Comparative Experiments on Task Space Control with Redundancy Resolution

Jun Nakanishi*†, Rick Cory*†, Michael Mistry‡, Jan Peters‡, and Stefan Schaal†‡
* *ICORP, Japan Science and Technology Agency, Kyoto, Japan*
† *ATR Computational Neuroscience Laboratories, Kyoto, Japan*
‡ *Computer Science and Neuroscience, University of Southern California, Los Angeles, USA*
{*jun, cory*}*@atr.jp*, {*mmistry, jrpeters, sschaal*}*@usc.edu*

***Abstract*— Understanding the principles of motor coordination with redundant degrees of freedom still remains a challenging problem, particularly for new research in highly redundant robots like humanoids. Even after more than a decade of research, task space control with redundancy resolution still remains an incompletely understood theoretical topic, and also lacks a larger body of thorough experimental investigation on complex robotic systems. This paper presents our first steps towards the development of a working redundancy resolution algorithm which is robust against modeling errors and unforeseen disturbances arising from contact forces.**

To gain a better understanding of the pros and cons of different approaches to redundancy resolution, we focus on a comparative empirical evaluation. First, we review several redundancy resolution schemes at the velocity, acceleration and torque levels presented in the literature in a common notational framework and also introduce some new variants of these previous approaches. Second, we present experimental comparisons of these approaches on a seven-degree-of-freedom anthropomorphic robot arm. Surprisingly, one of our simplest algorithms empirically demonstrates the best performance, despite, from a theoretical point, the algorithm does not share the same beauty as some of the other methods. Finally, we discuss practical properties of these control algorithms, particularly in light of inevitable modeling errors of the robot dynamics.

***Index Terms*— Redundancy resolution, Inverse kinematics, Task space control, Null space optimization, Dynamical decoupling**

I. Introduction

Understanding the principles of natural movement generation has previously been and continues to be one of the most interesting and important open problems in the fields of robotics and neural control of movement. One of the major difficulties that arises is resolving the redundant degrees of freedom (DOFs) of a task in a general, but also task supporting way. While originally explored in robotic systems with only few redundant DOFs, the problem of redundancy resolution has recently become increasingly prominent again within the context of highly complex robotic systems such as humanoid robots. These machines have a very large number of redundant DOFs and emphasize redundancy resolution schemes that are robust under strong disturbances, real-time control in dynamically changing environments, a hierarchy of multiple task constraints, and esthetically appealing movement for working in a human environment.

The literature opn kinematic redundancy resolution offers a wide variety of techniques. A typical approach makes use of the Jacobian pseudoinverse with local null-space optimization in order to determine the inverse kinematics transformation [1]. Within this general framework, kinematic redundancy can be resolved at the velocity [2], acceleration [3]–[6] and torque levels [7], [8], where the desired joint velocities, accelerations, and torques are computed, respectively, for a desired endeffector velocity, acceleration, and force. It should be noted, however, that many of the previous investigations of control with inverse kinematics were confined to pure simulation studies, and also only investigated rather simple movements, like pointing, reaching, or circular drawing movements. Moreover, stability properties of most algorithms are only incompletely understood. A noteworthy exception is a recent study by Arimoto [9], which proposes a provably stable redundancy resolution algorithm for reaching movements based on the Jacobian transpose with joint velocity damping. One of the advantages of this approach is that it does not suffer from singularity problems associated with matrix inversion of the Jacobian. However, it is difficult to apply this method to tasks other than reaching for a static target; for example, tasks which require control of position, velocity and/or acceleration of the task coordinates, as is the case when tracking a moving target or balancing a pole [10].

From our point of view, even after more than a decade of research, inverse kinematics control remains a surprisingly incompletely understood theoretical topic, and also lacks a larger body of thorough empirical investigation on complex robotic systems that do not lend themselves to highly accurate

This research was supported in part by National Science Foundation grants ECS-0325383, IIS-0312802, IIS-0082995, ECS-0326095, ANI-0224419, a NASA grant AC#98-516, an AFOSR grant on Intelligent Control, Communications Research Laboratory of Japan, the ERATO Kawato Dynamic Brain Project funded by the Japan Science and Technology Agency, and the ATR Computational Neuroscience Laboratories.

dynamics and kinematics models, as is typically the case for complex humanoids. It is particularly the latter domain that we would like to address in our research. Our ultimate goals in this domain are to i) quantify the principles (or optimization criteria) used in human-like redundancy resolution, ii) suggest computational models and robust algorithms that can deal with expected and unexpected contact forces, iii) investigate human behavior based on theory driven experimental design, and iv) address learning algorithms for redundancy resolution. In our previous studies, we have addressed computational efficiency in redundancy resolution algorithms and inverse kinematics learning based on velocity based approaches [11], [12].

From a more algorithmic point of view, we are interested in developing a robust inverse dynamics controller with redundancy resolution that satisfies the following requirements:

1) avoids high-gain control or integrators in order to robustly deal with unforeseen impacts in a dynamically changing environment
2) has the ability to incorporate force and kinematic constraints
3) can work with any combination of desired acceleration, velocity, and position specification
4) resolves redundancy in a task supporting way
5) is insensitive to sensory noise or slow sampling rates of task space sensory feedback (e.g. 60Hz vision)
6) is robust in face of kinematic and dynamic modeling errors

This paper presents our first steps towards addressing these issues by focusing on a comparative evaluation of various previously suggested redundancy resolution theories, including comprehensive experimental comparisons on an anthropomorphic robot. We first review several theories presented in the literature in a common notational framework and also introduce some new variants of the previous approaches. We then experimentally test these approaches on the seven-degree-of-freedom anthropomorphic robot Sarcos Master Arm and discuss their practical properties, particularly in light of inevitable modeling errors of the robot dynamics.

Our results provide useful insights into the development of a robust algorithm that fulfills the constraints listed above and offer several simplifications and new variants of previously suggested algorithms in the literature. Moreover, we believe that our results will be helpful to outline a possible program to investigate redundancy resolution in behavioral research for computational neuroscience.

II. Problem Setup

The rigid body dynamics equations of a robot are given in the form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \tag{1}$$

where $\mathbf{q} \in \Re^n$ is the joint angle vector, $\mathbf{M}(\mathbf{q})$ is the inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is the Coriolis/centrifugal vector, $\mathbf{g}(\mathbf{q})$ is the gravity vector, and $\boldsymbol{\tau}$ is the joint torque vector.

The forward kinematics and differential relationship between the joint coordinates and the task coordinates are given as

$$\mathbf{x} = \mathbf{f}(\mathbf{q}) \tag{2}$$

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \tag{3}$$

$$\ddot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} \tag{4}$$

where $\mathbf{J}(\mathbf{q})$ is the Jacobian matrix and $\mathbf{x} \in \Re^m$, where $m < n$, is the task coordinate vector.

The basic idea in redundancy resolution in the velocity and acceleration levels is to compute the desired joint velocities and accelerations respectively as

$$\dot{\mathbf{q}}_d = \mathbf{J}^\dagger \dot{\mathbf{x}}_d + (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})\boldsymbol{\xi}_1 \tag{5}$$

$$\ddot{\mathbf{q}}_d = \mathbf{J}^\dagger (\ddot{\mathbf{x}}_d - \dot{\mathbf{J}}\dot{\mathbf{q}}) + (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})\boldsymbol{\xi}_2 \tag{6}$$

where $\mathbf{J}^\dagger$ is the pseudoinverse defined by $\mathbf{J}^\dagger = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}$, $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ are arbitrary vectors, and $\dot{\mathbf{x}}_d$ and $\ddot{\mathbf{x}}_d$ are the given desired task space velocities and accelerations, respectively. $(\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})$ projects arbitrary vectors $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ onto the null space of the Jacobian $\mathbf{J}$. Note that by equating (6) and the analytical differentiation of (5), sufficient conditions for these two resolution schemes to be consistent are derived in [4] and [5], respectively, as

$$\boldsymbol{\xi}_2 = \dot{\mathbf{J}}^\dagger \mathbf{J}(\dot{\mathbf{q}} - \boldsymbol{\xi}_1) + \dot{\boldsymbol{\xi}}_1 \tag{7}$$

and

$$\boldsymbol{\xi}_2 = \dot{\mathbf{J}}^T (\mathbf{J}^\dagger)^T (\dot{\mathbf{q}} - \boldsymbol{\xi}_1) + \dot{\boldsymbol{\xi}}_1 \tag{8}$$

where $\dot{\mathbf{J}}^\dagger = \frac{d}{dt}(\mathbf{J}^\dagger)$ for notational convenience. Note that it can be shown that (7) and (8) are equivalent.

In the torque based redundancy resolution, the inertia-weighted pseudoinverse $\bar{\mathbf{J}} = \mathbf{M}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}$ is used to compute the desired joint torque command exploiting the dynamical decoupling property of the joint torques that create task space forces, and the joint torques that act only in the null space of the task [7], [8].

In the next section, we will discuss the specific formulations of various redundancy resolution schemes in more detail.

III. Controller Formulations

In this paper, we consider the task of tracking a moving target in task space with the desired positions, velocities and accelerations given by $\mathbf{x}_d, \dot{\mathbf{x}}_d$ and $\ddot{\mathbf{x}}_d$, respectively. For redundancy resolution, we employ a null-space posture optimization with the cost function

$$g(\mathbf{q}) = \frac{1}{2}(\mathbf{q} - \mathbf{q}_{rest})^T \mathbf{K}_w (\mathbf{q} - \mathbf{q}_{rest}) \tag{9}$$

where $\mathbf{K}_w$ is a weighting positive definite diagonal matrix and $\mathbf{q}_{rest}$ is some rest (preferred) posture.

Using this formulation, we will review several of the most prominent algorithms for redundancy resolution, focusing on algorithms at the velocity, acceleration and torque levels, and also in a unified notational framework that will illuminate similarities and differences.

A. Velocity based Control

Velocity based redundancy resolution computes the desired joint velocities for a given task space desired velocity. Among the most popular is the resolved motion rate control approach based on Liegeois' pseudoinverse with null-space optimization [1].

1) Velocity based Control with Joint Velocity Integration: This section introduces a velocity based redundancy resolution scheme with joint velocity integration used in our traditional implementaion of endpoint control (e.g., pole-balancing [10]).

Given the velocities and positions of the target in task space, we define the task space velocity command $\dot{\mathbf{x}}_r$ from the addtion of a position error term with a target velocity feedforward term as

$$\dot{\mathbf{x}}_r = \dot{\mathbf{x}}_d + \mathbf{K}_p(\mathbf{x}_d - \mathbf{x}) \tag{10}$$

where $\mathbf{K}_p$ is a positive definite gain matrix. Note that this scheme does not use the target acceleration information $\ddot{\mathbf{x}}_d$.

The reference joint velocities are calculated based on a pseudoinverse solution with minimization of the cost function (9) in the null space

$$\dot{\mathbf{q}}_r = \mathbf{J}^\dagger \dot{\mathbf{x}}_r - (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})\alpha \nabla g. \tag{11}$$

In (11), the null-space vector is chosen to be the negative gradient of the cost function as $\boldsymbol{\xi}_1 = -\alpha \nabla g = -\alpha \left(\frac{\partial g}{\partial \mathbf{q}}\right)^T$ where α is a positive constant. The reference joint accelerations and positions are obtained by numerical differentiation and integration of the reference joint velocities (11), respectively as

$$\ddot{\mathbf{q}}_r = \frac{d}{dt}\dot{\mathbf{q}}_r \simeq \frac{\dot{\mathbf{q}}_r(t) - \dot{\mathbf{q}}_r(t - \Delta t)}{\Delta t} \tag{12}$$

$$\mathbf{q}_r = \int_{t_0}^{t} \dot{\mathbf{q}}_r \, dt' \simeq \mathbf{q}_r(t - \Delta t) + \dot{\mathbf{q}}_r(t)\Delta t \tag{13}$$

where Δt is the sampling period. The final control signal is calculated using the computed torque control method with a PD controller as

$$\begin{aligned}\boldsymbol{\tau} = {} & \mathbf{M}(\mathbf{q}_r)\ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q}_r, \dot{\mathbf{q}}_r) + \mathbf{g}(\mathbf{q}_r) \\ & + \mathbf{K}_{q,d}(\dot{\mathbf{q}}_r - \dot{\mathbf{q}}) + \mathbf{K}_{q,p}(\mathbf{q}_r - \mathbf{q})\end{aligned} \tag{14}$$

where $\mathbf{K}_{q,d}$ and $\mathbf{K}_{q,p}$ are positive definite gain matrices.

This method has two disadvantages. One is that we do not use the information of the target accelerations. The other is the requirement of numerical differentiation and integration of the desired joint velocities to obtain the desired joint accelerations and joint positions. Numerical differentiation is very sensitive to sensor noise, and integration has the problem of windup if the robot motion is constrained by an external disturbance for a long period, which ultimately accumulates a large error giving rise to unrealizable joint torques.

2) Velocity based Control without Joint Velocity Integration: When considering contact forces with the environment, it is desirable to avoid integrators. In this section, we introduce a velocity based controller without joint velocity integration. Suppose we have the reference task space velocity command and the reference joint velocities obtained from the Liegeois' null-space optimization,

$$\dot{\mathbf{x}}_r = \dot{\mathbf{x}}_d + \mathbf{K}_p(\mathbf{x}_d - \mathbf{x}) \tag{15}$$

$$\dot{\mathbf{q}}_r = \mathbf{J}^\dagger \dot{\mathbf{x}}_r - \alpha(\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})\nabla g. \tag{16}$$

An inverse dynamics control law can be formulated with the addition of a joint velocity feedback term as

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \mathbf{K}_{q,d}(\dot{\mathbf{q}}_r - \dot{\mathbf{q}}) \tag{17}$$

where the reference joint accelerations $\ddot{\mathbf{q}}_r$ are calculated through numerical differentiation as in (13). The control law (17) can be rearranged as

$$\begin{aligned}\boldsymbol{\tau} &= \mathbf{M}\ddot{\mathbf{q}}_r + \mathbf{C} + \mathbf{g} + \mathbf{K}_{q,d}(\dot{\mathbf{q}}_r - \dot{\mathbf{q}}) && (18)\\ &= \mathbf{M}(\ddot{\mathbf{q}}_d + \mathbf{J}^\dagger \mathbf{K}_p \dot{\mathbf{e}}) + \mathbf{C} + \mathbf{g} + \mathbf{K}_{q,d}(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) \\ &\quad + \mathbf{K}_{q,d}\mathbf{J}^\dagger \mathbf{K}_p \mathbf{e} && (19)\end{aligned}$$

using the relationship (5) with $\boldsymbol{\xi}_1 = -\alpha \nabla g$, where $\mathbf{e} = \mathbf{x}_d - \mathbf{x}$. As in the previous velocity based controller, this method has a disadvantage in that it ignores the information of the target accelerations. The second disadvantage is that the behavior of the task space dynamics cannot be easily specified as compared to that of acceleration and torque based schemes which we discuss below.

B. Acceleration based Control

1) Hsu's Controller [3]: This section presents an acceleration based redundancy resolution scheme proposed in [3]. In this method, the control law is given by

$$\boldsymbol{\tau} = \mathbf{M}\left(\mathbf{J}^\dagger(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + \boldsymbol{\phi}_N\right) + \mathbf{C} + \mathbf{g} \tag{20}$$

where

$$\ddot{\mathbf{x}}_r = \ddot{\mathbf{x}}_d + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} \tag{21}$$

$$\boldsymbol{\phi}_N = (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})(\dot{\mathbf{h}} + \mathbf{K}_N\mathbf{e}_N) - (\mathbf{J}^\dagger\dot{\mathbf{J}}\mathbf{J}^\dagger + \dot{\mathbf{J}}^\dagger)\mathbf{J}(\mathbf{h} - \dot{\mathbf{q}}) \tag{22}$$

$$\mathbf{e}_N = (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})(\mathbf{h} - \dot{\mathbf{q}}) \tag{23}$$

$\mathbf{h}$ is a vector function, $\mathbf{K}_N$ is a positive definite matrix, $\mathbf{e} = \mathbf{x}_d - \mathbf{x}$, and $\mathbf{K}_p$ and $\mathbf{K}_p$ are positive definite gain matrices. It can be shown that the control law (20) with (21) yields the task space tracking error dynamics

$$\ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = 0. \tag{24}$$

This implies that this controller achieves asymptotic tracking in the task space, i.e., $\mathbf{e} \to 0$ as $t \to \infty$ assuming that $\mathbf{J}^\dagger$ is full rank [3].

A slightly unclear issue in this formulation is how the complex null-space vector $\boldsymbol{\phi}_N$ in (22) is derived in [3]. As a result of our analysis, the null-space vector (22) can be rearranged as

$$\begin{aligned}\boldsymbol{\phi}_N &= (\mathbf{I}-\mathbf{J}^\dagger\mathbf{J})[\dot{\mathbf{h}} + \mathbf{K}_N(\mathbf{I}-\mathbf{J}^\dagger\mathbf{J})(\mathbf{h} - \dot{\mathbf{q}})] \\ &\quad - (\mathbf{J}^\dagger\dot{\mathbf{J}}\mathbf{J}^\dagger + \dot{\mathbf{J}}^\dagger)\mathbf{J}(\mathbf{h} - \dot{\mathbf{q}}) \\ &= (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})[\dot{\mathbf{h}} + \mathbf{K}_N(\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})(\mathbf{h} - \dot{\mathbf{q}})] \\ &\quad - (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\dot{\mathbf{J}}^\dagger\mathbf{J}(\mathbf{h} - \dot{\mathbf{q}}) \\ &= (\mathbf{I}-\mathbf{J}^\dagger\mathbf{J})[\dot{\mathbf{J}}^\dagger\mathbf{J}(\dot{\mathbf{q}}-\mathbf{h}) + \dot{\mathbf{h}} - \mathbf{K}_N(\mathbf{I}-\mathbf{J}^\dagger\mathbf{J})(\dot{\mathbf{q}}-\mathbf{h})]\end{aligned} \tag{25}$$

Thus, the controller in [3] is equivalent to the case where

$$\boldsymbol{\tau} = \mathbf{M}\ddot{\mathbf{q}}_r + \mathbf{C} + \mathbf{g} \tag{26}$$

$$\ddot{\mathbf{q}}_r = \mathbf{J}^\dagger(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + \boldsymbol{\phi}_N \tag{27}$$

$$= \mathbf{J}^\dagger(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\boldsymbol{\xi}_2 \tag{28}$$

$$\boldsymbol{\xi}_2 = \underbrace{\dot{\mathbf{J}}^\dagger\mathbf{J}(\dot{\mathbf{q}} - \boldsymbol{\xi}_1) + \dot{\boldsymbol{\xi}}_1}_{\text{same as (7)}} - \mathbf{K}_N(\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})(\dot{\mathbf{q}} - \boldsymbol{\xi}_1) \tag{29}$$

with $\mathbf{h} = \boldsymbol{\xi}_1$ in (25). Notice that from (7) and (25), we can consider that the acceleration null space vector (29) is derived from analytical differentiation of the velocity based resolution having an additional term. In the subsequent robot implementation, we choose $\boldsymbol{\xi}_1 = -\alpha\nabla g$ such that

$$\boldsymbol{\xi}_2 = \dot{\mathbf{J}}^\dagger\mathbf{J}(\dot{\mathbf{q}} + \alpha\nabla g) + (\dot{\nabla} g) - \mathbf{K}_N(\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})(\dot{\mathbf{q}} + \alpha\nabla g). \tag{30}$$

Note that in this controller the inertia matrix premultiplies the null space term as

$$\boldsymbol{\tau} = \mathbf{M}\mathbf{J}^\dagger(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + \mathbf{C} + \mathbf{g} + \mathbf{M}\boldsymbol{\phi}_N. \tag{31}$$

2) Simplified Acceleration based Control Variation 1 (with nullspace pre-multiplication of $\mathbf{M}$*):* As we have seen above, the acceleration based method proposed in [3] has a rather complicated null space vector (see (30)). This section introduces an acceleration based controller with a simplified null space vector.

Consider a control law

$$\boldsymbol{\tau} = \mathbf{M}\ddot{\mathbf{q}}_r + \mathbf{C} + \mathbf{g} \tag{32}$$

where

$$\ddot{\mathbf{q}}_r = \mathbf{J}^\dagger(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\boldsymbol{\xi}_2 \tag{33}$$

$$\boldsymbol{\xi}_2 = -\mathbf{K}_{q,d}\dot{\mathbf{q}} - \alpha\nabla g \tag{34}$$

and $\ddot{\mathbf{x}}_r = \ddot{\mathbf{x}}_d + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}$.

In this controller, we simplify the null space vector $\boldsymbol{\xi}_2$ and introduce a Liegeois-like null space projection with a damping term in the joint acceleration space. This controller achieves asymptotic tracking in task space since we have

$$\ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = 0. \tag{35}$$

Note that in this controller the inertia matrix effectively premultiplies the null space term as

$$\boldsymbol{\tau} = \mathbf{M}\mathbf{J}^\dagger(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + \mathbf{C} + \mathbf{g} + \mathbf{M}(\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\boldsymbol{\xi}_2. \tag{36}$$

3) Simplified Acceleration based Control Variation 2 (without nullspace pre-multiplication of $\mathbf{M}$*):* In the controller above, a null space optimization term is introduced in the acceleration space as in (33). As a variant of this controller, we introduce a control law in which a PD term using a Liegeois-like null space projection is directly introduced in the torque command:

$$\boldsymbol{\tau} = \mathbf{M}\ddot{\mathbf{q}}_r + \mathbf{C} + \mathbf{g} + (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\boldsymbol{\xi}_2 \tag{37}$$

$$= \mathbf{M}\mathbf{J}^\dagger(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + \mathbf{C} + \mathbf{g} + (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})(-\mathbf{K}_{q,d}\dot{\mathbf{q}} - \alpha\nabla g) \tag{38}$$

where

$$\ddot{\mathbf{q}}_r = \mathbf{J}^\dagger(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) \tag{39}$$

$$\boldsymbol{\xi}_2 = -\mathbf{K}_{q,d}\dot{\mathbf{q}} - \alpha\nabla g \tag{40}$$

and $\ddot{\mathbf{x}}_r = \ddot{\mathbf{x}}_d + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}$.

With this control law, we have the following closed loop dynamics:

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{M}\mathbf{J}^\dagger(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})(-\mathbf{K}_{q,d}\dot{\mathbf{q}} - \alpha\nabla g) \tag{41}$$

By pre-multiplying $\mathbf{J}\mathbf{M}^{-1}$ to the both hand side, (41) can be rearranged as

$$\ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = \mathbf{J}\mathbf{M}^{-1}(\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})(\mathbf{K}_{q,d}\dot{\mathbf{q}} + \alpha\mathbf{K}_w(\mathbf{q}_{rest} - \mathbf{q})) \tag{42}$$

Note that as (42) indicates there is interference between the range and null space since the vector on the right hand side of (42) drives the task space tracking error dynamics. However, as our empirical results demonstrate in Section V, we achieved excellent tracking performance although the effect of interference is not clear because of the difficulty in the mathematical analysis of (42). Moreover, in practice, the effect of the interference cannot be easily measured since the effect of modeling errors of the robot dynamics additionally drives the task space error dynamics (42) in a complicated manner and makes it more difficult to analyze. Further investigation will be required to gain better understanding of this interference issue.

C. Torque based Control

1) Khatib's Controller [7]: A prominent framework for torque based redundancy resolution scheme was proposed based on operational space control in [7].

The dynamics which describe the motion of the task space coordinates are given by

$$\bar{\mathbf{M}}(\mathbf{x})\ddot{\mathbf{x}} + \bar{\mathbf{C}}(\mathbf{x}, \dot{\mathbf{x}}) + \bar{\mathbf{g}}(\mathbf{x}) = \mathbf{F} \tag{43}$$

where

$$\bar{\mathbf{M}} = (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1} \tag{44}$$
$$\bar{\mathbf{C}} = (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}(\mathbf{J}\mathbf{M}^{-1}\mathbf{C} - \dot{\mathbf{J}}\dot{\mathbf{q}}) \tag{45}$$
$$\bar{\mathbf{g}} = (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{M}^{-1}\mathbf{g}. \tag{46}$$

From these equations, a control law for the task space dynamics (43) is designed to achieve asymptotic tracking of the desired endeffector trajectory $\mathbf{x}_d$ as

$$\mathbf{F} = \bar{\mathbf{M}}\ddot{\mathbf{x}}_r + \bar{\mathbf{C}} + \bar{\mathbf{g}} \tag{47}$$

where

$$\ddot{\mathbf{x}}_r = \ddot{\mathbf{x}}_d + \mathbf{K}_p(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{K}_p(\mathbf{x}_d - \mathbf{x}). \tag{48}$$

From (43) and (47), it is straightforward to see

$$\ddot{\mathbf{e}} + \mathbf{K}_p\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = 0 \tag{49}$$

suggesting that $\mathbf{e} \to 0$ as $t \to \infty$, where $\mathbf{e} = \mathbf{x}_d - \mathbf{x}$.

A joint torque command is formulated as

$$\begin{aligned}
\boldsymbol{\tau} &= \mathbf{J}^T\mathbf{F} + (\mathbf{I} - \mathbf{J}^T\bar{\mathbf{J}}^T)\boldsymbol{\tau}_{null} && (50)\\
&= \mathbf{J}^T(\bar{\mathbf{M}}\ddot{\mathbf{x}}_r + \bar{\mathbf{C}} + \bar{\mathbf{g}}) + (\mathbf{I} - \mathbf{J}^T\bar{\mathbf{J}}^T)\boldsymbol{\tau}_{null} && (51)\\
&= \mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}[\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{J}\mathbf{M}^{-1}(\mathbf{C} + \mathbf{g})] \\
&\quad + (\mathbf{I} - \mathbf{J}^T\bar{\mathbf{J}}^T)\boldsymbol{\tau}_{null} && (52)\\
&= \mathbf{M}\bar{\mathbf{J}}[\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{J}\mathbf{M}^{-1}(\mathbf{C}+\mathbf{g})] + (\mathbf{I} - \mathbf{J}^T\bar{\mathbf{J}}^T)\boldsymbol{\tau}_{null} && (53)
\end{aligned}$$

where $\bar{\mathbf{J}}$ is the inertia-weighted pseudoinverse defined by

$$\bar{\mathbf{J}} = \mathbf{M}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1} \tag{54}$$

and $\boldsymbol{\tau}_{null}$ is an arbitrary torque vector. In this paper, we choose $\boldsymbol{\tau}_{null} = -\mathbf{K}_{q,d}\dot{\mathbf{q}} - \alpha\nabla g$. It can be shown that the inertia-weighed pseudoinverse of the Jacobian matrix is the unique choice which dynamically decouples the joint torque vector $\boldsymbol{\tau}$ into the task space forces $\mathbf{F}$ and the torques acting only on the null space motions of the manipulator $(\mathbf{I} - \mathbf{J}^T\bar{\mathbf{J}}^T)\boldsymbol{\tau}_{null}$ [7], [8]. In other words, the null space term $(\mathbf{I} - \mathbf{J}^T\bar{\mathbf{J}}^T)\boldsymbol{\tau}_{null}$ in (50) does not produce any force, and thus motion, in the task space.

2) Dynamical Decoupling Controller Variation 1 (without nullspace pre-multiplication of $\mathbf{M}$*, and compensation of* $\mathbf{C}$ *and* $\mathbf{g}$ *in joint space):* As presented in the previous section, in Khatib's redundancy resolution scheme [7], Coriolis and gravitational terms are compensated in the task space. Motivated by the discussions in [8], we suggest a variant of torque based control by pre-compensating Coriolis and gravitational terms in joint space.

Consider the control law

$$\begin{aligned}
\boldsymbol{\tau} &= \mathbf{C} + \mathbf{g} + \mathbf{J}^T\mathbf{F} + (\mathbf{I} - \mathbf{J}^T\bar{\mathbf{J}}^T)\boldsymbol{\tau}_{null} \\
&= \mathbf{C} + \mathbf{g} + \mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + (\mathbf{I} - \mathbf{J}^T\bar{\mathbf{J}}^T)\boldsymbol{\tau}_{null} \\
&= \mathbf{M}\bar{\mathbf{J}}(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + \mathbf{C} + \mathbf{g} + (\mathbf{I} - \mathbf{J}^T\bar{\mathbf{J}}^T)\boldsymbol{\tau}_{null} && (55)
\end{aligned}$$

where

$$\mathbf{F} = (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) \tag{56}$$
$$\boldsymbol{\tau}_{null} = -\mathbf{K}_{q,d}\dot{\mathbf{q}} - \alpha\nabla g. \tag{57}$$

With this control law, we have the following task space and null space closed loop dynamics, respectively:

Task space:

$$\ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\dot{\mathbf{e}} = 0 \tag{58}$$

Null space:

$$(\mathbf{I} - \mathbf{J}^T\bar{\mathbf{J}}^T)(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}_{q,d}\dot{\mathbf{q}} + \alpha\mathbf{K}_w(\mathbf{q} - \mathbf{q}_{rest})) = 0 \tag{59}$$

The task space error dynamics (58) suggest asymptotic tracking of the task space desired trajectory $\mathbf{x}_d$. However, the exact behavior of the nullspace dynamics cannot be easily determined, as careful analysis of (59) suggests.

3) Dynamical Decoupling Controller Variation 2 (with nullspace pre-multiplication of $\mathbf{M}$*, and compensation of* $\mathbf{C}$ *and* $\mathbf{g}$ *in joint space):* In (55) above, it is possible to choose the null space vector $\boldsymbol{\tau}_{null}$ as

$$\boldsymbol{\tau}_{null} = \mathbf{M}\ddot{\mathbf{q}}_0. \tag{60}$$

With this choice of the null space vector, the control law will be

$$\begin{aligned}
\boldsymbol{\tau} &= \mathbf{C} + \mathbf{g} + \mathbf{J}^T\mathbf{F} + (\mathbf{I} - \mathbf{J}^T\bar{\mathbf{J}}^T)\mathbf{M}\ddot{\mathbf{q}}_0 \\
&= \mathbf{C} + \mathbf{g} + \mathbf{J}^T\mathbf{F} + \mathbf{M}(\mathbf{I} - \bar{\mathbf{J}}\mathbf{J})\ddot{\mathbf{q}}_0 \\
&= \mathbf{M}(\bar{\mathbf{J}}(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + (\mathbf{I} - \bar{\mathbf{J}}\mathbf{J})\ddot{\mathbf{q}}_0) + \mathbf{C} + \mathbf{g} && (61)
\end{aligned}$$

where

$$\mathbf{F} = (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) \tag{62}$$
$$\ddot{\mathbf{q}}_0 = -\mathbf{K}_{q,d}\dot{\mathbf{q}} - \alpha\nabla g. \tag{63}$$

With this control law, we have the following task space and null space closed loop dynamics respectively:

Task space:

$$\ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\dot{\mathbf{e}} = 0 \tag{64}$$

Null space:

$$(\mathbf{I} - \bar{\mathbf{J}}\mathbf{J})(\ddot{\mathbf{q}} + \mathbf{K}_{q,d}\dot{\mathbf{q}} + \alpha\mathbf{K}_w(\mathbf{q} - \mathbf{q}_{rest})) = 0 \qquad (65)$$

The error dynamics (58) suggest asymptotic tracking of the task space desired trajectory $\mathbf{x}_d$, and we have slightly more simplified null space dynamics as compared to (59) ;however, still difficult to analyze and conclude stability in nullspace.

IV. Experimental Setup

For experimental evaluations, we used the Sarcos Master Arm, a seven degree-of-freedom hydraulically actuated anthropomorphic robot arm (Fig. 1). In order to implement the controllers introduced above, an inverse dynamics model of the robot is required. Additionally, torque based control schemes require explicit representation of the inertia matrix. As CAD data only captures the complex dynamics of idealized rigid bodies, and not the hydraulics routing throughout the arm and the hydraulic actuators, we resorted to parameter estimation methods [13] to obtain the necessary inertial and center of mass parameters of the robot arm. Interestingly, the base parameter set obtained from this least squares estimation procedure resulted in a non-positive definite inertia matrix $\mathbf{M}(\mathbf{q})$ at certain configurations of the robot, indicating a physically inconsistent base parameter set. If this physically inconsistent model were used for inverse dynamics control, the performance of the robot would be degraded and sometimes could even cause instabilities.

In [14], a sufficient condition is discussed to test whether the inertia matrix is positive definite, and to check whether a given parameter set is physically consistent. However, this method does not provide a parameter estimation procedure to guarantee physical consistency of the inertia matrix which we need in practice. For this purpose, we derived a novel nonlinear parameter estimation method, which ensures physically consistent inertia parameters by using a reparameterization with the help of Cholesky decompositions [15]. Without this nonlinear parameter identification method, most of the above controllers would not be implementable in a stable fashion.

V. Experimental Evaluations

The purpose of this paper is to implement the various controllers mentioned above on the same experimental platform and empirically evaluate their properties. As a benchmark task movement, we consider the task of tracking a planar "figure-8" pattern in task space at three different speeds (slow speed: 8 seconds per cycle, medium speed: 4 seconds per cycle, and fast speed: 2 seconds per cycle). We compare the tracking results of the following eight controllers and discuss their practical implementation issues.

1) velocity based control with joint velocity integration (Sec. III-A.1, (14))

Fig. 1. 7 degree-of-freedom hydraulic robot: Sarcos Master Arm

2) velocity based control without joint velocity integration (Sec. III-A.2, (19))
3) Hsu's acceleration based control (Sec. III-B.1, (20))
4) Simplified acceleration based control variation 1 with nullspace premultiplication of $\mathbf{M}$ (Sec. III-B.2, (32))
5) Simplified acceleration based control variation 2 without nullspace premultiplication of $\mathbf{M}$ (Sec. III-B.3, (38))
6) Khatib's torque based control (Sec. III-C.1, (53))
7) Dynamical decoupling control variation 1 without nullspace premultiplication of $\mathbf{M}$ (Sec. III-C.2, (55))
8) Dynamical decoupling control variation 2 with nullspace premultiplication of $\mathbf{M}$ (Sec. III-C.3, (61))

Figure 2 shows the experimental results of the end-point trajectories in the slow figure-8 pattern (8 seconds per cycle) and Figure 3 shows the results of the fast figure-8 pattern (2 seconds per cycle). Each trajectory has some offset from the target trajectory (thin black solid line). This offset is primarily due to an imperfect dynamics model of the robot. In Table I, the root mean squared errors (RMS) between the actual and target task space trajectory is given as a quantitative comparison. In numerical simulations, all the controllers worked equally well. However, in the experiments, we observed qualitative differences in each controller under the influence of practical effects such as inaccuracy of the parameters of the rigid dynamics model, sensory noise, and unmodeled nonlinear dynamics such as complex hydraulics dynamics. These practical implementation issues are discussed as follows:

- **Velocity based approach:** Velocity based controllers (controllers 1 and 2) were straightforward to implement. However, there was a practical limitation of the choice of task space position gain to improve tracking performance because the task space position gain effectively appears as a task space damping gain (cf. (19)) through

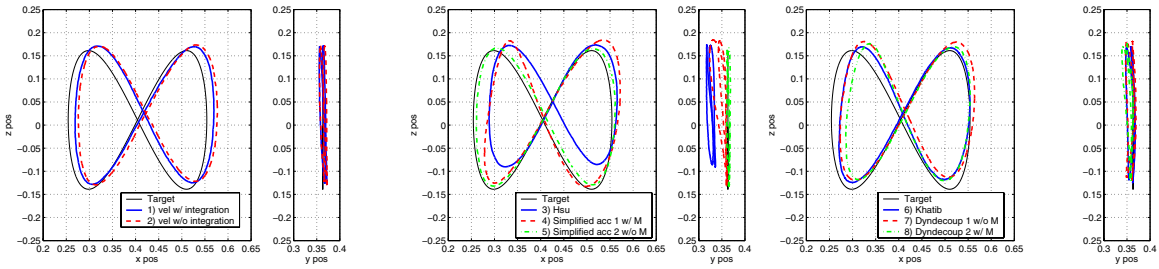

Fig. 2. Tracking results of slow movement (8 seconds per period). Left: 1) velocity based controller w/ joint velocity integration and 2) velocity based controller w/o joint velocity integration. Center: 3) Hsu's controller, 4) simplified acceleration controller w/ nullspace premultiplication of $\mathbf{M}$, 5) and simplified acceleration controller w/o nullspace premultiplication of $\mathbf{M}$, Right: 6) Khatib's controller, 7) dynamical decouping controller w/o nullspace premultiplication of $\mathbf{M}$, and 8) dynamical decoupling controller w/ nullspace premultiplication of $\mathbf{M}$.

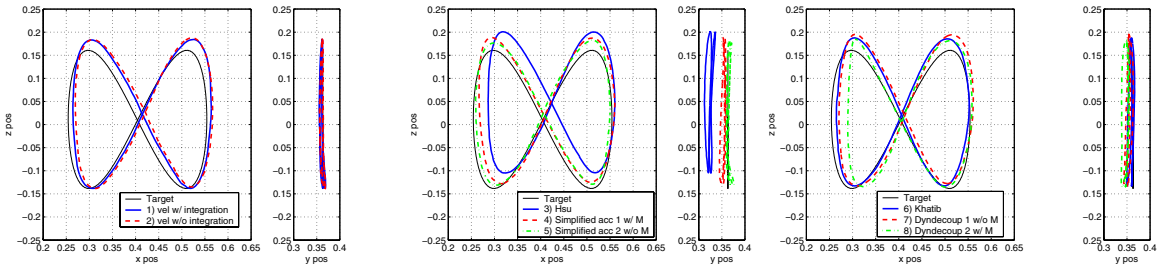

Fig. 3. Tracking results of fast movement (2 seconds per period). Left: 1) velocity based controller w/ joint velocity integration and 2) velocity based controller w/o joint velocity integration. Center: 3) Hsu's controller, 4) simplified acceleration controller w/ nullspace premultiplication of $\mathbf{M}$, 5) and simplified acceleration controller w/o nullspace premultiplication of $\mathbf{M}$, Right: 6) Khatib's controller, 7) dynamical decouping controller w/o nullspace premultiplication of $\mathbf{M}$, and 8) dynamical decoupling controller w/ nullspace premultiplication of $\mathbf{M}$.

TABLE I

ROOT MEAN SQUARED ERROR (RMS) [m] RESULTS OF THE TRACKING WITH THE DIFFERENT CONTROL LAWS

	1) vel. w/ integration	2) vel. w/o integration	3) Hsu	4) acc. 1 w/ $\mathbf{M}$	5) acc. 2 w/o $\mathbf{M}$	6) Khatib	7) dyn. dec. 1 w/o $\mathbf{M}$	8) dyn. dec. 2 w/ $\mathbf{M}$
slow speed	0.0198	0.0250	0.0562	0.0338	0.0101	0.0191	0.0270	0.0292
medium speed	0.0176	0.0229	0.0560	0.0303	0.0111	0.0190	0.0269	0.0289
fast speed	0.0182	0.0214	0.0560	0.0250	0.0177	0.0197	0.0288	0.0284

numerical differentiation of the reference joint velocities to obtain the reference joint accelerations. We found that increasing the task space position gain lead to instability due to too high velocity gains and also noise in the velocity measurements.

- **Remarks on Hsu's controller:** In Hsu's controller (controller 3), it was difficult to find robust control in null space and much effort was required to tune the parameters of the relatively unintuitive and complex null-space optimization term. We noticed a significant problem in attemping to stabilize low inertia DOFs like the wrist, and also suffered from significant tracking noise due to the numerical derivation of the time derivative of the pseudo inverse.
- **Torque based control with inertia-weighted pseudoinverse:** In torque based control (controllers 6–8), we observed a slightly skewed figure 8 pattern (e.g., see Figure 2 right) while theoretically this approach would achieve dynamical decoupling between task space and null space. Presumably, this is due to inaccuracies of the estimated inertia matrix. Moreover, computing the inertia-weighted pseudoinverse required a ridge regression technique in order to obtain a numerically stable inverse of the inertia matrix.
- **Effect of premultiplying the nullspace term by the inertia matrix:** In acceleration and torque based control, if the nullspace term is premultiplied by the inertia matrix (controllers 3, 4 and 8), we observed some

strange null space movement. This is most likely due to the inaccuracy of the estimated inertia matrix. We suspect that inaccurate off-diagonal elements of the estimated inertia matrix introduce undesirable coupling among joint movements.

- **Performance comparison:** Among all the controllers, the experiments demonstrated that the simplified acceleration based control variation 2 (controller 5) introduced in Section III-B.3 was the most promising approach in terms of task performance, ease of parameter tuning, and general robustness. While our analysis implies that there is interference between the task space and null space, our evaluations suggest that this is not a significant problem in practice, at least not in the test movements that we explored. We will address this issue in our future work.

VI. Conclusion

This paper presented comparative empirical investigations of various redundancy resolution algorithms. We first reviewed several redundancy algorithms suggested in the literature and also introduced some new variants of the previous approaches. We then experimentally evaluated the performance of these algorithms on a seven-degree-of-freedom anthropomorphic robot arm.

Our results indicate that algorithms that make sophisticated use of the robot model are in general more sensitive to errors in the robot model than simpler algorithms, even if these simpler algorithms are theoretically inferior. In particular, a robust resolved acceleration-based controller that we introduced in this paper performed remarkably well, and in general better than all other tested algorithms. As we expect that modeling errors will remain inevitable and significant for complex robots like humanoids, such simple and computationally efficient algorithms may be the best compromise for control with redundant degrees of freedom.

From a viewpoint of optimal control framework, our mathematical study suggests that some of the redundacy resolution algorithms discussed in this paper can be derived from a generalized Gauss' principle with different metrics for a cost function in a unified manner [16]. We hope that this result provides insight into further understanding of the properties of these redundancy resolution algotithms.

Future work will address further experiments with the suggested various redundancy algorithms on different robotic platforms and will also investigate robustness issues in the presence of external perturbations and contact forces. We will also address learning algorithms for redundancy resolution, applications to balance control for biped locomotion, and behavioral studies to identify the principles of redundancy resolution in humans' behavior.

Acknowledgment

We are grateful to Professor Roy Featherstone for valuable discussions and clarifications of the issues on redundancy resolution.

References

[1] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 7, no. 12, pp. 868–871, 1977.

[2] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.

[3] P. Hsu, J. Hauser, and S. Sastry, "Dynamic control of redundant manipulators," *Journal of Robotic Systems*, vol. 6, no. 2, pp. 133–148, 1989.

[4] A. D. Luca and G. Oriolo, "Issues in acceleration resolution of robot redundancy," in *3rd IFAC Symposium on Robot Control*, 1991, pp. 93–98.

[5] K. Senda, "Quasioptimal control of space redundant manipulators," in *AIAA Guidance, Navigation, and Control Conference*, 1999, pp. 1877–1885.

[6] J. M. Hollerbach and K. C. Suh, "Redundancy resolution of manipulators through torque optimization," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 4, pp. 308–316, 1987.

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

[8] R. Featherstone and O. Khatib, "Load independence of the dynamically consistent inverse of the jacobian matrix," *International Journal of Robotics Research*, vol. 16, no. 2, pp. 168–170, 1997.

[9] S. Arimoto, "A natural resolution of bernstein's degrees-of-freedom problem in case of multi-joint reaching," in *Proceedings of IEEE International Conference on Robotics and Biomimetics*, 2004.

[10] S. Schaal, "Learning from demonstration," in *Advances in Neural Information Processing Systems*, M. C. Mozer, M. Jordan, and T. Petsche, Eds., 1997, vol. 9, pp. 1040–1046.

[11] G. Tevatia and S. Schaal, "Inverse kinematics for humanoid robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2000, pp. 294–299.

[12] A. D'Souza, S. Vijayakumar, and S. Schaal, "Learning inverse kinematics," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001, pp. 298–303.

[13] C. H. An, C. G. Atkeson, and J. M. Hollerbach, *Model-Based Control of a Robot Manipulator*. MIT Press, 1988.

[14] K. Yoshida and W. Khalil, "Verification of the positive definiteness of the inertial matrix of manipulators using base inertial parameters," *International Journal of Robotics Research*, vol. 19, no. 5, pp. 498–510, 2000.

[15] S. Schaal, R. Cory, J. Nakanishi, M. Mistry, and J. Peters, "Physcially consistent nonlinear parameter identification for rigid body dynamics," (in preparation).

[16] J. Peters, M. Mistry, F. Udwadia, R. Cory, J. Nakanishi, and S. Schaal, "A unifying methodology for the control of robotic systems," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, (to appear).