Auton Robot (2008) 24: 1–12
DOI 10.1007/s10514-007-9051-x

A unifying framework for robot control with redundant DOFs

Jan Peters · Michael Mistry · Firdaus Udwadia · Jun Nakanishi · Stefan Schaal

Received: 19 October 2006 / Accepted: 5 September 2007 / Published online: 16 October 2007

Abstract Recently, Udwadia (Proc. R. Soc. Lond. A 2003:1783–1800, 2003) suggested to derive tracking controllers for mechanical systems with redundant degrees-of-freedom (DOFs) using a generalization of Gauss' principle of least constraint. This method allows reformulating control problems as a special class of optimal controllers. In this paper, we take this line of reasoning one step further and demonstrate that several well-known and also novel nonlinear robot control laws can be derived from this generic methodology. We show experimental verifications on a Sarcos Master Arm robot for some of the derived controllers. The suggested approach offers a promising unification and simplification of nonlinear control law design for robots obeying rigid body dynamics equations, both with or without external constraints, with over-actuation or underactuation, as well as open-chain and closed-chain kinematics.

Keywords Non-linear control · Robot control · Tracking control · Gauss' principle · Constrained mechanics · Optimal control · Kinematic redundancy

J. Peters (✉) · M. Mistry · F. Udwadia · S. Schaal
University of Southern California, Los Angeles, CA 90089, USA
e-mail: jrpeters@usc.edu

J. Peters
Department Schölkopf, Max-Planck Institute for Biological Cybernetics, Spemannstraße 38, Rm 223, 72076 Tübingen, Germany

J. Nakanishi · S. Schaal
ATR Computational Neuroscience Laboratories, Kyoto 619-0288, Japan

J. Nakanishi
ICORP, Japan Science and Technology Agency, Kyoto 619-0288, Japan

1 Introduction

The literature on robot control with redundant degrees-of-freedom (DOFs) has introduced many different approaches of how to resolve kinematic redundancy in complex robots and how to combine redundancy resolution with appropriate control methods (e.g., see Nakanishi et al. 2005 for an overview). For instance, methods can be classified to operate out of velocity-based, acceleration-based, and force-based principles, they can focus on local or global redundancy resolution strategies (Baillieul and Martin 1990), and they can have a variety of approaches how to include optimization criteria to maintain control in the null space of a movement task. When studying the different techniques, it sometimes appears that they were created from ingenious insights of the original researchers, but that there is also a missing thread that links different techniques to a common basic principle.

Recently, Udwadia (2003) suggested a new interpretation of constrained mechanics in terms of tracking control problem, which was inspired by results from analytical dynamics with constrained motion. The major insight is that tracking control can be reformulated in terms of constraints, which in turn allows the application of a generalization of Gauss' principle of least constraint[1] (Udwadia and Kalaba 1996; Bruyninckx and Khatib 2000) to derive a control law. This insight leads to a specialized point-wise optimal control framework for controlled mechanical systems. While it is not applicable to non-mechanical control problems with arbitrary cost functions, it yields an important class of opti-

[1]Gauss' principle least constraint (Udwadia and Kalaba 1996) is a general axiom on the mechanics of constrained motions. It states that if a mechanical system is constrained by another mechanical structure the resulting acceleration $\ddot{\mathbf{x}}$ of the system will be such that it minimizes $(\ddot{\mathbf{x}} - \mathbf{M}^{-1}\mathbf{F})^T\mathbf{M}^{-1}(\ddot{\mathbf{x}} - \mathbf{M}^{-1}\mathbf{F})$ while fulfilling the constraint.

mal controllers, i.e., the class where the problem requires task achievement under minimal squared motor commands with respect to a specified metric. In this paper, we develop this line of thinking one step further and show that it can be used as a general way of deriving robot controllers for systems with redundant DOFs, which offers a useful unification of several approaches in the literature. We discuss the necessary conditions for the stability of the controller in task space if the system can be modeled with sufficient precision and the chosen metric is appropriate. For assuring stability in configuration space further considerations may apply. To exemplify the feasibility of our framework and to demonstrate the effects of the weighting metric, we evaluate some of the derived controllers on an end-effector tracking task with an anthropomorphic robot arm.

This paper is organized as follows: firstly, a optimal control framework based on (Udwadia 2003) is presented and analyzed. Secondly, we discuss different robot control problems in this framework including joint and task space tracking, force and hybrid control. We show how both established and novel controllers can be derived in a unified way. Finally, we evaluate some of these controllers on a Sarcos Master Arm robot.

2 A unifying methodology for robot control

A variety of robot control problems can be motivated by the desire to achieve a task accurately while minimizing the squared motor commands, e.g., we intend to track a trajectory with minimum generated torques. Such problems can be formalized as a type of minimum effort control. In this section, we will show how the robot dynamics and the control problem can be brought into a general form which, subsequently, will allow us to compute the optimal control commands with respect to a desired metric. We will augment this framework such that we can ensure the necessary conditions for stability both in the joint space of the robot as well as in the task space of the problem.

2.1 Formulating robot control problems

We assume the well-known rigid-body dynamics model of manipulator robotics with n degrees of freedom given by the equation

$$\mathbf{u} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}), \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^n$ is the vector of motor commands (i.e., torques or forces), $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n$ are the vectors of joint position, velocities and acceleration, respectively, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n\times n}$ is the mass or inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$ denotes centrifugal and Coriolis forces, and $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^n$ denotes forces due to gravity (Yoshikawa 1990; Wit et al. 1996). At many points we will write the dynamics equations as $\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{u}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}})$ where $\mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}) = -\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{G}(\mathbf{q})$ for notational convenience. We assume that a sufficiently accurate model of our robot system is available.

A task for the robot is assumed to be described in form of a constraint, i.e., it is given by a function

$$\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, t) = 0 \tag{2}$$

where $\mathbf{h} \in \mathbb{R}^k$ with an arbitrary dimensionality k. For example, if the robot is supposed to follow a desired trajectory $\mathbf{q}_{\text{des}}(t) \in \mathbb{R}^n$, we could formulate it by $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{q} - \mathbf{q}_{\text{des}}(t) = 0$; this case is analyzed in detail in Sect. 3.1. The function h can be considered a task function in the sense of the framework proposed in (Samson et al. 1991).

We consider only tasks where (2) can be reformulated as

$$\mathbf{A}(\mathbf{q}, \dot{\mathbf{q}}, t)\ddot{\mathbf{q}} = \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}, t), \tag{3}$$

which can be achieved for most tasks by differentiation of (2) with respect to time, assuming that h is sufficiently smooth. For example, our previous task, upon differentiation, becomes $\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_{\text{des}}(t)$ so that $\mathbf{A} = \mathbf{I}$ and $\mathbf{b} = \ddot{\mathbf{q}}_{\text{des}}(t)$. An advantage of this task formulation is that non-holonomic constraints can be treated in the same general way.

In Sect. 3, we will give task descriptions first in the general form of (2), and then derive the resulting controller, which is the linear in accelerations, as shown in (3).

2.2 Optimal control framework

Let us assume that we are given a robot model and a constraint formulation of the task as described in the previous section. In this case, we can characterize the desired properties of the framework as follows: first, the task has to be achieved perfectly, i.e., $h(\mathbf{q}, \dot{\mathbf{q}}, t) = 0$, or equivalently, $\mathbf{A}(\mathbf{q}, \dot{\mathbf{q}}, t)\ddot{\mathbf{q}} = \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}, t)$, holds at all times. Second, we intend to minimize the control command with respect to some given metric, i.e., $J(t) = \mathbf{u}^T\mathbf{N}(t)\mathbf{u}$, at *each* instant of time, with positive semi-definite matrix $\mathbf{N}(t)$. The solution to this point-wise optimal control problem (Spo 1984; Spong et al. 1986) can be derived from a generalization of Gauss' principle, as originally suggested in (Udwadia 2003). It is also a generalization of the propositions in (Udwadia and Kalaba 1996; Bruyninckx and Khatib 2000). We formalize this idea in the following proposition.

Proposition 1 *The class of controllers which minimizes*

$$J(t) = \mathbf{u}^T\mathbf{N}(t)\mathbf{u}, \tag{4}$$

for a mechanical system $\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{u}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}})$ *while fulfilling the task constraint*

$$\mathbf{A}(\mathbf{q}, \dot{\mathbf{q}}, t)\ddot{\mathbf{q}} = \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}, t), \tag{5}$$

is given by

$$\mathbf{u} = \mathbf{N}^{-1/2}(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})^{+}(\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}\mathbf{F}), \tag{6}$$

where $\mathbf{D}^{+}$ *denotes the pseudo-inverse for a general matrix* $\mathbf{D}$, *and* $\mathbf{D}^{1/2}$ *denotes the symmetric, positive definite matrix for which* $\mathbf{D}^{1/2}\mathbf{D}^{1/2} = \mathbf{D}$.

Proof By defining $\mathbf{z} = \mathbf{N}^{1/2}\mathbf{u} = \mathbf{N}^{1/2}(\mathbf{M}\ddot{\mathbf{q}} - \mathbf{F})$, we obtain the accelerations $\ddot{\mathbf{q}} = \mathbf{M}^{-1}\mathbf{N}^{-1/2}(\mathbf{z} + \mathbf{N}^{1/2}\mathbf{F})$. Since the task constraint $\mathbf{A}\ddot{\mathbf{q}} = \mathbf{b}$ has to be fulfilled, we have

$$\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2}\mathbf{z} = \mathbf{b} - \mathbf{A}\mathbf{M}^{-1}\mathbf{F}. \tag{7}$$

The vector $\mathbf{z}$ which minimizes $J(t) = \mathbf{z}^T\mathbf{z}$ while fulfilling equation (7), is given by $\mathbf{z} = (\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})^{+}(\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}\mathbf{F})$, and as the motor command is given by $\mathbf{u} = \mathbf{N}^{-1/2}\mathbf{z}$, the proposition holds. □

The choice of the metric $\mathbf{N}$ plays a central role as it determines how the control effort is distributed over the joints. Often, we require a solution which has a kinematic interpretation; such a solution is usually given by a metric like $\mathbf{N} = (\mathbf{M} \cdot \mathbf{M})^{-1} = \mathbf{M}^{-2}$. In other cases, the control force $\mathbf{u}$ may be required to comply with the principle of virtual displacements by d'Alembert for which the metric $\mathbf{N} = \mathbf{M}^{-1}$ is more appropriate (Udwadia and Kalaba 1996; Bruyninckx and Khatib 2000). In practical cases, one would want to distribute the forces such that joints with stronger motors get a higher workload which can also be achieved by a metric such as $\mathbf{N} = \mathrm{diag}(\hat{\tau}_1^{-2}, \hat{\tau}_2^{-2}, \ldots, \hat{\tau}_n^{-2})$ where the nominal torques $\hat{\tau}_i$ are used for the appropriate distribution of the motor commands. In Sect. 3, we will see how the choice of $\mathbf{N}$ results in several different controllers.

2.3 Necessary conditions for stability

Up to this point, this framework has been introduced in an idealized fashion neglecting the possibility of imperfect initial conditions and measurement noise. Therefore, we modify our approach slightly for ensuring stability. However, the stability of this framework as well as most related approaches derivable form this framework cannot be shown conclusively but only in special cases (Hsu et al. 1989; Arimoto 1996). Therefore, we can only outline the necessary conditions for stability, i.e., (i) the achievement of the task which will be achieved through a modification of the framework in Sect. 2.3.1 and (ii) the prevention of undesired side-effects in joint-space. The later are a result of under-constrained tasks, i.e., tasks where some degrees of freedom of the robot are redundant for task achievements, can cause undesired postures or even instability in joint-space. This problem will be treated in Sect. 2.3.2.

2.3.1 Task achievement

Up to this point, we have assumed that we always have perfect initial conditions, i.e., that the robot fulfills the constraint in (3) at startup, and that we know the robot model perfectly. Here, we treat deviations to these assumptions as disturbances and add means of disturbance rejections to our framework. This disturbance rejection can be achieved by requiring that the desired task is an attractor, e.g., it could be prescribed as a dynamical system in the form

$$\dot{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{f}_{\mathbf{h}}(\mathbf{h}, t), \tag{8}$$

where $\mathbf{h} = \mathbf{0}$ is a globally asymptotically stable equilibrium point – or a locally asymptotically stable equilibrium point with a sufficiently large region of attraction. Note that $\mathbf{h}$ can be a function of robot variables (as in end-effector trajectory control in Sect. 3.2) but often it suffices to choose it as a function of the state vector (for example for joint-space trajectory control as in Sect. 3.1). In the case of holonomic tasks (such as tracking control for a robot arm), i.e. $h_i(\mathbf{q}, t) = 0$, $i = 1, 2, \ldots, k$ we can make use of a particularly simple form and turn this task into an attractor

$$\ddot{h}_i + \delta_i \dot{h}_i + \kappa_i h = 0, \tag{9}$$

where δ_i and κ_i are chosen appropriately. We will make use of this 'trick' in order to derive several algorithms. Obviously, different attractors with more desirable convergence properties (and/or larger basins of attraction) can be obtained by choosing $\mathbf{f}_{\mathbf{h}}$ appropriately.

If we have such task-space stabilization, we can assure that the control law will achieve the task at least in a region near to the desired trajectory. We demonstrate this issue in the following proposition.

Proposition 2 *If we can assure the attractor property of the task* $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{0}$, *or equivalently,* $\mathbf{A}(\mathbf{q}, \dot{\mathbf{q}}, t)\ddot{\mathbf{q}} = \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}, t)$, *and if our robot model is accurate, the optimal controller of* (6) *is will achieve the task asymptotically.*

Proof When combining the robot dynamics equation with the controller, and after reordering the terms, we obtain

$$\mathbf{A}\mathbf{M}^{-1}(\mathbf{M}\ddot{\mathbf{q}} - \mathbf{F}) = (\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})^{+}(\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}\mathbf{F}). \tag{10}$$

If we now premultiply the equation with $\mathbf{D} = \mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2}$, and noting that $\mathbf{D}\mathbf{D}^{+}\mathbf{D} = \mathbf{D}$, we obtain $\mathbf{A}\ddot{\mathbf{q}} = \mathbf{D}\mathbf{D}^{+}\mathbf{b} = \mathbf{b}$. The equality follows because the original trajectory defined by $\mathbf{A}\ddot{\mathbf{q}} = \mathbf{b}$ yields a consistent set of equations. If this equation describes an attractor, we will have asymptotically perfect task achievement. □

In some cases, such as joint trajectory tracking control discussed in Sect. 3.1, Proposition 2 will suffice for a stability proof in a Lyapunov sense (Yoshikawa 1990; Wit et al.

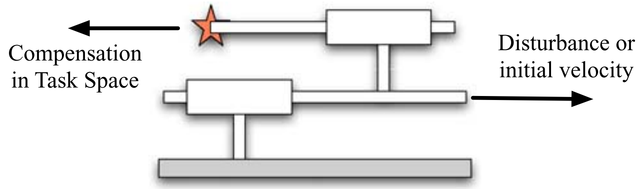

Fig. 1 In the presence of disturbances or non-zero initial conditions, stable task dynamics will not result in joint-space stability

1996). However, for certain tasks such as end-effector tracking control discussed in Sect. 3.1, this is not the case and stability can only be assured in special cases (Hsu et al. 1989; Arimoto 1996).

2.3.2 Prevention of control problems in joint-space

Even if stability in task space can be shown, it is not immediately clear whether the control law is stable in joint-space. Example 1, illustrates a problematic situation where a redundant robot arm achieves an end-effector tracking task and is provably stable in task-space, but nevertheless also provably *unstable in joint-space*.

Example 1 Let us assume a simple prismatic robot with two horizontal, parallel links as illustrated in Fig. 1. The mass matrix of this robot is a constant given by $\mathbf{M} = \mathrm{diag}(m_1, 0) + m_2\mathbf{1}$ where $\mathbf{1}$ denotes a matrix having only ones as entries, and the additional forces are $\mathbf{F} = \mathbf{C} + \mathbf{G} = \mathbf{0}$. Let us assume the task is to move the end-effector $x = q_1 + q_2$ along a desired position x_{des}, i.e., the task can be specified by $\mathbf{A} = [1, 1]$, and $b = \ddot{x}_{\text{des}} + \delta(\dot{x}_{\text{des}} - \dot{x}) + \kappa(x_{\text{des}} - x)$ after double differentiation and task stabilization. While the task dynamics are obviously stable (which can be verified using the constant Eigenvalues of the system), the initial condition $q_1(t_0) = x_{\text{des}}(t_0) - q_2(t_0)$ would result in both $q_i(t)$'s diverging into opposite directions for any non-zero initial velocities or in the presence of disturbances for arbitrary initial conditions. The reason for this behavior is obvious: the effort of stabilizing in joint space is not task relevant and would increase the cost.

While this example is similar to problems with non-minimum phase nonlinear control systems (Isidori 1995), the problems encountered are not the failure of the task controller, but rather due to internal dynamics, e.g., hitting of joint limits. From this example, we see that the basic general framework of constrained mechanics does not always suffice to derive useful control laws, and that it has to be augmented to incorporate joint stabilization for the robot without affecting the task achievement. One possibility is to introduce a joint-space motor command $\mathbf{u}_1$ as an additional component of the motor command $\mathbf{u}$, i.e.,

$$\mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2(\mathbf{u}_1), \tag{11}$$

where the first component $\mathbf{u}_1$ denotes an arbitrary joint-space motor command for stabilization, while the second component $\mathbf{u}_2(\mathbf{u}_1)$ denotes the task-space motor command generated with the previously explained equations. The task-space component depends on the joint-space component as it has to compensate for it in the range of the task space. We can show that task achievement $\mathbf{A}\ddot{\mathbf{q}} = \mathbf{b}$ by the controller is not affected by the choice of the joint-space control law $\mathbf{u}_1$.

Proposition 3 *For any chosen joint stabilizing control law* $\mathbf{u}_1 = f(\mathbf{q})$, *the resulting task space control law* $\mathbf{u}_2(\mathbf{u}_1)$ *ensures that the joint-space stabilization acts in the null-space of the task achievement.*

Proof When determining $\mathbf{u}_2$, we consider $\mathbf{u}_1$ to be part of our additional forces in the rigid body dynamics, i.e., we have $\tilde{\mathbf{F}} = \mathbf{F} + \mathbf{u}_1$. We obtain $\mathbf{u}_2 = \mathbf{N}^{-1/2}(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})^+(\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}\tilde{\mathbf{F}})$ using Proposition 1. By reordering the complete control law $\mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2(\mathbf{u}_1)$, we obtain

$$\begin{aligned}\mathbf{u} &= \mathbf{u}_1 + \mathbf{N}^{-1/2}(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})^+(\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}(\mathbf{F} + \mathbf{u}_1)),\\ &= \mathbf{N}^{-1/2}(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})^+(\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}\mathbf{F})\\ &\quad + [\mathbf{I} - \mathbf{N}^{-1/2}(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})^+\mathbf{A}\mathbf{M}^{-1}]\mathbf{u}_1,\\ &= \mathbf{N}^{-1/2}(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})^+(\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}\mathbf{F})\\ &\quad + \mathbf{N}^{-1/2}[\mathbf{I} - (\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})^+(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})]\mathbf{N}^{1/2}\mathbf{u}_1.\end{aligned} \tag{12}$$

The task space is defined by $\mathbf{N}^{-1/2}(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})^+$, and the matrix $\mathbf{N}^{-1/2}[\mathbf{I} - (\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})^+(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})]$ makes sure that the joint-space control law and the task space control law are $\mathbf{N}$-orthogonal, i.e., the task accomplishment is independent of the joint-space stabilization. □

Despite that the task is still achieved, the optimal control problem is affected by the restructuring of our control law. While we originally minimized $J(t) = \mathbf{u}^T\mathbf{N}(t)\mathbf{u}$, we now have a modified cost function

$$\tilde{J}(t) = \mathbf{u}_2^T\mathbf{N}(t)\mathbf{u}_2 = (\mathbf{u} - \mathbf{u}_1)^T\mathbf{N}(t)(\mathbf{u} - \mathbf{u}_1), \tag{13}$$

which is equivalent to stating that the complete control law $\mathbf{u}$ should be as close to the joint-space control law $\mathbf{u}_1$ as possible under task achievement.

This reformulation can have significant advantages if used appropriately. For example, in a variety of applications —such as using the robot as a haptic interface—a compensation of the robot's gravitational, coriolis and centrifugal forces in joint space can be useful. Such a compensation can only be derived when making use of the modified control law. In this case, we set $\mathbf{u}_1 = -\mathbf{F} = \mathbf{C} + \mathbf{G}$, which allows us to obtain

$$\mathbf{u}_2 = \mathbf{N}^{-1/2}(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})^+\mathbf{b}, \tag{14}$$

which does not contain these forces, and we would have a complete control law of $\mathbf{u} = \mathbf{C} + \mathbf{G} + \mathbf{N}^{-1/2}(\mathbf{A}\mathbf{M}^{-1} \times \mathbf{N}^{-1/2})^{+}\mathbf{b}$.

2.4 Hierarchical extension

In complex high-dimensional systems, we can often have a large number of tasks $\mathbf{A}_1\ddot{\mathbf{q}} = \mathbf{b}_1$, $\mathbf{A}_2\ddot{\mathbf{q}} = \mathbf{b}_2, \ldots, \mathbf{A}_n\ddot{\mathbf{q}} = \mathbf{b}_n$ that have to be accomplished in parallel. These tasks often partially conflict, e.g., when the number of tasks exceeds the number of degrees of freedom or some of these tasks cannot be achieved in combination with each other. Therefore, the combination of these tasks into a single large task is not always practical and, instead, the tasks need prioritization, e.g., the higher the number of the task, the higher its priority. Task prioritized control solutions have been discussed in the literature (Nakamura et al. 1987; Hollerbach and Suh 1987; Maciejewski and Klein 1985; Hanafusa et al. 1981; Yamane and Nakamura 2003; Sentis and Khatib 2004; Siciliano and Slotine 1991; Khatib et al. 2004; Sentis and Khatib 2005). Most previous approaches were kinematic and discussed only a small, fixed number of tasks; to our knowledge, (Sentis and Khatib 2004, 2005) were among the first to discuss arbitrary task numbers and dynamical decoupling, i.e., a different metric from our point of view. The proposed framework allows the generalization for arbitrary metrics and more general problems as will be shown in Proposition 3. The prioritized motor command is given by

$$\mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2(\mathbf{u}_1) + \mathbf{u}_3(\mathbf{u}_1 + \mathbf{u}_2) + \cdots + \mathbf{u}_n(\mathbf{u}_1 + \cdots + \mathbf{u}_{n-1}),$$

where $\mathbf{u}_n(\mathbf{u}_1 + \cdots + \mathbf{u}_{n-1})$ is the highest-priority control law as a function of $\mathbf{u}_1, \ldots, \mathbf{u}_{n-1}$ and cancels out all influence $\mathbf{u}_1 + \cdots + \mathbf{u}_{n-1}$ which prohibit the execution of its task. The motor commands for each degree of freedom can be given by the following Proposition:

Proposition 4 *A set of hierarchically prioritized constraints $\mathbf{A}_i\ddot{\mathbf{q}} = \mathbf{b}_i$ where $i = 1, 2, \ldots, n$ represents the priority (here, a higher number i represents a higher priority) can be controlled by*

$$\mathbf{u} = \mathbf{u}_1 + \sum_{i=2}^{n} \mathbf{u}_i \left(\sum_{k=1}^{i-1} \mathbf{u}_k \right),$$

where $\mathbf{u}_i(\mathbf{u}_\Sigma) = \mathbf{N}^{-1/2}(\mathbf{A}_i\mathbf{M}^{-1}\mathbf{N}^{-1/2})^{+}(\mathbf{b} - \mathbf{A}_i\mathbf{M}^{-1}(\mathbf{F} + \mathbf{u}_\Sigma))$. For any $k < i$, the lower-priority control law $\mathbf{u}_k$ acts in the null-space of the higher-priority control law $\mathbf{u}_i$ and any higher-priority control law $\mathbf{u}_i$ cancels all parts of the lower-priority control law $\mathbf{u}_k$ which conflict with its task achievement.

Proof We first simply create the control laws $\mathbf{u}_1$ and $\mathbf{u}_2(\mathbf{u}_1)$ as described before and then make use of Proposition 3, which proves that this approach is correct for $n = 2$. Let us assume now that it is true for $n = m$. In this case, we can consider $\tilde{\mathbf{u}}_1 = \mathbf{u}_1 + \mathbf{u}_2 + \cdots + \mathbf{u}_m$ our joint-space control law and $\tilde{\mathbf{u}}_2 = \mathbf{u}_{m+1}$ the task-space control law. If we now make use of Proposition 3 again, we realize that $\tilde{\mathbf{u}}_1 = \mathbf{u}_1 + \mathbf{u}_2 + \cdots + \mathbf{u}_m$ acts in the null-space of $\tilde{\mathbf{u}}_2 = \mathbf{u}_{m+1}$ and that all components of $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m$ which conflict with $\mathbf{u}_{m+1}$ will be canceled out. Therefore, the proposition also holds true for $n = m + 1$. This proves the proposition by induction. □

From the viewpoint of optimization, the control laws obtained in Proposition 4 have a straightforward interpretation like the combination of joint and task-space control laws: each subsequent control law is chosen so that the control effort deviates minimally from the effort created from the previous control laws.

Example 2 Robot locomotion is a straightforward example for such an approach. Traditionally, all tasks are often meshed into one big tasks (Pratt and Pratt 1998). However, the most essential task is the balancing of the robot to prevent accidents; it can, for instance, be achieved by a balancing task $\mathbf{A}_3\ddot{\mathbf{q}} = \mathbf{b}_3$ similar to a spring-damper system pulling the system to an upright position. Additionally, the center of the torso should follow a desired trajectory – unless the desired path would make the robot fall. This gait generating task would be given by $\mathbf{A}_2\ddot{\mathbf{q}} = \mathbf{b}_2$. Additionally, we want to have joint-space stability as the unconstrained degrees of freedom such as the arms might otherwise move all the time. The joint-space stabilization can be expressed as a constraint $\mathbf{A}_1\ddot{\mathbf{q}} = \mathbf{b}_1$ pulling the robot towards a rest posture. The combined motor command is now given by $\mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2(\mathbf{u}_1) + \mathbf{u}_3(\mathbf{u}_1 + \mathbf{u}_2)$ with the single control laws are obtained by $\mathbf{u}_i(\mathbf{u}_\Sigma) = \mathbf{N}^{-1/2}(\mathbf{A}_i\mathbf{M}^{-1}\mathbf{N}^{-1/2})^{+}(\mathbf{b} - \mathbf{A}_i\mathbf{M}^{-1}(\mathbf{F} + \mathbf{u}_\Sigma))$ with $i = 1, 2, 3$.

Ideas similar to Example 2 have been explored in (Yamane and Nakamura 2003; Sentis and Khatib 2004, 2005; Khatib et al. 2004) and we are currently working on applying this framework to locomotion similar to Example 2.

3 Robot control laws

The previously described framework offers a variety of applications in robotics—we will only focus on some the most important ones in this paper. Most of these controllers which we will derive are known from the literature, but often from very different, and sometimes convoluted, building principles. In this section, we show how a large variety of control laws for different situations can be derived in a simple and straightforward way by using the unifying framework that has been developed hereto. We derive control laws for joint-space trajectory control for both fully actuated and overactuated "muscle-like" robot systems from our framework. We also discuss task-space tracking control systems, and show that most well-known inverse kinematics controllers are applications of the same principle. Additionally, we will discuss how the control of constrained manipulators through impedance and hybrid control can be easily handled within our framework.

3.1 Joint-space trajectory control

The first control problem we address is joint-space trajectory control. We consider two different situations: (a) We control a fully actuated robot arm in joint-space, and (b) we control an overactuated arm. The case (b) could, for example, have agonist-antagonist muscles as actuators similar to a human arm.[2]

3.1.1 *Fully actuated robot*

The first case which we consider is the one of a robot arm which is actuated at every degree of freedom. We have the trajectory as constraint with $\mathbf{h}(\mathbf{q}, t) = \mathbf{q}(t) - \mathbf{q}_d(t) = \mathbf{0}$. We turn this constraint into an attractor constraint using the idea in Sect. 2.3.1, yielding

$$(\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_d) + \mathbf{K}_D(\dot{\mathbf{q}} - \dot{\mathbf{q}}_d) + \mathbf{K}_P(\mathbf{q} - \mathbf{q}_d) = 0, \tag{15}$$

where $\mathbf{K}_D$ are positive-definite damping gains, and $\mathbf{K}_P$ are positive-definite proportional gains. We can bring this constraint into the form $\mathbf{A}(\mathbf{q}, \dot{\mathbf{q}})\ddot{\mathbf{q}} = \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})$ with

$$\mathbf{A} = \mathbf{I}, \tag{16}$$

$$\mathbf{b} = \ddot{\mathbf{q}}_d + \mathbf{K}_D(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) - \mathbf{K}_P(\mathbf{q}_d - \mathbf{q}). \tag{17}$$

Proposition 1 can be used to derive the controller. Using $(\mathbf{M}^{-1}\mathbf{N}^{-1/2})^{+} = \mathbf{N}^{1/2}\mathbf{M}$ as both matrices are of full rank, we obtain

$$\begin{aligned}\mathbf{u} &= \mathbf{u}_1 + \mathbf{N}^{-1/2}(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})^{+}(\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}(\mathbf{F} + \mathbf{u}_1)),\\ &= \mathbf{M}(\ddot{\mathbf{q}}_d + \mathbf{K}_D(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_P(\mathbf{q}_d - \mathbf{q})) + \mathbf{C} + \mathbf{G}. \end{aligned}\tag{18}$$

Note that—not surprisingly—all joint-space motor commands or virtual forces $\mathbf{u}_1$ always disappear from the control law and that the chosen metric $\mathbf{N}$ is not relevant—the derived solution is unique and general. This equation is a well-known text book control law, i.e., the *Inverse Dynamics Control Law* (Yoshikawa 1990; Wit et al. 1996).

[2] An open topic of interest is to handle underactuated control systems. This will be part of future work.

3.1.2 *Overactuated robots*

Overactuated robots, as they can be found in biological systems, are inheritently different from the previously discussed robots. For instance, these systems are actuated by several linear actuators, e.g., muscles that often act on the system in form of opposing pairs. The interactions of the actuators can be modeled using the dynamics equations of

$$\mathbf{D}\mathbf{u} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}), \tag{19}$$

where $\mathbf{D}$ depends on the geometric arrangement of the actuators. In the simple model of a two degree-of-freedom robot with antagonistic muscle-like activation, it would be given by

$$\mathbf{D} = \begin{bmatrix} -l & +l & 0 & 0 \\ 0 & 0 & -l & +l \end{bmatrix}, \tag{20}$$

where size of the entries $\mathbf{D}_{ij}$ denotes the moment arm length l_i and the sign of $\mathbf{D}_{ij}$ whether its agonist ($\mathbf{D}_{ij} > 0$) or antagonist muscle ($\mathbf{D}_{ij} < 0$). We can bring this equation into the standard form by multiplying it with $\mathbf{D}^{+}$, which results in a modified system where $\tilde{\mathbf{M}}(\mathbf{q}) = \mathbf{D}^{+}\mathbf{M}(\mathbf{q})$, and $\tilde{\mathbf{F}}(\mathbf{q}, \dot{\mathbf{q}}) = -\mathbf{D}^{+}\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{D}^{+}\mathbf{G}(\mathbf{q})$. If we express the desired trajectory as in the previous examples, we obtain the following controller

$$\mathbf{u} = \tilde{\mathbf{M}}^{1/2}(\mathbf{A}\tilde{\mathbf{M}}^{-1/2})^{+}(\mathbf{b} - \mathbf{A}\tilde{\mathbf{M}}^{-1}\tilde{\mathbf{F}}), \tag{21}$$

$$\begin{aligned} &= \mathbf{D}^{+}\mathbf{M}(\ddot{\mathbf{q}}_d + \mathbf{K}_D(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) - \mathbf{K}_P(\mathbf{q}_d - \mathbf{q}))\\ &\quad + \mathbf{D}^{+}(\mathbf{C} + \mathbf{G}). \end{aligned}\tag{22}$$

While immediately intuitive, it noteworthy that this particular controller falls out of the presented framework in an natural way. It is straightforward to extend Proposition 1 to show that this is the constrained optimal solution to $J(t) = \mathbf{u}^T\mathbf{D}\mathbf{N}(t)\mathbf{D}\mathbf{u}$ at any instant of time.

3.2 End-effector trajectory control

While joint-space control of a trajectory $\mathbf{q}(t)$ is straightforward and the presented methodology appears to simply repeat earlier results from the literature—although derived from a different and unified perspective—the same cannot be said about end-effector control where goal is to control the position $\mathbf{x}(t)$ of the end-effector. This problem is generically more difficult as the choice of the metric $\mathbf{N}$ determines the type of the resulting controller in an important way, and as the joint-space of the robot often has redundant degrees of freedom resulting in problems as already presented in Example 1. In the following, we will show how to derive different approaches to end-effector control from the presented framework, which will yield both established as well as novel control laws.

The task description is given by the end-effector trajectory as constraint with $\mathbf{h}(\mathbf{q},t) = \mathbf{f}(\mathbf{q}(t)) - \mathbf{x}_d(t) = \mathbf{x}(t) - \mathbf{x}_d(t) = \mathbf{0}$, where $\mathbf{x} = \mathbf{f}(\mathbf{q})$ denotes the forward kinematics. We turn this constraint into an attractor constraint using the idea in Sect. 2.3.1, yielding

$$(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_d) + \mathbf{K}_D(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) + \mathbf{K}_P(\mathbf{x} - \mathbf{x}_d) = 0, \tag{23}$$

where $\mathbf{K}_D$ are positive-definite damping gains, and $\mathbf{K}_P$ are positive-definite proportional gains. We make use of the differential forward kinematics, i.e.,

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \tag{24}$$

$$\ddot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}}. \tag{25}$$

These equations allow us to formulate the problem in form of constraints, i.e., we intend to fulfill

$$\ddot{\mathbf{x}}_d + \mathbf{K}_D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{K}_P(\mathbf{x}_d - \mathbf{x}) = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}}, \tag{26}$$

and we can bring this equation into the form $\mathbf{A}(\mathbf{q},\dot{\mathbf{q}})\ddot{\mathbf{q}} = \mathbf{b}(\mathbf{q},\dot{\mathbf{q}})$ with

$$\mathbf{A}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{J}, \tag{27}$$

$$\mathbf{b}(\mathbf{q},\dot{\mathbf{q}}) = \ddot{\mathbf{x}}_d + \mathbf{K}_D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{K}_P(\mathbf{x}_d - \mathbf{x}) - \dot{\mathbf{J}}\dot{\mathbf{q}}. \tag{28}$$

These equations determine our task constraints. As long as the robot is not redundant $\mathbf{J}$ is invertible and similar to joint-space control, we will have one unique control law. However, when $\mathbf{J}$ is not uniquely invertible the resulting controller depends on the chosen metric and joint-space control law.

3.2.1 Separation of kinematics and dynamics control

The choice of the metric $\mathbf{N}$ determines the nature of the controller. A metric of particular importance is $\mathbf{N} = \mathbf{M}^{-2}$ as this metric allows the decoupling of kinematics and dynamics control as we will see in this section. Using this metric in Proposition 1, we obtain a control law

$$\begin{aligned}
\mathbf{u} &= \mathbf{u}_1 + \mathbf{N}^{-1/2}(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})^{+}(\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}(\mathbf{F} + \mathbf{u}_1)),\\
&= \mathbf{M}\mathbf{J}^{+}(\ddot{\mathbf{x}}_d + \mathbf{K}_D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{K}_P(\mathbf{x}_d - \mathbf{x}) - \dot{\mathbf{J}}\dot{\mathbf{q}})\\
&\quad + \mathbf{M}(\mathbf{I} - \mathbf{J}^{+}\mathbf{J})\mathbf{M}^{-1}\mathbf{u}_1 - \mathbf{M}\mathbf{J}^{+}\mathbf{J}\mathbf{M}^{-1}\mathbf{F}.
\end{aligned}$$

If we choose the joint-space control law $\mathbf{u}_1 = \mathbf{u}_0 - \mathbf{F}$, we obtain the control law

$$\begin{aligned}
\mathbf{u} &= \mathbf{M}\mathbf{J}^{+}(\ddot{\mathbf{x}}_d + \mathbf{K}_D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{K}_P(\mathbf{x}_d - \mathbf{x}) - \dot{\mathbf{J}}\dot{\mathbf{q}})\\
&\quad + \mathbf{M}(\mathbf{I} - \mathbf{J}^{+}\mathbf{J})\mathbf{M}^{-1}\mathbf{u}_0 + \mathbf{C} + \mathbf{G}.
\end{aligned} \tag{29}$$

This control law is the combination of a *resolved-acceleration kinematic controller* (Yoshikawa 1990; Hsu et al. 1989) with a model-based controller and an additional null-space term. Often, $\mathbf{M}^{-1}\mathbf{u}_0$ is replaced by a desired acceleration term for the null-space stabilization. Similar controllers have been introduced in (Park et al. 2002, 1995; Chung et al. 1993; Suh and Hollerbach 1987). The null-space term can be eliminated by setting $\mathbf{u}_0 = \mathbf{0}$; however, this can result in instabilities if there are redundant degrees of freedom. This controller will be evaluated in Sect. 4.

3.2.2 Dynamically consistent decoupling

As noted earlier, another important metric is $\mathbf{N} = \mathbf{M}^{-1}$ as it is consistent with the principle of d'Alembert, i.e., the resulting control force can be re-interpreted as mechanical structures (e.g., springs and dampers) attached to the end-effector; it is therefore called dynamically consistent. Again, we use Proposition 1, and by defining $\tilde{\mathbf{F}} = \mathbf{F} + \mathbf{u}_1$ obtain the control law

$$\begin{aligned}
\mathbf{u} &= \mathbf{u}_1 + \mathbf{N}^{-1/2}(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})^{+}(\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}\tilde{\mathbf{F}}),\\
&= \mathbf{u}_1 + \mathbf{M}^{1/2}(\mathbf{J}\mathbf{M}^{-1/2})^{T}(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{T})^{-1}(\mathbf{b} - \mathbf{J}\mathbf{M}^{-1}\tilde{\mathbf{F}}),\\
&= \mathbf{u}_1 + \mathbf{J}^{T}(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{T})^{-1}(\mathbf{b} - \mathbf{J}\mathbf{M}^{-1}\tilde{\mathbf{F}}),\\
&= \mathbf{J}^{T}(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{T})^{-1}(\ddot{\mathbf{x}}_d + \mathbf{K}_D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}})\\
&\quad + \mathbf{K}_P(\mathbf{x}_d - \mathbf{x}) - \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{J}\mathbf{M}^{-1}(\mathbf{C} + \mathbf{G}))\\
&\quad + \mathbf{M}(\mathbf{I} - \mathbf{M}^{-1}\mathbf{J}^{T}(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{T})^{-1}\mathbf{J})\mathbf{M}^{-1}\mathbf{u}_1.
\end{aligned}$$

It turns out that this is another well-known control law suggest in (Khatib 1987) with an additional null-space term. This control-law is especially interesting as it has a clear physical interpretation (Udwadia and Kalaba 1996; Bruyninckx and Khatib 2000; Udwadia 2003): the metric used is consistent with principle of virtual work of d'Alembert. Similarly as before we can compensate for coriolis, centrifugal and gravitational forces in joint-space, i.e., setting $\mathbf{u}_1 = \mathbf{C} + \mathbf{G} + \mathbf{u}_0$. This yields a control law of

$$\begin{aligned}
\mathbf{u} &= \mathbf{J}^{T}(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{T})^{-1}(\ddot{\mathbf{x}}_d + \mathbf{K}_D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}})\\
&\quad + \mathbf{K}_P(\mathbf{x}_d - \mathbf{x}) - \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}}) + \mathbf{C} + \mathbf{G}\\
&\quad + \mathbf{M}[\mathbf{I} - \mathbf{M}^{-1}\mathbf{J}^{T}(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{T})^{-1}\mathbf{J}]\mathbf{M}^{-1}\mathbf{u}_0.
\end{aligned} \tag{30}$$

The compensation of the forces $\mathbf{C} + \mathbf{G}$ in joint-space is often desirable for this metric in order to have full control over the resolution of the redundancy as gravity compensation purely in task space often results in postures that conflict with joint limits and other parts of the robot.

3.2.3 Further metrics

Using the identity matrix as metric, i.e., $\mathbf{N} = \mathbf{I}$, punishes the squared motor command without reweighting, e.g., with inertial terms. This metric could be of interest as it distributes the "load" created by the task evenly on the actuators. This metric results in a control law

$$\mathbf{u} = (\mathbf{J}\mathbf{M}^{-1})^{+}(\ddot{\mathbf{x}}_d + \mathbf{K}_D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{K}_P(\mathbf{x}_d - \mathbf{x}) - \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{J}\mathbf{M}^{-1}(\mathbf{C} + \mathbf{G})) + (\mathbf{I} - (\mathbf{J}\mathbf{M}^{-1})^{+}\mathbf{J}\mathbf{M}^{-1})\mathbf{u}_1. \quad (31)$$

To our knowledge, this controller has not been presented in the literature.

Another, fairly practical idea would be to weight the different joints depending on the maximal torques $\tau_{\max,i}$ of each joint; this would result in a metric $\mathbf{N} = \operatorname{diag}(\tau_{\max,1}^{-1}, \ldots, \tau_{\max,n}^{-1})$.

These alternative metrics may be particularly interesting for practical application where the user wants to have more control over the natural appearance of movement, and worry less about the exact theoretical properties—humanoid robotics, for instance, is one of such applications. In some cases, it also may not be possible to have accurate access to complex metrics like the inertia matrix, and simpler metrics will be more suitable.

3.3 Controlling constrained manipulators: impedance & hybrid control

Contact with outside objects alters the robot's dynamics, i.e., a generalized contact force $\mathbf{F}_C \in \mathbb{R}^6$ acting on the end-effector changes the dynamics of the robot to

$$\mathbf{u} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \mathbf{J}^T\mathbf{F}_C. \quad (32)$$

In this case, the interaction between the robot and the environment has to be controlled. This kind of control can both be used to make the interaction with the environment safe (e.g., in a manipulation task) as well as to use the robot to simulate a behavior (e.g., in a haptic display task). We will discuss impedance control and hybrid control as examples of the application of the proposed framework; however, further control ideas such as parallel control can be treated in this framework, too.

3.3.1 Impedance control

In impedance control, we want the robot to simulate the behavior of a mechanical system such as

$$\mathbf{M}_d(\ddot{\mathbf{x}}_d - \ddot{\mathbf{x}}) + \mathbf{D}_d(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{P}_d(\mathbf{x}_d - \mathbf{x}) = \mathbf{F}_C, \quad (33)$$

where $\mathbf{M}_d \in \mathbb{R}^{6\times 6}$ denotes the mass matrix of a desired simulated dynamical system, $\mathbf{D}_d \in \mathbb{R}^6$ denotes the desired damping, $\mathbf{P}_d \in \mathbb{R}^6$ denotes the gains towards the desired position, and $\mathbf{F}_C \in \mathbb{R}^6$ the forces that result from this particular dynamical behavior. Using (25) from Sect. 3.2, we see that this approach can be brought in the standard form for tasks by

$$\mathbf{M}_d\mathbf{J}\ddot{\mathbf{q}} = \mathbf{F}_C - \mathbf{M}_d\ddot{\mathbf{x}}_d - \mathbf{D}_d(\dot{\mathbf{x}}_d - \mathbf{J}\dot{\mathbf{q}}) - \mathbf{P}_d(\mathbf{x}_d - \mathbf{f}(\mathbf{q})) - \mathbf{M}_d\dot{\mathbf{J}}\dot{\mathbf{q}}.$$

Thus, we can infer the task description

$$\mathbf{A} = \mathbf{M}_d\mathbf{J},$$
$$\mathbf{b} = \mathbf{F}_C - \mathbf{M}_d\ddot{\mathbf{x}}_d - \mathbf{D}_d(\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}_d) - \mathbf{P}_d(\mathbf{f}(\mathbf{q}) - \mathbf{x}_d) - \mathbf{M}_d\dot{\mathbf{J}}\dot{\mathbf{q}}, \quad (34)$$

and apply our framework for deriving the robot control law as shown before.

Kinematic separation of simulated system and the manipulator Similar as in end-effector tracking control, a practical metric is $\mathbf{N} = \mathbf{M}^{-2}$ it basically separates the simulated dynamic system from the physical structure of the manipulator on a kinematic level. For simplicity, we make use of the joint-space control law $\mathbf{u}_1 = \mathbf{C} + \mathbf{G} + \mathbf{u}_0$ similar as before. This results in the control law

$$\begin{aligned}\mathbf{u} &= \mathbf{u}_1 + \mathbf{N}^{-1/2}(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})^{+}(\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}(\mathbf{F} + \mathbf{u}_1)),\\ &= \mathbf{M}(\mathbf{M}_d\mathbf{J})^{+}(\mathbf{F}_C - \mathbf{M}_d\ddot{\mathbf{x}}_d - \mathbf{D}_d(\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}_d) - \mathbf{P}_d(\mathbf{f}(\mathbf{q}) - \mathbf{x}_d) - \mathbf{M}_d\dot{\mathbf{J}}\dot{\mathbf{q}})\\ &\quad + \mathbf{C} + \mathbf{G} + (\mathbf{I} - \mathbf{M}(\mathbf{M}_d\mathbf{J})^{+}\mathbf{M}_d\mathbf{J}\mathbf{M}^{-1})\mathbf{u}_0. \end{aligned} \quad (35)$$

As $(\mathbf{M}_d\mathbf{J})^{+} = \mathbf{J}^T\mathbf{M}_d(\mathbf{M}_d\mathbf{J}\mathbf{J}^T\mathbf{M}_d)^{-1} = \mathbf{J}^{+}\mathbf{M}_d^{-1}$ since $\mathbf{M}_d$ is invertible, we can simplify this control law to become

$$\mathbf{u} = \mathbf{M}\mathbf{J}^{+}\mathbf{M}_d^{-1}(\mathbf{F}_C - \mathbf{M}_d\ddot{\mathbf{x}}_d - \mathbf{D}_d(\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}_d) - \mathbf{P}_d(\mathbf{f}(\mathbf{q}) - \mathbf{x}_d)) - \mathbf{M}\mathbf{J}^{+}\dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{C} + \mathbf{G} + \mathbf{M}(\mathbf{I} - \mathbf{J}^{+}\mathbf{J})\mathbf{M}^{-1}\mathbf{u}_0. \quad (36)$$

We note that $\ddot{\mathbf{x}}_d = \mathbf{M}_d^{-1}(\mathbf{F}_C - \mathbf{M}_d\ddot{\mathbf{x}}_d - \mathbf{D}_d(\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}_d) - \mathbf{P}_d(\mathbf{f}(\mathbf{q}) - \mathbf{x}_d))$ is a desired acceleration in task-space. This insight clarifies the previous remark about the separation of the simulated system and the actual physical system: we have a first system which describes the interaction with the environment—and additionally we use a second, inverse-model type controller to execute the desired accelerations with our robot arm.

Dynamically consistent combination Similar as in end-effector control, a practical metric is $\mathbf{N} = \mathbf{M}^{-1}$ which combines both the simulated and the physical dynamic systems employing Gauss' principle. For simplicity, we make use of the joint-space control law $\mathbf{u}_1 = \mathbf{C} + \mathbf{G} + \mathbf{u}_0$ similar as before. This approach results into the control law

$$\begin{aligned}\mathbf{u} &= \mathbf{u}_1 + \mathbf{N}^{-1/2}(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})^{+}(\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}(\mathbf{F} + \mathbf{u}_1)),\\ &= \mathbf{u}_1 + \mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}(\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}(\mathbf{F} + \mathbf{u}_1)),\\ &= \mathbf{M}^{1/2}(\mathbf{M}_d\mathbf{J}\mathbf{M}^{-1/2})^{+}(\mathbf{F}_C - \mathbf{D}_d(\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}_d)\\ &\quad - \mathbf{P}_d(\mathbf{f}(\mathbf{q}) - \mathbf{x}_d) - \mathbf{M}_d\dot{\mathbf{J}}\dot{\mathbf{q}})\\ &\quad + \mathbf{C} + \mathbf{G} + (\mathbf{I} - \mathbf{M}(\mathbf{M}_d\mathbf{J})^{+}\mathbf{M}_d\mathbf{J}\mathbf{M}^{-1})\mathbf{u}_0. \end{aligned} \tag{37}$$

As $(\mathbf{M}_d\mathbf{J}\mathbf{M}^{-1/2})^{+} = \mathbf{M}^{-1/2}\mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}\mathbf{M}_d^{-1}$ since $\mathbf{M}_d$ is invertible, we can simplify this control law into

$$\begin{aligned}\mathbf{u} &= \mathbf{J}^T(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}\mathbf{M}_d^{-1}(\mathbf{F}_C - \mathbf{D}_d(\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}_d)\\ &\quad - \mathbf{P}_d(\mathbf{f}(\mathbf{q}) - \mathbf{x}_d)) - \mathbf{M}\mathbf{J}^{+}\dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{C} + \mathbf{G}\\ &\quad + (\mathbf{I} - \mathbf{M}\mathbf{J}^{+}\mathbf{J}\mathbf{M}^{-1})\mathbf{u}_0. \end{aligned} \tag{38}$$

We note that the main difference between this and the previous impedance control law is the location of the matrix $\mathbf{M}$.

3.3.2 Hybrid control

In hybrid control, we intend to control the desired position of the end-effector $\mathbf{x}_d$ and the desired contact force exerted by the end-effector $\mathbf{F}_d$. Modern hybrid control approaches are essentially similar to our introduced framework (Wit et al. 1996). Both are inspired by constrained motion and use this insight in order to achieve the desired task. In traditional hybrid control, a natural or artificial, idealized holonomic constraint $\phi(\mathbf{q}, t) = \mathbf{0}$ acts on our manipulator, and subsequently the direction of the forces is determined through the virtual work principle of d'Alembert. We can make significant contributions here as our framework is a generalization of the Gauss' principle that allows us to handle even non-holonomic constraints $\phi(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{0}$ as long as they are given in the form

$$\mathbf{A}_\phi(\mathbf{q}, \dot{\mathbf{q}})\ddot{\mathbf{q}} = \mathbf{b}_\phi(\mathbf{q}, \dot{\mathbf{q}}). \tag{39}$$

$\mathbf{A}_\phi$, $\mathbf{b}_\phi$ depend on the type of the constraint, e.g., for scleronomic, holonomic constraints $\phi(\mathbf{q}) = \mathbf{0}$, we would have $\mathbf{A}_\phi(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{J}_\phi$ and $\mathbf{b}_\phi(\mathbf{q}, \dot{\mathbf{q}}) = -\dot{\mathbf{J}}_\phi\dot{\mathbf{q}}$ with $\mathbf{J}_\phi = \partial\phi/\partial\mathbf{q}$ as in (Wit et al. 1996). Additionally, we intend to exert the contact force $\mathbf{F}_d$ in the task; this can be achieved if we choose the joint-space control law

$$\mathbf{u}_1 = \mathbf{C} + \mathbf{G} + \mathbf{J}_\phi^T\mathbf{F}_d. \tag{40}$$

From the previous discussion, this constraint is achieved by the control law

$$\mathbf{u} = \mathbf{u}_1 + \mathbf{N}^{-1/2}(\mathbf{A}_\phi\mathbf{M}^{-1}\mathbf{N}^{-1/2})^{+}(\mathbf{b}_\phi - \mathbf{A}_\phi\mathbf{M}^{-1}(\mathbf{F} + \mathbf{u}_1)), \tag{41}$$

$$\begin{aligned} &= \mathbf{C} + \mathbf{G} + \mathbf{N}^{-1/2}(\mathbf{A}_\phi\mathbf{M}^{-1}\mathbf{N}^{-1/2})^{+}\mathbf{b}_\phi\\ &\quad + \mathbf{N}^{-1/2}(\mathbf{I} - (\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})^{+}\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})\mathbf{N}^{1/2}\mathbf{J}_\phi^T\mathbf{F}_d. \end{aligned} \tag{42}$$

Note that the exerted forces act in the null-space of the achieved tracking task; therefore both the constraint and the force can be set independently.

4 Evaluations

The main contribution of this paper is the unifying methodology for deriving robot controllers. Each of the presented controllers in this paper is a well founded control law which, from a theoretical point of view, would not need require empirical evaluations, particularly as most of the control laws are already well-known from the literature and their stability properties have been explored before. Nevertheless, it is useful to highlight one component in the suggested framework, i.e., the impact of the metric $\mathbf{N}$ on the particular performance of a controller. For this purpose, we chose to evaluate the three end-effector controllers from Sect. 3.2: (i) the resolved-acceleration kinematic controller (with metric $\mathbf{N} = \mathbf{M}^{-2}$) in (29), (ii) Khatib's operational space control law ($\mathbf{N} = \mathbf{M}^{-1}$) in (30), and (iii) the identity metric control law ($\mathbf{N} = \mathbf{I}$) in (31).

As an experimental platform, we used the Sarcos Dextrous Master Arm, a hydraulic manipulator with an anthropomorphic design shown in Fig. 2. Its seven degrees of freedom mimic the major degrees of freedom of the human arm, i.e., there are three DOFs in the shoulder, one in the elbow and three in the wrist. The robot's end-effector was supposed to track a planar "figure-eight (8)" pattern in task space at

Fig. 2 Sarcos Master Arm robot, as used for the evaluations on our experiments

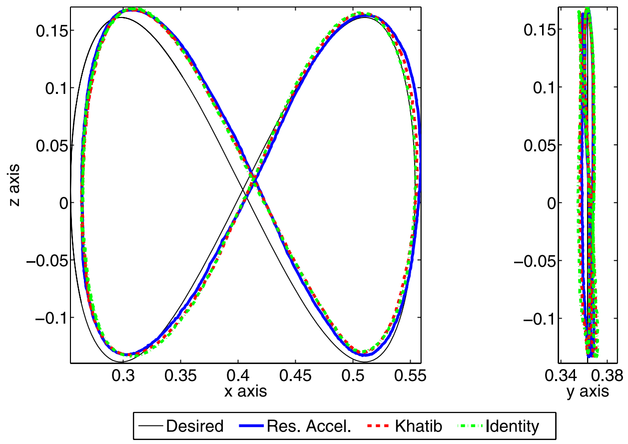

Fig. 3 This figure shows the three end-effector trajectory controllers tracking a "figure eight (8)" pattern at 8 seconds per cycle. On the *left* is the x-z plane with the y-z plane on the *right*. All units are in meters

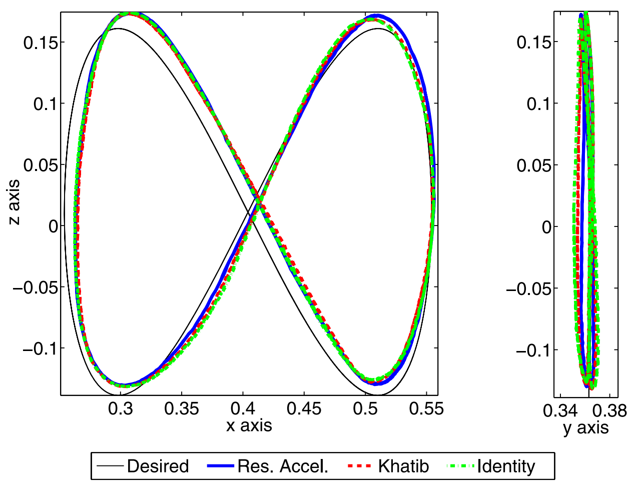

Fig. 4 The same three controllers tracking the same "figure eight (8)" pattern at a faster pace of 4 seconds per cycle. The labels and units remain the same as in Fig. 3

two different speeds. In order to stabilize the null-space trajectories, we choose a PD control law in joint space which pulls the robot towards a fixed rest posture, $\mathbf{q}_{\text{rest}}$, given by

$$\mathbf{u}_0 = \mathbf{M}(\mathbf{K}_{P0}(\mathbf{q}_{\text{rest}} - \mathbf{q}) - \mathbf{K}_{D0}\dot{\mathbf{q}}).$$

Additionally we apply gravity, centrifugal and Coriolis force compensation, such that $\mathbf{u}_1 = \mathbf{u}_0 + \mathbf{C} + \mathbf{G}$. For consistency, all three controllers are assigned the same gains both for the task and joint space stabilization.

Figure 3 shows the end-point trajectories of the three controllers in a slow pattern of eight seconds per cycle "figure-eight (8)". Figure 4 shows a faster pace of four seconds per cycle. All three controllers have similar end-point trajectories and result in fairly accurate task achievement. Each one has an offset from the desired trajectory (thin black line), primarily due to the imperfect dynamics model of the robot. The root mean squared errors (RMS) between the actual and the desired trajectory in task-space for each of the controllers are shown in Table 1.

Table 1 This table shows the root mean squared error results of the tracking achieved by the different control laws

Metric	Slow RMS error [m]	Fast RMS error [m]
$\mathbf{N} = \mathbf{M}^{-2}$	0.0122	0.0130
$\mathbf{N} = \mathbf{M}^{-1}$	0.0126	0.0136
$\mathbf{N} = \mathbf{I}$	0.0130	0.0140

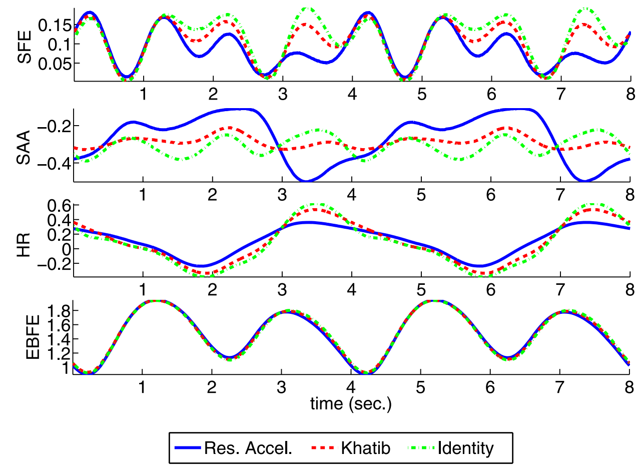

Fig. 5 Joint space trajectories for the four major degrees of freedom, i.e., shoulder flexion-extension (SFE), shoulder adduction-abduction (SAA), humeral rotation (HR) and elbow flexion-extension (EBFE), are shown here. Joint angle units are in radians. The labels are identical to the ones in Fig. 3

While the performance of the three controllers is very similar in task space, we did notice that the resolved-acceleration kinematic controller ($\mathbf{N} = \mathbf{M}^{-2}$) had a slight advantage. The reason for this performance difference is most likely due to errors in the dynamics model: the effect of these errors is amplified by the inversion of the mass matrix in the control laws given in (30), (31) while the decoupling of the dynamics and kinematics provided by the controller in (29) can be favorable as the effect of modeling errors is not increased. More accurate model parameters of the manipulator's rigid body dynamics would result in a reduction of the difference between these control laws (observable in Figs. 3 and 4) as we have confirmed in simulations.

Figure 5 illustrates how the joint space trajectories appear for the fast cycle. Although end-point trajectories were very similar, joint space trajectories differ significantly due to the different optimization criteria of each control law, which emphasizes the importance of the choice of the metric $\mathbf{N}$.

5 Conclusion

In this paper we presented an optimal control framework which allows the development of a unified approach for deriving a number of different robot control laws for rigid body dynamics systems. We demonstrated how we can make use of both the robot model and a task description in order to create control laws which are optimal with respect to the squared motor command under a particular metric while *perfectly* fulfilling the task *at each instant of time*. We have discussed how to realize stability both in task as well as in joint-space for this framework.

Building on this foundation, we demonstrated how a variety of control laws—which on first inspection appear rather unrelated to one another—can be derived using this straightforward framework. The covered types of tasks include joint-space trajectory control for both fully actuated and overactuated robots, end-effector trajectory control, impedance and hybrid control.

The implementation of three of the end-effector trajectory control laws resulting from our unified framework on a real-world Sarcos Master Arm robot was carried out as an empirical evaluation. As expected, the behavior in task space is very similar for all three control laws; yet, they result in very different joint-space behaviors due to the different cost functions resulting from the different metrics of each control law.

The major contribution of this paper is the unified framework that we have developed. It allows a derivation of a variety of previously known controllers, and promises easy development of a host of novel ones, in particular control laws with additional constraints. The particular controllers reported in this paper were selected primarily for illustrating the applicability of this framework and demonstrating its strength in unifying different control algorithms using a common building principle. In future work, we will evaluate how this framework can yield a variety of new control laws for underactuated tasks and robots, for non-holonomic robots and tasks, and for robots with flexible links and joints.

Acknowledgements 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, the ERATO Kawato Dynamic Brain Project funded by the Japanese Science and Technology Agency, and the ATR Computational Neuroscience Laboratories.

References

Arimoto, S. (1996). *Control theory of nonlinear mechanical systems: a passivity-based and circuit-theoretic approach. Oxford engineering science series*. Oxford: Oxford University Press.

Baillieul, J., & Martin, D. P. (1990). Resolution of kinematic redundancy. In *Proceedings of symposium in applied mathematics* (Vol. 41, pp. 49–89), San Diego, May 1990. Providence: American Mathematical Society.

Bruyninckx, H., & Khatib, O. (2000). Gauss' principle and the dynamics of redundant and constrained manipulators. In *Proceedings of the 2000 IEEE international conference on robotics & automation* (pp. 2563–2569).

Chung, W., Chung, W., & Youm, Y. (1993). Null torque based dynamic control for kinematically redundant manipulators. *Journal of Robotic Systems*, *10*(6), 811–834.

Craig, J. (1989). *Introduction to robotics: mechanics and control*. New York: Prentice Hall.

Hanafusa, H., Yoshikawa, T., & Nakamura, Y. (1981). Analysis and control of articulated robot with redundancy. In *Proceedings of IFAC symposium on robot control* (Vol. 4, pp. 1927–1932).

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

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

Isidori, A. (1995). *Nonlinear control systems*. Berlin: Springer.

Kazerounian, K., & 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*, *3*(1), 43–53.

Khatib, O., Sentis, L., Park, J., & Warren, J. (2004). Whole body dynamic behavior and control of human-like robots. *International Journal of Humanoid Robotics*, *1*(1), 29–43.

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

Minamide, N., & Nakamura, K. (1969). *Minimum error control problem in Banach space*. Research Report of Automatic Control Lab 16, Nagoya University, Nagoya, Japan.

Nakamura, Y. (1991). *Advanced robotics: redundancy and optimization*. Reading: Addison-Wesley.

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

Nakanishi, J., Cory, R., Mistry, M., Peters, J., & Schaal, S. (2005). Comparative experiments on task space control with redundancy resolution. In *IEEE international conference on intelligent robots and systems (IROS 2005)*, Edmonton, Alberta, Canada.

Park, J., Chung, W.-K., & Youm, Y. (1995). Specification and control of motion for kinematically redundant manipulators. In *Proceedings international conference of robotics systems*.

Park, J., Chung, W.-K., & Youm, Y. (2002). Characterization of instability of dynamic control for kinematically redundant manipulators. In *Proceedings of the IEEE international conference on robotics and automation*.

Pratt, J., & Pratt, G. (1998). Intuitive control of a planar bipedal walking robot. In *Proceedings of the 1998 IEEE international conference on robotics & automation* (pp. 1024–2021).

Samson, C., Borgne, M. L., & Espiau, B. (1991). *Oxford engineering science series. Robot control: the task function approach*. Oxford: Oxford University Press.

Sciavicco, L., & Siciliano, B. (1996). *Modeling and control of robot manipulators*. New York: McGraw-Hill.

Sentis, L., & Khatib, O. (2004). A prioritized multi-objective dynamic controller for robots in human environments. In *International conference on humanoid robots*, Los Angeles, USA. New York: IEEE/RSJ.

Sentis, L., & Khatib, O. (2005). Control of free-floating humanoid robots through task prioritization. In *International conference on robotics and automation* (pp. 1730–1735), Barcelona, Spain. New York: IEEE.

Siciliano, B., & Slotine, J. (1991). A general framework for managing multiple tasks in highly redundant robotic systems. In *International conference on advanced robotics* (pp. 1211–1216), Pisa, Italy. New York: IEEE.

Spo (1984). *On pointwise optimal control strategies for robot manipulators*. Princeton: Princeton University Press.

Spong, M., Thorp, J., & Kleinwaks, J. (1986). The control of robot manipulators with bounded input. *IEEE Transactions on Automatic Control*, *31*(6), 483–490.

Suh, K. C., & Hollerbach, J. M. (1987). Local versus global torque optimization of redundant manipulators. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 619–624).

Udwadia, F. E. (2003). A new perspective on tracking control of nonlinear structural and mechanical systems. *Proceedings of the Royal Society of London Series A*, *2003*, 1783–1800.

Udwadia, F. E., & Kalaba, R. E. (1996). *Analytical dynamics: a new approach*. Cambridge: Cambridge University Press.

Wahba, G., & Nashed, M. Z. (1973). The approximate solution of a class of constrained control problems. In *Proceedings of the sixth Hawaii international conference on systems sciences*, Hawaii.

Wit, C. A. C. D., Siciliano, B., & Bastin, G. (1996). *Theory of robot control*. Telos: Springer.

Yamane, K., & Nakamura, Y. (2003). Natural motion animation through constraining and deconstraining at will constraining and deconstraining at will. *IEEE Transactions on Visualization and Computer Graphics*, *9*(3).

Yoshikawa, T. (1990). *Foundations of robotics: analysis and control*. Cambridge: MIT Press.

Jan Peters heads the Robot and Reinforcement Learning Lab (R2L2) at the Max-Planck Institute for Biological Cybernetics (MPI) in April while being an invited researcher at the Computational Learning and Motor Control Lab at the University of Southern California (USC). Before joining MPI, he graduated from University of Southern California with a Ph.D. in Computer Science in March 2007. Jan Peters studied Electrical Engineering, Computer Science and Mechanical Engineering. He holds two German M.Sc. degrees in Informatics and in Electrical Engineering (from Hagen University and Munich University of Technology) and two M.Sc. degrees in Computer Science and Mechanical Engineering from USC. During his graduate studies, Jan Peters has been a visiting researcher at the Department of Robotics at the German Aerospace Research Center (DLR) in Oberpfaffenhofen, Germany, at Siemens Advanced Engineering (SAE) in Singapore, at the National University of Singapore (NUS), and at the Department of Humanoid Robotics and Computational Neuroscience at the Advanced Telecommunication Research (ATR) Center in Kyoto, Japan. His research interests include robotics, nonlinear control, machine learning, and motor skill learning.

Michael Mistry received a B.S.E. degree in Electrical Engineering and Mechanical Engineering and Applied Mechanics from the University of Pennsylvania in 1998, and an M.A in Cognitive and Neural Systems from Boston University in 2002. In 2003, he began his Ph.D. studies in Computer Science at the University of Southern California (USC) under guidance of Stefan Schaal. Since then, he has since received an M.S. in Aerospace and Mechanical Engineering (2005) and an M.S. in Computer Science (2007) from USC. His research currently focuses on the representation, learning, and control of dexterous motor skills in humans and humanoid robots.

Firdaus Udwadia works in the areas of Analytical Dynamics, Nonlinear Dynamical Systems, Structural and Nonlinear Control, and Applied Mathematics. He received his Ph.D. from the California Institute of Technology. He has published more than 170 papers in refereed journals, and is the co-author of the book Analytical Dynamics: a new approach, published by Cambridge University Press. He is a Fellow of the American Society of Mechanical Engineers, and also a Fellow of the American Institute of Aeronautics and Astronautics. In 2006, he received the Outstanding Technical Contributions Award from the American Society of Civil Engineers.

Jun Nakanishi received the B.E. and M.E. degrees both in mechanical engineering from Nagoya University, Nagoya, Japan, in 1995 and 1997, respectively. He received a Ph.D. degree in engineering from Nagoya University in 2000. He also studied in the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor, USA, from 1995 to 1996. He was a Research Associate at the Department of Micro System Engineering, Nagoya University, from 2000 to 2001, and was a presidential postdoctoral fellow at the Computer Science Department, University of Southern California, Los Angeles, USA, from 2001 to 2002. He joined ATR Human Information Science Laboratories, Kyoto, Japan, in 2002. He is currently a researcher at ATR Computational Neuroscience Laboratories and with the Computational Brain Project, ICORP, Japan Science and Technology Agency. His research interests include motor learning and control in robotic systems. He received the IEEE ICRA 2002 Best Paper Award.

Stefan Schaal is an Associate Professor at the Department of Computer Science and the Neuroscience Program at the University of Southern California, and an Invited Researcher at the ATR Human Information Sciences Laboratory in Japan, where he held an appointment as Head of the Computational Learning Group during an international ERATO project, the Kawato Dynamic Brain Project (ERATO/JST). Before joining USC, Dr. Schaal was a postdoctoral fellow at the Department of Brain and Cognitive Sciences and the Artificial Intelligence Laboratory at MIT, an Invited Researcher at the ATR Human Information Processing Research Laboratories in Japan, and an Adjunct Assistant Professor at the Georgia Institute of Technology and at the Department of Kinesiology of the Pennsylvania State University. Dr. Schaal's research interests include topics of statistical and machine learning, neural networks, computational neuroscience, functional brain imaging, nonlinear dynamics, nonlinear control theory, and biomimetic robotics. He applies his research to problems of artificial and biological motor control and motor learning, focusing on both theoretical investigations and experiments with human subjects and anthropomorphic robot equipment.