Few-Shot Imitation Learning of Motion- and Contact-Based Tasks using Dynamic Movement Primitives

Master Thesis

Freek Bolding
s2019183

October 9, 2024

Supervisors:
dr. E.H.F. van Asseldonk
dr.ir. M. Vlutters
ir. A.H.G. Overbeek
dr.ir. D. Dresscher

Faculty of Engineering Technology
Department of Biomechanical Engineering
University of Twente

Preface

In this thesis, I present my work on an improved Learning from Demonstration framework for accessible programming of robots. The work concludes the graduation period of the Master of Science programme in Biomedical Engineering, Biorobotics track at the University of Twente.

As research is a collaborative effort, I could not have written this thesis without the help of my supervisors. I would like to thank them for providing insightful feedback and taking the time to supervise me. I am especially grateful for my daily supervisors, dr.ir. Mark Vlutters and ir. Alex Overbeek. Mark, I have really enjoyed my time at the Nakama robotics lab. I have appreciated our open discussions during sprint meetings. It was insightful to get a look into the daily life of an assistant professor, and the flat hierarchy of the group has made me feel comfortable and at home. Alex, your guidance and insights have been invaluable during this process. Our (long) discussions and brainstorms about learning from demonstration have always left me inspired and motivated. You have been a terrific supervisor.

Furthermore, I want to thank my friends and family for all their support and the shared times. Ultimately, I want to thank my partner Anna Florax. Her unwavering support has enabled me to keep going during the challenging moments of this thesis.

To the reader, I hope you enjoy going through my thesis. Should you at any point feel the desire to discuss robotic learning from demonstration, control, and/or 3D orientations; I am always happy to chat.

Freek Bolding, Enschede, October 2024

Summary

Traditionally, robots have primarily been deployed in industrial settings, where they carry out repetitive, pre-programmed tasks with high precision. To allow the transfer to unpredictable, unstructured environments populated with humans, the robots must be able to adapt to their environment. The Learning from Demonstration (LfD) paradigm facilitates this by allowing humans to (re-)program robots by demonstrating the robot how it should perform its tasks.

In this thesis, an LfD framework is designed and implemented with certain desirable properties. It allows generalization of demonstrated tasks based on changes in the environment, balances robot compliance and position tracking by using variable impedance control, and allows the robot to perform motion- and contact-based tasks. Additionally, corrections are given to prevent singularities and artefacts when modelling quaternion orientations with Dynamic Movement Primitives (DMP).

Demonstrations consisting of poses and forces are recorded. These are synchronized using Dynamic Time Warping with a custom cost measure to handle multivariate time series containing values with different units. Afterwards, a novel geometry-aware rolling window approach identifies time-varying Gaussian distributions from the data to obtain a single, mean demonstration and correspondent standard deviations. The mean demonstration is learned with a DMP, adapted to reproduce orientation trajectories without singularities or artefacts. Motion and force trajectories are reproduced simultaneously using a hybrid force-impedance controller. The impedance stiffness of the controller is modulated in real-time based on the identified variability between demonstrations.

The framework is validated on two real world tasks: watering a plant and drawing on a board. The experiments show that the framework possesses the desired properties. The variable impedance stiffness, driven by variance between the demonstrations, is able to track the pose accurately when required and be compliant when possible. The DMP allows for generalization to any target pose. The hybrid force-impedance controller enables simultaneous motion and force tracking.

The force tracking is less accurate than desired, and it is recommended to experiment with closed-loop force control for improved force tracking. Future work should focus on pairing the framework with a constraint frame identification method to reproduce motion and forces in a frame aligned with the environment's geometry. Additionally, a computer vision model with object segmentation and pose estimation may enable the framework to automatically adapt trajectories to changes in the environment.

Contents

List of Symbols and Abbreviations

The symbols and abbreviations are sorted in order of appearance.

Symbol	Definition
$SO(3)$	3-dimensional Special Orthogonal group; the 3D rotation group
$\mathbf{n}$	Rotation axis
θ	Rotation angle
$\mathbf{r}$	Rotation vector
$\mathbf{R}$	Rotation matrix
$\mathbf{q}$	(Unit) Quaternion
$\mathbb{S}^3$	3-sphere
$\boldsymbol{\xi}$	Sample of demonstration
$\boldsymbol{\xi}_\mu$	Inferred representative demonstration
$\mathcal{W}^\mathcal{P}$	Temporal alignment weighing factor for position distance
$\mathcal{W}^\mathcal{O}$	Temporal alignment weighing factor for orientation distance
$\mathcal{W}^\mathcal{F}$	Temporal alignment weighing factor for force distance
$\mathcal{W}^\mathcal{T}$	Temporal alignment weighing factor for torque distance
$\mathbf{q}$	Quaternion
$\overline{\mathbf{q}}$	Quaternion conjugate
$\bar{x}_w$	Weighted mean
W	Amount of elements in a window
w_i	Weight of element i; or weight of Gaussian basis function i (DMP)
σ_w	Weighted standard deviation
t_c	Time at the center of the window
$\bar{\mathbf{q}}_w$	Weighted quaternion average
$\sigma_w^\mathcal{O}$	Weighted orientation standard deviation
τ	Temporal scaling factor
x_0	Initial state/position
x_g	Goal state/position
ν	Time-scaled velocity
$f(s)$	Non-linear forcing term
s	Phase variable
K	Stiffness constant
D	Damping constant
$\psi(s)$	Gaussian basis function
c_i	Center of Gaussian basis function i
h_i	Width of Gaussian basis function i
α	Phase decay constant
$\mathbf{e}_\mathbf{q}$	Quaternion error, see rotation vector
$\mathbf{z}$	Time-scaled quaternion error velocity
θ_{thres}	Quaternion DMP singularity avoidance region threshold angle
$\mathbf{q}^c$	Generalized joint coordinates
$\mathbf{M}(\mathbf{q}^c)$	Inertia matrix
$\mathbf{C}(\mathbf{q}^c, \dot{\mathbf{q}}^c)$	Coriolis and centrifugal matrix
$\mathbf{g}(\mathbf{q}^c)$	Gravitational forces
$\boldsymbol{\tau}$	Applied joint torques
$\boldsymbol{\tau}^{ext}$	External joint torques
$\mathbf{F}^{ext}$	External wrench applied to robot
$\mathbf{J}(\mathbf{q}^c)$	Jacobian matrix
$\mathbf{F}_{imp}$	Impedance control desired end effector wrench
$\mathbf{K}_{imp}$	Impedance stiffness matrix
$\mathbf{D}_{imp}$	Impedance damping matrix

continues on next page

Symbol	Definition
$\boldsymbol{\omega}$	Angular velocity
$\mathbf{F}_f$	Force control desired end effector wrench
$\mathbf{S}$	Constraint axes selection matrix
$\mathbf{I}$	Identity matrix
$\boldsymbol{\tau}_{imo}$	Impedance control joint torques
$\boldsymbol{\tau}_f$	Force control joint torques
$\boldsymbol{\tau}_{comp}$	Gravity and Coriolis compensation joint torques
$\mathbf{K}_f^P$	Proportional force feedback control gain matrix
$\mathbf{K}_f^I$	Integral force feedback control gain matrix
K_{var}	Variable impedance stiffness
K_{min}	Minimum variable impedance stiffness
K_{max}	Maximum variable impedance stiffness
σ_{min}	Standard deviation threshold corresponding to K_{max}
σ_{max}	Standard deviation threshold corresponding to K_{min}
Ψ_{base}	Robot base frame
Ψ_{EE}	Robot end effector frame
Ψ_{F}	Robot force frame
$\lvert\epsilon\rvert$	Absolute error

Abbreviation	Definition
LfD	Learning from Demonstration
IL	Imitation Learning
PbD	Programming by Demonstration
DMP	Dynamic Movement Primitive
DoF	Degrees of Freedom
BC	Behavioral Cloning
IRL	Inverse Reinforcement Learning
ProMP	Probabilistic Movement Primitive
GMM	Gaussian Mixture Model
TP-GMM	Task-Parameterized Gaussian Mixture Model
SEDS	Stable Estimator of Dynamical Systems
GMR	Gaussian Mixture Regression
GP	Gaussian Process
EMG	Electomyography
HMM	Hidden Markov Model
EE	End Effector
MIFS	Mutual Information-based Feature Selection
HSMM	Hidden Semi-Markov Model
TP-HSMM	Task-Parametrized Hidden Semi-Markov Model
PD	Proportional Derivative
PI	Proportional Integral
DTW	Dynamic Time Warping
F/T	Force/Torque
CI	Confidence Interval
SD	Standard Deviation
BIC	Bayesian Information Criterion

1 General Introduction

1.1 Context

Traditionally, robots have primarily been deployed in industrial settings, where they carry out repetitive, pre-programmed tasks with high precision. These industrial robots perform tasks with great accuracy and speed in a structured, predictable environment [1]. However, advancements in computer vision, sensor technology, and artificial intelligence have led to growing interest in the development of service robots [2]. Unlike their industrial counterparts, service robots are designed to perform tasks that assist humans outside of industrial automation applications. These robots work in unstructured and dynamic environments, which are unpredictable and subject to changes. Furthermore, operation in human-populated settings introduces additional safety concerns. This increases programming complexity, as the robots are required to adapt online to changes in the environment. As it is not feasible to pre-program every scenario a robot might encounter, alternative programming techniques are required.

One promising solution is the Learn from Demonstration (LfD) paradigm [3]. Also known as Imitation Learning (IL) and Programming by Demonstration (PbD), LfD aims to make robotics accessible and adaptable by enabling operators to teach robots through demonstration instead of explicit programming. This requires no programming ability of the user and thus allows anyone to reprogram robots as needed. The user demonstrates the task by manually guiding the robot through the motion, letting it observe as the operator performs the task, or through teleoperation [4].

1.2 Problem Statement

1.2.1 Generalization

However, simply reproducing the demonstration is often not sufficient for successful task completion. Consider a pick-and-place task. Reproducing the exact demonstration will only be successful if the object is at the same location and needs to be moved to the same goal. Since it is infeasible to demonstrate every possible situation, the robot must adapt learned behaviour to unseen situations. This is known as generalization [3]. The robot's generalization capability is largely determined by the chosen method of task representation [5].

One widely used task encoding method is the Dynamic Movement Primitive (DMP) [6]. It encodes a demonstration in a stable dynamic system, which can be interpreted as a mass-spring-damper system. Each degree of freedom is encoded with its own dynamic system. The dynamic system ensures stable convergence from any initial position to any target position. By applying an external force, the system can be adapted to follow arbitrary movements. After this non-linear forcing function is learned, the initial and target positions can be adjusted to generate motions that match the original demonstration while reaching the desired positions. However, in its default form [6] it cannot learn from multiple demonstrations, nor is it able to correctly encode the non-Euclidean nature of Cartesian orientations [7].

By modification and extension of the DMP method, an LfD framework would be able to provide the robot with the ability to adapt and modulate its actions based on the state of the environment. These generalizations should extend to the reproduction of orientations, as rotation of the end effector is often crucial for adapting to surroundings and the reproduction of various tasks.

1.2.2 Compliant Robots

Additionally, it is desired for robots that coexist with humans and unstructured environments to show compliant behaviour [8]. While collisions are preferably avoided, this can not always be guaranteed. In the event of a collision, a compliant robot will be able to yield to external forces and allow itself to be moved in response to the impact.

Although robots can be built using compliant elements such as springs and elastic materials [9], rigid robots can be made to exhibit compliant behaviour through control. Impedance control, first introduced by Hogan [10], implicitly regulates the interaction forces by acting as a virtual spring-damper system with a desired stiffness. As the robot is displaced from its reference, it pushes back with a force that is dependent on the stiffness of the virtual spring. A high stiffness resembles position control, where the robot aims to achieve its desired position regardless of the required interaction forces. Conversely, a low stiffness will render the robot compliant, allowing it to be pushed and moved with ease. Intuitively, a lower stiffness reduces the chance and magnitude of damage or injury by decreasing interaction forces.

However, low impedance stiffness comes at the cost of tracking performance. Therefore, the robot would ideally be able to adapt its impedance stiffness in real-time, based on performance requirements. The robot should then be as compliant as possible, while still achieving adequate tracking performance.

1.2.3 Contact Tasks

Finally, many useful tasks that robots may carry out require exerting force on the environment, such as polishing, wiping, ironing, and assembly. While early research primarily focused on reproducing the kinematic portion of demonstrated tasks, recent work has aimed to equip robots with the ability to learn force-based tasks [3]. This ability greatly enhance the robot's versatility in what tasks it can reproduce, but it comes at the cost of controller complexity. The controller must be capable of reproducing motion and forces simultaneously, taking in consideration the geometry constraints of the environment.

1.3 Scope of Thesis

The goal of this thesis is to develop an LfD framework that addresses the challenges discussed above (Section 1.2.1-1.2.3). The framework should provide an intuitive manner for non-expert operators to program the robots through the demonstration of tasks. The framework must be able to adapt learned policies to changes in the environment. It must be able to infer a balance of tracking performance and compliance from the provided demonstrations. Finally, it must be capable of reproducing force-relevant contact tasks. The desired end result is an implemented and validated LfD framework that possesses these properties.

1.4 Contribution of Thesis

Earlier work has focused on solving one or more of the challenges described, as will be discussed in the Related Work section of the research paper. However, to the best of the author's knowledge, no single framework has addressed these issues simultaneously in an LfD framework that can adapt trajectories to new situations, automatically balance tracking performance and robot compliance by inferring impedance stiffness from variance in demonstrations, and reproduce both motion and contact tasks. Furthermore, it proposes a novel method of obtaining a smooth average and standard deviation for demonstrations containing end-effector pose and wrench information that takes the non-Euclidean nature of the rotation group into account. Finally, it proposes an adaptation of the orientation DMP proposed by Koutras et al. [11] and gives guidance on preventing oscillating behaviour and avoiding singularities when using orientation DMPs.

2 Background

2.1 Demonstration Interface

In Learning from Demonstration, three different methods for providing demonstrations to the robot are distinguished [3], [4]: 1) Passive Observation, 2) Teleoperation, and 3) Kinesthetic Teaching. Each of these interfaces has its own characteristics.

In Passive Observation, the operator demonstrates the task without using the robot. The demonstration is recorded using motion trackers or a vision system. The main drawback of this method is the need to map the sensory input and actions of the (human) demonstrator to the (robot) learner. This is known as the correspondence problem [12].

Tele-operation requires the demonstrator to control the robot through an external interface and can be either unilateral or bilateral, depending on whether the controlled robot sends motion and force information back to the operator [13]. It allows an operator to control robots in environments that are unsafe or unreachable to humans. However, there are still challenges in teleoperation, such as time delays in multimodal feedback, synchronisation control, and dealing with differing configurations on the master and slave side [14].

Kinesthetic teaching allows the operator to physically move the robot's end-effector through the desired motions of the task. This enables the operator to directly feel exerted forces and robot body limitations [15]. While it allows for straightforward recording of the demonstration, the quality and smoothness of demonstrations are impacted by the demonstrator's proficiency and the robot's impedance.

2.2 3D Orientations

One of the main goals of this framework is the generalization of the demonstrated task to changes in the environment. This requires adjusting the end effector orientation trajectories in a mathematically correct manner, taking into consideration the intricacies of the 3D orientation space that is denoted by $SO(3)$. As will become apparent in the Related Work section of the research paper, many LfD works neglect to extend their proposed method to take the non-Euclidean nature of $SO(3)$ into account. This section aims to provide the reader with some background in 3D orientations.

Position and orientation describe how a rigid body is placed in space, referred to as its pose. The position describes a translation from a reference placement to its current placement, while the orientation describes a rotation from a reference placement to its current placement. Unlike translations, rotations do not commute, i.e. the order of applied rotations affects the final orientation. Additionally, rotating a rigid body a full circle results in the same orientation, while any translation will never result in the same position. These differences lead to different mathematical representations. Positions can be described with a vector in 3D space, while orientations require more complex parameterizations. Multiple parameterizations of 3D orientations exist: Euler angles, axis-angle representation, rotation vectors, unit quaternions, and rotation matrices [16].

Euler angles use three sequential rotations around specific axes to represent the orientation. Depending on the chosen convention, the rotation axes may rotate with the body (intrinsic rotations) or be fixed (extrinsic rotations). Although Euler angles are an intuitive and minimal representation of rotations, i.e. using only three parameters, they suffer from various drawbacks: Gimbal lock occurs when multiple rotation axes align. Discontinuities may arise when the orientation passes certain boundaries, e.g. π rad, and the Euler angles suddenly jump. Additionally, if the range of the angles is unlimited to prevent discontinuities, redundancies occur where multiple combinations of angles can represent the same orientation.

The axis-angle representation is related to Euler's rotation theorem, which states that any orientation can be achieved by a single rotation around a fixed axis. It represents orientations with a single rotation angle $\theta \in \mathbb{R}$ around a unit rotation axis $\mathbf{n} \in \mathbb{R}^3$. Any orientation can be represented when θ is in the range of $[0, \pi]$. This has singularities at $\theta = 0$, where any axis $\mathbf{n}$ is valid, and at $\theta = \pi$, where each axis and its negation represent the same orientation.

The rotation vector $\mathbf{r} \in \mathbb{R}^3$ is a more concise form of the axis-angle representation and is defined as $\mathbf{r} = \theta\mathbf{n}$. It does not suffer from the singularity at $\theta = 0$. However, it does result in singularities when $\theta = 2\pi k$, with k being a non-zero integer, should these values be in the permissible range of θ. In such scenarios, any rotation axis $\mathbf{n}$ is valid and $\mathbf{r}$ may exhibit rapid changes near this singularity. This singularity returns in the quaternion DMP formulation and will be discussed in the research paper.

The rotation matrix $\mathbf{R} \in SO(3)$ is a widely used tool to represent and apply rotations. Rotations can be applied using straightforward matrix multiplication. They provide a smooth and continuous representation of 3D orientations. It is a non-minimal parametrization, consisting of nine parameters. Its orthogonality constraint removes six degrees of freedom, leaving three degrees of freedom to represent 3D rotation. While applying rotations is straightforward, its nine parameters make representing orientations less convenient.

Finally, unit quaternions $\mathbf{q} \in \mathbb{S}^3$ can be used to represent orientations. They can be written as $\mathbf{q} = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$, where $w, x, y, z \in \mathbb{R}$ and $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are imaginary basis vectors. The four parameters are constrained to have a unit norm, resulting in three degrees of freedom. It provides a double cover of $SO(3)$, with $\mathbf{q}$ and $-\mathbf{q}$ describing the same orientation. Both $SO(3)$ and S^3 are Riemannian manifolds, which locally resemble Euclidean space [17]. Unit quaternions can be converted to rotation vectors by taking the quaternion logarithm and scaling the result by a factor of 2. See Appendix A of the research paper for a more detailed overview of quaternions and their operations. Unit quaternions provide a smooth and non-singular way to encode 3D orientations, which is more compact than rotation matrices. Because of these properties, unit quaternions are selected as the orientation parametrization in this thesis.

3 Research Paper

UNIVERSITY OF TWENTE. NAKAMA ROBOTICS LAB

Graduation Committee
dr. E.H.F. van Asseldonk
dr.ir. M. Vlutters
ir. A.H.G. Overbeek
dr.ir. D. Dresscher

PAPER

Few-Shot Imitation Learning of Motion- and Contact-Based Tasks using Dynamic Movement Primitives

Freek Bolding

Abstract

Robots are diversifying into sectors beyond manufacturing, such as healthcare and agriculture, offering enhanced productivity. However, operating in dynamic, human-populated environments requires adaptability and compliance. Imitation learning simplifies the reprogramming of robots by enabling operators to demonstrate desired behavior. By encoding the demonstrations in dynamic movement primitives (DMPs), task reproductions can be generalized to adapt to changes in the environment. This paper proposes a geometry-aware LfD framework based on DMPs that allows for generalization, variable compliance, and the reproduction of motion-based and contact-based tasks. The variable compliance is driven by variability between the demonstrations, pursuing an automatic balance between desired compliance and tracking performance. Demonstrations, which include motion and force data, are synchronized and processed using a novel weighted rolling window approach that identifies their mean and standard deviations. The mean demonstration is learned with the dynamic movement primitive. The hybrid force-impedance controller enables simultaneous reproduction of motion and force in perpendicular directions. The framework was validated on real-world tasks, demonstrating its practicality. In addition to the proposed framework, contributions include recommended corrections when encoding quaternion trajectories using DMPs to prevent singularities and artefacts.

Key words: Learning from Demonstration, Variable impedance, Dynamic Movement Primitives, Hybrid Force-Impedance Control

1 Introduction

Robots are increasingly being adopted outside of manufacturing applications, such as healthcare, security, agriculture and home assistance [1]. They offer several potential benefits, including increased productivity, consistent quality of work, and lower workforce expenses [2]. However, working in unstructured, dynamic, and potentially human-populated environments brings additional challenges compared to the static, predictable factory setting. For one, the robot must be able to learn from- and adapt to its environment.

Imitation learning (IL), or Learning from Demonstration (LfD), aims to simplify reprogramming of the robot by allowing end-users to demonstrate desired behavior [3]. The robot learns a policy, cost/reward function, or task plan to encode and reproduce the demonstrated task [4]. Policy-learning methods aim to learn a direct function, i.e. the policy, to generate desired behavior [4]. By adapting the learned policy based on changes in the environment, the robot is able to generalize task reproduction in unseen situations [3].

The Dynamic Movement Primitive (DMP), introduced by IJspeert et al. [5], is a widely used LfD method to allow this generalization. The motions are encoded in a spring-damper system, whose initial and goal positions can be adjusted depending on the environment. However, its encoding of independent DoF is unsuitable for correctly learning Cartesian orientation trajectories, which is required for the reproduction of any task involving rotations of the end-effector. This has been addressed with the orientation DMP [6, 7, 8], but these can introduce potential singularities and artefacts that should be avoided, as will be discussed later. Additionally, DMPs can only learn from a single demonstration by default, hindering the integration of information based on multiple demonstrations.

As the robot reproduces a movement through the unstructured environment, it should exhibit compliant behavior [9]. This reduces damage to the environment in the event of a collision, as the robot is able to absorb impact energy by yielding to external forces. A popular approach to regulate robot compliance is impedance control [10]. It resembles a virtual spring-damper between the robot and its environment. The stiffness of the virtual spring dictates a balance between the safety and tracking performance of the robot, where increased stiffness increases the robot's reference tracking accuracy while decreased stiffness increases the robot's compliance [11].

Preferably, the impedance stiffness is set based on the tracking performance required, such that compliance can be high when less precision is required. As suggested by Enayati et al. [12], this required precision may be inferred

Key Points

- A geometry-aware Learning from Demonstration framework is proposed that combines force control with variable impedance control for the reproduction of motion- and contact-based tasks.
- The variable stiffness of the impedance controller is driven by the variance between demonstrations.
- The proposed DMP allows for stable trajectories between any initial- and goal pose within the robot's workspace, which includes singularity-free generalization of three-dimensional orientations.
- The framework is validated on two real-world tasks.

from the provided demonstrations of a task. When certain parts of a task are consistent among demonstrations, it can imply that precise tracking of the movement is required. Conversely, if there is more variation between the demonstrations, precise tracking may be relaxed in exchange for increased robot compliance.

Finally, while early LfD research mainly focused on pure kinematic reproduction, many practical tasks require physical interaction with the environment. Tasks such as polishing, wiping, ironing, and assembly require controlled forces to be exerted while moving parallel to the surface. To perform these tasks, it is essential that the robot reproduces demonstrated forces and motion simultaneously. This is possible using a hybrid position and force control scheme [13], which splits the DoF into a position- and force-controlled subset. By controlling the position with an impedance controller, this hybrid controller would be able to exert forces as desired while moving in a compliant manner and therefore reproduce demonstrations of contact-based tasks.

This paper proposes an LfD framework to address the described challenges of generalization, variable compliance and contact tasks simultaneously, based on the DMP method. Recorded demonstrations are processed to obtain a single, representative demonstration and the variance between them. This demonstration is learned by a DMP to allow for generalization based on task parameters. During reproduction, the hybrid-force impedance controller tracks independent motion and force profiles in a compliant manner. The framework is 'geometry-aware', meaning that the geometric constraints of three-dimensional orientations are taken into account.

The main contributions of this work and the proposed LfD framework are:

- A modular framework that combines the generalization capabilities of DMP, the safety-performance balance of variable impedance control, and contact-based task learning.
- A geometry-aware method for identifying time-varying, smooth, deterministic Gaussian distributions over demonstrations using weighted rolling window mean and standard deviations.
- An adaptation of orientation DMPs, with specified corrections to prevent singularities and artefacts.

In this study, the proposed framework is implemented and validated on two real-world tasks. The paper is structured as follows: Section 2 examines the current state-of-the-art in learning from demonstration and modern efforts in variable stiffness and contact-based task learning. Section 3 gives a description of the framework and its components. Section 4 shows the effect of the adapted orientation DMP and provides an experimental evaluation of the framework. Section 5 discusses interpretations, limitations and possible extensions. Finally, Section 6 will provide a conclusive remarks.

2 Related Work

2.1 Imitation Learning Methods

Two major classes of IL can be distinguished: Behavioral Cloning (BC) and Inverse Reinforcement Learning (IRL) [14]. BC aims to find a policy that directly maps from the current state or context to a control input. It tries to mimic the expert demonstrations without inferring the reasoning behind the actions.

Instead, IRL tries to recover a reward function from the expert demonstrations and then searches for a policy that maximizes the expected return. This assumes that the demonstrations are optimal with respect to the reward function. The main advantage of IRL compared to BC is a more succinct representation of the task [15]. Aditionnaly, by learning the underlying reward function, it could potentially adapt better to changed environmental dynamics. However, the problem is ill-posed; there exist infinite reward functions that result in the same optimal policy [16]. Solving it can have substantial computational costs and require several demonstrations, which may not always be present. In contrast, BC generally converges faster to an optimal policy [15].

Within BC, several encoding methods have been proposed. One widely used method is the previously mentioned DMP [5], which will be discussed extensively in Section 3.4. Other notable encoding methods include the Probabilistic Movement Primitive (ProMP) [17], Gaussian Mixture Model (GMM) [18], Task-Parametrized Gaussian Mixture Model (TP-GMM) [19], and Stable Estimator of Dynamical Systems (SEDS) [20].

ProMP is a probabilistic alternative to DMP, capable of learning from multiple demonstrations. It parametrizes a trajectory distribution from the demonstrations using a Hierarchical Bayesian Model, which enables multiple desirable properties such as encoding variance of movements and coupling between degrees of freedom [17]. However, it cannot be used in one-shot learning and cannot generalize to situations outside the distribution of demonstrations. Furthermore, it is unclear how to extend the framework to correctly encode Cartesian orientations.

GMM aims to model the underlying distribution of demonstrated movements by representing it as a weighted combination of multiple Gaussian components. The Gaussians are fit using the Expectation-Maximisation algorithm [18]. Trajectories can be retrieved using Gaussian Mixture Regression (GMR). GMM is a broad probabilistic model that can model various relationships. One straightforward example is a time-parametrized trajectory, which captures how positions change over time. However, this does not support generalization to new situations as the GMM cannot infer or extrapolate beyond the data it was trained on. Consequently, TP-GMMs were introduced [19]. It fits multiple GMMs in different reference frames simultaneously. By moving the reference frame(s), motion can be adapted to new situations.

SEDS uses GMMs to learn stable first-order dynamical systems. It uses the Lyapunov Stability Theorem to constrain the learned parameters of the GMM. The result is a dynamic system that guarantees stable trajectories while resembling the demonstration as close as possible under the constraints. However, a recent comparison of motion encoding frameworks showed that generalized trajectories were not guaranteed to converge to the target, resulting in cases with an end-point error larger than 10cm [21]. Such errors were not found in DMPs, GMMs or TP-GMMs. The study found the DMP model to be the most efficient with respect to hyperparameter tuning, encoding, and generalisation accuracy [21].

2.2 Variable Impedance Control

Several recent developments have implemented variable impedance control laws in an LfD framework. Abu-Dakka et al. [22] tried to estimate a stiffness profile from the collected demonstrations. This was done by recording Cartesian positions, velocities, accelerations and associated forces. The time-varying stiffnesses were obtained by an interaction model and encoded in a GMM/GMR. This allows the framework to adjust impedance stiffness based on perceived forces during task execution. However, they do not use robot orientations in their framework.

Enayati et al. [12] uses GMM/GMR to encode position and interaction force, and the covariance of the Gaussians to set impedance stiffness. The variance is obtained using eigendecomposition of the covariance matrix of each Gaussian. The impedance stiffness is therefore driven by the variability of demonstrations, where high variability decreases stiffness as position tracking could be relaxed. Although their framework uses the concept of driving impedance stiffness by demonstration variance, it does not include encoding of orientations and torques and the used classic GMM has limited generalization capabilities.

The idea of estimating stiffness profiles from the variance of demonstrations is also used by Arduengo et al. [23] and Zhang et al. [24]. Arduengo et al. [23] proposed a variable impedance stiffness LfD framework based on heteroscedastic Gaussian Processes (GP). In contrast to standard GPs, heteroscedastic GPs can model time-varying uncertainty. This is used to encode task variability that modulates the robot's stiffness. To deal with orientations, they use rotation vectors. They have a one-to-one mapping by limiting the rotation angle to π rad. However, by doing this they introduce discontinuities in the orientation description. For example, taking the mean of two rotations around a certain axis with rotation angles of $\pi - \epsilon$ rad and $-\pi + \epsilon$ rad would result in an angle of 0 rad, instead of $\pm\pi$ rad. Zhang et al. [24] use rotation vectors as well, which are encoded in GMM/GMR and then fed to a DMP model. However, this leads to the same orientation issues.

Wu et al. [25] uses EMG signals to estimate the arm impedance of the human demonstrator. The task trajectories were encoded in either a GMM or SEDS model. The GMM encodes the estimated impedance stiffness of the arm and trajectory information without generalization ability. SEDS is used for its stable generalizing to a target point, but it does not encode impedance stiffness. Depending on the phase of the reproduction, they switch between GMM and SEDS encoding. However, this requires the system to decide when to switch the encoding method based on which benefit is required. Additionally, they do not encode end-effector orientations.

2.3 Contact-Based Task Learning

While most early LfD methods focused on learning just the kinematic portion of demonstrations, recent work has investigated learning force-relevant tasks [3]. Contrary to motion-based tasks that move unconstrained in free space, contact-based tasks such as polishing, ironing, drilling, and assembling require forces to be exerted on the environment. This requires a controller capable of reproducing motion and force simultaneously.

Kormushev et al. [26] have used a DMP to store motion and force profiles. Demonstration recording occurred in two phases; first, the positional profile was recorded through kinesthetic teaching. Afterwards, the operator used teleoperation with a haptic device to command forces while the robot reproduced the motion profile with impedance control. Finally, the motion and force profiles were reproduced with simultaneous impedance and force control. However, this can lead to conflicting actions between the two controllers. For instance, if the robot learned to exert force on a table, and during reproduction the table is lower than in the original demonstration, the impedance controller will attempt to push upwards towards the original reference, resulting in a lower exerted force than measured. Additionally, they do not include orientations and torques in their task encoding.

Steinmetz et al. [27] store motion and force profiles in a DMP as well. While they do include orientations as quaternions in their DMP, they model each component independently, only normalizing them afterwards for reproductions, not taking their interdependent nature into account. They use separate Cartesian motion and force controllers, switching between them based on thresholds of desired position and desired force. Additionally, they only allowed forces in the z-direction, reducing the framework's versatility.

Rozo et al. [28] proposed a framework that learns manipulation skills primarily from force-torque data. During demonstrations, the robot is controlled through teleoperation with a haptic device. The GMM is extended with a Hidden Markov Model (HMM), where the hidden state of the HMM determines the used Gaussian mixture. They choose the encoded variables based on the task. For example, the training samples of a ball-in-box task contain EE torques and angular velocities, while the training samples of a pouring task contain EE torques and robot joint positions. They use a machine learning algorithm (MIFS-U) to select which input variables to prioritize.

The earlier described variable impedance framework of Enayati et al. [12] also includes explicit force control. Similar to [26], they regulate force and position in all directions simultaneously. The force feedback controller tries to match the force profile, even during pure motion tasks. The variable controller gains help emphasize force or position regulation. However, again similar to [26], this can lead to the force control and impedance control parts of the controller relaying contradicting control effort.

Le et al. [29] record end effector poses and external wrenches during demonstrations. These are reproduced with impedance control, where the stiffness and attractor reference are set such that the robot reproduces both positions and forces simultaneously. It uses a Task-Parametrized Hidden Semi-Markov Model (TP-HSMM) that consists of a Hidden Semi-Markov Model (HSMM) with underlying TP-GMMs as a component of the HSMM. A Riemannian manifold formulation is used for end-effector orientations to ensure results are geometrically valid. While combining force and position in a single spring attractor demo allows for simultaneous control, it makes independent generalization of position or force difficult.

Finally, Wang et al. [30] combine a DMP with hybrid force-motion control. The measured position and force profiles are learned as separate DoFs in the DMP. The DMP allows them to generalize the position and force profiles as desired. The controller uses PD position control and PI force control simultaneously. However, this controller does

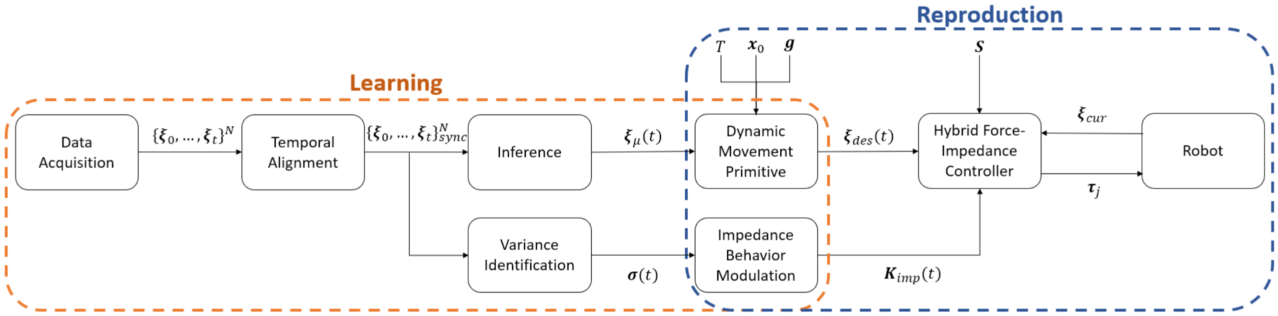

Figure 1. The framework with its components. Demonstrations containing EE poses and wrenches are recorded in Data Acquisition (Section 3.1). These are synchronized using dynamic time warping in Temporal Alignment (Section 3.2). The inference block finds a mean demonstration, while the variance identification finds the standard deviation (Section 3.3). The DMP uses the mean demonstration and task variables to generalize the learned policy to unseen situations (Section 3.4). The hybrid force-impedance controller allows the robot to follow the desired Cartesian motion and force profiles by controlling the robot's joint torques Section 3.5). The standard deviation is used to modulate the impedance behavior of this controller (Section 3.6).

not result in explicit compliant behavior. It is also unclear in which reference frame the motion and force are measured. Finally, they do not include orientations in their framework.

3 Framework

In this chapter, the proposed framework is discussed. It combines properties of the discussed related work to include task generalization, variable compliance, orientation encoding, and the capability to reproduce both motion- and contact-based tasks. The complete framework can be seen in Figure 1. It shows the pipeline of how demonstrations are processed and learned, from data acquisition to reproduction on the robot. The components are discussed in order in this section. When only a single demonstration of the task is available, temporal alignment, inference, and variance identification are skipped. The single demonstration is learned immediately with the Dynamic Movement Primitive, and the impedance controller has constant stiffness as there is no variance between demonstrations to drive the variable stiffness.

3.1 Data Acquisition

During data acquisition, demonstrations are recorded. The proposed framework is demonstration-interface agnostic, requiring robot end-effector poses and wrenches that could be provided with either passive observation, teleoperation, or kinesthetic teaching. When a task is recorded, the end-effector poses, external wrenches, and their timestamps are stored. Orientations of the end-effector are stored as quaternions. The end-effector poses are defined in the base frame of the robot. The external wrenches are defined in a frame located at the end-effector's desired contact point, but rotated such that its axes are always aligned with the robot's base frame.

3.2 Temporal Alignment of Demonstrations

When multiple demonstrations of the same task are given, there will be temporal variation as one demonstration may have started earlier, executed one part faster, or even paused midway through. By aligning the demonstrations in time, key points of the demonstrations occur at similar time points. Without temporal alignment, differences in timing would appear as variations in the task. It is therefore necessary that all demonstrations are synchronized.

One widely used temporal alignment algorithm is Dynamic Time Warping (DTW) [31]. It uses Dynamic Programming to find a minimal cost alignment between time series. Given two time series $\mathbf{A} := (a_1, \dots, a_N)$ of length N and $\mathbf{B} := (b_1, \dots, b_M)$ of length M, DTW finds warping path $\mathbf{P} = (\boldsymbol{p}_1, \dots, \boldsymbol{p}_L)$ of length L with $\boldsymbol{p}_l = (n_l, m_l)$ that minimizes the cost between $\mathbf{A}$ and $\mathbf{B}$ subject to three conditions:

i. *Boundary condition:* The first and last elements of both time series are aligned, i.e. $\boldsymbol{p}_1 = (1, 1)$ and $\boldsymbol{p}_L = (N, M)$
ii. *Monotonicity condition:* Any subsequent element of the warping path can not contain an earlier element of either sequence, i.e. $n_1 \leq n_2 \leq \dots \leq n_L$ and $m_1 \leq m_2 \leq \dots \leq m_L$
iii. *Step size condition:* A subsequent element of the warping path either increases either n, m, or both by one, i.e. $\boldsymbol{p}_{l+1} - \boldsymbol{p}_l \in \{(1,0), (0,1), (1,1)\}$ for $l \in [1 : L-1]$

These conditions ensure that for every element a_n there is at least one assigned element b_m and vice versa. The cost, or distance, between elements a_n and b_m is calculated with a cost measure $c(a_n, b_m) \to \mathbb{R}_{\geq 0}$. When demonstrations involve multiple physical quantities, such as position, orientation, force and torque, $\mathbf{A}$ and $\mathbf{B}$ are multivariate time series whose elements $\boldsymbol{a}_n, \boldsymbol{b}_m$ contain values in different units, i.e. m, rad, N, and Nm, whose distances cannot simply be added together. To tackle this, the distance between two elements is calculated separately for each physical quantity. These distances are then multiplied by unit-specific weights to make them unitless and compensate for differences in scale. Additionally, the weights allow the user to prioritize which dimensions to align. These weights are hyperparameters to be set by the user. This leads to the following proposed cost measure:

$$
\begin{aligned}
c(\boldsymbol{a}_n, \boldsymbol{b}_m) = &\mathcal{W}^{\mathcal{P}} d_{lin}(\boldsymbol{a}_n^{\mathcal{P}}, \boldsymbol{b}_m^{\mathcal{P}}) + \mathcal{W}^{\mathcal{O}} d_{rot}(\boldsymbol{a}_n^{\mathcal{O}}, \boldsymbol{b}_m^{\mathcal{O}}) + \\
&\mathcal{W}^{\mathcal{F}} d_{lin}(\boldsymbol{a}_n^{\mathcal{F}}, \boldsymbol{b}_m^{\mathcal{F}}) + \mathcal{W}^{\mathcal{T}} d_{lin}(\boldsymbol{a}_n^{\mathcal{T}}, \boldsymbol{b}_m^{\mathcal{T}}),
\end{aligned} \tag{1}
$$

where $^{\mathcal{P}}$, $^{\mathcal{O}}$, $^{\mathcal{F}}$, $^{\mathcal{T}}$ refer to the position, orientation, force, and torque portion of one element respectively; $\mathcal{W}$ denotes the weighting factor of a particular unit; and distance functions d_{lin} and d_{rot} are defined as

$$d_{lin}(\boldsymbol{x}, \boldsymbol{y}) = ||\boldsymbol{x} - \boldsymbol{y}|| \tag{2}$$

$$d_{rot}(\mathbf{q}_x, \mathbf{q}_y) = ||2 \log(\mathbf{q}_x * \overline{\mathbf{q}_y})||, \tag{3}$$

where $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3$ denote Euclidean data such as positions, forces, and torques and $\mathbf{q}_x, \mathbf{q}_y \in \mathbb{S}^3$ denote unit quaternions representing orientations. The rotation distance function describes the rotation angle between two orientations. This is equal to the length of the arc between the two points on $\mathbb{S}^3$, formed by the set of all unit quaternions [32]. This will be discussed further in Section 3.4.3, or see Appendix A for quaternion definitions and operations. To make sure the smallest arc, known as the geodesic, is returned, the sign of $\mathbf{q}_x * \overline{\mathbf{q}_y}$ is flipped if its real part is negative.

For each pair of elements in **A** and **B**, the cost $c(\mathbf{a}_n, \mathbf{b}_m)$ is calculated and stored in cost matrix $\mathbf{C} \in \mathbb{R}^{N \times M}$. Dynamic programming finds the warping path with the lowest cost. The time vector of **A** and corresponding aligned elements of **B** are taken to obtain a demonstration $\mathbf{B}'$ that is aligned to **A**. Should multiple elements of **B** be aligned to the same time element of **A**, the mean between those elements is taken. All demonstrations of the task are aligned to a single base demonstration in this manner.

3.3 Inference and Variance Identification

After the temporal alignment of all demonstrations, it is required to obtain one single representative demonstration of the task since the DMP only encodes a single demonstration. Additionally, to set the variable stiffness of the impedance controller, a measure of the spread between the dimensions across the various dimensions must be found to drive the variable impedance stiffness. The component finding the single representative demonstration is called inference, as it infers one representative demonstration from the distribution of individual demonstrations. The inference method should be aware of the non-Euclidean nature of $SO(3)$. Other commonly used methods such as GMR[18] and GPR[33] assume a Euclidean structure by default [34]. While this works for joint angles or Cartesian translations, it cannot correctly handle rotations of $SO(3)$.

For this reason, a geometry-aware weighted rolling window mean and standard deviation method is proposed. The moving average calculates a mean for each dimension separately. For Euclidean data, such as positions, forces and torques, the following weighted mean and standard deviation are used:

$$\bar{x}_W = \frac{\sum_{i=1}^{W} w_i x_i}{\sum_{i=1}^{W} w_i} \tag{4}$$

$$\sigma_W = \sqrt{\frac{\sum_{i=1}^{W} w_i (x_i - \bar{x}_W)^2}{\left(\sum_{i=1}^{W} w_i\right) - 1}} \tag{5}$$

where W denotes the elements in the window and w_i denotes the weight of element i. If the sum of weights is equal or smaller than 1, the bias correction of -1 in the denominator should be removed to prevent division by zero or complex standard deviations. However, this indicates that a longer window should be chosen to increase the amount of samples in the window. In other cases, this bias correction, known as Bessel's correction [35], corrects for the fact that the sample average is used instead of the (unknown) population mean.

The element weights are set to be negatively proportional to the distance from the center of the moving window. The weights are calculated using a 4^{th}-order polynomial that ensures a smooth transition of the weights without sharp changes or discontinuities. This is achieved by solving the polynomial for the boundary conditions where the weight is equal to 1 at the center of the window, equal to 0 at the edge of the window, and its first derivatives at the center and edge of the window are equal to 0. The function after solving the boundary condition problem is given by:

$$w_i = \frac{2}{\left(\frac{T}{2}\right)^3}(|t_c - t_i|)^3 - \frac{3}{\left(\frac{T}{2}\right)^2}(|t_c - t_i|)^2 + 1, \tag{6}$$

where T denotes the length of the rolling window, t_c denotes the time at the center of the window and t_i denotes the time of element i in the window. The smooth weighing function guarantees that the identified mean and standard deviation are smooth as well. For clarification, a plot of the function of (6) is shown in Figure 2.

The weighted average of multiple quaternions is obtained using the eigendecomposition method proposed by Markley et al. [36], see Appendix A. To the best of the authors' knowledge, there exists no universally accepted method for taking the standard deviation or any other measure of the spread of a collection of quaternions. To control the impedance stiffness of each rotation axis separately, a standard deviation per rotation axis should be determined. Therefore, a method is proposed based on the axis-angle representation of relative quaternions. First, the weighted quaternion average of that window is calculated. Then, for all quaternions $\mathbf{q}_i$ in the window W, the rotation vector $\mathbf{r}_i \in \mathbb{R}^3$ between $\bar{\mathbf{q}}_W$ and $\mathbf{q}_i$ is calculated:

$$\mathbf{r_i} = \theta_i \mathbf{n}_i = 2\log(\bar{\mathbf{q}}_W * \overline{\mathbf{q}_i}), \tag{7}$$

where, similar to (3), the sign of $\bar{\mathbf{q}}_W * \overline{\mathbf{q}_i}$ should be flipped if its real part is negative. Since $\mathbf{r_i}$ is the product of the rotation angle θ_i with rotation axis $\mathbf{n}_i$, it measures a rotational distance from the average orientation in each Cartesian axis. Notably, it is equal to the angular velocity required to rotate from $\mathbf{q}_i$ to $\bar{\mathbf{q}}_W$ in unit time. The weighted orientation standard deviation $\sigma_W^{\mathcal{O}} \in \mathbb{R}^3$ is then obtained with

$$\sigma_W^{\mathcal{O}} = \left(\frac{\sum_{i=1}^{N} \mathbf{r_i} \circ \mathbf{r_i}}{\left(\sum_{i=1}^{W} w_i\right) - 1} \right)^{\circ \frac{1}{2}}, \tag{8}$$

where $\circ$ denotes the element-wise product and $^{\circ\frac{1}{2}}$ denotes the element-wise root.

The weighted average and weighted standard deviation for all dimensions for all time points t_c are calculated, which results in representative demonstration $\xi_\mu(t)$, containing poses and wrenches, and standard deviations $\sigma(t)$. These are then learned with the DMP.

3.4 Dynamic Movement Primitive

3.4.1 *Original Formulation*

Dynamic movement primitives (DMPs) were proposed by IJspeert et al. as an approach to model movements with a system of differential equations possessing desirable properties [5]. Different formulations allow for both transient

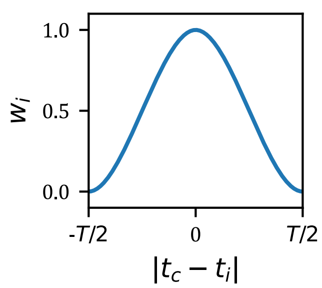

Figure 2. Plot of the rolling window weighing function given in Equation 6.

and periodic motions, although this paper focuses solely on transient motions. For a single dimension, the chosen system represents a globally stable second-order linear system perturbed by a non-linear forcing term:

$$\begin{aligned}\tau\dot{v} &= K(x_g - x) - Dv + (x_g - x_0)f(s)\\ \tau\dot{x} &= v,\end{aligned} \tag{9}$$

where x represents the position of the system; x_0 and x_g are the start and goal position; K and D are the positive stiffness and damping constants respectively; τ is the temporal scaling factor; v is the time-scaled velocity and $f(s)$ is the non-linear forcing term used to produce any arbitrary movement, dependent on phase variable s. This second-order dynamical system is known as the transformation system. Note that the notation differs from the original paper [5] and instead follows [37] to show the spring-damper character. D is set to be equal to $2\sqrt{K}$ to make the system critically damped. The non-linear forcing function is defined as

$$f(s) = \frac{\sum_i w_i \varphi_i(s) s}{\sum_i \varphi_i(s)}, \tag{10}$$

with weighing coefficients w_i and Gaussian basis functions $\varphi(s)$, defined as

$$\varphi(s) = \exp(-h_i(s - c_i)^2) \tag{11}$$

with centers c_i and widths h_i. The phase variable s parametrises time t by monotonically decreasing from 1 to 0. As s goes to 0 at the end of the movement, $f(s)$ goes to 0 as well. This leaves only the spring-damper nature of the system, guaranteeing convergence of x to x_g.

The phase variable s is governed by the so-called canonical system

$$\tau\dot{s} = -\alpha s, \tag{12}$$

with α determining the rate of the exponential decay. s is set to 1 at the start of the movement and can used as a measure of the progress of the movement. The phase variable decouples the system from absolute time, allowing temporal scaling of the movement by modification of the temporal scaling factor τ.

Learning a demonstrated trajectory requires finding w_i. Given a demonstrated trajectory $x_d(t)$ and its time derivatives $\dot{x}_d(t)$ and $\ddot{x}_d(t)$, it is possible to rewrite (9) and find the desired forcing function f_d as

$$f_d(t) = \frac{-K(x_g - x(t)) + D\tau\dot{x}(t) + \tau^2\ddot{x}(t)}{x_g - x_0}, \tag{13}$$

where $x_0 = x_d(0)$ and $x_g = x_d(T)$. After integrating the canonical system (12) to find $s(t)$, finding the weights w_i that minimize the squared difference between $f_d(s)$ and $f(s)$ is a weighted linear regression problem and can be solved efficiently using a least squares solver [37, 38].

The DMP can generate new trajectories to adapt to changes in the environment by adjusting the start x_0, goal x_g, and duration τ as desired. These can either be set manually or autonomously, which requires a system capable of perceiving the environment and adapting the parameters accordingly.

To extend the formulation to multiple dimensions, each dimension is given its own transformation system with independent forcing functions. These are then synchronized by a single, shared canonical system.

3.4.2 *Revised Formulation*

As described by Pastor et al. [37], the original transformation system of (9) has some drawbacks: First, if the start and goal position of a demonstration are equal, the non-linear function $f(s)$ is multiplied by 0 and cannot drive the transformation system. Second, if $x_g - x_0$ is close to zero, small changes in x_g may lead to large accelerations. Finally, when generalizing to a new goal $x_{g,\text{new}}$ that makes $(x_{g,\text{new}} - x_0)$ change its sign compared to $(x_{g,\text{original}} - x_0)$, the resulting reproduction is mirrored. Therefore, the following transformation system was proposed [39, 37]:

$$\begin{aligned}\tau\dot{v} &= K(x_g - x) - Dv - K(x_g - x_0)s + Kf(s)\\ \tau\dot{x} &= v,\end{aligned} \tag{14}$$

where the non-linear forcing function and canonical system are defined as before, see (10) and (12) respectively. This formulation has two main changes; the non-linear forcing function is not multiplied by $(x_g - x_0)$ anymore to prevent the problems described above and the term $K(x_g - x_0)s$ is added, which prevents acceleration jumps at the beginning of a movement. Rearranging (14) yields the desired forcing function f_d as

$$f_d(t) = -(x_g - x_d(t)) + \frac{D\tau\dot{x}_d(t) + \tau^2\ddot{x}_d(t)}{K} + (x_g - x_0)s. \tag{15}$$

This formulation is used in the proposed framework for translations of the end-effector.

3.4.3 *Orientations*

The DMP formulations detailed above describe motions of a single degree of freedom. All degrees of freedom are generated independently and synchronized through the canonical system. While this works for independent degrees of freedom such as robot joints or Cartesian positions, it does not work for the constrained representations of Cartesian orientations, such as unit quaternions or rotation matrices [40]. The set of all orientations, denoted by $SO(3)$, is not a vector space [7]. Minimal representations of $SO(3)$, i.e. using only three parameters, contain discontinuities or singularities such as gimbal lock. A singularity-free representation of $SO(3)$ are unit quaternions $\mathbf{q} = v + \mathbf{u} \in \mathbb{S}^3$, with $\mathbb{S}^3$ being a unit sphere in $\mathbb{R}^4$, scalar part $v \in \mathbb{R}$, and vector part $\mathbf{u} \in \mathbb{R}^3$. See Appendix A for quaternion definitions and operations. While the quaternion representation of orientations is non-minimal, it has only four parameters instead of nine when using rotation matrices. Due to the unit length constraint of quaternions, it is mathematically incorrect to use four independent transformation systems to encode quaternion-based orientations in a DMP [41].

Ude et al. [7] have proposed a quaternion-based transformation system, which has been the dominant approach for orientation DMPs since then [8]. However, as shown by Koutras et al. [8], changes in the rotation axis during a demonstration break the linearity of the transformation system. This leads to undesired oscillatory behaviour when the rotation axis is not fixed. To overcome this, instead of encoding quaternions directly they propose to encode the rotation vector between the current orientation $\mathbf{q}$ and the goal orientation $\mathbf{q}_g$. They define the quaternion error $\mathbf{e_q} \in \mathbb{R}^3$ as

$$\mathbf{e_q} = 2\log(\mathbf{q}_g * \bar{\mathbf{q}}), \tag{16}$$

which is equal to a rotation around a fixed axis $\mathbf{n} \in \mathbb{R}^3$ by rotation angle $(\theta_g - \theta)$ (see (33) in Appendix A):

$$\mathbf{e_q} = (\theta_g - \theta)\mathbf{n}. \tag{17}$$

Koutras et al. [8] propose the following transformation system to encode the quaternion error in a DMP:

$$\begin{aligned} \tau\dot{\mathbf{z}} &= -\mathbf{K}\mathbf{e_q} - \mathbf{D}\mathbf{z} + \mathbf{e_{q_0}} \circ \mathbf{F}(s) \\ \tau\dot{\mathbf{e}}_\mathbf{q} &= \mathbf{z}, \end{aligned} \tag{18}$$

where $\mathbf{z} \in \mathbb{R}^3$ is the scaled quaternion error velocity, $\mathbf{e_{q_0}} = 2\log(\mathbf{q}_g * \bar{\mathbf{q}}_0)$ is the initial quaternion error, and $\mathbf{F}(s) \in \mathbb{R}^3$ is the multi-dimensional non-linear forcing term trained to reproduce the demonstration. $\mathbf{K} \in \mathbb{R}^3$ and $\mathbf{D} \in \mathbb{R}^3$ are constant, diagonal, positive matrices. The notation has again been changed for consistency with the rest of the paper. Integrating (18) results in the quaternion error, which can be transformed to the desired quaternion by solving (16) for $\mathbf{q}$:

$$\mathbf{q} = \overline{\exp\left(\frac{1}{2}\mathbf{e_q}\right)} * \mathbf{q}_g. \tag{19}$$

The transformation system of (18) resembles the original DMP formulation found in (9). However, it also brings the same issues of scaling by the initial error as described in Section 3.4.2. Therefore, the following transformation system is proposed:

$$\begin{aligned} \tau\dot{\mathbf{z}} &= -\mathbf{K}\mathbf{e_q} - \mathbf{D}\mathbf{z} + \mathbf{K}\mathbf{e_{q_0}}s + \mathbf{K}\mathbf{F}(s) \\ \tau\dot{\mathbf{e}}_\mathbf{q} &= \mathbf{z}, \end{aligned} \tag{20}$$

which removes the scaling of $F(s)$ by $\mathbf{e_{q_0}}$ and introduces the term of $\mathbf{K}\mathbf{e_{q_0}}s$ to prevent acceleration jumps at the beginning, again as discussed in Section 3.4.2. To learn the desired forcing function given a demonstrated orientation trajectory $\mathbf{q}_d(t)$, it is first required to calculate $\mathbf{e}_{\mathbf{q}_d}(t), \dot{\mathbf{e}}_{\mathbf{q}_d}(t), \ddot{\mathbf{e}}_{\mathbf{q}_d}(t)$. The desired quaternion error $\mathbf{e}_{\mathbf{q}_d}(t)$ is obtained by solving (16) for all $\mathbf{q}$ in $\mathbf{q}_d(t)$. $\dot{\mathbf{e}}_{\mathbf{q}_d}(t)$ and $\ddot{\mathbf{e}}_{\mathbf{q}_d}(t)$ can be obtained by simply differentiating $\mathbf{e}_{\mathbf{q}_d}(t)$. Then, the desired forcing function is computed with:

$$\mathbf{F}_d(s) = -\mathbf{e}_{\mathbf{q}_d}(t) + \frac{D\tau\dot{\mathbf{e}}_{\mathbf{q}_d}(t) + \tau^2\ddot{\mathbf{e}}_{\mathbf{q}_d}(t)}{K} + \mathbf{e_{q_0}}s. \tag{21}$$

Identical to translation DMP, each dimension of $\mathbf{F}(s)$ is approximated with (10). However, three additional considerations need to be taken into account before using quaternion DMPs, which are not discussed by Koutras et al. [8]:

i. *Ensure smooth function:* To learn a smooth forcing function $\mathbf{F}(s)$, it is crucial that $\mathbf{e}_{\mathbf{q}_d}(t)$ is smooth as well. While $\mathbf{q}$ and $-\mathbf{q}$ describe the same orientation, $2\log(\mathbf{q})$ is not equal to $2\log(-\mathbf{q})$. One will describe a rotation of θ rad around axis $\mathbf{n}$, while the other describes a rotation of $2\pi - \theta$ rad around axis $-\mathbf{n}$. This will lead to different $\mathbf{e}_{\mathbf{q}_d}$. To ensure smooth $\mathbf{e}_{\mathbf{q}_d}(t)$, the geodesic distance on $\mathbb{S}^3$ between adjacent $\mathbf{q}$ should minimized by flipping the sign of $\mathbf{q}_{i+1}$ if $d_{rot}(\mathbf{q}_i, \mathbf{q}_{i+1}) > d_{rot}(\mathbf{q}_i, -\mathbf{q}_{i+1})$, with d_{rot} defined in (3).

ii. *Avoid singularities:* The quaternion error $\mathbf{e_q}$ of (16) contains a singularity when $\mathbf{q} = -\mathbf{q}_g$. Additionally, when $\mathbf{q}$ is close to $-\mathbf{q}_g$, small changes in $\mathbf{q}$ can lead to large changes in the rotation axis $\mathbf{n}$. This singularity has been pointed out earlier in [7]. To deal with it, they propose to keep the previous $\mathbf{e_q}$ when reaching the singularity. To prevent sharp changes in $\mathbf{e_q}$ resulting from rapid changing $\mathbf{n}$, this paper proposes to keep the previous $\mathbf{e_q}$ while $||\mathbf{e_q}||$ would be within a certain angle θ_{thres} of 2π rad. A suitable threshold angle has empirically been found to be 0.02 rad, see Section 4.2.1.

iii. *Rectify goal quaternion:* As mentioned, the sign of $\mathbf{q}$ determines the rotation direction. If the goal quaternion $\mathbf{q}_g$ is on the other side of the 3-sphere w.r.t. the initial quaternion as during learning (i.e. if $\mathbf{q}_g$ is on the opposite side of $\mathbf{q}_0$ during reproduction when they were on the same side during learning, or vice versa), the learned forcing function will not match the dynamical system. The goal quaternion must therefore be flipped if the alignment of $\mathbf{q}_g$ and $\mathbf{q}_0$ during reproduction does not match the alignment during training.

The effects of forgoing one of these considerations will be shown in Section 4.2.1.

3.4.4 *Wrenches and Standard Deviations*

The end-effector wrenches, consisting of end-effector forces and torques, are not encoded using the transformation system of (14), but directly approximated in the same manner as the forcing function using (10). This is because we generally only want to track the force profile during reproduction and do not need the stable convergence between an initial and goal wrench provided by a DMP. Each component of the wrench is independently approximated, which results in a set of weighing coefficients $\mathbf{w}$ per component. The desired wrench can be extracted at any given time during task reproduction by filling the current phase in (10). The standard deviation profiles obtained in Section 3.3 are encoded in this manner as well.

3.5 Hybrid Force-Impedance Controller

To allow the simultaneous tracking of motion and force profiles provided by the DMP, a hybrid force-impedance controller is used. It is an adaption of classic hybrid force-position control [13], using variable stiffness impedance control [11] in place of position control. Impedance control is applied to unconstrained directions and force control is applied to constrained directions.

The equations of motion of a manipulator can be expressed as

$$\mathbf{M}(\mathbf{q}^c)\ddot{\mathbf{q}}^c + \mathbf{C}(\mathbf{q}^c, \dot{\mathbf{q}}^c)\dot{\mathbf{q}}^c + \mathbf{g}(\mathbf{q}^c) = \boldsymbol{\tau} + \boldsymbol{\tau}^{ext}, \tag{22}$$

with the generalized joint coordinates $\mathbf{q}^c$ (not to be confused with quaternions $\mathbf{q}$), inertia matrix $\mathbf{M}(\mathbf{q}^c)$, Coriolis and centrifugal matrix $\mathbf{C}(\mathbf{q}^c, \dot{\mathbf{q}}^c)$, gravitational forces $\mathbf{g}(\mathbf{q}^c)$, joint torques $\boldsymbol{\tau}$ and external joint torques $\boldsymbol{\tau}^{ext}$ [42]. An applied external wrench $\mathbf{F}^{ext}$ is mapped to external joint torques with

$$\boldsymbol{\tau}^{ext} = -\mathbf{J}(\mathbf{q}^c)^T\mathbf{F}^{ext}, \tag{23}$$

using Jacobian matrix $\mathbf{J}(\mathbf{q}^c)$[42]. Both impedance control and force control exert an end-effector wrench to realize a desired behaviour. Task space impedance control implements a virtual spring-damper at the end-effector to interact with the environment in a safe and energy-efficient manner [43]. The desired impedance control end-effector wrench is obtained with

$$\begin{aligned} \mathbf{F}^{\mathcal{P}}_{imp} &= \mathbf{K}^{\mathcal{P}}_{imp}(\mathbf{x}_d - \mathbf{x}) - \mathbf{D}^{\mathcal{P}}_{imp}\dot{\mathbf{x}} && \text{(positions)} \\ \mathbf{F}^{\mathcal{O}}_{imp} &= \mathbf{K}^{\mathcal{O}}_{imp}(\log(\mathbf{q}_d * \bar{\mathbf{q}})) - \mathbf{D}^{\mathcal{O}}_{imp}\boldsymbol{\omega} && \text{(orientations),} \end{aligned} \tag{24}$$

where $\mathbf{F}^{\mathcal{P}}_{imp}$ and $\mathbf{F}^{\mathcal{O}}_{imp}$ denote the translational (forces) and rotational (torques) components of the wrench, respectively; $\mathbf{K}^{\mathcal{P}}_{imp}$ and $\mathbf{K}^{\mathcal{O}}_{imp}$ denote the translational and rotational stiffness; $\mathbf{D}^{\mathcal{P}}_{imp}$ and $\mathbf{D}^{\mathcal{O}}_{imp}$ denote the translational and rotational damping, set for critical damping; $\mathbf{x}$ and $\mathbf{x}_d$ denote the current and desired end-effector position; $\mathbf{q}$ and $\mathbf{q}_d$ denote the current and desired end-effector orientation as quaternion; and ω denotes the current angular velocity. The translational and angular velocity are calculated with $\mathbf{J}(\mathbf{q}^c)\dot{\mathbf{q}}^c$[44].

When the robot is constrained in certain directions, the interaction force can be controlled using force control. The wrench containing the desired interaction force is denoted with $\mathbf{F}_f$. A selection matrix $\mathbf{S}$ is used to specify which axes of the robot are constrained and should therefore use force control instead of impedance control. It stores which of the Cartesian dimensions $[x, y, z, \theta_x, \theta_y, \theta_z]$ is constrained (= 1) or free (= 0) in a diagonal matrix. As an example, a selection matrix $\mathbf{S} = \mathrm{diag}([0, 0, 1, 0, 1, 0])$ indicates that the translations along the z-axis and rotations around the y-axis are constrained. The selection matrix is then $\mathbf{S} = \mathrm{diag}([0, 0, 1, 0, 1, 0])$. The desired force wrench $\mathbf{F}_f$ and the desired impedance wrench $\mathbf{F}_{imp}$ are then multiplied by $\mathbf{S}$ and $(\mathbf{I} - \mathbf{S})$ respectively to separate the desired wrenches acting along the constrained and free axes. The control law is finally given by:

$$\tau = \underbrace{-\mathbf{J}(\mathbf{q}^c)^T(\mathbf{I} - \mathbf{S})\mathbf{F}_{imp}}_{\tau_{imp}} + \underbrace{-\mathbf{J}(\mathbf{q}^c)^T\mathbf{S}\mathbf{F}_f}_{\tau_f} + \tau_{comp}, \tag{25}$$

where τ_{imp} and τ_f denote the impedance- and force-control joint torques, respectively. Gravity and Coriolis compensations can be included in τ_{comp}. As described in [42], feedback control can be added to the force control term with a PI controller using

$$\tau_f = -\mathbf{J}(\mathbf{q}^c)^T\mathbf{S}\left(\mathbf{F}_f + \mathbf{K}_f^P(\mathbf{F}_f - \mathbf{F}^{ext}) + \mathbf{K}_f^I\int(\mathbf{F}_f - \mathbf{F}^{ext})dt\right), \tag{26}$$

where $\mathbf{K}_f^P$ and $\mathbf{K}_f^I$ are proportional and integral gain matrices, respectively. However, Iskander et al. [42] found that the performance benefit of force feedback was minor. Although it reduced steady-state force error and decreased dynamic force oscillations, it came at a cost. During fast motions, where force errors are to be expected, contact force deviations could generate high joint torques reaching saturation limits, especially during impacts with a surface. Instead, omitting the force feedback terms increases robustness and simplifies the controller implementation, which has been done in this paper. Should the robot possess a redundant degree of freedom, an additional term for null-space impedance control may be added to (25) to control the null-space motion [45].

3.6 Impedance Behavior Modulation

During reproduction, the learned standard deviations control the impedance stiffness gains. The impedance stiffness is set individually for each direction. The impedance stiffness gains are scaled linearly between a minimum and maximum according to

$$K_{imp}(\sigma) = \begin{cases} K_{min}, & \text{if } \sigma > \sigma_{max} \\ K_{max}, & \text{if } \sigma < \sigma_{min} \\ \frac{(\sigma-\sigma_{min})(K_{min}-K_{max})}{\sigma_{max}-\sigma_{min}} + K_{max}, & \text{otherwise,} \end{cases} \tag{27}$$

where K_{min}, K_{max} denote the minimum and maximum impedance stiffness in that direction, σ_{max}, σ_{min} denote the corresponding standard deviation thresholds, and σ denotes the identified standard deviation in that direction. K_{min}, K_{max}, σ_{min}, σ_{max} are strictly positive constants that need to be set in advance. This is a linear decreasing function of the standard deviation that saturates at K_{max} and K_{min}. This approach allows for the selection of K_{min}, K_{max} based on the requirements of the specific use case. Corresponding σ_{max}, σ_{min} should then be experimentally determined based on the tracking accuracy of the robot at these stiffness values. Due to the difference in unit, K_{min}, K_{max} have different values for translational and rotational stiffness. The damping is set for critical damping.

4 Framework Evaluation

4.1 Setup

The proposed framework was implemented on a 7 DOF Franka Research 3 (Franka Robotics GmbH) robot. It was equipped with the 6-axis SensONE Serial (Bota Systems AG) Force Torque sensor. The software was implemented using ROS middleware [46]. The robot controller was programmed in C++ 14, while the other parts of the framework were written in Python 3.8. The controller runs at a rate of 1 kHz. The operating system of the workstation was Ubuntu 20.04, with real-time kernel PREEMPT_RT 5.15.125-rt66.

The robot frames used in this framework are shown in Figure 3. The recorded end effector pose contains the position and orientation of Ψ_{EE} with respect to Ψ_{base}. The recorded wrench is measured in Ψ_F, which has the same origin as Ψ_{EE} but is always aligned with Ψ_{base}. This ensures that the poses and wrenches are measured and reproduced along aligned axes, which allows using the selection matrix $\mathbf{S}$ to specify which directions are constrained. Currently, only directions aligned with the static Ψ_{base} can be specified as (un)constrained.

To show the effects of the orientation DMP corrections proposed at the end of Section 3.4.3, a demonstration consisting of a clockwise rotation of π rad around the z-axis and a consequent counter-clockwise rotation of 2π around the same axis is used. Section 4.2.1 shows how the proposed corrections prevent artefacts. While this is an artificial example, real tasks involving orientations (e.g. screwing, part alignment, environment scanning) can run into the issues

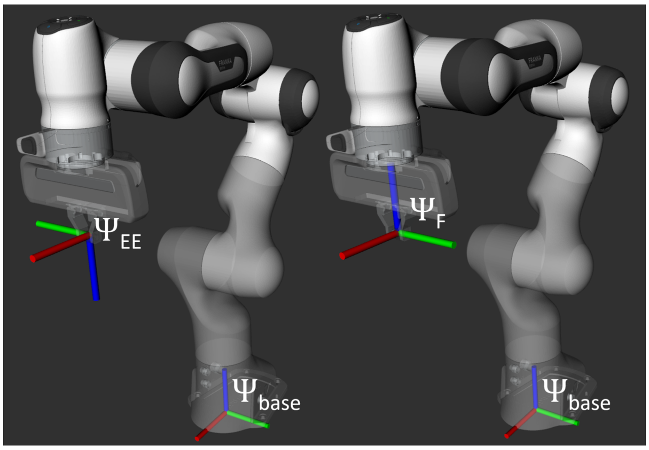

Figure 3. The robot frames used by the framework. The end effector pose is obtained by finding the pose of Ψ_{EE} in base frame Ψ_{base}. The end effector wrench is measured in Ψ_F. This frame is coincident with Ψ_{EE} but always aligned with Ψ_{base}. During the drawing task, Ψ_{EE} and Ψ_{EE} are located at the tip of the attached pen. The red, green, and blue axes denote the x-, y-, and z-axis, respectively.

showcased using this example.

To verify and evaluate the framework's capabilities, it was tested on two real-world tasks which were demonstrated by a human operator and reproduced by the robot. These tasks were chosen to illustrate the versatility of the framework in performing diverse tasks. For these experiments, due to the ease of task demonstration and the direct force feedback to the demonstrator, kinesthetic teaching was used as the demonstration interface.

The first task, showcasing a motion-based task, involved the watering of a plant. A watering can, rigidly attached to the end-effector of the robot, was placed some distance from the plant. It was then raised and moved towards the plant, rotated around the y-axis until water flowed out, and rotated back. This task was demonstrated five times from different starting positions. Figure 4 depicts timelapses of the watering plant task, showing the demonstration, reproduction, and generalization to a new pose.

The second task, showcasing a contact-based task, entailed pushing a pen on a whiteboard to draw a pattern within two borders of a maze. The robot was aligned such that its x-axis was perpendicular to the plane of the whiteboard. The borders were parallel to either the y- or z-axis of the robot. The width of these borders varied over the progress of the motion. The pen was attached to the robot using a custom, 3D-printed holder connected to the robot's mounting flange. The task was demonstrated five times, all beginning from the start of the maze. Since this was a force-relevant task, the end-effector wrenches were measured using the F/T sensor.

The experiments, in addition to showcasing the framework's workflow on example applications, aim to verify and evaluate the three goals of the framework:

i. The framework is able to vary impedance stiffness over time based on the variance. This will be evaluated in Section 4.2.2 by comparing the motion reference tracking performance of the proposed variable impedance control law to the constant high and constant low impedance stiffness, and to the identified standard deviations of the watering task.

ii. The framework is able to generalize task reproductions from and to any reachable target pose. This will be verified in Section 4.2.3 by showing how the DMP adapts a learned movement to different poses in a stable manner.

iii. The framework is able to simultaneously reproduce forces in constrained directions and motions in unconstrained directions. This will be evaluated Section 4.2.4 by comparing the motion and force tracking performance of the hybrid controller using the variable impedance control law to the identified standard deviations of the drawing task.

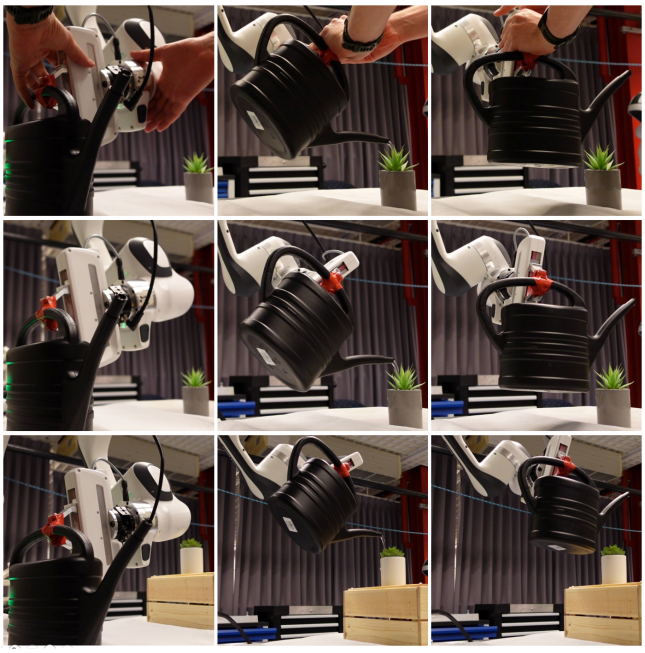

Figure 4. Timelapses of the watering plant experiment. The top row shows the task demonstration. The middle row shows the reproduction of the task with the same task parameters, using the variable impedance control law. The bottom row shows the reproduction while generalizing to changes in the environment..

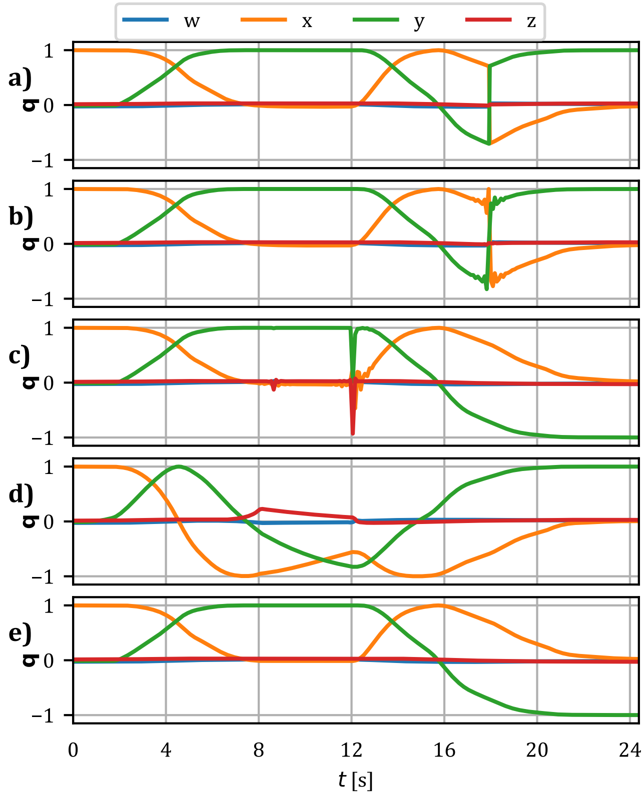

Figure 5. Plots showing the effect of forgoing the proposed quaternion DMP corrections. **a)** shows the quaternions of the demonstration consisting of a rotation of a clockwise rotation of π rad around the z-axis and a subsequent counter-clockwise rotation of 2π around the same axis. **b)** shows the reproduction of the demonstration if the distance on $\mathbb{S}^3$ between consecutive quaternions in the demonstration is not minimized. **c)** shows the reproduction of the demonstration if no correction is applied to avoid the region where $||\mathbf{e_q}||$ approaches 2π rad. **d)** shows the reproduction of the demonstration if no correction is applied to make sure the goal quaternion of the reproduction is on the same hemisphere of $\mathbb{S}^3$ w.r.t. the initial quaternion as during the learning of the forcing function. **e)** shows the reproduction of the demonstration using all proposed corrections described in Section 3.4.3.

4.2 Results

4.2.1 *Quaternion DMP Corrections*

The effects of the orientation DMP corrections are shown in Figure 5. The quaternion orientation of the end-effector during the rotation demonstration can be seen in Figure 5.**a**. The orientation trajectory is smooth, although the quater-

Table 1. Root mean square standard deviation (SD) and root mean square error of reproductions with variable impedance stiffness (K_{var}), constant low impedance stiffness (K_{min}), and constant high impedance stiffness (K_{max}). The values correspond to the watering task and plots shown in Figure 8.

Dimension	SD	K_{var}	K_{min}	K_{max}
x [m]	0.033	0.012	0.040	0.012
z [m]	0.014	0.009	0.066	0.008
θ_y [rad]	0.051	0.126	0.257	0.118

nion value suddenly jumps around $t = 18$ s. While the quaternion negated its sign, it still describes the same orientation due to the unit quaternion's double cover of $SO(3)$.

If this demonstration is directly learned with the orientation DMP without correcting the jump in quaternion, it will result in a non-continuous step in the desired non-linear forcing function of (21). Since the amount of Gaussian kernels approximating the desired forcing function is finite, this step cannot be reproduced exactly. This results in visible oscillations around the quaternion's shift in sign, which can be seen in Figure 5.**b**.

If during a demonstration the norm of the quaternion error $\mathbf{e_q}$ approaches 2π, the rotation axis $\mathbf{n}$ can change rapidly. This happens when $\mathbf{q}$ gets close to $-\mathbf{q}_g$, i.e. approaches the opposite side of the 3-sphere. Small changes in $\mathbf{q}$ can result in significant changes in the rotation axis $\mathbf{n}$, which results in significant changes of desired $\mathbf{e_q}$. This is visible in Figure 5.**c** around $t = 12$ s. To mitigate this, $\mathbf{e_q}$ is set to the previous value for $\mathbf{q}$ that would result in a rotation angle $(\theta_g - \theta) > 2\pi - \theta_{\mathrm{thres}}$. θ_{thres} was empirically determined. Too large values would lead to overly conservative behavior, reducing the system's ability to reproduce trajectories. Conversely, setting θ_{thres} too small would fail to prevent oscillations. A threshold θ_{thres} of 0.02 rad was found to be a suitable value.

If the goal quaternion during reproduction is not on the same side of the 3-sphere w.r.t. the initial quaternion as during the learning of the forcing function, the reproduction will appear as shown in Figure 5.**d**. The spring-damper system of the reproduction will rotate in the opposite direction compared to during the learning of the demonstration. This results in the learned forcing function not complementing the new spring-damper system. Consequently, the reproduced motion will not resemble the original demonstration.

Finally, taking into account all proposed corrections for the described problems results in a smooth DMP reproduction, which is shown in Figure 5.**e**. It can be seen that the quaternions follow a smooth trajectory and that the reproduction method does not introduce any artefacts. Reproducing this trajectory on a robot results in approximately the same movement as the demonstration; only the rotation angle when $||\mathbf{e_q}||$ approaches 2π rad is slightly reduced. This reduction in angle is less than θ_{thres}, which is 0.02 rad.

4.2.2 *Variable Impedance Control*

To show the steps of the framework and the effect of the variable impedance control, the watering plant task was used. The demonstrations, containing poses of the end-effector, were first temporally aligned as described in Section 3.2. Afterwards, a mean representative demonstration and its standard deviation per dimension were inferred from the synchronized demonstrations following Section 3.3. The results of these steps are illustrated in Figure 6. This figure shows the x- and z-position and the rotation around the y-axis. The rotation around the y-axis was obtained by converting the quaternions into Euler angles using extrinsic rotations in the XYZ order. Euler angles are used to show the 'independent' rotation around the y-axis during the tilting of the watering can.

The standard deviations are used to modulate the impedance stiffness per axis as described in Section 3.6 following (27). The impedance parameters were set empirically. First, the steady-state errors were measured for different impedance stiffness values after the application of a step reference. The minimum impedance stiffness was then chosen based on the desired maximum steady-state error. The maximum impedance stiffness was chosen based on a minimum desired compliance level, i.e. the minimum compliance that the robot should exhibit during operation. The corresponding standard deviation values were chosen to be equal to the measured steady-state error at these stiffness values. For this setup, the following parameters were determined:

$$
\begin{aligned}
K^{\mathcal{P}}_{min} &= 40\ \frac{\mathrm{N}}{\mathrm{m}}, & \sigma^{\mathcal{P}}_{max} &= 0.05\ \mathrm{m} \\
K^{\mathcal{P}}_{max} &= 600\ \frac{\mathrm{N}}{\mathrm{m}}, & \sigma^{\mathcal{P}}_{min} &= 0.0025\ \mathrm{m} \\
K^{\mathcal{O}}_{min} &= 3\ \frac{\mathrm{Nm}}{\mathrm{rad}}, & \sigma^{\mathcal{O}}_{max} &= \frac{\pi}{6}\ \mathrm{rad} \\
K^{\mathcal{O}}_{max} &= 15\ \frac{\mathrm{Nm}}{\mathrm{rad}}, & \sigma^{\mathcal{O}}_{min} &= \frac{\pi}{90}\ \mathrm{rad}
\end{aligned}
$$

These parameters, in conjunction with the identified standard deviation (shown as 95% CI) as seen in the right column of Figure 6, result in the variable stiffness profiles shown in Figure 7. For evaluation, the mean demonstration of the watering task shown in the right column of Figure 6 was reproduced three times, using the variable impedance stiffness and high and low constant impedance stiffness. The error between the reference trajectory and the robot trajectory was recorded. Figure 8 shows the error over time during reproduction of the mean watering task for the proposed variable impedance stiffness, constant impedance stiffness equal to K_{min}, and constant impedance stiffness equal to K_{max}. The standard deviation is plotted as well. It can be seen that the reproduction with K_{min} has the greatest pose tracking error, as expected. K_{var} and K_{max} show comparable error profiles, with K_{max} appearing to have a slightly lower error. It can be seen that for the position tracking in the x- and z-direction, the error of K_{var} and K_{max} is generally below the standard deviation. However, the rotation tracking around the y-axis shows worse performance, with all tested impedance stiffnesses having errors significantly exceeding the standard deviation during the task phase where the watering can was tilted forward. The root mean square of the standard deviation and pose errors are shown in Table 1.

4.2.3 *Generalization*

To show the generalization capability of the DMP formulation, the watering plant task was used. The initial and goal pose were changed from the reference, to showcase the framework adapting to changes in its environment. The changes represent a changed initial position of the watering can and a changed position of the plant. The initial pose was translated 0.05 m in the x-axis and rotated $\frac{\pi}{66}$ rad around the y-axis. The goal pose was translated 0.15 m in the z-axis and rotated $-\frac{\pi}{16}\pi$ rad around the y-axis. Figure 9 shows that the DMP adapts to these changes, while still following the shape of the demonstration. The bottom row of Figure 4 shows the watering can being raised to reach the elevated plant during the generalization.

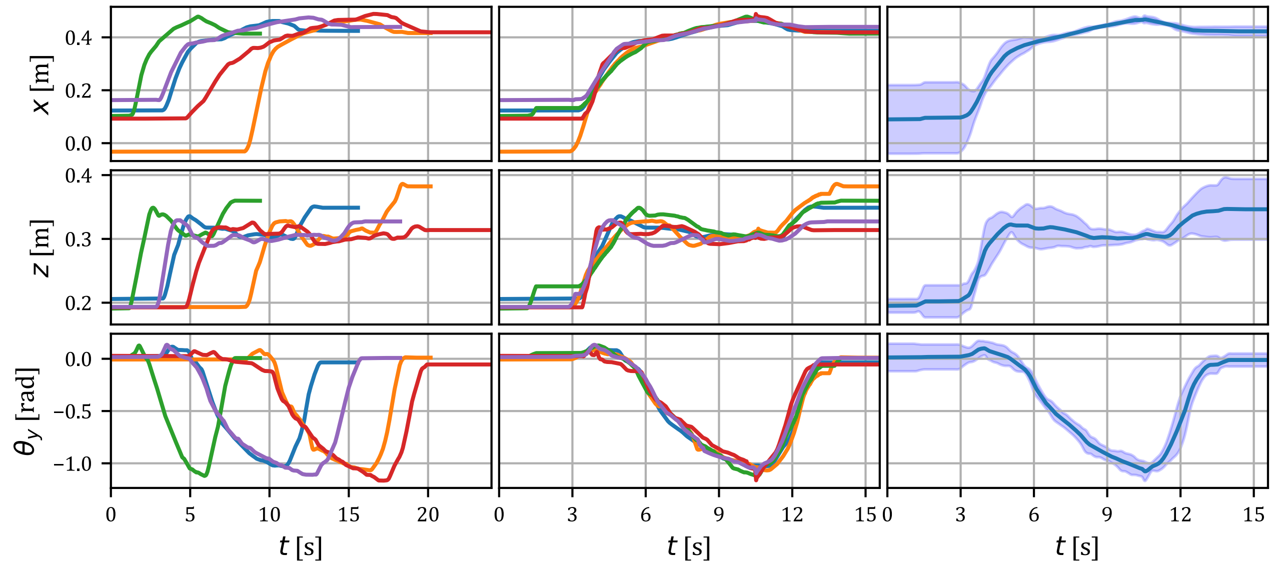

Figure 6. Time series of the watering plant experiment. Only the x- and z-position and rotation around the y-axis are shown for brevity. The left column shows the original demonstrations. The colored lines represent the different demonstrations. The centre column shows the demonstrations after temporal alignment. The right column shows the 'mean' representation of the task and the identified standard deviation. The shaded region shows the 95% confidence interval. Since the standard deviation of rotation around the y-axis is obtained using (8) and not based on Euler angles, perceived visual variance in the aligned Euler angles in the bottom-center plot does not directly concur with the identified standard deviation shown (through the 95% CI) in the bottom-right plot.

4.2.4 *Contact Task Reproduction*

Finally, to show the simultaneous motion and force reproduction allowed by the framework, the drawing task was used. The five demonstrations, consisting of end-effector poses and measured external forces, were temporally aligned. The mean, representative demonstration was derived and the standard deviation in the individual pose and wrench dimensions were determined. Although the standard deviation for force and torque dimensions is determined, it does not serve a use in the framework. The stills in Figure 10 show the demonstration and reproduction of the task. Figure 11 shows the mean and identified standard deviation as 95% CI in the yz-plane.

The reproduction used the hybrid control law of (25), with the proposed variable impedance stiffness. The force control portion of the hybrid controller did not use the force

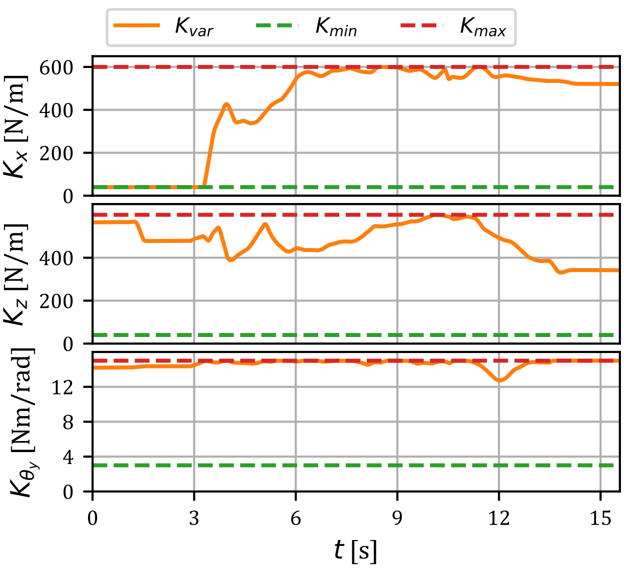

Figure 7. The impedance stiffness over time for the watering task. The orange line shows the variable impedance stiffness It is obtained by using the impedance parameters described in Section 4.2.2 and filling the standard deviation as seen in Figure 8 into (27). The green and red line shows the minimum and maximum impedance stiffness.

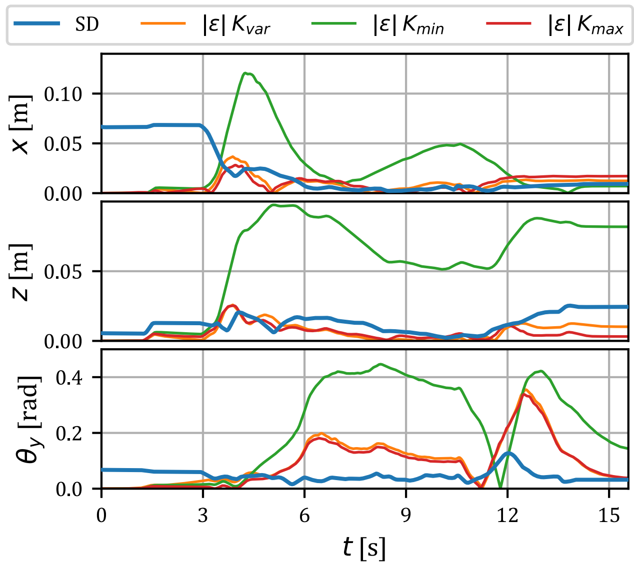

Figure 8. The standard deviation and absolute error over time during reproduction of the watering task for various impedance stiffnesses. The blue line shows the identified standard deviation as seen in Figure 6. The orange line shows the absolute error during reproduction using the variable stiffness of Figure 7. The green line uses a low constant stiffness and the red line uses a high constant stiffness. The root mean squares of these errors are given in Table 1.

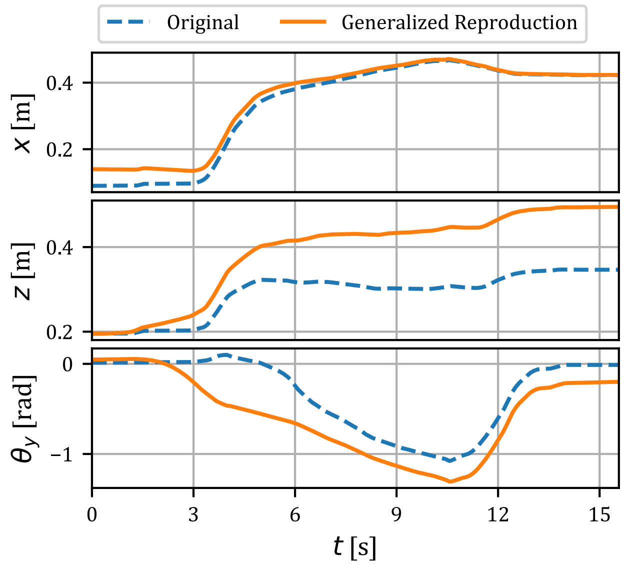

Figure 9. Reference of watering task showing the generalization capabilities of the DMP. The dashed blue line shows the original reference, obtained from the regression shown in Figure 6. The solid orange line shows how the DMP follows the demonstration shape with a different start pose in x and θ_y direction and a different goal pose in z and θ_y direction.

feedback control terms of (26). The used selection matrix was $\mathbf{S} = \mathrm{diag}([1, 0, 0, 0, 0, 0])$, denoting constraints in the translational x-direction and unconstrained motion in the other directions. The tracking error during reproduction in y-, z-position and x-force is shown together with the standard deviation in Figure 12. It can be seen that the position tracking errors are generally lower than the standard deviations. However, the force error is significantly greater than the identified standard deviation. The respective root mean square standard deviations and root mean square error are shown in Table 2.

5 Discussion

The goal of this work was to design and implement an LfD framework that is geometry-aware, capable of reproducing motion and forces based on which dimensions are constrained, uses variance between demonstrations to modulate impedance behavior, and was able to generalize any learned task from and to any reachable task pose. The results from the experiments confirm that the proposed

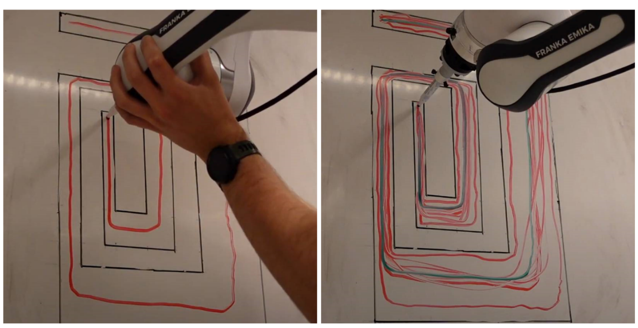

Figure 10. Stills of the whiteboard drawing experiment. The stills show the task demonstration (left) and reproduction (right). The walls of the maze are varying in width, which has been captured by the various demonstrations.

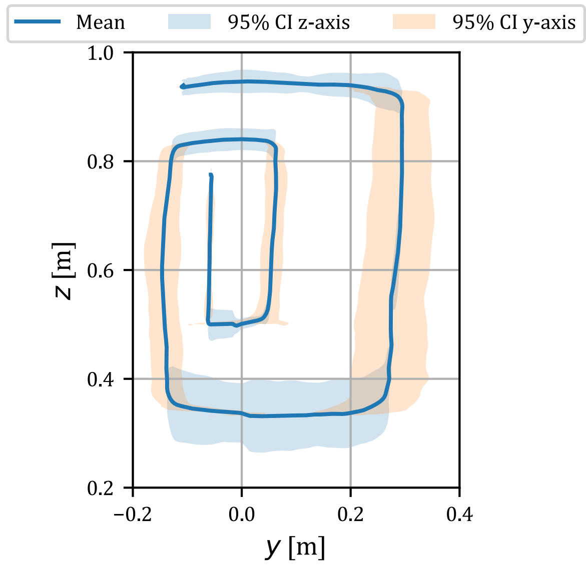

Figure 11. The mean and and standard deviation as 96% Confidence Interval of the drawing demonstrations in the yz-plane.

Table 2. Root mean square standard deviation (SD) and root mean square error of reproduction with variable impedance stiffness (K_{var}). The values correspond to the drawing task and plots shown in Figure 12.

Dimension	SD	K_{var}
x [m]	0.019	0.009
z [m]	0.016	0.014
F_y [N]	1.418	3.646

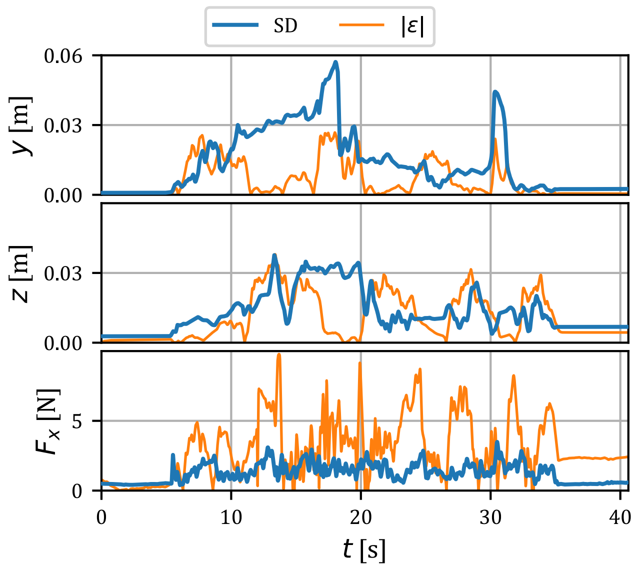

Figure 12. The standard deviation and absolute error over time during reproduction of the drawing task. The blue line shows the standard deviation, while the orange line shows the absolute error. The root mean squares of these errors are given in Table 2.

framework possesses these desired features.

The watering task showed that the proposed variable impedance controller was able to achieve comparable reference tracking results to a constant high stiffness impedance controller. The position tracking error shown in Figure 8 was generally below the standard deviation of the demonstrations, indicating that the reproduction of the demonstration was effectively less than one standard deviation away from the mean demonstration. By reducing the impedance stiffness based on larger standard deviations, the controller is able to balance compliance and reference tracking performance. As remarked in the introduction, increased robot compliance increases safety around humans and objects by allowing the robot to absorb and dissipate more energy.

However, the watering task did show relatively poor tracking performance for the rotations around the y-axis. One probable explanation for this is an inaccurate configuration of the watering can mass and inertia, in combination with the dynamic behavior of the water inside of the watering can. As the can is tilted, the water inside shifts. This results in a moment trying to undo the tilting of the can. The compliant nature of the impedance controller constitutes that the control is not able to nullify this moment completely, resulting in a worse orientation tracking performance.

The drawing task showed that the hybrid force impedance controller with variable impedance stiffness was able to track reference positions as the robot simultaneously exerts a force in a constrained direction. The position tracking error, as shown in Figure 12 and Table 2, was again generally below the standard deviation of the demonstrations. However, the force tracking performed considerably worse, resulting in a root mean square error of more than twice the root mean square standard deviation. One major contributor is the open-loop nature of the force controller. The performance of this feedforward controller is not adequate during motion parallel to the direction of force. A potential solution would be to incorporate the force torque sensor readings in the control loop of the robot and add PI force feedback control as suggested in (26). This would require a stable, tuned force feedback control system that can deal with sudden changes in measured forces. This might necessitate different tuned controller gains depending on the specific task speed, as remarked in [42]. Should feedback control be implemented, the identified standard deviation of the forces may be used to dynamically change the feedback gains based on the required accuracy of force tracking.

Additionally, the hybrid motion and force control require the constraints to align with the frame in which the poses and wrenches are stored, which in this case is the base frame Ψ_{base}. For example, if the whiteboard was angled during the reproduction of the drawing task, the trajectories should be expressed in a frame aligned with the new geometry of the constraints. This requires the identification of such constraint frames. Earlier work, such as by Subramani et al. [47], has focussed on inferring constraints from data but requires the prior definition of constraint models. Future work will focus on model-free identification of constraint frames. These frames allow the motion and forces to be transformed to align with the geometric constraints, enabling hybrid motion and force control on any geometry.

Encoding the demonstrations in the DMP enables the framework to generalize the task reproductions to changed situations, as shown in Figure 9. The amount of generalization is limited only by the workspace of the robot and potential collisions. The proposed quaternion DMP is able to generate stable trajectories, free from artefacts resulting from the orientation encoding. This does require the proposed corrections to be applied, as it can otherwise lead to the issues described and shown in Section 4.2.1. To prevent the singularity at $|\mathbf{e_q}| = 2\pi$, this paper proposes to avoid that region by keeping the quaternion error constant if it gets too close to the singularity. While it is effective, it is not ideal since the quaternion might leave the avoidance region in a different place than where it entered, resulting in a jump of the quaternion error. To prevent this, it would be better to look ahead to where the quaternion trajectory would leave the singularity avoidance region, and then smoothly interpolate between the two values to prevent this sudden jump from happening.

Compared to other work, the main advantage of the proposed framework is the combination of properties that are not found simultaneously in other approaches. As an example, the variable impedance works discussed in [22], [12], and [25] do not specifically encode end-effector orientations. Because of this, these works will either fail to reproduce orientation trajectories or introduce artefacts. While [23] and [24] do consider the non-Euclidean nature of rotations, they parametrize rotations as rotation vectors. This introduces discontinuities in the rotation encoding, which do not occur when using unit quaternions, such as in the proposed framework, or rotation matrices. Various other works focused on contact-based task learning such as [26], [27], and [30] fail to account for orientations as well. The framework proposed by [29] combines variable impedance control, force-based tasks and quaternion orientations. However, the variable impedance is optimized to simultaneously reproduce force and positions using an attractor model, instead of being optimized to balance compliance and tracking performance for safe operation around humans.

The framework's modular nature allows each component to be swapped or adjusted as desired. For instance, it would be possible to use a GMM or a GP as the model to infer a task policy from demonstration, provided that they are geometry-aware, e.g. the GMM proposed by Kim et al. [34]. Additionally, the vast existing research on the DMP framework results in many possible extensions that can be implemented without too much complication. These include object avoidance [48], which adds coupled terms to the DMP transformation system to steer around obstacles; phase stopping [8], which slows the evolution of the phase variable during disturbances to resume task reproduction when the disturbance disappears; smooth motion primitive sequencing [37], which allows sequential execution of motion primitives without halting of the system; smooth online goal switching [8], which allows smooth transitions to new goals to prevent sudden accelerations when the goal is updated online; rhytmic movements [38], which allows encoding of cyclic tasks such as sawing or stirring; and incorporation of viapoints[49], which allows adjustments of the DMP trajectory to pass specific intermediate points.

Another promising direction for future research is the integration of computer vision models to dynamically adjust the goal pose of the DMP. Such a model would segment objects from images and use depth vision to estimate their pose. This information could then be used to automatically adjust the trajectory of the DMP. For instance, in the plant watering trial, the target pose was manually altered to align with the plant's new position. Ideally, the robotic system does this autonomously by using the sensory information. Incorporating real-time vision-based information of the environment would improve task accuracy and robustness, especially in unstructured settings.

6 Conclusion

In this paper, an LfD framework was proposed that is capable of reproducing motion- and contact-based tasks from a single or multiple demonstration(s). The proposed framework uses a geometry-aware inference and standard deviation method to estimate time-dependent Gaussian distributions for each Cartesian dimension. The means are encoded in an orientation DMP model, which reproduces and generalizes end-effector translations and orientations without artefacts and singularities. A hybrid force impedance controller allows for simultaneous reproduction of motion and forces. The variable impedance stiffness, inferred from the variance between demonstrations, provides an automatic balance between reference tracking performance and robot compliance. The framework was validated on real-world tasks; watering a plant and drawing on a board. Results showed that the framework acted as desired and that position tracking was satisfactory. However, the rotation tracking during the watering task and the force tracking during the drawing task were relatively poor. This may be improved by identifying and configuring the dynamical properties of the end effector more accurately and adding force feedback control.

The work's contributions are threefold. Firstly, the framework exhibits a combination of properties not seen before in LfD work. Secondly, the proposed rolling window method allows for smooth processing of multiple demonstrations containing positions, orientations, forces and torques. Thirdly, the proposed orientation DMPs corrections prevent singularities and artefacts that could occur in their original formulation. Relevant future work would be constraint frame identification for hybrid motion/force control along any geometry and the use of computer vision models to automatically update trajectories based on changes in the environment.

References

1. Sharma N, Pandey JK, Mondal S. A Review of Mobile Robots: Applications and Future Prospect. International Journal of Precision Engineering and Manufacturing 2023 Sep;24(9):1695–1706. https://doi.org/10.1007/s12541-023-00876-7.
2. Lee I. Service Robots: A Systematic Literature Review. Electronics 2021 Oct;10(21):2658. https://www.mdpi.com/2079-9292/10/21/2658.
3. Billard AG, Calinon S, Dillmann R. Learning from Humans. In: Siciliano B, Khatib O, editors. Springer Handbook of Robotics Cham: Springer International Publishing; 2016.p. 1995–2014. https://doi.org/10.1007/978-3-319-32552-1_74.
4. Ravichandar H, Polydoros AS, Chernova S, Billard A. Recent Advances in Robot Learning from Demonstration. Annual Review of Control, Robotics, and Autonomous Systems 2020 May;3(1):297–330. https://www.annualreviews.org/doi/10.1146/annurev-control-100819-063206.
5. Ijspeert AJ, Nakanishi J, Schaal S. Trajectory formation for imitation with nonlinear dynamical systems. In: IEEE International Conference on Intelligent Robots and Systems, vol. 2; 2001. .
6. Pastor P, Kalakrishnan M, Meier F, Stulp F, Buchli J, Theodorou E, et al. From dynamic movement primitives to associative skill memories. Robotics and Autonomous Systems 2013 Apr;61(4):351–361. https://linkinghub.elsevier.com/retrieve/pii/S0921889012001716.
7. Ude A, Nemec B, Petric T, Morimoto J. Orientation in Cartesian space dynamic movement primitives. In: 2014 IEEE International Conference on Robotics and Automation (ICRA) Hong Kong, China: IEEE; 2014. p. 2997–3004. http://ieeexplore.ieee.org/document/6907291/.
8. Koutras L, Doulgeri Z. A correct formulation for the Orientation Dynamic Movement Primitives for robot control in the Cartesian space. In: Proceedings of the Conference on Robot Learning Proceedings of Machine Learning Research, Osaka, Japan: PMLR; 2019. p. 293–302. https://proceedings.mlr.press/v100/koutras20a.html.
9. Deniša M, Petrič T, Gams A, Ude A. A Review of Compliant Movement Primitives. In: Robot Control; 2016.
10. Hogan N. Impedance Control: An Approach to Manipulation. In: 1984 American Control Conference San Diego, CA, USA: IEEE; 1984. p. 304–313. https://ieeexplore.ieee.org/document/4788393/.
11. Abu-Dakka FJ, Saveriano M. Variable Impedance Control and Learning—A Review. Frontiers in Robotics and AI 2020 Dec;7:590681. https://www.frontiersin.org/articles/10.3389/frobt.2020.590681/full.
12. Enayati N, Mariani S, Wahrburg A, Zanchettin AM. Variable-Impedance and Force Control for Robust Learning of Contact-rich Manipulation Tasks from User Demonstration. IFAC-PapersOnLine 2020;53(2):9834–9840. https://linkinghub.elsevier.com/retrieve/pii/S2405896320334492.
13. Raibert MH, Craig JJ. Hybrid Position/Force Control of Manipulators. Journal of Dynamic Systems, Measurement, and Control 1981 Jun;103(2):126–133. https://asmedigitalcollection.asme.org/dynamicsystems/article/103/2/126/400298/Hybrid-PositionForce-Control-of-Manipulators.
14. Osa T, Pajarinen J, Neumann G, Bagnell JA, Abbeel P, Peters J. An Algorithmic Perspective on Imitation Learning. Foundations and Trends in Robotics 2018;7(1-2).
15. Adams S, Cody T, Beling PA. A survey of inverse reinforcement learning. Artificial Intelligence Review 2022 Aug;55(6):4307–4346. https://doi.org/10.1007/s10462-021-10108-x.
16. Kroemer O, Niekum S, Konidaris G, A Review of Robot Learning for Manipulation: Challenges, Representations, and Algorithms. arXiv; 2020. http://arxiv.org/abs/1907.03146, arXiv:1907.03146 [cs].
17. Paraschos A, Daniel C, Peters J, Neumann G. Probabilistic Movement Primitives. Advances in Neural Information Processing Systems 2013;.
18. Calinon S, Guenter F, Billard A. On Learning, Representing, and Generalizing a Task in a Humanoid Robot. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 2007 Apr;37(2):286–298. http://ieeexplore.ieee.org/document/4126276/.
19. Calinon S. A tutorial on task-parameterized movement learning and retrieval. Intelligent Service Robotics 2016;9(1).
20. Khansari-Zadeh SM, Billard A. Imitation learning of globally stable non-linear point-to-point robot motions using nonlinear programming. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems Taipei: IEEE; 2010. p. 2676–2683. http://ieeexplore.ieee.org/document/5651259/.
21. Jahn L, Wörgötter F, Kulvicius T, Comparison of Motion Encoding Frameworks on Human Manipulation Actions. arXiv; 2024. http://arxiv.org/abs/2211.13024, arXiv:2211.13024 [cs].
22. Abu-Dakka FJ, Rozo L, Caldwell DG. Force-based variable impedance learning for robotic manipulation. Robotics and Autonomous Systems 2018 Nov;109:156–167. https://linkinghub.elsevier.com/retrieve/pii/S0921889018300125.
23. Arduengo M, Colomé A, Lobo-Prat J, Sentis L, Torras C.

Gaussian-process-based robot learning from demonstration. Journal of Ambient Intelligence and Humanized Computing 2023 Feb;https://link.springer.com/10.1007/s12652-023-04551-7.

24. Zhang Y, Zhao F, Liao Z. A DMP-based Framework for Efficiently Generating Complete Stiffness Profiles of Human-like Variable Impedance Skill from Demonstrations; 2021.
25. Wu R, Zhang H, Zhu Y, Zang X, Zhao J. Impedance adjustment for learning human variable impedance skill. Mechatronics 2023 Nov;95:103036. https://linkinghub.elsevier.com/retrieve/pii/S0957415823000922.
26. Kormushev P, Calinon S, Caldwell DG. Imitation Learning of Positional and Force Skills Demonstrated via Kinesthetic Teaching and Haptic Input. Advanced Robotics 2011 Jan;25(5):581–603. https://www.tandfonline.com/doi/full/10.1163/016918611X558261.
27. Steinmetz F, Montebelli A, Kyrki V. Simultaneous kinesthetic teaching of positional and force requirements for sequential in-contact tasks. In: IEEE-RAS International Conference on Humanoid Robots, vol. 2015-December; 2015. ISSN: 21640580.
28. Rozo L, Jiménez P, Torras C. A robot learning from demonstration framework to perform force-based manipulation tasks. Intelligent Service Robotics 2013;6(1).
29. Le AT, Guo M, Duijkeren NV, Rozo L, Krug R, Kupcsik AG, et al. Learning Forceful Manipulation Skills from Multi-modal Human Demonstrations. In: IEEE International Conference on Intelligent Robots and Systems; 2021. ISSN: 21530866.
30. Wang N, Chen C, Nuovo AD. A Framework of Hybrid Force/Motion Skills Learning for Robots. IEEE Transactions on Cognitive and Developmental Systems 2021 Mar;13(1):162–170. https://ieeexplore.ieee.org/document/8964480/.
31. Dynamic Time Warping. In: Müller M, editor. Information Retrieval for Music and Motion Berlin, Heidelberg: Springer; 2007.p. 69–84. https://doi.org/10.1007/978-3-540-74048-3_4.
32. Huynh DQ. Metrics for 3D Rotations: Comparison and Analysis. Journal of Mathematical Imaging and Vision 2009 Oct;35(2):155–164. http://link.springer.com/10.1007/s10851-009-0161-2.
33. Schneider M, Ertel W. Robot Learning by Demonstration with local Gaussian process regression. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems; 2010. p. 255–260. https://ieeexplore.ieee.org/document/5650949, iSSN: 2153-0866.
34. Kim S, Haschke R, Ritter H. Gaussian Mixture Model for 3-DoF orientations. Robotics and Autonomous Systems 2017 Jan;87:28–37. https://linkinghub.elsevier.com/retrieve/pii/S0921889015302244.
35. Nobach H, Practical Realization of Bessel's Correction for a Bias-Free Estimation of the Auto-Covariance and the Cross-Covariance Functions. arXiv; 2023. http://arxiv.org/abs/2303.11047, arXiv:2303.11047 [eess, stat].
36. Markley FL, Cheng Y, Crassidis JL, Oshman Y. Averaging Quaternions. Journal of Guidance, Control, and Dynamics 2007 Jul;30(4):1193–1197. https://arc.aiaa.org/doi/10.2514/1.28949.
37. Pastor P, Hoffmann H, Asfour T, Schaal S. Learning and generalization of motor skills by learning from demonstration. In: Proceedings - IEEE International Conference on Robotics and Automation; 2009. ISSN: 10504729.
38. Ijspeert AJ, Nakanishi J, Hoffmann H, Pastor P, Schaal S. Dynamical movement primitives: learning attractor models for motor behaviors. Neural Computation 2013 Feb;25(2):328–373.
39. Hoffmann H, Pastor P, Park DH, Schaal S. Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance; 2009. p. 2587–2592.
40. Saveriano M, Abu-Dakka FJ, Kramberger A, Peternel L. Dynamic Movement Primitives in Robotics: A Tutorial Survey. The International Journal of Robotics Research 2023 Nov;42(13):1133–1184. http://arxiv.org/abs/2102.03861, arXiv:2102.03861 [cs].
41. Pastor P, Righetti L, Kalakrishnan M, Schaal S. Online movement adaptation based on previous sensor experiences. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems San Francisco, CA: IEEE; 2011. p. 365–371. http://ieeexplore.ieee.org/document/6095059/.
42. Iskandar M, Ott C, Albu-Schäffer A, Siciliano B, Dietrich A. Hybrid Force-Impedance Control for Fast End-Effector Motions. IEEE Robotics and Automation Letters 2023 Jul;8(7):3931–3938. https://ieeexplore.ieee.org/document/10107744/.
43. Hogan N. Impedance Control: An Approach to Manipulation: Part I—Theory. Journal of Dynamic Systems, Measurement, and Control 1985 Mar;107(1):1–7. https://asmedigitalcollection.asme.org/dynamicsystems/article/107/1/1/400604/Impedance-Control-An-Approach-to-Manipulation-Part.
44. Albu-Schaffer A, Ott C, Frese U, Hirzinger G. Cartesian impedance control of redundant robots: recent results with the DLR-light-weight-arms. In: 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422), vol. 3 Taipei, Taiwan: IEEE; 2003. p. 3704–3709. http://ieeexplore.ieee.org/document/1242165/.
45. Xiong G, Zhou Y, Yao J. Null-space impedance control of 7-degree-of-freedom redundant manipulators based on the arm angles. International Journal of Advanced Robotic Systems 2020 May;17(3):1729881420925297. https://doi.org/10.1177/1729881420925297, publisher: SAGE Publications.
46. Quigley M, Gerkey B, Conley K, Faust J, Foote T, Leibs J, et al. ROS: an open-source Robot Operating System. ICRA Workshop on Open Source Software 2009;.
47. Subramani G, Zinn M, Gleicher M. Inferring geometric constraints in human demonstrations. In: Proceedings of The 2nd Conference on Robot Learning PMLR; 2018. p. 223–236. https://proceedings.mlr.press/v87/subramani18a.html, iSSN: 2640-3498.
48. Dae-Hyung Park, Hoffmann H, Pastor P, Schaal S. Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In: Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots Daejeon: IEEE; 2008. p. 91–98. http://ieeexplore.ieee.org/document/4755937/.
49. Sidiropoulos A, Doulgeri Z, Dynamic via-points and improved spatial generalization for online trajectory planning with Dynamic Movement Primitives. arXiv; 2023. http://arxiv.org/abs/2212.13473, arXiv:2212.13473 [cs].

A Quaternion Operations

Quaternions $\mathbf{q}$ consist of a real scalar part $w \in \mathbb{R}$ and an imaginary vector part $\mathbf{v} \in \mathbb{R}^3$. They can be written as:

$$\mathbf{q} = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}, \tag{28}$$

with $[x\ y\ z]^T = \mathbf{v}$ and imaginary basis vectors $\mathbf{i}, \mathbf{j}, \mathbf{k}$. Unit quaternions are quaternions that are constrained to have a magnitude of 1 ($||\mathbf{q}||^2 = w^2 + x^2 + y^2 + z^2 = 1$) and lay therefore on the surface of a sphere in $\mathbb{R}^4$, denoted by $\mathbb{S}^3$. They can be used to represent orientations in a compact, singularity-free way.
Unit quaternions can easily be expressed with the axis-angle representation of a 3D rotation using Euler's formula, which generalizes to quaternions:

$$\mathbf{q} = e^{\frac{\theta}{2}(n_x\mathbf{i}+n_y\mathbf{j}+n_z\mathbf{k})} = \cos\frac{\theta}{2} + \sin\frac{\theta}{2}(n_x\mathbf{i} + n_y\mathbf{j} + n_z\mathbf{k}), \tag{29}$$

which has unit rotation axis $\mathbf{n} = (n_x, n_y, n_z)$ and rotation angle θ. It shows that $\mathbf{q}$ and $-\mathbf{q}$ describe the same rotation, which is known as the quaternion's double cover of $SO(3)$.
The product of two quaternions is called the Hamilton product and is defined as follows:

$$\mathbf{q}_1 * \mathbf{q}_2 = \begin{bmatrix} w_1 w_2 - \mathbf{v}_1^T \mathbf{v}_2 \\ w_1\mathbf{v}_2 + w_2\mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2 \end{bmatrix} \tag{30}$$

This product is not commutative, but it is associative. The inverse of a unit quaternion is equal to its conjugate:

$$\mathbf{q}^{-1} = \overline{\mathbf{q}} = \begin{bmatrix} w & -\mathbf{v}^T \end{bmatrix}^T. \tag{31}$$

Given two unit quaternions $\mathbf{q}_1$ and $\mathbf{q}_2$, the rotation to go from $\mathbf{q}_1$ to $\mathbf{q}_2$ is calculated with

$$\Delta\mathbf{q} = \frac{\mathbf{q}_2}{\mathbf{q}_1} = \mathbf{q}_2 * \mathbf{q}_1^{-1} = \mathbf{q}_2 * \overline{\mathbf{q}_1}. \tag{32}$$

The logarithm of a unit quaternion is defined as:

$$\mathbf{v} = \log(\mathbf{q}) = \begin{cases} \frac{\theta}{2}\mathbf{n}, & \text{if } \theta \neq 0 \\ \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T, & \text{if } \theta = 0 \end{cases} \tag{33}$$

which can be obtained from (29), where $\frac{\theta}{2}$ can be obtained with $\arccos(w)$. This is a three-dimensional vector, which can be represented as a purely imaginary quaternion with $\mathbf{q} = 0 + \mathbf{v}$. Note that since quaternion multiplication is non-commutative, $\log(\mathbf{q}_1\mathbf{q}_2) \neq \log(\mathbf{q}_1) + \log(\mathbf{q}_2)$. The exponential map is defined as:

$$\mathbf{q} = \exp(\mathbf{v}) = \begin{cases} \begin{bmatrix} \cos(||\mathbf{v}||) & \sin(||\mathbf{v}||)\frac{\mathbf{v}^T}{||\mathbf{v}||} \end{bmatrix}^T, & \text{if } ||\mathbf{v}|| \neq 0 \\ \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T, & \text{if } ||\mathbf{v}|| = 0 \end{cases} \tag{34}$$

The logarithm of the unit quaternion can be used to define a distance metric between two unit quaternions:

$$d(\mathbf{q}_1, \mathbf{q}_2) = ||2\log(\mathbf{q}_2 * \overline{\mathbf{q}_1})||, \tag{35}$$

which returns the rotation angle $\theta \in [0, 2\pi)$, and is equal to the length of an arc connecting the two points on the unit sphere [32]. Note that if the real part of $\mathbf{q}_2 * \overline{\mathbf{q}_1}$ is negative, the rotation angle will be greater than π rad. In that case, negate one of the quaternions to find the shortest arc, known as the geodesic.

To obtain the (weighted) average quaternion, the eigen-decomposition method proposed by Markley et al. [36] can be used. Given n unit quaternions $\mathbf{q}_i$ and corresponding weights w_i, we can construct the weighted auto-correlation matrix $\mathbf{C} := \sum_{i=1}^{n} w_i \mathbf{q}_i \mathbf{q}_i^T$. The unweighted average quaternion is obtained if all weights are equal to 1. The average quaternion $\bar{\mathbf{q}}$ can be found by solving the maximization problem

$$\bar{\mathbf{q}} = \arg\max_{\mathbf{q} \in \mathbb{S}^3} \mathbf{q}^T \mathbf{C} \mathbf{q}, \tag{36}$$

whose solution is the eigenvector associated with the maximum eigenvalue of $\mathbf{M}$. Note the difference between the quaternion average $\bar{\mathbf{q}}$ and the quaternion conjugate $\overline{\mathbf{q}}$.

4 General Discussion and Conclusion

The goal of this research was to develop an LfD framework that allows the robot to generalize learned tasks to changes in the environment, commands the robot to be more compliant when variability in demonstrations is high, and allows the robot to reproduce both motion and contact tasks. The designed framework was implemented on a real robot and validated by learning and reproducing two practical tasks. The results of the experiments confirm that the framework possesses the desired properties.

4.1 Limitations

However, the experiments did show a poor tracking performance for rotations during the watering task and for forces during the drawing task. The poor rotation tracking is likely caused by inaccurate configuration of the inertial properties of the watering can. Tilting the can causes the center of mass (COM) to move to the side, resulting in a moment that tries to counteract the tilt. The spring-damper nature of the impedance controller then results in an equilibrium position that does not match the reference, leading to an increased tracking error. This could be improved by configuring the end effector mass and COM more accurately, allowing the gravity compensation to counteract the moment of the watering can instead of the impedance controller.

The relatively poor force tracking could potentially be improved by implementing force feedback control. Unfortunately, during the development of the hybrid controller, I did not have access to the force torque sensor yet. It was only attached near the end of my research. Therefore, I could use it to record forces during demonstrations, but I did not have time to rewrite the controller to incorporate force measurements for feedback control. Franka's external wrench estimate, obtained by transforming the measured joint torques to the end effector frame using the Jacobian, was available, but quick tests indicated that the wrench estimate was not sufficiently accurate and that closed-loop force control did not improve the force reproduction. Additionally, as mentioned in the paper, force feedback control may introduce issues when large collision forces are present. Still, it would be valuable to implement force feedback control based on the force measurements of the force/torque sensor and compare it to feedforward control.

Another limitation, which was not mentioned in the research paper, is the unverified stability of the variable impedance controller. While constant stiffness impedance control is passive and therefore stable, it is possible that varying impedance parameters can result in unstable behavior. However, according to Kronander et al., reasonable varying stiffness profiles should not show destabilizing tendencies [18]. Moreover, controller instability was not encountered during experiments with the framework. Nevertheless, controller stability must be guaranteed before the framework can be used in real applications. Kronander et al. also proposed a state-independent stability constraint that guarantees stable variable stiffness profiles [18]. Guaranteed stability through passivity may also be implemented using a virtual energy tank [19]. Unfortunately, due to time constraints, I did not manage to implement their proposed verification and potential modification of the stiffness profile. This should be implemented and discussed before a potential publication of the paper.

4.2 Future Research

The framework contains many hyperparameters that require tuning. These are described in Appendix B, in addition to their effect, recommended tuning method, and used value during this study. In future work, it would be interesting to explore and compare the effects of different parameter values.

To further enhance the versatility of the framework, it should be paired with a method to identify constraint geometry. This would allow the framework to encode motion and forces in the constraint frame. The motion and force profiles can then be reproduced along any geometry, rather than being limited to geometry aligned to the robot's base frame which is the case now.

Furthermore, by combining the framework with a computer vision model capable of segmenting and estimating the pose of objects in the environment, the robot would be able to automatically alter the DMP goal states during reproduction based on the robot's perceptions. Currently, this adjustment has to be done manually, which limits the robot's autonomous behavior.

4.3 Conclusion

In this thesis, an LfD framework was developed, capable of reproducing both motion- and contact-based tasks from one or multiple demonstrations. The framework employs a geometry-aware inference method combined with a standard deviation approach to estimate time-dependent Gaussian distributions for each Cartesian dimension. These means are encoded within an orientation DMP model, which ensures smooth reproduction and generalization of end-effector translations and orientations without artefacts or singularities. A hybrid force-impedance controller enables the simultaneous reproduction of motion and force, with variable impedance stiffness automatically inferred from the variance between demonstrations. This balances tracking performance and robot compliance. The framework was validated on real-world tasks, demonstrating its practicality.

Future work should focus on identifying constraint frames for hybrid motion-force control along arbitrary geometries and incorporating computer vision models to automatically adapt trajectories based on changes in the environment.

References

[1] G. Singh and V. Banga, “Robots and its types for industrial applications,” en, *Materials Today: Proceedings*, vol. 60, pp. 1779–1786, 2022, ISSN: 22147853. DOI: 10.1016/j.matpr.2021.12.426. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2214785321081682 (visited on 07/15/2024).

[2] I. Lee, “Service Robots: A Systematic Literature Review,” en, *Electronics*, vol. 10, no. 21, p. 2658, Oct. 2021, ISSN: 2079-9292. DOI: 10.3390/electronics10212658. [Online]. Available: https://www.mdpi.com/2079-9292/10/21/2658 (visited on 07/15/2024).

[3] A. G. Billard, S. Calinon, and R. Dillmann, “Learning from Humans,” en, in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds., Cham: Springer International Publishing, 2016, pp. 1995–2014, ISBN: 978-3-319-32552-1. DOI: 10.1007/978-3-319-32552-1_74. [Online]. Available: https://doi.org/10.1007/978-3-319-32552-1_74 (visited on 09/07/2024).

[4] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, “Recent Advances in Robot Learning from Demonstration,” en, *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 297–330, May 2020, ISSN: 2573-5144, 2573-5144. DOI: 10.1146/annurev-control-100819-063206. [Online]. Available: https://www.annualreviews.org/doi/10.1146/annurev-control-100819-063206 (visited on 07/18/2024).

[5] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, “An Algorithmic Perspective on Imitation Learning,” *Foundations and Trends in Robotics*, vol. 7, no. 1-2, 2018, ISSN: 1935-8253. DOI: 10.1561/2300000053.

[6] A. J. Ijspeert, J. Nakanishi, and S. Schaal, “Trajectory formation for imitation with nonlinear dynamical systems,” in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2, 2001. DOI: 10.1109/iros.2001.976259.

[7] A. Ude, B. Nemec, T. Petric, and J. Morimoto, “Orientation in Cartesian space dynamic movement primitives,” en, in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China: IEEE, May 2014, pp. 2997–3004, ISBN: 978-1-4799-3685-4. DOI: 10.1109/ICRA.2014.6907291. [Online]. Available: http://ieeexplore.ieee.org/document/6907291/ (visited on 06/27/2024).

[8] M. Deniša, T. Petrič, A. Gams, and A. Ude, “A Review of Compliant Movement Primitives,” in *Robot Control*, 2016. DOI: 10.5772/64058.

[9] M. J. Pollayil, F. Angelini, G. Xin, *et al.*, “Choosing Stiffness and Damping for Optimal Impedance Planning,” English, *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 1281–1300, Apr. 2023, Publisher: Institute of Electrical and Electronics Engineers Inc., ISSN: 1941-0468. DOI: 10.1109/TRO.2022.3216078. [Online]. Available: https://www.research.ed.ac.uk/en/publications/choosing-stiffness-and-damping-for-optimal-impedance-planning (visited on 08/31/2024).

[10] N. Hogan, “Impedance Control: An Approach to Manipulation,” en, in *1984 American Control Conference*, San Diego, CA, USA: IEEE, Jul. 1984, pp. 304–313. DOI: 10.23919/ACC.1984.4788393. [Online]. Available: https://ieeexplore.ieee.org/document/4788393/ (visited on 06/25/2024).

[11] L. Koutras and Z. Doulgeri, “A correct formulation for the Orientation Dynamic Movement Primitives for robot control in the Cartesian space,” en, in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, Osaka, Japan: PMLR, Nov. 2019, pp. 293–302. [Online]. Available: https://proceedings.mlr.press/v100/koutras20a.html.

[12] Y. Mohammad and T. Nishida, “Tackling the Correspondence Problem,” en, in *Active Media Technology*, D. Hutchison, T. Kanade, J. Kittler, *et al.*, Eds., vol. 8210, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2013, pp. 84–95, ISBN: 978-3-319-02749-4 978-3-319-02750-0. DOI: 10.1007/978-3-319-02750-0_9. [Online]. Available: http://link.springer.com/10.1007/978-3-319-02750-0_9 (visited on 07/18/2024).

[13] Y. Yang, C. Hua, J. Li, and X. Guan, “Finite-time output-feedback synchronization control for bilateral teleoperation system via neural networks,” *Information Sciences*, vol. 406-407, pp. 216–233, Sep. 2017, ISSN: 0020-0255. DOI: 10.1016/j.ins.2017.04.034. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025517306904 (visited on 07/18/2024).

[14] W. Si, N. Wang, and C. Yang, "A review on manipulation skill acquisition through teleoperation-based learning from demonstration," en, *Cognitive Computation and Systems*, vol. 3, no. 1, pp. 1–16, 2021, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1049/ccs2.12005, ISSN: 2517-7567. DOI: 10.1049/ccs2.12005. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1049/ccs2.12005 (visited on 07/18/2024).

[15] P. Kormushev, S. Calinon, and D. G. Caldwell, "Imitation Learning of Positional and Force Skills Demonstrated via Kinesthetic Teaching and Haptic Input," en, *Advanced Robotics*, vol. 25, no. 5, pp. 581–603, Jan. 2011, ISSN: 0169-1864, 1568-5535. DOI: 10.1163/016918611X558261. [Online]. Available: https://www.tandfonline.com/doi/full/10.1163/016918611X558261 (visited on 06/25/2024).

[16] J. Diebel, "Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors," *Matrix*, vol. 58, Jan. 2006.

[17] D. Q. Huynh, "Metrics for 3D Rotations: Comparison and Analysis," en, *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 155–164, Oct. 2009, ISSN: 0924-9907, 1573-7683. DOI: 10.1007/s10851-009-0161-2. [Online]. Available: http://link.springer.com/10.1007/s10851-009-0161-2 (visited on 06/25/2024).

[18] K. Kronander and A. Billard, "Stability Considerations for Variable Impedance Control," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1298–1305, Oct. 2016, ISSN: 1552-3098, 1941-0468. DOI: 10.1109/TRO.2016.2593492. [Online]. Available: http://ieeexplore.ieee.org/document/7560657/ (visited on 07/28/2024).

[19] E. Shahriari, A. Kramberger, A. Gams, A. Ude, and S. Haddadin, "Adapting to contacts: Energy tanks and task energy for passivity-based dynamic movement primitives," en, in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, Birmingham: IEEE, Nov. 2017, pp. 136–142, ISBN: 978-1-5386-4678-6. DOI: 10.1109/HUMANOIDS.2017.8239548. [Online]. Available: http://ieeexplore.ieee.org/document/8239548/ (visited on 07/22/2024).

[20] "Dynamic Time Warping," en, in *Information Retrieval for Music and Motion*, M. Müller, Ed., Berlin, Heidelberg: Springer, 2007, pp. 69–84, ISBN: 978-3-540-74048-3. DOI: 10.1007/978-3-540-74048-3_4. [Online]. Available: https://doi.org/10.1007/978-3-540-74048-3_4 (visited on 07/27/2024).

[21] M. Cuturi and M. Blondel, *Soft-DTW: A Differentiable Loss Function for Time-Series*, en, arXiv:1703.01541 [stat], Feb. 2018. [Online]. Available: http://arxiv.org/abs/1703.01541 (visited on 07/27/2024).

[22] S. Calinon, F. Guenter, and A. Billard, "On Learning, Representing, and Generalizing a Task in a Humanoid Robot," en, *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 286–298, Apr. 2007, ISSN: 1083-4419. DOI: 10.1109/TSMCB.2006.886952. [Online]. Available: http://ieeexplore.ieee.org/document/4126276/ (visited on 07/15/2024).

[23] M. Arduengo, A. Colomé, J. Lobo-Prat, L. Sentis, and C. Torras, "Gaussian-process-based robot learning from demonstration," en, *Journal of Ambient Intelligence and Humanized Computing*, Feb. 2023, ISSN: 1868-5137, 1868-5145. DOI: 10.1007/s12652-023-04551-7. [Online]. Available: https://link.springer.com/10.1007/s12652-023-04551-7 (visited on 06/25/2024).

A Disclaimer on the usage of AI Tools

This disclaimer on the usage of Artificial Intelligence tools was included following the guidelines of the University of Twente on the use of AI in Education:

During the preparation of this work, the author used the following tools for the following reasons:

1. **ChatGPT:** ChatGPT was used sparingly to help improve the academic tone of certain sentences. It served solely for inspirational purposes and all text in this report has been written by the author. Neither ChatGPT nor any other LLM was used as a source of information or to generate chunks of text.
2. **Grammarly:** Grammarly was used to check and improve the grammar and spelling of the text of this report.

After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the content of the work.

B Hyperparameters

In this framework, there are various hyperparameters to set and choices to make. While some values will have a clear optimum, others will require tuning and design choices. This section gives an overview of the hyperparameters present in this framework per component, a description of the effect of each parameter, a proposed method of determining the value, and the final choice used in the experimental evaluation.

Component	Parameter	Effect	How to determine?	Choice
Data Acquisition	Sample rate	Sample rate of recording demonstrations. Too high will result in superfluous data and increased computation time. Too low will result in choppy demonstrations and potential loss of information.	Set to some reasonable value. Both the robot and the FT sensor need to be able to send data at this sample rate.	100 Hz
Temporal Alignment	Base demonstration for alignment	Determines the shape of the task reproduction, as all other demonstrations will be warped to match this one.	An optimal choice is difficult[1].	Use the first loaded demonstration as base.
	Temporal Alignment Method	Determines how the demonstrations are aligned to each other. Choices include linear time scaling, DTW[20], and soft-DTW[21].	Choose based on the desired properties of the method.	DTW, as it follows literature and does not introduce an extra hyperparameter like soft-DTW (γ).
	Soft-DTW γ	Determines smoothness of DTW result. $\gamma = 0$ resembles DTW. Increasing values of γ smooth the warping path until it resembles linear time scaling.	Choose based on desired smoothness.	Soft-DTW is not used, so neither is this parameter.
	Dimension distance weights	Determines weighing/prioritizing of distance per dimension (translation, orientation, force, torque).	Choose based on prioritized dimension. The range of values per dimension needs to be taken into account[2].	$W^P = 1.0$ $W^O = 0.11$ $W^F = 0.05$ $W^T = 0.07$
Inference and Variance Identification	Inference Method	Determines how the representative demonstration is inferred from the synchronized demonstrations. Choices are Rolling Window (proposed in this thesis), GMR[22], and GPR[23]	Chosen method should be geometry-aware. If it is, the choice can be made based on the number of parameters, reproduction accuracy and method of variance identification.	Rolling Window
	Rolling Window window length	Determines smoothness of inferred demonstration	Window should be large enough to contain multiple samples. Windows of a larger length act as a low-pass filter with lower cut-off frequency.	0.04 s

continues on next page

Component	Parameter	Effect	How to determine?	Choice
	Rolling Window element weighing function	Determines how samples in the window are weighted. It can be constant to disable weighted average and standard deviation.	When continuous demonstrations are desired, the weighing function must smoothly go to zero at the edges of the window.	4-th order polynomial that ensures the weight and its derivative are zero at the edge of the window.
	GMM number of Gaussians	Balances reproduction accuracy, computation time, and generalizability to new situations (risk of overfitting)	An information criterion for automatic selection, such as the BIC.	GMM is not used, so neither is this parameter.
Dynamic Movement Primitive	DS linear stiffness	Determines the time constant of the dynamic system.	Should aim for low values to 'stretch' the movement over the whole duration, while still converging 'sufficiently'[3] close to the target position.	100
	Number of Gaussian kernels	Balance between computation time and approximation accuracy.	Lower number reduces computation time and increases smoothness, at the cost of approximation accuracy. Longer demonstrations require more Gaussians to capture the same signal frequencies.	500
	Gaussian centers and widths	Determines how the Gaussians are distributed over the demonstration.	The centers should be distributed equally over time, meaning that they are logarithmically spaced in the phase domain. The widths determine the overlap between Gaussians.	$c_i = e^{\left(-\alpha \frac{i-1}{N-1}\right)}$ $h_i = \frac{1}{(c_{i+1}-c_i)^2}$ [7]
	Numerical integration method	Determines the integration method. Affects speed, accuracy and stability of integration	Determine based on accuracy and computation time requirements. Experimenting with different methods is recommended.	Runge-Kutta method of order 5 (4).
	Integration maximum timestep	Sets maximum allowed step size (of time) for numerical integration. Low values give a smoother output of states but increase computation time.	The integration method by default has an adaptive step size. If the error is low, it will increase the step size until the maximum allowed step. If the step size is too large, robot movement will be choppy.	0.1 s
	Orientation DMP singularity avoidance region	Determines the region around the quaternion error singularity of 2π that should be avoided.	Experiment to find a suitable value. Values that are too small will fail to prevent oscillations, while large values will result in overly conservative behavior, reducing the system's ability to reproduce orientation trajectories.	0.02 rad

continues on next page

Component	Parameter	Effect	How to determine?	Choice
Controller	Translational impedance stiffness	Determines the balance between tracking accuracy and compliance (safety) of the robot.	Perform experiments to test tracking errors for different stiffness values. Based on tracking error, choose minimum, maximum, and constant constant stiffness and corresponding standard deviations.	$K_{min} = 40$ N/m $K_{max} = 600$ N/m $K_{cons} = 400$ N/m $\sigma_{max} = 0.05$ m $\sigma_{min} = 0.0025$ m
	Rotational impedance stiffness	Determines the balance between tracking accuracy and compliance (safety) of the robot.	Perform experiments to test tracking errors for different stiffness values. Based on tracking error, choose minimum, maximum, and constant constant stiffness and corresponding standard deviations.	$K_{min} = 3\ \frac{\text{Nm}}{\text{rad}}$ $K_{max} = 15\ \frac{\text{Nm}}{\text{rad}}$ $K_{cons} = 12\ \frac{\text{Nm}}{\text{rad}}$ $\sigma_{max} = \frac{\pi}{6}$ rad $\sigma_{min} = \frac{\pi}{90}$ rad
	Variable stiffness profile	Determines the function that relates standard deviation to an impedance stiffness.	Up to choice. The function should saturate at K_{max} and K_{min} for σ_{min} and σ_{max}, respectively.	Linear scaling between K_{max} and K_{min}
	Force control feedback gains	The feedback gains determine the control effort to counteract differences between desired and actual exerted force.	Should be empirically tuned based on tracking accuracy, responsiveness, noise sensitivity and stability.	Accurate force readings were not yet available in the controller, so feedback gains K_f^P and K_f^I were set to 0.

[1]Ideally, one 'mean' demonstration is created that is an average of the temporal shape of all demonstrations which is then used as the base demonstration. Otherwise, one could try all combinations and then choose the base demonstration which results in the least total warping, but this is very computationally expensive. Finally, one could choose an arbitrary base demonstration and accept that the resulting reproduction is dependent on the chosen base. Surprisingly, I could not find literature discussing this conundrum.

[2]Typical distance ranges using the Franka Research 3 Robot are: Translation distance = $[0, 1]$ m; Orientation distance = $[0, 2\pi]$ rad; Force distance = $[0, 15]$ N; Torque distance = $[0, 10]$ Nm. These differences in range implicitly weigh the different dimensions, which can be compensated for using a correct choice of explicit weights.

[3]The meaning of 'sufficiently' close is a design choice. High stiffness converges to the target faster, at the cost of larger accelerations. It should be decided how close the system should have converged to the target at the end of the movement.

C Experiment Data

C.1 Motion Task: Watering Plants

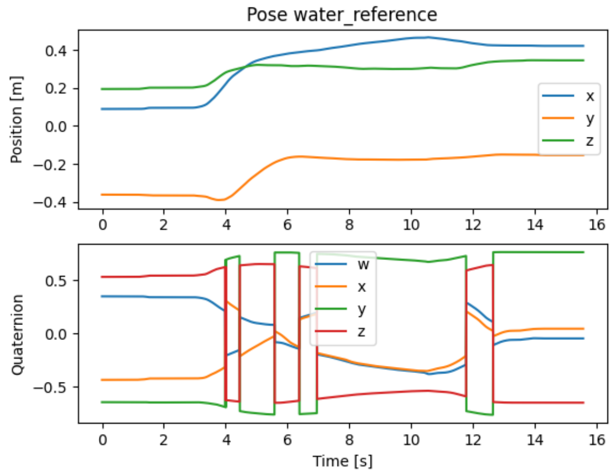

Figure 1: Full pose time series of the reference for the watering task.

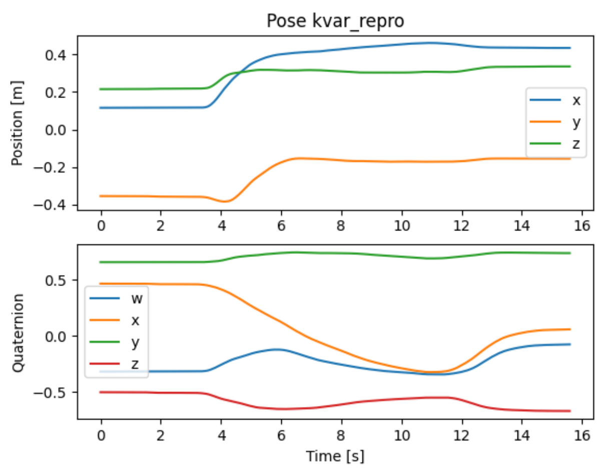

Figure 2: Full pose time series of the K_{var} reproduction for the watering task.

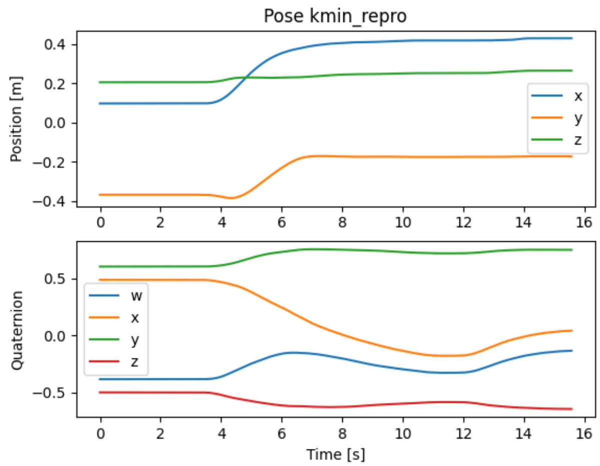

Figure 3: Full pose time series of the K_{min} reproduction for the watering task.

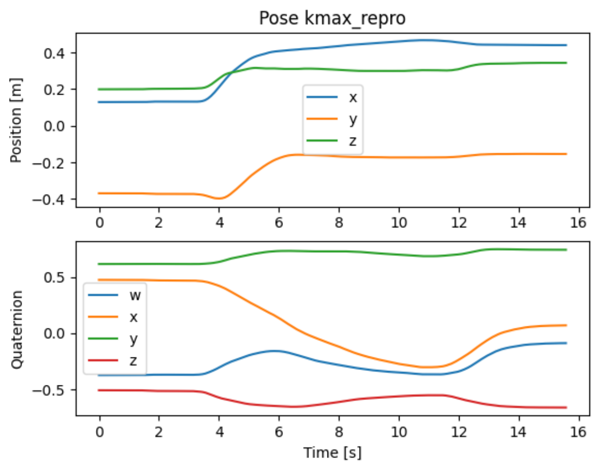

Figure 4: Full pose time series of the K_{max} reproduction for the watering task.

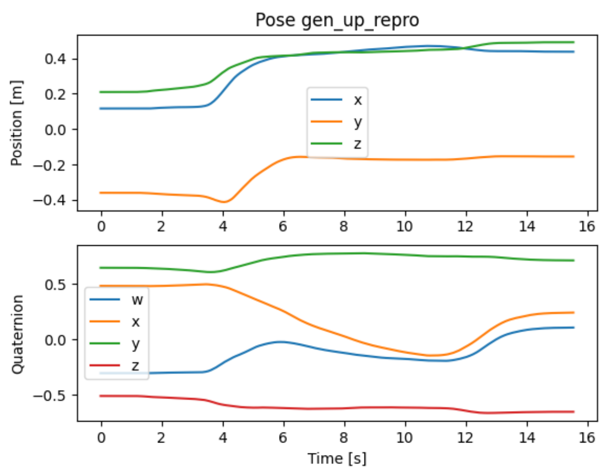

Figure 5: Full pose time series of the K_{var} generalized reproduction for the watering task.

C.2 Contact Task: Drawing on Whiteboard

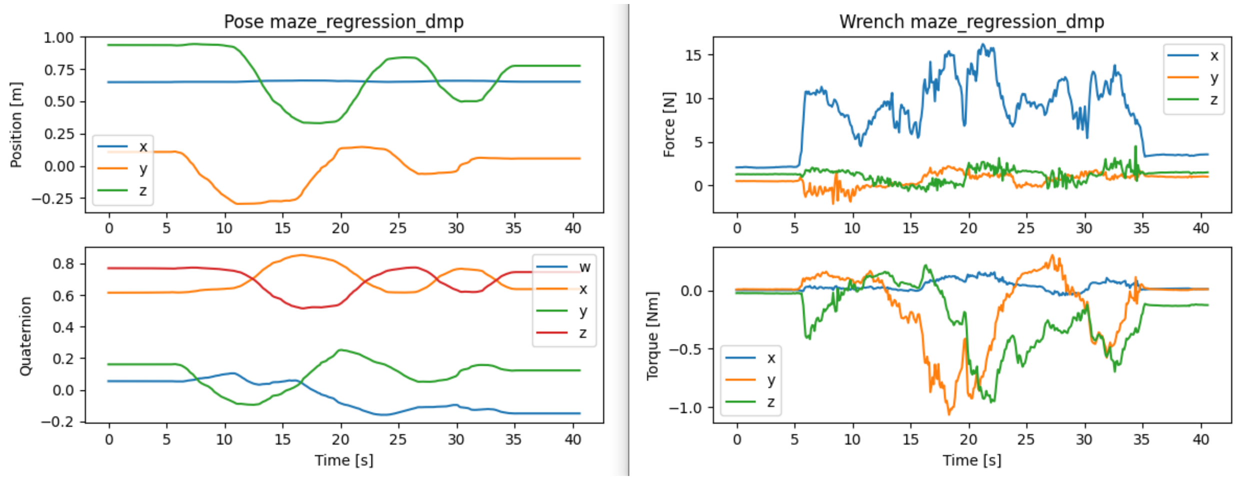

Figure 6: Full pose and wrench time series of the reference for the drawing task.

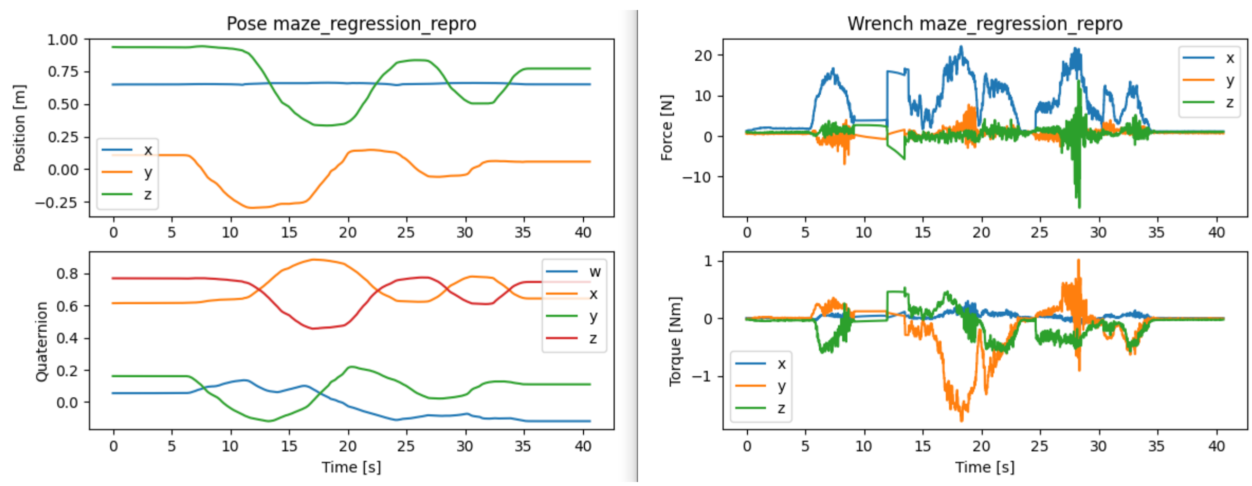

Figure 7: Full pose and wrench time series of the reproduction of the drawing task, which follows the reference of Figure 6.