Learning to Optimize Control Policies and Evaluate Reproduction Performance from Human Demonstrations

Paul Gesel[1], Dain LaRoche[2], Sajay Arthanat[3], and Momotaz Begum[1]

***Abstract*— We are interested in learning from demonstration (LfD) that can both learn and execute a trajectory and evaluate the quality of a previously unseen trajectory in the domain of assistive robotics. To this end, we propose a novel continuous inverse optimal control (IOC) formulation that simultaneously learns an optimal time-invariant controller and an evaluation metric from human demonstrations. We assume that the expert's objective function is a weighted combination of physically meaningful basis objective functions. The evaluation metric is derived from the learned expert's objective function. The benefit of this approach is twofold: 1) the controller can be optimized with respect to the learned evaluation metric and subject to the robot's dynamic limitations and 2) the evaluation metric can evaluate the quality of a demonstrated trajectory. We validate our approach with two experiments in a robot guided therapy setting: 1) evaluating demonstrated exercises with the learned metric and 2) reproducing both unconstrained trajectories and trajectories subject to the robot's dynamic constraints.**

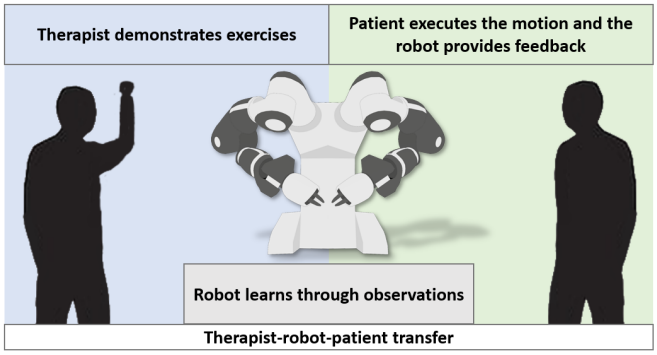

Fig. 1: Learning: the therapist demonstrates to the YuMi robot. Reproduction: the YuMi robot reproduces the motion for the patient. Evaluation: the patient executes the motion and the YuMi robot provides feedback.

I. Introduction

The goal of trajectory learning from demonstration (LfD) is to model and reproduce generalized trajectories, capable of adapting to new initial and final position, robustness to perturbations, and obstacle avoidance [1]. Trajectory LfD approaches often rely on a low-level controller to generate trajectories, including: Dynamic Movement Primitives (DMPs) [2], Stable Estimator of Dynamical Systems (SEDs) [3], Gaussian Mixture Models (GMMs) and Gaussian Mixture Regression (GMR) [4]. These approaches have shown remarkable success in learning trajectories for simple tasks, such as playing tic-tac-toe [5], pick-n-place [6], handwriting [3], and more complex tasks such as ball-in-a-cup [7], playing racket sports [8], and assistive strategies for an exoskeleton [9].

These applications typically do not require evaluating the quality of a demonstration. The provided demonstrations are usually given by an expert making the learning unidirectional. An example application where this is not the case is robot guided exercise therapy [10], [11]. Here, as illustrated in fig. 1, a domain expert (e.g. a therapist) teaches a robot, with human-like upper-body anthropomorphism, a therapeutic exercise. The robot learns and reproduces the motion for a patient who then attempts to perform the exercise originally demonstrated by the domain expert. Finally, the robot provides feedback on the quality of the reproduction by the patient. In this application, the learning occurs from two perspectives (from the therapist to the robot and the robot to the patient), which must be accounted for in the LfD algorithm. Additionally, the algorithm must: 1) learn to reproduce exercises without violating the robot's dynamic limitations, such as maximum joint velocity, 2) learn to evaluate a demonstration with respect to physically meaningful metrics, and 3) operate in a high-dimensional continuous space with limited computation. Notably in the second requirement, the patient may incorrectly execute the motion due to impairments, misunderstanding, or different starting and ending joint angles, for which the algorithm must identify as incorrect. If the LfD algorithm can capture the therapist's intent, then the therapist-robot-human skill transfer will be more effective. The state of the art in trajectory LfD does not offer any such algorithm. We propose a trajectory LfD approach to bridge this gap.

In this paper, we build upon our previous work in trajectory LfD, namely the Phase Space Model (PSM) [12]. The PSM is a dynamic system-based controller that learns a trajectory from a single demonstration [12]. We extended the PSM in [13] to include optimality with respect to a weighted combination of basis objective functions. Here, we propose an inverse optimal control (IOC) extension of the PSM in order to simultaneously learn a stable closed loop controller and an evaluation metric. This extension enables the PSM to offer distinct advantages over contemporary trajectory-LfD models, including: 1) physical trajectory characteristics, such as control effort, can be expressed as linear functions of the model parameters, 2) the PSM parameters can be optimized with respect to these quantities by solving a quadratic program,

[1] Department of Computer Science, University of New Hampshire, USA paul.gesel@unh.edu, mbegum@cs.unh.edu
[2] Department of Kinesiology, University of New Hampshire, USA Dain.LaRoche@unh.edu
[3] Department of Occupational Therapy, University of New Hampshire, USA Sajay.Arthanat@unh.edu

3) the dynamic limitations of a robot can be expressed as linear inequality constraints in the optimization, and 4) the quality of demonstrated motion can be determined with a learned objective function composed of physical trajectory characteristics. Since PSM parameter optimizations can be formulated as quadratic programs, the PSM offers a distinct computational advantage for iterative IOC methods.

II. Related Works

In this section, we summarize works on algorithms for trajectory-LfD and how the concept of inverse reinforcement learning (IRL) has been applied to LfD. Additionally, we review robot-guided exercise training research and discuss how trajectory learning and evaluation have been employed.

A. Inverse reinforcement learning for trajectory LfD

Providing meaningful feedback on the patient's motion requires a concept of a metric or reward criteria. IRL and IOC are promising frameworks for solving this problem. The purpose of these frameworks is to determine an objective function from demonstrations, which eliminates the need to design an objective function by hand. IRL has shown success in learning from experts. For example, in [14], expert level performance in aerial acrobatics was achieved with a remote-controlled helicopter. Behavior modeling, as seen in [15], is an IRL approach related to human motion learning. Here, the pedestrian's behavior is modeled with a policy, enabling the prediction of their future position by simulating the learned policy. These approaches, however, often suffer from tractability. Most methods require solving a Markov Decision Process (MDP) during each step of reward learning, which is computationally expensive [16]. Since formulating an MDP often relies on discretizing the action and/or state space, IRL approaches can become intractable for high dimensional spaces.

B. Trajectory LfD

Low-level trajectory learning approaches, such as DMPs [2], GMM [4], and SEDs [3], are usually computationally efficient. However, it is not trivial to formulate a convex optimization of physical trajectory characteristics with respect to their parameters. Additionally, these approaches do not explicitly consider the robot's dynamics.

Dynamic system-based learning from demonstration [2], [3], and statistical learning [17], [18] are among the state of the art methods for low-level trajectory learning. There are two primary limitations of existing LfD approaches for robot guided exercise therapy: 1) not explicitly considering the robot's dynamic limitations and 2) relying on simple hand-picked evaluation metrics to assess exercises. Approaches that exhibit the first limitation often assume the existence of a low-level controller that can track a desired kinematic trajectory. The DMP is a popular dynamic system-based approach which learns motion from a single demonstration [2]. It is robust to perturbations and small goal adaptations but does not to generate meaningful motion in situations that are significantly different than the original demonstration. Since DMP is implicitly time dependent, a heuristic is required to re-index time in the presence of large temporal perturbations. A more recent extension of DMP includes an IRL formulation in [19]. The objective function is structured in a general form, which can express non-linear terms on the state cost. However, the minimization problem is non-convex and only locally optimal solutions can be found. In [20], the DMP formulation is modified to learn from multiple demonstrations via a probabilistic approach. The trajectory learning in [4] creates a statistical representation of the demonstrated motion via a GMM. This approach learns a time-dependent trajectory distribution from multiple demonstrations. Unlike DMP, the GMM approach does not easily adapt to perturbations. Lastly, SEDs learns time-invariant dynamics from multiple demonstration through a statistical encoding with stability constraints. Similar to the GMM approach, SEDs relies on a low-level controller to follow the desired trajectory.

C. Robot-guided exercise training

Most rehabilitation approaches with socially assistive robots have relied on simple hand-picked evaluation metrics to assess the patients exercise execution. In [21], achieving high accuracy required positioning the participant in a specific location in the video and required calibration. Their metric defined success as the number of times the participant's arm reached 90 percent of its extent divided by the number of attempts by the patient. In [22], exercises performed by a participant are marked correct if their arm reaches a pre-specified goal location. Yet another example of a hand-crafted metric is seen in [23]. The robot characterizes an exercise as correct if the arm forms a specific angle with the normal of the floor.

Some more recent approaches have employed LfD powered robots for the exercise therapy setting. In [24], An LfD powered fitness coach is developed and exhibits some limitation. The proposed evaluation metric directly compares the patient's and therapist's joint trajectories with Dynamic Time Warping (DTW). This enables the robot to identify incorrectly executed movements, but it does not necessarily provide meaningful feedback. In [10], the GMM method from [4] was applied to the therapeutic exercise setting. The advantage of this approach is the ability to generate robot trajectories and assess the patient execution with a single framework. However, this approach does not take the robot's dynamics into account when reproducing the motion nor does it provide specific feedback on the execution. In [25], a spiking neural network is proposed to learn motions from an expert in the exercise setting. This work, however, does not provide an evaluation metric. Previous works have noted the importance of considering the robots dynamic limitations in reproducing motions. In [24], an optimization problem is formulated that maximizes the robot's reproduction accuracy, while considering the stability of the robot.

III. The Proposed Approach

By formulating the IOC problem for the PSM, we can simultaneously learn an evaluation metric and reproduce

optimal trajectories in a single framework. The set of viable basis objective functions for the PSM is expressive enough to represent several quantitative metrics seen in the therapeutic exercise literature [26], [27]. We assume a weighted combination of these metrics compose the expert's objective for therapeutic exercises. We formulate an optimization problem to learn the weights from expert demonstrations. The learned objective function 1) allows the robot to reproduce trajectories that are optimized with respect to the expert's objective, while not exceeding its dynamic limitations and 2) offers a way to quantitatively evaluate a patient's exercise execution. Fulfilling these two criteria will help the robot to transfer skill from the therapist to the patient.

A. PSM review

The PSM is a trajectory-LfD framework that models n-dimensional trajectories with piece-wise linear time invariant differential equations. The PSM formulation requires that trajectories abide by the following criteria:

1) Trajectories are governed by second order dynamics.
2) The acceleration is directly controllable.
3) There exists a direction in the demonstration's n-dimensional space, such that the velocity projected onto that direction is always positive.
4) The position and velocity are observable and the noise is negligible.

The first assumption implies that the position and velocity are part of the system's state and continuous. Let $q \in \mathbb{R}^n$ be an n-dimensional position variable. It follows that $\dot{q} \in \mathbb{R}^n$ and $\ddot{q} \in \mathbb{R}^n$ are vectors of the velocity and acceleration variables. To achieve time invariant control, the acceleration is modeled as a function of state, that is $\ddot{q} = h(q, \dot{q})$, where $h : \mathbb{R}^{2n} \mapsto \mathbb{R}^n$. Assumptions 2 and 4 are required for this control structure. Assumption 3 implies that there exists a matrix transformation V, such that $\dot{q'} = V^{-1}\dot{q}$ and $\dot{q'}_1 > 0$ for all time. This transformation aligns the first dimension's axis with the direction referred to in assumption 3. We use the prime notation to denote position, velocity, and acceleration in transformed coordinates. The concept of the PSM controller is described in eqs. (1-11). First, the system's position and velocity are projected into transformed coordinates with eqs. (1, 2). In the transformed coordinates, the acceleration $\ddot{\hat{q}}$ is a piece-wise function of position q' parameterized by $k \in \mathbb{R}^{s-1}$, $c \in \mathbb{R}^{s-1}$, and parameter tensor $A \in \mathbb{R}^{p^o \times (s-1) \times (n-1)}$ as seen in eqs. (9, 10). Here, p^o is the polynomial order in eq. (6). The hat notion refers to the modeled trajectory values. The piece-wise differential equations require cut points distributed along q'_1. Let $c^p \in \mathbb{R}^s$ be an ordered vector of constants from q'^{min} to q'^{max}. For example, $c^p = [0, 1, 2, 3, 4]^T$, $q'^{min} = 0$, $q'^{max} = 4$, and $s = 5$. Several equations are indexed with i, which ranges from 2 to n. In the transformed coordinates, eqs. (4-8) correspond to physically meaningful quantities: eq. (4) describes the control effort, eq. (5) describes the kinetic energy, eq. (6) describes the position, eq. (7) describes the direction of motion, and eq. (8) describes the curvature of motion. Finally, the acceleration is projected back into the original coordinates as seen in eq. (11). Feedback terms p and d act as a PD controller to reject perturbations.

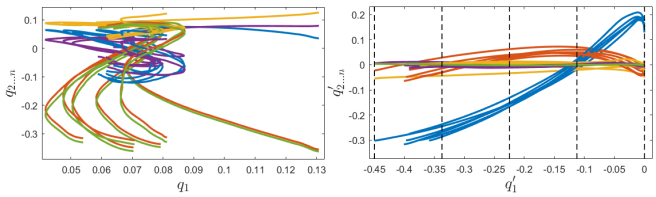

Fig. 2: left: demonstrations in the original coordinates as a function of q_1; right: demonstrations in eigenvector coordinates as a function of q'_1. The five vertical dashed black lines correspond to cut point vector c^p with $s = 5$.

$$q' = V^{-1}q \tag{1}$$

$$\dot{q'} = V^{-1}\dot{q} \tag{2}$$

$$j(x) = \underset{y \in \{1 \ldots s\}}{\text{argmax}} \{y, \mid x - c^p_y > 0\} \tag{3}$$

$$\hat{u} = k_{j(x)}x + c_{j(x)} \tag{4}$$

$$\dot{\hat{q}}^2_1 = 2\int_{q'^I_1}^{q'_1} \hat{u} dx + \dot{q'}^{I\,2}_1 \tag{5}$$

$$\hat{q}_i = \sum_{d=1}^{p^o} A_{dj(q')(i-1)} q'^{\,d-1}_1 \tag{6}$$

$$\frac{d\hat{q}_i}{d\hat{q}_1} = \sum_{d=1}^{p^o} (d-1) A_{dj(q')(i-1)} q'^{\,d-2}_1 \tag{7}$$

$$\frac{d^2\hat{q}_i}{d\hat{q}^2_1} = \sum_{d=1}^{p^o} (d-2)(d-1) A_{dj(q')(i-1)} q'^{\,d-3}_1 \tag{8}$$

$$\ddot{\hat{q}}_1 = \hat{u} \tag{9}$$

$$\ddot{\hat{q}}_i = \frac{d^2\hat{q}_i}{d\hat{q}^2_1}\dot{\hat{q}}^2_1 + \frac{d\hat{q}_i}{d\hat{q}_1}\ddot{\hat{q}}_1 \tag{10}$$

$$\ddot{q} = V(\ddot{\hat{q}} + p(\hat{q} - q') + d(\dot{\hat{q}} - \dot{q'})) \tag{11}$$

The PSM must be instantiated with initial position $q'^I \in \mathbb{R}^n$ and velocity $\dot{q'}^I \in \mathbb{R}^n$ in order to optimize the parameters. Equality constraints must be imposed on $\hat{q}_i$, $\frac{d\hat{q}_i}{d\hat{q}_1}$, $\frac{d\hat{q}^2_i}{d\hat{q}^2_1}$, and $\hat{u}$ at $c^p_2, c^p_3, \ldots c^p_{s-1}$ to ensure that the piece-wise functions are continuous. Additionally, constraints for the initial and final position and velocity must added. We define the set $\mathcal{C}$ to contain all constraints the PSM optimization is subjected to. In total, there are $n^p = p^o(s-1)(n-1)+2(s-1)$ parameters and $n^c = 3(n-1)+s$ constraints, resulting in $n^f = n^p - n^c$ free parameters. Inequality constraints must be added to ensure $\dot{\hat{q}}^2$ is always greater than zero, so that assumption 3 is not violated. Imposing the previously discussed constraints will ensure eq. (11) is a stable closed loop controller.

We use the demonstration's eigenvector matrix as the linear transformation V. In general, the transformation does not guarantee q'_1 is always increasing, hence we set any negative $\dot{q'}_1$ to zero. Notably, if a time invariant controller is not required, then time may be used for q_1 and the identity matrix for V. Fig. 2 illustrates the coordinate system transformation with the eigenvector matrix. The main benefit of this formulation is that trajectory characteristics, such as $\hat{q}$ and $\dot{\hat{q}}^2_1$

are expressed as linear combination of PSM parameters A, k, and c. Consequently, a convex optimization can be formulated given quadratic objective functions. Additionally, constraints can be added to ensure physically feasible solutions.

B. Quadratic basis objective functions

The PSM parameters can be optimized with respect to a weighted combination of quadratic basis objective functions. We define a PSM trajectory characteristic function (TCF) as a mapping from q_1' to a physical trajectory characteristic. Example trajectory characteristics include: control effort, displacement, direction, etc. We only consider TCFs (f) that can be expressed as a linear combination of PSM parameters, as seen in eq. (12).

$$p_i = [k_i, c_i, A_i^f]^T$$

$$f(q_1') = \sum_{i}^{n^p} m_i(q_1')p_i \tag{12}$$

Here, $m_i : \mathbb{R} \mapsto \mathbb{R}$ is a function of q_1' and A^f is the tensor A flattened into a one dimensional vector. All possible PSM quadratic basis objective functions (J^{bm}) follow the structure of eq. (13).

$$J^{bm} = \int_{q_1'^I}^{q_1'^F} (f(x) - v(x))^2 dx \tag{13}$$

Here, $q_1'^F$ is the final position, and v is a function from q_1' to a desired TCF value.

Similar to the PSM TCF, we define an expert TCF $g_d : \mathbb{R} \mapsto \mathbb{R}$ as a function from q_1' to the physical trajectory characteristic. Each demonstration has unique trajectory characteristics, hence the expert TCF is dependent on the demonstration index d, where $d \in \{1, 2...n^d\}$ and n^d is the number of demonstrations. It follows that the expert basis objective function (J^{be}) in eq. (14) is a function of the specific demonstration.

$$J^{be}(d) = \int_{q_1'^I}^{q_1'^F} (g_d(q_1') - v(q_1'))^2 dx \tag{14}$$

We assume that the expert's objective function (J^e) is composed of a weighted combination of n^b quadratic basis objective functions, as shown in eq. (15),

$$w^T J^e(d) = w_1 J_1^{be}(d) + w_2 J_2^{be}(d) + ...w_{n^b} J_{n^b}^{be}(d) \tag{15}$$

where w is a vector of weights. The corresponding PSM objective function (J^m) is show in eq. (16).

$$w^T J^m = w_1 J_1^{bm} + w_2 J_2^{bm} + ...w_{n^b} J_{n^b}^{bm} \tag{16}$$

Since the basis objective functions are quadratic, the optimal PSM parameters corresponding to J^m can then be solved with a quadratic program.

Eqs. (4-8) are linear functions of PSM parameters, hence, they are valid TCFs. The desired TCF value should selected such that the basis objective function represents a metric described in the human movement literature. For example, the minimum control effort metric is described by setting the PSM TCF (f) to $\hat{u}$ in eq. (4) and v to 0. Table I illustrates some PSM basis objective functions analogous to those seen in the human movement literature. Notably, basis objective functions are used for each cut point of the PSM and for each physical trajectory characteristic, e.g. energy, discomfort, etc.

TABLE I PSM representations of basis objective functions from human movement literature [28] [27]. w^E, w^{dis}, w^{eff} are weight vectors corresponding to kinetic energy, discomfort, and effort objective functions. The energy objective assumes a unit mass point particle model and q^N is a neutral joint configuration.

Energy	$\int_{q_1'^I}^{q_1'^F} w_{j(x)}^E (2\int_{q_1'^I}^{x} \hat{u}dy + \dot{q_1'^I}^2)^2 dx$
Discomfort	$\int_{q_1'^I}^{q_1'^F} w_{j(x)}^{dis} (\hat{q} - q^N)^2 dx$
Effort	$\int_{q_1'^I}^{q_1'^F} w_{j(x)}^{eff} (\hat{u})^2 dx$

C. Learning weights

We want to find a weight vector w that minimizes the difference of the expert and PSM basis objective functions. Given a set of basis objective functions, the inverse learning problem can be formulated. We express this difference for the dth demonstration in eq. (17).

$$J(p, w, d) = w^T J^m(p, d) - w^T J^e(d) \tag{17}$$

Here, p denotes the PSM parameters. The PSM objective function J^m is written as a function of the demonstration index because the quadratic basis objective functions are integrated over the demonstration's q_1' dimension. For the PSM and expert objective function to correspond, the initial and final positions of a demonstration must be used for $q_1'^I$ to $q_1'^F$. The goal is to learn the PSM parameters with the minimum possible under-performance with respect to the demonstrator, given the worst-case possible realization of the weight vector. We describe this objective in eq. (18).

$$\max_{w \in \mathbb{R}^{n^b}} \left\{ \frac{1}{n^d} \sum_{d=1}^{n^d} \min_{p \in \mathbb{R}^{n^p}} \left\{ J(p, w, d) \mid \mathcal{C}^d \right\} \mid \|w\|_\infty \leq 1, w \geq 0 \right\} \tag{18}$$

Here, $\mathcal{C}^d$ refers to the PSM constraints discussed in the previous section instantiated for the dth demonstration. Notably, we do not include a constraint on the final position and velocity in the optimization. The inner minimization is convex and can be solved with a quadratic program. For a small step in w, we assume p is constant, yielding the following gradient approximation in w.

$$\frac{1}{n^d} \sum_{d=1}^{n^d} J^m(p, d) - J^e(d) \tag{19}$$

We use a constrained nonlinear gradient-based optimizer to learn the weight vector w. Conceptually, the gradient indicates that weights of PSM basis objective functions with lower error

than those of the demonstration's basis objective functions should be increased.

Once the weights are learned, the expert objective function value $w^T J^e$ can be used to evaluate a given demonstration. The final penalty is shown in eq. (20).

$$penalty = \frac{w^T J^e}{w^T J^m} \tag{20}$$

Notably, the value is normalized by the PSM objective function value $w^T J^m$ to ensure trajectories with small $q_1'^F - q_1'^I$ are appropriately penalized.

IV. Experimental Results

We conducted a human subject study in a therapeutic exercise setting. A therapist performed several commonly prescribed therapeutic exercises, namely shoulder press, lateral raise, forward raise, and scaption. The therapist provided 3 expert demonstrations for training the models and 6 demonstrations for testing the models, composed of 3 correct and 3 erroneous demonstrations. We denote the training demonstrations as the 'therapist demonstration set', the 3 additional correct demonstrations as the 'correct patient demonstration set', and the erroneous demonstrations as the 'erroneous patient demonstration set'. Given these demonstrations, we will show two primary results. First, our method can learn the weights of the expert's objective function and thereby reproduce the exercises. Second, the learned objective can be used to distinguish correctly and incorrectly performed exercises.

All demonstrations were recorded with a Qualisys motion capture system at 300 Hz. The demonstrations were smoothed to eliminate noise and subsequently cropped to only include the concentric phase of each exercise. Demonstrations were then converted to the 4-dimensional joint space of the robot, via a vector based inverse kinematic solution [13]. This procedure calculates joint configurations for the robot that correspond to the therapist's configuration for each demonstration. Fig. 3 illustrates the therapist demonstrating the exercises and the robot's corresponding joint configurations. For our implementation, we construct a PSM controller with 6 cut points for each exercise. Each PSM controller has its own transformed coordinate system q', and the cut points are distributed along the q_1' axis. The cut points are chosen such that all transformed demonstrations are between c_1^p and c_s^p. We used basis objective functions corresponding to control effort, kinetic energy, position, direction, and curvature for all exercise models. We validate our approach in the next sections with exercise reproduction and patient evaluation experiments.

A. Experiment 1: exercise reproduction

Each PSM was trained with the 'therapist demonstration set' using the weight learning approach described in section 3.3. As a baseline comparison, we implemented DMP as described in [20] to learn exercises from multiple demonstrations. Similarly, we set the initial conditions and goal equal to that of the demonstrations. Fig. 4 shows both PSM and DMP trajectories reproduced for the forward raise exercise. In addition to reproducing expert trajectories, inequality constraints can be added to the PSM optimization to ensure that the robot's dynamic limitations are not exceeded. Fig. 4 illustrates a trajectory reproduction with a maximum $\dot{q'}_1$ of $\sqrt{2}\ \frac{rad}{s}$. Both methods generate trajectories similar to the expert, but the PSM benefits from slightly improved accuracy. However, the main benefit of the PSM is that explicit inequality constraints can be added, yielding optimal behavior subject to the robot's dynamic limitations. In fig. 4, the velocity constraint scales the motion over time, while maintaining the shape of the movement and thus preserving the integrity of the exercise. The constraint of $\sqrt{2}\ \frac{rad}{s}$ was arbitrarily selected to demonstrate the time scaling.

B. Experiment 2: patient evaluation

Using the trained PSMs from the previous experiment, we evaluated both the correct and erroneous patient demonstration sets with eq. (20). As a baseline comparison, we used the GMM approach proposed in [29], [30] and implemented in [10]. This approach is explicitly time dependent, requiring that demonstrated trajectories be aligned in time. Hence, all times were scaled between 0 and 1 for the GMM approach. Similar to the PSMs, the GMMs were trained on the 'therapist demonstration set' and evaluated on the correct and erroneous patient demonstration sets. We used the suggested expectation maximization algorithm to train the GMMs. Additionally, we used the evaluation metric from [10] with the full covariance matrix as the weight term.

The minimum, maximum, and mean for each of the PSM and GMM log scaled evaluations are shown in fig. 5. Both methods yield a considerably reduced penalty on the correctly performed exercises as compared to the erroneous exercises. However, the PSM approach results in a tighter range on the correct demonstrations. The consistent penalty value for the PSM approach suggests that the learned objective function captures key characteristics of the correctly demonstrated exercises.

In addition to generating a total penalty, a subset of basis objective functions can be used to give feedback on specific meaningful physical quantities. For example, fig. 6 shows the combined error for the all basis objective functions corresponding to the movement direction. Differentiating the combined basis objective functions with respect to q_1', generates an instantaneous penalty for the direction basis objective functions. We denote this quantity as the instantaneous penalty. Fig. 6 illustrates the instantaneous penalty for the direction basis objective functions alongside the direction the PSM and erroneous patient. The deviation in the directions are reflected in the instantaneous penalty.

Given the instantaneous penalties that reflect physically meaningful quantities, a simple heuristic can be applied to extract feedback for the demonstrator. For example, if the difference between the demonstrated and optimal trajectory instantaneous penalties is exceeds a threshold value, then the user can be given feedback.

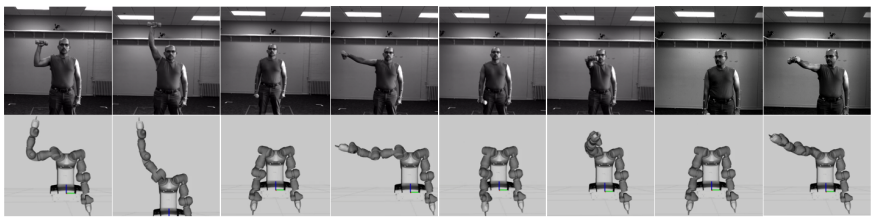

Fig. 3: robot configurations corresponding to the therapist's demonstrations for shoulder press, lateral raise, forward raise, and scaption.

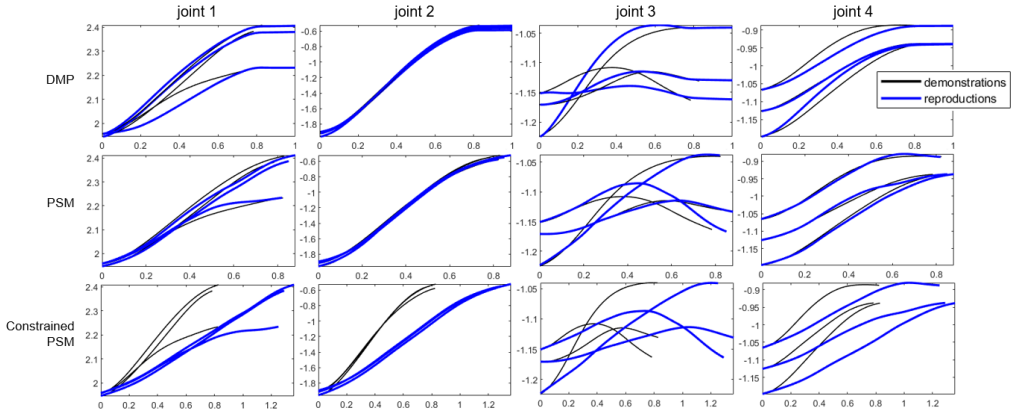

Fig. 4: PSM, DMP, and the expert joint trajectories (rad) are plotted as a function of time (s) for the forward raise exercise. The constraint $\dot{q'}_1 \leq \sqrt{2}\frac{rad}{s}$ is imposed on the PSM trajectories in the last row.

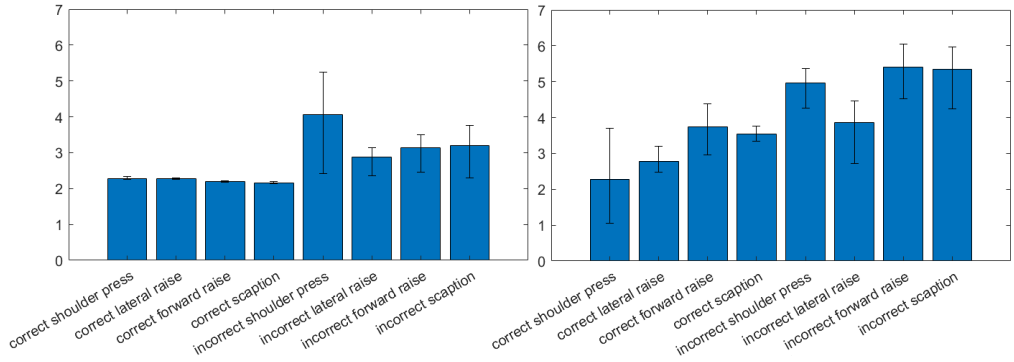

Fig. 5: left: log scaled PSM penalties for correct and incorrect demonstrations; right: log scaled GMM penalties

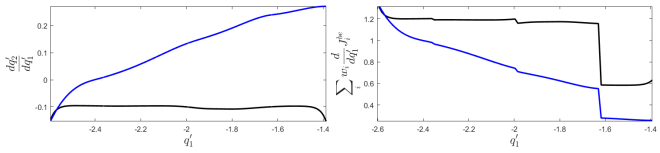

Fig. 6: left: the optimal direction $\frac{q'_2}{q'_1}$ (in blue) and an erroneous demonstration (in black) for the lateral raise exercise; right: instantaneous penalty for the direction basis objective functions of the PSM and erroneous demonstration. Here, i indexes all direction basis objective functions.

V. Conclusion

We proposed a novel IOC approach based on the PSM controller and validated our approach with results in the therapeutic exercise domain. The weights of the PSM basis objective functions were successfully learned for all exercises and the robot was able to reproduce trajectories similar the expert demonstrations. The learned objective function also served as an effective evaluation metric to give quantitative feedback on physically meaningful trajectory characteristics. Finally, the proposed approach yields a closed loop controller, which can be optimized subject to dynamic limitations, such as velocity limits. Given the learned controller, the robot can help transfer skills from the therapist to the patient

Acknowledgment

This work was supported in part by the National Science Foundation (IIS 1830597).

References

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robot. Auton. Syst.*,

vol. 57, no. 5, p. 469–483, May 2009. [Online]. Available: https://doi.org/10.1016/j.robot.2008.10.024

[2] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 763–768.

[3] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, Oct 2011.

[4] A. B. S. Calinon, F. Guenter, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 2, pp. 286–298, 2007.

[5] S. Niekum, S. Osentoski, G. Konidaris, S. Chitta, B. Marthi, and A. G. Barto, "Learning grounded finite-state representations from unstructured demonstrations," *The International Journal of Robotics Research*, vol. 34, no. 2, pp. 131–157, 2015.

[6] F. Stulp, E. A. Theodorou, and S. Schaal, "Reinforcement learning with sequences of motion primitives for robust manipulation," *IEEE Transactions on robotics*, vol. 28, no. 6, pp. 1360–1370, 2012.

[7] J. Kober and J. Peters, "Policy search for motor primitives in robotics," *Mach. Learn. J*, vol. 84, pp. 171–203, 01 2008.

[8] K. Mülling, J. Kober, and J. Peters, "Learning table tennis with a mixture of motor primitives," *Proceedings of the 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2010), 411-416 (2010)*, 12 2010.

[9] M. Hamaya, T. Matsubara, T. Noda, T. Teramae, and J. Morimoto, "Learning assistive strategies for exoskeleton robots from user-robot physical interaction," *Pattern Recognition Letters*, vol. 99, pp. 67 – 76, 2017, user Profiling and Behavior Adaptation for Human-Robot Interaction. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865517301198

[10] S. M. Nguyen, P. Tanguy, and O. Remy-Neris, "Computational architecture of a robot coach for physical exercises in kinaesthetic rehabilitation," in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2016, pp. 1138–1143.

[11] Y. Meng, C. Munroe, Y. Wu, and M. Begum, "A learning from demonstration framework to promote home-based neuromotor rehabilitation," in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2016, pp. 1126–1131.

[12] P. Gesel, M. Begum, and D. L. Roche, "Learning motion trajectories from phase space analysis of the demonstration," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 7055–7061.

[13] P. Gesel, D. LaRoche, S. Arthanat, and M. Begum, "Learning adaptive human motion via phase space analysis of demonstrated trajectories," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[21] P. Gadde, H. Kharrazi, H. Patel, and K. MacDorman, "Toward monitoring and increasing exercise adherence in older adults by robotic intervention: A proof of concept study," *Journal of Robotics*, vol. 2011, 08 2011.

[14] P. Abbeel, A. Coates, and A. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *I. J. Robotic Res.*, vol. 29, pp. 1608–1639, 11 2010.

[15] B. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. Bagnell, M. Hebert, A. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," 12 2009, pp. 3931–3936.

[16] K. Li and J. Burdick, "Inverse reinforcement learning in large state spaces via function approximation," 07 2017.

[17] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4674–4683, 2017.

[18] Y. Cheng, W. Zhao, C. Liu, and M. Tomizuka, "Human motion prediction using adaptable neural networks," *ArXiv*, vol. abs/1810.00781, 2018.

[19] M. Kalakrishnan, P. Pastor, L. Righetti, and S. Schaal, "Learning objective functions for manipulation," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 1331–1336.

[20] C. Chen, C. Yang, C. Zeng, N. Wang, and Z. Li, "Robot learning from multiple demonstrations with dynamic movement primitive," in *2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM)*, 2017, pp. 523–528.

[22] J. Fasola and M. J. Mataric, "Robot exercise instructor: A socially assistive robot system to monitor and encourage physical exercise for the elderly," in *19th International Symposium in Robot and Human Interactive Communication*, 2010, pp. 416–421.

[23] M. Matarić, J. Eriksson, D. Feil-Seifer, and C. Winstein, "Socially assistive robotics for post-stroke rehabilitation," *Journal of neuroengineering and rehabilitation*, vol. 4, p. 5, 02 2007.

[24] B. Görer, A. Salah, and H. L. Akin, "A robotic fitness coach for the elderly," 01 2013.

[25] T. Obo, C. K. Loo, and N. Kubota, "Imitation learning for daily exercise support with robot partner," in *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2015, pp. 752–757.

[26] A. Vakanski, J. Ferguson, and S. Lee, "Metrics for performance evaluation of patient exercises during physical therapy," *American Journal of Physical Medicine and Rehabilitation*, vol. 5, 06 2017.

[27] J. Yang, R. Marler, H. Kim, J. Arora, and K. Abdel-Malek, "Multi-objective optimization for upper body posture prediction," vol. 4, 08 2004.

[28] B. Berret, E. Chiovetto, F. Nori, and T. Pozzo, "Evidence for composite cost functions in arm movement planning: An inverse optimal control approach," *PLOS Computational Biology*, vol. 7, no. 10, pp. 1–18, 10 2011. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1002183

[29] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 37, no. 2, pp. 286–298, 2007.

[30] S. Calinon, *Robot Programming by Demonstration: A Probabilistic Approach*. EPFL/CRC Press, 2009, ePFL Press ISBN 978-2-940222-31-5, CRC Press ISBN 978-1-4398-0867-2.