Advances in Cognitive Systems X (2021) 1-6 Submitted X/20XX; published X/20XX

Hybrid Models for Control of Changing-Contact Manipulation Tasks

Saif Sidhik SXS1412@BHAM.AC.UK
Mohan Sridharan M.SRIDHARAN@BHAM.AC.UK
Intelligent Robotics Lab, University of Birmingham, UK

Dirk Ruiken DIRK.RUIKEN@HONDA-RI.DE
Honda Research Institute Europe GmbH, Germany

Abstract

Many robot manipulation tasks involve discrete action sequences characterized by continuous dynamics, while the transitions between these discrete *dynamic modes* are characterized by discontinuous dynamics. The individual modes may represent different types of contacts, surfaces, or other factors, and each mode and transition between the modes may require a different control strategy. This paper describes a piece-wise continuous, hybrid control framework for such manipulation tasks. The underlying representation enables the robot to automatically and efficiently detect the transitions between known modes, recognize new modes, and incrementally learn a dynamics model for variable impedance (i.e., stiffness) control in each mode, invariant to the direction of motion and the magnitude of applied forces. The framework is evaluated on a robot manipulator sliding an object along a surface to achieve a desired motion trajectory in the presence of changes in surface friction, applied force, or the type of contact between the object and the surface.

1. Introduction

Consider a robot manipulator (Figure 1) that has to slide an object over a surface along a given motion pattern. This task's dynamics, i.e., the relationships between the forces acting on the robot and the resultant accelerations, vary markedly before and after the object comes in contact with the surface. The dynamics also vary based on the type of contact (e.g., surface or edge contact), surface friction, applied force, and other factors. We consider such manipulation tasks involving changes in dynamics due to changes in the nature of contact as "*changing-contact*" manipulation tasks. Core industrial assembly tasks, e.g., peg insertion, screwing, stacking, and pushing, 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 model of the dynamics. However, it is possible to construct a *hybrid* model with continuous dynamics within each of a number of discrete *dynamic modes* with distinct strategies to control motion (Kroemer et al., 2019). The overall dynamics are then *piece-wise continuous*, with the robot transitioning between the modes as appropriate (Lee et al., 2017).

To use a piece-wise continuous hybrid control framework, the robot needs a transition model (i.e., transition controller) that automatically identifies the mode at any point in time. In addition,

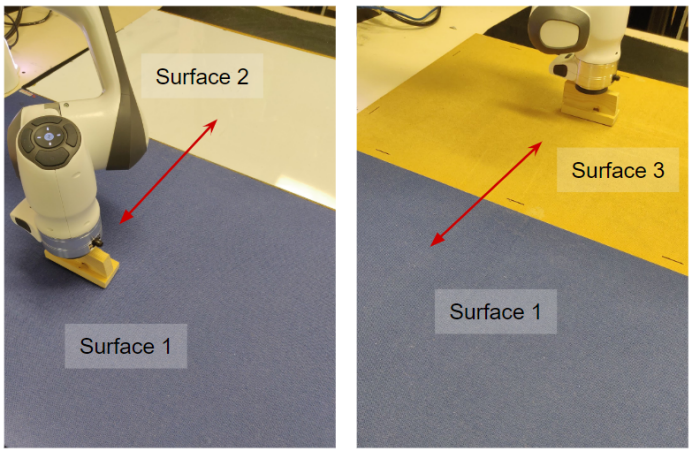

Figure 1: Manipulator sliding an object in a pattern along three surfaces with different friction.

to accurately follow the motion pattern, the robot will need to account for changes in related factors within a mode, and for any previously unseen modes experienced while performing the task. Existing methods that support one or more of these desired capabilities learn from large labeled training datasets, use comprehensive knowledge of domain dynamics, impose unrealistic assumptions or hardware requirements, or use a representation that makes the methods computationally expensive. Research in human motor control, on the other hand, indicates that people start performing any new task with higher arm stiffness to account for unforeseen disturbances, and quickly acquire experience to carry out the task accurately with much lower stiffness. They do so by building internal models of task dynamics with generic (domain- or task-independent) and specific (domain- or task-dependent) representations, and use these models to predict hand or object configurations and the forces during task execution (Burdet et al., 2001; Kawato, 1999; Shadmehr & Krakauer, 2008). We map these findings to the following tenets for a hybrid control framework for changing-contact manipulation tasks:

- The hybrid model has one or more dynamic modes, each with a predictor (i.e., "forward/dynamics model") of sensor measurements, a control law, and a relevance condition.
- The motion pattern, predictors, and control laws are defined concisely in the Cartesian space, and any given mode uses an abstract task-dependent representation of state.
- A mode change is confirmed using samples collected at high stiffness, and the robot adapts an existing forward model or constructs a new one to operate at a suitable stiffness.

where the first two tenets are related to representation and the third to information processing. The combination of these tenets is novel. We describe a computational framework that implements these tenets to make the following contributions:

- An approach to incrementally build a non-linear, piece-wise continuous model of any given task's dynamics without prior knowledge of all its modes or the order in which they appear.

- A transition controller that automatically identifies the modes of any given task, and identifies transitions to existing or new modes during task execution.
- A representation that enables efficient adaptation of the dynamics model of each mode, and an abstract representation that makes the identification of modes independent of the motion direction and magnitude of applied forces.
- An approach to automatically and incrementally adapt the forward model of any mode's dynamics, using the prediction error to revise the gain parameters of the control law for accurate and compliant motion in that mode.

The novelty is in the first three contributions; the last one builds on our prior work on variable impedance control of continuous contact tasks (Mathew et al., 2019). We describe these contributions and explore the control and learning problems in changing-contact manipulation tasks in the context of a robot arm sliding a block on a surface in a predefined motion pattern. This representative task exhibits the complexities and discontinuities involved in changing-contact manipulation tasks due to discrete dynamics changes, and supports controlled experiments. In addition, we limit sensor input to that from a force-torque sensor at the end of the manipulator, and demonstrate the framework's ability to perform the desired task reliably in the presence of discrete changes in surface friction, applied force, and type of contact.

In our formulation, for any given task, the robot is given (a) a target motion pattern, i.e., sequence of poses; (b) sensor inputs (i.e., force-torque and velocity measurements at wrist); and (c) a transition model with one or more modes, each with a forward model, control law, and relevance condition. The robot's *performance task* is to follow the motion pattern accurately and smoothly (i.e., compliant motion without jerk) using the dynamics model and control law of the mode identified by the transition model. The *adaptation task* is to incrementally revise the parameter values of the forward model and the gain parameter values of the control law of the current mode. The *learning task* is to incrementally construct a dynamics model when a new mode is identified by the transition controller. We begin by describing our computational framework and its components in Section 2. We then discuss the results of experimental evaluation in Section 3, review related work in Section 4, and conclude in Section 5.

2. Problem Formulation and Framework

In this section, we first formulate changing-contact manipulation tasks as a piece-wise continuous hybrid system and describe the basic framework (Section 2.1). We then describe the adaptation of the forward model and the control law within any given dynamic mode (Section 2.2), followed by the approach to detect mode changes and model previously unseen dynamic modes (Section 2.3).

2.1 Piece-wise Continuous Hybrid System

In a piece-wise continuous hybrid system, the state can be described as the tuple $\langle m, s \rangle$ where $m \in M$ is a *mode* from a discrete set of modes M, and $s \in S_m$ is an element in a continuous subspace $S_m \subseteq \mathbb{R}^d$ associated with m. This formulation assumes that subspaces do not intersect or

overlap, i.e., $S_m \cap S_n = \emptyset \quad \forall \quad m \neq n$. The evolution of s within a mode m is determined by a discrete-time continuous function $S_m(.)$, but the state transition is discrete and discontinuous at the boundaries between modes. Lee et al. (2017) called the boundary between modes m and m', where the transition occurs, as *guard regions* that are denoted by $G_{m,m'} \subseteq S_m$. In the guard regions, s is transported to $s_{t+1} \in S_{m'}$ through a *reset function* $r_{m,m'}(.)$. State propagation is thus governed by:

$$s_{t+1} = \begin{cases} r_{m_t,m_{t+1}}(s_t) + w_t & \text{if } s_t \in G_{m_t,m_{t+1}} \\ S_m(s_t) + w_t & \text{if } s_t \in S_{m_t} \end{cases} \tag{1}$$

where w_t is additive (Gaussian) process noise in sensor measurements. In the context of the manipulation tasks considered in this paper, the forces, torques, and velocities measured by the robot at its end-effector constitute the observable state (s) of the system that varies continuously within each contact mode. This formulation makes the reasonable assumption that properties such as friction are continuous across the surface of each object. The control strategy guiding the object's motion in the static or smoothly changing environment within that mode can be considered to determine the function $S_m(.)$ governing the evolution of s in that mode. When mode changes occur in the guard regions, the dynamics corresponds to a new state in a mode n where the state evolution is guided by function $S_n(.)$. For changing-contact tasks, measurements within the guard region are pronounced and significantly different compared with the readings within a dynamic mode. The mode switches impose structure on manipulation tasks; as we describe in Section 2.3, the transitions can be considered as triggers for changing the current dynamics model of the domain.

In our framework, each mode comprises a: (i) forward (dynamics) model that predicts part of the observable state (i.e., end effector forces and torques); (ii) control law that includes a feedforward term and feedback terms based on the error between forward model predictions and actual measurements; and (iii) relevance condition that (in)validates a mode based on the magnitude of changes in sensor measurements. A simple transition model could have just one mode or use an ad hoc strategy that assigns particular models to particular parts of the motion pattern. Recall that the performance task is to follow the assigned motion pattern (e.g., computed by an external planner) using the dynamics model and control law of the suitable mode. Our formulation, however, seeks to account for changes in relevant factors (e.g., surface friction and applied force) in order to improve: (i) performance within each dynamic mode through online adaptation of the parameters of the forward model and the control law; and (ii) overall performance by automating the recognition of mode changes and the learning of models for previously unseen modes. We being by describing the framework's components and the adaptation strategy in each mode.

2.2 Adaptive Forward Model and Control Law

For any given mode, we build on our prior work on continuous contact manipulation tasks (Mathew et al., 2019) to train the dynamics model and design the control law. Specifically, the continuous dynamics of a mode is modeled as an Incremental Gaussian Mixture Model (IGMM) (Song & Wang, 2005). IGMM internally uses a variant of the Expectation-Maximization (EM) algorithm to fit the model parameters. In our implementation, the GMM was incrementally fit over points $\mathbf{X} = (X_1, ..., X_T)$, with $X_t = [S_{t-1}, D_t]$ where each point contains information about a subset of

previous observable state (S_{t-1}), along with the *current* values of the subset of the observable state to be predicted (D_t). Once trained, the forward model provides a function:

$$f : S_t \mapsto D_{t+1} \tag{2}$$

that predicts D_{t+1} at the next time step as a function of the current (measured) value of S_t, using Gaussian Mixture Regression (GMR) (Sung, 2004). Recall that in our work, the primary sensor is the force-torque sensor at the end effector, and that the motion pattern, forward model, and control law are defined in the Cartesian ("task") space instead of the high-dimensional ("joint") space of the joints of the manipulator. For the dynamics model, the observable state thus includes the end-effector forces, torques, and velocities ($[F_{ee}, \tau, \dot{x}]$), with $S_{t-1} = [F_{ee_{t-1}}, \tau_{t-1}, \dot{x}_{t-1}]$ and $D_t = [F_{ee_t}, \tau_t]$. We used the magnitude of force, torque, and velocity instead of their 3D vector representation because the magnitudes of frictional forces and torques are ideally independent of the motion direction. This simplified representation is sufficient to predict the end-effector forces and torques along (or against) the motion direction, which is what the models are supposed to do. Components along other axes can be computed when needed and this simplification makes the modeling process simpler, computationally efficient, and *independent of the direction of motion*.

There are a few options to set the initial values of the parameters of a dynamics model. For any known modes, the robot sets these values based on sensor measurements during a single demonstration of the task, e.g., a human manually moving the arm along the desired motion pattern. For other unknown modes, the robot sets the initial values based on sensor measurements collected in a time interval immediately after a transition to a guard region is determined; we describe this approach in Section 2.3. A key feature of our framework is that whenever a mode is active, its parameter values are revised continuously based on sensor measurements obtained during task execution.

The predictions from a forward model provide the feed-forward term that cancels out the effect of the environment (friction) forces during motion, in the control law:

$$u_t = \mathbf{K_t^P} \Delta x_t + \mathbf{K_t^d} \Delta \dot{x}_t + u_t^{fc} + \lambda_{t-1} k_t \tag{3}$$

$$u_t^{fc} = \mathbf{K_t^f} \Delta F_t + F_t^d \tag{4}$$

$$\mathbf{K}_t^p = \mathbf{K}_{free}^p + (1 - \lambda_{t-1})(\mathbf{K}_{max}^p - \mathbf{K}_{free}^p) \tag{5}$$

$$\lambda_t = 1 - \frac{1}{1 + e^{-r(\varepsilon_t - \varepsilon_0)}} \tag{6}$$

where the control command (u_t) to the robot end-effector at time t, is a task space wrench, i.e., a vector of 3D force and 3D torque. Equation 3 is a hybrid force-motion variable impedance control law in which the first two terms implement a proportional-derivative (PD) feedback controller for motion control ($\mathbf{K_t^P}$ and $\mathbf{K_t^d}$ are the positive definite stiffness and damping matrices), the third term is a force feedback control signal (u_t^{fc}), and the last term is based on prediction error of the forward model. The *state* for the control law considers the pose (x_t) and velocity ($\dot{x}$) in task space, i.e., in 6D coordinates (three each for position and orientation). Equation 4 implements a simple proportional controller for force control with $\mathbf{K_t^f}$ as the gain matrix and ΔF as the error in task-space force; in our illustrative task, the direction for force control is orthogonal to direction of motion control. The last term in Equation 3 is a control signal based on the prediction of the forward model (k_t)

in the current mode and a weighting factor $\lambda \in [0, 1]$ based on the error in the prediction (ε_t)—we use a logistic function (Equation 6) whose (hyper)parameters (growth rate r, sigmoid midpoint ε_0) are tuned experimentally. The feed-forward term thus contributes only if the mode's dynamics are modeled properly. Equation 5 updates the stiffness parameter of the overall control law based on the prediction error; $\mathbf{K}^p_{max}$ is the maximum allowed stiffness, $\mathbf{K}^p_{free}$ is the minimum stiffness needed for accurate pose tracking in the absence of external disturbances (i.e., in free space), and $\mathbf{K}^d_t = \sqrt{\mathbf{K}^p_t/4}$ is a known constraint for critically-damped systems (Ijspeert et al., 2013).

We have established the advantages of a variable impedance control formulation for continuous contact tasks in prior work (Mathew et al., 2019). When used in conjunction with the continuously-revised forward models, the gain parameters are adapted automatically to account for changes in related factors within the mode, minimizing the corresponding prediction errors. In addition, the hybrid force-motion controller provides compliance in the direction of force control while following the desired motion pattern. As a result, the manipulator is able to adapt a mode's gain parameters automatically if, for instance, the surface is tilted or raised during task execution.

2.3 Recognizing Mode Changes and New Modes

The adaptation strategy described above cannot recover from sudden substantial changes in sensor measurements characterizing any mode change. We define the *relevance condition* for each mode as a threshold on the magnitude of change in the measured end-effector forces and torques. This threshold is typically a significant fraction ($10-20\%$) of the range of sensor measurements observed during the demonstration of the task. During task execution, the current mode is invalid when this threshold is exceeded and the robot briefly uses a high-stiffness control strategy, collecting sensor data to confirm and respond to the transition. If the transition is found to be to a previously modeled mode, the robot starts using (and revising) the corresponding dynamics model (and control law). If the transition is to a previously unseen mode, the robot uses the constructed sensor data to construct a dynamics model (for the new mode) that is then revised over time.

We begin by describing the choice of the feature representation (i.e., state descriptor) for each mode, which strongly influences the reliability and efficiency of the method for recognizing mode changes. This representation is task dependent but the objective is to identify attributes that concisely and uniquely represent the modes and vary substantially only when mode change occurs. Recall that factors of interest are changes in the surface friction, type of contact, and applied force. For our illustrative task of sliding an object over surfaces with different values of friction, 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 the surface normals do not change; μ is then give by:

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

One concise feature representation for this task is thus $\frac{\|F^t_{ee}\|}{R^t}$, with the superscript denoting time steps; this has the effect of making mode classification independent of the applied force's magnitude.

For changes in the type of contact, the end-effector orientation is a useful feature. However, small changes in orientation can lead to significant changes in the measured torques (see Figure 2),

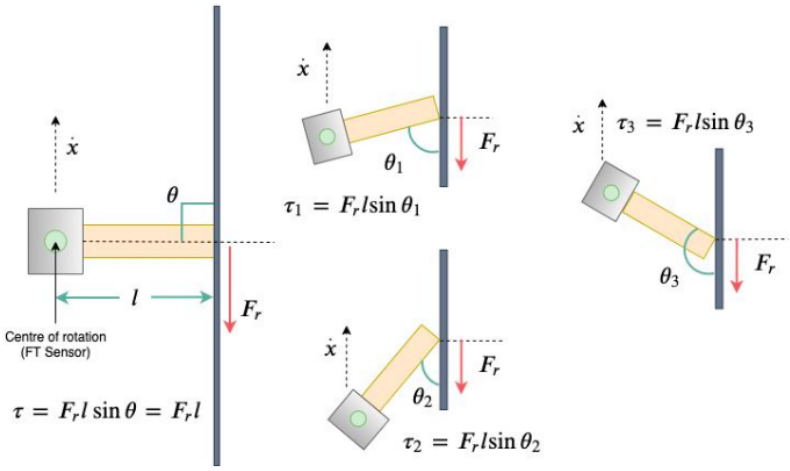

Figure 2: The torque measured at the pivot (τ) varies for different relative orientation of the object (θ), unlike the force at the tip (F_r). The object is moving at $\dot{x}$ resulting in a frictional resistance F_r in the opposite direction at point of contact.

resulting in the recognition of different modes. A more reasonable feature is the magnitude of the end-effector torques that can be measured using the force-torque sensor in the wrist:

$$\tau = F_r l \sin\theta \tag{8}$$

where F_r is the force at the tip, l is the length of the pivot arm, and θ is the orientation between the surface normals. Figure 2 shows that for any object, τ is different for the different types of contacts. With the magnitude of the torques ($\|\tau\|$) as the feature representation, modes can be classified independent of the motion direction and object orientation. However, this representation would not work when the magnitude of the applied force differs. If we instead assume that the force measured at the wrist (F_{ee}) approximates the force at the tip of the object (F_r), which is a reasonably assumption, Equations 7 and 8 imply that $\frac{\|\tau^t\|}{R^t}$ is invariant to the magnitude of the applied force for a fixed relative orientation between the objects in contact:

$$\tau = \mu R l \sin(\theta) \tag{9}$$

Ideally $\frac{\|\tau\|}{R}$ is constant for each mode (based on θ) provided object geometry (l) and friction (μ) do not change. Experimental studies reveal that this property by itself is insufficient to distinguish between contacts when the applied normal force changes because the assumption about kinematic friction (i.e., that $F_r = \mu R$) does not hold in many real-world situations (Baraff, 1991). We thus use $[\frac{\|\tau\|}{R}, \frac{\|F_{ee}\|}{R}]$ 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. Note that a similar one-time exercise can be used to determine a feature representation with the desired generalization capability for other changing-contact tasks.

Given this abstract feature representation of each mode, our approach for recognizing mode changes is based on an online incremental clustering algorithm called Balanced Iterative Reduc-

ing and Clustering using Hierarchies (BIRCH) (Zhang et al., 1997); we use the implementation of BIRCH in the Scikit-learn library (Pedregosa et al., 2011). This algorithm efficiently and incrementally clusters a data stream without having to examine all existing data points in all clusters. Recall that a batch of sensor data is collected at high stiffness when there is a sudden substantial difference in the sensor measurements. We implement the clustering algorithm in the abstract feature space described above, using the collected sensor data. The fraction of the feature vectors assigned to any existing cluster determines the confidence in the corresponding mode being the current mode. If the highest confidence value is above a threshold (e.g., twice as likely of the second highest confidence value), the corresponding mode is triggered and the associated dynamics model is used and revised as described in Section 2.2. If the input feature vectors are not sufficiently similar to any existing mode, a new mode is created and the initial parameter values of the corresponding dynamics model are set using the sensor data in the cluster for that mode. Note that the purpose of clustering is to obtain the confidence measure and construct the dynamics model (if a new mode is created); the "average" of the samples in a cluster, in the abstract representation, is sufficient to model a mode.

Table 1: Control loop of framework

Input : Desired motion pattern as sequence of task space way-points, Control parameters: $\mathbf{K}^p_{free}, \mathbf{K}^p_{max}$; Dynamics models $M = \{f_i : i \in [1, N]\}$; Current mode: $m = 0$.

```
1  while Motion pattern not complete do
2      if Object in contact with surface then
3          if mode transition detected then
               // Set high stiffness
4              K_t^p ← K_max^p
               // Detect (new/existing) mode
5              m = detect_classify_mode()
               // Populate new model for new mode
6              if new mode found then
7                  M = M ∪ f_m
8              end
9          end
10         Update and use f_m for control (Section 2.2)
11     else
12         K_t^p ← K_free^p
13     end
14 end
```

Algorithm 1 is an overview of the framework's control loop for a changing-contact manipulation task, e.g., sliding an object on a surface. It proceeds until a desired motion pattern is completed. Our control, adaptation, and learning methods are not used when the manipulator is moving in free space (lines 11-13); they are only used after there is contact with a surface (lines 2-10). The robot detects mode changes when there are substantial changes in the sensor measurements (line 3). The

robot responds by setting a high stiffness (line 4), collecting sensor measurements, determining the transition to a new or existing mode (line 5), and creating new models if necessary (lines 6-8). In the absence of a mode transition (e.g., the detected change in sensor measurement was an anomaly), the robot continues with the current mode and dynamics model (line 10).

3. Experimental Setup and Evaluation

We used a 7-DoF Franka Emika Panda manipulator robot for our experiments—see Figure 1. The robot had to slide an object along an assigned motion pattern on a surface, and we considered variations in surface friction ("changing surface" task) and in contact types ("changing contact type" task). This desired motion pattern is provided by an external planning module or encoded based on a single demonstration of the task by the human designer, e.g., human moves the manipulator along a desired path. We experimentally evaluated the following hypotheses:

H1: Building separate dynamics models for the different modes results in better performance than using a single model that is revised continuously;

H2: Our hybrid framework provides reliable and efficient performance for changing-contact manipulation tasks; and

H3: Our hybrid framework's performance is robust to changes in motion direction and changes in applied force.

where **H1** explored the need for learning different dynamics models for different modes; **H2** and **H3** examined whether the framework can reliably and efficiently transition to the appropriate mode (and dynamics model) in the presence of changes in direction of motion and applied forces. We used the *root mean square error* (RMSE) in related measurements (e.g., end effector position, forces, and stiffness) as the key performance measure. Unless stated otherwise, each data point in the results below is the result of 10 repeated trials on the robot. Since our approach is significantly different from state of the art approaches for changing-contact manipulation tasks (e.g., those based on deep learning in the space of joint angles of the robot manipulator), we do not provide an experimental comparison with these approaches but discuss them in Section 4. A video demonstrating the operation of our framework and some results discussed in this paper can be found online[1].

3.1 H1: Need for Multiple Models

We ran two experiments to evaluate hypothesis **H1**, i.e., the need for separate dynamics models for different modes of changing-contact tasks. The robot's task was to slide an object (rigidly fixed to the end-effector) over a flat surface, and the surface friction was changed to obtain two distinct surfaces. In the first experiment, the robot had a dynamics model for the first (rougher) surface but not for the second (smoother) surface. We expected the robot to overshoot the trajectory and experience a sudden increase in the joint torques when it transitioned from the first surface to the second.

1. https://youtu.be/m210rxIDZ7Q

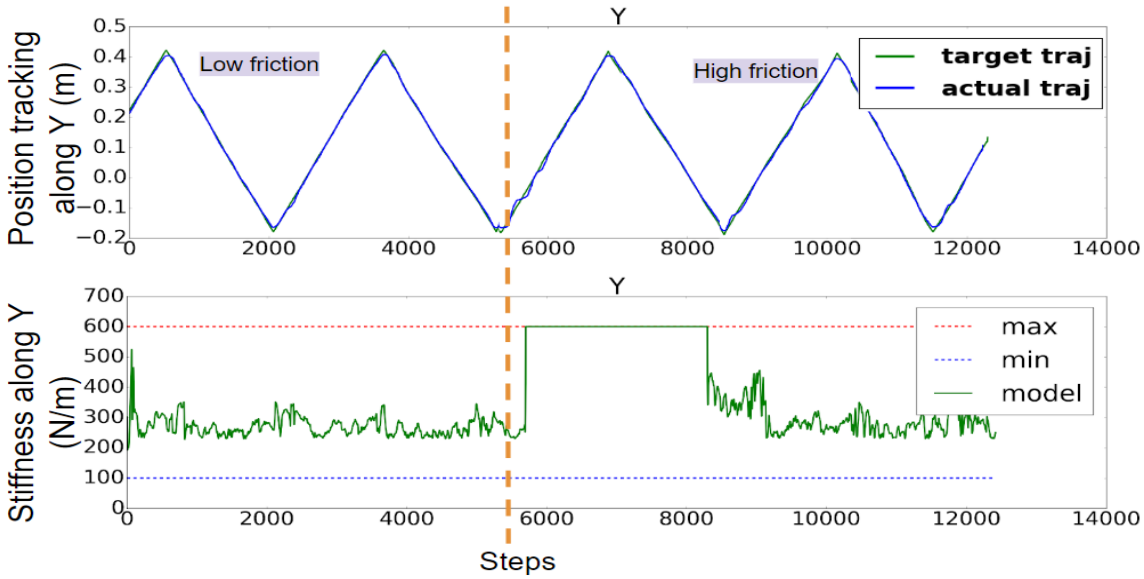

Figure 3: Performance when a dynamics model is constructed from scratch for a new mode. **Top:** position tracking; **Bottom:** variation in controller stiffness. Robot spends considerable time under high stiffness when transition to new mode occurs (dashed vertical red line).

Experimental results matched these expectations. In 90% of the trials, the robot was unable to complete the task; it stopped before the trajectory was completed. The feed-forward values predicted by the model for the rougher surface were much higher than the actual values for the smoother surface. This discrepancy made the robot overshoot when it transitioned to the smoother surface; the joint torques reached safety limits and the robot stopped moving. These results indicated that a single incrementally-revised dynamics model was unable to handle pronounced, discrete mode changes.

In the second experiment, the robot repeated the same task starting with a single dynamics model but used a high stiffness controller to build a new model when a change to the second surface is detected; the robot does not initially have a forward model for the second surface. We considered both transitions (i.e., rougher to smoother surface and vice versa), and the robot performed better than using a single dynamics model that is incrementally revised; the task was completed successfully in all the trials. Figure 3 shows the position tracking performance in one such trial. Operating with a high value of stiffness until a reliable forward model is constructed results in more energy being expended than when the dynamics models for the two modes are available. When the dynamics models for the two surfaces are available, the robot is able to switch between them when needed, spending much less time under high-stiffness, as observed by comparing the stiffness plots in Figure 3 and 4. The difference in performance is statistically significant, e.g., RMSE for the position tracking plots in Figure 3 and Figure 4 are 0.017 and 0.015 respectively. We repeated these experiments for other combinations of surfaces (with different friction) and for motion patterns executed over more than two different surfaces with different surface friction. In each case, the robot was

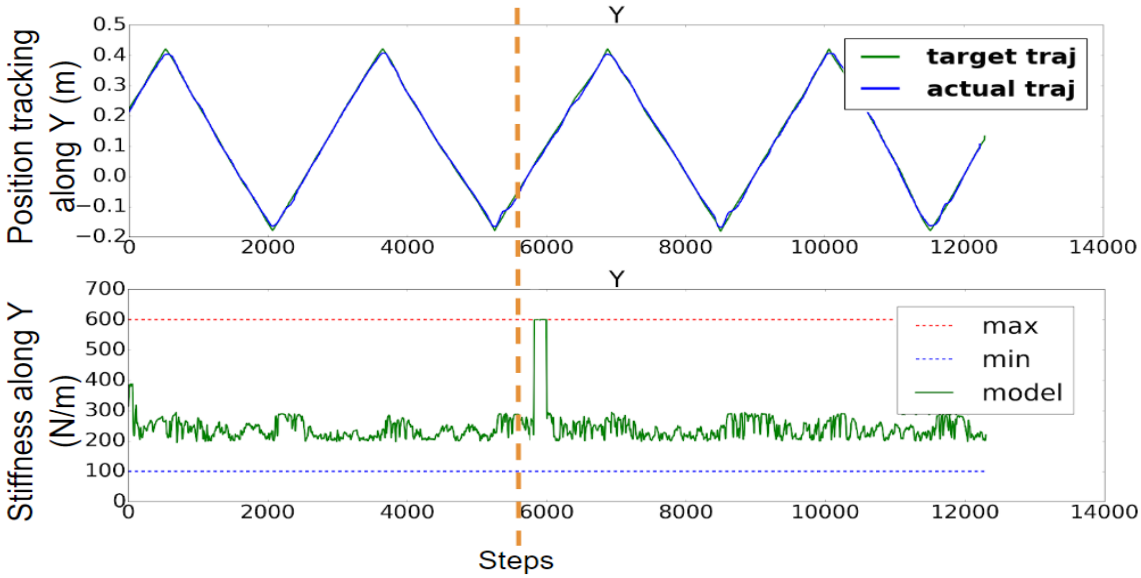

Figure 4: Performance when separate dynamics models are available for two distinct modes. **Top:** position tracking; **Bottom:** variation in controller stiffness. The robot spends very little time under high stiffness when the transition between modes occurs (dashed vertical red line).

able to detect the new mode and incrementally revise the parameter values of the new mode, and to transition to using the existing models when appropriate. These results support hypothesis **H1**.

3.2 H2+H3: Detecting Different Surfaces

To evaluate hypotheses **H2** and **H3**, we first considered the changing surface task. The robot was asked to slide an object along a desired trajectory, and it experienced three previously unseen surfaces with different values of friction. We expected the robot to identify a transition to each new mode (i.e., each surface) and incrementally build a dynamics 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 the corresponding dynamics model.

Figure 5 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 the plot) because of the similarity in their friction values.

Figure 6 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 the

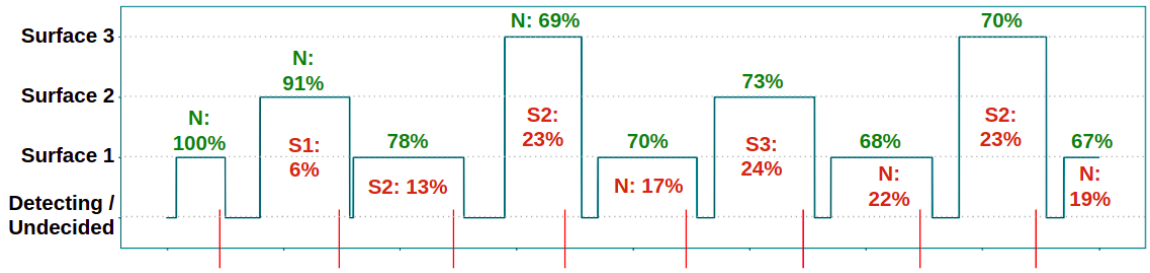

Figure 5: Modes detected and their confidence values; red vertical lines on the x-axis indicate actual mode transitions. The number on top of a peak (in green) indicates the confidence with which the transition was identified; the number below a peak (in red) corresponds to the mode with the next highest confidence. "N" indicates a transition to a new mode.

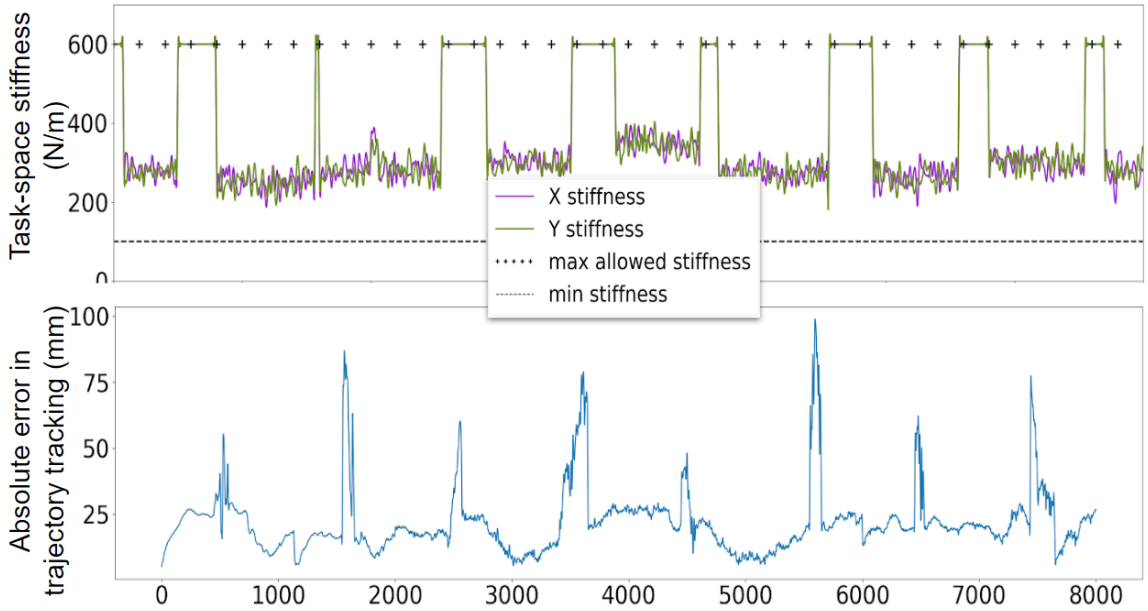

Figure 6: Performance for changing-surface task. **Top:** controller stiffness. **Bottom:** absolute error in trajectory tracking. The spikes during trajectory tracking correspond to a temporary, incorrect feed-forward prediction by the previous model after the guard regions.

high-stiffness mode and identified the current mode; the robot then used suitable low(er) stiffness to complete the task. As discussed earlier, switching to a previously seen mode requires a much shorter period of high stiffness compared with building a new dynamics model. These results support hypothesis **H2** and to some extent **H3**.

3.3 H2+H3: Different Types of Contacts

To further explore hypotheses **H2** and **H3**, we ran trials with the changing contact type task. The robot's task was to slide an object along a desired trajectory on a surface while experiencing each

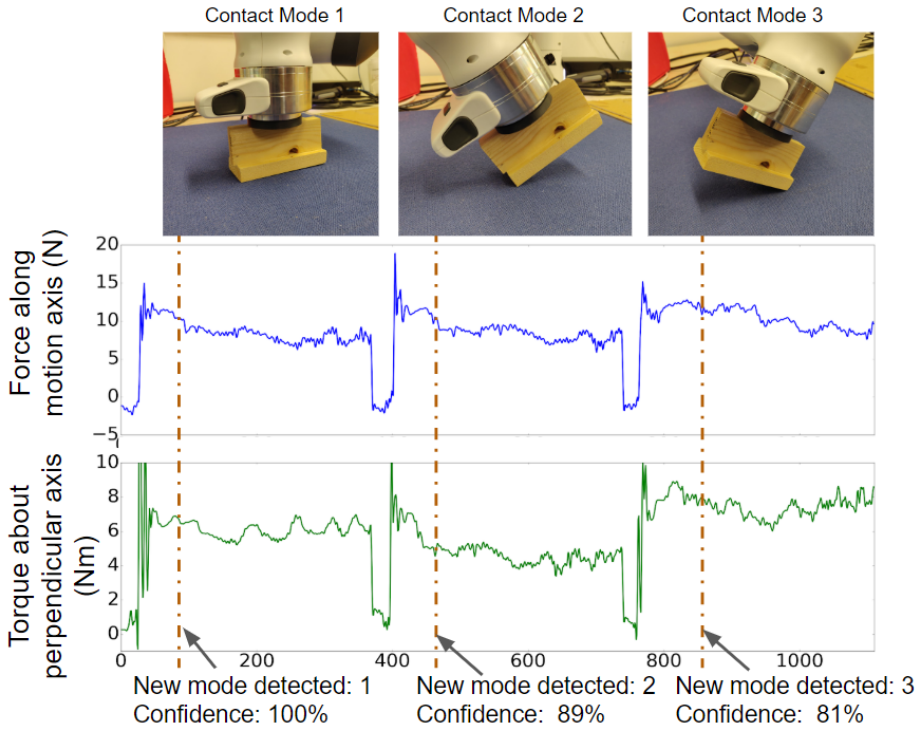

Figure 7: Illustrative trial of changing contact type task. **Top:** Contact types when the end-effector's motion is towards the right; **Middle:** Forces measured along the direction of motion; **Bottom:** Torques measured about axis parallel to surface and perpendicular to direction of motion. Spikes in the measurements correspond to contact transitions; dashed (brown) vertical lines indicate when the framework managed to build a reliable dynamics model of the corresponding mode.

of the three different types of contacts shown in the top part of Figure 7. As stated earlier, the motion pattern was extracted from a single demonstration of the task; the robot did not have prior knowledge of the different dynamic modes. During each trial, the robot approached the table to execute a particular type of contact while maintaining a normal force of $10N$. Contact with the surface triggered a transition, and the robot proceeded to slide the object along the surface with the force of $10N$. As before, we expected the robot to confirm any transition to a previously seen mode using data from a brief period of high stiffness motion, and to use the corresponding dynamics model. Also, we expected any transition to a previously unseen mode to require data collection over a longer period of high stiffness to build a dynamics model of the mode. We also expected the robot to be robust to changes in sequence of contacts, motion direction, and normal force.

Figure 7 summarizes results for a particular trial. It shows the incremental construction of dynamics models for each mode, along with the variation of end-effector force and the torque measured along the axis that is most affected by the motion. We observe that separate models were learned for the three contacts (i.e., three different modes) with high confidence.

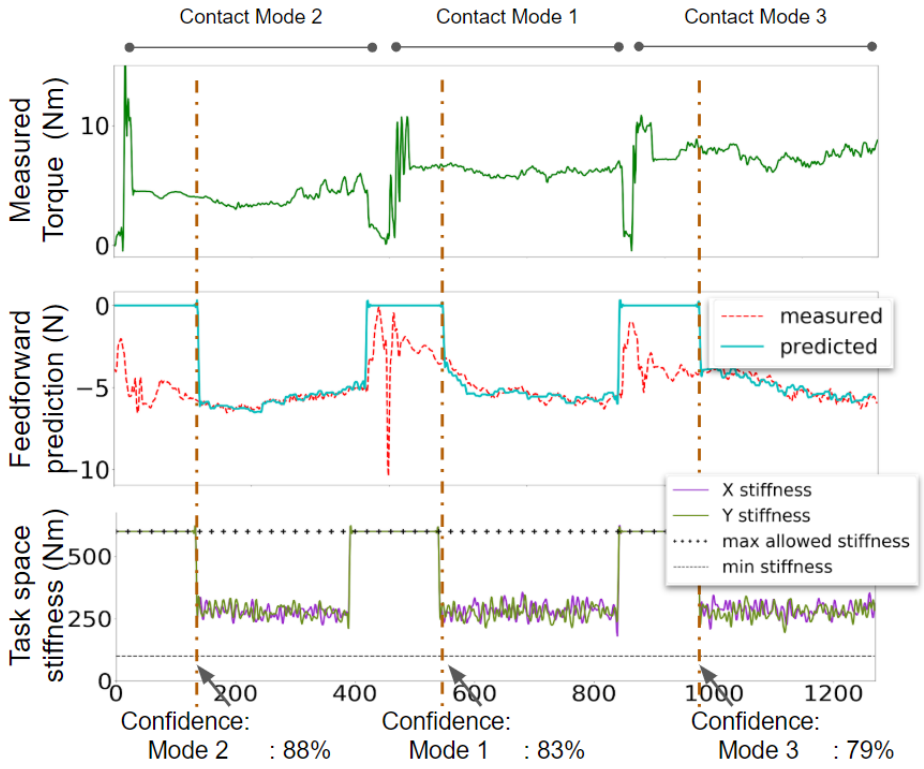

Figure 8: Testing previously constructed dynamics models for the changing contact type task, but with contacts appearing in a different sequence. **Top:** Torques measured about the axis parallel to surface and perpendicular to motion direction; spikes in measurements correspond to contact; **Middle:** End-effector forces predicted by forward model for the current mode; **Bottom:** Variations in the controller stiffness based on errors in the predicted forces.

Next, Figure 8 shows the results of testing the existing dynamics models with the same overall trajectory but with a different sequence of contacts. We observed that the robot was still able to recognize the modes quickly and accurately, regardless of the order in which they occurred. The second plot in Figure 8 shows the end-effector forces predicted by the dynamics model for the current contact mode. The feed-forward term was used and revised online when the model is reliable, but the term's value was zero when the robot had not identified the mode. Similarly, the stiffness parameters of the impedance controller were varied according to the prediction error of the dynamics model; recall that high stiffness values are used temporarily when the robot is collecting data to build the dynamics model for the mode.

Next, Figure 9 demonstrates the robustness of the framework to motion along a direction different from that used during training. The feed-forward model predictions and the corresponding variable impedance behavior for one of the trials is shown, along with the dynamics model chosen with the highest confidence (bottom of the figure). The modes were always identified correctly.

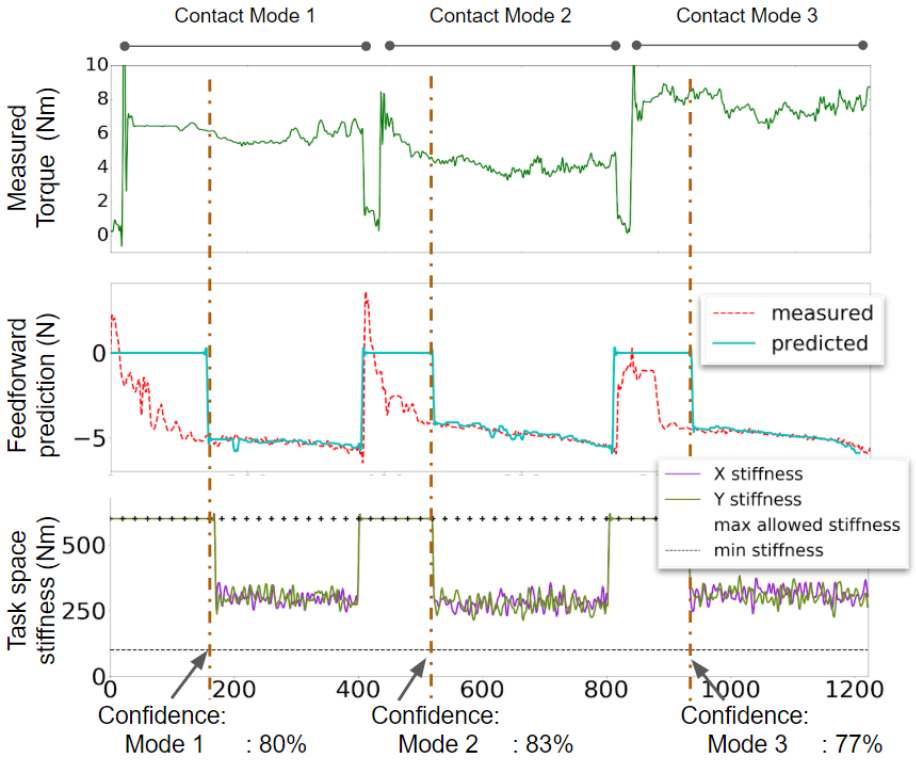

Figure 9: Testing previously constructed dynamics models for the changing contact type task, but with motion in a different direction. **Top:** Torques measured about the axis parallel to surface and perpendicular to direction of motion; **Middle:** End-effector forces predicted by forward model for the current mode; **Bottom:** Variations in the controller stiffness based on errors in predicted forces.

The framework was then tested for the same task and set of contacts but with the manipulator applying a different constant normal force on the surface. The robot's performance in one experimental trial of identifying the modes and adapting the the existing dynamics models is summarized in Figure 10. These results match those in Table 2; although the confidence associated with the modes was a little lower and the robot took a little longer to recognize the modes when the normal force was changed, the hybrid framework was able to recognize the modes correctly and complete the task successfully using variable impedance control. The lower confidence can be attributed to the kinetic friction assumption (i.e., that $\mu = F/R$) not being completely correct in many real world tasks. The results in Figures 7–9 also indicate that the time taken to recognize modes is longer if the modes under consideration are similar, e.g., modes 1 and 3 in these experiments. These results support hypotheses **H2** and **H3** and indicate the need for further research on the choice of the abstract feature representation for the dynamic modes of changing-contact manipulation tasks.

Discussion: Overall, the results in Sections 3.1–3.3 demonstrate the effectiveness of our framework on a physical robot performing a sliding task as an instance of changing-contact manipulation tasks. Section 3.1 demonstrated the need for having multiple dynamics models for performing a

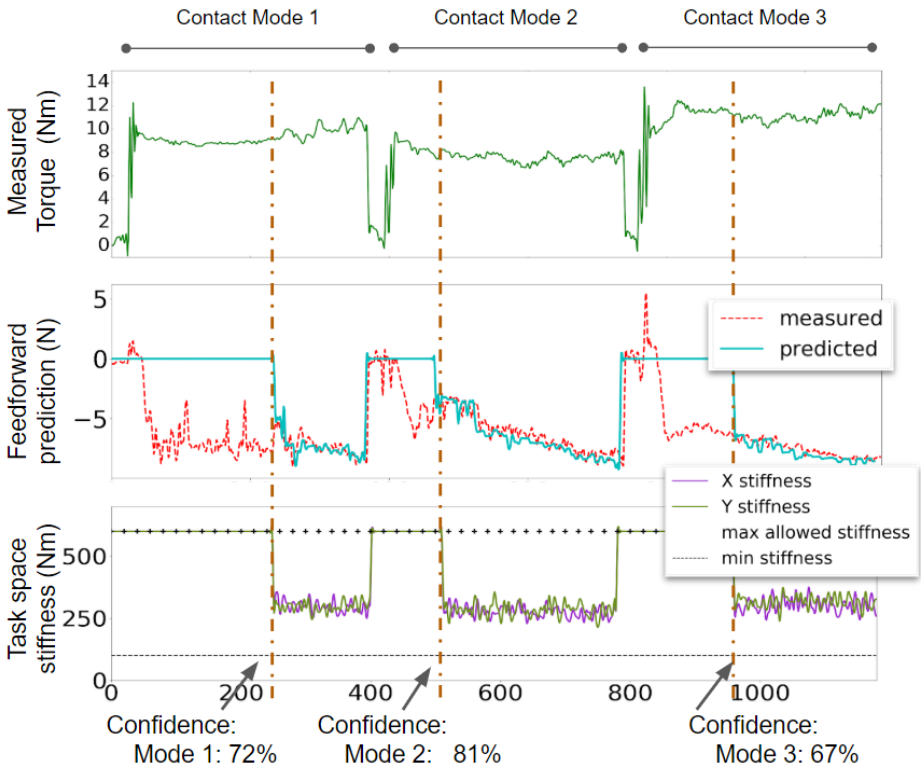

Figure 10: Testing previously constructed dynamics models for the changing contact type task, but with a different normal force (20N instead of 10N). **Top:** Torques measured about the axis parallel to surface and perpendicular to direction of motion; **Middle:** End-effector forces predicted by current mode's forward model; **Bottom:** Variations in the controller's stiffness values due to errors in predicted forces.

manipulation task involving discrete dynamics change; we studied this in the context of transitions between surfaces of different friction. At the point of transition, the use of the current dynamics model resulted in prediction and trajectory tracking errors. Also, learning a new model from scratch when a mode change is detected resulted in a longer period of operation under high stiffness than when the robot just had to transition to the previously learned dynamics model of this new surface.

Next, Section 3.2 demonstrated the framework's ability to reliably detect discrete mode changes and build dynamics models for the mode without any previous knowledge of the dynamics. The framework was also supported the identification of previously seen modes (surfaces), and the reuse of the corresponding dynamics model for performing the task with a suitable (lower) stiffness while being robust to changes in the direction of motion. Finally, Section 3.3 illustrated the framework's ability to detect different types of contacts (as modes). The robot was able to acquire the dynamics models of these modes, and to reuse these models when the task had to be performed with the contact types appearing in a different sequence, changes in motion direction, and with different applied (normal) contact force.

Normal Force $= 10N$	Ground Truth		
Detected Mode	Contact 1	Contact 2	Contact 3
Contact 1	83	9	16
Contact 2	2	88	1
Contact 3	14	2	79
New Mode	1	1	4
Normal Force $= 20N$	Ground Truth		
Detected Mode	Contact 1	Contact 2	Contact 3
Contact 1	81	10	17
Contact 2	3	86	1
Contact 3	15	2	77
New Mode	1	2	5

Table 2: Confusion matrix of average confidence ($\%$) over 10 trials of mode recognition based on previously constructed dynamics models for the three types of contacts (top part of Figure 7). **Top:** Normal force of $10N$; **Bottom:** Normal force of $20N$.

4. Related Work

Many methods have been developed to address the contrl and learning problems in robot manipulation (Kroemer et al., 2019), especially methods based on reinforcement learning (RL) (Stulp et al., 2012) and those combining deep networks and RL for learning flexible behaviors from complex data (Andrychowicz et al., 2018; Hausman et al., 2018; Lowrey et al., 2018). These data-driven methods require large labeled datasets collected through multiple repetitions of the task by the robot. These requirements are difficult to satisfy in practical domains, especially on a physical robot. Also, the training process optimizes several parameters and the internal representations and decision making mechanisms are opaque, making it computationally expensive to learn action policies and difficult to transfer them to new tasks. Although sim-to-real strategies have been developed to reduce the need for training on real robots, aspects such as the dynamics of rigid bodies with friction are too complicated to be modeled in a real-time dynamics simulator (Johnson et al., 2016). Also, these methods are not well-suited for a hybrid system formulation because they implicitly or explicitly consider a single model for the entire manipulation task (Kroemer et al., 2019).

RL and optimal control methods for robot manipulation often assume the task dynamics are smooth. The use of adaptation or learning strategies with a hybrid formulation of robot control has been limited (Lee et al., 2017), with many methods focusing on bipedal locomotion (Nakamura et al., 2007). Planning methods for manipulation often take the dynamics of manipulation into account (Toussaint et al., 2018; Jain & Niekum, 2018), but they assume prior knowledge of the models of the system, and of the actions and modes of interest. Unlike other online learning methods (Yang et al., 2011), our framework does not require a periodically repeating trajectory, and it does not learn a time-series of controller parameters to be used in a repeatable dynamic environment. Instead, our framework adapts its controller based on the current dynamic forces experienced.

Many methods have shown the benefits of incorporating modes or phases in the design of controllers (Romano et al., 2011), and many methods learn controllers for such multi-phase tasks (Buşoniu et al., 2018; Koval et al., 2016). 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). This makes the learned policy dependent on the specific movements and their sequence.

In a departure from existing work, our framework for changing-contact manipulation is inspired by an understanding of (and insights from) human motor control behavior. In particular, people start performing any new task with higher arm stiffness and quickly learn to perform the task with much lower stiffness. They do so by building internal models of task dynamics that are used to predict hand/object configurations and forces experienced while transitioning between discrete modes during task execution (Burdet et al., 2001; Kawato, 1999; Shadmehr & Krakauer, 2008). Our framework significantly expands approaches that incorporate modes in the design of controllers to support: (a) automatic recognition of modes and identification of new modes invariant to the direction of motion and magnitude of the applied force; and (b) incremental learning and revision of dynamic models for variable impedance control in the individual modes.

5. Conclusions and Future Work

This paper described a computational framework inspired by human motor control, which formulated changing-contact manipulation tasks as a piece-wise continuous, hybrid system. Any such task is considered to be made up of discrete modes with continuous dynamics and distinct control strategies. Each mode comprises a forward (predictive) model, a hybrid force-motion (feedback) control law, and a relevance condition. The use of different representations for a mode's components enables the robot to automatically, reliably, and efficiently identify mode changes and incrementally adapt the dynamics models for the modes. Unlike data-driven methods that require many labeled training examples, our framework is able to build and revise the dynamics model for each observed mode from very few examples. Unlike existing control methods for manipulation tasks, our method is not limited to the sequence of modes seen during demonstrations, and it does not require prior information about the number of modes in the task (Lee et al., 2017; Niekum et al., 2013). In this paper, we have described the experimental evaluation of our framework's capabilities on a physical robot performing the representative changing-contact manipulation task of sliding an object held in its end-effector along a desired motion trajectory on a surface. Experimental results illustrate the ability to reliably follow the desired motion trajectory in the presence of changing surface friction, type of contacts, and applied force, invariant to changes in the motion direction and magnitude of applied forces. In addition, the framework formulates changing-contact manipulation tasks such that it can be applied to other tasks in this category such as peg-insertion, block pushing, stacking, etc. Furthermore, our hybrid framework may be adapted to other dynamics and control problems such as a mobile robot navigating, exploring, and collecting samples from different terrains.

Our future work will address the limitations of the current framework and explore new directions. First, the current strategy of switching between modes (and dynamics models) is not completely smooth, with occasional spikes in sensor measurements in the guard (i.e., transition) regions.

Future work will investigate the accurate prediction of the time and location of contact, and use this information to achieve smooth dynamics in changing-contact tasks. Second, the current approach for detecting mode changes will be difficult to use in tasks with many more modes. To address this limitation, we will explore other representations of the different modes, including those in continuous spaces. Third, we will expand the use of our framework to other examples of changing-contact manipulation tasks, and additional factors that influence such tasks. It would also be interesting to explore the automatic selection of the abstract feature representation suitable for the modes of each changing-contact manipulation task. The longer-term objective is to enable reliable, efficient, and smooth control in the context of a robot manipulator performing complex assembly tasks with multiple objects in complex domains.

References

Andrychowicz, M., et al. (2018). Learning dexterous in-hand manipulation. *arXiv preprint arXiv:1808.00177*.

Baraff, D. (1991). Coping with friction for non-penetrating rigid body simulation. *ACM SIGGRAPH computer graphics*, *25*, 31–41.

Burdet, E., Osu, R., Franklin, D. W., Milner, T. E., & Kawato, M. (2001). The Central Nervous System Stabilizes Unstable Dynamics by Learning Optimal Impedance. *Nature*, *414*, 446.

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.

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

Ijspeert, A. J., Nakanishi, J., Hoffmann, H., Pastor, P., & Schaal, S. (2013). Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, *25*, 328–373.

Jain, A., & Niekum, S. (2018). Efficient hierarchical robot motion planning under uncertainty and hybrid dynamics. *arXiv preprint arXiv:1802.04205*.

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.

Kawato, M. (1999). Internal Models for Motor Control and Trajectory Planning. *Current Opinion in Neurobiology*, (pp. 718–727).

Koval, M. C., Pollard, N. S., & Srinivasa, S. S. (2016). Pre-and post-contact policy decomposition for planar contact manipulation under uncertainty. *The International Journal of Robotics Research*, *35*, 244–264.

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

Lee, G., Marinho, Z., Johnson, A. M., Gordon, G. J., Srinivasa, S. S., & Mason, M. T. (2017). Unsupervised learning for nonlinear piecewise smooth hybrid systems. *arXiv preprint arXiv:1710.00440*.

Lowrey, K., Kolev, S., Dao, J., Rajeswaran, A., & Todorov, E. (2018). Reinforcement learning for non-prehensile manipulation: Transfer from simulation to physical system. *International Conference on Simulation, Modeling, and Programming for Autonomous Robots* (pp. 35–42).

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. *International Conference on Humanoid Robots*.

Nakamura, Y., Mori, T., Sato, M.-a., & Ishii, S. (2007). Reinforcement learning for a biped robot based on a cpg-actor-critic method. *Neural networks*, *20*, 723–735.

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

Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

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.

Shadmehr, R., & Krakauer, J. (2008). A Computational Neuroanatomy for Motor Control. *Experimental Brain Research*, *185*, 359–381.

Song, M., & Wang, H. (2005). Highly efficient incremental estimation of Gaussian mixture models for online data stream clustering. *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* (pp. 174–183).

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*. Doctoral dissertation, Rice University.

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

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