

Online Learning of Feed-Forward Models for Task-Space Variable Impedance Control

Michael J Mathew^{*1}, Saif Sidhik^{*1}, Mohan Sridharan¹, Morteza Azad¹, Akinobu Hayashi², Jeremy Wyatt¹

Abstract—

During the initial trials of a manipulation task, humans tend to keep their arms stiff in order to reduce the effects of any unforeseen disturbances. After a few repetitions, humans perform the task accurately with much lower stiffness. Research in human motor control indicates that this behavior is supported by learning and continuously revising internal models of the manipulation task. These internal models predict future states of the task, anticipate necessary control actions, and adapt impedance quickly to match task requirements. Drawing inspiration from these findings, we propose a framework for online learning of a time-independent forward model of a manipulation task from a small number of examples. The measured inaccuracies in the predictions of this model dynamically update the forward model and modify the impedance parameters of a feedback controller during task execution. Furthermore, our framework includes a hybrid force-motion controller that provides compliance in particular directions while adapting the impedance in other directions. These capabilities are evaluated on continuous contact tasks such as pulling non-linear springs, polishing a board, and stirring porridge.

I. MOTIVATION

Robot manipulation in dynamically changing environments is a challenging open problem. A robot arm stirring porridge, for instance, has to adapt its stiffness as the resistance offered by the porridge varies as a function of the viscosity of the fluid. A robot arm polishing a table has to adapt its stiffness to the frictional forces as it attempts to follow a desired motion pattern on the surface of the table; it must also maintain a suitable force along the surface normal while offering some compliance, e.g., when the table is tilted or moved up. Existing work on robot manipulation is unable to achieve the desired behavior in such tasks. These methods learn or compute stiffness values based on large labeled training datasets or comprehensive knowledge of the domain dynamics, impose unrealistic assumptions or hardware requirements, or use a state representation that makes it computationally expensive to estimate the stiffness parameters. On the other hand, research in human motor control indicates that when performing a new manipulation task, humans initially use higher arm stiffness to accurately follow the desired trajectory in the presence of unforeseen external disturbances. With sufficient experience, humans perform the task accurately with much lower stiffness. This behavior is achieved by building internal models of the task dynamics to predict the configurations of the object and the

hand, and the forces, during task execution [1], [2], [3], [4]. Studies in psycho-physics also indicate that humans learn to vary stiffness during manipulation [5], [6]. Our framework draws inspiration from such research to make a significant departure from existing literature on robot manipulation. It makes the following key contributions:

- A forward model of any given manipulation task is learned from a small number of examples and revised incrementally during task execution.
- Impedance parameters are defined as a state-dependent, time-independent property in task space. The measured error in the state predicted by the forward models is used to adapt these parameters during task execution.
- A hybrid force-motion controller contextually separates the directions in which the arm has to be stiff or compliant based on the task(s) at hand.

We evaluate these capabilities on three challenging continuous contact tasks: (i) pulling a combination of springs of different stiffness to a particular height and moving them in a desired pattern; (ii) polishing boards with different surface friction based on a desired motion pattern; and (iii) stirring porridge based on a desired motion pattern in the presence of changing viscosity. We show that our framework supports learning and generalization from limited training examples, and rapid adaptation of impedance parameters for different related tasks and environments. We discuss related work in Section II and describe our framework in Section III. Experimental results are discussed in Section IV, and the conclusions are in Section V.

II. RELATED WORK

Many existing methods for robot manipulation use machine learning algorithms to compute suitable values of the impedance parameters for the task at hand [7], [8], [9], [10], [11]. These methods either represent the desired stiffness profiles as a time series or as a task-specific policy, and need large labeled training datasets or comprehensive knowledge of the robot's dynamics and other mathematical models. It is often difficult to provide such training datasets or accurate domain knowledge. Research in classical control has developed many methods for adapting stiffness to achieve precise movement, e.g., hybrid force control [12], parallel force control [13], and impedance control [14]. These approaches require accurate knowledge of the system's dynamics and precise feedback schemes, which are difficult to provide in practical domains. Other methods have varied stiffness from the perspective of the object being manipulated [15], [16], [17]. These methods have mostly been designed for grasping

^{*} Authors contributed equally to this work

¹ University of Birmingham, Birmingham, B15 2TT, UK, [mjm522, sxs1412, m.sridharan, m.azad, j.l.wyatt]@bham.ac.uk

²Honda Research Institute Europe GmbH, Offenbach/Main, Germany, akinobu.hayashi@honda-ri.de

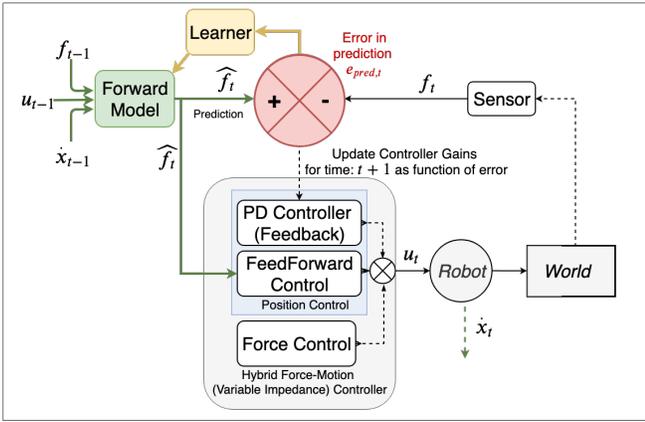


Fig. 1: Block diagram of proposed framework

and require accurate analytic models of the object; it is challenging to provide such models in dynamic domains. Many methods also make unrealistic assumptions such as quasi-static action, zero slippage, and point contacts, and require explicit representation of intrinsic parameters such as friction, mass, and coefficient of restitution [18].

A robot can use a *variable impedance controller* to change the impedance parameters to match the desired motion profile [14]. It is possible to achieve accurate motion and better rejection of perturbances by using a higher impedance (i.e. stiffness), but being very stiff expends more energy and makes it difficult to be *compliant* to external forces. Existing variable impedance control methods are time-dependent or include joint space parameters as a part of the state description [7], [19]. This dependence makes the task model out of sync with task execution in the presence of unforeseen perturbances, limiting the ability to adapt impedance.

Forward models have been used widely to predict the behaviour of the robot [20], [21] or the objects being manipulated [22]. The main challenge in building such models is the selection of state features to successfully learn a policy that predicts forces from the current state.

III. APPROACH

In our framework (Figure 1), the human designer either provides a desired motion pattern (i.e., profile) or moves the robot arm along the motion pattern for any given task. The robot learns a forward model of the task from the demonstration or a small number of trials trying to achieve the motion profile. The learned model predicts the force experienced in the next state and determines a feed-forward term in the control command. The prediction error measured during task execution revises the forward model and the gain (i.e., impedance) parameters of a feedback (PD) motion controller that provides the feedback term in the control command. A hybrid force-motion controller separately controls force along the direction(s) in which compliance is desired.

A. Basic Formulation

We formulate impedance control in the task space of the robot. The use of task (i.e., Cartesian) space controllers (with task-specific parameters) is independent of the type

of manipulator. They abstract multiple equivalent joint space trajectories into a task-space motion profile. A task-space controller is typically designed to make the robot behave as if a mass-spring-damper were attached between the end-effector tip and the motion way-point. Shaping a robot’s inertia to behave like a mass-spring-damper system without resulting in incorrect impedance behaviour is challenging; it imposes the impractical requirement of accurately measuring the external forces acting on the robot [23]. In practice, the desired impedance behaviour is limited to designing *stiffness* and *damping* parameters of the controller while keeping the inertia unchanged, resulting in a compliance control problem [23]. Since arbitrarily varying stiffness and damping parameters may result in instabilities [20], we use empirically estimated bounds for these parameters in our experiments.

In the basic formulation, the forward model maps the control command u_t , and measured force f_t to a predicted force f_{t+1} to be experienced in the resultant state. The force vector typically includes frictional forces along the plane of motion and the force along the surface normal. In Section III-D, we describe our hybrid force-motion controller, which has a PD controller with fixed gains to control the force along the direction in which compliance is desired. The basic motion controller thus has a feed-forward term and a feedback term, which reduces the number of training samples for variable impedance control [24]. The controller equation is:

$$u_t = \mathbf{K}_t^p \Delta x_t + \mathbf{K}_t^d \Delta \dot{x}_t + k_t \quad (1)$$

where u_t is the control command to the robot (i.e., task space force) at time t , \mathbf{K}_t^p and \mathbf{K}_t^d are the (positive definite) stiffness and damping matrices of the feedback controller; k_t is the feed-forward term provided by the forward model; and Δx and $\Delta \dot{x}$ are the errors in the end-effector position and velocity at each instant. During task execution, the forward model’s predictions determine k_t , and the prediction error controls \mathbf{K}_t^p and \mathbf{K}_t^d (Section III-C).

B. Learning and Using the Forward Model

The forward model is learned over a few trials as the robot attempts to follow the desired sequence of points (i.e., profile) in the task space for any given task. To create a time-independent forward model, a Gaussian Mixture Model (GMM) is fit over points of the form $p = [S_{t-1}, f_t]$, where S_t can be any combination of features that uniquely represent the robot’s state for the task, and f_t is the force felt at the end-effector at time t . S_t can contain information about end-effector position (x_t), velocity (\dot{x}_t), forces (f_t), etc. We explored two representations for S_t . The first is of the form $S_t = [\dot{x}_t, f_t]$. The second, motivated by studies of motor control [4], is of the form $S_t = [\dot{x}_t, f_t, u_t]$ where, u_t is the task space control command. This is similar to the “efferent copy” mechanism in animal motor control, where a copy of movement-producing signals are used by internal forward models to predict the effects of actions.

To incrementally update the GMM’s parameters (and create new components when needed) during task execution, we used the Incremental GMM (IGMM) method [25], [26],

[27]. IGMM internally uses a variant of the Expectation-Maximisation (EM) algorithm to fit the model and maximize the following likelihood function:

$$L(\theta) = p(\mathbf{X}|\theta) = \prod_{n=1}^T p(X_n|\theta) = \prod_{n=1}^T \left[\sum_{j=1}^M p(X_n|j)p(j) \right] \quad (2)$$

where $\theta = (\mu_j, \sigma_j, p_j)$ for $j = 1 \dots M$ are the parameters of the M components of the GMM. $\mathbf{X} = (X_1, \dots, X_T)$ represents the points to be fit, with $X_t = [S_{t-1}, f_t]$. Each point contains information about the *previous* end-effector state, along with the *current* force. So, when the learned model is used during task execution, the force for the next time instant, $(f_{t+1}|S_t)$, is predicted as a function of the robot’s current state using Gaussian Mixture Regression (GMR) [28].

C. Varying Feedback Gains

Many manipulation tasks can be accomplished using a very high stiffness (\mathbf{K}_{max}^p), but this expends energy. Also, if the robot has to perform a task in free-space, accurate trajectory following can be achieved with a much lower stiffness (\mathbf{K}_{free}^p). If the learned forward model is accurate, the feed-forward term should cancel out the external forces, reducing motion to that in free-space. Similar to human behaviour with a familiar manipulation task, the feedback gains can then be closer to \mathbf{K}_{free}^p . The feedback gains at each step (K_t) of the controller (Equation 1) are given by:

$$\mathbf{K}_t^p = \mathbf{K}_{free}^p + F(e_{pred,t-1})(\mathbf{K}_{max}^p - \mathbf{K}_{free}^p) \quad (3)$$

where $e_{pred,t}$ is the forward model’s prediction error at time t , and $F(x) : x \rightarrow [0, 1]$. With this formulation, the robot will be more compliant (stiff) when the model’s predictions are more accurate (inaccurate). The damping term is updated using the known constraint of the damping factor for an critically-damped system [29]:

$$\mathbf{K}_t^d = \sqrt{\frac{\mathbf{K}_t^p}{4}} \quad (4)$$

D. Hybrid Force-Motion Controller

Some manipulation tasks require compliance in some directions when an unexpected force is experienced. A robot following a motion profile to polish a planar surface must maintain contact force along the surface normal. If the robot experiences unexpected (e.g., frictional) forces along the plane, it has to become stiffer to follow the motion profile. However, if the surface were suddenly raised, the formulation described above would predict incorrect forces and increase stiffness, resulting in damage to the robot or the surface as the robot pushes down hard on the surface. We use a hybrid force-motion controller to intuitively separate the “compliant” and “stiff” directions. Such controllers define artificial constraints on the robot’s degrees of freedom. These constraints specify the desired values for velocities in the k directions of motion, and the forces in the remaining $6 - k$ directions for contact reaction. Through force control along

the directions in which compliance is desired, and motion control in the other directions, the robot can maintain the required normal force while following the trajectory on the surface. In our framework, these directions are currently defined manually for each task and considered to provide contextual information. The revised controller equation is:

$$u_t = \mathbf{K}_t^p \Delta x_t + \mathbf{K}_t^d \Delta \dot{x}_t + k_t + u_{fc} \quad (5)$$

where u_{fc} specifies part of the command signal produced by the direct force control.

IV. EXPERIMENTAL SETUP AND RESULTS

We experimentally evaluated the following hypotheses about the capabilities of our framework:

- H1:** Using feed-forward model along with stiffness adaption improves trajectory tracking performance.
- H2:** Adding efferent copy as the input to the forward model creates a better model of task dynamics, resulting in improved trajectory tracking.
- H3:** Updating the forward model online supports adaptation to new and changing environments.

where **H1** tests the effectiveness of adapting stiffness based on the accuracy of the model; **H2** compares the choices of the feature vector (Section III-A); and **H3** assesses if the framework can adapt to new environments. The *root mean square* (RMS) measure is used to quantify the error in achieving the desired motion profile, and suitable plots provide a qualitative indication of performance. Since our approach is different from popular approaches for such manipulation tasks (e.g., based on deep learning), we do not provide an experimental comparison but discuss the advantages of our approach in Section II and Section V.

We used three tasks to evaluate the hypotheses; we henceforth refer to them as “nonlinear spring pulling”, “board polishing” and “porridge stirring”. We used a 7-DoF Sawyer robot for our experiments; a video can be viewed online¹. The forward model is learned with the feature vector $p = [S_{t-1}, f_t]$, and $S_t = [\dot{x}_t, f_t]$ except for testing **H2** (Section III-B). The forward model learns the probability distribution of feature vectors; GMR conditions on S_t to predict f_{t+1} .

The first task involved pulling springs (attached to the end effector) to a particular height and then moving along a desired trajectory (Figure 2, left). Due to the non-linear force response to extension, the end effector experiences different changes in force in different directions. The baselines for comparison used constant low impedance and constant high impedance. The low impedance parameters were sufficient to move the end-effector along the desired trajectory in the absence of springs (\mathbf{K}_{free}^p), and the high impedance parameters (\mathbf{K}_{max}^p) were sufficient for pulling the spring in the absence of the forward model.

We conducted multiple trials with the forward model learned in the first trial and improved subsequently; results are shown in Figures 4– 5. We observe that the prediction accuracy of the forward model improves over the trials and

¹<https://youtu.be/hbzZu01xa18>

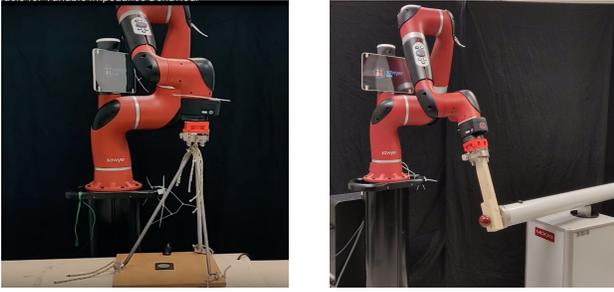


Fig. 2: **Left:** (Non-linear spring pulling) the objective is to pull a combination of springs to a desired height and then along a motion pattern; **Right:** (Porridge stirring) Sawyer is attached to a Moog HapticMaster which emulates an environment whose viscosity increases as it moves (in X-Z plane).

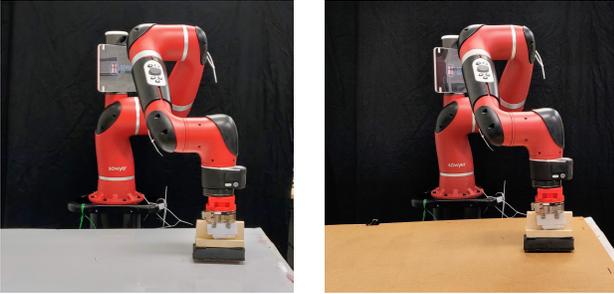


Fig. 3: (Board-polishing) **Left:** Surface 1 with low friction; **Right:** Surface 2 with higher friction.

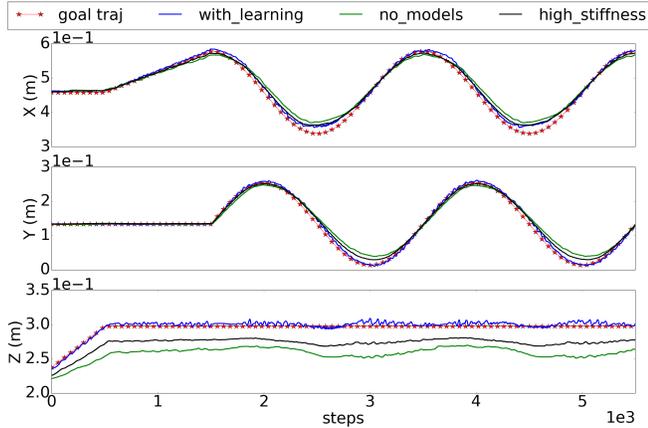


Fig. 4: (Non-linear spring pulling) Position tracking.

the position is tracked accurately; Table I quantifies the errors with no models (i.e., with constant low stiffness), high stiffness, or with the learned and revised forward models. Accuracy is better with the forward-model than in its absence. Performance improves further when the impedance parameters are updated online; just using the feed-forward term is not enough to perform tasks that involve unexpected forces acting on the system since the model is imperfect.

To further evaluate **H1**, we explored the board polishing task; the robot had to polish a surface of unknown friction coefficient by moving its end-effector (a whiteboard eraser) along a given trajectory while applying 10N downward force—see Figure 3. Here, a hybrid force-motion controller offers compliance along the surface normal (the z-axis).

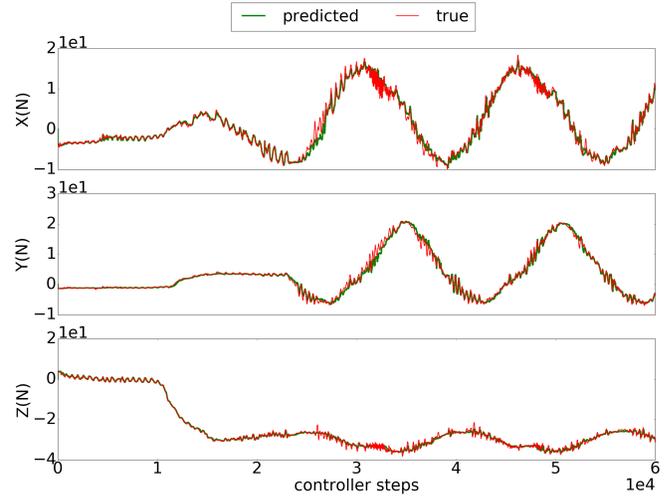


Fig. 5: (Non-linear spring pulling) Force prediction.

Condition	X (m)	Y (m)	Z (m)
no models	0.017 ± 0.009	0.015 ± 0.009	0.038 ± 0.010
high stiffness	0.012 ± 0.011	0.009 ± 0.007	0.023 ± 0.011
with learning	0.010 ± 0.010	0.006 ± 0.00	0.004 ± 0.00

TABLE I: (Non-linear spring pulling) Trajectory tracking errors along the three axes.

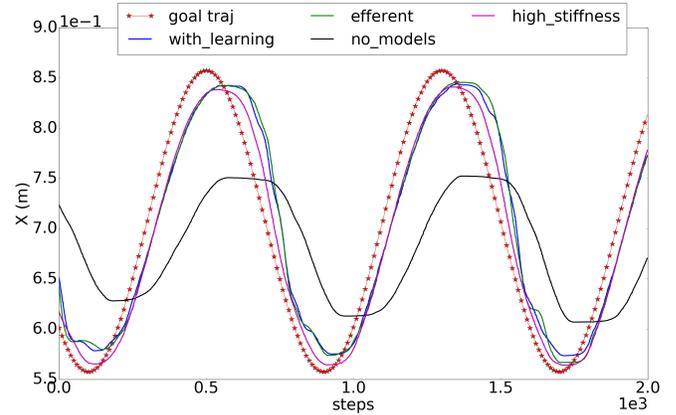


Fig. 6: (Board polishing) Surface 1 position tracking. Red: target; Pink: constant stiffness K_{free}^P ; Black: constant stiffness K_{max}^P ; Blue: adapting impedance without efferent copy; Green: adapting impedance with efferent copy.

While moving, the robot has to learn to predict the frictional forces that it experiences hampering its smooth motion. The initial model is learned by making the robot follow a trajectory (an epicycle) that is (intentionally) considerably different from the one it has to follow during task execution (sine wave).

Figure 6 shows that in the absence of the forward model, the robot is unable to follow the desired trajectory since it does not know the interaction forces. Using the feed-forward model improves tracking performance, with a further improvement when online impedance adaptation is used. The performance of the framework is comparable with that

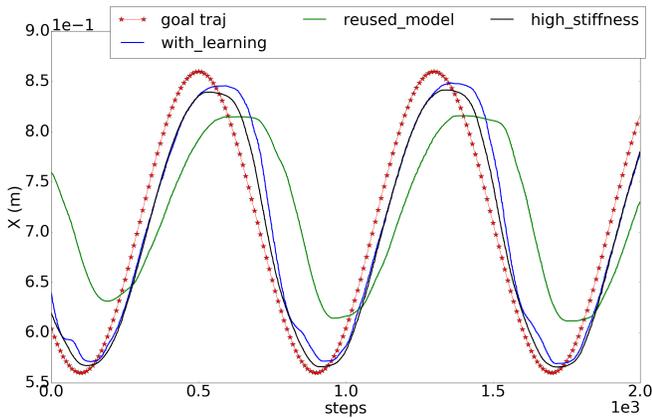
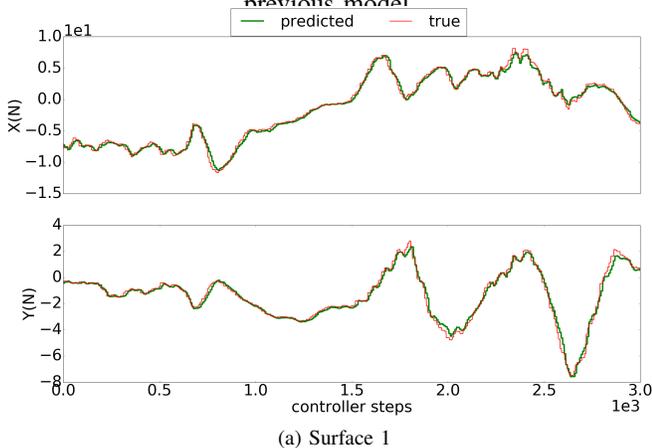
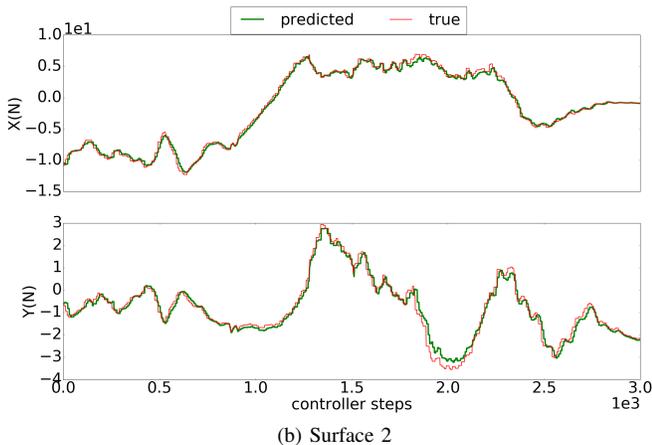


Fig. 7: **(Board polishing)** Surface 2 position tracking. Red: target; pink: constant stiffness K_{max}^p ; Green: surface 1 model without adaptation; Blue: online adaptation of previous model



(a) Surface 1

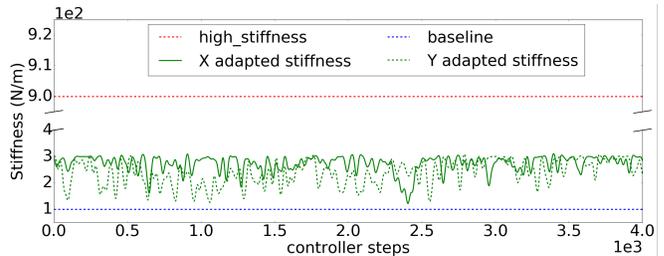


(b) Surface 2

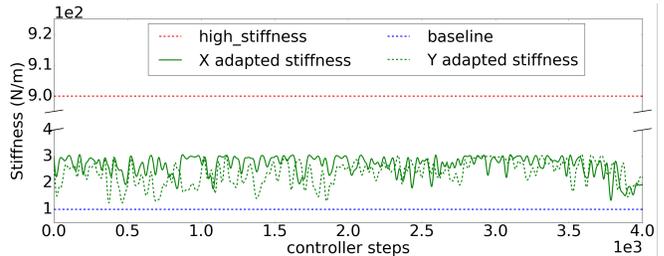
Fig. 8: **(Board polishing)** Force prediction.

of a high stiffness controller while requiring much smaller impedance parameters—Figures 8a and 9a. The average RMS errors in trajectory tracking are summarized in Table II with low stiffness (“no models”), high stiffness, or with the learned (and continuously updated) forward models. These results support the validity of **H1**.

Next, to evaluate **H2**, we conducted trials of board polish-



(a) Surface 1



(b) Surface 2

Fig. 9: **(Board polishing)** Stiffness adaptation

Condition	X (m)	Y (m)
no models	0.091 ± 0.042	0.054 ± 0.010
high stiffness	0.027 ± 0.011	0.007 ± 0.014
efferent copy	0.036 ± 0.024	0.008 ± 0.011
with learning	0.038 ± 0.023	0.008 ± 0.014

TABLE II: **(Board Polishing)** Surface 1: trajectory tracking errors along X and Y axes.

ing task with the efferent copy in the feature vector of the forward model. Results in Table II (“efferent copy”) indicate that there is no significant improvement in performance in comparison with the forward model that does not use the efferent copy. We believe this is because the forward model is able to obtain enough information for force prediction from the current end-effector velocity and forces, making the information encoded in the efferent copy redundant. This observed performance, and the fact that adding dimensions to the state-space makes the learning more computationally demanding, led us to *not* use the efferent copy in the subsequent experiments.

To evaluate **H3**, we focused on the ability to generalize and adapt. The results with the board polishing task indicate that the framework generalizes across different trajectories since the model was learned using a trajectory different from that used during task execution. This is a key advantage of learning the forward models in task-space instead of joint-

Condition	X (m)	Y (m)
high stiffness	0.027 ± 0.014	0.007 ± 0.012
reused model	0.084 ± 0.039	0.023 ± 0.022
with learning	0.035 ± 0.022	0.006 ± 0.001

TABLE III: **(Board Polishing)** Surface 2: trajectory tracking errors along X and Y axes.

Condition	X (m)	Z (m)
no models	0.031 ± 0.015	0.035 ± 0.018
high stiffness	0.012 ± 0.009	0.014 ± 0.012
forward model	0.037 ± 0.017	0.026 ± 0.010
forward model with stiffness adaption	0.014 ± 0.008	0.009 ± 0.007

TABLE IV: (**Porridge stirring**) Trajectory tracking errors along X and Z axes.

space. Next, adaptability to new forces was tested by performing the same (board polishing) task using a surface with significant different friction (Figure 3, right). Figures 7, 8b, and 9b show that when the learned model was used for this surface without online improvement, the robot was not able to follow the trajectory accurately. However, if the learned model is revised during task execution, it quickly achieves performance similar to that with the first surface. The RMS errors in trajectory tracking are summarised in Table III with high stiffness, learned model that is reused with revision (“reused model”), or with the learned and continuously updated forward model (“with learning”). This capability of the framework to generalise to different surfaces and trajectories is the key advantage of using a *task-space, time-independent* variable impedance control framework.

Next, the adaptability of the framework to environments that change during task performance is tested with the porridge stirring task. The viscosity of porridge changes as it is stirred. For the experiment on a real robot, the changes in viscosity are emulated on a MOOG HapticMaster [30]. Specifically, the viscosity (damping factor) of the environment is increased continuously (in X-Z plane) until it reaches a maximum predefined value. The end-effector of the Sawyer is attached to the end-effector of the HapticMaster (Figure 2, right). The Sawyer has to move its end-effector along a predefined motion trajectory while adapting to the viscous resistance from the environment.

Figure 10 indicates that using constant impedance parameters is not enough for tracking the desired trajectory (shown in red) accurately, even with the maximum allowed stiffness K_{max}^p (black). Similarly, using just the feed-forward term predicted by the forward model without impedance adaptation is insufficient for such dynamically changing environments (shown in green), owing to the dynamics of the environment that are not known to the robot. However, when online impedance adaptation is enabled, the robot is able to follow the trajectory more accurately (shown in pink). The trajectory tracking errors during task execution are summarised in Table IV with low stiffness, high stiffness, constant forward model, or with a continuously updated forward model. These results further establish the effectiveness of the framework. Next, Figure 11 indicates that the force predictions (by forward model) are much more accurate towards the end of the task when the environment finally becomes static as the change in damping saturates. The model has (by then) learned to accurately predict the effects of that damping factor on force. As a result, the impedance

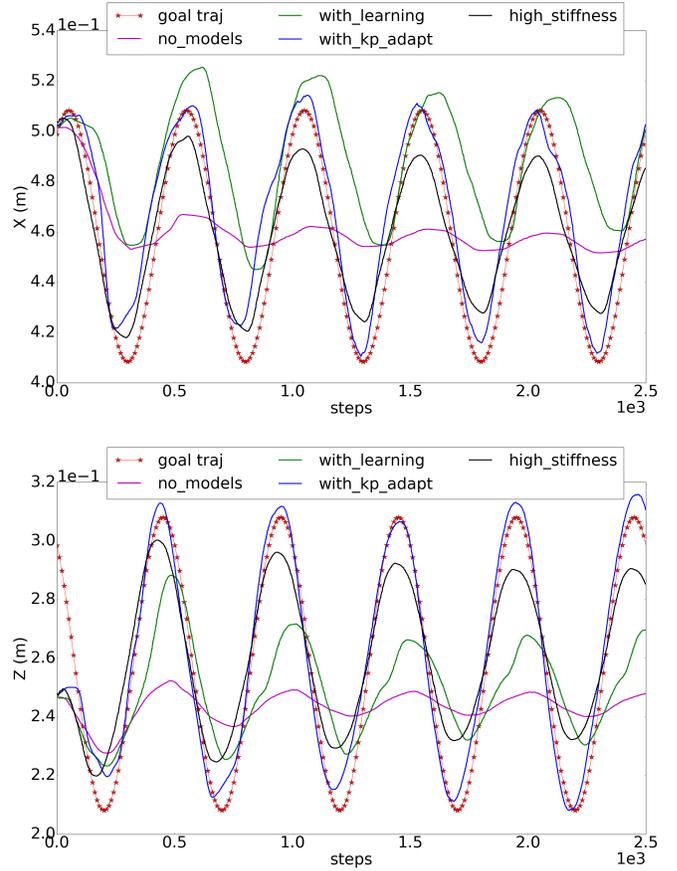


Fig. 10: (**Porridge stirring**) Position tracking (X-Z axes)

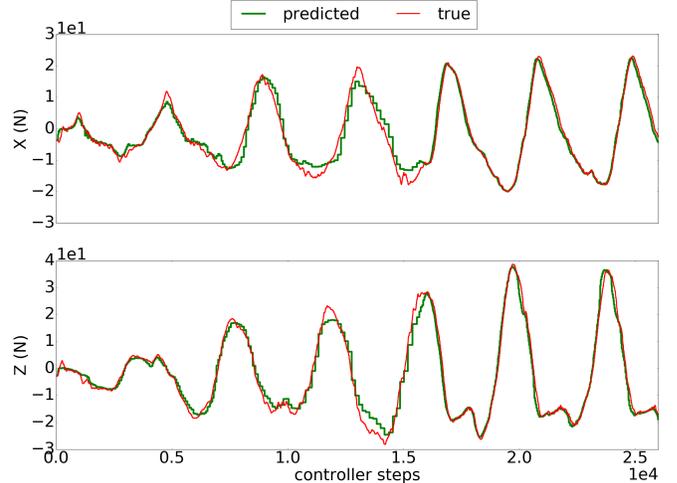


Fig. 11: (**Porridge stirring**) Force prediction

parameters are smaller, as shown in Figure 12. Note that the stiffness values are still lower than the high stiffness value (K_{max}^p).

We also ran trials that examined the ability of the hybrid force-motion controller to provide compliance along the surface normal with the board polishing task. Specifically, we ran trials in which the board was moved up or tilted during task execution. We (qualitatively) observed that the robot was able to provide compliance along the surface normal in each such trial. We do not include any quantitative

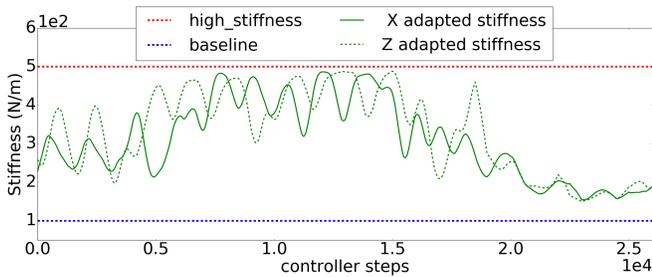


Fig. 12: (Porridge stirring) Stiffness adaptation

results corresponding to these experiments but this situation is included in the accompanying video demonstration.

V. DISCUSSION AND CONCLUSIONS

Variable impedance control is vital for reliable and safe robot manipulation. Learning impedance parameters directly is difficult and requires a large, labeled, training dataset. In this paper, we described a framework inspired by findings in human motor control. Our framework incrementally and continuously learns a time-independent, task-space forward model of any given manipulation task, using the model to predict interaction forces. The prediction error is used to revise the forward model and the impedance parameters for feedback control.

The framework was tested on three distinct tasks: non-linear spring pulling, board polishing, and porridge stirring. Other studies have used models created using knowledge of domain mechanics to make predictions about robot and object motions [22], [31]. However, they make unrealistic assumptions such as point contacts, friction cone approximations, and no slippage, unlike our method. Our initial studies with a linear spring provided insights on why the forward model (by itself) will not suffice, especially in the presence of previously unseen forces. We then introduced the task-space variable impedance feedback controller. One of the techniques proposed in [32] learns a state-dependent stiffness but the learning relies heavily on multiple demonstrations through an expensive special-purpose hardware device.

A key challenge (in our framework) was to choose suitable feature vectors to representing task state. We explored two distinct representations as described in Section III. Experimental results indicated that the use of the efferent copy as input to the forward model does not produce a marked improvement (over the other representation) but this hypothesis deserves further exploration in the context of other tasks. We did notice that the use of the end effector velocity instead of position as a feature in the state vector helped generalize across different trajectories and surfaces. Existing approaches such as [7] and [19] would fail to provide such generalization due to the explicit dependence on time for representing force and impedance parameters. Experimental results thus also demonstrate the advantages of learning forward models in the task space in comparison with existing work that learns forward models in the joint space [20], [8]. In addition, a crucial difference in comparison with other approaches for variable impedance control [16], [17], [15] is that we do not

use analytic models that require accurate knowledge of the system dynamics.

The porridge stirring task was the most challenging task we explored in this paper. It is unlike many other tasks considered by existing work in that it involved dynamic changes during task execution. Also, the factor that changes, i.e., the viscosity of the environment, opposes motion, the forces that the robot experiences are a function of the velocity rather than position. At the same time, this task provides a controlled emulation of a dynamically changing environment. The successful stiffness adaptation with good performance in this task thus provides strong evidence about the capability of our framework. Some existing approaches [20], [21] attempt to learn an action-effect correlation, usually from demonstrations provided by an expert or from experience obtained during trials [33], [10]. However, these methods requires explicit mathematical representations of the task, robot, and the objects involved.

Overall, the experimental evaluation provided promising results strongly indicating the ability of our framework to learn forward models and achieve task-space, variable impedance control in different continuous contact tasks. Future work will further examine the capabilities of this framework on other robots performing different tasks, and explore the extension of this framework to tasks that require the manipulator to make and break contacts.

REFERENCES

- [1] J. R. Flanagan, M. C. Bowman, and R. S. Johansson, "Control strategies in object manipulation tasks," *Current opinion in neurobiology*, vol. 16, no. 6, pp. 650–659, 2006.
- [2] M. Kawato, "Internal models for motor control and trajectory planning," *Current opinion in neurobiology*, no. 6, pp. 718–727, 1999.
- [3] R. S. Johansson and K. J. Cole, "Sensory-motor coordination during grasping and manipulative actions," *Current opinion in neurobiology*, vol. 2, no. 6, pp. 815–823, 1992.
- [4] R. Shadmehr and J. W. Krakauer, "A computational neuroanatomy for motor control," *Experimental brain research*, vol. 185, no. 3, pp. 359–381, 2008.
- [5] D. W. Franklin, E. Burdet, K. P. Tee, R. Osu, C.-M. Chew, T. E. Milner, and M. Kawato, "CNS learns stable, accurate, and efficient movements using a simple algorithm," *Journal of neuroscience*, vol. 28, no. 44, pp. 11 165–11 173, 2008.
- [6] E. Burdet, R. Osu, D. W. Franklin, T. E. Milner, and M. Kawato, "The central nervous system stabilizes unstable dynamics by learning optimal impedance," *Nature*, vol. 414, no. 6862, p. 446, 2001.
- [7] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, "Learning variable impedance control," *IJRR-2011*, vol. 30, no. 7, pp. 820–833, 2011.
- [8] M. Kalakrishnan, L. Righetti, P. Pastor, and S. Schaal, "Learning force control policies for compliant manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 4639–4644.
- [9] M. Howard, D. J. Braun, and S. Vijayakumar, "Transferring human impedance behavior to heterogeneous variable impedance actuators," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 847–862, 2013.
- [10] A. Kupcsik, M. P. Deisenroth, J. Peters, A. P. Loh, P. Vadakkepat, and G. Neumann, "Model-based contextual policy search for data-efficient generalization of robot skills," *Artificial Intelligence*, vol. 247, pp. 415–439, 2017.
- [11] H. van Hoof, T. Hermans, G. Neumann, and J. Peters, "Learning robot in-hand manipulation with tactile features," in *IEEE International Conference on Humanoid Robots*, 2015.
- [12] J. K. Salisbury, "Active stiffness control of a manipulator in cartesian coordinates," in *IEEE Conference on Decision and Control, including the Symposium on Adaptive Processes*, vol. 19, 1980, pp. 95–100.

- [13] S. Chiaverini and L. Sciavicco, "The parallel approach to force/position control of robotic manipulators," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 4, pp. 361–373, 1993.
- [14] N. Hogan, "Impedance control: An approach to manipulation," in *IEE-ACC-1984*. IEEE, 1984, pp. 304–313.
- [15] M. Li, H. Yin, K. Tahara, and A. Billard, "Learning object-level impedance control for robust grasping and dexterous manipulation," in *International Conference on Robotics and Automation*, 2014, pp. 6784–6791.
- [16] S. A. Schneider and R. H. Cannon, "Object impedance control for cooperative manipulation: Theory and experimental results," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 383–394, 1992.
- [17] T. Wimböck, C. Ott, A. Albu-Schäffer, and G. Hirzinger, "Comparison of object-level grasp controllers for dynamic dexterous manipulation," *International Journal of Robotics Research*, vol. 31, no. 1, pp. 3–23, 2012.
- [18] M. Kopicki, S. Zurek, R. Stolkin, T. Moerwald, and J. L. Wyatt, "Learning modular and transferable forward models of the motions of push manipulated objects," *Autonomous Robots*, vol. 41, no. 5, pp. 1061–1082, 2017.
- [19] M. Deniša, A. Gams, A. Ude, and T. Petrič, "Learning compliant movement primitives through demonstration and statistical generalization," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 5, pp. 2581–2594, 2015.
- [20] K. J. A. Kronander, "Control and learning of compliant manipulation skills," p. 204, 2015. [Online]. Available: <http://infoscience.epfl.ch/record/210787>
- [21] B. Huang, M. Li, R. L. De Souza, J. J. Bryson, and A. Billard, "A modular approach to learning manipulation strategies from human demonstration," *Autonomous Robots*, vol. 40, no. 5, pp. 903–927, 2016.
- [22] Y. Fan, W. Gao, W. Chen, and M. Tomizuka, "Real-time finger gaits planning for dexterous manipulation," *IFAC-2017*, vol. 50, no. 1, pp. 12 765–12 772, 2017.
- [23] T. Wimbock, C. Ott, and G. Hirzinger, "Impedance behaviors for two-handed manipulation: Design and experiments," in *International Conference on Robotics and Automation*, 2007, pp. 4182–4189.
- [24] N. T. Alberto, M. Mistry, and F. Stulp, "Computed torque control with variable gains through Gaussian process regression," in *IEEE-RAS 2014*, 2014.
- [25] M. Song and H. Wang, "Highly efficient incremental estimation of Gaussian mixture models for online data stream clustering," in *SPIE Conference Series*, K. L. Priddy, Ed., vol. 5803, March 2005, pp. 174–183.
- [26] W. Ahmad, "Incremental learning of gaussian mixture models."
- [27] P. Engel and M. Heinen, "Incremental learning of multivariate gaussian mixture models," in *BSAI*. Springer, 2010.
- [28] H. G. Sung, "Gaussian mixture regression and classification," Ph.D. dissertation, Rice University, 2004.
- [29] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.
- [30] O. S. Haptics, *Moog Haptic Master Manual*, 2008 (accessed May 9, 2019). [Online]. Available: https://www.h3dapi.org/modules/mediawiki/index.php/MOOG_FCS_HapticMaster
- [31] C. K. Liu, "Dextrous manipulation from a grasping pose," in *ACM Transactions on Graphics*, vol. 28, no. 3. ACM, 2009, p. 59.
- [32] K. Kronander and A. Billard, "Learning compliant manipulation through kinesthetic and tactile human-robot interaction," *IEEE Transactions on Haptics*, vol. 7, no. 3, pp. 367–380, 2014.
- [33] S. Levine and V. Koltun, "Guided policy search," in *International Conference on Machine Learning*, 2013, pp. 1–9.