

Learning from Demonstration: Generalization via Task Segmentation

N Ettehad¹, S Manaffam¹ and A Behal¹

¹Authors are with ECE Department and NanoScience Technology Center, University of Central Florida (UCF), Orlando, FL 32816, USA

E-mail: nabil.ettehadi@knights.ucf.edu

Abstract. In this paper, a motion segmentation algorithm design is presented with the goal of segmenting a learned trajectory from demonstration such that each segment is locally maximally different from its neighbors. This segmentation is then exploited to appropriately scale (dilate/squeeze and/or rotate) a nominal trajectory learned from a few demonstrations on a fixed experimental setup such that it is applicable to different experimental settings without expanding the dataset and/or retraining the robot. The algorithm is computationally efficient in the sense that it allows facile transition between different environments. Experimental results using the Baxter robotic platform showcase the ability of the algorithm to accurately transfer a feeding task.

1. Introduction

One of the key challenges of learning from demonstration is to generalize a learned task to different situations and environments. This generalization to a new environment should be done such that the key features and constraints of the task (*i.e.*, key motions required to accomplish the task) be satisfied resulting in the successful implementation of the task in the new environment. In [1], a programming-by-demonstration framework is presented in order to generically extract a trajectory having the relevant features of a given task. The authors in [1] use Gaussian Mixture Model/Regression (GMM/GMR) to model the task and generate new trajectories. The GMM/GMR approach has been also used in [2] for manipulation tasks and is robust to the movement of obstacles included in the demonstrations. A number of methods have previously attempted to generalize a learned trajectory to different environments. The GMM/GMR is modified in [3] to adapt the means and covariances of a GMM to different locations and orientations of objects involved in a task. In [4], [5], and [6] the authors present a demonstration-guided motion planning (DGMP) to generate plans that avoid obstacles in a new environment while maintaining the critical features of the task. Another work has attempted to find sub-optimal controllers which reproduce a set of constraints of a given task [7]. However, the constraints are selected from a set of predefined constraints, *e.g.*, certain orientations for the end-effector. In [8], the authors use an invariant trajectory representation to separate essential motion information from context-specific information of the recorded demonstrations. Another work [9] uses dynamic movement primitives that are learned from human demonstrations with generalization to different start and goal points for a task.

In this paper, we propose an alternative approach to generalize a learned trajectory of a given task to different environments. In majority of the approaches discussed above, in order to learn and execute a task, a relatively big dataset, spanning the space in which the task would be performed, is needed.



However, our method does not require a big dataset; instead, a small number of demonstrations is needed to collect data with a robot for a fixed setup of the environment (no change in the position/orientation of the objects involved in the task). Consequently, the amount of data needed for training greatly reduces. Our approach takes the learned trajectory of the task for a fixed environment as an input and outputs an appropriate trajectory which successfully implements the learned task in a different environment. The approach consists of two parts: 1) segmentation of the learned trajectory w.r.t different motions that take place in each part of the trajectory, followed by 2) proper rotation and scaling of each segment based on the changes between the new environment and the environment in which the robot has been trained. The learned trajectory (input) is derived using the approach in [1]. Unlike neural network based methods, whose application is limited only to interpolation between the space over which they have been trained, our method is capable of generalization of the task to different environments even outside the training environment without the need for retraining the robot. Our approach only scales and rotates different segments of the trajectory, therefore it has less computational complexity than the approaches in [4]–[6]. Our method has been validated by running a task of feeding cereal from a bowl to a user.

2. Preliminaries

2.1. Notations and Definitions

The set of real n -vectors is denoted by \mathbb{R}^n and the set of real $m \times n$ matrices is denoted by $\mathbb{R}^{m \times n}$. Matrices and vectors are denoted by capital and lower-case bold letters, respectively. The set of natural numbers is referred to by \mathbb{N} . Uppercase letters are used for constants and lowercase letters for scalar variables. The end-effector's trajectory is denoted by $\mathbf{X}(t) = [\mathbf{X}_{pos}(t); \mathbf{X}_{ori}(t)] \in \mathbb{R}^{6 \times N}$ where $\mathbf{X}_{pos}(t) = [\mathbf{x}(t); \mathbf{y}(t); \mathbf{z}(t)] \in \mathbb{R}^{3 \times N}$ is a matrix containing 3 time-dependent vectors for Cartesian coordinates (positions), $\mathbf{X}_{ori}(t) = [\theta_x(t); \theta_y(t); \theta_z(t)] \in \mathbb{R}^{3 \times N}$ is a matrix containing 3 time-dependent vectors for the orientation of the end-effector (e.g., Euler angles), and N is the length of the trajectory (number of time samples). $\mathbf{R}_{\beta, \mathbf{a}} = [\mathbf{c}_1 \ \mathbf{c}_2 \ \mathbf{c}_3] \in \mathbb{R}^{3 \times 3}$ represents the rotation matrix around an arbitrary axis, \mathbf{a} , by an angle, β , where $\mathbf{c}_k \in \mathbb{R}^{3 \times 1}$ represents the k^{th} column vector of the rotation matrix $\mathbf{R}_{\beta, \mathbf{a}}$.

2.1.1. Definition (Landmark) Any entity relevant to the task influencing the behavior of the robot while the task is being performed. It is assumed that the shape of all landmarks do not change over the task. Landmark i is denoted by l_i .

2.1.2. Definition (Nominal Setup for Landmarks) Given a certain task, we define, as nominal, a known fixed setup in which the robot has been trained to perform that task. Specifically, the nominal setup, denoted by s_{nom} , is comprised of the fixed positions and/or orientations of all landmarks related to the task.

2.1.3. Definition (Nominal Trajectory) A given task is modelled as a Gaussian Mixture Model (GMM) using data collected from demonstrations (via kinesthetic teaching) performing the task in s_{nom} . The nominal trajectory of a given task, denoted by $\mathbf{X}_{nom}(t) = [\mathbf{X}_{nom_{pos}}(t); \mathbf{X}_{nom_{ori}}(t)] \in \mathbb{R}^{6 \times N}$, is the estimated mean of the GMM, containing all of the essential relevant features (key motions) of the task.

2.2. Motivation and Problem Definition

One can obtain a nominal trajectory for a given task by using the methodology described in [1]. One interesting question is how to generalize the nominal trajectory to a new setup different from the nominal setup (outside the training set) without expanding the training set for retraining the robot. While most Machine Learning/Neural Networks techniques are capable of interpolation, they are not able to produce a proper trajectory for the sets outside the subspace spanned by the training samples [10]. Hence, we need other solutions to the following problem: *Let $\mathbf{X}_{nom}(t) \in \mathbb{R}^{6 \times N}$ be the nominal trajectory of a given task, T , performed in the nominal setup, s_{nom} . Given a new setup s with the same landmarks, find*

$\mathbf{X}_{new}(t) \in \mathbb{R}^{6 \times N}$ performing T in s , using $\mathbf{X}_{nom}(t)$, s_{nom} and s , without retraining the robot?

Please note that in the problem above, there are no assumptions imposed on the distances and the positions between the landmarks in the new setup s compared to s_{nom} . Let us consider a trajectory $\mathbf{X}_{nom}(t)$ of an arbitrary given task T , with n landmarks in the nominal setup s_{nom} . Now consider a new setup s . At first glance, the solution to the above problem is to rotate and scale each part of $\mathbf{X}_{nom}(t)$ lying between two consecutive landmarks l_i and l_{i+1} , according to their rotation and scaling between s_{nom} and s . However, by using this method the key motions of $\mathbf{X}_{nom}(t)$ would not be preserved. For example, consider a feeding task with two landmarks: a bowl full of cereal and a human user (the spoon is strictly attached to the end-effector). Scooping cereal is a key motion having a different nature compared to other key motions of this task (e.g., keeping the loaded spoon level). Using the technique in [1], we can find $\mathbf{X}_{nom}(t)$ of this task in s_{nom} . Now, let us move the bowl away from the user defining the new setup s . As discussed above, by uniform stretching of the entire trajectory between the only existing landmarks, the robot would continue scooping even outside of the bowl since the corresponding segment of the trajectory within the bowl is also stretched; this would result in the end-effector tipping the bowl over. Hence, uniform scaling and/or rotation of a nominal trajectory (in general) does not guarantee the successful implementation of a given task. To avoid this problem, a segmentation technique is needed to divide the nominal trajectory of a given task into different segments. This segmentation technique should distinguish the different key motions taking place at each part of the task and segment the nominal trajectory accordingly. By having such segments, the nominal trajectory can be scaled and rotated intelligently (i.e., segment by segment) during execution according to the differences between the two setups s and s_{nom} . Hence, we can state our problem definition as follows: Let $\mathbf{X}_{nom}(t) = [\mathbf{X}_{nom_{pos}}(t); \mathbf{X}_{nom_{ori}}(t)] \in \mathbb{R}^{6 \times N}$ be the nominal trajectory of a certain task, T , in the nominal setup, s_{nom} .

A. Find all the segments of $\mathbf{X}_{nom}(t)$ corresponding to different key motions of the task T .

B. Now let the number of the segments from problem 1 be M . Given a new setup s with the same landmarks in different positions and/or orientations, find the set $\mathcal{M} = \{\mathbf{R}_{\theta_i, \mathbf{a}_i}, \alpha_{c,i}, \forall i \in \{1, \dots, M\}$ and $\forall c \in \{x, y, z\}\}$.

3. Problem A: Segmentation of the Nominal Trajectory

In the previous section, we divided the problem into two parts. The first part is intended to segment the nominal trajectory corresponding to different key motions of the given task. A motion is constructed by a set of velocity commands (both linear and angular). At each segment, certain velocities, having specific directions and amplitudes, are commanded to perform the appropriate motion required for that part of the task. Thus, for the purpose of the segmentation, we base our analysis on the nominal velocity trajectory $\Lambda(t) \in \mathbb{R}^{6 \times N}$. In lieu of a binary representation as motion/no-motion, we embed the motion in a higher dimensional space by representing $\Lambda(t)$ as a quantized level-valued signal. Next, we discuss quantization and segmentation.

3.1. Quantization

Nominal velocity trajectory $\Lambda(t) \in \mathbb{R}^{6 \times N}$ is quantized by the following approach: we uniformly quantize linear and angular velocity components into q_v and q_ω levels, respectively. The length of two consecutive levels in our quantization are denoted by δ_v and δ_ω for linear and angular velocity, respectively. Thus, the linear velocity signals belonging to interval $(-\delta_v, +\delta_v)$ and angular velocity signals belonging to $(-\delta_\omega, +\delta_\omega)$, are mapped to level 0 denoting no motion. The choice of quantization steps can be formulated as a detection problem where δ_v and δ_ω are calculated for a chosen small probability of false alarm¹. For more details regarding this subject, the readers are referred to literature in detection theory [11]. The number of sets q_v and q_ω are derived as follows: $q_v = \max\{v_x(i), v_y(i), v_z(i)\}/\delta_v$, $q_\omega = \max\{\omega_x(i), \omega_y(i), \omega_z(i)\}/\delta_\omega$. Each quantization level is

¹ The probability that there is no motion taking place in reality but a motion is falsely detected.

labeled with d_{v_x} , d_{v_y} , and d_{v_z} corresponding to v_x , v_y and v_z , respectively (e.g., $d_{v_x}(i) = 2$ means that v_x at time stamp i falls into the second level of our quantization). Similarly, we use d_{ω_x} , d_{ω_y} , and d_{ω_z} for the quantizations level labels of angular velocities. Now, we have a quantized nominal velocity trajectory of the end-effector denoted by $\mathbf{D}_{nom} = [\mathbf{d}_{nom}(i)]_{6 \times N}$ for $i = 1, 2, \dots, N$ where $\mathbf{d}_{nom}(i) = [d_{v_x}(i); d_{v_y}(i); d_{v_z}(i); d_{\omega_x}(i); d_{\omega_y}(i); d_{\omega_z}(i)] \in \mathbb{R}^6$ is the quantized nominal velocity vector of the end-effector at each time stamp i . It is important to mention that the quantization levels take into account both magnitude and sign of the velocities. This is required because two motions in opposite directions with the same magnitude should be distinguished from each other.

3.2. Segmentation Algorithm

\mathbf{D}_{nom} is segmented by the following approach: we first uniformly divide \mathbf{D}_{nom} into r mutually-exclusive, equally-lengthed units in time called search-segments: $\mathbf{D}_{nom} = \{\mathbf{D}_{nom_1}, \mathbf{D}_{nom_2}, \dots, \mathbf{D}_{nom_r}\}, \mathbf{D}_{nom_i} \cap \mathbf{D}_{nom_j} = \emptyset, \bigcup_{i=1}^r \mathbf{D}_{nom_i} = \mathbf{D}_{nom}, \forall i \neq j \in \{1, 2, \dots, r\}$.

The number r is a parameter, chosen such that it is at least the size of the smallest acceptable task segment but significantly smaller than data size N ; one good choice would be: $r \approx \frac{N}{10}$. The center of each search-segment is defined as the average of its members, i.e., if search-segment $\mathbf{D}_{nom_i} \in \mathbb{R}^{6 \times \lceil \frac{N}{r} \rceil}$ starts at time stamp f_i and ends at time stamp $f_i + \lceil \frac{N}{r} \rceil$, its center is calculated as: $\bar{\mathbf{d}}_{nom_i} = \frac{1}{\lceil \frac{N}{r} \rceil} \sum_{j=f_i}^{f_i + \lceil \frac{N}{r} \rceil} \mathbf{d}_{nom_i}(j) / (\lceil \frac{N}{r} \rceil + 1)$. As previously stated, we aim to merge these search-segments to form the

larger segments representing the different key motions taking place in different parts of the nominal trajectory. In order to do that, a measure of similarity between two consecutive search-segments should be defined to decide if two search-segments need to be merged or not. This measure is defined as the absolute value of the difference between the center of search-segment i and $i+1$: $\eta(i) = |\bar{\mathbf{d}}_{nom_{i+1}} - \bar{\mathbf{d}}_{nom_i}|$, $\forall i = 1, 2, \dots, r-1$. To reduce the effect of noise, for any two consecutive search-segments $i-1$ and i , the mean of the defined measure of time stamp $i-1$, i.e., $\bar{\eta}(i-1)$, is compared with $\eta(i)$: if $\bar{\eta}(i-1) \leq \eta(i)$, it means there exists more similarities between the corresponding velocities in search-segment i and the mean of the members of the very last segment of this step, compared to the similarities between the corresponding velocities in search-segment i and $i+1$. Thus, search-segment i merges with the very last segment of this step; otherwise, the very last segment of this step is closed as a complete segment and the search for a new segment initializes with search-segment i being the first member of it. The following pseudo code provides the complete algorithm which returns *seg_final* as the output:

```

Step 1.  $\eta[r] \leftarrow \infty, k \leftarrow 1, \mathbf{D}_{nom_1} \leftarrow seg\_temp, \bar{\eta}[1] \leftarrow \eta[1]$  //initialization
Step 2. For  $i \leftarrow 2$  to  $r$  do
    Step 2.1. If  $\eta[i] \geq \bar{\eta}[i-1]$ 
        Step 2.1.1.  $seg\_temp \leftarrow [seg\_temp, \mathbf{D}_{nom_i}[:, f_i : f_i + \lceil \frac{N}{r} \rceil]]$ 
        Step 2.1.2.  $\bar{\eta}[i] \leftarrow \frac{(i-1) \times \bar{\eta}[i-1] + \eta[i]}{i}$ 
    Step 2.2. else
        Step 2.2.1.  $seg\_final[k] \leftarrow seg\_temp$  //close seg_temp as a completed segment.
        Step 2.2.2  $k \leftarrow k + 1$ 
        Step 2.2.3.  $seg\_temp \leftarrow [\mathbf{D}_{nom_i}[:, f_i : f_i + \lceil \frac{N}{r} \rceil]]$ 
        Step 2.2.4.  $\bar{\eta}[i] \leftarrow \eta[i]$ 

```

4. Problem B: Off-line Adaptation of The Nominal Trajectory to a New Setup for Landmarks

Now, by having the segmented nominal trajectory, we can properly scale and rotate each segment based on the differences between s and s_{nom} . Since the nominal trajectory needs to be rotated, here we use the rotation matrix representation for $\mathbf{X}_{nom_{ori}}(t)$ rather than Euler angle representation, i.e., at each time stamp t we have $\hat{\mathbf{X}}_{nom_{ori}}(t) = [\mathbf{c}_1(t) \ \mathbf{c}_2(t) \ \mathbf{c}_3(t)] \in \mathbb{R}^{3 \times 3}$ denoting the rotation matrix for the orientation

of the end-effector. Now, suppose a segment k starts at time index j and finishes at time index $j + \lceil N/r \rceil$. The new segment in $\mathbf{X}_{new}(t)$ is derived from the corresponding segment in $\tilde{\mathbf{X}}_{nom}$ as follows:

$$\begin{aligned}\mathbf{X}_{new_{pos}}(t) &= \mathbf{X}_{new_{pos}}(t-1) + (\mathbf{A}_k \times \mathbf{R}_{\theta_k, \mathbf{a}_k} \times [\mathbf{X}_{nom_{pos}}(t) - \mathbf{X}_{nom_{pos}}(t-1)]) \\ \hat{\mathbf{X}}_{new_{ori}}(t) &= \mathbf{R}_{\theta_k, \mathbf{a}_k} \times \hat{\mathbf{X}}_{nom_{ori}}(t) \text{ for } t = j, j+1, j+2, \dots, j + \lceil N/r \rceil - 1\end{aligned}\quad (1)$$

where $\mathbf{A}_k = \text{diag}(\alpha_{x,k}, \alpha_{y,k}, \alpha_{z,k})$ is the scale factor matrix calculated using the corresponding scale factors in each Cartesian components and t is the time index. The initial condition for this equation, *i.e.*, $\mathbf{X}_{new}(j-1)$ is equal to the last point of the previous segment $k-1$, which has been previously properly rotated and scaled using (1). Note that since the shape of landmarks are assumed to be constant and both positions and orientations are tied with time stamps, it is sufficient to only scale the positions. By proper rotation and scaling of all segments, the characteristics of the motions in different segments (key motions taking place at different segments) will remain intact. Finally, by back projecting $\hat{\mathbf{X}}_{new_{ori}}(t)$ to Euler angle representation, *i.e.*, $\hat{\mathbf{X}}_{new_{ori}}(t) \rightarrow \mathbf{X}_{new_{ori}}(t)$, the appropriate trajectory of the task for setup s is achieved as: $\mathbf{X}_{new}(t) = [\mathbf{X}_{new_{pos}}(t); \mathbf{X}_{new_{ori}}(t)] \in \mathbb{R}^{6 \times N}$.

5. Experimental Results

In order to demonstrate the validity of our method, we have chosen the task of feeding cereal to a human user. There are two landmarks involved in the task: landmark l_1 is a bowl full of cereal and l_2 is the mouth of a user which is considered to be a point in the space². The nominal setup is $s_{nom} = \{x_{nom_b} = 0.806, y_{nom_b} = 0.811, z_{nom_b} = -0.070, x_{nom_m} = 0.858, y_{nom_m} = 0.489, z_{nom_m} = -0.001\}$ meter w.r.t. the base frame of the robot. figure 1(a) shows the nominal setup for the task. We use a 7-DOF Baxter robot in order to conduct our experiment. An arbitrary coordinate frame is set to be placed at the center of the bowl and the positions and orientations of the spoon are being recorded w.r.t this coordinate frame. 11 demonstrations of the task are recorded by a human teacher performing the task in the fixed nominal setup s_{nom} with the robot using kinesthetic teaching. Using [1] the nominal trajectory is derived from the demonstrations. Next, we run our segmentation algorithm on $\mathbf{X}_{nom}(t)$ consisting of 870 samples. We test with two different number of initial search-segments, $r_1 = 28$ and $r_1 = 68$ which set the length of each search-segment to be $\lceil \frac{870}{28} \rceil = 32$, and $\lceil \frac{870}{68} \rceil = 13$, respectively. The algorithm results in 6 segments for $r_1 = 28$ and 7 segments for $r_2 = 68$ as shown in figures 1. As illustrated in figure 1(b), the algorithm is able to successfully capture total number of 6 segments; 3 segments for the bowl, namely, approaching the bowl (blue segment), scooping (red segment), and leaving the bowl (green segment), 1 segment for the translation (yellow) of the spoon from the bowl to mouth of the user, another segment for the reversed translation (black), and finally the last segment (light blue) as the resetting of the spoon in preparation for the next feeding cycle. For the case where $r_2 = 68$, we get 7 segments; however, the additional segment shown in figure 1(c) is a very small blue segment at the beginning of the trajectory likely resulting from the fact that increase in the number of search-segments may lead to artifacts related to reduction in window size. Nonetheless, the result that both $r_1 = 28$ and $r_2 = 68$ yield almost identical results for segmentation, shows that the algorithm is robust to the initial number of search-segments. Next, we discuss the results for experiments using the 3 different setups: $s_1 = \{0.457, 1.007, -0.07, 0.849, 0.704, -0.001\}$, $s_2 = \{0.564, 0.941, -0.07, 0.931, 0.773, -0.001\}$ and $s_3 = \{0.608, 0.896, -0.07, 0.761, 0.704, -0.001\}$ meter. Since the size of the bowl has not changed, as it was assumed in the definition of landmarks, only the parts of nominal trajectory corresponding to the first and the second translation phases (as described above) are scaled by their corresponding scale factor matrices \mathbf{A}_1 and \mathbf{A}_2 , respectively. Moreover, the whole trajectory is rotated using $\mathbf{R}_{\theta, \mathbf{a}}$ (see equation (1)). For s_1 we have: $\mathbf{A}_1 = \text{diag}[2.32, 2.29, 1]$, $\mathbf{A}_2 = \text{diag}[2.32, 2.29, 1]$ and $\mathbf{R}_{43.15^\circ, \mathbf{z}}$. For s_2 we have: $\mathbf{A}_1 = \text{diag}[1.6, 1.6, 1]$, $\mathbf{A}_2 = \text{diag}[1.23, 1.18, 1]$ and $\mathbf{R}_{56.26^\circ, \mathbf{z}}$. For s_3 we have: $\mathbf{A}_1 = \text{diag}[1.6, 1.6, 1]$,

² There is no specific action taken by the robot/user at the position of the mouth; only that the robot stops moving for an amount of time.

$A_2 = \text{diag}[0.75, 0.78, 1]$ and $R_{29.40^\circ, z}$. We successfully tested our algorithm on these 3 setups. The new trajectories for s_1 , s_2 , and s_3 are shown in figure 2. For better comparison, all the trajectories are plotted in their new bowl frames. As it can be seen, the segments corresponding to the bowl have remained intact while the other segments have been scaled properly to implement the task successfully for each setup.³

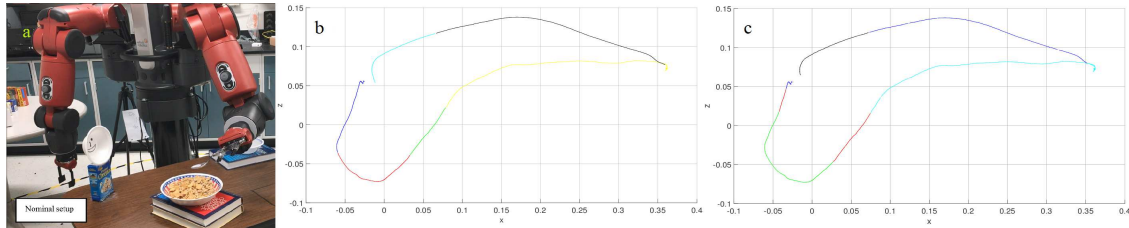


Figure 1. (a) Nominal setup for the task of feeding in our experiment. Projection of the segmented trajectory on xz plane with total number of search-segments (b) $r_1 = 28$ and (c) $r_2 = 68$

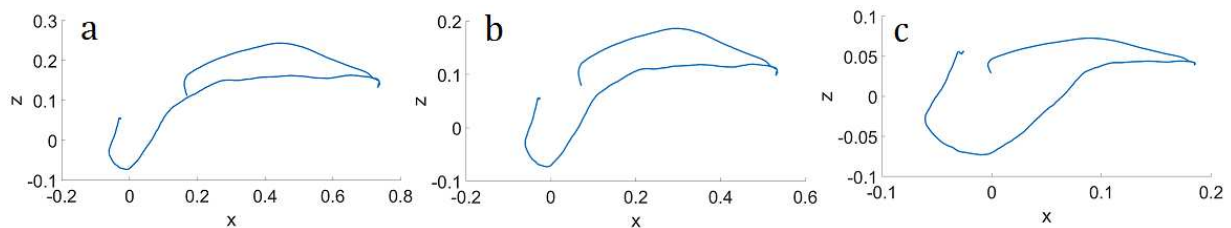


Figure 2. Scaled trajectories for setup (a) s_1 , (b) s_2 and (c) s_3 .

6. Conclusions

In this paper, we presented a method (a segmentation algorithm along with an off-line adaptation law) to generalize a learned nominal trajectory of a given task to different setups without expanding the training set to retrain the robot. The method was successfully validated on 3 different setups for a feeding task.

Acknowledgments:

This study was funded in part by NIDILRR grant # H133G120275, and in part by NSF grant numbers IIS-1409823 and IIS-1527794. However, these contents do not necessarily represent the policy of the aforementioned funding agencies, and you should not assume endorsement by the Federal Government.

References

- [1] Calinon S, Guenter F and Billard A, “On learning, representing, and generalizing a task in a humanoid robot” April 2007. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 286-298, .
- [2] Calinon S, et al. “Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework” 2009. *Humanoid Robots, Humanoids 9th IEEE-RAS International Conference on* pp. 582-588.
- [3] Calinon S, Alizadeh T and Caldwell G D. “On improving the extrapolation capability of task-parameterized movement models” 2013. *IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 610-616.
- [4] Ye G and Alterovitz R, “Demonstration-guided motion planning” 2011. *International Symposium on Robotics Research*. vol. 5.

³ The full video of the experiment is available at: <https://youtu.be/3ecCBCUitqI>

- [5] Bowen C and Alterovitz R, “Closed-loop global motion planning for reactive execution of learned tasks,” 2014. *IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL*, pp. 1754-1760.
- [6] Bowen C, Ye G and Alterovitz R. “Asymptotically optimal motion planning for learned tasks using time-dependent cost maps” 2015. *IEEE Transactions on Automation Science and Engineering* *12.1*, pp. 171-182.
- [7] Alissandrakis A, et al. “An approach for programming robots by demonstration: Generalization across different initial configurations of manipulated objects” 2005. *International Symposium on Computational Intelligence in Robotics and Automation, IEEE*, pp. 61-66 .
- [8] Vochten M, De Laet T and De Schutter J. “Generalizing demonstrated motions and adaptive motion generation using an invariant rigid body trajectory representation” 2016. *IEEE International Conference on Robotics and Automation (ICRA), Stockholm*, pp. 234-241.
- [9] Paxton C, Hager G D and Bascetta L. “An incremental approach to learning generalizable robot tasks from human demonstration” 2015. *Robotics and Automation (ICRA), IEEE International Conference on*, pp. 5616-21
- [10] Bishop C M. *Pattern Recognition and Machine Learning (Information Science and Statistics)* 2006. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- [11] Wilmshurst T H. *Signal Recovery from Noise in Electronic Instrumentation*, Second Edition 1990. Taylor & Francis, ISBN: 978-0-7503-0058-2.