Learning Motor Control Parameters for Motion Strategy Analysis of Parkinson's Disease Patients

Felix Burget Christoph Maurer Wolfram Burgard Maren Bennewitz

***Abstract*— Although the neurological impairments of Parkinson's disease (PD) patients are well known to go along with motor control deficits, e.g., tremor, rigidity, and reduced movement, not much is known about the motor control parameters affected by the disease. In this paper, we therefore present a novel approach to human motions analysis using motor control strategies with joint weight parameterization. We record the motions of healthy subjects and PD patients performing a hand coordination task with the whole-body XSens MVN motion capture system. For our motion strategy analysis we then follow a two step approach. First, we perform a complexity reduction by mapping the recorded human motions to a simplified kinematic model of the upper body. Second, we reproduce the recorded motions using a Jacobian weighted damped least squares controller with adaptive joint weights. We developed a method to iteratively learn the joint weights of the controller with the mapped human joint trajectories as reference input. Finally, we use the learned joint weights for a quantitative comparison between the motion control strategies of healthy subjects and PD patients. Other than expected from clinical experience, we found that the joint weights are almost evenly distributed along the arm in the PD group. In contrast to that, the proximal joint weights of the healthy subjects are notably larger than the distal ones.**

I. Introduction

Within the last decade, recording and analyzing human motion data has gained an increased interest in a variety of research fields, ranging from medical science, neuroscience, computer graphics, to robotic applications. The way the data is used, however, differs between these fields. The former fields are mainly interested in understanding human motion and its underlying principles in order to improve therapy methods for patients with neurological or physiological diseases, whereas computer graphics and robotics aims at generating human-like motions for artificial multi-joint robotic systems to improve the appearance, coexistence, collaboration, and safety in human-robot interaction scenarios. So far, robotic research has used human motion data to map hand/end-effector and joint trajectories to robotic platforms for tele-operation applications or applied the data as reference input for motion planning algorithms and control schemes, rather than investigating the underlying motion control strategy that generated the observed human trajectories. Altogether, these approaches have in common that they consider the human motion samples as a desirable, natural human kinesic behavior to be adopted by robotic platforms or reflected by schemes modelling the human musculoskeletal system. Contrary, we aim in this work towards analyzing the distinction between the human motion samples and separate them into different groups, namely the motion of healthy subjects and Parkinson's disease (PD) patients whose motor control is affected by a neurological disorder. Note that the end-effector trajectories of different subjects performing the same task, e.g., moving an object from one location to another, are typically similar, whereas the joint trajectories generated by the individual motor control scheme might differ significantly depending on the constitution of the subject.

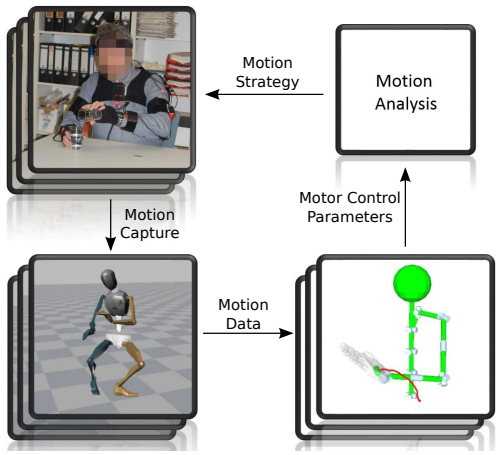

Fig. 1. Motion strategy analysis: experimental setup for data collection (top left), motion representation (bottom left), and motion model used to learn the motor control parameters (bottom right).

In this paper, we present a novel approach to investigate the hypothesis that members of the respective group share a common motor control strategy to select among the infinite set of joint trajectory solutions achieving a given task. We focus our analysis on the hand coordination task of pouring water from one glass into another, as depicted in Fig. 1. Our approach relies on motion data of healthy and PD subjects collected using a whole-body motion capture suit. In a first step, we map the data to a simplified scale-adaptive artificial model of the human upper body in order to perform dimensionality reduction. Afterwards, we use this model to track the mapped human end-effector trajectories based on a variable damped least squares control scheme with adaptive joint weights. The joint weight parameterization of the control scheme allows executing the same end-effector trajectory with arbitrary joint trajectories that, in turn, reflect different

All authors are with the *BrainLinks-BrainTools* Cluster of Excellence, Univ. of Freiburg, Germany. F. Burget and W. Burgard are with the Institute of Computer Science, University of Freiburg, Germany. C. Maurer is with the University Medical Center, University of Freiburg, Germany. M. Bennewitz is with the Institute of Computer Science, University of Bonn, Germany. This work has been supported by the German Research Foundation (DFG) within the Cluster of Excellence *BrainLinks-BrainTools* (EXC1086).

motor control strategies. To determine the respective motor control strategy for each subject, we developed a technique to iteratively learn the joint weights of the controller so as to match the observed joint trajectories. Based on the resulting joint weights, we carry out a quantitative comparison of the differences between the motions of PD patients and healthy subjects and infer their motion control strategy principles. According to our results, it turns out that we can differentiate between two motor control strategies, referred to as the *proximal* and *distributed* motion strategy, adopted by the two groups for task achievement. To the best of our knowledge, this is the first approach learning indicative motor control parameters in order to explain the effects of neurological diseases onto the musculoskeletal system in human subjects.

II. Related Work

Previous approaches dealing with human motion analysis can be subdivided into two main categories. The first category deals with the segmentation of the motion data into different actions or emotions. Approaches from the second category try to build computational models reflecting the underlying principles of human motion through optimization of different criteria or objective functions. In this work, both categories are relevant because the motor control deficits of PD patients can be interpreted as a result of a specific permanent emotional state or a motor control strategy following an objective function that is different from the one adopted by healthy subjects. In the following, we discuss representative approaches for each category.

Rahimi *et al.* [1] presented an approach that uses principal component analysis (PCA) to identify kinematic variables that best represent mobility tasks performed by PD patients. This method uses motion data of patients at different stages of PD recorded in their home environments using a full-body motion capture suit. Subsequently, the data were analyzed to determine possible variability between tasks, subjects, and trials. The results, however, state that no specific movement profile among patients for each task has been found.

Das *et al.* [2] use a support vector machine (SVM) to discriminate mild vs. severe Parkinson's disease symptoms. The authors recorded motions of PD patients performing various motor control tasks and trained a motor task specific SVM classifier based on different sets of features. Das *et al.* report an average classification accuracy of approximately 90%.

Day *et al.* [3] tested the hypothesis that predictive motor behavior is abnormal in Parkinson's disease by recording the performance of healthy and PD subjects tracking a repeated and an unpredictable pattern of a moving spot with their hand. Despite of the obvious motor control deficits of the patients, the authors found that their tracking performace, evaluated w.r.t. the measured tracking lags, is comparable to that of healthy subjects.

Barbič *et al.* [4] investigated three techniques for automatic segmentation of motion capture data into distinct actions, e.g., walking, drinking, or sitting down. The two presented online segmentation techniques are PCA and probabilistic PCA. The third approach is a batch process using Gaussian mixture models for segmentation. All methods achieved good results in the experiments, though probabilistic PCA has found to provide the overall best performance. Based on the work in [4], Zhou *et al.* [5] proposed aligned cluster analysis, an extension of standard kernel *k*-means clustering for temporal segmentation of human motion data into actions. Here, the extension allows a variable number of features in the cluster means and the use of a dynamic time warping kernel to achieve temporal invariance. In a further extension [6], Zhou *et al.* developed an approach to implement a hierarchical decomposition of human motion data, where actions such as running or walking can be further decomposed into motion primitives of smaller temporal scale.

Cimen *et al.* [7] presented a technique using a set of posture, dynamic, and frequency-based descriptors for emotion classification of motion data. Based on different feature combinations, this approach applies a SVM learning algorithm to classify recorded motions into four distinct emotional states. Aristidou *et al.* [8] propose a method to automatically extract motion qualities from dance performances, in terms of laban movement analyses (LMA) for motion analysis and indexing purposes. Using the four LMA components body, effort, shape, and space, the authors analyze correlations between the performer's acting emotional states.

Campos *et al.* [9] provided an overview of human arm movement control theories and the different paradigms that have been used in modelling arm control. The authors distinguish between descriptive, dynamic, stochastic and motor execution models and analyze their relevance for rehabilitation practices. Flash *et al.* [10] presented an approach for modelling voluntary human arm movements mathematically by defining an objective function representing the rate of change of acceleration. By minimizing the objective function using dynamic optimization, the method predicts trajectories for point-to-point and curved motions that resemble the observed motions of human subjects. Based on this work, Todorov *et al.* [11] proposed a novel mathematical model that accurately predicts the speed profiles of a human arm in straight reaching and extemporaneous drawing movements. The results indicate that the relationship between end-effector path and speed profile of a complex arm movement is stronger than previously thought.

Albrecht *et al.* [12] developed an approach that uses physically inspired optimization principles describing a human's motion based on bilevel optimization methods. These principles are subsequently used to generate reaching motion trajectories for a humanoid robot that are similar to the recorded human behavior.

In contrast to all the above approaches, we aim at investigating the different motor control strategies adopted by healthy subjects and patients affected by motor control deficits. We hereby rely on the optimization of the joint weights for a weighted damped least squares controller so that the generated motion matches the observed reference motions.

Fig. 2. XSens MVN motion capture suit used to record motions of healthy and PD subjects performing clinical experiments.

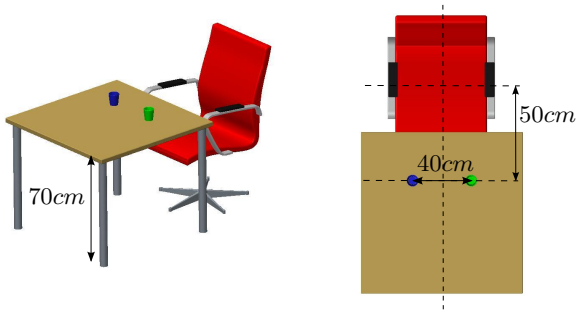

Fig. 3. Hand coordination task: Subjects are asked to pour water from one glass (green, at the left of the subject) into another, empty glass (blue).

III. Motion Database

The basis of our motor control analysis is a database of motions recorded from healthy and PD subjects using a whole-body motion capture suit (XSens MVN [13]) equipped with 17 inertial measurement units (IMU), shown in Fig. 2.

A. Motion Data

The IMU sensor data of the suit is sampled at a frequency rate of 120Hz and mapped to an artificial human avatar composed of 23 segments and 22 joints. Since we are investigating motor control deficits of PD patients in hand coordination tasks, we are using only the data of the upper body, i.e., the trajectories of the spine and arm joints.

B. Motor Control Task

To analyze the motions of PD patients and compare them to motions of healthy subjects, we set up a task, where water needs to be poured from one glass into another using the left hand (see Fig. 3). The task has been designed such that the resulting motions are composed of two sections, the coarse subtask of transitioning the glass filled with water to the empty glass and the delicate subtask of pouring the water from one glass into the other without spilling.

IV. Motion Representation

The intrinsic model of the motion capture system for representing the motions recorded from the upper body of subjects, i.e., the spine and arm kinematic chain, is composed of 8 spherical joints, and thus 24 degrees of freedom (DOF).

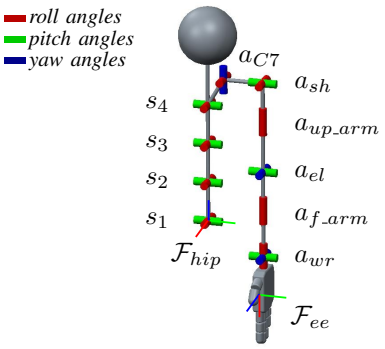

Fig. 4. Artificial kinematic model of the human upper body.

Empirical analysis of the recorded data, however, revealed that contrary to the model presented some human joints have less than three rotation axes, thus resulting in only 19 DOF actively contributing to the observed motions. Due to this fact, we built a simplified artificial model used as a compact representation for the motions of a human's upper body in hand coordination tasks.

A. Artificial Model of the Human Upper Body

The artificial kinematic model, shown in Fig. 4, is composed of the previously mentioned 19 joints dominating the execution of the hand coordination task. Here, the first 8 joints represent the motion of the spine and the remaining 11 joints the motion of the kinematic arm chain. The configuration of the entire model is defined by the following vector of joint angles

$$\boldsymbol{q}^h = (\boldsymbol{q}_{spine}, \boldsymbol{q}_{arm})^T, \tag{1}$$

with

$$\boldsymbol{q}_{spine} = (s_1^R, s_1^P, s_2^R, s_2^P, s_3^R, s_3^P, s_4^R, s_4^P), \tag{2}$$

$$\begin{aligned}\boldsymbol{q}_{arm} = (&a_{C7}^R, a_{C7}^Y, a_{sh}^P, a_{sh}^R, a_{up\_arm}^R,\\ &a_{el}^Y, a_{el}^P, a_{f\_arm}^R, a_{wr}^Y, a_{wr}^P, a_{wr}^R),\end{aligned} \tag{3}$$

where $\boldsymbol{q}_{spine}$ and $\boldsymbol{q}_{arm}$ denote the configurations of the spine and arm chain, respectively. Here, the vector elements s_j^k, a_j^k refer to the roll, pitch, and yaw angle $k \in \{R, P, Y\}$ of joint j.

B. Motion Mapping

In order to map the recorded motions to our simplified representation, we assign the joint trajectories $\boldsymbol{q}_i^h$ of the 19 dominant human joints to the respective joints of our artificial model. Furthermore, we record the trajectory of the hand $\boldsymbol{x}_e^h = (\boldsymbol{x}_e, \dot{\boldsymbol{x}}_e)^T$, referred to as the end-effector in the following, over the entire motion sequence. The end-effector pose trajectory $\boldsymbol{x}_e$ of frame $\mathcal{F}_{ee}$ expressed w.r.t. the fixed frame $\mathcal{F}_{hip}$ is obtained by solving the forward kinematics for each configuration $\boldsymbol{q}^h(t)$ captured by the system at time t. The end-effector velocity trajectory $\dot{\boldsymbol{x}}_e$ is determined by differentiating the end-effector pose trajectory with respect to time (Δt = 8ms for a sampling rate of 120Hz).

We use the same kinematic representation with the trajectory information $\boldsymbol{x}_e^h$ and $\boldsymbol{q}^h$ in the following to implement a

mathematical model that learns the underlying characteristics of the motions in terms of motor control parameters.

V. Learning Motor Control Parameters

When applying a mathematical model for motor control there exist infinite solutions of joint trajectories that achieve the desired human end-effector trajectory $\boldsymbol{x}_e^h$. A common approach is to select a solution, generated through the optimization of a specific objective function, that is assumed to mirror the underlying motor control principles and thus the resulting joint motions $\boldsymbol{q}^h$ adopted by human beings. While the minimum-jerk model [11] has found to yield a close fit to natural human arm motions, it does not allow to make any statements concerning the differences between the motion of two different subjects. The motor control analysis presented in this work relies on an iterative method for fitting a mathematical model to the observed human motion by adapting motor control parameters based on joint trajectory error information.

A. End-Effector Trajectory Tracking

As a first step, our approach reproduces the recorded end-effector trajectory $\boldsymbol{x}_e^h$ using a mathematical model, i.e., a controller, for motion generation. A classical approach from the literature to generate a desired end-effector path for a kinematic structure is based on the inverse differential kinematics control scheme [14]. In this approach, a desired end-effector trajectory $\boldsymbol{x}_d$ is tracked by numerical integration of joint velocities over a given interval Δt with the initial configuration $\boldsymbol{q}(0)$. The joint positions required at time t_{k+1} to move the end-effector along the desired trajectory from pose $\boldsymbol{x}_d(t_k)$ to $\boldsymbol{x}_d(t_{k+1})$ are computed as

$$\boldsymbol{q}(t_{k+1}) = \boldsymbol{q}(t_k) + \dot{\boldsymbol{q}}(t_k)\Delta t, \tag{4}$$

with

$$\dot{\boldsymbol{q}}(t) = \mathbf{J}^{-1}(\boldsymbol{q}(t))(\dot{\boldsymbol{x}}_d(t) + \mathbf{K}\boldsymbol{e}(t)), \tag{5}$$

where $\mathbf{J}^{-1}$ is the Jacobian inverse evaluated in configuration $\boldsymbol{q}$, $\dot{\boldsymbol{x}}_d$ the desired velocity along the end-effector path, and $\mathbf{K}$ a positive definite diagonal gain matrix whose scalar values can be chosen so as to give individual weights to the components of the error $\boldsymbol{e}$. Note that the time dependency of the variables is omitted from now on in favour of a compact notation. The operational space error between the desired $\boldsymbol{x}_d$ and the actual end-effector position and orientation $\boldsymbol{x}_c$ is denoted as $\boldsymbol{e}$ and is defined as follows

$$\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{x}_c. \tag{6}$$

$\boldsymbol{e}$ accounts for the numerical drift of the solution involved in the integration process in Eq. (4). This ensures that the end-effector pose corresponding to the computed joint variables matches the desired one. Inversion of the Jacobian, however, is only feasible if the number of operational space variables r is equal to the number of joint space variables n, i.e., if $\mathbf{J}$ is a square matrix. When $r < n$, as in our case where $r = 6$ and $n = 19$, a manipulator is said to be redundant and we need to refer to a modified control scheme. A solution scheme for redundant manipulators is obtained by

$$\dot{\boldsymbol{q}} = \tilde{\mathbf{J}}^{\dagger}(\dot{\boldsymbol{x}}_d + \mathbf{K}\boldsymbol{e}) + (\mathbf{I}_n - \tilde{\mathbf{J}}^{\dagger}\mathbf{J})\dot{\boldsymbol{z}}, \tag{7}$$

where the Jacobian inverse in Eq. (5) has been replaced with the damped least squares pseudoinverse, defined as

$$\tilde{\mathbf{J}}^{\dagger} = \mathbf{J}_{dls}^{\dagger} = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T + \lambda^2\mathbf{I}_r)^{-1}. \tag{8}$$

The term λ^2 represents a dynamic damping factor used for stabilization of the solution in the vicinity of kinematic singularities, where the Jacobian becomes ill-conditioned from a numerical viewpoint. In accordance with [15], we use the following definition

$$\lambda^2 = \begin{cases} 0, & \text{if } \sigma \geq \epsilon \\ (1 - (\frac{\sigma}{\epsilon})^2)\lambda_{max}^2, & \text{if } \sigma < \epsilon \end{cases} \tag{9}$$

where σ is the manipulability measure evaluated at each configuration [16], λ_{max} is the maximum damping factor, and ϵ is the activation threshold, respectively. Using an activation threshold ensures that damping is only applied when needed. Moreover, we consider a second term in Eq. (7) projecting a gradient $\dot{\boldsymbol{z}}$ into the null-space of the inverse differential kinematics solution. This gradient can be used to fulfill additional tasks without perturbing the end-effector trajectory tracking performance. Here, we choose $\dot{\boldsymbol{z}}$ such that the joint values of our model are kept within the range of human joints, thus respecting the natural kinematic constraints of the human musculoskeletal system. Following the approach presented in [17], we define the gradient as follows

$$\dot{z}_i = \begin{cases} \frac{q_i - \tilde{q}_{i_{max}}}{\Delta\bar{q}_i}, & \text{if } q_i > \tilde{q}_{i_{max}} \\ \frac{q_i - \tilde{q}_{i_{min}}}{\Delta\bar{q}_i}, & \text{if } q_i < \tilde{q}_{i_{min}} \\ 0, & \text{else,} \end{cases} \tag{10}$$

where $\dot{z}_i$ is the joint limit gradient for the i-th joint. Here, q_i indicates the current joint value and $\Delta\bar{q}_i = \bar{q}_{i_{max}} - \bar{q}_{i_{min}}$ its respective absolute joint range. The variables $\tilde{q}_{i_{min}}$ and $\tilde{q}_{i_{max}}$ are used to select an upper and lower threshold for the joint limit gradient activation defined as

$$\tilde{q}_{i_{min}} = \bar{q}_{i_{min}} + \gamma\Delta\bar{q}_i, \tag{11}$$

$$\tilde{q}_{i_{max}} = \bar{q}_{i_{max}} - \gamma\Delta\bar{q}_i, \tag{12}$$

with $\gamma \in [0.0, 0.5]$. To track the human end-effector trajectory with the damped least squares control scheme we set $\boldsymbol{x}_d = \boldsymbol{x}_e$ and $\dot{\boldsymbol{x}}_d = \dot{\boldsymbol{x}}_e$ and choose the first configuration of the recorded motion $\boldsymbol{q}^h(0)$ at time $t = 0$ as the initial configuration $\boldsymbol{q}(0)$ in our control setup.

B. Adaptive End-Effector Trajectory Tracking

The joint trajectories generated by Eq. (7) follow from the minimization of a specific cost functional and represent only one possible solution to track a desired end-effector trajectory $\boldsymbol{x}_e^h$. Alternative solutions can be obtained by adding further objective functions, optimizing the motion with respect to different criteria. In this work, we follow the approach of Schinstock *et al.* [18] that parameterizes

Algorithm 1: Joint Weights Learning ($\boldsymbol{q}^h$, $\boldsymbol{x}_e^h$, $rmse_{thr}$, $\Delta rmse_{thr}$)

1 $\boldsymbol{w}$, $\boldsymbol{w}^{inc}$, $\boldsymbol{w}^{grad} \leftarrow$ INIT_JOINT_WEIGHTS()
2 **for** $i = 1$ **to** *max_iter* **do**
3 *j_idx* $\leftarrow 0$
4 **while** *j_idx != num_joints* **do**
5 $\boldsymbol{q}^c \leftarrow$ RUN_J_WDLS_CONTROL($\boldsymbol{w}$, $\boldsymbol{x}_e^h$, $\boldsymbol{q}^h(0)$)
6 *w_update* $\leftarrow$ UPDATE_JOINT_WEIGHT($\boldsymbol{q}^c_{j\_idx}$, $\boldsymbol{q}^h_{j\_idx}$, $w^{grad}_{j\_idx}$, $rmse_{thr}$, $\Delta rmse_{thr}$)
7 **if** *(w_update = FALSE)* **then**
8 *j_idx* $\leftarrow$ *j_idx* + 1
9 **end**
10 **end**
11 **end**
12 **return** $\boldsymbol{w}$

Algorithm 2: UPDATE_JOINT_WEIGHT ($\boldsymbol{q}_i^c$, $\boldsymbol{q}_i^h$, w^{grad}, $rmse_{thr}$, $\Delta rmse_{thr}$)

1 $rmse_i^t \leftarrow$ COMPUTE_JOINT_TRAJECTORY_ERROR($\boldsymbol{q}_i^c$, $\boldsymbol{q}_i^h$)
2 $\Delta rmse_i^t \leftarrow rmse_i^t - rmse_i^{t-1}$
3 **if** *(*$rmse_i^t < rmse_{thr}$ or $\Delta rmse_i^t < \Delta rmse_{thr}$*)* **then**
4 **return** *FALSE*
5 **else**
6 **if** *(*$\Delta rmse_i^t > 0$*)* **then**
7 $w_i^{grad} \leftarrow -w_i^{grad}$
8 $w_i^{inc} \leftarrow$ *inc_scale_factor* $* w_i^{inc}$
9 **end**
10 $w_i \leftarrow w_i + (w_i^{grad} * w_i^{inc})$
11 **return** *TRUE*
12 **end**

the control scheme using joint weights in order to be able to generate arbitrary joint trajectory solutions. To do so, we use in Eq. (7) an extended variant of $\mathbf{J}^{\dagger}_{dls}$, referred to as the weighted damped least squares pseudoinverse

$$\tilde{\mathbf{J}}^{\dagger} = \mathbf{J}^{\dagger}_{wdls} = \mathbf{J}_w^T(\mathbf{J}_w\mathbf{J}_w^T + \lambda^2\mathbf{I}_r)^{-1}, \tag{13}$$

with

$$\mathbf{J}_w = \mathbf{J}\mathbf{W}_q, \tag{14}$$

where $\mathbf{W}_q$ is a $n \times n$ diagonal matrix of joint weights for a kinematic model such as the one depicted in Fig. 4, defined as

$$\mathbf{W}_q = \mathsf{diag}(w_1, w_2, \ldots, w_n). \tag{15}$$

Solving Eq. (7) using weighted damped least squares yields $\dot{\boldsymbol{q}}_{\boldsymbol{w}}$, from which an approximated solution is obtained by

$$\dot{\boldsymbol{q}} = \mathbf{W}_q\dot{\boldsymbol{q}}_w. \tag{16}$$

Such a parameterization of the control law allows for modeling different motor control strategies that all track the same desired end-effector trajectory. In case of $\mathbf{W}_q$ being the identity matrix, Eq. (13) coincides with the damped least squares solution defined in Eq. (8). On the other hand, lower weights can be chosen for the joints of the spine to generate a motion that is dominated by the joints of the arm kinematic chain. In the following, we present an iterative scheme to determine the values for the joint weights w_i, required to replicate the motions recorded from healthy and PD subjects as close as possible.

C. Joint Weights Learning

The approach to determine the joint weights, given a human end-effector trajectory $\boldsymbol{x}_e^h$ and joint trajectory solution $\boldsymbol{q}^h$, is described as pseudocode in Alg. 1. The additional input parameters $rmse_{thr}$ and $\Delta rmse_{thr}$ define thresholds for the error between the control-based and the recorded human joint trajectories. In Line 1 of Alg. 1 the joint weight $\boldsymbol{w}$, weight increment $\boldsymbol{w}^{inc}$ and gradient direction $\boldsymbol{w}^{grad}$ vectors are initialized. Afterwards, max_iter runs of the joint weights learning algorithm are performed, where each iteration corresponds to the stepwise optimization of all joints weights along the kinematic chain of our model. At the beginning of each iteration the index j_idx is set to 0, indicating that we start the weights optimization process for the first joint of the spine segment, i.e., s_1^R in Eq. (2). Using the initial configuration $\boldsymbol{q}^h(0)$ and the current joint weights $\boldsymbol{w}$, we track the human end-effector trajectory $\boldsymbol{x}_e^h$ by running the weighted damped least squares control scheme presented in Sec. V-B and obtain the joint trajectory solution $\boldsymbol{q}^c$ (Line 5 of Alg. 1). Subsequently, the joint trajectories $\boldsymbol{q}^c_{j\_idx}$ and $\boldsymbol{q}^h_{j\_idx}$ of the j_idx-th joint, generated by the controller and the human, respectively, are used as input for the UPDATE_JOINT_WEIGHT function, described in Alg. 2. Here, we determine in a first step the root-mean-square error $rmse_i^t$ between the trajectories $\boldsymbol{q}_i^c$ and $\boldsymbol{q}_i^h$ of joint i and the change of that error $\Delta rmse_i^t$ with respect to the one computed in the previous iteration $rmse_i^{t-1}$ (Lines 1 and 2 of Alg. 2). If $rmse_i^t$ or $\Delta rmse_i^t$ is below the thresholds $rmse_{thr}$ or $\Delta rmse_{thr}$, respectively, the algorithm returns *FALSE*, indicating that the similarity between the control-based and human solution for the motion of joint i has either reached a satisfactory level or cannot be further improved through joint weight adaption (Lines 3 and 4 of Alg. 2). Otherwise, we evaluate whether the root-mean-square error has been decreased by the last joint weight modification performed at $t-1$ (Line 6 of Alg. 2). If that is not the case, the joint weight gradient direction $w_i^{grad} \in \{1, -1\}$ is switched and the current joint weight increment w_i^{inc} is reduced by multiplying it with a constant factor inc_scale_factor (Lines 7 and 8 of Alg. 2). This factor helps to avoid undesired oscillation of the joint weight and to ensure convergence of the learning algorithm. If the joint weight update performed in the last iteration, has improved the similarity between the control-based and human joint trajectory the values of the variables w_i^{grad} and w_i^{inc} remain the same. In a final step, the weight w_i of joint i is increased or decreased depending on the current gradient direction w_i^{grad} and *TRUE* is returned, indicating that the joint weight has been modified (Lines 10 and 11 of Alg. 2).

Depending on the weight update status w_update returned by the UPDATE_JOINT_WEIGHT function, the joint weights learning algorithm proceeds in two different ways (Line 6 of Alg. 1). If $w\_update =$ *TRUE*, the weighted damped least squares controller is run again with the new joint weight w_i followed by another joint trajectory error evaluation. On the other hand, if $w\_update =$ *FALSE*, the

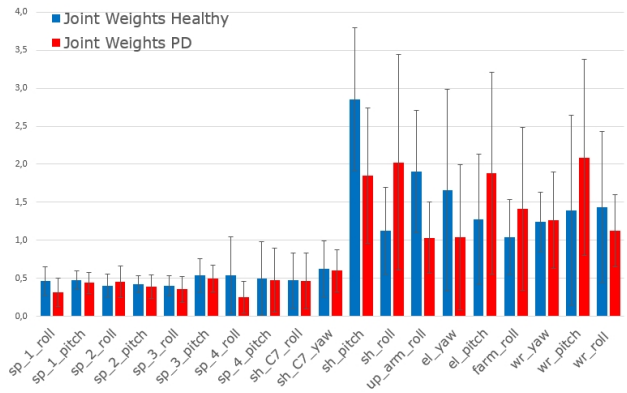

Fig. 5. Average joint weights learned for the healthy (blue) and PD (red) subject group. The y-axis values correspond to the final weights for the joints indicated on the x-axis.

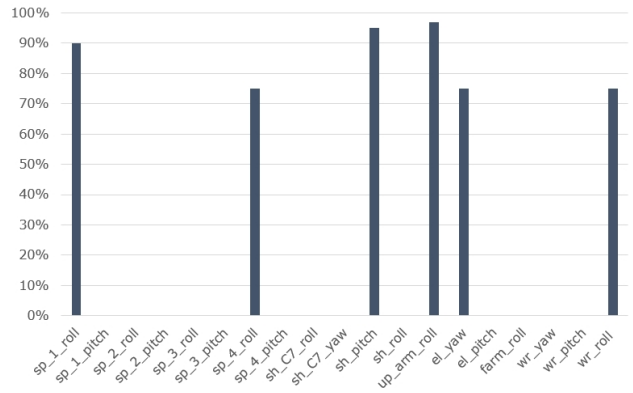

Fig. 6. Results from the *independent sample t-test*: Comparing the joint weight means of the healthy and PD group against each other.

variable j_idx is incremented to consider the weight of the next joint along the chain for the optimization process (Lines 7 and 8 of Alg. 1). When the maximum number of iterations max_iter is reached the final joint weight vector $\boldsymbol{w}$ learned for the subject is returned (Line 12 of Alg. 1). Note, that the joint trajectories generated by the control scheme are not independent from each other, i.e., an improvement in similarity to the human motion achieved for a specific joint trajectory solution $\boldsymbol{q}_i^c$ by modification of the respective joint weight w_i may deteriorate the joint trajectory solution $\boldsymbol{q}_j^c$ of another joint j. Therefore, we run the entire weight optimization process max_iter times to find a compromise between conflicting joint weights, mutually deteriorating each others similarity to the recorded human joint trajectory solutions.

VI. Implementation Details

The artificial human upper body model is generated in ROS (Robot Operating System) by defining an *urdf* file (Universal Robot Description Format). For recording the human end-effector trajectories we perform forward kinematics using the KDL library [19]. The error gain matrix $\mathbf{K}$ used in Eq. (5) is set to the identity. For the joint limits avoidance task, we set the activation parameter $\gamma = 0.2$. The parameters used to determine the damping factor in Eq. (9) are set to $\lambda_{max} = 0.04$ and $\epsilon = 0.0008$. For the weight learning algorithm, we assign an initial value of 1 to all vector elements of $\boldsymbol{w}$ and $\boldsymbol{w}^{grad}$. The weight increment $\boldsymbol{w}^{inc}$ is set to 0.8 for all joints. For the weight reduction factor we use $inc\_scale\_factor = 0.8$ and for the joint trajectory error thresholds $rmse_{thr} = 0.02$ and $\Delta rmse_{thr} = 0.001$. In total we perform $max\_iter = 10$ runs of the weight optimization process.

VII. Experiments

In the following, we present experimental results for a database composed of motions recorded from healthy and PD subjects. The results include an evaluation of the trajectory tracking performance of our controller as well as an analysis of the joint weights learned by our algorithm for two different groups, PD patients and healthy subjects.

A. Experimental Setup

The two groups were recruited from the clinics movement disorders outpatient clinic for the study of the hand coordination task, described in Sec. III-B. All participants gave their written informed consent and their data was pseudonomized at study inclusion, all in accordance with the Helsinki Declaration and to the local ethics committee (Ethikkommission der Medizinischen Fakultät der Ludwig-Maximilians-Universität).

Six PD patients participated in this study. They were 2 female and 4 male ranging from 48 to 74 (mean 65) years of age. None had any additional disorder influencing postural control. Patients were on their regular medication in ON state, 2 had deep brain stimulation. Half of the patients had pathological side *left* and *right*, respectively. The momentary state of patients mobility was assessed just prior to the experiment with the Unified Parkinsons Disease Rating Scale (UPDRS mean 22.33±12.94 SD). Six control subjects were recruited from relatives of the authors and (former) university personnel, 3 female and 3 male ranging from 48 to 62 (mean 57) years of age. None had history of neurological disorders of any sort or orthopaedic disorders requiring surgery or regular medication. All subjects were right-handed. Each subject performed six repetitions of the task, leading to an overall database of 36 motions for each group.

B. Trajectory Tracking Performance

The mean Cartesian tracking error between the recorded human end-effector trajectories $\boldsymbol{x}_e^h$ and the associated end-effector trajectories obtained from the weighted damped least squares control scheme using the final joint weights $\boldsymbol{w}$, learned from Alg. 1, is 0.64 cm. Using the final joint weights, the mean residual error between the control-based and the corresponding human reference joint trajectories is found to be $0.029\,\mathrm{rad}$, i.e., $1.69°$.

C. Motion Strategy Analysis

The primary goal of our work is to investigate the hypothesis that healthy subjects and PD patients follow two different motion strategies to achieve the same task. Here, we consider

the joint weights as indicative motor control parameters from which we want to infer underlying motion strategies adopted by the two groups. In this context, learning those parameters can be considered a prerequisite for the following analysis. In total, we learned the joint weights for 36 healthy and PD affected hand coordination motions, respectively. Fig. 5 shows the mean and standard deviation for the weight of each joint and group. These first results suggest, that the motions of healthy and PD subjects primarily differ in the way the arm kinematic chain is actuated for task achievement. Healthy subjects show a strong tendency towards a *proximal* motion strategy, with decreasing joint weights along the arm chain, whereas PD subjects follow a *distributed* motion strategy, with balanced joint weights along the arm chain.

In order to determine whether there is a significant difference between the joint activity in healthy and PD subjects, we additionally performed an *independent sample t-test* for each weight. The results depicted in Fig. 6 confirm that, in particular, the weights for the proximal joints, shoulder pitch sh_pitch and upper arm roll up_arm_roll, are significantly higher for healthy subjects than for PD patients (with a significance level of $\alpha = 0.05$ and $\alpha = 0.025$, respectively).

Given these insights, it seems that healthy subjects naturally use primarily the proximal joints to minimize the distance to a given hand pose target. On the other hand, distal joint activity is set aside for delicate hand pose adjustments. PD seems to affect this motion strategy in terms of reduced proximal joints activity. In order to complete the same task successfully the reduced proximal motion is compensated by raising the motion contribution of the remaining joints.

Note that in our experiments, the activation of the spine joints were similar between the two groups, although PD is well known to affect the postural stability of subjects. This finding may result from patients recovering stability by making use of the arm and back rest provided by the chair they are sitting on while performing the task. In the future, we will investigate whether the omission of arm and back rest support yields significant differences between the weights learned for the spine joints for healthy and PD subjects.

VIII. Conclusions

In this paper we present a novel approach to differentiate between the underlying motion strategies adopted by healthy subjects and patients whose motor control is affected by a neurological disorder. We propose to learn indicative motor control parameters of a control scheme based on captured motion data. Our technique relies on a parameterization of the control scheme by means of joint weights, reflecting the activity level of joints contributing to the motion task. As we have shown in the experiments, the chosen control scheme is capable of closely replicating the recorded human end-effector and joint trajectories using the learned joint weights obtained from our algorithm.

According to our results on a motion database of healthy and Parkinson's disease subjects, there exist different motion strategies adopted by the two groups, referred to as the *proximal* and *distributed* motion strategy. Healthy subjects follow a hierarchical joint activation paradigm, whereas PD subjects show a balanced joint activation pattern. In general, the advantage of this novel measure of motor behavior lies in its independency of movement amplitudes and volition. Because joint contributions are not easily visible even for experienced neurologists, it might open a new field of motion analysis, yielding to new measures of motor deficits, which might be also used for evaluation of therapeutic interventions such as deep brain stimulation in Parkinsons disease, even in a closed-loop fashion.

References

[1] F. Rahimi, C. DuVal, M. Jog, C. Bee, A. South, M. Jog, R. Edwards, and P. Boissy, "Capturing whole-body mobility of patients with parkinson disease using inertial motion sensors: Expected challenges and rewards," in *Prof. of the Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, 2011.

[2] S. Das, L. Trutoiu, A. Murai, D. Alcindor, M. Oh, O. D. L. Torre, and J. Hodgins, "Quantitative measurement of motor symptoms in parkinsons disease: A study with full-body motion capture data," in *Proc. of the Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, 2011.

[3] B. Day, J. Dick, and C. Marsden, "Patients with parkinson's disease can employ a predictive motor strategy." *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 47, no. 12, pp. 1299–1306, 1984.

[4] J. Barbic, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard, "Segmenting motion capture data into distinct behaviors," in *Proc. of the Int. Conf. on Graphics Interface*, 2004.

[5] F. Zhou, F. De la Torre Frade, and J. K. Hodgins , "Aligned cluster analysis for temporal segmentation of human motion," in *Proc. of the IEEE Conf. on Automatic Face and Gestures Recognition*, 2008.

[6] ——, "Hierarchical aligned cluster analysis for temporal clustering of human motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 35, no. 3, pp. 582–596, 2013.

[7] G. Cimen, H. Ilhan, T. Capin, and H. Gurcay, "Classification of human motion based on affective state descriptors," *Computer Animation and Virtual Worlds*, vol. 24, no. 3-4, pp. 355–363, 2013.

[8] A. Aristidou and Y. Chrysanthou, "Feature extraction for human motion indexing of acted dance performances," in *Proc. of the Int. Conf. on Computer Graphics Theory and Applications*, 2014.

[9] F. Campos and J. Calado, "Approaches to human arm movement controla review," *Annual Reviews in Control*, vol. 33, no. 1, 2009.

[10] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *The Journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.

[11] E. Todorov and M. I. Jordan, "Smoothness maximization along a predefined path accurately predicts the speed profiles of complex arm movements," *Journal of Neurophysiology*, vol. 80, no. 2, pp. 696–714, 1998.

[12] S. Albrecht, K. Ramirez-Amaro, F. Ruiz-Ugalde, D. Weikersdorfer, M. Leibold, M. Ulbrich, and M. Beetz, "Imitating human reaching motions using physically inspired optimization principles," in *Proc. of IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2011.

[13] D. Roetenberg, H. Luinge, and P. Slycke, "Xsens MVN: Full 6DOF human motion tracking using miniature inertial sensors," xsens, Tech. Rep., 2009.

[14] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, 1st ed. Springer, 2008.

[15] S. Chiaverini, O. Egeland, and R. Kanestrom, "Achieving user-defined accuracy with damped least-squares inverse kinematics," in *Int. Conf. on Advanced Robotics*, 1991.

[16] T. Yoshikawa, "Manipulability of robotic mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[17] F. Chaumette and E. Marchand, "A redundancy-based iterative approach for avoiding joint limits: application to visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, 2001.

[18] D. Schinstock, T. Faddis, and R. Greenway, "Robust inverse kinematics using damped least squares with dynamic weighting," NASA, Tech. Rep., 1994.

[19] R. Smits, "KDL: Kinematics and Dynamics Library," http://www.orocos.org/kdl.