Operational Space Control: A Theoretical and Empirical Comparison

Jun Nakanishi
ICORP, Computational Brain Project,
Japan Science and Technology Agency,
Saitama 332-0012, Japan
and
ATR Computational Neuroscience Laboratories,
Department of Humanoid Robotics and
Computational Neuroscience,
Kyoto 619-0288, Japan
jun@atr.jp,

Rick Cory
Computer Science and Artificial Intelligence Laboratory,
Massachusetts Institute of Technology,
Cambridge, MA 02139, USA
rcory@csail.mit.edu

Michael Mistry
Jan Peters
Computer Science & Neuroscience,
University of Southern California, Los Angeles,
CA 90089-2520, USA
{mmistry, jrpeters}@usc.edu

Stefan Schaal
ATR Computational Neuroscience Laboratories,
Department of Humanoid Robotics and
Computational Neuroscience,
Kyoto 619-0288, Japan
and
Computer Science & Neuroscience,
University of Southern California, Los Angeles,
CA 90089-2520, USA
sschaal@usc.edu

Abstract

Dexterous manipulation with a highly redundant movement system is one of the hallmarks of human motor skills. From numerous behavioral studies, there is strong evidence that humans employ compliant task space control, i.e. they focus control only on task variables while keeping redundant degrees-of-freedom as compliant as possible. This strategy is robust towards unknown disturbances and simultaneously safe for the operator and the environment. The theory of operational space control in robotics aims to achieve similar performance properties. However, despite various compelling theoretical lines of research, advanced operational space control is hardly found in actual robotics implementations, in particular new kinds of robots like humanoids and service robots, which would strongly profit from compliant dexterous manipulation. To analyze the pros and cons of different approaches to operational space control, this paper focuses on a theoretical and empirical evaluation of different methods that have been suggested in the literature, but also some new variants of op-

The International Journal of Robotics Research
Vol. 27, No. 6, June 2008, pp. 737–757
DOI: 10.1177/0278364908091463

Figures 1, 3–7 appear in color online: http://ijr.sagepub.com

erational space controllers. We address formulations at the velocity, acceleration, and force levels. First, we formulate all controllers in a common notational framework, including quaternion-based orientation control, and discuss some of their theoretical properties. Second, we present experimental comparisons of these approaches on a seven-degree-of-freedom anthropomorphic robot arm with several benchmark tasks. As an aside, we also introduce a novel parameter estimation algorithm for rigid body dynamics, which ensures physical consistency, as this issue was crucial for our successful robot implementations. Our extensive empirical results demonstrate that one of the simplified acceleration-based approaches can be advantageous in terms of task performance, ease of parameter tuning, and general robustness and compliance in the face of inevitable modeling errors.

KEY WORDS—operational space control, redundant manipulators, compliant control, theory, review, empirical comparison

1. Introduction

Understanding the principles of natural movement generation has been one of the most interesting and important open problems in the fields of robotics and the neural control of movement. Among the key characteristics of human movement are compliant control and intelligent exploitation of a highly redundant movement system. From numerous behavioral and neuroscientific studies (e.g., Saltzman and Kelso (1987); Scholz and Schoner (1999); Todorov and Jordan (2002); Mistry et al. (2005); Schaal and Schweighofer (2005)), it is known that humans focus on compliant task space control, i.e. they control task relevant variables with higher gain, while keeping redundant degrees as uncontrolled and compliant as possible.

In classical robotics, compliant redundant actuator systems have mostly been avoided in favor of robots with minimal degrees-of-freedom (DOFs), analytical solutions to inverse kinematics, and stiff high-gain control in order to maximize tracking accuracy with low computational control effort. For single-task robots in highly specialized (mostly static) environments, as typical in industrial applications, this approach is viable. In contrast, the new wave of humanoid, assistive, and entertainment robots is characterized by having a large number of redundant DOFs, and the need to perform multiple tasks in sequence or parallel, and to operate in non-deterministic human environments with unforeseeable force perturbations (Khatib et al. 2004). For these systems, compliant task space control seems to be the most suitable control approach in order to maximize safety and gently and robustly reject disturbances.

Operational space control (or task space control) revolves around how to resolve redundancy and how to produce appropriate motor commands in configuration space such that both end-effector and DOFs achieve a prescribed level of impedance. The literature offers a wide variety of techniques. A typical approach makes use of some form of a Jacobian pseudo-inverse with local null space optimization in order to determine the inverse kinematics transformation (Whitney 1969; Liegeois 1977; Baillieul and Martin 1990)[1]. Within this general framework, kinematic redundancy can be resolved at the velocity (Nakamura et al. 1987), acceleration (Hollerbach and Suh 1987; Hsu et al. 1989; De Luca and Oriolo 1991; De Luca et al. 1992; Senda 1999), and force levels (Khatib 1987; Featherstone and Khatib 1997), where the desired joint velocities, accelerations, and torques are computed, respectively, for a desired end-effector velocity, acceleration, and force.

While kinematic redundancy resolutions in the spirit of resolved-motion-rate-control (Whitney 1969; Liegeois 1977) and resolved-acceleration-control (Luh et al. 1980) have found widespread application in many robotics projects, the application of proper control with redundancy resolution, i.e. operational space control, is less ubiquitous, and almost completely absent in complex redundant robots like humanoids and assistive robots. Moreover, stability properties of most algorithms are not fully understood, and more extensive empirical evaluations on actual robots are scarce. One noteworthy exception is a recent study by Arimoto et al. (2005), which proposes a provably stable redundancy resolution algorithm for reaching movements based on the Jacobian transpose with joint velocity damping. Some of the advantages of the Jacobian transpose approach, e.g. Sciavicco and Siciliano (1988); Arimoto et al. (2005), are that it is computationally efficient and it does not suffer from singularity problems associated with matrix inversion of the Jacobian, while the Jacobian pseudo-inverse approach would require some techniques to stabilize matrix inversion near singularities (Nakamura and Hanafusa 1986; Wampler 1986; Wampler and Leifer 1988). However, at the current state of the Jacobian transpose approach, it is difficult to apply the Jacobian transpose method to tasks other than reaching for a static target, e.g., dynamic tasks which require control of position, velocity, and/or acceleration of the task coordinates (as is the case when tracking a moving target or manipulating a dynamic object (Schaal 1997)).

The goal of this paper is to examine the suitability of operational space control methods for complex high DOF robots. While impressive results have been generated with advanced operational space controllers on simulated humanoid robots in recent studies (Khatib et al. 2004; Sentis and Khatib 2005), there is a lack of understanding of whether simpler control methods could achieve similar results, and in how far the success of idealized simulations extends to actual robot implementations. We will focus on the most prominent velocity-based, acceleration-based, and force-based controllers in the literature, and also several new variants. All the controllers

[1] We only consider methods of local (or differential) redundancy resolution, and not global methods (e.g., Nakamura and Hanafusa (1987); Suh and Hollerbach (1987); Kazerounian and Wang (1988); Martin et al. (1989); Baillieul and Martin (1990)), as the latter cannot easily be inserted into real-time control approaches.

are formulated in a unified notational framework, including quaternion-based orientation control (Yuan 1988; Caccavale et al. 1998; Xian et al. 2004). Efficient real-time implementations are achieved using Featherstone's spatial algorithms (Featherstone 1987) on a Sarcos Dexterous Master arm. We also introduce a parameter estimation algorithm for rigid body dynamics which ensures physical consistency of the parameters, as this issue was crucial in our implementations. In the end, we discuss the practical properties of different approaches, particularly in light of inevitable modeling errors of the robot dynamics.

2. Problem Setup

We focus on rigid body dynamics systems, whose equations of motion 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 (generalized) joint angle vector, $\mathbf{M}(\mathbf{q})$ is the inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is the Coriolis/centripetal vector, $\mathbf{g}(\mathbf{q})$ is the gravity vector, and $\boldsymbol{\tau}$ is the joint torque vector. The "dot" above a variable denotes its time derivative.

The forward kinematics and differential relationship between the joint coordinates and the operational space 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 operational space coordinate vector. In the following, we denote the desired task space positions, velocities, and accelerations as $\mathbf{x}_d$, $\dot{\mathbf{x}}_d$, and $\ddot{\mathbf{x}}_d$, respectively.

The idea of redundancy resolution at the velocity level is to invert (3) to compute the desired joint velocities as

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

where, in general, $\mathbf{J}^{\dagger}$ is the pseudo-inverse defined by $\mathbf{J}^{\dagger} = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}$ and $\boldsymbol{\xi}_1$ is an arbitrary vector. Note that there is a subtle issue in terms of which state is used to compute the Jacobian, i.e. the current state $\mathbf{q}$ or the desired state $\mathbf{q}_d$—the latter would obviously be the mathematically correct way. However, since most controllers developed below will have no access to $\mathbf{q}_d$ or $\dot{\mathbf{q}}_d$, we will always use the current state $(\mathbf{q}, \dot{\mathbf{q}})$ in the control laws, implicitly assuming that the error between the desired (or reference) state and the current state is small enough. $(\mathbf{I}-\mathbf{J}^{\dagger}\mathbf{J})$ projects $\boldsymbol{\xi}_1$ onto the null space of the Jacobian $\mathbf{J}$ (e.g., Liégeois (1977)) such that $\boldsymbol{\xi}_1$ can be interpreted as a desired velocity behavior that is only effective in the null space and does not interfere with the task achievement. Discussions of the properties of generalized pseudo-inverses for redundancy resolution can be found in Mussa-Ivaldi and Hogan (1991); Doty et al. (1993).

For redundancy resolution at the acceleration level, i.e. methods that directly compute joint accelerations from desired task space accelerations, one solves (4) for $\ddot{\mathbf{q}}$ and obtains

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

where $\boldsymbol{\xi}_2$ is a an arbitrary vector, similar to $\boldsymbol{\xi}_1$ (e.g., Hsu et al. (1989)), which controls the desired acceleration behavior in the null space.

The velocity-based approach has traditionally been preferred in many robotics applications due to its simplicity. However, for second-order systems, such as rigid body dynamics systems, the acceleration-based approach is more appealing, especially when used in conjunction with an inverse dynamics control approach that explicitly needs knowledge of accelerations in joint space. From a practical point of view, the inconvenient $\dot{\mathbf{J}}$ term in the acceleration-based controller can be obtained either from numerical differentiation or analytical formulae (Ahmed 1992).

Note that by equating (6) and the analytical time derivative of (5), in De Luca and Oriolo (1991); De Luca et al. (1992), a sufficient condition for these two redundancy resolution schemes to be consistent was derived as

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

while Senda (1999) formulated the consistency condition as

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

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

The acceleration-based form of redundancy resolution naturally extends to force-based redundancy methods, i.e. situations where the desired end-effector force is prescribed and the corresponding joint torques are computed directly. In essence, the inertia-weighted pseudo-inverse $\bar{\mathbf{J}} = \mathbf{M}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}$ replaces the Moore–Penrose pseudo-inverse in Equation (6), but several other changes need to be considered, too—this topic will be covered in more detail below. The appeal of force-based redundancy resolution stems from its dynamical decoupling property, which ensures that the joint torques that create operational space forces are orthogonal to the joint torques that create the null space behavior (Khatib 1987; Featherstone and Khatib 1997).

Throughout this paper, we will employ a simple optimization criterion for redundancy resolution, with a cost function

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

where $\mathbf{K}_w$ is a positive definite (PD) diagonal weighting matrix and $\mathbf{q}_{\text{rest}}$ is some rest (preferred) posture. This criterion has been shown to be useful when creating human-like movement in anthropomorphic robots (Cruse et al. 1990), but other criteria could also be used for different purposes, e.g. based on manipulability indices (Yoshikawa 1984, 1985), distance from the obstacle (Maciejewski and Klein 1985), or joint torques (Hollerbach and Suh 1987). We chose to restrict our developments and evaluation to the simple cost in Equation (9), as this criterion is numerically unproblematic and intuitively easy to understand. Thus, empirical evaluations will not be contaminated by potential algorithmic and numerical problems resulting from the null space optimization. Future work will address a comprehensive evaluation of null space cost criteria for task space control in a dedicated publication.

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

3. Controller Formulations

We consider the following eight operational space controllers, which are of practical and/or theoretical relevance for robotics applications (for more details of each controller formulation, see the explanations in the following subsections):

- Velocity-based control:
  1. velocity-based control *with* joint velocity integration (Section 3.1.1);
  2. velocity-based control *without* joint velocity integration (Section 3.1.2).
- Acceleration-based control:
  3. Acceleration-based control as in Hsu et al. (1989) (Section 3.2.1);
  4. simplified acceleration control variation 1 (*with* null space pre-multiplication by the inertia matrix **M**) (Section 3.2.2);
  5. simplified acceleration control variation 2 (*without* null space pre-multiplication of **M**) (Section 3.2.3).
- Force-based control:
  6. Gauss control (Khatib 1987) (Section 3.3.1);
  7. dynamical decoupling control variation 1 (*without* null space pre-multiplication of **M**) (Section 3.3.2);
  8. dynamical decoupling control variation 2 (*with* null space pre-multiplication of **M**) (Section 3.3.3).

Each controller employs some form of inverse dynamics control such that high accuracy tracking can be achieved without the need for high PD gains, which is one prerequisite for compliant control. Except for the velocity-based controllers, there is no need for explicit joint space reference trajectories in the controllers, which allows the null space of the robot motion to be maximally unconstrained, which is another prerequisite for compliant control. Such potential for compliant motion is really what sets advanced operational space controllers apart from traditional joint space control methods.

3.1. Velocity-based Control

Velocity-based control computes the desired joint torques for a given end-effector velocity (Nakamura et al. 1987) in at least two different ways.

3.1.1. Velocity-based Control with Joint Velocity Integration

In this classical base-line controller, given the reference task space velocity command $\dot{\mathbf{x}}_r$, the reference joint velocities $\dot{\mathbf{q}}_r$ are obtained from the Liegeois' resolved motion rate control approach with null space optimization (Liegeois 1977)

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

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

where $\dot{\mathbf{x}}_d$ and $\mathbf{x}_d$ are the desired task space velocities and positions, respectively, $\mathbf{K}_p$ is a PD gain matrix, α is a positive constant, and $g(\mathbf{q})$ is a null space cost function, chosen to be Equation (9), as mentioned before.

The reference joint accelerations and positions are obtained by numerical differentiation and integration of the reference joint velocities (11), respectively, as

$$\ddot{\mathbf{q}}_r = \frac{\mathrm{d}}{\mathrm{d}t}\dot{\mathbf{q}}_r \simeq \frac{\dot{\mathbf{q}}_r(t) - \dot{\mathbf{q}}_r(t - \Delta t)}{\Delta t}, \qquad (12)$$

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

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

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

where $\mathbf{K}_{q,d}$ and $\mathbf{K}_{q,p}$ are PD gain matrices. With this control law, the task space error dynamics become

$$\begin{aligned} & \ddot{\mathbf{e}} + \mathbf{K}_p\dot{\mathbf{e}} + \mathbf{J}\dot{\mathbf{J}}^{\dagger}\mathbf{K}_p\mathbf{e} \\ + \quad & \mathbf{J}\mathbf{M}(\mathbf{q}_r)^{-1}\left(\mathbf{K}_{q,d}\mathbf{J}^{\dagger}\mathbf{K}_p\mathbf{e} + \mathbf{K}_{q,p}\int_{t_0}^{t}\mathbf{J}^{\dagger}\mathbf{K}_p\mathbf{e}\,\mathrm{d}t'\right) \\ = \quad & -\mathbf{J}\mathbf{M}(\mathbf{q}_r)^{-1}\left[\mathbf{K}_{q,d}(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_{q,p}(\mathbf{q}_d - \mathbf{q})\right. \\ + \quad & \left.\Delta\mathbf{M}\ddot{\mathbf{q}} + \Delta\mathbf{C} + \Delta\mathbf{g}\right], \qquad (15) \end{aligned}$$

where

$$\Delta\mathbf{M} = \mathbf{M}(\mathbf{q}_r) - \mathbf{M}(\mathbf{q}), \qquad (16)$$

$$\Delta\mathbf{C} = \mathbf{C}(\mathbf{q}_r, \dot{\mathbf{q}}_r) - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}), \qquad (17)$$

$$\Delta\mathbf{g} = \mathbf{g}(\mathbf{q}_r) - \mathbf{g}(\mathbf{q}), \qquad (18)$$

and

$$\mathbf{q}_d = \int_{t_0}^{t}\dot{\mathbf{q}}_d\,\mathrm{d}t', \qquad (19)$$

$$\dot{\mathbf{q}}_d = \mathbf{J}^{\dagger}\dot{\mathbf{x}}_d + \alpha(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})\xi_1, \qquad (20)$$

$$\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d + \mathbf{J}^{\dagger}\mathbf{K}_p\mathbf{e}, \qquad (21)$$

$$\mathbf{e} = \mathbf{x}_d - \mathbf{x}. \qquad (22)$$

While this commonly employed method is easy to implement and exhibits a large amount of practical robustness due to error stabilization terms both in task space due to Equation (10) and joint space due to the PD terms in Equation (14), there are several main disadvantages. One is that we do not use the information of the target accelerations, which usually leads to lower tracking performance or the need for high task space gains and servo rates. The other is the requirement of numerical differentiation and integration of the reference joint velocities to obtain the reference joint accelerations and joint positions. Numerical differentiation is rather sensitive to sensor noise, and integration has the problem of windup if the robot motion is unexpectedly constrained by an external disturbance for a long period, which ultimately accumulates a large error and gives rise to unrealizable joint torques and potentially catastrophic failures. As a last point, servoing an explicit joint space reference trajectory reduces null space compliance in comparison with all other methods (see below) that do not need such a reference[2].

3.1.2. Velocity-based Control without Joint Velocity Integration

As mentioned in the previous paragraph, when considering contact forces with the environment, it is desirable to avoid integrators. For a given reference task space velocity command $\dot{\mathbf{x}}_r$ and the reference joint velocities $\dot{\mathbf{q}}_r$ from Equations (10) and (11), an alternative velocity-based controller can be formulated that, by means of using an inverse dynamics control law, avoids velocity integration:

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \mathbf{K}_{q,d}(\dot{\mathbf{q}}_r - \dot{\mathbf{q}}), \qquad (23)$$

where the reference joint accelerations $\ddot{\mathbf{q}}_r$ are calculated through numerical differentiation as in (13). Note that the added joint velocity feedback term at the very right of Equation (23) is crucial for stability (see below).

The control law (23) can be rearranged as

$$\begin{aligned} \boldsymbol{\tau} &= \mathbf{M}\ddot{\mathbf{q}}_r + \mathbf{C} + \mathbf{g} + \mathbf{K}_{q,d}(\dot{\mathbf{q}}_r - \dot{\mathbf{q}}) \\ &= \mathbf{M}(\ddot{\mathbf{q}}_d + \mathbf{J}^{\dagger}\mathbf{K}_p\dot{\mathbf{e}} + \dot{\mathbf{J}}^{\dagger}\mathbf{K}_p\mathbf{e}) + \mathbf{C} + \mathbf{g} \\ &+ \mathbf{K}_{q,d}(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_{q,d}\mathbf{J}^{\dagger}\mathbf{K}_p\mathbf{e}, \qquad (24) \end{aligned}$$

where $\mathbf{e} = \mathbf{x}_d - \mathbf{x}$. With this control law, we have the following task space error dynamics:

$$\begin{aligned} & \ddot{\mathbf{e}} + \mathbf{K}_p\dot{\mathbf{e}} + \left(\mathbf{J}\mathbf{M}^{-1}\mathbf{K}_{q,d}\mathbf{J}^{\dagger}\mathbf{K}_p + \mathbf{J}\dot{\mathbf{J}}^{\dagger}\mathbf{K}_p\right)\mathbf{e} \\ = \quad & -\mathbf{J}\mathbf{M}^{-1}\mathbf{K}_{q,d}(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}). \qquad (25) \end{aligned}$$

This formulation reveals a practical problem with this controller, as the task space position gain $\mathbf{K}_p$ also affects the task space velocity gain in the term multiplying $\dot{\mathbf{e}}$ in (25). Thus, task space stiffness and damping cannot be controlled independently, which limits tracking and impedance performance. From (24), it can also be recognized that $\mathbf{K}_{q,d} > 0$ is required to avoid that $\mathbf{e}$ has a non-zero stiffness term for steady-state behavior, i.e. in order to have steady-state stability (where $\dot{\mathbf{J}}^{\dagger} = 0$). As in the previous velocity-based controller, this method also has a disadvantage in that it ignores the information of the target accelerations.

3.2. Acceleration-based Control

Instead of velocities, acceleration-based control computes desired joint accelerations for a given end-effector reference, this time specified as a reference acceleration (Hsu et al. 1989; De Luca and Oriolo 1991; Senda 1999; Hollerbach and Suh 1987). For tracking with a second-order system such as a rigid body dynamics robot, this formulation is the most natural and offers improved tracking ability due to explicit incorporation of acceleration information. Various versions of such controllers exist:

[2] There is also evidence that humans do not maintain explicit joint space trajectories in task space control (Mistry et al. 2005).

3.2.1. Acceleration-based Controller in Hsu et al. (1989)

This early approach was proposed in Hsu et al. (1989). For a given reference acceleration in task space

$$\ddot{\mathbf{x}}_r = \ddot{\mathbf{x}}_d + \mathbf{K}_d(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{K}_p(\mathbf{x}_d - \mathbf{x}), \tag{26}$$

the control law is given by

$$\boldsymbol{\tau} = \mathbf{M}\left(\mathbf{J}^\dagger(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + \phi_N\right) + \mathbf{C} + \mathbf{g}, \tag{27}$$

where

$$\begin{aligned} \phi_N &= (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})(\dot{\mathbf{h}} + \mathbf{K}_N\mathbf{e}_N) \\ &- (\mathbf{J}^\dagger\dot{\mathbf{J}}\mathbf{J}^\dagger + \dot{\mathbf{J}}^\dagger)\mathbf{J}(\mathbf{h} - \dot{\mathbf{q}}), & (28) \\ \mathbf{e}_N &= (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})(\mathbf{h} - \dot{\mathbf{q}}), & (29) \end{aligned}$$

$\mathbf{h}$ is a vector function $\mathbf{h} = -\alpha\nabla g$, and $\mathbf{K}_N$ is a PD gain matrix. It can be shown that the control law (27) with (26) yields the task space tracking error dynamics

$$\ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = 0, \tag{30}$$

which implies that this controller achieves asymptotic tracking in operational space, i.e. $\mathbf{e} \to 0$ as $t \to \infty$ assuming that $\mathbf{J}^\dagger$ is full rank (Hsu et al. 1989).

In the original paper, Hsu et al. (1989) propose a rather complex null space vector ϕ_N in (28) in order to provide a proof of convergence for velocity tracking in null space. Besides mathematical convenience, it is not stated in Hsu et al. (1989) which intuition led to the formulation of this null space term. We analyzed this term, which, after some rearrangement, can be written as

$$\begin{aligned} \phi_N &= (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})[\dot{\mathbf{h}} + \mathbf{K}_N(\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})(\mathbf{h} - \dot{\mathbf{q}})] \\ &- (\mathbf{J}^\dagger\dot{\mathbf{J}}\mathbf{J}^\dagger + \dot{\mathbf{J}}^\dagger)\mathbf{J}(\mathbf{h} - \dot{\mathbf{q}}) \\ &= (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})[\dot{\mathbf{h}} + \mathbf{K}_N(\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})(\mathbf{h} - \dot{\mathbf{q}})] \\ &- (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\dot{\mathbf{J}}^\dagger\mathbf{J}(\mathbf{h} - \dot{\mathbf{q}}) \\ &= (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})[\dot{\mathbf{J}}^\dagger\mathbf{J}(\dot{\mathbf{q}} - \mathbf{h}) + \dot{\mathbf{h}} \\ &- \mathbf{K}_N(\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})(\dot{\mathbf{q}} - \mathbf{h})]. & (31) \end{aligned}$$

Thus, the controller in Hsu et al. (1989) is equivalent to the case where

$$\begin{aligned} \boldsymbol{\tau} &= \mathbf{M}\ddot{\mathbf{q}}_r + \mathbf{C} + \mathbf{g}, & (32) \\ \ddot{\mathbf{q}}_r &= \mathbf{J}^\dagger(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + \phi_N & (33) \\ &= \mathbf{J}^\dagger(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\boldsymbol{\xi}_2, & (34) \\ \boldsymbol{\xi}_2 &= \underbrace{\dot{\mathbf{J}}^\dagger\mathbf{J}(\dot{\mathbf{q}} - \boldsymbol{\xi}_1) + \dot{\boldsymbol{\xi}}_1}_{\text{same as (7)}} - \mathbf{K}_N(\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})(\dot{\mathbf{q}} - \boldsymbol{\xi}_1), & (35) \end{aligned}$$

with $\boldsymbol{\xi}_1 = \mathbf{h} = -\alpha\nabla g$ in (35). Thus, from (7) and (31), we can deduce that the acceleration null space vector (35) was derived from an analytical differentiation of a velocity-based redundancy resolution approach, which has one additional term, i.e. the right-hand term in (35). This term helps to achieve some of the stability proofs in the original paper (Hsu et al. 1989) and seems to serve primarily analytical convenience.

For our robot implementations below, we choose $\boldsymbol{\xi}_1 = -\alpha\nabla g$ such that

$$\boldsymbol{\xi}_2 = \dot{\mathbf{J}}^\dagger\mathbf{J}(\dot{\mathbf{q}} + \alpha\nabla g) + (\dot{\nabla g}) - \mathbf{K}_N(\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})(\dot{\mathbf{q}} + \alpha\nabla g). \tag{36}$$

Note that in this controller the inertia matrix $\mathbf{M}$ pre-multiplies the null space term as

$$\boldsymbol{\tau} = \mathbf{M}\mathbf{J}^\dagger(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + \mathbf{C} + \mathbf{g} + \mathbf{M}\phi_N. \tag{37}$$

A recurrent topic of this paper will be that such a pre-multiplication of the null space term with $\mathbf{M}$ can practically be problematic if $\mathbf{M}$ has modeling errors.

3.2.2. Simplified Acceleration-based Control Variation 1 (With Null Space Pre-multiplication of M*)*

The surprising complexity of the null space term (28) in Hsu et al. (1989) raises the question of whether there are simpler alternative formulations for acceleration-based operational space controllers. A possible approach can be written as

$$\boldsymbol{\tau} = \mathbf{M}\ddot{\mathbf{q}}_r + \mathbf{C} + \mathbf{g}, \tag{38}$$

where

$$\begin{aligned} \ddot{\mathbf{q}}_r &= \mathbf{J}^\dagger(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\boldsymbol{\xi}_2, & (39) \\ \boldsymbol{\xi}_2 &= -\mathbf{K}_{q,d}\dot{\mathbf{q}} - \alpha\nabla g, & (40) \end{aligned}$$

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

In this controller, we replaced the null space vector $\boldsymbol{\xi}_2$ of Hsu et al. (1989) in (35) by introducing a Liegeois-like null space projection with damping term in joint space. This controller achieves asymptotic tracking in operational space since we have

$$\ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = 0. \tag{41}$$

A principled derivation of this controller was provided by Peters et al. (2005) from the viewpoint of analytical mechanics, which demonstrated that this controller shares exactly the same derivation and stability properties as Khatib's force-based operational space controller (Khatib 1987) (see below).

Note that in this controller the inertia matrix effectively pre-multiplies the null space term as

$$\boldsymbol{\tau} = \mathbf{M}\mathbf{J}^\dagger(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + \mathbf{C} + \mathbf{g} + \mathbf{M}(\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\boldsymbol{\xi}_2. \tag{42}$$

3.2.3. Simplified Acceleration-based Control Variation 2 (Without Null Space Pre-multiplication of **M***)*

From our practical experience, pre-multiplication of the null space optimization term by the inertia matrix $\mathbf{M}$ can be problematic if the inertia matrix has modeling inaccuracies. Thus, as a variant, we introduce an acceleration-based control law with null space-projected PD control term, i.e. a torque (and not an acceleration) null space command:

$$\begin{aligned}\boldsymbol{\tau} &= \mathbf{M}\ddot{\mathbf{q}}_r + \mathbf{C} + \mathbf{g} + (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})\boldsymbol{\xi}_2 && (43)\\ &= \mathbf{M}\mathbf{J}^{\dagger}(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + \mathbf{C} + \mathbf{g} \\ &+ (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})(-\mathbf{K}_{q,d}\dot{\mathbf{q}} - \alpha\nabla g), && (44)\end{aligned}$$

where

$$\begin{aligned}\ddot{\mathbf{q}}_r &= \mathbf{J}^{\dagger}(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}), && (45)\\ \boldsymbol{\xi}_2 &= -\mathbf{K}_{q,d}\dot{\mathbf{q}} - \alpha\nabla g, && (46)\end{aligned}$$

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

This control law results in the following closed loop dynamics:

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{M}\mathbf{J}^{\dagger}(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})(-\mathbf{K}_{q,d}\dot{\mathbf{q}} - \alpha\nabla g). \quad (47)$$

By pre-multiplying $\mathbf{J}\mathbf{M}^{-1}$ to both sides, (47) can be rearranged as

$$\begin{aligned}&\ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}\\ &= \mathbf{J}\mathbf{M}^{-1}(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})(\mathbf{K}_{q,d}\dot{\mathbf{q}} + \alpha\mathbf{K}_w(\mathbf{q}_{\text{rest}} - \mathbf{q})). && (48)\end{aligned}$$

Equation (48) indicates that there is interference between the range and null space since the vector on the right-hand side of (48) drives the task space tracking error dynamics as long the null space projection of $\boldsymbol{\xi}_2$ in (46) is not zero. Surprisingly, as our empirical results demonstrate in Section 6, we achieved excellent and robust tracking performance with this controller, despite the theoretical possibility of interference.

3.3. Force-based Control

Force-based operational control directly computes the desired joint torques for a given end-effector reference command, given as a force (Khatib 1987; Featherstone and Khatib 1997). Again, multiple versions can be considered:

3.3.1. Gauss Controller (Operational Space Controller in Khatib (1987))

A prominent framework for force-based redundancy resolution was proposed by Khatib (1987). The desired task dynamics are specified as

$$\bar{\mathbf{M}}(\mathbf{x})\ddot{\mathbf{x}} + \bar{\mathbf{C}}(\mathbf{x}, \dot{\mathbf{x}}) + \bar{\mathbf{g}}(\mathbf{x}) = \mathbf{F}, \quad (49)$$

where

$$\begin{aligned}\bar{\mathbf{M}} &= (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}, && (50)\\ \bar{\mathbf{C}} &= (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}(\mathbf{J}\mathbf{M}^{-1}\mathbf{C} - \dot{\mathbf{J}}\dot{\mathbf{q}}), && (51)\\ \bar{\mathbf{g}} &= (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}\mathbf{J}\mathbf{M}^{-1}\mathbf{g}. && (52)\end{aligned}$$

From these equations, a control law for the desired operational space dynamics (49) is designed as

$$\mathbf{F} = \bar{\mathbf{M}}\ddot{\mathbf{x}}_r + \bar{\mathbf{C}} + \bar{\mathbf{g}}, \quad (53)$$

where

$$\ddot{\mathbf{x}}_r = \ddot{\mathbf{x}}_d + \mathbf{K}_p(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{K}_p(\mathbf{x}_d - \mathbf{x}). \quad (54)$$

The corresponding joint torque control law becomes

$$\begin{aligned}\boldsymbol{\tau} &= \mathbf{J}^T\mathbf{F} + (\mathbf{I} - \mathbf{J}^T\bar{\mathbf{J}}^T)\boldsymbol{\tau}_{\text{null}} && (55)\\ &= \mathbf{M}\bar{\mathbf{J}}\left(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{J}\mathbf{M}^{-1}(\mathbf{C} + \mathbf{g})\right)\\ &+ (\mathbf{I} - \mathbf{J}^T\bar{\mathbf{J}}^T)\boldsymbol{\tau}_{\text{null}}, && (56)\end{aligned}$$

where $\bar{\mathbf{J}}$ is the inertia-weighted pseudo-inverse $\bar{\mathbf{J}} = \mathbf{M}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}$. This control law achieves asymptotic tracking in operational space as

$$\ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = 0. \quad (57)$$

In this paper, we choose $\boldsymbol{\tau}_{\text{null}} = -\mathbf{K}_{q,d}\dot{\mathbf{q}} - \alpha\nabla g$. This control law has many interesting characteristics (Khatib 1987; Featherstone and Khatib 1997; Bruyninckx and Khatib 2000), including the fact that it dynamically decouples operational and null space dynamics, and that it interferes with the natural dynamics of the robot in a minimal way, according to the so-called Gauss' principle (Udwadia and Kalaba 1996). A principled derivation of this control law was provided in Peters et al. (2005), which demonstrated its analytically close relationship with the simplified acceleration controller above. It should be noted that an implementation of this control law requires explicit knowledge of the inertia matrix $\mathbf{M}$ and the terms $\mathbf{C} + \mathbf{g}$ —in normal inverse dynamics computations, these terms are only used implicitly and are hidden in the recursive nature of the inverse dynamics algorithms. Computing $\mathbf{M}$ explicitly is costly as it requires $O(n^2)$ complexity, while normal efficient inverse dynamics algorithms are $O(n)$. All other algorithms presented in this paper can work out of these $O(n)$ standard inverse dynamics methods (e.g., the Newton–Euler method) (Featherstone 1987).

3.3.2. Dynamical Decoupling Controller Variation 1 (Without Null Space Pre-multiplication of **M***, and Compensation of* **C** *and* **g** *in Joint Space)*

In the control law presented in Section 3.3.1, Coriolis and gravitational terms are compensated only in operational space.

This formulation can be problematic, e.g., if gravitational forces in null space push the robot against its joint stops and/or cause collisions with other body parts. It is often more suitable to perform full gravity compensation of the robot, such that the null space term of the control law can properly achieve subordinate objectives, e.g., collision avoidance. Motivated by the discussions in Featherstone and Khatib (1997), we consider a variant of force-based control by pre-compensating Coriolis and gravitational terms in joint space:

$$\begin{aligned}\boldsymbol{\tau} &= \mathbf{C} + \mathbf{g} + \mathbf{J}^T\mathbf{F} + (\mathbf{I} - \mathbf{J}^T\bar{\mathbf{J}}^T)\boldsymbol{\tau}_{\text{null}} \\ &= \mathbf{M}\bar{\mathbf{J}}(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + \mathbf{C} + \mathbf{g} + (\mathbf{I} - \mathbf{J}^T\bar{\mathbf{J}}^T)\boldsymbol{\tau}_{\text{null}}, \quad (58)\end{aligned}$$

where

$$\begin{aligned}\mathbf{F} &= (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}), & (59)\\ \boldsymbol{\tau}_{\text{null}} &= -\mathbf{K}_{q,d}\dot{\mathbf{q}} - \alpha\nabla g. & (60)\end{aligned}$$

Again, a principled derivation of this control law from first-order principles of analytical mechanics can be found in Peters et al. (2005).

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

Task space:

$$\ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = 0. \quad (61)$$

Null space:

$$(\mathbf{I} - \mathbf{J}^T\bar{\mathbf{J}}^T)(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}_{q,d}\dot{\mathbf{q}} + \alpha\mathbf{K}_w(\mathbf{q} - \mathbf{q}_{\text{rest}})) = 0. \quad (62)$$

The task space error dynamics (61) suggest asymptotic tracking of the desired trajectory $\mathbf{x}_d$. However, the exact behavior of the null space dynamics cannot be determined easily, as a stability analysis of (62) seems to be rather difficult. This difficulty of understanding the null space stability properties is, however, a problem that is shared by all operational space controllers. So far, only empirical evaluations can help to assess the null space robustness.

3.3.3. Dynamical Decoupling Controller Variation 2 (With Null Space Pre-multiplication of M, and Compensation of C and g in Joint Space)

In (58) above, it is possible to choose the null space vector $\boldsymbol{\tau}_{\text{null}}$ as

$$\boldsymbol{\tau}_{\text{null}} = \mathbf{M}\ddot{\mathbf{q}}_0. \quad (63)$$

With this choice, the control law changes to

$$\begin{aligned}\boldsymbol{\tau} &= \mathbf{C} + \mathbf{g} + \mathbf{J}^T\mathbf{F} + (\mathbf{I} - \mathbf{J}^T\bar{\mathbf{J}}^T)\mathbf{M}\ddot{\mathbf{q}}_0 \\ &= \mathbf{C} + \mathbf{g} + \mathbf{J}^T\mathbf{F} + \mathbf{M}(\mathbf{I} - \bar{\mathbf{J}}\mathbf{J})\ddot{\mathbf{q}}_0 \\ &= \mathbf{M}\left(\bar{\mathbf{J}}(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}) + (\mathbf{I} - \bar{\mathbf{J}}\mathbf{J})\ddot{\mathbf{q}}_0\right) + \mathbf{C} + \mathbf{g}, \quad (64)\end{aligned}$$

where

$$\begin{aligned}\mathbf{F} &= (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}(\ddot{\mathbf{x}}_r - \dot{\mathbf{J}}\dot{\mathbf{q}}), & (65)\\ \ddot{\mathbf{q}}_0 &= -\mathbf{K}_{q,d}\dot{\mathbf{q}} - \alpha\nabla g. & (66)\end{aligned}$$

As already suggested in acceleration-based control, the topic of pre-multiplying the inertia matrix with the null space term can cause problems with the performance in face of modeling errors of **M**. With this control law, we have the following task space and null space closed loop dynamics, respectively:

Task space:

$$\ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = 0, \quad (67)$$

Null space:

$$(\mathbf{I} - \bar{\mathbf{J}}\mathbf{J})(\ddot{\mathbf{q}} + \mathbf{K}_{q,d}\dot{\mathbf{q}} + \alpha\mathbf{K}_w(\mathbf{q} - \mathbf{q}_{\text{rest}})) = 0. \quad (68)$$

The error dynamics (67) suggest asymptotic tracking of the desired trajectory $\mathbf{x}_d$, and we have a slightly simplified null space dynamics as compared to (62); however, it still remains difficult to analyze and conclude stability in null space.

4. Orientation Control with Quaternion Feedback

The enumeration of the controllers in the previous section did not make any attempts to be specific about the nature of the controlled task. In general, position and orientation control need to be considered in operational space (besides force control, which we will not address explicitly here). While operational space position control is rather straightforward, orientation control is more complex. An elegant and numerically robust solution to this problem can be formulated with the help of quaternions (Yuan 1988; Caccavale et al. 1998; Xian et al. 2004), instead of Euler angles or roll–pitch–yaw angles, which are frequently used in the robotics literature. Quaternions have desirable properties, e.g., (a) quaternion derivatives can be integrated over time to obtain the resultant orientation representation, (b) they do not have singularities, and (c) it is straightforward to convert quaternions and quaternion derivatives to other orientation representations.

In order to formulate task space control with quaternions, let us denote a unit quaternion as

$$\boldsymbol{\alpha} = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \eta, & \epsilon_1, & \epsilon_2, & \epsilon_3, \end{bmatrix}, \quad (69)$$

where η is the scalar part and $\boldsymbol{\epsilon}$ is the vector part. The unit quaternion needs to fulfill $\eta^2 + \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 = 1$. A spatial orientation can be described by a rotation of φ about a unit vector $\mathbf{r}$ ($\|\mathbf{r}\| = 1$) and represented in terms of a quaternion as

$$\begin{aligned}\eta &= \cos\left(\frac{\varphi}{2}\right), & (70)\\ \epsilon &= \mathbf{r}\sin\left(\frac{\varphi}{2}\right). & (71)\end{aligned}$$

For orientation control, in Yuan (1988), the orientation error is formulated using the unit quaternion as[3]

$$\mathbf{e}_o = \delta\boldsymbol{\epsilon} = \eta_d\boldsymbol{\epsilon} - \eta\boldsymbol{\epsilon}_d + [\boldsymbol{\epsilon}_d\times]\boldsymbol{\epsilon}, \qquad (72)$$

where

$$[\boldsymbol{\epsilon}_d\times] = \begin{bmatrix} 0 & -\epsilon_{3d} & \epsilon_{2d} \\ \epsilon_{3d} & 0 & -\epsilon_{1d} \\ -\epsilon_{2d} & \epsilon_{1d} & 0 \end{bmatrix}, \qquad (73)$$

$\boldsymbol{\alpha}_d = \begin{bmatrix} \eta_d, & \epsilon_{1d}, & \epsilon_{2d}, & \epsilon_{3d} \end{bmatrix}$ is the desired orientation and $\boldsymbol{\alpha} = \begin{bmatrix} \eta, & \epsilon_1, & \epsilon_2, & \epsilon_3 \end{bmatrix}$ is the current orientation.

In velocity-based control, given the desired task space translatory velocities $\dot{\mathbf{p}}_d$, positions $\mathbf{p}_d$, angular velocities $\boldsymbol{\omega}_d$, and orientation $\boldsymbol{\alpha}_d$ (i.e. as a quaternion), the task space translatory reference velocity $\dot{\mathbf{p}}_r$ and angular reference velocity $\boldsymbol{\omega}_r$ are defined, respectively, as

$$\begin{aligned} \dot{\mathbf{p}}_r &= \dot{\mathbf{p}}_d + \mathbf{K}_p(\mathbf{p}_d - \mathbf{p}), && (74) \\ \boldsymbol{\omega}_r &= \boldsymbol{\omega}_d - \mathbf{K}_o\mathbf{e}_o \\ &= \boldsymbol{\omega}_d - \mathbf{K}_o(\eta_d\boldsymbol{\epsilon} - \eta\boldsymbol{\epsilon}_d + [\boldsymbol{\epsilon}_d\times]\boldsymbol{\epsilon}), && (75) \end{aligned}$$

such that the augmented task space reference velocities are

$$\dot{\mathbf{x}}_r = \begin{bmatrix} \dot{\mathbf{p}}_r \\ \boldsymbol{\omega}_r \end{bmatrix}. \qquad (76)$$

With this re-definition, all the controllers in Section 3.1 can be applied.

In analogy, in acceleration- and force-based control, the task space translatory reference acceleration $\ddot{\mathbf{p}}_r$ and angular reference acceleration $\dot{\boldsymbol{\omega}}_r$ are defined, respectively, as

$$\begin{aligned} \ddot{\mathbf{p}}_r &= \ddot{\mathbf{p}}_d + \mathbf{K}_d(\dot{\mathbf{p}}_d - \dot{\mathbf{p}}) + \mathbf{K}_p(\mathbf{p}_d - \mathbf{p}), && (77) \\ \dot{\boldsymbol{\omega}}_r &= \dot{\boldsymbol{\omega}}_d + \mathbf{K}_a(\boldsymbol{\omega}_d - \boldsymbol{\omega}) - \mathbf{K}_o\mathbf{e}_o \\ &= \dot{\boldsymbol{\omega}}_d + \mathbf{K}_a(\boldsymbol{\omega}_d - \boldsymbol{\omega}) \\ &- \mathbf{K}_o(\eta_d\boldsymbol{\epsilon} - \eta\boldsymbol{\epsilon}_d + [\boldsymbol{\epsilon}_d\times]\boldsymbol{\epsilon}). && (78) \end{aligned}$$

The task space reference accelerations are augmented as

$$\ddot{\mathbf{x}}_r = \begin{bmatrix} \ddot{\mathbf{p}}_r \\ \dot{\boldsymbol{\omega}}_r \end{bmatrix}. \qquad (79)$$

Then, all the controllers in Sections 3.2 and 3.3 can be applied.

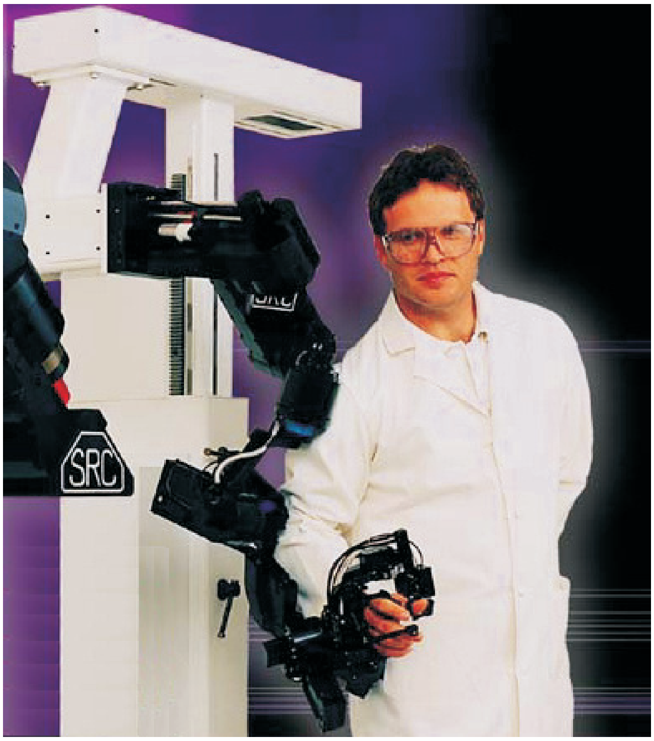


Fig. 1. A seven degree-of-freedom hydraulic robot, a Sarcos Master Arm, which was used in our experiments.

5. Estimation of Physically Consistent Rigid Body Parameters

For experimental evaluations, we used a Sarcos Master Arm, a seven DOF hydraulically actuated anthropomorphic robot arm (Figure 1). A local analog force control loop in every DOF realizes a very compliant and back-drivable torque control mode, exactly what we need to explore operational space controllers. In order to implement the controllers introduced in the above sections, it is necessary to obtain the rigid body dynamics model in (1), which requires knowledge of the inertial parameters of each robot link. Ideally, these parameters can be obtained from computer-aided design (CAD) data. However, due to the significant contribution of hydraulic actuation in our robot, CAD data turned out to be a poor fit to model the robot dynamics, such that we resorted to numerical parameter identification. While estimation techniques such as An et al. (1988) generate good inverse dynamics models for simple joint space controllers such as computed torque or classical inverse dynamics control, they run into problems with complex task space controllers, as investigated in this paper. The reason for these problems is due to the ordinary least-squares procedure employed for identifying the link parameters—this procedure has no mechanism to ensure physical correctness of the inertial parameters, i.e. positive mass parameters, PD iner-

[3] Note that there is a sign change in the third term in this orientation error in comparison with Yuan (1988). This modification is due to the difference of representing angular velocity in world and not local coordinates.

tia matrices, and the constraints imposed by the parallel axis theorem in converting center-of-mass inertia to joint-axis coordinate systems. As a result, it was possible to obtain a non-PD inertia matrix $\mathbf{M}(\mathbf{q})$ at certain configurations of the robot, which destabilized some of the suggested controllers. To deal with this problem, we derived a modified parameter estimation method that enforces physical consistency (see Appendix B).

6. Experimental Evaluations

6.1. Benchmark Tasks

We evaluated the performance and properties of the various controllers introduced in the above sections in actual robot implementations on the Sarcos Master Arm robot. All evaluations were initially debugged on a physical simulator of the robot that uses exactly the same control code as generated for the real robot—in numerical simulation, all controllers achieved the same excellent performance, which is thus not reported here. As benchmark tasks, the following nine experiments were selected after some experimentation, with the goal to provide interesting comparisons under different movement conditions:

- Tracking a “figure 8” pattern:

  1. figure 8, slow speed, high task space gain;
  2. figure 8, fast speed, high task space gain.

- Drawing a “star-like” pattern:

  3. star-like pattern, slow speed, high task space gain;
  4. star-like pattern, fast speed, high task space gain;
  5. star-like pattern, slow speed, low task space gain;
  6. star-like pattern, fast speed, low task space gain.

- Tracking a “figure 8” pattern with orientation control:

  7. figure 8 with orientation, slow speed, high task space gain;
  8. figure 8 with orientation, fast speed, high task space gain.

- Step response:

  9. move the end-effector horizontally with a step command, high task space gain.

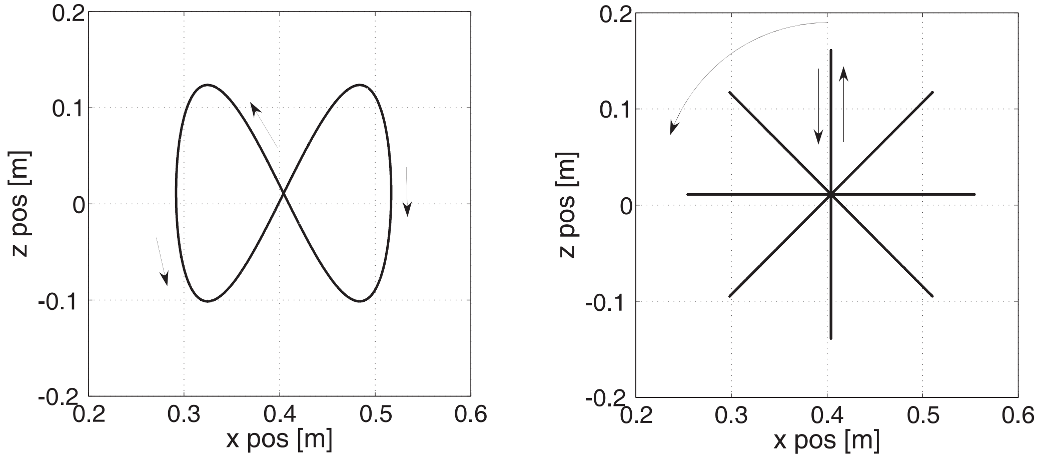
Figure 2 depicts the desired movement patterns. In experiments (1) and (2), the task is to track a planar “figure 8” pattern in the vertical plane (height and width: 0.225 m) in task space at two different speeds (slow speed 4 s and fast speed 2 s per cycle). In experiments (3)–(6), the task is to draw a “star-like” pattern in the vertical plane by first pointing outwards from the center and then inwards back to the center in eight directions in a sequential manner (slow 1 s and fast 0.5 s for each pointing movement). The desired pointing movement is specified by a minimum-jerk trajectory (Hogan 1984). This “star-like” pattern is often considered in human motor control experiments and has components of rather high acceleration, which challenges tracking accuracy. In experiments (7) and (8), in addition to the task of tracking the desired position trajectory, the end-effector’s orientation is also specified to keep a fixed desired pose. In experiment (9), we move the end-effector with a step command in the horizontal direction in order to examine the transient behavior of the controllers. The magnitude of the step command was 5 cm.

Some special consideration needs to be given to the gain settings of the controllers, as these gains are crucial for our empirical comparisons. Our robot has joint space PD gains that were tuned to give an approximate 200 ms rise time for a 0.03 rad step input under the least inertial loading of each DOF. This results in a moderately stiff PD controller, i.e. it is easy to push the robot’s end-effector 10–20 cm away from its setpoint. The joint space PD gains were used for the velocity-based controllers, and as null space gains in the acceleration-based controller without $\mathbf{M}$ pre-multiplication, the dynamic decoupling controller without $\mathbf{M}$ pre-multiplication, and the Gauss controller.

For null space gains with $\mathbf{M}$ pre-multiplication, we started with a uniform value of 50 as a gain for all DOFs, which resulted in very compliant null space motion. When we noticed during the experiments that some null space gains were too low, e.g., as manifested by hitting joint space limits; we increased the corresponding null space gains until the behavior of the robot was acceptable. Typically, the wrist DOFs required such higher gains as, otherwise, we would have excessive wrist motion. Pre-multiplication of the null space control term with $\mathbf{M}$ effectively changes the null space gains. For the chosen test tasks, however, the condition number of the inertia matrix was on average 40, such that the inertia matrix only rescaled the null space gains in a moderate way, and, in particular, did not lead to degenerate null space gains.

In general, we aimed at a uniform appearance of the behavior across all controllers as shown in the video extensions of this paper (Appendix A). Null space gains were tuned such that it was easy to push the robot in the null space significantly away from its nominal movement behavior, as also demonstrated in Extension 5. Such compliant behavior, as explained in the introduction, is really the goal of operational space control, and it also observed in human movement studies.

In order to find the performance limits, the task space gains were tuned until the first signs of instability appeared. Reducing these gains by 20% resulted in our “high task space gain setting”. Dividing these high task space gains by two, we obtained the “low task space gain setting”. Thus, in our experiments, for the high task space gain setting, we used $\mathbf{K}_p = 10\mathbf{I}$, $\mathbf{K}_o = 50\mathbf{I}$ for the velocity-based controllers, and



(a) Figure 8 target pattern

(b) Star-like target pattern

Fig. 2. Desired movement patterns.

$\mathbf{K}_p = 50\mathbf{I}$, $\mathbf{K}_o = 1000\mathbf{I}$, and $\mathbf{K}_d = \sqrt{50}\mathbf{I}$, $\mathbf{K}_a = \sqrt{1000}\mathbf{I}$ for the acceleration- and force-based controllers. For the low task space gain setting, we used $\mathbf{K}_p = 5\mathbf{I}$, $\mathbf{K}_o = 25\mathbf{I}$ for the velocity based controllers, and $\mathbf{K}_p = 25\mathbf{I}$, $\mathbf{K}_o = 500\mathbf{I}$, and $\mathbf{K}_d = \sqrt{25}\mathbf{I}$, $\mathbf{K}_a = \sqrt{500}\mathbf{I}$ for the acceleration- and force-based controllers[4].

We collected the data from three runs for each experiment for each the eight controllers described in Section 3.

It should be noted that in all experiments, we made sure that none of the actuators reached its torque, velocity, or acceleration limits. These limits have been determined empirically over years of experimentation with our robot hardware, and our control systems has specialized watchdogs to report any form of saturation effects.

6.2. Experimental Results

Figures 3–6 show examples of tracking results of experiments (1), (4), (6), and (9) with the goal of illustrating qualitative differences among the controllers. The gray line is the target trajectory and the solid line is the actual trajectory. Table 1 shows the root mean squared (RMS) errors between the actual and target trajectory (position), and Table 2 shows the RMS orientation error of experiments (7) and (8). As an orientation error measure, we chose the L_2 norm of the orientation error feedback term $\mathbf{e}_o$ in (72). In these tables, bold numbers indicate the best tracking performance and italic numbers indicate the second best tracking performance.

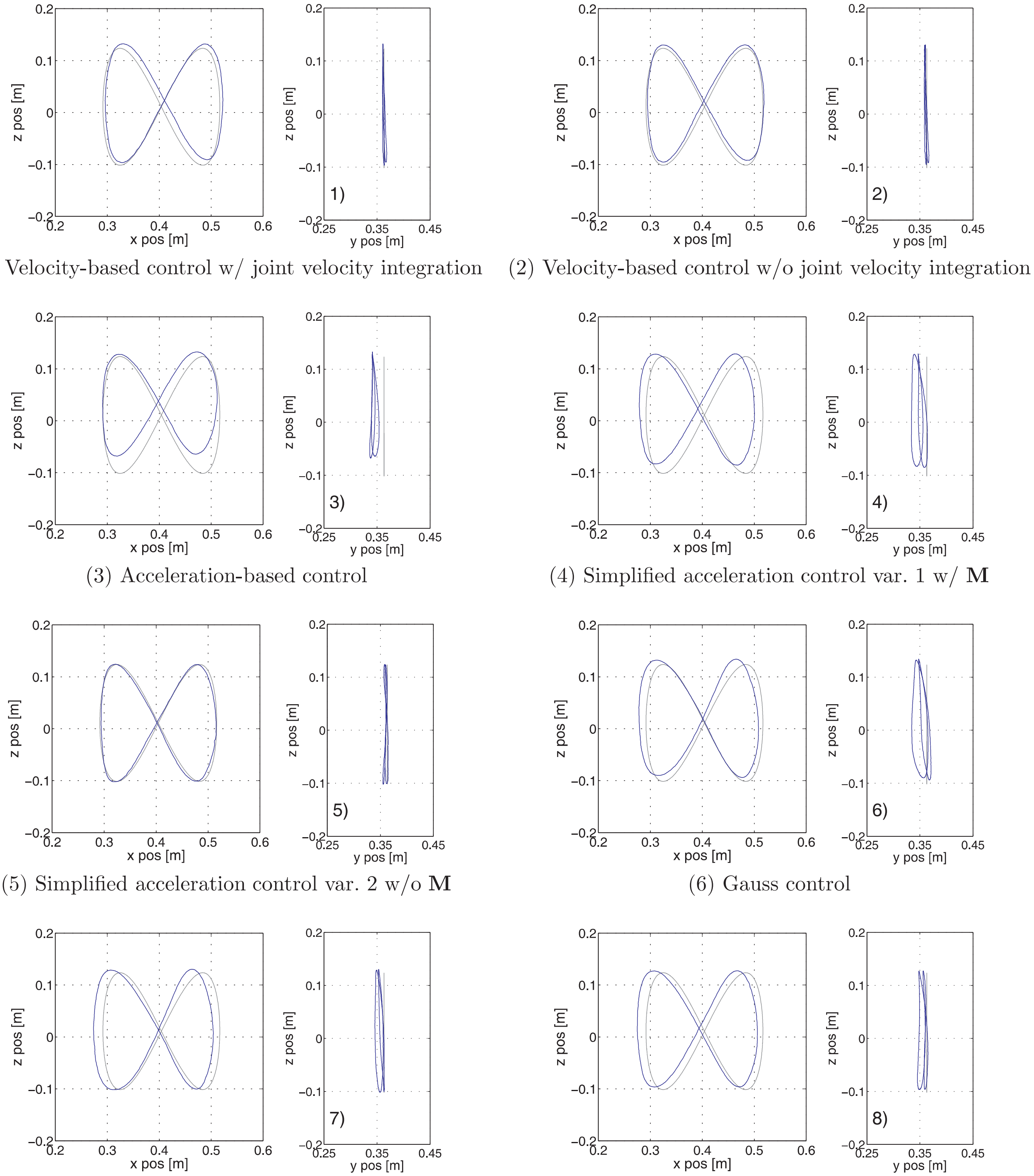
Video extensions 1–5 show examples of the movement of the robot with the simplified acceleration- based control (controller 5). Extensions 1 and 2 illustrate the movement in experiments (1) and (2) (“figure 8” pattern at slow and fast speeds, respectively). Extension 3 is from experiment (4), i.e. a star-like pattern with fast speed. Extension 4 was recorded from experiment (7) (“figure 8” with orientation control at slow speed) demonstrating the effectiveness of orientation control by placing a cup containing water on the end-effector. In addition, Extension 5 demonstrates the level of compliance and robustness of the control by manually perturbing the robot during the movement.

The experimental results can be summarized as follows:

- **Overall performance comparison:** Among all the controllers, the simplified acceleration-based control (controller 5) is the most promising approach in terms of task performance, ease of parameter tuning, and general robustness and compliance. The comprehensive experimental evaluation in this paper demonstrates the effectiveness of this approach in face of inevitable modeling errors. In particular, the results of experiment (6) (fast star-like pattern with low gain), which is the most demanding task of all, showed that the simplified acceleration-based control (controller 5) still achieves remarkably good tracking performance while other acceleration/force-based controllers had significantly degraded performance.

- **Velocity-based approach:** Velocity-based controllers (controllers 1 and 2) were straightforward to implement and achieved overall good performance. However, there

[4] In the acceleration-based controller (3) (Section 3.2.1), we had to reduce orientation gains because of instability, and $\mathbf{K}_o = 400\mathbf{I}$, $\mathbf{K}_a = \sqrt{400}\mathbf{I}$ were used in the experiments reported in this paper. This controller is harder to tune as there are more interwoven algebraic and derivative terms, which require one to find a tradeoff between filtering of numeric derivatives and choosing higher gain parameters.



(1) Velocity-based control w/ joint velocity integration

(2) Velocity-based control w/o joint velocity integration

(3) Acceleration-based control

(4) Simplified acceleration control var. 1 w/ $\mathbf{M}$

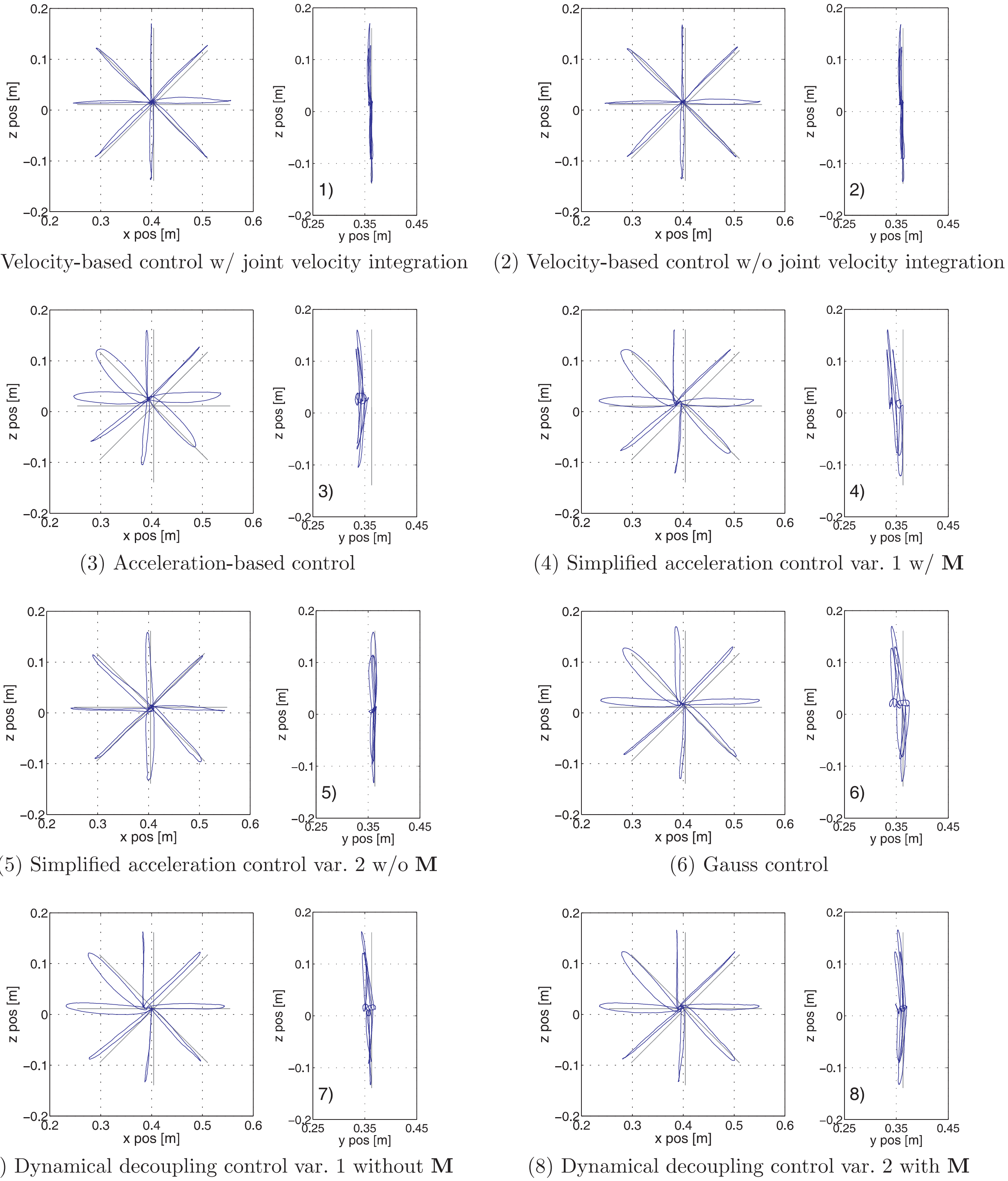
(5) Simplified acceleration control var. 2 w/o $\mathbf{M}$

(6) Gauss control

(7) Dynamical decoupling control var. 1 without $\mathbf{M}$

(8) Dynamical decoupling control var. 2 with $\mathbf{M}$

Fig. 3. Tracking results of a slow figure 8 movement (4 s per period). Target: gray line, actual: solid line. See also Extensions 1, 2, and 5.



(1) Velocity-based control w/ joint velocity integration

(2) Velocity-based control w/o joint velocity integration

(3) Acceleration-based control

(4) Simplified acceleration control var. 1 w/ $\mathbf{M}$

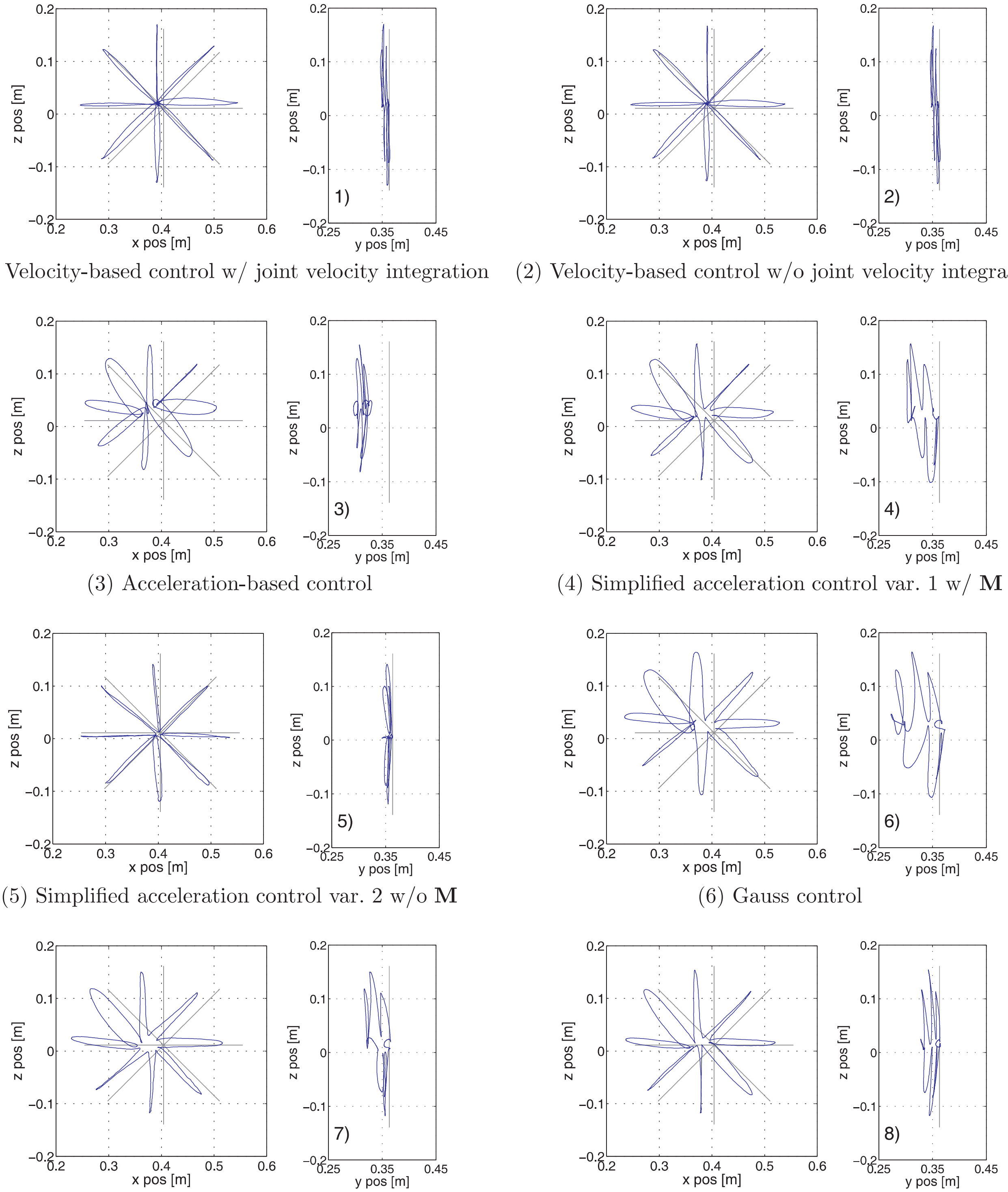
(5) Simplified acceleration control var. 2 w/o $\mathbf{M}$

(6) Gauss control

(7) Dynamical decoupling control var. 1 without $\mathbf{M}$

(8) Dynamical decoupling control var. 2 with $\mathbf{M}$

Fig. 4. Tracking results of fast star movement (0.5 s for each pointing movement) with high gains. Target: gray line, actual: solid line. See also Extension 3.



(1) Velocity-based control w/ joint velocity integration

(2) Velocity-based control w/o joint velocity integration

(3) Acceleration-based control

(4) Simplified acceleration control var. 1 w/ $\mathbf{M}$

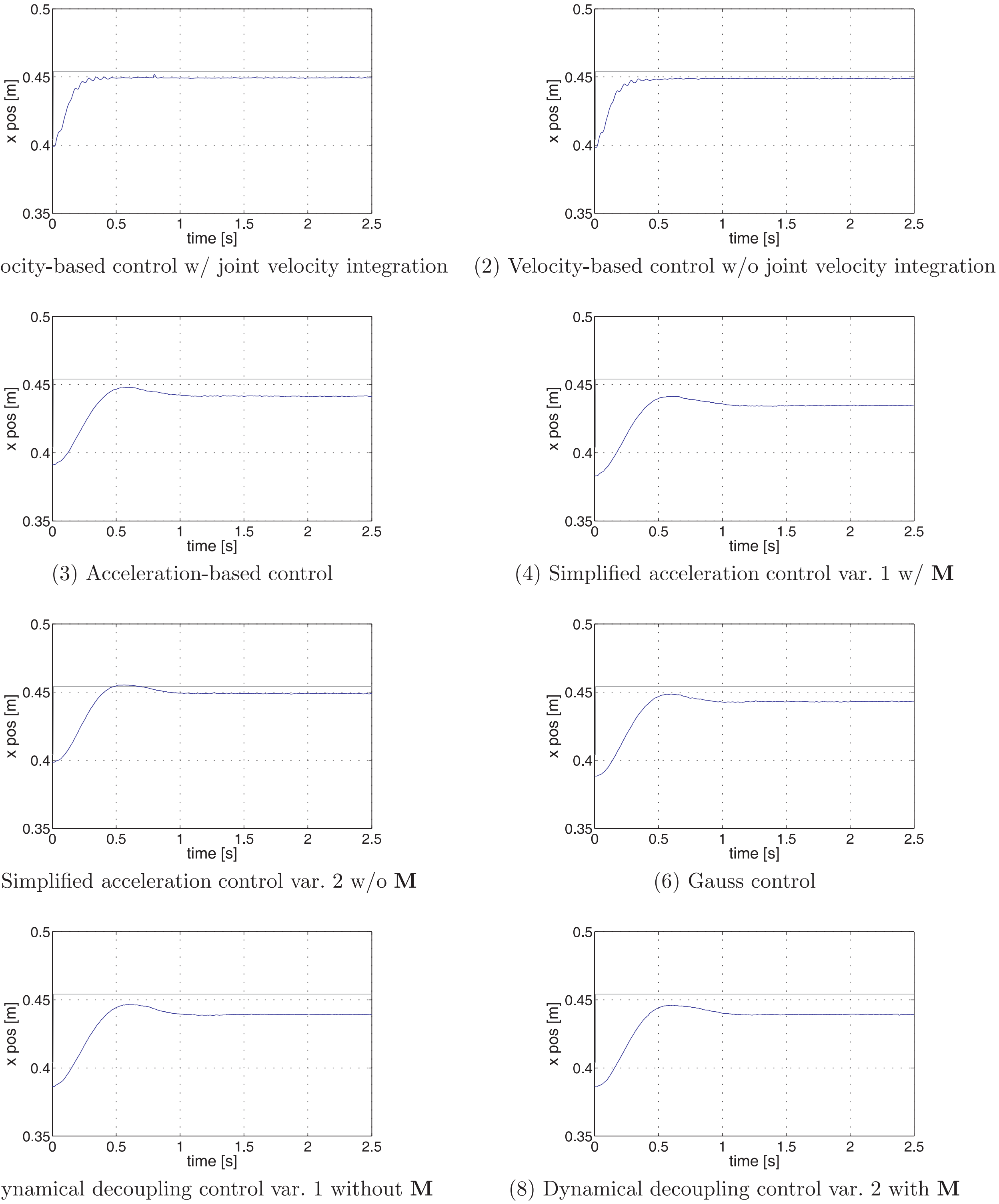
(5) Simplified acceleration control var. 2 w/o $\mathbf{M}$

(6) Gauss control

(7) Dynamical decoupling control var. 1 without $\mathbf{M}$

(8) Dynamical decoupling control var. 2 with $\mathbf{M}$

Fig. 5. Tracking results of fast star movement (0.5 s for each pointing movement) with low gains. Target: gray line, actual: solid line.



(1) Velocity-based control w/ joint velocity integration

(2) Velocity-based control w/o joint velocity integration

(3) Acceleration-based control

(4) Simplified acceleration control var. 1 w/ $\mathbf{M}$

(5) Simplified acceleration control var. 2 w/o $\mathbf{M}$

(6) Gauss control

(7) Dynamical decoupling control var. 1 without $\mathbf{M}$

(8) Dynamical decoupling control var. 2 with $\mathbf{M}$

Fig. 6. Results of step response. The step command was given to move the end-effector by 5 cm in the horizontal direction from the initial posture. Target: gray line, actual: solid line.

Table 1. Root mean squared (RMS) tracking error (m) of the benchmark tasks using different control laws averaged over three experimental runs. The bold numbers indicate the smallest tracking error and the italic numbers indicate the second smallest tracking error for each task, respectively.

	(1) Vel.	(2) Vel.	(3) Acc.	(4) Acc. 1	(5) Acc. 2	(6) Gauss	(7) Dyn.	(8) Dyn.
	w/integ-	w/o integ-		w/ **M**	w/o **M**		dec. 1	dec. 2
	ration	ration					w/o **M**	w/ **M**
Figure 8 (slow)	0.0100	*0.0090*	0.0312	0.0256	**0.0080**	0.0218	0.0191	0.0187
Figure 8 (fast)	**0.0144**	**0.0144**	0.0322	0.0317	*0.0258*	0.0293	0.0325	0.0291
Star (slow, high gain)	**0.0072**	0.0090	0.0323	0.0265	*0.0074*	0.0180	0.0178	0.0133
Star (fast, high gain)	*0.0117*	0.0125	0.0351	0.0309	**0.0144**	0.0240	0.0246	0.0221
Star (slow, low gain)	**0.0153**	0.0188	0.0662	0.0451	*0.0155*	0.0570	0.0405	0.0343
Star (fast, low gain)	*0.0209*	0.0223	0.0685	0.0518	**0.0197**	0.0600	0.0439	0.0396
Figure 8 w/ orient (slow)	**0.0087**	**0.0087**	0.0216	0.0177	*0.0166*	0.0194	0.0188	0.0251
Figure 8 w/ orient (fast)	*0.0160*	**0.0155**	0.0314	0.0390	0.0380	0.0396	0.0405	0.0387
Step response	**0.0096**	*0.0105*	0.0317	0.0292	0.0123	0.0201	0.0198	0.0198

Table 2. Root mean squared (RMS) orientation error of the figure 8 task with orientation control using different control laws averaged over three experimental runs. As an orientation error measure, the L_2 norm of the orientation error feedback term $e_o = \eta_d \epsilon - \eta \epsilon_d + [\epsilon_d \times]\epsilon$ in (72) is considered. The bold numbers indicate the smallest tracking error and the italic numbers indicate the second smallest tracking error for each task, respectively.

	(1) Vel.	(2) Vel.	(3) Acc.	(4) Acc. 1	(5) Acc. 2	(6) Gauss	(7) Dyn.	(8) Dyn.
	w/integ-	w/o integ-		w/ **M**	w/o **M**		dec. 1	dec. 2
	ration	ration					w/o **M**	w/ **M**
Figure 8 w/ orient (slow)	**0.0219**	*0.0240*	0.0751	0.0296	0.0260	0.0311	0.0307	0.0274
Figure 8 w/ orient (fast)	**0.0259**	*0.0299*	0.0814	0.0428	0.0429	0.0455	0.0443	0.0393

is a practical limitation in the choice of the task space position gain because the task space position gain effectively appears as a task space damping gain due to numerical differentiation of the joint space reference velocity (11). This effect implies that (a) we cannot increase position gains too much to improve the tracking performance, as it may lead to instability due to a too large damping gain and noise in the velocity measurements, and (b) the impedance behavior of the task space dynamics cannot be easily specified because of this coupling of task space position and damping gains. In fact, we found that increasing the task space position gain lead to instability. From a subjective point of view, all velocity-based controllers “felt” highly overdamped and not very compliant when manually perturbing the robot. These effects can be observed in the results of the step response experiment (experiment (9)). Some small oscillations are observed in the transient of the velocity-based controllers (controllers 1 and 2) in Figure 6 while acceleration- and force-based controllers achieve smooth transients. Also, the transient behavior of velocity-based controllers looks like that of a first-order system which implies overdamped response against external perturbations.

- **Acceleration-based controller in Hsu et al. (1989) (controller 3):** The experimental results demonstrated that we could not achieve good performance and we found that the controller was not very robust and required significant tuning effort. For orientation control, the choice of the orientation gain of $\mathbf{K}_o = 1000\mathbf{I}$ resulted in instability, while the same choice of the orientation gain worked well in other acceleration- and force-based controllers. Also, this controller was highly sensitive to noise from numerical differentiation, which was generated when obtaining time derivatives of Jacobian and pseudo-inverse matrices. Moreover, we noticed a significant problem in attempting to stabilize low inertia

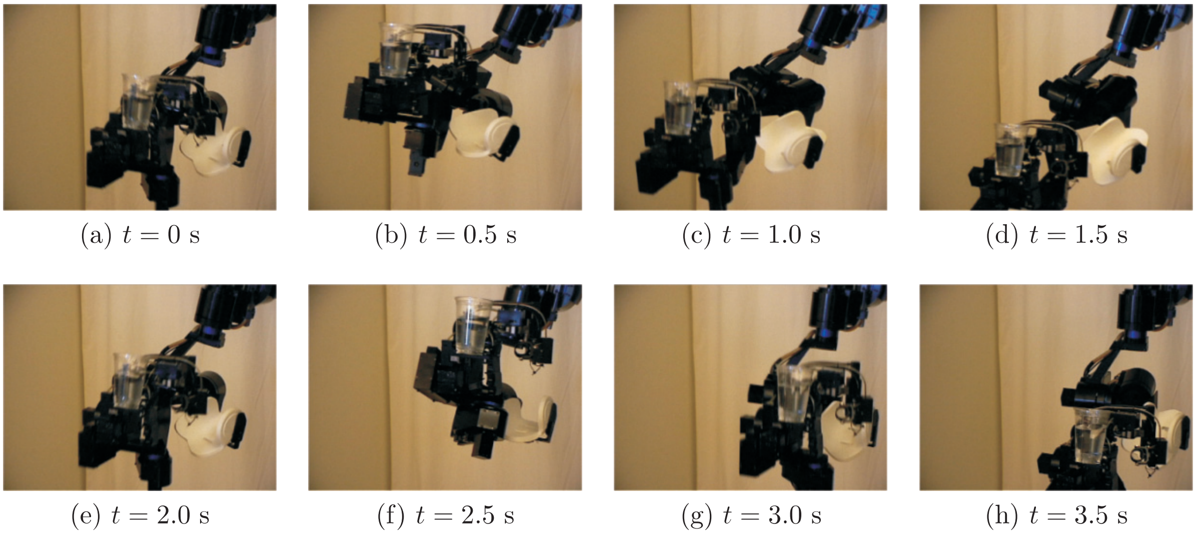

(a) $t = 0$ s (b) $t = 0.5$ s (c) $t = 1.0$ s (d) $t = 1.5$ s
(e) $t = 2.0$ s (f) $t = 2.5$ s (g) $t = 3.0$ s (h) $t = 3.5$ s

Fig. 7. Snapshots of a slow figure 8 movement with orientation control. The task is to track the figure 8 pattern by keeping the end-effector's posture fixed. In order to demonstrate orientation control, a cup filled with water was placed on the end-effector. See also Extension 4 for a video.

DOFs such as the wrist. We believe that the difficulties experienced with this controller are a result of the complex formulation of its null space terms. While theoretically justified, the additional complexity of this controller (as compared with the other acceleration-based controllers presented in this paper), makes tuning to achieve desired performance significantly more challenging. Largely, this issues results from the tradeoff of gain parameters and the amount of filtering that needs to be performed on numerical derivatives for Jacobians.

- **Force-based control with inertia-weighted pseudo-inverse:** Force-based control approaches using the inertia-weighted pseudo-inverse have the desirable property of dynamical decoupling between task and null space. However, in practice, performance of these force-based controllers (controllers 6–8) was not as good as the simplified acceleration-based control (controller 5). This effect is in particular due to inaccuracies of the estimated inertia matrix, as this matrix and its inverse are used in many different places of the control law. Computationally, the calculation of the inertia-weighted pseudo-inverse requires explicit extraction of the inertia matrix from the rigid body dynamics, which is computationally expensive (Featherstone 1987).

- **Explicit use of the inertia matrix:** The performance of the algorithms which explicitly use the inertia matrix (force-based algorithms with inertia-weighted pseudo-inverse (controllers 6–8) or controllers including a null space term that is pre-multiplied by the inertia matrix (controllers 3, 4, and 8)) significantly degrade, especially in the tasks with fast movements. This implies that these algorithms require highly accurate inertia matrix estimation to be successful, as we confirmed in simulation studies (not presented here).

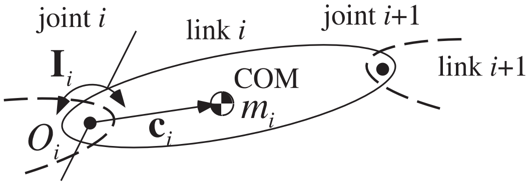


Fig. 8. Link inertial parameters and link coordinates for parameter identification. m is the link mass, $\mathbf{I}$ is the inertia matrix about the local link (joint-axis) coordinates, and $\mathbf{c}$ is the center-of-mass (COM) location in the joint-axis coordinates.

- **Orientation control with quaternion feedback:** The quaternion-based orientation control was successfully implemented for all controllers. Except for the acceleration-based controller (controller 3), we achieved good overall performance in the regulation of the desired posture of the end-effector while tracking the figure 8 pattern. We could place a cup containing water on the end-effector without spilling water in the figure 8 task with orientation control, by keeping the end-effector's posture fixed while the end-effector was tracking the figure 8 pattern (see Figure 7 and Extension 4). However, we observed that the task space position tracking performance was not as good as that of controllers without orientation constraints. We attribute this slightly worse tracking performance to the reduced amount of redundancy that can be exploited by feedback control to compensate for modeling errors.

- **Transient behavior in the step response:** As already mentioned above, velocity-based controllers achieved good tracking performance with small steady-state tracking error. However, we observed some undesirable oscillations in the transient of the step response and highly overdamped response against external perturbations. Figure 6 shows that acceleration and force controllers have smooth transients with a typical profile for a second- order system. The simplified acceleration-based control (controller 5) achieved the smallest steady-state tracking error.

7. Conclusion

This paper was motivated by the need to bring dexterous and compliant control to complex and highly redundant robots such as humanoid robots, entertainment robots, assistive robots, etc. The hope is that such a compliant mode of control achieves higher levels of tolerance towards unforeseen disturbances in dynamic—e.g., human—environments, and that safe operation in human environments can be achieved more easily. We focused on operational space control approaches, as this general framework offers to generate maximal compliance in task-redundant DOFs while only constraining the task-relevant DOFs. As there seem to be a lack of application of modern operational control in complex redundant robots, we set out to conduct an extensive theoretical and empirical evaluation of existing approaches in the literature and some new variants of operational space controller. We formulated all the controllers in a unified notational framework, including quaternion-based orientation control. We also introduced a parameter estimation algorithm for rigid body dynamics that ensures physically consistent parameter identification, a crucial component in the proper working of some of the examined controllers. All controllers were evaluated in on several movement tasks with a 7 DOF compliant Sarcos Master Arm robot.

Several theoretical and experimental insights were obtained. First, the most commonly applied velocity-based operational space controllers are severely limited in the impedance characteristics they can achieve. Nevertheless, they are easy to implement, and achieve good tracking performance, but it is just that they are not very suitable for high-performance compliant control.

Second, the theoretically most advanced force-based operational space controllers exhibit a significant dependence on highly accurate model identification of the robot. For instance, the appealing operational space control approach by Khatib (1987) degrades rapidly in the face of modeling errors and scored surprisingly low in our experimental evaluations (while working perfectly in simulations). One of the critical components responsible for this degradation is the rigid body dynamics inertia matrix, that, when inaccurate, creates many sources of errors in the motor commands due to its appearance in pseudo-inverses and pre-multiplication of negative feedback terms. As an aside, these force-based operational controllers are rather expensive to compute, with computational costs at least reaching $O(n^2)$ in the number of DOFs n.

Third, acceleration-based operational space controllers fared surprisingly well in our evaluations, particularly after cautiously avoiding the null space control terms with the rigid body inertia matrix. A simplified acceleration-based controller (controller 5) had the overall best performance in terms of tracking results and general robustness. It even worked surprisingly well for the demanding task of the fast star-like movement with low task space gain settings, in which the performance of many other controllers significantly degraded. This simplified acceleration-based controller is easy to implement in the framework of efficient Newton–Euler rigid body dynamics formulations ($O(n)$ in computational complexity).

From a theoretical point of view, it needs to be noticed that none of the operational space controllers of this paper has fully understood stability properties—stability can usually only be demonstrated in the task space, while the null space dynamics so far resist insightful general analytical investigations (Hsu et al. (1989) provided a first step to such an analysis). An interesting recent investigation by Peters et al. (2005) demonstrated that many operational space controllers can be derived from a unified optimal control framework employing a generalized Gauss' principle with different metrics for a cost function. If stability could be proven for this family of operational space controllers, operational space control would be lifted to a more solid foundation.

We hope that this paper will provide a rather comprehensive overview of the state-of-the-art in operational space control, and offer the practitioner in robotics a well-discussed set of choices on how to approach compliant control in redundant robots. Our future work will investigate these controllers in the context of compliant biped locomotion, manipulation tasks with force perturbations, and general humanoid robotics. We will also address learning algorithms for operational space control as well as behavioral studies to identify the principles of redundancy resolution in humans.

Appendix A: Index to Multimedia Extensions

The multimedia extensions to this article can be found online by following the hyperlinks from www.ijrr.org.

Table 3. Multimedia extensions.

Extension	Type	Description
1	Video	Experiment (1) ("figure 8" movement at slow speed) with the controller 5
2	Video	Experiment (2) ("figure 8" movement at fast speed) with the controller 5
3	Video	Experiment (4) (star-like pattern at fast speed with high task space gain) with the controller 5
4	Video	Experiment (7) ("figure 8" movement with orientation control at slow speed) with the controller 5. A cup containing water is placed on the end-effector to demonstrate the effectiveness of orientation control
5	Video	Manually applying perturbations to the robot during the motion to show the level of compliance and robustness of the control using the controller 5)

Appendix B: Estimation of Physically Consistent Rigid Body Dynamics Parameters

In order to ensure physically consistent inertial parameters, we derived the following non-linear parameter estimation method using a re-parametrization of the basic link parameters. There are 11 parameters (10 inertial and one friction parameters) to be estimated for each DOF (see Figure 8), mass m, three center-of- mass coefficients multiplied by the mass mc_x, mc_y, mc_z, six inertial parameters (in joint axis, not center-of-mass coordinates) $I_{xx}, I_{xy}, I_{xz}, I_{yy}, I_{yz}, I_{zz}$ (cf. An et al. (1988)), and viscous friction d. These parameters are arranged in an 11-dimensional vector $\boldsymbol{\theta}$ as

$$\begin{aligned} \boldsymbol{\theta} &= [m, mc_x, mc_y, mc_z, \\ &\quad I_{xx}, I_{xy}, I_{xz}, I_{yy}, I_{yz}, I_{zz}, d]^T. \end{aligned} \tag{80}$$

To ensure physical consistency, the constraints in (81)–(89) need to be satisfied for each DOF, which are derived from the parallel axis theorem, a Cholesky decomposition of the inertia matrix (PD inertia matrix around the center of mass), and the need for positive mass and friction parameters. Thus, one can conceive that the original parameter vector $\boldsymbol{\theta}$ was generated through a non-linear transformation from an 11-dimensional virtual parameter vector $\tilde{\boldsymbol{\theta}} = [\tilde{\theta}_1, \ldots, \tilde{\theta}_{11}]^T$:

$$\theta_1 = \tilde{\theta}_1^2, \tag{81}$$

$$\theta_2 = \tilde{\theta}_1^2 \tilde{\theta}_2, \quad \theta_3 = \tilde{\theta}_1^2 \tilde{\theta}_3, \quad \theta_4 = \tilde{\theta}_1^2 \tilde{\theta}_4, \tag{82}$$

$$\theta_5 = \tilde{\theta}_5^2 + \tilde{\theta}_1^2 \left(\tilde{\theta}_3^2 + \tilde{\theta}_4^2 \right), \tag{83}$$

$$\theta_6 = \tilde{\theta}_5 \tilde{\theta}_6 - \tilde{\theta}_1^2 \tilde{\theta}_2 \tilde{\theta}_3, \tag{84}$$

$$\theta_7 = \tilde{\theta}_5 \tilde{\theta}_7 - \tilde{\theta}_1^2 \tilde{\theta}_2 \tilde{\theta}_4, \tag{85}$$

$$\theta_8 = \tilde{\theta}_6^2 + \tilde{\theta}_8^2 + \tilde{\theta}_1^2 \left(\tilde{\theta}_2^2 + \tilde{\theta}_4^2 \right), \tag{86}$$

$$\theta_9 = \tilde{\theta}_6 \tilde{\theta}_7 + \tilde{\theta}_8 \tilde{\theta}_9 - \tilde{\theta}_1^2 \tilde{\theta}_3 \tilde{\theta}_4, \tag{87}$$

$$\theta_{10} = \tilde{\theta}_7^2 + \tilde{\theta}_9^2 + \tilde{\theta}_{10}^2 + \tilde{\theta}_1^2 \left(\tilde{\theta}_2^2 + \tilde{\theta}_3^2 \right), \tag{88}$$

$$\theta_{11} = \tilde{\theta}_{11}^2. \tag{89}$$

Given the above formulation, any arbitrary set of virtual parameters gives rise to a physically consistent set of actual parameters for the rigid body dynamics model. In an ideal rigid body dynamics scenario, only the virtual parameters that correspond to the correct physical parameters will optimize the robot performance. For a robotic system with n DOFs, Equations (81)–(89) are repeated for each DOF. For parameter estimation, we replaced the least-squares approach of An et al. (1988) by a two-stage procedure. First, the unconstrained parameters are estimated as in classical methods. Second, the parameters are projected onto the constraint space. As shown in Ting et al. (2006), this projection step is a convex optimization problem with a unique optimum. A more advanced parameter identification procedure using a Bayesian approach was also presented in Ting et al. (2006), but its discussion would leave the scope of this paper.

We obtained the estimates of the model parameters from 40 min of robot data recorded in response to sufficiently exciting desired trajectories of pseudorandom motor commands (including sine waves of various frequencies at the joint level and discrete endpoint movements at various speeds).

Acknowledgments

We would like to thank Prof. Roy Featherstone for discussions and clarifications of the theoretical issues on redundancy resolution. This research was supported in part by National Science Foundation grants ECS-0325383, IIS-0312802, IIS-0082995, ECS-0326095, ANI-0224419, the DARPA program on Learning Locomotion, a NASA grant AC#98-516, an AFOSR grant on Intelligent Control, the ERATO Kawato Dynamic Brain Project funded by the Japan Science and Technology Agency, and the ATR Computational Neuroscience Laboratories.

Reference

Ahmed, S. (1992). Issues in repeatability of redundant manipulator control. *Ph.D. thesis*, The Ohio State University.

An, C. H., Atkeson, C. G. and Hollerbach, J. M. (1988). *Model-based Control of a Robot Manipulator*. MIT Press.

Arimoto, S. et al. (2005). Natural resolution of ill-posedness of inverse kinematics for redundant robots: a challenge to Bernstein's degrees-of-freedom problem. *Advanced Robotics*, **19**(4): 401–434.

Baillieul, J. and Martin, D. P. (1990). Resolution of kinematic redundancy. *Proceedings of Symposia in Applied Mathematics*, **41**: 49–89.

Bruyninckx, H. and Khatib, O. (2000). Gauss' principle and the dynamics of redundant and constrained manipulators. *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2563–2568.

Caccavale, F. et al. (1998). Resolved-acceleration control of robot manipulators: A critical review with experiments. *Robotica*, **16**(5): 565–573.

Cruse, H. et al. (1990). On the cost functions for the control of the human arm movement. *Biological Cybernetics*, **62**(6): 519–28.

De Luca, A. and Oriolo, G. (1991). Issues in acceleration resolution of robot redundancy. *Third IFAC Symposium on Robot Control*, pp. 93–98.

De Luca, A., Oriolo, G. and Siciliano, B. (1992). Robot redundancy resolution at the acceleration level. *Laboratory Robotics and Automation*, **4**, 97–106.

Doty, K. L., Melchiorri, C. and Bonivento, C. (1993). A theory of generalized inverses applied to robotics. *International Journal of Robotics Research*, **12**(1): 1–19.

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

Featherstone, R. (1987). *Robot Dynamics Algorithms*. Dordrecht, Kluwer Academic Publishers.

Featherstone, R. and Khatib, O. (1997). Load independence of the dynamically consistent inverse of the Jacobian matrix. *International Journal of Robotics Research*, **16**(2): 168–170.

Hogan, N. (1984). An organizing principle for a class of voluntary movements. *Journal of Neuroscience*, **4**(11): 2745–2754.

Hollerbach, J. M. and Suh, K. C. (1987). Redundancy resolution of manipulators through torque optimization. *IEEE Journal of Robotics and Automation*, **RA-3**(4): 308–316.

Hsu, P., Hauser, J. and Sastry, S. (1989). Dynamic control of redundant manipulators. *Journal of Robotic Systems*, **6**(2): 133–148.

Kazerounian, K. and Wang, Z. (1988). Global versus local optimization in redundancy resolution of robotic manipulators. *International Journal of Robotics Research*, **7**(5): 3–12.

Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation*, **RA-3**(1): 43–53.

Khatib, O. et al. (2004). 'Human-centered robotics and interactive haptic simulation. *International Journal of Robotics Research*, **23**(2): 167–478.

Liégeois, A. (1977). Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man and Cybernetics*, **7**(12): 868–871.

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

Maciejewski, A. A. and Klein, C. A. (1985), Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. *International Journal of Robotics Research*, **4**(3): 109–117.

Martin, D. P., Baillieul, J. and Hollerbach, J. M. (1989). Resolution of kinematic redundancy using optimization techniques. *IEEE Transactions on Robotics and Automation*, **5**(4): 529–533.

Mistry, M., Mohajerian, P. and Schaal, S. (2005). Arm movement experiments with joint space force fields using an exoskeleton robot. *IEEE Ninth International Conference on Rehabilitation Robotics*, pp. 408–413.

Mussa-Ivaldi, F. A. and Hogan, N. (1991). Integrable solutions of kinematic redundancy via impedance control. *International Journal of Robotics Research*, **10**(5): 481–491.

Nakamura, Y. and Hanafusa, H. (1986). Inverse kinematic solutions with singularity robustness for redundant manipulator control. *ASME Journal of Dynamic Systems, Measurement, and Control*, **108**, 163–171.

Nakamura, Y. and Hanafusa, H. (1987). Optimal redundancy control of robot manipulators. *International Journal of Robotics Research*, **6**(1): 32–42.

Nakamura, Y., Hanafusa, H. and Yoshikawa, T. (1987). Task-priority based redundancy control of robot manipulators. *International Journal of Robotics Research*, **6**(2): 3–15.

Peters, J. et al. (2005). A unifying methodology for the control of robotic systems. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3522–3529.

Saltzman, E. and Kelso, S. J. A. (1987). Skilled actions: a task-dynamic approach. *Psychological Review*, **94**(1): 84–106.

Schaal, S. (1997). Learning from demonstration. *Advances in Neural Information Processing Systems*, Vol. 9, Mozer, M. C., Jordan, M. and Petsche, T. (eds). MIT Press, pp. 1040–1046.

Schaal, S. and Schweighofer, N. (2005). Computational motor control in humans and robots. *Current Opinion in Neurobiology*, **15**(6): 675–682.

Scholz, J. P. and Schoner, G. (1999). The uncontrolled manifold concept: identifying control variables for a functional task. *Experimental Brain Research*, **126**(3): 289–306.

Sciavicco, L. and Siciliano, B. (1988). A solution algorithm of the inverse kinematic problem for redundant manipulators. *IEEE International Journal of Robotics and Automation*, **4**(4): 403–410.

Senda, K. (1999). Quasioptimal control of space redundant manipulators. *AIAA Guidance, Navigation, and Control Conference*, pp. 1877–1885.

Sentis, L. and Khatib, O. (2005). Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, **2**(4): 505–518.

Suh, K. C. and Hollerbach, J. M. (1987). Local versus global torque optimization of redundant manipulators. *Proceedings of IEEE/RSJ International Conference on Robotics and Automation*, pp. 619–624.

Tevatia, G. and Schaal, S. (2000). Inverse kinematics for humanoid robots. *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 294–299.

Ting, J.-A. et al. (2006). A Bayesian approach to nonlinear parameter identification for rigid body dynamics. *Robotics: Science and Systems Conference*.

Todorov, E. and Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, **5**(11): 1226–1235.

Udwadia, F. E. and Kalaba, R. E. (1996). *Analytical Dynamics: A New Approach*. Cambridge, Cambridge University Press.

Wampler, C. W. (1986). Manipulator inverse kinematic solutions based on damped least-squares solutions. *IEEE Transactions on Systems, Man and Cybernetics*, **SMC-16**, 91–101.

Wampler, C. W. and Leifer, L. J. (1988). Applications of damped least-squares methods to resolved-rate and resolved-acceleration control of manipulators. *ASME Journal of Dynamic Systems, Measurement, and Control*, **110**, 31–38.

Whitney, D. E. (1969). Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man-Machine Systems*, **10**(2): 47–53.

Xian, B. et al. (2004). Task-space tracking control of robot manipulators via quaternion feedback. *IEEE Transactions on Robotics and Automation*, **20**(1): 160–167.

Yoshikawa, T. (1984). Analysis and control of robot manipulators with redundancy. *Robotics Research: The First International Symposium*, Brady, M. and Paul, R. (eds). MIT Press, pp. 735–747.

Yoshikawa, T. (1985). Manipulability of robotic mechanisms. *International Journal of Robotics Research*, **4**(2): 3–9.

Yuan, J. S.-C. (1988). Closed-loop manipulator control using quaternion feedback. *IEEE Journal of Robotics and Automation*, **4**(4): 434–440.