Manuscript submitted to
AIMS' Journals
Volume **X**, Number **0X**, XX **200X**

Website: http://AIMsciences.org

pp. **X–XX**

A BILEVEL OPTIMIZATION APPROACH TO OBTAIN OPTIMAL COST FUNCTIONS FOR HUMAN ARM MOVEMENTS

Sebastian Albrecht

Chair for Mathematical Optimization, Department of Mathematics
Technische Universität München (TUM)
Boltzmannstr. 3, 85747 Garching b. München, Germany

Marion Leibold

Institute of Automatic Control Engineering,
Technische Universität München (TUM)
Arcisstr. 21, 80290 München, Germany

Michael Ulbrich

Chair for Mathematical Optimization, Department of Mathematics
Technische Universität München (TUM)
Boltzmannstr. 3, 85747 Garching b. München, Germany

(Communicated by the associate editor name)

Abstract. Using a bilevel optimization approach, we investigate the question how humans plan and execute their arm motions. It is known that human motions are (approximately) optimal for suitable and unknown cost functions subject to the dynamics. We investigate the following inverse problem: Which cost function out of a parameterized family (e.g., convex combinations of functions suggested in the literature) reproduces recorded human arm movements best? The lower level problem is an optimal control problem governed by a nonlinear model of the human arm dynamics. The approach is analyzed for a dynamical 3D model of the human arm. Furthermore, results for a two-dimensional experiment with human probands are presented.

1. Introduction.

1.1. Inverse Optimal Control Problem.

The analysis of human behavior is complex because different layers are involved: first, the human brain and the central nervous system (CNS), then reflexes, adaption and learning, and finally the dynamics of the human body and its inherent noise. Due to its complexity, only a part of this problem is tackled here. This paper analyzes human arm movements by identifying parameters in an optimal control problem that is governed by a dynamic model of the arm and of some muscles.

2000 *Mathematics Subject Classification.* 90C90, 49N45, 49M25.

Key words and phrases. Bilevel optimization, optimal control, human arm motions, direct discretization method.

This work is in part funded by the CoTeSys cluster of excellence "Cognition for Technical Systems" within the Excellence Initiative of the DFG. Part of the numerical computations were performed on a Linux cluster supported by DFG grant INST 95/919-1 FUGG.

The dynamical properties of the skeleton of the human arm are modeled by standard rigid-body dynamics (e.g. [59]). The non-rigid effects of the human arm are neglected because their influence on the overall arm movement is assumed to be small. In the following, a model with two degrees of freedom and a three-dimensional model with 5 degrees of freedom are investigated. Furthermore, various muscle models of different accuracy and complexity fit into this framework. A lumped-muscle model consisting of muscle pairs is utilized here as a compromise between complexity and accuracy. To analyze rest-to-rest movements without constraints on the movement time, a linear ODE [56] is used to model the muscle dynamics. The combination of rigid body dynamics and the human muscle dynamics yield a first-order ODE which is later used in the optimization.

Human arm movements show common features, which leads to the assumption that humans minimize a cost function. This hypothesis, being one of the fundamentals of bionics, forms the basis for the approach presented in this paper. Thus, human motion planning is viewed as an optimal control problem governed by an ODE model of the dynamics with a suitable cost function. There exist several cost functions in the literature and some of them will be reviewed (see Sec. 1.2.1). They are usually suitable for explaining certain types of motions but inappropriate for others, and the task of finding the correct cost function needs to be investigated systematically. As Todorov [50] stated, "the true performance criterion in most cases is likely to involve a mix of cost terms". Since human arm movements vary just as much as the shapes of human bodies and their physical training situation, a cost function is sought which describes the arm movements of one person in its current state of adaption to the task. Such a description can be used in robotics to predict the behavior of interacting humans, to control robots in a more human-like way or to cancel time delay in robot control by telepresence [13, 29, 46].

The mathematical problem of finding such a cost function while considering the dynamics is a bilevel program and the existence of a global solution is proved using results of bilevel optimization theory (see Sec. 3). A first-discretize-then-optimize framework is utilized and the bilevel program is then transformed into a one-level program using the KKT-conditions of the discretized lower level optimal control problem. The resulting problem is solved by the interior-point algorithm `IPOPT` and problem modifications are analyzed to improve the performance of the algorithm.

To analyze experimental data, rest-to-rest movements of probands are recorded. A special experimental setup of two linear axes in combination with a handle is used to record the human trajectories and to generate external forces on the human hand. These external forces, which correspond to a virtual mass and a virtual damper acting on the hand, are included in the optimization process.

1.2. **State of the Art.**

1.2.1. *Generation of Human Arm Motions.* Point-to-point motions of the human arm have intensively been studied in literature. Consequently, many properties were discovered and some potential explanations for stereotypical characteristics were proposed:

It is reported that the corresponding hand paths are smooth, roughly straight [21], but slightly curved [4] and the tangential velocity profile is bell-shaped [2, 32, 33].

Based on the fundamental hypothesis of bionics, Flash and Hogan [21] proposed for rest-to-rest arm movements that humans minimize hand jerk, which is the third

time derivative of the hand position. The resulting trajectories are fairly straight and to obtain curved paths like those actually observed, via-points have to be included. Uno et al. [52] investigated the time derivative of the torques in the joints as a cost function minimized by humans. This cost takes the dynamics directly into account and yields slightly curved hand paths with bell-shaped velocity profiles. Some variations of this criterion exist: First, the *minimum commanded torque change* [34], which includes only those torques generated actively by the human. Second, the *minimum muscle tension change* [52], which considers the dynamics of the muscles generating the motion.

For specialized arm motions further cost functions are known: For example, minimizing the movement time yields bang-bang control, which is not observed in humans doing point-to-point movements without a time constraint. If instead the peak velocity or the peak acceleration is optimized a trapezoidal velocity or acceleration profile is obtained. These can be found in musical movements [35] but not in the case considered here. Other cost functions are based on a minimum energy assumption. Most of these cost functions are closely related to the discussed criteria like jerk or torque change. Nelson [35] and Todorov [50], however, came to the conclusion that energy minimization alone fails to account for average behavior. Harris and Wolpert [24] propose a cost function which is based on the signal-dependent noise in human muscles. They argue that humans do their movements in such a way that the variance of the hand position at the end point is minimized. Todorov [49] presents another approach to include noise characteristics of human muscles. A stochastic optimal controller for linear dynamic systems is developed.

1.2.2. *Inverse Optimization and Bilevel Optimization.* Many different strategies have been proposed to reach the goal of human-like robot control. They originate from different disciplines, where the intended goals range form biologically inspired imitation learning [44] to mapping human motions onto animated characters [22]. Various learning strategies (e.g. [1, 8]) or parameter identification methods (e.g. [48]) have been discussed with the aim to improve the control strategies by a reward function or to reproduce observed skilled motions. In general, the focus is on a direct mapping from states to actions and not on identifying the objective function leading to this mapping. Thus, recorded motions are reproduced but the generalizability to new tasks is limited.

Finding the objective function itself is the goal of inverse optimization. A related approach in engineering is inverse reinforcement learning where an underlying Markov decision problem is assumed [43]. Various algorithms are known to determine the unknown reward function (e.g. [36, 41, 42]), but, due to the relatively free structure of the underlying problem and the considered reward functions, the solution that fits the learning data best might not be generalizable to other data. In contrast, a limited set of physically-inspired cost functions are considered in the current paper, and consequently the solution is interpretable in the context of the dynamics of the system.

Since the early works on bilevel programming (e.g. [9, 10]) bilevel optimization problems have been the subject of intensive research. Bilevel programs are strongly related to other important problem classes, e.g., mathematical games and mathematical programs with equilibrium constraints (MPECs). Refer to section 3.1 and to [7, 12, 16, 30, 53] for more details on bilevel programming.

The combination of an optimal control problem with a superimposed parameter identification yields an "inverse optimal control problem", which is a special bilevel program. Only few studies are known for this problem.

Liu et al. [28] consider a given optimization criterion of minimizing joint torques squared while utilizing a dynamical model of the human. Fitting data of motion capture, various model parameters are determined by a nonlinear inverse optimization technique.

A numerical method is presented by Mombaur et al. [31] to solve another human motion based bilevel program. The considered task is to find the objective function for overall locomotion path generation. A simple ODE model is utilized to model the dynamics of the human body, because it is not "the aim of the model [...] to describe locomotion on the detailed level of joint trajectories" [31]. The lower level of the bilevel program is solved by the multiple shooting method `MUSCOD` [18]. This direct method solves the obtained nonlinear problem with an SQP method. To solve the upper level program, the derivative-free method `BOBYQA` [40] is used, which is based on an interpolation based trust region technique. First experimental results show that a combination of five cost functions yields good results.

1.3. **Contribution of this Paper.** The main contribution of the present paper is to investigate a parametrized family of cost functions and to consider the inverse problem of finding the cost function that fits best recorded data of human motion. Utilizing this approach, human motions can be transferred to robots or used to predict human arm movements. The general form of an inverse optimization problem with nonlinear dynamics is stated and a solution strategy is presented. The existence of a global solution is proved under weak conditions and problem modifications are analyzed to improve the performance of the used algorithm.

The remainder of this paper is organized as follows: In section 2 the dynamical arm model is introduced and standard cost functions are reviewed. The bilevel program is transformed into a one-level NLP in section 3 and a solution strategy is investigated in section 4. Section 5 analyzes recorded human arm data and, finally, a short discussion (section 6) concludes this paper.

2. **Human Arm Motions.** The presented approach is based on an optimal control formulation of human motions. To this end, the dynamics of the systems have to be modeled and a cost function has to be stated.

2.1. **Human Arm Dynamics.** Two rigid links are used to model the bones of the human arm, and two joints, the shoulder and the elbow, are considered. The two-dimensional model differs from the 3D one only in the degrees of freedom of these joints.

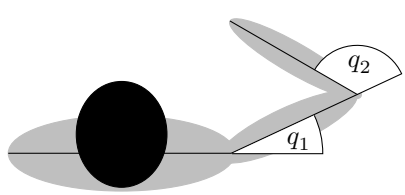

FIGURE 1. Schematic representation of the 2D model of the human arm with joint angles q_i

The joint angles are denoted by $q_i(t)$ and the corresponding time derivatives by $\dot{q}_i(t)$ and $\ddot{q}_i(t)$. The vectors $\boldsymbol{q}(t)$, $\dot{\boldsymbol{q}}(t)$, $\ddot{\boldsymbol{q}}(t)$ are compositions of the joint angles and their time derivatives, respectively. In the two-dimensional model the elbow and the shoulder have only one degree of freedom, whereas in the three-dimensional model of the human arm the elbow is modeled by two degrees of freedom and the shoulder is assumed to have three degrees of freedom.
Using the inertia matrix $\boldsymbol{M}(\boldsymbol{q}(t))$ and a term $\boldsymbol{\rho}(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))$ combining centrifugal and Coriolis terms, the arm dynamics can be written as follows:

$$\boldsymbol{M}(\boldsymbol{q}(t))\ddot{\boldsymbol{q}}(t) + \boldsymbol{\rho}(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t)) = \boldsymbol{T}(t), \tag{1}$$

where $\boldsymbol{T}(t)$ are the torques acting on the joints. Several different approaches (e.g. [20, 27, 59]) are known to derive the dynamic equations (1).

Given position and velocity of a human arm the acceleration $\ddot{\boldsymbol{q}}(t)$ can be controlled by the torques $\boldsymbol{T}(t)$:

$$\ddot{\boldsymbol{q}}(t) = \boldsymbol{M}(\boldsymbol{q}(t))^{-1} \left(\boldsymbol{T}(t) - \boldsymbol{\rho}(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))\right). \tag{2}$$

In the presented model the joint torques $\boldsymbol{T}(t)$ are a combination of the torques $\boldsymbol{T}_m(t)$ generated by the muscles forces, the torques $\boldsymbol{T}_p(t)$ resulting from passive properties of the human arm and the torques $\boldsymbol{T}_e(t)$ induced on the arm by external forces:

$$\boldsymbol{T}(t) = \boldsymbol{T}_m(t) + \boldsymbol{T}_p(t) + \boldsymbol{T}_e(t).$$

First, given the muscle forces $\boldsymbol{F}_m(t)$ and the moment arms $\boldsymbol{R}$, the resulting joint torques are given by $\boldsymbol{T}_m(t) = \boldsymbol{R}\ \boldsymbol{F}_m(t)$. Second, joint damping B_i results in the passive torque term $T_{p,i}(t) = B_i\ \dot{q}_i(t)$. The last torque $\boldsymbol{T}_e(t) = \boldsymbol{J}_p(\boldsymbol{q}(t))\ \boldsymbol{F}_e(t)$ is determined by the external force $\boldsymbol{F}_e(t)$ and the Jacobian of the hand position $\boldsymbol{p}(t)$.

To complete the model of the human arm given by ODE (2) the dynamical behavior of the muscles has to be modeled. Muscle behavior seems to be highly nonlinear and there exists a huge variety of muscle models of different complexities (e.g. [25, 26, 56, 57]). To keep the main focus on the mathematics a simple ODE model is used, but it could be replaced by a more realistic model without changing the general framework. The following second-order low-pass filter taken from Winter [56] models a muscle-like behavior:

$$\tau^2 \ddot{\boldsymbol{F}}_m(t) + 2\tau \dot{\boldsymbol{F}}_m(t) + \boldsymbol{F}_m(t) = \boldsymbol{u}(t), \tag{3}$$

with the time constant $\tau = 0.04$[s] and $\boldsymbol{u}(t)$ denoting the corresponding control.

In conclusion, the dynamics of a human arm are modeled here by the combination of equations (1) and (3). The complete system can be rewritten as a first-order ODE

$$\dot{\boldsymbol{v}}(t) = \boldsymbol{\varphi}(\boldsymbol{v}(t), \boldsymbol{u}(t)),$$

with the state $\boldsymbol{v}(t) = \left[\boldsymbol{q}(t)^T, \dot{\boldsymbol{q}}(t)^T, \boldsymbol{F}_m(t)^T, \dot{\boldsymbol{F}}_m(t)^T\right]^T$.

2.2. **Optimization Principles.** Based on the assumption that humans minimize a suitable cost function when performing their arm motions, several cost functions have been presented in literature for the two-dimensional case. The following basic cost functions are used in our examples:

2.2.1. *Minimum Jerk.* Flash and Hogan [21] introduce

$$J_1 = \int_{t_0}^{t_f} \left\| \frac{d^3}{dt^3} \boldsymbol{p}(t) \right\|^2 \, dt,$$

the integral of hand jerk squared over the total time interval of the movement $[t_0, t_f]$. Here the Cartesian coordinates of the hand position $\boldsymbol{p}(t)$ are utilized. A variation of this cost function is the *minimum joint jerk criterion*

$$J_2 = \int_{t_0}^{t_f} \left\| \frac{d^3}{dt^3} \boldsymbol{q}(t) \right\|^2 \, dt,$$

which yields slightly curved hand paths.

2.2.2. *Minimum Torque Change.* Another cost function (J_3 below) was introduced by [51], which is the integral over the squared first time derivative of the torques $\boldsymbol{T}$ in the joints. Similar to the minimum jerk criterion a variation is considered here, the *minimum muscle tension change* (J_4 below) [52], which considers the dynamics of the muscles generating the motion:

$$J_3 = \int_{t_0}^{t_f} \left\| \frac{d}{dt} \boldsymbol{T}(t) \right\|^2 \, dt \text{ and } J_4 = \int_{t_0}^{t_f} \left\| \frac{d}{dt} \boldsymbol{F}_m(t) \right\|^2 \, dt.$$

2.2.3. *Convex Family of Cost Functions.* In the following convex combinations of these cost functions J_i will be considered. Our goal is to find that convex combination which fits recorded human motions best. Therefore, the set of all feasible cost functions S is defined by

$$S_J := \left\{ J_{\boldsymbol{\alpha}} = \sum_i \alpha_i J_i \;\middle|\; \boldsymbol{\alpha} \in S \right\} \text{ where } S = \left\{ \boldsymbol{\alpha} \mid \alpha_i \geq 0, \; \sum_i \alpha_i = 1 \right\}.$$

The scalars α_i are the weights corresponding to the costs J_i and the vector of all weights $\boldsymbol{\alpha}$ is called the respective weight distribution.

3. **Mathematical Problem Formulation.** The biomechanical dynamics of the human arm in combination with a cost function J lead to an optimization problem of the following type:

$$\min J(\boldsymbol{v}(t), \boldsymbol{u}(t)) \text{ subject to } \dot{\boldsymbol{v}}(t) = \boldsymbol{\varphi}(\boldsymbol{v}(t), \boldsymbol{u}(t)), \tag{4}$$

where $\boldsymbol{v}(t) \in \mathbb{R}^n$ is the state vector for a given point of time $t \in [t_0, t_f]$ and $\boldsymbol{u}(t) \in \mathbb{R}^p$ is the corresponding control vector.
Using the function $\boldsymbol{b} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n_{bc}}$, boundary conditions are defined for selected states at start t_0 and end t_f by appointing given values:

$$\boldsymbol{b}(\boldsymbol{v}(t_0), \boldsymbol{v}(t_f)) = \boldsymbol{0}.$$

The controls are bounded by $\boldsymbol{u}(t) \in [\boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}]$ for given constants $\boldsymbol{u}_{\min}$ and $\boldsymbol{u}_{\max} \in \mathbb{R}^p$. This for itself is a well-posed problem, the *lower level program.*

Given recorded arm data $\boldsymbol{p}_{\text{data}}$, the goal is to choose a cost function $J_{\boldsymbol{\alpha}}$ out of a given family of considered costs S_J, parameterized by the weight distribution $\boldsymbol{\alpha} \in S$, such that the associated computed arm data $\boldsymbol{p}_{\text{comp}}(\boldsymbol{\alpha})$ approximate $\boldsymbol{p}_{\text{data}}$ best:

$$\min_{\boldsymbol{\alpha} \in S} d(\boldsymbol{p}_{\text{data}}, \boldsymbol{p}_{\text{comp}}(\boldsymbol{\alpha})),$$

for a suitable metric d (see appendix 7.2). This problem is referred to as the *upper level program*, because the computed arm data $\boldsymbol{p}_{\text{comp}}(\boldsymbol{\alpha})$ is a function of the state

$\boldsymbol{v}(t)$ and the control $\boldsymbol{u}(t)$ minimizing the lower level optimization problem for this cost $J_{\boldsymbol{\alpha}}$.

By these means the pair of an upper level and a lower level optimization problem is a *bilevel program*. Therefore in a first step basic bilevel programming is revisited and then a discretization strategy enabling a problem reformulation is presented, followed by a discussion of a first-discretize-then-optimize framework. Finally, a solution strategy for the reformulated problem is given.

3.1. **Bilevel Programs.** This section follows the introduction to bilevel programming by Dempe [15].

A *lower level program* is given by

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x},\boldsymbol{y}) \text{ subject to } \boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) \leq \boldsymbol{0} \text{ and } \boldsymbol{h}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{0} \tag{5}$$

with a fixed $\boldsymbol{y} \in \mathbb{R}^m$ and with the cost functions $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, the inequality conditions $\boldsymbol{g} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ and the equality conditions $\boldsymbol{h} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q$. All functions are assumed to be sufficiently smooth.

By using the solution set $\Psi(\boldsymbol{y})$ of (5) for a given $\boldsymbol{y}$-value, the *upper level program* can be stated as

$$\min_{\boldsymbol{x}\in\mathbb{R}^n,\boldsymbol{y}\in\mathbb{R}^m} F(\boldsymbol{x},\boldsymbol{y}) \text{ subject to } \boldsymbol{x} \in \Psi(\boldsymbol{y}) \text{ and } \boldsymbol{y} \in Y, \tag{6}$$

where $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is the upper level cost function and $Y \subset \mathbb{R}^m$ a closed set. Again, F is assumed to be sufficiently smooth. In most cases the set Y itself can be specified by inequality constraints $\boldsymbol{G} : \mathbb{R}^m \to \mathbb{R}^k$ and equality constraints $\boldsymbol{H} : \mathbb{R}^m \to \mathbb{R}^l$:

$$Y = \{\boldsymbol{y} \in \mathbb{R}^m \mid \boldsymbol{G}(\boldsymbol{y}) \leq \boldsymbol{0},\ \boldsymbol{H}(\boldsymbol{y}) = \boldsymbol{0}\},$$

which in particular means that the upper level constraints are independent of the variable $\boldsymbol{x}$ and thereby not coupling $\boldsymbol{x}$ and $\boldsymbol{y}$.

Whether the upper level cost function can actually be minimized over $\boldsymbol{x}$ as opposed to choosing an arbitrary $\boldsymbol{x}$ out of the set $\Psi(\boldsymbol{y})$ depends on the selected class of bilevel programs. Here the special case of a unique lower level solution is considered; thus, both scenarios coincide with each other.

Denote the feasible set for a given $\boldsymbol{y}$ by

$$\Phi(\boldsymbol{y}) := \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) \leq \boldsymbol{0},\ \boldsymbol{h}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{0}\}.$$

Given the uniqueness of the lower level solution, the bilevel program can be rewritten as

$$\min_{\boldsymbol{y}\in\mathbb{R}^m} F(\boldsymbol{x}(\boldsymbol{y}),\boldsymbol{y}) \text{ subject to } \boldsymbol{y} \in Y.$$

Two conditions have to be specified to prove the existence of a global optimal solution for the case of a unique lower level solution:
First, the compactness (C) of the feasible set for the lower level program

$$\{(\boldsymbol{x},\boldsymbol{y}) \mid \boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) \leq \boldsymbol{0},\ \boldsymbol{h}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{0},\ \boldsymbol{y} \in Y,\}$$

and second, the Mangasarian-Fromowitz constraint qualification (MFCQ) for the lower level program for all $(\boldsymbol{x},\boldsymbol{y})$ where $\boldsymbol{y} \in Y$ and $\boldsymbol{x} \in \Phi(\boldsymbol{y})$ (cf. [15]).

If the conditions (MFCQ) and (C) hold true, the solution function $\boldsymbol{x}(\boldsymbol{y})$ is continuous [15]. Y is a closed set by the basic assumption on the upper level program (6). The compactness of the set

$$\{(\boldsymbol{x},\boldsymbol{y}) | \boldsymbol{y} \in Y, \boldsymbol{x} \in \Phi(\boldsymbol{y}), \boldsymbol{x} = \boldsymbol{x}(\boldsymbol{y})\}$$

follows by combining this property with (C) and the continuity of $\boldsymbol{x}(\boldsymbol{y})$. Consequently by the classical Weierstraß theorem the continuous function $F(\boldsymbol{x},\boldsymbol{y})$ has a global minimum on this set.

Summing up the following theorem [15] is obtained:

Theorem 3.1. *Consider the bilevel program (5), (6) and let the assumptions (C) and (MFCQ) be satisfied for all points $(\boldsymbol{x},\boldsymbol{y}) \in \mathbb{R}^n \times Y$ with $\boldsymbol{x} \in \Phi(\boldsymbol{y})$. Assume that the bilevel program (5) has a unique solution $\boldsymbol{x}(\boldsymbol{y})$ for all $\boldsymbol{y} \in Y$ and that the bilevel program is feasible. Then a global optimal solution of the bilevel program exists.*

For the special problem of human arm movements this theorem will guarantee the existence of a global minimum.

There are many different approaches to handle such a bilevel program by transformation to a standard (one-level) optimization problem. One approach, for example, utilizes the Implicit Function Theorem to obtain a local function $\boldsymbol{x}(\boldsymbol{y}) : \mathbb{R}^m \to \mathbb{R}^n$ which is then integrated into the upper level program formulations. This yields implicitly defined constraints or cost functions (e.g. [14, 38]).
Another approach is based on the definition of a function $\phi(\boldsymbol{y})$ by

$$\phi(\boldsymbol{y}) = \min_{\boldsymbol{x}\in\mathbb{R}^n} \{ f(\boldsymbol{x},\boldsymbol{y}) \mid \boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) \leq \boldsymbol{0},\ \boldsymbol{h}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{0} \}.$$

In this case the lower level program can be replaced by the non-differentiable constraint $f(\boldsymbol{x},\boldsymbol{y}) = \phi(\boldsymbol{y})$ (e.g. Shimizu95,Ye95).
The KKT-conditions of nonlinear optimization theory are utilized for the approach presented here. If the lower level program is substituted by its KKT-conditions a one-level program with equilibrium constraints is obtained, a so-called MPEC, but in most cases the bilevel program and the MPEC are not equivalent (e.g. [5, 6, 19, 39]).

In general, the problem that for all feasible points standard CQs (e.g. MFCQ) are violated arises for optimization problems with KKT-conditions as part of their constraints. In our case the only inequality constraints imposed on the lower level are bounds on the controls. The numerical results show that these constraints do not become active for the movements analyzed here. Hence, all bounds are inactive and so no complementarity condition causes this dilemma. Here, under suitable conditions, it is possible to show that a CQ holds; consequently, KKT-conditions are necessary for the one-level program and can therefore be used by the solution algorithm.

3.2. **Discretization Strategy.** The optimal control problem needs to be discretized to directly utilize the bilevel theory stated above. A given ordinary differential equation (ODE) $\dot{\boldsymbol{v}}(t) = \boldsymbol{\varphi}(\boldsymbol{v}(t),\boldsymbol{u}(t))$ can be discretized in many different ways. There exists a variety of single-step and multi-step methods known from the numerical solution theory of ODEs [17, 23].

The discretized version is assumed to have the following structure:

$$\boldsymbol{D}\bar{\boldsymbol{v}} = \bar{\boldsymbol{\varphi}}(\bar{\boldsymbol{v}},\bar{\boldsymbol{u}}), \tag{7}$$

where $\boldsymbol{D}$ is the discrete derivation matrix, $\bar{\boldsymbol{\varphi}}$ the discretized right-hand side, and $\bar{\boldsymbol{v}} = \left(\boldsymbol{v}(t_0)^T,\dots,\boldsymbol{v}(t_N)^T\right)^T \in \mathbb{R}^{nN}$, $\bar{\boldsymbol{u}} = \left(\boldsymbol{u}(t_0)^T,\dots,\boldsymbol{u}(t_N)^T\right)^T \in \mathbb{R}^{pN}$ are the discretized state and control, respectively.

Hence the following discretized lower level program is obtained:

$$\min \bar{J}(\bar{\boldsymbol{v}},\bar{\boldsymbol{u}}) \text{ subject to } \bar{\boldsymbol{g}}(\bar{\boldsymbol{u}}) \leq \boldsymbol{0},\ \boldsymbol{D}\bar{\boldsymbol{v}} = \bar{\boldsymbol{\varphi}}(\bar{\boldsymbol{v}},\bar{\boldsymbol{u}}) \text{ and } \bar{\boldsymbol{b}}(\bar{\boldsymbol{v}}) = \boldsymbol{0}, \tag{8}$$

where $\bar{J}$ is an adequate discrete version of J (compare equation (4)) and $\bar{\boldsymbol{b}}$ states the boundary conditions for the discretized state.

The equality constraints are combined in the following function:

$$\bar{\boldsymbol{h}}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}}) := \begin{pmatrix} \boldsymbol{D}\bar{\boldsymbol{v}} - \bar{\boldsymbol{\varphi}}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}}) \\ \bar{\boldsymbol{b}}(\bar{\boldsymbol{v}}) \end{pmatrix}.$$

We assume that the discretization strategy guarantees that the columns of the matrix $\nabla\bar{\boldsymbol{h}}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}})$ are linearly independent.

3.3. **Reformulation of the Problem.** In this section the inverse optimal control problem is reformulated to utilize the results of section 3.1 and the first-order necessary conditions are used to state the bilevel program as a one-level program (see (9)).

The human motions analyzed here are self-paced rest-to-rest arm movements. Physically motivated bounds on the controls do not become active for these movements for all considered lower level cost functions. Therefore, in the following we can assume that the controls are bounded without control bounds being active. This setting allows us to prove the existence of a global solution of the bilevel program by theorem 3.1.

The compactness of the set of feasible points for the lower level problem is ensured due to the underlying differential equations and the given boundary values in combination with the bounded controls $\bar{\boldsymbol{u}}$ and external forces $\boldsymbol{F}_e$.

In this setting the linear independence constraint qualification (LICQ), implying (MFCQ), holds if the columns of the matrix $\left(\nabla\bar{\boldsymbol{h}}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}}), \nabla\bar{\boldsymbol{g}}_{\mathcal{A}}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}})\right)$ are linearly independent. Since no active control bounds are observed, the (LICQ) follows from the linear independency of the columns of $\nabla\bar{\boldsymbol{h}}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}})$ guaranteed by the utilized discretization strategy.

In addition, given the lower level cost functions of section 2.2, the solution of the problem (5) can be assumed to be unique for an arm movement with respect to the upper level cost function. This way all prerequisites are fulfilled and theorem 3.1 states that for a feasible task a global minimum exists.

A local minimizer $(\bar{\boldsymbol{v}}^*, \bar{\boldsymbol{u}}^*)$ of problem (8) has to fulfill the first-order necessary conditions of nonlinear constrained optimization, the *KKT-conditions*, if a constraint qualification (CQ) holds. In the case of omitted control bounds, which do not become active for the arm motions considered here and can therefore be checked a-posteriori, these conditions read as follows:

$$\exists \boldsymbol{\lambda} \in \mathbb{R}^{n(N-1)+n_{bc}} : \ \boldsymbol{0} = \bar{\boldsymbol{h}}(\bar{\boldsymbol{v}}^*, \bar{\boldsymbol{u}}^*), \ \boldsymbol{0} = \nabla\bar{J}(\bar{\boldsymbol{v}}^*, \bar{\boldsymbol{u}}^*) + \nabla\bar{\boldsymbol{h}}(\bar{\boldsymbol{v}}^*, \bar{\boldsymbol{u}}^*)\boldsymbol{\lambda}.$$

Note that the KKT-conditions are necessary for the non-convex problem but not sufficient, which means that the set of all KKT-points $(\bar{\boldsymbol{v}}^*, \bar{\boldsymbol{u}}^*)$ contains the set of all local minimizer of the problem, but not every KKT-point is a local minimizer. Therefore, by using the KKT-conditions as only information about the lower level program, the solution of a created one-level program might be suboptimal with regard to the lower level cost function.

So far the upper level program was stated in the general form

$$\min_{\boldsymbol{\alpha} \in S} d(\boldsymbol{p}_{\text{data}}, \boldsymbol{p}_{\text{comp}}(\boldsymbol{\alpha})),$$

where $\boldsymbol{p}_{\text{comp}}(\boldsymbol{\alpha})$ was the corresponding arm data for a solution of the lower level problem belonging to $J_{\boldsymbol{\alpha}}$. Now the KKT-conditions are used to reformulate the bilevel program as a one-level program:

$$\min d\left(\boldsymbol{p}_{\text{data}}, \bar{\boldsymbol{p}}_{\text{comp}}(\boldsymbol{\alpha}, \bar{\boldsymbol{v}}, \bar{\boldsymbol{u}}, \boldsymbol{\lambda})\right), \tag{9}$$

subject to

$$\begin{aligned} \mathbf{0} &\leq \boldsymbol{\alpha} \\ 1 &= \sum_{i=1}^{m} \alpha_i \\ \mathbf{0} &= \bar{\boldsymbol{h}}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}}) \\ \mathbf{0} &= \sum_{i=1}^{m} \alpha_i \nabla \bar{J}_i(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}}) + \nabla \bar{\boldsymbol{h}}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}})\boldsymbol{\lambda}, \end{aligned}$$

where $\bar{\boldsymbol{p}}_{\text{comp}}(\boldsymbol{\alpha}, \bar{\boldsymbol{v}}, \bar{\boldsymbol{u}}, \boldsymbol{\lambda})$ is the arm data whose distance from the recorded human arm data $\boldsymbol{p}_{\text{data}}$ is measured.

The KKT-conditions are necessary for the one-level program (9), too, if a constraint qualification holds. To show that (LICQ) is fulfilled, the constraints

$$\bar{\boldsymbol{H}}(\boldsymbol{\alpha}, \bar{\boldsymbol{v}}, \bar{\boldsymbol{u}}, \boldsymbol{\lambda}) := \begin{pmatrix} \sum_{i=1}^{m} \alpha_i - 1 \\ \bar{\boldsymbol{h}}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}}) \\ \sum_{i=1}^{m} \alpha_i \nabla \bar{J}_i(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}}) + \nabla \bar{\boldsymbol{h}}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}})\boldsymbol{\lambda} \end{pmatrix}$$

and $\bar{\boldsymbol{G}}(\boldsymbol{\alpha}, \bar{\boldsymbol{v}}, \bar{\boldsymbol{u}}, \boldsymbol{\lambda}) := -\boldsymbol{\alpha}$ are defined. This leads to the matrix

$$\nabla \bar{\boldsymbol{H}}(\boldsymbol{\alpha}, \bar{\boldsymbol{v}}, \bar{\boldsymbol{u}}, \boldsymbol{\lambda}) = \begin{pmatrix} 1 & \mathbf{0}^T & \nabla \bar{J}_1(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}})^T \\ \vdots & \vdots & \vdots \\ 1 & \mathbf{0}^T & \nabla \bar{J}_m(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}})^T \\ \mathbf{0} & \nabla \bar{\boldsymbol{h}}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}}) & \sum_{i=1}^{m} \alpha_i \nabla^2 \bar{J}_i(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}}) + \nabla^2 \bar{\boldsymbol{h}}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}})\boldsymbol{\lambda} \\ \mathbf{0} & \mathbf{0} & \nabla \bar{\boldsymbol{h}}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}})^T \end{pmatrix}$$

whose columns are linearly independent under the assumption that the columns of $\nabla \bar{h}$ are linearly independent and that the second order sufficient optimality conditions hold at the solution of the lower level problem. To keep the notation simple, the symbol ∇ here denotes the gradient (transposed Jacobian) with respect to $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}})$ and $\nabla^2 \bar{\boldsymbol{h}}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}})\boldsymbol{\lambda}$ stands for the Hessian of $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}}) \mapsto \boldsymbol{\lambda}^T \bar{\boldsymbol{h}}(\bar{\boldsymbol{v}}, \bar{\boldsymbol{u}})$.

The (LICQ) then holds here because the columns of the matrix $(\nabla \bar{\boldsymbol{H}}, \nabla \bar{\boldsymbol{G}}_{\mathcal{A}})$ are linearly independent. With this it is shown that the KKT-conditions are necessary for the upper level program, being an assumption for the numerical solution strategy of section 4.

4. Numerical Solution Strategy.

Program (9) is a standard nonlinear optimization problem of the type

$$\min f(\boldsymbol{x}) \text{ subject to } \boldsymbol{x} \geq \mathbf{0} \text{ and } \boldsymbol{c}(\boldsymbol{x}) = \mathbf{0}. \tag{10}$$

This minimization problem can be solved by many different algorithms of nonlinear optimization theory [37]. Here the interior-point filter line-search algorithm `IPOPT` of Wächter and Biegler [54, 55] for large-scale nonlinear programming is used. The basic structure of this algorithm is reviewed in the following, since problem modifications will be investigated to improve the performance for this special problem.

This algorithm is based on a homotopy method applied to the primal-dual equations:

$$\begin{aligned} \nabla f(\boldsymbol{x}) + \nabla \boldsymbol{c}(\boldsymbol{x})\ \boldsymbol{\lambda} - \boldsymbol{z} &= \boldsymbol{0} \\ \boldsymbol{c}(\boldsymbol{x}) &= \boldsymbol{0} \\ \boldsymbol{X}\boldsymbol{Z}\boldsymbol{e} - \boldsymbol{\mu}\boldsymbol{e} &= \boldsymbol{0}, \end{aligned} \tag{11}$$

where $\boldsymbol{e}$ is a vector of ones and the vectors $\boldsymbol{\lambda}$ and $\boldsymbol{z}$ are the Lagrange multipliers belonging to the equality constraint $\boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0}$ and the inequality constraint $\boldsymbol{x} \geq \boldsymbol{0}$, respectively. The diagonal matrices $\boldsymbol{X}$ and $\boldsymbol{Z}$ are defined by $\boldsymbol{X} := \mathrm{diag}(\boldsymbol{x})$ and $\boldsymbol{Z} := \mathrm{diag}(\boldsymbol{z})$. In the course of the algorithm the homotopy parameter μ slackening the complementarity condition $\boldsymbol{X}\boldsymbol{Z}\boldsymbol{e} = \boldsymbol{0}$ is driven to zero.

It should be noted that the equations (11) for $\mu = 0$ in combination with the inequality conditions $\boldsymbol{x}, \boldsymbol{z} \geq \boldsymbol{0}$ are the KKT-conditions for problem (10).

The optimizer `IPOPT` terminates if the total error $E_\mu(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{z})$ is below a user-given tolerance `TOL`. With the scaling parameters $s_d,\ s_c \geq 0$ the total error is computed by

$$E_\mu(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{z}) = \max\left\{ \frac{||\nabla f(\boldsymbol{x}) + \nabla \boldsymbol{c}(\boldsymbol{x})\boldsymbol{\lambda} - \boldsymbol{z}||_\infty}{s_d}, ||\boldsymbol{c}(\boldsymbol{x})||_\infty, \frac{||\boldsymbol{X}\boldsymbol{Z}\boldsymbol{e} - \mu\boldsymbol{e}||_\infty}{s_c} \right\}.$$

For each homotopy step j a problem of the type (11) has to be solved for a given parameter μ_j, which is done by applying a damped Newton's method. This means that by using the linearized system a Newton direction is calculated and in this direction a step width is chosen out of $(0, 1]$.

In detail, for given μ_j and $(\boldsymbol{x}_k, \boldsymbol{\lambda}_k, \boldsymbol{z}_k)$ with $\boldsymbol{x}_k$ and $\boldsymbol{z}_k \geq 0$ a search direction is obtained by solving

$$\begin{pmatrix} \boldsymbol{W}_k & \boldsymbol{A}_k & -\boldsymbol{I} \\ \boldsymbol{A}_k^T & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{Z}_k & \boldsymbol{0} & \boldsymbol{X}_k \end{pmatrix} \begin{pmatrix} \boldsymbol{d}_k^x \\ \boldsymbol{d}_k^\lambda \\ \boldsymbol{d}_k^z \end{pmatrix} = - \begin{pmatrix} \boldsymbol{g}_k + \boldsymbol{A}_k\boldsymbol{\lambda}_k - \boldsymbol{z}_k \\ \boldsymbol{c}(\boldsymbol{x}_k) \\ \boldsymbol{X}_k\boldsymbol{Z}_k\boldsymbol{e} - \mu_j\boldsymbol{e} \end{pmatrix}, \tag{12}$$

where the matrix $\boldsymbol{W}_k := \nabla^2_{xx} L(\boldsymbol{x}_k, \boldsymbol{\lambda}_k, \boldsymbol{z}_k)$ is the Hessian of the Lagrange function $L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{z}) := f(\boldsymbol{x}) + \boldsymbol{c}(\boldsymbol{x})^T\boldsymbol{\lambda} - \boldsymbol{z}^T\boldsymbol{x}$ with respect to $\boldsymbol{x}$, $\boldsymbol{g}_k = \nabla f(\boldsymbol{x}_k)$ the gradient of f, and $\boldsymbol{A_k} := \nabla \boldsymbol{c}(\boldsymbol{x}_k)$ the transposed Jacobian of the constraints.

Given the search directions $(\boldsymbol{d}_k^x, \boldsymbol{d}_k^\lambda, \boldsymbol{d}_k^z)$ a line-search filter method is used to assure global convergence of `IPOPT`. In this context it is necessary to guarantee certain descent properties for the search directions; therefore the system (12) is slightly altered again [54, 55].

4.1. Problem Modifications to Increase Solver Performance.

The time needed to compute a solution of (9) depends critically on the choice of the initial values for $\boldsymbol{\alpha}$, $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{u}}$. Furthermore, it has to be noted that only a local minimizer is calculated and thus the choice of the initial value influences whether the result is close to the global minimum or not.

To considerably speed up the computation and to assure that the solution is close to the global minimum the following strategy is used. Starting from the fact that the value of the upper level cost is nonnegative and should get close to zero a further constraint could be added to the problem (9):

$$f(\boldsymbol{x}) = 0. \tag{13}$$

Note that IPOPT will only assure $f(\boldsymbol{x}^*) \leq \texttt{TOL}$ if the constraint $f(\boldsymbol{x}) = 0$ is prescribed. If the condition $f(\boldsymbol{x}^*) \leq \texttt{TOL}$ is not fulfilled for an optimum, it is possible to relax the constraint by a constant ε, which in turn guarantees for the solution: $f(\boldsymbol{x}^*) \in [\varepsilon - \texttt{TOL}, \varepsilon + \texttt{TOL}]$. For such a relaxation the following results of the non-relaxed case apply accordingly.

The effects of adding a constraint of the type (13) on the operation of the algorithm IPOPT is now briefly analyzed. Denoting by ρ_k the multiplier corresponding to the constraint $f(\boldsymbol{x}) = 0$, we obtain

$$\begin{pmatrix} \boldsymbol{W}_k + \rho_k \boldsymbol{H}_k & \boldsymbol{A}_k & -\boldsymbol{I} & \boldsymbol{g}_k \\ \boldsymbol{A}_k^T & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{Z}_k & \boldsymbol{0} & \boldsymbol{X}_k & \boldsymbol{0} \\ \boldsymbol{g}_k^T & \boldsymbol{0} & \boldsymbol{0} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{d}_k^x \\ \boldsymbol{d}_k^\lambda \\ \boldsymbol{d}_k^z \\ \boldsymbol{d}_k^\rho \end{pmatrix} = - \begin{pmatrix} (1+\rho_k)\boldsymbol{g}_k + \boldsymbol{A}_k \boldsymbol{\lambda}_k - \boldsymbol{z}_k \\ \boldsymbol{c}(\boldsymbol{x}_k) \\ \boldsymbol{X}_k \boldsymbol{Z}_k \boldsymbol{e} - \mu_j \boldsymbol{e} \\ f(\boldsymbol{x}_k) \end{pmatrix}, \tag{14}$$

where $\boldsymbol{W}_k$, $\boldsymbol{A}_k$, and $\boldsymbol{g}_k$ are defined as in (12) and $\boldsymbol{H}_k = \nabla^2 f(\boldsymbol{x}_k)$. The last block row gives

$$\boldsymbol{g}_k^T \boldsymbol{d}_k^x = -f(\boldsymbol{x}_k),$$

which assures that for $f(\boldsymbol{x}_k) > 0$ the search direction $\boldsymbol{d}_k^x$ is a descent direction.

We now interpret the system (14) as a modification of (12). Comparing the first three equations of (14) with (12), we see that the Hessian $\boldsymbol{W}_k$ in the (1,1) block of (12) is replaced by $\boldsymbol{W}_k + \rho_k \boldsymbol{H}_k$. Furthermore, the first block row

$$-\boldsymbol{r}_k^1 = -(\boldsymbol{g}_k + \boldsymbol{A}_k \boldsymbol{\lambda}_k - \boldsymbol{z}_k)$$

of the right hand side of (12) is replaced by $-\boldsymbol{r}_k^1 - (\rho_k + \boldsymbol{d}_k^\rho)\boldsymbol{g}_k$. This can be viewed as the primal dual Newton system (12) for the function

$$(1+\rho_k) f(\boldsymbol{x}) + \boldsymbol{d}_k^\rho \boldsymbol{g}_k^T (\boldsymbol{x} - \boldsymbol{x}_k)$$

instead of $f(\boldsymbol{x})$. Here, the value of $\boldsymbol{d}_k^\rho$ has to guarantee that the step $\boldsymbol{d}_k^x$ satisfies $\boldsymbol{g}_k^T \boldsymbol{d}_k^x = -f(\boldsymbol{x}_k)$.

It is also possible to interpret (14) as a modification of (12) in the (1,1) block only. To cancel the additional term $\boldsymbol{g}_k \boldsymbol{d}_k^\rho$ on the left and $-\rho_k \boldsymbol{g}_k$ on the right of the first row in (14), we can multiply the last row of (14) by $\eta_k \boldsymbol{g}_k$, where $\eta_k := -\frac{\rho_k + d_k^\rho}{f(\boldsymbol{x}_k)}$, and add it to the first row. Then the first three equations are equivalent to (12) after replacing the Hessian $\boldsymbol{W}_k$ in the (1,1) block by $\boldsymbol{W}_k + \rho_k \boldsymbol{H}_k + \eta_k \boldsymbol{g}_k \boldsymbol{g}_k^T$. Here, η_k, which implicitly depends on d_k^ρ, is chosen such that $\boldsymbol{d}_k^x$ satisfies $\boldsymbol{g}_k^T \boldsymbol{d}_k^x = -f(\boldsymbol{x}_k)$.

The constraint $f(\boldsymbol{x}) = 0$ is only added in an initial phase and removed when we get closer to the solution. This is done since in general the constraint $f(\boldsymbol{x}) = 0$ is incompatible unless $f = 0$ holds at the solution. This could be circumvented by using the constraint $f(\boldsymbol{x}) = \varepsilon$ instead. But even then, the gradient of this constraint is linearly dependent to $\nabla f(\boldsymbol{x})$ and thus the multipliers at the solution would not be unique. In fact, we could rescale ρ by a factor α and compensate this by scaling the other multipliers with $\frac{1+\alpha\rho}{1+\rho}$ as long as the latter fraction is positive.

We use n_{start} to prescribe the number of IPOPT iterations in which the additional constraint $f(x) = 0$ is present, and then it is removed.

4.2. **Reconstruction Test.** The following framework, termed "reconstruction", is considered to validate the presented transformation-based approach leading to problem (9):

First, for a given weight $\boldsymbol{\alpha}$ the cost function $J = \sum_i \alpha_i J_i$ is known and the lower level problem can be solved. Using the state vector of the obtained local minimum,

the hand positions $\boldsymbol{p}(t_i)$ can be computed for each point of time t_i. These hand positions are then used to define a curve Γ_{data} by applying a linear approximation.

Second, the bilevel program (9) is solved for this constructed curve Γ_{data}, resulting in a locally optimal solution $\boldsymbol{\alpha}^*$ for the weight vector. In a last step the two vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$ are compared.
Naturally, the more the characteristics of the trajectories belonging to the individual criteria J_i differ, the better the results of the reconstruction get.

The first goal is to randomly choose a combination of these cost functions and to solve the lower level program for this combined cost to get Γ_{data}. To generate the weights $\boldsymbol{\alpha}_{\text{data}}$ directly by the use of a uniform distribution generally results in a disproportional weighting, which means that one cost function dominates the others, because the differences in the value of the lower level cost functions vary considerably if two trajectories are compared. To avoid this problem the following approach is chosen:

First, the lower level program is solved for each basic cost J_i, $i \in \{1, \dots, m\}$, individually yielding the optimal states $\bar{\boldsymbol{v}}_i$.

Second, the values of the cost functions for the different trajectories are calculated $c_{ij} = f_i(\bar{\boldsymbol{v}}_j)$, $i, j \in \{1, \dots, m\}$ and for each lower level cost the corresponding scaling factor d_i is chosen to be the maximal difference in these values:

$$d_i = \max\{|c_{ik} - c_{il}|\}.$$

Then given the uniformly distributed weights $\boldsymbol{\beta}_{\text{data}}$ the following $\boldsymbol{\alpha}_{\text{data}}$ values are obtained:

$$\alpha_{\text{data},i} = \frac{\beta_{\text{data},i}}{d_i}.$$

This construction of $\boldsymbol{\alpha}_{\text{data}}$ assures the equilibrium of the cost function scales. Note that in the most general case the offsets $\min\{|c_{ik}|, k = 1, \dots, m\}$ have to be included into this scaling procedure, too.

Two properties of the solution strategy are analyzed for a given $\boldsymbol{p}_{\text{data}}$. On the one hand, the influence of the problem modifications on the value of the upper level cost function, and on the other hand, the dependence of the final cost values on the initial vectors.

To validate that noise does not change the general properties, additional computations are analyzed where white noise is added to the hand trajectory Γ_{data} yielding Γ_{noise}. The standard deviation of the normal distribution with mean zero is chosen to be 10^{-3} times the distance of start and end point. As a matter of cause, larger values of noise result in larger deviations in the reconstructed weights. Here the size is chosen in range of errors observed in the experiments. No noise is added to the start and end point because the boundary conditions in the bilevel program are assumed to be fixed. The differences between the two lines Γ_{data} and Γ_{noise} result from the fact that the noisy data can in general be less well fitted than the exact data. Note that, especially in case of added noise, an upper level cost value smaller than 10^{-4} is acceptable, because it corresponds to differences between the trajectories on the scale of a millimeter.

4.2.1. *Influence of Problem Modification.* In this first scenario a initial vector combining α, $\bar{\boldsymbol{v}}$, $\bar{\boldsymbol{u}}$ and $\boldsymbol{\lambda}$ is fixed and the number n_{start} (see the end of section 4.1) is

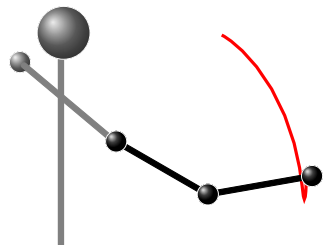

FIGURE 2. Sketch of three-dimensional arm model with an example hand trajectory [illustration not to scale]

varied. The presented results are obtained using the 3D arm model. Similar results are obtained for the more simple two-dimensional case.

The final values of the upper level cost functions for various n_{start}-values are used to compute the means displayed in figure 3. The numbers of `IPOPT` iterations with the additional constraint are given by $n_{\text{start}} \in \{0, 4, 8, 12, 16, 20\}$.

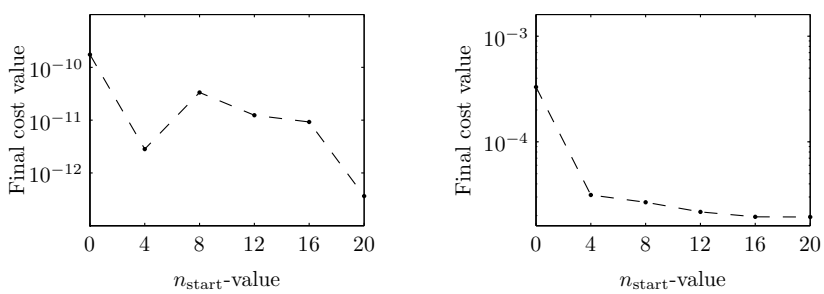

FIGURE 3. Final values of upper level cost function over n_{start} (Mean over 100 random initial vectors) [left: noise-free, right: noisy data]

The optimization results of 100 random initial vectors were utilized to compute the means of the logarithmic cost values. In case `IPOPT` did not terminate at an optimum before 100 iterations were reached, the value of the cost function at the 100th iteration was used in the computation of the mean. The left plot of figure 3 shows the ideal case without noise and the right one the results if noise is considered. Note that in the displayed example the minimal cost value is smaller than the above-mentioned bound.

In the plots of figure 4 the two examples are displayed in more detail. Here each curve belongs to an optimization running with a fixed n_{start}-value. The mean values are computed by constantly extending each curve by the final value. Forking can be observed at points where the n_{start}-value of a curve is reached. In both cases the curves with $n_{\text{start}} = 20$ form a lower bound for the curve belonging to a lower n_{start}-value, meaning the modifications result in faster cost reduction.

These plots show that a certain number of iterations with the additional constraint assures that the solution comes reasonably close to the global optimum. Note that there exists a bound for the value of n_{start} such that for all larger values no further descent can be observed (Here the value is about 20.)

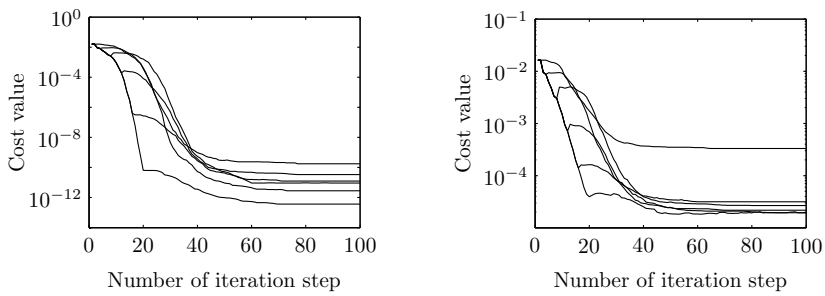

FIGURE 4. Upper level cost during optimization for different n_{start}-values (Mean over 100 random initial vectors) [left: noise-free, right: noisy data]

4.2.2. *Dependence on the Initial Values.* In this section the dependence of the solution on the choice of the initial guess of $\boldsymbol{\beta}$ is analyzed. Two aspects are of interest: the final value of the upper level cost and the "distance" between the computed weight $\boldsymbol{\beta}$ and $\boldsymbol{\beta}_{\text{data}}$.

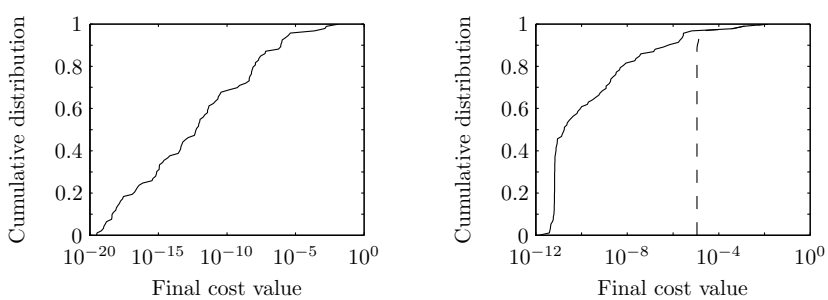

FIGURE 5. Cumulative distribution in dependence on the final cost value [left: noise-free, right: noisy data (dashed: absolute, solid: relative)]

The resulting final values for the upper level cost function are displayed in a cumulative distribution. The x-axis value of figure 5 gives the final value of the upper level cost function and the y-value the corresponding relative frequency. Note that here the x-axis scale is logarithmic and that the relation seems to be nearly linear in the noise-free case. In case noise is considered the dashed line displays the absolute value and the solid line gives the difference relative to the minimum of all 100 optimization runs.

The following cumulative distributions display the accuracy of one example of the computed scalars β_i. Note that the general tendency can be described by a log-linear relation. Again, the dashed line displays the absolute values of the cost function and the solid line the relative values with respect to the computed minimal value.

In addition, it can be analyzed how the computed β_i-values correlate with the values of the upper level cost function. In figure 7 the log-log-plot of the difference $|\beta_i - \beta_{\text{data},i}|$ versus the value of the upper level cost function is displayed for one

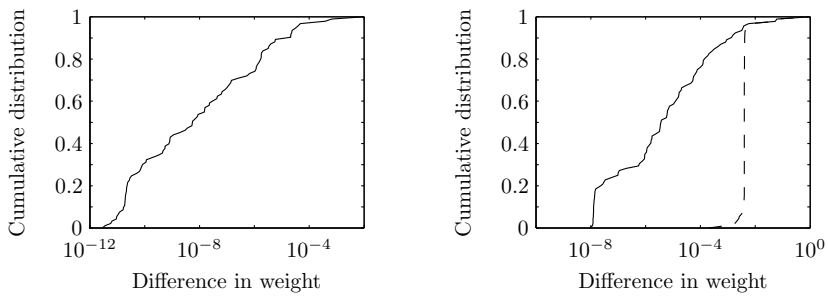

FIGURE 6. Cumulative distributions of one computed weight β_i [left: noise-free, right: noisy data (dashed: absolute, solid: relative)]

example. A reduction of the upper level cost results in a reduction of the distance between $\boldsymbol{\beta}$ and $\boldsymbol{\beta}_{\text{data}}$.

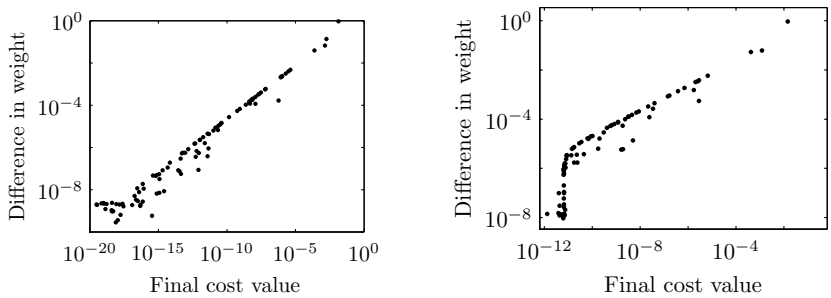

FIGURE 7. Differences between computed and given weights over the final cost [left: noise-free, right: noisy data (relative cost values)]

Summing up, the reconstruction framework shows that adding the constraint (13) for a certain number of iteration steps n_{start} considerably reduces the final value of the upper level cost function.

In addition it is shown that the results clearly depend on the choice of the initial values, but the usage of random initial values indicates that the global solution can be approached sufficiently with a small number of trials. Finally, it is observed that the better the local solution approximates the global one, the more precise are the values of the weights $\boldsymbol{\beta}$.

5. Bilevel Optimization Results for Recorded 2D Motions.

5.1. Experiment Set-up.

The following set-up is chosen to record a planar 2D movement under the influences of a simulated mass and simulated damping on the hand:

Two linear axes are combined by mounting one perpendicular on top of the other to model the two degrees of freedom. The height of the chair of the probands is adjusted such that the shoulder and the hand at the handle are within a horizontal plane. Ideally for a two-dimensional model the elbow should stay in this plane, too, but to obtain normal arm movements the probands are not fixed at the elbow; consequently, the elbow position could drift off which causes small errors in the model.

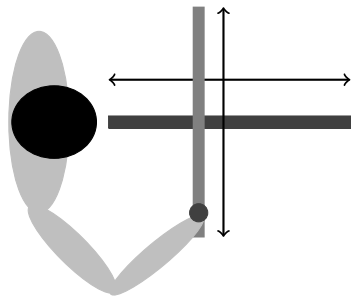

FIGURE 8. Schematic representation of the experimental set-up: 2 linear axes and a handle

The goal of the experiment is to move the handle between indicated positions under the influence of external dynamics.

5.2. **2D Arm Model.** A planar two-link arm model with three (1 biarticular and 2 monoarticular) muscle pairs is considered; hence, the arm configuration is described by the coordinates $\boldsymbol{q} \in \mathbb{R}^2$. The dynamics of the arm are given by equation (1), but in the presence of external forces depending on the acceleration of the hand position the transformation to equation (2) has to be adapted.

The following external forces which are acting on the hand are included in the formalism:

$$\boldsymbol{F}_e = -m\ddot{\boldsymbol{p}} - b\dot{\boldsymbol{p}}, \tag{15}$$

where m is the simulated mass and b is the damping factor. The hand position is here denoted by $\boldsymbol{p}(\boldsymbol{q})$ and the corresponding first and second derivatives by $\boldsymbol{J}_p(\boldsymbol{q})$ and $\boldsymbol{H}_p(\boldsymbol{q})$. Consequently, the identities $\dot{\boldsymbol{p}} = \boldsymbol{J}_p\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{p}} = \boldsymbol{J}_p\ddot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^T\boldsymbol{H}_p\dot{\boldsymbol{q}}$ hold. Inserting these into equation (15) yields

$$\boldsymbol{T}_e = \boldsymbol{J}_p^T\boldsymbol{F}_e = -m\boldsymbol{J}_p^T\boldsymbol{J}_p\ddot{\boldsymbol{q}} - \underbrace{m\boldsymbol{J}_p^T\dot{\boldsymbol{q}}^T\boldsymbol{H}_p\dot{\boldsymbol{q}} - b\boldsymbol{J}_p^T\boldsymbol{J}_p\dot{\boldsymbol{q}}}_{=:\boldsymbol{\rho}_e(\boldsymbol{q},\dot{\boldsymbol{q}})}.$$

Finally the adapted version of equation (1) reads

$$\left(\boldsymbol{M}(\boldsymbol{q}) + m\boldsymbol{J}_p(\boldsymbol{q})^T\boldsymbol{J}_p(q)\right)\ddot{\boldsymbol{q}} + \boldsymbol{\rho}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{T}_m + \boldsymbol{T}_p - \boldsymbol{\rho}_e(\boldsymbol{q},\dot{\boldsymbol{q}}).$$

Note that the matrix $m\boldsymbol{J}_p(\boldsymbol{q})^T\boldsymbol{J}_p(\boldsymbol{q})$ is positive semidefinite since $m \geq 0$ and thereby the combined matrix $\boldsymbol{M}(\boldsymbol{q}) + m\boldsymbol{J}_p(\boldsymbol{q})^T\boldsymbol{J}_p(\boldsymbol{q})$ is always positive definite.

The arm parameters can be found in the appendix 7.1.

5.3. **Optimization Results.** The first step to obtain optimization results is to extract suitable data out of all the recorded one. For each proband ten trials are recorded for all virtual combinations of mass and damper, but some trials differed considerably from the main scheme of the other trials. Therefore, six out of ten trials are selected by the following process: The first of the ten trials is in all cases discarded to minimize correlation of succeeding combinations. Out of the remaining ones the two-thirds are selected which have minimal distance to the median of all trials.

For a moderate ratio between b and m the recorded trajectories show similarities by inspection (compare figure 9). It can be noted that for these cases the variance in the resulting weight distributions is reasonably small, but for those cases where no similarities can be found by inspection the weight distributions seem to be random.

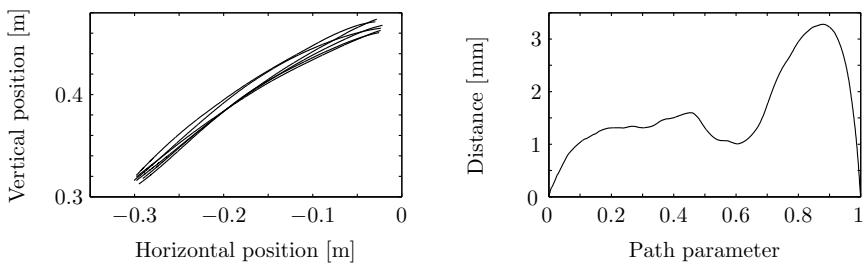

FIGURE 9. Six selected hand paths of a proband with $m = 2$ and $b = 0$ (left) and the mean distance between ten recorded hand paths and the computed optimal ones (right)

In the following plots each cost function J_i is assigned a color: White for J_1, dark gray for J_2, light gray for J_3, and black for J_4.

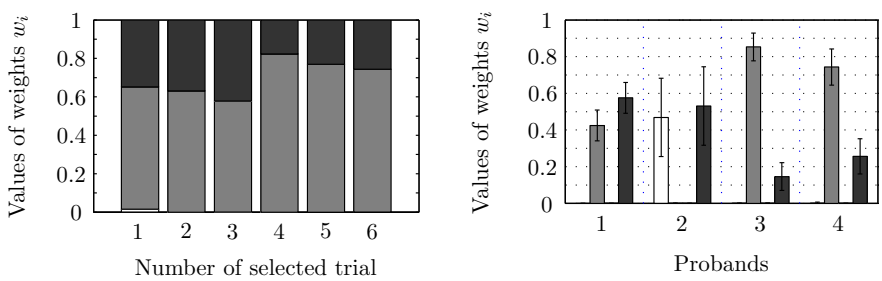

FIGURE 10. Computed optimal weight distributions for the 6 selected trials (left) and the corresponding mean distributions of all four probands (right)

The weight distributions differ between subjects when the same external mass and damper are considered; however, all by themselves are quite stable, i.e. the standard deviation is small compared to the general distribution of weights (compare figure 10). For some external influences a combination of only two optimization criteria yields optimal results, especially if the mass m is small. There is hardly one recorded trajectory where all four cost function have considerable influence.

It has to be noted that the obtained weight distributions describe only the human arm motions of the specific proband at the current state of adaption to the task. The distribution changes considerably if the virtual mass m is increased (figure 11).

For all analyzed motions the maximal distance between a recorded trajectory and the respective solution is relatively small (in most cases about 3mm, but in all cases less than 1cm); therefore, one could argue that the chosen cost functions are sufficient to reproduce the hand trajectories for the given task.

A more detailed discussion with focus on the influence of combinations of masses and dampers can be found in [3]. We thank especially C. Passenberg, who arranged the experimental setup and conducted the experiments with the human probands, for the excellent cooperation and the experimental data used in this section.

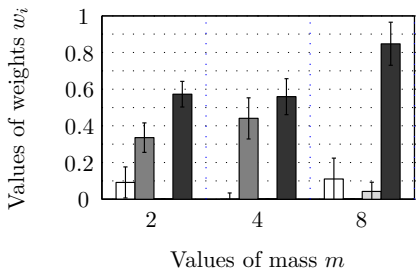

FIGURE 11. The weight distributions of one proband for different masses with constant damping $b = 0$

6. **Summary and Discussion.** The general question, which arose from invariant features of human movements, is whether humans move in some way optimal. Hence it is a common approach to seek an optimality criterion humans minimize.

This work focused on point-to-point movements, which means that at the given start and end point the arm has to be at rest. To apply optimization theory the dynamics of the human arm were modeled. Here a rigid-body-model was combined with a simple muscle model yielding an ODE description of the dynamics, which was later extended to include forces on the hand depending on velocity and acceleration.

Some standard cost functions from literature were stated and in combination with the ODE constraint a nonlinear optimization problem was obtained. A bilevel program was then obtained by composing this lower level program with an upper level program, where the optimal linear combination of the standard cost functions was sought. Bilevel theory was used to prove the existence of a global minimum for the bilevel program under the following weak assumptions: the rows of the Jacobian of the equality constraints are linearly independent, the controls have to be bounded and the lower level program has to have a unique solution.

A discretization strategy was applied to the ODE constraints of the lower level program and then the bilevel program was transformed into a one-level program by utilizing the KKT-conditions. This nonlinear optimization problem was solved with the optimizer `IPOPT` and the performance of which was enhanced by adding a further constraint.

To estimate the quality of obtained results a reconstruction framework was presented and finally results for some example data were given.

The goal of the approach presented here was not to find a cost function all humans minimize while doing point-to-point movements, but just to find that one which comes closest to the movements of one proband at a time. This specialization was necessary due to inter-subject discrepancies.

Many research groups have already worked in the field of applying force-fields to a human arm, however here the focus was not on the adaption of the human movement to the external influences but rather the goal was to describe the movements at the current state of adaption.

In the recent years the demand for human-like robots increased. The advances in hardware design have to be combined with control strategies which allow for human-robot interaction. To enable cooperating humans to better anticipate the

robots' movements, the here obtained cost functions could be used. Additionally, the results can also be utilized to predict human movements, which have to be known to plan the robots' appropriate responses. Furthermore, a prediction of the human movement can be used in telepresence to reduce the time delay.

7. Appendix.

Symbol	Quantity	$i=1$	$i=2$	Unit
m_i	Mass of bone i	1.8	1.6	[kg]
l_i	Length of bone i	0.32	0.32	[m]
$p_{cm,i}$	position of center of mass of bone i	0.16	0.16	[m]
I_i	Inertia of bone i	0.015	0.013	[kg m^2]
B_i	Damping of joint i	0.3	0.2	[Nms/rad]

TABLE 1. Parameters for the arm model

7.1. 2D Arm Model.

The analytical terms for $\boldsymbol{M}$, $\boldsymbol{\rho}$ of equation (1) and for $\boldsymbol{J}_p$ can be found in [47].

The matrix $\boldsymbol{R}$ of the moment arms is given by

$$\boldsymbol{R} = \begin{pmatrix} 0.03 & 0 & 0.025 \\ 0 & 0.03 & 0.04 \end{pmatrix}.$$

Those values are based on [47] and [11].

7.2. Upper Level Cost Function.

In this section an approach is presented to measure the distance between a recorded hand path (in $\mathbb{R}^2$ or $\mathbb{R}^3$) and a computed one. Let $\boldsymbol{p}_{\text{data}}(t)$ denote the recorded hand position at time t. The main focus should be on the curve $\Gamma_{\text{data}} = \{\boldsymbol{p}_{\text{data}}(t) | \, t \in [t_0, t_f]\}$ as a subset of $\mathbb{R}^k$ with $k=2$ or $k=3$.

For a given state vector $\boldsymbol{v}(t_i)$ the corresponding hand position is denoted by $\boldsymbol{p}_{\text{comp}}(t_i)$ and consequently the vector $\bar{\boldsymbol{p}}_{\text{comp}}$ belongs to $\bar{\boldsymbol{v}}$.

Here the computation of the cost function can be reduced to the computation of the distance of a point $\boldsymbol{p}_{\text{comp}}(t_i)$ from the curve Γ_{data}. Keeping in mind that the task $\min d\left(\boldsymbol{p}_{\text{comp}}(t_i), \Gamma_{\text{data}}\right)$ is an optimization problem of its own, the distance between points of equal path length is calculated. For that purpose we compute the relative path length l_{rel} of the hand path from the start to the point $\boldsymbol{p}_{\text{comp}}(t_i)$ with respect to the path length of the total hand movement $\bar{\boldsymbol{p}}_{\text{comp}}$, where the path is approximated by connected line segments:

$$l_{\text{rel}}(\boldsymbol{p}_{\text{comp}}(t_i)) = \frac{\sum_{j=0}^{i-1} ||\boldsymbol{p}_{\text{comp}}(t_{j+1}) - \boldsymbol{p}_{\text{comp}}(t_j)||}{\sum_{j=0}^{N-1} ||\boldsymbol{p}_{\text{comp}}(t_{j+1}) - \boldsymbol{p}_{\text{comp}}(t_j)||}.$$

Consequently, a point $\boldsymbol{\gamma}_{\text{comp}}(l)$ can be computed for a given relative path length l and likewise $\boldsymbol{\gamma}_{\text{data}}(l)$. Thus the following upper level cost function is obtained for

given l_j, $j = 1, \ldots, N-1$:

$$d\left(\boldsymbol{p}_{\text{data}}, \bar{\boldsymbol{p}}_{\text{comp}}\right) = \sum_{j=1}^{N-1} \left|\left|\boldsymbol{\gamma}_{\text{comp}}(l_j) - \boldsymbol{\gamma}_{\text{data}}(l_j)\right|\right|^2 .$$

REFERENCES

[1] P. Abbeel and A.Y. Ng, *Apprenticeship learning via inverse reinforcement learning*, in "Proc. Int. Conf. on Machine learning," (2004), 1–8.

[2] W. Abend, E. Bizzi and P. Morasso, *Human arm trajectory formation*, Brain, **105** (1982), 331–348.

[3] S. Albrecht, C. Weber, M. Sobotka, A. Peer, M. Buss and M. Ulbrich, *Optimization criteria for human trajectory formation in dynamic virtual environments*, in "Haptics: generating and perceiving tangible sensations," Lecture Notes in Computer Science, **6192** (2010), 257–262.

[4] C. Atkeson and J. Hollerbach, *Kinematic features of unrestrained vertical arm movements*, J. Neurosci., **5** (1985), 2318–2330.

[5] J.F. Bard, *An algorithm for the general bilevel programming problem*, Oper. Res., **8** (1983), 260–272.

[6] J.F. Bard, *Convex two-level optimization*, Math. Program., **40** (1988), 15–27.

[7] J.F. Bard, "Practical bilevel optimization. Nonconvex optimization and its applications," Kluwer Academic Publishers, Dordrecht, 1998.

[8] A. Billard and M. Mataric, *Learning human arm movements by imitation: evaluation of a biologically inspired connectionist architecture*, Robot. Auton. Syst., **27** (2001), 145–160.

[9] J. Bracken and J. McGill, *Mathematical programs with optimization problems in the constraints*, Oper. Res., **21** (1973), 37–44.

[10] W. Candler and R. Norton, *Multilevel Programming*, World Bank Development Research Center: Technichal Report, **20** (1977).

[11] R.F. Chandler, C.E. Claser, J.T. McConville, H.M. Reynolds and J.W. Young, *Investigation of inertial properties of the human body*, AMLR Technical Report, Wright-Patterson Air Force Base, OH (1975).

[12] B. Colson, P. Marcotte and G. Savard, *An overview of bilevel optimization*, Ann. Oper. Res., **153** (2007), 235–256.

[13] B. Corteville, E. Aertbelien, H. Bruyninckx, J. De Schutter and H. Van Brussel, *Human-inspired robot assistant for fast point-to-point movements*, in "Proc. IEEE ICRA," (2007), 3639–3644.

[14] S. Dempe, *A necessary and a sufficient optimality condition for bilevel programming problems*, Optimization, **25** (1992), 341–354.

[15] S. Dempe, "Foundations of Bilevel Programming," Kluwer Academic Publishers, Dordrecht, 2002.

[16] S. Dempe, *Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints*, Optimization, **52** (2003), 333–359.

[17] P. Deuflhard and F. Bornemann, "Scientific Computing with Ordinary Differential Equations," Springer, Berlin, 2002.

[18] M. Diehl, and D.B. Leineweber, and A.A. Schäfer, "MUSCOD-II User's Manual," IWR-Preprint, University of Heidelberg, Germany, 2001.

[19] T. Edmunds and J.F. Bard , *Algorithms for nonlinear bilevel mathematical programming*, IEEE Trans. Sys. Man. Cyb., **21** (1991), 83–89.

[20] R. Featherstone, "Robot dynamic algorithms," Kluwer Academic Publishers, Dordrecht, 1987.

[21] T. Flash and N. Hogan, *The coordination of arm movements: an experimentally confirmed mathematical model*, J. Neurosci., **5** (1985), 1688–1703.

[22] M. Gleicher, *Retargetting motion to new characters*, in "Proc. Int. Conf. on Computer graphics and interactive techniques," (1998), 33–42.

[23] E. Hairer, Norsett and G. Wanner, "Solving Ordinary Differential Equations," Springer, New York, 1993.

[24] C. Harris and D. Wolpert, *Signal-dependend noise determines motor planning*, Nature, **394** (1998), 780–784.

[25] H. Hatze, *Neuromusculoskeletal control systems modeling - a critical survey of recent developments*, IEEE Trans. Autom. Control., **25** (1980), 375–385.

[26] A. Hill, *The heat of shortening and the dynamic constants of muscle*, Proc. Roy. Soc., **126** (1938), 136–195.

[27] K.W. Lilly, "Efficient dynamic simulation of robotic mechanisms," Kluwer Academic Publishers, Dordrecht, 1993.

[28] C.K. Liu, A. Hertzmann and Z. Popovic, *Learning physics-based motion style with nonlinear inverse optimization*, ACM Trans. Graph., **24** (2005), 1071–1081.

[29] Y. Maeda, T. Hara and T. Arai, *Human-robot cooperative manipulation with motion estimation*, in "Proc. IEEE IROS," **4** (2001), 2240–2245.

[30] O. Marcotte and G. Savard, *Combinatorial nature of bilevel programs reviewed*, in "Graph theory and combinatorial optimization" (eds. D. Avis, A. Hertz, and O. Marcotte), Springer, New York (2005).

[31] K. Mombaur, A. Truong and J.-P. Laumond, *From human to humanoid locomotion- an inverse optimal control approach*, Auton. Robot., **28** (2010), 369–383.

[32] P. Morasso, *Spatial control of arm movements*, Exp. Brain. Res., **42** (1981), 223–227.

[33] P. Morasso, *Three dimensional arm trajectories*, Biol. Cybern., **48** (1983), 187–194.

[34] E. Nakano, H. Imamizu, R. Osu, Y. Uno, H. Gomi, T. Yoshioka and M. Kawato, *Quanitative examinations of internal representations for arm trajectory planning: Minimum commanded torque change model*, J. Neurophysiol., **81** (1999), 2140–2155.

[35] W. Nelson, *Physical principles for economies of skilled movements*, Biol. Cybern., **46** (1983), 135–147.

[36] A.Y. Ng and S. Russell, *Algorithms for inverse reinforcement learning*, in "Proc. Int. Conf. on Machine learning," (2000), 663–670.

[37] J. Nocedal and S. Wright, "Numerical Optimization," Springer, New York, 2006.

[38] J.V. Outrata, *On the numerical solution of a class of Stackelberg problems*, Z. Oper. Res., **34** (1990), 255–277.

[39] J.V. Outrata, M. Kocvara and J. Zowe, "Nonsmooth approach to optimization problems with equilibrium constraints," Kluwer Academic Publishers, Dordrecht, 1998.

[40] M.J. Powell, *The BOBYQA algorithm for bound constrained optimization without derivatives*, in "Cambridge NA Report NA2009/06," University of Cambridge, Cambridge, 2009.

[41] D. Ramachandran and E. Amir, *Bayesian inverse reinforcement learning*, in "Proc. Int. Conf. on Artificial intelligence," (2007), 2586–2591.

[42] N.D. Ratliff, J.A. Bagnell and M.A. Zinkevich, *Maximum margin planning*, in "Proc. Int. Conf. on Machine learning," (2006), 729–736.

[43] S. Russell, *Learning agents for uncertain environments (extended abstract)*, in "Proc. Int. Conf. on Computational learning theory," (1998), 101–103.

[44] S. Schaal, *Is imitation learning the route to humanoid robots*, Trends in cognitive sciences, **3** (1999), 233–242.

[45] K. Shimizu and M. Lu, *A global optimization method for the stackelberg problem with convex functions via problem transformations and concave programming*, IEEE Trans. Sys. Man. Cyb., **25** (1995), 1635–1640.

[46] C. Smith and H.I. Christensen, *A minimum jerk predictor for teleoperation with variable time delay*, in "Proc. IEEE IROS," (2009), 5621–5627.

[47] S. Ströve, *Impedance characteristics of neuromusculoskeletal model of the human arm*, Biol. Cybern., **81** (1999), 475–494.

[48] W. Suleiman, E. Yoshida, F. Kanehiro and J.P. Laumond, *On human motion imitation by humanoid robot*, in "Proc. IEEE ICRA," (2008), 2697–2704.

[49] E. Todorov, *Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system*, Neural. Comput., **18** (2005), 1084–1108.

[50] E. Todorov, *Optimality principles in sensorimotor control*, Nature Neurosci., **7** (2004), 907–915.

[51] Y. Uno, M. Kawato and R. Suzuki, *Formation and control of optimal trajectory in human multijoint arm movements*, Biol. Cybern., **61** (1989), 89–101 .

[52] Y. Uno, R. Suzuki and M. Kawato, *Minimum muscle-tension-change model which reproduces human arm movement*, in "Proc. 4th Symp. Biol. Physiol. Eng.," (1989), 299–302.

[53] L.N. Vicente and P.H. Calamai, *Bilevel and multilevel programming: a bibliography review*, J Global. Optim., **5** (1994), 291–306.

[54] A. Wächter and L.T. Biegler, *Line search filter methods for nonlinear programming: motivation and global convergence*, SIAM J. Optim., **16** (2005), 1–31.

[55] A. Wächter and L.T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Math. Program., **106** (2006), 25–57.
[56] D. Winter "Biomechanics and motor control of human movement," Wiley, New York, 2005.
[57] J. Winters and L. Stark, *Analysis of fundamental human movement patterns through the use of in-depth antagonistic muscle models*, IEEE Trans. Biomedical Eng., **10** (1985), 826–839.
[58] J.J. Ye and D.L. Zhu, *Optimality conditions for bilevel programming problems*, Optimization, **33** (1995), 9–27.
[59] T. Yoshikawa, "Foundations of robotics: analysis and control," MIT Press, Cambridge, 1990.

Received xxxx 20xx; revised xxxx 20xx.

E-mail address: `albrecht@ma.tum.de`
E-mail address: `marion.sobotka@tum.de`
E-mail address: `mulbrich@ma.tum.de`