Whole Body Humanoid Control From Human Motion Descriptors

Behzad Dariush, Michael Gienger, Bing Jian, Christian Goerick, Kikuo Fujimura

2008

Preprint:

This is an accepted article published in Proceedings of the International Conference on Robotics and Automation (ICRA). The final authenticated version is available online at: https://doi.org/[DOI not available]

2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Whole Body Humanoid Control From Human Motion Descriptors

Behzad Dariush Michael Gienger Bing Jian Christian Goerick Kikuo Fujimura

Abstract— **Many advanced motion control strategies developed in robotics use captured human motion data as valuable source of examples to simplify the process of programming or learning complex robot motions. Direct and online control of robots from observed human motion has several inherent challenges. The most important may be the representation of the large number of mechanical degrees of freedom involved in the execution of movement tasks. Attempting to map all such degrees of freedom from a human to a humanoid is a formidable task from an instrumentation and sensing point of view. More importantly, such an approach is incompatible with mechanisms in the central nervous system which are believed to organize or simplify the control of these degrees of freedom during motion execution and motor learning phase. Rather than specifying the desired motion of every degree of freedom for the purpose of motion control, it is important to describe motion by low dimensional motion primitives that are defined in Cartesian (or task) space. In this paper, we formulate the human to humanoid retargeting problem as a task space control problem. The control objective is to track desired task descriptors while satisfying constraints such as joint limits, velocity limits, collision avoidance, and balance. The retargeting algorithm generates the joint space trajectories that are commanded to the robot. We present experimental and simulation results of the retargeting control algorithm on the Honda humanoid robot ASIMO.**

I. INTRODUCTION

In recent years, there has been a growing interest in using captured human motion data as examples to simplify the process of programming or learning complex robot motions [1]. Captured human motion has been used to develop algorithms for "learning from demonstration", a form of learning whereby a robot learns a task by watching the task being performed by a human [2]. One goal of "learning from demonstration" has been to replace the time-consuming manual programming of a robot by an automatic programming process, solely driven by showing the robot the task by an expert teacher. Captured human motion has also been used in computer animation to 'retarget' motion of one articulated figure to another figure with a similar structure [3], [4], [5], [6], [7].

Many existing approaches to robot motion control using observed human motion consist of two separate steps. The first step typically involves a constrained nonlinear optimization procedure that takes as input the human motion in various forms and "retargets" the motion onto a robot's structure [8], [9]. Constraints are typically imposed such that the resulting motion conforms to the robot's kinematic, dynamic, and balance constraints [10]. Given sufficient computation time, a retargetted motion can be realized with a prescribed performance measure subject to the constraints. This procedure is typically processed in batch mode and the resulting motion is then used as the reference trajectory to be executed by the robot's motion controller. Presumably, the retargetted motion is admissible by the robot's structure and can to a certain degree be executed by the robot's control scheme during run-time.

Such a two step approach, consisting of a retargetting procedure followed by joint space control has been demonstrated to work for simple motions [8] and more complex motions [11]. However, these approaches have several drawbacks. First, methods based on nonlinear optimization are computationally expensive for interactive applications and there is no guarantee of convergence to a solution, particulary for high degree of freedom structures. More importantly, there is usually no provision for sensory feedback from the robot's current state to the motion retargetter. Therefore, there is a lack of robustness to disturbances in the environment as well as robustness to parametric and non-parametric uncertainties in the robot model and controller. Perhaps the most important limitation with the existing approaches that use joint space retargetting deals with a lack of formalism to handle the degree of freedom problem, first posed by Bernstein [12]. Bernstein drew attention to the fundamental importance of this problem for explaining the control of behavior. He posed the following difficult but incisive question: How can a neuromuscular system with an exorbitant number of degrees of freedom be made to act as if it had but a few degrees of freedom? Bernstein conjectured that controlled operation of such a system requires a reduction of mechanical redundancy, effectively by reducing the number of degrees of freedom.

Formalizing Bernstein's conjecture into a control structure would allow for the representation of the large number of mechanical degrees of freedom involved in the execution of movement tasks by lower dimensional motion descriptors. We refer to these motion descriptors as task descriptors because they are used to describe motion by higher level task variables. A control policy using task descriptors is generally performed in task space rather than joint space. Such an approach is compatible with Bernstein's hypothesis and current views in motor learning that suggest the central nervous system organizes or simplifies the control of these degrees of freedom during motion execution and motor

B.Dariush and K. Fujimura are at the Honda Research Institute, USA, 800 California St. Suite 300, Mountain View CA 94041 dariush(kfujimura)@honda-ri.com

M. Gienger and C. Goerick are at Honda Research Institute, EU, Carl-Legien-Strae 30 63073 Offenbach/Main Germany, michael.gienger(christian.goerick)@honda-ri.de

B. Jian is with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611 bjian@cise.ufl.edu

learning phase.

Pragmatically, task space control has several inherent advantages and challenges over it's joint space counterpart. The advantages include flexibility in the representation of motion which implies flexibility in the sensing and instrumentation required to capture the human motion. It may not be feasible to fully instrument the human demonstrator with markers, nor is it feasible to instrument the environment with sensors to extract the joint variables which describe all degrees of freedom. Furthermore, the description of the motion may be more appropriately described by Cartesian variables, rather than joint variables. A task space control framework provides flexibility in how the desired motion is parameterized, such that the controlled variables may be represented by position and orientation information describing the task.

In this paper, we present a unified task space control framework to directly control robots from human observations. The proposed approach is based on a control theoretic framework that involves the decomposition of the control structure into tracking control of task descriptors in Cartesian or task space and balance control to ensure that the robot is stable. Kinematic constraints are automatically enforced within this framework.

II. Retargeting

This section describes a task space control framework used to generate motion for all degrees of freedom in our upper body humanoid model from a set of upper body human motion descriptors. The upper body motion descriptors are first normalized (or scaled) to the humanoid robot's dimensions. The objective of the retargeter is to compute the robot's upper body generalized coordinates which enforce the robot's kinematic and dynamic constraints while minimizing the Cartesian tracking error between the normalized human motion descriptors and the corresponding motion descriptors on the humanoid robot's upper body. Although the lower body retargeting is not considered in this paper, we will use the existing balance controller in our robot and a whole body motion control system to ensure that the robot is balanced [13].

A. Kinematic Model

Consider two different articulated body models - a human model and a humanoid robot model. These two models are not necessarily equivalent: that is, they may have different dimensions, physical parameters, and degrees of freedom. We associate a set of desired human motion descriptors with the human model. These descriptors are derived by post-processing the observed motion descriptors detected by our vision system. The post-processing involves low pass filtering, interpolation, and normalization, i.e. re-scaling human data to the humanoid. The desired motion descriptors are expressed in Cartesian (or task) space, and may be referred to as task descriptors to follow the same nomenclature used in task oriented control literature.

Let n represent the number of degrees of freedom to retarget and m be the task dimension. Suppose the robot operates a task descriptor in the full six dimensional task space $(n \geq m, m = 6)$. The position and orientation of the task descriptor is described by, respectively, the vector ${}^{o}p$, and the rotation matrix ${}^{o}R(\Theta)$, where Θ represents the Euler angles. The notation of a leading superscript describes the frame that a quantity is referred to. For simplicity, hereafter, the leading superscript is suppressed for any quantity referred to the base frame. Let the vector $q = [q_1, \cdots, q_n]^T$ describe the degrees of freedom which fully characterize the configuration space, or joint space, of the upper-body humanoid robot. The mapping between the joint space velocities and task space velocities is obtained by considering the differential kinematics relating the two spaces,

$$\dot{x} = J(q)\,\dot{q} \tag{1}$$

where $J \in \Re^{m \times n}$ is the Jacobian of the task descriptor. The spatial velocity vector is defined by $\dot{x} = \begin{bmatrix} w & \dot{p} \end{bmatrix}^T$, where w and $\dot{p}$ are vectors corresponding to the angular velocity of the task frame and the linear velocity of the task position referenced to the base frame, respectively. The angular velocity can be computed from,

$$w = \mathrm{H}(\Theta)\ \dot{\Theta} \tag{2}$$

where the transformation matrix H depends on the particular set of Euler angle sequence considered. The Jacobian matrix may be decomposed to its rotational and translational components, denoted by J_o and J_p, respectively.

$$J = \begin{bmatrix} J_o \\ J_p \end{bmatrix} \tag{3}$$

When the dimension of the task descriptor is described by both position and orientation variables, the Jacobian matrix has six rows $(m = 6)$. If for example, only the position descriptor p is observable, the relevant variables in Equation 1 are $\dot{x} = \dot{p}$, and $J = J_p$.

B. Cartesian Tracking Control

Cartesian tracking control refers to a control policy that produces the joint variables (q) such that the Cartesian error between the computed motion descriptors and the normalized human motion descriptors are minimized. The tracking performance is very much subject to the robot's kinematic constraints as well as the execution of multiple and often conflicting task descriptor requirements. The tracking control approach used here is based on a task space kinematic control method known as closed loop inverse kinematics (CLIK). The basis for the solution of the CLIK algorithm is the differential kinematics relating task variables and joint variables.

Suppose we assign a desired motion of a task descriptor in the full six dimensional space. The associated differential kinematics of the desired motion is expressed by, $\dot{x}_d = \begin{bmatrix} w_d & \dot{p}_d \end{bmatrix}^T$, where the desired angular velocity can be computed from 2. From Equation (1), we can compute the joint velocity vector $\dot{q}$,

$$\dot{q} = J^*\ \dot{x}_d \tag{4}$$

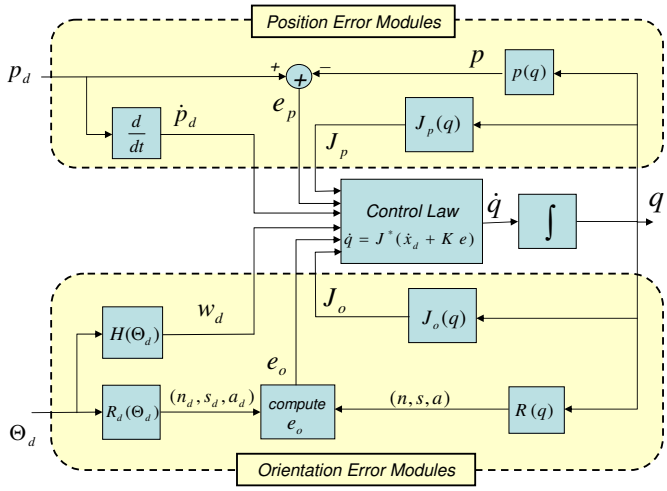

Fig. 1. Detailed system diagram of a first order closed loop inverse kinematics (CLIK1) tracking control with partitioned position and orientation control modules

where J^* denotes the regularized right pseudo-inverse of J weighted by the positive definite matrix W_1 and regularized by the positive definite damping matrix W_2,

$$J^* = W_1^{-1} J^T \ (J W_1^{-1} J^T + W_2)^{-1} \tag{5}$$

The damping matrix is necessary if J is ill-conditioned, the details of which will be discussed in Section II-D.1. If $W_2 = \mathbf{0}$, then Equation (5) is simply the weighted right pseudo-inverse of J. Furthermore, if J is a square non-singular matrix, W_1 is the identity matrix, and $W_2 = 0$, we can simply replace J^* by the standard matrix inversion J^{-1}.

In the numerical implementation of Equation (4), the reconstruction of joint variables q is entrusted to a numerical integration of $\dot{q}$ which may suffer from numerical drift. To avoid this problem, a feedback correction term is often introduced to replace the task descriptor velocity $\dot{x}_d$, by $\dot{x}_d + K\ e$,

$$\dot{q} = J^*(\dot{x}_d + K\ e) \tag{6}$$

where K is a diagonal 6×6 positive definite gain matrix, and $e = [e_o \ e_p]^T$ is a vector that expresses the position error (e_p) and the orientation error (e_o) between the desired and computed task descriptors. The position error is simply defined as $e_p = p_d - p$, where p_d and p correspond to the desired and computed task positions, respectively.

To compute the orientation error, we require the desired task frame. The orientation is typically described by three Euler angles, denoted here by the vector Θ_d. The Euler angles can also be calculated if the desired rotation matrix R_d is known. Let $R_d = [n_d \quad s_d \quad a_d]$ and $R = [n \quad s \quad a]$ correspond to the desired and computed unit vector triple representation of the task frame orientation, respectively. A functional expression of the orientation error in terms of an angle and axis error is given by [14],

$$e_o = \frac{1}{2}(n \times n_d + s \times s_d + a \times a_d) \tag{7}$$

The block-diagram of the first order tracking control is illustrated in Figure 1. This so called closed loop inverse kinematic (CLIK) control algorithm is used in order to arrive at a control command to follow a time-varying desired position and orientation of task descriptors, i.e. tracking control problem.

C. Handling Multiple Tasks

Typically, we are confronted with not just one, but multiple task descriptors which need to be performed simultaneously. Suppose there are k number of sub-tasks that must be executed. Let $\dot{x}_i$ represent the spatial velocity of the i_{th} task descriptor and J_i the associated Jacobian. We present two methods for handling multiple tasks, namely task augmentation and task prioritization.

1) Task Augmentation: Task augmentation refers to the concatenation of the individual spatial velocities $\dot{x}_i$ into a $6k \times 1$ vector $\dot{x}$, and the concatenation of the associated task Jacobian matrix J_i to form the $6k \times n$ matrix J, such that,

$$\dot{x} = [\dot{x}_1 \quad \cdots \quad \dot{x}_k]^T \tag{8}$$

$$J = \begin{bmatrix} J_1^T & \cdots & J_k^T \end{bmatrix}^T \tag{9}$$

Likewise, $\dot{x}_d$ in the augmented space is the concatenation of the each desired task descriptor spatial velocity. The solution of tracking control algorithm in the augmented system follows exactly the same way as that previously described by Equation (6). The tracking error rate for each element of a task descriptor can be controlled by the feedback gain matrices. The trajectory tracking error convergence rate depends on the eigne-values of the feedback gain matrix K in Equation (6); the larger the eignevalues, the faster the convergence. In practice, such systems are implemented as discrete time approximation of the continuous time system; therefore, it is reasonable to predict that an upper bound exists on the eigenvalues; depending on the sampling time. A particular task (or specific directions of particular task) can be more tightly controlled by increasing the eigenvalue of K associated with direction of the particular task.

2) Task Prioritization: Suppose we wish to assign a priority on the execution of a particular task descriptor or groups of task descriptors. For example, suppose we form $i = 1 \cdots k$ sub-groups of tasks (or subtasks). The total task description is therefore composed of k subtasks with the order of priority. For execution of tasks in group i, the differential kinematic relationship between the joint velocity $\dot{q} \in R^n$ and the Cartesian variable $\dot{x}_i$ is expressed by,

$$\dot{q} = J_i^+(q)\dot{x}_i \tag{10}$$

where J_i is the Jacobian matrix of the i_{th} task descriptor and J^+ is typically defined as right pseudo-inverse of J, given by $J^+ = J^T(JJ^T)^{-1}$. A solution for a prioritized two task problem has been outlined in [15]. A prioritized solution can be extended to more than two tasks following the same procedure, as given in Algorithm 1.

A key fact in Algorithm 1 is that all N_i's are orthogonal projectors, which is used to derive the identity $N_{i-1}\hat{J}_i^+ =$

Algorithm 1: General solution for multiple tasks with the order of priority

Input: $J_i \in \Re^{m_i \times n}$, $\dot{x}_i \in \Re^{m_i}$, $i = 1, \ldots, k$, where k is the number of subtasks
Output: $\dot{q}$
begin
  $N_0 = I$
  for $i \leftarrow 1$ **to** k **do**
    $v_i = \dot{x}_i$
    $\hat{J}_i = J_i N_{i-1}$
    $\hat{v}_i = v_i - J_i \sum_{j=1}^{i-1} (\hat{J}_j^+ \hat{v}_j)$
    $N_i = N_{i-1}(I - \hat{J}_i^+ \hat{J}_i)$
  $\dot{q} = \sum_{i=1}^{k} (\hat{J}_i^+ \hat{v}_i) + N_k z$ where $z \in \Re^n$ is an arbitrary vector
end

$\hat{J}_i^+$. Note that $\{N_i\}$ form a sequence of orthogonal projectors with decreasing ranks. The final solution space can be considered as the intersection of the k solution subspaces determined by the k subtasks. Let J be the matrix of size $\sum m_i \times n$ obtained by stacking J_i's, the exact solution space is not empty if and only if J is of full row rank, i.e. $rank(J) = \sum m_i$. However, when the matrix J is rank deficient, i.e. $rank(J) < \sum m_i$, the system can only have solution in the least squares sense and the resulting joint velocity $\dot{q}$ may become very large due to the singularity of J_i. For more detailed discussion on handling this singularity issue, we refer to [16]. A practical approach is to replace the pseudo-inverse $\hat{J}_i^+$ in the last step of the above algorithm by a singularity robust inverse, e.g. the damped least squares inverse [17], which will be discussed in Section II-D.1.

D. Constraints

The humanoid robot typically has kinematic and dynamic constraints which must be enforced. Constraints to avoid joint limits, self collisions, and collisions with the environment are examples of kinematic constraints. Singular configurations also impose constraints on the allowable regions of the workspace that the robot can pass through without regularization. Moreover, the robot may also have limits on the allowable joint velocities and joint torques. In many robotics applications, these constraints are sometimes handled in the null-space for a convenient utilization of redundant degrees of mobility. For our application, there may not be a sufficient number of redundant degrees of freedom to control the internal motion. These constraints will be handled directly in an augmented task space formulation.

1) Handling Singularities : In configurations where the Jacobian matrix in Equation (1) becomes rank deficient, the mechanism is said to be at a singular configuration. In the neighborhood of singular configurations, a small change in x may require a very large change in q. This causes a large error in the task motion, since the joint torques and velocities required to execute such a motion exceed the physical capabilities of the robot.

It is possible to allow the mechanisms to pass through singular points and their neighborhood through the use of a singularity robust inverse of the Jacobian matrix [17], also known as the damped least squares method (DLS) [18]. The most general form of the damped least squares inverse was defined in Equation (5) and repeated below,

$$J^* = W_1^{-1} J^T \ (J W_1^{-1} J^T + W_2)^{-1} \tag{11}$$

where $W_2 = \lambda^2 I$ is the damping term, $\lambda > 0$ is the damping factor, and I is the identity matrix. Small values of λ give accurate solutions but low robustness to the occurrence of singular and near-singular configurations. Large values of λ result in low tracking accuracy even when a feasible and accurate solution would be possible. The damping factor establishes the relative weight between the two objectives. There exists methods for adaptively selecting the damping factor based on some measure of closeness to the singularity at the current configuration [19] [20].

2) Joint Limit Avoidance: Joint limit avoidance is achieved based on a Weighted Least-Norm (WLN) solution, originally proposed by [21]. We utilize a the original WLN solution to come up with the appropriate weighting matrix W_1. However, our solution differs slightly from the original WLN solution since we use the closed loop inverse kinematics solution defined by Equation (6) . Suppose W_1 is an $n \times n$ diagonal matrix with diagonal elements w_i $i = 1 \cdots n$ defined by,

$$w_i = \begin{cases} 1 + |\frac{\partial H}{\partial q_i}| & if \quad \Delta |\frac{\partial H}{\partial q_i}| \geq 0 \\ 1 & if \quad \Delta |\frac{\partial H}{\partial q_i}| < 0 \end{cases} \tag{12}$$

where $H(q)$ is the performance criterion to avoid joint limits and $\frac{\partial H(q)}{\partial q_i}$ is its gradient defined by,

$$\frac{\partial H(q)}{\partial q_i} = \frac{(q_{i,max} - q_{i,min})^2 \ (2q_i - q_{i,max} - q_{i,min})}{4(q_{i,max} - q_i)^2 \ (q_i - q_{i,min})^2}$$

where q_i represents the generalized coordinates of the i_{th} degree of freedom, and $q_{i,min}$ and $q_{i,max}$ are the lower and upper joint limits, respectively. The gradient $\frac{\partial H(q)}{\partial q_i}$ is equal to zero if the joint is at the middle of its range and goes to infinity at either limit. The second condition in Equation (12) allows the joint to move freely if the joint is moving away from the limit because there is no need to restrict or penalize such motions.

3) Joint Velocity Limits: Joint velocity constraints are frequently handled by clamping the velocities when they reach their limit. While such an approach preserve the time required to execute the entire motion, it may not preserve the original trajectory profile. For fast human motions, the re-targeted robot motion profile may become significantly altered as a result of velocity clamping. We propose an alternative method to limit joint velocities by adaptively modulating the time between two successive time samples such that the motion profile q is re-scaled in time, but is not altered in its profile.

To simplify notation, we drop the subscript i, previously referred to quantities associated with joint i (i.e. $q = q_i$ and

$\dot{q} = \dot{q}_i$). Let $\dot{q}_s \ \ (s = 1 \cdots N)$ represent a length N sequence corresponding to a discrete time representation of $\dot{q}(t)$. In the discrete implementation of the algorithm, the discrete time sequence at sample $s + 1$ is given by,

$$t_{s+1} = t_s + \Delta t_s \tag{13}$$

where Δt_s is time between sample s and $s + 1$. To avoid velocity limits, we can replace the time sequence in Equation (13) with the following,

$$t'_{s+1} \ = \ \begin{cases} t'_s + \Delta t_s \ \epsilon_s & if \quad \epsilon_s \geq 1 \\ t'_s + \Delta t_s & if \quad \epsilon_s < 1 \end{cases} \tag{14}$$

where ϵ_s is a time-modulation factor defined by $\epsilon_s = \frac{|\dot{q}_s|}{\dot{q}_{lim}}$, and $\dot{q}_{lim}$ is the joint velocity limit associated with a particular degree of freedom. By definition, $\epsilon_s \geq 1$ implies that the joint velocities are equal or above their limits and corrective action is required by modulating (expanding) time between sample s and $s + 1$. The resulting joint motion with the expanded timescale will conform to the velocity limits, without modifying the shape of the joint motion profile. Furthermore, $\epsilon_s < 1$ implies the joint velocities are below their limits and no time modulation is required. Equation 14 must be performed for each joint at each time-stamp. Note that each joint in a multi-degree of freedom robot may may have a different modulation factor at each instant in time. In order to synchronize the sample time for all joints, a naive, yet simple solution would be to compute ϵ_s for all the joints, and select the largest among them for use in Equation 14.

The time modulation scheme presented above preserves the original motion profile, but may expand the total time required to execute the motion. In order to preserve the motion profile as well as the total execution time, it is possible to design alternative time modulation schemes where Δt_s is expanded when joint velocities exceed their limits and compressed when the joint velocities are below their limits. This may be performed in such a way that the overall execution time remains unchanged. Finally, it should be mentioned that the design requirements for time modulations may require that the first and second order time derivatives of q meet certain smoothness and continuity requirements. In such a case, it is possible to use blending functions into the design of ϵ_s to maintain smoothness.

E. Balance Control

The presented control scheme does not yet consider the constraints that are required to maintain balance during standing and walking. These aspects are not handled within the retargeting framework, but rather by a separate walking and balancing controller that is described in [22], [23]. In detail, the retargeted motion is commanded to the whole body motion controller. The motion generated by the whole body controller will cause some momentum and moment of momentum from a desired reference. This deviation is compensated by the ZMP based balance controller by shifting the upper body in forward- and lateral direction. As depicted in Figure 2, the whole body control and the ZMP control operate cooperatively.

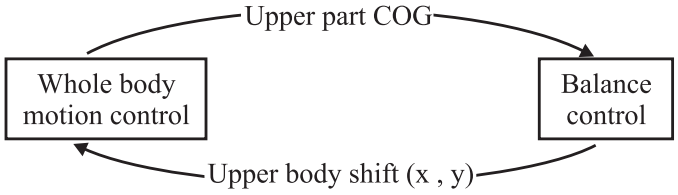

Fig. 2. Separation of balance- and whole body posture control

To account for the body shift, the upper body translational degrees of freedom are incorporated in the kinematic model of the robot. However, they are not actively driven, but rather considered as external input into the controller equations of the whole body control. Whole body control and ZMP control are coupled through momentum- and state feedback, which turns out to be an efficient way to separate these controllers.

III. Motion Interface

We have developed a motion interface to provide a communication link and command interface between off-board computations to generate the retargeted joint commands and the on-board real time control. The motion interface provides a comprehensive way to give motion commands to the robot, without having the user to care about issues such as synchronization, delays, or on-board control for maintaining balance, and collision avoidance. Such an interface is desirable since the real-time implementation requires synchronization between critical control processes that may not be satisfied dependably with a network connection. Issues like balance control, self collision, and other critical aspects are handled within the real-time controller. The motion interface has been successfully used in various other applications, as for instance in [24]. Details of the collision avoidance algorithm is given in [13], [25].

IV. Results

The retargeting algorithm has been tested in simulation for the upper body using an extensive set of simple and complex human motion data. We have also performed experimental results for whole body control of the humanoid robot ASIMO. Our human motion descriptors consist of up to eight upper body Cartesian positions corresponding to the waist joint, two shoulder joints, two elbow joints, and two wrist joints, and the neck joint (see Figure 3).

Two human motion data-sets were used as input to the retargeting algorithm. The first data-set was obtained from the Carnegie Mellon University (CMU) human motion data base [26]. We also obtained human motion data from a real time, marker-less, human pose tracking system developed at Honda Research Institute, USA [27]. For both data-sets, the human motion descriptors were anthropometrically scaled to the ASIMO link segment dimensions.

A. Simulation Results from CMU data-set

The raw human motion data in the CMU data-set is represented in C3D file format, from which we computed

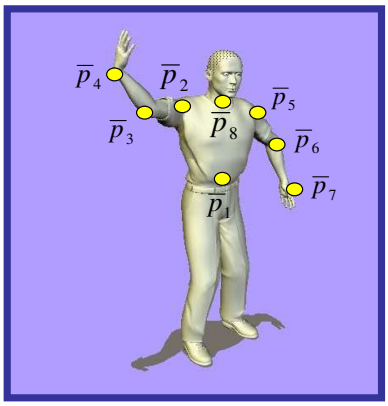

Fig. 3. Eight upper body human motion descriptors, represented by $\bar{p}_i$ are used to control the humanoid robot. These motion descriptors are normalized to the dimensions of the humanoid robot ASIMO. The scaled task descriptors are denoted by p_i in the text.

eight upper body Cartesian positions shown in Figure 3. The raw data was already filtered, and sampled at 120 HZ. Here, we report the results for a drinking motion and a reaching motion.

Figure 4 illustrates the effectiveness of the joint limit avoidance for a drinking motion. The red spheres in the ASIMO model depict the desired (or target) position of the task descriptor and the blue spheres depict the associated task descriptor attached to ASIMO. The upper and lower joint limits are illustrated by the dashed lines, at 0 degrees and -177 degrees, respectively. When joint limit avoidance is turned off (left figure), the computed task descriptor (blue sphere) attached to the wrist can track the desired task descriptor (red sphere) very well. However, the joint limit at the elbow is violated. When joint limit avoidance is turned on, the elbow joint limit is not violated. Since the elbow joint cannot flex beyond its joint limit, the computed wrist task descriptor does not track the desired wrist task descriptor.

Figure 5 illustrates the tracking error results for the CMU reaching motion for prioritizing the eight task descriptors into three priority groups. The high priority group corresponds to the Cartesian position of the waist. The medium priority group corresponds to the Cartesian position of the right and left wrists. Finally, the low priority group corresponds to the position of the elbows, shoulders, and neck. The results clearly illustrate that the prioritized strategy is effective. The non-zero error in the highest priority group is attributed the use of a damping factor used in the damped least squares jacobian inverse. The damping term is effective in handling task singularities as well as algorithmic singularities, but introduces errors in the task position. To minimize the task error, an adaptive damping factor may be used.

B. Experimental Results

Figures 6-8 show snapshots of three different motions retargeted on the ASIMO humanoid robot. The recorded task descriptors for these motions were obtained using a single depth camera human pose tracking system. No markers were attached to the human demonstrator. We used the same eight

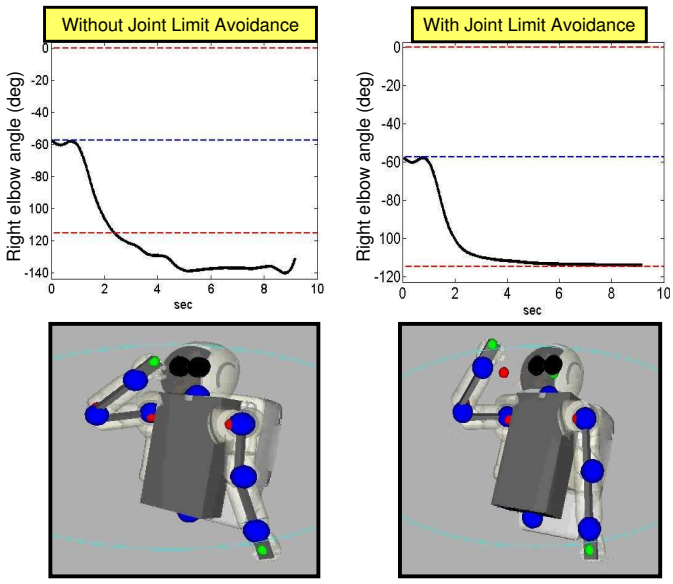

Fig. 4. Illustration of right elbow joint limit avoidance for a drinking motion from the CMU data-set. The upper and lower joint limits are illustrated by the dashed lines, at 0 degrees and -177 degrees, respectively.

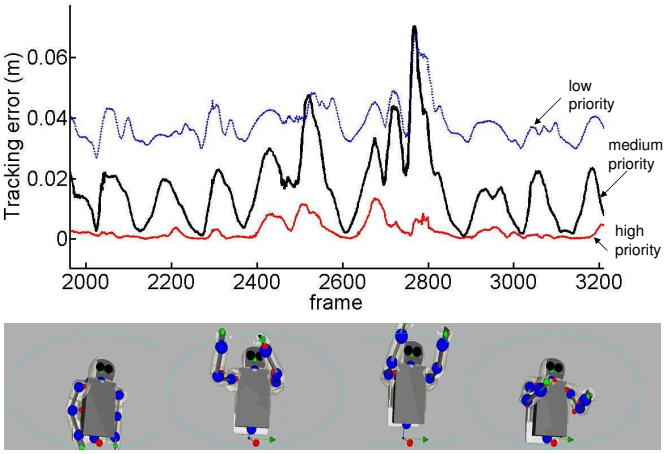

Fig. 5. Illustration of tracking error for a reaching motion from the CMU data-set. The task descriptors are assigned three priority levels. Lower figure illustrates snapshots of the reaching motion on the ASIMO humanoid model.

upper body task descriptors as in the CMU motion set. For online processing, on average, this data generates the human motion descriptors at 4 samples/sec. This data was scaled, low pass filtered, and interpolated to produce the normalized human motion descriptors at 100 samples/sec. The desired motion descriptors are extremely noisy and not very precise as compared to the CMU motion data-set. Nevertheless, we can achieve reasonable whole body control of ASIMO from the retargeted results. The lower body balance controller makes adjustments to the waist in response to motions in the upper body. The slight discrepancy between the observed human motion and the ASIMO motion is attributed to several factors: 1) imprecision by the marker-less visual system in obtaining the exact position of the joint position. 2) errors introduced when scaling data to ASIMO dimensions and 3)

mismatch in the shape and degrees of freedom between the human demonstrator and ASIMO.

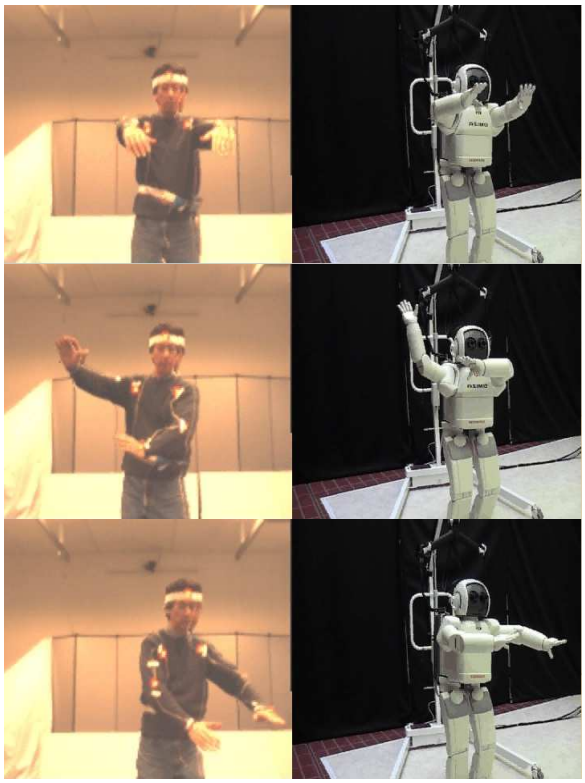

Fig. 6. Taiji Motion

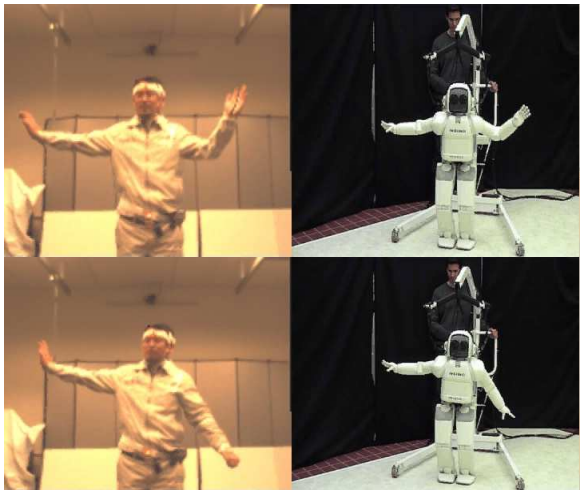

Fig. 7. Policeman Guiding Traffic

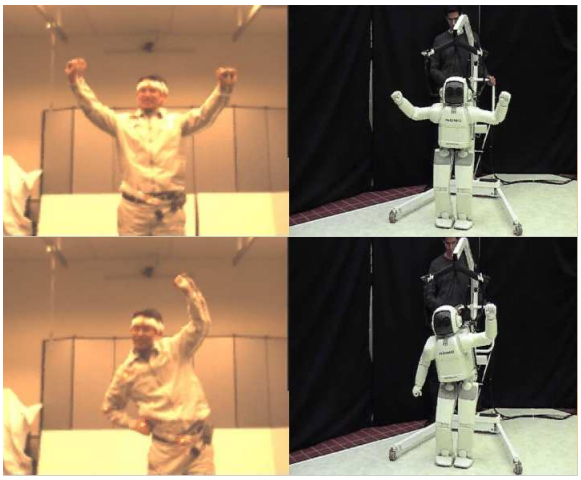

Fig. 8. Exercise Motion

V. Summary and Future Work

The formulation presented in this paper provides preliminary theoretical, simulation, and experimental results toward the development of a unified, control theoretic framework to control robots from observed human motion. The most important attribute of this framework is that it provides a basis for controlling robot motions from a set of high level motion descriptors defined in task space. The task descriptors can be any quantity, measured or inferred, that can be represented as a function of the robot configuration. Since joint variables can be considered as task descriptors, a joint space formulation is a special case of the proposed retargeting control methodology.

We have also developed a prioritized strategy for task management, whereby each task descriptor, or group of task descriptors, may be assigned a different priority. Such an assignment is application specific and depends on the relative importance of a particular task descriptor in executing a given motion. A grasping task, for example, may require assigning a high priority to the position and orientation of the gripper, while other degrees of freedom can be used to avoid obstacles, joint limits, singularities, and to maintain balance. Priority can also be used to encode the level confidence in the measurements of the task variables. Measurements with higher confidence can be assigned a higher priority. Typically, vision systems can more reliably detect and track anatomical landmarks corresponding to the hands and the head, whereas the elbow positions are often difficult to measure reliably.

Although we have reported results of eight upper body task descriptors in this paper, it should be noted that the above formulation can handle arbitrary number of task descriptors. The algorithm is suitable when there is redundant degrees of freedom as well as when the system is over-constrained. In fact, for many of the motions tested, we observed that utilizing as few as four task descriptors (waist, two hands, head) can reproduce realistic and natural looking robot motions. This attribute enables flexibility in sensing and instrumentation required to acquire human motion, as well as flexibility in controlling the robot from a reasonable, yet arbitrary number of task descriptors.

In future work, we plan to incorporate self-collision and obstacle avoidance constraints explicitly in the retargeting algorithm. Currently, collision avoidance is handled in the real-time control system on-board the robot. We also plan to fully integrate the single camera marker-less system with our regtargeting and whole body motion control algorithm

in order to perform online human motion retargeting.

REFERENCES

[1] A. Nakazawa, S. Nakaoka, K. Ikeuchi, and K. Yokoi. Imitating human dance motions through motion structure analysis. In *Intl. Conference on Intelligent Robots and Systems (IROS)*, pages 2539–2544, Lausanne, Switzerland, 2002.

[2] S. Schaal. Learning from demonstration. In M.C. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, chapter 9, pages 1040–1046. MIT Press, 1997.

[3] M. Gleicher. Retargeting motion to new characters. In *Proceedings of SIGGRAPH98*, pages 33–42, ACM, New York, 1998.

[4] J. Lee and S. Y. Shin. A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of SIGGRAPH99*, pages 39–48, ACM, New York, 1999.

[5] S. Tak, O. Song, and H. Ko. Motion balance filtering. *Comput. Graph. Forum. (Eurograhics 2000)*, 19(3):437–446, 2000.

[6] S. Tak and H. Ko. A physically-based motion retargeting filter. *ACM Trans. on Graphics*, 24(1):98–117, 2005.

[7] K. J. Choi and H. S. Ko. On-line motion retargeting. *Journal of Visualization and Computer Animation*, 11(5):223–235, 2000.

[8] A. Ude, C.G. Atkeson, and M. Riley. Programming full-body movements for humanoid robots by observation. *Robotics and Autonomous Systems*, 47:93–108, 2004.

[9] Nancy Pollard, Jessica K Hodgins, M.J. Riley, and Chris Atkeson. Adapting human motion for the control of a humanoid robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '02)*, May 2002.

[10] Y. Tamiya M. Inaba S. Kagami, F. Kanehiro and H. Inoue. Autobalancer: An online dynamic balance compensation scheme for humanoid robots. In *Int. Workshop Alg. Found. Robot.(WAFR)*, Lausanne, Switzerland, 2000.

[11] S. Nakaoka, A. Nakazawa, F. Kanehiro, K. Kaneko, M. Morisawa, H. Hirukawa, and Katsushi Ikeuchi. Learning from observation paradigm: Leg task models for enabling a biped humanoid robot to imitate human dances. *Int. J. Robotics Research*, pages 829–844, 2007.

[12] N. Bernstein. *The coordination and regulation of movements*. Pergamon, London, 1967.

[13] M. Gienger, H. Janssen, and C. Goerick. Task-oriented whole body motion for humanoid robots. In *Proceedings of the 2005 5th IEEE-RAS International Conference on Humanoid Robots*, pages 238–244, Los Angeles, USA, 2005.

[14] J.Y.S. Luh, M.W. Walker, and R.P.C. Paul. Resolved-acceleration control of mechanical manipulators. *IEEE Transactions on Automatic Control*, 25:468–474, 1980.

[15] Y. Nakamura. *Advanced Robotics, Redundancy and Optimization*. Adisson-Wesley, 1991.

[16] B. Siciliano and J. Slotine. A general framework for managing multiple tasks in highly redundantrobotic systems. In *International conference on Advanced Robotics*, volume 2, pages 1211–1216, Pisa, Italy, 1991.

[17] Y. Nakamura and H. Hanafusa. Inverse kinematic solution with singularity robustness for robot manipulator control. *ASME J. Dyn. Sys. Meas., Contr.*, 108(3):163–171, 1986.

[18] C. W. Wampler. Manipulator inverse kinematic solutions based on vector formulations and damped least squares methods. *IEEE Trans. Sys., Man, Cyber.*, 16(1):93–101, 1986.

[19] S. Buss and J. S. Kim. Selectively damped least squares for inverse kinematics. *Journal of Graphics Tools*, 10(3):37–49, 2005.

[20] A. A. Maciejewski and C. A. Klein. Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. *International Journal of Robotics Research*, 4:109–117, 1985.

[21] T. F. Chan and R. V. Dubey. A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. *IEEE Transactions on Robotics and Automation*, 11(2), 1995.

[22] 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*, Hawaii, USA, 2001.

[23] Honda Motor Corp. The Honda humanoid robot ASIMO. Internet page. http://www.world.honda.com/ASIMO.

[24] B. Bolder, M. Dunn, M. Gienger, H. Janssen, H. Sugiura, and C. Goerick. Visually guided whole body interaction. In *Proceedings of the International Conference on Robotics and Automation*, Rome, Italy, 2007.

[25] H. Sugiura, M. Gienger, H. Janssen, and C. Goerick. Real-time collision avoidance with whole body motion control for humanoid robots. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS 2007)*, 2007.

[26] Carnegie Mellon University. CMU graphics lab motion capture database. Internet page. http://mocap.cs.cmu.edu/.

[27] Y. Zhu and K. Fujimura. Constrained optimization for human pose estimation from depth sequences. In *Proceedings of Asian Conference on Computer Vision*, Tokyo, Japan, 2007.