A Unifying Methodology for the Control of Robotic Systems

Jan Peters*, Michael Mistry*, Firdaus Udwadia*, Rick Cory†‡, Jun Nakanishi†‡, and Stefan Schaal*‡
* *University of Southern California, Los Angeles, CA 90089, USA*
† *ICORP, Japan Science and Technology Agency, Kyoto, 619-0288, Japan*
‡ *ATR Computational Neuroscience Laboratories, Kyoto, 619-0288, Japan*
{*jrpeters, mmistry, fudwdia,*}*@usc.edu*, {*cory, jun*}*@atr.jp, sschaal@usc.edu*

***Abstract*— Recently, [1] suggested to derive tracking controllers for mechanical systems using a generalization of Gauss' principle of least constraint. This method allows us to reformulate control problems as a special class of optimal control. We take this line of reasoning one step further and demonstrate that well-known and also several 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 the derived controllers. We believe that 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.**

***Index Terms*— Non-linear control, robot control, tracking control.**

I. Introduction

Despite the progress in robotics over the last decades, only a few general building principles for designing robot controllers have been obtained. To date, robot controllers are often derived from insights such as the reduction of the controlled system onto a linear system by linearization or by inversion of the dynamics of the robot [2], [3]. While this approach is viable for many problems, it is in a sense limiting because it ignores potentially useful properties of the inherent nonlinearities. Only few statements can be made about the quality of such controllers underlying cost functions which sometimes cannot even be obtained. General optimal control techniques on the other hand are often not applicable as a closed-form solution usually does not exist and numerical solutions for high-dimensional systems are often prohibitively expensive in terms of computations due to the 'Curse of Dimensionality' [4], [5].

Recently, a novel way of thinking about tracking control of mechanical systems was suggested in [1] inspired by results from analytical dynamics with constrained motion. The major insight in [1] 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] [6] in order to derive a controller. As it is outlined already in [1], this insight leads to a specialized 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 optimal 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 a step further and show that it can be used as a general way of solving robotic control problems which unifies many approaches to robot control found in the literature to date. We can demonstrate 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 the joint space further considerations may apply. To demonstrate the feasibility of our framework, we evaluate a few derived controllers on a robot arm with a simple end-effector tracking task.

This paper is organized as follows: firstly, a novel optimal control framework based on [1] 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 robot arm.

II. A Novel Methodology for the Control of Robotic Systems

A variety of robot control problems can be motivated by the desire to achieve a task perfectly while minimizing the squared motor commands. In this section, we will show how the robot dynamics and the control problem can be brought into a general form which will then allow us to compute the optimal control with respect to a desired metric. We will augment this framework so that we can assure stability both in the joint space of the robot as well as in the task space of the problem.

A. Formulating Robot Control Problems

In order to formulate our framework, we will introduce the specifics of the assumed underlying robot model and show how a task can be specified.

Robot Model: We assume the well-known rigid-body dynamics model of manipulator robot arms with n degrees of freedom given by the equation

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

[1]Gauss' principle of least constraint [6] 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.

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 gravity [2], [3]. At many points we will write the dynamics equations by $\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{u}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}})$ where $\mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}) = -\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{G}(\mathbf{q})$ as specified in [1], [6]. We assume that an accurate model of our robot system is available.

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

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

where $\mathbf{h} \in \mathbb{R}^k$ where the dimensionality is arbitrary. 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 Section III-A. We consider only tasks wherein equation (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 equation (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 Section III, we will always give the task description first in the general form in Equation (2), and then derive the resulting controller using the form which is the linear in accelerations, given in Equation (3).

B. Optimal Control Framework

Let us assume that we are given a robot model and a constraint description of the task as described in the previous section. In this case, we can describe 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, $\mathbf{A}\ddot{\mathbf{q}} = \mathbf{b}$, holds at all times. Second, we intend to minimize the control force with respect to some given metric, i.e., $J(t) = \mathbf{u}^T\mathbf{N}(t)\mathbf{u}$, at *each* instant of time. The solution to this can be derived from a generalization of Gauss' principle as originally suggested in [1]. We formalize this here in the following theorem.

Theorem 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

$$\mathbf{A}\ddot{\mathbf{q}} = \mathbf{b}, \tag{5}$$

is given by

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

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

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

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 theorem holds. ∎

The choice of the metric $\mathbf{N}$ plays a central role, because it determines the type of solution. Often, we require a solution which has a kinematic interpretation; such a solution is usually given by a metric like $\mathbf{N} = \mathbf{M}^{-2}$. 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. In Section III, we will see how the choice of $\mathbf{N}$ results in several different controllers.

Note that this framework has been suggested in general in [1], [6], and the special case with a metric $\mathbf{N} = \mathbf{M}^{-1}$ has been presented in [7] with respect to robot control.

C. Stability Analysis

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 this framework slightly and show how we can ensure stability. This modification will be introduced in Section II-C.1. Furthermore, we realize that the case of underconstrained tasks, i.e., tasks where some degrees of freedom of the robot are redundant for the given task, can cause undesired properties or even instability in joint-space; we will treat this problem in Section II-C.2.

1) Stability in Task Space : Up to this point, we have assumed that we always have perfect initial conditions and that we know the robot model perfectly. However, we have to compensate for the fact that we might not be sitting perfectly on the trajectory from the start or that we might get disturbed out of this trajectory. [1] suggested that this 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 Section III-B) but often it suffices to choose it to be state vector (for example for joint-space trajectory control as in Section III-A). In the case of holomonic 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 as suggested in [1] 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 $\mathbf{f_h}$ appropriately.

If we have a task-space stabilization as discussed in the paragraph above, we can assure that the control law is stable in task space at least in a region near about the desired trajectory. We show this in the following theorem.

Theorem 2: If we can assure the attractor property of the task $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{0}$, or equivalently, $\mathbf{A}\ddot{\mathbf{q}} = \mathbf{b}$, and if our robot model is accurate, it is straightforward to show that the controller is stable in task space.

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

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

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

An analysis of the stability properties of the derived controllers when an imperfect robot model is given will be part of future work.

2) Stability in Joint Space : While the stability in task space is fairly well-understood, it is not immediately clear whether the control law is stable in joint-space. It is fairly straightforward to create a counter-example. Example 1, illustrates a situation where a redundant robot arm is stable in task-space while unstable in joint-space.

Example 1: Let us assume the simplest possible robot, a prismatic robot with two horizontal, parallel links. 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 $\mathbf{F} = \mathbf{0}$. 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 this obviously is stable in task-space, the initial condition $q_1(t_0) = x_{\text{des}}(t_0) - q_2(t_0)$ would result into both $q_i(t)$'s diverging into opposite directions. The reason for this is obvious: the effort of stabilizing in joint space is not task relevant – any solution stabilizing this problem in joint-space would increase the cost.

From this example, we see that the general framework does not always suffice but that it has to be modified so that we can incorporate a minimal control which in practice stabilizes the robot without affecting the task achievement. One possibility to stabilize the robot in joint-space is by having a joint-space motor command $\mathbf{u}_1$ as an additional component of the 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. We can show that the fulfillment of the task $\mathbf{A}\ddot{\mathbf{q}} = \mathbf{b}$ by the controller is not affected by the choice of the joint-space control law $\mathbf{u}_1$.

Theorem 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-stabilizing control law acts in the null-space of the task.

Proof: When determining $\mathbf{u}_2$, we consider $\mathbf{u}_1$ to be part of our forces, i.e., we have $\tilde{\mathbf{F}} = \mathbf{F} + \mathbf{u}_1$. We obtain $\mathbf{u}_2 = \mathbf{N}^{-1/2}\left(\mathbf{AM}^{-1}\mathbf{N}^{-1/2}\right)^{+}\left(\mathbf{b} - \mathbf{AM}^{-1}\tilde{\mathbf{F}}\right)$ using Theorem 1. By reordering the complete control law $\mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2(\mathbf{u}_1)$, we obtain

$$\begin{aligned}
\mathbf{u} &= \mathbf{u}_1 + \mathbf{N}^{-1/2}\left(\mathbf{AM}^{-1}\mathbf{N}^{-1/2}\right)^{+}\left(\mathbf{b} - \mathbf{AM}^{-1}(\mathbf{F} + \mathbf{u}_1)\right), \\
&= \mathbf{N}^{-1/2}\left(\mathbf{AM}^{-1}\mathbf{N}^{-1/2}\right)^{+}(\mathbf{b} - \mathbf{AM}^{-1}\mathbf{F}) \\
&\quad + (\mathbf{I} - \mathbf{N}^{-1/2}\left(\mathbf{AM}^{-1}\mathbf{N}^{-1/2}\right)^{+}\mathbf{AM}^{-1})\mathbf{u}_1, \\
&= \mathbf{N}^{-1/2}\left(\mathbf{AM}^{-1}\mathbf{N}^{-1/2}\right)^{+}(\mathbf{b} - \mathbf{AM}^{-1}\mathbf{F}) \qquad (12) \\
&\quad + \mathbf{N}^{-1/2}[\mathbf{I} - \left(\mathbf{AM}^{-1}\mathbf{N}^{-1/2}\right)^{+}(\mathbf{AM}^{-1}\mathbf{N}^{-1/2})]\mathbf{N}^{1/2}\mathbf{u}_1,
\end{aligned}$$

The task space is defined by $\mathbf{N}^{-1/2}\left(\mathbf{AM}^{-1}\mathbf{N}^{-1/2}\right)^{+}$, and that the matrix $\mathbf{N}^{-1/2}[\mathbf{I} - \left(\mathbf{AM}^{-1}\mathbf{N}^{-1/2}\right)^{+}(\mathbf{AM}^{-1}\mathbf{N}^{-1/2})]$ makes sure that the joint-space control law and the task space control law are $\mathbf{N}$-orthogonal. ■

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, 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

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

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

III. Robot Control Laws

The previously described framework offers a variety of applications in robotics – we will only be able to give 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 building principles. In this section, we show how a vast 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.

A. Joint-Space Trajectory Control

The first control problem we attempt to tackle 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].

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 Section II-C.1, yielding

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

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

$$\mathbf{A} = \mathbf{I}, \quad (16)$$
$$\mathbf{b} = \ddot{\mathbf{q}}_d + \mathbf{K}_D(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) - \mathbf{K}_P(\mathbf{q}_d - \mathbf{q}). \quad (17)$$

In this case, we can use Theorem 1 and 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

$$\begin{aligned}
\mathbf{u} &= \mathbf{u}_1 + \mathbf{N}^{-1/2}\left(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2}\right)^{+}(\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}(\mathbf{F} + \mathbf{u}_1)),\\
&= \mathbf{M}^{1/2}\left(\mathbf{M}^{-1/2}\right)^{-1}(\ddot{\mathbf{q}}_d + \mathbf{K}_D(\dot{\mathbf{q}}_d - \dot{\mathbf{q}})\\
&\quad + \mathbf{K}_P(\mathbf{q}_d - \mathbf{q}) - \mathbf{M}^{-1}(-\mathbf{C} - \mathbf{G})),\\
&= \mathbf{M}(\ddot{\mathbf{q}}_d + \mathbf{K}_D(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_P(\mathbf{q}_d - \mathbf{q})) \quad (18)\\
&\quad + \mathbf{C} + \mathbf{G}. \quad (19)
\end{aligned}$$

Note that 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. It turns out that this a well-known control law, i.e., the **Inverse Dynamics Control Law** [2], [3].

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

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

$$\mathbf{D}\mathbf{u} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}), \quad (20)$$

where $\mathbf{D}$ depends on our type of muscle. In the simplest model for a two degrees of freedom robot it could be given by

$$\mathbf{D} = \begin{bmatrix} -l & +l & 0 & 0 \\ 0 & 0 & -l & +l \end{bmatrix}. \quad (21)$$

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 have expressed the trajectory like in previous examples, and we obtain the following controller

$$\begin{aligned}
\mathbf{u} &= \tilde{\mathbf{M}}^{1/2}\left(\mathbf{A}\tilde{\mathbf{M}}^{-1/2}\right)^{+}\left(\mathbf{b} - \mathbf{A}\tilde{\mathbf{M}}^{-1}\tilde{\mathbf{F}}\right), \quad (22)\\
&= \mathbf{D}^{+}\mathbf{M}(\ddot{\mathbf{q}}_d + \mathbf{K}_D(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) - \mathbf{K}_P(\mathbf{q}_d - \mathbf{q})) \quad (23)\\
&\quad + \mathbf{D}^{+}(\mathbf{C} + \mathbf{G}). \quad (24)
\end{aligned}$$

While immidiately intuitive, it is somehow surprising that this particular controller should fall out of the presented framework. Due to a lack of hardware and realistic simulators, we cannot evaluate this approach within the scope of this paper.

B. 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, the same cannot be said about end-effector control where the position $\mathbf{x}(t)$ of the end-effector is moved along some given trajectory. This problem is generically more difficult as the choice of the metric $\mathbf{N}$ determines the type of the solution 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 context, we will show how to derive different approaches to end-effector control from the presented framework; this yields 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 Section II-C.1, yielding

$$(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_d) + \mathbf{K}_D(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) + \mathbf{K}_P(\mathbf{x} - \mathbf{x}_d) = 0, \quad (25)$$

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

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad (26)$$

$$\ddot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}}. \tag{27}$$

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

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

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

$$\mathbf{A}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{J}, \tag{29}$$

$$\mathbf{b}(\mathbf{q},\dot{\mathbf{q}}) = \ddot{\mathbf{x}}_d + \mathbf{K}_D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{K}_P(\mathbf{x}_d - \mathbf{x}) - \dot{\mathbf{J}}\dot{\mathbf{q}}. \tag{30}$$

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 invertible the resulting controller depends on the chosen metric and joint-space control law.

1) Separation of Kinematics and Dynamics Control: The choice of the metric $\mathbf{N}$ determines the type of the task. 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 Theorem 1, we obtain a control law

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

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

$$\begin{aligned}\mathbf{u} &= \mathbf{M}\mathbf{J}^{+}(\ddot{\mathbf{x}}_d + \mathbf{K}_D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{K}_P(\mathbf{x}_d - \mathbf{x}) - \dot{\mathbf{J}}\dot{\mathbf{q}}) \\ &\quad + \mathbf{M}(\mathbf{I} - \mathbf{J}^{+}\mathbf{J})\mathbf{M}^{-1}\mathbf{u}_0 + \mathbf{C} + \mathbf{G}.\end{aligned} \tag{31}$$

This control law is the combination of a **resolved-acceleration kinematic controller** [2], [8] with a model-based controller and an additional null-space term. Similar controllers have been introduced in [9]–[12]. 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 Section IV.

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., it is dynamically consistent and therefore the resulting control force can be re-interpreted as mechanical structures (e.g., springs and dampers) attached to the end-effector. Again, we apply Theorem 1, and by defining $\tilde{\mathbf{F}} = \mathbf{F} + \mathbf{u}_1$ obtain the control law

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

It turns out that this is another well-known control law suggest in [13] with an additional null-space term. This control-law is used in [1] and is especially interesting as it has a clear physical interpretation [1], [6], [7]: 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

$$\begin{aligned}\mathbf{u} &= \mathbf{J}^{T}\left(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{T}\right)^{-1}(\ddot{\mathbf{x}}_d + \mathbf{K}_D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) \\ &\quad + \mathbf{K}_P(\mathbf{x}_d - \mathbf{x}) - \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}}) + \mathbf{C} + \mathbf{G} \\ &\quad + \mathbf{M}(\mathbf{I} - \mathbf{M}^{-1}\mathbf{J}^{T}\left(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^{T}\right)^{-1}\mathbf{J})\mathbf{M}^{-1}\mathbf{u}_0.\end{aligned} \tag{32}$$

The compensation of the forces in joint-space is often desirable for this metric in order to have full control over the resolution of the redundancy as the gravity compensation in task space often results into strange postures.

3) Further Metrics: Using the identity matrix as metric, i.e., $\mathbf{N} = \mathbf{I}$, punishes the squared motor command without reweighting. 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

$$\begin{aligned}\mathbf{u} &= \left(\mathbf{J}\mathbf{M}^{-1}\right)^{+}(\ddot{\mathbf{x}}_d + \mathbf{K}_D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) \\ &\quad + \mathbf{K}_P(\mathbf{x}_d - \mathbf{x}) - \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{J}\mathbf{M}^{-1}(\mathbf{C} + \mathbf{G})) \\ &\quad + (\mathbf{I} - \left(\mathbf{J}\mathbf{M}^{-1}\right)^{+}\mathbf{J}\mathbf{M}^{-1})\mathbf{u}_1.\end{aligned} \tag{33}$$

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 $\mathbf{N} = \mathrm{diag}(\tau_{\max,1}^{-1}, \ldots, \tau_{\max,n}^{-1})$.

C. Controlling Constrained Manipulators: Impedance & Hybrid Control

Contact with outside objects fundamentally 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

$$\mathbf{u} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \mathbf{J}^{T}\mathbf{F}_C. \tag{34}$$

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.

1) Impedance Control : In impedance control, we want the robot to simulate the behavior of a mechanical system such as

$$\mathbf{M}_d(\ddot{\mathbf{x}}_d - \ddot{\mathbf{x}}) + \mathbf{D}_d(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{P}_d(\mathbf{x}_d - \mathbf{x}) = \mathbf{F}_C, \tag{35}$$

where $\mathbf{M}_d \in \mathbb{R}^{6\times 6}$ denotes the mass matrix of the desired system, $\mathbf{F}_C \in \mathbb{R}^6$ denotes the measured external forces exerted onto the system, $\mathbf{D}_d \in \mathbb{R}^6$ denotes the desired damping, and $\mathbf{P}_d \in \mathbb{R}^6$ denotes the gains towards the desired position. Using Equation (27) from Section III-B, we see that this can simply be brought in the standard form for tasks by

$$\begin{aligned}\mathbf{M}_d\mathbf{J}\ddot{\mathbf{q}} = \mathbf{F}_C - \mathbf{M}_d\ddot{\mathbf{x}}_d - \mathbf{D}_d(\dot{\mathbf{x}}_d - \mathbf{J}\dot{\mathbf{q}}) \\ - \mathbf{P}_d(\mathbf{x}_d - \mathbf{f}(\mathbf{q})) - \mathbf{M}_d\dot{\mathbf{J}}\dot{\mathbf{q}},\end{aligned} \tag{36}$$

after dropping all indices. From this we can infer the task description given by

$$\mathbf{A} = \mathbf{M}_d\mathbf{J}, \tag{37}$$

$$\begin{aligned}\mathbf{b} = \mathbf{F}_C - \mathbf{M}_d\ddot{\mathbf{x}}_d - \mathbf{D}_d(\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}_d) \\ - \mathbf{P}_d(\mathbf{f}(\mathbf{q}) - \mathbf{x}_d) - \mathbf{M}_d\dot{\mathbf{J}}\dot{\mathbf{q}}.\end{aligned} \tag{38}$$

A major question in this context is the choice of the correct joint-space control law $\mathbf{u}_1(\mathbf{q}, \dot{\mathbf{q}})$, and the right metric to achieve such tasks.

a) Separation of both Systems through Kinematics: Similar as in end-effector control, a practical metric is $\mathbf{N} = \mathbf{M}^{-2}$ as this basically separates both dynamic systems into two separate ones as it will become apparent in this section. 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

$$\begin{aligned}\mathbf{u} &= \mathbf{u}_1 + \mathbf{N}^{-1/2}\left(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2}\right)^{+}\left(\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}(\mathbf{F} + \mathbf{u}_1)\right), \\ &= \mathbf{M}\left(\mathbf{M}_d\mathbf{J}\right)^{+}(\mathbf{F}_C - \mathbf{M}_d\ddot{\mathbf{x}}_d - \mathbf{D}_d(\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}_d) \\ &\quad - \mathbf{P}_d(\mathbf{f}(\mathbf{q}) - \mathbf{x}_d) - \mathbf{M}_d\dot{\mathbf{J}}\dot{\mathbf{q}}) + \mathbf{C} + \mathbf{G} \\ &\quad + (\mathbf{I} - \mathbf{M}\left(\mathbf{M}_d\mathbf{J}\right)^{+}\mathbf{M}_d\mathbf{J}\mathbf{M}^{-1})\mathbf{u}_0.\end{aligned} \tag{39}$$

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

$$\begin{aligned}\mathbf{u} = \mathbf{M}\mathbf{J}^{+}\mathbf{M}_d^{-1}(\mathbf{F}_C - \mathbf{M}_d\ddot{\mathbf{x}}_d - \mathbf{D}_d(\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}_d) \\ - \mathbf{P}_d(\mathbf{f}(\mathbf{q}) - \mathbf{x}_d)) - \mathbf{M}\mathbf{J}^{+}\dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{C} + \mathbf{G} \\ + \mathbf{M}(\mathbf{I} - \mathbf{J}^{+}\mathbf{J})\mathbf{M}^{-1}\mathbf{u}_0.\end{aligned} \tag{40}$$

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 clarifies the previous remark: 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.

b) Dynamically Consistent Combination: Similar as in end-effector control, a practical metric is $\mathbf{N} = \mathbf{M}^{-1}$ which combines both dynamic systems into a big one 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 results into the control law

$$\begin{aligned}\mathbf{u} &= \mathbf{u}_1 + \mathbf{N}^{-1/2}\left(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2}\right)^{+}\left(\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}(\mathbf{F} + \mathbf{u}_1)\right), \\ &= \mathbf{u}_1 + \mathbf{J}^T\left(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\right)^{-1}\left(\mathbf{b} - \mathbf{A}\mathbf{M}^{-1}(\mathbf{F} + \mathbf{u}_1)\right), \\ &= \mathbf{M}^{1/2}\left(\mathbf{M}_d\mathbf{J}\mathbf{M}^{-1/2}\right)^{+}(\mathbf{F}_C - \mathbf{D}_d(\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}_d) \\ &\quad - \mathbf{P}_d(\mathbf{f}(\mathbf{q}) - \mathbf{x}_d) - \mathbf{M}_d\dot{\mathbf{J}}\dot{\mathbf{q}}) + \mathbf{C} + \mathbf{G} \\ &\quad + (\mathbf{I} - \mathbf{M}\left(\mathbf{M}_d\mathbf{J}\right)^{+}\mathbf{M}_d\mathbf{J}\mathbf{M}^{-1})\mathbf{u}_0.\end{aligned} \tag{41}$$

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

$$\begin{aligned}\mathbf{u} = \mathbf{J}^T\left(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\right)^{-1}\mathbf{M}_d^{-1}(\mathbf{F}_C - \mathbf{D}_d(\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}_d) \\ - \mathbf{P}_d(\mathbf{f}(\mathbf{q}) - \mathbf{x}_d)) - \mathbf{M}\mathbf{J}^{+}\dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{C} + \mathbf{G} \\ + (\mathbf{I} - \mathbf{M}\mathbf{J}^{+}\mathbf{J}\mathbf{M}^{-1})\mathbf{u}_0.\end{aligned} \tag{42}$$

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

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, common hybrid control approaches are essentially similar to our introduced framework [3]. 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 holomonic 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-holomic constraints $\phi(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathbf{0}$ as long as they are given in the form

$$\mathbf{A}_\phi(\mathbf{q}, \dot{\mathbf{q}})\ddot{\mathbf{q}} = \mathbf{b}_\phi(\mathbf{q}, \dot{\mathbf{q}}). \tag{43}$$

$\mathbf{A}_\phi$, $\mathbf{b}_\phi$ depend on the type of the constraint, e.g., for scleronomic, holomonic 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 [3]. 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

$$\mathbf{u}_1 = \mathbf{C} + \mathbf{G} + \mathbf{J}_\phi^T\mathbf{F}_d. \tag{44}$$

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

$$\mathbf{u} = \mathbf{u}_1 + \mathbf{N}^{-1/2}\left(\mathbf{A}_\phi\mathbf{M}^{-1}\mathbf{N}^{-1/2}\right)^{+}\left(\mathbf{b}_\phi - \mathbf{A}_\phi\mathbf{M}^{-1}(\mathbf{F} + \mathbf{u}_1)\right), \tag{45}$$

$$\begin{aligned}&= \mathbf{C} + \mathbf{G} + \mathbf{N}^{-1/2}\left(\mathbf{A}_\phi\mathbf{M}^{-1}\mathbf{N}^{-1/2}\right)^{+}\mathbf{b}_\phi \\ &\quad + \mathbf{N}^{-1/2}(\mathbf{I} - \left(\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2}\right)^{+}\mathbf{A}\mathbf{M}^{-1}\mathbf{N}^{-1/2})\mathbf{N}^{1/2}\mathbf{J}_\phi^T\mathbf{F}_d.\end{aligned} \tag{46}$$

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

IV. Evaluations

The main contribution of this paper is the unifying methodology for deriving robot controllers. In order to demonstrate the framework's feasibility for providing implementable controllers for real robots, we have chosen a few of the controllers

(a) Simulated Robot Arm (b) SARCOS Master Arm

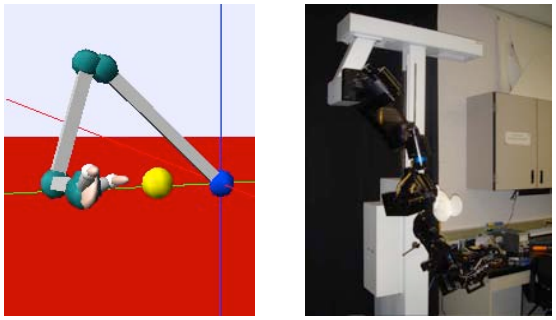

Fig. 1. Setups in which we evaluate the designed controllers: (a) a physical simulation of the SARCOS Master Arm, (b), the robot arm.

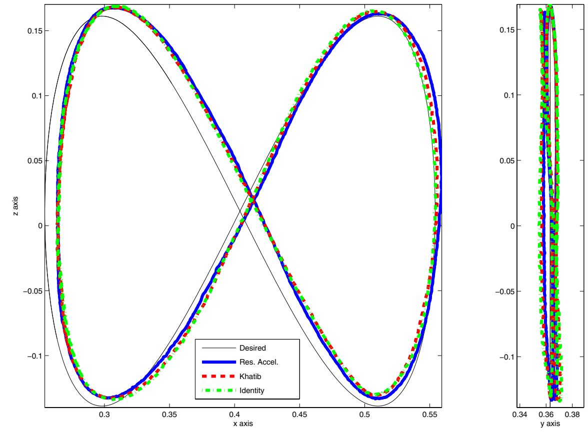

Fig. 2. This figure shows the three end-effector trajectory controllers tracking a “figure eight (8)” pattern at 8 seconds per cycle. On the left is the x-z plane with the y-z plane on the right. All units are in meters.

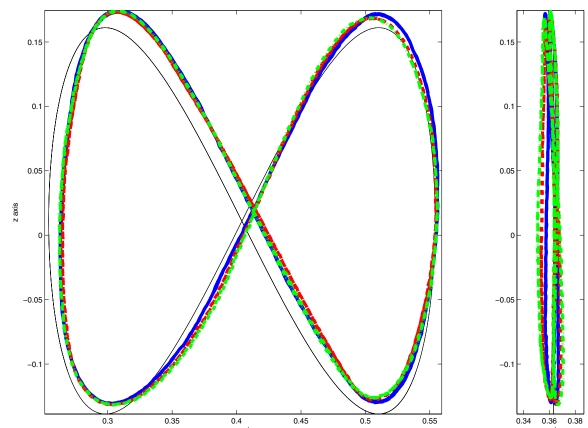

Fig. 3. The same three controllers tracking the same “figure eight (8)” pattern at a faster pace of 4 seconds per cycle. The labels and units remain the same as in Figure 2.

derived here and evaluate them with a simple tracking task. In future work, we plan to evaluate all controllers presented in this paper with more complex tasks.

The joint-space trajectory controller derived in this paper is already well established in the literature, and such that further evaluation is not necessary. Of more interest to us are the end-effector controllers, since they introduce added complexity, particularly the problem of redundancy resolution. Due to a lack of force sensors on our experimental platform, we are unable to implement the impedance or hybrid controllers, but plan to do so in our future work. For this paper, we evaluate the three end-effector controllers from Section III-B: (i) the resolved-acceleration kinematic controller (with metric $\mathbf{N} = \mathbf{M}^{-2}$) in Equation (31), (ii) Khatib's operational space control law ($\mathbf{N} = \mathbf{M}^{-1}$) in Equation (32), and (iii) the identity metric control law ($\mathbf{N} = \mathbf{I}$) in Equation (33).

As an experimental platform, we use the Sarcos Dextrous Master Arm, a hydraulic manipulator with an anthropomorphic design shown in Figure 1 (b). Its seven degrees of freedom mimic the major degrees of freedom of the human arm, i.e., the three in the shoulder, one in the elbow and in the wrist. The robot's end-effector tracks a planar “figure-eight (8)” pattern in task space at two different speeds. In order to stabilize the null-space trajectories, we choose a PD control in joint space which pulls the robot towards a fixed rest posture, $\mathbf{q}_{\text{rest}}$; this control law is given by

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

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 2 shows the end-point trajectories of the three controllers in a slow pattern of 8 seconds per cycle “figure-eight (8)”. Figure 3 shows a faster pace of 4 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 (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 the Table I.

As expected, the performance of the three controllers is very similar in task space. However, the resolved-acceleration kinematic controller ($\mathbf{N} = \mathbf{M}^{-2}$) appears to have a slight advantage here. The reason is most likely due to errors in the dynamics model, since the effect of these is amplified by the inversion of the mass matrix in the control laws given in Equations (32, 33) while the decoupling of the dynamics and kinematics provided by the controller in Equation (31) can be favorable as the effect of the modeling error is not increased. Clearly, more accurate model parameters of the manipulator's rigid body dynamics would result in a reduction of the gap between these control laws as we have confirmed in

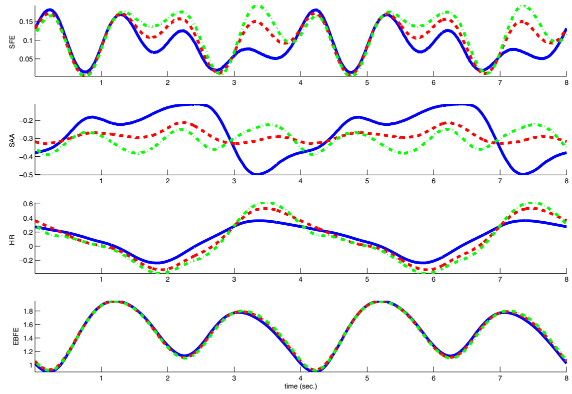

Fig. 4. Joint space trajectories for the four major degrees of freedom, i.e., shoulder flexion-extension (SFE), shoulder adduction-abduction (SAA), humeral rotation (HR) and elbow flexion-extension (EBFE), are shown here. Joint angle units are in radians. The labels are identical to the ones in Figure 2.

simulations. Figure 4 shows 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.

V. Conclusion

In this paper we have presented a novel optimal control framework which allows the development of a unified approach for deriving robot control laws. We have shown in detail how we can make use of both the robot model and a task description in order to create the control law which is optimal with respect to the squared motor command under a 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 that 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 implemention of three of the end-effector trajectory control laws resulting from our unified framework on a real-world Sarcos Master Arm robot has been carried out. 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 the easy development of a host of novel ones. The particular controllers reported in this paper were selected primarly for illustarting the applicability of this framework and showing its strength in unifying different control algorithms using a common building principle. In future work, we will show how this framework can yield a variety of new and interesting control laws for underactuated tasks and robots, for non-holomonic robots and tasks, and for robots with flexible links and joints.

TABLE I

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

Acknowledgment

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

[1] F. E. Udwadia, “A new perspective on tracking control of nonlinear structural and mechanical systems.” *Proc. R. Soc. Lond. A*, vol. 2003, pp. 1783–1800, 2003.

[2] T. Yoshikawa, *Foundations of Robotics: Analysis and Control*. MIT Press, 1990.

[3] C. A. C. D. Wit, B. Siciliano, and G. Bastin, *Theory of Robot Control*. Springer-Verlag Telos, 1996.

[4] R. E. Bellman and R. E. Kalaba, *Dynamic programming and modern control theory*. Academic Press, 1965.

[5] A. E. Bryson, *Applied Optimal Control: Optimization, Estimation, and Control*. Hemisphere Pub. Corp., 1981.

[6] F. E. Udwadia and R. E. Kalaba, *Analytical Dynamics: A New Approach*. Cambridge University Press, 1996.

[7] H. Bruyninckx and O. Khatib, “Gauss’ principle and the dynamics of redundant and constrained manipulators,” *Proceedings of the 2000 IEEE International Conference on Robotics & Automation*, pp. 2563–2569, April 2000.

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

[9] J. Park, W.-K. Chung, and Y. Youm, “Characterization of instability of dynamic control for kinematically redundant manipulators,” *Proc. IEEE Int. Conference on Robotics and Automation*, 2002.

[10] ——, “Specification and control of motion for kinematically redundant manipulators,” *Proc. Internationational Conference of Robotics Systems*, 1995.

[11] W. Chung, W. Chung, and Y.Youm, “Null torque based dynamic control for kinematically redundant manipulators,” *Journal of Robotic Systems*, vol. 10, no. 6, pp. 811–834, 1993.

[12] K.C.Suh and J. M. Hollerbach, “Local versus global torque optimization of redundant manipulators,” *Proc. IEEE Int. Conference on Robotics and Automation*, pp. 619–624, 1987.

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