Online Learning of Feed-Forward Models for Variable Impedance Control in Manipulation Tasks

Michael J Mathew*[1], Saif Sidhik*[1], Mohan Sridharan[1], Morteza Azad[1], Akinobu Hayashi[2], Jeremy Wyatt[1]

***Abstract*— While performing a new manipulation task, humans tend to be stiffer in the initial trials to improve task accuracy and counter any unforeseen disturbances. After a sufficient number of repetitions, humans are able to perform the task with lower stiffness without causing any significant reduction in task performance. Existing literature in human and animal motor control indicates that learned internal models of the manipulation task and the environment are used to predict the state in response to particular control actions, adapting stiffness on the fly to minimise energy usage. Inspired by these findings, we describe a framework that, in a significant departure from existing work, supports online learning of a time-independent forward model of any given manipulation task in the task space from a small number of samples. Errors in the predictions of the forward model are used to dynamically revise the model and adapt the impedance parameters of the feedback controller. Furthermore, a hybrid force-motion controller supports compliance in certain directions while adapting the stiffness in other directions. We evaluate our framework's capabilities in the context of continuous-contact manipulation tasks with varying external forces.**

I. Introduction

Consider the task of polishing a table. It involves following a particular trajectory on the surface of the table while applying a force normal to the surface. Frictional forces oppose the motion on the board surface. Research indicates that humans performing such tasks that involve continuous interaction with objects or the environment typically use higher (arm) stiffness when they perform the task for the first time. With sufficient experience, humans learn internal models of the task's dynamics, and use these models to predict the *task state*, i.e., the configuration of the object(s) and the hand in the context of the task. The task state dictates the variation in stiffness, and the choice of stiffness significantly influences performance [1]. Studies in psychophysics also indicate that learning to vary stiffness is a key step in manipulation tasks [2], [3]. Humans with a good learned model are able to adapt stiffness to the task, typically performing the task with much less stiffness than before.

A robot manipulator must also vary its stiffness as a function of the system state during task execution, which is usually achieved using a *variable impedance controller*. Varying the impedance varies the motion achieved as a result of the force experienced during interactions with the environment [4]. Increasing the impedance (i.e., stiffness) results in better rejection of perturbing forces and more accurate motion, but it results in more energy being expended and makes it difficult to be *compliant* to external forces. Also, a key shortcoming of existing variable impedance control methods is their explicit dependence on time in the joint-space description of state. This dependence causes the task model and task execution to go out of sync in the presence of perturbing forces, and limits the ability to adapt impedance. We seek to address these challenges by developing a framework that draws inspiration from findings in human (and animal) motor control. Figure 1 shows a block diagram of our framework, which makes a significant departure from existing work in the form of the following key contributions:

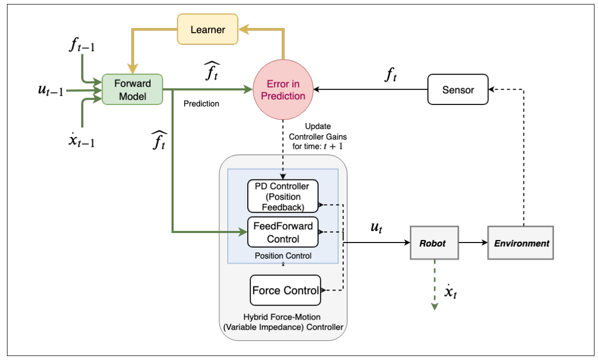


Fig. 1: Block diagram of proposed framework

- Learning of forward model of any given manipulation task from a small number of demonstrations, with the learned model being revised incrementally in real-time during task execution;
- Definition of stiffness parameters as a state-dependent, time-independent property in task space, with the measured error in the state predicted by the forward models being used for the online adaptation of these parameters to the task at hand;
- Introduction of a hybrid force-motion controller that separates the directions in which the arm has to be stiff or compliant based on the tasks at hand.

We demonstrate experimentally that the proposed formulation enables rapid learning and generalisation of forward models from very few training examples, and that the models for a particular manipulation task can be reused for other similar manipulation tasks.

II. Related Works

Research in robot control has devoted considerable attention to high precision movements in free space. These methods focus on following trajectories accurately but do

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

not support safe control under contact conditions. The main challenge in that accurate position tracking requires suitable stiffness whereas compliance to external forces comes with a loss in the accuracy of position tracking.

There are several approaches in classical control for adapting the stiffness, e.g., hybrid force control [5], parallel force control [6], and impedance control [4]. The main limitation of these approaches is the requirement of the knowledge of accurate dynamics model of the system and precision in the feedback schemes, which are designed from a manipulator perspective. Alternatively, there are approaches to design varying stiffness control from the object perspective as shown in [7], [8], [9]. However, most of these approaches are specifically designed for grasping and are based on accurate analytic models of the object. Usually when a manipulator is used to manipulate an object, it is very difficult to precisely perform a system identification of the object using the manipulator alone. Learning-based approaches [10] require a lot of external hardware to collect demonstrations from human experts to encode and replicate the stiffness variation in tasks.

Learning-based approaches have been used to solve different manipulation tasks [11], [12], [13], [14], [15], [16], [17]. However, these works either represent the stiffness profiles as a time series or as a task-specific blind policy. Also, learning the variable impedance parameters of the controller for a task requires the knowledge of the dynamics of the robot, which requires explicit mathematical model or large training data.

Therefore, instead of learning to vary impedance parameters directly, the proposed framework varies impedance parameters by learning a forward model of the task. It is observed that the human motor system uses predictive models of the effects that motor actions have on sensory data [18], [19]. It has been shown that humans learn a predictive model to anticipate the forces that can be expected while performing a task [1], which is critical for the success of the task. These predictions are used for different purposes such as feed-forward control, motor system coordination, action planning and monitoring.

Forward/predictive models have been widely used in literature to perform manipulation tasks. They are generally used to predict the behaviour of the robot [20], [21] and/or the objects [22] being effected while performing the task. The main challenge in the development of such a forward model is the selection of relevant state features of the task that can enable a learner to successfully learn a policy that can predict the forces from the current state.

One approach for using predictive models is to build a model informed by knowledge of mechanics to make predictions about robot and object motions [22], [23]. They assume that Newtonian mechanics governs the objects dynamics and estimate parameters like the mass, inertia, Coriolis component and gravity effects using regression techniques. However, most methods make unrealistic assumptions such as quasi-static action, zero slippage, point contacts etc. Furthermore, to make precise predictions, these approaches require explicit representation of intrinsic parameters, such as friction, mass, mass distribution, and coefficients of restitution, which are not trivial to estimate [24].

Another approach is to use learning to build a forward model. These learn a more action-effect correlation, usually from demonstrations provided by an expert [20], [21] or from experience during trials [25], [26]. These methods do not require explicit mathematical representations of the task, robot, or the objects involved. However, the choices regarding the model representation, learning algorithm, state representation, etc. are challenges that are to be tackled while applying learning-based methods.

III. APPROACH

It is observed that the human motor system uses predictive models of the effects that motor actions have on sensory data [18], [19]. These predictions are used for different purposes such as feed-forward control, motor system coordination, action planning and monitoring. Inspired by human control literature [1], [27], equipping a robot manipulator with the ability to learn a predictive forward model of the task environment will enable the robot to predict the forces it may experience in the next state and apply appropriate control signals to counter these forces. The predictions provided by such a forward model can then be used as a feed-forward term in the control command. Incorporating a feed-forward term along with the feedback term in the controller has been shown to reduce the number of training samples required for learning a variable impedance policy [28]. The corresponding controller equation can be written as:

$$u_t = \mathbf{K^p_t}\Delta x_t + \mathbf{K^d_t}\Delta \dot{x}_t + k_t \tag{1}$$

where u_t is the control command to the robot at time t, $\mathbf{K^p_t}$ and $\mathbf{K^d_t}$ represent the (positive definite) stiffness and damping matrices of the feedback controller respectively; k_t is the feed-forward term provided by the forward model; Δx and $\Delta \dot{x}$ are the errors in the end-effector position and velocity.

In the proposed framework, the impedance control is formulated and deployed in the task-space of the robot. Task-space controllers (cartesian-space controllers) are intuitive and have task-specific parameters. Hence, usage of the same framework becomes independent of the type of robot manipulator. Task-space controllers are typically designed as a *mass-spring-damper* system. The goal of the controller is to make the robot behave like a spring attached between the end-effector tip and the motion *way-point*. Shaping *inertia* of the robot to behave like a mass-spring-damper system is generally tricky [29] and requires accurate measurement of the external forces acting on the robot. Since measuring forces on every acting point is impractical, it is difficult to modulate the inertia tensor without leading to incorrect impedance behaviour. For this reason, in practice the desired impedance behaviour is limited to designing *stiffness* and *damping* parameters of the controller while keeping the natural inertia unchanged, resulting in the compliance control problem [29].However, arbitrarily varying stiffness and damping parameters may result in unstable behaviour of the robot [30]. In all our experiments, the bounds within which the stiffness and damping parameters are varied was found empirically.

During the training phase, the forward model is initialised by executing the provided demonstration in constant stiffness. This forward model is further improved online in subsequent trials by the robot. At each state, the system is trained to predict the force at the next time instant given the current state, in a supervised manner (Section III-A).

During task execution, the predictions from the forward model are used as a feed-forward term (k_t) in the controller, while the error in the prediction controls the feedback gains ($\mathbf{K_t^p}$ and $\mathbf{K_t^d}$). The feedback control gains are revised proportional to the prediction error (Section III-B. As the model is updated online, the error in predictions is expected to reduce over time, and the feedback gains can be adjusted appropriately.

Figure 1 is a graphical representation of the proposed framework. The feed-forward component of the controller uses the prediction from the forward model. This together with the feedback component, provides the motion control command to the robot. The difference between the measured and predicted state, i.e., the error signal, is used to update the model and to decide the subsequent feedback gains for the motion controller .The force control is used to control the forces in the directions orthogonal to motion, if required. This way of separating force and motion in the task-space can be used to separate the directions to be stiff and compliant.

A. Learning and Using the Forward Model

If the robot has a forward model telling it what forces it will experience in the next instance, an appropriate feed-forward value can be used to compensate for the forces. This will allow the robot to execute a task without being hindered by the changes in external force.

To create a state-dependent (time-independent) forward model, the proposed framework uses a Gaussian Mixture Model (GMM) to represent the forward model. The GMM is fit over points of the form $p = [S_{t-1}, f_t]$, where S_t can be any combination of features that can uniquely represent the state of the robot for the task at an instant, and f_t is the force felt at the end-effector at time t. The state vector S_t can contain information in terms of end-effector position (x_t), velocity ($\dot{x}_t$), forces (f_t), etc.

The model is fit using the Expectation-Maximisation (EM) algorithm, where the following likelihood function is maximised:

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

where $\theta = (\mu_j, \sigma_j, p_j)$ for $j = 1...M$ are the parameters of the M components of the mixture model. $\mathbf{X} = (X_1, X_2, ..., X_T)$ represents the points to be fit by the GMM, with $X_t = [S_{t-1}, f_t]$.

Note that each point contains the information about the *previous* end-effector state, along with the *current* force. This is so that when the model is used for prediction during task execution, the force for the next time instant can be predicted using the current state of the robot. The force at next instant $(f_{t+1}|S_t)$ is computed from the learned model using Gaussian Mixture Regression (GMR) [31].

1) Choice of Feature Vector: In this work, two types feature vectors were tested to represent the state S_t. The first is of the form $S_t = [\dot{x}_t, f_t]$. The second type is motivated by computational motor control studies [27] and is of the form $S_t = [\dot{x}_t, f_t, u_t]$ where, u_t is the computed control command in the robot task space (end-effector force to be applied). This is similar to the 'efferent copy' mechanism in animal motor control, where a copy of movement-producing signals are used by the internal forward models to predict the effects of the actions, so as to compensate for them.

2) Online Incremental Learning of Forward Model: A variant of GMM called Incremental GMM (IGMM) [32], [33], [34] is available which can incrementally learn a mixture model for the joint density of the oncoming data. IGMM incrementally updates the density estimate taking only the newly arrived data and the previously estimated density. It is also able to create a new mixture component if the new instantaneous data point is below the acceptable likelihood for the current model.

The proposed framework makes use of IGMM to learn and improve the forward model *online* during task execution. Due to the fast incremental learning, the approach is able to quickly adapt to different tasks when initialised with a model that was learned for a similar task. This is shown for a robot polishing different surfaces.

B. Varying Feedback Gains

Considering a robot pulling a spring in different directions, it is easy to see that the robot can perform the task successfully with very high stiffness ($\mathbf{K}^p_{max}$), but this would require the robot to become less compliant throughout the task, hence spending more energy. If the robot had to perform the same pulling action, but this time in free-space, it would obviously require a much lower stiffness value $\mathbf{K}^p_{free}$) for accurate trajectory tracking.

If the learned forward model is accurate, then the feed-forward term should cancel out the external forces in the task, essentially making it a motion in free-space. Hence, the more accurate the learned model becomes, the lower the feedback gains can be (closer to $\mathbf{K}^p_{free}$). This is similar to human behaviour for a new manipulation task. The feedback gains at each instant (K_t) in the controller (Equation 1) can therefore be varied according to

$$\mathbf{K}^p_t = \mathbf{K}^p_{free} + F(e_{pred,t-1}) \times (\mathbf{K}^p_{max} - \mathbf{K}^p_{free}) \qquad (2)$$

where $e_{pred,t}$ is the error in prediction of the forward model at time t, and $F(x)$ is a function that maps $x \rightarrow [0, 1]$, such as the logistic function. This way of modulating the feedback term ensures that if the model is predicting accurately, the robot will be more compliant, while wrong predictions ensure that the robot becomes stiff for following the trajectory more accurately. Since the forward model is updated online, the error in predictions is expected to reduce over time, and the feedback gains should become lower.

The damping term is updated using the constraint of the damping factor for an over-damped system, given by the following equation:

$$\mathbf{K}_t^d = \sqrt{\frac{\mathbf{K}_t^p}{4}} \tag{3}$$

C. Separating the Directions to be Stiff and Compliant

Not all manipulation tasks can be achieved using the above formulation. Some tasks require the robot to become compliant when it experiences unexpected force. Consider a task where the robot has to polish a planar surface. Here, the robot has to maintain the contact force in the direction normal to the surface, while following the trajectory defined in the plane of the surface. Suppose the surface is raised suddenly during the task. If the above formulation is used, the model would predict the wrong forces and the robot would increase its stiffness. This would result in the robot trying to push the surface down, damaging itself and/or the surface, which is not the desired behaviour.

On the other hand, if the robot experiences unexpected forces along the directions of the plane of the robot (eg. frictional forces), the robot has to become stiffer to follow the trajectory more accurately (and incrementally learn the forward model). Therefore, unexpected forces along different directions demand different behaviour from the robot – some directions require the robot to become stiff in the presence of unexpected forces, while others require compliance.

The most intuitive and straightforward way of separating the ‘compliant’ and ‘stiff’ directions is using a hybrid force-motion control framework. Hybrid force-motion controller works by defining *artificial* constraints to the robot’s degree of freedom. The task is expressed in terms of these constraints which specify desired values (to be imposed by the control law) for the velocities in the k directions feasible for motion, and for the forces in the remaining $(6-k)$ directions feasible for contact reaction.

By using direct force control along the directions to be compliant, and motion control for the directions to be stiff, the robot can maintain the required normal force along the trajectory, while following the trajectory on the surface. These directions can be defined manually for each task. The final control equation thus becomes:

$$u_t = \mathbf{K_t^p} \Delta x_t + \mathbf{K_t^d} \Delta \dot{x}_t + k_t + u_{fc} \tag{4}$$

where u_{fc} specifies the command part produced by the direct force controller so as to maintain the desired force along the desired direction(s).

IV. Experiments and Results

With the framework formulated as described, experiments were devised to test the following three hypotheses:

Hypothesis 1: Using feed-forward model along with stiffness adaption improves trajectory tracking performance.

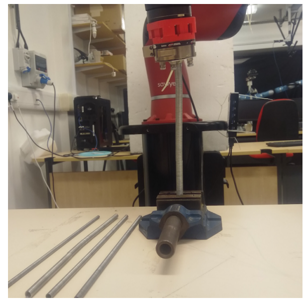


Fig. 2: Linear Spring-Pulling Task Setup

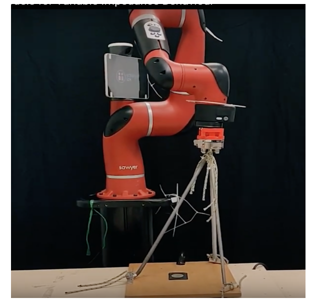


Fig. 3: Non-linear Spring-Pulling Task Setup

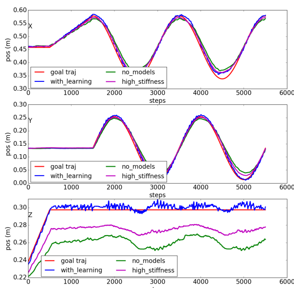


Fig. 4: Non-linear Spring: Position Tracking

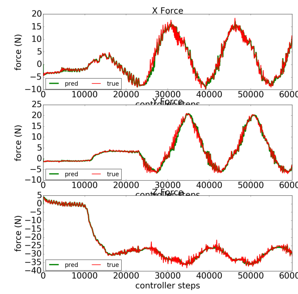


Fig. 5: Non-linear Spring: Force Prediction

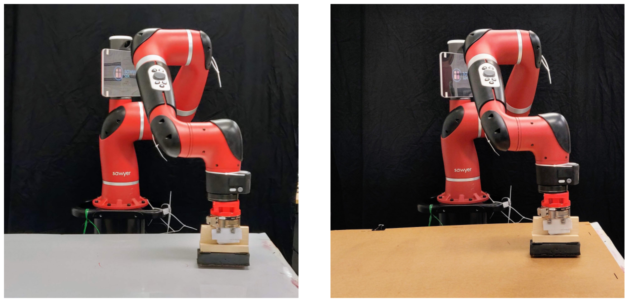
Hypothesis 2: Adding commanded force (efferent copy) to the input of the forward model has a significant impact on the performance.

Hypothesis 3: Defining the forward model as a function of the state allows previously learned model to be adapted online to similar tasks.

The first hypothesis tests the effectiveness of using the forward model and subsequently adapting stiffness based on the accuracy of the model. The second hypothesis compares the choices of the feature vector (Section III-A.1). The third hypothesis is used to test the generalisability of the framework.

To test the hypotheses two set of experiments were designed using a 7-DoF Sawyer robot (Video: `https://youtu.be/vjJwexVziS0`). The forward model is learned online as described in Section III-A, with the feature vector $p = [S_{t-1}, f_t]]$. For testing hypotheses 1 and 3, the state vector was chosen to be $S_t = [\dot{x}_t, f_t]$. For testing hypothesis 2, the forward model uses the computed force vector in the task space (u_t) in addition with the existing feature vector; $S_t = [\dot{x}_t, f_t, u_t]$. The forward model IGMM learns the parameters which models the probability distribution of the feature vectors. Later this learnt model can be used to condition on S_t to predict f_{t+1} using GMR.

The first task involved pulling springs along a given trajectory. The existence of the spring is unknown to the robot controller. This experiment was tested first with a linear spring (as a proof-of-concept; Figure 2), and then with a non-linear spring setup (Figure 3), which is significantly more challenging due to its non-linear force response to the extension. When the end-effector moves, the spring (fixed at one end) starts pulling the end-effector. The end effector



Left: Surface 1; Right: Surface 2

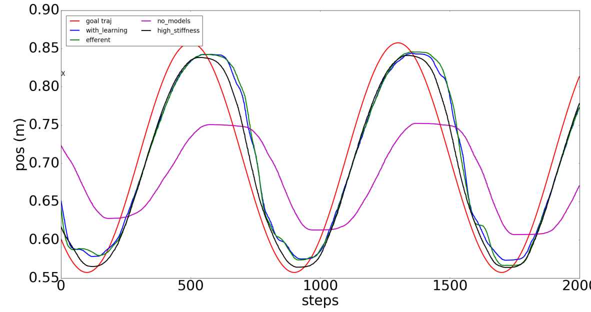
Fig. 6: Board-Polishing Task Setup

experiences force changes depending on the spring extension. Compared to the linear spring setup, the robot experiences different forces in different directions for the non-linear task setup. The baselines for the experiment was created by performing the tasks first using constant low impedance parameters, and then with constant high parameters. The lower baseline parameters were chosen such that they were enough to move the end-effector along the desired trajectory in the absence of springs ($\mathbf{K}^p_{free}$), and the constant high impedance parameters ($\mathbf{K}^p_{max}$) were successful at pulling the spring in the absence of the forward model.

After baselines creation, the task was repeated for following the trajectory (Equation 1), using the feed-forward term from the forward model and impedance adaptation as described in Section III. The task involved pulling the spring along the desired trajectory multiple times, without any previous training. Here, the model is learned online for the first time in the first lap of the trajectory and then improved online in the subsequent rounds. The results for the non-linear spring task are shown in Figures 4 and 5 (results of linear spring task are not included for brevity and due to result redundancy). Figure 5 shows the accuracy of the forward model as it executes the task. The prediction improves significantly as the model gets additional experience compared to the first task trial.

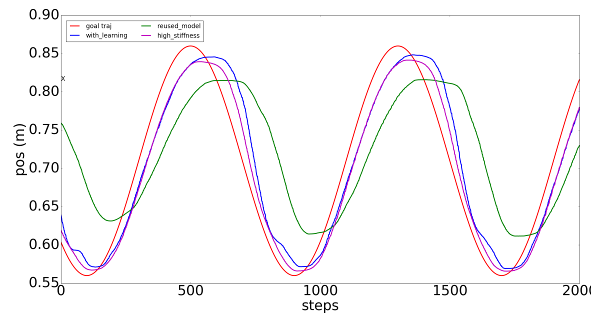
Figure 4 shows that the accuracy of position tracking is better with using the forward-model than without using it. The performance further improves significantly by varying the impedance parameters following the model accuracy. It can be concluded that using just the feed-forward term is not enough to perform the task accurately unless the learned model is perfect. Hence, the proposed framework improves the model online and adapts the impedance parameters depending on the model accuracy.

To further investigate the first hypothesis, the robot has to wipe a surface of unknown friction coefficient. The robot has to move its end-effector (a whiteboard eraser) on the surface along a given trajectory while applying 10N downward force. Here, the directions for the robot to be 'stiff' and 'compliant' against unexpected forces are separated using a hybrid force-motion controller, as explained in Section III-C. While moving, the robot experiences frictional forces hampering the smooth motion of the robot, which it has to learn to predict. The model initially learned by making the robot wipe the surface following a trajectory that is considerably different from the one it has to follow during task execution.



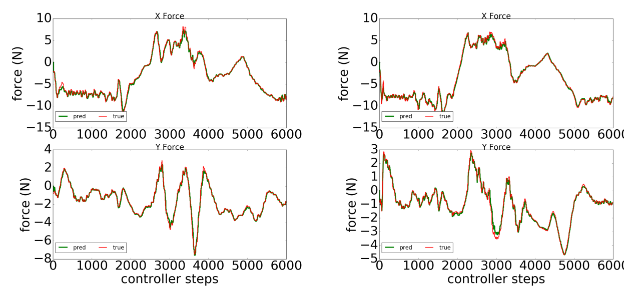
Red: Target; Pink: Using constant stiffness $\mathbf{K}^p_{free}$, Black: Using constant stiffness $\mathbf{K}^p_{max}$, Blue: Adapting impedance without efferent copy; Green: With efferent copy in feature vector

Fig. 7: Surface 1 Position Tracking



Red: Target; Pink: Using constant stiffness $\mathbf{K}^p_{max}$, Green: Using Surface 1 model without adaptation; Blue: Using online adaptation of previous model

Fig. 8: Surface 2 position tracking

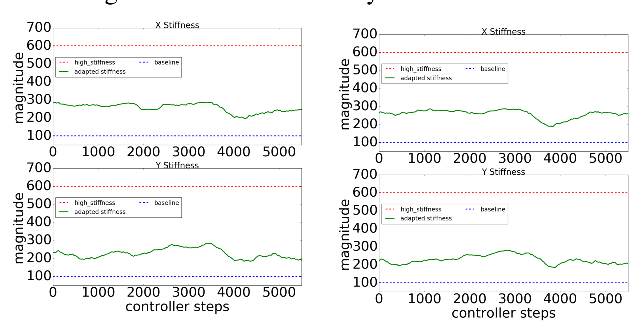


(a) Surface 1

(b) Surface 2

Red: Actual; Green: Predicted

Fig. 9: Forces Predicted by Forward Model



(a) Surface 1

(b) Surface 2

Red: $\mathbf{K}^p_{max}$; Blue: $\mathbf{K}^p_{free}$; Green: $\mathbf{K}^p_t$

Fig. 10: Stiffness Adaptation

Figure 7 show that in the absence of the forward model, the robot is unable to follow the desired trajectory well since the robot does not know the interaction forces. However, the use of the feed-forward model improves the tracking performance, and it further improved with the online impedance adaptation. The results show that the performance of the framework is comparable to a high stiffness controller strategy while requiring much lower impedance parameters (Figure 10a) and hence consume less energy.

However, as can be seen from Figure 7, the results do not show any significant improvement in adding the efferent copy to the input feature vector of the forward model (hypothesis 2). This lack is because the model can obtain maximum information for predicting the future forces from the current end-effector velocity and forces, leaving the commanded control to be redundant. Due to this and the fact that adding dimensions to the state-space makes the learning more computationally demanding, the final framework did not make use of the commanded forces (efferent copy) in the feature vector.

The third hypothesis required testing the generalisation capability of the framework. The framework is generalisable across different trajectories since the model learned the model for the surface polishing task using a trajectory that is different from the task trajectory.

The adaptability of the framework to new forces was tested by performing the same task using a different surface. When the previously learned model was directly used for this surface (without online improvement), the robot was not able to follow the trajectory accurately. However, by modifying this model online during task execution, it was able to achieve performance similar to the first surface quickly. This capability of the framework to generalise to different surfaces and trajectories is the critical advantage of using a *task-space, time-independent* variable impedance control framework.

V. Discussion

Variable impedance is vital for reliable and safe contact manipulation task. Learning impedance parameters directly is difficult and would require large training data. Using motivations from human motor control, we presented a framework that can learn forward model of a task online, which can predict interaction forces in a state-dependent fashion. The accuracy in prediction is used to vary the impedance online during task execution. The framework was tested on two sets of tasks: spring-pulling and board-polishing. The importance of having a time-independent variable impedance controller defined in the task-space for having a framework that can generalise to new environments, was demonstrated by deploying it on a new surface for the board-polishing task.

References

[1] M. Kawato, “Internal models for motor control and trajectory planning,” *Current opinion in neurobiology*.

[2] 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*.

[3] 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*.

[4] N. Hogan, “Impedance control: An approach to manipulation,” in *IEE-ACC-1984*.

[5] J. K. Salisbury, “Active stiffness control of a manipulator in cartesian coordinates,” in *Decision and Control including the Symposium on Adaptive Processes, 1980 19th IEEE Conference on*.

[6] S. Chiaverini and L. Sciavicco, “The parallel approach to force/position control of robotic manipulators,” *IEEE TRA*.

[7] S. A. Schneider and R. H. Cannon, “Object impedance control for cooperative manipulation: Theory and experimental results,” *IEEE TRA*.

[8] T. Wimböck, C. Ott, A. Albu-Schäffer, and G. Hirzinger, “Comparison of object-level grasp controllers for dynamic dexterous manipulation,” *IJRR*.

[9] M. Li, H. Yin, K. Tahara, and A. Billard, “Learning object-level impedance control for robust grasping and dexterous manipulation,” in *ICRA-2014*.

[10] K. Kronander and A. Billard, “Learning compliant manipulation through kinesthetic and tactile human-robot interaction,” *IEEE transactions on haptics*.

[11] M. Kalakrishnan, L. Righetti, P. Pastor, and S. Schaal, “Learning force control policies for compliant manipulation,” in *IROS-2011*.

[12] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, “Learning variable impedance control,” *IJRR-2011*.

[13] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, “Learning, planning, and control for quadruped locomotion over challenging terrain,” *IJRR-2011*.

[14] D. J. Braun, F. Petit, F. Huber, S. Haddadin, P. Van Der Smagt, A. Albu-Schäffer, and S. Vijayakumar, “Robots driven by compliant actuators: Optimal control under actuation constraints,” *IEEE Transactions on Robotics*.

[15] M. Howard, D. J. Braun, and S. Vijayakumar, “Transferring human impedance behavior to heterogeneous variable impedance actuators,” *IEEE Transactions on Robotics*.

[16] A. Kupcsik, M. Deisenroth, J. Peters, L. Ai Poh, V. Vadakkepat, and G. Neumann, “Model-based contextual policy search for data-efficient generalization of robot skills,” 2017.

[17] H. van Hoof, T. Hermans, G. Neumann, and J. Peters.

[18] J. R. Flanagan, M. C. Bowman, and R. S. Johansson, “Control strategies in object manipulation tasks,” *Current opinion in neurobiology*.

[19] R. S. Johansson and K. J. Cole, “Sensory-motor coordination during grasping and manipulative actions,” *Current opinion in neurobiology*.

[20] K. J. A. Kronander, “Control and learning of compliant manipulation skills,” 2015.

[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*.

[22] Y. Fan, W. Gao, W. Chen, and M. Tomizuka, “Real-time finger gaits planning for dexterous manipulation,” *IFAC-2017*.

[23] C. K. Liu, “Dextrous manipulation from a grasping pose,” in *ACM-TOG-2009*.

[24] 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*.

[25] S. Levine and V. Koltun, “Guided policy search,” in *ICML-2013*.

[26] 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*.

[27] R. Shadmehr and J. W. Krakauer, “A computational neuroanatomy for motor control,” *Experimental brain research*.

[28] N. T. Alberto, M. Mistry, and F. Stulp, “Computed torque control with variable gains through Gaussian process regression,” in *IEEE-RAS 2014*.

[29] T. Wimbock, C. Ott, and G. Hirzinger, “Impedance behaviors for two-handed manipulation: Design and experiments,” in *ICRA-2007*.

[30] K. J. A. Kronander, “Control and learning of compliant manipulation skills.”

[31] H. G. Sung, “Gaussian mixture regression and classification,” Ph.D. dissertation, Rice University, 2004.

[32] M. Song and H. Wang, in *SPIE Conference Series*, K. L. Priddy, Ed.

[33] W. Ahmad, “Incremental learning of gaussian mixture models.”

[34] P. Engel and M. Heinen, “Incremental learning of multivariate gaussian mixture models,” in *BSAI*. Springer, 2010.