An Adaptive Framework for Trajectory Following in Changing-Contact Robot Manipulation Tasks

Saif Sidhik[a], Mohan Sridharan[a], Dirk Ruiken[b]

[a] *Intelligent Robotics Lab, University of Birmingham, Birmingham, UK*
[b] *Honda Research Institute Europe GmbH, Germany*

Abstract

We describe an adaptive control framework for changing-contact robot manipulation tasks that require the robot to make and break contacts with objects and surfaces. The piecewise continuous interaction dynamics of such tasks make it difficult to construct and use a single dynamics model or control strategy. Also, the nonlinear dynamics during contact changes can damage the robot or the domain objects. Our framework enables the robot to incrementally improve its prediction of contact changes in such tasks, efficiently learn models for the piecewise continuous interaction dynamics, and to provide smooth and accurate trajectory tracking based on a task-space variable impedance controller. We experimentally compare the performance of our framework against that of representative control methods to establish that the adaptive control, prediction, and incremental learning capabilities of our framework are essential to achieve the desired smooth control of changing-contact robot manipulation tasks.

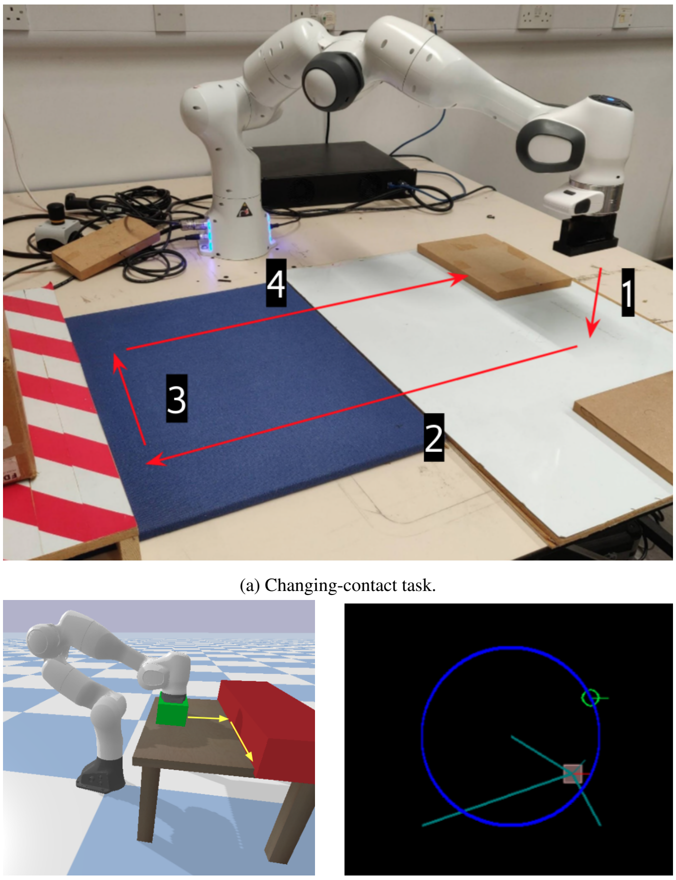
Keywords: Robot manipulation tasks, discontinuous interaction dynamics, variable impedance control, adaptive control

1. Introduction

As a motivating example, consider the robot manipulator in Figure 1a, which has to move its end-effector along a motion trajectory generated by an external planning system for a given goal. The target motion trajectory requires the robot to make and break contact with objects and surfaces. This task's dynamics, i.e., the relationship between the forces acting on the robot and the resultant accelerations, vary markedly before and after contact, e.g., after completing segment '1' and making contact with the table in Figure 1a. They also vary based on factors such as type of contact, surface friction (e.g., surface change in the middle of segments '2' and '4'), and applied force. Many industrial assembly tasks, e.g., peg insertion and stacking, and many human manipulation tasks are '*changing-contact*' tasks. The interaction dynamics of the robot performing these tasks are discontinuous when a contact is made or broken and continuous elsewhere, making it difficult to construct a single dynamics model and a strategy for smooth control of the robot's motion. A *two-level* model can be designed with separate continuous dynamics and distinct control laws within each of a set of discrete *dynamic modes*. The overall dynamics are then *piecewise continuous*, with the robot transitioning between modes as needed (Kroemer et al., 2019). Even with such a model, the nonlinear interaction dynamics can cause large discontinuities in dynamics in the transition regions, and the associated high forces, jerk, and vibrations can damage the robot and the objects in the domain.

Given a target motion pattern for a task of interest, we seek a control framework that enables the robot to accurately follow the pattern while ensuring that it: (i) uses low stiffness whenever possible to support compliance and expend less energy; (ii) can handle the continuously changing interaction dynamics that arise when the task requires the robot to be in continuous contact with an object (or environment); (iii) can handle the piecewise continuous interaction dynamics resulting from discrete changes in the environment or from making or breaking contacts; (iv) can quickly adapt to previously unseen interaction dynamics by acquiring and using a suitable adaptive controller; (v) can accurately anticipate collisions and contact changes in the target task, and switch smoothly to a *transition phase controller* that reduces the impact forces and vibrations during these transitions; and (vi) requires as few trials as possible to learn the predictive models necessary to perform the task. The associated measures of performance include trajectory-tracking accuracy, variation in controller stiffness, delays in task-completion, variation in impact forces, and smoothness of motion.

As a step towards the ideal control framework mentioned above, we present an adaptive framework for smooth control of changing-contact robot manipulation tasks. It unifies our work on a variable impedance controller for continuous contact tasks (Mathew et al., 2019), an extension for piecewise-continuous dynamics due to contact changes (Sidhik et al., 2020), and a recent approach for predicting contact changes and handling the associated discontinuities in dynamics (Sidhik et al., 2021). We advocate the need for a hybrid framework with one or more discrete modes, each with a predictive (forward) dynamics model of sensor measurements, a control law, and a relevance condition. The hybrid framework also includes *transition-phase controllers* for handling discontinuities corresponding to contact changes caused by collisions or impact-less mode transitions. Our choice of representation for model learning enables efficient adaptation of each mode's dynamics model and makes the identification of modes independent of the mo-



(a) Changing-contact task.

(b) 3D simulation setup.

(c) 2D test scenario.

Figure 1: (a) Illustrative changing-contact task where the robot experiences discontinuities in dynamics due to different surface friction in the middle of motions "2" and "4", and collisions at the end of "1", "2" and "4"; (b) Simulated environment used in the experiments in Section 4.4; robot approaches table from top, slides end-effector (green) along table until it collides with wall, and slides along wall and table; (c) 2D *multi-spring environment* with the robot lagging behind the target pattern with the AIC method used in Section 4.2.2: red block is robot end-effector, green lines are springs attached to end-effector.

tion direction and the magnitude of applied forces. Our controller within each mode is inspired by research in human motor control, which indicates that people learn to adapt arm stiffness to any new task by building and using internal models of task dynamics with generic (i.e., domain- or task-independent) and specific (i.e., domain- or task-dependent) representations in order to predict hand or object configurations and the forces during task execution (Kawato, 1999).

Our framework's design is subject to two caveats. First, for information and feedback, our framework primarily relies on the poses, velocities, and force-torque measurements at the end-effector of the robot. We intentionally do not use other sensors such as cameras, e.g., to provide external estimates of contact modes or locations, although it is possible to include them in our framework. This limitation is imposed to thoroughly explore the control capabilities we can support using these end-effector measurements; also, the other sensors are not always available and their use introduces additional uncertainty. Second, the target motion pattern, predictive models, and control laws are defined in the Cartesian space because it helps simplify the control and learning problems of interest and makes our framework more generalizable and independent of the robot platform. The main components of our framework are:

1. An adaptive variable impedance control strategy used within any given continuous contact mode, with the ability to automatically and incrementally adapt the forward model of the mode's dynamics. The forward model is used to revise the gain parameters (i.e., stiffness and damping) of the corresponding control law for accurate and compliant motion within that mode.

2. An approach to model the piecewise-continuous dynamics of changing-contact manipulation tasks without prior knowledge of all its modes or the order in which these modes appear. The approach also automatically identifies the modes for any given task, and transitions to appropriate existing or new modes during the execution of the target motion trajectory.

3. A contact change handling module that incrementally improves its predictions of contact locations, uses these estimated contact locations to minimize time spent in a transition phase controller, and achieves smooth motion dynamics during mode transitions by automatically adapting the velocity profiles and parameters of the controller.

Our related prior conference papers described the control of continuous-contact mode tasks, including the ability to learn forward models and adapt the parameters of an impedance controller for each mode (Mathew et al., 2019); the control of manipulation tasks involving changes in surfaces and types of contacts (Sidhik et al., 2020); and an initial framework that explored the use of transition-phase controllers for changing-contact manipulation tasks (Sidhik et al., 2021). These correspond to the first two components and part of the third component listed above. In this paper, we describe all components of our current framework for completeness, focusing on the *key novel contributions*:

(a) Detailed description of the use of transition-phase controllers in our framework to address the discontinuities of changing-contact manipulation tasks. We also describe new mechanisms that incrementally predict the contact changes, achieve the desired smooth transition to/from these controllers from/to the continuous mode controllers, and provide the desired smooth velocity (motion) profile.

(b) Extensive experimental evaluation in the context of changing-contact manipulation tasks—see Figure 1. In particular, we discuss new results comparing our framework's capabilities with existing baselines in controlled simulation environments, establishing the need for incremental learning, and evaluating our framework on a physical robot performing changing-contact tasks on a tabletop.

The experimental evaluation includes the comparison of the adaptive variable impedance control component for continuous dynamics (our base controller) with other adaptive control strategies from literature in a custom-built simulated world with different types of continuously changing environments. Then, the need for an incremental control and learning strategy for handling discontinuities in dynamics is demonstrated by comparing against a state of the art framework for offline, long-term prediction in hybrid systems. This baseline is then modified to show the advantages of the key design and representational choices of our framework. Furthermore, we report results of evaluating our framework on a physical robot performing changing-contact manipulation tasks on a tabletop.

We begin with a review of related work in Section 2. We then describe our framework and its components in Section 3, followed by a discussion of the experimental results in Section 4 and the conclusions in Section 5. The novel contributions are further emphasized in the appropriate sections.

2. Related work

A robot manipulator typically interacts with the environment through contacts that can be discrete, e.g., making or breaking contact with objects during assembly, or continuous, e.g., polishing or deburring surfaces. This section will review related methods for control of changing-contact manipulation tasks, including those that seek to model and address the piecewise continuous interactions and the nonlinearities in the dynamics.

2.1. Forward models in manipulation

Since it is difficult to provide the robot with comprehensive knowledge of target domain and task, methods have been developed to enable the robot to learn from experience, e.g., create a predictive (*forward*) model of action effects and use it to select actions (Beetz et al., 2010). Studies in human motor control also indicate the creation and use of such forward models for control tasks (Flanagan et al., 2003).

There are three kinds of forward models: (i) analytic models, which are typically modeled mathematically using Newtonian physics; (ii) learned models, which are built from data using machine learning methods; and (iii) hybrid models, which include a combination of learned and analytic models. Analytic models make predictions about robot and object motions (Fan et al., 2017; Liu, 2009) based on knowledge of mechanics and assumptions such as quasi-static mechanics, zero slippage, and point contacts (Chatterjee, 1999; Fazeli et al., 2020). Since these assumptions do not hold for many practical tasks and domains, forward models obtained by analytic methods often result in inaccurate prediction. In addition, these methods often require an explicit representation of their intrinsic parameters such as friction, mass, and coefficients of restitution, which are non-trivial to estimate (Kopicki et al., 2017).

Building forward models using machine learning methods involves learning an action-effect correlation using data obtained from expert demonstrations (Kronander and Billard, 2014; Huang et al., 2016) or from multiple trials executed by the robot (Levine and Koltun, 2013; Kupcsik et al., 2013). Many such methods have been developed to address the learning and control problems in robot manipulation (Kroemer et al., 2019), especially methods that draw on reinforcement learning (RL) principles (Stulp et al., 2012) and combine deep networks and RL for learning flexible behaviors from complex data (Andrychowicz et al., 2018; Hausman et al., 2018). In recent times, researchers have explored the use of deep neural networks (DNNs) for end-to-end learning of changing-contact manipulation tasks without explicitly learning a policy that models the domain dynamics (Nagabandi et al., 2020; Ajay et al., 2019). Although these methods reduce the need for domain models and prior knowledge, they require large labeled datasets, pose high-dimensional optimization problems, and tend to select the smoothest interpolation of the training data, which conflicts with the discontinuous impact dynamics of changing-contact manipulation tasks.

Methods that combine analytical and learning methods to build forward models tend to learn the difference between analytical models and the true dynamics of the interactions, capturing the improvements needed in the analytical model to match the observed environment (Gandhi et al., 2017). Despite the advantage of requiring fewer labeled training examples, these strategies rely on multiple restrictive assumptions regarding the type of contacts, friction models, object dynamics, etc.. They also require significant prior knowledge about the mathematical models to be used, and require at least a few trials of the possible scenarios (in the target domain) to be modeled. Even with state of the art sim-to-real strategies that reduce the amount of training needed on real robots, it is challenging to model aspects such as the dynamics of rigid bodies with friction in a real-time dynamics simulator (Johnson et al., 2016). These methods thus often require several hundred trials in the real world for fine-tuning the models acquired in simulation before they can be used on a physical robot (Ajay et al., 2019).

The piecewise continuous nature of changing-contact manipulation tasks has been used by some methods to build multi-level models, with a higher-level model identifying the current “mode” while the other levels model the dynamics of that mode (Buşoniu et al., 2018). Planning methods for manipulation often consider the discontinuities but assume prior knowledge of the models and modes, and often require many synthetically-generated training examples (Toussaint et al., 2018). Our framework, on the other hand, incrementally learns the dynamics of the robot’s interaction with its environment without prior knowledge of the dynamics (other than the target motion trajectory).

2.2. Controllers for continuous contact tasks

The model of the “plant” (system or robot) is nonlinear or unknown in many robot manipulators and practical control systems. Obtaining such a model is more challenging for manipulators due to factors such as interactions with other objects, unknown tools or objects at the end-effector etc. Researchers have developed many control schemes to address uncertainties in the system, and to adapt its parameters in response to observations of the system’s response. These adaptive controllers can be broadly grouped into model-reference adaptive controllers (MRAC), self-tuning regulators, and gain scheduling.

MRAC methods compute control signals that force the system's behavior to match that of a reference model (Zhang and Wei, 2017). They often employ approximations of the system's parameters and revise them using prior/observed data, but an incorrect reference model can result in instabilities. Self-tuning regulators typically model the plant as a linear time-varying system, and adapt or estimate the controller parameters online (Pezzato et al., 2020). They can converge to an optimal controller under certain conditions and are 'trainable' for repetitive tasks, with the estimate of parameters from a particular trial used as the initial estimate in the next trial. However, they struggle to adapt to changing objectives, which is common in robot manipulation. Gain-scheduling methods learn a time-indexed sequence of control parameters by repeating the task to update gains (Kramberger et al., 2018), or by learning gain profiles from training data (Lee et al., 2015). They include variable impedance control methods based on reinforcement learning or 'learning from demonstrations' (Abu-Dakka et al., 2018; Calinon et al., 2010; Rozo et al., 2016), which provide a time-varying impedance profile as a function of robot state or any observation made by the robot. Since these methods often need large training sets or comprehensive domain knowledge, more recent methods have the agent repeat a task until a desired level of performance is achieved (Gams et al., 2014; Kramberger et al., 2018). They try to reduce trajectory tracking error and *periodic* disturbances by learning a corrective term for the control law that is a linear function of tracking error, measured disturbance, or time.

Our adaptive variable impedance controller (AVIC) for each mode is similar to the self-tuning regulators, but it builds on our prior work on control for continuous-contact tasks (Mathew et al., 2019) and piecewise-continuous interaction dynamics (Sidhik et al., 2020); it uses the prediction error of an incrementally learned forward model to guide the adaptation of control parameters (Section 3.1).

2.3. Control of hybrid systems

Many methods have been developed for the control of hybrid systems. For example, stabilizing controllers based on Lyapunov arguments have been developed for switched systems (Johansson, 2003; De Schutter et al., 2009), methods based on optimal control theory have been used for hybrid systems in manufacturing (De Schutter et al., 2009), and model predictive control (MPC) methods have been applied to hybrid systems (De Schutter et al., 2009).

The hybrid system formulation with a set of modes or the design of the corresponding controller with phases provides benefits for robot manipulation (Romano et al., 2011). For a pick-and-place task, these controllers can have distinct objectives and behavior requirements for phases such as 'approach', 'grasp' and 'release'. Different strategies for sequencing motion primitives have also been used to solve manipulation tasks, but they assume the existence of a library of modes or motion primitives, or segment a sequence of primitives from human demonstrations (Niekum et al., 2013). Since these methods do not consider the interaction dynamics or try to reduce effects of impact, the learned policy is dependent on the environment, movements, and the sequence of modes.

In a departure from the existing methods, our framework supports the construction and use of adaptive variable impedance controllers (AVIC) in the task-space for each identified mode in the target task. As we show later, this design improves generalization and adaptability.

2.4. Modeling impacts and contact changes

Collisions and the associated impact dynamics introduce critical challenges to motion planning and control of robots in applications such as locomotion (Wieber et al., 2016) and manipulation (Kemp et al., 2007). Even a single collision is a complex interaction where object interpenetration is prevented by material deformation, and which often occurs on a scale far below the resolution of practical sensors (Halm and Posa, 2021). Predicting the dynamics of collisions accurately is challenging even with precise knowledge of the material and geometry of the objects and initial conditions, and it is impractical to provide the necessary knowledge in most domains (Chatterjee, 1997). It is thus common for robotics researchers to make coarse approximations of the contact dynamics, e.g., rigid-body assumption, to make the problem more tractable; for background information, please see (Brogliato, 2019). When impacts occur, rigid-body models approximate the event as an instantaneous change in velocity due to an impulsive force. However, seemingly minor changes in the mathematical models can result in significantly different predictions under identical initial conditions, and existing methods are often unable to capture real-world behaviors with available models (Fazeli et al., 2020; Stoianovici and Hurmuzlu, 1996).

Unrealistic contact models are an important reason for the gap between simulated and real-world performance in robotics (Parmar et al., 2021). In addition, velocity measurements are extremely sensitive to time, as they change almost instantaneously during impact, and sensor measurements are noisy in the real world. The smoothing effect of deep neural networks, which are increasingly becoming the state of the art for many robotics problems, is particularly harmful for modeling impacts and collisions. There is a sparsity of reliable data points that can be collected during and around the time of impact, and regular sensors are not very reliable during impact. For a good discussion of the main challenges in using deep learning or analytical methods to model contact dynamics, please see Parmar et al. (2021); these challenges include the degradation of model performance with increasing stiffness.

The challenges in modeling contact dynamics can be decoupled by first predicting the *position of contact points* and then using a 'safer' controller in the predicted contact regions. Static contact properties such as *contact positions* and *direction of impact* can be estimated reliably, particularly if the plan (i.e., motion trajectory) is known beforehand, using either tactile (e.g., force-torque) sensors or coarse depth images (e.g., point clouds) of the objects. Acquiring training samples for such measurements are also usually easier than an analytical analysis of the interaction.

Existing work has demonstrated a robot's ability to acquire the labels of training samples by interacting with the domain objects and learning high-level rules that can be used for planning (Ugur and Piater, 2015). Such interactive perception methods have also been used to estimate constraints or the physical properties of objects (Katz and Brock, 2011; Barragän et al., 2014). These methods do not need extensive pre-training; also, they help disambiguate between scenarios and to observe otherwise latent properties (Kroemer et al., 2019). For instance, a robot can figure out if an object is movable by pushing it. However, such methods require the robot to perform the target task multiple times to build models or optimize model parameters, especially when modeling the dynamics of the interaction. In our contact-anticipation model (Section 3.3), we focus on interactively improving the the robot's estimate of the location of the contacts involved in the task under consideration; we adapt and use a Kalman-filter for this task. These estimated contact regions are then used to define 'transition' regions in the robot's workspace, where it can use a safer control strategy because it expects contact changes to occur.

2.5. *Transition-phase controllers*

Using a set of 'special' controllers is a common strategy for handling contact changes. Methods that use a transition-phase controller for changing-contact robot manipulation tasks focus on minimizing the discontinuities in the dynamics such that the controller is stable and the desired motion pattern is followed accurately after transition (Mills and Lokhorst, 1993; Marth et al., 1993; Sidhik et al., 2020). However, most methods switch to a different controller only after a contact is detected, which can result in significant disruptions in the dynamics when the switch is made. These disruptions can significantly increase the energy intake, cause sudden spikes in force and acceleration, and potentially damage the robot or domain objects. Other work reduces the effects of impact in the guard regions of manipulation tasks using a transition-phase controller with a constant low velocity (Hyde et al., 1997). Unlike existing methods, our approach for handling contact changes predicts the position of contacts, modifies the velocity and stiffness to reduce impact and vibration during transitions to/from a transition-phase controller, and automatically revises the approach velocity to obtain the desired force on impact (Section 3.5.2).

To minimize delays and deviation from the target motion pattern, the approach velocity of the transition-phase controller should come into effect only when the robot is about to make a contact. Modifying the velocity requires changes to the timeline of the motion pattern, causing deviations and potentially sacrificing tracking accuracy. For kinematic time-optimal motion, different variants of trapezoidal velocity profiles are typically used (Biagiotti and Melchiorri, 2008). Motion smoothness (at least up to jerk) can be guaranteed by making the transition trajectory continuous. To make the motion smooth in acceleration and jerk, the motion profile needs to be at least C^4 smooth. Many methods exist to create C^4 smooth trajectories using multiple trajectory segments (Ahn et al., 2004; Nam and Yang, 2004). There are also many minimum-jerk motion profiles in literature (Piazzi and Visioli, 2000; Freeman, 2012). For example, a trapezoidal C^4-smooth motion profile for point-to-point motion has been described using a seventh-order C^3 polynomial function as the velocity profile (Grassmann et al., 2018). Such methods have many hyper-parameters that need to be tuned for the target task, and requires knowledge of the limits of the system's jerk and higher order motion derivatives. Our framework, on the other hand, uses a novel velocity profile (for switching to the transition-phase controller) that is simpler in formulation, has no additional hyper-parameters, and has continuous derivatives of all orders at every point of the function, making it C^∞ smooth (see Section 3.5.2).

3. Framework description

Figure 2 is an overview of our framework which seeks to model the piecewise-continuous interaction dynamics and address the discontinuities during changing-contact manipulation tasks. The framework's base controller is a hybrid force-motion, adaptive, variable impedance controller for tasks involving continuous contact and contact changes without impact, as described in Sections 3.1 and 3.2. For any particular contact mode, this controller enables the robot to incrementally model and revise a forward model that predicts the sensor measurements. The measured prediction error is used to revise the forward model and the gain (i.e., stiffness) parameters of the control law for accurate and compliant motion in that mode. Also, the robot is able to automatically detect known modes or previously unknown modes; for the former (i.e., known mode), the robot uses the corresponding forward model and control law, whereas it acquires a new forward model for the latter.

Since the base controller does not address the discontinuities experienced during contact mode changes, our framework includes a contact change handling module. This module incrementally revises its task-space predictions of contact locations, i.e., the anticipated mode transition regions, using a Kalman filter (Section 3.3). The other innovation is the use of these predictions to minimize the time spent in the transition phase, and to adaptively set the parameters of a transition phase controller that is used to achieve a smooth motion profile and a desired impact force (if collisions are expected), as described in Sections 3.4 and 3.5. Once the mode transition is completed, the robot uses the corresponding forward model and controller for the resultant model.

As stated earlier, our framework takes as input the task-space target motion pattern and the corresponding force control target profile for the robot to follow. The target trajectory P is provided as a sequence of mappings from time to the end-effector pose and force (for force control). It is obtained through a single demonstration of the task by a human moving the manipulator. Since our framework's controllers operate in the Cartesian (i.e., task) space, the trajectory is in the form of segments that are (each) assumed to be smooth, continuous, and jerk-free. Transition between segments is accompanied by a change in the direction of (force, motion) control, and P does not explicitly account for any collisions, i.e., contact points are not labeled.

The framework's inputs also include sensor measurements of the forces, torques, and position at the end-effector. In the

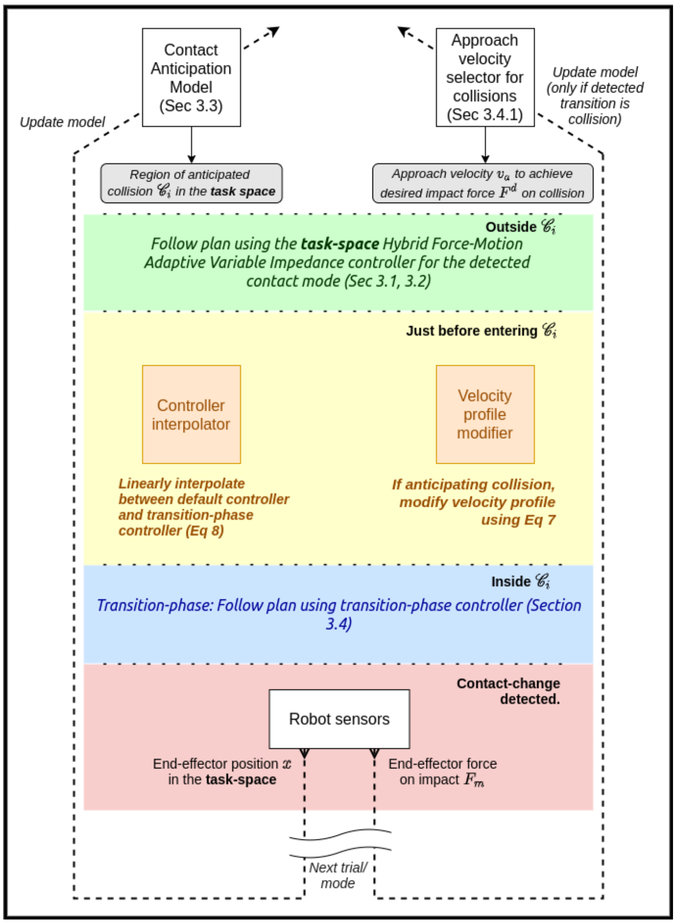

Contact Anticipation Model (Sec 3.3)
Update model
Approach velocity selector for collisions (Sec 3.4.1)
Update model (only if detected transition is collision)
Region of anticipated collision $\mathscr{C}_i$ in the **task space**
Approach velocity v_a to achieve desired impact force F^d on collision
Outside $\mathscr{C}_i$
Follow plan using the **task-space** Hybrid Force-Motion Adaptive Variable Impedance controller for the detected contact mode (Sec 3.1, 3.2)
Just before entering $\mathscr{C}_i$
Controller interpolator
Velocity profile modifier
Linearly interpolate between default controller and transition-phase controller (Eq 8)
If anticipating collision, modify velocity profile using Eq 7
Inside $\mathscr{C}_i$
Transition-phase: Follow plan using transition-phase controller (Section 3.4)
Contact-change detected.
Robot sensors
End-effector position x in the **task-space**
End-effector force on impact F_m
Next trial/ mode

Figure 2: Overview of the framework for smooth control of changing-contact manipulation tasks. Framework's components are: (i) an adaptive variable impedance control strategy based on an incrementally updated predictive model for each of a set of discrete continuous-contact modes; (ii) an incremental approach to automatically model the piecewise-continuous dynamics of changing-contact manipulation tasks without prior knowledge of the modes; and (iii) a contact anticipation model that incrementally estimates contact change locations, and handles discontinuities by adapting velocity profiles and parameters of suitable transition-phase controllers.

absence of our framework, the robot would use a position controller to follow the provided motion pattern without compensating for the changing interaction dynamics as the task progresses. The robot would thus have to use a high-stiffness controller within a contact mode to counter the continuously varying environment dynamics. There would still be significant discontinuities in the motion dynamics when the contact modes change, i.e., when the robot makes or breaks contact, or when the environmental factors (e.g., surface friction) change significantly. High-impact collisions could also damage the robot or objects involved. We begin the description of our framework's components and capabilities with the description of the forward model and controller used in any particular mode.

3.1. Forward model and controller for a mode

The basic controller in our framework follows the standard impedance control formulation with a force control term, making it a hybrid force-motion impedance controller:

$$\boldsymbol{u}_t = \boldsymbol{H}_t + \mathbf{K}^{\mathbf{p}}_t \Delta \boldsymbol{x}_t + \mathbf{K}^{\mathbf{d}}_t \Delta \dot{\boldsymbol{x}}_t + \boldsymbol{u}^{\mathbf{fc}}_t + \boldsymbol{u}^{\mathbf{ff}}_t \tag{1}$$

where $\boldsymbol{u}_t$ is the robot's control command (i.e., task space force) at time t; $\boldsymbol{H}$ denotes the other dynamics compensation terms (inertia, Coriolis and gravity); $\mathbf{K^p}_t$ and $\mathbf{K^d}_t$ are the (positive definite) stiffness and damping matrices of the feedback controller for motion; $\boldsymbol{u}^{\mathbf{fc}}_t$ is force feedback control command to achieve orthogonal force targets if desired; and $\Delta \boldsymbol{x}_t$ and $\Delta \dot{\boldsymbol{x}}_t$ are the errors in the end-effector position and velocity at each instant. In the absence of external disturbances, the feed-forward term $\boldsymbol{u}^{\mathbf{ff}}_t$ can be zero. However, if there are external wrenches acting on the end-effector, a good forward model could be used to provide appropriate values for $\boldsymbol{u}^{\mathbf{ff}}_t$ that can help the robot follow P accurately without being affected by those disturbances.

In our formulation, the forward model of each contact mode is learned as the robot attempts to follow P for the given task. To avoid explicit dependence on time, a Gaussian Mixture Model (GMM) is fit over points of the form $[\boldsymbol{S}_{t-1}, \boldsymbol{D}_t]$, where $\boldsymbol{S}_t$ can be any combination of features that uniquely represent the robot's state for the task, and $\boldsymbol{D}_t$ denotes that interaction effects experienced by the robot's end-effector at time t. $\boldsymbol{S}_t$ can contain information about end-effector pose ($\boldsymbol{x}_t$), velocity ($\dot{\boldsymbol{x}}_t$), and forces ($\boldsymbol{F}^{ee}_t$), while $\boldsymbol{D}_t$ can denote measurable interaction effects such as end-effector forces ($\boldsymbol{F}^{ee}_t$) and torques ($\boldsymbol{\tau}^{ee}_t$).

In our framework, we aim to predict the end-effector forces and torques from previous measurements of end-effector velocities and wrenches. However, instead of their 3D vector representations we use the *magnitudes* of force, torque, and end-effector velocity (linear and angular separately) for modeling and prediction. Since the magnitudes of frictional forces and torques are independent of the direction of motion (for objects having consistent friction properties) and depend only on the relative speed of motion of the objects, such a simplified representation is sufficient to model and predict the end-effector forces and torques along the direction of motion. This reduced representation of forces and torques makes the process of acquiring the forward models computationally efficient and *independent of the direction of motion*. Any forward model predicts the forces and torques along (or against) the direction of motion. Since the end-effector's direction of motion is always known, the components of force and torques along the axes of motion can be recovered when needed. The state space where the forward model is learned can therefore be represented as $X_t = [\boldsymbol{S}_{t-1}, \boldsymbol{D}_t]$, with:

$$\boldsymbol{S}_{t-1} = \left[||\dot{\boldsymbol{x}}^{lin}_{t-1}||, ||\dot{\boldsymbol{x}}^{rot}_{t-1}||, ||\boldsymbol{F}^{ee}_{t-1}||, ||\boldsymbol{\tau}^{ee}_{t-1}|| \right] \tag{2}$$

$$\boldsymbol{D}_t = [||\boldsymbol{F}^{ee}_t||, ||\boldsymbol{\tau}^{ee}_t||] \tag{3}$$

where $||\dot{\boldsymbol{x}}^{lin}||$, $||\dot{\boldsymbol{x}}^{rot}||$, $||\boldsymbol{F}^{ee}||$, and $||\boldsymbol{\tau}^{ee}||$ are the magnitudes of the linear velocity, angular velocity, force, and torque (respectively) at the end-effector.

The forward model's predictions determine the feed-forward term $\boldsymbol{u}^{\mathbf{ff}}_t$ in Equation 1, which cancels out the effect of the predicted wrenches during motion, revising the control equation:

$$\boldsymbol{u}_t = \boldsymbol{H}_t + \mathbf{K}_t^{\mathbf{p}}\Delta\boldsymbol{x}_t + \mathbf{K}_t^{\mathbf{d}}\Delta\dot{\boldsymbol{x}}_t + \lambda_{t-1}\boldsymbol{W}_t^{pred} + \boldsymbol{u}_t^{\mathbf{fc}} \tag{4a}$$

$$\mathbf{K}_t^{\mathbf{p}} = \mathbf{K}_{free}^{\mathbf{p}} + (1-\lambda_{t-1})(\mathbf{K}_{max}^{\mathbf{p}} - \mathbf{K}_{free}^{\mathbf{p}}) \tag{4b}$$

$$\lambda_t = 1 - \frac{1}{1+e^{-r(\varepsilon_t-\varepsilon_0)}} \tag{4c}$$

where $\lambda_{t-1}\boldsymbol{W}_t^{pred}$ is the weighted feed-forward wrench (i.e., end-effector forces and torques) predicted by the forward model associated with the current mode m_t. The factor λ_t, a function of the accuracy of the forward model at instant t, maps the error in prediction from the forward model (ε_t) to a value between 0 and 1, e.g., logistic function in Equation 4c. The logistic growth rate r and the Sigmoid midpoint ε_0 are hyper-parameters tuned for the task. This control law formulation relies on the feed-forward term only if the forward model's predictions are found to be accurate; if not, the feedback control term is prioritized. Equation 4b describes the adaptation of stiffness parameters as a function of the prediction accuracy. $\mathbf{K}_{max}^{\mathbf{p}}$ is the maximum allowed stiffness, and $\mathbf{K}_{free}^{\mathbf{p}}$ is the minimum stiffness for accurate position tracking in the absence of external disturbances (i.e., free space motion). The damping term is updated as $\mathbf{K}_t^{\mathbf{d}} = \sqrt{\mathbf{K}_t^{\mathbf{p}}/4}$ considering the manipulator to be a critically-damped system. This formulation enables the robot to follow P accurately using high feedback gains if the forward model is unreliable, but be compliant if the feed-forward term provides suitable compensation.

To incrementally update a forward model during task execution, our framework uses an online variant of GMM called the Incremental GMM (IGMM) (Song and Wang, 2005; Engel and Heinen, 2010; Ahmad, 2006). IGMM can update model parameters and incorporate additional components in the mixture model using hyperparameters (closeness and frequency) set by the designer. For more information about the incremental algorithm used, please see (Engel and Heinen, 2010). IGMM internally uses a variant of the Expectation-Maximization (EM) to maximize the following likelihood function:

$$L(\theta) = p(\mathbf{X}|\theta) = \prod_{n=1}^{T} p(X_n|\theta) = \prod_{n=1}^{T}\left[\sum_{j=1}^{M} p(X_n|j)p(j)\right] \tag{5}$$

where $\theta = (\mu_j, \sigma_j, p_j)$ for $j = 1...M$ are the parameters of the M components of the GMM; and $\mathbf{X} = (X_1, ..., X_T)$ represents the points to be fit, with $X_t = [\boldsymbol{S}_{t-1}, \boldsymbol{D}_t]$. Each point contains information about the *previous* end-effector state, along with the *current* wrench. Once trained, the forward model provides a function:

$$f_{\mathrm{fm}} : \boldsymbol{S}_t \mapsto \boldsymbol{D}_{t+1} \tag{6}$$

that predicts D_{t+1} at the next time step as a function of the current (measured) value of $\boldsymbol{S}_t$, using Gaussian Mixture Regression (GMR) (Sung, 2004). The incremental nature of our dynamics model allows the robot to capture the smoothly varying interaction dynamics between the robot end-effector and the environment within any particular contact mode.

3.2. Detecting contact modes

In a typical manipulation task, the interaction dynamics are non-smooth when a contact change occurs, moving the state of the hybrid piecewise continuous system to a new contact mode. Our approach for recognizing known modes and identifying new ones is based on the observation that any change in mode is accompanied by a sudden, significant change in the sensor measurements. In our framework, the robot responds to pronounced changes in force-torque measurements by briefly using a high-stiffness control strategy while quickly obtaining a batch of sensor measurements to confirm and respond to the transition. The robot learns a new dynamics model if a new mode is detected, and transitions to (and revises) an existing dynamics model if the transition is to a known mode.

The key factor influencing the reliability and generalizability of our approach is the choice of feature representation for the modes. This representation is task-dependent and should be able to uniquely identify the different contact modes in the task. For example, for the task of sliding an object over surfaces with different values of friction (see Figure 8), the property that strongly influences the end-effector forces (F^{ee}) is the friction coefficient between the object and the surface. When two objects slide over each other at constant velocity, F^{ee} is proportional to the applied normal force (R) and the friction coefficient (μ) (assuming the relative orientation of their surface normals do not change); μ can then be estimated as:

$$\mu \propto \frac{\|F^{ee}\|}{R} \tag{7}$$

A concise feature representation for this task is thus $\frac{\|F_t^{ee}\|}{R_t}$, which has the effect of making mode classification independent of the magnitude of the applied force.

Similar features can be identified for distinguishing between other contact modes in specific tasks. For instance, for tasks that require the robot to slide over the same surface using different types of contact (e.g., edge contact, face contact), the end-effector torque measured by the robot provides a more distinguishable feature for differentiating between modes. For such tasks, $[\frac{\|\tau\|}{R}, \frac{\|F^{ee}\|}{R}]$ can thus be used as the feature representation for each mode; it supports generalization over different normal forces while reliably capturing the factors influencing the nature of the contact. We have previously demonstrated the effectiveness of such a feature representation in distinguishing between contact types while being invariant to the direction of motion as well as to the magnitude of the applied normal force (Sidhik et al., 2020). A similar one-time exercise can be used to determine a feature representation with the desired generalization capability for other changing-contact tasks.

The management of modes is based on an online incremental clustering algorithm called Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) (Zhang et al., 1997) in the Scikit-learn library (Pedregosa et al., 2011). This algorithm incrementally clusters incoming data while respecting given memory and time constraints, without examining all data points or clusters. Each cluster is considered to represent a mode in the abstract feature representation space, with the clusters updated

Algorithm 1: Control loop of base controller for piecewise-continuous dynamics

```
Input : Desired motion pattern as sequence of task
        space way-points; Control parameters:
        $\mathbf{K}^p_{free}, \mathbf{K}^p_{max}$; Dynamics models of modes
        $\mathcal{M} = \{f_i : i \in [1, N]\}$; Current mode: $m = 0$.
1  while Motion pattern not complete do
2      if mode transition detected then
           // Set high stiffness
3          $\mathbf{K}^p_t \leftarrow \mathbf{K}^p_{max}$
           // Detect (new/existing) mode
4          $m$ = detect_classify_mode()
           // Populate new model for new mode
5          if new mode found then
6              $\mathcal{M} = \mathcal{M} \cup f_m$
7          end
8      end
9      Update $f_{\mathrm{fm}|m}$ online and use it for control in the
         identified mode (Eq. 4)
10 end
```

using batches of the feature data. The fraction of the input feature vectors assigned to any cluster determines the confidence in the corresponding mode being the current mode. If the highest such confidence value is above a threshold, the dynamics model of that mode is used and revised until a mode change occurs. If the feature vectors are not sufficiently similar to an existing cluster, a new cluster (i.e., mode) and the corresponding dynamics model are constructed, used, and revised (Section 3.1) until a mode transition occurs.

Algorithm 1 describes the framework's control loop for a changing-contact manipulation task (e.g., sliding an object on a surface) based on the base controller (i.e., a hybrid force-motion variable impedance controller) described so far. The robot attempts to move through the given motion pattern until it is completed. When a change in contact mode is detected (line 2) based on a substantial change in the sensor measurements (from the end-effector), the robot responds by setting a high stiffness (line 3), collecting sensor measurements, confirming whether a transition has occurred to a new or existing mode (line 4), and creating new forward (dynamics) models if necessary (lines 5-7). In the absence of a mode transition (i.e., the change in sensor measurements that triggered the high stiffness operation was an anomaly), the robot continues with the current dynamics model for subsequent motion (line 9).

3.3. Predicting contact changes

A robot manipulator following a given motion pattern in a static environment (i.e., no moving objects other than the robot) can experience discontinuities in dynamics due to two types of transitions: (i) impact-based transitions (i.e., collisions); and (ii) impact-less transitions. Collisions occur when the moving end-effector comes in contact with a fixed object in its workspace. Any collision-based mode change will suddenly remove at least one degree of freedom (DoF) of the robot in the functional coordinate space. Collisions are thus characterized by large spikes in force-torque measurements as a function of the approach velocity and factors such as coefficient of restitution and hardness of the objects involved. Also, there is a discontinuity in the end-effector velocity, with at least one dimension set to zero in the functional coordinates, leading to spikes in acceleration and jerk as well. Discontinuities due to impact-less transitions, on the other hand, occur when there are significant changes in the dynamics due to other factors instead of collisions. A simple example involves the end-effector sliding across two surfaces with different values of frictional resistance (Figure 1a). The robot experiences sudden changes in sensor measurements as it crosses the boundaries between the surfaces, but the transition may not result in large impact forces or a significant drop in velocity; unlike the impact-based transition, the impact-less transitions occur without a loss in degree of motion freedom.

While the base controller in our framework detects mode changes and uses (or learns) appropriate forward models in each mode, it does not address the discontinuous dynamics during the transition between modes. To handle the discontinuities, the robot needs to be able to anticipate these transitions. Anticipating impact-based transition (i.e., collisions) by predicting impact forces or time to collision is challenging because these parameters are influenced by robot dynamics and controller parameters, e.g., reducing the velocity or stiffness reduces the impact force and increases time to contact. On the other hand, static contact parameters such as end-effector position during impact and direction of contact force can be predicted more reliably. These parameters do not change significantly between task repetitions if we can make the reasonable assumption that the motion pattern and environmental attributes do not change significantly between repetitions of the task.

Similar to the target motion pattern and the base controller in our framework, a contact's position is also represented in the task-space of the robot. Our contact anticipation model encodes the robot's belief about the position of each expected contact in the target motion pattern as a multivariate Gaussian, with the covariance ellipsoid representing the associated uncertainty and the "region of anticipated mode transition" C. During any trial of the task, the robot expects a specific contact position $\mathbf{c}$ to lie within the corresponding C. The robot uses a Kalman filter to update its estimate of the location of each contact over a small number of trials. The state update equation is: $\dot{\mathbf{c}} = \mathbf{A}\mathbf{c}+\mathbf{B}\mathbf{u}_k+w$, where $\mathbf{c}$ is the contact position, $\mathbf{A}$ is the object's self-activation (I for positively activated objects), $\mathbf{B}$ is the control matrix capturing the effect of action $\mathbf{u}$ on contact position, and w is Gaussian noise modeling the uncertainty in the contact location. The sensor model uses the end-effector pose (obtained by forward kinematics with joint positions) as a measurement when a contact is detected; noise in the sensor model depends on the joint encoder noise and forward kinematics. The corrected estimate of the contact position results in a reduced covariance ellipsoid for subsequent trials. Although this approach supports contact with movable objects, we only consider contact with stationary objects in this paper. These simplifications result in Gaussian updates using the noisy measurements based on the robot's kinematics model (since sensor input is from an FT sensor) each time the robot experiences a contact change.

3.4. Transition-phase controller

Given the ability to predict contact changes, as described above, the robot still needs a suitable controller to use during the transition. A key contribution of our framework is the use of a transition-phase controller to achieve this objective. In our framework, the transition-phase controller used during mode transitions has the same structure as our base (variable impedance) controller (i.e., Equation 4a). The difference is in the choice of the control parameters, which can be different based on the type of transition: impact-based or impact-less. The control equation is thus given by:

$$\boldsymbol{u} = \mathbf{K}^{\mathbf{p}*}\Delta\boldsymbol{x} + \mathbf{K}^{\mathbf{d}*}\Delta\dot{\boldsymbol{x}} + \boldsymbol{u}^{\mathbf{ff}} + \boldsymbol{H} \tag{8}$$

where $\mathbf{K}^{\mathbf{p}*}$ and $\mathbf{K}^{\mathbf{d}*}$ are the stiffness and damping parameters that can vary depending on the type of transition. Next, we describe how this controller supports the desired characteristics for the two types of transitions.

3.4.1. Transition-phase controller for collisions:

For impact-based transitions, the permitted impact force may differ based on the task, e.g., a particular force may damage delicate objects but it may be necessary for other tasks. Our controller allows the designer to impose a task-specific limit on the maximum allowed impact force. In addition, we figured out experimentally that reducing the controller stiffness helps reduce the jerk in motion after impact by providing compliance, but has no significant effect on impact forces because the error and stiffness term in the feedback control loop come into effect only after contact is made. Also, the approach velocity was observed to be directly proportional to the impact force, especially when the robot registered a contact while moving in free space. Based on these insights, the transition-phase controller was designed to operate at a lower stiffness (e.g., a fraction of the maximum allowed impact force for the task) to reduce vibrations during the transition. Also, we used a simple linear (regression) model to capture the relationship between impact force and approach velocity. This model was then used to compute the approach velocity for any (given) desired force on impact.

Since the robot does not initially have a model of the relationship between impact force and approach velocity, it starts with a safe low velocity during the first trial of any given task with a target force on impact. The difference between the target and measured impact force is used to revise the approach velocity for the next iteration of the task:

$$\Delta v_a = \beta(F_d - F_m) \tag{9}$$

where Δv_a is the change in approach velocity, F_d is the desired impact force along direction of motion, F_m is the measured impact force, and β is a learning rate that is ideally less than or equal to the slope of the function relating impact force to approach velocity. Over a small number of trials, this method enables the robot to compute and use a task-specific approach velocity for a desired impact force.

3.4.2. Transition-phase controller for impact-less transitions:

If the robot expects an impact-less transition in C, it does not have to reduce its velocity and can continue following the original plan. Reducing velocity would delay task completion, which is undesirable. At the same time, high stiffness values will result in steady motion and more accurate sensor measurements, which will help in mode identification. Our transition-phase controller thus uses a high stiffness for impact-less transitions. Specifically, the parameter values for this transition phase controller (Equation 8) are set to be the maximum allowed values of the adaptive variable impedance controller (base controller) for the current task, i.e., $\mathbf{K}^{\mathbf{p}*} = \mathbf{K}^{\mathbf{p}}_{max}$.

3.5. Smooth transition between controllers

The transition-phase controller described above is designed to reduce discontinuities during mode changes. However, the robot still needs a strategy to smoothly transition from the base controller to a transition-phase controller as it enters the region of anticipated mode transition (C). Specifically, the robot needs to transition from base controller output $\mathbf{u_1}$ (for current contact mode) to a suitable transition-phase controller with output $\mathbf{u_2}$. A key characteristic of our framework is its ability to meet these requirements as described below.

3.5.1. Transition strategy:

We use a linear interpolation between $\mathbf{u_1}$ and $\mathbf{u_2}$ over a time window $[0, T]$ to achieve the smooth transition to the transition-phase controller:

$$\mathbf{u} = (1-\alpha)\mathbf{u_1} + \alpha\mathbf{u_2}; \qquad \alpha = t/T \qquad t \in [0, T] \tag{10}$$

where T is the desired duration of the transition between the controllers. Since $\mathbf{u_1}$ and $\mathbf{u_2}$ are represented in the same (task) space, as long as the outputs from the two controllers ($\mathbf{u_1}$ and $\mathbf{u_2}$) are individually smooth, the output of the combination will also be smooth. Recall that the output of the fixed transition-phase controller ($\mathbf{u_2}$) corresponds to a (relatively) low- or high-stiffness depending on the type of transition. The interpolation ensures that the switch to the transition-phase controller occurs smoothly by the time the robot reaches p_c, the first point in its motion trajectory that intersects with the estimated region of expected contact change (C). A similar approach is used to smoothly (and quickly) transition from the transition-phase controller to a normal controller after contact is made.

3.5.2. Velocity profile shaping:

Recall that transition-phase controllers for handling collisions use a lower velocity than in the original kinematic sequence P to reduce the force on impact. Also, the switch to this controller will take place at different points in the trajectory as the region C is revised over time. The trajectory's timeline thus has to be modified to account for the modified velocity profile. To achieve this objective, we enable the robot to create a new velocity profile and the corresponding time-mapping. Our formulation results in motion that is smooth and continuous at all orders, i.e., it is C^∞ smooth.

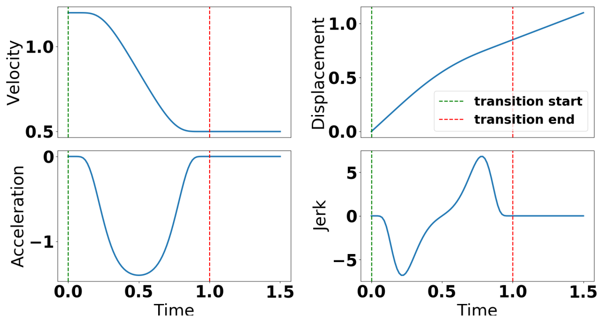


Figure 3: Velocity plots with matched position, acceleration, and jerk plots. Velocity varies from 1.2 to 0.5 in unit time.

Without loss of generality, assume that the target motion pattern (P) is along one dimension. Also, assume that the transition starts at time t_1 with v_1 and has to be completed at t_2 with velocity v_2 as the robot crosses boundary point p_c of C. Our velocity profile is then defined as:

$$v(\tau) = \begin{cases} v_1 + \frac{(v_2 - v_1)e^{-1/\tau}}{e^{-1/\tau} + e^{-1/(1-\tau)}} & \text{if } 0 < \tau < 1, \\ v_1 & \text{if } \tau \leq 0, \\ v_2 & \text{if } \tau \geq 1 \end{cases} \tag{11}$$

where $\tau = t/T = t/(t_2 - t_1)$. For $\tau \in (0, 1)$, $e^{-1/\tau}$ has continuous derivatives at all orders at every point τ on the real line. Since $v(\tau)$ has a strictly positive denominator for all points in its domain and velocity limits are enforced $\forall \tau \notin [0, 1]$, this profile provides a smooth transition from v_1 to v_2 over $[t_1, t_2]$ and $v(\tau)$ is continuous despite its piece-wise definition. Acceleration and jerk are computed as first- and second-order derivatives of $v(\tau)$ with respect to τ, and the motion trajectory is obtained by integrating the velocity profile. Figure 3 shows some illustrative plots to demonstrate that all motion derivatives are continuous.

4. Experimental evaluation

In this section, we describe the experimental evaluation of our framework's capabilities in the context of changing-contact robot manipulation tasks. We first describe results of incrementally updating and using forward models for control in continuously and smoothly varying environments (Section 4.1). We then show the advantage of our adaptive variable impedance (base) controller in such continuous-contact environments by comparing it with representative adaptive control strategies (Section 4.2). Next, we experimentally demonstrate our framework's ability to identify and model different contact modes in tasks where the interaction dynamics changes discretely (Section 4.3). In Section 4.4, we experimentally justify key choices made in the formulation of our framework for handling piecewise continuous dynamics, by comparing with a sophisticated, long-term prediction algorithm. Then, in Section 4.5, we evaluate the overall framework's ability to produce smooth motion in the presence of collisions and impact-less mode transitions. We experimentally evaluated four hypotheses:

H1: Incrementally updated forward models reliably and efficiently capture the smoothly changing interaction dynamics in a contact mode, and enable our base controller to provide better performance than traditional adaptive control strategies;

H2: Our base controller with the contact mode recognition strategy enables the robot to detect and model contact modes, and to reliably follow the target motion pattern under piecewise-continuous dynamics of changing-contact manipulation tasks;

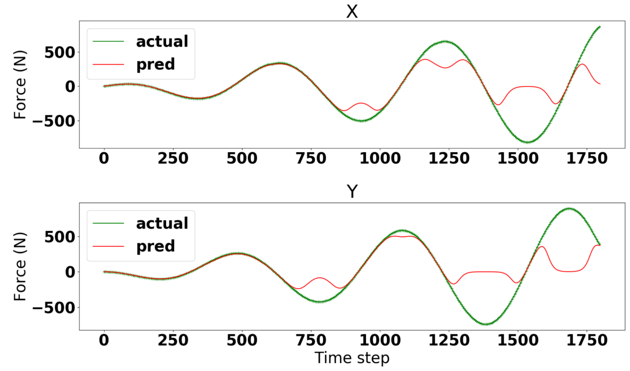
H3: Incremental updates of the predictive models and adaptive control strategies are required to effectively model and account for the piecewise-continuous interaction dynamics of changing-contact manipulation tasks; and

H4: Our framework's contact anticipation module incrementally and accurately predicts the contact locations, and our transition-phase controller and transition strategy help reduce discontinuities during mode changes in changing-contact manipulation tasks.

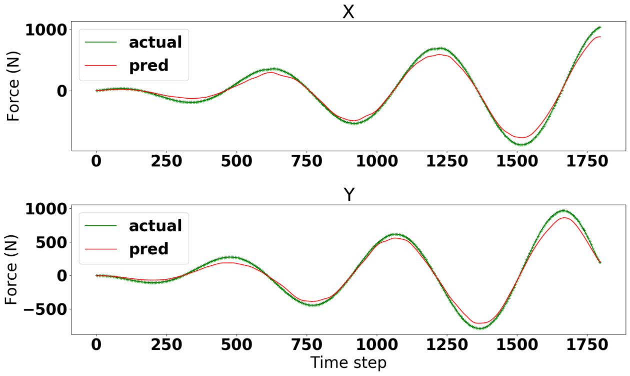
As stated earlier, we use simulation environments and a physical robot in our experiments. Experimental comparison with baselines is mostly performed in 2D and 3D simulation environments that support multiple trials and an in-depth analysis of the interaction dynamics under controlled settings. We report results of experimentally evaluating our framework on a seven degrees of freedom (DoF) Franka Emika robot manipulator arm; we also use a model of this robot in our 3D simulation experiments. Additional details of the simulation environments and experiments are provided in the appropriate sections below.

It is challenging to choose the baselines and environments for experimental evaluation because none of the existing frameworks support all the capabilities of our framework. For example, purely analytical control methods may not support the desired adaptation and incremental learning capabilities (Pezzato et al., 2020; Yang et al., 2011). State of the art data-driven methods (in the joint space or task space) may provide good performance in situations similar to those in the training set, but will not support runtime adaptation to previously unseen situations (Khader et al., 2020). We thus chose some well-established methods and environments that support some subset of the desired capabilities, and enable us to evaluate these capabilities and/or demonstrate the need for these capabilities.

In the simulation environments and on the physical robot, we considered changing-contact manipulation tasks with a target motion pattern—see Figure 1. These tasks involve impact-less mode changes, impact-based mode transitions, or both. The performance measures in these tasks include accuracy of (or the error in) following the target motion pattern, recognizing mode changes, estimating location of contact changes, and/or achieving the desired value of relevant parameters (e.g., force on impact). We also consider the task completion time and qualitative measures of the smoothness of interaction dynamics.



(a) Predictions from fixed GMM forward model.



(b) Predictions from incremental GMM model.

Figure 4: Comparison of fixed and incrementally revised GMM models in for the "porridge stirring" task in a 2D simulated environment; *red:* predicted sensor values; *green:* actual measurements. Incrementally revised forward models provide better prediction ability.

4.1. Continuous-contact manipulation

Our prior work provided a proof of concept demonstration of smooth control of a seven DoF robot arm for continuous-contact manipulation tasks (Mathew et al., 2019). We also compared the tracking performance of our controller with a baseline fixed-stiffness position-tracking controller to demonstrate the need for variable impedance when interacting in a continuously changing dynamic environment.

To further demonstrate the importance (and benefits) of incrementally adapting the forward model in response to continuous changes in dynamics (**H1**), we compared our incrementally-revised forward model with a fixed GMM model pretrained using the same data. We used a custom-built 2D simulated environment based on the Box2D physics engine (Catto, 2017) with noiseless measurements—see Figure 1c.

We simulated a porridge-stirring task in simulation with the viscosity increasing as the robot moves. Specifically, the robot was provided a target motion pattern of moving in a circular path in an environment in which (unknown to the robot) the viscosity increases continuously from 0 Ns/m^2 to 80 Ns/m^2 in steps of 0.1 Ns/m^2 every simulation step (the total circular trajectory had 600 steps). The sensor measurements collected during the initial phase were used to build a fixed GMM model for one-step prediction of end-effector forces based on previous end-effector velocity and force. This model and a high-stiffness controller served as the baseline for comparison with our incrementally updated GMM model and the variable impedance controller. The ability of the trained models to predict the sensor measurements were then evaluated in an environment in which the viscosity increases from $0 - 150$ Ns/m^2, and results from one such trial are summarized in Figure 4a.

We observed that the fixed and incrementally-updated models are able to predict the forces accurately when the measurements are similar to the training data. However, when the sensor measurements vary from those observed during training, the prediction accuracy is poor with the fixed GMM model but it continues to be good with the incrementally-revised GMM models. Even when the predictions are accurate with the fixed GMM model, it is largely due to the use of a constant high-stiffness controller. Although we do not show it here, we did verify that when the fixed GMM model is used to guide our variable impedance controller (in Equation 4b), it causes unreliable trajectory tracking. In addition, for the pretrained forward model to be useful, the training data will need to include samples from different regions of the state space; this imposes additional training time and memory requirements. These results indicate the need for using an incrementally revised forward model to account for continuously-changing interaction dynamics within a contact mode, i.e., these results indicate support hypothesis **H1**.

4.2. Comparison with adaptive control methods

To further test **H1**, we compared our base controller, i.e., the adaptive variable impedance controller (AVIC) of our framework, with one representative method from each of three classes of adaptive control methods in the presence of continuously and smoothly varying interaction dynamics.

4.2.1. Comparison with MRAC:

The first baseline we considered was an established MRAC method (Tarokh, 1991). Recall from Section 2.2 that MRAC methods attempt to adapt their control law based on a predefined reference model. In the chosen MRAC method, the reference model uses a second-order spring-damper system for each dimension i, parameterized with user-defined values for desired natural frequency ω_i and damping ratio ζ_i. The control law continuously adapts its parameters to minimize tracking errors.

MRAC was tested in a "multi-spring" simulated environment in which the robot had to move along a circular trajectory while the end-effector is being pulled by three springs of different stiffness—see Figure 1c. Tuning the large number of hyperparameters was found to be quite cumbersome for continuously varying environments, and performance was sensitive to the choice of parameter values. For example, in Figure 5, trajectory tracking is affected by the choice of the frequency parameter ω of the reference model, with stability affected adversely when the value of ω is changed from 1.5 to 1. In addition, the selection of hyperparameters proved to be challenging in the "porridge stirring" environment used in Section 4.1 and performance comparable to our framework could not be achieved using the MRAC method.

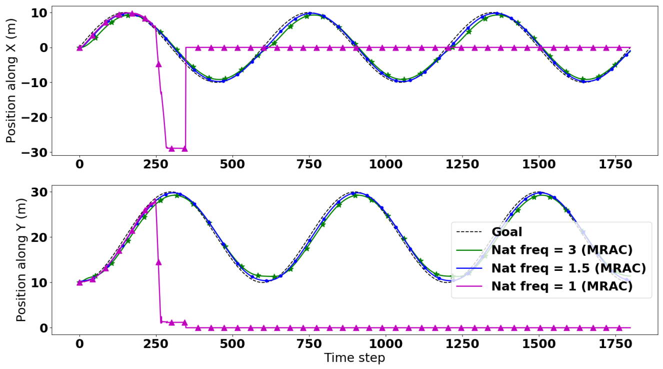


Figure 5: Trajectory tracking performance of MRAC method is highly dependent on the design of the reference model.

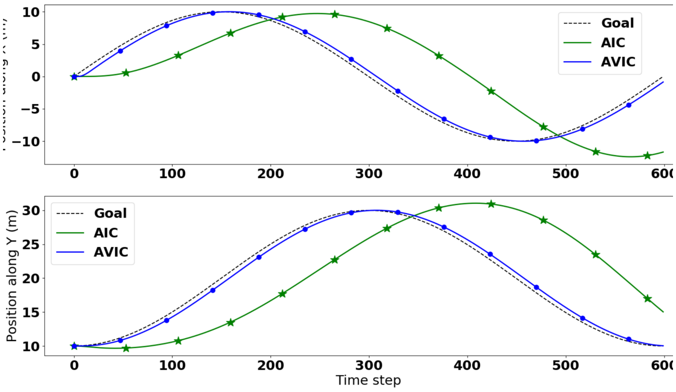
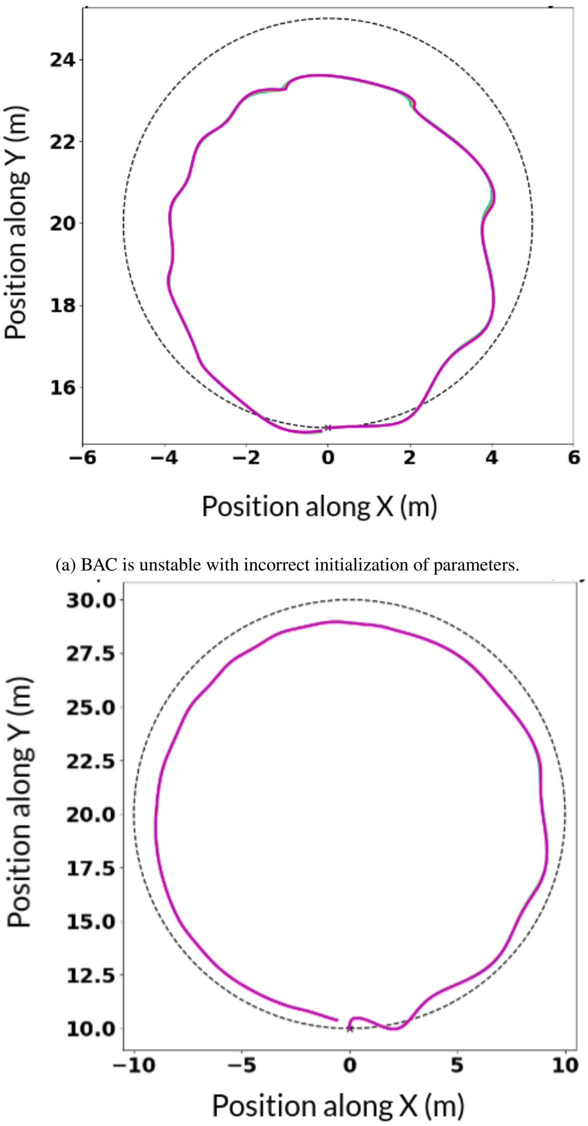


Figure 6: Our controller provides significantly better trajectory tracking compared with the AIC controller.



(a) BAC is unstable with incorrect initialization of parameters.

(b) BAC parameters struggle to converge in 30 trials in "multi spring" environment.

Figure 7: Trajectory following performance of BAC controller in a particular trial of two different experiments; dotted line is the target trajectory and solid line is the observed path of the robot: (a) performance is sensitive to parameter initialization; and (b) parameter values do not converge in a limited number of trials in more complex environments.

4.2.2. *Comparison with a self-tuning regulator:*

As an example of a self-tuning controller, we chose the Active Inference Control (AIC) method (Pezzato et al., 2020). This adaptive control strategy supports continuous updates in the presence of changing environmental dynamics; the control parameters are iteratively updated to account for unaccounted energy in order to reach a fixed target. Given time, AIC converges to a fixed target smoothly and without any appearance of instability for any dynamically changing environment. However, this is not useful when the robot has to follow a motion trajectory as the controller's parameters will typically not converge before the target changes to the next point in the trajectory. As a result, the robot always lags behind the target with AIC; Figure 6 shows the trajectory following performance observed in a particular experimental trial. We also observed that unlike AIC, our base controller compensated for external disturbances by directly predicting and canceling the forces, thus minimizing the lag in trajectory tracking.

4.2.3. *Comparison with a gain-scheduling controller:*

We implemented Biomimetic Adaptive Control (BAC), an RL-based gain-scheduling approach that iteratively updates control parameters (gains) at each point in a repeating trajectory (Yang et al., 2011). Given data from multiple trials of a trajectory, parameters at each time step are updated using values at the same time step from previous iterations such that a cost function based on tracking error is minimized.

In our experimental trials, we observed that bad parameter initialization for BAC caused instabilities and tracking irregularities that accumulated as the trajectory progressed. This is because BAC does not account for the temporal relationship between controller parameters within a trial. One example of such a trial with incorrect parameter initialization is shown in Figure 7a. Due to this sensitivity to initialization, it is difficult to ensure smooth trajectory tracking when learning a time-indexed parameter profile.

We also observed that although BAC can provide good trajectory tracking for simple environments given good parameter initialization and a sufficient number of trials (Yang et al., 2011), the parameters often do not converge in a limited number of trials in more complex environments. For example, Figure 7b shows the trajectory following results of a particular trial in the "multi-spring" environment. BAC and other such gain-scheduling controllers also rely on the duration of the task (i.e., number of time steps in each trial) to be fixed across trials.

Results from the comparison with the three classes of adaptive controllers show that the incremental learning (of the predictive forward models) and the online adaptation of the feedback gains enables our controller to perform better in the presence of continuous and smoothly changing interaction dynamics, which further supports hypothesis **H1**.

4.3. Evaluation for piecewise continuous interaction dynamics

Next, we evaluated the ability of our base controller to accurately detect contact modes, and to acquire or revise the predictive models of these modes for reliable trajectory following in the presence of piecewise-continuous dynamics corresponding to discrete contact mode changes (**H2**). *We used a 7-DoF physical robot for these experiments*. We have demonstrated in prior work that a single model is insufficient for accurate trajectory tracking in the presence of such dynamics, motivating the need for piecewise-continuous models (Sidhik et al., 2020). In this paper, we report results of comparing our controller with a position- and force-tracking controller to demonstrate the need for having separate controllers depending on the interaction dynamics.

We used a "changing surface" task for the evaluation. The robot had to slide an object attached to its end-effector along a desired trajectory, and it experienced three previously unseen surfaces with different values of friction—see Figure 8. As described in Sections 3.1 and 3.2, we expected the robot to identify a transition to each new mode (i.e., each surface) and incrementally build a predictive (forward) model for the mode while operating under high stiffness. Once the dynamics models for a mode had been built, we expected the robot to respond to subsequent transitions to this mode by using and revising the corresponding dynamics model. The abstract feature representation used for distinguishing between the contact modes was $\frac{\|F^t_{ee}\|}{R^t}$, as mentioned in Section 3.2, which represents surfaces as a distribution of their friction coefficients.

Figure 9 shows the robot's ability to detect mode changes in one trial of this experiment. The robot was able to identify transitions to existing or new modes with high confidence. In each instance, the second best choice of mode was associated with a much lower value of confidence. The results also show that our hybrid framework and feature representation make performance robust to changes in the direction of motion, i.e., a new mode is not identified when the manipulator moves over a previously seen surface in a new direction. There was some similarity in the confidence values for surfaces 2 and 3 (S2 and S3 in Figure 9) because of the similarity in their friction values. Although it should be possible to quickly detect a previously seen mode, the clustering-based approach for mode detection can make the identification of new (or previously seen) modes depend on the number of modes for which the robot has already learned models (Sidhik et al., 2020).

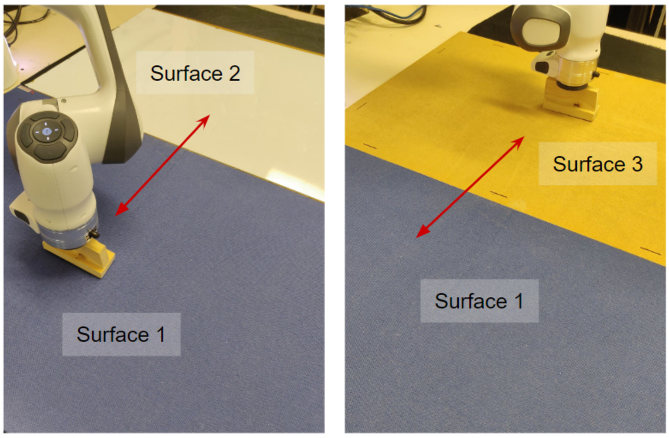


Figure 8: Robot manipulator sliding an object attached to its end-effector in a given motion pattern along three surfaces characterized by different values of friction.

Figure 10 shows the trajectory tracking error and the values of the stiffness parameters of the controller during the trial. The peaks in the trajectory error plot correspond to a sudden change of surface. During each such instance, the predictions made by the dynamics model of the previous mode caused a momentary error in the trajectory tracking ability, until the robot switched to using a high stiffness controller and identified the current mode; the robot then used suitable low(er) stiffness to complete the task. It can also be seen that switching to a previously seen mode requires a much shorter period of high stiffness compared with building a new dynamics model. These results show that the framework is able to identify and learn the predictive models for different contact modes, supporting (**H2**). In other work, we have demonstrated the ability to identify and learn the predictive models for the modes corresponding to a robot sliding objects using different types of contacts (e.g. edge contact, surface contact); the robot's performance was observed to be robust to changes in the direction of motion and the applied normal force (Sidhik et al., 2020).

4.4. Incremental learning and adaptation

Next, we explored the need for incrementally updating the predictive models for the discrete modes, and the need for adaptive control strategies, to handle the piece-wise continuous dynamics of changing-contact manipulation tasks (**H3**). As the baseline for comparison, we chose a state of the art framework that performs offline long-term prediction of dynamics (Khader et al., 2020). It identifies the task's dynamic modes from a training dataset of demonstrations of the desired

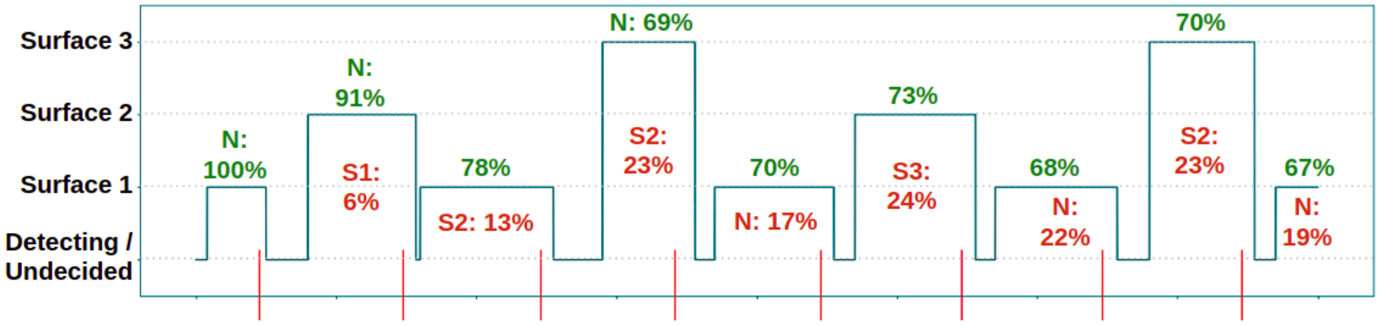
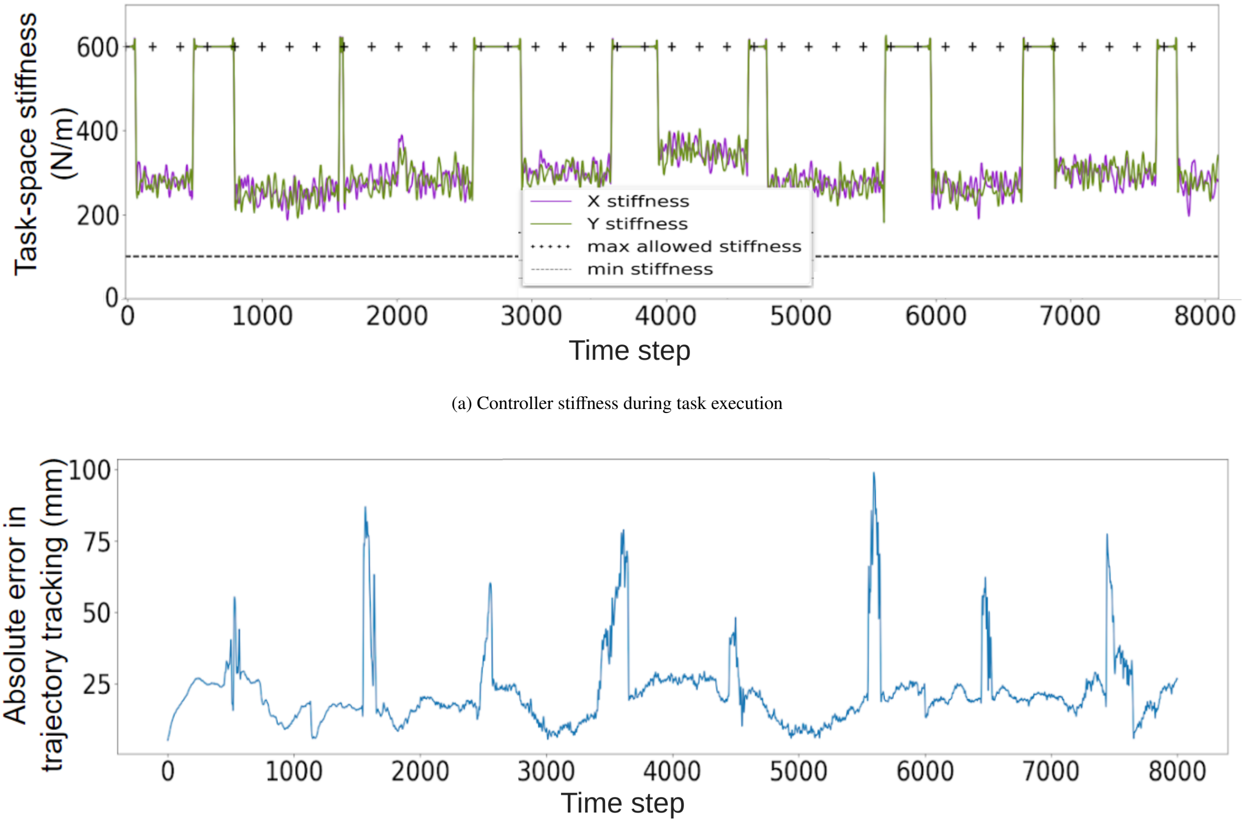


Figure 9: Modes detected and their confidence values; red vertical lines on the x-axis indicate actual mode transitions. The number on top of a peak (in green) indicates the confidence with which the transition was identified; the number below a peak (in red) corresponds to the mode with the next highest confidence. "N" indicates a transition to a new mode.



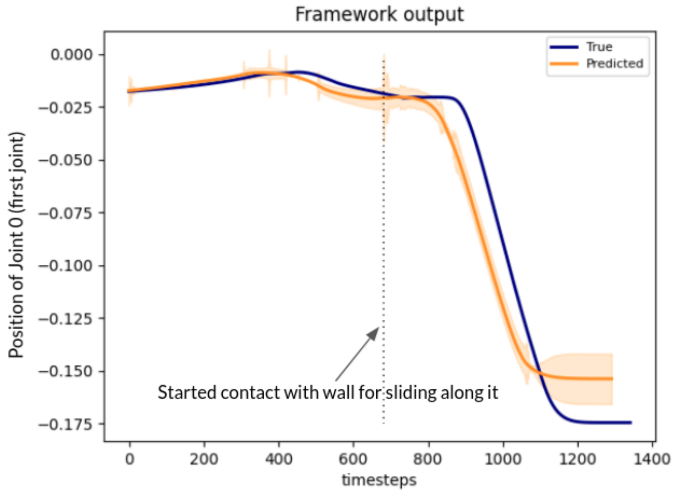
(a) Controller stiffness during task execution

(b) Absolute error in trajectory tracking; the spikes correspond to a temporary, incorrect feed-forward prediction by the previous model.

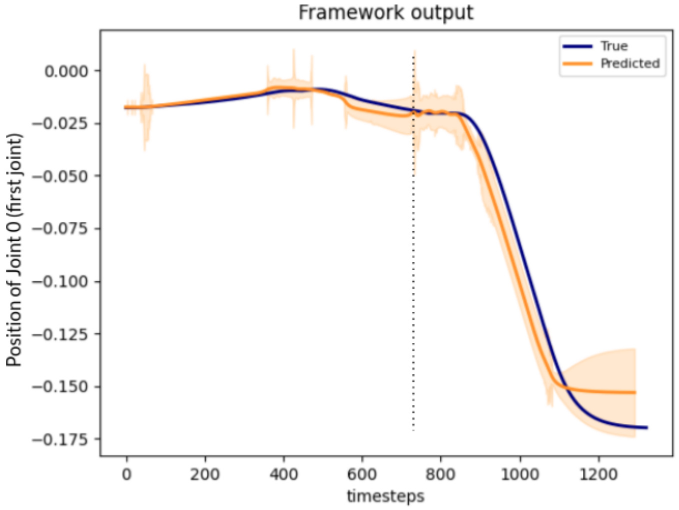
Figure 10: Performance of our base controller in the changing-surface task.

motion, uses multi-class Support Vector Machines to predict mode changes, builds separate Gaussian Process (GP) dynamics model for each mode, and provides a probabilistic algorithm for multi-step prediction of joint-space state variables for changing-contact manipulation tasks. It was proposed as a data-efficient way of predicting long-term state evolution of joint-space dynamics for tasks involving discontinuous dynamics; although it was *not* designed as a solution for real-time control in such tasks, it presents an interesting option for exploring **H3**.

We simulated the 7-DoF physical robots in the PyBullet 3D simulation environment (Coumans and Bai, 2016–2021) and setup a task similar to the one used with the baseline framework. As shown in Figure 1b, the robot (with a block fixed to the end-effector) approaches a table to make contact with it, slides the block along the table until it collides with a wall, and slides on the table along the wall. We mapped **H3** to two sub-claims that we then evaluated: (i) unlike our framework, the baseline is unreliable for tasks with discontinuous dynamics unless the robot uses high-stiffness control or is trained and tested in the same environment; and (ii) real-time update of the



(a) System trained with wall friction 0.6, tested with friction 0.9.



(b) System trained with wall friction 0.1-0.7, tested with friction 0.9.

Figure 11: Baseline framework's prediction of one joint's position; dotted vertical line indicates end-effector's contact with the wall. Predictions are reliable only when used with a high stiffness controller.

predictive models provides more reliable performance than the baseline framework's offline, long-term prediction of dynamics in the presence of unexpected environmental changes.

4.4.1. Long-term prediction of joint dynamics:

We first trained the baseline system with a dataset of trials in which the robot used a fixed medium-stiffness controller and the friction of the wall (large red block in Figure 1b) was fixed to 0.6. Testing was then done with a higher friction value (0.9) and the same controller. Figure 11a shows the prediction and actual values of the positions of one of the joints involved in the task (the first joint). Since the offline framework was not aware of the change in friction, it predicted the joints to move as freely as it did during training. The true joint positions were, however, affected by the higher surface friction of the wall resulting in the "true" values of the joint's position lagging behind the "predicted" values.

For further exploration, the training set was modified to include wall friction values between 0.1 – 0.7 in increments of 0.1; the system was then tested on a wall with friction 0.9. The predictions were similar to the those in the previous set of experiments, but with a wider band of uncertainty around the predictions due to the larger variability in the training data—see Figure 11b. In addition, when a high stiffness controller is used for training and testing, the robot follows the trajectory quite closely regardless of friction. This results in lower variability across trials, as indicated by the narrower uncertainty band around the prediction and more accurate predictions—see Figure 12. These results indicate that learning dynamics in the space of joint positions and velocities poses a hard high-dimensional problem. Also, the learned forward models do not capture the interaction dynamics accurately, and predictions are reliable only when used with a high stiffness controller.

The main reason for the baseline framework's poorer perfor-

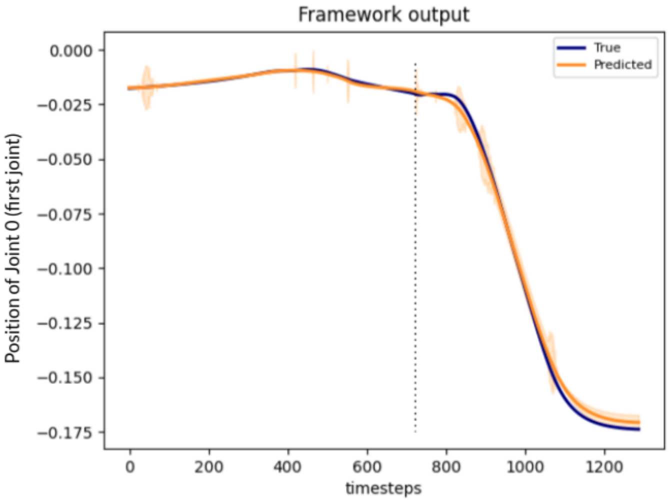


Figure 12: Prediction of the position of one joint produced by the baseline framework. System trained and tested using a constant high-stiffness controller (friction values as in Figure 11b).

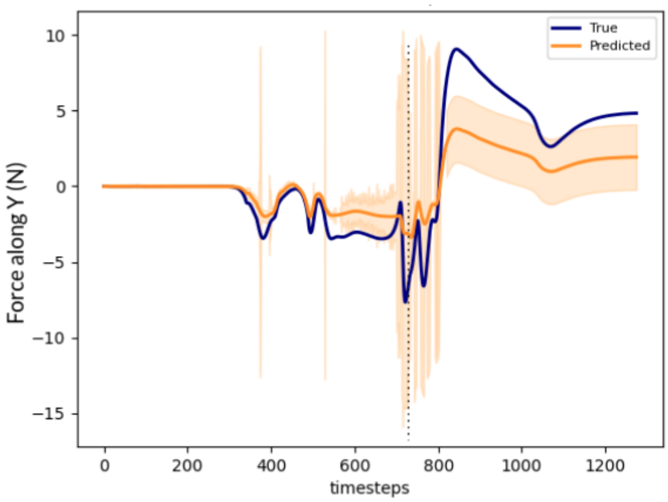


Figure 13: Prediction of end-effector force along Y-axis (along the wall). System trained with wall friction values between 0.1 – 0.7, tested on wall with friction 0.9.

mance with the previously unseen value of surface friction is that it is unaware of the environment's dynamics before it performs the prediction. This is not necessarily a flaw of the framework as it was not designed for such use cases. However, our experiments indicate that for accurate long-term prediction of joint-space dynamics, either the environment has to remain unchanged and/or the robot has to use a high-stiffness controller; the former requirement is not realistic and the latter is not desirable. For environments whose dynamics can change after training, the system will at least have to be trained with many examples of different environments. Our framework addresses these issues by learning and revising a model of the dynamics in the task space (instead of joint space) and accessing a more direct estimate of the interaction dynamics in the form of end-effector forces and torques.

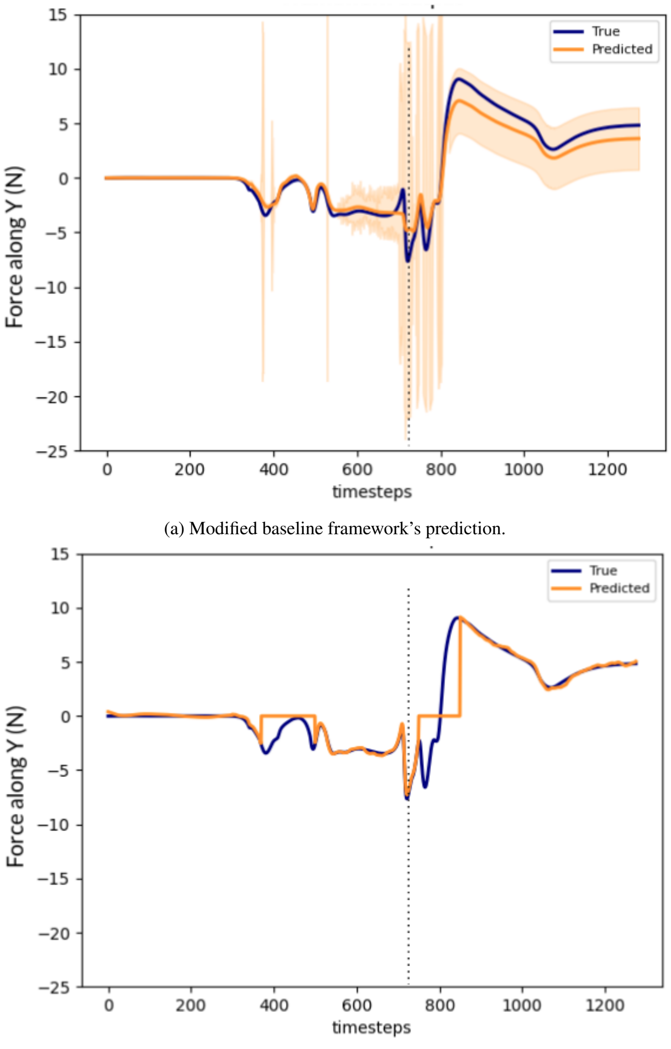
4.4.2. Long-term prediction in the task-space:

Next, the baseline framework was modified to consider the interaction dynamics in the task space by reformulating the framework's GPs (that model dynamics of each mode) to model $p(F_{ee_{t+1}}|F_{ee_t}, \dot{x}_t])$, similar to the forward model of our base controller (Section 3.1).

The system was again trained with examples of the evolution of the state when the surface friction was between 0.1 and 0.7; the trained models were compared with the measurements obtained when the surface friction was 0.9. Figure 13 shows the measured and predicted force measurements along the Y-axis, the direction along which the wall's friction acts. The spikes in Figure 13 at ≈ 400 along the X-axis corresponded to when the robot made contact with the table, and the spikes between 700 − 800 corresponded to when the robot collided with the wall. We observed that the actual values of friction force after contact with the wall did not fall within the predicted band of uncertainty. Also, the effect of different levels of friction was more observable in the task space, indicating that it is more meaningful to represent the interaction dynamics for changing-contact tasks in the task space. This experiment suggests that interaction dynamics cannot be predicted reliably without feedback during task execution.

4.4.3. One-step task-space prediction with real-time feedback:

In the next experiment, real-time measurements from the robot ($[F_{ee_t}, \dot{x}_t]$) were provided as feedback to the baseline framework during testing. At each timestep, the system then only had to predict sensor values for the next timestep using the dynamics model for the identified mode. Although this strategy produced better results than before by extrapolating from the learned model to handle previously unseen values—see Figure 14a—the predictions are still not accurate. This is because the baseline framework's model is not incrementally updated as the environment's dynamics changes. Our hybrid framework, on the other hand, supports incremental, real-time updates to the dynamics models. This resulted in more reliable predictions for the same experiment, as shown in Figure 14b, leading to more accurate trajectory tracking in the presence of discretely changing dynamics in previously unseen environments.



(a) Modified baseline framework's prediction.

(b) Predictions from the forward model of our base controller.

Figure 14: Prediction of end-effector force along Y-axis (along the wall). Each system was trained with wall friction between 0.1 − 0.7, and tested with friction 0.9. Dotted vertical line indicates when the robot made contact with the wall (obstacle).

Note that the training of the baseline framework required significant time and computational resources depending on the density and the length of the data in the training set. The implementation of the GP in this framework took ≈ 55 minutes training time on an 8-core, 16GB RAM computer without using a GPU, for a dataset consisting of 40 trials of the task mentioned above; the long-term prediction process during testing takes ≈ 20 minutes.

Overall, the experiments discussed above demonstrate that the incremental, real-time revision of the predictive, forward (dynamics) models is critical for changing-contact manipulation tasks. Also, performance improves significantly when the learning is done in the task space and suitable feedback of the system state is available and used. The results of these experiments support hypothesis **H3**. The observed results are also influenced by the choice of the state space and feature representation used for control and learning. Our framework support

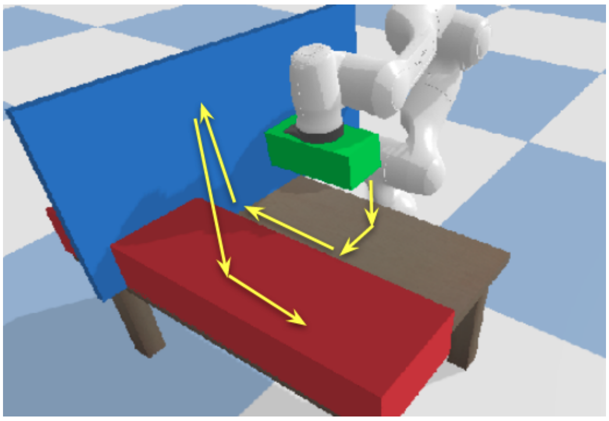


Figure 15: A simulated sliding task that involves making contacts with different surfaces in the workspace.

such incremental learning and adaptive control based on a suitable representation; it is thus able to provide smooth control of changing-contact robot manipulation tasks.

4.5. Evaluating overall framework

We then evaluated our framework's ability to accurately anticipate mode transitions and smoothly switch to the appropriate transition phase controllers in the anticipated regions of contact (**H4**). Specifically, we explored the ability to anticipate and handle discontinuities in target motion patterns with multiple collisions (Section 4.5.1), and those with impact-less mode transitions (Section 4.5.2).

4.5.1. Multiple collisions:

The motion profiles (e.g., velocity, acceleration profiles) of a changing-contact manipulation task are expected to have large spikes when discontinuities arising from collisions are not handled. Since these spikes can damage a physical robot, we first explored the contact anticipation and handling ability in simulation. Figure 15 shows the simulation environment and the task. The robot had to move in free space toward the table and make contact with it, slide along the table's surface (i.e., force control in one direction) until it came in contact with a wall (i.e., side of red object in Figure 15), slide along this wall (i.e., force control in two directions) to make contact with second wall (i.e., blue object in Figure 15), slide up this wall (force control along one direction) until end of wall is reached, break contact with wall and move through free space until contact is made with top of red surface, and slide along the red surface until goal location is reached. The baseline used for comparison had the simulated manipulator attempt to follow the target motion pattern without anticipating contacts and handling discontinuities.

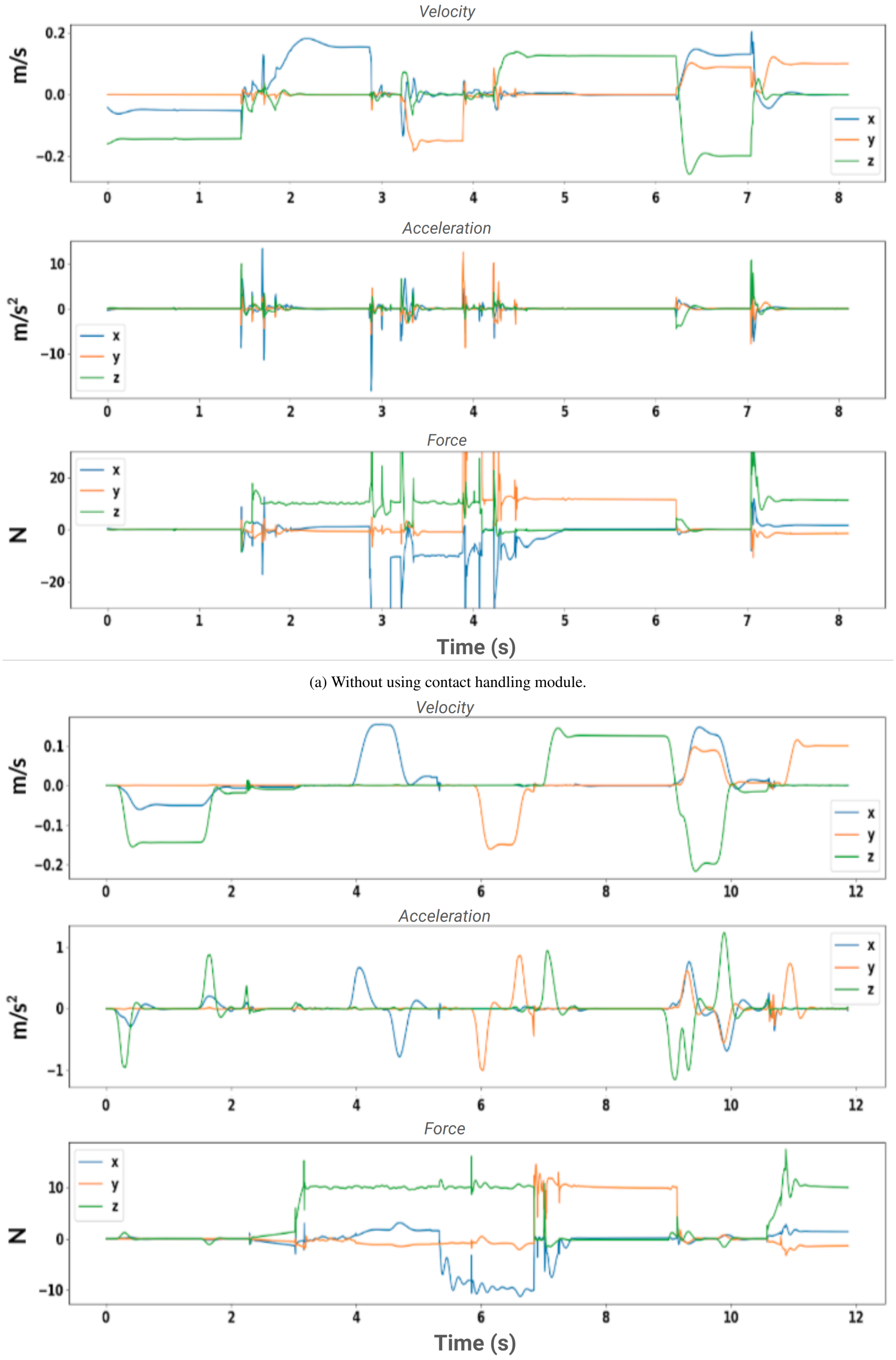
As shown in Figure 16a, in the absence of the module anticipating and handling contact changes, the robot experienced high impact forces and discontinuities in the dynamics. There were thus large sharp peaks in acceleration (i.e., high jerk values) during the motion pattern, e.g., the accelerations were as high as 18 m/s^2 and the peak impact forces were well above 50N; this could damage a real robot. When the same task was performed using our framework, noisy initial estimates of the contacts were provided to mimic values from a noisy vision sensor; recall that the framework can identify contact mode changes even without these estimates. Using our framework, the robot was able to complete the task and provide significantly smoother motion (than the baseline) even in the first trial. The velocity and acceleration curves were smoother, even when there were multiple contacts. Also, the peak forces were significantly lower, with the highest impact force being less than ≤ 8N (a 84% reduction from the baseline), and the peak acceleration on impact was ≤ 1m/s^2, due to the lower velocity in the transition phase. However, the time to complete the task increased by about 3.7 seconds due to the modified velocity profiles of the plan. The framework thus traded off task completion time for smoothness of motion. We hypothesized that this delay could be reduced as the robot incrementally revised its estimate of the location of contacts, and explore this hypothesis further on a physical robot.

Prediction Error (m)	Initial	Final (trial 5)
Contact 1 (Z-axis)	0.12 ± 0.3	0.016 ± 0.039
Contact 2 (Y-axis)	0.09 ± 0.2	0.011 ± 0.04
Contact 3 (X-axis)	0.1 ± 0.2	0.018 ± 0.036

Table 1: Error in the estimated contact location along the most significant axis (in parenthesis) in the first and fifth trials of the task in Figure 17. Value along the diagonal of the covariance matrix is shown as the standard deviation (± term).

As shown in Figure 17, in the robot experiments, the 7-DoF physical robot manipulator with a wooden block attached to its end-effector was asked to move vertically down to the table (contact 1), slide along y-axis (the table's surface) to a wall (contact 2), and slide along the wall (while in contact with the table's surface) to another obstacle (contact 3). The robot was provided noisy (i.e., incorrect) initial estimates of the contact positions (see Table 1). Over a few iterations of this task, the robot had to improve its estimate of the contact positions and reduce the deviation for the the given motion pattern. The robot also had to modify its approach velocity from the initial value of 0.05 m/s to achieve a desired impact force of 8 N. Since each contact in the target motion pattern corresponds to a different kind of environment dynamics (e.g., motion in free space, motion against surface friction), the velocity required to attain the desired impact force was expected to be different. The robot also had to incrementally revise its approach velocity for each contact until the desired velocity for that mode was achieved. Furthermore, the robot had to minimize any spikes in the velocity or acceleration profiles.

Figure 18a shows the velocity, acceleration, and end-effector force in the first trial, and Figure 18b shows these values after five trials. The results indicated a reduction in the spikes in the motion profiles. The reduction was not as pronounced as in the simulation environment (Figure 16a) because we set a low approach velocity to prevent damage to the physical robot in the initial trials, i.e., when it did not have a good estimation of the location of the collisions. However, the results in Figure 18a,



(a) Without using contact handling module.

(b) When using our contact handling module.

Figure 16: End-effector velocity, acceleration, and force while performing the changing-contact manipulation task in Figure 15: (a) without using the module that handles contact changes; (b) using our framework that includes the contact handling module. Our framework significantly reduces the discontinuities (i.e., spikes in velocity profiles).

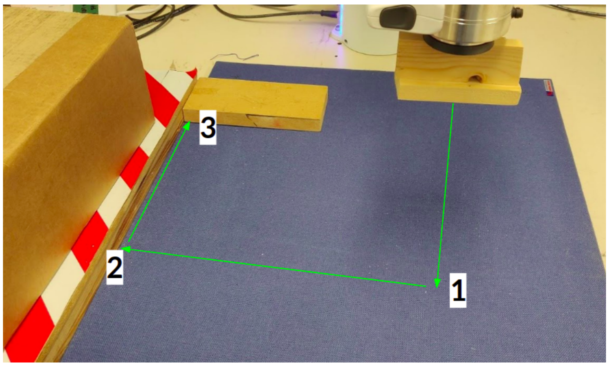


Figure 17: A changing-contact manipulation task that involves the block in the robot's end-effector making contact with the table's surface at "1", a wall at "2", and another object at "3".

Figure 18b, and Table 1 indicated a significant reduction in the uncertainty in the estimate of the contact positions, i.e., in the size of the covariance ellipsoids, and the robot spent significantly less time using the transition-phase controller and the associated lower velocity. The last plot in Figure 18a and Figure 18b show the activation of our framework without and with the contact anticipation module and related adaptation. The motion pattern was completed in 14.4 s in the first trial and in 9.2 s in the fifth trial. These results support **H4**.

The covariance ellipsoids converged in the first three trials of the task, but the task was repeated to evaluate the ability to compute and set the approach velocity for different transition-phase controllers. The robot converged to a suitable approach velocity for the first contact (from motion in free space) in five iterations. It was, however, difficult for the robot to adjust its approach velocities for contacts 2 and 3, which required the robot to use force control along one and two directions (respectively). Contact 3 was particularly challenging because it involved sliding along two different surfaces, with the different frictional resistance offered by the two surfaces resulting in very noisy readings from the force-torque sensor. Since the impact force was along the same direction as friction, it was more difficult to isolate the impact force from the force due to surface friction, which made updating the approach velocity more challenging.

Although the results documented in this subsection related to the experiments conducted on a physical robot have already been published in our previous work (Sidhik et al., 2021), the extensive experiments conducted with a simulated robot in a suitable environment are novel; they demonstrated our framework's ability to anticipate collisions and thereby reduce impact forces by trading off between accuracy and task completion time.

4.5.2. *Impact-less transitions:*

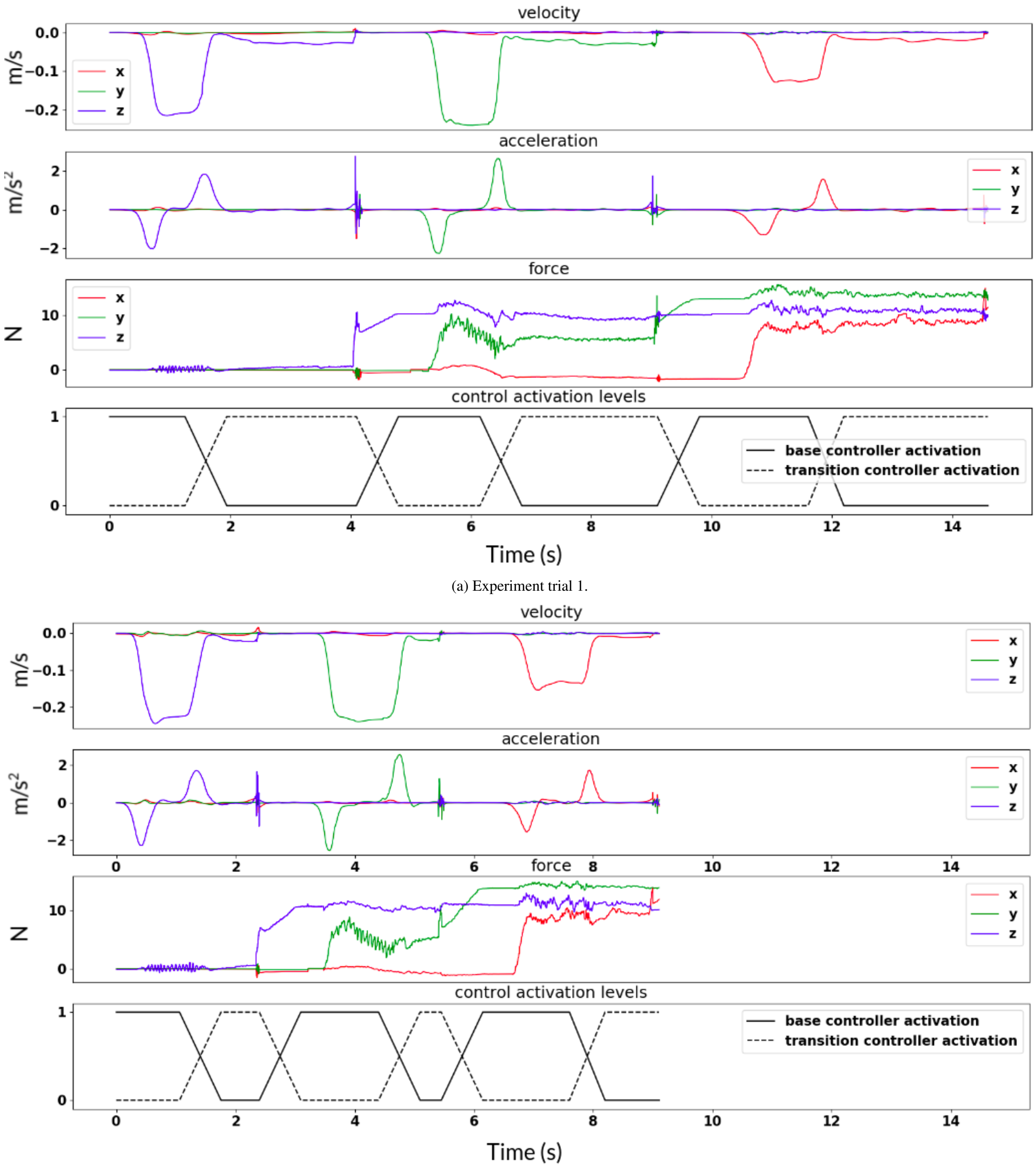
In this experiment, the objective was to test the effectiveness of the framework in accurately predicting the location of impact-less contact changes, i.e., mode changes that are not due to collisions, and handling the associated discontinuities.

The experiment involved the physical robot following a motion pattern similar to that previous experiment (in Section 4.5.1 above). Specifically, the robot had to approach a surface (surface A) from above, slide along the surface to make contact with a wall, slide along the wall (on the table surface), and then slide on the table's surface until it comes in contact with another object—see Figure 1a. The difference was that the surface along which the robot had to slide changed suddenly along segments '2' and '4', i.e., the robot experienced a surface with a different friction value. The switch along segment '2' was to a surface with a higher friction, and that along segment '4' was (back) to the surface with a lower friction. As in the previous experiment, the robot was given a noisy initial estimate of the position of the impact-based contact change, but it was unaware of the existence or location of the surface switches along segments '2' and '4' in Figure 1a; it had to anticipate them in subsequent trials once it realized that it had experienced previously unknown contact changes in the first trial of the target motion pattern. The desired behavior from the robot was that it learned to anticipate collisions as well as impact-less contact changes, and smoothly switches to an appropriate transition-phase controller to reduce any associated discontinuities in the motion dynamics.

In this experiment, we focus on the performance of the framework along the two segments in which the surfaces changes, i.e., the previously unknown impact-less mode transitions, occur. Figure 19 shows the velocity, acceleration, and end-effector force profiles in these regions measured during the first trial of the task. As expected, the robot experienced sudden spikes in velocity and acceleration (jerk) when the surface changed unexpectedly. Recall that this is due to the wrong predictions provided by the dynamics model which provides a feed-forward term for the controller that either overestimates or underestimates the environment forces in the new mode. When the robot detected such a discontinuity, it identified this as a contact change and quickly switched to a high-stiffness controller to verify whether a mode change had actually occurred. Once a transition to a new mode was confirmed, the robot learned a new dynamics model for the mode which it used with the base controller while in that mode. Although the surface changed again along segment '4', the robot recognized this as a mode that it had experienced before and transitioned (with the intermediate use of a transition-phase controller) to using the model and controller of the mode.

In the second trial of this task, the robot expects these mode switches; the shaded regions in Figure 20a represent the anticipated region of impact-less contact change. In this region, the robot switches to a high stiffness controller to identify the new mode quickly while also reducing sudden velocity-acceleration spikes during motion. The size of the anticipated region of contact change reduced in the subsequent trial, as shown in Figure 20b, which further reduced the time the robot spends using a transition-phase controller. These results further support hypothesis **H4**.

We made an interesting observation during this experiment: the robot was able to detect the change from the surface with lower friction to the surface with higher friction (along segment '2') much more easily than the transition from higher friction to lower friction. This difference can be attributes to the fact



(a) Experiment trial 1.

(b) Experiment trial 5.

Figure 18: Velocity, acceleration, force, and controller activation levels in: (a) experimental trial 1; and (b) experimental trial 5; for a task involving multiple collisions. Use of our framework reduces uncertainty in estimates of contact positions, reduces the time spent using the transition-phase controller, and reduces discontinuities in the motion.

that during the transition from the surface with lower friction to that with higher friction, the increase in friction at the region of the block in contact with the new surface offers the highest resistance and contributes to the frictional resistance of the block. This increase is rather pronounced, as observed in Figure 19. During the transition along segment '4' from the surface with higher friction to that with lower friction, on the other hand, the trailing part of the block is in contact the surface with higher friction that still offers resistance. The robot thus experiences a more gradual change (reduction) in the frictional resistance, and the discontinuity is not that pronounced as with the mode change in the other segment. This observation motivated the

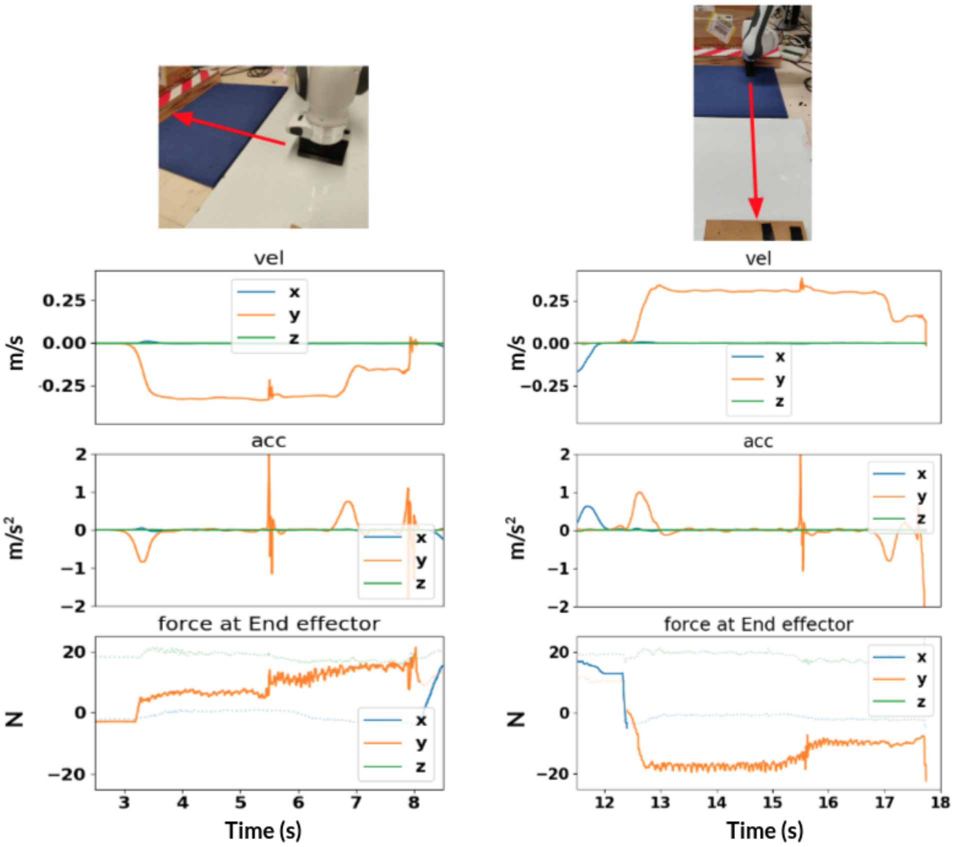


Figure 19: Selected sections of the velocity (top), acceleration (middle), and force (bottom) plots in the first trial of the task; the robot is unaware of the impact-less transitions in the task beforehand. (**Left**) plots for the first impact-less transition along segment '2' in Figure 1a; (**Right**) plots for the second impact-less transition along segment '4' in Figure 1a. (Note: X & Z axes in the force plot are not relevant in the motion during the chosen segment, and have therefore been faded out.)

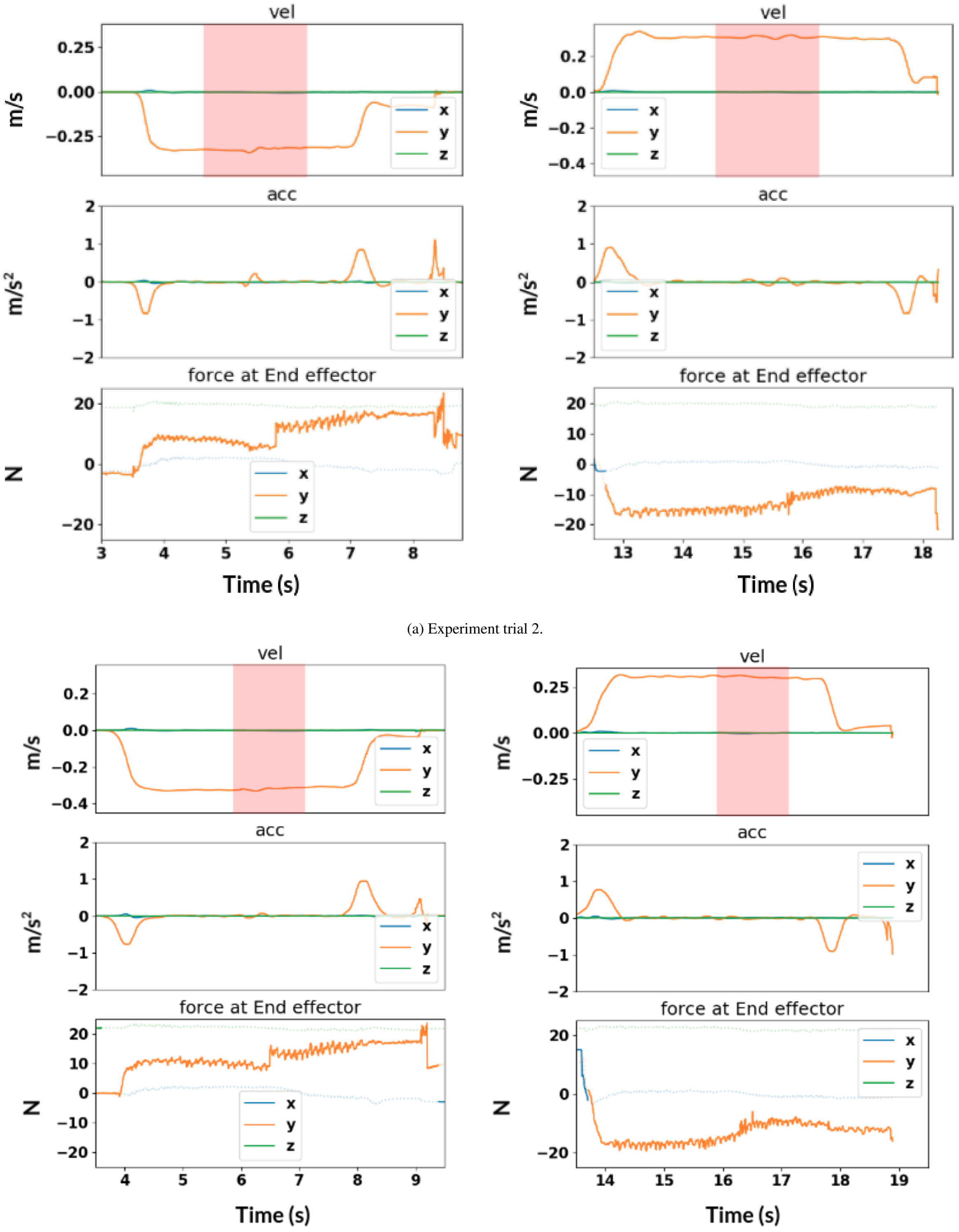
need for experimentally setting a threshold on the error between the end-effector measurements and the predictions from the forward model for that mode.

5. Conclusion

Many robot (and human) manipulation tasks are changing-contact manipulation tasks. They are characterized by piecewise continuous interaction dynamics, with discontinuities due to collisions and changes in surfaces, types of contact, and other factors, and continuous elsewhere. While it is possible to construct a hybrid framework with continuous dynamics within each of a set of discrete modes, it is difficult to provide comprehensive information about the modes in practical tasks. In addition, such a hybrid framework cannot, by itself, address the discontinuities that exist during the transition between the modes; these discontinuities can damage the robot or the environmental objects.

In this paper, we presented an adaptive framework for changing-contact manipulation tasks inspired by studies in human motor control. The framework has a mode detection module which can recognize existing modes and identify new contact modes based on an abstract feature representation. Each such mode is characterized by a suitable predictive/forward dynamics model, a control law, and a relevance condition. The forward model is learned and revised incrementally during run-time, with the error between the predicted and actual measurements (of end-effector forces and torques) being used to adapt the gain (i.e., stiffness and damping) parameters in the corresponding control law. Our framework also includes a contact handling module that uses a Kalman filter to incrementally improve estimates of contact positions. These contact position estimates help revise the velocity profile to minimize the time spent in a transition-phase controller, ensure smooth transition to and from this controller, and achieve a desired force on impact. Our representational choices enable us to simplify and address the associated challenges reliably and efficiently.

We showed the effectiveness of the overall framework in the context of a physical robot performing manipulation tasks that involve multiple contact changes, and in the context of 2D and 3D simulation environments in which a robot had to execute motion patterns involving making and breaking contacts. We experimentally demonstrated the need for adaptive control and incremental learning strategy in the presence of tasks and environments with a mix of continuous and discontinuous dynamics. We compared our framework's performance with a representative method from each of three different classes of adaptive control methods from the existing literature, and with a sophisticated framework for offline, long-term prediction in the context of discontinuous dynamics, to highlight the capabilities



(a) Experiment trial 2.

(b) Experiment trial 3.

Figure 20: Selected sections of the velocity (top), acceleration (middle), and force (bottom) plots in (a) experimental trial 2; and (b) experimental trial 3. In each of (a) and (b), (**left**) plots for the first impact-less transition along segment '2' in Figure 1a; and (**right**) plots for the second impact-less transition along segment '4' in Figure 1a. (Note: X & Z axes in the force plot are not relevant in the motion during the chosen segment, and have therefore been faded out.)

of our framework. Our experimental results support our claim that our framework enables smooth control of changing-contact robot manipulation tasks.

Our framework opens up many directions of further research. First, we only focused on collisions due to translational motion, and did not address collisions due to rotations of the end-effector. This could be addressed by defining a region of anticipated collision in $\mathcal{SO}(3)$. Second, we observed that updating the approach velocity for collisions when the robot is already in contact with another surface is more complicated. This is because of the difficulty in differentiating between the sensor measurements obtained due to reactive forces from the existing contact and the sensor measurements obtained due to the impact force generated by the collision with another object. One way to address this issue is to learn a better forward model (for each contact mode) that can accurately predict the forces due to the first contact. Third, we only modified the velocity profile to achieve the desired smooth motion, and future work will explore the relationship between stiffness values and the impact forces. Proof of experiments indicate that this is a challenging problem. Moreover, reducing the stiffness during approach to a contact position makes the motion more sensitive to inertia. This behavior is due to the lag in tracking the target trajectory and the uncompensated end-effector mass, which are due to the lower value of the stiffness used as the robot approaches a contact position. The long-term objective of this research is to achieve smooth and reliable motion in different changing-contact manipulation tasks.

References

Abu-Dakka, F.J., Rozo, L., Caldwell, D.G., 2018. Force-based variable impedance learning for robotic manipulation. Robotics and Autonomous Systems 109, 156–167.

Ahmad, W., 2006. Incremental learning of gaussian mixture models URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.79.3582&rep=rep1&type=pdf.

Ahn, K., Chung, W.K., Yourn, Y., 2004. Arbitrary states polynomial-like trajectory (aspot) generation, in: 30th Annual Conference of IEEE Industrial Electronics Society, 2004. IECON 2004, IEEE. pp. 123–128.

Ajay, A., Bauza, M., Wu, J., Fazeli, N., Tenenbaum, J.B., Rodriguez, A., Kaelbling, L.P., 2019. Combining physical simulators and object-based networks for control, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE. pp. 3217–3223.

Andrychowicz, M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., et al., 2018. Learning dexterous in-hand manipulation. arXiv preprint arXiv:1808.00177 .

Barragän, P.R., Kaelbling, L.P., Lozano-Pérez, T., 2014. Interactive bayesian identification of kinematic mechanisms, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 2013–2020.

Beetz, M., Stulp, F., Esden-Tempski, P., Fedrizzi, A., Klank, U., Kresse, I., Maldonado, A., Ruiz, F., 2010. Generality and legibility in mobile manipulation. Autonomous Robots 28, 21.

Biagiotti, L., Melchiorri, C., 2008. Trajectory planning for automatic machines and robots. Springer Science & Business Media.

Brogliato, B., 2019. Nonsmooth Mechanics. Models, Dynamics and Control: Erratum/Addendum. Ph.D. thesis. INRIA Grenoble-Rhone-Alpes.

Buşoniu, L., de Bruin, T., Tolić, D., Kober, J., Palunko, I., 2018. Reinforcement learning for control: Performance, stability, and deep approximators. Annual Reviews in Control 46, 8–28.

Calinon, S., Sardellitti, I., Caldwell, D.G., 2010. Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE. pp. 249–254.

Catto, E., 2017. Box2d. http://box2d.org. URL: https://box2d.org/.

Chatterjee, A., 1997. Rigid body collisions: some general considerations, new collision laws, and some experimental data. Cornell University.

Chatterjee, A., 1999. On the realism of complementarity conditions in rigid body collisions. Nonlinear Dynamics 20, 159–168.

Coumans, E., Bai, Y., 2016–2021. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org.

De Schutter, B., Heemels, W., Lunze, J., Prieur, C., et al., 2009. Survey of modeling, analysis, and control of hybrid systems. Handbook of Hybrid Systems Control–Theory, Tools, Applications , 31–55.

Engel, P., Heinen, M., 2010. Incremental learning of multivariate gaussian mixture models, in: BSAI, Springer.

Fan, Y., Gao, W., Chen, W., Tomizuka, M., 2017. Real-time finger gaits planning for dexterous manipulation. IFAC-2017 50, 12765–12772.

Fazeli, N., Zapolsky, S., Drumwright, E., Rodriguez, A., 2020. Fundamental limitations in performance and interpretability of common planar rigid-body contact models, in: Robotics Research. Springer, pp. 555–571.

Flanagan, J.R., Vetter, P., Johansson, R.S., Wolpert, D.M., 2003. Prediction precedes control in motor learning. Current Biology 13, 146–150.

Freeman, P., 2012. Minimum jerk trajectory planning for trajectory constrained redundant robots .

Gams, A., Nemec, B., Ijspeert, A.J., Ude, A., 2014. Coupling movement primitives: Interaction with the environment and bimanual tasks. IEEE Transactions on Robotics 30, 816–830.

Gandhi, M., Pan, Y., Theodorou, E., 2017. Pseudospectral model predictive control under partially learned dynamics. arXiv preprint arXiv:1702.04800 .

Grassmann, R., Johannsmeier, L., Haddadin, S., 2018. Smooth point-to-point trajectory planning in $se(3)$ with self-collision and joint constraints avoidance, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE. pp. 1–9.

Halm, M., Posa, M., 2021. Set-valued rigid body dynamics for simultaneous frictional impact. arXiv preprint arXiv:2103.15714 .

Hausman, K., Springenberg, J.T., Wang, Z., Heess, N., Riedmiller, M., 2018. Learning an embedding space for transferable robot skills, in: International Conference on Learning Representations.

Huang, B., Li, M., De Souza, R.L., Bryson, J.J., Billard, A., 2016. A modular approach to learning manipulation strategies from human demonstration. Autonomous Robots 40, 903–927.

Hyde, J.M., Tremblay, M.R., Cutkosky, M.R., 1997. An object-oriented framework for event-driven dextrous manipulation, in: Experimental Robotics IV. Springer, pp. 51–61.

Johansson, M.K.J., 2003. Piecewise linear control systems: a computational approach. volume 284. Springer.

Johnson, A.M., Burden, S.A., Koditschek, D.E., 2016. A hybrid systems model for simple manipulation and self-manipulation systems. The International Journal of Robotics Research 35, 1354–1392.

Katz, D., Brock, O., 2011. A factorization approach to manipulation in unstructured environments, in: Robotics Research. Springer, pp. 285–300.

Kawato, M., 1999. Internal Models for Motor Control and Trajectory Planning. Current Opinion in Neurobiology , 718–727arXiv:https://www.ncbi.nlm.nih.gov/pubmed/10607637.

Kemp, C.C., Edsinger, A., Torres-Jara, E., 2007. Challenges for robot manipulation in human environments [grand challenges of robotics]. IEEE Robotics & Automation Magazine 14, 20–29.

Khader, S.A., Yin, H., Falco, P., Kragic, D., 2020. Data-efficient model learning and prediction for contact-rich manipulation tasks. IEEE Robotics and Automation Letters 5, 4321–4328.

Kopicki, M., Zurek, S., Stolkin, R., Moerwald, T., Wyatt, J.L., 2017. Learning modular and transferable forward models of the motions of push manipulated objects. Autonomous Robots 41, 1061–1082.

Kramberger, A., Shahriari, E., Gams, A., Nemec, B., Ude, A., Haddadin, S., 2018. Passivity based iterative learning of admittance-coupled dynamic movement primitives for interaction with changing environments, in: International Conference on Intelligent Robots and Systems, pp. 6023–6028.

Kroemer, O., Niekum, S., Konidaris, G., 2019. A review of robot learning for manipulation: Challenges, representations, and algorithms. arXiv preprint arXiv:1907.03146 .

Kronander, K., Billard, A., 2014. Learning compliant manipulation through kinesthetic and tactile human-robot interaction. IEEE Transactions on Haptics 7, 367–380.

Kupcsik, A.G., Deisenroth, M.P., Peters, J., Neumann, G., et al., 2013. Data-efficient generalization of robot skills with contextual policy search, in: AAAI 2013, pp. 1401–1407.

Lee, A.X., Lu, H., Gupta, A., Levine, S., Abbeel, P., 2015. Learning force-based manipulation of deformable objects from multiple demonstrations, in: International Conference on Robotics and Automation, pp. 177–184.

Levine, S., Koltun, V., 2013. Guided policy search., in: International Conference on Machine Learning, pp. 1–9.

Liu, C.K., 2009. Dextrous manipulation from a grasping pose, in: ACM Transactions on Graphics, ACM. p. 59.

Marth, G., Tarn, T.J., Bejczy, A.K., 1993. Stable phase transition control for robot arm motion, in: [1993] Proceedings IEEE International Conference on Robotics and Automation, IEEE. pp. 355–362.

Mathew, M.J., Sidhik, S., Sridharan, M., Azad, M., Hayashi, A., Wyatt, J., 2019. Online Learning of Feed-Forward Models for Task-Space Variable Impedance Control, in: International Conference on Humanoid Robots.

Mills, J.K., Lokhorst, D.M., 1993. Control of robotic manipulators during general task execution: A discontinuous control approach. The International Journal of Robotics Research 12, 146–163.

Nagabandi, A., Konolige, K., Levine, S., Kumar, V., 2020. Deep dynamics models for learning dexterous manipulation, in: Conference on Robot Learning, PMLR. pp. 1101–1112.

Nam, S.H., Yang, M.Y., 2004. A study on a generalized parametric interpolator with real-time jerk-limited acceleration. Computer-Aided Design 36, 27–36.

Niekum, S., Chitta, S., Barto, A.G., Marthi, B., Osentoski, S., 2013. Incremental semantically grounded learning from demonstration., in: Robotics: Science and Systems, Berlin, Germany. pp. 10–15607.

Parmar, M., Halm, M., Posa, M., 2021. Fundamental challenges in deep learning for stiff contact dynamics. arXiv preprint arXiv:2103.15406 .

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

Pezzato, C., Ferrari, R., Corbato, C.H., 2020. A novel adaptive controller for robot manipulators based on active inference. IEEE Robotics and Automation Letters 5, 2973–2980.

Piazzi, A., Visioli, A., 2000. Global minimum-jerk trajectory planning of robot manipulators. IEEE transactions on industrial electronics 47, 140–149.

Romano, J.M., Hsiao, K., Niemeyer, G., Chitta, S., Kuchenbecker, K.J., 2011. Human-inspired robotic grasp control with tactile sensing. IEEE Transactions on Robotics 27, 1067–1079.

Rozo, L., Calinon, S., Caldwell, D.G., Jimenez, P., Torras, C., 2016. Learning physical collaborative robot behaviors from human demonstrations. IEEE Transactions on Robotics 32, 513–527.

Sidhik, S., Sridharan, M., Ruiken, D., 2020. Learning hybrid models for variable impedance control of changing-contact manipulation tasks, in: Annual Conference on Advances in Cognitive Systems (ACS).

Sidhik, S., Sridharan, M., Ruiken, D., 2021. Towards an online framework for changing-contact robot manipulation tasks, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5203–5210. doi:10.1109/IROS51168.2021.9636278.

Song, M., Wang, H., 2005. Highly efficient incremental estimation of Gaussian mixture models for online data stream clustering, in: Priddy, K.L. (Ed.), Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, pp. 174–183. doi:10.1117/12.601724.

Stoianovici, D., Hurmuzlu, Y., 1996. A Critical Study of the Applicability of Rigid-Body Collision Theory. Journal of Applied Mechanics 63, 307–316. doi:10.1115/1.2788865.

Stulp, F., Theodorou, E.A., Schaal, S., 2012. Reinforcement learning with sequences of motion primitives for robust manipulation. IEEE Transactions on robotics 28, 1360–1370.

Sung, H.G., 2004. Gaussian mixture regression and classification. Ph.D. thesis. Rice University.

Tarokh, M., 1991. Hyperstability approach to the synthesis of adaptive controllers for robot manipulators, in: Proceedings. 1991 IEEE International Conference on Robotics and Automation, pp. 2154–2159 vol.3. doi:10.1109/ROBOT.1991.131947.

Toussaint, M., Allen, K., Smith, K.A., Tenenbaum, J.B., 2018. Differentiable physics and stable modes for tool-use and manipulation planning., in: Robotics: Science and Systems.

Ugur, E., Piater, J., 2015. Bottom-up learning of object categories, action effects and logical rules: From continuous manipulative exploration to symbolic planning, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 2627–2633.

Wieber, P.B., Tedrake, R., Kuindersma, S., 2016. Modeling and control of legged robots, in: Springer handbook of robotics. Springer, pp. 1203–1234.

Yang, C., Ganesh, G., Haddadin, S., Parusel, S., Albu-Schaeffer, A., Burdet, E., 2011. Human-like adaptation of force and impedance in stable and unstable interactions. IEEE transactions on robotics 27, 918–930.

Zhang, D., Wei, B., 2017. A review on model reference adaptive control of robotic manipulators. Annual Reviews in Control 43, 188–198.

Zhang, T., Ramakrishnan, R., Livny, M., 1997. Birch: A new data clustering algorithm and its applications. Data Mining and Knowledge Discovery 1, 141–182.