Learning Hybrid Models for Variable Impedance 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 comprise 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 sliding an object over a surface along a desired pattern, as shown in Figure 1. For the system comprising the robot and its environment, the dynamics of the task, 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 tasks that involve changes in dynamics due to changes in the nature of contact as "*changing-contact*" tasks. Tasks such as peg insertion, screwing, stacking, and pushing, which form the basis of industrial assembly tasks, are changing-contact tasks. The interaction dynamics of the robot system in such tasks are continuous except when a contact is made or broken. Similarly, many robot (and human) manipulation tasks are changing-contact tasks characterized by discontinuities in the dynamics when the nature of the interaction between objects changes. These discontinuities make it difficult to learn a single model of the task's dynamics, but it is possible to construct a *hybrid* system with continuous dynamics within each of a number of discrete *dynamic modes* that may need distinct control strategies (Kroemer et al., 2019). Then, the overall task's dynamics are piece-wise continuous, with the corresponding hybrid model being used to transition between the individual modes over time (Lee et al., 2017).

Constructing separate (continuous) models for the different modes introduces the need for a transition model (i.e., a strategy) that accurately chooses the mode under operation at any point in time, revises the existing dynamics models to adapt to changes within any given mode, and identifies and learns dynamics models for previously unseen modes. Existing methods achieve this objective by learning from large labeled training datasets, using comprehensive knowledge of domain dynamics, imposing unrealistic assumptions or hardware requirements, and/or using a state representation that makes it computationally expensive to learn the dynamics models. Research in human motor control, on the other hand, indicates that humans start performing any new task with higher arm stiffness to account for unforeseen disturbances, but quickly acquire experience to perform the task accurately with much lower stiffness. They do so by building internal models of task dynamics based on different representations to predict the configurations (of object and hand) and the forces during task execution (Burdet et al., 2001; Kawato, 1999; Shadmehr & Krakauer, 2008). These findings are mapped to the following tenets that form the basis of our computational framework:

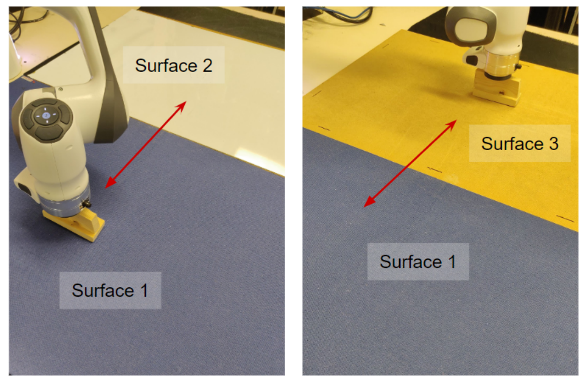

Figure 1: Manipulator sliding an object in a pattern along three surfaces with different friction.

- Each dynamic mode of the hybrid model comprises a forward model (i.e., predictor), a control law, and a relevance condition.
- The forward model's state is based on the end effector sensor measurements, with the controller operating in the task space (instead of joint space) and any given mode using an abstract task-dependent state representation.
- The relevance condition determines when the current mode is irrelevant; this is confirmed using samples collected under high stiffness, and either a new dynamic model is learned or an existing one is revised based on the prediction error.

The combination of these tenets (first two related to representation, third related to information processing) is novel and our framework implements these tenets to make the following contributions:

- Incrementally learns a non-linear, piece-wise continuous model of the dynamics of any given task without prior knowledge of its modes or the order in which the modes appear.
- Incorporates a transition model that automatically creates clusters corresponding to the modes of any given task, and identifies transition to existing or new modes during task execution.

- Introduces a reduced feature representation that makes the learning of dynamics models computationally efficient, and makes the identification of modes independent of the motion direction and magnitude of applied forces.
- Incrementally learns and revises a probabilistic model of any given mode's dynamics, using the model for variable impedance (i.e., stiffness) control 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). The underlying learning and control challenges are explored in the context of a manipulator sliding an object on a surface in a desired (given) motion pattern defined in the task space. 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. We describe our framework in Section 2. Section 3 discusses the experimental results, followed by review of related work in Section 4, and the conclusions in Section 5.

2. Problem Formulation and Framework

This section first describes the formulation of changing-contact manipulation tasks as a piece-wise continuous hybrid system (Section 2.1). Section 2.2 describes the control strategy and learning of continuous dynamics within a single mode. Section 2.3 explains the detection and modeling of the discrete dynamic modes.

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 the 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 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 the context of the sliding task considered in this paper, the forces and torques 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 in that mode can be considered to determine the function $S_m(.)$ governing the evolution of s in that mode. When mode changes occur (i.e., in the guard regions), the dynamics corresponds to a new state in 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 when compared with the readings within a dynamic mode. The mode switches impose structure on manipulation tasks; the transitions can be considered as triggers for changing the current model of the domain.

In our framework, each mode comprises a (i) forward model that predicts part of the observable state (end effector forces and torques), with the error between predictions and actual measurements used to revise the model and provide a component of the control signal; (ii) control law that includes a feed-forward term and feedback terms based on the measured error in predictions; and (iii) relevance condition that (in)validates a mode based on the magnitude of changes in sensor measurements. We describe these components below.

2.2 Forward Model and Control Law

We build on our previous work (which explored continuous contact tasks) (Mathew et al., 2019) to learn the dynamics model for each mode and develop the control strategy. Specifically, the continuous dynamics of each mode is learned using an Incremental Gaussian Mixture Model (IGMM) (Song & Wang, 2005). IGMM internally uses a variant of the Expectation-Maximization (EM) algorithm to fit the model. 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). During task execution, the learned 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 this paper, the primary sensor is the force-torque sensor at the end effector; the observable state 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 the motion direction; it makes the learning process simpler, more computationally efficient, and *independent of the direction of motion*. The learned model always predicts the forces and torques along (or against) the direction of motion; the components along other axes can be computed when needed.

The predictions from the forward model provide the feed-forward term that cancels out the effect of the environment forces (friction) 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 u_t, the control command 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 (with $\mathbf{K_t^p}$ and $\mathbf{K_t^d}$ being 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 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 learned 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 (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). Note that the hybrid force-motion controller provides the appealing property of compliance in the direction of force control while following the desired motion pattern. As a result, the manipulator is able to adapt smoothly and automatically if, for instance, the surface is tilted or raised during task execution.

2.3 Recognizing Mode Changes and New Modes

Any change in mode is accompanied by a sudden significant change in the sensor readings. In our framework, the *relevance condition* is defined as a threshold on the magnitude of change in the measured end-effector forces and torques. A common threshold is determined by the designer for all modes associated with any given task; the current mode is relevant until this threshold is exceeded. The robot responds to any change in mode by briefly using a high-stiffness control strategy while quickly obtaining a batch of sensor data to confirm and respond to the transition. Once a mode change is confirmed, the robot learns a new dynamics model for a new mode or uses (and revises) an existing dynamics model if the change is to a previously known mode.

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). This algorithm incrementally and dynamically clusters incoming data for given memory and time constraints, without having to examine all existing data points or clusters. We used the implementation of BIRCH in the Scikit-learn library (Pedregosa et al., 2011). Each cluster represents a mode in a suitable feature space (more details below), and the clusters are updated using a set/batch of feature data. The fraction of the input feature vectors assigned to any existing cluster determines the confidence in the corresponding mode being the current mode. If the highest such confidence value is above a task-dependent threshold, the corresponding mode is triggered and the associated dynamics model is used and revised. If the input feature vectors are not sufficiently similar to an

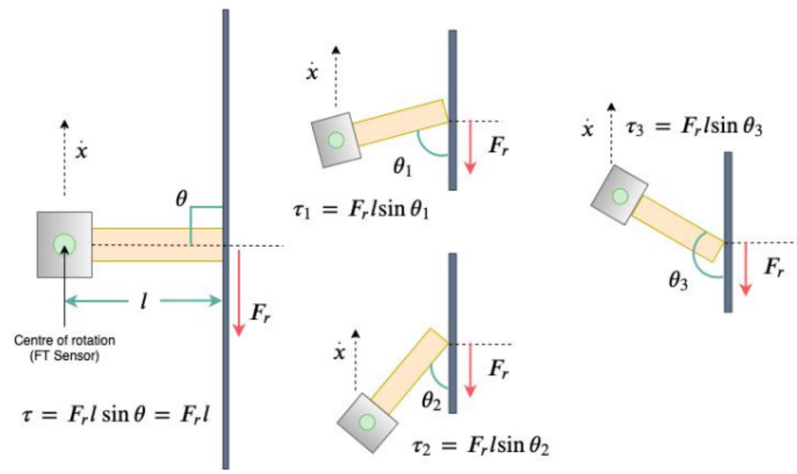

Figure 2: The torque measured at the pivot (τ) varies for different relative orientation of the object (θ), unlike the force at the tip (F_r). The object is moving at $\dot{x}$ resulting in a frictional resistance F_r in the opposite direction at point of contact.

existing cluster, a new cluster (i.e., mode) is constructed and the the corresponding dynamics model is learned—see Section 2.2.

The key factor influencing the reliability and efficiency of this approach is the choice of the feature representation (i.e., the state descriptor) for the mode. This representation is task dependent but the objective is to identify properties that concisely and uniquely representing 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 the 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 their surface normals do not change); μ is then give by:

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

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

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

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       $\mathbf{K}^p_t \leftarrow \mathbf{K}^p_{max}$
      `// Detect (new/existing) mode`
5   m = detect_classify_mode()
      `// Populate new model for new mode`
6   **if** *new mode found* **then**
7         $M = M \cup f_m$
8   **end**
9   **end**
10   Update and use f_m for control (Section 2.2)
11   **else**
12     $\mathbf{K}^p_t \leftarrow \mathbf{K}^p_{free}$
13   **end**
14 **end**

Algorithm 1 is an overview of the framework's control loop for a manipulator sliding an object on a surface; it proceeds until a desired motion pattern is completed. Control and learning methods are used only after the object is in contact with the surface (lines 2-10), not when the manipulator is moving in free space (lines 11-13). The robot detects mode changes when there are large changes in the sensor measurements (line 3). The robot responds to any detected mode change by setting a high stiffness (line 4), collecting feature samples, 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, 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 a desired 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 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 hypothesis:

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

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

H3: The framework's performance is robust to changes in motion direction and applied forces.

where **H1** explores the need for learning different dynamics models for different modes; **H2** and **H3** examine whether the framework can reliably and efficiently transition to the appropriate mode (and model) in the presence of changes in direction of motion and applied forces. We used the *root mean square error* (RMSE) in the context of different measurements (e.g., end effector position, forces, stiffness etc) 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 a significant departure from state of the art approaches for manipulation tasks (e.g., those based on deep learning), we do not provide an experimental comparison with these approaches but include a discussion in Section 4. A video demonstrating the operation of our framework and some of the experimental results discussed in this paper can be found online[1].

3.1 H1: Need for Multiple Models

We first evaluated hypothesis **H1**, i.e., the need for separate dynamics models for different modes in the context of changing-contact tasks. As the robot was sliding an object (rigidly fixed to the end-effector) over a flat surface, the surface friction was changed to obtain two distinct surfaces. The robot had a dynamics model for the first surface but not for the second surface.

Experimental results indicated that in 90% of the trials, the robot was unable to complete the task, i.e., the robot stops performing the task before the trajectory is completed. The feed-forward values being predicted by the model for the rougher surface were much higher than those required for the smoother surface, making the robot overshoot (when it transitioned to the smoother surface)

1. `https://youtu.be/m210rxIDZ7Q`

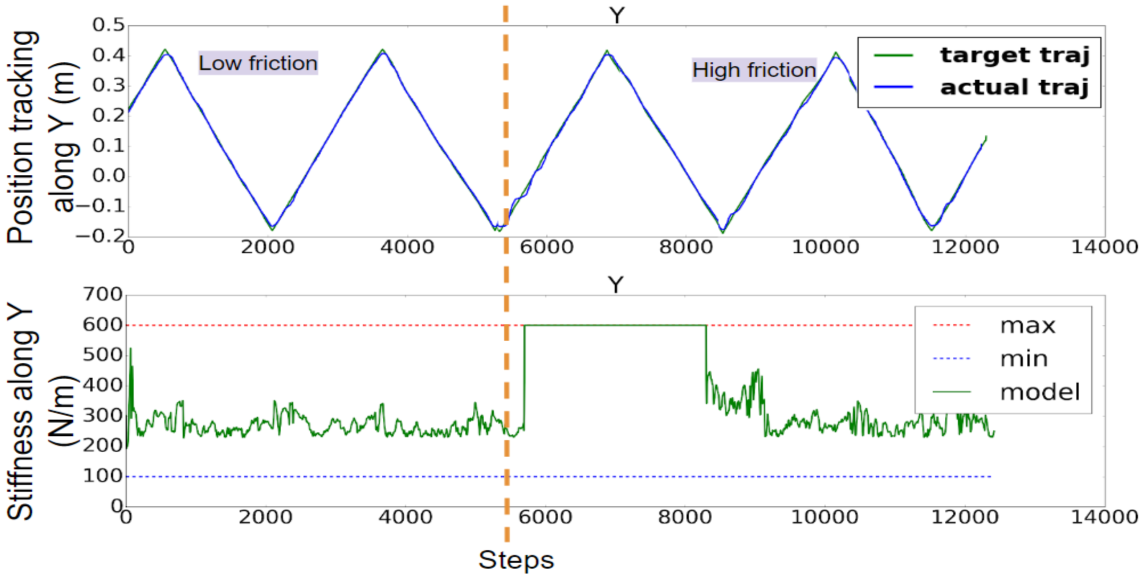

Figure 3: Performance when a dynamics model is learned from scratch for a new mode. **Top:** position tracking; **Bottom:** variation in controller stiffness.

and reach the safety limits imposed on the joint torques, resulting in the robot stopping its motion. In other words, the observed performance is unreliable with a single incrementally revised dynamics model when there are pronounced discrete changes in the mode.

Next, the robot performed the same task using a single model with the difference that it uses a high stiffness controller to learn a new model from scratch when a change to the second surface is detected; recall that the robot does not initially have a forward model for the second surface. This strategy performed better than with a single incrementally learned dynamics (forward) model, and the task was completed successfully in all the trials—Figure 3. The robot had to operate with high-stiffness until a reliable forward model for the new mode had been created, which expends much more energy than necessary. On the other hand, when the models for the two surfaces (i.e., modes) are available, the robot is able to switch between them spending much less time under high-stiffness—Figure 4. The difference in performance is statistically significant, e.g., RMSE for the 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 over more than two different surfaces with different surface friction. In each case, the robot was able to detect the new mode and learn a new model from scratch, and to transition to using the existing models when appropriate (discussed in Section 3.2). 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. As the robot was sliding an object between two surfaces, one surface was randomly changed to that with a different value of friction. Starting with no knowledge about the surfaces, the robot incrementally identified

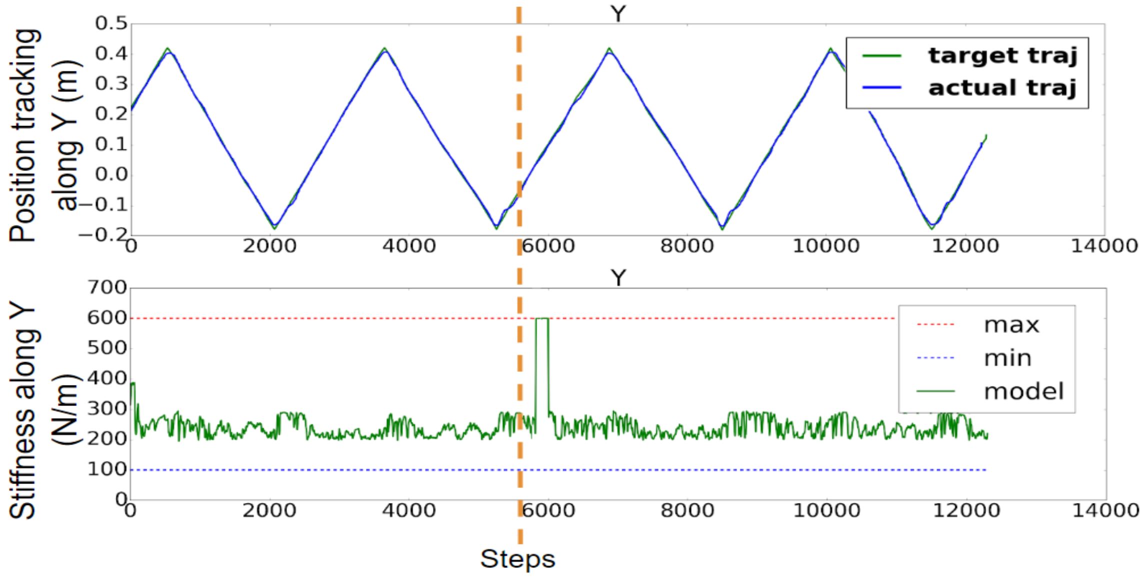

Figure 4: Performance when learned models are used for distinct modes. **Top:** position tracking; **Bottom:** variation in controller stiffness.

each dynamic mode and built a dynamics model for each mode (i.e., each distinct surface) while operating briefly under high stiffness. Once it learned the dynamics models for the different modes, it responded to subsequent mode changes by using the corresponding model.

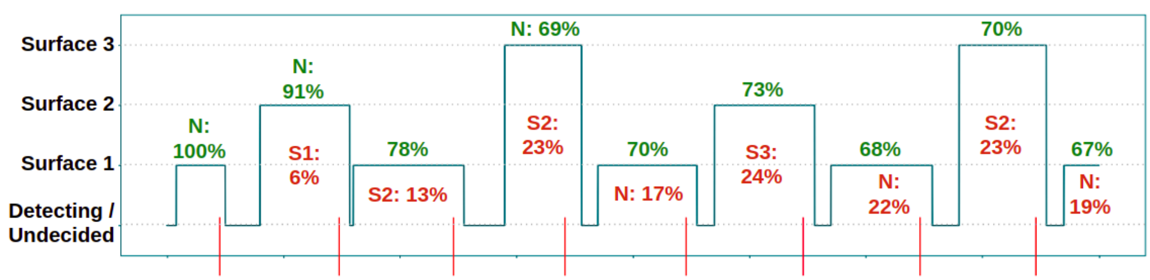

Figure 5: Modes detected and their confidence values. The numbers on top of a peak (in green) indicate the confidence with which the transition was identified. The number below a peak (in red) shows the mode with the next highest confidence. "N" indicates a transition to a new mode. The red vertical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

Figure 5 summarizes the results over one trial of this experiment. We observe that the framework is able to identify transitions to existing or new modes with high confidence. In each instance, the second best choice of mode is associated with a much lower value of confidence. The results also indicate that the algorithms and the underlying feature representation make the performance robust to changes in the direction of motion, i.e., a new mode is not identified when the manipulator moves

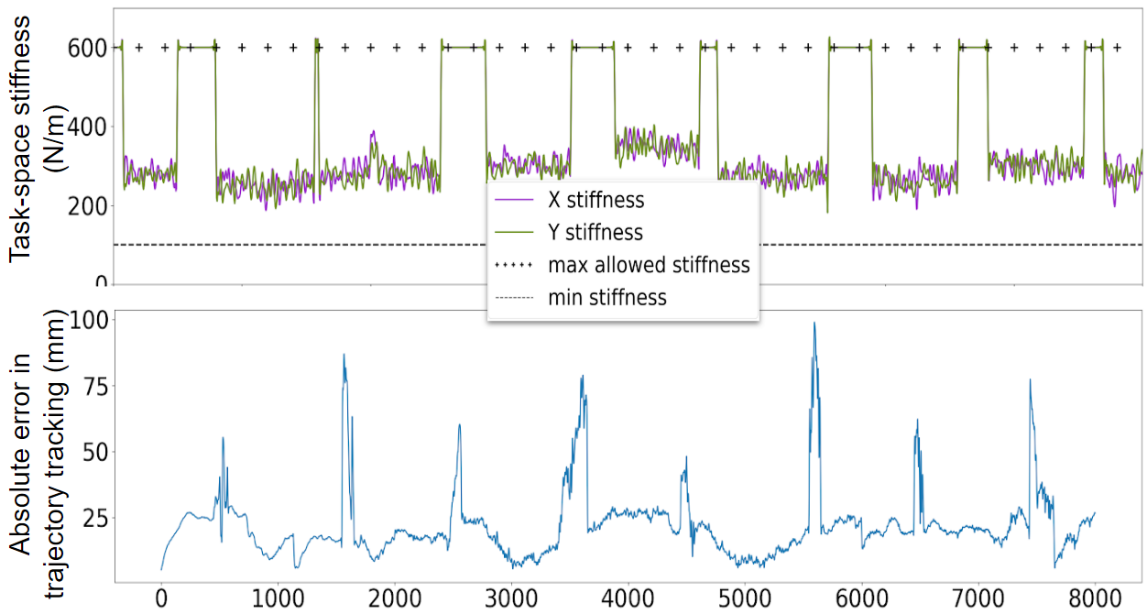

Figure 6: Performance for changing-surface task. **Top:** variation of controller stiffness. **Bottom:** absolute error in trajectory tracking. The spikes during trajectory tracking correspond to a temporary, incorrect feed-forward prediction by the previous model after the guard regions.

over a previously seen surface in a new direction. There is some confusion between surfaces 2 and 3, but this is because of the similarity in their friction values.

Figure 6 shows the absolute error in trajectory tracking during this task and the corresponding stiffness parameters used by the controller. The peaks in the error plot correspond to the sudden change of surface. The prediction made by the model of the previous mode caused a momentary loss of trajectory tracking ability, until the robot switches to the high-stiffness mode for identifying the current mode. Once the robot identified the current mode, it used lower stiffness to complete the task. As discussed above, switching to a previously learned mode requires a much shorter period of high stiffness (and expends much less energy) compared with learning a new dynamics model from scratch. These results support hypothesis **H2**, and to some extent **H3**.

3.3 H2+H3: Different Types of Contacts

Next we conducted experiments with the changing contact type task. The robot had to slide an object along a trajectory on a surface under three different types of contacts—see top of Figure 7. The robot started with no prior knowledge of the task except the motion pattern provided by the single demonstration of the task. 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 triggers a transition; the robot proceeds to slide the object (in its grip) along the surface with the force of $10N$. This is initially done at a high stiffness if it is learning a new dynamics model, or at a suitably low stiffness if the transition is to an existing mode/model.

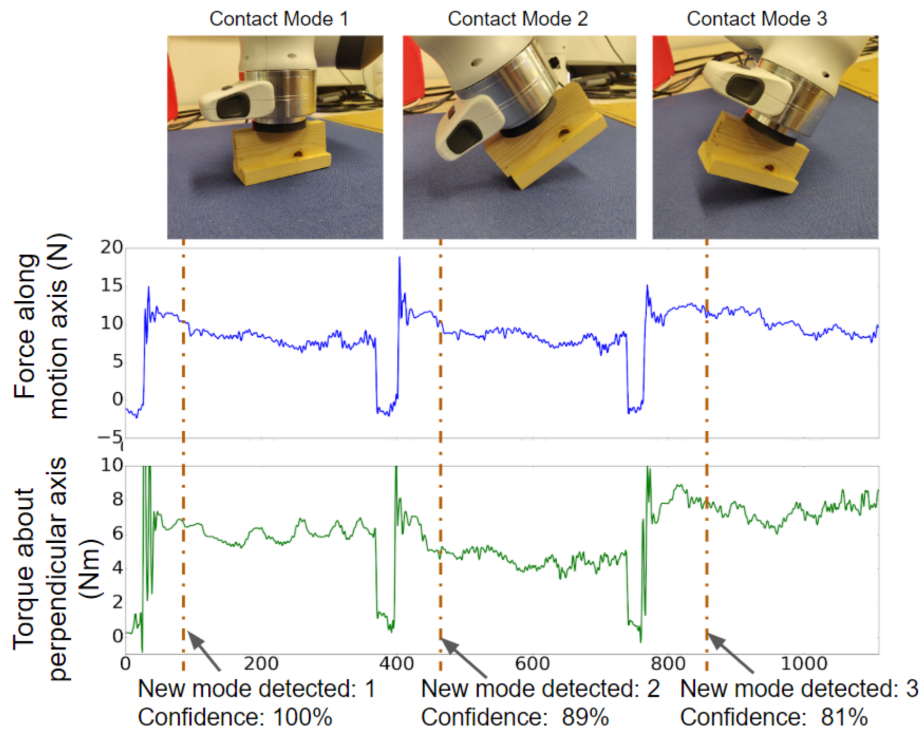

Figure 7: Initial trial of the changing contact type task. **Top:** The different contacts are shown such that the end-effector's motion is towards the right; **Middle:** The forces measured along the direction of motion; **Bottom:** Torques measured about axis parallel to surface and perpendicular to direction of motion. The spikes in the measurements correspond to contact transitions; dashed vertical lines (in brown) indicate when the framework has managed to learn a reliable dynamics model.

Figure 7 summarizes the learning of a model for each contact type, along with the variation of end-effector force and torque measured along the axis that is most affected by the motion. We observe that separate models are learned for the three contacts with significant confidence.

Next, Figure 8 shows that when the learned dynamics models were tested on the same task after changing the sequence of contacts, the robot was still able to recognize the contact modes accurately. The second plot in the figure shows the end-effector forces predicted by the dynamics model for the contact mode. The feed-forward term is used and revised online when the model is reliable, but the term is zero when the robot is yet to identify the mode. Similarly, the stiffness parameters of the impedance controller are varied according to the prediction error of the dynamics model; recall that a high stiffness is used when the dynamics model has not been learned.

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 model chosen with the highest confidence (bottom of the figure). *The identified modes match the true modes in all cases.*

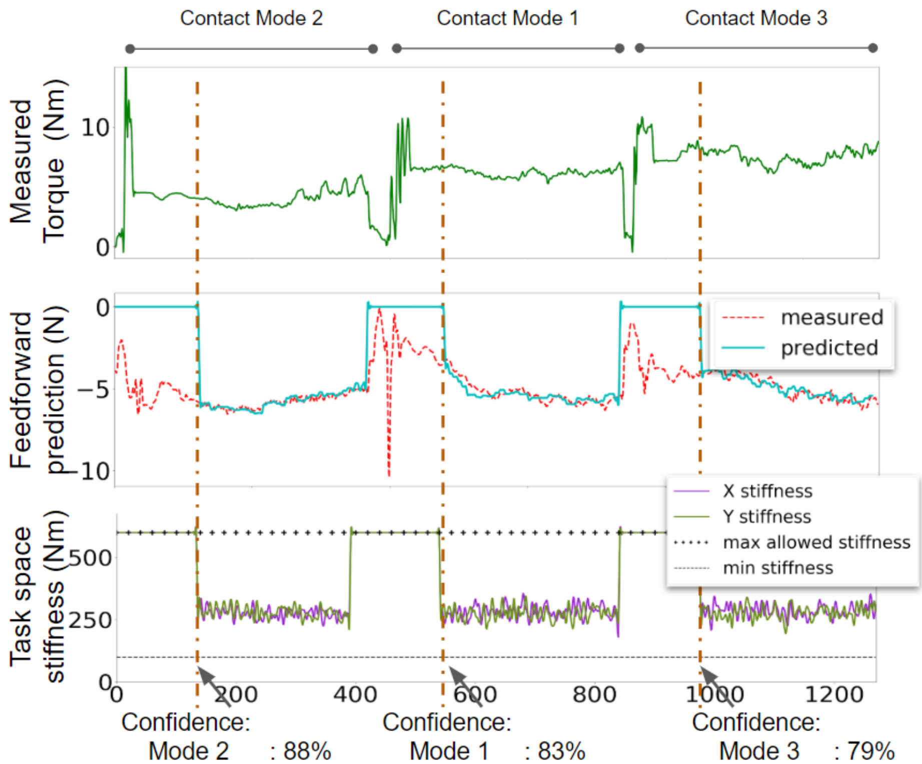

Figure 8: Testing previously learned dynamics models when the contacts appear in a different sequence. **Top:** Torques measured about the axis parallel to surface and perpendicular to direction of motion; The spikes in the measurements correspond to contact; **Middle:** End-effector forces predicted by the forward model for the current mode; **Bottom:** Variations in the controller stiffness due to the predicted forces.

The framework was then tested for the same task and contacts while applying a different constant normal force on the surface as the manipulator was sliding over the surface. The results are summarized in Figure 10. We observe the ability to identify modes and adapt the existing models of any given node during one trial of experimental evaluation. These results match those in Table 2; although the confidence associated with the modes is a little lower and the time taken to recognize the modes is a little more when the normal force is changed, the framework is still able to recognize the modes correctly and the task is completed successfully using variable impedance control. The lower confidence can be attributed to the kinetic friction assumption (i.e., that $\mu = F/R$) being unrealistic in many real world tasks. Note that the results in Figures 7–9 also indicate that the time taken to recognize the modes will be longer if the modes under consideration are similar, e.g., modes 1 and 3 are similar in these experiments. These results support hypotheses **H2** and **H3** and indicate the need for further research on the choice of features used to represent the modes.

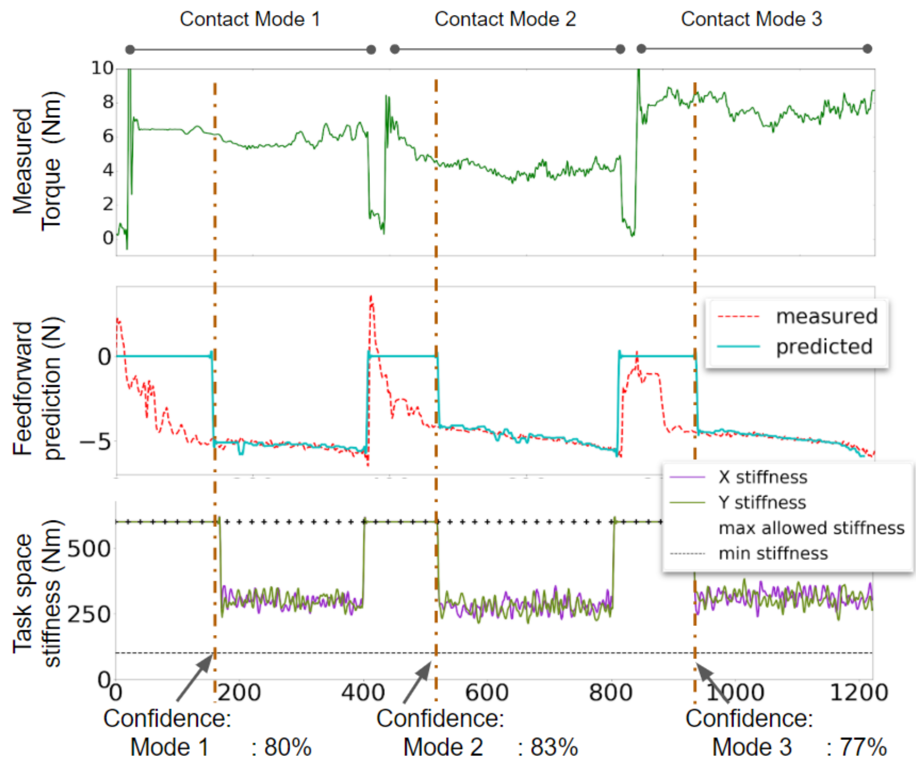

Figure 9: Testing the previously learned dynamics models for motion in a different direction. **Top:** Torques measured about the axis parallel to surface and perpendicular to direction of motion; **Middle:** End-effector forces predicted by the forward model for the current mode; **Bottom:** Controller stiffness variation due to the predicted forces.

4. Related Work

Many methods have been developed to address the learning and control 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, often 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).

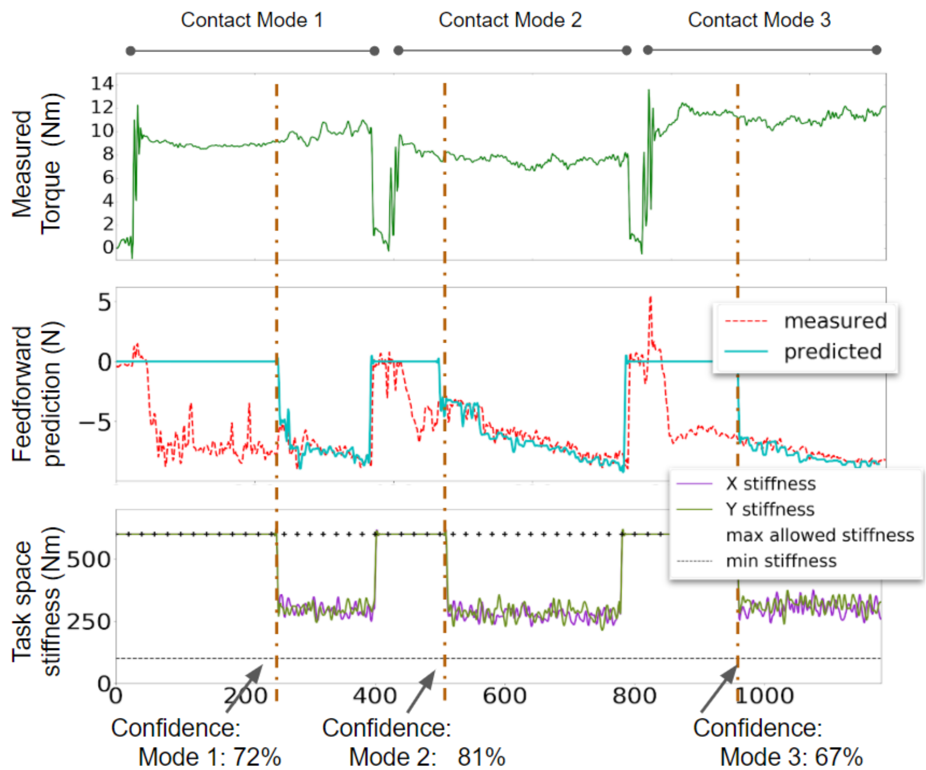

Figure 10: Testing the previously trained transition model with a different normal force (20N instead of 10N). **Top:** Torques measured about axis parallel to surface and perpendicular to direction of motion; **Middle:** End-effector forces predicted by the forward model for the current mode; **Bottom:** Controller stiffness variation due to the predicted forces.

RL and optimal control methods for robot manipulation often assume the task dynamics are smooth. The application of learning strategies to a hybrid systems 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 that models of the system, and knowledge of actions and modes, are known a priori. 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 learns to adapt 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.

	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

	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 ($\%$) across 10 trials associated with mode recognition based on the learned dynamics models for three types of contacts. **Top:** Normal force of $10N$; **Bottom:** Normal force $20N$.

In a departure from existing work, our framework for changing-contact manipulation draws inspiration from human motor control. It 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 create clusters for the modes of the task, identify mode changes, and incrementally learn and revise dynamics models for the modes. Unlike data-driven methods that require many labeled training examples, our framework is able to learn and revise the dynamics model for each observed mode from very few examples. Unlike existing control methods for related 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). Experimental results on a physical robot manipulator indicate the ability to reliably follow the desired motion trajectory on a surface 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 the manipulation problem such that it can be applied to different changing contact tasks such as peg-insertion, block pushing, stacking, etc. Also, the approach based on hybrid models may also be applied to other dynamics and control problems such as mobile robot navigation exploring and mapping different terrains, and aerial robots performing surveillance of different environments.

Our future work will address the limitations of the current framework and explore new directions. For instance, the current strategy of switching between modes (and dynamics models) is not smooth, with occasional spikes in sensor measurements in the guard (i.e., transition) regions. Also, we will explore tasks with many more modes, which may potentially require a functional formulation over the set of possible modes. Another direction for future research is to investigate other examples of changing-contact manipulation tasks, and additional factors that influence such tasks. In addition, it would also be interesting to explore the automatic selection (or learning) of the feature representation suitable for the modes of each changing-contact manipulation task. The longer-term objective is to enable reliable, efficient, and smooth learning and 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.