Trajectory Learning from Human Demonstrations via Manifold Mapping

Michihisa Hiratsuka[1], Ndivhuwo Makondo[1], Benjamin Rosman[2] and Osamu Hasegawa[1]

Abstract— **This work proposes a framework that enables arbitrary robots with unknown kinematics models to imitate human demonstrations to acquire a skill, and reproduce it in real-time. The diversity of robots active in non-laboratory environments is growing constantly, and to this end we present an approach for users to be able to easily teach a skill to a robot with any body configuration. Our proposed method requires a motion trajectory obtained from human demonstrations via a Kinect sensor, which is then projected onto a corresponding human skeleton model. The kinematics mapping between the robot and the human model is learned by employing Local Procrustes Analysis, which enables the transfer of the demonstrated trajectory from the human model to the robot. Finally, the transferred trajectory is modeled using Dynamic Movement Primitives, allowing it to be reproduced in real time. Experiments in simulation on a 4 degree of freedom robot show that our method is able to correctly imitate various skills demonstrated by a human.**

I. INTRODUCTION

The robots active in real world settings are not only constantly growing in number, but their capabilities are continuously improving as well. It is envisioned that they will soon be ubiquitous in assisting humans with a variety of tasks in every day environments. It is often the case that these platforms are designed to physically resemble humans, but for many settings this is inappropriate. Either way, it is important that they can be adapted to new situations by extending their sets of behaviors or skills.

Numerous approaches have been proposed to equip robots with skills. Conventionally this is done by directly programming the robot, but this requires tedious effort to ensure the skill is natural, efficient and safe. As an alternative approach, the paradigm of Learning from Demonstration (LfD) has been extensively studied in recent years. In this approach, new trajectories are generated from learned and generalized trajectories provided from demonstrations from a human [1].

In this work, we propose a novel system to enable imitation learning for robot arms from human demonstrations by learning a kinematic mapping between a human model and a robot. The contribution of this paper is a data-driven approach based on manifold mapping that is suitable in cases where an accurate kinematics model of a robot is not available. This may be the case, for example, when the specifications required for modeling the robot are not released by the manufacturer. It is often the case that a robot manipulator may need to be modeled using parameters that were measured by hand because the manufacturer was unable to release them [2]. These hand measurements can lead to an inaccurate kinematic model, requiring further calibration. Another example is in dealing with robots whose bodies change over time, potentially as a result of modification, repair, or material damage [3].

This paper is organized as follows. In section II, we relate our problem to existing work. In section III we introduce and explain the building blocks of our system, namely Local Procrustes Analysis (LPA) and Dynamic Movement Primitives (DMPs). In section IV, we present details of the proposed system. In section V, we evaluate the performance of each component and the system through multiple experiments in simulations on a 4-DoF robot arm, and finally section VI concludes.

II. RELATED WORK

A. Learning from demonstration

The framework of LfD is divided into two fundamental phases: motion transfer and skill modeling. There are two widely used approaches to transfer a motion: kinesthetic guiding and motion capture [4]. The former physically moving the robot in question. This is an intuitive way for users to teach motions to a robot and there are no correspondence issues since the robot is typically aware of its current configuration, i.e., through joint encoders [4,5]. Resulting behaviors tend to be unnatural, because kinesthetic guiding cannot exactly imitate a skill the same way humans would naturally perform it, and it may be difficult for the human to smoothly manipulate the robot. Alternatively, in motion capture systems, users demonstrate a behavior which is extracted by the system, and then converted to a suitable form after some pre-processing steps to remove noise in order to project it onto the robot. One of the main challenges of this approach is how to project recorded data onto the body of the robot, since its morphology may be different from that of the human – the correspondence problem [4]. Typically a body-suit system [6,7] or a camera-based system [8,9] is employed as a motion capturing system. The latter is preferable for demonstrators because it does not constrain performances, allowing for more natural or complex demonstrations.

This research was supported by Core Research for Evolutional Science and Technology (CREST) program of Japan Science and Technology Agency (JST).

[1]Michihisa Hiratsuka, Ndivhuwo Makondo and Osamu Hasegawa are with the Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Yokohama, Japan. m.hiratsuka0716@gmail.com, makondo.n.aa@m.titech.ac.jp, oh@haselab.info

[2]Benjamin Rosman is with the Mobile Intelligent Autonomous Systems, Council for Scientific and Industrial Research, Pretoria, South Africa; and also with the School of Computer Science and Applied Mathematics, University of the Witwatersrand, South Africa. brosman@csir.co.za

Regarding modeling a skill or trajectory, the data-driven approach has received considerable attention recently. Reward function learning [10], statistical modeling [11] and dynamical systems [12-15] are amongst the most popular approaches. In particular, dynamic movement primitives (DMPs) are widely used due to their flexibility and stability [13,14], and involve encoding a behavior as a dynamical system.

In this work, we propose using a manifold mapping technique to deal with the correspondence problem in camera-based systems for motion transfer, and use DMPs to model the skills, as discussed in section III.

B. Skill transfer

The projection into joint space is hereafter referred to as behavior transfer, and the task space projection as skill transfer. Behavior transfer is suitable for skills that do not require complex interactions with the external world, e.g. dancing or walking. This is however unsuitable for any behavior requiring interaction with other objects, as unless the robot has exactly the same kinematic structure as the demonstrator, it cannot perform the task because the end-effector positions in task space do not match those of a human with a different morphology. In contrast, skill transfer is suitable when the target skills require an interaction with some component of the external world, such as for grasping and manipulation. To correctly execute those skills in a human environment, the robot's end-effector should track the Cartesian trajectory, such that the arm follows certain desired configurations, e.g., those similar to that of a human arm.

To achieve skill transfer, a conventional approach is to provide a Cartesian trajectory in task space and solve the inverse kinematics (IK) to obtain the corresponding joint trajectory of the robot. For example, a Kinect-based system may be used to capture human motion and project this directly onto a robot arm [16], using incremental IK to obtain a joint trajectory from a given Cartesian trajectory from a human. High quality path tracking can then be achieved with the aid of an accurate known kinematics model. In [17], they extract the end-effector Cartesian trajectory of the demonstration and find parameters of a mapping to the corresponding robot trajectory by Stochastic Optimization of the Embodiment Mapping, combining IK and the optimization.

The common hypothesis in [16] and [17] is that an accurate kinematics model exists for the target robot, so their trajectories are reliant on the quality of the model. Although these methods have been shown to work well when an accurate kinematics model of the robot is known, data-driven approaches have been proposed to deal with cases in which an accurate robot model is not available. This involves learning a mapping from human demonstrations to the robot, from corresponding samples of the human and robot data. Examples of this can be seen in applications, e.g. robotic imitation of human poses [7, 18]. Most of these works focus on behavior transfer rather than skill transfer. We adopt a data-driven approach to realize skill transfer.

III. Overview and System Components

We propose a novel framework to enable skill transfer to robots without the need for a kinematics model of the robots. The outline of the proposed system is shown in Fig. 1, and is described as follows. A demonstrator first performs the desired skill which is recorded as a trajectory of body part positions. This trajectory is then projected onto a human skeleton model, which corresponds to the demonstrator's body structure. We collect robot training data by moving the robot around its workspace, and then generate human model data that corresponds to these robot end-effector poses. This is done in the area of the workspace common to both the robot and the human model. Once we have collected a data set of corresponding configurations, we can learn a mapping between them using LPA [19] (section III-A). After learning this mapping function, any trajectory represented as a set of joint states and Cartesian positions can be transferred onto a target robot. The transferred trajectory is finally modeled as a DMP (section III-B) to improve reproduction stability.

Our system has the following features:

- Any human motion can be transferred to any robot, as long as the workspace of the robot overlaps with that of the human model. The user only needs to demonstrate the skill to the robot.
- It does not require any kinematic model for the target robots. The projection is established by collecting data of corresponding poses between humans and robots. Thus, human-likeness of the trajectory can be preserved.
- Once a skill is equipped on a robot, it can be transferred to another robot, by collecting corresponding samples in the workspaces and employing the same architecture.

A. Learning kinematics mapping: Local Procrustes Analysis

To learn a joint space mapping between a human model and a robot, we employ Local Procrustes Analysis. LPA is an extension of Procrustes Analysis (PA) [20] – a linear manifold mapping method – for non-linear mapping. The objective is to learn a mapping f that maps data points between a source manifold $\boldsymbol{Z}^s$ and a target manifold $\boldsymbol{Z}^t$. We assume that the manifolds have the same dimensions and are composed of the same number of data points. Briefly, in LPA these manifolds are separated into multiple clusters such that a set of alignment functions can describe the mapping f, assuming local linearity and smoothness of the domains.

LPA approximates a non-linear mapping as follows. First, LPA reduces the dimensionality of the manifolds to a joint latent space. Each dataset is represented by a mixture of local regions, such that each region contains data points which can be mapped from a region in the source manifold to the corresponding region in the target manifold. The regions in each manifold are described as a Gaussian Mixture Model (GMM), which is fit to the data using the Expectation-Maximization (EM) algorithm. We assume that the source manifold $\boldsymbol{Z}^s$ is to be mapped to the target manifold $\boldsymbol{Z}^t$. The manifolds are required to have the same dimension and same number of paired data points such that $\boldsymbol{z}_i^s$ corresponds

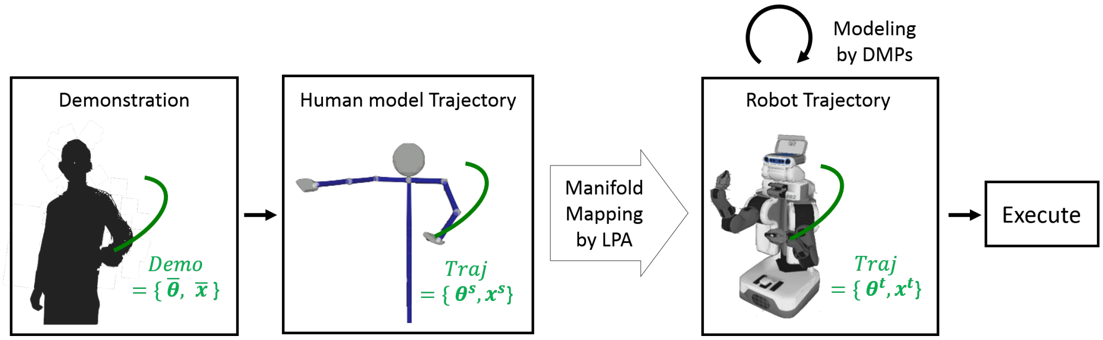


Fig. 1: Proposed system overview. The demonstration is recorded via the Kinect sensor and converted to a pair of the end-effector Cartesian trajectory and joint trajectory on a human model. This is then projected onto a robot via a mapping learned offline using LPA to obtain $\boldsymbol{\theta}^t$ from $\boldsymbol{\theta}^s$, and modeled by DMPs to be independently reproduced.

to $\boldsymbol{z}_i^t$, where $\boldsymbol{Z}^s = \{\boldsymbol{z}_i^s\}$ and $\boldsymbol{Z}^t = \{\boldsymbol{z}_i^t\}$ for $i = 1...N$. In the training phase, $\hat{\boldsymbol{Z}}^s$, dimensionally reduced from $\boldsymbol{Z}^s$, is modeled by a GMM and then clustered by assigning points to components with the highest responsibilities. After obtaining the corresponding clusters, the linear mapping of each k cluster f_k can be learned by the Procrustes method as follows.

First, data points in each cluster are transformed as

$$\boldsymbol{s} = \boldsymbol{B}_k^s(\boldsymbol{z}_k^s - \boldsymbol{w}_k^s) \tag{1}$$

$$\boldsymbol{t} = \boldsymbol{B}_k^t(\boldsymbol{z}_k^t - \boldsymbol{w}_k^t) \tag{2}$$

where $s \in \boldsymbol{M}^s$ and $t \in \boldsymbol{M}^t$ are the normalized elements, $\boldsymbol{w}_k^s$ and $\boldsymbol{w}_k^t$ are the means of the data, $\boldsymbol{z}_k^s$ and $\boldsymbol{z}_k^t$ are paired elements of each manifold in cluster k. Matrices $\boldsymbol{B}_k^s$ and $\boldsymbol{B}_k^t$ are transformation matrices obtained via Singular Value Decomposition (SVD) of $\boldsymbol{Z}_k^s$ and $\boldsymbol{Z}_k^t$ respectively.

This transform whitens $\boldsymbol{M}_s$ and $\boldsymbol{M}_t$. Then, the alignment function f_k is defined as

$$f_k(\boldsymbol{s}) = \boldsymbol{A}\boldsymbol{s}, \text{ s.t. } f_k : \boldsymbol{M}^s \mapsto \boldsymbol{M}^t \tag{3}$$

where $\boldsymbol{A}$ is a J by J matrix, and J is the dimension of the manifolds. The mapping is approximated by minimizing the expected loss of the transformation. The problem can be reformalized as

$$\boldsymbol{A} = \arg\min_{\boldsymbol{A}} L(\boldsymbol{A}) \tag{4}$$

with

$$\begin{aligned} L(\boldsymbol{A}) &= E\{(\boldsymbol{t} - \boldsymbol{A}\boldsymbol{s})^T(\boldsymbol{t} - \boldsymbol{A}\boldsymbol{s})\} \\ &= tr(\boldsymbol{\Sigma}_{tt} - 2\boldsymbol{A}^T\boldsymbol{\Sigma}_{ts} + \boldsymbol{A}^T\boldsymbol{\Sigma}_{ss}\boldsymbol{A}), \end{aligned} \tag{5}$$

where $\boldsymbol{\Sigma}_{ss}$,$\boldsymbol{\Sigma}_{tt}$ and $\boldsymbol{\Sigma}_{ts}$ are covariance matrices and L indicates a loss function. $L(\boldsymbol{A})$ converges to the minimum when the derivative of $L(\boldsymbol{A})$ is equal to 0, giving

$$\boldsymbol{A} = \boldsymbol{\Sigma}_{ss}^{-1}\boldsymbol{\Sigma}_{ts} \tag{6}$$

To estimate the number of components, K, and parameters of each mixture model, $\{\boldsymbol{\pi}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$, the EM algorithm is initialized using a hierarchical clustering scheme [19].

Finally, the set of mapping functions $\{\boldsymbol{A}_k, \boldsymbol{B}_k^s, \boldsymbol{B}_k^t, \boldsymbol{w}_k^s, \boldsymbol{w}_k^t\}$ and the pairs of corresponding clusters are obtained. A new corresponding point can be predicted as

$$\boldsymbol{d}_*^t = \sum_{j=1}^{K} \gamma_k({\boldsymbol{B}_k^t}^{-1}\boldsymbol{A}_k\boldsymbol{z}_{k*}^s + \boldsymbol{w}_k^t), \tag{7}$$

where γ_k is the weight of the kth Gaussian component, and $\boldsymbol{z}_{k*}^s = \boldsymbol{B}_k^s(\boldsymbol{d}_*^s - \boldsymbol{w}_k^s)$ from (1).

B. Modeling a motion: Dynamic Movement Primitives

We briefly introduce DMPs to encode a trajectory with a small set of weight parameters. Each degree of freedom (DoF) q of the demonstration is modeled with the following set of non-linear differential equations

$$\begin{aligned} \tau\dot{\upsilon} &= \alpha_\upsilon(\beta_\upsilon(q_{goal} - q) - \upsilon) + g(s) && (8) \\ \tau\dot{q} &= \upsilon && (9) \\ \tau\dot{s} &= -\alpha_s s, && (10) \end{aligned}$$

where α_υ and β_υ are variables related to the responsibility of the system, τ specifies the time scale, υ is a supplemental variable, and α_s is a variable for the canonical system. The function g is a non-linear function responsible for representing the trajectory from the initial position q_0 to the final goal q_{goal}, and often modeled with Local Weighted Regression (LWR). To converge to the query point, α_υ, β_υ, α_s and τ should be carefully chosen. A trajectory is encoded into a DMP as follows. One of its DoFs can be described in terms of its position, velocity and acceleration profile as q_{demo}, $\dot{q}_{demo}$, $\ddot{q}_{demo}$, composed of a set of U points, and its goal position q_{final}. Equations (8) and (9) are combined into one second-order differential equation,

$$g_{target,i} = \tau\ddot{q}_{demo,i} - \alpha_\upsilon(\beta_\upsilon(q_{goal} - q_{demo,i}) - \dot{q}_{demo,i}), \tag{11}$$

where i refers to the ith point of U. Once g is approximated, the DMP can generate a new trajectory following (8) - (10).

IV. System Framework

We now present a detailed discussion of our proposed framework for enabling skill learning from human demonstrations. The full system is illustrated in Fig. 1, and is composed of four steps. 1) A human demonstration is extracted from camera data (section IV-A). 2) These demonstrations are projected onto a human model (section IV-B). 3) A kinematics mapping between the human model and the target robot is prepared offline (section IV-C). 4) Using this mapping, the demonstration is transferred to the robot as a DMP (section IV-D).

A. Human demonstration extraction

The demonstrator first needs to record demonstrations to teach a robot. These are recorded as a set of trajectories, with hand orientations described in the form of rotation matrices.

The requirement for our system was to use a camera with depth information, and as such we used the Microsoft Kinect v2 sensor. This provides standard software for extracting both the joint positions and rotations from the demonstrated human trajectories.

B. Projecting demonstrations onto human model

In order to decompose the human motions into several DoFs joint trajectories and Cartesian positions, we prepared a human skeleton model. This model is based on a Master Motor Map (MMM) [21], which is a commonly used reference kinematic model to describe a human skeleton with 52 DoFs. As a preprocessing step, each corresponding angle is calculated from each pair of joint positions at each timestep. This provides a dataset of joint trajectories and the end-effector Cartesian position trajectories.

Following this, the dataset is projected onto the human model to obtain its corresponding trajectories. We additionally employ LPA (section III-A) in this step, as the prepared human model is likely to differ slightly in its morphology from the true human demonstrator. Using LPA, the differences between the demonstrator and model can be compensated for, retaining the same end-effector trajectory. In particular, when the ratio between corresponding links is the similar, the transform between the source and the target is approximately linear. Note that this means only one reference skeleton model is required, and is scaled to every user by employing LPA.

C. Learn mapping between human model and target robot

To learn a mapping between a human model and a robot, we need to collect data of corresponding poses between the two. For skill transfer, this takes the form of joint space mapping. We collect corresponding points by placing the end-effector of the human model and the robot at the same position in the workspace. Our objective is to learn a set of mapping functions $\boldsymbol{f} = \{f_1, f_2, ..., f_M\}$ using LPA, as shown in Fig. 2. M represents the number of clusters in LPA. Since the robot may have different arm lengths from the demonstrator, joint angles corresponding to the same end-effector position may be different as well.

In our experiments, we make use of a numerical IK solver to collect joint angles of the human model corresponding to each end-effector position of the robot, and we provide the robot joint values as the initial solution to the IK solver. This guides the IK solver to solutions that correspond to the robot's configuration, which is exactly what we need. The

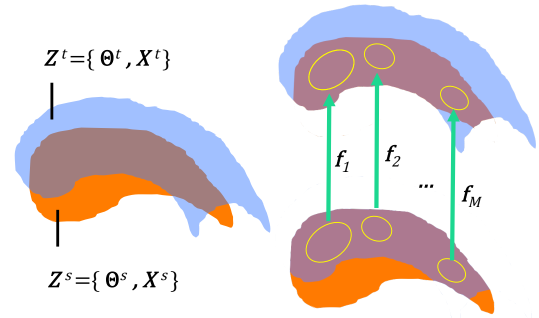


Fig. 2: Left: the relationship of workspaces of two domains. Right: the bottom region in orange represents the human arm workspace as the source domain, and the top region in blue is the workspace of the robot arm – the target domain.

IK solver is then only needed for the human models and not for the target robots.

Finally, LPA learns the mapping between the two datasets, and note that LPA was developed mainly to transfer kinematics data from a large amount of data in the source domain to a small amount of data in the target domain in order to improve learning of kinematic models in the target domain. We consider $\boldsymbol{Z}^s = \{\boldsymbol{\Theta}^s, \boldsymbol{X}^s\}$ as the source dataset, $\boldsymbol{Z}^t = \{\boldsymbol{\Theta}^t, \boldsymbol{X}^t\}$ as the target, where the end-effector positions are given by $\boldsymbol{X}$ and joint coordinates by $\boldsymbol{\Theta}$.

Here, our objective is to approximate the kinematics mapping between the source and the target, in order to map a new trajectory from the source to the target. Once the mapping function is learned, it can be used to project new demonstrations onto the robot online.

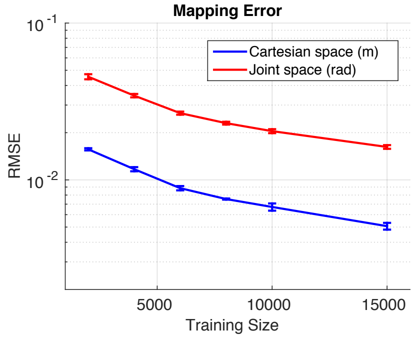
D. Transfer human demonstration onto robot

Through the mappings described above, human demonstrations projected onto the human model are then transferred onto the robot. This is done by representing the joint trajectories as DMPs (section III-B). After the projection, joint trajectories of each DoF are encoded in each DMP such that the set of DMPs corresponds to one motion. This allows for smooth reproductions by the robot, and generalizations of the skills to new start and goal positions, as opposed to replaying the taught motion.

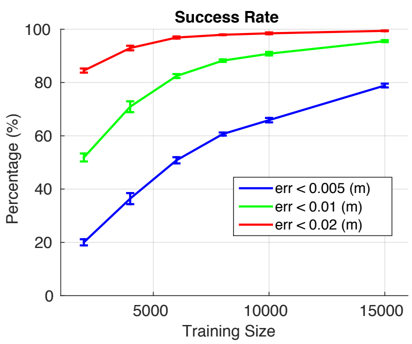
V. Experiments

These experiments involve the transfer of behaviors from a human to a PR2 robot. In particular, we analyzed the mapping accuracy of LPA for projecting human data onto the PR2 robot, and then using the mapping, we demonstrate the skill transfer.

PR2 has longer arms than the human model: 40 cm for the upper arm and 32 cm for the forearm, as opposed to 30 cm and 31 cm on the human model. Both the human model and the PR2 have 7 DoFs for each arm, but for our tasks the first 4 DoFs are sufficient and so we learn the mapping for these DoFs. All results reported in this section are averaged over five runs.

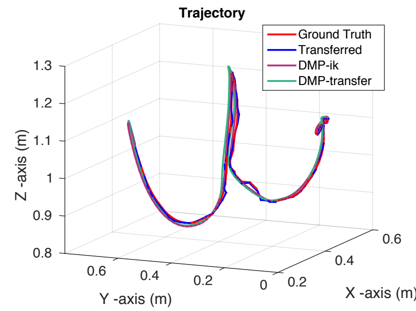


(a) Error in Cartesian and joint space

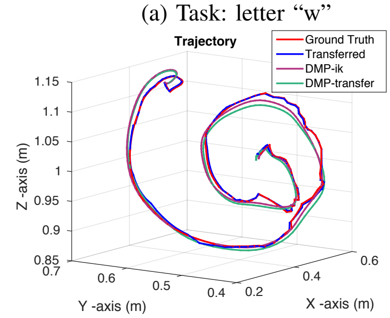


(b) Success rate

Fig. 3: Mapping accuracy



(a) Task: letter "w"



(b) Task: spiral

Fig. 4: Transferred trajectories

A. Mapping accuracy

We first evaluate the mapping accuracy. To compute a mapping between the left arm of the human model and that of the PR2 in simulation, we cluster the data for LPA in joint space and for the initialization step, we optimize the parameters through an offline grid search. The ground-truth dataset for the mapping is a randomly generated set of 5,000 corresponding instances of configurations of the human model and the PR2.

Fig. 3 shows the mapping errors (RMSE) in Cartesian space and joint space, and the success rate in Cartesian space of the mapping as a function of error tolerance. As expected, in general, the accuracy improves as the training size increases. We note that the RMSE in Cartesian space remains at 0.0051 m with 15,000 training points. For the success rate, we set tolerances for desirable error as 0.01 m, and acceptable error as 0.02 m. 95.5 % of the test data was then desirable, and more than 99.4 % acceptable.

The results obtained in this section show that LPA is able to learn an accurate mapping for the first 4 DoFs and that we can expect accurate projections of human demonstrations onto the robot.

B. Trajectory mapping and modeling with DMP

We next demonstrate the trajectory mapping performance of the proposed system. In this experiment, two tasks were performed by a human: writing the letter "w" and drawing a spiral. Fig. 5 shows a series of frames with a human teacher demonstrating writing the letter "w" and the PR2 executing it using our system. The mapping was trained using 15,000 points. Each trajectory is projected onto the human model, and then mapped onto the PR2 and encoded as a DMP. In Fig. 4, the red trajectories are the human demonstrations from the human model, the blue trajectories were transferred to the PR2, and the cyan trajectories were modeled by DMP from transferred trajectories, and reproduced when provided with the same start and goal positions. To evaluate our method, we also use inverse kinematics (unknown to our method) to track human demonstrations with the PR2 and encode the joint-space trajectories as a DMP. These comparisons to IK trajectories are shown in purple.

Trajectories for both motions were accurately transferred, and their RMSE were less than 0.01 m: 0.0028 m for the spiral, and 0.005 m for the "w". RMSE between DMP-ik and DMP-transfer trajectories were also small: 0.0066 m for the spiral, and 0.0055 m for the "w". The results show that our method performs as well as the method relying on the kinematics model of the robot.

As the ground truth trajectories show, the Kinect sensor data contains non-negligible noise and trajectories are not smooth. This can be eliminated by smoothing or optimizing the DMPs through adding more weights [22].

VI. Conclusion

In this work, we proposed a novel system to project human demonstrations onto a robot. By combining LPA and DMPs, it was shown experimentally that our proposed method can imitate human demonstrations and reproduce them with minimal errors. Notably, this system only requires a single demonstration in order to teach a motion to a robot with a different kinematic structure, without relying on the kinematics model of the robot. This is desirable to use in

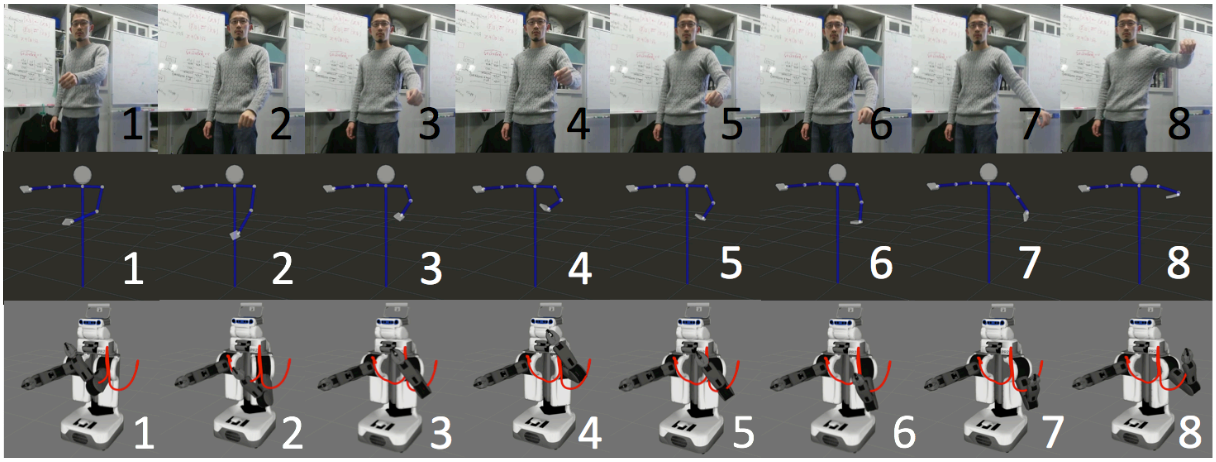


Fig. 5: Demonstration and imitation, writing a letter "w"

real-world applications where an accurate kinematics model may not be available.

In future work, we aim to combine our approach with reinforcement learning, or other optimization techniques, to refine the taught motions considering the limitations of the robot and/or obstacles in the environment and allow for more general reproductions. Furthermore, this data-driven approach can be improved with the combination of analytical models [23,24]. The limitation of our current system is the assumption by LPA that the manifolds have the same dimensionality. Although one can manually select corresponding dimensions (i.e., robot DoFs) between a human and robot for a particular task, our future work will examine inferring this automatically from data.

References

[1] R. Dillmann, Teaching and learning of robot tasks via observation of human performance, Robotics and Autonomous Systems, 47 (2-3), pp. 109-116, 2004.

[2] N. Makondo, J. Claassens, N. Tlale and M. Braae, Geometric technique for the kinematic modeling of a 5 DOF redundant manipulator, in Robotics and Mechatronics Conference (ROBMECH), pp. 1-7, 2012.

[3] J. Sturm, C. Plagemann and W. Burgard, Body schema learning for robotic manipulators from visual self-perception, Journal of Physiology - Paris, 103(3-5), pp. 220-231, 2009.

[4] B. Argall, S. Chernova, M. Veloso and B. Browning, A survey of robot learning from demonstration, Robotics and autonomous systems 57.5, pp. 469-483, 2009.

[5] B. Akgun, M. Cakmak, J. Yoo and A. Thomaz, Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective, in Proc. of the seventh annual ACM/IEEE Int. Conf. on Human-Robot Interaction. ACM, 2012.

[6] J. Koenemann, F. Burget, and M. Bennewitz, Real-time Imitation of Human Whole-Body Motions by Humanoids, in IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 2806-2812, 2014.

[7] C. Stanton, A. Bogdanovych and E. Ratanasena, Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning, in Proc. of Australasian Conf. on Robotics and Automation (ACRA), 2012.

[8] A. Faudzi, L. Chuan and T. Kit, Task Learning Utilizing Curve Fitting Method for Kinect Based Humanoid Robot, in Proc. of Int. Conf. on Robotics and Biomimetics (ROBIO), pp. 1505-1511, 2014.

[9] R. Vuga, M. Ogrinc, A. Gams, T. Petric, N. Sugimoto,A. Ude and J. Morimoto, Motion Capture and Reinforcement Learning of Dynamically Stable Humanoid Movement Primitives, in IEEE Int. Conf. on Robotics and Automation (ICRA), pp.5284-5190, 2013.

[10] P. Ranchod, B. Rosman and G. Konidaris, Nonparametric Bayesian Reward Segmentation for Skill Discovery Using Inverse Reinforcement Learning, in IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), pp. 471-477, 2015.

[11] S. Calinon, F. D'halluin, E. Sauser, D. Caldwell and A. Billard, Learning and reproduction of gestures by imitation: An approach based on Hidden Markov Model and Gaussian Mixture Regression, IEEE Robotics and Automation Magazine, 17:2, pp. 44-54, 2010.

[12] S. 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, 2011.

[13] A. Ijspeert, J. Nakanishi and S. Schaal, Learning rhythmic movements by demonstration using nonlinear oscillators, in IEEE Int. Conf. on Intelligent Robots and Systems (IROS), pp. 958-963, 2002.

[14] S. Schaal, P. Mohajerian, and A. Ijspeert, Dynamics systems vs. optimal control-a unifying view, Progress in Brain Research 165 (6), pp.425-445, 2007.

[15] D. Forte, A. Gams, J. Morimoto, A. Ude, On-line motion synthesis and adaptation using a trajectory database, Robotics and Autonomous Systems, vol. 60, pp. 1327-1339, 2012.

[16] A. Jha, S. Chiddarwar and M. Andulkar, An integrated approach for robot training using Kinect and human arm kinematics, Advances in Computing, Communications and Informatics (ICACCI), pp. 216-221, 2015.

[17] G. Maeda, M. Ewerton, D. Koert and J. Peters, Acquiring and Generalizing the Embodiment Mapping from Human Observations to Robot Skills, IEEE Robotics and Automation Letters, 2016.

[18] A. Shon, K. Grochow and R. Rao, Robotic imitation from human motion capture using Gaussian processes, in IEEE-RAS Int. Conf. on Humanoid Robots, pp. 129-134, 2005.

[19] N. Makondo, B. Rosman and O. Hasegawa, Knowledge transfer for learning robot models via local procrustes analysis, in IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids), pp. 1075-1082, 2015.

[20] B. Bocsi, L. Csato, and J. Peters, Alignment-based Transfer Learning for Robot Models, Proc. of the IEEE Int. Joint Conf. on Neural Networks (IJCNN), pp. 1-7, 2013.

[21] P. Azad, T. Asfour, and R. Dillmann, Toward an unified representation for imitation of human motion on humanoids, in IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 2558-2563, 2007.

[22] J. Rosado, F. Silva, V. Santos and Z. Lu, Reproduction of Human Arm Movements Using Kinect-Based Motion Capture Data, in Proc. of the IEEE Int. Conf. on Robotics and Biomimetics (ROBIO), pp. 885-890, 2013.

[23] D. Nguyen-Tuong and J. Peters, Using model knowledge for learning inverse dynamics, in IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 2677-2682, 2010.

[24] K. Chen and J. Yi, On the Relationship between Manifold Learning Latent Dynamics and Zero Dynamics for Human Bipedal Walking, in IEEE Int. Conf. on Intelligent Robots and Systems (IROS), pp. 971-976, 2015.