J Intell Robot Syst (2015) 80 (Suppl 1):S133–S148
DOI 10.1007/s10846-015-0218-y

Generalization of Force Control Policies from Demonstrations for Constrained Robotic Motion Tasks

A Regression-Based Approach

Vasiliki Koropouli · Sandra Hirche · Dongheui Lee

Received: 19 May 2014 / Accepted: 11 February 2015 / Published online: 22 February 2015

Abstract Although learning of control policies from demonstrations has been thoroughly investigated in the literature, generalization of policies to new contexts still remains a challenge given that existing approaches exhibit limited performance when generalizing to new tasks. In this article, we propose two policy generalization approaches employed for generalizing motion-based force control policies with the view of performing constrained motions in presence of motion-dependent external forces. The key concept of the proposed methods is using, apart from policy values, also policy derivatives or differences which express how the policy varies with respect to variations in its input and combine these two kinds of information to generalize the policy at new inputs. The first proposed approach learns policy and policy derivative values by linear regression and combines these data into a first-order Taylor-like polynomial to estimate the policy at new inputs. The second approach learns policy and policy difference data by locally weighted regression and combines them in a superposition fashion to estimate the policy at new inputs. The policy differences in this approach represent variations of the policy in the direction of minimizing the distance between the new incoming and average-demonstrated inputs. The proposed approaches are evaluated in real-world robot constrained motion tasks by using a linear-actuated, two degrees-of-freedom haptic device.

Keywords Learning by demonstration · Force control policies · Policy learning · Policy derivative · Policy generalization

V. Koropouli (✉) · D. Lee
Chair of Automatic Control Engineering,
Technische Universität München,
Karlstr. 45, 80333 Munich, Germany
e-mail: vicky@lsr.ei.tum.de

D. Lee
e-mail: dhlee@tum.de

S. Hirche
Chair of Information-Oriented Control,
Technische Universität München,
Barer str. 21, 80333 Munich, Germany
e-mail: hirche@tum.de

1 Introduction

Robots need to exhibit skillful force regulation skills while manipulating objects of the environment in order to efficiently achieve the desired goal of a task. Given that humans exhibit exceptional skills in manipulating their environment by regulating arm force and impedance [1–3], learning from human demonstrations is a promising route to transferring advanced force tuning skills to robots. The prominent challenge in learning from demonstration lies in the ability to generalize learned skills to similar tasks in the future. Let us here illustrate a generalization paradigm which is treated in the scenario of this article. Let us consider a robot end-effector which has learned how to perform certain movements inside a deformable and homogeneous environment while experiencing certain state-dependent forces from the environment. Given that the environment is homogeneous, the external forces only depend on the task's motion states. To illustrate this, consider that following a motion path in short depth from an object's surface is a different task than following the same path deeper inside the object where the manipulating mass increases significantly and imposes different constraints on the end-effector, see Fig. 1a. In case that a new movement, different than those demonstrated, has to be realized in the same environment, new visited states give rise to new state-dependent counteraction forces and adjustment of applied force is required in order for the end-effector to follow the new path. The problem of computing the force which is required such that a desired motion is realized is widely known as inverse dynamics [4]. If the inverse dynamics model of a plant can be acquired or learned, this model can serve as a feedforward control policy for the plant [3], see Fig. 1b. In case that the dynamics of a task cannot be exactly modelled and serve as an ideal feedforward controller, a wise alternative is to learn these dynamics from demonstrated task data. In [4], robot's inverse dynamics are learned by Locally Weighted Projection Regression (LWPR), support vector regression and Gaussian process regression and the learning performance of these methods is compared. LWPR is also employed in [5] for inverse dynamics' learning where a priori knowledge about robot's rigid body dynamics is incorporated in learning with the view of efficient generalization. Use of robot's rigid body dynamics in learning inverse dynamics is also performed in [6] where a Gaussian-process semiparametric regression approach is employed. In our present work, we focus on generalization strategies of robot's feedforward force control policies from motion-task demonstrations, with the view of successfully generalizing to new motions which impose different motion-dependent disturbances.

Learning of force skills for robotic manipulation tasks has recently received large attention. Learning of force and torque data is performed in [7] by Gaussian Mixture Regression (GMR) for a container-emptying task and in [8] by Hidden Markov Models (HMM) for a ball-in-ball and a pouring task. In [9], positional and force skills are separately demonstrated and learned in the form of mixtures of dynamical systems. However, in dynamic interaction tasks, position and force cannot be viewed independently and, instead, the dynamics of the task has to be learned. In [10], the end-effector is represented by a spring-damper system whose position, velocity, acceleration and applied force on the environment are demonstrated and used as input data to learn the reference position of the spring-damper system by Gaussian Mixture Modeling (GMM) for cooperative transportation tasks. Gams et al. [11]

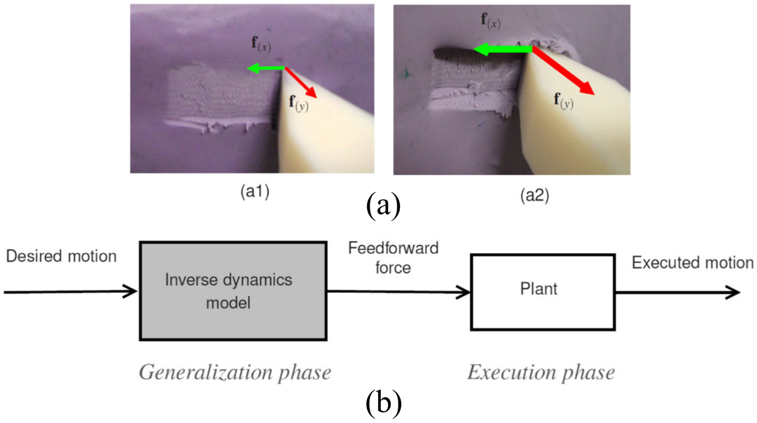


Fig. 1 **a** Illustrating an engraving task at different depths inside a sufficiently homogeneous plasticine object. Different environmental disturbance $\{f_{(x)}, f_{(y)}\}$ is experienced in each case due to the changing manipulating mass. Engraving in a (a1) low depth, (a2) high depth. **b** An inverse dynamics model can be viewed as a feedforward control policy which outputs a force estimate for a desired motion to be executed. If the executed motion is identical to the desired motion, the inverse dynamics model is considered ideal or, alternatively, the policy generalization problem has an ideal solution

proposes the modulation of dynamic movement primitives [12] by coupling terms enclosing sensory feedback, in order to assign to the robot a desired dynamic behavior for manipulation tasks where the coupling terms are learned from demonstrated data by iterative learning control. Furthermore, in [13], interaction force patterns, represented by dynamical systems, are learned by regression from single demonstrations while their ability to generalize is limited to changing the final goal of the force pattern. Given that, in real-world scenarios, both motion and force information matters, a manipulation framework is presented in [14] where motion and force primitives are combined with force control and optimization for grasping. For the purpose of learning and generalizing grasping skills, demonstrated motion and force data are employed in [15] to estimate the desired positions and interaction forces of grasping fingertips by using GMM and HMM.

Apart from force, impedance-based behaviors are also investigated. PI^2 reinforcement learning is employed in [16] to learn variable impedance control and in [17] to learn desired end-effector impedance to execute tasks in presence of stochastic force fields. In addition, in [18, 19], motion primitives are learned and kinesthetically modulated by controlling the robot joints' stiffness for physical interaction tasks while in [23] motion learning is combined with optimal feedback control for haptic assistance. Motion and interaction primitives are also learned and combined with impedance control for human-humanoid physical contact tasks in [22]. In [21], impedance behaviors are encoded in terms of task force and visual information. In [20], a neuroscience-based controller which adjusts impedance, feedforward force and position to perform various contact tooling tasks such as cutting, drilling and surface exploration is proposed and evaluated in simulations.

In this work, we learn generalization of force control policies for constrained motion tasks inside homogeneous and deformable environments. Although learning of control policies from data has been widely treated in the literature, policy generalization still remains a challenge and necessitates further methodical investigation. A review on learning control policies is presented in [24]. Learning of force control policies has been treated in [25, 26] by using Reinforcement Learning (RL). As an advancement to the state-of-the-art RL methods, a highly efficient probabilistic inference algorithm is proposed in [27] for fast policy search from scratch. However, RL and other policy search algorithms require multiple execution trials for success and are not suitable for manipulation of deformable objects where successful task generalization is desired within a single execution to avoid non-desired object deformation caused by many trials. From the viewpoint of regression, techniques such as Linear Regression (LR) [28] and Locally Weighted Regression (LWR) [29] as well as incremental techniques such as Receptive Field Weighted Regression (RFWR) [29] and Locally Weighted Projection Regression (LWPR) [30] can be employed for learning and generalization of control policies. RFWR and LWPR are advanced techniques which allow for policy generalization by incrementally modifying their learning structure based on new incoming data.

Despite the powerful capabilities of the previous approaches in policy learning, the problem of policy generalization from data still remains a challenge. Although existing approaches achieve to efficiently generalize to regions very close to the demonstrated data, this generalization ability degrades as the distance from the demonstrated data increases. In this article, we wish to learn force generalization skills with the view of performing motion tasks under varying motion-dependent disturbances. At this point, let us define a policy as the mapping from a set of inputs to a set of outputs and a policy derivative as the mapping from a set of differences between inputs to a set of differences between outputs. In addition, let us define a policy difference as a variation of a policy's output in response to a variation of its input. The keypoint to our approach is learning, apart from policy values, also policy derivatives or policy differences and combining these two kinds of information for approximating a policy in new regions of the input space. Use of policy derivatives has been previously proposed for identification of unknown systems by Gaussian processes [31, 32]. In [31], policy and policy derivative values are employed in modelling of nonlinear dynamic systems using Gaussian processes and in [32], Gaussian process models are built for predictive control based on derivative observations.

When using derivative/difference information for policy identification, two general issues arise. The first issue consists of how to extract the policy derivative/difference information from given data, given that policy derivatives and differences cannot be measured. The second issue consists of how to exploit this derivative/difference information for policy approximation. In this article, we propose two approaches for generalizing force control policies. The first approach combines policy and policy derivative information learned by Linear Regression (LR) for generalization. Preliminary results of this approach are presented in [33]. The second approach combines policy and policy difference information learned by Locally Weighted Regression (LWR) for policy generalization. We evaluate the proposed approaches in real-world constrained robot motion tasks and compare their performance with the performance of LWR and LWPR.

This article is structured as follows. First, in Section 2, we define our problem. In Section 3, we present LR and LWR which are employed for learning and, in Section 4, we present two methods for policy generalization based on LR and LWR. In Section 5, we evaluate the approaches in experiments and, finally, in Section 6, we make a discussion.

2 Problem Formulation

Our goal consists of developing a method for generalizing force control policies given a set of task demonstrations, with the view of executing constrained movements inside deformable and homogeneous environments where only state-dependent external forces exist. We consider policies whose output is force and input is motion data. As we explain in the introduction, this is, in essence, an inverse dynamics problem and consists of estimating the force which is responsible for a certain motion to be realized, see Fig. 1b. A constrained movement can, in general, be realized by applying different control policies which may consist, for example, of high-, fixed-gain position control, adaptive control or a human-inspired force control policy. Different policies, though, generate different forces to accomplish the same movement by imposing, in this way, different stress on the end-effector and the environment. Control-engineering schemes such as high-gain position and adaptive control may lead to the generation of high forces or overshoots which may be harmful to the environment or the end-effector or cause a non-desired effect in terms of the task goal. For this, in this article, we propose to learn force control policies from expert demonstrations, which express how humans control applied force during tasks. By doing this, a robot can be endowed with high-standard motor control skills which are important in delicate manipulation tasks whereby the environment needs to be cautiously treated and high forces are not desired or are even prohibited.

During motion inside a deformable environment, motion dynamics between different directions are physically coupled and this coupling imposes interconnection between force control policies of different directions. Based on this, we define a task-space force control policy in the i-th direction as

$$f_{d_{(i)}} = \pi\left(\mathbf{s}_{d_{(i)}}\right) \tag{1}$$

where $f_{d_{(i)}}$ is a demonstrated force and $\mathbf{s}_{d_{(i)}}$ a vector of demonstrated motion variables which is defined as $\mathbf{s}_{d_{(i)}} = \left[x_{d_{(i)}}\ \dot{x}_{d_{(i)}}\ \ddot{x}_{d_{(i)}}\ \mathbf{c}_{(i)}\right]$. The $x_{d_{(i)}}$, $\dot{x}_{d_{(i)}}$, $\ddot{x}_{d_{(i)}}$ represent position, velocity and acceleration respectively in the i-th direction and $\mathbf{c}_{(i)}$ is a vector-valued function which represents the coupling between the i-th and the other $j \neq i$ directions. The coupling function is $\mathbf{c}_{(i)} = \mathbf{c}_{(i)}\left(x_{d_{(j)}}, \dot{x}_{d_{(j)}}\right)$, $\forall j \neq i$ and establishes a dependence of the force $f_{d_{(i)}}$ on the position and velocity states of the remaining directions $j \neq i$.
Notation: In the remainder of this article, we denote the time index by lower case numbers $(\cdot)_i$, index of motion direction by lower case numbers inside parentheses $(\cdot)_{(i)}$ and demonstration index by upper case numbers inside parentheses $(\cdot)^{(i)}$.

In the remainder of this section, for reasons of simplicity and without loss of generality, we restrict our analysis to a single direction of movement and we omit the directional index $(\cdot)_{(i)}$. Based on this, the problem we wish to solve can be defined as follows.

Problem: Given data $\{\mathbf{s}_d,\ f_d\}$ where $\{\mathbf{s}_d\} = \left\{\mathbf{s}_d{}^{(k)}\right\}$, $\{\mathbf{f}_d\} = \left\{\mathbf{f}_d{}^{(k)}\right\}$, $k = 1, \ldots, K$ is the demonstration index, $K \in \mathbb{N}$, $K \geq 2$ where K is the number of demonstrations,

- learn the control policy $\pi : \{\mathbf{s}_d\} \xrightarrow{\pi} \{f_d\}$,
- given $\mathbf{s}' = [x'\ \dot{x}'\ddot{x}'\ \mathbf{c}'] \notin \{\mathbf{s}_d\}$, estimate the value of the policy $\pi(\mathbf{s}')$ at the new input $\mathbf{s}'$.

Let us consider K demonstrations of a task with N datapoints per demonstration. To learn the force control policy in either direction, data pairs from all demonstrations are concatenated as $(\{\mathbf{s}_{d_1}, f_{d_1}\}, \ldots, \{\mathbf{s}_{d_{K\times N}}, f_{d_{K\times N}}\})$ where the motion vector of demonstration k at time i is $\mathbf{s}_{d_i}^{(k)} = \mathbf{s}_{d_{i+N(k-1)}}$ and the corresponding force element is $f_{d_i}^{(k)} = f_{d_{i+N(k-1)}}$.

Figure 2 illustrates our system during demonstration of a task in a single direction of movement. The system consists of the human, end-effector, manipulation tool and environment. The end-effector behaves as an admittance and is position-controlled. The demonstrated signals which are measured are the end-effector force f_d, end-effector position x_d and velocity $\dot{x}_d$. T_x and T_f are some unknown transformation matrices of position and force respectively. The force which is measured by the sensor at the end-effector, while the tool interacts with the environment, is $f_d = f_h + f_c - f_e$ where f_h is the human force input, f_e some force sensed from the environment and f_c a force due to the presence of the position controller. The tool tip position x_s is not measured. In addition, the forces f_e, f_h and f_c are not measurable.

3 Background Theory

In this section, we analyze two non-incremental regression techniques, LR [28] and LWR [29], which are employed by the proposed generalization approaches.

3.1 Linear Regression

In Linear Regression, a control policy is represented by $\pi = \mathbf{w}^T \phi(\mathbf{s}_d)$ where $\mathbf{w} \in \mathbb{R}^{D+1}$ is a parameter vector and $\phi(\mathbf{s}_d) = [\mathbf{s}_d \ 1]^T$ is a basis-function model where $\mathbf{s}_d \in \mathbb{R}^{1\times D}$ is a state vector. The policy π is learned by minimizing the cost [34]

$$R = \sum_{i=1}^{K\times N} \| f_{d_i} - \pi(\mathbf{s}_{d_i}) \|^2 \tag{2}$$

which becomes

$$R = \sum_{i=1}^{K\times N} \left(f_{d_i} - \mathbf{w}^T \phi(\mathbf{s}_{d_i}) \right)^2 . \tag{3}$$

By minimizing (3) with respect to $\mathbf{w}$, we receive

$$\sum_{i=1}^{K\times N} f_{d_i} \phi^T(\mathbf{s}_{d_i}) = \mathbf{w}^T \sum_{i=1}^{K\times N} \phi(\mathbf{s}_{d_i}) \phi^T(\mathbf{s}_{d_i})$$

and, thus, the estimated parameter vector $\mathbf{w}$ is given by

$$\mathbf{w} = \mathbf{w}_1^T H^{-1} \tag{4}$$

where $\mathbf{w}_1 = \sum_{i=1}^{K\times N} f_{d_i} \phi(\mathbf{s}_{d_i})$ and $H = \sum_{i=1}^{K\times N} \phi(\mathbf{s}_{d_i}) \phi^T(\mathbf{s}_{d_i})$.

The least-square risk function (2) can be modified by assigning different weights w_i^* to different observations as

$$R^* = \sum_{i=1}^{K\times N} w_i^* \| f_{d_i} - \pi(\mathbf{s}_{d_i}) \|^2 . \tag{5}$$

The w_i^* determines how much the i-th observation influences the final parameter estimates. The parameters estimated by minimizing (5) are called Weighted Least-Square (WLS) estimates. A common application of WLS is in the case where the observations f_{d_i} have different variances σ_i where the weights should be optimally set as $w_i^* = \frac{1}{\sigma_i^2}$ so that the smallest standard error of the estimation is achieved. In general, the weights w_i^* can be determined upon the special characteristics of the estimation setting.

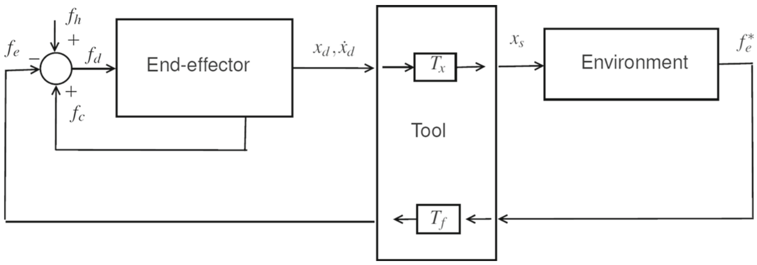


Fig. 2 The robot end-effector interacting with the environment, in a single direction of movement, during the task demonstration phase

3.2 Locally Weighted Regression

In LWR, the policy is represented as the normalized weighted sum of a set of linear models. The linear models represent receptive fields with centers $\mathbf{c}_m$, $m = 1, \dots, M$ where M is the number of fields [29]. The policy is defined as

$$\pi(\mathbf{s}) = \frac{\sum_{m=1}^{M} w_m \pi_m(\mathbf{s})}{\sum_{m=1}^{M} w_m}, \qquad \pi_m(\mathbf{s}) = \tilde{\mathbf{s}}^T \mathbf{b}_m \tag{6}$$

where $\tilde{\mathbf{s}} = [(\mathbf{s}-\mathbf{c}_m)\ 1]^T$ and $\mathbf{c}_m \in \mathbb{R}^{1\times D}$. The weights w_m are defined by Gaussian functions as

$$w_m = exp\left(-\frac{1}{2\sigma^2}||\mathbf{s}-\mathbf{c}_m||^2\right).$$

The regression parameters $\mathbf{b}_m$ are estimated by

$$\mathbf{b}_m = \frac{\sum_{i=1}^{KN} w_{im} f_{d_i} \tilde{\mathbf{s}}_{d_i}}{\sum_{i=1}^{KN} w_{im} \tilde{\mathbf{s}}_{d_i}^T \tilde{\mathbf{s}}_{d_i}}$$

where

$$w_{im} = exp\left(-\frac{1}{2\sigma^2}||\mathbf{s}_{d_i}-\mathbf{c}_m||^2\right),\ i = 1, \dots, KN.$$

4 Policy Generalization Techniques Based on LR and LWR

In the previous section, we present two non-incremental regression techniques, namely LR and LWR, for learning control policies. Apart from non-incremental techniques, incremental techniques are also developed [29, 30] aiming at generalization. Despite this progress, generalization problems and primarily extrapolation still remain a challenge and the question of whether further approaches can be developed to exploit and do the best with the data available for learning toward the goal of efficient generalization in specific contexts, still awaits an answer. Here, we propose two policy generalization techniques based on LR and LWR.

4.1 Policy generalization by Weight Differential Learning (GWDL) based on LR

This approach is inspired by differential calculus and approximates a function at a point by a first-order polynomial expansion which resembles the Taylor polynomial. However, in contrast to Taylor polynomials whose weighting coefficients are represented by the derivative of the function at a known point, in GWDL the weight of the first-order term of the expansion expresses the mapping from a set of differences between demonstrated inputs to a set of differences between demonstrated outputs as it is analyzed later, and this mapping is learned by LR from demonstrated data. This coefficient is called weight differential, is symbolized by $\Delta\mathbf{w}$ and expresses the rate of change of the policy with respect to its input. Let us define the rate of change of the policy π as

$$\Delta\mathbf{s}_d \xrightarrow{\frac{\Delta\pi}{\Delta\mathbf{s}_d}} \Delta f_d, \qquad \frac{\Delta\pi}{\Delta\mathbf{s}_d} \triangleq \Delta\mathbf{w} \tag{7}$$

where Δ symbolizes a finite difference, f_d is the demonstrated force and $\frac{\Delta\pi}{\Delta\mathbf{s}_d}$ denotes the derivative of the policy π with respect to the state $\mathbf{s}_d$. We observe that the derivative of the policy is a new policy with input data $\Delta\mathbf{s}_d$ and output data Δf_d. LR is employed to learn this new policy and the weight vector which is learned by LR is considered the differential of the weighting vector $\mathbf{w}$, see Eq. 4. More specifically, to learn the $\Delta\mathbf{w}$, a new observation dataset D is generated, which consists of the differences between datapoints of every two demonstrations. Let us assume the dataset

$$D^{k_1,k_2} = \left\{ \left(\mathbf{s}_d^{(k_1)} - \mathbf{s}_d^{(k_2)}\right),\ \left(f_d^{(k_1)} - f_d^{(k_2)}\right) \right\}$$

which consists of the input and output differences of all datapoints between every two demonstrations k_1 and k_2. Finally, all datasets D^{k_1,k_2} are concatenated into a single dataset D as

$$D = \left\{D^{k_1,k_2},\ k_1,\ k_2 = 1, \dots, K\right\}. \tag{8}$$

By applying LR on the dataset D, the parameter vector $\Delta\mathbf{w}$ is estimated. Following learning of the $\Delta\mathbf{w}$, the force control policy is approximated as follows:

(i) The observed motion vectors from all K demonstrations are concatenated as $\left\{\mathbf{s}_d^{(1)}, \dots, \mathbf{s}_d^{(K)}\right\}$ and the average over demonstrations motion $\mathbf{s}^{(av)}$ is computed as

$$\mathbf{s}^{(av)} = \sum_{k=1}^{K} \mathbf{s}_d^{(k)}/K.$$

and has time length equal to N.

(ii) Given a new motion input $\mathbf{s}'_j$, the point of the average motion pattern $\mathbf{s}^{(av)}$ which lies closest to the query point $\mathbf{s}'_j$ is computed as:

$$\mathbf{s}_{min} = \arg \min_{\mathbf{s}_i^{(av)}} \|\mathbf{s}'_j - \mathbf{s}_i^{(av)}\|, \; i = 1, \ldots, N \quad (9)$$

where $\| \cdot \|$ denotes the Euclidean distance.

(iii) The policy at the new input is approximated by the first-order expansion

$$f'_j = \pi\left(\mathbf{s}'_j\right) = \mathbf{w}^{\mathrm{T}}\tilde{\mathbf{s}}_{min} + \Delta\mathbf{w}^{\mathrm{T}}\left(\tilde{\mathbf{s}}'_j - \tilde{\mathbf{s}}_{min}\right) \quad (10)$$

where $\tilde{\mathbf{s}}_{min} = [\mathbf{s}_{min}\ 1]^{\mathrm{T}}$, $\tilde{\mathbf{s}}'_j = [\mathbf{s}'_j\ 1]^{\mathrm{T}}$ and $\mathbf{w}$ is learned by LR on the dataset $\{\mathbf{s}_d, f_d\}$. In Eq. 10, we observe that the average input data act as known points and the policy varies with respect to the average demonstrated behavior in order to generalize to new inputs.

The proposed algorithm has a strong intuitive meaning in that, to predict future actions, we need to know the difference of the new task goal from previous goals and how this difference is mapped onto the action space represented, here, by the force. Instead of estimating the $\Delta\mathbf{w}$ once from the whole set D of demonstrated data, one could alternatively estimate the $\Delta\mathbf{w}$ locally in a region around the $\mathbf{s}_{min}$ for every new $\mathbf{s}_{min}$. However, by trying this, we notice that the estimated force f' becomes noisy due to the different value of $\Delta\mathbf{w}$ for each new $\mathbf{s}'_j$. For this, the $\Delta\mathbf{w}$ is estimated only once globally from the whole set of demonstrated data.

An important point of the present algorithm is that it takes into account the average over demonstrations motion trajectory and compares each new input with this average trajectory in order to find the $\mathbf{s}_{min}$. There are several reasons why the average motion trajectory can serve here as good reference for generalization. In our scenario, demonstrated motions lie fairly close to each other and, thus, their average is considered to be representative of the visited motion domain and enclose the important constraints of the task. However, if demonstrated motions lie far from each other and span a large region of the input space, simple averaging would fail to enclose all the important features of the task because the motion average would derive from largely different data. Furthermore, in our scenario, we do not aim at reaching some goal positions but rather following a certain motion pattern and, given this, the average can serve as a representative of the motion route of the task. Even in cases where goal positions have to be reached, by making all the demonstrated motions pass from these goal points, the corresponding average would also preserve this goal information. An alternative to comparing with the average trajectory would be to compare with all the demonstrated input points. By doing this, we do not notice any considerable improvement in the generalization ability of the algorithm but mostly an increased computational cost of the approach. The reason for this is because demonstrated motions in our setting lie close to each other and comparing with each demonstrated motion point instead of the average did not offer further noticeable information about the task. In addition, comparison with each demonstrated input point increases the search space and requires memorization of the whole set of demonstrated inputs. Another alternative to motion averaging is to encode the motion data in time space by a probabilistic model such as GMR and use the motion estimate of the model as reference which to compare the new incoming motion inputs with. Such a probabilistic approach estimates the relevance of the input motions [35] and may prove more successful in extracting the main features of a task in case that demonstrated motions span a large domain. However, we should notice that even probabilistic approaches learn from the whole set of demonstrated data and their estimates represent some kind of system's average behavior [35].

4.2 Policy Generalization Through Estimation of Policy Differences by LWR (DLWR)

A plausible question which emerges is whether the use of nonlinear models for learning the relationship between inputs and outputs and differences of inputs and differences of outputs, instead of a global linear model as in LR, can allow for better policy approximation accuracy. Based on this notion, we develop here an alternative approach based on LWR for policy approximation. The derivation of this approach is as follows. It is evident that the force policy at a new input $\mathbf{s}'_j$ can be written as $f'_j(\mathbf{s}'_j) = f'_j(\mathbf{s}_{min} + (\mathbf{s}'_j - \mathbf{s}_{min}))$. By representing this policy by LWR, we write

$$f'_j\left(\mathbf{s}_{min} + \left(\mathbf{s}'_j - \mathbf{s}_{min}\right)\right) = \frac{\sum_{m=1}^{M} w_m \pi_m\left(\mathbf{s}_{min} + \left(\mathbf{s}'_j - \mathbf{s}_{min}\right)\right)}{\sum_{m=1}^{M} w_m}. \quad (11)$$

Based on Eq. 6, it is

$$\pi_m\left(\mathbf{s}_{min}+\left(\mathbf{s}'_j-\mathbf{s}_{min}\right)\right) = \left(\tilde{\mathbf{s}}_{min}\right)^T \mathbf{b}_m + \left(\tilde{\mathbf{s}}'_j-\tilde{\mathbf{s}}_{min}\right)^T \mathbf{b}_m = \pi_m(\mathbf{s}_{min}) + \pi_m\left(\mathbf{s}'_j-\mathbf{s}_{min}\right),$$

and, thus, Eq. 11 becomes

$$f'_j\left(\mathbf{s}_{min}+\left(\mathbf{s}'_j-\mathbf{s}_{min}\right)\right)=\pi\left(\mathbf{s}'_j\right)=\underbrace{\frac{\sum_{m=1}^{M} w_m\pi_m(\mathbf{s}_{min})}{\sum_{m=1}^{M} w_m}}_{\pi_j} + \underbrace{\frac{\sum_{m=1}^{M} w_m\pi_m\left(\mathbf{s}'_j-\mathbf{s}_{min}\right)}{\sum_{m=1}^{M} w_m}}_{\Delta\pi_j} \quad (12)$$

which is also written as

$$f'_j\left(\mathbf{s}'_j\right) = \pi_j\left(\mathbf{s}_{min}\right) + \Delta\pi_j\left(\mathbf{s}'_j-\mathbf{s}_{min}\right). \quad (13)$$

In LWR, the policy $\pi(\mathbf{s}'_j)$ is once learned from the input-output dataset $\{\mathbf{s}_d,\ f_d\}$. However, in our current approach, the policy $\pi(\mathbf{s}'_j)$ is decomposed into two different policies π_j and $\Delta\pi_j$ and it is proposed that these policies are separately learned from different datasets. More specifically, the π_j is learned by LWR from the dataset $\{\mathbf{s}_d, f_d\}$ as described in Section 3.2 while $\Delta\pi_j$ is learned from the dataset D defined in Eq. 8. In addition, the w_m of π_j and $\Delta\pi_j$ are separately defined based on the different training datasets. The concept of learning separately the π_j and $\Delta\pi_j$ is straightforward in that the $\Delta\pi_j$ receives as arguments differences of inputs and has to be learned from differences of inputs as well. The $\Delta\pi_j$ determines the policy increment with respect to the known policy value π_j in the direction of minimizing the distance between the new input $\mathbf{s}'_j$ and $\mathbf{s}_{min}$, see Eq. 9.

The performance of the two estimation laws, namely (10) of GWDL and (13) of DLWR, cannot be analytically compared because the two generalization laws are based on different concepts and there is not an explicit analytic correspondence between them. Although DLWR is based on LWR which can represent higher-complexity input-to-output mappings compared to LR, the DLWR generalization law is not any more a pure LWR representation but combines, in a superposition fashion, representations which are learned by LWR. In the same way, the GWDL generalization law combines information learned by LR in a 2-term Taylor polynomial and is not any more a linear as LR but a nonlinear law. On one hand, the algebra of LR provides a strong intuition for learning the policy and policy derivative values from a known dataset since the input-to-output mapping is represented by a single vector which plays the role of the rate of change of the policy, see Eq. 7. In turn, the fact that the policy derivative $\Delta\mathbf{w}$ is available motivates for applying Taylor expansion for policy approximation at new inputs. On the other hand, the DLWR generalization law resembles the concept of gradient-based policy update, as prescribed by Eq. 13, where for each new input, the policy is updated in the direction of minimizing the distance between the new input $\mathbf{s}'_j$ and the closest point $\mathbf{s}_{min}$ of the average demonstrated trajectory. The difference between gradient-based policy search and estimation law (13) is that both terms in Eq. 13 are updated based on the new input and no information from previous time points is employed. Comparison between GWDL and DLWR is performed in an experimental level as presented in Section 5.

5 Experimental Evaluation

In this section, we evaluate GWDL and DLWR and compare them with LWR and LWPR in real-world robot constrained motion tasks.

5.1 Setup

Force control policies are learned from demonstrations with the goal of executing constrained motions by using a 2-degrees-of-freedom linear-actuated haptic device (*ThrustTube*), see Fig. 3. The motions

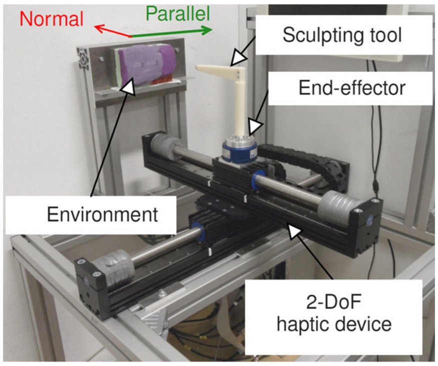


Fig. 3 Experimental setup. The two directions of movement, normal and parallel, are visualized by a red-and a green-color axis respectively

are executed on a plasticine object which is sufficiently homogeneous in practice. The end-effector of the device moves in two directions, one normal and one parallel to the object's initially planar surface. A sculpting tool is firmly attached on the end-effector for engraving the plasticine object and the system end-effector-tool behaves as a rigid body which only realizes translational motion. Prior to demonstration, the end-effector is placed such that the tool tip just touches the object's surface. During demonstration, the user moves the handle of the end-effector in the two directions while the tool moves inside the plasticine material. Three demonstrations are executed by first moving the end-effector in the direction normal to the object's surface up to a certain depth and then moving it in the direction parallel to the object's surface up to a certain length. The demonstrated motion depths and lengths slightly differ across demonstrations. During demonstration, a force/torque sensor ($JR3$) measures the end-effector force f_d while the device's encoders measure end-effector position x_d and velocity $\dot{x}_d$ in both directions, see Fig. 2. During task demonstration and generalization, the haptic device behaves as an admittance and is position-controlled. The parameters of the admittance are stiffness 10 N/m, damping 30 Ns/m and mass 40 Kg and the sampling rate is equal to 1 KHz.

The experimental scenario of learning force control skills for constrained motions inside deformable materials is exciting and, at the same time, demanding because when a complex-shaped end-effector dynamically interacts with the environment: (i) physical coupling of task dynamics between different directions exists which makes generalization challenging, (ii) the task dynamics are highly nonlinear due to the complex physics at the place of interaction, and (iii) the applied force rigorously depends on the motion states and, thus, the motion-to-force mapping policy is highly sensitive to motion variations. This policy's sensitivity helps to reveal the generalization precision of the proposed approaches. Thus, the considered experimental scenario is able to reveal the efficiency and accuracy of the proposed algorithms in learning complex nonlinear mappings which are sensitive to input variations.

As explained in Section 2, the learned force is the end-effector force f_d which is measured by a force sensor during demonstration. The noise of the force sensor is negligible, which ensures that measured force is the same for exactly the same task; same motion, environment, end-effector, demonstrator and demonstrator's motor policy. The demonstrator motor policy refers to the way the demonstrator executes the task, who may, for instance, apply high forces and move aggressively or move in a compliant manner by applying low forces instead. Although the demonstrator motor policy may present slight variations according to the difficulty of the task (harder environment may motivate for more aggressive motor policy), these variations can also be learned from data. Here, we consider that the demonstrator employs the same motor policy across demonstrations. In addition, all demonstrations are executed by the same end-effector, demonstrator and in the same homogeneous environment. Based on this, the only factor which can result in inconsistency of force demonstrations is the intrinsic motor output variability of the human motor control system [2]. However, in our scenario, the force variations due to human motor output variability are negligible compared to task-related forces and are neglected. Given that during motion inside a deformable and homogeneous environment, applied forces meaningfully depend on the motion states, even similar motion states may lead to different measured forces. In order to properly infer the force control policy of a task and successfully generalize to new movements, we demonstrate motions which lie rather close to each other (motions of similar normal depths and tangential lengths) and explore small regions of the input space. From the proximity of the demonstrated motions, we ensure that the average motion trajectory employed for generalization by GWDL and DLWR is characteristic of the visited motion domain and encodes the important features of the task, see Section 4.1.

5.2 Representation of Force Control Policies and Task Performance Criterion

5.2.1 Force Control Policies

Following learning of a force control policy from a set of demonstrated movements, our goal consists of generalizing this policy to new movements which are executed in the same environment with that of demonstrations but impose new motion-varying forces on the end-effector. First, an engraving task is demonstrated three times as it is described in Section 5.1. The force control policy in the two directions of movement is represented by

$$\begin{aligned} \pi &= \left[\pi_{(n)}\left(\mathbf{s}_{d_{(n)}}\right)\ \pi_{(p)}\left(\mathbf{s}_{d_{(p)}}\right)\right]^T, \\ \mathbf{s}_{d_{(n)}} &= \left[x_{d_{(n)}}\ \dot{x}_{d_{(n)}}\ \ddot{x}_{d_{(n)}}\ x_{d_{(p)}}\ \dot{x}_{d_{(p)}}\right], \\ \mathbf{s}_{d_{(p)}} &= \left[x_{d_{(p)}}\ \dot{x}_{d_{(p)}}\ \ddot{x}_{d_{(p)}}\ x_{d_{(n)}}\ \dot{x}_{d_{(n)}}\right] \end{aligned} \tag{14}$$

where the indices 'n' and 'p' stand for the 'normal' and 'parallel' direction respectively, $\pi_{(n)}(\mathbf{s}_{d_{(n)}}) = f_{d_{(n)}}$ and $\pi_{(p)}(\mathbf{s}_{d_{(p)}}) = f_{d_{(p)}}$. Note that the normal and parallel control policies are interconnected to each other. Multidimensional Dynamic Time Warping is applied to align the force and motion data of different demonstrations before learning [36].

The proposed generalization approaches do not aim at dealing with irrelevant input data. In our scenario, relevancy of the input to the output is assumed such that a task-realistic policy is estimated, which can efficiently generalize to new inputs in the future. Given that physical coupling between motion dynamics of different directions is obvious to exist in our scenario, we set as inputs of the control policy in each direction the set which contains, apart from the motion states of this direction, the coupling position and velocity states of the other directions as well, see Eq. 14. When the task-relevant inputs cannot be inferred, Mutual Information Analysis (MIA) can be employed to estimate the dependence of the output on the input data and extract the relevant inputs. Application of MIA for a task learning scenario is presented in [7]. Here, we analyze the relevancy of our inputs by computing the Pearson's correlation coefficient which is one of the simplest metrics in MIA and describes the linear dependence between two datasets. Let us assume two variables X_1 and X_2. The Pearson's correlation coefficient is given by

$$P = \frac{E[X_1X_2] - E[X_1]E[X_2]}{\sigma_{X_1}\sigma_{X_2}} \tag{15}$$

where $E(\cdot)$ denotes the expected value and σ the standard deviation. We compute the Pearson coefficient between: (i) parallel position $X_1 = x_{d_{(p)}}$ and normal force $X_2 = f_{d_{(n)}}$; $P = 0.4113$, (ii) parallel velocity $X_1 = \dot{x}_{d_{(p)}}$ and normal force $X_2 = f_{d_{(n)}}$; $P = -0.1322$, (iii) normal position $X_1 = x_{d_{(n)}}$ and parallel force $X_2 = f_{d_{(p)}}$; $P = -0.3931$, and (iv) normal velocity $X_1 = \dot{x}_{d_{(n)}}$ and parallel force $X_2 = f_{d_{(p)}}$; $P = -0.6310$. We observe that a non-negligible dependence does exist between the normal force and parallel motion states and the parallel force and normal motion states and, thus, the corresponding inputs are considered relevant to the task's control policy.

5.2.2 *Task Performance Criterion*

We distinguish two main cases of generalization where: i) the new motions lie inside the range of experienced motions and we call it *policy interpolation* and ii) the new motions lie outside the range of experienced motions and we call it *policy extrapolation*. In the generalization phase, the generalized force f' is applied and the corresponding realized position x_m is measured, see Fig. 1b. The generalization performance criterion is represented by the tracking error between the desired and measured position trajectory as

$$E_{(n)}=\sum_{i=1}^{L}\left(x_{(n),i}-x_{m_{(n),i}}\right)^2, \quad E_{(p)}=\sum\nolimits_{i=1}^{L}\left(x_{(p),i}-x_{m_{(p),i}}\right)^2 \tag{16}$$

where $x_{(n)}$ and $x_{(p)}$ represent the desired positions and $x_{m_{(n)}}$ and $x_{m_{(p)}}$ the measured positions after execution by any generalization method while i is the timepoint index and L the number of datapoints of the desired trajectory.

5.3 Experiments

We demonstrate the generalization performance of GWDL and DLWR and compare it with performance of LWR and LWPR. First, we show policy generalization in certain interpolation and extrapolation cases and then we further evaluate GWDL within a larger range of motions.

5.3.1 *Test Case 1: Interpolation to New Normal Motion*

We test policy interpolation to a new movement which lies inside the range of the demonstrated movements. The desired trajectories are

$$x_{(p)} = x_{(p)}^{(3)}, \quad x_{(n)} = 0.8 \times x_{(n)}^{(3)}$$

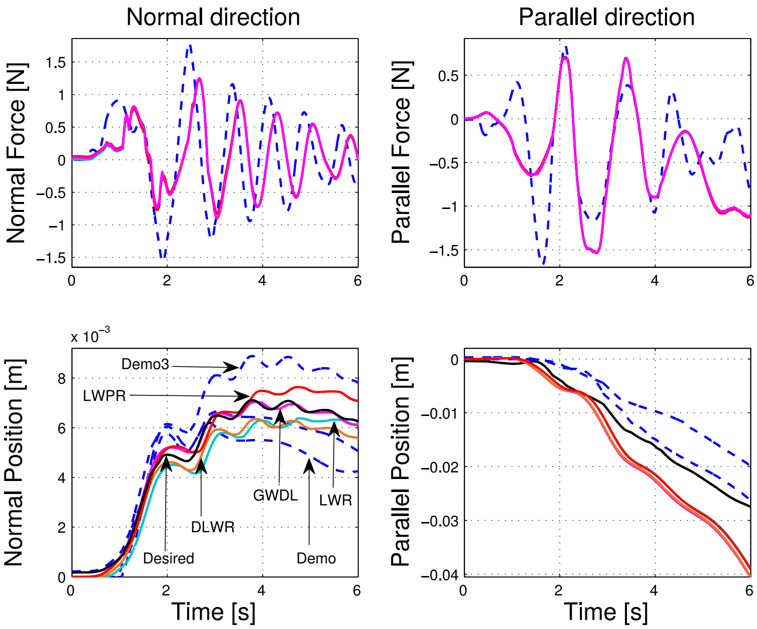


Fig. 4 Test case 1. **Top row**: (*blue dashed*) demonstrated force, (*magenta*) generalized force by GWDL. **Bottom row**: position (*blue dashed*) demonstrated, (*turquoise*) generalized by LWR, (*orange*) generalized by DLWR, (*red*) generalized by LWPR, (*magenta*) generalized by GWDL. Desired position trajectories $x_{(n)}$ and $x_{(p)}$ are shown by a black-color line. 'Demo' stands for the demonstrated trajectory

where $x^{(3)}_{(p)}$ and $x^{(3)}_{(n)}$ are position trajectories of the third demonstration while $\dot{x}_{(n)}$ and $\ddot{x}_{(n)}$ are computed from $x_{(n)}$. Generalization is realized by LWR, LWPR, DLWR and GWDL and the results are shown in Fig. 4. The figure depicts demonstrated and generalized forces as well as measured positions for all the four methods. The generalized force profiles of LWR, LWPR and DLWR are overlapped by the force profile of GWDL in the figure due to the small differences between their values. From Fig. 4, we observe that GWDL achieves accurate execution of the desired normal trajectory $x_{(n)}$ while LWR, DLWR and LWPR exhibit lower performance and do not efficiently approximate $x_{(n)}$. On the other hand, we observe that all the methods exhibit almost the same performance in the parallel direction of movement where the tracking error is observed to be higher than that of the normal direction. Table 1 shows the values of the tracking errors (16) for all the methods. In this task, although the desired parallel movement $x_{(p)}$ belongs to demonstrations, generalization performance in the parallel direction exhibits relatively high error which is probably due to the normal position and velocity components of the motion vector $\mathbf{s}_{d_{(p)}}$, which do not belong to demonstrations, see Eq. (14).

5.3.2 Test Case 2: Extrapolation to New Normal Motion

Here, we test extrapolation to a new normal positional trajectory which lies outside the experienced position range. The desired trajectories are

$$x_{(p)} = x^{(3)}_{(p)}, \quad x_{(n)} = 1.3 \times x^{(3)}_{(n)}$$

where $\dot{x}_{(n)}$ and $\ddot{x}_{(n)}$ are computed from $x_{(n)}$. Figure 5 visualizes the demonstrated and generalized forces by LWR, LWPR, DLWR and GWDL as well as the corresponding measured position for each method. The

Table 1 Tracking errors of LWR, LWPR, DLWR and GWDL for test cases 1, 2 and 3

Test case	Error	LWR	DLWR	LWPR	GWDL
Case 1	$E_{(n)}$	0.0027	0.0027	0.0014	0.0004
Case 1	$E_{(p)}$	0.1244	0.1310	0.1283	0.1398
Case 2	$E_{(n)}$	0.0066	0.0059	0.0172	0.0033
Case 2	$E_{(p)}$	0.1035	0.0602	0.1193	0.029
Case 3	$E_{(n)}$	0.004	0.0025	0.0016	0.000648
Case 3	$E_{(p)}$	0.0332	0.0356	0.0361	0.0374

Errors are expressed in $[m]$.

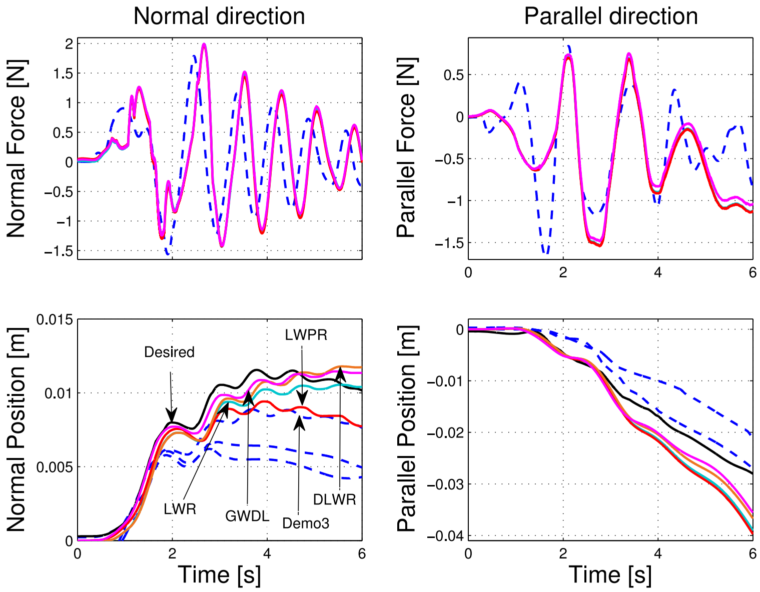


Fig. 5 Test case 2. **Top row**: (*blue dashed*) demonstrated force, (*magenta*) generalized force by GWDL. **Bottom row**: position (*blue dashed*) demonstrated, (*turquoise*) generalized by LWR, (*orange*) generalized by DLWR, (*red*) generalized by LWPR, (*magenta*) generalized by GWDL. Desired position trajectories $x_{(n)}$ and $x_{(p)}$ are shown by a black-color line. 'Demo' stands for the demonstrated trajectory

force profiles of LWR, LWPR and DLWR are overlapped by the force profile of GWDL in the figure due to the small difference between their values. Table 1 shows the values of the tracking errors (16) for the four methods. We observe that DLWR and GWDL have similar performance in the normal direction and outperform LWR and LWPR. LWPR and LWR exhibit lower performance in both directions of movement. Note that the generalized position by LWPR closely approximates the demonstrated trajectory $x^{(3)}_{(n)}$. Although DLWR and GWDL outperform the two other methods, they still exhibit limitation in following the desired position trajectories. This is expected given that extrapolation is harder than interpolation [37].

5.3.3 Test Case 3: Generalization to New Normal and Parallel Motion

In this task, we perform interpolation to a new normal trajectory same as in test case 1 and extrapolation to a new parallel trajectory. The desired positions are

$$x_{(p)} = 1.2 \times x^{(3)}_{(p)}, \quad x_{(n)} = 0.8 \times x^{(3)}_{(n)}$$

while corresponding velocity and acceleration values of both directions are computed from the positions $x_{(n)}$ and $x_{(p)}$. Figure 6 shows the results of generalization by LWR, LWPR, DLWR and GWDL while Table 1 shows the values of the corresponding tracking errors (16) for each method. We observe that GWDL outperforms the other methods in the normal direction of movement and achieves to approximate the desired trajectory with low error. On the other hand, LWR exhibits the lowest performance in the normal direction. In the parallel direction, we observe that all the methods exhibit almost the same performance and they successfully follow $x_{(p)}$. Note that all the methods exhibit higher normal tracking error compared to the error of the test case 1. Given that the normal motion vector $\mathbf{s}_{d_{(n)}}$ also depends on the parallel motion states which now lie outside the experienced data range, it is expected to notice some higher normal error here compared to case 1.

5.3.4 Comparison of Test Cases 1, 2 and 3

By considering all the previous test cases, we notice some compromise in performance of GWDL between the normal and parallel direction of movement. This derives from the fact that the motion vectors $\mathbf{s}_{d_{(n)}}$ and $\mathbf{s}_{d_{(p)}}$ are coupled through position and velocity and, thus, generalization in one direction depends on the motion states of the other direction as well. Someone would think of decoupling $\mathbf{s}_{d_{(n)}}$ and $\mathbf{s}_{d_{(p)}}$ but, in that case, the input motion space would not correspond to the true physics of the task. Given all

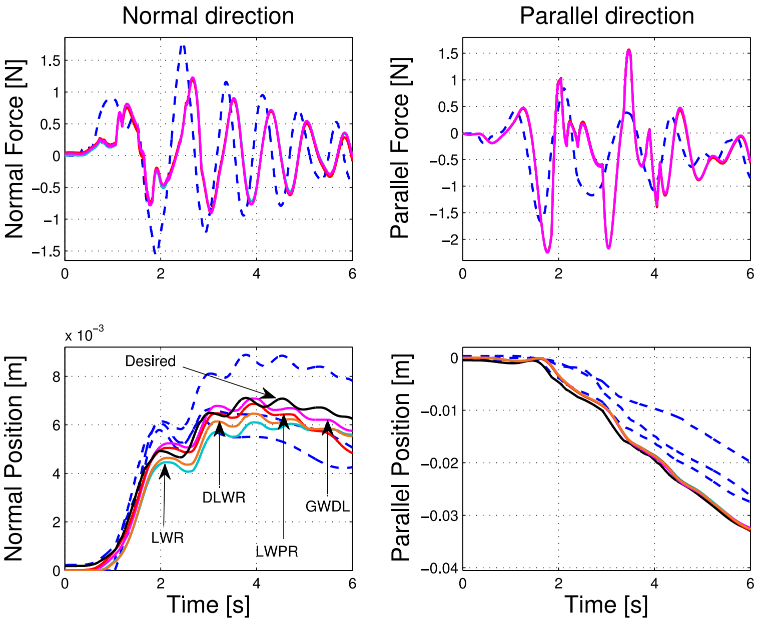


Fig. 6 Test case 3. **Top row**: (*blue dashed*) demonstrated force, (*magenta*) generalized force by GWDL. **Bottom row**: position (*blue dashed*) demonstrated, (*turquoise*) generalized by LWR, (*orange*) generalized by DLWR, (*red*) generalized by LWPR, (*magenta*) generalized by GWDL. Desired positional trajectories $x_{(n)}$ and $x_{(p)}$ are shown by a black-color line

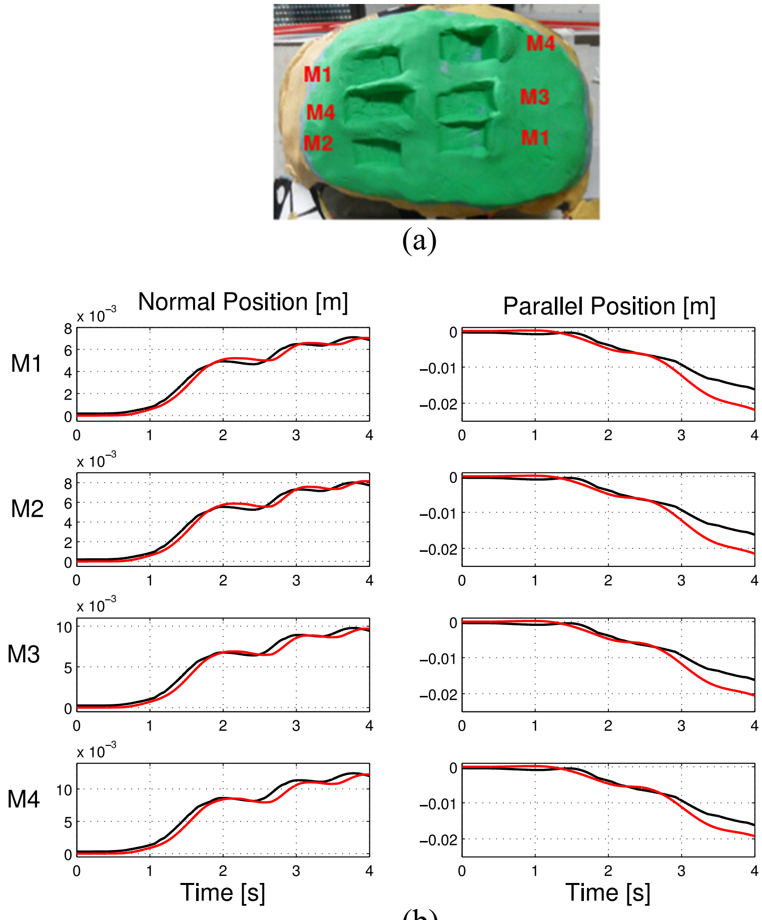
the test cases, we observe that the GWDL approach performs in overall most successfully among all the tested approaches within the range of examined tasks and proves a promising technique for approximating unknown motor control policies from data. DLWR exhibits in overall lower performance than GWDL. The low performance of DLWR could motivate for future modifications of the algorithm in the following ways: (i) apart from learning only the $\mathbf{b}_m$ parameters in Eq. 12, the weigths w_m should also be learned, and (ii) the increment-based law (13) may have to be modified in a certain way for better performance. Given that humans follow a computationally complex strategy of endpoint force regulation while performing constrained movements under varying motion-dependent disturbances, expressing this strategy from few data and also by employing relatively simple estimation techniques is a big challenge.

5.3.5 Further Evaluation of GWDL

In many real-world scenarios, robots are expected to execute constrained motion tasks such as writing or engraving on an object [38], sculpting [39] or cutting of human tissue in robotic surgery [40]. In all previous cases, execution of accurate motions in multiple directions of movement simultaneously is required by ideally taking into account the properties of the environment and the end-effector. One route to solving such constrained motion problems is the application of feedback control such as stiff position or adaptive control [41] or, for instance, a force-controlled velocity planning approach [42]. A different route to executing a motion task is learning of the task dynamics by imitation first and then employing the learned skills to successfully reproduce the task in future trials. One of the main advantages of learning by imitation is that it allows to learn a human-like force tuning policy and treat the environment as an expert task demonstrator would do in contrast to control schemes which may apply higher than desired forces on the object and violate the physical constraints of the system. It is obvious that, in constrained motion tasks such as robotic surgery for hip replacement where high-standard safety and absolute guarantee of performance are required, the ideal policy would consist of combining feedforward control skills learned by imitation with feedback control to efficiently deal with perturbations. In the present article, we limit our investigation to learning feedforward control policies by demonstration.

Here, we further evaluate the efficiency of GWDL by showing generalization to constrained motions executed onto a plasticine object whose surface is planar

Fig. 7 **a** Generalizing to new motions by GWDL. **b** *Black line*: desired trajectory, *red line*: generalized trajectory



(a)

(b)

prior to the task. Following learning from the three demonstrations as in the previous test cases, learned force control policies are generalized by GWDL to the following motions:

- Movement 1 (M1): $x_{(n)} = 0.8 \times x^{(3)}_{(n)},\ x_{(p)} = x^{(3)}_{(p)}$.
- Movement 2 (M2): $x_{(n)} = 0.9 \times x^{(3)}_{(n)},\ x_{(p)} = x^{(3)}_{(p)}$.
- Movement 3 (M3): $x_{(n)} = 1.1 \times x^{(3)}_{(n)},\ x_{(p)} = x^{(3)}_{(p)}$.
- Movement 4 (M4): $x_{(n)} = 1.4 \times x^{(3)}_{(n)},\ x_{(p)} = x^{(3)}_{(p)}$.

Figure 7b shows the desired movements M1, M2, M3 and M4 and corresponding generalized movements while Fig. 7a visualizes the effect of the generalized movements on the plasticine material. From Fig. 7b, we observe that GWDL efficiently generalizes to new normal movements while its generalization in the parallel direction exhibits lower efficiency. This may be due to increased task complexity in the parallel direction because of the larger interaction surface of the tool with the environment in this direction, which makes the tool encounter larger perturbations in that direction. We also observe that interpolation exceeds extrapolation performance.

6 Discussion

In this work, we are interested in learning force generalization skills from task demonstrations with the goal of executing constrained movements inside deformable materials. In our scenario, we assume that the environment is sufficiently homogeneous and external forces only depend on the motion states of the task. Given that forces are only state-dependent, by estimating the mapping between applied forces and visited motion states, the force control policy of the task can be estimated and generalized to similar motions within the same environment in the future. Humans build force-motion mappings, known as internal inverse dynamics models, for feedforward motion control and trajectory planning [3]. In this work, we learn force-motion mapping policies from a finite set of demonstrated data in order to endow robots with certain human-like adroitness.

To reveal the motion-to-force mapping policy from data, we investigate the role that the policy's derivatives and differences can play in representing this mapping. A policy's derivative expresses how fast the policy varies with respect to variations in its input. The policy's derivatives are learned by LR from data and combined with learned policy data in a differential calculus-inspired fashion to give an estimate of the policy at new inputs. In addition, policy's differences express how much the policy varies from a baseline behavior given a variation of its input from the demonstrated data. Policy's differences are represented and learned by LWR and combined with policy values, also learned by LWR, to estimate the policy at new inputs. LR and LWR are static in the sense that they represent how a policy varies with respect to some fixed input in contrast to incremental techniques which update their learning structure given a new input. However, if we can make LR and LWR capable of expressing how a policy varies with respect to variations in its input, then the learned static and dynamic characteristics of the policy can be appropriately integrated in order to give an estimate of the policy at new inputs. This notion constitutes the key concept of our proposed approach.

From the experimental results, we observe that GWDL in overall outperforms DLWR, LWR and LWPR both in interpolation and extrapolation cases. Interpolation is observed to be an easier task than extrapolation for all the tested methods [37]. Although GWDL outperforms LWR, LWPR and DLWR, it still exhibits limitations in its generalization performance in extrapolation cases. We deduce that representing and generalizing motor control policies from certain human data is a challenging issue which can be further investigated in the future and in the context of regression theory in order to elucidate the motion-force associations that humans employ in specific scenarios which are of great practical importance for autonomous systems. In conclusion, GWDL proves the most successful technique for force policy generalization among all the examined methods in our scenario and can serve as a well-aimed and competent feedforward force controller in similar applications where input to output mappings are able to be learned from demonstrated data.

Acknowledgments This research has been supported by the scholarships ”Bavarian Elite Aid Act (BayEFG)” and ”TUM-Equal Opportunities for Women in Research and Academic Teaching” and partially supported by the TUM-Institute for Advanced Study.

References

1. Burdet, E., Tee, K.P., Mareels, I., Milner, T.E., Chew, C.M., Franklin, D.W., Osu, R., Kawato, M.: Stability and motor adaptation in human arm motions. Biol. Cybern. **94**, 20–32 (2006)
2. Burdet, E., Osu, R., Franklin, D.W., Milner, T.E., Kawato, M.: The central nervous system stabilizes unstable dynamics by learning optimal impedance. Nature **414**, 446–449 (2001)
3. Kawato, M.: Internal models for motor control and trajectory planning. Curr. Opin. Neurobiol. **9**, 718–727 (1999)
4. Nguyen-Tuong, D., Peters, J., Seeger, M., Schölkopf, B.: Learning Inverse Dynamics: a comparison. Eur. Symp. Artif. Neural Netw. (ESANN), 13–18 (2008)
5. Sun de la Cruz, J., Kulić, D., Owen, W.: Online incremental learning of inverse dynamics incorporating prior knowledge. Autonomous and intelligent systems (2011)
6. Nguyen-Tuong, D., Peters, J.: Using model knowledge for learning inverse dynamics. IEEE international conference on robotics and automation (2010)
7. Rozo, L., Jiménez, P., Torras, C.: Sharpening haptic inputs for teaching a manipulation skill to a robot. In: International Conference on Applied Bionics and Biomechanics, pp. 370–377 (2010)
8. Rozo, L., Jiménez, P., Torras, C.: A robot learning from demonstration framework to perform force-based manipulation tasks. Intel. Serv. Robotics **6**(1), 33–51 (2013)
9. Kormushev, P., Calinon, S., Caldwell, D.G.: Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input. Adv. Robot. **25**(5), 581–603 (2011)
10. Rozo, L., Calinon, S., Caldwell, D.G.: Learning force and position constraints in human-robot cooperative transportation. In: Proceedings of IEEE International Symposium on Robot and Human Interactive Communication (Ro-Man) (2014)
11. Gams, A., Nemec, B., Ijspeert, A.J., Ude, A.: Coupling movement primitives: interaction with the environment and bimanual tasks. IEEE Trans. Robot. **30**(4), 816–830 (2014)
12. Ijspeert, A., Nakanishi, J., Schaal, S.: Learning attractor landscapes for learning motor primitives. In: Advances in Neural Information Processing Systems 15, pp. 1547–1554. MIT Press, Cambridge (2003)

13. Koropouli, V., Lee, D., Hirche, S.: Learning interaction control policies by demonstration. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 344–349 (2011)
14. Righetti, L., Kalakrishnan, M., Pastor, P., Binney, J., Kelly, J., Voorhies, R., Sukhatme, G., Schaal, S.: An autonomous manipulation system based on force control and optimization. Autonomous Robots, Special Issue: Autonomous Grasping and Manipulation **36**(1-2), 11–30 (2014)
15. Schmidts, A., Lee, D., Peer, A.: Imitation learning of human grasping skills from motion and force data. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1002–1007 (2011)
16. Buchli, J., Stulp, F., Theodorou, E., Schaal, S.: Learning variable impedance control. Int. J. Robot. Res., 820–833 (2011)
17. Stulp, F., Buchli, J., Ellmer, A., Mistry, M., Theodorou, E., Schaal, S.: Reinforcement learning of impedance control in stochastic force fields. In: IEEE International Conference on Development and Learning (ICDL), vol. 2, pp. 1–6 (2011)
18. Lee, D., Ott, C.: Incremental kinesthetic teaching of motion primitives using the motion refinement tube. Auton. Robot. **31**(2), 115–131 (2011)
19. Lee, D., Ott, C.: Incremental motion primitive learning by physical coaching using impedance control. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2010)
20. Ganesh, G., Jarrasse, N., Haddadin, S., Albu-Schaeffer, A., Burdet, E.: A versatile biomimetic controller for contact tooling and haptic exploration. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 3329–3334 (2012)
21. Rozo, L., Calinon, S., Caldwell, D.G., Jimenez, P., Torras, C.: Learning collaborative impedance-based robot behaviors. In: AAAI Conference on Artificial Intelligence, pp. 1422–1428 (2013)
22. Lee, D., Ott, C., Nakamura, Y.: Mimetic communication with impedance control for physical human-robot interaction. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), pp. 1535–1542 (2009)
23. Medina, J.R., Lorenz, T., Lee, D., Hirche, S.: Adaptive risk-sensitive optimal feedback control for haptic assistance. In: Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3639–3645 (2012)
24. Schaal, S., Atkeson, C.: Learning Control in Robotics. IEEE Robot. Autom. Mag. **17**(2), 20–29 (2010)
25. Kalakrishnan, M., Righetti, L., Pastor, P., Schaal, S.: Learning force control policies for compliant manipulation. Intelligent Robots and Systems (IROS), pp. 4639–4644 (2011)
26. Pastor, P., Kalakrishnan, M., Chitta, S., Theodorou, E., Schaal, S.: Skill learning and task outcome prediction for manipulation. In: International Conference on Robotics and Automation (2011)
27. Deisenroth, M.P., Fox, D., Rasmussen, C.E.: Gaussian processes for data-efficient learning in robotics and control. IEEE Transactions on Pattern Analysis and Machine Intelligence (2014)
28. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
29. Schaal, S., Atkeson, C.G.: Constructive incremental learning from only local information. Neural Comput. **10**, 2047–2084 (1997)
30. Vijayakumar, S., Schaal, S.: Locally weighted projection regression: an O(n) algorithm for incremental real time learning in high dimensional spaces. In: Proceedings of the 17th International Conference on Machine Learning, vol. 1, pp. 288–293 (2000)
31. Solak, E., Murray-Smith, R., Leithead, W.E., Leith, D.J., Rasmussen, C.E.: Derivative observations in Gaussian process models of dynamic systems. In: Advances in Neural Information Processing Systems 15, pp. 1033–1040. MIT press, Cambridge (2003)
32. Kocijan, J., Leith, D.J.: Derivative observations used in predictive control. In: Proceedings of Melecon, vol. 1, pp. 379–382 (2004)
33. Koropouli, V., Hirche, S., Lee, D.: Learning and generalizing force control policies for sculpting. Intelligent Robots and Systems (IROS), pp. 1493–1498 (2012)
34. Howard, M., Klanke, S., Gienger, M., Goerick, C., Vijayakumar, S.: A novel method for learning policies from variable constraint data. Auton. Robot. **27**(2), 105–121 (2009)
35. Calinon, S., Guenter, F., Billard, A.: On learning, representing, and generalizing a task in a humanoid robot. IEEE Transactions on Systems, Man and Cybernetics. Part B. Cybernetics: A Publication of the IEEE Systems, Man, and Cybernetics Society **37**(2), 286–298 (2007)
36. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. Knowl. Inf. Syst. **7**(3), 358–386 (2005)
37. Hahn, J.G.: The hazards of extrapolation in regression analysis. J. Qual. Technol. **9**(4) (1997)
38. Sahai, R., Griffith, S., Stoytchev, A.: Interactive identification of writing instruments and writable surfaces by a robot. In: Proceedings of Robotics Science and Systems (RSS), Workshop: Mobile Manipulation in Human Environments (2009)
39. Owen, W., Croft, E., Benhabib, B.: Stiffness optimization for two-armed robotic sculpting. Ind. Robot. Int. J. **35**(1), 46–57 (2008)
40. Kazanzides, P., Zuhars, J., Mittelstadt, B.D., Taylor, R.H.: Force sensing and control for a surgical robot. In: IEEE International Conference on Robotics and Automation, pp. 612–617 (1992)
41. Koropouli, V., Gusrialdi, A., Lee, D.: ESC-MRAC of MIMO systems for constrained robotic motion tasks in deformable environments. In: European Control Conference (ECC), pp. 2109–2114 (2014)
42. Zuhars, J., Hsia, T.C.: Nonhomogeneous material milling using a robot manipulator with force controlled velocity. In: IEEE International Conference on Robotics and Automation, vol. 2, pp. 1461–1467 (1995)