

Kinodynamically Consistent Motion Retargeting for Humanoids

GHASSAN BIN HAMMAM

*Department of Electrical and Computer Engineering, The Ohio State University
Columbus, OH 43210, U.S.A.
bin-hammam.1@osu.edu*

PATRICK M. WENSING

*Department of Mechanical Engineering, Massachusetts Institute of Technology
Cambridge, MA 02139, U.S.A.
pwensing@mit.edu*

BEHZAD DARIUSH

*Honda Research Institute USA
Mountain View, CA 94043, U.S.A.
bdariush@honda-ri.com*

DAVID E. ORIN

*Department of Electrical and Computer Engineering, The Ohio State University
Columbus, OH 43210, U.S.A.
orin.1@osu.edu*

Human-to-humanoid motion retargeting is an important tool to generate human-like humanoid motions. This retargeting problem is often formulated as a Cartesian control problem for the humanoid from a set of task points in the captured human data. Classically, Cartesian control has been developed for redundant systems. While redundancy fundamentally adds new subtask capabilities, the degree to which secondary objectives can be faithfully executed cannot be determined in advance. In fact, a robot that exhibits redundancy with respect to an operational task may have insufficient degrees of freedom to satisfy more critical constraints. In this paper, we present a Cartesian space resolved acceleration control framework to handle execution of operational tasks and constraints for redundant and non-redundant task specifications. The approach is well suited for online control of humanoid robots from captured human motion data expressed by Cartesian variables. The current formulation enforces kinematic constraints such as joint limits, self-collisions, and foot constraints and incorporates a dynamically-consistent redundancy resolution approach to minimize costly joint motions. The efficacy of the proposed algorithm is demonstrated by simulated and real-time experiments of human motion replication on a Honda humanoid robot model. The algorithm closely tracks all input motions while smoothly and automatically transitioning between regimes where different constraints are binding.

Keywords: motion retargeting; motion transfer; task-space control; resolved acceleration control; constraint avoidance; weighted damped-least-squares inverse.

1. Introduction

Humanoid motion generation using human-to-humanoid motion transfer, also known as motion retargeting, is an important and useful method to produce human-like motions. This approach has been utilized often in previous works.¹⁻⁸ Due to kinematic and mass distribution differences between humans and humanoids it is often the case that the original human motion requires adjustment to satisfy the many constraints which limit the motion of the humanoid. These modifications should be designed to be minimal in order to maintain visual similarity to the original motion, yet need to be large enough for the final motion to satisfy the kinematic and dynamic constraints such as joint limits, self-collisions, balance, and torque limits. This paper presents new methods to address joint limits and self-collisions in the performance of human motion retargeting tasks.

A desired motion for retargeting is usually captured using a number of markers in Cartesian space to define motion features.⁹ The positions of these markers, scaled to the geometry of the humanoid, serve as a natural target for control. As a result, task-space control is often used to follow this motion in the humanoid. The task-space, or operational-space, control methodology has been studied extensively for the last decades to coordinate robot motions, whereby the control targets can be represented in task velocity, acceleration,¹⁰ or force/torque domains.¹¹ In typical applications involving task-space control, the robot's number of Degrees of Freedom (DoFs) exceeds what is required to perform the operational tasks. In such circumstances, the robot is said to exhibit redundancy since there exist an infinite number of joint motions that produce the specified task motion. In retargeting, the existence of redundancy depends on the number of markers employed.

In scenarios which exhibit redundancy with respect to the tasks, there is opportunity to achieve other objectives such as avoiding obstacles,^{12,13} avoiding structural limits (e.g., joint limits and self-collisions^{5,14,15}), minimizing energy consumption, creating balanced motions, etc. In early research on redundancy resolution schemes, many important objectives, such as the satisfaction of kinematic constraints, were considered as secondary tasks, and were performed in the null space of the higher priority operational tasks.^{16,17}

In the problem of motion retargeting, however, it is often the case that redundancy does not exist. Still, even in circumstances with redundancy, formulating constraints as secondary tasks does not guarantee that they will be satisfied. In humanoids, satisfaction of these constraints is critical and therefore must be given priority over execution of other tasks. Satisfying these constraints during retargeted human motions is a challenge due to the restrictive joint limits and high likelihood of simultaneous self-collisions in the humanoid. As a result, the number of limiting constraints during a retargeted motion is often large. Real-time motion retargeting provides additional challenges, as the retargeted human motion is unknown a priori and the number of binding constraints often changes as the motion evolves. Therefore, it is not possible to establish in advance whether the robot has sufficient DoFs

to execute the operational tasks while satisfying the constraints.

One suggested solution is to formulate constraints as the highest priority objective and project the operational tasks onto the constraint null space.¹⁸ However, this method handles joint limits and collision avoidance by discontinuously changing the priority of newly binding constraints, which could cause nonsmooth transitions during the execution of the motion. Additionally, since this method holds the joint at its limit, additional mechanisms must determine when it is safe to disengage the constraint from the highest priority. For real-time motion retargeting, joint angles often move to and from their joint limits repeatedly, which makes this method unsuitable.

To handle dynamics constraints, other methods have used convex optimization schemes in order to produce trajectories that satisfy dynamic feasibility (balance), which can be targeted as the highest optimization objective.^{19–21} Methods for joint limits and collision avoidance have yet to be addressed fully in these frameworks. Previous work by Dariush et al.⁵ handles upper-body retargeting only and operates at the velocity level, which makes it incompatible to handle dynamic constraints. Another approach utilizes inverse kinematics with multi-contact operational space,²² but without modeling self-collision constraints. Other motion transfer approaches are discussed in general^{6–8} without dealing with constraints. Offline motion generation methods are proposed without kinematics constraints,⁴ and with inequality constraints only using nonlinear optimization.²³ A recent paper proposed a uniform optimization method to integrate the dynamic balance with inequality constraints such as joint limits; however, their approach is not yet real-time due to computation cost.²⁴ Although their approach should be general to handle self-collision, this generality has not yet been demonstrated and the computational requirements for this important class of constraints have not yet been addressed.

In the new real-time control approaches described here, kinematic constraints are explicitly handled using a weighted damped-least-squares inverse solution at the velocity and acceleration levels. This weighting matrix is strategically designed to dampen motion approaching constraint surfaces while minimally disturbing task tracking. Further, a new strategy is proposed which leads to a dynamically-consistent Jacobian pseudo-inverse when no constraints are binding. The new formulation also incorporates foot contact constraints to obtain a whole-body solution, extending the upper-body capabilities from Dariush et al.²⁵ This approach is shown to enable smooth and efficient transitions between regimes with different binding constraints which are of key importance for human-to-humanoid motion retargeting.

The remainder of the paper is organized as follows. Section 2 provides the background material needed to develop whole-body task-space controllers. Section 3 then presents a method for task-space control at the velocity level which satisfies joint limit and self-collision constraints. A new adaptive pseudo-inverse weighting is proposed to recover the dynamically-consistent pseudo-inverse when no constraints are binding. The following section, Section 4, then describes a method to extend this approach for whole-body constrained resolved acceleration control (CRAC) which

includes contact constraints at the feet. The efficacy of the proposed algorithm is demonstrated in Section 5 through simulated and real-time experiments on the Honda humanoid robot model.²⁶ The reference human motions are complex, fast, and exhibit frequent self-collision and joint limit violations when retargeted onto the robot without provisions for handling the kinematic constraints. Applications of the whole-body CRAC approach are studied when different subsets of motion capture markers are used as the control targets. With each different subset of markers, the humanoid experiences different degrees of redundancy to track the desired tasks. Despite this fact, the CRAC approach described here satisfies constraints in all cases. The paper concludes with a summary in Section 6.

2. Background

The equations of motion of a robotic mechanism in joint space can be written as:

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\tau}_g(\mathbf{q}) - \mathbf{J}_e^T \mathbf{f}_e, \quad (1)$$

where \mathbf{q} , $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$, and $\boldsymbol{\tau}$ denote $(n \times 1)$ generalized vectors of joint position, velocity, acceleration and torque variables, respectively. $\mathbf{H}(\mathbf{q})$ is an $(n \times n)$ joint-space inertia matrix. \mathbf{C} is the $(n \times 1)$ vector of Coriolis and centrifugal terms. $\boldsymbol{\tau}_g$ is the $(n \times 1)$ vector of gravity terms. \mathbf{J}_e is a Jacobian matrix, and \mathbf{f}_e is the external force acting on the system.

In the absence of an external force acting on the system, control of the system described by Eq. 1 can be handled using a nonlinear model-based compensation approach with the following structure,

$$\boldsymbol{\tau} = \hat{\mathbf{H}}(\mathbf{q}) \boldsymbol{\alpha} + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \hat{\boldsymbol{\tau}}_g(\mathbf{q}). \quad (2)$$

The vector $\boldsymbol{\alpha}$ represents a *resolved acceleration* in terms of joint variables. The notation $\hat{\cdot}$ denotes estimates of the components of the dynamic model. Provided the model parameters in Eq. 2 match those of Eq. 1, the closed-loop dynamics become linearized and decoupled with,

$$\ddot{\mathbf{q}} = \boldsymbol{\alpha}. \quad (3)$$

In practice, additional stabilizing terms may be added to the control law in Equation 2.²⁷ The literature on task-level tracking control of robotic systems offers a variety of techniques for computing $\boldsymbol{\alpha}$. The underlying mechanism for such control techniques is referred to as resolved acceleration control. The aim is to design a control law to ensure tracking of a desired position and/or orientation of a set of task bodies specified in Cartesian-space.

Suppose there are N task bodies, with a total task dimension of m . In general, each task body can contribute up to six task dimensions, three for position and three for orientation. Indexing the task bodies by k , the spatial velocity vector of the k_{th} task body is given by,

$$\mathbf{v}_k = [\boldsymbol{\omega}_k^T \dot{\mathbf{p}}_k^T]^T, \quad (4)$$

where $\boldsymbol{\omega}_k$ and $\dot{\boldsymbol{p}}_k$ are vectors corresponding to the angular and linear velocity of a task frame fixed to the task body. The mapping between joint variables and task variables is obtained by considering the differential kinematics,

$$\mathbf{v} = \mathbf{J} \dot{\mathbf{q}}, \quad (5)$$

$$\mathbf{a} = \mathbf{J} \ddot{\mathbf{q}} + \dot{\mathbf{J}} \dot{\mathbf{q}}, \quad (6)$$

where $\mathbf{a} = \dot{\mathbf{v}}$, and \mathbf{v} and \mathbf{J} correspond to the $(m \times 1)$ augmented spatial velocity vector and the $(m \times n)$ augmented task Jacobian matrix formed by concatenation of the individual tasks,

$$\mathbf{v} = [\mathbf{v}_1^T \quad \cdots \quad \mathbf{v}_k^T \quad \cdots \quad \mathbf{v}_N^T]^T, \quad (7)$$

$$\mathbf{J} = [\mathbf{J}_1^T \quad \cdots \quad \mathbf{J}_k^T \quad \cdots \quad \mathbf{J}_N^T]^T. \quad (8)$$

The augmented desired spatial velocity and acceleration vectors, denoted by $(\mathbf{v}_d, \mathbf{a}_d)$, can be constructed in the same fashion.

While control at the velocity level is simple to implement and can exhibit a large amount of practical robustness, it still has many disadvantages over acceleration control. A recent study comparing different task-space controllers at the velocity and acceleration levels concluded that task-space control at the acceleration level generally fared the best in terms of tracking performance, ease of parameter tuning, and general robustness and compliance.²⁷

An effective method to decouple and linearize the closed-loop task dynamics is to form the resolved acceleration control input $\boldsymbol{\alpha}$ in Eq. 3 using the relationship,

$$\boldsymbol{\alpha} = \mathbf{J}^\dagger (\mathbf{a}^* - \dot{\mathbf{J}}\dot{\mathbf{q}}) + \boldsymbol{\alpha}_n. \quad (9)$$

Here, $\boldsymbol{\alpha}_n$ is a joint acceleration vector lying in the null space of \mathbf{J} , which is available for redundancy resolution, traditionally to achieve additional objectives such as obstacle avoidance and joint limit avoidance. \mathbf{J}^\dagger denotes the damped-least-squares pseudo-inverse of \mathbf{J} and may be weighted by a positive definite matrix \mathbf{W} ,

$$\mathbf{J}^\dagger = \mathbf{W}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{W}^{-1} \mathbf{J}^T + \lambda^2 \mathbf{I})^{-1}, \quad (10)$$

where \mathbf{I} is the identity matrix, and λ is the damping factor. Also in Eq. 9, \mathbf{a}^* is an augmented spatial acceleration vector derived from the body feedback laws,

$$\mathbf{a}_k^* = \mathbf{a}_{k,d} + \mathbf{K}_v (\mathbf{v}_{k,d} - \mathbf{v}_k) + \mathbf{K}_p \mathbf{e}_k, \quad (11)$$

where \mathbf{K}_p and \mathbf{K}_v are positive definite diagonal position and velocity feedback gain matrices, respectively. For each task body, the error vector \mathbf{e}_k describes the orientation and position error between the desired and computed task descriptors,^{10,28}

$$\mathbf{e}_k = \begin{bmatrix} \frac{1}{2} (\mathbf{n} \times \mathbf{n}_d + \mathbf{s} \times \mathbf{s}_d + \mathbf{c} \times \mathbf{c}_d) \\ \mathbf{p}_d - \mathbf{p} \end{bmatrix}, \quad (12)$$

where $\mathbf{R}_d = [\mathbf{n}_d \quad \mathbf{s}_d \quad \mathbf{c}_d]$ and $\mathbf{R} = [\mathbf{n} \quad \mathbf{s} \quad \mathbf{c}]$ correspond to the desired and computed unit vector triple representation of the task frame orientation, respectively.

The solution in Eq. 9 using Eq. 10 with $\alpha_n = \mathbf{0}$ realizes the following objective,¹⁶

$$\arg \min_{\alpha} \|\mathbf{x}\|^2 + \lambda^2 \|\alpha\|_{\mathbf{W}}^2 \quad (13)$$

$$\text{where } \mathbf{x} = \mathbf{a}^* - \dot{\mathbf{J}}\dot{\mathbf{q}} - \mathbf{J}\alpha \quad (14)$$

where $\|\cdot\|_{\mathbf{W}}$ implies the \mathbf{W} weighted norm. As with all pseudo-inverses, Eq. 10 is applicable to all cases regardless of the dimensions or rank of \mathbf{J} .¹⁶ In the following sections, methods are described which show how \mathbf{W} can be chosen adaptively in order to satisfy joint limit and self-collision constraints.

3. Kinematic Constraints

The second-order inverse kinematics procedure described by Eq. 9 is an effective method to perform trajectory conversion from task space to joint space.²⁸ To a certain degree, the null-space terms can be designed to perform secondary objectives such as keeping the robot away from kinematic constraints. In these methods, the gradients of a potential function are projected into the null space of the Jacobian and constraints are handled as secondary tasks. Therefore, constraints could be violated if the null space of the task Jacobian has a dimension of zero.

In our earlier work, we described an algorithm for solving the first-order constrained closed-loop inverse kinematics (CCLIK) problem which proved to be an effective and stable solution for self-collision avoidance.^{5,29} The method used a weighted least-squares solution to constrain joints from violating their limits and to avoid collisions, even in the absence of task redundancy. In this section, we present an overview of the CCLIK method which will be used in the derivation of our constrained resolved acceleration control formulation described in Section 4.

We define an error corrected spatial velocity vector, derived from feedback correction of the desired spacial velocity,

$$\mathbf{v}^* = (\mathbf{v}_d + \mathbf{K}_p \mathbf{e}) . \quad (15)$$

A first-order closed-loop inverse kinematics solution can be obtained as follows:^{5,29}

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \mathbf{v}^* . \quad (16)$$

Within the pseudo-inverse \mathbf{J}^\dagger , the weight matrix \mathbf{W} may be appropriately designed to enforce kinematic constraints by penalizing those joints whose motion directs the segments toward constraints. Kinematic constraints include joint limit constraints as well as constraints for self-collision (or self-penetration). The weight matrix can be further conditioned to achieve dynamic consistency (minimization of kinetic energy). While dynamically-consistent Jacobian pseudo-inverses were initially introduced for operational-space task decoupling,¹¹ here their use enables movements which require reduced power by penalizing the motions of more massive bodies. To achieve both kinematic and dynamic consistency, we construct \mathbf{W} as the sum of weight matrices,

$$\mathbf{W} = \mathbf{W}_K + \mathbf{W}_D , \quad (17)$$

where \mathbf{W}_K is a positive-semidefinite weight matrix to enforce kinematic constraints, and \mathbf{W}_D is a positive definite weight matrix to achieve dynamic consistency. In the following, we describe the design of the weight matrices in further detail.

We construct \mathbf{W}_K as a diagonal matrix whose elements are derived by considering the gradients of the joint limit and collision potential functions. The matrix \mathbf{W}_K is influenced by the $n \times n$ joint limit weight matrix \mathbf{W}_{JL} and the $n \times n$ collision avoidance weight matrix \mathbf{W}_{COL} . While a rigorous formulation of the influence of each constraint is warranted and is currently being examined, we present a simple and effective solution based on our empirical results. The proposed composite weight matrix is given by,

$$\mathbf{W}_K = a \mathbf{W}_{JL} + (1 - a) \mathbf{W}_{COL}, \quad (18)$$

where a is a scalar index which can be used to modulate the contribution of the joint limit weight matrix and the collision weight matrix. We have found that the following selection is effective for the various motions considered,

$$a = \frac{1}{(N_c + 1)}, \quad (19)$$

where N_c corresponds to the number of pairs of body segments checked for self-collision explained in Section 3.2. To construct \mathbf{W}_{JL} and \mathbf{W}_{COL} , we consider candidate joint limit and collision potential functions, denoted by $h(\mathbf{q})$ and $f(\mathbf{q})$, respectively.^{5, 25, 29}

3.1. Joint Limit Constraints

Joint limit avoidance may be achieved by the proper selection of the diagonal matrix \mathbf{W}_{JL} .^{5, 30} To construct \mathbf{W}_{JL} , we consider a candidate joint limit potential function, denoted by $h(\mathbf{q})$, that tends to infinity at the joint limits. The gradient of h , denoted as ∇h , represents the joint limit gradient function, an $n \times 1$ vector whose entries point in the direction of the fastest rate of increase of h . The gradient term associated with the i_{th} ($i = 1 \dots n$) degree of freedom is denoted by,

$$\nabla h_i = \frac{\partial h(\mathbf{q})}{\partial q_i}, \quad (20)$$

and described as follows,³⁰

$$\nabla h_i = \frac{(q_{i,max} - q_{i,min})^2 (2q_i - q_{i,max} - q_{i,min})}{4(q_{i,max} - q_i)^2 (q_i - q_{i,min})^2},$$

where q_i represents the generalized coordinate 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 ∇h_i is equal to zero if the joint is at the middle of its range and goes to infinity at either limit. As described in³⁰, the joint limit weight matrix \mathbf{W}_{JL} is constructed as an $n \times n$ diagonal matrix with diagonal elements w_{JL_i} . Each scalar w_{JL_i} is defined

by,

$$w_{JL_i} = \begin{cases} |\nabla h_i| & \text{if } \frac{d}{dt}|\nabla h_i| \geq 0, \\ 0 & \text{if } \frac{d}{dt}|\nabla h_i| < 0. \end{cases} \quad (21)$$

The term $\frac{d}{dt}|\nabla h_i|$ represents the rate of change in the magnitude of the joint limit gradient function. A positive value indicates the joint is moving toward its limit while a negative value indicates the joint is moving away from its limit. Intuitively, when this weighting matrix is used in the damped pseudo-inverse (Eq. 16) at the velocity level, large weights minimize joint motions on DoFs approaching their limits. As a joint moves towards its limit, the weight becomes very large and causes the joint to slow down. When the joint is about to hit its limit, this weight is near infinity and causes the joint to virtually stop. If a joint is moving away from its limit, Eq. 21 imposes no motion penalty, and the joint is able to move freely.

3.2. Collision Constraints

Constructing the appropriate collision weight matrix \mathbf{W}_{COL} is more complex. Consider collision between two unconnected segments (segments which do not share a joint). Let $d(\mathbf{q}) \geq 0$ correspond to the minimum distance between the two bodies. Let $g(d)$ represent a candidate collision function that has a maximum value at $d = 0$ and decays exponentially toward zero as d increases. Through composition of these functions, a configuration-space candidate collision function can be constructed:

$$f(\mathbf{q}) = (g \circ d)(\mathbf{q}) = g(d(\mathbf{q})).$$

We define the gradient of f , denoted as ∇f , as the collision gradient function, an $n \times 1$ vector whose entries point in the direction of the fastest rate of increase of f . The collision gradient function may be described as,

$$\nabla f = \frac{\partial f}{\partial \mathbf{q}} = \frac{\partial g}{\partial d} \frac{\partial d}{\partial \mathbf{q}}. \quad (22)$$

In case of self-collisions, the second term in Eq. 22 may be computed as follows,

$$\frac{\partial d}{\partial \mathbf{q}} = \frac{1}{d} [\mathbf{J}_a^T (\mathbf{p}_a - \mathbf{p}_b) + \mathbf{J}_b^T (\mathbf{p}_b - \mathbf{p}_a)]^T, \quad (23)$$

where \mathbf{p}_a and \mathbf{p}_b represent position vectors, referred to the base, of the two collision points, and \mathbf{J}_a and \mathbf{J}_b are the associated Jacobian matrices. The coordinates \mathbf{p}_a and \mathbf{p}_b can be obtained using a standard collision detection software package.³¹ In case of collision with the environment, the Jacobian associated with the environment collision point is zero. Similar to the joint limit matrix, \mathbf{W}_{COL} may be constructed by an $n \times n$ diagonal matrix with diagonal elements w_{COL_i} ($i = 1 \dots n$) defined by,

$$w_{COL_i} = \begin{cases} |\nabla f_i| & \text{if } \frac{d}{dt}|\nabla f_i| \geq 0, \\ 0 & \text{if } \frac{d}{dt}|\nabla f_i| < 0. \end{cases} \quad (24)$$

The elements of ∇f represent the degree to which each degree of freedom influences the distance to collision. It is appropriate to select a collision function $g(d)$ such

that its gradient is zero when d is large and infinity when d approaches zero. One such candidate function is,

$$g(d) = \rho e^{-\alpha d} d^{-\beta}, \quad (25)$$

where α and β are parameters to control the rate of decay and ρ controls the amplitude. The derivative of g with respect to d is,

$$\frac{\partial g}{\partial d} = -\rho e^{-\alpha d} d^{-\beta} (\beta d^{-1} + \alpha). \quad (26)$$

It follows that ∇f may be computed from Eqs. 22, 23, and 26.

The term $\frac{d}{dt}|\nabla f|$ in Eq. 24 represents the rate of change in the magnitude of the collision gradient function. A positive value indicates the joint motion is causing the collision point to move toward collision while a negative value indicates the joint motion is causing the collision point to move away from collision. When a collision point is moving toward collision, the associated weight factor, described by the first condition in Eq. 24, becomes very large. When this weighting is used in a pseudo-inverse (Eq. 16) at the velocity level, it causes the joints affecting the collision to slow down. When two segments are about to collide, the weight factor is near infinity and the joints contributing to collision then virtually stop. If two segments are moving away from collision, there is no need to restrict or penalize the motions. In this scenario, the second condition in Eq. 24 allows the joint to move freely.

Suppose instead that a total of N_c segment pairs are checked for self-collision. Let j ($j = 1 \dots N_c$) be the index of the j_{th} collision pair, and \mathbf{W}_{COL_j} be the collision weight formed by the above procedure. These weightings are combined simply as:

$$\mathbf{W}_{COL} = \frac{1}{N_c} \sum_{j=1}^{N_c} \mathbf{W}_{COL_j}. \quad (27)$$

3.3. Incorporation of Dynamic Consistency

The weight matrix \mathbf{W} can be used to realize additional performance objectives. In particular, the instantaneous kinetic energy may be minimized by choosing the joint-space inertia matrix \mathbf{H} as the weight matrix in Eq. 10,

$$\mathbf{J}^\dagger = \mathbf{H}^{-1} \mathbf{J}^T \left(\mathbf{J} \mathbf{H}^{-1} \mathbf{J}^T + \lambda^2 \mathbf{I} \right)^{-1}. \quad (28)$$

When the damping term (λ) in Eq. 28 approaches zero, the inverse becomes the inertia weighted inverse. This inverse is called the *dynamically-consistent* generalized inverse.^{32,33} This inverse optimizes the instantaneous kinetic energy when it is used in the solution of the first-order inverse kinematics (Eq. 16), and minimizes acceleration energy when it is used in the second-order inverse kinematics (Eq. 9). It has been argued to provide a “natural” choice for redundancy resolution in operational-space control³⁴ due to its connections with the Gauss’ principle of least constraint. In addition to these motivations, our experiments in this work show that minimizing

both the kinetic and acceleration energy helps to reduce the average absolute joint power as well.

To achieve both kinematic consistency (satisfying kinematic constraints) and dynamic consistency (minimizing kinetic/acceleration energy), two sub-weightings must be combined in the weight matrix of Eq. 17. An effective solution is to set \mathbf{W}_D as,

$$\mathbf{W}_D = \mathbf{H}. \quad (29)$$

It should be noted that when the system is away (or moving away) from kinematic constraints, the diagonal matrix \mathbf{W}_K is the zero matrix. In this case, if $\mathbf{W}_D = \mathbf{H}$, Eq. 16 results in a minimization of the instantaneous kinetic energy. When the system approaches a kinematic constraint, the diagonal elements of \mathbf{W}_K , which are attributed to motion in the direction of the constraints, will become increasingly large and the associated joint velocities will approach zero. In this case, the influence of very large elements in \mathbf{W}_K will have a dominant effect over those in \mathbf{W}_D , and the higher priority objective to satisfy the kinematic constraints will be executed.

4. Constrained Resolved Acceleration Control

This section will describe how the CCLIK approach from the previous section can be extended for acceleration-level control. Additionally, methods are presented which deal with foot contact constraints, which enables application for whole-body control.

4.1. Basic Approach

Let $\boldsymbol{\alpha}_c$ be a vector corresponding to the constrained resolved accelerations, which may be folded into the dynamic equations of motion as in the control law of Eq. 2,

$$\boldsymbol{\tau} = \hat{\mathbf{H}}(\mathbf{q}) \boldsymbol{\alpha}_c + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \hat{\boldsymbol{\tau}}_g(\mathbf{q}). \quad (30)$$

We construct $\boldsymbol{\alpha}_c$ as follows,

$$\boldsymbol{\alpha}_c = \boldsymbol{\alpha} + \mathbf{G}_v (\dot{\mathbf{q}}_c - \dot{\mathbf{q}}), \quad (31)$$

where $\dot{\mathbf{q}}_c$ is computed from the first-order constrained kinematics inversion of Eq. 16, and \mathbf{G}_v is a diagonal positive definite joint velocity gain matrix. For systems which exhibit redundancy, it can also be advantageous to exploit the redundancy and include a null-space projection term when computing $\boldsymbol{\alpha}$, such that,

$$\boldsymbol{\alpha} = \mathbf{J}^\dagger (\mathbf{a}^* - \dot{\mathbf{J}}\dot{\mathbf{q}}) + \mathbf{N}\boldsymbol{\zeta}, \quad (32)$$

where $\boldsymbol{\zeta}$ is an arbitrary vector and $\mathbf{N} = \mathbf{I} - \mathbf{J}^\dagger \mathbf{J}$ projects $\boldsymbol{\zeta}$ into the null space of \mathbf{J} . The vector $\boldsymbol{\zeta}$ may be designed to achieve other desired objectives.

The closed-loop task error dynamics resulting from the control law Eq. 31 are,

$$\ddot{\mathbf{e}} + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = -\mathbf{J}\mathbf{G}_v (\dot{\mathbf{q}}_c - \dot{\mathbf{q}}). \quad (33)$$

The left side of Eq. 33 describes the task error dynamics, which exhibit asymptotic convergence to zero when the right side is zero. The right side characterizes the constraint error dynamics, i.e. the error in tracking the reference constraint velocities $\dot{\mathbf{q}}_c$. Equation 33 reveals that as long as $\dot{\mathbf{q}}_c - \dot{\mathbf{q}}$ is non-zero, the terms on the right side will interfere with the task-space tracking error dynamics. Proper selection of \mathbf{G}_v is important, as larger gains promote faster convergence to the reference constraint velocities which reduces interference with the task tracking. However, an excessively large \mathbf{G}_v may introduce instability in practical digital control implementations.

Let us examine the effect of \mathbf{W} in the solution of $\boldsymbol{\alpha}$ in Eq. 32. Recall that in the first-order inverse kinematics solution described by Eq. 16, the weight matrix \mathbf{W} forced any joint velocity contributing to violation of a kinematic constraint to zero. In contrast, the use of the constraint matrix \mathbf{W} in Eq. 32 forces the acceleration $\boldsymbol{\alpha}$ contributing to violation of a kinematic constraint to zero. The addition of reference velocity feedback in Eq. 31, however, effectively dampens motion as the robot configuration approaches the constraint manifold. To see this more exactly, consider, for instance, joint i approaching its limit. As this occurs, its diagonal entry of \mathbf{W}_K becomes increasingly large. The damped inverses used to compute $\boldsymbol{\alpha}$ and $\dot{\mathbf{q}}_c$ in Eqs. 32 and 16 then force the i -th components of $\boldsymbol{\alpha}$ and $\dot{\mathbf{q}}_c$ towards zero. As a result, the joint damping provided through \mathbf{G}_v dominates the component of $\boldsymbol{\alpha}_c$ for this joint, effectively stopping its motion.

Equation 31 constitutes the generalized constrained resolved acceleration vector. The solution is designed to satisfy constraints even when there are insufficient DoFs to execute both the constraints and the operational tasks. In contrast, approaches that use null-space gradient projections only to satisfy secondary objectives break down when the Jacobian of the primary objective becomes rank deficient due to task singularities.

4.2. Incorporation of Foot Constraints

For whole-body motion control of legged humanoid robots, we must also enforce the foot contact constraints. The position/orientation of each foot can be treated as a task variable and controlled in a prioritized redundancy resolution scheme with two sub-tasks. The higher priority sub-tasks enforce the foot contact constraints for the lower body, while the lower priority sub-tasks operate in the null space of the foot constraints in order to handle tracking of upper-body operational tasks. We therefore divide the resolved acceleration vector into two prioritized sub-tasks,

$$\boldsymbol{\alpha} = \boldsymbol{\alpha}_{LB} + \boldsymbol{\alpha}_{UB}, \quad (34)$$

where $\boldsymbol{\alpha}_{LB}$ and $\boldsymbol{\alpha}_{UB}$ are the whole-body resolved accelerations for the lower-body (high priority) and upper-body (low priority) sub-tasks, respectively.

Let \mathbf{a}_f^* (obtained from Eq. 11) be the error corrected acceleration of the foot

(right and left) sub-tasks,

$$\mathbf{a}_f^* = \begin{bmatrix} \mathbf{a}_{rf}^* \\ \mathbf{a}_{lf}^* \end{bmatrix}, \quad (35)$$

where rf and lf denote right and left foot, respectively. Note that when the feet are fixed on the ground, $\mathbf{a}_{f,d} = \mathbf{v}_{f,d} = \mathbf{0}$. The solution of the lower-body sub-task α_{LB} can be computed from Eq. 9 as follows,

$$\alpha_{LB} = \mathbf{J}_f^\dagger \left(\mathbf{a}_f^* - \dot{\mathbf{J}}_f \dot{\mathbf{q}} \right), \quad (36)$$

where,

$$\mathbf{J}_f = [\mathbf{J}_{rf}^T \quad \mathbf{J}_{lf}^T]^T, \quad (37)$$

is the augmented Jacobian for right and left feet. The generalized inverse, \mathbf{J}_f^\dagger , is weighted to achieve kinematic and dynamic consistency as described in Eq. 17. Joint limits at the knee can be selected to allow the robot to achieve a nearly upright stance and produce more natural and energy efficient motions.

Here, it is assumed that all motion capture markers are located on the upper body, since ground contact constraints utilize all the available motion freedoms for the legs. When joint limit or collision constraints are active in the legs, upper-body motions may be limited, however, due to limitations in torso movement. Let \mathbf{a}_u^* be the error corrected acceleration of the upper-body sub-tasks including the head, arms, and torso. The solution to the upper-body (second priority) sub-task lies on the null space of the augmented feet Jacobian,

$$\alpha_{UB} = \hat{\mathbf{J}}_u^\dagger (\mathbf{a}_u^* - \dot{\mathbf{J}}_u \dot{\mathbf{q}} - \mathbf{J}_u \alpha_{LB}), \quad (38)$$

$$\hat{\mathbf{J}}_u = \mathbf{J}_u \left(\mathbf{I} - \mathbf{J}_f^\dagger \mathbf{J}_f \right), \quad (39)$$

where

$$\mathbf{J}_u = [\mathbf{J}_1^T \quad \dots \quad \mathbf{J}_k^T \quad \dots \quad \mathbf{J}_{N_u}^T]^T \quad (40)$$

is the augmented Jacobian for upper-body and torso tasks. The parameter N_u denotes the number of upper-body tasks. Note that despite the contact constraints being treated as a first priority, the results show that this prioritization does not interfere with the satisfaction of joint limits and self-collision constraints in whole-body motions using the adaptive-weighting strategy proposed.

It then follows that the whole-body constrained resolved acceleration vector with foot contact constraints can be obtained from Eq. 31, where α is determined from Eqs. 34, 36, and 38, and the solution for $\dot{\mathbf{q}}_c$ must take into account the foot constraints in a two priority sub-task solution,¹²

$$\dot{\mathbf{q}}_c = \mathbf{J}_f^\dagger \mathbf{v}_f^* + \hat{\mathbf{J}}_u^\dagger \left(\mathbf{v}_u^* - \mathbf{J}_u \mathbf{J}_f^\dagger \mathbf{v}_f^* \right). \quad (41)$$

The vectors \mathbf{v}_f^* and \mathbf{v}_u^* are determined from Eq. 15. The first term in Eq. 41 describes the highest priority objective to enforce the foot constraint. The second term lies

on the null space of the primary objective and is included to provide tracking of the upper-body operational tasks. In addition, term \mathbf{N} in Eq. 32 can be computed as,

$$\mathbf{N} = \mathbf{N}_f \mathbf{N}_u, \quad (42)$$

where \mathbf{N}_f and \mathbf{N}_u are null-space matrices of \mathbf{J}_f and $\hat{\mathbf{J}}_u$, respectively. Once again, the generalized inverses \mathbf{J}_f^\dagger and $\hat{\mathbf{J}}_u^\dagger$, described above are weighted by \mathbf{W} to satisfy the kinematic constraints and achieve dynamic consistency.

5. Whole-Body Experiments

The proposed control system for human motion retargeting was demonstrated on the 29 DoF Honda-like Humanoid robot model.²⁶ This system has a total mass of 43 kg, with mass distribution given in Table 5. As can be seen, the torso is the heaviest individual link in the humanoid and is connected without a waist joint. For general modeling of various phases of locomotion, including flight, the torso’s motion is referenced to the ground through a 6-DoF un-actuated floating-base joint.³⁵ The

Table 1. Humanoid robot model’s DoFs and mass distribution.

	DoFs	Mass % of total
Torso	6 floating-base	44
Arms	3 × 2 Shoulders 1 × 2 Elbows 1 × 2 Wrists	8
Head	1 Neck	3
Legs	3 × 2 Hips 1 × 2 Knees 2 × 2 Ankles	45
System Total	29	43 kg

efficacy of the proposed control system was examined through pre-recorded as well as online motion capture sequences. A set of eight upper-body Cartesian positions, shown in Figure 1, were extracted from the motions. These positions correspond to the waist, two shoulder joints, two elbow joints, two wrist joints, and the neck joint. Further, the position and orientation of each foot define six additional task variables, used to constrain the foot’s movement. The extracted positions were low pass filtered, interpolated, and scaled to the robot dimensions. The processed motion then corresponded to a set of possible desired tasks, which were to be tracked using the CRAC framework. We performed experiments using three possible task sets, as described below. In all experiments, the feet were constrained to maintain a side-by-side double support stance. In counting the degree of redundancy to achieve the tasks, we should notice that there are a number of inactive DoFs (N_{inact}), such as the neck and wrist joints, which do not affect any task dynamics. These DoFs correspond to columns in \mathbf{J} which are entirely zero.

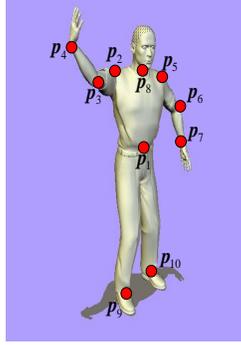


Fig. 1. Task descriptors.

(i) **Task Set 1: Over-constrained tasks**

The specified task involves all position descriptors and also the orientation for the feet. The task dimension for this setup is $m = 3 \times 8 + 2 \times 6 = 36$, thus it is inherently over-constrained.

(ii) **Task Set 2: Five degrees of redundancy**

The specified task involves three position descriptors, the left and right wrists and the torso at the waist. The total task dimension with the feet descriptors is $m = 21$, providing five degrees of redundancy when there is no constraint violation (assuming the augmented task Jacobian \mathbf{J} is full-rank). $N_{inact} = 3$ in this setup.

(iii) **Task Set 3: Tracking and constraint handling**

The specified task involves the feet, right wrist, and the torso at the waist only. The task dimension for this setup is $m = 18$. In this setup the left arm joints are not involved; i.e., $N_{inact} = 7$. This reduced-scaled experiment was developed to highlight the effects of constraints on task tracking.

Collision information was obtained using a standard collision detection software package.³¹ A total of 16 collision pairs were implemented in all experiments ($N_c = 16$) that included all upper-body links. Joint limits were set to provide appropriate ranges of motions within the kinematic capabilities of the humanoid robot.

In all experiments, the values for the feedback gain parameters were: $\mathbf{K}_p = 300\mathbf{I}$, $\mathbf{K}_v = 2\sqrt{\mathbf{K}_p}$, $\mathbf{G}_v = 30\mathbf{I}$. For all collision pairs, the collision avoidance function (Eq. 25) used: $\rho_j = 1$, $\beta_j = 2$, and $\alpha_j = 50$. The selected damping factor was $\lambda = 0.1$.

5.1. Pre-recorded Motion Retargeting Experiments

In this section, the desired task motions were derived from motions in the Carnegie Mellon University (CMU) human motion capture database.³⁶ We considered four highly dynamic and complex motions that produced multiple, simultaneous violations of joint limits and self-collisions when the motion, in its raw form, was

retargeted to the robot model. The motions considered were a boxing sequence, chicken-dance sequence, exercise sequence, and reaching sequence. Snapshots of the raw human reaching motion, displayed on an avatar model,³⁶ are illustrated in the top row of Fig. 2. The set of motions considered was diverse, with other motions, such as the chicken dance, including significant torso excursion in comparison to the reaching motion. Further details on these motions, including a movie of each captured motion, can be found in the CMU motion database.³⁶

Without considering joint limit and self-collision constraints, the motion was retargeted using Task Set 1. The results are shown in the bottom row of Fig. 2. The yellow highlighted regions indicate the occurrence of self-collisions. The average number of joint limit and self-collision constraint violations before constraint filtering is shown in Fig. 3. As can be observed, in general, for each motion, the number of violations in Task Set 2 is higher than the number in Task Set 1 because of the existence of more movement freedom (redundancy) with Task Set 2. An important challenge in all scenarios above is how to deal with situations where there are insufficient DoFs to execute the operational tasks and to handle the constraints simultaneously. As shown in Fig. 3, the average number of constraint violations for the exercise motion is more than 9 for Task Set 2, exceeding the 5 degrees of redundancy which are available when task tracking alone is considered.

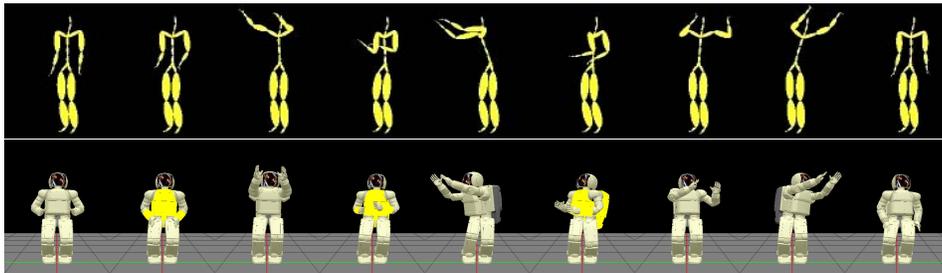


Fig. 2. Snapshots from the CMU retargeted reaching motion before constraint filtering (top - raw data, bottom - retargeted data). The yellow highlighted region in the bottom snapshots indicates occurrence of collisions before constraint filtering.

5.1.1. Enforcing Kinematic Constraints

Figure 4 shows a histogram of the number of constraint violations when the joint limit and self-collision constraints are not enforced in retargeting the reaching motion with Task Set 2. Each constraint violation translates to a reduction of at least one degree of mobility^a. Suppose we wish to retarget this motion with Task Set 2

^aEach joint limit violation restricts one degree of freedom. As well, self-collision avoidance restricts the motion of joints contributing toward the collision, removing at least one degree of mobility.

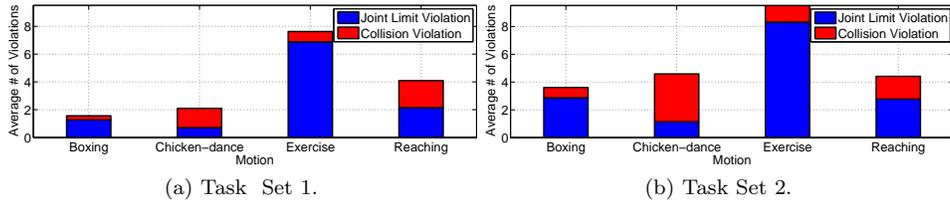


Fig. 3. Average number of constraint violations per frame for the selected CMU motions during Task Sets:(a) 1, and (b) 2, before constraint filtering. Videos of the human motions for these experiments can be found in the CMU motion database.³⁶

($m = 21$) while enforcing the constraints. In this case, there may be redundancy to satisfy both the constraints and task tracking only when the constraint violations number less than five (i.e., $n - N_{inact} - m = 29 - 3 - 21$). As shown in Fig. 4, there are periods in the latter half of the motion where the number of constraint violations exceeds five, indicating that the robot must sacrifice task tracking to satisfy the constraints. This simulation illustrates that even in the case where the operational tasks exhibit significant amount of redundancy (when considered alone), if the motion is not planned in advance, it is not possible to determine the actual redundancy when constraints are considered as well.

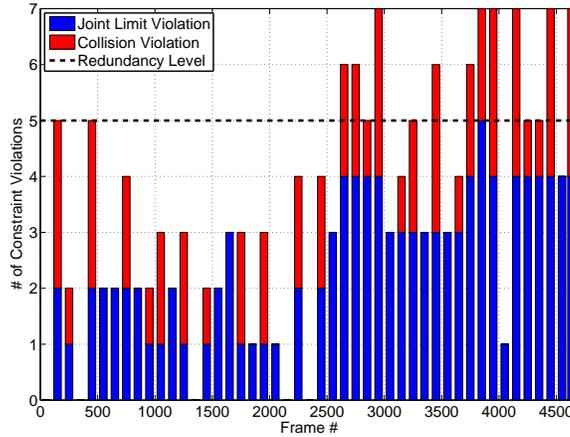


Fig. 4. Histogram of constraint violations for the CMU reaching motion before constraint filtering using the Task Set 2 setup. In Task Set 2, redundancy is established only if the total number of constraint violations is less than five.

To illustrate the effectiveness of the proposed approach in satisfying constraints in the absence of redundancy, consider the simulation results in Fig. 5. In particular, the plots show self-collision distance results for the scenario described in Task Set 1 for a selected period from the reaching motion. The figure shows the mini-

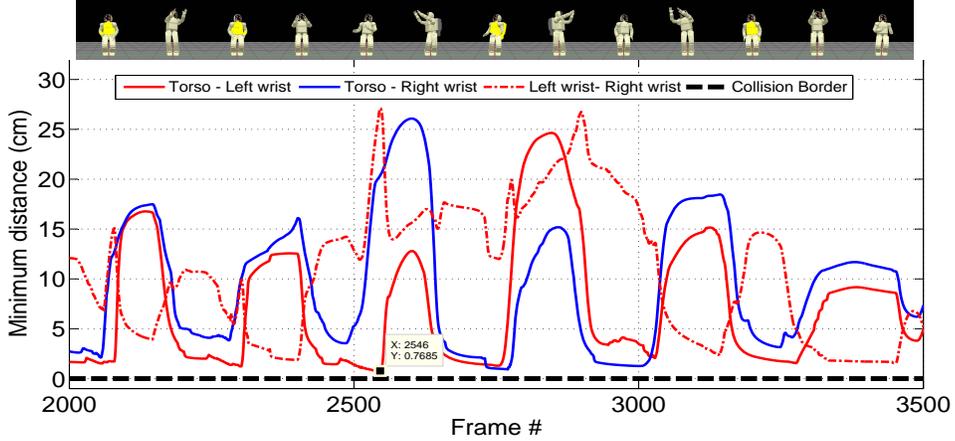


Fig. 5. The minimum distances between three segment pairs after the reaching motion is dynamically filtered to avoid collisions. The snapshots (top) are approximately aligned with the time-line (frame-number). The yellow highlighted region indicates occurrence of collisions before constraint filtering.

minimum distances between three segment pairs following the use of CRAC to avoid self-collision. The simulated robot snapshots are approximately aligned with the time-line (frame-number). The yellow highlighted regions in the snapshots indicate occurrence of self-collisions before filtering. As observed in the snapshots, the large separation between wrists and torso happens when the arms are extended away from the body to reach out and there is no self-collision. When the arms are moved closer to the torso, the distances become smaller, and CRAC successfully avoids multiple self-collisions as indicated by positive distance values across all frames.

Fig. 6 shows the plots of the joint angles of the right arm while performing the reaching motion after CRAC with the same task set (Task Set 1). As observed, all right arm joint angles, as an example, remain within the joint limits throughout the motion. It is also observed that the joints are allowed to move to and from their maximum limits as shown for the shoulder2 joint of the right arm, which controls the flexion and extension movement (along with the elbow) in the sagittal plane. This joint is almost fully extended and reaches as close as about 0.0035 rad (0.2 deg) to the joint upper limit (q_{max}), as observed around frame numbers 1250, 3000, and 4000. This happens mainly when the robot reaches far out as shown in Fig. 2.

Our method is capable of tracking the desired tasks adequately while satisfying all constraints, even in cases where the task dimension alone exceeds the system DoFs. This is highlighted in Figs. 7a and 7b. As it can be seen, the right wrist tracks the desired trajectory closely and maintains its features in both experiments. As expected, the tracking is generally better in the case of Task Set 2 due to the existence of the redundancy to achieve the task. On the other hand, in some instances

the tracking performance is degraded, such as around time 10s, 25s, 27s, and 30s. This degraded tracking occurs because the task is unreachable given the humanoid’s kinematic limitations. In order to confirm this, we repeated this experiment with Task Set 3 where the system has additional redundancy. As expected, some deviations are still observed as seen in the bottom plot of Fig. 8 around time 23s, 25s, 27s, 30s, 32s, 35s, 37s, 39s, 42s, and 44s. These deviations occur to avoid self-collisions as seen in the top plots of Fig. 8. These mainly occur when the right wrist is moving toward the torso. Note that these unavoidable constraints are possible due to the geometric and kinematics differences between the human and humanoid. Another interesting observation is that the tracking is not affected and stays excellent outside these self-collision periods. Therefore, the proposed method blends smoothly task tracking and constraint avoidance while guaranteeing constraint avoidance. Similar results were observed for the case of unavoidable joint limit violations.

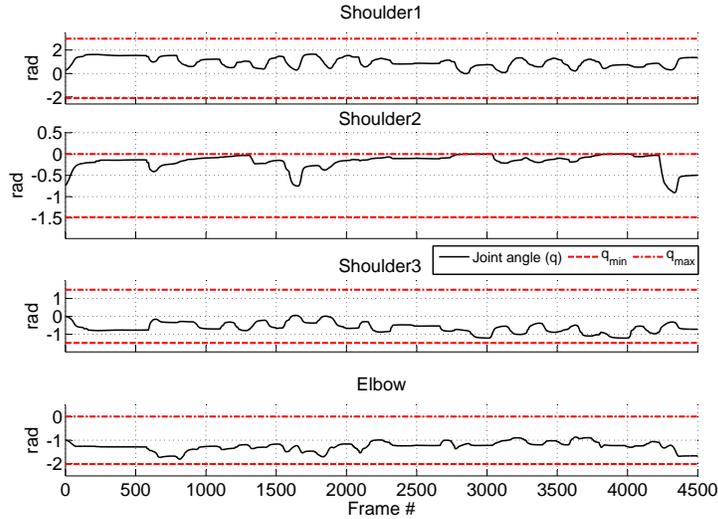


Fig. 6. Plots of right arm joint angles during the reaching motion with Task Set 1. Joint limits are shown by the dashed lines.

5.1.2. *Dynamic Consistency*

We also simulated the effect of using a dynamically-consistent solution on the accuracy and energetic cost of the resulting robot motion. In particular, we assumed a reference trajectory that included the Cartesian motions of the hands (wrists) and torso as described by Task Set 2. This gives rise to five degrees of redundancy to achieve the task alone, and provides opportunity to realize a solution which minimizes the instantaneous kinetic energy. The kinematic and dynamically-consistent solution was obtained with $\mathbf{W}_D = \mathbf{H}$. For comparison, we also considered the case

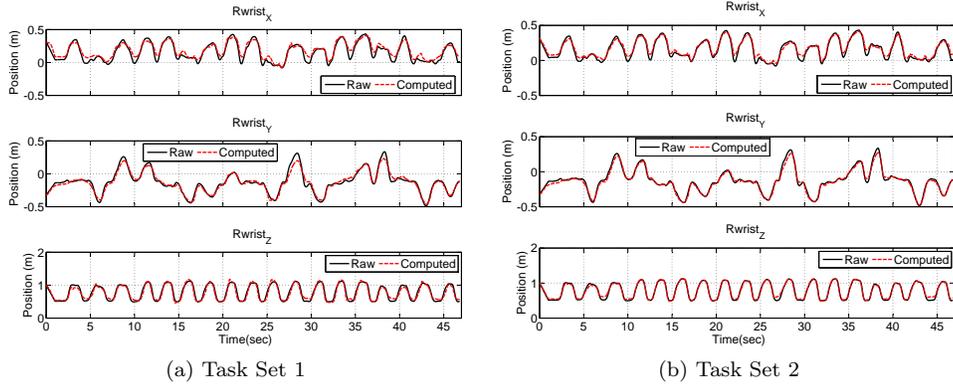


Fig. 7. Plots of right wrist task tracking during the reaching motion for Task Sets 1 and 2.

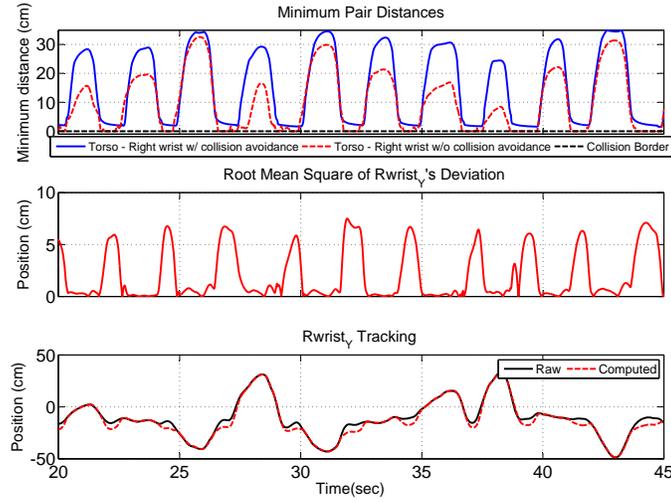


Fig. 8. Figures showing the tracking performance during self-collision avoidance for a period of the reaching motion in the Task Set 3 setup. The top plots show the minimum pair distances between the torso and right wrist with and without enforcing collision avoidance. The middle plot shows the root mean square of the lateral component of the right wrist task descriptor. The bottom plots show the tracking of the y position. Joint limit avoidance is not enforced in these plots.

when dynamic consistency was not applied with $\mathbf{W}_D = \mathbf{I}$. For the duration of the motion, the average absolute power required to actuate all joints was calculated as,

$$P_{tot} = \frac{1}{N_s} \sum_{j=1}^{N_s} |\boldsymbol{\tau}_j^T \dot{\mathbf{q}}_j| \quad (43)$$

where N_s is the number of motion samples (total frames). The comparison P_{tot} between the kinematically and kinodynamically-consistent solutions is depicted in

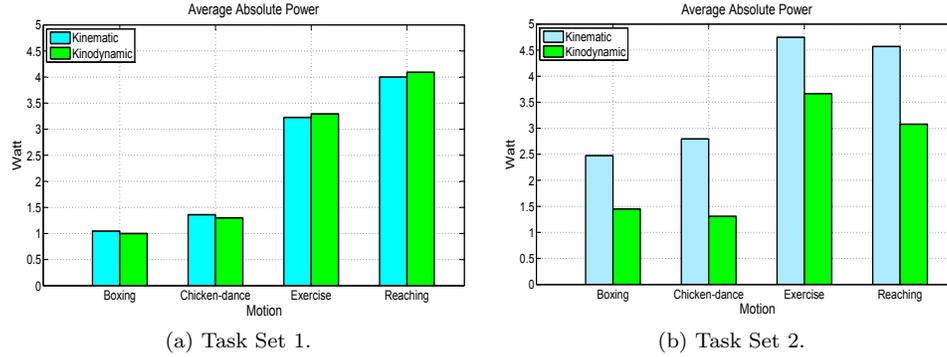


Fig. 9. The average absolute power to perform the pre-recorded motions for Task Sets 1 and 2. Videos of the human motions for these experiments can be found in the CMU motion database.³⁶

Fig. 9. Since only the hand and torso positions are provided as the reference trajectory, a solution which does not include the inertia weighting fails to penalize large torso movement, resulting in higher energetic cost. This is particularly true for highly dynamic movements such as reaching where kinematic consistency alone yields unnecessary torso movements. For comparison, we repeated the experiment with Task Set 1 setup, which has no redundancy. As expected the total power in the two solutions remains almost the same. Note that using the kinodynamic solution does not compromise the tracking accuracy. Experiment results show that the Root Mean Squared Deviation (RMSD) of the task remains nearly the same between the two solutions for all motion sequences in all experiment setups.

5.2. Online Motion Retargeting Experiments

In this section, we describe online whole-body experiments involving real-time human motion capture and retargeting. A motion capture system was developed based on *OpenNI* SDK skeleton tracking³⁷ in conjunction with a Microsoft *Kinect* unit 3D sensor.³⁸ The captured motion is first scaled from *OpenNI*'s skeleton to our humanoid robot model. Subsequently, all task descriptors are extracted and interpolated to 100 frames per second (fps). The proposed retargeting method is used to filter the captured motion and then retarget it to the humanoid model online (live). This software was tested live successfully with various motions on an HP Intel Dual Core (2.8 GHz) PC with 3 Gbytes of RAM. We have selected a sample human volleyball motion which features blocking and spiking without jumping. Snapshots from the human volleyball motion are illustrated in the top row of Fig. 10, with the retargeted motion, using Task Set 1, in the bottom row of Fig. 10.

The proposed method, including collision detection and avoidance computation, is computationally fast and thus suited to online applications. In general, the performance always exceeds the real-time requirement for the online retargeting, which is 100 Hz for our setups. In fact, the method can process more than 400 fps in the

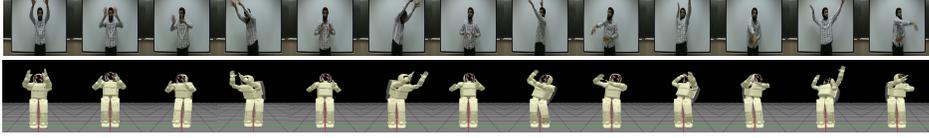


Fig. 10. Snapshots from live volleyball motion (top) and after retargeting with CRAC (bottom).

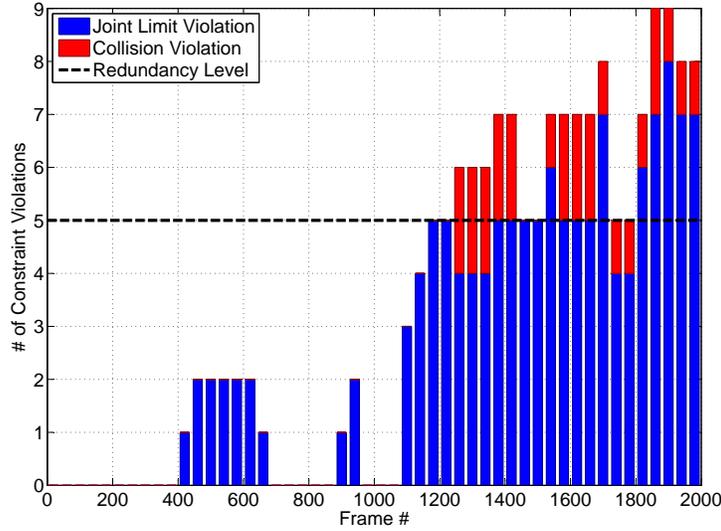


Fig. 11. Histogram of constraint violations for the online volleyball motion before constraint filtering using the Task Set 2 setup. In Task Set 2, redundancy is established only if the total number of constraint violations are less than five.

slowest scenario and about 3000 fps in the best case. The performance is affected by the collision detection configuration for SWIFT++ as well as the task dimensions.

5.2.1. Enforcing Kinematic Constraints

In Fig. 11, we show the corresponding histogram of the number of constraint violations when the joint limit and self-collision constraints are not enforced in retargeting the volleyball motion with Task Set 2. As shown in Fig. 11, the system does not have redundancy to satisfy both the constraints and task tracking in most of the instants after Frame 1200. As can be seen, joint limit violations occurred more frequently because the blocking and spiking movements require fully extended arms. In addition, self-collisions happened especially between the two wrists (hands) and torso during swinging motion.

As expected, all joint angles remain within the joint limits throughout the execution of the retargeted volleyball motion. Again the shoulder2 and elbow joints of

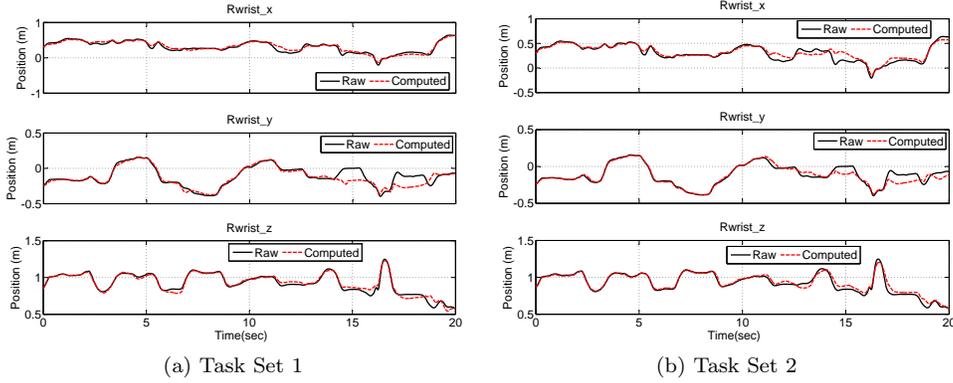


Fig. 12. Plots of right wrist tracking during the online volleyball motion with Task Set 1 and 2.

the right arm are fully extended and reach very close to the limits, such as when the robot reaches far out with its arms during the full extended block movement or at the end of the spiking movement, as shown in Fig. 10. Graphs of these joint angles provide similar conclusions to Fig. 6 and thus are omitted for space.

Despite the lack of redundancy to both satisfy the constraints and perfectly track the tasks, our online method tracks the tasks while satisfying all constraints. This is highlighted in Fig. 12a and Fig. 12b. As illustrated, the right wrist task is tracked closely and maintains its features in the first half of both figures. In addition, the tracking performance is better in the case of Task Set 2 due to the existence of additional redundancy. In general, the results demonstrate good tracking performance until time= 12s (frame 1200). This is expected since there are very few constraint violations in the raw motion in the first 1200 frames, as shown in Fig. 11. On the other hand, after frame 1200, the tracking deviation is more pronounced due to the higher number of constraint violations in the raw motion. In addition, some tasks in the latter part of the motion are unreachable due to other geometric constraints.

To see the the effect of constraints on tracking more clearly, we carried out CRAC with Task Set 3, in which the system has a yet additional redundancy. As expected, some deviations are still noticeable as seen in the bottom plot of Fig. 13 around times 16s, and 17s to 19 (highlighted by dashed boxes). These deviations occur when unavoidable self-collisions are present as seen in the top plots of Fig. 13. These mainly happen when the right wrist is moving toward the torso. Another interesting observation is that the tracking performance is excellent outside these constraint periods. That means, the proposed method smoothly blends the task tracking with constraint avoidance while enforcing constraint avoidance. Similar results were observed in case of unavoidable joint limit violations.

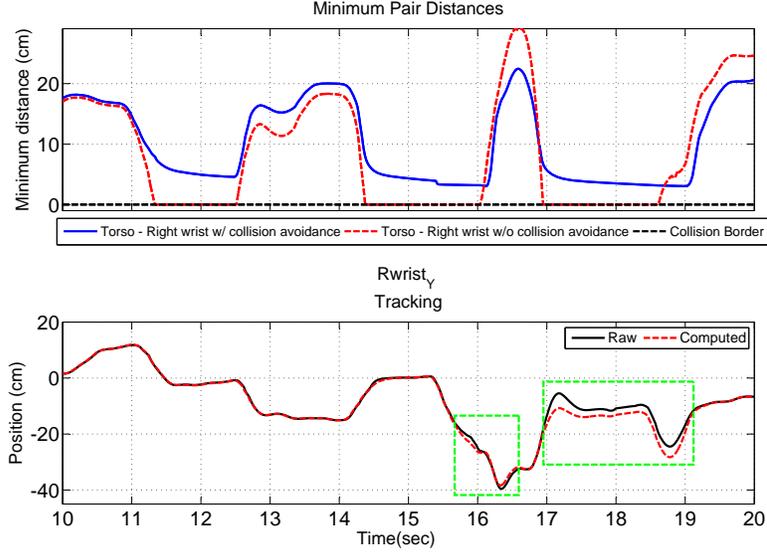


Fig. 13. Tracking performance versus the self-collision avoidance during a period of the online volleyball motion with Task Set 3. The top row shows the minimum pair distance between the torso and right wrist with and without enforcing the collision avoidance. The bottom row shows the tracking of the lateral component of the right wrist task. The dashed rectangles indicate areas where the task deviation is relatively large. Joint limit avoidance is not enforced in these plots.

5.2.2. Dynamic Consistency

We simulated the effect of using a dynamically-consistent solution in the volleyball online motion as well. The results were similar to those for the pre-recorded motions. We found that, in particular for the Task Set 2 setup, applying the kinodynamic solution in retargeting the volleyball motion reduces the total average absolute power by about 20% (from 4.7 Watt to 3.7 Watt), as compared to applying the kinematic solution only. On the other hand, as expected, there was no significant power reduction when the system is inherently over-constrained as in the Task Set 1 setup. Similarly to before, applying the kinodynamic solution does not reduce the tracking performance in either setup as results show that the RMSD of the tasks remains nearly the same between the two solutions.

5.3. Discussion

The capabilities of this framework encourage its future application in experimental humanoid systems. Although the method does require estimates of the dynamic parameters of the humanoid, the success of the joint damping approach applied here should ultimately depend more critically on the kinematic parameters of the humanoid. As constraint surfaces are approached, resolved accelerations α_c are computed to dampen joint motions. Our process to select joint accelerations α_c is

nearly an entirely kinematic process. Only the inertia weighting $\mathbf{W}_D = \mathbf{H}$ for the dynamically-consistent solution uses dynamic parameters, and our results show that constraints are still met when other weightings ($\mathbf{W}_D = \mathbf{I}$) are used. Thus, formation of $\boldsymbol{\alpha}_c$ to properly avoid constraints should only be sensitive to kinematic modeling errors. The proper realization of this resolved acceleration for a real humanoid is then in concept no more difficult than the challenges facing other dynamics-based whole-body controllers. An exact and comparative characterization of model robustness for different dynamics-based control approaches is an interesting and important future topic for humanoid control studies.

To enable future experimental validation, the framework has also been extended to include dynamic constraints on balance [39, Chap. 5]. While the full details of this extension are beyond the scope of the current work, the high-level idea of the extension is straightforward. If a solution of the whole-body CRAC controller presented here violates dynamic balance, the motion may be modified to bring it into balance using methods similar to previous work by Bin Hammam et al.⁴⁰ Although the accuracy of dynamic parameters will affect the performance of this and other dynamics-based control approaches, many recent results in experimental humanoid systems show a promising future for model-based humanoid control.

6. Summary and Conclusions

A generalized human-to-humanoid whole-body motion retargeting algorithm, using constrained resolved acceleration control (CRAC), has been proposed. The approach is suitable for execution of operational tasks in the presence of kinematic constraints (joint limits and self-collision) even when the robotic system does not exhibit redundancy to achieve the task. The algorithm introduced a control formula that incorporates constraint handling and task tracking in a unified framework and makes smooth and automatic transitions between regimes where different constraints are binding. The algorithm is real time and is suitable for online control of humanoid motion, where the possibility of a large number of kinematic constraint violations exists and the motion is not planned in advance. In addition, the proposed resolved acceleration approach recovers a dynamically-consistent Jacobian pseudo-inverse when away from all constraints in order to minimize costly motions of the more massive bodies in the system. The satisfaction of the constraints across a wide range of experiments has been demonstrated through online and offline application of CRAC to dynamic whole-body motions for human-to-humanoid motion retargeting. The satisfaction of important joint limit and self-collision constraints across these examples encourages future work to evaluate the proposed framework on an experimental humanoid.

7. Acknowledgments

The authors would like to thank Dr. Youding Zhu for his development and support of the motion capturing system. This work was supported by a research contract from

the Honda Research Institute USA to The Ohio State University. Other funding was provided by Grant No. CNS-0960061 from the National Science Foundation with a subaward to The Ohio State University. Part of the data used in this project was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217.

References

1. S. Tak, O. Song, and H. Ko, Motion balance filtering, *Comput. Graph. Forum. (Eurographics)* **19**(3) (2000) 437–446.
2. A. Nakazawa, S. Nakaoka, K. Ikeuchi, and K. Yokoi, Imitating human dance motions through motion structure analysis, in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (Lausanne, Switzerland, 2002), pp. 2539–2544.
3. S. Tak and H. Ko, A physically-based motion retargeting filter, *ACM Transactions on Graphics* **24**(1) (2005) 98–117.
4. N. Naksuk, C. G. Lee, and S. Rietdyk, Whole-body human-to-humanoid motion transfer, in *IEEE/RAS Int. Conf. on Hum. Robots* (Tsukuba, Japan, 2005), pp. 104–109.
5. B. Dariush, M. Gienger, A. Arumbakkam, Y. Zhu, B. Jian, K. Fujimura, and C. Goerick, Online transfer of human motion to humanoids, *International Journal of Humanoid Robotics* **6**(2) (2009) 265–289.
6. K. Yamane and J. Hodgins, Simultaneous tracking and balancing of humanoid robots for imitating human motion capture data, in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (St. Louis, MO, 2009), pp. 2510–2517.
7. K. Yamane and J. Hodgins, Control-aware mapping of human motion data with stepping for humanoid robots, in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (Taipei, 2010), pp. 726–733.
8. H. Park and C. Lee, Cooperative-dual-task-space-based whole-body motion balancing for humanoid robots, in *IEEE Int. Conf. on Robotics and Automation* (Karlsruhe, Germany, 2013), pp. 4797–4802.
9. B. Dariush, M. Gienger, A. Arumbakkam, C. Goerick, Y. Zhu, and K. Fujimura, Online and markerless motion retargeting with kinematic constraints, in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (Nice, France, 2008), pp. 191–198.
10. J. Luh, M. Walker, and R. Paul, Resolved-acceleration control of mechanical manipulators, *IEEE Transactions on Automatic Control* **25**(3) (1980) 468–474.
11. O. Khatib, A unified approach for motion and force control of robot manipulators: The operational space formulation, *IEEE Journal of Robotics and Automation*. **3**(1) (1987) 43–53.
12. A. A. Maciejewski and C. A. Klein, Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments, *International Journal of Robotics Research* **4**(3) (1985) 109–117.
13. O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, *International Journal of Robotics Research* **5**(1) (1986) 90–98.
14. H. Sugiura, M. Gienger, H. Janssen, and C. Goerick, Real-time collision avoidance with whole body motion control for humanoid robots, in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (San Diego, CA, 2007), pp. 2053–2058.
15. O. Stasse, A. Escande, N. Mansard, S. Miossec, P. Evrard, and A. Kheddar, Real-time (self)-collision avoidance task on a HRP-2 humanoid robot, in *IEEE Int. Conf. on Robotics and Automation* (Pasadena CA, 2008), pp. 3200–3205.
16. Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*, (Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990), 1st Edition.

17. P. Hsu, J. Hauser, and S. Sastry, Dynamic control of redundant manipulators, *Journal of Robotic Systems* **6**(2) (1989) 133–148.
18. L. Sentis and O. Khatib, A whole-body control framework for humanoids operating in human environments, in *IEEE Int. Conf. on Robotics and Automation* (Orlando, FL, 2006), pp. 2641–2648.
19. P. Wensing and D. Orin, Generation of dynamic humanoid behaviors through task-space control with conic optimization, in *IEEE Int. Conf. on Robotics and Automation* (Karlsruhe, Germany, 2013), pp. 3103–3109.
20. A. Macchietto, V. Zordan, and C. R. Shelton, Momentum control for balance, *ACM Trans. Graph.* **28**(3) (2009) 80:1–80:8.
21. Y. Abe, M. da Silva, and J. Popović, Multiobjective control with frictional contacts, in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2007), pp. 249–258.
22. L. Saab, O. Ramos, F. Keith, N. Mansard, P. Soueres, and J. Fourquet, Dynamic whole-body motion generation under rigid contacts and other unilateral constraints, *IEEE Transactions on Robotics* **29**(2) (2013) 346–362.
23. A. El Khoury, F. Lamiroux, and M. Taïx, Optimal motion planning for humanoid robots, in *IEEE Int. Conf. on Robotics and Automation* (Karlsruhe, Germany, 2013), pp. 3136–3141.
24. M. Nikolić, B. Borovac, M. Raković, and S. Savić, A further generalization of task-oriented control through tasks prioritization, *International Journal of Humanoid Robotics* **10**(03) (2013) 1350012.
25. B. Dariush, G. Bin Hammam, and D. Orin, Constrained resolved acceleration control for humanoids, in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (Taipei, 2010), pp. 710–717.
26. Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, The intelligent ASIMO: system overview and integration, in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (Lausanne, Switzerland, 2002), pp. 2478–2483.
27. J. Nakanishi, R. Cory, M. Mistry, and J. Peters, Operational space control: A theoretical and empirical comparison, *International Journal of Robotics Research* **27**(6) (2008) 737–757.
28. F. Chiaverini, B. Siciliano, and O. Egeland, Review of damped least-squares inverse kinematics with experiments on an industrial robot manipulator, *IEEE Transactions on Control Systems Tech.* **2**(2) (1994) 123–134.
29. B. Dariush, Y. Zhu, A. Arumbakkam, and K. Fujimura, Constrained closed loop inverse kinematics, in *IEEE Int. Conf. on Robotics and Automation* (Anchorage, AK, 2010), pp. 2499–2506.
30. 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) 286–292.
31. UNC Chapel Hill: Swift++ Library, Speedy Walking via Improved Feature Testing for Non-Convex Objects, Internet page, (2001), <http://gamma.cs.unc.edu/SWIFT++/>.
32. O. Khatib, Reduced effective inertia in macro-/mini-manipulator systems, in *The Fifth Int. Symposium on Robotics Research* (Cambridge, MA, USA, 1990), pp. 279–284.
33. O. Khatib, Inertial properties in robotic manipulation: An object-level framework, *International Journal of Robotics Research* **14**(1) (1995) 19–36.
34. H. Bruyninckx and O. Khatib, Gauss’ principle and the dynamics of redundant and constrained manipulators, in *IEEE Int. Conf. on Rob. and Automation* (2000), pp. 2563–2568.
35. R. Featherstone and D. Orin, Chapter 2: Dynamics, in *Springer Handbook of Robotics*,

- B. Siciliano and O. Khatib (eds.), (Springer, New York, 2008).
36. Carnegie Mellon University, CMU graphics lab motion capture database, Internet page, <http://mocap.cs.cmu.edu/>.
 37. PrimeSense, OpenNI SDK library, Internet page, www.openni.org/openni-sdk/.
 38. PrimeSense, Primesense's technology for Microsoft Kinect, Internet page, www.primesense.com/casestudies/kinect/.
 39. G. Bin Hammam, *Whole-Body Motion Retargeting for Humanoids*, PhD thesis, The Ohio State University, Columbus OH, (2014).
 40. G. Bin Hammam, D. Orin, and B. Dariush, Whole-body humanoid control from upper-body task specifications, in *IEEE Int. Conf. on Robotics and Automation* (Anchorage, Alaska, USA, 2010), pp. 3398–3405.