effector is allowed to move (see figure 4). If one dimension of this box is set to zero, the effector may move on a plane. Similarly, setting two box-dimensions to zero, the effector may move on a straight line in the third, free direction. Setting all interval dimensions to zero leads to the standard motion control tracking the reference trajectory exactly. Therefore, the proposed approach can be seen as an extension to common trajectory generation methods. Figure 5 illustrates the determination of the gradient. It computes as

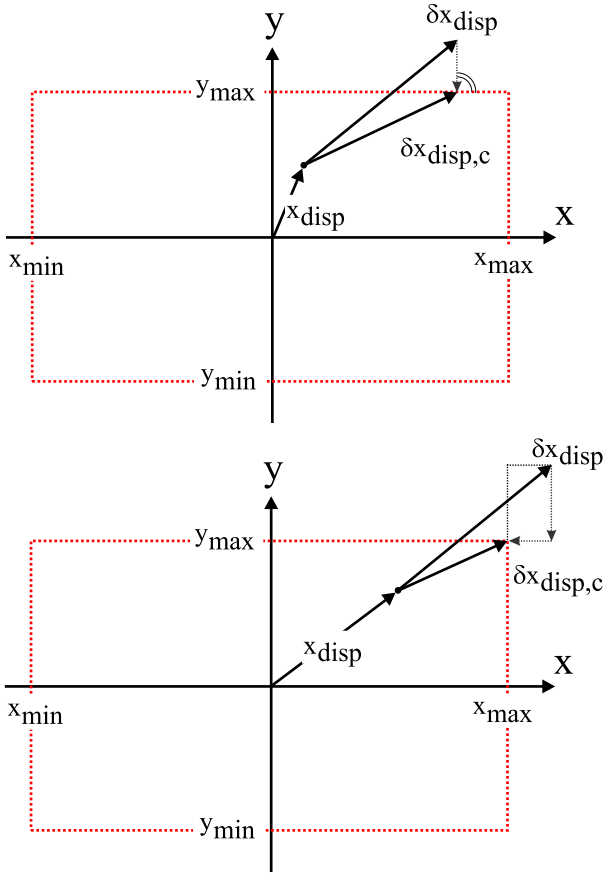

Figure 5: Clipping to position intervals

$$\delta \boldsymbol{x}_{disp} = -\alpha_{pos} \left(\frac{\partial H}{\partial \boldsymbol{x}}\right)^T \tag{10}$$

where α_{pos} defines a convergence step width. The integrated $\delta \boldsymbol{x}_{disp}$ is superposed with the reference trajectory, and it is checked if the updated effector command lies within the permitted boundary. If the boundary is exceeded, the displacement vector $\boldsymbol{x}_{disp}$ is clipped to stay within the permitted region.

Figure 5 illustrates this for the 2-dimensional case. In the upper diagram, the y-limit is exceeded, the x-limit is within the range. The modified displacement vector is shown with subscript "C". In the lower diagram, both x- and y-limits are violated. According to the same scheme, the corrected vector will move toward the edge of the tolerance rectangle.

It has to be mentioned that the clipping procedure will modify the gradient direction, thus leading to a non-optimal gradient descent. However, since the angle between optimal gradient and clipped displacement will never exceed $90°$ and the displacements are small, there will always be a decrease of the cost function.

The interval formulation for the 2-D effector axis attitude description is depicted in figure 6. The commanded effector axis $\boldsymbol{a}_{cmd}$ is allowed to move within a cone with symmetry axis being the reference axis and opening angle φ being the displacement boundary. The cone edge is of unit length, so that the depicted circumference is the intersection of the cone and a unit sphere.

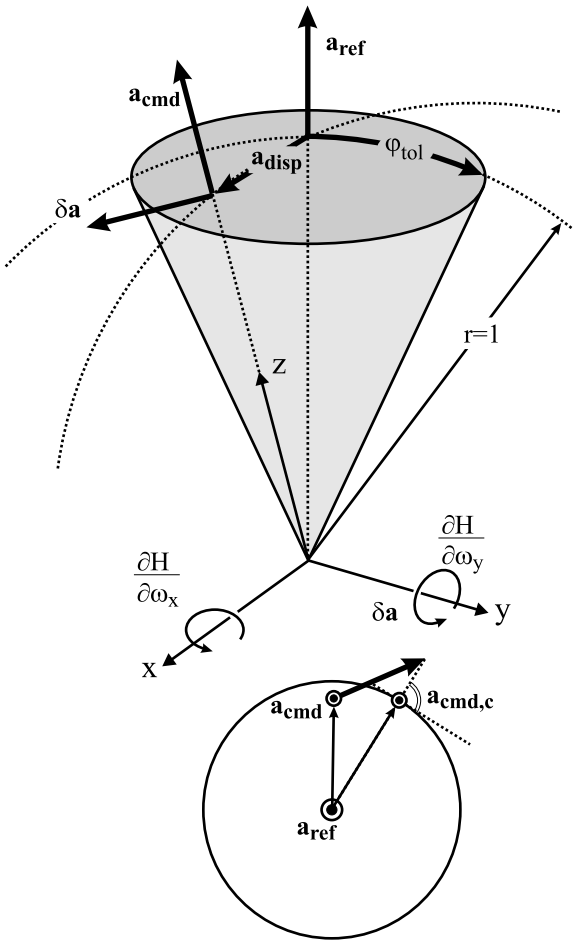

Figure 6: Clipping to cone interval

The tangential displacement on the unit sphere results from the gradients $\frac{\partial H}{\partial \omega_x}$ and $\frac{\partial H}{\partial \omega_y}$:

$$\delta \boldsymbol{a} = -\alpha_{att} \begin{pmatrix} \frac{\partial H}{\partial \omega_x} \\ \frac{\partial H}{\partial \omega_y} \\ 0 \end{pmatrix} \times \boldsymbol{a}_{cmd} \tag{11}$$

with α_{att} being a step width. If the propagated command axis $\boldsymbol{a}_{cmd} = \boldsymbol{a}_{ref} + \boldsymbol{a}_{disp}$ lies within the tolerance cone, no clipping has to be carried out. Otherwise, the command axis has to be clipped according to the lower part of figure 6. Here fore, a rotation axis that is orthogonal on the plane spanned by $\boldsymbol{a}_{ref}$ and $(\boldsymbol{a}_{cmd} + \delta \boldsymbol{a})$ is computed:

$$\boldsymbol{a}_{rot} = \boldsymbol{a}_{ref} \times (\boldsymbol{a}_{cmd} + \delta \boldsymbol{a}) \quad . \tag{12}$$

Knowing the rotation axis, the computation of the new effector command axis is simply

$$\boldsymbol{a}_{cmd,c} = \varphi_{max} \, \frac{\boldsymbol{a}_{rot}}{|\boldsymbol{a}_{rot}|} \times \boldsymbol{a}_{cmd} \tag{13}$$

where $(\varphi_{max} \cdot \boldsymbol{a}_{rot})$ corresponds to the angular velocity that rotates $\boldsymbol{a}_{cmd}$ on the tolerance cone boundary in the desired direction and index c means "clipped".

V. Validation

The bottle pouring example is used to validate the proposed schemes, see figure 7. The robot holds a glass in its right, and a bottle in its left hand. The task vector is composed of the following parameters:

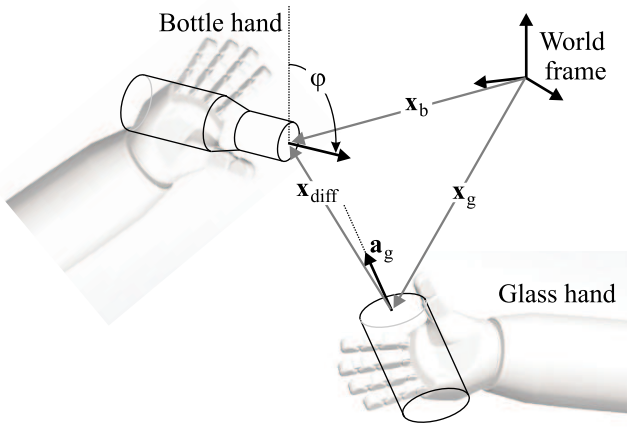

Figure 7: Pouring example

- Position of the glass opening center with respect to the bottle opening $\boldsymbol{x}_{diff}$: The position difference between bottle and glass in horizontal (x- and y) direction is zero. The vertical difference is 200 mm.
- Angle φ between the bottle symmetry axis and the vertical axis (negative $\boldsymbol{g}$-axis): To control the pouring process, a trajectory drives this angle from $0°$ to $180°$ for the pouring and in reverse direction for bringing the bottle back to its original position.

- 2-dimensional glass axis direction: The glass symmetry axis is upright.

The task vector is 6-dimensional, the remaining degrees of freedom are determined by the joint limit avoidance criterion. The trajectory for the bottle inclination is created with a speed-limited higher order low pass filter.

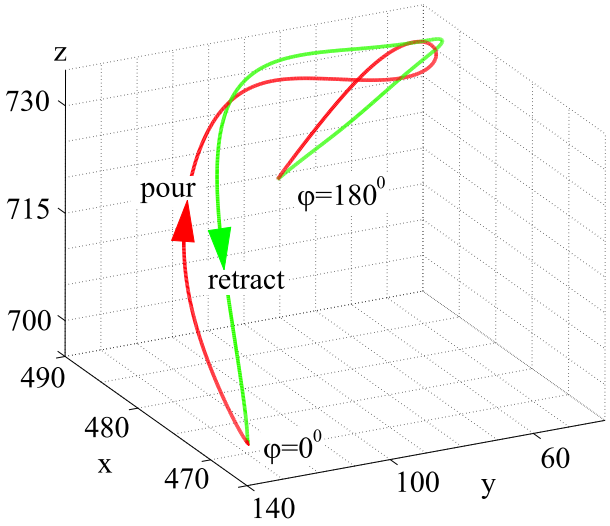

Figure 8: Position of the glass during pouring

One advantageous feature of the proposed task formulation is the resulting motion. Figure 8 shows the trajectory of the glass during the pouring motion. While the relative motion of glass and bottle is being controlled accurately, their absolute position is determined by the null space criterion. This looks quite natural and further has the benefit of being almost optimal in terms of the formulated cost function criteria. The second benefit is the simplification of the task specification for the user. The task variables are very transparent, and the complexity of generating a 3-D trajectory is left to the controller.

To show the benefit of exploiting the tolerance regions, the following cases have been simulated:

1) No displacement intervals
2) Displacement cone for the glass symmetry axis: $\pm 15°$
3) Additional displacement in horizontal direction of the bottle center with respect to the glass center: $\pm 20mm$
4) Additional displacement in vertical direction of the bottle center with respect to the glass center: $\pm 40mm$

With these settings, kinematic simulations of a pouring task have been carried out. Since our robot is position controlled, the simulation results are very close to the real robot characteristics.

Figure 9 shows the associated cost functions of the above mentioned four test cases through the pouring motion. The bottle is held upright for $\varphi = 0°$, and upside down for $\varphi = 180°$. In case 4, the peak cost is reduced to less than 70% as compared to case 1. Still, this task description is feasible,

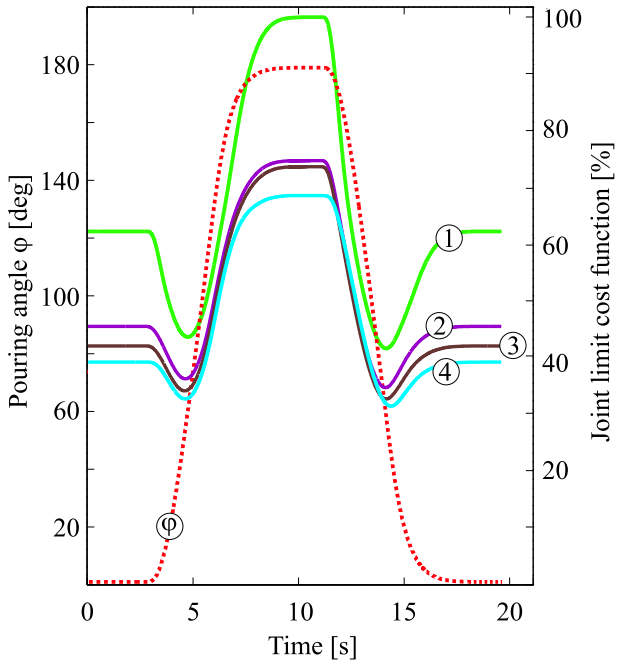

Figure 9: Joint limit avoidance cost function during pouring

since the chosen displacement intervals would allow the task to be carried out successfully.

Cases 1 and 4 have been investigated on the real robot, see figure 10. The upper row shows the motion for the fully constrained case 1, while the lower row shows the motion for the displacement intervals of case 4. While the receiving hand is vertical for case 1, it can nicely be seen that the displacement cone is being fully exploited in case 4. The inclination of the lower hand also leads to both hands moving on a lower height throughout the pouring motion.

VI. Conclusion

The paper describes a flexible whole-body motion method for humanoid robots. It is shown how to represent a task in different ways, describing relative positions of effectors and spatial orientations with one or two degrees of freedom. The inverse kinematics are extended to allow for task-specific displacement intervals around a reference. It is shown how to map optimization criteria into these intervals. The proposed scheme allows the interval boundaries to be dynamically modify at run-time. An illustrative example proves that such intervals significantly increase the quality of the motion. All proposed methods work in real-time and have successfully been tested in simulations and on *ASIMO*.

Acknowledgment

The authors would like to thank the members of Honda's *ASIMO* and robot research teams in Japan for their support.

References

[1] A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," in *IEEE Transactions on Systems, Man, and Cybernetics*, 12 1977, vol. SMC-7 no. 12.

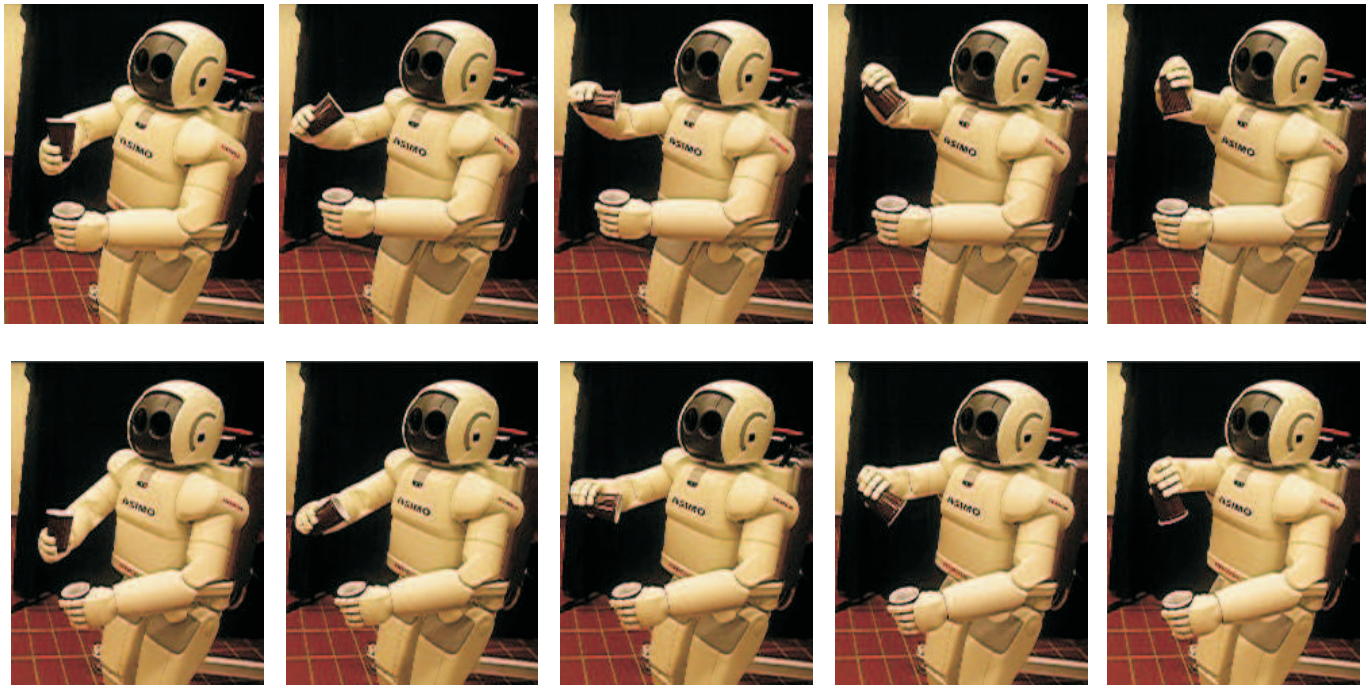

Figure 10: Upper row: Motion according to fully constrained case (1), Lower row: Motion according to case (4)

[2] Y. Nakamura, *AdvancedRobotics: Redundancy and Optimization*, Addison-Wesley Publishing Company, 1991.

[3] S. Buss, "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods," http://www.math.ucsd.edu/ ˜sbuss/ResearchWeb/ikmethods/.

[4] A. Maciejewski, "Dealing with the ill-conditioned equations of motion for articulated figures," *IEEE Comput. Graphics Appl.*, vol. 10, no. 3, pp. 63 – 71, 1990.

[5] G. Tevatia and S. Schaal, "Inverse kinematics for humanoid robots," in *IEEE Conference on Robotics and Automation (ICRA)*, 4 2000.

[6] N. Sian, K. Yokoi, S. Kajita, and K. Tanie, "A framework for remote execution of whole body motions for humanoid robots," in *Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots*, 11 2004.

[7] T. Sugihara and Y. Nakamura, "Whole-body cooperative balancing of humanoid robot using cog jacobian," in *Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.

[8] R. Boulic, R. Mas, and D. Thalmann, "Interactive identification of the center of mass reachable space for an articulated manipulator," in *IEEE International Conference of Advanced Robotics (ICAR)*, 7 1997, pp. 589 – 594.

[9] K. Nishiwaki, M. Kuga, S. Kagami, N. Inaba, and H. Inoue, "Whole-body cooperative balanced motion generation for reaching," in *Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots*, 11 2004.

[10] F. Chaumette and E. Marchand, "A redundancy-based iterative approach for avoiding joint limits: Application to visual servoing," in *IEEE Transactions on Robotics ans Automation*, 10 2001, vol. 17 no. 5.

[11] S. Choi and B. Kim, "Obstace avoidance control for redundant manipulators using collidability measure," *Robotica*, vol. 18, pp. 143 – 151, 2000.

[12] E. Marchand, F. Chaumette, and A. Rizzo, "Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 11 1996, vol. 3, pp. 1083 – 1090.

[13] J. English and A. Maciejewski, "On the implementation of velocity control for kinematically redundant manipulators," in *IEEE Transactions on Systems, Man, and Cybernetics*, May 2000, vol. 30 of *Part A: Systems and Humans*, pp. 233 – 237.

[14] R. Williams, "Local performance optimization for a class of redundant eight-degree-of-freedom manipulators," NASA Technical Paper 3417, NASA, 3 1994.

[15] P. Baerlocher and R. Boulic, "Task-priority formulations for the kinematic control of highly redundant articulated structures," in *IEEE International Conference on Intelligent Robot and Systems (IROS)*, 11 1998, pp. 323 – 329.

[16] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE International Journal of Robotics and Automation*, vol. RA-3, no. 1, pp. 43 – 53, 2 1987.

[17] L. Sentis and O. Khatib, "Prioritized multi-objective dynamics and control of robots in human environments," in *Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots*, 11 2004.

[18] Honda Motor, "Internet page," http://world. honda.com/ASIMO.

[19] M. Gienger, H. Janssen, and C. Goerick, "Task-oriented whole body motion for humanoid robots," in *Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots*, 12 2005.

[20] M. Hirose, Y. Haikawa, T. Takenaka, and K. Hirai, "Development of humanoid robot asimo," in *IEEE/RSJ International Conference on Intelligent Robots and Systems – Workshop 2*, 2001.

[21] H. Bremer, *Dynamik und Regelung mechanischer Systeme*, Teubner Verlag, Stuttgart, 1988.