Learning to Pick at Non-Zero-Velocity From Interactive Demonstrations

Anna Mészáros , Giovanni Franzese , and Jens Kober , *Senior Member, IEEE*

Abstract—**This work investigates how the intricate task of a continuous pick & place (P&P) motion may be learned from humans based on demonstrations and corrections. Due to the complexity of the task, these demonstrations are often slow and even slightly flawed, particularly at moments when multiple aspects (i.e., end-effector movement, orientation, and gripper width) have to be demonstrated at once. Rather than training a person to give better demonstrations, non-expert users are provided with the ability to interactively modify the dynamics of their initial demonstration through teleoperated corrective feedback. This in turn allows them to teach motions outside of their own physical capabilities. In the end, the goal is to obtain a faster but reliable execution of the task. The presented framework learns the desired movement dynamics based on the current Cartesian position with Gaussian Processes (GPs), resulting in a reactive, time-invariant policy. Using GPs also allows online interactive corrections and active disturbance rejection through epistemic uncertainty minimization. The experimental evaluation of the framework is carried out on a Franka-Emika Panda. Tests were performed to determine i) the framework's effectiveness in successfully learning how to quickly pick & place an object, ii) ease of policy correction to environmental changes (i.e., different object sizes and mass), and iii) the framework's usability for non-expert users.**

Index Terms—**Compliance and impedance control, imitation learning, incremental learning.**

I. INTRODUCTION

MORE often than not, robots employ a pick and place (P&P) strategy wherein they approach the object, stop and grip it and only then resume moving. We as humans, on the other hand, tend to pick things in a single fluent and quick motion. Of course, robots should also be able to complete a task fairly quickly, which in the case of P&P introduces a number of challenges, both from a control point of view [1] as well as a learning point of view [2].

Learning from Demonstration (LfD) has become a popular approach for allowing non-expert users to teach robots and thus more easily integrate them into the working and daily environment [3]. Yet these provided demonstrations are sub-optimal compared to what the robot might be able to achieve, e.g.,

demos having slower dynamics. Concurrently, it is important to consider that often, the execution of a task cannot simply be sped up uniformly. For example, when learning a P&P movement, retaining a high velocity when approaching the object can generate high impact forces which can cause the object to bounce away or topple over, potentially damaging the item in question as well as making it impossible to pick on time. We as people are able to identify such constraints and adapt accordingly, and can transfer this knowledge to the robot through demonstrations.

This work studies the feasibility of robot picking only using time-independent policies learned from human demonstrations and corrections. Our previous work [4] already revealed the effective application of minimum uncertainty GPs for learning variable impedance control in force application tasks like cleaning, plugging, and pushing. In none of the previous cases, however, were the dynamics of the end-effector (EE) orientation or gripper learned nor were there critical contact dynamics involved. Teaching more degrees of freedom while asking for fast performance makes the task of non-zero-velocity picking a challenging benchmark for studying the potential of learning from non-expert human teachers.

The main contributions of this work over the previous are:
1) Proposing a framework for interactively altering the speed and shape of robot motion dynamics in a decoupled manner through teleoperated correction.
2) A novel minimum uncertainty inference for learning the desired non-linear constraints of EE orientation and gripper width w.r.t. the EE position dynamics, while avoiding dangerous extrapolations.
3) Showing the benefit of uncertainty minimisation for enabling local motion consistency when dealing with critical precision tasks like fast picking, while being compliant in the interaction.
4) Extending the framework for generalizing to different object positions thanks to the parametrization w.r.t. moving reference frames.

Fig. 1 summarizes the three phases of learning in the teaching of a re-shelving operation: the initialization of the policy with kinesthetic demonstration, the shaping of the dynamics with teleoperated corrections, and the final evaluation of the autonomous task execution.

II. BACKGROUND AND RELATED WORK

When executing high-speed manipulation tasks which involve establishing contact with an object, it is important to consider the behaviour around the moment of impact. A reoccurring approach observed in existing works consists of adapting the relative velocity in order to mitigate the effects of the impact [5].
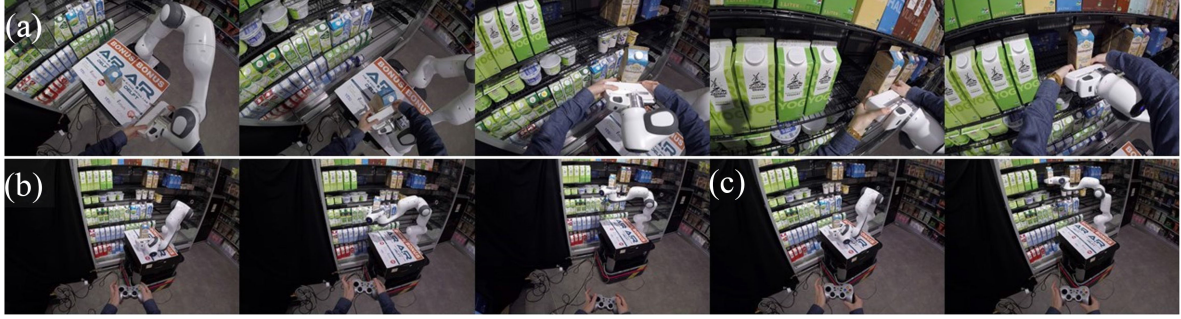
Fig. 1. Learning flow for teaching a robot how to reshelve an item.; a) starting with a single demonstration, followed by b) multiple rounds of correction after which, c) the robot is able to autonomously carry out the task.

Another strategy, which has been employed to absorb impacts particularly in catching tasks [6], involves utilising a follow-through behaviour which continues to track the predicted path of an object even after interception [7]. Alternatively, one can incorporate compliant behaviour into a provided attractor using impedance control [8]. While it is unable to mitigate the initial impact force irrespective of the set stiffness since the main contribution to this force is the velocity of the impacting objects, it is beneficial for absorbing the post-impact forces [9].

We can conclude that matching the velocity of an object likely achieves the best reduction of impact force, however, such an approach may not be optimal when considering the total time of the trajectory execution. This is especially true for static objects, wherein matching velocities would effectively bring the robot to a stand-still prior to the picking action. A better approach, therefore, is to interactively learn the feasible non-zero contact velocity while ensuring moderate impact forces.

Being able to adapt/correct the learned velocity with ease plays a key role in speeding up the overall execution of the demonstrated trajectory while also considering that the movement dynamics may require different degrees of adaptation at different points of the trajectory; for example slowing down prior to the moment of interception. Different works explore speed adaptation during trajectory execution using different function approximators. One approach involves altering the phase rate of probabilistic movement primitives (ProMPs) [10], whereas others propose a modified version of Dynamical Movement Primitives (DMPs) in which the speed is altered through an additional phase-dependent temporal scaling factor [11], or where the temporal scaling factor is changed through corrections and subsequently translated to changes in the learned dynamic movement [12]. The mentioned works modulate the velocity either using optimisation approaches or defined functions, or in the case of [12] where human corrections are used, the corrections are provided in a coupled manner for both the trajectory shape and speed. Our approach instead focuses on combining imitation learning and human interactive feedback [13] to provide corrections to speed and shape in a decoupled manner through teleoperation.

An alternative to phase-dependent methods, like DMPs, can be obtained as the formulation of the motion as a reactive controller according to

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) \qquad (1)$$

where $\boldsymbol{x}$ is the robot state and f identifies the transition of the robot state. GPs have been used for shaping a motion from human demonstrations through the local modification of a stable field [14]. However, none of the other works on learning state-dependent dynamical systems take into account the information of the uncertainty to increase motion consistency, and reduce covariate shift. Furthermore, in the context of interactive learning, we introduced the idea of decoupling the corrections of shape and velocity and investigated how this can be beneficial for allowing non-expert users to teach challenging tasks.

III. METHODOLOGY

The goal of this framework is to enable a user to teach the robot the desired motion through demonstration and teleoperated correction, see Algorithm 1. The robot is learning the desired minimum uncertainty dynamical system on the end-effector, formalized in Section III-A and the dynamics of the gripper orientation and width as a function of the current robot position, formalized in Section III-B. The main aim is to show that it is possible to learn a policy and later correct the velocity so as to achieve and surpass the performance of a skilled demonstrator. All of these aspects are modelled with Gaussian Processes, allowing interactive corrections of the dynamics and actions online, see Section III-C.

A. Learning a Minimum Uncertainty Dynamical System

A non-linear dynamical system can be described by (1). This type of formulation would fit perfectly in a velocity controller, however, due to the necessity of dealing with impacts — for which an impedance controller is more suitable [9] — we can rewrite the motion dynamics into its integral form, i.e. we are controlling the desired next point of the motion and not the current desired velocity, based on

$$\boldsymbol{x}_{\text{des}} = \boldsymbol{x}_t + \int_t^{t+\Delta t} \dot{\boldsymbol{x}} \, dt$$

$$= \boldsymbol{x}_t + (\boldsymbol{x}_{t+\Delta t} - \boldsymbol{x}_t) = \boldsymbol{x}_t + \Delta \boldsymbol{x}(\boldsymbol{x}_t) \qquad (2)$$

where $\boldsymbol{x}_{\text{des}}$ is the desired attractor position. Since $\dot{\boldsymbol{x}}$ is a function of the current position $\boldsymbol{x}$, the integral attractor distance $\Delta \boldsymbol{x}$ is going to be a function of the robot position $\boldsymbol{x}_t$. The dynamical system can be seen as an external (and slower) control loop where the attractor position is updated as a function of the robot position while the inner (and faster) impedance control loop

simulates the dynamics of a critically damped second order dynamical system towards the chosen attractor. As an analogy to humans, the slower loop can be seen as the intention update when generating a motion according to the current perceived arm position while the impedance control represents the compliance of the muscles and the joints in the interaction with the environment.

The desired Δx is fitted with a Gaussian Process (GP) using the data of a kinesthetic demonstration and user-provided corrections. A GP is a non-parametric regression method [15] where the mean and variance of the evaluation point are denoted as

$$\boldsymbol{\mu} = \boldsymbol{k}_*(\boldsymbol{\xi}, \boldsymbol{x})^\top \boldsymbol{K}(\boldsymbol{\xi}, \boldsymbol{\xi})^{-1} \boldsymbol{y}, \tag{3}$$

$$\Sigma = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}_*(\boldsymbol{\xi}, \boldsymbol{x})^\top \boldsymbol{K}(\boldsymbol{\xi}, \boldsymbol{\xi})^{-1} \boldsymbol{k}_*(\boldsymbol{\xi}, \boldsymbol{x}), \tag{4}$$

where $\boldsymbol{x}$ is the evaluation point, $\boldsymbol{\xi}$ is the input database and $\boldsymbol{y}$ is the output database, and $\boldsymbol{\mu}$ and Σ are the mean and variance of the regression in the evaluation point. The chosen kernel function of the process in this study is the sum of an Automatic Relevance Determination Squared Exponential kernel and a White Noise kernel according to

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_f^2 \, e^{\left(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{\Theta}(\boldsymbol{x}_i - \boldsymbol{x}_j) \right)} + \sigma_n^2 \delta_{ij} \tag{5}$$

where δ_{ij} is the Kroneker delta, Θ is a diagonal matrix of the horizontal lengthscales, σ_f is the vertical lengthscale, and σ_n is the observation noise. These hyper-parameters are the result of the likelihood maximization of sampling $\boldsymbol{y}$ from the fitted Gaussian Process. To avoid over/underfitting, we employed a constrained optimization between reasonable bounds for the search of the optimal hyperparameters.

Finally, something to consider when learning a dynamical system in a reactive formulation is that the next robot position is a function of the learned desired transition but also the external disturbances. This may lead the robot in a position where its policy is not confident anymore, i.e., high epistemic uncertainty. Depending on where this occurs, the robot may not be able to successfully pick up the object or bring it to its goal and execute its motion. When we, as humans, execute a motion we try to remain in regions where we are confident about what we have learned up to that point. To encode this behaviour also in the robot, the dynamical system was superposed with another dynamical system that brings the robot towards regions of low uncertainty. From a control point of view, this results in adding another attractor field that is proportional to the gradient of the variance manifold [4] according to

$$\Delta \boldsymbol{x}_{\text{stable}}(\boldsymbol{x}) = -\alpha \nabla \Sigma = \alpha \left(2 \boldsymbol{k}_*^\top \boldsymbol{K}^{-1} \frac{\partial \boldsymbol{k}_*}{\partial \boldsymbol{x}} \right), \tag{6}$$

where $\boldsymbol{x}$ is the evaluation point, and α is an automatically modulated constant which ensures that the product of $\Delta \boldsymbol{x}_s$ with the robot impedance $\boldsymbol{K_s}$ is never higher than a set threshold. This repulsive field can be seen as a *behavioural* stiffness: considering a variance manifold as a potential energy, similar to elastic energy, the robot is always acting towards the minimization of this quantity; similarly, the lower level control, "the muscles," is trying to converge to the attractor in order to minimize its *physical* tension. Thus, the Minimum Uncertainty Dynamical

---

Algorithm 1: Teaching Framework.

1   **a) Kinesthetic Demonstration(s)**
2   **while** *Trajectory Recording* **do**
3   Receive$\big(\boldsymbol{x}_t, \sin(\boldsymbol{\theta}^{\text{dem}}(\boldsymbol{x}_t)), \cos(\boldsymbol{\theta}^{\text{dem}}(\boldsymbol{x}_t)), w^{\text{dem}}(\boldsymbol{x}_t) \big)$
4     $\Delta \boldsymbol{x}^{\text{demo}}(\boldsymbol{x}_{t-1}) = \boldsymbol{x}_t - \boldsymbol{x}_{t-1}$
5   **end**
6   Train(GPs)
7   **b) Interactive Corrections**
    Data: $\Delta \boldsymbol{x}^{\text{dem}}, \gamma^{\text{dem}}, \sin \boldsymbol{\theta}^{\text{dem}}, \cos \boldsymbol{\theta}^{\text{dem}}, w^{\text{dem}}$
8   **while** *Control Active* **do**
9   Receive$(\boldsymbol{x})$
10   **if** *Received Human feedback* $\Delta \boldsymbol{x}^c, \gamma^c, w^c$ **then**
11   Correct$(\Delta \boldsymbol{x}^c \to \Delta \boldsymbol{x}^{\text{dem}}, \gamma^c \to \gamma^{\text{dem}},$
       $w^c \to w^{\text{dem}})$
12   **end**
13    $[\Delta \boldsymbol{x}, \Sigma] = \text{GP}_{\Delta \boldsymbol{x}}(\boldsymbol{x})$
14    $\gamma = \text{GP}_\gamma(\boldsymbol{x})$
15    $w_{\text{des}} = \text{GP}_w^{MU}(\boldsymbol{x})$
16    $[\sin(\boldsymbol{\theta}), \cos(\boldsymbol{\theta})] = \text{GP}_\theta^{MU}(\boldsymbol{x})$
17    $\boldsymbol{\theta}_{\text{des}} = \arctan2\left(\sin(\boldsymbol{\theta}), \cos(\boldsymbol{\theta}) \right)$
18    $\boldsymbol{x}_{\text{des}} = \boldsymbol{x} + \gamma \Delta \boldsymbol{x} - \alpha \nabla \Sigma$
19   Send$(\boldsymbol{x}_{\text{des}}, \boldsymbol{\theta}_{\text{des}}, w_{\text{des}})$
20   **end**

System (*MUDS*) can be summarized as

$$\boldsymbol{x}_{\text{des}} = \boldsymbol{x} + \Delta \boldsymbol{x}(\boldsymbol{x}) - \alpha \boldsymbol{\nabla} \Sigma(\boldsymbol{x}). \tag{7}$$

The superposition of this field is not conflicting because when close to the data, the prediction is non-zero while the uncertainty is zero with a small gradient. As the uncertainty increases, the prediction starts vanishing towards the mean of the independent Process (in our case zero) while the stabilization field increases its magnitude. This results in redirecting the robot towards regions of low uncertainty.

B. Minimum Uncertainty Inference

When learning a complex task like a fluent P&P, the dynamics of the end-effector position have to be augmented with the dynamics of the gripper orientation and width. Because in a trajectory the dynamics of the orientation and gripper are coupled with the dynamics of the end-effector, we decided to learn the controlled action as a function of the robot's position with another GP. However, if the predictions are done based on the current position, when outside of the region of certainty, the robot would output the mean of an independent Process (i.e., zero radians for the orientation along all three axes and maximum gripper width) which could lead to an undesirable generalization, e.g., tilting or dropping objects. In order to solve this problem, we propose a *minimum uncertainty inference*, obtained by projecting $\boldsymbol{x}$ in the highest correlated sample of the database according to

$$\boldsymbol{x} = \underset{\boldsymbol{\xi}_i}{\arg\max} \left(k\left(\boldsymbol{x}, \boldsymbol{\xi}_i \right) \right) \tag{8}$$

where k is the kernel function with the optimized hyper-parameters. This minimum uncertainty inference can be interpreted as a "mental" projection of the robot's current state on
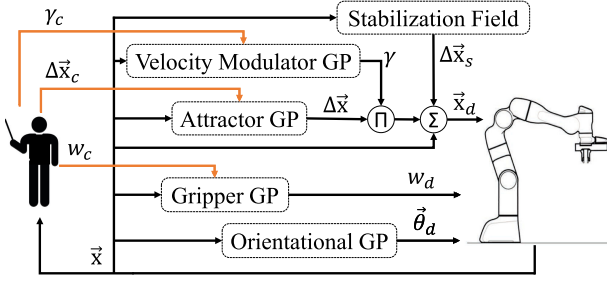
Fig. 2. A schematic representation of the human-in-the-loop giving corrections to the learned policy. The human has a visual feedback of the current robot motion and gives corrections with a joystick.

the highest correlated state (according to the kernel function) collected during the demonstration(s). The aim is to explicitly avoid extrapolating outside the original demonstrated data while still using the property of a smooth regressor of the GP. This behaviour also matches the philosophy of actively taking actions that would always minimize the uncertainty on the current robot state. When the evaluation of the GP is performed with this minimum uncertainty rule, we denote them with the superscript MU.

In order to fit the desired angles with a regressor, it is necessary to have a smooth and *continuous* representation of the angles. To this end we fit both $\sin(\boldsymbol{\theta})$ and $\cos(\boldsymbol{\theta})$ transformations of the Euler angles and convert them back after the MU inference during robot control (l. 17 Algorithm 1).

C. Interactive Policy Correction With Human-in-The-Loop

After learning from kinesthetic demonstrations the desired transition $\Delta\boldsymbol{x}$, Euler angles $\boldsymbol{\theta}$ and the gripper width w in the different points of the recorded trajectory, we still need to allow the user to correct the policy during the robot execution. Our goal is to obtain a fast continuous picking operation. With increasing velocities, kinesthetic interactions with a robot manipulator can become unsafe, and tuning both the attractor and gripper locally becomes very challenging. Furthermore, it also gives rise to ambiguity on the interpretation of the interaction forces as intended corrections or undesired disturbances [12]. For this reason, we opted for teleoperated corrections on the desired movement, local velocity and gripper width. Thus, due to the necessity of modifying the magnitude of the attractor distance proportionally in *all directions* (when higher/lower velocity are requested), a *scaling* factor is learned as a function of the position, resulting in a desired attractor

$$\boldsymbol{x}_{\text{des}} = \boldsymbol{x} + f(\boldsymbol{x}) = \boldsymbol{x} + \gamma(\boldsymbol{x})\Delta\boldsymbol{x}(\boldsymbol{x}) - \alpha\boldsymbol{\nabla}\Sigma(\boldsymbol{x}) \quad (9)$$

where $\gamma(\boldsymbol{x})$ is the attractor scaling factor. With this formulation, corrections can be allocated in the 3 different components of the vector or on the total magnitude of the vector itself. The complete control loop with human-in-the-loop corrections can be seen in Fig. 2. Overall, corrections are provided to the output values $\boldsymbol{y}_{demo}$ of the different GPs for the attractor distance $\Delta\boldsymbol{x}$, scaling factor γ and the width of the gripper prongs w, all of which are initialised with the kinesthetic demonstration. With the evaluation of the kernel, the corrective input can be smoothly spread to surrounding data points in accordance with

their correlation. The update rule was thus chosen as

$$\boldsymbol{y}^{\text{demo}} = \boldsymbol{y}^{\text{demo}} + \boldsymbol{k}_*^n(\boldsymbol{\xi}, \boldsymbol{x})\epsilon_\mu \quad (10)$$

where $\boldsymbol{k}_*^n$ is the correlation vector $\boldsymbol{k}_*$ normalised such that $\sigma_f = 1$, and ϵ_μ is the given correction provided at $\boldsymbol{x}$.

It has previously been shown that spreading the corrections on the database is more user-friendly, as well as time and data efficient [4] than a simpler data aggregation [16], since otherwise the GP model would essentially average between the different outputs for a given input, leading to a slow learning. Additionally, this constraint of spreading the corrections only on existing points of the database avoids to modify the shape of the variance manifold, keeping the motion always close to the kinesthetic demonstration, according to (6), while still shaping the motion dynamics, encoded in $\gamma(\boldsymbol{x})\Delta\boldsymbol{x}(\boldsymbol{x})$.

IV. VALIDATION EXPERIMENTS

Experiments were carried out to evaluate the effectiveness, usability and robustness of the method. In Section IV-A, the framework's base functionality of taking slow demonstrations and allowing the correction of the dynamics through corrective feedback is tested, along with an ablation study to verify the utility of uncertainty minimization. In Section IV-B, a baseline comparison to a method that also addresses the problem of interactive velocity modulation is performed. Section IV-C analyses how well a learned policy can accommodate changes in object properties such as size and weight. In Section IV-D, a straightforward generalization w.r.t. different object locations is briefly analysed. Lastly, in Section IV-E a user validation study was carried out with non-experts to establish the usability of the proposed method. A video of the experiments can be found attached to this paper [1].

We used the 7 DoF Franka-Emika Panda with an impedance controller and a ROS communication network for the online attractor update with a frequency of 100 Hz. Furthermore, in order to avoid overloading the GP with superfluous data, the recording of the trajectory is carried out at 10 Hz considering that whatever the human is showing at higher frequency is noise that would anyway be filtered out by the GP fitting and the impedance policy.

A wireless Logitech F710 Gamepad was used for teleoperated corrections. The Gamepad was chosen due to the number of required inputs, it being an established ergonomic input device in the gaming industry, as well as ensuring that users remain at a safe distance from the robot at all times considering the high-speed motion dynamics. Due to the limited number of continuous inputs, both the gripper and scaling factor corrections are provided through discrete increments. The attractor corrections are provided through the continuous inputs of the two thumbsticks, with the movement in the x-y-plane regulated by the left thumbstick and the height regulated by the right thumbstick. As an added safety feature, one of the triggers was utilised as a *safety button* which, when released, ends the execution of the algorithm, halting the robot. Lastly, users can comfortably start the execution from any point along the trajectory as well as bring the robot to the start of the trajectory. As a final remark, it is worth

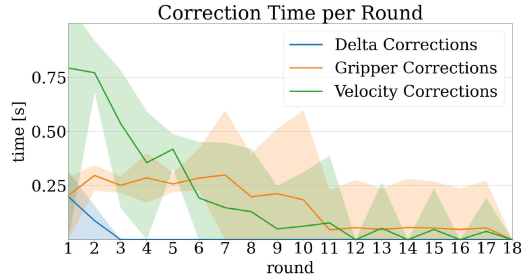---

[1]https://youtu.be/XoW6AkK793g

Fig. 3. Range of correction times per round for each aspect depicted by the shaded areas, with the average times depicted by the solid lines. Statistics made over 5 repetitions.

TABLE I
METHOD PERFORMANCE (5 DEMOS, 50 EXECUTIONS)

	Demo [s]	Fdbk [s]	Total Time [s]	Rounds	Success Rate [%]
Max	11.70	10.324	165.44	17.0	100
Mean	10.94	6.796	97.47	10.4	82
Min	10.10	4.560	66.61	6.0	50

underlining that the capability of correcting the orientation after the demonstration was not enabled due to the limitations of the teleoperation interface, not due to any limitations surrounding the algorithm itself and is thus left to future work.

A. Interactive Fluent Pick & Place With MUDS

For this experiment, a *single demonstration* was provided wherein the end-effector orientation, gripper width, and attractor distance are obtained and used for initialising the respective GP models. The goal of the task is to i) reduce the execution time by 4 times w.r.t. the demonstration time of the motion with kinesthetic teaching, and ii) have an execution time of 3 s or less. We repeated the experiment 5 times.

Within less than 3 min it was possible to fully train the robot to pick & place the object with the desired performance, four out of five times. Only a fraction of that time was needed for the demonstration (avg. 11 s) and explicit feedback from the human (avg. 6.8 s). This points towards primarily needing fine-tuning corrections from the human, which is further supported by the time spent giving corrections for each of the three correctable aspects (see Fig. 3).

It is worth noting that a correction round refers to an execution of a trajectory with optional user corrections, which can be stopped at any point of the execution and not just at the goal. The time spent correcting the attractor was minimal, as it was only required around the moment when the object is reached. This is because the human tends to stop at the object during the demonstration to avoid knocking it over and to deal with the closing of the gripper. To avoid that the motion stops, minor corrections to the attractor were provided for ensuring it follows the desired continuous picking motion. Afterwards only corrections for the gripper and scaling factor are provided. Whenever corrections to the scaling factor were provided, resulting in higher velocity, corrections to the gripper had to be provided as well to offset the communication delay of the gripper. Due to the unreliability of the gripper, despite corrections to the timing, the gripper still sometimes closed at the incorrect moment. Nevertheless, after corrections, an average success rate of 82% out of 10 autonomous executions of 5 different trained policies (41 successes over 50 executions in total) could still be achieved. For the complete performance details, please refer to Table I.

To verify the existence of gripper unreliability we measured the delay between sending the command for closing the gripper and the actual moment of closing. Measurements were gathered from 20 rollouts. While the average delay was 0.93 s, it ranged from 0.56 s to 1.54 s. Considering this stochasticity, the best strategy is to push the object at non-zero velocity for a long enough time so that it encompasses the possible moments at which the gripper might close.

One of the main concerns when increasing the velocity along a trajectory is diverging from said trajectory, particularly in curves. While the shape of the trajectory did change slightly, divergence from the trajectory could be avoided thanks to the uncertainty minimisation even when the attractor magnitude was noticeably increased compared to the original demonstration. This can be observed within the attractor vector fields in Fig. 4. This is an important feature of the proposed method, opening an alternative to many methods that are not dealing with covariate shift when they try to generalize. The goal was to show that even if the dynamics of the trajectory are modified, the obtained trajectory is not changing much, resembling the original demonstration.

To further evaluate the benefit of the uncertainty minimisation on the training as well as on the final execution, we performed an ablation study. The desired policy was trained once with the uncertainty minimisation active (w/ UM) and once without it (w/o UM). It was observed that the uncertainty minimisation made the training easier since it kept the robot close to the demonstrated trajectory. This translated to a shorter training time of 70 s w/ UM, whereas w/o UM 218 s were needed. We then performed two tests for observing the effect on the execution; one with a perturbation to the robot's initial position and one without such perturbation. The policies were rolled out 20 times each. The effect of the uncertainty minimisation was observed in the success rates of the P&P as well as the average distance error (ADE) of the executed trajectory w.r.t. the demonstrated trajectory. Without perturbations the policy w/ UM achieved the higher success rate of 95% and lower ADE of 0.023 m whereas the policy w/o UM only achieved a success rate of 45% and an ADE of 0.051 m. Similar results are observed when the perturbation is added, where the success rate of the policy w/ UM was 100% and the ADE was 0.034 m whereas the policy w/o UM only achieved 50% and had an increased ADE of 0.090 m.

For an evaluation of its benefit for reaching a goal while rejecting disturbances we would like to refer the reader to our prior work [4].

B. Baseline Comparison

We compare to a state-of-the-art approach in interactive dynamics modulation presented in [12]. The base method was replicated based on the details given in the paper with the only major change being that we do not learn the orientation with the DMPs. Since the focus of this baseline comparison was on the modulation of the translational dynamics, the gripper and orientation were controlled with the GPs, conditioned on the robot's current position for all the tests. Corrections were given with the joystick in both cases out of safety concerns when the robot is moving at high velocity.
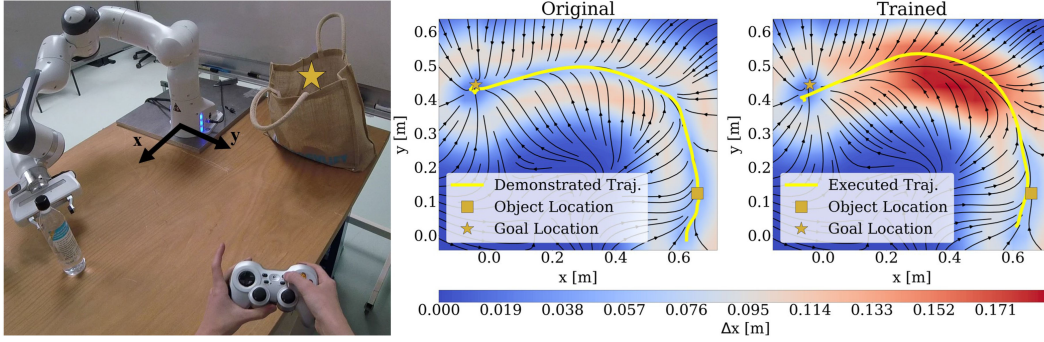
Fig. 4. Use case of robot assistance in grocery packing. In the attractor vector-field the arrows denotes the direction of the attractor and the color gradient denotes the magnitude of the attractor. The vector field based on original demonstration, with the demonstrated trajectory is compared with the one after training, with the executed trajectory.

We initialized the DMP, a version of our algorithm using the scaling factor (*V1*) and a version without the scaling factor (*V2*) with a single demonstration of picking the object given along the y-axis. Then, the object was displaced 7 cm to the side (x-direction) to compare the ability of both algorithms for reshaping and speeding up the motion.

What could be noticed with the DMP-based approach is that when a correction in x-direction was given the robot would virtually stop and only occasionally move forward. The cause of this was determined to be the dot-product of the position error with the predicted velocity $\widetilde{p}^\top \dot{p}_d$. This is used for changing the temporal scaling factor τ of the DMPs, such that when the error is in the direction of the velocity the evolution of the DMP is sped up whereas in the opposite case, the evolution of the DMP is slowed down. In our case, although the predicted velocity along x was very small, it was occasionally negative which could account for the undesired slowing down of the motion. Only in the moments when the velocity became positive along this axis did the robot move forward. As for speeding up, this was later possible along the y-axis, however, the generated acceleration was rather high even when a small correction was given. This is very likely due to the fast convergence of τ. The total training time for a successful picking policy was 197 s. The final achieved execution time was 8.43 s which was 1.17 times faster than the original.

With MUDS the correction in x-direction did not affect the motion in y-direction. Through the correction of the attractor distance along each of the axes, the shape of the trajectory along each of the axes could be easily altered. When using the scaling factor γ (*V1*), the speed along each of the axes of motion increases proportionally. Alternatively, if one chooses to not use γ and directly affect the velocity by changing the attractor distance along an axis (*V2*), one can ensure that the corrections do not affect the remaining axes. Depending on whether the velocity increase should be proportional in all directions (e.g., speeding up a diagonal motion in x-y-direction) or only along a single axis, the two approaches of altering the velocity help account for both possibilities. With *V1* 48 s were needed to train a successful picking policy whereas 46 s were needed for *V2*. The final execution times were 2.67 s for *V1* and 2.24 s for *V2* which translated to an increase of speed by 3.71 and 4.41 times respectively.

Fig. 5. L-R: rigid (250 g), rigid (900 g), flexible (100 g), small & deformable (250 g).

TABLE II
PERFORMANCE IN INTERACTIVE ADAPTATION

	Rigid (250 g) source	Rigid (900 g) new | adp	Flexible (100 g) new | adp	Small & def. (250 g) new | adp
Correction Time [s]	51	46 | 0	59 | 38	73 | 24
Rounds	5	5 | 0	7 | 4	8 | 4
Success [%]	88	96 | 98	98 | 100	98 | 96

C. Interactive Adaptation to New Object Properties

It can be that we want to pick up a different object after having learned a desired P&P behaviour. Even small changes in object properties can result in failure when using the same policy. Rather than demonstrating and retraining the strategy for every new object, or relying on hard-coded rules to adapt to these changes, corrections can be used to adapt the learned policy. A selection of four different objects was taken (seen in Fig. 5) to make a comparison of training from a new demonstration (new) and training a policy by adapting an existing policy (adp), as reported in Table II .

For the latter case, the initial policy was trained on a rigid water-bottle with a weight of 250 g (① in Fig. 5), our 'source' object. Once a satisfactory policy was achieved, the training object was swapped out for another object. The policy was then executed and corrected if necessary. Corrections were provided until the policy was successfully executed with the new object, after which an evaluation of the performance was performed. Subsequently, a different object was swapped in and the learned policy was *reset to the initial policy*.

TABLE III
PERFORMANCE OF NON-EXPERTS WHO SUCCESSFULLY FINISHED THE TASK

	T1: With Attractor Scaling				T2: Without Attractor Scaling			
	Demo Time [s]	Training Time [s]	Rounds	Exec. Time [s]	Demo Time [s]	Training Time [s]	Rounds	Exec. Time [s]
Max	34.10	600.00	36.00	4.97	14.90	285.00	23.00	4.00
Mean	13.04	323.30	19.40	3.42	8.63	121.22	9.11	2.81
Min	6.40	129.00	6.00	2.17	3.90	0.00	0.00	2.07

For each new object, the policy could be successfully corrected. For the same object but with a greater weight ② the initial policy carried out the policy successfully in the first execution, hence it was deemed that no corrections were necessary. For the flexible object ③ due to its lighter weight and ease at which it could be knocked over, minor corrections to both the velocity and gripper had to be given. Lastly, for the deformable object ④ it was necessary to reduce the speed for a successful picking. Otherwise, the object kept being knocked over upon impact due to its smaller support polygon. Nevertheless, for all three objects with their different properties it was possible to alter the policy within less time than what is needed for training from a new demonstration (see Table II).

It is important to note that the strategies for the separate objects are not stored as this would require a further form of knowledge representation or policy parametrization, which is outside the scope of this work. This evaluation does, however, show that adapting an existing policy is faster than learning from scratch, which can be beneficial for gathering knowledge more quickly.

D. Generalizing to New Positions

An important point of any algorithm is the generalisation capability. The above experiments were confined to policies trained within a global frame, making their generalizability limited. This can be overcome by using the position w.r.t. a local reference frame as input. To this end, a minor alteration had to be made where two policies were learned; one w.r.t. a local frame within the target object, and one w.r.t. a local frame at the goal. To determine which policy should be used when, a simple heuristic was applied which stated that the robot first moves w.r.t. the object and after picking it up moves w.r.t. the goal [17]. It is important to note that with the current approach there is a limit to how much the relative distance between the two frames can be changed w.r.t. the demonstrated one since when switching the frame of reference, the new policy must remain confident otherwise it will arrest the motion for safety.

To validate this extension we performed a short experiment where we trained the two policies (w.r.t. the object and w.r.t. the goal) with the frames fixed in one position. After the policy was successfully trained we placed the object in 20 different locations. The distance of these positions from the training location were taken from the ranges $x \in [-0.26; 0.02]$, $y \in [-0.30; 0.28]$, and $z \in [0; 0.08]$ all while considering locations physically feasible for the robot.

The total training time amounted to 99.4 s of which 78.9 s were needed for the corrections. Out of the 20 executions 13 were successful without any external influence, and 3 were successful once the human physically guided the robot into the region of certainty. For the latter 3, this was in fact a desired behaviour and a design choice to ensure that the robot does not generalise

and potentially behave in an unsafe manner in situations it has never seen. If a person wants to add information on how to behave in these areas, this can be done by adding new points as was addressed in [4], but this was not the focus of the proposed method. The remaining 4 executions resulted in clear failure. Out of these, 2 were in the case where the object was placed at a greater height than the demonstration. After successfully picking up the object, the robot proceeded to get stuck against the surface of the table since the policy w.r.t. the goal dictated that it should be following a trajectory that was below its current position.

E. Are Humans Great Teachers? A User Study

Since the aim of the proposed method is to enable people, who may not have a background in robotics and machine learning, to teach a robot, a preliminary user validation study was carried out. A total of ten participants aged 23 to 28 took part in this study (approved by TU Delft HREC). The same setup as in Fig. 4 was used, with the bag being replaced by a small square tower to provide a clearer goal. Half an hour of familiarisation with the setup was given before the actual trials began. There were two trials of ten minutes which were presented in a *randomised* order. In one trial (*T1*), users were required to perform a kinesthetic demonstration at a speed they were comfortable with. Afterwards, they had the possibility to correct the demonstration with the possibility to scale the attractor distance. To ensure that the main contribution to the velocity resulted from the scaling factor, the attractor $\Delta \mathbf{x}$ itself was bounded to 4 cm. In the other trial (*T2*), users were required to provide a *fast* kinesthetic demonstration. The attractor for this trial was left unbounded and any corrections for the velocity had to be performed by directly altering the attractor in the three Cartesian directions. *A trial was considered successful if the final trajectory execution time was 4 s or less.* The goal of this study was two-fold; i) verifying the feasibility of allowing non-experts to teach the robot non-zero-velocity P&P and ii) determining which correction approach users may prefer. In terms of performance, all participants were able to successfully pick & place the object in T1. Only one was unable to reach the 4 s goal. For T2, only one was unable to teach the task successfully.

Nevertheless, overall good teaching performance could be observed in both trials. For T1, users were able to teach the task within, on average, *5.4 min* with 19 correction rounds. The average time at which the robot could successfully pick & place the object that they could teach was 3.4 s with the *best time being 2.2 s*. For reference, the time needed to demonstrate the behaviour at a fast pace in T2 was at best 3.9 s, but generally participants needed more than 5 s to carry out the demonstrations (Table III for detailed results). *It thus becomes clear that overall non-experts are not able to or are not comfortable with providing*

fast demonstrations. Provided a faster demonstration, the time needed for corrections however did tend to be lower.

Participants were also asked which correction approach they preferred (T1 or T2). Within the group of participants, there was no clear preference towards one method or the other. There were, however, clear personal preferences. Half preferred to correct the complete translational dynamics with one input, claiming that it made it easier for trajectory shaping or more intuitive for altering the velocity since it compared more closely to the controls that are familiar from video games. Meanwhile, the rest found it easier to focus on correcting one aspect at a time, thus preferring to first correct the trajectory before increasing the velocity with the scaling factor γ, since there was less chance of accidentally affecting the other aspect with the corrections. This means that by opting for only one correction approach, the performance and comfort of some people would be hampered. For this reason it is important that the method gives people the possibility of using either of the two approaches.

V. CONCLUSION AND FUTURE WORK

We demonstrated that the motion dynamics of a user's demonstration can be successfully altered in a non-uniform manner using teleoperated user corrections. This allows users to overcome the limitations they had during the demonstration and teach the actual desired behaviour. It further allows users to compensate for delays within the system which are not directly known to them but are observable in the system's performance. Additionally, generalization to different object positions was obtained by switching between the two dynamical systems, learned in the respective reference frames. This proved how the variance minimization can be successfully used also to transition between two different frames. This opens many possibility of creating a sequence of multiple simpler dynamical systems for accomplishing complex robot tasks, i.e., assembling multiple components.

It was additionally shown that non-experts are able to successfully teach a non-zero-velocity motion for picking & placing objects. Irrespective of their prior experience or lack thereof with robots, they were able to successfully train this complex task, teaching and correcting the motion dynamics of many degrees of freedom. It could be seen that when only using the kinesthetic demonstration, people generally could not attain the desired execution time even with a fast demonstration. However, with the help of corrections to the motion dynamics, an execution speed outside of their demonstration capabilities became achievable. Since people have different preferences of teaching and correcting robots, we concluded that the final framework requires the velocity corrections to be provided both in a coupled (with only Δx) and decoupled manner (with γ and bounded Δx).

Certain aspects remain to be addressed for better generalization and performance of the proposed framework. A next step would be to study how to obtain haptic corrections of the policy while ensuring a fast but safe human-robot interaction. Further work is also needed in order to account for obstacles and reshape the vector field accordingly.

REFERENCES

[1] J. J. van Steen, N. van de Wouw, and A. Saccon, "Robot control for simultaneous impact tasks via QP based reference spreading," in *Proc. Amer. Control Conf.*, 2022, arXiv:2111.05211.

[2] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, 2019, Art. no. eaat8414.

[3] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 3, pp. 297–330, 2020.

[4] G. Franzese, A. Mészáros, L. Peternel, and J. Kober, "ILoSA: Interactive learning of stiffness and attractors," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2021, pp. 7778–7785.

[5] R. Vuga, B. Nemec, and A. Ude, "Speed adaptation for self-improvement of skills learned from user demonstrations," *Robotica*, vol. 34, no. 12, pp. 2806–2822, 2016.

[6] S. Kim, A. Shukla, and A. Billard, "Catching objects in flight," *IEEE Trans. Robot.*, vol. 30, no. 5, pp. 1049–1065, Oct. 2014.

[7] S. S. M. Salehian, M. Khoramshahi, and A. Billard, "A dynamical system approach for softly catching a flying object: Theory and experiment," *IEEE Trans. Robot.*, vol. 32, no. 2, pp. 462–471, Apr. 2016.

[8] M. Bogdanovic, M. Khadiv, and L. Righetti, "Learning variable impedance control for contact sensitive tasks," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6129–6136, Oct. 2020.

[9] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Requirements for safe robots: Measurements, analysis and new insights," *Int. J. Robot. Res.*, vol. 28, no. 11/12, pp. 1507–1527, 2009.

[10] D. Koert, J. Pajarinen, A. Schotschneider, S. Trick, C. Rothkopf, and J. Peters, "Learning intention aware online adaptation of movement primitives," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3719–3726, Oct. 2019.

[11] B. Nemec, N. Likar, A. Gams, and A. Ude, "Human robot cooperation with compliance adaptation along the motion trajectory," *Auton. Robots*, vol. 42, no. 5, pp. 1023–1035, 2018.

[12] T. Kastritsi, F. Dimeas, and Z. Doulgeri, "Progressive automation with DMP synchronization and variable stiffness control," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3789–3796, Oct. 2018.

[13] R. Perez-Dattari, C. Celemin, G. Franzese, J. Ruiz-del Solar, and J. Kober, "Interactive learning of temporal features for control: Shaping policies and state representations from human feedback," *IEEE Robot. Autom. Mag.*, vol. 27, no. 2, pp. 46–54, Jun. 2020.

[14] K. Kronander, M. Khansari, and A. Billard, "Incremental motion learning with locally modulated dynamical systems," *Robot. Auton. Syst.*, vol. 70, pp. 52–62, 2015.

[15] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: The MIT Press, 2006.

[16] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer, "HG-DAgger: Interactive imitation learning with human experts," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 8077–8083.

[17] G. Franzese, C. Celemin, and J. Kober, "Learning interactively to resolve ambiguity in reference frame selection," in *Proc. Conf. Robot Learn.*, 2020, pp. 1298–1311.