Automatic Parameterization of Motion and Force Controlled Robot Skills

Vamsi Krishna Origanti[1], Thomas Eiband[1], and Dongheui Lee[1,2]

[1] German Aerospace Center (DLR), Institute of Robotics and Mechatronics
[2] Technical University of Munich, Human-centered Assistive Robotics, Germany

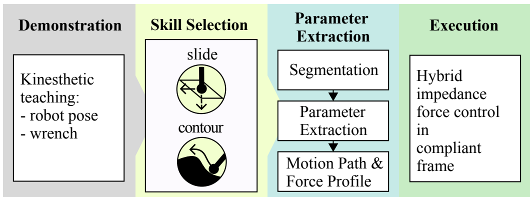
Abstract. Compliant robot tasks such as grinding require a robot to use a specific control strategy and to consider a number of process parameters. It is demanding to program such behaviors from scratch. Therefore, so called contact skills can be employed that are pre-programmed control strategies, which are optimized for the intended task. With that level of abstraction, which is defining skills that are specific to the task, only the skill's parameters need to be identified and not the whole strategy to be implemented. In order to allow non-experts to transfer such complex behaviors to a robot, we present two different contact skills and how they are automatically parameterized by a human demonstration. This process learns the robot behavior in one shot while considering task goals, such as desired forces and motions. We evaluated our framework in the PyBullet physics simulator and showed that the parameterized skills follow the task goals while generalizing to changes in the environment.

1 Introduction

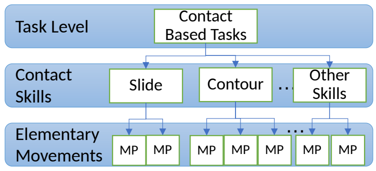
In today's industrial applications, robots are employed in processes with large lot sizes and are manually programmed by experts. When it comes to small lot sizes and highly customized products, the demand for intuitive programming techniques raises in order to implement new processes quickly. Hereby, the Learning from Demonstration (LfD) [1] technique enables defining a task by nonexperts. However, it has been shown that the robot could suffer from a bad teacher's demonstration performance [2]. We encounter this by incorporating expert knowledge in the form of so called robot skills.

In the LfD context, a skill is a predefined robot behaviour parameterized by the demonstrations [3] [4]. Since force-based tasks are relatively hard to implement when compared to only kinematic tasks, we can make use of expert knowledge embedded in the skills in the form of a specific controller. An example scenario is a surface processing task with a grinding tool. Hereby, a constant normal force against the surface is desired for a stable grinding process and a motion needs to be tracked along the surface. The demonstration of a human might not be optimal and contain undesired variations. Therefore, a designated skill is able to perform force control along the surface normal and motion control on the surface plane, which is known as hybrid position force control.

We identified a number of so called contact skills in a previous work [5], which are intended to solve motion and force-based tasks. Additionally, we were able



(a) Learning from Demonstration pipeline by parameterization and execution of a selected skill.



(b) Hierarchical Representation of Contact-Based Tasks, Skills, and Movement Primitives (MP).

Fig. 1: Overview

to automatically recognize the skill within the demonstration with a data-driven classification approach (used for *Skill Selection* in Fig. 1a). In this work, we focus on two of these skills, namely *slide* and *contour*. *Slide* is intended to apply force onto the normal direction of a surface, for example as used in grinding applications. *Contour* is a skill to apply a force profile onto the normal direction of a nonlinear shape, for example, used in deburring of sharp edges. The main motivation of this work is to present an automatic parameterization technique in combination with a control strategy, such that the robot can parameterize each contact skill automatically from the demonstration. Finally, we want the robot to reproduce the skill by adapting to uncertainties in the environment using our presented hybrid controller.

Our LfD pipeline is shown in Fig. 1a and starts with the observation of a human demonstration, uses *Skill Selection* as presented in [5], and employs *Parameter Extraction* and *Execution* as presented in this paper. In detail, we propose a strategy of how two different contact skills can be parameterized automatically from a human demonstration without manual parameter setting. This enables non experts to bootstrap new tasks that involve motion and force control without manual programming. To enable such control objectives, we extract *compliant frames* (CF) automatically from the demonstration, which aligns motion and force control axes towards the environmental constraints.

The main contributions of this work are given as follows:

1. A skill learning strategy with automatic extraction of skill parameters from demonstration data.
2. A skill execution strategy that facilitates a hybrid impedance-force controller that enables the rigid transformation to the CF.

2 Related Work

2.1 Contact Skill Learning

Contact skills require a control strategy to handle contact forces besides the motion execution. Pais et al. [6] proposed a framework for learning robot skills through motion segmentation and constraints extraction. The approach compares the variance in a time window to the variance among multiple demonstrations, which are required to learn the task. In contrast, our approach learns from a single shot as a reasonable simplification for real world applications. In addition, we enable a tailored solution for different task constraints by proposing so called contact skills.

Kober et al. [7] presented a way of learning movement primitives (MP) that depend not only on motion characteristics but also on force interactions. Similar to the method employed in [6], they enabled learning by a preceding segmentation based on Zero Velocity Crossings (ZVC), later computed scores to choose a reference frame and control variables. Each segment of the trajectory is considered as MP, encoded as Dynamic Movement Primitive (DMP). This method also requires multiple demonstrations to evaluate the scores and therefore is demanding for the human teacher. Reference frames were predefined for the task, which we extract directly from the data. A method for simultaneous teaching of position and force for contact-based tasks was presented in [8]. DMPs were used to learn the motion as well as force profile. A hybrid controller was employed for reproduction. A task specific reference frame was not considered as the task was performed in a fixed frame and with a predefined control strategy for each dimension. Learning of sequential skills considering force interactions was presented in [9]. A dynamical system with a linear attractor was chosen as MP and a hybrid position force controller similar to [7] employed. The trajectory is segmented to form movement primitives, based on multiple demonstrations. The underlying task frames were manually defined.

The aforementioned methods did not extract any reference frames from demonstration data but used one or multiple predefined frames. On the contrary, Conkey et.al [10] extracted a dynamic frame, which is called a constraint frame by considering the motion and observed force profile. They also defined a selection matrix to specify a control strategy in this constraint frame.

Gao et al. [11] presented the learning of force relevant skills from demonstrations, where a skill is represented as a function of position, velocity, interaction forces, and task constraints. In contrast to our approach, here multiple demonstrations are required that are encoded in Gaussian Mixture Models (GMM). Task execution was performed on a hybrid controller based on admittance control, whereas, in our work, an impedance based hybrid controller is used. Also, the learning is carried out in a generic way, which is in contrast to our approach, where we perform skill specific learning.

2.2 Hybrid Position-Force Control

Pure impedance control is not sufficient to reproduce all facets of a compliant contact task, for example, to fulfill the requirement of closed loop force control.

A hybrid position-force controller that offers a separation of position and force control in orthogonal subspaces, initially proposed in [12] is a suitable methodology. A similar control strategy was employed in [7] and [9] based on a task level inverse dynamics approach, which was initially proposed in [13] as a hybrid control strategy. In [8], a similar control strategy is adopted to teach and reproduce in-contact tasks with an impedance controller for motion control and a PI controller for force control. However, force control was fixed to to the z axis.

Hybrid control strategies aforementioned under LfD and contact-based tasks are suitable when tracking interaction forces in a fixed Cartesian frame. However, these strategies are not sufficient when a task frame undergoes a transformation during execution. Conkey et al. [10] addressed a similar issue based on [14], wherein a dynamically changing constraint frame was embedded in the control equation. This method transforms a selection matrix that is intuitively defined in the constraint frame into the end effector frame of the robot. Marin et al. [15] presented a unified hybrid position force controller based on the Kinesthetic filtering method proposed in [16] and [17]. They clarify that when a compliant frame undergoes a rigid transformation, the motion and force commands issued in the compliant frame need to be transformed, in other words, kinesthetically filtered to have the controlled subspaces separated in the new frame.

We employ a generic hybrid controller for both of our skills. To do so, we define control parameters like stiffness and selection matrix in the compliant frame that is task specific. Then, we consider the rigid transformations between the end effector and compliant frame in the control loop, similar to [10] and [15]. In contrast to [10], we use an impedance controller for position control and a PI controller for force control. In comparison to [15], we also consider the transformation of the stiffness matrix to be expressed in the compliant frame, which allows us to interpret and sensibly define the stiffness values for each skill.

3 Skill Architecture

We define our skills to have static attributes, which are part of the skill implementation, and adaptable parameters, which are task specific. Our skill-based learning framework is based on a motion segmentation and skill encoding process. We assume that a skill can be represented as a sequence of MPs and perform trajectory segmentation on the skill level to encode the resulting segments as MPs (see Fig. 1b). Then, compliant frames are extracted from the demonstration in order to define the directions of motion and force control. Finally, we extract previously identified parameters specific to each skill to make it ready for execution.

3.1 Skill Attributes

Skill attributes are skill specific and static parts of the skill implementation as programmed by an expert. They determine values and techniques of how the skill is learned and executed.

Segmentation Method Segmentation is one of the most common steps observed in many research works prior to MP learning. Considering only a single MP for a complex trajectory could lead to information loss and bad generalization performance [18]. By segmenting the trajectory into multiple MPs and by sequencing them, complex tasks can be learned with more accuracy.

We use an unsupervised learning method to segment a trajectory by grouping similar points [19], consisting of robot pose and force. The issue of handling spatial and temporal clustering is addressed in [20–22] for segmenting trajectory data in the robotics domain. A similar methodology is adopted here to segment demo data using Agglomerative Clustering in the spatial domain but for temporal constraint, in contrast with aforementioned references, we considered K-Nearest Neighbors (KNN) Graph as a prior, to define temporal constraints as shown in Fig.2.

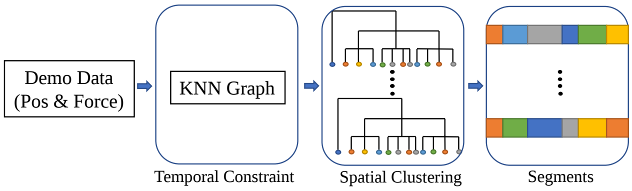


Fig. 2: Segmentation by Agglomerative Clustering with temporal constraints.

Selection Matrix We define a selection matrix with respect to an automatically extracted compliant frame (CF) in order to select either force or motion control in each Cartesian dimension. This information is used as a control parameter of our proposed hybrid motion force controller. We state that every skill has its unique way to deal with force application. Therefore, the selection matrix is a skill specific attribute. Having the Cartesian dimensions of $[x, y, z, r_x, r_y, r_z]$, a selection matrix example expressed in CF is given as ${}^{CF}\boldsymbol{S} = \mathrm{diag}([0, 0, 1, 0, 0, 0])$, which specifies force control in the z axis and motion control in the x, y plane as well as in the orientation dimensions $[r_x, r_y, r_z]$.

Compliant Frame Constraint A CF denoted as transformation ${}^{EF}\boldsymbol{T}_{CF}$ between end effector frame (EF) and compliant frame (CF) is extracted from the demonstration data with respect to an underlying constraint as shown in Fig. 3b and 4b. The constraint type is statically predefined for each skill.

3.2 Skill Parameters

Skill parameters are task specific and are extracted from the individual demonstration of a skill. The following parameters are commonly used for both skills, but extracted with respect to different constraints as described later.

Compliant Frame A compliant frame is a task frame coordinate system, which axes allow a meaningful separation into motion and force subspaces. It is extracted based on the predefined compliant frame constraint. Fig. 3b shows an example of a CF extracted for the *slide* skill from demonstrated data. The compliant frame is oriented such that the z axis resembles the surface normal and the x axis is aligned with the motion direction. The compliant frame can vary with respect to the end effector frame. This formulation allows to define specific controllers for different axes of the compliant frame, for instance, force control along the z axis and motion control in the x, y plane.

Motion Path and Force Profiles We use DMPs to encode each of the previously extracted segments containing position and orientation path as well as force profile. Hereby, we use a standard formulation [23] for both positions and forces and an extended formulation [24] for orientations.

4 Skill Implementation

The previously described attributes need to be set for each skill and the introduced parameters need to be extracted. In the following, we define two contact skills with their attributes and parameters. We further describe, how the parameters are extracted from the demonstration.

4.1 Slide

The *slide* skill is intended to move along a planar surface while applying force, for instance, required in grinding, sanding, polishing or wiping a surface. As shown in Fig. 1b, learning a skill involves segmentation, later parameter extraction specific to the skill and then performs DMP learning on the selected variables. Hereby, after segmenting the trajectory into MPs, a constant force should be applied normal to the plane, which is extracted during learning and applied throughout the motion during reproduction. In order to reproduce this skill, the robot should learn the trajectories of motion and contact normal force and apply it to the environment as shown in Fig. 3b. The attributes and parameters can be found in Table 1.

Compliant Frame Extraction: Considering the *slide* skill, a constant normal force needs to be applied on a plane of a trajectory. Hence, a compliant frame can be defined on the plane, and such forces can be applied only on the z axis of the compliant frame. In order to define such a compliant frame, a plane needs to be fit on the segment.

Table 1: Skill: Slide

Attribute	
Segmentation Method	Agglomerative Clustering
Selection Matrix ${}^{CF}\boldsymbol{S}$	diag($[1, 1, 0, 1, 1, 1]$)
CF Constraint	plane on motion data (Fig. 3b)
Parameter	
Compliant Frame ${}^{EF}\boldsymbol{T}_{CF}$	Constant CF for each segment
Position Path	DMP in x, y dimensions of CF.
Orientation Path	Quaternion DMP.
Force Profile	Constant force value obtained from the average of values in low variance region of the Z axis force profile in CF.

4.2 Contour

For a *contour* skill, the force varies at each point throughout the profile. Along with demonstrated motion path, force application on to the environment also needs to be considered for learning and reproducing. Instead of applying forces in all directions in the world frame, a varying normalized force learned from the demonstration is applied at each time step in the CF. In this way, we can control the motion and force application simultaneously. The overview of the skill learning procedure is shown in Fig. 1a. In order to learn and reproduce this skill, a force profile and a varying CF will be extracted at each time step throughout the trajectory as shown in Fig. 4b

Instead of learning varying force profiles, one could extract a constant force value by averaging forces from the demonstration data, considering that human demonstrations are not precise throughout the trajectory. However, the requirement of constant force or varying force is based upon user requirements. In our experiments, we considered a varying force profile, which is learned from the data. The attributes and parameters can be found in Table 2.

Compliant Frame Extraction: For the *contour* skill, the CF is a local coordinate system that varies dynamically during execution. First, we extract the z axis of the CF as proposed by Conkey et.al [10]. Hereby, the z axis is aligned with the normalized force vector of unit length obtained from the demonstration data. For the alignment of the two remaining orthogonal axis, there exist infinite solutions. In order to obtain a unique solution, the two remaining orthogonal axes are constructed by the strategy that was presented in [25]. It finds the rotation with minimum cost between two frames and therefore is guaranteed to find a unique rotation matrix for each point in space. This unique rotation matrix defines the orientation of the CF.

Table 2: Skill: Contour

Attribute	
Segmentation Method	Agglomerative Clustering
Selection Matrix ${}^{CF}\boldsymbol{S}$	diag([1, 1, 0, 1, 1, 1])
CF Constraint	adapted to environmental shape (Fig. 4b)
Parameter	
Compliant Frame ${}^{EF}\boldsymbol{T}_{CF}$	Series of variable frames with z axis aligned to normalized contact force vector.
Position Path	DMP in x, y dimensions of CF.
Orientation Path	Quaternion DMP.
Force Profile	Trajectory of force obtained by computing magnitude of normalized force vector at each point of path.

5 Skill Execution

To reproduce contact based skills, we need to track motions and forces simultaneously. Therefore, we employ a hybrid impedance force controller that acts with one of the control modes in each task space dimension. We derive a hybrid control scheme $\boldsymbol{\tau} = \boldsymbol{\tau}_{ic} + \boldsymbol{\tau}_{fc}$ based on [12], written as

$$\begin{aligned}\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q})({}^{EF}\boldsymbol{K}_c({}^{EF}\boldsymbol{S})({}^{EF}\boldsymbol{e}_p) + \boldsymbol{D}_x\boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{\theta}} + \boldsymbol{g}(\boldsymbol{q})) \\ +\boldsymbol{J}^T(\boldsymbol{q})(\boldsymbol{K}_p({}^{EF}\tilde{\boldsymbol{S}})({}^{EF}\boldsymbol{e}_f) + \boldsymbol{K}_i({}^{EF}\tilde{\boldsymbol{S}})\int^{EF}\boldsymbol{e}_f dt).\end{aligned} \tag{1}$$

where $\boldsymbol{\tau}_{ic}$ denotes a joint torque command, $\boldsymbol{J}(\boldsymbol{q})$ is the Jacobian of the robot, $\boldsymbol{K}_c$ the stiffness, $\boldsymbol{e}_p$ the position error, $\boldsymbol{D}_x$ denotes damping and $\boldsymbol{g}(q)$ the gravity compensation term of the nonlinear dynamic system. $\boldsymbol{\tau}_{fc}$ denotes a joint torque command, $\boldsymbol{e}_f$ denotes the force error. $\boldsymbol{K}_p$ and $\boldsymbol{K}_i$ are proportional and integral constants of the PI controller respectively. $\boldsymbol{S}$ and $\widetilde{\boldsymbol{S}}$ denote the diagonal selection matrix and its complement. The selection matrix enables the selection of position control or force control in each dimension. However, the classical hybrid controller mentioned above does not consider a compliant frame (CF), which undergoes a task specific transformation extracted as a skill parameter. In consequence, stiffness parameter $\boldsymbol{K}_c$ and selection matrix $\boldsymbol{S}$ are diagonal matrices based on interaction forces, which are also with respect to the CF. Now, stiffness $\boldsymbol{K}_c$, selection matrix $\boldsymbol{S}$ and its compliment $\widetilde{\boldsymbol{S}} = (\boldsymbol{I}_6 - \boldsymbol{S})$ are expressed in EF. The transformation of 6×6 stiffness matrices between frames is provided in [26] and [27] and can be achieved with adjoint matrices derived from a homogeneous transformation matrix. Similarly, adjoint matrices are used to transform twists and wrench quantities represented in 6×1 vectors respectively.

Consider that the CF is represented as homogeneous transformation matrix ${}^{EF}\boldsymbol{T}_{CF}$ with rotation $\boldsymbol{R}$ and translation $\boldsymbol{p}$: According to [28], The adjoint matrices from the above homogeneous transformation can be written as

$$
{}^{EF}\boldsymbol{Ad}_{CF} = \begin{bmatrix} \boldsymbol{R} & [\hat{\boldsymbol{p}}]\,\boldsymbol{R} \\ \boldsymbol{0} & \boldsymbol{R} \end{bmatrix}, \quad {}^{EF}\boldsymbol{Adg}_{CF} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{0} \\ [\hat{\boldsymbol{p}}]\,\boldsymbol{R} & \boldsymbol{R} \end{bmatrix}, \tag{2}
$$

Using above adjoint matrices, stiffness $\boldsymbol{K}_c$, selection $\boldsymbol{S}$ and its complement $\widetilde{\boldsymbol{S}}$ matrices can be represented in CF from EF as

$$
\begin{aligned}
\boldsymbol{\Omega} &= {}^{EF}\boldsymbol{K}_c\,{}^{EF}\boldsymbol{S} \\
\boldsymbol{\Omega} &= {}^{EF}\boldsymbol{Adg}_{CF}\,{}^{CF}\boldsymbol{K}_c\,{}^{CF}\boldsymbol{S}\,{}^{EF}\boldsymbol{Ad}_{CF}{}^{-1} \\
\tilde{\boldsymbol{\Omega}} &= {}^{EF}\tilde{\boldsymbol{S}} = {}^{EF}\boldsymbol{Adg}_{CF}\,{}^{CF}\tilde{\boldsymbol{S}}\,{}^{EF}\boldsymbol{Adg}_{CF}{}^{-1}
\end{aligned} \tag{3}
$$

By substituting equation (3) in (1), we obtain the final control law

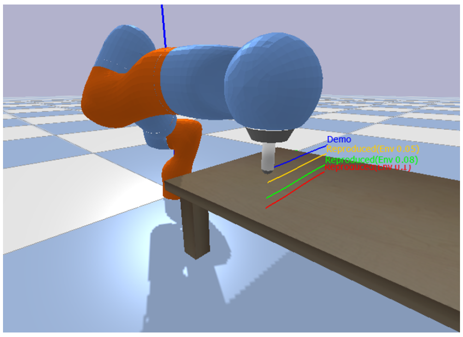
$$
\boldsymbol{\tau} = \boldsymbol{J}^T\boldsymbol{q}\left(\boldsymbol{\Omega}\,{}^{EF}\boldsymbol{e}_p + \boldsymbol{D}_x\boldsymbol{J}\boldsymbol{q}\dot{\boldsymbol{\theta}} + g(\boldsymbol{q})\right) + \left(\boldsymbol{K}_p\widetilde{\boldsymbol{\Omega}}\,{}^{EF}\boldsymbol{e}f + \boldsymbol{K}_i\int\widetilde{\boldsymbol{\Omega}}\,{}^{EF}\boldsymbol{e}_f dt\right). \tag{4}
$$

6 Experiments

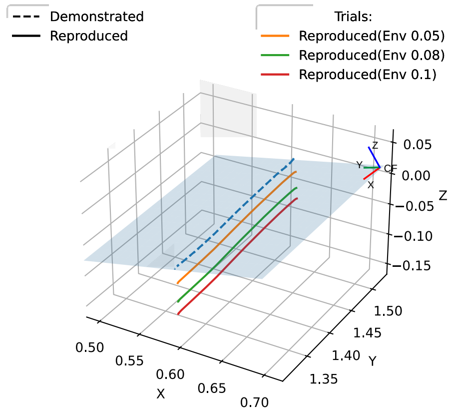
6.1 Experimental Setup

We evaluate the execution performance of both skills in a PyBullet simulation environment [29]. Beforehand, we collected demonstrations via kinesthetic teaching on a real DLR LWR-IV robot [30].

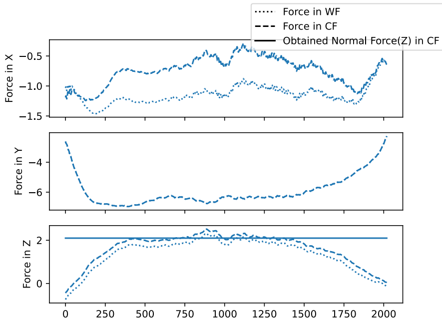
The extracted DMP parameters were used to generate the position, orientation, and force trajectories. Along with this, the selection matrix and the CF are passed to the hybrid impedance force controller. The controller outputs torque commands to the DLR LWR-IV robot model in the simulation environment.



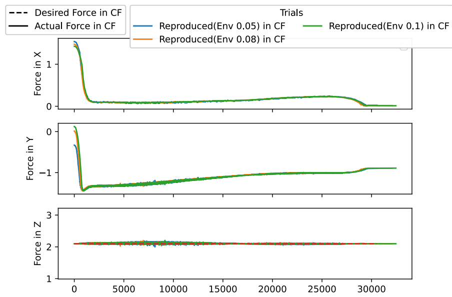
(a) Visualization of 3D Trajectories in PyBullet.



(b) 3D trajectory in world frame with extracted CF as plane.



(c) Force Profiles from demonstration (in WF and CF) and learned normal force f_z in CF.



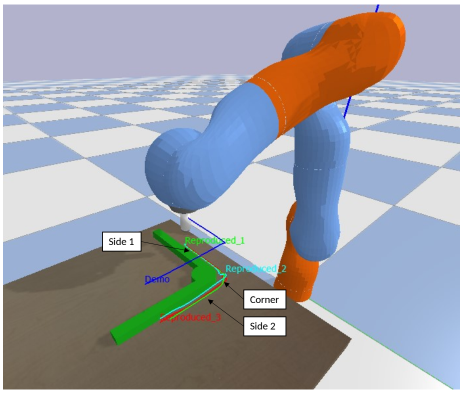
(d) Force Profiles from reproduction and learned normal force f_z in CF.

Fig. 3: Experimental results of *slide* skill.

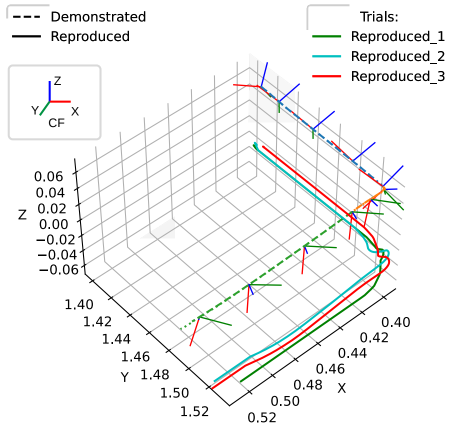
6.2 Results

Slide Figure 3 shows the reproduction results of the *slide* skill with a varying environment in each trial. Hereby, we shifted the surface height to prove that the desired force is always applied with the help of the skill. The reproduction results are compared with the demonstration trajectory.

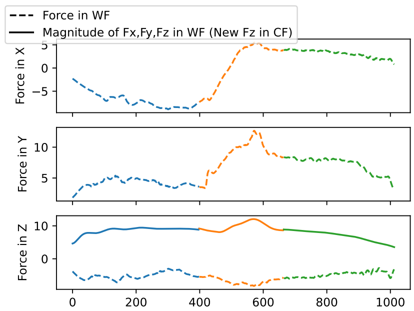
From the Fig. 3 it can be interpreted that, the *slide* skill adapted to the changes in the environment and reproduced the skill with the desired interaction behaviour in all the trials. Figure 3d shows that the *slide* skill is able to apply a constant normal force in z direction of the CF for the entire trajectory with comparable performance in all trials. Forces in x and y directions result from the friction along the plane and occur as expected.



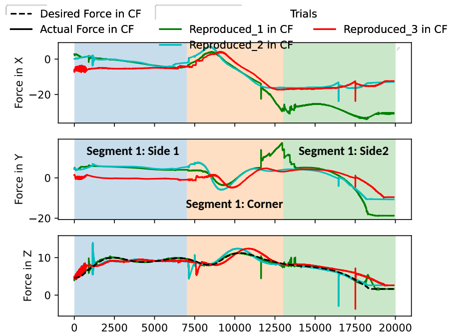
(a) Visualization of 3D Trajectories in PyBullet.



(b) 3D Trajectory in WF with extracted dynamic CF.



(c) Force Profiles from Demonstration in WF and Learned f_z in varying CF.



(d) Force Profiles from Reproduction comparison with Learned f_z in CF.

Fig. 4: Experimental results of *contour* skill.

Contour Figure 4 shows the reproduction results of the *contour* skill with a varying environment like lower table height and a displaced contouring object in each trial.

Figure 4 shows that the skill adapted to the environmental changes and reproduced the task with a similar interaction behaviour in all trials. In trial 1, environment is shifted 0.08m longitudinally in z direction of WF and for trials 2 and 3, environment is further shifted 0.02m laterally in y and x directions respectively. Fig 4b, 3D trajectories of demonstration and reproductions are shown along with the dynamically varying compliant frame that was extracted at each point. The Figure 4d shows that the *contour* skill is able to interact with the environment by applying a force profile in the varying CF at each points. Forces in x and y axis are naturally caused due to frictional forces in the direction of motion control.

7 Conclusions and Future Work

We proposed a learning framework for contact based skills that is capable of reproducing the desired behavior based on a single demonstration. Our automatic skill parameterization technique extracts parameter values that are used in a hybrid impedance force controller. This enables rigid transformations of compliant frames to account for motion and force constraints in specific axes. The proposed methodology for skill parameterization was implemented for two different contact skills and was evaluated in the PyBullet simulation environment.

As a future work, our framework could be extended to further interaction strategies with predefined constraints, where we proposed a number of examples in [5]. We plan to integrate also skill specific algorithms with multiple execution phases for more complex tasks, such as peg-in-hole.

References

1. S. Calinon and D. Lee, "Learning Control," *Humanoid Robotics: A Reference*, pp. 1–52, 2018.
2. A. Sena and M. Howard, "Quantifying teaching behavior in robot learning from demonstration," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 54–72, 2020.
3. M. R. Pedersen, L. Nalpantidis, R. S. Andersen, C. Schou, S. Bøgh, V. Krüger, and O. Madsen, "Robot skills for manufacturing: From concept to industrial deployment," *Robotics and Computer-Integrated Manufacturing*, vol. 37, pp. 282–291, feb 2016.
4. T. Eiband, M. Saveriano, and D. Lee, "Learning haptic exploration schemes for adaptive task execution," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, pp. 7048–7054, 2019.
5. T. Eiband and D. Lee, "Identification of common force-based robot skills from the human and robot perspective," in *Humanoid Robots (Humanoids), IEEE-RAS 20th International Conference on.* IEEE, 2021.
6. A. L. P. Ureche, K. Umezawa, Y. Nakamura, and A. Billard, "Task Parameterization Using Continuous Constraints Extracted from Human Demonstrations," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1458–1471, 2015.
7. J. Kober, M. Gienger, and J. J. Steil, "Learning movement primitives for force interaction tasks," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 3192–3199, 2015.
8. F. Steinmetz, A. Montebelli, and V. Kyrki, "Simultaneous kinesthetic teaching of positional and force requirements for sequential in-contact tasks," *IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-Decem, pp. 202–209, 2015.
9. S. Manschitz, M. Gienger, J. Kober, and J. Peters, "Learning sequential force interaction skills," *Robotics*, vol. 9, no. 2, 2020.
10. A. Conkey and T. Hermans, "Learning Task Constraints from Demonstration for Hybrid Force/Position Control," *arXiv*, pp. 162–169, 2018.
11. X. Gao, J. Ling, X. Xiao, and M. Li, "Learning Force-Relevant Skills from Human Demonstration," *Complexity*, vol. 2019, 2019.
12. J. J. Craig and M. H. Raibert, "A systematic method of hybrid position/force control of a manipulator," in *Proceedings - International Computer Software and Applications Conference.* IEEE Computer Society, 1979, pp. 446–451.

13. T. Luksch, M. Gienger, M. Muhlig, and T. Yoshiike, "Adaptive movement sequences and predictive decisions based on hierarchical dynamical systems," *IEEE International Conference on Intelligent Robots and Systems*, pp. 2082–2088, 2012.
14. O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.
15. A. G. Marin and R. Weitschat, "Unified impedance and hybrid force-position controller with kinestatic filtering," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem, pp. 3353–3359, 2016.
16. H. Lipkin and J. Duffy, "Hybrid twist and wrench control for a robotic manipulator," pp. 138–144, 1988.
17. A. Abbati-Marescotti, C. Bonivento, and C. Melchiorri, "On the invariance of the hybrid position/force control," *Journal of Intelligent and Robotic Systems*, vol. 3, no. 3, pp. 233–250, 1990.
18. J. F. S. Lin, M. Karg, and D. Kulić, "Movement Primitive Segmentation for Human Motion Modeling: A Framework for Analysis," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 3, pp. 325–339, 2016.
19. D. Kulić, W. Takano, and Y. Nakamura, "Combining automated on-line segmentation and incremental clustering for whole body motions," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2591–2598, 2008.
20. F. Zhou, S. Member, F. D. Torre, and J. K. Hodgins, "for Temporal Clustering of Human Motion," *Ieee Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. X, pp. 1–15, 2013.
21. C. Xiong, N. Shukla, W. Xiong, and S. C. Zhu, "Robot learning with a spatial, temporal, and causal and-or graph," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, pp. 2144–2151, 2016.
22. X. Xie, H. Liu, M. Edmonds, F. Gaol, S. Qi, Y. Zhu, B. Rothrock, and S. C. Zhu, "Unsupervised learning of hierarchical models for hand-object interactions," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4097–4102, 2018.
23. A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in neural information processing systems*, 2003, pp. 1547–1554.
24. A. Ude, B. Nemec, T. Petrič, and J. Morimoto, "Orientation in Cartesian space dynamic movement primitives," *Proceedings - IEEE International Conference on Robotics and Automation*, no. 4, pp. 2997–3004, 2014.
25. M. Tomas and H. John F, "Efficitently Building a Matrix to Rotate One Vector to Another," pp. 7–9, 1999. [Online]. Available: http://library1.nida.ac.th/termpaper6/sd/2554/19755.pdf
26. J. Lončarić, "Normal Forms of Stiffness and Compliance Matrices," *IEEE Journal on Robotics and Automation*, vol. 3, no. 6, pp. 567–572, 1987.
27. R. M. Murray, *A mathematical introduction to robotic manipulation.* CRC press, 2017.
28. R. M. Murray, Z. Li, and S. Shankar Sastry, *A mathematical introduction to robotic manipulation*, 2017.
29. E. Coumans and Y. Bai, "Pybullet, a Python Module for Physics Simulation for Games, Robotics and Machine Learning," 2016. [Online]. Available: https://pybullet.org
30. A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger, "The DLR lightweight robot: Design and control concepts for robots in human environments," *Industrial Robot*, vol. 34, no. 5, pp. 376–385, 2007.