Learning and generalizing tasks on humanoid robots with an automatic multisensory segmentation method

Victor Barberteguy[1,2] and Fumio Kanehiro[2]

***Abstract*— We provide a complete framework for learning and reproducing tasks from human demonstrations. This framework adapts recent developments in automatic, unsupervised segmentation of time-series to humanoid robotics by preprocessing the data obtained from a broad range of the robot's sensors, to then repropduce the learned task in similar environments.**

In more detail, we reproduce and extend the acquired multi-step task using Dynamic Movement Primitives in simulation for the JVRC1 Robot, and further validate it in the real world with the HRP-4C Robot, thus showcasing the capacity of our approach to create an extensive library of reusable skills for complex humanoids.

I. Introduction

A. Learning from Demonstration

Demonstrations provide an intuitive way for non-professional users to specify tasks to the robot [1]. Most of robot's controllers to date, albeit successful to achieve complex tasks, often require theoretical knowledge and is time-consuming. To address these issues, research about robot learning from demonstration (LfD) has become increasingly important over the past two decades with the surge of Machine Learning (ML) algorithms [2], [3]. In this framework, users can demonstrate a task that is then learned and generalized by the robot. Generalization is defined as the capacity to perform a learned task in a similar environment as the one of the demonstration, but under different initial or goal poses.

Several techniques are found in the literature to provide demonstrations to the robot. This can be done through simulation [4], kinesthesic teaching [5], motion capture [6] or with teleoperation devices.

In all the above cited works, some prefer 1-shot imitation learning, where the demonstration can be a seed for an initial policy that is then derived and learned through Reinforcement Learning (RL) [7], [8], Inverse RL [9] or Hierarchical RL [10] that can possibly be corrected with online coaching [11]. On the contrary, other methods rely on a set of demonstrations to perform probabilistic inference based on Hidden Markov Models [12], Neural Networks [13] or using Granger Causality [14]. The former often derives only one policy that is locally optima, and thus the learning process is repeated for each task, whereas the latter methods prefer to decompose the task in elementary *skills* with segmentation methods, thus enhancing the reusability of the stored skills. Our framework is of the latter category, that is further detailed in I-B. Whatever the method used, once the task learned, the great majority of the references cited above use Dynamic Movement Primitives (DMP) [15], [16]. It provides a suitable framework to reproduce trajectories and to smoothly adapt to new conditions.

[1]Ecole Polytechnique, Palaiseau, France
[2]CNRS-AIST JRL (Joint Robotics Laboratory), IRL, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan

Fig. 1: Picture of the experimental setup. On the right, the demonstrator wearing the teleoperation device. On the left, the HRP-4C robot that is teleoperated.

B. Segmentation of the tasks

Splitting a complex task such as grasping an object or pouring water has the advantage of making the result more generalizable. Indeed, every part of the trajectory (every *skill*) has its own level of generalizability thanks to DMP, thus resulting in an intra-task generalization, better than a single end-to-end DMP. Nonetheless, a complex task is almost all the time demonstrated without specifying the number of segments: the demonstrator naturally achieves the task. Consequently, one challenge of this approach is to automatically segment the trajectory in human-interpretable skills that, once stored in a library, can be reused to learn new skills [17].

Recent methods provide plans to the robot about the different phases of the demonstration using natural language or computer vision [18] [19]. Besides, probabilistic approaches are employed to segment a demonstration in a unsupervised fashion. The number of phases involved as well as the temporal position of changepoint (CP) are inferred automatically. These techniques include Hidden Markov Models (HMM) [20], Gaussian Mixture Regression (GMR) [21] [22] and more seldom reinforcement learning techniques [23]. Throughout the upcited references, the features that are selected to learn the task are manually selected by the user. Working with such specially chosen features often produces satisfactory results inasmuch as the skilled demonstrator knows which feature (positions, orientations, state of the gripper, ...) is relevant to study. However, the feature selection has to be rethinked from scratch for each new task.

Contributions: The proposed framework integrates state-of-the-art generalization techniques developed mainly for armed robots to complex humanoids, supplemented by a preprocessing method that automatically selects the most relevant features of the studied task and the ClaSP [24] algorithm to segment multivariate time-series in complex demonstrations.

C. Outline

Section II describes the overall learning and generalizing framework and the automatic and multisensory segmentation method used in it. Section III puts on display the simulation and experimental results. Conclusions and future work possibilities are finally explored in section IV.

II. Learning from demonstrations

A. Gathering demonstrations

Building a dataset that covers a wide range of configurations is essential to reach an efficient generalization of the task, hence providing mutliple demonstrations to the robot. To do so, we demonstrated the same task in a similar environment (in the same room, with similar objects on the table, ...) but with different conditions. These include different position parameters (poses of the object, goal poses, grasping poses) as well as variate dynamic variables (speed of the movement). Having such set of demonstrations limits the *covariate shift*, that's to say the difference of distribution between the training dataset and the real-world cases. Nonetheless, we can not get rid of this covariate shift in the extent that the action space of a given task in the real world is most often far bigger than what can represent a few demonstrations. In addition to that, our approach is to have a dataset corresponding to what might come natural to a non-professional user.

B. Feature Selection

Once the demonstrations gathered, the choice of the features to study is another core step that often requires the experience of a skilled user. Whereas a great majority of LfD techniques selects manually the relevant features to study, we chose to take the measurements of all the sensors available as well as the proprioceptive information, namely 6D Cartesian position and orientation and angular and linear velocites obtained from the robot's solver. Mainly two reasons justify that:

- A non-skilled user has to be able to demonstrate a new task only by achieving it in a natural way while wearing the teleoperation device
- Capitalize on our similarity with humanoid robots. Indeed, recording sensory parameters amounts to add a kind of *sensory integration* to the robot. *Sensory integration* was first theorized by Dr A. Jeans Ayres in 1972 in the field of neuroscience [25]. It states that when thinking about achieving some known task, our brain processes not only how to reach the goal (the different steps), but also the sensations we had when achieving this task in the past. A telling example of *sensory integration* is that when we grab an object, the brain area corresponding to our sense of touch activates, and we expect to have some sensation when grabbing an object. Thus, taking sensor into account while building the dataset can help to detect new phases of the movement [26] and correcting the movement online (after the learning phase).

Having gathered a set of demonstrations containing numerous features, we now want to automatically extract the most relevant features to segment the task in skills reproducible with DMP.

A first step is to temporally align the signals in order to compare the features across the dataset. Indeed, our goal is to find the most correlated features. Intuitively, if one feature has the same shape in all the trials, it is likely that it will be useful for segmentation, and that it is critical to define the task. For example, in a grasping task, the opening of the gripper clearly identifies the different phases of the movement and has the same shape across all the trials. On the contrary, we expect that other parameters like the position along the z-axis have no importance in the grasping task.

To align the signals, we use Dynamical Time Warping (DTW). The main idea behind DTW is to find the optimal alignment between the two sequences by warping their time axes. This warping process allows for the matching of similar patterns, despite variations in timing and duration. It works by finding the minimum distance path through a grid or matrix that represents the pairwise distances between elements of the two sequences.

We can then compute a *correlation matrix* for each feature across all trials (Fig. 3). The mean of the coefficients of these matrices can therefore be interpreted as the similarity across the datasets. This first result, albeit convincing, include a bias due to the temporal alignment of the signals. Indeed, it is possible to end up with a highly correlated feature when the signals are aligned, but in reality very distant. In fact, the distance computed with DTW can also be used as a tool to measure similarity. Taking again the statistical mean of the two by two distances for every feature and normalizing the values between 0 and 1, we can penalize the correlation coefficients computed afterwards. We employ the normalized DTW distances as weights for the correlation coefficients. This allows to take into account the distance between the trials, eventually leading to a more objective measure of the relevance of the features. We see in Fig. 4 that the most correlated features are even more detached from all the other

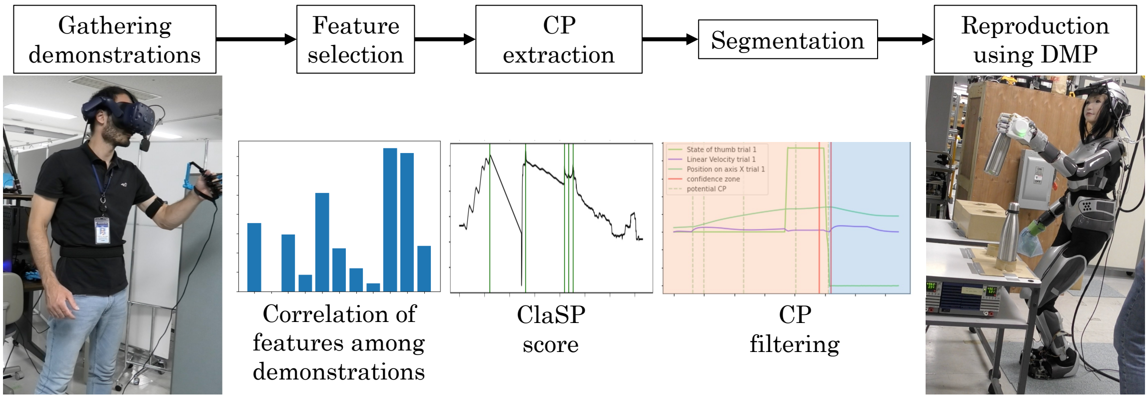


Fig. 2: Proposed task learning and generalization framework. First, we demonstrate the task with a teleoperation device (image on the left). The data is then preprocessed and features that are the most correlated (histogram on the mid-left) are selected for the segmentation with the ClaSP algorithm (image on the middle). This allows for the potential CPs to be identified and filtered with sensor values (image on the mid-right). Finally, we can reproduce the task with Dynamic Movement Primitives on the HRP-4C robot (image on the right).

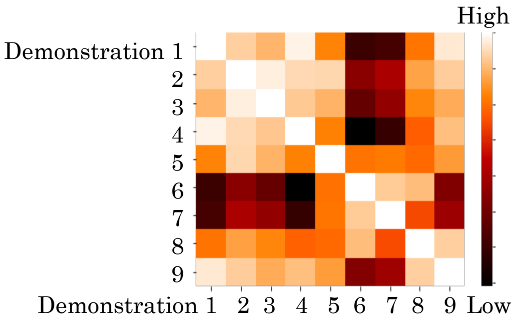
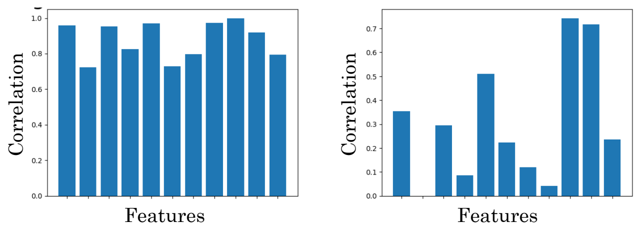


Fig. 3: Example of a correlation matrix that shows how one feature (here, the angular velocity of the end-effector) is correlated across the 9 demonstrations.



(a) Average on the correlation coefficients only

(b) Taking into account the distance computed with Dynamical Time Warping

Fig. 4: Histogram of feature correlation across trials. The relevance of the feature is a lot clearer on the right figure that takes into account the distance computed with DTW.

features.

Finally, we select the features to pass in argument for the segmentation phase. There is a trade-off between selecting the most correlated features and discarding others - thus possibly missing important information - and choosing too much, - thus thwarting the segmentation with uncorrelated features -. In addition to the most correlated feature F, we select the features that are above $CorrCoef_F \times T$, with T a threshold that we set at 0.9.

C. Segmentation

To segment demonstrations, we used the ClaSP Algorithm that outputs a list of potential CP of a multivariate time series.

Then, we tried different methods to combine the profiles of the features studied. We considered averaging the profiles and multiply the profiles, but also weighted average and product. As intuitively the most correlated features have a score profile that is more relevant for the segmentation. Thus, we weight the score profiles with the correlation coefficients computed in II-B before averaging or multiplying them. As we want to reduce the number of little segments, we penalize the score profile of the points next to potential breakpoints. By computing these CPs recursively, two detected CPs at step i or lower are the bounds of the new time series studied at step $i+1$, we penalize the points near the edges of the profiles by applying a Kaiser filter. The output of the Kaiser filter is displayed on Fig. 5.

This provides us with a score profile that contains potential CPs. We then either discard or validate the CPs thanks to sensory events. Indeed, as explored in [26], new phases in manipulation tasks coincide with a sensory event. We reuse this idea and compute a confidence zone around the beginning and endings of sensory events. We first clean the sensory data, that's to say remove the first difference and perform a linear regression over a sliding window to smooth the curve. Then, we detect the sensory events (when the data goes from a low state to a high state and vice-versa, with a threshold set as $0.4 \times max\_value$. Having detected this set of points S for the time series T and the minimum length of a segment min_s, we thus obtain the confidence zone:

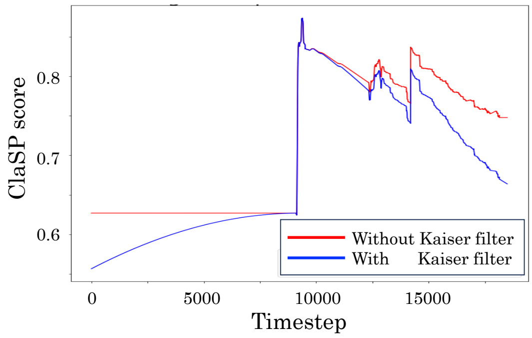


Fig. 5: ClaSP profile after being filtered with the Kaiser filter. The value of the filtered profile fades on the edges of the profile, thus limiting the possibility to have a big sequence of short subtasks.

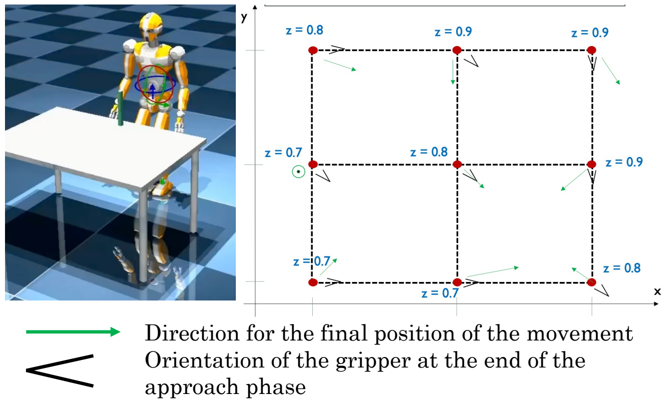


Fig. 6: Setup for the dataset built in simulation. On the left, the Mujoco interface with the JVRC1 [28] at the beginning of the grasping task. On the left, the positions in which is positioned the object to grab, with the pose of the gripper before grasping the stick and the direction in which the latter is lifted.

$$\bigcup_{s \in S} [\max(0, s - min_s/2), \min(s + min_s/2, \text{len}(T))] \quad (1)$$

We finally validate the CPs computed with ClaSP if it belongs to the confidence zone. That way, we have the CPs that corresponds to the sensory events , but we do not accept CPs only if the sensors are activated.

III. Results and Discussion

We first design a controller to build a dataset in simulation on the JVRC1 robot. To that end, we designed a finite state machine (FSM) controller with B-splines with mc-rtc[1] and mc-mujoco [27] to simulate a one-handed grasping task of a stick in 9 different configurations as seen on Fig. 6. Then, we build a real-world dataset, with the teleoperation device with the HRP-4C robot (Fig. 1). The user has a the visual input of the humanoid's front camera, and achieves the task naturally, except that there is no force feedback on the fingers, meaning that the demonstrator is not constrained by the physical shape of the object during the demonstrations.

The same process is employed to build two datasets for a pushing task, where the goal is merely to push a box on a table using a hand. The video of the demonstration with teleoperation are available here.

In both cases, mc-rtc logs all the sensor and proprioceptive values that are then used to learn and generalize the task. The task can be represented by all its normalized feature values in skill maps as in Fig. 7. Using this map, we extract appropriate features and segment it into interpretable phases. The 3 features selected for the grasping task were the state of the gripper, the linear velocity and the position on the x axis.

To consider that a segmentation is correct, a demonstration of a grasping task must have one CP at the transition between the approach phase (when getting closer to the object) and the moving-object phase (grabbing the object and move it towards an other position). Figure 8 is a correct segmentation for a grasping task. As we can see on this figure, for the considered trial, five potential CPs were identified, but only one is within the confidence zone, thus giving in the end a correct segmentation for this grasping task. Over the dataset, we compute the success rate as the number of trials correctly segmented over the total number of trials.

As for the pushing task, two CPs are expected to be identified. We also have an approach phase when getting closer to a box to push. The pushing phase occurs when the gripper is in contact with the object. Finally, the third phase is simply returning to the initial position.

Note that we consider only correct these segmentations, but even if too much CPs are detected, the movement reproduction with DMPs will still be functional. In this case, main issues are that the segments are not interpretable by a human, and end up in task that are not learned in an optimal fashion.

Results are shown on Fig. 9 for the grasping task and the pushing task, and leads us to choose the weighted product method to compute the combined profiles. Indeed, it shows some better outcome than other methods both in simulation and on real-world data.

While only 50% of the demonstrations are correctly segmented in the real grasping task, the results obtained in simulation are around 80%. This is due to noisy logs when using the teleoperation device, especially on sensory data, thus having a confidence zone that is not as accurate as that of the simulated data. We detect too much sensory events, eventually finding too much CPs (suboptimal partition of the task). Teleoperating - and more generally working with - humanoid robots that have complex hybrid behaviors ends up with having noisy data that is harder to exploit than on robots that have less DoFs (robotic arms, quadruped robots, ...).

Such results can also come from the demonstrations contained in the dataset. All real-world demonstrations are more

[1]https://jrl-umi3218.github.io/mc_rtc/index.html

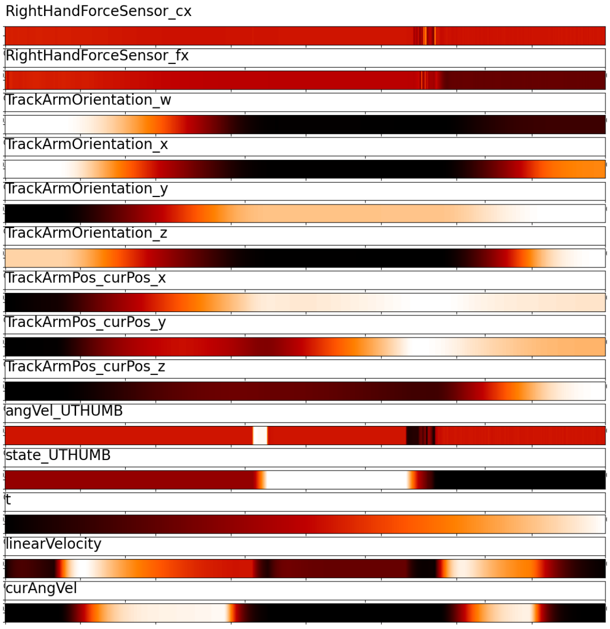


Fig. 7: Representation of a task in a skill map. All the features are normalized between 0 and 1. Throughout the entire task (represented in timesteps on the abscissa), we observe the evolution of all the features studied. The higher the value of the normalized feature, the clearer on the skill map.

similar to each other than that of the simulation dataset. In simulation, the demonstrations by the JVRC1 robot covered a surface of $40 \times 40cm$ on the table, while it is restrained to $15 \times 15cm$ on real-world demonstrations by the HRP-4C robot because of the differences in kinematics and dynamics parameters of the robots and controllers. We tend to have correlated features in the real world dataset that should not be correlated for the considered task, eventually leading in computing a profile that have "false" CPs that are not discarded because the sensory values (from the force or torque sensors for instance) that are too noisy, thus detecting wrong confidence zones that eventually end up with no CP detected or the wrong CP detected.

To reproduce the task, we compute DMP models for each segment of each demonstration. We learn one DMP for the 6d-pose and one for the state of the gripper. Both are coordinated by the same phase variable.

When performing again this learned skill given a new goal position, we interpolate the new goal with all the precedently learned goals to find a new DMP model that fits to the task. Tasks have been reproduced using this method for the simulated grasping and pushing tasks with 100% rate of success for the demonstrations that were correctly segmented. Snapshots of the reproduction are showed on Fig. 10. The execution time to reproduce the task was longer (approximately 1.5 times longer with a processor i9-10885H due to bigger calculations at runtime to compute the next desired position with the DMP). The reproduction on the real robot is a more complex issue that is left to future work.

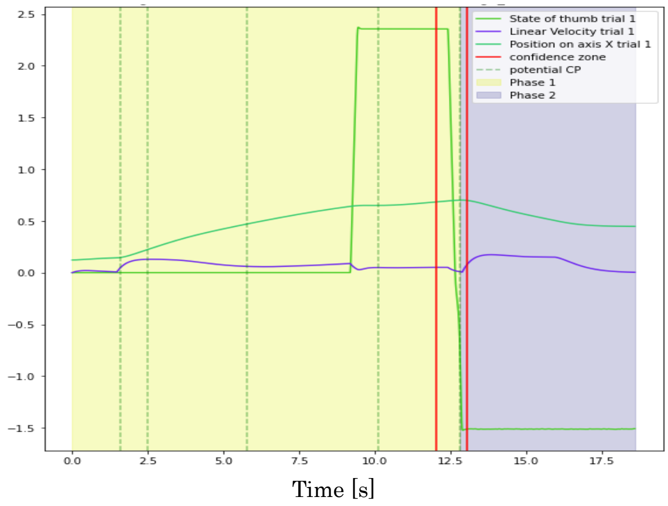


Fig. 8: Correct segmentation of a grasping task. We have one CP inside the confidence zone, that is when the gripper is closing itself on the object. The different phases are coloured in beige (approach phase) and blue (move object phase).

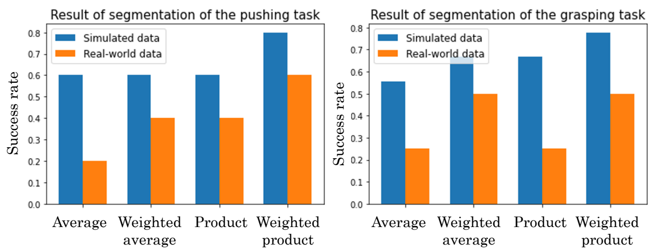


Fig. 9: Results of the segmentation for the tasks considered. While approximately 80% of the task are successfully reproduced in simulation (blue), the average percentage is around 50% on the real robot (orange).

IV. Conclusion and Future Work

Throughout this work, we presented an end-to-end framework to learn and generalize a new task for humanoid robot by automatically segment demonstrations provided in a natural way through teleoperation into reproducible skills. On top of that, automatically extracting the relevant features to learn the task with a correlation analysis corroborated with sensory information enables non-skilled users to learn tasks to the robots, therefore fostering humanoids integration in real-world work sites. Our approach, albeit not perfect, shows satisfactory results on simulated data (with around 80% of success) but has to be refined to be efficient on real-world data.

This work mainly paves the for two potential prolongations.

- *Increase the efficiency and the automation.* This would include compute 6d-pose estimation of objects based on visual input, or refine our DMP implementation to make it more efficient, add goal-switching and obstacle avoidance as stated in [29].

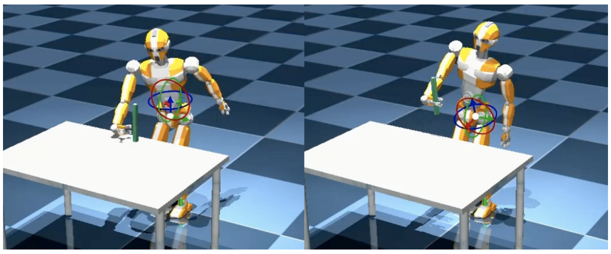


Fig. 10: Reproduction of the grasping task in simulation on the JVRC1 robot. On the left, the end of the approach phase: the gripper is open near the stick (the position of the stick was manually input). On the right, the end of the lift phase: the robot holds the object in the final position (also manually set).

- *Increase the generalizability and the objectivity.* One could think about replacing the arbitrary thresholds in the feature selection or in the segmentation algorithm by statistical methods, or trying to avoid covariate shift by generating demonstrations through meta-learning [30].

References

[1] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in Cognitive Sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[2] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[3] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent Advances in Robot Learning from Demonstration," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 297–330, 2020.

[4] C. Sammut, S. Hurst, D. Kedzier, and D. Michie, "Learning to Fly," in *Machine Learning Proceedings 1992*, 1992, pp. 385–393.

[5] Z. Wu, W. Lian, C. Wang, M. Li, S. Schaal, and M. Tomizuka, "Prim-LAfD: A Framework to Learn and Adapt Primitive-Based Skills from Demonstrations for Insertion Tasks," 2022, arXiv:2212.00955 [cs].

[6] K. Ramirez-Amaro, M. Beetz, and G. Cheng, "Understanding the intention of human activities through semantic perception: observation, understanding and execution on a humanoid robot," *Advanced Robotics*, vol. 29, no. 5, pp. 345–362, 2015.

[7] M. Vecerik, O. Sushkov, D. Barker, T. Rothorl, T. Hester, and J. Scholz, "A Practical Approach to Insertion with Variable Socket Position Using Deep Reinforcement Learning," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 754–760.

[8] 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.

[9] J. Rouot and J.-B. Lasserre, "On inverse optimal control via polynomial optimization," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 721–726.

[10] Y. Zhao, P. Liu, W. Zhao, and X. Tang, "Variational Diversity Maximization for Hierarchical Skill Discovery," *Neural Processing Letters*, vol. 55, no. 1, pp. 839–855, 2023.

[11] B. D. Argall, B. Browning, and M. Veloso, "Learning robot motion control with demonstration and advice-operators," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 399–404.

[12] M. A. Rana, M. Mukadam, S. R. Ahmadzadeh, S. Chernova, and B. Boots, "Towards Robust Skill Generalization: Unifying Learning from Demonstration and Motion Planning," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 109–118.

[13] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5628–5635.

[14] C. Chuck, K. Black, A. Arjun, Y. Zhu, and S. Niekum, "Granger-causal hierarchical skill discovery," 2023, arXiv:2306.09509 [cs.AI].

[15] A. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 2, 2002, pp. 1398–1403.

[16] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.

[17] F. Meier, E. Theodorou, F. Stulp, and S. Schaal, "Movement segmentation using a primitive library," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 3407–3412.

[18] R. Caccavale, M. Saveriano, A. Finzi, and D. Lee, "Kinesthetic teaching and attentional supervision of structured tasks in human–robot interaction," *Autonomous Robots*, vol. 43, no. 6, pp. 1291–1307, 2019.

[19] A. Saran, E. S. Short, A. Thomaz, and S. Niekum, "Enhancing Robot Learning with Human Social Cues," in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2019, pp. 745–747.

[20] 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.

[21] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and Reproduction of Gestures by Imitation," *IEEE Robotics & Automation Magazine*, vol. 17, no. 2, pp. 44–54, 2010.

[22] 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 (Cybernetics)*, vol. 37, no. 2, pp. 286–298, 2007.

[23] O. Kroemer, C. Daniel, G. Neumann, H. Van Hoof, and J. Peters, "Towards learning hierarchical skills for multi-phase manipulation tasks," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1503–1510.

[24] A. Ermshaus, P. Schäfer, and U. Leser, "ClaSP: parameter-free time series segmentation," *Data Mining and Knowledge Discovery*, vol. 37, no. 3, pp. 1262–1300, 2023.

[25] A. J. Ayres, "Improving Academic Scoresthrough Sensory Integration," *Journal of Learning Disabilities*, vol. 5, no. 6, pp. 338–343, 1972.

[26] Z. Su, O. Kroemer, G. E. Loeb, G. S. Sukhatme, and S. Schaal, "Learning manipulation graphs from demonstrations using multimodal sensory signals," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2758–2765.

[27] R. P. Singh, P. Gergondet, and F. Kanehiro, "mc-mujoco: Simulating articulated robots with fsm controllers in mujoco," in *2023 IEEE/SICE International Symposium on System Integration (SII)*, 2023.

[28] M. Okugawa, K. Oogane, M. Shimizu, Y. Ohtsubo, T. Kimura, T. Takahashi, and S. Tadokoro, "Proposal of inspection and rescue tasks for tunnel disasters — task development of japan virtual robotics challenge," in *2015 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2015.

[29] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic Movement Primitives in Robotics: A Tutorial Survey," 2021, arXiv:2102.03861 [cs].

[30] T. Yu, C. Finn, A. Xie, S. Dasari, T. Zhang, P. Abbeel, and S. Levine, "One-Shot Imitation from Observing Humans via Domain-Adaptive Meta-Learning," 2018, arXiv:1802.01557 [cs].